Apigee™

Apigee Edge for Private Cloud

v4.16.05

May 31, 2016

Install and Configuration Guide

Contact Information

INDIA

No.17/2, 2B Cross, 7th Main, 2 & 3 Floor, Off 80 Feet Road, 3rd Block Koramangala, Bangalore 560034

Call +91 80 67696800

www.apigee.com

USA

10 Almaden Boulevard, 16th Floor, San Jose CA 95113

Call +1 (408) 343-7300

www.apigee.com

UK

3 Sheldon Square
London W2 6HY
Call: +44 (0) 750 123 2390

www.apigee.com/

Contents

Overview

This document provides an overview of the Apigee Edge for Private Cloud installation. The full document is primarily divided into two parts:

- *Architectural Overview* — the system architecture and an overview of the installation process and requirements.
- *Installation* — outline of the steps needed to install and initially configure a custom deployment of Apigee Edge for Private Cloud.

This version of this document has details specific to **version 4.16.05**. Any references that are specific to previous versions are oversights and should be reported as bugs.

What's New

See the Apigee Edge for Private Cloud release notes for this product version:

http://apigee.com/docs/release-notes/content/apigee-edge-release-notes

Access the Apigee Community

The Apigee Community is a free resource where you can contact Apigee as well as other Apigee customers with questions, tips, and other issues. Before posting to the community, be sure to first search existing posts to see if your question has already been answered.

Architectural Overview

Before installing Apigee Edge for Private Cloud, you should be familiar with the overall organization of Edge modules and software components.

Apigee Edge for Private Cloud

Apigee Edge for Private Cloud consists of the following modules:

- Apigee Edge Gateway (aka API Services)
- Apigee Edge Analytics
- Apigee API BaaS
- Apigee Edge Developer Channel
- Apigee Edge Monetization Services (aka Developer Services Monetization)

Note: Apigee Edge Developer Channel is not available for installation by the Edge for Private Cloud installer. Developer Channel is available for on-premises installation by a separate script. If you want to install Developer Channel, contact Apigee Support.

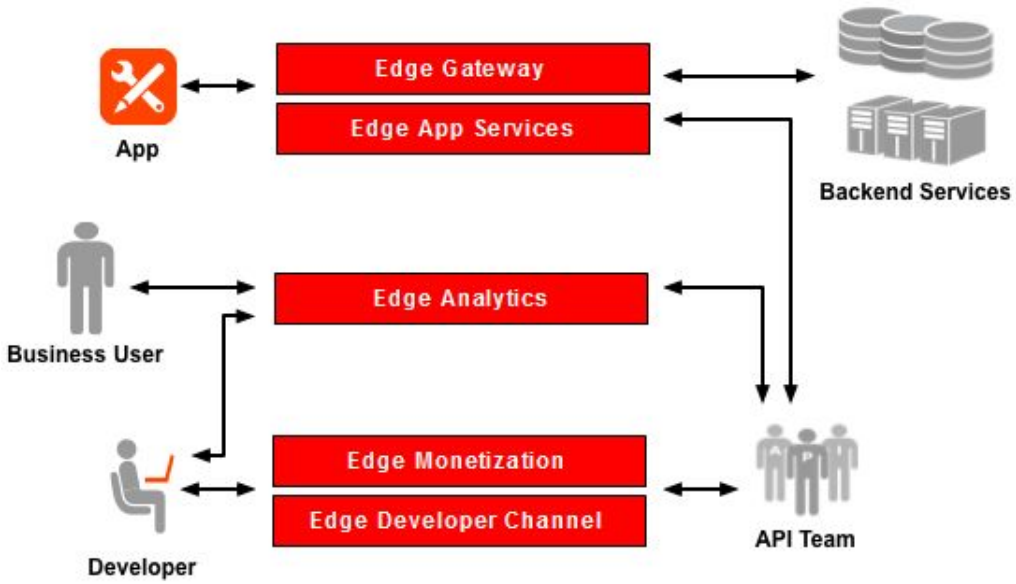


Figure 1: Apigee Edge for Private Cloud Architecture

Apigee Edge Gateway

Edge Gateway is the core module of Apigee Edge and is the main tool for managing your APIs. The Gateway UI provides tools for adding and configuring your APIs, setting up bundles of resources, and managing developers and apps. The Gateway offloads many common management concerns from your backend API. When you add an API, you can apply policies for security, rate-limiting, mediation, caching, and other controls.

You can also customize the behavior of your API by applying custom scripts, making call outs to third-party APIs, and so on.

Software Components

Edge Gateway is built from the following primary components:

- Edge Management Server
- Apache ZooKeeper
- Apache Cassandra
- Edge Router
- Edge Message Processor
- OpenLDAP
- Edge UI
- Play Framework

Edge Gateway is designed so that these may be all installed on a single host or distributed among several hosts.

Apigee Edge Analytics

Edge Analytics has powerful API analytics to see long-term usage trends. You can segment your audience by top developers and apps, learn about usage by API method to know where to invest, and create custom reports on business-level information.

As data passes through Apigee Edge, several default types of information are collected including URL, IP, user ID for API call information, latency, and error data. You can use policies to add other information, such as headers, query parameters, and portions of a request or response extracted from XML or JSON.

All data is pushed to Edge Analytics where it is maintained by the analytics server in the background. Data aggregation tools can be used to compile various built-in or custom reports.

Software Components

Edge Analytics comprises the following:

- Qpid, which consists of the following
  - Apache Qpid messaging system
  - Apigee Qpid Server service - A Java service from Apigee used to manage Apache Qpid
- Postgres, which consists of the following:
  - PostgreSQL database
  - Apigee Postgres Server service - A Java service from Apigee used to manage the PostgreSQL database

Apigee API BaaS

API BaaS is a complete backend as a service (BaaS) for powering mobile and Web apps that you install as an addition to Edge. API BaaS gives app developers access to a flexible data store and key differentiating features such as social graphs, geolocation, user management, push notifications, performance monitoring,

and more. API BaaS makes these features available with SDKs for iOS, Android, JavaScript, and others, letting app developers focus on creating the rich features and user experience that truly differentiate a client app rather than burning time implementing core backend services and infrastructure.

API BaaS Features

The Apigee documentation site has extensive information on API BaaS features. See http://apigee.com/docs/app-services/content/app-services-features (or http://apigee.com/docs/content/documentation-archives to find the docs that correspond to earlier versions of the product)

The following diagram illustrates how API BaaS components interact.

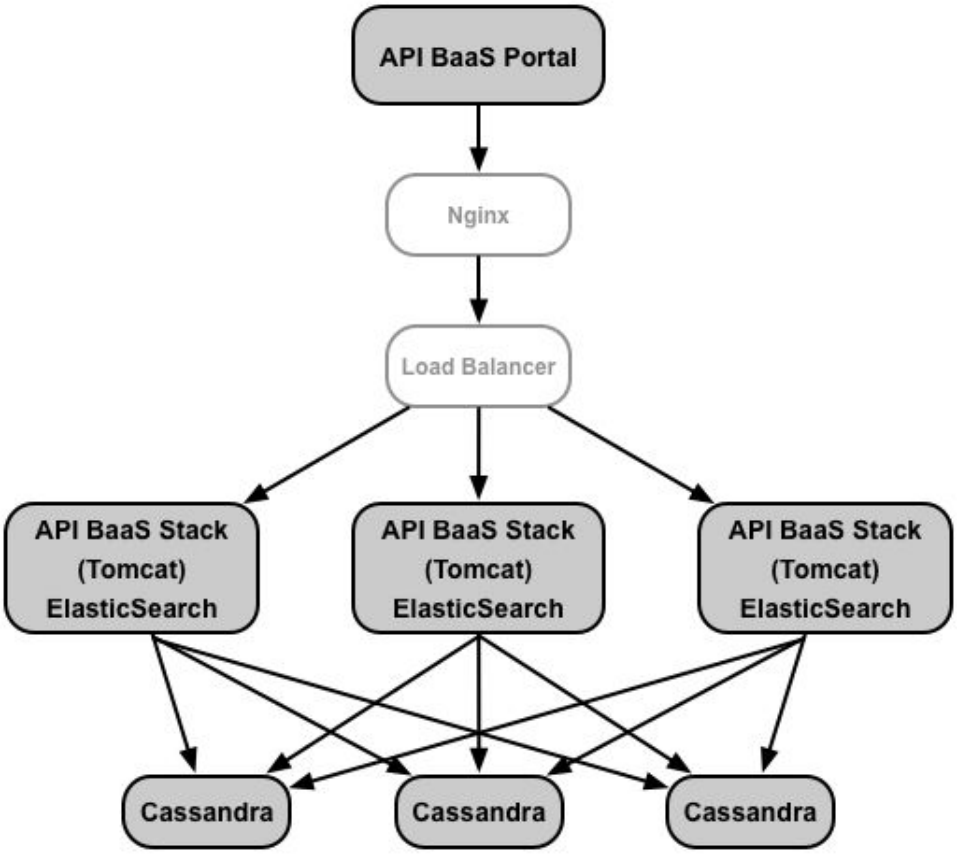


Figure 2: API BaaS Overview and Architecture

Software Components

API BaaS is built from the following primary components:

- API BaaS Stack - deployed in the Tomcat webserver
- API BaaS Portal - UI deployed in the Nginx web server
- ElasticSearch - distributed full-text search engine. ElasticSearch can be installed on the same node as API BaaS Stack, or on its own node.

You can scale the API BaaS REST API capability horizontally by adding Tomcat servers and using a Load Balancer to route web requests to all of your active servers.

For more information on getting started with API BaaS, see http://apigee.com/docs/content/build-apps-home (or http://apigee.com/docs/content/documentation-archives to find the docs that correspond to earlier versions of the product).

Apigee Edge Developer Channel

Edge Developer Channel is a template portal for content and community management. It is based on the open source Drupal (http://www.drupal.org) project. The default setup allows creating and managing API documentation, forums, and blogs. A built-in test console allows testing of APIs in real time from within the portal.

Apart from content management, Developer Channel has various features for community management such as manual/automatic user registration and moderating user comments. Role-Based Access Control (RBAC) model controls the access to features on the Developer Channel. For example, you can enable controls to allow registered user to create forum posts, use test consoles, and so on.

The Apigee Edge for Private Cloud deployment script does not include Developer Channel deployment. Developer Channel deployment on-premises is supported by its own installation script. If you want to install and configure Developer Channel, contact Apigee Support.

Apigee Edge Monetization Services

Edge Monetization Services is a new powerful extension to Apigee Edge for Private Cloud. As an API provider, you need an easy-to-use and flexible way to monetize your APIs so that you can generate revenue for the use of those APIs. Monetization Services solves those requirements. Using Monetization Services, you can create a variety of rate plans that charge developers for the use of your APIs bundled into packages. The solution offers an extensive degree of flexibility: you can create pre-paid plans, post-paid plans, fixed-fee plans, variable rate plans, “freemium” plans, plans tailored to specific developers, plans covering groups of developers, and more.

In addition, Monetization Services includes reporting and billing facilities. For example, as an API provider, you can get summary or detailed reports on traffic to your API packages for which developers purchased a rate plan. You can also make adjustments to these records as necessary. And you can create billing documents (which include applicable taxes) for the use of your API packages and publish those documents to developers.

You can also set limits to help control and monitor the performance of your API packages and allow you to react accordingly, and you can set up automatic notifications for when those limits are approached or reached.

Note: The core Apigee Edge (Gateway and Analytics) is a prerequisite for using Monetization Services.

Monetization Services Features

The key features of Edge Monetization Services include:

- Fully integrated with the API platform means real-time interaction

- Support all business models “out of the box” from simple fee-based plans to the most complex charging/revenue share plans (easy to create and modify plans)
- Rate transactions on volume or “custom attributes” within each transaction. Transaction can be made up of APIs from Gateway PLUS other systems (external to Apigee Edge)
- Automated tools such as limits and notifications to monitor performance and manage the process
- Integrated developer/partner workflow and controls to manage purchase through the billing/payment
- Fully self service for business users and developers/partners, so no need for costly technical intervention
- Integrated with any backend sales, accounting and ERP system

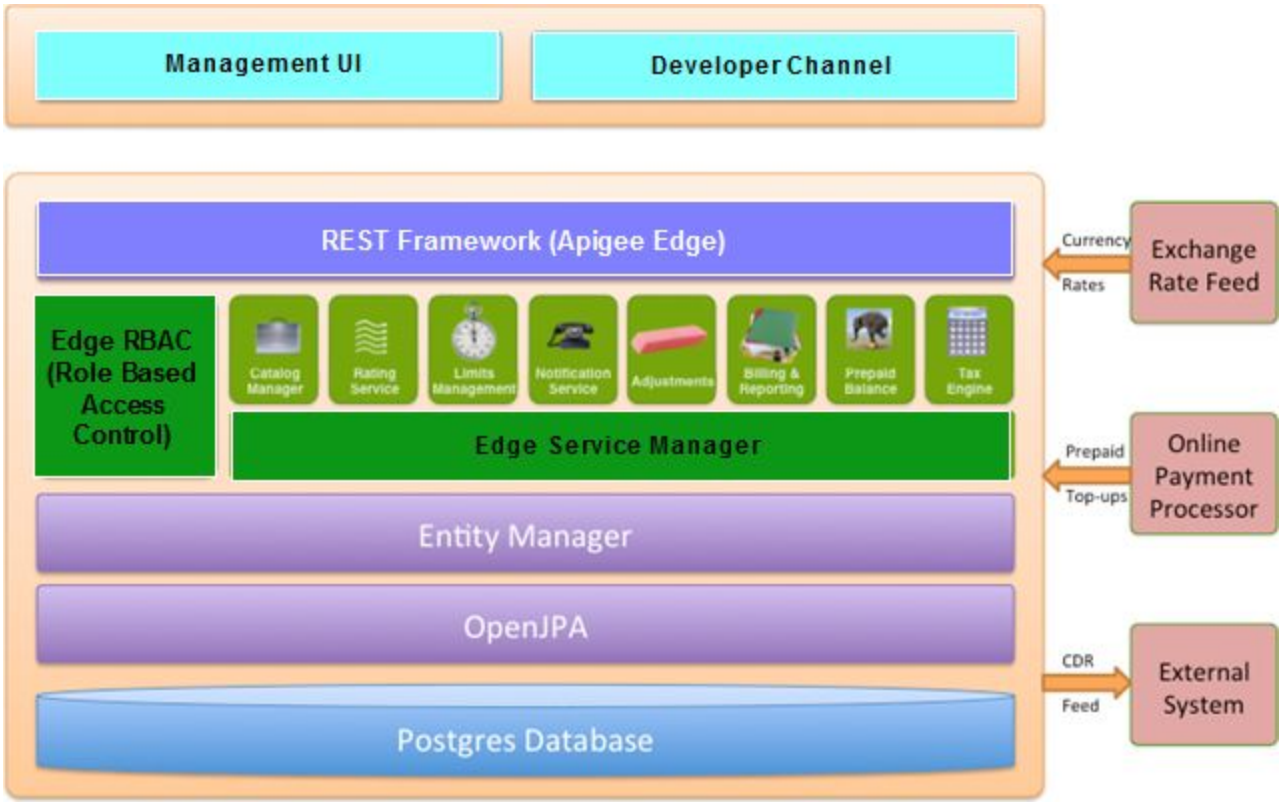


Figure 3: Edge Monetization Services Overview

Software Components

Edge Monetization Services is built on top of the following primary components:

- Edge Management Server
- Edge Message Processor

For more information on getting started with Monetization Services using Edge UI, see http://apigee.com/docs/monetization-services/content/get-started-using-monetization-services (or http://apigee.com/docs/content/documentation-archives to find the docs that correspond to earlier versions of the product).

If you wish to install Edge Monetization Services, see Installing Monetization Services.

On-Premises Deployment

An on-premises installation of core Apigee Edge for Private Cloud (Gateway and Analytics) provides the infrastructure required to run API traffic on behalf of the on-premises client's customers.

The components provided by the on-premises installation of Edge Gateway include (but are not limited to):

- A **Router** handles all incoming API traffic from a load balancer, determines the organization and environments for the API proxy that handles the request, balances requests across available Message Processors, and then dispatches the request. The Router terminates the HTTP request, handles the SSL traffic, and uses the virtual host name, port, and URI to steer requests to the appropriate Message Processor.
- A **Message Processor** processes API requests. The Message Processor evaluates an incoming request, executes any Apigee policies, and calls the back-end systems and other systems to retrieve data. Once those responses have been received, the Message Processor formats a response and returns it to the client.
- An **Apache Cassandra** is the runtime data repository that stores application configurations, distributed quota counters, API keys, and OAuth tokens for applications running on the gateway.
- An **Apache ZooKeeper** contains configuration data about the location and configuration of the various Apigee components, and notifies the different servers of configuration changes.
- An **OpenLDAP** (LDAP) to manage system and organization user and roles.
- A **Management Server** to hold these pieces together. The Management Server is the endpoint for Edge Management API requests. It also interacts with the Edge UI.
- A **UI** provides browser-based tooling that lets you perform most of the tasks necessary to create, configure, and manage API proxies, API products, apps, and users.

The components provided by the on-premises installation of Edge Analytics include:

- A **Qpid Server** manages queuing system for analytics data.
- A **Postgres Server** manages the PostgreSQL analytics database.

The following diagram illustrates how Apigee Edge components interact.

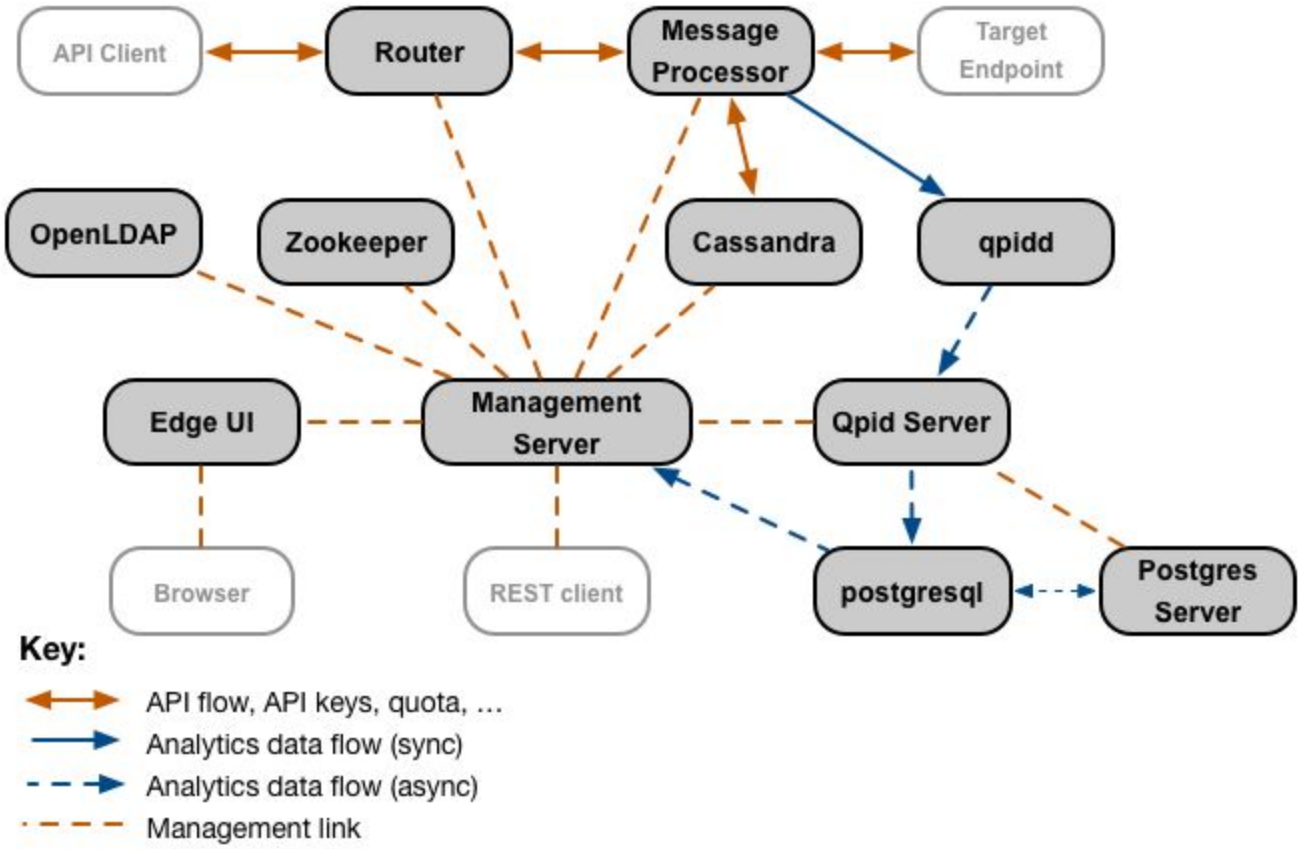


Figure 4: Conceptual Component Interactions

Installation Topologies and System Requirements

This section describes the Edge installation topologies and system requirements.

Installation topologies

The Installation Guide covers the following basic, on-premises installation scenarios. In addition to these, you have options to choose other scenarios by customizing these basic scenarios that best meet the requirements of your business.

1) **All-in-one Installation**: A single host runs all Edge components. Note that this configuration is only to be used for getting started with Edge or for initial prototyping. It is not to be used as a deployment or production environment.

Figure 5: All-in-one Setup

2) **Standalone installation (2-host, SA-SAX):** In this scenario, a single host runs Gateway standalone servers and associated components — Apigee Management Server, Apache ZooKeeper, Apache Cassandra, OpenLDAP, Edge UI, Apigee Router, and Apigee Message Processor. The other host runs Analytics standalone components —Qpid Server and Postgres Server.

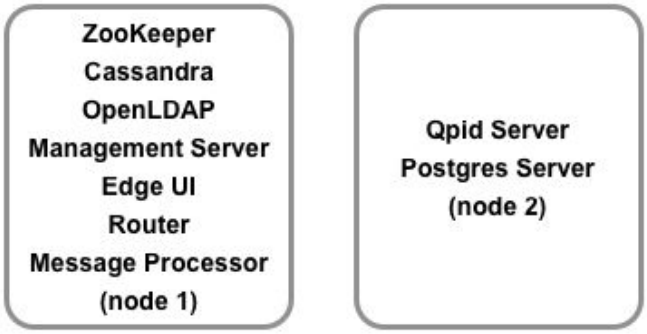


Figure 6: Standalone Setup

3) **5-host clustered installation (MIN HA-2SAX):** In this scenario, three hosts run ZooKeeper and Cassandra clusters. One of those three hosts also runs the Apigee Management Server, OpenLDAP, and Edge UI. Two of those three hosts also run Apigee Router + Message Processor. Two hosts run Apigee Analytics.

   Note: This scenario combines cluster and Gateway components to reduce the number of servers used. To achieve optimal performance, the cluster can also be deployed on three different servers. This scenario also introduces a master-standby replication between two Postgres nodes if analytics statistics are mission critical.

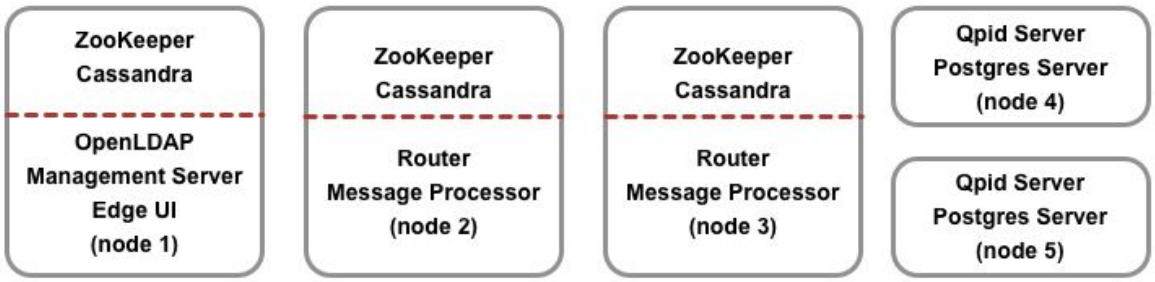


Figure 7: Five-host Clustered Setup

4) **9-host clustered installation (Performance HA Setup):** This scenario is similar to five-host clustered installation but has different Analytics components setup to achieve performance high availability.

   Note: This scenario introduces a master-standby replication between two Postgres nodes if Analytics statistics are mission critical.

