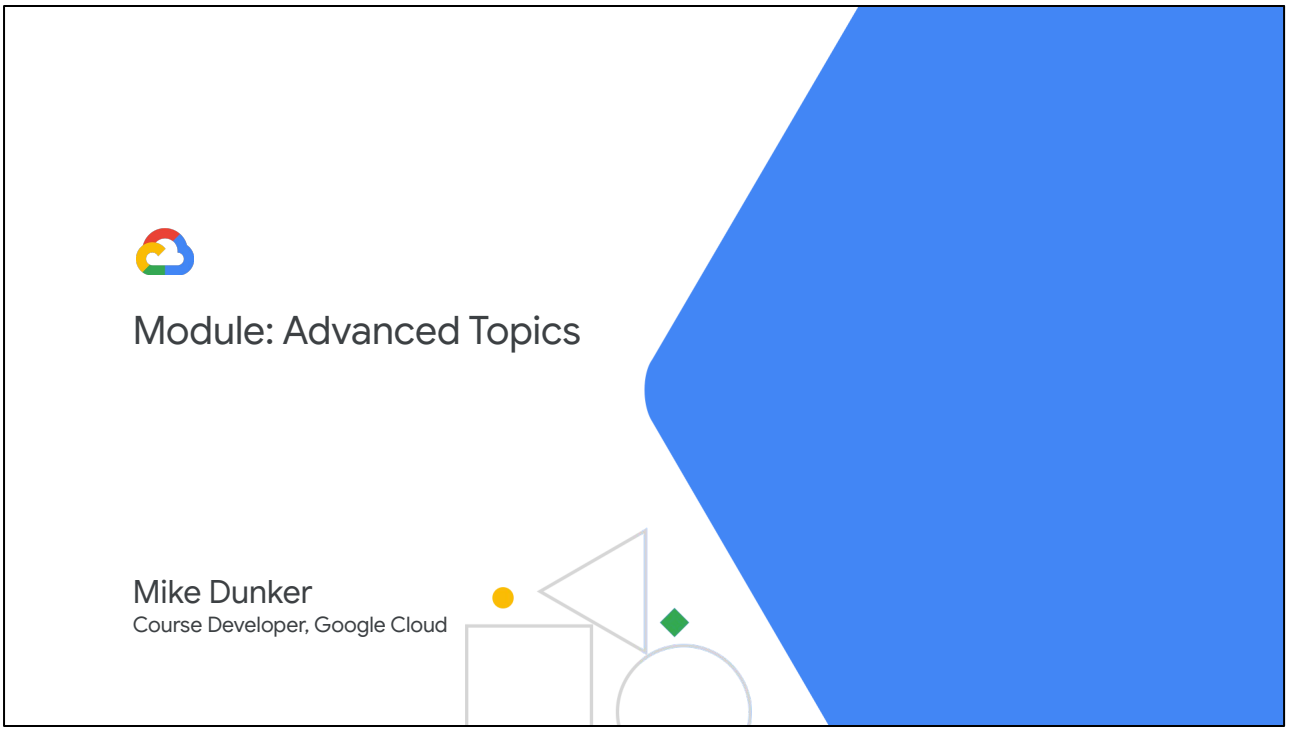
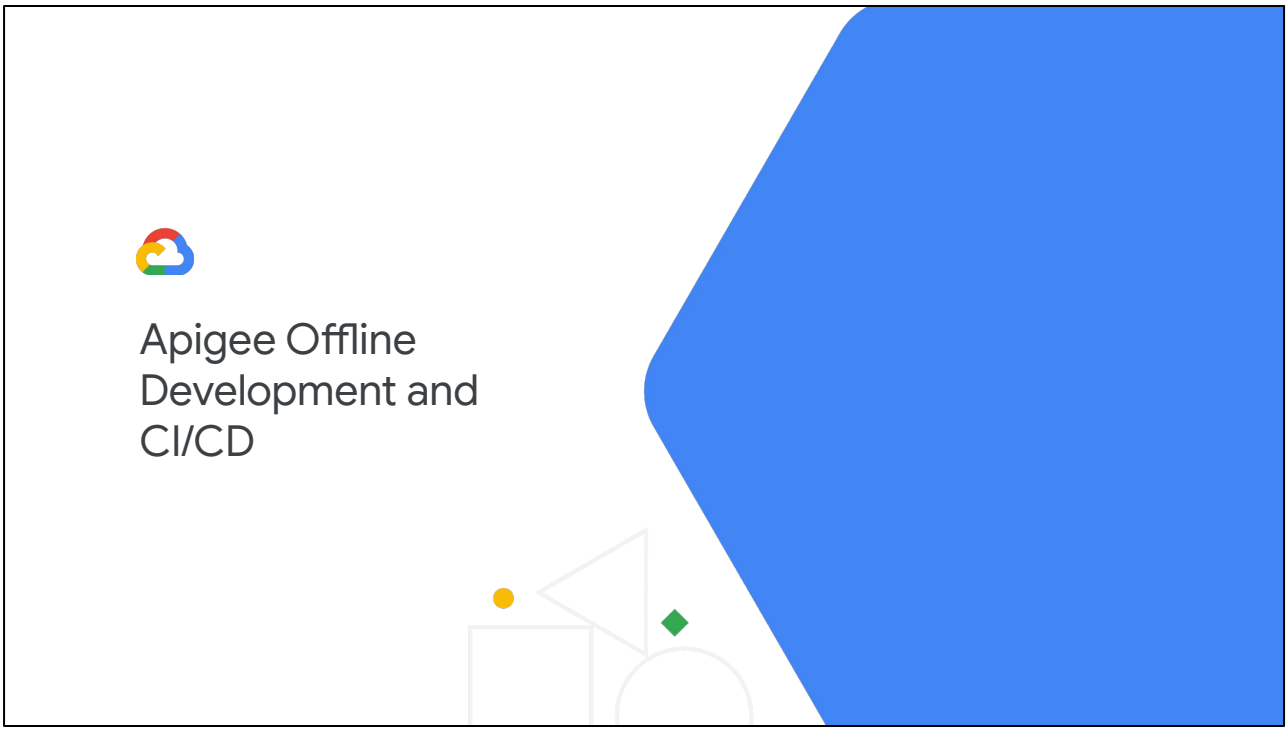


