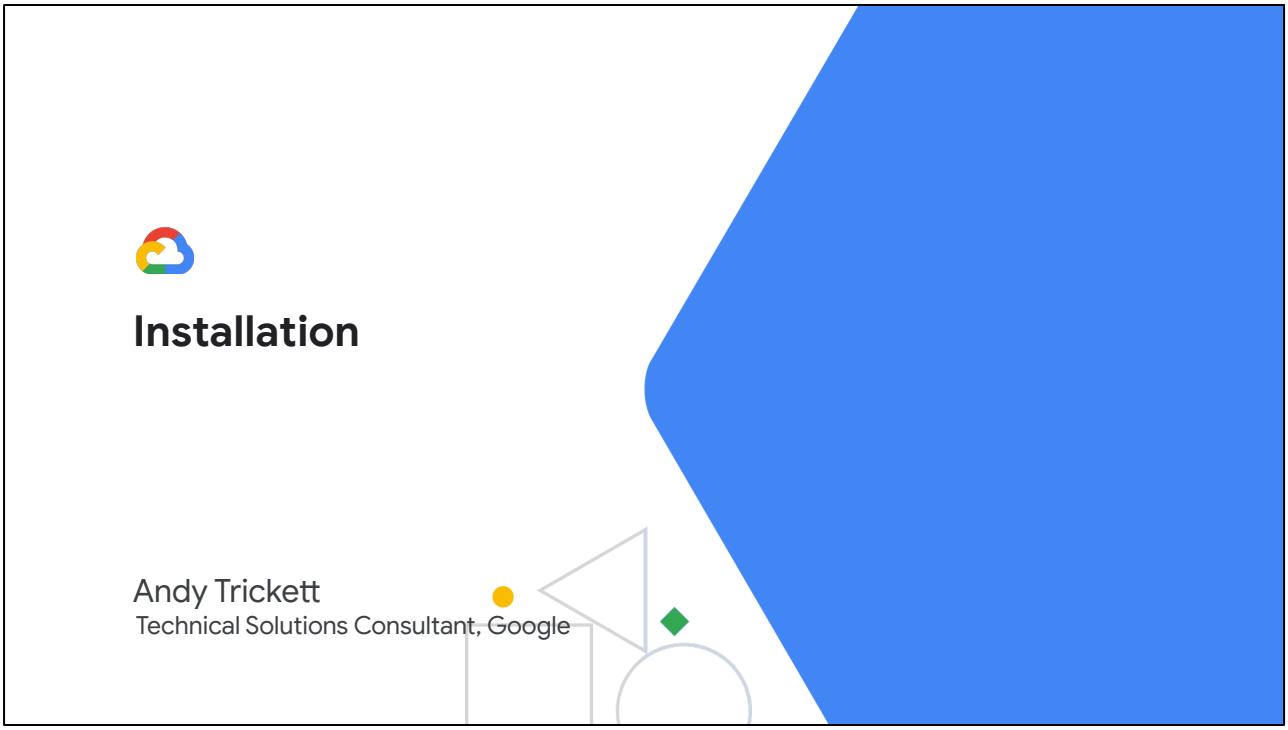
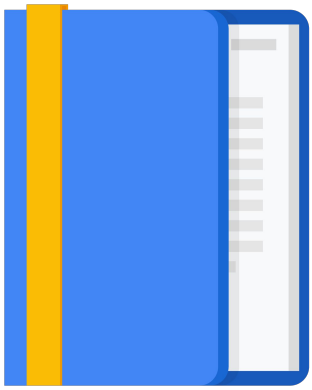


Welcome to the Installation module of the On-premises Installation and Fundamentals course of Google Cloud's Apigee API Platform.

In this module, you will learn how to install the Apigee platform and complete a lab to install Apigee for private cloud.

Agenda

- Installation planning
- Installation dependencies
- Installation process
- Lab
- Installing the new Edge UI
- Maintenance and uninstall process



We will start by discussing the planning process, where you will learn how to design your installation topology.

You will learn how to prepare for the installation by provisioning infrastructure and fulfilling prerequisites for the installation.

After that, we will discuss the installation process. This will involve bootstrapping your hosts, setting up edge profiles, and provisioning one or more organizations.

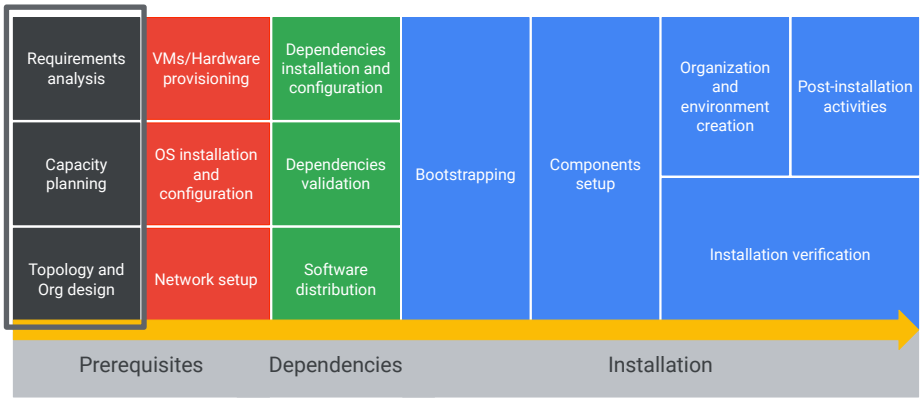
You then complete a lab to install Apigee for private cloud on Google Compute Engine.

Finally, we discuss installing the new Edge UI and explore maintenance and post installation activities.

Let's get started.

Installation planning

A successful installation is driven by the appropriate topology design and the implementation and validation of all prerequisites.



Investing the time to plan your installation is critical to success when installing Apigee for private cloud.

The upfront investment pays for itself in time saved later during the installation process.

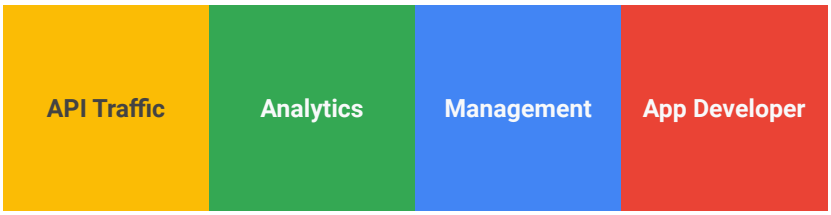
Let's discuss the planning activities that you will need to perform.

These are:
requirements analysis,

- capacity planning,
- installation topology design, and
- organization design.

Understanding requirements

Requirement analysis and design for Apigee Edge topologies is contextual to the business and technical needs across each of these functional areas.



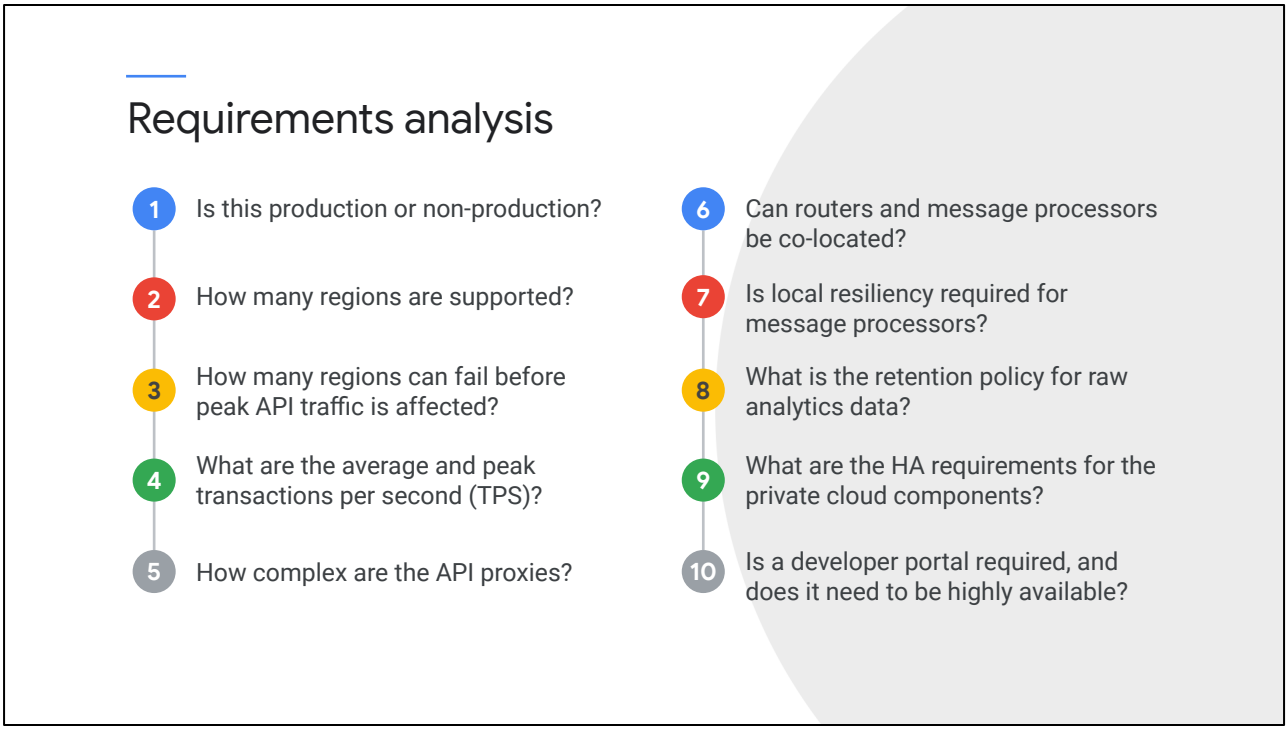
The requirements analysis process provides a key input to the design of the installation topology.

It drives many of the decisions you will make when determining how many hosts you will need and what their relationships will be.

When planning your requirements, you must consider these four areas of operation:

- API traffic
- Analytics
- Platform management
- and App developer management

Let’s discuss some of the important considerations in these four areas when analyzing your requirements.



Here are some of the questions to ask when analyzing your requirements:

Is the planet intended for non-production or production use?

How many regions are included the installation? And how many can fail before peak API traffic is impacted? How many hosts are you willing to lose in each region before a regional failover is necessary?

What are the average and peak loads you expect to experience? How large are your average request and response payloads?

How complex do you expect your API proxies to be? How long do you need to retain detailed analytics data?

Do you need redundant management servers, and if so, how many do you need?

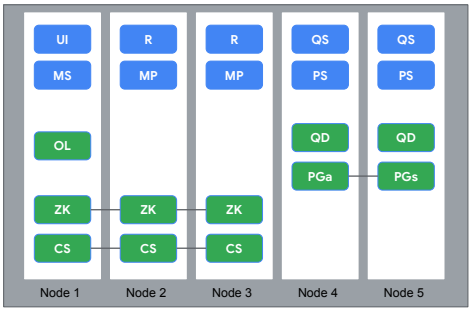
Do you intend to use the developer portal?

These questions can help you determine if an architecture template is suitable for your installation, or whether it is more appropriate to custom design your own installation topology.

Topology design

Edge topology diagram

- Placement of components per VM (node)
- Placement of virtual machines on each network zone, region, and availability zone



Apigee for Private Cloud is horizontally scalable and provides great flexibility with regard to adding logical and physical capacity to any installation.

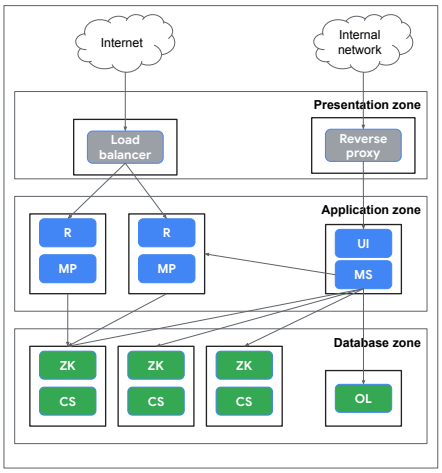
Changing requirements and evolving assumptions can easily be accommodated after the initial installation.

The flexibility provided by the architecture allows you to design the topology in an agile manner.

It is important to lay out a solid foundation. The flexibility in the architecture that is provided by the product, won't fully compensate for some of the technical and budget limitations that are imposed by traditional IT infrastructure.

Topology design involves the placement of Apigee private cloud components on each virtual machine or node that is used in the installation. You must also consider the placement of the VMs or nodes in network, region, and availability zones.

Topology design



Network topology diagram

- Placement of virtual machines on each network zone, region, and availability zone along with all relevant network devices and software such as load balancers
- Incoming and outgoing connections between virtual machines, including Edge components
- Incoming traffic to virtual hosts and load balancer, and outgoing traffic to target servers (backend systems)
- References to TLS termination, if applicable on Edge components

Another important consideration in your installation topology is network zoning.

