

Apigee Overview and Set Up of Apigee X

Nandita Sahu · [Follow](#)

15 min read · Oct 1, 2022

31

Q 1



With the rapid transition to online platforms, competition among digital businesses is increasing dramatically. It is no longer enough to just use the API, and be able to manage and scale it. So now is time for Apigee X, which speeds up the process of deployment and scaling and is several times faster than other methods. Fast response and efficiency gives Apigee owners an

advantage when working with API. Apigee's main goal is to simplify the digital experience and improve the interaction with the API.

You have data, you have services, and you want to develop new business solutions quickly, both internally and externally. With Apigee, you can build API proxies — RESTful, HTTP-based APIs that interact with your services. With easy-to-use APIs, developers can be more productive, increasing your speed to market.

What is Apigee?

Apigee is a platform for developing and managing APIs. By fronting services with a proxy layer, Apigee provides an abstraction or facade for your backend service APIs and provides security, rate limiting, quotas, analytics, and more.

Making your services available on the web

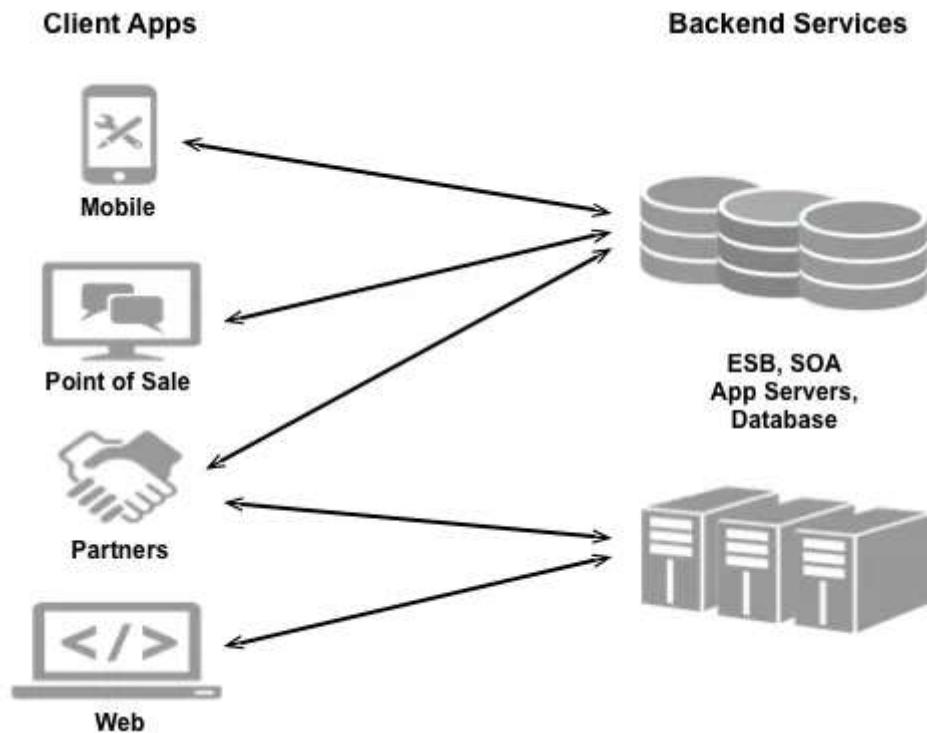
Companies today want to make their backend services available on the web so that these services can be consumed by apps running on mobile devices and desktops. A company might want to expose services that provide product pricing and availability information, sales and ordering services, order tracking services, and any other services required by client apps.

Companies often expose services as a set of HTTP endpoints. Client app developers then make HTTP requests to these endpoints. Depending on the endpoint, the service might then return data, formatted as XML or JSON, back to the client app.

The client apps that consume these services can be implemented as standalone apps for a mobile device or tablet, as HTML5 apps running in a browser, or as any other type of app that can make a request to an HTTP

endpoint and consume any response data. These apps might be developed and released by the same company that exposed the services, or by third-party app developers who make use of publicly available services.

The following image shows this type of model:



Because providers make their services available over the web, they must ensure that they have taken all necessary steps to secure and protect their services from unauthorized access. As a service provider, consider:

- **Security:** How will you control access to your services to prevent unauthorized access?
- **Compatibility:** Will your services work across different platforms and devices?

- **Measurability:** How can you monitor your services to make sure they are available?

And many other considerations

After a client app has been released those accesses any services, the service provider is then required to make sure that those services continue to work overtime as they add, modify, or delete those services. The service provider must also have a way to keep app developers aware of any changes to the services to ensure that client apps stay in sync with those services.

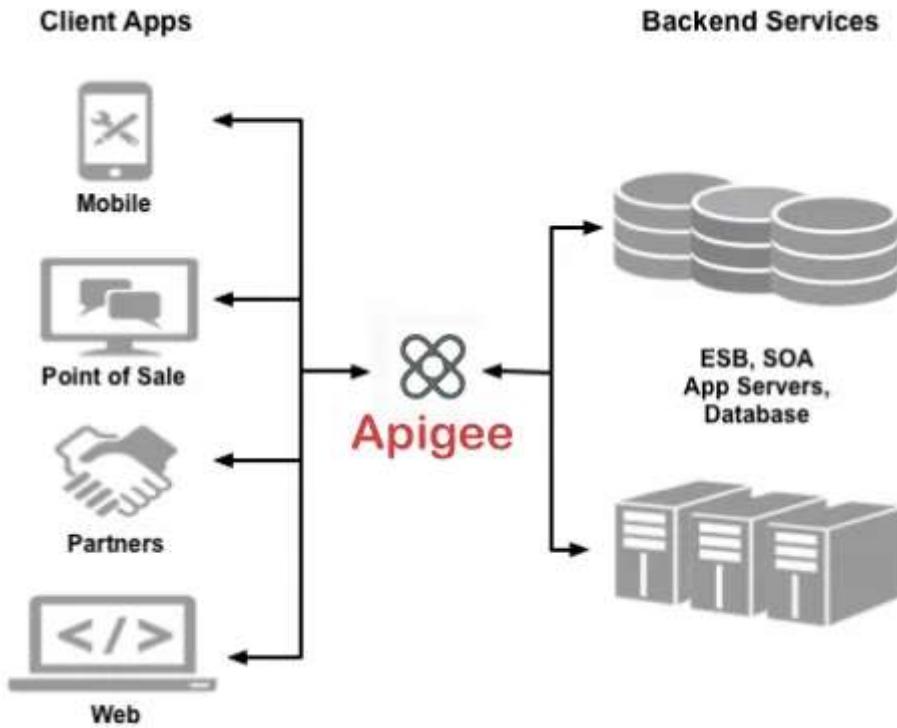
Client app developers face challenges when trying to consume services from different providers. There are many technologies available today for use by a service provider to expose its services. The same client app might have to use one mechanism to consume a service from one provider, and a different mechanism to consume a service from a different provider. App developers can even face the situation where they have to use different mechanisms to consume services from the same provider.

Make services available through Apigee

Apigee enables you to provide secure access to your services with a well-defined API that is consistent across all of your services, regardless of service implementation. A consistent API:

- Makes it easy for app developers to consume your services.
- Enables you to change the backend service implementation without affecting the public API.
- Enables you to take advantage of the analytics, developer portal, and other features built into Apigee.

The following image shows an architecture with Apigee handling the requests from client apps to your backend services:



Rather than having app developers consume your services directly, they access an API proxy created on Apigee. The API proxy functions as a mapping of a publicly available HTTP endpoint to your backend service. By creating an API proxy you let Apigee handle the security and authorization tasks required to protect your services, as well as to analyze and monitor those services.

Because app developers make HTTP requests to an API proxy, rather than directly to your services, developers do not need to know anything about the implementation of your services. All the developer needs to know is:

- The URL of the API proxy endpoint.
- Any query parameters, headers, or body parameters passed in a request.

- Any required authentication and authorization credentials.
- The format of the response, including the response data format, such as XML or JSON.

The API proxy isolates the app developer from your backend service. Therefore you are free to change the service implementation as long as the public API remains consistent. For example, you can change a database implementation, move your services to a new host, or make any other changes to the service implementation. By maintaining a consistent frontend API, existing client apps will continue to work regardless of changes on the backend.

You can use policies on the API proxy to add functionality to a service without having to make any changes to the backend service. For example, you can add policies to your proxy to perform data transformations and filtering, add security, execute conditional logic or custom code, and to perform many other actions. The important thing to remember is you implement policies on Apigee, not on your backend server.

Why do you need Apigee?

While exposing the APIs to the public, you may face challenges like:

1. Securing your APIs from OWASP threats.
2. Enforcing DDoS protection, and OAuth and JWT based access control.
3. Generating rate-plans and monetizing APIs.
4. Generating business use-case related analytics reports.
5. Global Reach and Performance

6. Privacy and Compliance

Apigee can perform these activities with minimal configuration. This drastically reduces the development effort required to make the APIs public ready.



The backend APIs are deployed in Google Cloud, Azure and AWS with each cloud deployment having a Load Balancer in front. Apigee can be placed in front of the Load Balancers. Apigee is connected to the backend APIs over the public internet. To ensure authentication, Mutual TLS is enabled between Apigee and the Load Balancers.

Comparison between Apigee products

Product	Where hosted	Managed by
Apigee Edge for Public Cloud 	Apigee's cloud	Apigee
Apigee Edge for Private Cloud 	The customer's private data center.	Customer
Apigee X	Google Cloud	Apigee
Apigee hybrid	Both Google Cloud and the customer's private data center	Apigee manages the management plane and the customer manages the runtime plane.

Apigee Edge

- Apigee Edge is the public cloud, SaaS (Software as a Service) version of Apigee.
- Both the runtime and the management plane are in the cloud, and not under your control.
- No direct networking setup between Apigee cloud and your VPC (Virtual Private Cloud).
- All connections to Apigee, and Apigee to the backend services are over the public internet.
- Apigee Edge can reside in front of any cloud provider or on-prem service.