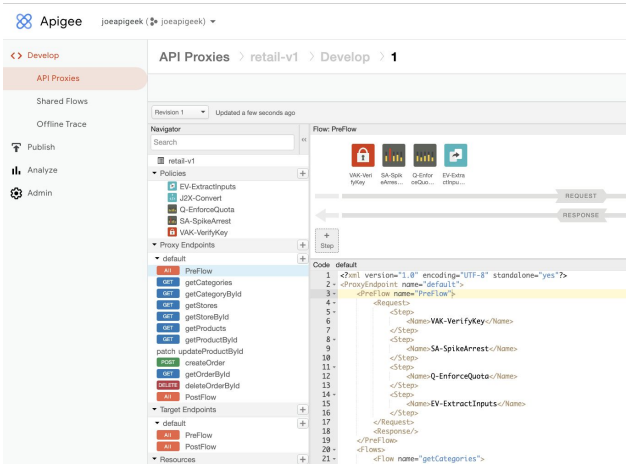
In this module you will learn how to integrate Apigee build processes into automated build tools, and you will learn about the different deployment options available for Apigee.



This lecture will discuss how Apigee supports offline development and Continuous Integration, Continuous Delivery, or CI/CD.

API proxy editor

- Apigee Console provides a drag-and-drop graphical editor for proxy development.
- The editor is useful for learning Apigee proxy development and for building test proxies.
- API engineers can seamlessly switch between modifying and testing proxies.



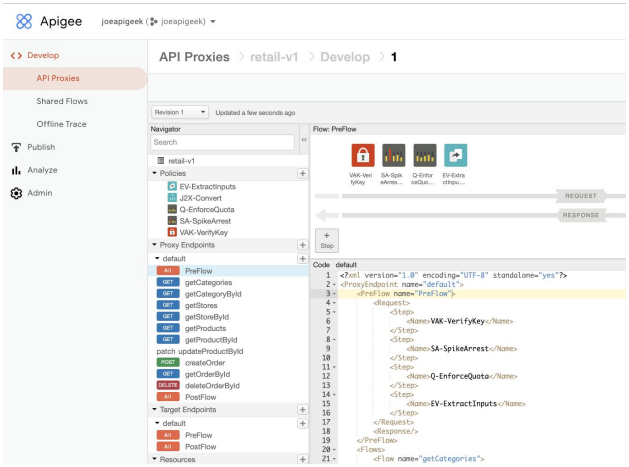
As you completed the labs for this series of courses, you used the proxy editor to build your proxies.

The editor is a great way to learn about proxy development.

When building a proxy, an API engineer can seamlessly switch between the editing and testing of the proxy.

