



Informatica™

Informatica® Intelligent Cloud Services
October 2023

Advanced Clusters

Informatica Intelligent Cloud Services Advanced Clusters
October 2023

© Copyright Informatica LLC 2020, 2024

Publication Date: 2024-04-18

Table of Contents

| | |
|--|-----------|
| Chapter 1: Advanced clusters..... | 7 |
| Advanced cluster types. | 8 |
| Fully-managed clusters. | 8 |
| Creating a fully-managed cluster. | 9 |
| Submitting jobs to a fully-managed cluster. | 9 |
| Stopping a fully-managed cluster. | 9 |
| Self-service clusters. | 10 |
| Local clusters. | 10 |
| Default local clusters. | 11 |
| Default staging and log locations. | 11 |
| | |
| Chapter 2: Setting up AWS..... | 13 |
| Step 1. Complete prerequisites. | 13 |
| Verify privileges in your organization. | 14 |
| Verify AWS subscriptions. | 14 |
| Learn about roles and policies in the AWS environment. | 14 |
| Learn about resource access. | 16 |
| Step 2. Create storage locations for cluster files. | 20 |
| Step 3. Create the VPC and subnets (optional). | 21 |
| Create subnets with enough IP addresses. | 21 |
| Verify the routing configuration. | 21 |
| Accept inbound traffic. | 21 |
| Step 4. Create user-defined security groups for Amazon EC2. | 22 |
| Create the ELB security group. | 22 |
| Create the master security group. | 23 |
| Create the worker security group. | 24 |
| Use default security groups (alternative). | 24 |
| Step 5. Download and install a Secure Agent. | 25 |
| Step 6. Allow domains in AWS. | 25 |
| Step 7. Create IAM roles. | 26 |
| Create the cluster operator role. | 26 |
| Create the cluster operator policy. | 27 |
| Attach the cluster operator policy. | 30 |
| Configure the maximum CLI/API session duration for the cluster operator role. | 30 |
| Create or reuse the Secure Agent role. | 30 |
| Add the AssumeRole permission to the Secure Agent role. | 31 |
| Configure the trust relationship for the cluster operator role to include the Secure Agent role. | 31 |
| Create user-defined master and worker roles. | 32 |
| Encrypt staging data and log files at rest (optional). | 42 |
| Create role-based security policies for Amazon data sources (optional). | 42 |

| | |
|--|-----------|
| Create or reuse a log access policy for the Secure Agent role. | 44 |
| Step 8. Configure environment variables (optional). | 46 |
| Step 9. Configure the Elastic Server. | 46 |
| IAM policy reference. | 48 |
| Cluster operator role actions. | 48 |
| Master role actions. | 55 |
| Worker role actions. | 58 |
| Master and worker role types reference. | 60 |
| Master and worker policy restriction reference. | 61 |
| Chapter 3: Setting up Google Cloud. | 63 |
| Step 1. Complete prerequisites. | 63 |
| Verify privileges in your organization. | 63 |
| Verify Google Cloud services. | 64 |
| Learn about resource access. | 64 |
| Learn about Google Cloud cluster. | 67 |
| Step 2. Create storage locations for cluster files. | 67 |
| Step 3. Create the VPC and subnets (optional). | 68 |
| Create a subnet with enough IP addresses. | 68 |
| Create a Google Cloud NAT gateway. | 68 |
| Create firewall rules in the VPC network. | 69 |
| Step 4. Download and install a Secure Agent. | 70 |
| Step 5. Allow domains in Google Cloud. | 71 |
| Step 6. Configure a proxy for the cluster (optional). | 71 |
| Step 7. Create roles and service accounts. | 72 |
| Create a Secure Agent role and service account. | 72 |
| Create a master role and service account. | 76 |
| Create a worker node role and service account. | 77 |
| Step 8. Configure the JAVA_HOME environment variable. | 77 |
| Step 9. Create a staging connection | 78 |
| Chapter 4: Setting up Microsoft Azure. | 79 |
| Step 1. Complete prerequisites. | 79 |
| Verify privileges in your organization. | 79 |
| Verify Microsoft Azure products. | 80 |
| Learn about resource access. | 80 |
| Step 2. Download and install a Secure Agent. | 83 |
| Step 3. Allow domains in Azure | 83 |
| Step 4. Configure a proxy for the cluster (optional). | 84 |
| Step 5. Create storage accounts for cluster files. | 84 |
| Step 6. Create a cluster resource group. | 85 |
| Step 7. Create a managed identity for the Secure Agent. | 85 |
| Create a managed identity. | 85 |

| | |
|---|------------|
| Create an agent role. | 85 |
| Add role assignments. | 88 |
| Step 8. Create a service principal for the cluster. | 88 |
| Create a service principal. | 89 |
| Create a cluster role. | 89 |
| Add a role assignment. | 91 |
| Store the credentials in a key vault. | 91 |
| Add an access policy to the key vault. | 91 |
| Step 9. Create a managed identity to access sources and targets (optional). | 91 |
| Step 10. Configure the JAVA_HOME environment variable (optional). | 92 |
| Step 11. Create a staging connection (optional). | 92 |
| | |
| Chapter 5: Setting up a self-service cluster. | 93 |
| Step 1. Complete prerequisites. | 93 |
| Create a self-service cluster. | 93 |
| Allow domains for self-service clusters. | 94 |
| Verify privileges in your organization. | 94 |
| Learn about resource access | 95 |
| Step 2. Download and install a Secure Agent. | 96 |
| Step 3. Configure a user-managed service account. | 96 |
| Create an optimized cluster role. | 97 |
| Create a service account and bind the cluster role. | 98 |
| Configure the service account in the advanced configuration. | 99 |
| Use an Informatica-managed service account (alternative). | 99 |
| Additional configuration for clusters on AWS. | 100 |
| Configure cluster authentication. | 100 |
| Configure cluster nodes with IMDSv2. | 100 |
| | |
| Chapter 6: Setting up a local cluster. | 101 |
| Prepare for local clusters. | 101 |
| Download and install a Secure Agent. | 102 |
| Troubleshoot a local cluster. | 103 |
| | |
| Chapter 7: Advanced configurations. | 104 |
| CLAIRE-powered configurations. | 105 |
| CLAIRE recommendations. | 106 |
| AWS properties. | 107 |
| Validating the configuration. | 111 |
| GPU worker instance type. | 112 |
| Graviton worker instance type. | 112 |
| Spot Instances. | 113 |
| High availability. | 114 |
| Accessing a new staging location. | 114 |

| | |
|---|------------|
| Propagating tags to cloud resources. | 115 |
| Default tags for cloud resources. | 115 |
| Data encryption. | 116 |
| Google Cloud properties. | 116 |
| Propagating labels to cloud resources. | 119 |
| Data encryption. | 119 |
| Microsoft Azure properties. | 120 |
| Validating the configuration. | 123 |
| Spot Instances. | 123 |
| High availability. | 125 |
| Accessing a new staging location. | 125 |
| Propagating tags to cloud resources. | 125 |
| Default tags for cloud resources. | 126 |
| Data encryption. | 126 |
| Local cluster advanced configuration. | 127 |
| Change staging and log locations (optional). | 127 |
| Local cluster properties. | 127 |
| Configure cloud permissions. | 130 |
| Data encryption. | 132 |
| Self-service cluster properties. | 133 |
| Runtime Properties. | 136 |
| Resource requirements for cluster nodes. | 136 |
| Reconfiguring resource requirements. | 137 |
| Resource requirements example. | 138 |
| Initialization scripts. | 138 |
| Initialization script failures. | 138 |
| Updating the runtime environment or the staging location. | 139 |
| Chapter 8: Troubleshooting. | 140 |
| Troubleshooting an advanced cluster. | 140 |
| Troubleshooting an advanced cluster on AWS. | 142 |
| Troubleshooting an advanced cluster on Microsoft Azure. | 144 |
| Troubleshooting an advanced cluster subtask. | 145 |
| Troubleshooting a self-service cluster. | 147 |
| Shutting down the Secure Agent machine and cloud resources. | 148 |
| Appendix A: Command reference. | 149 |
| generate-policies-for-userdefined-roles.sh. | 149 |
| list-clusters.sh. | 150 |
| delete-clusters.sh. | 151 |
| cluster-operations.sh. | 153 |
| Index. | 155 |

CHAPTER 1

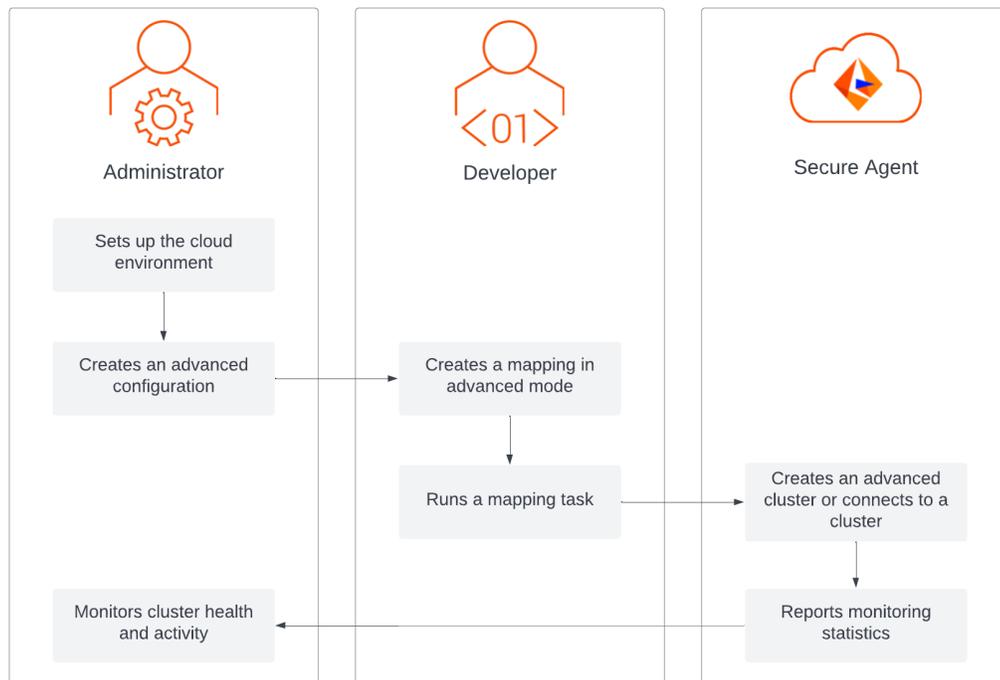
Advanced clusters

An advanced cluster is a Kubernetes cluster that provides a distributed processing environment on the cloud. Fully-managed and self-service clusters can run data logic using a scalable architecture, while local clusters use a single node to quickly onboard projects for advanced use cases.

To use an advanced cluster, you perform the following steps:

1. Set up your cloud environment so that the Secure Agent can connect to and access cloud resources.
2. In Administrator, create an advanced configuration to define the cluster and the cloud resources.
3. In Monitor, monitor cluster health and activity while developers in your organization create and run jobs on the cloud.

The following image shows the workflow that you use to get an advanced cluster up and running in your organization:



Advanced cluster types

Different types of advanced clusters are available to enable advanced functionality in mappings and help you choose the infrastructure that best supports your organization's processing requirements.

You can use the following types of advanced clusters in your organization:

Fully-managed cluster

A cluster that provides a serverless infrastructure that intelligently scales based on your workload and offers the lowest total cost of ownership for your organization. For more information, see [“Fully-managed clusters” on page 8](#).

Self-service cluster

A Kubernetes cluster that your organization runs and you reuse to run mappings. The Kubernetes cluster can run on either AWS or Microsoft Azure. For more information, see [“Self-service clusters” on page 10](#).

Local cluster

A simple, single-node cluster that you can start on the Secure Agent machine. You can use a local cluster to quickly onboard projects for advanced use cases. For more information, see [“Local clusters” on page 10](#).

Advanced cluster in a serverless runtime environment

If your organization uses AWS, you can create a serverless runtime environment that includes an advanced cluster for your organization to use. For more information, see *Runtime Environments*.

Fully-managed clusters

A fully-managed cluster provides a serverless infrastructure that intelligently scales based on your workload and offers the lowest total cost of ownership for your organization.

The Secure Agent manages the entire Kubernetes lifecycle, including cluster startup, shutdown, auto-scaling, and upgrade. The agent manages the compute infrastructure and can create the advanced cluster using Spot Instances to further reduce costs for your organization.

A fully-managed cluster includes the following capabilities:

- The cluster scales based on the size of the workload and the resource boundaries that you specify. Jobs consume fewer resources during smaller workloads, and the cluster accommodates bursts in the processing load.
- The cluster consumes resources only while you're running jobs. The Secure Agent determines when to stop the cluster based on the cluster shutdown method that you select in the advanced configuration.
- CLAIRE®, Informatica's AI engine, uses machine learning to auto-tune the jobs that run on the cluster to achieve optimal job performance.
- A secondary tuning process on the cluster analyzes the data size in a mapping and the cluster capacity to further auto-tune the job.
- The cluster allows you to configure permissions to set access limits on the Secure Agent to your environment.
- High availability, recovery, and resilience ensure that jobs can continue running smoothly during interruptions.
- The data remains in your cloud environment.

Creating a fully-managed cluster

When a developer runs a job, the Secure Agent uses the advanced configuration that's associated with the job's runtime environment to create a fully-managed cluster.

The agent performs the following tasks:

1. Creates a cluster configuration that includes configuration information about the cluster. The configuration is stored using YAML files that the Secure Agent populates.
2. Provisions the necessary resources to create a cluster.

Note: Informatica uses a secure pathway to fetch job-related container images for cluster nodes from the Informatica-specific JFrog repository. For clusters on Google Cloud, it also accesses the public internet to fetch files that are required to create the logical cluster layer on cluster nodes.

Submitting jobs to a fully-managed cluster

When the fully-managed cluster is running, the Secure Agent submits jobs to run on the cluster.

To submit a job to the cluster, the Secure Agent generates an execution plan that divides the data logic in the mapping into multiple Spark tasks. The cluster launches Spark drivers and Spark executors to process the Spark tasks simultaneously.

As developers run additional jobs, the cluster provisions and deprovisions resources to adapt to the size and number of jobs. For example, the cluster can provision additional cluster nodes and cluster storage during processing bursts.

Each job generates a session log, a Spark driver log, Spark executor logs, and an agent job log.

Stopping a fully-managed cluster

The Secure Agent stops a fully-managed cluster based on the cluster shutdown method that you select in the advanced configuration.

The Secure Agent can either wait to shut down the cluster after an idle timeout, or the agent can perform a smart shutdown that is based on historical data.

The Secure Agent also stops the cluster in the following situations:

- The cluster fails to start or fails to stop.
- The agent cannot reach the Kubernetes API server within a certain amount of time.

After the Secure Agent stops the cluster, the agent verifies that all cluster resources are deleted, except for some Informatica binaries that remain in the staging location in the `infa_rpm.tar` file. The binaries are required in order to run jobs on the cluster, and the file is reused the next time that the agent starts the cluster.

The agent deletes the `infa_rpm.tar` file in the following situations:

- You clear the runtime environment in the advanced configuration.
- You associate the advanced configuration with a different runtime environment.
- The agent process on the Secure Agent machine shuts down.

The agent restarts the cluster when a developer runs another job in the same runtime environment.

Self-service clusters

A self-service cluster is a Kubernetes cluster that your organization runs and you reuse to run mappings.

The Kubernetes cluster can run on AWS or Microsoft Azure. You can use self-managed clusters, including Amazon EC2 instances and Azure Virtual Machines, or you can use service-managed clusters, including Amazon Elastic Kubernetes Service and Azure Kubernetes Service.

Using a self-service cluster gives you finer controls over the compute environment by providing isolation through namespaces, contexts, annotations, and tolerations. Because you manage the cluster, the Secure Agent needs minimal permissions in your environment.

Compared to a fully-managed cluster, a self-service cluster provides the following advantages:

- You have more control over the cluster control plane.
- You have full access to the cluster and manage all components.
- You have more control over the deployment and administration of your cluster. For example, you can implement multiple node groups or use different instance types for different nodes.

To connect the Secure Agent to a self-service cluster, use the kubeconfig file that is generated for the cluster. The kubeconfig file is a YAML file that contains the cluster configuration. Enter the path to the kubeconfig file in the advanced configuration so that the Secure Agent can connect to the self-service cluster and submit jobs to the cluster.

Because the Secure Agent doesn't manage the cluster, the Secure Agent doesn't scale the cluster based on the workload. When you shut down the cluster, the Secure Agent removes all job-related resources from the cluster.

Local clusters

A local cluster is a simple, single-node cluster that you can start on the Secure Agent machine. You can use a local cluster to quickly onboard projects for advanced use cases. A local cluster can only run mappings in advanced mode.

A local cluster can run on-premises or in the following cloud environments:

- AWS
- Google Cloud
- Microsoft Azure
- Oracle Cloud Infrastructure

You can set up a local cluster on a virtual machine with minimal permissions and resource requirements.

Note: When you install the Secure Agent on Oracle Cloud Infrastructure, you can create local clusters but you can't create any other types of advanced clusters.

The local cluster has a single node with processing capacity that depends on the local machine. The cluster can access staging and log locations on the cloud or in local storage that is attached to the cluster node. The local cluster times out after five minutes if there are no jobs running on the cluster.

Before you run mappings in advanced mode on a local cluster, make sure that the agent has enough resources so that it can create a cluster and run jobs successfully, especially if the agent is already running other jobs. If the agent doesn't have enough resources, the jobs that are already running on the agent and the

mappings in advanced mode will fail. It's recommended to have at least 8 cores and 32 GB of memory on the agent machine.

Default local clusters

The agent can create a default local cluster on the agent machine so that you can begin developing and running advanced functionality on small data sets to test mapping logic.

When you run a mapping in advanced mode using an agent that's not associated with an advanced configuration, a default advanced configuration is created and associated with the agent. The agent can use the default configuration to create a default local cluster that can process the mapping.

A default advanced configuration is created in the following situations:

- You run a mapping in advanced mode.
- You create a mapping task based on a mapping in advanced mode and select a runtime environment.
- You preview data in a mapping in advanced mode.

You can view the advanced configuration for the default cluster on the **Advanced Clusters** page in Administrator. You can edit the configuration to modify the staging location, log location, mapping task timeout, and runtime properties. You can also monitor the default local cluster on the **Advanced Clusters** page in Monitor.

A default advanced configuration is not created if the operating system on the agent machine can't host a local cluster. In this case, you need to manually create an advanced configuration in Administrator and associate the advanced configuration with the runtime environment.

When your organization is ready to run mappings to process production-scale workloads, complete the following tasks:

1. Set up your cloud environment to host a larger advanced cluster.
2. Create an advanced configuration for the cluster.
3. Edit the default advanced configuration and dissociate it from the runtime environment.
4. Edit the new advanced configuration and associate it with the runtime environment.

A larger cluster can also resolve memory and performance issues that developers encounter during testing.

Default staging and log locations

A default local cluster stores staging and log files on the agent machine when you run jobs on the cluster.

The following table lists the default staging and log locations:

| Default location | Description |
|----------------------------------|--|
| file:/// \$ADVANCED_MODE_STAGING | Default staging location in the following directory on the agent machine: <code><agent installation directory>/apps/ Advanced_Mode_Staging_Dir</code> |
| file:/// \$ADVANCED_MODE_LOG | Default log location in the following directory on the agent machine: <code><agent installation directory>/apps/Advanced_Mode_Log_Dir</code> |

The agent machine must have enough space in the default staging and log locations so that jobs can run successfully. You can change the staging and log locations by editing the advanced configuration for the cluster.

If Data Integration creates a subtask for the Data Integration Server to process data logic in a mapping in advanced mode, the cluster and the Data Integration Server share staging files. To read and write staging files in the staging location, the Data Integration Server uses Hadoop Files V2 Connector.

CHAPTER 2

Setting up AWS

Before you create an advanced configuration in your organization, set up your cloud environment so that the Secure Agent can create an advanced cluster.

Complete the following tasks:

1. Complete the prerequisites. Verify that you have the necessary privileges and learn about resource access in the cloud environment.
2. Create storage locations for cluster files. The advanced cluster requires Amazon S3 storage to store staging, log, and initialization script files.
3. Optionally, create a VPC and subnets. If you don't create a VPC and subnets and specify them in your advanced configuration, the cluster creates a default VPC and subnet when you run a job on the cluster.
4. Create user-defined security groups for Amazon EC2. Security groups define inbound and outbound rules for traffic into and out of the load balancer, master nodes, and worker nodes. You can also use default security groups instead of user-defined security groups.
5. Download and install the Secure Agent on a Linux virtual machine on Amazon EC2. Set up the agent on a virtual machine that meets the minimum resource requirements.
6. Allow domains in AWS. The cluster requires to access certain domains to fetch artifacts and to access sources and targets.
7. Create IAM roles. The cluster operator, Secure Agent, master nodes, and worker nodes use IAM roles and policies to provide authentication when the cluster runs a job.
8. Optionally, configure environment variables on the Secure Agent machine. Some environment variables are required to run shell commands.
9. Configure the Elastic Server. The Elastic Server manages the advanced cluster and the jobs that run on the cluster.

Note: In an AWS environment, you can use a serverless runtime environment instead of performing these tasks and creating an advanced configuration. For more information, see *Runtime Environments*.

Step 1. Complete prerequisites

Before you set up your environment, verify the requirements for your environment and your cloud platform.

Complete the following tasks:

- Verify that you have the correct privileges in your organization.
- Verify that you have the necessary AWS subscriptions.
- Learn about the roles and policies in your environment.

- Learn how the Secure Agent and the advanced cluster access resources on your cloud platform.

Verify privileges in your organization

Verify that you are assigned the correct privileges for advanced configurations in your organization.

Privileges for advanced configurations provide you varying access levels to the **Advanced Clusters** page in Administrator as well as Monitor.

You must have at least the read privilege to view the advanced configurations and to monitor the advanced clusters.

Verify AWS subscriptions

Verify that you have the necessary AWS subscriptions to create an advanced cluster in an AWS environment.

You must have the following services on AWS:

Amazon Elastic Block Service (Amazon EBS)

Amazon EBS volumes are attached to Amazon EC2 instances as local storage. The local storage is used to store information that the Serverless Spark engine needs to run advanced jobs. For example, local storage is used to store the content of the Spark image. The Spark engine also requires local storage to process data logic and to persist data during processing.

Amazon Elastic Compute Cloud (Amazon EC2)

Amazon EC2 instances are launched to host an advanced cluster. One Amazon EC2 instance hosts the master node, and additional instances host the worker nodes.

Amazon EC2 Auto Scaling

Amazon EC2 Auto Scaling automatically adds or removes cluster nodes in the advanced cluster based on job-processing requirements.

Amazon Elastic Load Balancing (Amazon ELB)

A load balancer accepts incoming advanced jobs from a Secure Agent and provides an entry point for the jobs to an advanced cluster.

Amazon Identity and Access Management (IAM)

AWS IAM provides access control that you can use to specify which services and resources an advanced cluster can access in your AWS environment.

Amazon Route 53

Nodes in an advanced cluster communicate information with other nodes in the same cluster using Route 53.

Amazon Simple Storage Service (Amazon S3)

An advanced cluster is staged in Amazon S3 buckets. Amazon S3 is also used to store logs that are generated for advanced jobs.

Learn about roles and policies in the AWS environment

The Secure Agent and the advanced cluster use IAM roles and the IAM policies that you attach to those roles to access and process data in an AWS environment. For example, the agent and the cluster use the roles to

manage cloud resources such as EC2 instances and to access data on S3 like staging, log, and initialization script files.

Roles

An AWS environment uses the following IAM roles:

Cluster operator role

The cluster operator role is an IAM role that has elevated permissions to manage the cloud resources that host an advanced cluster.

Secure Agent role

The Secure Agent role is an IAM role for the Secure Agent. This IAM role is attached to the Secure Agent machine which is the Amazon EC2 instance where the Secure Agent runs.

The Secure Agent uses the Secure Agent role to assume the cluster operator role to manage an advanced cluster. The Secure Agent also uses the Secure Agent role to process jobs and access some resources on the cloud.

Master role

The master role is an IAM role that defines the permissions for the master nodes in an advanced cluster.

Worker role

The worker role is an IAM role that defines the permissions for the worker nodes in an advanced cluster.

For more information about the roles, see [“Step 7. Create IAM roles” on page 26](#).

Policies

Each IAM role uses one or more IAM policies.

The following table describes the policies and the roles that use each policy:

| Policy | Used by role | Description |
|----------------------------------|----------------------------------|---|
| cluster_operator_policy | Cluster operator role | Required. Provides the minimal access permissions to create and manage cloud resources for an advanced cluster. |
| assume_role_agent_policy | Secure Agent role | Required. Allows the Secure Agent to use the Secure Agent role to assume the cluster operator role. |
| data_source_access_policy | Secure Agent role Worker role | Required if you use role-based security for Amazon data sources and want to create a unique policy. Provides access to the Amazon data sources in an advanced job. |
| log_access_agent_policy | Secure Agent role | Required if you do not configure a trust relationship between the Secure Agent role and worker role. Provides access to the log location to upload the agent job log at the end of an advanced job. |
| minimal_master_policy | Master role | Required. Provides the minimal access permissions for the master role. |
| staging_log_access_master_policy | Master role | Required. Provides access to the staging and log locations. |

| Policy | Used by role | Description |
|----------------------------------|--------------|--|
| init_script_master_policy | Master role | Required only if you use an initialization script. Provides access to the initialization script path and the location that stores init script and cloud-init logs. |
| minimal_worker_policy | Worker role | Required. Provides the minimal access permissions for the worker role. |
| ebs_autoscaling_worker_policy | Worker role | Required only if EBS volumes auto-scale. Provides permissions to auto-scale the EBS volumes. |
| staging_log_access_worker_policy | Worker role | Required. Provides access to the staging and log locations. |
| init_script_worker_policy | Worker role | Required only if you use an initialization script. Provides access to the initialization script path and the location that stores init script and cloud-init logs. |

Learn about resource access

To process data, the Secure Agent and the advanced cluster access the resources that are part of an advanced job, including resources on the cloud platform, source and target data, and staging and log locations.

Resources are accessed to perform the following tasks:

- Design a mapping
- Create an advanced cluster
- Run a job, including data preview
- Poll logs

Designing a mapping

When you design a mapping, the Secure Agent accesses sources and targets so that you can read and write data.

For example, when you add a Source transformation to a mapping, the Secure Agent accesses the source to display the fields that you can use in the rest of the mapping. The Secure Agent also accesses the source when you preview data.

The Secure Agent accesses sources and targets based on the type of connectors that the job uses:

Connectors with direct access to Amazon data sources

If the mapping uses a connector with direct access to Amazon data sources, the Secure Agent uses role-based security or credential-based security to access the source or target. For role-based security, the Secure Agent uses the Secure Agent role to access data sources. If you specify an IAM role at the connection level, the agent assumes the connection-level role to access the data sources at run time. For credential-based security, the Secure Agent accesses the source or target through connection-level AWS credentials.

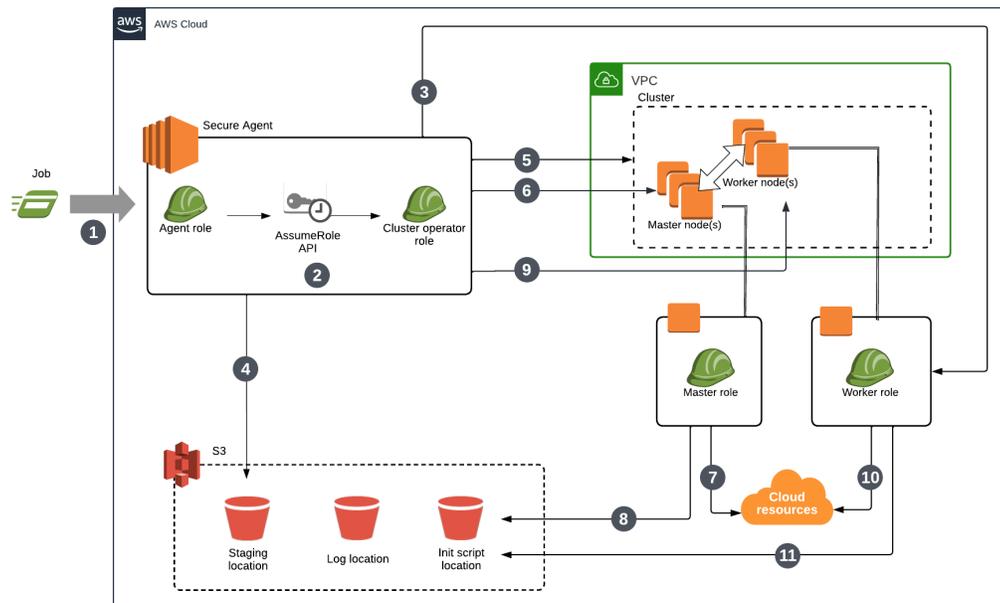
Connectors without direct access to Amazon data sources

If the mapping does not use a connector with direct access to Amazon data sources, the Secure Agent uses the connection properties to access the source or target. For example, the Secure Agent might use the user name and password that you provide in the connection properties to access a database.

Creating an advanced cluster

To create an advanced cluster, the Secure Agent uses the cluster operator role to store cluster details in the staging location and to create the cluster. The master and worker nodes use the master and worker roles to access cloud resources.

The following image shows the process that the Secure Agent uses to create a cluster:



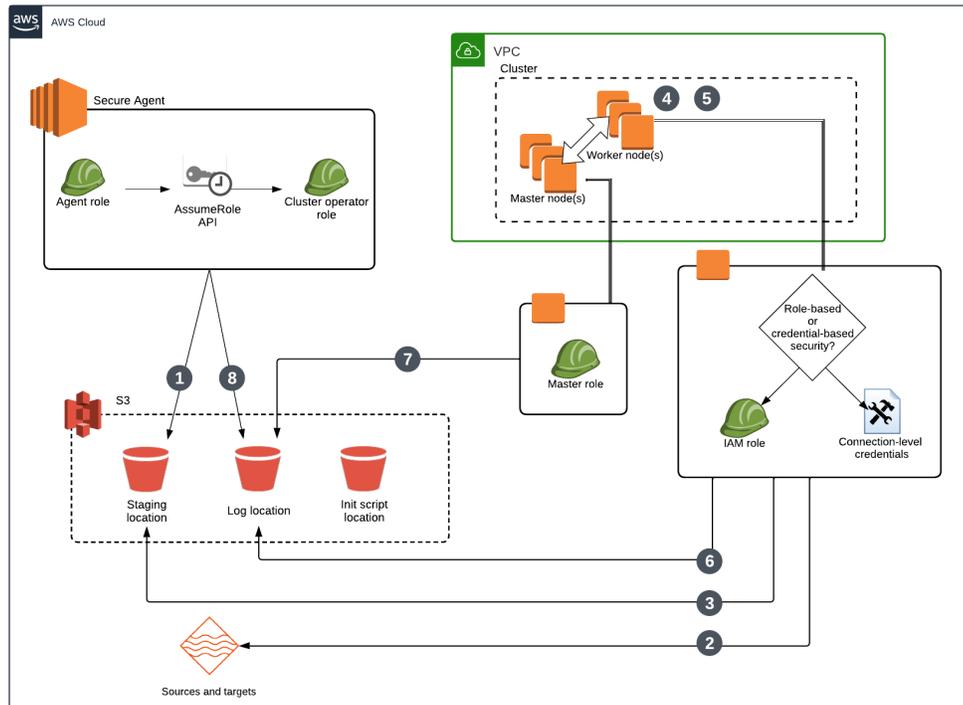
The following steps describe the process that the Secure Agent uses to create a cluster:

1. You run a job.
2. The Secure Agent assumes the cluster operator role to gain elevated privileges on AWS. The cluster operator role allows the Secure Agent to assume the master and worker roles.
3. If you create a user-defined worker role, the Secure Agent uses the worker role and verifies that the cluster can access staging and log locations.
4. The Secure Agent uses the cluster operator role to store cluster details in the staging location.
5. The Secure Agent uses the cluster operator role to create the cluster.
6. The Secure Agent uses the cluster operator role to create cluster resources for the master node.
7. The master node uses the master role to access cloud resources.
8. The master node uses the master role to access the initialization script.
9. The Secure Agent uses the cluster operator role to create cluster resources for the worker nodes and creates an Auto Scaling group with the minimum number of worker nodes.
10. The worker nodes use the worker role to access cloud resources.
11. The worker nodes use the worker role to access the initialization script.

Running a job with direct access to Amazon data sources

To run a job that uses a connector with direct access to Amazon data sources, the cluster accesses Amazon resources using role-based security or credential-based security.

The following image shows the process that the Secure Agent and cluster nodes use to run the job:



The following steps describe the process that the Secure Agent and cluster nodes use to run the job:

1. The Secure Agent assumes the cluster operator role to store job dependencies in the staging location.
2. The worker nodes use the connection-level role, the worker role, or connection-level AWS credentials to access source data based on the job security type. If you use role-based security, the worker nodes use the connection-level role or the worker role. If you use credential-based security, the worker nodes use the connection-level credentials. The authentication configured at the connection level takes precedence.
3. The worker nodes use the connection-level role, worker role, or connection-level credentials to access the staging location to get job dependencies and stage temporary data.
4. The worker nodes use the worker role to auto-scale EBS volumes if the job requires more storage space.
5. The master node uses the master role to scale cluster nodes based on resource requirements.
6. The worker nodes use the worker role to store logs in the log location.
7. The master node uses the master role to store logs in the log location.
8. The Secure Agent uses the Secure Agent role to upload the agent job log to the log location.

Security types

Worker nodes access Amazon resources in the following ways based on the security type:

Credential-based security

If you set up credential-based security, worker nodes use connection-level AWS credentials to access Amazon resources, including Amazon data sources and the staging location. The worker nodes use the worker role to access the log location.

Credential-based security overrides role-based security. If any source or target in the job provides AWS credentials, the worker nodes reuse the credentials to access the staging location. For example, if a job uses a JDBC V2 source and an Amazon S3 V2 target, the worker nodes use the AWS credentials that access the S3 target to access the staging location for the job.

Role-based security

If you set up role-based security, worker nodes use either the connection-level role or the worker role to access Amazon resources, including Amazon data sources, the staging location, and the log location. The role configured at the connection level takes precedence over the worker role.

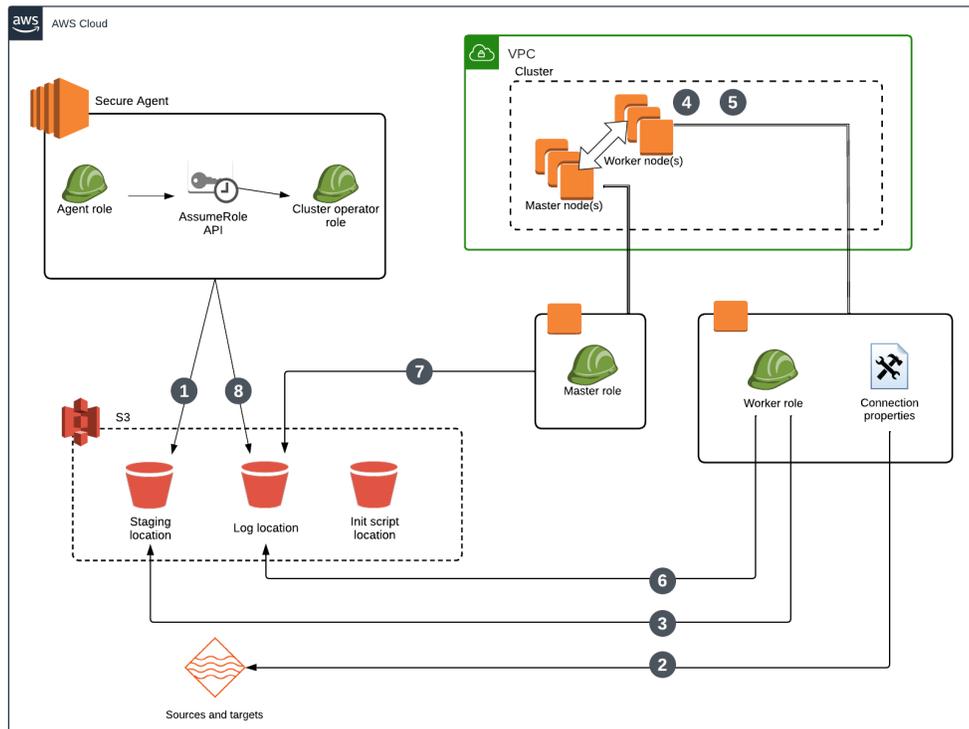
Note: If you use default master and worker roles, the policies that are attached to the Secure Agent role are passed to the worker role. The policies that are passed to the worker role can grant the worker role access to Amazon resources.

Running a job without direct access to Amazon data sources

To run a job that doesn't use a connector with direct access to Amazon data sources, the cluster accesses Amazon resources using the connection properties and the worker role.

For example, JDBC V2 Connector doesn't have direct access to Amazon data sources. To run a job that uses JDBC V2 Connector, the cluster uses the connection properties to read and temporarily stage the data before processing and writing the data to the target.

The following image shows the process that the Secure Agent and cluster nodes use to run the job:



The following steps describe the process that the Secure Agent and cluster nodes use to run the job:

1. The Secure Agent assumes the cluster operator role to store job dependencies in the staging location.
2. The worker nodes use connection properties to access source data.
3. The worker nodes use the worker role to access the staging location to get job dependencies and stage temporary data.
4. The worker nodes use the worker role to auto-scale EBS volumes if the job requires more storage space.
5. The master node uses the master role to scale cluster nodes based on resource requirements.
6. The worker nodes use the worker role to store logs in the log location.
7. The master node uses the master role to store logs in the log location.
8. The Secure Agent uses the Secure Agent role to upload the agent job log to the log location.

Note: If any connector in the job uses AWS credentials to directly access a source or target, the connection-level AWS credentials override the worker role to gain access to the staging location.

Polling logs

When you use Monitor, the Secure Agent accesses the log location to poll logs.

The Secure Agent polls logs based on the type of connectors that the job uses:

Connectors with direct access to Amazon data sources

If the job uses a connector with direct access to Amazon data sources, the Secure Agent uses either credential-based security or role-based security to access the log location. For credential-based security, the Secure Agent polls logs through the connection-level AWS credentials. For role-based security, the Secure Agent polls logs through the permissions in the Secure Agent role.

Connectors without direct access to Amazon data sources

If the job does not use a connector with direct access to Amazon data sources, the Secure Agent polls logs through the permissions in the Secure Agent role.

