Apigee^{TM}

Apigee Edge for Private Cloud

v4.15.07.00

September 8, 2015

Install and Configuration Guide

THE CONTENTS OF THIS PUBLICATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT OF INTELLECTUAL PROPERTY.

APIGEE CORPORATION SHALL NOT UNDER ANY CIRCUMSTANCES BE LIABLE TO ANY PERSON FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES, INCLUDING WITHOUT LIMITATION, DAMAGES RESULTING FROM THE USE OF OR RELIANCE ON THE INFORMATION IN THIS PUBLICATION, LOSS OF PROFITS, REVENUE OR DATA, EVEN IF APIGEE CORPORATION HAS BEEN PREVIOUSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

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

One Kingdom Street, 4th Floor
Paddington Central
London W2 6BD
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 4.15.07.00**. Any references that are specific to previous versions are oversights and should be reported as bugs.

Note: Apigee Edge releases use the following version numbering scheme: V.YY.MM.##, where:

- V specifies the major version number.
- YY and MM specify the year and month of the release.
- ## is the service pack number. Initial releases are V.YY.MM.00.

Release numbers are embedded in the distribution file names. Please be sure that you are using the version of the *Installation Steps* that corresponds to the distribution file version.

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 Edge API Backend (aka App Services)
- 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.

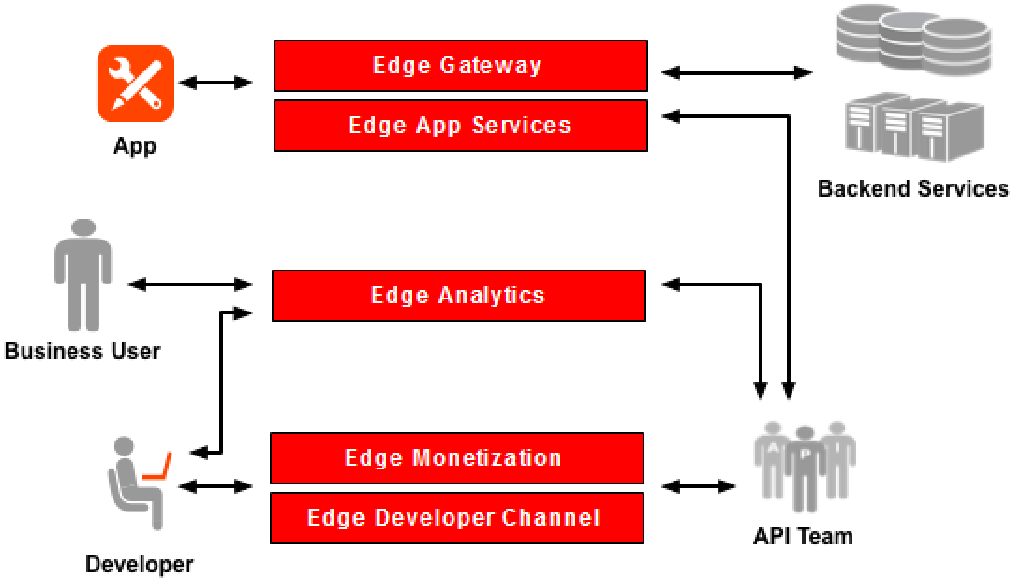


Figure 1: Apigee Edge for Private Cloud Architecture

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

Apigee API Backend

API Backend is a complete backend as a service (BaaS) for powering mobile and Web apps that you install as an addition to Edge. API Backend 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 Backend 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 Backend Features

The Apigee documentation site has extensive information on API Backend 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 Backend components interact.

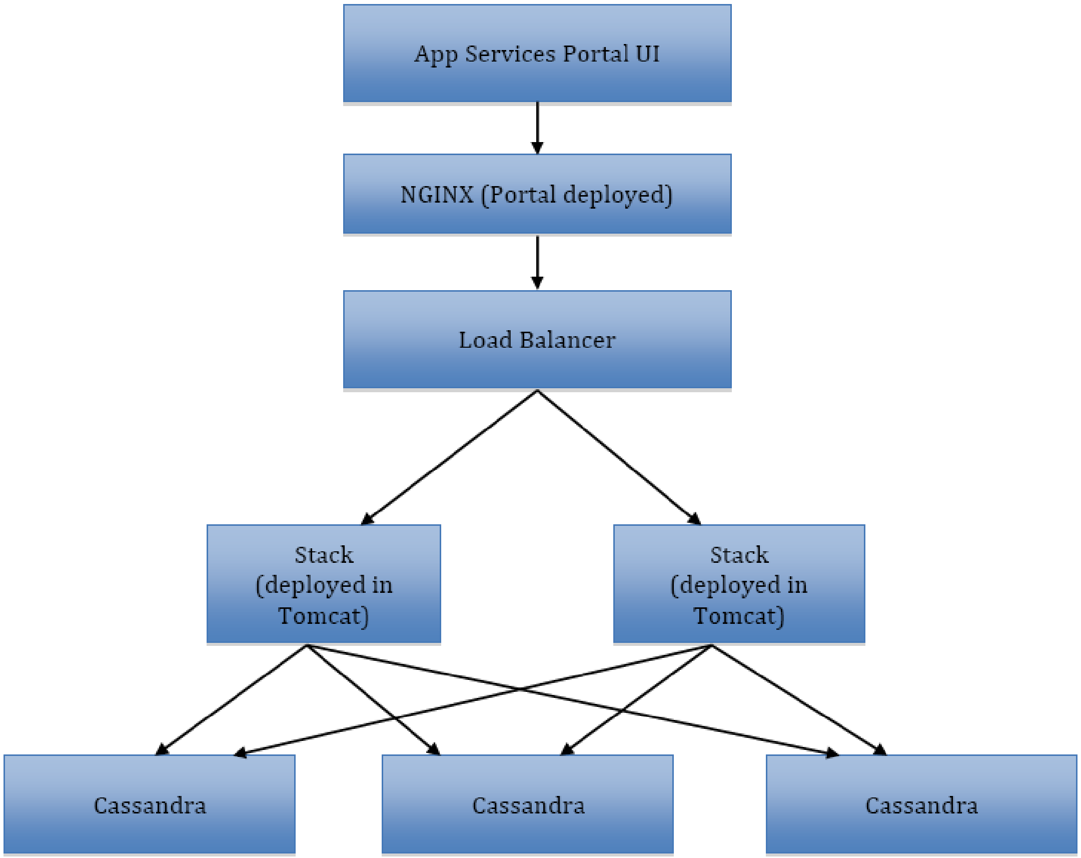


Figure 2: API Backend Overview and Architecture

Software Components

API Backend is built from the following primary components:

- Application Services Stack - deployed in the Tomcat webserver
- Application Services Portal (API Backend UI) - deployed in the NGINX server
- Load Balancer (NGINX) - installed with the Portal to balance requests across the Application Services Stacks.

You can scale the API Backend 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 Backend using Edge UI, 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

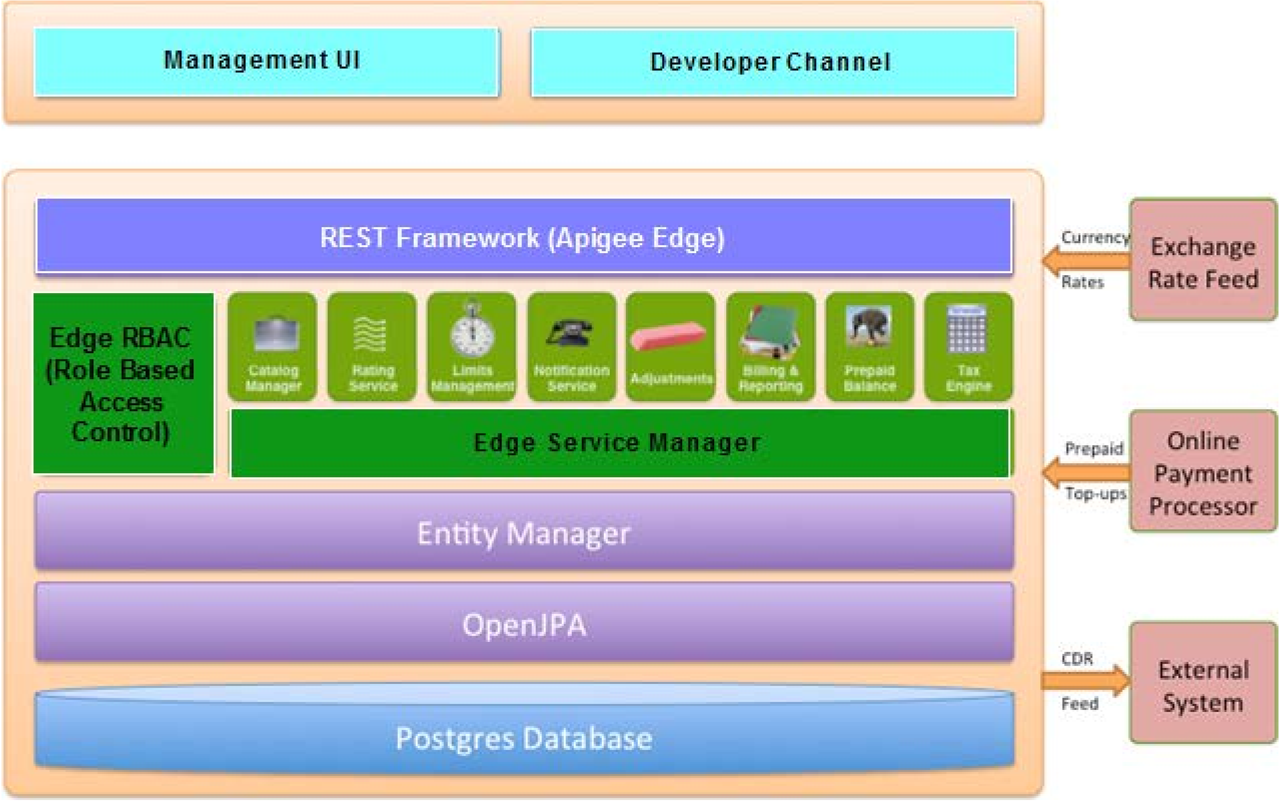


Figure 3: Edge Monetization Services Overview

Software Components

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

- Edge Management Server
- Edge Message Processor
- Edge Qpid Server

- Edge Postgres Server

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 3-host Installation with 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.

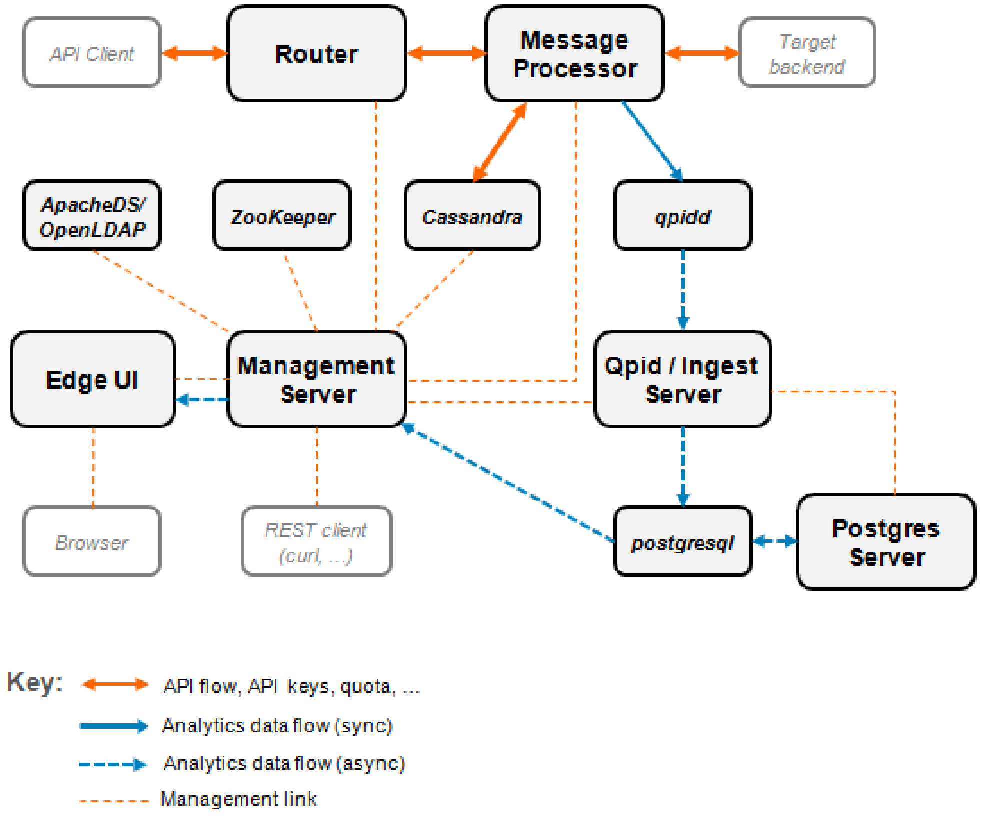
The services provided by the on-premises installation of Edge Gateway include (but are not limited to):

- A **Router** handles all incoming API traffic, determines the API proxy that handles the request, balances requests across available Message Processors, and 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 node. (Routers are Apigee software.)
- A **Message Processor** proxies API traffic for a specific organization and environment and executes all policies.
- An **Apache Cassandra** stores application configurations, distributed quota counters, API keys, and OAuth tokens for applications running on the gateway.
- An **Apache ZooKeeper** contains configuration data about all the services of the zone and which 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. This offers an API that is used by the Central Services server to communicate with the servers in each on-premises installation.
- 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 services 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.



Apigee Edge components

- **Management Server**
  Central API management component
- **Edge UI**
  Management Console
- **Router**
  Handles all incoming API traffic and dispatches it
- **Message Processor**
  Proxies API traffic and executes all policies
- **Qpid / Ingest Server**
  Manages Qpid Daemon (qpidd) and Analytics data flow
- **Postgres Server**
  Manages postgresql and aggregates analytics data

3rd party components

- **ZooKeeper**
  Manages overall configuration (optionally as cluster)
- **Cassandra**
  Stores deployments and runtime information, e.g. quota, API keys, ... (optionally as cluster)
- **ApacheDS / OpenLDAP**
  Manages system and organization users and roles
  OpenLDAP also allows master-master replication
- **Qpid Daemon (qpidd)**
  Queuing system for analytics data
- **Postgres DB (postgresql)**
  Analytics database (master / standby replication supported)

Figure 4: Conceptual Component Interactions

Installation

This section covers initial installation and initial testing to verify the installation.

Overview

This version of 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.

Note: The first 5 install scenarios cover core Apigee Edge (Gateway and Analytics) installation. The next 2 scenarios describe Gateway and Analytics setup with API Backend installation. The last one covers Monetization Services installation that runs within any existing setup.

1) **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.

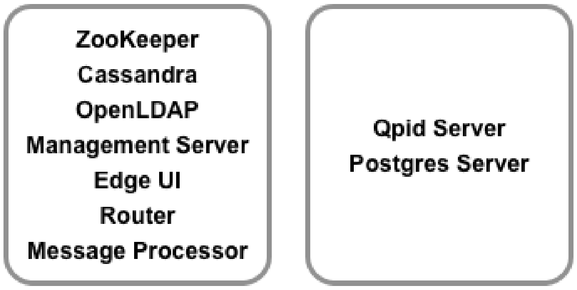


Figure 5: Standalone Setup

2) **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.

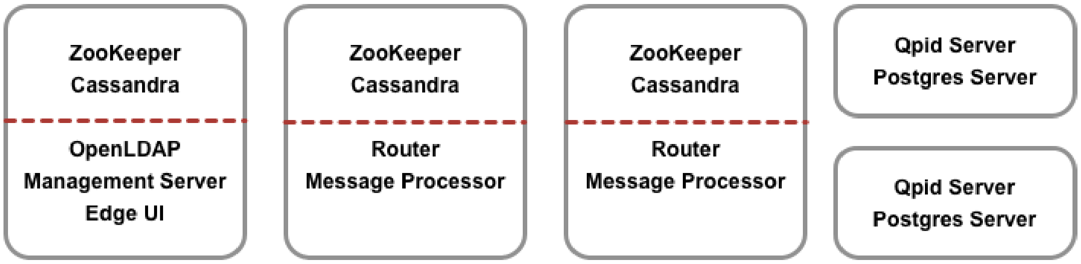


Figure 6: Five-host Clustered Setup

3) **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.

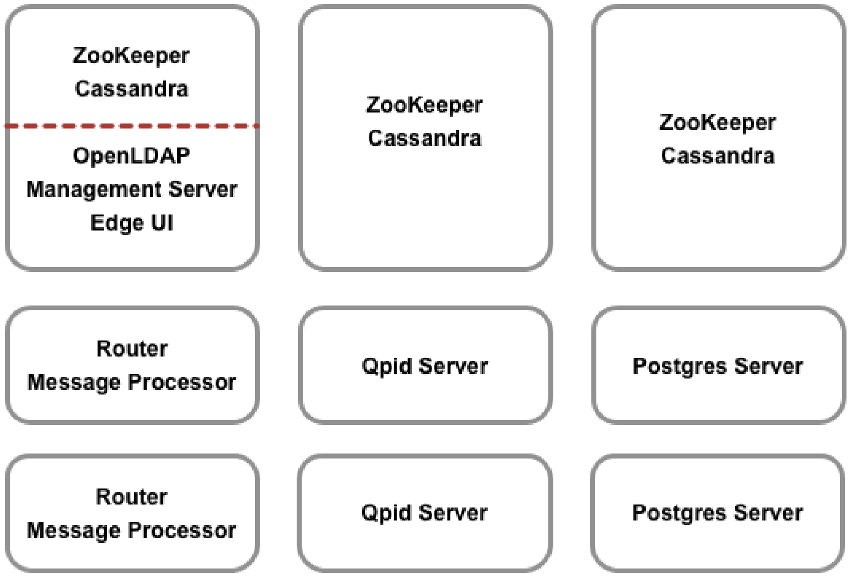


Figure 7: Nine-host Clustered Setup

4) **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.

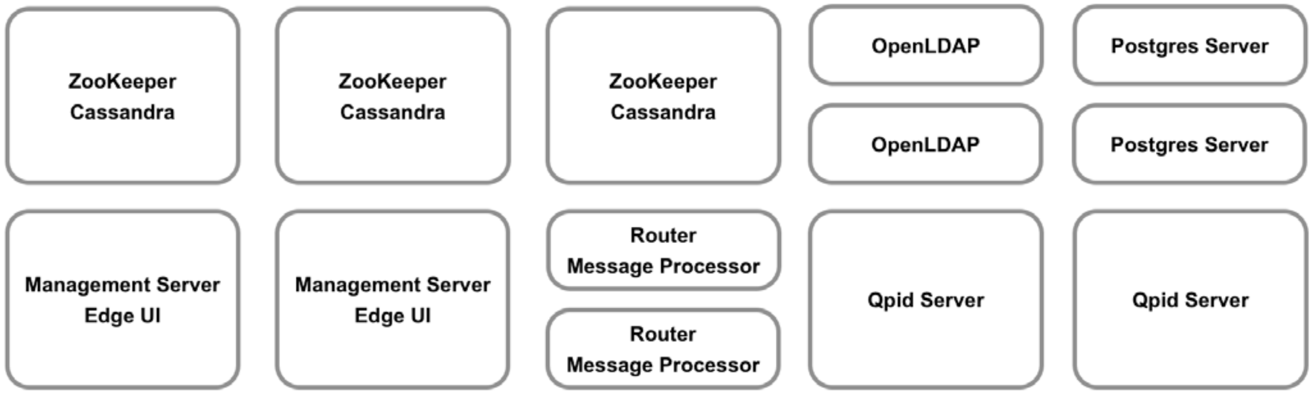


Figure 8: 13-host Clustered Setup

5) **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).

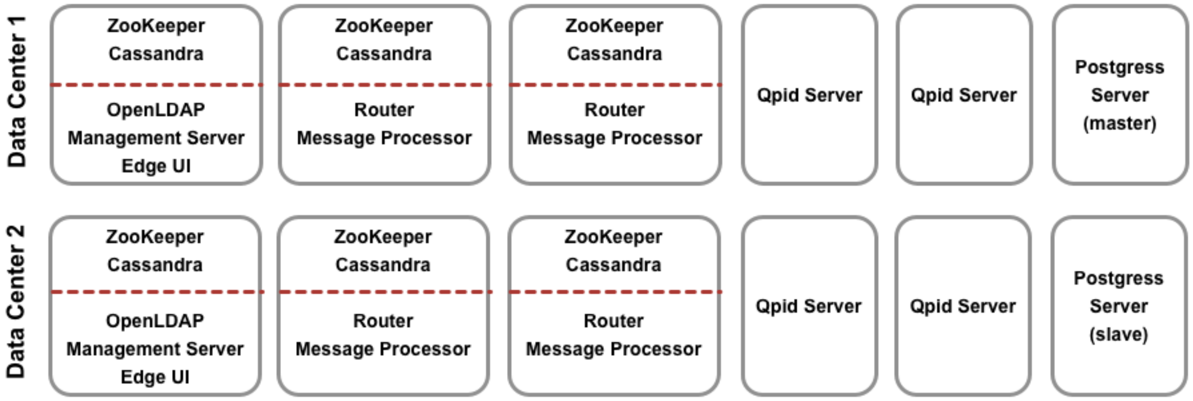


Figure 9: 12-host Clustered Setup

6) **8-host clustered installation (Min HA with API Backend):** In this scenario, five hosts run an existing five-host clustered setup. You then install API Backend Services on three hosts. Two hosts run API Backend Stack. One host runs API Backend Portal and load balancer.

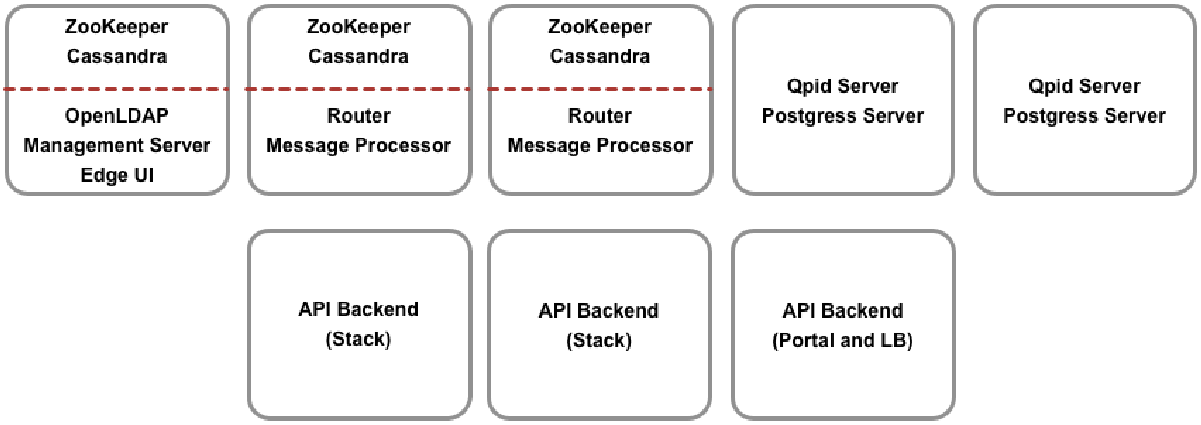


Figure 10: Eight-host Clustered Setup

7) **12-host clustered installation (Performance HA with API Backend):** In this scenario, nine hosts run an existing nine-host clustered setup. Two hosts run API Backend Stack. One host runs API Backend Portal and load balancer.

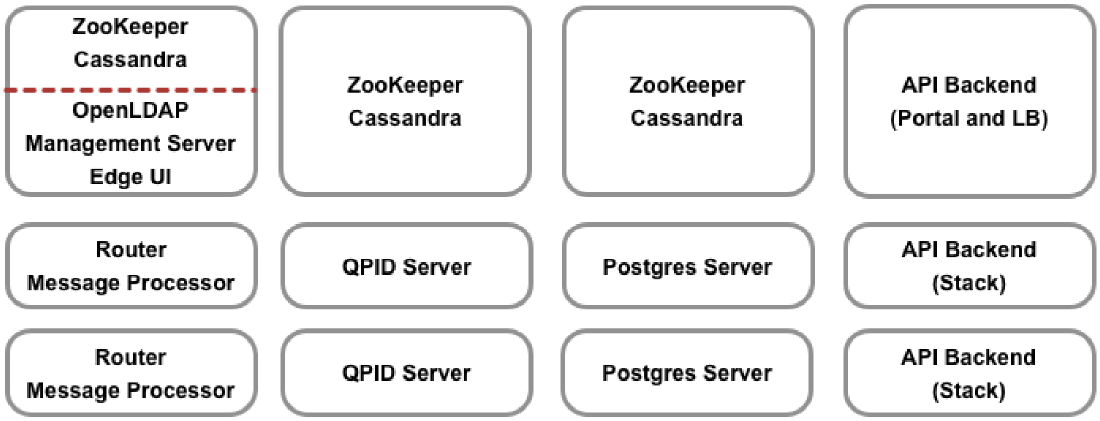


Figure 11: 12-host Clustered Setup

8) **3-host installation (MS-RMP-SAX with MO):** Monetization Services runs within any existing Apigee Edge setup. In this scenario, you install Monetization Services on three hosts. One host runs Apigee Management Server, Apache ZooKeeper, Apache Cassandra, OpenLDAP, and Edge UI. One host runs Apigee Router + Message Processor. One host runs Apigee Analytics. Apigee Monetization Services runs on all hosts.

