

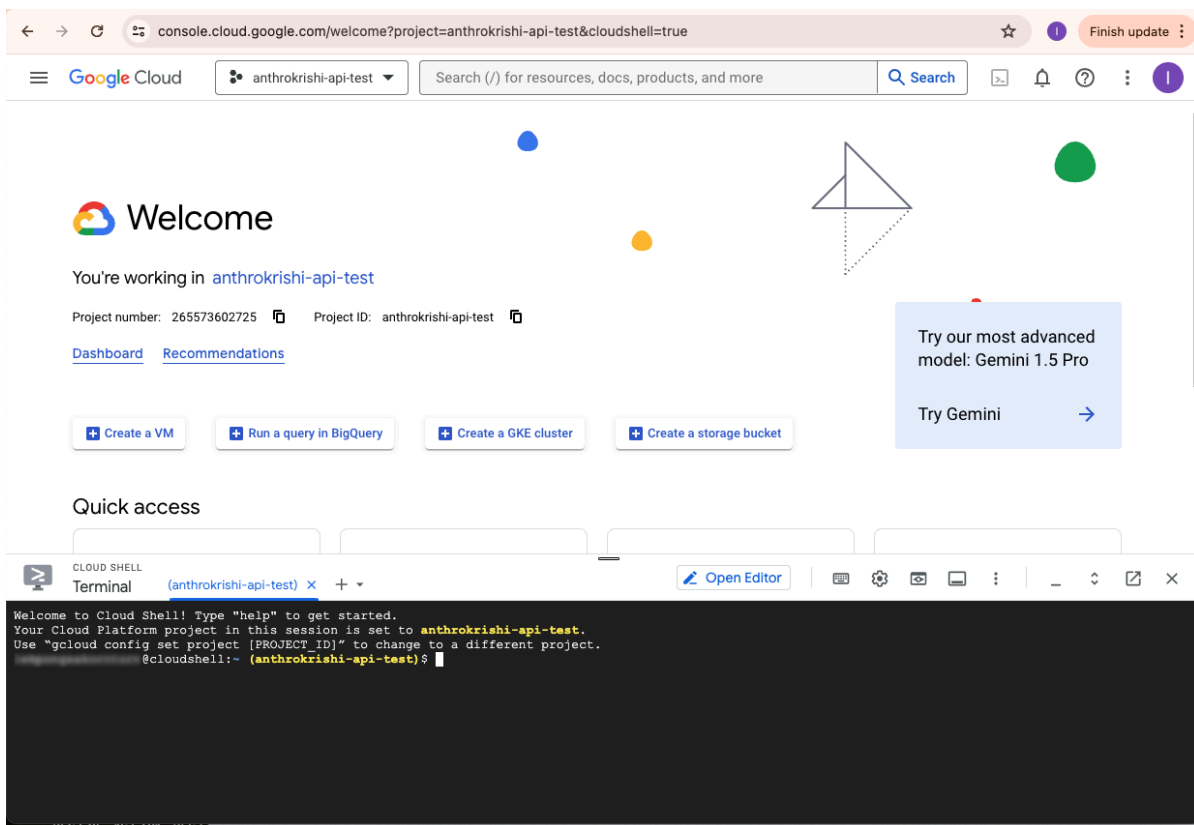
Anthrokrishi API | Customer Onboarding

Prerequisites

- GCP Billing and Project setup. Please use this [checklist](#) to enable and use all the necessary services.
- Obtain your GCP organization ID. Please follow this [guideline](#).

Cloud Setup Process

1. Setting up and deploying the API service
 - 1.1. Open a [Cloud Shell](#) in the GCP console.:



1.2. Clone the terraform script from this [repo](#).

Unset

```
git clone https://github.com/google/alu-api
```

```
@cloudshell:~ (anthrokrishi-api-test) $ git clone https://github.com/google/alu-api.git
```

1.3. Initialize Terraform

Unset

```
terraform init
```

```
@cloudshell:~/anthrokrishi-terraform/terraform (anthrokrishi-api-dev) $ terraform init
```

Initializing the backend...

Initializing modules...

- apigee-x-bridge-mig in modules/apigee-x-bridge-mig

Downloading git::https://github.com/terraform-google-modules/cloud-foundation-fabric.git?ref=v28.0.0 for apige

- nip-development-hostname in modules/nip-development-hostname

Downloading git::https://github.com/terraform-google-modules/cloud-foundation-fabric.git?ref=v28.0.0 for project...

- project in .terraform/modules/project/modules/project

Downloading git::https://github.com/terraform-google-modules/cloud-foundation-fabric.git?ref=v28.0.0 for vpc...

- vpc in .terraform/modules/vpc/modules/net-vpc

Initializing provider plugins...

- Reusing previous version of hashicorp/google from the dependency lock file

- Reusing previous version of hashicorp/google-beta from the dependency lock file

- Installing hashicorp/google v5.5.0...

- Installed hashicorp/google v5.5.0 (signed by HashiCorp)

- Installing hashicorp/google-beta v5.25.0...

- Installed hashicorp/google-beta v5.25.0 (signed by HashiCorp)

Terraform has made some changes to the provider dependency selections recorded in the .terraform.lock.hcl file. Review those changes and commit them to your version control system if they represent changes you intended to make.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see any changes that are required for your infrastructure. All Terraform commands should now work.

If you ever set or change modules or backend configuration for Terraform, rerun this command to reinitialize your working directory. If you forget, other commands will detect it and remind you to do so if necessary.

```
@cloudshell:~/anthrokrishi-terraform/terraform (anthrokrishi-api-dev) $
```

1.4. Preview the actions that Terraform plans to make to your infrastructure (Optional)

Unset

```
terraform plan -var-file="config.tfvars"
```

1.5. Executes the actions proposed in a Terraform plan

Unset

```
terraform apply -var-file="config.tfvars"
```

```
@cloudshell:~/anthrokrishi-terraform/terraform (anthrokrishi-api-dev)$ terraform apply -var-file="config.tfvars"
var.project_id
Project id (also used for the Apigee Organization).
Enter a value: anthrokrishi-api-dev
```

Type your GCP project ID

```
+ skip_initial_version_creation = false
+ terraform_labels              = (known after apply)
}

# module.apigee-x-core.module.kms-org-db.google_kms_key_ring.default[0] will be created
+ resource "google_kms_key_ring" "default" {
+   id           = (known after apply)
+   location     = "asia-south1"
+   name         = "apigee-x-org"
+   project      = "anthrokrishi-api-dev"
}

# module.apigee-x-core.module.kms-org-db.google_kms_key_ring_iam_binding.authoritative["roles/cloudkms.cryptoKeyEncrypterDecrypter"] will be created
+ resource "google_kms_key_ring_iam_binding" "authoritative" {
+   etag         = (known after apply)
+   id           = (known after apply)
+   key_ring_id = (known after apply)
+   members      = (known after apply)
+   role         = "roles/cloudkms.cryptoKeyEncrypterDecrypter"
}

Plan: 45 to add, 0 to change, 1 to destroy.

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes
```

