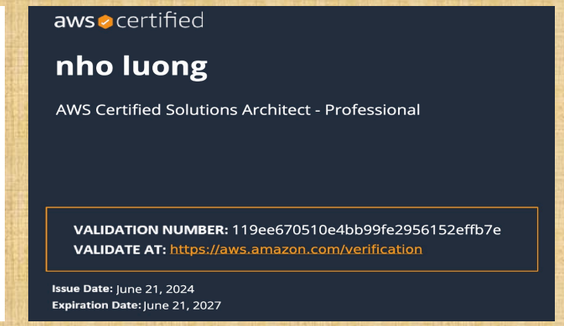
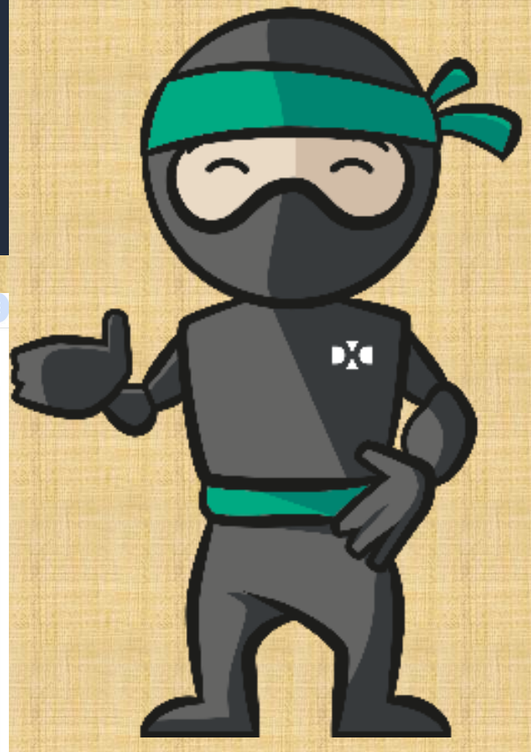
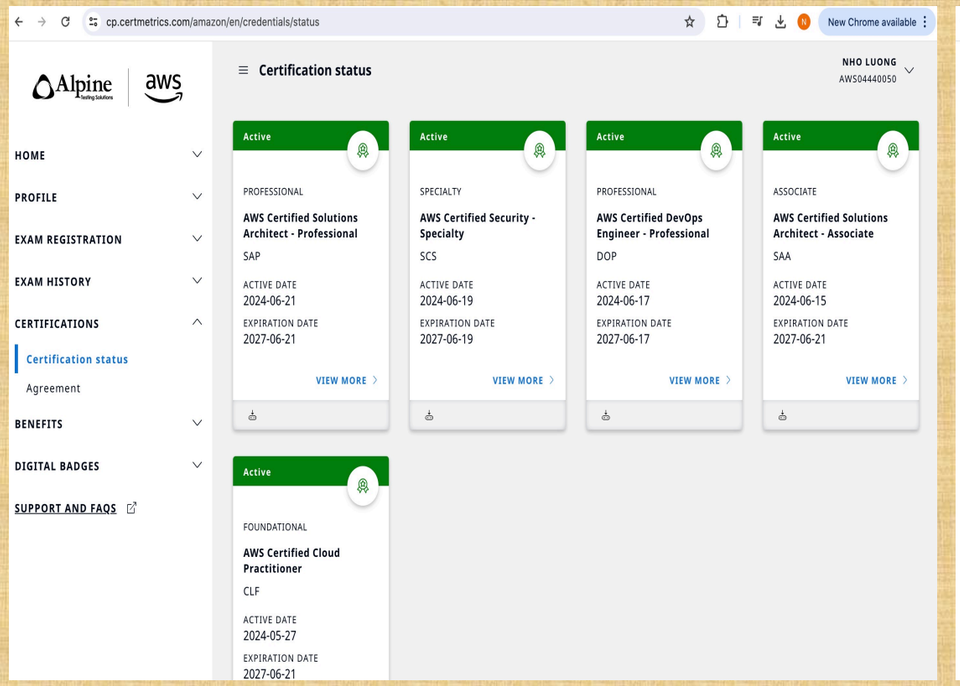
Terraform on Google GKE Kubernetes

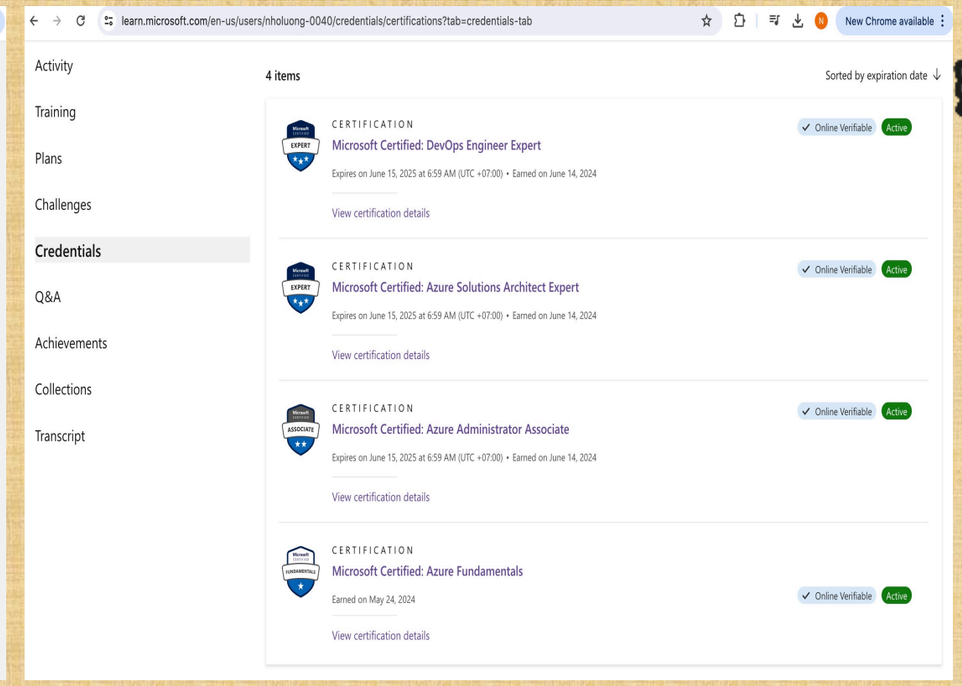
Author: Nho Luong

Skill: DevOps Engineer Lead









Author: Nho Luong

Skill: DevOps Engineer Lead

Terraform on Google Kubernetes Engine

Terraform Fundamentals

Terraform CLI Install	1
Terraform Commands	2
Terraform Language Basics	3
Meta-Argument Provider	4
Input Variables & Output Values	5
Meta-Argument count	6
Terraform Data sources	7
Meta-Argument for_each	8
Remote Backend Remote State Data source	9

40
Real-World Demos

Step by Step
GitHub
Documentation

GKE Clusters

01	GKE Public Standard Cluster
02	GKE Private Standard Cluster
03	GKE Private Standard Cluster (Private Endpoint)
04	GKE Private Autopilot Cluster

GKE Scaling

01	GKE Cluster Autoscaling
02	Horizontal Pod Autoscaling
03	Vertical Pod Autoscaling

Author: Nho Luong

Skill: DevOps Engineer Lead

Terraform on Google Kubernetes Engine

GKE Storage

GCP Compute Engine Persistent Disk CSI Driver	1
GCP Cloud SQL	2
GCP Cloud Storage FUSE CSI Driver	3
GCP Filestore CSI Driver	4

GKE Gateway API

Gateway API Basics (Cloud Regional Application LB)	1
Gateway API Static IP (GCP External IP Address)	2
Gateway API SSL (Kubernetes Secrets)	3
Gateway API SSL (GCP Certificate Manager)	4

40
Real-World Demos

Step by Step
GitHub
Documentation

GKE Gateway API

05	Gateway API SSL (HTTP to HTTPS Redirect)
06	Gateway API Routing (Context Path Routing)
07	Gateway API Routing (Domain Name Routing)
08	Gateway API Routing (Traffic Splitting)
09	Gateway API (Health Checks + Session Affinity)
10	Cloud Domains & Cloud DNS
11	Gateway API Prod Grade SSL (Certificate Manager + Cloud DNS + Cloud Domains)
12	Gateway API Prod Grade SSL (Certificate Manager + Cloud DNS + AWS Route53)
13	Gateway API Global (Cloud Global Application LB)

Author: Nho Luong

Skill: DevOps Engineer Lead

Terraform on Google Kubernetes Engine

GKE Infra DevOps

Terraform Modules	1
GKE Cluster (Custom Terraform Module)	2
GKE Infra DevOps (GitHub + Cloud Build)	3

GKE Workload DevOps

Kubernetes Deployment (Custom Terraform Module)	1
GKE Workload DevOps (GitHub + Cloud Build)	2
GKE Continuous Integration (GitHub + Cloud Build + Artifact Registry)	3
GKE Continuous Delivery (GitHub + Cloud Build)	4

40
Real-World Demos

Step by Step GitHub Documentation

Author: Nho Luong

Skill: DevOps Engineer Lead

GitHub Step-by-Step Documentation

- terraform-on-google-kubernetes-engine
  - 01-Terraform-Install-Tools
  - 02-Terraform-Commands
  - 03-Terraform-Language-Basics
  - 04-Terraform-MetaArgument-provider
  - 05-Terraform-Variables-Output-Values
  - 06-Terraform-MetaArgument-count
  - 07-Terraform-Datasources
  - 08-Terraform-MetaArgument-foreach
  - 09-GKE-Public-Standard-Cluster
  - 10-Kubernetes-Resources-yaml
  - 11-Kubernetes-Resources-Terraform
  - 12-GKE-Private-Standard-Cluster-Autoscaler
  - 13-GKE-Horizontal-Pod-Autoscaling
  - 14-GKE-Vertical-Pod-Autoscaling
  - 15-GKE-Private-Standard-Cluster-private-endpoint
  - 16-GKE-Private-Autopilot-cluster
  - 17-GKE-Storage-Persistent-Disks
  - 18-GKE-Storage-CloudSQL
  - 19-GKE-Cloud-Storage-FUSE-CSI
  - 20-GKE-Storage-Filestore

40 Real-World Demos

- 21-GKE-Gateway-API
  - 01-GKE-LB-Gateway-API-Basic
  - 02-GKE-LB-Gateway-API-StaticIP
  - 03-GKE-Gateway-API-Selfsigned-SSL-k8sSecrets
  - 04-GKE-Gateway-API-Selfsigned-SSL-CertManager
  - 05-GKE-Gateway-API-HTTP-to-HTTPS-Redirect
  - 06-GKE-Gateway-API-ContextPath-Routing
  - 07-GKE-Gateway-API-Domain-Routing
  - 08-GKE-Gateway-API-Traffic-Splitting
  - 09-GKE-Gateway-API-HealthChecks-SessionAffinity
  - 10-Cloud-Domains-and-Cloud-DNS
  - 11-GKE-Gateway-API-ProdSSL-CloudDNS
  - 12-GKE-Gateway-API-ProdSSL-ExternalDomainProvider
  - 13-GKE-Gateway-API-Global-LB
- 22-Terraform-Modules
- 23-GKE-Infra-Custom-Terraform-Modules
- 24-GKE-Infra-DevOps-CloudBuild-GitHub
- 25-GKE-Workloads-Custom-Terraform-Modules
- 26-GKE-Workloads-DevOps-CloudBuild-GitHub
- 27-GKE-App-Continuous-Integration
- 28-GKE-App-Continuous-Delivery

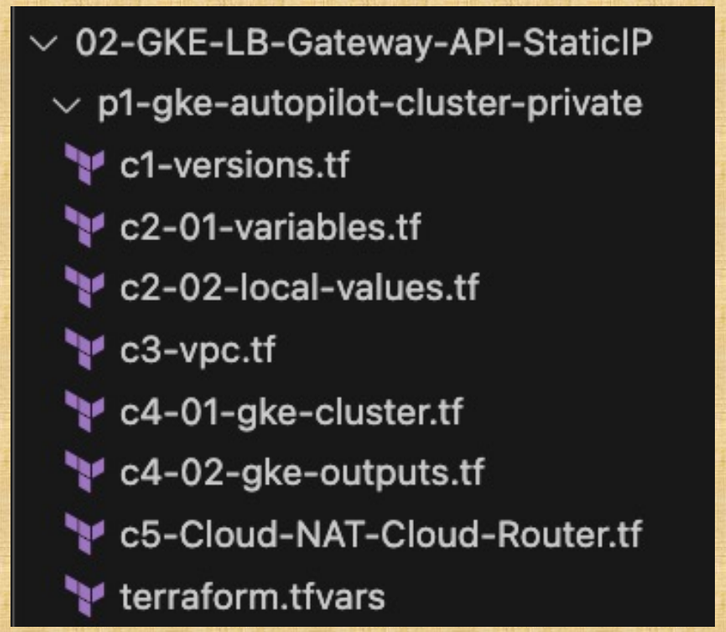
Author: Nho Luong

Skill: DevOps Engineer Lead

How are Terraform Configs organized ?

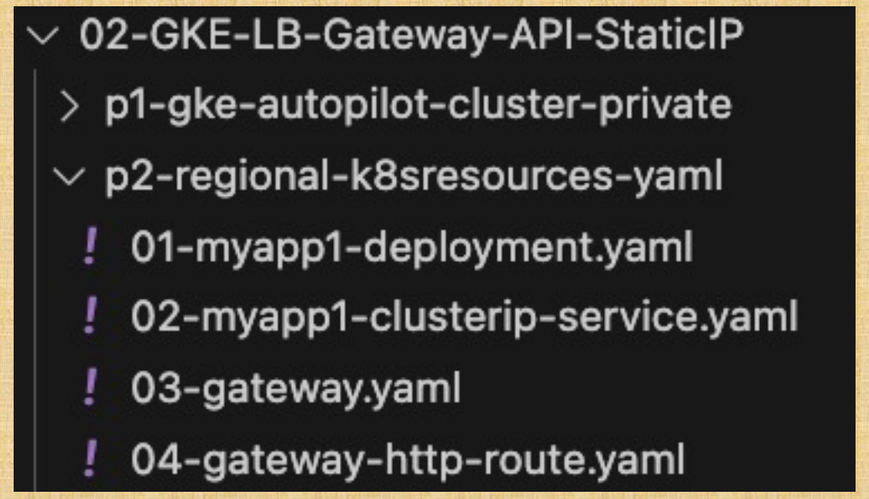
Project-1

GKE Cluster
Terraform Manifests



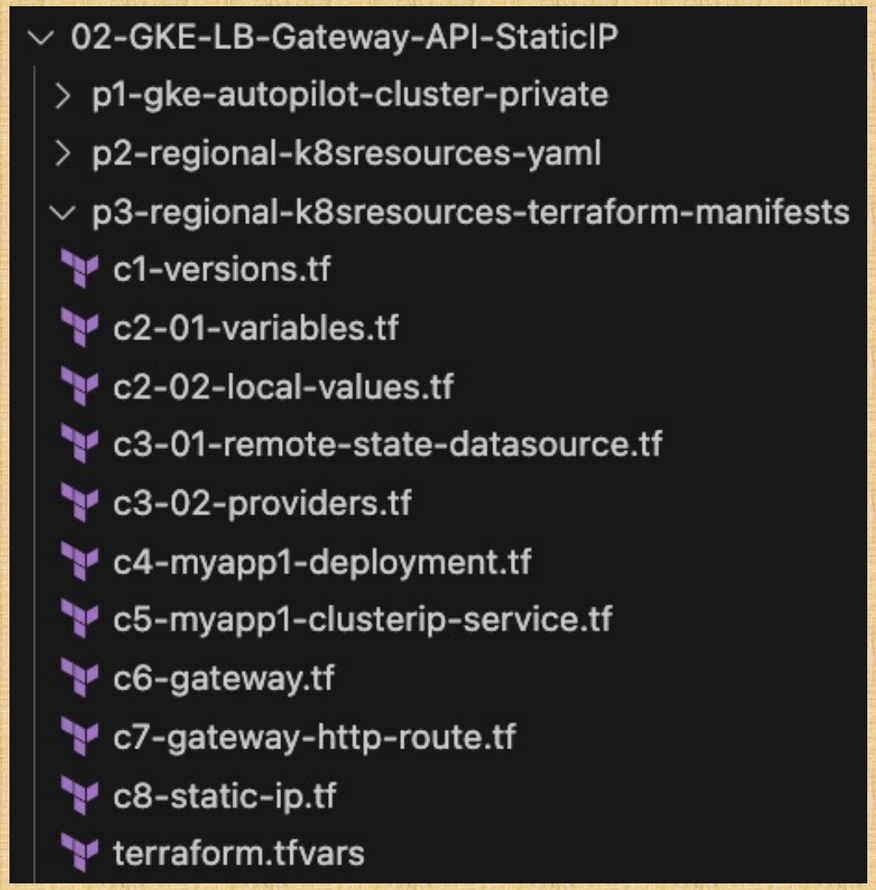
Project-2

Kubernetes
YAML Manifests



Project-3

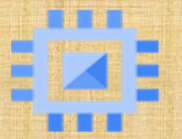
Kubernetes Terraform
Manifests



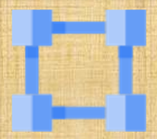
Author: Nho Luong

Skill: DevOps Engineer Lead

Terraform on Google Cloud



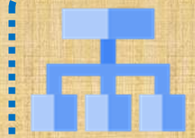
Google Compute Engine



Virtual Private Cloud

Cloud Monitoring

Cloud SQL (MySQL)

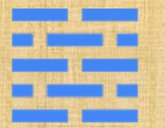


Cloud Load Balancing

Cloud Logging

Cloud IAM

Google Kubernetes Engine

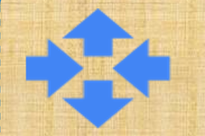


Cloud Firewalls

Cloud Domains

Cloud DNS

Cloud External IP



Cloud Router

Cloud NAT

Cloud Build

Persistent Disks

Certificate Manager

Cloud Storage

18+ GCP Services

Many more **Sub-services** under each service

Automated with Terraform

Author: Nho Luong

Skill: DevOps Engineer Lead

GitHub Repository

Repository Used For	Repository URL
Main Repository with step-by-step Docs	https://github.com/nholuongut/terraform-on-google-cloud
Terraform Google CI Repo	https://github.com/nholuongut/terraform-google-ci
GCP GKE App DevOps Repo	https://github.com/nholuongut/terraform-gcp-gke-k8s-devops
Terraform GCP GKE K8s Repo	https://github.com/stacksimplify/terraform-gcp-gke-k8s-devops
Terraform GCP GKE Infra DevOps	https://github.com/nholuongut/terraform-gcp-gke-infra-devops
Terraform GCP GKE App DevOps	git@github.com:nholuongut/terraform-gcp-gke-app-devops.git

Author: Nho Luong

Skill: DevOps Engineer Lead

Terraform Fundamentals

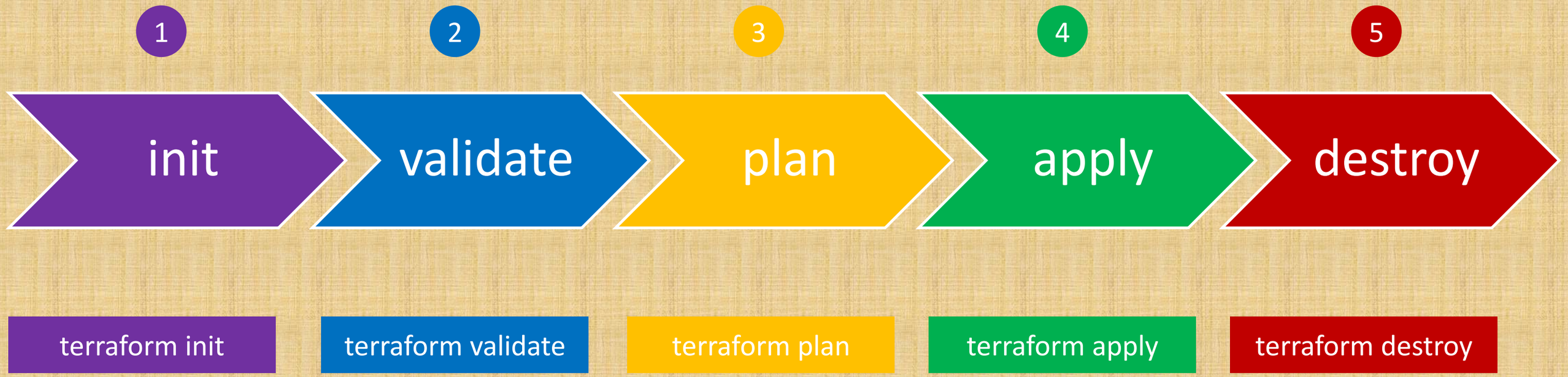
Demo-01: Install CLI Tools: gcloud CLI, Terraform CLI, VSCode Editor	7 lectures • 26min
Demo-02: Terraform Commands (init, validate, plan, apply and destroy)	3 lectures • 27min
Demo-03: Terraform Language Basics	0 lectures • 1hr 15min
Demo-04: Terraform Meta-Argument - Provider (Multiple Providers Demo)	2 lectures • 11min
Demo-05: Terraform Input Variables and Output Values	5 lectures • 38min
Demo-06: Terraform Meta-argument: count	4 lectures • 28min
Demo-07: Terraform Datasources	2 lectures • 11min
Demo-08: Terraform Meta-argument: for_each	3 lectures • 20min

Terraform Fundamentals

Author: Nho Luong

Skill: DevOps Engineer Lead

Terraform Workflow



Author: Nho Luong

Skill: DevOps Engineer Lead

Terraform language uses a limited number of **top-level block** types, which are **blocks** that can appear outside of any other **block** in a TF configuration file.

Terraform Top-Level Blocks

Most of **Terraform's features** are implemented as **top-level** blocks.

Fundamental Blocks

- Terraform Block
- Providers Block
- Resources Block

Variable Blocks

- Input Variables Block
- Output Values Block
- Local Values Block

Calling / Referencing Blocks

- Data Sources Block
- Modules Block

NEW BLOCKS – Added Recently

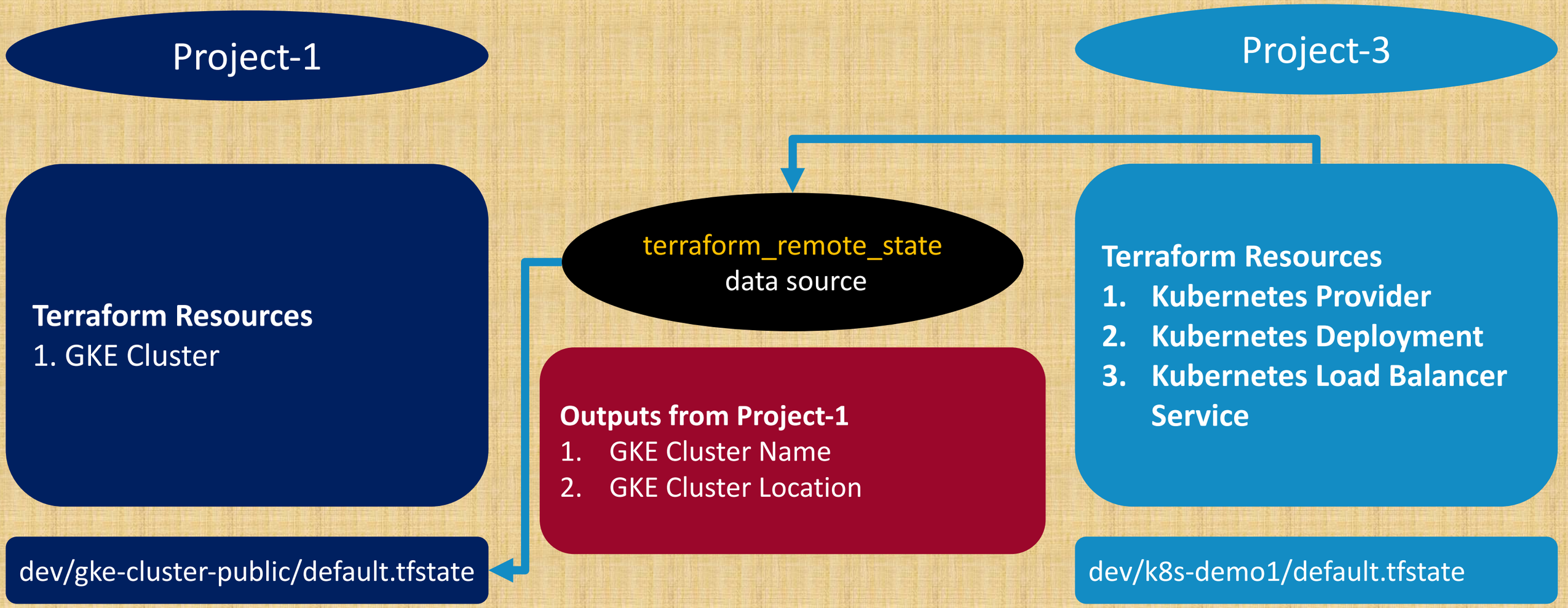
- Import Block
- Moved Block
- Removed Block
- Check Block

Author: Nho Luong

Skill: DevOps Engineer Lead

Terraform Remote State Datasource

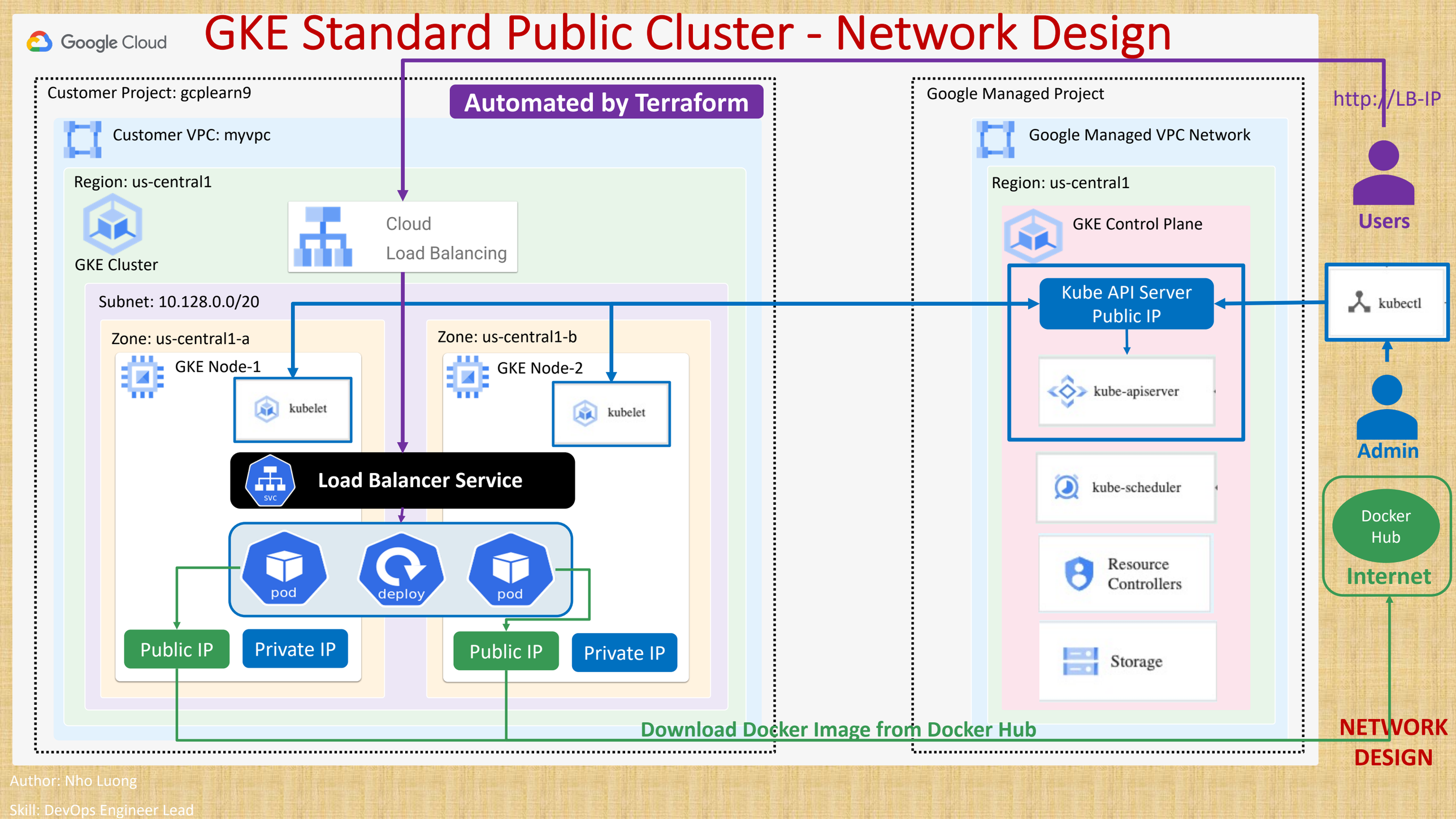
The terraform_remote_state data source retrieves the root module output values from project-1 Terraform configuration, using the latest state snapshot from the remote backend.



Author: Nho Luong

Skill: DevOps Engineer Lead

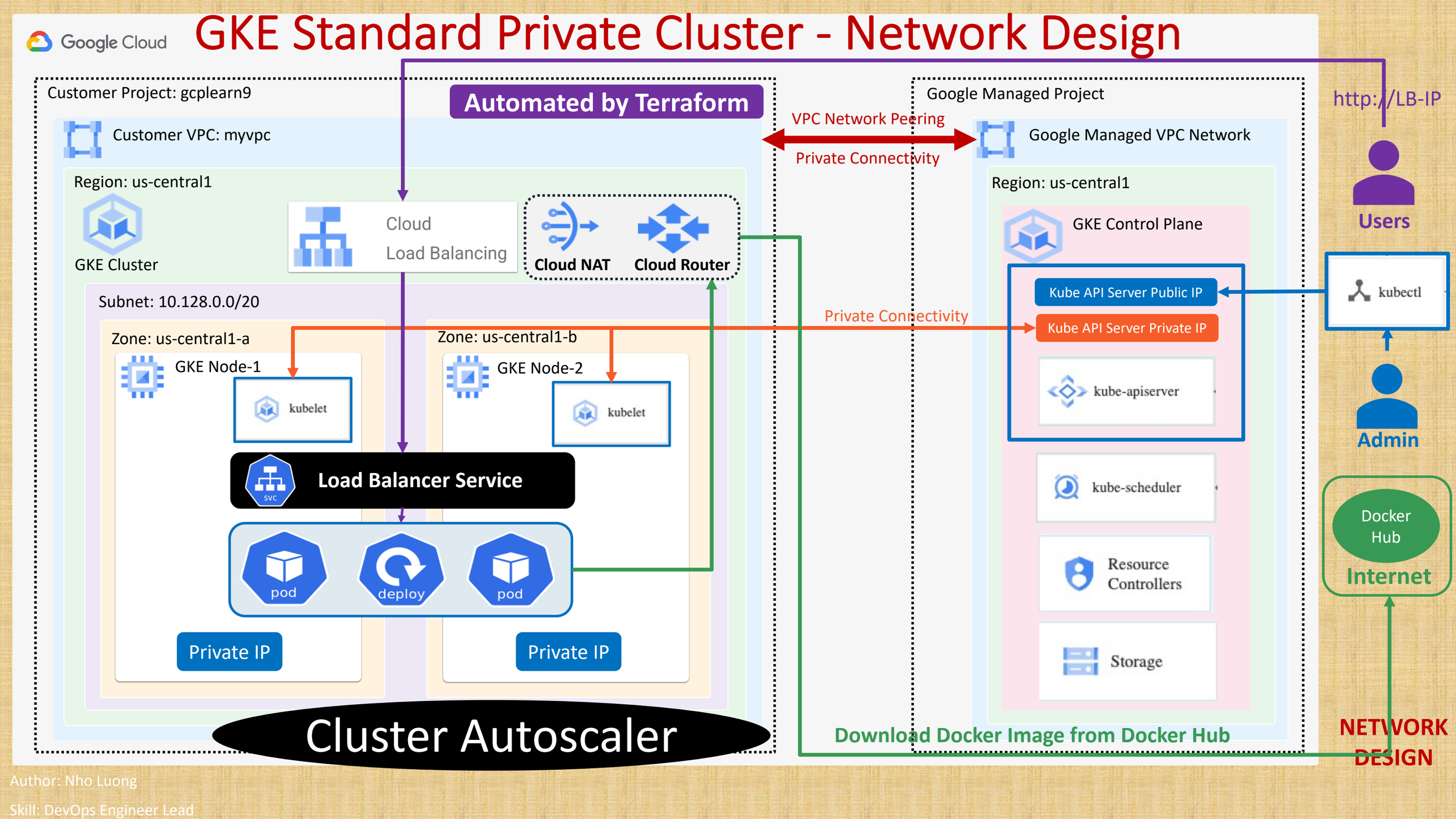
GKE Standard Public Cluster - Network Design



Author: Nho Luong

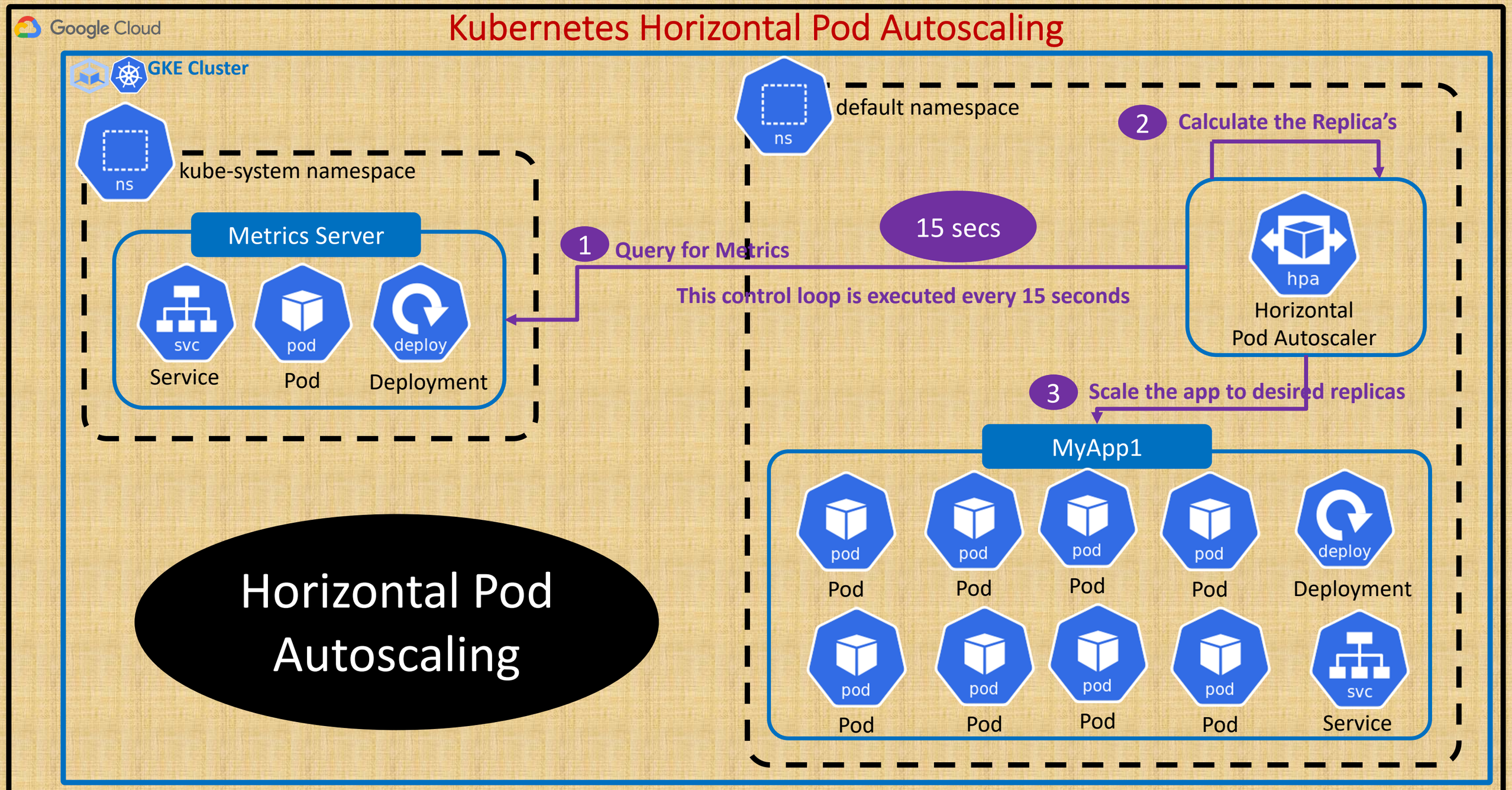
Skill: DevOps Engineer Lead

GKE Standard Private Cluster - Network Design



Author: Nho Luong

Skill: DevOps Engineer Lead

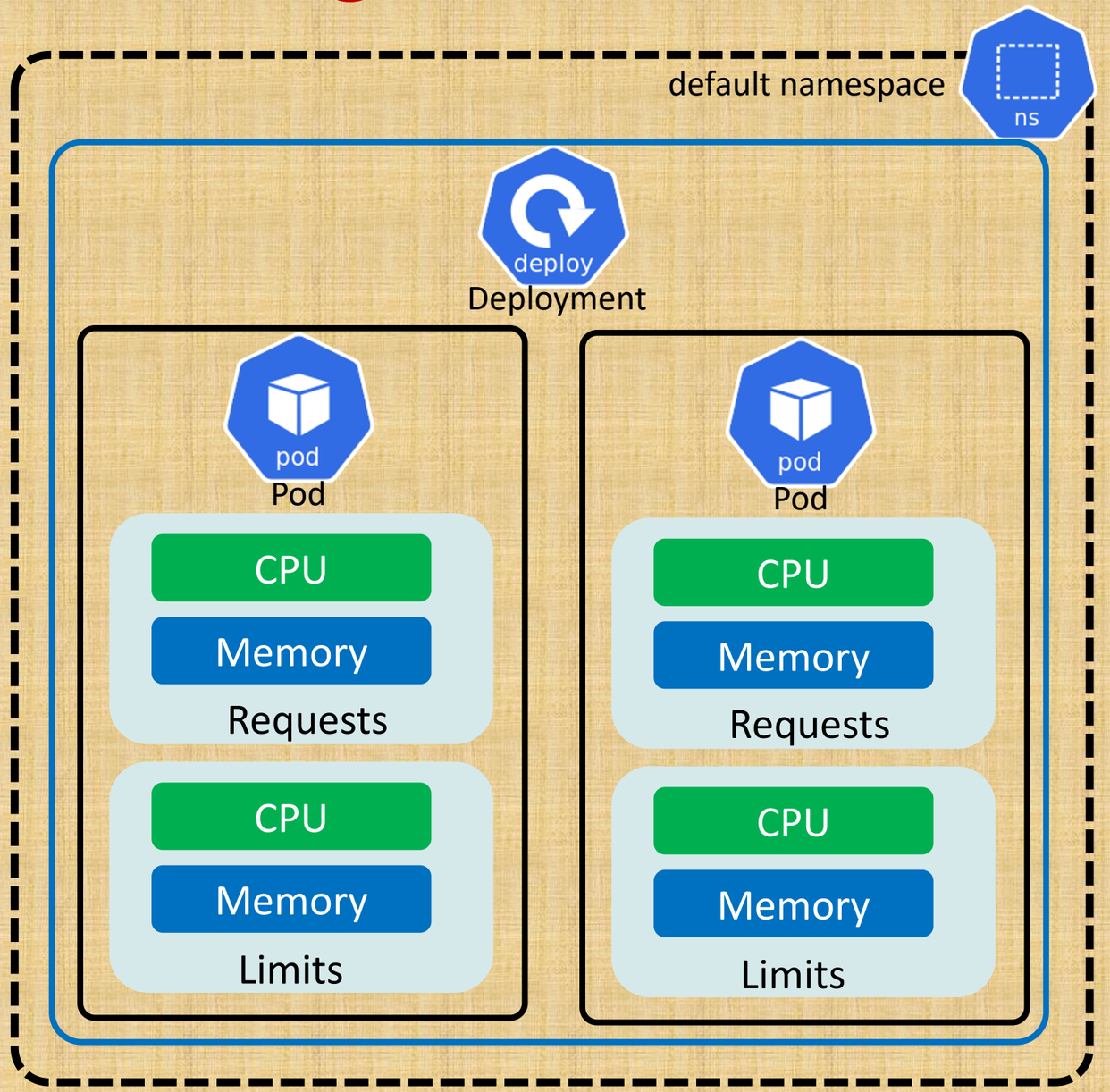


Author: Nho Luong

Skill: DevOps Engineer Lead

Kubernetes Vertical Pod Autoscaling

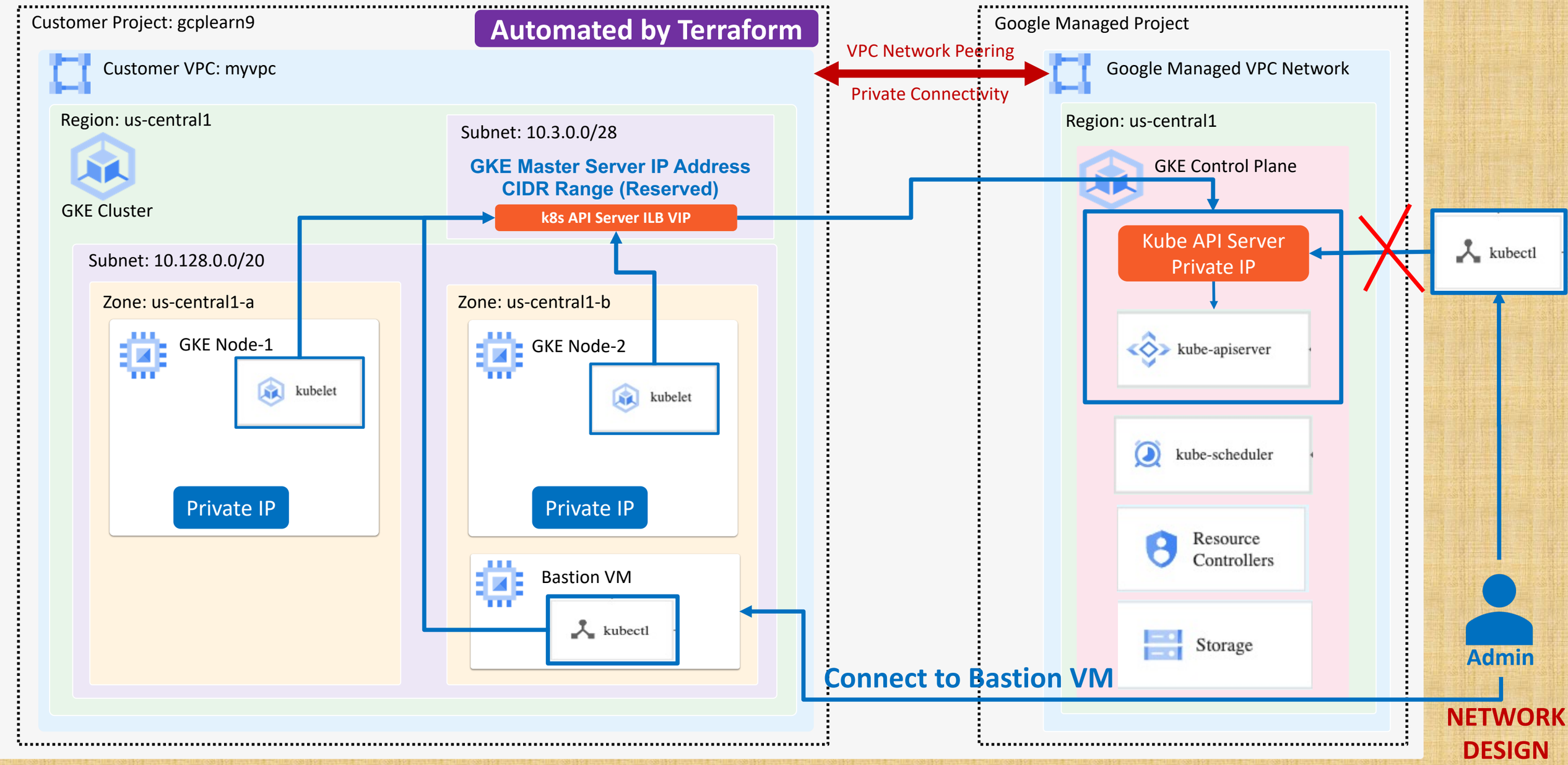
Vertical Pod Autoscaling



Author: Nho Luong

Skill: DevOps Engineer Lead

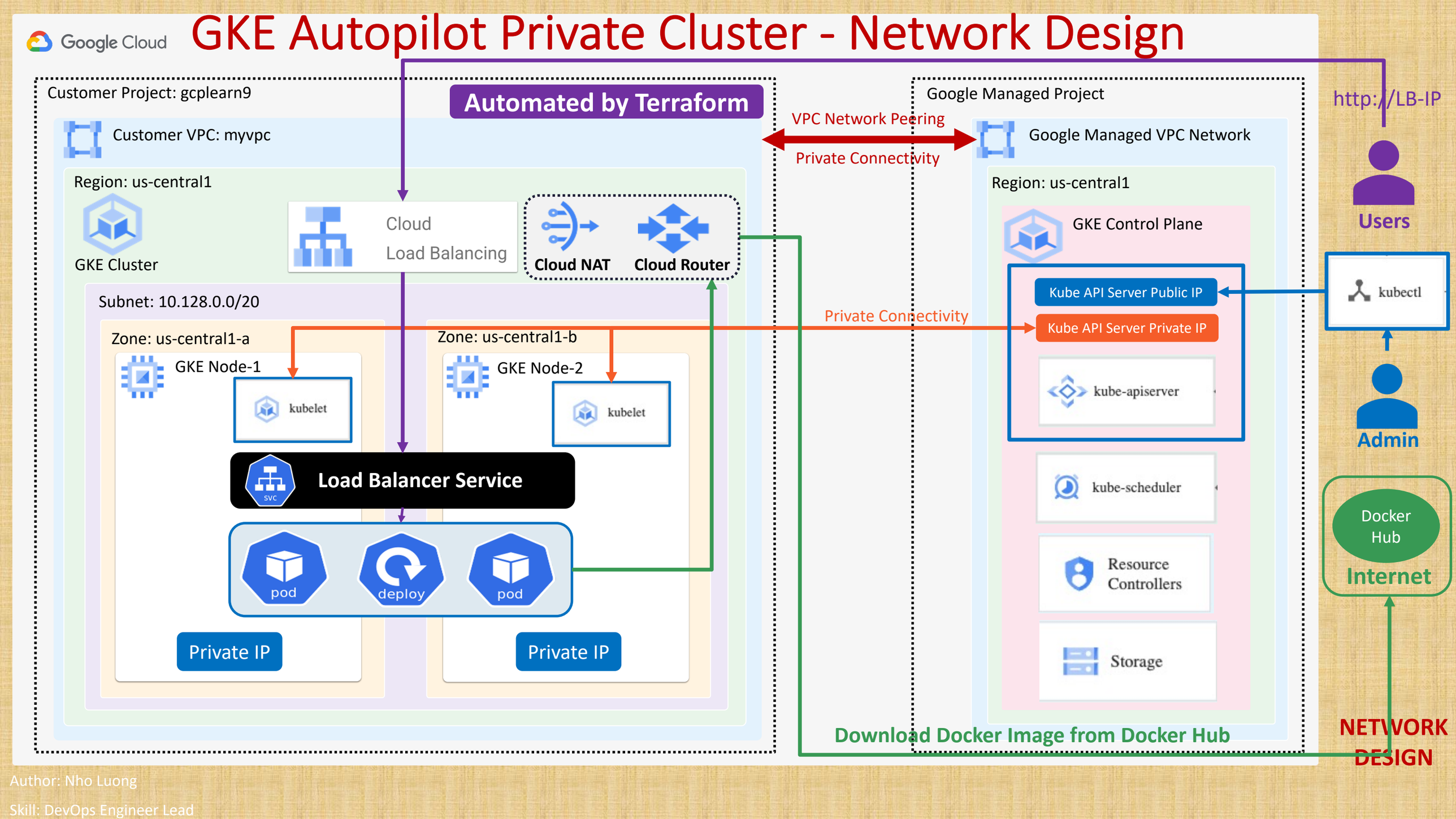
GKE Standard Private Cluster with Private Endpoint



Author: Nho Luong

Skill: DevOps Engineer Lead

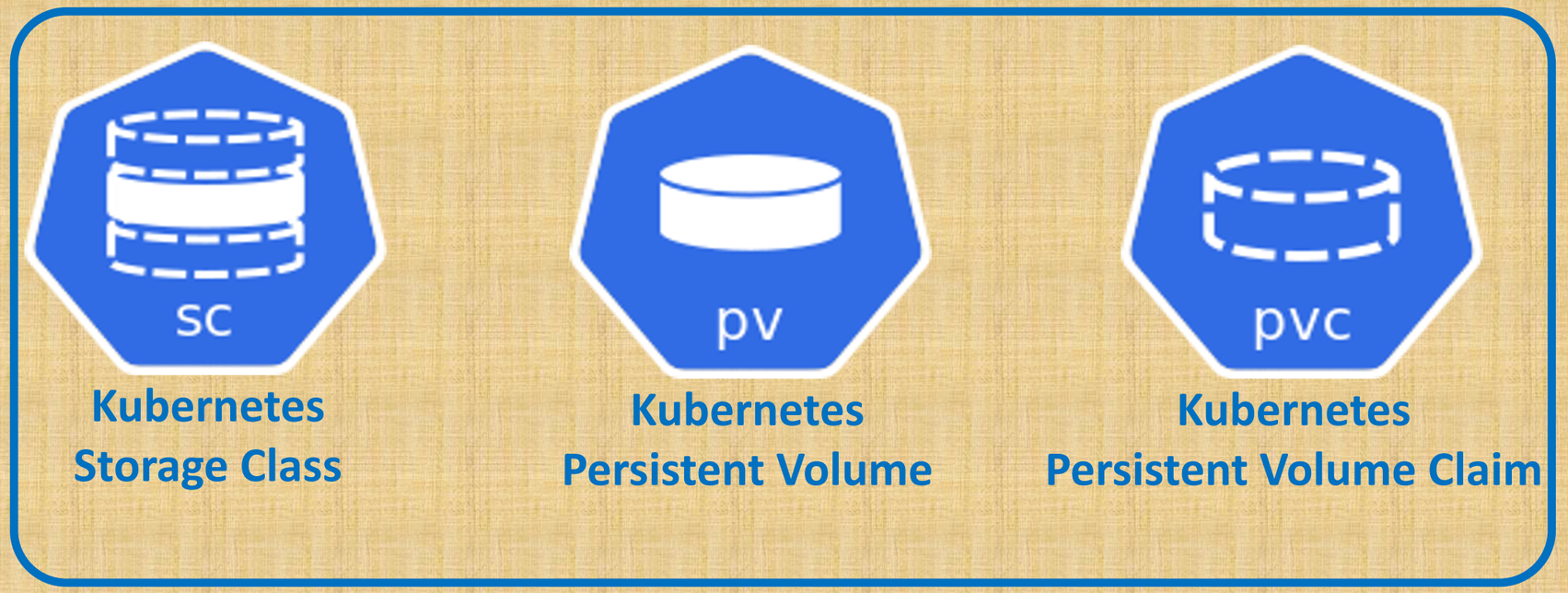
GKE Autopilot Private Cluster - Network Design

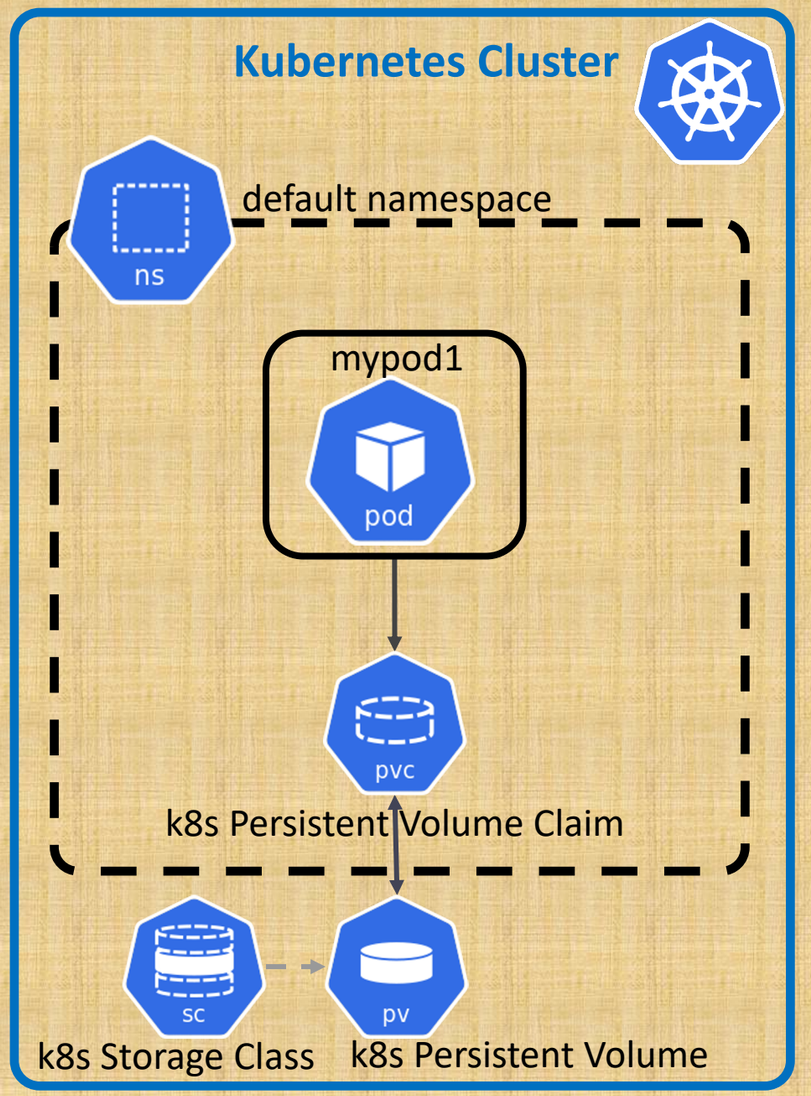


Author: Nho Luong

Skill: DevOps Engineer Lead

Kubernetes Storage

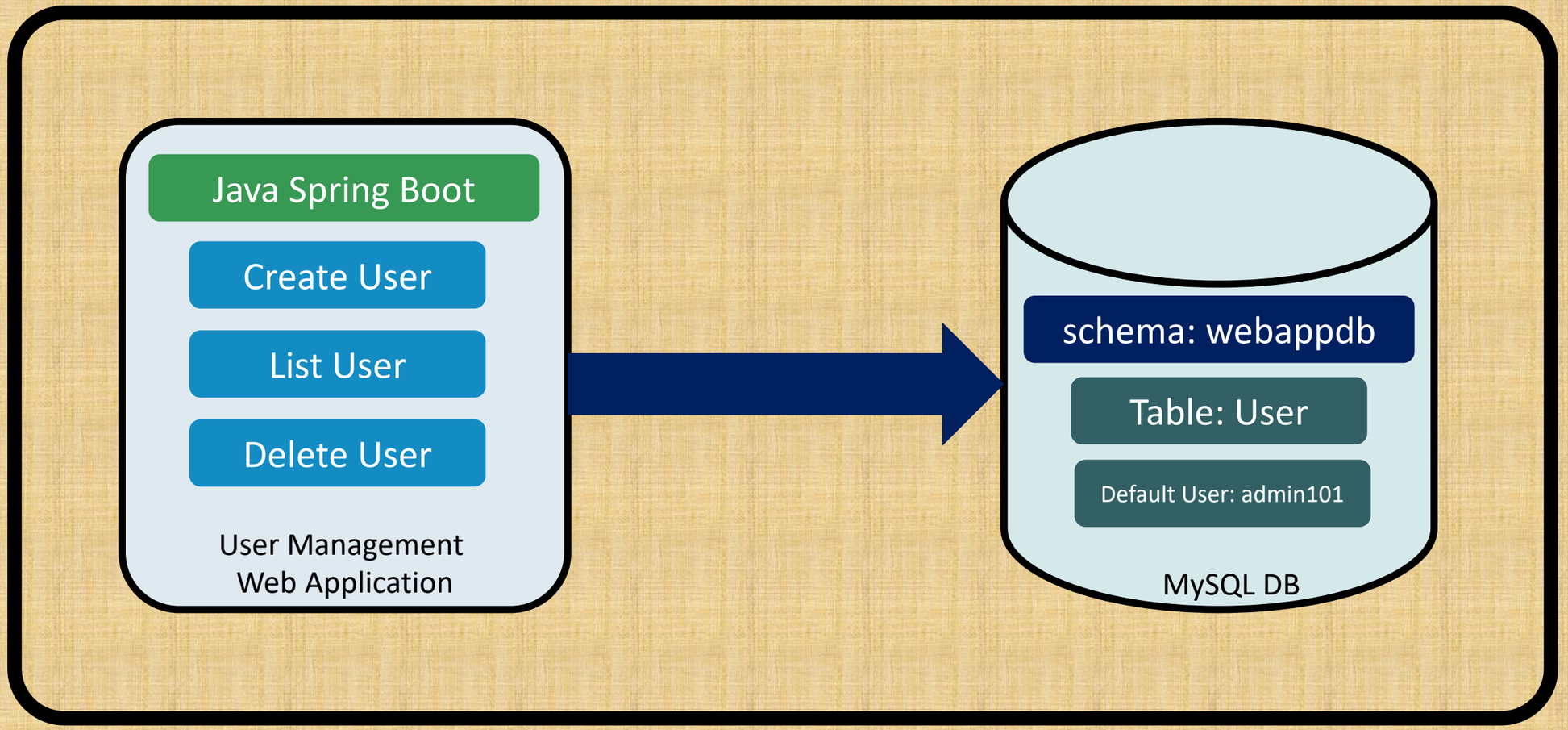




Author: Nho Luong

Skill: DevOps Engineer Lead

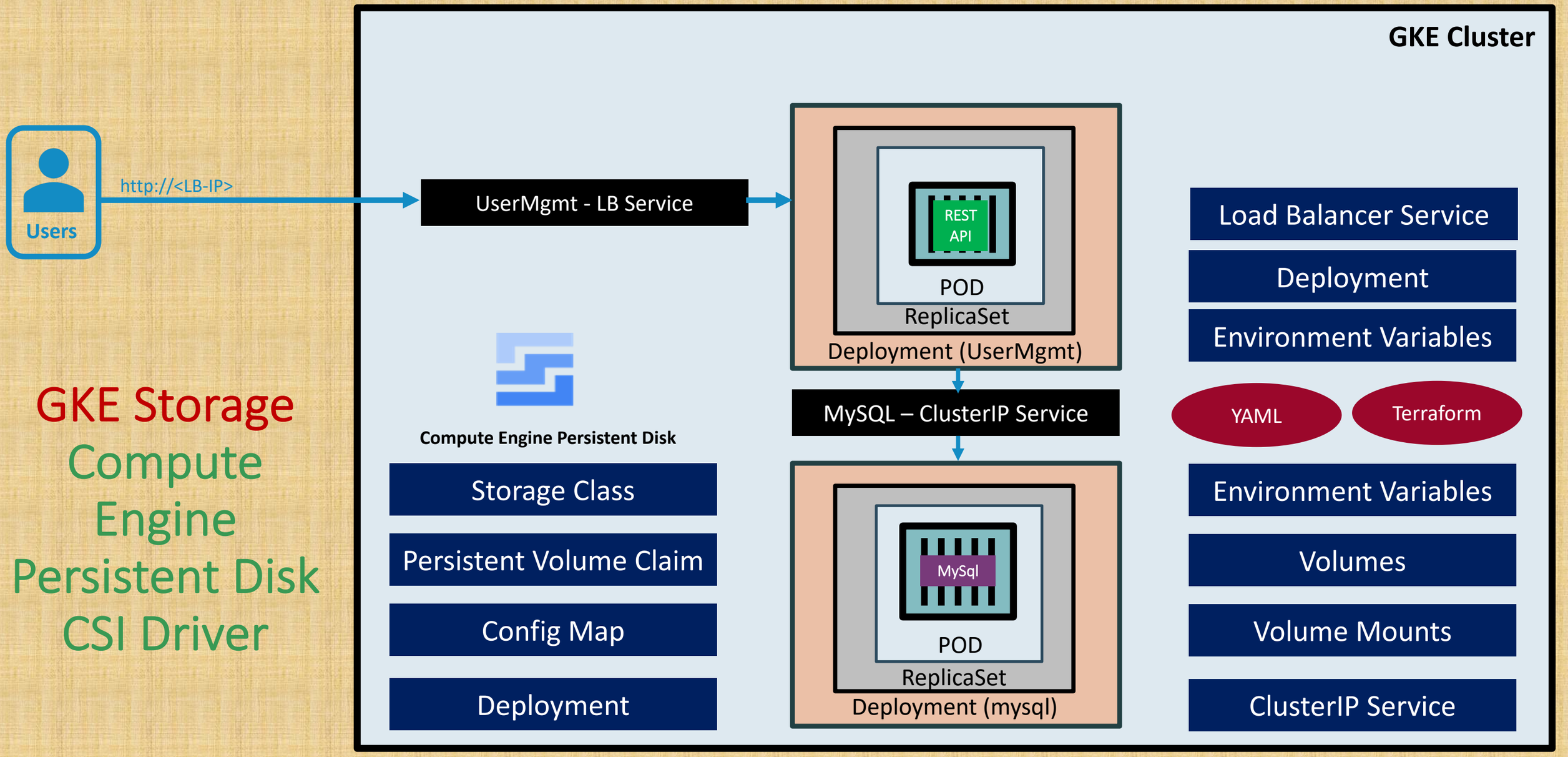
User Management Web Application - Architecture



Author: Nho Luong

Skill: DevOps Engineer Lead

GKE Storage with Compute Engine Persistence Disks

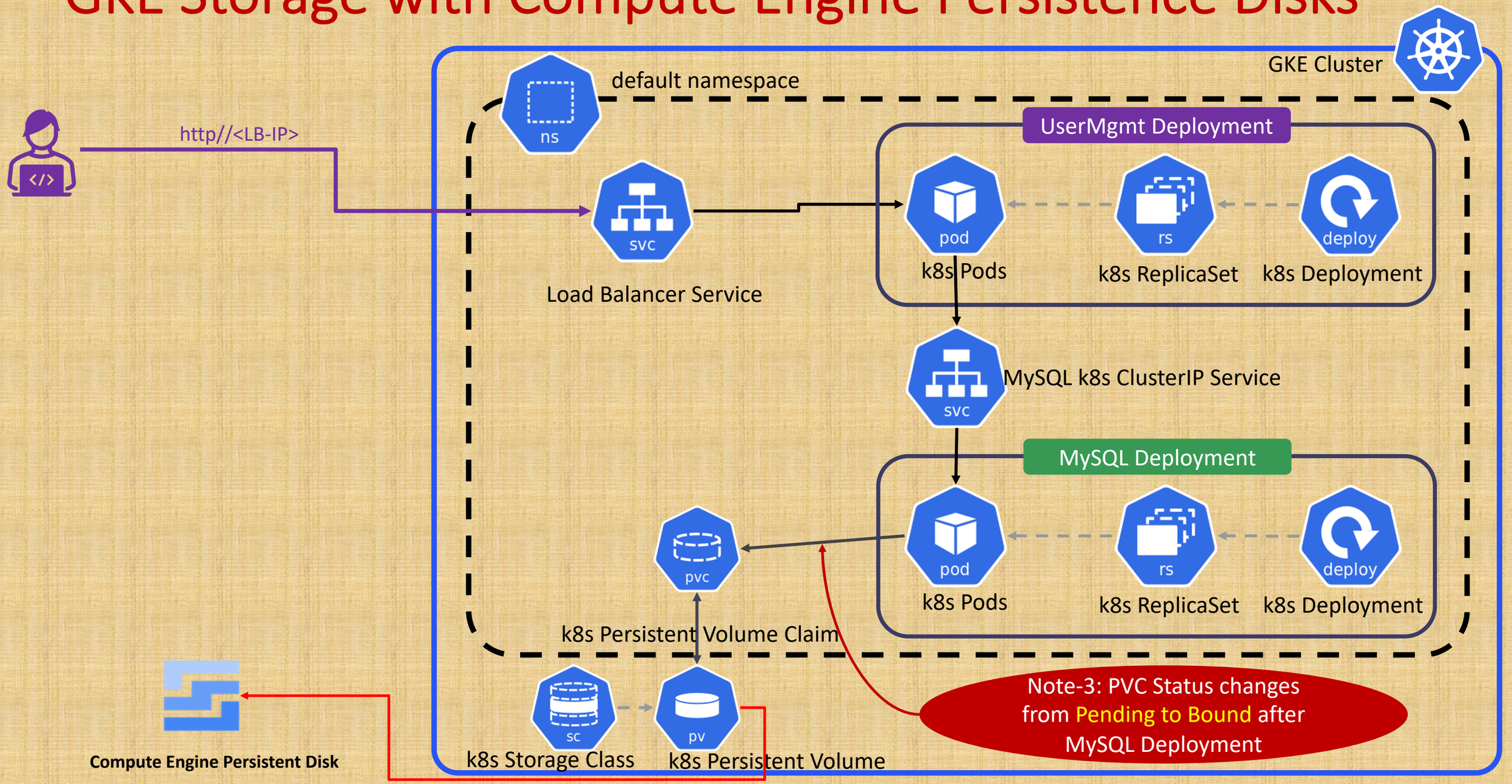


GKE Storage
Compute Engine Persistent Disk CSI Driver

Author: Nho Luong

Skill: DevOps Engineer Lead

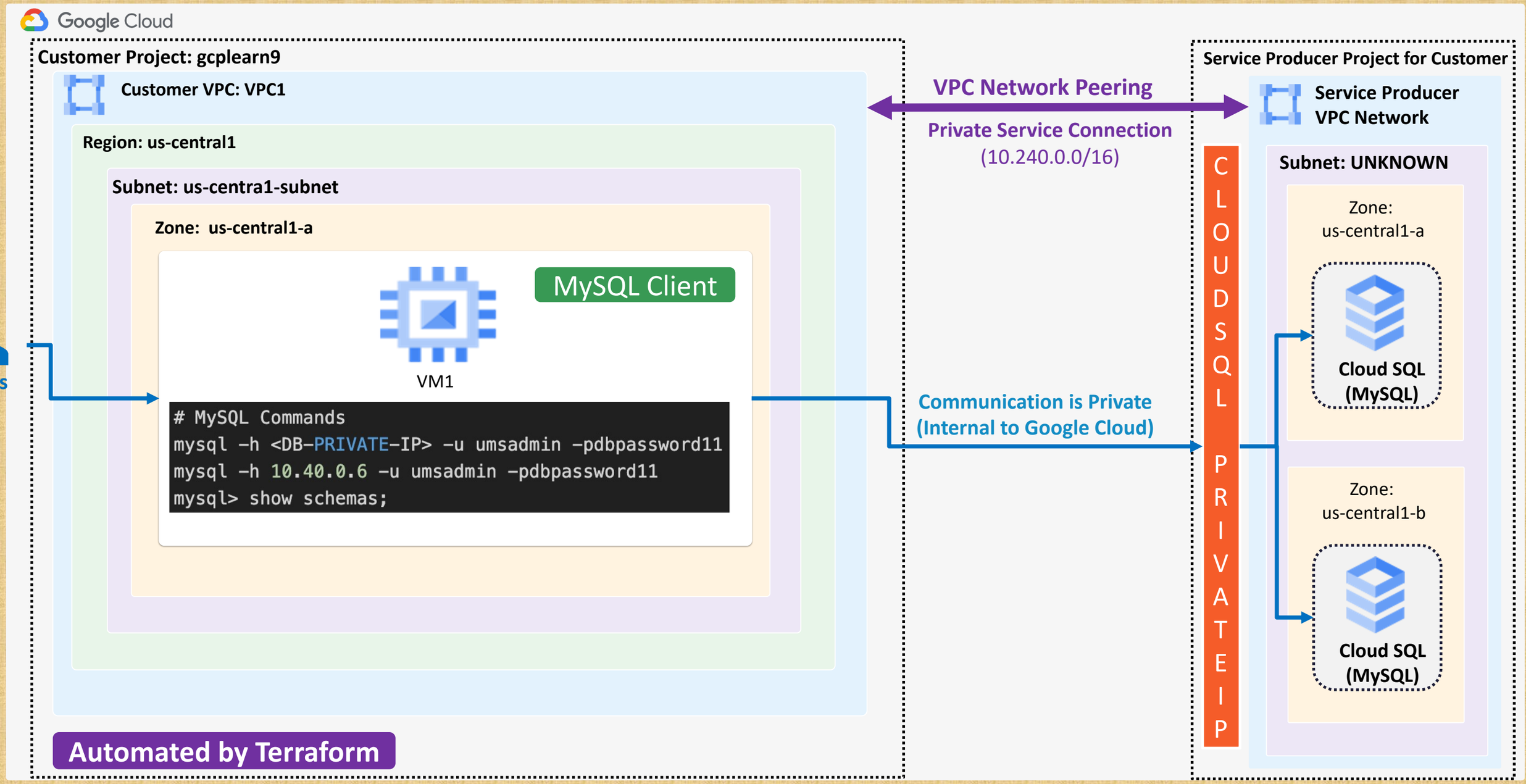
GKE Storage with Compute Engine Persistence Disks



Author: Nho Luong

Skill: DevOps Engineer Lead

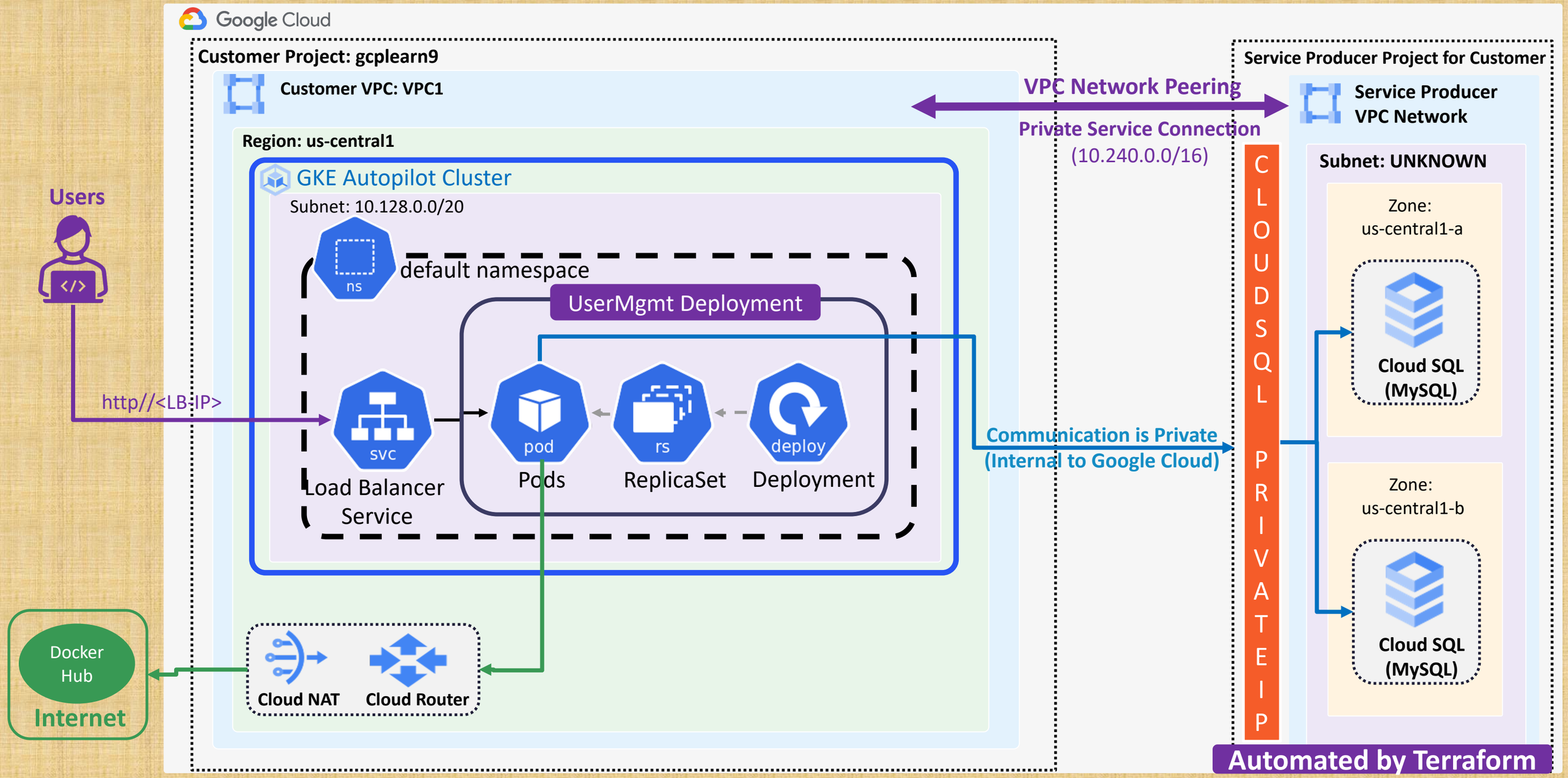
GCP Cloud SQL Private IP



Author: Nho Luong

Skill: DevOps Engineer Lead

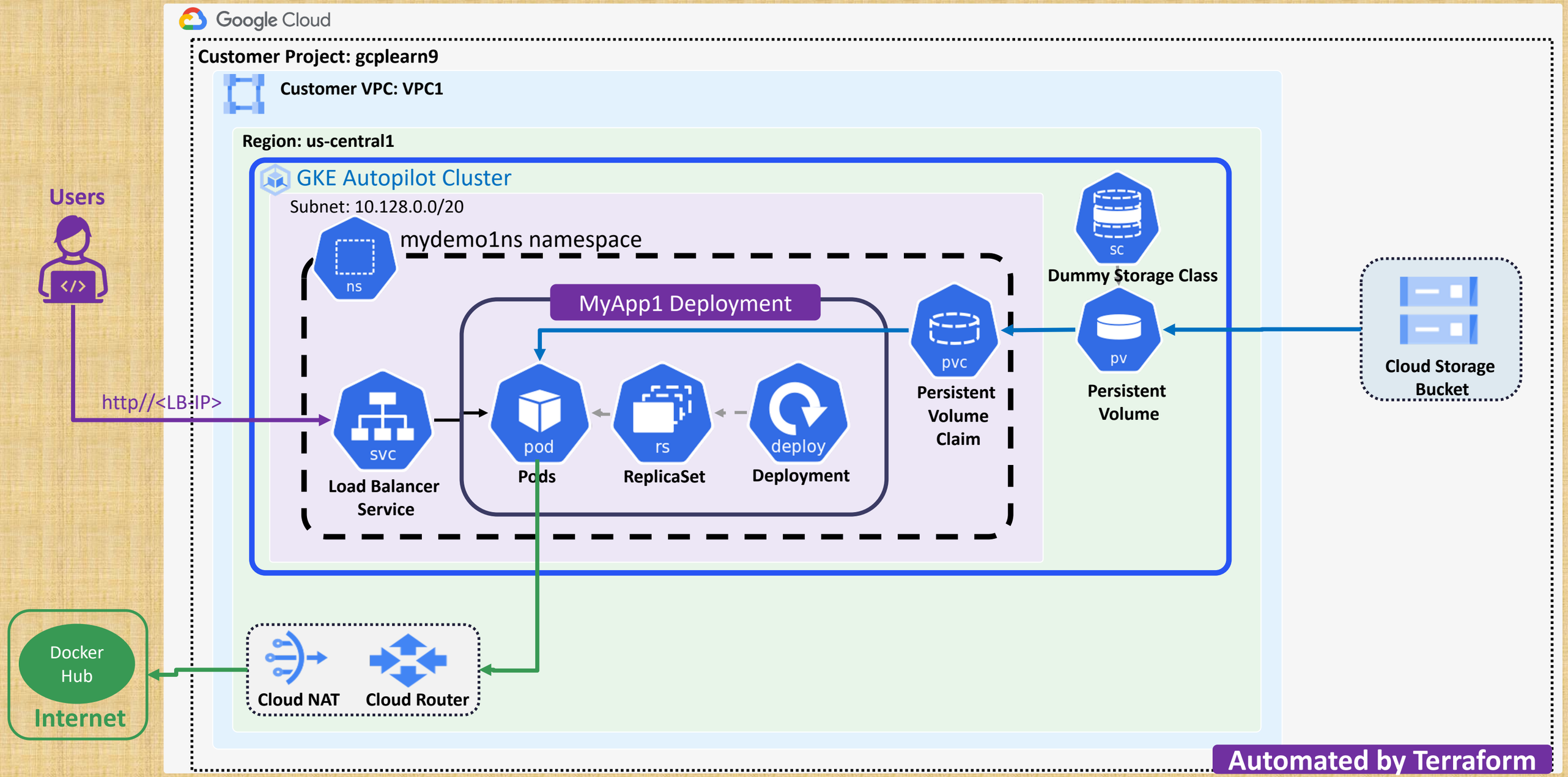
GCP Cloud SQL Private IP + GKE Autopilot Cluster + User Management Web Application



Author: Nho Luong

Skill: DevOps Engineer Lead

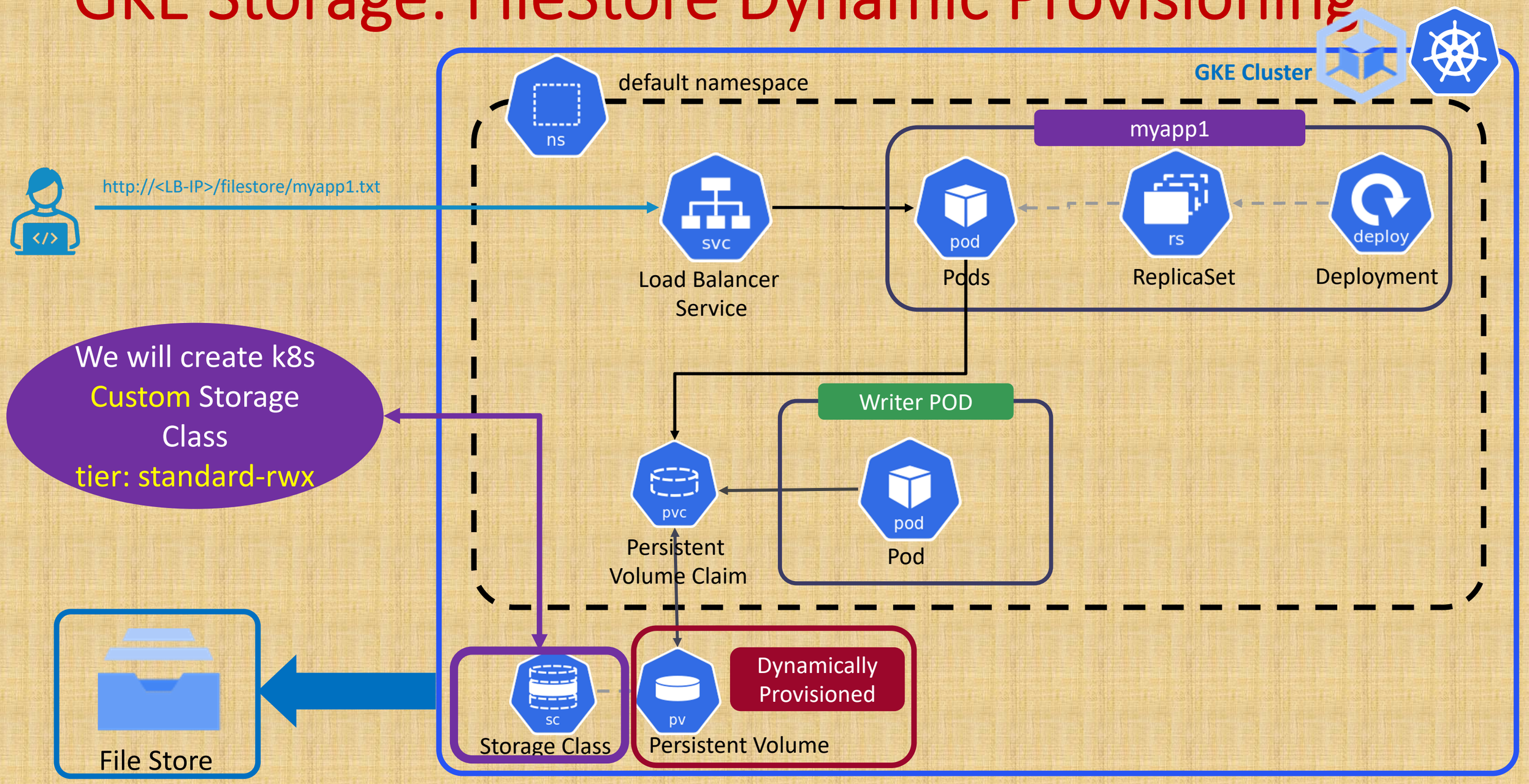
GKE Autopilot Cluster + Cloud Storage Bucket as Volume Mount



Author: Nho Luong

Skill: DevOps Engineer Lead

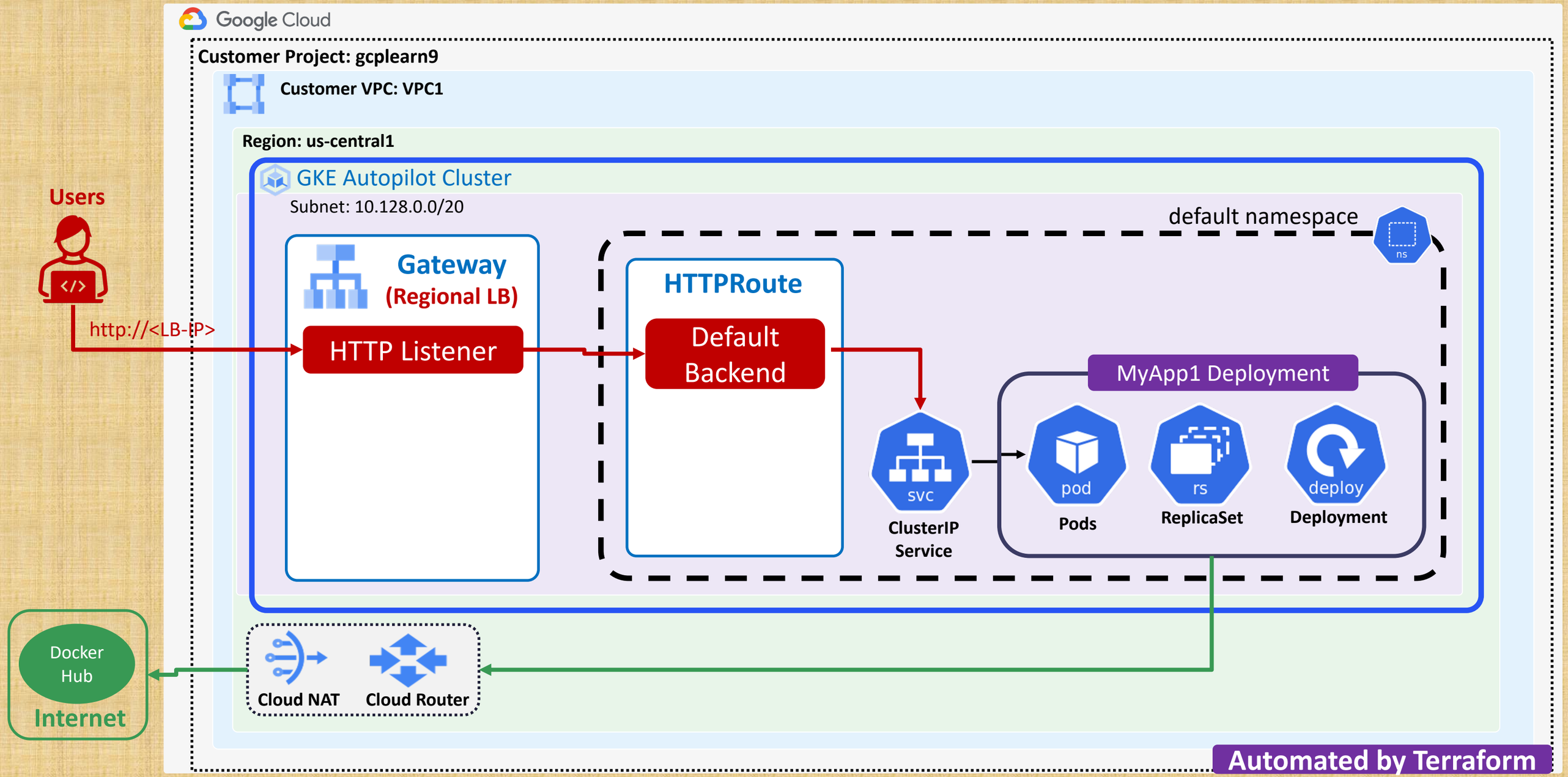
GKE Storage: FileStore Dynamic Provisioning



Author: Nho Luong

Skill: DevOps Engineer Lead

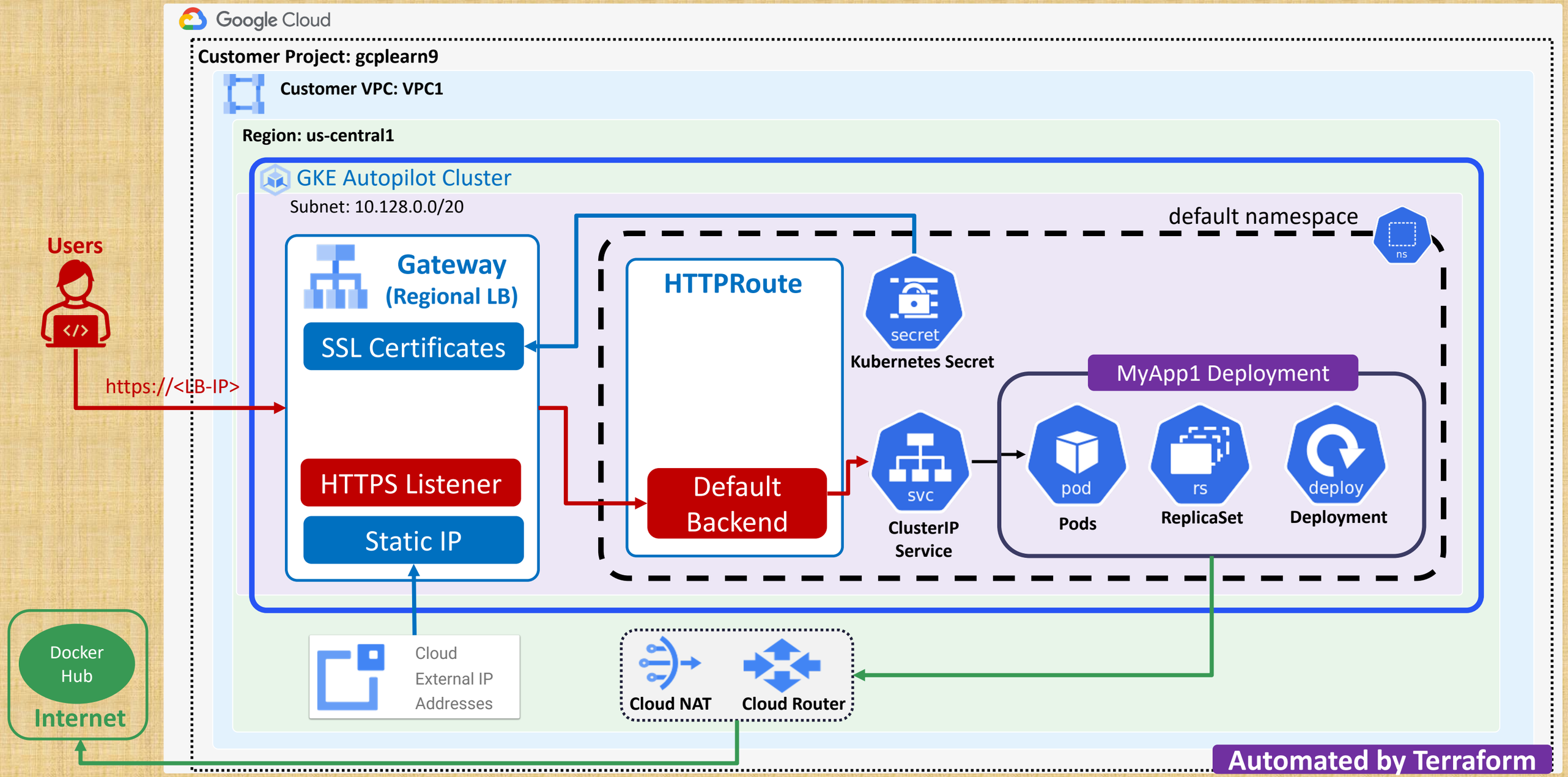
Gateway API Basics - Regional Application Load Balancer



Author: Nho Luong

Skill: DevOps Engineer Lead

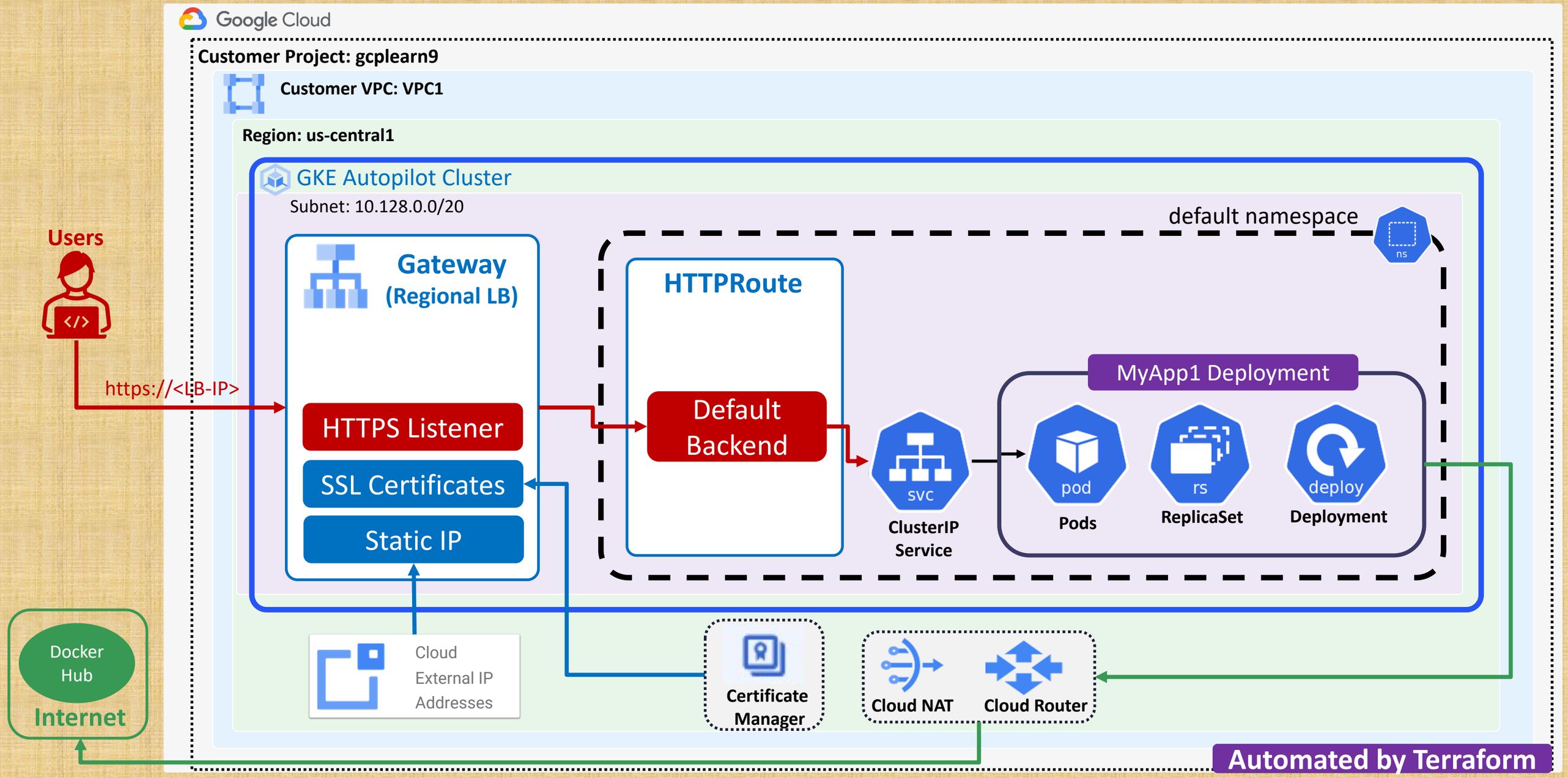
Gateway API Basics + Static IP + Self-signed SSL with Kubernetes Secret



Author: Nho Luong

Skill: DevOps Engineer Lead

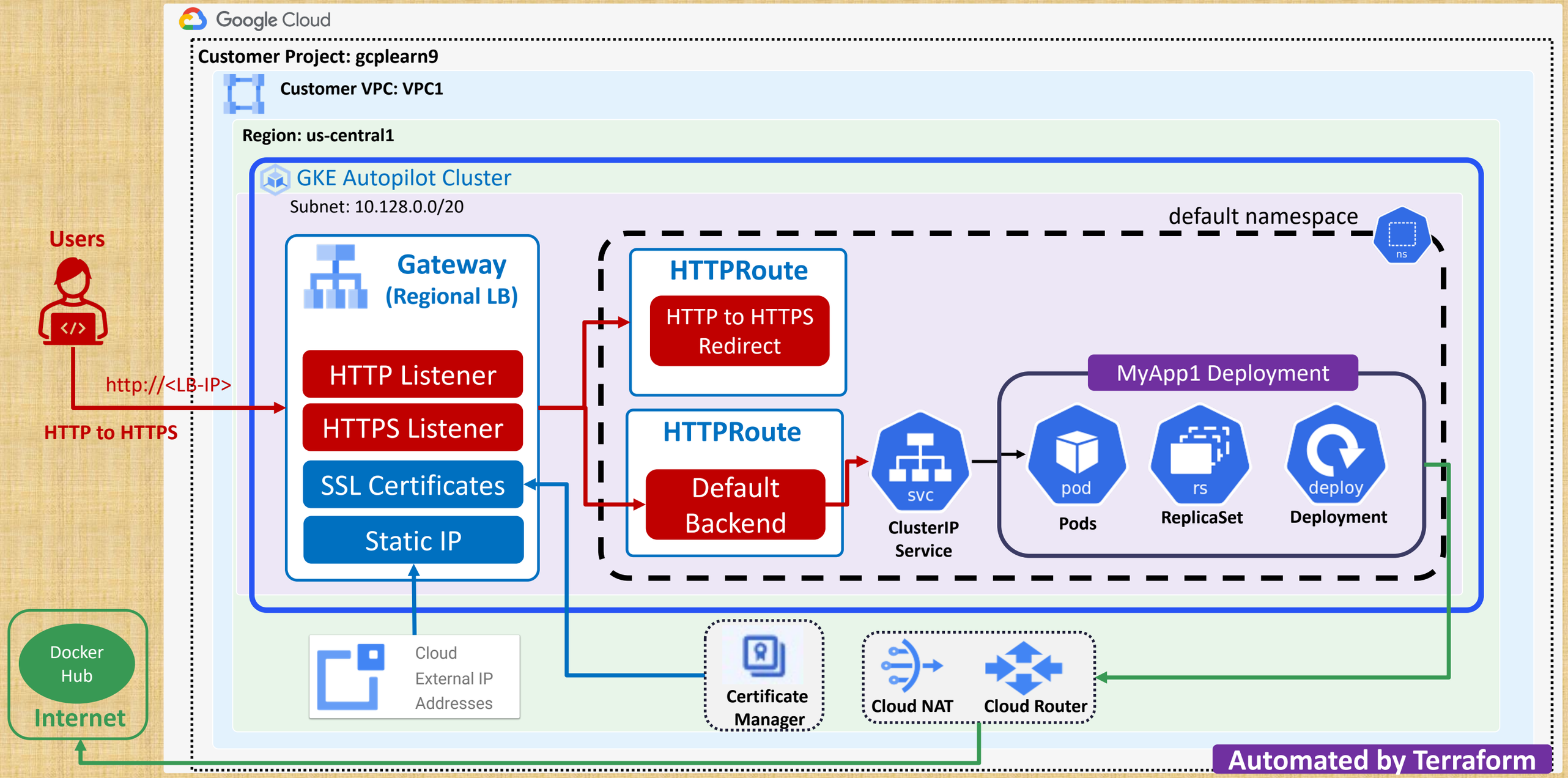
Gateway API Basics + Static IP + Self-signed SSL with Certificate Manager