A common requirement that is driven by security or regulatory policy, is the splitting of your installation across one or more network zones, which are separated by network firewalls - for example, you may have a DMZ zone, an application zone, and a database zone.

Apigee uses many TCP ports for internal and external communication.

Increasing the number of network zones in your architecture will increase the complexity of the network firewall configuration required to support that architecture.

Where feasible, try to minimize the number of network zones to keep installation and support of the system straightforward.

You should also consider where to terminate TLS encryption. TLS can be terminated on the load balancer or on the router component.

Topology design

Hardware specification

- CPU, RAM requirements
- Minimum storage requirements

Component	RAM	CPU	HDD
Cassandra/ZooKeeper	16 GB	8 core	250 GB
Message processor/Router	16 GB	8 core	100 GB
Message processor	16 GB	8 core	100 GB
Router	16 GB	8 core	100 GB
Postgres/Qpid	16 GB	8 core	500 GB
Postgres active or standby	16 GB	8 core	500 GB
Qpid	8 GB	4 core	30 GB
OpenLDAP/UI/Management server	8 GB	4 core	60 GB
UI/Management server	4 GB	2 core	60 GB
OpenLDAP	4 GB	2 core	60 GB

The table shown, lists the minimum CPU, RAM, and storage requirements for each of the Apigee private cloud components.

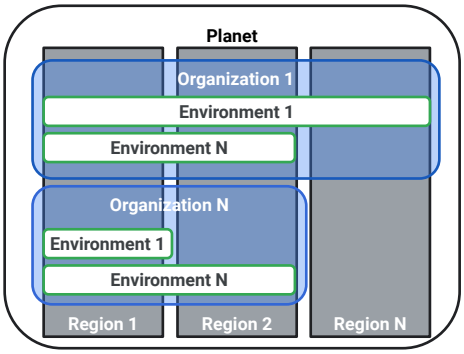
Note that the storage requirements shown, are in addition to the space required by the operating system.

Depending on your applications and network traffic, your installation may require more or fewer resources than indicated.

For example, the table includes entries for the message processor and router components that are installed either on the same machine or on separate machines.

The table has similar entries for the Qpid and OpenLDAP components.

Topology design



Organization and environment design

- Organization names
- Administrator information: e-mails, first and last names
- Environment names
- Virtual host: Virtual host names, aliases, and ports

Once you have decided on an installation topology, you will need to determine the correct Apigee organization and environment configuration for that architecture.

Organizations allow you to create security boundaries between system users. Environments allow you to group and configure deployable resources.

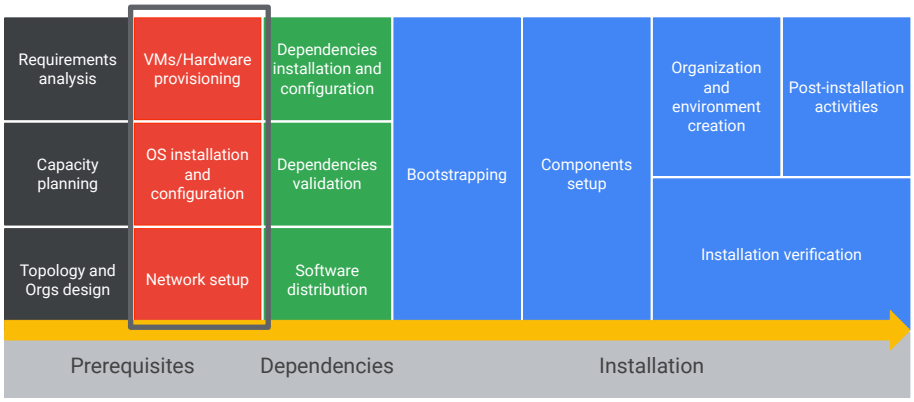
Physical message processor capacity for handling API requests is allocated at the environment level.

By default, all message processor capacity is shared across all environments. It is also possible to allocate dedicated capacity to one or more environments for security or performance reasons.

A common mapping for private cloud installations is to create a single organization named after your company, with environments used to represent various stages in your SDLC, such as dev, test, QA, staging, and production.

The organization name is the same across non-production and production installations, and the environments are split across those installations as appropriate.

Provisioning and setup



Analyzing your requirements and determining your initial installation topology is the first step to successfully serving your API platform users.

Capacity planning however, should be an ongoing process, and it actually starts ***before*** your initial installation.

Build your installation topology in a flexible way that allows you to add processing capacity and redundancy as needed. We'll discuss capacity planning thoroughly in a later course.

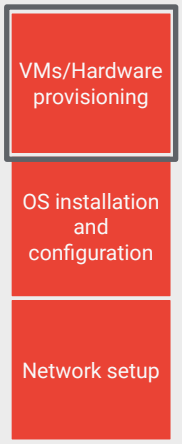
Let’s now focus on infrastructure provisioning and installation prerequisites.

Hardware, OS, and network provisioning

Hardware provisioning

- Hardware specification is defined during design.
- Apigee Edge is capable of running on physical and virtualized hardware, including IaaS providers such as Compute Engine.

https://docs.apigee.com/private-cloud/latest/installation-requirements



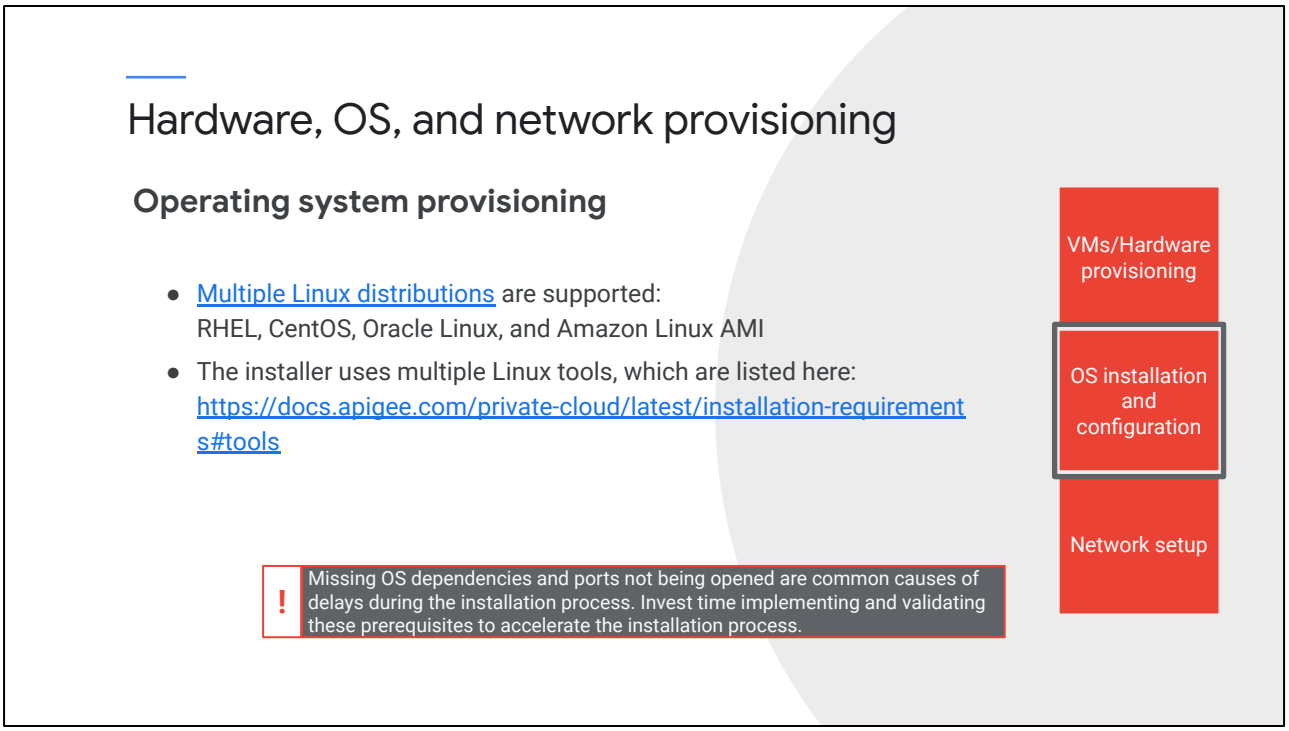
With your installation topology design complete, you have the information needed to prepare for the installation.

The first step is to provision infrastructure. The host resource requirements for an Apigee installation vary and depend on whether you're deploying a non-production or a production planet.

Production resource recommendations are described at the link shown here. If you are building a non-production planet, it may be possible to scale back performance while still retaining full functionality.

The services with the highest resource requirements are the Message Processor, Cassandra, and PostgreSQL.
All three will consume significant amounts of CPU and memory.

Cassandra and PostgreSQL performance can be affected by storage IOPS. These two services will benefit greatly from the random access performance provided by solid state disks.



Along with hardware that meets our recommended specifications, you will need a supported operating system.

A full list of supported operating systems can be found using the link shown. Multiple linux distributions like Red Hat Enterprise Linux and CentOS are supported.

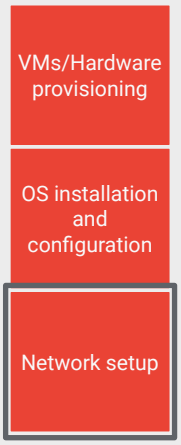
The Apigee for private cloud installer also uses several Linux tools listed at the link provided.

Hardware, OS, and network provisioning

Network provisioning

- Follow the Apigee Edge port requirements and make required firewall changes.
- Validate OS configuration to ensure that iptables, SELinux, TCP wrappers, and other security features on the OS and network allow connectivity between processes on the same machine and across machines.

! Missing OS dependencies and ports not being opened are common causes of delays during the installation process. Invest time implementing and validating these prerequisites to accelerate the installation process.



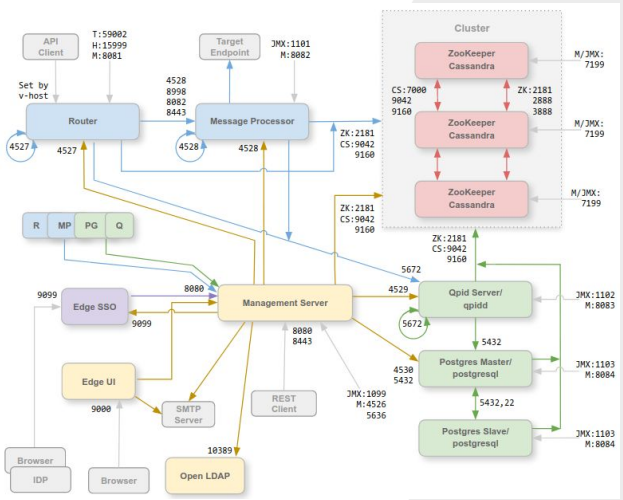
As part of the infrastructure provisioning process, you should ensure that the necessary firewall ports are opened between network zones both within regions and across regions.

Both VM and physical host firewalls must allow traffic for the ports required by the components to communicate with each other.

You should also validate the configuration of host-based network restrictions such as those enforced by iptables and TCP wrappers to allow connectivity between processes running on the same machine and across machines.

Components connectivity

Single-region view



https://docs.apigee.com/private-cloud/v4.50.00/port-requirements#single-data-center

VMs/Hardware provisioning

OS installation and configuration

Network setup

The port requirements for a single data center or region configuration are documented at the link shown.

All ports prefixed by "M" are used to manage the component and ***must*** be open on the component for access by the Management Server.

You can optionally configure TLS access for certain connections using different ports.

By default, the communications between components is not encrypted. You can add encryption by installing Apigee mTLS. We discuss this topic in a later course.

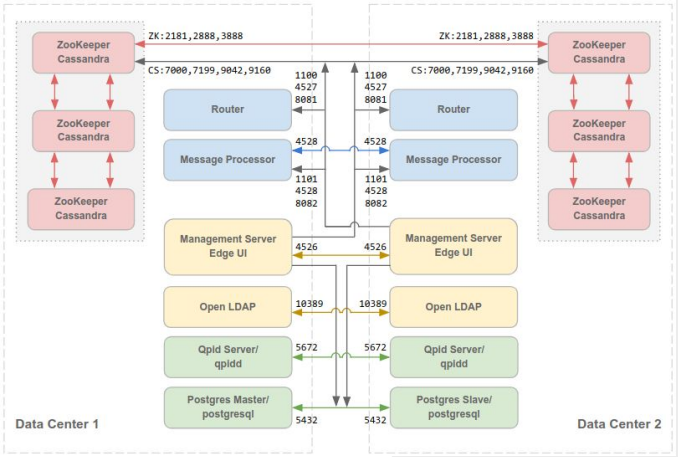
We recommend that you spend time checking and validating firewall port requests before the installation to ensure their accuracy. In our experience with customers, this is one of the most common reasons for installation delays.

A single missing port could cause the installation of a component to fail. So invest enough time in this process to ensure that no ports are omitted.

