



DevOps Automation

Learning objectives

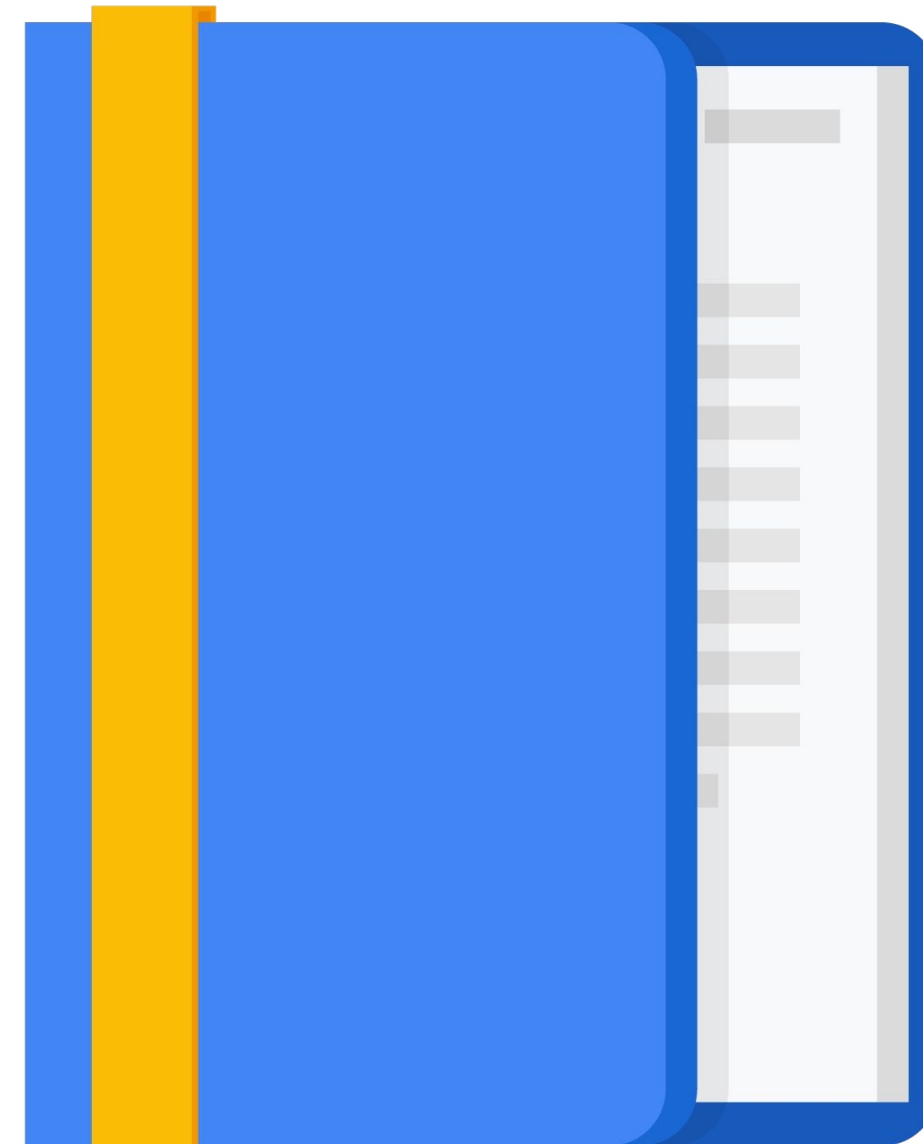
- Automate service deployment using CI/CD pipelines.
- Leverage Cloud Source Repositories for source and version control.
- Automate builds with Cloud Build and build triggers.
- Manage container images with Container Registry.
- Investigate infrastructure with code using Cloud Deployment Manager and Terraform.

