GCP Associate Cloud Engineer Master Cheat Sheet

Overall Exam Structure

Here You can find a link to the BluePrint

- What is the Job Role Description from Google?

```
An Associate Cloud Engineer deploys applications, monitors operations of multiple
projects, and maintains enterprise solutions to ensure that they meet target
performance metrics. This individual has experience working with public clouds and
on-premises solutions. They are able to use Google Cloud Console and the command-
line interface to perform common platform-based tasks to maintain one or more
deployed solutions that leverage Google-managed or self-managed services on Google
Cloud.
```

At a high level they are tasked to do:

- Deploy applications
- Monitor operations of multiple projects
- Maintains enterprise solutions to ensure they meet target performance metrics
- Experience work with public clouds and on-premies solutions
- Able to sue Google Cloud Console and the CLI
- Perform common platform-based task
- Maintains one or more deploy solutions
- Leverages Google-managed ort self-managed services on Google Cloud

Exam Domains

1. Setting up a cloud solution environment
2. Planning and configuring a cloud solution
3. deploying and implementing a cloud solution
4. Ensuring successful operation of a cloud solution
5. Configuring access and security

Here You can find a link to the BluePrint

Exam Question Break Down

Understand	Determine the key question Kicker **&&** Figure out what everything means -question and responses
Eliminate	Get rid of responses that have fake info or other errors **&&** Get Rid of responses that conflict with the key question
Evaluate	Think through all the tradeoffs for remaining responses **&&** consider bot stated and implied dimensions
Choose	Pick exactly the right number **&&** select the best options or eliminate the worst ones
Validate	Make sure your responses answer the key question **&&** make sure your responses don't conflict with any details

---

Key Building Blocks

- Compute Engine -- VMs Disks, Network
- Cloud Functions -- Event-Driven serverless functions
- Kubernetes Engine -- Manged Kubernetes/Containers
- Cloud Storage -- Object Storage and Serving
- Persistent Disk -- VM attached Disk (Hard Disks)
- Cloud Filestore -- Manged NFS Server
- Cloud TPU (TensorFlow) -- Specialized Hardware for ML
- Cloud SQL -- Manged MySQL and PostgresQL

- Cloud Spanner Horizontally Scalable Relational DB
- Cloud Firestore -- Strongly-consistent Serverless Document DB
- Cloud Dataflow -- Stream/batch data processing (Apache Beam)
- Cloud Dataproc -- Manged Spark and Hadoop
- Cloud Pub/Sub -- Global Real-time Messaging
- Google BigQuery -- Data Warehouse/Analytics
- Virtual Private Cloud -- Software Defined Networking
- Stackdriver -- GoogleCloud Monitoring Suite of tools
- Cloud Identity -- Manage Users
- Cloud IAM -- Resources Access Control

Links

- Here You can find a link to the BluePrint
- Here you can find the GCP Solutions Website
- Wiki Google's Tools
- SRE Books

GCP Design and Structure Links

- Data Center Video1
- Data Center Video2
- Regions Map
- Regions and Zones Doc
- Network Map
- Global Load Balancing
- Network Pricing

- Pricing Calculator
- BeyondCorp
- GCP Security Design
- Resource Quotas(Soft Limits)

Account Set Up

Trial Account from GCP is pretty sweet. Links are below for more information.

- GCP Free Trial
- Free Trial Restrictions
- GCP Always Free

Setting up the Account

- You are going to want to create a separate email an google account to set up. Trial Account Link
- You can do this in the browser by using `Incognito Mode`
- Depending on how much data of yourself you have saved into your browser it will be a bit of pain. Just make sure you add all you details properly
- Also set up some type of 2-Step or Two Factor Verification.
- It might sound weird but you want to ensure that you use least privilege while going about. Least Privilege

Exploring the GCP Console

- Link to GCP Console
- Link to Google Cloud Status Dashboard

Setup Billing Export

```
Tools for monitoring, analyzing and optimizing cost have become an important part
of managing development. Billing export to BigQuery enables you to export your
daily usage and cost estimates automatically throughout the day to a BigQuery
```

`dataset you specify. You can then access your billing data from BigQuery.` -- GCP Docs

- Export Cloud Billing data to BigQuery
- Export must be set up per billing account
- Resources should be placed into appropriate projects
- Resources should be tagged with lables
- Billing export is not in real-time
  - Delay is in hours.

Set up a Billing Alert

- GCP Docs on Budgets and Billing Alerts
- To help you with project planning and controlling costs, you can set a budget. Setting a budget lets you track how your spend is growing toward that amount.
- You can apply a budget to either a billing account or a project, and you can set the budget at a specific amount or match it to the pervious month's spend. You can also create alerts to notify billing administrators when spending exceeds a percentage of your budget.

Setup Non-Admin User Access

- Billing Access Docs
- Billing IAM
  - Role: Billing Account User
  - Purpose: Link projects to billing accounts.
  - Level: Organization or Billing Account
  - Use Case: This role has very restricted permissions, so you can grant it broadly, typically in combination with Project Creator. These two roles allow a user to create new project linked to the billing account on which the role is granted.

  - The point here is to tightly control the access of who manages the billing.
  - Remember to enable 2FA and multi-step auth whenever you are given the option for it.