Components connectivity

Multi-region view

https://docs.apigee.com/private-cloud/v4.50.00/port-requirements#multiple-data-centers



VMs/Hardware provisioning

OS installation and configuration

Network setup

The port requirements for a multi data center or multi region configuration are documented at the link shown.

In a multi data center installation of Apigee private cloud, you must ensure that the nodes or VMs in the data centers can communicate over the ports shown.

All Management Servers must be able to access all Message Processors, Cassandra, and Postgres nodes in all other data centers.

All Message Processors in all data centers must all be able to access each other over port 4528.

For security reasons, other than the ports shown above and any others required by your own network requirements, no other ports should be open between data centers.

Agenda

Installation planning

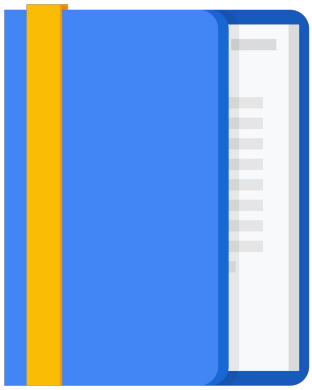
Installation dependencies

Installation process

Lab

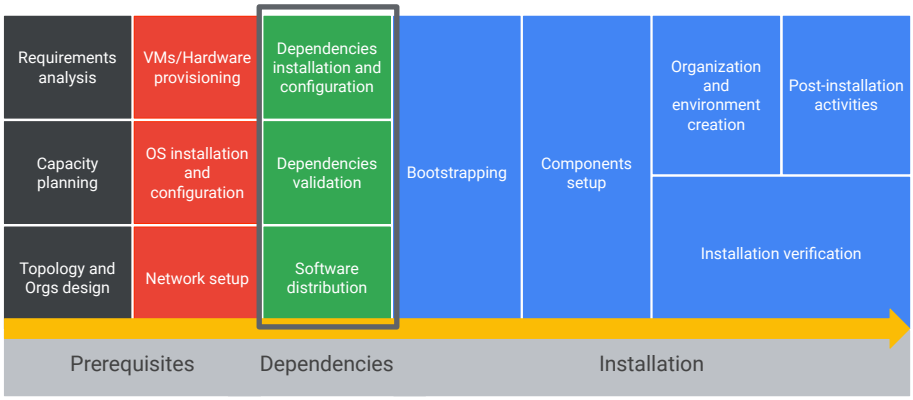
Installing the new Edge UI

Maintenance and uninstall process



Let’s now discuss the installation dependencies in Apigee Edge for private cloud.

Installation dependencies



Several dependencies must be satisfied before you install Apigee for private cloud.

These include both system and software dependencies.

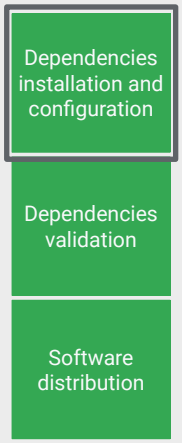
Edge software dependencies

Installation directory

- Apigee Edge installer places the software under /opt/apigee.
  - If required, the location of the software (binaries, data, and/or logs) can be changed by using symlinks.

OS service account

- The installer creates a Linux user (apigee:apigee) that is used as service account.
- If required, the Apigee user can be provisioned in advance.
  - The Apigee user must belong to the Apigee group, and the home directory for the user must exist.
- The Apigee user doesn't require authentication credentials.
  - The user is only used to own software and data files (within /opt/apigee) and to run processes.



Apigee for private cloud is installed under the /opt/apigee directory on your installation machine.

You cannot change this path, but you can use a symbolic link to point it to an alternative location.

Apigee services will run as the UID of a user named Apigee with the primary group also named Apigee. If the user and group do not already exist, they will be created automatically at the time of installation.

If you use an alternative location to install the software using a symbolic link, you must first create the Apigee user and group and then change ownership of both the symbolic link and its target to that user and group.

The Apigee user account that the installer creates has no assigned password. The account exists only to provide an unprivileged UID under which Apigee services may run and cannot be accessed interactively.

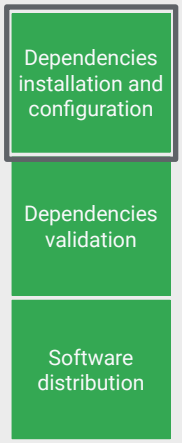
Apigee Edge software dependencies

License file

- An Apigee private cloud license file is required during the installation process.
  - This file must be accessible by the Apigee user and stored in the /tmp/apigee directory for installation.
  - The license file is copied to a permanent location under /opt/apigee during the setup process.

System limits

- The limits configuration for the apigee user in files /etc/security/limits.d/<XX-apigee.conf> must be updated on the message processor and Cassandra nodes.
- Values are described here: Installation requirements: system limits



As part of your Apigee for private cloud purchase, you receive a set of credentials for accessing the software repository and a license file that enables your private cloud installation.

Ensure that you have these available, because you will need them at installation time.

On the Cassandra and Message Processor nodes, a few system limits must be updated. They are detailed in the installation requirements link shown.

Edge software dependencies

Java

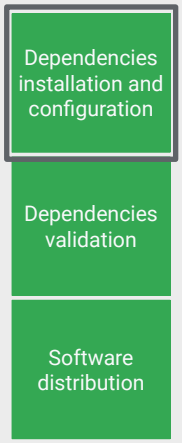
- JDK is required. Both Oracle JDK and Open JDK are supported.
- Edge installer can install the JDK during the installation process.
- Supported JDK versions can be installed manually.

The Apigee Edge installer is capable of installing Java.

You can install a specific, compatible, version of Java in advance.

- If you do, ensure that JAVA_HOME points to the root of the JDK for the user performing the installation.
- To install JDK, run the command:

```
yum -y install java-1.8.0-openjdk-devel
```



A supported version of Java is required to run Apigee services. The links shown here list currently supported versions.

You can use either the Oracle JDK or the Open JDK.

If you don't install a supported JDK in advance, the installer will automatically install the correct version of Open JDK.

Temporary working directory

/tmp/apigee

The installation process uses a temporary working directory to store:

- License file
- Configuration files
- Bootstrap scripts
- Software installer
- Other dependencies

Make sure that the directory exists and is accessible by the Apigee user.
If it does not exist, create it with the following command:

```
mkdir /tmp/apigee
```

In addition to software dependencies, you must validate additional dependencies before proceeding to install Apigee for private cloud.

The installation process uses a temporary working directory on the installation machine to store the license file, configuration files, and bootstrap and utility scripts.

The directory used is typically /apigee under the /tmp directory.

The labs used in this course automatically create this directory for you and provision a temporary license file.

Enable components connectivity

Iptables

- Stop iptables or review and configure it to allow components connectivity:

  CentOS 6.x:

  ```
  sudo /etc/init.d/iptables stop
  ```

  CentOS 7.x:

  ```
  systemctl stop firewalld
  systemctl disable firewalld
  ```

hostname

- Run the `hostname` command to confirm:
  - The correct host name for the machine
  - The correct private IP address for the machine (with the `-I` option)

TCP Wrappers

- TCP Wrappers can block communication of some ports and can affect OpenLDAP, PostgreSQL, and Cassandra installation.
- On those nodes, check `/etc/hosts.allow` and `/etc/hosts.deny` to ensure that there are no port restrictions on the required OpenLDAP, PostgreSQL, and Cassandra ports.

You should also validate the configuration of host-based network restrictions such as those enforced by iptables and TCP wrappers to allow connectivity between processes.

The default firewall on some linux systems is firewalld. You must stop and disable **firewalld** if installed.

Confirm the correct hostname and IP address of the nodes that will be used to install the software components using the hostname command.

Enable components connectivity

SELinux

To temporarily set SELinux to permissive mode, execute the following command:

CentOS 6.x:

```
echo 0 > /selinux/enforce
```

CentOS 7.x:

```
setenforce 0
```

To re-enable SELinux after installing Edge:

CentOS 6.x:

```
echo 1 > /selinux/enforce
```

CentOS 7.x

```
setenforce 1
```

To permanently disable SELinux or set it to permissive mode:
Open /etc/sysconfig/selinux in an editor and set:
`SELINUX=disabled`
or
`SELINUX=permissive`

Save your edits.
Restart the node.

SE or security enhanced linux is a security architecture for Linux systems that allows administrators to have more control over who can access the system.

SELinux must be temporarily disabled or set to permissive mode for the installation to succeed. You can re-enable it after the installation finishes.

System entropy

- If required, install and enable rng-tools:

```
yum install rng-tools -y
systemctl start rngd
systemctl status rngd
```

- Verify by executing this command:

```
head -n10 /dev/random
```

  The command should execute and return immediately, with no delay.

Usage of the ZooKeeper service requires that the system have enough entropy.

Also, installation and usage of the management server and other components may be affected by a lack of entropy on the system.

To determine if this is the case, try to read from the kernel's /dev/random device and measure the time it takes to retrieve the data.

Repeat the process at least 10 times. If you start noticing increased delay, this is an indication that there is not enough entropy on the system.

To address this issue, install rng-tools on the server using the commands shown.

Install software dependencies

Network Security Services (NSS)

Verify that you have installed NSS **v3.19**, or later.

- To check your current version, run the command:

```
yum info nss
```

- To update NSS, run the command:

```
yum update nss
```

Network Security Services (NSS) is a set of libraries that supports the development of security-enabled client and network applications.

NSS and the NSS util package are required by some of the Apigee private cloud components for their functionality.

You should verify that you have installed NSS v3.19 or higher.

Install software dependencies

Time synchronization

We recommend that you have the servers' time synchronized.

- To validate network time synchronization, use one of the following commands:

```
ntpdate
 or
chronyc tracking
```

- If there is no time synchronization daemon, you can install one with:

```
yum install ntp
 or
yum install chrony
```

Apigee recommends that your servers' times are synchronized.

If not already configured, the **ntpdate** or **chronyc** utility could serve this purpose. These utilities can verify whether the servers are time-synchronized.

You can use the yum install command to install the utilities. Note that you set up the server time zone in UTC.

System limits

Cassandra

- On the Cassandra nodes, set soft and hard limits for user (default is “apigee") in `/etc/security/limits.d/90-apigee-edge-limits.conf`

```
apigee soft memlock unlimited
apigee hard memlock unlimited
apigee soft nofile 32768
apigee hard nofile 65536
apigee soft as unlimited
apigee hard as unlimited
```

Message processor

- On the message processor nodes, set soft and hard limits for user (default is “apigee") in `/etc/security/limits.d/90-apigee-edge-limits.conf`

```
apigee soft nofile 65536
apigee hard nofile 65536
```

You must also ensure that you have set certain system limits on the Cassandra and Message Processor nodes.

On the Cassandra nodes, set soft and hard memlock, nofile, and address space limits for the installation user.

On the Message Processor nodes, set the maximum number of open file descriptors to 65,536.

These steps can be completed as a post installation step if the Apigee user is not yet created.

Additional repositories

Enable EPEL repo

You must enable Extra Packages for Enterprise Linux (or EPEL) to install or update Edge or to create a local repository.

The command you use depends on your version of RedHat/CentOS/Oracle Linux:

- For RedHat/CentOS/Oracle 7.x:
  ```
  sudo yum install
  https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
  ```
- For RedHat/CentOS/Oracle 6.x:
  ```
  sudo yum install
  https://dl.fedoraproject.org/pub/epel/epel-release-latest-6.noarch.rpm
  ```

https://fedoraproject.org/wiki/EPEL

The Apigee software depends on packages from your distributors' repositories and the EPEL repository.

You can provide access to your distributors' repositories through the internet, a proxy, or a local mirror.

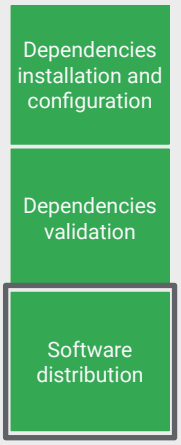
The EPEL repository can also be accessed remotely or through a local mirror.

More information on accessing EPEL is at the link shown.

Software distribution

Apigee software distribution

- Apigee Edge software is packaged as RPMs and hosted on a repository at software.apigee.com.
- During the installation process, the installer downloads all necessary software components from the repository onto local machines.
- Access to software.apigee.com is protected. In order to download the software, you require the credentials provided by Google.
- If Internet connectivity is unavailable, an offline installation process is provided:
  - This process allows you to clone the software.apigee.com repository locally and distribute it via your internal network.



Apigee hosts a software repository at software.apigee.com for hosts with internet access. To access the repository, you need credentials that are provided as part of the Apigee private cloud provisioning process.

During the installation process, the installer connects to the repository to download all necessary software components onto local machines.

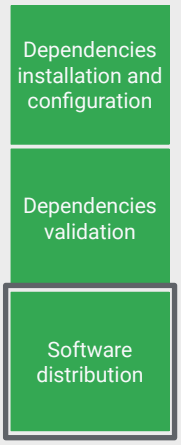
For hosts without access to the internet, you can create a local network mirror of that repository.

