Apigee Lab 2: Using Target Servers

experiment Lab schedule 1 hour 30 minutes universal_currency_alt No cost

show_chart Introductory

Overview

When you build proxies, it is important to not hardcode environment-specific configuration in a proxy.

In this lab, you will add a new environment and environment group to your organization. You will then use an environment-specific target server to specify the backend target URL.

Objectives

In this lab, you learn how to perform the following tasks:

- Create a new environment and environment group within an organization, and attach the environment to the runtime instance.
- Create a named target server in an environment.
- Modify a proxy to call a named target server instead of a hardcoded URL.

Setup

For each lab, you get a new Google Cloud project and set of resources for a fixed time at no cost.

1. Sign in to Qwiklabs using an **incognito window**.

2. Note the lab's access time (for example, `1:15:00`), and make sure you can finish within that time.
   There is no pause feature. You can restart if needed, but you have to start at the beginning.

3. When ready, click **Start lab**.

4. Note your lab credentials (**Username** and **Password**). You will use them to sign in to the Google Cloud Console.

5. Click **Open Google Console**.

6. Click **Use another account** and copy/paste credentials for **this** lab into the prompts.
   If you use other credentials, you'll receive errors or **incur charges**.

7. Accept the terms and skip the recovery resource page.

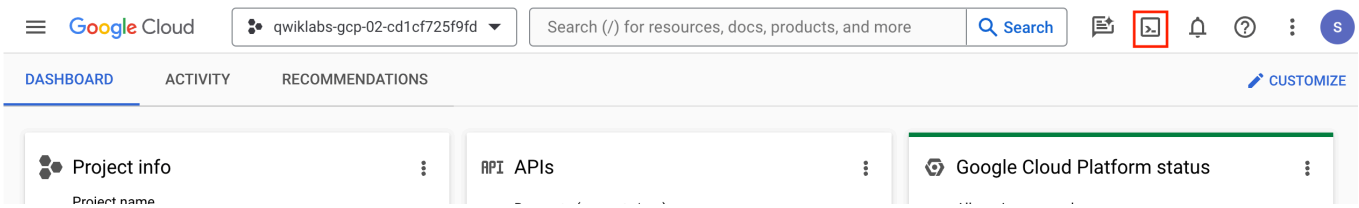
> **Note:** Do not click **End Lab** unless you have finished the lab or want to restart it. This clears your work and removes the project.

Activate Google Cloud Shell

Google Cloud Shell is a virtual machine that is loaded with development tools. It offers a persistent 5GB home directory and runs on the Google Cloud.

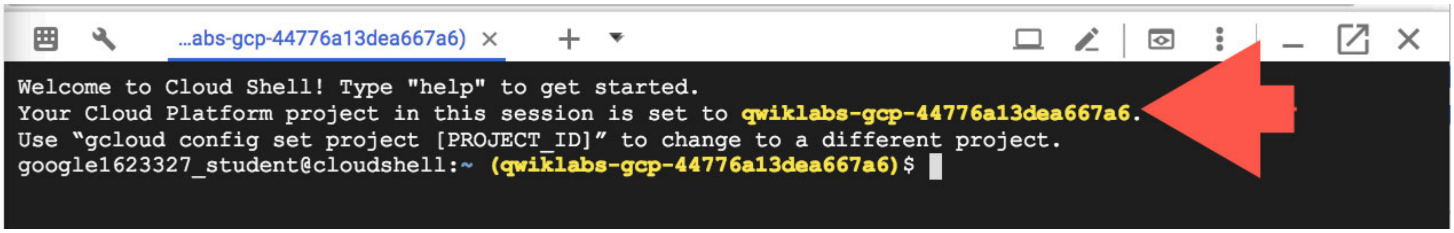
Google Cloud Shell provides command-line access to your Google Cloud resources.