Explore Cloud Shell and Editor

- Cloud Shell Docs

What is Google Cloud Shell

`Google Cloud Shell provides you with command-line access to your cloud resources directly from your browser. You can easily manage your projects and resources without having to install the Google Cloud SDK or other tools on your system. With Cloud Shell the Cloud SDK gcloud CLI and other utilities you need are always available, up to date and fully authenticated when you need them` -- GCP Docs

Highlights

- Web browser access
  - No need for local terminal
  - Automatic SSH Key management
- 5GB of persistent storage
- Easy access to preinstalled tools like:
  - gcloud, bq, kubectl, docker, npm/node, pip/python, ruby, vim, emacs, bash
- Preauthorized and always up-to-date
- Web Preview of Web app running on local port

Helpful Cloud Shell Commands

- `dl <filename>`

Data Flows

aCloudGuru Lecture Video Links

- Mental Models
- Mental Model Example

- Zooming In and Out

IT is all about Data flows. Try to link up in you mind, a mental model for how this data flows through the cloud system and how you move data in systems that you build on top of the cloud platform

Cloud Service Type	Data Flow
Network	Moving
Compute	Processing
Storage	Remembering

Mental Models

- A simplified representation of reality, which is ....
- Used by your mind to anticipate events or draw conclusions
- Systems combine
  - Build larger systems out of smaller ones (using abstractions)
  - Zooming and out

A look at Google Projects

- GCP Projects

Basic Services

GCS: Google Cloud Storage

- Google Storage : Making Data Public
- Google Storage: bucket locations

- In this lab I included a `./lab-content/` directory that has a empty text file and some memes I enjoy to upload, edit permissions, move and delete in the cloud. Feel Free to use them as you like.

GSC: Google Cloud Storage (gsutil cli)

- note use this commands only as a guide
  - Also get used to googling for the google docs on for the google commands
  - their documentation is pretty good and very well organized

```
pwd
ls

gcloud config list

gsutil ls
gsutil ls gs://storage-lab-console/
gsutil ls gs://storage-lab-console/**

gsutil mb --help

gsutil mb -l northamerica-northeast1 gs://storage-lab-cli
gsutil ls

gsutil label get gs://storage-lab-console/
gsutil label get gs://storage-lab-console/ >bucketlabels.json
cat bucketlabels.json

gsutil label get gs://storage-lab-cli/
gsutil label set bucketlabels.json gs://storage-lab-cli/
gsutil label get gs://storage-lab-cli/

gsutil label ch -l "extralabel:extravalue" gs://storage-lab-cli

gsutil versioning get gs://storage-lab-cli/
gsutil versioning set on gs://storage-lab-cli/
gsutil versioning get gs://storage-lab-cli/

gsutil ls gs://storage-lab-cli/
gsutil cp README-cloudshell.txt gs://storage-lab-cli/
gsutil ls gs://storage-lab-cli/

gsutil ls gs://storage-lab-cli/
gsutil ls -a gs://storage-lab-cli/
gsutil rm gs://storage-lab-cli/README-cloudshell.txt
gsutil ls gs://storage-lab-cli/
gsutil ls -a gs://storage-lab-cli/

gsutil cp gs://storage-lab-console/** gs://storage-lab-cli/
gsutil ls gs://storage-lab-cli/
gsutil ls -a gs://storage-lab-cli/
gsutil acl ch -u AllUsers:R gs://storage-lab-cli/Selfie.jpg
```

Google Compute Engine

Starting a Google Compute Engine VM

- Some Commands you will find helpful
- Again don't be afraid of googling for the gcloud commands

```
gcloud config get-value project

gcloud compute instances list

gcloud services list
gcloud services list -h
gcloud services list --enabled
gcloud services list --available

gcloud services list --available | grep compute
gcloud services -h

gcloud compute instances list

gcloud services list

gcloud compute instances create myvm
gcloud compute instances delete myvm
gcloud compute instances list
```

The run down on gcloud

Links to helpful Documentation

- gcloud Overview
- gcloud Syntax
- gcloud Properties
- gcloud Configurations

Important Exam Info

- CLI tool for GCP
- Best friends with `gsutil` and `bq`
  - All share same configuration set via `gcloud config`

  - `gsutil` could have been "`gcloud storage`"
  - `bq` cloud have been "`gcloud bigquery`"

- In general more powerful than the console but less powerful than REST API
- Alpha and Beta versions are available via "`gcloud alpha`" or "`gcloud beta`"

Basic Systnax

- `gcloud <global flags> <service/product> <group/area> <command> <flags> <parameters>`
- always drill down from left to right
- Examples:

  - `gcloud --project myprojid compute instances list`
  - `gcloud --project=myprojid compute instances list`
  - `gcloud compute instances create myvm`
  - `gcloud services list --available`
  - `gsutil ls`
  - `gsutil mb -l northamerica-northeast1 gs://storage-lab-cli`
  - `gsutil label set bucketlabels.json gs://storage-lab-cli`

Global Flags

- `--help`
- `-h`
- `--project <ProjectId>`
- `--account <Account>`
- `--filter` -- sometimes better than using grep
- `--format`

  - Can choose between JOSN, YAML, CSV etc
  - Can pipe out JSON to "`jq`" command for further processing
- `--quiet` or `-q`

Configuration Properties

- Values entered once and used by any command that needs them
- Can be overridden on a specific command with corresponding flag
- Used very often for account, project, region, and zone

  - Set "core/account" or "account" to replace "--account"
  - Set "core/project" or "project" to replace "--project"

- set "compute/region" to replace --region
- set "compute/zone to replace --zone

- Set with `gcloud config set <property> <value>`
- check with `gcloud config get-value <property`
- clear with `gcloud config unset <property>`

Configurations

- Can maintain groups of settings and switch between them
- Most useful when using multiple projects
- interactive workflow to set common properties in a config with `gcloud init`
- list all properties in a configuration with `gcloud config list`
- list all possible configurations with `gcloud config configurations list`
  - `IS_ACTIVE` column shows which one is currently being used
  - other column list account, project, region, zone, and the name of the config
- make new config with `gcloud config configurations create ITS_NAME`
- Start using config with `gcloud config configurations activate ITS_NAME`
  - Or use for just one command with `--configurations=ITSNAME`

