Platform Operation Part 2

Google Cloud

Topics

- Code with Config
- Component Reachability

apigee

Code with Config

Google Cloud

Configuring Edge

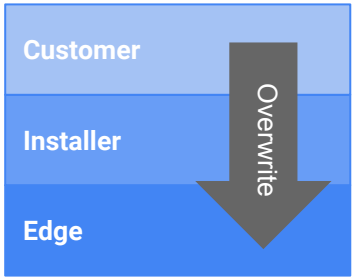
- Post-installation configuration of Edge uses a combination of properties files and `apigee-service` actions
  - Example: To configure SSL on the management API, edit properties files to set the necessary properties
  - Changes to properties files require you to restart the affected Edge component

- The technique of editing properties files is referred to as “code with config”
  - Consists of a key/value lookup tool based on settings in properties files
  - Keys are known as tokens
  - To configure Edge, set token values in properties files

- Code with config allows Edge components to ship with default values which are overridden by the installer and by customers
  - Installer overrides are written to `/opt/apigee/token`
  - Customer overrides are written to `/opt/apigee/customer`

- Code with config tokens are read from each location (default, installer, customer) and merged into the final runtime configuration for each component
  - Customer settings take priority
  - Installer settings come next
  - Default settings shipped with the product are used if no overrides are found

- The final runtime configuration is written to `/opt/apigee/<component>/conf`



apigee

Configuring Edge: Properties Files

- The `/opt/apigee/customer/application` directory contains customer-specific overrides
- Each Edge component has a dedicated properties file named named after the application name (component name without the prefix

  Examples:
  - `message-processor.properties`
  - `router.properties`
- The `/opt/apigee/customer/application` directory is empty by default
  - If you need to set a token value for a component without an existing properties file, create the properties file
- To set a component property edit the corresponding properties file to set the token to a value, then restart the component using `apigee-service,` for example, below shows settings for changing Java properties for the message processor

```
bin_setenv_min_mem=512m
bin_setenv_max_mem=1024m
bin_setenv_meta_space_size=1024m
```

- The `/opt/apigee/customer` directory is never touched when an upgrade performed; all customer overrides are guaranteed to be retained as long as they are written to that directory

apigee

Configuring Edge: Tokens

To easily determine all available token values, view `/opt/apigee/<component>/token/default.properties`. This file includes the defaults that ship for all tokens and which made be overridden by the installer in `/opt/apigee/token` or but the customer in `/opt/apigee/customer`.

Before you set a token in the properties file for the component, you can first determine its current value by using the following command:

```
/opt/apigee/apigee-service/bin/apigee-service <component> configure -search <token>
```

where <component> is the name of the component, and <token> is the token to inspect.

This command searches the hierarchy of properties files to determine the current value of the token.

For example, to check the current value of the conf_router_HTTP.request.line.limit token for the router:

```
/opt/apigee/apigee-service/bin/apigee-service edge-router configure -search conf_router_HTTP.request.line.limit
```

You should see output in the form:

```
Found key conf_router_HTTP.request.line.limit, with value, 4k, in /opt/apigee/edge-router/token/default.properties
```

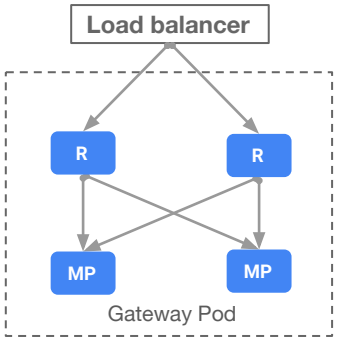
apigee

Component Reachability

Google Cloud

Router and Message Processor Reachability

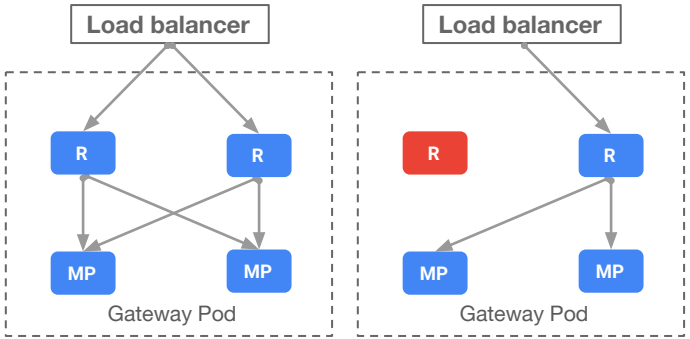
It is a good practice to disable reachability a server during maintenance, such as for a server restart or upgrade. When reachability is disabled, no traffic is directed to the server.



All R and MPs reachable

Router and Message Processor Reachability

It is a good practice to disable reachability a server during maintenance, such as for a server restart or upgrade. When reachability is disabled, no traffic is directed to the server.