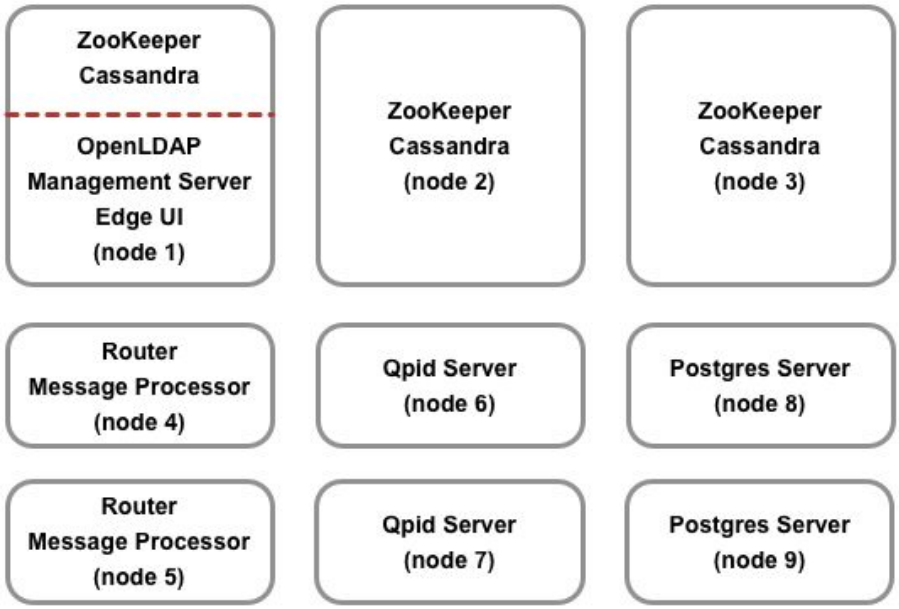


Figure 8: Nine-host Clustered Setup

5) **13-host clustered installation (Performance HA with separate data zone):** This scenario is an enhancement of nine-host clustered installation covering separate data zones for data and Apigee servers in one datacenter setup. Here LDAP is installed as an independent separate node.

   Note: This scenario uses master-master OpenLDAP replication and master-standby Postgres replication in one datacenter setup.

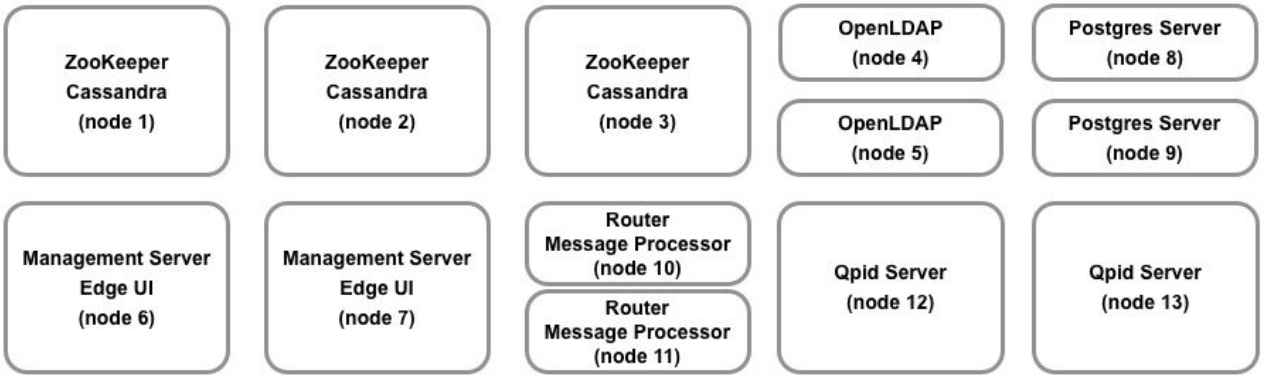


Figure 9: 13-host Clustered Setup

6) **12-host clustered installation (MIN API traffic DR / AX HA):** This scenario covers disaster recovery and analytics high availability across two datacenters using API-DN support. For more information on API-DN, see Appendix B: API-DN Support.

   Note: This scenario uses master-master OpenLDAP replication and master-standby Postgres replication (across two datacenters).

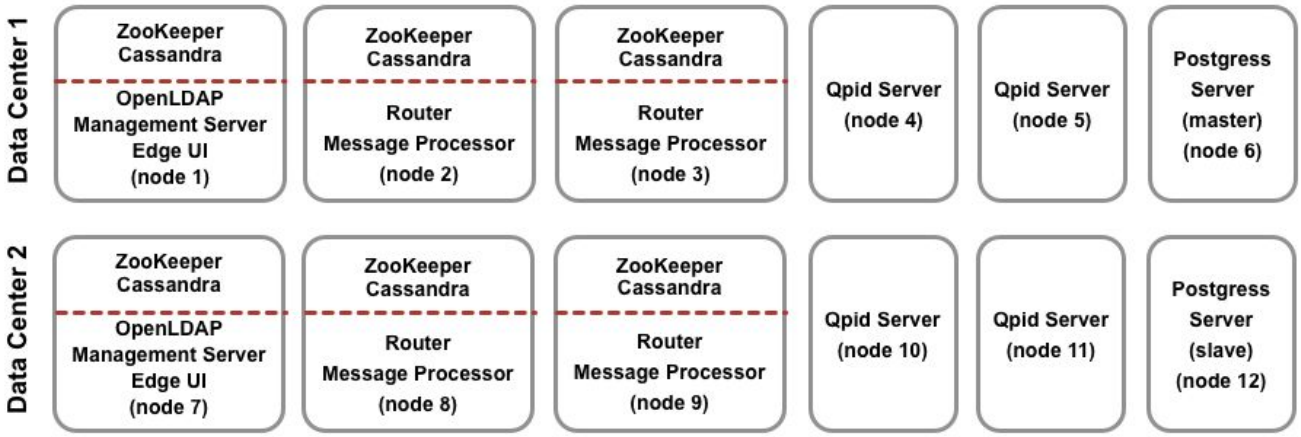


Figure 10: 12-host Clustered Setup

7) 7-host and 10-host API BaaS Installation: In this scenario, you install API BaaS on 10 hosts. The Cassandra nodes can be dedicated to API BaaS, or can be shared with Edge.

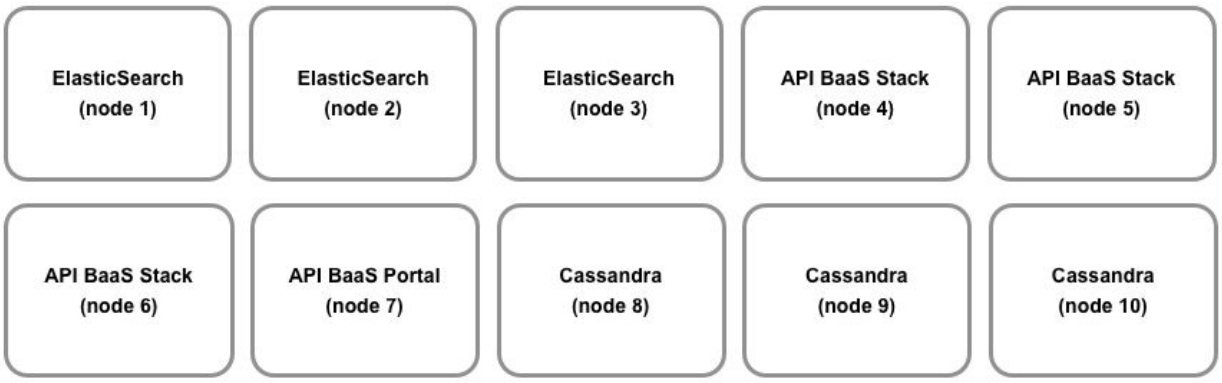


Figure 11: 10-host Clustered Setup

8) 7-host and 10-host API BaaS Installation: In this scenario, three hosts run the API BaaS Stack and ElasticSearch.

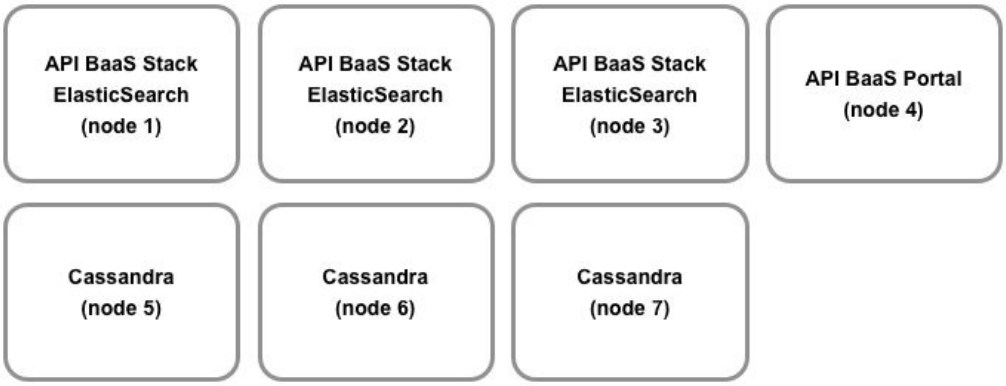


Figure 12: 7-host Clustered Setup

9) Installing Monetization Services: Monetization Services runs within any existing Apigee Edge setup. In this scenario, you install Monetization Services the Apigee Management Server and Message Processor. To install Monetization on Edge where the Edge installation has multiple Postgres nodes, the Postgres nodes must be configured in Master/Standby mode. You cannot install Monetization on Edge if you have multiple Postgres master nodes.

Installation Requirements

This section explains the requirements for Apigee Edge for Private Cloud installation.

Hardware Requirements

You must meet the basic hardware configurations that support the basic host installation. For all installation scenarios described above, the following tables list the minimum hardware requirements for the installation components.

In these tables the hard disk requirements are in addition to the hard disk space required by the operating system. Depending on your applications and network traffic, your installation might require more or fewer resources than listed below.

Installation Component	RAM	CPU	Minimum hard disk
Cassandra	16GB	8-core	250GB local storage with SSD or fast HDD supporting 2000 IOPS
Message Processor/Router on same machine	8/16GB	4/8-core	100GB
Analytics - Postgres/Qpid on same server (not recommended for production)	16GB*	8-core*	500GB - 1TB** local storage with SSD or fast HDD. For installations greater than 250 TPS (transactions per second), HDD with 1000 IOPS is recommended.
Analytics - Postgres standalone	16GB*	8-core*	500GB - 1TB** local storage with SSD or fast HDD supporting 2000 IOPS
Analytics - Qpid standalone	8GB	4-core	20GB - 500GB local storage with SSD or fast HDD For installations greater than 250 TPS, HDD with local storage supporting 1000 IOPS is recommended.

Other (OpenLDAP, UI, Management Server)	4GB	2-core	60GB
*Adjust Postgres system requirements based on throughput: ● Less than 250 TPS: 8GB, 4-core can be considered with local storage supporting 1000 IOPS ● Greater than 250 TPS: 16GB, 8-core, local storage supporting 1000 IOPS ● Greater than 1000 TPS: 16GB, 8-core, local storage supporting 2000 IOPS ● Greater than 2000 TPS: 32GB, 16-core, local storage supporting 2000 IOPS ● Greater than 4000 TPS: 64GB, 32-core, local storage supporting 4000 IOPS			
**The Postgres hard disk value is based on the out of the box analytics captured by Edge. If you add custom values to the analytics data, then these values should be increased accordingly. Use the following formula to estimate the required storage: (# bytes/request) * (requests per second) * (seconds per hour) * (hours of peak usage per day) * (days per month) * (months of data retention) = bytes of storage needed For example: (500 bytes of analytics data per request) * 100 req/sec * 3600 secs/hr * 18 hours peak usage per day * 30 days/month * 3 months retention = 291,600,000,000 bytes or 292 GB.			

In addition, the following lists the hardware requirements if you wish to install the Monetization Services:

Component with Monetization	**RAM**	**CPU**	**Hard disk**
Management Server (with Monetization Services)	8GB	4-core	60GB
Analytics - Postgres/Qpid on same server	16GB	8-core	500GB - 1TB with SSD or Fast HDD, or use the rule from the table above
Analytics - Postgres standalone	16GB	8-core	500GB - 1TB with SSD or Fast HDD, or use the rule from the table above
Analytics - Qpid standalone	8GB	4-core	40GB

The following lists the hardware requirements if you wish to install API BaaS:

API BaaS Component	RAM	CPU	Hard disk
ElasticSearch*	8GB	4-core	60 - 80GB
API BaaS Stack *	8GB	4-core	60 - 80GB
API BaaS Portal	1GB	2-core	20GB
Cassandra (Optional — typically you use the same Cassandra cluster for both Edge and API BaaS Services)	16GB	8-core	250GB local storage with SSD or fast HDD supporting 2000 IOPS
* You can install ElasticSearch and API BaaS Stack on the same node. If you do, configure ElasticSearch to use 4GB of memory (default). If ElasticSearch is installed on its own node, then configure it to use 6GB of memory.			

Note:

- If the root file system is not large enough for the installation, it is recommended to place the data onto a larger disk.
- If an older version of Apigee Edge for Private Cloud was installed on the machine, ensure that you delete the folder `/tmp/java` before a new installation.
- The system wide temporary folder `/tmp` needs execute permissions in order to start Cassandra.
- If user “apigee” was created prior to the installation, ensure that “/home/apigee” exists as home directory and is owned by “apigee:apigee”.

Operating System and third-party software requirements

These installation instructions and the supplied installation files have been tested on the operating systems and third-party software listed here: https://apigee.com/docs/api-services/reference/supported-software

Creating the apigee user

The installation procedure creates a Unix system user named 'apigee'. Edge directories and files are owned by 'apigee', as are Edge processes. That means Edge components run as the 'apigee' user. if necessary, you can run components as a different user. See Binding the Router to a protected port for an example.

Installation directory

By default, the installer writes all files to the `/opt/apigee` directory. You cannot change this directory location.

Note: While you cannot change this directory, you can create a symlink to map /opt/apigee to another location. See Creating a symlink from /opt/apigee for more information.

In the instructions in this guide, the installation directory is noted as /<inst_root>/apigee, where /<inst_root> is /opt by default.

Java

You need a supported version of Java1.8 installed on each machine prior to the installation. Supported JDKs are listed here:
https://apigee.com/docs/api-services/reference/supported-software

Ensure that JAVA_HOME points to the root of the JDK for the user performing the installation.

Network Setting

It is recommended to check the network setting prior to the installation. The installer expects that all machines have fixed IP addresses. Use the following commands to validate the setting:

- hostname returns the name of the machine
- hostname -i returns the IP address for the hostname that can be addressed from other machines.

Depending on your operating system type and version, you might have to edit /etc/hosts and /etc/sysconfig/network if the hostname is not set correctly. See the documentation for your specific operating system for more information.

Cassandra

All Cassandra nodes have to be connected to a ring.

Cassandra automatically adjusts its Java heap size based on the available memory. For more, see Tuning Java resources. In the event of a performance degradation or high memory consumption.

After installing the Edge for Private Cloud, you can check that Cassandra is configured correctly by examining the /<inst_root>/apigee/apigee-cassandra/conf/cassandra.yaml file. For example, ensure that the Edge for Private Cloud installation script set the following properties:

- cluster_name
- initial_token
- partitioner
- seeds
- listen_address
- rpc_address
- snitch

Warning: Do not edit this file.

PostgreSQL database

After you install Edge, you can adjust the following PostgreSQL database settings based on the amount of RAM available on your system:

```
conf_postgresql_shared_buffers = 35% of RAM      # min 128kB
conf_postgresql_effective_cache_size = 45% of RAM
conf_postgresql_work_mem = 512MB       # min 64kB
```

Note: These settings assume that the PostgreSQL database is only used for Edge analytics, and not for any other purpose.

To set these values:

1. Edit `postgresql.properties`:

   ```
   > vi /<inst_root>/apigee/customer/application/postgresql.properties
   ```

   If the file does not exist, create it.
2. Set the properties listed above.
3. Save your edits.
4. Restart the PostgreSQL database:

   ```
   > /<inst_root>/apigee/apigee-service/bin/apigee-service apigee-postgresql restart
   ```

jsvc

“jsvc” is a prerequisite for using API BaaS. Version 1.0.15-dev is installed when you install the API BaaS.

Network Security Services (NSS)

Network Security Services (NSS) is a set of libraries that supports development of security-enabled client and server applications. You should ensure that you have installed NSS v3.19, or later.

To check your current version:

```
> yum info nss
```

To update NSS:

```
> yum update nss
```

See this article from RedHat for more information.

AWS AMI

If you are installing Edge on an AWS Amazon Machine Image (AMI) for Red Hat Enterprise Linux 7.x, you must first run the following command:

```
> yum-config-manager --enable rhui-REGION-rhel-server-extras
rhui-REGION-rhel-server-optional
```

Tools

The installer uses the following UNIX tools in the standard version as provided by EL5 or EL6.

awk	dirname	ls	rpm	unzip
basename	echo	perl	rpm2cpio	useradd
bash	expr	pgrep (from procps)	sed	wc
bc	grep	ps	tar	yum
curl	hostname	pwd	tr	chkconfig
date	id	python	uname	sudo

Note:

- The executable for the tool ‘useradd’ is located in `/usr/sbin` and for chkconfig in `/sbin`.
- With sudo access you can gain access over the environment of the calling user, for example, usually one would call “`sudo <command>`” or “`sudo PATH=$PATH:/usr/sbin:/sbin <command>`”.
- Ensure that you have “patch” tool installed prior to a service pack (patch) installation.

ntpdate – It is recommended to have the servers time synchronized. If not already configured, ‘ntpdate’ utility could serve this purpose, which verifies whether servers are time synchronized. You can use “`yum install ntp`” to install the utility. This is particularly useful for replicating OpenLDAP setups. Note that you set up server time zone in UTC.

openldap 2.4 – The on-premises installation requires OpenLDAP 2.4. If your server has an Internet connection, then the Edge install script downloads and installs OpenLDAP. If your server does not have an Internet connection, you must ensure that OpenLDAP is already installed before running the Edge install script. On RHEL/CentOS, you can run "`yum install openldap-clients openldap-servers`" to install the OpenLDAP.

For 13-host installations, and 12-host installations with two Data Centers, you require OpenLDAP replication because there are multiple nodes hosting OpenLDAP.

Firewalls and Virtual Hosts

The term “virtual” commonly gets overloaded in the IT arena, and so it is with an Apigee Edge for Private Cloud deployment and *virtual hosts*. To clarify, there are two primary uses of the term “virtual”:

- **Virtual machines (VM):** *Not* required, but some deployment use VM technology to create isolated servers for their Apigee components. VM hosts, like physical hosts, can have network interfaces and firewalls. These installation instructions do not specifically support VM installations.
- **Virtual hosts:** Web endpoints, analogous to an Apache *virtual host*.

A router in a VM can expose multiple virtual hosts (as long as they differ from one another in their *host alias* or in their interface *port*).

Just as a naming example, a single physical server “A” might be running two VMs, named “VM1” and “VM2”. Let’s assume VM1 exposes a virtual Ethernet interface, which gets named `eth0` inside the VM, and which is assigned IP address `111.111.111.111` by the virtualization machinery or a network DHCP server; and then assume VM2 exposes a virtual Ethernet interface also named `eth0` and it gets assigned an IP address `111.111.111.222`.

We might have an Apigee router running in each of the two VMs. The routers expose virtual host endpoints as in this hypothetical example:

The Apigee router in VM1 exposes three virtual hosts on its `eth0` interface (which has some specific IP address), `api.mycompany.com:80`, `api.mycompany.com:443`, and `test.mycompany.com:80`.

The router in VM2 exposes `api.mycompany.com:80` (same name and port as exposed by VM1).

The physical host’s operating system might have a network firewall; if so, that firewall must be configured to pass TCP traffic bound for the ports being exposed on the virtualized interfaces (`111.111.111.111:{80, 443}` and `111.111.111.222:80`). In addition, each VM’s operating system may provide its own firewall on its `eth0` interface and these too must allow ports `80` and `443` traffic to connect.

The *basepath* is the third component involved in routing API calls to different API proxies that you may have deployed. API proxy bundles can share an endpoint if they have different basepaths. For example, one basepath can be defined as `http://api.mycompany.com:80/` and another defined as `http://api.mycompany.com:80/salesdemo`.

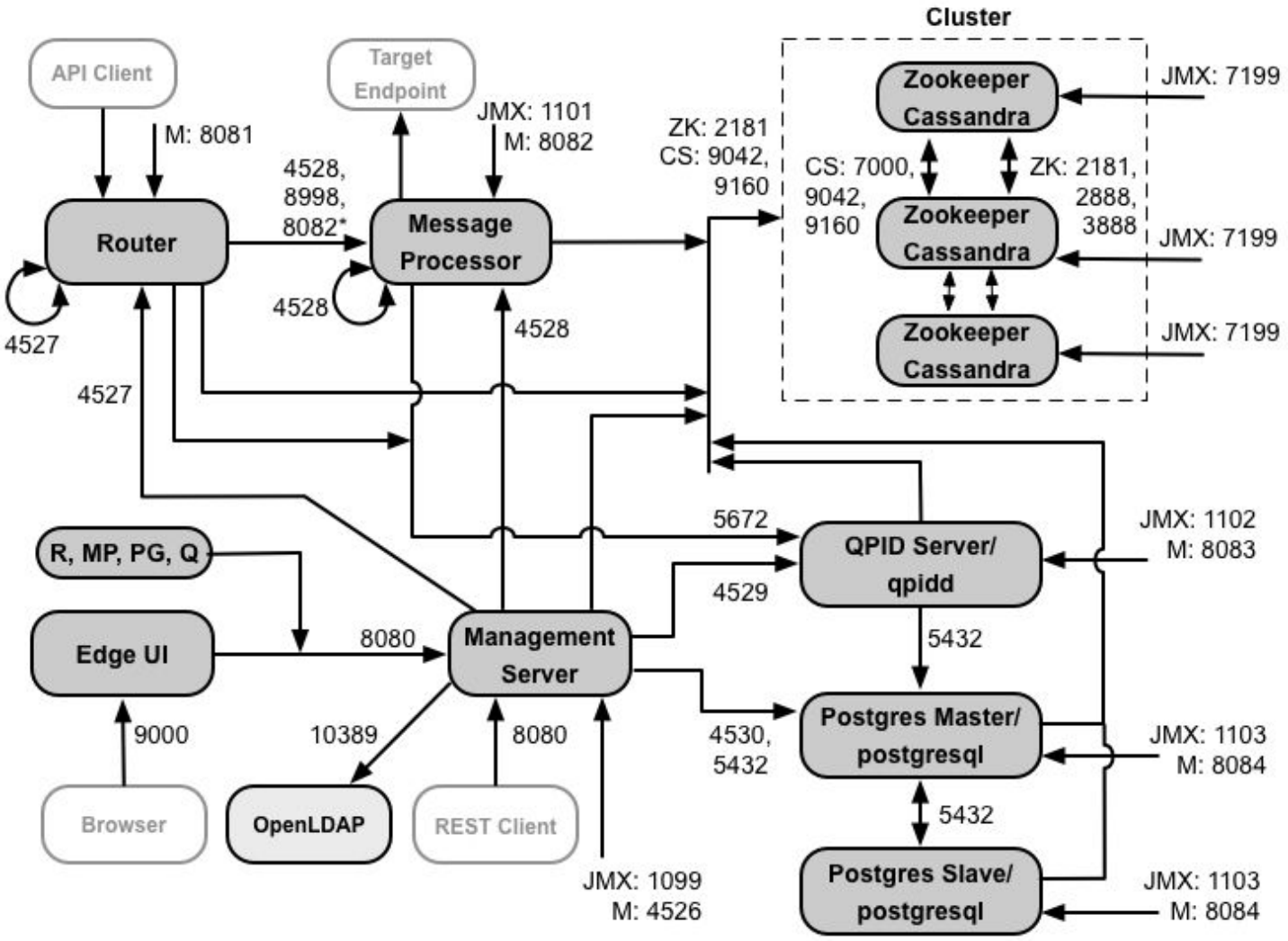
In this case, you need a load balancer or traffic director of some kind splitting the `http://api.mycompany.com:80/` traffic between the two IP addresses (`111.111.111.111` on VM1 and `111.111.111.222` on VM2). This function is specific to your particular installation, and is configured by your local networking group.