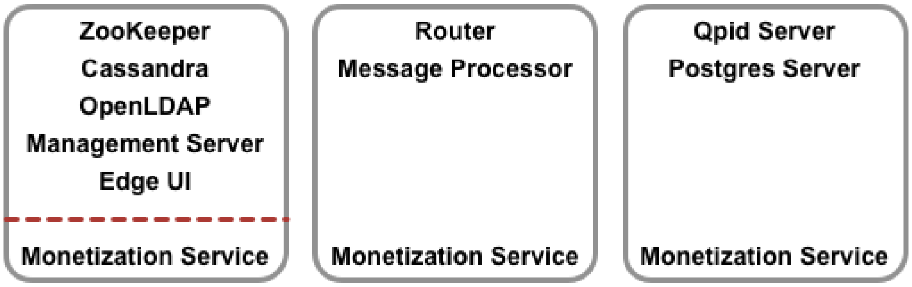


Figure 12: Three-host Setup (with Monetization Services)

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 Backend Services.

API Backend Services Component	RAM	CPU	Hard disk
API Backend Stack (shared Cassandra server with Edge)	8GB	4-core	60 - 80GB
API Backend Portal	1GB	2-core	20GB

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

These installation instructions and the supplied installation files have been tested on the operating systems 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.

Installation directory

By default, the installer writes all files to the `/opt/apigee4` directory. However, you can specify a different location to the installer.

In the instructions in this guide, the installation directory is noted as /`<inst-root>/apigee4`, where /`<inst-root>` is `/opt` by default.

Java

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

Warning: This release of Edge does not support JDK 6. If you are currently using JDK 6, you must upgrade to JDK 7.

Note: In order to avoid issues with java versions, `apigee-install.sh` prompts for java path, checks the version and then links it into `<inst-root>/apigee4/share/jdk`

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

Every script sources `<inst-root>/apigee4/bin/apigee-env.sh` which adds:

- `export JAVA_HOME=<inst-root>/share/jdk`
- `export PATH=$JAVA_HOME/bin:$PATH`

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 for using API Backend.

For all Cassandra nodes, make sure to set the initial and maximum Java heap size to 8GB. Set the heap size after installing Edge by editing the file /`<inst-root>/apigee4/conf/cassandra/cassandra-env.sh` on the Cassandra node and setting the following properties:

```
MAX_HEAP_SIZE="8G"
HEAP_NEWSIZE="8G"
```

As a rule of thumb, the heap size should not be more than 50% of the total system RAM.

After installing the Edge for Private Cloud, you can check that Cassandra is configured correctly by examining the `/<inst-root>/apigee4/conf/cassandra/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`

PostgreSQL database

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

```
shared_buffers = 35% of RAM      # min 128kB
effective_cache_size = 45% of RAM
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.conf`:

   ```
   > vi /<inst-root>/apigee4/data/postgresql/pgdata/postgresql.conf
   ```

2. Set the properties listed above.
3. Save your edits.
4. Restart the PostgreSQL database:

   ```
   > /<inst-root>/apigee4/bin/apigee-service postgresql restart
   ```

jsvc

“jsvc” is a prerequisite for using API Backend. Version 1.0.15-dev is installed in `/<install-root>/apigee4/share/apache-tomcat–7.x.y/bin` when you install the API Backend.

Tools

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

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

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.

rsync – The replication script, `apigee-postgres-replication-setup.sh` requires the 'rsync' utility on master and standby PostgreSQL servers prior to establishing master-standby PostgresSQL setup. You can use "`yum -y install rsync`" to install the utility.

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`.

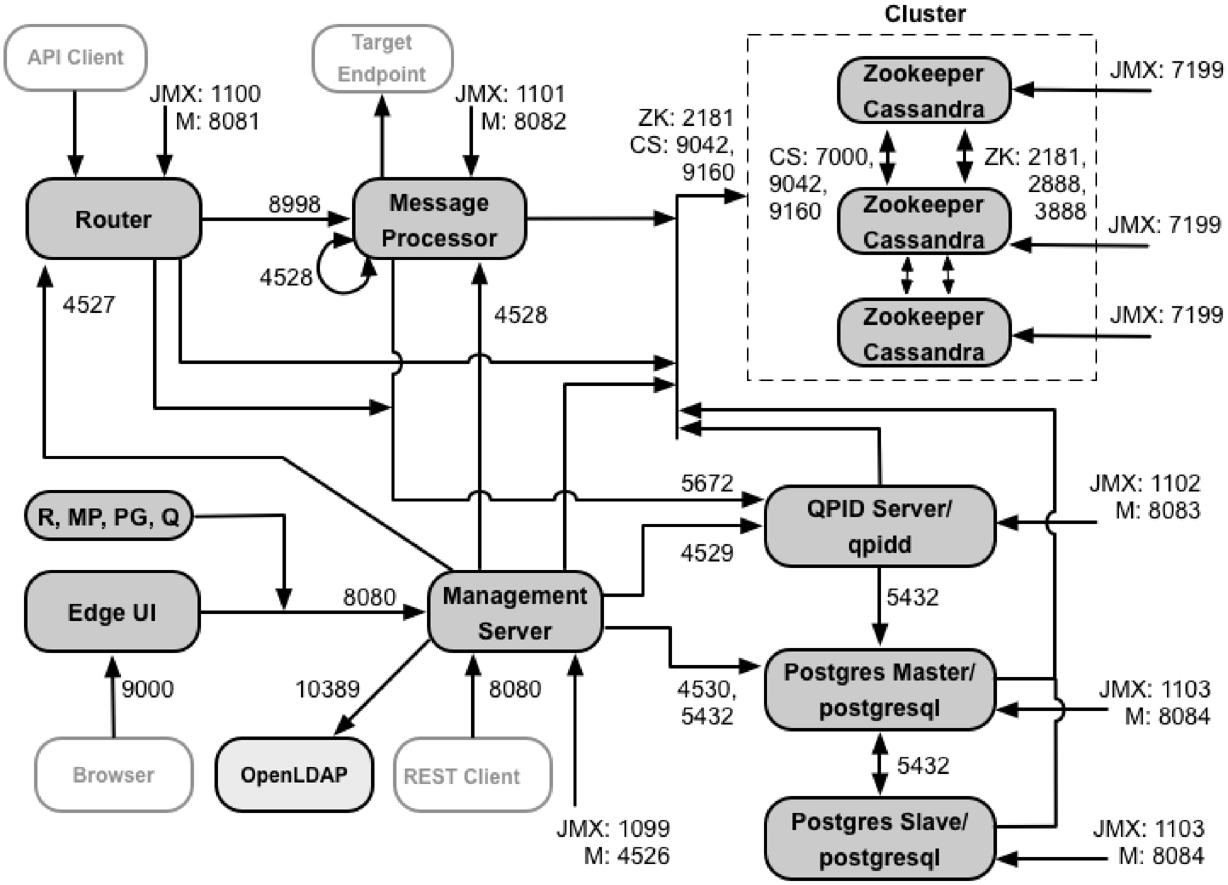
In this case, you need a load balancer or traffic director of some kind splitting the `http://api.mycompany.com:80/` traffic between the two IP addresses (`111.111.111.111` on VM1 and `111.111.111.222` on VM2). This function is specific to your particular installation, and is configured by your local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

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:

- 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).
- A Router must open port 4527 as its management port. If you have multiple Routers, they use port 4527 to communicate with each other.
- Access to JMX ports can be configured to require a username/password. See the Edge *Operations Guide,* available on the Apigee ftp site: ftp://ftp.apigee.com/, 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 the Edge *Operations Guide,* available on the Apigee ftp site: ftp://ftp.apigee.com/, 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
	1101	JMX port
	4528	For distributed cache and management calls
Router	8081	Default management port for Router
	1100	JMX port
	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

Component	Port	Description
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) Router (1100) Message Processor (1101) Qpid Server (1102) Postgres Server (1103)

Port Number	Purpose	Source Component	Destination Component
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	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 component opens a different port; the exact port used depends on the configuration	Management API clients	Router (8081) Message Processor (8082) Qpid Server (8083) Postgres Server (8084)
8998	Communication between router an message processor	Router	Message Processor

Port Number	Purpose	Source Component	Destination Component
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. In `/<inst-root>/apigee4/conf/apigee/message-processor/system.properties` for the message processor, uncomment following property:

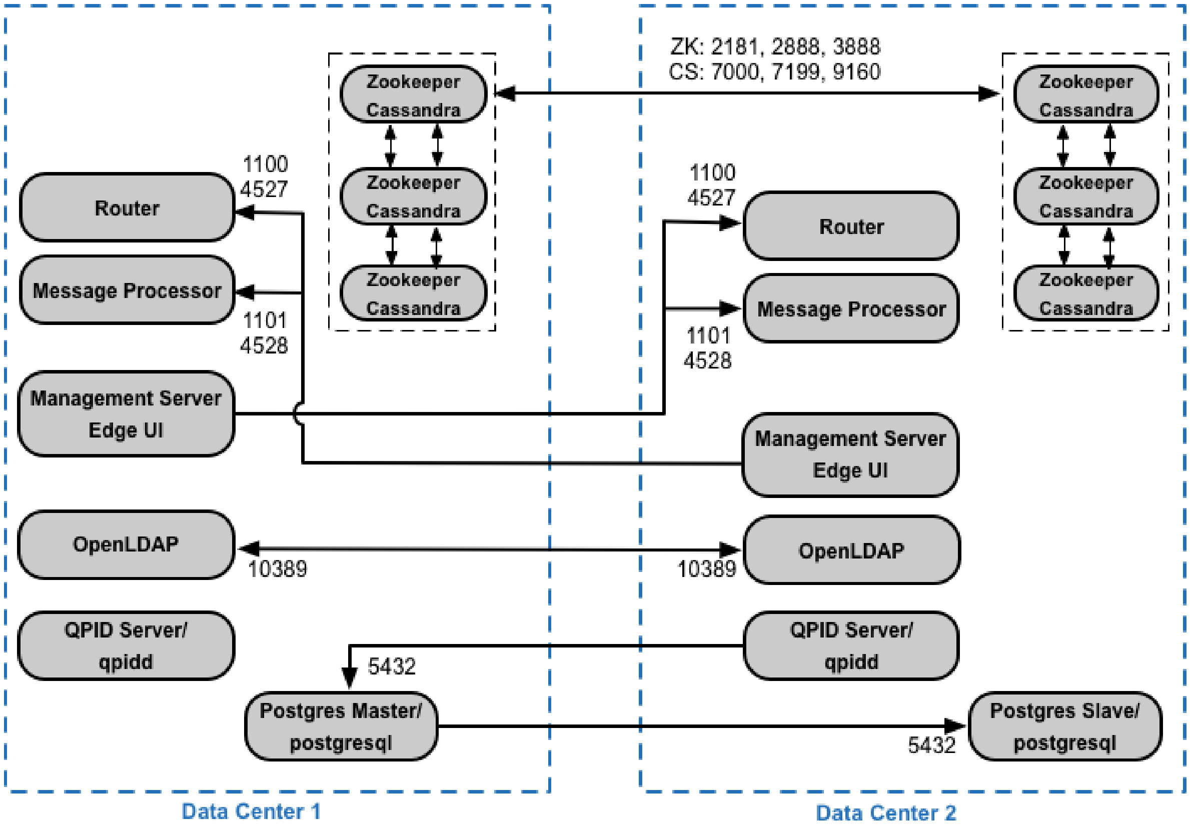
   ```
   #casssandra.maxConnectTimeInMillis = -1
   ```

3. On the routers, uncomment the following line in the `/<inst-root>/apigee4/conf/apigee/router/system.properties` file:

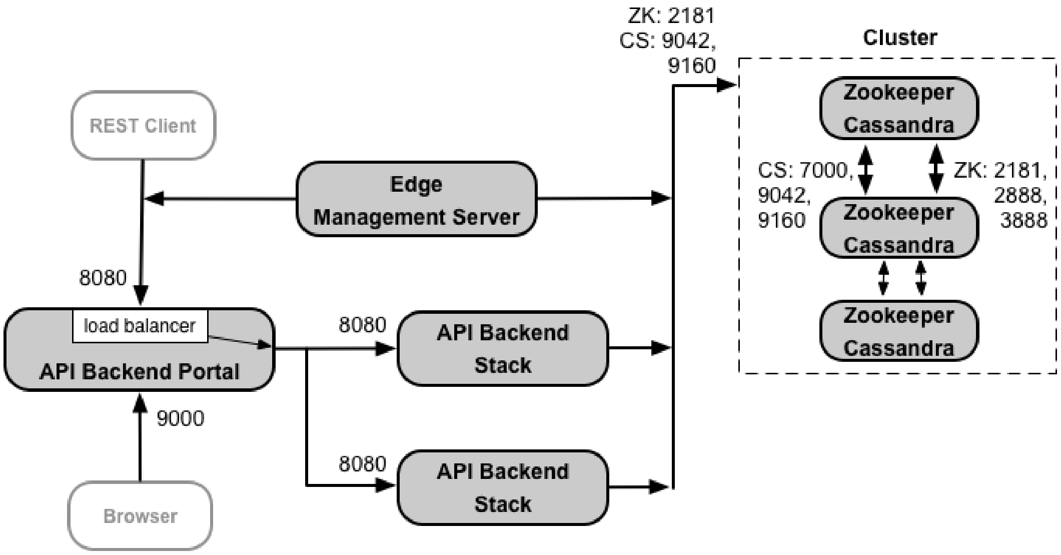
   ```
   #casssandra.maxConnectTimeInMillis = -1
   ```

4. Restart the routers and message processors.

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:



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



Note that most of the Edge connections have been omitted from this diagram for simplicity.

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

Component	Port	Description
API Backend Portal (with load balancer)	8080	Port for the load balancer
	9000	Port for the API backend UI
API Backend Stack	8080	Port where API request are received

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>/conf/apigee/management-server/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: `/var/log/apigee/management-server`. In this case you can contact Apigee Support for upgrade details.

You can upgrade the license by using the script, /`<inst-root>/apigee/bin/install-license.sh`. Note that the management server stops if the license file doesn't exist or is invalid.

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`".
- **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. To configure SELinux:

  1. Edit the `/etc/sysconfig/selinux` file:

     ```
     > sudo vi /etc/sysconfig/selinux
     ```

  2. Set `SELINUX=disabled` or `SELINUX=permissive`.
  3. Save your edits.
  4. Restart the machine.

- **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
  ```

- **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 ZIP file is available on the Apigee ftp server at `ftp.apigee.com`. Contact Apigee Support if you require credentials to access the ftp server.
- **System limits:**
  - On Cassandra nodes, set soft and hard nofile for user (default is “apigee") in `/etc/security/limits.conf` to 32768.
  - 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.

- **LDAP configuration file (non-root installation):** The Edge LDAP configuration uses OpenLDAP. OpenLDAP is also used in the dual management server setup to support LDAP master-master replication. For **non-root installation**, you can install the following OpenLDAP using:

  - `yum install openldap-clients openldap-servers`

  To verify OpenLDAP prerequisites installation, run the following script on Management Server:

  `/<inst-rt>/apigee4/share/installer/apigee-openldap-check-prerequisites.sh`

  Note: In root installation type, the installer (*`apigee-setup.sh`*) automatically detects and installs the prerequisites.

- **Analytics dependency files (non-root installation):** In root installation type, the configuration script, *`apigee-setup.sh`* installs all prerequisites for Analytics components (Qpid and Postgres) via global yum under the installation root. Ensure that the system is connected to the internet for installing these components.

  For **non-root installation**, you need to use the following scripts to install the prerequisites:

  Qpid prerequisite packages: The script, *`apigee-qpid-install-prerequisites.sh`* installs the following Qpid packages during analytics installation on Red Hat /CentOS:

  - db4-cxx, glibc, libaio, libstdc++, libuuid, xerces-c, xqilla, boost-filesystem, boost-program-options, boost-system, boost-test, nspr, nss, nss-util, cyrus-sasl-lib

  The script also sets soft and hard nofile for user (default is “apigee") in `/etc/security/limits.conf` to 5000.

  To verify Qpid prerequisites installation, run the following script on Qpid Server:

  `/<inst-rt>/apigee4/share/installer/apigee-qpid-check-prerequisites.sh`

  PostgreSQL database prerequisite packages: The script, *`apigee-postgres-install-prerequisites.sh`* installs the following PostgreSQL database packages during analytics installation on Red Hat /CentOS:

  - uuid, libxslt, rsync

  The script also adds kernel.sem (500 32000 32 1024), kernel.shmall (4294967296) and kernel.shmmax (68719476736) to `/etc/sysctl.conf`.

  To verify PostgreSQL database prerequisites installation, run the following script on the hosting node:

  `/<inst-rt>/apigee4/share/installer/apigee-postgres-check-prerequisites.sh`

Installation Procedures

The Installation Guide covers the following basic, on-premises installation scenario, as described in the “Overview” section above:

- Standalone installation (SA-SAX)
- 5-host clustered installation (MIN HA-2SAX)
- 9-host clustered installation (Performance HA)
- 13-host clustered installation (Performance HA with separate data zone)
- 12-host clustered installation (MIN API traffic DR / AX HA, API-DN support, 2DC setup)
- 8-host clustered installation (Min HA with API Backend)
- 12-host clustered installation (Performance HA with API Backend)
- 3-host installation (MS-RMP-SAX with MO)

Basic Host Installation

There are two steps to install and configure each host in the system:

- **Installation:** Run the `apigee-install` script. Usage:

```
apigee-install [-h | [-r <path>] [-d <path>] [-j <java home>]]
  -h | --help : Display usage information
  -r | --root <path> : Installation root, "/" not allowed
  -d | --dataroot <path> : Data storage root, "/" not allowed
  -j | --java <java home> : JDK 1.7 home
```

- **Configuration:** Run the `apigee-setup` script. Usage:

```
apigee-setup.sh [ -h | [-p profile] [-f <filename>]]
  -h : Display usage information
  -p <install_profile> : Valid profiles are abs, abp, aio, asa, ds,
     lb, ld, mo, mp, ms, ps, qs, r, rmp, sa, sax, abs, abp
  -f <filename> : Use defaults from CONFIG file
```

Together, these scripts install the Apigee Edge components on the same host or multiple hosts according to the installation scenario (number of server hosts) you choose.

When you run the `apigee-setup`, it prompts you with a choice of Apigee installation profiles:

- Single machine setups:

```
sa  = Gateway Standalone (Management Server, Router and Message Processor)
sax = Analytics Standalone (Qpid and Postgres Server)
aio = All In One (Gateway and Analytics Standalone)
asa = API Backend Standalone (Cassandra, API Backend Stack and Portal)
```

- Cluster node setup for ZooKeeper and Cassandra (min 3 nodes):

```
ds  = Datastore Cluster Node
```

- LDAP setup for OpenLDAP:

```
ld  = LDAP Node
```

- Load Balancer setup (API Backend):

```
lb = Load Balancer
```

- Separate components setup:

```
ms  = Gateway Management Server

r   = Gateway Router

mp  = Gateway Message Processor

rmp = Gateway Router and Message Processor

qs = Analytics Qpid Server

ps  = Analytics Postgres Server

mo  = Monetization Server

abs = API Backend Stack

abp = API Backend Portal
```

Which choice you make depends on what you are installing on which host. This is explained in detail below, for the scenario provided for the installations.

This script also lets you to choose Apigee Analytics configuration:

```
Enable Analytics support y/n (y):
```

This option lets you to configure, for example, Message Processors and Management Servers to search for Analytics (which is most probably installed on a different machine)

Note: When you run `apigee-setup.sh` for the Management Server, make note of the Management Server's IP address to use when installing the Router.

After the install completes, you can check the version of all installed Edge components by using the following command:

```
> /<inst-root>/apigee4/bin/get-version.sh
```

Normal vs Privileged Access

The following table contains the list of scripts that require normal or sudo privileges under `/<inst-root>/apigee4/bin` folder.

Note: The scripts calling Management Server API require the Apigee global system administrator credentials.

Installation Type	Privileged Access	Normal Access
As root or via sudo	all-status.sh all-stop.sh apigee-uninstall.sh backup*.sh chpasswd-openldap.sh chpasswd-system.sh create-trust.sh postgres-*.sh restore*.sh set-autostart.sh start.sh ui-set-smtp.sh unset-autostart.sh	add-env.sh check.sh create-org.sh create-user.sh enable-ax.sh get-version.sh setup-org.sh
As normal user	set-autostart.sh unset-autostart.sh	add-env.sh all-start.sh all-status.sh all-stop.sh apigee-uninstall.sh backup*.sh check.sh chpasswd-openldap.sh chpasswd-system.sh create-org.sh create-user.sh enable-ax.sh get-version.sh postgres-*.sh psql.sh qpid-stat.sh restore*.sh setup-org.sh ui-set-smtp.sh

File System Structure

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

Log Files

Components	Location
Management Server	<inst-root>/apigee4/var/log/apigee/management-server
Router	<inst-root>/apigee4/var/log/apigee/router

Components	Location
Message Processor	<inst-root>/apigee4/var/log/apigee/message-processor
Apigee Qpid Server	<inst-root>/apigee4/var/log/apigee/qpid-server
Apigee Postgres Server	<inst-root>/apigee4/var/log/apigee/postgres-server
Edge UI	<inst-root>/apigee4/var/log/apigee/ui
ZooKeeper	<inst-root>/apigee4/var/log/zookeeper
Cassandra	<inst-root>/apigee4/var/log/cassandra
Qpidd	<inst-root>/apigee4/var/log/qpidd
PostgreSQL database	<inst-root>/apigee4/var/log/postgresql

Data

Components	Location
Management Server	<data-root>/apigee4/data/apigee/management-server
Router	<data-root>/apigee4/data/apigee/router
Message Processor	<data-root>/apigee4/data/apigee/message-processor
Apigee Qpid agent	<data-root>/apigee4/data/apigee/qpid-server
Apigee Postgres agent	<data-root>/apigee4/data/apigee/postgres-server
ZooKeeper	<data-root>/apigee4/data/zookeeper
OpenLDAP	<data-root>/apigee4/conf/openldap *
Cassandra	<data-root>/apigee4/data/cassandra/data
Qpidd	<data-root>/apigee4/data/qpid/data
PostgreSQL database	<data-root>/apigee4/data/postgres/pgdata

* OpenLDAP combines configuration and data into one folder by default.

Init Scripts

Components	Location
Management server	<inst-root>/apigee4/etc/init.d/apigee-management-server
Router	<inst-root>/apigee4/etc/init.d/apigee-router
Message processor	<inst-root>/apigee4/etc/init.d/apigee-message-processor
Apigee Qpid agent	<inst-root>/apigee4/etc/init.d/apigee-qpid-server
Apigee Postgres agent	<inst-root>/apigee4/etc/init.d/apigee-postgres-server
Edge UI	<inst-root>/apigee4/etc/init.d/apigee-ui

Components	Location
ZooKeeper	<inst-root>/apigee4/etc/init.d/apigee-zookeeper
Cassandra	<inst-root>/apigee4/etc/init.d/apigee-cassandra
Qpidd	<inst-root>/apigee4/etc/init.d/apigee-qpidd
PostgreSQL database	<inst-root>/apigee4/etc/init.d/apigee-postgresql

Proxy JAR Files

There's a directory, `<inst-root>/apigee4/var/apigee/custom_jars/` where you can add JARs for use with Java callouts (note that `<inst-root>/apigee4/var` is a symbolic link to `<data-root>/apigee4/data`). This directory is not disturbed during upgrades and is automatically available in classpath.

Standalone 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.

The standalone installation consists of following basic steps:

1) One host runs the Gateway standalone installation setup.
2) The other host runs Analytics standalone setup, comprising Qpid and Postgres

These steps are detailed in the following sections.