Author: Nho Luong

Skill: DevOps Engineer Lead

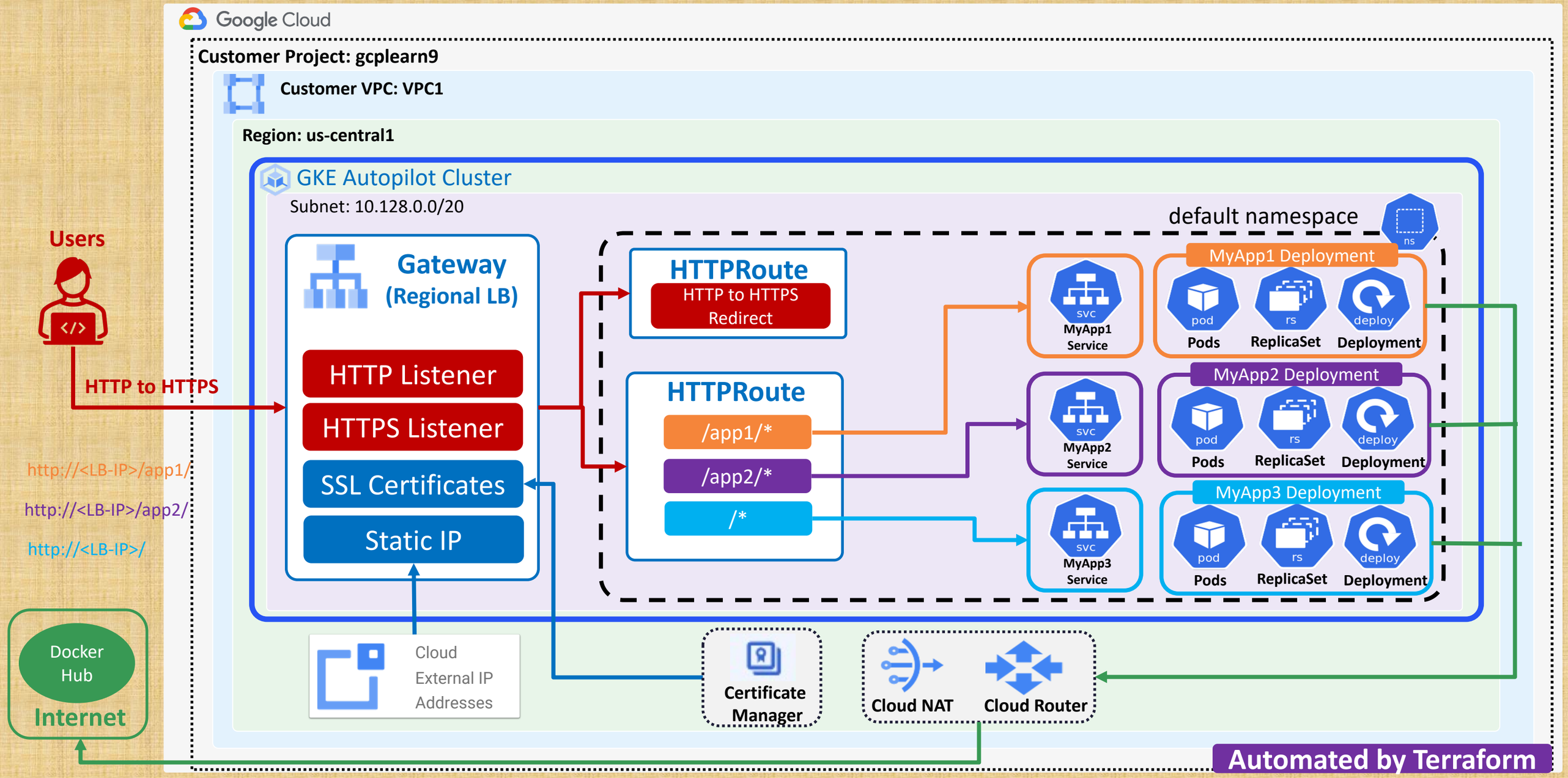
Gateway API Basics + Static IP + SSL Certificate Manager + HTTP to HTTPS



Author: Nho Luong

Skill: DevOps Engineer Lead

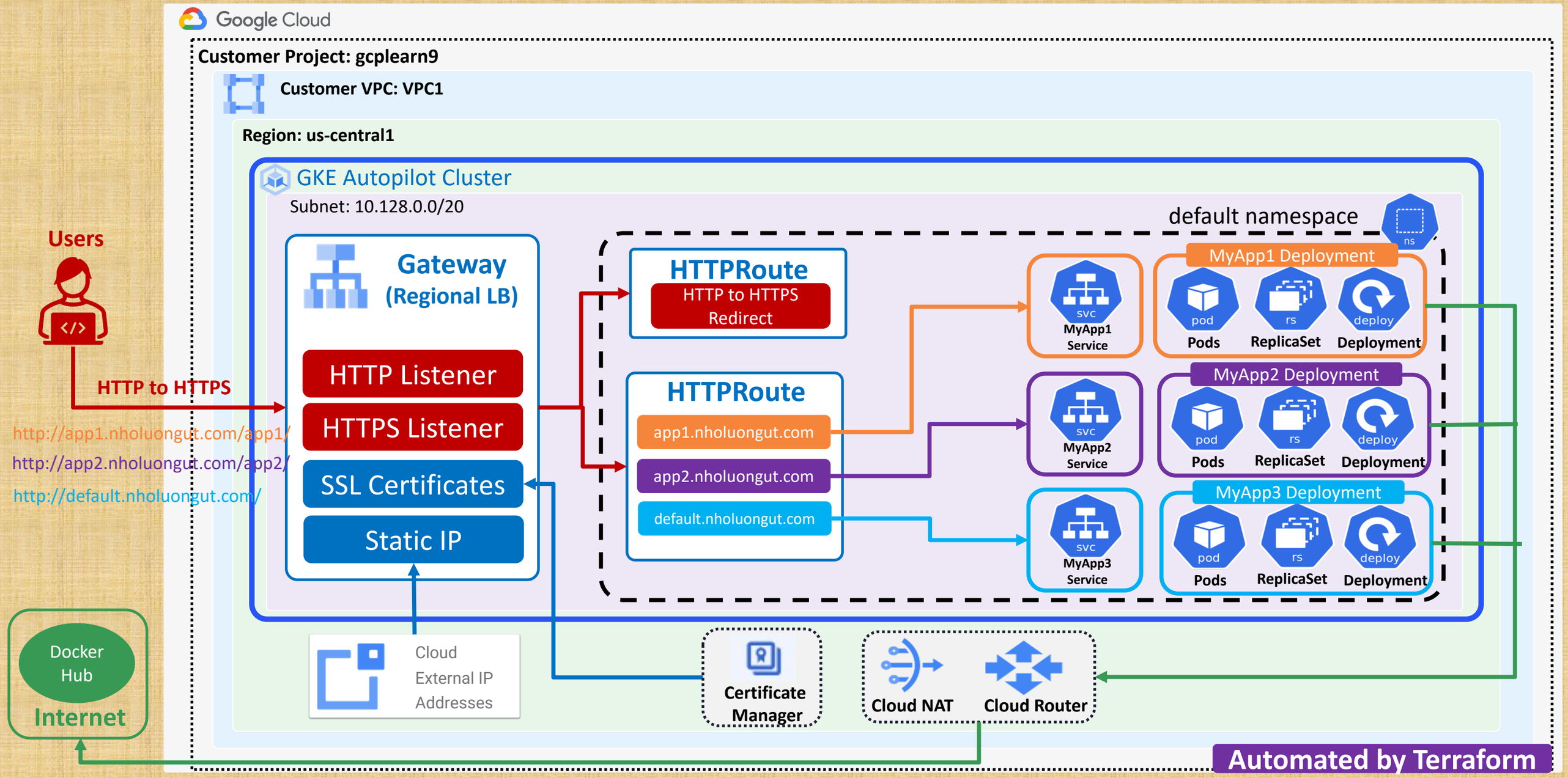
Gateway API Basics + Static IP + SSL Certificate Manager + HTTP to HTTPS + Context Path based Routing



Author: Nho Luong

Skill: DevOps Engineer Lead

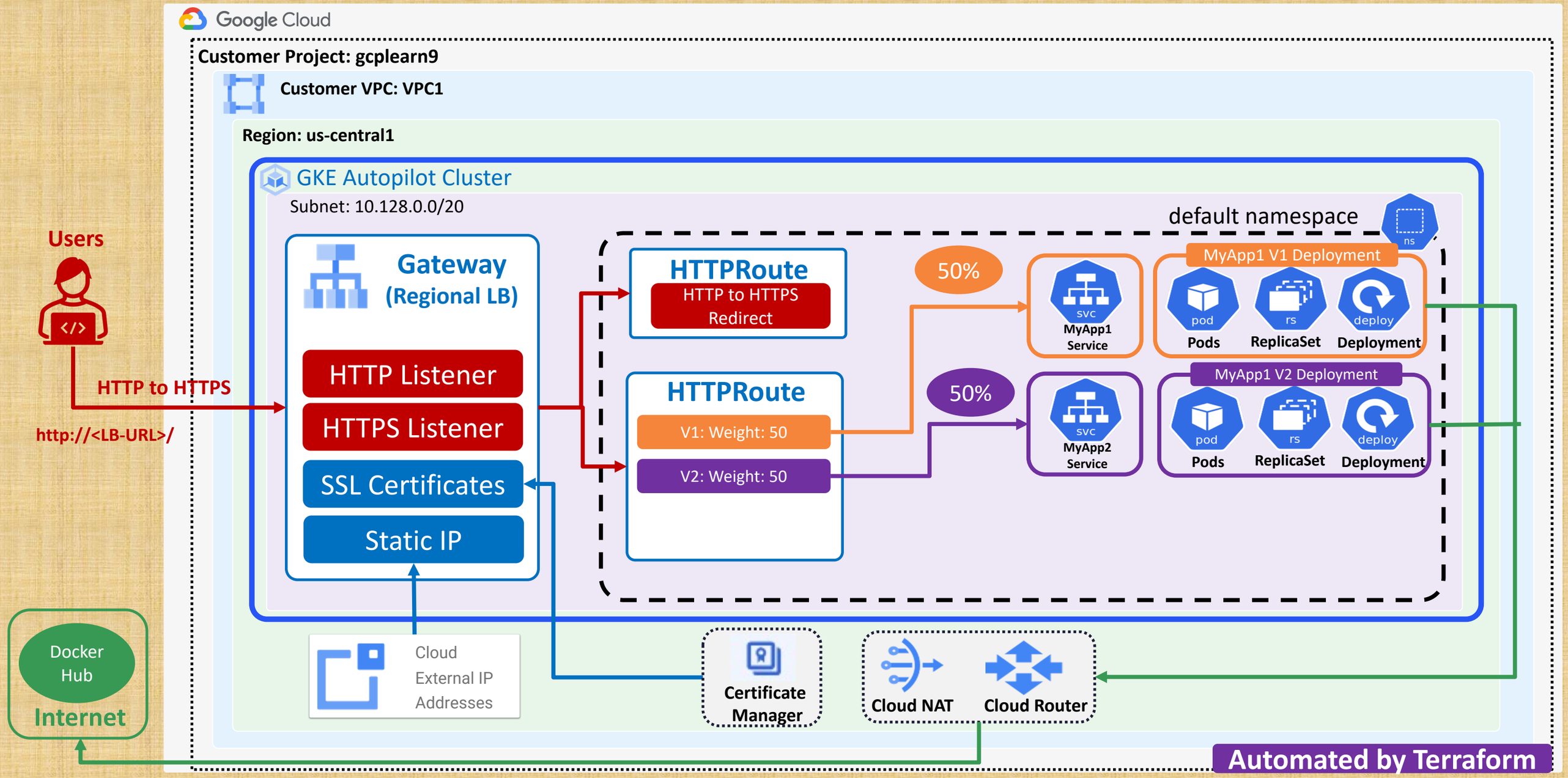
Gateway API Basics + Static IP + SSL Certificate Manager + HTTP to HTTPS + Domain name-based Routing



Author: Nho Luong

Skill: DevOps Engineer Lead

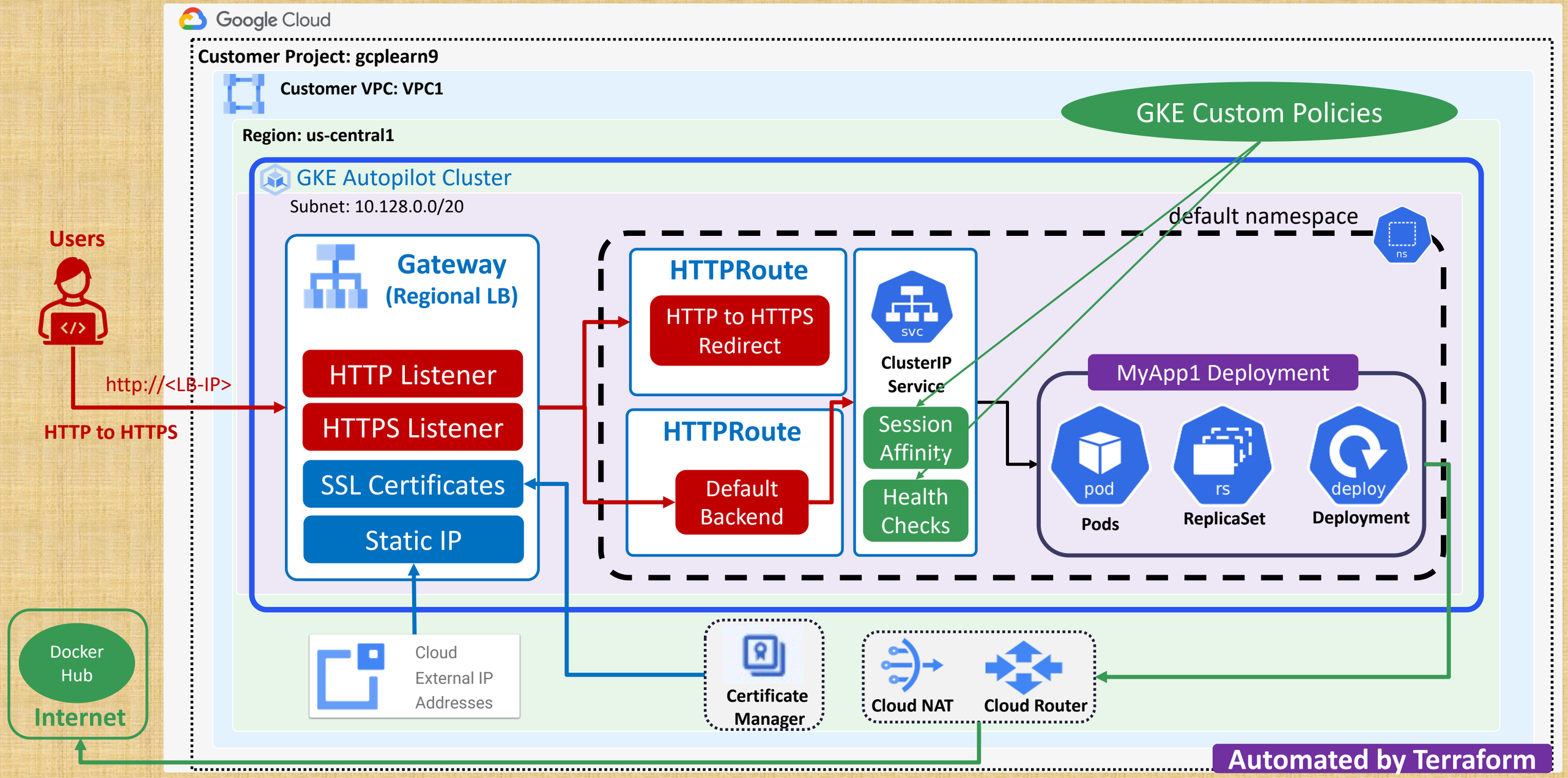
Gateway API Basics + Static IP + SSL Certificate Manager + HTTP to HTTPS + Traffic Splitting



Author: Nho Luong

Skill: DevOps Engineer Lead

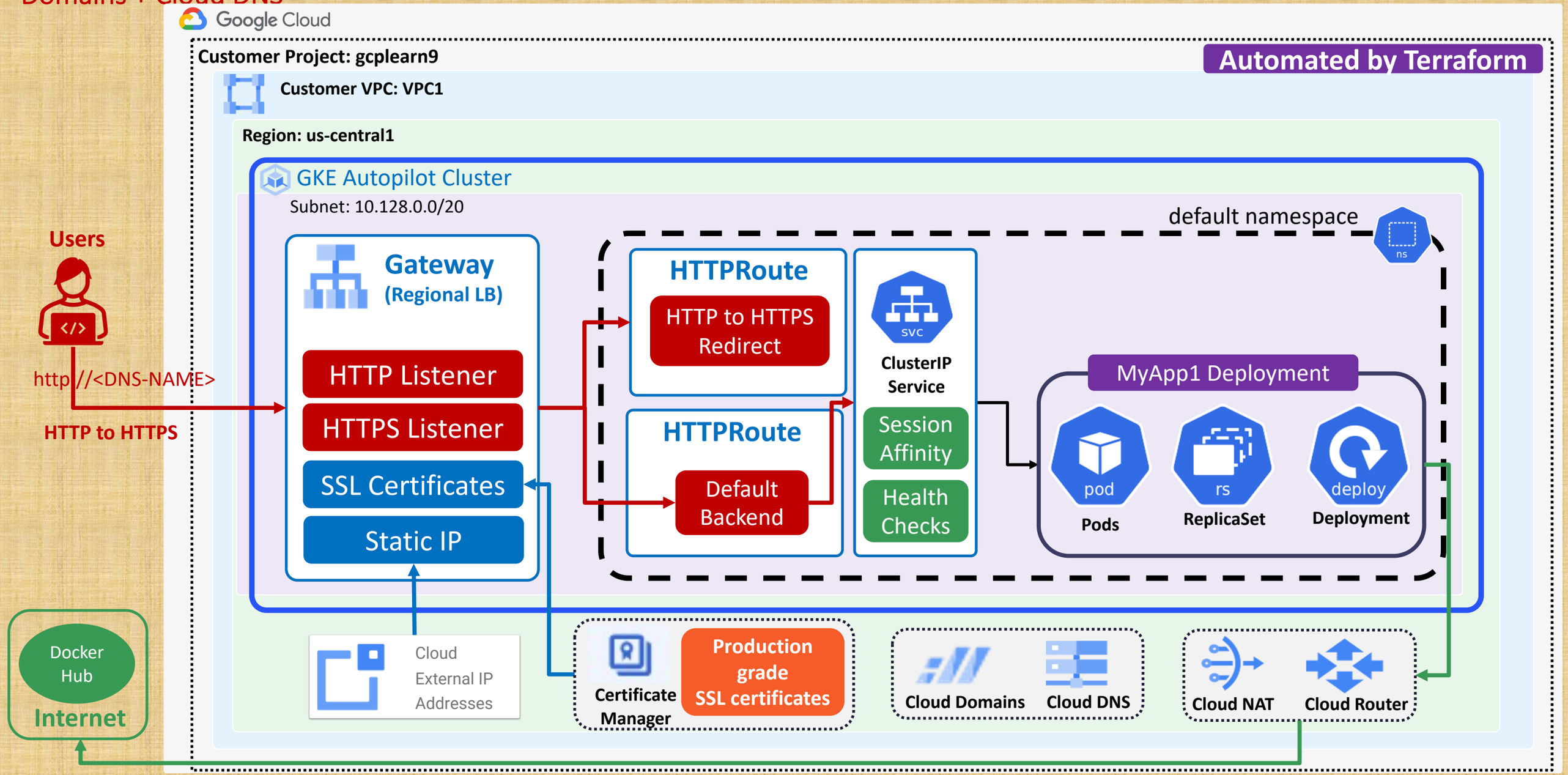
Gateway API + Static IP + SSL Certificate Manager + HTTP to HTTPS + Health Checks + Session Affinity



Author: Nho Luong

Skill: DevOps Engineer Lead

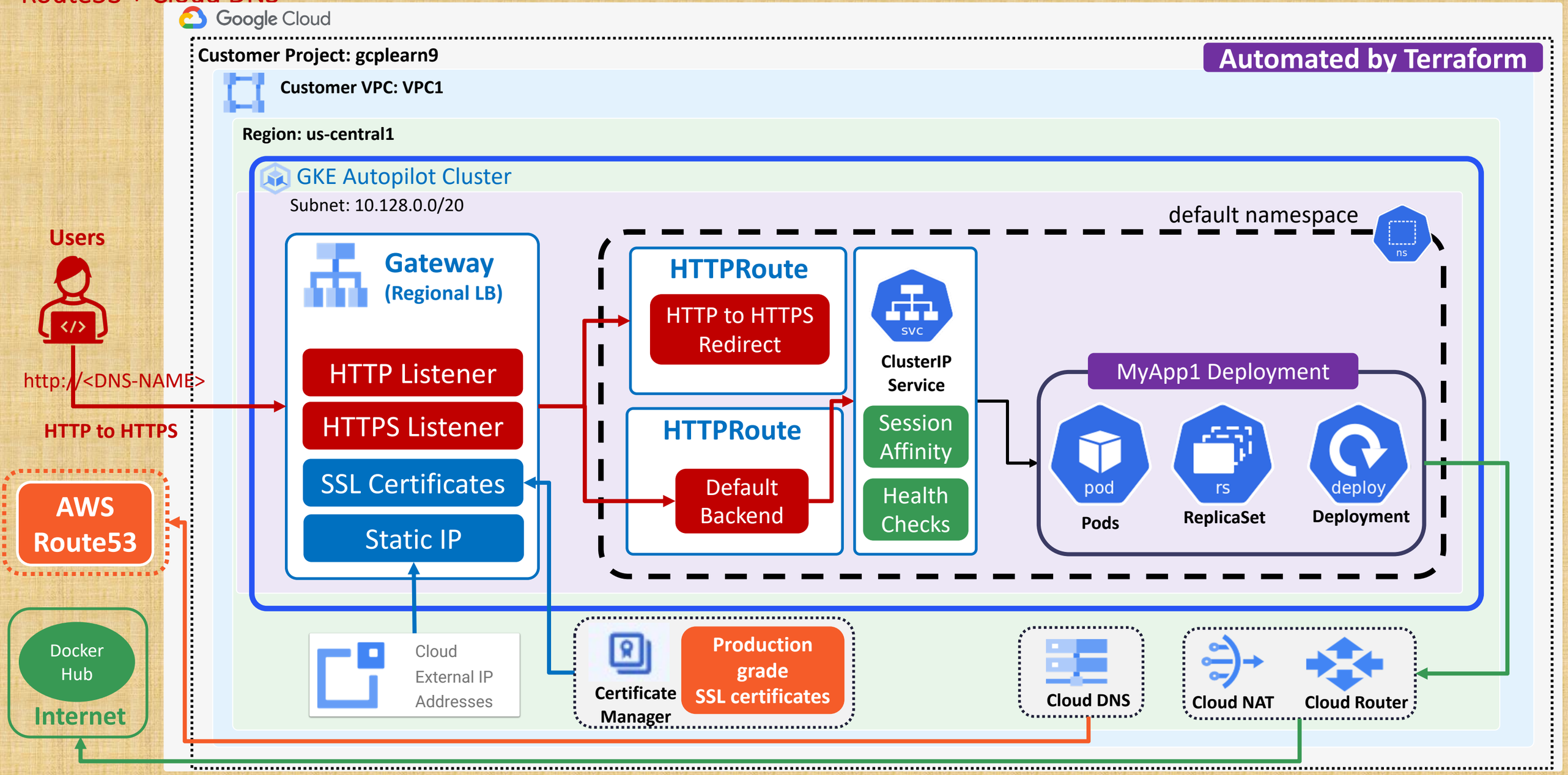
Gateway API + Static IP + SSL Certificate Manager + HTTP to HTTPS + Health Checks + Session Affinity + Cloud Domains + Cloud DNS



Author: Nho Luong

Skill: DevOps Engineer Lead

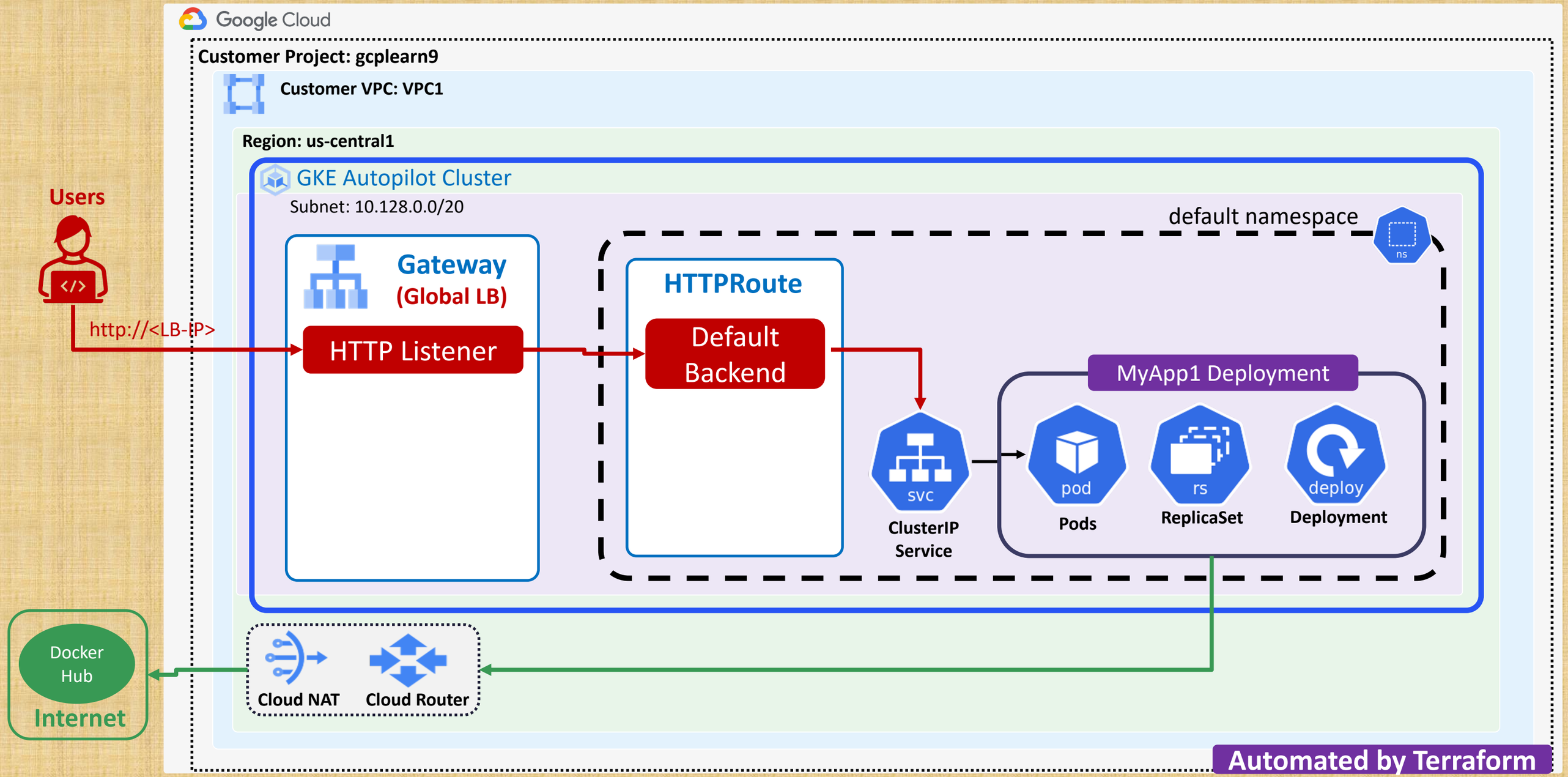
Gateway API + Static IP + SSL Certificate Manager + HTTP to HTTPS + Health Checks + Session Affinity + AWS Route53 + Cloud DNS



Author: Nho Luong

Skill: DevOps Engineer Lead

Gateway API – Global Application Load Balancer



Author: Nho Luong

Skill: DevOps Engineer Lead

Terraform Registry - Use Publicly Available Modules

The Terraform Registry hosts a broad collection of publicly available Terraform modules for configuring many kinds of common infrastructure.

Modules Demo 1

These modules are free to use, and Terraform can download them automatically if you specify the appropriate source and version in a module call block.

```
# Module: VPC
module "vpc" {
    source  = "terraform-google-modules/network/google"
    version = "~> 9.1"
    project_id   = var.gcp_project
    network_name = "${local.name}-vpc"
    routing_mode = "GLOBAL"
    subnets = [
        {
            subnet_name           = "${local.name}-${var.gcp_region1}-subnet"
            subnet_ip             = "10.128.0.0/20"
            subnet_region         = var.gcp_region1
        }
    ]
}
```

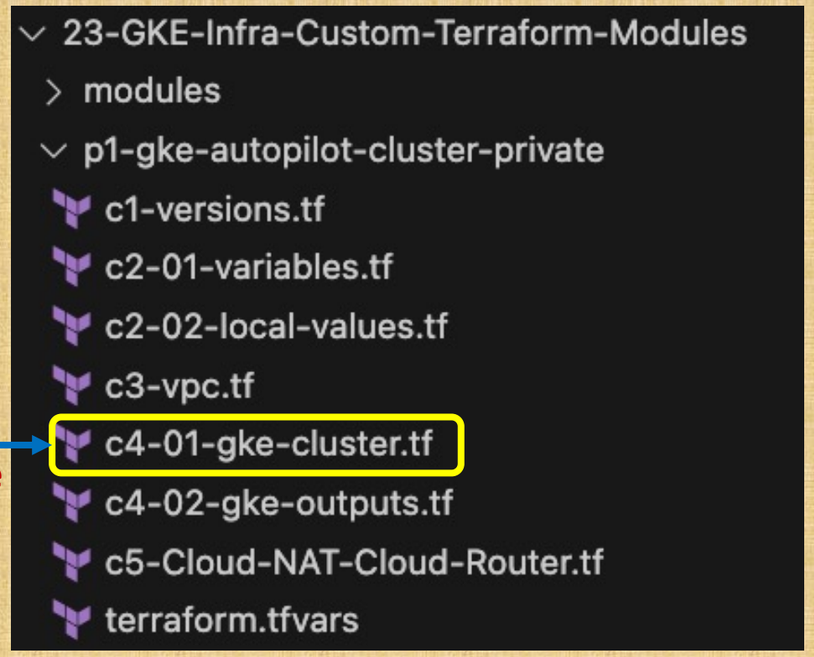
Author: Nho Luong

Skill: DevOps Engineer Lead

Build a Local Terraform Module

GKE
Custom
Terraform Module

Root Module



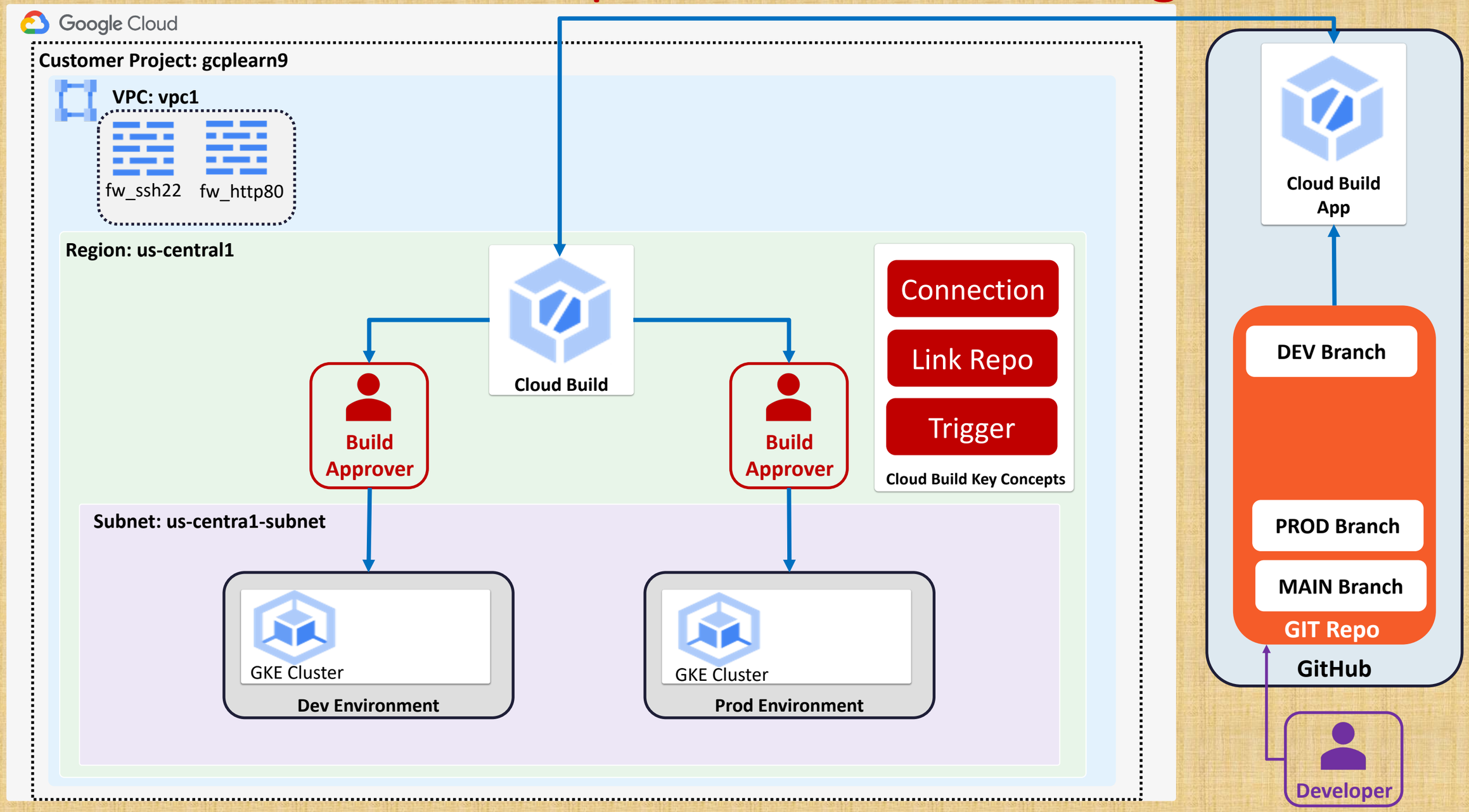
Call GKE Cluster child module in Root Module

Child Module

Author: Nho Luong

Skill: DevOps Engineer Lead

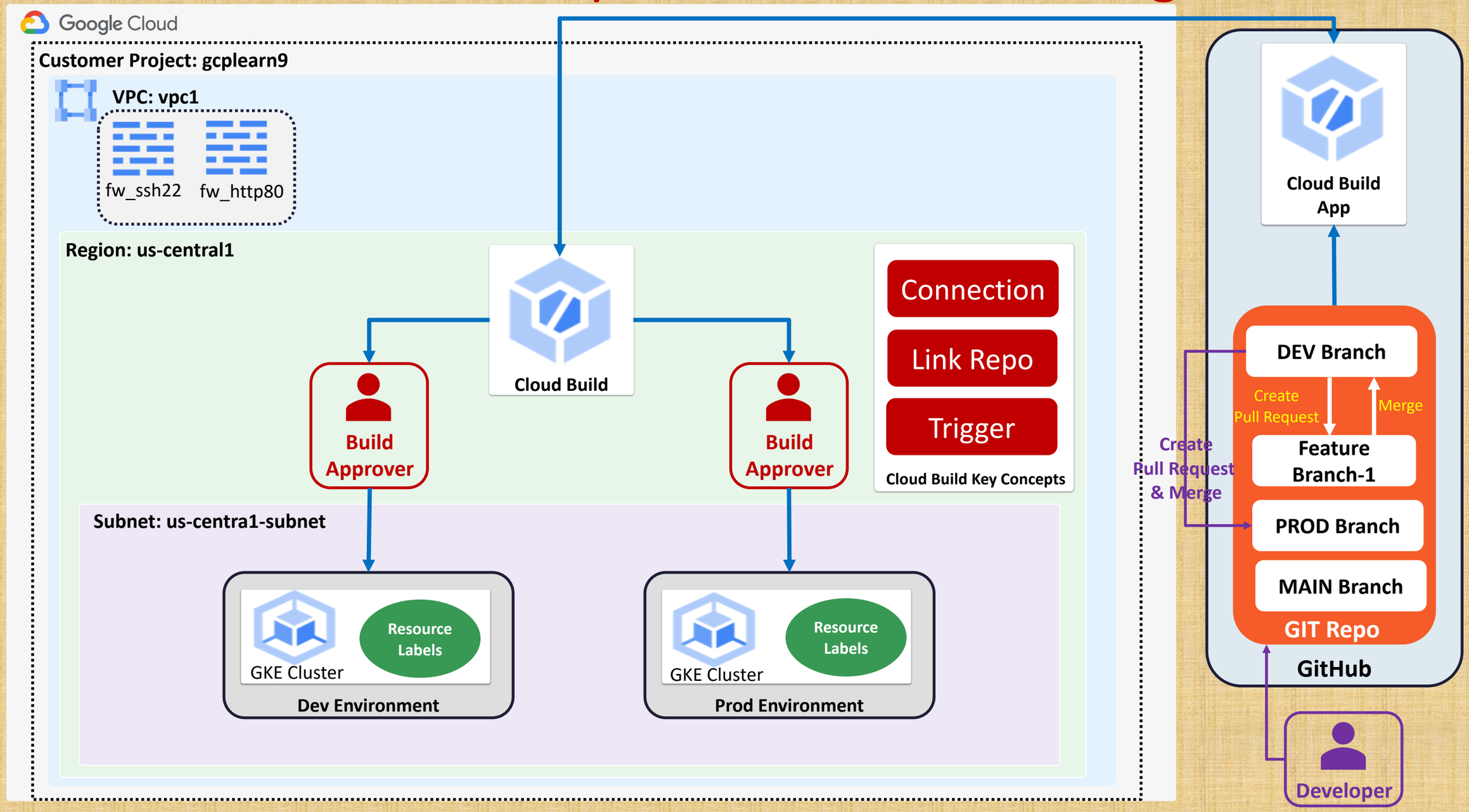
GCP GKE INFRA DevOps for Terraform Configs



Author: Nho Luong

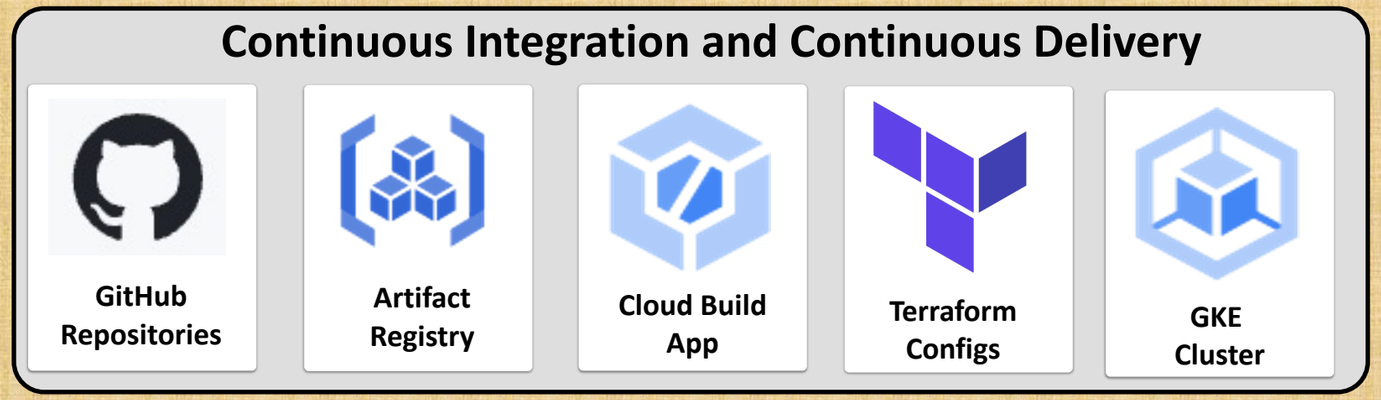
Skill: DevOps Engineer Lead

GCP GKE INFRA DevOps for Terraform Configs



Author: Nho Luong

Skill: DevOps Engineer Lead



Demo 25

Create Kubernetes Deployment Custom Terraform Module

Demo 26

Implement k8s workload DevOps pipelines
(Dev and Prod Environments)

Demo 27

App: Implement Continuous Integration Pipeline with Artifact Registry

Demo 28

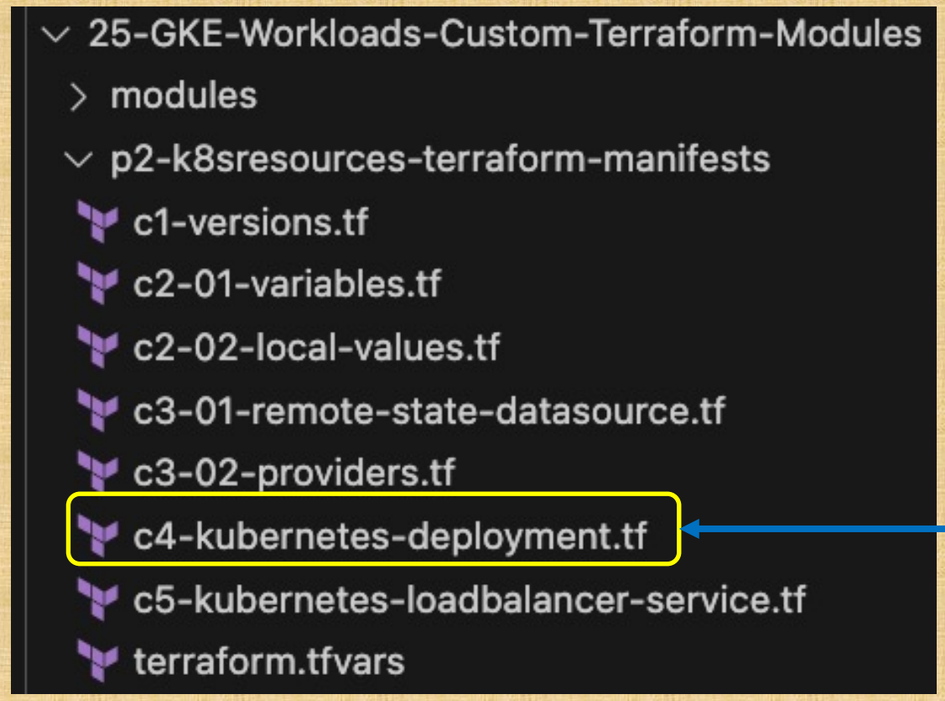
App: Implement Continuous Integration and Continuous Delivery Pipeline (CI CD)

Author: Nho Luong

Skill: DevOps Engineer Lead

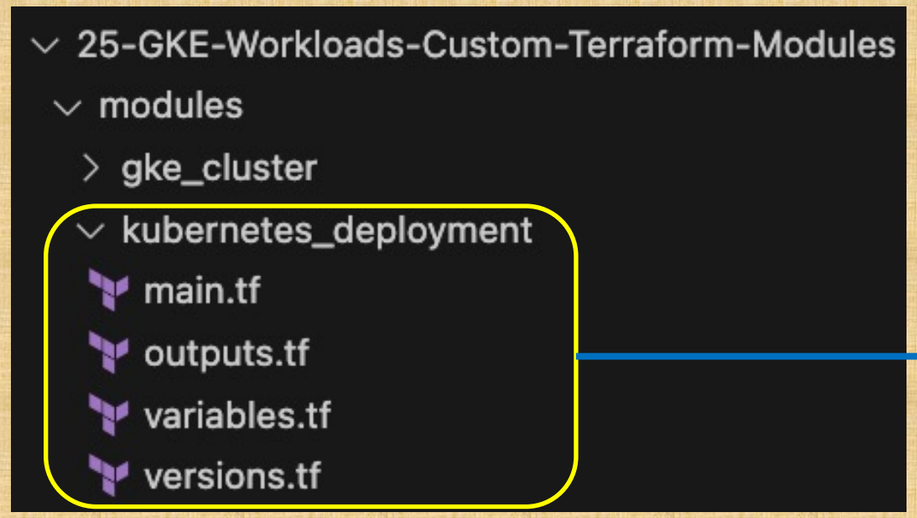
Build a Local Terraform Module

Root Module



Call GKE Cluster child module in Root Module

Child Module

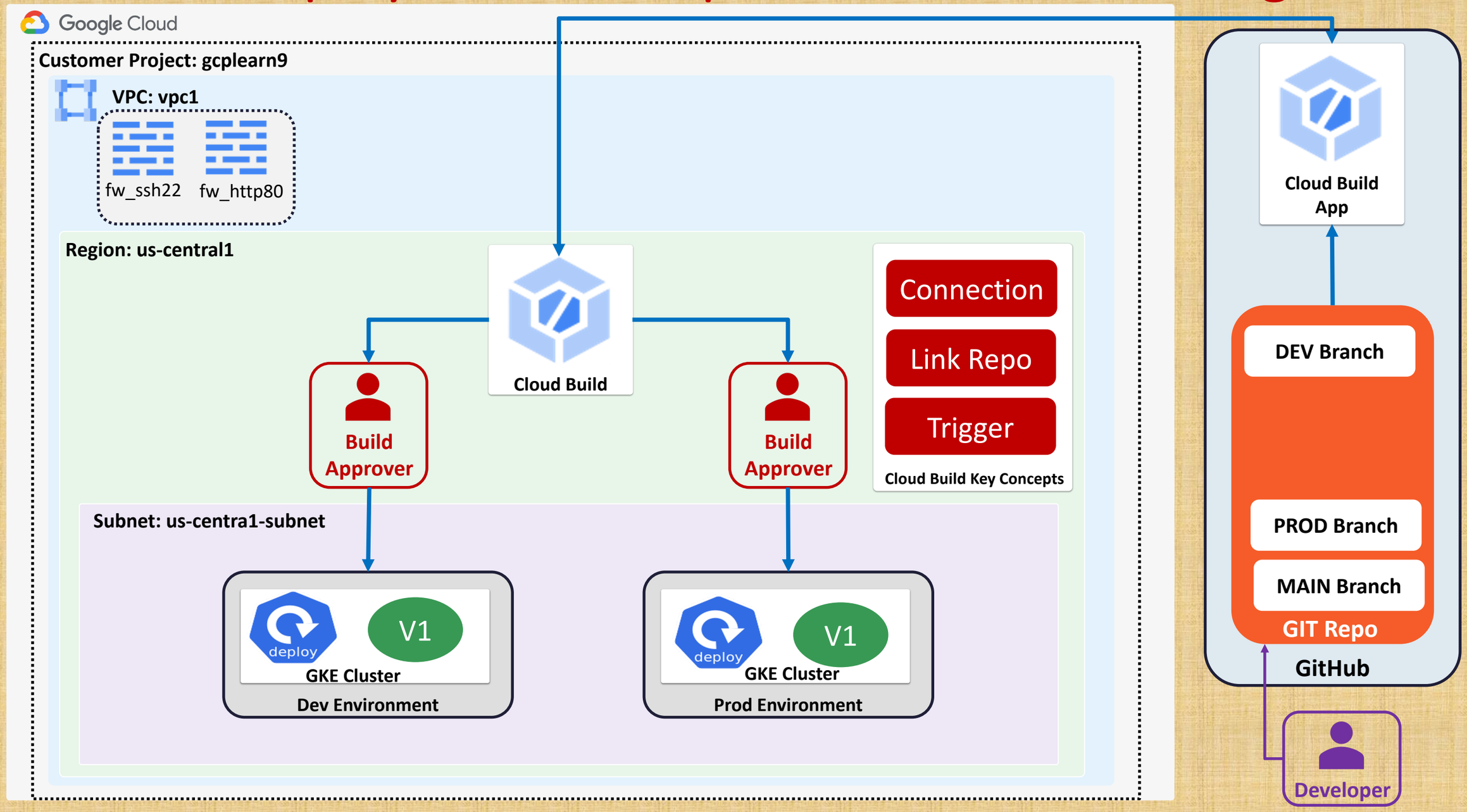


Kubernetes Deployment Custom Terraform Module

Author: Nho Luong

Skill: DevOps Engineer Lead

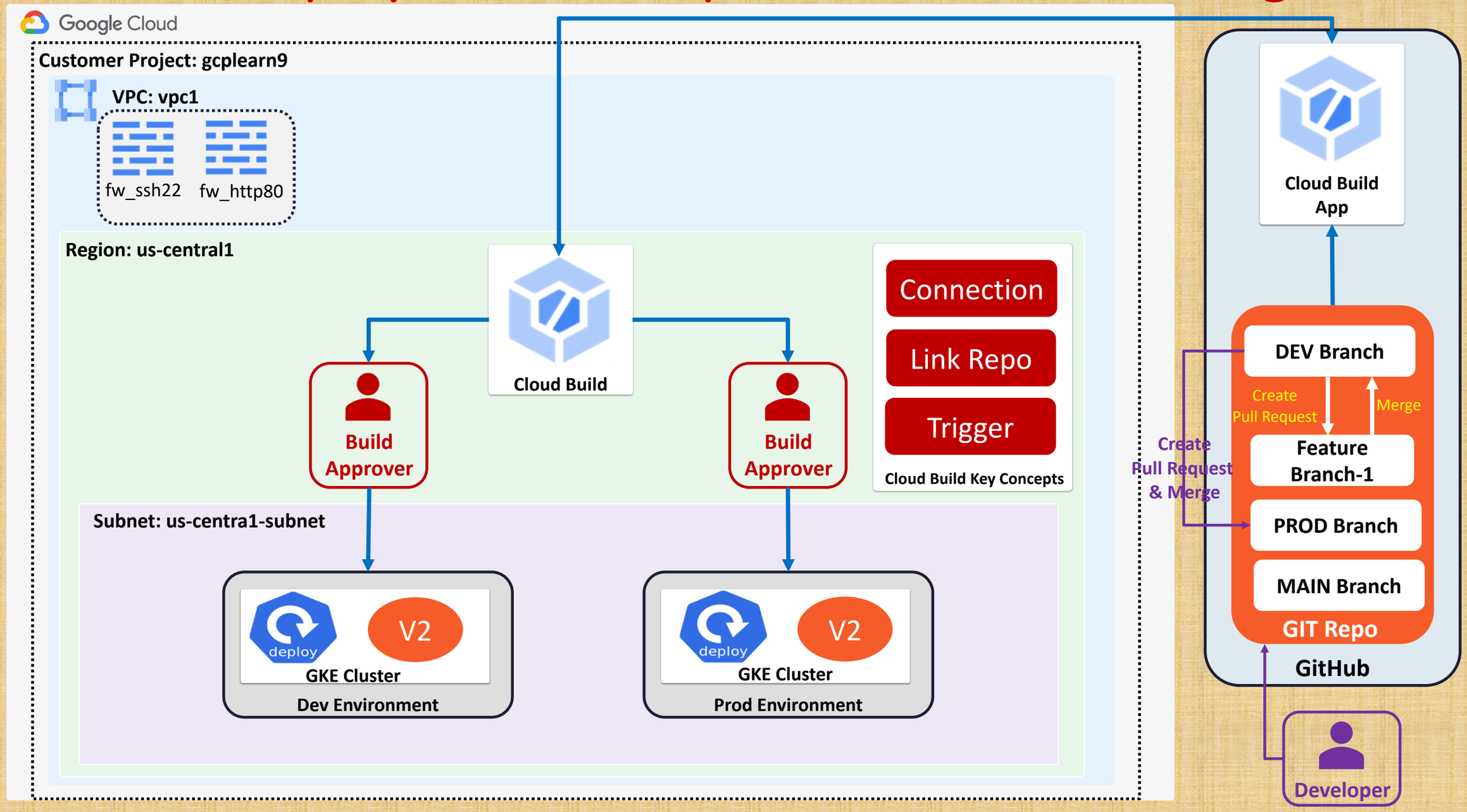
Kubernetes Deployment DevOps for Terraform Configs



Author: Nho Luong

Skill: DevOps Engineer Lead

Kubernetes Deployment DevOps for Terraform Configs



Author: Nho Luong

Skill: DevOps Engineer Lead

Continuous Integration

Continuous Integration Pipeline

GitHub

Google Cloud Build

Google Artifact Registry

To Store App Code (Application Code + Docker Files)

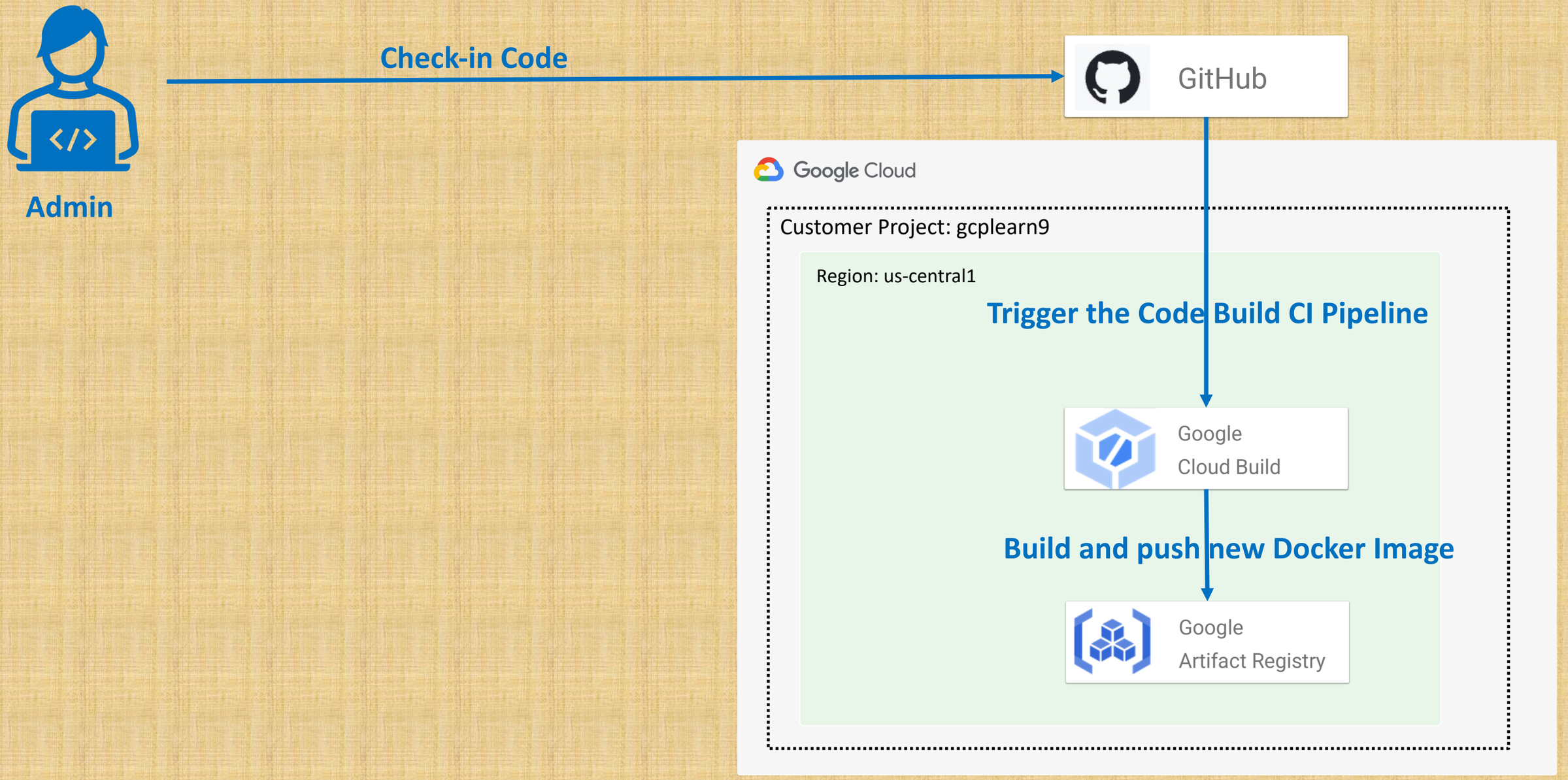
To Build Packages (Build Docker Image)

To Store Packages (Store Docker Image)

Author: Nho Luong

Skill: DevOps Engineer Lead

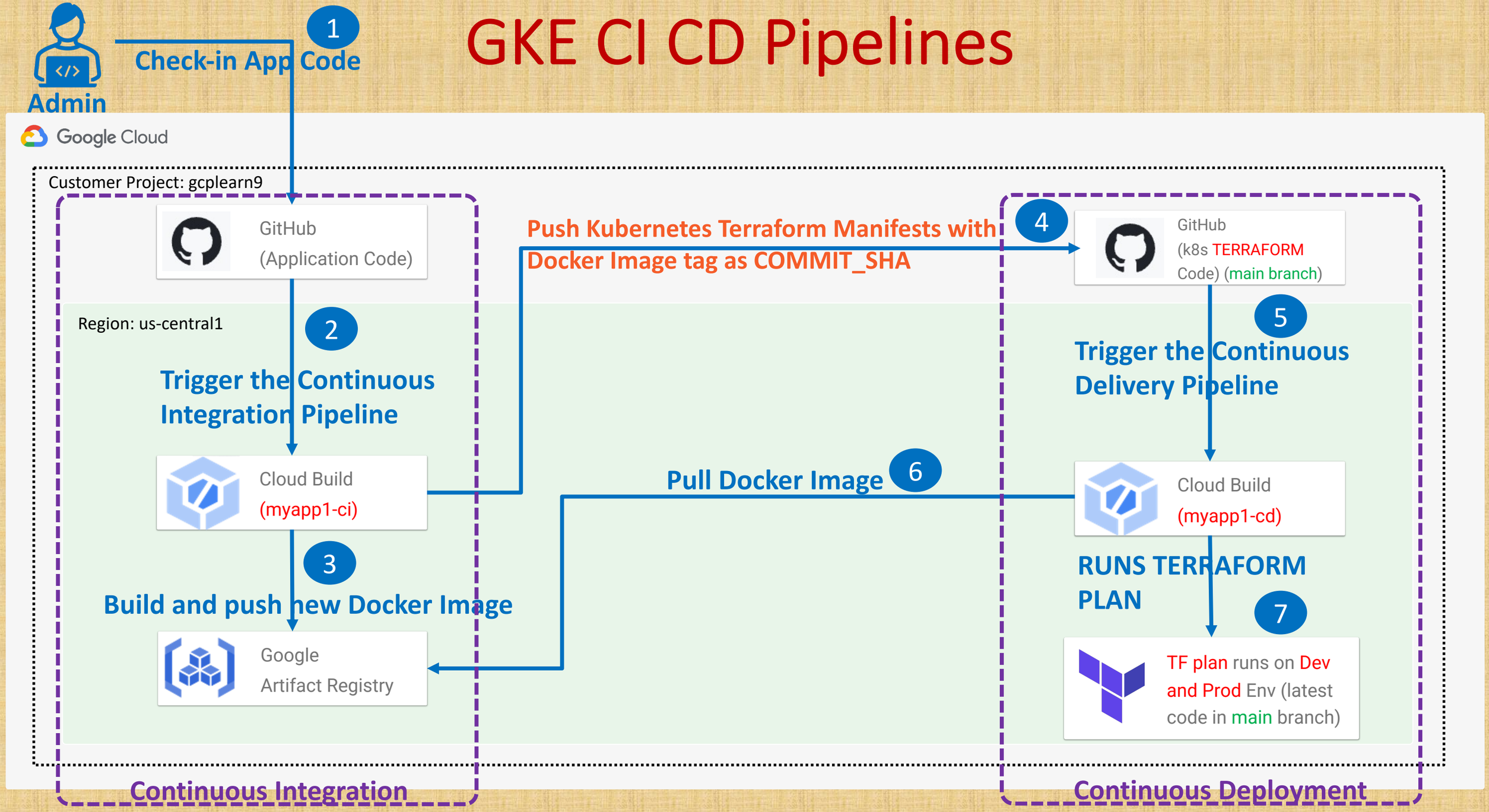
Google Cloud Build: Continuous Integration



Author: Nho Luong

Skill: DevOps Engineer Lead

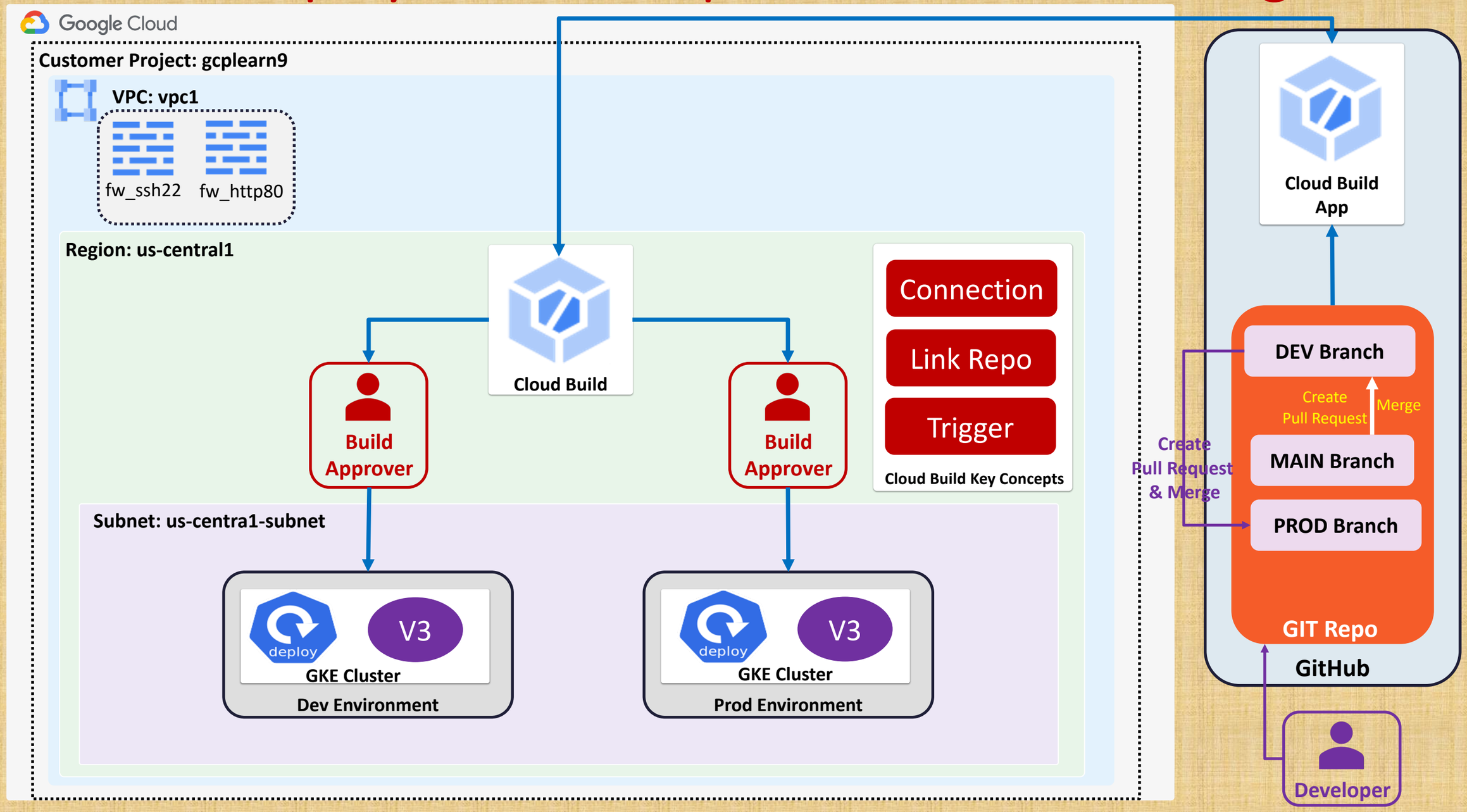
GKE CI CD Pipelines



Author: Nho Luong

Skill: DevOps Engineer Lead

Kubernetes Deployment DevOps for Terraform Configs



Author: Nho Luong

Skill: DevOps Engineer Lead

END OF INTRODUCTION

Author: Nho Luong

Skill: DevOps Engineer Lead

Terraform on Google Kubernetes Engine

Terraform Fundamentals	
Terraform CLI Install	1
Terraform Commands	2
Terraform Language Basics	3
Meta-Argument Provider	4
Input Variables & Output Values	5
Meta-Argument count	6
Terraform Data sources	7
Meta-Argument for_each	8

GCP Terraform on Google Cloud : DevOps SRE 30 Real-World Demos

Sections common

in both courses

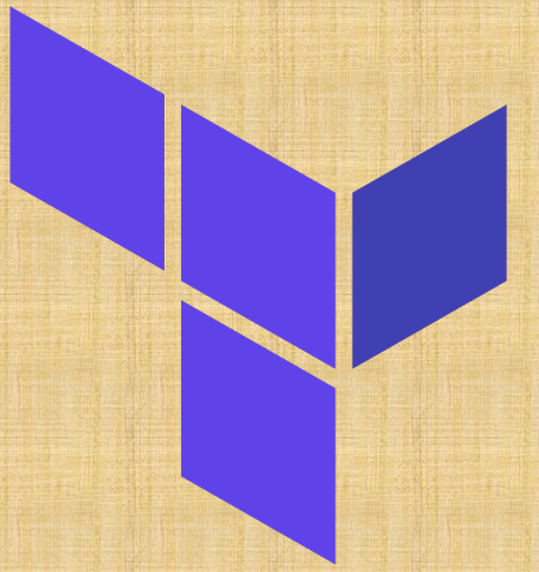
GCP GKE Terraform on Google Kubernetes Engine DevOps SRE IaC

Author: Nho Luong

Skill: DevOps Engineer Lead

Demo

Terraform Installation CLI Tools



Author: Nho Luong

Skill: DevOps Engineer Lead

Terraform Installation

gcloud CLI

Terraform CLI

VS Code Editor

Terraform plugin for VS Code

Mac OS

Windows OS

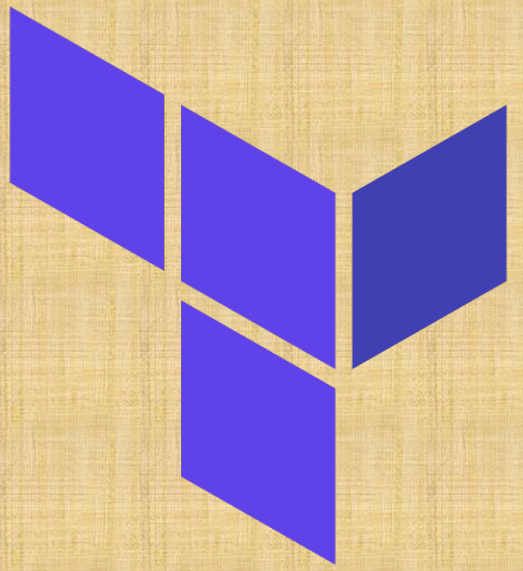
Author: Nho Luong

Skill: DevOps Engineer Lead

Demo

Terraform

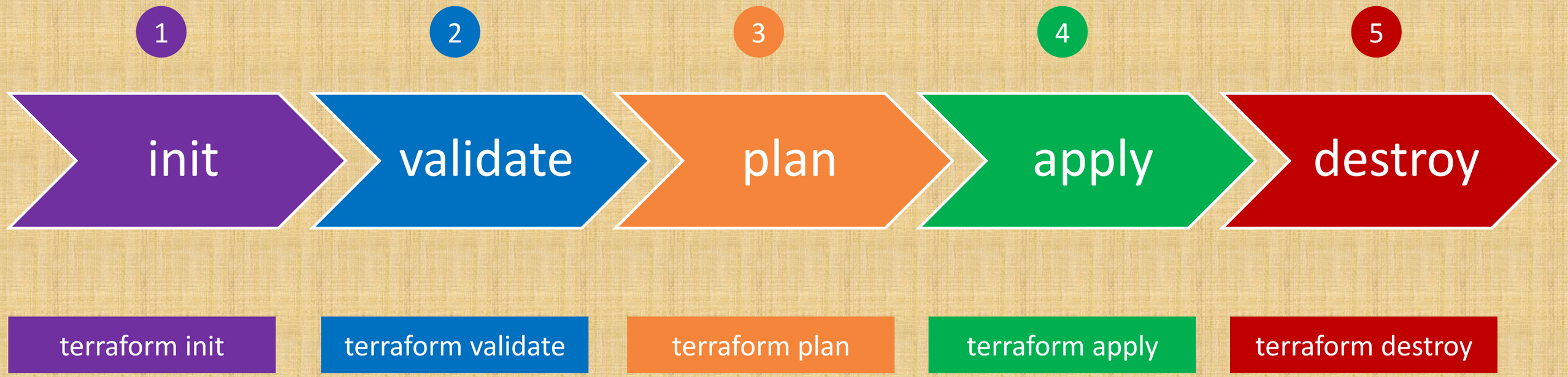
Command Basics



Author: Nho Luong

Skill: DevOps Engineer Lead

Terraform Workflow



Author: Nho Luong

Skill: DevOps Engineer Lead

Terraform Workflow

1 2 3 4 5

init

- Used to Initialize a working directory containing terraform config files
- This is the first command that should be run after writing a new Terraform configuration
- Downloads Providers

validate

- Validates the terraform configurations files in that respective directory to ensure they are syntactically valid and internally consistent.

plan

- Creates an execution plan
- Terraform performs a refresh and determines what actions are necessary to achieve the desired state specified in configuration files

apply

- Used to apply the changes required to reach the desired state of the configuration.
- By default, apply scan s the current directory for the configuration and applies the changes appropriately.

destroy

- Used to destroy the Terraform-managed infrastructure
- This will ask for confirmation before destroying.

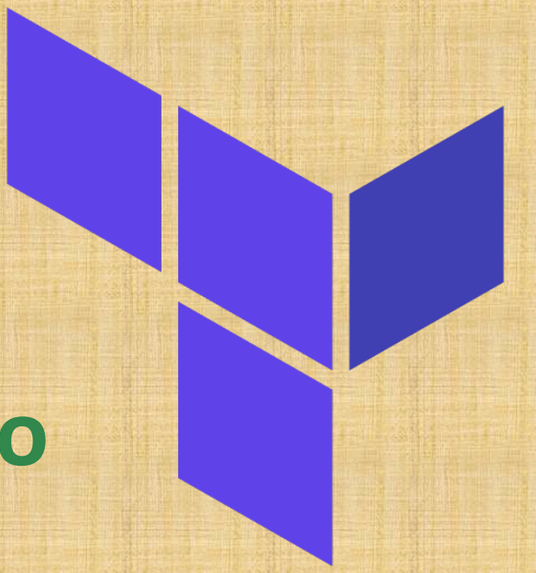
Author: Nho Luong

Skill: DevOps Engineer Lead

Demo

Terraform

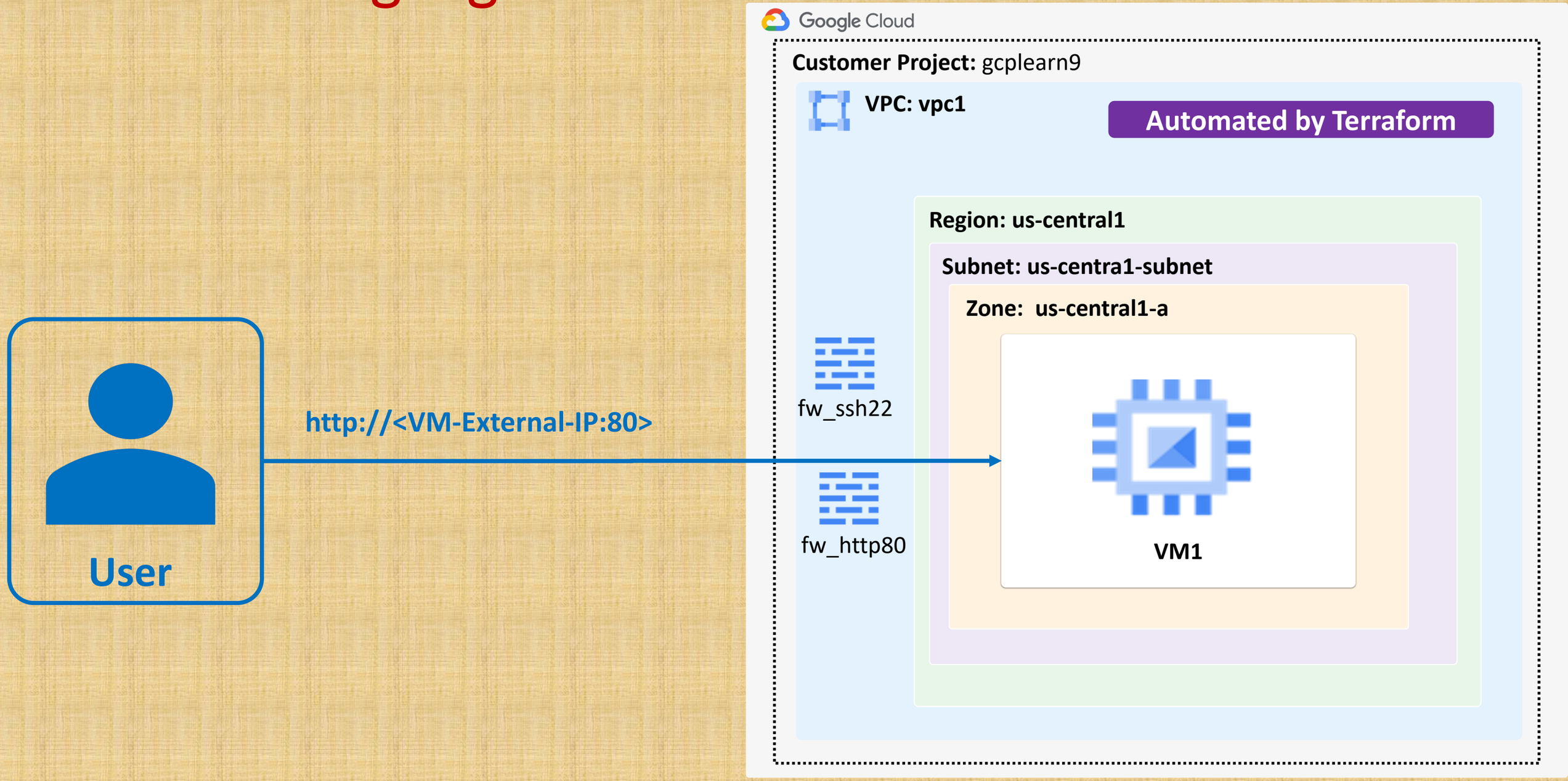
What are we going to automate as part of this demo using Terraform?



Author: Nho Luong

Skill: DevOps Engineer Lead

Terraform Language Basics



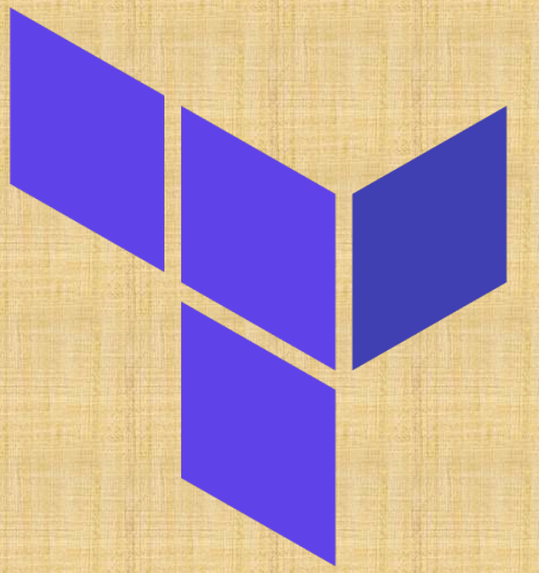
Author: Nho Luong

Skill: DevOps Engineer Lead

Demo

Terraform

Language Basics

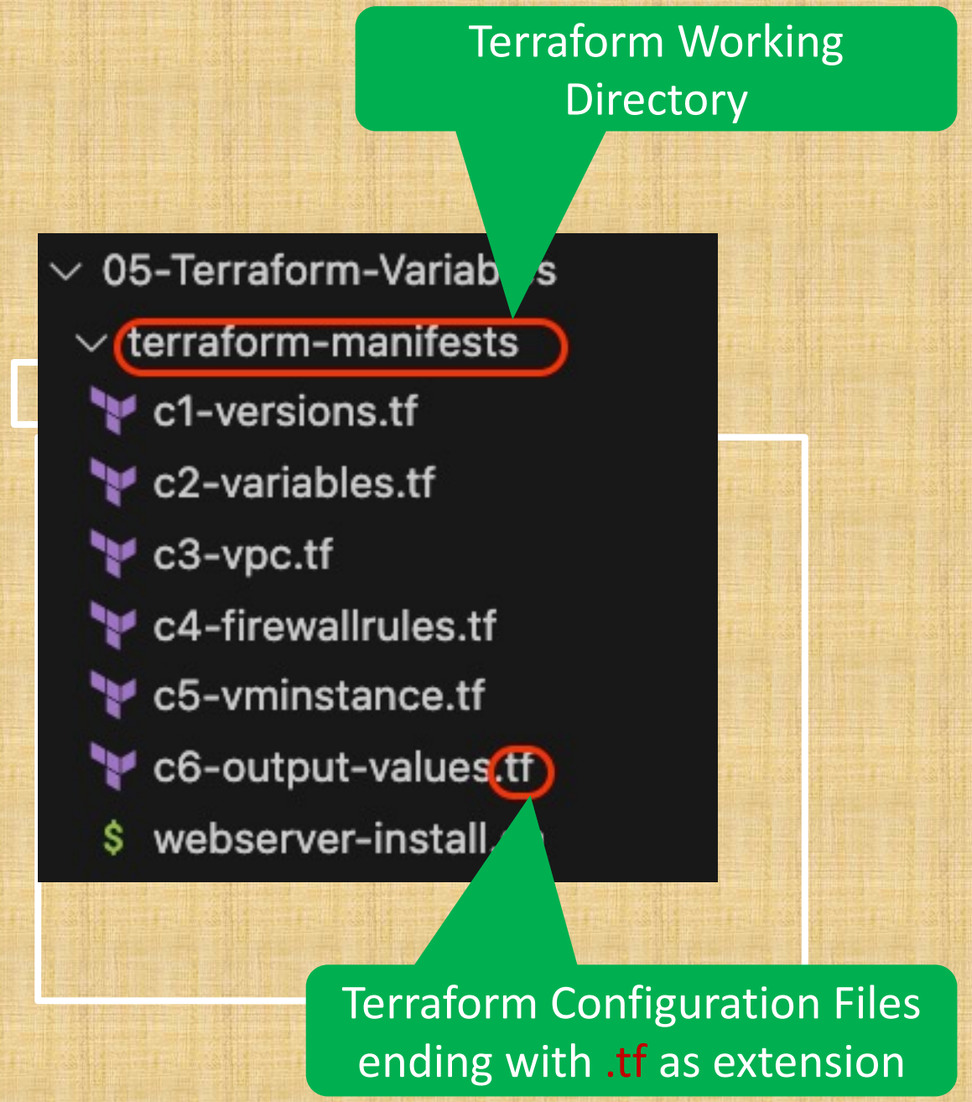


Author: Nho Luong

Skill: DevOps Engineer Lead

Terraform Language Basics - Files

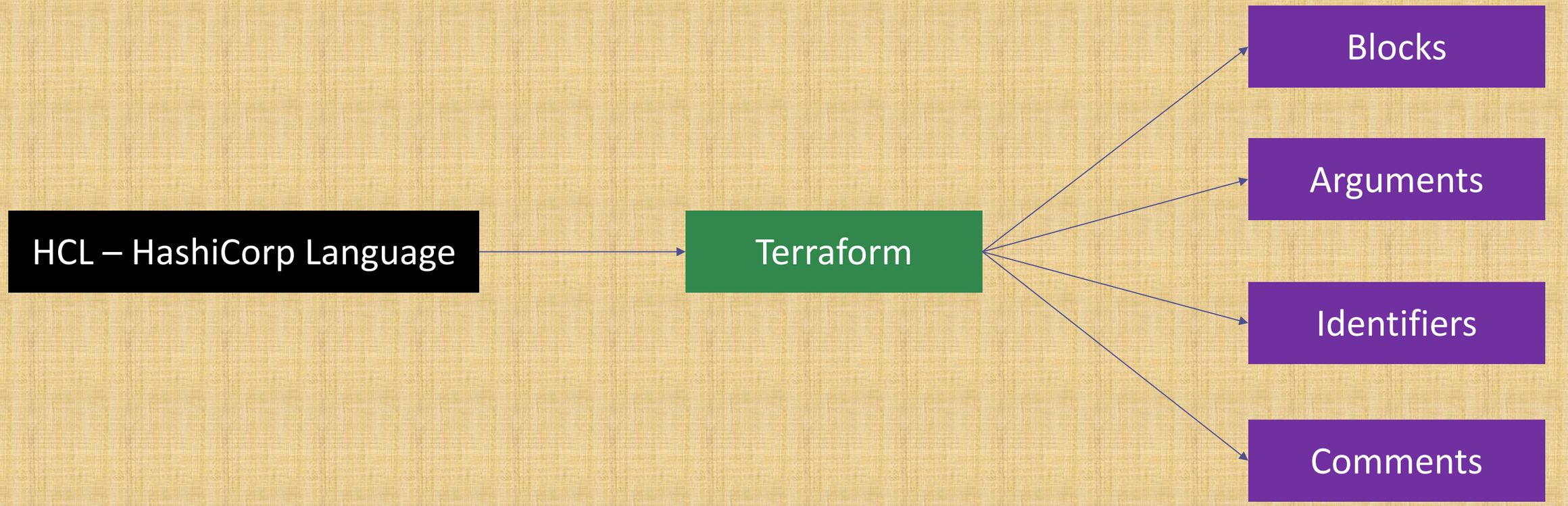
- Code in the Terraform language is stored in plain text files with the .tf file extension.
- There is also a JSON-based variant of the language that is named with the .tf.json file extension.
- We can call the files containing terraform code as Terraform Configuration Files or Terraform Manifests
- Terraform files should be saved in a directory (Terraform Working Directory)



Author: Nho Luong

Skill: DevOps Engineer Lead

Terraform Language Basics - Configuration Syntax



Author: Nho Luong

Skill: DevOps Engineer Lead

Terraform Language Basics - Configuration Syntax

```
# Template
<BLOCK TYPE> "<BLOCK LABEL>" "<BLOCK LABEL>" {
  # Block Body
  <IDENTIFIER> = <EXPRESSION> # Argument
}

# Resource: Subnet
resource "google_compute_subnetwork" "mysubnet" {
  name = "subnet1"
  region = "us-central1"
  ip_cidr_range = "10.128.0.0/20"
  network = google_compute_network.myvpc.id
}
```

Based on Block Type, block labels can be 1 or 2
Example:
Resources: 2 block labels
Variables: 1 block label

Block Labels

Block Type

Arguments

Top Level & Block inside Blocks

Top Level Blocks: resource, provider, terraform, variable, data, module, output

Block Inside Block: provisioners, resource specific blocks like tags

Author: Nho Luong

Skill: DevOps Engineer Lead

Terraform Language Basics - Configuration Syntax

```
# Template
<BLOCK TYPE> "<BLOCK LABEL>" "<BLOCK LABEL>" {
  # Block Body
  <IDENTIFIER> = <EXPRESSION> # Argument
}

# Resource: Subnet
resource "google_compute_subnetwork" "mysubnet" {
  name          = "subnet1"
  region        = "us-central1"
  ip_cidr_range = "10.128.0.0/20"
  network       = google_compute_network.myvpc.id
}
```

Argument Name [OR] Identifier

Argument Value [OR] Expression

Author: Nho Luong

Skill: DevOps Engineer Lead

Terraform Language Basics - Configuration Syntax

Multi-line comment /* ..... */

Single Line Comments with # or //

```
/*
# Template
<BLOCK TYPE> "<BLOCK LABEL>" "<BLOCK LABEL>" {
  # Block Body
  <IDENTIFIER> = <EXPRESSION> # Argument
}
*/

# Resource: Subnet
resource "google_compute_subnetwork" "mysubnet" {
  name          = "subnet1"
  region        = "us-central1"
  ip_cidr_range = "10.128.0.0/20"
  network       = google_compute_network.myvpc.id // GET VPC ID
}
```

Author: Nho Luong

Skill: DevOps Engineer Lead

Terraform language uses a limited number of **top-level block** types, which are **blocks** that can appear outside of any other **block** in a TF configuration file.

Terraform Top-Level Blocks

Most of **Terraform's features** are implemented as **top-level** blocks.

Fundamental Blocks

- Terraform Block
- Providers Block
- Resources Block

Variable Blocks

- Input Variables Block
- Output Values Block
- Local Values Block

Calling / Referencing Blocks

- Data Sources Block
- Modules Block

NEW BLOCKS – Added Recently

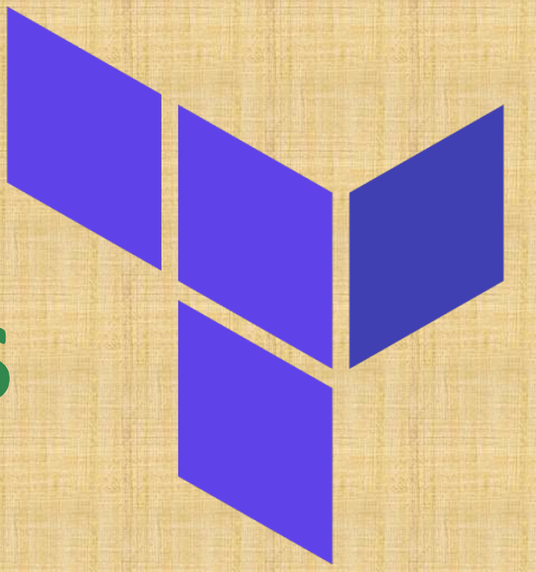
- Import Block
- Moved Block
- Removed Block
- Check Block

Author: Nho Luong

Skill: DevOps Engineer Lead

Demo

Terraform
Fundamental Blocks



Author: Nho Luong

Skill: DevOps Engineer Lead

Terraform Basic Blocks

Terraform Block

- Special block used to configure some behaviors
- Specifying a required Terraform Version
- Specifying Provider Requirements
- Configuring a Terraform Backend (Terraform State)

Provider Block

- HEART of Terraform
- Terraform relies on providers to interact with Remote Systems
- Declare providers for Terraform to install providers & use them
- Provider configurations belong to Root Module

Resource Block

- Each Resource Block describes one or more Infrastructure Objects
- Resource Syntax: How to declare Resources?
- Resource Behavior: How Terraform handles resource declarations?
- Provisioners: We can configure Resource post-creation actions

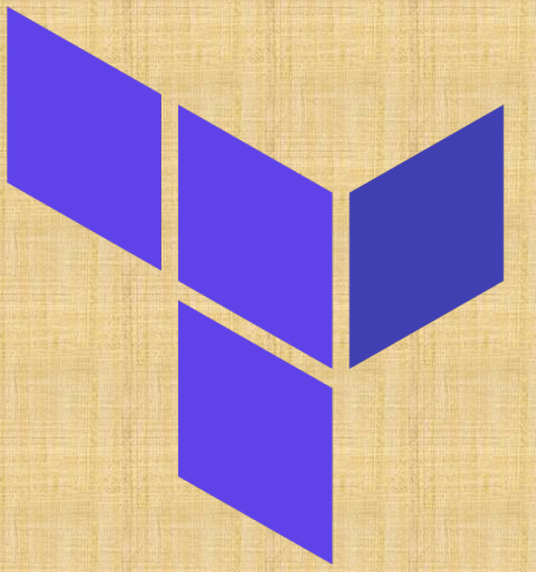
Author: Nho Luong

Skill: DevOps Engineer Lead

Demo

Terraform

Terraform Block



Author: Nho Luong

Skill: DevOps Engineer Lead

Terraform Block

- This block can be called in 3 ways. All means the same.
  - Terraform Block
  - Terraform Settings Block
  - Terraform Configuration Block
- Each terraform block can contain a number of settings related to Terraform's behavior.
- Within a terraform block, only constant values can be used; arguments may not refer to named objects such as resources, input variables, etc, and may not use any of the Terraform language built-in functions.

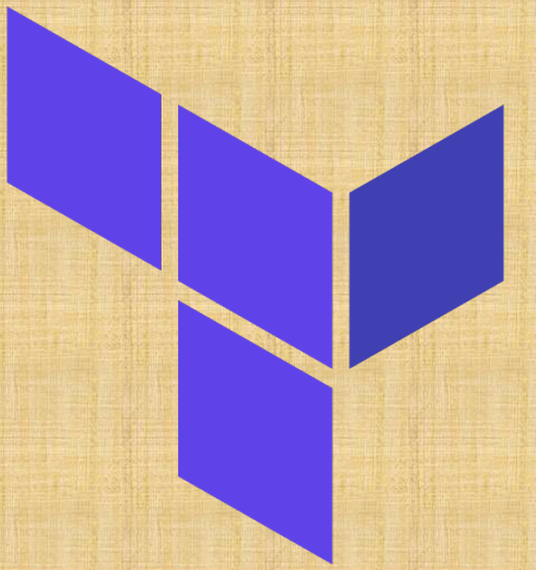
```
# Terraform Settings Block
terraform {
  required_version = ">= 1.8"
  required_providers {
    google = {
      source = "hashicorp/google"
      version = ">= 5.26.0"
    }
  }
 backend "gcs" {
   bucket  = "kalyanbucket201"
   prefix  = "terraform/state"
 }
}
```

Author: Nho Luong

Skill: DevOps Engineer Lead

Demo

Terraform Providers



Author: Nho Luong

Skill: DevOps Engineer Lead

Terraform Providers

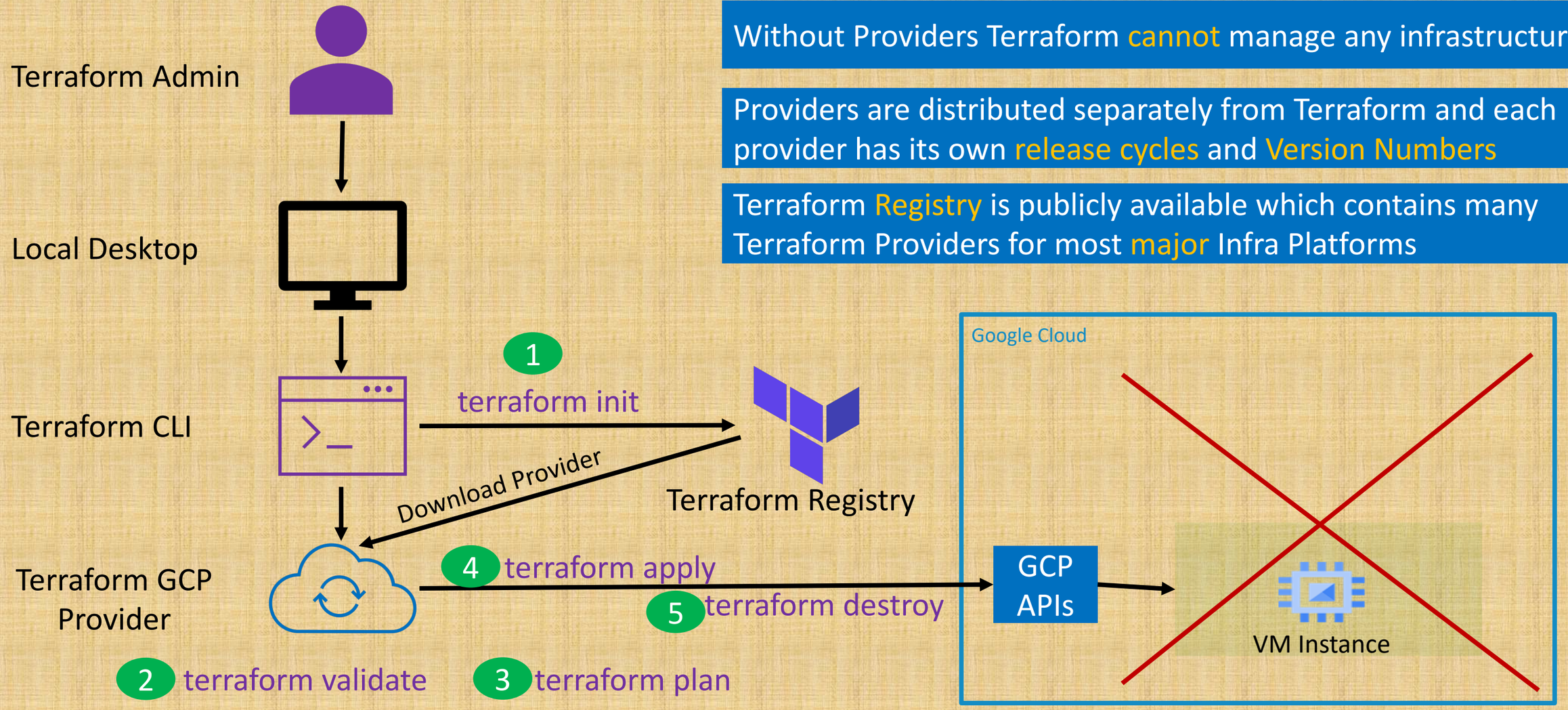
Providers are HEART of Terraform

Every Resource Type (example: VM Instance), is implemented by a Provider

Without Providers Terraform cannot manage any infrastructure.