Step 2. Create storage locations for cluster files

In Amazon S3, create locations to store staging, log, and initialization script files.

Create the following storage locations:

- A location that the cluster will use to store staging files at run time
- A location that the cluster will use to store log files for the advanced jobs that run on the cluster
- Optionally, a location where you can store initialization scripts that cluster nodes will run to install additional software on the cluster

The staging location stores temporary data, such as artifacts that the cluster distributes across cluster nodes and data that you preview in a mapping. Because an error might prevent a mapping from clearing preview data in the staging location, make sure that the users who have access to the staging location are permitted to view source data.

If you create any initialization scripts, add the scripts to the appropriate location.

Step 3. Create the VPC and subnets (optional)

If you create your own VPC and subnets to host an advanced cluster, prepare the VPC and subnets according to cluster requirements.

Complete the following tasks:

- Create subnets that support enough IP addresses to assist an elastic load balancer and the nodes in the advanced cluster.
- Verify the routing configuration to make sure that the VPC and subnets can route requests in the cluster.
- Accept inbound traffic on the Secure Agent machine so that the Spark driver can communicate with the Secure Agent.

Create subnets with enough IP addresses

Create subnets that support enough IP addresses to assist an elastic load balancer and the nodes in the advanced cluster.

For each subnet, calculate the number of required IP addresses according to the following guidelines:

1. Add eight IP addresses to make sure that the elastic load balancer can scale properly.
2. Add one IP address for the master node. If you want to use a cluster that is highly available, add 3 IP addresses instead.
3. Add IP addresses equal to the maximum number of worker nodes.

For example, if the advanced cluster can have a maximum of 10 worker nodes, each subnet must support at least 19 IP addresses.

Verify the routing configuration

Verify that the VPC and subnets can route requests in an advanced cluster.

To make sure that the VPC and subnets can route requests, verify the following items on AWS:

- The VPC contains all necessary networking components, including a route table, an internet gateway, and a network ACL.
- DNS hostnames and DNS resolution are enabled.
- The route table allows any EC2 instance to use the internet gateway that is attached to the VPC.

For more information, refer to the AWS documentation.

Accept inbound traffic

Accept inbound traffic on the Secure Agent machine so that the Spark driver can communicate with the Secure Agent.

Complete the following tasks:

1. Add an inbound rule to the AWS security group that is attached to the Secure Agent machine.
2. Specify the port 0-65535 to accept inbound traffic.
3. Specify the VPC in CIDR notation.

Step 4. Create user-defined security groups for Amazon EC2

Create ELB, master, and worker security groups to fine-tune security settings in your AWS environment. Configure the appropriate inbound and outbound rules for each security group. After you complete these tasks, you can specify the security groups in an advanced configuration.

If you're looking for a quick setup, you can use the default security groups that the Secure Agent creates. For more information, see ["Use default security groups \(alternative\)" on page 24](#). You cannot mix and match default and user-defined security groups. For example, if you create a user-defined ELB security group, you must also create user-defined master and worker security groups.

For detailed instructions about how to create security groups for Amazon EC2, refer to the AWS documentation.

Create the ELB security group

The ELB security group defines the inbound rules between the Kubernetes API server and clients that are external to the advanced cluster. It also defines the outbound rules between the Kubernetes API server and cluster nodes. This security group is attached to the load balancer that the agent provisions for the advanced cluster.

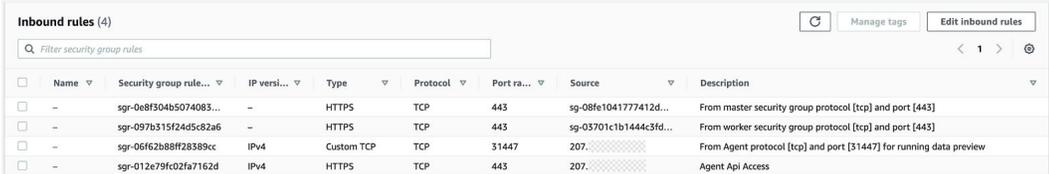
Inbound rules

The inbound rules identify the nodes outside of the advanced cluster that can access the Kubernetes API server using HTTPS.

The inbound rules must allow the following traffic:

- Incoming traffic from the Secure Agent that creates the advanced cluster.
- Incoming traffic from master nodes in the same cluster.
- Incoming traffic from worker nodes in the same cluster.
- Incoming traffic from the Secure Agent using TCP port 31447. The Secure Agent uses this port to run data preview jobs. If you need to change this port number, contact Informatica Global Customer Support.

The following image shows the required inbound rules:



| Name | Security group rule | IP version | Type | Protocol | Port range | Source | Description |
|------|------------------------|------------|------------|----------|------------|------------------------|---|
| - | sgr-0e8f304b5074083... | - | HTTPS | TCP | 443 | sg-08fe1041777412d... | From master security group protocol [tcp] and port [443] |
| - | sgr-097b515f24d5c82a6 | - | HTTPS | TCP | 443 | sg-03701c1b1444c3fd... | From worker security group protocol [tcp] and port [443] |
| - | sgr-06162b88ff28389cc | IPv4 | Custom TCP | TCP | 31447 | 207.... | From Agent protocol [tcp] and port [31447] for running data preview |
| - | sgr-012e79fc02fa7162d | IPv4 | HTTPS | TCP | 443 | 207.... | Agent Api Access |

Outbound rules

Use the default outbound rule to allow all outbound traffic.

You can restrict the destination of this rule, but the destination must include HTTPS traffic to all master nodes in the cluster.

Create the master security group

The master security group defines the inbound rules between the master nodes and the worker nodes in the advanced cluster, the ELB security group, and the Secure Agent. It also defines outbound rules to other nodes. This security group is attached to all master nodes in the cluster.

Inbound rules

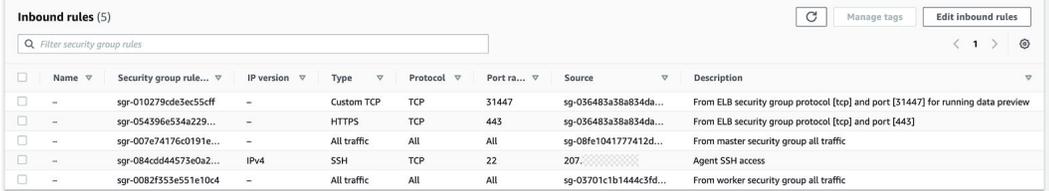
Inbound rules must allow the following traffic:

- Incoming traffic from worker nodes in the same cluster. For example, worker nodes accessing the API server through the service named "kubernetes," or kube-proxy forwarding the network traffic inside or outside the cluster. You can simplify the inbound rules for worker nodes by configuring the rule for custom TCP and UDP with port range 1024 - 65535, as well as HTTPs with TCP at port 443.
- Incoming traffic from other master nodes in the same cluster.
- Incoming traffic using HTTPS over TCP at port 443 from the ELB security group in the same cluster.
- Incoming traffic using SSH over port 22.
- Incoming traffic using TCP port 31447, which is from the ELB security group in the same cluster. The Secure Agent uses this port to run data preview jobs.

When you create and use a user-defined master security group, the Secure Agent ignores the following default rules for SSH access from outside the cluster:

- The IP address of the Secure Agent, from where the cluster is created, can use the SSH protocol to connect to worker nodes through port 22.
- The ability to configure the source Classless Inter-Domain Routing (CIDR) address using a custom property.
- The configuration of SSH port using a custom property.
- The ability to set a local file path on an agent node for a public key using a custom property.

The following image shows the required inbound rules:



| <input type="checkbox"/> | Name | Security group rule... | IP version | Type | Protocol | Port ra... | Source | Description |
|--------------------------|------|------------------------|------------|-------------|----------|------------|------------------------|--|
| <input type="checkbox"/> | - | sgr-010279cde3ec55cfff | - | Custom TCP | TCP | 31447 | sg-036483a38a834da... | From ELB security group protocol (tcp) and port [31447] for running data preview |
| <input type="checkbox"/> | - | sgr-054396e54a2229... | - | HTTPS | TCP | 443 | sg-036483a38a834da... | From ELB security group protocol (tcp) and port [443] |
| <input type="checkbox"/> | - | sgr-007e74176c0191e... | - | All traffic | All | All | sg-08fe1041777412d... | From master security group all traffic |
| <input type="checkbox"/> | - | sgr-084cdd44573e0a2... | IPv4 | SSH | TCP | 22 | 207.172.153.100 | Agent SSH access |
| <input type="checkbox"/> | - | sgr-0082f35e551e10c4 | - | All traffic | All | All | sg-03701c1b1444c3fd... | From worker security group all traffic |

Outbound rules

Use the default outbound rule to allow all outbound traffic.

Outbound traffic from the master node can include the other master nodes; the ELB security group; worker nodes; Secure Agents; other managed services on AWS such as Amazon S3, EC2, and IAM; other storage services; and other public services.

Create the worker security group

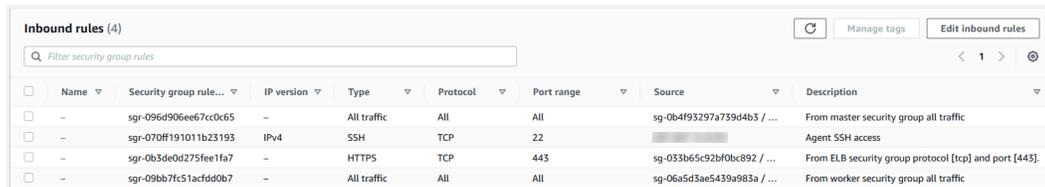
The worker security group defines the inbound and outbound rules between worker nodes in the advanced cluster and other nodes. This security group is attached to all worker nodes in the cluster.

Inbound rules

Inbound rules must allow the following traffic:

- Incoming traffic from other worker nodes in the cluster. For example, communication between related Pods.
- Incoming traffic from any master node in the cluster. For example, the master node contacts the kubelet on worker nodes to get logs or support port forwarding.
- Incoming traffic using HTTPS with TCP at port 443 from the ELB security group in the same cluster.
- Incoming SSH access from outside the cluster. This rule is the same as the SSH inbound rule defined for the master security group and is needed only if you want to access the worker node using SSH.

The following image shows the required inbound rules:



| <input type="checkbox"/> | Name | Security group rule... | IP version | Type | Protocol | Port range | Source | Description |
|--------------------------|------|------------------------|------------|-------------|----------|------------|----------------------------|--|
| <input type="checkbox"/> | - | sg-096d906ee67cc0c65 | - | All traffic | All | All | sg-0b4f93297a739d4b3 / ... | From master security group all traffic |
| <input type="checkbox"/> | - | sg-070ff191011b23193 | IPv4 | SSH | TCP | 22 | | Agent SSH access |
| <input type="checkbox"/> | - | sg-0b3de0d275fee1fa7 | - | HTTPS | TCP | 443 | sg-033b65c92bf0bc892 / ... | From ELB security group protocol [tcp] and port [443]. |
| <input type="checkbox"/> | - | sg-09bb7fc51acfd0b7 | - | All traffic | All | All | sg-06a5d3ae5439a983a / ... | From worker security group all traffic |

Outbound rules

Use the default outbound rule to allow all outbound traffic.

Outbound traffic from worker nodes can include the ELB security group; master nodes; other worker nodes; the Secure Agent; other managed services on AWS such as Amazon S3, EC2, and IAM; other storage services; and other public services. Additionally, the outbound rules must allow advanced jobs to communicate with data sources, such as Redshift and Snowflake databases, and external services, such as REST endpoints that the Secure Agent exposes.

Use default security groups (alternative)

When the Secure Agent creates an advanced cluster, it can generate a default ELB security group, master security group, and worker security group. These default security groups define communication guidelines between Kubernetes clients, the API server, master nodes, worker nodes, and other services.

To allow the Secure Agent to generate the default security groups, the cluster operator policy for the cluster operator role requires the following permissions:

```
ec2:DescribeSecurityGroups
ec2:CreateSecurityGroup
ec2>DeleteSecurityGroup
ec2:AuthorizeSecurityGroupEgress
ec2:AuthorizeSecurityGroupIngress
ec2:RevokeSecurityGroupEgress
ec2:RevokeSecurityGroupIngress
```

For more information about the cluster operator role and the cluster operator policy, see [“Step 7. Create IAM roles” on page 26](#).

Step 5. Download and install a Secure Agent

Download and install a Secure Agent on a Linux virtual machine on an Amazon EC2 instance. This EC2 instance is known as the Secure Agent machine.

The following table lists the minimum resource requirements on the Secure Agent machine:

| Component | Minimum requirement |
|---------------|---------------------|
| Cores per CPU | At least four |
| Memory | 16 GB |
| Disk Space | 100 GB |

After you install a Secure Agent, install OpenSSL on the Secure Agent machine.

For more information about installing a Secure Agent, see [Secure Agent installation](#).

Step 6. Allow domains in AWS

When the Secure Agent creates an advanced cluster in an AWS environment, the cluster nodes need access to certain domains to fetch artifacts, such as machine images, and to access sources and targets.

Add the following domains to the outbound allowlists for your security groups:

```
artifacthub.informaticacloud.com
infacloud.jfrog.io
infacloud-ct-cdie-docker.jfrog.io
infacloud-discale-docker-stable.jfrog.io

.s3.amazonaws.com
.s3.<staging bucket region>.amazonaws.com
awscli.amazonaws.com
```

Note: You need to install AWS CLI as part of the cluster creation.

If you use an Amazon S3 or Amazon Redshift object as a source or target, allow inbound traffic to each source and target bucket that the agent will access.

If you use GPU-enabled worker instances, also allow the following domains:

```
.docker.com
.docker.io
.nvidia.com
.nvidia.github.io
```

Also allow the appropriate region for AWS:

```
sts.amazonaws.com
```

To enable a regional endpoint connection, contact Informatica Global Customer Support to get the required custom property setting.

Note: If your organization does not use an outgoing proxy server, contact Informatica Global Customer Support to disable the proxy settings used for S3 access.

Step 7. Create IAM roles

Create the cluster operator, Secure Agent, master, and worker roles, and create the appropriate policies for each role to perform cluster operations in the AWS environment.

To create the IAM roles, complete the following tasks:

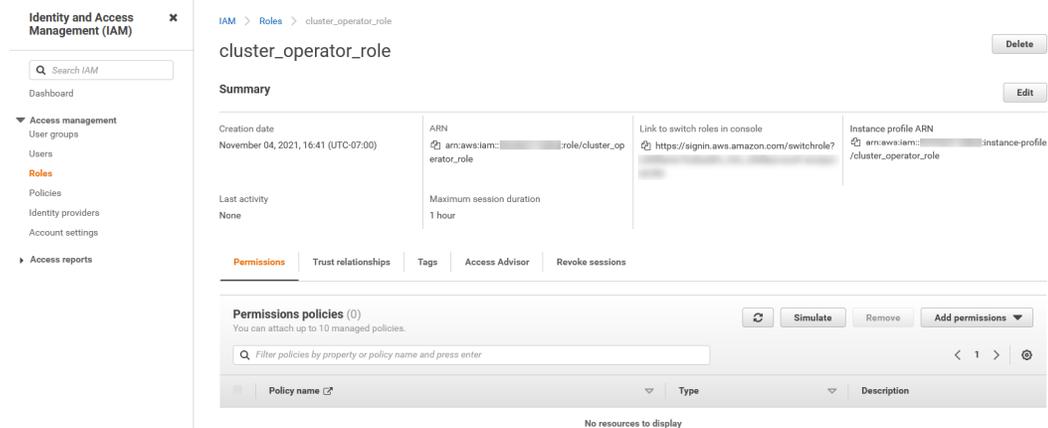
1. Create the cluster operator role.
2. Create the cluster operator policy.
3. Attach the cluster operator policy to the cluster operator role.
4. Configure the maximum CLI/API session duration for the cluster operator role.
5. Create or reuse the Secure Agent role.
6. Add the AssumeRole permission to the Secure Agent role.
7. Configure the trust relationship for the cluster operator role to include the Secure Agent role.
8. Create user-defined master and worker roles.
9. Optionally, encrypt staging data and log files at rest.
10. Optionally, create role-based security policies for Amazon data sources.
11. Create or reuse a cluster storage access policy for the Secure Agent role.

Note: To minimize the Secure Agent's permissions in your environment, avoid attaching the cluster operator role to the Secure Agent machine.

Create the cluster operator role

In AWS, create an IAM role for the cluster operator. Name the role `cluster_operator_role`.

The following image shows how the cluster operator role might appear in the AWS Management Console:



For instructions about creating an IAM role, refer to the AWS documentation. AWS provides several ways to create an IAM role, such as using the AWS Management Console or the AWS CLI.

Create the cluster operator policy

Create an IAM policy for the cluster operator role. Name the policy `cluster_operator_policy`. The cluster operator policy contains the permissions that the cluster operator role needs to create and manage cloud resources for an advanced cluster.

The following image shows how the cluster operator policy might appear in the AWS Management Console:

The screenshot shows the AWS IAM console interface for the `cluster_operator_policy`. The left sidebar contains navigation options like Dashboard, Access management, and Policies. The main content area shows the policy summary, including the Policy ARN and a description: "Minimum permissions required to create a Kubernetes cluster." Below this, there are tabs for Permissions, Policy usage, Tags, Policy versions, and Access Advisor. The Permissions tab is active, showing a table of permissions for 315 services. The table has columns for Service, Access level, Resource, and Request condition.

| Service | Access level | Resource | Request condition |
|------------------|---|---------------|-------------------|
| EC2 | Full: Tagging Limited: List, Read, Write | All resources | None |
| EC2 Auto Scaling | Full: Read Limited: List, Write | All resources | None |
| ELB | Full: List Limited: Read, Write, Tagging | All resources | None |
| ELB v2 | Limited: Read, Write, Tagging | All resources | None |
| IAM | Limited: List, Read, Write, Permissions management, Tagging | All resources | None |
| KMS | Limited: Read | All resources | None |
| Price List | Limited: Read | All resources | None |
| S3 | Limited: List, Read, Write, Tagging | Multiple | None |
| STS | Limited: Write | All resources | None |

You can use the following JSON document as a template for the policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "s3:GetLifecycleConfiguration",
        "s3:GetBucketTagging",
        "s3:GetBucketWebsite",
        "s3:GetBucketLogging",
        "s3:ListBucket",
        "s3:GetAccelerateConfiguration",
        "s3:GetBucketVersioning",
        "s3:GetReplicationConfiguration",
        "s3:PutObject",
        "s3:GetObjectAcl",
        "s3:GetObject",
        "s3:GetEncryptionConfiguration",
        "s3:PutBucketTagging",
        "s3:GetBucketRequestPayment",
        "s3:GetBucketCORS",
        "s3:GetObjectTagging",
        "s3:PutObjectTagging",
        "s3:GetBucketLocation",
        "s3:GetObjectVersion",
        "s3:DeleteObjectTagging",
        "s3:DeleteObjectVersion",
        "s3:DeleteObject"
      ],
      "Resource": [
        "arn:aws:s3:::staging_bucket",
        "arn:aws:s3:::staging_bucket/folder/",
        "arn:aws:s3:::staging_bucket/folder/*",
        "arn:aws:s3:::logging_bucket",
      ]
    }
  ]
}
```

```

        "arn:aws:s3:::logging_bucket/folder/",
        "arn:aws:s3:::logging_bucket/folder/*"
    ]
},
{
    "Sid": "VisualEditor1",
    "Effect": "Allow",
    "Action": [
        "ec2:DescribeInternetGateways",
        "ec2:AttachInternetGateway",
        "ec2:CreateInternetGateway",
        "ec2:DetachInternetGateway",
        "ec2>DeleteInternetGateway",
        "ec2:CreateKeyPair",
        "ec2:ImportKeyPair",
        "ec2:DescribeKeyPairs",
        "ec2>DeleteKeyPair",
        "ec2:CreateRoute",
        "ec2>DeleteRoute",
        "ec2:DescribeRouteTables",
        "ec2:CreateRouteTable",
        "ec2:ReplaceRouteTableAssociation",
        "ec2:AssociateRouteTable",
        "ec2:DisassociateRouteTable",
        "ec2>DeleteRouteTable",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeVpcs",
        "ec2:CreateVpc",
        "ec2>DeleteVpc",
        "ec2:ModifyVpcAttribute",
        "ec2:DescribeSubnets",
        "ec2:CreateSubnet",
        "ec2>DeleteSubnet",
        "ec2:DescribeSecurityGroups",
        "ec2:CreateSecurityGroup",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:RevokeSecurityGroupIngress",
        "ec2:AuthorizeSecurityGroupEgress",
        "ec2:RevokeSecurityGroupEgress",
        "ec2>DeleteSecurityGroup",
        "ec2:CreateTags",
        "ec2:DescribeTags",
        "ec2>DeleteTags",
        "ec2:CreateVolume",
        "ec2:DescribeVolumes",
        "ec2>DeleteVolume",
        "ec2:DescribeImages",
        "ec2:DescribeInstanceAttribute",
        "ec2:ModifyInstanceAttribute",
        "ec2:RunInstances",
        "ec2:DescribeInstances",
        "ec2:StartInstances",
        "ec2:StopInstances",
        "ec2:DescribeInstanceTypes",
        "ec2:TerminateInstances",
        "ec2:DescribeRegions",
        "ec2:DescribeAvailabilityZones",
        "ec2:CreateLaunchTemplate",
        "ec2:DescribeLaunchTemplateVersions",
        "ec2:DescribeLaunchTemplates",
        "ec2>DeleteLaunchTemplate",
        "ec2:CreateLaunchTemplateVersion",
        "ec2>DeleteLaunchTemplateVersions",
        "autoscaling:AttachLoadBalancers",
        "autoscaling:DescribeTags",
        "autoscaling:CreateAutoScalingGroup",
        "autoscaling:DescribeAutoScalingGroups",
        "autoscaling:DescribeScalingActivities",
        "autoscaling:UpdateAutoScalingGroup",
        "autoscaling>DeleteAutoScalingGroup",
        "autoscaling:TerminateInstanceInAutoScalingGroup",
    ]
}

```

```

        "elasticloadbalancing:AddTags",
        "elasticloadbalancing:DescribeTags",
        "elasticloadbalancing:ApplySecurityGroupsToLoadBalancer",
        "elasticloadbalancing:AttachLoadBalancerToSubnets",
        "elasticloadbalancing:ConfigureHealthCheck",
        "elasticloadbalancing>CreateLoadBalancer",
        "elasticloadbalancing:DescribeLoadBalancers",
        "elasticloadbalancing>DeleteLoadBalancer",
        "elasticloadbalancing>CreateLoadBalancerListeners",
        "elasticloadbalancing:DescribeInstanceHealth",
        "elasticloadbalancing:DescribeLoadBalancerAttributes",
        "elasticloadbalancing:ModifyLoadBalancerAttributes",
        "elasticloadbalancing:RegisterInstancesWithLoadBalancer",
        "pricing:GetProducts",
        "iam:GetInstanceProfile",
        "iam:GetContextKeysForPrincipalPolicy",
        "iam:ListInstanceProfiles",
        "iam:SimulatePrincipalPolicy",
        "iam>CreateInstanceProfile",
        "iam>DeleteInstanceProfile",
        "iam>CreateRole",
        "iam:GetRole",
        "iam:ListRoles",
        "iam:PassRole",
        "iam:ListRolePolicies",
        "iam>CreateServiceLinkedRole",
        "iam>DeleteRole",
        "iam:GetRolePolicy",
        "iam:AddRoleToInstanceProfile",
        "iam:ListAttachedRolePolicies",
        "iam:ListInstanceProfilesForRole",
        "iam:RemoveRoleFromInstanceProfile",
        "iam:PutRolePolicy",
        "iam:AttachRolePolicy",
        "iam:DetachRolePolicy",
        "iam>DeleteRolePolicy",
        "iam:GetUser",
        "kms:DescribeKey",
        "kms:Get*",
        "sts:AssumeRole",
        "sts:DecodeAuthorizationMessage"
    ],
    "Resource": "*"
}
]
}
}

```

Add permissions to the template based on your organizational requirements. For information about each permission, see [“IAM policy reference” on page 48](#).

The cluster operator role also requires the following permissions for public Informatica-managed Kubernetes clusters:

```

{
  "Sid": "VisualEditor0",
  "Effect": "Allow",
  "Action": [
    "ec2:GetLaunchTemplateData",
    "ec2:ModifyLaunchTemplate",
    "ec2:DescribeLaunchTemplates",
    "ec2:DescribeLaunchTemplateVersions"
  ],
  "Resource": "arn:aws:ec2:*:543463116864:launch-template/*.k8s.local"
}

```

The actions on Amazon S3 must be specified for all staging, log, and initialization script locations that you provide in advanced configurations.

For example, if you use staging location `dev/Staging/`, log location `dev/Logging/`, and initialization script location `dev/InitScript/`, the policy must list the following resources for actions on Amazon S3:

```
"Resource": [
  "arn:aws:s3:::dev",
  "arn:aws:s3:::dev/Staging/",
  "arn:aws:s3:::dev/Staging/*",
  "arn:aws:s3:::dev/Logging/",
  "arn:aws:s3:::dev/Logging/*",
  "arn:aws:s3:::dev/InitScript/",
  "arn:aws:s3:::dev/InitScript/*"
]
```

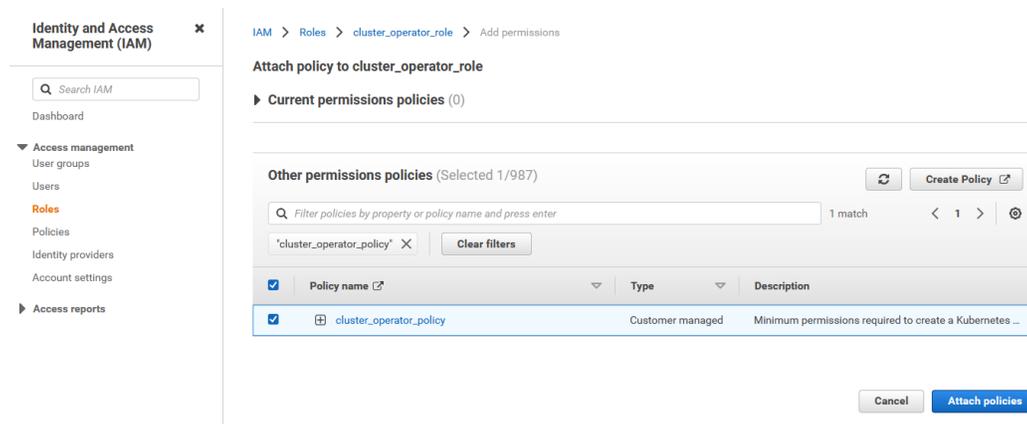
If you use a different set of staging, log, and initialization script locations in another advanced configuration, you must add those locations as resources to the same policy.

To accommodate S3 locations that change frequently, you can use wildcards. For more information, refer to the AWS documentation.

Attach the cluster operator policy

In AWS, attach the IAM policy `cluster_operator_policy` to the IAM role `cluster_operator_role`.

The following image shows how the AWS Management Console might appear when you attach the cluster operator policy to the cluster operator role:



Configure the maximum CLI/API session duration for the cluster operator role

In the IAM role `cluster_operator_role`, set the maximum CLI/API session duration to at least 30 minutes.

When you increase the duration, the Secure Agent has longer access to cloud resources within a single session, and you can run longer jobs on an advanced cluster.

For more information, refer to the AWS documentation.

Create or reuse the Secure Agent role

The Secure Agent requires an IAM role to access certain cloud resources while a job is running. This IAM role is attached to the Amazon EC2 instance where the Secure Agent is installed.

You can either create or reuse the Secure Agent role. Name this IAM role `agent_role`.

Create the Secure Agent role

To create the Secure Agent role, complete the following tasks in AWS:

1. Create an IAM role named `agent_role`.
2. Attach the IAM role `agent_role` to the Amazon EC2 instance where the Secure Agent is installed.

Reuse the Secure Agent role

If you already created an IAM role that is attached to the Amazon EC2 instance where the Secure Agent is installed, you can designate the IAM role to be the Secure Agent role.

Add the AssumeRole permission to the Secure Agent role

The Secure Agent needs to assume the cluster operator role to gain elevated permissions to manage an advanced cluster. For the Secure Agent to assume the cluster operator role, the Secure Agent role needs to have the AssumeRole permission.

To configure the AssumeRole permission, complete the following tasks in AWS:

1. Create the following IAM policy called `assume_role_agent_policy`:

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "sts:AssumeRole",
    "Resource": "arn:aws:iam::{{account-id}}:role/cluster_operator_role"
  }
}
```

Note: The value in the Resource element is the ARN of the cluster operator role.

2. Attach the IAM policy `assume_role_agent_policy` to the IAM role `agent_role`.

Configure the trust relationship for the cluster operator role to include the Secure Agent role

Because the Secure Agent needs to assume the cluster operator role, the cluster operator role needs to trust the Secure Agent.

Edit the trust relationship of the IAM role `cluster_operator_role` and specify the following policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::{{account-id}}:role/agent_role"
      },
      "Action": "sts:AssumeRole",
    }
  ]
}
```

Note: The value in the Principal element is the ARN of the Secure Agent role.

Optionally, you can configure an external ID to limit the entities that can assume the cluster operator role. Every time that the Secure Agent attempts to assume the cluster operator role, it must specify the external ID.

For example, you can configure the external ID "123" using the following policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::{{account-id}}:role/agent_role"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "sts:ExternalId": "123"
        }
      }
    }
  ]
}
```

Create user-defined master and worker roles

Create user-defined master and worker roles to fine-tune permissions for the master and worker nodes in an advanced cluster. The nodes use the permissions to run the Spark applications in an advanced job. After you complete these tasks, you can specify the master and worker instance profiles in an advanced configuration.

If you're looking for a quick setup, you can use default master and worker roles. For more information, see ["Use default master and worker roles \(alternative\)" on page 41](#) and ["Master and worker role types reference" on page 60](#).

To create user-defined roles, complete the following tasks:

1. Create the master and worker roles.
2. Create master policies.
3. Create worker policies.
4. Attach the policies to the master and worker roles.
5. Allow the cluster operator role to assume the worker role.
6. Allow the cluster operator role to assume the master role.

The master and worker roles, the instance profiles, and the cluster operator role must be defined under the same AWS account.

When the Secure Agent starts the advanced cluster, the agent uses the cluster operator role to validate whether the instance profiles exist and whether the master and worker roles have access to required cluster directories, such as staging, log, and initialization script locations. If validation fails, the cluster fails to be created.

Create the master and worker roles

In AWS, create IAM roles for the master and worker nodes. Name the roles `master_role` and `worker_role`, respectively.

When you create the master and worker roles, AWS automatically generates an instance profile for each role.

If the policy content provides access to staging, log, and initialization script locations for multiple advanced clusters, you can reuse the same instance profiles across different advanced configurations.

Create master policies

Create IAM policies for the master role. You can define each policy as an inline policy or a managed policy.

The following table describes each IAM policy:

| Policy | Description |
|---|--|
| <code>minimal_master_policy</code> | Required. Provides the minimal access permissions for the master role. |
| <code>staging_log_access_master_policy</code> | Required. Provides access to the staging and log locations. |
| <code>init_script_master_policy</code> | Required only if you use an initialization script. Provides access to the initialization script path and the location that stores init script and cloud-init logs. |

For information about each permission and why it's required, see ["IAM policy reference" on page 48](#). For information about editing the policies, see ["Master and worker policy restriction reference" on page 61](#).

Note: You can also generate the policy content by running the `generate-policies-for-userdefined-roles.sh` command. For more information about the command, see ["generate-policies-for-userdefined-roles.sh" on page 149](#). The command creates the output file `my-userdefined-master-worker-role-policies.json`.

minimal_master_policy

The IAM policy `minimal_master_policy` lists the minimal requirements for the user-defined master role.

You can use the following JSON document as a template for the `minimal_master_policy`:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstances",
        "ec2:DescribeRegions",
        "ec2:DescribeRouteTables",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeSubnets",
        "ec2:DescribeVolumes"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeVpcs",
        "ec2:CreateTags",
        "ec2:CreateVolume",
        "ec2:DescribeVolumesModifications",

```

```

        "ec2:ModifyInstanceAttribute",
        "ec2:ModifyVolume"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "ec2:AttachVolume",
        "ec2:DeleteVolume",
        "ec2:DetachVolume"
    ],
    "Resource": [
        "*"
    ],
    "Condition": {
        "StringLike": {
            "ec2:ResourceTag/KubernetesCluster": "*.k8s.local"
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "autoscaling:DescribeAutoScalingGroups",
        "autoscaling:DescribeLaunchConfigurations",
        "autoscaling:DescribeAutoScalingInstances",
        "autoscaling:DescribeTags",
        "autoscaling:DescribeScalingActivities"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "autoscaling:SetDesiredCapacity",
        "autoscaling:TerminateInstanceInAutoScalingGroup",
        "autoscaling:UpdateAutoScalingGroup"
    ],
    "Resource": [
        "*"
    ],
    "Condition": {
        "StringLike": {
            "autoscaling:ResourceTag/KubernetesCluster": "*.k8s.local"
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "elasticloadbalancing:AddTags",
        "elasticloadbalancing:AttachLoadBalancerToSubnets",
        "elasticloadbalancing:ApplySecurityGroupsToLoadBalancer",
        "elasticloadbalancing:ConfigureHealthCheck",
        "elasticloadbalancing>DeleteLoadBalancer",
        "elasticloadbalancing>DeleteLoadBalancerListeners",
        "elasticloadbalancing:DescribeLoadBalancers",
        "elasticloadbalancing:DescribeLoadBalancerAttributes",
        "elasticloadbalancing:DetachLoadBalancerFromSubnets",
        "elasticloadbalancing:DeregisterInstancesFromLoadBalancer",
        "elasticloadbalancing:ModifyLoadBalancerAttributes",
        "elasticloadbalancing:RegisterInstancesWithLoadBalancer",
        "elasticloadbalancing:SetLoadBalancerPoliciesForBackendServer"
    ],
    "Resource": [

```

```

        "*"
    ],
    "Condition": {
        "StringLike": {
            "elasticloadbalancing:ResourceTag/KubernetesCluster": "*.k8s.local"
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "elasticloadbalancing:AddTags",
        "elasticloadbalancing>DeleteListener",
        "elasticloadbalancing>DeleteTargetGroup",
        "elasticloadbalancing:DeregisterTargets",
        "elasticloadbalancing:DescribeListeners",
        "elasticloadbalancing:DescribeLoadBalancerPolicies",
        "elasticloadbalancing:DescribeTargetGroups",
        "elasticloadbalancing:DescribeTargetHealth",
        "elasticloadbalancing:ModifyListener",
        "elasticloadbalancing:ModifyTargetGroup",
        "elasticloadbalancing:RegisterTargets",
        "elasticloadbalancing:SetLoadBalancerPoliciesOfListener"
    ],
    "Resource": [
        "*"
    ],
    "Condition": {
        "StringLike": {
            "elasticloadbalancing:ResourceTag/KubernetesCluster": "*.k8s.local"
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "iam:ListServerCertificates",
        "iam:GetServerCertificate"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "s3:Get*"
    ],
    "Resource": [
        "arn:aws:s3:::<cluster-staging-dir1>/*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
    ],
    "Resource": [
        "*"
    ]
}
]
}

```

staging_log_access_master_policy

The IAM policy `staging_log_access_master_policy` provides access to the staging and log locations.

You can use the following JSON document as a template for the `staging_log_access_master_policy`:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketLocation",
        "s3:GetEncryptionConfiguration",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::<cluster-staging-bucket-name1>",
        "arn:aws:s3:::<cluster-logging-bucket-name1>"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObjectAcl",
        "s3:GetObject",
        "s3:DeleteObject",
        "s3:PutObjectAcl"
      ],
      "Resource": [
        "arn:aws:s3:::<cluster-staging-dir1>/*",
        "arn:aws:s3:::<cluster-logging-dir1>/*"
      ]
    }
  ]
}
```

[init_script_master_policy](#)

The IAM policy `init_script_master_policy` is required by the Cluster Computing System to allow the master node to access the initialization script and init script logging directories for the cluster.

You can use the following JSON document as a template for the `init_script_master_policy`:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketLocation",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::<cluster-init-script-bucket-name1>"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::<cluster-init-script-dir1>/*"
      ]
    }
  ]
}
```

Create worker policies

Create IAM policies for the worker role. You can define each policy as an inline policy or a managed policy.

The following table describes each IAM policy:

| Policy | Description |
|---|--|
| <code>minimal_worker_policy</code> | Required. Provides the minimal access permissions for the worker role. |
| <code>ebs_autoscaling_worker_policy</code> | Required only if EBS volumes auto-scale. |
| <code>staging_log_access_worker_policy</code> | Required. Provides access to the staging and log locations. |
| <code>init_script_worker_policy</code> | Required only if you use an initialization script. Provides access to the initialization script path and the location that stores init script and cloud-init logs. |

For information about each permission and why it's required, see [“IAM policy reference” on page 48](#). For information about editing the policies, see [“Master and worker policy restriction reference” on page 61](#).

Note: You can also generate the policy content by running the `generate-policies-for-userdefined-roles.sh` command. For more information about the command, see [“generate-policies-for-userdefined-roles.sh” on page 149](#). The command creates the output file `my-userdefined-master-worker-role-policies.json`.

[minimal_worker_policy](#)

The IAM policy `minimal_worker_policy` lists the minimal requirements for the user-defined worker role.

You can use the following JSON document as a template for the `minimal_worker_policy`:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstances",
        "ec2:DescribeRegions"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTags"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:volume/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "autoscaling:DescribeAutoScalingInstances",
        "autoscaling:DescribeTags"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:Get*"
      ],
    }
  ]
}
```

```

    "Resource": [
      "arn:aws:s3:::<cluster-staging-dir1>/*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "kms:Encrypt",
      "kms:Decrypt",
      "kms:ReEncrypt*",
      "kms:GenerateDataKey*",
      "kms:DescribeKey"
    ],
    "Resource": [
      "*"
    ]
  }
]
}

```

ebs_autoscaling_worker_policy

The IAM policy `ebs_autoscaling_worker_policy` is required by the worker nodes to auto-scale EBS volumes.

