



## Upgrade

Hansel Miranda

Course Developer, Google Cloud



Google Cloud

In this module, you will learn how to upgrade your Apigee hybrid installation to a new release of the software distribution.

You will complete a lab to upgrade the hybrid runtime plane to a newer version.

We will also discuss rollback procedures so you can roll back the Apigee hybrid installation to a previous version.

---

## Agenda

[Upgrade Process](#)

Lab

Rollback Process



In this lecture, we will review the process to upgrade Apigee hybrid to a newer version of the software distribution.

## Upgrade prerequisites

Before performing an upgrade:

- Review official documentation and release notes for specific considerations related to upgrade path and target version.

[Apigee hybrid release notes](#)

- Back up your hybrid runtime Cassandra database (recommended).



Before performing an upgrade, you should always review the release notes for the Apigee hybrid release version you are upgrading to.

The release notes contains important information about the new features that are released, any features that are in beta, bug fixes, any required configuration file changes, and any updates to supporting software platforms.

It is also recommended to back up your hybrid runtime Cassandra database following the instructions in the Apigee hybrid [documentation](#).

## Kubernetes platform upgrade

- Upgrading your version of Apigee hybrid requires a supported version of the Kubernetes platform, as documented in the release notes.
- Follow the platform documentation to upgrade Kubernetes to the required version for Apigee hybrid.

Apigee Hybrid Runtime Platform
<a href="#">GKE</a>
<a href="#">Anthos clusters on VMware</a>
<a href="#">Azure Kubernetes Service (AKS)</a>
<a href="#">Elastic Kubernetes Service (EKS)</a>
<a href="#">Anthos clusters on AWS</a>



Upgrading your version of Apigee hybrid requires a supported version of the Kubernetes platform.

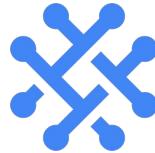
The minimum required Kubernetes version is documented in the release notes of the Apigee hybrid release version you are upgrading to.

Confirm that your Kubernetes platform version is supported. If it isn't, follow your platform provider documentation to upgrade it.

## Upgrade Anthos Service Mesh

To upgrade Anthos Service Mesh, perform these steps:

- Upgrade Anthos Service Mesh using the documentation for your Kubernetes platform.
- Update the [istio-operator.yaml](#) file to specify ingress ports and other properties as documented in the Apigee hybrid upgrade documentation.
- Apply the changes using the [istioctl](#) command.



To minimize downtime when upgrading Anthos Service Mesh:

- Run at least 2 hybrid runtime clusters.
- Divert API traffic away from the cluster to be upgraded.
- Take the cluster to be upgraded offline, and apply the upgrade.
- Repeat the process for the remaining clusters.



To upgrade Anthos Service Mesh, follow the steps documented for your Kubernetes platform.

After installation, you update the `istio-operator.yaml` file with ingress ports and other properties as specified in the Apigee hybrid documentation.

You then apply these changes using the `istioctl` command.

Upgrading Apigee hybrid may require downtime due to the upgrade of the ingress gateway in the cluster.

To minimize downtime in production environments, we recommend that you run the hybrid runtime plane in 2 or more clusters.

Apply the upgrade to the clusters one at a time after diverting API traffic away from the cluster being upgraded.

## Bootstrapping

Download the [hybrid release package](#) for your operating system.

For Linux 64-bit:

```
curl -LO \
https://storage.googleapis.com/apigee-release/hybrid/apigee-hybrid-setup/{version}/api
geectl_linux_64.tar.gz
```

For example: To download version 1.4.0 for 64-bit Linux:

```
curl -LO \
https://storage.googleapis.com/apigee-release/hybrid/apigee-hybrid-setup/1.4.0/apigee
ctl_linux_64.tar.gz
```



You first download the new release of the Apigee hybrid distribution for your operating system.