Type 'Yes' to confirm terraform apply

```

ns/asia-south1/instanceGroupManagers/apigee-asia-south1]
module.mig-17xlb.google_compute_backend_service.mig_backend: Creating...
google_cloud_run_v2_service.main: Creation complete after 1m16s [id=projects/anthrokrishi-api-dev/locations/asia-south1/services/anthrokrishi-api]
google_cloud_run_v2_service.iam_member.member: Creating...
google_cloud_run_v2_service.iam_member.member: Creation complete after 6s [id=projects/anthrokrishi-api-dev/locations/asia-south1/services/anthrokrishi-api/iamMembers]
module.mig-17xlb.google_compute_backend_service.mig_backend: Still creating... [10s elapsed]
module.mig-17xlb.google_compute_backend_service.mig_backend: Still creating... [20s elapsed]
module.mig-17xlb.google_compute_backend_service.mig_backend: Still creating... [30s elapsed]
module.mig-17xlb.google_compute_backend_service.mig_backend: Still creating... [40s elapsed]
module.mig-17xlb.google_compute_backend_service.mig_backend: Still creating... [50s elapsed]
module.mig-17xlb.google_compute_backend_service.mig_backend: Still creating... [1m0s elapsed]
module.mig-17xlb.google_compute_backend_service.mig_backend: Creation complete after 1m7s [id=projects/anthrokrishi-api-dev/global/backendServices]
module.mig-17xlb.google_compute_url_map.url_map: Creating...
module.mig-17xlb.google_compute_url_map.url_map: Still creating... [10s elapsed]
module.mig-17xlb.google_compute_url_map.url_map: Creation complete after 12s [id=projects/anthrokrishi-api-dev/global/urlMaps/apigee-xlb]
module.mig-17xlb.google_compute_target_https_proxy.https_proxy: Creating...
module.mig-17xlb.google_compute_target_https_proxy.https_proxy: Still creating... [10s elapsed]
module.mig-17xlb.google_compute_target_https_proxy.https_proxy: Creation complete after 12s [id=projects/anthrokrishi-api-dev/global/targetHttpsProxies]
module.mig-17xlb.google_compute_global_forwarding_rule.forwarding_rule: Creating...
module.mig-17xlb.google_compute_global_forwarding_rule.forwarding_rule: Still creating... [10s elapsed]
module.mig-17xlb.google_compute_global_forwarding_rule.forwarding_rule: Still creating... [20s elapsed]
module.mig-17xlb.google_compute_global_forwarding_rule.forwarding_rule: Creation complete after 22s [id=projects/anthrokrishi-api-dev/global/forwardingRules]
module.project.google_project_service.project_services["compute.googleapis.com"] (deposed object f13fbbab): Destroying... [id=anthrokrishi-api-dev/global/projectServices/compute.googleapis.com]
module.project.google_project_service.project_services["compute.googleapis.com"]: Destruction complete after 0s

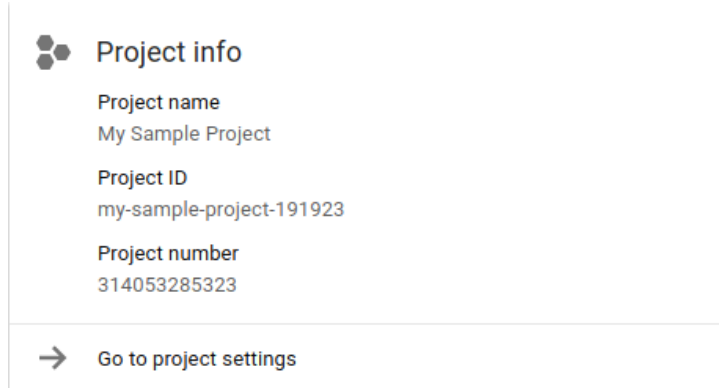
Apply complete! Resources: 10 added, 0 changed, 1 destroyed.
@cloudshell:~/anthrokrishi-terraform/terraform (anthrokrishi-api-dev) $ █

```

Terraform apply completed

2. Obtain your GCP organization ID and GCP project ID

- GCP organization ID
 - The organization resource ID is a unique identifier for an organization resource and is automatically created when your organization resource is created. Organization resource IDs are formatted as decimal numbers, and cannot have leading zeros.
 - You can get your organization resource ID using the Google Cloud console, the gcloud CLI, or the Cloud Resource Manager API. ([see more](#))
- GCP project ID
 - A project ID is a unique string used to differentiate your project from all others in Google Cloud. After you enter a project name, the Google Cloud console generates a unique project ID that can be a combination of letters, numbers, and hyphens.
 - Go to the [Dashboard page](#) in the Google Cloud console to get your project ID.

A screenshot of a 'Project info' card. It features a header with a hexagonal icon and the text 'Project info'. Below this, there are three rows of information: 'Project name' with the value 'My Sample Project', 'Project ID' with the value 'my-sample-project-191923', and 'Project number' with the value '314053285323'. At the bottom of the card, there is a horizontal line followed by a right-pointing arrow and the text 'Go to project settings'.

Project info card on the Dashboard page

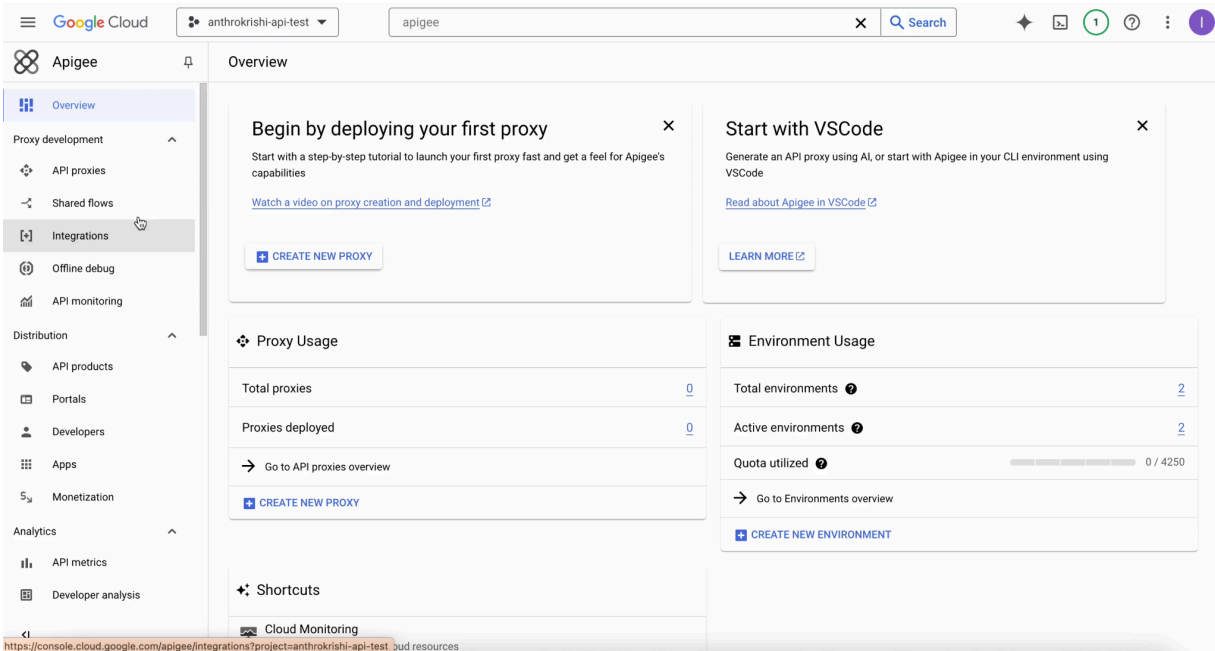
3. GCP organization ID and GCP project ID submission

The customer is required to send a GCP organization ID and a GCP project ID to the Google Team. And the Google team will add the customer's organization ID to the allow list and give the data access permission to the Service Account in the customer's project. Please submit your IDs via this [Google Form](#).