1. In Cloud console, on the top right toolbar, click the Open Cloud Shell button.


Google Cloud
qwiklabs-gcp-02-cd1cf725f9fd
Search (/) for resources, docs, products, and more
Search
S
DASHBOARD
ACTIVITY
RECOMMENDATIONS
CUSTOMIZE
Project info
Project name
APIs
Google Cloud Platform status

2. Click **Continue**.

It takes a few moments to provision and connect to the environment. When you are connected, you are already authenticated, and the project is set to your *PROJECT_ID*. For example:



```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to qwiklabs-gcp-44776a13dea667a6.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
google1623327_student@cloudshell:~ (qwiklabs-gcp-44776a13dea667a6)$
```

gcloud is the command-line tool for Google Cloud. It comes pre-installed on Cloud Shell and supports tab-completion.

- You can list the active account name with this command:

```
gcloud auth list
```
content_c

Output:

```
Credentialed accounts:
 - @.com (active)
```

Example output:

```
Credentialed accounts:
 - google1623327_student@qwiklabs.net
```

- You can list the project ID with this command:

content_c

```
gcloud config list project
```

Output:

```
[core]
project =
```

Example output:

```
[core]
project = qwiklabs-gcp-44776a13dea667a6
```

Note: Full documentation of **gcloud** is available in the gcloud CLI overview guide .

Preloaded assets

This asset has already been added to the Apigee organization:

- The **retail-v1** API proxy

Note: Revision 1 of the retail-v1 proxy is marked as deployed, and is immutable. If you ever make a mistake in your proxy code that you can't recover from, you can select revision 1 and restart editing from there.

Task 1. Create a prod environment and environment group

In this task, you create a new environment group and environment for an organization.

Examine eval environment group and environment

1. In the Google Cloud console, on the **Navigation menu** (☰), select **Integration Services > Apigee > Management > Environments**.

2. Click on the **eval** environment.

   > **Note:** This lab uses an Apigee evaluation organization. By default, an evaluation organization has a single environment named "eval."

   This page shows the details for the eval environment. The eval environment is in the **eval-group** environment group.

   If the runtime is not yet available, you will be able to deploy API proxies to the environment, but the proxies will not accept API requests until the runtime is up, and the eval environment is attached.

   Once the runtime is up and the environment is attached, the runtime will query the management plane for the current API proxy deployments, and the runtime will host the API proxy and accept traffic.

3. In the left menu, click **Management > Environments**, and then select the **Environment Groups** tab.

   The page should show the default environment group named **eval-group**. It shows the hostnames and environments that are in the environment group. You should see the eval environment, as well as a long hostname. Selecting **Edit** from the more menu (⋮) lets you edit the environment group's hostnames and environments.

Create a prod environment group

1. Click **+Create Environment Group**.

2. Specify the following environment group settings:

Property	**Value**
Environment group name	**prod-group**
Hostnames	api- `qwiklabs-gcp-01-b16d63aa7546` .apiservices.dev

   Do not set **Environments (optional)**.

3. Click **Create**.

Create a prod environment

1. Select the **Environments** tab, and then click **+Create Environment**.

2. Specify the following environment settings:

Property	**Value**
Name	**prod**
Display name	**prod**
Environment group	*select* **prod-group**, and then click **OK**
Deployment type	*select* **Proxy**

> **Note:** The "Deployment type" setting controls the type of proxy development and deployment you will do within an environment.
>
> The "Proxy" deployment type is done using the Apigee UI, and is the type of development and deployment used in the labs for this course.

3. Click **Create**.

   The status of the new prod environment is **Inactive (not attached to an instance)**. A target server may be created for the environment, but proxies deployed to the environment won't handle traffic until the environment is successfully attached to an instance.

Task 2. Create a new target server for the eval environment

In this task, you create a new target server configuration for the eval environment.

Create a new target server

1. On the left navigation menu, select **Management > Environments**, and then click **eval**.

2. Select the **Target Servers** tab, and then click **+Create Target Server**.

3. Specify the following target server values:

Property	**Value**
Enable Target Server	*selected*
Target Server name	**TS-Retail**
Host	**gcp-cs-training-01-test.apigee.net**
Protocol	*select* **HTTP**
Port	**443**
Enable SSL	*selected*

   When the **SSL** box is selected, the dialog shows additional configuration fields. Leave the default SSL values set to their defaults.