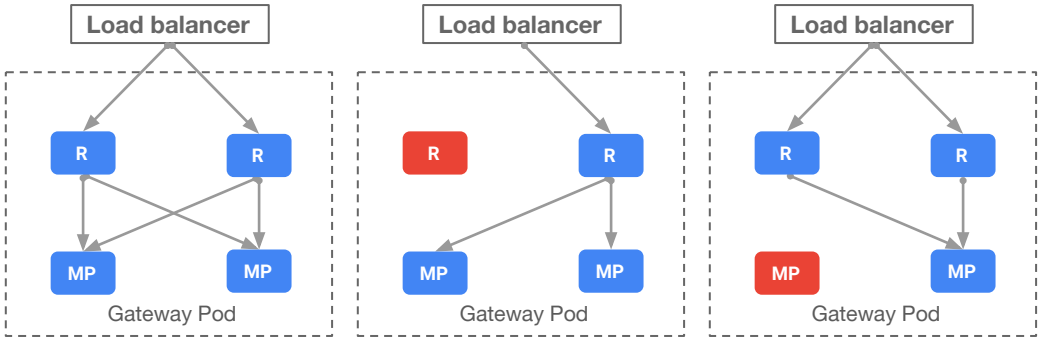
All R and MPs reachable

R reachability disabled
Load balancer will stop sending traffic to unreachable R

Load balancer configuration is performed by customer

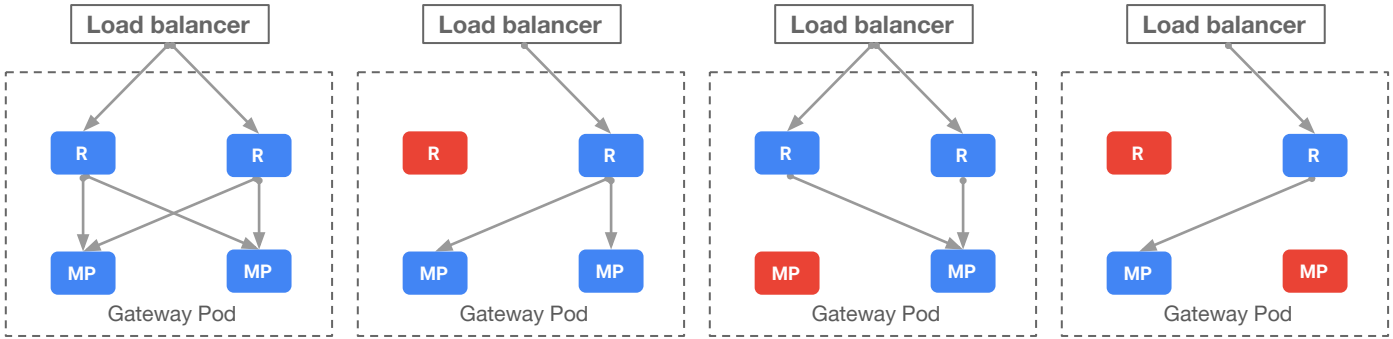
Router and Message Processor Reachability

It is a good practice to disable reachability a server during maintenance, such as for a server restart or upgrade. When reachability is disabled, no traffic is directed to the server.



Router and Message Processor Reachability

It is a good practice to disable reachability a server during maintenance, such as for a server restart or upgrade. When reachability is disabled, no traffic is directed to the server.



Disabling/enabling reachability on a Router

In a production environment, you typically have a load balancer in front of the Edge Routers. Load balancers monitor port 15999 on the Routers to ensure that the Route is available. To make a Router unreachable, you can block port 15999 on the Router. If the load balancer is unable to access the Router on port 15999 it no longer forwards requests to the Router.

For example, you can block the port by using the following iptables/firewalld command on the Router node:

```
sudo iptables -A INPUT -i eth0 -p tcp --dport 15999 -j REJECT
sudo firewall-cmd --zone=public --remove-port=12345/tcp --permanent
```

Once the change is complete, you can use the -D option to reverse the specific change for iptables or --add-port if you're using firewalld.:

```
sudo iptables -D INPUT -i eth0 -p tcp --dport 15999 -j REJECT
sudo firewall-cmd --zone=public --add-port=12345/tcp --permanent
```

apigee

Disabling/enabling reachability on a Message Processor

To disable reachability on Message Processor, you can just stop the Message Processor:

```
/opt/apigee/apigee-service/bin/apigee-service edge-message-processor stop
```

The Message Processor first processes any pending messages before it shuts down. Any new requests are routed to other available Message Processors.

To restart the Message Processor, use the following commands:

```
/opt/apigee/apigee-service/bin/apigee-service edge-message-processor start
/opt/apigee/apigee-service/bin/apigee-service edge-message-processor wait_for_ready
```

The wait_for_ready command returns the following message when the Message Processor is ready to process messages:

```
Checking if message-processor is up: message-processor is up.
```

Thank You

Google Cloud