4. Setting up an Apigee and generate an API key

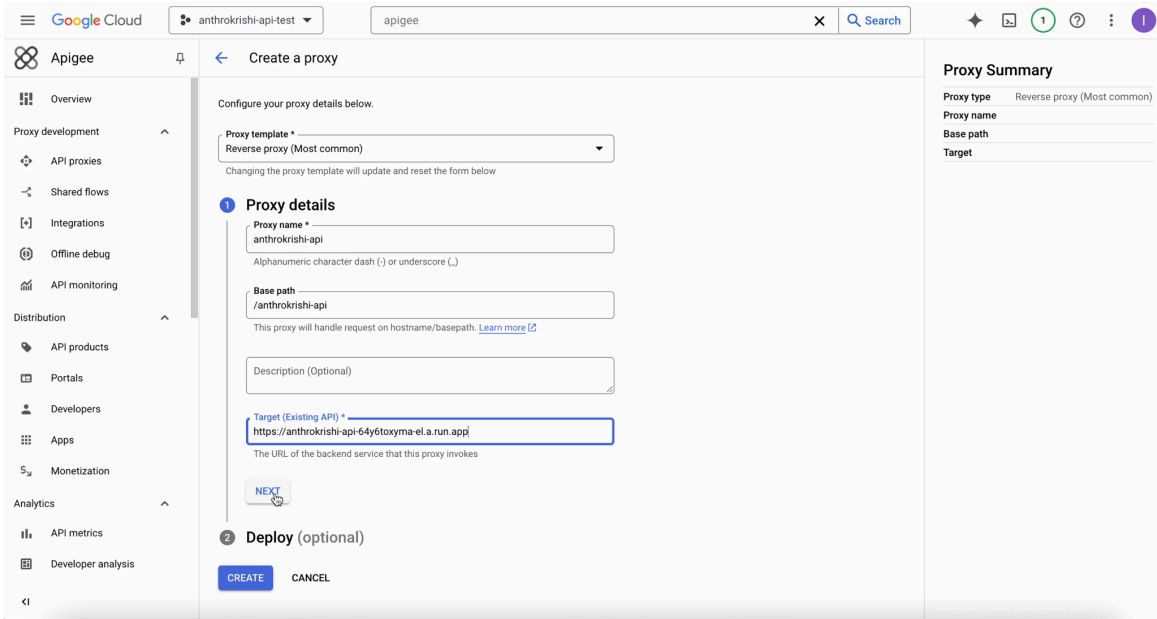
4.1. Create an proxy

4.1.1. Go to the [Apigee](#) console in the GCP console

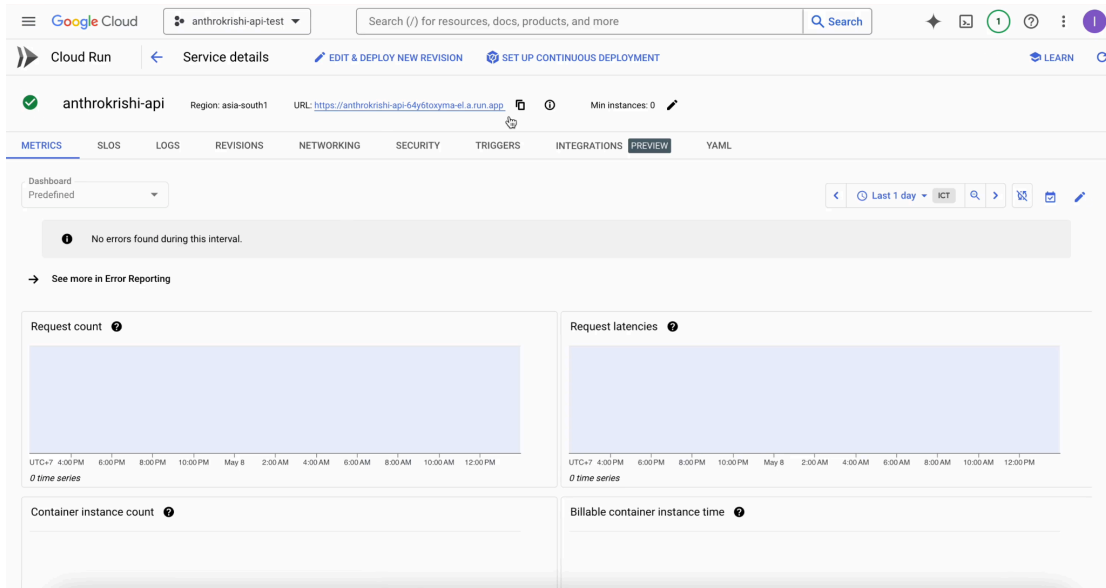


4.1.2. Navigate to "API Proxies" in the left-hand navigation menu and click **Create New Proxy**

- Set the "Proxy name" to "anthrokrishi-api"
- Set the "Base path" to "/anthrokrishi-api"
- Set the "Target (Existing API)" using an endpoint API from the [Cloud run](#) service
- Click **Create**



Configure the proxy details



Get a target URL from the Cloud run

The screenshot displays the Apigee console interface for the 'anthrokrishi-api' proxy. The left sidebar contains navigation options like Overview, Proxy development, and Distribution. The main content area shows the 'Proxy summary' with tabs for OVERVIEW, DEVELOP, and DEBUG. Under 'Deployments', a table lists two entries for revision 1, one for 'dev' and one for 'prod', both marked as 'UNDEPLOY'. Below this, the 'Revisions' section shows a table with one entry for revision 1, last modified on May 8, 2024. A 'Deployment status' dialog box is overlaid on the bottom right, confirming the deployment of revision 1 to both 'dev' and 'prod' environments.

A proxy is deployed

4.1.3. Set the policies

- Go to “API Proxies” and click the “anthrokrishi-api”.
- In the API Proxy editor, click the **Develop** tab.
- Click the **PreFlow** to open the policy editor.
- Click the **Add icon**

The screenshot shows the 'API Proxies' overview in the Apigee console. A table lists the 'anthrokrishi-api' proxy, which is deployed to 2 environments. The table columns include Name, Environment, Monitoring, Last Modified, and Actions. The 'anthrokrishi-api' entry is highlighted, and a mouse cursor is visible over it.

Google Cloud | anthrokrishi-api-test | apigee | Search

Apigee | anthrokrishi-api | SWITCH TO CLASSIC

Overview | OVERVIEW | DEVELOP | DEBUG

Proxy development

- API proxies
- Shared flows
- Integrations
- Offline debug
- API monitoring

Distribution

- API products
- Portals
- Developers
- Apps
- Monetization

Analytics

- API metrics
- Developer analysis

Proxy summary

DEPLOY | DUPLICATE | DELETE

Deployments

Status	Revision	Environment	
✓	1	prod	UNDEPLOY
✓	1	dev	UNDEPLOY

Revisions

Filter | Filter revisions

Revision	Description	Last modified	Endpoint summary
1		May 8, 2024	VIEW

https://console.cloud.google.com/apigee/proxies/anthrokrishi-api/develop/1?project=...

Google Cloud | anthrokrishi-api-test | apigee | Search

Apigee | anthrokrishi-api | SWITCH TO CLASSIC

Overview | OVERVIEW | DEVELOP | DEBUG

REVISION: 1 | Updated a minute ago | DEPLOY | SAVE

Search proxy

anthrokrishi-api

- Policies
- Proxy endpoints
 - default
 - ALL PreFlow
 - ALL PostFlow
 - Target endpoints
 - default
 - ALL PreFlow
 - ALL PostFlow
 - Integration endpoints
 - Resources