Install Standalone Gateway: Machine 1

Installation

1) Unzip the application distribution file from your download directory, and change directory (`cd`) into `apigee-edge-4.15.07.00.`

   ```
   unzip /download_dir/apigee-edge-4.15.07.00.zip
   ```

   ```
   cd apigee-edge-4.15.07.00
   ```

   Note: Unzip the file on the Linux machine itself. Do not unzip it on another machine, such as a Windows machine, and then copy the unzipped files to the Linux machine.

2) At the command prompt, run the `apigee-install` script.

   ```
   ./apigee-install.sh
   ```

3) Enter the JDK 1.7 path to verify the correct Java version.
4) Specify the installation root (<inst-root>) under which all software and configuration files will be stored. Specify the default (/opt) unless you have a specific requirement.
5) Specify the data root (<data-root>) under which all runtime data will be stored. Specify the default (/opt) unless you have a specific requirement.

With the above steps, Apigee Edge installation and configuration files are stored under installation root (<inst-root>) and runtime data (<data-root>) is stored under data root. Now you need to configure the system.

Configuration

Note: Make a note of the server's IP address to use later in the installation procedure.

1) At the command prompt, run the `apigee-setup` script.

   ```
   /<inst-root>/apigee4/share/installer/apigee-setup.sh
   ```

2) When you are prompted to choose Apigee profile, choose option “sa” for the Gateway standalone installation.

3) Specify the new Apigee Edge email address for the global system administrator.

   Note: You must remember global Apigee Edge admin credential to gain access to the UI and API calls.

4) Specify the Apigee admin password for the global system administrator. Retype the password, and then wait as the system installs the components.

   Note: The password must be at least 8 characters.

   Once it fetches the correct IP address and admin credentials, it starts ZooKeeper and Cassandra and then finishes the configuration. The configuration successfully set up schemas in Cassandra.

5) Specify the new root password for LDAP. Retype the password, and then wait as the system installs the components.

   This will successfully change the LDAP password on the host. This confirms that OpenLDAP schemas and password are successfully configured.

6) When you are prompted to choose analytics support, press “y” to enable analytics support.

7) Enter the IP addresses or DNS name of the ZooKeeper node. This is the same IP or DNS as the machine on which you are installing Edge.

8) Specify the router and message processor POD. Choose the default POD (gateway) unless you have a specific requirement.

9) Specify the path to the Apigee license file. For more information, see Licensing.

10) Specify if you want the Message Processor/Router to bind to 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).

11) Specify the SMTP host and port. SMTP allows UI to send password confirmation mails.

    Note: You can skip SMTP configuration now and configure later by using `/<inst-root>/apigee4/bin/ui-set-smtp.sh`.

    a) When you are prompted to use SSL, press “y” to enable SSL support.

    b) Specify the email address for the SMTP user.

    c) Specify the password for the SMTP user. Retype the password, and then wait as the system installs the components.

    On successful configuration, it registers all datastores in the server. It also enables the security for management server. SMTP is also configured and this will allow the UI to send password confirmation mails. A successful configuration returns the 20X HTTP response.

Note that JMX is enabled by default for Cassandra. The JMX remote access to Cassandra does not require a password. You can change this by running `cassandra-jmx-auth.sh` and refer users to it in installation in `apigee-setup.sh`. This may require sudo privileges to create the jmxremote.password file.

```
sudo /<inst-root>/apigee4/bin/cassandra-jmx-auth.sh
```

The following components are successfully up and running:

- Management Server
- Message Processor
- Router
- Edge UI
- OpenLDAP
- ZooKeeper
- Cassandra

12) Make a note of the management console URL. When the installation completes, the management console URL is displayed. It looks similar to the following: `http://192.168.124.111:9000/`

    This is the URL you enter into a browser to access the Apigee Edge user interface. Please note that you will only be able to access the UI after you associate a user with an organization as part of the onboarding process described below in the section Onboarding.

13) **(Optional)** - This release of Edge installs Cassandra Version 2.0.15, which supports password-based authentication. You can enable Cassandra authentication as part of the install. If enabled, any access to Cassandra requires a valid username and password.

    To enable Cassandra authentication, see the instructions in the Edge *Operations Guide* available on the Apigee ftp site: ftp://ftp.apigee.com/.

This completes the Apigee Edge and its associated components setup. The onboarding (creating an organization, environment, and user and associated roles) steps are explained in the Onboarding section.

Testing

This section provides brief descriptions of test scripts that are provided with the installation distribution file.

Test scripts

The following test creates a test organization, environment and API, then accesses the API, and finally cleans up the test entries.

1) CD to the `test` directory the `test` directory under `/<inst-root>/apigee4`.

2) Run the command, `./test1-sa.sh`

3) Specify the Apigee admin password.

   Note: This is the global system administrator password that you have set in the Apigee Edge installation.

4) On successful administrator password entry, the script does the following:

   a. It fetches the server UUID.

   b. It creates an organization and associates POD with the organization.

   c. It creates an environment and associates Message Processor with the environment.

   d. It creates a virtual host.

   e. It imports a simple passthrough proxy and deploys the application to the “test” environment.

   f. It executes the test to make sure everything is working as expected.

   g. Finally, the script undeploys and then deletes the API proxy, virtual host, environment, and organization.

A successful test returns the 20X HTTP response.

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

Install Standalone Analytics: Machine 2

Note: For root installation, this step installs the prerequisites for Qpid and Postgres installation via global yum. In order to allow non-root installation, Analytics Servers (Qpid, Postgres) have some prerequisites that must be installed prior to installation. For more information, see "Analytics dependency files" under Installation Checklist section.

Note: If you install analytics as a user other than the user who installed the Management Server, then you must edit the `apigee4/conf/apigee/management-server/query-service.properties` file on the Management Server. In that file, set the `pgDefaultUser` and `dwDefaultUser` properties to the username of the person who installed analytics.

Installation

1) Unzip the application distribution file from your download directory, and change directory (cd) into `apigee-edge-4.15.07.00`

   ```
   unzip /download_dir/apigee-edge-4.15.07.00.zip
   ```

   ```
   cd apigee-edge-4.15.07.00
   ```

   Note: Unzip the file on the Linux machine itself. Do not unzip it on another machine, such as a Windows machine, and then copy the unzipped files to the Linux machine.

2) At the command prompt, run the `apigee-install` script.

   ```
   ./apigee-install.sh
   ```

3) Enter the JDK 1.7 path to verify the correct Java version.
4) Specify the installation root (<inst-root>) under which all software and configuration files will be stored. Specify the default (/opt) unless you have a specific requirement.
5) Specify the data root (<data-root>) under which all runtime data will be stored. Specify the default (/opt) unless you have a specific requirement.

With the above steps, Apigee Edge installation and configuration files are stored under installation root (<inst-root>) and runtime data (<data-root>) is stored under data root. Now you need to configure the Analytics service with the Management Server.

Configuration

1) At the command prompt, run the `apigee-setup` script.

   ```
   sudo /<inst-root>/apigee4/share/installer/apigee-setup.sh
   ```

2) When you are prompted to choose Apigee profile, choose option “sax” for the Analytics standalone installation.
3) Enter the IP address or DNS name of Management Server that you noted earlier and used in the Gateway installation above.

   Caution: Do not use the host name mapping to 127.0.0.1. It may cause some problems when they attempt to resolve the host name of the machine.

4) Specify the Apigee Edge email address for the global system administrator.
5) Specify the Apigee admin password for the global system administrator.

   Note: This is the global system administrator credential that you have set in the Gateway installation.

6) Enter the IP addresses or DNS names of ZooKeeper. Ensure that you separate the entries by spaces.

   Note: During all Apigee components configuration (except for Management Server), `apigee-setup.sh` prompts only for the ZooKeeper nodes in this datacenter. It does not prompt for Cassandra nodes.

   On successful configuration, it registers the Qpid Daemon (Qpidd), Apigee Qpid Server, PostgreSQL database, and Apigee Postgres Server with the Management Server. A successful configuration returns the 20X HTTP response. This completes the Analytics installation.

Sanity Validation

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

1) Verify that Management Server and Qpid Server are in central POD.

   On Management Server, run the following CURL command:

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

2) Verify that Router and Message Processor are in gateway POD.

   On Management Server, run the following CURL command:

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

3) Verify that Postgres server is in analytics POD.

   On Management Server, run the following CURL command:

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

Install SmartDocs: Machine 1

Note: Ensure that you install SmartDocs on **Machine 1** where Management Server is installed.

Note: If you use a load balancer, install SmartDocs after installing and configuring the load balancer.

1) Install and configure SmartDocs by running the script on **Machine 1:**

   ```
   /<inst-root>/apigee4/bin/setup-smartdocs.sh
   ```

2) Unless you have a good reason to change them, accept the default values for the organization name (`smartdocs`), environment (`prod`), virtual host name (`default`) virtual host port (`59002`), and virtual host alias (empty).

3) Enter the system administrator password.

4) Enter the IP address of the router or load balancer. In this configuration, use the IP address of Machine 1.

5) The script automatically restarts the Management Server.

6) Restart each PostgreSQL database server, one server at a time, by using the command:

   ```
   /<inst-root>/apigee4/bin/apigee-service postgresql restart
   ```

The `/<inst-root>/apigee4/conf/apigee/management-server/apimodel.properties` file contains two properties, `authurl` and `testapi.proxy`, that are automatically set by the `apigee-setup.sh` script.

The URL specified by `authurl` is used to authenticate the user making edits to a SmartDocs page, and has the form:

```
http://managementServerIP:8080/v1/users/{user}/authenticate
```

The URL specified by `testapi.proxy` sets the location of the Edge router where SmartDocs page requests are sent. The `apigee-setup.sh` script assumes that the router is listening to port 59002, and the URL has the form:

```
http://routerIP:59002/smartdocs/v1
```

To change this port, edit `apimodel.properties` and set the `testapi.proxy` property to the correct IP and port number.

You also have to update the `default` virtual host created in the `smartdocs` organization to use the new port number. For example, you can change the port number to 55555 by using the following cURL command:

```
curl -X PUT -H "Content-Type:application/xml" \
http://<ms-IP>:8080/v1/o/smartdocs/e/prod/virtualhosts/default \
-d '<VirtualHost name="default">
     <HostAliases/>
     <Interfaces/>
     <Port>55555</Port>
   </VirtualHost>' \
-u myname:mypass
```

For more information on modifying a virtual host, see Creating a virtual host for an on-premises Edge installation.

This is also the IP and port number that you use to configure the Developer Services portal to connect with SmartDocs, as described here: http://apigee.com/docs/developer-services/content/using-smartdocs-document-apis
(or http://apigee.com/docs/content/documentation-archives to find the docs that correspond to earlier versions of the product).

Note: If you have multiple routers connected to a load balancer, specify the IP address and port number of the load balancer, not of a router.

If you allow developers to access SmartDocs pages from outside your firewall, these URLs must be publicly available. If you edit this file to change the URLs, you must restart the Management Server.

For SmartDocs troubleshooting information, see *Apigee Edge Operations Guide,* available on the Apigee ftp site: ftp://ftp.apigee.com/.

Onboarding

Note: Ensure that you execute the onboarding steps on the **Machine 1** where Management Server is installed.

This section explains the creation of users, organizations and environments (test and prod) in the server. It also allows you to enable analytics for the environments.

Note: When creating an organization, you automatically create a predefined set of user roles in the organization. Currently, Apigee Edge for Private Cloud supports the roles - orgadmin, readonlyadmin, opsadmin, user and businessuser, all having a default permission of full access to entities (APIs, API products, apps, developers, and reports) in an Apigee organization. Depending on your needs, you can customize the pre-defined or configure more complex roles and permissions using RBAC API. See http://apigee.com/docs/api/user-roles (or http://apigee.com/docs/content/documentation-archives to find the docs that correspond to earlier versions of the product).

The onboarding script, `/<inst-root>/apigee4/bin/setup-org.sh` uses a combination of following scripts that can also be executed independently.

- `create-user.sh`
- `create-org.sh`
- `create-roles.sh`
- `add-env.sh`
- `enable-ax.sh`

By default, the maximum length of the organization name and environment name is 20 characters when using any of the following scripts: `setup-org.sh`, `add-env.sh`, and `create-org.sh`. This limit does not apply if you use the Edge API directly to create the organization or environment.

You can override this limit by changing the following properties in `/<inst-root>/apige4/bin/apigee-env.sh`:

- `APIGEE_ORG_NAME_LENGTH=20`
- `APIGEE_ENV_NAME_LENGTH=20`

Onboarding Steps

1) Run the command on Machine 1 (or on the Management Server node for configurations where the Management Server is not on Machine 1):

```
/<inst-root>/apigee4/bin/setup-org.sh
```

2) Specify the system administrator password.

   Note: This is the **global system administrator** password that you have set in the Apigee Edge installation.

3) Creates a new user. This user functions as the **organization administrator** for the organization. Each organization can have a different organization administrator:

    a) When you are prompted to create a new user, press “y”.

    b) Enter the user’s email ID as username.

    c) Enter the user's first name.

    d) Enter the user's last name.

    e) Specify the new password for user. Retype the password, and then wait as the system adds the user.

    A successful user creation returns the 20X HTTP response.

    Note: As an alternative to creating a new user, you can specify the email address of an existing user to function as the organization administrator.

4) Creates organization

    a) When prompted, enter the organization name. **Organization names must use all lowercase letters and cannot contain spaces, underscores, or periods.** Do not create organization names that include uppercase letters.

    This creates an organization and associates the gateway PODs of all regions for the organization. Also, the system admin is added as an organization admin, which means that the system admin can add new users or modify existing users in the organization.

    A successful organization creation returns the 20X HTTP response.

5) Creates environment

    The script first validates the existence of Message Processors.

    a) When prompted, enter the following:

        - Environment name as “prod”
        - Virtual host port as “9001”
        - Virtual host name as "default"
        - Optionally virtual host alias. If you already have a DNS record that allows access to the virtual host, specify the host alias and port, for example, “myapi.example.com:9001”, otherwise leave blank.

    b) When prompted to add another environment, select "y":

        - Environment name as “test”
        - Virtual host port as “9002”
        - Virtual host name as "default"

- Optionally virtual host alias. If you already have a DNS record that allows access to the virtual host, specify the host alias and port, for example, “myapi.example.com:9002” ”, otherwise leave blank.

This creates two environments – `prod` and `test` – and associates each environment with the Message Processor(s). Also, a virtual host is created for each environment.

A successful association returns the 20X HTTP response.

After running the script, you can access your APIs deployed to the `prod` environment by using a URL in the form:

```
http://<router-ip>:9001/{project-base-path}/{resource-name}
```

If the API is deployed to the `test` environment, use a URL in the form:

```
http://<router-ip>:9002/{project-base-path}/{resource-name}
```

Note that these URLs use the IP address of the Router and the virtual host port on the Router to access your APIs. Until you create virtual hosts with host aliases, and DNS records for your virtual hosts, you can use the IP address and port explicitly to access your APIs. For more, see http://apigee.com/docs/api-services/content/virtual-hosts.

6) If prompted, enable Analytics for prod/test

   a) When prompted to enable Analytics for the given environment, press “y”.

The script validates the existence of Qpid and Postgres in the PODs of all regions. Then it starts the Analytics onboarding for the given organization and environment. A successful onboarding returns the 20X HTTP response.

Note: Ensure that you must have at least 10GB of free disk space on the server where Qpid is installed. This disk space is required for Qpid data folder (usually `"/<data-root>/apigee/data/qpid"`) that is used during the Analytics enabling process.

Note: If you enable analytics for one environment in an organization, you must enable analytics for all environments in the organization.

Verification

On completion of onboarding, verify the status of the system by issuing the following CURL commands on the Machine 1 (or on the machine where the Management Server is installed for configurations where that is not Machine 1).

You can 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
```

If you enabled analytics, then use this command:

```
curl -u username:password
http://localhost:8080/v1/organizations/<orgname>/environments/<envname>/p
rovisioning/axstatus
```

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

```
> /<inst-root>/apigee4/bin/psql.sh
```

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

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

Enter `\q` to quit `psql`.

Logging in to the Edge UI

Now that you are associated with an organization, you can 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.

To login:

1) 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.

2) 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.

3) Click Sign In, the browser redirects to:

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

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

4) 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

Administrating Edge

For information on how to perform system administration tasks, see the *Apigee Edge Operations Guide,* available on the Apigee ftp site: ftp://ftp.apigee.com/. That document contains procedures for:

- Creating new users
- Creating new organizations, environment, and virtual hosts
- Modifying system passwords
- Monitoring the system
- Configuring SSL
- and information on performing many other system administration tasks

5-host Clustered 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.

The five-host clustered installation consists of following basic steps:

1) Install the ZooKeeper and Cassandra clusters on first three hosts.
2) Install the Apigee Management Server on the first host, which also installs OpenLDAP and the Edge UI.
3) Install the Apigee Message Processor + Router on the second and third host.
4) Install the Apigee Analytics standalone on the fourth and fifth host.
5) Set up master-standby replication between Postgres servers.

These steps are detailed in the following sections.

Install Datastore Cluster Node: Machine 1, 2 and 3

Installation

1) Unzip the application distribution file from your download directory, and change directory (`cd`) into `apigee-edge-4.15.07.00`

   ```
   unzip /download_dir/apigee-edge-4.15.07.00.zip
   ```

   ```
   cd apigee-edge-4.15.07.00
   ```

   Note: Unzip the file on the Linux machine itself. Do not unzip it on another machine, such as a Windows machine, and then copy the unzipped files to the Linux machine.

2) At the command prompt, run the `apigee-install` script.

   ```
   ./apigee-install.sh
   ```

3) Enter the JDK 1.7 path to verify the correct Java version.
4) Specify the installation root (<inst-root>) under which all software and configuration files will be stored. Specify the default (/opt) unless you have a specific requirement.
5) Specify the data root (<data-root>) under which all runtime data will be stored. Specify the default (/opt) unless you have a specific requirement.

With the above steps, Apigee Edge installation and configuration files are stored under installation root (<inst-root>) and runtime data (<data-root>) is stored under data root. Now you need to configure the datastore cluster node.

Configuration

1) At the command prompt, run the `apigee-setup` script.

   ```
   /<inst-root>/apigee4/share/installer/apigee-setup.sh
   ```

2) When you are prompted to choose Apigee profile, choose option "ds" for the ZooKeepeer and Cassandra cluster node installation.

3) When you are prompted to configure ZooKeeper node on this machine, press "y".

4) Enter ZooKeeper nodes. Ensure that you enter 3 or more ZooKeeper IP addresses or DNS names separated by spaces.

   During datastore cluster configuration, `apigee-setup.sh` prompts for all ZooKeeper nodes with ":observer". Specify that modifier only when creating multiple data centers as described in a 12-host installation. In a single data center installation, as you are installing here, omit that modifier.

   Note: The IP addresses or DNS names must be listed in the same order on all ZooKeeper nodes in the cluster in order to match the IDs. For example, ZooKeeper ID for this host is 1 and followed by 2 and 3 for the successive hosts.

5) When you are prompted to configure Cassandra node on this machine, press "y".

6) Enter Cassandra nodes. Ensure that you enter 3 or more Cassandra IP addresses or DNS names separated by spaces.

   During datastore cluster configuration, `apigee-setup.sh` prompts for all Cassandra nodes with ":dc,ra" modifier. Specify this modifier only when creating multiple data centers as described in a 12-host installation. In a single data center installation, as you are installing here, omit that modifier.

   Note: 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.

7) Specify the Cassandra cluster name.

The configuration successfully completes the datastore setup in the server. A successful configuration returns the 20X HTTP response. Note that JMX is enabled by default for Cassandra. The JMX remote access to Cassandra does not require a password. You can change this by running `cassandra-jmx-auth.sh` and refer users to it in relocated installs in apigee-setup.sh. This may require sudo privileges to create the `jmxremote.password` file.

```
sudo /<inst-root>/apigee4/bin/cassandra-jmx-auth.sh
```

Configure Apigee Management Server: Machine 1

Note: Make a note of the server's IP address to use later in the installation procedure.

1) At the command prompt, run the `apigee-setup` script.

```
/<inst-root>/apigee4/share/installer/apigee-setup.sh
```

2) When you are prompted to choose Apigee profile, choose option “ms” for the Management Server installation (which also installs the Edge UI).

3) Specify the new Apigee Edge email address for the global system administrator.

   Note: You must remember global Apigee Edge admin credential to gain access to the UI and API calls.

4) Specify the Apigee admin password for the global system administrator. Retype the password, and then wait as the system installs the components.

   Note: The password must be at least 8 characters long.

5) When you are prompted to choose “Use Existing ZooKeeper Cluster”, press “y”.

6) When you are prompted to choose “Use Existing Cassandra Cluster”, press “y”.

   Once it fetches the correct IP address and admin credentials, it starts ZooKeeper and Cassandra and then finishes the configuration. A successful configuration returns the 20X HTTP response. The configuration successfully set up schemas in Cassandra.

7) When you are prompted to choose “Use existing LDAP host”, press “n” to install LDAP server on the same machine.

8) When prompted, choose the LDAP server type as option 1, OpenLDAP as standalone.

9) Specify the new root password for LDAP. Retype the password, and then wait as the system installs the components.

   This will successfully change the LDAP password on the Management Server. This confirms that LDAP configuration is successfully completed.

10) When you are prompted to choose Analytics support, press “y” to enable Analytics support.

11) Enter the IP addresses or DNS names of the ZooKeeper nodes. Ensure that you enter the IP addresses or DNS names separated by spaces.

    Note: During datastore cluster configuration, `apigee-setup.sh` prompts for all ZooKeeper nodes with “:observer” modifier and all Cassandra nodes with “:dc,ra” modifier. Specify these modifiers only when creating multiple data centers as described in a 12-host installation across two datacenters. In a single data center installation, as you are installing here, omit those modifiers.

12) Specify the message processor and router POD. Choose the default POD (gateway) unless you have a specific requirement.

13) Specify the path to the Apigee license file. For more information, see Licensing.

14) Specify the SMTP host and port. SMTP allows UI to send password confirmation mails.

    Note: You can skip SMTP configuration now and configure later by using `/<inst-root>/apigee4/bin/ui-set-smtp.sh`.

a) When you are prompted to use SSL, press “y” to enable SSL support.

b) Specify the email address for the SMTP user.

c) Specify the password for the SMTP user. Retype the password, and then wait as the system installs the components.

d) Specify the SMTP host.

On successful configuration, it registers all datastores in the server. It also enables the security for Management Server. A successful configuration returns the 20X HTTP response. The following components are successfully up and running:

- Management Server
- Edge UI
- OpenLDAP

15) Make a note of the management console URL. When the installation completes, the management console URL is displayed. It looks similar to the following:
`http://192.168.124.201:9000/`

This is the URL you enter into a browser to access the Apigee Edge user interface. Please note that you will be able to access the UI when you are associated with an organization.

Install Apigee Router and Message Processor: Machine 2 and 3

Configuration

1) At the command prompt, run the `apigee-setup` script.