Providers are distributed separately from Terraform and each provider has its own release cycles and Version Numbers

Terraform Registry is publicly available which contains many Terraform Providers for most major Infra Platforms



Author: Nho Luong

Skill: DevOps Engineer Lead

Terraform Providers

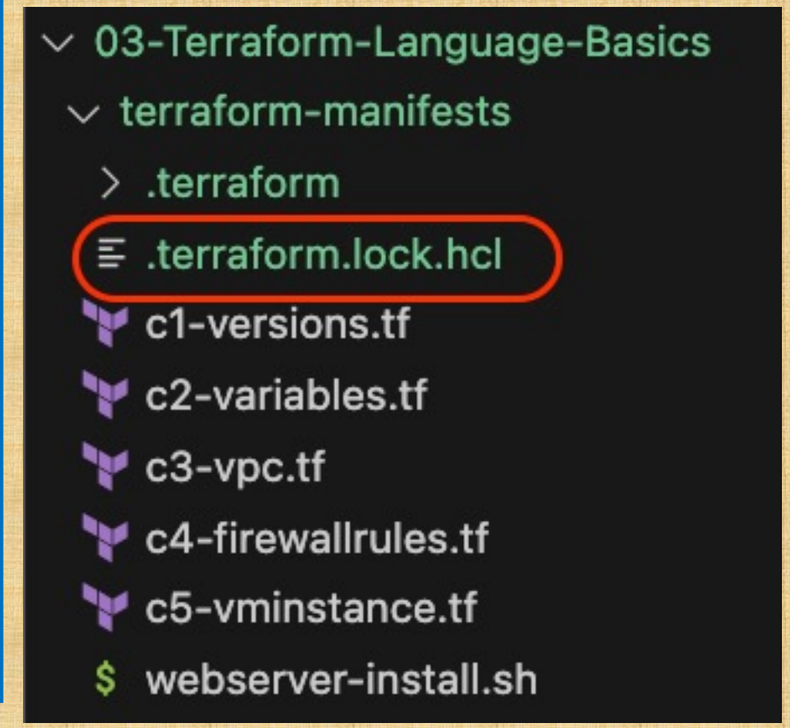
Provider Requirements

```
# Terraform Settings Block
terraform {
  required_version = ">= 1.8"
  required_providers {
    google = {
      source = "hashicorp/google"
      version = ">= 5.26.0"
    }
  }
}
```

Provider Configuration

```
# Terraform Provider Block
provider "google" {
  project = "gcplearn9"
  region = "us-central1"
}
```

Dependency Lock File



Author: Nho Luong

Skill: DevOps Engineer Lead

Dependency Lock File

terraform-on-google-cloud > 03-Terraform-Language-Basics > terraform-manifests > .terraform.lock.hcl

```
1  # This file is maintained automatically by "terraform init".
2  # Manual edits may be lost in future updates.
3
4  provider "registry.terraform.io/hashicorp/google" {
5    version     = "5.26.0"
6    constraints = ">= 5.26.0"
```

Author: Nho Luong

Skill: DevOps Engineer Lead

Required Providers

```
# Terraform Settings Block
terraform {
  required_version = ">= 1.8"
  required_providers {
    google = {
      source = "hashicorp/google"
      version = ">= 5.26.0"
    }
  }
}

# Terraform Provider Block
provider "google" {
  project = "gcplearn9"
  region = "us-central1"
}
```

Local Names

Local Names are Module specific and should be unique per-module

Terraform configurations always refer to local name of provider outside required_provider block

Users of a provider can choose any local name for it (mygoogle, google1).

Recommended way of choosing local name is to use preferred local name of that provider (For Google Provider: hashicorp/google, preferred local name is google)

Source

It is the primary location where we can download the Terraform Provider

Source addresses consist of three parts delimited by slashes (/)

[<HOSTNAME>/]<NAMESPACE>/<TYPE>

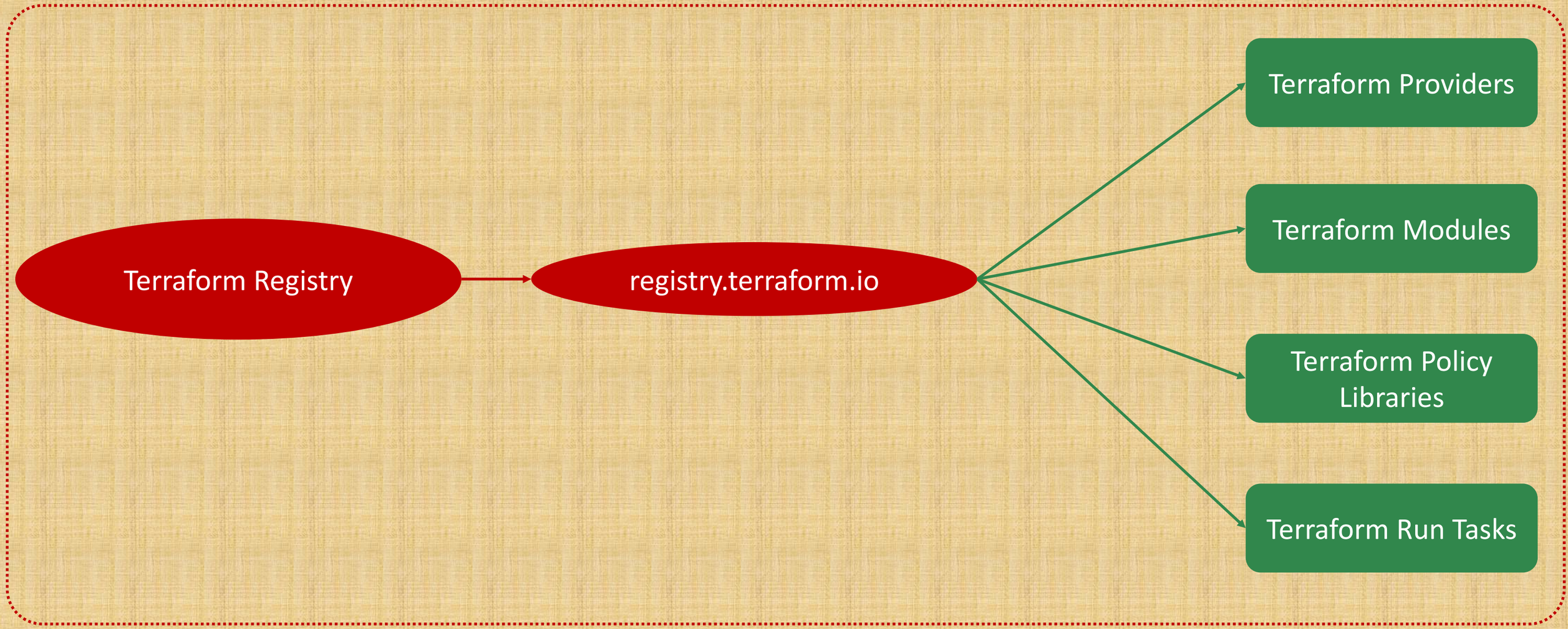
registry.terraform.io/hashicorp/google

Registry Name is optional as default is going to be Terraform Public Registry

Author: Nho Luong

Skill: DevOps Engineer Lead

Terraform Registry

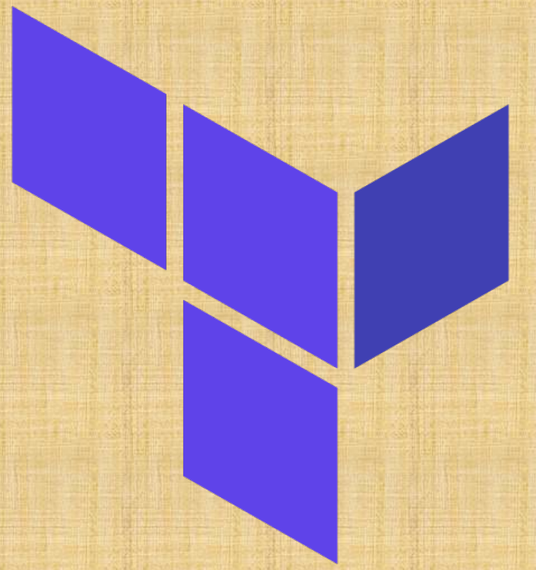


Author: Nho Luong

Skill: DevOps Engineer Lead

Demo

Terraform State



Author: Nho Luong

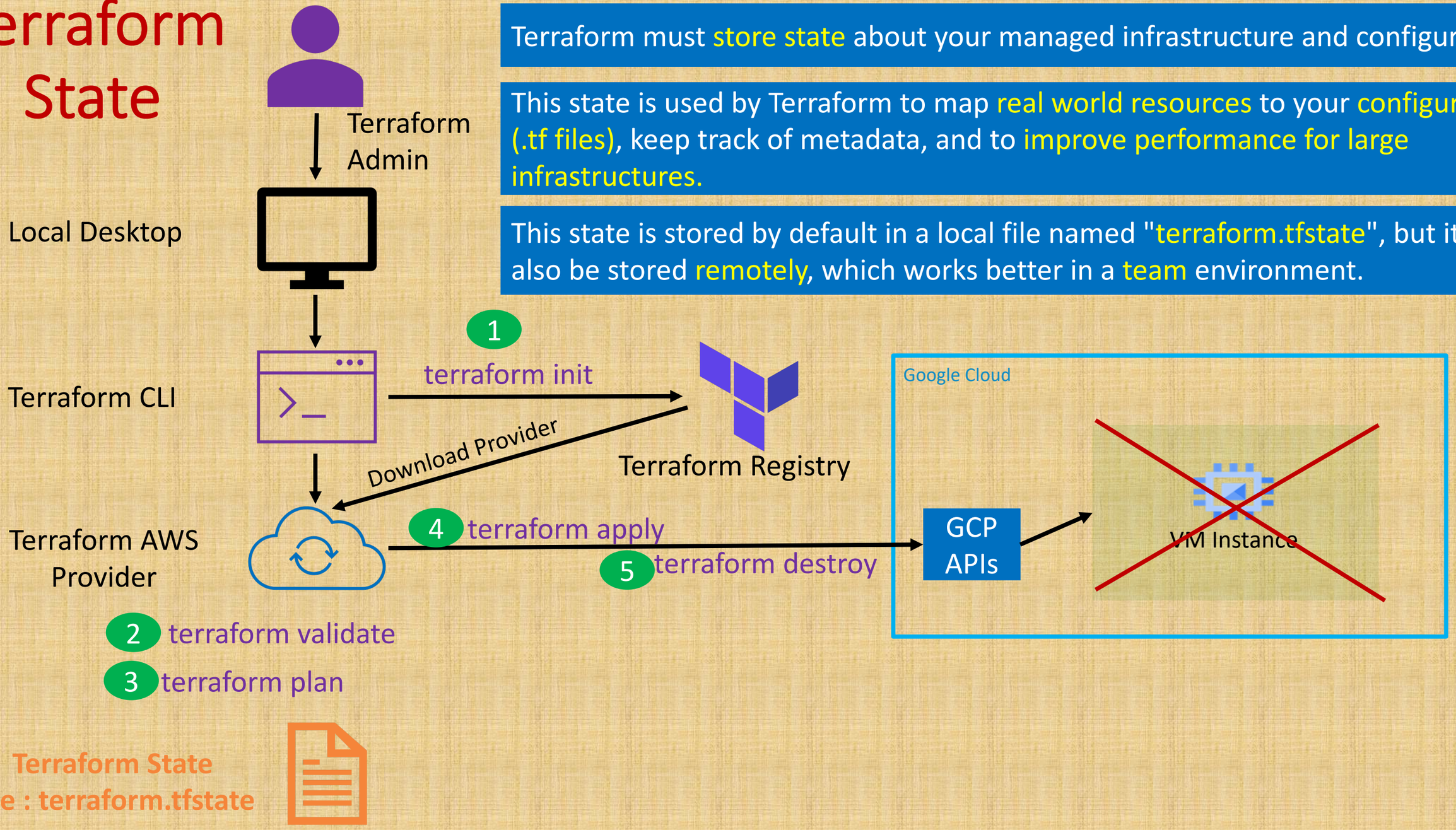
Skill: DevOps Engineer Lead

Terraform State

Terraform must store state about your managed infrastructure and configuration

This state is used by Terraform to map real world resources to your configuration (.tf files), keep track of metadata, and to improve performance for large infrastructures.

This state is stored by default in a local file named "terraform.tfstate", but it can also be stored remotely, which works better in a team environment.



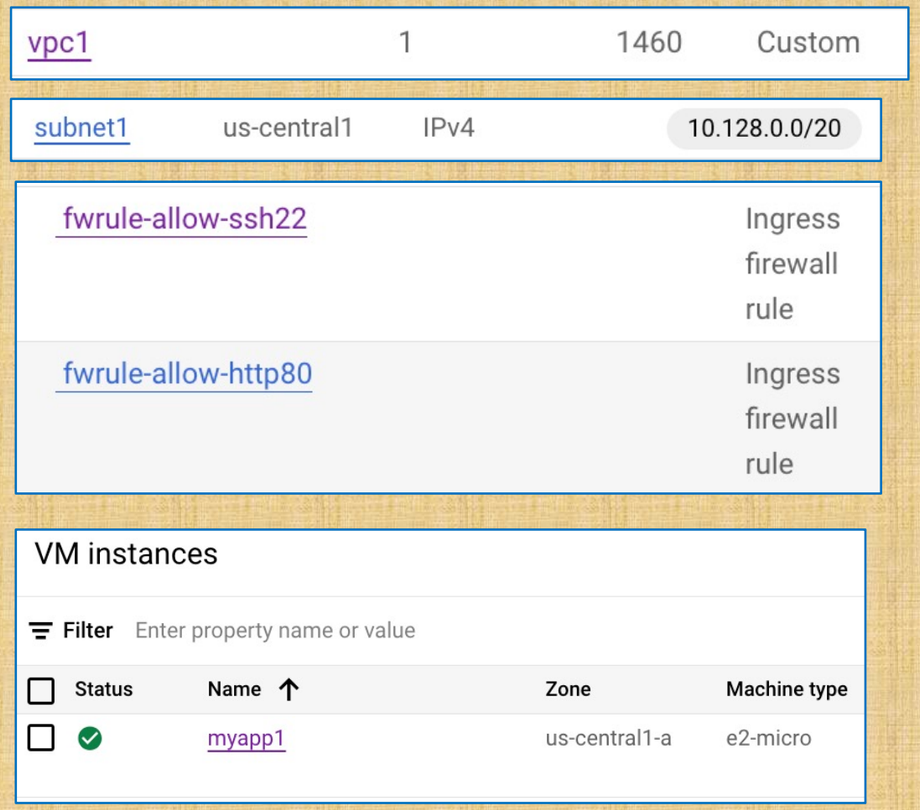
Author: Nho Luong

Skill: DevOps Engineer Lead

Desired & Current Terraform States

Terraform Configuration Files

Real World Resource
VM Instance, VPC, Subnet,
Firewall Rules



Desired State

Current State

Author: Nho Luong

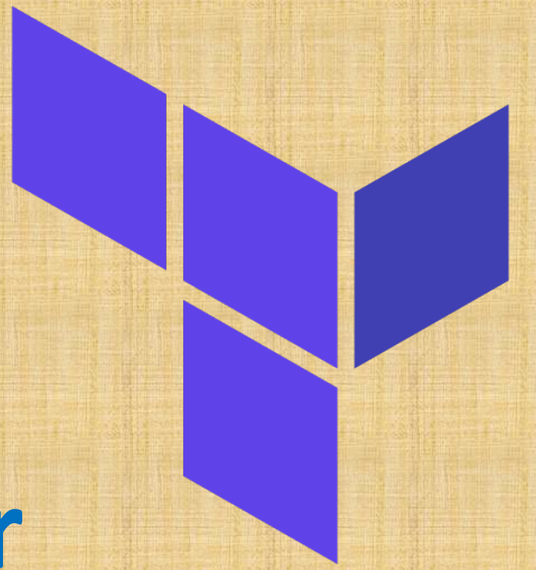
Skill: DevOps Engineer Lead

Demo-4

Terraform

Multiple Providers

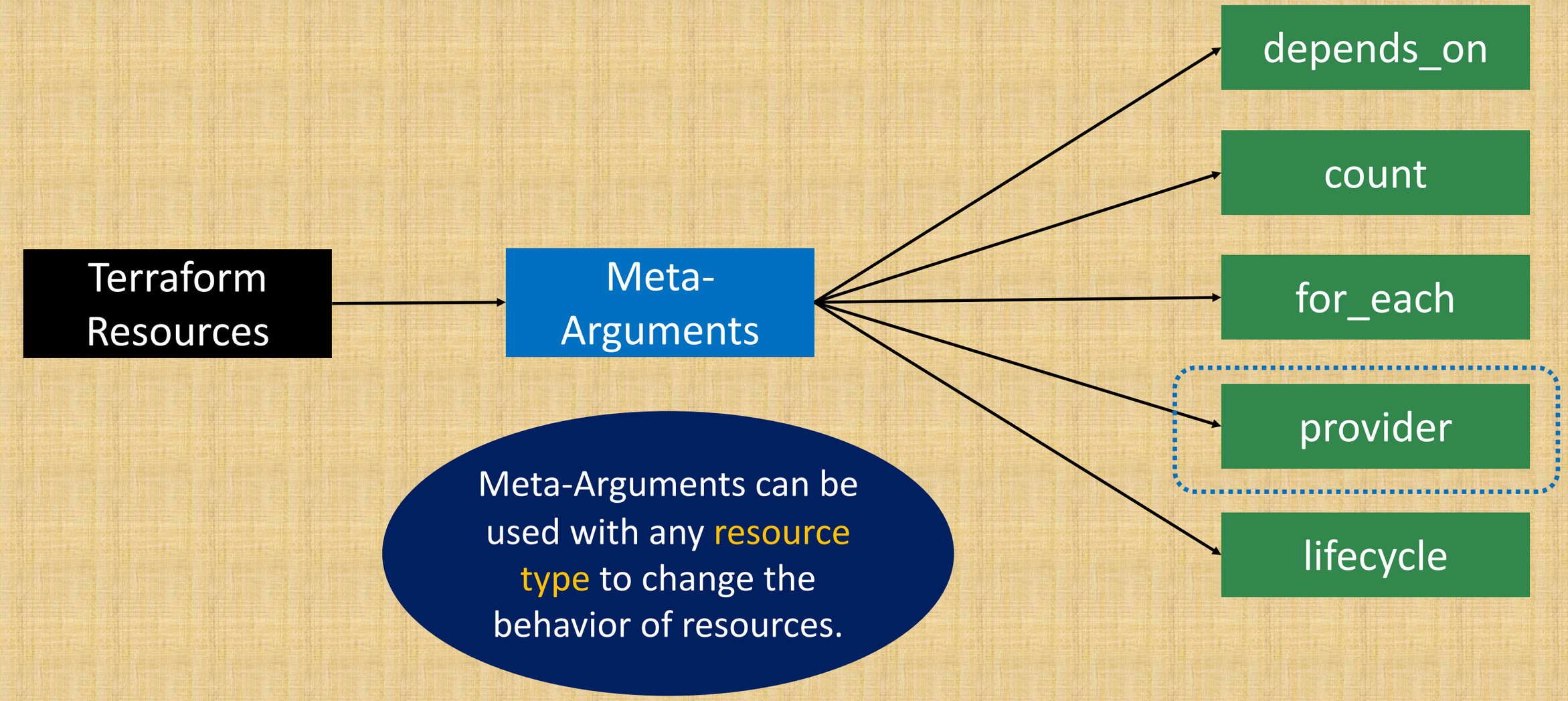
Meta-Argument: provider



Author: Nho Luong

Skill: DevOps Engineer Lead

Resource Meta-Arguments



Author: Nho Luong

Skill: DevOps Engineer Lead

Terraform Multiple Providers

- We can define multiple configurations for the same provider (Ex: google) and select which one to use on a per-resource or per-module basis.
- The primary reason for this is to support multiple regions for a cloud platform, additionally for multiple environments (dev, prod) considering each environment has different provider configurations (Example: different authentication credentials for dev and prod)
- We can use the alternate provider in a resource, data or module by referencing it as <PROVIDER NAME>.<ALIAS>

```
# Terraform Provider-1: us-central1
provider "google" {
  project = "gcplearn9"
  region = "us-central1"
  alias = "us-central1"
}

# Terraform Provider-2: europe-west1
provider "google" {
  project = "gcplearn9"
  region = "europe-west1"
  alias = "europe-west1"
}
```

```
# Resource: Subnet1
resource "google_compute_subnetwork" "mysubnet1" {
  provider = google.us-central1   # Define provider to use
  name = "subnet1"
  ip_cidr_range = "10.128.0.0/20"
  network = google_compute_network.myvpc.id
}

# Resource: Subnet2
resource "google_compute_subnetwork" "mysubnet2" {
  provider = google.europe-west1   # Define provider to use
  name = "subnet2"
  ip_cidr_range = "10.132.0.0/20"
  network = google_compute_network.myvpc.id
}
```

Author: Nho Luong

Skill: DevOps Engineer Lead

Terraform Multiple Providers

- We can define multiple configurations for the same provider (Ex: google) and select which one to use on a per-resource or per-module basis.
- **Usage**
  - To support multiple regions for a cloud platform (we will practically implement this)
  - For supporting multiple environments (dev, prod) considering each environment has different provider configurations
    - **Example:** Different authentication credentials for dev and prod configured in their respective provider blocks
- We can use the alternate provider in a resource, data or module by referencing it as <PROVIDER NAME>.<ALIAS>

```
# Terraform Provider-1: us-central1
provider "google" {
  project = "gcplearn9"
  region = "us-central1"
  alias = "us-central1"
}

# Terraform Provider-2: europe-west1
provider "google" {
  project = "gcplearn9"
  region = "europe-west1"
  alias = "europe-west1"
}
```

```
# Resource: Subnet1
resource "google_compute_subnetwork" "mysubnet1" {
  provider = google.us-central1   # Define provider to use
  name = "subnet1"
  ip_cidr_range = "10.128.0.0/20"
  network = google_compute_network.myvpc.id
}

# Resource: Subnet2
resource "google_compute_subnetwork" "mysubnet2" {
  provider = google.europe-west1   # Define provider to use
  name = "subnet2"
  ip_cidr_range = "10.132.0.0/20"
  network = google_compute_network.myvpc.id
}
```

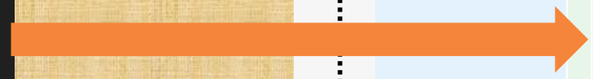
Author: Nho Luong

Skill: DevOps Engineer Lead

Terraform Multiple Providers

```hcl
# Terraform Provider-1: us-central1
provider "google" {
  project = "gcplearn9"
  region = "us-central1"
  alias = "us-central1"
}
```

```hcl
# Terraform Provider-2: europe-west1
provider "google" {
  project = "gcplearn9"
  region = "europe-west1"
  alias = "europe-west1"
}
```



Author: Nho Luong

Skill: DevOps Engineer Lead

What are we going to learn?

Demo: Terraform Concept: Multiple Providers

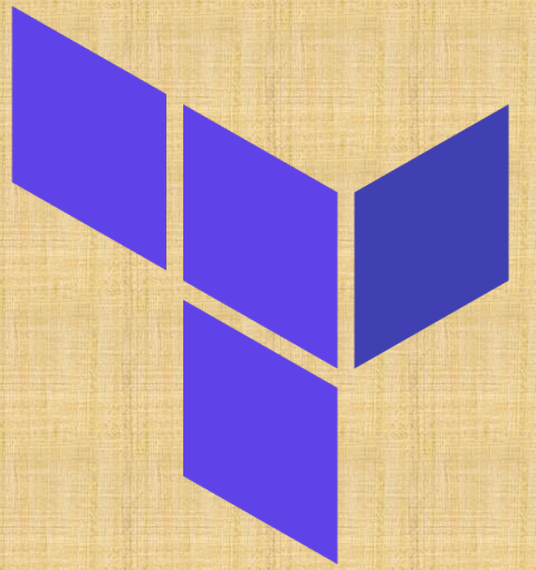
- C1: Update multiple providers
- C2: Use provider alias in subnet Terraform resources

Author: Nho Luong

Skill: DevOps Engineer Lead

Demo-5

Terraform
Input Variables
Output Values



Demo: Terraform Concept: Input Variables and Output Values

Author: Nho Luong

Skill: DevOps Engineer Lead

Terraform Input Variables

- Input variables allow us to customize the Terraform Resources or modules without altering the source code.
- **Key Benefit:** Allows us to make the code reusable by just changing variables
- We have multiple options to declare variables
  - varibales.tf (Default value)
  - terraform.tfvars
  - vm.auto.tfvars
  - vm.tfvars
  - Environment Variables
  - --var-file flag
  - --var flag

variables.tf

```
# GCP Compute Engine Machine Type
variable "machine_type" {
  description = "Compute Engine Machine Type"
  type = string
  default = "e2-small"
}
```

terraform.tfvars

terraform.tfvars

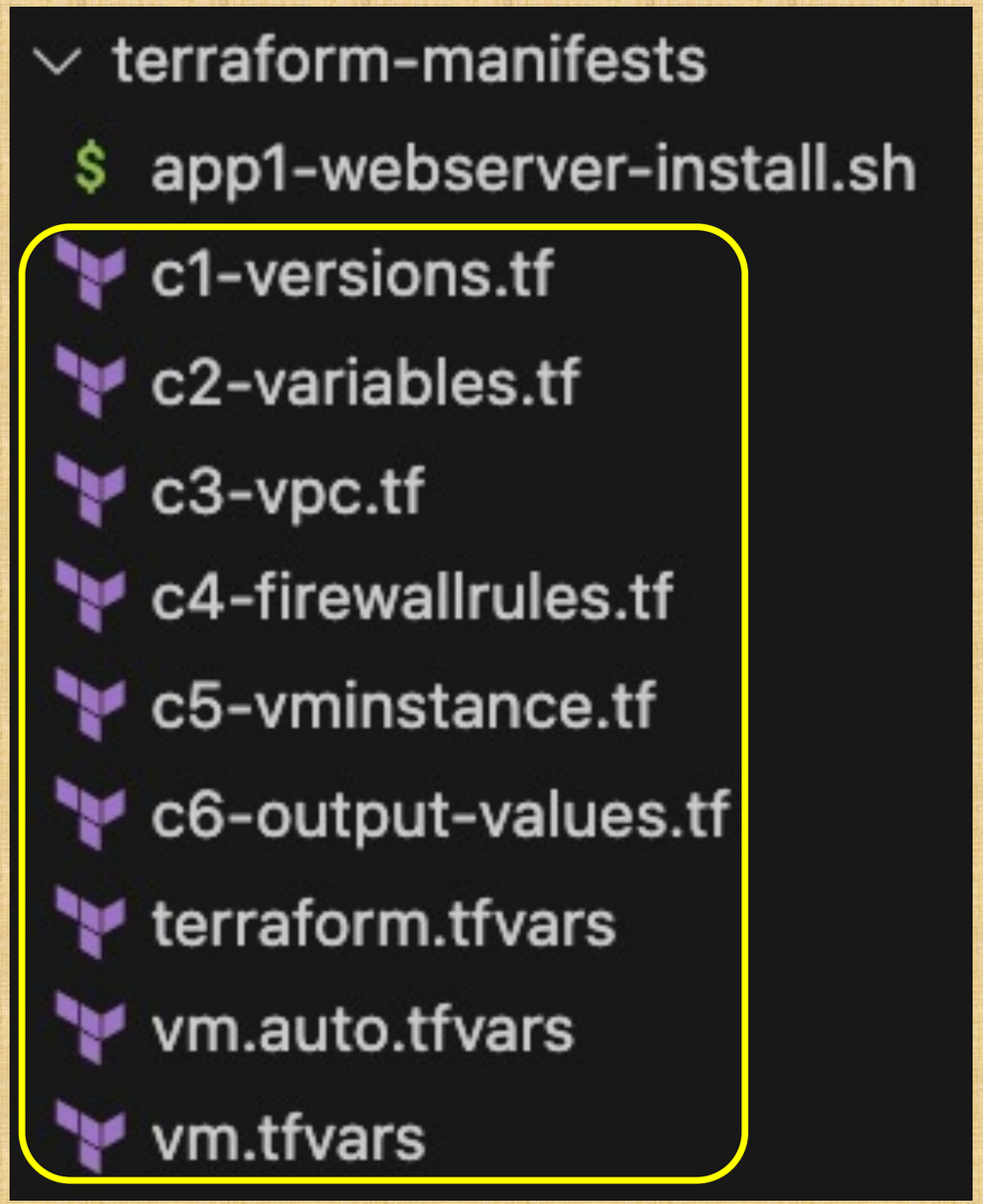
terraform-on-google-cloud > 05-Terraform-Variables-Output-Values >

```
1  gcp_project   = "gcplearn9"
2  gcp_region1   = "us-central1"
3  machine_type  = "e2-micro"
```

Author: Nho Luong

Skill: DevOps Engineer Lead

What are we going to learn?



Demo: Terraform Concept: Input Variables

C2	Understand and define Terraform Input Variables
C1	Update Provider block with Terraform input variables
C3	Update subnet region with input variable
C5	Update VM Instance machine type using input variable
TF VARS	Play with terraform.tfvars, vm.auto.tfvars and vm.tfvars

Author: Nho Luong

Skill: DevOps Engineer Lead

Terraform Output Values

- Output values are like return values in programming languages.
- Output values provide about your infrastructure available on the command line
- With output values we can expose information for other Terraform configurations to use

Output Values - Attributes

```
# Terraform Output Values
## ATTRIBUTES
output "vm_instanceid" {
  description = "VM Instance ID"
  value = google_compute_instance.myapp1.instance_id
}

output "vm_selflink" {
  description = "VM Instance Self link"
  value = google_compute_instance.myapp1.self_link
}
```

Output Values - Arguments

```
## ARGUMENTS
output "vm_name" {
  description = "VM Name"
  value = google_compute_instance.myapp1.name
}

output "vm_machine_type" {
  description = "VM Machine Type"
  value = google_compute_instance.myapp1.machine_type
}
```

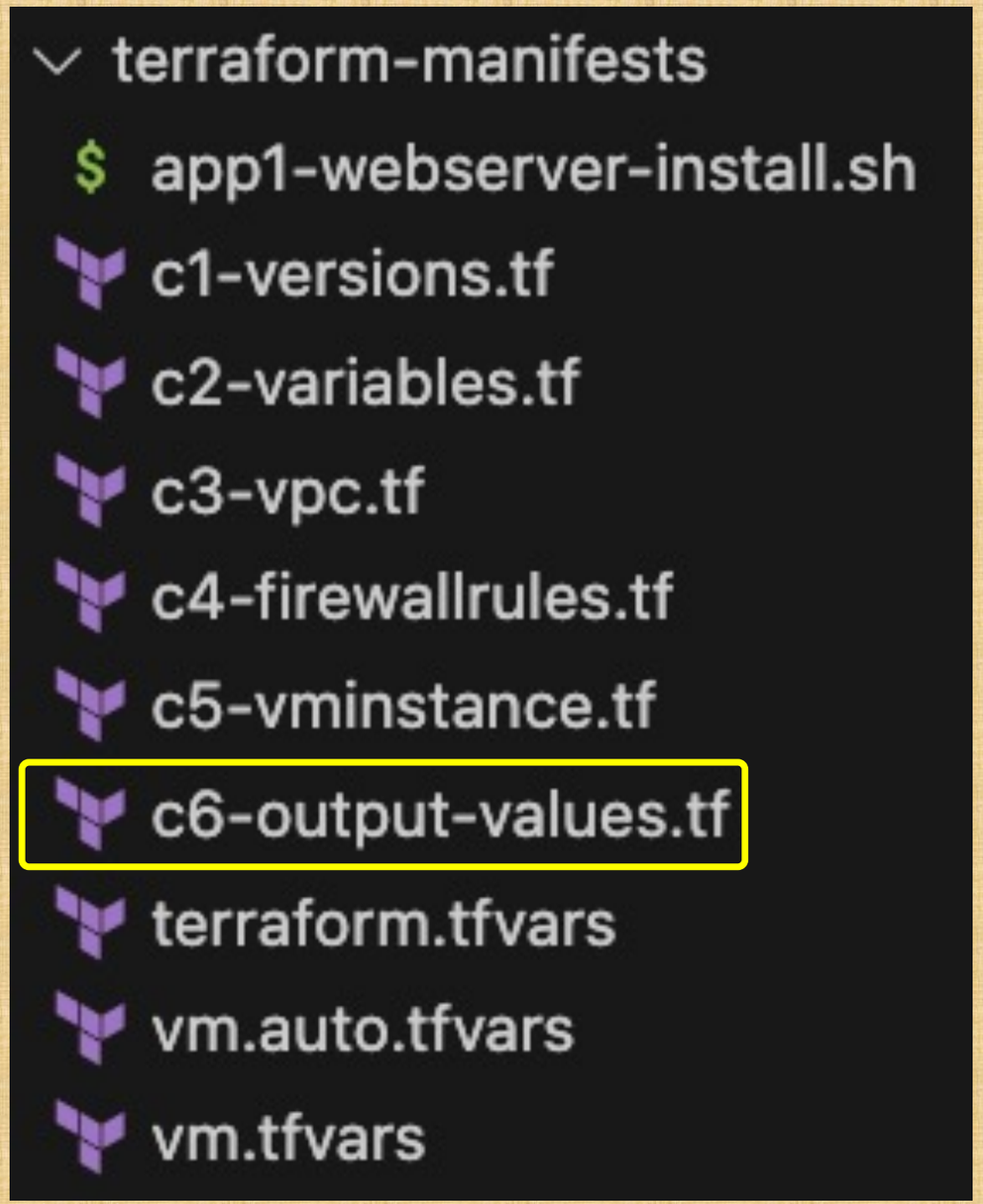
Output Values

```
Outputs:

vm_external_ip = "34.45.238.213"
vm_id = "projects/gcplearn9/zones/us-central1-a/instances/myapp1"
vm_instanceid = "3515727697506246169"
vm_machine_type = "e2-micro"
vm_name = "myapp1"
vm_selflink = "https://www.googleapis.com/compute/v1/projects/gcplearn9/zones/us-central1-a/instances/myapp1"
```

Author: Nho Luong

Skill: DevOps Engineer Lead

What are we going to learn?



Demo: Terraform Concept: Output Values

C6

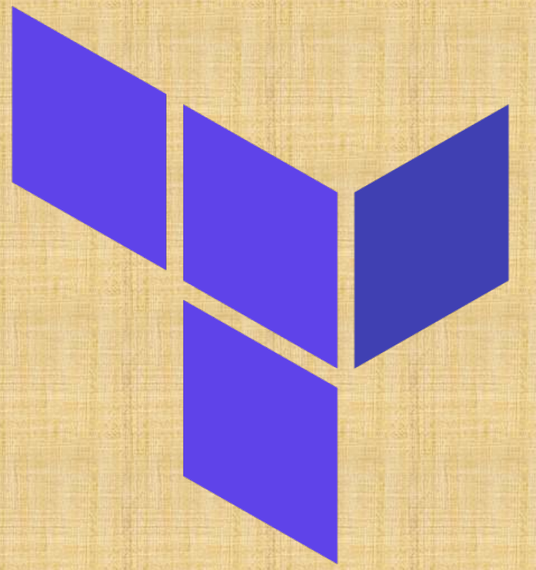
Understand and define Terraform Output values

Author: Nho Luong

Skill: DevOps Engineer Lead

Demo-6

Terraform Meta-Argument count



Demo: Terraform Concept: Meta-argument count

Author: Nho Luong

Skill: DevOps Engineer Lead

Terraform Meta-Argument: count

- By default, resource block creates a single instance of a resource
  - **For example:** 1 VM Instance
- With the meta-argument count we can create specified number of similar instances of a resource
  - **For example:** count = 2 (will create 2 VM instances from the same defined resource block)
- **count.index:** The distinct index number (starting with 0) corresponding to this instance.

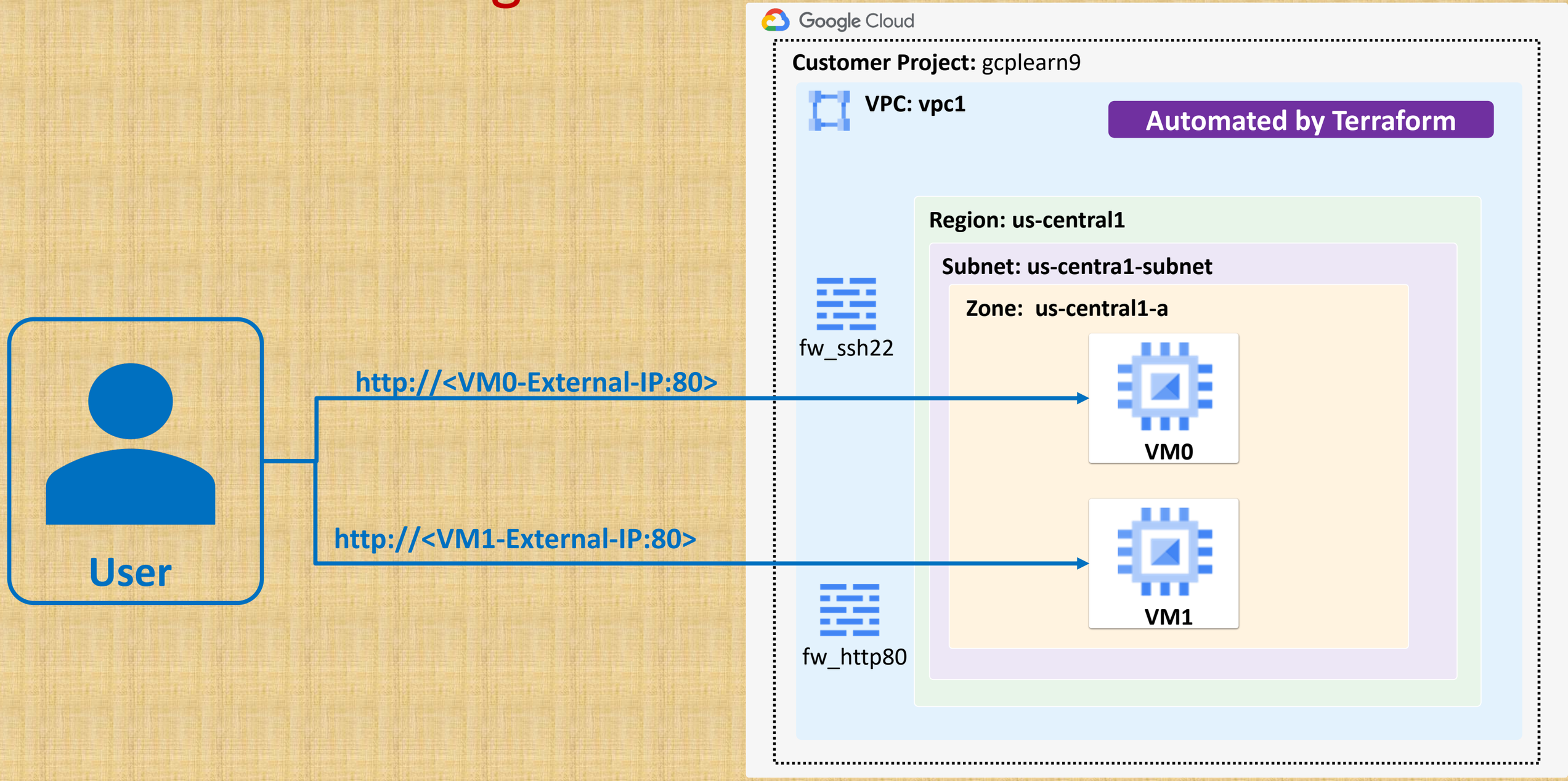
Resource Meta-Argument: count

```
# Resource Block: Create a Compute Engine instance
resource "google_compute_instance" "myapp1" {
  # Meta-Argument: count
  count = 2
  name         = "myapp1-vm-${count.index}"
  machine_type = var.machine_type
  zone         = "us-central1-a"
```

Author: Nho Luong

Skill: DevOps Engineer Lead

Terraform Meta-argument: count



Author: Nho Luong

Skill: DevOps Engineer Lead

Terraform: For expression

- **For Expression:** Enables efficient management of multiple resources by looping through lists, maps, or sets
- **Examples:**
  - For loop with list
  - For loop with map
  - For loop with map advanced
  - Legacy splat operator
  - Latest generalized splat operator

For expression

```
# Output - For Loop with List
output "for_output_list" {
  description = "For Loop with List"
  value = [for instance in google_compute_instance.myapp1: instance.name]
}

# Output - For Loop with Map
output "for_output_map1" {
  description = "For Loop with Map"
  value = {for instance in google_compute_instance.myapp1: instance.name => instance.instance_id}
}

# Output - For Loop with Map Advanced
output "for_output_map2" {
  description = "For Loop with Map - Advanced"
  value = {for c, instance in google_compute_instance.myapp1: c => instance.name}
}
```

```
# Output Legacy Splat Operator (Legacy) - Returns the List
output "legacy_splat_instance" {
  description = "Legacy Splat Operator"
  value = google_compute_instance.myapp1.*.name
}

# Output Latest Generalized Splat Operator - Returns the List
output "latest_splat_instance" {
  description = "Generalized latest Splat Operator"
  value = google_compute_instance.myapp1[*].name
}
```

Author: Nho Luong

Skill: DevOps Engineer Lead

What are we going to learn?

Demo: Terraform Concept: Meta-argument count

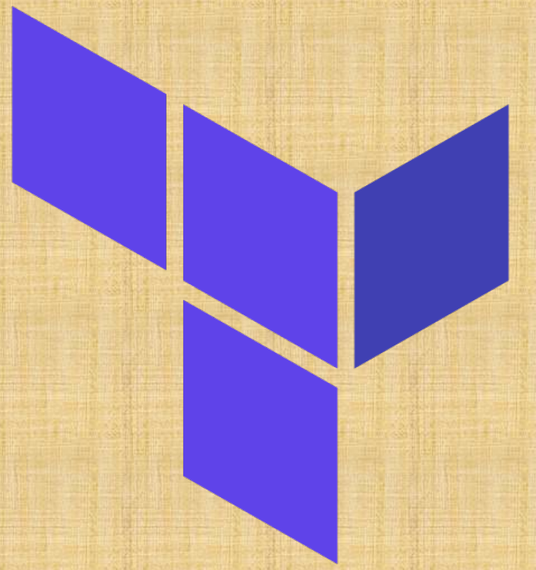
NC	No Changes from C1 to C4
C5	Define VM Instance with Meta-argument count
C6	Define Terraform output values with Terraform for expression and also learn about Splat operator

Author: Nho Luong

Skill: DevOps Engineer Lead

Demo-7

Terraform
Data sources



Demo: Terraform Concept: Datasources

Author: Nho Luong

Skill: DevOps Engineer Lead

Terraform Data sources

- **Data sources:** allow Terraform to use information defined outside of Terraform
- A data source is accessed via a special kind of resource known as a data resource, declared using a data block:
- **Use case:**
  - **Step-01:** Get the available compute zones in that respective region using google_compute_zone datasource
  - **Step-02:** Create two VM Instances in two available compute zones using the Meta-argument count

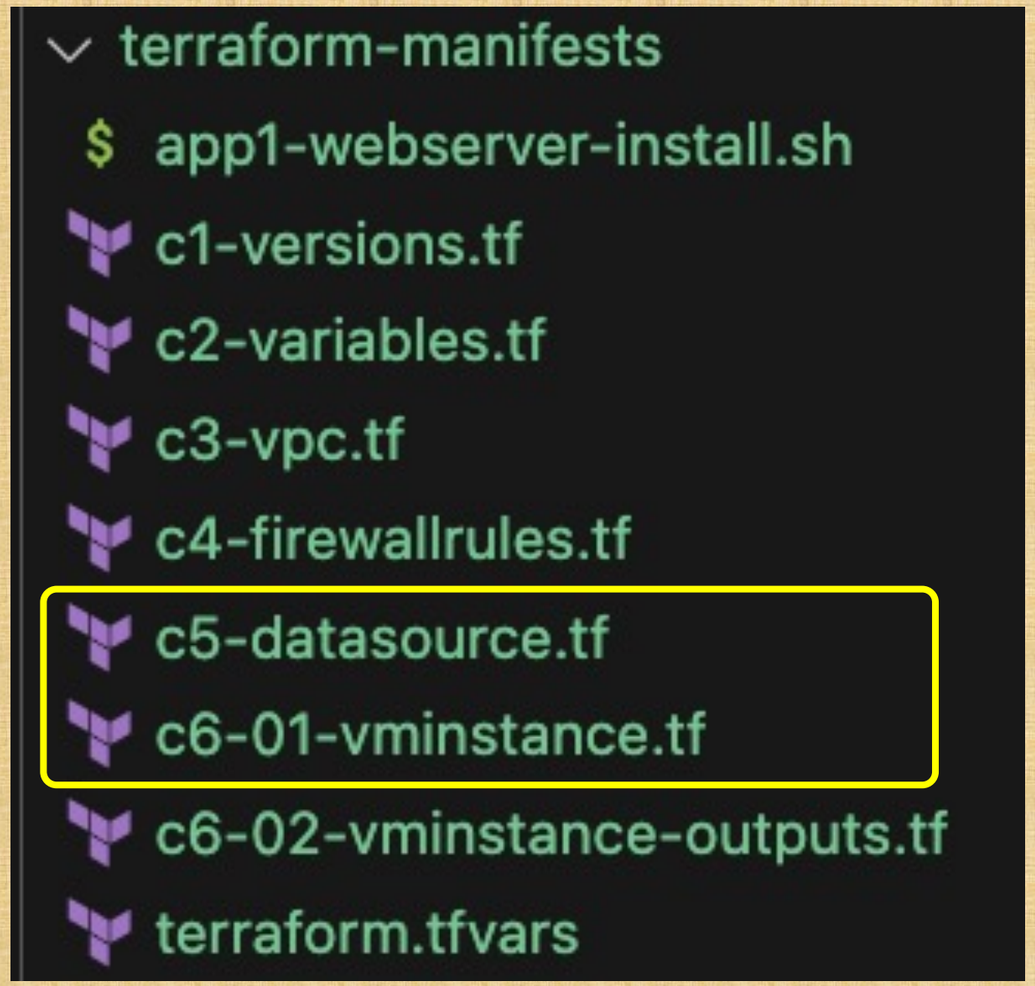
Terraform Data source

```
# Terraform Datasources
/* Datasource: Get a list of Google
Compute zones that are UP in a region */
data "google_compute_zones" "available" {
  status = "UP"
}

# Output value
output "compute_zones" {
  description = "List of compute zones"
  value = data.google_compute_zones.available.names
}
```

Author: Nho Luong

Skill: DevOps Engineer Lead

What are we going to learn?



Demo: Terraform Concept: Datasources

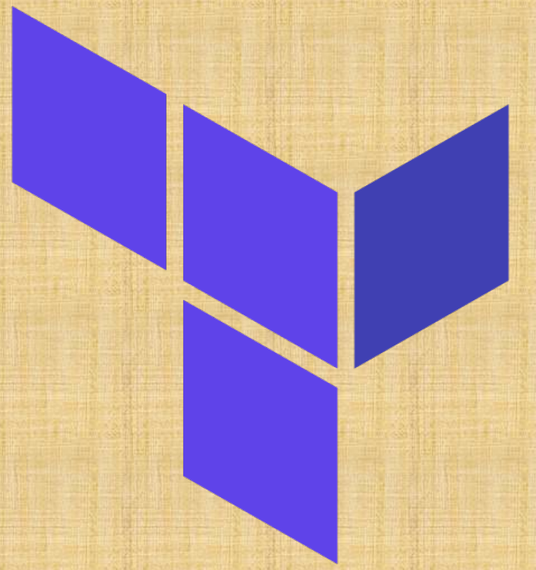
NC	No Changes from C1 to C4
C5	Define a data source to get a list of all available zones in that region
C6-01	Update VM Instance with zone value updated from Terraform data source

Author: Nho Luong

Skill: DevOps Engineer Lead

Demo-8

Terraform Meta-Argument for_each



Demo: Terraform Concept: Meta-Argument for_each

Author: Nho Luong

Skill: DevOps Engineer Lead

Terraform Meta-Argument: for_each

- By default, resource block creates a single instance of a resource
  - **For example:** 1 VM Instance
- The for_each meta-argument accepts a map or a set of strings and creates an instance for each item in that map or set
  - **For example:** Creates 1 VM instance for each compute zone
- **Each instance has a**
  - distinct infrastructure object associated with it
  - each is separately created, updated, or destroyed when the configuration is applied

Resource Meta-Argument: for_each

```
# Resource Block: Create a Compute Engine instance
resource "google_compute_instance" "myapp1" {
  # Meta-Argument: for_each
  for_each = toset(data.google_compute_zones.available.names)
  name         = "myapp1-vm-${each.key}"
  machine_type = var.machine_type
  zone         = each.key # You can also use each.value because for list
  tags         = [tolist(google_compute_firewall.fw_ssh.target_tags)[0],
```

- **each.key:** The map key (or set member) corresponding to this instance.
- **each.value:** The map value corresponding to this instance (If a set was provided, each.key == each.value)

Author: Nho Luong

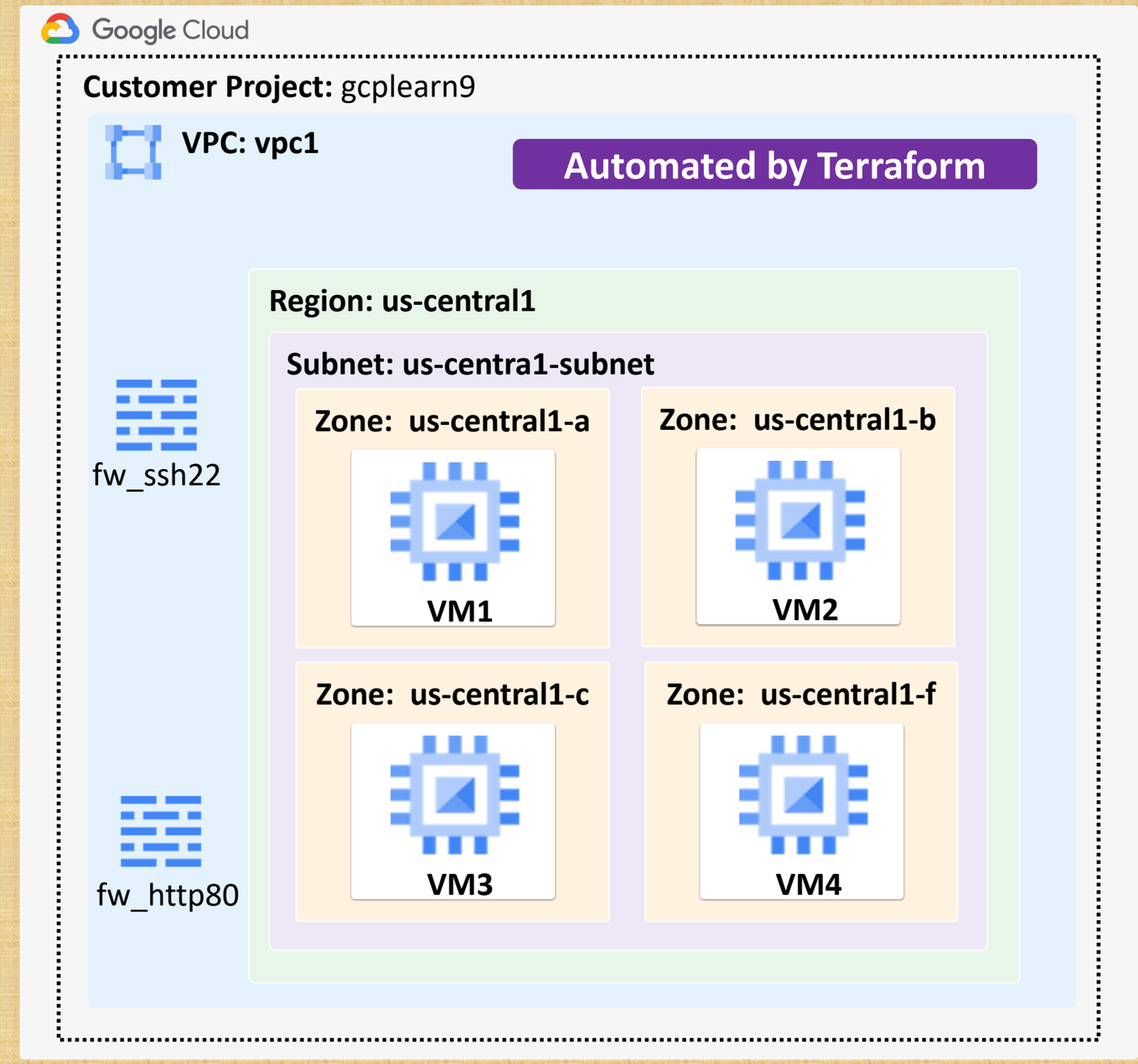
Skill: DevOps Engineer Lead

Terraform Meta-argument: for_each

Demo-1: for_each with Set values

Creates VM instance for each available compute zone in a region

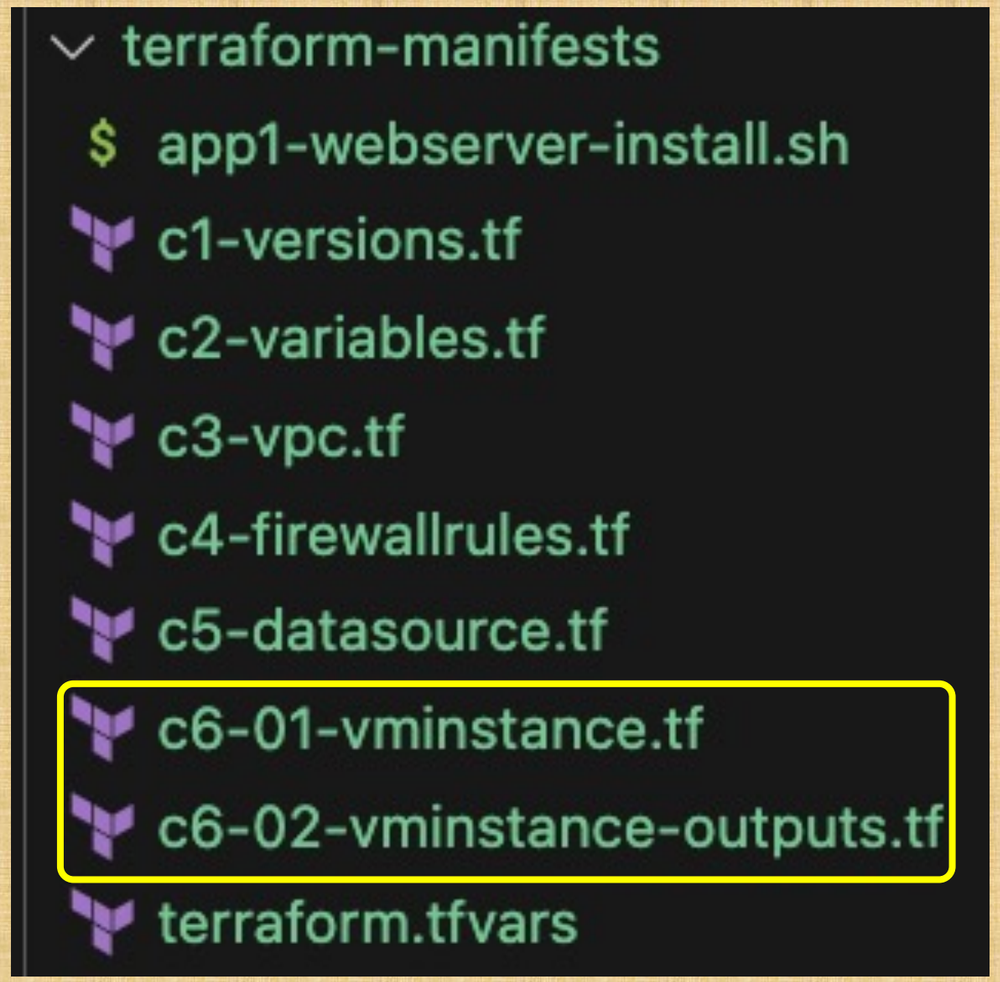
```
# Resource Block: Create a Compute Engine instance
resource "google_compute_instance" "myapp1" {
  # Meta-Argument: for_each
  for_each = toset(data.google_compute_zones.available.names)
  name         = "myapp1-vm-${each.key}"
  machine_type = var.machine_type
  zone         = each.key # You can also use each.value because
```



Author: Nho Luong

Skill: DevOps Engineer Lead

What are we going to learn?



Demo-01: Terraform Concept: Meta-argument for_each

NC	No Changes from C1 to C5
C6-01	VM Instance with for_each Meta-argument with Set Values
C6-02	Terraform outputs for VM Instance

Author: Nho Luong

Skill: DevOps Engineer Lead

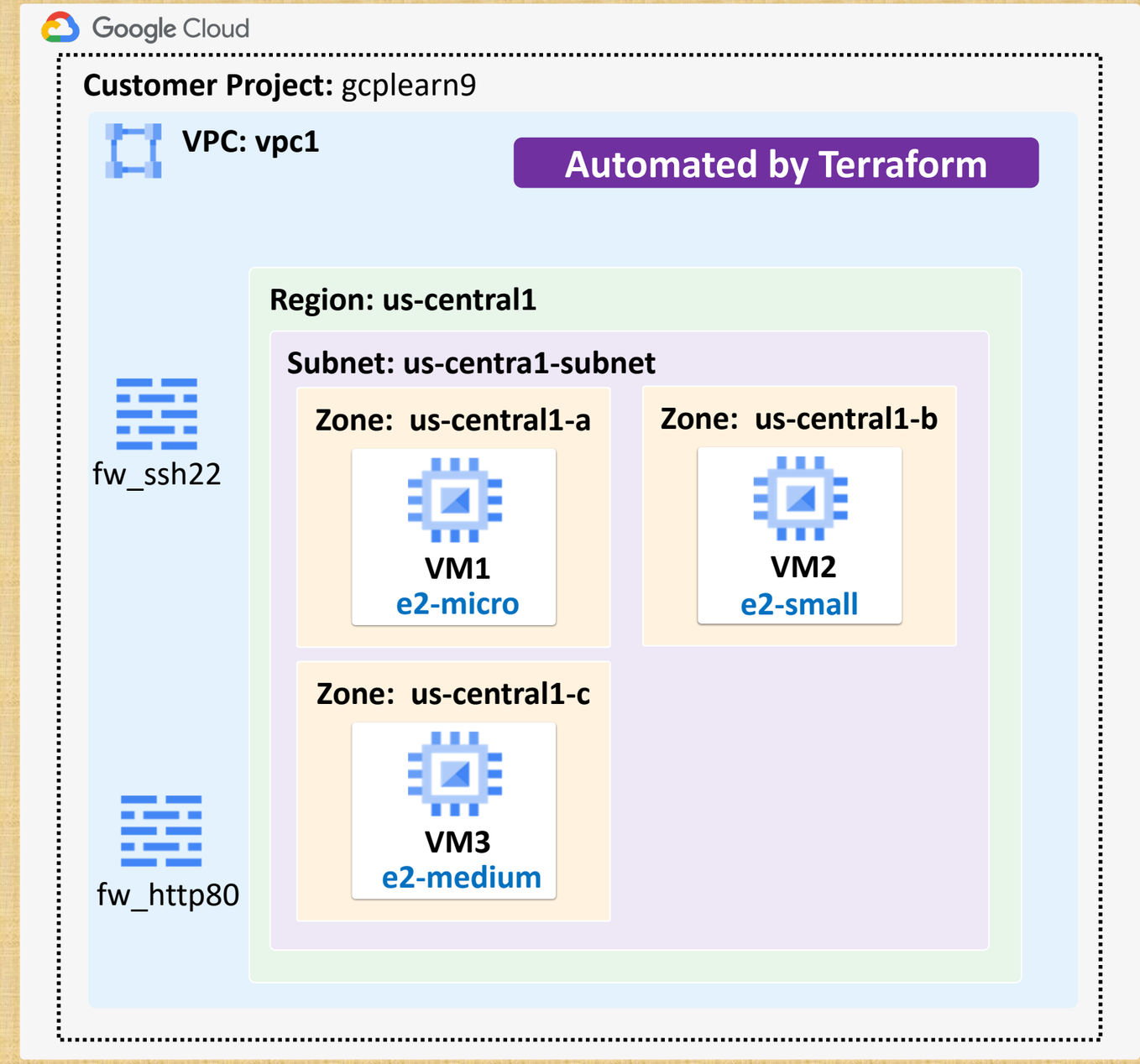
Terraform Meta-argument: for_each

Demo-2: for_each with Map values

Creates VM instance with different machine type in each compute zone

```hcl
# Define a map with zone as key and machine_type as value
variable "zone_machine_map" {
  type = map(string)
  default = {
    "us-central1-a" = "e2-micro"
    "us-central1-b" = "e2-small"
    "us-central1-c" = "e2-medium"
  }
}
```

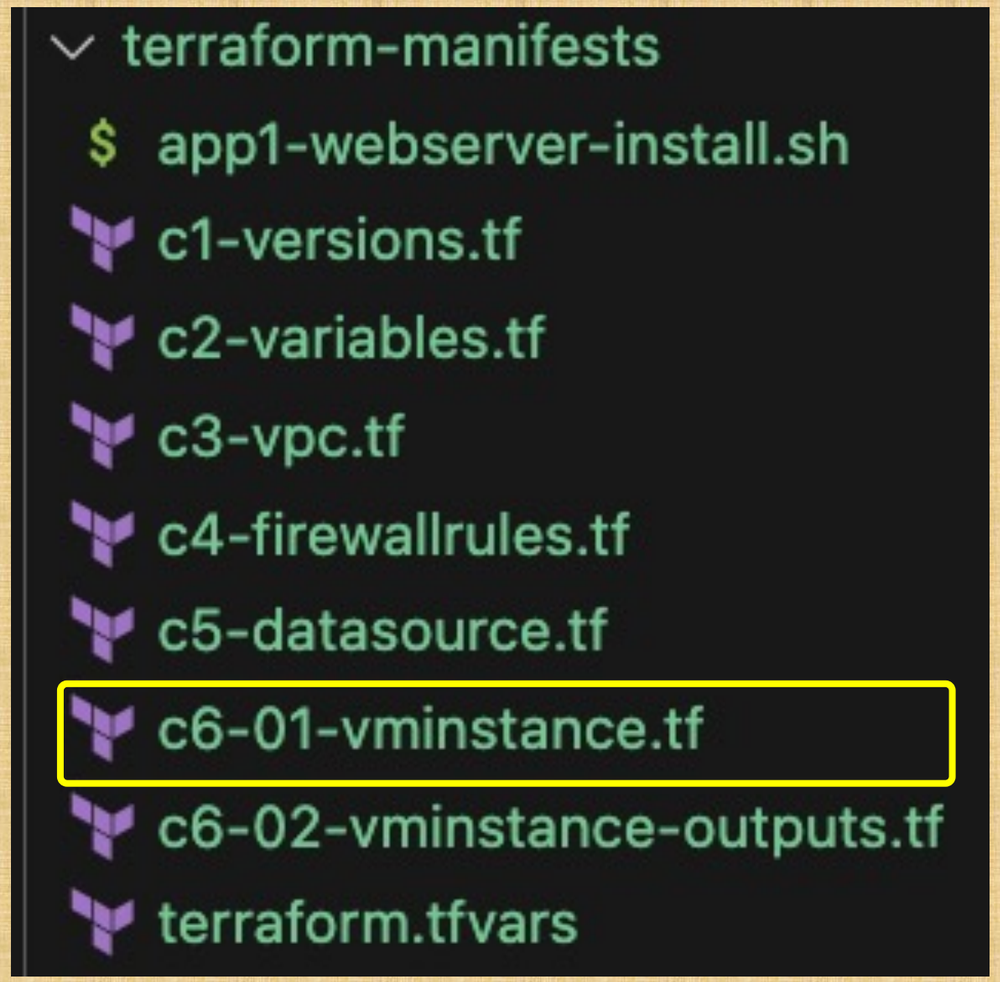
```hcl
# Resource Block: Create a Compute Engine instance
resource "google_compute_instance" "myapp1" {
  # Meta-Argument: for_each
  for_each = var.zone_machine_map
  name         = "myapp1-vm-${each.key}"
  machine_type = each.value
  zone         = each.key
```



Author: Nho Luong

Skill: DevOps Engineer Lead

What are we going to learn?



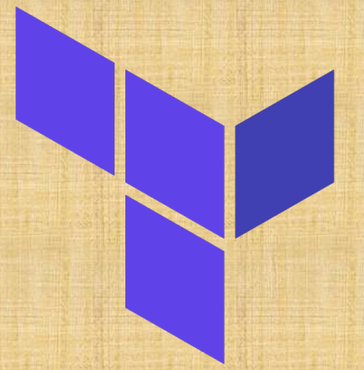
Demo-02: Terraform Concept: Meta-argument for_each

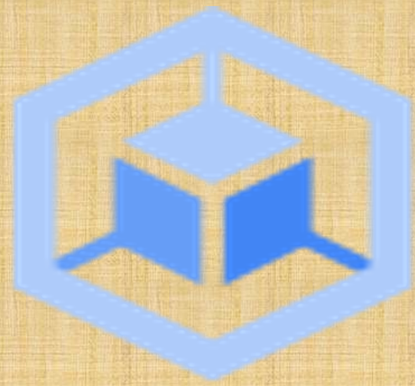
NC	No Changes from C1 to C5
C6-01	VM Instance with for_each Meta-argument with Map Values

Author: Nho Luong

Skill: DevOps Engineer Lead

Demo-9





Google Kubernetes Engine

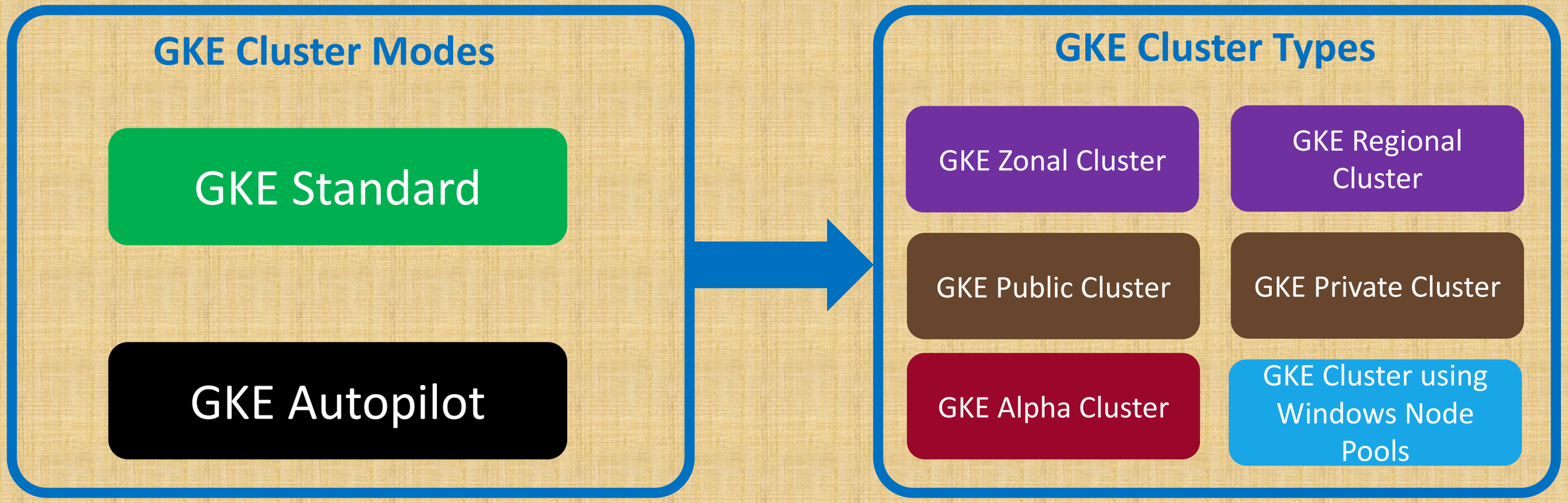
Create GKE Standard Public Cluster

Terraform New Concept: Terraform Remote Backend

Author: Nho Luong

Skill: DevOps Engineer Lead

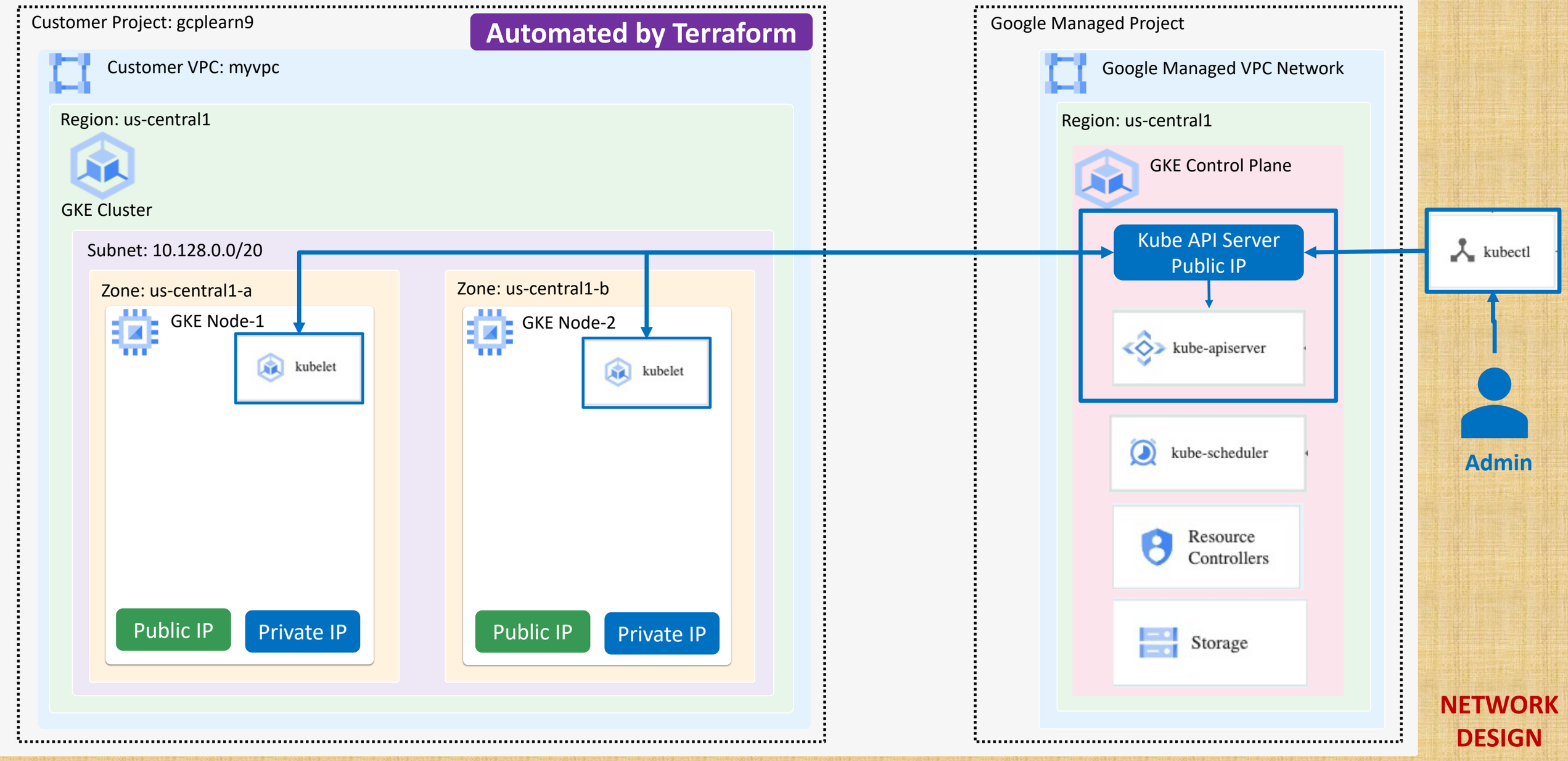
GKE Cluster Modes & Types



Author: Nho Luong

Skill: DevOps Engineer Lead

GKE Standard Public Cluster - Network Design

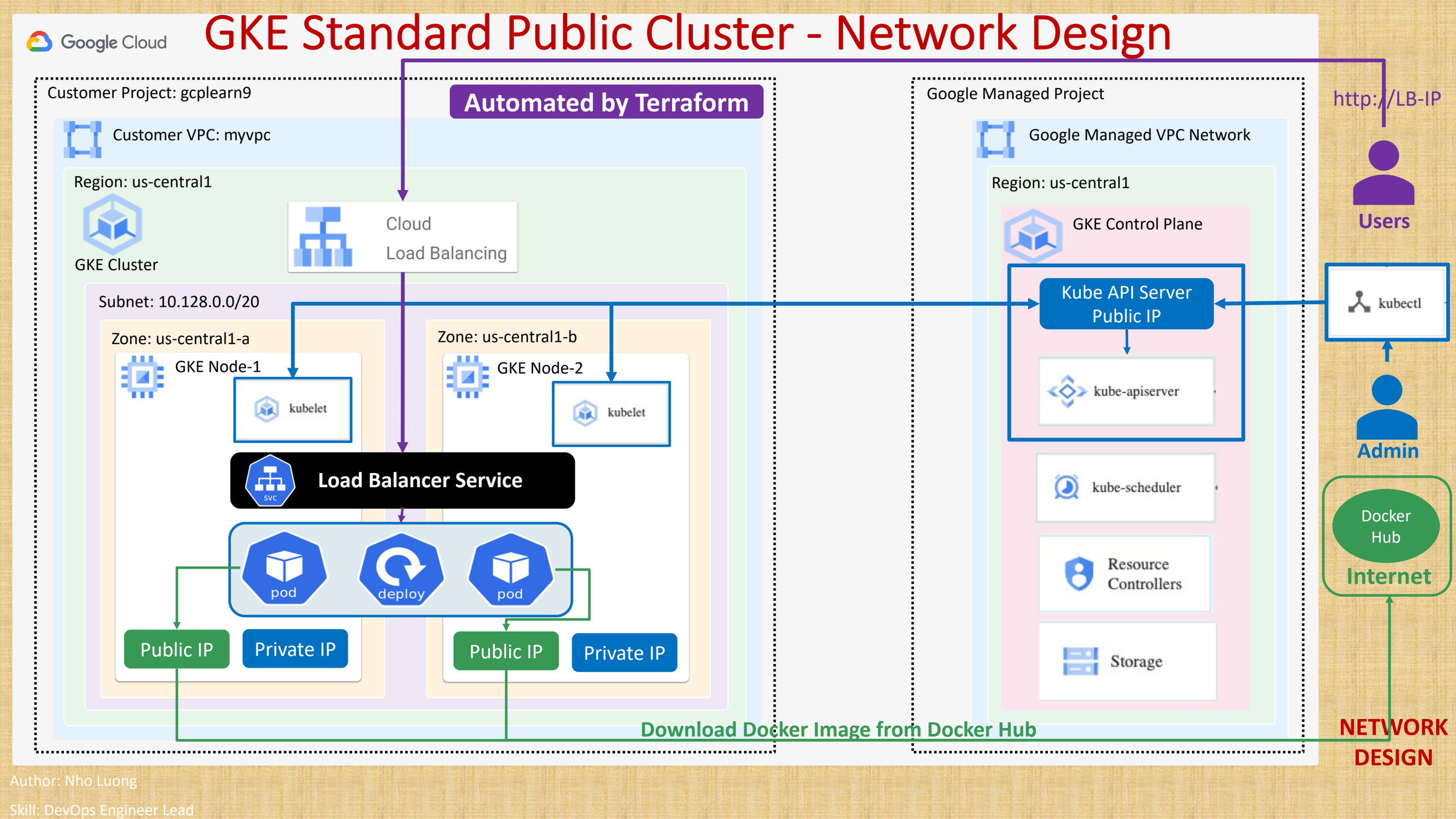


NETWORK DESIGN

Author: Nho Luong

Skill: DevOps Engineer Lead

GKE Standard Public Cluster - Network Design



Author: Nho Luong

Skill: DevOps Engineer Lead

What is Terraform Backend ?

Backends are responsible for storing state and providing an API for state locking.

Local Backend

Terraform
Local
State
Storage

Inside Terraform Local Working Directory

Remote Backend

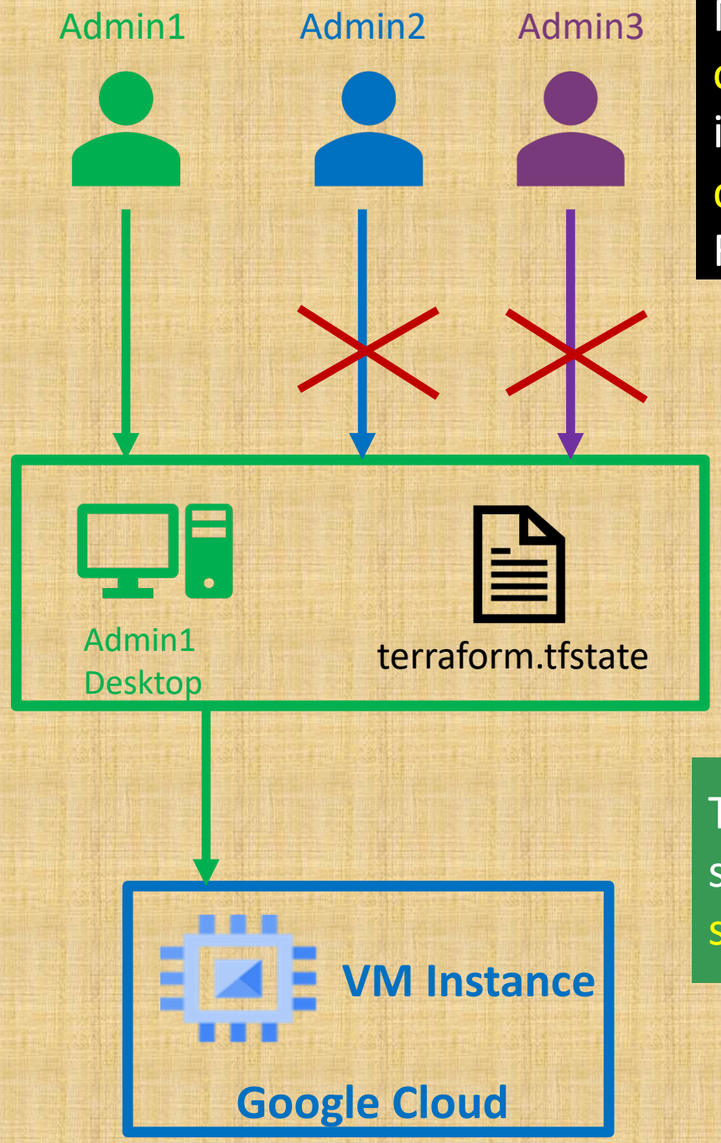
Terraform
Remote
State
Storage

Google Cloud Storage Buckets

Author: Nho Luong

Skill: DevOps Engineer Lead

Local State File



Multiple Team members cannot update the infrastructure as they don't have access to State File

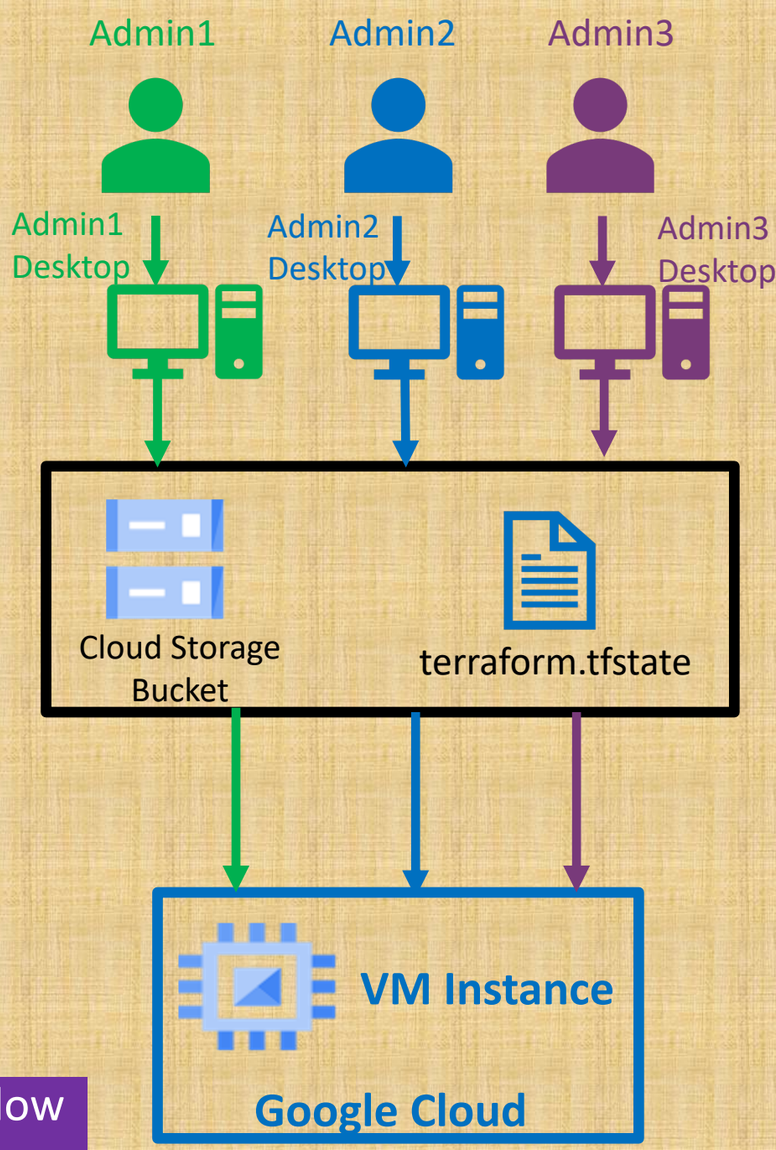
This means we need store the state file in a shared location.

Remote State File

Using Terraform Backend concept we can use GCP Cloud storage bucket as the shared storage for State Files

If two team members are running Terraform at the same time, you may run into race conditions as multiple Terraform processes make concurrent updates to the state files, leading to conflicts, data loss, and state file corruption.

State Locking will not allow the above scenario

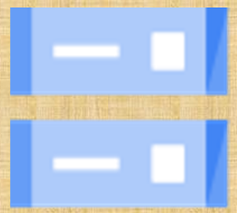


Author: Nho Luong

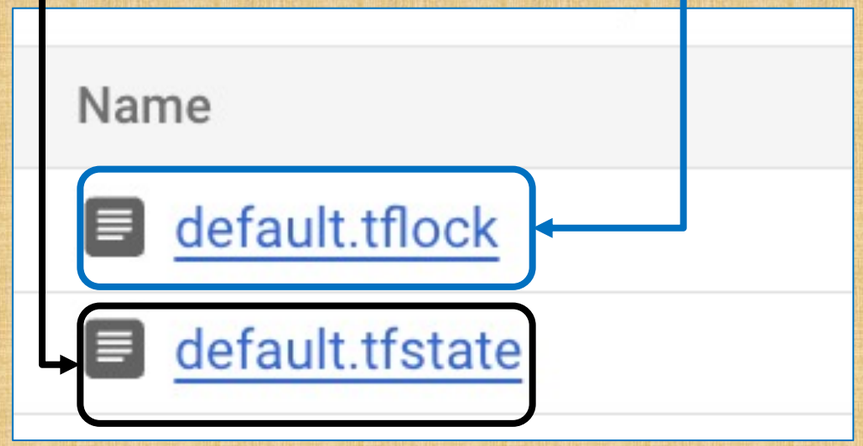
Skill: DevOps Engineer Lead

What is Terraform State Storage and State Locking ?

Terraform State Storage



GCP Cloud Storage Bucket



Terraform State Locking

GCP Cloud Storage Bucket

Author: Nho Luong

Skill: DevOps Engineer Lead

Terraform Remote Backend for GCP Terraform Resources

Terraform Remote Backend as GCP Cloud Storage Bucket

```
# Terraform Settings Block
terraform {
  required_version = ">= 1.9"
  required_providers {
    google = {
      source = "hashicorp/google"
      version = ">= 5.38.0"
    }
  }
  backend "gcs" {
    bucket = "terraform-on-gcp-gke"
    prefix = "dev/gke-cluster-public"
  }
}
```

Author: Nho Luong

Skill: DevOps Engineer Lead

Project-1: GKE Cluster

What are we going to learn?

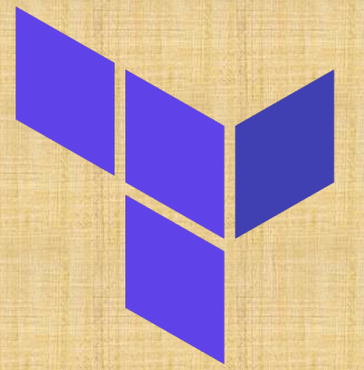
Demo: GKE Standard Public Cluster + Terraform Remote Backend

NC	No Changes from C2-01, C2-02, C4
C1	Add Remote Backend as Cloud Storage Bucket
C3	Standard VPC and Subnet, no major changes
C5-01	IAM Service Account
C5-02	GKE Cluster
C5-03	GKE Linux Node Pool
C5-04	GKE Outputs

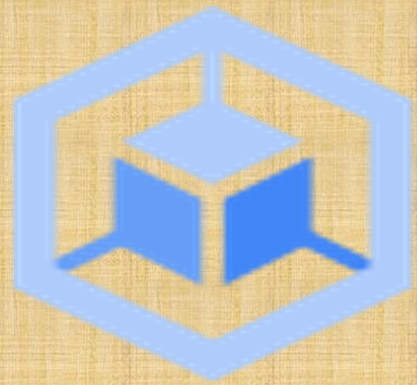
Author: Nho Luong

Skill: DevOps Engineer Lead

Demo-10



Google Kubernetes Engine



Kubernetes Deployment

Kubernetes Load Balancer Service

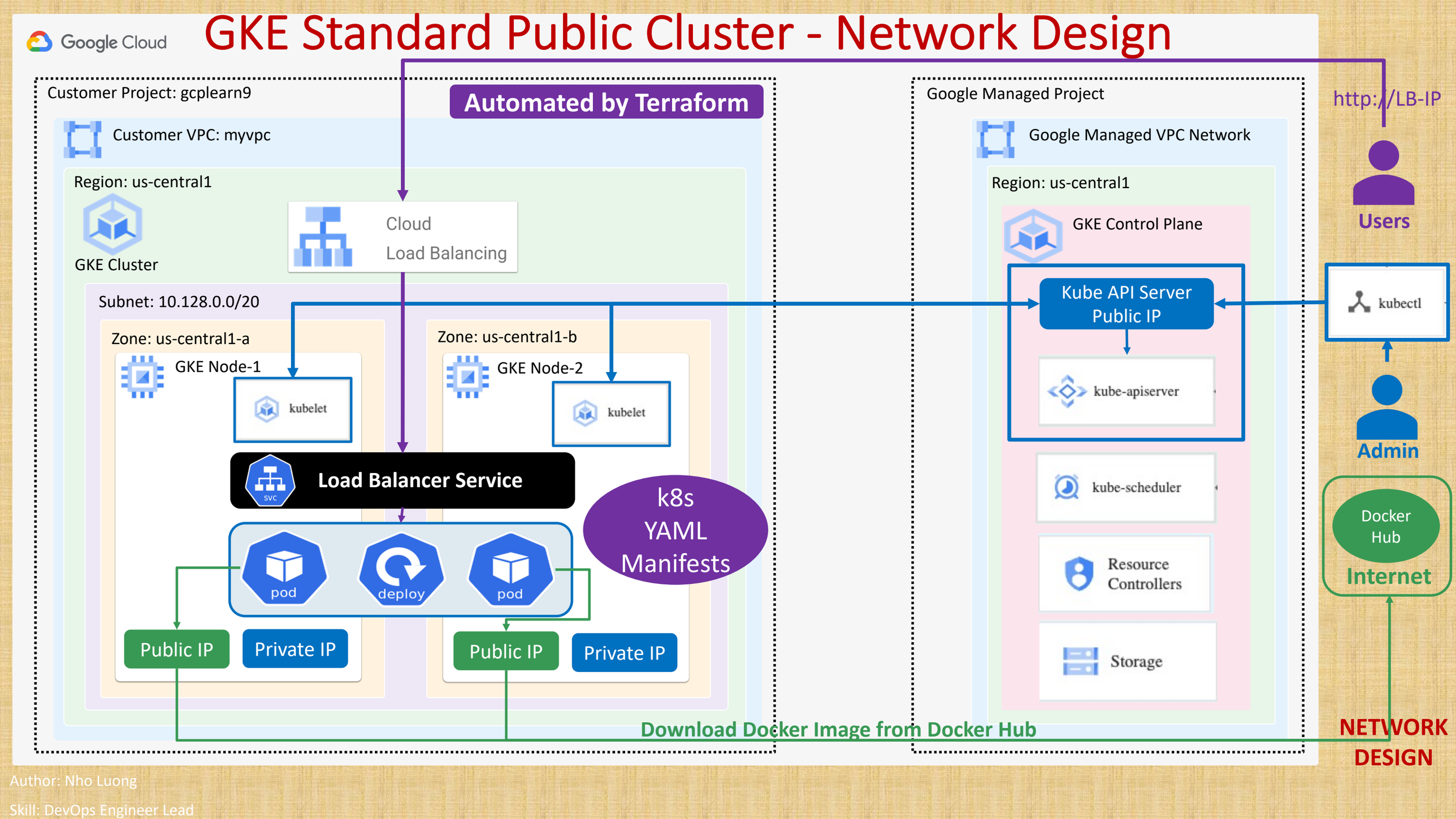
YAML Manifests

Demo: Kubernetes YAML Manifests

Author: Nho Luong

Skill: DevOps Engineer Lead

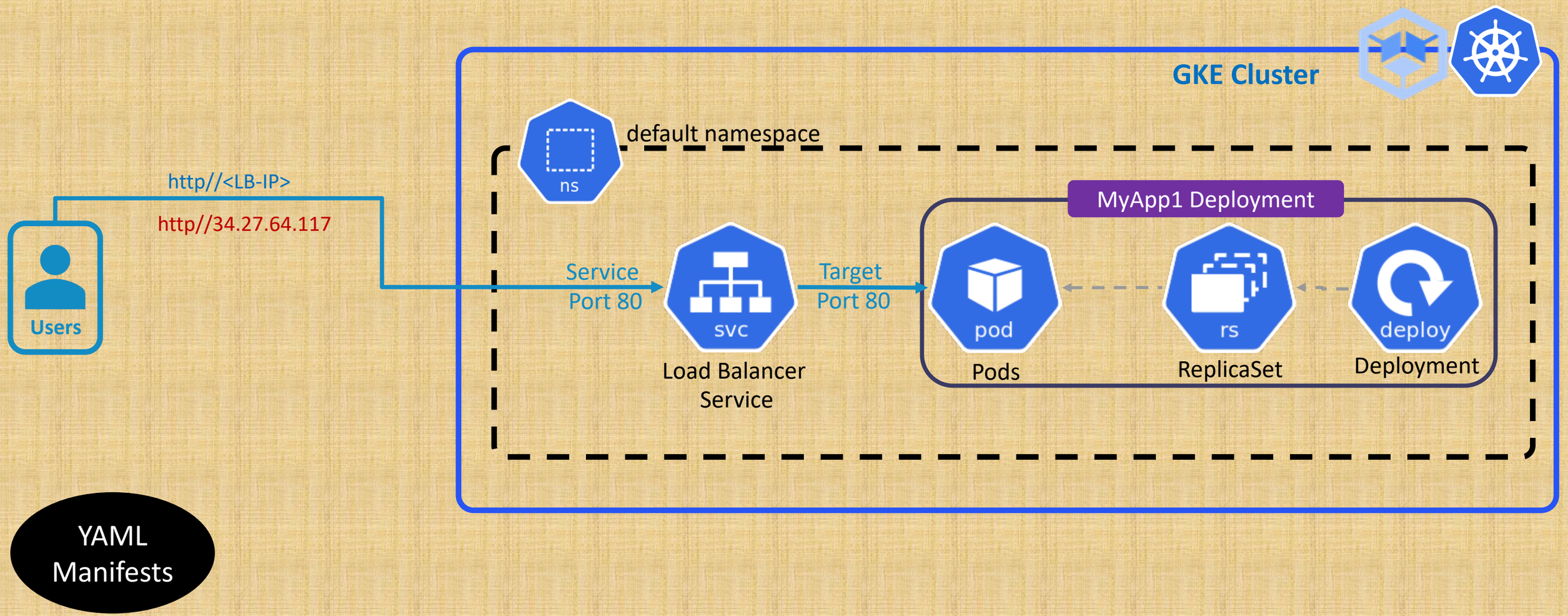
GKE Standard Public Cluster - Network Design



Author: Nho Luong

Skill: DevOps Engineer Lead

GKE- Kubernetes Load Balancer Service



YAML Manifests

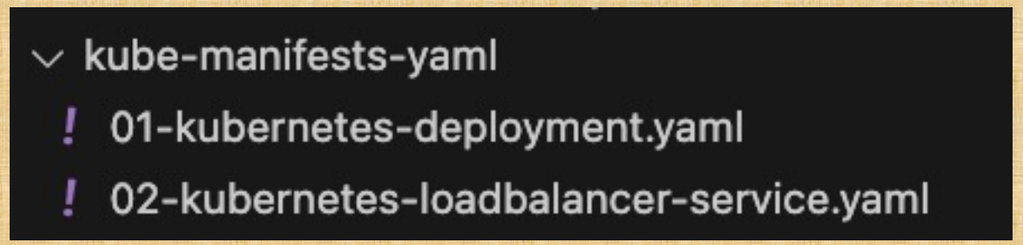
Author: Nho Luong

Skill: DevOps Engineer Lead

Project-2: k8s YAML Manifests

What are we going to learn?