Agenda

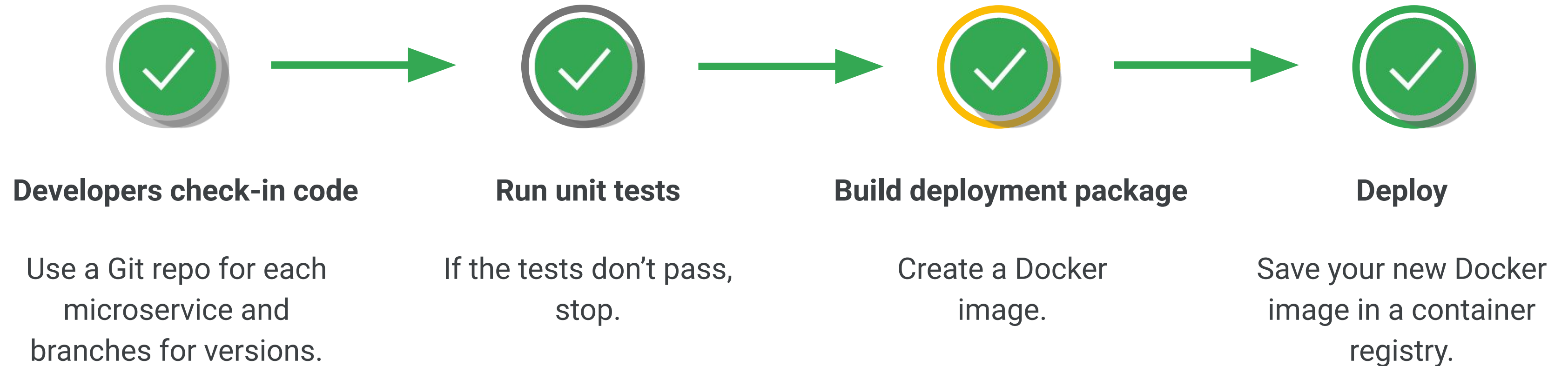
Continuous Integration Pipelines

Infrastructure as Code

Lab



Continuous integration pipelines automate building applications



Google provides the components required for a continuous integration pipeline

Cloud Source Repositories
Developers push to a central repository when they want a build to occur.

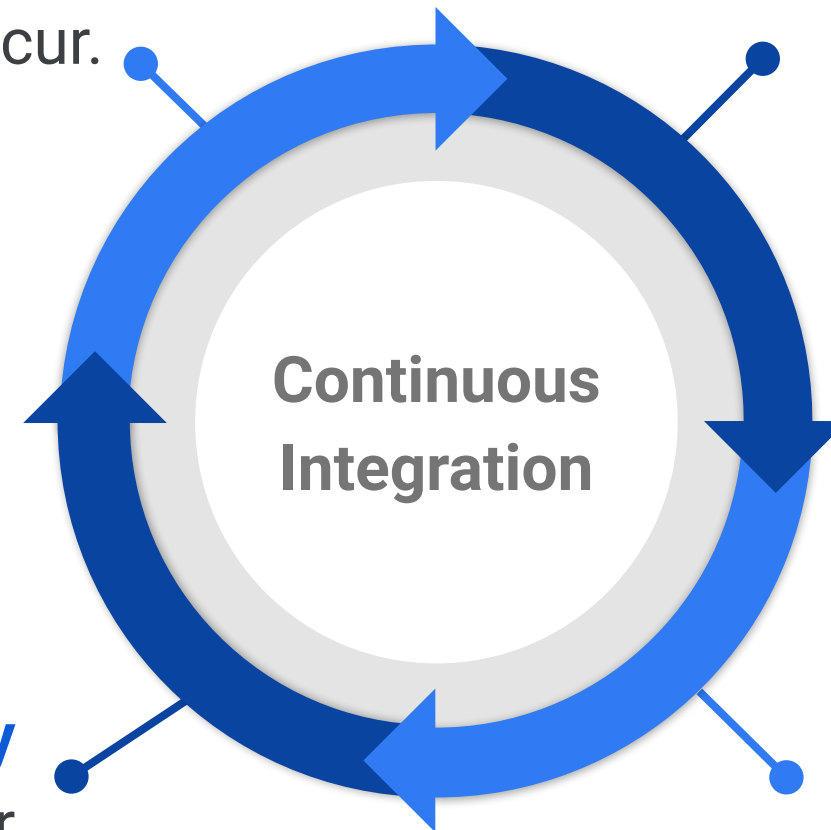
Cloud Build

Build system executes the steps required to make a deployment package or Docker image.