The basepath is set when you deploy an API. From the above example, you can deploy two APIs, `mycompany` and `testmycompany`, for the organization `mycompany-org` with the virtual host that has the host alias of `api.mycompany.com` and the port set to `80`. If you do not declare a basepath in the deployment, then the router does not know which API to send incoming requests to.

However, if you deploy the API `testmycompany` with the base URL of `/salesdemo`, then users access that API using `http://api.mycompany.com:80/salesdemo`. If you deploy your API `mycompany` with the base URL of `/` then your users access the API by the URL `http://api.mycompany.com:80/`.

Edge port requirements

The need to manage the firewall goes beyond just the virtual hosts; both VM and physical host firewalls must allow traffic for the ports required by the components to communicate with each other.
The following image shows the ports requirements for each Edge component:



Notes on this diagram:

- *Port 8082 on the Message Processor only has to be open for access by the Router when you configure TLS/SSL between the Router and Message Processor. If you do not configure TLS/SSL between the Router and Message Processor, the default configuration, port 8082 still must be open on the Message Processor to manage the component, but the Router does not require access to it.
- The ports prefixed by "M" are ports used to manage the component and must be open on the component.
- The following components require access to port 8080 on the Management Server: Router, Message Processor, UI, Postgres, and Qpid.
- A Message Processor must open port 4528 as its management port. If you have multiple Message Processors, they must all be able to access each other over port 4528 (indicated by the loop arrow in the diagram above for port 4528 on the Message Processor). If you have multiple Data Centers, the port must be accessible from all Message Processors in all Data Centers.

- While it is not required, you can open port 4527 on the Router for access by any Message Processor. Otherwise, you might see error messages in the Message Processor log files.
- A Router must open port 4527 as its management port. If you have multiple Routers, they must all be able to access each other over port 4527 (indicated by the loop arrow in the diagram above for port 4527 on the Router).
- Access to JMX ports can be configured to require a username/password. See http://docs.apigee.com/api-services/latest/how-monitor for more information.
- You can optionally configure SSL access for certain connections, which can use different ports. See SSL in the Apigee online documentation for more.
- You can optionally open ports on individual nodes to allow `ssh` access.
- You can configure the Management Server to send emails through an external SMTP server. If you do, you must ensure that the Management Server can access the necessary port on the SMTP server. See http://docs.apigee.com/api-services/latest/how-monitor for more information.

The table below shows the ports need to be opened in firewalls, by Edge component:

Component	Port	Description
Standard HTTP ports	80, 443	HTTP plus any other ports you use for virtual hosts
Management Server	8080	Port for Edge management API calls. These components require access to port 8080 on the Management Server: Router, Message Processor, UI, Postgres, and Qpid.
	1099	JMX port
	4526	For distributed cache and management calls
Management UI	9000	Port for browser access to management UI
Message Processor	8998	Message Processor port for communications from Router
	8082	Default management port for Message Processor. If you configure TLS/SSL between the Router and Message Processor, used by the Router to make health checks on the Message Processor.
	1101	JMX port
	4528	For distributed cache and management calls
Router	8081	Default management port for Router
	4527	For distributed cache and management calls

ZooKeeper	2181	Used by other components like Management Server, Router, Message Processor and so on
	2888, 3888	Used internally by ZooKeeper for ZooKeeper cluster (known as ZooKeeper ensemble) communication
Cassandra	7000, 9042, 9160	Apache Cassandra ports for communication between Cassandra nodes
	7199	JMX port
Qpid	5672	Used for communications from the Router and Message Processor to Qpid server
	8083	Default management port on Qpid server
	1102	JMX port
	4529	For distributed cache and management calls
Postgres	5432	Used for communication from Qpid/Management Server to Postgres
	8084	Default management port on Postgres server
	1103	JMX port
	4530	For distributed cache and management calls
LDAP	10389	OpenLDAP
SmartDocs	59002	The port on the Edge router where SmartDocs page requests are sent.
Note: In addition, you may need to open ports in the firewalls for testing. For example, 59001, and so on.		

The next table shows the same ports, listed numerically, with the source and destination components:

Port Number	Purpose	Source Component	Destination Component
<virtual host port#>	HTTP plus any other ports you use for virtual host API call traffic. Ports 80 and 443 are most commonly used; the Message Router can terminate SSL connections.	External client (or load balancer)	Listener on Message Router

1099 through 1103	JMX Management	JMX Client	Management Server (1099) Message Processor (1101) Qpid Server (1102) Postgres Server (1103)
2181	Zookeeper client communication	Management Server Router Message Processor Qpid Server Postgres Server	Zookeeper
2888 and 3888	Zookeeper internode management	Zookeeper	Zookeeper
4526 through 4530	RPC Management ports used for distributed cache and calls from the Management Servers to the other components	Management Server	Management Server (4526) Router (4527) Message Processor (4528) Qpid Server (4529) Postgres Server (4530)
4528	For distributed cache calls	Router Message Processor	Message Processor
5432	Postgres client	Qpid Server	Postgres
5672	Used for sending analytics from Router and Message Processor to Qpid	Router Message Processor	Qpid daemon
7000	Cassandra inter-node communications	Cassandra	Other Cassandra node
7199	JMX management	JMX client	Cassandra
8080	Management API port	Management API clients	Management Server
8081 through 8084	Component API ports, used for issuing API requests directly to individual components. Each	Management API clients	Router (8081) Message Processor (8082) Qpid Server (8083)

	component opens a different port; the exact port used depends on the configuration		Postgres Server (8084)
8998	Communication between router and message processor	Router	Message Processor
9000	Default Edge management UI port	Browser	Management UI Server
9042	CQL native transport	Router Message Processor Management Server	Cassandra
9160	Cassandra thrift client	Router Message Processor Management Server	Cassandra
10389	LDAP port	Management Server	ApcheDS/OpenLDAP
59002	The router port where SmartDocs page requests are sent	SmartDocs	Router

A Message Processor keeps a dedicated connection pool open to Cassandra, which is configured to never timeout. When a firewall is between a message processor and Cassandra server, the firewall can time out the connection. However, the message processor is not designed to reestablish connections to Cassandra.

To prevent this situation, Apigee recommends that the Cassandra server, message processor, and routers be in the same subnet so that a firewall is not involved in the deployment of these components.

If a firewall is between the router and message processors, and has an idle tcp timeout set, our recommendations is to:

1. Set `net.ipv4.tcp_keepalive_time = 1800` in sysctl settings on Linux OS, where 1800 should be lower than the firewall idle tcp timeout. This setting should keep the connection in an established state so that the firewall does not disconnect the connection.
2. On all Message Processors, edit `/<inst_root>/apigee/customer/application/message-processor.properties` to add the following property. If the file does not exist, create it.

   ```
   conf_system_casssandra.maxconnecttimeinmillis=-1
   ```

3. Restart the Message Processor:

```
> /opt/apigee/apigee-service/bin/apigee-service edge-message-processor
restart
```

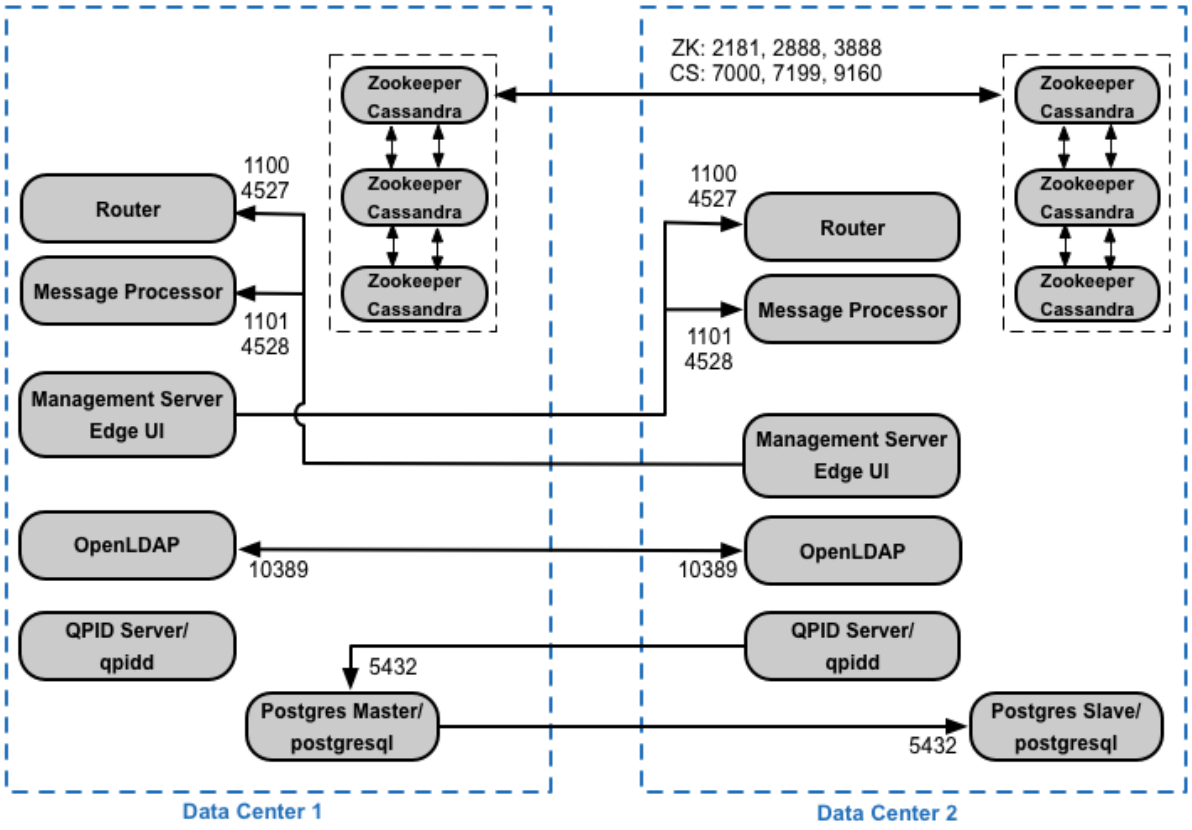
4. On all Routers, edit `/<inst_root>/apigee/customer/application/router.properties` to add the following property. If the file does not exist, create it.

```
conf_system_casssandra.maxconnecttimeinmillis=-1
```

5. Restart the Router:

```
> /opt/apigee/apigee-service/bin/apigee-service edge-router restart
```

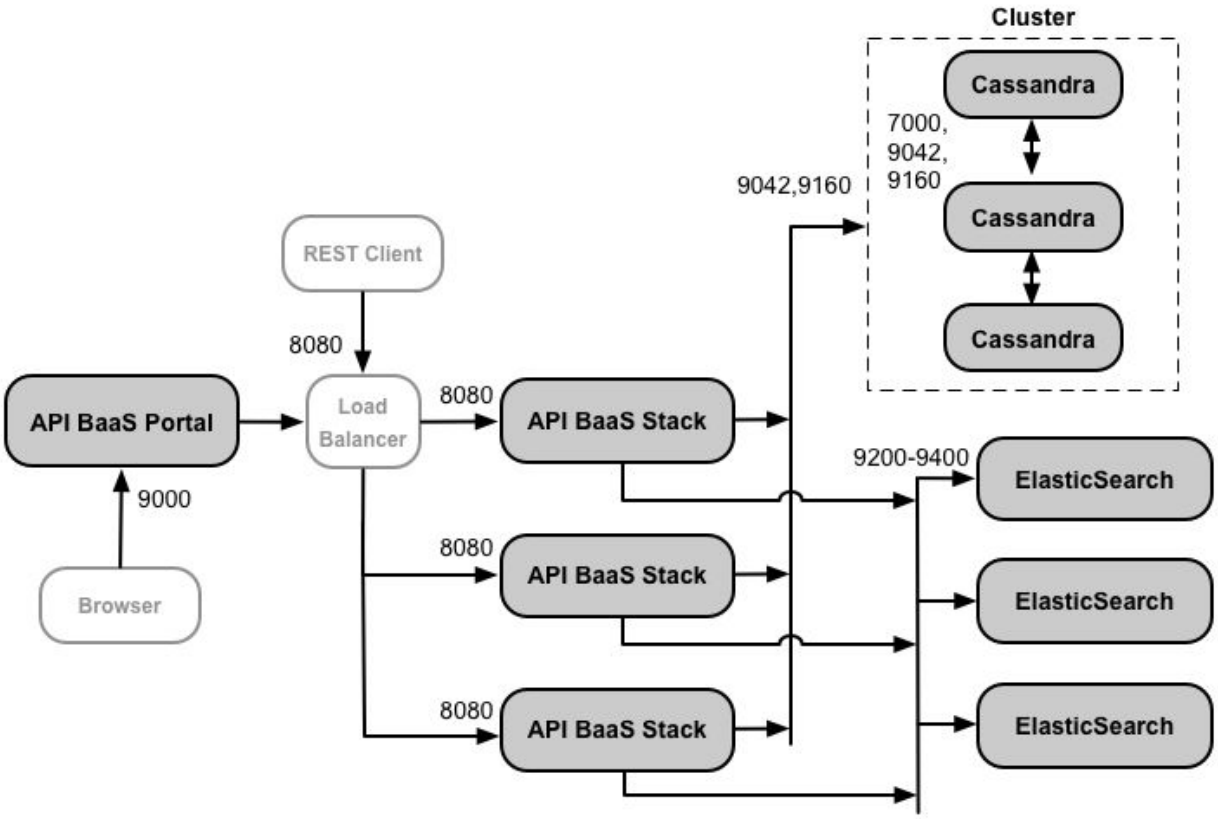
If you install the 12 host clustered configuration with two Data Centers, ensure that the nodes in the two Data Centers can communicate over the ports shown below:



Note: All Message Processors in all Data Centers must all be able to access each other over port 4528.

API BaaS port requirements

If you opt to install the API BaaS, you add the API BaaS Stack and API BaaS Portal components. These components use the ports shown in the figure below:



The Cassandra nodes can be dedicated to API BaaS, or can be shared with Edge.

A production installation of API BaaS uses a load balancer between the API BaaS Portal node and API BaaS Stack nodes. When configuring the Portal, and when making BaaS API calls, you specify the IP address or DNS name of the load balancer, not of the Stack nodes.

The table below shows the default ports that need to be opened in firewalls, by component:

Component	Port	Description
API BaaS Portal	9000	Port for the API BaaS UI
API BaaS Stack	8080	Port where API request are received
ElasticSearch	9200 to 9400	For communicating with API BaaS Stack and for communicating between ElasticSearch nodes

Licensing

Each installation of Edge requires a unique license file that you obtain from Apigee. You will need to provide the path to the license file when installing the management server, for example `/tmp/license.txt`.

The installer copies the license file to /`<inst_root>/apigee/customer/conf/license.txt`

If license file is valid, the management server validates the expiry and allowed Message Processor (MP) count. If any of the license settings is expired, you can find the logs in the following location: /`<inst_root>/apigee/var/log/edge-management-server/logs`. In this case you can contact Apigee Support for migration details.

Installation Checklist

Note:

- If an older version of Apigee Edge for Private Cloud was installed on the machine, ensure that you delete the folder `/tmp/java` before a new installation.
- If user “apigee” was created prior to the installation, ensure that “/home/apigee” exists as home directory and is owned by “apigee:apigee”.

The checklist covers the preceding prerequisites and provides a list of required files to obtain before proceeding. Here is a summary of the primary requirements covered there.

- **Installation user**: The user performing this installation must be the root user, or a user with sudo privileges. In many of the commands below, if you are not logged in as root, prefix the commands with "`sudo`".
- **Edge system administrator credentials**: As part of the installation, you are prompted to specify an email address and password used to create the Edge system administrator account. Never use these credentials for anything other than Edge system administration. You can later create different users and user types to create and manage API proxies, apps, and all other user-level tasks.

  To change the administrator password do not use the Edge UI to change the administrator password. See http://docs.apigee.com/api-services/latest/resetting-passwords for more information.
- **OS:** For operating system requirements, see https://apigee.com/docs/api-services/reference/supported-software.
- **Java:** Java requirements are covered under Prerequisites above. Refer to https://apigee.com/docs/api-services/reference/supported-software proceeding.

  Ensure that `JAVA_HOME` points to the root of the JDK for the user performing the installation.
- **Firewalls:** Firewall/host requirements are covered under Prerequisites above. Refer to the Firewalls and Virtual Hosts section before proceeding.

- **TCP Wrappers**: TCP Wrappers can block communication of some ports and can affect OpenLDAP, Postgres, 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, Postgres, and Cassandra ports.
- **SELinux**: Depending on your settings for SELinux, Edge can encounter issues with installing and starting Edge components. If necessary, you can disable SELinux or set it to permissive mode during installation, and then re-enabling it after installation. See Prerequisite: Disable SELinux for more.
- **iptables**: Validate that there are no iptables policies preventing connectivity between nodes on the required Edge ports. If necessary, you can stop iptables during installation using the command:

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

  On CentOS 7.x:

  ```
  > systemctl stop firewalld
  ```

- **License file:** A valid license file must be obtained to install Apigee Edge. Licensing information is covered under Prerequisites above. Refer to the Licensing section before proceeding.
- **Distribution files:** The Apigee Edge distribution files are installed as a set of RPMs and dependencies.
- **System limits:**
  - On Cassandra nodes, set soft and hard memlock, nofile, and address space (as) limits for installation user (default is “apigee") in `/etc/security/limits.conf` as shown below:

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

  - On Message Processor nodes, set the maximum number of open file descriptors to 64K by using the command:

    ```
     > ulimit -n 65535
    ```

    If necessary, you can raise that limit. For example, if you have a large number of temporary files open at any one time.

Installation Considerations

A typical Edge installation consists of Edge components distributed across multiple nodes. After you install Edge on a node, you then install and configure one or more Edge components on the node.

Installation process

Installing Edge on a node is a multi-step process:

- Disable SELinux on the node or set it to permissive mode. See Prerequisite: Disable SELinux for more.
- Select your Edge configuration from the list of recommended topologies. For example, you can install Edge on a single node for testing, or on 13 nodes for production. See Installation topologies for more.
- On each node in your selected topology, install the Edge `apigee-setup` utility:
    - Download the Edge `bootstrap_4.16.05.sh` file to `/tmp/bootstrap_4.16.05.sh`.
    - Install the Edge `apigee-service` utility and dependencies.
    - Install the Edge `apigee-setup` utility and dependencies.

      See Install the Edge apigee-setup utility for more.
- Use the `apigee-setup` utility to install one or more Edge components on each node based on your selected topology.

  See Install Edge components on a node.
- On the Management Server node, use the `apigee-setup` utility to install `apigee-provision`, the utilities that you use to create and manage Edge organizations.

  See Onboard an organization for more.

Handling an installation failure

In the case of a failure during the installation of an Edge component, you can try to correct the issue, and then run the installer again. The installer is designed to be run repeatedly in cases where it detects a failure, or if you later want to change or update a component after installation.

Who can perform the install

The Apigee Edge distribution files are installed as a set of RPMs and dependencies. To install, uninstall, and update Edge, the Edge commands must be run by the root user or by a user that has full sudo access. For full sudo access, that means the user has sudo access to perform the same operations as root.

Any user who wants to run the following commands or scripts must either be root, or be a user with full sudo access:

- **apigee-service** utility:
  - **apigee-service** commands: `install`, `uninstall`, `update`.
  - **apigee-all** commands: `install`, `uninstall`, `update`.
- **setup.sh** script to install Edge components (Unless you have already used "`apigee-service intall`" to install the required RPMs. Then root or full sudo access if not required.)
- **update.sh** script to update Edge components

Also, the Edge installer creates a new user on your system, named "apigee". Many Edge commands invoke sudo to run as the "apigee" user.

Any user who wants to run all other commands than the ones shown above must be a user with full sudo access to the "apigee" user. These commands include:

- **apigee-service** utility commands, including:
  - **apigee-service** commands such as `start`, `stop`, `restart`, `configure`.
  - **apigee-all** commands such as `start`, `stop`, `restart`, `configure`.

To configure a user to have full sudo access to the "apigee" user, edit the sudoers file to add:

```
installUser        ALL=(apigee)       NOPASSWD: ALL
```

where ***installUser*** is the username of the person working with Edge.

Any files or resources used by the Edge commands must be accessible to the "apigee" user. This includes the Edge license file and any config files.

Note: You can set the **RUN_USER** property for an Edge component to specify a different user than "apigee". If you do, then all of the Edge commands for that component invoke sudo to run as that user. Files or resources must then be accessible to that user.

When creating a configuration file, you can change its owner to "apigee:apigee" to ensure that it is accessible to Edge commands:

1. Create the file in an editor as any user.
2. Chown the owner of the file to "apigee:apigee" or, if you changed the user running the Edge service from the "apigee" user, chown the file to the user who is running the Edge service.

Silent installation of Edge components

You must pass a configuration file to the `apigee-setup` utility that contains the information about the Edge installation. The only requirement on silent installations is that the configuration file must be accessible or readable by the "apigee" user. For example, put the file in the /tmp directory on the node and chown it to "apigee:apigee".

All information in the configuration file is required except for the Edge system administrator's password. If you omit the password, the `apigee-setup` utility prompts you to enter it on the command line.

See Creating a configuration file for more.

Internet or non-Internet installation

To install Edge on a node, the node must be able to access the Apigee repository:

- **Nodes with an external Internet connection**

  Nodes with an external internet connection access the Apigee repository to install the Edge RPMs and dependencies.

- **Nodes without an external Internet connection**

  Nodes without an external Internet connection can access a mirrored version of the Apigee repository that you set up internally. This repository contains all Edge RPMs, but you have to ensure that you have all other dependencies available from repos on the internal network.

Setting up a virtual host

A virtual host on Edge defines the domains and Edge Router ports on which an API proxy is exposed, and, by extension, the URL that apps use to access an API proxy. A virtual host also defines whether the API proxy is accessed by using the HTTP protocol, or by the encrypted HTTPS protocol.

As part of the Edge onboarding process, you have to create an organization, environment, and virtual host. Edge provides the `setup-org` command to make this process easier for new users.

When you create the virtual host, you must specify the following information:

- **The name** of the virtual host that you use to reference it in your API proxies.
- **The port** on the Router for the virtual host. Typically these ports start at 9001 and increment by one for every new virtual host.
- **The host alias** of the virtual host. Typically the DNS name of the virtual host.

The Edge Router compares the `Host` header of the incoming request to the list of available host aliases as part of determining the API proxy that handles the request. When making a request through a virtual host, either specify a domain name that matches the host alias of a virtual host, or specify the IP address of the Router and the `Host` header containing the host alias.

For example, if you created a virtual host with a host alias of myapis.apigee.net on port 9001, then a cURL request to an API through that virtual host could use one of the following forms:

- If you have a DNS entry for **myapis.apigee.net**:

  ```
  curl http://myapis.apigee.net:9001/{proxy-base-path}/{resource-path}
  ```

- If you do not have a DNS entry for **myapis.apigee.net**:

```
curl http://<routerIP>:9001/{proxy-base-path}/{resource-path}
     -H 'Host: myapis.apigee.net'
```

In this form, you specify the IP address of the Router, and pass the host alias in the `Host` header.

Note: The curl command, most browsers, and many other utilities automatically append the `Host` header with the domain as part of the request, so you can actually use a curl command in the form:

```
curl http://<routerIP>:9001/{proxy-base-path}/{resource-path}
```

Options when you do not have a DNS entry for the virtual host

One option when you do not have a DNS entry is to set the host alias to the IP address of the Router and port of the virtual host, as `<routerIP>:port`. For example:

```
192.168.1.31:9001
```

When you make a curl command in the form below:

```
curl http://<routerIP>:9001/{proxy-base-path}/{resource-path}
```

This option is preferred because it works well with the Edge UI.

If you have multiple Routers, add a host alias for each Router, specifying the IP address of each Router and port of the virtual host.

Alternatively, you can set the host alias to a value, such as **temp.hostalias.com**. Then, you have to pass the `Host` header on every request:

```
curl -v http://<routerIP>:9001/{proxy-base-path}/{resource-path}
     -H 'host: temp.hostalias.com'
```

Or, add the host alias to your `/etc/hosts` file. For example, add this line to `/etc/hosts`:

```
192.168.1.31   temp.hostalias.com
```

Then you can make a request as if you had a DNS entry:

```
curl -v http://myapis.apigee.net:9001/{proxy-base-path}/{resource-path}
```

Configuring Edge components post installation

To configure Edge after installation, you use a combination of `.properties` files and Edge utilities. For example, to configure SSL on the Edge UI, you edit `.properties` files to set the necessary properties. Changes to `.properties` files require you to restart the affected Edge component.