You can also package the repository as a tarball for distribution to each host in your cluster. For full details on the mirroring process, consult the installation guide.

Software distribution

RPM installation

- Edge software is packaged as RPMs.
- A root user, or a user with relevant privileges, is required to perform the RPM installation.



Because the installer uses RPMs, root access or a user with relevant privileges is required on all Apigee hosts to perform the installation.

After installation, you can manage the platform with only access to the Apigee user account.

Agenda

Installation planning

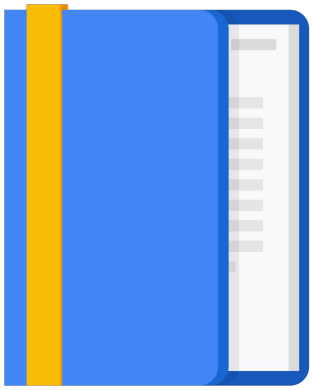
Installation dependencies

Installation process

Lab

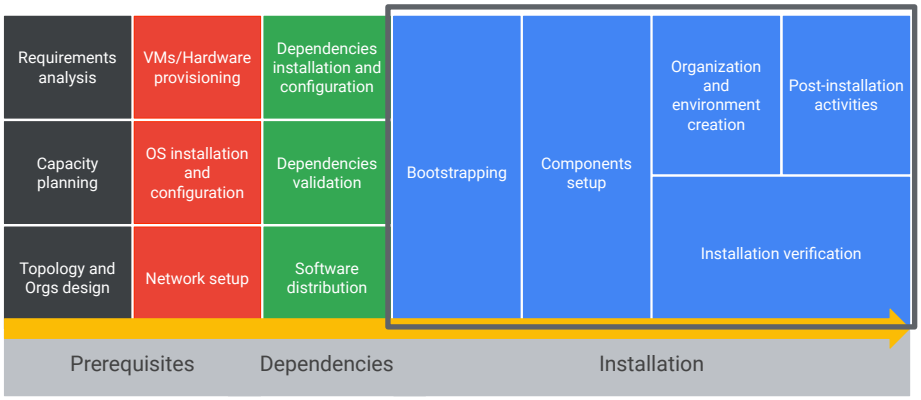
Installing the new Edge UI

Maintenance and uninstall process



In this lesson, we will talk about the process to install the Apigee Edge platform for private cloud.

Installation



The installation process comprises a set of distinct tasks:

Bootstrapping

Setup of the private cloud software components

Creating the Apigee organization and one or more environments

Post-installation activities

and finally, verifying the installation.

License file

- A license file is distributed as part of the software purchase and provisioning process.
- The file contains a key that is used by the software.
- Copy the license file (license.txt) to the /tmp/apigee directory on the installation machine.
- The license file is copied from the temporary working directory to a permanent location as part of the installation process.

At this point, the installation topology is complete.

Infrastructure has been provisioned, and prerequisites are in place.

A license file would have been distributed to you as part of the purchase and provisioning process.

It's time to now bootstrap the hosts in your cluster.

Bootstrapping

The bootstrap process performs the following actions:

- Creates the Apigee Edge directory structure: /opt/apigee.
- Creates the user account: apigee:apigee.
- Downloads and installs an initial set of utilities that are later used to install and configure individual components.

Bootstrap script-naming convention:

bootstrap_<version>.sh

Bootstrapping is the term we use for the first step of the installation process.

To bootstrap, you download and run a script from the Apigee software repository.

The script will configure your system to access the Apigee software repository and install common base utilities that all Apigee hosts require in order to complete the rest of the installation process.

During the bootstrapping process, you configure each host to communicate with the default repository at software.apigee.com, your local network mirror of that repository, or a file system location.

The bootstrap file allows you to choose the version of Apigee Edge for private cloud that will be installed from the Apigee software repository.

Bootstrapping

1. Download the bootstrap script:

```
curl https://software.apigee.com/bootstrap_<version>.sh -o
/tmp/apigee/bootstrap_<version>.sh
```

2. Execute the bootstrap script:

```
bash /tmp/apigee/bootstrap_<version>.sh apigeeuser=<uName>
```

uName is your company username.
The script will prompt for your password to software.apigee.com.

This information is provided together with the license file.

The first step in the process is to download the bootstrap script from the desired repository.

The command shown uses the repository hosted by Apigee at software.apigee.com.

In your environment, you may be downloading the script from your local mirror. If you are using the tarball software distribution method, the bootstrap script will already be available in your local file system at the location to which you decompressed the tarball.

The bootstrap script is downloaded to the /tmp/apigee directory. Note the version string embedded in the filename. Each release has its own bootstrap script, so be sure to download the correct script for your desired release.

The next step is to execute the bootstrap script. You can either make the script executable or prefix the script path with bash as shown here. Arguments to the script vary, but if you're using the Apigee hosted repository at software.apigee.com, you'll need to provide at least the Apigee user argument. You can also pass in the Apigee password argument on the command line for non-interactive use. In this example, the script will prompt for the correct password.

You may have to use some other arguments if you are hosting your own repository mirror or tarball. For instance, the protocol argument can be set to http:// or file:// if you are not using the default TLS-encrypted https connection. The Apigee repo host

argument is useful if you are using a local mirror of the repository.

Bootstrapping, cont.

The bootstrapping process installs the apigee-service utility.

apigee-service enables you to execute multiple commands, including the installation of other utilities.

3. Install apigee-setup using apigee-service:

```
/opt/apigee/apigee-service/bin/apigee-service apigee-setup install
```

apigee-setup enables you to install and configure individual components after bootstrapping and base installation are completed.

The final step in the bootstrapping process is to install the Apigee setup utility.

This utility is not automatically installed by the bootstrap script, but it is required to drive the remainder of the installation.

To install the utility, run Apigee service, Apigee setup, install. At this point, you have successfully bootstrapped the first host in the cluster.

The same bootstrapping process can be completed across all other hosts at this time before you continue with other parts of the installation.

Components setup

Using apigee-setup

The Apigee Edge installer uses a profile-based approach to install and configure components.

- To install a specific Apigee private cloud service profile, run:
  `/opt/apigee/apigee-setup/bin/setup.sh -p <profile> -f <config-file>`

-p: identifies the profile to be installed.
-f: provides the fully qualified path to the configuration file to be used.

The Apigee Edge installer uses a profile-based approach to install and configure components.

Edge component services are grouped into profiles, which are used to direct the actions of the setup script.

In addition to a profile name, the setup script requires a text configuration file, which is populated with values that can vary based on the profile being applied.

The profile setup process configures the services on each host, which are needed to fulfill one or more roles in the cluster.

Profiles

Profile	Description	Profile	Description
c	Install Cassandra only	**ui**	Install the Edge UI
zk	Install ZooKeeper only	**qs**	Install Qpid Server only
ds	Install ZooKeeper and Cassandra	**ps**	Install Postgres Server only
ld	Install OpenLDAP only	**sax**	Install the analytics components: Qpid and Postgres
ms	Install Edge Management Server, which also installs the Edge UI and OpenLDAP	**mo**	Install Monetization
r	Install Edge router only	**sa**	Install Edge standalone Includes:c, zk,ms, rmp; Omits: qs, pg
mp	Install Edge message processor only	**aio**	Install all components on a single node
rmp	Install Edge router and message processor	**sso**	Install Edge SSO
ue	Install New Edge experience		

This table shows the common profiles you might use in a typical Apigee private cloud installation.

The **r** and **mp** profiles allow you to independently install routers and message processors in situations where they reside on separate hosts. Deployments that place routers in a DMZ usually use these profiles.

The **sa** profile places all gateway and management services components on a single host.

A good option for a demo or a development deployment is the combination of the **sa** and **sax** profiles.

The **aio**, or all-in-one profile, install all the components on a single host. This can be useful for a proof of concept or to deploy a fully functional Apigee planet on a developer machine.

The two inputs to the setup program are the profile name and a configuration file.

These profiles are applied to each host used in the installation.

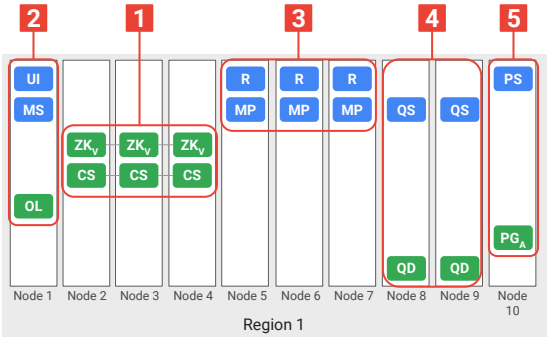
Installation order: single region

1. Data store hosts (CS, ZK)
2. Management hosts (UI, MS, OL)
3. Router, message processor (R, MP)
4. Qpid hosts (QS, QD)
5. PostgreSQL hosts (PS, PG)

- Create organizations and environments

If applicable, install:

- Developer portal
- Monetization
- Edge SSO
- New Edge UI



The installation of the Apigee private cloud components follows a specific order.

You start with the Cassandra and Zookeeper datastore components. This creates the foundation for building your cluster and will allow for replication between regions if you have multiple data centers in your design.

After the datastores are installed, you need to populate them with all the required schemas and data structures. This is done as part of the installation of the management server component, which is installed next.

The management server profile includes the Openldap component. The placement of Apigee Edge components can be quite flexible. If your topology design requires Openldap to be on a different node than the Management server, it must be deployed first.

The next step is to deploy the Message processor and Router components. They can coexist on the same node, or they can be installed on separate nodes. In the latter scenario, the message processor component must be deployed first.

After you're done with the runtime plane you move on to the analytics components. You first install the Apache Qpid message queue and the Edge components that manage and consume messages from the queues.

Qpid also handles the buffering of analytics data. In order to store this data

permanently, we finally install the postgreSQL database and Postgres server components.

This completes the installation of the Apigee Edge private cloud components. You can now create the logical entities, such as organizations and environments in the cluster.

After an organization and environment is provisioned, you then install the Apigee Developer portal and optionally the API Monetization module.

If you are integrating with an external identity provider, you would need to install the Apigee SSO module.

Finally, to use the new Edge experience, install the new Edge UI.

Silent configuration file

A configuration file:

- Is a text file.
- Is a mechanism to configure Edge components.
- Is region-specific.
- Is required for each management server.
- Contains all mandatory and optional variables.
- Must be accessible to the Apigee user account.
- Is needed on all nodes.
- Is used for both application components and organization/environments.
- Is required during installation, upgrades or adding capacity.
- Should use a standard naming convention: edge-configuration-<version>-<region>.txt

The installation process uses a text configuration file that contains a collection of variables and their values.

It is used to provide the apigee-setup utility with all the information required to configure the Apigee Edge components.

Configuration files are region-specific: one configuration file is required per region.

If the region contains more than one Management Server, an additional configuration file is required for each Management Server.

Configuration files contain required and optional variables. The list of variables is documented in the Apigee Edge Installation Guide: https://docs.apigee.com/private-cloud/latest/edge-configuration-file-reference

The configuration file must be accessible to the apigee user account. It is considered a best practice to place the configuration file in the /tmp/apigee directory on all nodes used in the installation.

The location and name of the configuration file is passed as a parameter to the apigee-setup utility during the installation process.

The configuration file is also used during the creation of organizations, environments, and users.

We strongly recommend that you keep a copy of the configuration files in a secure location or check it into source control. You may need the configuration files at a later time when performing upgrades or adding capacity to the platform.

To organize configuration files used with multi-region installations, we suggest a

Sample configuration file: single region

```
IP1=<node 1>
IP2=<node 2>
IP3=<node 3>
IP4=<node 4>
IP5=<node 5>

HOSTIP="$(hostname -i)"
MSIP="$IP1"
ADMIN_EMAIL="<email@example.com>"
APIGEE_ADMINPW="<password>"
LICENSE_FILE="/tmp/apigee/license.tx
t"
USE_LDAP_REMOTE_HOST="n"
LDAP_TYPE="1"
APIGEE_LDAPPW="<password>"
MP_POD="gateway"
REGION="dc-1"
ZK_HOSTS="$IP1 $IP2 $IP3"
ZK_CLIENT_HOSTS="$IP1 $IP2 $IP3"
CASS_HOSTS="$IP1:1,1 $IP2:1,1
$IP3:1,1"
PG_MASTER="$IP4"
PG_STANDBY="$IP5"
PG_PWD="<password>"

SKIP_SMTP="n"
SMTPHOST="<smtp.example.com>"
SMTPPORT="25"
SMTPUSER="<smtp@example.com>"
SMTPPASSWORD="<password>"
SMTPSSL="y"
BIND_ON_ALL_INTERFACES="y"
```

Here is a sample configuration file.

You would replace the right side of the name-value pairs in angle brackets with appropriate values for your installation.

For example, the node 1 through 5 values must be replaced with the internal IP addresses of the nodes used in your installation.

Host IP should always contain the IP address at which the host is reachable by the rest of the cluster.