4. Click **Create**.

5. Confirm that your values match the values in the image.

Task 3. Create a target server in the prod environment

In this task, you create a target server for the prod environment.

1. On the left navigation menu, select **Management > Environments**, and then click **prod**.

2. Select the **Target Servers** tab, and then click **+Create Target Server**.

3. Specify the following target server values:

Property	**Value**
Enable Target Server	*selected*
Target Server name	**TS-Retail**
Host	**gcp-cs-training-01-prod.apigee.net**
Protocol	*select* **HTTP**
Port	**443**
Enable SSL	*selected*

   These settings are identical to the settings for the eval target server, except for the Host.

4. Click **Create**.

Task 4. Update the API proxy to use the target server

In this task, you modify your retail API proxy to use the target server instead of the hardcoded target URL.

Update the API proxy

1. On the left navigation menu, select **Proxy Development > API proxies**, and then click **retail-v1**.

2. Click the **Develop** tab.

   You are modifying the version of the retail-v1 proxy that was created during Lab 1.

3. In the Navigator pane, click **Target Endpoints > default**.

   The TargetEndpoint configuration is displayed in the Code box.

4. In the **HTTPTargetConnection** section, on line 16, is the hardcoded URL.

   Replace the hardcoded URL with a reference to the target server you created.

   Replace:

   ```
   <HTTPTargetConnection>
     <Properties/>
     <URL>https://gcp-cs-training-01-test.apigee.net/training/db</URL>
   </HTTPTargetConnection>
   ```

   with:

   ```
   <HTTPTargetConnection>
     <LoadBalancer>
       <Server name="TS-Retail"/>
     </LoadBalancer>
     <Path>/training/db</Path>
   </HTTPTargetConnection>
   ```

   content_c

   Note that `/training/db` is now specified in the **Path** element. The target server, **TS-Retail**, specifies the hostname that will be used.

   Combined, the URL used for the call to the backend will be the same as the hardcoded URL in the eval environment. When deployed to the prod environment, the hostname will automatically change for the prod backend without any changes to the code.

5. To save the changes, click **Save**.

   The message box indicates that the changes could not be saved to revision 1 because it is deployed.

6. Click **Save As New Revision**.

7. Click **Deploy**.

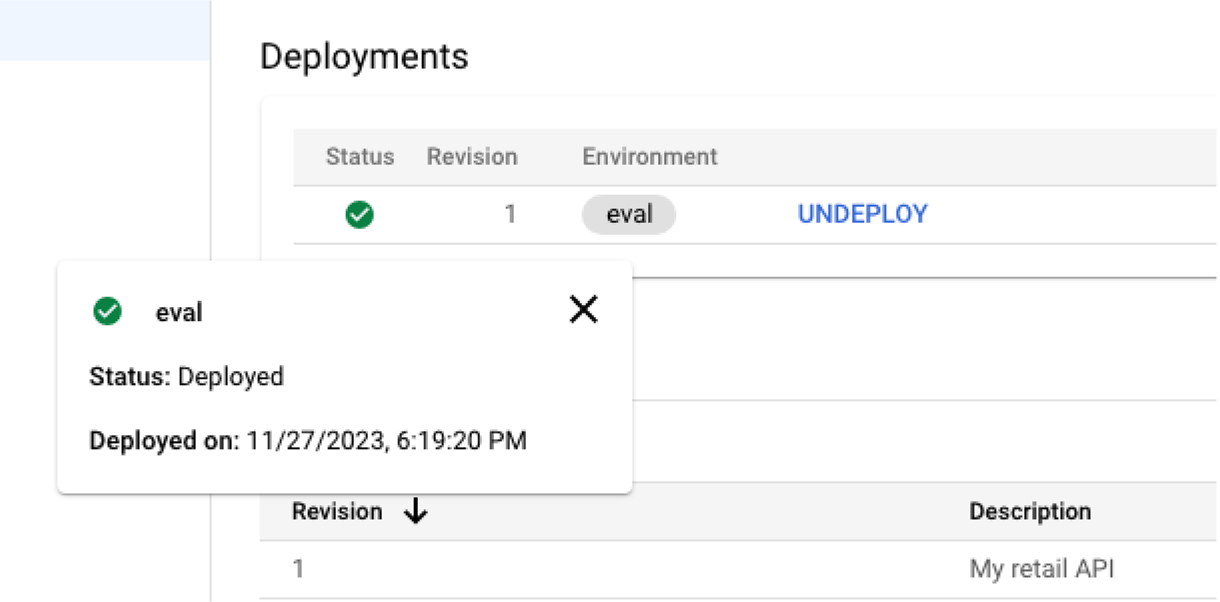
8. To specify that you want the new revision deployed to the eval environment, select **eval** as the **Environment**, and then click **Deploy**.