Online editing

- It is possible to delete or overwrite proxy revisions when using the console.
- Enterprises and teams should use source control and CI/CD for proxy development and deployment.
- Apigee utilities allow deployment of proxies and configuration from build automation tools.



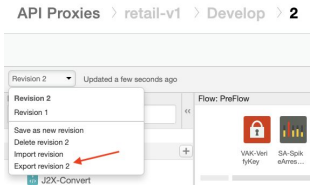
There are some problems with online editing of API proxies, though.

Deployed revisions are immutable and cannot be edited. However, non-deployed revisions can be edited or deleted, and there is no history of changes in a revision. The Apigee console makes it easy to overwrite or delete revisions of an API proxy.

It is important to store development artifacts in source control, especially when working as a team. Enterprise teams also generally use CI/CD to provide better control over the release process.

Apigee provides utilities that allow you to integrate the deployment of proxies and configuration from build automation tools.

Storing an API proxy in source control



- Export revision menu downloads a proxy bundle zip file.
- The unzipped bundle can be stored in source control.

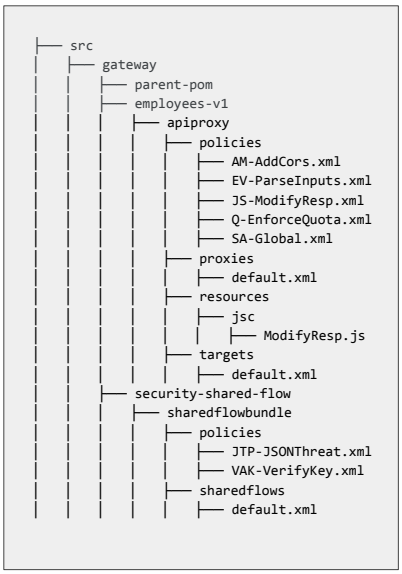
```
$ unzip unzip retail-v1_rev2_2021_02_26.zip
Archive:  retail-v1_rev2_2021_02_26.zip
   creating: apiproxy/
   creating: apiproxy/targets/
  inflating: apiproxy/targets/default.xml
   creating: apiproxy/proxies/
  inflating: apiproxy/proxies/default.xml
   creating: apiproxy/policies/
  inflating: apiproxy/policies/VAK-VerifyKey.xml
  inflating: apiproxy/policies/SA-SpikeArrest.xml
  inflating: apiproxy/policies/Q-EnforceQuota.xml
  inflating: apiproxy/policies/EV-ExtractInputs.xml
  inflating: apiproxy/policies/J2X-Convert.xml
  inflating: apiproxy/retail-v1.xml
$
```

When you download an API proxy revision, it is retrieved as a zip file.

The downloaded bundle can be unzipped and stored in source control.

Maven plugins

- Apache Maven is an open source build automation tool
- Build and deploy Apigee proxies using the *apigee-deploy-maven-plugin*



Apache Maven is an open source build automation tool that can be used as part of a CI/CD pipeline.

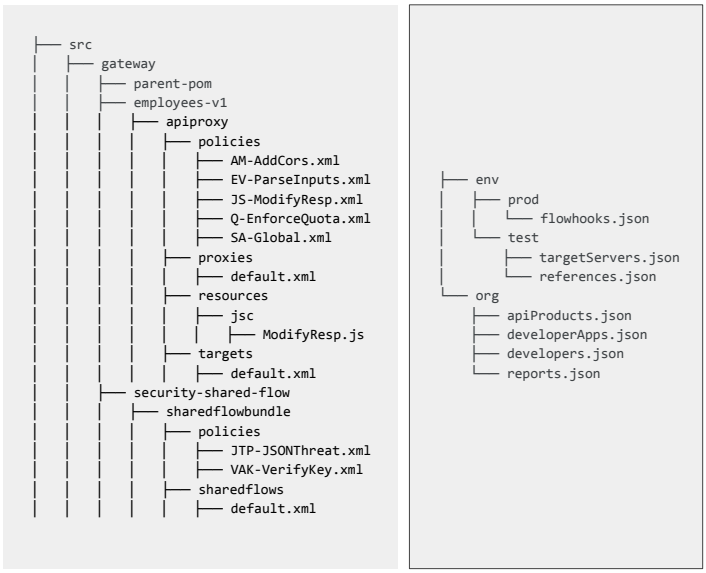
There are two Maven plugins that can be used for managing Apigee.

The Apigee deploy maven plugin is used to build and deploy Apigee proxies and shared flows.

This plugin uses the same directory structure as is extracted from a downloaded zip file bundle.