The `.properties` files are located in the `/opt/apigee/customer/application` directory. Each component has its own `.properties` file in that directory. For example, `router.properties` and `management-server.properties`.

Note: If you have not set any properties for a component, the `/opt/apigee/customer/application` directory might not contain a `.properties` file for the component. In that case, create one.

To set a property for a component, edit the corresponding `.properties` file, and then restart the component:

```
> /opt/apigee/apigee-service/bin/apigee-service component restart
```

For example:

```
> /opt/apigee/apigee-service/bin/apigee-service edge-router restart
```

When you update Edge, the `.properties` files in the `/opt/apigee/customer/application` directory are read. That means the update retains any properties that you set on the component.

See http://docs.apigee.com/api-services/latest/how-configure-edge for more information on Edge configuration.

Invoking commands on Edge components

Edge installs management utilities under `/opt/apigee/apigee-service/bin` that you can use to manage an Edge installation. For example, you can use the `apigee-all` utility to start, stop, restart, or determine the status of all Edge components on the node:

```
/opt/apigee/apigee-service/bin/apigee-all stop|start|restart|status|version
```

Use the `apigee-service` utility to control and configure individual components. The `apigee-service` utility has the form:

```
/opt/apigee/apigee-service/bin/apigee-service component action
```

For example, to restart the Edge Router:

```
/opt/apigee/apigee-service/bin/apigee-service edge-router restart
```

You can determine the list of components installed on the node by examining the `/opt/apigee` directory. That directory contains a subdirectory for every Edge component installed on the node. Each subdirectory is prefixed by:

- `apigee` - a third-party component used by Edge. For example, `apigee-cassandra`.
- `edge` - an Edge component from Apigee. For example, `edge-management-server`.
- `edge-mint` - a Monetization component. For example `edge-mint-management-server`.
- `baas` - an API BaaS component. For example `baas-usergrid`.

The complete list of actions for a component depends on the component itself, but all components support the following actions:

- `start`, `stop`, `restart`
- `status`, `version`
- `backup`, `restore`

- `install, uninstall`

Accessing log files

The log files for each component are contained in the `/opt/apigee/var/log` directory. Each component has its own subdirectory. For example, the logs for the Management Server are in the directory:

```
/opt/apigee/var/log/edge-management-server
```

Common Yum commands

The Edge installation tools for Linux rely on Yum to install and update components. You might have to use several Yum commands to manage an installation on a node.

- Clean all Yum caches:

  ```
  sudo yum clean all
  ```

- To update an Edge component:

  ```
  sudo yum update componentName
  ```

  For example:

  ```
  sudo yum update apigee-setup
  sudo yum update edge-management-server
  ```

File System Structure

Edge installs all files in the `/opt/apigee` directory.

Note: You cannot change this directory location. However, you can create a symlink to map it to a different location. See Creating a symlink from /opt/apigee for more.

In this guide and in the Edge *Operations Guide*, the root installation directory is noted as:

```
<inst_root>/apigee
```

The installation uses the following file system structure to deploy Apigee Edge for Private Cloud.

Log Files

Components	Location
Management Server	<inst_root>/apigee/var/log/edge-management-server
Router	<inst_root>/apigee/var/log/edge-router
Message Processor	<inst_root>/apigee/var/log/edge-message-processor

Apigee Qpid Server	<inst_root>/apigee/var/log/edge-qpid-server
Apigee Postgres Server	<inst_root>/apigee/var/log/edge-postgres-server
Edge UI	<inst_root>/apigee/var/log/edge-ui
ZooKeeper	<inst_root>/apigee/var/log/apigee-zookeeper
OpenLDAP	<inst_root>/apigee/var/log/apigee-openldap
Cassandra	<inst_root>/apigee/var/log/apigee-cassandra
Qpidd	<inst_root>/apigee/var/log/apigee-qpidd
PostgreSQL database	<inst_root>/apigee/var/log/apigee-postgresql

Data

Components	**Location**
Management Server	<data_root>/apigee/data/edge-management-server
Router	<data_root>/apigee/data/edge-router
Message Processor	<data_root>/apigee/data/edge-message-processor
Apigee Qpid agent	<data_root>/apigee/data/edge-qpid-server
Apigee Postgres agent	<data_root>/apigee/data/edge-postgres-server
ZooKeeper	<data_root>/apigee/data/apigee-zookeeper
OpenLDAP	<data_root>/apigee/data/apigee-openldap
Cassandra	<data_root>/apigee/data/apigee-cassandra/data
Qpidd	<data_root>/apigee/data/apigee-qpid/data
PostgreSQL database	<data_root>/apigee/data/apigee-postgres/pgdata

Install the Edge apigee-setup utility

To install Edge on a node, you first install the Edge `apigee-setup` utility. If you are in an environment where your nodes do not have an external internet connection, you must also install a local copy of the Apigee repo.

Creating a symlink from /opt/apigee

Edge installs all files in the `/opt/apigee` directory. You cannot change this directory. However, if desired, you can create a symlink to map `/opt/apigee` to another location.

Before you create the symlink, you must first create a user and group named "apigee". This is the same group and user created by the Edge installer.

To create the symlink, perform these steps before downloading the bootstrap_4.16.05.sh file. You must perform all of these steps as root:

1. Create the "apigee" user and group:

   ```
   > groupadd -r apigee
   > useradd -r -g apigee -d /opt/apigee -s /sbin/nologin -c "Apigee platform user" apigee
   ```

2. Create a symlink from `/opt/apigee` to your desired install root:

   ```
   > ln -Ts /srv/myInstallDir /opt/apigee
   ```

   where ***/srv/myInstallDir*** is the desired location of the Edge files.

3. Change ownership of the install root and symlink to the "apigee" user:

   ```
   > chown -h apigee:apigee /srv/apigee /opt/apigee
   ```

Prerequisite: Disable SELinux

You must disable SELinux, or set it to permissive mode, before you can install Edge `apigee-setup` utility or any Edge components. If necessary, after installing Edge, you can re-enable SELinux.

Note: This step is not required on SUSE installations.

- To **temporarily** set SELinux to permissive mode, execute the following command:
  a. **On a Linux 6.x operating system**:

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

     To re-enable SELinux after installing Edge:

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

b. **On a Linux 7.x operating system**:

```
setenforce 0
```

To re-enable SELinux after installing Edge:

```
setenforce 1
```

- To **permanently** disable SELinux or set it to permissive mode:
  a) Open `/etc/sysconfig/selinux` in an editor.
  b) Set `SELINUX=disabled` or `SELINUX=permissive`
  c) Save your edits.
  d) Restart the node.
  e) If necessary, re-enable SELinux after Edge installation by repeating this procedure to set `SELINUX=enabled`.

Install Edge apigee-setup utility on a node with an external internet connection

To install Edge on a node with an external Internet connection:

1. Obtain the username and password from Apigee that you use to access the Apigee repository. If you have an existing username:password for the Apigee ftp site, you can use those credentials.
2. Log in to your node as root to install the Edge RPMs

   Note: While RPM installation requires root access, you can perform Edge configuration without root access.
3. Disable SELinux as described in Prerequisite: Disable SELinux.
4. Download the Edge bootstrap_4.16.05.sh file to `/tmp/bootstrap_4.16.05.sh`:

   ```
   > curl https://software.apigee.com/bootstrap_4.16.05.sh -o
   /tmp/bootstrap_4.16.05.sh
   ```
5. Install the Edge `apigee-service` utility and dependencies:

   ```
   > sudo bash /tmp/bootstrap_4.16.05.sh apigeeuser=uName
   apigeepassword=pWord
   ```

   where **`uName:pWord`** are the username and password you received from Apigee. If you omit **`pWord`**, you will be prompted to enter it.

By default, the installer checks to see that you have Java 1.8 installed. If you do not, it installs it for you. Use the `JAVA_FIX` option to specify how to handle Java installation. `JAVA_FIX` takes the following values:

I = Install OpenJDK 1.8 (default)
C = Continue without installing Java
Q = Quit. For this option, you have to install Java yourself.

The installation of the `apigee-service` utility creates the `/etc/yum.repos.d/apigee.repo` file that defines the Apigee repository. To view the definition file, use the command:

```
> cat /etc/yum.repos.d/apigee.repo
```

To view the repo contents, use the command:

```
> sudo yum -v repolist 'apigee*'
```

6. Use `apigee-service` to install the `apigee-setup` utility:

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

7. Use `apigee-setup` to install and configure Edge components on the node. See Install Edge components on a node for more.

Install Edge apigee-setup utility on a node with no external Internet connection

If your Edge nodes are behind a firewall, or in some other way are prohibited from accessing the Apigee repository over the Internet, then you must create a local repository, or *mirror*, of the Apigee repo. That mirror must then be accessible to all nodes. Once created, nodes can then access that local mirror to install Edge.

After you create a local Edge repository, you might later have to update it with the latest Edge release files. The following sections describe how to create a local repository, and how to update it.

Create a local Apigee repository

To create a local Apigee repo:

1. Obtain the username and password from Apigee that you use to access the Apigee repository. If you have an existing username:password for the Apigee ftp site, you can use those credentials.
2. Log in to your RedHat or CentOS node as root to install the Edge RPMs.

   Note: While RPM installation requires root access, you can perform Edge configuration without root access.

3. Make sure you have the latest version of `yum-utils`:

   ```
   > sudo yum update yum-utils
   ```

4. Disable SELinux as described in Prerequisite: Disable SELinux.
5. Download the Edge bootstrap_4.16.05.sh file to `/tmp/bootstrap_4.16.05.sh`:

   ```
   > curl https://software.apigee.com/bootstrap_4.16.05.sh -o
   /tmp/bootstrap_4.16.05.sh
   ```

6. Install the Edge `apigee-service` utility and dependencies:

   ```
   > sudo bash /tmp/bootstrap_4.16.05.sh apigeeuser=uName
   apigeepassword=pWord
   ```

   where **`uName:pWord`** are the username and password you received from Apigee. If you omit **`pWord`**, you will be prompted to enter it.

7. Install the `apigee-mirror` utility on the node:

   ```
   > /opt/apigee/apigee-service/bin/apigee-service apigee-mirror install
   ```

8. Use the `apigee-mirror` utility to sync the Apigee repo to the `/opt/apigee/data/apigee-mirror/repos/` directory.

   To minimize the size of the repo, include the `--only-new-rpms` to download just the latest RPMs. You need approximately 800 MB of disk space for the download:

   ```
   > /opt/apigee/apigee-service/bin/apigee-service apigee-mirror sync
   --only-new-rpms
   ```

   If you want to entire repo, including older RPMs, omit `--only-new-rpms`. You need approximately 6 GB of disk space for the full download:

   ```
   > /opt/apigee/apigee-service/bin/apigee-service apigee-mirror sync
   ```

   You now have a local copy of the Apigee repo. The next section describes how to install the Edge `apigee-setup` utility from the local repo.

9. **(Optional)** If you want to install Edge from the local repo onto the same node that hosts the local repo, then you need to first run the following commands:

   a) Run `bootstrap_4.16.05.sh` from the local repo to install the `apigee-service` utility:

   ```
   > sudo bash /opt/apigee/data/apigee-mirror/repos/bootstrap_4.16.05.sh
   ```

```
apigeeprotocol="file://"
apigeerepobasepath=/opt/apigee/data/apigee-mirror/repos
```

b) Use `apigee-service` to install the `apigee-setup` utility:

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

c) Use `apigee-setup` to install and configure Edge components on the node. See Install Edge components on a node for more.

Install apigee-setup on a remote node from the local repo

You have two options for installing Edge from the local repo. You can either:

- Create a .tar file of the repo, copy the .tar file to a node, and then install Edge from the .tar file.
- Install a webserver on the node with the local repo so that other nodes can access it. Apigee provides the Nginx webserver for you to use, or you can use your own webserver.

Install from the .tar file:

1. On the node with the local repo, use the following command to package the local repo into a single .tar file named `/opt/apigee/data/apigee-mirror/apigee-4.16.05.tar.gz`:

   ```
   > /opt/apigee/apigee-service/bin/apigee-service apigee-mirror package
   ```

2. Copy the .tar file to the node where you want to install Edge. For example, copy it to the `/tmp` directory on the new node.
3. On the new node, disable SELinux as described in Prerequisite: Disable SELinux.
4. On the new node, untar the file to the /tmp directory:

   ```
   > tar -xzf apigee-4.16.05.tar.gz
   ```

   This command creates a new directory, named `repos`, in the directory containing the .tar file. For example `/tmp/repos`.

5. Install the Edge `apigee-service` utility and dependencies from `/tmp/repos`:

   ```
   > sudo bash /tmp/repos/bootstrap_4.16.05.sh apigeeprotocol="file://"
   apigeerepobasepath=/tmp/repos
   ```

   Notice that you include the path to the `repos` directory in this command.

6. Use `apigee-service` to install the `apigee-setup` utility:

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

7. Use `apigee-setup` to install and configure Edge components on the node. See Install Edge components on a node for more.

Install from the repo using the Nginx webserver:

1. Install the Nginx webserver on the repo node:

   ```
   /> opt/apigee/apigee-service/bin/apigee-service apigee-mirror nginxconfig
   ```

2. By default, Nginx is configured to use localhost as the server name and port 3939. To change these values
   a. Open `/opt/apigee/customer/application/mirror.properties` in an editor. Create the file if it does not exist.
   b. Set the following values as necessary:

      ```
      conf_apigee_mirror_listen_port=3939
      conf_apigee_mirror_server_name=localhost
      ```

   c. Restart Nginx:

      ```
      > /opt/nginx/scripts/apigee-nginx restart
      ```

3. By default, the repo requires a username:password of `admin:admin`. To change these credentials, set the following environment variables:

   ```
   MIRROR_USERNAME=uName
   MIRROR_PASSWORD=pWord
   ```

4. On the new node, disable SELinux as described in Prerequisite: Disable SELinux.
5. On the remote node, download the Edge bootstrap_4.16.05.sh file to `/tmp/bootstrap_4.16.05.sh`:

   ```
   > /usr/bin/curl http://uName:pWord@remoteRepo:3939/bootstrap_4.16.05.sh -o
   /tmp/bootstrap_4.16.05.sh
   ```

   where **`uName:pWord`** are the username and password you set above for the repo, and ***remoteRepo*** is the IP address or DNS name of the repo node.

6. On the remote node, install the Edge `apigee-service` utility and dependencies:

   ```
   > sudo bash /tmp/bootstrap_4.16.05.sh apigeerepohost=remoteRepo:3939
   apigeeuser=uName apigeepassword=pWord apigeeprotocol=http://
   ```

   where **`uName:pWord`** are the repo username and password.

7. On the remote node, use `apigee-service` to install the `apigee-setup` utility:

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

8. Use `apigee-setup` to install and configure Edge components on the remote node. See Install Edge components on a node for more.

Update a local Apigee repository

To update the repo, you must download the latest bootstrap_4.16.05.sh file, then perform a new sysnc:

1. Download the Edge bootstrap_4.16.05.sh file to `/tmp/bootstrap_4.16.05.sh`:

   ```
   > curl https://software.apigee.com/bootstrap_4.16.05.sh -o
   /tmp/bootstrap_4.16.05.sh
   ```

2. Perform the sync:

   ```
   > /opt/apigee/apigee-service/bin/apigee-service apigee-mirror sync
   --only-new-rpms
   ```

   If you want to entire repo:

   ```
   > /opt/apigee/apigee-service/bin/apigee-service apigee-mirror sync
   ```

Clean a local Apigee repo

Cleaning the local repo deletes `/opt/apigee/data/apigee-mirror` and `/var/tmp/yum-apigee-*`.

To clean the local repo, use:

```
> /opt/apigee/apigee-service/bin/apigee-service apigee-mirror clean
```

Add or update Edge 4.16.01 in a 4.16.05 repo

If you have to maintain installations for both Edge 4.16.05 and 4.16.01, you can maintain a repo that contains both versions. From that repo, you can then install either Edge 4.16.05 and 4.16.01.

To add 4.16.01 to an 4.15.05 repo:

1. Ensure that you have installed the 4.16.05 version of the `apigee-mirror` utility:

   ```
   > /opt/apigee/apigee-service/bin/apigee-service apigee-mirror version
   ```

   You should see a result in the form below, where ***xyz*** is the build number:

   `apigee-mirror-`**`4.16.05-0.0.`*`xyz`***

2. Use the `apigee-mirror` utility to download Edge 4.16.01 to your repo. Notice how you prefix the command with **`apigeereleasever=4.16.01`**:

   ```
   > apigeereleasever=4.16.01 /opt/apigee/apigee-service/bin/apigee-service
   apigee-mirror sync --only-new-rpms
   ```

   Use this same command to later update the 4.16.01 repo.

3. Examine the `/opt/apigee/data/apigee-mirror/repos` directory to see the file structure:

   ```
   > ls /opt/apigee/data/apigee-mirror/repos
   ```

   You should see the following files and directories:

   ```
   apigee  apigee-repo-1.0-6.x86_64.rpm  bootstrap_4.16.01.sh
   bootstrap_4.16.05.sh  thirdparty
   ```

   Notice how you have a bootstrap file for both versions of Edge. The `apigee` directory also contains separate directories for each version of Edge.

4. To package the repo into a .tar file, use the following command:

   ```
   > /opt/apigee/apigee-service/bin/apigee-service apigee-mirror package
   ```

   This command packages both the 4.16.05 and 4.16.01 repos into the same .tar file. You cannot package only part of the repo.

To install Edge from the local repo or .tar file, just make sure to run the correct bootstrap file by using one of the following commands:

- If installing from a .tar file, run the correct bootstrap file from the repo:

  ```
  > sudo bash /tmp/repos/bootstrap_4.16.0X.sh apigeeprotocol="file://"
  apigeerepobasepath=/tmp/repos
  ```

  To complete the installation, follow the remaining steps from Install from the .tar file:.

- If installing using the Nginx webserver, download and then run the correct bootstrap file from the repo:

  ```
  > /usr/bin/curl http://uName:pWord@remoteRepo:3939/bootstrap_4.16.0X.sh -o
  /tmp/bootstrap_4.16.0X.sh

  > sudo bash /tmp/bootstrap_4.16.0X.sh apigeerepohost=remoteRepo:3939
  apigeeuser=uName apigeepassword=pWord apigeeprotocol=http://
  ```

To complete the installation, follow the remaining steps from Install from the repo using the Nginx webserver:.

Install Edge components on a node

After you install the Edge `apigee-setup` utility on a node, use the `apigee-setup` utility to install one or more Edge components on the node.

Note: See Install the Edge apigee-setup utility for more on installing the Edge `apigee-setup` utility.

The `apigee-setup` utility uses a command in the form:

```
> /opt/apigee/apigee-setup/bin/setup.sh -p component -f configFile
```

where ***component*** is the Edge component to install, and ***configFile*** is the silent configuration file containing the installation information. The configuration file must be accessible or readable by the "apigee" user. For example, put the file in the /tmp directory on the node.

For example, to install the Edge Management Server:

```
> /opt/apigee/apigee-setup/bin/setup.sh -p ms -f /tmp/myConfig
```

Installation considerations

As you write your config file, take into consideration the following options.

Setting up Postgres master-standby replication

By default, Edge installs all Postgres nodes in master mode. However, in most production systems, you configure them to use master-standby replication so that if the master node fails, the standby node can continue to server traffic.

You can enable and configure master-standby replication at install time by using properties in the silent config file. Or, you can enable master-standby replication after installation. For more, see Set up Master-Standby Replication for Postgres.

Enabling Cassandra authentication

By default, Cassandra installs without authentication enabled. That means anyone can access Cassandra. You can enable authentication after installing Edge, or as part of the installation process.

You can enable Cassandra authentication as install time by using properties in the silent config file. Or, you can enable it after installation.

Note: While you can enable authentication when you install Cassandra, you cannot change the default username and password. You have to perform that step manually after installation of Cassandra completes.

For more, see Enable Cassandra authentication.

Binding the Router to a protected port

If you want to bind the Router to a protected port, such as port numbers less than 1000, then you have to configure the Router to run as a user with access to those ports. By default, the Router runs as the user "apigee" which does not have access to privileged ports.

To run the Router as a different user:

1. As root, create the file `/opt/apigee/etc/edge-router.d/RUN_USER.sh`.
2. Add the following entry to the file:

   ```
   RUN_USER=root
   ```

   If you do not want to run the Router as root, specify a user with access to the port.
3. Save the file.
4. If you specified a user other than root, change the owner of the file to that user:

   ```
   > chown USER:USER /opt/apigee/etc/edge-router.d/RUN_USER.sh
   ```
5. Restart router:

   ```
   > /opt/apigee/apigee-service/bin/apigee-service edge-router restart
   ```

Specifying the components to install

The following table lists the options you pass to the `-p` option of the `apigee-service` utility to specify which components to install on the node:

Component	Description
c	Install Cassandra only.
ld	Install OpenLDAP only.
ms	Install Edge Management Server, which also installs the Edge UI and OpenLDAP. If you set USE_LDAP_REMOTE_HOST=y in the config file, then OpenLDAP installation is skipped and the Management Server uses OpenLDAP installed on a different node.
r	Install Edge Router only.

mp	Install Edge Message Processor only.
rmp	Install Edge Router and Message Processor.
qs	Install Qpid Server only.
ps	Install Postgres Server only.
mo	Install Monetization.
ds	Install ZooKeeper and Cassandra.
ui	Install the Edge UI.
sa	Install Edge standalone, meaning Cassandra, ZooKeeper, Management Server, OpenLDAP, Edge UI, Router, and Message Processor. This option omits the Edge analytics components: Qpid and Postgres. Use this option for development and testing only, not for production.
sax	Install analytics components, meaning Qpid and Postgres. Use this option for development and testing only, not for production.
aio	Install all components on a single node. Use this option for development and testing only, not for production.

Creating a configuration file

The configuration file contains all the information necessary to install Edge. You can often use the same configuration file to install all components in an Edge installation.

However, you will have to use different configuration files, or modify your configuration file, if:

- You are installing multiple OpenLDAP servers and need to configure replication as part of a 13-node installation. Each file requires different values for `LDAP_SID` and `LDAP_PEER`.
- You are creating multiple data centers as part of a 12-node installation. Each data center requires different settings for properties such as `ZK_CLIENT_HOSTS` and `CASS_HOSTS`.

Example configuration file

Shown below is an example of a complete silent configuration file for a 9 node Edge installation. Edit this file as necessary for your configuration. Use the `-f` option to `setup.sh` to include this file. Also shown below are example configuration files for each Edge topology.