Demo: Deploy Kubernetes Deployment and Service in GKE Standard Public Cluster



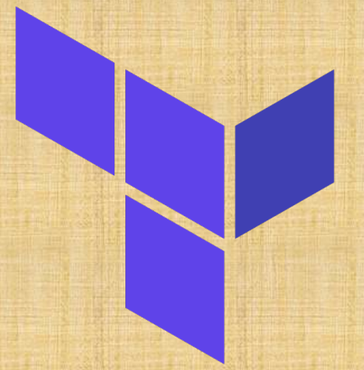
01 Kubernetes Deployment

02 Kubernetes Load Balancer Service

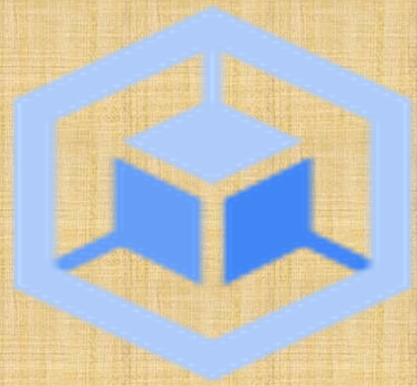
Author: Nho Luong

Skill: DevOps Engineer Lead

Demo-11



Google Kubernetes Engine



Kubernetes Deployment

Kubernetes Load Balancer Service

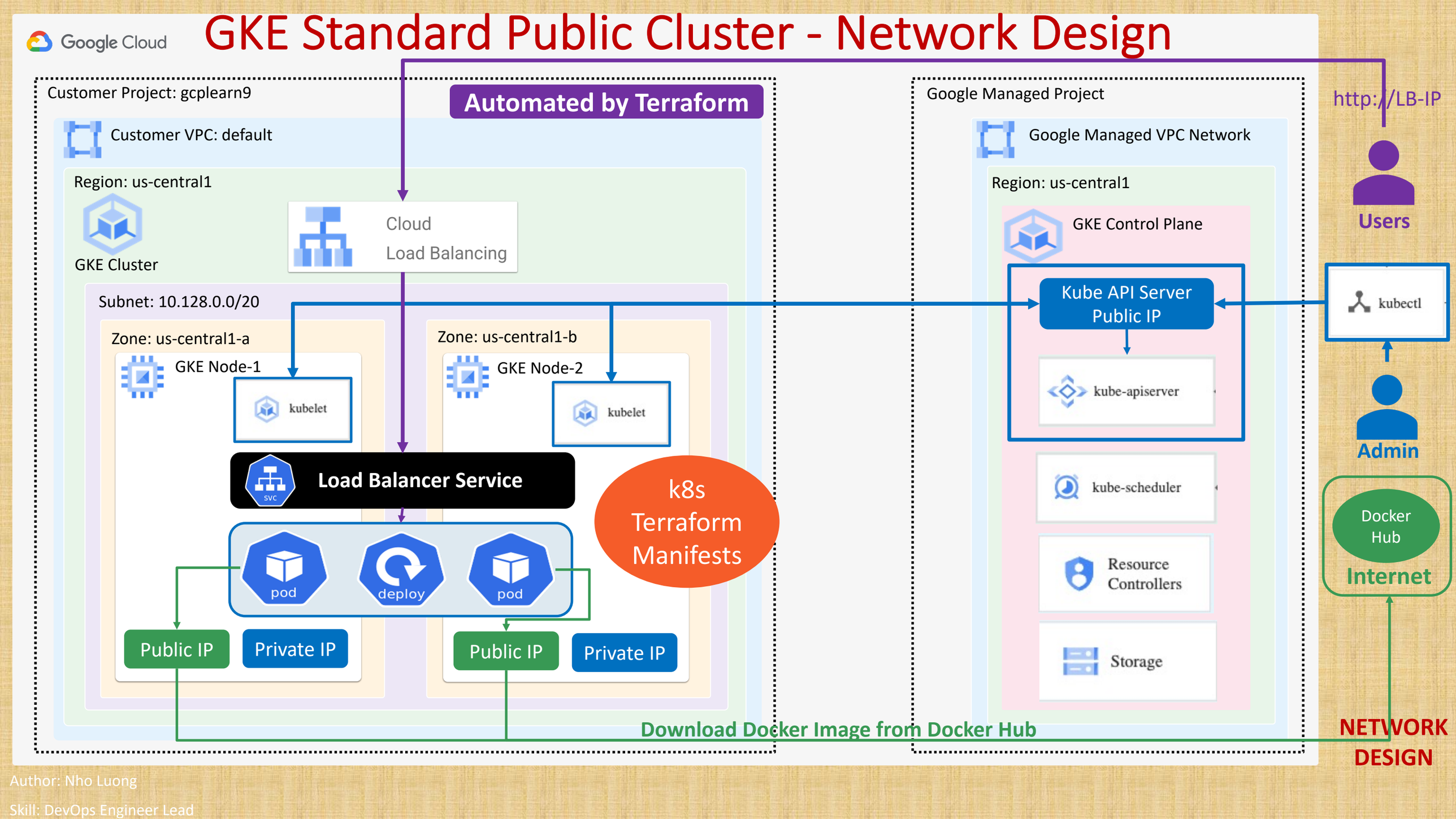
k8s
Terraform
Manifests

Demo: Kubernetes Deployment and Service using Terraform + Terraform Remote State Data source concept

Author: Nho Luong

Skill: DevOps Engineer Lead

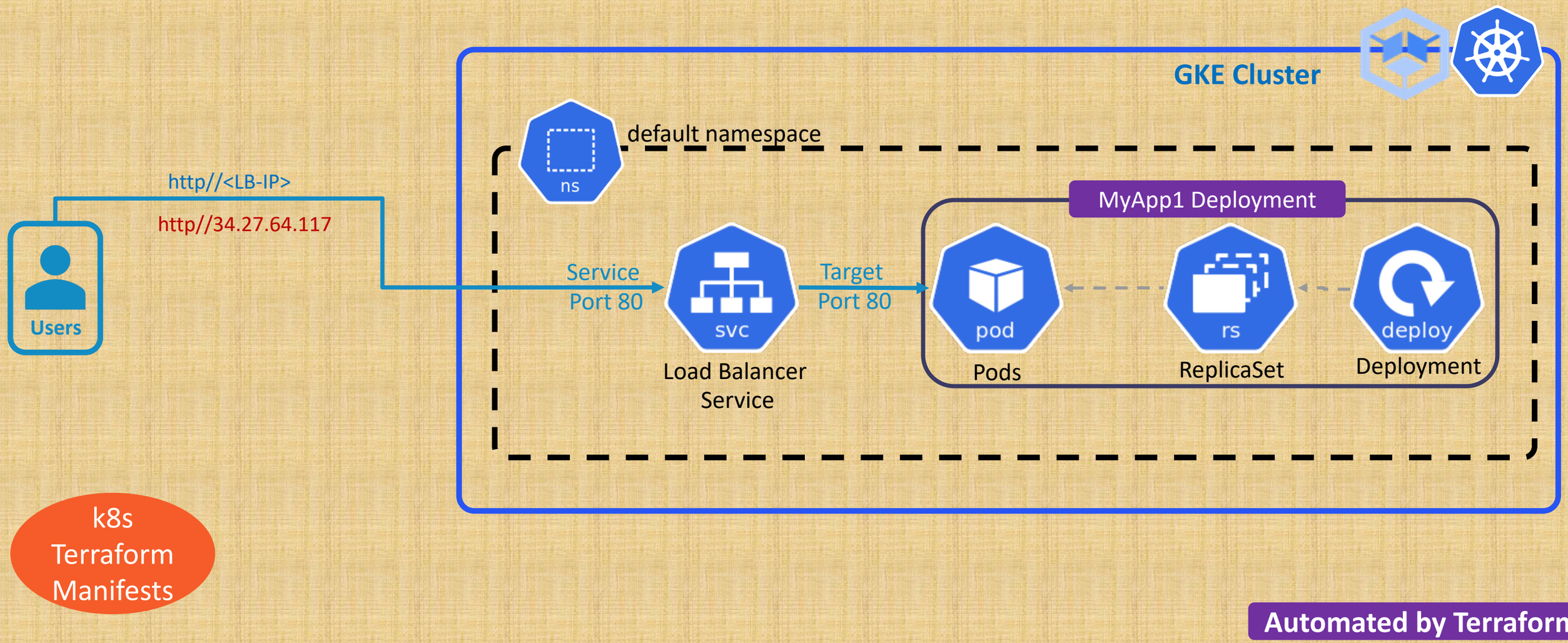
GKE Standard Public Cluster - Network Design



Author: Nho Luong

Skill: DevOps Engineer Lead

GKE- Kubernetes Load Balancer Service



k8s
Terraform
Manifests

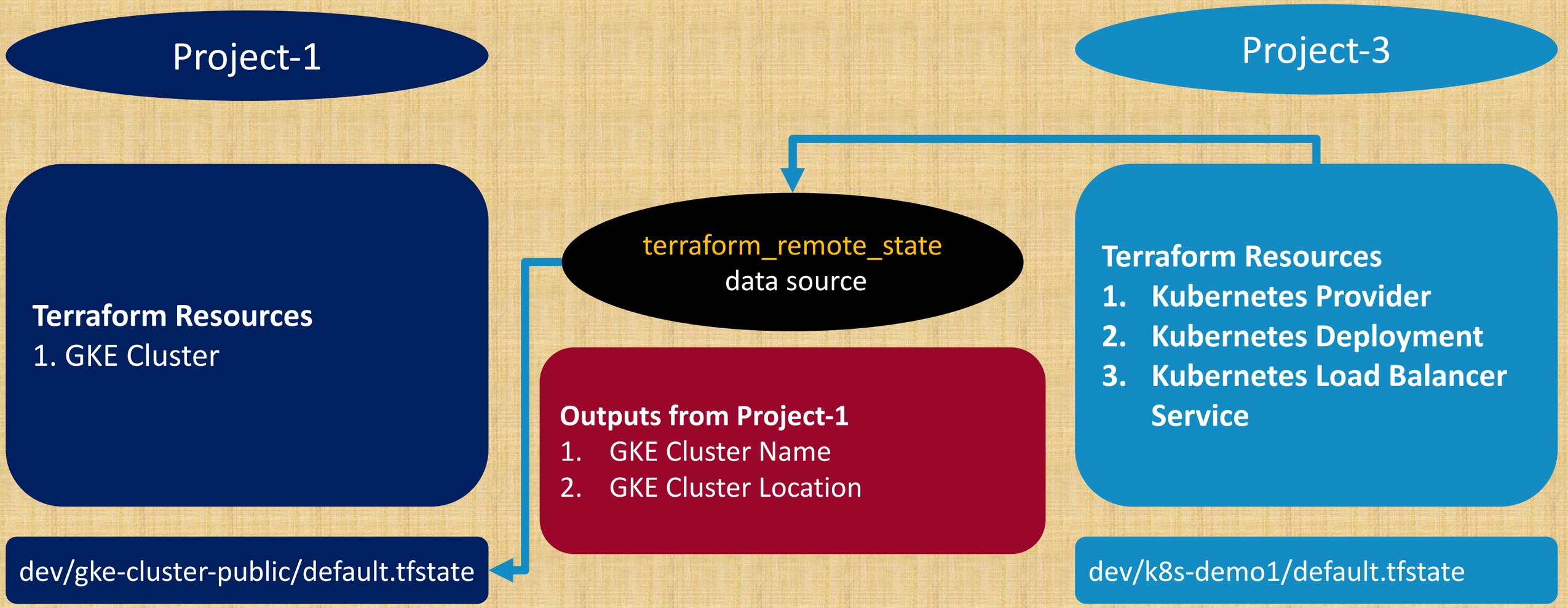
Automated by Terraform

Author: Nho Luong

Skill: DevOps Engineer Lead

Terraform Remote State Datasource

The terraform_remote_state data source retrieves the root module output values from project-1 Terraform configuration, using the latest state snapshot from the remote backend.



Author: Nho Luong

Skill: DevOps Engineer Lead

Project-3: k8s Terraform Manifests

What are we going to learn?

Demo: Kubernetes Deployment and Service using Terraform + Terraform Remote State Data source concept

C1	Add Remote Backend as Cloud Storage Bucket
C2-01	Standard Input Variables as project-1
C2-02	Standard Local values as project-1
C3-01	Terraform Remote State Data source concept to access outputs from Project-1
C3-02	Kubernetes Provider to connect to GKE Cluster created using Project-1
C4	Kubernetes Deployment using Terraform
C5	Kubernetes Load Balancer Service using Terraform

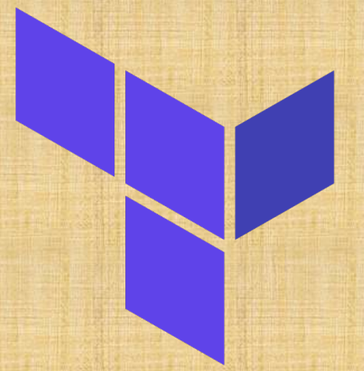
Author: Nho Luong

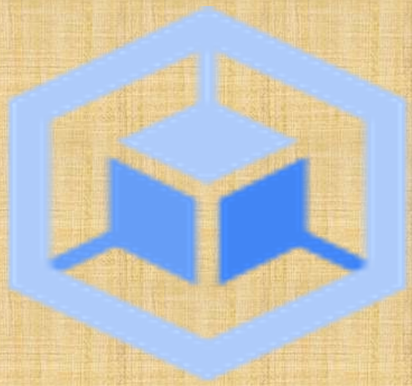
Skill: DevOps Engineer Lead

Demo-12

Google Kubernetes Engine

Create GKE Standard Private Cluster





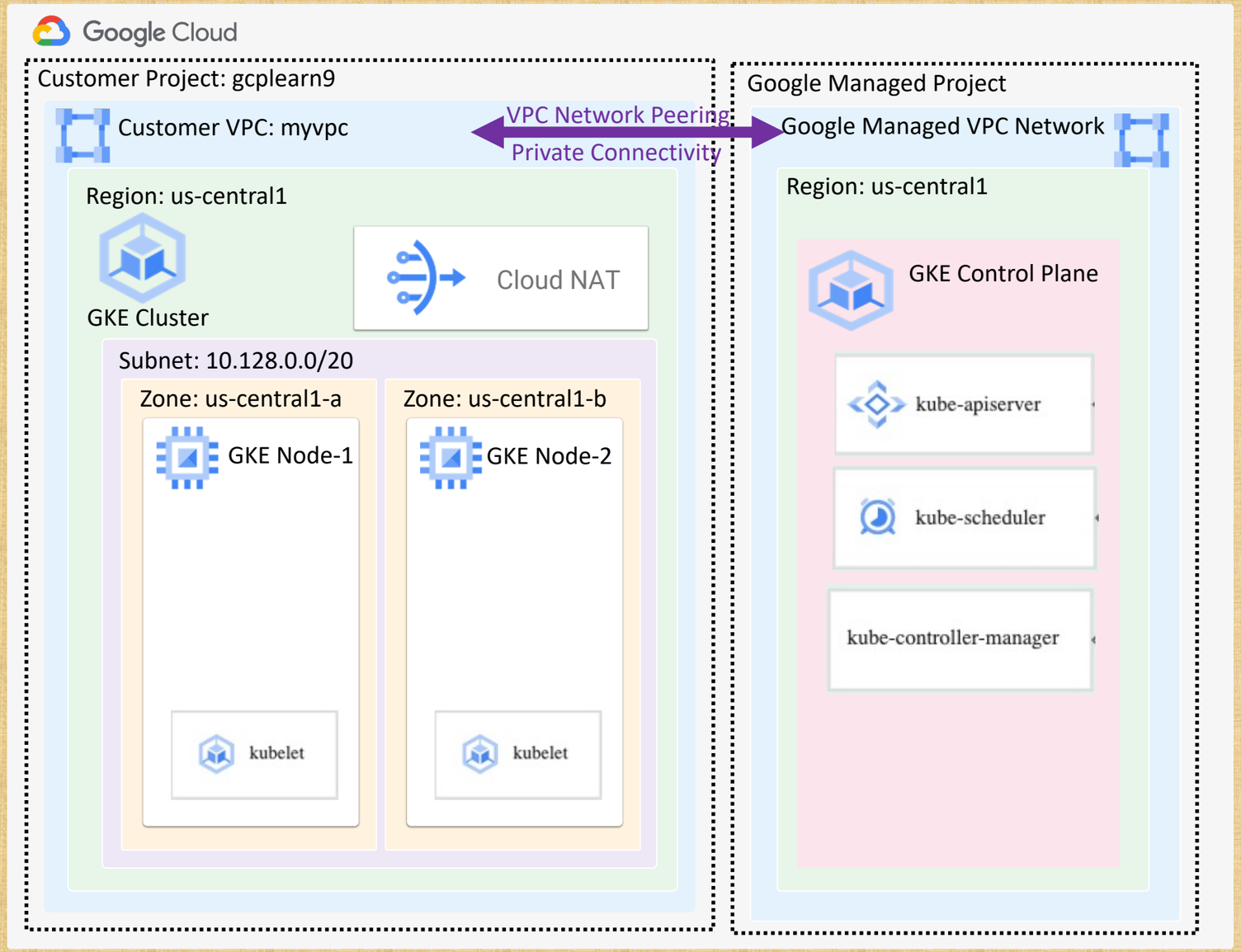
Demo: GKE Standard Private Cluster with Public Kubernetes API Server Endpoint + Cluster Autoscaler

Author: Nho Luong

Skill: DevOps Engineer Lead

GKE Private Cluster - Network Design

- **Private Clusters:** Control Plane VPC network is connected to customer (our) VPC network using VPC Network peering
- Our VPC network contains GKE nodes
- Google managed VPC network contains GKE cluster Control plane
- Traffic between nodes and control plane is routed using internal IP addresses only

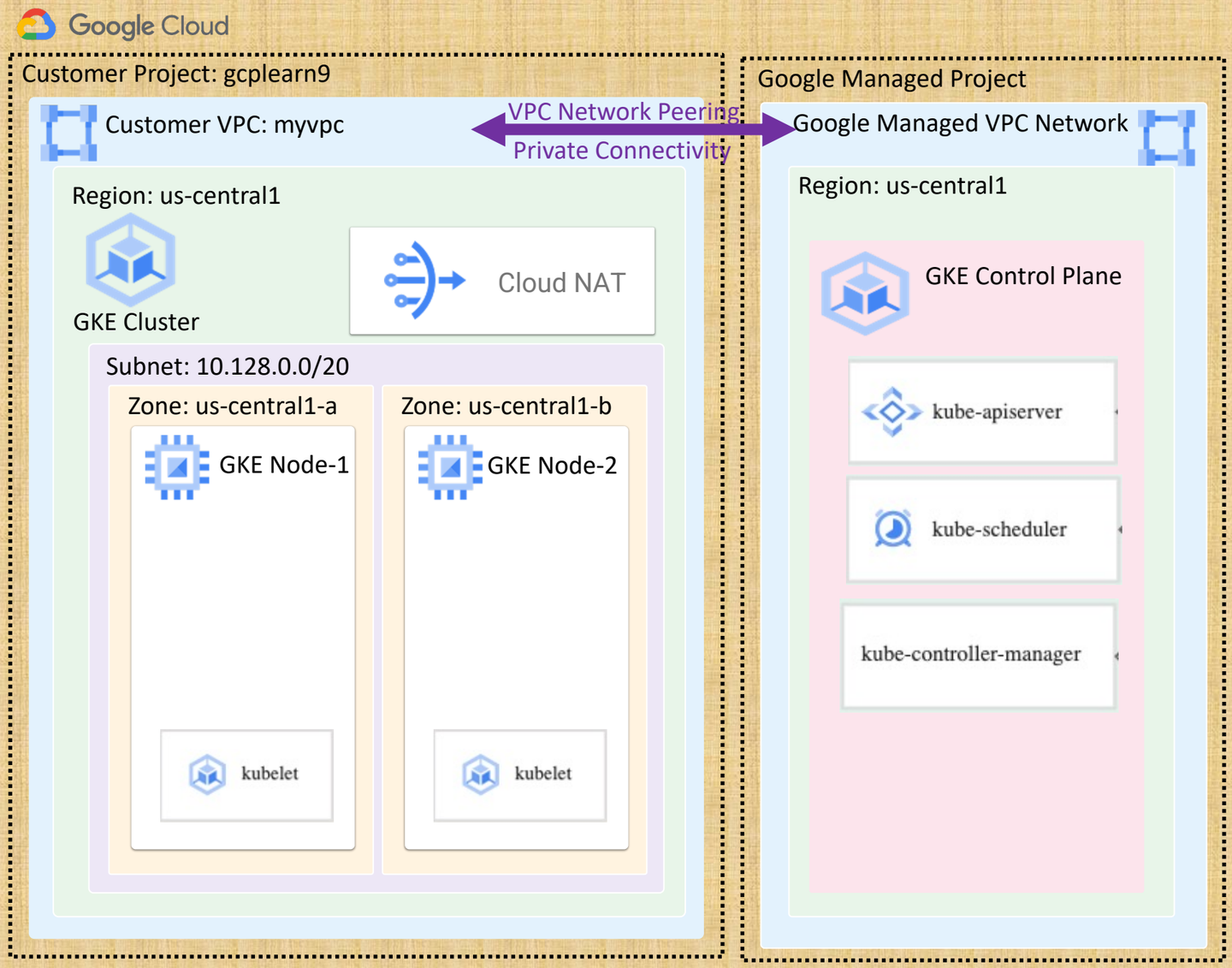


Author: Nho Luong

Skill: DevOps Engineer Lead

GKE Private Clusters

- You can create and configure private clusters in Standard or Autopilot.
- Private clusters will have nodes that do not have external IP addresses
- If you want to provide outbound internet access for certain private nodes, you can use Cloud NAT and Cloud Router



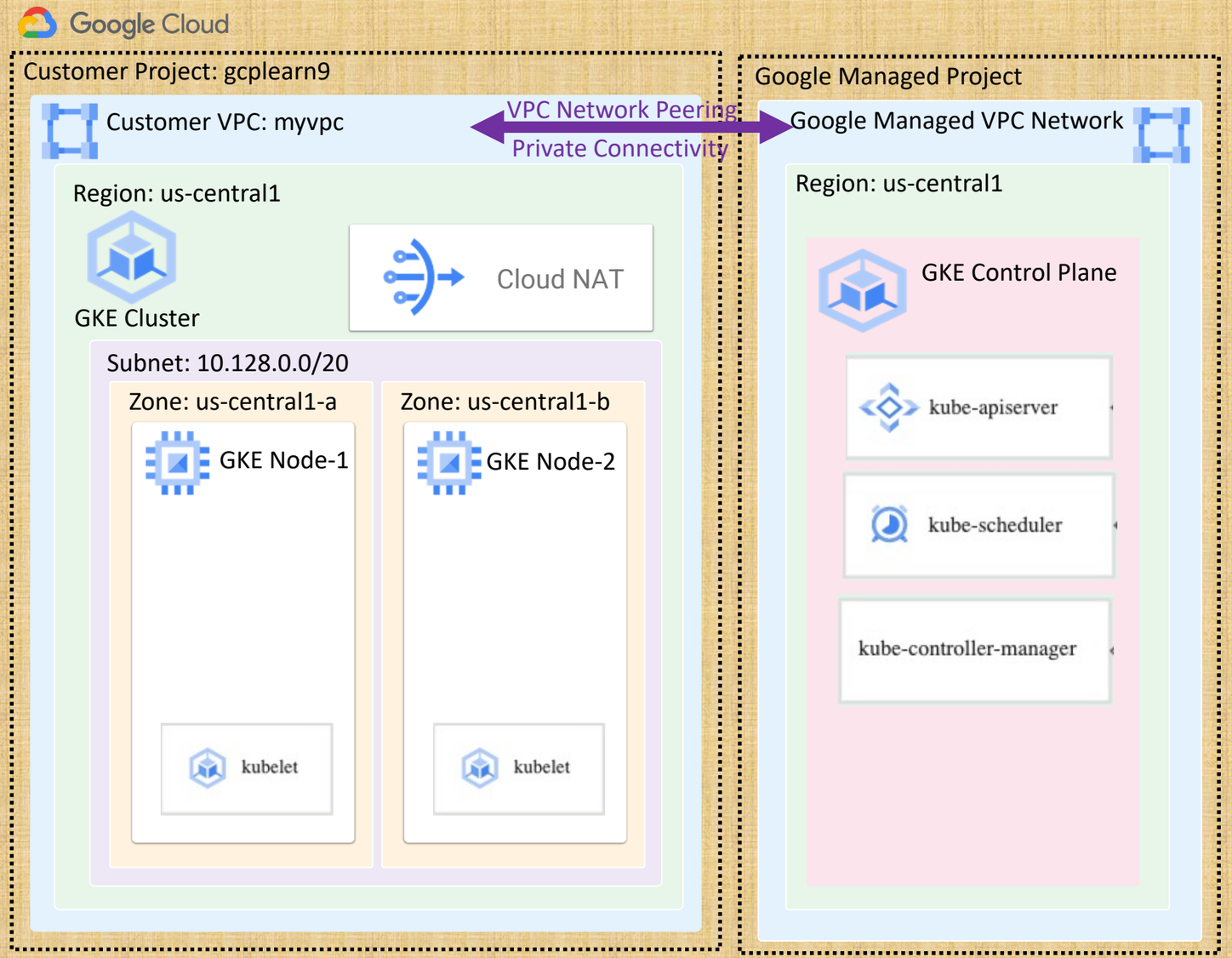
https://cloud.google.com/kubernetes-engine/docs/concepts/alias-ips#benefits

Author: Nho Luong

Skill: DevOps Engineer Lead

GKE Private Clusters

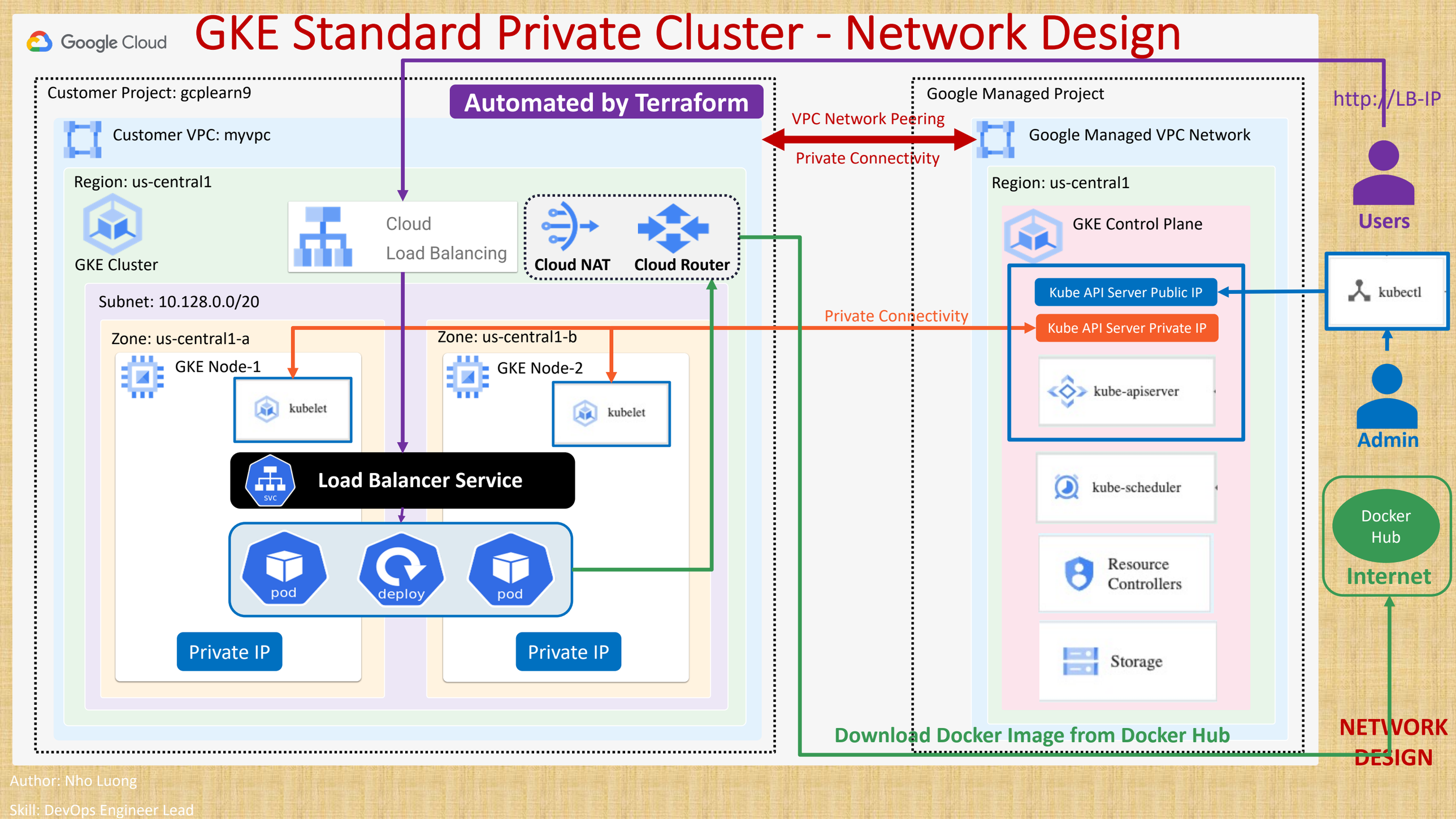
- In private clusters, private google access enabled by default on VPC subnet to access other Google cloud APIs and services using private network
  - **Example-1:** To access container images from Artifact Registry
  - **Example-2:** To send logs to Cloud Logging



Author: Nho Luong

Skill: DevOps Engineer Lead

GKE Standard Private Cluster - Network Design

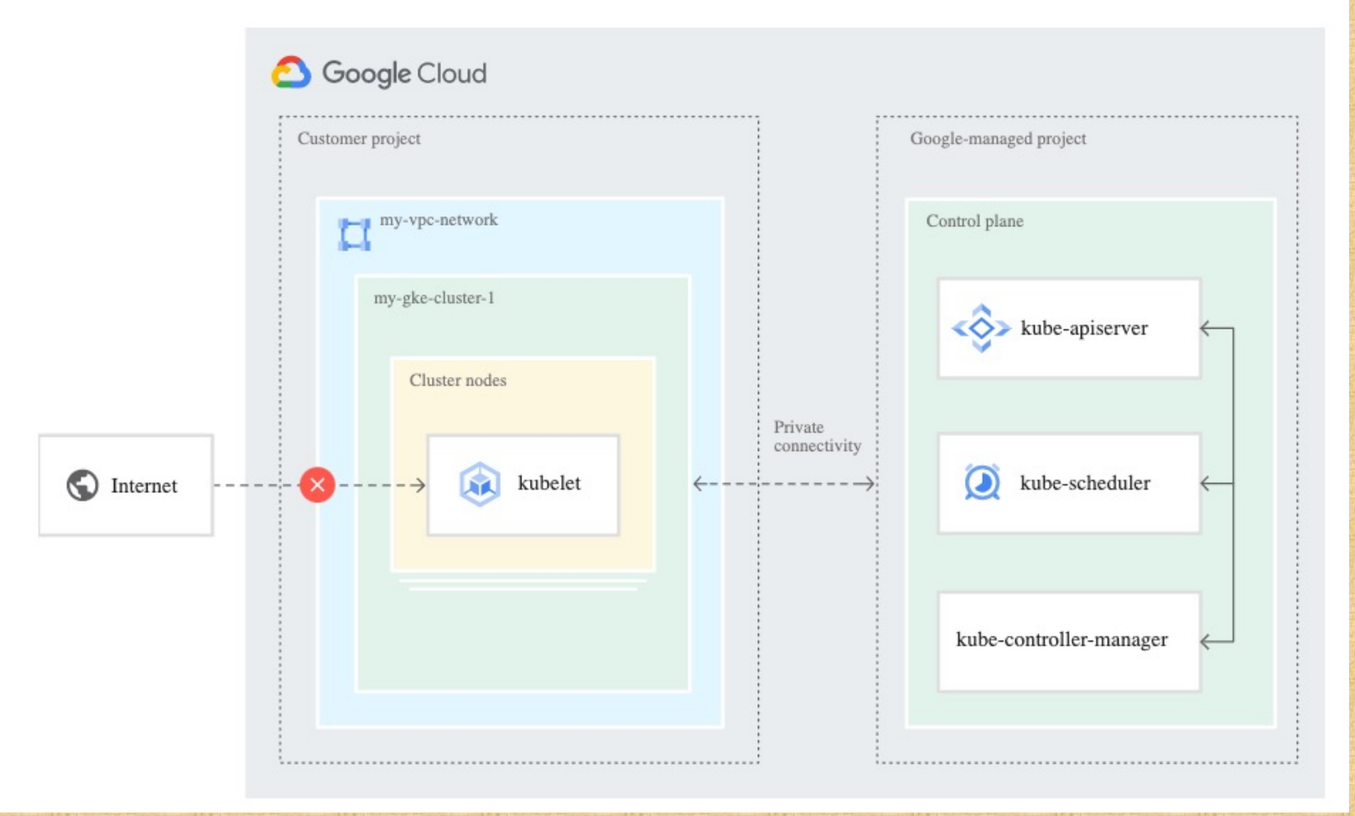


Author: Nho Luong

Skill: DevOps Engineer Lead

GKE Private Cluster - Access using kubectl

- **Least secure Option**
  - Public endpoint access: Enabled
  - Authorized Networks: Disabled
  - Accessible via Internet
- **Medium secure option: Demo-12**
  - Public endpoint access: Enabled
  - Authorized Networks: Enabled
  - Accessible via authorized internet IP ranges (Example: CloudShell, local desktop, from specified network in authorized network)
- **High secure option: Demo-15**
  - Public endpoint access: Disabled
  - Accessible via
    - VM in google cloud VPC network
    - On-premise network provided Cloud VPN or Cloud Interconnect is configured

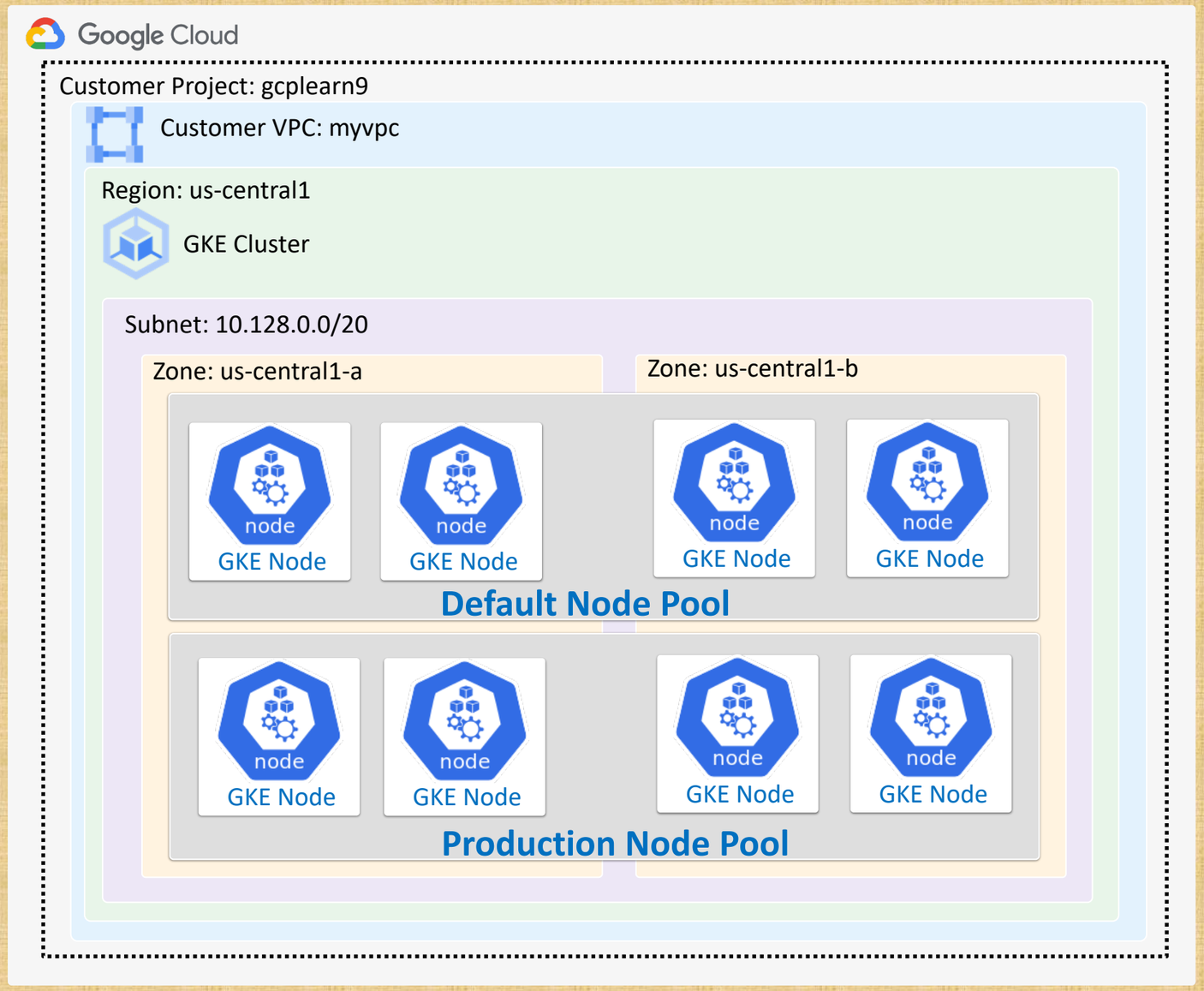


Author: Nho Luong

Skill: DevOps Engineer Lead

Kubernetes Cluster Autoscaler

- **Cluster Autoscaler:** Automatically resizes the number of nodes in a node pool, based on the demands of your workloads.
  - Minimum Size: 3
  - Maximum Size: 12
- **Scale-Out:** Gradually increases nodes to maximum size based on need (When pods are unschedulable)
- **Scale-In:** Decreases nodes to minimum size when nodes are idle (no workloads scheduled on them)

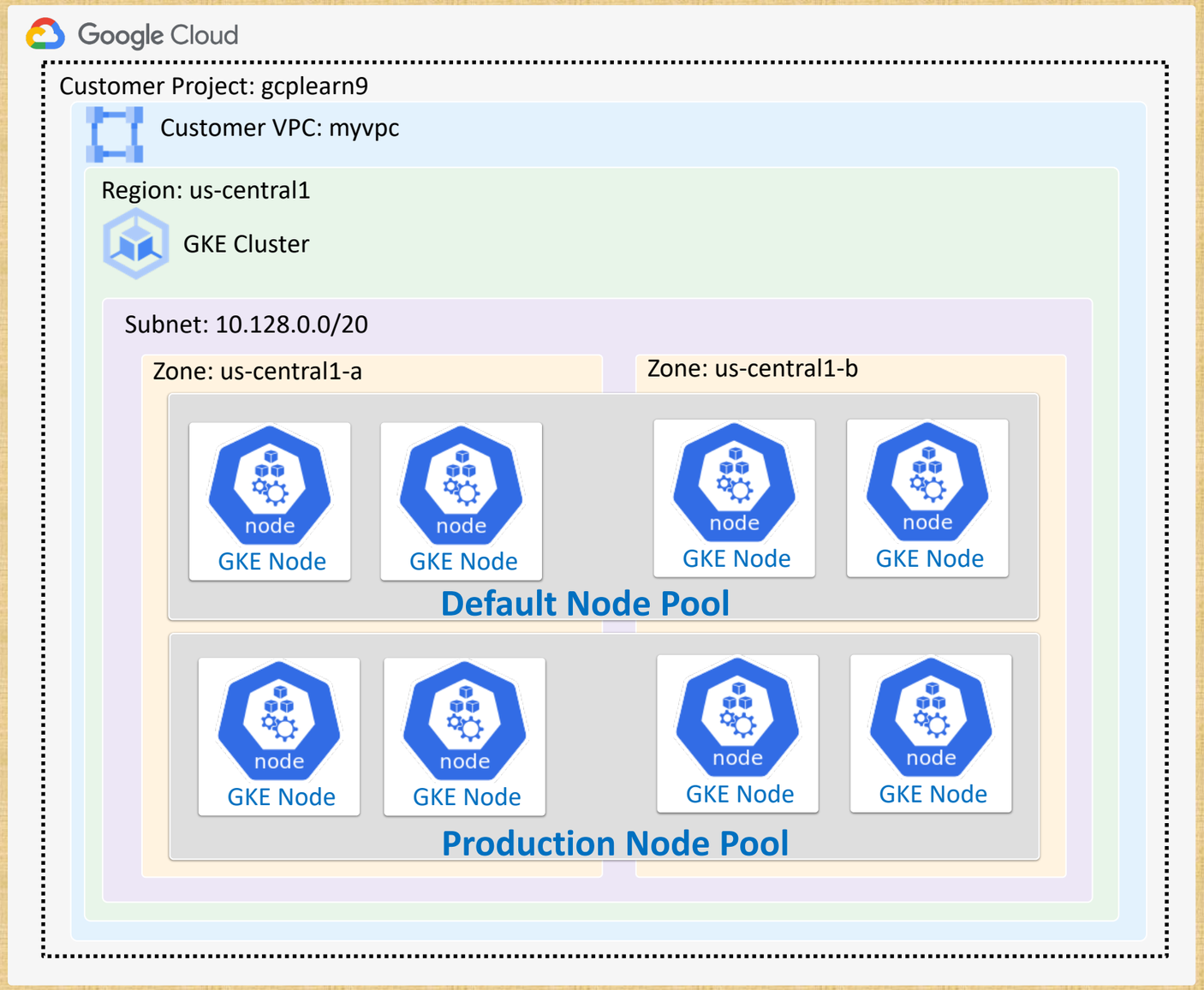


Author: Nho Luong

Skill: DevOps Engineer Lead

Kubernetes Cluster Autoscaler

- **Benefits**
  - Increases the availability of your workloads
  - Control the costs
  - Effective use of system resources (CPU, memory)
- **Standard vs Autopilot Clusters**
  - Standard Cluster uses Cluster Autoscaler
  - Autopilot cluster uses Node auto-provisioning
    - GKE automatically takes care of nodes and manage node pools.

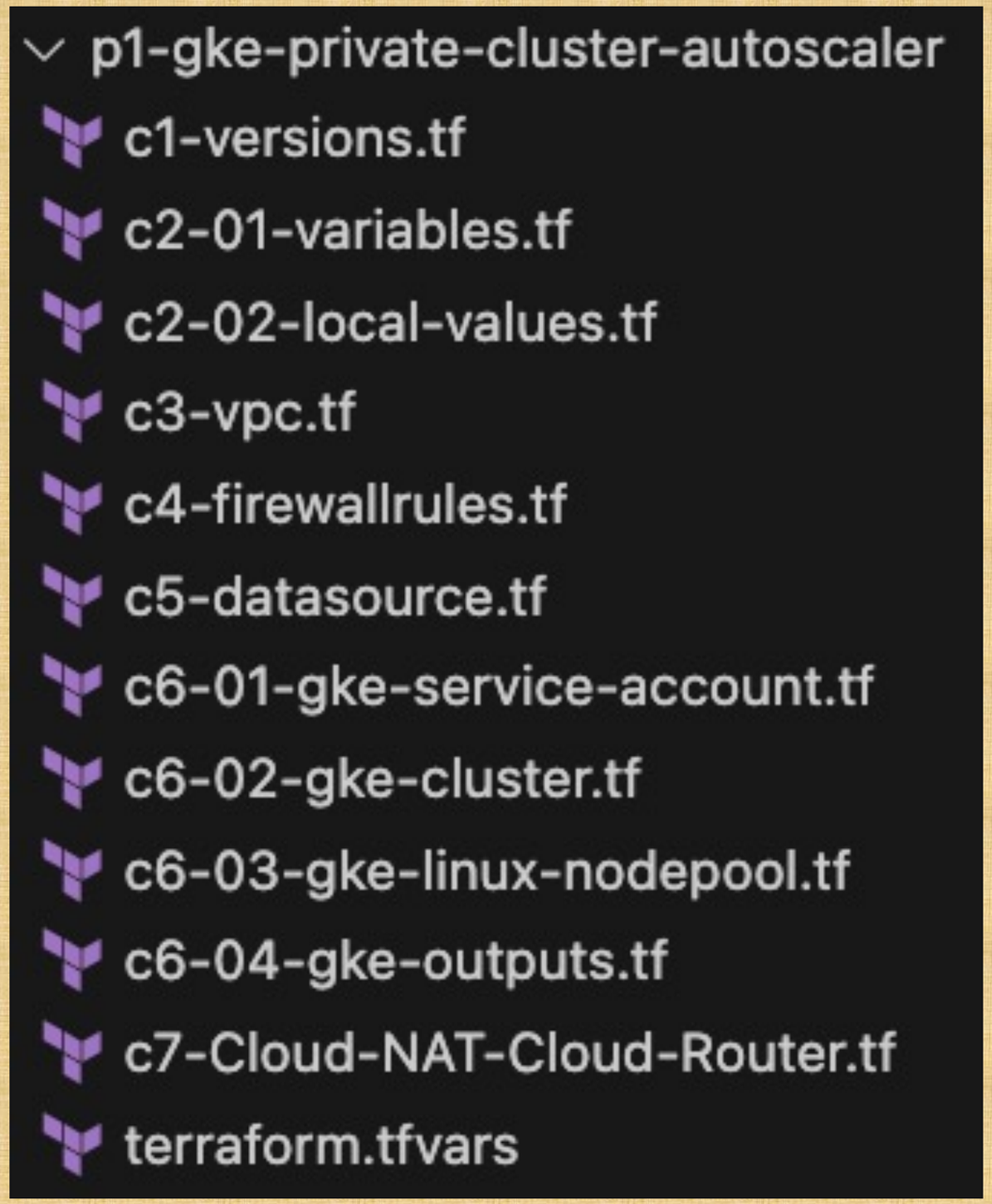


Author: Nho Luong

Skill: DevOps Engineer Lead

Project-1: GKE Cluster

What are we going to learn?



Demo: GKE Standard Private Cluster + k8s API Server Public Endpoint + Cluster Autoscaler

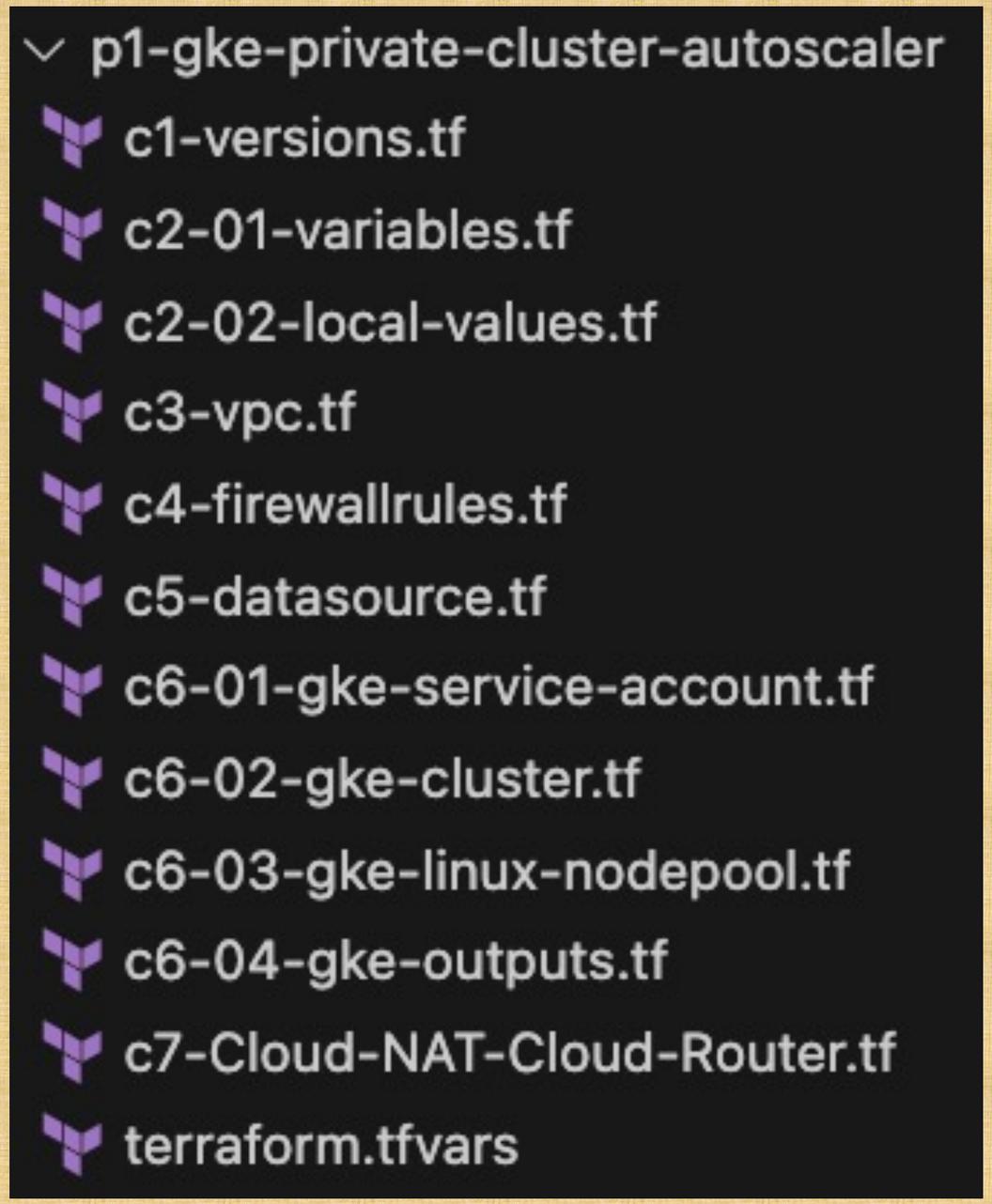
NC	No Changes from C2-02, C6-01
C1	Add Remote Backend as Cloud Storage Bucket
C2-01	Add new variables related to Private Cluster IP ranges
C3	Create a subnet with secondary IP ranges
C6-02	GKE Private cluster
C6-03	GKE Linux Node Pool + Cluster Autoscaler
C6-04	GKE Outputs

Author: Nho Luong

Skill: DevOps Engineer Lead

Project-1: GKE Cluster

What are we going to learn?



Demo: GKE Standard Private Cluster + k8s API Server Public Endpoint

C7

Cloud NAT and Cloud Router Resources

TFVARS

Update Input variable values: terraform.tfvars related to Private Cluster CIDR Ranges

Author: Nho Luong

Skill: DevOps Engineer Lead

Project-3: k8s Terraform Manifests

What are we going to learn?

Demo: Kubernetes Deployment and Service using Terraform + Terraform Remote State Data source concept

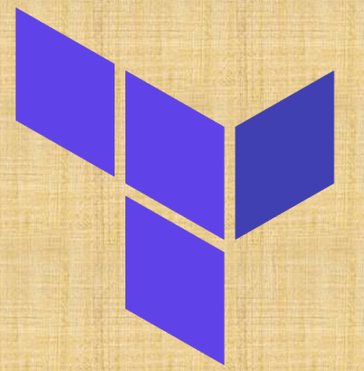
NC	No Changes: c1, c2-01, c2-02, c3-02, c4, c5
C3-01	Update project-1 Terraform state file bucket prefix in Remote State Data source

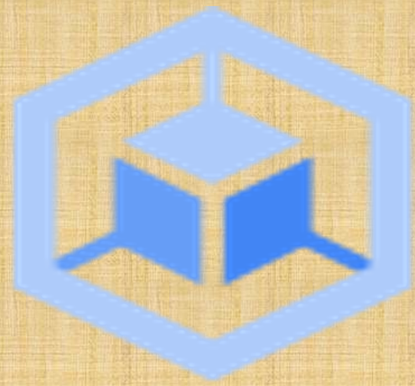
Author: Nho Luong

Skill: DevOps Engineer Lead

Demo-13

Google Kubernetes Engine Horizontal Pod Autoscaler





Demo: GKE Horizontal Pod Autoscaler

Author: Nho Luong

Skill: DevOps Engineer Lead

Kubernetes Horizontal Pod Autoscaling

- Automatically increase or decrease number of pods in response to
  - Workloads CPU and Memory Utilization
  - Custom metrics reported from within Kubernetes Cluster
  - External metrics (Load Balancers, Messaging Services …)
  - Custom metrics using Managed Service for Prometheus
- HPA automatically scales the pods in workload types
  - Kubernetes ReplicaSet
  - Kubernetes Replication Controller
  - Kubernetes Deployment
  - Kubernetes StatefulSet
- This can help our applications
  - **Scale out** to meet increased demand or
  - **Scale in** when resources are not needed, thus freeing up your worker nodes for other applications

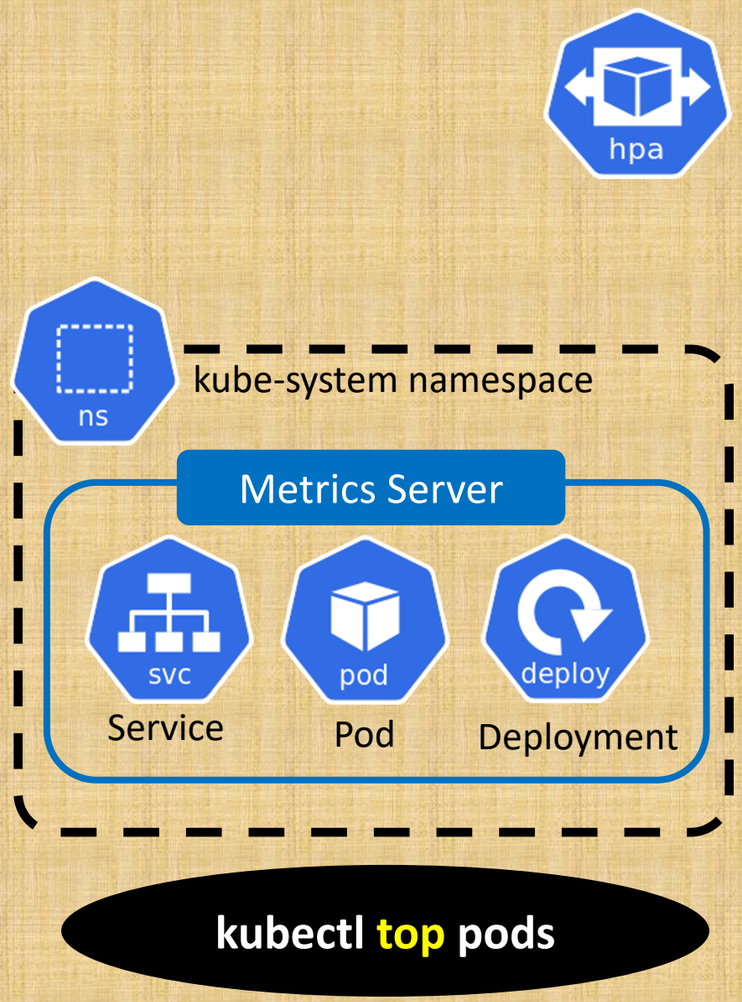
HPA Imperative Command: kubectl autoscale deployment my-app --max 6 --min 4 --cpu-percent 50

Author: Nho Luong

Skill: DevOps Engineer Lead

Kubernetes Metrics Server

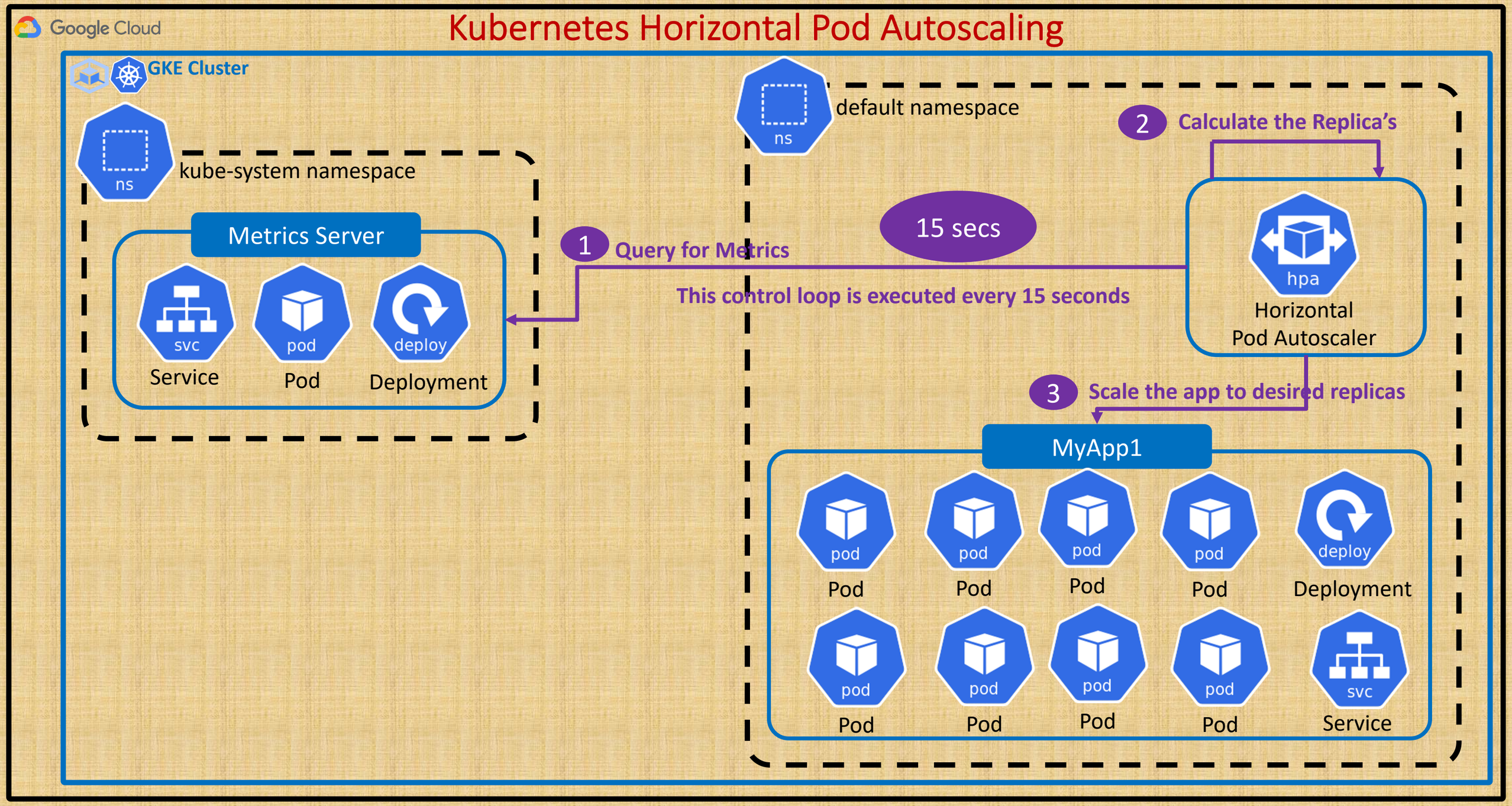
- Metrics Server collects resource metrics from Kubelets and exposes them in Kubernetes apiserver through Metrics API
- Metrics API can also be accessed by kubectl top, making it easier to debug autoscaling pipelines.
- Fast Autoscaling solution, collecting metrics every 15 seconds
- Metrics Server is not meant for non-autoscaling purposes. For example, don't use it to forward metrics to monitoring solutions.
- Resource efficiency, using 1 mili core of CPU and 2 MB of memory for each node in a cluster
- Metrics Server used for CPU/Memory based horizontal autoscaling
- Metrics Server used for automatically adjusting/suggesting resources needed by containers (Vertical Autoscaling)



Author: Nho Luong

Skill: DevOps Engineer Lead

Kubernetes Horizontal Pod Autoscaling



Author: Nho Luong

Skill: DevOps Engineer Lead

Kubernetes Horizontal Pod Autoscaling

Autoscaling v1

```yaml
apiVersion: autoscaling/v1
kind: HorizontalPodAutoscaler
metadata:
 name: hpa-myapp1
spec:
 scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: myapp1-deployment
 minReplicas: 1
 maxReplicas: 10
 targetCPUUtilizationPercentage: 50
```

Autoscaling v2

```yaml
apiVersion: autoscaling/v2
kind: HorizontalPodAutoscaler
metadata:
  name: cpu
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: myapp1-deployment
  minReplicas: 1
  maxReplicas: 10
  metrics:
  - type: Resource
    resource:
      name: cpu
      target:
        type: Utilization
        averageUtilization: 30
```

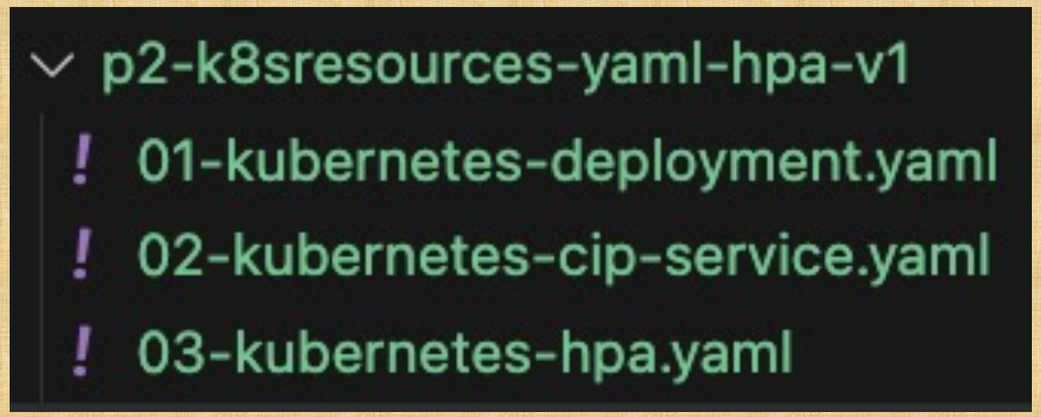
Horizontal
Pod Autoscaler

Author: Nho Luong

Skill: DevOps Engineer Lead

What are we going to learn?

Project-2: HPA V1



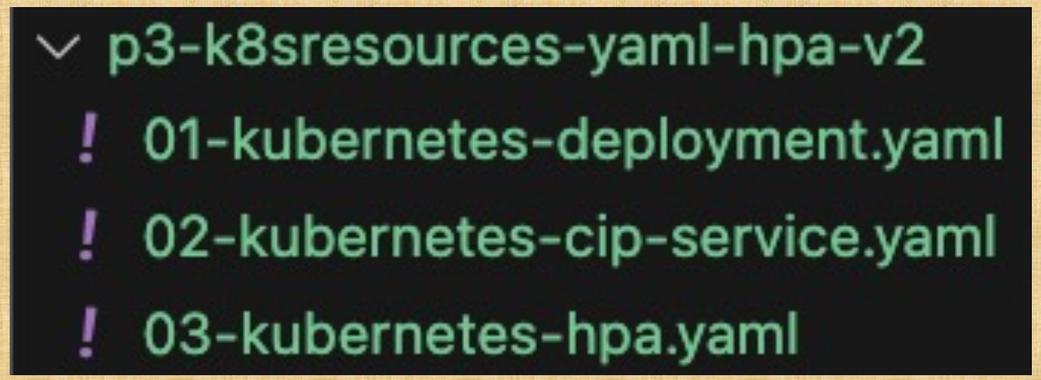
Demo: HPA using Autoscaler V1 API Version

01 Kubernetes Deployment

02 Kubernetes ClusterIP Service

03 Kubernetes HPA Autoscaler V1 API Version

Project-3: HPA V2



Demo: HPA using Autoscaler V1 API Version

01 Kubernetes Deployment

02 Kubernetes ClusterIP Service

03 Kubernetes HPA Autoscaler V2 API Version

Author: Nho Luong

Skill: DevOps Engineer Lead

Project-4: HPA V2 Terraform Manifests

What are we going to learn?

Demo: GKE Standard Private Cluster + HPA V2

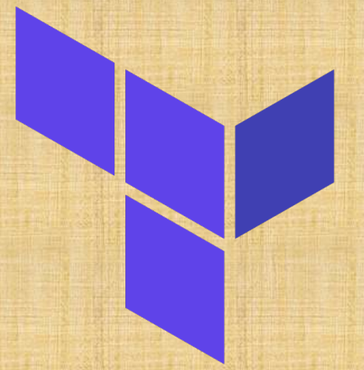
NC	No Changes from C1 to C4
C5	Kubernetes ClusterIP Service
C6	Kubernetes HPA Autoscaler V2 API Version

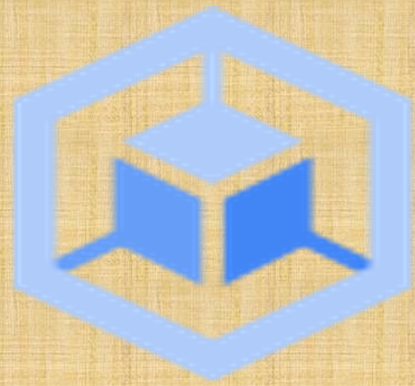
Author: Nho Luong

Skill: DevOps Engineer Lead

Demo-14

Google Kubernetes Engine Vertical Pod Autoscaler





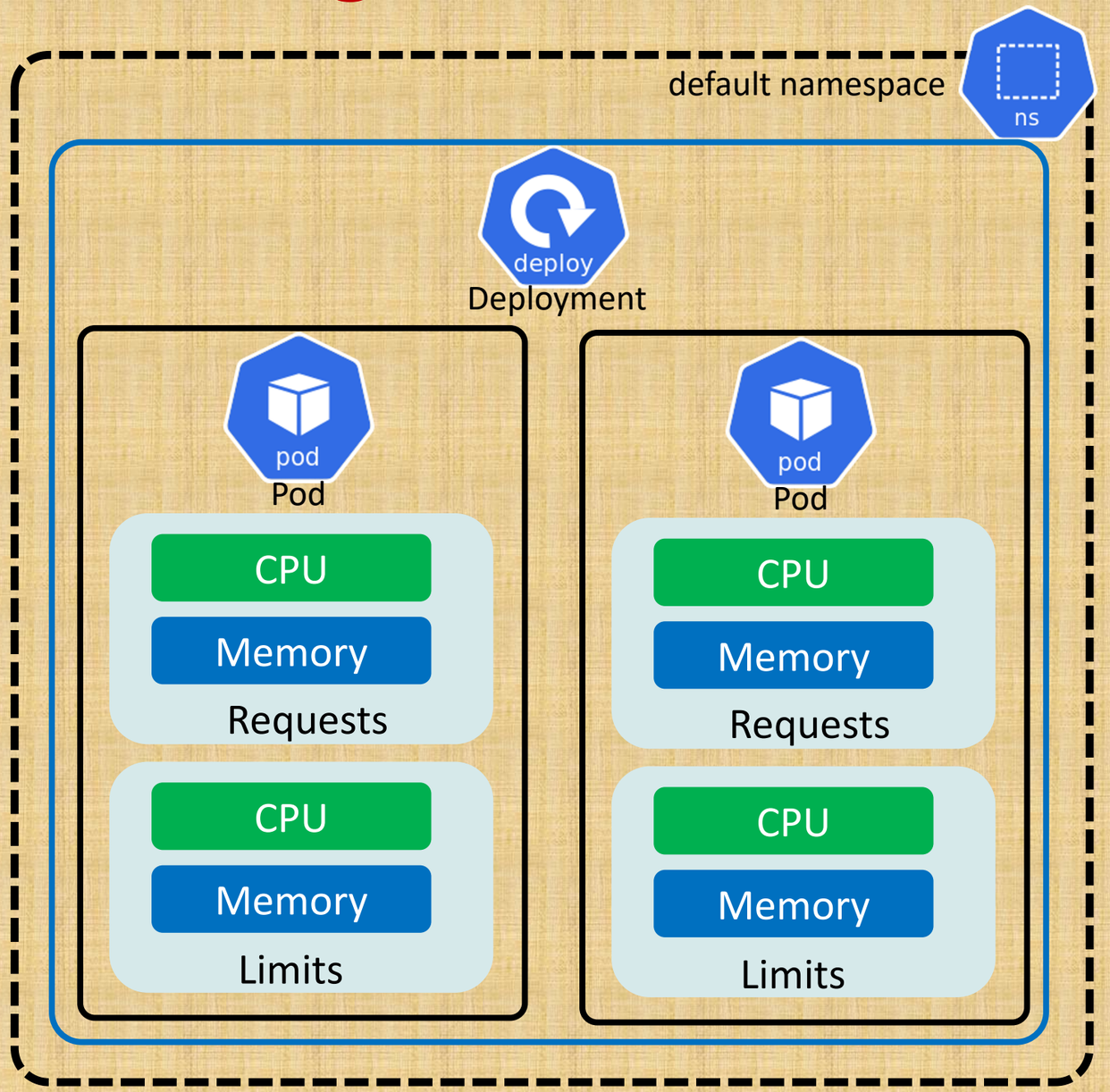
Demo: GKE Vertical Pod Autoscaler

Author: Nho Luong

Skill: DevOps Engineer Lead

Kubernetes Vertical Pod Autoscaling

- **Vertical Pod autoscaling:** Lets you analyze and set CPU and memory resources required by Pods
- **Manual**
  - VPA recommends cpu and memory requests and limits
  - We can review the recommendation and update the values manually
- **Automatic**
  - VPA recommends and automatically updates the cpu and memory requests and limits
  - VPA evicts and recreates the pod during the resource request updates

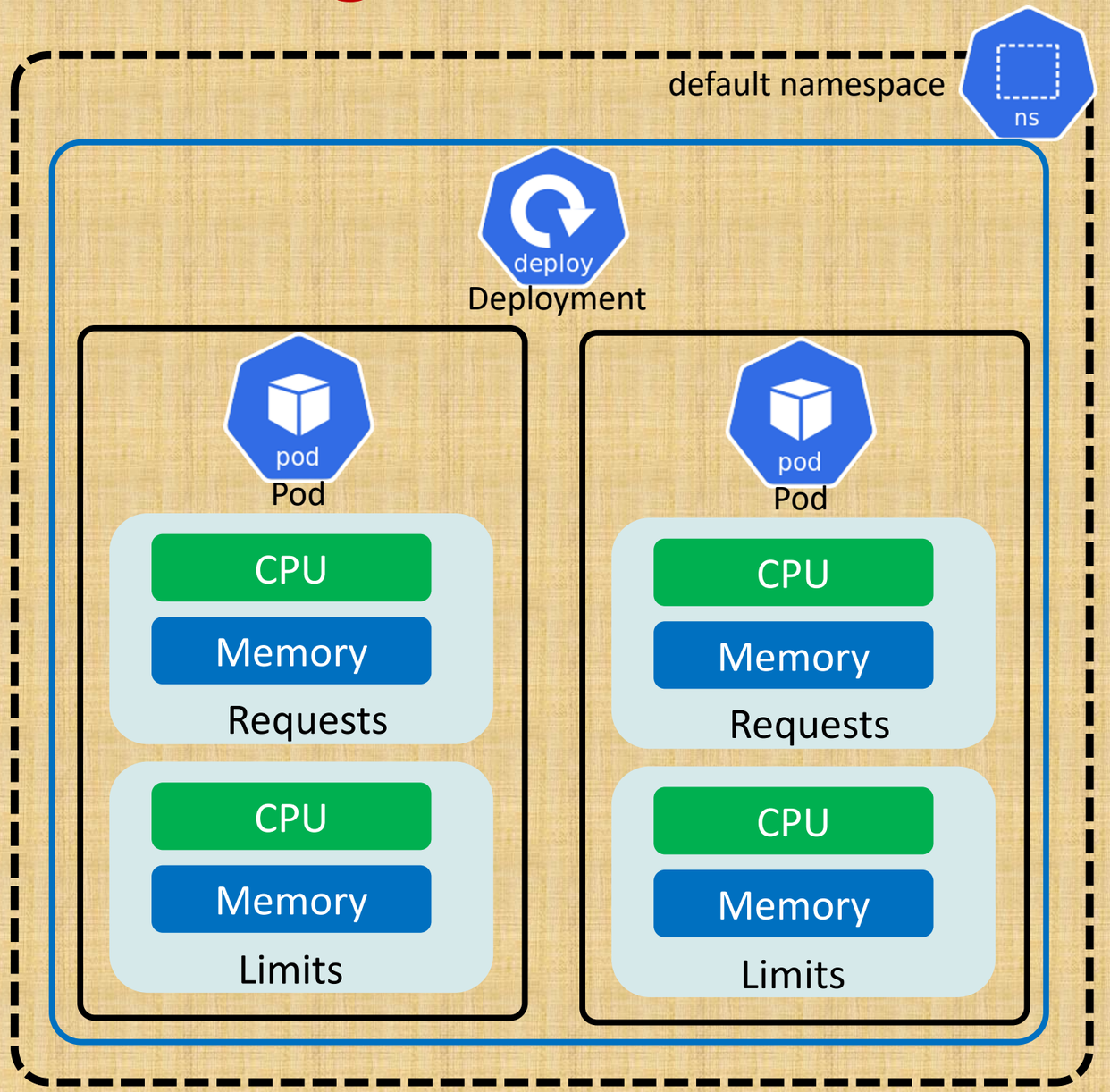


Author: Nho Luong

Skill: DevOps Engineer Lead

Kubernetes Vertical Pod Autoscaling

- **Standard vs Autopilot clusters**
  - VPA need to be enabled at workload level for standard clusters
  - VPA is by default enabled in Autopilot clusters
- **Benefits**
  - Cluster nodes are used efficiently because Pods use exactly what they need
  - You don't have to run time-consuming benchmarking tasks to determine the correct values for CPU and memory requests
  - Pods are scheduled onto nodes that have the appropriate resources available
  - Runs Vertical Pod Autoscaler Pods as control plane processes instead of Deployment on worker nodes (GKE special feature)
- **Important Note:**
  - Enable Cluster Autoscaler before enabling VPA
  - VPA can talk to Cluster Autoscaler to increase the number of nodes if needed for the VPA enabled workloads

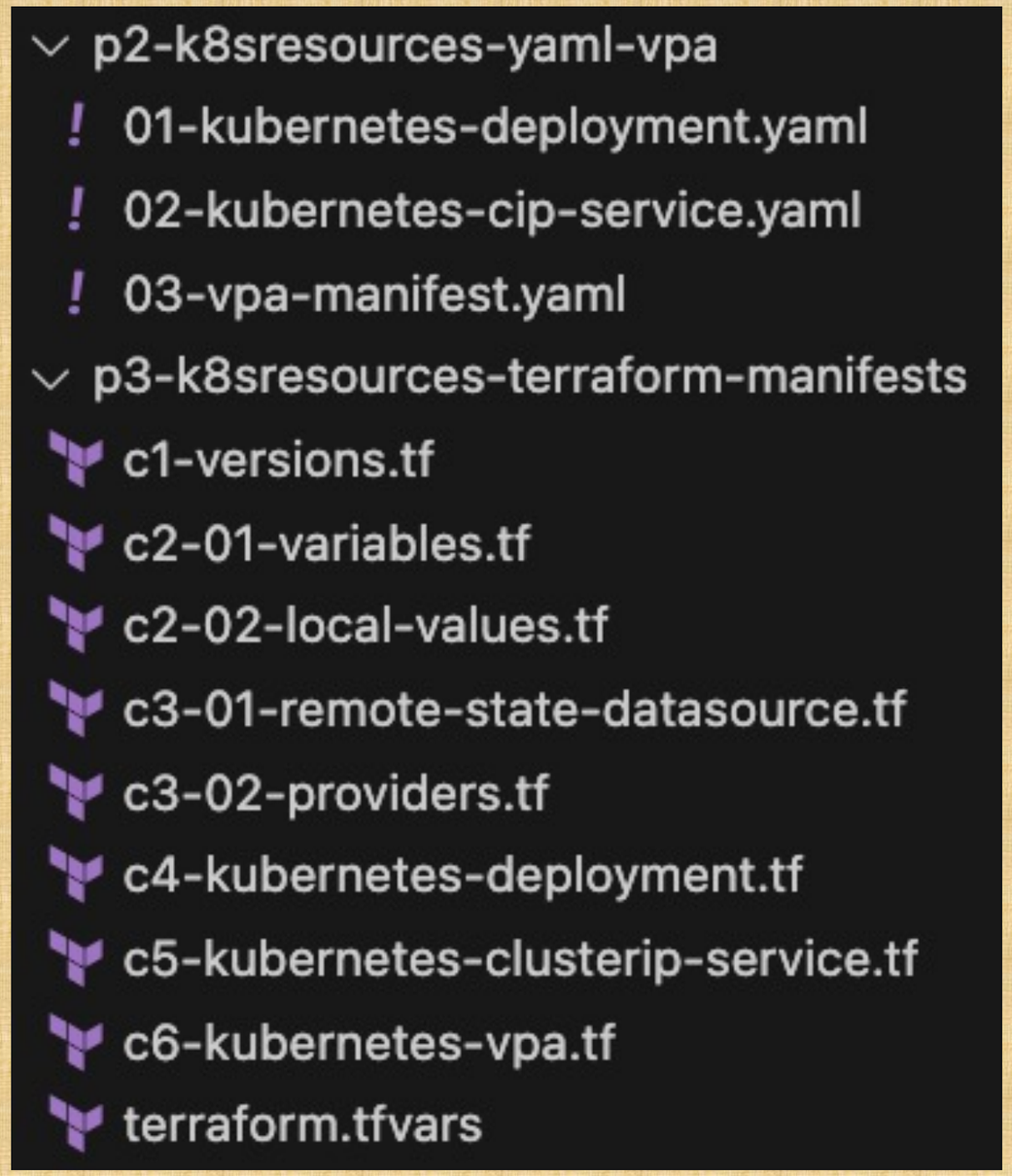


Author: Nho Luong

Skill: DevOps Engineer Lead

Project-2 & 3: VPA YAML & Terraform Manifests

What are we going to learn?



Demo: Project-2: VPA YAML Manifests

NC	No Changes – 01, 02
03	Kubernetes Vertical Pod Autoscaler

Demo: Project-3: VPA Terraform Manifests

NC	No Changes from C1 to C5
C6	Kubernetes Vertical Pod Autoscaler

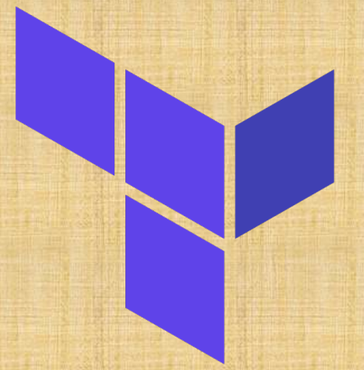
Author: Nho Luong

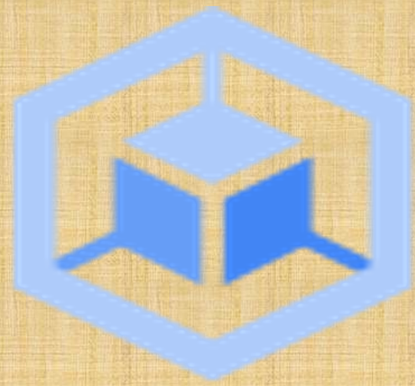
Skill: DevOps Engineer Lead

Demo-15

Google Kubernetes Engine

Create GKE Standard Private Cluster with k8s API Server Private Endpoint





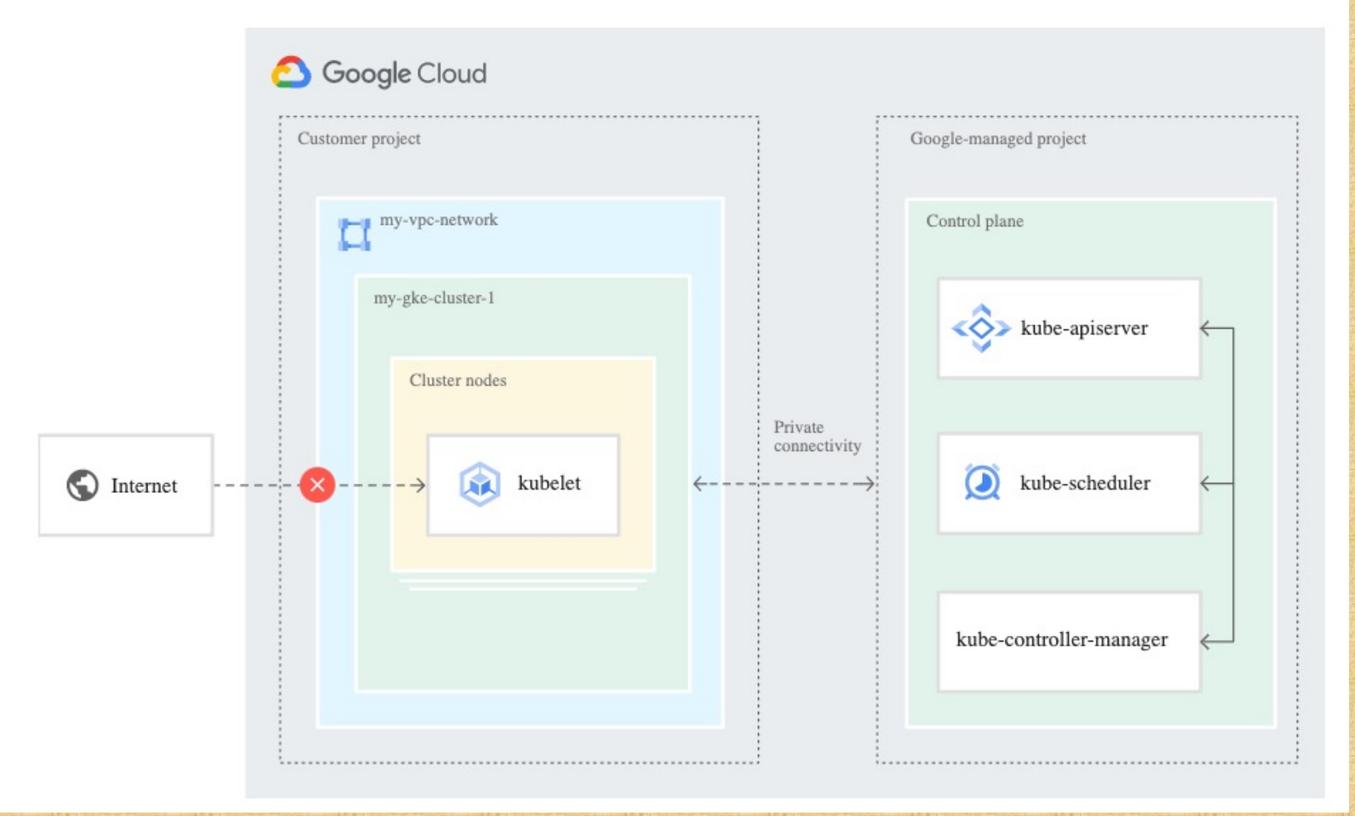
Demo: GKE Standard Private Cluster with Private Kubernetes API Server Endpoint

Author: Nho Luong

Skill: DevOps Engineer Lead

GKE Private Cluster - Access using kubectl

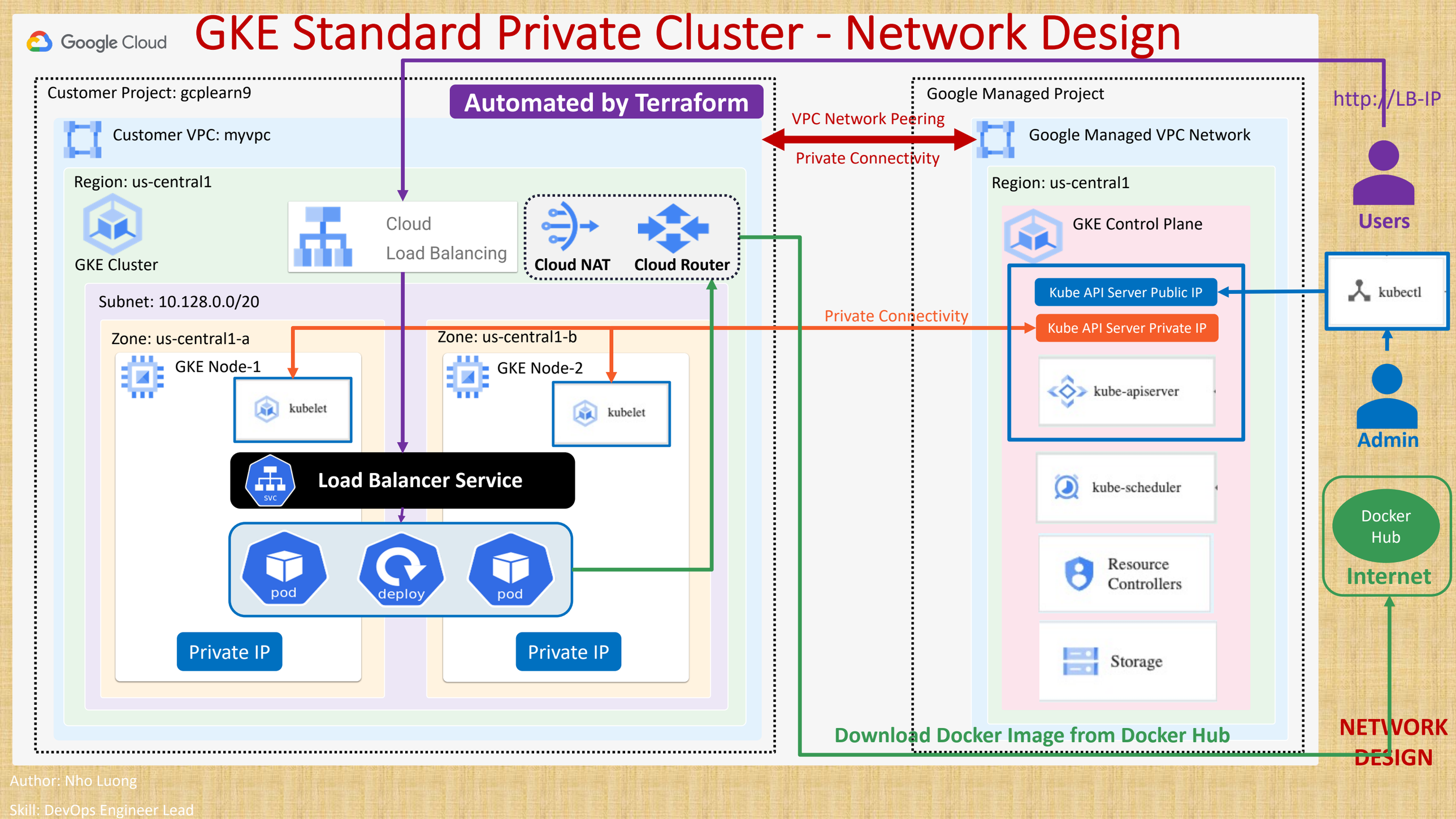
- **Least secure Option**
  - Public endpoint access: Enabled
  - Authorized Networks: Disabled
  - Accessible via Internet
- **Medium secure option: Demo-12**
  - Public endpoint access: Enabled
  - Authorized Networks: Enabled
  - Accessible via authorized internet IP ranges (Example: CloudShell, local desktop, from specified network in authorized network)
- **High secure option: Demo-15**
  - Public endpoint access: Disabled
  - Accessible via
    - VM in google cloud VPC network
    - On-premise network provided Cloud VPN or Cloud Interconnect is configured



Author: Nho Luong

Skill: DevOps Engineer Lead

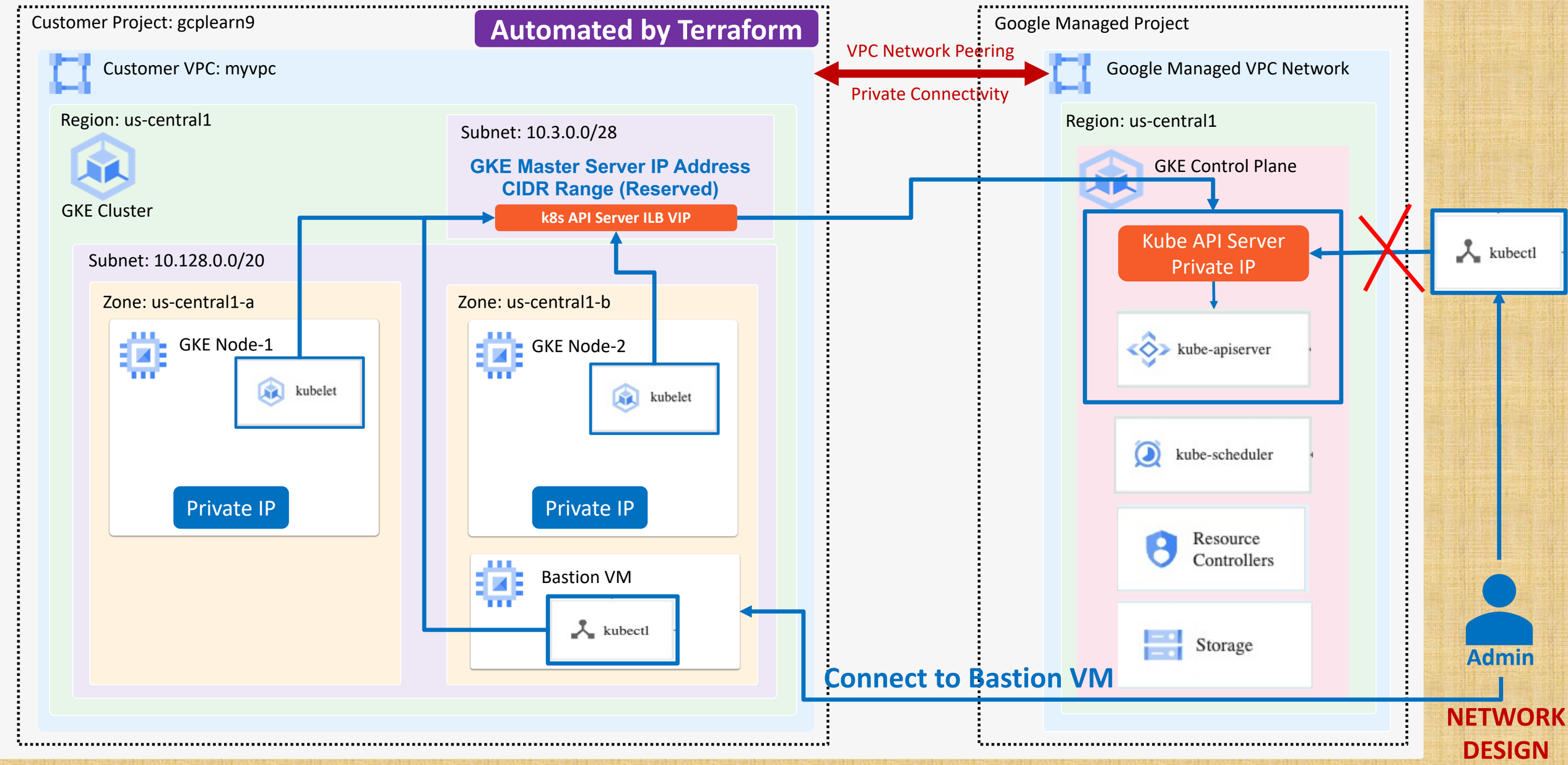
GKE Standard Private Cluster - Network Design



Author: Nho Luong

Skill: DevOps Engineer Lead

GKE Standard Private Cluster with Private Endpoint



Author: Nho Luong

Skill: DevOps Engineer Lead

Project-1: GKE Private Cluster + Private Endpoint

What are we going to learn?

Demo: GKE Standard Private Cluster + Private k8s API Server EnpPoint

NC	No Changes from C1 to C3
C4	Create Firewall rule with source as IAP IP Range (Bastion VM accessible using IAP only)
C6-02	GKE cluster – Enable Private Endpoint setting
C6-03	Comment Tags where earlier SSH firewall referenced
C8	Bastion VM Instance

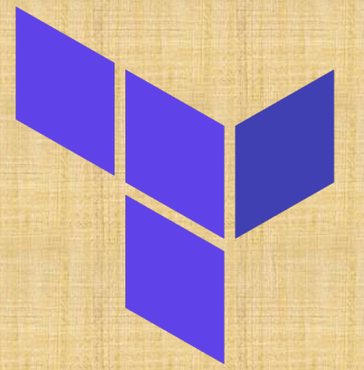
Author: Nho Luong

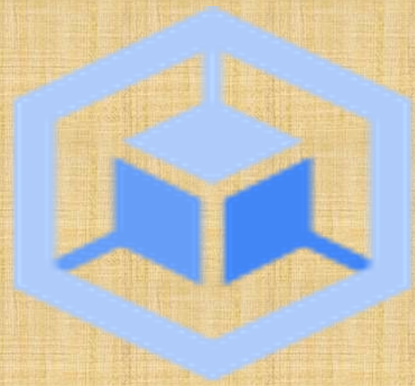
Skill: DevOps Engineer Lead

Demo-16

Google Kubernetes Engine

Create GKE Autopilot Private cluster with k8s API Server Public endpoint





Demo: GKE Autopilot Private Cluster + k8s API Server Public API endpoint

Author: Nho Luong

Skill: DevOps Engineer Lead

GKE Autopilot Cluster

	Autopilot mode	Standard mode
	Optimized Kubernetes cluster with a hands-off experience	Kubernetes cluster with node configuration flexibility
	CONFIGURE TRY THE DEMO	CONFIGURE TRY THE DEMO
Scaling	Automatic based on workload	You configure scaling
Nodes	Google manages and configures your nodes	You manage and configure your nodes
Configuration	Streamlined configuration ready to use	You can configure all options
Workloads supported	Most workloads except these limitations	All Kubernetes workloads
Billing method	Pay per pod	Pay per node (VM)
SLA	Kubernetes API and node availability	Kubernetes API availability

Complete Comparison: https://cloud.google.com/kubernetes-engine/docs/resources/autopilot-standard-feature-comparison

Author: Nho Luong

Skill: DevOps Engineer Lead

GKE Autopilot Cluster

Pricing: Pay per pod, Pay for the CPU, memory and storage used by our workloads

Pricing: Not charged for system workloads, unused capacity on nodes, Operating System Costs

Security: Clusters have a super hardened configs with many security settings enabled by default.

Security: GKE automatically applies the security patches.

Node Management: Google manages Worker Nodes (Creation, Upgrades, Repairs)

Networking: very advanced. All pod traffic passes through VPC Firewalls.