GCE In and Out Lab

Helpful links

- [Filters in gcloud]()
- [Instance MetaData Reference]()

Helpful Commands

```
# Check the elected project
gcloud config list

# Show any .ssh folder
pwd
ls
ls -a .ssh

# Get our bearings in Cloud Shell
```

```
whoami
hostname
curl api.ipify.org

# Check that we have nothing running
gcloud compute instances list

# Don't create a default VM
# Cancel: gcloud compute instances create myhappyvm

# Look at how to set the machine type
gcloud compute instances create myhappyvm -h
gcloud compute instances create myhappyvm --help
gcloud compute machine-types list

# See how to filter
gcloud topic filters

# Show some free-tier-eligible options
gcloud compute machine-types list --filter="NAME:f1-micro"
gcloud compute machine-types list --filter="NAME:f1-micro AND ZONE~us-west"

# Set our defaults to Los Angeles
gcloud config set compute/zone us-west2-b
gcloud config set compute/region us-west2

# Start our instance
gcloud compute instances create --machine-type=f1-micro myhappyvm
ping -c 3 myhappyvm
ping -c 3 internalipaddress
ping -c 3 externalipaddress

# Connect to the VM
ssh externalipaddress
gcloud compute ssh myhappyvm

# Get our bearings -- Skip?
whoami
hostname
curl api.ipify.org

# Get back to Cloud Shell
exit
curl api.ipify.org

# Look at the Cloud Shell .ssh files
cd .ssh
ls
cat google_compute_engine.pub
head -n 10 google_compute_engine

# Log back onto the VM
gcloud compute ssh myhappyvm

# See that our key is authorized
cd .ssh
ls
cat authorized_keys
cd ..
```

```
# Check out the metadata
curl metadata.google.internal/computeMetadata/v1/
curl -H "Metadata-Flavor: Google" metadata.google.internal/computeMetadata/v1/
curl -H "Metadata-Flavor: Google"
metadata.google.internal/computeMetadata/v1/project/
curl -H "Metadata-Flavor: Google"
metadata.google.internal/computeMetadata/v1/project/project-id
curl -H "Metadata-Flavor: Google"
metadata.google.internal/computeMetadata/v1/project/attributes/
curl -H "Metadata-Flavor: Google"
metadata.google.internal/computeMetadata/v1/project/attributes/ssh-keys

# Look at some instance metadata
curl -H "Metadata-Flavor: Google"
metadata.google.internal/computeMetadata/v1/instance/
curl -H "Metadata-Flavor: Google"
metadata.google.internal/computeMetadata/v1/instance/name
curl -H "Metadata-Flavor: Google"
metadata.google.internal/computeMetadata/v1/instance/service-accounts/default/
curl -H "Metadata-Flavor: Google"
metadata.google.internal/computeMetadata/v1/instance/service-
accounts/default/email

# See what gcloud knows
gcloud config list

# Look at our buckets
gsutil ls
gsutil ls gs://storage-lab-cli/

# Attempt to delete the VM from within the VM
gcloud compute instances delete myhappyvm

# Exit back to Cloud Shell and actually delete the VM
exit
gcloud compute instances delete myhappyvm
```

GCE Via the Console

Helpful links

- Creating Instances
- Preemptible Instances
- Running Start Up Scripts
- Service Accounts
- It is important to note the granulatity of how google handles permissons.

- Even the SSH keys from the console verses the gcloud cloudshell CLI are different
- You can set the scope for the vm to have API wide access
- also note that the SSH keys via the console are generated with an expiry

What is Security

Helpful Links

- Information Security - Wikipedia
- Google Search: "Public Bucket Breach
- Security by Design
- OWASP Top 10 (2017)

Definition

Information security, sometimes shortened to InfoSec, is the practice of preventing unauthorized access, use, disclosure, disruption, modification, inspection, recording or destruction of information.... Its primary focus is the balanced protection of the confidentiality, integrity and availability of data (also know=n as the CIA triad) while maintaining a focus on efficient policy implementation, all without hampering organization productivity.

-- Wikipedia

- Proper Data Flow
  - You cannot view data you shouldn't
  - You cannot change data you shouldn't
  - You *can* access data you *should*
- Controlling Data Flow
  - Authentication - who are you?
  - Authorization - what are you allowed to do?
  - Accounting - What did you do ?

- (Resiliency) - Make sure the system keeps running

What enables security in GCP?

- Security Products
- Security Features
- Security Mindset
  - Includes Availability Mindset

Principles of Key Security Mindset

- Least privilege
- defense in depth (layer)
- Fail Securely

Key Security Products / Features in GCP -- Authentication

- Identity
  - Humans in G Suite, Cloud Identity
  - Applications and services use Service Accounts
- Identity hierarchy
  - Google Groups
- Can use Google Cloud Directory Sync (GCDS) to pull from LDAP

Key Security Products / Features in GCP -- Authorization

- Identity hierarchy -- Google Groups
- Resource hierarchy (Organization , Folders, Projects
- Identity and Access Management (IAM)
  - Permissions
  - Roles
  - Bindings

- GSC ACLS
- Billing Management
- Network structure and restrictions

Key Security Products / Features in GCP -- Accounting

- Audit / Activity Logs (provided by Stackdriver)
- Billing export
    - Billing Export
    - BigQuery
    - To file as CSV or JSON in a GCS Bucket
- GCS Object Lifecycle Management

Identity and Access Management (IAM) Breakdown

Resource Hierarchy

- Resource
    - Something you create in GCP
- Project
    - Container for a set of related resources
- Folder
    - Contains an number of Projects and Subfolders
- Organization
    - Tied to G Suite or Cloud Identity domain.