anthrokrishi-api Proxy revision

Details

Display name

Description

Contains

Proxy endpoints | default

Target endpoints | default

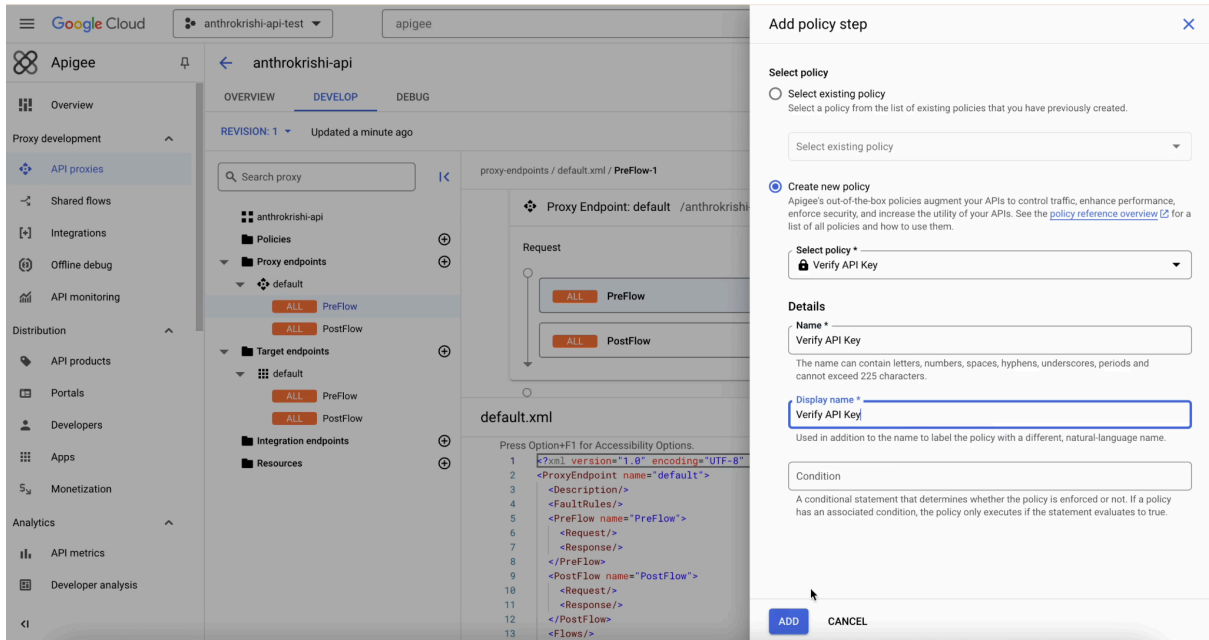
anthrokrishi-api.xml

```

1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <APIProxy revision="1" name="anthrokrishi-api">
3   <DisplayName/>
4   <Description/>
5   <CreatedAt>1715151547323</CreatedAt>
6   <LastModifiedAt>1715151547323</LastModifiedAt>
7   <BasePaths>/anthrokrishi-api</BasePaths>
8   <ProxyEndpoints>
9     <ProxyEndpoint>default</ProxyEndpoint>
10  </ProxyEndpoints>
11  <TargetEndpoints>
12    <TargetEndpoint>default</TargetEndpoint>
13  </TargetEndpoints>
14 </APIProxy>

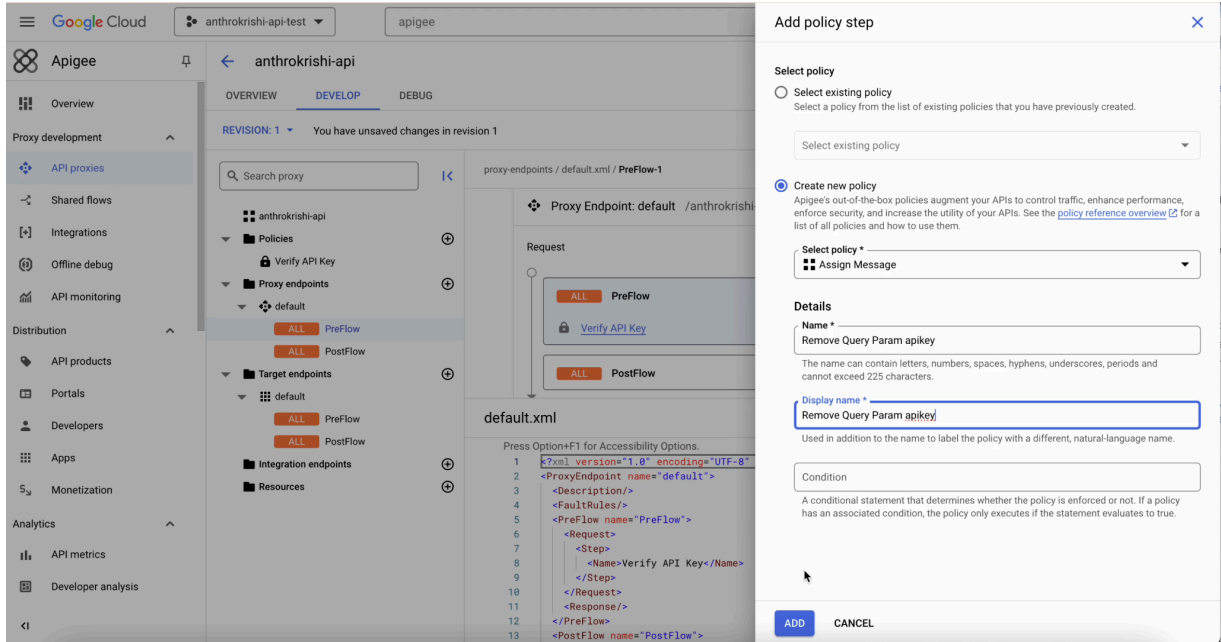
```

- In the add policy step editor, choose “Create new policy” and select “Verify API Key”.
 - Set the “Name” to “Verify API Key”.
 - Set the “Display name” to “Verify API Key”.
- Click **Add**.