GKE Autopilot Cluster

Scaling: When high load, GKE provisions new nodes for those pods (Node Autoprovisioning)

Resource Management: If we don't specify resource values in workloads, Autopilot sets pre-configured default values

Release Management: All Autopilot clusters are enrolled in GKE release channel which ensures control plane and nodes run on latest versions in that channel

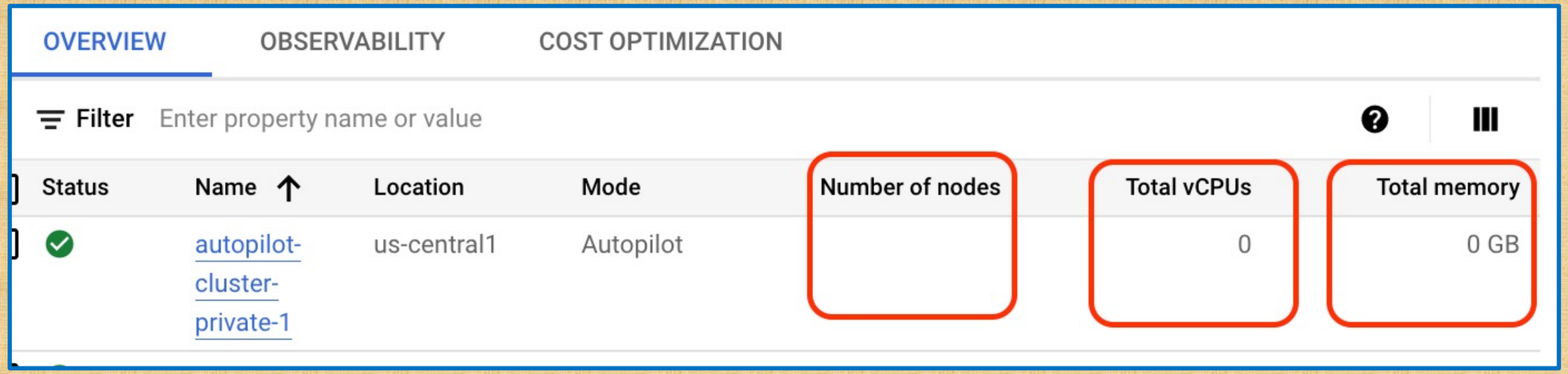
Managed Flexibility: Autopilot supports specific hardware requirements (Compute Classes: General Purpose, Balanced, Scale-Out)

Reduced Operational Complexity: Autopilot reduces platform administration overhead by removing need to continuously monitor nodes, scaling and scheduling operations

Author: Nho Luong

Skill: DevOps Engineer Lead

Autopilot Cluster



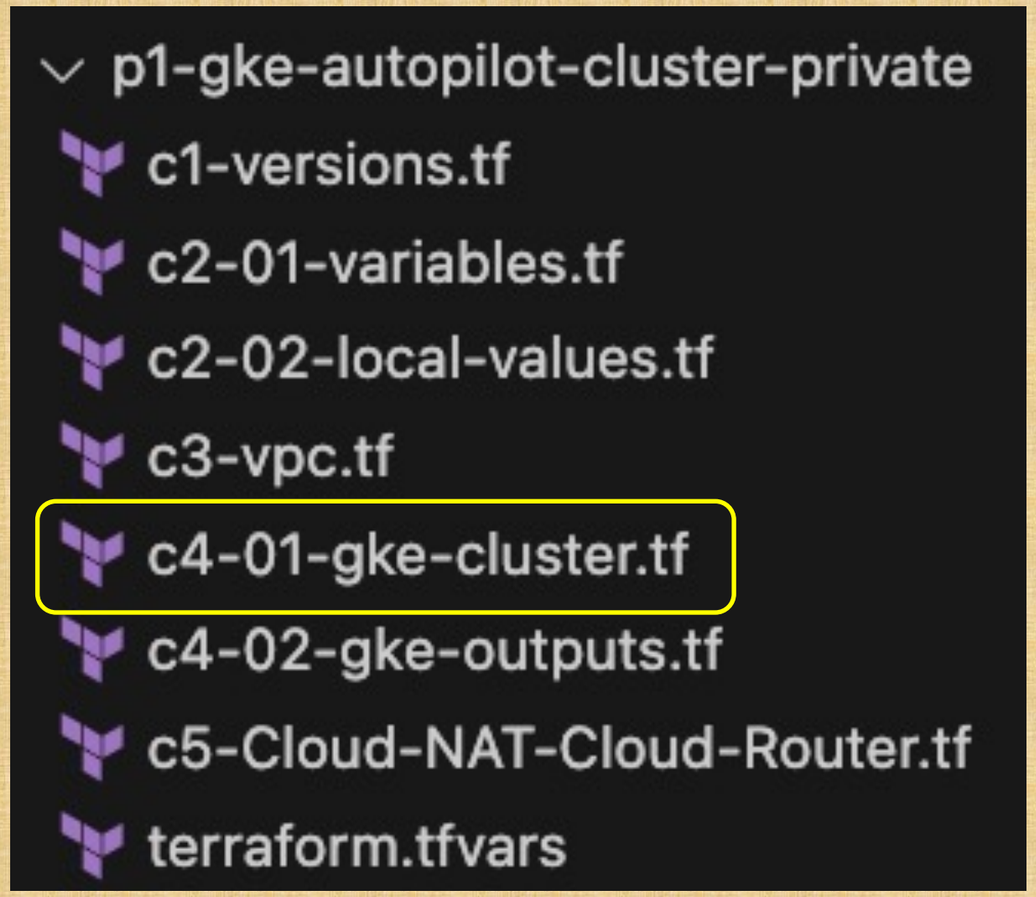
Observation: Once the Autopilot cluster is created, if we don't deploy any workloads, after some time Number of nodes, Total vCPUs and Total Memory all will come to zero.

Author: Nho Luong

Skill: DevOps Engineer Lead

Project-1: GKE Autopilot Private Cluster

What are we going to learn?



Demo: GKE Autopilot Private Cluster + k8s API Server Public Endpoint

NC — No Changes from C1 to C3, C5, C4-02

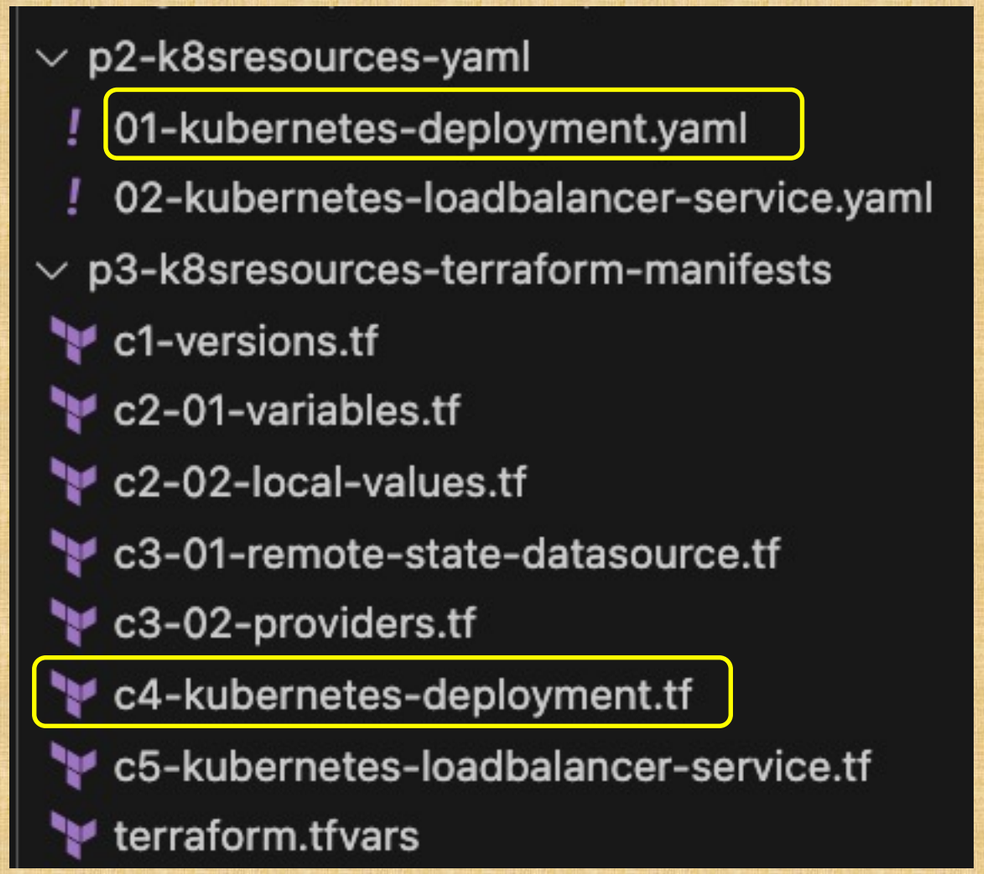
C4-01 — Create GKE Autopilot private cluster with `enable_autopilot = true`

Author: Nho Luong

Skill: DevOps Engineer Lead

Project-2 & 3: YAML & Terraform Manifests

What are we going to learn?



Demo: Project-2: YAML Manifests

NC	No Changes in 02
01	Kubernetes Deployment: Add Container Resource Request and Limits (Optional but recommended)

Demo: Project-3: Terraform Manifests

NC	No Changes from C1 to C3-02 and C5
C4	Kubernetes Deployment: Add Container Resource Request and Limits (Optional but recommended)

Author: Nho Luong

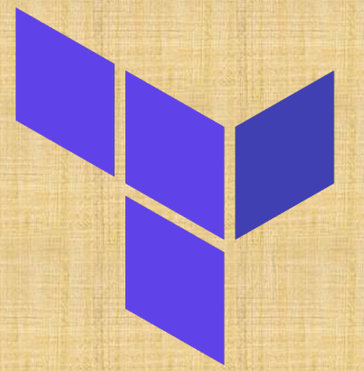
Skill: DevOps Engineer Lead

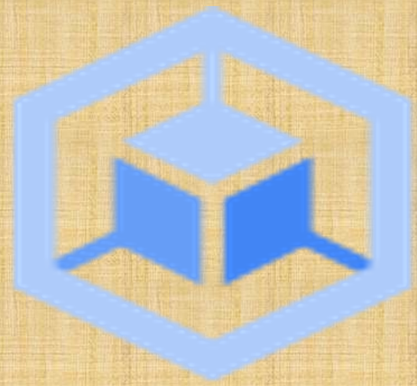
Demo-17

Google Kubernetes Engine

Kubernetes Storage

Compute Engine Persistent Disks





Demo: GKE Autopilot cluster + Compute Engine persistent disks as storage

Author: Nho Luong

Skill: DevOps Engineer Lead

Kubernetes Storage - Key Resources

Kubernetes
Storage Class

Kubernetes
Persistent Volume

Kubernetes
Persistent Volume Claim

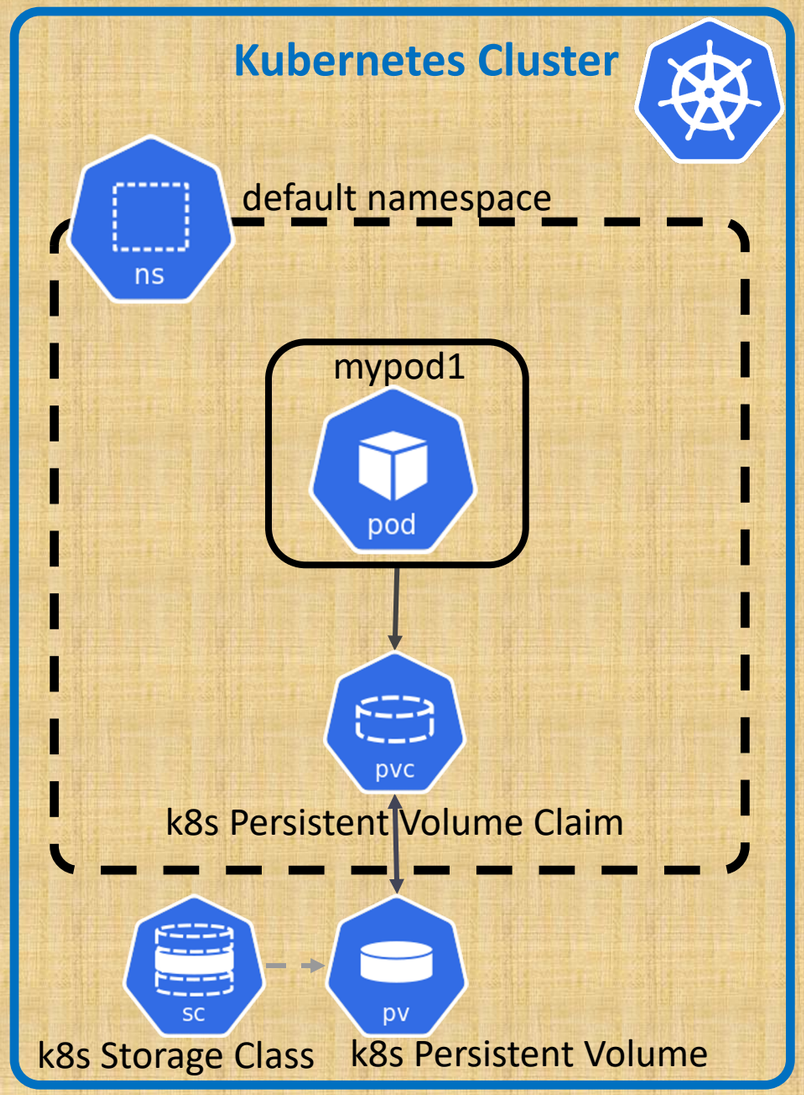
Author: Nho Luong

Skill: DevOps Engineer Lead

Kubernetes Storage Class

- **Storage Class:** Provides a way for administrators to define the classes of storage they offer
  - **standard-rwo:** Provides balanced disks
  - **premium-rwo:** Provides SSD disks

```
Kalyans-Mac-mini:google-kubernetes-engine kalyanreddy$ kubectl describe sc premium-rwo
Name:                  premium-rwo
IsDefaultClass:        No
Annotations:           components.gke.io/component-name=pdcsi,components.gke.io/component-version=0.13.2,components.gke.io/lay
er=addon
Provisioner:           pd.csi.storage.gke.io
Parameters:            type=pd-ssd
AllowVolumeExpansion:  True
MountOptions:          <none>
ReclaimPolicy:         Delete
VolumeBindingMode:     WaitForFirstConsumer
Events:                <none>
Kalyans-Mac-mini:google-kubernetes-engine kalyanreddy$ kubectl describe sc standard-rwo
Name:                  standard-rwo
IsDefaultClass:        Yes
Annotations:           components.gke.io/layer=addon,storageclass.kubernetes.io/is-default-class=true
Provisioner:           pd.csi.storage.gke.io
Parameters:            type=pd-balanced
AllowVolumeExpansion:  True
MountOptions:          <none>
ReclaimPolicy:         Delete
VolumeBindingMode:     WaitForFirstConsumer
Events:                <none>
Kalyans-Mac-mini:google-kubernetes-engine kalyanreddy$
```



Author: Nho Luong

Skill: DevOps Engineer Lead

Kubernetes Persistent Volume Claim

- **Persistent Volume Claim (PVC):** PVC objects request a specific size, access mode, and StorageClass for the PersistentVolume
- If a PersistentVolume that satisfies the request exists or can be provisioned, the PVC is bound to that PersistentVolume.
- **Access Modes**
  - **ReadWriteOnce:** The volume can be mounted as read-write by a single node.
  - **ReadOnlyMany:** The volume can be mounted read-only by many nodes.
  - **ReadWriteMany:** The volume can be mounted as read-write by many nodes.
    - PersistentVolume resources that are backed by Compute Engine persistent disks don't support this access mode.

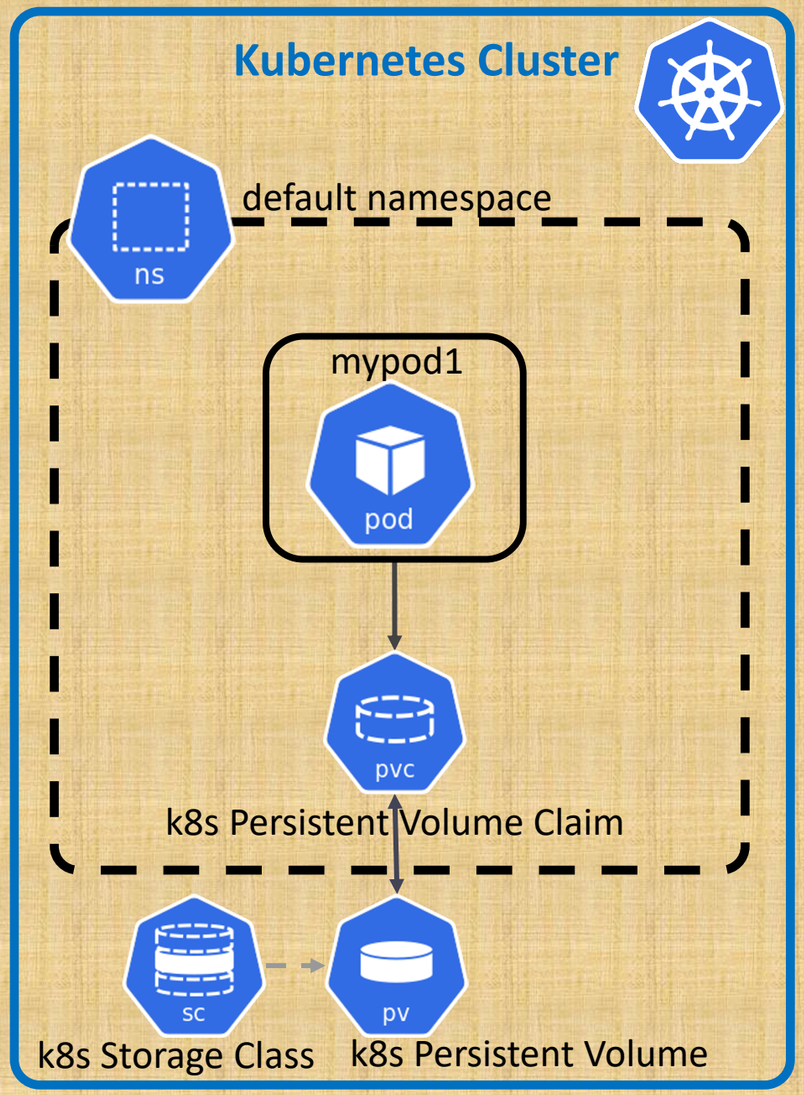
```yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mypvc1
spec:
  accessModes:
    - ReadWriteOnce
  storageClassName: standard-rwo
  resources:
    requests:
      storage: 1Gi
```

Author: Nho Luong

Skill: DevOps Engineer Lead

Kubernetes Persistent Volume

- **Persistent Volume:** In GKE, a PersistentVolume is typically backed by a Google persistent disk
- A PersistentVolume can be dynamically provisioned, we do not have to manually create and delete the backing storage
- PersistentVolume resources are cluster resources that exist independently of Pods.
  - Doesn’t impact due to cluster changes
  - Doesn’t impact due to pods deleted or recreated
- Persistent Volume will be provisioned based on configurations defined in Storage Class and Persistent Volume Claim
  - **Storage Class:** Storage type (pd-balanced, pd-ssd)
  - **PVC:**
    - **Size:** 1GB, 2GB
    - **Access Modes:** ReadWriteOnce, ReadOnlyMany, ReadWriteMany

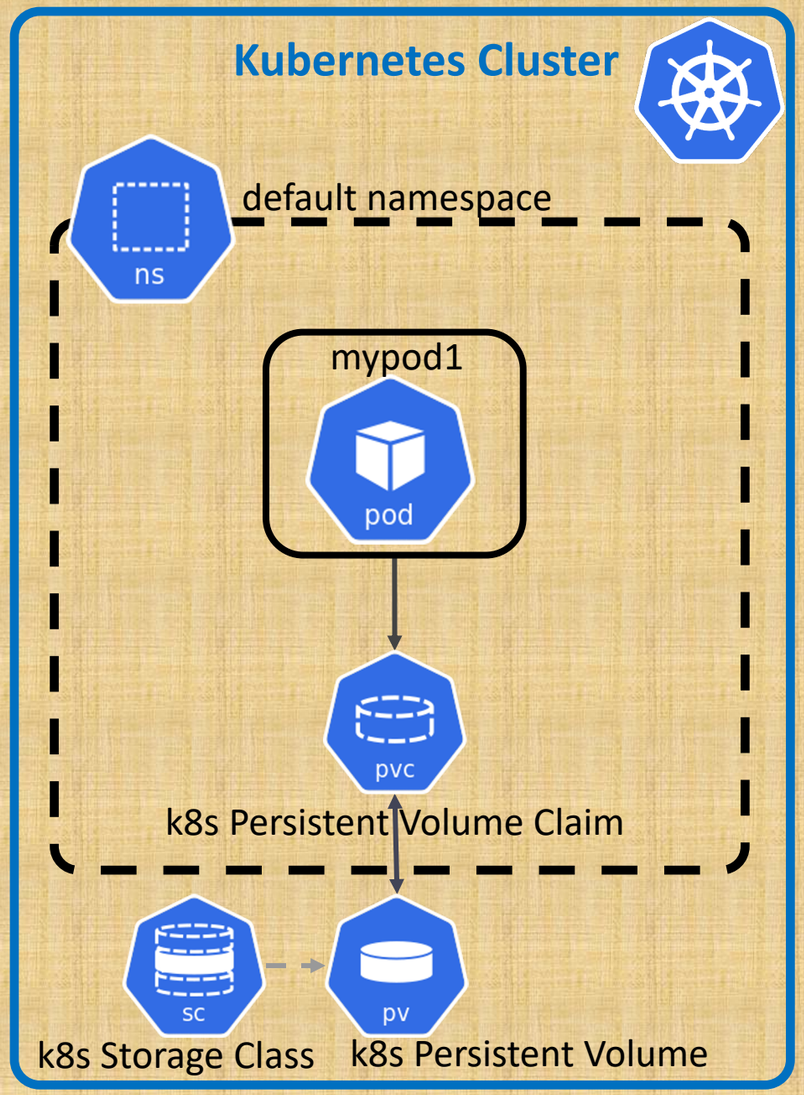


Author: Nho Luong

Skill: DevOps Engineer Lead

Kubernetes Persistent Volume

- **Resource Types [IMPORTANT]**
  - Storage Class and Persistent Volume or cluster level resources
  - Persistent Volume Claim (PVC) is a namespace level resource



Author: Nho Luong

Skill: DevOps Engineer Lead

GKE Cluster - Default Kubernetes Storage Classes

Persistent Disks
Default Storage Class

standard
kubernetes.io/gce-pd

Uses In-Tree Provisioner - Old

NOT RECOMMENDED

standard-rwo
(Balanced Disk)

premium-rwo
(SSD Disk)

Uses CSI Provisioner (Container Storage Interface – GKE PD CSI) – LATEST & GREATEST

RECOMMENDED

```
dkalyanreddy@cloudshell:~ (gcplearn9)$ kubectl get sc
NAME                 PROVISIONER             RECLAIMPOLICY   VOLUMEBINDINGMODE      ALLOWVOLUMEEXPANSION   AGE
premium-rwo          pd.csi.storage.gke.io   Delete          WaitForFirstConsumer   true                   27h
standard             kubernetes.io/gce-pd    Delete          Immediate              true                   27h
standard-rwo (default)   pd.csi.storage.gke.io   Delete          WaitForFirstConsumer   true                   27h
```

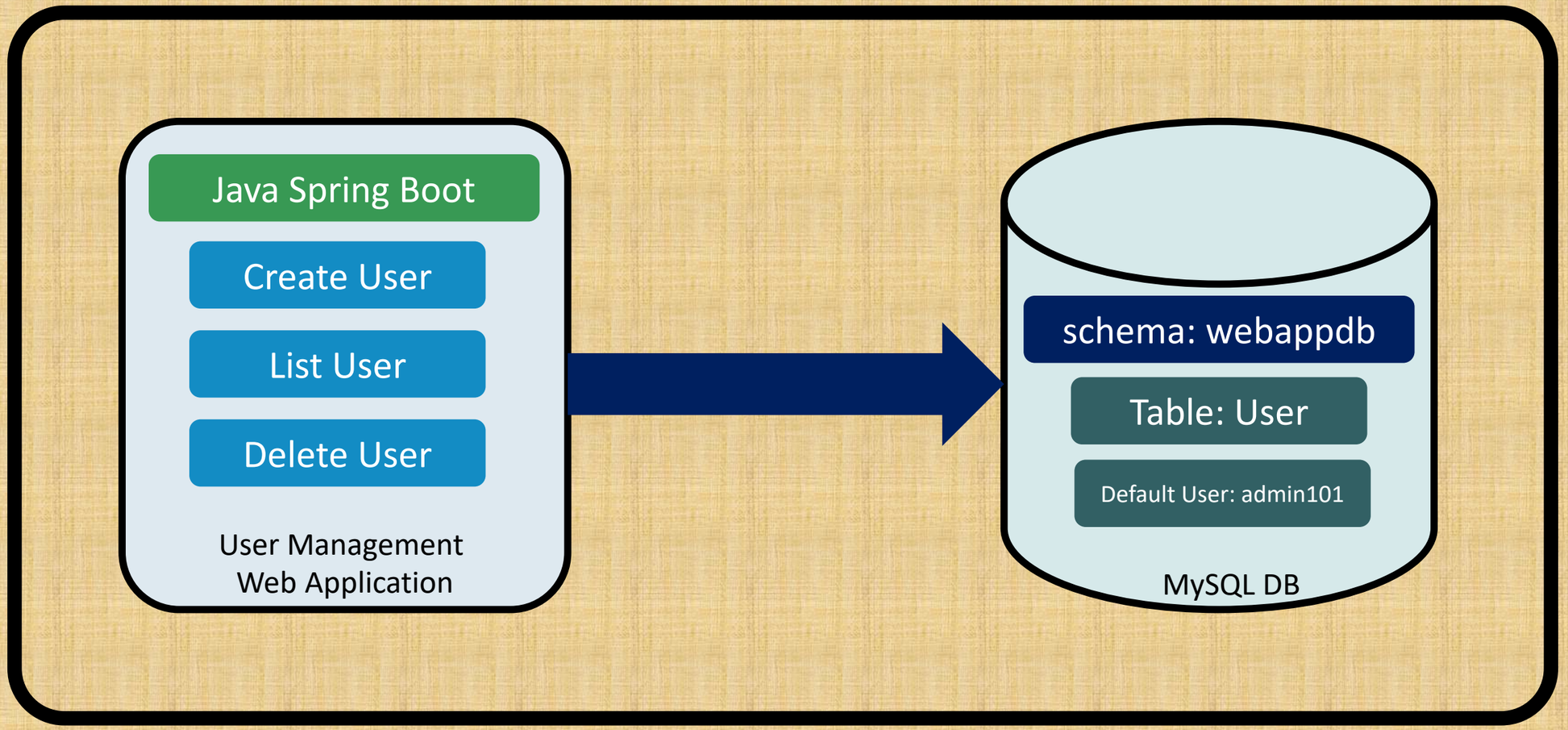
Cluster Feature should be enabled

Compute Engine persistent disk CSI Driver	Enabled

Author: Nho Luong

Skill: DevOps Engineer Lead

User Management Web Application - Architecture

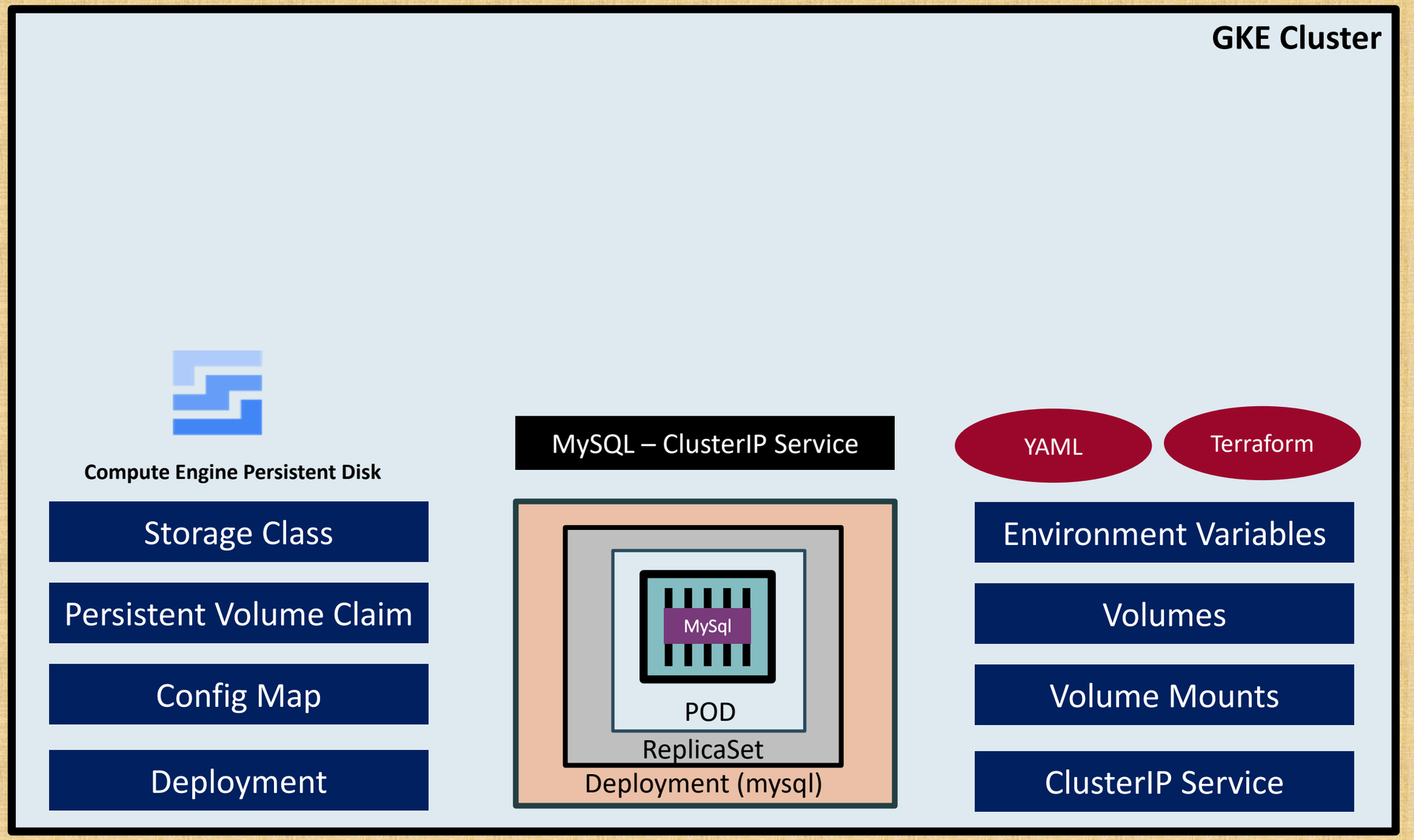


Author: Nho Luong

Skill: DevOps Engineer Lead

GKE Storage with Compute Engine Persistence Disks

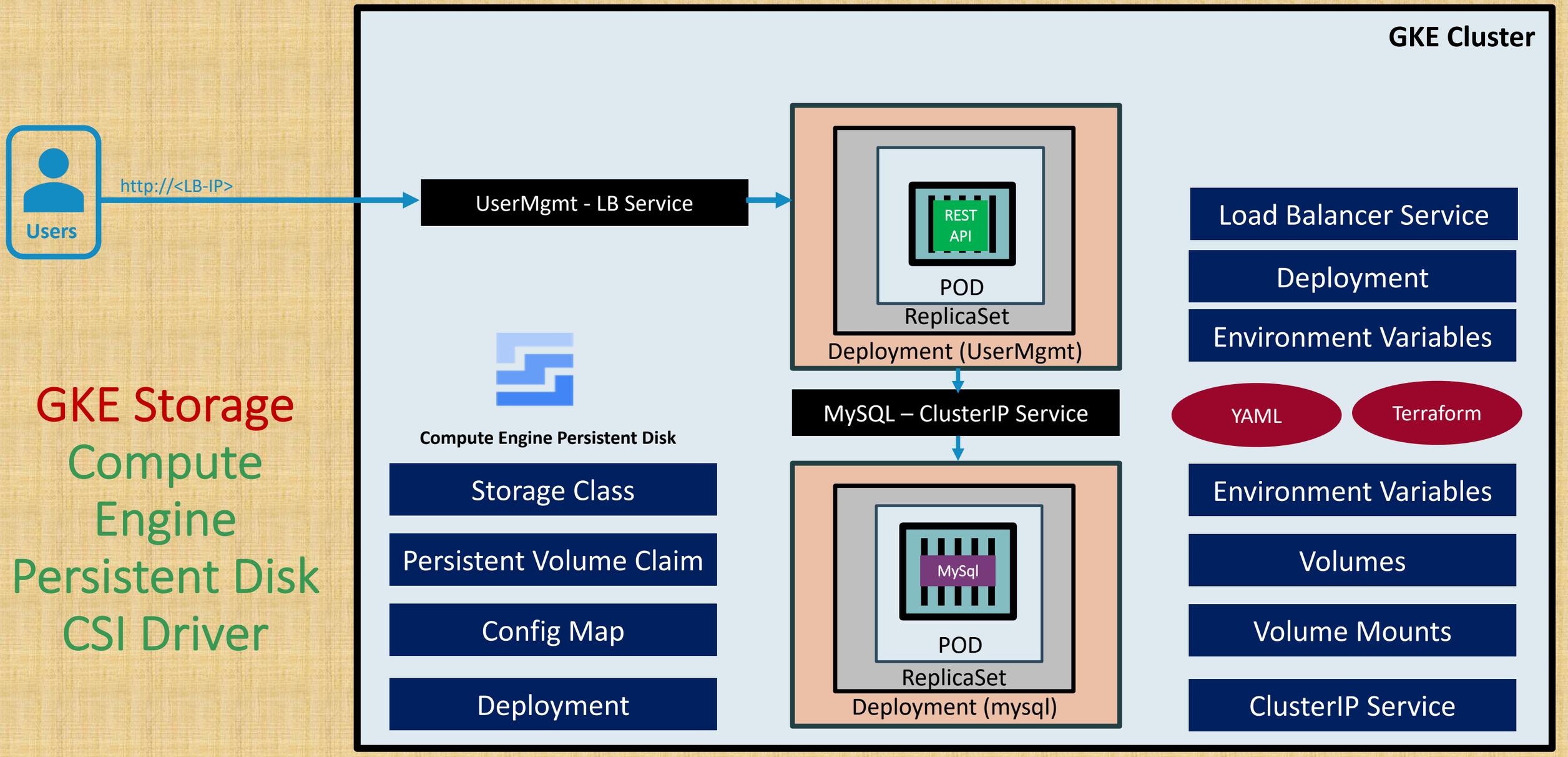
GKE Storage
Compute Engine Persistent Disk CSI Driver



Author: Nho Luong

Skill: DevOps Engineer Lead

GKE Storage with Compute Engine Persistence Disks

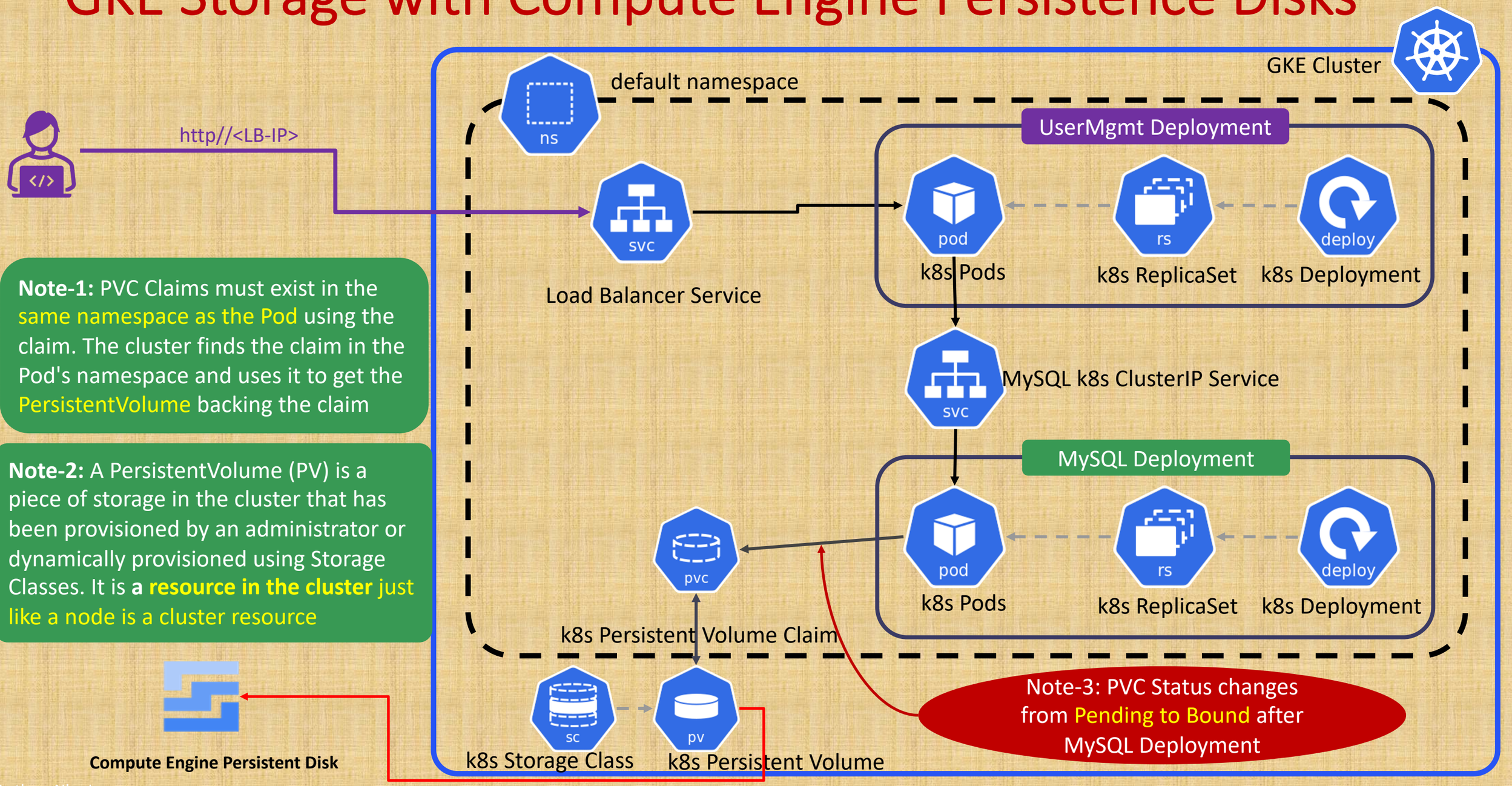


GKE Storage
Compute Engine Persistent Disk CSI Driver

Author: Nho Luong

Skill: DevOps Engineer Lead

GKE Storage with Compute Engine Persistence Disks



Note-1: PVC Claims must exist in the same namespace as the Pod using the claim. The cluster finds the claim in the Pod's namespace and uses it to get the PersistentVolume backing the claim

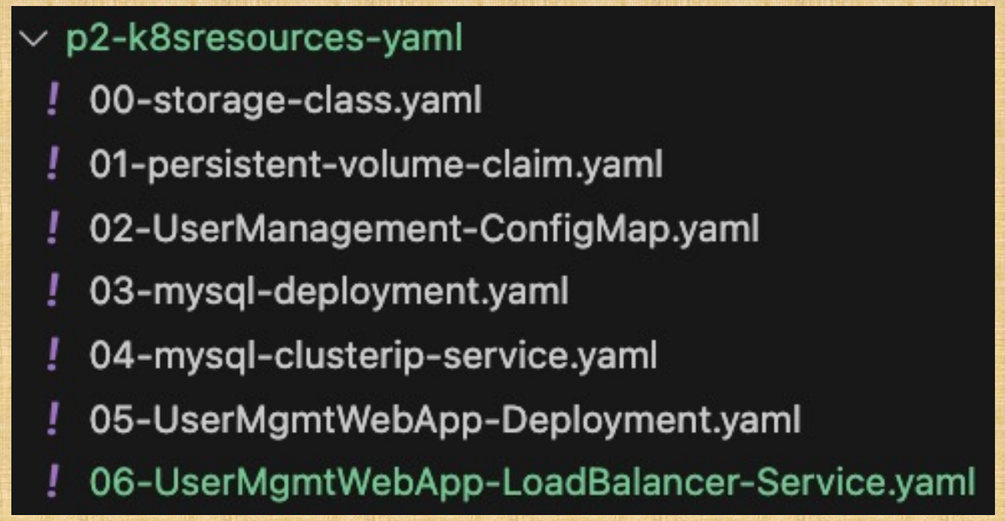
Note-2: A PersistentVolume (PV) is a piece of storage in the cluster that has been provisioned by an administrator or dynamically provisioned using Storage Classes. It is **a resource in the cluster** just like a node is a cluster resource

Author: Nho Luong

Skill: DevOps Engineer Lead

Project-2: YAML Manifests

What are we going to learn?



Demo: Project-2: YAML Manifests

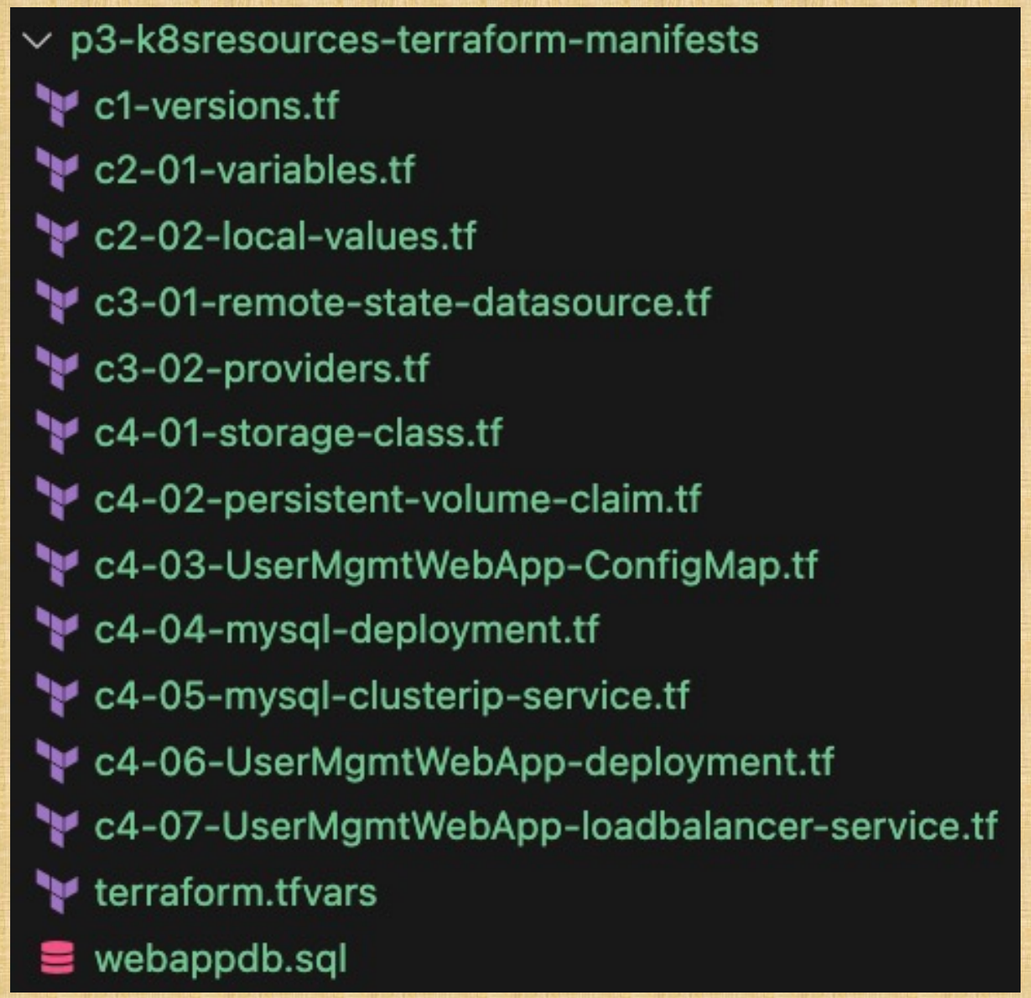
00	Custom Kubernetes Storage Class
01	Kubernetes Persistent Volume Claim
02	Kubernetes ConfigMap
03	Kubernetes Deployment for MySQL Database
04	Kubernetes ClusterIP Service with clusterIP:None
05	User Management Web Application Kubernetes Deployment
06	User Management Web Application Kubernetes Load Balancer Service

Author: Nho Luong

Skill: DevOps Engineer Lead

Project-3: Terraform Manifests

What are we going to learn?



Demo: Project-3: Terraform Manifests

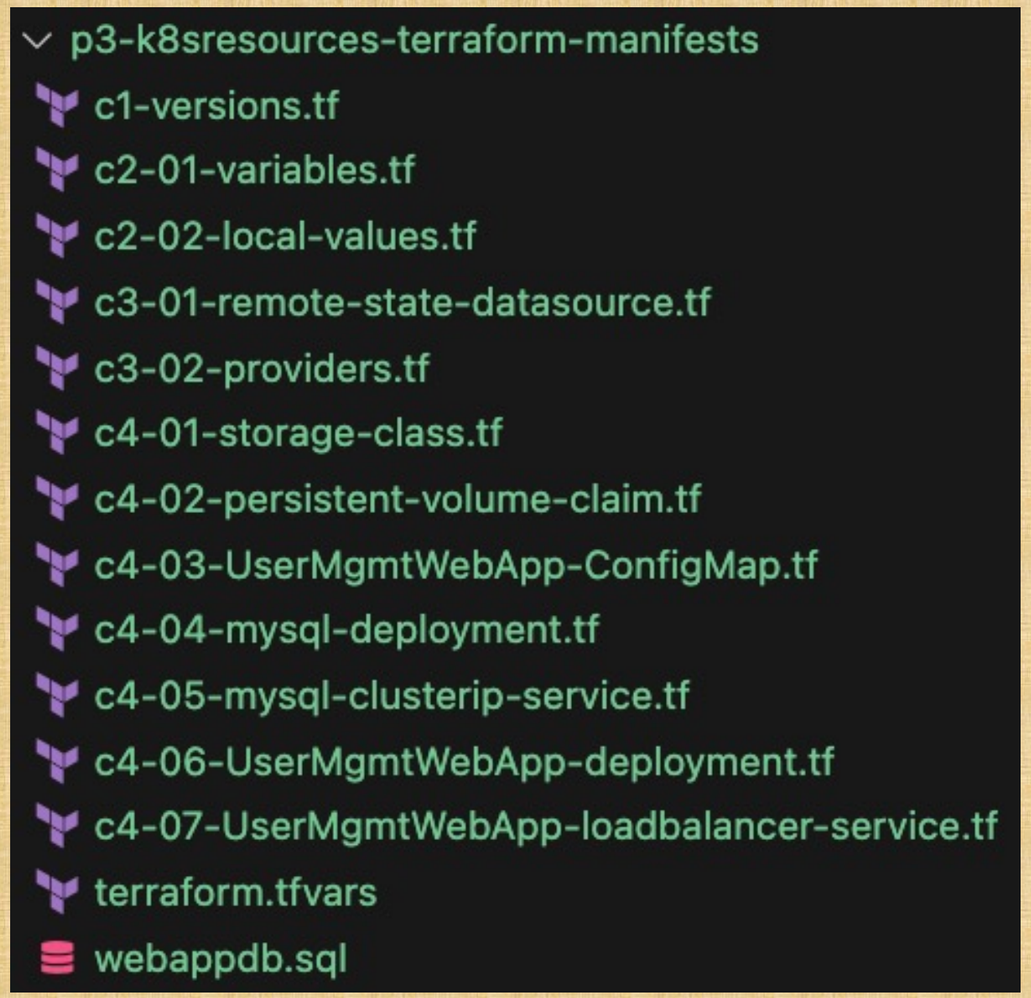
NC	No Changes from C1 to C3-02
c4-01	Custom Kubernetes Storage Class
c4-02	Kubernetes Persistent Volume Claim
c4-03	Kubernetes ConfigMap
c4-04	Kubernetes Deployment for MySQL Database
c4-05	Kubernetes ClusterIP Service with clusterIP:None
c4-06	User Management Web Application Kubernetes Deployment

Author: Nho Luong

Skill: DevOps Engineer Lead

Project-3: Terraform Manifests

What are we going to learn?



Demo: Project-3: Terraform Manifests

NC	No Changes from C1 to C3-02
c4-07	User Management Web Application Kubernetes Load Balancer Service
SQL	webappdb.sql file to create base database schema

Author: Nho Luong

Skill: DevOps Engineer Lead

Demo-18

Google Kubernetes Engine

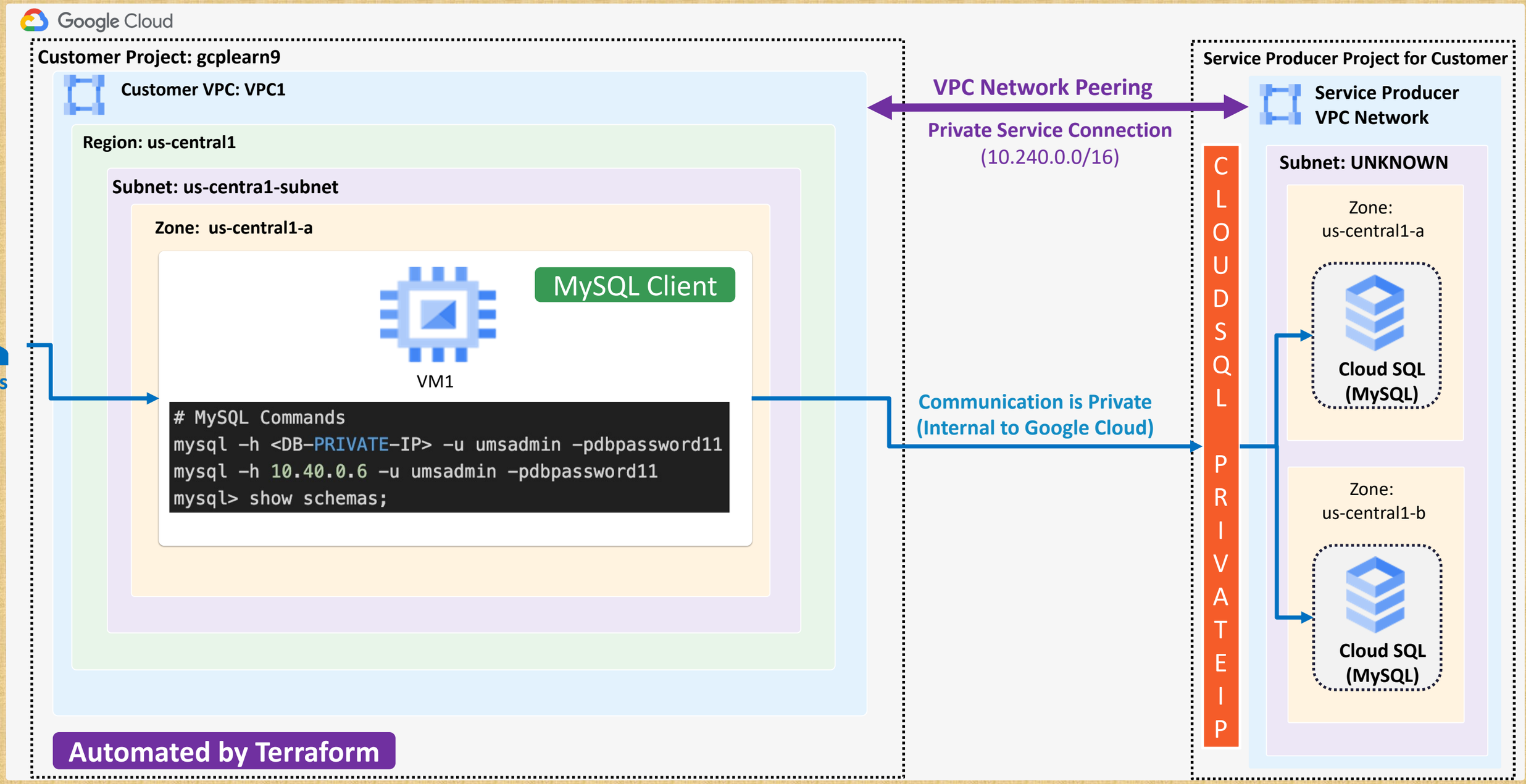
Cloud SQL MySQL Database as Storage for Applications Deployed on GKE Cluster

Demo: GKE Autopilot cluster + Cloud SQL MySQL DB

Author: Nho Luong

Skill: DevOps Engineer Lead

GCP Cloud SQL Private IP



Author: Nho Luong

Skill: DevOps Engineer Lead

Project-2: Terraform Manifests

What are we going to learn?

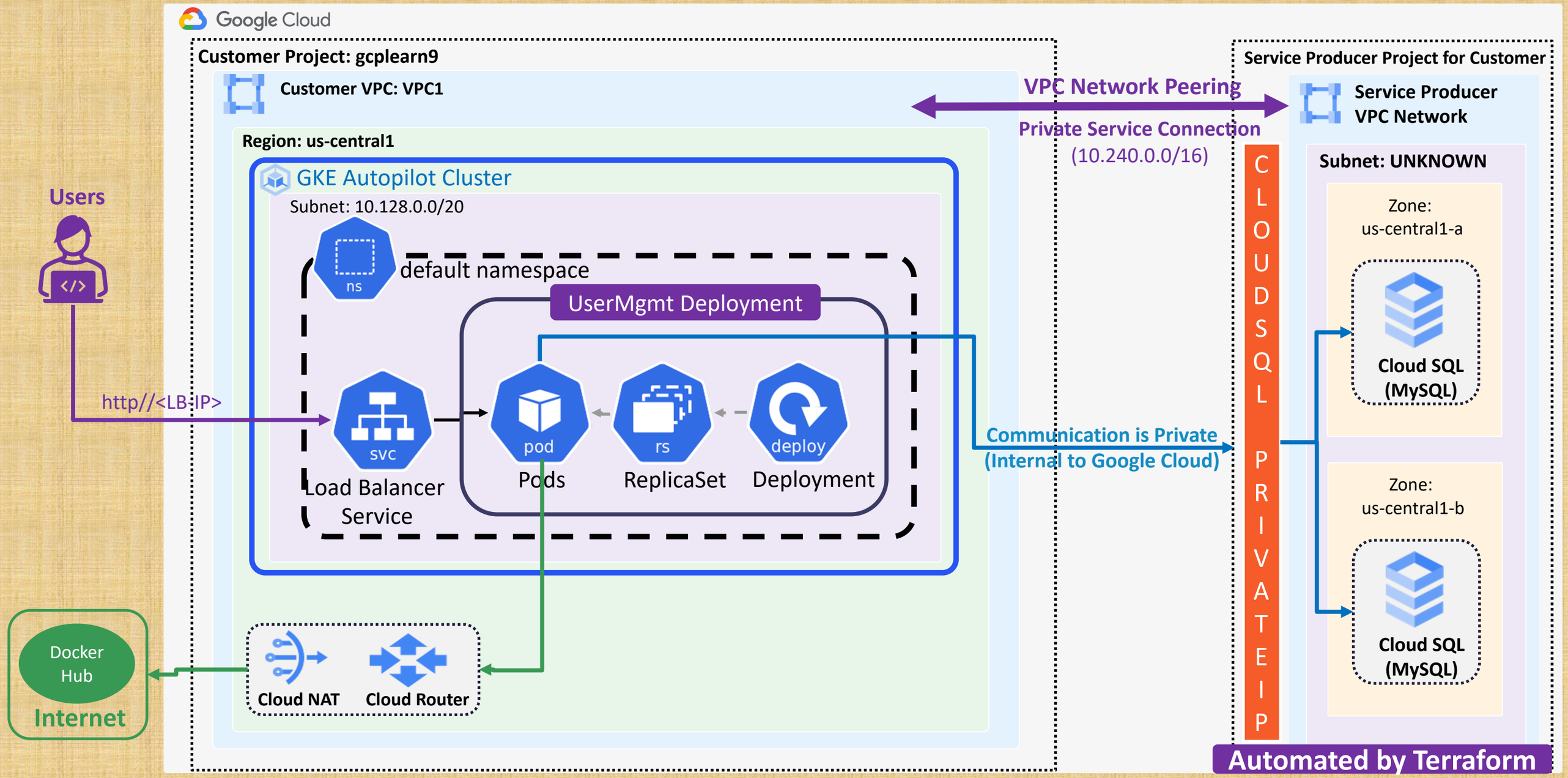
Demo: Project-2: Cloud SQL Private Database

c1	Define Remote Backend in Cloud Storage Bucket for this demo
c2-01	Define Cloud SQL MySQL Database version variable
c3-01	Define Remote State data source to access GKE cluster from Project-1
c3-02	Create VPC Network peering between our VPC and Google Managed Project VPC
c4-01	Create Cloud SQL Terraform Resources
c4-02	Define Cloud SQL Outputs mainly Database IP, so that it can accessed in Project-3 UMS Web App
C5	VM Instance to test the DB created is accessible privately from our customer VPC

Author: Nho Luong

Skill: DevOps Engineer Lead

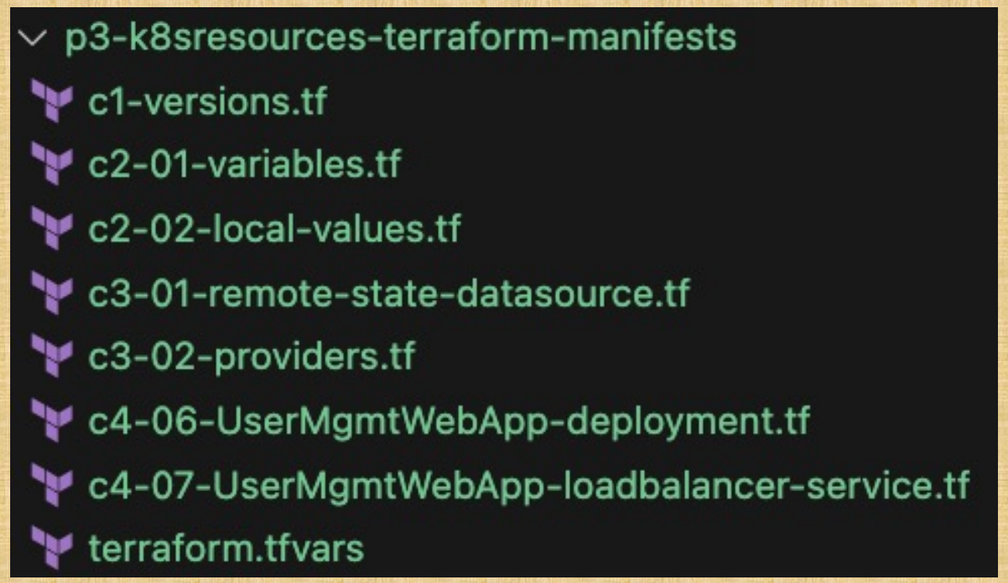
GCP Cloud SQL Private IP + GKE Autopilot Cluster + User Management Web Application



Author: Nho Luong

Skill: DevOps Engineer Lead

Project-3: Terraform Manifests



What are we going to learn?

Demo: Project-3: UMS Web Application using Cloud SQL MySQL Database

c1	Define Remote Backend in Cloud Storage Bucket for this demo
c2-01	Define Input variable
c3-01	Define Remote State data source to access GKE cluster from Project-1 and Cloud SQL IP from Project-2
c3-02	Define Kubernetes Provider
c4-06	UMS WebApp k8s Deployment pointing to Cloud SQL MySQL Database
c4-07	UMS WebApp Load Balancer Service

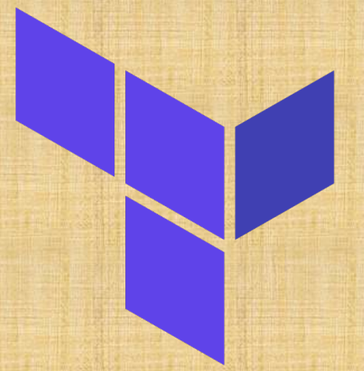
Author: Nho Luong

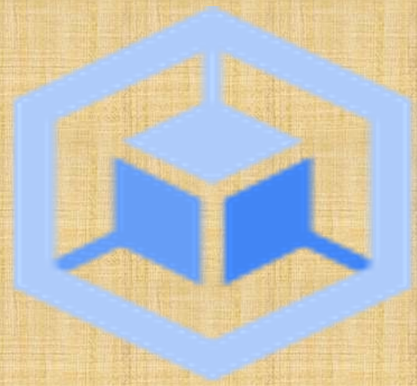
Skill: DevOps Engineer Lead

Demo-19

Google Kubernetes Engine

Cloud Storage FUSE CSI Driver





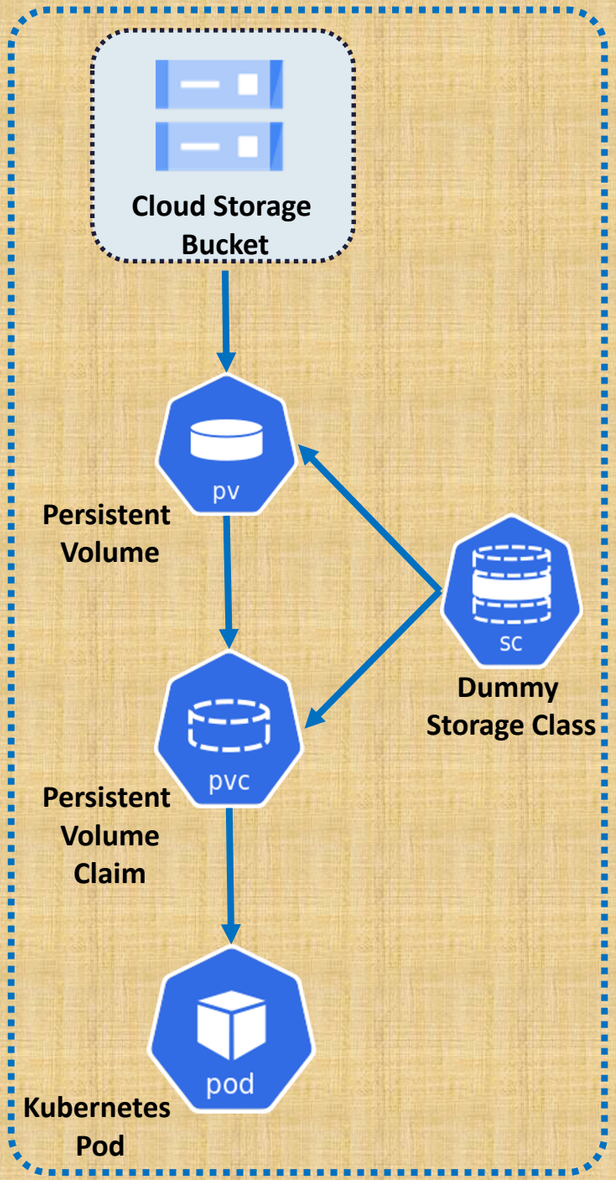
Demo: GKE Autopilot cluster + Cloud Storage Buckets

Author: Nho Luong

Skill: DevOps Engineer Lead

Cloud Storage FUSE CSI Driver

- **FUSE: Filesystem in Userspace**
  - Fuse is an interface primarily used to export a filesystem to the Linux kernel
- **Cloud Storage FUSE CSI Driver**
  - Allow you to mount Cloud Storage Buckets as a file system
  - Applications can access the objects in a bucket using common File IO operations (e.g. open, read, write, close) rather than using cloud-specific APIs
  - In short, it lets you use the Kubernetes API to consume pre-existing Cloud Storage buckets as volumes
  - Driver can be deployed automatically just using enable/disable options
  - Supports GKE Standard and Auto-pilot clusters

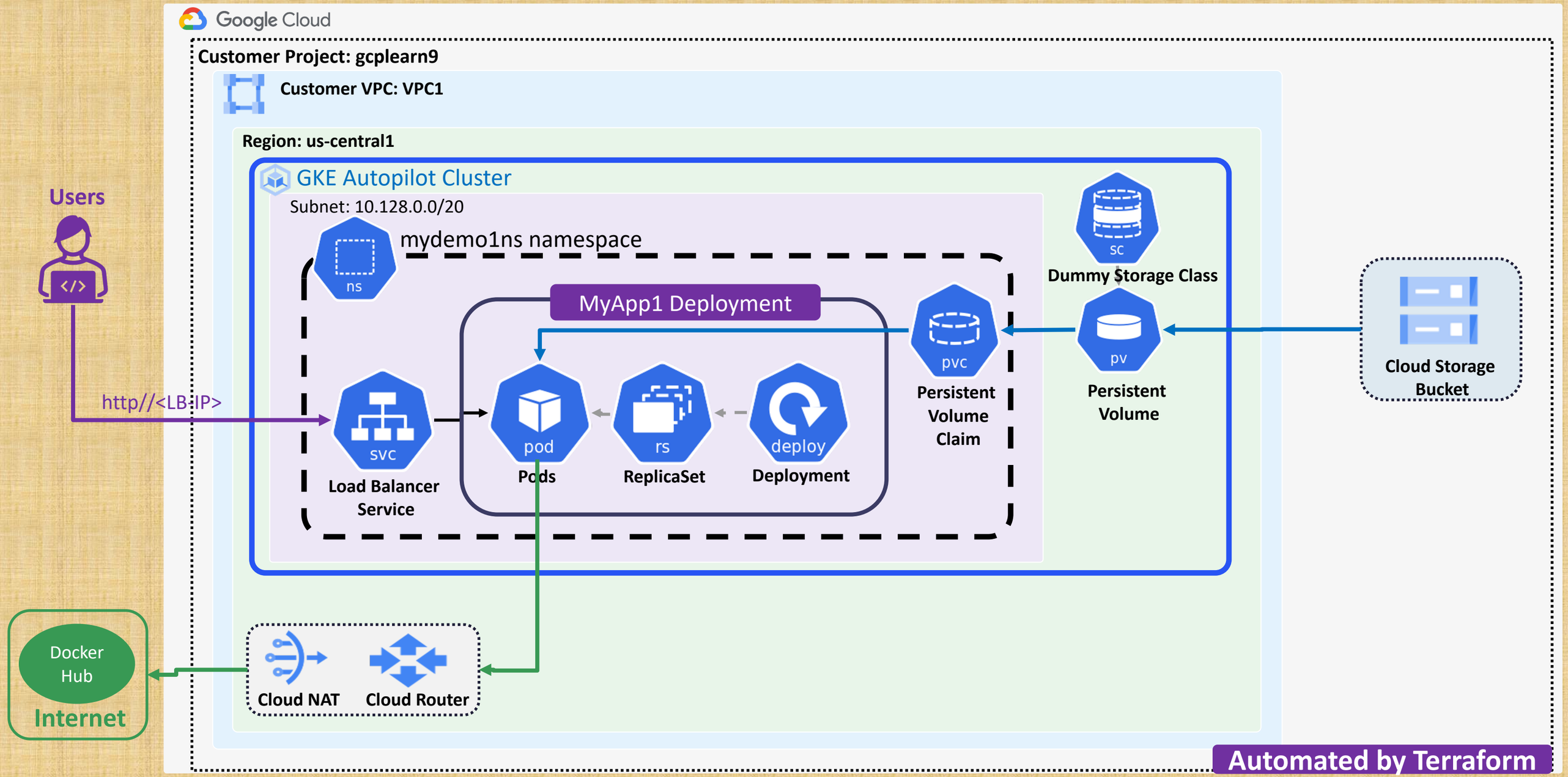


Reference: https://cloud.google.com/kubernetes-engine/docs/how-to/persistent-volumes/cloud-storage-fuse-csi-driver

Author: Nho Luong

Skill: DevOps Engineer Lead

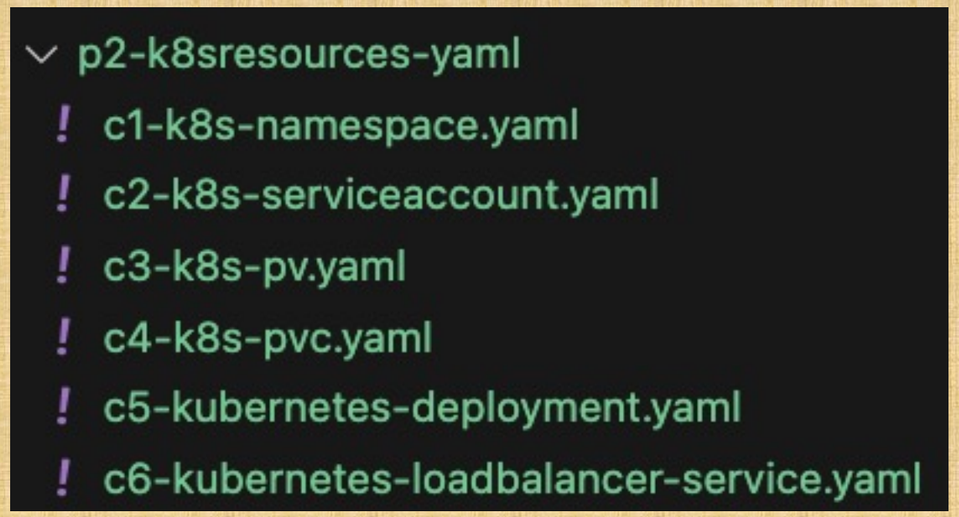
GKE Autopilot Cluster + Cloud Storage Bucket as Volume Mount



Author: Nho Luong

Skill: DevOps Engineer Lead

Project-2: YAML Manifests



Manual Steps (Required)

Cloud Storage Bucket

IAM Policy Binding

What are we going to learn?

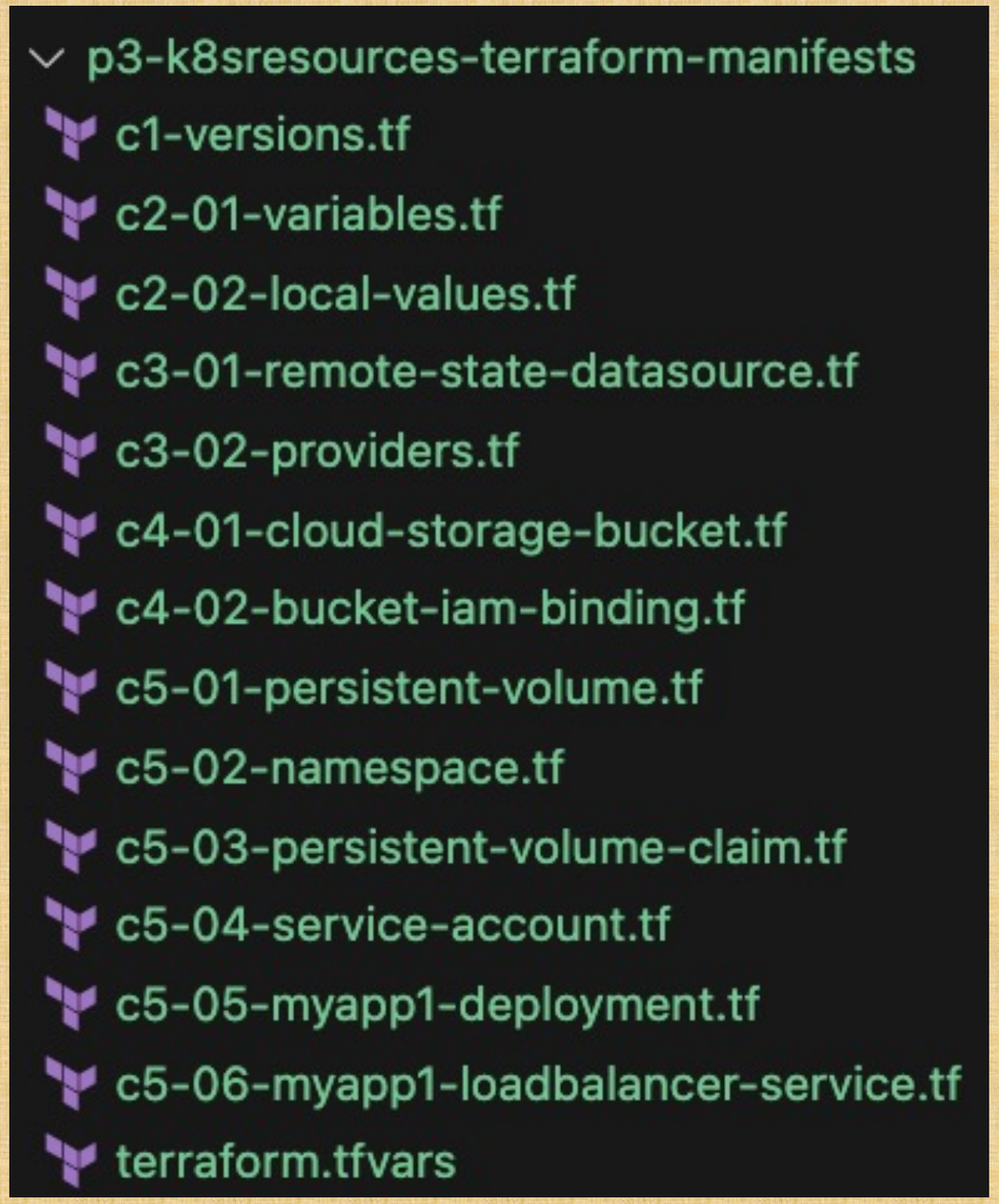
Demo: Project-2: GKE Autopilot cluster + Cloud Storage Buckets (GCS FUSE CSI)

c1	Kubernetes Namespace
c2	Kubernetes Service Account
c3	Kubernetes Persistent Volume using Cloud Storage Bucket
c4	Kubernetes Persistent Volume Claim
c5	Kubernetes Deployment mounted with Cloud Storage bucket as volume
c6	Kubernetes Load Balancer Service

Author: Nho Luong

Skill: DevOps Engineer Lead

Project-3: Terraform Manifests



What are we going to learn?

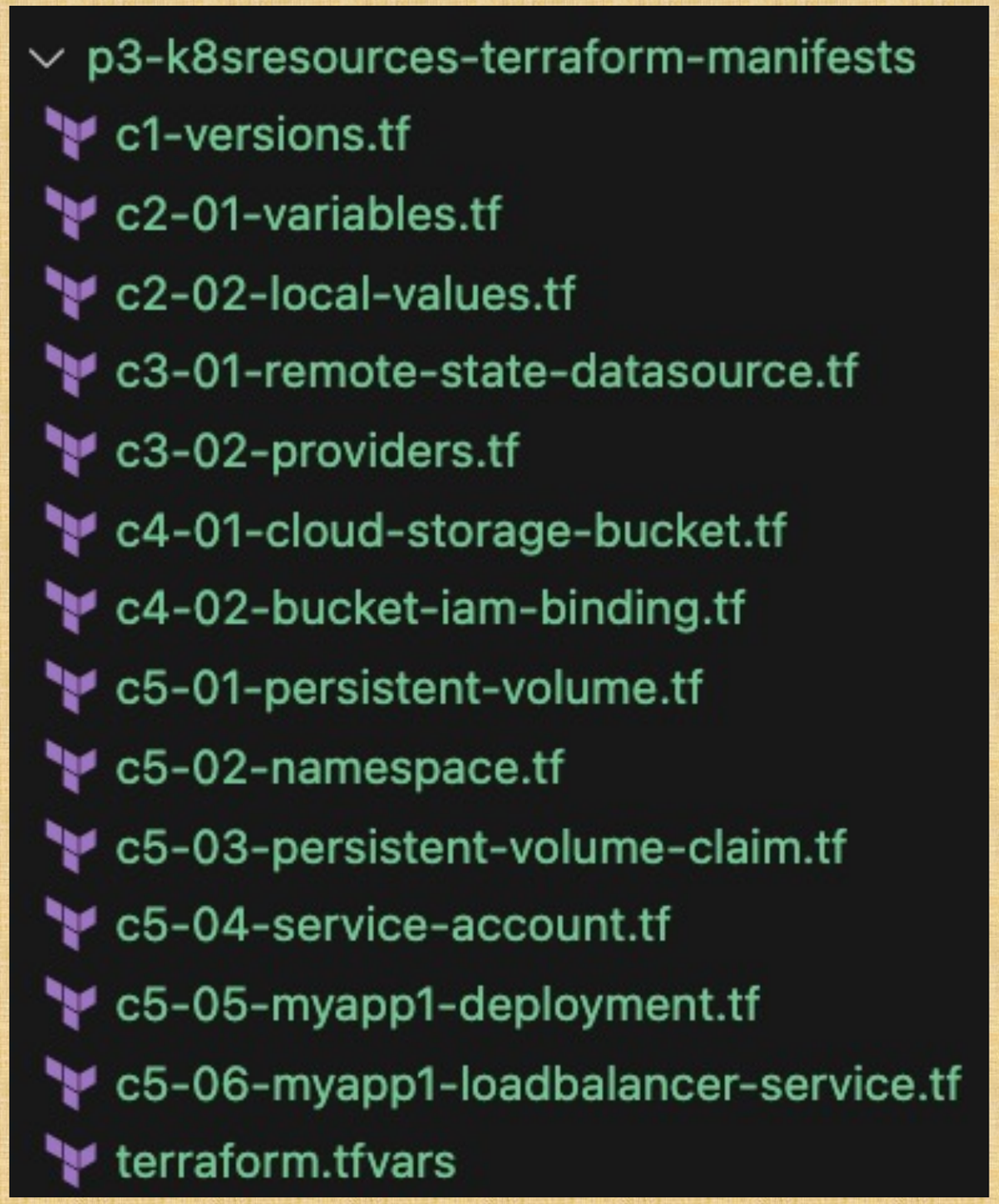
Demo: Project-3: GKE Autopilot cluster + Cloud Storage Buckets (GCS FUSE CSI)

NC	No changes c1 to c3
c4-01	Create Cloud Storage Bucket
c4-02	Cloud Storage Bucket IAM Binding
c5-01	Kubernetes Persistent Volume using Cloud Storage Bucket
c5-02	Kubernetes Namespace
c5-03	Kubernetes Persistent Volume Claim
c5-04	Kubernetes Service Account

Author: Nho Luong

Skill: DevOps Engineer Lead

Project-3: Terraform Manifests



What are we going to learn?

Demo: Project-3: GKE Autopilot cluster + Cloud Storage Buckets (GCS FUSE CSI)

c5-05 — Kubernetes Deployment mounted with Cloud Storage bucket as volume

c5-06 — Kubernetes Load Balancer Service

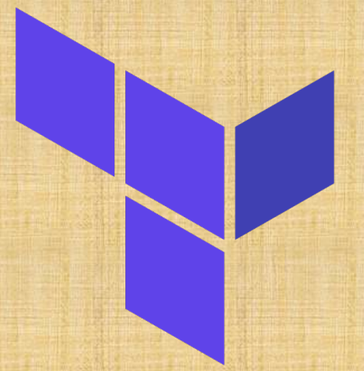
Author: Nho Luong

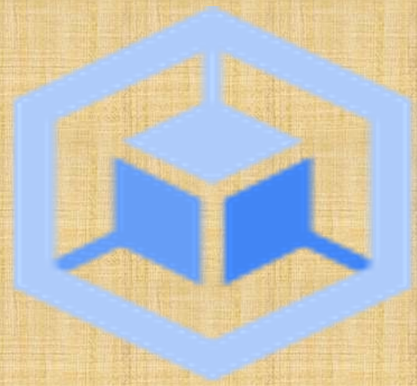
Skill: DevOps Engineer Lead

Demo-20

Google Kubernetes Engine

File Store CSI Driver





Demo: GKE Autopilot cluster + File Store

Author: Nho Luong

Skill: DevOps Engineer Lead

File Store CSI Driver - Storage Classes

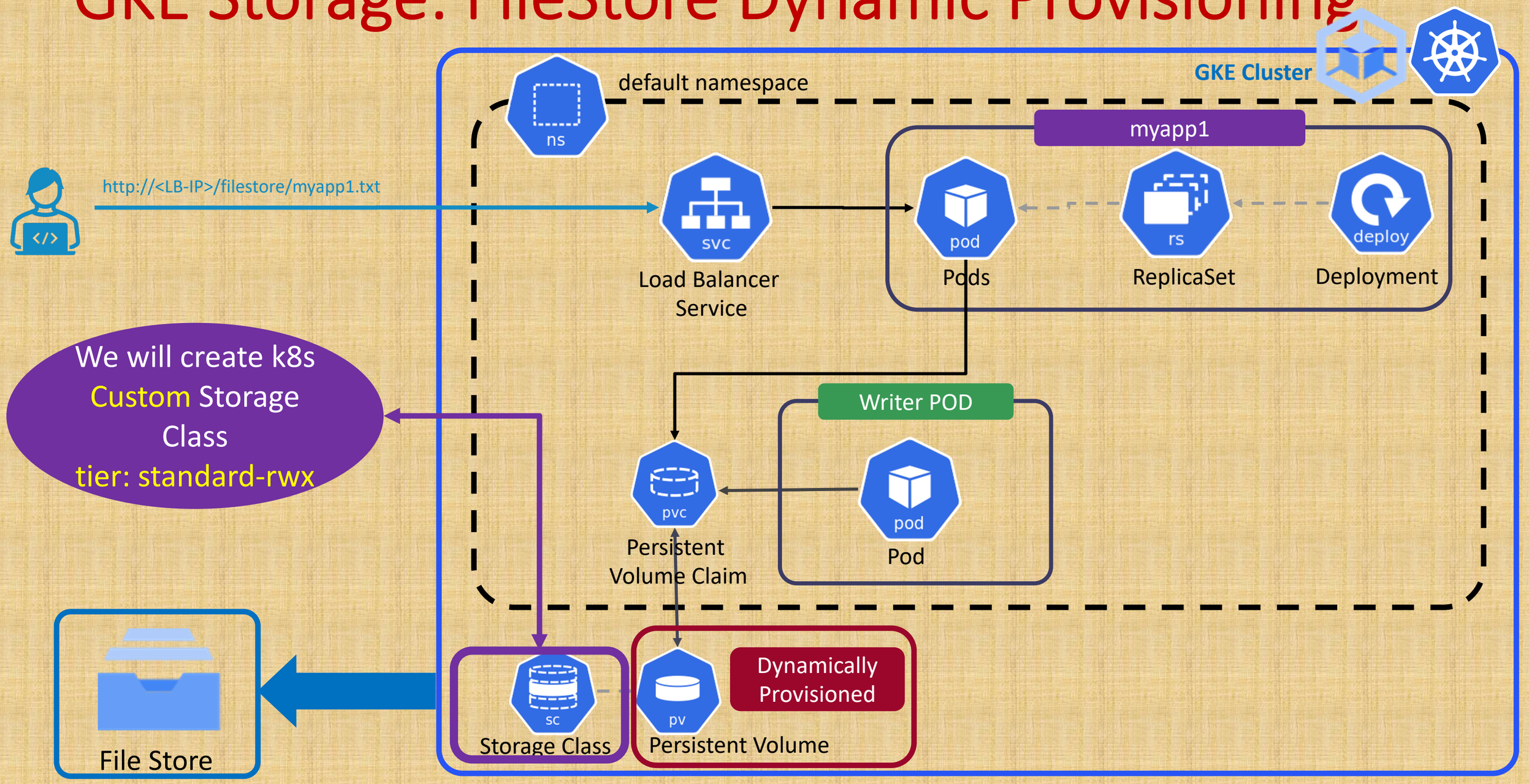
Service tier	GKE StorageClass	Number of shares per instance	GKE PVC size	Filestore capacity	Incremental change	Deployment	Access mode
Basic HDD **Standard**	`standard-rwx`	One	1 TiB to 64 TiB	1 TiB to 64 TiB	1 GiB	Zonal	read/write many
Basic SSD **Premium**	`premium-rwx`	One	2.5 TiB to 64 TiB	2.5 TiB to 64 TiB	1 GiB	Zonal	read/write many
Enterprise **Enterprise**	`enterprise-rwx`	One	1 TiB to 10 TiB	1 TiB to 10 TiB	256 GiB	Regional	read/write many
Enterprise with the multishares feature **Enterprise**	`enterprise-multishare-rwx`	Up to 80	10 GiB to 1 TiB	Pool of Filestore instances, each 1 TiB to 10 TiB in size Each Filestore instance hosts up to 80 PVs (shares), each 10 GiB to 1 TiB in size	256 GiB per instance 1 GiB per PVC (share)	Regional	read/write many

Reference: https://cloud.google.com/filestore/docs/multishares#storageclasses_and_dynamic_volume_provisioning

Author: Nho Luong

Skill: DevOps Engineer Lead

GKE Storage: FileStore Dynamic Provisioning

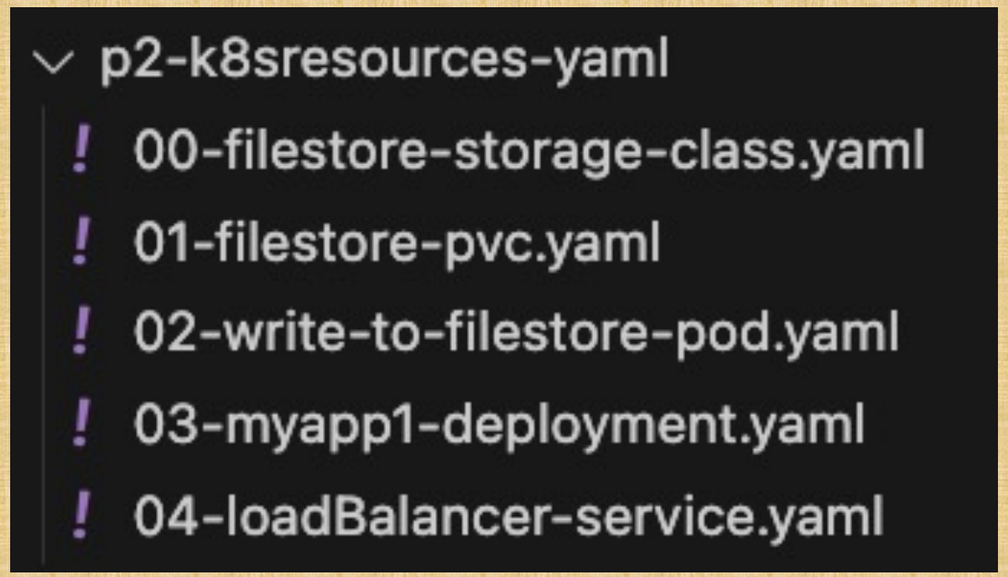


Author: Nho Luong

Skill: DevOps Engineer Lead

Project-2: YAML Manifests

What are we going to learn?



Demo: Project-2: GKE Autopilot cluster + File Store

00	Kubernetes Custom Storage Class
01	Kubernetes Persistent Volume Claim
02	Kubernetes Pod - Write to File store
03	Kubernetes MyApp1 Deployment (Read content from file store by mounting it)
04	Kubernetes Load Balancer Service

Author: Nho Luong

Skill: DevOps Engineer Lead

Project-3: Terraform Manifests

What are we going to learn?

Demo: Project-3: GKE Autopilot cluster + File Store

NC	No changes c1 to c3
c4-01	Create Custom Storage Class
c4-02	Kubernetes Persistent Volume Claim
c5-01	Kubernetes Pod – Write to File Store instance
c5-02	Kubernetes MyApp1 Deployment (Read content from file store by mounting)
c5-03	Kubernetes Load Balancer Service

Author: Nho Luong

Skill: DevOps Engineer Lead

Kubernetes Gateway API

Introduction

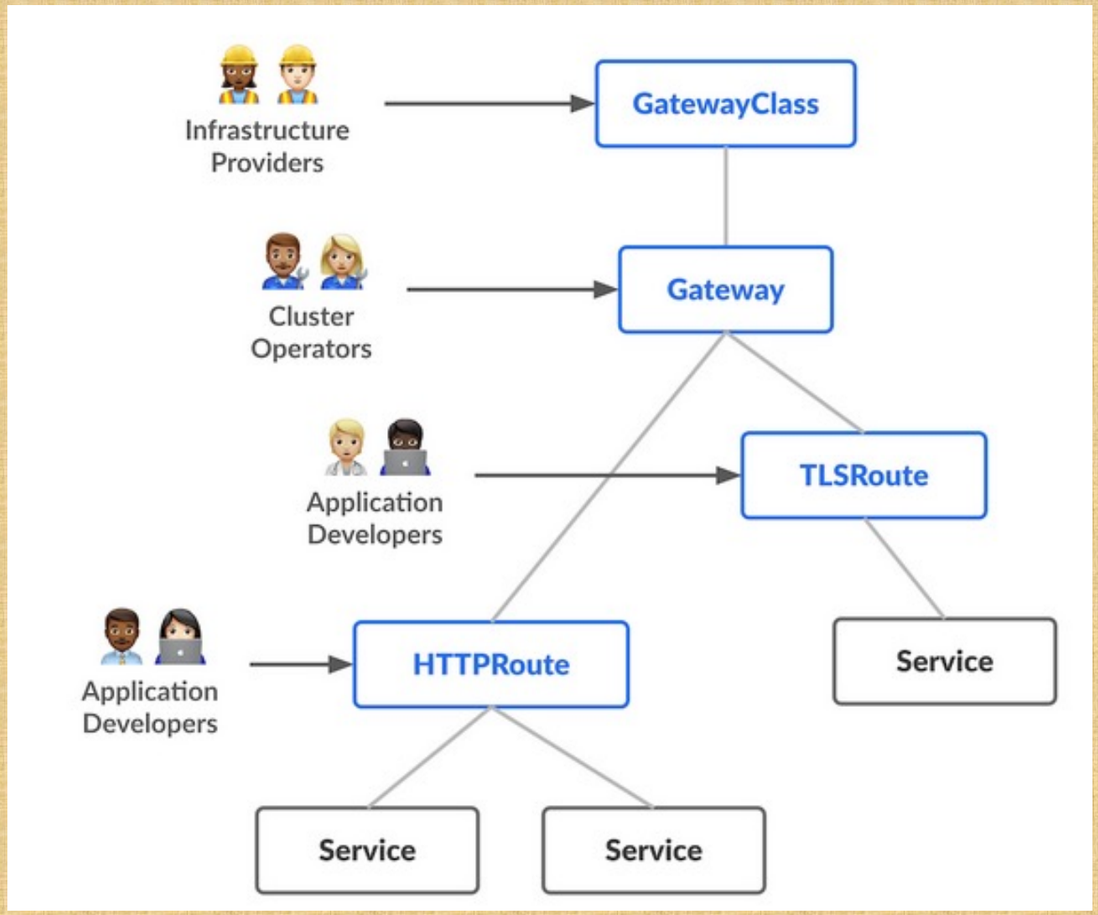
Author: Nho Luong

Skill: DevOps Engineer Lead

Kubernetes Gateway API

- **Why do we need to use Gateway API ?**
  - Kubernetes Ingress is frozen and new features are being added to the Gateway API
  - Gateway API is going to be the replacement for Ingress going forward

- **What is Gateway API ?**
  - Gateway API is an advanced version of Ingress
  - Ingress supports only Layer7 protocols (HTTP, HTTPS)
  - Gateway API supports
    - **Layer7 Protocols:** HTTP and HTTPS
    - **Layer4 Protocols:** TCP and UDP
    - **gRPC Protocol**



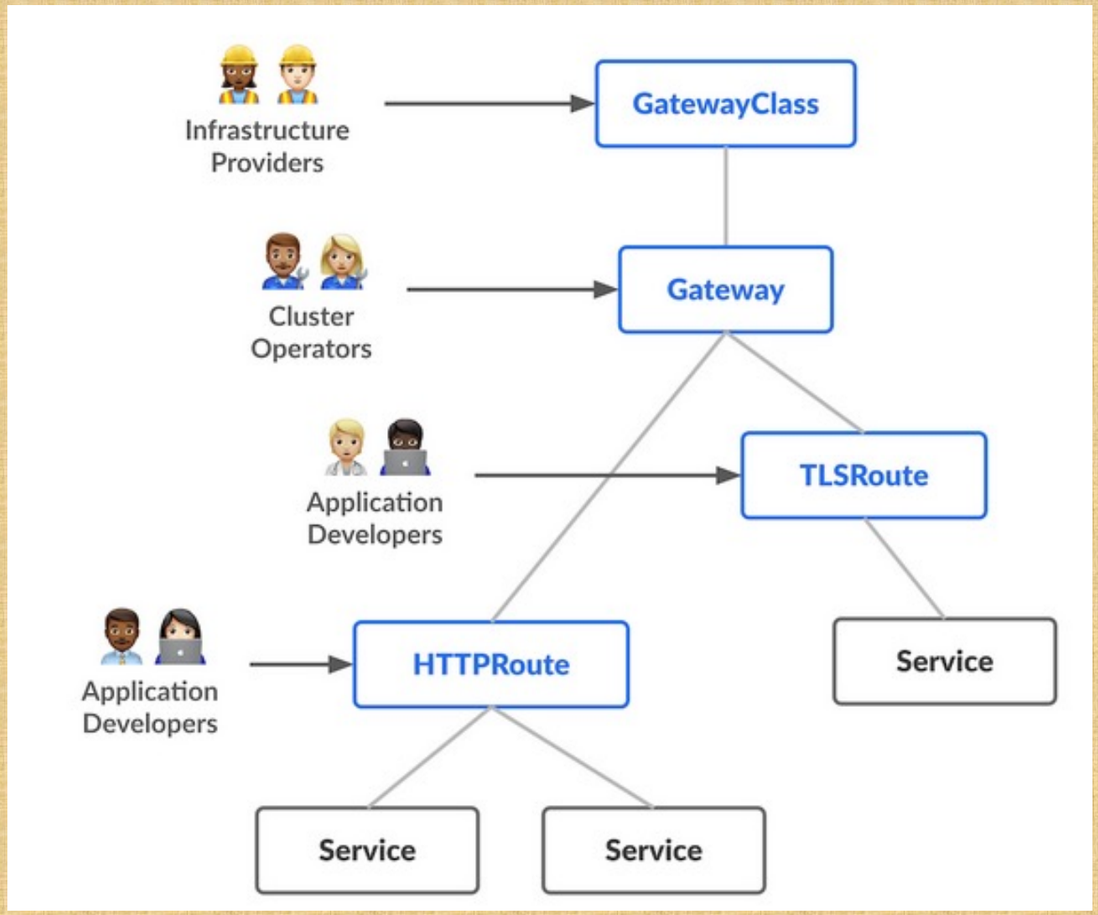
Reference: https://gateway-api.sigs.k8s.io/

Author: Nho Luong

Skill: DevOps Engineer Lead

Kubernetes Gateway API

- **Role-oriented Design:** The Gateway API is a role-oriented resource model
  - Users with specific roles can manage those specific Gateway API Resources
  - **Example:**
    - Infrastructure Providers can manage GatewayClass
    - Cluster Operators can manage Gateways
    - Application Developers can define TLSRoute, HTTPRoute



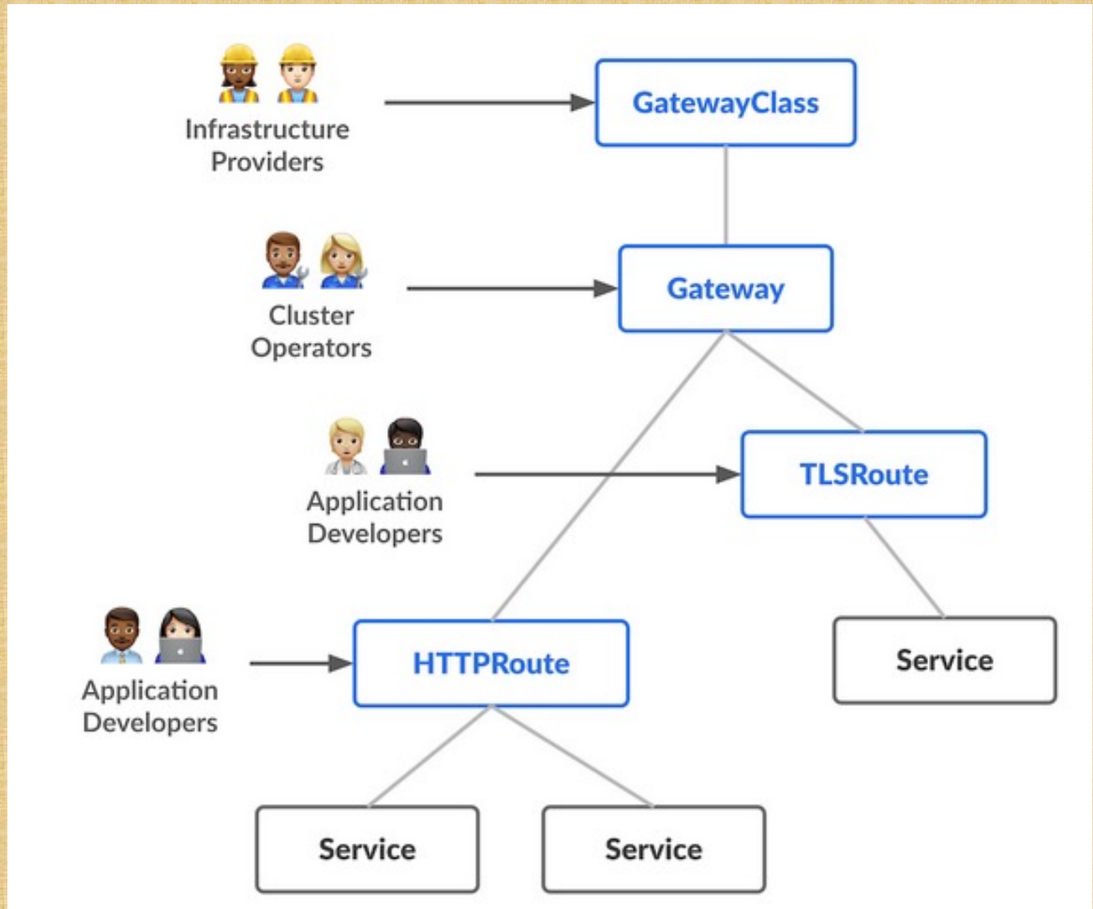
Reference: https://gateway-api.sigs.k8s.io/

Author: Nho Luong

Skill: DevOps Engineer Lead

Kubernetes Gateway API Terminology

- **GatewayClass**
  - Template for creating load balancers in a cluster.
  - It's a cluster-scoped resource
- **Gateway**
  - Defines where and how the load balancers listen for traffic.
  - Cluster operators create Gateways in their clusters based on a GatewayClass.
- **HTTPRoute**
  - Defines protocol-specific rules for routing requests from a Gateway to Kubernetes services.
  - Application developers create HTTPRoutes to expose their HTTP applications using Gateways.