The easiest way to do this is to use the **hostname -i** command if that command returns a single address.

MSIP sets the address of the local management server in the region.

ADMIN_EMAIL and APIGEE_ADMINPW set the username and password of the initial sysadmin user for the Apigee Edge planet.

LICENSE_FILE is the path to the license file that you were provided. It must exist on the local file system of each management server at installation time.

The USE_LDAP_REMOTE_HOST variable should almost always be set as shown. If you want to separate the Apigee internal Openldap instance from the rest of the

management stack, you can set this to y.

LDAP_TYPE can be set to one for non-replicated LDAP, or two for replicated LDAP. If you set it to two, you will also need to set additional LDAP variables that describe the settings for the replication peer as documented in the install guide.

APIGEE_LDAPPW sets the Openldap root dn password.

MP_POD should be left as gateway unless you are customizing your gateway pod configuration.

REGION will always be prefixed with dc- followed by an integer value starting at one and counting up for each subsequent region.

ZK_HOSTS contains a full list of all ZooKeeper hosts in the planet. For planets with more than one region, you may want to use a special syntax to designate observer nodes.
Check the install guide for complete details on this topic.

ZK_CLIENT_HOSTS is a list of the ZooKeeper hosts in this region, excluding any hosts in other regions. No special syntax for observers should be included here.

CASS_HOSTS describes all Cassandra hosts in the planet. An important detail to understand here is that all Cassandra hosts in the current region need to be listed first, followed by Cassandra hosts in other regions. The value of this variable will differ based on the region you are installing the Apigee Edge components.

PG_MASTER and PG_STANDBY define the active and standby hosts for the analytics databases, and PG_PWD will set the password for the analytics database.

SKIP_SMTP is a flag that will instruct the installer to configure SMTP if set to **n**. The remaining SMTP settings configure the system to communicate with your mail relay, if one is available.

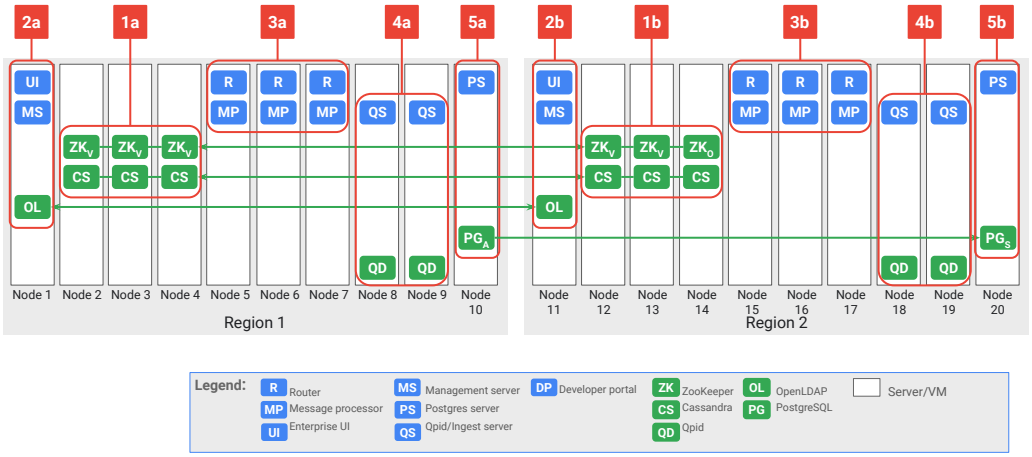
BIND_ON_ALL_INTERFACES controls whether routers and message processors bind on all interfaces, or just the interface to which the host IP variable address that is assigned.
This should normally be set as shown.

There are other variables that you can set to control the behavior of the setup script.

If you are deploying an Apigee planet with multiple regions, you will probably want to include settings for LDAP replication, and you will need to alter the contents of the ZK_HOSTS, ZK_CLIENT_HOSTS, and CASS_HOSTS variables.

Refer to the Apigee private cloud documentation for more details and examples of configuration file settings.
https://docs.apigee.com/private-cloud/latest/edge-configuration-file-reference

Installation order: Multi-region



When installing two or more regions, the installation order remains the same across component type. Install all components of the same type at one time (across all regions) before proceeding with the next component type.

As shown in the diagram, first install the data store components in both regions, shown here as Steps 1a and 1b.

Next, install the management components in steps 2a and 2b. After that, install the router and message processor components in steps 3a and 3b in all regions.

Then install the Qpid components in steps 4a and 4b followed by the PostgreSQL and Postgres server components in steps 5a and 5b.

Note that if you are installing the Openldap (OL) component separately from the Management Server (MS), install the Openldap component first.
And if you are installing the Router (R) and Message Processor (MP) components on separate nodes, install the Message Processor component first.

1. 1a, 1b: Data store hosts (CS, ZK)
2. 2a, 2b: Management hosts (UI, MS, OL)*
3. 3a, 3b: Router, message processor (R, MP)**
4. 4a, 4b: Qpid hosts (QS, QD)
5. 5a, 5b: PostgreSQL hosts (PS, PG)

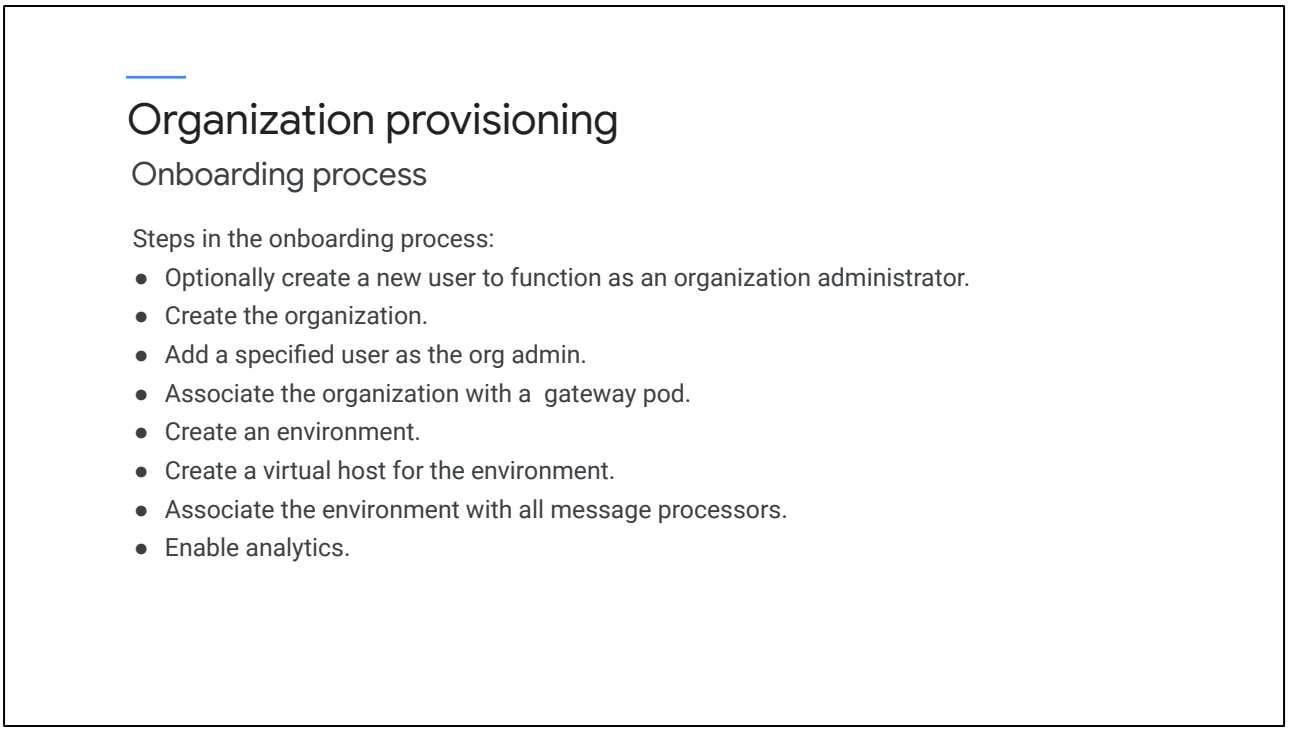
Sample configuration file: multi-region

```
HOSTIP="$(hostname -i)"
MSIP="$DC1IP1"
ADMIN_EMAIL="<email@example.com>"
APIGEE_ADMINPW="<password>"
LICENSE_FILE="/tmp/apigee/license.txt"
USE_LDAP_REMOTE_HOST="n"
LDAP_TYPE="2"
LDAP_SID="1"
LDAP_PEER="$DC2IP1"
APIGEE_LDAPPW="<password>"
MP_POD="gateway"
REGION="dc-1"
ZK_HOSTS="$DC1IP2 $DC1IP3 $DC1IP4 $DC2IP2 $DC2IP3
$DC2IP4:observer"
ZK_CLIENT_HOSTS="$DC1IP2 $DC1IP3 $DC1IP4"
CASS_HOSTS="$DC1IP2:1,1 $DC1IP3:1,1 $DC1IP4:1,1 $DC2IP2:2,1
$DC2IP3:2,1 $DC2IP4:2,1"
PG_MASTER="$DC1IP5"
PG_STANDBY="$DC2IP5"
PG_PWD="postgres"
SKIP_SMTP="y"
SMTPHOST="smtp.example.com"
SMTPPORT="25"
SMTPUSER="smtp@example.com"
SMTPPASSWORD="<password>"
SMTPSSL="n"
BIND_ON_ALL_INTERFACES="y"
```

```
HOSTIP="$(hostname -i)"
MSIP="$DC2IP1"
ADMIN_EMAIL="<email@example.com>"
APIGEE_ADMINPW="<password>"
LICENSE_FILE="/tmp/apigee/license.txt"
USE_LDAP_REMOTE_HOST="n"
LDAP_TYPE="2"
LDAP_SID="2"
LDAP_PEER="$DC1IP1"
APIGEE_LDAPPW="<password>"
MP_POD="gateway"
REGION="dc-2"
ZK_HOSTS="$DC1IP2 $DC1IP3 $DC1IP4 $DC2IP2 $DC2IP3
$DC2IP4:observer"
ZK_CLIENT_HOSTS="$DC2IP2 $DC2IP3 $DC2IP4”
CASS_HOSTS="$DC2IP2:2,1 $DC2IP3:2,1 $DC2IP4:2,1 $DC1IP2:1,1
$DC1IP3:1,1 $DC1IP4:1,1"
PG_MASTER="$DC1IP5"
PG_STANDBY="$DC2IP5"
PG_PWD="postgres"
SKIP_SMTP="y"
SMTPHOST="smtp.example.com"
SMTPPORT="25"
SMTPUSER="smtp@example.com"
SMTPPASSWORD="<password>"
SMTPSSL="n"
BIND_ON_ALL_INTERFACES="y"
```

Here is a sample configuration file for a multi-region installation of Apigee Edge for private cloud.

Note that the entries shown in red are region-specific name-value pairs.



After the Apigee private cloud components are installed and running, you follow an onboarding process to provision an initial set of objects on the platform.

This process is part of Edge's automation capabilities and is used to provision all the elements that make up a fully functional system.

Onboarding is performed using the setup-org function of the apigee provision command by running the apigee service utility. This command must be run on the Management Server.

The command optionally creates a new user to function as the organization administrator, creates the organization, and adds the specified user as the org admin.

It also creates an environment and a virtual host for that environment and associates the environment with all the message processors.

It finally enables analytics on the platform.

Organization and environment creation

apigee-provision

- An organization and environment can be created using the apigee-provision utility:

```
/opt/apigee/apigee-service/bin/
apigee-service apigee-provision
setup-org -f configuration_file
```

- Organization names must be unique.
- Environment names must be unique within the organization.
- At least one organization and one environment must exist.

Sample configuration file:

```
IP1=<node 1>
MSIP="$IP1"
ADMIN_EMAIL="<email@example.com>"
APIGEE_ADMINPW="<password>"
NEW_USER="y"
USER_NAME="orgadmin@apigee.com"
FIRST_NAME="OrgAdminName"
LAST_NAME="OrgAdminLastName"
USER_PWD="<password>"
ORG_NAME="<orgname>"
ORG_ADMIN="<user_name>"
ENV_NAME="prod"
VHOST_PORT="9001"
VHOST_NAME="default"
VHOST_ALIAS="<virtualhostalias>"
USE_ALL_MPS="y"
```

To create an organization and environment, use the apigee-provision utility.

The utility uses the same configuration file mechanism as that used to install the Apigee platform components.

At least one organization and one environment must exist before you can start using Apigee Edge for private cloud.

Note that organization names must be unique, and environment names must be unique within the organization.