Container Registry
Store your Docker images or deployment packages in a central location for deployment.

Build triggers

Watches for changes in the Git repo and starts the build.



Cloud Source Repositories provides managed Git repositories

Control access to your repos using IAM within your Google Cloud projects.

Cloud Source Repositories

This repository

Search for code or files

Cloud Console

frontend > master

Debug application Clone Edit code

FilesOutline

Repository root

__pycache__statictemplates

Dockerfileapp.yamlauth.pyconfig.pykubernetes-config.yamlmain.py

main_test.pyrequirements.txt

Repository Root

Directories

__pycache__statictemplates

Files

Dockerfileauth.pykubernetes-config.yamlmain_test.py

app.yamlconfig.pymain.pyrequirements.txt

History

Snapshots

Logpoints

Logs

ID	Description	Commit Date	Author
3635640	reduced wait for readiness and updated firebase to this proj	Feb 26 11:46 AM	Kevin Rattan
cec5d61	added readiness	Feb 26 5:32 AM	Kevin Rattan
4ec8cff	fixed liveness probe	Feb 26 4:18 AM	Kevin Rattan
cc430e9	moved to 0.3	Feb 26 3:37 AM	Kevin Rattan

 Google Cloud

Cloud Build lets you build software quickly across all languages

- Google-hosted Docker build service
 - Alternative to using Docker build command
- Use the CLI to submit a build
`gcloud builds submit --tag gcr.io/your-project-id/image-name .`

Cloud Build

History

Triggers

Settings

Build history

REFRESH

Filter builds

Build	Source	Git commit	Trigger name	Trigger	Started	Duration	Artifacts
912335e9-b540...	—	—	—	—	12/30/19, 5:25 PM	13 sec	—
26911ada-69fd...	—	—	—	—	12/30/19, 3:04 PM	21 sec	—
b5f186d8-10a4...	—	—	—	—	12/30/19, 2:42 PM	49 sec	—
a1d3ce2a-5c19...	gs://test-1-263611_cloudbuild/source/1577728920.88-9fbbb839c48644f0a0c822792553fa82.tgz	—	—	—	12/30/19, 1:02 PM	49 sec	gcr.io/test-1-263611/cloud-run-image:v0.1
1196a20c-a3b4...	gs://test-1-263611_cloudbuild/source/1577723841.44-53c69e2b69fe4ee8929fb00a4c63774f.tgz	—	—	—	12/30/19, 11:37 AM	45 sec	gcr.io/test-1-263611/devops-image:v0.2

Build triggers watch a repository and build a container whenever code is pushed

Supports Maven, custom builds, and Docker

← Create trigger

1 Select source 2 Select repository

Select source

Choose a repository hosting option

☒ Cloud Source Repository
☐ GitHub
☐ Bitbucket

Continue Cancel

← Create trigger

☒ Select source 2 Select repository 3 Trigger

Select repository

Source: Cloud Source Repository

Filter repositories

☒ default
Cloud Source Repository

Continue Cancel

Trigger settings

Source: Cloud Source Repository Repository: <https://source.developers.google.com/p/rem>

Name (Optional)

Build My Docker Container

Trigger type ?

☒ Branch
☐ Tag

Branch (regex) ?

Matches the branch: master

.*

Build configuration

☒ Dockerfile
Specify the path within the Git repo
☐ cloudbuild.yaml
Specify the path to a Cloud Build configuration file in the Git repo [Learn more](#)



















Dockerfile directory (Optional) ?