Docs

- Resource hierarchy for access control
- G Cloud IAM Overview

IAM Breakdown -- Permissions & Roles

Helpful Links

- IAM Overview
- Understanding Roles
- Understanding IAM custom roles
- Predefined Roles

Permissions

- A permission allows you to perform certain action
- Each one follow the form `Service.Resource.Verb`
- Usually correspond to REST API methods
- Examples:
    - `pubsub.subscriptions.consume`
    - `pubsub.topics.publish`

Roles

- A role is a collection of Permissions to use or manage GCP Resources
- Primitive Roles - Project-level and often too broad
    - viewer is read-only
    - editor can view and change things
    - owner can also control access & billing
- Predefined Roles - Give granular access to specific GCP resources
    - E.g.: `roles/bigquery.dataEditor, roles/pubsub.subscriber`
    - For the exam read through the list of roles for each product! --Think about why each exists
- Custom Role - Project or Org-Level Collections you define of granular permissions

App Engine - Predefined Role Example

Role Name	Role Title	Description
roles/ appengine.AppAdmin	App Engine Admin	Read/Write/Modify access to all application configuration and settings.
roles/ appengine.serviceAdmin	App Engine Service Admin	Read-only access to all application configuration and settings. Write access to module-level and version-level settings. Cannot deploy a new version.
roles/ appengine.deployer	App Engine Deployer	Read-only access to all application configuration and settings. Write access only to create a new version; cannot modify existing versions other than deleting versions that are not receiving traffic
roles/ appengine.appViewer	App Engine Viewer	Read-only access to all application configurations and settings.
roles/ appengine.codeViewer	App Engine Code Viewer	Read-only access to all configuration settings, and deployed source code

Members and Groups

- IAM Overview Docs - Members and Groups

Members

- A member is some Google-Known identity
- Each member is identified by a unique email address
- Can be:
    - user: Specific Google Account: `{G Suit, Cloud Identity, Gmail or validated email}`

  - `serviceAccount`: Service account for apps/services
  - `group` : Google group of users and service accounts
  - `domain`: whole domain managed by GSuite or Cloud Identity
  - `allAuthenticatedUsers -- *ANY* Google account or service account

Groups

- " A Google Group is a named collection of google accounts and service accounts".
- "Every group has unique email address that is associated with the group."
- You can never act *as* the group
  - But membership in a group can grant capabilities to individuals
- Use them for everything!
- Can be used for owner when within an organization
- *Can* nest groups in an organization
  - Example: one group for each department, all those in group for all staff

IAM Breakdown - Policies

- IAM Overview Docs - Policies
- Granting, Changing and Revoking Access to Resources
- "gcloud add-iam-policy-binding"

Policies

- A Policy binds Members to Roles for some scope of Resources
- Answers: Who can do what to which things?
- Attached to some level in the Resource Hierarchy
  - Organization, Folder, Project, Resource
- Roles and Members listed in policy, but Resource identified by attachment
- Always additive ("Allow") and never subtractive (no "Deny")
  - "Child policies cannot restrict access granted at a higher level

- Use groups!

Managing Policy Bindings

- Can use `get-iam-policy`, edit the JSON/YAML file, and `set-iam-policy` back
- But you should use:
- `gcloud [GROUP] add-iam-policy-binding [RESOURCE-NAME] --role [ROLE-ID-TO-GRANT] --member user: [USER-EMAIL]`
- `gcloud [GROUP] remove-iam-policy-binding [RESOURCE-NAME] --role [ROLE-ID-TO-REVOKE] --member user: [USER-EMAIL]`
- Atomic operations are better because changes:
  - are simpler, less work, and less error-prone (than editing the yaml or json)
  - *Avoids race conditions, so changes cna happen simultaneously*

IAM Breakdown -- Wrap Up

- IAM Overview Documentation
- IAM FAQ
- IAM Best Practices

Billing Access Control

Helpful Links

- Overview of Cloud Billing Access Control
- Apply for monthly invoiced billing

Billing Accounts

- A billing account represents some way to pay for GCP service usage
- Type of Resource that lives outside of Projects
- Can belong to an Organization (i.e. be owned by it)

  - Inherits Org-level IAM policies
- Can be linked to projects but:
  - Does not own them
  - No impact on project IAM

Billing Account User

Role: Billing Account User **Purpose:** link project to billing accounts **Level:** organizations or billing account. **Use Case**: This role has very restricted permissions, so you can grant it broadly, typically in combinations with Project Creator. These two roles allow a suer to create new projects linked to the billing account on which the role is granted.

Billing IAM Roles

Role	Purpose	Scope
Billing Account Creator	Create new self-serve billing accounts.	Org
Billing Account Admin	Manage billing accounts (but not create them).	Billing Account
Billing Account User	Link projects to billing accounts	Billing Account
Billing Account Viewer	View Billing account cost information and transactions	Billing account
Project Billing manager	Link/unlink the project to/from a billing account	Project

Monthly Invoiced Billing

- Get billed monthly and pay by invoice due date
- Can pay via check or wire transfer

- can increase project and quota limits
- billing administrator of org's current billing account contacts Cloud Billing Support
    - to determine eligibility
    - to apply to switch to monthly invoicing
- Eligibility depends on
    - Account age
    - typical monthly spend
    - country

Networking in GCP

Routing Overview

- About Software Defined Networking
- More general than the OSI 7 Layer Model
- Not about any particular routing scheme
- Sets the stage for routing tables and routes
- Routing here is about the *data flow*