Maven plugins

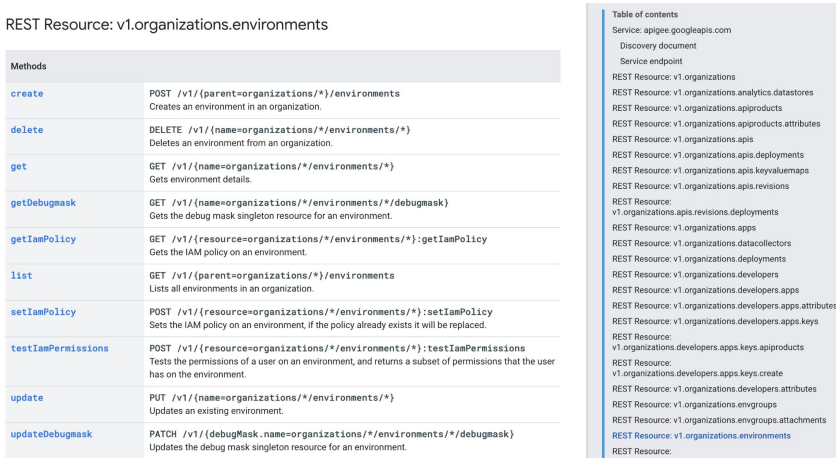
- Apache Maven is an open source build automation tool
- Build and deploy Apigee proxies using the *apigee-deploy-maven-plugin*
- Manage and deploy Apigee configuration using the *apigee-config-maven-plugin*



The Apigee config maven plugin is used to manage and deploy Apigee configuration entities.

This plugin can be used as part of a CI/CD process to deploy changes to configuration.

Apigee API (apigee.googleapis.com)



REST Resource: v1.organizations.environments

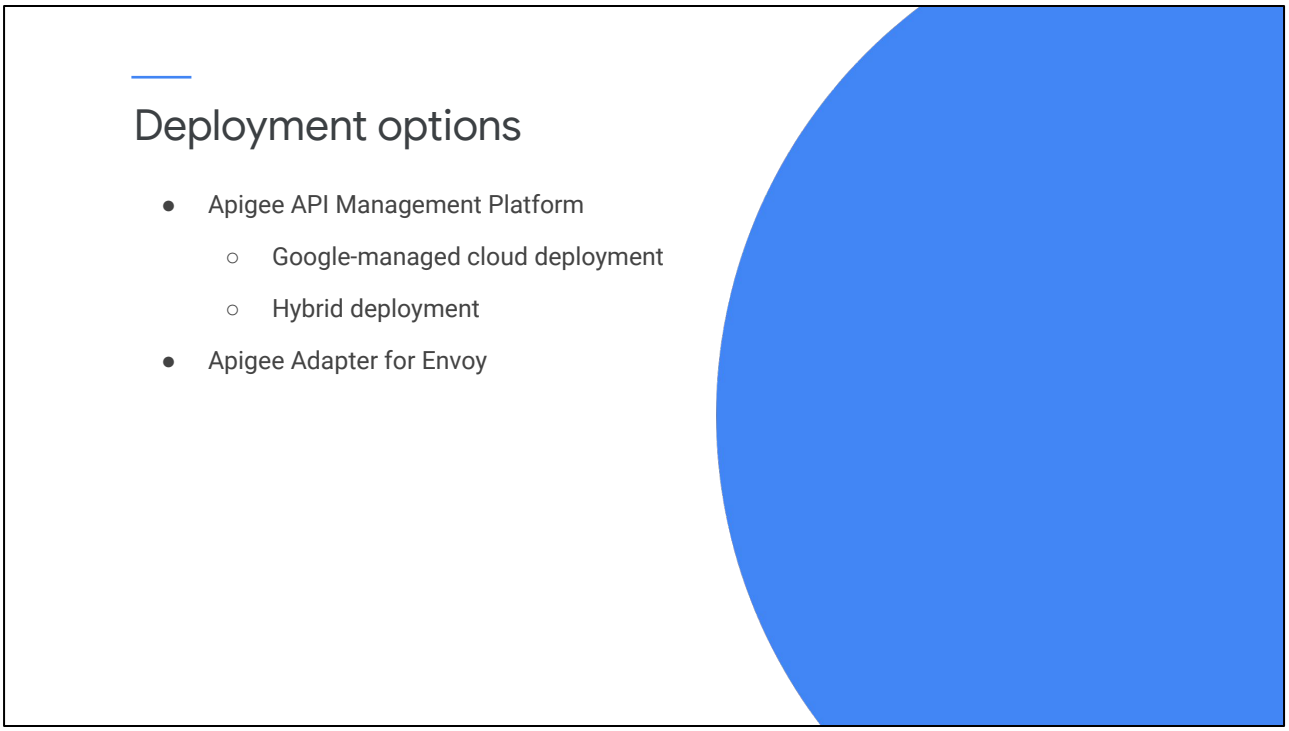
Methods	
create	POST /v1/{parent=organizations/*}/environments Creates an environment in an organization.
delete	DELETE /v1/{name=organizations/*/environments/*} Deletes an environment from an organization.
get	GET /v1/{name=organizations/*/environments/*} Gets environment details.
getDebugmask	GET /v1/{name=organizations/*/environments/*/debugmask} Gets the debug mask singleton resource for an environment.
getIamPolicy	GET /v1/{resource=organizations/*/environments/*}:getIamPolicy Gets the IAM policy on an environment.
list	GET /v1/{parent=organizations/*}/environments Lists all environments in an organization.
setIamPolicy	POST /v1/{resource=organizations/*/environments/*}:setIamPolicy Sets the IAM policy on an environment, if the policy already exists it will be replaced.
testIamPermissions	POST /v1/{resource=organizations/*/environments/*}:testIamPermissions Tests the permissions of a user on an environment, and returns a subset of permissions that the user has on the environment.
update	PUT /v1/{name=organizations/*/environments/*} Updates an existing environment.
updateDebugmask	PATCH /v1/{debugMask.name=organizations/*/environments/*/debugmask} Updates the debug mask singleton resource for an environment.