   > **Note:** Leave the "Service Account" field empty.

9. Click **Confirm**.

Check deployment status

A proxy that is deployed and ready to take traffic will show a green status.



Deployments

Status	Revision	Environment	
✓	1	eval	UNDEPLOY

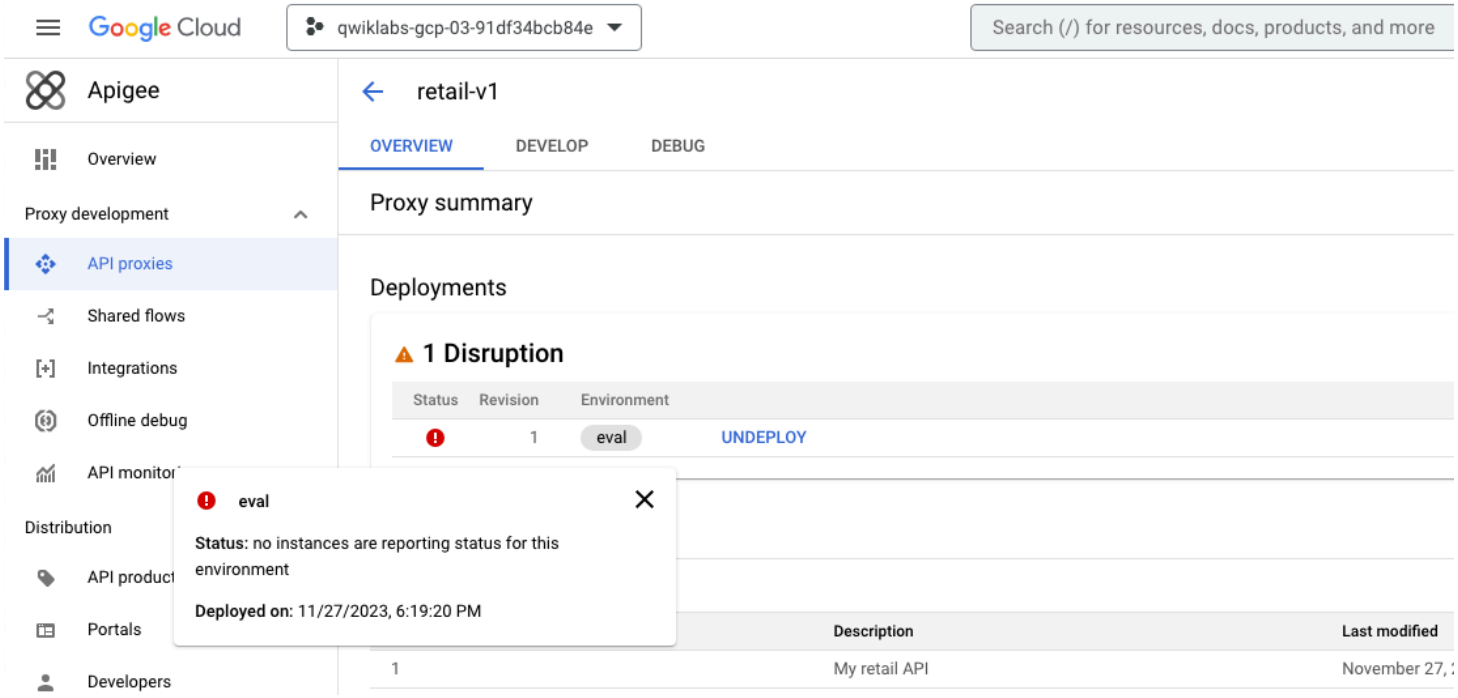
✓ eval ✕

Status: Deployed

Deployed on: 11/27/2023, 6:19:20 PM

Revision ↓	Description
1	My retail API

When a proxy is marked as deployed but the runtime is not yet available and the environment is not yet attached, you may see a red warning sign. Hold the pointer over the **Status** icon to see the current status.



Status	Revision	Environment	
●	1	eval	UNDEPLOY

eval

Status: no instances are reporting status for this environment

Deployed on: 11/27/2023, 6:19:20 PM

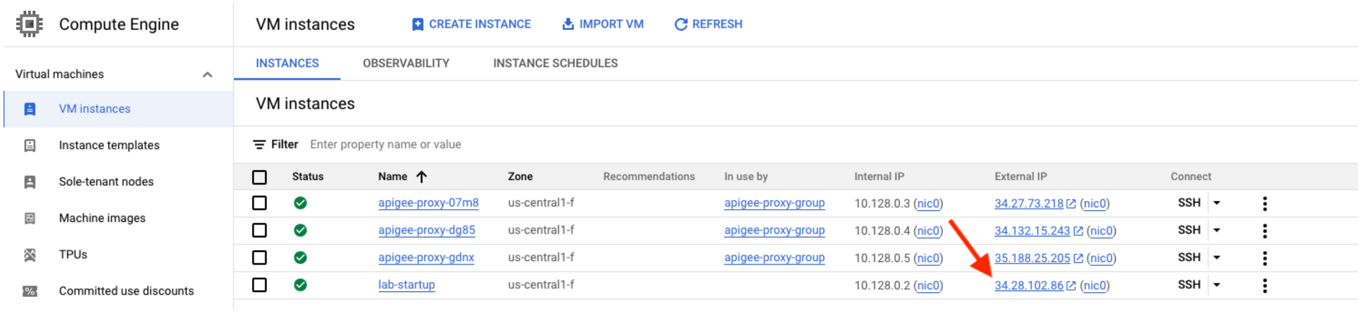
	Description	Last modified
1	My retail API	November 27,

If the proxy is deployed and shows as green, your proxy is ready for API traffic. If your proxy is not deployed because there are no runtime pods, you can check the status of provisioning.

Check provisioning dashboard

1. In the Google Cloud Console, navigate to **Compute Engine > VM instances**.

2. To open the Lab Startup Tasks dashboard, click on the **External IP** for the **lab-startup** VM.



□	Status	Name ↑	Zone	Recommendations	In use by	Internal IP	External IP	Connect
□	✓	apigee-proxy-07m8	us-central1-f		apigee-proxy-group	10.128.0.3 (nic0)	34.27.73.218 (nic0)	SSH
□	✓	apigee-proxy-dg85	us-central1-f		apigee-proxy-group	10.128.0.4 (nic0)	34.132.15.243 (nic0)	SSH
□	✓	apigee-proxy-gdnx	us-central1-f		apigee-proxy-group	10.128.0.5 (nic0)	35.188.25.205 (nic0)	SSH
□	✓	lab-startup	us-central1-f			10.128.0.2 (nic0)	34.28.102.86 (nic0)	SSH