The directory will also be used as the Docker build context

/

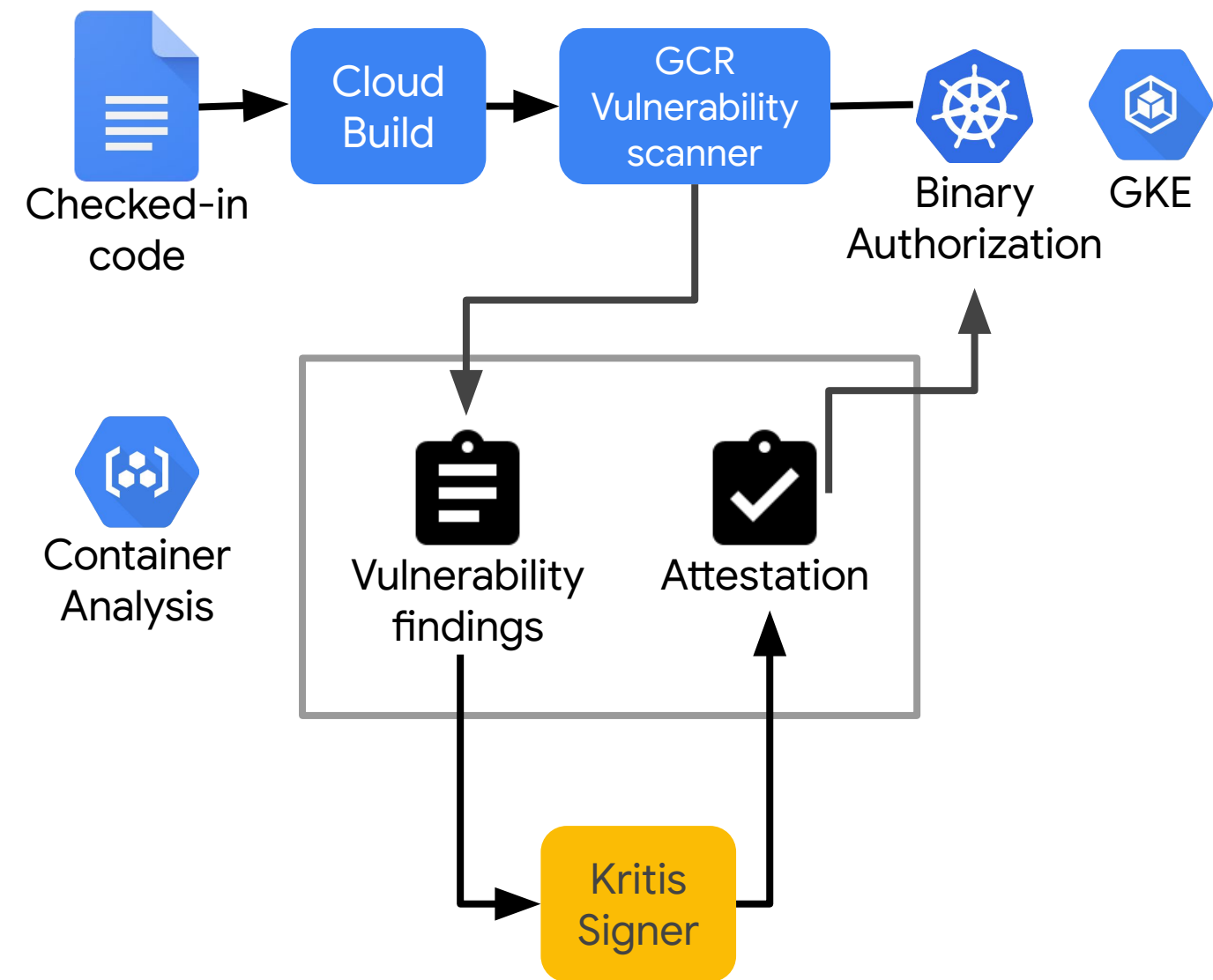
Container Registry is a Google Cloud–hosted Docker repository

- Images built using Cloud Build are automatically save in Container Registry.
 - Tag images with the prefix **gcr.io/your-project-id/image-name**
- Can use Docker push and pull commands with Container Registry.
 - `docker push gcr.io/your-project-id/image-name`
 - `docker pull gcr.io/your-project-id/image-name`

 Container Registry	Repositories REFRESH												
 Images	doug-rehnstrom												
 Settings	<input type="text" value="Filter"/>												
	<table><thead><tr><th>Name ^</th><th>Hostname</th></tr></thead><tbody><tr><td> app-engine-tmp</td><td>us.gcr.io</td></tr><tr><td> converter</td><td>gcr.io</td></tr><tr><td> pets-app</td><td>gcr.io</td></tr><tr><td> petsbook</td><td>gcr.io</td></tr><tr><td> space-invaders</td><td>gcr.io</td></tr></tbody></table>	Name ^	Hostname	 app-engine-tmp	us.gcr.io	 converter	gcr.io	 pets-app	gcr.io	 petsbook	gcr.io	 space-invaders	gcr.io
Name ^	Hostname												
 app-engine-tmp	us.gcr.io												
 converter	gcr.io												
 pets-app	gcr.io												
 petsbook	gcr.io												
 space-invaders	gcr.io												