Apigee X

- Apigee X is the latest variant released by Google.
- Like Apigee Edge, in Apigee X, both the management plane and the runtime are managed by GCP (Google Cloud Platform).

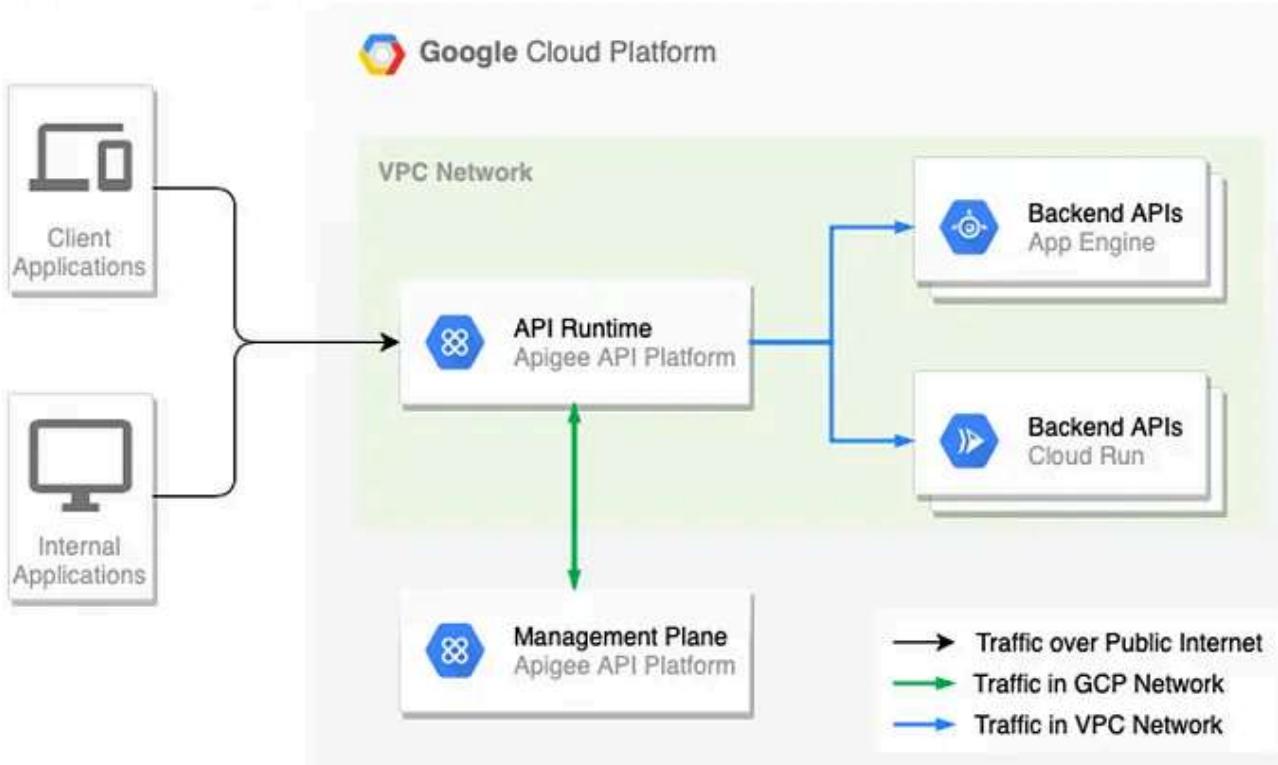
- The Apigee runtime resides in your VPC Network, and a GCP HTTP(s) Load Balancer serves as the entry point into Apigee.

Apigee Hybrid

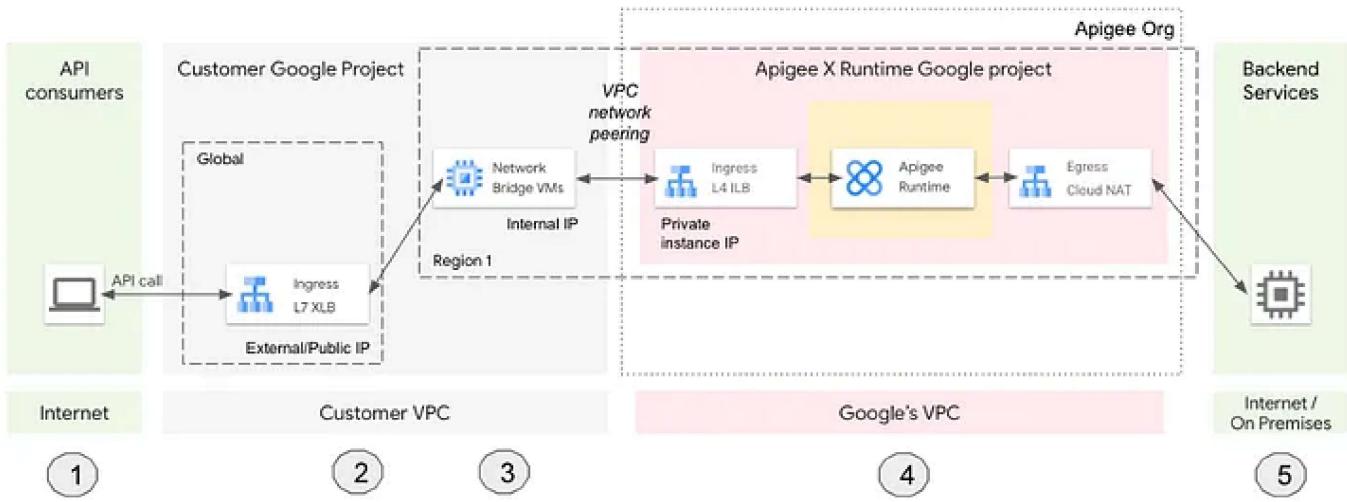
- In Apigee Hybrid you control the runtime in any cloud providers.
- Since Anthos GKE cluster can be deployed on multiple cloud providers, Apigee Hybrid can also be installed in different providers.

Instead of public internet, which is used by Apigee Edge, we can use Apigee X or Apigee Hybrid which reside in the same network as your services thus adding security and reducing latency.

Apigee X & Hybrid Use Case



Apigee architecture



The following illustration shows the lifecycle of an API proxy call as it moves through the provisioned Apigee system components:

- A client app calls an Apigee API proxy.
- The request lands on a global L7 external HTTPS load balancer (XLB). The XLB is configured with an external/public IP and a TLS certificate.
- The XLB sends the request to a virtual machine (VM). The VM serves as a bridge between your VPC and Google's VPC (managed by Apigee).
- The VM sends the request to Apigee, which processes the API proxy request.
- Apigee sends the request to the backend service, and the response is sent back to the client.

To get a better overview of Apigee X Architecture : [Click Here](#)

Steps for Set up of Apigee X

1. Some of the prerequisite — Before setting up Apigee

Set up — Google Cloud billing account, create a new google cloud account and install gcloud cli and curl.

Prerequisites

[Send feedback](#)[Version 1.8 \(latest\) ▾](#)

Before you can install Apigee hybrid, you must meet the following prerequisites:

Done?	Prerequisite	Description
<input type="checkbox"/>	Google Cloud billing account	You must have a Google Cloud account with active billing. For more information, see Create a new billing account .
<input type="checkbox"/>	Google Cloud project	Create a new Google Cloud project for the Apigee hybrid installation. For more information, see Creating a project . If you'd prefer to use Cloud Identity to manage your organizational identity with Apigee, see Cloud Identity Help . Although not needed for Apigee hybrid, Cloud Identity might be useful if you plan to use other Google Cloud services or have multiple members of your organization access the product.
<input type="checkbox"/>	Software	Install the following software on your administrative machine: <ul style="list-style-type: none">• gcloud CLI (required for command line provisioning steps).<ul style="list-style-type: none">• After the gcloud CLI is installed, run <code>gcloud components update</code> to get the latest gcloud components.• curl (recommended)

I have created a project named as Apigee

⚠ You have 14 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)

MANAGE QUOTAS

Project name * Apigee ?

Project ID: apigee-364214. It cannot be changed later. [EDIT](#)

Location * No organization [BROWSE](#)

Parent organization or folder

CREATE **CANCEL**

2. Then you add your project to your billing account. Go to billing → Manage Billing Accounts → Add your project to your billing account.

Billing				
MY BILLING ACCOUNTS		MY PROJECTS		
<input type="text"/> Filter Enter property name or value				
Name	ID	Billing account ↑	Billing account ID	Actions
Apigee	apigee-364214	My Billing Account	013538D55484-B9118C	⋮
practice	plasma-figure-361706	My Billing Account	013538D55484-B9118C	⋮
Apigee-Terraform	apigee-terraform-364110	Billing is disabled	—	⋮
final	final-362308	Billing is disabled	—	⋮

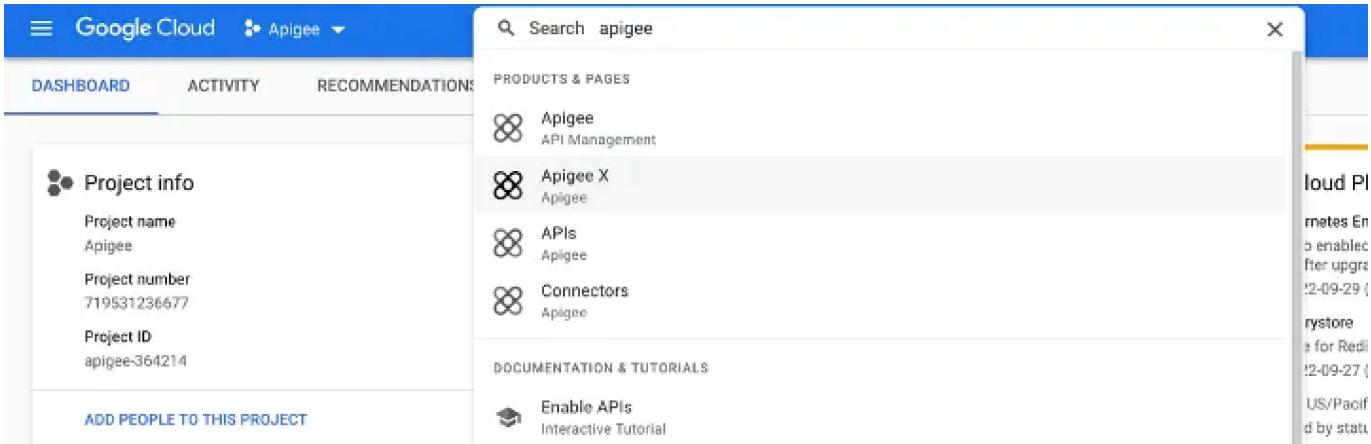
3. Enable these APIs in your Apigee project