Note: The definition of the `IP#` variables for the Router, Message Processor, Qpid, and Postgres nodes are for illustrating the node configuration; they are not actually used.

```
# IP address or DNS name of nodes.
IP1=192.168.1.1  # Management Server, OpenLDAP, UI, ZooKeeper, Cassandra
IP2=192.168.1.2  # ZooKeeper, Cassandra
IP3=192.168.1.3  # ZooKeeper, Cassandra
IP4=192.168.1.4  # Router, Message Processor
IP5=192.168.1.5  # Router, Message Processor
IP6=192.168.1.6  # Qpid
IP7=192.168.1.7  # Qpid
IP8=192.168.1.8  # Postgres
IP9=192.168.1.9  # Postgres

# Must resolve to IP address or DNS name of host - not to 127.0.0.1 or
localhost.
HOSTIP=$(hostname -i)

# Set Edge sys admin credentials.
ADMIN_EMAIL=your@email.com
APIGEE_ADMINPW=yourPassword          # If omitted, you are prompted for it.

# Location of Edge license file.
LICENSE_FILE=/tmp/license.txt

# Management Server information.
MSIP=$IP1  # IP or DNS name of Management Server node.
# Specify the port the Management Server listens on for API calls.
# APIGEE_PORT_HTTP_MS=8080      # Default is 8080.

#
# OpenLDAP information.
#
# Set to y if you are connecting to a remote LDAP server.
# If n, Edge installs OpenLDAP when it installs the Management Server.
USE_LDAP_REMOTE_HOST=n

# If connecting to remote OpenLDAP server, specify the IP/DNS name and port.
# LDAP_HOST=$IP1           # IP or DNS name of OpenLDAP node.
# LDAP_PORT=10389          # Default is 10389.
APIGEE_LDAPPW=yourLdapPassword
```

```
# Specify OpenLDAP without replication, 1, or with replication, 2.
LDAP_TYPE=1

# Set only if using replication.
# LDAP_SID=1    # Unique ID for this LDAP server.
# LDAP_PEER=    # IP or DNS name of LDAP peer.


BIND_ON_ALL_INTERFACES=y

# The Message Processor and Router pod.
MP_POD=gateway

# The name of the region, corresponding to the data center name.
REGION=dc-1     # Use dc-1 unless installing in a
                # multi-data center environment.

# ZooKeeper information.
# See table below if installing in a multi-data center environment.
ZK_HOSTS="$IP1 $IP2 $IP3"       # IP/DNS names of all ZooKeeper nodes.
ZK_CLIENT_HOSTS="$IP1 $IP2 $IP3"     # IP/DNS names of all ZooKeeper nodes.

# Cassandra information.
CASS_CLUSTERNAME=Apigee    # Default name is Apigee.

# IP or DNS names of the Cassandra hosts separated by spaces.
CASS_HOSTS="$IP1:1,1 $IP2:1,1 $IP3:1,1"

# Set to enable Cassandra authentication.
# CASS_AUTH=y   # The default value is n.
# Cassandra uname/pword required if you enabled Cassandra authentication.
# CASS_USERNAME=
# CASS_PASSWORD=

# Optionally use to enable Postgres master-standby replication.
# PG_MASTER=IPorDNSofNewMaster
# PG_STANDBY=IPorDNSofOldMaster

# SMTP information.
SKIP_SMTP=n          # Skip now and configure later by specifying "y".
SMTPHOST=smtp.gmail.com
SMTPUSER=your@email.com
SMTPPASSWORD=yourEmailPassword
SMTPSSL=y
SMTPPORT=465         # If no SSL, use a different port, such as 25.
```

The following table contains additional information about these properties:

Property	Note

IP/DNS names	Do not use a host name mapping to 127.0.0.1 or an IP address of 127.0.0.1 when specifying the IP address of a node.
`ADMIN_EMAIL` `APIGEE_ADMINPW`	The system administrator's password must be at least 8 characters long and contain one uppercase letter, one lowercase letter, one digit or one special character. If you omit the password, you will be prompted for it.
`LICENSE_FILE`	The location of the license file, which must be accessible to the "apigee" user. For example, store it in the /tmp directory and chmod 777 on the file. The file is copied to the Edge installation directory.

`USE_LDAP_REMOTE_HOST`
`LDAP_HOST`
`LDAP_PORT`
	If `USE_LDAP_REMOTE_HOST` is n, Edge automatically installs OpenLDAP when it installs the Management Server.
	Set `USE_LDAP_REMOTE_HOST` to y if you are connecting to a remote LDAP server. OpenLDAP is not installed with the Management Server.
	If you are connecting to a remote OpenLDAP server, use `LDAP_HOST`
	and `LDAP_PORT` to specify the IP address or DNS name and port number of the host.

`LDAP_TYPE`
`LDAP_SID`
`LDAP_PEER`
	Set `LDAP_TYPE=1` for OpenLDAP with no replication. `LDAP_TYPE=2` corresponds to OpenLDAP with replication.
	If your Edge topology uses a single OpenLDAP server, specify 1. If your Edge installation uses multiple OpenLDAP nodes, such as in a 13-node production installation, specify 2.
	If you enable replication, set the following properties:
	● `LDAP_SID=1` - Unique ID for this LDAP server. Each LDAP node uses a different ID. For example, set to 2 for LDAP peer.
	● `LDAP_PEER=10.0.0.1` - IP or DNS name of LDAP peer.

`BIND_ON_ALL_INTERFACES`	If set to "y" then the Router/Message Processor bind (listen) on all interfaces (IPs). If set to “n” then the Router/Message

	Processor bind (listen) on a specific interface, the IP returned by the "hostname -i" command).
`MP_POD`	Specify the name of the Message Processor and Router pod. By default, the name is `gateway`.
`REGION`	Region name. By convention, names are typically in the form dc-# where # corresponds to an integer value. For example, dc-1, dc-2, etc. You can use dc-1 unless installing in a multi-data center environment. In a multiple data center installation, the value is dc-1, or dc-2, etc. depending on which data center you are installing. However, you are not restricted to using only names in the form dc-#. You can use any name for the region.
`ZK_HOSTS`	The IP addresses or DNS names of the ZooKeeper nodes. The IP addresses or DNS names must be listed in the same order on all ZooKeeper nodes. In a multi-data center environment, list all ZooKeeper nodes from both data centers. Specify the ":observer" modifier on ZooKeeper nodes only when creating multiple data centers as described in a 12-host installation. In a single data center installation, omit that modifier. See 12-host clustered installation for more.
`ZK_CLIENT_HOSTS`	The IP addresses or DNS names of the ZooKeeper nodes used by this data center. The IP addresses or DNS names must be listed in the same order on all ZooKeeper nodes. In a single data center installation, these are the same nodes as specified by `ZK_HOSTS`. In a multi-data center environment, list only the ZooKeeper nodes in this data center. See 12-host clustered installation for more.
`CASS_HOSTS`	The IP addresses or DNS names of the Cassandra nodes. The first two nodes will be used as seed servers. The IP addresses or DNS names must be listed in the same order on all Cassandra nodes. Cassandra nodes can have an optional ":dc,ra" suffix that specifies the data center and rack of the Cassandra node.

	Specify this modifier only when creating multiple data centers as described in a 12-host installation. In a single data center installation, omit that modifier. For example '192.168.124.201:1,1 = datacenter 1 and rack/availability zone 1, and '192.168.124.204:2,1 = datacenter 2 and rack/availability zone 1. In a multi-datacenter environment, to overcome firewall issues, `CASS_HOSTS` have to be ordered in a manner (as shown in above example) such that the nodes of the current datacenter are placed at the beginning. See 12-host clustered installation for more.

`CASS_AUTH`
`CASS_USERNAME`
`CASS_PASSWORD`
	If you enable Cassandra authentication, `CASS_AUTH=y`, you can pass the Cassandra user name and password by using these properties.

`PG_MASTER` `PG_STANDBY`	Set to enable Postgres master-standby replication, in the form: `PG_MASTER=`***`IPorDNSofNewMaster`*** `PG_STANDBY=`***`IPorDNSofOldMaster`***

`SKIP_SMTP`
`SMTPHOST`
`SMTPUSER`
`SMTPPASSWORD`
`SMTPSSL`
`SMTPPORT`
	Configure SMTP so Edge can send emails for lost passwords and other notifications.
	If SMTP user credentials are not required, omit `SMTPUSER` and `SMTPPASSWORD`.

Order of component installation

The order of component installation is based on your desired topology.

All of the installation example shown below assume that you are installing:

- With Cassandra authentication disabled (default). See Enable Cassandra authentication for more.
- With Postgres master-standby replication disabled (default). See Set up Master-Standby Replication for Postgres for more.

Note: You must disable SELinux or set it to permissive mode before you install Edge components. See Prerequisite: Disable SELinux for more.

Note: The Installation Checklist details the installation prerequisites and provides a list of required files to obtain before proceeding with the installation. Ensure that you have reviewed the checklist before beginning the installation process.

Installation log files

By default, the `setup.sh` utility writes log information about the installation to:

`/opt/apigee/var/log/apigee-setup/setup.log`

If the user running the `setup.sh` utility does not have access to that directory, it writes the log to the `/tmp` directory as a file named `setup_`***`username`***`.log`.

If the user does not have access to /tmp, the `setup.sh` utility fails.

All-in-one Installation

Note: See a video of an Edge all-in-one install here.

1. Install all components on a single node using the command:

   ```
   > /opt/apigee/apigee-setup/bin/setup.sh -p aio -f configFile
   ```

2. Delete any files in `/opt/nginx/conf.d`:

   ```
   > rm -f /opt/nginx/conf.d/*
   ```

3. Restart the Edge Router:

   ```
   > /<inst_root>/apigee/apigee-service/bin/apigee-service edge-router
   restart
   ```

4. Test the installation as described at Test the install.
5. Onboard your organization as described at Onboard an organization.

Shown below is a silent configuration file for this topology:

```
# With SMTP
IP1=IPorDNSnameOfNode
HOSTIP=$(hostname -i)
ADMIN_EMAIL=opdk@apigee.com
APIGEE_ADMINPW=Secret123
LICENSE_FILE=/tmp/license.txt
MSIP=$IP1
LDAP_TYPE=1
APIGEE_LDAPPW=secret
BIND_ON_ALL_INTERFACES=y
MP_POD=gateway
REGION=dc-1
ZK_HOSTS="$IP1"
```

```
ZK_CLIENT_HOSTS="$IP1"
CASS_HOSTS="$IP1"
SKIP_SMTP=n
SMTPHOST=smtp.example.com
SMTPUSER=smtp@example.com
# 0 for no username
SMTPPASSWORD=smtppwd
# 0 for no password
SMTPSSL=n
SMTPPORT=25
```

2-host standalone installation

1. Install Standalone Gateway and node 1

   ```
   > /opt/apigee/apigee-setup/bin/setup.sh -p sa -f configFile
   ```

2. On node 1:
   a. Delete any files in `/opt/nginx/conf.d`:

      ```
      > rm -f /opt/nginx/conf.d/*
      ```

   b. Restart the Edge Router:

      ```
      > /<inst_root>/apigee/apigee-service/bin/apigee-service edge-router
      restart
      ```

3. Install Analytics on node 2:

   ```
   > /opt/apigee/apigee-setup/bin/setup.sh -p sax -f configFile
   ```

4. Test the installation as described at Test the install.
5. Onboard your organization as described at Onboard an organization.

Shown below is a silent configuration file for this topology:

```
# With SMTP
IP1=IPorDNSnameOfNode1
HOSTIP=$(hostname -i)
ADMIN_EMAIL=opdk@apigee.com
APIGEE_ADMINPW=Secret123
LICENSE_FILE=/tmp/license.tx
t
MSIP=$IP1
LDAP_TYPE=1
APIGEE_LDAPPW=secret
BIND_ON_ALL_INTERFACES=y
MP_POD=gateway
```

```
REGION=dc-1
ZK_HOSTS="$IP1"
ZK_CLIENT_HOSTS="$IP1"
CASS_HOSTS="$IP1"
SKIP_SMTP=n
SMTPHOST=smtp.example.com
SMTPUSER=smtp@example.com
# 0 for no username
SMTPPASSWORD=smtppwd
# 0 for no password
SMTPSSL=n
SMTPPORT=25
```

5-host clustered installation

1. Install Datastore cluster on nodes 1, 2 and 3:

   ```
   > /opt/apigee/apigee-setup/bin/setup.sh -p ds -f configFile
   ```

2. Install Management Server on node 1:

   ```
   > /opt/apigee/apigee-setup/bin/setup.sh -p ms -f configFile
   ```

3. On nodes 2 and 3:
   a. Install Router and Message Processor:

      ```
      > /opt/apigee/apigee-setup/bin/setup.sh -p rmp -f configFile
      ```

   b. Delete any files in `/opt/nginx/conf.d`:

      ```
      > rm -f /opt/nginx/conf.d/*
      ```

   c. Restart the Edge Router:

      ```
      > /<inst_root>/apigee/apigee-service/bin/apigee-service edge-router
      restart
      ```

4. Install Analytics on node 4 and 5:

   ```
   > /opt/apigee/apigee-setup/bin/setup.sh -p sax -f configFile
   ```

5. Test the installation as described at Test the install.
6. Onboard your organization as described at Onboard an organization.

Shown below is a silent configuration file for this topology:

```
# With SMTP
IP1=IPorDNSnameOfNode1
IP2=IPorDNSnameOfNode2
```

```
IP3=IPorDNSnameOfNode3
HOSTIP=$(hostname -i)
ADMIN_EMAIL=opdk@apigee.com
APIGEE_ADMINPW=Secret123
LICENSE_FILE=/tmp/license.txt
MSIP=$IP1
USE_LDAP_REMOTE_HOST=n
LDAP_TYPE=1
APIGEE_LDAPPW=secret
BIND_ON_ALL_INTERFACES=y
MP_POD=gateway
REGION=dc-1
ZK_HOSTS="$IP1 $IP2 $IP3"
ZK_CLIENT_HOSTS="$IP1 $IP2
$IP3"
CASS_HOSTS="$IP1 $IP2 $IP3"
SKIP_SMTP=n
SMTPHOST=smtp.example.com
SMTPUSER=smtp@example.com
# 0 for no username
SMTPPASSWORD=smtppwd
# 0 for no password
SMTPSSL=n
SMTPPORT=25
```

9-host clustered installation

1. Install Datastore Cluster Node on node 1, 2 and 3:

   ```
   > /opt/apigee/apigee-setup/bin/setup.sh -p ds -f configFile
   ```

2. Install Apigee Management Server on node 1:

   ```
   > /opt/apigee/apigee-setup/bin/setup.sh -p ms -f configFile
   ```

3. On nodes 4 and 5:
   a. Install Router and Message Processor:

      ```
      > /opt/apigee/apigee-setup/bin/setup.sh -p rmp -f configFile
      ```

   b. Delete any files in `/opt/nginx/conf.d`:

      ```
      > rm -f /opt/nginx/conf.d/*
      ```

   c. Restart the Edge Router:

      ```
      > /<inst_root>/apigee/apigee-service/bin/apigee-service edge-router
      restart
      ```

4. Install Apigee Analytics Qpid Server on node 6 and 7:

```
> /opt/apigee/apigee-setup/bin/setup.sh -p qs -f configFile
```

5. Install Apigee Analytics Postgres Server on node 8 and 9:

```
> /opt/apigee/apigee-setup/bin/setup.sh -p ps -f configFile
```

6. Test the installation as described at Test the install.
7. Onboard your organization as described at Onboard an organization.

Shown below is a silent configuration file for this topology:

```
# With SMTP
IP1=IPorDNSnameOfNode1
IP2=IPorDNSnameOfNode2
IP3=IPorDNSnameOfNode3
IP8=IPorDNSnameOfNode3
IP9=IPorDNSnameOfNode3
HOSTIP=$(hostname -i)
ADMIN_EMAIL=opdk@apigee.com
APIGEE_ADMINPW=Secret123
LICENSE_FILE=/tmp/license.txt
MSIP=$IP1
USE_LDAP_REMOTE_HOST=n
LDAP_TYPE=1
APIGEE_LDAPPW=secret
BIND_ON_ALL_INTERFACES=y
MP_POD=gateway
REGION=dc-1
ZK_HOSTS="$IP1 $IP2 $IP3"
ZK_CLIENT_HOSTS="$IP1 $IP2
$IP3"
CASS_HOSTS="$IP1 $IP2 $IP3"
SKIP_SMTP=n
# PG_MASTER=$IP8
# PG_STANDBY=$IP9
SMTPHOST=smtp.example.com
SMTPUSER=smtp@example.com
# 0 for no username
SMTPPASSWORD=smtppwd
# 0 for no password
SMTPSSL=n
SMTPPORT=25
```

13-host clustered installation

1. Install Datastore Cluster Node on node 1, 2 and 3:

   ```
   > /opt/apigee/apigee-setup/bin/setup.sh -p ds -f configFile
   ```

2. Install OpenLDAP on node 4 and 5:

   ```
   > /opt/apigee/apigee-setup/bin/setup.sh -p ld -f configFile
   ```

3. Install Apigee Management Server on node 6 and 7:

   ```
   > /opt/apigee/apigee-setup/bin/setup.sh -p ms -f configFile
   ```

4. Install Apigee Analytics Postgres Server on node 8 and 9:

   ```
   > /opt/apigee/apigee-setup/bin/setup.sh -p ps -f configFile
   ```

5. On nodes 10 and 11:

   a. Install Router and Message Processor:

      ```
      > /opt/apigee/apigee-setup/bin/setup.sh -p rmp -f configFile
      ```

   b. Delete any files in `/opt/nginx/conf.d`:

      ```
      > rm -f /opt/nginx/conf.d/*
      ```

   c. Restart the Edge Router:

      ```
      > /<inst_root>/apigee/apigee-service/bin/apigee-service edge-router
      restart
      ```

6. Install Apigee Analytics Qpid Server on node 12 and 13:

   ```
   > /opt/apigee/apigee-setup/bin/setup.sh -p qs -f configFile
   ```

7. Test the installation as described at Test the install.
8. Onboard your organization as described at Onboard an organization.

Shown below is a silent configuration file for this topology:

```
# For all components except
OpenLDAP
IP1=IPorDNSnameOfNode1
IP2=IPorDNSnameOfNode2
IP3=IPorDNSnameOfNode3
IP4=IPorDNSnameOfNode4
IP5=IPorDNSnameOfNode5
```

```
# For OpenLDAP on IP4 and IP5
IP1=IPorDNSnameOfNode1
IP2=IPorDNSnameOfNode2
IP3=IPorDNSnameOfNode3
IP4=IPorDNSnameOfNode4
IP5=IPorDNSnameOfNode5
IP6=IPorDNSnameOfNode6
IP7=IPorDNSnameOfNode7
```

```
IP6=IPorDNSnameOfNode6
IP7=IPorDNSnameOfNode7
IP8=IPorDNSnameOfNode8
IP9=IPorDNSnameOfNode9
HOSTIP=$(hostname -i)
ADMIN_EMAIL=opdk@apigee.com
APIGEE_ADMINPW=Secret123
LICENSE_FILE=/tmp/license.txt
# First Management Server on IP6
MSIP=$IP6
MSUM=1
USE_LDAP_REMOTE_HOST=y
LDAP_HOST=$IP4
LDAP_PORT=10389
# Second Management Server on IP7
# MSIP=$IP7
# USE_LDAP_REMOTE_HOST=y
# LDAP_HOST=$IP5
# LDAP_PORT=10389
# Same password for both OpenLDAPs.
APIGEE_LDAPPW=secret
BIND_ON_ALL_INTERFACES=y
MP_POD=gateway
REGION=dc-1
ZK_HOSTS="$IP1 $IP2 $IP3"
ZK_CLIENT_HOSTS="$IP1 $IP2 $IP3"
CASS_HOSTS="$IP1 $IP2 $IP3"
# PG_MASTER=$IP8
# PG_STANDBY=$IP9
SKIP_SMTP=n
SMTPHOST=smtp.example.com
SMTPUSER=smtp@example.com
# 0 for no username
SMTPPASSWORD=smtppwd
# 0 for no password
SMTPSSL=n
SMTPPORT=25
```

```
IP8=IPorDNSnameOfNode8
IP9=IPorDNSnameOfNode9
HOSTIP=$(hostname -i)
ADMIN_EMAIL=opdk@apigee.com
APIGEE_ADMINPW=Secret123
# First OpenLDAP Server on IP4
MSIP=$IP6
USE_LDAP_REMOTE_HOST=n
LDAP_TYPE=2
LDAP_SID=1
LDAP_PEER=$IP5
# Second OpenLDAP Server on IP5
# MSIP=$IP17
# USE_LDAP_REMOTE_HOST=n
# LDAP_TYPE=2
# LDAP_SID=2
# LDAP_PEER=$IP4
# Set same password for both
OpenLDAPs.
APIGEE_LDAPPW=secret
```

12-host clustered installation

Before you install Edge on a12-host clustered topology (two data centers), you must understand how to set the ZooKeeper and Cassandra properties in the silent config file.

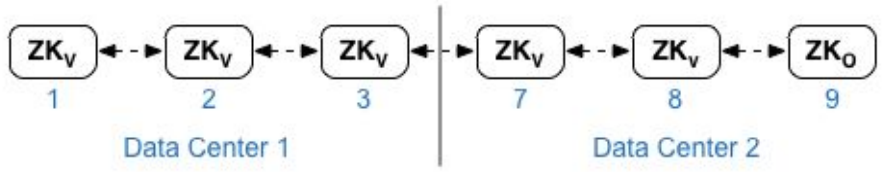
Note: Shown below is a complete config file for both data centers.

- **ZooKeeper**

  For the `ZK_HOSTS` property for both data centers, specify the IP addresses or DNS names of all ZooKeeper nodes from both data centers, in the same order, and mark any nodes with the with ":observer" modifier. Nodes without the ":observer" modifier are called "voters". You must have an odd

number of "voters" in your configuration.

In this topology, the ZooKeeper host on host 9 is the observer:



For the `ZK_CLIENT_HOSTS` property for each data center, specify the IP addresses or DNS names of only the ZooKeeper nodes in the data center, in the same order, for all ZooKeeper nodes in the data center. In the example configuration file shown below, node 9 is tagged with the ":observer" modifier so that you have five voters: Nodes 1, 2, 3, 7, and 8.

- **Cassandra**

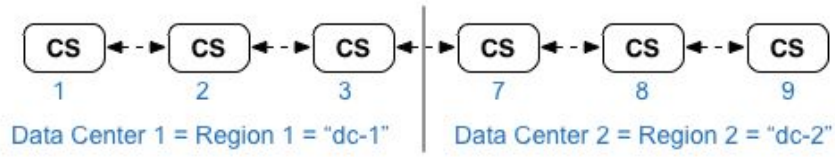
  All datacenters must to have the same number of Cassandra nodes.

  For `CASS_HOSTS` for each data center, ensure that you specify all Cassandra IP addresses or DNS names for both data centers. For data center 1, list the Cassandra nodes in that data center first. For data center 2, list the Cassandra nodes in that data center first. List the Cassandra nodes in the same order for all Cassandra nodes in the data center.

  All Cassandra nodes must have a suffix ':<d>,<r>', for example '<ip>:1,1 = datacenter 1 and rack/availability zone 1 and '<ip>:2,1 = datacenter 2 and rack/availability zone 1.
  For example, "192.168.124.201:1,1 192.168.124.202:1,1 192.168.124.203:1,1 192.168.124.204:2,1 192.168.124.205:2,1 192.168.124.206:2,1"

  The first node in rack/availability zone 1 of each datacenter will be used as the seed server.

  In this deployment model, Cassandra setup will look like this:



1. Install Datastore Cluster Node on node 1, 2, 3, 7, 8, and 9:

   ```
   > /opt/apigee/apigee-setup/bin/setup.sh -p ds -f configFile
   ```

2. Install Apigee Management Server with OpenLDAP replication on node 1 and 7:

   ```
   > /opt/apigee/apigee-setup/bin/setup.sh -p ms -f configFile
   ```

3. On nodes 2, 3, 8, and 9:

1. Install Router and Message Processor:

   ```
   > /opt/apigee/apigee-setup/bin/setup.sh -p rmp -f configFile
   ```

2. Delete any files in `/opt/nginx/conf.d`:

   ```
   > rm -f /opt/nginx/conf.d/*
   ```

3. Restart the Edge Router:

   ```
   > /<inst_root>/apigee/apigee-service/bin/apigee-service edge-router
   restart
   ```

4. Install Apigee Analytics Qpid Server on node 4, 5, 10, and 11:

   ```
   > /opt/apigee/apigee-setup/bin/setup.sh -p qs -f configFile
   ```

5. Install Apigee Analytics Postgres Server on node 6 and 12:

   ```
   > /opt/apigee/apigee-setup/bin/setup.sh -p ps -f configFile
   ```

6. Test the installation as described at Test the install.
7. Onboard your organization as described at Onboard an organization.