Binary authorization allows you to enforce deploying only trusted containers into GKE

- Enable binary authorization on GKE cluster.
- Add a policy that requires signed images.
- When an image is built by Cloud Build an “attestor” verifies that it was from a trusted repository (Source Repositories, for example).
- Container Registry includes a vulnerability scanner that scans containers.

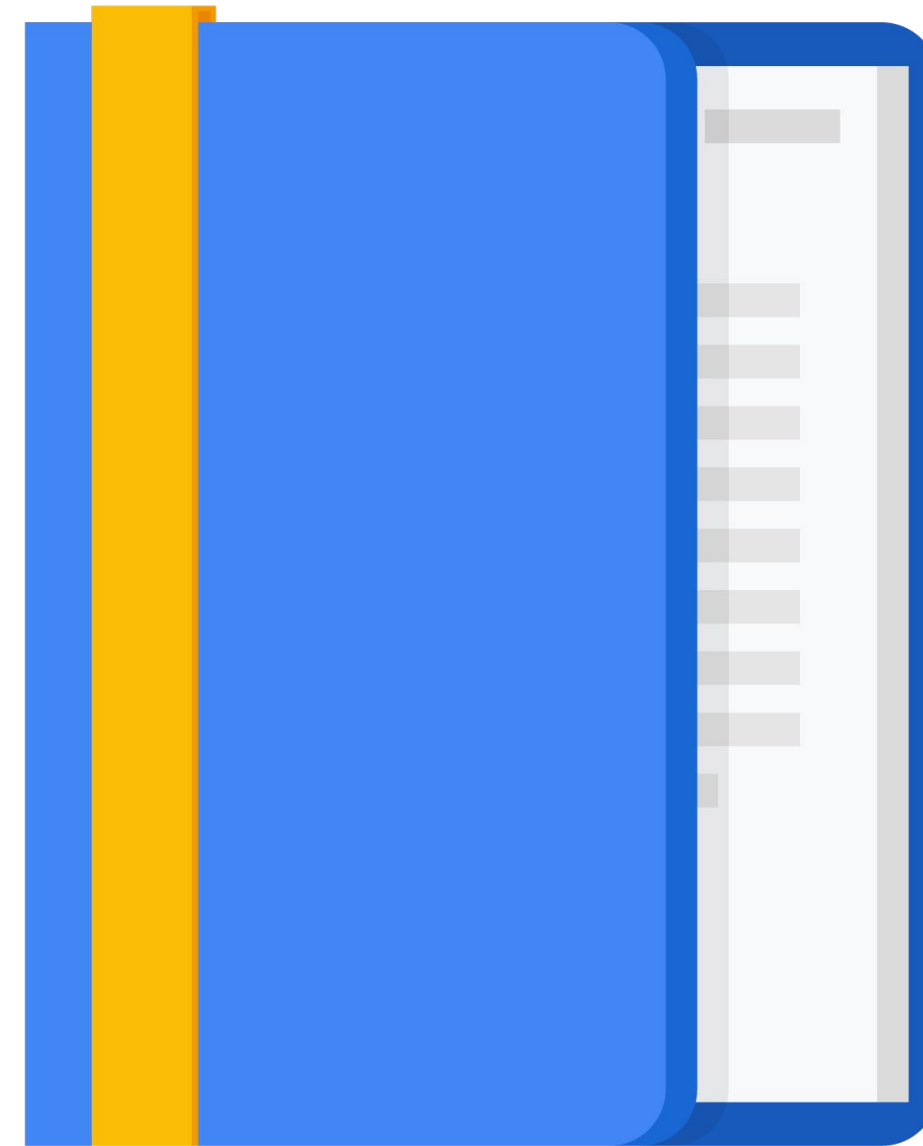


Agenda

Continuous Integration Pipelines

Infrastructure as Code

Lab



Moving to the cloud requires a mindset change

On-Premises

Buy machines.