\$ export PROJECT_ID=YOUR_PROJECT_ID

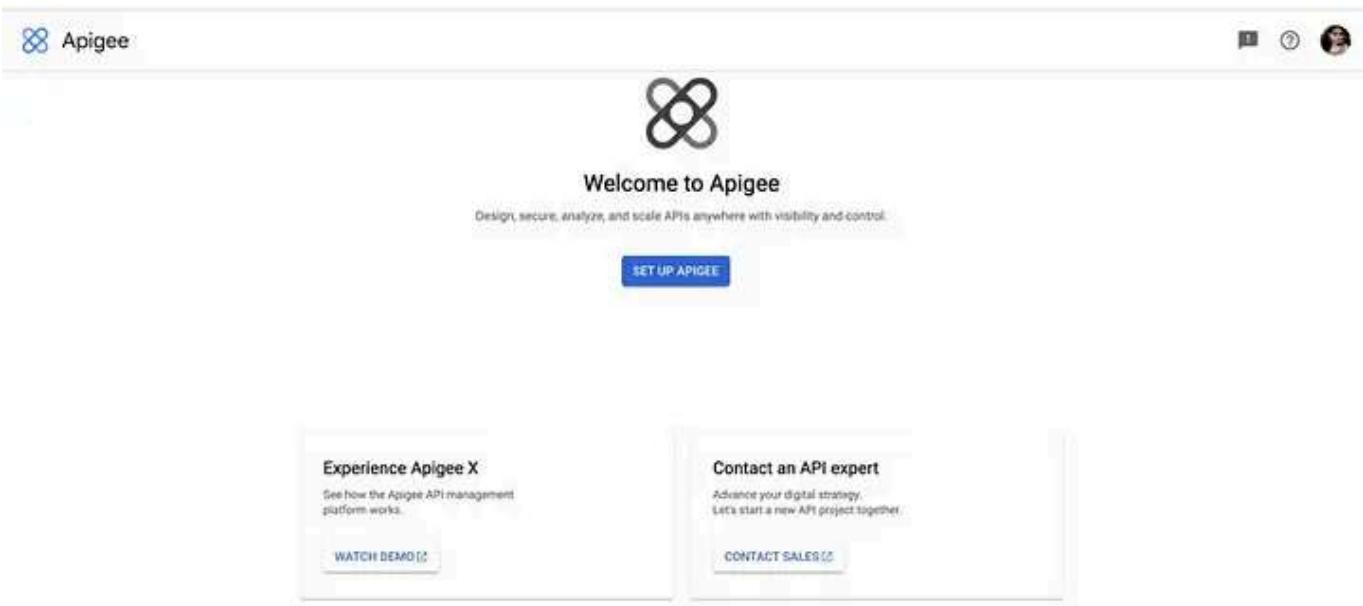
```
$ gcloud services enable \
apigee.googleapis.com \
apigeeconnect.googleapis.com \
cloudresourcemanager.googleapis.com \
compute.googleapis.com \
container.googleapis.com \
pubsub.googleapis.com --project $PROJECT_ID
```

```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to apigee-364214.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
haidevops@cloudshell:~ (apigee-364214)$ export PROJECT_ID=apigee-364214
haidevops@cloudshell:~ (apigee-364214)$ gcloud services enable \
    apigee.googleapis.com \
    apigeeconnect.googleapis.com \
    cloudresourcemanager.googleapis.com \
    compute.googleapis.com \
    container.googleapis.com \
    pubsub.googleapis.com --project $PROJECT_ID
Operation "operations/acf.p2-719531236677-e9f4fe10-a9b1-43eb-9245-0527e95bd4bc" finished successfully.
```

4. Go to the Apigee X Page → Search on Bar



This is the Apigee Dashboard



5. Set Up Apigee for your project – Choose Evaluation plan for Apigee free trial for 60 days or Pay as you go plan. For this demo we are going through Pay as you plan

Set up Apigee

Configure [Apigee X](#) to start using it with your Google Cloud (GCP) projects.

Learn more about [Apigee](#).

Project *
apigee-364214

Enter a GCP project ID. If you have a GCP project with an Apigee entitlement, enter it.
[Create a new GCP project](#)

Apigee subscription is not enabled for this GCP project. If you have purchased a subscription, please contact your sales rep.

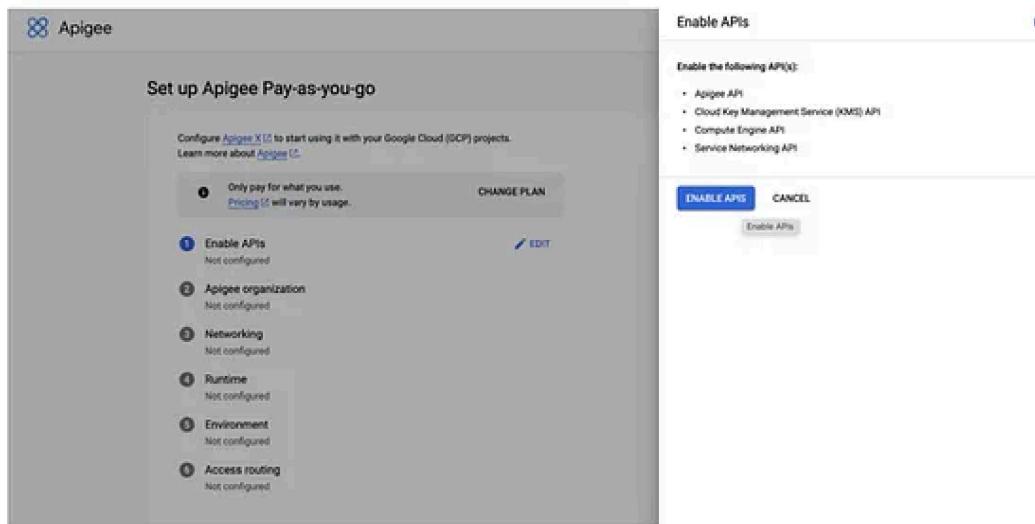
Choose a pricing plan:

- Evaluation**
Try out the powerful features in your own sandbox. [Learn more](#)
- Apigee Subscription**
Build and scale your enterprise-wide API programs. [Learn more](#)
- Pay-as-you-go**
No entitlement required. Pay only for what you use, as you use it. [Learn more](#)

START EVALUATION **CONTACT SALES** **USE PAY-AS-YOU-GO**

6. Enable these APIs in your project for Apigee X → Apigee X , Cloud Key Management Service, Service Networking APIs

Enabling the APIs



To use Apigee, you must enable the following APIs for your Cloud project:

API Name	Location	Description
Apigee	apigee.googleapis.com	Provides communications between your project and other services and Cloud APIs.
Service Networking	servicenetworking.googleapis.com	Enables your managed services on internal IP addresses to connect to service consumers.
Compute Engine	compute.googleapis.com	Used for managing your Apigee runtime.
Cloud Key Management Service (KMS)	cloudkms.googleapis.com	Manages keys and performs cryptographic operations for direct use by other Cloud resources.

Set up Apigee Pay-as-you-go

Configure [Apigee X](#) to start using it with your Google Cloud (GCP) projects.
Learn more about [Apigee](#).

1 Only pay for what you use.
[Pricing](#) will vary by usage.

[CHANGE PLAN](#)

 **Enable APIs**

Enabled: Apigee API, Cloud Key Management Service (KMS) API,
Compute Engine API, Service Networking API

2 **Apigee organization**

Not configured

 [EDIT](#)

3 **Networking**

Not configured

4 **Runtime**

Not configured

5 **Environment**

Not configured

6 **Access routing**

Not configured

7. After enabling APIs, create an Apigee Organization. Remember that the project ID and org name are always the same.

To create a new organization in the Apigee provisioning wizard:

- If it is not currently open, open the [Apigee provisioning wizard](#). The wizard returns to the next incomplete install task.
- In the wizard, click **Edit** next to **Apigee organization**:

- The Create an Apigee organization view displays:

The screenshot shows two overlapping windows. On the left is the 'Set up Apigee Pay-as-you-go' dashboard, which includes a plan selection section ('Only pay for what you use.'), a list of configuration steps (Enable APIs, Apigee organization, Networking, Runtime, Environment, Access routing), and a 'CHANGE PLAN' button. On the right is the 'Create an Apigee organization' dialog box. It has fields for 'Apigee organization name' (set to 'apigee-364214') and 'Analytics hosting region' (set to 'us-central1'). Below these are sections for 'Runtime database encryption key' (warning about SLA) and 'Choose a Cloud KMS location' (set to 'us-central1'). A third section for 'Choose a customer-managed encryption key' shows 'No valid keys found'. At the bottom are 'CREATE KEY', 'DON'T SEE YOUR KEY? ENTER KEY RESOURCE ID.', 'CANCEL', and 'OK' buttons.

From the **Analytics hosting region** drop-down list, select the physical location where you want your analytics data stored. For example: us-central1.

Under **Runtime database encryption key** make the following selections:

- Choose a Cloud KMS location. The dropdown menu lists two groupings of locations: multi-regional and dual-regional are grouped together, and regional locations are in another grouping. Note that if you select a regional location, we cannot offer an SLA higher than 99.9%.
- Next, choose a customer-managed encryption key. If a key already exists in the KMS location you selected, you can pick it.