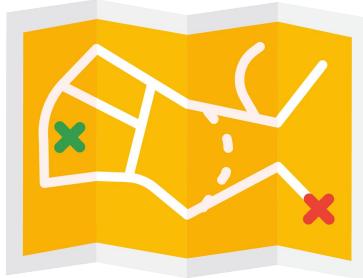
The software distribution packages are hosted in a public cloud storage repository.

## Re-map the installation directory

Follow these steps to re-map the installation directory of your hybrid runtime installation:

1. Identify the base install directory that contains `$APIGEECTL_HOME`.
2. Extract the new version of the Apigee hybrid software in this directory.
3. Rename the `current` apigeectl directory to the previous version of the software distribution.
4. Rename the `newly extracted directory` to apigeectl.

Detailed instructions are documented [here](#).



The next step in the upgrade process is to re-map your installation directory.

Identify the base installation directory that was created when Apigee hybrid was originally installed.

To find the base directory, inspect the value of the `APIGEECTL_HOME` environment variable. It will point to a directory that resides within the base directory.

Extract the new version of Apigee hybrid software in the base installation directory.

Rename the current apigeectl directory. For example, if the current version is 1.1.1, rename the apigeectl directory to apigeectl\_1.1.1.

Rename the newly extracted installation directory to apigeectl. This is now where the environment variable `$APIGEECTL_HOME` points to.

## Update configuration

- Depending on the version being upgraded to, changes may be necessary in the overrides.yaml configuration file.
- Make a copy of your overrides.yaml file to save the current version.
- Make the corresponding changes as documented in the hybrid release notes [documentation](#) for the runtime components.

```
overrides.yaml
...
envs:
  - name: test
    serviceAccountPaths:
      synchronizer: keypath/sync-sa.json
      udca: keypath/udca-sa.json
virtualhosts:
  - name: testenvgrp
    sslCertPath: ./certs/fullchain.pem
    sslKeyPath: ./certs/privkey.pem
...
...
```



For some newer versions of Apigee hybrid, you may have to update the overrides.yaml file used to configure your runtime plane cluster.

Make a copy of your current overrides file, and save the old file in case you ever need to roll back.

Make the required changes to the overrides file based on the release documentation.

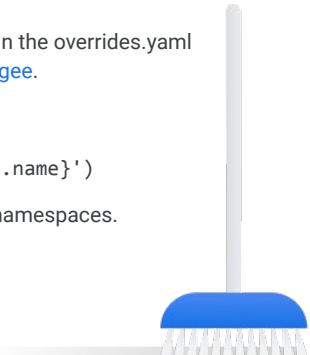
## Job cleanup

Before applying the changes to the cluster, [clean up any completed jobs](#) in the namespaces used by the hybrid runtime. Use the following steps:

- Run the `kubectl delete job` command for the given namespace.
- Replace `{namespace}` with the name you used when configuring hybrid in the `overrides.yaml` file. If you did not specify a namespace, use the default namespace: [apigee](#).

```
$kubectl delete job -n {namespace} \
$(kubectl get job -n {namespace}
-o=jsonpath='{.items[?(@.status.succeeded==1)].metadata.name}'')
```

- Repeat the previous command for the [apigee-system](#) and [istio-system](#) namespaces.



It is a good practice to clean up any completed jobs in your cluster from the previous installation.

To delete a job, use the `kubectl delete job` command, passing in the name of the completed job to be deleted.

Repeat the command for all the completed jobs in the [apigee-system](#) and [istio-system](#) namespaces used in Apigee hybrid.

## Apply the upgrade

Follow these steps to apply the upgrade to the cluster:

1. To initialize the cluster, use the new version of the `overrides.yaml` configuration file:  
`$APIGEECTL_HOME/apigeectl init -f new-overrides.yaml`  
`init` - applies configurations for prerequisite components, such as Istio and Apigee Deployment, to the cluster.
2. Check to determine when the initialization is complete:  
`$APIGEECTL_HOME/apigeectl check-ready -f new-overrides.yaml`
3. When `check-ready` indicates that all component pods are ready, apply the upgrade:  
`$APIGEECTL_HOME/apigeectl apply -f new-overrides.yaml`
4. Finally, re-run `check-ready` to determine when the upgrade is complete.