Agenda

Installation planning

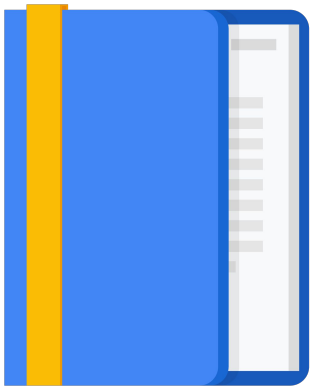
Installation dependencies

Installation process

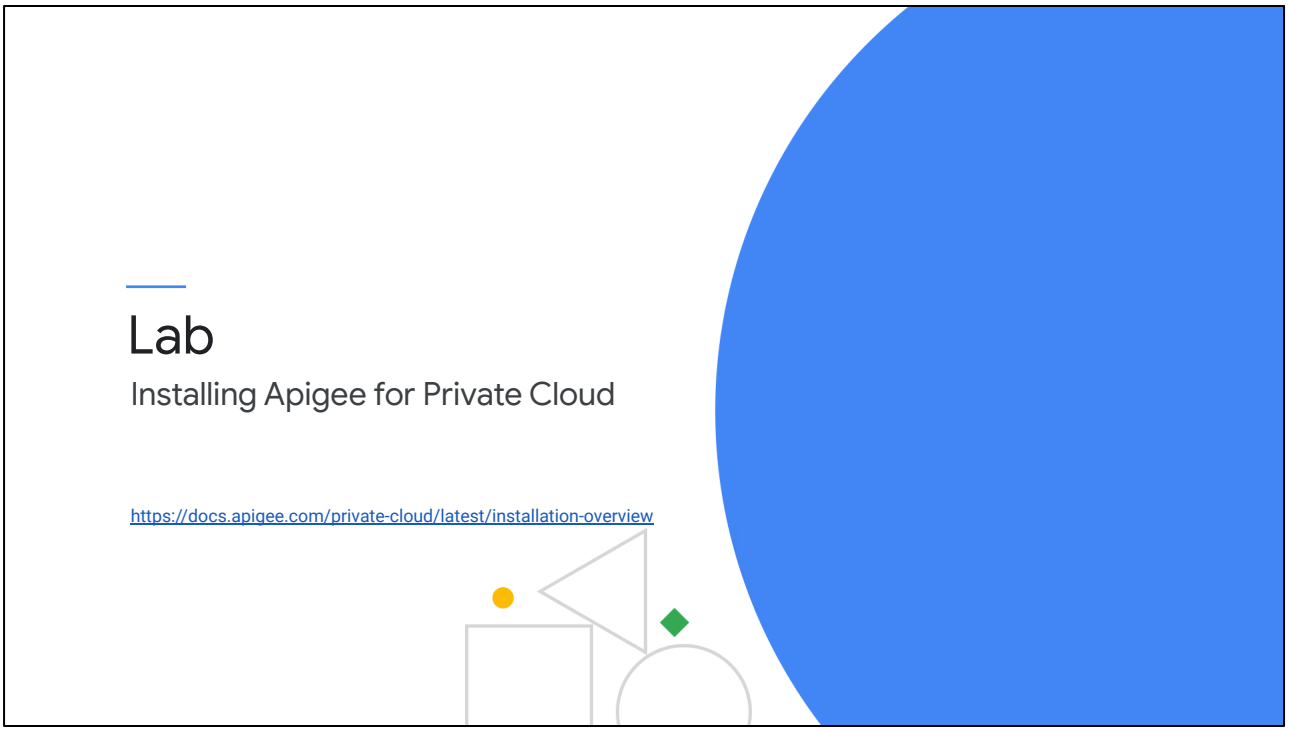
Lab

Installing the new Edge UI

Maintenance and uninstall process



We will now complete a lab to install the Apigee Edge platform for private cloud.



The lab takes you through the process of installing the Apigee for private cloud components on Google Compute Engine.

It uses the same process and utilities that you use when performing customer installations of the platform.

You can get more details of the installation process at the Apigee for private cloud documentation website.

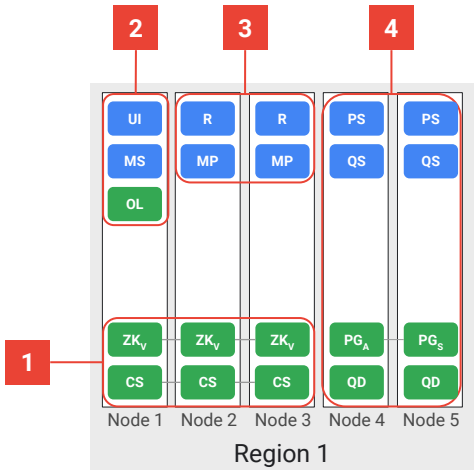
Edge installation and configuration lab: Scope

Scope:

- Single region
- 5-node Apigee Edge install
- VMs instances running CentOS
- Installation using silent install process

Steps overview:

1. Bootstrapping
2. Installing components
3. Creating the organization and environment
4. Validating the installation



In this lab you use a 5-node installation of Apigee Edge for private cloud in a single region.

The high level steps include bootstrapping, installing the Apigee components, creating the Apigee organization and environment and validating the installation.

The bootstrap process can be completed on all nodes at the same time, while the components are installed one node at a time.

Bootstrapping

On all nodes, download and execute the Edge bootstrap script:

```
curl -s https://software.apigee.com/bootstrap_4.19.06.sh
-o /tmp/apigee/bootstrap_4.19.06.sh
```

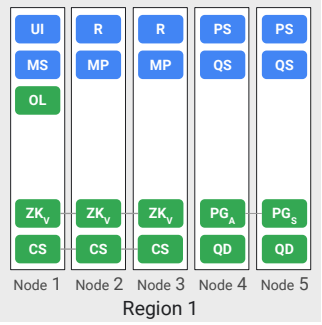
Verification:

```
cat /tmp/apigee/bootstrap_4.19.06.sh
```

```
sudo bash /tmp/apigee/bootstrap_4.19.06.sh
apigeeuser=<username> apigeepassword=<password>
```

Verification:

```
sudo cat /etc/yum.repos.d/apigee.repo
sudo yum -v repolist 'apigee*'
sudo rpm -qa | egrep "(apigee-|edge-)"
```



The first step in the install process is to download and run the bootstrap script for the Apigee private cloud version to be installed, on all nodes in the cluster.

Each Apigee Edge release version has it's own bootstrap script.

In our lab we use software.apigee.com as the Apigee private cloud repository from where the script is downloaded.

Access to the repository is not public, so you will need credentials that will be provided as part of the lab. The credentials are located on a VM in a text file located in the /tmp/apigee directory.

After successful execution of the bootstrap script, you should have a configured YUM repository and the apigee-service utility installed.

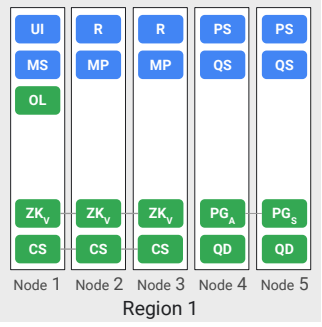
Bootstrapping

On all nodes, install the apigee-setup utility:

```
/opt/apigee/apigee-service/bin/apigee-service apigee-setup install
```

Verification:

```
ls /opt/apigee/apigee-setup
```



The next step is to install the apigee-setup utility on all the nodes in the cluster.

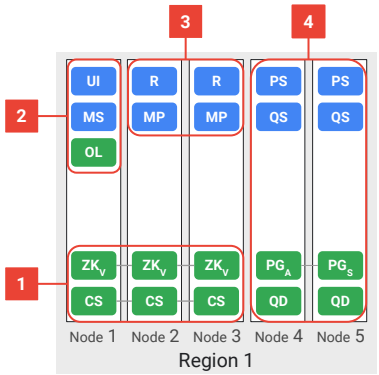
This is achieved with the help of the apigee-service tool.

To verify that apigee-setup was installed, check if the /opt/apigee/apigee-setup directory exists and inspect its contents.

Components setup

Following the order specified on the diagram, one node at the time, install the components using Apigee profiles:

1. `/opt/apigee/apigee-setup/bin/setup.sh -p ds -f /tmp/apigee/edge-configuration.txt`
2. `/opt/apigee/apigee-setup/bin/setup.sh -p ms -f /tmp/apigee/edge-configuration.txt`
3. `/opt/apigee/apigee-setup/bin/setup.sh -p rmp -f /tmp/apigee/edge-configuration.txt`
4. `/opt/apigee/apigee-setup/bin/setup.sh -p sax -f /tmp/apigee/edge-configuration.txt`



Next, we install the components on the corresponding nodes by executing the setup command with the correct profile.

The syntax of the setup command is **setup.sh -p**, the profile you wish to apply, **-f**, and the path to your configuration file.

The first profile that is installed is data storage services that include the ZooKeeper and Cassandra components.
To install the components, run the setup script on nodes one through three using the DS profile.

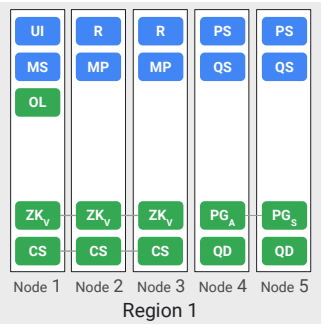
The next profile that is installed is management services that include the management server, Openldap and UI components.
To install the components, run the setup script on node one using the MS profile.

Next, we install the gateway services profile that includes the router and message processor components.
To install the components, run the setup script on nodes two and three using the RMP profile.

The final profile that is installed is analytics services that include the Qpid server, Qpid daemon, Postgres server and PostgreSQL components.
To install the components, run the setup script on nodes four and five using the standalone analytics or SAX profile.

Components setup verification

- Components status:
  `/opt/apigee/apigee-service/bin/apigee-all status`
- Cassandra status:
  `/opt/apigee/apigee-cassandra/bin/nodetool ring`
- ZooKeeper:
  `/opt/apigee/apigee-zookeeper/bin/zkServer.sh status`
- Postgres active:
  `/opt/apigee/apigee-service/bin/apigee-service apigee-postgresql postgres-check-master`
- Postgres standby:
  `/opt/apigee/apigee-service/bin/apigee-service apigee-postgresql postgres-check-standby`



After the setup is complete, you can check the status of the processes using the apigee-service or apigee-all utilities.

apigee-service enables you to execute commands on a component. One such commands is: status

apigee-all is a wrapper script that runs apigee-service followed by the <command>, on all the installed components on a node.

The Cassandra database has a specific tool to check the status of the ring, called “nodetool”.

To check the status of a ZooKeeper ensemble, you can use the provided helper script zkServer.sh.

To check the status of Postgres database replication, you can use apigee-service commands.

Organization and environment creation

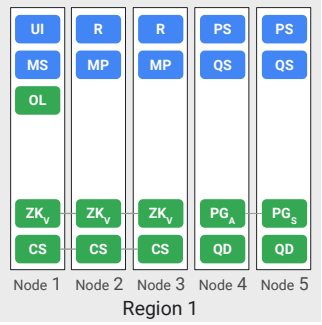
On Management Server node 1:

Install the apigee-provision utility:

```
/opt/apigee/apigee-service/bin/apigee-service
apigee-provision install
```

Create the organization and environment:

```
/opt/apigee/apigee-service/bin/apigee-service
apigee-provision setup-org -f
/tmp/apigee/edge-configuration-setup-org.txt
```



At this point, all of our Apigee Edge services are installed, but we still need to create an organization and environment in order to use the platform.

For this lab, we are going to create an organization named traininglab, with an environment named prod. As with the profile setup process, we will use a configuration file to describe our configuration settings.

The configuration file is supplied to a utility that does the work of provisioning the organization and environment for us.

We use the Apigee service utility to install the Apigee provision utility from the Apigee software repository.
The command is, **apigee-service apigee-provision install**.

After the provisioning utility is installed, run the command to set up the organization: **apigee-service apigee-provision setup-org -f**, and provide the path to your configuration file.

Organization and environment verification

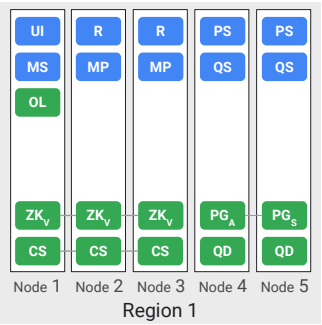
On Management Server node 1:

Verification:

Login in UI: `http://<ui-public-ip>:9000/login`

Use the username and password (USER_NAME and USER_PWD values) for the organization administrator specified in the organization setup configuration file:

`/tmp/apigee/edge-configuration-setup-org.txt`



At this point you have a fully functional installation of Apigee Edge for private cloud.

The organization and environment that was created are ready to use, and you can start deploying APIs to the environment.

To verify the organization and environment, login to the Apigee Enterprise UI using a web browser and navigating to the login url using node 1's IP address and port 9000.

Log in with the sysadmin credentials that was provided in the setup configuration file. Once the login is successful, your are presented with the Apigee UI dashboard page.

In the next few sections we describe some recommended installation verification steps, and post-installation and recurring maintenance tasks that should be completed.