Create a new key

Complete the following steps to create a [customer-managed encryption key](#) (CMEK) within Cloud KMS.

1 Key ring

Select a key ring for your new key

A key ring organizes keys in a specific Google Cloud location and allows you to manage access control on groups of keys. [Learn more](#)



Create key ring

Key ring name *

apigee-key

Key ring names can contain letters, numbers, underscores (_), and hyphens (-). Key rings can't be renamed or deleted.

Key ring location *

us-central1

[CONTINUE](#)

2 Key

Define a symmetric encryption key used to encrypt and decrypt data

3 Review

Confirm details for key creation

[CANCEL](#)

[CREATE](#)

Create a new key

Complete the following steps to create a [customer-managed encryption key](#) (CMEK) within Cloud KMS.

1 Key ring

Select a key ring for your new key

Name apigee-key
Location us-central1

2 Key

Define a symmetric encryption key used to encrypt and decrypt data

The key material will be generated for you and automatically rotated every 90 days.

After creation, you can change the rotation schedule. [Learn more](#)

Key name *

apigee-key-encryption

Key names can contain letters, numbers, underscores (_), and hyphens (-). Keys can't be renamed or deleted.

Protection level *

Software

Protection level determines how cryptographic operations are performed. [Learn more](#)

[CONTINUE](#)

3 Review

Confirm details for key creation

[CANCEL](#)

[CREATE](#)

Create a new key

Complete the following steps to create a [customer-managed encryption key](#) (CMEK) within Cloud KMS.

Key ring

Select a key ring for your new key

Name	apigee-key
Location	us-central1

Key

Define a symmetric encryption key used to encrypt and decrypt data

Name	apigee-key-encryption
------	-----------------------

3 Review

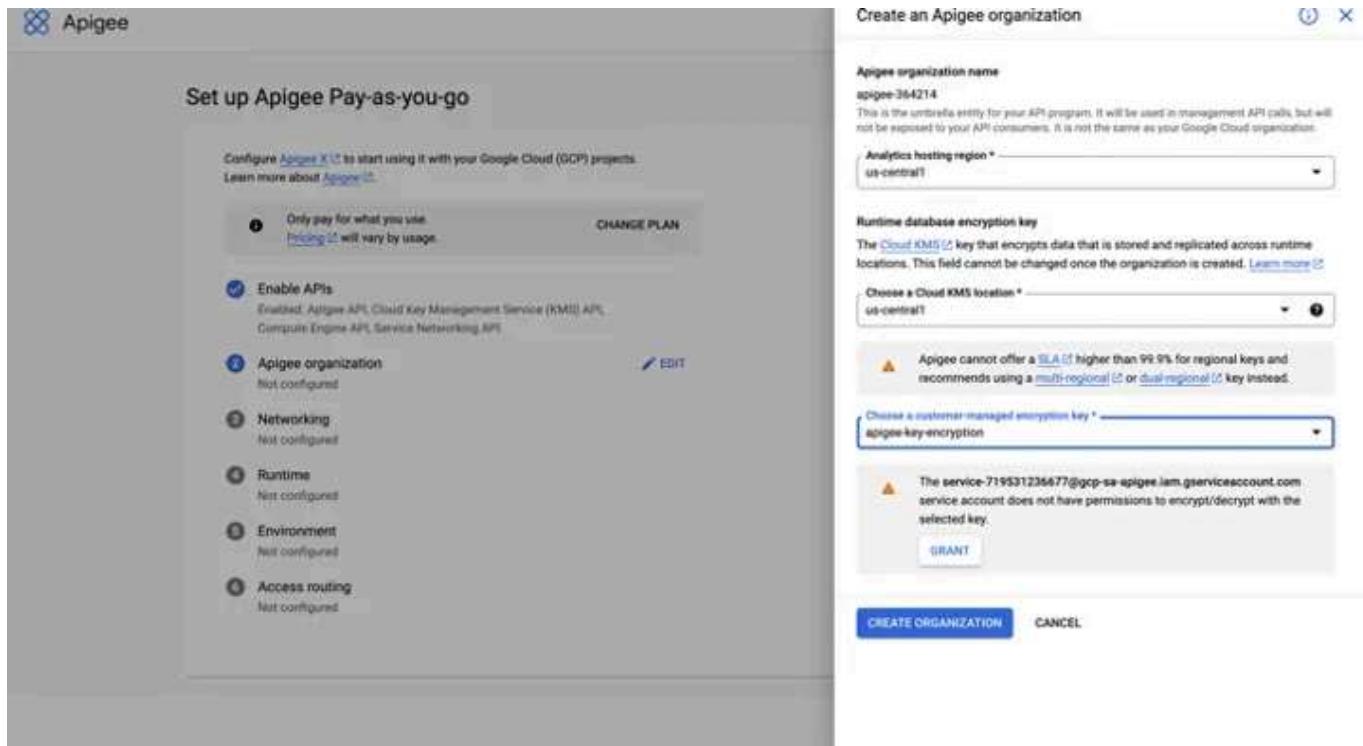
Confirm details for key creation

The key and key ring will be created in project **apigee-364214**. Before submitting, you may reopen any of the previous steps to review or make changes.

Keys and key rings cannot be deleted after creation. [Learn more](#)

[CANCEL](#) [CREATE](#)

Click **Grant** to grant the service account permission to encrypt/decrypt with the selected key and then create Organization.



8. Service Networking automates the private connectivity setup (using VPC Network Peering) between your network and Apigee.

Select your VPC network from the **Authorized network** drop-down list. If you have not created a network, select the **default VPC**, which is created for all Cloud projects.

In the **Reserve peering ranges** section, choose one of these options:

- **Automatically allocate IP ranges:** Choose this option if you want Apigee to manage the IP range for you. We create two ranges, one of length /22 and one of length /28.

Set up networking



The Apigee runtime uses an IP range from your local VPC network. Select a network to reserve an IP range and to create a private connection for Google services. [See VPC networks in your project](#)

Authorized network *

default



Reserve Peering Ranges *

Automatically allocate IP range

Google Cloud Platform will automatically allocate IP ranges of prefix lengths /22 and /28 with names in format: "google-managed-services-default-[prefix-length]".



Not recommended if using dynamic routes (such as Cloud Router) or a network from a host project.

Select one or more existing IP ranges or create a new one

Select or create an IP range



ALLOCATE AND CONNECT

CANCEL

9. Create a Apigee Runtime Instance — An instance, or *runtime instance*, is where your project and related services are stored; it provides the user-facing endpoint for your services.

The screenshot shows the Apigee Pay-as-you-go setup interface. On the left, there's a sidebar with several configuration items: Only pay for what you use (with a note about pricing), Enable APIs (listing Cloud Key Management Service (KMS) API, Compute Engine API, Service Networking API), Apigee organization (Org name: apigee-354214, Analytics region: us-central1), Networking (Network: default), Runtime (Not configured), Environment (Not configured), and Access routing (Not configured). On the right, a modal window titled 'Set up runtime' is open. It asks 'Let's set up your first runtime region.' and shows a dropdown for 'Runtime hosting region' set to 'us-central1'. Below it, 'IP range allocation' is set to 'Automatic (Recommended)' (Apigee automatically selects an available range with prefix size /22). There's also a 'Disk encryption key' section where a customer-managed encryption key named 'apigee-key-encryption' is selected. At the bottom of the modal are 'CREATE RUNTIME' and 'CANCEL' buttons.

Set up runtime region — select the region in which you want your instance hosted.

Specify how you want to allocate an IP range. Choose between these options:

- **Automatic (Recommended)** — Apigee selects an available CIDR range with a prefix size of /22. No further action on your part is required.
- **Custom** — In advanced use cases, you may need to specify exactly which IP range you want Apigee to use. You must specify a custom IP range, and it must have a prefix size of /22. The range must be available as part of a private connection between your project and Apigee

Under **Disk encryption key**, choose a customer-managed encryption key. If a key already exists, you can pick it.

This request can take 40 minutes or longer to complete because Apigee creates the new instance, installs the Apigee resources on it, and sets up load balancing.

Set up Apigee Pay-as-you-go

Configure [Apigee X](#) to start using it with your Google Cloud (GCP) projects.
Learn more about [Apigee](#).

Only pay for what you use. [Pricing](#) will vary by usage. [CHANGE PLAN](#)

- Enable APIs**
Enabled: Apigee API, Cloud Key Management Service (KMS) API, Compute Engine API, Service Networking API
- Apigee organization**
Org name: apigee-364214, Analytics region: us-central1
- Networking**
Network: default
- Runtime**
Runtime location: us-central1, Internal IP: 10.5.188.2
- 5 Environment** [!\[\]\(8049c40dca7516f2acb55efdbab1427a_img.jpg\) EDIT](#)
Not configured
- Access routing**

10. Now you can create a new environment and attach it to a new environment group.

An *environment group* is a logical grouping of environments. You define your hostnames on an environment group rather than individual environments so that they can be shared. A service on the group called the *ingress* redirects

requests to different environments within the group based on the hostnames assigned to the group.

The screenshot shows the 'Set up Apigee Pay-as-you-go' page. On the left, there's a sidebar with various configuration sections like 'Enable APIs', 'Apigee organization', 'Networking', 'Runtime', 'Environment', and 'Access routing'. A note says 'Configure Apigee X to start using it with your Google Cloud (GCP) projects.' To the right, there's a diagram illustrating environment setup. It shows three environment groups: 'Development' (apigee-dev.appspot.com), 'Test' (apigee-test.appspot.com), and 'Production' (apigee.apigee.org). Each group contains three environments: 'default-dev', 'payment-dev', and 'catalog-dev' for development; 'default-test', 'payment-test', and 'catalog-test' for test; and 'default-prod', 'payment-prod', and 'catalog-prod' for production. Below the diagram, text explains that all environments in a group share the same hostname, and there's a section for creating a first environment group named 'dev-group' with a specific hostname 'apigee.nanditadevops.cloud'.