Table of contents

The Apigee API can be used if the Maven plugins do not work for you. Nearly everything that can be done using the Apigee console can also be done via the Apigee API.

The Apigee API can be called from CI/CD toolchains to manage the API lifecycle. You can call the Apigee API by using IAM credentials. If you do not have access to particular organizations, environments, or entities from the Apigee console, you will not be able to use those credentials to access those entities via the Apigee API.

This lecture will discuss the different deployment options available when you use Apigee.



There are three deployment options for Apigee's fully featured API management platform.

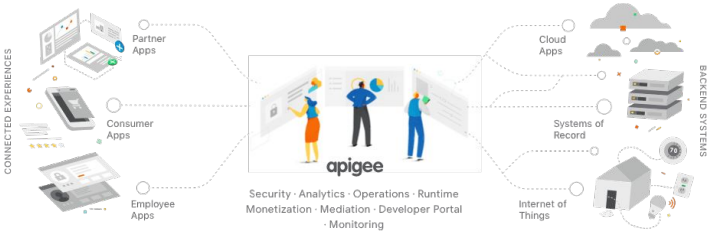
Apigee can be run in Google Cloud, where the infrastructure is fully managed by Google.

Apigee's hybrid deployment model allows customer-managed runtimes, with infrastructure management shared between Google and the customer.

Google also provides the Apigee adapter for Envoy, a lightweight runtime gateway deployment option that lets customers deploy API management functionality in close proximity to backend services.

Let's learn about all of these deployment options.

Fully managed in Google Cloud



- Fully managed, full lifecycle cloud-hosted API management platform

Apigee can be deployed as a fully managed, full lifecycle API management platform running within Google Cloud.

This is a software-as-a-service solution, with the entire platform hosted on Google Cloud and managed by Google.

The managed model requires the least amount of management effort, allowing you to focus your resources on building your API program.

This is the deployment model you have been using for the labs in this course.

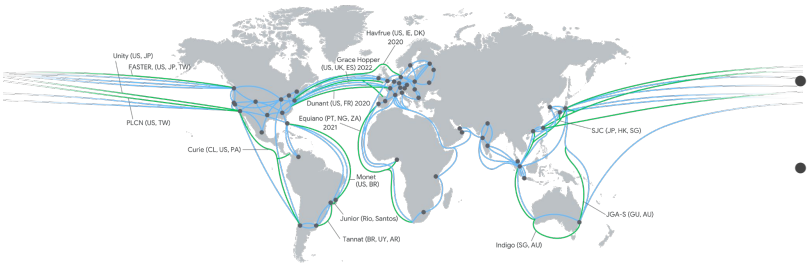
Fully managed in Google Cloud

- Fully managed, full lifecycle cloud-hosted API management platform
- Available in cloud regions around the world

An organization can be hosted in your choice of Google Cloud regions around the world.