Reference: https://gateway-api.sigs.k8s.io/

Author: Nho Luong

Skill: DevOps Engineer Lead

Demo-01

Google Kubernetes Engine

Gateway API Basics

Cloud Application Load Balancer

Demo: GKE Autopilot cluster + Gateway API (Cloud Application Load Balancer)

Author: Nho Luong

Skill: DevOps Engineer Lead

Kubernetes Gateway API

- In GKE, Gateway API is implemented using GKE Gateway controller
- **GKE Gateway classes**

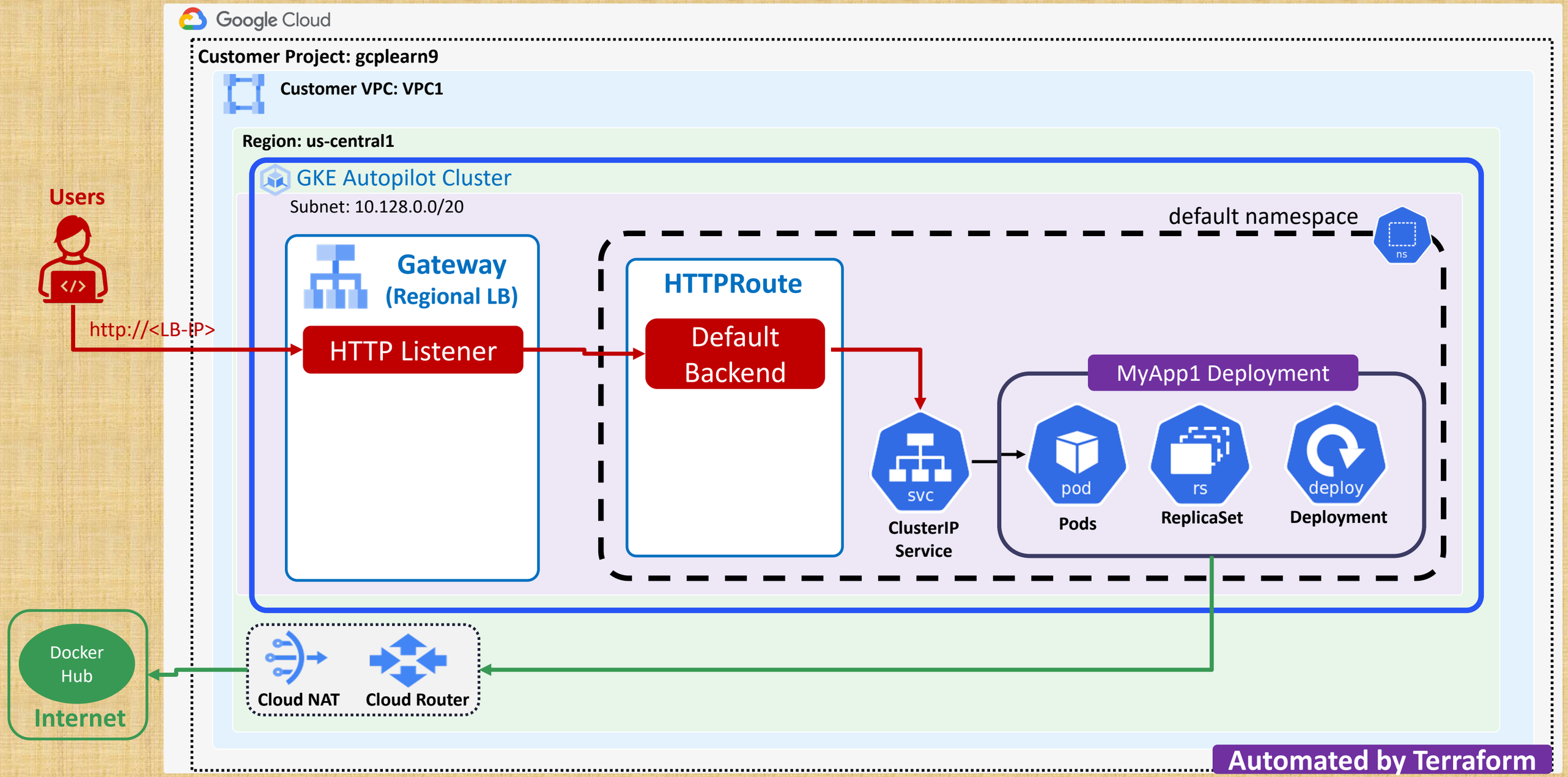
Gatewayclass	GCP Load Balancer	Load Balancer Scope
gke-l7-global-external-managed	Global External Application Load Balancer	Global
gke-l7-regional-external-managed	Regional External Application Load Balancer	Regional
gke-l7-rilb	Internal Application Load Balancer	Regional
gke-l7-gxlb-mc	Classic Application Load Balancer	Global

Reference: https://cloud.google.com/kubernetes-engine/docs/concepts/gateway-api

Author: Nho Luong

Skill: DevOps Engineer Lead

Gateway API Basics

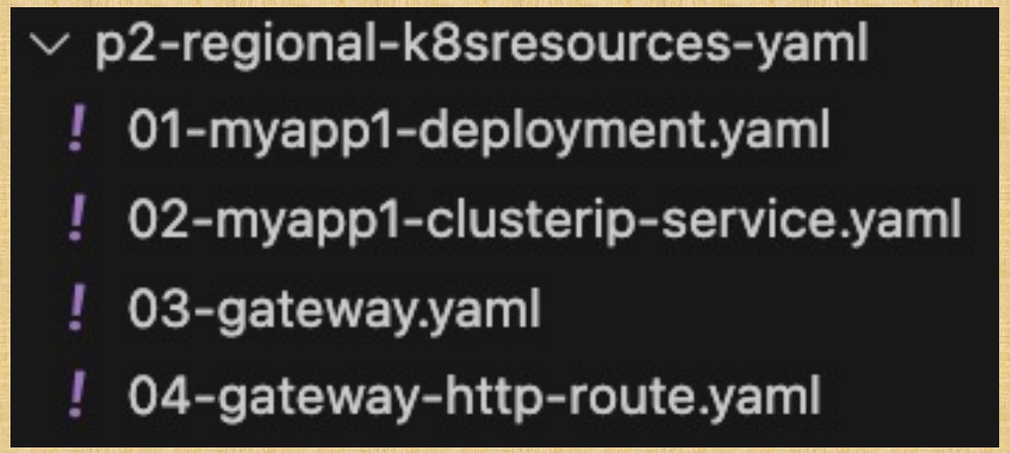


Author: Nho Luong

Skill: DevOps Engineer Lead

Project-2: YAML Manifests

What are we going to learn?



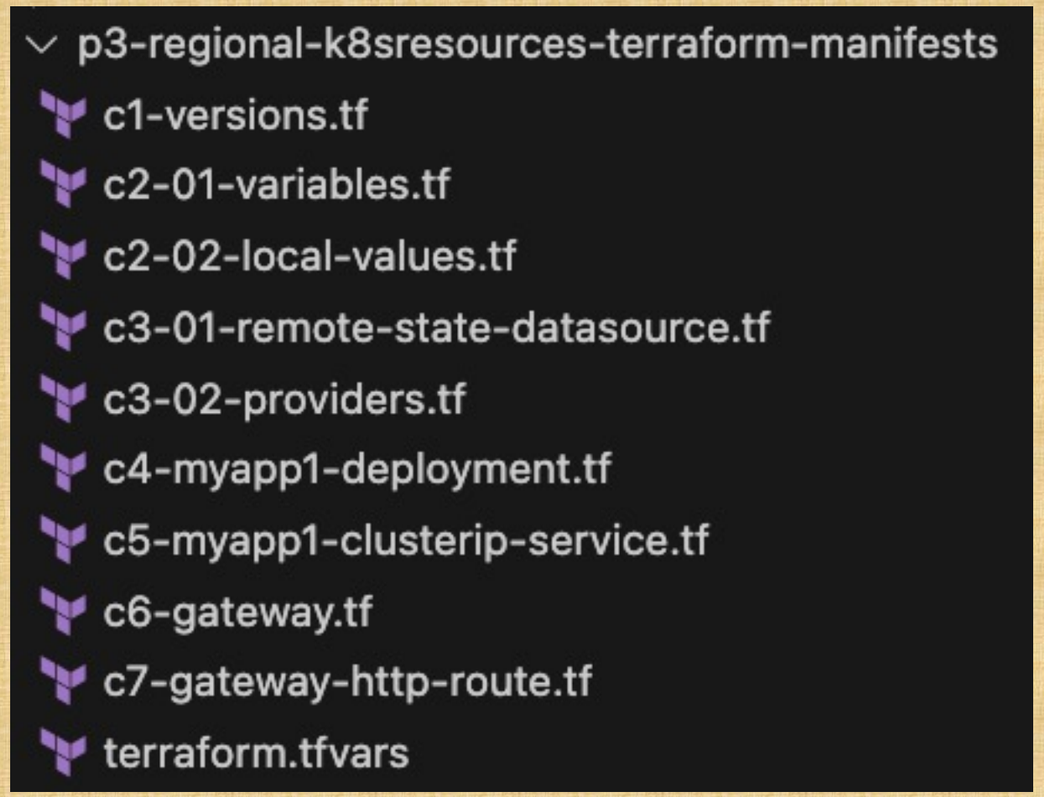
Demo: Project-2: GKE Autopilot cluster + Gateway API + basic scenario

NC	No changes 01 and 02
03	Gateway Manifest
04	Gateway HTTPRoute Manifest

Author: Nho Luong

Skill: DevOps Engineer Lead

Project-3: Terraform Manifests



What are we going to learn?

Demo: Project-3: GKE Autopilot cluster + Gateway API + basic scenario

NC	No changes c1 to c5
c6	Create Gateway Manifest which should create Regional Application Load Balancer
c7	Create Gateway HTTPRoute

Author: Nho Luong

Skill: DevOps Engineer Lead

Demo-02

Google Kubernetes Engine

Gateway API

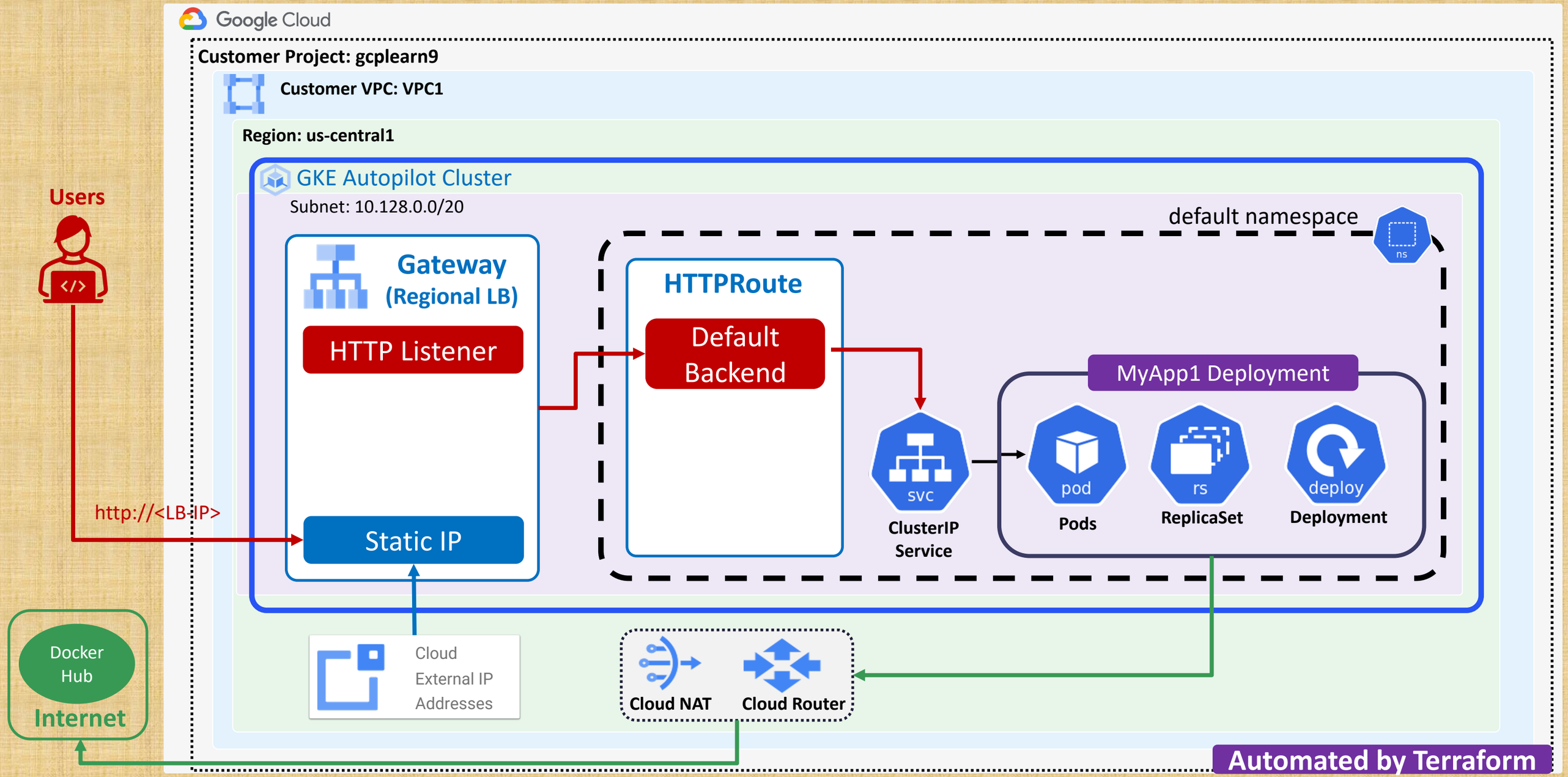
Cloud Application Load Balancer

Demo: GKE Autopilot cluster + Gateway + Static IP

Author: Nho Luong

Skill: DevOps Engineer Lead

Gateway API Basics + Static IP

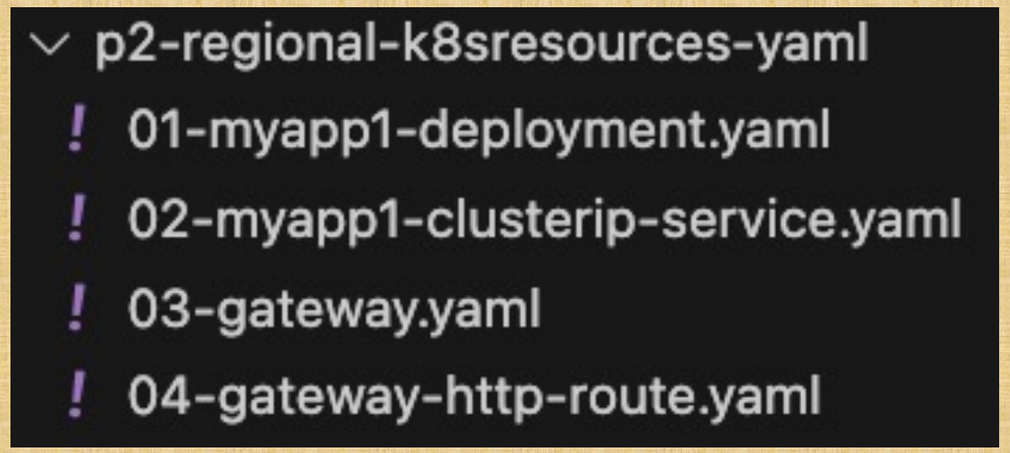


Author: Nho Luong

Skill: DevOps Engineer Lead

Project-2: YAML Manifests

What are we going to learn?



Demo: Project-2: GKE Autopilot cluster + Gateway + Regional Static IP

NC	No changes 01, 02 and 04
03	Create Static IP using gcloud and Update Gateway Manifest with Static IP

Author: Nho Luong

Skill: DevOps Engineer Lead

Project-3: Terraform Manifests

What are we going to learn?

Demo: Project-3: GKE Autopilot cluster + Gateway + Regional Static IP

NC	No changes c1 to c5, c7
c8	Create Regional Static IP
c6	Update Gateway manifest with Static IP

Author: Nho Luong

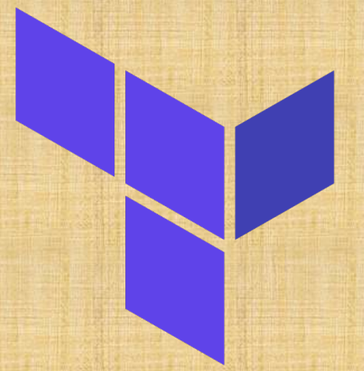
Skill: DevOps Engineer Lead

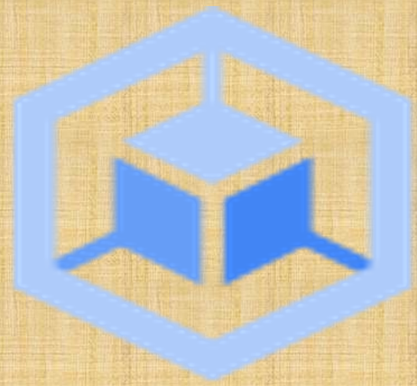
Demo-03

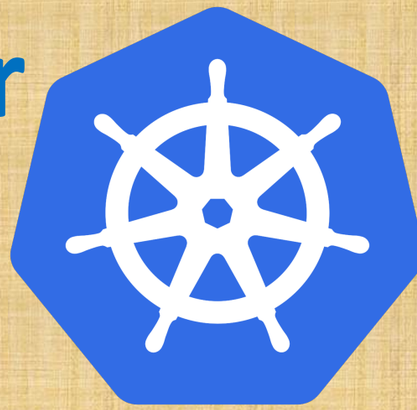
Google Kubernetes Engine

Gateway API

Cloud Application Load Balancer





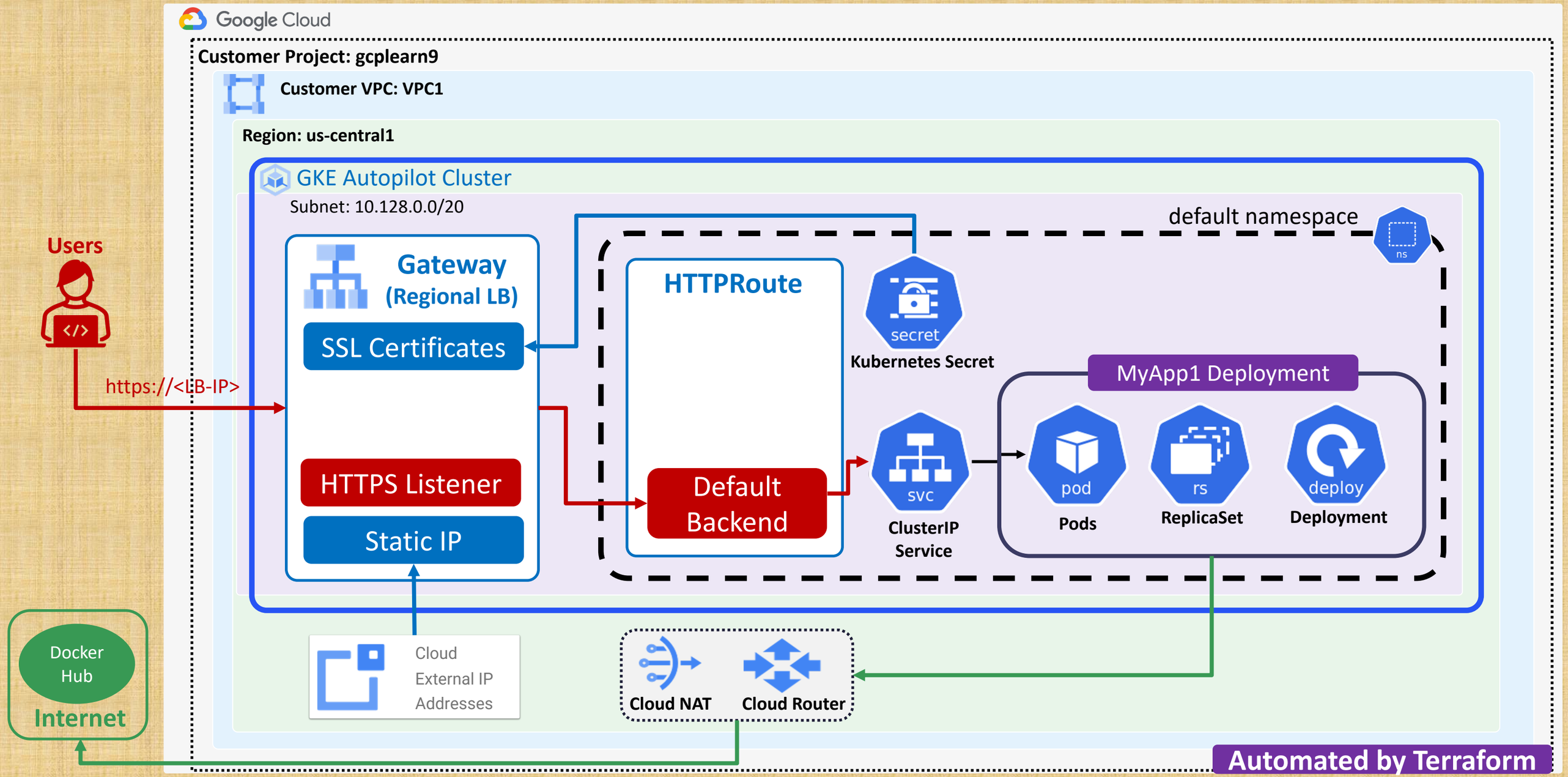


Demo: GKE Autopilot cluster + Gateway + Static IP + Self-signed SSL with k8s Secret

Author: Nho Luong

Skill: DevOps Engineer Lead

Gateway API Basics + Static IP + Self-signed SSL with Kubernetes Secret

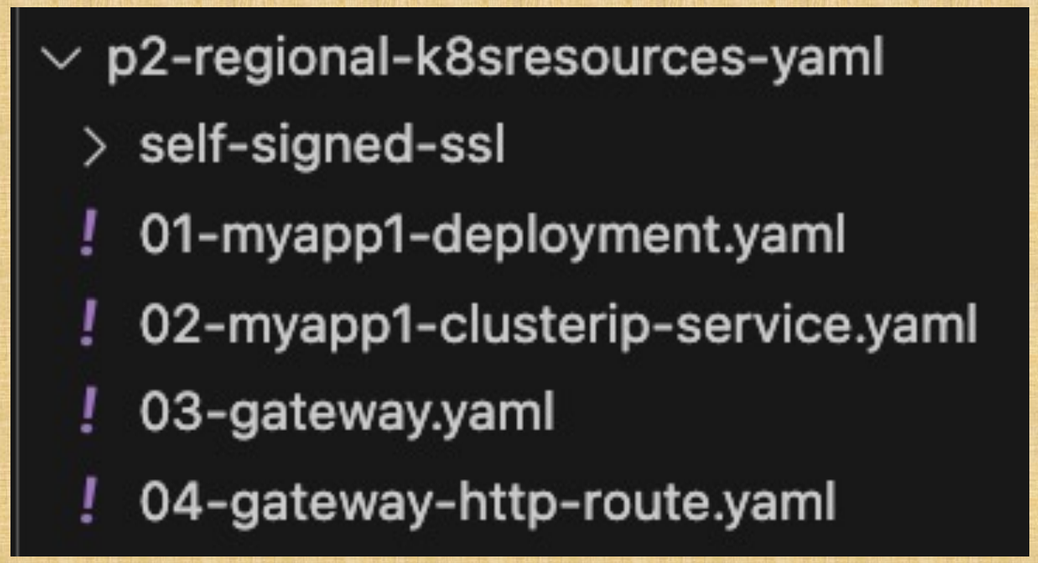


Author: Nho Luong

Skill: DevOps Engineer Lead

Project-2: YAML Manifests

What are we going to learn?



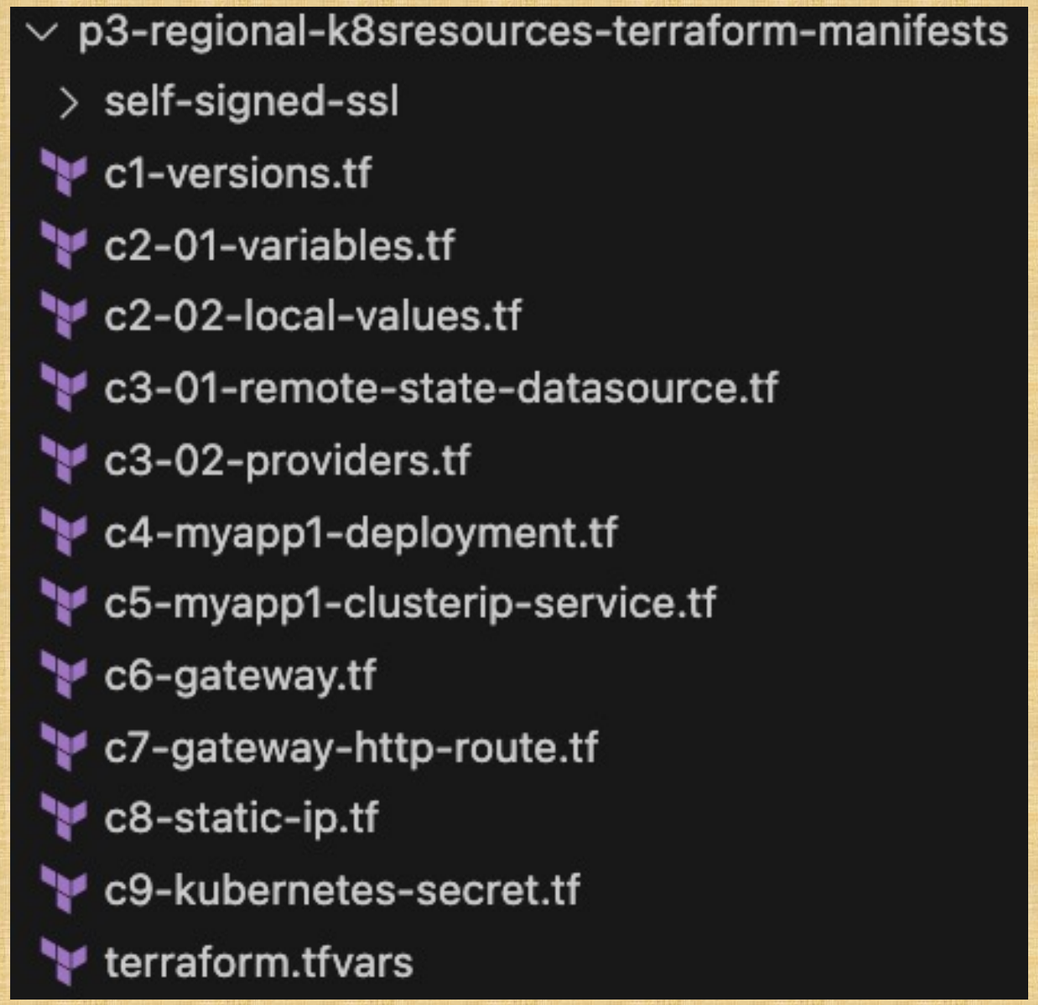
Demo: Project-2: GKE Autopilot cluster + Gateway + Regional Static IP + Self-signed SSL k8s secret

NC	No changes 01, 02 and 04
03	Create SSL Cert, Kubernetes secret and map the k8s secret in Gateway manifest
03	Update Listener to HTTPS and Port to 443

Author: Nho Luong

Skill: DevOps Engineer Lead

Project-3: Terraform Manifests



What are we going to learn?

Demo: Project-3: GKE Autopilot cluster + Gateway API + Regional Static IP + Self-signed SSL k8s secret

NC	No changes c1 to c5, c7, c8
c9	Create Kubernetes Secret with cert and key
c6	Update Gateway manifest with Kubernetes secret
c6	Update Listener to HTTPS and Port to 443

Author: Nho Luong

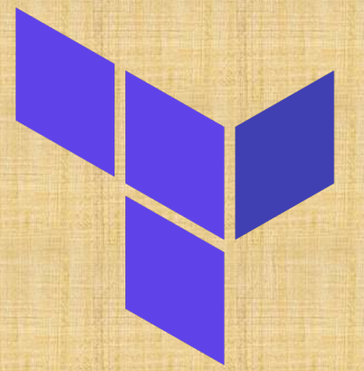
Skill: DevOps Engineer Lead

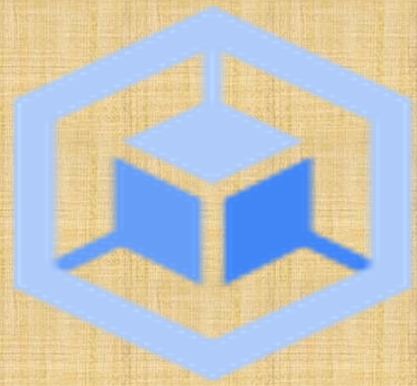
Demo-04

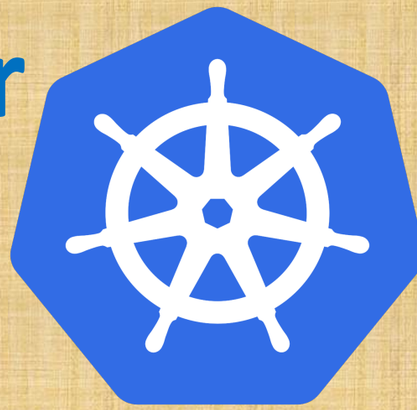
Google Kubernetes Engine

Gateway API

Cloud Application Load Balancer





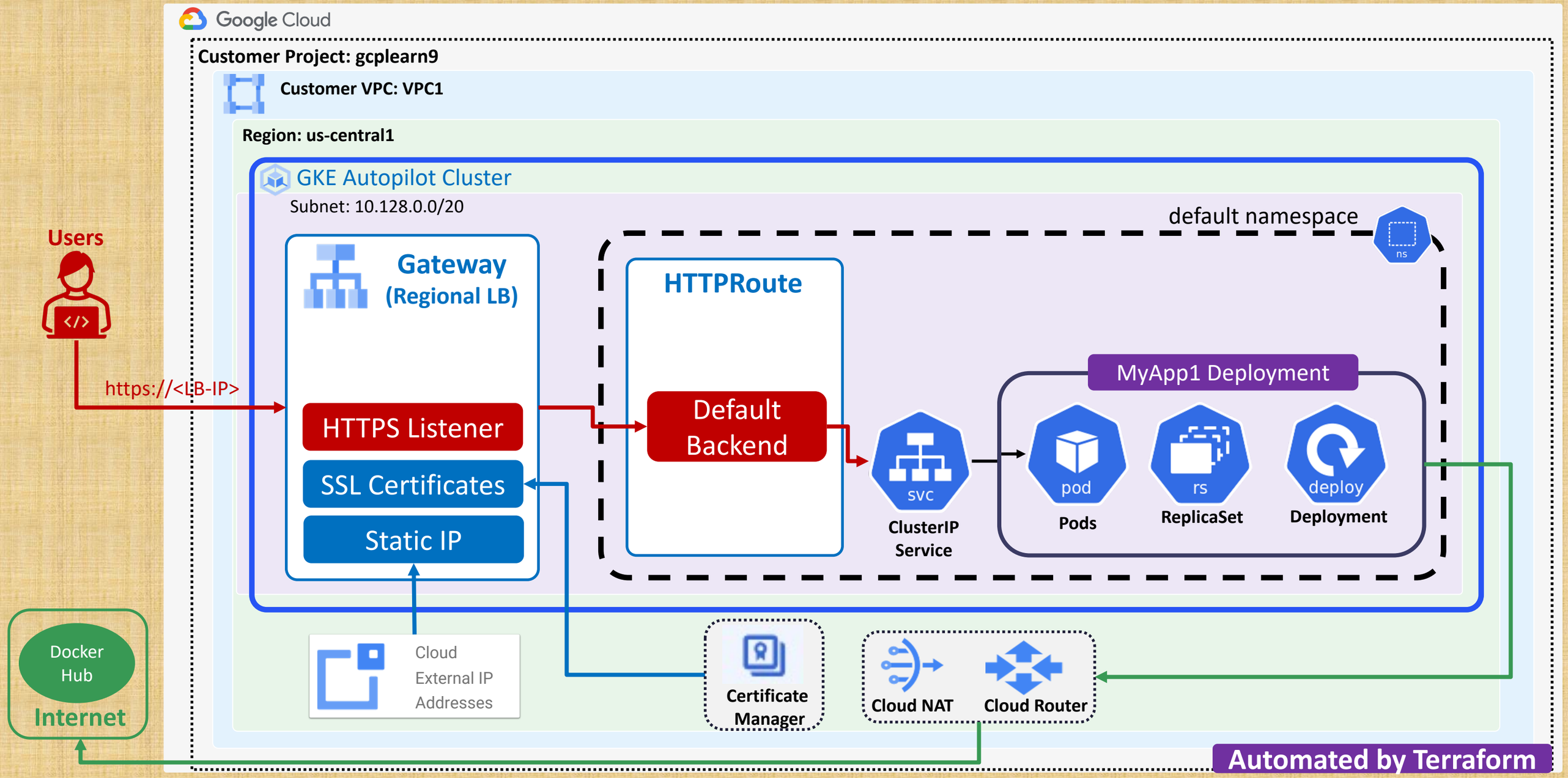


Demo: GKE Autopilot cluster + Gateway + Static IP + Self-signed SSL with Certificate Manager

Author: Nho Luong

Skill: DevOps Engineer Lead

Gateway API Basics + Static IP + Self-signed SSL with Certificate Manager

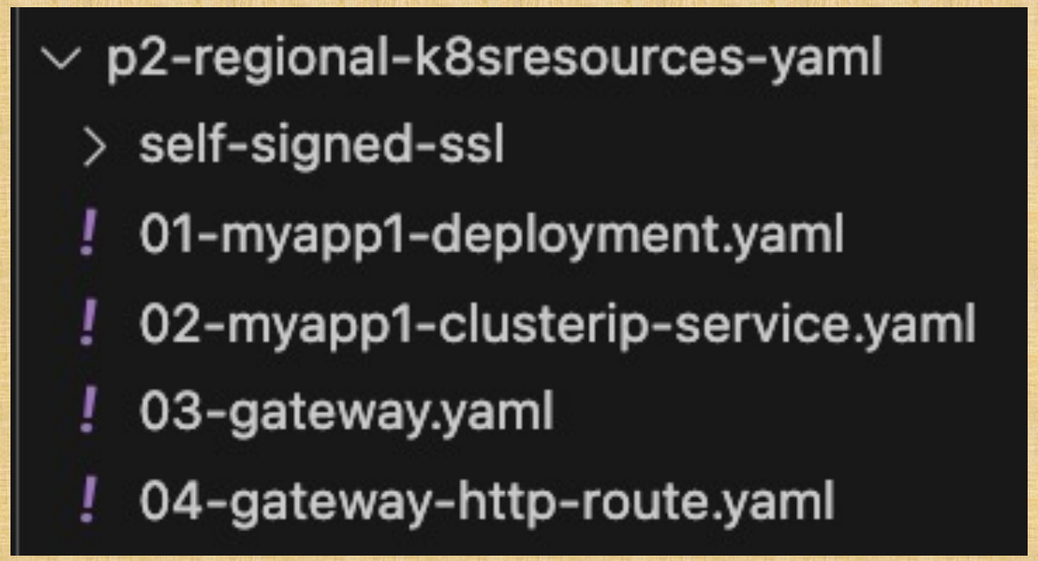


Author: Nho Luong

Skill: DevOps Engineer Lead

Project-2: YAML Manifests

What are we going to learn?



Demo: Project-2: GKE Autopilot cluster + Gateway + Regional Static IP + Self-signed SSL Certificate Manager

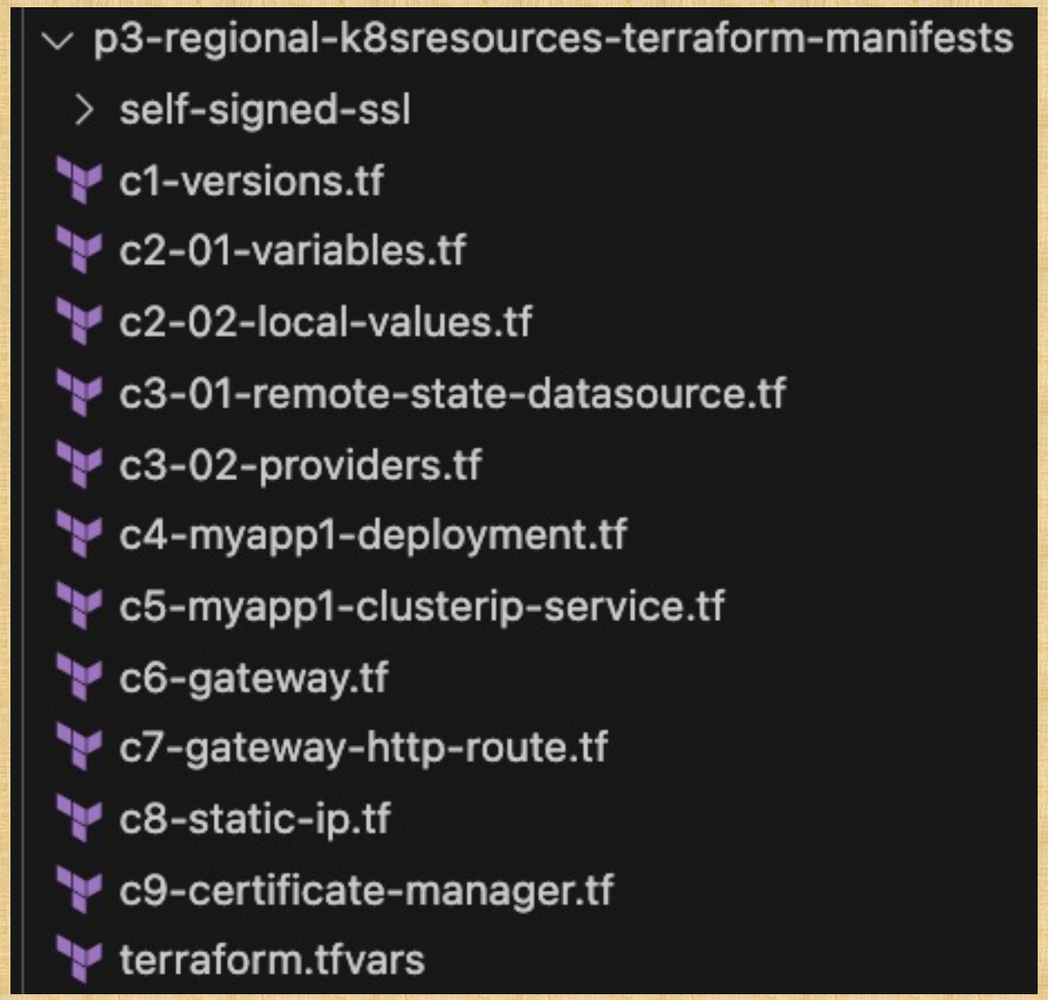
NC	No changes 01, 02 and 04
03	Create SSL Cert, Certificate Manager Object and reference it in Gateway Manifest

Author: Nho Luong

Skill: DevOps Engineer Lead

Project-3: Terraform Manifests

What are we going to learn?



Demo: Project-3: GKE Autopilot cluster + Gateway + Regional Static IP + Self-signed SSL Certificate Manager

NC	No changes c1 to c5, c7, c8
c9	Create Self-signed SSL certificate using Certificate Manager
c6	Update Gateway manifest with Certificate Manager object

Author: Nho Luong

Skill: DevOps Engineer Lead

Demo-05

Google Kubernetes Engine

Gateway API

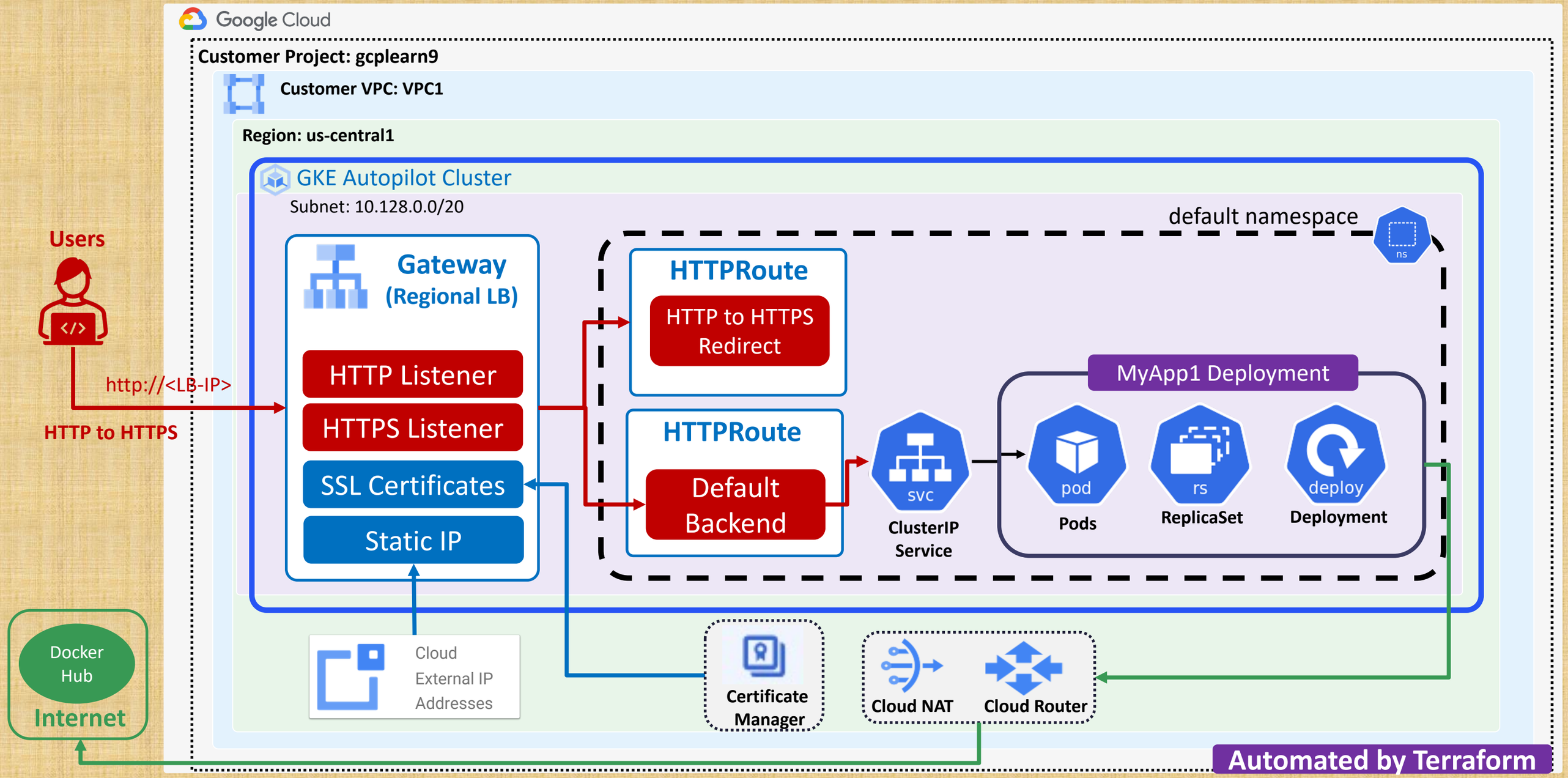
Cloud Application Load Balancer

Demo: GKE Autopilot cluster + Gateway + Static IP + Self-signed SSL with Certificate Manager + HTTP to HTTPS Redirect

Author: Nho Luong

Skill: DevOps Engineer Lead

Gateway API Basics + Static IP + SSL Certificate Manager + HTTP to HTTPS

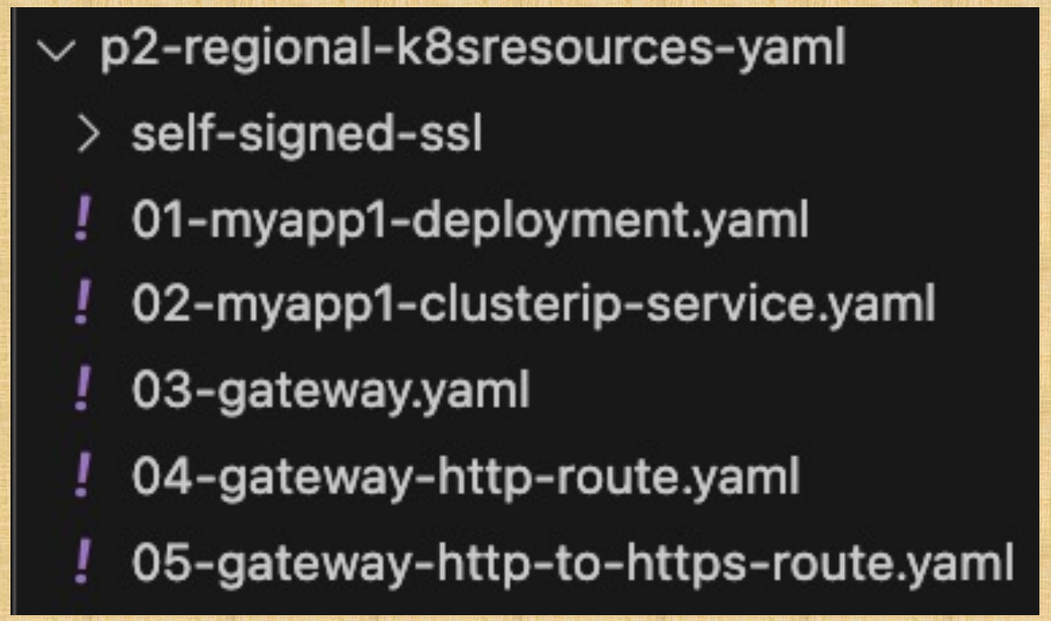


Author: Nho Luong

Skill: DevOps Engineer Lead

Project-2: YAML Manifests

What are we going to learn?



Demo: Project-2: Gateway + Regional Static IP + Self-signed SSL Certificate Manager + HTTP to HTTPS Redirect

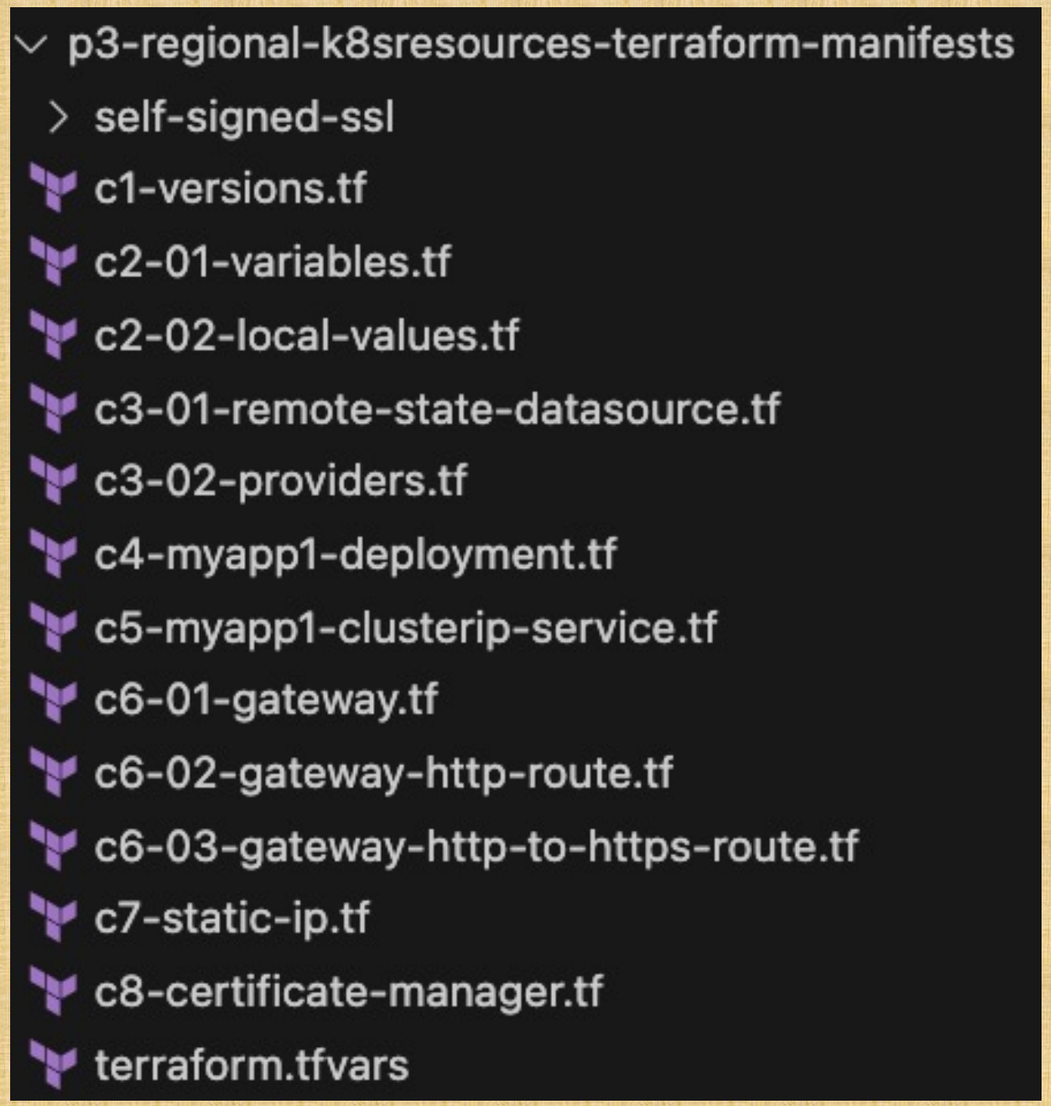
NC	No changes 01, 02 and 04
03	Add both HTTP and HTTPS listeners
04	Update route with “sectionName: https”, in short map this route to HTTPS listener in gateway
05	Create new HTTP to HTTPS redirect Route

Author: Nho Luong

Skill: DevOps Engineer Lead

Project-3: Terraform Manifests

What are we going to learn?



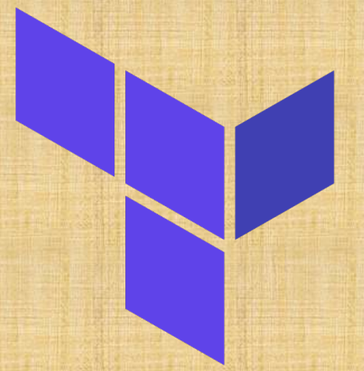
Demo: Project-3: Gateway + Regional Static IP + Self-signed SSL Certificate Manager + HTTP to HTTPS Redirect

NC	No changes c1 to c5, c7, c8
C6-01	Create both HTTP and HTTPS Listeners in Gateway manifest
C6-02	Update route with “sectionName: https”, in short map this route to HTTPS listener in gateway
C6-03	Create new HTTP to HTTPS redirect Route

Author: Nho Luong

Skill: DevOps Engineer Lead

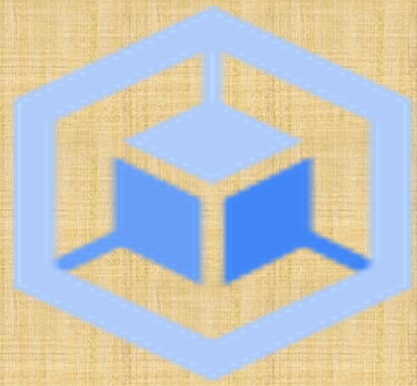
Demo-06

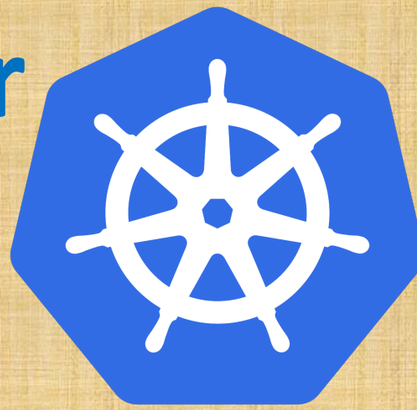


Google Kubernetes Engine

Gateway API

Cloud Application Load Balancer



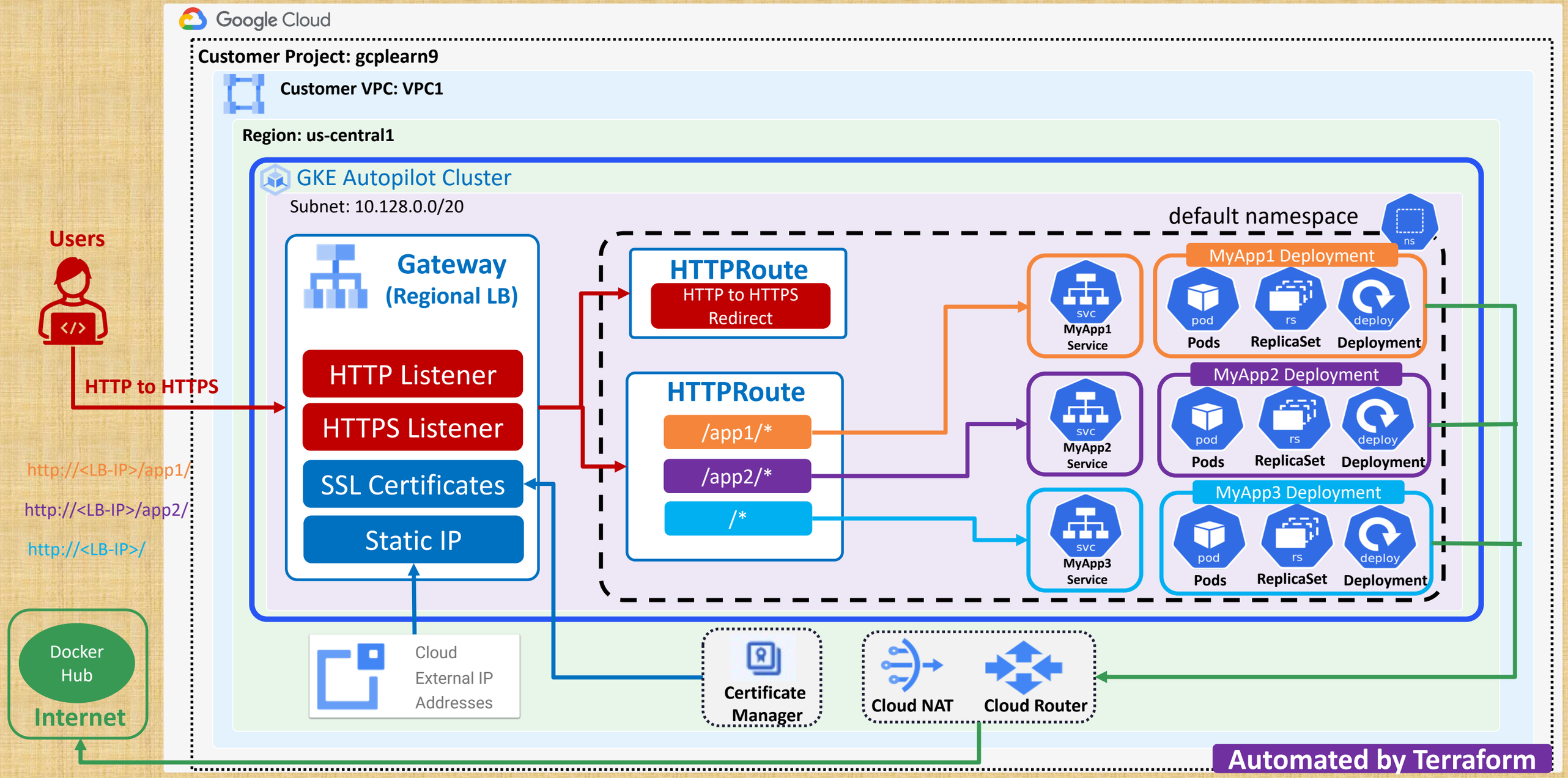


Demo: GKE Autopilot cluster + Gateway + Static IP + Self-signed SSL with Certificate Manager + HTTP to HTTPS Redirect + Context Path based Routing

Author: Nho Luong

Skill: DevOps Engineer Lead

Gateway API Basics + Static IP + SSL Certificate Manager + HTTP to HTTPS + Context Path based Routing

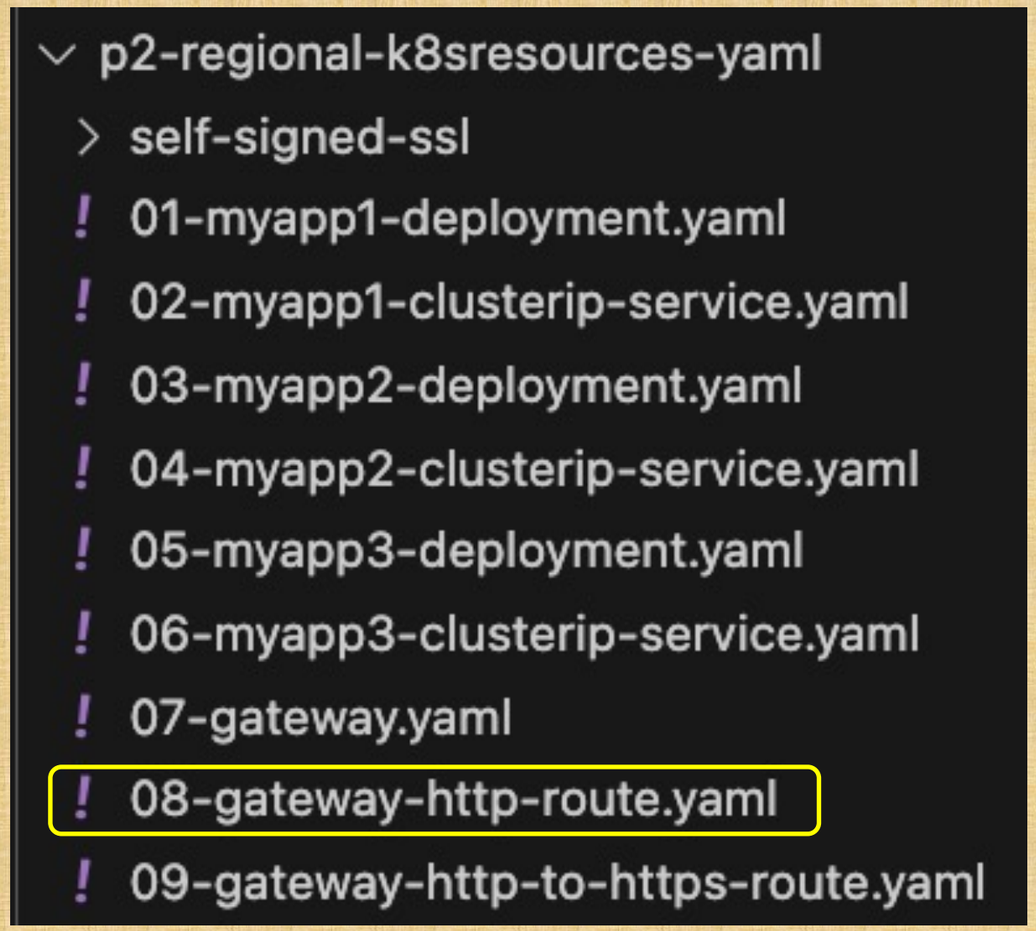


Author: Nho Luong

Skill: DevOps Engineer Lead

Project-2: YAML Manifests

What are we going to learn?



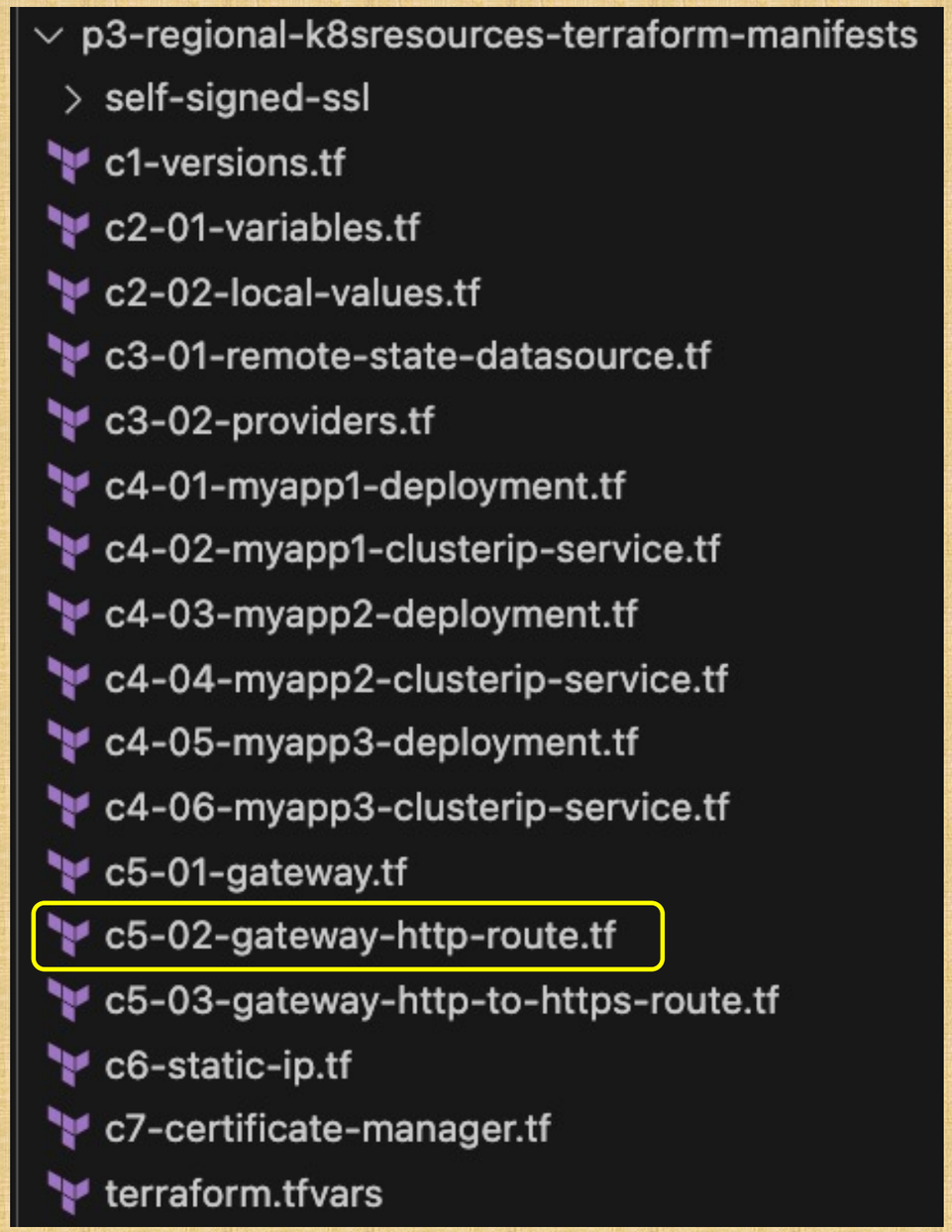
Demo: Project-2: Gateway + Regional Static IP + Self-signed SSL Certificate Manager + HTTP to HTTPS + Context Path Routing

01 to 06	MyApp1, 2, 3 Deployments and Cluster IP Services
07	Gateway – No changes
09	HTTP to HTTPS redirect – No changes
08	Define Context path-based routing

Author: Nho Luong

Skill: DevOps Engineer Lead

Project-3: Terraform Manifests



What are we going to learn?

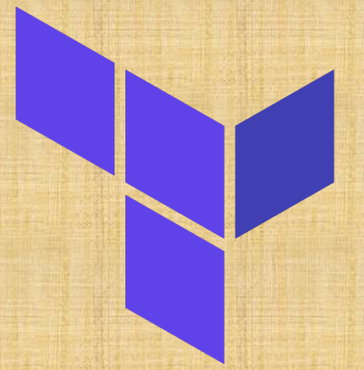
Demo: Project-3: Gateway + Regional Static IP + Self-signed SSL Certificate Manager + HTTP to HTTPS + Context Path Routing

NC	No changes c1 to c3-02, c6, c7
C4	MyApp1, 2, 3 Deployments and Cluster IP Services
C5-01	No changes in Gateway manifest
C5-03	No changes in HTTP to HTTPS redirect
C5-02	Define Context path-based routing

Author: Nho Luong

Skill: DevOps Engineer Lead

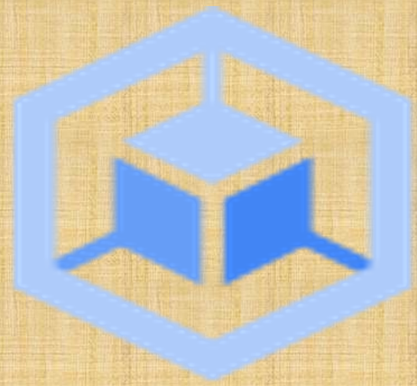
Demo-07

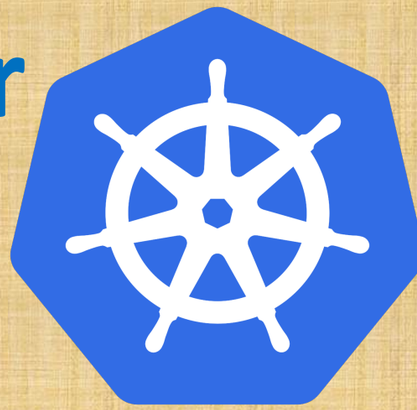


Google Kubernetes Engine

Gateway API

Cloud Application Load Balancer



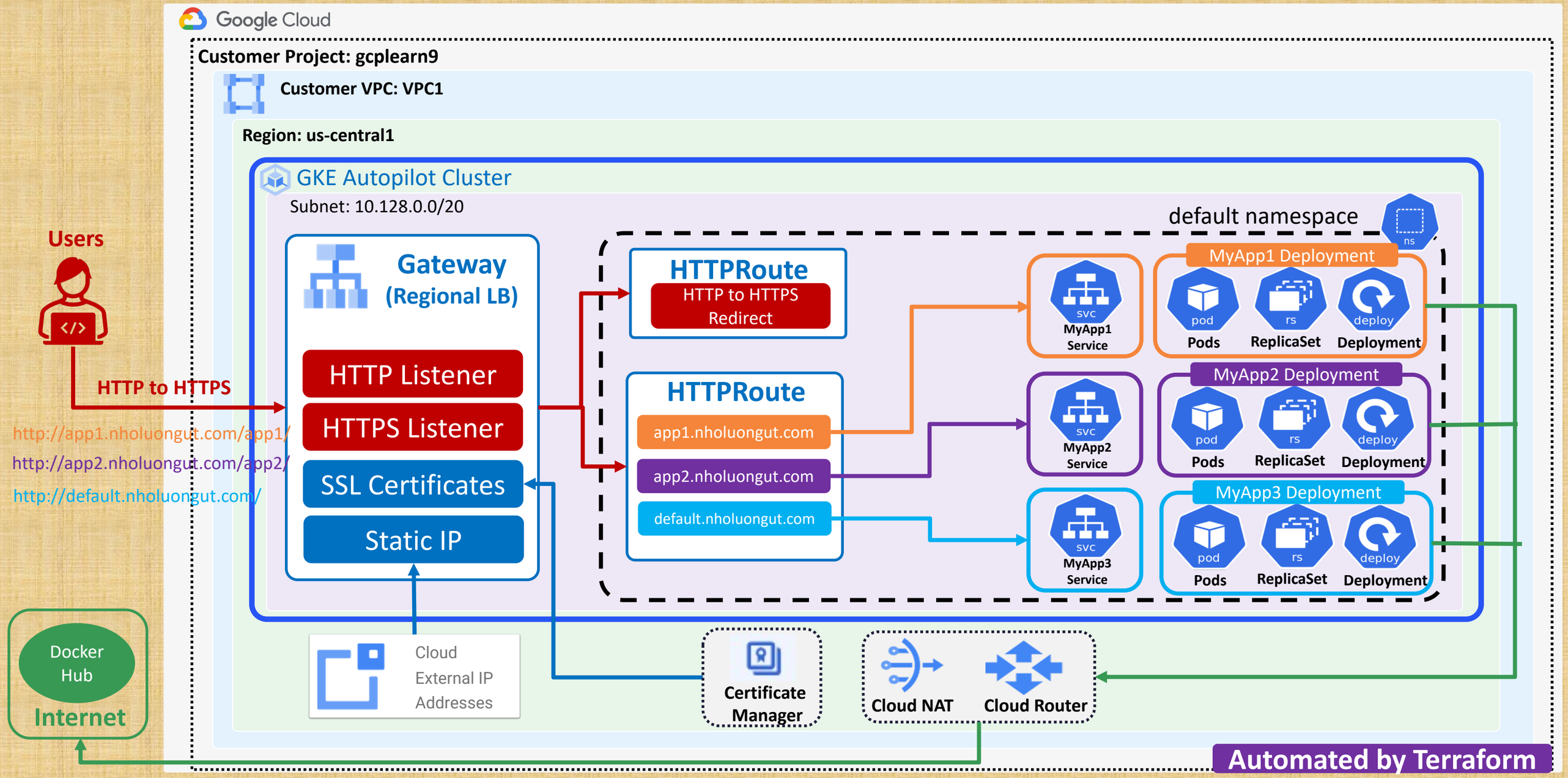


Demo: GKE Autopilot cluster + Gateway + Static IP + Self-signed SSL with Certificate Manager + HTTP to HTTPS Redirect + **Domain name based Routing**

Author: Nho Luong

Skill: DevOps Engineer Lead

Gateway API Basics + Static IP + SSL Certificate Manager + HTTP to HTTPS + Domain name-based Routing

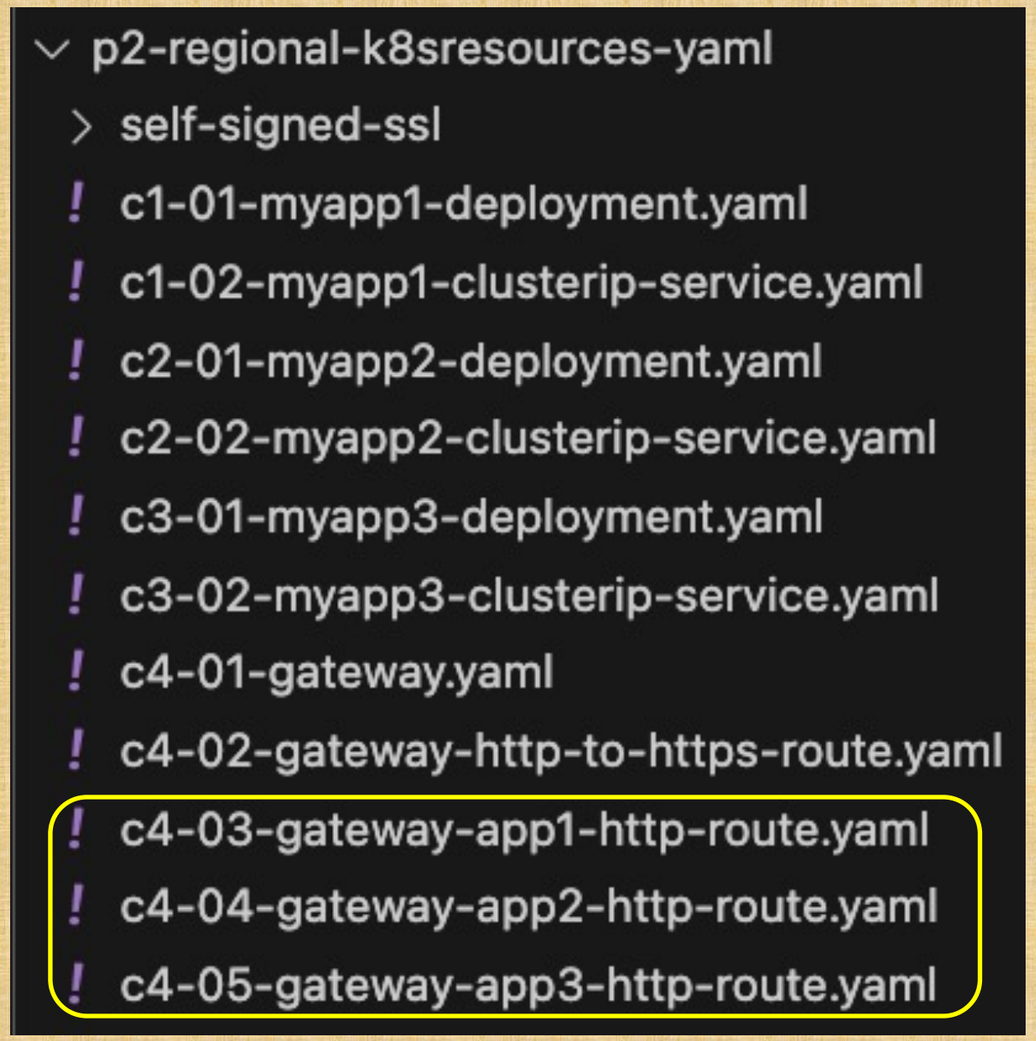


Author: Nho Luong

Skill: DevOps Engineer Lead

Project-2: YAML Manifests

What are we going to learn?



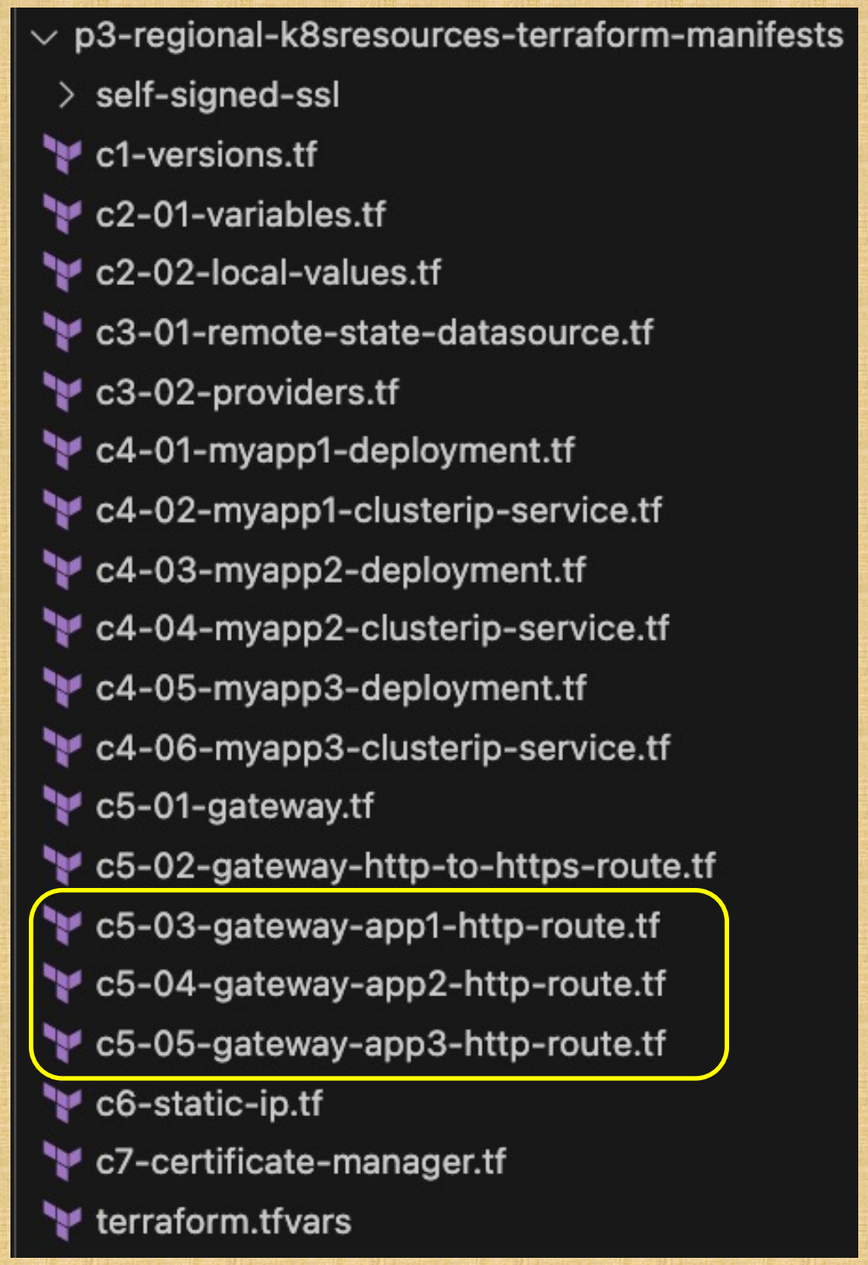
Demo: Project-2: Gateway + Regional Static IP + Self-signed SSL Certificate Manager + HTTP to HTTPS + Domain Name Routing

C1 to C4-02	MyApp1, 2, 3 Deployments and Cluster IP Services
c4-01	Gateway – No changes
c4-02	HTTP to HTTPS redirect – No changes
c4-03	App1 HTTP Route with domain name as app1.stacksimplify.com
c4-04	App2 HTTP Route with domain name as app2.stacksimplify.com
c4-05	App3 HTTP Route with domain name as default.stacksimplify.com

Author: Nho Luong

Skill: DevOps Engineer Lead

Project-3: Terraform Manifests



What are we going to learn?

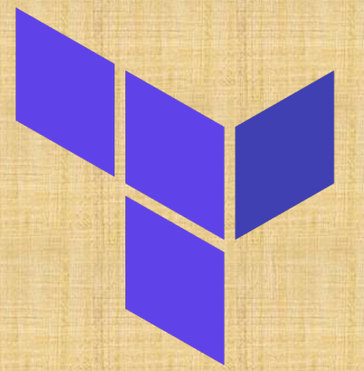
Demo: Project-3: Gateway + Regional Static IP + Self-signed SSL Certificate Manager + HTTP to HTTPS + Domain Name Routing

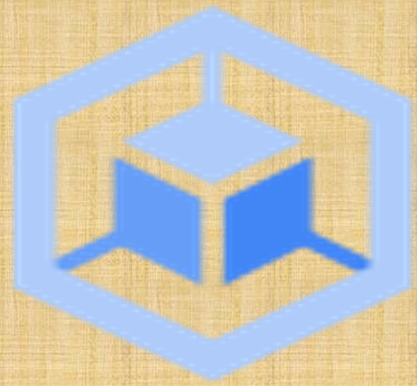
NC	No changes c1 to c5-02, c6, c7
c5-03	App1 HTTP Route with domain name as app1.stacksimplify.com
c5-04	App2 HTTP Route with domain name as app2.stacksimplify.com
c5-05	App3 HTTP Route with domain name as default.stacksimplify.com

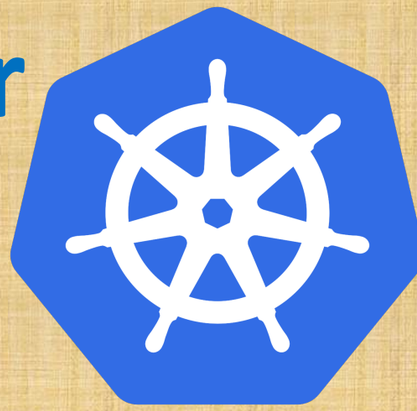
Author: Nho Luong

Skill: DevOps Engineer Lead

Demo-08







Google Kubernetes Engine

Gateway API

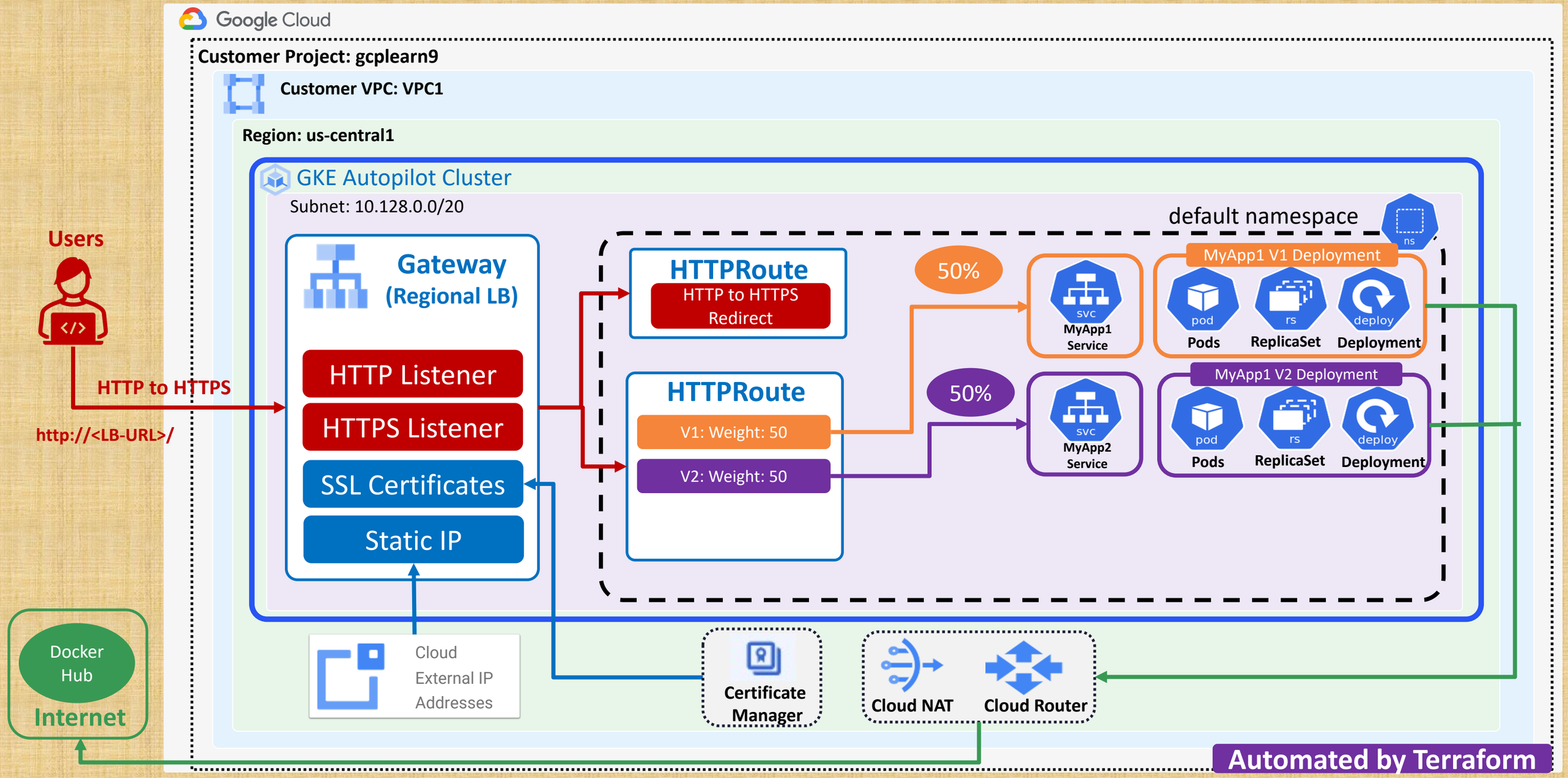
Cloud Application Load Balancer

Demo: GKE Autopilot cluster + Gateway + Static IP + Self-signed SSL with Certificate Manager + HTTP to HTTPS Redirect + **Traffic Splitting**

Author: Nho Luong

Skill: DevOps Engineer Lead

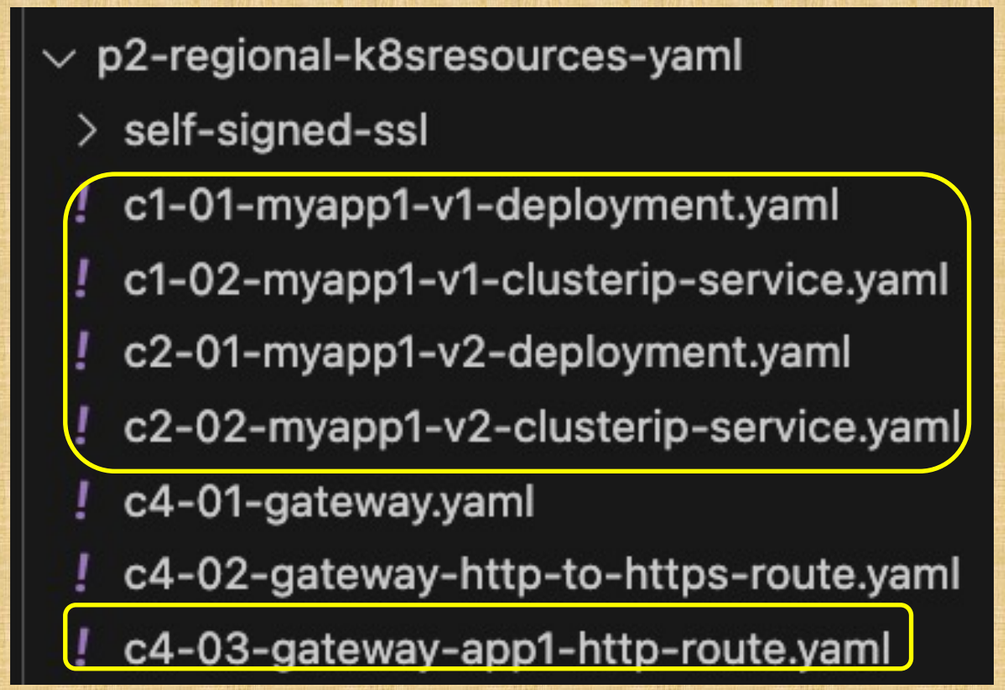
Gateway API Basics + Static IP + SSL Certificate Manager + HTTP to HTTPS + Traffic Splitting



Author: Nho Luong

Skill: DevOps Engineer Lead

Project-2: YAML Manifests



What are we going to learn?

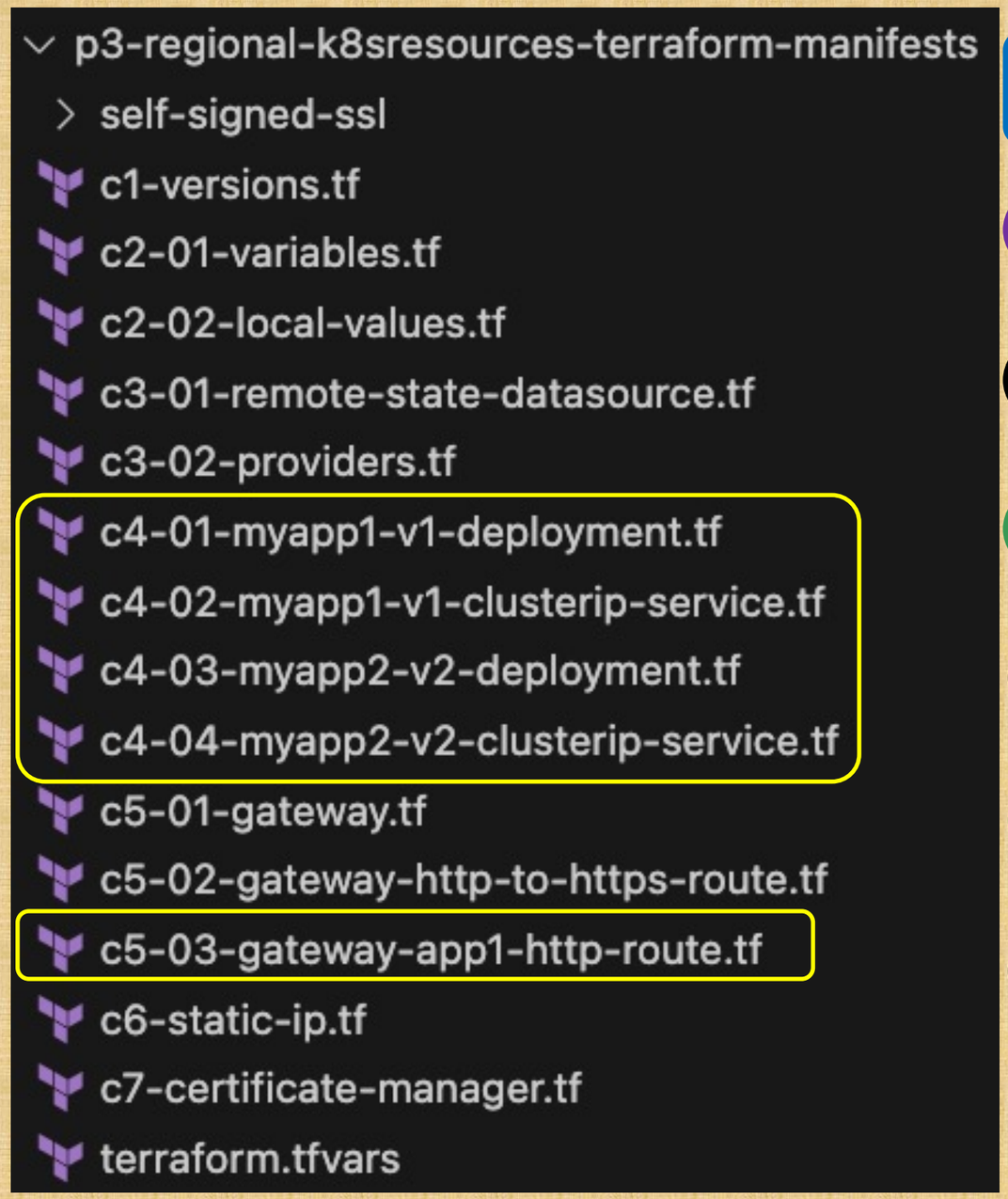
Demo: Project-2: Gateway + Regional Static IP + Self-signed SSL Certificate Manager + HTTP to HTTPS + Traffic Splitting

NC	NO Changes c4-01, c4-02,
c1-01	MyApp1 V1 Kubernetes Deployment
c1-02	MyApp1 V1 Cluster IP Service
c2-01	MyApp1 V2 Kubernetes Deployment
c2-02	MyApp1 V2 Cluster IP Service
c4-03	HTTP Route: V1 and V2 Traffic Splitting

Author: Nho Luong

Skill: DevOps Engineer Lead

Project-3: Terraform Manifests



What are we going to learn?