- In the **Environment group name** field, specify a name for your environment group. For example, dev-group. You cannot change the name of an environment group once it has been created.
- In the **Environment group hostname** field, specify the hostname that routes requests into this group. For example: apigee.nanditadevops.cloud
- You can specify a single hostname only in this field.
- In the **Environment name** field, specify a name for your environment. For example, dev. You cannot change the name of an environment once it has been created.

- Click **Create & attach** to create the new environment and attach it to the new environment group.

The screenshot shows the Apigee X interface. At the top, there's a header with the title 'dev-group'. Below the header, there are two sections: 'Hostnames' containing 'apigee.nanditadevops.cloud' and 'Environments' containing 'dev-env'. The main area is titled 'About' and contains the following configuration details:

Ready for deployment	dev-env	Display Name	dev-env
Name	dev-env	Description	<no description set>
Deployment type	Proxy		
Current aggregate node count	2		
Minimum nodes	2		
Maximum nodes	<not set>		

11. In the last step you need to choose whether to expose your new cluster to external requests or to keep it private (and only allow requests from within the firewall).

The screenshot shows the Apigee Pay-as-you-go setup interface. On the left, a sidebar lists configuration steps: Only pay for what you use, Enable APIs, Apigee organization, Networking, Runtime, Environment, and Access routing. The 'Enable APIs' step is expanded, showing sub-options like Apigee API, Cloud Key Management Service (KMS) API, Compute Engine API, and Service Networking API. The 'Access routing' step is marked as 'Not configured'. On the right, a detailed configuration panel titled 'Configure access to the "dev-group" env group' is displayed. It asks 'Should "dev-group" be accessible from the Internet?' with two options: 'No internet access' (selected) and 'Enable internet access'. Under 'Enable internet access', it explains that an HTTPS load balancer will be created and assigned a static external IP address, which will point your selected domain to all regions. A diagram illustrates the architecture: a Load Balancer (https://apigee-handlertwpe.cloud) connects to a Compute Engine VM instance group (us-central1 runtime, 10.6.188.2) and a Compute Engine VM instance group (Future regions). Below this, there's a section for SSL Certificate (for apigee.nanditadevops.cloud), with options for self-managed or Google-managed certificates.

No Internet Access — Your APIs will only be accessible from your internal network. Each of the regions will be using different internal IP Address. This IP address is the internal access point for all requests. You will send a request to this IP address from a machine that is also inside the VPC.

Enable Internet Access — An HTTPS Load Balancer will be assigned a static external IP address. The load balancer will point your selected domain to all regions. Apigee creates a managed instance group (MIG) containing multiple VMs to proxy traffic between the load balancer and the Apigee runtime.

VM Instances								
Filter	Enter property name or value							
Status	Name	Zone	Recommendations	In-use by	Internal IP	External IP	Connect	
<input type="checkbox"/>	green apigee-proxy-uch0	us-central1-f		apigee-proxy-us-central1	10.128.0.3 (nic0)		SSH	⋮
<input type="checkbox"/>	green apigee-proxy-wz17	us-central1-c		apigee-proxy-us-central1	10.128.0.2 (nic0)		SSH	⋮

The screenshot shows the Apigee X interface for managing proxy configurations. On the left, a sidebar lists various services: Load balancing, Cloud DNS, Cloud CDN, Cloud NAT, Traffic Director, Service Directory, Cloud Domains, and Private Service Connect. The main area is titled 'apigee-proxy-url-map'.

Frontend:

Protocol	IP:Port	Certificate	SSL Policy	Network Tier
HTTPS	34.120.137.104:443	apigee-ssl-cert	GCP default	Premium

Host and path rules:

Hosts	Paths	Backend
All unmatched (default)	All unmatched (default)	apigee-proxy-backend

Backend:

Backend services:

1. apigee-proxy-backend

Endpoint protocol	Named port	Timeout	Health check	Cloud CDN	Logging
HTTPS	https	302 seconds	hc-apigee-proxy-443	Disabled	Disabled

Marketplace

Whole Set Up Process for Apigee

Set up Apigee Pay-as-you-go

Configure [Apigee X](#) to start using it with your Google Cloud (GCP) projects.
Learn more about [Apigee](#).

Only pay for what you use.
[Pricing](#) will vary by usage.

[CHANGE PLAN](#)

Enable APIs

Enabled: Apigee API, Cloud Key Management Service (KMS) API, Compute Engine API, Service Networking API

Apigee organization

Org name: apigee-364214, Analytics region: us-central1

Networking

Network: default

Runtime

Runtime location: us-central1, Internal IP: 10.5.188.2

Environment

Group: dev-group, Environment: dev-env

Access routing

Internet access enabled. Domain: apigee.nanditadevops.cloud

[CONTINUE](#)

12. Copy the External IP address for the domain name and add it to the A record in Cloud DNS — Create an A record that points apigee.nanditadevops.cloud to 34.120.137.104

Set up Apigee Pay-as-you-go



Congratulations! Your Apigee organization is ready.

Recommended next steps

1. Configure DNS

Create an A record that points `apigee.nanditadevops.cloud` to `34.120.137.104`

2. Test your Apigee runtime end to end

Deploy a 'Hello world' API to the dev-env environment and call the API. [Learn more](#)

[DEPLOY API](#)

3. Add developers and other collaborators

Use [GCP IAM](#) to grant permissions to manage your API program

All done?

[OPEN APIGEE CONSOLE](#)

For creating zone in Cloud DNS – Enable Cloud DNS API



Cloud DNS API

[Google Enterprise API](#)

Highly Available Global DNS Network

[ENABLE](#)

[TRY THIS API](#)

[OVERVIEW](#)

[PRICING](#)

[DOCUMENTATION](#)

13. Create a zone with the below details — Make it public or private , put a zone-name as per your choice and add your domain name that you have purchased and click create

A DNS zone is a container of DNS records for the same DNS name suffix. In Cloud DNS, all records in a managed zone are hosted on the same set of Google-operated authoritative name servers. [Learn more](#)

If you don't have a domain yet, purchase one through [Cloud Domains](#).

Zone type [?](#)

- Private
 Public

Zone name *

nginx-apigee



Example: example-zone-name

DNS name *

nanditadevops.cloud



Example: myzone.example.com

DNSSEC *

Off



Description

Cloud Logging [?](#)

- On
 Off

After creating your zone, you can add resource record sets and modify the networks your zone is visible on.

CREATE

CANCEL

Add the nameserver records from Cloud DNS to your Go daddy account.

Edit nameservers

Enter My Own Nameservers

Changing nameservers is risky, and change could potentially lead to your website disappearing from public view.

ns-cloud-c1.googledomains.com	
ns-cloud-c2.googledomains.com	
ns-cloud-c3.googledomains.com	
ns-cloud-c4.googledomains.com	

 [Add Nameserver](#)

14. Create an A record that points apigee.nanditadevops.cloud to 34.120.137.104 in your Cloud DNS zone.

Create record set

DNS Name — ?

Resource Record Type — ?

TTL * — ?

TTL Unit — ?

Routing Policy

Default record type

Weighted Round Robin

Geo-Based

IPv4 Address ?

IPv4 Address 1 * ? trash

Example: 192.0.2.91

+ ADD ITEM

CREATE CANCEL

EQUIVALENT COMMAND LINE ▼

15. Created a Backend Server Nginx in one of compute engine.

Name * ?

Labels ?

Region * ?
 Region is permanent

Zone * ?
 Zone is permanent

Machine configuration

Machine family

Machine types for common workloads, optimized for cost and flexibility

Series ▼

Powered by Intel Skylake CPU platform or one of its predecessors

Machine type ▼



vCPU

1

Memory

3.75 GB

▼ CPU PLATFORM AND GPU

Display device

Enable to use screen capturing and recording tools.

Enable display device

<input type="checkbox"/>	Status	Name ↑	Zone	Recommendations	In use by	Internal IP	External IP	Connect
<input checked="" type="checkbox"/>	Green	apigee-instance	us-central1-a			10.128.0.5 (nic0)	34.135.7.234 (nic0)	SSH ▼ ⋮

Installed Nginx in Compute Engine

```
haidevops@apigee-instance:~$ sudo apt install nginx -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  fontconfig-config fonts-dejavu-core geoip-database libdeflate0 libfontconfig1 libgd3 libgeoip1 libicu67 libjbig0 libjbig20 libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream libnginx-mod-streaming libxml2 libxslt1.1 nginx-common nginx-core
Suggested packages:
  libgd-tools geoip-bin fcgiwrap nginx-doc ssl-cert
The following NEW packages will be installed:
  fontconfig-config fonts-dejavu-core geoip-database libdeflate0 libfontconfig1 libgd3 libgeoip1 libicu67 libjbig0 libjbig20 libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream libnginx-mod-streaming libxml2 libxslt1.1 nginx-common nginx-core
0 upgraded, 29 newly installed, 0 to remove and 0 not upgraded.
Need to get 18.0 MB of archives.
After this operation, 60.1 MB of additional disk space will be used.
Get:1 http://deb.debian.org/debian bullseye/main amd64 fonts-dejavu-core all 2.37-2 [1069 kB]
Get:2 http://deb.debian.org/debian bullseye/main amd64 fontconfig-config all 2.13.1-4.2 [281 kB]
Get:3 http://deb.debian.org/debian bullseye/main amd64 geoip-database all 20191224-3 [3032 kB]
Get:4 http://deb.debian.org/debian bullseye/main amd64 libdeflate0 amd64 1.7-1 [53.1 kB]
Get:5 http://deb.debian.org/debian bullseye/main amd64 libfontconfig1 amd64 2.13.1-4.2 [347 kB]
Get:6 http://deb.debian.org/debian bullseye/main amd64 libjpeg62-turbo amd64 1:2.0.6-4 [151 kB]
Get:7 http://deb.debian.org/debian bullseye/main amd64 libjbig0 amd64 2.1-3.1+b2 [31.0 kB]
Get:8 http://deb.debian.org/debian bullseye/main amd64 libwebp6 amd64 0.6.1-2.1 [258 kB]
```

```
haidevops@apigee-instance:~$ systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
  Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
  Active: active (running) since Sat 2022-10-01 16:17:38 UTC; 27s ago
    Docs: man:nginx(8)
   Process: 1065 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Process: 1066 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
 Main PID: 1254 (nginx)
   Tasks: 2 (limit: 4391)
   Memory: 2.9M
      CPU: 29ms
     CGroup: /system.slice/nginx.service
             └─1254 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
                 ├─1257 nginx: worker process

Oct 01 16:17:38 apigee-instance systemd[1]: Starting A high performance web server and a reverse proxy server...
Oct 01 16:17:38 apigee-instance systemd[1]: nginx.service: Failed to parse PID from file /run/nginx.pid: Invalid argument
Oct 01 16:17:38 apigee-instance systemd[1]: Started A high performance web server and a reverse proxy server.
```

16. Added an A Record in Cloud DNS Zone with the External IP Address of the compute engine.

DNS Name ?