You can use the following JSON document as a template for the `ebs_autoscaling_worker_policy`:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ec2:DescribeVolumes",
        "ec2:CreateVolume",
        "ec2:ModifyInstanceAttribute"
      ],
      "Effect": "Allow",
      "Resource": [
        "*"
      ]
    },
    {
      "Action": [
        "ec2:CreateTags"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:ec2:*:*:volume/*"
      ]
    },
    {
      "Action": [
        "ec2:AttachVolume",
        "ec2:DetachVolume"
      ],
      "Condition": {
        "StringLike": {
          "ec2:ResourceTag/KubernetesCluster": "*.k8s.local"
        }
      },
      "Effect": "Allow",
      "Resource": [
        "arn:aws:ec2:*:*:instance/*"
      ]
    },
    {
      "Action": [
        "ec2:AttachVolume",
        "ec2:DetachVolume",
        "ec2>DeleteVolume"
      ],

```

```

        "Condition": {
            "StringLike": {
                "ec2:ResourceTag/CREATED_BY": "infa-storage-scalerd-*"
            }
        },
        "Effect": "Allow",
        "Resource": [
            "arn:aws:ec2:*:*:volume/*"
        ]
    }
]
}

```

staging_log_access_worker_policy

The IAM policy `staging_log_access_worker_policy` is required by the Cluster Computing System to permit worker nodes to access staging and logging directories.

You can use the following JSON document as a template for the `staging_log_access_worker_policy`:

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation",
                "s3:GetEncryptionConfiguration",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::<cluster-staging-bucket-name1>",
                "arn:aws:s3:::<cluster-logging-bucket-name1>"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObjectAcl",
                "s3:GetObject",
                "s3:DeleteObject",
                "s3:PutObjectAcl"
            ],
            "Resource": [
                "arn:aws:s3:::<cluster-staging-dir1>/*",
                "arn:aws:s3:::<cluster-logging-dir1>/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:DescribeKey"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}

```

init_script_worker_policy

The IAM policy `staging_log_access_worker_policy` is required by the Cluster Computing System to allow worker nodes to access the initialization script and init script logging directories.

You can use the following JSON document as a template for the `init_script_worker_policy`:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketLocation",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::<cluster-init-script-bucket-name>"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::<cluster-init-script-dir1>/*"
      ]
    }
  ]
}
```

Attach the policies to the master and worker roles

Attach each IAM policy to the appropriate IAM role: either `master_role` or `worker_role`.

The following table lists the policies to attach to each role:

| Role | Policies |
|--------------------------|--|
| <code>master_role</code> | <ul style="list-style-type: none">- <code>minimal_master_policy</code>- <code>staging_log_access_master_policy</code>- <code>init_script_master_policy</code> |
| <code>worker_role</code> | <ul style="list-style-type: none">- <code>minimal_worker_policy</code>- <code>ebs_autoscaling_worker_policy</code>- <code>staging_log_access_worker_policy</code>- <code>init_script_worker_policy</code> |

Allow the cluster operator role to assume the worker role

The cluster operator role must be able to assume the worker role to validate an advanced configuration.

Edit the trust relationship of the IAM role `worker_role` and specify the following policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::<AWS account>:role/<cluster_operator_role>"
        ]
      },
      "Service": "ec2.amazonaws.com"
    },
    {
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Allow the cluster operator role to assume the master role

The cluster operator role must be able to assume the master role to validate an advanced configuration.

Edit the trust relationship of the IAM role `master_role` and specify the following policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam:<AWS account>:role/<cluster_operator_role>"
        ],
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Use default master and worker roles (alternative)

For a quick setup, you can use default master and worker roles. In this case, the Secure Agent automatically creates the roles when the agent starts an advanced cluster.

The agent attaches policies to the roles based on the permissions that are required by Kubernetes services. If you use role-based security and jobs have direct access to Amazon data sources, the agent also identifies the policies that are attached to the Secure Agent role and passes the policies to the worker role.

To use default roles, add the following policy to the IAM role `cluster_operator_role`:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:AddRoleToInstanceProfile",
        "iam:CreateInstanceProfile",
        "iam:CreateRole",
        "iam>DeleteInstanceProfile",
        "iam>DeleteRole",
        "iam>DeleteRolePolicy",
        "iam:GetInstanceProfile",
        "iam:GetRole",
        "iam:GetRolePolicy",
        "iam:GetUser",
        "iam>ListAttachedRolePolicies",
        "iam>ListInstanceProfiles",
        "iam>ListInstanceProfilesForRole",
        "iam>ListRolePolicies",
        "iam>ListRoles",
        "iam:PassRole",
        "iam:PutRolePolicy",
        "iam:RemoveRoleFromInstanceProfile",
        "iam:AttachRolePolicy",
        "iam:DetachRolePolicy",
        "iam:CreateServiceLinkedRole"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

Encrypt staging data and log files at rest (optional)

Optionally, set up Amazon S3 default encryption for S3 buckets to automatically encrypt staging data and log files that are stored on Amazon S3.

You can set up Amazon S3 default encryption for S3 buckets using one of the following encryption options:

Server-Side Encryption with Amazon S3-Managed Keys (SSE-S3)

Use SSE-S3 to encrypt individual staging and log files or to encrypt the S3 buckets that contain the staging and log locations.

Server-Side Encryption with AWS KMS-Managed Keys (SSE-KMS)

Use SSE-KMS to encrypt individual staging and log files. If you create user-defined master and worker roles, you can also encrypt the S3 buckets that contain the staging and log locations.

For more information about the encryption options, refer to the AWS documentation.

If you use SSE-KMS and create user-defined master and worker roles, you can restrict the customer master key (CMK) IDs that the master and worker roles can access to encrypt and decrypt data.

Specify the key IDs in the policies that are attached to the master and worker roles. In each policy, edit the Resource element in the following statement that determines actions on AWS Key Management Service (KMS):

```
{
  "Effect": "Allow",
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",
    "kms:DescribeKey"
  ],
  "Resource": [
    "*"
  ]
}
```

Note: If you use SSE-KMS, you must use the default AWS-managed CMK on your Amazon account. You cannot create a custom CMK.

Create role-based security policies for Amazon data sources (optional)

Role-based security uses IAM roles to access data sources. If a connector directly accesses AWS, such as Amazon S3 V2 Connector or Amazon Redshift V2 Connector, create policies to allow the Secure Agent and worker roles to have access to data sources and fine-tune their permissions in your AWS environment.

You can skip this step if you use connectors that don't have direct access to AWS. For example, JDBC V2 Connector uses a driver to query data on Amazon Aurora and does not directly access the underlying data.

If you're looking for a quick setup, you can use credential-based security. For more information, see ["Use credential-based security \(alternative\)" on page 44](#).

Complete the following tasks:

1. Create policies for the Secure Agent and worker roles.
2. Optionally, configure cross-account access.

By default, the agent and worker roles access data sources, but you can specify an IAM role at the connection level to access the data sources instead of using the agent and worker roles.

If you use default master and worker roles, consider the following guidelines:

- If you edit the Secure Agent role, you must restart the agent to update the master and worker roles.
- The default worker role doesn't honor the permission boundaries for the Secure Agent role.
- The staging location, log location, and cluster operator role must be in the same AWS account.

Step 10.1. Create policies for the Secure Agent and worker roles

Create policies to allow the Secure Agent and worker roles to access Amazon data sources in an advanced job. Create and distribute the policies based on the worker role type.

User-defined worker role

If you create a user-defined worker role, you can provide access to the data sources in one of the following ways:

Create a new managed policy

To create a new managed policy, complete the following tasks:

1. Create the policy that the connector requires. Name the policy `data_source_access_policy`. For information about connector requirements, see the help for the appropriate connector.
2. Attach the policy `data_source_access_policy` to both the Secure Agent role and worker role.

Reuse the IAM policy `staging_log_access_worker_policy`

To reuse the IAM policy `staging_log_access_worker_policy` that is attached to the worker role, complete the following tasks:

1. Specify the data sources in the Resource elements.

For example, the Resource element in the following statement specifies the staging and log locations:

```
{
  "Effect": "Allow",
  "Action": [
    "s3:PutObject",
    "s3:GetObjectAcl",
    "s3:GetObject",
    "s3:DeleteObject",
    "s3:PutObjectAcl"
  ],
  "Resource": [
    "arn:aws:s3:::<cluster-staging-dir1>/*",
    "arn:aws:s3:::<cluster-logging-dir1>/*"
  ]
}
```

Below `"arn:aws:s3:::<cluster-logging-dir1>/*"`, add the data sources.

2. Add the Secure Agent role to the trust relationship of the worker role.
3. Add the worker role to the trust relationship of the Secure Agent role.

Default worker role

If you use the default worker role, complete the following tasks:

1. Create the policy that the connector requires. Name the policy `data_source_access_policy`. For information about connector requirements, see the help for the appropriate connector.
2. Attach the policy `data_source_access_policy` to the Secure Agent role. The Secure Agent will automatically pass the policy to the worker role.

Step 10.2. Configure cross-account access (optional)

If you require cross-account access to S3 buckets in multiple Amazon accounts and you use user-defined master and worker roles, set up cross-account IAM roles in AWS.

When you set up cross-account IAM roles in AWS, complete the following tasks:

- Edit the policies in the user-defined worker role to access the S3 resources in each account.
- Add a bucket policy to the S3 buckets in each account that permits the user-defined worker role to access the bucket.

Note: You cannot combine cross-account access with default master and worker roles and role-based security. If your organization requires cross-account access, consider one of the following options:

- Create user-defined master and worker roles. For more information, see [“Create user-defined master and worker roles” on page 32](#).
- Use credential-based security. For more information, see [“Use credential-based security \(alternative\)” on page 44](#).

For information about how to set up cross-account IAM roles, refer to the AWS documentation.

Use credential-based security (alternative)

For a quick setup, you can reuse the AWS credentials that you configure in a data source's connection properties instead of configuring IAM roles. Cluster nodes use the connection-level credentials to access the staging and log locations only when the same S3 bucket stores the data sources, staging files, and log files.

For example, if a job uses a JDBC V2 source and an Amazon S3 V2 target, cluster nodes use the Amazon S3 V2 credentials to access the staging location for the job.

Note: The AWS credentials in the connection must be able to access the Amazon S3 staging location that the job uses, and credentials override IAM roles. If you configure AWS credentials for a connector and the same credentials cannot access both the data sources and the staging location in an advanced job, the job fails.

If you require cross-account access to S3 buckets in multiple Amazon accounts, provide credentials for each Amazon account at the connection level.

Create or reuse a log access policy for the Secure Agent role

The Secure Agent needs permissions to access the log location to upload the agent job log at the end of an advanced job.

You can either create or reuse an IAM policy for log access.

Create a log access policy

To create an IAM policy for log access, complete the following tasks in AWS:

1. Create the following IAM policy named `log_access_agent_policy`:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketLocation",
        "s3:GetEncryptionConfiguration",
        "s3:ListBucket"
      ]
    }
  ]
}
```

```

        "Resource": [
            "arn:aws:s3:::<cluster-logging-bucket-name>"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "s3:PutObject",
            "s3:GetObjectAcl",
            "s3:GetObject",
            "s3:DeleteObject",
            "s3:PutObjectAcl"
        ],
        "Resource": [
            "arn:aws:s3:::<cluster-logging-dir1>/*"
        ]
    }
]
}

```

Specify the log location in the Resource elements.

2. Attach the IAM policy `log_access_agent_policy` to the IAM role `agent_role`.

Reuse a log access policy

If you create user-defined master and worker roles, you can reuse the policy content that is generated for the CCS and required for the worker role.

The policy content includes access to the log location that the Secure Agent needs. For more information about user-defined master and worker roles, see [“Create user-defined master and worker roles” on page 32](#).

To reuse the policy, complete the following tasks:

1. Edit the trust relationship of the worker role and specify the following policy to trust the IAM role `agent_role`:

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::{{account-id}}:role/<agent_role>"
                ],
                "Service": "ec2.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}

```

2. Edit the trust relationship of the IAM role `agent_role` and specify the following policy to trust the worker role:

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::{{account-id}}:role/<worker role>"
                ],
                "Service": "ec2.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}

```

```

    ]
}

```

Step 8. Configure environment variables (optional)

To run commands such as `list-clusters.sh` and `delete-clusters.sh`, configure environment variables on the Secure Agent machine.

The following table describes each environment variable:

| Environment Variable | Description |
|------------------------|---|
| JAVA_HOME | Java version on the Secure Agent machine that is used to run commands. The Java version on the Secure Agent machine must be compatible with JDK 8. |
| PRIVILEGED_ROLE_ARN | ARN of the IAM role <code>cluster_operator_role</code> . Used by the <code>list-clusters.sh</code> and <code>delete-clusters.sh</code> commands. |
| AGENT_ROLE_EXTERNAL_ID | External ID that the Secure Agent uses to assume the IAM role <code>cluster_operator_role</code> . Used by the <code>list-clusters.sh</code> and <code>delete-clusters.sh</code> commands. |

Step 9. Configure the Elastic Server

In Administrator, configure the service properties for the Elastic Server.

The following image shows the Elastic Server properties:

| Type | Name | Value | Sensitive |
|-----------------|-------------------------------------|---|--------------------------|
| PARAMFILE_CFG | parameterfile_access_flag | 'true' | <input type="checkbox"/> |
| PARAMFILE_CFG | parameterfile_access_directory | '/\$AGENT_HOME/apps/data/userparameters,/\$AGENT_HOME/apps/Data_Integration_Server/data/userparameters' | <input type="checkbox"/> |
| LOG4J_CFG | log4l_app_log_level | 'INFO' | <input type="checkbox"/> |
| AWS_CFG | agent_role_external_id_key | | <input type="checkbox"/> |
| AWS_CFG | privileged_role_arn_key | arn:aws:iam::<account id>:role/cluster_operator_role | <input type="checkbox"/> |
| AWS_CFG | role_session_duration_secs_key | | <input type="checkbox"/> |
| AWS_CFG | aws_regional_endpoint_enabled | 'false' | <input type="checkbox"/> |
| AZURE_CFG | azure_agent_role_identity_client_id | | <input type="checkbox"/> |
| CONCURRENCY_CFG | allow_queuing | 'true' | <input type="checkbox"/> |
| CONCURRENCY_CFG | max_concurrent_jobs | | <input type="checkbox"/> |

You can configure the following Elastic Server properties:

| Type | Name | Description |
|---------------|--------------------------------|---|
| PARAMFILE_CFG | parameterfile_access_flag | Indicates whether developers can download parameter files that are stored on the Secure Agent machine. Default is 'true.' |
| PARAMFILE_CFG | parameterfile_access_directory | List of directories on the Secure Agent machine that allow parameter file download. Developers can download parameter files from any of the specified directories or subdirectories. Default is '\$AGENT_HOME/apps/data/userparameters, /\$AGENT_HOME/apps/Data_Integration_Server/data/userparameters.' |
| LOG4J_CFG | log4j_app_log_level | Level of detail that the Elastic Server writes to log files. Enter the logging level as a string, such as 'INFO.' As the logging level increases, the messages that the Elastic Server writes to log files include the messages in the prior logging levels. For example, if the logging level is INFO, the log contains FATAL, ERROR, WARNING, and INFO code messages. The following values are valid: <ol style="list-style-type: none"> 1. FATAL. Includes nonrecoverable system failures that cause the service to shut down or become unavailable. 2. ERROR. Includes connection failures, failures to save or retrieve metadata, and service errors. 3. WARNING. Includes recoverable system failures or warnings. 4. INFO. Includes system and service change messages. 5. TRACE. Logs user request failures. 6. DEBUG. Logs user request logs. |
| AWS_CFG | agent_role_external_id_key | External ID that the Secure Agent specifies when the agent attempts to assume the cluster operator role. Required if you configure an external ID in the trust relationship of the cluster operator role. This property takes effect only in an AWS environment. |
| AWS_CFG | privileged_role_arn_key | ARN of the cluster operator role. Required when you set up separate cluster operator and Secure Agent roles in an AWS environment. This property takes effect only in an AWS environment. |
| AWS_CFG | role_session_duration_secs_key | Session duration of the AWS AssumeRole API in seconds. By default, the session duration is 1800 seconds (30 minutes). Overrides the maximum CLI/API session duration that is configured for the cluster operator role. If the session duration configured for the Elastic Server is longer than session duration for the cluster operator role, the Secure Agent might fail to assume the cluster operator role. This property takes effect only in an AWS environment. |

| Type | Name | Description |
|-----------------|-------------------------------------|--|
| AZURE_CFG | azure_agent_role_identity_client_id | Client ID of the managed identity <code>agent_identity</code> . Required when <code>agent_identity</code> is a user-assigned managed identity and the Secure Agent machine has at least one other managed identity. This property takes effect only in an Azure environment. |
| CONCURRENCY_CFG | allow_queuing | Indicates whether the Elastic Server queues Spark tasks. Default is true. |
| CONCURRENCY_CFG | max_concurrent_jobs | Maximum number of concurrent Spark tasks that the Elastic Server can process. |

For more information about Secure Agent services, see *Secure Agent Services*.

IAM policy reference

The cluster operator role, the master role, and the worker role require IAM policies to create and manage cloud resources in an advanced cluster. This section describes the actions that each role requires in the IAM policies.

Cluster operator role actions

Add actions to the IAM policy for the cluster operator role to allow the role to create and manage cloud resources.

The cluster operator role requires actions defined by the following services on AWS:

- Amazon EC2
- Amazon S3
- AWS Auto Scaling
- AWS Key Management Service
- AWS Security Token Service
- Elastic Load Balancing
- Identity and Access Management
- Pricing

Amazon EC2 actions

Amazon Elastic Compute Cloud (EC2) provides computing resources on the cloud. Amazon EC2 actions must apply to all AWS resources.

Internet gateway

The following table describes the actions for internet gateways:

| Action | Description |
|-----------------------------|---|
| ec2:CreateInternetGateway | Required only when the Secure Agent creates a VPC and subnets for the cluster. The Secure Agent creates a VPC and subnets by default. |
| ec2:AttachInternetGateway | Required only when the Secure Agent creates a VPC and subnets for the cluster. The Secure Agent creates a VPC and subnets by default. |
| ec2:DescribeInternetGateway | Required. Describes the internet gateway. |
| ec2:DetachInternetGateway | Required only when the Secure Agent creates a VPC and subnets for the cluster. The Secure Agent creates a VPC and subnets by default. |
| ec2>DeleteInternetGateway | Required only when the Secure Agent creates a VPC and subnets for the cluster. The Secure Agent creates a VPC and subnets by default. |

Key pair

The cluster operator creates AWS EC2 key pairs, which allows end users to connect to EC2 instances. The cluster operator role requires the following actions to manage key pairs:

```
ec2:CreateKeyPair
ec2:ImportKeyPair
ec2:DescribeKeyPair
ec2>DeleteKeyPair
```

Network

The cluster operator role requires the `ec2:DescribeNetworkInterfaces` action to describe network interfaces.

Route

The cluster operator role requires the following actions only when the Secure Agent creates a VPC and subnets for the cluster:

```
ec2:CreateRoute
ec2>DeleteRoute
```

The Secure Agent creates a VPC and subnets by default.

Route table

The following table describes the actions for route tables:

| Action | Description |
|-------------------------|---|
| ec2:CreateRouteTable | Required only when the Secure Agent creates a VPC and subnets for the cluster. The Secure Agent creates a VPC and subnets by default. |
| ec2:DescribeRouteTables | Required. Returns route table details. |

| Action | Description |
|----------------------------------|---|
| ec2:ReplaceRouteTableAssociation | Required only when the Secure Agent creates a VPC and subnets for the cluster. The Secure Agent creates a VPC and subnets by default. |
| ec2:AssociateRouteTable | Required only when the Secure Agent creates a VPC and subnets for the cluster. The Secure Agent creates a VPC and subnets by default. |
| ec2:DisassociateRouteTable | Required only when the Secure Agent creates a VPC and subnets for the cluster. The Secure Agent creates a VPC and subnets by default. |
| ec2>DeleteRouteTable | Required only when the Secure Agent creates a VPC and subnets for the cluster. The Secure Agent creates a VPC and subnets by default. |

VPC

The following table describes the actions for VPCs:

| Action | Description |
|------------------------|---|
| ec2:CreateVpc | Required only when the Secure Agent creates a VPC and subnets for the cluster. The Secure Agent creates a VPC and subnets by default. |
| ec2:DescribeVpcs | Required. Describes VPC details. |
| ec2:ModifyVpcAttribute | Required only when the Secure Agent creates a VPC and subnets for the cluster. The Secure Agent creates a VPC and subnets by default. |
| ec2>DeleteVpc | Required only when the Secure Agent creates a VPC and subnets for the cluster. The Secure Agent creates a VPC and subnets by default. |

Subnet

The following table describes the actions for subnets:

| Action | Description |
|--------------------|---|
| ec2:CreateSubnet | Required only when the Secure Agent creates a VPC and subnets for the cluster. The Secure Agent creates a VPC and subnets by default. |
| ec2:DescribeSubnet | Required. Describe subnet details. |
| ec2>DeleteSubnet | Required only when the Secure Agent creates a VPC and subnets for the cluster. The Secure Agent creates a VPC and subnets by default. |

Security group

The following table describes the actions for security groups:

| Action | Description |
|-----------------------------------|--|
| ec2:CreateSecurityGroup | Optional. Required only if you want to create and use user-defined Amazon EC2 security groups. |
| ec2:DescribeSecurityGroups | Required. Describes security group details. |
| ec2:AuthorizeSecurityGroupEgress | Optional. Required only if you want to create and use user-defined Amazon EC2 security groups. |
| ec2:AuthorizeSecurityGroupIngress | Optional. Required only if you want to create and use user-defined Amazon EC2 security groups. |
| ec2:RevokeSecurityGroupEgress | Optional. Required only if you want to create and use user-defined Amazon EC2 security groups. |
| ec2:RevokeSecurityGroupIngress | Optional. Required only if you want to create and use user-defined Amazon EC2 security groups. |
| ec2>DeleteSecurityGroup | Optional. Required only if you want to create and use user-defined Amazon EC2 security groups. |

For more information about user-defined security groups, see [“Step 4. Create user-defined security groups for Amazon EC2” on page 22](#).

Tags

The following table describes the actions for tags:

| Action | Description |
|------------------|---|
| ec2:CreateTags | Required. Adds tags for Kubernetes infrastructure, such as Amazon EC2. Kubernetes identifies resources through tags. Tags allow you to manage resources and add conditional statements. |
| ec2:DescribeTags | Required. Describes tags for Kubernetes infrastructure, such as Amazon EC2. |
| ec2>DeleteTags | Required. Deletes tags for Kubernetes infrastructure, such as Amazon EC2. |

Volumes

The cluster operator manages etcd volumes directly. An advanced cluster uses etcd volumes to store metadata. The cluster operator role requires the following actions to manage etcd volumes:

```
ec2:CreateVolumes
ec2:DescribeVolumes
ec2>DeleteVolumes
```

Image

The cluster operator role requires the `ec2:DescribeImages` action to get the AMI (Amazon Machine Image) details from the Amazon EC2 instance.

Instances

The following table describes the actions for instances:

| Action | Description |
|---|--|
| ec2:DescribeInstanceAttribute | Required. Gets details of the created Amazon EC2 instances. |
| ec2:ModifyInstanceAttribute | Required. Allows the cluster operator to manage and create Amazon EC2 instances. |
| ec2:RunInstances | Required. Allows the cluster operator to manage and create Amazon EC2 instances. |
| ec2:DescribeInstances ec2:DescribeInstanceType | Required. Gets details of the created Amazon EC2 instances. |
| ec2:TerminateInstances | Required. Terminates EC2 instances created by the cluster operator role. |

Region

The following table describes the actions for regions:

| Action | Description |
|-------------------------------|--|
| ec2:DescribeRegions | Required. Describes the region you selected in the advanced configuration. |
| ec2:DescribeAvailabilityZones | Required. Describes details of availability zones. |

Launch template

The cluster operator uses a launch template to launch EC2 instances. The cluster operator role requires the following actions to manage launch templates:

```
ec2:CreateLaunchTemplate
ec2:DescribeLaunchTemplates
ec2>DeleteLaunchTemplate
ec2:CreateLaunchTemplateVersion
ec2:DescribeLaunchTemplateVersions
ec2>DeleteLaunchTemplateVersions
ec2:GetLaunchTemplateData
ec2:ModifyLaunchTemplate
```

Amazon S3 actions

The following table lists the Amazon S3 actions that the cluster operator role requires and the resources that each action must apply to:

| Action | Resource |
|-------------------------------|--|
| s3:GetBucketLocation | "arn:aws:s3:::<cluster-staging-bucket-name>" "arn:aws:s3:::<cluster-logging-bucket-name>" |
| s3:GetEncryptionConfiguration | "arn:aws:s3:::<cluster-staging-bucket-name>" "arn:aws:s3:::<cluster-logging-bucket-name>" |

| Action | Resource |
|-----------------|--|
| s3:ListBucket | "arn:aws:s3:::<cluster-staging-bucket-name>" "arn:aws:s3:::<cluster-logging-bucket-name>" |
| s3:PutObject | "arn:aws:s3:::<cluster-staging-dir>/*" "arn:aws:s3:::<cluster-logging-dir>/*" |
| s3:GetObjectAcl | "arn:aws:s3:::<cluster-staging-dir>/*" "arn:aws:s3:::<cluster-logging-dir>/*" |
| s3:GetObject | "arn:aws:s3:::<cluster-staging-dir>/*" "arn:aws:s3:::<cluster-logging-dir>/*" |
| s3>DeleteObject | "arn:aws:s3:::<cluster-staging-dir>/*" "arn:aws:s3:::<cluster-logging-dir>/*" |
| s3:PutObjectAcl | "arn:aws:s3:::<cluster-staging-dir>/*" "arn:aws:s3:::<cluster-logging-dir>/*" |

AWS Auto Scaling actions

The cluster operator uses an Auto Scaling group to manage advanced clusters.

The cluster operator role requires the following actions on all AWS resources for scalable cluster nodes and node recovery:

```

autoscaling:AttachLoadBalancers
autoscaling:CreateAutoScalingGroup
autoscaling:DescribeAutoScalingGroups
autoscaling:UpdateAutoScalingGroup
autoscaling>DeleteAutoScalingGroup
autoscaling:DescribeScalingActivities
autoscaling:DescribeTags
autoscaling:TerminateInstanceInAutoScalingGroup

```

AWS Key Management Service actions

The cluster operator role requires the `kms:DescribeKey` action when root volume encryption is enabled and the customer-managed key (CMK) is provided for the cluster operator role. This action applies to all AWS resources.

AWS Security Token Service actions

The following table describes the STS actions:

| Action | Description |
|--------------------------------|--|
| sts:AssumeRole | Required when you use the user-defined master role and worker role. |
| sts:DecodeAuthorizationMessage | Optional. Used to decode the encrypted message received from the AWS response. |

Elastic Load Balancing actions

The cluster operator requires a load balancer for high availability, master node access control, and other features.

The cluster operator role requires the following Elastic Load Balancing actions on all AWS resources:

```
elasticloadbalancing:AddTags
elasticloadbalancing:DescribeTags
elasticloadbalancing:ApplySecurityGroupsToLoadBalancer
elasticloadbalancing:AttachLoadBalancerToSubnets
elasticloadbalancing:ConfigureHealthCheck
elasticloadbalancing:CreateLoadBalancer
elasticloadbalancing:DescribeLoadBalancers
elasticloadbalancing>DeleteLoadBalancer
elasticloadbalancing:CreateLoadBalancerListeners
elasticloadbalancing:DescribeInstanceHealth
elasticloadbalancing:DescribeLoadBalancerAttributes
elasticloadbalancing:ModifyLoadBalancerAttributes
elasticloadbalancing:RegisterInstancesWithLoadBalancer
```

Identity and Access Management actions

The Identity and Access Management actions apply to all AWS resources.

Instance profiles

The following table describes the actions for instance profiles:

| Action | Description |
|---|---|
| iam:AddRoleToInstanceProfile | Optional if you do not specify master and worker instance profiles. |
| iam:CreateInstanceProfile | Optional when you provide master and worker roles. |
| iam:DeleteInstanceProfile | Optional when you provide master and worker roles. |
| iam:GetContextKeysForPrincipalPolicy iam:SimulatePrincipalPolicy | Required. Allows permission validation, including advanced configuration validation and upgrade validation. |
| iam:GetInstanceProfile | Required. Retrieves information about the specified instance profile, including the instance profile path, GUID, ARN, and role. |
| iam:ListInstanceProfiles | Required. Lists the instance profiles that have the specified path prefix. |

Roles

The following table describes the actions for IAM roles:

| Action | Description |
|-----------------------------|--|
| iam:CreateRole | Optional when you provide master and worker roles. |
| iam:CreateServiceLinkedRole | Required. Creates an IAM role that is linked to a specific AWS service. |
| iam>DeleteRole | Optional when you provide master and worker roles. |
| iam:GetRole | Required. Retrieves information about the specified role, including the role path. |

| Action | Description |
|----------------------|--|
| iam:ListRolePolicies | Required. Retrieves information about the specified role, including the role path. |
| iam:ListRoles | Required. Retrieves information about the specified role, including the role path. |

Policies

The following table describes the actions for IAM policies:

| Action | Description |
|---|---|
| iam:AttachRolePolicy iam>DeleteRolePolicy iam:DetachRolePolicy iam:PutRolePolicy | Optional when you provide master and worker roles. |
| iam:GetRolePolicy | Required. Retrieves the specified inline policy document that AWS embeds with the specified IAM role. |
| iam>ListAttachedRolePolicies | Required. Lists all managed policies that are attached to the specified IAM role. |
| iam>ListInstanceProfilesForRole | Required. Lists the instance profiles that have the associated IAM role. |
| iam:RemoveRoleFromInstanceProfile | Required. Removes the specified IAM role from the specified EC2 instance profile. |

Users

The cluster operator role requires the `iam:GetUser` action to retrieve information about the specified IAM user, including the path, unique ID, and ARN.

Pricing actions

The cluster operator role requires the `pricing:GetProducts` action when you use Spot Instances.

Master role actions

Add actions to the IAM policy for the master role to allow the role to access and manage cloud resources.

The master role requires actions defined by the following services on AWS:

- Amazon EC2
- Amazon S3
- AWS Auto Scaling
- AWS Key Management Service
- Elastic Load Balancing
- Identity and Access Management

Amazon EC2 actions

Amazon Elastic Compute Cloud (EC2) provides computing resources on the cloud. Amazon EC2 actions must apply to all AWS resources.

The following table describes the actions that the master role requires:

| Action | Description |
|---|---|
| ec2:DescribeInstances | Required. Allows Kubernetes to describe instances. |
| ec2:DescribeRegions | Required. Allows Kubernetes to describe regions. |
| ec2:CreateRoute | Optional. Required only when the Secure Agent creates a VPC and subnets for the cluster. The Secure Agent creates a VPC and subnets by default. |
| ec2:DescribeRouteTables | Required. Sets up Kubernetes infrastructure. |
| ec2>DeleteRoute | Optional. Required only when the Secure Agent creates a VPC and subnets for the cluster. The Secure Agent creates a VPC and subnets by default. |
| ec2:CreateSecurityGroup | Optional. Required only when you use the default security groups that the cluster operator role creates. |
| ec2:CreateSecurityGroup ec2:AuthorizeSecurityGroupIngress ec2:RevokeSecurityGroupIngress ec2>DeleteSecurityGroup | Optional. Required only when you use the default security groups that the cluster operator role creates. |
| ec2:DescribeSubnets | Required. Creates master node, for example, describes the details of subnets. |
| ec2:DescribeVpc | Required. Creates master node, for example, describes the details of a VPC. |
| ec2:CreateTags | Required. Adds tags for Kubernetes infrastructure such as EC2. |
| ec2:ModifyInstanceAttribute | Required. Modifies attributes of an instance. |
| ec2:CreateVolume | Required. Creates storage such as EBS volumes. |
| ec2:DescribeVolumes | Required. Gets details of created volumes for ED2 node. |
| ec2:DescribeVolumesModifications | Required. Describes the most recent volume modification request for the specified EBS volumes. |
| ec2:ModifyVolume | Required. Modifies the volumes. |
| ec2:AttachVolume | Required. Attaches the volumes. |
| ec2:DetachVolume | Required. Detaches the created volumes. |
| ec2>DeleteVolume | Required. Deletes the created volumes. |

Amazon S3 actions

The following table describes the Amazon S3 actions that the master role requires and the resources that each action must apply to:

| Action | Resource | Description |
|-------------------------------|--|---|
| s3:GetBucketLocation | "arn:aws:s3:::<cluster-staging-bucket-name>" "arn:aws:s3:::<cluster-logging-bucket-name>" "arn:aws:s3:::<cluster-init-script-bucket-name>" | Required. The action must apply to the initialization script location if you use an initialization script to start the cluster. |
| s3:GetEncryptionConfiguration | "arn:aws:s3:::<cluster-staging-bucket-name>" "arn:aws:s3:::<cluster-logging-bucket-name>" | Required |
| s3:ListBucket | "arn:aws:s3:::<cluster-staging-bucket-name>" "arn:aws:s3:::<cluster-logging-bucket-name>" "arn:aws:s3:::<cluster-init-script-bucket-name>" | Required. The action must apply to the initialization script location if you use an initialization script to start the cluster. |
| s3:PutObject | "arn:aws:s3:::<cluster-staging-dir>/*" "arn:aws:s3:::<cluster-logging-dir>/*" | Required |
| s3:GetObjectAcl | "arn:aws:s3:::<cluster-staging-dir>/*" "arn:aws:s3:::<cluster-logging-dir>/*" | Required |
| s3:GetObject | "arn:aws:s3:::<cluster-staging-dir>/*" "arn:aws:s3:::<cluster-logging-dir>/*" "arn:aws:s3:::<cluster-init-script-dir>/*" | Required. The action must apply to the initialization script location if you use an initialization script to start the cluster. |
| s3>DeleteObject | "arn:aws:s3:::<cluster-staging-dir>/*" "arn:aws:s3:::<cluster-logging-dir>/*" | Required |
| s3:PutObjectAcl | "arn:aws:s3:::<cluster-staging-dir>/*" "arn:aws:s3:::<cluster-logging-dir>/*" | Required |

AWS Auto Scaling actions

The master node manages the Auto Scaling group to enable scalable cluster nodes and node recovery.

The master role requires the following actions to manage the Auto Scaling group:

```
autoscaling:DescribeAutoScalingInstances
autoscaling:DescribeTags
autoscaling:DescribeAutoScalingGroups
autoscaling:DescribeLaunchConfigurations
autoscaling:DescribeScalingActivities
autoscaling:SetDesiredCapacity
autoscaling:TerminateInstanceInAutoScalingGroup
autoscaling:UpdateAutoScalingGroup
```

AWS Key Management Service actions

The master role requires the following actions on all AWS resources to manage access to master keys:

```
kms:Encrypt
kms:Decrypt
kms:ReEncrypt
```

```
kms:GenerateDataKey
kms:DescribeKey
```

Elastic Load Balancing actions

The master node manages load balancing rules for an advanced cluster.

The master role requires the following actions on all AWS resources:

```
elasticloadbalancing:AddTags
elasticloadbalancing:AttachLoadBalancerToSubnets
elasticloadbalancing:DetachLoadBalancerFromSubnets
elasticloadbalancing:ApplySecurityGroupsToLoadBalancer
elasticloadbalancing:ConfigureHealthCheck
elasticloadbalancing:DescribeLoadBalancers
elasticloadbalancing>DeleteLoadBalancer
elasticloadbalancing:DescribeListeners
elasticloadbalancing:ModifyListener
elasticloadbalancing>DeleteLoadBalancerListeners
elasticloadbalancing:DescribeLoadBalancerAttributes
elasticloadbalancing:ModifyLoadBalancerAttributes
elasticloadbalancing:RegisterInstancesWithLoadBalancer
elasticloadbalancing:DeregisterInstancesFromLoadBalancer
elasticloadbalancing:SetLoadBalancerPoliciesForBackendServer
elasticloadbalancing:DescribeListener
elasticloadbalancing>DeleteListener
elasticloadbalancing:DescribeTargetGroups
elasticloadbalancing:ModifyTargetGroup
elasticloadbalancing:RegisterTargets
elasticloadbalancing:DescribeTargetHealth
elasticloadbalancing>DeleteTargetGroup
elasticloadbalancing:DeregisterTargets
elasticloadbalancing:SetLoadBalancerPoliciesOfListener
elasticloadbalancing:DescribeLoadBalancerPolicies
```

Identity and Access Management actions

The Identity and Access Management actions apply to all AWS resources.

The following table describes the actions:

| Action | Description |
|----------------------------|--------------------------------------|
| iam:ListServerCertificates | Required. Lists server certificates. |
| iam:GetServerCertificate | Required. Gets server certificates. |

Worker role actions

Add actions to the IAM policy for the worker role to allow the role to access and manage cloud resources.

The worker role requires actions defined by the following services on AWS:

- Amazon EC2
- Amazon S3
- AWS Auto Scaling
- AWS Key Management Service

Amazon EC2 actions

Amazon Elastic Compute Cloud (EC2) provides computing resources on the cloud.

The following table describes the Amazon EC2 actions that the worker role requires:

| Action | Resource | Description |
|-----------------------------|--|---|
| ec2:DescribeInstances | All -- "*" | Required. Allows Kubernetes to describe instances. |
| ec2:DescribeRegions | All -- "*" | Required. Allows Kubernetes to describe regions. |
| ec2:CreateTags | All -- "*" | Required. Adds tags for Kubernetes infrastructure, for example EC2. |
| ec2:DescribeVolumes | All -- "*" | Required for storage scaling. |
| ec2:CreateVolume | All -- "*" | Required for storage scaling. |
| ec2:ModifyInstanceAttribute | All -- "*" | Required for storage scaling. |
| ec2:AttachVolume | "arn:aws:ec2:*:*:volume/*" "arn:aws:ec2:*:*:instance/*" | Required for storage scaling. |

Amazon S3 actions

The following table describes the Amazon S3 actions that the worker role requires and the resources that each action must apply to:

| Action | Resource | Description |
|-------------------------------|--|---|
| s3:GetBucketLocation | "arn:aws:s3:::<cluster-staging-bucket-name>" "arn:aws:s3:::<cluster-logging-bucket-name>" "arn:aws:s3:::<cluster-init-script-bucket-name>" | Required. The action must apply to the initialization script location if you use an initialization script to start the cluster. |
| s3:GetEncryptionConfiguration | "arn:aws:s3:::<cluster-staging-bucket-name>" "arn:aws:s3:::<cluster-logging-bucket-name>" | Required. |
| s3:ListBucket | "arn:aws:s3:::<cluster-staging-bucket-name>" "arn:aws:s3:::<cluster-logging-bucket-name>" "arn:aws:s3:::<cluster-init-script-bucket-name>" | Required. The action must apply to the initialization script location if you use an initialization script to start the cluster. |
| s3:PutObject | "arn:aws:s3:::<cluster-staging-dir>/*" "arn:aws:s3:::<cluster-logging-dir>/*" | Required. |
| s3:GetObjectAcl | "arn:aws:s3:::<cluster-staging-dir>/*" "arn:aws:s3:::<cluster-logging-dir>/*" | Required. |
| s3:GetObject | "arn:aws:s3:::<cluster-staging-dir>/*" "arn:aws:s3:::<cluster-logging-dir>/*" "arn:aws:s3:::<cluster-init-script-dir>/*" | Required. The action must apply to the initialization script location if you use an initialization script to start the cluster. |

| Action | Resource | Description |
|-----------------|--|-------------|
| s3:DeleteObject | "arn:aws:s3:::<cluster-staging-dir>/*" "arn:aws:s3:::<cluster-logging-dir>/*" | Required. |
| s3:PutObjectAcl | "arn:aws:s3:::<cluster-staging-dir>/*" "arn:aws:s3:::<cluster-logging-dir>/*" | Required. |

AWS Auto Scaling actions

The worker role requires Auto Scaling actions on all AWS resources.