Demo: Project-3: Gateway + Regional Static IP + Self-signed SSL Certificate Manager + HTTP to HTTPS + Traffic Splitting

NC	No changes c1 to c5-02, c6, c7
c4-01	MyApp1 V1 Kubernetes Deployment
c4-02	MyApp1 V1 Cluster IP Service
c4-03	MyApp1 V2 Kubernetes Deployment
c4-04	MyApp1 V2 Cluster IP Service
c5-03	HTTP Route: V1 and V2 Traffic Splitting

Author: Nho Luong

Skill: DevOps Engineer Lead

Demo-09

Google Kubernetes Engine

Gateway API

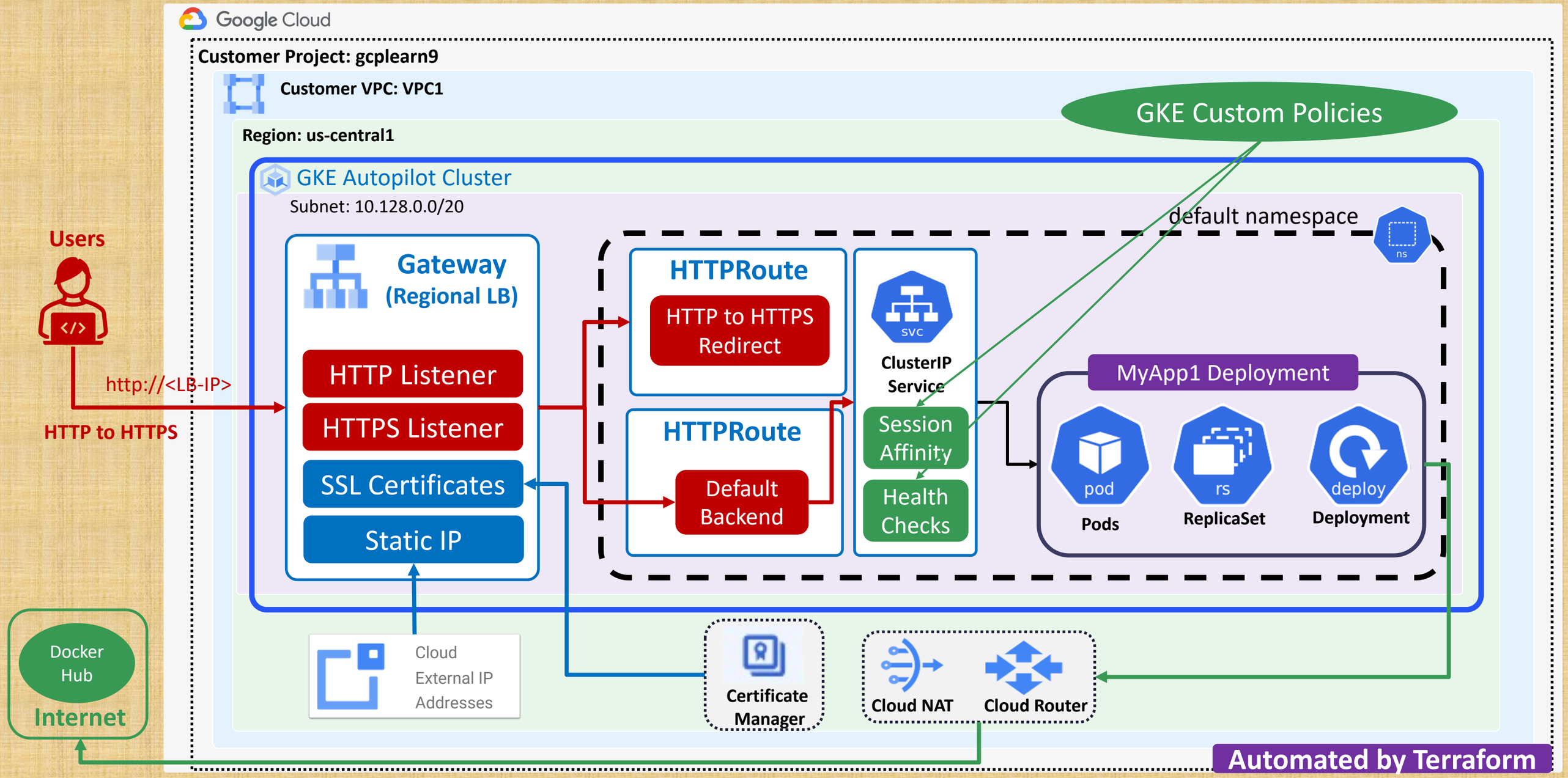
Cloud Application Load Balancer

Demo: GKE Autopilot cluster + Gateway + Static IP + Self-signed SSL with Certificate Manager + HTTP to HTTPS Redirect + Health Checks + Session Affinity

Author: Nho Luong

Skill: DevOps Engineer Lead

Gateway API + Static IP + SSL Certificate Manager + HTTP to HTTPS + Health Checks + Session Affinity

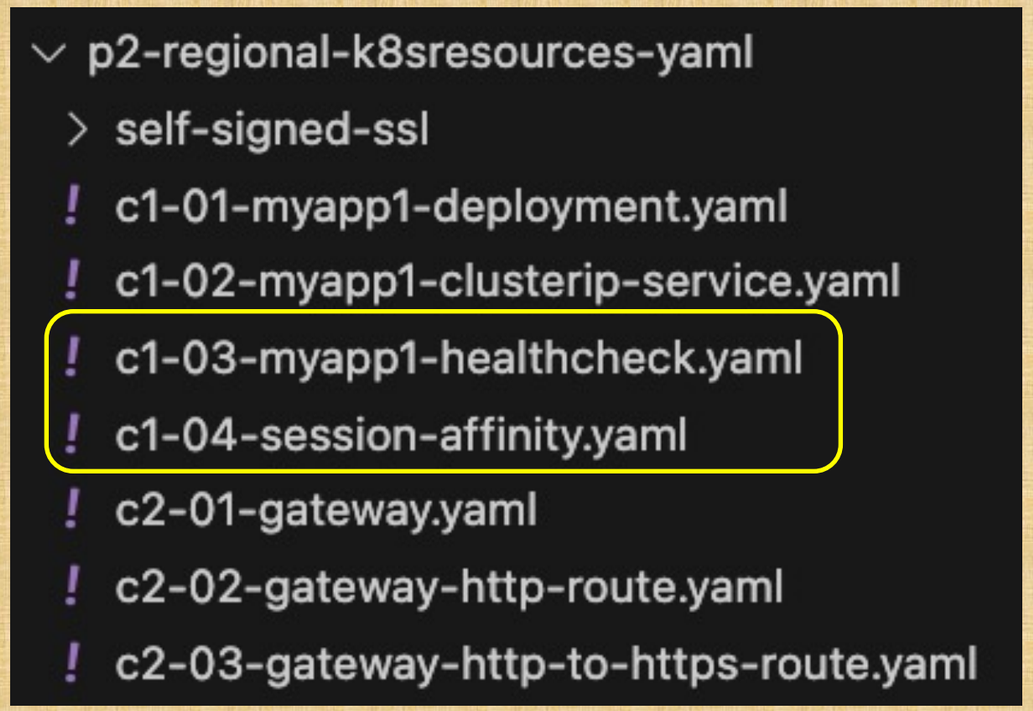


Author: Nho Luong

Skill: DevOps Engineer Lead

Project-2: YAML Manifests

What are we going to learn?



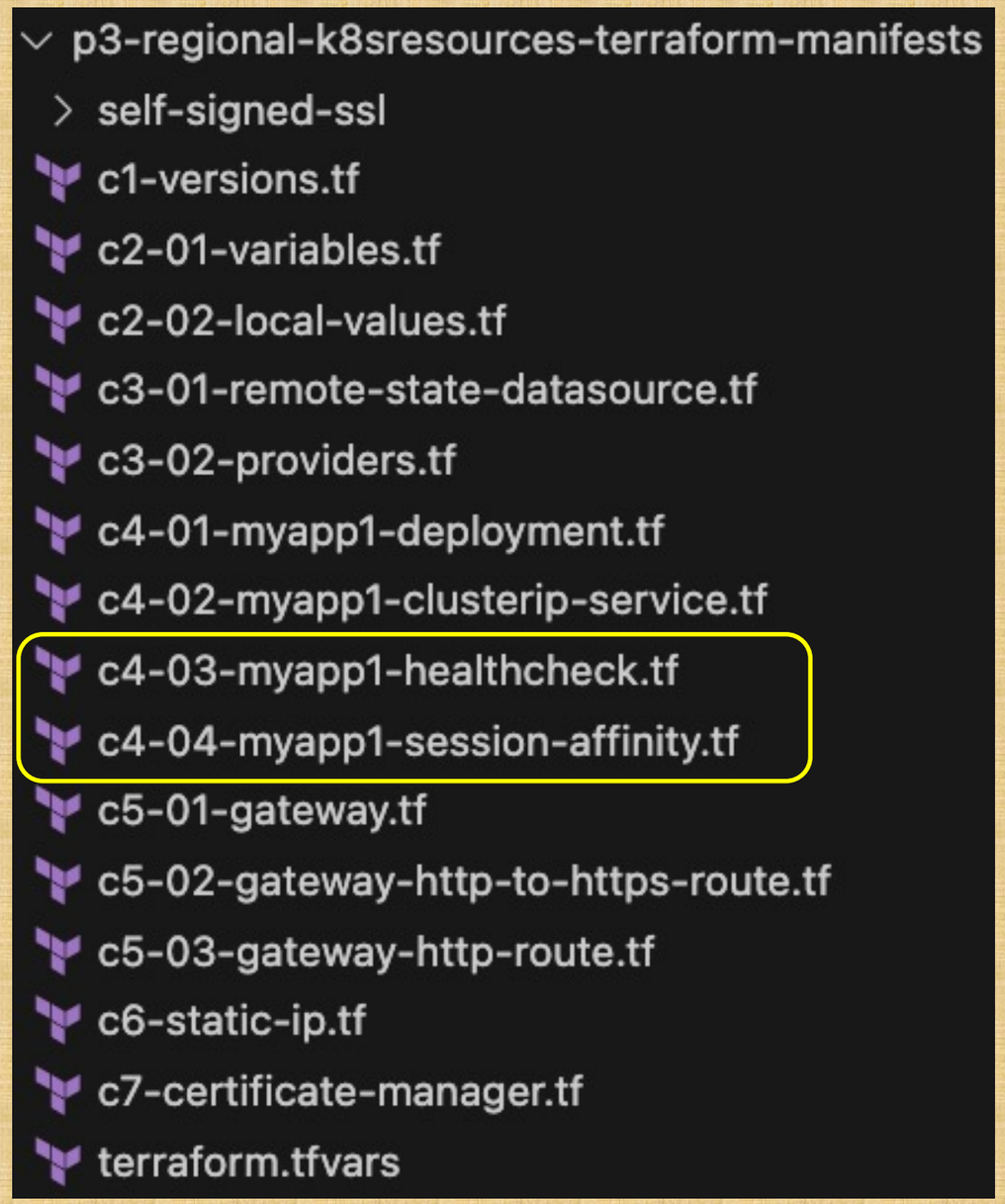
Demo: Project-2: Gateway + Regional Static IP + Self-signed SSL Certificate Manager + HTTP to HTTPS + Health Checks + Session Affinity

NC	NO Changes c1-01, c1-02, c2-01, c2-02, c2-04
c1-03	MyApp1 Health Check Policy
c1-05	MyApp1 Session Affinity Policy

Author: Nho Luong

Skill: DevOps Engineer Lead

Project-3: Terraform Manifests



What are we going to learn?

Demo: Project-3: Gateway + Regional Static IP + Self-signed SSL Certificate Manager + HTTP to HTTPS + Health Checks + Session Affinity

NC	No changes c1 to c4-02, c5-01 to c7
c4-03	MyApp1 Health Check Policy
c4-04	MyApp1 Session Affinity Policy

Author: Nho Luong

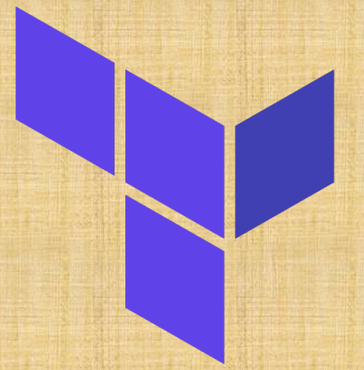
Skill: DevOps Engineer Lead

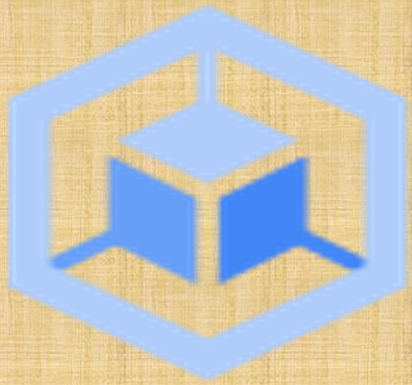
Demo-10

Google Kubernetes Engine

Gateway API

Cloud Domains + Cloud DNS





Author: Nho Luong

Skill: DevOps Engineer Lead

Demo

Google Cloud Networking

Cloud Domains

Author: Nho Luong

Skill: DevOps Engineer Lead

Google Cloud Domains

- **Cloud Domains:** Primarily used for Domain Registration and management
- **Key Benefits**
  - Register new domains
  - Bills for purchasing/renewing domains will use the same Cloud Billing account
  - Automatic renewal of registered domains
  - Let's you manage domain registrations per project, not per user
  - Supports DNSSEC which protects your domains from spoofing and cache poisoning attacks
  - Tightly integrated with Google Cloud DNS (Domain Name system) for creating and managing DNS records

Author: Nho Luong

Skill: DevOps Engineer Lead

Google Cloud Domains

- **Important Note:** From September 7, 2023 onwards Squarespace acquired all domain registrations and related customer accounts from Google Domains
- Cloud Domains uses Google Domains as underlying domain registrar
- Now after squarespace acquisition, Cloud Domains uses Squarespace Domains as underlying registrar
- **How does this impact Cloud Domains in Google Cloud ?**
  - **Complete Faq:** https://cloud.google.com/domains/docs/faq
  - In a shorter note, google supports Cloud Domains even after this acquisition
  - **Following features in Cloud Domain will work as-is**
    - Searching and registering new domains
    - Renewing existing domains
    - Update your contact details and DNS settings
    - Transfer a registered domain to another registrar (GoDaddy, Namecheap, AWS Route53)
    - Transfer domain from another registrar to Cloud Domain - NOT POSSIBLE

Additional Reference: https://cloud.google.com/domains/docs/deprecations/feature-deprecations

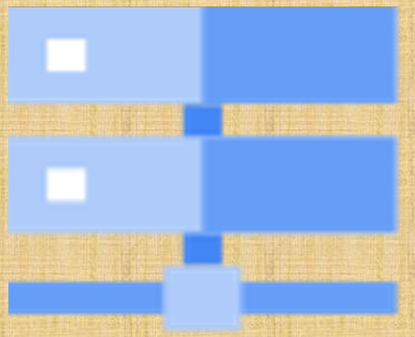
Author: Nho Luong

Skill: DevOps Engineer Lead

Demo

Google Cloud Networking

Cloud DNS



Author: Nho Luong

Skill: DevOps Engineer Lead

Google Cloud DNS

- **Cloud DNS:** High-performance, resilient, global Domain Name System (DNS) service
- Cloud DNS translates requests for domain names like google.com into IP addresses like 74.125.200.113

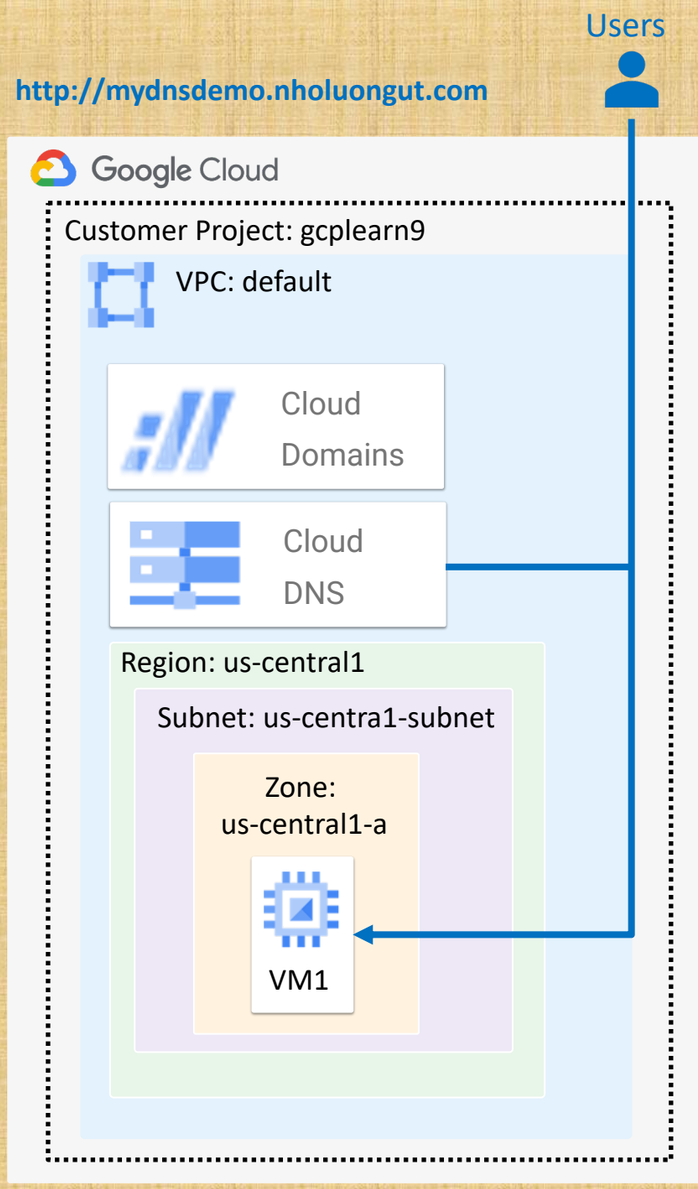
```
nholu@nholuongs-MacBook-Pro Github % nslookup google.com
Server:         8.8.8.8
Address:        8.8.8.8#53

Non-authoritative answer:
Name:   google.com
Address: 74.125.200.113
Name:   google.com
Address: 74.125.200.102
Name:   google.com
Address: 74.125.200.101
Name:   google.com
Address: 74.125.200.100
Name:   google.com
Address: 74.125.200.138
Name:   google.com
Address: 74.125.200.139
```

Author: Nho Luong

Skill: DevOps Engineer Lead

Google Cloud DNS - Terminology

- **DNS Zones:** It is a container of DNS records for the same DNS name suffix (Example suffix: google.com)
  - **Public Zone**
    - Visible to the internet
    - Primarily used for DNS management of your internet facing applications
    - DNSSEC (DNS Security) protects your domains from spoofing and cache poisoning attacks
  - **Private Zone**
    - Contains DNS records that are only visible internally within your Google Cloud networks
    - Easy to manage internal DNS solution for all our internal needs (For Internal applications, VM Instances etc)



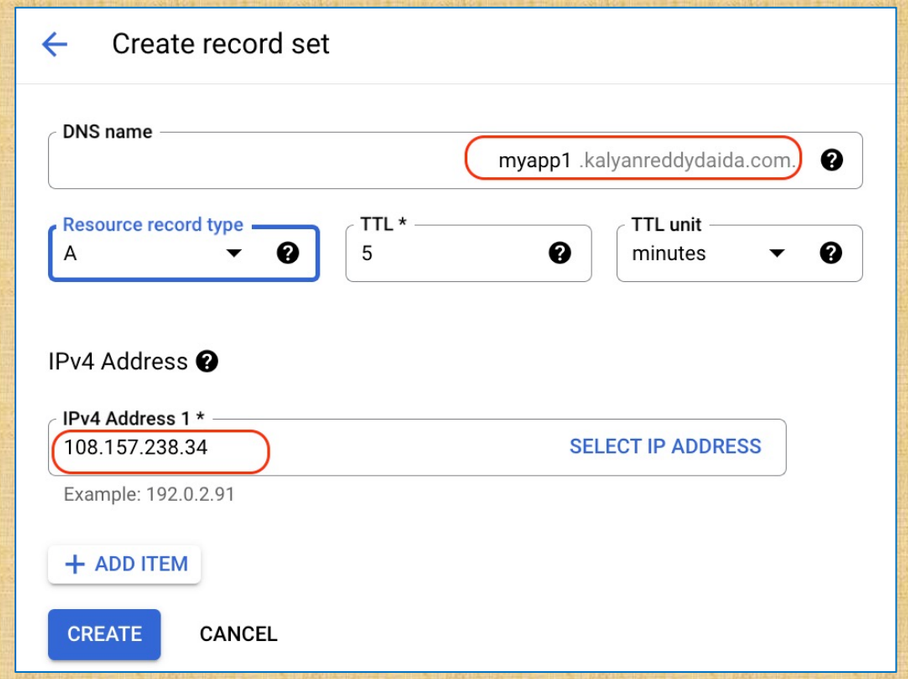
Author: Nho Luong

Skill: DevOps Engineer Lead

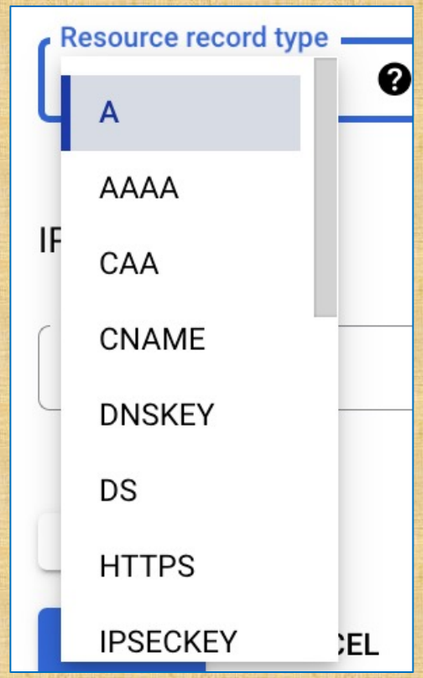
Google Cloud DNS - Terminology

- **Record Sets:** actual DNS records
  - DNS Name to IP Address mapping
  - We have different record types, but primarily we use Address record (A) type which maps hostnames to IPv4 address

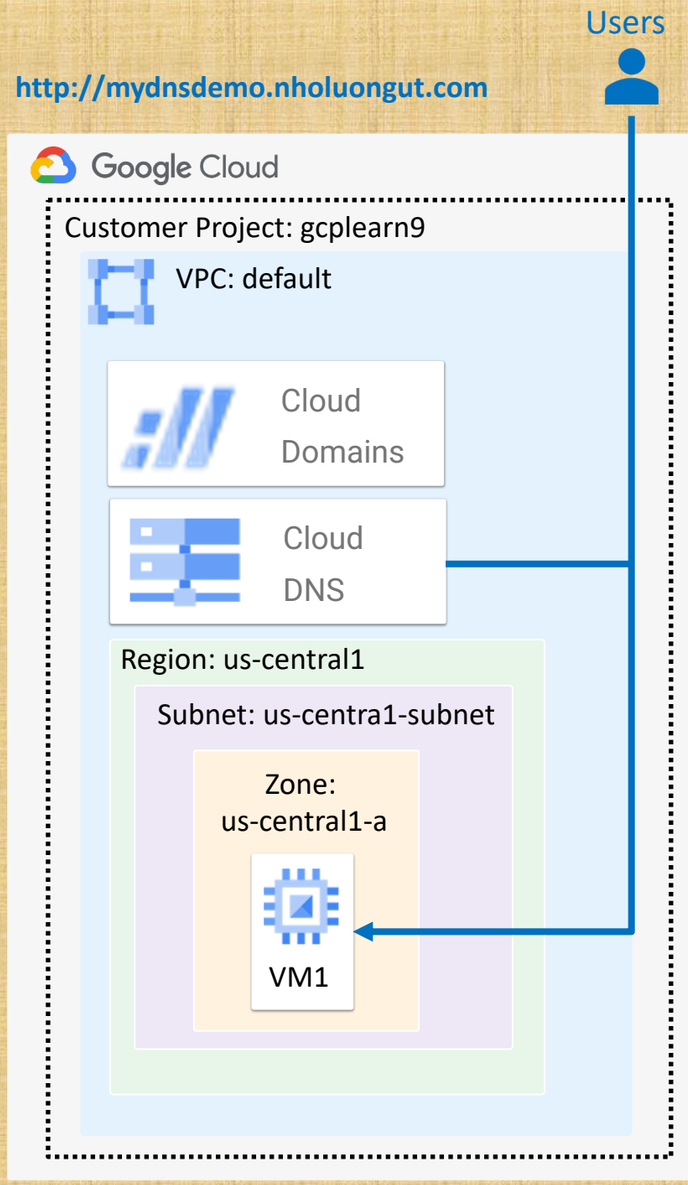
DNS Record Set



DNS Record Type



https://cloud.google.com/dns/docs/records-overview#supported_dns_record_types

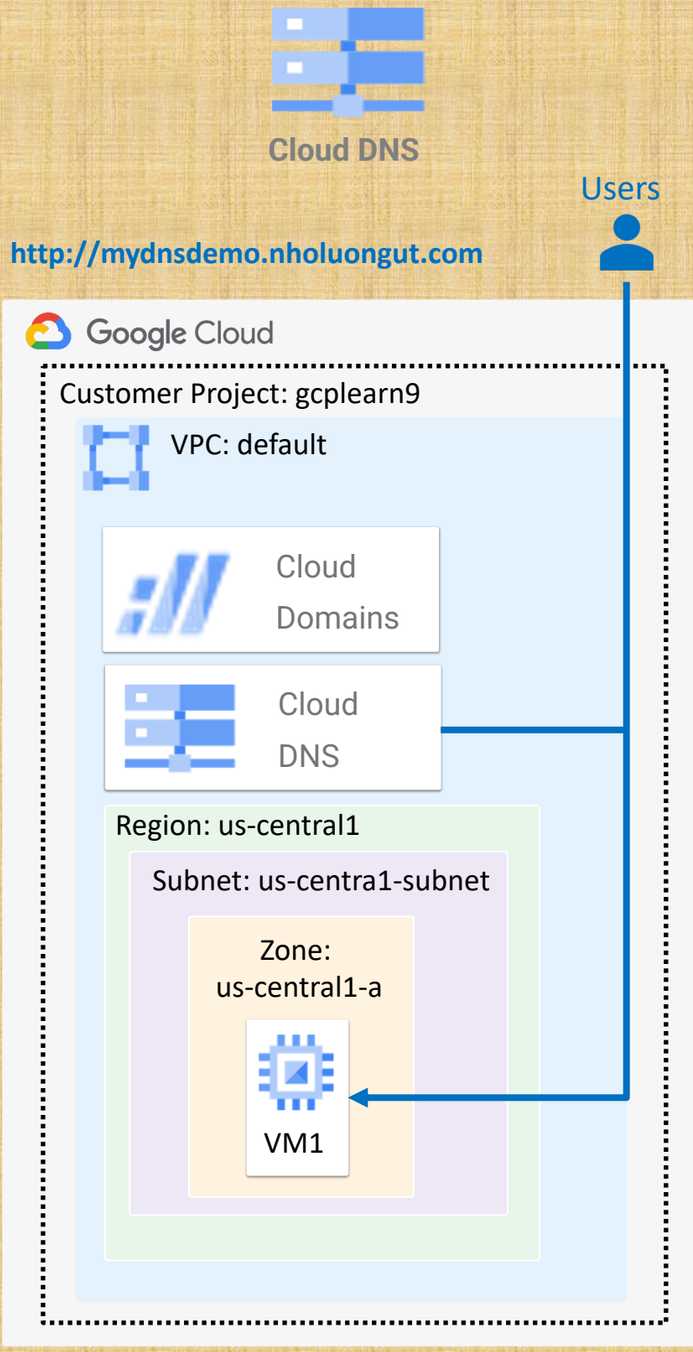


Author: Nho Luong

Skill: DevOps Engineer Lead

Google Cloud DNS - Features

- **Integration with Cloud IAM**
  - Provides secure domains management with full control and visibility for domain resources
- **Integration with Cloud Logging**
  - Private DNS logs a record for every DNS query in Cloud Logging
  - We can view and export logs to supported destinations
- **Fast anycast Nameservers**
  - **Anycast:** Uses the nearest nameserver to users for DNS lookup and resolution
  - Provides very high availability across globe and low latency
- **DNS registration and management**
  - Cloud DNS is tightly integrated with Cloud Domains which takes care of DNS registration and management

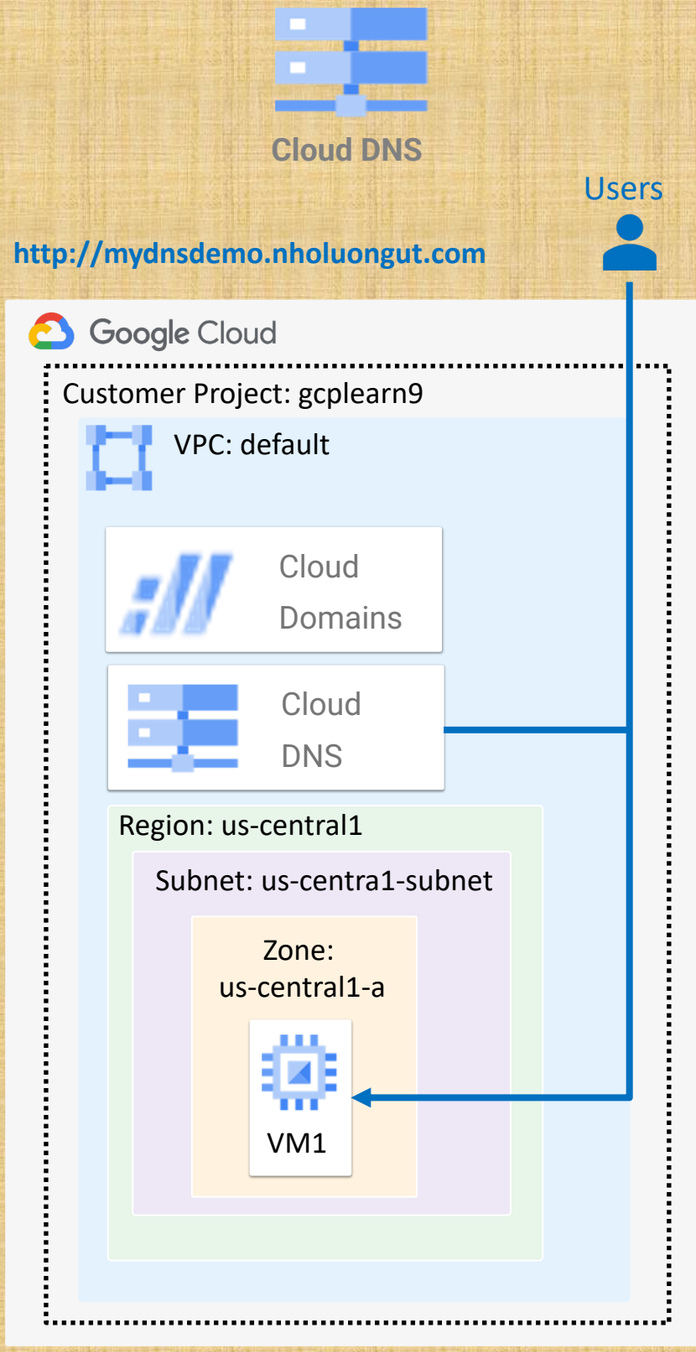


Author: Nho Luong

Skill: DevOps Engineer Lead

Google Cloud DNS - Features

- **Container-native Cloud DNS**
  - Natively integrated with Google Kubernetes Engine (GKE)
  - Provides in-cluster Service DNS resolution
  - Provides high-throughput, scalable DNS resolution for every GKE node
- **DNS Peering**
  - Sharing DNS data
  - All or a portion of DNS namespace can be configured to be sent from one network to another
  - Will respect all DNS configurations defined in peered network
- **DNS Forwarding**
  - Primarily used for hybrid-cloud architecture
  - Forward DNS queries from Cloud DNS to on-premise DNS servers where actual DNS resolution takes place



Author: Nho Luong

Skill: DevOps Engineer Lead

Demo-11

Google Kubernetes Engine

Gateway API

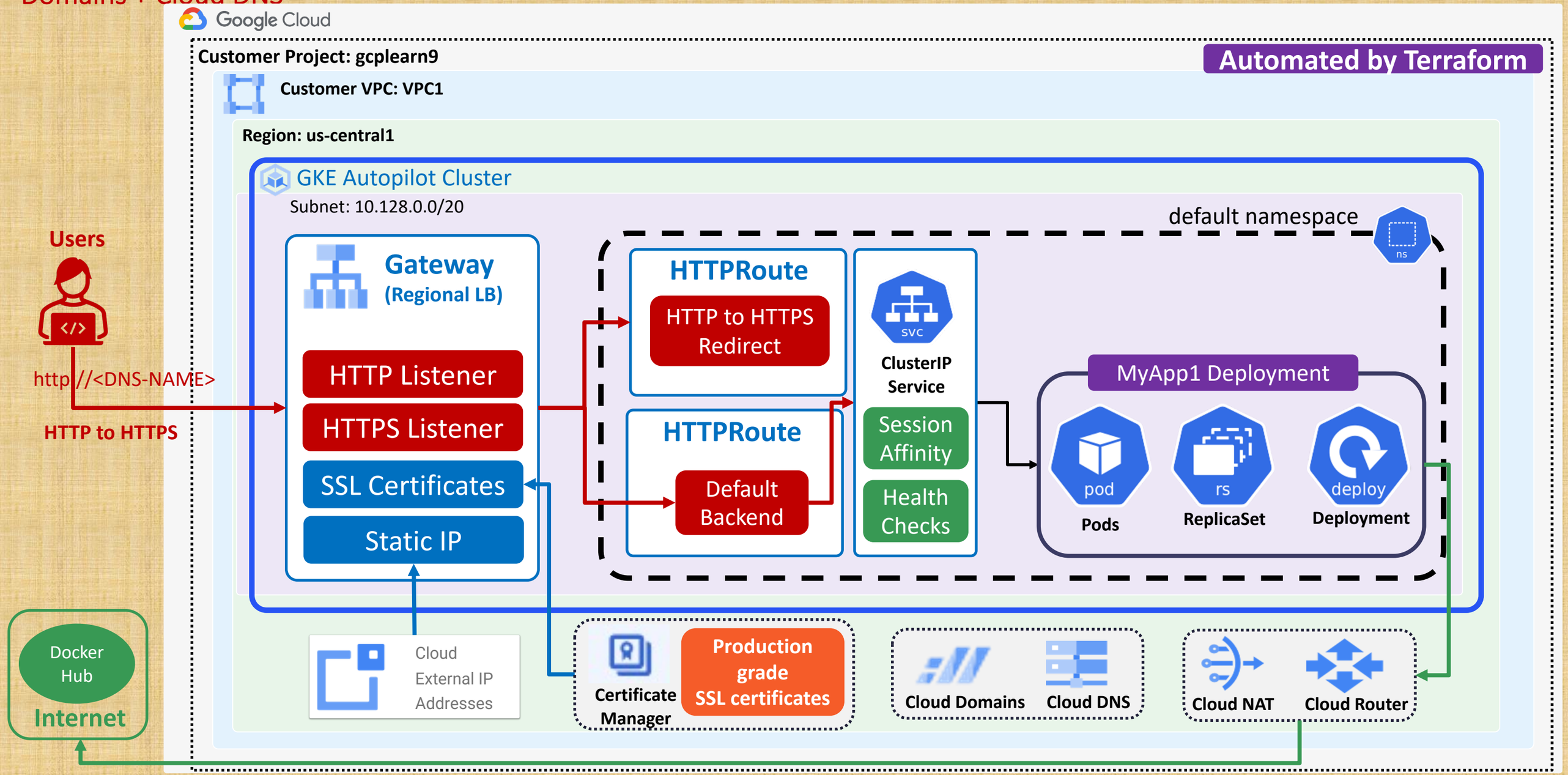
Cloud Domains + Cloud DNS

Demo: GKE Autopilot cluster + Gateway + Static IP + **PROD GRADE SSL with Certificate Manager** + HTTP to HTTPS Redirect + Health Checks + Session Affinity + **Cloud Domains + Cloud DNS**

Author: Nho Luong

Skill: DevOps Engineer Lead

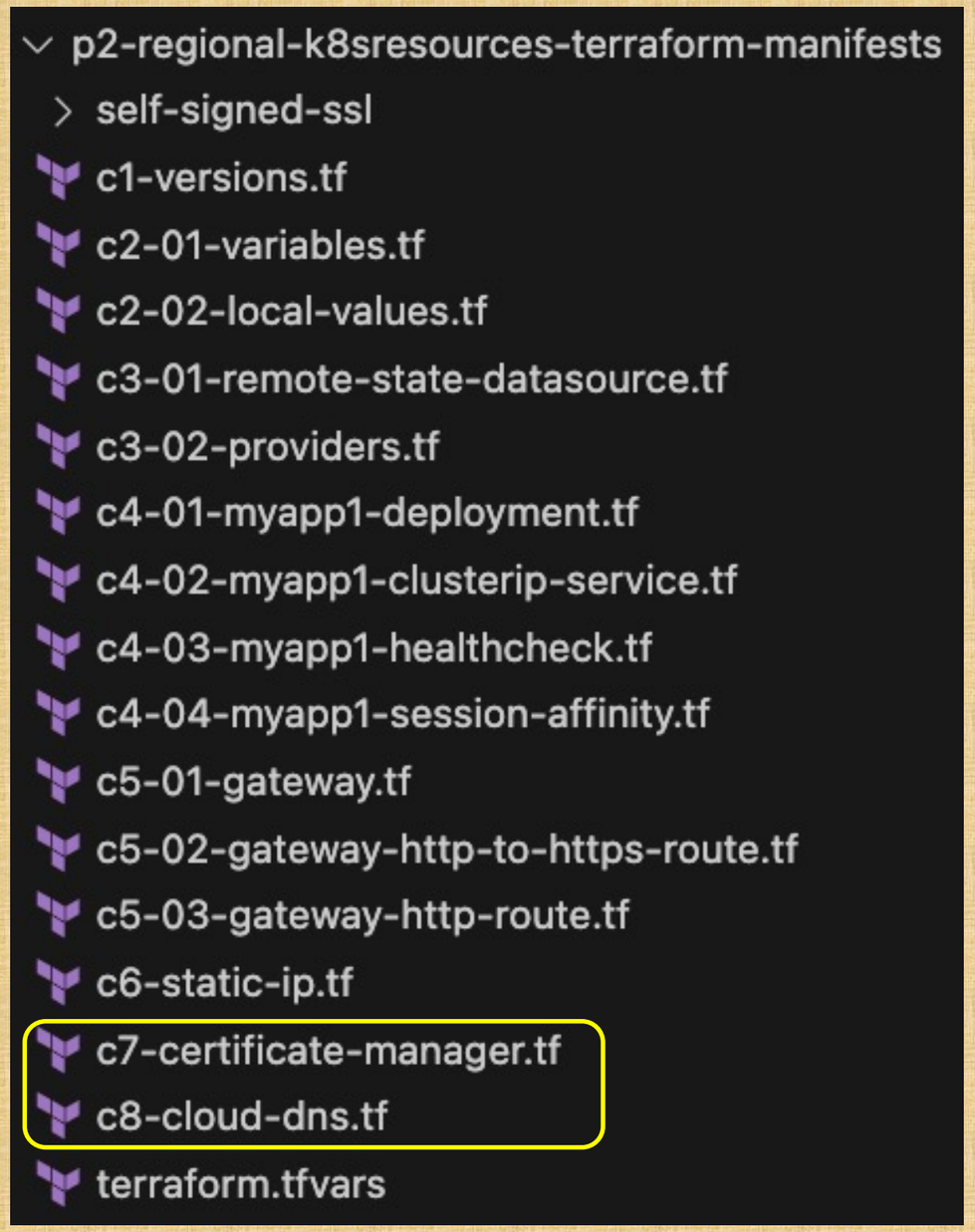
Gateway API + Static IP + SSL Certificate Manager + HTTP to HTTPS + Health Checks + Session Affinity + Cloud Domains + Cloud DNS



Author: Nho Luong

Skill: DevOps Engineer Lead

Project-2: Terraform Manifests



What are we going to learn?

Demo: Project-2: Gateway + Static IP + Prod grade SSL Certificate Manager + HTTP to HTTPS + Health Checks + Session Affinity + Cloud DNS + Cloud Domains

NC	No changes c1 to c6
C7	Create production grade SSL certificates using Certificate Manager with DNS Authorization
C8	Create record sets in Cloud DNS

Author: Nho Luong

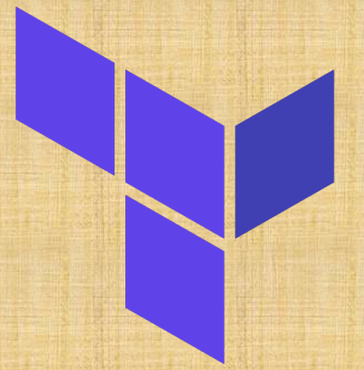
Skill: DevOps Engineer Lead

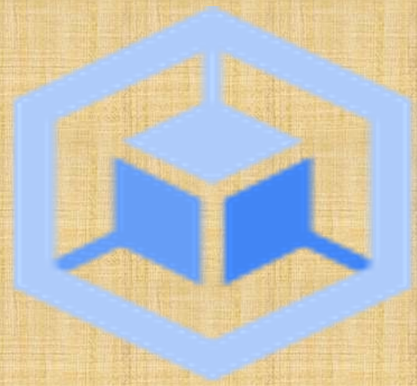
Demo-12

Google Kubernetes Engine

Gateway API

AWS Route53 + Cloud DNS



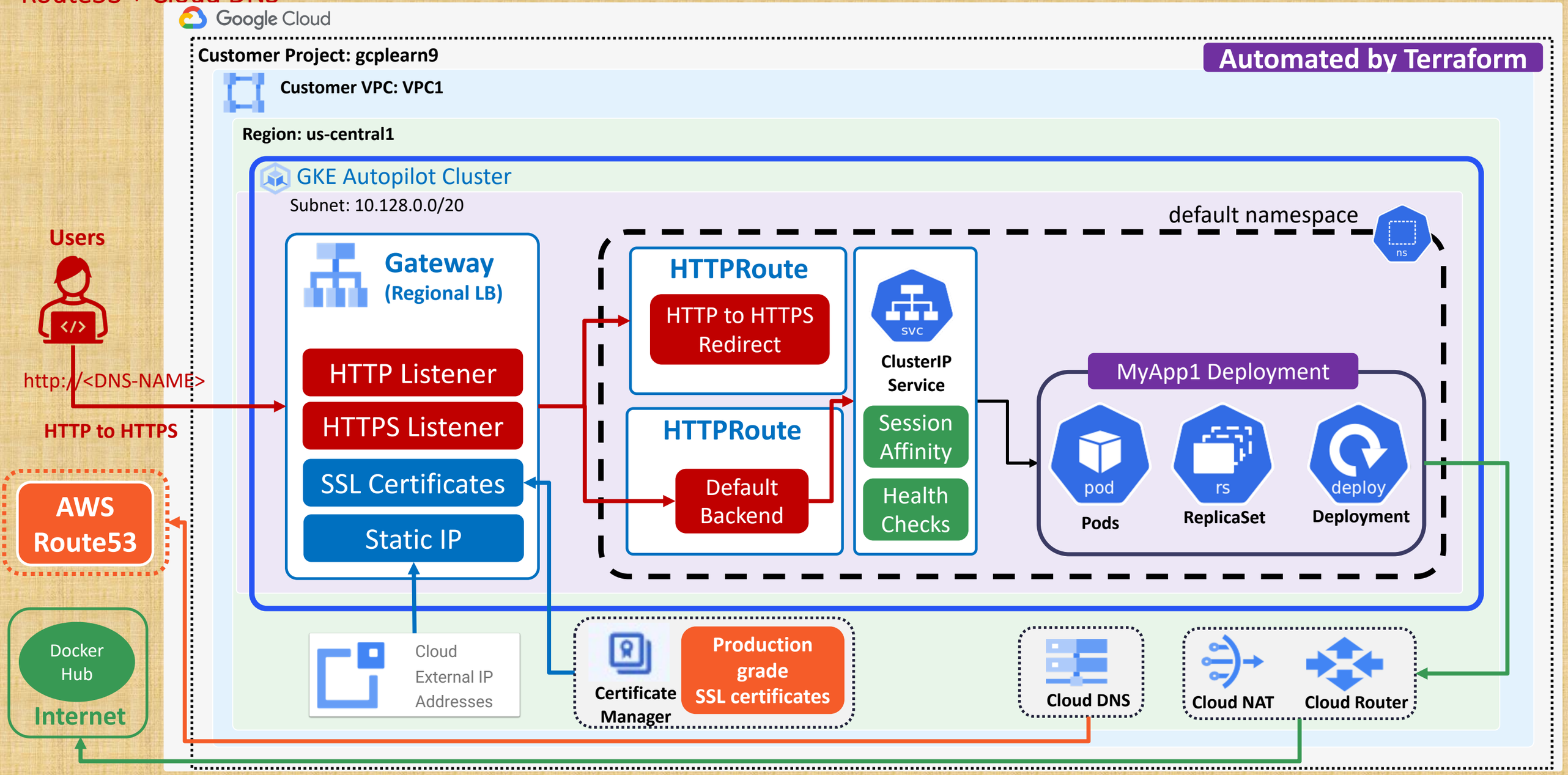


Demo: GKE Autopilot cluster + Gateway + Static IP + **PROD GRADE SSL** with Certificate Manager + HTTP to HTTPS Redirect + Health Checks + Session Affinity + **AWS Route53 + Cloud DNS**

Author: Nho Luong

Skill: DevOps Engineer Lead

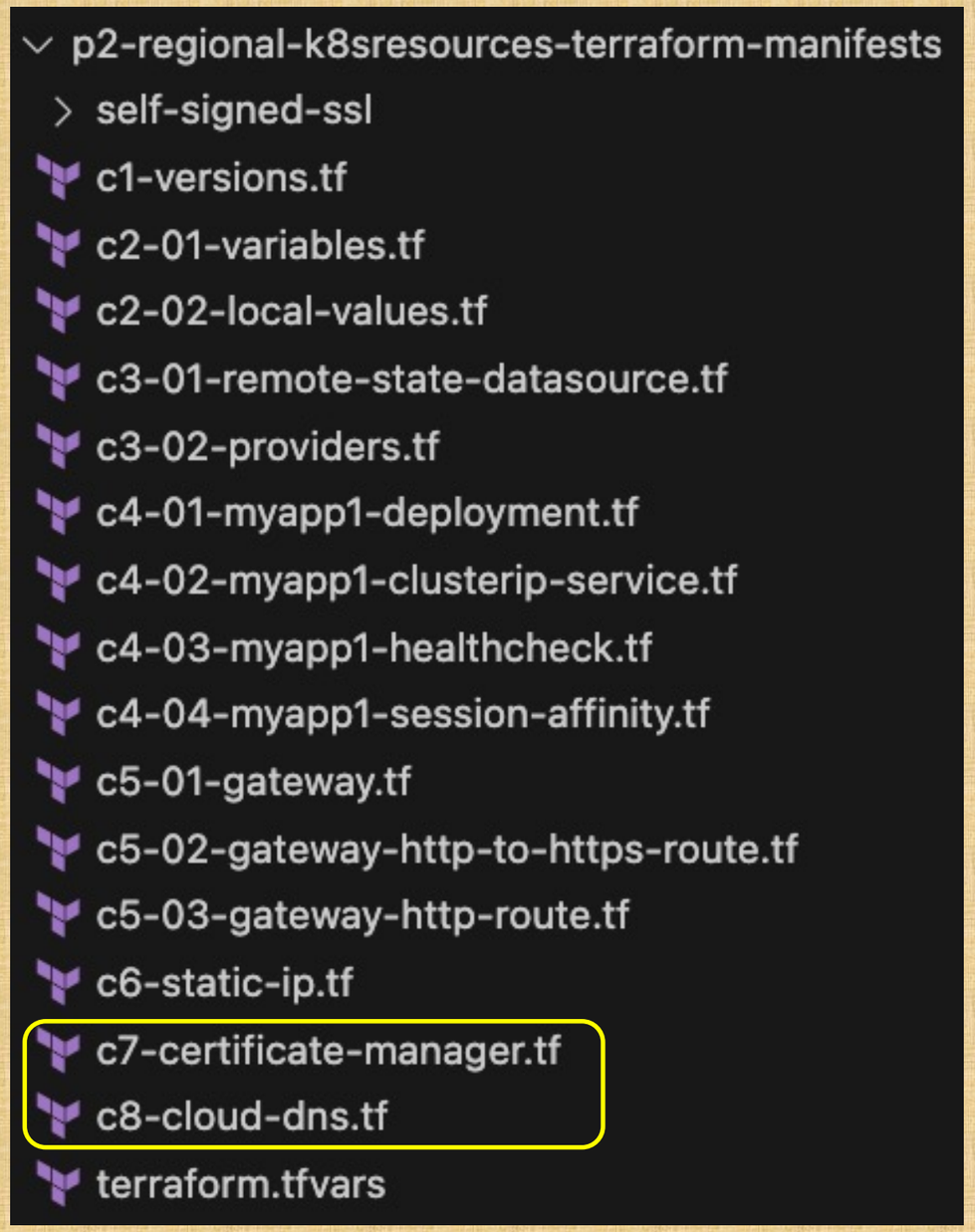
Gateway API + Static IP + SSL Certificate Manager + HTTP to HTTPS + Health Checks + Session Affinity + AWS Route53 + Cloud DNS



Author: Nho Luong

Skill: DevOps Engineer Lead

Project-2: Terraform Manifests



What are we going to learn?

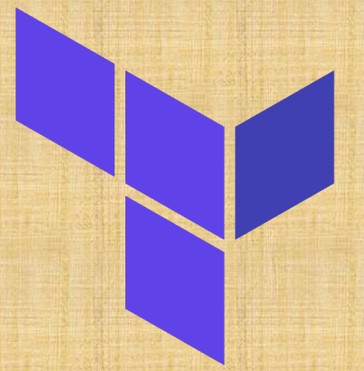
Demo: Project-2: Gateway + Static IP + Prod Grade SSL Certificate Manager + HTTP to HTTPS + Health Checks + Session Affinity + Cloud DNS + AWS Route53

NC	No changes c1 to c6
C7	Create production grade SSL certificates using Certificate Manager with DNS Authorization
C8	Create record sets in Cloud DNS

Author: Nho Luong

Skill: DevOps Engineer Lead

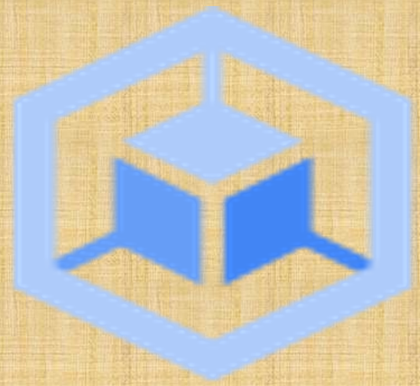
Demo-13

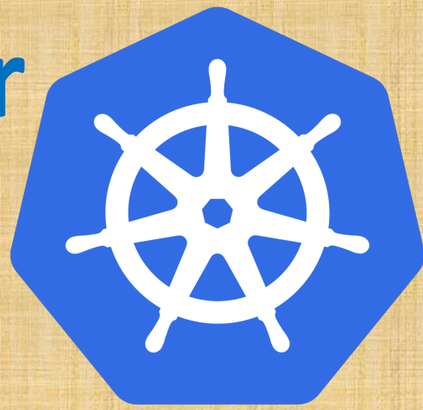


Google Kubernetes Engine

Gateway API

Cloud Application Load Balancer



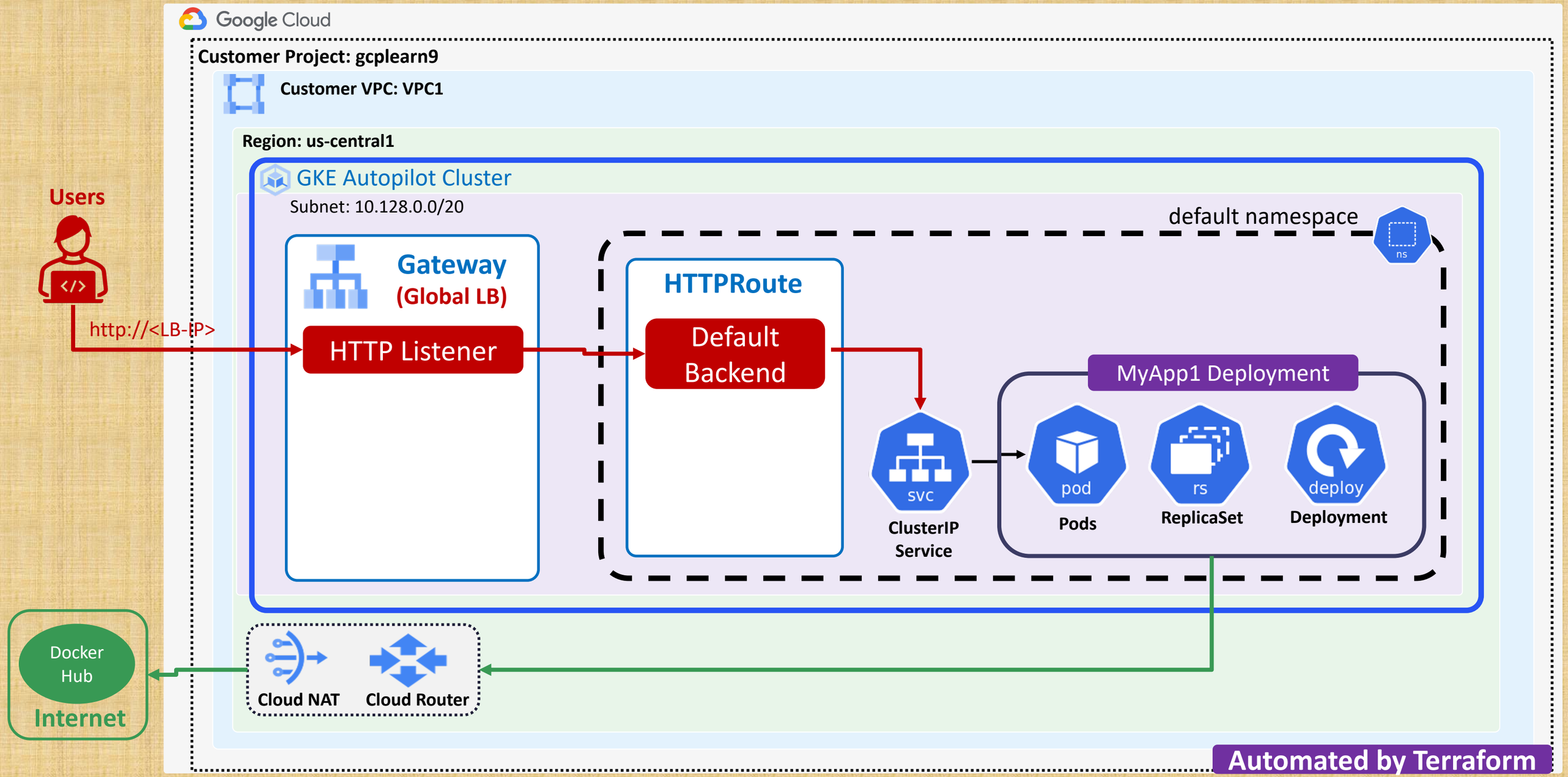


Demo: GKE Autopilot cluster + Gateway (Global Application Load Balancer)

Author: Nho Luong

Skill: DevOps Engineer Lead

Gateway API – Global Application Load Balancer

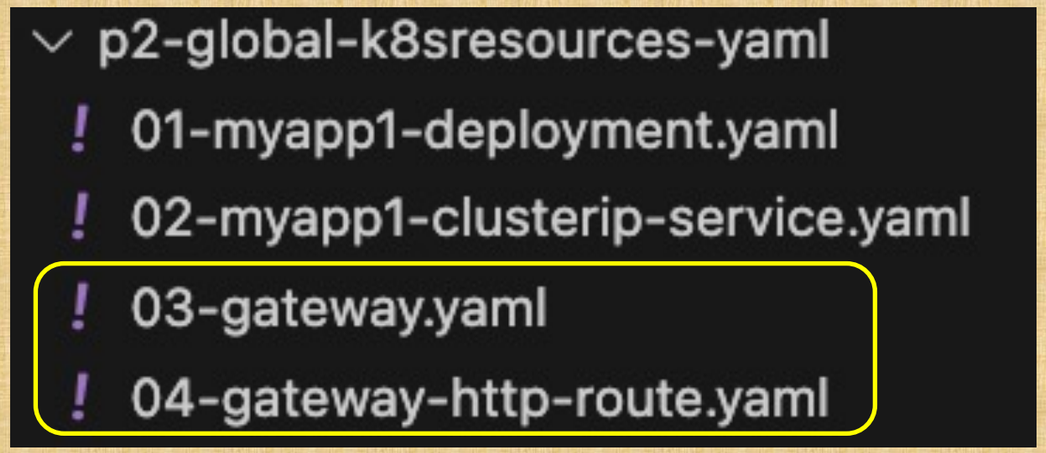


Author: Nho Luong

Skill: DevOps Engineer Lead

Project-2: YAML Manifests

What are we going to learn?



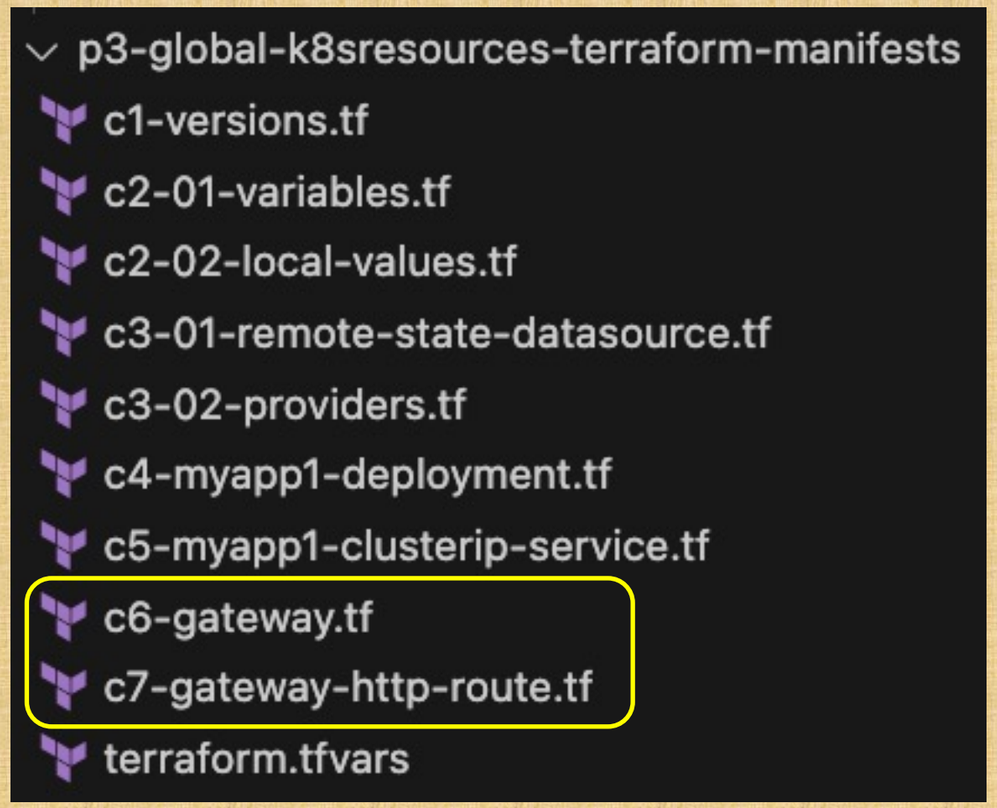
Demo: Project-2: Gateway (Global Application Load Balancer)

NC	NO Changes 01, 02
03	Gateway with gatewayClassName: gke-l7-global-external-managed
04	Gateway HTTPRoute

Author: Nho Luong

Skill: DevOps Engineer Lead

Project-3: Terraform Manifests



What are we going to learn?

Demo: Project-3: Gateway (Global Application Load Balancer)

NC	No changes c1 to c5
c6	Gateway with gatewayClassName: gke-l7-global-external-managed
c7	Gateway HTTPRoute

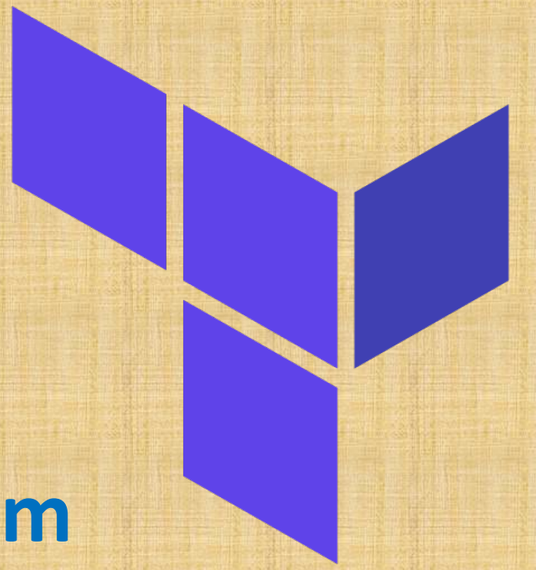
Author: Nho Luong

Skill: DevOps Engineer Lead

Demo-22

Terraform Modules

Use pre-built Modules from Terraform Registry



Demo: Custom VPC (Uses Terraform Module from Terraform Public Registry) + Firewall Rules + VM Instance

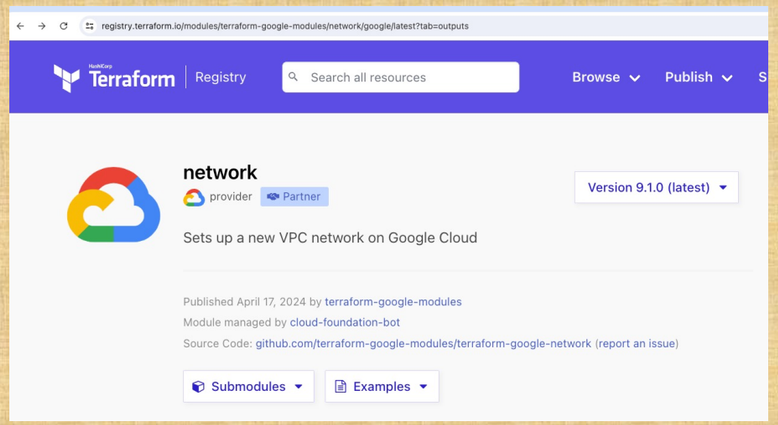
Author: Nho Luong

Skill: DevOps Engineer Lead

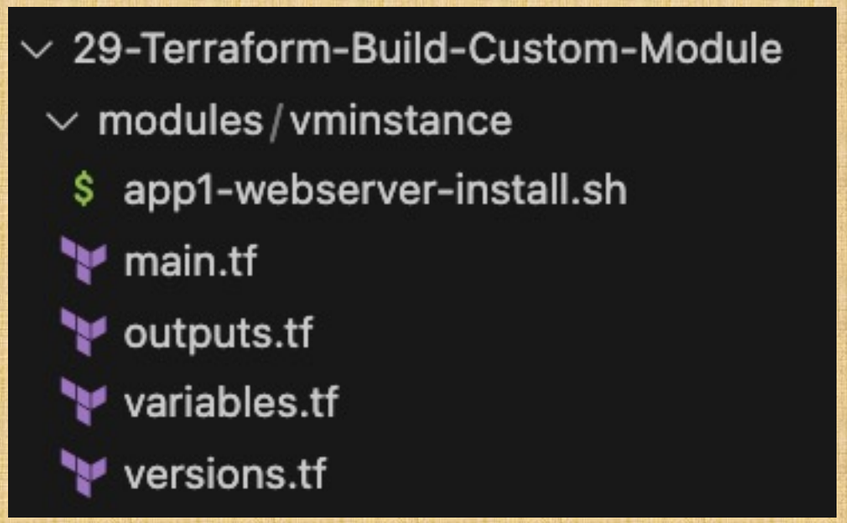
Terraform Modules

- A module consists of a collection of .tf files kept together in a directory.
- Modules are the main way to package and reuse resource configurations with Terraform.
- We can compare modules to functions in our standard programming languages (define once and reuse wherever needed)
- **Example:**
  - Define a VPC Module once, publish it to a Terraform registry
  - Call the VPC module wherever needed (reuse it)

Public Module on Terraform Registry



Custom Module

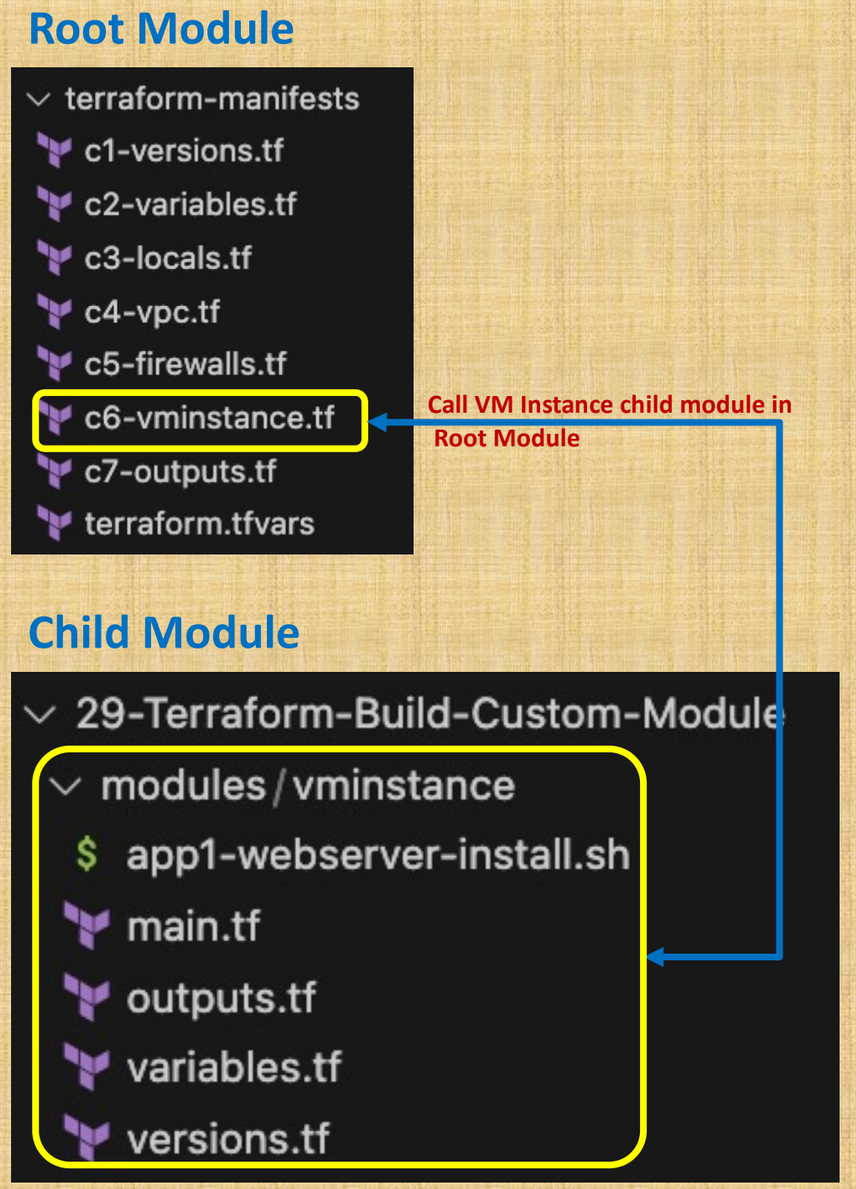


Author: Nho Luong

Skill: DevOps Engineer Lead

Terraform Modules

- **Root Module:** Every Terraform configuration has at least one module, known as its root module, which consists of the resources defined in the .tf files in the main working directory
- A Terraform module (usually the root module of a configuration) can call other modules to include their resources into the configuration.
- **Child Module:** A module that has been called by another module is often referred to as a child module.
- Child modules can be called multiple times within the same configuration, and multiple configurations can use the same child module.



Author: Nho Luong

Skill: DevOps Engineer Lead

Terraform Modules

- **Module Types**
  - **Local Modules (Local Paths)**
    - Accessible using folder path locally
    - Accessible locally in that desktop or for others via shared drives
  - **Public Modules**
    - Modules published to Terraform Registry
    - Publicly accessible to anyone (Open source)
    - Free to use, use with caution, will contain continuous upgrades and might break our current code
    - If used, need to follow strong version constraints or fixed module versions to avoid major outages during upgrades
  - **Private Modules**
    - Modules developed and published to private registry in their respective organizations
    - Terraform Registry also supports private registry concept
    - Accessible only to that respective organization users (Highly secure, internal to that organization)

Author: Nho Luong

Skill: DevOps Engineer Lead

Terraform Modules

- **Module Storage and Access:** The module installer supports installation from several different source types.
  - Local Paths
  - Terraform Registry
  - GitHub
  - HTTP URLs
  - S3 Buckets
  - GCS Buckets
  - and many more

Local paths

```
module "consul" {
  source = "./consul"
}
```

TF Registry

```
module "vpc" {
    source  = "terraform-google-modules/network/google"
    version = "~> 9.1"
```

GitHub

```
module "consul" {
  source = "github.com/hashicorp/example"
}
```

HTTP URL

```
module "vpc" {
  source = "https://example.com/vpc-module.zip"
}
```

S3 Bucket

```
module "consul" {
  source = "s3::https://s3-eu-west-1.amazonaws.com/examplecorp-terraform-modules/vpc.
}
```

GCS Bucket

```
module "consul" {
  source = "gcs::https://www.googleapis.com/storage/v1/modules/foomodule.zip"
}
```

Author: Nho Luong

Skill: DevOps Engineer Lead

Terraform Registry - Use Publicly Available Modules

The Terraform Registry hosts a broad collection of publicly available Terraform modules for configuring many kinds of common infrastructure.

Modules Demo 1

These modules are free to use, and Terraform can download them automatically if you specify the appropriate source and version in a module call block.

```
# Module: VPC
module "vpc" {
    source  = "terraform-google-modules/network/google"
    version = "~> 9.1"
    project_id   = var.gcp_project
    network_name = "${local.name}-vpc"
    routing_mode = "GLOBAL"
    subnets = [
        {
            subnet_name           = "${local.name}-${var.gcp_region1}-subnet"
            subnet_ip             = "10.128.0.0/20"
            subnet_region         = var.gcp_region1
        }
    ]
}
```

Author: Nho Luong

Skill: DevOps Engineer Lead

What are we going to learn?

Demo: Custom VPC (Uses Terraform Module from Terraform Public Registry) + Firewall Rules + VM Instance

NC	No Changes from C1, C2, C3
C4	Create VPC Module using Terraform public registry
C5	Update firewall rules with VPC ID from VPC Module
C6	Update Subnet with Subnet ID from VPC Module
C7	Update VPC ID and Subnet ID from VPC Module

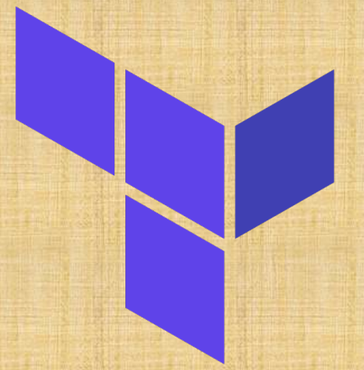
Author: Nho Luong

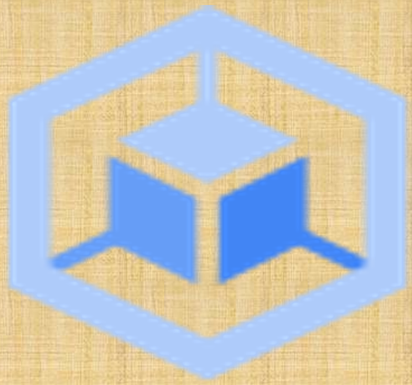
Skill: DevOps Engineer Lead

Demo-23

Google Kubernetes Engine

Terraform Custom Module





Demo: Create GKE Autopilot Cluster by building a custom Terraform Module

Author: Nho Luong

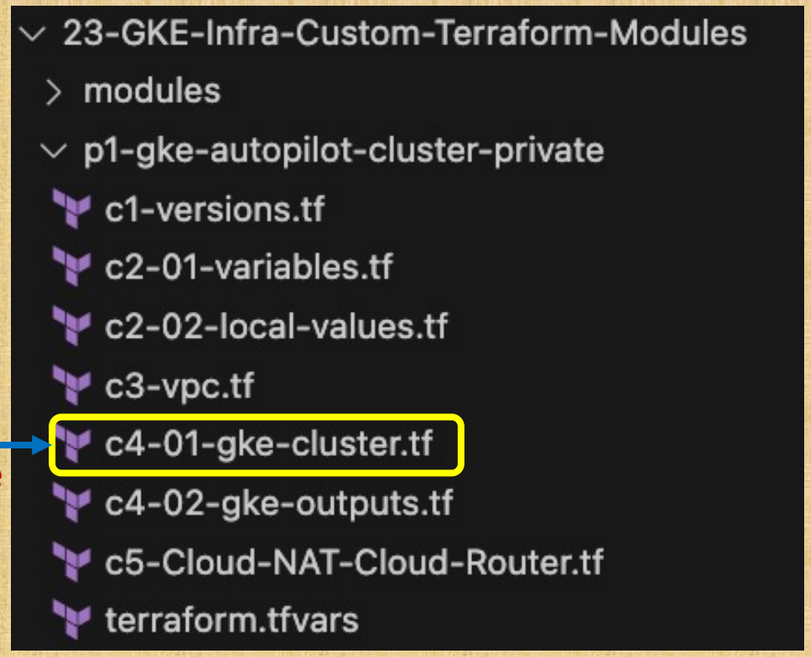
Skill: DevOps Engineer Lead

Build a Local Terraform Module

Modules
Demo 2

Build a Local Terraform Module and call it from Root Module and Test

Root Module



Call GKE Cluster child module in Root Module

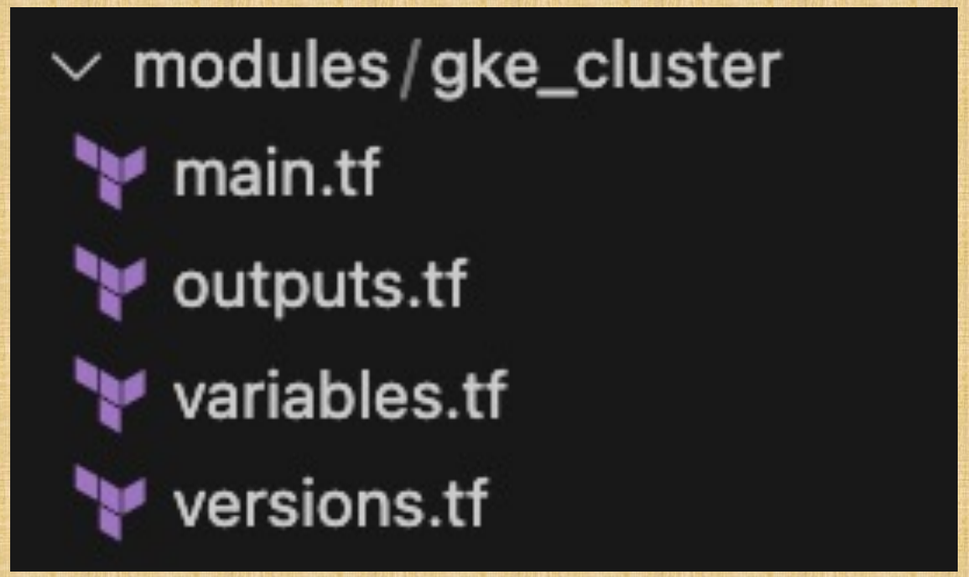
Child Module

Author: Nho Luong

Skill: DevOps Engineer Lead

Modules: GKE Cluster Custom Module

What are we going to learn?



Demo: Create GKE Cluster Custom Module

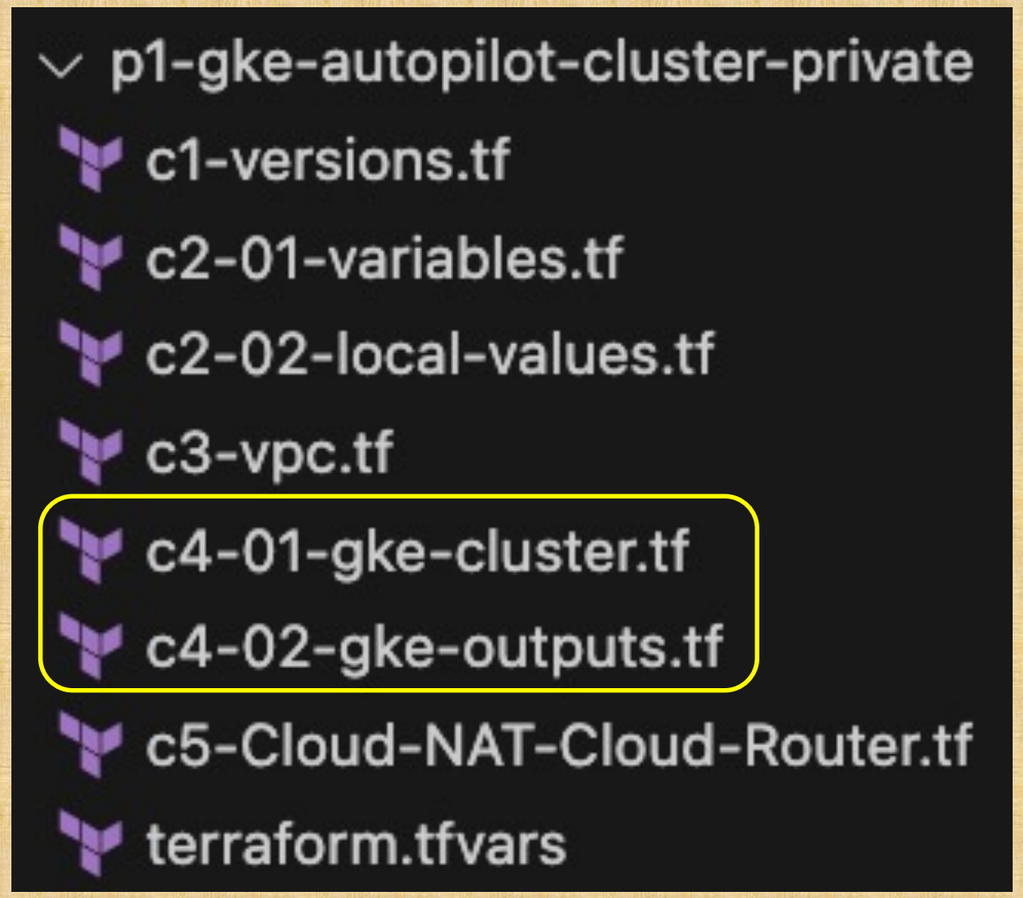
versions	Define Terraform Provider versions
variables	Define Terraform Input Variables
main	Create GKE Cluster Terraform resource
outputs	Define Terraform Outputs

Author: Nho Luong

Skill: DevOps Engineer Lead

Project-1: Terraform Manifests

What are we going to learn?



Demo: Project-1: Create GKE Autopilot Cluster by calling a custom Terraform Module

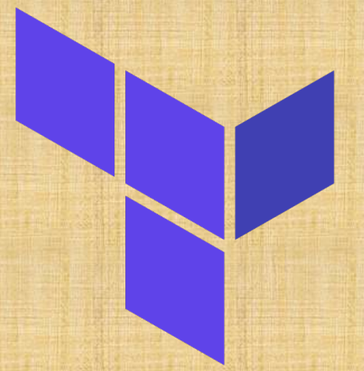
NC	No changes c1 to c3, c5
c4-01	Create GKE cluster by calling custom gke_cluster Terraform module
c4-02	Define GKE Terraform outputs

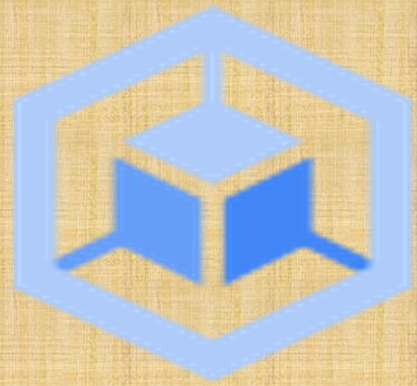
Author: Nho Luong

Skill: DevOps Engineer Lead

Demo-24

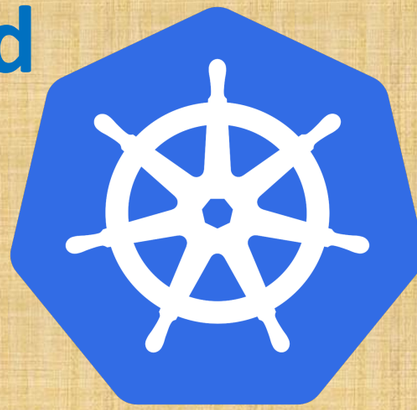
Google Kubernetes Engine





GKE Infra DevOps

GCP Cloud Build, GitHub GCP Cloud Build App and GitHub



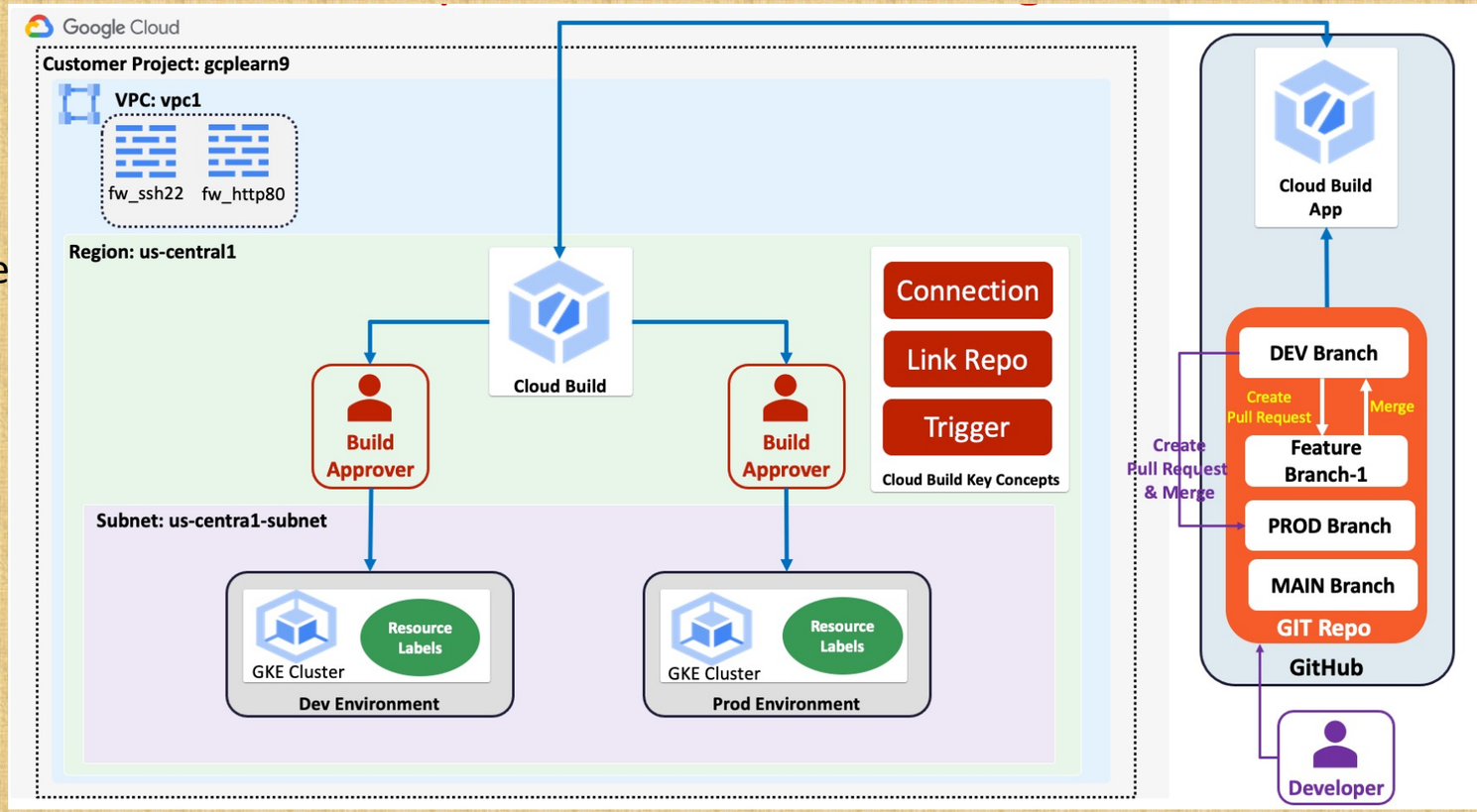
Demo: Custom VPC + + GKE Cluster (Custom or Local Terraform Module)
Implement DevOps Pipelines for Terraform Configs on GCP GKE

Author: Nho Luong

Skill: DevOps Engineer Lead

GCP GKE DevOps for Terraform Configs

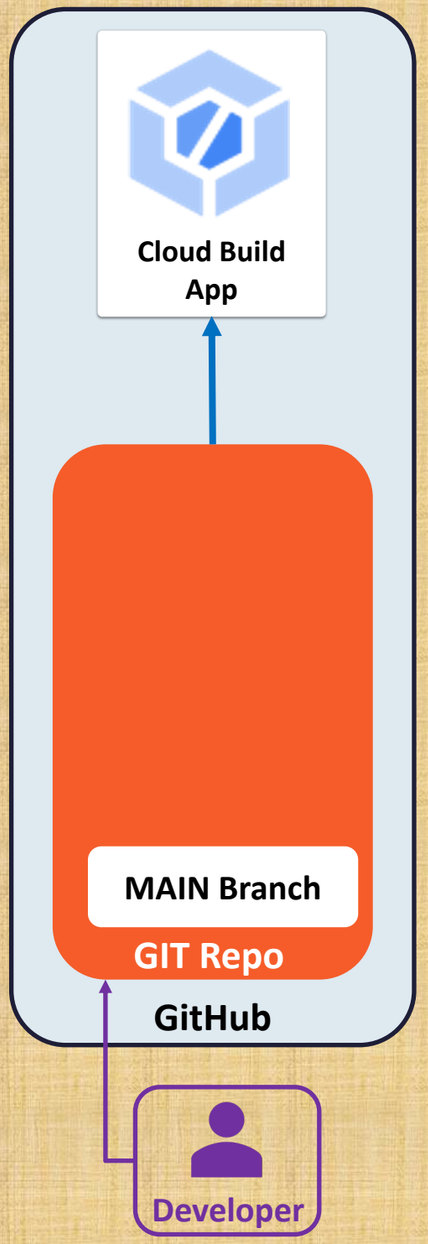
- **The following resources are used for this implementation**
- **GitHub Repositories**
  - GitHub Branches
    - Main branch
    - Dev branch
    - Prod branch
    - Feature branch
  - GitHub Pull Request and Merge concepts
  - GCP Cloud Build App from GitHub Marketplace
- **GCP Cloud Build**
  - Connections
  - Link Repositories
  - Triggers
- **GCP Resources**
  - GCP VPC
  - GCP Firewall Rules
  - GCP GKE Autopilot cluster
- **Terraform Concepts**
  - Terraform Resources
  - Terraform custom modules
  - Terraform Remote Backend using Google Cloud Storage Bucket



Author: Nho Luong

Skill: DevOps Engineer Lead

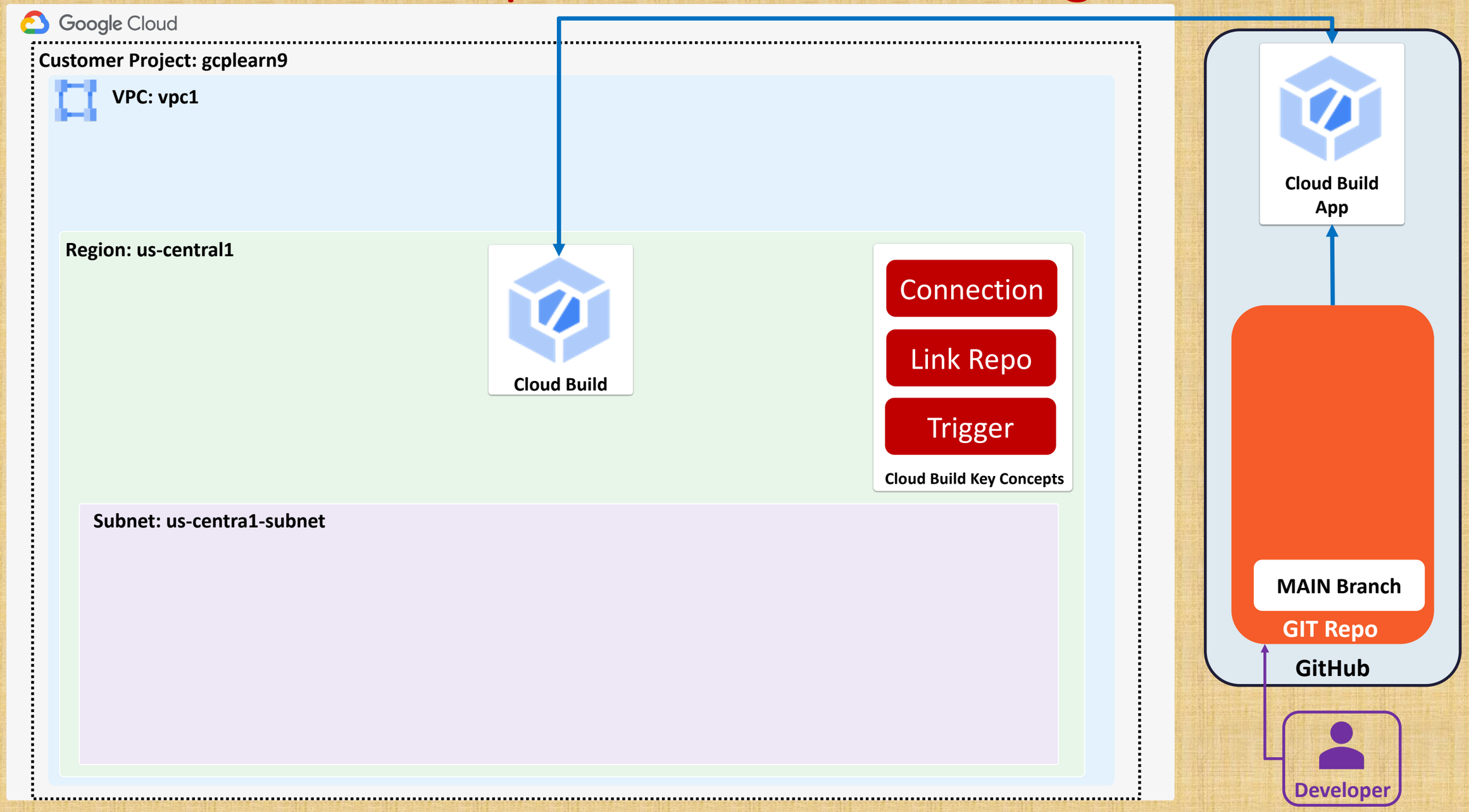
GCP GKE DevOps for Terraform Configs



Author: Nho Luong

Skill: DevOps Engineer Lead

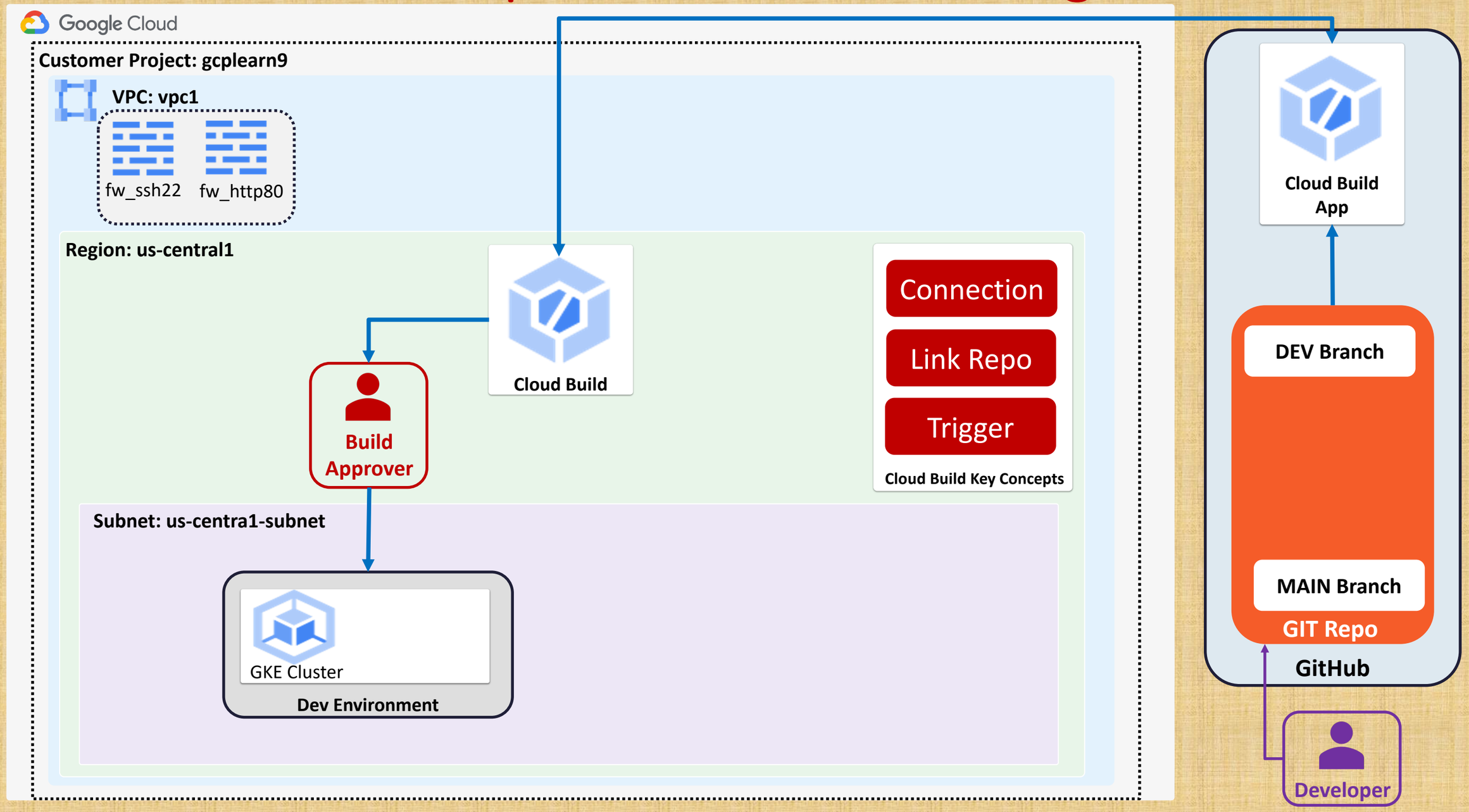
GCP GKE DevOps for Terraform Configs



Author: Nho Luong

Skill: DevOps Engineer Lead

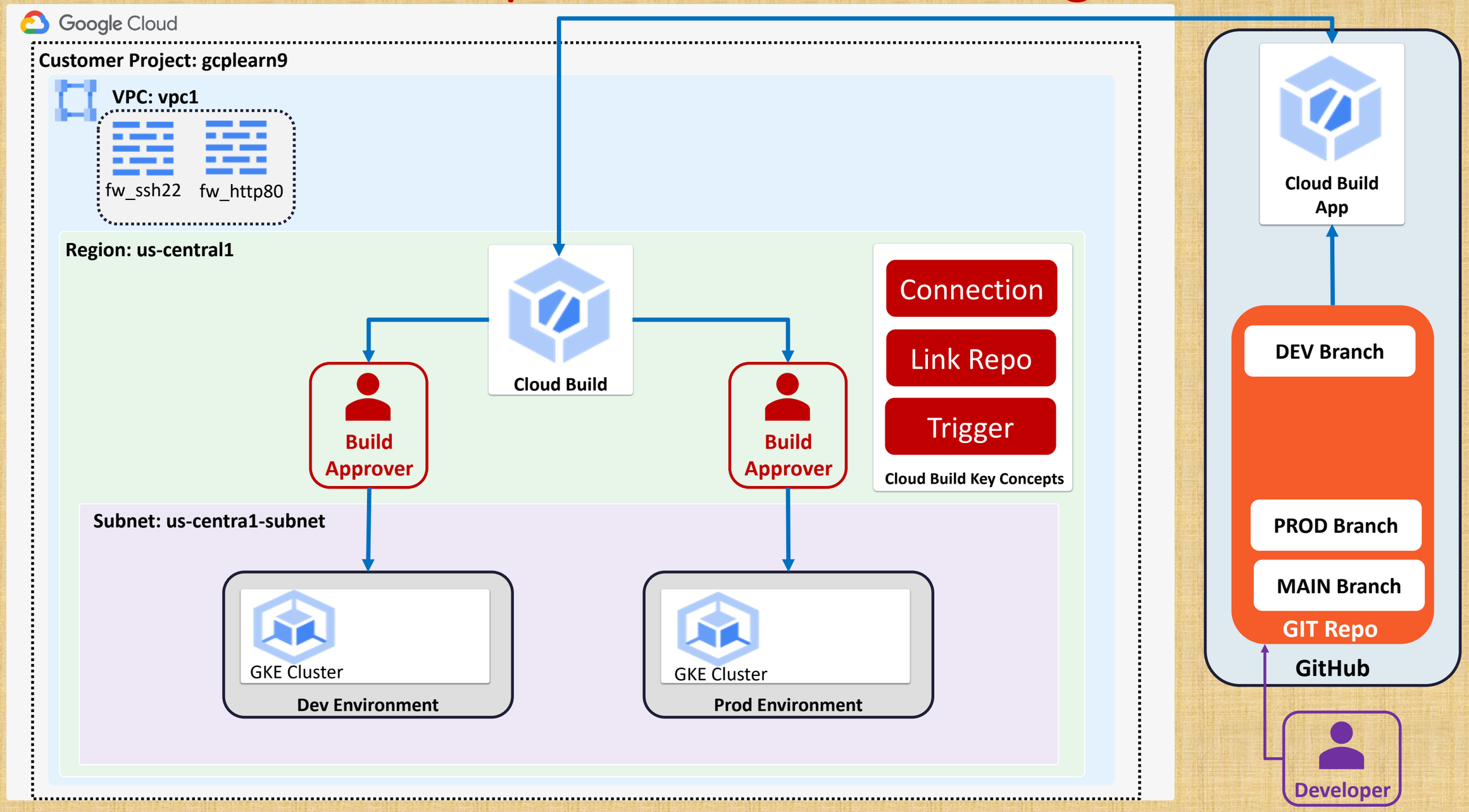
GCP GKE DevOps for Terraform Configs



Author: Nho Luong

Skill: DevOps Engineer Lead

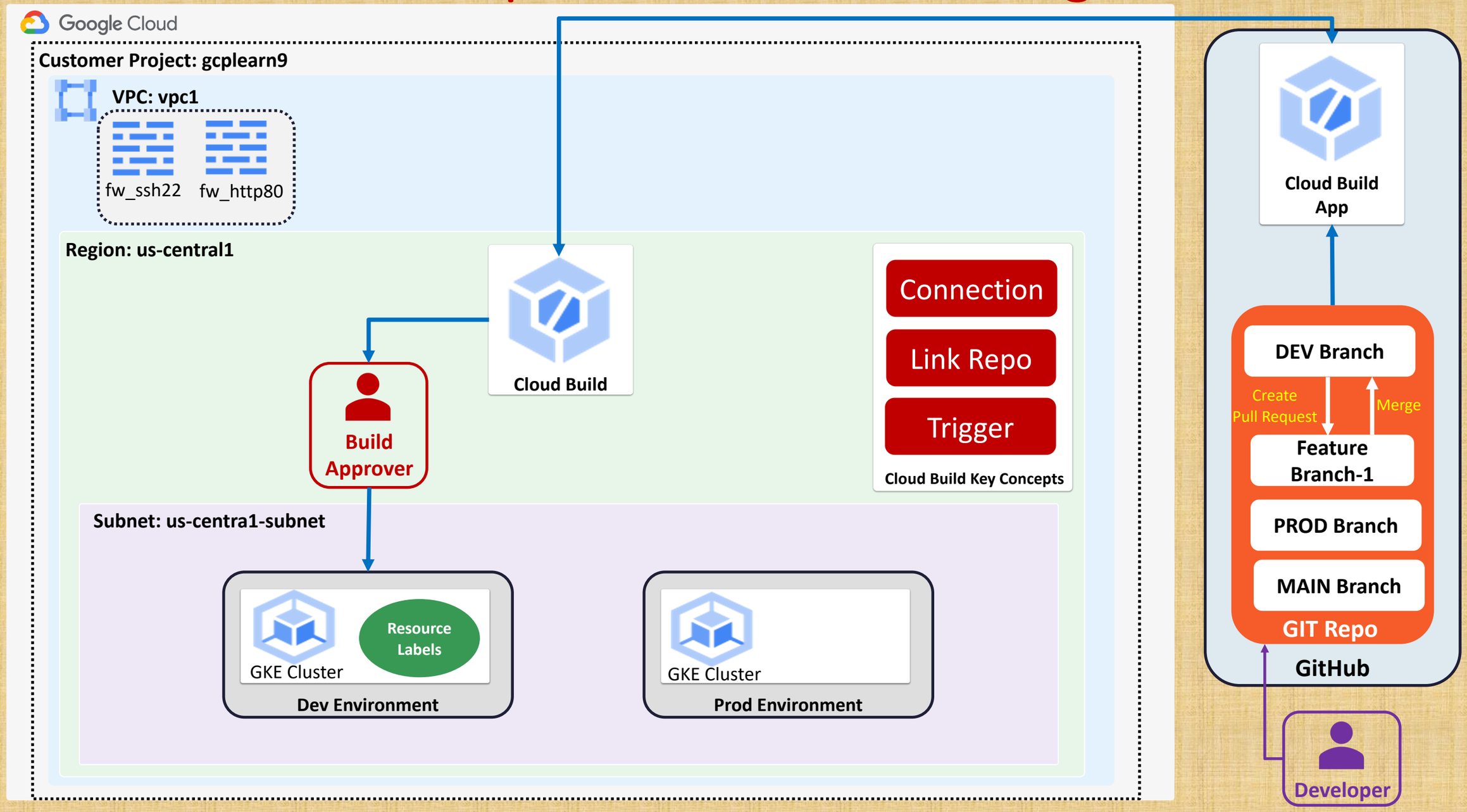
GCP GKE DevOps for Terraform Configs



Author: Nho Luong

Skill: DevOps Engineer Lead

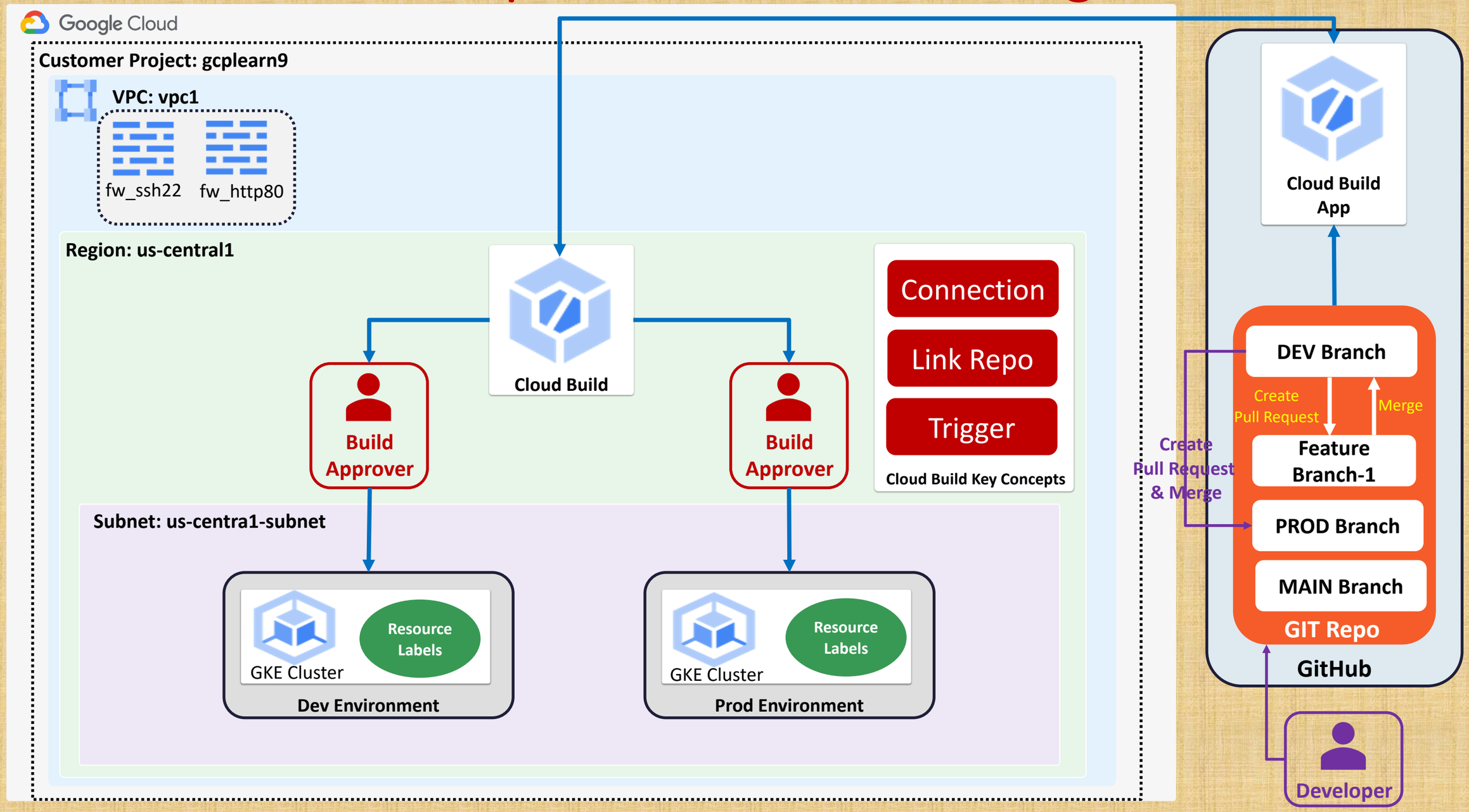
GCP GKE DevOps for Terraform Configs



Author: Nho Luong

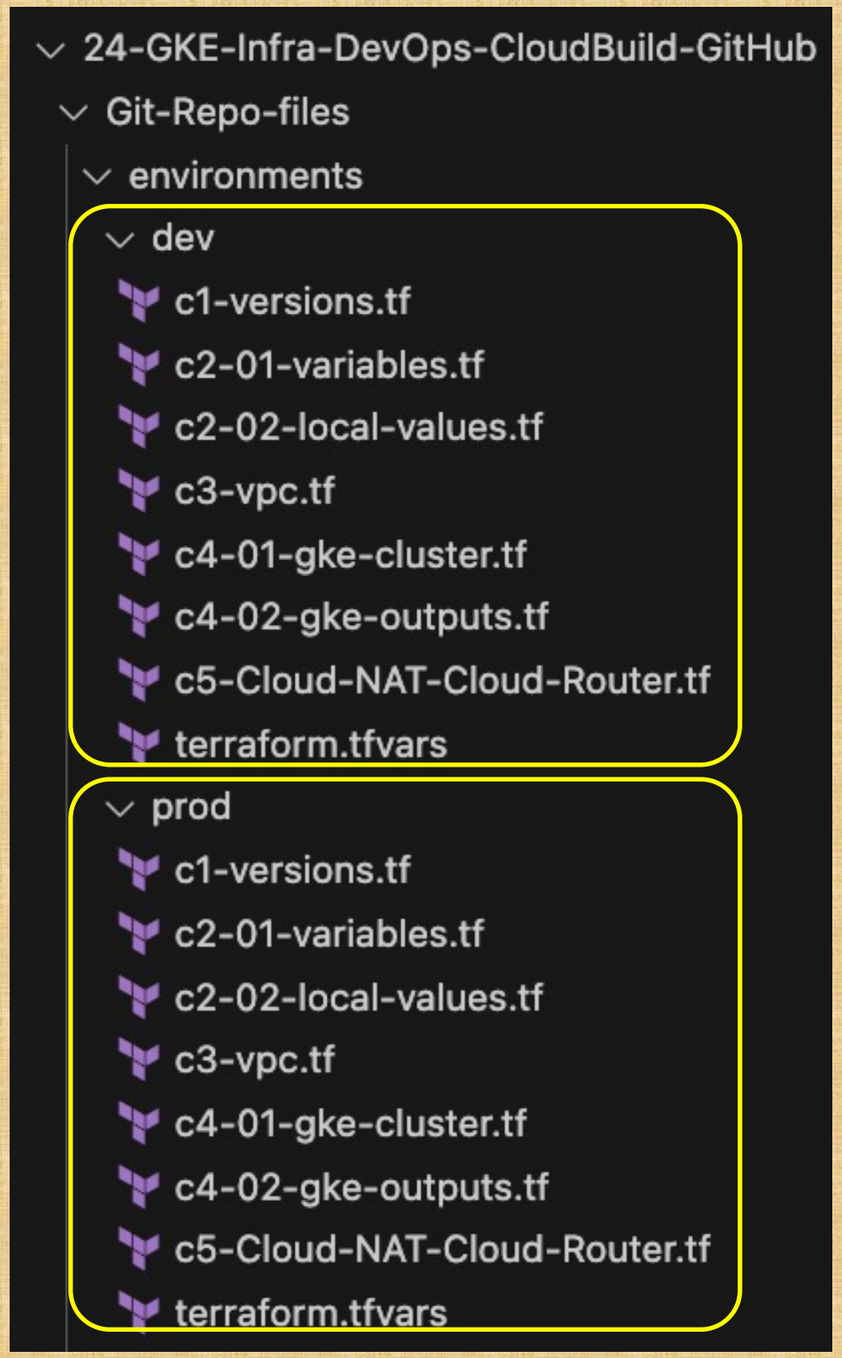
Skill: DevOps Engineer Lead

GCP GKE DevOps for Terraform Configs



Author: Nho Luong

Skill: DevOps Engineer Lead



What are we going to learn?