Resource Record Type ?

TTL * ?

TTL Unit ?

Routing Policy

Default record type

Weighted Round Robin

Geo-Based

IPv4 Address ?

IPv4 Address 1 *

Example: 192.0.2.91

+ ADD ITEM

CREATE CANCEL

EQUIVALENT COMMAND LINE ▼

You can see how my domain is reflecting nginx page

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org. Commercial support is available at nginx.com.

Thank you for using nginx.

Now let's go to the Apigee X Dashboard

You are currently on apigee-364214 Apigee organization (GCP Project ID: apigee-364214) . If this is not the Apigee organization you want to use, please select another organization or [create a new one](#) .

 API Proxies <p>Route, transform, and secure your traffic through the API gateway</p>	 API Products <p>Control how developers consume your APIs</p>	 Portals <p>Publish APIs and reference docs, and on-board developers</p>
 Learn More <p>Tutorials, tips, and documentation for the new Edge experience</p>	 Community <p>Have questions? Get answers from other Edge users and Apigeeeks</p>	

17.Create a Nginx API Proxy for Backend Server – Go to Develop → API Proxies → Select Reverse Proxy → Add name and add target path (domain) and base path.

The **Proxy Base Path** is part of the URL used to make requests to your API. Apigee uses the URL to match and route incoming requests to the proper API proxy.

Target (Existing API): This defines the target URL that Apigee invokes on a request to the API proxy. The service is hosted at Apigee and returns simple data. It requires no API key or access token.

The screenshot shows the 'Create Proxy' interface in the Apigee developer console. The left sidebar has 'API Proxies' selected. The main area shows three options:

- Reverse proxy (most common)**: Route inbound traffic to a backend service. Or [Use OpenAPI Spec](#).
- No target**: Create a simple API proxy that does not route to any backend target. Or [Use OpenAPI Spec](#).
- Upload proxy bundle**: Import an existing proxy from a zip archive.

The screenshot shows the 'Proxy details' step in the 'Create Proxy: nginx-proxy' wizard. The left sidebar has 'API Proxies' selected. The main area shows the following fields:

- Name:** nginx-proxy (Available)
- Base path:** /nginx-proxy (This proxy will handle requests on base name/base path. [Learn more](#))
- Description:** (Optional)
- Target (Existing API):** http://nginx.nanditadevops.cloud (The URL of the backend service that this proxy invokes)

At the bottom are 'Previous' and 'Next' buttons.

The screenshot shows the Apigee UI for creating a proxy named 'nginx-proxy'. The left sidebar has 'API Proxies' selected. The main area is titled 'Create Proxy: nginx-proxy' and is on the 'Policies' step (step 2 of 3). The 'Proxy details' step (step 1) is completed, indicated by a checkmark.

Common policies:

- Security: Authorization:** Radio buttons for API Key, OAuth 2.0, and Pass through (no authorization). 'Pass through (no authorization)' is selected.
- Quota:** A checkbox for 'Impersonate per App' is checked. A note says: 'Available only for proxies with authentication. Quota details are configured in API Products. Learn more.'
- Security: Browser:** A checkbox for 'Add CORS headers' is checked. A note says: 'Required for enabling web browser access to this proxy. Learn more.'

Buttons at the bottom right: 'Previous' (disabled), 'Next' (blue).

The screenshot shows the Apigee UI for creating a proxy named 'nginx-proxy'. The left sidebar has 'API Proxies' selected. The main area is titled 'Create Proxy: nginx-proxy' and is on the 'Summary' step (step 3 of 3). The 'Proxy details' and 'Policies' steps are also completed, indicated by checkmarks.

Summary:

Proxy name	nginx-proxy
Proxy type	Reverse Proxy
Proxy base path	/nginx-proxy
Target	http://nginx.nanditadevops.cloud
Policies	Authorization - Pass through (no authorization)

Optional Deployment:

A checkbox for 'dev-env' is checked. A note says: 'The proxy will begin handling requests only after it's deployed to an environment.'

Buttons at the bottom right: 'Previous' (disabled), 'Create and deploy' (blue).

- ✓ **Created proxy: nginx-proxy**
- ✓ **Deployed proxy to environment: dev-env**



18. After creating a new proxy, you must deploy it so that you can try it out. You can see that nginx-proxy is deployed with revision 1.

A screenshot of the Apigee X 'nginx-proxy' proxy summary page. The top navigation bar shows the proxy name and a 'SWITCH TO CLASSIC' link. Below the bar, there are three tabs: 'OVERVIEW' (which is selected), 'DEVELOP', and 'DEBUG'. Under the 'Proxy summary' section, there are three buttons: 'DEPLOY' (with a plus sign icon), 'DUPLICATE' (with a copy icon), and 'DELETE' (with a trash bin icon). The 'Deployments' section shows a table with one row: Revision 1, Status dev-env. The 'Endpoints' section includes 'Proxy endpoints' (default endpoint with basepath /nginx-proxy) and 'Target endpoints' (default endpoint with target connection http://nginx.nanditadevops.cloud). The bottom of the page has a footer with links to 'Documentation', 'Support', 'Community', and 'Feedback'.

19. Add a step of Response Cache for your Nginx-Proxy – To see the variation of latency in the proxy

Add Step

Policy Instance

TRAFFIC MANAGEMENT

- Quota
- Spike Arrest
- Response Cache
- Lookup Cache
- Populate Cache
- Invalidate Cache
- Reset Quota

SECURITY

- Basic Authentication
- XML Threat Protection
- JSON Threat Protection
- Regular Expression Protection
- OAuth v2.0