Helpful Links

- OSI Model of Networking - Webopedia
- Routing - Wikipedia

Routing - To Google's Networking (Premium Routing Tier)

Links

- Premium Routing Tier Blog Post
- Hot potato / cold potato routing

Getting data to Google's Network

- Take a look at this gif to get an idea:

Routing - To the Right Resource

- GCP Cloud Load Balancing Overview
- Latency Reduction -- solved with **Cross Region Load Balancing** with Global Anycast IPs
  - Use servers physically close to client
- Load Balancing -- solved by using **Cloud Load Balancer**
  - separate from auto-scaling
- System Design -- Solved by using **HTTP(S) Load Balancer** (with URL Map)
  - Different servers may handle different parts of the system
  - Especially when using microservices (instead of a monolith)

Unicast vs Anycast

- Unicast: there is only one **uni**que device in the world that can handle this; send it there
- Anycast: there are multiple devices that could handle this; send it to **any** one; but ideally the closest

Layer 4 vs Layer 7

- TCP is usually called Layer 4
  - it works solely with IP addresses
- HTTP and HTTPS works at layer 7
  - these know about URLs and paths
- Each layer is built on the one below it
- Therefore:
  - To route based on URL paths routing needs to understand Layer 7
  - Layer 4 cannot rout based on the URL paths defined in Layer 7

So what about DNS?

- Name resolution with DNS can be the first step in routing
- But, that comes with a number of problems:
  - Layer 4! -- Cannot route L4 based on L7 URL Paths
  - Chunky -- DNS queries often cached and reused for huge client sets
  - sticky -- DNS lookups "lock on" and refreshing per request has a high cost
    - Extra latency because each request includes another round-trip!
    - More money for additional DNS request processing
  - Not robust -- relies on the client always doing the right thing
    - spoiler alert they don't!

Routing -- Among Resources (VPC)

Helpful Links

- AcloudGuru -- Primer on Subnets and CIDRs
- Classless inter-domain routing CIDR Blocks on Wikipedia
- Private Network on Wikipedia

Getting data from one resource another

- VPC (global) is a Virtual Private Cloud -- Your private SDN space in GCP
  - Not just resource to resource -- also manges the doors to outside & peers
- subnets (regional) create logical spaces to contain resources
  - all subnets can reach all others -- globally without any need for VPNs
- Routes (global) define "next hop" for traffic based on destination IP
  - routes are global and apply by Instance level tags, not by subnet
  - no route to the internet gateway means no such data can flow
- firewall rules (global) further filter data flow that would otherwise route
  - all firewall rule are global and apply by Instance-level tags or service account
  - default firewall rules are restrictive inbound and permissive outbound

VPC - Automode Lab

- subnet ranges

VPC - Custom Mode Lab

Part 1

- VPC Documentation

Part 2

- Understanding Iam Custom Roles
- Creating and Managing Custom Roles
- Service Accounts Overview
- Creating and enabling service accounts

Part 3

- Firewall Rules Overview
- Configuring Network Tags
- Filtering by Service Account vs. Network Tag
- Updating manged Instance Groups (e.g. Rolling update)
- acloudguru: Thead for editing instance in the web console

Add tag to VM via `gcloud`

```
gcloud compute instances add-tags frontend-instance-group-knf1 --zone=us-west1-c -
-tags=open-ssh-tag
```

Challenge Lab

- A Lab Challenge built on the previous step

Desired Result -- SETUP

- Two-tier setup: frontend and back, each auto scaled across 2+ zones
- Use ICMP (ping) to represent allowed traffic
- Frontend:
  - accepts incoming traffic from internet
  - can connect outbound to backend and internet
- Backend:
  - only accepts incoming from frontend or other backend
  - no outbound expect other backend

Desire Result -- Validation

- From Cloud shell or your computer:
  - can ping frontend instances
  - cannot ping backend instances
- when SSHed to a frontend instances
  - can ping the backend
  - can ping google.com
- When SSHed to a backend instance
  - cannot ping the frontend
  - cannot ping google.com
  - can ping other backend instances

My Solution

- Start with 2 sets of managed instance groups like in the first lab
- create a `open-ssh` firewall rule with `500` priority and `0.0.0.0/0`
- **THIS WILL NOT WORK AS SHOWN IN THE VIDEO**
- delete the firewall rule
- open the gcloud terminal
- set your gcloud config : `gcloud config set project <project-id>`
- **I Will come back to this at a later date to finish**

Networking Exam Tips

- Practice CIDR Blocks
  - /16, /24, /28, etc
  - use https://cidr.xyz/
  - CIDR /16 is the same as 255.255.0.0
  - CIDR /24 is the same as 255.255.255.0

  - CIDR /32 is the same as 255.255.255.255
- Practice / Learn common ports
  - **HTTP == 80**
  - **HTTPS == 443**
  - **SSH == 22**
  - ICMP == NO PORT
  - RDP == 3389
  - SQL == 1443
  - MySQL == 3306
  - Postgres == 5432

Subnet CIDR Ranges

- In GCP you *can* edit a subnet to increase its CIDR range
- No need to recreate subnet or instances
- New range must contain old range (i.e. old range must be a subset)

Shared VPC

- Shared VPC
- In an Organization, you can share VPCs among Multiple projects
  - **Host project:** One project owns the shared VPC
  - **Service project:** other projects granted Access to use all/part of Shared VPC
- Lets multiple projects coexist on same local network (private IP space)
- This would let a centralized team manage network security

Google Kubernetes Engine

- Kubernetes Cheat Sheet

Helpful links

- GKE Overview
- GKE Concepts
- Cluster Architecture
- Pods
- Deployments
- StatefulSets
- DaemonSets
- Services and Service Types
- HTTP(s) load balancing with Ingress
- GKE Storage