You are now ready to apply the upgrade.

Run the `apigeectl init` command to re-initialize the cluster with the updated `overrides.yaml` configuration file.

This will update the Apigee hybrid system components. Run the `apigeectl check-ready` command to confirm that the initialization has completed.

If you want to check for configuration errors before modifying the cluster further, use the `--dry-run` option with the `apigeectl apply` command.

Then, update the hybrid workload components in the cluster by running the `apigeectl apply` command with the updated `overrides.yaml` file.

Finally, re-run the `apigeectl check-ready` command to confirm that all the containers in the cluster are running.

## Apply the upgrade (production)

For production environments, you should apply the upgrade separately to each set of runtime components in the cluster.

- Initialize the cluster by running: `$APIGEECTL_HOME/apigeectl init -f new-overrides.yaml`

When the system components are ready, apply your overrides to upgrade these components in the following sequence, using the commands shown:

1. Cassandra datastore:  
`$APIGEECTL_HOME/apigeectl apply -f new-overrides.yaml --datastore`
2. Telemetry components:  
`$APIGEECTL_HOME/apigeectl apply -f new-overrides.yaml --telemetry`
3. Organization-scoped components (MART, Watcher, and Apigee Connect).  
`$APIGEECTL_HOME/apigeectl apply -f new-overrides.yaml --org`
4. Environment-scoped components (UDCA, Synchronizer, Runtime) either all together or individually.  
`$APIGEECTL_HOME/apigeectl apply -f new-overrides.yaml --all-envs -or-`  
`$APIGEECTL_HOME/apigeectl apply -f new-overrides.yaml --env {env}`
5. Finally, check to determine when the all the components are ready:  
`$APIGEECTL_HOME/apigeectl check-ready -f new-overrides.yaml`



For production environments, you should apply the upgrade separately to each set of runtime components in the cluster.

After the cluster is initialized, apply the new overrides file first to the Cassandra datastore component, then to the telemetry components, the organization-scoped components, and finally to the environment-scoped components.

Finally, run the `apigeectl check-ready` command to confirm that all the containers in the cluster are running.

## Rolling updates

- Apigee hybrid supports [rolling updates](#).
  - In Kubernetes, rolling updates allow Deployment updates to take place by incrementally updating pod instances with new ones.
- To trigger an in-place update, modify the desired settings in the overrides file and apply it to the cluster.
- For example, to increase runtime memory from 1Gi to 5Gi:
  - Create a new configuration file with the new settings.
  - Apply the changes to the cluster using the apigeectl command.
- After the configuration changes are applied, updated pods start up and replace the existing pods.

```
...  
runtime:  
replicaCountMin: 2  
replicaCountMax: 20  
resources:  
cpu: 1000m  
memory: 1Gi  
...
```

```
...  
runtime:  
replicaCountMin: 2  
replicaCountMax: 20  
resources:  
cpu: 1000m  
memory: 5Gi  
...
```



Apigee hybrid supports rolling updates to the runtime components in the cluster.

In Kubernetes, rolling updates are implemented by incrementally updating pod instances with new ones that are scheduled on nodes with available resources.

In the example, a new overrides.yaml configuration file is created with the memory property of the runtime message processor component increased from its current setting.

You then apply the new configuration to the cluster by running the apigeectl apply command.

After the configuration changes are applied, updated pods start up and replace the existing pods.