Shown below is a silent configuration file for this topology. Notice that this config file:

- Configures OpenLDAP with replication across two OpenLDAP nodes.
- Specifies the ":observer" modifier on one ZooKeeper node. In a single data center installation, omit that modifier.

```
# Datacenter 1
IP1=IPorDNSnameOfNode1
IP2=IPorDNSnameOfNode2
IP3=IPorDNSnameOfNode3
IP7=IPorDNSnameOfNode7
IP8=IPorDNSnameOfNode8
IP9=IPorDNSnameOfNode9
HOSTIP=$(hostname -i)
MSIP=$IP1
ADMIN_EMAIL=opdk@apigee.com
APIGEE_ADMINPW=Secret123
LICENSE_FILE=/tmp/license.txt
USE_LDAP_REMOTE_HOST=n
LDAP_TYPE=2
LDAP_SID=1
LDAP_PEER=$IP7
```

```
# Datacenter 2
IP1=IPorDNSnameOfNode1
IP2=IPorDNSnameOfNode2
IP3=IPorDNSnameOfNode3
IP7=IPorDNSnameOfNode7
IP8=IPorDNSnameOfNode8
IP9=IPorDNSnameOfNode9
HOSTIP=$(hostname -i)
MSIP=$IP7
ADMIN_EMAIL=opdk@apigee.com
APIGEE_ADMINPW=Secret123
LICENSE_FILE=/tmp/license.txt
USE_LDAP_REMOTE_HOST=n
LDAP_TYPE=2
LDAP_SID=2
LDAP_PEER=$IP1
APIGEE_LDAPPW=secret
```

```
APIGEE_LDAPPW=secret
BIND_ON_ALL_INTERFACES=y
MP_POD=gateway-1
REGION=dc-1
ZK_HOSTS="$IP1 $IP2 $IP3 $IP7 $IP8
$IP9:observer"
ZK_CLIENT_HOSTS="$IP1 $IP2 $IP3"
CASS_HOSTS="$IP1:1,1 $IP2:1,1 $IP3:1,1
$IP7:2,1 $IP8:2,1 $IP9:2,1"
SKIP_SMTP=n
SMTPHOST=smtp.example.com
SMTPUSER=smtp@example.com
# 0 for no username
SMTPPASSWORD=smtppwd
# 0 for no password
SMTPSSL=n
SMTPPORT=25
```

```
BIND_ON_ALL_INTERFACES=y
MP_POD=gateway-2
REGION=dc-2
ZK_HOSTS="$IP1 $IP2 $IP3 $IP7 $IP8
$IP9:observer"
ZK_CLIENT_HOSTS="$IP7 $IP8 $IP9"
CASS_HOSTS="$IP7:2,1 $IP8:2,1 $IP9:2,1
$IP1:1,1 $IP2:1,1 $IP3:1,1"
SKIP_SMTP=n
SMTPHOST=smtp.example.com
SMTPUSER=smtp@example.com
# 0 for no username
SMTPPASSWORD=smtppwd
# 0 for no password
SMTPSSL=n
SMTPPORT=25
```

Test the install

Apigee provides test scripts that you can use to validate your installation.

Run the validation tests

Each step of the validation testing process returns an HTTP 20X response code for a successful test.

To run the test scripts:

1. Install `apigee-validate` on a Management Server node:

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

2. Run the setup command on a Management Server node to invoke the test scripts:

   ```
   > /opt/apigee/apigee-service/bin/apigee-service apigee-validate setup -f configFile
   ```

   The ***configFile*** file must contain the following property:

   ```
   APIGEE_ADMINPW=sysAdminPword
   ```

   If omitted, you will be prompted for the password.

   By default, the `apigee-validate` utility creates a virtual host on the Router that uses port 59001. If that port is not open on the Router, you can optionally include the `VHOST_PORT` property in the config file to set the port. For example:

   ```
   VHOST_PORT=9000
   ```

3. The script then does the following:
   a. Creates an organization and associates it with the pod.
   b. Creates an environment and associates the Message Processor with the environment.
   c. Creates a virtual host.
   d. Imports a simple health check proxy and deploys the application to the “test” environment.
   e. Import the SmartDocs proxy.
   f. Executes the test to make sure everything is working as expected.

A successful test returns the 20X HTTP response.

To remove the organization, environment and other artifacts created by the test scripts:

1. Run the following command:

   ```
   > /opt/apigee/apigee-service/bin/apigee-service apigee-validate clean -f
   configFile
   ```

   where ***configFile*** is the same file you used to run the tests.

   Note: If you get errors from the testing and the troubleshooting methodology, contact Apigee Support and provide the error log.

Verify pod installation

Now that you have installed the Apigee Analytics, it is recommended that you perform following basic but important validation:

1. Verify that the Management Server is in the central POD. On Management Server, run the following CURL command:

   ```
   curl -u sysAdminEmail:password http://localhost:8080/v1/servers?pod=central
   ```

   You should see output in the form:

   ```
   [ {
    "internalIP" : "192.168.1.11",
    "isUp" : true,
    "pod" : "central",
    "reachable" : true,
    "region" : "dc-1",
    "tags" : {
      "property" : [ ]
    },
    "type" : [ "application-datastore", "scheduler-datastore",
   "management-server", "auth-datastore", "apimodel-datastore",
   "user-settings-datastore", "audit-datastore" ],
    "uUID" : "d4bc87c6-2baf-4575-98aa-88c37b260469"
   }, {
    "externalHostName" : "localhost",
    "externalIP" : "192.168.1.11",
    "internalHostName" : "localhost",
    "internalIP" : "192.168.1.11",
    "isUp" : true,
    "pod" : "central",
    "reachable" : true,
   ```

```
 "region" : "dc-1",
 "tags" : {
   "property" : [ {
     "name" : "started.at",
     "value" : "1454691312854"
   }, ... ]
 },
 "type" : [ "qpid-server" ],
 "uUID" : "9681202c-8c6e-4da1-b59b-23e3ef092f34"
} ]
```

2. Verify that the Router and Message Processor are in gateway POD. On Management Server, run the following CURL command:

   ```
   curl -u sysAdminEmail:password http://localhost:8080/v1/servers?pod=gateway
   ```

   You see output similar to the central pod but for the Router and Message Processor.

3. Verify that Postgres is in the analytics POD. On Management Server, run the following CURL command:

   ```
   curl -u sysAdminEmail:password
   http://localhost:8080/v1/servers?pod=analytics
   ```

   You see output similar to the central pod but for Postgres.

Onboard an organization

Use the `setup-org` command to perform the onboarding process. You must run the command on the Management Server node.

As part of the onboarding process, the script:

- Optionally creates a new user to function as the organization administrator.
- Creates the organization.
- Adds the specified user as the org admin. The user must already exist; otherwise the script issues an error.
- Associates the organization with a pod, by default is associates it with the "gateway" pod.
- Create an environment.
- Create a virtual host for the environment.
- Associate the environment with all Message Processor(s).
- Enables analytics.

Note: You cannot create two organizations with the same name. In that case, the second create will fail

Silent configuration file for onboarding

Pass a configuration file to the `setup-org` command. Invoke the `setup-org` command and specify the `-f` option, including the path to the silent configuration file:

```
> /opt/apigee/apigee-service/bin/apigee-service apigee-provision setup-org -f
/tmp/configFile
```

The only requirement on silent installations is that the configuration file must be accessible or readable by the "apigee" user. For example, put the file in the /tmp directory on the node.

Shown below is an example silent configuration file. Edit it as necessary for your requirements:

```
IP1=192.168.1.1

# Specify the IP or DNS name of the Management Server.
MSIP="$IP1"

# Specify the Edge admin credentials.
ADMIN_EMAIL="admin@email.com"
APIGEE_ADMINPW=adminPassword    # If omitted, you are prompted for it.

# Specify organization name and administrator.
ORG_NAME=myorg  # lowercase only, no spaces, underscores, or periods.
```

```
#
# Set the organization administrator.
# Do not use sys admin as organization administrator.
#
# Create a new user for the organization administrator.
NEW_USER="y"
# New user information if NEW_USER="y".
USER_NAME=new@user.com
FIRST_NAME=new
LAST_NAME=user
USER_PWD="newUserPword"
ORG_ADMIN=new@user.com

# Specify an existing user as the organization admin,
# omit USER_NAME, FIRST_NAME, LAST_NAME, USER_PWD.
# NEW_USER="n"
# ORG_ADMIN=existing@user.com

# Specify environment name.
ENV_NAME=prod

# Specify virtual host information.
VHOST_PORT=9001
VHOST_NAME=default

# If you have a DNS entry for the virtual host.
VHOST_ALIAS=myorg-test.apigee.net

# If you do not have a DNS entry for the virtual host,
# specify the IP and port of each router as a space-separated list:
# VHOST_ALIAS="firstRouterIP:9001 secondRouterIP:9001"

# Optionally configure SSL for virtual host.
# VHOST_SSL=y     # Set to "y" to enable SSL on the virtual host.
# KEYSTORE_JAR=   # JAR file containing the cert and private key.
# KEYSTORE_NAME=  # Name of the keystore.
# KEYSTORE_ALIAS= # The key alias.
# KEY_PASSWORD=   # The key password, if it has one.

# Specify the analytics group.
#AXGROUP=axgroup-001           # Default name is axgroup-001.
```

Notes:

- For `VHOST_ALIAS`, if you already have a DNS record that you will use to access to the virtual host, specify the host alias and optionally the port, for example, “myapi.example.com”.

  If you do not yet have a DNS record, see Setting up a virtual host for more information.

- For SSL configuration, see Keystores and Truststores and Configuring SSL access to an API for the Private Cloud for more information on creating the JAR file, and other aspects of configuring SSL.

Onboarding

1. Install `apigee-provision` on the Management Server node:

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

2. Run the command on the Management Server node:

   ```
   > /opt/apigee/apigee-service/bin/apigee-service apigee-provision setup-org
   -f configFile
   ```

   The configuration file must be accessible or readable by the "apigee" user. For example, put the file in the /tmp directory on the node.

Onboarding Verification

On completion of onboarding, verify the status of the system by issuing the following CURL commands on the Management Server node.

1. Check for user and organization status on the Management Server by issuing the following CURL commands:

   ```
   > curl -u <adminEmail>:<admin passwd> http://localhost:8080/v1/users

   > curl -u <adminEmail>:<admin passwd>
   http://localhost:8080/v1/organizations

   > curl -u <adminEmail>:<admin passwd>
   http://localhost:8080/v1/organizations/<orgname>/deployments
   ```

2. If you enabled analytics, then use this command:

   ```
   > curl -u <adminEmail>:<admin passwd>
   http://localhost:8080/v1/organizations/<orgname>/environments/<envname>/pr
   ovisioning/axstatus
   ```

3. You can also check the PostgreSQL database status by running the following command on Machine 2 to start psql:

   ```
   > psql -h /opt/apigee/var/run/apigee-postgresql -U apigee apigee
   ```

   At the command prompt, enter the following command to view the analytics table for your organization:

```
apigee=# : \d analytics."<orgname>.prod.fact"
```

Use the following command to exit psql:

```
apigee=# \q
```

4. Access the Apigee Edge user interface using a web browser. Remember that you already noted the management console URL at the end of the installation.
    a. Launch your preferred browser and enter the URL of the Edge UI. It looks similar to the following, where the IP address is for Machine 1, or for whichever machine you installed the UI on for alternative configurations:

    ```
    http://192.168.56.111:9000/login
    ```

    9000 is the port number used by the UI. If you are starting the browser directly on the server hosting the Edge UI, then you can use a URL in the form:

    ```
    http://localhost:9000/login
    ```

    Note: Ensure that port 9000 is open.

    b. On the console login page, specify the Apigee system admin username/password.

    Note: This is the global system administrator password that you have set during the installation. Alternately, you can:

    Sign in as the organization administrator that you created above when creating the organization. Sign up for a new Apigee user account and use the new user credential to login.

    c. Click Sign In, the browser redirects to:

    ```
    http://192.168.56.111:9000/platform/#/<orgname>/
    ```

    and opens a dashboard which allows you to configure the organization created before (if logged in using Apigee admin credentials).

    d. If you are new to Edge, you can now create your first API proxy. For more information, see the following tutorials:

    Create your API
    Create your API in XML

Enable Cassandra authentication

By default, Cassandra installs without authentication enabled. That means anyone can access Cassandra. You can enable authentication after installing Edge, or as part of the installation process.

If you decide to enable authentication on Cassandra, it uses the following default credentials:

- username = 'cassandra'
- password = 'cassandra'

You can use this account, set a different password for this account, or create a new Cassandra user. Add, remove, and modify users by using the Cassandra CREATE/ALTER/DROP USER statements.

For more information, see http://www.datastax.com/documentation/cql/3.0/cql/cql_reference/cqlCommandsTOC.html.

Enable Cassandra authentication during installation

You can enable Cassandra authentication as install time. However, while you can enable authentication when you install Cassandra, you cannot change the default username and password. You have to perform that step manually after installation of Cassandra completes.

Note: Use this procedure when installing Cassandra by using the "-p c", "-p ds", "-p sa", or "-p aio" options.

To enable Cassandra authentication at install time, include the `CASS_AUTH` property in the configuration file for all Cassandra nodes:

```
CASS_AUTH=y     # The default value is n.
```

The following Edge components access Cassandra:

- Management Server
- Message Processors
- Routers
- Qpid servers
- Postgres servers

Therefore, when you install these components, you must set the following properties in the configuration file to specify the Cassandra credentials:

```
CASS_USERNAME=cassandra
CASS_PASSWORD=cassandra
```

To change the Cassandra credentials after installing Cassandra:

1. Log into Cassandra using the `cqlsh` tool and the default credentials:

```
> /opt/apigee/apigee-cassandra/bin/cqlsh cassIP cassPort -u cassandra -p
cassandra
```

Where:

a. `cassIP` is the IP address of the Cassandra node.

b. `cassPort` is the Cassandra port, which by default is 9160.

c. The default user is `cassandra`.

d. The default password is `cassandra`. If you changed the password previously, use the current password.

2. Run the following command as the `cqlsh>` prompt to update the password:

```
cqlsh> ALTER USER cassandra WITH PASSWORD 'NEW_PASSWORD';
```

3. Exit the cqlsh tool:

```
cqlsh> exit
```

4. Repeat on all Cassandra nodes, ensuring that you use the same password on all nodes.

5. When installing the Management Server, Message Processors, Routers, Qpid servers, and Postgres servers, set the following properties in the config file:

```
CASS_USERNAME=cassandra
CASS_PASSWORD=NEW_PASSWORD
```

To enable Cassandra authentication post installation

To enable authentication:

- Update all Edge components that connect to Cassandra with the Cassandra username and password.
- On all Cassandra nodes, enable authentication and set the username and password.

Use the `apigee-service` utility to update all Edge components that communicate with Cassandra:

1. On the Management Server node, run the following command:

```
> /opt/apigee/apigee-service/bin/apigee-service edge-management-server
store_cassandra_credentials -u CASS_USERNAME -p CASS_PASSWORD
```

Optionally, you can pass a file to the command containing the new username and password:

```
> apigee-service edge-management-server store_cassandra_credentials  -f
configFile
```

Where the `configFile` contains the following:

```
conf_credentials_cassandra.user=cassandra
conf_credentials_cassandra.password=CASS_PASSWROD
```

This command automatically restarts the Management Server.

2. Repeat this process on the:
   - All Message Processors
   - All Routers
   - All Qpid servers (edge-qpid-server)
   - Postgres servers (edge-postgres-server)

On all Cassandra nodes, enable authentication and set the username and password:

1. Log in to the first Cassandra node.
2. Run the following command:

   ```
   > /opt/apigee/apigee-service/bin/apigee-service apigee-cassandra
   enable_cassandra_authentication -e y
   ```

   This command enables authentication and restarts Cassandra.
3. Log into Cassandra using the `cqlsh` tool and the default credentials:

   ```
   > /opt/apigee/apigee-cassandra/bin/cqlsh cassIP cassPort -u cassandra -p
   cassandra
   ```

   Where
   a. `cassIP` is the IP address of the Cassandra node.
   b. `cassPort` is the Cassandra port, which by default is 9160.
   c. The default user is `cassandra`.
   d. The default password is `cassandra`. If you changed the password previously, use the current password.
4. Run the following command as the `cqlsh>` prompt to update the password:

   ```
   cqlsh> ALTER USER cassandra WITH PASSWORD 'NEW_PASSWORD';
   ```
5. Exit the cqlsh tool:

   ```
   cqlsh> exit
   ```

6. Repeat on all Cassandra nodes, ensuring that you use the same password on all nodes:

```
> /<inst_root>/apigee/apigee-service/bin/apigee-service apigee-cassandra
restart
```

Set up Master-Standby Replication for Postgres

By default, Edge installs all Postgres nodes in master mode. However, in most production systems, you configure them to use master-standby replication so that if the master node fails, the standby node can continue to server traffic.

If the master node ever fails, you can promote the standby server to the master. See the section "Handling a PostgresSQL Database Failover" in the *Edge Operations Guide* for more information.

To configure Master-Standby Replication at install time

You can configure master-standby replication at install time by including the following properties in the config file for the two Postgres nodes:

```
PG_MASTER=IPorDNSofNewMaster
PG_STANDBY=IPorDNSofOldMaster
```

The installer automatically configures the two Postgres node to function as master-standby with replication.

To configure Master-Standby Replication after installation

You can configure master-standby replication after installation by by using the following procedure:

1. Identify which Postgre node will be the master and which will be the standby server.
2. On the master node, edit the config file to set:

   ```
   PG_MASTER=IPorDNSofNewMaster
   PG_STANDBY=IPorDNSofOldMaster
   ```

3. Enable replication on the new master:

   ```
   > /opt/apigee/apigee-service/bin/apigee-service apigee-postgresql
   setup-replication-on-master -f configFIle
   ```

4. On the standby node, edit the config file to set:

   ```
   PG_MASTER=IPorDNSofNewMaster
   PG_STANDBY=IPorDNSofOldMaster
   ```

5. Stop the standby node:

   ```
   > /opt/apigee/apigee-service/bin/apigee-service apigee-postgresql stop
   ```

6. On the standby node, delete any existing Postgres data:

   ```
   > rm -rf /opt/apigee/data/apigee-postgresql/
   ```

Note: If necessary, you can backup this data before deleting it.

7. Configure the standby node:

```
> /opt/apigee/apigee-service/bin/apigee-service apigee-postgresql
setup-replication-on-standby -f configFile
```

Test Master-Standby Replication

On completion of replication, verify the replication status by issuing the following scripts on both servers. The system should display identical results on both servers to ensure a successful replication:

1. On the master node, run:

```
> /opt/apigee/apigee-service/bin/apigee-service apigee-postgresql
postgres-check-master
```

   Validate that it says it is the master.

2. On the standby node:

```
> /opt/apigee/apigee-service/bin/apigee-service apigee-postgresql
postgres-check-standby
```

   Validate that it says it is the standby.

Install SmartDocs

SmartDocs is installed automatically when you install and run the installation test scripts described in Test the install. As part of running the test scripts, you run the following command:

```
> /opt/apigee/apigee-service/bin/apigee-service apigee-validate setup
```

This command installs SmartDocs as part of running the tests.

To test that SmartDocs is installed, confirm that the `smartdocs.zip` file is located in the following directory:

```
/opt/apigee/apigee-validate/bundles/
```

Or run the following API call on the Management Server node:

```
> curl -v -u adminEmail:adminPword 0:8080/v1/o/VALIDATE/apis
```

This command should return the following if SmartDocs is installed:

```
[ "smartdocs", "passthrough" ]
```

7-host and 10-host API BaaS Installation

Note: The Installation Checklist details the installation prerequisites and provides a list of required files to obtain before proceeding with the installation. Ensure that you have reviewed the checklist before beginning the installation process.

Using a Load Balancer

A production installation of API BaaS uses a load balancer between the API BaaS Portal node and API BaaS Stack nodes. When configuring the Portal, you specify the IP address or DNS name of the load balancer, not of the Stack nodes.

As an alternative to a load balancer, you could use round-robin DNS. In this scenario, you create a DNS entry with multiple A records corresponding to BaaS stack IP addresses. During a DNS lookup, the DNS server automatically returns A record values in a round robin fashion.

Note: API BaaS testing and development environments can install all components on a single node, or with a single API BaaS Stack node. If your installation uses a single API BaaS Stack node, the load balancer is not required and you can specify the IP address or DNS name of the API BaaS Stack when configuring the Portal.

Connecting to Cassandra

While you can connect API BaaS and Edge to the same Cassandra cluster, Apigee recommends that you use separate clusters. Separate clusters maximize performance if you are experiencing high traffic loads on API BaaS.

Date synchronization

You must have the date/time on all servers synchronized. If not already configured, 'ntpdate' utility could serve this purpose, which verifies whether servers are time synchronized. You can use "`yum install ntp`" to install the utility.

Tomcat security

The API BaaS installer also installs the Apache Tomcat server on all API BaaS Stack nodes, including the Tomcat administrator UI. The installer leaves the default administrator credentials unchanged from `admin:admin`.

If necessary, you can change these credentials as part of securing Tomcat. For more information, see:

- https://tomcat.apache.org/tomcat-7.0-doc/manager-howto.html
- https://www.owasp.org/index.php/Securing_tomcat

Installation overview

After you install the Edge the `apigee-setup` utility on a node, use that utility to install one or more BaaS components on the node. The `apigee-setup` utility has the form:

```
> sudo /opt/apigee/apigee-setup/bin/setup.sh -p component -f configFile
```

Pass a configuration file to the `apigee-setup` utility that contains the information about the installation. If the configuration file is missing any required information, the `apigee-setup` utility prompts you to enter it on the command line.

The only requirement is that the configuration file must be accessible or readable by the "apigee" user. For example, put the file in the /tmp directory on the node.

For example, use the following command to install the API BaaS Stack:

```
> sudo /opt/apigee/apigee-setup/bin/setup.sh -p b -f myConfig
```

The Apigee `setup.sh` utility supports several options for installing API BaaS components. The instructions below use the standalone options (c, e, b, and p) but you can use different options based on your node configuration:

Option	Description
e	Install ElasticSearch only.
b	Install API BaaS Stack only, which also installs Tomcat.
p	Install API BaaS Portal only, which also installs the Nginx router to be used as a web server.
c	Install Cassandra only.
eb	Install ElasticSearch, API BaaS Stack, and Tomcat on the node.
ebp	Install ElasticSearch, API BaaS Portal, API BaaS Stack, and Tomcat. The portal is so lightweight no additional resources needed for this.
asa	Install all API components on a single node (Cassandra, Elasticsearch, API BaaS Stack, and API BaaS Portal). Use this option for development and testing only, not for production.

Creating a silent configuration file

Shown below is an example silent configuration file for a 10-node API BaaS installation. Edit this file as necessary for your configuration. Use the `-f` option to `setup.sh` to include this file.

```
# Specify IP address or DNS name of node.
IP1=192.168.1.1   # ElasticSearch
IP2=192.168.1.2   # ElasticSearch
IP3=192.168.1.3   # ElasticSearch
IP4=192.168.1.4   # API BaaS Stack
IP5=192.168.1.5   # API BaaS Stack
IP6=192.168.1.6   # API BaaS Stack
IP7=192.168.1.7   # API BaaS Portal
IP8=192.168.1.8   # Cassandra (shared with Edge or standalone)
IP9=192.168.1.9   # Cassandra (shared with Edge or standalone)
IP10=192.168.1.10  # Cassandra (shared with Edge or standalone)

# Must resolve to IP address or DNS name of host - not to 127.0.0.1 or localhost.
HOSTIP=$(hostname -i)

# Define the API BaaS administrator account.
AS_ADMIN="superuser"          # User name - default is "superuser".
AS_ADMIN_EMAIL=stackAdmin@email.com
AS_PASSWD=stackAdminPWrod

# Only if you are installing Cassandra.
# Specify Cassandra configuration information.
# CASS_HOSTS="$IP8:1,1 $IP9:1,1 $IP10:1,1"

# If connecting to existing Cassandra nodes,
# specify Cassandra IPs.
CASS_HOSTS="$IP8 $IP9 $IP10"

# Cassandra uname/pword.
# Even if Cassandra authentication is disabled,
# you must still pass values for these properties.
CASS_USERNAME=cassandra        # Default value
CASS_PASSWORD=cassandra        # Default value

# Specify BaaS Cassandra connection information.
# Specify the data center name.
BAAS_CASS_LOCALDC=dc-1                    # Default is dc-1.

# Replication is in the form "dataCenterName:#CassandraNodes".
# For example, for dc-1 with three Cassandra nodes, it is dc-1:3.
BAAS_CASS_REPLICATION=dc-1:3

# ElasticSearch IPs or DNS names, separated by spaces.
ES_HOSTS="$IP1 $IP2 $IP3"
```

```
# API BaaS Stack information.
# Default cluster name is "apigee_baas"
BAAS_USERGRID_CLUSTERNAME="apigee_baas"

# URL and port of the load balancer for the API BaaS Stack nodes,
# or IP/DNS and port 8080 of a single Stack node with no load balancer.
BAAS_USERGRID_URL=http://myloadbalancer:8443

# API BaaS Portal information.
# URL and port number of load balancer, if there is one in front of the Portal,
# or the URL and port of the Portal node.
BAAS_PORTAL_URL="http://$IP7:9000"

# Portal port. Default value is 9000.
BAAS_PORTAL_LISTEN_PORT=9000

# SMTP information. BaaS requires an SMTP server.
SMTPHOST=smtp.gmail.com
SMTPPORT=465
SMTPUSER=your@email.com
SMTPPASSWORD=yourEmailPassword
SMTPSSL=y
```

The following table contains additional information about these properties:

Property	Note
CASS_HOSTS	If you are installing Cassandra, specify the Cassandra node IPs and include the ":dc,ra" modifier that specify the data center and rack of the Cassandra node. For example '192.168.124.201:1,1 = datacenter 1 and rack/availability zone 1, and '192.168.124.204:2,1 = datacenter 2 and rack/availability zone 1.
CASS_USERNAME CASS_PASSWORD	Cassandra user name and password. If Cassandra authentication is disabled, you still have to pass these values. However, the values are ignored.
BAAS_CASS_LOCALDC	The region names must be in the form dc-# where # corresponds to an integer value. For example, dc-1, dc-2, etc. If you are connecting to a Cassandra cluster installed with Edge, you can ask the Edge system administrator for this value. In an Edge single data center installation, the default value is dc-1. If you installed Cassandra as part of installing the API BaaS, then during Cassandra installation you added the ":dc,ra" modifier to the Cassandra IP addresses. The first value "dc" is the data center

	number. The data center name is the string "dc-" with the data center number as a suffix.
`BAAS_CASS_REPLICATION`	The format is "`dataCenterName:#CassandraNodes`". For example, for dc-1 with three Cassandra nodes, it is dc-1:3.