Organizations can also be hosted in multiple regions, promoting high availability in case of regional outages, and reducing latency by positioning API gateways close to clients and backends.

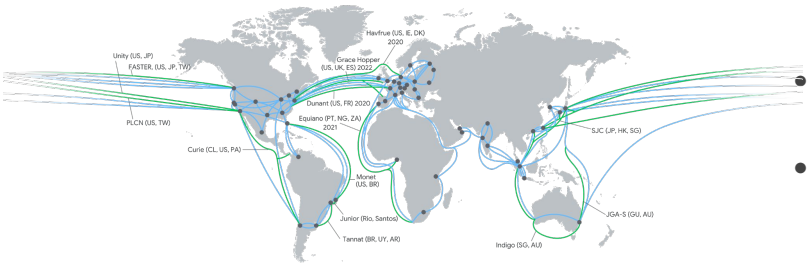
Fully managed in Google Cloud



- Fully managed, full lifecycle cloud-hosted API management platform
- Available in cloud regions around the world
- Leverage Google Cloud's fast, worldwide private network

With managed Apigee, customers can take advantage of Google Cloud's fast, worldwide private network, as well as other Google Cloud services and features.

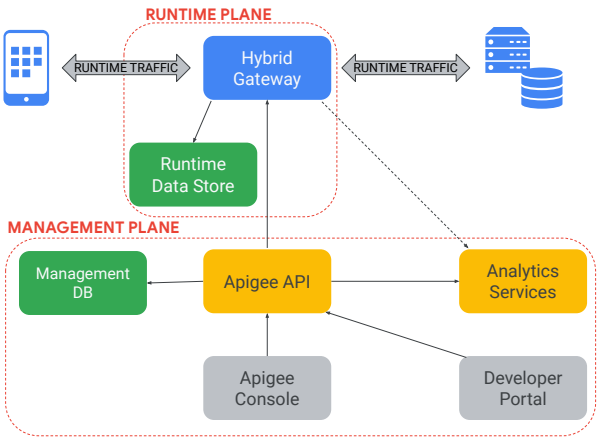
Fully managed in Google Cloud



- Fully managed, full lifecycle cloud-hosted API management platform
- Available in cloud regions around the world
- Leverage Google Cloud's fast, worldwide private network
- Enterprise offerings include entitlements for hybrid orgs

Another benefit of the enterprise offerings of managed Apigee is that they include entitlements for hybrid deployments of Apigee.

Hybrid deployment



- Management plane
  - Hosted in Google Cloud, managed by Google
- Runtime plane
  - Managed by customer in data center or private cloud
  - Deployed as services running in a supported Kubernetes platform
- API traffic remains in runtime plane that is controlled by the customer

Apigee's full lifecycle API management platform can also be deployed using a hybrid deployment model.

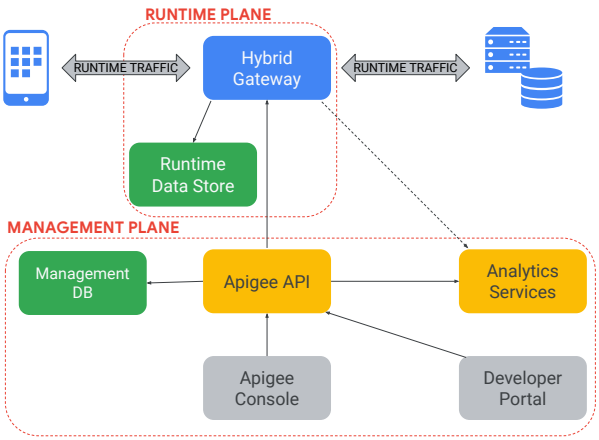
The Apigee management plane is hosted in Google Cloud, and managed by Google.

The Apigee runtime plane is deployed as containerized services on a supported Kubernetes platform, running in a Google Cloud project, a customer data center, or a private cloud, and managed by the customer.

The hybrid deployment model for Apigee allows runtime API traffic to remain within customer-controlled boundaries.

Why use hybrid?

- Latency: Position the gateway as close to workloads as possible
- Security: Process API data within required network boundaries
- Customizability: Use your own TLS ciphers, static IPs, VPNs, etc. on API Gateways
- Fully featured: Full lifecycle API management



There are many benefits of using the hybrid deployment model.

Gateways can be deployed to multiple clouds and data centers, thus allowing API proxies to handle API requests as close to backend workloads as possible.

API traffic can remain inside specified network boundaries, which can help adhere to security requirements.