The following table describes the Auto Scaling actions:

| Action | Description |
|--|--|
| autoscaling:DescribeAutoScalingInstances | Required. Allows Kubernetes to describe autoscaling instances. |
| autoscaling:DescribeTags | Required. Allows Kubernetes to describe tags. |

AWS Key Management Service actions

The worker role requires the following actions on all AWS resources to manage access to master keys:

```
kms:Encrypt
kms:Decrypt
kms:ReEncrypt
kms:GenerateDataKey
kms:DescribeKey
```

Master and worker role types reference

Compare user-defined and default master and worker roles to decide which role types better fulfill your organizational requirements.

The following table compares each role type based on key areas:

| Area | User-defined roles | Default roles |
|---|---|--|
| Creation of master and worker roles | You have greater visibility of the master and worker roles and the policies that are attached to each role. | Roles are created automatically and it is more difficult to monitor the policies that are attached to each role. |
| Ability to edit policies | You can restrict some resources in the policies. | You cannot edit the policies. |
| Number of IAM permissions that the cluster operator role requires | Fewer IAM permissions are required. | More IAM permissions are required. |

| Area | User-defined roles | Default roles |
|--|---|--|
| Credential-based security for direct access to Amazon data sources | No impact on master and worker roles. | No impact on master and worker roles. |
| Role-based security for direct access to Amazon data sources | You must manually verify that the worker role and the Secure Agent role can both access the data sources that you use in an advanced job. You can also configure cross-account access to S3 buckets in multiple Amazon accounts. | You must only verify that the Secure Agent role can access the data sources that you use in an advanced job. The worker role can always access the same data sources as the Secure Agent role because the policies that are attached to the Secure Agent role are automatically attached to the worker role. You cannot configure cross-account access to S3 buckets in multiple Amazon accounts. |
| Role-sharing | You can use the same master and worker roles across multiple advanced configurations. | Separate master and worker roles are created for each advanced configuration. You cannot reuse roles. |
| Modifying staging and log locations | You must manually update the staging and log locations in the policies. | Policies are automatically updated. |
| Product upgrades | A product upgrade might change the policies that the master and worker roles require. If the policies change, you must regenerate the policy content and restrict access to resources again. | Policies are automatically updated. |

For more information about how the master and worker roles are used, see ["Learn about resource access" on page 16](#).

Master and worker policy restriction reference

You can restrict resources in the master and worker policies to limit the resources that the master and worker nodes can access.

You can restrict the following elements depending on their values:

Resource elements with the value *

If the value for a Resource element is the wildcard *, you cannot restrict the resources.

For example, the generated policy for the master node might have the following statement:

```
{
  "Effect": "Allow",
  "Action": [
    "ec2:DescribeInstances",
    "ec2:DescribeRegions",
    "ec2:DescribeRouteTables",
    "ec2:DescribeSecurityGroups",
    "ec2:DescribeSubnets",
    "ec2:DescribeVolumes"
  ]
}
```

```

    ],
    "Resource": [
        "*"
    ]
  },

```

Because the value for the Resource element is the wildcard *, you cannot edit the Resource element.

If you edit a Resource element whose value is the wildcard *, the Secure Agent might fail to identify the required resources to start an advanced cluster and the cluster might not start properly.

If you encrypt staging data and log files using SSE-KMS, you can edit the resources in the statement that contains actions on AWS Key Management Service (KMS) even though the Resource element is the wildcard *. For more information, see [“Encrypt staging data and log files at rest \(optional\)” on page 42](#).

Resource elements without the value *

If the value for a Resource element is not the wildcard *, you can restrict the Resource element to specify the resources that the statement covers.

For example, a generated policy for the worker node might have the following statement:

```

{
  "Effect": "Allow",
  "Action": [
    "s3:Get*"
  ],
  "Resource": [
    "arn:aws:s3:::<cluster-staging-dir1>/*",
    "arn:aws:s3:::<cluster-staging-dir2>/*"
  ]
},

```

Because the value for the Resource element is not the wildcard *, you can edit the resources in the statement. In this example, you can restrict the Resource element to the S3 resources that define one or more staging locations.

You can provide staging, log, and initialization script locations for multiple advanced clusters to share the same policy content between clusters that use different advanced configurations.

To avoid cross-region data-transfer costs, use S3 buckets that are in the same region. To help you manage each bucket, use different buckets for staging locations, log locations, initialization scripts, and data sources.

CHAPTER 3

Setting up Google Cloud

Before you create an advanced configuration in your organization, set up your cloud environment so that the Secure Agent can create an advanced cluster.

Complete the following tasks:

1. Verify the requirements for your environment.
2. Create storage locations for cluster files.
3. Optionally, create a VPC and subnets.
4. Download and install the Secure Agent on a Linux virtual machine on Google Cloud.
5. Allow specific domains in Google Cloud.
6. Optionally, configure a proxy for the cluster.
7. Create roles and service accounts.
8. Optionally, configure the JAVA_HOME environment variable.
9. Create a staging connection.

Step 1. Complete prerequisites

Before you set up your environment, verify the requirements for your environment and your cloud platform.

Complete the following tasks:

- Verify that you have the correct privileges in your organization.
- Verify that you have the necessary Google Cloud services.
- Learn how the Secure Agent and the advanced cluster access resources on your cloud platform.

Verify privileges in your organization

Verify that you are assigned the correct privileges for advanced configurations in your organization.

Privileges for advanced configurations provide you varying access levels to the **Advanced Clusters** page in Administrator as well as Monitor.

You must have at least the read privilege to view the advanced configurations and to monitor the advanced clusters.

Verify Google Cloud services

Verify that you have the necessary services to create an advanced cluster on Google Cloud.

You must have the following services on your Google account:

Google Cloud Storage

Staging data and log files for an advanced cluster and advanced jobs are stored on Google Cloud Storage.

Google Compute Engine

A virtual machine hosts the Secure Agent.

VPC Network

A VPC network and subnet to host advanced cluster.

Network Service

A network service to provide load-balancing and Cloud NAT.

Learn about resource access

To process data, the Secure Agent and the advanced cluster access the resources that are part of an advanced job, including resources on the cloud platform, source and target data, and staging and log locations.

Resources are accessed to perform the following tasks:

- Design a mapping
- Create an advanced cluster
- Run a job, including data preview
- Poll logs

Designing a mapping

When you design a mapping, the Secure Agent accesses sources and targets so that you can read and write data.

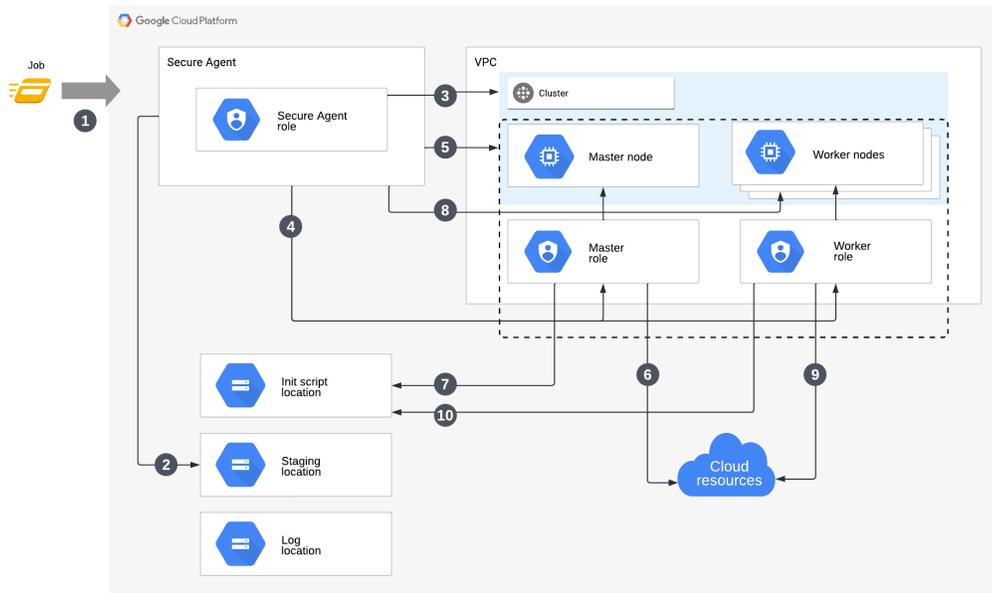
For example, when you add a Source transformation to a mapping, the Secure Agent accesses the source to display the fields that you can use in the rest of the mapping. The Secure Agent also accesses the source when you preview data.

To access a source or target, the Secure Agent uses the permissions in the Secure Agent service account.

Creating an advanced cluster

To create an advanced cluster, the Secure Agent uses the Secure Agent role to store cluster details in the staging location and to create the cluster. The master and worker nodes use either the master and worker roles or the Secure Agent role to access cloud resources.

The following image shows the process that the Secure Agent uses to create a cluster:



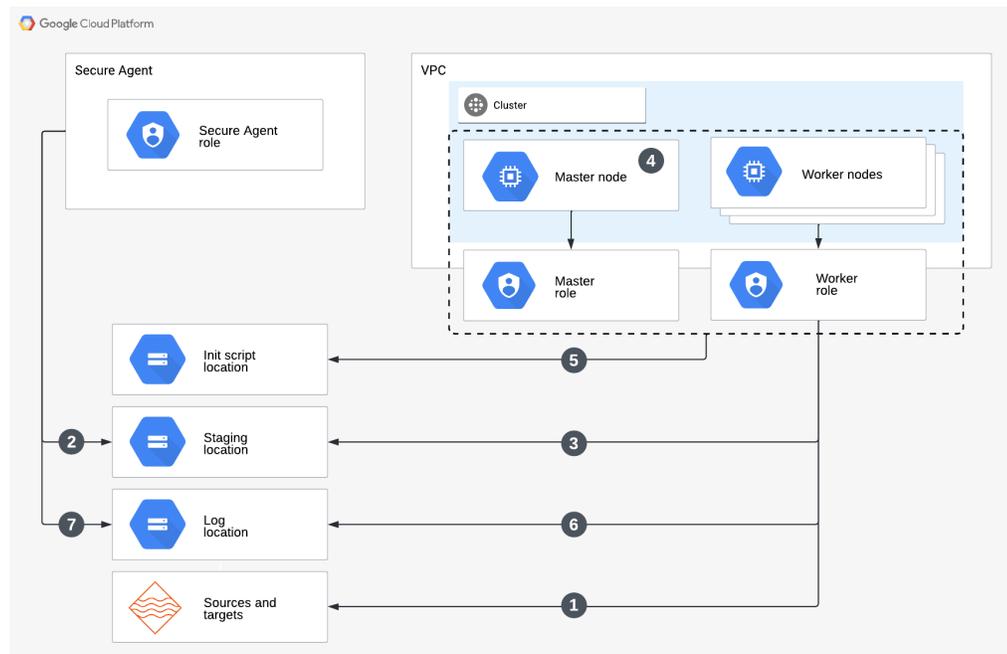
The following steps describe the process that the Secure Agent uses to create a cluster:

1. You run a job.
2. The Secure Agent uses the Secure Agent role to store cluster details in the staging location.
3. The Secure Agent uses the Secure Agent role to create the cluster.
4. If you create master and worker roles and service accounts, the Secure Agent attaches the service accounts to the cluster nodes.
5. The Secure Agent uses the Secure Agent role to create cluster resources for the master node.
6. The master node uses the master role to access cloud resources.
7. The master node uses the master role to access the initialization script. If you didn't create master and worker roles and service accounts, the master node uses the Secure Agent role.
8. The Secure Agent uses the Secure Agent role to create cluster resources for the worker nodes and creates a managed instance group with the minimum number of worker nodes.
9. The worker nodes use the worker role to access cloud resources. If you didn't create master and worker roles and service accounts, the worker nodes use the Secure Agent role.
10. The worker nodes use the worker role to access the initialization script. If you didn't create master and worker roles and service accounts, the worker nodes use the Secure Agent role.

Running a job

To run a job, the Secure Agent, master node, and worker nodes access sources and targets, as well as the staging, log, and initialization script locations.

The following image shows the process that the Secure Agent and cluster nodes use to run the job:



The following steps describe the process that the Secure Agent and cluster nodes use to run the job:

1. The worker nodes use the worker role to access source and target data.
2. The Secure Agent uses the Secure Agent role to store job dependencies in the staging location.
3. The worker nodes use the worker role to get job dependencies and stage temporary data in the staging location.
4. The master node uses the master role to orchestrate processes on the cluster.
5. The master node uses the master role to access and run the initialization script on the master node and to scale up the worker nodes. The added worker node uses the worker role to access the initialization script again to run the script on the worker node.
6. The worker nodes use the worker role to store logs in the log location.
7. The Secure Agent uses the Secure Agent role to upload the agent job log to the log location.

If you create master and worker roles and service accounts, the master and worker nodes use their respective roles. Otherwise, the master and worker nodes use the Secure Agent role.

Polling logs

When you use Monitor, the Secure Agent accesses the log location to poll logs.

To poll logs from the log location, the Secure Agent uses the permissions in the Secure Agent service account.

Learn about Google Cloud cluster

The Google Cloud cluster uses the Google Cloud CentOS 7 OS image published by Informatica.

The following packages are included in the image:

- cloud-init
- device-mapper-persistent-data
- docker-ce
- gnupg2
- gzip
- kernel-devel
- kernel-headers
- kubeadm
- kubelet
- libxml2-python
- lvm2
- tar
- unzip
- wget
- yum-utils

The OS image also includes the following docker images:

- calico/kube-controllers
- calico/node
- calico/cni
- calico/pod2daemon-flexvol
- coreos/flannel
- coreos/flannel-cni
- imega/jq
- kube-scheduler

Step 2. Create storage locations for cluster files

In Google Cloud Storage, create locations to store staging, log, and initialization script files.

Create the following storage locations:

- A location that the cluster will use to store staging files at run time
- A location that the cluster will use to store log files for the advanced jobs that run on the cluster
- Optionally, a location where you can store initialization scripts that cluster nodes will run to install additional software on the cluster

The staging location stores temporary data, such as artifacts that the cluster distributes across cluster nodes and data that you preview in a mapping. Because an error might prevent a mapping from clearing preview

data in the staging location, make sure that the users who have access to the staging location are permitted to view source data.

If you create any initialization scripts, add the scripts to the appropriate location.

Step 3. Create the VPC and subnets (optional)

If you create your own VPC and subnets to host an advanced cluster, prepare the VPC network and subnets according to cluster requirements.

To prepare the network and subnets, complete the following tasks after you create a VPC:

1. Create a subnet that supports enough IP addresses for the nodes in the advanced cluster.
2. Create a Google Cloud NAT gateway.
3. Create firewall rules in the VPC network to allow TCP traffic.

Create a subnet with enough IP addresses

Create a subnet that supports enough IP addresses for all the nodes in the advanced cluster within your VPC network.

Calculate the number of required IP addresses according to the following guidelines:

- Add one IP address for the master node.
- Add IP addresses equal to the maximum number of worker nodes.

For example, if the advanced cluster can have a maximum of 10 worker nodes, each subnet must support at least 11 IP addresses.

Create a Google Cloud NAT gateway

If you need to connect to the internet from private nodes that do not have external IP addresses, create a Google Cloud Network Address Translator (NAT) gateway.

In Google Cloud NAT, create a NAT gateway in the VPC network with the following configuration:

- Use the same region as the subnet.
- Use a Cloud Router that uses the default settings.
- Use the default value for the NAT mapping source.
- Manually create a new static public IP address to use for the NAT IP address.

Ensure that the NAT gateway is running before you run an advanced job.

The following image shows an example NAT gateway configuration in the Google Cloud Console:

Network services ← Create a NAT gateway

Cloud NAT lets your VM instances and container pods communicate with the internet using a shared, public IP address.

Cloud NAT uses NAT gateway to manage those connections. A NAT gateway is region and VPC network specific. If you have VM instances in multiple regions, you'll need to create a NAT gateway for each region. [Learn more](#)

Gateway name *
dev-example-nat
Lowercase letters, numbers, hyphens allowed

Select Cloud Router

Network *
dev-example-vpc

Region *
us-west2 (Los Angeles)
One subnet

Cloud Router *
new-router

NAT mapping

Source (internal)
Primary and secondary ranges for all subnets
Select which subnets to map to the NAT gateway. Primary IP addresses are used by VM instances and secondary IP addresses are used by container pods. [Learn more](#)

NAT IP addresses
Manual

IP address
privateipaddress

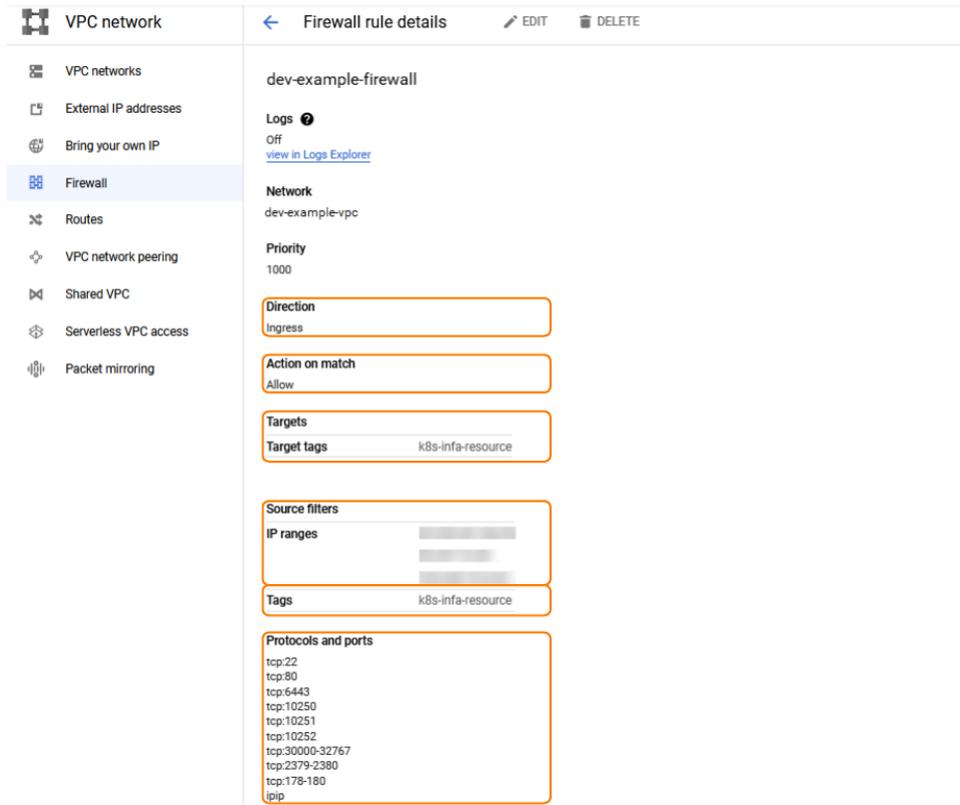
Create firewall rules in the VPC network

Create a firewall rule for the VPC network to allow TCP traffic from the IP addresses of the Secure Agent machine and the NAT gateway.

In Google Cloud, create a firewall rule for the VPC network with the following configuration:

- Set the direction of traffic to ingress traffic.
- Allow matches.
- Add the following target tag: `k8s-infa-resource`
- Set the primary source filter to filter by IP ranges. Use CIDR notation to set the source IP ranges to the static IP addresses of the Secure Agent machine and the NAT gateway created in step 2.
- Set the secondary source filter to filter by source tags. Add the following source tag: `k8s-infa-resource`
- Specify the following protocols and ports:
 - TCP ports: 22, 80, 6443, 10250, 10251, 10252, 30000-32767, 2379-2380, 178-180
 - Other protocols: `ipip`

The following image shows how the firewall rule might appear in the Google Cloud Console:



Step 4. Download and install a Secure Agent

Download and install a Secure Agent on a Linux virtual machine on Google Cloud.

The following table lists the minimum resource requirements on the Secure Agent machine:

| Component | Minimum requirement |
|---------------|---------------------|
| Cores per CPU | At least four |
| Memory | 16 GB |
| Disk Space | 100 GB |

After you install a Secure Agent, install OpenSSL on the Secure Agent machine.

For more information about installing a Secure Agent, see [Secure Agent installation](#).

Step 5. Allow domains in Google Cloud

When the Secure Agent creates an advanced cluster in Google Cloud, the cluster nodes need access to certain domains to fetch artifacts, such as machine images, and to access sources and targets.

Add the following domains to the outbound allowlist of your firewall:

```
artifacthub.informaticacloud.com
infacloud.jfrog.io
infacloud-ct-cdie-docker.jfrog.io
infacloud-discale-docker-stable.jfrog.io
.storage.cloud.google.com
.google.com
.le100.net
https://storage.googleapis.com
```

Step 6. Configure a proxy for the cluster (optional)

Use a proxy server to create an indirect connection to network services for security and performance reasons. For example, you can use a proxy server to get through a firewall, and some proxies provide caching mechanisms.

To use a proxy server for the cluster, edit the proxy server for the Secure Agent. Exclude the metadata server on Google Cloud and the IP addresses that you plan to assign to the cluster.

You can edit the proxy server details in the following file:

```
<Secure Agent installation directory>/apps/agentcore/conf/proxy.ini
```

Configure the property `InfaAgent.NonProxyHost` to exclude IP addresses or host names.

To exclude the metadata server on Google Cloud and cluster IP addresses from the proxy, perform the following steps:

1. Open the `proxy.ini` file.
2. Update the value for `InfaAgent.NonProxyHost` to exclude the metadata server and cluster IP addresses.

For example, the following value excludes the metadata server and uses two formats to exclude cluster IP addresses in the CIDR block `172.16.0.0/16`:

```
InfaAgent.NonProxyHost=metadata|metadata.google.internal|172.16.*|172.16.0.0/16
```

Note: The pipe character (`|`) is a delimiter that combines a list of host names and IP addresses. You can enter a wildcard to the left for host names or to the right for IP addresses.

3. Restart the Secure Agent for the changes to take effect.

The proxy details appear on the Secure Agent Manager settings page for the proxy server.

After the changes take effect, the Secure Agent communicates with the metadata server and the cluster without passing through the proxy, while commands to communicate with the cluster are required to pass through the proxy.

For more information about configuring a proxy to exclude non-proxy hosts, see [Using a proxy server for the Secure Agent](#).

Step 7. Create roles and service accounts

Create a Secure Agent role and service account to grant the agent permissions to create and manage an advanced cluster on Google Cloud. You can include the master node and worker node permissions in the Secure Agent role, or you can create separate roles and service accounts for the cluster nodes.

Create the following roles and Google service accounts:

- Secure Agent role and service account
- Optionally, a master node role and service account
- Optionally, a worker node role and service account

A Google Cloud service account is always linked to a Google Cloud project. Make sure that you use only one set of credentials for both the source and target when you run an advanced job.

Create a Secure Agent role and service account

Create a Secure Agent role and service account to grant permissions to the Secure Agent.

Create a Secure Agent role

Create a Secure Agent role to define the set of permissions for the Secure Agent.

1. In the Google Cloud web console, navigate to **IAM & Admin > Roles**.
2. Create a role.
3. Enter a role title, description, and ID.

You can use `<username-agent-role>` as a format for the ID.

4. Add permissions for the role.

For more information about permissions, see [“Permissions for the Secure Agent role” on page 73](#).

Create a Secure Agent service account

Create a Secure Agent service account that uses the Secure Agent role.

1. In the Google Cloud web console, navigate to **IAM & Admin > Service Accounts**.
2. Create a service account.
3. Enter service account details such as name, ID, and description.
4. Enter details for the service account access to the project.
5. Select the Secure Agent role `<username-agent-role>`.
6. Set the Secure Agent service account as the default service account on the Secure Agent machine.

Permissions for the Secure Agent role

The following table lists the minimum required permissions for the Secure Agent role:

| Operations | Permissions |
|---|--|
| <ul style="list-style-type: none"> - Create an external static IP address - Delete or release an IP address | <pre>compute.addresses.create compute.addresses.delete compute.addresses.get compute.addresses.list compute.addresses.use</pre> |
| <ul style="list-style-type: none"> - Create a target pool - Get details for a target pool - Delete a target pool | <pre>compute.targetPools.addInstance compute.targetPools.create compute.targetPools.delete compute.targetPools.get compute.targetPools.list compute.targetPools.removeInstance compute.targetPools.update compute.targetPools.use</pre> |
| <ul style="list-style-type: none"> - Create a forwarding rule - Get details for a rule creation - Delete a forwarding rule | <pre>compute.forwardingRules.create compute.forwardingRules.delete compute.forwardingRules.get compute.forwardingRules.list compute.forwardingRules.setTarget compute.forwardingRules.update</pre> |
| <ul style="list-style-type: none"> - Create an instance template - Get details for an instance template - Delete an instance template - Add a disk to an instance | <pre>compute.instanceTemplates.create compute.instanceTemplates.delete compute.instanceTemplates.get compute.instanceTemplates.list compute.instanceTemplates.useReadOnly compute.disks.create compute.disks.delete compute.disks.get compute.disks.list compute.disks.resize compute.disks.setLabels compute.disks.update compute.disks.use</pre> |

| Operations | Permissions |
|---|---|
| <ul style="list-style-type: none"> - Create a regional and zonal group - Get details or description of regional instance groups - Delete a regional instance group | <pre> compute.addresses.create compute.addresses.delete compute.addresses.get compute.addresses.list compute.addresses.use compute.instanceGroupManagers.create compute.instanceGroupManagers.delete compute.instanceGroupManagers.get compute.instanceGroupManagers.list compute.instanceGroupManagers.update compute.instanceGroupManagers.use compute.instanceGroups.create compute.instanceGroups.delete compute.instanceGroups.get compute.instanceGroups.list compute.instanceGroups.update compute.instanceGroups.use compute.instances.addAccessConfig compute.instances.attachDisk compute.instances.create compute.instances.delete compute.instances.deleteAccessConfig compute.instances.detachDisk compute.instances.get compute.instances.getEffectiveFirewalls compute.instances.list compute.instances.osAdminLogin compute.instances.osLogin compute.instances.reset compute.instances.resume compute.instances.setDiskAutoDelete compute.instances.setLabels compute.instances.setMachineResources compute.instances.setMachineType compute.instances.setMetadata compute.instances.setMinCpuPlatform compute.instances.setServiceAccount compute.instances.setTags compute.instances.start compute.instances.startWithEncryptionKey compute.instances.stop compute.instances.suspend compute.instances.update compute.instances.updateAccessConfig compute.instances.updateNetworkInterface compute.instances.updateSecurity compute.instances.use compute.subnetworks.use compute.subnetworks.useExternalIp compute.subnetworks.get </pre> |
| <ul style="list-style-type: none"> - Delete, upload, and list Google Cloud Storage metadata and logs | <pre> storage.objects.create storage.objects.delete storage.objects.get storage.objects.list storage.objects.update storage.buckets.get </pre> |
| <ul style="list-style-type: none"> - Create, use, and delete a resource within a VPC and subnet | <pre> compute.subnetworks.get compute.subnetworks.use compute.subnetworks.useExternalIp </pre> |
| <ul style="list-style-type: none"> - Work with a project | <pre> resourcemanager.projects.get </pre> |

| Operations | Permissions |
|--|--|
| - Use a service account | iam.serviceAccounts.actAs |
| - Create, use, and delete an internal IP address | compute.addresses.createInternal compute.addresses.deleteInternal compute.addresses.useInternal |
| - Create, use, and delete a regional backend service | compute.regionBackendServices.create compute.regionBackendServices.delete compute.regionBackendServices.get compute.regionBackendServices.list compute.regionBackendServices.update compute.regionBackendServices.use |
| - Create, use, and delete a regional health check | compute.regionHealthChecks.create compute.regionHealthChecks.delete compute.regionHealthChecks.get compute.regionHealthChecks.list compute.regionHealthChecks.update compute.regionHealthChecks.use compute.regionHealthChecks.useReadOnly |

To allow the Secure Agent to create a VPC network and subnets, add the following permissions to the Secure Agent role:

| Operations | Permissions |
|---|---|
| - Create, use, and delete a VPC network | compute.networks.access compute.networks.create compute.networks.delete compute.networks.get compute.networks.list compute.networks.use |
| - Create, use, and delete a subnetwork | compute.subnetworks.create compute.subnetworks.delete compute.subnetworks.get compute.subnetworks.list compute.subnetworks.update compute.subnetworks.use compute.subnetworks.useExternalIp |
| - Create, use, and delete a Cloud Router | compute.routers.create compute.routers.delete compute.routers.get compute.routers.list compute.routers.use |
| - Create, use, and delete a firewall rule - Add a firewall rule to a VPC network | compute.firewalls.create compute.firewalls.delete compute.firewalls.get compute.firewalls.list compute.firewalls.update compute.networks.updatePolicy |

If you do not create separate roles and service accounts for the cluster nodes, add the following permissions to the Secure Agent role:

| Node type | Operations | Permissions |
|-----------|--|---|
| Master | - Scale up or down an instance group for worker nodes | compute.regions.get compute.instanceGroups.list compute.instanceGroups.update compute.instanceGroups.use compute.instanceGroups.get |
| Worker | - Upload initialization script notification to the staging location - Upload initialization script logs to the log location | storage.objects.create storage.objects.delete storage.objects.get storage.objects.list storage.objects.update |

Create a master role and service account

Optionally, you can create a separate master role and service account to reduce the number of permissions that are assigned to the Secure Agent role. The master role will grant the permissions only to the master node.

Create a master role

Create a master role to define the set of permissions for the master node.

1. In the Google Cloud web console, navigate to **IAM & Admin > Roles**.
2. Create a role.
3. Enter a role title, description, and ID.

You can use `<username-master-role>` as a format for the ID.

4. Add permissions to the role.

The following table describes the permissions that the role needs:

| Operations | Permissions |
|---|---|
| - Scale up or down an instance group for worker nodes | compute.regions.get compute.instanceGroups.list compute.instanceGroups.update compute.instanceGroups.use compute.instanceGroups.get |

Create a master service account

Create a master service account that uses the master role.

1. In the Google Cloud web console, navigate to **IAM & Admin > Service Accounts**.
2. Create a service account.
3. Enter service account details such as name, ID, and description.
4. Enter details for the service account access to the project.
5. Select the master role `<username-master-role>`.

Create a worker node role and service account

Optionally, you can create a separate worker node role and service account to reduce the number of permissions that are assigned to the Secure Agent role. The worker role will grant the permissions only to the worker nodes.

Create a worker role

Create a worker role to define the set of permissions for the worker nodes.

1. In the Google Cloud web console, navigate to **IAM & Admin > Roles**.
2. Create a role.
3. Enter a role title, description, and ID.

You can use `<username-worker-role>` as a format for the ID.

4. Add permissions to the role.

The following table describes the permissions that the role needs:

| Operations | Permissions |
|---|--|
| <ul style="list-style-type: none">- Upload initialization script notification to the staging location- Upload initialization script logs to the log location | <code>storage.objects.create</code> <code>storage.objects.delete</code> <code>storage.objects.get</code> <code>storage.objects.list</code> <code>storage.objects.update</code> |

Create a worker service account

Create a worker service account that uses the worker role.

1. In the Google Cloud web console, navigate to **IAM & Admin > Service Accounts**.
2. Create a service account.
3. Enter service account details such as name, ID, and description.
4. Enter details for the service account access to the project.
5. Select the worker role `<username-worker-role>`.

Step 8. Configure the JAVA_HOME environment variable

To run commands such as the `cluster-operations.sh` command, you must configure the `JAVA_HOME` environment variable on the Secure Agent machine.

The Java version on the Secure Agent machine must be compatible with JDK 8.

Step 9. Create a staging connection

Create a staging connection to the staging location so that the advanced cluster can share staging data with the Data Integration Server.

1. In Administrator, open the **Connections** page.
2. Create a connection to Google Cloud Storage.
In the connection properties, enter the bucket name of the location that you created to store staging files in [“Step 2. Create storage locations for cluster files” on page 67](#).
3. Open the **Advanced Clusters** page.
4. Create an advanced configuration or edit the existing advanced configuration for the cluster.
5. On the **Platform Configuration** tab, configure the staging location to specify the same bucket name that you entered in the connection properties. You can specify a folder path within the bucket.
6. On the **Runtime Properties** tab, add the property `clusterconfig.stagingConnectionName` and set the value to the name of the connection.

CHAPTER 4

Setting up Microsoft Azure

Before you create an advanced configuration in your organization, set up your cloud environment so that the Secure Agent can create an advanced cluster.

Complete the following tasks:

1. Verify requirements for your environment.
2. Download and install a Secure Agent on a Linux virtual machine on the Azure cloud.
3. Allow specific domains in Azure.
4. Optionally, configure a proxy for the cluster.
5. Create storage accounts for cluster files.
6. Create a cluster resource group
7. Create a managed identity for the Secure Agent.
8. Create a service principal for the cluster.
9. Optionally, create a managed identity to access sources and targets.
10. Optionally, configure the JAVA_HOME environment variable.
11. Optionally, create a staging connection.

Step 1. Complete prerequisites

Before you set up your environment, verify the requirements for your environment and your cloud platform.

Complete the following tasks:

- Verify that you have the correct privileges in your organization.
- Verify that you have the necessary Microsoft Azure products.
- Learn how the Secure Agent and the advanced cluster access resources on your cloud platform.

Verify privileges in your organization

Verify that you are assigned the correct privileges for advanced configurations in your organization.

Privileges for advanced configurations provide you varying access levels to the **Advanced Clusters** page in Administrator as well as Monitor.

You must have at least the read privilege to view the advanced configurations and to monitor the advanced clusters.

Verify Microsoft Azure products

Verify that you have the necessary Microsoft Azure products to create an advanced cluster in an Azure environment.

You must have the following products on your Azure account:

Azure Data Lake Storage Gen2

Staging data and log files for an advanced cluster and jobs are stored on the Azure cloud.

Linux Virtual Machines

A Linux virtual machine hosts the Secure Agent.

Virtual Network (VNet)

An advanced cluster is created in a VNet. You can specify an existing VNet, or the Secure Agent can create a VNet based on the region that you provide.

Key Vault

If you create a service principal to perform cluster operations, a key vault stores the service principal credentials. The Secure Agent accesses the key vault to retrieve the credentials.

Load Balancer

A load balancer accepts incoming jobs from a Secure Agent and provides an entry point for the jobs to an advanced cluster.

Learn about resource access

To process data, the Secure Agent and the advanced cluster access the resources that are part of an advanced job, including resources on the cloud platform, source and target data, and staging and log locations.

Resources are accessed to perform the following tasks:

- Design a mapping
- Create an advanced cluster
- Run a job, including data preview
- Poll logs

Designing a mapping

When you design a mapping, the Secure Agent accesses sources and targets so that you can read and write data.

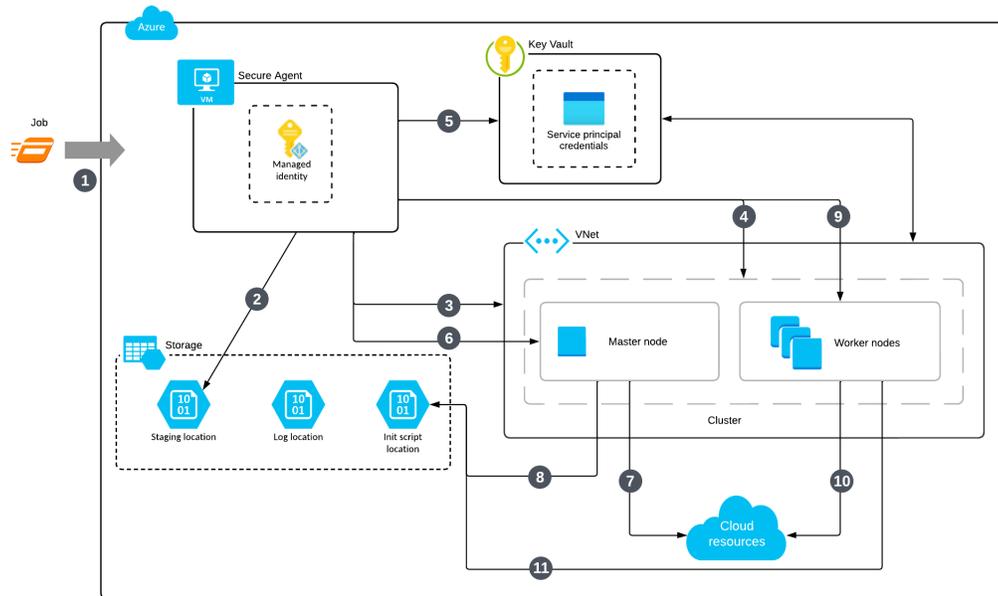
For example, when you add a Source transformation to a mapping, the Secure Agent accesses the source to display the fields that you can use in the rest of the mapping. The Secure Agent also accesses the source when you preview data.

To access a source or target, the Secure Agent uses the connection properties. For example, the Secure Agent might use the user name and password that you provide in the connection properties to access a database.

Creating an advanced cluster

To create an advanced cluster, the Secure Agent authenticates with the managed identity to store cluster details in the staging location and to create the cluster. The master and worker nodes use the service principal to access cloud resources.