`BAAS_USERGRID_URL`	In a production environment, this is the URL and port of the load balancer that is in front of the API BaaS Stack nodes, in the form:
	
	`http://myStackLoadBalancer:port`
	
	In a testing or development environment, where you only have a single API BaaS Stack node, this can be the URL and port number of an API BaaS Stack node, in the form:
	
	`http://stackIPorDNS:8080`
	
	The port number for the API BaaS Stack server is 8080.

`BAAS_PORTAL_URL`	The URL and port number of the load balancer, if there is one in front of the Portal, in the form:
	
	`http://myPortalLoadBalancer:port`
	
	If there is no load balancer, the URL and port number of the Portal node, in the form:
	
	`http://portalIPorDNS:9000`
	
	By default, the port number for the API BaaS Portal is 9000.

`BAAS_PORTAL_LISTEN_PORT`	The port number for the API BaaS Portal server is 9000. If this port is not available, specify a different port. If you are setting `BAAS_PORTAL_URL` to the URL of the Portal node, the port numbers must be the same for both properties.

Optional - Install Cassandra: Machine 8, 9, and 10

While you can connect API BaaS to the same Cassandra cluster as used by Edge, Apigee recommends that you use separate clusters.

The Cassandra cluster can use authentication, or Cassandra authentication can be disabled. See Enable Cassandra authentication for more.

1. Install the Edge `apigee-setup` utility on the node using the internet or non-internet procedure. See Install the Edge apigee-setup utility for more.

2. At the command prompt, run the setup script:

   ```
   > /opt/apigee/apigee-setup/bin/setup.sh -p c -f configFile
   ```

   The “-p c” option specifies to install Cassandra.

   The configuration file must be accessible or readable by the "apigee" user. For example, put the file in the `/tmp` directory on the node.

The configuration successfully completes the datastore setup on the node.

Note that JMX is enabled by default for Cassandra. The JMX remote access to Cassandra does not require a password. You can configure Cassandra to use authentication for JMX. For more, see http://docs.apigee.com/api-services/latest/how-monitor.

Set up Cassandra cron job

Set up a cron job that uses `nodetool` to flush for locks to run every hour on every Cassandra node.

Note: This step is required. You must set up this cron job on Cassandra nodes even if you are connecting to Cassandra nodes on an Edge installation.

If you have multiple Cassandra nodes, offset the cron job on each server by five minutes so that all nodes do not flush at the same time.

The cron job must execute the following command:

```
/opt/apigee/apigee-cassandra/bin/nodetool -h IP_address flush Apigee_Baas_Locks
```

where ***IP_address*** is the IP address of the Cassandra node.

Install ElasticSearch: Machine 1, 2, and 3

Note: If you are installing ElasticSearch and the API BaaS Stack on the same node, you can use the "-p eb" option to the `setup` utility to install them both at the same time.

To install the ElasticSearch:

1. Install the Edge `apigee-setup` utility on the node using the internet or non internet procedure. See Install the Edge apigee-setup utility for more.
2. At the command prompt, run the setup script:

   ```
   > /opt/apigee/apigee-setup/bin/setup.sh -p e -f configFile
   ```

   The “-p e” option specifies to install ElasticSearch.

   The configuration file must be accessible or readable by the "apigee" user. For example, put the file in the `/tmp` directory on the node.

3. (Optional) If you install ElasticSearch on a standalone node, meaning it is not installed with API BaaS Stack, then adjust the default memory option to increase the memory allocated for ElasticSearch from 4GB to 6GB:

    a. Open `/opt/apigee/customer/application/elasticsearch.properties` in an editor. If this file does not exist, create it.

    b. Set the `setenv_elasticsearch_max_mem_size` property to 6g (the default is 4g):

    ```
    setenv_elasticsearch_max_mem_size=6g
    ```

    c. Save the file.

    d. Run the following command:

    ```
    > /opt/apigee/apigee-service/bin/apigee-service apigee-elasticsearch
    restart
    ```

The configuration successfully completes the setup on the node.

Install API BaaS Stack: Machine 4, 5, and 6

Note: If you are installing ElasticSearch and the API BaaS Stack on the same node, you can use the "-p eb" option to the `setup` utility to install them both at the same time.

To install the API BaaS Stack:

1. Install the Edge `apigee-setup` utility on the node using the internet or non internet procedure. See Install the Edge apigee-setup utility for more.
2. At the command prompt, run the setup script:

    ```
    > /opt/apigee/apigee-setup/bin/setup.sh -p b -f configFile
    ```

    The “-p b” option specifies to install API BaaS Stack.

    The configuration file must be accessible or readable by the "apigee" user. For example, put the file in the `/tmp` directory on the node.

After the installer fetches the correct admin credentials, it installs Tomcat, creates API BaaS keyspaces, and sets up the API BaaS Stack on the server. SMTP is also configured to allow the UI to send password confirmation emails.

Install API BaaS Portal: Machine 7

To install the API BaaS Portal:

1. Install the Edge `apigee-setup` utility on the node using the internet or non internet procedure. See Install the Edge apigee-setup utility for more.
2. At the command prompt, run the setup script:

   ```
   > /opt/apigee/setup/bin/setup.sh -p p -f configFile
   ```

   The “-p p” option specifies to install API BaaS Portal.

   The configuration file must be accessible or readable by the "apigee" user. For example, put the file in the `/tmp` directory on the node.

The installer starts the Nginx webserver and then finishes the API BaaS Portal configuration.

Make a note of the API BaaS Portal URL. This is the URL you enter into a browser to access the API BaaS Portal user interface.

Onboarding a new organization

Onboarding is the process of creating an organization and organization administrator. After creating the organization and organization administrator, you can log in to the API BaaS Portal UI and make requests to the API BaaS REST API.

When you create an organization, the organizat administrator's email address:

- Must be different from the system administrator's email address.
- Must be unique among all other organizations. That is, you cannot create two organizations with the same email address for the organization administrator. However, after creating the organization, you can add additional administrators that can be duplicated across multiple organizations.

To perform onboarding, use the `create_org_and_user.py` Python script. Invoking this script with no command-line arguments causes it to prompt you for all information:

```
> python create_org_and_user.py
```

Alternatively, you can pass any or all options as command line argument. You are prompted for any information that you omit from the command line:

```
> python create_org_and_user.py -o '<org name>'
```

```
> python create_org_and_user.py -o '<org name>' -a '<new admin email>' -p '<new admin password>'
```

To create an organization:

1. Change directory to `/opt/apigee/baas-usergrid/bin`.
2. Invoke the `create_org_and_user.py` Python script.

You are prompted for the BaaS system administrator username and password so that only a sys admin can run it.

3. Log in to the API BaaS Portal in a web browser using the URL you noted at the end of the API BaaS Portal URL installation.

   To access the portal, enter the API BaaS Portal URL in the form:

   ```
   http://{portalExternalIP}:9000/
   ```

   Note: The IP is the external IP address/host name of Portal machine. Ensure that port is open.

4. When the portal login screen appears, you can either:
   a. Log in using the organization administrator's username and password.
   b. Log in using the system administrators administrator's username and password.

Accessing the API BaaS REST API

To access the API BaaS REST API, use a URL in the form:

```
https://{loadBalancerIP}:8080/{your-org}/{your-app}
```

In a development environment, you can install all API BaaS components on a single node, meaning you have a single API BaaS Stack. Or, you might have a small environment with a single API BaaS Stack node and no load balancer. In these types of environments, you can make API calls directly to the API BaaS Stack node:

```
curl -v "http://portalExternalIP:8080/status"
```

For more information on getting started with API BaaS Portal, see the Apigee documentation at: http://apigee.com/docs/content/build-apps-home.

Installing Monetization Services

Note: The Installation Checklist details the installation prerequisites and provides a list of required files to obtain before proceeding with the installation. Ensure that you have reviewed the checklist before beginning the installation process.

Monetization Services is an extension to Apigee Edge, hence it does not run as a standalone process. It runs within any existing Apigee Edge setup.

Monetization requirements

- If you are installing Monetization on an Edge topology that use multiple Management Server nodes, such as a 13-node installation, then you must install both Edge Management Server nodes before installing Monetization.
- To install Monetization on Edge where the Edge installation has multiple Postgres nodes, the Postgres nodes must be configured in Master/Standby mode. You cannot install Monetization on Edge if you have multiple Postgres master nodes. For more, see Set up Master-Standby Replication for Postgres.

Installation overview

The following steps illustrate how to add Monetization Services on an existing Apigee Edge installation:

- Use the `apigee-setup` utility to update the Apigee Management Server node to enable the Monetization Services, for example, catalog management, limits and notifications configuration, billing and reporting.

  If you have multiple Management Server nodes, such as a 13-node installation, then you must install both Edge Management Server nodes before installing Monetization.
- Use the `apigee-setup` utility to update the Apigee Message Processor to enable the runtime components of the Monetization Services, for example, transaction recording policy and limit enforcement. If you have multiple Message Processors, install Monetization on all of them.
- Perform the Monetization onboarding process for your Edge organizations.
- Configure the Developer Services portal to support monetization. For more information, see http://apigee.com/docs/monetization/content/configure-monetization-developer-portal.

Creating a silent configuration file for Monetization

Shown below is an example silent configuration file for a Monetization installation. Edit this file as necessary for your configuration. Use the `-f` option to `setup.sh` to include this file.

Note: Typically, you add these properties to the same configuration file that you used to install Edge, as shown in Creating a configuration file.

```
# Edge configuration properties
# Specify IP address or DNS name of node.
IP1=192.168.1.1  # Management Server, OpenLDAP, UI, ZooKeeper, Cassandra
IP2=192.168.1.2  # ZooKeeper, Cassandra
IP3=192.168.1.3  # ZooKeeper, Cassandra
IP4=192.168.1.4  # Router, Message Processor
IP5=192.168.1.5  # Router, Message Processor
IP6=192.168.1.6  # Qpid
IP7=192.168.1.7  # Qpid
IP8=192.168.1.8  # Postgres
IP9=192.168.1.9  # Postgres

# Must resolve to IP address or DNS name of host - not to 127.0.0.1 or localhost.
HOSTIP=$(hostname -i)

# Edge sys admin credentials
ADMIN_EMAIL=your@email.com
APIGEE_ADMINPW=yourPassword    # If omitted, you are prompted for it.

#

# Monetization configuration properties.

#

# Postgres credentials from Edge installation.
PG_USER=apigee           # Default from Edge installation
PG_PWD=postgres          # Default from Edge installation

# Specify Postgres server.
MO_PG_HOST="$IP8"  # Only specify one Postgres node.

# Create a Postgres user for Monetization.
MO_PG_USER=postgre # Default username is "postgre"
MO_PG_PASSWD=moUserPWord

# Specify one ZooKeeper host.
# Ensure this is a ZooKeeper leader node in a multi-datacenter environment.
ZK_HOSTS="$IP2"

# Specify Cassandra information.
# Ensure CASS_HOSTS is set to the same value as when you installed Edge.
CASS_HOSTS="$IP1:1,1 $IP2:1,1 $IP3:1,1"
CASS_CLUSTERNAME=Apigee              # Default is "Apigee", unless it
                                     # was changed during Edge install.

# Cassandra uname/pword required only if you enabled Cassandra authentication.
# CASS_USERNAME=
# CASS_PASSWORD=
```

```
# Specify the region.
# Default is dc-1 unless you are in a multi-datacenter environment.
REGION=dc-1

# If your Edge config file did not specify SMTP information, add it.
# Monetization requires an SMTP server.
SMTPHOST=smtp.gmail.com
SMTPPORT=465
SMTPUSER=your@email.com
SMTPPASSWORD=yourEmailPassword
SMTPSSL=y
```

Notes:

- If your Edge config file did not specify SMTP information, add it. Monetization requires an SMTP server.
- In a single data center installation, all ZooKeeper nodes are by default configures as leaders. When you are installing Edge across multiple data centers, some ZooKeeper nodes will be configured as observers. Ensure that the `ZK_HOSTS` property above specifies a leader node in a multiple data center installation.
- If you enable Cassandra authentication, you can pass the Cassandra user name and password by using the following properties:
    - `CASS_USERNAME`
    - `CASS_PASSWORD`

Integrate Monetization Services with all Management Servers

Use the following procedure to integrate monetization on Management Server nodes.

1. If you are installing Monetization on an Edge topology that uses multiple Management Server nodes, such as a 13-node installation, then ensure that you installed both Management Server nodes before installing Monetization.
2. On the Management Server node, run the setup script:

   ```
   > /opt/apigee/apigee-setup/bin/setup.sh -p mo -f configFile
   ```

   The “-p mo” option specifies to integrate Monetization.

   The configuration file must be accessible or readable by the "apigee" user. For example, put the file in the `/tmp` directory on the node.
3. If you are installing Monetization on multiple Management Server nodes, repeat step 2 on the second Management Server node.

On successful configuration, an RDBMS schema for Monetization Services is created in the PostgreSQL database. This completes the integration of Monetization Services and its associated components with Postgres Server.

Integrate Monetization Services with all Message Processors

Use the following procedure to integrate monetization on all Message Processor nodes.

1. On the first Message Processor node, at the command prompt, run the setup script:

   ```
   > /opt/apigee/apigee-setup/bin/setup.sh -p mo -f configFile
   ```

   The “-p mo” option specifies to integrate Monetization.

   The configuration file must be accessible or readable by the "apigee" user. For example, put the file in the `/tmp` directory on the node.
2. Repeat this procedure on all Message Processor nodes.

On successful configuration, the Message Processor is updated with Monetization Services. This completes the integration of Monetization Services and its associated components with the Message Processors.

Monetization Onboarding

To create a new organization with monetization enabled, you first create the organization as you would for any new organization. For more information, see Onboard an organization.

Additional Onboarding to enable Monetization for an organization

To complete monetization onboarding of an organization, you have to:

1. Create the monetization group: `mxgroup`.
2. Add Qpid to the group.
3. Enable monetization for the organization.
4. Enable notification settings for the organization.
5. Repeat this process for all organizations where you want to enable Monetization.

Use the `enable-monetization` command to perform all of these tasks. This script takes a configuration file containing the following properties:

```
MSIP=IPorDNSofManagementServer
APIGEE_PORT_HTTP_MS=8080       # Default is 8080.
ADMIN_EMAIL=your@email.com
APIGEE_ADMINPW=yourPassword          # If omitted, you are prompted for it.
CASS_HOSTS="$IP1:1,1 $IP2:1,1 $IP3:1,1"
QPID_HOST="$IP6 $IP7"                # Space-separated list IP/DNS names of
```

```
                                    # all QPid nodes.
QPID_PORT=8083          # Default is 8083.
REGION=dc-1
ORG_NAME=myorg          # The Edge org where you want to enable monetization.
MX_GROUP=mxgroup        # Default Monetization group.
```

Notes:

- Set `CASS_HOSTS` and `REGION` to the same values as you used when installing Monetization.

To run the script:

1. Invoke the script:

   ```
   > /opt/apigee/apigee-service/bin/apigee-service apigee-provision
   enable-monetization -f configFile
   ```

   The configuration file must be accessible or readable by the "apigee" user. For example, put the file in the `/tmp` directory on the node.

   This script replicates the organization, products, developers and applications from Cassandra database to Monetization PostgreSQL database. After successful installation of Monetization Services the data is synchronized automatically.

When you next log in to the Edge UI, you see the Monetization entry in the top-level menu for the organization:

Configure the Developer Services portal

To configure the Developer Services portal to support monetization, see http://apigee.com/docs/monetization/content/configure-monetization-developer-portal.

Adding a Management Server node to a Monetization Installation

If you add a Management Server to an existing Edge installation, you must ensure that you add Monetization services to the new Management Server and configure all Management Servers so they can communicate.

To add a Management Server:

1. Install the **new** Management Server.
2. Install Monetization on the **new** Management Server.

3. On the **original** Management Server, call the following:

   ```
   > /opt/apigee/apigee-service/bin/apigee-service
   edge-mint-management-server mint-configure-mgmt-cluster
   ```

4. Restart the **original** Management Server:

   ```
   > /opt/apigee/apigee-service/bin/apigee-service edge-management-server
   restart
   ```

5. On the **new** Management Server, call the following:

   ```
   > /opt/apigee/apigee-service/bin/apigee-service
   edge-mint-management-server mint-configure-mgmt-cluster
   ```

6. Restart the **new** Management Server:

   ```
   > /opt/apigee/apigee-service/bin/apigee-service edge-management-server
   restart
   ```

Additional configuration

Provide Billing Documents as PDF Files

Monetization displays billing documents to end users in HTML format. To provide billing documents as PDF files, you can integrate Monetization with a billing system that provides PDF generation or license a supported third-party PDF library.

Configure Organization Settings

- Backend settings: The following table (**Table 3:** Organization-level attributes) lists the organization level attributes that are available to configure a mint organization. You can use a `PUT` call to add/update these attributes as.

  ```
  curl -u ${ADMIN_EMAIL}:${ADMINPW} -v
  http://<management-ip>:8080/v1/organizations/{orgId} -d '{org object with
  attributes}' -X PUT
  ```

  For example, the output of the above CURL command will look something like this:

  ```
  {
        ...
            "displayName": "Orgnization name",
            "name": "org4",
            "properties": {
  ```

```
            "property": [
               ...
                {
                    "name": "MINT_CURRENCY",
                    "value": "USD"
                },
                {
                    "name": "MINT_COUNTRY",
                    "value": "US"
                },
                {
                    "name": "MINT_TIMEZONE",
                    "value": "GMT"
                }
            ]
        }
}
```

Table 3: Organization-level attributes

Attributes	Description
MINT_TAX_MODEL	Accepted values are DISCLOSED, UNDISCLOSED, HYBRID (default is null)
MINT_CURRENCY	ISO currency code (default is null)
MINT_TAX_NEXUS	Tax nexus (default is null)
MINT_DEFAULT_PROD_TAX_CATEGORY	Default product tax category (default is null)
MINT_IS_GROUP_ORG	IS group organization (default is false)
MINT_HAS_BROKER	Has broken (default is false)
MINT_TIMEZONE	Timezone (default is null)
MINT_TAX_ENGINE_EXTERNAL_ID	Tax engine ID (default is null)
MINT_COUNTRY	Organization's country (default is null)
MINT_REG_NO	Organization's registration number, United Kingdom gives different number than tax ID (default is null)
MINT_BILLING_CYCLE_TYPE	PRORATED, CALENDAR_MONTH (default is CALENDAR_MONTH)

MINT_SUPPORTED_BILLING_TYPE	PREPAID/POSTPAID/BOTH (default is PREPAID)
MINT_IS_SEPARATE_INV_FOR_FEES	Indicates whether a separate fee invoice should be generated (default is false)
MINT_ISSUE_NETTING_STMT	Indicates whether netting statement should be issued (default is false)
MINT_NETTING_STMT_PER_CURRENCY	Indicates whether netting statement should be generated per currency (default is false)
MINT_HAS_SELF_BILLING	Indicates whether the organization has self billing (default is false)
MINT_SELF_BILLING_FOR_ALL_DEV	Indicates whether the organization has self billing for all developers(default is false)
MINT_HAS_SEPARATE_INV_FOR_PROD	Indicates whether the organization has separate invoice per product (default is false)
MINT_HAS_BILLING_ADJUSTMENT	Indicates whether the organization supports billing adjustments (default is false)
features.isMonetizationEnabled	Used by the management UI to display monetization specific menu (default is false)
ui.config.isOperator	Used by management UI to display provider as Operator verses Organization (default is true)

- For configuring business organization settings using the management UI, see http://apigee.com/docs/monetization-services/content/get-started-using-monetization-services.

Note: If you are using Monetization Services Limits and Notifications features, please instruct your developers to attach a Limit Policy in the proxy flow after the access token validation policy.

Limit Policy is an explicit policy designed to block an API call if certain limit has been reached. The policy checks business limits and raises a fault if there are any limits exceeding the configured value. This is an extension of raise fault policy but the conditions are derived from business variables.

An UI template is available in the management UI for proxy developers. Proxy developer should attach mint policy in the message flow. Upon execution of this policy the fault will be raised with the fault response as per policy. If `ContinueOnError` is set to true then the fault will not be raised and flow variables "`mint.limitsViolated`", "`mint.isDeveloperSuspended`" and "`mint.limitsPolicyError`" variables will be set which could be used for further exception handling if required.

Updating Apigee Edge to 4.16.05

This section explains the update and rollback processes from one version of Apigee Edge to another version.

Which Edge versions can you update to 4.16.05

You can only update Apigee Edge version 4.16.01.x to 4.16.05.

If you have a version of Edge previous to version 4.16.01.x, then you must first migrate to version 4.16.01.x and then update to version 4.16.05.

Who can perform the update

The user running the update should be the same as the user who originally installed Edge, or a user running as root.

After you install the Edge RPMs, any user can configure them.

Required upgrade to Java JDK Version 8

This release of Edge requires that you have installed Java JDK version 8 on all Edge processing nodes. You can install the Oracle JDK 8 or OpenJDK 8. If Java JDK 8 is not installed already, the update script can install it for you.

As part of the update to Java 8, some TLS ciphers are no longer available in Oracle JDK 1.8. For the complete list, see the section "Default Disabled Cipher Suites" http://docs.oracle.com/javase/8/docs/technotes/guides/security/SunProviders.html.

Warning: This release of Edge does not support JDK 7. If you are currently using JDK 7, you must upgrade to JDK 8. If you rollback the Edge 4.16.05 installation, you can optionally reconfigure Edge to use Java JDK 7.

Disk space requirements for update

Ensure that you have at least 1 GBytes of free disk space before you perform the update.

Automatic propagation of property settings from 4.16.01.x

If you have set any properties by editing `.properties` files in `/opt/apigee/customer/application` then these values are retained by the update.

Updating the apigee-validate utility

In 4.16.01, you installed and ran the `apigee-validate` utility on a **Message Processor** node. In 4.16.05, the `apigee-validate` utility has been updated to run on the **Management Server** node.

When you update to 4.16.05, you have two options on how you update the `apigee-validate` utility:

1. **Apigee recommended** - Install and run the `apigee-validate` utility on the Management Server node.

   You can optionally uninstall the `apigee-validate` utility the from the Message Processor nodes. If you leave it on the Message Processor node, you must update it to 4.16.05.

2. Alternatively, update the `apigee-validate` utility on the Message Processor node, and run it from there. However, Apigee recommends that you install and run it from the Management Server.

Update prerequisites

Take care of following prerequisites before upgrading Apigee Edge:

- **Backup all nodes**

  Before you update, it is recommended to perform a complete backup of all nodes for safety reasons. Use the procedure for your current version of Edge to perform the backup.

  This allows you to have a backup plan, in case the update to a new version doesn't function properly. For more information on backup, see http://docs.apigee.com/api-services/latest/backup-and-restore.

- **Ensure Edge is running**

  Ensure that Edge is up and running during update process by using the command:

  ```
  > /<inst_root>/apigee/apigee-service/bin/apigee-all status
  ```

Handling a failed update

In the case of an update failure, you can try to correct the issue, and then run `update.sh` again. You can run the update multiple times and it continues the update from where it last left off.

If the failure requires that you roll back the update to your previous version, see Rollback Process for more.

Logging update information

By default, the `update.sh` utility writes log information to:

`/opt/apigee/var/log/apigee-setup/update.log`

If the user running the `update.sh` utility does not have access to that directory, it writes the log to the `/tmp` directory as a file named `update_`***`username`***`.log`.