The hybrid deployment model provides full network customizability, allowing the customer to use chosen TLS ciphers, VPNs, and static IP addresses.

Like the managed cloud deployment model, hybrid deployments provide fully featured API management.

Apigee Adapter for Envoy

- Envoy is an open source, high performance edge and service proxy with a small memory footprint.
- The adapter turns Envoy into an Apigee-managed API gateway that proxies API traffic.
- An instance of the adapter is tied to a central organization and environment.

Envoy is an open source, high performance edge and service proxy that is designed for cloud-native applications. Envoy has broad industry support.

The Apigee adapter for Envoy turns Envoy into an Apigee-managed API gateway that can proxy API traffic.

Each instance of the adapter is tied to a specific organization and environment running in a Google Cloud-managed or hybrid deployment.

Why use the Apigee adapter for Envoy?

- Runs close to backend services: Stays within enterprise-approved network boundaries for security or compliance purposes.
- Asynchronous communication with management plane: Communicates asynchronously without affecting latency.
- Security: Validates API keys and signed JWT tokens without calling an Apigee gateway.
- Configuration-based enforcement: Secures and manages traffic for many microservices as easily as for one.
- Analytics data: Analytics data is delivered to Apigee asynchronously.
- Traffic Management: Utilize quota to manage backend traffic.

The Apigee adapter for Envoy is lightweight and easy to manage, and can be run close to your backend services. Your API traffic does not need to call a central Google Cloud-managed or hybrid organization, allowing your traffic to stay within enterprise-approved network boundaries for security or compliance purposes.

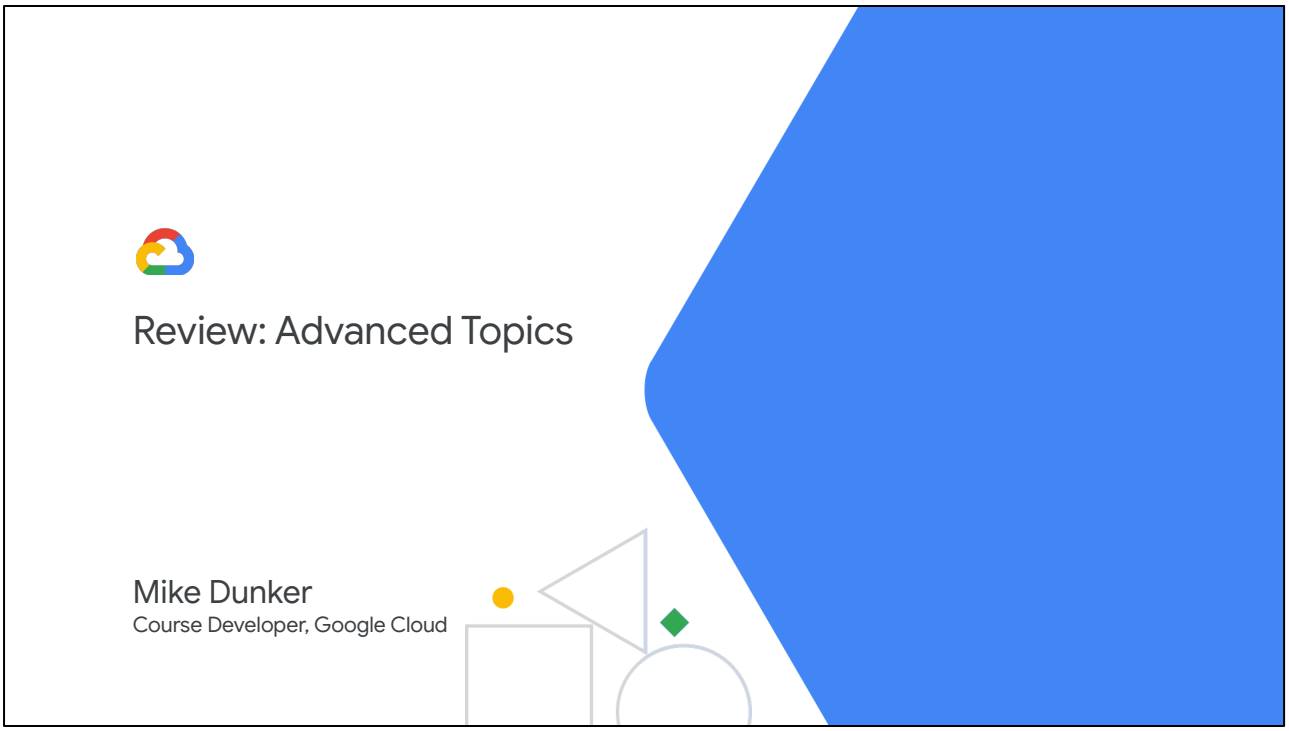
The Envoy adapter communicates with the management plane asynchronously. This allows you to use configuration from the central organization without affecting latency.