Introduction

- Learn Kubernetes

Course Outline

- K8s Big Picture
- K8s App Architecture
- K8s Networking
- K8s Storage
- From code to K8s
- K8s deployments
- scaling K8s Apps

- RBAC and Admision Control
- Other Kubernetes stuff

Kubernetes Big Picture

A Kubernetes Primer

- Cloud Native Apps! Cloud Native on Wikipedia

The Kubernetes API

- Everything in K8s is defined here (resources)
- REST
- CRUD
- HTTP Methods (GET POST etc)
- use `kubectl` to post `yaml` to the api server
- Update desired state and current state matches desired state
- API is broken up into parts
    - SIGs (special interest groups)look after the API

Kubernetes Objects

- containers are wrapped up in a **Pod**!
    - Contains one or more containers
    - smallest unit deployable in K8s
    - Atomic unit of scheduling
    - Object on the cluster
- **Deploy**
    - Object on the cluster
    - defined in **apps/v1** API group

  - scaling
  - rolling updates

Getting a cluster

- I will be using GCP!
- Just go in to GCP
  - Click on GKE and make a cluster

App Architecture

Theory

- Kubernetes objects can spin up cloud resources
- code is wrapped in containers which is wrapped in *deployments*
- Deployment is an object kin Kubernetes
- secrets is an object
- all of this can be managed through the Kubernetes API server

Sample App

- I forked this from nigelpoulton

  https://github.com/DRpandaMD/k8s-sample-apps

Kubernetes Networking

Common Networking Requirements

- big scalable networks
- service discovery!
- highly dynamic networks are the new normal
- endpoints added and removed from network based on active scaling up and down

- also covers failure, rolling updates, etc etc.

Networking in the Sample App

- Take a look at the `k8s-sample-apps/mysql-wordpress-pd/wordpress-deployment.yaml`
- they are in a declarative format. which is great for K8s and Documentation.

Kubernetes Networking Basics

- All nodes can talk
- All Pods can talk without NAT
- every pod gets its own IP

Kubernetes Service Fundamentals

- enter Stable network abstraction
- every service gets a name and IP -- they are stable
- services are auto registered in coreDNS
- use labels for pod traffic routing
- end point object maps ips from pods and labels

Service Types

- Cluster Ip is the default:
  - gets own IP
  - own accessible from within the cluster
- NodePort:
  - Gets cluster wide **PORT**
  - Accessible from outside the cluster
- LoadBalancer
  - integrates with public cloud platform (AWS, Azure GCP)

The Service Network

- Separate from:
  - Node network
  - Pod network
- "kube-proxy"
- IPTABLES Mode:
  - Default since K8s 1.2
  - Doesn't Scale well
  - not really designed for Load Balancing
- IPVS Mode
  - Stable (GA) since Kubernetes 1.11
  - Uses Linux kernel IP Virtual Server
  - native layer-4 load balancer
  - supports more algorithms (than just RoundRobin)

Networking Demo

- `kubectl get nodes` used to list nodes in the cluster
- `kubectl apply -f ./ping-deploy.yml` creates a deployment based on the .yml
- `kubectl get deploy` gets the deployment object here its just pingtest
- `kubectl get pods -o wide` shows the wide landscape of how the pods are set against the nodes
- `kubectl get nodes -o jsonpath='{.items[*].spec.podCIDR}'` shows another output of just the CIDR blocks the nodes are sitting on.
- `kubectl exec -it pingtest-6bcdfcdc5b-8t6zg bash` just like docker this will connect into said pod with bash (these pods will change)
- run `apt-get install iputils-ping curl dnsutils iproute2 -y` to install some utilities that the base container image does not come with and will needed to run some tests
  - you will have to run `apt-get update` FIRST!!
- inside the pod you can `ping` the other pods it will work
- exit the pod

- `kubectl apply -f ./simple-web.yml` deploys the new service
- `kubectl get svc` will list the service
- jump back into the pod you had the ip tests
    - if you don't remember (Like I did) just install the tools again on the new pod
- `curl hello-svc:8080` which is the format of `<SERVICE_NAME: Pod_Port>`
    - this will dump out the HTML text of the pod
- `curl <Any_Node_IP>:30001` which is the format of `<NODE_IP: NODE_PORT>`
    - This will ALSO dump out the html text of the pod
- Now to set up the public load balancer
- `kubectl apply --f lb.yml` to deploy the new lb service
- `kubectl get svc --watch` to watch the lb-svc and get an external IP
- copy and paste the external IP and put it into a browser and the web page should render
    - "Hello Cloud Gurus"

Kubernetes Storage

Kubernetes Storage Big Picture

- Kubernetes Volumes abstract the data from the pods -- decoupling storage from Pods
- File and Block are First Class citizens in Kubernetes
    - Standards based
    - pluggable backend
    - Rich API
- What are your storage requirements
    - Speed?
    - Replication?
    - Resiliency
    - ...etc
- This all gets handled by the storage backend

- Containers Storage Interface puts the storage into the hands of the Persistent Volume Subsystem.
    - Persistent Volume (PV) -- storage
    - Persistent Volume Claim (PVC) -- ticket to use the PV
    - Storage Class (SC) -- makes it dynamic

The Container Storage Interface (CSI)

- maintained at https://github.com/container-storage-interface/spec
- decouples storage from the main code base of K8s

The kubernetes PersistentVolume Subsystem

- GCP Persistent Disks
    - Standard
    - SSD
- GCEPersistentDisk PlugIn
- to use a PV needs to PV Claim