Keep machines running for years.

Prefer fewer big machines.

Machines are capital expenditures.

Cloud

Rent machines.

Turn machines off as soon as possible.

Prefer lots of small machines.

Machines are monthly expenses.

In the cloud, all infrastructure needs to be disposable

- Don't fix broken machines.
 - Don't install patches.
 - Don't upgrade machines.
 - If you need to fix a machine, delete it and re-create a new one.
- To make infrastructure disposable, automate everything with code:
 - Can automate using scripts.
 - Can use declarative tools to define infrastructure.

Infrastructure as code (IaC) allows for the quick provisioning and removing of infrastructures

- Build an infrastructure when needed.
- Destroy the infrastructure when not in use.
- Create identical infrastructures for dev, test, and prod.
- Can be part of a CI/CD pipeline.
- Templates are the building blocks for disaster recovery procedures.
- Manage resource dependencies and complexity.

- Google Cloud supports many IaC tools.



Cloud Deployment Manager is Google Cloud's native IaC tool

- Define infrastructure using YAML syntax.
- Can create dynamic templates using Python or Jinja.
- Use gcloud to create, update, and delete deployments.

```
resources:
# Configure a VM
- name: devops-vm
  type: compute.v1.instance
  properties:
    zone: us-central1-a
    machineType: zones/us-central1-a/machineTypes/f1-micro
    disks:
      - deviceName: boot
        type: PERSISTENT
        boot: true
        autoDelete: true
        initializeParams:
          sourceImage: projects/debian-cloud/global/images...
# Add VM to default network and give it an external IP
networkInterfaces:
- network: global/networks/default
  accessConfigs:
    - name: External NAT
      type: ONE_TO_ONE_NAT
```

Terraform is similar to Deployment Manager but can be used on multiple public and private clouds

- Considered a first-class tool in Google Cloud.
- Already installed in Cloud Shell.

```
provider "google" {  
  credentials = ""  
  project     = "project name"  
  region      = "us-central1"  
}  
  
resource "google_compute_instance" {  
  name          = "instance name"  
  machine_type  = "n1-standard-1"  
  zone          = "us-central1-f"  
  
  disk {  
    image = "image to build instance"  
  }  
}  
  
output "instance_ip" {  
  value = "${google_compute.ip_address}"  
}
```


Lab

Building a DevOps Pipeline



Cloud Source
Repositories



Cloud Build



Container
Registry

Objectives

- Create a Git Repository
- Create a Simple Python Application
- Test Your Web Application in Cloud Shell
- Define a Docker Build
- Manage Docker Images with Cloud Build and Container Registry
- Automate Builds with Triggers
- Test Your Build Changes

Quiz

Which Google Cloud tools can be used to build a continuous integration pipeline?

- A. Cloud Source Repositories
- B. Cloud Build
- C. Container Registry
- D. All of the above

Quiz

Which Google Cloud tools can be used to build a continuous integration pipeline?

- A. Cloud Source Repositories
- B. Cloud Build
- C. Container Registry
- D. All of the above

Quiz

List some reasons to automate infrastructure creation using code tools like Deployment Manager and Terraform.

Quiz

List some reasons to automate infrastructure creation use code tools like Deployment Manager and Terraform.

- Easier to make dev, test, and prod environments the same
- Easier to change and fix infrastructure over time
- Simplify administration
- Automate provisioning and decommissioning
- Save money

Quiz

What Google Cloud feature would be easiest to use to automate a build in response to code being checked into your source code repository?

- A. Build triggers
- B. Cloud Functions
- C. App Engine
- D. Cloud Scheduler

Quiz

What Google Cloud feature would be easiest to use to automate a build in response to code being checked into your source code repository?

A. Build triggers

B. Cloud Functions

C. App Engine

D. Cloud Scheduler

Review

DevOps Automation

More resources

Google Cloud DevOps solutions

<https://cloud.google.com/devops/>

Deployment Manager

<https://cloud.google.com/deployment-manager/>

Terraform on Google Cloud

<https://cloud.google.com/community/tutorials/getting-started-on-gcp-with-terraform>