Onboarding verification

List users:

```
curl -u <adminEmail>:<adminPassword> http://<ms-private-ip>:8080/v1/users
```

List organizations:

```
curl -u <adminEmail>:<adminPassword>
http://<ms-private-ip>:8080/v1/organizations
```

Describe an organization:

```
curl -u <adminEmail>:<adminPassword>
http://<management-server-private-ip>:8080/v1/organizations/<orgname>
```

The Apigee organization is a security boundary that encapsulates API proxies, configurations, products, users and other Apigee objects.

To list the users created in your Apigee Edge instance, you can use the /v1/users management API.

To list all created organizations, use the the /v1/organizations management API.

To describe the attributes of a particular organization, use the the /v1/organizations management API and append the organization name to the url.

All of these API calls help to validate your Apigee private cloud installation.

Onboarding verification

Describe analytics provisioning:

```
curl -u <adminEmail>:<adminPassword>
http://<ms-private-ip>:8080/v1/organizations/traininglab/environments/prod
/provisioning/axstatus
```

Describe analytics tables:

```
psql -h /opt/apigee/var/run/apigee-postgresql -U apigee
apigee=# \d analytics."<orgname>.<envname>.fact"
apigee=# \q
```

To validate the provisioning of analytics groups we use the /axstatus management API call for the organization and environment that was provisioned.

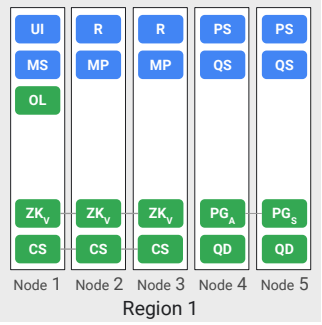
The API returns the members of the analytics group in the form of UUIDs of the Qpid servers and Postgres servers, and the status of the analytics group.

Another method to validate analytics provisioning is to login to the PostgreSQL database and describe the database schema.

apigee-validate

When executed, apigee-validate will:

- Create an organization.
- Create an environment for the organization.
- Associate an environment to message processors.
- Create a virtual host for the environment.
- Create and deploy an API model.
- Create and deploy an API proxy.
- And more ...



apigee-validate is a functional validation utility that enables you to execute a number of actions to validate the setup of an Apigee Edge installation.

Using apigee-validate you can perform functional tests by creating various entities in the platform.

The utility also has a clean up process to delete the Organization and associated entities once the functional validation is completed.

apigee-validate is able to perform a fully automated end to end functional validation cycle, expediting the process to validate the installation.

apigee-validate

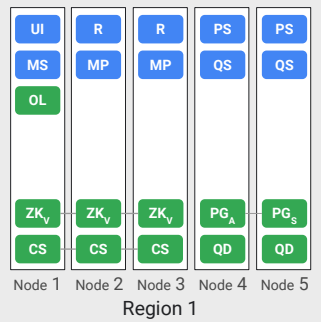
Validate the installation:

```
/opt/apigee/apigee-service/bin/apigee-service apigee-validate install
```

apigee-validate command:

```
/opt/apigee/apigee-service/bin/apigee-service
apigee-validate <action> -f <configuration-file>
```

- Offers two actions: **setup** and **clean**
- Uses the same configuration file mechanism as the installation process



To install apigee-validate, use the apigee-service utility with the apigee-validate install command.

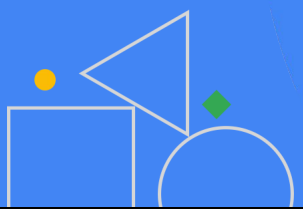
To run apigee-validate, use the apigee-service utility with the apigee-validate setup action and optionally supply a configuration configuration file

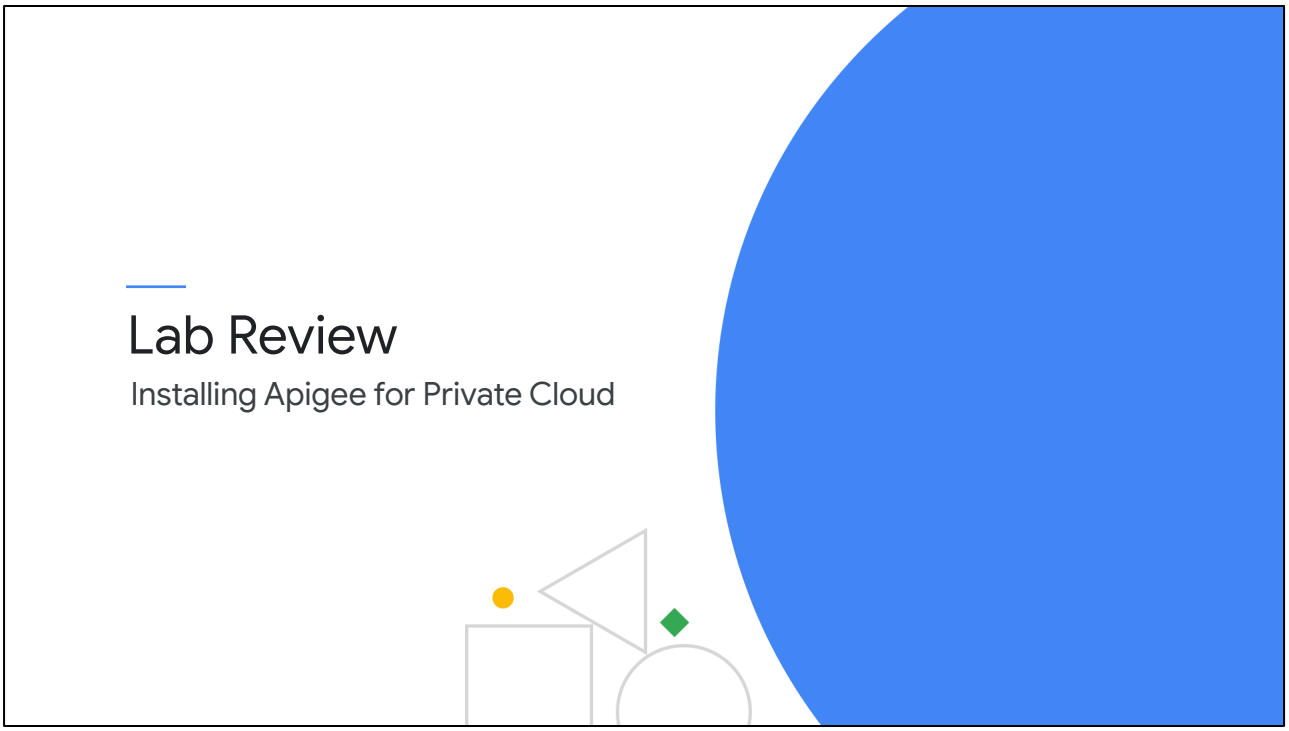
The setup action triggers the execution of the end-to-end functional test, resulting on the creation of an organization and related entities including the deployment of a test API proxy bundle.

The clean action deletes the test organization and all the related entities that were created with the setup action.

Lab

Installing Apigee for Private Cloud





In this lab you installed the Apigee private cloud platform on Google Compute Engine.

You first bootstrapped the process by downloading the various utilities that are required to perform the install.

You then installed the apigee service profiles on all the VMs used in the installation.

You also provisioned an Apigee organization and environment on the platform.

Finally, you validated the installation by performing standard onboarding tasks and ran the apigee validate utility to validate the setup of the platform.

By completing this lab, you learned about the various services and components that are used, and the process to install Apigee for private cloud.

Agenda

Installation planning

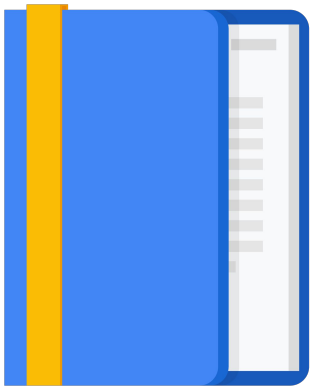
Installation dependencies

Installation process

Lab

Installing the new Edge UI

Maintenance and uninstall process



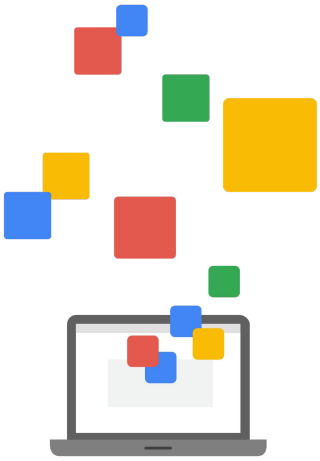
Apigee has released a major update to its API management user interface.

This New Edge user experience builds on top of the existing features of the Apigee Edge platform, and adds some enhancements.

We discuss the new Apigee Edge UI in this lesson.

Overview

- The new Edge UI is part of the general release of Apigee Edge for Private Cloud version 4.19.01.
- The new UI brings new features and enhancements to the API management platform and is fully backward compatible.
- Not all features are supported in the New Edge experience for Apigee Private Cloud.
- You must enable an external IDP to use the new Edge UI in Apigee for Private Cloud.
-



The new Edge UI works with your existing Apigee Edge for Private cloud installation version 4.19.01 or later.

To use the new UI, you must install and configure Apigee single sign-on (SSO).

Your existing API proxies and applications will work with the new UI, with no migration required.

Feature comparison

Feature	New Edge Experience	Classic Edge Experience
API proxy development	✚	✚
Analytics dashboards	✚	✚
Organization administration	✚	✚
Environment configuration	✚	✚
Portal content development using Drupal	✚	✚
Monetization	✚	✚
Proxy creation with OpenAPI v3	✚	✖
Create, edit, and delete virtual hosts	✚	✖
LDAP-based authentication	✖	✚

Most features in the current or classic Edge UI are also supported in the new Edge UI.

There are features like the support for OpenAPI specification v3 and the creating, editing and deleting of virtual hosts that are only supported in the new Edge UI.

Note that any new features and experiences will only be introduced in the new Edge UI.

Installation

Prerequisites

- Use a new node which meets the following requirements:
  - 4GB RAM / 2 CPU cores / 60GB disk space
  - Java 1.8
  - apigee-setup utility installed
- Configure the load balancer in front of the new UI to listen on the same port as the new UI component.
- Enable an external IDP by installing Apigee SSO.
- Create a configuration file with your settings.

The best approach to migrate from the classic UI to the new UI is to install the new UI on a new node.

Due to the specifics of the component, the load balancer in front of the Edge UI must listen on the same port as the new UI.

For example, if the new UI is listening on the default port 3001, then configure the load balancer to listen on port 3001. You can change the default port in the new UI configuration.

You must install and configure the Apigee SSO module as the new UI cannot use Apigee OpenLDAP for authentication.

Finally, you need to configure the new UI using properties that are defined in a configuration file.

Refer to the link provided for details on creating the configuration file for the new UI.

Example configuration file

```
    IP1=management_server_IP
    IP2=edge_UI_server_IP
    ADMIN_EMAIL=your@email.com
    APIGEE_ADMINPW=your_password
    APIGEE_PORT_HTTP_MS=8080
    MSIP=$IP1
    MS_SCHEME=http
    EDGEUI_ENABLE_UNIFIED_UI=y
    MANAGEMENT_UI_PORT=3001
    MANAGEMENT_UI_IP=$IP2
    MANAGEMENT_UI_APP_ENV=OPDK
    MANAGEMENT_UI_SCHEME=http
    MANAGEMENT_UI_PUBLIC_URIS=$MANAGEMENT_UI_SCHEME://$MA
NAGEMENT_UI_IP:$MANAGEMENT_UI_PORT
    MANAGEMENT_UI_SSO_REGISTERED_PUBLIC_URIS=$MANAGEMENT_
UI_PUBLIC_URIS
    MANAGEMENT_UI_SSO_CSRF_SECRET=YOUR_CSRF_SECRET
    MANAGEMENT_UI_SSO_CSRF_EXPIRATION_HOURS=24
    MANAGEMENT_UI_SSO_STRICT_TRANSPORT_SECURITY_AGE_HOURS
=8760
    MANAGEMENT_UI_SSO_ENABLED=y
    MANAGEMENT_UI_SSO_CLIENT_OVERWRITE=y
    MANAGEMENT_UI_SSO_CLIENT_ID=newueclient
    MANAGEMENT_UI_SSO_CLIENT_SECRET=your_client_ssosecret
    SHOEHORN_SCHEME=http
    SHOEHORN_IP=$MANAGEMENT_UI_IP
    SHOEHORN_PORT=9000
    CLASSIC_UI_IP=$MSIP
    CLASSIC_UI_PORT=9000
    CLASSIC_UI_SCHEME=http
    EDGEUI_PUBLIC_URIS=$CLASSIC_UI_SCHEME://$CLASSIC_UI_I
P:$CLASSIC_UI_PORT
    EDGEUI_SSO_REGISTERD_PUBLIC_URIS=$EDGEUI_PUBLIC_URIS
    EDGEUI_SSO_ENABLED=y
    EDGEUI_SSO_CLIENT_NAME=edgeui
    EDGEUI_SSO_CLIENT_SECRET=ssoClient123
    EDGEUI_SSO_CLIENT_OVERWRITE=y
    SSO_PUBLIC_URL_HOSTNAME=$IP1
    SSO_PUBLIC_URL_PORT=9099
    SSO_PUBLIC_URL_SCHEME=http
    MANAGEMENT_UI_SKIP_VERIFY=y
    SSO_ADMIN_NAME=ssoadmin
    SSO_ADMIN_SECRET=your_sso_admin_secret

    INSERT_IDP_CONFIG_BLOCK_HERE (SAML, LDAP direct, or
LDAP indirect)

    INSERT_SMTP_CONFIG_BLOCK_HERE
```

Here is a sample of the properties you must define in the new Edge UI configuration file.