The following image shows the process that the Secure Agent uses to create a cluster:



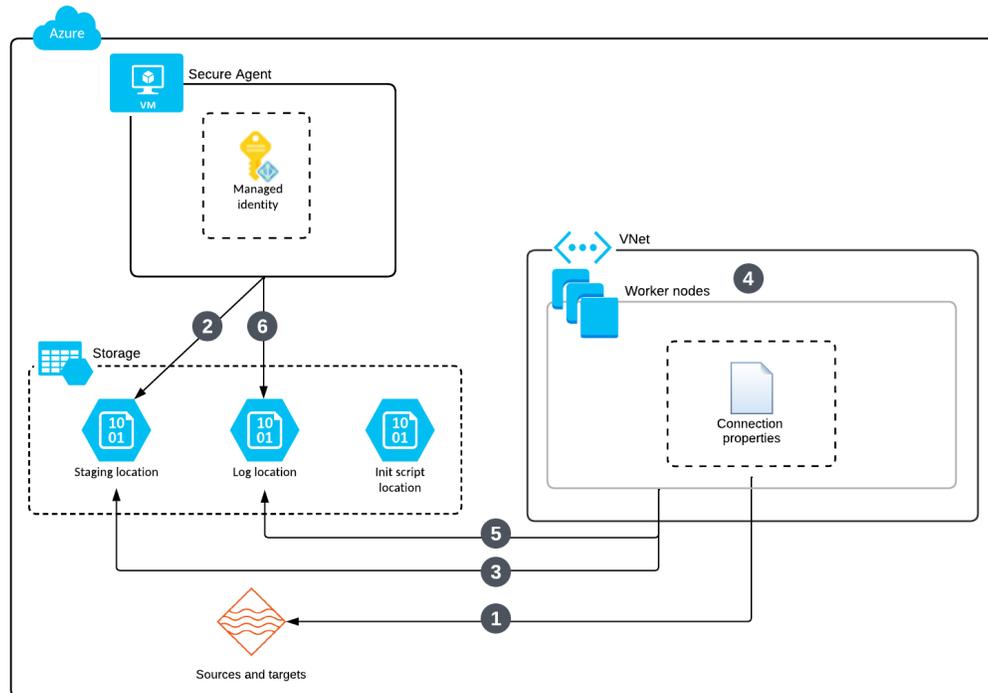
The following steps describe the process that the Secure Agent uses to create a cluster:

1. You run a job.
2. The Secure Agent authenticates with the managed identity to store cluster details in the staging location.
3. The Secure Agent authenticates with the managed identity to create prerequisite resources that the cluster needs, such as a network security group and load balancer.
4. The Secure Agent authenticates with the managed identity to get the access keys to the storage accounts and uses a secure channel to make the keys available to cluster nodes.
5. The Secure Agent makes the service principal credentials available to the cluster.
6. The Secure Agent authenticates with the managed identity to create cluster resources for the master node and a Virtual Machine Scale Set for the master node.
7. The master node uses the service principal to access cloud resources.
8. The master node accesses the initialization script using the storage account key that the Secure Agent fetched through the managed identity.
9. The Secure Agent authenticates with the managed identity to create cluster resources for the worker nodes and creates a Virtual Machine Scale Set with the minimum number of worker nodes.
10. The worker nodes use the service principal to access cloud resources.
11. The worker nodes access the initialization script using the storage account key that the Secure Agent fetched through the managed identity.

Running a job

To run a job, the Secure Agent and the worker nodes access sources and targets, as well as the staging and log locations. The worker nodes and the Azure disks auto-scale according to resource requirements.

The following image shows the process that the Secure Agent and worker nodes use to run the job:



The following steps describe the process that the Secure Agent and worker nodes use to run the job:

1. The worker nodes use the connection properties to access source and target data.
The connection properties access the data either using a storage account key or a managed identity. To use a managed identity, the identity must be assigned to the Secure Agent, and the agent role must have permissions to detect all user-assigned managed identities that are assigned to the Secure Agent machine, and be able to assign the identities to all cluster nodes.
2. The Secure Agent authenticates with the managed identity to store job dependencies in the staging location.
3. The worker nodes get job dependencies and stage temporary data in the staging location using the storage account key that the Secure Agent fetched through the managed identity. The Secure Agent also passes the key to the Spark job so that the Spark driver and Spark executors can use the same key to access the staging location.
4. The worker nodes and the Azure disks auto-scale using the service principal.
5. The worker nodes store logs in the log location after fetching the storage account key through the managed identity.
6. The Secure Agent authenticates with the managed identity to upload the agent job log to the log location.

Polling logs

When you use Monitor, the Secure Agent accesses the log location to poll logs.

To poll logs from the log location, the Secure Agent uses the permissions in the managed identity that is assigned to the Secure Agent machine.

Step 2. Download and install a Secure Agent

Download and install a Secure Agent on a Linux virtual machine on the Azure cloud. This VM is known as the Secure Agent machine.

The following table lists the minimum resource requirements on the Secure Agent machine:

| Component | Minimum requirement |
|---------------|---------------------|
| Cores per CPU | At least four |
| Memory | 16 GB |
| Disk Space | 100 GB |

After you install a Secure Agent, install OpenSSL on the Secure Agent machine.

For more information about installing a Secure Agent, see [Secure Agent installation](#).

Step 3. Allow domains in Azure

When the Secure Agent creates an advanced cluster in a Microsoft Azure environment, the cluster nodes need access to certain domains to fetch artifacts, such as machine images, and to access sources and targets to run mappings.

Add the following domains to the outbound allowlists of your network security groups:

```
artifacthub.informaticacloud.com
infacloud.jfrog.io
infacloud-ct-cdie-docker.jfrog.io
infacloud-discale-docker-stable.jfrog.io

*.azure.com
*.azure.net
*.database.windows.net
*.microsoft.com
*.microsoftonline.com
*.windows.net
azure.com
azure.net
ifconfig.me
microsoft.com
microsoftonline.com
windows.net
```

Note: To use the default NTP service configuration, open your firewall and set the default NTP service UDP port to 123 for all outbound servers. If you don't want to use this configuration, you can set up a custom

image that has a custom NTP service configuration. For more information, contact Informatica Global Customer Support.

Step 4. Configure a proxy for the cluster (optional)

Use a proxy server to create an indirect connection to network services for security and performance reasons. For example, you can use a proxy server to get through a firewall, and some proxies provide caching mechanisms.

To use a proxy server for the cluster, edit the proxy server for the Secure Agent. Exclude the IP addresses that you plan to assign to the cluster.

You can edit the proxy server details in the following file:

```
<Secure Agent installation directory>/apps/agentcore/conf/proxy.ini
```

Configure the property `InfaAgent.NonProxyHost` to exclude IP addresses.

Perform the following steps:

1. Open the `proxy.ini` file.
2. Update the value for `InfaAgent.NonProxyHost` to exclude the cluster IP addresses.

For example, the following value uses two formats to exclude cluster IP addresses in the CIDR block `172.16.0.0/16`:

```
InfaAgent.NonProxyHost=localhost|127.|[\\:\:1]|123.432.|172.16.*|172.16.0.0/16
```

To use managed identity authentication to connect to a source or target, exclude the IP address for the metadata service, `169.254.169.254`.

Note: The pipe character (`|`) is a delimiter that combines a list of host names and IP addresses. You can enter a wildcard to the left for host names or to the right for IP addresses.

3. Restart the Secure Agent for the changes to take effect.

The proxy details appear on the Secure Agent Manager settings page for the proxy server.

After the changes take effect, the Secure Agent communicates with the cluster without passing through the proxy, while commands to communicate with the cluster are required to pass through the proxy.

For more information about configuring a proxy to exclude non-proxy hosts, see

[Using a proxy server for the Secure Agent](#).

Step 5. Create storage accounts for cluster files

You can store data using Azure Data Lake Storage Gen2.

In Azure, create the following storage accounts using a hierarchical namespace:

- A storage account with the following locations:
 - A location that the cluster will use to store staging files at run time
 - A location that the cluster will use to store log files for the advanced jobs that run on the cluster
- Optionally, a storage account where you can store initialization scripts that cluster nodes will run to install additional software on the cluster

Then, add these storage accounts to a resource group named `storage_resource_group`.

The staging location stores temporary data, such as artifacts that the cluster distributes across cluster nodes and data that you preview in a mapping. Because an error might prevent a mapping from clearing preview data in the staging location, make sure that the users who have access to the staging location are permitted to view source data.

If you create any initialization scripts, add the scripts to the appropriate location.

Step 6. Create a cluster resource group

In Azure, create a resource group named `cluster_resource_group`.

The Secure Agent uses this resource group to store cluster resources such as master and worker node VMs, virtual machine scale sets, network interfaces, and load balancers.

Step 7. Create a managed identity for the Secure Agent

The Secure Agent uses a managed identity to log in to the Microsoft Azure cloud and to create an advanced cluster. If you run the `list-clusters.sh` and `delete-clusters.sh` commands, the Secure Agent uses the managed identity to authenticate to the Azure CLI.

In Azure, complete the following tasks:

1. Create a managed identity.
2. Create an agent role.
3. Add role assignments to assign the agent role to the managed identity and to assign the managed identity to the Secure Agent machine.

Create a managed identity

Create a managed identity named `agent_identity`.

You can use a system-assigned managed identity or create a user-assigned managed identity. If you create a user-assigned managed identity and there are multiple identities attached to the Secure Agent machine, set the Elastic Server property `azure_agent_role_identity_client_id` to the client ID of `agent_identity`.

For information about creating a managed identity, refer to the Microsoft Azure documentation. Microsoft Azure provides best practices for managed identities and can help you decide whether to use a system-assigned or user-assigned managed identity.

Create an agent role

Create an agent role to define the permissions for the managed identity `agent_identity`.

Create a custom role named `agent_role` with the following role definition:

```
{
  "properties":{
```

```

"roleName":"agent_role",
"description":"",
"assignableScopes":[
  "/subscriptions/<subscription ID>/resourceGroups/<cluster_resource_group>",
  "/subscriptions/<subscription ID>/resourceGroups/<storage_resource_group>",
  "/subscriptions/<subscription ID>/resourceGroups/<vnet_resource_group>"
],
"permissions":[
  {
    "actions":[
      "Microsoft.Resources/subscriptions/resourcegroups/read",
      "Microsoft.Storage/storageAccounts/read",
      "Microsoft.Storage/storageAccounts/write",
      "Microsoft.Storage/storageAccounts/listKeys/action",
      "Microsoft.Compute/virtualMachineScaleSets/delete",
      "Microsoft.Compute/virtualMachineScaleSets/write",
      "Microsoft.Compute/virtualMachineScaleSets/read",
      "Microsoft.Network/loadBalancers/delete",
      "Microsoft.Network/loadBalancers/write",
      "Microsoft.Network/loadBalancers/read",
      "Microsoft.Network/networkSecurityGroups/delete",
      "Microsoft.Network/networkSecurityGroups/write",
      "Microsoft.Network/networkSecurityGroups/read",
      "Microsoft.Network/virtualNetworks/delete",
      "Microsoft.Network/virtualNetworks/write",
      "Microsoft.Network/virtualNetworks/read",
      "Microsoft.Network/publicIPAddresses/delete",
      "Microsoft.Network/publicIPAddresses/write",
      "Microsoft.Network/publicIPAddresses/read",
      "Microsoft.Network/publicIPAddresses/join/action",
      "Microsoft.Network/virtualNetworks/subnets/join/action",
      "Microsoft.Network/networkSecurityGroups/join/action",
      "Microsoft.Network/loadBalancers/backendAddressPools/join/action",
      "Microsoft.Compute/virtualMachineScaleSets/publicIPAddresses/read",
      "Microsoft.Compute/virtualMachineScaleSets/networkInterfaces/read",
      "Microsoft.Compute/virtualMachineScaleSets/virtualMachines/read",
      "Microsoft.Compute/virtualMachines/instanceView/read",
      "Microsoft.Compute/virtualMachineScaleSets/virtualMachines/instanceView/
read",
      "Microsoft.Compute/virtualMachineScaleSets/instanceView/read",
      "Microsoft.Authorization/roleAssignments/read",
      "Microsoft.Authorization/roleDefinitions/read",
      "Microsoft.Compute/virtualMachines/read",
      "Microsoft.ManagedIdentity/userAssignedIdentities/assign/action"
    ],
    "notActions":[
    ],
    "dataActions":[
    ],
    "notDataActions":[
    ]
  }
]
}
}
}

```

The following table describes the permissions:

| Permission | Description |
|--|--|
| Microsoft.Resources/subscriptions/resourcegroups/read | Required. Checks if the cluster resource group exists. |
| Microsoft.Resources/subscriptions/resourcegroups/write Microsoft.Resources/subscriptions/resourcegroups/delete | Required when the cluster resource group is not specified in the advanced configuration. If the cluster resource group is not specified in the advanced configuration, then the Secure Agent creates a new resource group in a subscription named <cluster-instance-id>-rg. |
| Microsoft.Storage/storageAccounts/read Microsoft.Storage/storageAccounts/write Microsoft.Storage/storageAccounts/listKeys/action | Required. Lists storage account keys and performs storage operations. These actions assume that the staging storage account is within the cluster resource group. |
| Microsoft.Compute/virtualMachineScaleSets/delete Microsoft.Compute/virtualMachineScaleSets/write Microsoft.Compute/virtualMachineScaleSets/read | Required. Discovers and manages virtual machine scale sets (VMSS) for master and worker nodes. |
| Microsoft.Network/loadBalancers/delete Microsoft.Network/loadBalancers/write Microsoft.Network/loadBalancers/read | Required. Discovers and manages load-balancer used for API-server endpoint. |
| Microsoft.Network/networkSecurityGroups/delete Microsoft.Network/networkSecurityGroups/write Microsoft.Network/networkSecurityGroups/read | Required. Discovers and manages network security groups created for master and worker nodes. If the network security group (NSG) is attached to a subnet, these permissions override rules specified in the subnet. |
| Microsoft.Network/virtualNetworks/read | Required. Discovers the VNet for a cluster. |
| Microsoft.Network/virtualNetworks/delete Microsoft.Network/virtualNetworks/write | Required when a VNet is not specified in the cluster asset. |
| Microsoft.Network/publicIPAddresses/delete Microsoft.Network/publicIPAddresses/write Microsoft.Network/publicIPAddresses/read Microsoft.Network/publicIPAddresses/join/action | Required. Discovers and manages the public IP address associated with the cluster end-point. The join action is required to let the load-balancer use this public IP address. |
| Microsoft.Network/virtualNetworks/subnets/join/action | Required. Allows master and worker nodes to join a specific subnet. This permission is required for any form of VNet setting. If you use an existing VNet, the scope for this permission must include the resource group that holds the VNet. |
| Microsoft.Network/virtualNetworks/subnets/read | Required if you use an existing VNet. The scope for this permission must include the resource group that holds the VNet. |
| Microsoft.Network/networkSecurityGroups/join/action | Required. Allows the master and worker nodes to attach a pre-created network security group (NSG). |

| Permission | Description |
|---|--|
| Microsoft.Network/loadBalancers/backendAddressPools/join/action | Required. Allows the master and worker nodes to be added to a cluster end-point. Master nodes are added to the cluster end-point during cluster provisioning. |
| Microsoft.Compute/virtualMachineScaleSets/publicIPAddresses/read Microsoft.Compute/virtualMachineScaleSets/networkInterfaces/read | Required. Used by the Secure Agent to get the IP addresses assigned to the master and worker nodes. The Secure Agent uses these permissions to connect to master nodes using SSH and download the kubeconfig file for a given cluster. |
| Microsoft.Compute/virtualMachineScaleSets/virtualMachines/read Microsoft.Compute/virtualMachines/instanceView/read Microsoft.Compute/virtualMachineScaleSets/virtualMachines/instanceView/read Microsoft.Compute/virtualMachineScaleSets/instanceView/read | Required. Checks the master and worker node status. |
| Microsoft.Compute/virtualMachineScaleSets/manualupgrade/action | Required when you use the initialization script. Also required to manually update the master and worker nodes to apply a script extension. |
| Microsoft.Authorization/roleAssignments/read Microsoft.Authorization/roleDefinitions/read | Required. Validates the advanced configuration. |
| Microsoft.Compute/virtualMachines/read Microsoft.ManagedIdentity/userAssignedIdentities/assign/action | Required when you use managed identity authentication to connect to a source or target. The Secure Agent uses these permissions to detect the managed identity of the agent and assign the identity to the virtual machine scale sets. |

Add role assignments

Add role assignments to assign the agent role to the managed identity. Then, assign the managed identity to the Secure Agent machine.

Complete the following tasks:

1. Assign the custom role `agent_role` to the managed identity named `agent_identity`.
2. Assign the managed identity `agent_identity` to the machine where the Secure Agent is installed.

Step 8. Create a service principal for the cluster

Create a service principal to perform cluster operations on an advanced cluster. You will use this service principal to populate the advanced configuration.

In Azure, complete the following tasks:

1. Create a service principal.
2. Create a cluster role.

3. Add a role assignment to assign the cluster role to the service principal.
4. Store the service principal credentials in a key vault.
5. Add an access policy to the key vault.

Create a service principal

Create a service principal named `cluster_principal`.

For instructions about creating a service principal, refer to the Microsoft Azure documentation.

Create a cluster role

Create a cluster role to define the permissions for the service principal `cluster_principal`.

Create a custom role named `cluster_role` with the following role definition:

```
{
  "properties":{
    "roleName":"cluster_role",
    "description":"",
    "assignableScopes":[
      "/subscriptions/<subscription ID>/resourceGroups/<cluster_resource_group>",
      "/subscriptions/<subscription ID>/resourceGroups/<storage_resource_group>",
      "/subscriptions/<subscription ID>/resourceGroups/<vnet_resource_group>",
      "/subscriptions/<subscription ID>/resourceGroups/
<managed_identity_resource_group>"
    ],
    "permissions":[
      {
        "actions":[
          "Microsoft.Compute/virtualMachineScaleSets/virtualMachines/read",
          "Microsoft.Compute/virtualMachineScaleSets/read",
          "Microsoft.Compute/virtualMachineScaleSets/delete/action",
          "Microsoft.Compute/virtualMachines/instanceView/read",
          "Microsoft.Compute/virtualMachineScaleSets/virtualMachines/instanceView/
read",
          "Microsoft.Compute/virtualMachineScaleSets/instanceView/read",
          "Microsoft.Compute/virtualMachineScaleSets/write",
          "Microsoft.Network/networkSecurityGroups/join/action",
          "Microsoft.Network/virtualNetworks/subnets/join/action",
          "Microsoft.Network/loadBalancers/backendAddressPools/join/action",
          "Microsoft.Compute/disks/delete",
          "Microsoft.Compute/disks/write",
          "Microsoft.Compute/disks/read",
          "Microsoft.Compute/virtualMachineScaleSets/virtualMachines/write",
          "Microsoft.ManagedIdentity/userAssignedIdentities/assign/action"
        ],
        "notActions":[],
        "dataActions":[],
        "notDataActions":[]
      }
    ]
  }
}
```

The following table describes the permissions:

| Permission | Description |
|---|---|
| Microsoft.Compute/virtualMachineScaleSets/virtualMachines/read Microsoft.Compute/virtualMachineScaleSets/write Microsoft.Network/loadBalancers/backendAddressPools/join/action Microsoft.Network/networkSecurityGroups/join/action | Required. Used by the Secure Agent to discover cluster resources. |
| Microsoft.Network/virtualNetworks/subnets/join/action | Required. Used by the Secure Agent to discover cluster resources. If you use an existing VNet, the scope for this permission must include the resource group that holds the VNet. |
| Microsoft.Network/virtualNetworks/subnets/read | Required if you use an existing VNet. The scope for this permission must include the resource group that holds the VNet. |
| Microsoft.Compute/virtualMachineScaleSets/read Microsoft.Compute/virtualMachines/instanceView/read Microsoft.Compute/virtualMachineScaleSets/virtualMachines/instanceView/read Microsoft.Compute/virtualMachineScaleSets/instanceView/read | Required. Used by the Secure Agent to discover master and worker nodes running in Azure. |
| Microsoft.Network/virtualNetworks/subnets/join/action Microsoft.Compute/virtualMachineScaleSets/write Microsoft.Network/networkSecurityGroups/join/action | Required when the cluster auto-scales to add a worker node to the cluster. |
| Microsoft.Compute/disks/write Microsoft.Compute/disks/read Microsoft.Compute/disks/delete | Required when storage auto-scales. These permissions manage disks on Azure. |
| Microsoft.Compute/virtualMachineScaleSets/virtualmachines/write | Required when the storage and cluster auto-scales. These permissions attach Azure disks to worker nodes. |
| Microsoft.Network/virtualNetworks/subnets/join/action | Required when the storage and cluster auto-scale. |
| Microsoft.Network/networkSecurityGroups/join/action | Required when the storage and cluster auto-scale. The Secure Agent uses this permission to update the metadata attached to master and worker nodes. |
| Microsoft.ManagedIdentity/userAssignedIdentities/assign/action | Required when you use managed identity authentication to connect to a source or target. The service principal uses this permission to assign managed identities to virtual machines in the virtual machine scale sets. |

Add a role assignment

Add a role assignment to assign the custom role `cluster_role` to the service principal `cluster_principal`.

Store the credentials in a key vault

Create a new key vault and generate a secret to store the credentials for the service principal `cluster_principal`.

Add an access policy to the key vault

Add an access policy to the key vault that allows the managed identity `agent_identity` to access the credentials for the service principal `cluster_principal`.

1. Add an access policy to the key vault.
2. In the access policy, select the secret that you generated for the service principal `cluster_principal`.
3. Grant the secret permission to the managed identity `agent_identity`.

Step 9. Create a managed identity to access sources and targets (optional)

To use managed identity authentication when you connect to a source or target, create a user-assigned managed identity that grants access to the data.

1. Create a managed identity named `<data source>_access_identity`.
2. Assign the Azure built-in role Storage Blob Data Contributor to `<data source>_access_identity`, and set the scope of the access to the storage account, resource group, or resource that contains your data.
3. Assign `<data source>_access_identity` to the Secure Agent machine.
4. In the resource group that contains your data, allow the Secure Agent managed identity and the cluster service principal to access the data. Assign the built-in role Managed Identity Operator to `agent_identity` and `cluster_principal`.

Alternatively, to limit the permissions given to the managed identities, you can create a custom role rather than using Managed Identity Operator. Assign the following permissions to the custom role:

```
"Microsoft.ManagedIdentity/userAssignedIdentities/*/read",  
"Microsoft.ManagedIdentity/userAssignedIdentities/*/assign/action",  
"Microsoft.Authorization/*/read",  
"Microsoft.Resources/subscriptions/resourceGroups/read"
```

Note: In the connection properties, ensure that you set **Client ID** to the client ID of `<data source>_access_identity`. For more information, see [Connections](#).

Step 10. Configure the JAVA_HOME environment variable (optional)

To run commands such as the `list-clusters.sh` and `delete-clusters.sh` commands, you must configure the `JAVA_HOME` environment variable on the Secure Agent machine.

The Java version on the Secure Agent machine must be compatible with JDK 8.

Step 11. Create a staging connection (optional)

If the Secure Agent machine doesn't have a system-assigned managed identity or a single user-assigned managed identity that can access the cluster staging location, create a staging connection to the staging location so that the advanced cluster can share staging data with the Data Integration Server.

1. In Administrator, open the **Connections** page.
2. Create a connection to Azure Data Lake Storage Gen2.
In the connection properties, enter the storage account name of the location that you created to store staging files in ["Step 5. Create storage accounts for cluster files" on page 84](#).
3. Open the **Advanced Clusters** page.
4. Create an advanced configuration or edit the existing advanced configuration for the cluster.
5. On the **Platform Configuration** tab, configure the staging location to specify the same storage account name that you entered in the connection properties. You can specify a folder path within the bucket.
6. On the **Runtime Properties** tab, add the property `clusterconfig.stagingConnectionName` and set the value to the name of the connection.

CHAPTER 5

Setting up a self-service cluster

After you create a Kubernetes cluster, set up your cloud environment so that the Secure Agent can connect to the cluster and use it as a self-service cluster in your organization.

Complete the following tasks:

1. Complete the prerequisites. Allow the cluster nodes to access necessary domains, verify that you have the necessary privileges, and learn about resource access in the cloud environment.
2. Download and install a Secure Agent. Set up the agent on a virtual machine that meets the minimum resource requirements.
3. Create a user-managed service account. Assign permissions to the service account so the Secure Agent can run mappings on a self-service cluster.

To use a self-service cluster on AWS, perform additional steps, including configuring cluster authentication.

The following YouTube video demonstrates how to create a Kubernetes cluster on Amazon EKS and register it as a self-service cluster in Data Integration:

[Setting up a Self-Service Cluster on Amazon EKS in Cloud Data Integration](#)

Step 1. Complete prerequisites

Before you set up your environment, perform the prerequisite tasks.

Complete the following tasks:

- Create a self-service cluster.
- Allow domains for the self-service cluster.
- Verify that you have the correct privileges in your organization.
- Learn how the Secure Agent and the self-service cluster access resources on your cloud platform.

Create a self-service cluster

Create a Kubernetes cluster in your virtual network and use the generated kubeconfig file that contains the configuration when you populate the Secure Agent.

Ensure that you use a supported Kubernetes version and the cluster meets the minimum resource specifications. For more information, see the

[Product Availability Matrix \(PAM\) for Informatica Intelligent Cloud Services](#) and ["Resource requirements for cluster nodes" on page 136](#).

For optimal cluster networking and performance, use the Calico plug-in with a self-service cluster. For more information, see [Project Calico documentation](#).

Add annotations and tolerations (optional)

You can define annotations to attach metadata to the cluster and tolerations to control the nodes that the cluster runs on.

Annotations

Annotations can add non-identifying metadata to Kubernetes objects. Some examples of annotations are the date the object was last updated, the name of the user who manages the object, phone numbers of persons responsible for the object, or tool information for debugging purposes. Annotations can hold any kind of information that is useful and can provide context about the resource. Annotations usually consist of machine-generated data. The metadata in an annotation can be small or large, structured or unstructured, and can include characters not permitted by labels. Clients such as tools and libraries can retrieve this metadata.

Tolerations

Tolerations are a Kubernetes Pod property that allows the Kubernetes scheduler to schedule Pods with matching taints. A taint is a Kubernetes node property that allows a node to repel a set of Pods. Tolerations are applied to Pods. Taints and tolerations work together to ensure that Pods are not scheduled on inappropriate nodes.

For more information about annotations and tolerations, see the Kubernetes documentation.

After you attach annotations and tolerations to the cluster, ensure that you configure them as key-value pairs in the **Advanced Configuration** tab of the advanced configuration. For more information, see [“Advanced configuration” on page 135](#).

Allow domains for self-service clusters

When you use a self-service cluster, the cluster nodes need access to certain domains to fetch artifacts and to access sources and targets.

Add the following domains to your outbound allowlist:

```
artifacthub.informaticacloud.com
infacloud.jfrog.io
infacloud-ct-cdie-docker.jfrog.io
infacloud-discale-docker-stable.jfrog.io
https://storage.googleapis.com
```

Verify privileges in your organization

Verify that you are assigned the correct privileges for advanced configurations in your organization.

Privileges for advanced configurations provide you varying access levels to the **Advanced Clusters** page in Administrator as well as Monitor.

Make sure that you have the read permissions to view the advanced configurations and monitor the self-service clusters.

Learn about resource access

To process data, the Secure Agent and the self-service cluster access the resources that are part of a job, including resources on the cloud platform, source and target data, and staging and log locations.

The agent and the cluster access resources to perform the following tasks:

- Design a mapping.
- Connect to a self-service cluster.
- Run a job, including a data preview job.
- Poll logs.

Designing a mapping

When you design a mapping, the Secure Agent accesses sources and targets so that you can read and write data. For example, when you add a Source transformation to a mapping, the Secure Agent accesses the source to display the fields that you can use in the rest of the mapping. The Secure Agent also accesses the source when you preview data.

To access a source or target, the Secure Agent uses the connection properties. For example, the Secure Agent might use the user name and password that you provide in the connection properties to access a database.

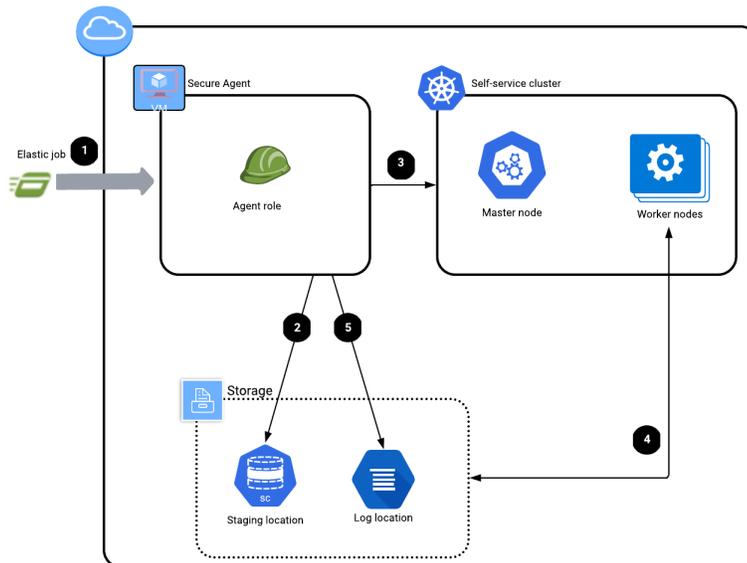
Connecting to a self-service cluster

To connect to a self-service cluster, the Secure Agent stores cluster information in the staging location.

Running a job

To run an advanced job, the Secure Agent and worker nodes access sources and targets, the staging location, and the log location.

The following image shows how resources are accessed when a job runs on a self-service cluster:



The following steps describe the process that the Secure Agent and cluster nodes use to run the job:

1. You run a job.

2. The Secure Agent stores cluster information in the staging location using the permission from the Secure Agent role.
3. The Secure Agent accesses the cluster using the kubeconfig file and submits a job to the self-service cluster.
4. The worker nodes use the access information to the staging and log storage accounts to get job dependencies from the staging location, to stage data in the staging location, and to store logs in the log location.
5. The Secure Agent uses the Secure Agent role to upload the agent job log to the log location.

Polling logs

When you use Monitor, the Secure Agent accesses the log location to poll logs. To poll logs from the log location, the Secure Agent uses the role that is assigned to the Secure Agent machine.

Step 2. Download and install a Secure Agent

Download and install a Secure Agent on a Linux virtual machine. The virtual machine can be an Amazon EC2 instance, Amazon EKS cluster, Azure virtual machine, or Azure Kubernetes Service cluster. This virtual machine is known as the Secure Agent machine.

The following table lists the minimum resource requirements on the Secure Agent machine:

| Component | Minimum requirement |
|---------------|---------------------|
| Cores per CPU | At least four |
| Memory | 16 GB |
| Disk Space | 100 GB |

For more information about installing a Secure Agent, see *Runtime Environments*.

Step 3. Configure a user-managed service account

Create a service account with the cluster role permissions that are required to run a mapping on a self-service cluster.

To configure a user-managed service account, complete the following tasks:

1. Create an optimized cluster role.
2. Create a service account and bind the cluster role.
3. Configure the service account in the advanced configuration.

If you're looking for a quick setup, you can use an Informatica-managed service account. For more information, see ["Use an Informatica-managed service account \(alternative\)" on page 99](#).

Create an optimized cluster role

Create a cluster role that has permission to access and edit cluster resources. A cluster role is global and not associated with a namespace. If you need to restrict permissions to a specific namespace, you can split the cluster role permissions into two different roles.

The permissions required depend on whether the cluster runs mappings with the Spark shuffle service. The Spark shuffle service helps the cluster perform dynamic allocation for Spark jobs. The service is responsible for persisting shuffle files beyond the lifetime of the executors, allowing the number of executors to scale up and down without losing computation.

Minimum permissions to run a mapping with the Spark shuffle service

The following code snippet shows the minimum permissions required to run a mapping with the Spark shuffle service:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: optimized-cluster-role
rules:
- apiGroups: [""]
  resources: ["services","pods","secrets","configmaps"]
  verbs: ["watch","list","get","create","update","patch","delete","deletecollection"]
- apiGroups: ["apps"]
  resources: ["daemonsets","deployments"]
  verbs: ["watch","list","get","create","update","patch","delete","deletecollection"]
- apiGroups: [""]
  resources: ["nodes"]
  verbs: ["watch","list","get","patch"]
- apiGroups: [""]
  resources: ["namespaces","persistentvolumeclaims"]
  verbs: ["watch","list","get"]
```

The permissions in this code snippet applies to all namespaces.

If you need to limit the permissions to certain namespaces only, split these permissions into two roles: Role and ClusterRole. ClusterRole contains permissions for resources that are global, while Role contains permissions for resources that are specific to a namespace.

The following code snippet shows the permissions for Role:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: rbac-informatica-np-admin
  namespace: informatica
rules:
- apiGroups: [""]
  resources: ["services","pods","secrets","configmaps"]
  verbs: ["watch","list","get","create","update","patch","delete","deletecollection"]
- apiGroups: ["apps"]
  resources: ["daemonsets","deployments"]
  verbs: ["watch","list","get","create","update","patch","delete","deletecollection"]
```

The following code snippet shows the permissions for ClusterRole:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: rbac-informatica-global-admin
rules:
- apiGroups: [""]
  resources: ["nodes"]
  verbs: ["watch","list","get","patch"]
- apiGroups: [""]
  resources: ["namespaces","persistentvolumeclaims"]
  verbs: ["watch","list","get"]
```

Minimum permissions to run a mapping without the Spark shuffle service

The following code snippet shows the minimum permissions required to run a mapping without the Spark shuffle service:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: optimized-cluster-role
rules:
- apiGroups: [""]
  resources: ["services","pods","secrets","configmaps"]
  verbs: ["watch","list","get","create","update","patch","delete","deletecollection"]
- apiGroups: [""]
  resources: ["nodes"]
  verbs: ["watch","list","get"]
- apiGroups: [""]
  resources: ["namespaces","persistentvolumeclaims"]
  verbs: ["watch","list","get"]
```

The permissions in this code snippet applies to all namespaces.

If you need to limit the permissions to certain namespaces only, split these permissions into two roles: Role and ClusterRole. ClusterRole contains permissions for resources that are global, while Role contains permissions for resources that are specific to a namespace.

The following code snippet shows the permissions for Role:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: rbac-informatica-np-admin
  namespace: informatica
rules:
- apiGroups: [""]
  resources: ["services","pods","secrets","configmaps"]
  verbs: ["watch","list","get","create","update","patch","delete","deletecollection"]
```

The following code snippet shows the permissions for ClusterRole:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: rbac-informatica-global-admin
rules:
- apiGroups: [""]
  resources: ["nodes"]
  verbs: ["watch","list","get"]
- apiGroups: [""]
  resources: ["namespaces","persistentvolumeclaims"]
  verbs: ["watch","list","get"]
```

Create a service account and bind the cluster role

To grant the permissions defined in the cluster role to the service account, create a service account and a cluster role binding.

Create a service account in the cluster in any namespace, and add the service account token to the kubeconfig file. Then, create a cluster role binding between the service account and the cluster role.

The cluster role binding ensures that the users have the access they need to perform their tasks, while maintaining the security and integrity of the cluster.

Configure the service account in the advanced configuration

Add the service account name to the advanced configuration to allow the Secure Agents to send jobs to the self-service cluster.

On the **Runtime Configuration** tab of the advanced configuration, set the property `infa.k8s.spark.custom.service.account.name` to the service account name. The Spark engine uses the service account for the Spark driver and the Spark shuffle service, if enabled.

Use an Informatica-managed service account (alternative)

If you don't provide the service account name using the property `infa.k8s.spark.custom.service.account.name`, Informatica creates a default service account, cluster role, and cluster role binding.

Informatica creates a service account called `infa-spark` and a cluster role binding called `infa-spark-role` for the Spark driver. This cluster role binding uses the default cluster role `edit` that's available in Kubernetes clusters. The `edit` role lets you perform basic actions like deploying Pods. For more information about the `edit` role, see Kubernetes documentation.

When the Spark shuffle service is enabled, Informatica creates a separate service account, cluster role, and cluster role binding on the cluster. Informatica assigns the following cluster role permissions to the service account:

```
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: spark-shuffle
  labels:
    {{- range $index, $value := .Values.shuffleDsServiceAccountLabels }}
      {{ $index }}: {{ $value }}
    {{- end }}
rules:
- apiGroups: [""]
  resources: ["events","endpoints"]
  verbs: ["create", "patch"]
- apiGroups: [""]
  resources: ["pods/eviction"]
  verbs: ["create"]
- apiGroups: [""]
  resources: ["pods/status"]
  verbs: ["update"]
- apiGroups: [""]
  resources: ["nodes"]
  verbs: ["watch","list","get","update", "patch"]
- apiGroups: [""]
  resources:
  ["pods","services","replicationcontrollers","persistentvolumeclaims","persistentvolumes"]
  verbs: ["watch","list","get"]
- apiGroups: ["apps"]
  resources: ["replicasets","daemonsets"]
  verbs: ["watch","list","get"]
- apiGroups: ["policy"]
  resources: ["poddisruptionbudgets"]
  verbs: ["watch","list"]
- apiGroups: ["apps"]
  resources: ["statefulsets"]
  verbs: ["watch","list","get"]
- apiGroups: ["storage.k8s.io"]
  resources: ["storageclasses"]
  verbs: ["watch","list","get"]
```

Additional configuration for clusters on AWS

To use a self-service cluster on AWS, perform additional configuration steps, including configuring cluster authentication and setting the hop limit.

Configure cluster authentication

When you create a self-service cluster on AWS, you can use the AWS CLI to allow the Secure Agent to authenticate to the cluster. Before you configure cluster authentication, ensure that the AWS CLI is installed on the Secure Agent machine.

Specify the AWS credentials in the kubeconfig file using the AWS CLI. Use the AWS CLI to define the appropriate profile to use. The environment variables that you set in the `exec` flow take precedence over the environment variables that are configured in your environment.

The following sample command demonstrates how to set up `kubectl` to use authentication tokens provided by AWS CLI authentication:

```
users:
- name: arn:aws:eks:ap-southeast-1:543463116864:cluster/cdie-eks-GT3YbtNg
  user:
    exec:
      apiVersion: client.authentication.k8s.io/v1alpha1
      args:
      - --region
      - ap-southeast-1
      - eks
      - get-token
      - --cluster-name
      - cdie-eks-GT3YbtNg
      command: aws
```

You can also authenticate a self-service cluster on AWS using Kubernetes client certificates and service account tokens. For more information about Kubernetes authentication strategies, see the Kubernetes documentation.

Note: In a cluster that uses AWS CLI authentication, a mapping might fail if it runs longer than the duration of the credentials. To avoid this, switch the authentication mechanism to service account token authenticator and run the mapping again.

Configure cluster nodes with IMDSv2

When you configure a self-service cluster on AWS with nodes that use Instance Metadata Service Version 2 (IMDSv2), ensure that the hop limit is 2 on the cluster nodes.

When you create a self-service cluster on Amazon EKS, cluster nodes have a hop limit of 2 by default.