Policy Type Response Cache

Display Name

Name

Target Endpoint

API Proxies > nginx-proxy > Develop > 1

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ResponseCache continueOnError="false" enabled="true" name="Response-Cache-1">
  <DisplayName>Response Cache-1</DisplayName>
  <Properties/>
  <CacheKey>
    <Prefix/>
    <KeyFragment ref="request.uri" type="string"/>
  </CacheKey>
  <Scope>Exclusive</Scope>
  <ExpirySettings>
    <ExpiryDate/>
    <TimeOfDay/>
    <TimeoutInSec ref="">3600</TimeoutInSec>
  </ExpirySettings>
  <SkipCacheLookup/>
  <SkipCachePopulation/>
</ResponseCache>

```

Apigee

apigee-364214



< Develop

API Proxies

Shared Flows

Offline Debug

Publish

Analyze

Admin

API Proxies > nginx-proxy > Debug > 2

OVERVIEW DEVELOP DEBUG PERFORMANCE

A new version of the Proxy Editor is available for preview

Try now Learn more

Send Requests

Method URL

get https://apigee.nanditadevops.cloud/nginx-proxy

Status

200

Transaction Map

Status Method URI Elapsed

9 304 GET /nginx-proxy 11 ms

8 304 GET /nginx-proxy 12 ms

7 304 GET /nginx-proxy 11 ms

6 304 GET /nginx-proxy 9 ms

5 304 GET /nginx-proxy 14 ms

4 304 GET /nginx-proxy 9 ms

3 304 GET /nginx-proxy 11 ms

2 304 GET /nginx-proxy 11 ms

1 304 GET /nginx-proxy 11 ms

View Options

Transaction Map

 Show Disabled Policies (none) Show Skipped Phases (2) Show All FlowLogs (12)

Phase Details

View Options

Transaction Map

 Show Disabled Policies (none) Show Skipped Phases (2) Show All FlowLogs (12)

Phase Details

View Options

Transaction Map

 Show Disabled Policies (none) Show Skipped Phases (2) Show All FlowLogs (12)

Phase Details

View Options

Transaction Map

 Show Disabled Policies (none) Show Skipped Phases (2) Show All FlowLogs (12)

Phase Details

View Options

Transaction Map

 Show Disabled Policies (none) Show Skipped Phases (2) Show All FlowLogs (12)

Phase Details

View Options

Transaction Map

 Show Disabled Policies (none) Show Skipped Phases (2) Show All FlowLogs (12)

Phase Details

View Options

Transaction Map

 Show Disabled Policies (none) Show Skipped Phases (2) Show All FlowLogs (12)

Phase Details

View Options

Transaction Map

 Show Disabled Policies (none) Show Skipped Phases (2) Show All FlowLogs (12)

Phase Details

View Options

Transaction Map

 Show Disabled Policies (none) Show Skipped Phases (2) Show All FlowLogs (12)

Phase Details

View Options

Transaction Map

 Show Disabled Policies (none) Show Skipped Phases (2) Show All FlowLogs (12)

Phase Details

View Options

Transaction Map

 Show Disabled Policies (none) Show Skipped Phases (2) Show All FlowLogs (12)

Phase Details

View Options

Transaction Map

 Show Disabled Policies (none) Show Skipped Phases (2) Show All FlowLogs (12)

Phase Details

View Options

Transaction Map

 Show Disabled Policies (none) Show Skipped Phases (2) Show All FlowLogs (12)

Phase Details

View Options

Transaction Map

 Show Disabled Policies (none) Show Skipped Phases (2) Show All FlowLogs (12)

Phase Details

View Options

Transaction Map

 Show Disabled Policies (none) Show Skipped Phases (2) Show All FlowLogs (12)

Phase Details

View Options

Transaction Map

 Show Disabled Policies (none) Show Skipped Phases (2) Show All FlowLogs (12)

Phase Details

View Options

Transaction Map

 Show Disabled Policies (none) Show Skipped Phases (2) Show All FlowLogs (12)

Phase Details

View Options

Transaction Map

 Show Disabled Policies (none) Show Skipped Phases (2) Show All FlowLogs (12)

Phase Details

View Options

Transaction Map

 Show Disabled Policies (none) Show Skipped Phases (2) Show All FlowLogs (12)

Phase Details

View Options

Transaction Map

 Show Disabled Policies (none) Show Skipped Phases (2) Show All FlowLogs (12)

Phase Details

View Options

Transaction Map

 Show Disabled Policies (none) Show Skipped Phases (2) Show All FlowLogs (12)

Phase Details

View Options

Transaction Map

 Show Disabled Policies (none) Show Skipped Phases (2) Show All FlowLogs (12)

Phase Details

View Options

Transaction Map

 Show Disabled Policies (none) Show Skipped Phases (2) Show All FlowLogs (12)

Phase Details

View Options

Transaction Map

 Show Disabled Policies (none) Show Skipped Phases (2) Show All FlowLogs (12)

Phase Details

View Options

Transaction Map

 Show Disabled Policies (none) Show Skipped Phases (2) Show All FlowLogs (12)

Phase Details

View Options

Transaction Map

 Show Disabled Policies (none) Show Skipped Phases (2) Show All FlowLogs (12)

Phase Details

View Options

Transaction Map

 Show Disabled Policies (none) Show Skipped Phases (2) Show All FlowLogs (12)

Phase Details

View Options

Transaction Map

 Show Disabled Policies (none) Show Skipped Phases (2) Show All FlowLogs (12)

Phase Details

View Options

Transaction Map

 Show Disabled Policies (none) Show Skipped Phases (2) Show All FlowLogs (12)

Phase Details

View Options

Transaction Map

 Show Disabled Policies (none) Show Skipped Phases (2) Show All FlowLogs (12)

Phase Details

View Options

Transaction Map

 Show Disabled Policies (none) Show Skipped Phases (2) Show All FlowLogs (12)

Phase Details

View Options

Transaction Map

 Show Disabled Policies (none) Show Skipped Phases (2) Show All FlowLogs (12)

Phase Details

View Options

Transaction Map

 Show Disabled Policies (none) Show Skipped Phases (2) Show All FlowLogs (12)

Phase Details

View Options

Transaction Map

 Show Disabled Policies (none) Show Skipped Phases (2) Show All FlowLogs (12)

Phase Details

View Options

Transaction Map

 Show Disabled Policies (none) Show Skipped Phases (2) Show All FlowLogs (12)

Phase Details

View Options

Transaction Map

 Show Disabled Policies (none) Show Skipped Phases (2) Show All FlowLogs (12)

Phase Details

View Options

Transaction Map

 Show Disabled Policies (none) Show Skipped Phases (2) Show All FlowLogs (12)

Phase Details

View Options

Transaction Map

 Show Disabled Policies (none) Show Skipped Phases (2) Show All FlowLogs (12)

Phase Details

View Options

Transaction Map

 Show Disabled Policies (none) Show Skipped Phases (2) Show All FlowLogs (12)

Phase Details

View Options

Transaction Map

 Show Disabled Policies (none) Show Skipped Phases (2) Show All FlowLogs (12)

Phase Details

View Options

Transaction Map

 Show Disabled Policies (none) Show Skipped Phases (2) Show All FlowLogs (12)

Phase Details

View Options

Transaction Map

 Show Disabled Policies (none) Show Skipped Phases (2) Show All FlowLogs (12)

Phase Details

View Options

Transaction Map

 Show Disabled Policies (none) Show Skipped Phases (2) Show All FlowLogs (12)



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
 Commercial support is available at nginx.com.

Thank you for using nginx.

20. Go to Publish field and create API Products. Add name, description, display name and add environment which was created.

The screenshot shows the "Products" section of the Apigee X interface. The left sidebar has categories: Develop, Publish (selected), API Products (highlighted in orange), Portals, Developers, Apps, Analyze, and Admin. The main area shows a "Product details" form with the following fields:

- Name: nginx-product