```
sudo /<inst-root>/apigee4/share/installer/apigee-setup.sh
```

2) When you are prompted to choose Apigee profile, choose option “rmp” for the Router and Message Processor installation

3) Enter the IP address or DNS name of Management Server used in the Management Server installation above.

Caution: Do not use the host name mapping to 127.0.0.1. It may cause some problems when they attempt to resolve the host name of the machine.

4) Specify the email address for the global system administrator.

5) Specify the Apigee admin password for the global system administrator.

Note: This is the global system administrator credential that you have set in the Management Server installation.

6) When you are prompted to choose Analytics support, press “y” to enable Analytics support.

7) Enter the IP addresses or DNS names of ZooKeeper. Ensure that you enter the IP addresses or DNS names separated by spaces.

Note: During all Apigee components configuration (sans management server), `apigee-setup.sh` prompts only for the ZooKeeper nodes in this datacenter. It does not prompt for Cassandra nodes.

8) Specify the Message Processor and Router POD. Choose the POD name selected during installation (default is "gateway").
9) Specify if you want the Message Processor/Router to bind to 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).

On successful configuration, it registers the Router and Message Processor with the Management Server. A successful configuration returns the 20X HTTP response. This completes the Router and Message Processor installation.

Optionally set Cassandra Authentication on Machine 1, 2, and 3

This release of Edge installs Cassandra to Version 2.0.15, which supports password-based authentication. You can enable Cassandra authentication as part of the install. If enabled, any access to Cassandra requires a valid username and password.

To enable Cassandra authentication, see the instructions in the Edge *Operations Guide,* available on the Apigee ftp site: ftp://ftp.apigee.com/.

Testing

This section provides brief descriptions of test scripts that are provided with the installation.

Test scripts

Run the following test scripts on the **Machine 2** or **3**. The test creates a test organization, environment and API, then accesses the API, and finally cleans up the test entries.

1) CD to the `test` directory under `/<inst-root>/apigee4/`, the Edge installation directory.
2) Run the command:

```
./test1-mp-setup.sh
```

3) Specify the Apigee admin password.

   Note: This is the global system administrator password that you have set in the Apigee Edge installation.

4) On successful entries, the script does the following:
   a. It fetches the server UUID.
   b. It creates an organization and associates POD with the organization.

c. It creates an environment and associates the environment with the Message Processor.

d. It creates a virtual host.

e. It imports a simple passthrough proxy and deploys the application to the environment.

A successful test returns the 20X HTTP response.

Test Execution

Now that you've deployed your passthrough proxy to the environment, you'll want to execute the test to make sure everything is working as expected. Make sure you are on the same machine (**Machine 2** or **3**) where the test scripts reside.

1) CD to the `test` directory under `/<inst-root>/apigee4/`, the Edge installation directory.

2) Run the command:

```
./test1-r-access.sh
```

This will confirm that the entire procedure from installation to test an API proxy is up and running.

Test Cleanup

The following test cleans up the test entries.

1) From the `test` directory under `/<inst-root>/apigee4/,` run the command:

```
./test1-mp-cleanup.sh
```

2) Specify the Apigee admin password.

   Note: This is the global system administrator password that you have set in the Management Server installation.

3) Specify the Message Processor POD.

On successful entries, the script undeploys and then deletes the API proxy, virtual host, environment, and organization.

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

Install Apigee Analytics: Machine 4 and 5

Note: For root installation, this step installs the prerequisites for Qpid and Postgres installation via global yum. In order to allow non-root installation, Analytics Servers (Qpid, Postgres) have some prerequisites that must be installed prior to installation. For more information, see "Analytics dependency files" under Installation Checklist section.

Note: If you install analytics as a user other than the user who installed the Management Server, then you must edit the `apigee4/conf/apigee/management-server/query-service.properties` file on the Management Server. In that file, set the `pgDefaultUser` and `dwDefaultUser` properties to the username of the person who installed analytics.

Installation

1) Unzip the application distribution file, and change directory (`cd`) into `apigee-edge-4.15.07.00`

   ```
   unzip apigee-edge-4.15.07.00.zip
   ```

   ```
   cd apigee-edge-4.15.07.00
   ```

   Note: Unzip the file on the Linux machine itself. Do not unzip it on another machine, such as a Windows machine, and then copy the unzipped files to the Linux machine.

2) At the command prompt, run the `apigee-install` script:

   ```
   ./apigee-install.sh
   ```

3) Enter the JDK 1.7 path to verify the correct Java version.

4) Specify the installation root (<inst-root>) under which all software and configuration files will be stored. Specify the default (/opt) unless you have a specific requirement.

5) Specify the data root (<data-root>) under which all runtime data will be stored. Specify the default (/opt) unless you have a specific requirement.

With the above steps, Apigee Edge installation and configuration files are stored under installation root (<inst-root>) and runtime data (<data-root>) is stored under data root. Now you need to configure the Analytics service with the Management Server.

Configuration

1) At the command prompt, run the `apigee-setup`script:

   ```
   /<inst-root>/apigee4/share/installer/apigee-setup.sh
   ```

2) When you are prompted to choose Apigee profile, choose option “sax” for the Analytics standalone installation.

3) Enter the IP address or DNS name of the Management Server used in the Management Server installation above.

   Caution: Do not use the host name mapping to 127.0.0.1. It may cause some problems when they attempt to resolve the host name of the machine.

4) Specify the Apigee Edge email address for the global system administrator.

5) Specify the Apigee admin password for the global system administrator.

Note: This is the global system administrator credential that you have set in the Management Server installation.

6) Enter the IP addresses or DNS names of ZooKeeper. Ensure that you enter the IP addresses or DNS names separated by spaces.

Note: During all Apigee components configuration (sans Management Server), `apigee-setup.sh` prompts only for the ZooKeeper nodes in this datacenter. It does not prompt for Cassandra nodes.

On successful configuration, it registers the Qpid Daemon (Qpidd), Apigee Qpid Server, PostgreSQL database, and Apigee Postgres Server with the Management Server. A successful configuration returns the 20X HTTP response. This completes the Analytics installation.

Note: Make a note of the server’s IP address to use when setting up master-standby replication for Postgres.

Set up Master-Standby Replication for Postgres

You can now set up a master-standby replication between Postgres servers. This process has two parts:

1. Setting password-less login (SSH Key) between master and standby Postgres machines.
2. Replicating master-standby Postgres.

Both procedures are described below.

Enabling Password-Less Login (SSH Key) between Master and Standby Postgres Machine

The following are the steps to set up password-less login used for accessing master/standby Postgres server as user ‘apigee’. Once completed, the master machine can SSH to the standby machine as the user 'apigee' without specifying a password, and vice versa.

Troubleshooting the connection

If you perform the procedure below but still cannot connect by using SSH without a password, you can try disabling SELinux, or setting it to permissive, on the master and standby machines. To configure SELinux:

1. Edit the `/etc/sysconfig/selinux` file:

   ```
   > sudo vi /etc/sysconfig/selinux
   ```

2. Set `SELINUX=disabled` or `SELINUX=permissive`
3. Save your edits.
4. Restart the machine and then restart Edge:

   ```
   > /<inst-root>/apigee4/bin/all-start.sh
   ```

Enabling password-less login:

1) On the master Postgres server

   a) Execute the following command to log in as the user 'apigee':

   ```
   > su - apigee
   ```

   If you are not logged in as root, use:

   ```
   > sudo su - apigee
   ```

   b) Execute the following command to generate public-private key for the user

   Note: Use the default filename and empty passphrase as shown below because logins are attempted from automated scripts.

   ```
   -bash-3.2$ cd ~
   -bash-3.2$ ssh-keygen -t rsa -f /home/apigee/.ssh/id_rsa -N ""
   ```

   The command writes the private key to `/home/apigee/.ssh/id_rsa` and the public key to `/home/apigee/.ssh/id_rsa.pub`.

2) On the standby Postgres server, repeat Step 1.

3) Copy the standby server's public key from the `/home/apigee/.ssh/id_rsa.pub` file to the file named `/home/apigee/.ssh/authorized_keys` on the master server.

   If the file `authorized_keys` does not exist, create it.

4) Set the permissions on `authorized_keys` on the master server:

   ```
   -bash-3.2$ chmod 600 /home/apigee/.ssh/authorized_keys
   ```

5) Copy the master server's public key from the `/home/apigee/.ssh/id_rsa.pub` file to the file named `/home/apigee/.ssh/authorized_keys` on the standby server.

   If the file `authorized_keys` does not exist, create it.

6) Set the permissions on `authorized_keys` on the standby server:

   ```
   -bash-3.2$ chmod 600 /home/apigee/.ssh/authorized_keys
   ```

7) Confirm that you can connect to the standby server from the master server with no password:

   ```
   -bash-3.2$ ssh apigee@StandbyNodeIP
   ```

   where ***`StandbyNodeIP`*** is the IP address of the standby server. If you are prompted for a password, then there is a configuration issue with the key. Retry this procedure until you can connect without a password.

8) Confirm that you can connect to the master server from the standby server with no password:

```
-bash-3.2$ ssh apigee@MasterNodeIP
```

where ***MasterNodeIP*** is the IP address of the master server. If you are prompted for a password, then there is a configuration issue with the key. Retry this procedure until you can connect without a password.

Replicating Master-Standby Postgres

In this scenario, Machine 4 acts as a master (primary Postgres server) and Machine 5 as a standby (secondary Postgres server).

1) On Machine 5, stop PostgreSQL database using the command:

```
/<inst-root>/apigee4/bin/apigee-service postgresql stop
```

2) On Machine 4, run the following command as the user 'apigee':

```
su - apigee
```

Then:

```
/<inst-root>/apigee4/share/installer/apigee-postgres-replication-setup.sh
```

a) When you are prompted to choose Postgres server type, choose option “m” for the master.

b) Enter the IP address or DNS name of the standby server.

Note: If you use a hostname in place of an IP address, ensure that this hostname is resolvable with "`dig`".

c) Enter data directory path of Postgres peer node, or accept the default.

d) Enter the path of the private SSH key file for accessing ‘standby’. If you used the procedure above to configure password-less login from master to standby, the path is `/home/apigee/.ssh/id_rsa`.

3) On Machine 5, repeat step 2. Note that here you choose option “s” for the standby and enter the IP address or DNS name of the master server. The script also starts PostgreSQL database.

Once the replication script reaches the peer node via SSH key, it starts Postgres and then completes the replication setup. At the end of step 3, you can see the sync location properties of master and standby servers. The system should display identical values for both servers to ensure a successful replication.

4) 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.

   a) On the master node, run:

      ```
      /<inst-root>/apigee4/bin/postgres-check-master.sh
      ```

      Validate that it says it is the master.

   b) On the standby node:

      ```
      /<inst-root>/apigee4/bin/postgres-check-standby.sh
      ```

      Validate that it says it is the standby.

The Apigee Postgres Server service determines its state - active or standby by querying whether the PostgreSQL database running on the machine is master or standby. This has two implications:

- Postgres Server starts successfully on the machine where PostgreSQL database is already running.
- When the role of PostgreSQL database changes from standby to master or vice versa, the respective Postgres Server must be restarted to detect the new state.

Sanity Validation

Follow the same steps as described in the Standalone Installation section: Sanity Validation.

Install SmartDocs: Machine 1

Note: Ensure that you install SmartDocs on **Machine 1** where Management Server is installed.

Note: If you use a load balancer, install SmartDocs after installing and configuring the load balancer.

1) Install and configure SmartDocs by running the following script on **Machine 1**:

   ```
   /<inst-root>/apigee4/bin/setup-smartdocs.sh
   ```

2) Unless you have a good reason to change them, accept the default values for the organization name (`smartdocs`), environment (`prod`), virtual host name (`default`) virtual host port (`59002`), and virtual host alias (empty).
3) Enter the system administrator password.
4) Enter the IP address of the router or load balancer.
5) The script automatically restarts the Management Server.

6) Restart each PostgreSQL database, one server at a time, by using the command:

```
/<inst-root>/apigee4/bin/apigee-service postgresql restart
```

The `/<inst-root>/apigee4/conf/apigee/management-server/apimodel.properties` file contains two properties, `authurl` and `testapi.proxy`, that are automatically set by the `apigee-setup.sh` script.

The URL specified by `authurl` is used to authenticate the user making edits to a SmartDocs page, and has the form:

```
http://managementServerIP:8080/v1/users/{user}/authenticate
```

The URL specified by `testapi.proxy` sets the location of the Edge router where SmartDocs page requests are sent. The `apigee-setup.sh` script assumes that the router is listening to port 59002, and the URL has the form:

```
http://routerIP:59002/smartdocs/v1
```

To change this port, edit `apimodel.properties` and set the `testapi.proxy` property to the correct IP and port number.

You also have to update the `default` virtual host created in the `smartdocs` organization to use the new port number. For example, you can change the port number to 55555 by using the following cURL command:

```
curl -X PUT -H "Content-Type:application/xml" \
http://<ms-IP>:8080/v1/o/smartdocs/e/prod/virtualhosts/default \
-d '<VirtualHost name="default">
     <HostAliases/>
     <Interfaces/>
     <Port>55555</Port>
   </VirtualHost>' \
-u myname:mypass
```

For more information on modifying a virtual host, see Creating a virtual host for an on-premises Edge installation.

This is also the IP and port number that you use to configure the Developer Services portal to connect with SmartDocs, as described here: http://apigee.com/docs/developer-services/content/using-smartdocs-document-apis
(or http://apigee.com/docs/content/documentation-archives to find the docs that correspond to earlier versions of the product).

Note: If you have multiple routers connected to a load balancer, specify the IP address and port number of the load balancer, not of a router.

If you allow developers to access SmartDocs pages from outside your firewall, these URLs must be publicly available. If you edit this file to change the URLs, you must restart the Management Server.

For SmartDocs troubleshooting information, see *Apigee Edge Operations Guide,* available on the Apigee ftp site: ftp://ftp.apigee.com/.

Onboarding

Follow the same steps as described in the Standalone Installation section: Onboarding.

Note:

- Organization names must use all lowercase letters and cannot contain spaces, underscores, or periods.
- Ensure that you execute the onboarding steps on the **Machine 1** where Management Server is installed.
- During onboarding, when you are prompted to choose multiple Message Processors, select "y" to associate all Message Processors. Otherwise, press "n" to proceed with single Message Processor.
- During onboarding, when enabling analytics for prod/test, you need to update master or standby status of PostgreSQL database (in two PostgreSQL database setup) when prompted. Or you need to manually run `enable-ax.sh` script and update the same.

Verification

Follow the same steps as described in the Standalone Installation section: Verification.

Logging into Edge UI

Follow the same steps as described in the Standalone Installation section: Log into Edge UI.

Administrating Edge

For information on how to perform system administration tasks, see the *Apigee Edge Operations Guide,* available on the Apigee ftp site: ftp://ftp.apigee.com/. That document contains procedures for:

- Creating new users
- Creating new organizations, environment, and virtual hosts
- Modifying system passwords
- Monitoring the system
- Configuring SSL
- and information on performing many other system administration tasks

9-host Clustered 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.

The nine-host clustered installation consists of following basic steps:

1) Install the ZooKeeper and Cassandra clusters on first three hosts.
2) Install the Apigee Management Server on the first host, which also installs OpenLDAP and the Edge UI.
3) Install the Apigee Message Processor + Router on the fourth and fifth host.
4) Install the Qpid Server on the sixth and seventh host.
5) Install the Postgres server on the eighth and ninth host.
6) Set up master-standby replication between Postgres servers.
7) Install SmartDocs.

These steps are detailed in the following sections.

Install Datastore Cluster Node: Machine 1, 2 and 3

Follow the same steps as described in the Five-host Clustered Installation section: Install Datastore Cluster Node.

Configure Apigee Management Server: Machine 1

Follow the same steps as described in the Five-host Clustered Installation section: Configure Apigee Management Server: Machine 1.

Install Apigee Router and Message Processor: Machine 4 and 5

Follow the same steps as described in the Five-host Clustered Installation section: Install Apigee Router and Message Processor.

Optionally set Cassandra Authentication

Follow the same steps as described in the Five-host Clustered Installation section: Optionally set Cassandra Authentication on Machine 1, 2, and 3.

Testing

Follow the same steps as described in the Five-host Clustered Installation section: Testing.

Install Apigee Analytics Qpid Server: Machine 6 and 7

Note: For root installation, this step installs the prerequisites for Qpid and Postgres installation via global yum. In order to allow non-root installation, Analytics Servers (Qpid, Postgres) have some prerequisites that must be installed prior to installation. For more information, see "Analytics dependency files" under the Installation Checklist section.

Note: If you install analytics as a user other than the user who installed the Management Server, then you must edit the `apigee4/conf/apigee/management-server/query-service.properties` file on the Management Server. In that file, set the `pgDefaultUser` and `dwDefaultUser` properties to the username of the person who installed analytics.

Installation

1) Unzip the application distribution file from your download directory, and change directory (`cd`) into `apigee-edge-4.15.07.00`

   ```
   unzip /download_dir/apigee-edge-4.15.07.00.zip
   ```

   ```
   cd apigee-edge-4.15.07.00
   ```

   Note: Unzip the file on the Linux machine itself. Do not unzip it on another machine, such as a Windows machine, and then copy the unzipped files to the Linux machine.

2) At the command prompt, run the `apigee-install` script.

   ```
   ./apigee-install.sh
   ```

3) Enter the JDK 1.7 path to verify the correct Java version.

4) Specify the installation root (<inst-root>) under which all software and configuration files will be stored. Specify the default (/opt) unless you have a specific requirement.

5) Specify the data root (<data-root>) under which all runtime data will be stored. Specify the default (/opt) unless you have a specific requirement.

With the above steps, Apigee Edge installation and configuration files are stored under installation root (<inst-root>) and runtime data (<data-root>) is stored under data root. Now you need to configure the Qpid Server with the Management Server.

Configuration

1) At the command prompt, run the `apigee-setup.sh` script.

   ```
   /<inst-root>/apigee4/share/installer/apigee-setup.sh
   ```

2) When you are prompted to choose Apigee profile, choose option “qs” for the Qpid Server installation.

3) Enter the Management Server's IP address or DNS name used in the Management Server installation above.

   Caution: Do not use the host name mapping to 127.0.0.1. It may cause some problems when they attempt to resolve the host name of the machine.

4) Specify the Apigee Edge email address for the global system administrator.

5) Specify the Apigee admin password.

   Note: This is the global system administrator credential that you have set in the management server installation.

6) Enter the IP addresses or DNS names of ZooKeeper. Ensure that you enter the IP addresses or DNS names separated by spaces.

On successful configuration, it registers the Qpid Daemon (Qpidd) and Qpid Server with the Management Server. A successful configuration returns the 20X HTTP response. This completes the Qpid Server installation.

Install Apigee Analytics Postgres Server: Machine 8 and 9

Note: For root installation, this step installs the prerequisites for Qpid and Postgres installation via global yum. In order to allow non-root installation, Analytics Servers (Qpid, Postgres) have some prerequisites that must be installed prior to installation. For more information, see "Analytics dependency files" under the Installation Checklist section.

Note: If you install analytics as a user other than the user who installed the Management Server, then you must edit the `apigee4/conf/apigee/management-server/query-service.properties` file on the Management Server. In that file, set the `pgDefaultUser` and `dwDefaultUser` properties to the username of the person who installed analytics.

Installation

1) Unzip the application distribution file from your download directory, and change directory (`cd`) into `apigee-edge-4.15.07.00`

   ```
   unzip /download_dir/apigee-edge-4.15.07.00.zip
   ```

   ```
   cd apigee-edge-4.15.07.00
   ```

   Note: Unzip the file on the Linux machine itself. Do not unzip it on another machine, such as a Windows machine, and then copy the unzipped files to the Linux machine.

2) At the command prompt, run the `apigee-install` script.

   ```
   ./apigee-install.sh
   ```

3) Enter the JDK 1.7 path to verify the correct Java version.

4) Specify the installation root (<inst-root>) under which all software and configuration files will be stored. Specify the default (/opt) unless you have a specific requirement.

5) Specify the data root (<data-root>) under which all runtime data will be stored. Specify the default (/opt) unless you have a specific requirement.

With the above steps, Apigee Edge installation and configuration files are stored under installation root (<inst-root>) and runtime data (<data-root>) is stored under data root. Now you need to configure the Postgres Server with the Management Server.

Configuration

1) At the command prompt, run the `apigee-setup.sh` script:

```
sudo /<inst-root>/apigee4/share/installer/apigee-setup.sh
```

2) When you are prompted to choose Apigee profile, choose option “ps” for the Postgres Server installation.
3) Enter the Management Server’s IP address or DNS name used in the Management Server installation above.

   Caution: Do not use the host name mapping to 127.0.0.1. It may cause some problems when they attempt to resolve the host name of the machine.
4) Specify the Apigee Edge email address for the global system administrator.
5) Specify the Apigee admin password.

   Note: This is the global system administrator credential that you have set in the management server installation.
6) Enter the IP addresses or DNS names of ZooKeeper. Ensure that you enter the IP addresses or DNS names separated by spaces.

On successful configuration, it registers the PostgreSQL database and Apigee Postgres Server with the Management Server. A successful configuration returns the 20X HTTP response. This completes the Postgres installation.

Note: Make a note of the server’s IP address to use when setting up master-standby replication for Postgres.

Set up Master-Standby Replication for Postgres

You can now set up a master-standby replication between Postgres servers. Use the same procedure as described in the Five-host Clustered Installation section: Set up Master-Standby Replication for Postgres.

Sanity Validation

Follow the same steps as described in the Standalone Installation section: Sanity Validation.

Install SmartDocs: Machine 1

Follow the same steps as described in the Five-host Clustered Installation section: Install SmartDocs: Machine 1.

Onboarding

Follow the same steps as described in the Standalone Installation section: Onboarding.

Note:

- Organization names must use all lowercase letters and cannot contain spaces, underscores, or periods.
- Ensure that you execute the onboarding steps on the **Machine 1** where Management Server is installed.
- During onboarding, when you are prompted to choose multiple Message Processors, select “y” to associate all Message Processors. Otherwise, press “n” to proceed with single Message Processor.
- During onboarding, when enabling analytics for prod/test, you need to update master or standby status of PostgreSQL database (in two database setup) when prompted. Or you need to manually run `enable-ax.sh` script and update the same.

Verification

Follow the same steps as described in the Standalone Installation section: Verification.

Log into Edge UI

Follow the same steps as described in the Standalone Installation section: Log into Edge UI.

Administrating Edge

For information on how to perform system administration tasks, see the *Apigee Edge Operations Guide,* available on the Apigee ftp site: ftp://ftp.apigee.com/. That document contains procedures for:

- Creating new users
- Creating new organizations, environment, and virtual hosts
- Modifying system passwords
- Monitoring the system
- Configuring SSL
- and information on performing many other system administration tasks

13-host Clustered 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.

The 13-host clustered installation consists of following basic steps:

1) Install the ZooKeeper and Cassandra clusters on the host 1, 2 and 3.
2) Install the OpenLDAP (with master-master OpenLDAP replication) on the host 4 and 5.
3) Install the Apigee Management Server on the host 6 and 7, which also installs the Edge UI.
4) Install Postgres (with master-standby Postgres replication) on the host 8 and 9.
5) Install the Apigee Message Processor + Router on the host 10 and 11.
6) Install Qpid on the host 12 and 13.
7) Install SmartDocs.

These steps are detailed in the following sections.

Install Datastore Cluster Node: Machine 1, 2 and 3

Follow the same steps as described in the Five-host Clustered Installation section: Install Datastore Cluster Node.

Install OpenLDAP Node: Machine 4 and 5

Installation

1) Unzip the application distribution file from your download directory, and change directory (cd) into `apigee-edge-4.15.07.00`.

   ```
   unzip /download_dir/apigee-edge-4.15.07.00.zip
   ```

   ```
   cd apigee-edge-4.15.07.00
   ```

   Note: Unzip the file on the Linux machine itself. Do not unzip it on another machine, such as a Windows machine, and then copy the unzipped files to the Linux machine.

2) At the command prompt, run the `apigee-install` script:

   ```
   ./apigee-install.sh
   ```

3) Enter the JDK 1.7 path to verify the correct Java version.
4) Specify the installation root (<inst-root>) under which all software and configuration files will be stored. Specify the default (/opt) unless you have a specific requirement.
5) Specify the data root (<data-root>) under which all runtime data will be stored. Specify the default (/opt) unless you have a specific requirement.

With the above steps, Apigee Edge installation and configuration files are stored under installation root (`<inst-root>`) and runtime data (`<data-root>`) is stored under data root. Now you need to configure the system.

Configuration

Note: Make a note of the server's IP address to use when setting up master-master LDAP replication on Management Server.

1) At the command prompt, run the `apigee-setup` script.

   ```
   /<inst-root>/apigee4/share/installer/apigee-setup.sh
   ```

2) When you are prompted to choose Apigee profile, choose option "ld" for the LDAP node installation.
3) Specify the Apigee Edge email address for the global system administrator.
4) When prompted, choose the LDAP server type as option 2, OpenLDAP with replication.
5) Specify the new root password for LDAP. Retype the password, and then wait as the system installs the components.

   Note: Make a note of the LDAP root password to use when setting up master-master LDAP replication on Management Server.

   Note: Ensure that you specify the identical LDAP password for both OpenLDAP servers. This is required to achieve successful OpenLDAP replication. Also, it is a good practice to follow similar approach to change both OpenLDAP passwords.
6) For Machine 4, specify the ID as the first server.

   For Machine 5, specify the ID as the second server.
7) For Machine 4, enter the IP address of Machine 5 as the peer OpenLDAP server.

   For Machine 5, enter the IP address of Machine 4 as the peer OpenLDAP server.

This confirms that LDAP installation and configuration are successfully completed.

Install Apigee Management Server: Machine 6 and 7

Installation

1) Unzip the application distribution file from your download directory, and change directory (`cd`) into `apigee-edge-4.15.07.00.`

   ```
   unzip /download_dir/apigee-edge-4.15.07.00.zip
   ```

   ```
   cd apigee-edge-4.15.07.00
   ```

Note: Unzip the file on the Linux machine itself. Do not unzip it on another machine, such as a Windows machine, and then copy the unzipped files to the Linux machine.

2) At the command prompt, run the `apigee-install` script.

   ```
   ./apigee-install.sh
   ```

3) Enter the JDK 1.7 path to verify the correct Java version.
4) Specify the installation root (<inst-root>) under which all software and configuration files will be stored. Specify the default (/opt) unless you have a specific requirement.
5) Specify the data root (<data-root>) under which all runtime data will be stored. Specify the default (/opt) unless you have a specific requirement.

With the above steps, Apigee Edge installation and configuration files are stored under installation root (<inst-root>) and runtime data (<data-root>) is stored under data root. Now you need to configure the system.

Configuration

Follow the same steps as described in the Five-host Clustered Installation section: Configure Apigee Management Server: Machine 1.

Note: In this install scenario, LDAP has been decoupled from the Management Server and installed as a separate node. That said, in order to set up an OpenLDAP replication, take care of the following additional configuration prompts when configuring Management Server using *`apigee-setup.sh`* script.

On Machine 6:

- When you are prompted to choose “Use existing LDAP host”, press “y” to use the existing LDAP Server.
- Enter the IP address or DNS name of the LDAP Server (Machine 4).
- Enter the port of LDAP Server. The default port is 10389.
- Enter the root password for LDAP that you noted earlier during LDAP Server installation.

On Machine 7:

- When you are prompted to choose “Use existing LDAP host”, press “y” to use the existing LDAP Server.
- Enter the IP address or DNS name of the LDAP Server (Machine 5).
- Enter the port of LDAP Server. The default port is 10389.
- Enter the root password for LDAP that you noted earlier during LDAP Server installation.

Install Analytics Postgres Server: Machine 8 and 9

Follow the same steps as described in the Nine-host Installation section: Install Analytics Postgres Server. When prompted to enter the IP address of the Management Server, enter the IP address of the first Management Server, the one you installed on Machine 6.

Note: Make a note of the server's IP address to use when setting up master-standby replication for Postgres.

Set up Master-Standby Replication for Postgres

You can now set up a master-standby replication between Postgres servers. Use the same procedure as described in the Five-host Clustered Installation section: Set up Master-Standby Replication for Postgres.

Install Apigee Router and Message Processor: Machine 10 and 11

Follow the same steps as described in the Five-host Clustered Installation section: Install Apigee Router and Message Processor.

Optionally set Cassandra Authentication

Follow the same steps as described in the Five-host Clustered Installation section: Optionally set Cassandra Authentication on Machine 1, 2, and 3.

Testing

Follow the same steps as described in the Five-host Clustered Installation section: Testing.

Install Apigee Analytics Qpid Server: Machine 12 and 13

Follow the same steps as described in the Nine-host Installation section: Install Analytics Qpid Server. When prompted to enter the IP address of the Management Server, enter the IP address of the first Management Server, the one you installed on Machine 6.

Sanity Validation

Follow the same steps as described in the Standalone Installation section: Sanity Validation.

Install SmartDocs: Machines 6

Follow the same steps as described in the Five-host Clustered Installation section: Install SmartDocs: Machine 1 to install SmartDocs on Machine 6.

Onboarding

Follow the same steps as described in the Standalone Installation section: Onboarding.

Note:

- Organization names must use all lowercase letters and cannot contain spaces, underscores, or periods.
- Ensure that you execute the onboarding steps on the **Machine 6 or 7** where Management Server is installed.
- During onboarding, when you are prompted to choose multiple Message Processors, select "y" to associate all Message Processors. Otherwise, press "n" to proceed with single Message Processor.
- During onboarding, when enabling analytics for prod/test, you need to update master or standby status of PostgreSQL database (in two database setup) when prompted. Or you need to manually run `enable-ax.sh` script and update the same.

Verification

Follow the same steps as described in the Standalone Installation section: Verification.

Log into Edge UI

Follow the same steps as described in the Standalone Installation section: Log into Edge UI.

Administrating Edge

For information on how to perform system administration tasks, see the *Apigee Edge Operations Guide,* available on the Apigee ftp site: ftp://ftp.apigee.com/. That document contains procedures for:

- Creating new users
- Creating new organizations, environment, and virtual hosts
- Modifying system passwords
- Monitoring the system
- Configuring SSL
- and information on performing many other system administration tasks

12-host Clustered Installation (Minimum API traffic DR / AX HA)

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.

The 12-host clustered installation (two data center setup) consists of following basic steps. Note that the order of installation is very important to achieve disaster recovery and high availability setup across two datacenters using API DN support.

1) Install the ZooKeeper clusters on the host 1, 2, 3, 7, 8 and 9.
2) Install the Cassandra clusters on the host 1, 2, 3, 7, 8 and 9.
3) Install the Apigee Management Server (with OpenLDAP replication) on the host 1, which also installs the Edge UI.
4) Install the Apigee Management Server (with OpenLDAP replication) on the host 7, which also installs the Edge UI.
5) Install the Apigee Message Processor + Router on the host 2, 3, 8 and 9.
6) Install Qpid on the host 4, 5, 10 and 11.
7) Install Postgres on the host 6 and 12.
8) Set up master-standby replication between Postgres on the host 6 and 12.
9) Install SmartDocs.

These steps are detailed in the following sections.

Install Datastore Cluster Node: Machine 1, 2, 3, 7, 8 and 9

Installation

Follow the same steps as described in the Five-host Clustered Installation section: Install Datastore Cluster Node - Installation.

Configuration

1) At the command prompt, run the `apigee-setup` script:

```
/<inst-root>/apigee4/share/installer/apigee-setup.sh
```

2) When you are prompted to choose Apigee profile, choose option “ds” for the ZooKeepeer and Cassandra cluster node installation.
3) When you are prompted to configure ZooKeeper node on this machine, press “y”.

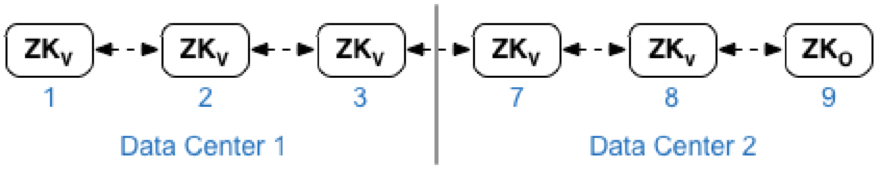
4) Enter ZooKeeper nodes. Ensure that you enter all ZooKeeper IP addresses or DNS names separated by spaces.

   Note: During datastore cluster configuration, `apigee-setup.sh` prompts for all ZooKeeper nodes with ":observer" modifier (if necessary) and all Cassandra nodes with ":dc,ra" modifier (if necessary) of all datacenters.

   Note:

   - The above mentioned IP addresses or DNS names must be listed in the same order on all ZooKeeper nodes in the cluster in order to match the IDs. For example, ZooKeeper ID for this host is 1 and followed by 2 and 3 for the successive hosts.
   - Observers should be added as '<ip>:observer', for example '**192.168.1.42:observer**'.

In this deployment model, add observer to the ZooKeeper host on host 9 as shown below.



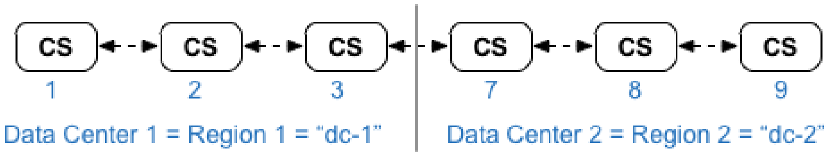
5) When you are prompted to configure Cassandra node on this machine, press "y".

6) Enter Cassandra nodes. Ensure that you enter all Cassandra IP addresses or DNS names separated by spaces.

   Note:

   - The IP addresses or DNS names must be listed in the same order on all Cassandra nodes.
   - All nodes can have a suffix ':<d>,<r>', for example '<ip>:1,2 = datacenter 1 and rack/availability zone 2 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"
   - If the suffix is missing, ':1,1' is assumed.
   - All datacenters need to have the same number of nodes.
   - The first node in rack/availability zone 1 of each datacenter will be used as seed servers.

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


CS
CS
CS
CS
CS
CS
1
2
3
7
8
9
Data Center 1 = Region 1 = "dc-1"
Data Center 2 = Region 2 = "dc-2"

7) Specify the Cassandra cluster name.

The configuration successfully completes the datastore setup in the server across two datacenters.

Configure Apigee Management Server (with OpenLDAP Replication): Machine 1

You now can set up an OpenLDAP replication.

Note: Refer to the Tools section under Prerequisites before installation.

Installation

Follow the same steps as described in the Five-host Clustered Installation section: Configure Apigee Management Server: Machine 1.

Note:

- When you are prompted to choose “LDAP server”, select “2” to choose the OpenLDAP with replication.
- Enter the server ID of the server. Here it is 1.
- Enter the IP address or DNS name of the peer OpenLDAP server.
- When you see this prompt "**Enter one ZooKeeper ip-address/fqdn or several addresses separated by spaces**", enter only the IP addresses of the ZooKeeper nodes in DataCenter 1.
- When you see this prompt "**Enter all Cassandra ip-addresses/fqdns with modifiers separated by spaces**", enter all Cassandra nodes in both DataCenters, including the suffix ':<d>,<r>'.
- You are prompted to enter DataCenter name after you specify the Message Processor and Router POD. In this case it is **dc-1**.

Configure Apigee Management Server (with OpenLDAP Replication): Machine 7

Follow the same steps as described above for Management Server installation on Machine 1.

Take care of the following during configuration:

- Select “2” to choose the OpenLDAP with replication.

  Note: Ensure that you specify the identical LDAP password for both OpenLDAP servers. This is required to achieve successful OpenLDAP replication. Also, it is a good practice to follow similar approach to change both OpenLDAP passwords.
- Enter the server ID of the server. Here it is 2.
- Enter the IP address or DNS name of the peer OpenLDAP server.

- When you see this prompt "**Enter one ZooKeeper ip-address/fqdn or several addresses separated by spaces**", enter only the IP addresses of the ZooKeeper nodes in DataCenter 2.
- When you see this prompt "**Enter all Cassandra ip-addresses/fqdns with modifiers separated by spaces**", enter all Cassandra nodes in both DataCenters, including the suffix ':<d>,<r>'.

Enter datacenter name as **dc-2** after you specify the Message Processor and Router POD.

Install Apigee Router and Message Processor: Machine 2, 3, 8 and 9

Follow the same steps as described in the Five-host Clustered Installation section: Install Apigee Router and Message Processor.

Note:

- You are prompted to enter the Management Server IP address. Specify the IP address for the Management Server for **dc-1** for Node 2 and 3, and the IP address for the Management Server for **dc-2** for Node 8 and Node 9.
- You are prompted to enter datacenter name. Specify **dc-1** for Node 2 and 3, and **dc-2** for Node 8 and 9.

Optionally set Cassandra Authentication

Follow the same steps as described in the Five-host Clustered Installation section: Optionally set Cassandra Authentication on Machine 1, 2, and 3.

Testing

Follow the same steps as described in the Five-host Clustered Installation section: Testing.

Install Analytics Qpid Server: Machine 4, 5, 10 and 11

Follow the same steps as described in the Nine-host Installation section: Install Analytics Qpid Server.

Note:

- For root installation, this step installs the prerequisites for Qpid and Postgres installation via global yum. In order to allow non-root installation, Analytics Servers (Qpid, Postgres) have some prerequisites that must be installed prior to installation. For more information, see "Analytics dependency files" under the Installation Checklist section.
- You are prompted to enter the Management Server IP address. Specify the IP address for the Management Server for **dc-1** for Node 4 and 5, and the IP address for the Management Server for **dc-2** for Node 10 and Node 11.

- When prompted to enter datacenter name, specify **dc-1** for Node 4 and 5, and **dc-2** for Node 10 and 11.

Install Analytics Postgres Server: Machine 6 and 12

Note: For root installation, this step installs the prerequisites for Qpid and Postgres installation via global yum. In order to allow non-root installation, Analytics Servers (Qpid, Postgres) have some prerequisites that must be installed prior to installation. For more information, see "Analytics dependency files" under the Installation Checklist section.

Follow the same steps as described in the Nine-host Installation section: Install Analytics Postgres Server.

Note:

- You are prompted to enter the Management Server IP address. Specify the IP address for the Management Server for **dc-1** for Node 6, and the IP address for the Management Server for **dc-2** for Node 12.
- When prompted to enter datacenter name, specify **dc-1** for Node 6 and **dc-2** for Node 12.
- Make note of the server’s IP address to use when setting up master-standby replication for Postgres.

Set up Master-Standby Replication between Postgres: Machine 6 and 12

You can now set up a master-standby replication between Postgres servers. It is recommended to set up a trust relationship between Postgres servers prior to the replication. The section below describes about setting up trust relationship.

Enabling Password-Less Login (SSH Key) between Master and Standby Postgres Machine

Follow the same steps as described in the Five-host Clustered Installation section: Enabling Password-Less Login (SSH Key) between Master and Standby Postgres Machine.

Replicating Master-Standby Postgres

In this scenario, Machine 6 acts as a master (primary Postgres server) and Machine 12 as a standby (secondary Postgres server).

Note: Refer to the Tools section under Prerequisites before installation.

Perform the following steps as mentioned below:

1) On Machine 12, stop the PostgreSQL database using the command:

```
/<inst-root>/apigee4/bin/apigee-service postgresql stop
```

2) On Machine 6, run the command:

```
/<inst-root>/apigee4/share/installer/apigee-postgres-replication-setup.sh
```

   a) When you are prompted to choose Postgres type, choose option “m” for the master.

   b) Enter the IP address or DNS name of the standby server.

   Note: If you use a hostname in place of an IP address, ensure that this hostname is resolvable with "`dig`".

   c) Enter data directory path of PostgreSQL database peer node.

   d) Enter the path of the private SSH key file for accessing ‘standby’. If you used the procedure above to configure password-less login from master to standby, the path is `/home/apigee/.ssh/id_rsa`.

3) On Machine 12, repeat the step 2. Note that here you choose option “s” for the standby and enter the IP address or DNS name of the master server. The script also restarts PostgreSQL database.

4) 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.

   c) On the master node, run:

```
/<inst-root>/apigee4/bin/postgres-check-master.sh
```

   d) On the standby node:

```
/<inst-root>/apigee4/bin/postgres-check-standby.sh
```

Sanity Validation

Follow the same steps as described in the Standalone Installation section: Sanity Validation.

Install SmartDocs on Machine 1

Follow the same steps as described in the Five-host Clustered Installation section: Install SmartDocs: Machine 1 to install SmartDocs on Machine 1.

Onboarding

Follow the same steps as described in the Standalone Installation section: Onboarding.

Note:

- Organization names must use all lowercase letters and cannot contain spaces, underscores, or periods.

- Ensure that you execute the onboarding steps on the **Machine 1** where Management Server is installed.
- During onboarding, when you are prompted to choose multiple Message Processors, select “y” to associate all Message Processors. Otherwise, press “n” to proceed with single Message Processor
- During onboarding, when enabling analytics for prod/test, you need to update master or standby status of PostgreSQL database (in two database setup) when prompted. Or you need to manually run `enable-ax.sh` script and update the same

Verification

Follow the same steps as described in the Standalone Installation section: Verification.

Log into Edge UI

Follow the same steps as described in the Standalone Installation section: Log into Edge UI.

Administrating Edge

For information on how to perform system administration tasks, see the *Apigee Edge Operations Guide,* available on the Apigee ftp site: ftp://ftp.apigee.com/. That document contains procedures for:

- Creating new users
- Creating new organizations, environment, and virtual hosts
- Modifying system passwords
- Monitoring the system
- Configuring SSL
- and information on performing many other system administration tasks

8-host clustered installation (Min HA with API Backend)

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.

The eight-host clustered installation consists of following basic steps:

1) Follow the same steps as described in five-host clustered installation on first five hosts.
2) Install the API Backend on next three hosts.
   - Install the API Backend Stack on the sixth and seventh host. Choose option “abs” for the stack installation.
   - Install the API Backend Portal on the eighth host. Choose option “abp” for the Portal installation.

The API Backend install steps are detailed in the following sections.

Install API Backend Stack: Machine 6 and 7

Installation

1) Unzip the application distribution file from your download directory, and change directory (`cd`) into `apigee-edge-4.15.07.00`

   ```
   unzip /download_dir/apigee-edge-4.15.07.00.zip
   ```

   ```
   cd apigee-edge-4.15.07.00
   ```

   Note: Unzip the file on the Linux machine itself. Do not unzip it on another machine, such as a Windows machine, and then copy the unzipped files to the Linux machine.

2) At the command prompt, run the `apigee-install` script.

   ```
   ./apigee-install.sh
   ```

3) Enter the JDK 1.7 path to verify the correct Java version.
4) Specify the installation root (<inst-root>) under which all software and configuration files will be stored. Specify the default (/opt) unless you have a specific requirement.
5) Specify the data root (<data-root>) under which all runtime data will be stored. Specify the default (/opt) unless you have a specific requirement.
6) Note the IP address of the server. You need the IP address when installing the BaaS Portal.

With the above steps, Apigee Edge installation and configuration files are stored under installation root (<inst-root>) and runtime data (<data-root>) is stored under data root. Now you need to configure the system.

Configuration

1) At the command prompt, run the `apigee-setup.sh` script:

   ```
   /<inst-root>/apigee4/share/installer/apigee-setup.sh
   ```

2) When you are prompted to choose Apigee profile, choose option "abs" for the API Backend Stack installation.

   Once it fetches the correct IP address, it starts ZooKeeper client and then finishes the configuration. Next step is API Backend Stack setup and configuration.

3) Enter the external IP address/host name of API Backend Stack.
4) Specify the API Backend administrator name.
5) Specify the email address of the API Backend administrator.
6) Specify the password for the API Backend administrator. Retype the password, and then wait as the system installs the components.

   Use this administrator username and password to log in to the API Backend Portal.

7) Specify the Cassandra cluster name for the Edge installation. Proceed with the default, `Apigee`, unless you have specified a different name for the Cassandra cluster when you installed Edge.
8) Enter the IP addresses of the Cassandra nodes from when you installed Edge. Ensure that you enter 3 or more Cassandra IP addresses or DNS names separated by comma (no space after comma).
9) Specify the SMTP host address and port. You must set an SMTP server before you can install the API Backend Portal.
10) Specify the email address for the SMTP user.
11) Specify the password for the SMTP user.

After it fetches the correct admin credentials, the configuration installs Tomcat and successfully creates API Backend keyspaces and sets up API Backend Stack on the server. SMTP is also configured and this will allow the UI to send password confirmation mails.

Install API Backend Portal: Machine 8

Installation

1) Unzip the application distribution file from your download directory, and change directory (`cd`) into `apigee-edge-4.15.07.00`

   ```
   unzip /download_dir/apigee-edge-4.15.07.00.zip
   cd apigee-edge-4.15.07.00
   ```

Note: Unzip the file on the Linux machine itself. Do not unzip it on another machine, such as a Windows machine, and then copy the unzipped files to the Linux machine.

2) At the command prompt, run the `apigee-install.sh` script:

   ```
   ./apigee-install.sh
   ```

3) Enter the JDK 1.7 path to verify the correct Java version.

4) Specify the installation root (`<inst-root>`) under which all software and configuration files will be stored. Specify the default (`/opt`) unless you have a specific requirement.

5) Specify the data root (`<data-root>`) under which all runtime data will be stored. Specify the default (`/opt`) unless you have a specific requirement.

With the above steps, Apigee Edge installation and configuration files are stored under installation root (`<inst-root>`) and runtime data (`<data-root>`) is stored under data root. Now you need to configure the system.

Configuration

1) At the command prompt, run the `apigee-setup` script.

   ```
   /<inst-root>/apigee4/share/installer/apigee-setup.sh
   ```

2) When you are prompted to choose Apigee profile, choose option “abp” for the API Backend Portal installation.

3) Enter API Backend Stack server as `host:port` for setting up load balancer. The load balancer is used to distribute requests across the available API Backend Stacks.

   Enter the external IP address/host name of all API Backend Stacks you’re deploying. By default, the port number for the API Backend Stack server is 8080. Once you’re connected to all Stacks, on the next prompt press enter to exit or move to next step.

4) Enter the port load balancer listens to on the Portal server. Proceed with the default value of 8080. If this port is not available, choose a different port, such as 8081.

5) Enter the external IP address/host name of API Backend Portal.

After it fetches the correct IP addresses, it starts the “nginx” (load balancer) and then finishes the API Backend Portal configuration.

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

This completes the API Backend Portal and its associated components setup.

Onboarding the API Backend Portal

You can access the API Backend Portal user interface using a web browser. Remember that you already noted the API Backend Portal URL at the end of the installation.

To access the portal, enter the API Backend 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.

The login screen appears. From the login screen, you can either:

1. **Log in** to the portal by using the admin username and password that you set when you installed the API Backend Stack.

   After you log in, you will see an organizations, TEST-ORGANIZATION, has already been created. As an admin, you can create additional organizations and users.

2. **Register** a new organization and user.

   Click the **Register** button on the log in screen. You are prompted to create a new organization and organization user. Note that you cannot create an organization by using the admin credentials but must specify credentials for a new user and organization.

Access the API BaaS REST API

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

```
https://{portalExternalIP}:8080/AppServices/your-org/your-app
```

For example, use the following cURL command to view the status of the API Backend Stack:

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

For more information on getting started with API Backend Portal, see the Apigee documentation at: 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).

12-host clustered installation (Performance HA with API Backend)

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.

The 12-host clustered installation consists of following basic steps:

1) Follow the same steps as described in nine-host clustered installation on first nine hosts.
2) Install the API Backend on next three hosts.
    - Install the API Backend Stack on the 10th and 11th host. Choose option “abs” for the Stack installation.
    - Install the API Backend Portal on the 12th host. Choose option “abp” for the Portal installation.

The API Backend install steps are detailed in the following sections.

Install API Backend Stack: Machine 10 and 11

Follow the steps as described in eight-host clustered installation section: Install API Backend Stack: Machine 6 and 7.

Install API Backend Portal: Machine 12

Follow the steps as described in eight-host clustered installation section: Install API Backend Portal: Machine 8.

Onboarding the API Backend Portal

You can access the API Backend Portal user interface using a web browser. Remember that you already noted the API Backend Portal URL at the end of the installation.

To access the portal, enter the API Backend 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.

The login screen appears. From the login screen, you can either:

1. **Log in** to the portal by using the admin username and password that you set when you installed the API Backend Stack.

   After you log in, you will see an organizations, TEST-ORGANIZATION, has already been created. As an admin, you can create additional organizations and users.

2. **Register** a new organization and user.

   Click the **Register** button on the log in screen. You are prompted to create a new organization and organization user. Note that you cannot create an organization by using the admin credentials but must specify credentials for a new user and organization.

Access the API BaaS REST API

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

```
https://{portalExternalIP}:8080/AppServices/your-org/your-app
```

For example, use the following cURL command to view the status of the API Backend Stack:

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

For more information on getting started with API Backend Portal, see the Apigee documentation at: 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).

3-host Installation with 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.

The initial three-host installation consists of three basic steps:

1) Install the Apigee Management Server on one host. Determine and note the IP address or DNS name of the Management Server. Choose profile “ms” to install Management Server.
2) Install the Apigee Router + Message Processor on the second host; provide the IP address or DNS name of the Management Server when it is needed. Choose profile “rmp” to install Router and Message Processor.
3) Install the Apigee Analytics standalone on the third host; provide the IP address or DNS name of the Management Server when it is needed. Choose profile “sax” to install Analytics Servers.

Note: Monetization requires that you have configured an SMTP server on the Management Server. If you have configured an SMTP server, see the *Apigee Edge Operations Guide,* available on the Apigee ftp site: ftp://ftp.apigee.com/.

To install Monetization Services on any multi-server setup of an Apigee Edge for Private Cloud, run *`apigee-setup.sh`* with “mo” profile in the following order: first on Postgres server, then on Management Server and finally on Message Processor.

Note: The order of installation is very important to successfully integrate Monetization Services in an existing setup.

The following steps illustrate how to add Monetization Services on an existing three-host Apigee Edge setup:

1) Update the Apigee Analytics (Postgres/Qpid Server) to configure the Monetization datastore. The queue configuration is changed to receive the same message by Analytics as well as by Monetization Services.
2) Update the Apigee Management Server to enable the management components of the Monetization Services, for example, catalog management, limits and notifications configuration, billing and reporting.
3) Update the Apigee Message Processor and Router to enable the runtime components of the Monetization Services, for example, transaction recording policy and limit enforcement.
4) Perform the onboarding process with additional Monetization steps.
5) Configure the Developer Services portal to support monetization. For more information, see http://apigee.com/docs/monetization/content/configure-monetization-developer-portal

(or http://apigee.com/docs/content/documentation-archives to find the docs that correspond to earlier versions of the product).

Note: Make a note of the server's IP addresses of the existing setup (all nodes) to use when you run the script, *apigee-setup.sh* for the Monetization Services.

These steps are detailed in the following sections.

Integrate Monetization Services with Postgres

Monetization Services use RDBMS to store management, configuration and transaction data. Analytics and Monetization database schemas are stored separately. You can either setup a brand new Postgres or create a separate schema in the Analytics database.

Installation

It is assumed that you have purchased the Monetization Services and in that case you must have received the Monetization RPM as a part of product distribution file. Hence no separate installation is required as it has already been completed in the existing three-host setup. Now you need to configure the system.

Configuration

1) At the command prompt, run the `apigee-setup.sh` script:

```
sudo /<inst-root>/apigee4/share/installer/apigee-setup.sh
```

2) When you are prompted to choose Apigee profile, choose option “mo” for the Monetization Service installation.

3) Specify the Apigee admin password for the global system administrator.

   Note: This is the global system administrator credential that you have set in the Management Server installation.

4) Enter the IP address or DNS name of the Postgres server.

5) Enter a Postgres username and password. The username 'postgres' is reserved - do not use it as the username.

6) Specify the SMTP host, port, whether to enable SSL, and the SMTP user information.

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.

Note:

- If Qpid Server is installed on a different node, ensure that you update Qpid Server with *apigee.setup.sh* script to configure the Monetization datastore.

- Monetization Billing Documents are stored as byte arrays in PostgreSQL database. If you want to use Monetization Billing Documents, ensure that you set *`bytea_output = 'escape'`* in `postgresql.conf` and then restart the server.

Integrate Monetization Services with Management Server

Installation

It is assumed that you have purchased the Monetization Services and in that case you must have received the Monetization RPM as a part of product distribution file. Hence no separate installation is required as it has already been completed in the existing three-host setup. Now you need to configure the system.

Configuration

1) At the command prompt, run the `apigee-setup.sh` script:

   ```
   sudo /<inst-root>/apigee4/share/installer/apigee-setup.sh
   ```

2) When you are prompted to choose Apigee profile, choose option “mo” for the Monetization Service installation.
3) Specify the Apigee admin password for the global system administrator.

   Note: This is the global system administrator credential that you have set in the Management Server installation.

4) Enter the IP address or DNS name of the Postgres server.
5) Enter the Postgres username and password that you specified above when you integrated Monetization with the Postgres server.
6) Specify the SMTP host, port, whether to enable SSL, and the SMTP user information.
7) After the installation script completes, set the locale.

   The shell locale on the Management Server must be set to a locale with UTF-8 support. Make sure to set the `LC_TYPE` and `LC_ALL` environment variables to your specific UTF-8 locale, as shown below:

   ```
   > export LC_CTYPE="en_US.UTF-8"
   > export LC_ALL="en_US.UTF-8"
   ```

   You can see the current value of these environment variables by using the following command:

   ```
   > locale
   ```

   If you update these environment variables, restart the Management Server:

```
/<inst-root>/apigee4/etc/init.d/apigee-management-server restart
```

On successful configuration, the Management Server is updated with Monetization Services. This completes the integration of Monetization Services and its associated components with Management Server.

Integrate Monetization Services with Message Processor and Router

Installation

It is assumed that you have purchased the Monetization Services and in that case you must have received the Monetization RPM as a part of product distribution file. Hence no separate installation is required as it has already been completed in the existing three-host setup. Now you need to configure the system.

Configuration

1) At the command prompt, run the `apigee-setup.sh` script:

   ```
   sudo /<inst-root>/apigee4/share/installer/apigee-setup.sh
   ```

2) When you are prompted to choose Apigee profile, choose option “mo” for the Monetization Services installation.

3) Specify the Apigee admin password for the global system administrator.

   Note: This is the global system administrator credential that you have set in the Management Server installation.

4) Specify the SMTP host, port, whether to enable SSL, and the SMTP user information.

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

Onboarding

Follow the same steps as described in the Standalone installation section: Onboarding.

Note:

- Organization names must use all lowercase letters and cannot contain spaces, underscores, or periods.
- Ensure that you execute the onboarding steps on the machine where Management Server is installed.
- Please follow Monetization specific onboarding instructions from the below section.

Additional Onboarding for Monetization

Create/merge 4G entities in Monetization database schema

Run the `mo_onboarding.sh` script on the Management Server to onboard an organization in Monetization database. This script can be executed while creating a new organization or while enabling monetization for an existing organization.

```
/<inst-root>/apigee4/share/installer/monetization/onboarding/mo_onboarding.sh
```

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.

Upload Notification Templates

For each onboarded organization, perform the following:

1) Run the `mint-onboard-org-notification-settings.sh` script to configure notification templates.

```
/<inst-root>/apigee4/share/installer/monetization/onboarding/mint-onboard-org-no
tification-settings.sh
```

2) Specify the global system administrator password.
3) Specify the organization name/ID. **Organization names must use all lowercase letters and cannot contain spaces, underscores, or periods**.
4) Enter the email address of the organization administrator.

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 (or http://apigee.com/docs/content/documentation-archives to find the docs that correspond to earlier versions of the product).

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.

Log into Edge UI

Follow the same steps as described in the Standalone Installation section: Log into Edge UI.

Configure the Developer Services portal

Configure the Developer Services portal to support monetization. For more information, see http://apigee.com/docs/monetization/content/configure-monetization-developer-portal (or http://apigee.com/docs/content/documentation-archives to find the docs that correspond to earlier versions of the product).

Upgrading Apigee Edge

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

Which Edge versions can you upgrade to 4.15.07.00

Depending on your current version of Edge, you can either:

- **Directly** upgrade to 4.15.07.00
- **Incrementally** upgrade, meaning you have to upgrade from your current version to another version of Edge, and then upgrade to 4.15.07.00.

For more information, see Which Edge for Private Cloud versions can you upgrade to 4.15.07.00.

Who can perform the upgrade

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

Required upgrade to Java JDK Version 7

This release of Edge requires that you have installed Java JDK version 7 on all Edge processing nodes. You can install the Oracle JDK 7 or OpenJDK 7.

⚠ **Warning**: This release of Edge does not support JDK 6. If you are currently using JDK 6, you must upgrade to JDK 7. If you rollback the Edge 4.15.07.00 installation, you can optionally reconfigure Edge to use Java JDK 6.

Required upgrade to Cassandra 2.0.15

As part up upgrading to Edge 4.15.07, the upgrade script installs Cassandra 2.0.15, replacing Cassandra 1.2.x.

⚠ **Warning**: If you upgrade to Cassandra 2.0.15, and then later decide to roll back your Edge 4.15.07.00 installation, then the rollback script does not restore your previous version of Cassandra. That means once you upgrade to Cassandra 2.0.15, you cannot rollback to the previous version. Therefore, if there is a chance that you will have to perform a rollback and want to go back to using Cassandra 1.2.x as part of that rollback, then you should install 4.15.07.00 in a testing environment and not in a production environment.

Required ApacheDS upgrade to OpenLDAP

OpenLDAP is now the only LDAP server supported by Edge for new and upgrade installations. Existing installations of Edge that upgrade to 4.15.07.00 must use OpenLDAP. If you are currently using ApacheDS, then the upgrade script installs OpenLDAP.

Warning: If you upgrade to OpenLDAP from ApacheDS, and then later decide to roll back your Edge 4.15.07.00 installation, then the rollback script does not restore ApacheDS. That means once you upgrade to OpenLDAP, you cannot rollback to ApacheDS. Therefore, if there is a chance that you will have to perform a rollback and want to go back to using ApacheDS as part of that rollback, then you should install 4.15.07.00 in a testing environment and not in a production environment.

Disk space requirements for upgrade

Ensure that you have at least 5 GBytes of free disk space before you perform the upgrade. This requirement is for the disk on which you unzip the Edge installation file, and for the disk where you upgrade Edge.

SSL changes on the Edge management UI when upgrading

If you are upgrading an installation prior to 4.14.07.x to 4.15.07.00, and you have enabled SSL on the Edge management UI, then the upgrade script performs the following action to configure SSL:

1. Reads the current SSL settings from the `/<inst-root>/apigee4/share/ui/conf/application.conf` file.
2. Creates a new file named `https-key.conf` in the `/<inst-root>/apigee4/share/ui/conf` directory on the Management Server node and copies the following properties from `application.conf`:

   ```
   https.keyStoreType=JKS
   https.keyStore=conf/keyStore.jks
   play.http.sslengineprovider=service.CustomSSLEngineProvider
   https.keyStorePasswordEncrypted=yourEncryptedKeystorePassword
   ```

3. Modifies `/<inst-root>/apigee4/bin/apigee-env.sh` to set the `UI_PORT` and `UI_HTTPS_PORT` properties:

   ```
   UI_PORT=p#
   UI_HTTPS_PORT=p#
   ```

   The values of these properties are read from the `http.port` and `https.port` properties in `application.conf`.

For more information on configuring SSL on Edge, see SSL in the Apigee online documentation.

Changes to default property values

Based on an analysis of cloud and on-premises deployments in releases prior to 4.15.04, this release of Edge changes the default values of properties in `http.properties` and `router.properties`. The modified properties and new default values are:

- In `apigee4/conf/apigee/message-processor/http.properties`
  - `HTTPTransport.io.timeout.millis=55000`
  - `HTTPClient.connect.timeout.millis=3000`
- In `apigee4/conf/apigee/router/router.properties`, `apigee4/conf/apigee/management-server/router.properties`, and `apigee4/conf/apigee/message-processor/router.properties`
  - `Client.pool.iotimeout=57000`
  - `ServerContainer.io.timeout.millis=58000`

The upgrade script does not modify these properties from their current values. Therefore, if these properties have the old default values or if you changed them from the old default values, the upgrade script retains their current values. After the upgrade completes, you can change them to the new default values, if applicable.

Upgrading when hashing OAuth tokens

To protect OAuth access and refresh tokens in the event of a database security breach, you can enable automatic token hashing in your Edge organization. When the feature is enabled, Edge automatically creates a hashed version of newly generated OAuth access and refresh. The un-hashed tokens are used in API calls, and Edge validates them against the hashed versions in the database.

In previous releases of Edge, you set the `hash.oauth.tokens.enabled` property in the `keymanagement.properites` file to `true` to enable hashing of OAuth tokens. That property has been deprecated in this release.

If the upgrade script detects that the `hash.oauth.tokens.enabled` property is set to `true`, then it issues the following message:

```
To continue using hashing for access token and refresh token, org level
properties must be set
```

```
Instructions to set the org level properties can be found in the install-config-
guide
```

After the upgrade completes, you must set the following organization-level properties to control OAuth token hashing:

```
features.isOAuthTokenHashingEnabled = true
```

```
features.OAuthTokenHashingAlgorithm = SHA1 | SHA256 | SHA384 | SHA512 | PLAIN
```

The first property enables hashing, and the second specifies the hashing algorithm.

Also, if you have existing hashed tokens and want to retain them until they expire, set the following properties in your organization, where the hashing algorithm matches the existing algorithm (for example, SHA1, the former Edge default). If the tokens were un-hashed, use PLAIN.

```
features.isOAuthTokenFallbackHashingEnabled = true
features.OAuthTokenFallbackHashingAlgorithm = SHA1 | SHA256 | SHA384 | SHA512 |
PLAIN
```

Use the following API call with system administrator credentials to set these properties:

Note: When you update the organization properties with the API call, be sure to include all the existing organization properties in the payload. If you don't, all existing organization properties are overwritten by the properties you set with this call. To see the current list of organization properties, use the Get Organization API call.

```
curl -u email:password -X PUT -H "Content-type:application/xml"
https://<ms-ip>:8080/v1/o/{myorg} -d \
"<Organization type="trial" name="MyOrganization">
    <Properties>
        <Property name="features.isOAuthTokenHashingEnabled">true</Property>
        <Property name="features.OAuthTokenHashingAlgorithm">SHA256</Property>
        <Property name="features.isOAuthTokenFallbackHashingEnabled">true</Property>
        <Property name="features.OAuthTokenFallbackHashingAlgorithm">SHA1</Property>
        <Property...(an existing property)
        <Property...(an existing property)
        <Property...(an existing property)
    </Properties>
</Organization>"
```

If you currently have unhashed tokens and want to hash then, Edge provides a script you can run. For more information, see the `apigee-edge-4.15.07.00/bin/Hash-AccessToken/Hash-AccessToken-0.0.1` directory under the directory where you unzipped the `apigee-edge-4.15.07.00.zip` file.

Order of machine upgrades

The order that you upgrade the machines in an Edge installation is important. The most important considerations to an upgrade are:

- You must upgrade **all** Cassandra and Postgres nodes before you upgrade any other nodes.
- You must upgrade **all** Qpid and Postgres nodes before you upgrade any Router and Message Processor nodes.
- If you have installed Monetization, you must upgrade **all** Qpid and Postgres nodes before you upgrade any Monetization nodes.
- If a step specifies that it should be performed on multiple machines, perform it in the specified machine order.
- You can determine the machine numbers from the installation procedure for your Edge configuration.

For a 2-host standalone installation:

1. Upgrade Cassandra and ZooKeeper on machine 1:

   ```
   /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c cs,zk
   ```

2. Upgrade Qpid and Postgres on machine 2:

   ```
   /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c qpid,ps
   ```

3. Upgrade LDAP on machine 1:

   ```
   /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c ldap
   ```

4. Upgrade Edge components on machine 2 and machine 1:

   ```
   /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c edge
   ```

5. Upgrade UI on machine 1:

   ```
   /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c ui
   ```

For a 5-host clustered installation:

1. Upgrade Cassandra and ZooKeeper on machine 1, 2, and 3:

   ```
   /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c cs,zk
   ```

2. Upgrade Qpid and Postgres on machine 4 and 5:

   ```
   /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c qpid,ps
   ```

3. Upgrade LDAP on machine 1:

   ```
   /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c ldap
   ```

4. Upgrade Edge components on machine 4, 5, 1, 2, and 3 in that order:

   ```
   /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c edge
   ```

5. Upgrade UI on machine 1:

   ```
   /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c ui
   ```

For a 9-host clustered installation:

1. Upgrade Cassandra and ZooKeeper on machine 1, 2, and 3:

   ```
   /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c cs,zk
   ```

2. Upgrade Qpid on machine 6 and 7:

   ```
   /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c qpid
   ```

3. Upgrade Postgres on machine 8 and 9:

   ```
   /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c ps
   ```

4. Upgrade LDAP on machine 1:

   ```
   /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c ldap
   ```

5. Upgrade Edge components on machine 6, 7, 8, 9, 1, 4, and 5 in that order:

   ```
   /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c edge
   ```

6. Upgrade UI on machine 1:

   ```
   /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c ui
   ```

For a 13-host clustered installation:

1. Upgrade Cassandra and ZooKeeper on machine 1, 2, and 3:

   ```
   /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c cs,zk
   ```

2. Upgrade Qpid on machine 12 and 13:

   ```
   /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c qpid
   ```

3. Upgrade Postgres on machine 8 and 9:

   ```
   /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c ps
   ```

4. Upgrade LDAP on machine 4 and 5:

   ```
   /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c ldap
   ```

5. Upgrade Edge components on machine 12, 13, 8, 9, 6, 7, 10, 11 in that order:

   ```
   /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c edge
   ```

6. Upgrade UI on machine 6 and 7:

   ```
   /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c ui
   ```

For a 12-host clustered installation:

1. Upgrade Cassandra and ZooKeeper:
   a. On machines 1, 2 and 3 in Data Center 1:

      ```
      /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c cs,zk
      ```

   b. On machines 7, 8, and 9 in Data Center 2

      ```
      /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c cs,zk
      ```

2. Upgrade Qpid:
   a. Machines 4, 5 in Data Center 1

      ```
      /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c qpid
      ```

   b. Machines 10, 11 in Data Center 2

      ```
      /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c qpid
      ```

3. Upgrade Postgres:
   a. Machines 6 in Data Center 1

      ```
      /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c ps
      ```

   b. Machines 12 in Data Center 2

      ```
      /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c ps
      ```

4. Upgrade LDAP:
   a. Machines 1 in Data Center 1

      ```
      /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c ldap
      ```

   b. Machines 7 in Data Center 2

```
/<install-root>/apigee4/share/installer/apigee-upgrade.sh -c ldap
```

5. Upgrade Edge components:
    a. Machines 4, 5, 6, 1, 2, 3 in Data Center 1

    ```
    /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c edge
    ```

    b. Machines 10, 11, 12, 7, 8, 9 in Data Center 2

    ```
    /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c edge
    ```

6. Upgrade UI:
    a. Machine 1 in Data Center 1

    ```
    /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c ui
    ```

    b. Machine 7 in Data Center 2

    ```
    /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c ui
    ```

For a 8-host clustered installation with API Backend:

1. Upgrade Cassandra and ZooKeeper on machine 1, 2, and 3:

    ```
    /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c cs,zk
    ```

2. Upgrade Qpid and Postgres on machine 4 and 5:

    ```
    /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c qpid,ps
    ```

3. Upgrade LDAP on machine 1:

    ```
    /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c ldap
    ```

4. Upgrade Edge components on machine 4, 5, 1, 2, and 3 in that order:

    ```
    /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c edge
    ```

5. Upgrade UI on machine 1:

    ```
    /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c ui
    ```

6. Upgrade API Backend Stack on machines 6 and 7:

    ```
    /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c abs
    ```

7. Upgrade API Backend Portal on machine 8:

    ```
    /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c abp
    ```

For a 12-host clustered installation with API Backend:

1. Upgrade Cassandra and ZooKeeper on machine 1, 2, and 3:

    ```
    /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c cs,zk
    ```

2. Upgrade Qpid on machine 6 and 7:

    ```
    /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c qpid
    ```

3. Upgrade Postgres on machine 8 and 9:

```
/<install-root>/apigee4/share/installer/apigee-upgrade.sh -c ps
```

4. Upgrade LDAP on machine 1:

   ```
   /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c ldap
   ```

5. Upgrade Edge components on machine 6, 7, 8, 9, 1, 4, and 5 in that order:

   ```
   /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c edge
   ```

6. Upgrade UI on machine 1:

   ```
   /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c ui
   ```

7. Upgrade API Backend Stack on machines 10 and 11:

   ```
   /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c abs
   ```

8. Upgrade API Backend Portal on machine 12:

   ```
   /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c abp
   ```

For a 3-host Monetization installation

1. Upgrade Cassandra and ZooKeeper on machine 1:

   ```
   /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c cs,zk
   ```

2. Upgrade Qpid and Postgres on machine 3:

   ```
   /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c qpid,ps
   ```

3. Upgrade LDAP on machine 1:

   ```
   /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c ldap
   ```

4. Upgrade Edge components on machine 3, 1, and 2:

   ```
   /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c edge
   ```

5. Upgrade UI on machine 1:

   ```
   /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c ui
   ```

6. Upgrade Monetization on machine 3, 1, and 2:

   ```
   /<install-root>/apigee4/share/installer/apigee-upgrade.sh -c mo
   ```

For a non-standard installation

If you have a non-standard installation, then upgrade Edge components in the following order:

1. ZooKeeper
2. Cassandra
3. Qpid
4. Postgres
5. LDAP

6. Edge, meaning the edge profile to the `apigee-upgrade.sh` script, on all nodes in the order: Qpid, Postgres, Management Server, Router, Message Processor
7. UI
8. API Backend Stack and Portal
9. Monetization

Disabling the check on upgrade order in the upgrade script

On an upgrade, the upgrade scripts checks to ensure that you have upgraded all Qpid and Postgres nodes before upgrading Router and Message Processor nodes. However, there are two scenarios where that check can fail:

- **Before upgrading a standalone node with no corresponding Qpid and Postgres node**

  The Edge installer lets you install a standalone (sa) profile consisting of the ZooKeeper, Cassandra, LDAP, Management Server, Router, Message Processor, and UI components all installed on a single node.

  You can create a standalone node for testing, but that configuration is not meant to be used for deployment.

- **Before upgrading Router and Message Processor nodes that cannot connect to the Management Server over port 8080**

  You must upgrade all Qpid and Postgres nodes before you upgrade any Router and Message Processor node. When you upgrade a Router or Message Processor, the upgrade script checks with the Management Server that the analytics nodes have already been upgraded. If the analytics nodes have not been upgraded, the script fails with an appropriate error message.

  To perform that check, the upgrade script makes a request to port 8080 of the Management Server node. However, because of firewall or other security reasons, port 8080 on the Management Server might not be accessible from Router and Message Processor nodes. If the Management Server is not accessible, then the upgrade script fails on the Router and Message Processor nodes.

If the upgrade script detects an error in the upgrade order, it outputs a message in the form:

```
Checking to ensure analytics is upgraded to 4.15.07 before upgrading run time
components...
--> curl -X GET http://1.1.1.100:8080/v1/o [...]
--> curl -X GET http://1.1.1.100:8080/v1/o/example/e [...]
--> curl -X GET
http://1.1.1.100:8080/v1/o/example/e/prod/analytics/admin/schema?type=fact [...]
[WARNING]: Could not verify if analytics components were upgraded.
```

```
After Confirming that the analytics components are upgraded, re-run
/apigee4/share/installer/apigee-upgrade.sh
```

If you are performing a silent upgrade, meaning one where you pass an *upgrade-config* file to the upgrade script, then you can bypass the check on the standalone, Router, or Message Processor node by adding the following line to the *upgrade-config* file:

```
SKIP_AX_UPGRADE_CHECK=y
```

Caution: Do not include the `SKIP_AX_UPGRADE_CHECK` flag unless you have already ensured that all analytics nodes have been upgraded. Upgrading Router and Message Processor nodes without first upgrading the analytics nodes can create an unstable Edge installation.

If you instead perform an interactive upgrade, meaning one where you respond to prompts from the upgrade script, then you must create an *upgrade-config* file with just one line for the `SKIP_AX_UPGRADE_CHECK` setting, and then pass that file to the upgrade script:

```
/<inst-root>/apigee4/share/installer/apigee-upgrade.sh -f upgrade-config
```

Zero-downtime upgrade

A zero-downtime upgrade, or rolling upgrade, lets you upgrade your Edge installation without bringing down Edge.

Note: Zero-downtime upgrade 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 upgrade the Router and any other components on the same machine as the Router, and then add the Router back to the load balancer.

1. Upgrade the machines in the correct order for your installation as described above in Order of machine upgrades.
2. When it is time to upgrade the Routers, select any one Router and make in inaccessible, as described below in Making a Router inaccessible and disabling a Message Processor.
3. Upgrade 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 accessible again.
5. Repeat steps 2 through 4 for the remaining Routers.
6. Continue the upgrade for any remaining machines in your installation.

Making a Router inaccessible and disabling a Message Processor

In a production setup, you will have multiple Routers and Message Processors to achieve optimal performance and you must enable/disable these Routers and Message Processors before/after upgrade. Take care of the following before/after upgrade:

- On combined Router and Message Processor node:
  - Before upgrade – perform the following:

i. Make the Router inaccessible by using the script:

```
/<inst-root>/apigee4/bin/reachability.sh -r -f false
```

ii. Remove Message Processor from the cluster using the script:

```
/<inst-root>/apigee4/bin/reachability.sh -m -f false
```

- o After upgrade - perform the following:

I. Add Message Processor to the cluster using the script:

```
/<inst-root>/apigee4/bin/reachability.sh -m -f true
```

II. Make the Router accessible using the script:

```
/<inst-root>/apigee4/bin/reachability.sh -r -f true
```

- On single Router node:
  - o Before upgrade – make the Router inaccessible:

    ```
    /<inst-root>/apigee4/bin/reachability.sh -r -f false
    ```

  - o After upgrade – make the Router accessible:

    ```
    /<inst-root>/apigee4/bin/reachability.sh -r -f true
    ```

- On single Message Processor node:
  - o Before upgrade – remove Message Processor from the cluster:

    ```
    /<inst-root>/apigee4/bin/reachability.sh -m -f false
    ```

  - o After upgrade – add Message Processor to the cluster:

    ```
    /<inst-root>/apigee4/bin/reachability.sh -m -f true
    ```

Upgrade prerequisites

Take care of following prerequisites before upgrading Apigee Edge:

- **Backup all nodes**

  Before you upgrade 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 upgrade to a new version doesn’t function properly. For more information on backup, see the *Apigee Edge Operations Guide,* available on the Apigee ftp site: ftp://ftp.apigee.com/.

- **Ensure Edge status is running**

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