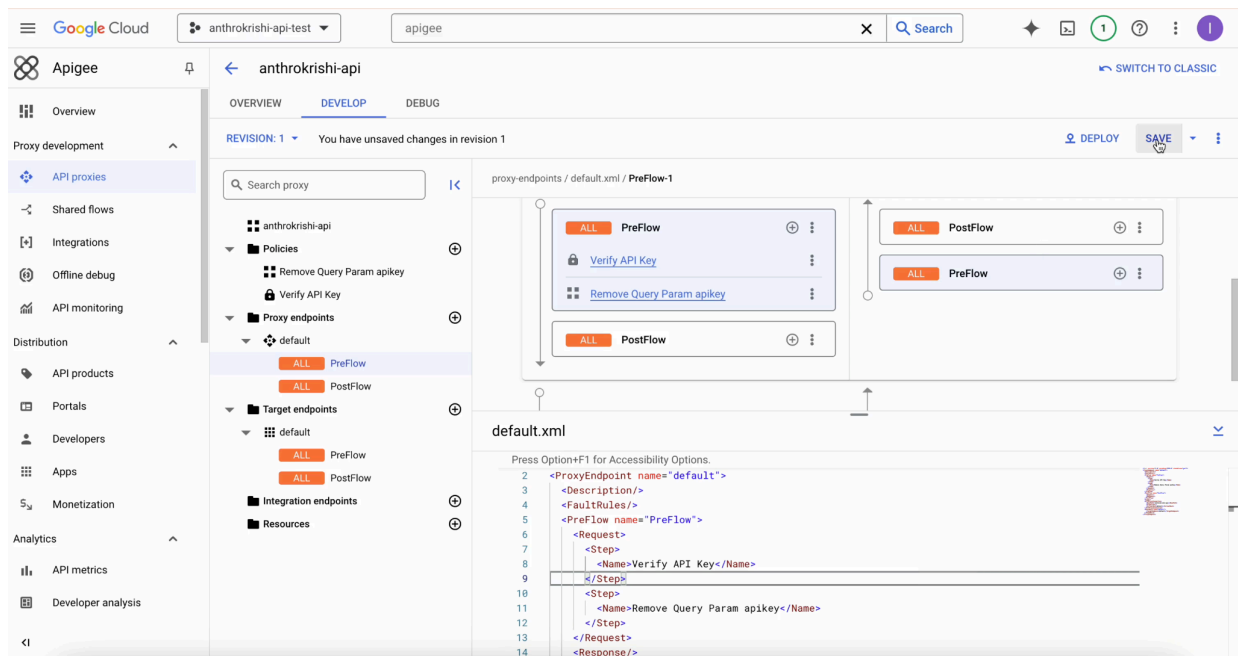
Create the 'Verify API Key' policy

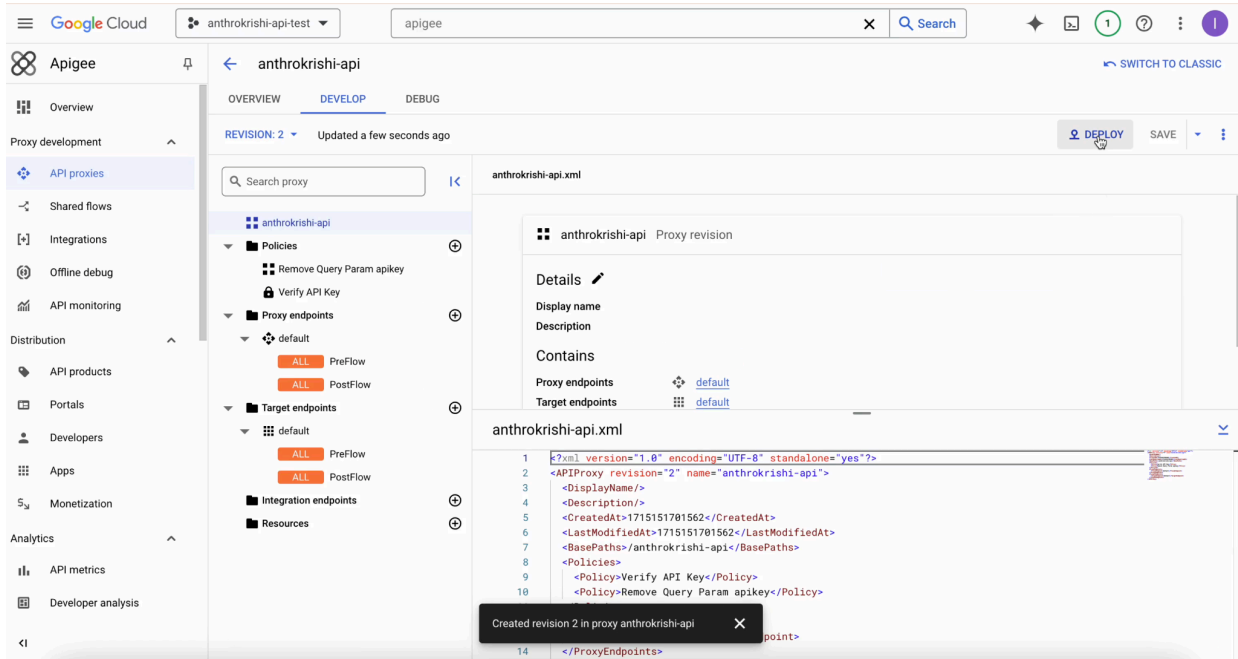
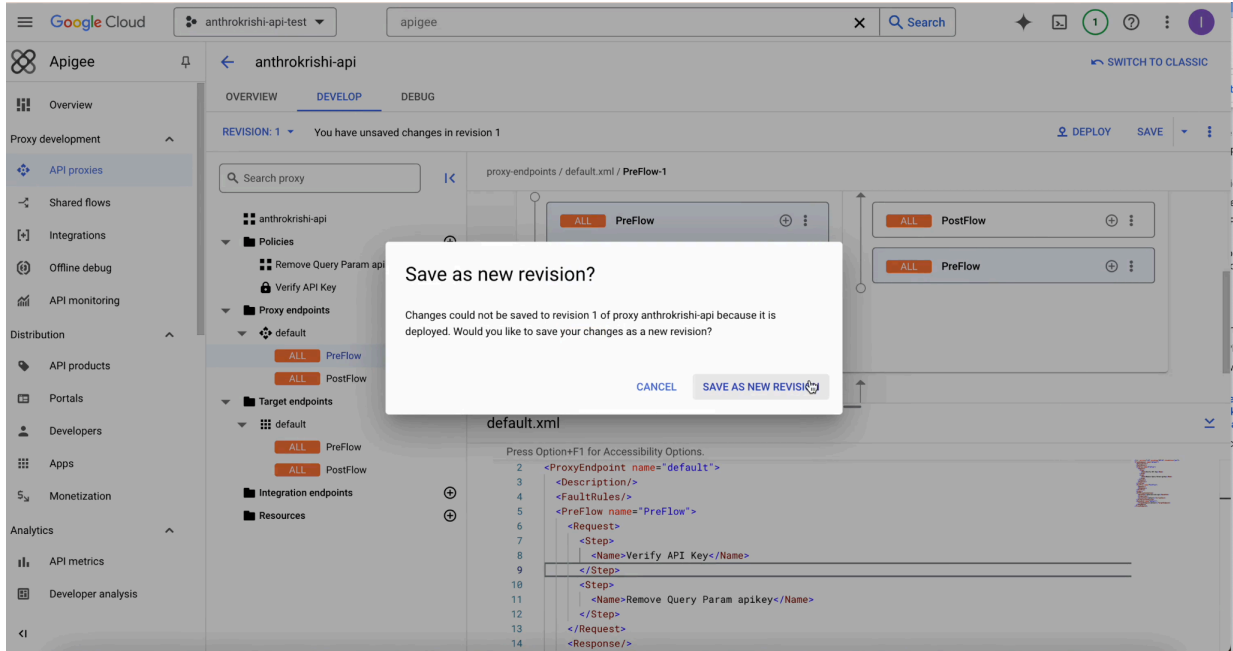
- Open add policy step editor again and choose “Create new policy” and select “Assign Message” in **Select policy** section.
 - Set the “Name” to “Remove Query Param apikey”.
 - Set the “Display name” to “Remove Query Param apikey”.
- Click **Add**.

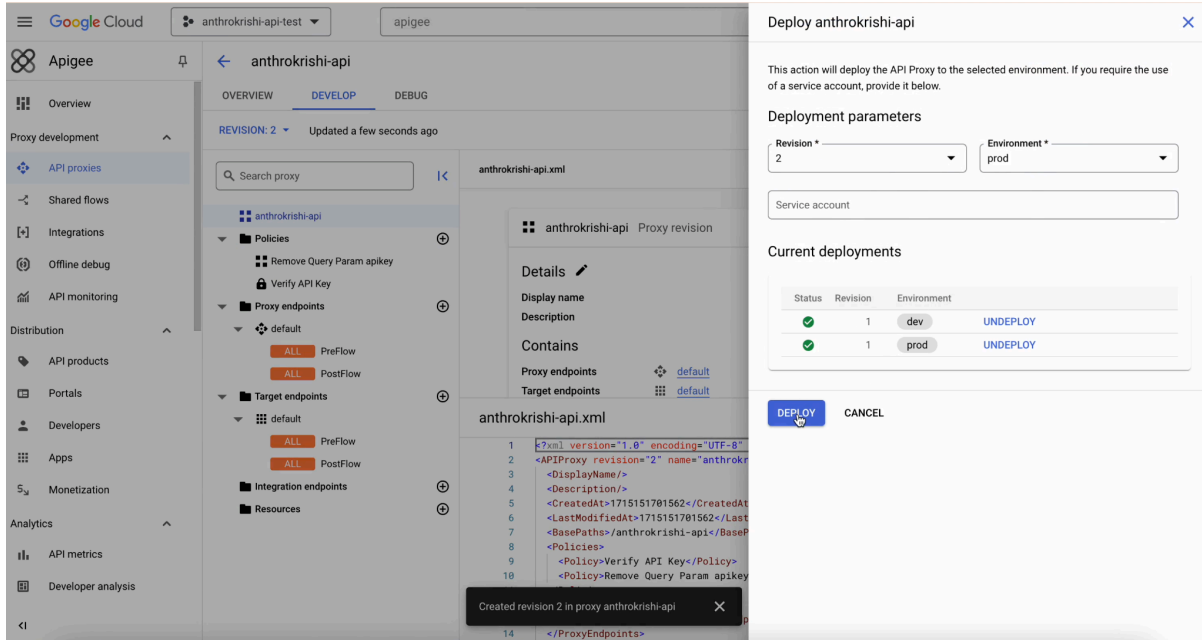


Create the 'Remove Query Param'

- In the API Proxy editor click **Save**.
- In the popup dialogue click **SAVE AS NEW REVISION**.
- Click **Deploy**.