Dynamic Provisioning with StorageClasses

- using storage classes allow for the dynamic creation of PV and binding

Storage Classes Demo

- It is important to note that by default Kubernetes (GKE) will make a `default` storage class
- you can find it by `kubectl get sc`
    - It will list the storage class
- you will need a secret for the database
    - `kubectl create secret generic mysql-pass --from-literal=password=<PASSWORD>`
- run `kubectl apply -f ./mysql-deployment.yaml`
- use `kubectl describe pv` && `kubectl describe pvc` to get details about the Persistent Volume and the Persistent Volume Claim

- clean up by:
  - `kubectl delete deployment <delpoyment_name>`
  - `kubectl delete pvc <pvc_name>`
  - `kubectl delete pv <pv_name>`
  - check the deployment, pods, pvc, and pv and it should all be cleaned up

From Code to Kubernetes

- apps start with code
- put it into a container image
- put the image in to a registry
- then use kubernetes to create an object that kubernetes can uses
  - deployments!!
- *coding* then *docker* then *kubernetes*

Demo

- Examine the directory code and dockerfile
- `docker image build -t drpandamd/node-app:0.1 .`
- `docker image push drpandamd/node-app:0.1`
- I went ahead and made some changes to his code and made it my own and deployed the deployment and the web-service
- for clean up you will want to make sure you kill the deployment and services made in kubernetes

Kubernetes Deployments

- deployments wrap up your pods
- make your changes in the deployment post it against the API server then stick it back into git
- with replicas set to 3:
  - rolling updates :

        - maxSurge: 1 -- will have one extra pod than the desired state during the update
        - maxUnavailable: 0 -- will ensure we never go below 3
    - This will roll out new pods with the new version and kill off the old pods one by one until all pods in the desired state of 3 have the new version. -- This will keep your app up while rolling the update
- minReadySeconds: 300 will give 5 minutes between each new pod -- this helps give time to ensure the pods stand up

Deployment Demo

- `kubectl get nodes`
- `kubectl version -o yaml`
- `kubectl apply -f deploy.yml`
- `kubectl get deployment test --watch`
- `kubectl describe deployment test`
- `kubectl get rs` -- rs here means replica set
- `kubectl rollout history deployment test`
- `kubectl apply -f deploy.yml --record`
- `kubectl rollout undo deploy test`

Scaling Applications Automatically

- Increasing load (cpu memory / connections messages in the queue)
- increase in pods to trigger more nodes
- Horizontal Pod AutoScaler
- Cluster AutoScaler

Horizontal Pod AutoScaler

- will scale pods automatically horizontally across nodes

HPA Demo

- to match up what Nigel has, i set my cluster to autoscale min 3 nodes max 6 nodes
- `kubectl get pods --namespace acg-ns` to check the pods made
- to run the load generator

```
kubectl run -i --tty loader --image=busybox /bin/sh

while true; do wget -q -O- http://acg-lb.acg-ns.svc.cluster.local; done
```

- `kubectl get hpa --namespace acg-ns` to watch the HPA do its thing
- `kubectl get pods --namespace acg-ns` to list the pods out in the name space
- `kubectl get deploy --namespace acg-ns -o yaml` to see the yaml of the deployment and how the deployment was changed based on information provided by the HPA
- make sure to clean up after you are done

Cluster Autoscaler

- make sure you have your pods listed in the yaml with resource requests!
- dont mess with the pools via the CLI or Console
- check your specific cloud for support (luckily GCP is on top of it)
- test for performance on BIG clusters

RBAC and Admission Control

- we want to control who access to the API server
  - actions are REST based
  - and CRUD (Create, Read, Update, Delete)
- via HTTPS
- AuthN -- prove your ID
- AuthZ -- is the user allowed to perform action
- Addition Control -- Mutate and validation
- Schema validation
- RBAC
  - Enabled since 1.6
  - GA since 1.8
  - Deny-by-default

Authentication

- Kubernetes does not do users
  - manage users externally
- Service Accounts
  - do happen in Kubernetes
  - for system components
  - managed by kubernetes
  - you should be using these and should manage them.

Authorization

- Basically : **who** can perform which **actions** on which **resources**
  - can *sam* execute a *create* on a *deployment*
  - can *bob* execute a *delete* on a *pods*
- out of the box K8s has Default Users -- its how you been doing everything in the cluster as it stands
  - it is too powerful for production
- You will need to create some **Roles & RoleBindings** for least privilege
- new roles and bindings are -- DENY ALL by default
  - so you will have to open up individual roles
- You can bind by *Role* or by *ClusterRole*
  - *role* is namespaced
  - *ClusterRole* is for the whole cluster
  - what you can do is create *ClusterRoles* then in the *RoleBinding* you can add namespaces to them.
  - That way you can make a few *ClusterRoles* but get granular in the *RoleBinding*
  - This prevents you from doing extra work with roles and assigning them