## AB Rolling updates

- An AB rolling update directs a small percentage of traffic to the updated pods and increases that percentage over time.
- After 100% of traffic is sent to the new pods, the old pods are deleted.
- To trigger an AB rolling update, use the revision property with a different value, and then modify the desired settings in the overrides file and apply it to the cluster.
- For example, to increase runtime memory from 1Gi to 5Gi:
  - Create a new configuration file with the new settings.
  - Set the revision property to a value other than the current value, and update the desired memory settings.
  - Apply the changes to the cluster using the apigeectl command.

```
...  
revision: blue  
...  
runtime:  
  replicaCountMin: 2  
  replicaCountMax: 20  
resources:  
  cpu: 1000m  
  memory: 1Gi  
...
```

```
...  
revision: green  
...  
runtime:  
  replicaCountMin: 2  
  replicaCountMax: 20  
resources:  
  cpu: 1000m  
  memory: 5Gi  
...
```



Apigee hybrid supports AB rolling updates to the runtime components configuration.

In this type of rolling update, a small percentage of traffic is initially directed to the updated pods. The traffic is increased incrementally until 100% of the traffic flows to the updated pods; the old pods are then deleted from the cluster.

To implement an AB rolling update, use the revision property in your overrides.yaml configuration file.

In the example, a new overrides.yaml configuration file is created with the revision property set to green and the memory property of the runtime message processor component increased from its current setting.

The revision property can be set to any value that is different from the current setting.

You then apply the new configuration to the cluster by running the apigeectl apply command.

[Performing an AB update](#)

## Official documentation

Apigee hybrid official documentation or the release notes may describe [additional steps](#) needed for upgrading between specific versions. We strongly recommend that you review this documentation as part of the upgrade planning process.

Upgrading Apigee hybrid:

- [Upgrade process](#)
- [Apigee hybrid release notes](#)
- [Apigee hybrid release process](#)



In addition to the Apigee hybrid release notes, you should also review the upgrade and release process documentation on the Apigee website.

The documentation may describe additional steps to be performed when upgrading your hybrid release between specific versions.

---

## Agenda

Upgrade Process

[Lab](#)

Rollback Process



We will now complete a lab to upgrade Apigee hybrid to a newer version of the software distribution.

# Lab

## Upgrading Apigee Hybrid



In this lab, you upgrade the hybrid runtime cluster on Google Kubernetes Engine.

Using a pre-installed older version of Apigee hybrid, you will upgrade the hybrid runtime plane to a later version.

You verify that the runtime plane is upgraded successfully by deploying a test API proxy to an environment and testing that the proxy works as expected.

## Lab review

### Upgrading Apigee Hybrid



In this lab, you upgraded the hybrid runtime cluster in Google Kubernetes Engine.

You verified that the upgrade was successful by deploying an API proxy to an environment, and verifying that it functions as expected.

## Agenda

Upgrade Process

Lab

[Rollback Process](#)



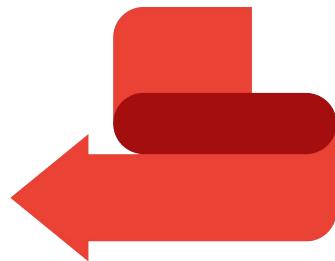
Let's discuss the process to roll back Apigee hybrid in the runtime plane.

## Rollback process overview

If an error occurs during or after an upgrade of hybrid, you can roll back the upgrade.

The rollback process includes a few steps similar to the upgrade process:

1. Clean up any completed jobs from the upgrade process.



If you observe errors in the Apigee hybrid runtime plane during or after an upgrade, you can roll the upgrade back.

To roll back an upgrade, follow these steps.

First, clean up any completed jobs in your cluster from the upgrade process.

To delete a job, use the `kubectl delete job` command, passing in the name of the completed job to be deleted.

Repeat the command for all the completed jobs in the `apigee-system` and `istio-system` namespaces used in Apigee hybrid.

## Rollback process overview

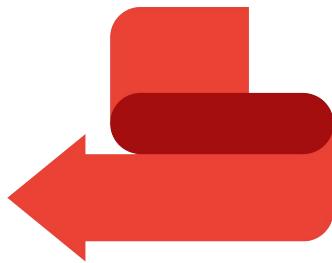
If an error occurs during or after an upgrade of hybrid, you can roll back the upgrade.