```
> /<inst-root>/apigee4/bin/check.sh
```

Running the apigee-upgrade.sh script

Use the Edge upgrade script, `apigee-upgrade.sh`, to upgrade your Edge installation. When you invoke the script on a node, you must use the `-c` option to specify a comma-separated list of components to upgrade:

```
/<inst-root>/apigee4/share/installer/apigee-upgrade.sh -c qpid,ps
```

This example upgrades Qpid and Postgres. The list of possible components includes:

```
ldap = OpenLDAP
cs = Cassandra
zk = Zookeeper
qpid = Qpid
ps = Postgres
edge = Edge
ui = Edge UI
mo = Monetization
abs = Api Backend Services
abp = Api Backend Portal
all = upgrade all components on machine (only use if an aio profile)
```

The `edge` option corresponds to the Edge Router, Message Processor, and Management Server components and Apigee-specific upgrades to third-party components such as Qpid, Postgres, Cassandra, and Zookeeper. If you specify `edge`, then the upgrade script upgrades all components on the node together, or whatever subset of those components present on the node.

Before you use the `edge` option on a node with Qpid, Postgres, Cassandra, or Zookeeper installed, you must have already upgraded those components using the specific upgrade option for the component (`qpid`, `ps`, `cs`, and `zk`). See Order of machine upgrades for the correct order of running this script for your Edge installation and configuration.

Use the `all` option only for an installation of Edge on a single node, called an "all-in-one" installation. This type of installation is used for testing, not for production.

For silent upgrade, specify a config file to the script:

```
/<inst-root>/apigee4/share/installer/apigee-upgrade.sh -f upgrade-config
```

The *upgrade-config* file does not have to contain all of the properties required by the upgrade script. If you omit any required properties, then the script will prompt you for them.

Handling a failed upgrade

In the case of an upgrade failure, you can try to correct the issue, and then run the `apigee-upgrade.sh` script again. You can run the script multiple times and it will continue the upgrade from where it last left off.

If the failure requires that you roll back the upgrade to your previous version, use the `apigee-rollback.sh` script. See Rollback Process for more.

Procedure for upgrading to 4.15.07.00

To upgrade Apigee Edge to 4.15.07.00, perform the following steps:

1) Ensure that you have installed Java JDK 7 (either Oracle JDK 7 or OpenJDK 7) on all Edge processing nodes.

   Warning: This release of Edge does not support JDK 6. If you are currently using JDK 6, you must upgrade to JDK 7.

2) Ensure that `JAVA_HOME` points to the root directory of JDK 1.7 for the user performing the installation.

3) If present, disable have any CRON jobs configured to perform a repair operation on Cassandra until after the upgrade completes.

4) On each node, unzip the application distribution file from your download directory, and change directory (`cd`) into `apigee-edge-4.15.07.00.`

   ```
   unzip /download_dir/apigee-edge-4.15.07.00.zip
   ```

   ```
   cd apigee-edge-4.15.07.00
   ```

   Note: Unzip the file on the Linux machine itself. Do not unzip it on another machine, such as a Windows machine, and then copy the unzipped files to the Linux machine.

5) On each node, run the **upgrade install** script from the `apigee-edge-4.15.07.00` directory. The script copies new Apigee Edge sources to the `<install-root>/apigee4/share/installer`, `bin`, and `contrib` folders and sets `OPDK_VERSION` to 4.15.07.00:

   ```
   ./apigee-upgrade-install.sh -r /opt
   ```

   where `-r` denotes the installation root of your existing Edge installation and is used to find `<install-root>/apigee4/bin/apigee-env.sh`. This example uses `/opt`, which is the default `<install-root>` directory.

   The script prompts you for the home directory of your installation of Java JDK 7. Alternatively, use the `-j` option to the script to specify that directory:

   ```
   ./apigee-upgrade-install.sh -r /opt -j /usr/bin/java
   ```

6) Run the **upgrade** script on each node in the correct order for your installation configuration.

   Caution: The order of executing the upgrade script on nodes is important. You must upgrade the components in the correct order. See Order of machine upgrades for more information.

If prompted, enter the default Cassandra username and password of "cassandra", or enter your own custom Cassandra username/password if you changed it.

Caution: Upgrade all nodes, including the Postgres **master** node, **until you have to upgrade the Postgres standby server**. Then proceed to step 6 below to upgrade the Postgres standby server. If you only have a single analytics node with both Qpid and Postgres installed on the same node, meaning you are not using a Postgres standby server, **proceed to step 7 after upgrading the Qpid/Postgres node**.

Run the following upgrade script on each node in the correct order:

```
/<inst-root>/apigee4/share/installer/apigee-upgrade.sh -c profile
```

where *profile* specifies the components to upgrade.

For silent upgrade, run the following command on each node:

```
/<inst-root>/apigee4/share/installer/apigee-upgrade.sh -c profile
     -f upgrade-config
```

where *upgrade-config* is a text file containing the upgrade options. If you omit any required properties from the *upgrade-config* file, then the script will prompt you for them.

You can use the same *upgrade-config* file on all nodes except for Management Server nodes. A sample *upgrade-config* file is shown below:

```
IP1=10.11.111.111
IP2=10.22.222.222
IP3=10.33.333.333
MSNUM=1
APIGEE_ADMINPW=secret12
APIGEE_LDAPPW=secret21
CASS_HOSTS="$IP1 $IP2 $IP3"
CASS_USERNAME=cassandra  # or set to custom username
CASS_PASSWORD=cassandra  # or set to custom password
```

Only include `MSNUM` in the *upgrade-config* file when running the script on a Management Server node. Otherwise, omit it. `MSNUM` specifies the ID of the Management Server node. For example, in a 13-host clustered installation, you will have two Management Server nodes. On the first Management Server set `MSNUM` to 1, and set it to 2 on the second. You can designate either Management Server as `MSNUM` 1 as long as the other Management Server is designated as `MSNUM` 2.

`CASS_HOSTS` specifies the space delimited lists of IPs of the Cassandra nodes. In most configurations, this is the same set of IPs as for the ZooKeeper nodes. However, if the Cassandra and ZooKeeper nodes are on different IPs, use the `ZK_HOSTS` option in *upgrade-config* to specify the ZooKeeper IPs.

You can use two other options in the *upgrade-config* file for controlling Cassandra upgrades. Note that these options are only available when set in the *upgrade-config* file; you will not be prompted for them during an interactive upgrade.

```
SKIP_CASS_SCHEMA_UPGRADE=0  # set to 1 to skip the upgrade
NO_CASS_BACKUP=0  # set to 1 to perform backup
```

When you have multiple Management Servers, the schema on all Cassandra nodes is upgraded when you upgrade the first Management Server. Therefore, you can set `SKIP_CASS_SCHEMA_UPGRADE` to 1 on all Management Server nodes after the first one.

The upgrade procedure requires that you back up all Cassandra nodes before you run the upgrade. The upgrade script also backs up Cassandra nodes when it performs the upgrade. If you have performed that backup, you can optionally set `NO_CASS_BACKUP` to 1 to configure the upgrade script to omit its backup.

7) If you configured Postgres Master-Standby replication, use the following procedure to upgrade a Postgres **standby** server after first upgrading the Postgres **master** server:

   Note: When upgrading the Postgres standby server, the Postgres master can remain on to serve live traffic.

   A. Take the standby PostgreSQL database out of read-only mode by touching the file `trigger_file`. If this is the first time you have run this command, `trigger_file` will not exist:

   ```
   > touch /<inst-root>/apigee4/data/postgresql/pgdata/trigger_file
   ```

   B. Upgrade the Postgres standby server:

   ```
   > /<inst-root>/apigee4/share/installer/apigee-upgrade.sh -c ps
   ```

   For silent upgrade, run the following command on each node:

   ```
   > /<inst-root>/apigee4/share/installer/apigee–upgrade.sh -c ps
     –f upgrade–config
   ```

   C. Once the upgrade completes, you will need to reestablish Postgres master-standby replication. On the master Postgres server, check ssh connectivity to the standby server from the "apigee" user:

   ```
   > su - apigee
   > ssh apigee@standby-ip
   ```

   This command should allow connectivity from the master to the standby without a password. Repeat this step on the standby Postgres by running:

   ```
   > su - apigee
   > ssh apigee@master-ip
   ```

If you cannot connect, see Enabling Password-Less Login (SSH Key) between Master and Standby Postgres Machine.

Note: In previous releases of Edge, you might have configured the connection so that it was the root user connecting to a remote host, and not the "apigee" user. If necessary, you can configure the "apigee" user for access. For more information, see Enabling Password-Less Login (SSH Key) between Master and Standby Postgres Machine.

D. Stop PostgreSQL database on the standby server:

```
> /<inst-root>/apigee4/bin/apigee-service postgresql stop
```

E. On your master, run the replication script:

```
/<inst-root>/apigee4/share/installer/apigee-postgres-replication-setup.sh
```

a) When you are prompted to choose Postgres server type, choose option “m” for the master.

b) Enter the IP address or DNS name of the standby server.

c) Enter data directory path of Postgres peer node.

d) Enter the name of SSH key file for accessing 'master' or ‘standby’ (in this case, it is standby) as user 'apigee'. If you used the procedure described in Enabling Password-Less Login (SSH Key) between Master and Standby Postgres Machine, then the path is `/home/apigee/.ssh/id_rsa`.

Note: The replication script validates the accessibility of Postgres peer node using the SSH key. Upon failure, it displays an error message and directs you to set up trust relationship between master and standby. For more, see Set up Master-Standby Replication for Postgres.

This process establishes replication and syncs the data between master and standby. If you have a lot of data, this can take some time, sometimes in excess of 30 minutes.

F. After the replication script completes, on the Postgres standby server run the script:

```
/<inst-root>/apigee4/share/installer/apigee-postgres-replication-setup.sh
```

Note that here you choose option “s” for the standby and enter the IP address or DNS name of the master server. The script also starts Postgres.

G. On completion of replication, verify the replication status by issuing the following scripts on both servers.

a) On the master node, run:

```
> /<inst-root>/apigee4/bin/postgres-check-master.sh
```

Validate that it says it is the master.

b) On the standby node, run:

```
> /<inst-root>/apigee4/bin/postgres-check-standby.sh
```

Validate that it says it is the standby server.

8) After completing the upgrade of all Qpid and Postgres nodes, including Postgres Master and Standby nodes if applicable, complete the upgrade process for the remaining machines in your Edge installation in the required order. See Order of machine upgrades for more.

9) Restart the Management server, Message Processors, and Routers.

- To stop and start all Apigee components on a node:

```
> /<inst-root>/apigee4/bin/all-stop.sh
> /<inst-root>/apigee4/bin/all-start.sh
```

- Or to restart individual components:

```
> /<inst-root>/apigee4/bin/apigee-service management-server restart
> /<inst-root>/apigee4/bin/apigee-service message-processor restart
> /<inst-root>/apigee4/bin/apigee-service router restart
```

10) Launch your preferred browser and log into management UI to check the functionality

```
http://<MS-IP:9000>/login
```

This completes the upgrade procedures of Apigee Edge to 4.15.07.00.

Active-Active Analytics Upgrade in a Multi Data Center Environment

This section describes the procedure to migrate an Analytics group to Active-Active configuration. In an Active-Active configuration, the database in each of the multiple data centers contains the latest data in case one data center fails.

Note: Perform this procedure after upgrading all components to 4.15.07.00.

There are two steps to this process. You must perform both steps to complete the upgrade.

Terminology

The procedures below use the following abbreviations and terms:

- **Qpid**: Apache Qpid broker
- **PG/postgres-server**: Database server

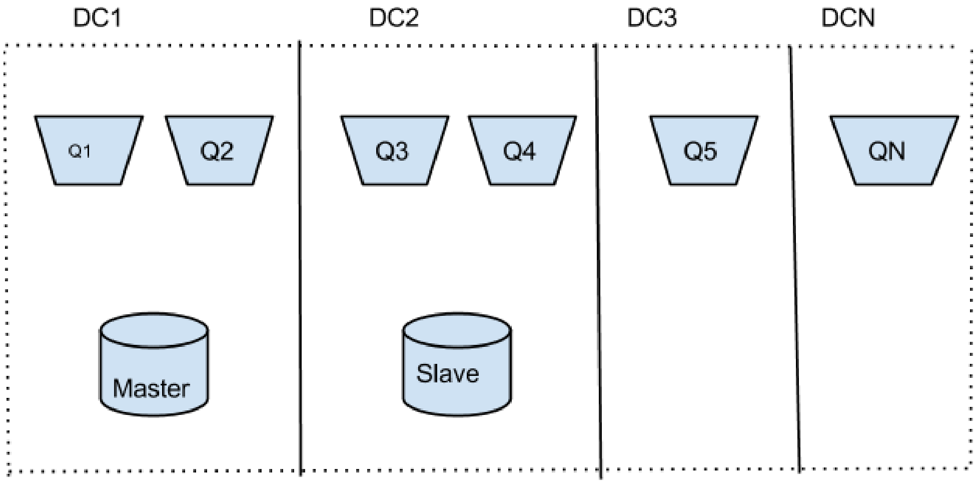
- **QPA**: Apigee Qpid server - Java process running in the same box as Apache Qpid
- **PGA**: Apigee Postgres server - Java process running in the same box as Postgres database
- **DC**: Data center
- **G1**: Current Group spanning multiple DCs
- **G2**: Target Group
- **Scope**: (org,env) combination

Upgrade Notes

- If a slave PG is added while a Qpid Agent/Mgmt server is running, then the Qpid Agent/Mgmt server must be restarted.
- The same Qpid Agent server cannot be shared across consumer groups, meaning across data centers.
- Existing DC QPAs need to be restarted when a new data center is added. The restart is required to create new consumer group queues in existing consumer group Qpids.
- When deleting a server from the main group, first delete the server from any consumer group and then from the main group.
- In case of Qpid deletion, the Qpid should be deleted from the main group and consumer group. If you delete it only from the consumer group, then data still flows through a Qpid removed only from the consumer group. Data does get insert into PG, so no data is lost.

Step 1: Migration

The following image shows an example of a master-slave PG set, where the master PG is in one DC and the slave PG is in another DC:



The goal of this procedure is to migrate this configuration to an Active-Active based configuration.

Assumptions:

- In-place server upgrade is preferable to adding all new servers.
- No loss of messages in ingestion flow during migration.

Migration procedure:

1. Stop QPAs in G1.
2. Make sure that PG database servers in G1 are replicated.
3. Stop PGs and PGAs in G1.

   Messages sent to AX system are collected in all G1 Qpid server queues.
4. Create the new group G2:

   ```
   curl -v -u <user>:<pwd> -X POST -H 'Accept:application/json' -H
   'Content-Type:application/json'
   "https://<management-ip>:<port>/v1/analytics/groups/ax/G2"
   ```
5. Assign the Qpids and PGs in G1 to G2. These servers are now shared between G1 and G2. Repeat the cURL command below for each Postgres server in G1.

   Note: Master-Slave representation has to be done as uui1, uuid2.

   ```
   curl -v -u <user>:<pwd>  -X POST -H 'Accept:application/json' -H
   'Content-Type:application/json'
   "https://<management-ip>:<port>/v1/analytics/groups/ax/G2/servers?u
   uid=<pg_uuid>&type=postgres-server"
   ```

   Repeat the command below for each Qpid server in G1:

   ```
   curl  -v -u <user>:<pwd>  -X POST -H 'Accept:application/json' -H
   'Content-Type:application/json'
   "https://<management-ip>:<port>/v1/analytics/groups/ax/G2/servers?u
   uid=<qpid_uuid>&type=qpid-server"
   ```
6. For each DC with PG in G1, create a consumer-group in G2. For example, in the figure above, DC1 would map to consumer group CG1, DC2 would map to consumer-group CG2. There would be no consumer group for DC3 ... DCN because they will not contain a PG server.

   a. For DC1:

   ```
   curl -v -u <user>:<pwd> -X POST -H 'Accept:application/json' -
   H 'Content-Type:application/json'
   "https://<management-ip>:<port>/v1/analytics/groups/ax/G2/cons
   umer-groups?name= CG1"
   ```

   b. For DC2:

   ```
   curl -v -u <user>:<pwd> -X POST -H 'Accept:application/json' -
   ```

```
H 'Content-Type:application/json'
"https://<management-ip>:<port>/v1/analytics/groups/ax/G2/cons
umer-groups?name= CG2"
```

7. Add the Qpid and PG servers in each DC (with PG) to respective consumer-group in G2, meaning servers in DC1 go to CG1, servers in DC2 go to CG2. To emphasize this, the Qpids would be added as consumers in consumer-group and PGs would be added as datastores in consumer-group.

    a. For DC1, add qpid-servers as consumers to CG1. Repeat the command below for each Qpid server:

    ```
    curl -v -u <user>:<pwd> -X POST -H 'Accept:application/json' -
    H 'Content-Type:application/json'
    "https://<management-ip>:<port>/v1/analytics/groups/ax/G2/cons
    umer-groups/CG1/consumers?UUID=<qpid-server-uuid>"
    ```

    b. Add postgres-servers as datastores to CG1. Repeat the below API for each postgres-server.

    Note: Master-Slave representation has to be done as uuid1, uuid2.

    ```
    curl -v -u <user>:<pwd> -X POST -H
    'Accept:application/json' -H 'Content-Type:application/json'
    "https://<management-ip>:<port>/v1/analytics/groups/ax/G2/cons
    umer-groups/CG1/datastores?UUID=<postgres-server-uuid>"
    ```

    c. Repeat #a and #b for DC2 using consumer group name as CG2.

8. The Qpid servers in remaining DCs in G1, such as DC3 to DCN, are not part off any consumer-group. They are just present in the servers section in G2.

9. Set group property `allow-cross-region-consumers` in G2:

    ```
    curl -v -u <user>:<pwd>  -X POST -H 'Accept:application/json' -H
    'Content-Type:application/json'
    "https://<management-ip>:<port>/v1/analytics/groups/ax/G2/propertie
    s?propName=allow-cross-region-consumers&propValue=true"
    ```

10. Restart all QPAs with latest RPM and check fanout configuration created in G2.

11. For example, the command `qpid-stat -e|grep G2` should reveal a fanout exchange. Also use the command `qpid-stat -q | grep G2`.

    In each consumer group, each Qpid should reveal queues with a name in the format: `ax-q-G2-<consumer-group-name>`. The number of queues created in each Qpid would be equal to the number of consumer-groups in G2.

12. On occasion, queue creation can take up to 15-20 minutes. Checks mentioned in #11 should be repeated a few times till the confirmation is obtained.

13. Because the Qpids are shared, QPAs should be listening to queues in the G1. Since the PG servers are still down, no messages will still be persisted. QPAs will try and throw errors on connecting to PG. This is fine.

14. Stop all the QPAs again.

    Messages sent to AX system are collected in all G1 Qpid server queues.

15. Move scope (S1) from G1 to G2.

    a. Delete S1 from G1:

    ```
    curl -v -u <user>:<pwd> -X DELETE
    'https://<management-ip>:<port>/v1/analytics/groups/ax/G1/scop
    es?org=<s1-org-name>&env=<s1-env-name>'
    ```

    b. Add S1 to G2:

    ```
    curl -v -u <user>:<pwd>  -H 'Accept:application/json' -H
    'Content-Type:application/json' -X POST
    'https://<management-ip>:<port>/v1/analytics/groups/ax/G2/scop
    es?org=<s1-org-name>&env=<s1-env-name>'
    ```

16. Ensure that MPs are now sending traffic to G2 instead of G1 by using the command:

    ```
    qpid-stat -q|grep G1
    ```

17. Use the data mover tool to move data from G1 Queues in all Qpids to G1 fanout exchange of any Qpid in G2.

    ```
    python exchange_sender.py -h {source_host} -q {source_queue} -t
    {target_host} -e {target_exchange}
    ```

    Repeat this command in every Qpid in G1, setting:
    - `source_host` as Qpid IP address.
    - `source_queue` is name of queue in `source_host`.
    - `target_host` as any Qpid server in G2.
    - `target_exchange` is name of exchange in `target_host`.

    Note: exchange_sender.py script is packaged inside distribution.

18. Ensure MASTER-SLAVE PGs (if any) in each consumer group are linked properly.

19. Start all PGs.

20. Execute Step 2 below to configure propagation across DCs.

21. Restart PGAs , QPAs.

    Active-Active setup is ready

Step 2: Configuring propagation across multiple data centers

The procedure in this step configures the Postgres server in each data center, and ensures that the different data centers can communicate with each other.

Assumptions:

- Postgres in each data center will have its own `analytics.api_pattern` table to which only it writes.
- The `analytics.api_pattern` table in each data center will have its own range of API-groupids, which cannot overlap.
- All PGs across data centers should be able to reach each other.

Ensure groupids in each analytics.api_pattern table do not overlap:

After the `analytics.api_pattern` table is created but before data starts flowing in, the sequence numbers needs to be changed for each data center.

Assuming there are 3 Data Centers, assume that max number of records that can exist in `analytics.api_pattern` table in any data center cannot exceed 500. The following tables lists the range of groupids for each data center:

Table 5: Data center sequence numbers

Data Center	Starting Range	Ending Range
Data Center 1	1	500
Data Center 2	501	1000
Data Center 3	1001	1500

By default, the range starts at 1 for a data center. Therefore, you do not have to do any configuration on Data Center 1. Use the procedure below to configure the sequence numbers for the other data centers:

1) Log in to the PostgreSQL database by using the command:

```
psql -U apigee -d apigee
```

2) Execute the following code in psql:

```
DO language plpgsql $$
DECLARE
    v_counter       int := 1;
    v_dummy         int;
    v_lower_range   int := <N>; -- Use the (DC starting range - 1) for N.
BEGIN
    WHILE v_counter <= v_lower_range
    LOOP
        SELECT nextval('analytics.group_id_seq') INTO v_dummy;
        v_counter := v_counter + 1;
    END LOOP;
```

```
END;
$$;
```

Ensure multi data center replication works correctly:

Use this procedure to ensure that the data centers can communicate with each other:

1. Create the following directory as a Postgres user:

   ```
   mkdir <inst-root>/apigee4/data/postgresql/pgdata/scripts
   ```

2. Copy `<inst-root>/bin/upsert_apipattern.sh` to `<inst-root>/apigee4/data/postgresql/pgdata/scripts`.
3. In each data center:
   a. Log into PostgreSQL database to get the psql prompt.
   b. Run the following command:

      ```
      CREATE EXTENSION dblink;
      ```

4. Restart apigee-qpid and apigee-pg processes.
5. Test if each DC can connect to each other. For example, use psql in Data Center 1 to connect to Data Center 2.
6. For each data center, use the following command to create a cron job that runs every five minutes:

   ```
   sh <inst-root>/apigee4/data/postgresql/pgdata/scripts/upsert_apipattern.sh
   <remoteDCPGHostname> <pguser>
   ```

7. Check the logs in `<inst-root>/apigee4/data/postgresql/pgdata/scripts/logs` to make sure the configuration is working correctly.

Rollback Process

The complete rollback process uses two scripts:

- `apigee-rollback.sh` - Rollback the upgrade but do not uninstall the 4.15.07.00 release. After running the `apigee-rollback.sh` script, you can retry the upgrade.
- `apigee-rollback-install.sh` - Uninstall 4.15.07.00 after running `apigee-rollback.sh` and, optionally, configures the node to use a specific version of the Java JDK.

Warning: The Edge 4.15.07.00 upgrade script installs Cassandra 2.0.15. If you upgrade to Cassandra 2.0.15, and then later decide to roll back your Edge 4.15.07.00 installation, then the rollback script does not restore your previous version of Cassandra. That means once you upgrade to Cassandra 2.0.15, you cannot rollback to the previous version. Therefore, if there is a chance that you will have to perform a rollback and want to go back to using Cassandra 1.2.x as part of that rollback, then you should install 4.15.07.00 in a testing environment and not in a production environment.

Warning: If you upgrade to OpenLDAP from ApacheDS, and then later decide to roll back your Edge 4.15.07.00 installation, then the rollback script does not restore ApacheDS. That means once you upgrade to OpenLDAP, you cannot rollback to ApacheDS. Therefore, if there is a chance that you will have to perform a rollback and want to go back to using ApacheDS as part of that rollback, then you should install 4.15.07.00 in a testing environment and not in a production environment.

Who can perform the rollback

The user running the rollback scripts should be the same as the user who originally upgraded Edge, or a user running as root.

Rollback 4.15.07.00

To rollback Apigee Edge from version 4.15.07.00, perform the following rollback steps:

1) Shutdown all Apigee component nodes (no zero downtime downgrade):

   ```
   /<inst-root>/apigee4/bin/all-stop.sh
   ```

2) On each node, run the `apigee-rollback.sh` script. The script downgrades the Apigee platform and configuration and resets `APIGEE_VERSION` and `OPDK_VERSION` to the older version:

   ```
   /<inst-root>/apigee4/share/installer/apigee-rollback.sh
   ```

   This script does not uninstall Edge 4.15.07.00. Therefore, if the upgrade failed for some reason, and you want to retry it, you can run the `apigee-upgrade.sh` script after running `apigee-rollback.sh`.

   Note: If your previous Edge release was configured to use Java JDK 6, you configured Edge to use Java JDK 7 as part of the upgrade process. The `apigee-rollback.sh` script does not reset the system to use Java JDK 6. You can use the process described in the Edge *Operations Guide,* available on the Apigee ftp site: ftp://ftp.apigee.com/, to configure the JDK version, or run the script described in the next step.

This completes the rollback procedures of Apigee Edge from version 4.15.07.00.

Appendix A: Silent 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.

There is a “default responses” option to the installer that is intended to let you create a responses file for silent installations (unattended installation). The response file lets you to store your responses to the required setup parameter prompts during a standard installation. An example response file, named `silent_config_example`, ships with Edge for Private Cloud. The file is located in the root directory after you unzip the `apigee-edge-4.15.xx.yy.zip` file.

The following installation scenarios (Gateway and Analytics) are described below:

- Standalone installation (Standalone Gateway and Analytics Setup)
- 5-host clustered installation (Minimum HA Setup)
- 9-host clustered installation (Performance HA Setup)
- 12-host clustered installation (Two Datacenter Setup)
- Monetization services installation (3-host installation)

This silent installation procedure only performs the Edge installation. You must still perform the onboarding procedure for each installation scenario. For example, after you perform a silent standalone installation, perform the onboarding steps described at Onboarding.

Note: The silent install procedure requires that the machine is configured such that the command `hostname -i` returns the IP address of the machine. If the machine is not configured correctly, edit `/etc/hosts`.

Standalone Installation

Install Gateway and Analytics on separate machines as follows.

Install and Configure Standalone Gateway: Machine 1