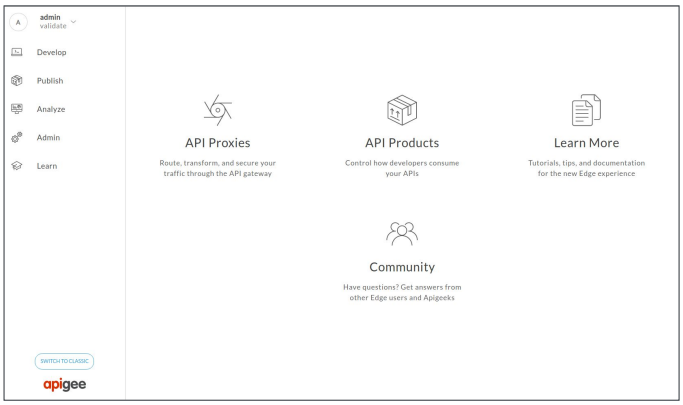
It contains all the information necessary to install and configure the new Edge UI and its shoehorn component.

For details on the configuration properties refer to the link provided on the previous slide.

Installation

Steps:

- Add a new node to your cluster.
- Download and install the apigee-setup utility.
- Execute the apigee-setup utility with the configuration file:
  `/opt/apigee/apigee-setup/bin/setup.sh -p ue -f configFile`
- Sign in to the new UI and verify its operation.



https://docs.apigee.com/private-cloud/v4.50.00/install-new-edge-ui#install-the-base-edge-ui-shoehorn

To install the new Edge UI, follow these steps:

First, prepare a new node for the installation.

Next, download and install the the apigee-setup utility for your version of Apigee Edge for private cloud.

Execute the setup utility using the user experience or ue profile and provide the path to the new UI configuration file.

Note that the utility installs both the new Edge UI and it's internal component called shoehorn.

For more details on the installation process review the documentation at the link provided.

Agenda

Installation planning

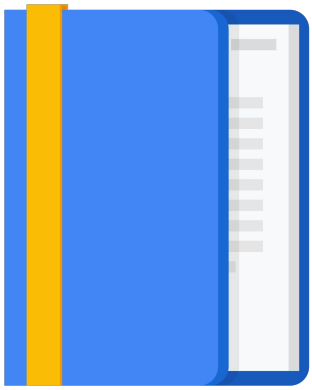
Installation dependencies

Installation process

Lab

Installing the new Edge UI

Maintenance and uninstall process



In this lesson, we will discuss post installation activities on the apigee private cloud platform.

Over the past few lessons, we completed the installation process so now it's time to perform a few additional maintenance tasks.

We will also discuss the process to uninstall Apigee for private cloud.

Setting server autostart

Run the following commands to enable/disable autostart on any node or individual component:

```
/opt/apigee/apigee-service/bin/apigee-all enable_autostart
/opt/apigee/apigee-service/bin/apigee-all disable_autostart
```

For a specific component:

```
/opt/apigee/apigee-service/bin/apigee-service <component>
enable_autostart
```

```
/opt/apigee/apigee-service/bin/apigee-service <component>
disable_autostart
```

After installation, you may wish to set apigee services to start when the host boots. This is not done for you automatically because it may not be necessary in every environment.

To enable service autostart, run apigee all enable_autostart.

To disable it, run apigee all disable_autostart.

You can do the same for individual Apigee components using the apigee service enable_autostart or disable_autostart commands and providing the component type.

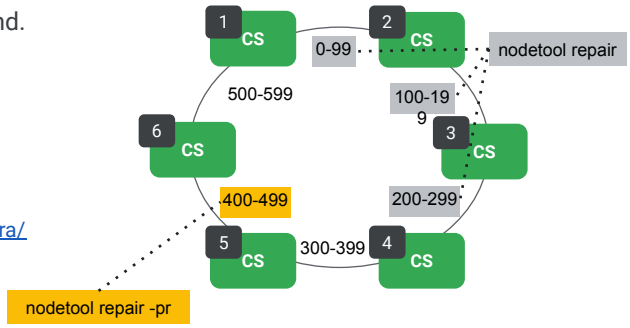
Cassandra: Anti-entropy maintenance

The Apache Cassandra ring requires periodic maintenance to ensure consistency across all nodes.

To perform this maintenance, use the Cassandra "nodetool repair" command.

```
nodetool -h <host> repair -pr
```

https://docs.datastax.com/en/archived/cassandra/2.1/cassandra/dml/dml_about_deletes_c.html



There are two maintenance activities that you will need to perform on the Apigee private cloud platform.

The first activity, is to run the anti entropy repair command on all Cassandra hosts once per week.

Executing nodetool -h localhost repair, will trigger a repair on the ring, with all nodes engaging in this process.

Since the command will place some amount of load on the Cassandra cluster, it is recommended to execute repairs on one node at a time at different hours or on different days in production environments.

Using repair with the partition range or -pr option is considered the best practice. Cassandra repair should be scheduled as a cron job.

This maintenance must be run on every Cassandra node to eliminate problems related to Cassandra "forgotten deletes".

For more information on "forgotten deletes" and Cassandra consistency, and for instructions on how to use nodetool, view the documentation at the link provided.

PostgreSQL: Pruning analytics data

The following command can be used to prune analytics data for a specific organization and environment:

```
/opt/apigee/apigee-service/bin/apigee-service apigee-postgresql
pg-data-purge <org_name> <env_name> <number_of_days_to_retain>
```

The second activity, is to purge detailed analytics data from the analytics database nightly.

As the amount of analytics data increases, you may need to "prune" the data that exists beyond your required data retention period.

This command queries the "childfactables" table in the "analytics" schema to determine which raw data partitions cover the dates for which data pruning is to be performed, and drops those tables.

Once the tables are dropped, the entries in "childfactables" related to those partitions are deleted.

Childfactables are daily-partitioned fact data. Every day new partitions are created and data gets ingested into the daily partitioned tables. At a later point in time, when the old fact data are no longer required, you can purge the corresponding childfactables.

Note that If you have configured active-standby replication between PostgreSQL servers, then you only need to run this script on the Postgres active database.

Backup

- Apigee Edge and API proxies may produce dynamically generated data and configuration (e.g., keys, UUIDs).
  - This data should be backed up on a regular basis.
- Backups are important in scenarios where a complete restore of the platform is required.
- Apigee provides utilities to perform backup and restore of Apigee private cloud components.

After the installation completes, you should setup backup jobs for each host.

Backups play an important role in scenarios where a complete restore of the platform is required.

Apigee Edge provides utilities to perform backup and restore operations. The commands are described in the management module of the On Premises Management, Security, and Upgrade course.

You may also want to set up TLS to secure the Apigee components running on the platform or set up user authentication with an external identity provider.

These topics are also fully discussed in the security module of the above mentioned course.

Uninstalling components

- It is possible to uninstall individual Apigee components.
- Apigee components function as part of an application cluster.
- Before uninstalling a component, you should un-assign any resources that the component has been assigned to.

Apigee components can be added and removed at any given time without causing downtime for the service, if the correct procedure is followed.

Note that the apigee components are not standalone, but function as part of an application cluster.

It is a best practice to unassign the components from the resources they've been assigned to (for example unassign a message processor from all environments it's been assigned to) before uninstalling and removing the component.

Uninstalling components

To uninstall a component, use the apigee-service utility:

```
/opt/apigee/apigee-service/bin/apigee-service <component> uninstall
```

For example, to uninstall the Edge UI:

```
/opt/apigee/apigee-service/bin/apigee-service edge-ui uninstall
```

To uninstall all Apigee components on the node, uninstall the apigee-service utility:

```
/opt/apigee/apigee-service/bin/apigee-service apigee-service uninstall
```

This command does not delete any data or log files; it only deletes the components.

If you need to uninstall a component, run the apigee-service utility with the uninstall action and provide the name of the component to uninstall.

The command leaves the local configuration and data for the service, but removes the binaries and scripts that were installed from the RPM package.

Uninstalling Apigee

To remove all components, data, RPMs, and the user created by the installation process, execute the following command on all nodes:

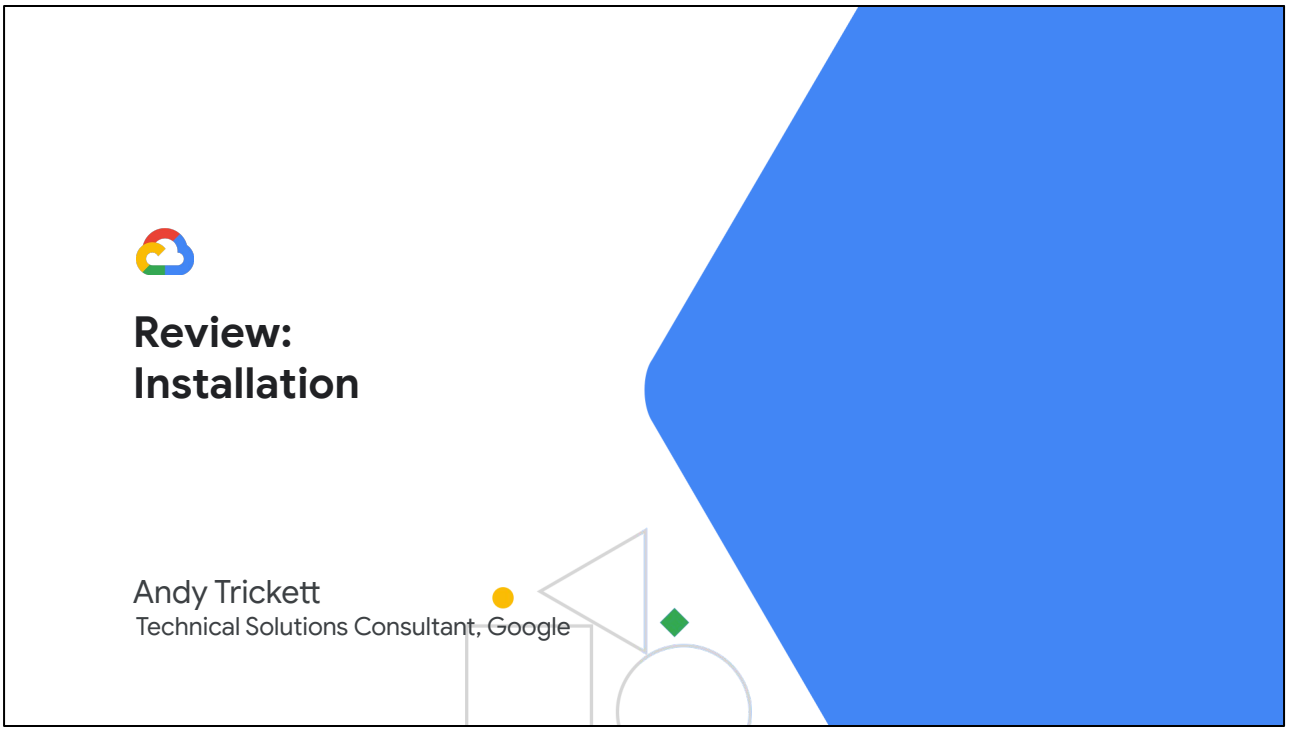
```
/opt/apigee/apigee-service/bin/apigee-all stop
sudo yum clean all
sudo rpm -e $(rpm -qa | egrep "(apigee-|edge-)")
sudo rm -rf /opt/apigee
sudo rm -rf /opt/nginx
/usr/sbin/userdel -r apigee
```

This example applies to CentOS 7.x.

Commands may vary between Linux distribution and versions.

To completely remove apigee from a host, first stop all apigee services using the apigee-service utility with the apigee-all stop command.

Then, remove all apigee and related packages using yum, remove the apigee installation directories, and finally delete the apigee user and group.



In this module, you will learned how to install the Apigee API platform for private cloud.

We discussed the planning activities and installation dependencies required to install the platform.

You learned the installation process and completed a lab to install Apigee Edge for private cloud on Google Compute Engine.

You also learned how to install the new Edge UI.

Finally, we discussed post installation maintenance activities and how to uninstall the Apigee private cloud platform.


Google Cloud