The rollback process includes a few steps similar to the upgrade process:

1. Clean up any completed jobs from the upgrade process.

2. Delete the Apigee Operators deployment.

```
$ kubectl -n apigee-system delete deployment  
apigee-controller-manager
```



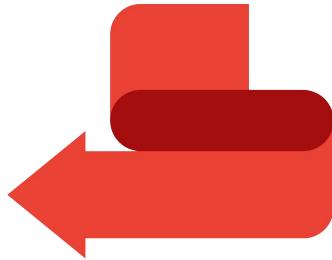
Delete the Apigee Operators deployment by using the `kubectl delete deployment` command.

## Rollback process overview

If an error occurs during or after an upgrade of hybrid, you can roll back the upgrade.

The rollback process includes a few steps similar to the upgrade process:

1. Clean up any completed jobs from the upgrade process.
2. Delete the Apigee Operators deployment.
3. Point the [apigeectl directory](#) back to the previous version.



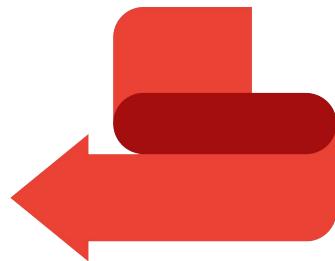
Change the APIGEECTL\_HOME environment variable to point to the directory that contains the original version of apigeectl.

## Rollback process overview

If an error occurs during or after an upgrade of hybrid, you can roll back the upgrade.

The rollback process includes a few steps similar to the upgrade process:

1. Clean up any completed jobs from the upgrade process.
2. Delete the Apigee Operators deployment.
3. Point the [apigeectl directory](#) back to the previous version.
4. Re-initialize the hybrid runtime plane with base Apigee components.



Run the `apigeectl init` command to re-initialize the cluster with the `overrides.yaml` configuration file from your previous hybrid installation.

Run the `apigeectl check-ready` command to confirm that the initialization has completed.

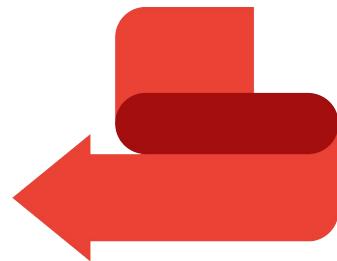
## Rollback process overview

If an error occurs during or after an upgrade of hybrid, you can roll back the upgrade.

The rollback process includes a few steps similar to the upgrade process:

1. Clean up any completed jobs from the upgrade process.
2. Delete the Apigee Operators deployment.
3. Point the [apigeectl directory](#) back to the previous version.
4. Re-initialize the hybrid runtime plane with base Apigee components.
5. Re-apply the [previous](#) version of the overrides.yaml configuration to the cluster.

It is a best practice to store your overrides.yaml file in source code control for easy access to previous revisions.



Next, run the apigeectl apply command to update the hybrid workload components in the cluster with the overrides.yaml configuration file from your previous hybrid installation.

Finally, re-run the apigeectl check-ready command to confirm that all the containers in the cluster are running.

It is a best practice to store your overrides.yaml files in source code control for easy access to previous revisions.

## References

Rollback steps may vary by software version and the target version to roll back to. Before you decide to roll back, it is important to consult the official documentation for details.



The steps to roll back might vary based on the source and target version of the hybrid software distribution.

Consult the rollback process documentation for your version of Apigee hybrid.



## Review: Upgrade

Hansel Miranda  
Course Developer, Google Cloud



In this module, you learned how to upgrade the Apigee hybrid runtime plane to a newer version of the software distribution.

You completed a lab to upgrade the runtime plane to a newer version.

You also learned about the rollback process in case you need to roll back Apigee hybrid due to errors in the newer version.

In the next module, you will learn how Apigee hybrid logs system informational and error messages and how you can use analytics and metrics data to monitor and troubleshoot Apigee hybrid.