If the user does not have access to /tmp, the `update.sh` utility fails.

Zero-downtime update

A zero-downtime update, or rolling update, lets you update your Edge installation without bringing down Edge.

Note: Zero-downtime update is only possible with a 5-node configuration and larger.

The key to zero-downtime upgrading is to remove each Router, one at a time, from the load balancer. You then update the Router and any other components on the same machine as the Router, and then add the Router back to the load balancer.

1. Update the machines in the correct order for your installation as described in Order of machine update.
2. When it is time to update the Routers, select any one Router and make it unreachable, as described below in Making a Router and Message Processor unreachable.
3. Update the selected Router and all other Edge components on the same machine as the Router. All Edge configurations show a Router and Message Processor on the same node.
4. Make the Router reachable again.
5. Repeat steps 2 through 4 for the remaining Routers.
6. Continue the update for any remaining machines in your installation.

Making a Router and Message Processor unreachable

In a production setup, you will have multiple Routers and Message Processors to achieve optimal performance and you must enable/disable reachability of these Routers and Message Processors before/after update.

The following API call configures a node as reachable or unreachable:

```
> curl -u adminEmail:pWord -X POST "http://<ms_IP>:8080/v1/servers/UUID" -d
"reachable=true|false"
```

where *UUID* is the UUID of the Message Processor or Router, and `reachable` is set to either true or false.

If you need to determine the UUID of the Router, use the following cURL command:

```
> curl http://<routerIP>:8081/v1/servers/self
```

If you need to determine the UUID of the Message Processor, use the following cURL command:

```
> curl http://<mpIP>:8082/v1/servers/self
```

Take care of the following before/after update:

- On combined Router and Message Processor node:
    - Before update – perform the following:
        i. Make the Router unreachable by using the API call shown above.
        ii. Make the Message Processor unreachable.
    - After update - perform the following:

I. Make the Message Processor reachable.

II. Make the Router reachable.

- On single Router node:
  - Before update, make the Router reachable.
  - After update, make the Router reachable.
- On single Message Processor node:
  - Before update, make the Message Processor unreachable.
  - After update, make the Message Processor reachable.

Using a silent configuration file

You must pass a silent configuration file to the update command. The silent configuration file should be the same one you used to instal Edge 4.16.01.

Procedure for updating to 4.16.05 on a node with an external internet connection

Use the following procedure to update the Edge components on a node:

1. If present, disable any CRON jobs configured to perform a repair operation on Cassandra until after the update completes.
2. Log in to your node as root to install the Edge RPMs

   Note: While RPM installation requires root access, you can perform Edge configuration without root access.
3. Disable SELinux as described in Prerequisite: Disable SELinux.
4. Download the Edge 4.16.05 `bootstrap_4.16.05.sh` file to `/tmp/bootstrap_4.16.05.sh`:

   ```
   > curl https://software.apigee.com/bootstrap_4.16.05.sh -o
   /tmp/bootstrap_4.16.05.sh
   ```
5. Install the Edge 4.16.05 `apigee-service` utility and dependencies:

   ```
   > sudo bash /tmp/bootstrap_4.16.05.sh apigeeuser=uName
   apigeepassword=pWord
   ```

   where **`uName:pWord`** are the username and password you received from Apigee. If you omit **`pWord`**, you will be prompted to enter it.

   By default, the installer checks to see that you have Java 1.8 installed. If you do not, it installs it for you.

Use the `JAVA_FIX` option to specify how to handle Java installation. `JAVA_FIX` takes the following values:

I = Install OpenJDK 1.8 (default)
C = Continue without installing Java
Q = Quit. For this option, you have to install Java yourself.

6. Use `apigee-service` to update the `apigee-setup` utility:

   ```
   > /opt/apigee/apigee-service/bin/apigee-service apigee-setup update
   ```

   This update to `apigee-service` installs the `update.sh` utility in `<inst_dir>/apigee/apigee-setup/bin`.

7. Install the `apigee-validate` utility on the Management Server:

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

   Note: If you have installed the `apigee-validate` utility on a Message Processor node, you can update it by using the following command on that node:

   ```
   > /opt/apigee/apigee-service/bin/apigee-service apigee-validate update
   ```

   However, for 4.16.05, Apigee recommends that you install and run the `apigee-validate` utility on the Management Server.

8. Edit the config file passed to the `apigee-validate` utility.

   In the previous Edge release, the config file used by `apigee-validate` required the following properties:

   ```
   APIGEE_ADMINPW=sysAdminPword
   MP_POD=gateway
   REGION=dc-1
   ```

   In this release, the config file only requires the `APIGEE_ADMINPW` property.

9. Run the update utility on your nodes in the order described below in Order of machine update:

   ```
   > /opt/apigee/apigee-setup/bin/update.sh -c component -f configFile
   ```

   Use the “-c” option to specify the component to update. The list of possible components includes:

   `ldap` = OpenLDAP

`cs` = Cassandra
`zk` = Zookeeper
`qpid` = qpidd
`ps` = postgresql
`edge` =All Edge components except Edge UI: Management Server, Message Processor, Router, QPID Server, Postgres Server
`ui` = Edge UI
`all` = update all components on machine (only use for an Edge `aio` installation profile or an API BaaS `asa` installation profile)
`e` = ElasticSearch
`b` = API BaaS Stack
`p` = API BaaS Portal
`ebp` = ElasticSearch, API BaaS Stack, and API BaaS Portal on the same node

The only requirement on the config file is that the configuration file must be accessible or readable by the "apigee" user. For example, put the file in the `/tmp` directory on the node.

10. Test the update by running the `apigee-validate` utility on the Management Server, as described in Test the install.

To later rollback the update, use the procedure described in Rollback Process.

Procedure for updating to 4.16.05 from a local repo

If your Edge nodes are behind a firewall, or in some other way are prohibited from accessing the Apigee repository over the Internet, then you can perform the update from a local repository, or *mirror*, of the Apigee repo.

After you create a local Edge repository, you have two options for updating Edge from the local repo:

- Create a .tar file of the repo, copy the .tar file to a node, and then update Edge from the .tar file.
- Install a webserver on the node with the local repo so that other nodes can access it. Apigee provides the Nginx webserver for you to use, or you can use your own webserver.

To update from a local 4.16.05 repo:

1. Create a local 4.16.05 repo as described in Create a local Apigee repository.

   Note: If you already have an existing 4.16.01 repo, you can add the 4.16.05 repo to it.

2. **To install apigee-service from a .tar file:**

   a. On the node with the local repo, use the following command to package the local repo into a single .tar file named `/opt/apigee/data/apigee-mirror/apigee-4.16.05.tar.gz`:

   ```
   > /opt/apigee/apigee-service/bin/apigee-service apigee-mirror package
   ```

b. Copy the .tar file to the node where you want to update Edge. For example, copy it to the `/tmp` directory on the new node.

c. On the new node, untar the file to the `/tmp` directory:

```
> tar -xzf apigee-4.16.05.tar.gz
```

This command creates a new directory, named `repos`, in the directory containing the .tar file. For example `/tmp/repos`.

d. Install the Edge `apigee-service` utility and dependencies from `/tmp/repos`:

```
> sudo bash /tmp/repos/bootstrap_4.16.05.sh apigeeprotocol="file://"
apigeerepobasepath=/tmp/repos
```

Notice that you include the path to the `repos` directory in this command.

3. **To install apigee-service using the Nginx webserver:**

a. Configure the Nginx web server as described in Install from the repo using the Nginx webserver:

b. On the remote node, download the Edge bootstrap_4.16.05.sh file to `/tmp/bootstrap_4.16.05.sh`:

```
> /usr/bin/curl
http://uName:pWord@remoteRepo:3939/bootstrap_4.16.05.sh -o
/tmp/bootstrap_4.16.05.sh
```

where **`uName:pWord`** are the username and password you set above for the repo, and ***remoteRepo*** is the IP address or DNS name of the repo node.

c. On the remote node, install the Edge `apigee-service` utility and dependencies:

```
> sudo bash /tmp/bootstrap_4.16.05.sh apigeerepohost=remoteRepo:3939
apigeeuser=uName apigeepassword=pWord apigeeprotocol=http://
```

where **`uName:pWord`** are the repo username and password.

4. Use `apigee-service` to update the `apigee-setup` utility:

```
> /opt/apigee/apigee-service/bin/apigee-service apigee-setup update
```

This update to `apigee-service` installs the `update.sh` utility in `<inst_dir>/apigee/apigee-setup/bin`.

5. Install the `apigee-validate` utility on the Management Server:

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

Note: If you have installed the `apigee-validate` utility on a Message Processor node, you can update it by using the following command on that node:

```
> /opt/apigee/apigee-service/bin/apigee-service apigee-validate update
```

However, for 4.16.05, Apigee recommends that you install and run the `apigee-validate` utility on the Management Server.

6. Edit the config file passed to the `apigee-validate` utility.

   In the previous Edge release, the config file used by `apigee-validate` required the following properties:

   ```
   APIGEE_ADMINPW=sysAdminPword
   MP_POD=gateway
   REGION=dc-1
   ```

   In this release, the config file only requires the `APIGEE_ADMINPW` property.

7. Run the update utility on your nodes in the order described below in Order of machine update:

   ```
   > /opt/apigee/apigee-setup/bin/update.sh -c component -f configFile
   ```

   Use the “-c” option to specify the component to update. The list of possible components includes:

   `ldap` = OpenLDAP
   `cs` = Cassandra
   `zk` = Zookeeper
   `qpid` = qpidd
   `ps` = postgresql
   `edge` =All Edge components except Edge UI: Management Server, Message Processor, Router, QPID Server, Postgres Server
   `ui` = Edge UI
   `all` = update all components on machine (only use for an Edge `aio` installation profile or an API BaaS `asa` installation profile)
   `e` = ElasticSearch
   `b` = API BaaS Stack
   `p` = API BaaS Portal
   `ebp` = ElasticSearch, API BaaS Stack, and API BaaS Portal on the same node

   The only requirement on the config file is that the configuration file must be accessible or readable by the "apigee" user. For example, put the file in the `/tmp` directory on the node.

8. Test the update by running the `apigee-validate` utility on the Management Server, as described in Test the install.

To later rollback the update, use the procedure described in Rollback Process.

Order of machine update

The order that you update the machines in an Edge installation is important. The most important considerations to an update are:

- You must update **all** Cassandra and ZooKeeper nodes before you update any other nodes.
- You must update **all** qpidd and postgresql nodes before you update any Router and Message Processor nodes.
- For any machine with multiple Edge components (Management Server, Message Processor, Router, QPID Server, Postgres Server), use the "-c edge" option to update them all at the same time.
- If a step specifies that it should be performed on multiple machines, perform it in the specified machine order.
- There is no separate step to update Monetization. It is updated when you specify the "-c edge" option.
- After you update a Router node, you must remove all files from the `/opt/nginx/conf.d` directory, and then restart the Router.

For a 1-host standalone installation

1. Update machine 1:

   ```
   > /opt/apigee/apigee-setup/bin/update.sh -c all -f configFile
   ```

2. Delete any files in `/opt/nginx/conf.d`:

   ```
   > rm -f /opt/nginx/conf.d/*
   ```

3. Restart the Edge Router:

   ```
   > /<inst_root>/apigee/apigee-service/bin/apigee-service edge-router
   restart
   ```

For a 2-host standalone installation

1. Update Cassandra and ZooKeeper on machine 1:

   ```
   > /opt/apigee/apigee-setup/bin/update.sh -c cs,zk -f configFile
   ```

2. Update qpidd and postgresql on machine 2:

   ```
   > /opt/apigee/apigee-setup/bin/update.sh -c qpid,ps -f configFile
   ```

3. Update LDAP on machine 1:

```
> /opt/apigee/apigee-setup/bin/update.sh -c ldap -f configFile
```

4. Update Edge components on machine 2 and machine 1:

```
> /opt/apigee/apigee-setup/bin/update.sh -c edge -f configFile
```

5. On node 1:
    a. Delete any files in `/opt/nginx/conf.d`:

    ```
    > rm -f /opt/nginx/conf.d/*
    ```

    b. Restart the Edge Router:

    ```
    > /<inst_root>/apigee/apigee-service/bin/apigee-service edge-router
    restart
    ```

6. Update UI on machine 1:

```
> /opt/apigee/apigee-setup/bin/update.sh -c ui -f configFile
```

For a 5-host clustered installation

1. Update Cassandra and ZooKeeper on machine 1, 2, and 3:

```
> /opt/apigee/apigee-setup/bin/update.sh -c cs,zk -f configFile
```

2. Update qpidd and postgresql on machine 4 and 5:

```
> /opt/apigee/apigee-setup/bin/update.sh -c qpid,ps -f configFile
```

3. Update LDAP on machine 1:

```
> /opt/apigee/apigee-setup/bin/update.sh -c ldap -f configFile
```

4. Update Edge components on machine 4, 5, 1, 2, 3:

```
> /opt/apigee/apigee-setup/bin/update.sh -c edge -f configFile
```

5. On nodes 2 and 3:
    a. Delete any files in `/opt/nginx/conf.d`:

    ```
    > rm -f /opt/nginx/conf.d/*
    ```

    b. Restart the Edge Router:

```
> /<inst_root>/apigee/apigee-service/bin/apigee-service edge-router
restart
```

6. Update UI on machine 1:

```
> /opt/apigee/apigee-setup/bin/update.sh -c ui -f configFile
```

For a 9-host clustered installation

1. Update Cassandra and ZooKeeper on machine 1, 2, and 3:

```
> /opt/apigee/apigee-setup/bin/update.sh -c cs,zk -f configFile
```

2. Update qpidd on machine 6 and 7:

```
> /opt/apigee/apigee-setup/bin/update.sh -c qpid -f configFile
```

3. Update postgresql on machine 8 and 9:

```
> /opt/apigee/apigee-setup/bin/update.sh -c ps -f configFile
```

4. Update LDAP on machine 1:

```
> /opt/apigee/apigee-setup/bin/update.sh -c ldap -f configFile
```

5. Update Edge components on machine 6, 7, 8, 9, 1, 4, and 5 in that order:

```
> /opt/apigee/apigee-setup/bin/update.sh -c edge -f configFile
```

6. On nodes 4 and 5:
    a. Delete any files in `/opt/nginx/conf.d`:

```
> rm -f /opt/nginx/conf.d/*
```

    b. Restart the Edge Router:

```
> /<inst_root>/apigee/apigee-service/bin/apigee-service edge-router
restart
```

7. Update UI on machine 1:

```
> /opt/apigee/apigee-setup/bin/update.sh -c ui -f configFile
```

For a 13-host clustered installation

1. Update Cassandra and ZooKeeper on machine 1, 2, and 3:

```
> /opt/apigee/apigee-setup/bin/update.sh -c cs,zk -f configFile
```

2. Update qpidd on machine 12 and 13:

```
> /opt/apigee/apigee-setup/bin/update.sh -c qpid -f configFile
```

3. Update postgresql on machine 8 and 9:

```
> /opt/apigee/apigee-setup/bin/update.sh -c ps -f configFile
```

4. Update LDAP on machine 4 and 5:

```
> /opt/apigee/apigee-setup/bin/update.sh -c ldap -f configFile
```

5. Update Edge components on machine 12, 13, 8, 9, 6, 7, 10, and 11 in that order:

```
> /opt/apigee/apigee-setup/bin/update.sh -c edge -f configFile
```

6. On nodes 10 and 11:
    a. Delete any files in `/opt/nginx/conf.d`:

    ```
    > rm -f /opt/nginx/conf.d/*
    ```

    b. Restart the Edge Router:

    ```
    > /<inst_root>/apigee/apigee-service/bin/apigee-service edge-router
    restart
    ```

7. Update UI on machine 6 and 7:

```
> /opt/apigee/apigee-setup/bin/update.sh -c ui -f configFile
```

For a 12-host clustered installation

1. Update Cassandra and ZooKeeper:
    a. On machines 1, 2 and 3 in Data Center 1:

    ```
    > /opt/apigee/apigee-setup/bin/update.sh -c cs,zk -f configFile
    ```

    b. On machines 7, 8, and 9 in Data Center 2

    ```
    > /opt/apigee/apigee-setup/bin/update.sh -c cs,zk -f configFile
    ```

2. Update qpidd:
    a. Machines 4, 5 in Data Center 1

    ```
    > /opt/apigee/apigee-setup/bin/update.sh -c qpid -f configFile
    ```

b. Machines 10, 11 in Data Center 2

```
> /opt/apigee/apigee-setup/bin/update.sh -c qpid -f configFile
```

3. Update postgresql:

a. Machines 6 in Data Center 1

```
> /opt/apigee/apigee-setup/bin/update.sh -c ps -f configFile
```

b. Machines 12 in Data Center 2

```
> /opt/apigee/apigee-setup/bin/update.sh -c ps -f configFile
```

4. Update LDAP:

a. Machines 1 in Data Center 1

```
> /opt/apigee/apigee-setup/bin/update.sh -c ldap -f configFile
```

b. Machines 7 in Data Center 2

```
> /opt/apigee/apigee-setup/bin/update.sh -c ldap -f configFile
```

5. Update Edge components:

a. Machines 4, 5, 6, 1, 2, 3 in Data Center 1

```
> /opt/apigee/apigee-setup/bin/update.sh -c edge -f configFile
```

b. Machines 10, 11, 12, 7, 8, 9 in Data Center 2

```
> /opt/apigee/apigee-setup/bin/update.sh -c edge -f configFile
```

c. On nodes 2, 3, 8 and 9:

i. Delete any files in `/opt/nginx/conf.d`:

```
> rm -f /opt/nginx/conf.d/*
```

ii. Restart the Edge Router:

```
> /<inst_root>/apigee/apigee-service/bin/apigee-service
edge-router restart
```

6. Update UI:

a. Machine 1 in Data Center 1

```
> /opt/apigee/apigee-setup/bin/update.sh -c ui -f configFile
```

b. Machine 7 in Data Center 2

```
> /opt/apigee/apigee-setup/bin/update.sh -c ui -f configFile
```

For a 7-host API BaaS installation

1. Update Cassandra on machine 5, 6, and 7:

   ```
   > /opt/apigee/apigee-setup/bin/update.sh -c cs -f configFile
   ```
2. Update ElasticSearch and API BaaS Stack on machine 1, 2, and 3:

   ```
   > /opt/apigee/apigee-setup/bin/update.sh -c e,b -f configFile
   ```
3. Update API BaaS Portal on machine 4:

   ```
   > /opt/apigee/apigee-setup/bin/update.sh -c p -f configFile
   ```

For a 10-host API BaaS installation

1. Update Cassandra on machine 8, 9, and 10:

   ```
   > /opt/apigee/apigee-setup/bin/update.sh -c cs -f configFile
   ```
2. Update ElasticSearch on machine 1, 2, and 3:

   ```
   > /opt/apigee/apigee-setup/bin/update.sh -c e -f configFile
   ```
3. Update API BaaS Stack on machine 4, 5, and 6:

   ```
   > /opt/apigee/apigee-setup/bin/update.sh -c b -f configFile
   ```
4. Update API BaaS Portal on machine 7:

   ```
   > /opt/apigee/apigee-setup/bin/update.sh -c p -f configFile
   ```

For a non-standard installation

If you have a non-standard installation, then update Edge components in the following order:

1. ZooKeeper
2. Cassandra
3. qpidd
4. postgresql
5. LDAP

6. Edge, meaning the "-c edge" profile on all nodes in the order: Qpid, Postgres, Management Server, Message Processor, Router.
7. On all Router nodes
    a. Delete any files in `/opt/nginx/conf.d`:

    ```
    > rm -f /opt/nginx/conf.d/*
    ```

    b. Restart the Edge Router:

    ```
    > /<inst_root>/apigee/apigee-service/bin/apigee-service edge-router
    restart
    ```

8. UI

Rollback Process

There are two scenarios where you might want to perform a rollback:

1. Rollback to an older release. For example from 4.16.05 to 4.16.01.
2. Rollback to an older version in the same release.

Use the procedure below to perform a rollback for both scenarios.

Who can perform the rollback

The user performing the rollback should be the same as the user who originally updated Edge, or a user running as root.

By default, Edge components run as the user "apigee". In some cases, you might be running Edge components as different users. For example, if the Router has to access privileged ports, such as those below 1000, then you have to run the Router as root or as a user with access to those ports. Or, you might run one component as one user, and another component as another user.

Which components can be rolled back

You should be aware of the following conditions when performing a rollback:

- To rollback any one of the following five components on a node, you must roll back any of the five installed on the node. For example, if you have the Management Server, Route, and Message Processor installed on the node, to roll back any one of them you must roll back all three.

  The five components are:
    - Management Server
    - Router
    - Message Processor
    - Qpid Server
    - Postgres Server

- Do not rollback Cassandra. This release of Edge contains an updated version of Cassandra. If you rollback any components, leave Cassandra at the 4.16.05 version.
- This release does not contain a new version of postgresql or qpidd. Therefore, you do not have to roll them back.

To rollback 4.16.05

To rollback Apigee Edge, perform the following rollback steps:

Note: If you are trying to rollback on a production system, contact Apigee Support for more information.

1. Stop the component to rollback:
    a. **If you are rolling back any one of the following components, you must stop them all: Management Server, Router, Message Processor, Qpid Server, or Postgres Server**:

    ```
    > apigee-service edge-management-server stop
    > apigee-service edge-router stop
    > apigee-service edge-message-processor stop
    > apigee-service edge-qpid-server stop
    > apigee-service edge-postgres-server stop
    ```

    b. **If you are rolling back any other component, stop just that component:**

    ```
    > apigee-service comp stop
    ```

2. If you are rolling back Monetization, uninstall it:

    ```
    > apigee-service edge-mint-gateway uninstall
    ```

3. Uninstall the component to rollback:
    a. If you are rolling back any of the following components, then uninstall them all: Management Server, Router, Message Processor, Qpid Server, or Postgres Server:

    ```
    > apigee-service edge-gateway uninstall
    ```

    b. **If you are rolling back any other component, uninstall just that component**

    ```
    > apigee-service comp uninstall
    ```

4. If you are rolling back the Router, then you have to delete the contents of `/opt/nginx/conf.d`:

    ```
    > cd  /opt/nginx/conf.d
    > rm -rf *
    ```

5. **To rollback the component to the 4.16.01 release:**
    a. Uninstall the 4.16.05 version of `apigee-setup`:

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

b. Download bootstrap.sh for the 4.16.01 release:

```
> curl https://software.apigee.com/bootstrap.sh -o /tmp/bootstrap.sh
-u uName:pWord
```

where ***uName:pWord*** are the username and password you received from Apigee. If you omit ***pWord***, you will be prompted to enter it.

c. Install the 4.16.01 Edge `apigee-service` utility and dependencies:

```
> sudo bash /tmp/bootstrap.sh apigeeuser=uName apigeepassword=pWord
```

where ***uName*** and ***pWord*** are the username and password you received from Apigee. If you omit ***pWord***, you will be prompted to enter it.

d. Install the 4.16.01 version of `apigee-setup`:

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

e. Install the 4.16.01 version of the component:

```
> /<instal_dir>/apigee/apigee-setup/bin/setup.sh -p comp -f
configFile
```

where ***comp*** is the component to install and ***configFile*** is your 4.16.01 configuration file.

6. **To rollback the component to a specific version of the 4.16.05 release:**

a. Download the specific component version:

```
> /<instal_dir>/apigee/apigee-service/bin/apigee-service comp-version
install
```

where ***comp-version*** is the component and version to install. For example:

```
> /<instal_dir>/apigee/apigee-service/bin/apigee-service
edge-ui-4.16.05-0.0.3649 install
```

If you are using the Apigee online repo, you can determine the available component versions by using the following command:

```
> yum --showduplicates list comp
```

For example:

```
> yum --showduplicates list edge-ui
```

b. Use apigee-setup to install the component:

```
> /<install_dir>/apigee/apigee-setup/bin/setup.sh -p comp -f
configFile
```

For example:

```
> /<install_dir>/apigee/apigee-setup/bin/setup.sh -p ui -f configFile
```

Note how you only specify the component name when you do the install.

10 Almaden Boulevard, 16th Floor, San Jose
CA 95113
USA

No. 17/2, 2B Cross, 7th Main,
2 & 3 Floor, Off 80 Feet Road, 3rd Block
Koramangala, Bangalore 560034
INDIA

3 Sheldon Square
London W2 6HY
UK

www.apigee.com