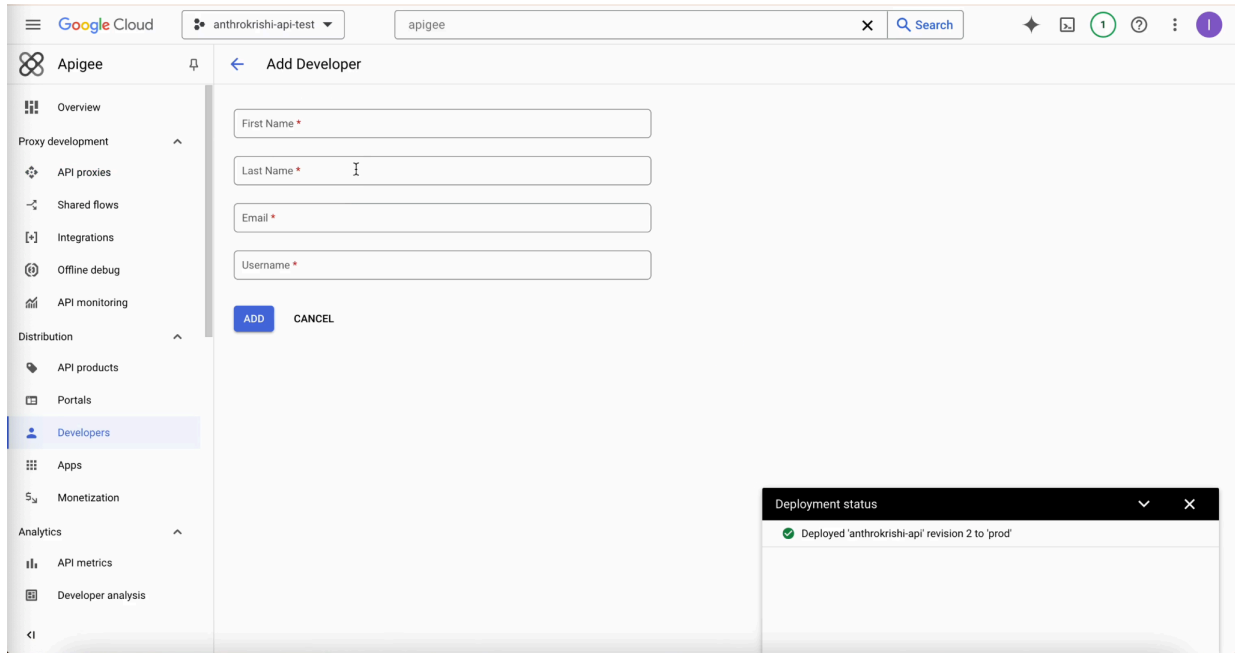




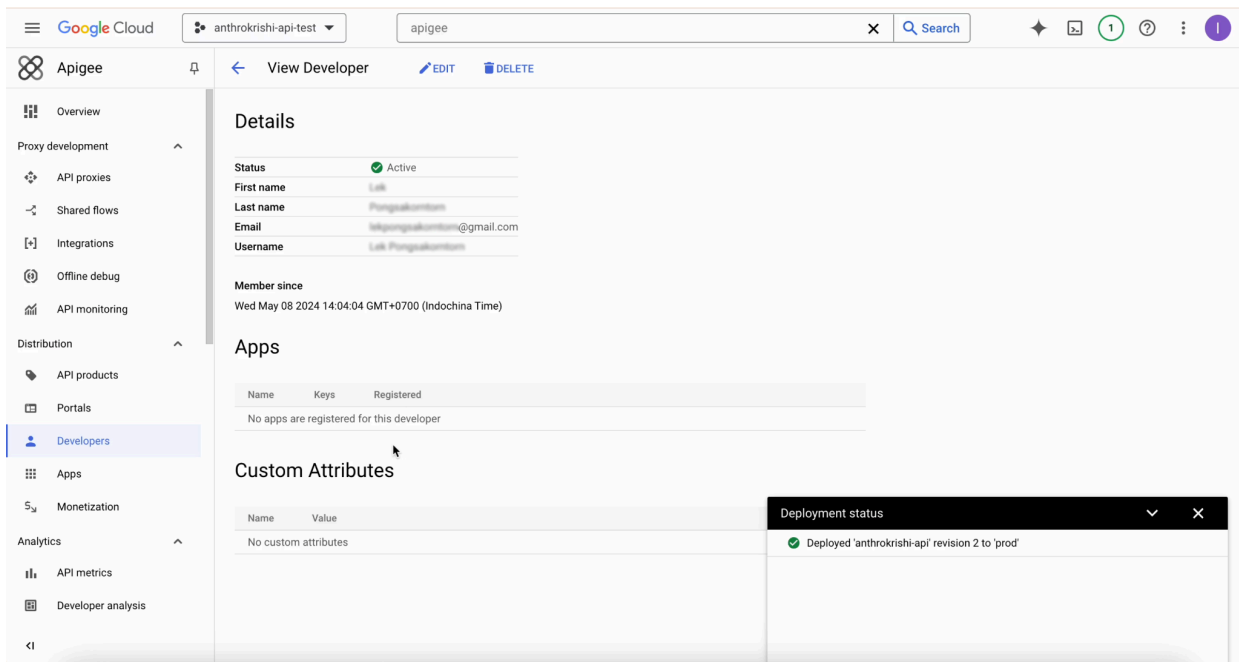
4.2. Add Developer

4.2.1. Navigate to "Developers" in the left-hand navigation menu and click.

- Set the "First Name",
- Set the "Last Name".
- Set the "Email".
- Set the "Username".
- Click **Add**



Add developer information



A developer was added

4.3. Create an API Product

4.3.1. Navigate to "API products" in the left-hand navigation menu and click.

- Set the "Name" to "anthrokrishi-api"
- Set the "Display Name" to "anthrokrishi-api"
- In the "Environment" section, select "prod" and "dev"

The screenshot shows the Apigee console interface. The left-hand navigation menu is open, with 'API products' selected. The main content area displays the 'Product details' configuration page for the product 'anthrokrishi-api'. The form includes the following fields and options:

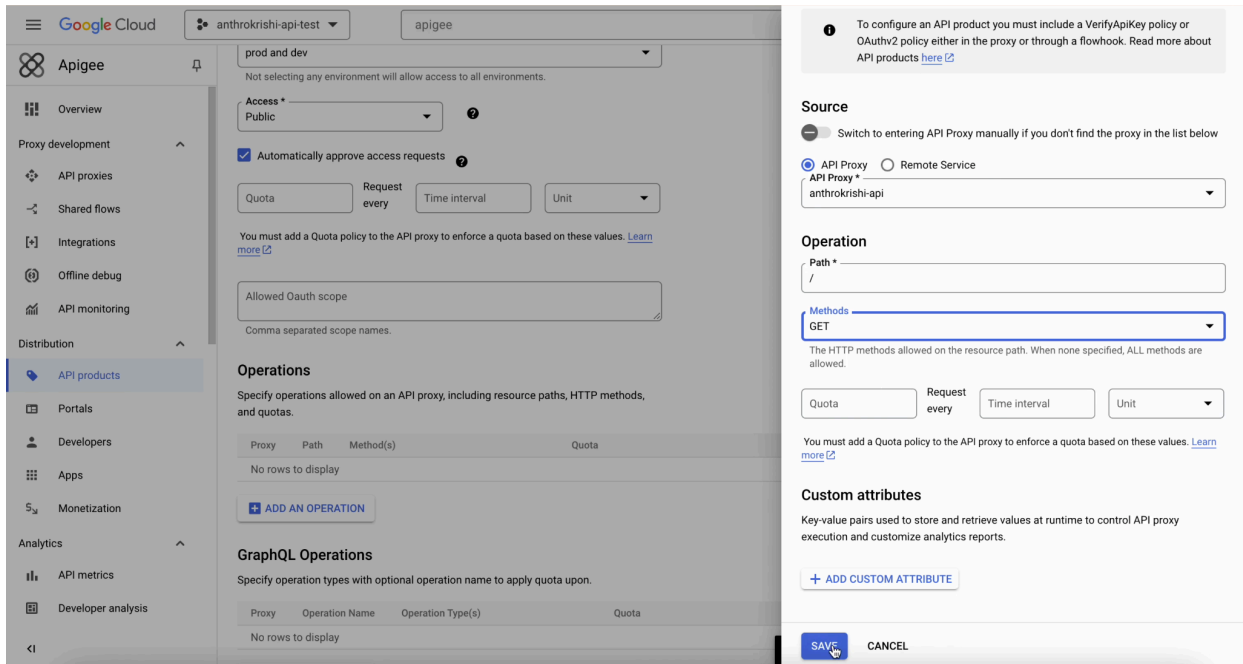
- Name ***: Input field containing 'anthrokrishi-api'.
- Display Name ***: Input field containing 'anthrokrishi-api'.
- Description**: Empty text area.
- Environment**: Dropdown menu with 'prod and dev' selected. Below it, a note states: 'Not selecting any environment will allow access to all environments.'
- Access ***: Dropdown menu with 'Public' selected.
- Automatically approve access requests**
- Quota**: Input field (empty), **Request every**: Input field (empty), **Time interval**: Input field (empty), **Unit**: Dropdown menu (empty).
- A note below the quota fields: 'You must add a Quota policy to the API proxy to enforce a quota based on these values. [Learn more](#)'
- Allowed OAuth scope**: Input field (empty). Below it, a note states: 'Comma separated scope names.'
- Operations**: Section header with a sub-note: 'Specify operations allowed on an API proxy, including resource paths, HTTP methods, and methods.'