For more information, refer to the AWS documentation.

CHAPTER 6

Setting up a local cluster

Before you create an advanced configuration in your organization, set up your cloud environment so that the Secure Agent can create a local cluster.

Complete the following tasks to set up a local cluster:

1. Complete the prerequisites.
2. Download and install the Secure Agent.
3. Troubleshoot a local cluster.

Prepare for local clusters

Before you deploy a local cluster, complete preparation steps such as adding domains to your allowlist and disabling your OS firewall. A local cluster runs mappings in advanced mode.

Before you deploy a local cluster, be sure you've completed the following preparation steps:

- Add the following domains to the outbound allowlist for your local cluster and ensure that they are reachable from the Secure Agent machine:

```
artifacthub.informaticacloud.com
infacloud.jfrog.io
infacloud-ct-cdie-docker.jfrog.io
infacloud-discale-docker-stable.jfrog.io
rhui3.<region>.<cloud>.ce.redhat.com (RHEL 8.x ONLY)
```

Example of the last domain above: `rhui3.us-west-2.aws.ce.redhat.com`

The local cluster needs to access these domains to fetch artifacts such as machine images, and to access sources and targets.

- Disable the OS firewall by running this command on the Secure Agent machine:

```
sudo systemctl disable firewalld
```

Disabling the OS firewall prevents disruption of jobs sent to the local cluster.

Download and install a Secure Agent

A local cluster requires a Secure Agent. Download and install a Secure Agent on a local Linux machine.

Step 1: Verify software and hardware requirements

Verify that you are running a supported version of Linux. For the list of supported Linux operating systems for the Secure Agent, see the [Product Availability Matrix \(PAM\) for Informatica Intelligent Cloud Services](#) in the Knowledge Base.

Verify that your machine meets the minimum hardware requirements to set up a local cluster:

- 8 vCPU, 32 GB memory
- 100 GB disk space for the root volume
- 20 GB disk space on the root volume for `/var`
- 30 GB disk space on the root volume for `/tmp`

To keep the cluster from hanging when it starts, allocate at least 30 GB of disk space each for `/var` and `/tmp`.

Step 2: Verify NOPASSWD sudo privileges

To run the local cluster on the Secure Agent, the user who starts the Secure Agent requires NOPASSWD sudo privileges on the Secure Agent machine. If it's not possible to grant the NOPASSWD privilege, perform one of the following workarounds:

Use the pmsuid file

1. Copy `pmsuid` from: `<Secure Agent home>/apps/At_Scale_Server/<latest version>/bin/Linux.64/`
to: `<Secure Agent home>/apps/At_Scale_Server/ext/`
2. Change the owner and group of `pmsuid` to root and set the `setgid` bit for the file.
3. Set `ccs.localcluster.deployment.mode=SUID` in the runtime properties section for the local cluster.
4. Use Monitor to stop the local cluster and run a job to start the cluster again.

Update the sudoers file

Edit the `/etc/sudoers` file, and add the following line:

```
<user ID> ALL=(ALL) NOPASSWD: /usr/bin/kubeadm, /usr/bin/tee, /usr/bin/yum, /usr/sbin/modprobe, /usr/sbin/sysctl, /usr/bin/systemctl, /usr/sbin/swapoff, /usr/bin/chown, /usr/bin/cp, /usr/bin/rm
```

Where `<user ID>` is a non-root user who doesn't have sudo privileges on the Secure Agent machine.

Step 3: Download and install the Secure Agent

Download and install a Secure Agent on a local Linux machine. For detailed information about installing a Secure Agent, see "Secure Agent installation on Linux" in *Runtime Environments*.

Troubleshoot a local cluster

Why did the advanced cluster fail to start?

To troubleshoot why the advanced cluster failed to start, examine the `ccs-operation.log` file in the following directory on the Secure Agent machine: `<Secure Agent installation directory>/apps/At_Scale_Server/<version>/ccs_home/`

The `ccs-operation.log` couldn't help me resolve the issue. Where else can I look?

If there isn't enough information in the `ccs-operation.log` file to help you resolve the issue, try reviewing the `cluster-operation.log` file that is dedicated to the instance of the advanced cluster.

When an external command set runs, the `ccs-operation` log displays the path to the cluster-operation logs. For example:

```
2020-06-15 21:22:36.094 [reqid:] [T:000057] INFO      :
c.i.c.s.c.ClusterComputingService [CCS_10400] Starting to run command set
[<command set>] which contains the following commands: [ <commands> ; ].
```

You can find the execution log in the following location:

```
/data2/home/cldagnt/SystemAgent/apps/At_Scale_Server/35.0.1.1/ccs_home/
3xukm9iqp5zeahyrb7rqoz.k8s.local/infra/cluster-operation.log
```

The specified folder contains all cluster-operation logs that belong to the instance of the cluster. You can use the logs to view the full stdout and stderr output streams of the command set.

Why are my jobs not being processed on the local cluster?

The OS firewall might be preventing the worker nodes from sending the jobs to the local cluster for processing.

Check if the firewall is running using this command:

```
sudo firewall-cmd --state
```

If the output of this command mentions "running", then the firewall is active.

Disable the OS firewall daemon with this command:

```
sudo systemctl disable firewalld
```

CHAPTER 7

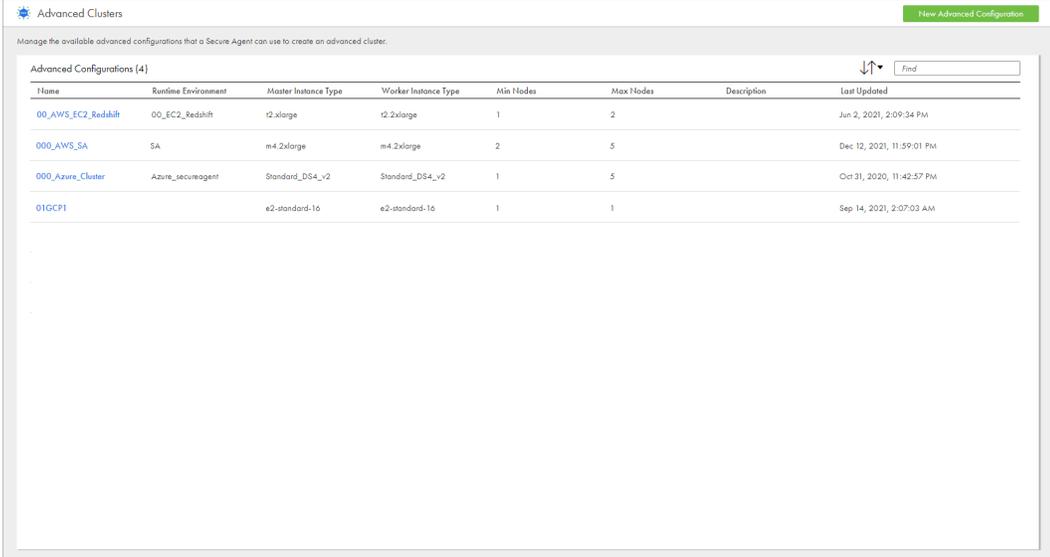
Advanced configurations

An advanced configuration is a set of properties that define the resources that you provision to create an advanced cluster. The properties that you configure in an advanced configuration depend on the cloud platform.

Create an advanced configuration on the **Advanced Clusters** page. When you configure the properties in the advanced configuration, you associate the configuration with a runtime environment. You can create an advanced configuration from an existing configuration. The new configuration uses the existing configuration as a template. You must specify a runtime environment for the new configuration and you can also edit the other properties.

After you create the configuration, use the page to view a summary of the configurations that are available in your organization. The summary includes information that you can quickly reference, such as the node instance types and the minimum and maximum number of nodes that the cluster can have.

The following image shows the **Advanced Clusters** page:



The screenshot shows the 'Advanced Clusters' page with a table of configurations. The table has columns for Name, Runtime Environment, Master Instance Type, Worker Instance Type, Min Nodes, Max Nodes, Description, and Last Updated. There are four rows of data.

| Name | Runtime Environment | Master Instance Type | Worker Instance Type | Min Nodes | Max Nodes | Description | Last Updated |
|---------------------|---------------------|----------------------|----------------------|-----------|-----------|-------------|---------------------------|
| 00_AWS_EC2_Redshift | 00_EC2_Redshift | i2.xlarge | i2.2xlarge | 1 | 2 | | Jun 2, 2021, 2:09:34 PM |
| 000_AWS_SA | SA | m4.2xlarge | m4.2xlarge | 2 | 5 | | Dec 12, 2021, 11:59:01 PM |
| 000_Azure_Cluster | Azure_secureagent | Standard_DS4_v2 | Standard_DS4_v2 | 1 | 5 | | Oct 31, 2020, 11:42:57 PM |
| 01GCP1 | | e2-standard-16 | e2-standard-16 | 1 | 1 | | Sep 14, 2021, 2:07:03 AM |

To run a job using the advanced configuration, use the runtime environment that is associated with the advanced configuration.

If you edit the advanced configuration when the advanced cluster is running, you must stop the cluster for the changes in the configuration to take effect. When you stop the cluster, the cluster is deleted and the running jobs are stopped. The cluster starts up again when you run another job.

You can create a new advanced configuration from an existing configuration. The new configuration uses the existing configuration as a template. You can change any of the fields in the new configuration, but by default you only need to change the **Runtime Environment** field.

You can delete an advanced configuration only when the Secure Agent is running. When you delete a configuration, all provisioned resources are automatically deleted. If the agent is not running and you want to delete the provisioned resources, run the commands to list and delete clusters. For more information about the commands, see [Appendix A, "Command reference" on page 149](#).

CLAIRE-powered configurations

Use a CLAIRE-powered configuration to set cost boundaries for an advanced cluster. CLAIRE, Informatica's AI engine, uses machine learning to configure the cluster to stay within the cost boundaries and to make recommendations to improve cluster performance and reduce infrastructure costs.

You can use a CLAIRE-powered configuration to create an advanced cluster in an AWS environment if CLAIRE recommendations are enabled in your organization.

When you use a CLAIRE-powered configuration, you can optimize the cluster for cost or for performance. Then, you specify the target average cost per hour and maximum cost per hour, and CLAIRE configures the cluster to stay within the cost boundaries.

The following table lists the key differences between a cluster that is optimized for cost and a cluster that is optimized for performance:

| Key area | Optimized for cost | Optimized for performance |
|----------------------|---|---|
| Infrastructure costs | Infrastructure costs are less likely to exceed the average cost per hour. | Infrastructure costs are more likely to exceed the average cost per hour, especially for large workloads. |
| Job priority | High-priority jobs might take longer to be scheduled. | High-priority jobs are more likely to run sooner. |
| Target duration | Jobs are less likely to meet their target durations. | Jobs are more likely to meet their target durations. |

CLAIRE keeps infrastructure costs within the average and maximum cost boundaries that you set by performing the following tasks:

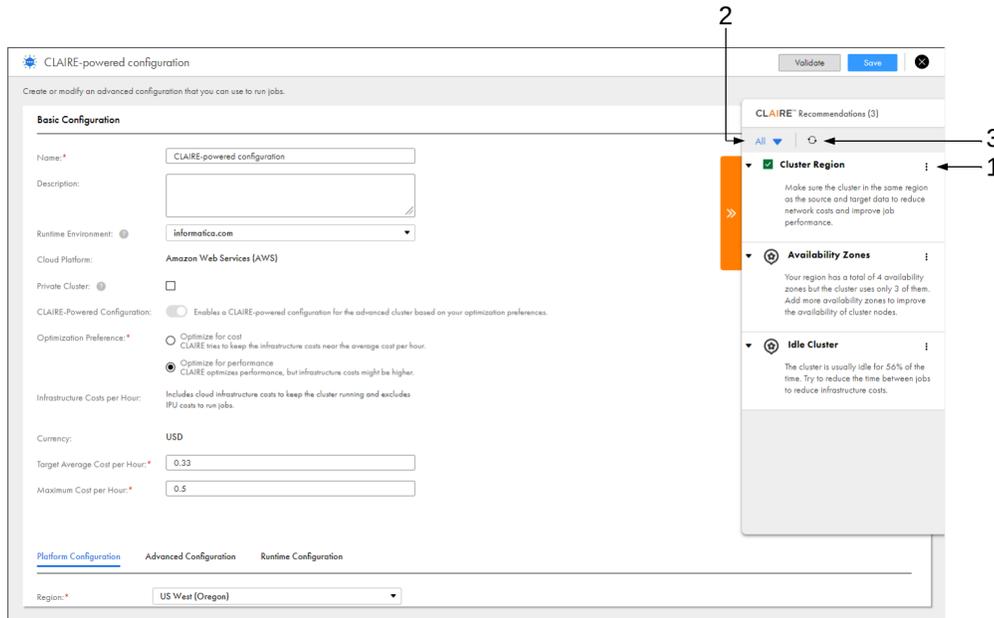
- Using Spot Instances where appropriate
- Selecting instance types based on the workload that the cluster runs
- Scaling the cluster based on the expected workload
- Scheduling jobs to efficiently use cluster resources
- Shutting down the cluster when the cluster is expected to be idle

CLAIRE also makes recommendations to improve cluster performance. You can view the recommendations in the **Recommendations** panel.

CLAIRE recommendations

View CLAIRE recommendations to improve cluster performance and reduce infrastructure costs in the **Recommendations** panel. The **Recommendations** panel is available in CLAIRE-powered configurations.

The following image shows the **Recommendations** panel:



1. Actions menu

Use the **Actions** menu to mark recommendations as complete or incomplete, or to opt in and opt out of recommendations.

CLAIRE automatically applies some recommendations, such as using Spot Instances. You can use the **Actions** menu to opt out of the recommendation or opt back in.

Other recommendations require manual action, such as changing the cluster region. These recommendations appear as to-do items. You can use the **Actions** menu to mark to-do items as complete, or mark them as incomplete. You can also opt out of the recommendation or opt back in.

2. Filter recommendations

Use the **Filter** menu to filter recommendations. You can use the following filters:

- **All** shows all recommendations.
- **To Do** shows recommendations that require manual action.
- **Applied** shows recommendations that are applied automatically and recommendations that are marked complete.
- **Opted-Out** shows recommendations that are opted out of.

3. Refresh recommendations

CLAIRE continues monitoring the cluster and the jobs that it runs, so it might suggest new recommendations over time. You can refresh the recommendations in the **Recommendations** panel, or you can use Monitor to check if any new recommendations are available for the advanced clusters in your organization.

AWS properties

To configure properties in an advanced configuration, click **New Advanced Configuration** or click the name of the configuration that you want to edit on the **Advanced Clusters** page.

Basic configuration

The following table describes the basic properties:

| Property | Description |
|---------------------|---|
| Name | Name of the advanced configuration. |
| Description | Description of the advanced configuration. |
| Runtime Environment | Runtime environment to associate with the advanced configuration. The runtime environment can contain only one Secure Agent. A runtime environment cannot be associated with more than one configuration. If you don't select a runtime environment, the validation process can't validate the communication link to the Secure Agent and that the Secure Agent has the minimum runtime requirements to start a cluster. |
| Cloud Platform | Cloud platform that hosts the cluster. Select Amazon Web Services (AWS). |
| Private Cluster | Creates an advanced cluster in which cluster resources have only private IP addresses. When you choose to create a private cluster, you must specify the VPC and subnet in the advanced properties. |

CLAIRE-powered configuration

Enable a CLAIRE-powered configuration to allow CLAIRE to configure the cluster to stay within cost boundaries and make recommendations to improve cluster performance and to reduce infrastructure costs. You can use a CLAIRE-powered configuration if CLAIRE recommendations are enabled in your organization.

The following table describes the CLAIRE-powered configuration properties:

| Property | Description |
|------------------------------|---|
| Optimization Preference | Cost or performance preference that CLAIRE uses to balance infrastructure costs with cluster performance. |
| Currency | Currency used to set the target average cost per hour and maximum cost per hour, such as USD. |
| Target Average Cost per Hour | Average cost per hour to run the advanced cluster. |
| Maximum Cost per Hour | Maximum cost per hour to run the advanced cluster. |

If you enable a CLAIRE-powered configuration, you configure fewer platform properties.

Platform configuration

The following table describes the platform properties:

| Property | Description |
|---------------------------|---|
| Region | Region in which to create the cluster. Use the drop-down menu to view the regions that you can use. |
| Master Instance Type | <p>Instance type to host the master node. Use the drop-down menu to view the instance types that you can use.</p> <p>The instance types that you can use depend on the availability zones that you select and your AWS account.</p> <p>For information to verify that the instance type that you select from the drop-down menu is supported in the selected availability zones and your AWS account, refer to the AWS documentation.</p> <p>Not applicable in a CLAIRE-powered configuration.</p> |
| Master Instance Profile | <p>Instance profile to be attached to the master node. The name must consist of alphanumeric characters with no spaces. You can also include any of the following characters: <code>_+=, .@-</code></p> <p>If you specify the master instance profile, you must also specify the worker instance profile.</p> |
| Worker Instance Type | <p>Instance type to host the worker nodes. Use the drop-down menu to view the instance types that you can use.</p> <p>The instance types that you can use depend on the availability zones that you select and your AWS account.</p> <p>For information to verify that the instance type that you select from the drop-down menu is supported in the selected availability zones and your AWS account, refer to the AWS documentation.</p> <p>Not applicable in a CLAIRE-powered configuration.</p> |
| Worker Instance Profile | <p>Instance profile to be attached to the worker nodes. The name must consist of alphanumeric characters with no spaces. You can also include any of the following characters: <code>_+=, .@-</code></p> <p>If you specify the worker instance profile, you must also specify the master instance profile.</p> |
| Number of Worker Nodes | <p>Number of worker nodes in the cluster. Specify the minimum and maximum number of worker nodes.</p> <p>Not applicable in a CLAIRE-powered configuration.</p> |
| Enable Spot Instances | <p>Indicates whether to use Spot Instances for worker nodes.</p> <p>Not applicable in a CLAIRE-powered configuration.</p> |
| Spot Instance Price Ratio | <p>Maximum percentage of On-Demand Instance price to pay for Spot Instances. Specify an integer value between 1 and 100.</p> <p>Required if you enable Spot Instances. If you do not enable Spot Instances, this property is ignored.</p> <p>Not applicable in a CLAIRE-powered configuration.</p> |

| Property | Description |
|--------------------------|---|
| Enable High Availability | <p>Indicates whether the cluster is highly available. An odd number of master nodes will be created based on the number of availability zones or subnets that you provide. You must provide at least three availability zones or subnets.</p> <p>For example, if you provide six availability zones, five master nodes are created with each master node in a different availability zone.</p> <p>Note: When you provide multiple availability zones or subnets, worker nodes are highly available. Worker nodes are created across the availability zones or subnets regardless of whether high availability is enabled.</p> <p>For more information about high availability, refer to the Kubernetes documentation.</p> <p>Not applicable in a CLAIRE-powered configuration.</p> |
| Availability Zones | <p>List of AWS availability zones where cluster nodes are created. The master node is created in the first availability zone in the list. If multiple zones are specified, the cluster nodes are created across the specified zones.</p> <p>If you specify availability zones, the zones must be unique and be within the specified region.</p> <p>The availability zones that you can use depend on your AWS account. To check which zones are available for your account, refer to the AWS documentation.</p> <p>Required if you do not specify a VPC. If you specify a VPC, you cannot provide availability zones. You must provide subnets instead of availability zones.</p> |
| EBS Volume Type | <p>Type of Amazon EBS volumes to attach to Amazon EC2 instances as local storage. You can use only EBS General Purpose SSD (gp2).</p> <p>Not applicable in a CLAIRE-powered configuration.</p> |
| EBS Volume Size | <p>Size of the EBS volume to attach to a worker node for temporary storage during data processing. The volume size scales between the minimum and maximum based on job requirements. The range must be between 50 GB and 16 TB.</p> <p>By default, the minimum and maximum volume sizes are 100 GB.</p> <p>This configuration property does not apply to Graviton-enabled clusters, as Graviton does not support storage scaling.</p> <p>Note: When the volume size scales down, the jobs that are currently running on the cluster might take longer to complete.</p> <p>Not applicable in a CLAIRE-powered configuration.</p> |
| Cluster Shutdown | <p>Cluster shutdown method. You can select one of the following cluster shutdown methods:</p> <ul style="list-style-type: none"> - Smart shutdown. The Secure Agent stops the cluster when no job is expected during the defined idle timeout, based on historical data. - Idle timeout. The Secure Agent stops the cluster after the amount of idle time that you define. <p>Not applicable in a CLAIRE-powered configuration.</p> |
| Mapping Task Timeout | <p>Amount of time to wait for a mapping task to complete before it is terminated. By default, a mapping task does not have a timeout.</p> <p>If you specify a timeout, a value of at least 10 minutes is recommended. The timeout begins when the mapping task is submitted to the Secure Agent.</p> |

| Property | Description |
|------------------|---|
| Staging Location | Location on Amazon S3 for staging data. You can use a path that includes the folders in the bucket, such as <code><bucket name>/<folder name></code> . Specify an S3 bucket in the same region as the cluster to improve latency. |
| Log Location | Location on Amazon S3 to store logs that are generated when you run an advanced job. You can use a path that includes the folders in the bucket, such as <code><bucket name>/<folder name></code> . Specify an S3 bucket in the same region as the cluster to improve latency. |

Advanced configuration

The following table describes the advanced properties:

| Property | Description |
|----------------------------|--|
| VPC | Amazon Virtual Private Cloud (VPC) in which to create the cluster. The VPC must be in the specified region. If you choose to not create a private cluster, you do not need to specify a VPC. In this case, the agent creates a VPC on your AWS account based on the region and the availability zones that you select. Note: If you plan to use the Sequence Generator transformation, you must specify a VPC and subnets. |
| Subnets | Subnets in which to create cluster nodes. Use a comma-separated list to specify the subnets. Required if a VPC is specified. Each subnet must be in a different availability zone within the specified VPC. If you do not specify a VPC, you cannot specify subnets. You must provide availability zones instead of subnets. Note: If you plan to use the Sequence Generator transformation, you must specify a VPC and subnets. |
| Initialization Script Path | Amazon S3 file path of the initialization script to run on each cluster node when the node is created. Use the format: <code><bucket name>/<folder name></code> . The script can reference other init scripts in the same folder or in a subfolder. The script must be a bash script. |
| ELB Security Group | Defines the inbound rules between the Kubernetes API server and clients that are external to the advanced cluster. Also defines the outbound rules between the Kubernetes API server and the cluster nodes. This security group attaches to the load balancer that the Secure Agent provisions for the advanced cluster. When you specify a security group, VPC and subnet information are required. For more information about security groups, see "Step 4. Create user-defined security groups for Amazon EC2" on page 22 . |
| Master Security Group ID | Defines the inbound rules between master nodes and worker nodes in the advanced cluster, ELB security group, Secure Agent, and outbound rules to other nodes. This security group attaches to all master nodes of the cluster. When you specify a security group, VPC and subnet information are required. For more information about security groups, see "Step 4. Create user-defined security groups for Amazon EC2" on page 22 . |

| Property | Description |
|--------------------------|---|
| Worker Security Group ID | <p>Defines the inbound and outbound rules between worker nodes in the advanced cluster and other nodes. This security group is attached to all worker nodes of the cluster.</p> <p>When you specify a security group, VPC and subnet information are required.</p> <p>For more information about security groups, see "Step 4. Create user-defined security groups for Amazon EC2" on page 22.</p> |
| AWS Tags | <p>AWS tags to apply to cluster nodes. Each tag has a key and a value. The key can be up to 127 characters long. The value can be up to 256 characters long.</p> <p>You can list a maximum of 30 tags. The Secure Agent also assigns default tags to cloud resources. The default tags do not contribute to the limit of 30 tags.</p> <p>Note: Issues can occur when you override default tags. Do not override the following default tags:</p> <ul style="list-style-type: none"> - Name - KubernetesCluster - k8s.io/cluster-autoscaler/enabled - k8s.io/cluster-autoscaler/<cluster instance ID>.k8s.local <p>The key cannot start with "aws:" because AWS reserves this phrase for their use.</p> <p>Tags cannot include UTF-8 characters \u241e and \u241f that correspond to record and unit separators represented by ASCII control characters 30 and 31.</p> |

Runtime configuration

The following table describes the runtime properties:

| Property | Description |
|--------------------|---|
| Encrypt Data | <p>Indicates whether temporary data on the cluster is encrypted.</p> <p>Note: Encrypting temporary data might slow down job performance.</p> |
| Runtime Properties | Custom properties to customize the cluster and the jobs that run on the cluster. |

Validating the configuration

You can validate the information needed to create or update an advanced configuration before you save the configuration properties.

The validation process performs the following validations:

- You have provided the necessary information on the configuration page.
- The information you provided is valid or in the correct format. For example, if the runtime environment you specified is associated with another cluster configuration.
- The information you provided is valid or in the correct format. For example, if the runtime environment you specified is associated with another cluster configuration.

If you encounter errors related to context keys when you validate an advanced configuration, then add key `ccs.k8s.policy.context.key` to the runtime property in the advanced configuration. You can use the following value structure to add the context keys:

```
"ContextKeyName-'keyName1',ContextKeyValues-'keyValue1',ContextKeyType-(string|stringList|numeric|numericList|boolean|booleanList|ip|ipList|binary|binaryList|date|dateList)&infaContextKeyName-'keyName2',ContextKeyValues-'keyValue2',ContextKeyType-(string|stringList|numeric|numericList|boolean|booleanList|ip|ipList|binary|binaryList|date|dateList)"
```

For example:

```
ccs.k8s.policy.context.key=ContextKeyName-'aws:username',ContextKeyValues-'kops',ContextKeyType-string&infaContextKeyName-'ec2:ResourceTag/CREATED_BY',ContextKeyValues-'SFA-TDS',ContextKeyType-string
```

For more information about context keys, contact Informatica Global Customer Support.

GPU worker instance type

When you configure the worker instance type for the advanced configuration, you can select a GPU-enabled instance type. Selecting a GPU-enabled instance type creates a GPU-enabled cluster. GPUs use a massive parallel architecture to accelerate concurrent processing, offering performance benefits in many cases.

You can select a worker instance type in the g4 and p3 instance families. For more information about these instance types, refer to AWS documentation.

If your organization uses an outgoing proxy server, allow traffic from the Secure Agent machine to the following domains:

```
.docker.io
.docker.com
.nvidia.com
.nvidia.github.io
```

When you create a GPU-enabled cluster, the Spark executors each use one GPU and four Spark executor cores by default. You can change the number of Spark executor cores using the Spark session property `spark.executor.cores`.

All mappings submitted to the cluster that can run on GPU will run on GPU. Spark tasks in the mapping that cannot run on GPU run on CPU instead. To see which Spark jobs run on GPU and which jobs run on CPU, check the Spark event log after the job completes.

Note: The output of a task that runs on GPU might be different than the output if the task ran on CPU. For example, floating-point values might be rounded differently. For more information about processing differences, refer to Spark RAPIDS documentation.

For rules and guidelines for mappings that run on GPU-enabled clusters, see the Data Integration help.

Graviton worker instance type

You can select an AWS Graviton 2 as a worker instance type to run mappings. Graviton is a CPU-based instance type that uses Advanced RISC Machines (ARM) Neoverse N1 cores to deliver computational technology.

You can select one of the following worker instance types:

- T4g
- M6g
- M6gd
- C6g
- C6gd
- C6gn
- R6g
- R6gd

For more information about these instance types, refer to AWS documentation.

Graviton guidelines and limitations

The following guidelines and limitations apply to Graviton worker instance types:

- Graviton worker instance types do not support some expression functions, for example numeric functions such as `rand` and special functions such as `is_date`.
- EBS volume size configuration on the advanced configuration page does not apply to Graviton worker instance types, as Graviton does not support storage scaling.
- You cannot use the Java transformation or Python transformation with a Graviton worker instance type.
- You cannot run mappings that contain flat files with escape characters, multiple column delimiters, multi character quotation mark, line breakers except `\n` and initial row skipped set to more than one.
- You cannot use a parquet source with snappy compression with a Graviton worker instance type.
- Depending on the complexity of your mapping, you may encounter some libs incompatibility error. You can confirm the root cause by checking the spark driver logs and search for `java.lang.UnsatisfiedLinkError`.

Spot Instances

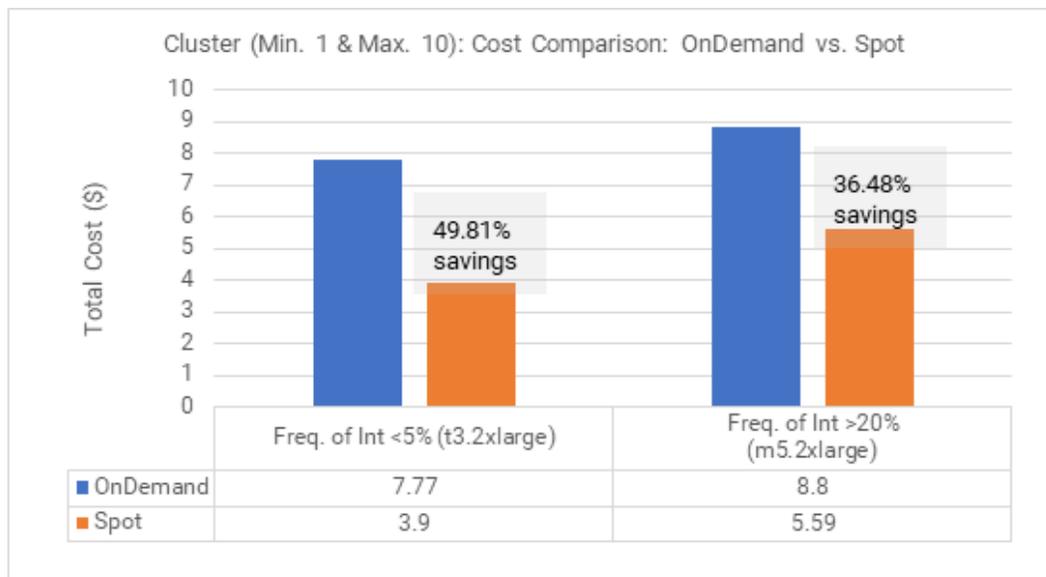
You can configure an advanced cluster to use Spot Instances to host worker nodes.

Spot Instances are spare compute capacity that cloud providers offer at a lower price than On-Demand Instances. This can result in significant cost savings when performing internal tests and debugging in development or QA environments. The performance of On-Demand and Spot Instances of the same instance type is similar.

Note: Spot Instances are not always available, and your cloud provider can interrupt running Spot Instances to reclaim the capacity. Therefore, you shouldn't use Spot Instances on strict SLA-bound jobs.

Spot Instances are most beneficial when the frequency of interruptions is under 5%. Use the [Spot Instance advisor](#) on AWS to see a list of instances with different levels of interruptions.

The following chart shows the potential savings between On-Demand and Spot Instances. The chart also shows the differences in savings with different levels of frequency of interruptions:



In the chart, you can see that when the frequency of interruption is below 5%, Spot Instances can save you nearly 50% on the total cost compared to On-Demand Instances. However, when the frequency of interruption exceeds 20%, your savings drops to 36%.

When you use Spot Instances, you set a Spot Instance price ratio. The Spot Instance price ratio is the maximum price you will pay for Spot Instances as a percentage of the On-Demand Instance price. For example, if On-Demand Instances cost \$0.68 an hour and you set the Spot Instance price ratio to 50, you will pay the current Spot Instance price as long as the price is \$0.34 an hour or less.

The Secure Agent always creates a number of On-Demand worker nodes equal to the minimum number of worker nodes that you configure. When you enable Spot Instances and the cluster scales up, the agent tries to create additional worker nodes on Spot Instances up to the maximum number of worker nodes. If Spot Instances are not available or cost more than the maximum price you set, the cluster uses On-Demand Instances for the worker nodes.

For example, if you set the minimum number of worker nodes to 5 and the maximum to 8, the agent creates 5 nodes on On-Demand Instances and tries to create 3 nodes on Spot Instances. If you set the maximum number of worker nodes equal to the minimum, the cluster uses only On-Demand Instances.

If your cloud provider interrupts a Spot node that is running an advanced job, the agent uses On-Demand nodes to complete the job.

High availability

An advanced cluster can become highly available to eliminate a single point of failure when the master node goes down. If you enable high availability and one master node goes down, other master nodes will be available and jobs on the cluster can continue running.

When a cluster is highly available, watch out for job failures in the following scenarios:

- If all master nodes go down, jobs will fail.
- If too many master nodes go down, the Kubernetes API server becomes unavailable. The threshold for the number of failures is $(n+1)/2$ where n is the number of master nodes. For example, if the cluster has 3 master nodes and 2 master nodes go down, the Kubernetes API server becomes unavailable and jobs fail on the cluster.

Accessing a new staging location

If you plan to use a new staging location, you must first change the staging location in the advanced configuration and then change the permissions to the staging location on AWS.

If you use role-based security, you must also change the permissions to the staging location on the Secure Agent machine.

If you change the permissions before changing the staging location in the configuration, advanced jobs fail with the following error:

```
Error while executing mapping. ExecutionId '<execution ID>'. Cause: [Failed to start cluster for [01000D25000000000005]. Error reported while starting cluster [Cannot apply cluster operation START because the cluster is in an error state.]].
```

To fix the error, perform the following tasks:

1. Revert the changes to the permissions for the staging location.
2. Edit the advanced configuration to revert the staging location.
3. Stop the cluster when you save the configuration.
4. Update the staging location in the configuration, and then change the permissions to the staging location on AWS.

Propagating tags to cloud resources

The Secure Agent propagates tags to cloud resources based on the AWS tags that you specify in an advanced configuration.

The agent propagates tags to the following resources:

- Auto Scaling group
- EBS volume
- EC2 instance
- IAM role*
- Launch template
- Load balancer*
- Public key
- Security group
- Subnet
- VPC

**If the key or value of a tag contains special characters, the agent does not propagate the tag to this resource.*

Note: The Secure Agent propagates tags only to cloud resources that the agent creates. If you create a VPC and subnets and specify the resources in an advanced configuration, the agent does not propagate AWS tags to the VPC and subnets.

If your enterprise follows a tagging policy, make sure to manually assign tags to the following resources:

- Internet gateway
- Network ACL
- Route table

Default tags for cloud resources

In addition to the cloud platform tags that you specify in an advanced configuration, the Secure Agent assigns several default tags to resources. The default tags assist the cluster operator, services on the cloud platform, and data governance. Do not override the default tags.

The following table describes tags that the agent also assigns to cluster nodes to report information about the cluster:

| Cloud platform tag | Description |
|----------------------------|---|
| infa:ccs:hostname | The host name of the Secure Agent machine that started the cluster. If the Secure Agent machine stops unexpectedly and the Secure Agent restarts on a different machine, the host name is the original Secure Agent machine. |
| infa:k8scluster:configname | Name of the advanced configuration that is used to create the cluster. |
| infa:k8scluster:workdir | Staging directory that the cluster uses. |

Some default tags do not have a namespace and can conflict with the user-defined tags that you specify in an advanced configuration. For example, the cluster operator automatically adds the Name and KubernetesCluster tags to all resources, but the tags do not have a namespace. If you specify a user-defined

tag with the same name, such as `KubernetesCluster`, the cluster operator overrides the user-defined tag with the default tag.

Note: Issues can occur when you override default tags. Do not override the following default tags:

- Name
- `KubernetesCluster`
- `k8s.io/cluster-autoscaler/enabled`
- `k8s.io/cluster-autoscaler/<cluster instance ID>.k8s.local`

Data encryption

Encryption protects the data that is used to process jobs. You can use encryption to protect data at rest, temporary data, and data in transit.

Encryption is available for the following types of data:

Data at rest

You can use the server-side encryption options on Amazon S3 to encrypt the following data at rest:

- Staging data on Amazon S3
- Log files on Amazon S3

For more information about encrypting staging data and log files, see [“Encrypt staging data and log files at rest \(optional\)” on page 42](#).

For information about encrypting source and target data, see the help for the appropriate connector in the Data Integration help.

Note: If you configure an encryption-related custom property in an Amazon S3 V2 connection, the Spark engine uses the same custom property to read and write staging data.

Temporary data

Temporary data includes cache data and shuffle data that cluster nodes generate.

To encrypt temporary data, enable encryption in the advanced configuration. If you enable encryption, temporary data is encrypted using the HMAC-SHA1 algorithm by default. To use a different algorithm, contact Informatica Global Customer Support.

Data in transit

By default, data in transit to and from Amazon S3, including staging data and log files, is encrypted using the Transport Layer Security (TLS) protocol.

Google Cloud properties

To configure properties in an advanced configuration, click **New Advanced Configuration** or click the name of the configuration that you want to edit on the **Advanced Clusters** page.

The basic properties describe the advanced configuration and define the cloud platform to host the advanced cluster. To configure the cluster, configure the platform, advanced, and runtime properties.