Admission Control - not yet GA

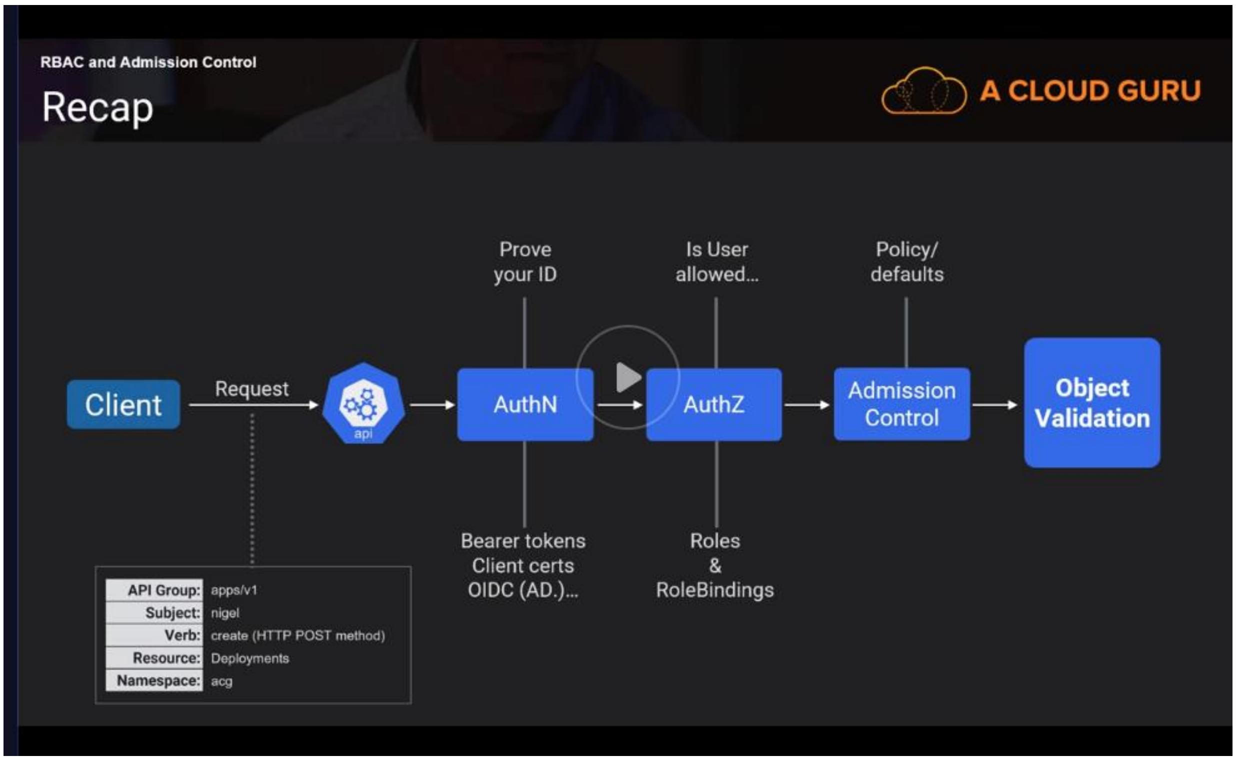
- webhooks with external admissions controllers
- mutating
- validating

RBAC Demo

- [Github Code]()
- he is using kops on AWS

RBAC Recap

Here is an overview I snipped from the mad lads over at acloudguru



API Group:	apps/v1
Subject:	nigel
Verb:	create (HTTP POST method)
Resource:	Deployments
Namespace:	acg

Other Kubernetes Stuff -- Stuff Nigel thinks is important

- DaemonSets - pod one runs every node
- StatefulSet -

- Job - specified number of pods to complete a specific task
- CronJob - scheduled Job
- PodSecurityPolicy
- Pod resource requests and limits
- ResourceQuotas - set limits against namespaces
- CustomResourceDefinition - adds some extensibility to Kubernetes

What is Next

- The Kubernetes Book
- Docker Dive Book
- Community KubeCon
- the podctl podcast
- kubernetes podcast from google
- kubernetes certfied administrator
- serverless Knative and openfaas
- serverice meshes
- prometheus
- API

Services Breadth

Google Cloud Products

Compute

Google Compute Engine

Google Kubernetes Engine

Google App Engine

Google Cloud Functions

Storage

Local SSD

Persistent Disk

Cloud FileStore

Cloud Storage(GCS)

Databases

Cloud SQL

Cloud Spanner

Cloud Spanner Instances

BigQuery

BigQuery Under the Hood

Cloud BigTable

Datastore

Datastore Queries

Firebase DBs

Data Transfer

Data Transfer Appliance

Storage Transfer Service

External Networking

Google Domains

Cloud DNS

Static IPs

Cloud Load Balancing

Cloud CDN

Internal Networking

Virtual Private Cloud

Google Cloud Hybrid Connectivity (Cloud Interconnect)

Cloud VPN

Dedicated Interconnect

Cloud Router

CDN Interconnect

Machine Learning / AI

- *Note: ACG calls it the ML Engine at some point google renamed it the 'AI Platform'*

Cloud Machine Learning Engine (AI Platform)

Case Study: Coca-Cola

Cloud Vision API (Vision AI)

Cloud Speech API (Speech-To-Text)

Cloud Natural Language API (Natural Language)

Cloud Translation API (Translation)

Dialogflow

Cloud Video Intelligence API(Video AI)

Cloud Job Discovery (Cloud Talent Solution)

Big Data and IoT

Big Data Lifecyle

Cloud Internet of Things(IoT) Core

Cloud Pub/Sub

Cloud Dataprep

Data Wrangling vs ETL

Cloud Dataproc

Cloud Dataflow

Dataflow Shuffle

Cloud Datalab

Jupyter Notebook

Cloud Data Studio

Cloud Genomics

Identity and Access - Core Security

- Handles the AAA Model - Authentication, Authorization and Accounting or Auditing

GCP Security Overview

Roles

Cloud Identity and Access Management (IAM)

Hierarchical Access Control

Service Accounts

Cloud Identity

Security Key Enforcement

Resource Manager

Resource Manager Hierarchy

Cloud Identity-Aware Proxy (IAP)

Cloud Audit Logging

Security Management - Monitoring and Response

Cloud Armor

Cloud Security Scanner

Cloud Data Loss Prevention (DLP) API

Event Threat Detection (ETC)

Cloud Security Command Center(SCC)

Getting started with Cloud SCC (blog post)

What is a SIEM (YouTube Video)