A 'Deployment status' window is visible at the bottom right of the console.

Configure the product details.

4.3.2. Add an operations

- Click **ADD AN OPERATION**
- Select an “API Proxy” and choose an “anthrokrishi-api”
- Set the “Path” to “/”
- In the “Methods” section, select “GET”.
- Click **Save**



An operation setting.

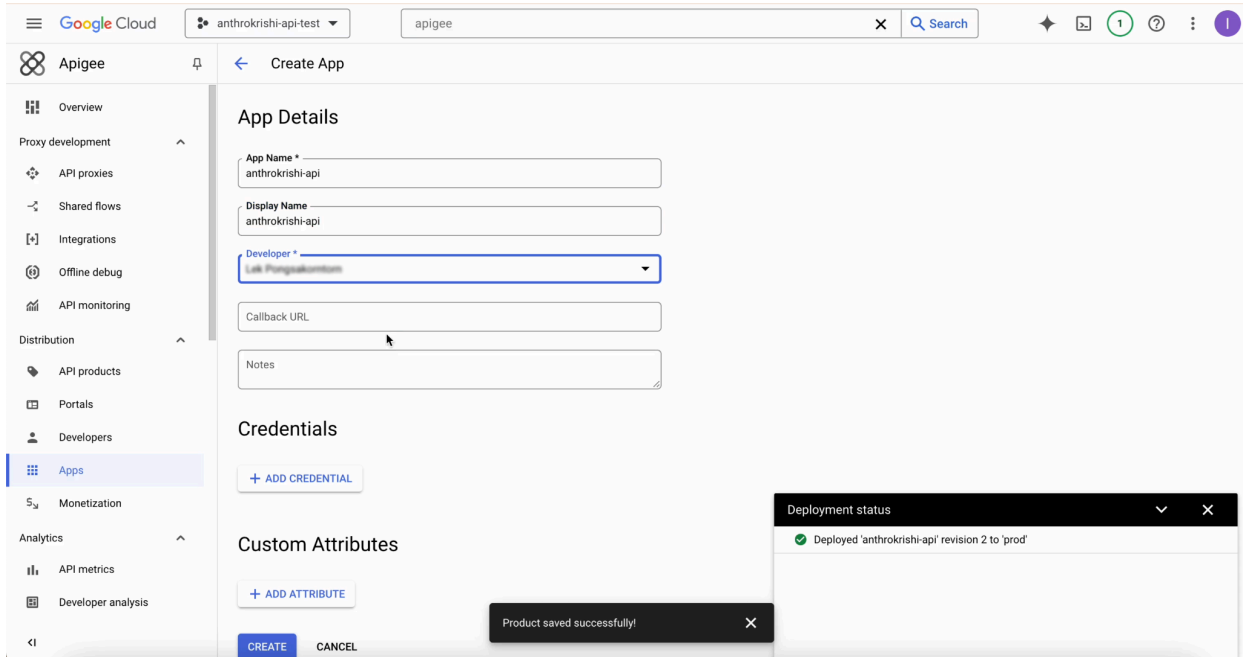
4.4. Generate an API Key

4.4.1. Navigate to the **Apps** in the left-navigation menu and click.

- Click **Create**.

4.4.2. Configure the App details.

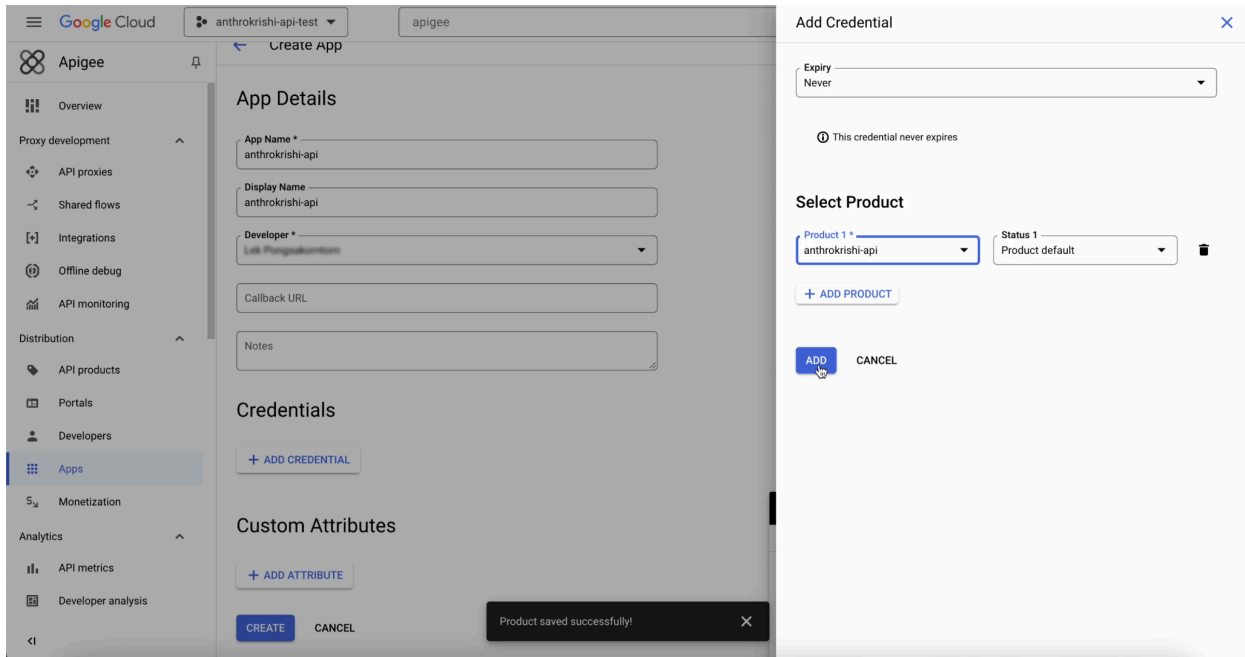
- Set the “App Name” to “anthrokrishi-api”.
- Set the “Display Name” to “anthrokrishi-api”.
- Select the “Developer”.