The adapter can validate API keys and signed JWT tokens, validating them against API products. The adapter asynchronously retrieves API product and API key information from the configured organization and environment.

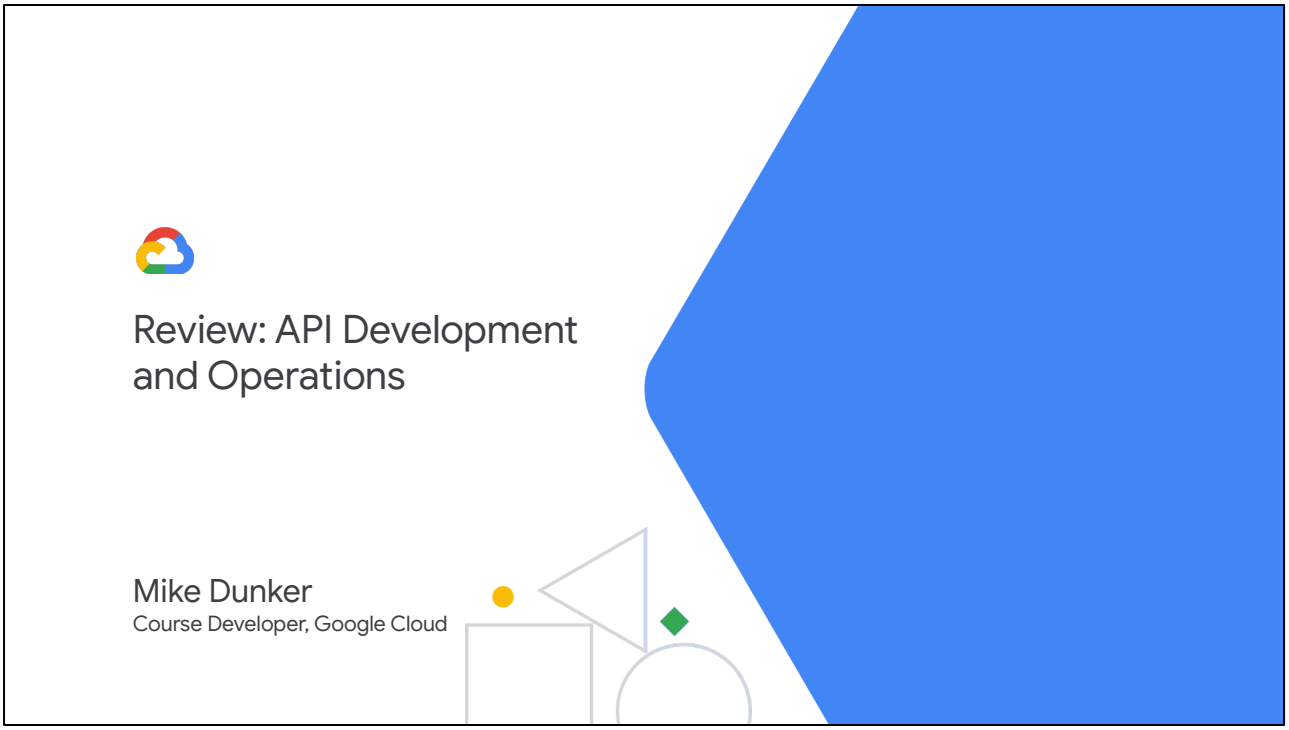
One benefit of the Apigee Adapter for Envoy is that it uses configuration-based enforcement. It is easy to manage the adapter for many microservices.

Analytics data for calls through the adapter is delivered to Apigee asynchronously, allowing full visibility for API traffic running through the adapter.

Quotas can also be enforced using the adapter. Spike arrest rate limiting is a built-in feature of Envoy which can also be used.



In this module, you learned about the deployment options for Apigee gateways, and how to use Apigee with automated build tools.



Thank you for taking the API Development and Operations course.

During this course you learned about API mediation and traffic management.

You learned how APIs can be published using a developer portal.

We discussed how Analytics can be used to understand the performance of your APIs and API program.

And you learned how Apigee proxies and configuration can be deployed using CI/CD tools, as well as how Apigee gateways can be deployed close to backend services.