Basic configuration

The following table describes the basic properties:

| Property | Description |
|---------------------|---|
| Name | Name of the advanced configuration. |
| Description | Description of the advanced configuration. |
| Runtime Environment | Runtime environment to associate with the advanced configuration. The runtime environment can contain only one Secure Agent. A runtime environment cannot be associated with more than one configuration. |
| Cloud Platform | Cloud platform that hosts the cluster. Select Google Cloud Platform (GCP). |
| Private Cluster | Creates an advanced cluster in which cluster resources have only private IP addresses. When you choose to create a private cluster, you must specify the VPC and subnet in the advanced properties. The Secure Agent must be in the same VPC network or a VPC network that can connect to the VPC that you specify in the advanced properties. |

Platform configuration

The following table describes the platform properties:

| Property | Description |
|------------------------|---|
| Region | Region in which to create the cluster. Use the drop-down menu to view the regions that you can use. |
| Master Instance Type | Instance type to host the master node. Use the drop-down menu to view the instance types that you can use. |
| Master Service Account | Service account to attach to the master node. |
| Worker Instance Type | Instance type to host the worker nodes. Use the drop-down menu to view the instance types that you can use. |
| Number of Worker Nodes | Number of worker nodes in the cluster. Specify the minimum and maximum number of worker nodes. |
| Worker Service Account | Service account to attach to the worker nodes. |
| Availability Zones | List of availability zones where cluster nodes are created. The master node is created in the first availability zone in the list. If multiple zones are specified, the cluster nodes are created across the specified zones. The zones must be unique and be within the specified region. |
| Disk Size | Size of the persistent disk to attach to a worker node for temporary storage during data processing. The disk size must be between 50 GB and 16 TB. |

| Property | Description |
|----------------------|---|
| Cluster Shutdown | Cluster shutdown method. You can select one of the following cluster shutdown methods: <ul style="list-style-type: none"> - Smart shutdown. The Secure Agent stops the cluster when no job is expected during the defined idle timeout, based on historical data. - Idle timeout. The Secure Agent stops the cluster after the amount of idle time that you define. |
| Mapping Task Timeout | Amount of time to wait for a mapping task to complete before it is terminated. By default, a mapping task does not have a timeout. If you specify a timeout, a value of at least 10 minutes is recommended. The timeout begins when the mapping task is submitted to the Secure Agent. |
| Staging Location | Location on Google Cloud Storage for staging data. The location name must start with <code>gs://</code> . |
| Log Location | Location on Google Cloud Storage to store logs that are generated when you run an advanced job. The location name must start with <code>gs://</code> . |

Advanced configuration

The following table describes the advanced properties:

| Property | Description |
|----------------------------|--|
| VPC | Google Cloud Virtual Private Cloud (VPC) in which to create the cluster. If you choose to not create a private cluster, you do not need to specify a VPC. In this case, the agent creates a VPC on your Google Cloud account based on the region and the availability zones that you select. |
| Subnet | Subnets in which to create cluster nodes. Use a comma-separated list to specify the subnets. Required if a VPC is specified. Each subnet must be in a different availability zone within the specified VPC. If you do not specify a VPC, you cannot specify subnets. You must provide availability zones instead of subnets. |
| IP Address Range | CIDR block that specifies the IP address range that the cluster can use. For example: <code>10.0.0.0/24</code> |
| Initialization Script Path | Google Cloud Storage file path of the initialization script to run on each cluster node when the node is created. Use the format: <code><bucket name>/<folder name></code> . The script can reference other init scripts in the same bucket or in a subdirectory. The script must be a bash script. |
| Cluster Labels | Labels to apply to cluster nodes. Each label has a key and a value. The key can be up to 63 characters long. You can list a maximum of 55 labels. The Secure Agent also assigns default labels to cloud resources. The default labels do not contribute to the limit of 55 labels. Labels cannot include UTF-8 characters <code>\u241e</code> and <code>\u241f</code> that correspond to record and unit separators represented by ASCII control characters 30 and 31. |

Runtime configuration

The following table describes the runtime properties:

| Property | Description |
|--------------------|--|
| Encrypt Data | Indicates whether temporary data on the cluster is encrypted. |
| Runtime Properties | Custom properties to customize the cluster and the jobs that run on the cluster. |

Propagating labels to cloud resources

The Secure Agent propagates labels to cloud resources based on the cluster labels that you specify in an advanced configuration.

The agent propagates labels to the following resources:

- Compute Engine instance
- Compute Engine instance template

If your enterprise follows a tagging policy, make sure to manually assign labels to other cloud resources.

Note: The Secure Agent propagates labels only to cloud resources that the agent creates. For example, if you create a network and specify the network in an advanced configuration, the agent does not propagate cluster labels to the network.

Data encryption

Encryption protects the data that is used to process jobs. You can use encryption to protect data at rest, temporary data, and data in transit.

Encryption is available for the following types of data:

Data at rest

By default, Google Cloud Storage encrypts staging data and log files. For more information, refer to the Google Cloud documentation.

For information about encrypting source and target data, see the help for the appropriate connector in the Data Integration help.

Temporary data

Temporary data includes cache data and shuffle data that the Spark engine generates on cluster nodes.

To encrypt temporary data, enable encryption in the advanced configuration. If you enable encryption, temporary data is encrypted using the HMAC-SHA1 algorithm by default. To use a different algorithm, contact Informatica Global Customer Support.

Data in transit

By default, Google Cloud Storage uses the Transport Layer Security (TLS) protocol to encrypt data in transit to and from Google Cloud Storage, including staging data and log files.

Microsoft Azure properties

To configure properties in an advanced configuration, click **New Advanced Configuration** or click the name of the configuration that you want to edit on the **Advanced Clusters** page.

The basic properties describe the advanced configuration and define the cloud platform to host the advanced cluster. To configure the cluster, configure the platform, advanced, and runtime properties.

Basic configuration

The following table describes the basic properties:

| Property | Description |
|---------------------|---|
| Name | Name of the advanced configuration. |
| Description | Description of the advanced configuration. |
| Runtime Environment | Runtime environment to associate with the advanced configuration. The runtime environment can contain only one Secure Agent. A runtime environment cannot be associated with more than one configuration. |
| Cloud Platform | Cloud platform that hosts the cluster. Select Microsoft Azure. |
| Private Cluster | Creates an advanced cluster in which cluster resources have only private IP addresses. When you choose to create a private cluster, you must specify the VNet and subnet in the advanced properties. |

Platform configuration

The following table describes the platform properties:

| Property | Description |
|------------------------|--|
| Region | Region in which to create the cluster. Use the drop-down menu to view the regions that you can use. |
| Master Instance Type | Instance type to host the master node. Use the drop-down menu to view the instance types that you can use. The list of available instance types is filtered based on the minimum number of resources that the cluster requires. |
| Worker Instance Type | Instance type to host the worker nodes. Use the drop-down menu to view the instance types that you can use. The instance types that you can use depend on your Azure account. For information to verify that the instance type that you select from the drop-down menu is supported on your account, refer to the Microsoft Azure documentation. |
| Number of Worker Nodes | Number of worker nodes in the cluster. Specify the minimum and maximum number of worker nodes. |
| Enable Spot Instances | Indicates whether to use Spot Instances for worker nodes. |

| Property | Description |
|---------------------------|---|
| Spot Instance Price Ratio | <p>Maximum percentage of On-Demand Instance price to pay for Spot Instances. Specify an integer value between 1 and 100.</p> <p>Required if you enable Spot Instances. If you do not enable Spot Instances, this property is ignored.</p> |
| Enable High Availability | <p>Indicates whether the cluster is highly available. You can enable high availability only if the region has availability zones 1, 2, and 3. One master node is created in each availability zone.</p> |
| Availability Zones | <p>List of availability zones where cluster nodes are created. The list of availability zones is populated automatically based on the region.</p> <p>If the region has availability zones 1, 2, and 3, worker nodes are created across the zones.</p> |
| Azure Disk Size | <p>Size of the Azure disk to attach to a worker node for temporary storage during data processing. The disk size scales between the minimum and maximum based on job requirements. The range must be between 80 GB and 16 TB.</p> <p>By default, the minimum and maximum disk sizes are 100 GB.</p> <p>Note: When the disk size scales down, the jobs that are currently running on the cluster might take longer to complete.</p> |
| Cluster Shutdown | <p>Cluster shutdown method. You can select one of the following cluster shutdown methods:</p> <ul style="list-style-type: none"> - Smart shutdown. The Secure Agent stops the cluster when no job is expected during the defined idle timeout, based on historical data. - Idle timeout. The Secure Agent stops the cluster after the amount of idle time that you define. |
| Mapping Task Timeout | <p>Amount of time to wait for a mapping task to complete before it is terminated. By default, a mapping task does not have a timeout.</p> <p>If you specify a timeout, a value of at least 10 minutes is recommended. The timeout begins when the mapping task is submitted to the Secure Agent.</p> |
| Resource Group (Storage) | <p>Storage resource group that holds the staging and log storage accounts.</p> <p>The resource group can be a maximum of 90 characters.</p> <p>If you specify an initialization script path, the storage account that holds the init script must be part of the same resource group.</p> |
| Staging Location | <p>Location on Azure Data Lake Storage Gen2 to store staging data that is generated when you run jobs.</p> <p>Use the format: <code>abfs(s)://<file system>@<storage account>.dfs.core.windows.net/<folder path></code></p> <p>If encryption is enabled, specify the ABFSS protocol. Otherwise, specify the ABFS protocol.</p> |
| Log Location | <p>Location on Azure Data Lake Storage Gen2 to store logs that are generated when you run a job.</p> <p>Use the format: <code>abfs(s)://<file system>@<storage account>.dfs.core.windows.net/<folder path></code></p> <p>If encryption is enabled, specify the ABFSS protocol. Otherwise, specify the ABFS protocol.</p> |

Advanced configuration

The following table describes the advanced properties:

| Property | Description |
|-----------------------------|---|
| Resource Group (Cluster) | Cluster resource group that holds cluster resources. If you do not specify a resource group, the agent creates a resource group to populate with cluster resources. The resource group can be a maximum of 90 characters. |
| Service Principal Client ID | Service principal that the agent uses to manage Azure resources. |
| Key Vault | Key vault that stores the service principal credentials. |
| Secret Name | Name of the secret that stores the service principal credentials. |
| VNet | Azure VNet in which to create the cluster. Use the format: <code>resourceGroup/VNet</code> . The VNet must be in the specified region. If you choose not to create a private cluster, you don't need to specify a VNet. In this case, the agent creates a VNet on your Azure account based on the region that you select. A VNet is optional if you're using custom network security groups. |
| Subnet | Required when a VNet is specified. Subnet in which to create cluster nodes. A subnet is optional if you're using custom network security groups. |
| IP Address Range | CIDR block that specifies the IP address range that the cluster can use. The IP address range cannot overlap with the IP addresses of the subnets. For example: <code>10.0.0.0/24</code> An IP address range is optional if you're using custom network security groups. |
| Initialization Script Path | Location on Azure Data Lake Storage Gen2 that stores the initialization script to run on each cluster node when the node is created. Use the format: <code>abfs(s)://<file system>@<storage account>.dfs.core.windows.net/<folder path>/file.sh</code> The script must be a bash script and it can reference other init scripts in the same folder. |
| Master Security Group ID | Security group that defines the inbound and outbound security rules for master nodes in the cluster. The Secure Agent attaches this security group to all master nodes in the cluster. Use the format: <code><resource group name>/<NSG name></code> The master security group can be a maximum of 155 characters. Note: If the advanced configuration includes the cluster resource group, and the NSG (network security group) belongs to the cluster resource group, you can use the network security group name as the value. This security group replaces the default master security group created by Data Integration. For more information, see the How-To article " Create user defined security groups in Azure ". When you specify a master security group, the worker security group is required. |

| Property | Description |
|--------------------------|--|
| Worker Security Group ID | <p>Security group that defines the inbound and outbound security rules for worker nodes in the cluster. The Secure Agent attaches this security group to all worker nodes in the cluster..</p> <p>Use the format: <resource group name>/<NSG name></p> <p>The worker security group can be a maximum of 155 characters.</p> <p>Note: If the advanced configuration includes the cluster resource group, and the NSG (network security group) belongs to the cluster resource group, you can use the network security group name as the value.</p> <p>This security group replaces the default worker security group created by Data Integration. For more information, see the How-To article "Create user defined security groups in Azure".</p> <p>When you specify a worker security group, the master security group is required.</p> |
| Azure Tags | <p>Tags on Microsoft Azure to apply to cluster nodes. Each tag has a key and a value.</p> <p>You can list a maximum of 30 tags. The Secure Agent also assigns default tags to cloud resources. The default tags do not contribute to the limit of 30 tags.</p> <p>Note: Issues can occur when you override default tags. For more information, see "Default tags for cloud resources" on page 126.</p> <p>Tags cannot include UTF-8 characters \u241e and \u241f that correspond to record and unit separators represented by ASCII control characters 30 and 31.</p> |

Runtime configuration

The following table describes the runtime properties:

| Property | Description |
|--------------------|---|
| Encrypt Data | <p>Indicates whether temporary data on the cluster is encrypted.</p> <p>Note: Encrypting temporary data might slow down job performance.</p> |
| Runtime Properties | <p>Custom properties to customize the cluster and the jobs that run on the cluster.</p> |

Validating the configuration

You can validate the information needed to create or update an advanced configuration before you save the configuration properties.

The validation process performs the following validations:

- You have provided the necessary information on the configuration page.
- The information you provided is valid or in the correct format. For example, if the runtime environment you specified is associated with another cluster configuration.
- The information you provided is valid or in the correct format. For example, if the runtime environment you specified is associated with another cluster configuration.

When you use managed identity as a Secure Agent credential, you need to add the key `ccs.azure.k8s.prevalidation.agent.clientid` to the runtime property in the advanced configuration.

Spot Instances

You can configure an advanced cluster to use Spot Instances to host worker nodes.

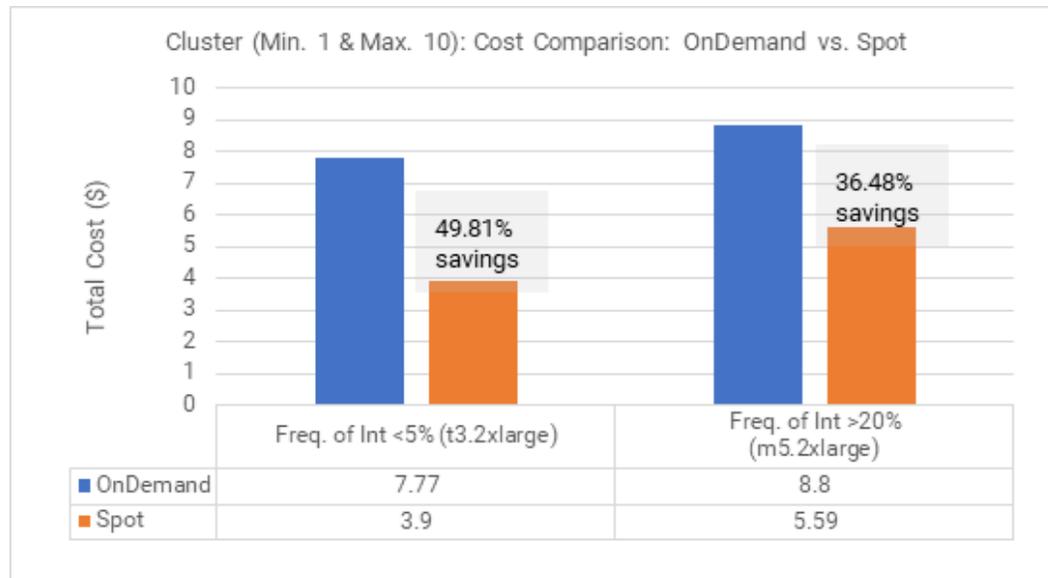
Spot Instances are spare compute capacity that cloud providers offer at a lower price than On-Demand Instances. This can result in significant cost savings when performing internal tests and debugging in

development or QA environments. The performance of On-Demand and Spot Instances of the same instance type is similar.

Note: Spot Instances are not always available, and your cloud provider can interrupt running Spot Instances to reclaim the capacity. Therefore, you shouldn't use Spot Instances on strict SLA-bound jobs.

Spot Instances are most beneficial when the frequency of interruptions is under 5%. Use the [Spot Instance advisor](#) on AWS to see a list of instances with different levels of interruptions.

The following chart shows the potential savings between On-Demand and Spot Instances. The chart also shows the differences in savings with different levels of frequency of interruptions:



In the chart, you can see that when the frequency of interruption is below 5%, Spot Instances can save you nearly 50% on the total cost compared to On-Demand Instances. However, when the frequency of interruption exceeds 20%, your savings drops to 36%.

When you use Spot Instances, you set a Spot Instance price ratio. The Spot Instance price ratio is the maximum price you will pay for Spot Instances as a percentage of the On-Demand Instance price. For example, if On-Demand Instances cost \$0.68 an hour and you set the Spot Instance price ratio to 50, you will pay the current Spot Instance price as long as the price is \$0.34 an hour or less.

The Secure Agent always creates a number of On-Demand worker nodes equal to the minimum number of worker nodes that you configure. When you enable Spot Instances and the cluster scales up, the agent tries to create additional worker nodes on Spot Instances up to the maximum number of worker nodes. If Spot Instances are not available or cost more than the maximum price you set, the cluster uses On-Demand Instances for the worker nodes.

For example, if you set the minimum number of worker nodes to 5 and the maximum to 8, the agent creates 5 nodes on On-Demand Instances and tries to create 3 nodes on Spot Instances. If you set the maximum number of worker nodes equal to the minimum, the cluster uses only On-Demand Instances.

If your cloud provider interrupts a Spot node that is running an advanced job, the agent uses On-Demand nodes to complete the job.

High availability

An advanced cluster can become highly available to eliminate a single point of failure when the master node goes down. If you enable high availability and one master node goes down, other master nodes will be available and jobs on the cluster can continue running.

When a cluster is highly available, watch out for job failures in the following scenarios:

- If all master nodes go down, jobs will fail.
- If too many master nodes go down, the Kubernetes API server becomes unavailable. The threshold for the number of failures is $(n+1) / 2$ where n is the number of master nodes. For example, if the cluster has 3 master nodes and 2 master nodes go down, the Kubernetes API server becomes unavailable and jobs fail on the cluster.

Accessing a new staging location

If you plan to use a new staging location, the Secure Agent must be able to access the location before you update the location in the advanced configuration.

To use the new staging location, complete the following tasks:

1. Update the permissions of the managed identity that is assigned to the Secure Agent machine.
2. Edit the staging location in the advanced configuration.

Propagating tags to cloud resources

The Secure Agent propagates tags to cloud resources based on the Azure tags that you specify in an advanced configuration.

The agent propagates tags to the following resources:

- Azure disk
- Load balancer
- Network security group
- Public IP address
- Resource group
- Virtual machine scale set
- VNet

If your enterprise follows a tagging policy, make sure to manually assign tags to other cloud resources.

Note: The Secure Agent propagates tags only to cloud resources that the agent creates. For example, if you create a VNet and specify the VNet in an advanced configuration, the agent does not propagate Azure tags to the VNet.

Default tags for cloud resources

In addition to the cloud platform tags that you specify in an advanced configuration, the Secure Agent assigns several default tags to cluster resources. Do not override the default tags.

The following table describes tags that the agent assigns to cluster resources:

| Cloud platform tag | Description |
|----------------------------|---|
| infa:ccs:hostname | The host name of the Secure Agent machine that started the cluster. If the Secure Agent machine stops unexpectedly and the Secure Agent restarts on a different machine, the host name is the original Secure Agent machine. |
| infa:k8scluster:configname | Name of the advanced configuration that is used to create the cluster. |
| infa:k8scluster:workdir | Staging directory that the cluster uses. |
| InfalInternalInitDone | Used internally. |
| KubernetesCluster | Identifies an advanced cluster. |

Some default tags do not have a namespace and can conflict with the user-defined tags that you specify in an advanced configuration, such as `KubernetesCluster`. If you specify a user-defined tag with the same name, you might override the tag and issues can occur on the advanced cluster.

Data encryption

Encryption protects the data that is used to process jobs. You can use encryption to protect data at rest, temporary data, and data in transit.

Encryption is available for the following types of data:

Data at rest

By default, Azure encrypts staging data and log files. For more information, refer to the Microsoft Azure documentation.

For information about encrypting source and target data, see the help for the appropriate connector in the Data Integration help.

Temporary data

Temporary data includes cache data and shuffle data that cluster nodes generate.

To encrypt temporary data, enable encryption in the advanced configuration. If you enable encryption, temporary data is encrypted using the HMAC-SHA1 algorithm by default. To use a different algorithm, contact Informatica Global Customer Support.

Data in transit

By default, Azure uses the Transport Layer Security (TLS) protocol to encrypt data in transit to and from cloud storage, including staging data and log files.

When encryption is enabled, you can specify the ABFSS protocol when you configure the staging and log locations in an advanced configuration. If encryption is not enabled, you must use the ABFS protocol.

Local cluster advanced configuration

Additional configuration information for local clusters, intended for advanced users or as directed by Informatica Global Customer Support.

Refer to the following topics as required:

- [“Change staging and log locations \(optional\)” on page 127](#). If you want to move the stage and log locations from the local file system to a cloud location.
- [“Local cluster properties” on page 127](#). Properties to change if you move the staging and log locations to the cloud. Other advanced properties for local clusters.
- [“Configure cloud permissions” on page 130](#). Configure cloud permissions if you change the staging and log locations to the cloud from local storage.
- [“Data encryption” on page 132](#). Learn about how encryption is used to protect data at rest, temporary data, and data in transit.

Change staging and log locations (optional)

When you run jobs on the local cluster, you can choose staging and logging directories on the Secure Agent machine's local file system or on a cloud location. By default, the local cluster uses a local file system path unless you've configured a cloud destination.

To change the staging or log location to a cloud location, complete the following tasks:

1. Refer to the following table to create the location in your cloud environment:

| Cloud environment | Create location |
|-------------------|---|
| AWS | Create the following Amazon S3 locations: <ul style="list-style-type: none">- An S3 location that the cluster uses to store staging files at run time- An S3 location that the cluster uses to store log files for the advanced jobs that run on the cluster |
| Microsoft Azure | Create a storage account using Azure Data Lake Storage Gen2 with locations for staging and log files. Use a hierarchical namespace. |
| Google Cloud | In a Google Cloud environment, create locations for staging and log files on Google Cloud Storage. |

2. Specify the location in the advanced configuration of the advanced cluster. For more information about the format of the staging and log locations, see [“Local cluster properties” on page 127](#)

Local cluster properties

To configure properties in an advanced configuration, click **New Advanced Configuration** or click the name of the configuration that you want to edit on the **Advanced Clusters** page.

The basic properties describe the advanced configuration. To configure the cluster, configure the platform and runtime properties.

Basic configuration

The following table describes the basic properties:

| Property | Description |
|---------------------|---|
| Name | Name of the advanced configuration. |
| Description | Description of the advanced configuration. |
| Runtime Environment | Runtime environment to associate with the advanced configuration. The runtime environment can contain only one Secure Agent. A runtime environment cannot be associated with more than one configuration. |
| Cloud Platform | Cloud platform that hosts the cluster. Select Local. |

Platform configuration

The following table describes the platform properties:

| Property | Description |
|----------------------|---|
| Mapping Task Timeout | <p>Amount of time to wait for a mapping task to complete before it is terminated. By default, a mapping task does not have a timeout.</p> <p>If you specify a timeout, a value of at least 10 minutes is recommended. The timeout begins when the mapping task is submitted to the Secure Agent.</p> |
| Staging Location | <p>Location of the staging data.</p> <p>For staging data on a local file system, specify the location in the following format:</p> <pre>file://<absolute path to the Secure Agent location></pre> <p>For example, to use /home/devbld/staging as the staging location, enter:</p> <pre>file:///home/devbld/staging</pre> <p>Data Integration creates the directory if it does not already exist. Note the extra '/' character from the absolute path.</p> <p>For staging locations on the cloud, specify the path in one of the following formats:</p> <ul style="list-style-type: none">- Amazon S3. <code>s3://<bucket name>/<folder path></code>- Google Cloud Storage. <code>gs://<bucket name>/<folder path>&:<project ID>/<region></code>- Microsoft Azure Data Lake Storage Gen2. <code>abfs(s)://<file system>@<storage account>.dfs.core.windows.net/<folder path>&:<resource group>/<region></code> <p>The region is optional. For a list of valid regions, refer to your cloud provider's documentation.</p> <p>The following examples show how region formats might differ on each cloud platform:</p> <ul style="list-style-type: none">- On AWS, use <code>us-west-2</code> for US West (Oregon).- On Google Cloud, use <code>us-west2</code> for Los Angeles.- On Microsoft Azure, use <code>westus2</code> for West US 2. <p>When the Secure Agent creates a local cluster on Oracle Cloud Infrastructure, the staging location must be on the local file system.</p> |
| Log Location | <p>Location of the logs.</p> <p>For logs on the local file system, specify the location in the following format:</p> <pre>file://<absolute path to the Secure Agent location></pre> <p>For example, to use /home/devbld/logging as the log location, enter:</p> <pre>file:///home/devbld/logging</pre> <p>Data Integration creates the directory if it does not already exist. Note the extra '/' character from the absolute path.</p> <p>For log locations on the cloud, specify the path in the following formats:</p> <ul style="list-style-type: none">- Amazon S3. <code>s3://<bucket name>/<folder path></code>- Google Cloud Storage. <code>gs://<bucket name>/<folder path>&:<project ID>/<region></code>- Microsoft Azure Data Lake Storage Gen2. <code>abfs(s)://<file system>@<storage account>.dfs.core.windows.net/<folder path>&:<resource group>/<region></code> <p>The region is optional. For a list of valid regions, refer to your cloud provider's documentation.</p> <p>The following examples show how region formats might differ on each cloud platform:</p> <ul style="list-style-type: none">- On AWS, use <code>us-west-2</code> for US West (Oregon).- On Google Cloud, use <code>us-west2</code> for Los Angeles.- On Microsoft Azure, use <code>westus2</code> for West US 2. <p>When the Secure Agent creates a local cluster on Oracle Cloud Infrastructure, the log location must be on the local file system.</p> |

Runtime configuration

The following table describes the runtime properties:

| Property | Description |
|--------------------|--|
| Encrypt Data | Indicates whether temporary data on the cluster is encrypted. |
| Runtime Properties | Custom properties to customize the cluster and the jobs that run on the cluster. |

Configure cloud permissions

Local clusters have simplified cloud permissions compared to the standard cloud deployments. Follow the configuration steps that are appropriate for your cloud platform.

Note: You don't need to configure cloud permissions when the staging and log locations are on the local file system (default).

Configure permissions for AWS

In an AWS environment, configure IAM roles for the Secure Agent and cluster operator.

Complete the following steps:

1. In AWS, create an IAM role named `agent_role` and attach it to the Amazon EC2 instance where the Secure Agent is installed. Alternatively, you can designate an existing IAM role to be the Secure Agent role.

Tip: For instructions about creating an IAM role, refer to the AWS documentation. AWS provides several ways to create an IAM role, such as using the AWS Management Console or the AWS CLI.

2. In AWS, create an IAM role for the cluster operator named `cluster_operator_role`.
3. Create the following IAM policy with the name `cluster_operator_policy`:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketLocation",
        "s3:GetEncryptionConfiguration",
        "s3:ListBucket",
        "s3:PutObject",
        "s3:GetObjectAcl",
        "s3:GetObject",
        "s3:DeleteObject",
        "s3:PutObjectAcl"
      ],
      "Resource": [
        "arn:aws:s3:::<cluster-staging-dir1>/**",
        "arn:aws:s3:::<cluster-logging-dir1>/**"
      ]
    }
  ]
}
```

Replace `<cluster-staging-dir1>` and `<cluster-logging-dir1>` with your staging and log locations, respectively. To accommodate S3 locations that change frequently, you can use wildcard characters. For more information, refer to the AWS documentation.

4. Attach the IAM policy `cluster_operator_policy` to the IAM role `cluster_operator_role`.

5. Configure the trust relationship for the cluster operator role to include the Secure Agent role. Because the Secure Agent needs to assume the cluster operator role, the cluster operator role needs to trust the Secure Agent.

Edit the trust relationship of the IAM role `cluster_operator_role` and specify the following IAM policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::{{account-id}}:role/agent_role"
      },
      "Action": "sts:AssumeRole",
    }
  ]
}
```

Note: The value in the Principal element is the ARN of the Secure Agent role.

Optionally, you can configure an external ID to allow only the Secure Agent to assume the cluster operator role.

For example, you can configure the external ID "123" using the following policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::{{account-id}}:role/agent_role"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "sts:ExternalId": "123"
        }
      }
    }
  ]
}
```

Configure permissions for Google Cloud

In a Google Cloud environment, configure a custom IAM role.

Configure an IAM role with the following permissions:

```
storage.buckets.get
storage.objects.create
storage.objects.delete
storage.objects.get
storage.objects.list
storage.objects.update
```

When you create the Google VM, specify a service account that has the required roles associated with it.

Configure permissions for Microsoft Azure

In a Microsoft Azure environment, create a managed identity and a custom role.

Complete the following steps:

1. Disable the firewall on the Secure Agent machine.
2. In Azure, create a managed identity named `agent_identity`. You can use an existing system-assigned managed identity or create a user-assigned managed identity. If you create a user-assigned managed identity, disable the system-assigned managed identity.
For instructions about creating a managed identity, refer to the Microsoft Azure documentation.
3. Create a custom role named `agent_role` with the following role definition:

```
{
  "properties":{
    "roleName":"agent_role",
    "description":"",
    "assignableScopes":[
      "/subscriptions/<subscription ID>/resourceGroups/<storage_resource_group>"
    ],
    "permissions":[
      {
        "actions":[
          "Microsoft.Storage/storageAccounts/read",
          "Microsoft.Storage/storageAccounts/write",
          "Microsoft.Storage/storageAccounts/listKeys/action"
        ],
        "notActions":[

        ],
        "dataActions":[

        ],
        "notDataActions":[

        ]
      }
    ]
  }
}
```

4. Assign the custom role `agent_role` to the managed identity named `agent_identity`.
5. Assign the managed identity `agent_identity` to the VM where the Secure Agent is installed.

Data encryption

Encryption protects the data that is used to process jobs. You can use encryption to protect data at rest, temporary data, and data in transit.

Encryption is available for the following types of data:

Data at rest

By default, each cloud platform encrypts staging and log files. For more information, refer to the cloud provider's documentation.

For information about encrypting source and target data, see the help for the appropriate connector.

Note: If you configure an encryption-related custom property in an Amazon S3 V2 connection, the cluster uses the same custom property to read and write staging data.

Temporary data

Temporary data includes cache data and shuffle data that cluster nodes generate.

To encrypt temporary data, enable encryption in the advanced configuration. If you enable encryption, temporary data is encrypted using the HMAC-SHA1 algorithm by default. To use a different algorithm, contact Informatica Global Customer Support.

Data in transit

By default, cloud providers use the Transport Layer Security (TLS) protocol to encrypt data in transit to and from cloud storage, including staging data and log files.

Note: When encryption is enabled on Microsoft Azure, you can specify the ABFSS protocol when you configure the staging and log locations in an advanced configuration. If encryption is not enabled, you must use the ABFS protocol.

Self-service cluster properties

To configure properties in an advanced configuration, click **New Advanced Configuration** or click the name of the configuration that you want to edit on the **Advanced Clusters** page.

The basic properties describe the advanced configuration and define the cloud platform that hosts the self-service cluster. To configure the cluster, configure the platform and runtime properties.

To learn about the minimum resource specifications that you need to set up a self-service cluster to run a mapping, see ["Resource requirements for cluster nodes" on page 136](#).

Basic configuration

The following table describes the basic properties:

| Property | Description |
|---------------------|---|
| Name | Name of the advanced configuration. |
| Description | Description of the advanced configuration. |
| Runtime Environment | Runtime environment to associate with the advanced configuration. The runtime environment can contain only one Secure Agent. A runtime environment cannot be associated with more than one configuration. If you don't select a runtime environment, the validation process can't validate the communication link to the Secure Agent and that the Secure Agent has the minimum runtime requirements to start a cluster. |
| Cloud Platform | Cloud platform that hosts the cluster. Select Self-Service Cluster. |

Platform configuration

The following table describes the platform properties:

| Property | Description |
|------------------------|--|
| Kubeconfig File Path | <p>Path of the kubeconfig file.</p> <p>A kubeconfig file organizes information about clusters, users, and authentication mechanisms.</p> <p>Example: <code><directory name>/<file_name>.yaml</code></p> <p>You can save the YAML file in any directory on the Secure Agent machine.</p> |
| Kube Context Name | <p>Name of the cluster context.</p> <p>A context defines a named cluster and user tuple which is used to send requests to the specified cluster using the provided authentication information.</p> |
| Cluster Version | <p>Version of the Kubernetes cluster server.</p> <p>The advanced configuration validates the major and minor versions of the Kubernetes cluster server, but does not validate the patch release version numbers.</p> |
| Namespace | <p>Namespace where Informatica deploys resources.</p> |
| Number of Worker Nodes | <p>Number of worker nodes in the cluster. Specify the minimum and maximum number of worker nodes.</p> |
| Cluster Idle Timeout | <p>Amount of time before Informatica-created cluster resource objects are deleted due to inactivity.</p> |
| Mapping Task Timeout | <p>Amount of time to wait for a mapping task to complete before it is terminated. By default, a mapping task does not have a timeout.</p> <p>If you specify a timeout, a value of at least 10 minutes is recommended. The timeout begins when the mapping task is submitted to the Secure Agent.</p> |
| Staging Location | <p>Complete path of the cloud location for staging data.</p> <p>Specify the path in one of the following formats:</p> <ul style="list-style-type: none"> - AWS. <code>s3://<bucket name>/<folder path></code> <p>Note: Specify an S3 bucket in the same region as the cluster to decrease latency.</p> <ul style="list-style-type: none"> - Microsoft Azure. <code>abfs(s)://<file system>@<storage account>.dfs.core.windows.net/<folder path>&:<resource group>/<region></code> <p>The region is optional. The default region is <code>westus2</code>.</p> <p>The Secure Agent needs permissions to access the staging location to store staging files at run time. You must provide appropriate IAM access permissions to both the Secure Agent machine and the worker nodes running in your cluster to access the staging location.</p> |
| Log Location | <p>Complete path of the cloud location for storing logs.</p> <p>Specify the path in one of the following formats:</p> <ul style="list-style-type: none"> - AWS. <code>s3://<bucket name>/<folder path></code> <p>Note: Specify an S3 bucket in the same region as the cluster to decrease latency.</p> <ul style="list-style-type: none"> - Microsoft Azure. <code>abfs(s)://<file system>@<storage account>.dfs.core.windows.net/<folder path>&:<resource group>/<region></code> <p>The region is optional. The default region is <code>westus2</code>.</p> <p>The Secure Agent needs permissions to access the staging location to store staging files at run time. You must provide appropriate IAM access permissions to both the Secure Agent machine and the worker nodes running in your cluster to access the staging location.</p> |

| Property | Description |
|----------------------|--|
| Labels | <p>Key-value pairs that Informatica attaches to the Kubernetes objects that it creates in the self-service cluster.</p> <p>You can use labels to organize and select subsets of objects. Each object can have a set of key-value labels defined. Each key must be unique for a given object.</p> <p>You cannot use the @ symbol in a label. For more information about the supported syntax and character set, see the Kubernetes documentation.</p> |
| Node Selector Labels | Use node selector labels to identify the nodes in the cluster on which Informatica can create Kubernetes objects. |

Advanced configuration

The following table describes the advanced properties:

| Property | Description |
|-------------|--|
| Annotations | <p>Key-value pairs that are used to attach arbitrary non-identifying metadata to objects. You can only define annotations for Pods in a cluster.</p> <p>For more information about annotations, see the Kubernetes documentation.</p> |
| Tolerations | <p>Key-value pairs that are used to ensure that Pods are scheduled on appropriate nodes.</p> <p>When you configure a toleration, set the following properties:</p> <ul style="list-style-type: none"> - Key - Operator - Value - Effect - Toleration Seconds <p>For more information about tolerations, see the Kubernetes documentation.</p> |

Runtime configuration

The following table describes the runtime properties:

| Property | Description |
|--------------------|---|
| Encrypt Data | <p>Indicates whether temporary data on the cluster is encrypted.</p> <p>Note: Encrypting temporary data might slow down job performance.</p> |
| Runtime Properties | Custom properties to customize the cluster and the jobs that run on the cluster. |

Runtime Properties

The following table describes the runtime properties that you can use to customize a self-service cluster and the jobs that run on the cluster:

| Property | Description |
|--|---|
| <code>infa.k8s.deploy.clusterkeygen.enable</code> | Manages clusterkeygen deployment. Set this property to false to disable the deployment. |
| <code>infa.k8s.custom.quota.name</code> | Specifies the quota name if defined in the cluster. |
| <code>ccs.app.control.enable</code> | <p>Manages application submission control. Set this property to false to disable the submission control.</p> <p>You can use the application submission control only if all of the following conditions are true:</p> <ul style="list-style-type: none">- All the cluster nodes are homogeneous.- The namespace or nodes are reserved for Informatica. Informatica can read the node details.- The namespace quota, if present, is honored and the quota name can be configured using custom flag. <p>When you define a quota on the cluster, you can use only the request in quota definition. Don't use limit in the quota definition because if you define the limits for the resources that you deploy, the resources won't get scheduled.</p> |
| <code>infacco.job.spark.kubernetes.scheduler.name</code> | Specifies the custom scheduler name for driver and executor pods. |

Resource requirements for cluster nodes

When you select instance types in an advanced configuration, make sure that the master and worker nodes have enough resources to run advanced jobs successfully.

Master node

The master node is recommended to have at least 8 GB of memory and 4 CPUs.

Note: Because processing on the master node is network-intensive, avoid T instance types in an AWS environment.

Worker nodes

Worker nodes are recommended to have at least 16 GB of memory and 8 CPUs.

The following table lists the default resource requirements for worker nodes:

| Component | Default memory requirement | Default CPU requirement |
|-----------------------|----------------------------|---|
| Kubernetes system | 1 GB per worker node | 0.5 CPU per worker node with an additional 0.5 CPU across the cluster |
| Spark shuffle service | 2 GB per worker node | 1 CPU per worker node |

| Component | Default memory requirement | Default CPU requirement |
|----------------|---------------------------------------|---|
| Spark driver | 4 GB | 0.75 CPU |
| Spark executor | 6 GB, or 3 GB per Spark executor core | 1.5 CPUs, or 0.75 CPU per Spark executor core |

Based on the default resource requirements, a cluster with one worker node requires 13 GB of memory and 4.25 CPUs.

When worker nodes are added to the cluster, each worker node reserves an additional 3 GB of memory and 1.5 CPU for the Kubernetes system and the Spark shuffle service. Therefore, a cluster with two worker nodes requires 16 GB of memory and 5.75 CPUs.

Reconfiguring resource requirements

If you cannot provision enough resources to fulfill the default requirements, you can reconfigure some of the requirements.

You can reconfigure the requirements for the following components:

Spark shuffle service

If you disable the shuffle service, the Spark engine cannot use dynamic allocation. For more details, contact Informatica Global Customer Support.

Spark driver

To reconfigure the amount of memory for the Spark driver, use the Spark session property `spark.driver.memory` in the mapping task. To set the memory in terms of GB, use a value such as 2G. To set the memory in terms of MB, use a value such as 1500m.

For information about reconfiguring the CPU requirement for the Spark driver, contact Informatica Global Customer Support.

Spark executor

To reconfigure the amount of memory for the Spark executor, use the Spark session property `spark.executor.memory` in the mapping task. Similar to the memory value for the Spark driver, you can specify the memory in GB or MB.

You can also change the number of Spark executor cores using the Spark session property `spark.executor.cores`. The default number of cores for GPU-enabled clusters is 4. The default number of cores for all other clusters is 2.

If you edit the number of cores, you change the number of Spark tasks that run concurrently. For example, two Spark tasks can run concurrently inside each Spark executor when you set `spark.executor.cores=2`.

For information about reconfiguring the CPU requirement for Spark executors, contact Informatica Global Customer Support.

Note: If you reduce the memory too low for the Spark driver and Spark executor, these components might encounter an `OutOfMemoryException`.

You cannot edit the resource requirements for the Kubernetes system. The resources are required to maintain a functional Kubernetes system.

For more information about the Spark session properties, see *Tasks* in the Data Integration help.

Resource requirements example

You have an advanced cluster with one worker node. The worker node has 16 GB of memory and 4 CPUs.

If you run an advanced job using the default requirements, the job fails. The Kubernetes system and the Spark shuffle service reserve 3 GB and 2 CPUs, so the cluster has a remaining 13 GB and 2 CPUs to run jobs. The job cannot run because the cluster requires 10 GB of memory and 2.25 CPUs to start the Spark driver and Spark executor.

If you cannot provision a larger instance type, you can reduce the CPU requirement by setting the following advanced session property in the mapping task:

```
spark.executor.cores=1
```

When the number of Spark executor cores is 1, the Spark executor requires only 0.75 CPUs instead of 1.5 CPUs.

If you process a small amount of data, the Spark driver and executor require only a few hundred MB, so you might consider reducing the memory requirements for the driver and executor as well. You can reduce the requirements in the following way:

```
spark.driver.memory=1G
spark.executor.memory=500M
```

After you reconfigure the resource requirements, the cluster must have at least 5 GB of memory and 3.5 CPUs. One worker node with 16 GB and 4 CPUs fulfills the requirements to run the job successfully.

Initialization scripts

Cluster nodes can run an initialization script based on an init script path that you specify in an advanced configuration. Each node runs the script when the node is created, and the script can reference other init scripts.

You might want to run an init script to install additional software on the cluster. For example, your enterprise policy might require each cluster node to contain monitoring and anti-virus software to protect your data.

Consider the following guidelines when you create the init script:

- The init script has privileges to modify anything on the file system, so avoid removing objects from the file system.
- The Secure Agent does not validate the syntax in the init script.

The init script path must be in cloud storage. You can place the scripts in a unique path on the cloud storage system, or you can place the scripts in the staging location.

Initialization script failures

When an initialization script fails on a cluster node, it can have a significant impact on the advanced cluster. An init script failure can prevent the cluster from scaling up or cause the Secure Agent to terminate the cluster.

Note the impact that an init script failure can have in the following situations:

Failure during cluster creation

If the init script fails on any node during cluster creation, the Secure Agent terminates the cluster.

Resolve the issues with the init script before running a job to start the cluster again.

Failure during a scale up event

If the init script fails on a node that is added to the cluster during a scale up event, the node fails to start and the cluster fails to scale up. If the cluster attempts to scale up again and the node continues to fail to start, it adds to the number of accumulated node failures until the Secure Agent terminates the cluster.

Failure while recovering a master node

If you enable high availability in an AWS environment and the init script fails on a recovered master node, the node fails to start and contributes to the number of accumulated node failures over the cluster lifecycle.

Accumulated failures over the cluster lifecycle

During the cluster lifecycle, the Secure Agent tracks the number of accumulated node failures that occur due to an init script within a certain time frame. If the number of failures is too high, the agent terminates the cluster.

Find the log files for the nodes where the init script failed and use the log files to resolve the failures before running a job to start the cluster again.

Updating the runtime environment or the staging location

To update the runtime environment or the staging location, perform one of the following tasks based on the status of the Secure Agent and the advanced cluster:

The Secure Agent and the advanced cluster are running.

If the agent and the cluster are running, complete the following tasks:

1. Update the runtime environment or the staging location in the advanced configuration.
2. Stop the cluster when you save the configuration.

The Secure Agent is unavailable or the advanced cluster cannot be reached.

If the agent is unavailable or the cluster cannot be reached, complete the following tasks:

1. Run the command to delete the cluster or make sure that all cluster resources are deleted by logging in to your account on the cloud platform. For information about commands, see [Appendix A, "Command reference" on page 149](#).
2. Update the runtime environment or the staging location in the advanced configuration.
3. Disable the cluster when you save the configuration.

Note: If you update the runtime environment, the new Secure Agent will create a new advanced cluster with a different cluster ID.

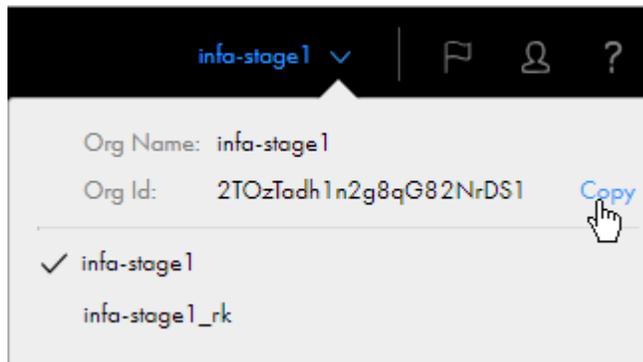
CHAPTER 8

Troubleshooting

Use the following sections to troubleshoot errors in advanced clusters.

Note: To get support for advanced clusters, you might need to give your organization ID to Informatica Global Customer Support. You can find your organization ID through the **Organization** menu in the upper right corner.

The following image shows the **Organization** menu:



To copy the organization ID, click the **Copy** option that appears when you hover the cursor to the right of the **Org ID** field.

You can also find your organization ID on the **Organization** page in Administrator.

Troubleshooting an advanced cluster

What should I do if the status of the advanced cluster is Unknown?

When the cluster status is Unknown, first verify that the Secure Agent is running. If the agent is not running, enable the agent and check whether the cluster starts running.

If the cluster does not start running, an administrator can run the command to list clusters. If the command output returns the cluster state as partial or in-use, the administrator can run the command to delete the cluster.

For more information about the commands, see the Administrator help.

I looked at the `ccs-operation.log` file to troubleshoot the advanced cluster, but there wasn't enough information. Where else can I look?

You can look at the `cluster-operation` logs that are dedicated to the instance of the advanced cluster. When an external command set begins running, the `ccs-operation` log displays the path to the `cluster-operation` logs.

For example:

```
2020-06-15 21:22:36.094 [reqid:] [T:000057] INFO :
c.i.c.s.c.ClusterComputingService [CCS_10400] Starting to run command set
[<command set>] which contains the following commands: [
  <commands> ;
]. The execution log can be found in the following location: [/data2/home/cldagnt/
SystemAgent/apps/At_Scale_Server/35.0.1.1/ccs_home/3xukm9iqp5zeahyrb7rqoz.k8s.local/infa/
cluster-operation.log].
```

The specified folder contains all `cluster-operation` logs that belong to the instance of the cluster. You can use the logs to view the full `stdout` and `stderr` output streams of the command set.

The number in the log name indicates the log's generation and each `cluster-operation` log is at most 10 MB. For example, if the cluster instance generated 38 MB of log messages while running external commands, the folder contains four `cluster-operation` logs. The latest log has 0 in the file name and the oldest log has 3 in the file name. You can view the messages in the `cluster-operation0.log` file to view the latest errors.

If you set the log level for the Elastic Server to `DEBUG`, the `ccs-operation` log shows the same level of detail as the `cluster-operation` logs.

How do I find the initialization script logs for the nodes where the init script failed?

To find the init script logs, complete the following tasks:

1. Locate the `ccs-operation.log` file in the following directory on the Secure Agent machine:
`<Secure Agent installation directory>/apps/At_Scale_Server/<version>/ccs_home/`
2. In the `ccs-operation.log` file, find a message that is similar to the following message:

```
Failed to run the init script for cluster [cluster instance ID] on the following
nodes: [cluster node IDs]. Review the log in the following S3 file path: [cloud
platform location].
```
3. Navigate to the cloud platform location that is provided in the message.
4. Match the cluster node IDs to the init script log file names for the nodes where the init script failed.

How are the resource requirements calculated in the following error message for an advanced cluster?

```
2019-04-26T19:04:11.762+00:00 <Thread-16> SEVERE: java.lang.RuntimeException:
[java.lang.RuntimeException: The Cluster Computing System rejected the Spark task
[InfaSpark0] due to the following error: [[CCS_10252] Cluster
[6bjwune8v4bkt3vneokii9.k8s.local] doesn't have enough resources to run the application
[spark--infaspark0e6674748-b038-4e39-a2a9-3fd49e63f289infaspark0-driver] which requires
a minimum resource of [(KB memory, mCPU)]. The cluster must have enough nodes, and each
node must have at least [(KB memory, mCPU)] to run this job.].]
```

The first resource requirement is the total number of resources that are required by the Spark driver and the Spark executor.

The second resource requirement is calculated based on the minimum resource requirements on each worker node to run a minimum of one Spark process.

The resources are calculated using the following formulas:

```
Memory: MAX(driver_memory, executor_memory)
CPU: MAX(driver_CPU, executor_CPU)
```

The Spark process can be either a Spark driver process or a Spark executor process. The cluster must have two nodes where each node fulfills the minimum requirements to run either the driver or the executor, or the cluster must have one node with enough resources to run both the driver and the executor.

Note: The resource requirements for the driver and executor depend on how you configure the following advanced session properties in the mapping task:

```
spark.driver.memory
spark.executor.memory
spark.executor.cores
```

For more information about minimum resource requirements, see the Administrator help.

I shut down the Secure Agent machine on my cloud platform, but some jobs are still running.

When you shut down the agent machine, the agent starts on a new machine, but jobs do not carry over to the new machine.

In Monitor, cancel the jobs and run them again. The agent on the new machine will start processing the jobs.

To avoid this issue, see the instructions to shut down the agent machine in the Administrator help.

Troubleshooting an advanced cluster on AWS

Why did the advanced cluster fail to start?

To find out why the advanced cluster failed to start, use the `ccs-operation.log` file in the following directory on the Secure Agent machine:

```
<Secure Agent installation directory>/apps/At_Scale_Server/<version>/ccs_home/
```

The following table lists some reasons why a cluster might fail to start:

| Reason | Possible Cause |
|--|---|
| The cluster operator failed to update the cluster. | The VPC limit was reached on your AWS account. |
| The master node failed to start. | The master instance type isn't supported in the specified region or availability zone or in your AWS account. |
| All worker nodes failed to start. | The worker instance type isn't supported in the specified region or availability zone or in your AWS account. |
| The Kubernetes API server failed to start. | The user-defined master role encountered an error. |

When a cluster fails to start due to at least one of these reasons, the `ccs-operation.log` file displays a `BadClusterConfigException`.

For example, you might see the following error:

```
2019-06-27 00:50:02.012 [T:000060] SEVERE : [CCS_10500] [Operation of <cluster instance ID>: start_cluster-<cluster instance ID>]:
com.informatica.cloud.service.ccs.exception.BadClusterConfigException: [[CCS_10207] The
cluster configuration for cluster [<cluster instance ID>] is incorrect due to the
following error: [No [Master] node has been created on the cluster. Verify that the
instance type is supported.]. The Cluster Computing System will stop the cluster soon.]
```

If the cluster encounters a `BadClusterConfigException`, the agent immediately stops the cluster to avoid incurring additional resource costs and to avoid potential resource leaks. The agent does not attempt to recover the cluster until the configuration error is resolved.

I ran a job to start the advanced cluster, but the VPC limit was reached.

When you do not specify a VPC in the advanced configuration for a cluster, the Secure Agent creates a new VPC on your AWS account. Because the number of VPCs on your AWS account is limited for each region, you might reach the VPC limit.

If you reach the VPC limit, edit the advanced configuration and perform one of the following tasks:

- Provide a different region.
- Remove the availability zones. Then, provide an existing VPC and specific subnets within the VPC for the cluster to use.

Any cloud resources that were provisioned for the cluster will be reused when the cluster starts in the new region or the existing VPC. For example, the Secure Agent might have provisioned Amazon EBS volumes before it received an error for the VPC limit. The EBS volumes are not deleted, but they are reused during the next startup attempt.

I ran a job to start the advanced cluster, but the cluster failed to be created with the following error:

```
Failed to create cluster [<cluster instance ID>] due to the following error:
[[CCS_10302] Failed to invoke AWS SDK API due to the following error: [Access Denied
(Service: Amazon S3; Status Code: 403; Error Code: AccessDenied; Request ID: <request
ID>; S3 Extended Request ID: <S3 extended request ID>)].].]
```

The Secure Agent failed to create the advanced cluster because Amazon S3 rejected the agent's request.

Make sure that the S3 bucket policies do not require clients to send requests that contain an encryption header.

How do I troubleshoot a Kubernetes API Server that failed to start?

If the Kubernetes API Server fails to start, the advanced cluster fails to start. To troubleshoot the failure, use the Kubernetes API Server logs instead.

To find the Kubernetes API Server logs, complete the following tasks:

1. Connect to the master node from the Secure Agent machine.
2. On the master node, locate the Kubernetes API Server log files in the directory `/var/log/`.

I updated the staging location for the advanced cluster. Now mappings fail with the following error:

```
Error while executing mapping. ExecutionId '<execution ID>'. Cause: [Failed to start
cluster for [01000D25000000000005]. Error reported while starting cluster [Cannot apply
cluster operation START because the cluster is in an error state.]].]
```

Mappings fail with this error when you change the permissions to the staging location before you change the S3 staging location in the advanced configuration.

If you plan to update the staging location, you must first change the S3 staging location in the advanced configuration and then change the permissions to the staging location on AWS. If you used role-based security, you must also change the permissions to the staging location on the Secure Agent machine.

To fix the error, perform the following tasks:

1. Revert the changes to the permissions for the staging location.

2. Edit the advanced configuration to revert the S3 staging location.
3. Stop the cluster when you save the configuration.
4. Update the S3 staging location in the configuration, and then change the permissions to the staging location on AWS.

I updated the staging location for the advanced cluster. Now the following error message appears in the agent job log:

```
Could not find or load main class com.informatica.compiler.InfaSparkMain
```

The error message appears when cluster nodes fail to download Spark binaries from the staging location due to access permissions.

Verify access permissions for the staging location based on the type of connectors that the job uses:

Connectors with direct access to Amazon data sources

If you use credential-based security for advanced jobs, make sure that the credentials in the Amazon S3 V2 and Amazon Redshift V2 connections can be used to access the staging location.

If you use role-based security for advanced jobs, make sure that the advanced cluster and the staging location exist under the same AWS account.

Connectors without direct access to Amazon data sources

If you use a user-defined worker role, make sure that the worker role can access both the staging location and the data sources in the advanced job.

If you use the default worker role, make sure that the Secure Agent role can access both the staging location and the data sources in the advanced job.

I restarted the Secure Agent machine and now the status of the advanced cluster is Error.

Make sure that the Secure Agent machine and the Secure Agent are running. Then, stop the advanced cluster in Monitor. In an AWS environment, the cluster might take 3 to 4 minutes to stop. After the cluster stops, you can run an advanced job to start the cluster again.

Is there anything I should do before I use a custom AMI to create cluster nodes?

If you use a custom AMI (Amazon machine image) to create cluster nodes, make sure that the AMI contains an installation of the AWS CLI.

The Secure Agent uses the AWS CLI to propagate tags to Amazon resources and to aggregate logs. The cluster nodes also use the AWS CLI to run initialization scripts.

For information about how to use a custom AMI, contact Informatica Global Customer Support.

Troubleshooting an advanced cluster on Microsoft Azure

I restarted the Secure Agent machine and now the status of the advanced cluster is Error.

Make sure that the Secure Agent machine and the Secure Agent are running. Then, stop the advanced cluster in Monitor. In an Azure environment, the cluster might take 10 minutes to stop. After the cluster stops, you can run a job to start the cluster again.

The init script failed with the following standard error on some nodes in the advanced cluster:

```
Created symlink from /etc/systemd/system/apt-daily.service to /dev/null.  
Created symlink from /etc/systemd/system/apt-daily-upgrade.service to /dev/null.  
Removed symlink /etc/systemd/system/timers.target.wants/apt-daily.timer.  
Removed symlink /etc/systemd/system/timers.target.wants/apt-daily-upgrade.timer.  
E: Could not get lock /var/lib/dpkg/lock-frontent - open (11: Resource temporarily  
unavailable)  
E: Unable to acquire the dpkg frontend lock (/var/lib/dpkg/lock-frontent), is another  
process using it?
```

The init script failed because the node was running an internal process at the same time as the init script. If you continue to see the error, wait for the internal process to complete by placing a sleep command for the required duration in your init script.

For example, you might use a sleep command as follows:

```
#!/bin/sh  
  
while(sudo lsof /var/lib/dpkg/lock-frontent)  
do  
echo "Sleeping 10s"  
sleep 10  
done  
  
sudo apt-get -y update  
sudo apt-get install -y expect
```

Troubleshooting an advanced cluster subtask

The job failed but there are many logs I can view. Where do I start?

Troubleshoot the job by examining the logs in the following order:

1. Execution plan. Debug the Scala code for the job.
2. Session log. Debug the logic that compiles the job and generates the Spark execution workflow.
3. Agent job log. Debug how the Secure Agent pushes the Spark execution workflow to the advanced cluster for processing.
4. Spark driver and executor logs. Debug how the advanced cluster runs the job.

You can download the execution plan, session log, agent job log, and Spark driver log in Monitor.

To find the Spark executor log, copy the advanced log location for a specific Spark task that failed. Then, navigate to the log location on your cloud platform and download the log.

I can't find all of the log files for the job that failed. I've tried to download the logs from both Monitor and the log location on my cloud platform.

The logs that are available for the job depend on the step where the job failed during processing.

For example, if the job fails before the job is pushed to the advanced cluster, the Spark driver and executor logs are not generated in the log location, and Monitor cannot query the logs from the cloud platform either.

You can recover some of the log files, but you might have to use other types of logs to troubleshoot the job.

I can't find the Spark driver and Spark executor logs. Can I recover them?

If you can't download the Spark driver log from the user interface, you can recover the log using the Spark driver Pod. You cannot recover Spark executor logs.

When the Secure Agent pushes a job to an advanced cluster, the Secure Agent creates one Spark driver Pod and multiple Spark executor Pods to run the Spark tasks. You can use the Spark driver Pod to recover the Spark driver log, but you cannot recover the Spark executor logs. The Spark driver Pod deletes the Spark executor Pods immediately after a job succeeds or fails.

Note: When a job succeeds or fails, the Spark driver Pod is deleted after 5 minutes by default. If you need to increase the limit to assist troubleshooting, contact Informatica Global Customer Support.

To recover the Spark driver log, perform the following tasks:

1. Find the name of the Spark driver Pod in the agent job log. For example, see the name of the Spark driver Pod in the following message:

```
2019/04/09 11:10:15.511      : INFO :Spark driver pod [spark-
passthroughparquetmapping-veryvery-longlongname-1234567789-
infaspark02843891945120475434-driver] was successfully submitted to the cluster.
```

If you cannot download the agent job log in Monitor, the log is available in the following directory on the Secure Agent machine:

```
<Secure Agent installation directory>/apps/At_Scale_Server/<version>/logs/job-logs/
```

The file name of the agent job log uses the format *AgentLog-<Spark job ID>.log*. You can find the Spark job ID in the session log. For example, the Spark job ID is *0c2c5f47-5f0b-43af-a867-da011452c19dInfaSpark0* in the following message of the session log:

```
2019-05-09T03:07:52.129+00:00 <LdtmWorkflowTask-pool-1-thread-9> INFO: Registered
job to status checker with Id 0c2c5f47-5f0b-43af-a867-da011452c19dInfaSpark0
```

2. Confirm that the Spark driver Pod exists. If the driver Pod was deleted, you cannot retrieve the Spark driver log.

To confirm that the driver Pod exists, navigate to the following directory on the Secure Agent machine:

```
<Secure Agent installation directory>/apps/At_Scale_Server/<version>/mercury/
services/shared/kubernetes/kubernetes_<version>/bin
```

In the directory, run the following command:

```
./kubectl get pods
```

3. Find the cluster instance ID in one of the following ways:

- Locate the cluster instance ID in the session log. For example, you might see the following message:

```
2019/05/07 16:22:00.20      : INFO :[SPARK_2005] Uploading the local file in the
path [/export/home/builds/ws/yxiao_hadoopvm_ML/Mercury/platformdiscale/main/
components/cluster/hadoop-tests/cats/edtm/spark/./target/
hadoop3a0b1db6-76ea-4317-8272-5b3a8dfd2171_InfaSpark0/
log4j_infa_spark.properties] to the following shared storage location: [s3a://
soki-k8s-local-state-store/k8s-infa/testcluster2.k8s.local/staging/
sess4280021555102778947/log4j_infa_spark.properties].
```

Note the following cloud storage location that you see in the message:

```
s3a://soki-k8s-local-state-store/k8s-infa/testcluster2.k8s.local/staging/
```

The cluster instance ID is the entry that follows "k8s-infa." In this case, the ID is *testcluster2.k8s.local*.

- Locate the cluster instance ID in the *ccs-operation.log* file. The file is located in the following directory on the Secure Agent machine:

```
<Secure Agent installation directory>/apps/At_Scale_Server/<version>/ccs_home/
```

4. Log in to the Secure Agent machine as the sudo user that started the agent.

5. Set the environment variable KUBECONFIG on the Secure Agent machine to the following value:
`<Secure Agent installation directory>/apps/At_Scale_Server/<version>/ccs_home/
<cluster ID>/kube/kubeconfig.yaml`
6. To retrieve the Spark driver log, navigate to the following directory on the Secure Agent machine:
`<Secure Agent installation directory>/apps/At_Scale_Server/<version>/mercury/
services/shared/kubernetes/kubernetes_<version>/bin`
In the directory, run the following command:
`./kubectl logs <Spark driver pod name>`

Troubleshooting a self-service cluster

A mapping run on a self-service cluster fails when the self-managed Kubernetes cluster is not reachable.

The mapping fails with the following error:

```
2022-06-23T04:42:10.872+00:00 <getThreadPoolTaskExecutor-502> INFO: Waiting for cluster  
with Cluster Instance ID : [16y6xhsvjkdeybtzdyldkx.k8s.local] to start.  
2022-06-23T04:42:13.394+00:00 <getThreadPoolTaskExecutor-502> SEVERE:  
WES_internal_error_An unexpected error occurred during execution.
```

Verify if you can access the self-managed Kubernetes cluster from the Secure Agent machine.

If you can access the self-managed Kubernetes cluster from the Secure Agent machine and if the mapping is still failing, wait for cluster's idle timeout (30 minutes) and monitor the cluster state. When the cluster state changes to STOP, start the cluster, and then run the mapping.

If you do not want to wait for the cluster's idle timeout, restart the Secure Agent process and then run the mapping.

When you run a mapping, if you stop the self-service cluster in between, the mapping fails with the following error after the restarting cluster:

```
<SparkTaskExecutor-pool-1-thread-11> SEVERE: Reattemptable operation failed with error:  
Failure executing: POST at: https://35.84.220.154:6443/api/v1/namespaces/default/pods.  
Message: pods "spark-infaspark0229e35d4-d9d1-4203-a2b1-d4692ace052finfaspark0-driver" is  
forbidden: error looking up service account default/infa-spark: serviceaccount "infa-  
spark" not found, metadata=ListMeta(_continue=null, remainingItemCount=null,  
resourceVersion=null, selfLink=null, additionalProperties={}), reason=Forbidden,  
status=Failure, additionalProperties={}
```

To resolve the error, restart the Secure Agent process and then run the mapping.

Shutting down the Secure Agent machine and cloud resources

When you shut down the Secure Agent machine, ensure that all cloud resources that were provisioned for an advanced cluster are deleted.

To properly shut down the Secure Agent machine, perform the following tasks:

1. In Monitor, stop the advanced cluster if the cluster is running.
2. In Administrator, stop the Secure Agent.
3. On the cloud platform, shut down the Secure Agent machine.

If you shut down the Secure Agent machine when the cluster is running, only the cluster nodes are shut down. Other resources remain on the cloud, such as any networks, staging data and log files, and storage devices.

If you shut down the Secure Agent machine before you stop the cluster or before you stop the Secure Agent, restart the Secure Agent machine and make sure that the Secure Agent is running. Then use Monitor to stop the cluster. After the cluster stops, stop the Secure Agent and shut down the Secure Agent machine.

Note: When you restart the Secure Agent machine, the cluster status becomes Error in Monitor.

APPENDIX A

Command reference

Use the provided shell commands to help you configure and manage cluster deployments. For example, you can run a command to delete a cluster that was not stopped completely.

Before you run commands

Before you can run commands, verify that you have configured the JAVA_HOME environment variable on the Secure Agent machine and that the Java version on the Secure Agent machine is compatible with JDK 8.

Running commands

Run the commands in the following directory on the Secure Agent machine:

```
<Secure Agent installation directory>/apps/At_Scale_Server/<version>/mercury/services/  
shared/kubernetes/kubernetes_<version>/scripts/
```

The version is the version number of the Elastic Server.

Note: When you run the commands, the current directory must be the directory where the scripts are located.

generate-policies-for-userdefined-roles.sh

Generates the policy content for the master and worker roles in an AWS environment.

The output is saved to the `my-userdefined-master-worker-role-policies.json` file. You can restrict certain elements in the policy content and attach the content as policies to the master and worker roles. For more information, see [“Create user-defined master and worker roles” on page 32](#).

The command uses the following options:

```
-h | -help  
-sd | -staging-dir=<cluster-staging-directory>  
-ld | -logging-dir=<cluster-logging-directory>
```

The following table describes each option:

| Option | Description |
|---------------------|---|
| -help -h | Access the help for the command. |
| -staging-dir -sd | Staging directory for the advanced cluster. Use the format <code>-staging-dir=bucket/folder</code> . The directory must include at least the bucket name. Do not include the prefix <code>s3://</code> . |
| -logging-dir -ld | Logging directory that stores logs for the advanced cluster and advanced jobs that run on the cluster. Use the format <code>-logging-dir=bucket/folder</code> . The directory must include at least the bucket name. Do not include the prefix <code>s3://</code> . |

list-clusters.sh

Lists all clusters in a staging directory.

The command uses the following options:

```
-h | -help

-d | -staging-dir=<cluster-bucket-location-without-prefix-s3://> (AWS environment) or
<staging-location-with-prefix-abfs[s]://> (Azure environment)

-azsrg | -azure-storage-resource-group

-ac | -azurecpath=azcredfilepath

-ct | -cluster-type
```

The following table describes each option:

| Option | Description |
|---|---|
| -help -h | Access the help for the command. |
| -staging-dir -d | Staging directory that is configured in the advanced configurations for the clusters. Use one of the following formats based on your cloud platform: - AWS. <code>-staging-dir=<bucket name>/<folder name></code> . The directory must include at least the bucket name. Do not include the prefix <code>s3://</code> . - Microsoft Azure. <code>-staging-dir=abfs(s)://<file system>@<storage account>.dfs.core.windows.net/<folder path></code> Specify the ABFSS protocol if encryption is enabled on the storage location. |
| -azsrg -azure-storage-resource-group | Storage resource group that holds the staging storage account and is configured in the advanced configurations for the clusters. |

| Option | Description |
|----------------------|---|
| -azurepath -ac | Location of the Azure credentials file on the Secure Agent machine which contains APPID, TENANTID, SERVICE PRINCIPAL, SUBSCRIPTION. Not applicable in an AWS environment. Note: Scripts that contain this option will fail. Use this option in a Microsoft Azure environment only if instructed by Informatica Global Customer Support. |
| -cluster-type -ct | Cluster type for advanced clusters in an AWS environment or local clusters in an AWS or Microsoft Azure environment. You can specify local, kubeadm, or kops. By default, the command runs on clusters that are managed by kubeadm. Not applicable in an Azure environment. |

delete-clusters.sh

Deletes clusters in a staging directory.

The command uses the following options:

```
-h | -help

-d | -staging-dir=<cluster-bucket-location-without-prefix-s3://> (AWS environment) or
<staging-location-with-prefix-abfs[s]://> (Azure environment)

-azsrg | -azure-storage-resource-group

-s | -deletable-states=state-1[,state-2,...]

-c | -clusters=cluster1[,cluster2,...]

-f | -force

-ac | -azurepath=azcredfilepath

-ct | -cluster-type
```

The following table describes each option:

| Option | Description |
|---|--|
| -help -h | Access the help for the command. |
| -staging-dir -d | Staging directory that is configured in the advanced configurations for the clusters. Use one of the following formats based on your cloud platform: - AWS. -staging-dir=<bucket name>/<folder name>. The directory must include at least the bucket name. Do not include the prefix s3://. - Microsoft Azure. -staging-dir=abfs(s)://<file system>@<storage account>.dfs.core.windows.net/<folder path> Specify the ABFSS protocol if encryption is enabled on the storage location. |
| -azsrg -azure-storage-resource-group | Storage resource group that holds the staging storage account and is configured in the advanced configurations for the clusters. |

| Option | Description |
|-------------------------|--|
| -deletable-states -s | <p>Comma-separated list of cluster states. If the state of a cluster matches one of the listed states, the cluster is deleted.</p> <p>You can list any of the following states:</p> <ul style="list-style-type: none"> - Deleted. Deletes clusters that are not using any resources on the cloud. In an AWS environment, the remaining information that is stored on the cloud is historical metadata on Amazon S3. The command wipes cluster storage and deletes the cluster state, creation history, and staging directories. - Metadata-only. Deletes clusters that have not started. In an AWS environment, the command deletes only the Kubernetes state store for the cluster. - Partial. Deletes clusters that failed to be started, or clusters that were started but were not stopped completely. In an AWS environment, the command runs the Kubernetes delete command to delete the cloud resources that were provisioned for the cluster. - In-use. Deletes clusters that are highly-likely to have virtual machines running, but the clusters might not have running jobs. In an AWS environment, the command runs the Kubernetes delete command to delete the cloud resources that were provisioned for the cluster. - All. Deletes clusters with any of the above states. <p>In a Microsoft Azure environment, deleting a cluster removes all cluster information from the staging directory.</p> <p>For example, use <code>-deletable-states=metadata-only,partial</code> to delete clusters that have not started and clusters that failed to start.</p> <p>To delete clusters with all of these states, use <code>-deletable-states=all</code>.</p> |
| -clusters -c | <p>Comma-separated list of clusters that you want the command to examine.</p> <p>For example, you might have a development environment and a test environment that use the same staging directory. You want to delete clusters with the states <code>partial</code> and <code>in-use</code> that are in the test environment but not the development environment. To delete clusters only from the test environment, list the clusters that are in the test environment.</p> |
| -force -f | <p>Skip additional prompts.</p> <p>If you do not use the <code>-force</code> option, the command lists each advanced cluster and prompts you to confirm that you want to delete the cluster. You can enter <code>Yes</code> or <code>No</code>.</p> <p>If you use the <code>-force</code> option, the clusters are deleted automatically.</p> |
| -azurecpath -ac | <p>Location of the Azure credentials file on the Secure Agent machine which contains APPID, TENANTID, SERVICE PRINCIPAL, SUBSCRIPTION. Not applicable in an AWS environment.</p> <p>Note: Scripts that contain this option will fail. Use this option in a Microsoft Azure environment only if instructed by Informatica Global Customer Support.</p> |
| -cluster-type -ct | <p>Cluster type for advanced clusters in an AWS environment or local clusters in an AWS or Microsoft Azure environment. You can specify <code>local</code>, <code>kubeadm</code>, or <code>kops</code>. By default, the command runs on clusters that are managed by <code>kubeadm</code>. Not applicable in an Azure environment.</p> |

For example, the following command examines certain clusters in the staging directory `autodeploy/devbld` and deletes the clusters that have the status `deleted`, `metadata-only`, or `in-use`:

```
delete-clusters.sh -d=autodeploy/devbld -deletable-states=deleted,metadata-only,in-use -c=testcluster.k8s.local,testcluster.k8s.local,testcluster2.k8s.local,testcluster3.k8s.local,testcluster4.k8s.local
```

cluster-operations.sh

Performs operations on clusters in a staging directory, such as listing the clusters or deleting the clusters.

The command uses the following syntax:

```
cluster-operations.sh <cloud environment> <operation> <argument1> <argument2>
[<argument3>...]
```

Use `gcp` as the cloud environment for Google Cloud. Use `local` as the cloud environment for a local cluster on Google Cloud.

The arguments that you use depend on the operation. You can use the following operations:

list

Lists the clusters in a staging directory.

When you use the list operation, use the following syntax:

```
cluster-operations.sh <cloud environment> list <staging location> <project ID>
```

The following table describes the arguments you use with the list operation:

| Argument | Description |
|------------------|---|
| Staging location | Staging directory that is configured in the advanced configurations for the clusters. In a Google Cloud environment, use the following syntax: <code>gs://<bucket>/<folder></code> |
| Project ID | Unique identifier of the Google Cloud project that contains the cluster resources. |

For example, the following command lists the clusters in staging folder of the project `myproject1`:

```
cluster-operations.sh gcp list gs://mybucket/cluster/staging myproject1
```

delete

Deletes the clusters in a staging directory.

When you use the delete operation, use the following syntax:

```
cluster-operations.sh <cloud environment> delete <staging location> <project ID>
<deletable states> <clusters> [force]
```

The following table describes the arguments you use with the delete operation:

| Argument | Description |
|------------------|---|
| Staging location | Staging directory that is configured in the advanced configurations for the clusters. In a Google Cloud environment, use the following syntax: <code>gs://<bucket>/<folder></code> |
| Project ID | Unique identifier of the Google Cloud project that contains the cluster resources. |

| Argument | Description |
|------------------|--|
| Deletable states | <p>Comma-separated list of cluster states. If the state of a cluster matches one of the listed states, the cluster is deleted.</p> <p>You can list any of the following states:</p> <ul style="list-style-type: none"> - Deleted. Deletes clusters that are not using any resources on the cloud. - Metadata-only. Deletes clusters that have not started. - Partial. Deletes clusters that failed to be started, or clusters that were started but were not stopped completely. - In-use. Deletes clusters that are highly-likely to have virtual machines running, but the clusters might not have running jobs. - All. Deletes clusters with any of the above states. <p>In a Google Cloud environment, deleting a cluster removes all cluster information from the staging directory.</p> |
| Clusters | <p>Comma-separated list of clusters that you want the command to examine.</p> <p>For example, you might have a development environment and a test environment that use the same staging directory. You want to delete clusters with the states <code>partial</code> and <code>in-use</code> that are in the test environment but not the development environment. To delete clusters only from the test environment, list the clusters that are in the test environment.</p> <p>You can also use <code>all</code> to examine all the clusters in the staging directory.</p> |
| Force | <p>Optional. Use <code>force</code> to skip additional prompts.</p> <p>If you do not use the <code>force</code> argument, the command lists each cluster and prompts you to confirm that you want to delete the cluster. You can enter <code>Yes</code> or <code>No</code>.</p> <p>If you use the <code>force</code> argument, the clusters are deleted automatically.</p> |

For example, the following command deletes all the deleted and partial clusters in the project `myproject1` without prompting you to confirm each cluster:

```
cluster-operations.sh gcp delete gs://mybucket/cluster/staging myproject1
deleted,partial all force
```

INDEX

A

- advanced cluster subtasks
 - troubleshooting [145](#)
- advanced clusters
 - access to resources [16](#), [17](#), [64](#), [65](#), [80–82](#)
 - advanced configuration [107](#)
 - AWS [142](#)
 - CLAIRE-powered configuration [105](#)
 - cluster operator policy [27](#)
 - Google Cloud integration tasks [63](#)
 - integration tasks [13](#), [79](#), [93](#)
 - local cluster [10](#)
 - Microsoft Azure [144](#)
 - overview [7](#), [8](#)
 - self-service cluster [10](#)
 - troubleshooting [142](#), [144](#)

C

- cloud permissions
 - AWS [130](#)
 - Google Cloud [131](#)
 - Microsoft Azure [132](#)
- cluster
 - configuring proxy settings [71](#), [84](#)
- clusters
 - cluster operations command [153](#)
 - commands [149](#), [153](#)

E

- elastic cluster
 - credential-based security [44](#)
 - elastic configuration [104](#)
 - role-based security [43](#), [44](#)
 - security [43](#), [44](#)
- elastic clusters
 - access to resources [18–20](#), [66](#), [83](#)
 - agent installation [25](#), [70](#), [83](#), [96](#)
 - AWS subscriptions [14](#)
 - cluster operator role [22](#), [26](#), [30](#), [31](#)
 - commands [46](#), [77](#), [92](#), [149–151](#)
 - data encryption [42](#), [116](#), [119](#), [126](#)
 - default roles [41](#), [60](#)
 - delete clusters command [151](#)
 - elastic configuration [116](#), [120](#), [133](#)
 - Elastic Server [46](#)
 - firewall rule [69](#)
 - generate policies command [149](#)
 - Google Cloud NAT gateway [68](#)
 - Google Cloud services [64](#)
 - high availability [114](#), [125](#)
 - initialization script [138](#)

- elastic clusters (*continued*)
 - JAVA_HOME [46](#), [77](#), [92](#)
 - labeling [119](#)
 - list clusters command [150](#)
 - managed identity [85](#)
 - maximum CLI/API session duration [30](#)
 - Microsoft Azure products [80](#)
 - organization privileges [14](#), [63](#)
 - prerequisites [13](#), [63](#), [79](#)
 - resource requirements [137](#), [138](#)
 - role-based security [42](#)
 - routing [21](#)
 - runtime environment [139](#)
 - Secure Agent role [26](#), [30](#), [31](#), [44](#)
 - security [42](#)
 - security principals [85](#), [88](#), [89](#), [91](#)
 - self service cluster [133](#)
 - service principal [88](#)
 - staging location [114](#), [125](#), [139](#)
 - subnet [21](#), [68](#)
 - tagging [115](#), [125](#), [126](#)
 - user-defined roles [32](#), [40](#), [41](#), [60](#), [61](#)
 - VPC [21](#), [68](#), [69](#)
 - worker role [60](#)

G

- Google Cloud custom roles [72](#)

L

- local cluster
 - AWS [103](#)
 - cloud permissions [130–132](#)
 - cluster properties [127](#)
 - data encryption [132](#)
 - install agent [102](#)
 - prerequisites [101](#)
 - setting up [101](#)
 - staging and log locations [127](#)
 - troubleshooting [103](#)

M

- master role [60](#)

P

- proxy settings
 - cluster [71](#), [84](#)

S

- security group
 - ELB security group [22](#)
 - master security group [22](#)
 - worker security group [22](#)
- self-service cluster
 - annotations [94](#)
 - custom properties [136](#)
 - overview [10](#)
 - tolerations [94](#)
- self-service clusters
 - AWS CLI authentication [100](#)

- self-service clusters (*continued*)
 - organization privileges [94](#)
 - prerequisites [93](#)
 - user-managed service account [96](#)

T

- troubleshooting
 - advanced cluster subtasks [145](#)
 - advanced clusters [142](#), [144](#)
 - local cluster [103](#)