Configure app details

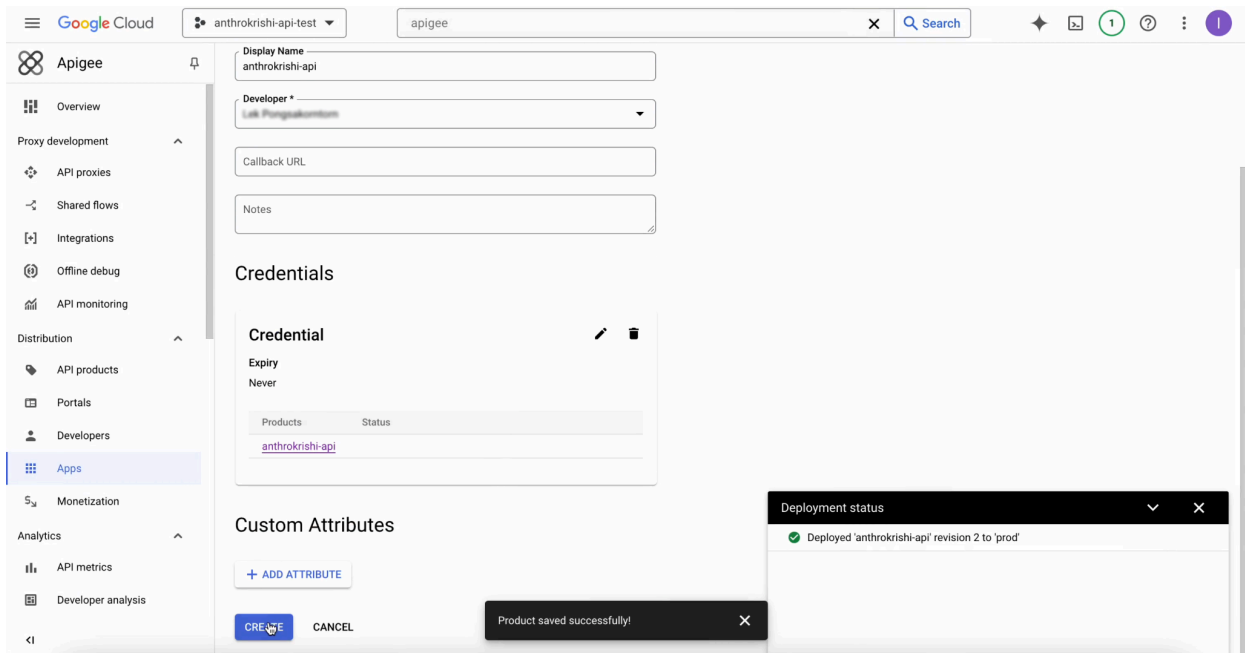
4.4.3. Add Credential (API Key)

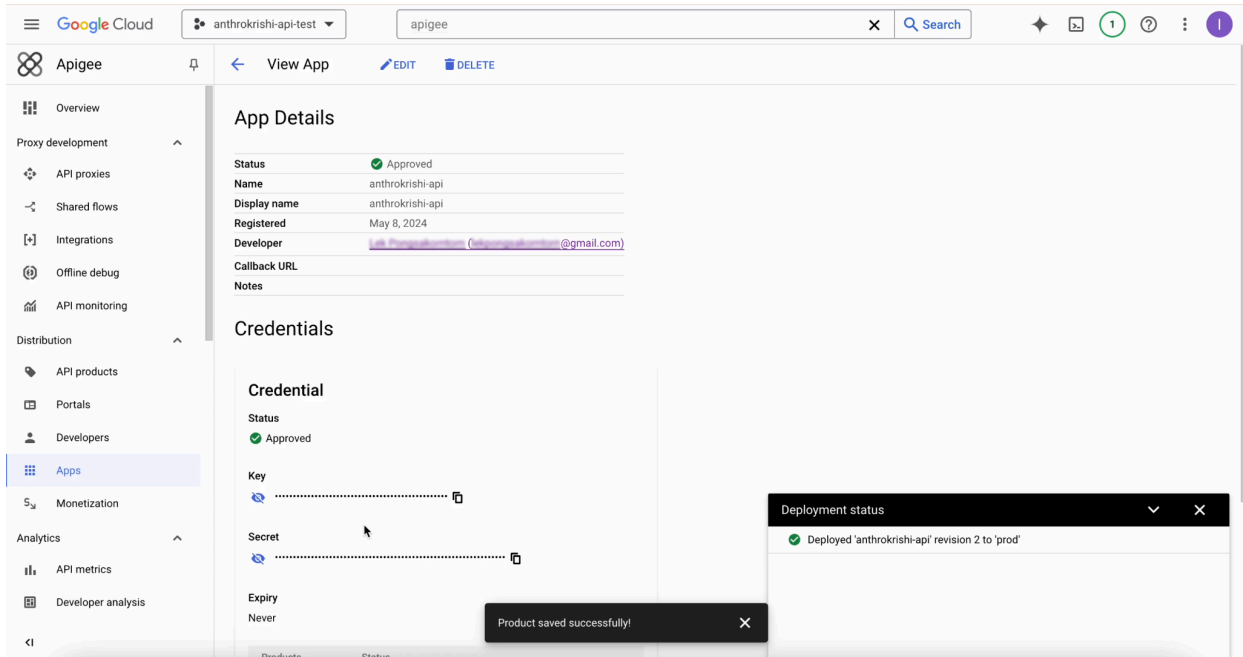
- Click **Add Credentials**.
- Set the “Expiry” to “Never”.
- In the “Select Product” section.
 - Click “Product 1” and choose “anthrokrishi-api” from the dropdown.
 - Click “Status 1” and choose “Product default” from the dropdown.
- Click **Add**.



The Panel to add credential

■ Click Create

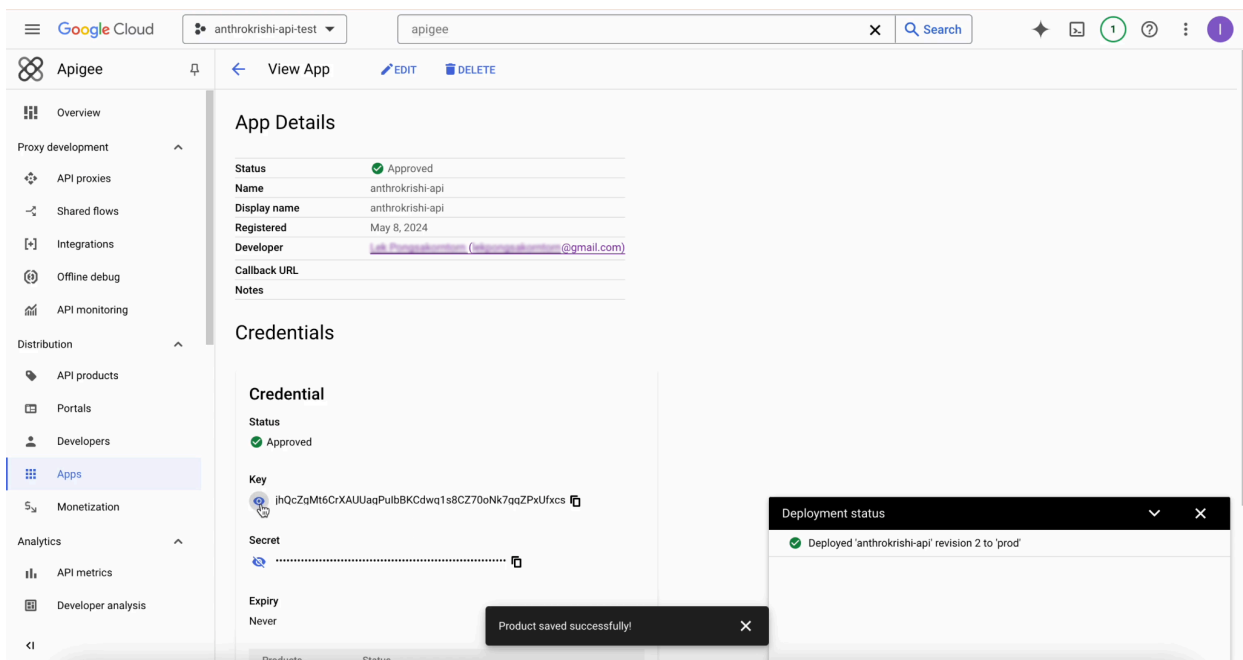




App and credential was created

4.4.4. Get an API Key

- In the **Credentials** section click the **Eye Icon** below the **Key** to get an API Key



An API key within the app e.g.

jhQcZqMt6CrXAUUaqPulbBKCdwq1s8CZ70oNk7qqZPxUfxcs