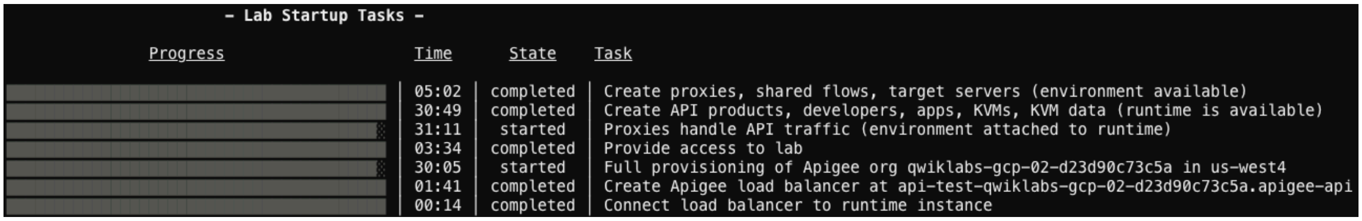
3. If you see a redirect notice page, click the link to the external IP address.

   A new browser window will open. Lab startup tasks are shown with their progress.

   - *Create proxies, shared flows, target servers* should be complete when you first enter the lab, allowing you to use the Apigee console for tasks like proxy editing.

- *Create API products, developers, apps, KVMs, KVM data* indicates when the runtime is available and those assets may be saved.

- *Proxies handle API traffic* indicates when the eval environment has been attached to the runtime and the deployed proxies can take runtime traffic.



```
                   - Lab Startup Tasks -

            Progress                 Time      State    Task

                                   | 05:02 | completed | Create proxies, shared flows, target servers (environment available)
                                   | 30:49 | completed | Create API products, developers, apps, KVMs, KVM data (runtime is available)
                                   | 31:11 |  started  | Proxies handle API traffic (environment attached to runtime)
                                   | 03:34 | completed | Provide access to lab
                                   | 30:05 |  started  | Full provisioning of Apigee org qwiklabs-gcp-02-d23d90c73c5a in us-west4
                                   | 01:41 | completed | Create Apigee load balancer at api-test-qwiklabs-gcp-02-d23d90c73c5a.apigee-api
                                   | 00:14 | completed | Connect load balancer to runtime instance
```

In this case, you need to wait for *Proxies handle API traffic* to complete.

While you are waiting

- Learn more about load balancing across backend servers.
- Learn more about environments and environment groups.

Task 5. Add the prod environment to the runtime instance

You can now deploy proxies to the prod environment, but they will not run unless the prod environment is associated with an instance.

Add the prod environment to the instance

1. Check the status of the instance:

```
curl -q -s -H "Authorization: Bearer $(gcloud auth print-access-
token)" -X GET
"https://apigee.googleapis.com/v1/organizations/${GOOGLE_CLOUD_PRO
```

content_c

A state of *CREATING* indicates that the runtime instance is not yet available. The state of *UPDATING* indicates that the instance was created but is now being updated. The state of *ACTIVE* indicates that the runtime instance is available. Once the runtime instance is ACTIVE, the eval environment will begin attaching to the instance.

2. To check the status of the **eval** attachment, run the following commands:

```
export ATTACHING_ENV=eval
export INSTANCE_NAME=eval-instance
echo "waiting for ${ATTACHING_ENV} attachment"
while : ; do export ATTACHMENT_DONE=$(curl -s -H "Authorization:
Bearer $(gcloud auth print-access-token)" -X GET
"https://apigee.googleapis.com/v1/organizations/${GOOGLE_CLOUD_PRO
| jq "select(.attachments != null) | .attachments[] |
select(.environment == \"${ATTACHING_ENV}\") | .environment" --
join-output); [[ "${ATTACHMENT_DONE}" != "${ATTACHING_ENV}" ]]
|| break; echo -n "."; sleep 5; done;
echo
echo "${ATTACHING_ENV} environment attached";
```

content_c

Note: Wait for the attachment of the eval environment to be complete before continuing.