```
> ./apigee-install.sh -j /usr/java/default -r /opt -d /opt
```

```
> /<inst-root>/apigee4/share/installer/apigee-setup.sh -p sa -f config-2h
```

Install and Configure Standalone Analytics: Machine 2

```
> ./apigee-install.sh -j /usr/java/default -r /opt -d /opt
```

```
> /<inst-root>/apigee4/share/installer/apigee-setup.sh -p sax -f config-2h
```

For example, the response file `config-2h` file will look something like this:

```
# Without SMTP
HOSTIP=$(hostname -i)
MSIP=$IP1
ADMIN_EMAIL=opdk@apigee.com
APIGEE_ADMINPW=secret12
LICENSE_FILE=/tmp/license.txt
LDAP_TYPE=1
APIGEE_LDAPPW=secret
ENABLE_AX=y
MP_POD=gateway
REGION=dc-1
USE_ZK_CLUSTER=n
ZK_HOSTS="$IP1"
ZK_CLIENT_HOSTS="$IP1"
USE_CASS_CLUSTER=n
CASS_HOSTS="$IP1"
SKIP_SMTP=y
BIND_ON_ALL_INTERFACES=y
```

```
# With SMTP
HOSTIP=$(hostname -i)
MSIP=$IP1
ADMIN_EMAIL=opdk@apigee.com
APIGEE_ADMINPW=secret12
LICENSE_FILE=/tmp/license.txt
LDAP_TYPE=1
APIGEE_LDAPPW=secret
ENABLE_AX=y
MP_POD=gateway
REGION=dc-1
USE_ZK_CLUSTER=n
ZK_HOSTS="$IP1"
ZK_CLIENT_HOSTS="$IP1"
USE_CASS_CLUSTER=n
CASS_HOSTS="$IP1"
SKIP_SMTP=n
SMTPHOST=smtp.example.com
SMTPPORT=25
SMTPUSER=smtp@example.com  # =0 for no username
SMTPPASSWORD=smtppwd       # =0 for no password
SMTPSSL=n
BIND_ON_ALL_INTERFACES=y
```

Notes:

- It is assumed that `hostname -i` returns the correct IP address, and that $IP1 is set correctly. You can either replace $IP1 with the actual IP address, or set it as shown below:

  ```
  IP1=10.11.111.111
  ```

- `LDAP_TYPE=1` corresponds to OpenLDAP with no replication. If your server has an Internet connection, then the Edge upgrade 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.
- Ensure that you mention REGION names as "dc-1" for data center 1, "dc-2" for data center 2, and so on.
- If `BIND_ON_ALL_INTERFACES` is set to "y" then Router/Message Processors bind (listen) on all interfaces (IPs). If set to "n" then Router/Message Processors bind (listen) on a specific interface (IP specified by the `HOSTIP` variable).

5-host Clustered Installation

Install and Configure Datastore Cluster Node: Machine 1, 2 and 3

```
> ./apigee-install.sh -j /usr/java/default -r /opt -d /opt
```

```
> /<inst-root>/apigee4/share/installer/apigee-setup.sh -p ds -f config-5h
```

Install Apigee Management Server: Machine 1

```
> /<inst-root>/apigee4/share/installer/apigee-setup.sh -p ms -f config-5h
```

Install Apigee Router and Message Processor: Machine 2 and 3

```
> /<inst-root>/apigee4/share/installer/apigee-setup.sh -p rmp -f config-5h
```

Install Apigee Analytics: Machine 4 and 5

```
> ./apigee-install.sh -j /usr/java/default -r /opt -d /opt
```

```
> /<inst-root>/apigee4/share/installer/apigee-setup.sh -p sax -f config-5h
```

Set up Master-Slave Replication for Postgres

Set up a master-standby replication between Postgres Servers. Use the same procedure as described in the Five-host Clustered Installation section: Set up Master-Standby Replication for Postgres.

For example, the sample config-5h file will look something like this:

```
HOSTIP=$(hostname -i)
MSIP=$IP1
ADMIN_EMAIL=opdk@apigee.com
APIGEE_ADMINPW=secret12
LICENSE_FILE=/tmp/license.txt
USE_LDAP_REMOTE_HOST=n
LDAP_TYPE=1
APIGEE_LDAPPW=secret
ENABLE_AX=y
MP_POD=gateway
REGION=dc-1
USE_ZK_CLUSTER=y
ZK_HOSTS="$IP1 $IP2 $IP3"
ZK_CLIENT_HOSTS="$IP1 $IP2 $IP3"
USE_CASS_CLUSTER=y
CASS_HOSTS="$IP1 $IP2 $IP3"
SKIP_SMTP=n
SMTPHOST=smtp.example.com
SMTPPORT=25
SMTPUSER=smtp@example.com  # =0 for no username
SMTPPASSWORD=smtppwd       # =0 for no password
SMTPSSL=n
BIND_ON_ALL_INTERFACES=y
```

Notes:

- It is assumed that `hostname -i` returns the correct IP address, and that $IP1, $IP2, $IP3 are set correctly (or add the IP addresses into the silent configuration file). You can either replace $IP1, $IP2, $IP3 with the actual IP addresses, or set them as shown below:

  ```
  IP1=10.11.111.111
  ...
  ```

- `LDAP_TYPE=1` corresponds to OpenLDAP with no replication. If your server has an Internet connection, then the Edge upgrade 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.
- If `USE_LDAP_REMOTE_HOST` is set to "y" then you need to provide `LDAP_HOST` and `LDAP_PORT` also.
- Ensure that you mention REGION names as “dc-1” for data center 1, “dc-2” for data center 2, and so on.
- If `BIND_ON_ALL_INTERFACES` is set to "y" then Router/Message Processors bind (listen) on all interfaces (IPs). If set to “n” then Router/Message Processors bind (listen) on a specific interface (IP specified by the `HOSTIP` variable).

9-host Clustered Installation

Install and Configure Datastore Cluster Node: Machine 1, 2 and 3

```
> ./apigee-install.sh -j /usr/java/default -r /opt -d /opt
```

```
> /<inst-root>/apigee4/share/installer/apigee-setup.sh -p ds -f config-9h
```

Install Apigee Management Server: Machine 1

```
> /<inst-root>/apigee4/share/installer/apigee-setup.sh -p ms -f config-9h
```

Install Apigee Router and Message Processor: Machine 4 and 5

```
> ./apigee-install.sh -j /usr/java/default -r /opt -d /opt
```

```
> /<inst-root>/apigee4/share/installer/apigee-setup.sh -p rmp -f config-9h
```

Install Apigee Qpid Server: Machine 6 and 7

```
> ./apigee-install.sh -j /usr/java/default -r /opt -d /opt
```

```
> /<inst-root>/apigee4/share/installer/apigee-setup.sh -p qs -f config-9h
```

Install Apigee Postgres Server: Machine 8 and 9

```
> ./apigee-install.sh -j /usr/java/default -r /opt -d /opt
```

```
> /<inst-root>/apigee4/share/installer/apigee-setup.sh -p ps -f config-9h
```

Set up Master-Slave Replication for Postgres

Set up a master-standby replication between Postgres Servers. Use the same procedure as described in the Five-host Clustered Installation section: Set up Master-Standby Replication for Postgres.

For example, the sample config-9h file will look something like this:

```
HOSTIP=$(hostname -i)
MSIP=$IP1
ADMIN_EMAIL=opdk@apigee.com
APIGEE_ADMINPW=secret12
LICENSE_FILE=/tmp/license.txt
USE_LDAP_REMOTE_HOST=n
LDAP_TYPE=1
APIGEE_LDAPPW=secret
ENABLE_AX=y
MP_POD=gateway
REGION=dc-1
USE_ZK_CLUSTER=y
ZK_HOSTS=”$IP1 $IP2 $IP3"
ZK_CLIENT_HOSTS="$IP1 $IP2 $IP3"
USE_CASS_CLUSTER=y
CASS_HOSTS="$IP1 $IP2 $IP3"
SKIP_SMTP=n
SMTPHOST=smtp.example.com
SMTPPORT=25
SMTPUSER=smtp@example.com  # =0 for no username
SMTPPASSWORD=smtppwd       # =0 for no password
SMTPSSL=n
BIND_ON_ALL_INTERFACES=y
```

Notes:

- It is assumed that `hostname -i` returns the correct IP address, and that $IP1, $IP2, $IP3 are set correctly (or add the IP addresses into the silent configuration file). You can either replace $IP1, $IP2, $IP3 with the actual IP addresses, or set them as shown below:

  ```
   IP1=10.11.111.111
  ...
  ```

- `LDAP_TYPE=1` corresponds to OpenLDAP with no replication. If your server has an Internet connection, then the Edge upgrade 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.

- If `USE_LDAP_REMOTE_HOST` is set to "y" then you need to provide `LDAP_HOST` and `LDAP_PORT` also.

- Ensure that you mention REGION names as “dc-1” for data center 1, “dc-2” for data center 2, and so on.

- If `BIND_ON_ALL_INTERFACES` is set to "y" then Router/Message Processors bind (listen) on all interfaces (IPs). If set to “n” then Router/Message Processors bind (listen) on a specific interface (IP specified by the `HOSTIP` variable).

12-host Clustered Installation

Install and Configure Datastore Cluster Node: Machine 1, 2, 3, 7, 8 and 9

```
> ./apigee-install.sh -j /usr/java/default -r /opt -d /opt
> /<inst-root>/apigee4/share/installer/apigee-setup.sh -p ds -f config-12h
```

Install Apigee Management Server: Machine 1 and 7

```
> /<inst-root>/apigee4/share/installer/apigee-setup.sh -p ms -f config-12h
```

Install Apigee Router and Message Processor: Machine 2, 3, 8 and 9

```
> /<inst-root>/apigee4/share/installer/apigee-setup.sh -p rmp -f config-12h
```

Install Apigee Qpid Server: Machine 4, 5, 10 and 11

```
> ./apigee-install.sh -j /usr/java/default -r /opt -d /opt
> /<inst-root>/apigee4/share/installer/apigee-setup.sh -p qs -f config-12h
```

Install Apigee Postgres Server: Machine 6 and 12

```
> ./apigee-install.sh -j /usr/java/default -r /opt -d /opt
> /<inst-root>/apigee4/share/installer/apigee-setup.sh -p ps -f config-12h
```

Set up Master-Slave Replication for Postgres

Set up a master-standby replication between Postgres servers. Use the same procedure as described in the Five-host Clustered Installation section: Set up Master-Standby Replication for Postgres.

For example, the sample config-12h file will look something like this:

Datacenter 1

```
HOSTIP=$(hostname -i)
MSIP=$IP1
ADMIN_EMAIL=opdk@apigee.com
APIGEE_ADMINPW=secret12
LICENSE_FILE=/tmp/license.txt
USE_LDAP_REMOTE_HOST=n
LDAP_TYPE=2
LDAP_SID=1
LDAP_PEER=$IP4
APIGEE_LDAPPW=secret
ENABLE_AX=y
MP_POD=gateway-1
REGION=dc-1
USE_ZK_CLUSTER=y
ZK_HOSTS="$IP1 $IP2 $IP3 $IP4 $IP5
$IP6:observer"
ZK_CLIENT_HOSTS="$IP1 $IP2 $IP3"
USE_CASS_CLUSTER=y
CASS_HOSTS="$IP1:1,1 $IP2:1,1 $IP3:1,1
$IP4:2,1 $IP5:2,1 $IP6:2,1"
SKIP_SMTP=n
SMTPHOST=smtp.example.com
SMTPPORT=25
SMTPUSER=smtp@example.com # =0 for no
username
SMTPPASSWORD=smtppwd      # =0 for no
password
SMTPSSL=n
BIND_ON_ALL_INTERFACES=y
```

Datacenter 2

```
HOSTIP=$(hostname -i)
MSIP=$IP4
ADMIN_EMAIL=opdk@apigee.com
APIGEE_ADMINPW=secret12
LICENSE_FILE=/tmp/license.txt
USE_LDAP_REMOTE_HOST=n
LDAP_TYPE=2
LDAP_SID=2
LDAP_PEER=$IP1
APIGEE_LDAPPW=secret
ENABLE_AX=y
MP_POD=gateway-2
REGION=dc-2
USE_ZK_CLUSTER=y
ZK_HOSTS="$IP1 $IP2 $IP3 $IP4 $IP5
$IP6:observer"
ZK_CLIENT_HOSTS="$IP4 $IP5 $IP6"
USE_CASS_CLUSTER=y
CASS_HOSTS="$IP1:1,1 $IP2:1,1 $IP3:1,1
$IP4:2,1 $IP5:2,1 $IP6:2,1"
SKIP_SMTP=n
SMTPHOST=smtp.example.com
SMTPPORT=25
SMTPUSER=smtp@example.com # =0 for no
username
SMTPPASSWORD=smtppwd      # =0 for no
password
SMTPSSL=n
BIND_ON_ALL_INTERFACES=y
```

Notes:

- It is assumed that `hostname -i` returns the correct IP address, and that $IP1, $IP2, ..., $IP6 are set correctly (or add the IP addresses into the silent configuration file). You can either replace $IP1, $IP2, ..., $IP6 with the actual IP addresses, or set them as shown below:

  ```
   IP1=10.11.111.111
  ...
  ```

- 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.
- `LDAP_TYPE=2` corresponds to OpenLDAP with replication. If your server has an Internet connection, then the Edge upgrade 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.

- If `USE_LDAP_REMOTE_HOST` is set to "y" then you need to provide `LDAP_HOST` and `LDAP_PORT` also.
- Ensure that you mention REGION names as “dc-1” for data center 1, “dc-2” for data center 2, and so on since Cassandra modifier :**1**,1 leads to dc-**1**, :**2**,1 to dc-**2**, so on.
- If `BIND_ON_ALL_INTERFACES` is set to "y" then Router/Message Processors bind (listen) on all interfaces (IPs). If set to “n” then Router/Message Processors bind (listen) on a specific interface (IP specified by the `HOSTIP` variable).

Monetization Services Installation

The initial three-host monetization installation consists of three basic steps:

1) Install the **Apigee Management Server** on one host.
2) Install the **Apigee Router + Message Processor** on the second host.
3) Install the **Apigee Analytics** standalone on the third host.

You then install Monetization services on these three hosts in the following order:

1) Install Monetization on the **Apigee Analytics** host.
2) Install Monetization on the **Apigee Management Server** host.
3) Install Monetization on the **Apigee Router + Message Processor** host.

Note: Monetization requires that you have configured an SMTP server on the Management Server.

Install and Configure Management Server: Machine 1

```
> ./apigee-install.sh -j /usr/java/default -r /opt -d /opt
```

```
> /<inst-root>/apigee4/share/installer/apigee-setup.sh -p ms -f config-mon
```

Install Apigee Router and Message Processor: Machine 2

```
> ./apigee-install.sh -j /usr/java/default -r /opt -d /opt
```

```
> /<inst-root>/apigee4/share/installer/apigee-setup.sh -p rmp -f config-mon
```

Install Apigee Analytics: Machine 3

```
> ./apigee-install.sh -j /usr/java/default -r /opt -d /opt
```

```
> /<inst-root>/apigee4/share/installer/apigee-setup.sh -p sax -f config-mon
```

Integrate Monetization with Apigee Analytics: Machine 3

Note: make sure you install Monetization on the Analytics host first.

```
> /<inst-root>/apigee4/share/installer/apigee-setup.sh -p mo -f config-mon
```

Integrate Monetization with Management Server: Machine 1

```
> /<inst-root>/apigee4/share/installer/apigee-setup.sh -p mo -f config-mon
```

Integrate Monetization with Router and Message Processor: Machine 2

```
> /<inst-root>/apigee4/share/installer/apigee-setup.sh -p mo -f config-mon
```

For example, the response file `config-mon` file will look something like this:

```
#With SMTP
HOSTIP=$(hostname -i)
MSIP=$IP1
ADMIN_EMAIL=opdk@apigee.com
APIGEE_ADMINPW=secret12
LICENSE_FILE=/tmp/license.txt
LDAP_TYPE=1
APIGEE_LDAPPW=secret
ENABLE_AX=y
MP_POD=gateway
REGION=dc-1
USE_ZK_CLUSTER=n
ZK_HOSTS="$IP1"
ZK_CLIENT_HOSTS="$IP1"
USE_CASS_CLUSTER=n
CASS_HOSTS="$IP1"
#SMTP server required
SKIP_SMTP=n
SMTPHOST=smtp.example.com
SMTPPORT=465
SMTPUSER=smtp@example.com
SMTPPASSWORD=smtppwd
SMTPSSL=y
BIND_ON_ALL_INTERFACES=y
# PG host is the IP address of the Postgres server
MO_PG_HOST=$IP2
MO_PG_USER=mintpguser
MO_PG_PASSWD=mintpguser
```

Notes:

- It is assumed that `hostname -i` returns the correct IP address, and that $IP1 and $IP2 are set correctly. You can either replace $IP1 and $IP2 with the actual IP address, or set them as shown below:

  ```
  IP1=10.11.111.111
  IP2=10.22.222.222
  ```

- LDAP_TYPE=1 corresponds to OpenLDAP with no replication. If your server has an Internet connection, then the Edge upgrade 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.

- Ensure that you mention REGION names as “dc-1” for data center 1, “dc-2” for data center 2, and so on.
- If `BIND_ON_ALL_INTERFACES` is set to "y" then Router/Message Processors bind (listen) on all interfaces (IPs). If set to “n” then Router/Message Processors bind (listen) on a specific interface (IP specified by the HOSTIP variable).

Appendix B: API-DN Support

Apigee API Delivery Network (API-DN) is like a Content Delivery Network (CDN) for your API, dramatically improving speed, reliability, and consistency of service to improve app end-user experience. API-DN also improves your scalability and protects your back-end systems by offloading intensive API functionality to the local regions.

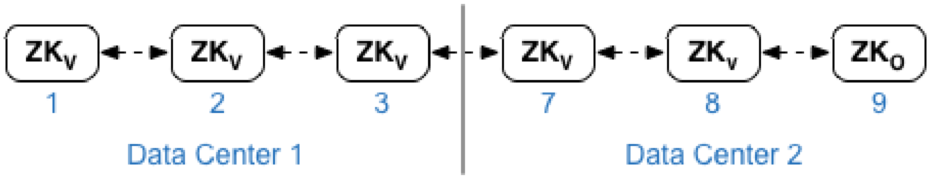
An on-premises deployment of Apigee Edge for Private Cloud with API-DN capabilities can decrease the latency disadvantage because you can take advantage of your datacenters in various parts of the world.

With an on-premises deployment, we've added support for API-DN, which provides the following features:

- **Support for multiple regions/datacenters** – An on-premises deployment treats each datacenter as a separate region and determines itself whether the setup has one or more datacenters. If more than one datacenter is found then setup prompts you to provide the datacenter name during installation of a node.
- **Support for ZooKeeper modifier** – ZooKeeper host now can be added as “observer”. Consider the following examples mentioned below:

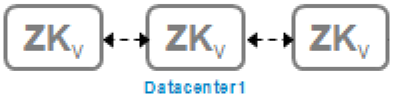
  Example 1: "192.168.124.201 192.168.124.202 192.168.124.203 192.168.124.204 192.168.124.205:observer"

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

  Example 2: "192.168.124.201 192.168.124.202 192.168.124.203"

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

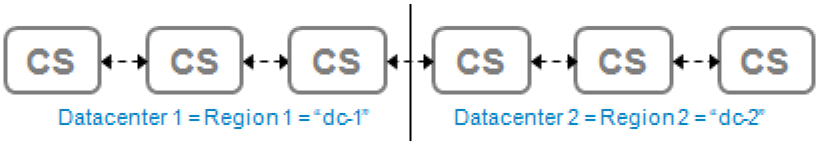
  

- **Support for complex Cassandra setup** – Cassandra host can have a suffix ':<d>,<r>', for example '<ip>:1,2 = datacenter 1 and rack/availability zone 2 and '<ip>:2,1 = datacenter 2 and rack/availability zone 1.

  Example 1: "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"

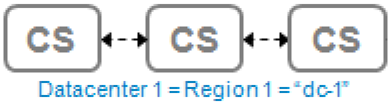
  If the suffix is missing, ':1,1' is assumed. All datacenters need to have the same number of nodes. The first node in rack/availability zone 1 of each datacenter will be used as seed servers.

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


CS
CS
CS
CS
CS
CS
Datacenter 1 = Region 1 = "dc-1"
Datacenter 2 = Region 2 = "dc-2"

Example 2: "192.168.124.201 192.168.124.202 192.168.124.203"

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



All parameters like seed nodes, token numbers and topology are calculated automatically by the setup.

- **Datacenter names** – Datacenter names are determined from the Cassandra setup. Default datacenter name is “dc-1”. In Cassandra setup example 1, it is clear that setup uses (:x,y) notation, then the datacenter/region name is derived as dc{x}. Hence datacenter/region1 is named as "dc-1" and datacenter/region 2 as "dc-2" as x=1 and 2.

  Similarly, in Cassandra setup example 2, the setup uses the default convention (without suffix). Hence datacenter name is dc-1.

- **Support for multiple Management Servers** – Now you can install multiple management servers using the same ZooKeeper and Cassandra nodes. The second management server detects whether the first management server has configured Cassandra schema and added security profile and if found, skip these configurations.

- **Support for registering datastores in datacenters without a Management Server** – Datastore nodes (Cassandra) get registered during installation of a management server. You can now register datastores without a Management Server. This is achieved by running the `/<inst-root>/apigee4/share/installer/apigee-register-non-ms-dc.sh` script on all datastore nodes with a management server in another datacenter. It is recommended to run this script after the installation of the management server(s).

- **OpenLDAP replication** – You now can set up an OpenLDAP master-master replication in multiple management servers’ deployment model.

  Note: The on-premises installation requires OpenLDAP *2.4* for LDAP master-master replication. On RHEL/CentOS, you can simply run "`yum install openldap-clients openldap-servers`" to install the utility.

- **Postgres replication** – You can now set up a master-standby replication between Postgres servers. One Postgres server can act as a master (primary Postgres server) and other as a standby (secondary Postgres server).

- **Creating trust relationship** – You can set up a trust relationship between master and standby using the procedure described in Set up Master-Standby Replication for Postgres.

Once the trust is successfully established, you can proceed for setting up the replication.

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

One Kingdom Street, 4th Floor
Paddington Central
London W2 6BD
UK

www.apigee.com