- Display Name: nginx-product
- Description: nginx-product
- Environment: devenv (selected from a dropdown)
- Access: Public (selected from a dropdown)
- Automatically approve access requests: checked
- Quota: 20 (Request every 1 minute, Unit: minute)
- Allowed OAuth scope: (empty input field)

The screenshot shows the 'Operations' section of the Apigee X API Product configuration. On the left sidebar, 'API Products' is selected under the 'Publish' category. In the main area, the 'Source' tab is active, showing an 'API Proxy' named 'nginx-proxy'. The 'Operation' section includes fields for 'Path' (set to '/'), 'Methods' (set to 'GET, PUT, and POST'), and a 'Quota' section with values '20', 'Request every', 'Time interval 1', and 'Unit minute'. Below these are sections for 'GraphQL Operations' and 'Custom Attributes', both currently empty. At the bottom are 'SAVE' and 'CANCEL' buttons.

Attach the API Product to the Nginx Proxy

21. Creating Portal for the API Product – Add a name and description to it.

The screenshot shows the 'Portals' creation dialog box. The 'Name' field is set to 'nginx-portal' and the 'Description' field is set to 'nginx portal'. The 'Create' button is highlighted in blue at the bottom right of the dialog. The background shows a blurred view of the Apigee X interface with the 'Portals' section visible.

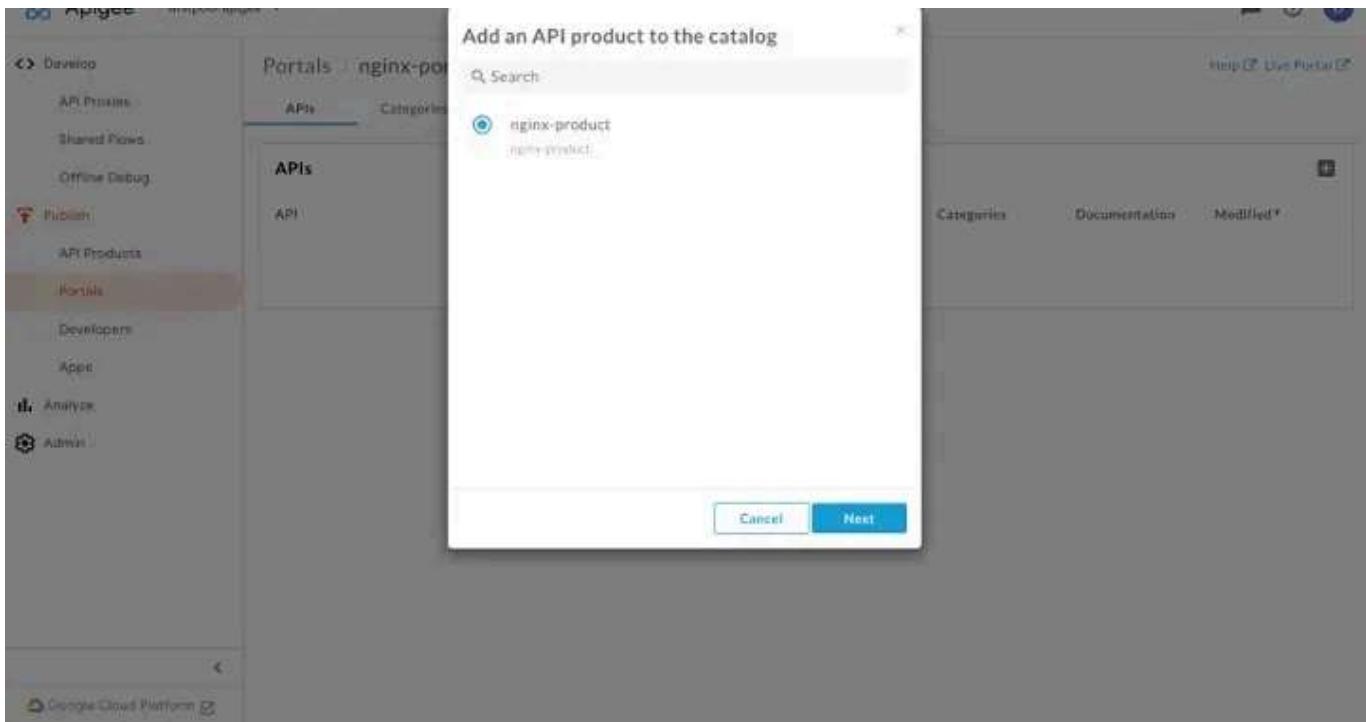
22. Choose API Catalog in the Nginx Portal created

The screenshot shows the Google Cloud Platform API Manager interface. The left sidebar has sections for Develop, API Proxies, Shared Flows, Offline Debug, Publish (with Portals selected), Developers, Apps, Analyze, and Admin. The main area shows 'Portals > nginx-portal > Overview'. A dropdown menu from the 'Overview' button includes Accounts, API catalog (selected), Pages, Menus, Assets, Themes, and Settings. Below the dropdown are four main cards: Accounts (Manage individual portal users, teams, & authentication), API catalog (Manage visibility and documentation of published APIs), Pages (Create and publish pages using Markdown), and Menus (Change the information architecture). At the bottom are three partially visible cards: Assets, Themes, and Settings.

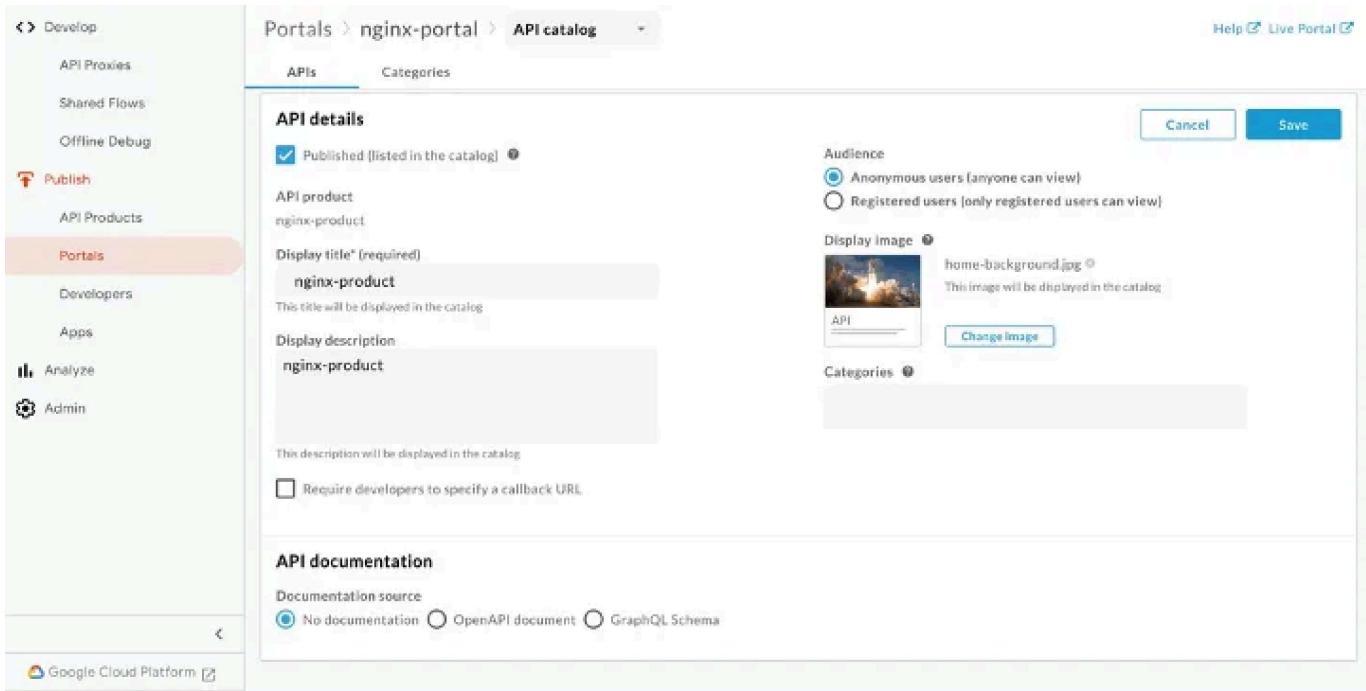
The screenshot shows the API Portal interface. The left sidebar has tabs for Develop, Publish, API Products, Portals (which is selected and highlighted in orange), Developers, Apps, Analyze, and Admin. The main area shows the navigation path: Portals > nginx-portal > API catalog > APIs. Below this, there are two tabs: APIs (selected) and Categories. A table titled "APIs" lists columns for API, Description, Visibility, Categories, Documentation, and Modified*. The table body contains the message "No APIs added.".

API	Description	Visibility	Categories	Documentation	Modified*
No APIs added.					

23. Click on the + Symbol and add an API product to the catalog



24. See the API Portal Dashboard – Click on the Live Portal placed on top right-hand corner. (Check on the box – Published, add a title and description, add an image and Documentation source)



25. It will take you to Developer's Portal and Developer can review the Application here

The screenshot shows the 'APIs > nginx-product' page in the Apigee developer portal. The left sidebar has tabs for Develop, Publish, API Products, Portals (which is selected), Developers, Apps, Analyze, and Admin. The main content area has tabs for APIs (selected) and Categories. The 'API details' section contains the following information:

Status	Published (listed in the catalog)	Audience	Anonymous users [anyone can view]
API product	nginx-product	Display Image	 home-background.jpg This image will be displayed in the catalog
Display title	nginx-product		
Display description	nginx-product	Categories	
Callback URL	Not required		

The 'API documentation' section indicates 'No documentation'.



26. Click on APIs – You can see nginx product created. Click on it to get a better view.

APIs

Filter by title & description



nginx-product

nginx-product

Refer Apigee Documentation for better understandingBuy me a coffee :) ← — — If you like my articles**Buy me a coffee**[Apigee](#)[Apigee Api Management](#)[Gcp](#)[Api Management](#)[Kubernetes](#)**Written by Nandita Sahu**

1.4K Followers · 17 Following

[Follow](#)

I am quick learner and always love to explore new tools and technologies. You can buy me a coffee :) <https://www.buymeacoffee.com/NanditaSahu>

Responses (1)



What are your thoughts?

Respond



Vkcheb

about 1 year ago

...

Hi Nandita.. i went through your blog about apigee x architecture. want to connect with you regarding some of the variants of architecture when apigee x is communicating from internet, on prem, gcp cloud or other clouds to and fro. how will the changes happen when we opt for one of them.

thanks,

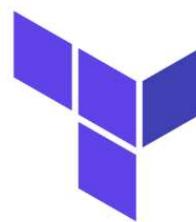
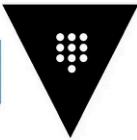
Vijay.

+917760380455.



Reply

More from Nandita Sahu

**Terraformer**

Nandita Sahu

How to Integrate HashiCorp Vault with Jenkins to secure your Secrets

What is Vault?

Jun 28, 2022

41

4



...



Nandita Sahu

Terraformer—Generate Terraform Files from Existing Infrastructure

What is Terraformer?

Jul 14, 2022

299

10



...



Nandita Sahu

How to define Dependencies between modules in Terragrunt

In this article , you will get an idea about how to define dependency blocks in Terragrunt...

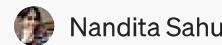
Jan 4, 2023

20

4



...



Nandita Sahu

HA VPN connections between GCP and AWS

In this article, you will get a brief idea about how to create highly available VPN...

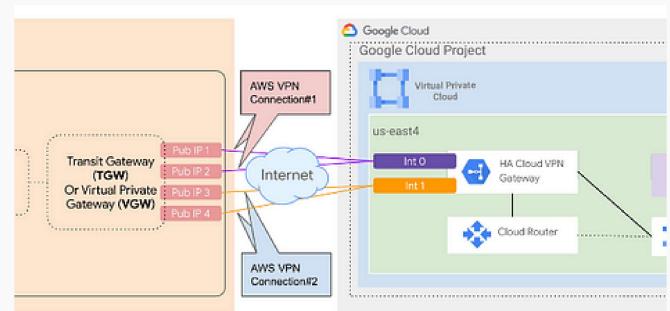
Oct 23, 2022

109

1



...



See all from Nandita Sahu

Recommended from Medium

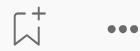


 Manish Sonal

Systems Design Interview: Design a Ticket Master

Systems design is subjective and it evolves with time and requirement updates, there is...

 Sep 22  48



...

Full-Stack App Hosting



 In FAUN—Developer Community  by Ali Hamza

Deploying a Full-Stack App on Google Cloud Platform: A Step-by...

This guide will explain how to deploy a full-stack application on GCP with each essential...

Nov 15  32



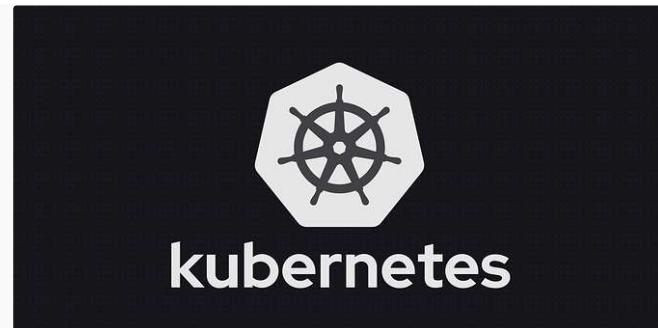
...

Lists



Natural Language Processing

1842 stories • 1466 saves



In Stackademic by Crafting-Code

I Stopped Using Kubernetes. Our DevOps Team Is Happier Than Ever

Why Letting Go of Kubernetes Worked for Us

Nov 19

3.8K

112



...



In Google Cloud - Community by minherz

All the ways to scrape Prometheus metrics in Google Cloud

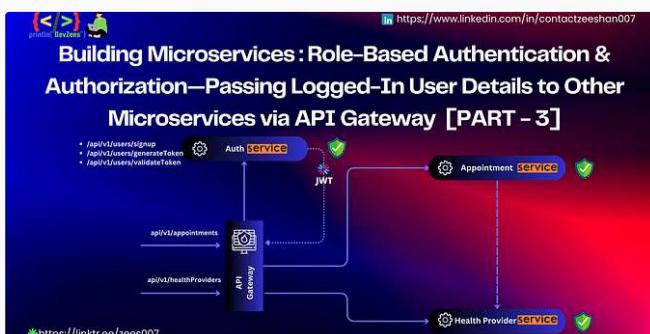
Look into methods of scraping Prometheus metrics emitted by your application on...

Nov 10

2



...



In Level Up Coding by Zeeshan Adil

Building Microservices [PART-3]: Role-Based Authentication &...

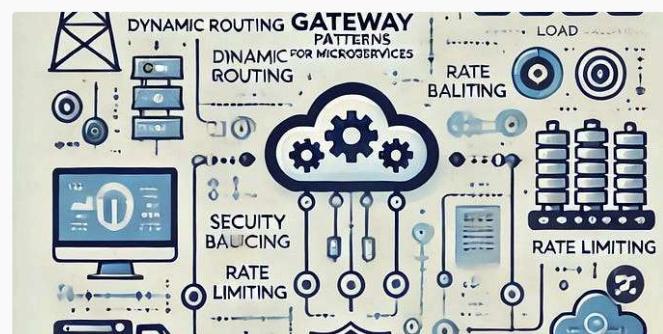
Welcome Back to DevZees ❤️

Nov 12

109



...



B Balian's technologies and innovation lab

API Gateway Patterns with Spring Cloud Gateway: Routing, Load...

In microservices architectures, an API Gateway acts as the single entry point for all...

Nov 12

2



...

See more recommendations