3. On the left navigation menu, select **Management** > **Instances**, and then click **eval-instance**.

4. Click **Edit**.

5. To edit the environment details, click the pencil icon (✎).

6. For **Environments (optional)**, select both **eval** and **prod**, and then click **OK**.

7. Click **Save**.

8. In Cloud Shell, to check the status of the **prod** attachment, run the following commands:

content_c

```
export ATTACHING_ENV=prod
export INSTANCE_NAME=eval-instance
echo "waiting for ${ATTACHING_ENV} attachment"
while : ; do export ATTACHMENT_DONE=$(curl -s -H "Authorization:
Bearer $(gcloud auth print-access-token)" -X GET
"https://apigee.googleapis.com/v1/organizations/${GOOGLE_CLOUD_PRO
| jq "select(.attachments != null) | .attachments[] |
select(.environment == \"${ATTACHING_ENV}\") | .environment" --
join-output); [[ "${ATTACHMENT_DONE}" != "${ATTACHING_ENV}" ]]
|| break; echo -n "."; sleep 5; done;
echo
echo "${ATTACHING_ENV} environment attached";
```

Leave this command running in Cloud Shell.

The eval environment should be ready to serve traffic, so you can test the API proxy in the eval environment now.

Deploy the proxy to the prod environment

1. On the left navigation menu, select **Proxy Development > API proxies**, and then click **retail-v1**.

2. Select the **Develop** tab.

3. Click **Deploy**.

4. To specify that you want the latest revision deployed to the prod environment, select **prod** as the **Environment**, and then click **Deploy**.

5. Click **Confirm**.

   The proxy will not finish deploying until the prod environment has fully attached to the runtime instance.

Task 6. Test the modified API proxy in the eval environment

In this task, you use the debug tool to test that the updated proxy still successfully calls the backend service.

1. Select the **Debug** tab.

2. Click **Start Debug Session**.

3. In the **Start debug session** pane, on the Environment dropdown, select **eval**.

   The deployed revision number will also show in the dropdown.

4. Click **Start**.

5. To make a call to proxy hosted in the eval environment, in **Cloud Shell**, send a request to your API proxy by using this curl command:

   ```
   curl -i -X GET "https://api-test-${GOOGLE_CLOUD_PROJECT}.apiservice
   ```

   The response from the curl command should be a **200** status code if your backend URL was correctly set. The backend service set the response headers, including the header named *backend*. In this case, the value of the backend header is *eval*.

   A transaction for this request should soon appear in the Transactions pane on the left. If the transaction doesn't show up, try sending the request again.

   > **Note:** Apigee debug traffic is retrieved by asynchronously polling for new API calls, so there will be a delay between when an API request is completed, and when it shows in the debug tool.

6. Select a transaction in the Transactions pane.

   When a transaction is selected, you'll see a trace of the request and response through Apigee.

7. In the transaction details pane, click **Target Response Flow Started**.

   In **Response headers**, you see that the backend sent a header named **backend** with a value of **eval**.

8. In the transaction details pane, click **AnalyticsPublisher**.

   Apigee captures analytics for the API calls that travel through your proxies. This step in the transaction details shows the variables that were captured.

   The variable **target.url** contains the hostname that was specified in the eval target server and called by the Apigee proxy: `gcp-cs-training-01-test.apigee.net` .

Task 7. Test the API proxy in the prod environment

In this task, you use the debug tool to test that the proxy, when deployed to the prod environment, calls the prod backend service.

Check to see whether the prod environment is attached yet

1. In Cloud Shell, confirm that the prod environment attachment has finished.

   > **Note:** You ran a command at the end of Task 5 that indicates when the prod environment is attached.

2. Select the **Debug** tab.

3. Click **Start Debug Session**.

4. In the **Start debug session** pane, on the Environment dropdown, select **prod**.

5. Click **Start**.

6. To make a call to proxy hosted in the prod environment, in Cloud Shell, send the following curl command:

   ```
   curl -i -X GET "https://api-${GOOGLE_CLOUD_PROJECT}.apiservices.dev
   ```

   content_c

   The value of the *backend* header is *prod*.

7. Verify the analytics variables for the prod request.

   The variable **target.url** contains the hostname that was specified in the prod target server and called by the Apigee proxy: `gcp-cs-training-01-prod.apigee.net` .

Congratulations!