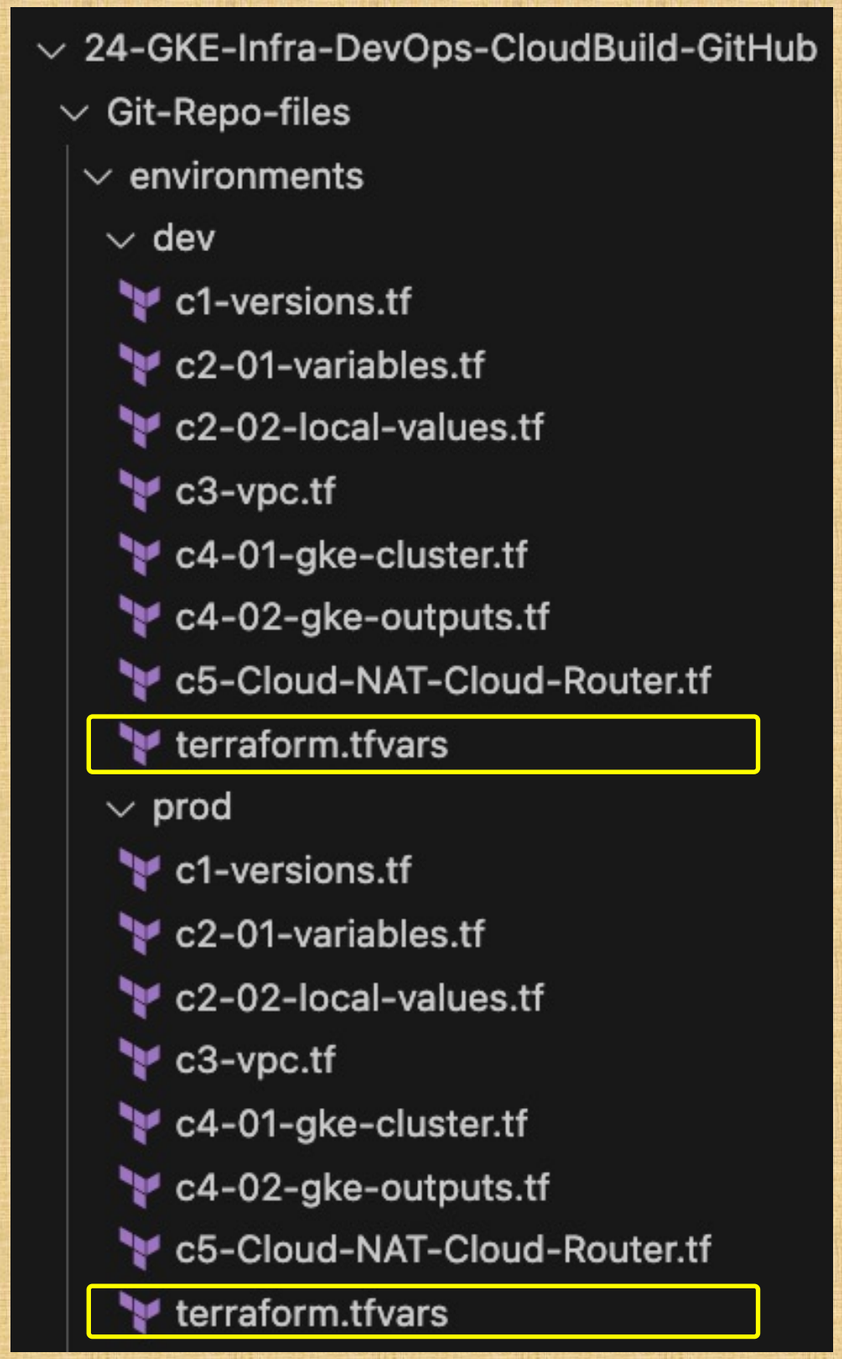
Demo: Custom VPC + GKE Cluster(Custom or Local Terraform Module)
Implement DevOps Pipelines for Terraform Configs on GCP GKE

All these TF Configs are copy from Root Module Demo-23 (p1-gke-autopilot-cluster-private folder)

DEV	**Development Environment**
C1	Update GCS Bucket (your bucket name) and State file prefix as “dev/gke-cluster”
TFVARS	Ensure (environment = "dev”) in terraform.tfvars file
PROD	**Production Environment**
C1	Update GCS Bucket (your bucket name) and State file prefix as “prod/gke-cluster”
TFVARS	Ensure (environment = ”prod”) in terraform.tfvars file

Author: Nho Luong

Skill: DevOps Engineer Lead



What are we going to learn?

Demo: Custom VPC + GKE Cluster(Custom or Local Terraform Module)
Implement DevOps Pipelines for Terraform Configs on GCP GKE

All these TF Configs are copy from Root Module Demo-23 (p1-gke-autopilot-cluster-private folder)

DEV — **Development Environment**

TFVARS — Update all CIDR ranges by ensuring it doesn't overlap with other environments

PROD — **Production Environment**

TFVARS — Update all CIDR ranges by ensuring it doesn't overlap with other environments

Author: Nho Luong

Skill: DevOps Engineer Lead

What are we going to learn?

Demo: Custom VPC + GKE Cluster(Custom or Local Terraform Module)
Implement DevOps Pipelines for Terraform Configs on GCP GKE

All these TF Configs are copy from Child Module Demo-23 (modules folder)

CHILD MODULE

GKE Cluster Module we have created in Demo-23

cloud build

cloudbuild.yaml contains Terraform commands with DevOps pipeline code

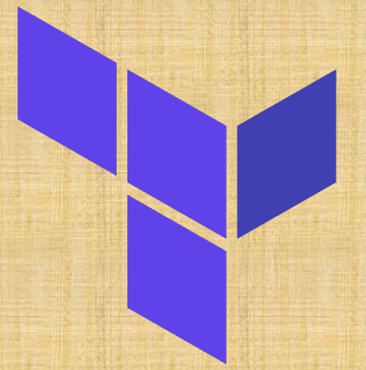
Author: Nho Luong

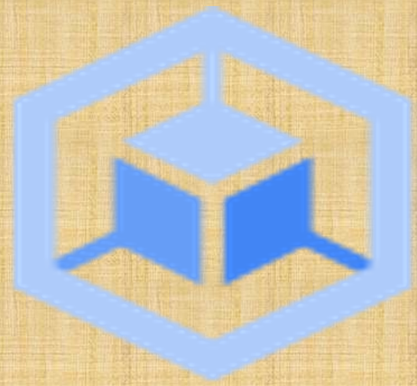
Skill: DevOps Engineer Lead

Demo-25, 26, 27, 28

Google Kubernetes Engine

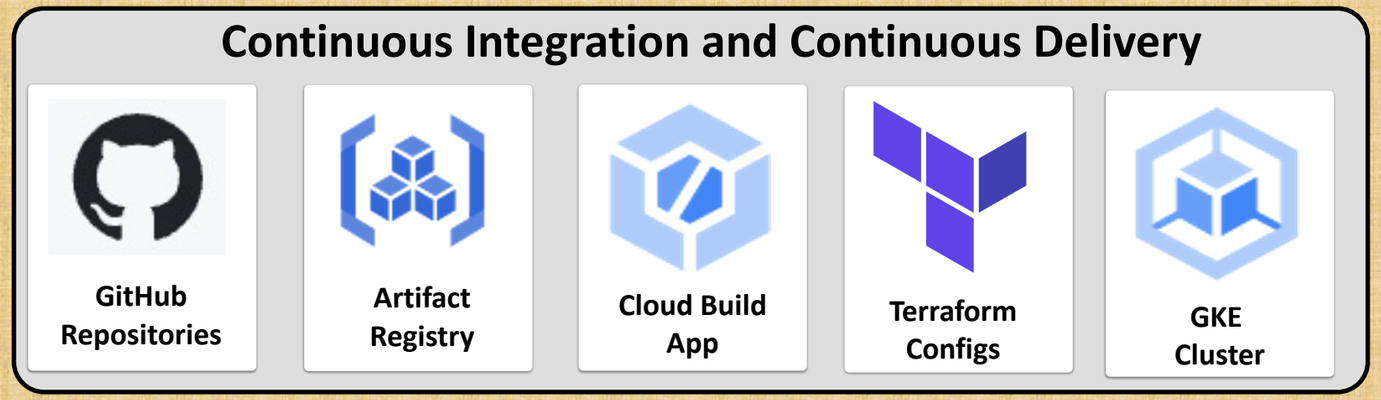
Continuous Integration
Continuous Delivery





Author: Nho Luong

Skill: DevOps Engineer Lead



Demo 25

Create Kubernetes Deployment Custom Terraform Module

Demo 26

Implement k8s workload DevOps pipelines
(Dev and Prod Environments)

Demo 27

App: Implement Continuous Integration Pipeline with Artifact Registry

Demo 28

App: Implement Continuous Integration and Continuous Delivery Pipeline (CI CD)

Author: Nho Luong

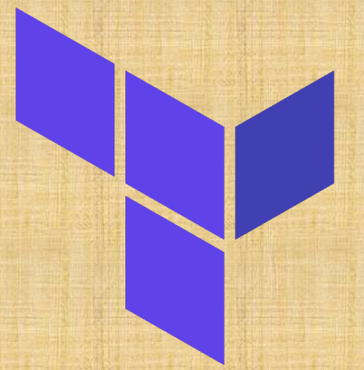
Skill: DevOps Engineer Lead

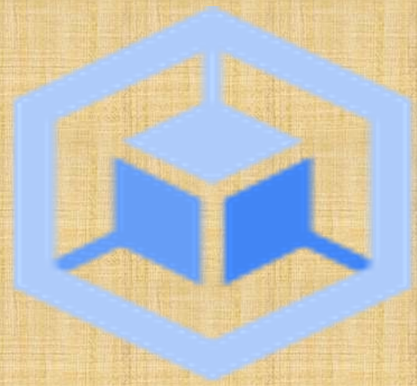
Demo-25

Google Kubernetes Engine

Terraform Custom Module

Kubernetes Deployment





Demo: Create Kubernetes Deployment by building a custom Terraform Module

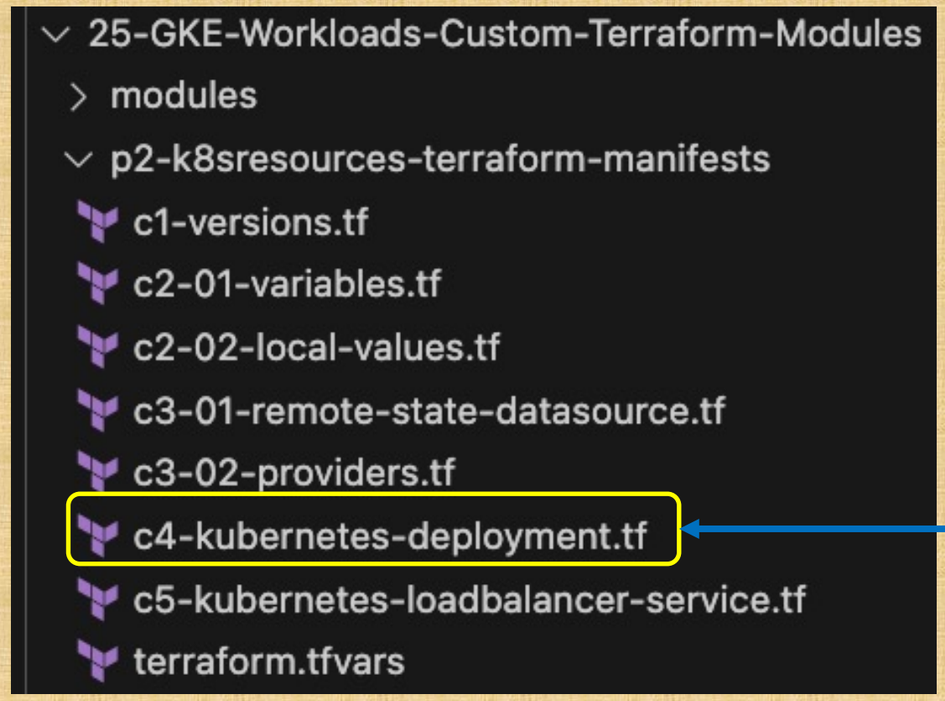
Author: Nho Luong

Skill: DevOps Engineer Lead

Build a Local Terraform Module

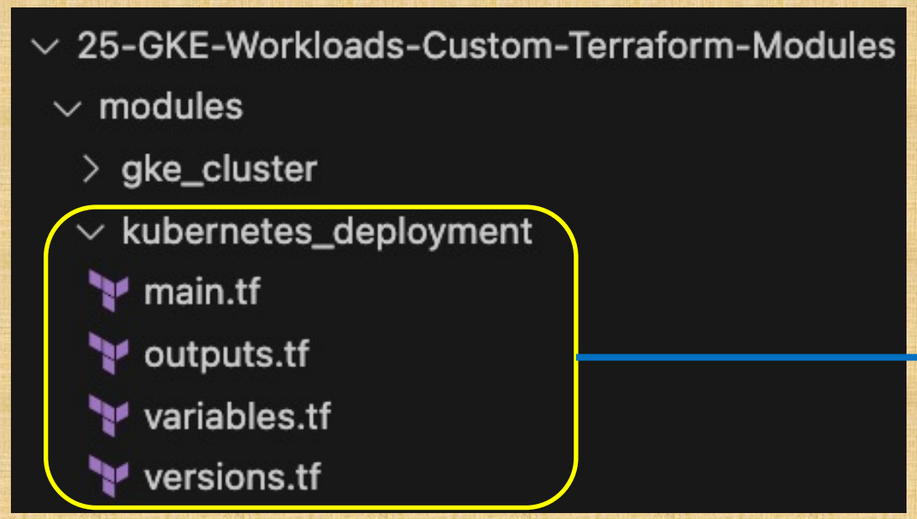
Modules Demo

Root Module



Call GKE Cluster child module in Root Module

Child Module



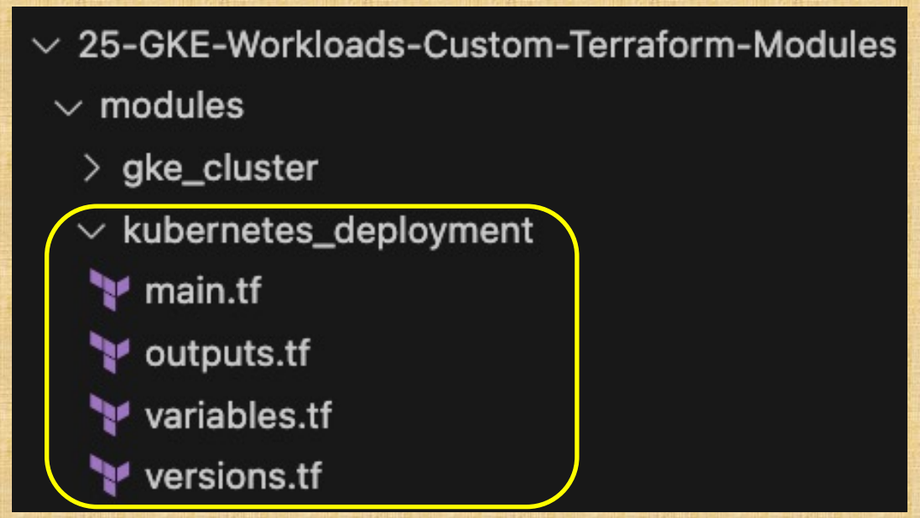
Build a Local Terraform Module and call it from Root Module and Test

Author: Nho Luong

Skill: DevOps Engineer Lead

Modules: Kubernetes Deployment Custom Module

Child Module



What are we going to learn?

Demo: Create GKE Cluster Custom Module

versions	Define Terraform Provider versions
variables	Define Terraform Input Variables
main	Create Kubernetes Deployment Terraform resource
outputs	Define Terraform Outputs

Author: Nho Luong

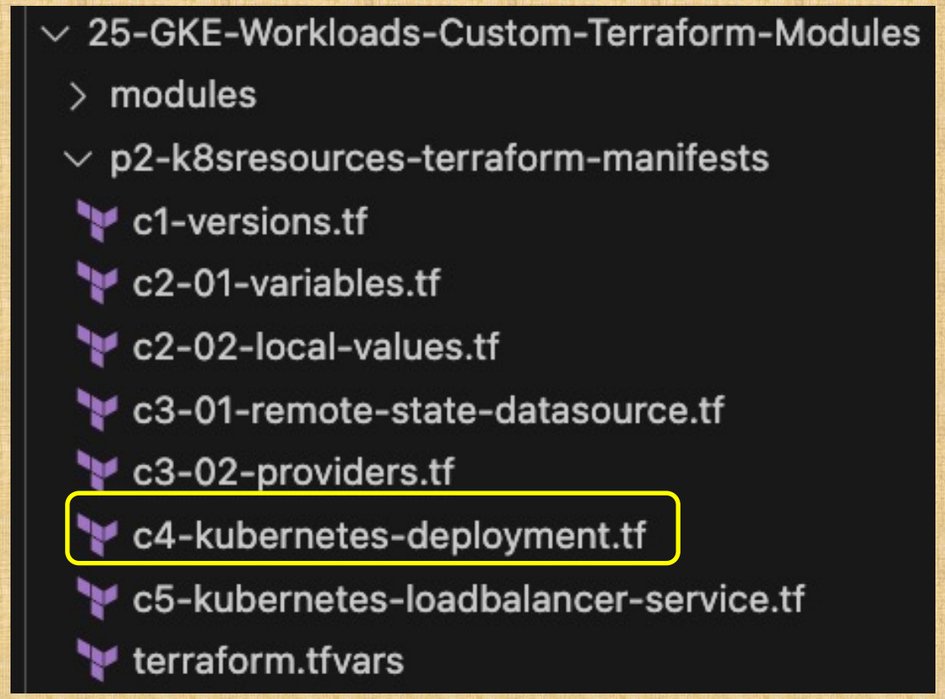
Skill: DevOps Engineer Lead

What are we going to learn?

Demo: Project-2: Deploy Kubernetes Deployment and Load Balancer Service by calling Terraform Module

Project-1: Terraform Manifests

Root Module

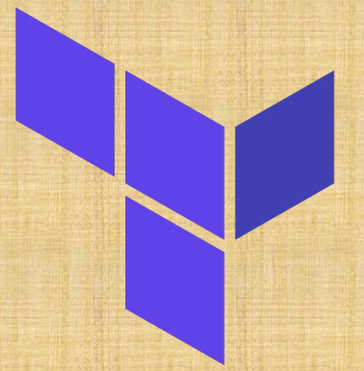


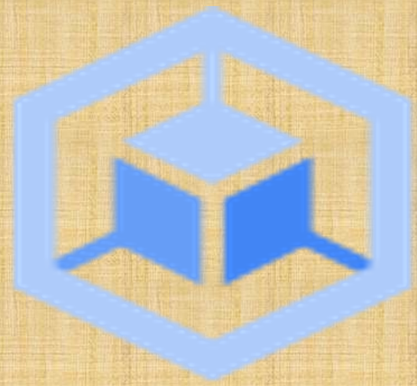
NC	No changes c1 to c3, c5
c4	Create Kubernetes Deployment by calling custom Kubernetes Deployment Terraform module

Author: Nho Luong

Skill: DevOps Engineer Lead

Demo-26

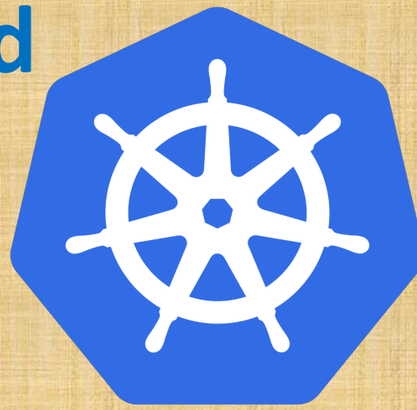




Google Kubernetes Engine

GKE Workload DevOps

GCP Cloud Build, GitHub GCP Cloud Build App and GitHub



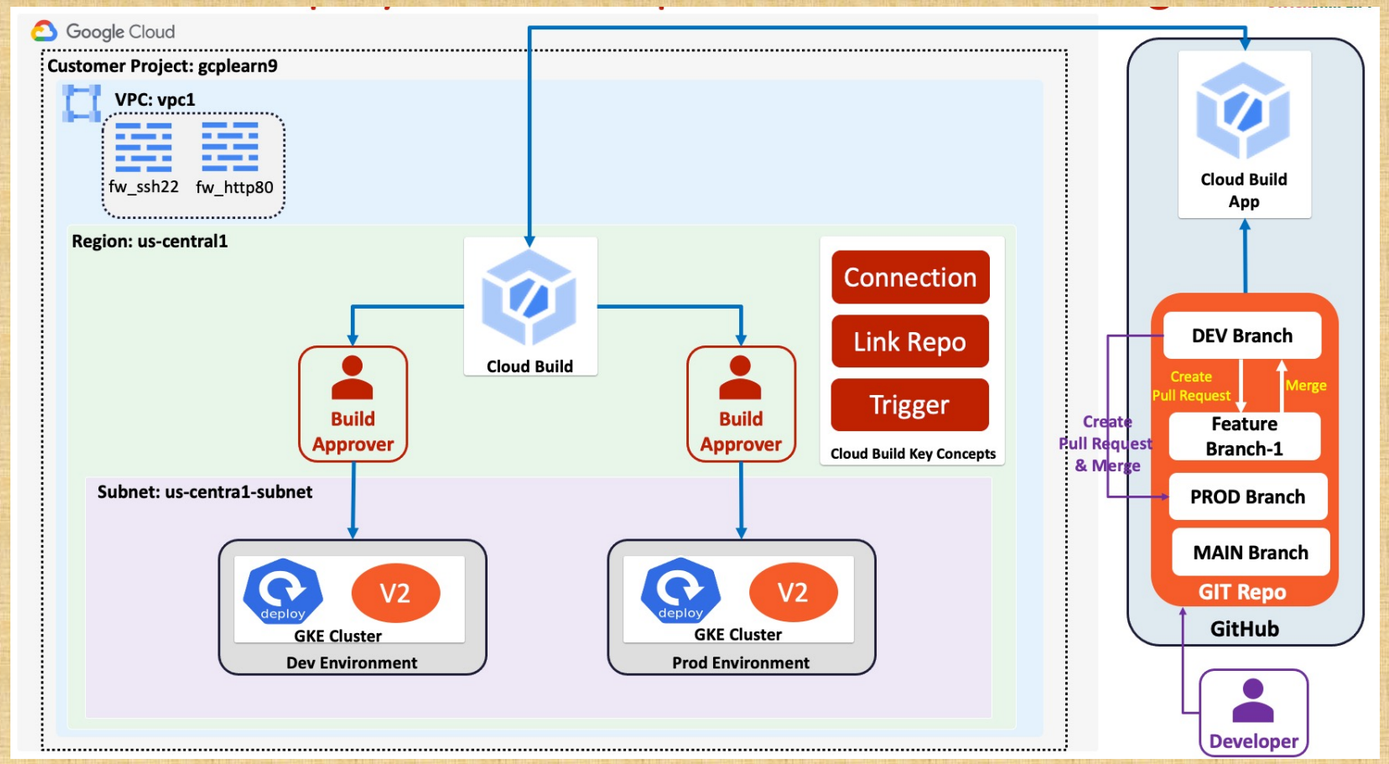
Demo: Implement DevOps Pipelines for **Kubernetes Deployment** Terraform Configs on GCP GKE

Author: Nho Luong

Skill: DevOps Engineer Lead

Kubernetes Deployment DevOps for Terraform Configs

- **The following resources are used for this implementation**
- **GitHub Repositories**
  - GitHub Branches
    - Main branch
    - Dev branch
    - Prod branch
    - Feature branch
  - GitHub Pull Request and Merge concepts
  - GCP Cloud Build App from GitHub Marketplace
- **GCP Cloud Build**
  - Connections
  - Link Repositories
  - Triggers
- **Kubernetes Resources**
  - Kubernetes Deployments
  - Kubernetes Load Balancer Services
- **Terraform Concepts**
  - Terraform Resources
  - Terraform custom modules
  - Terraform Remote Backend using Google Cloud Storage Bucket

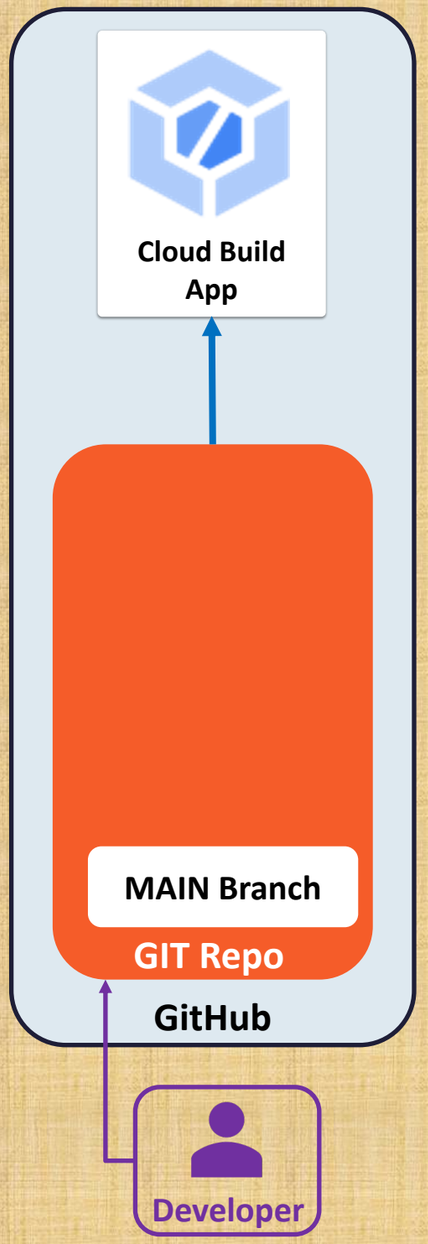


GIT Repo

Author: Nho Luong

Skill: DevOps Engineer Lead

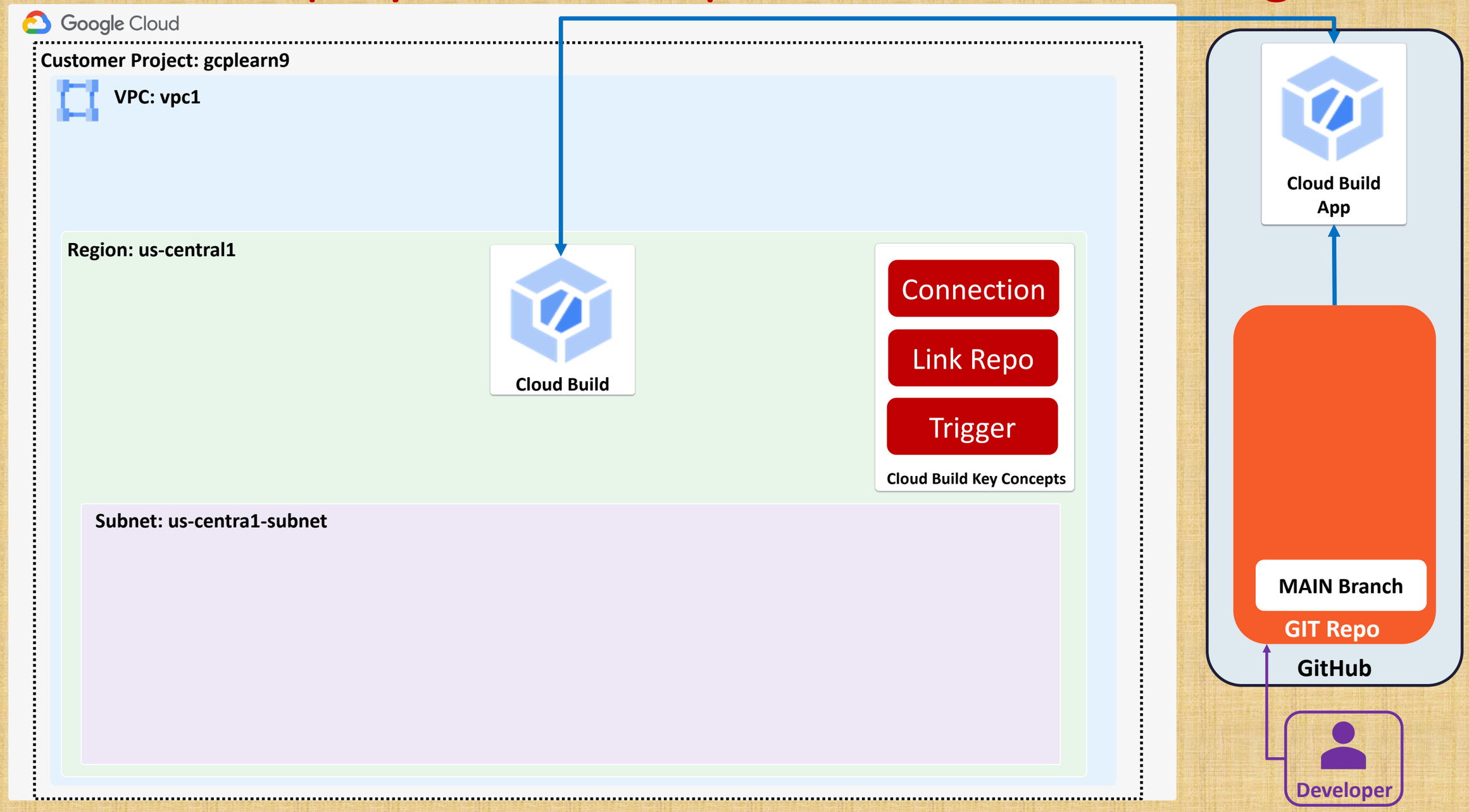
Kubernetes Deployment DevOps for Terraform Configs



Author: Nho Luong

Skill: DevOps Engineer Lead

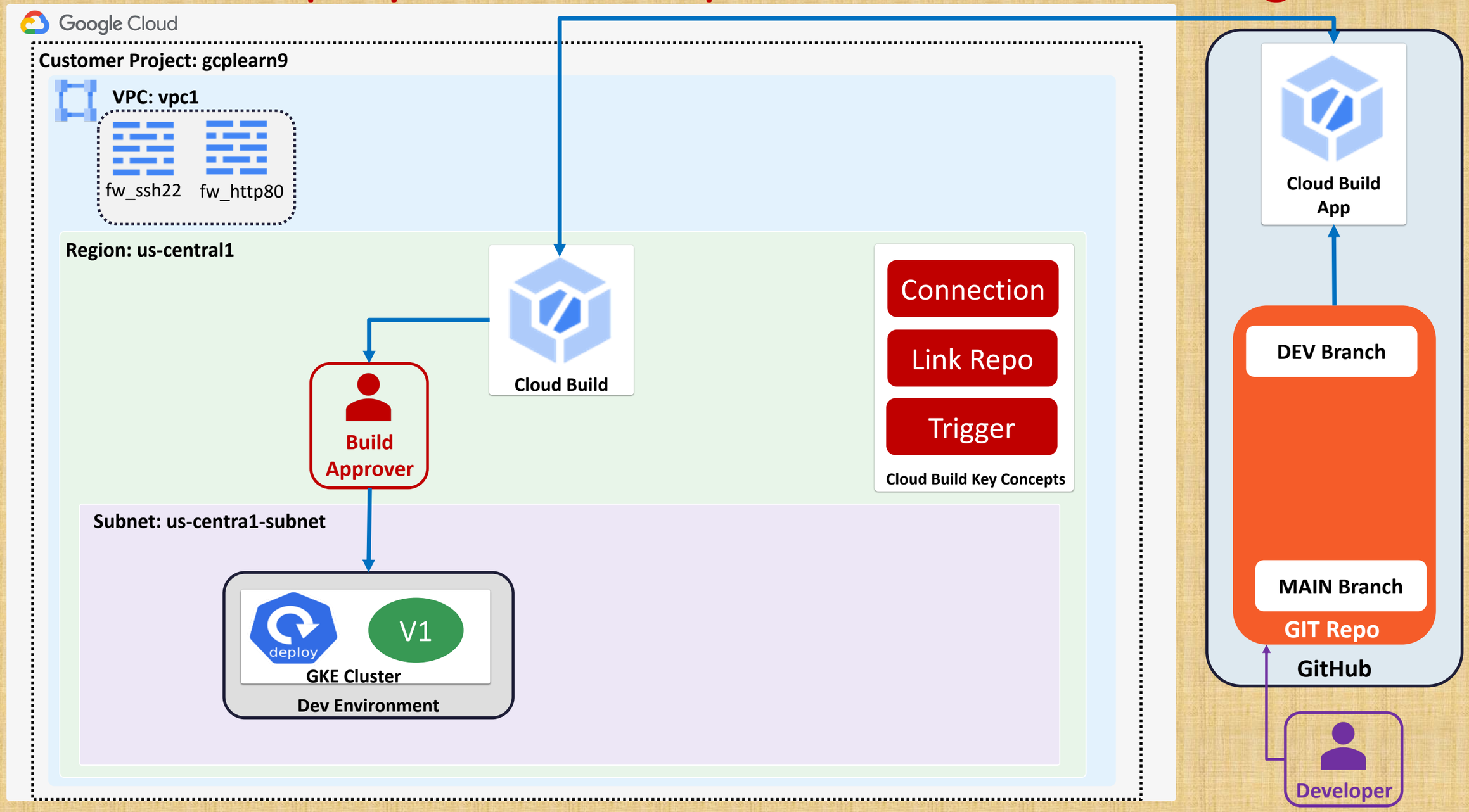
Kubernetes Deployment DevOps for Terraform Configs



Author: Nho Luong

Skill: DevOps Engineer Lead

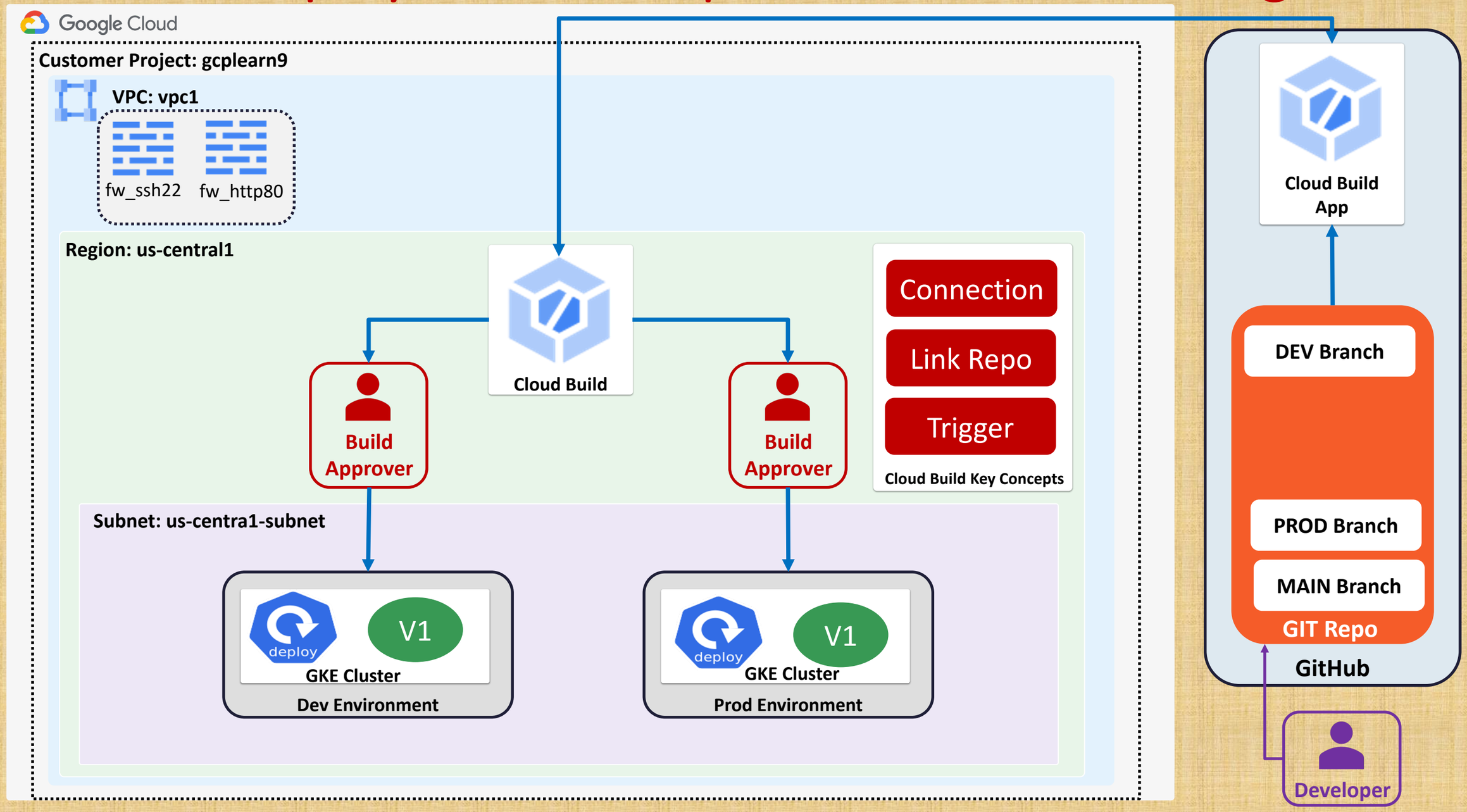
Kubernetes Deployment DevOps for Terraform Configs



Author: Nho Luong

Skill: DevOps Engineer Lead

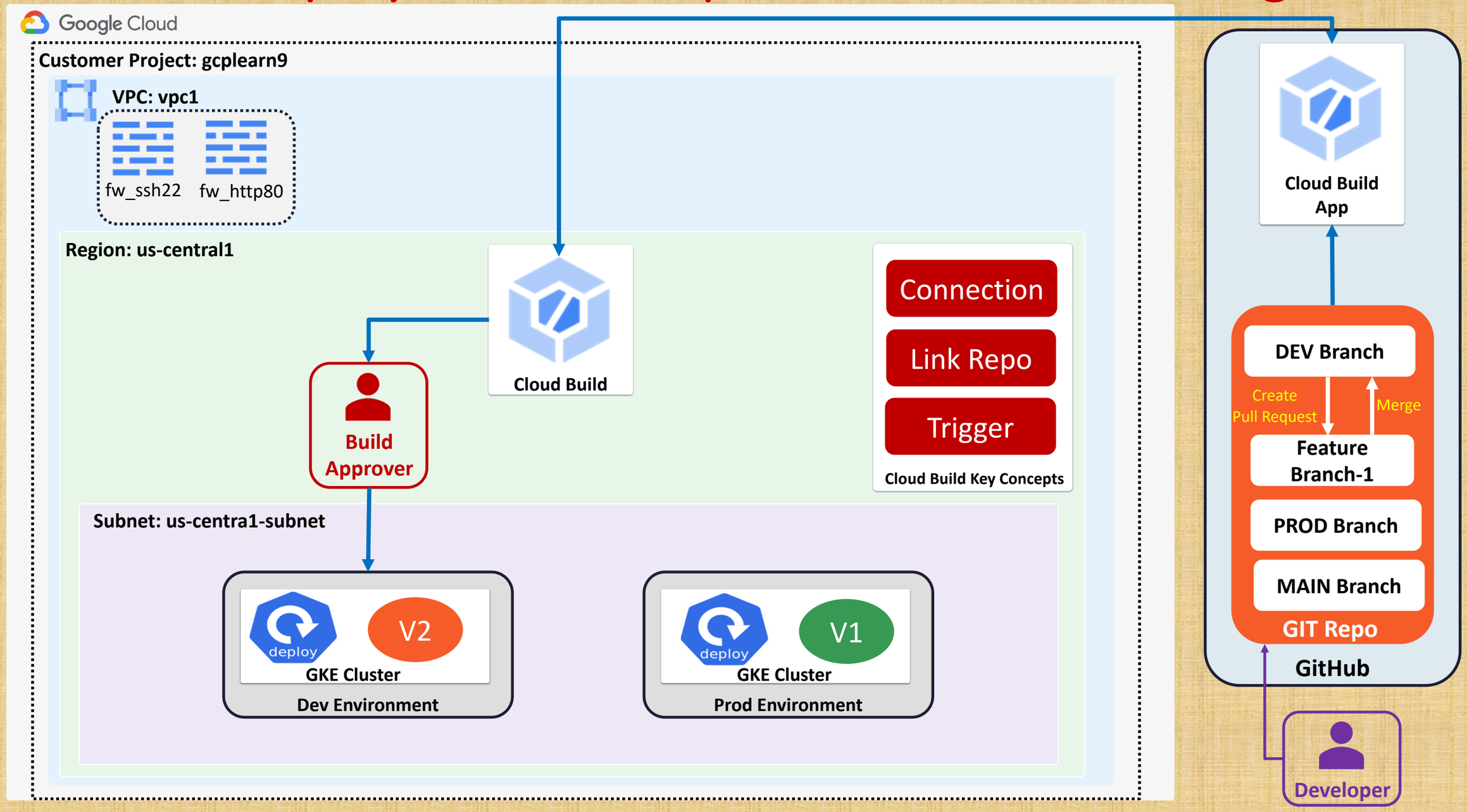
Kubernetes Deployment DevOps for Terraform Configs



Author: Nho Luong

Skill: DevOps Engineer Lead

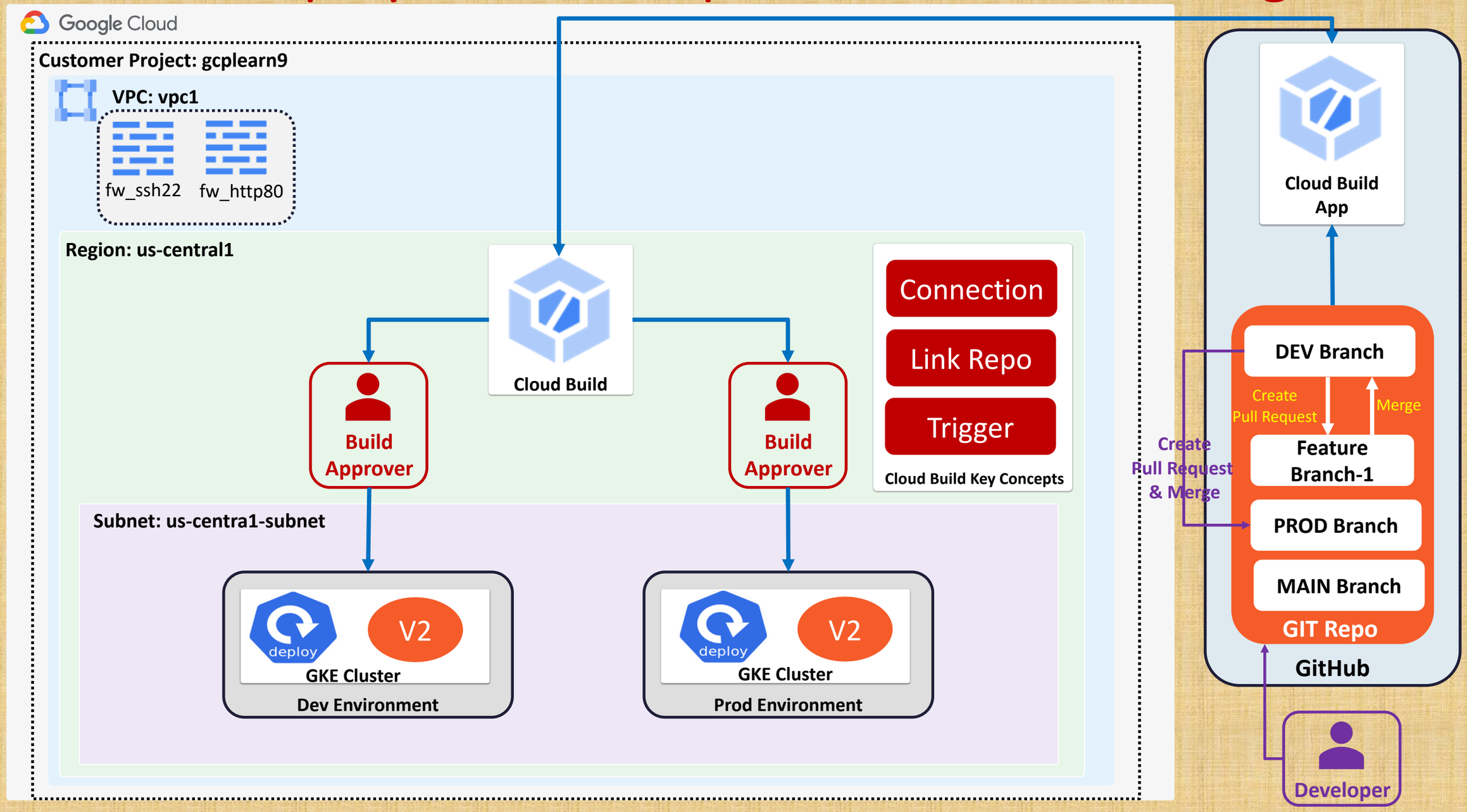
Kubernetes Deployment DevOps for Terraform Configs



Author: Nho Luong

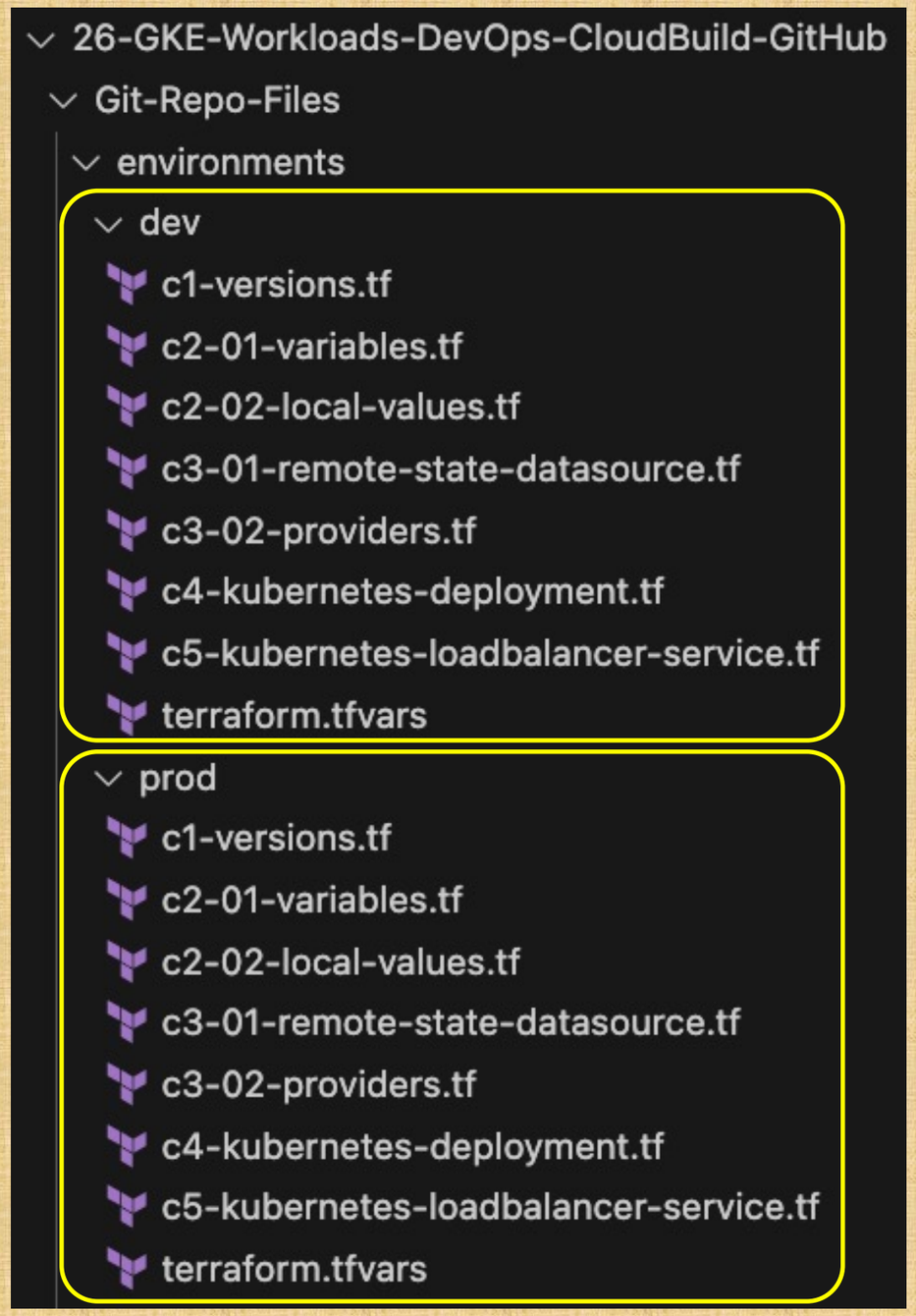
Skill: DevOps Engineer Lead

Kubernetes Deployment DevOps for Terraform Configs



Author: Nho Luong

Skill: DevOps Engineer Lead



What are we going to learn?

Demo: Implement DevOps Pipelines for Kubernetes Deployment Terraform Configs on GCP GKE

DEV	**Development Environment**
C1	Update GCS Bucket (your bucket name) and State file prefix as “workloads/dev/k8s-myapp1”
c3-01	Remote State Data source: Access GKE Dev Cluster Outputs “dev/gke-cluster”
TFVARS	Ensure (environment = "dev”) in terraform.tfvars file
PROD	**Production Environment**
C1	Update GCS Bucket (your bucket name) and State file prefix as “workloads/prod/k8s-myapp1”
c3-01	Remote State Data source: Access GKE Dev Cluster Outputs “prod/gke-cluster”
TFVARS	Ensure (environment = ”prod”) in terraform.tfvars file

Author: Nho Luong

Skill: DevOps Engineer Lead

What are we going to learn?

Demo: Implement DevOps Pipelines for Kubernetes Deployment Terraform Configs on GCP GKE

All these TF Configs are copy from Child Module Demo-25 (modules folder)

CHILD MODULE

Kubernetes Deployment Module we have created in Demo-25

cloud build

cloudbuild.yaml contains Terraform commands with DevOps pipeline code

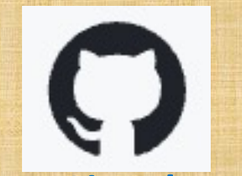
Author: Nho Luong

Skill: DevOps Engineer Lead

Demo-27

Google Kubernetes Engine

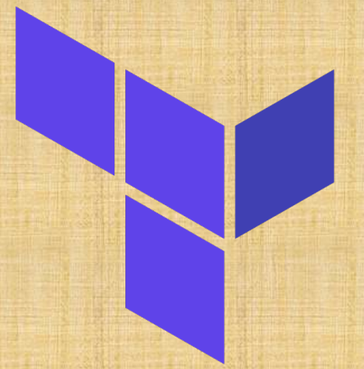
Continuous Integration

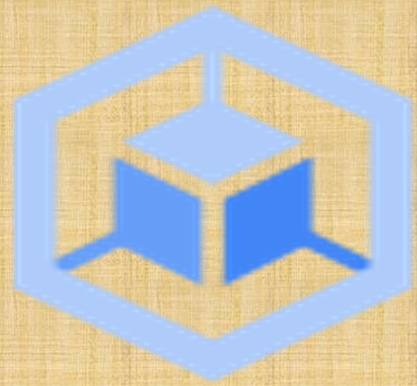


GitHub

Artifact Registry

Cloud Build





Demo: App on GKE Cluster - Continuous Integration

Author: Nho Luong

Skill: DevOps Engineer Lead

Continuous Integration

Continuous Integration Pipeline

GitHub

Google Cloud Build

Google Artifact Registry

To Store App Code (Application Code + Docker Files)

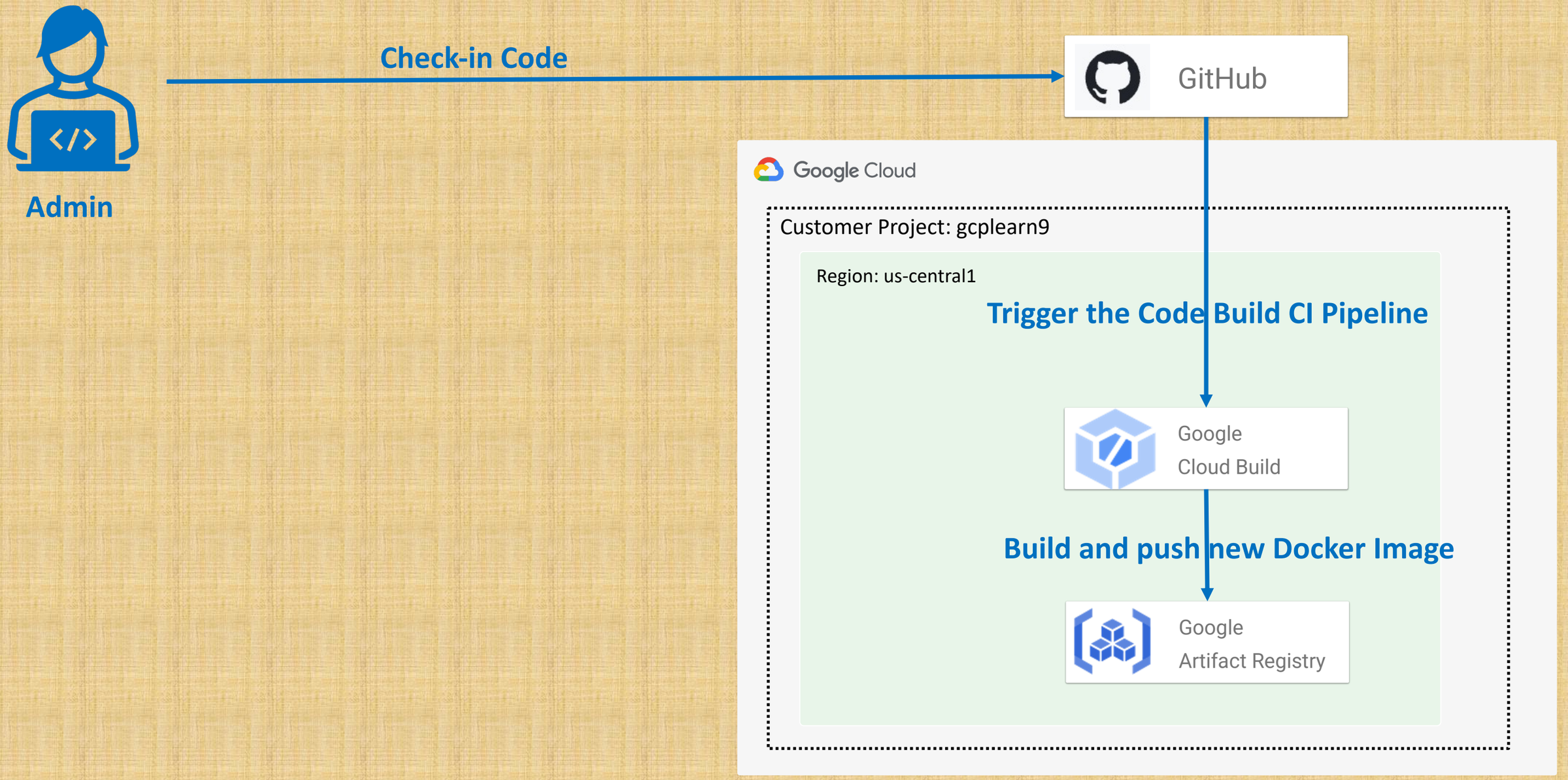
To Build Packages (Build Docker Image)

To Store Packages (Store Docker Image)

Author: Nho Luong

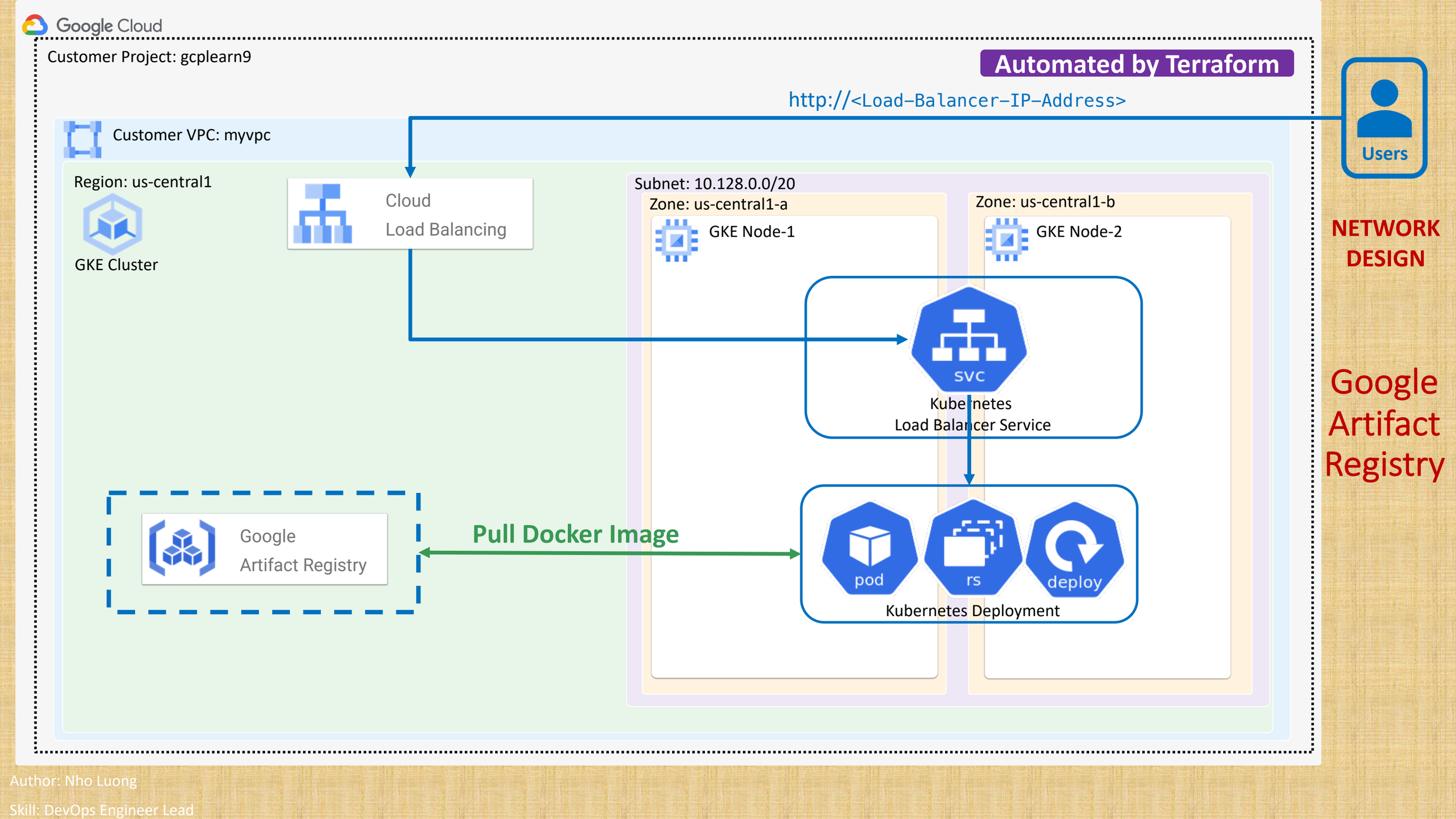
Skill: DevOps Engineer Lead

Google Cloud Build: Continuous Integration



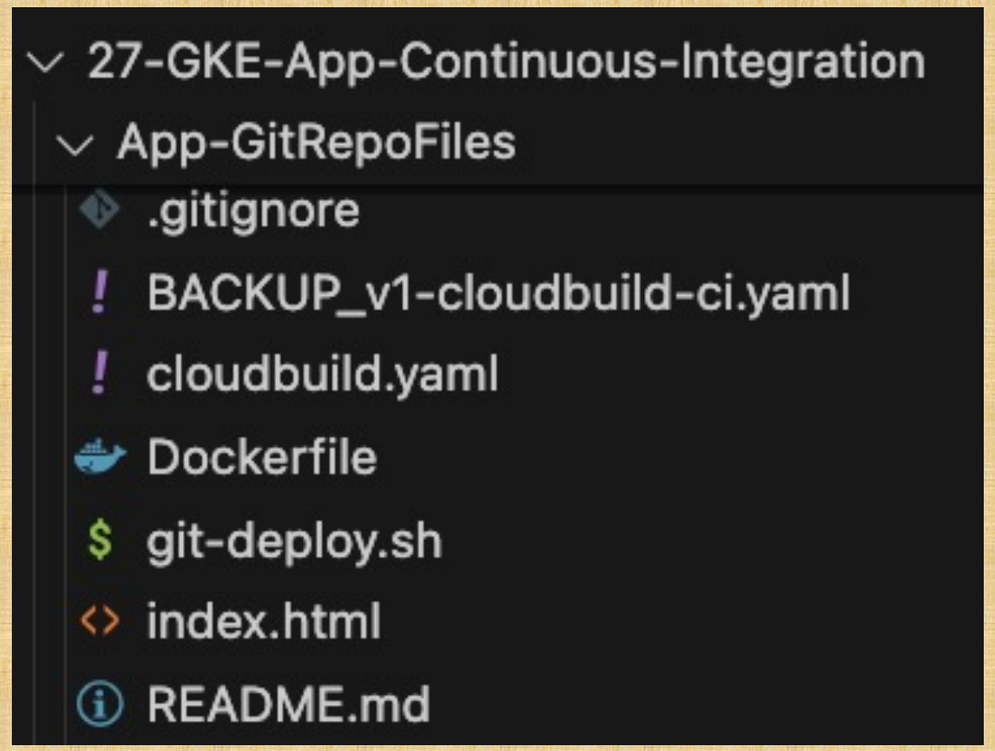
Author: Nho Luong

Skill: DevOps Engineer Lead



Author: Nho Luong
Skill: DevOps Engineer Lead

Git Repo Files



What are we going to learn?

Demo: App on GKE Cluster - Continuous Integration

index.html — Sample Application - index.html page

Dockerfile — Dockerfile to create Docker Image for our sample application

cloudbuild — cloudbuild.yaml used for implementing CI Pipeline in GCP Cloud Build

Author: Nho Luong

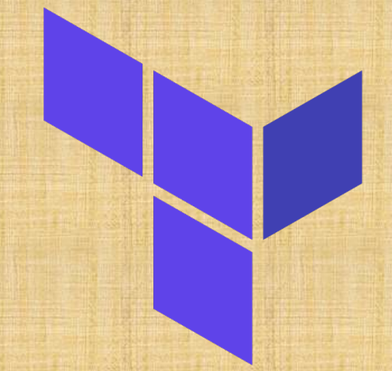
Skill: DevOps Engineer Lead

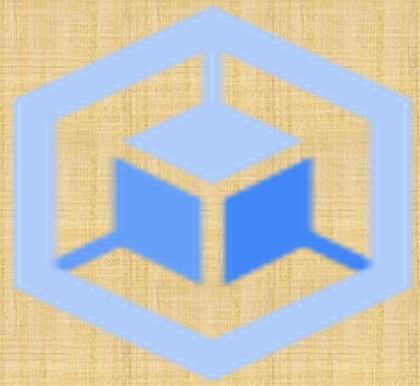
Demo-28

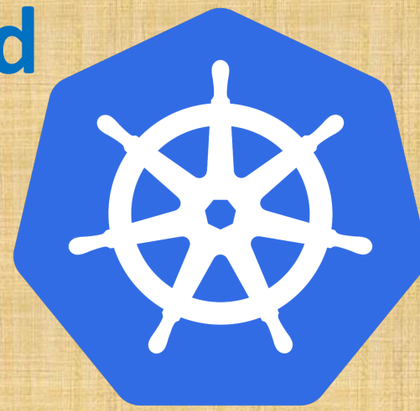
Google Kubernetes Engine

Continuous Integration & Delivery

GCP Cloud Build, GitHub GCP Cloud Build App and GitHub





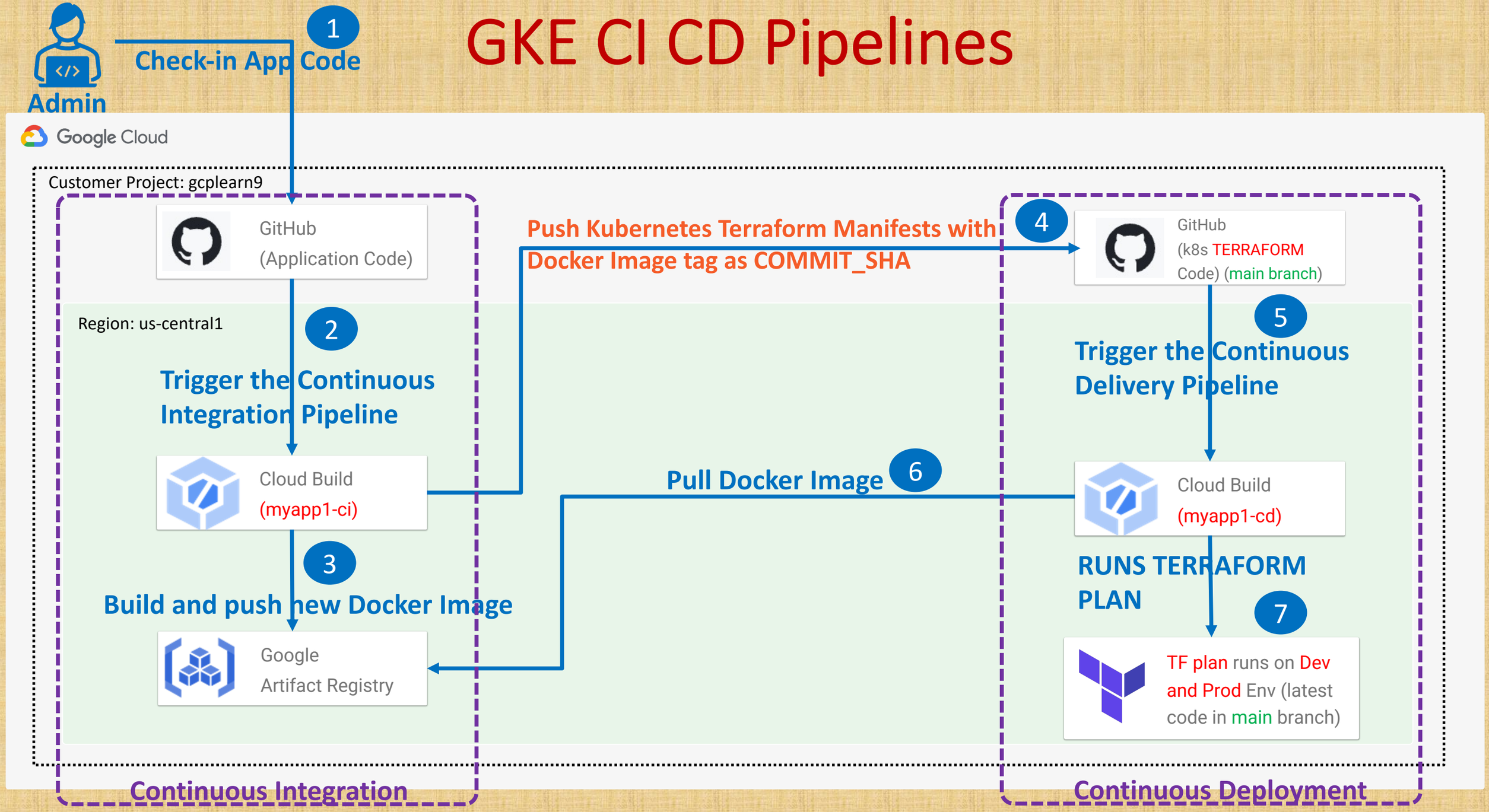


Demo: Implement Continuous Integration & Continuous Delivery for GKE Workloads (MyApp1)

Author: Nho Luong

Skill: DevOps Engineer Lead

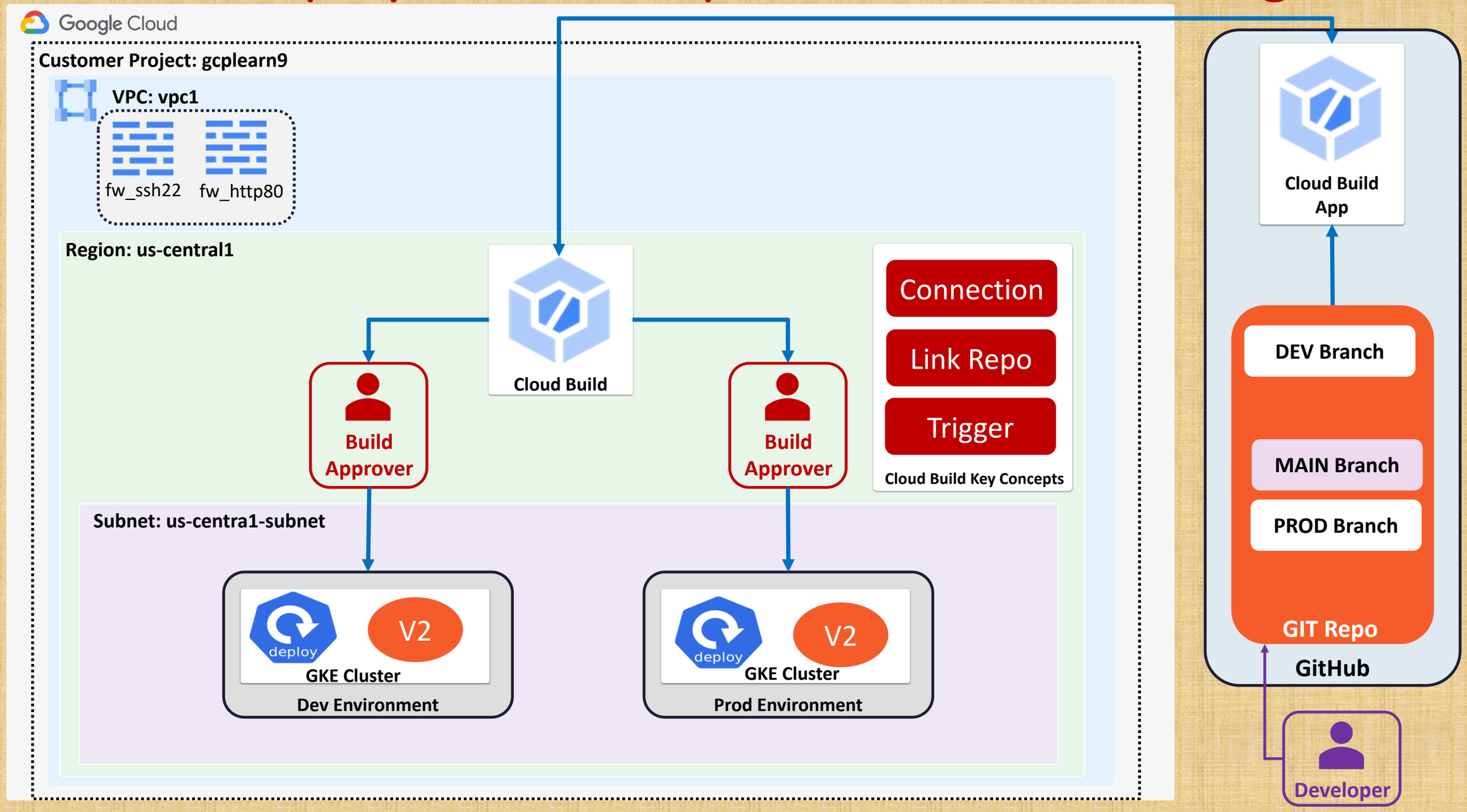
GKE CI CD Pipelines



Author: Nho Luong

Skill: DevOps Engineer Lead

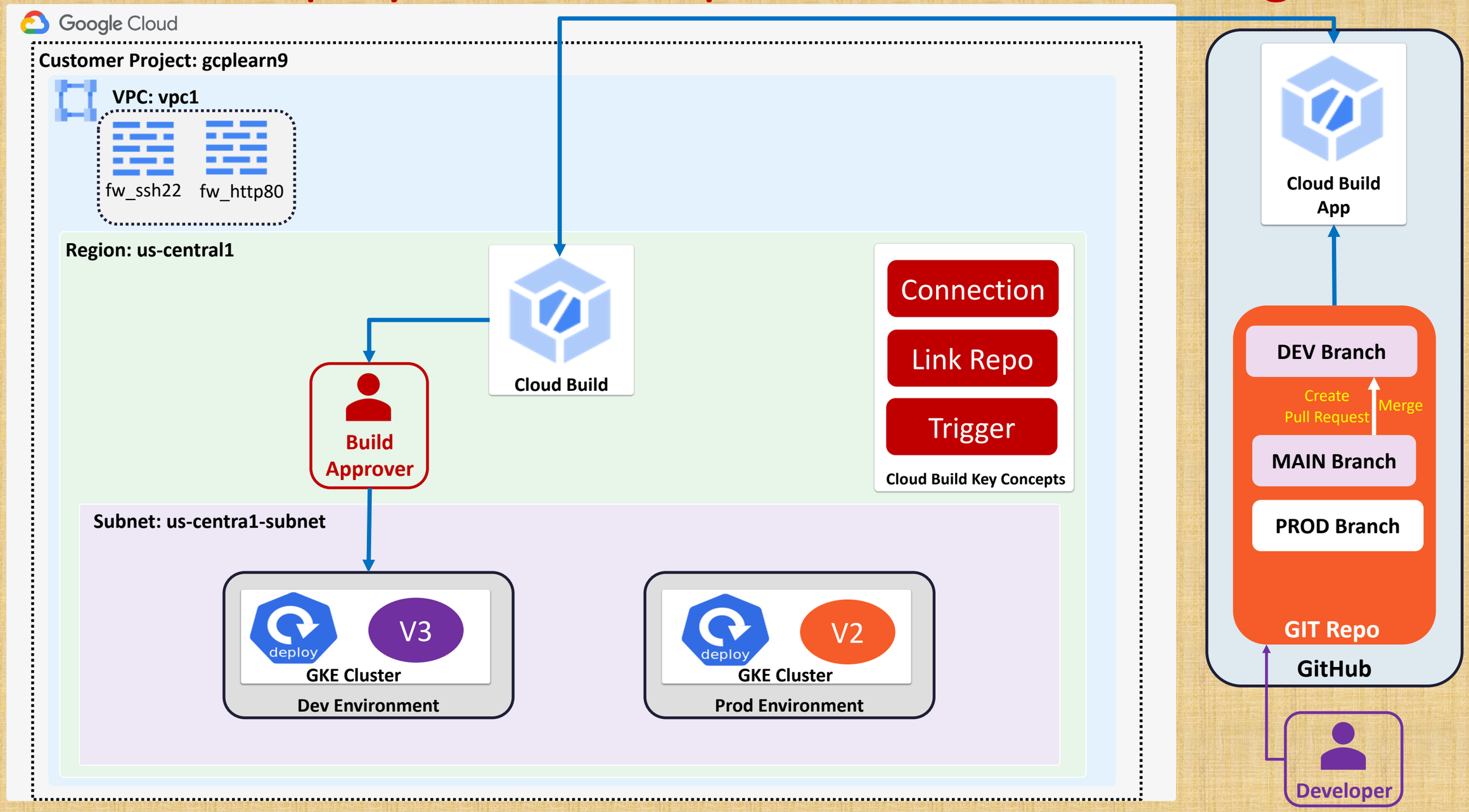
Kubernetes Deployment DevOps for Terraform Configs



Author: Nho Luong

Skill: DevOps Engineer Lead

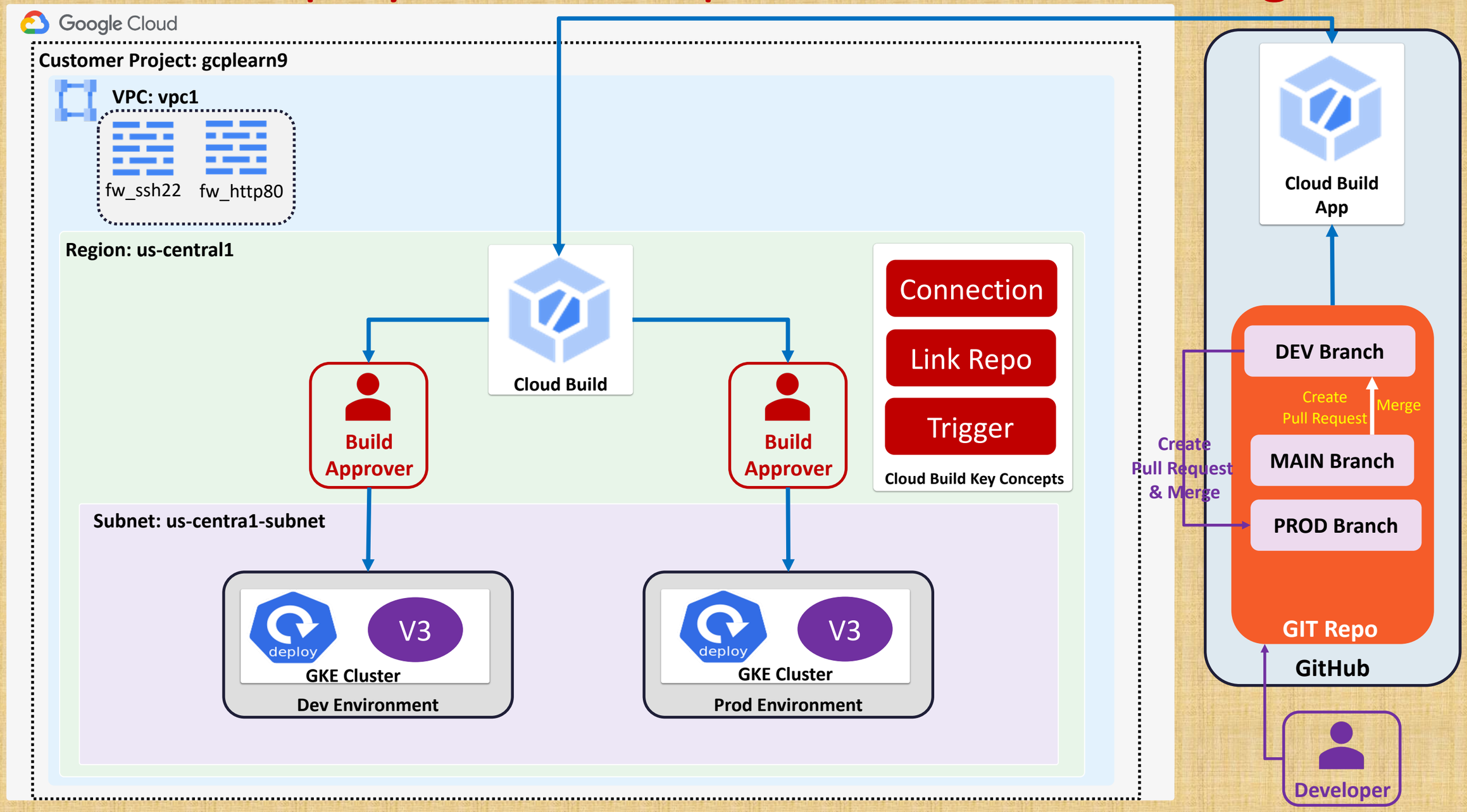
Kubernetes Deployment DevOps for Terraform Configs



Author: Nho Luong

Skill: DevOps Engineer Lead

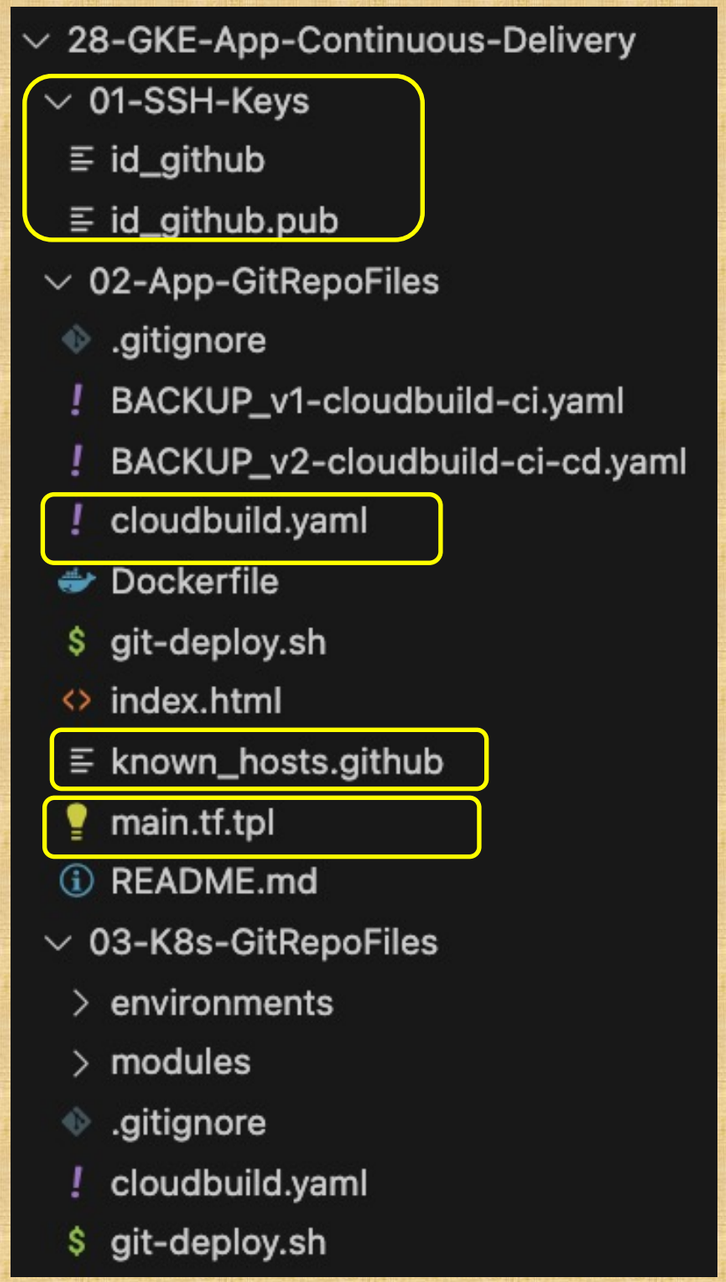
Kubernetes Deployment DevOps for Terraform Configs



Author: Nho Luong

Skill: DevOps Engineer Lead

Git Repo Files: App, k8s



What are we going to learn?

Demo: Implement Continuous Integration & Continuous Delivery for GKE Workloads (MyApp1)

App	App: GitRepoFiles
SSH Keys	Configure SSH Authentication from Cloud Build (From build run container) to K8S Repository
cloud build	Add cloudbuild.yaml with Continuous Delivery Code
Known hosts	Get the github.com public certificate to store in known_hosts file where build is run (In build run container)
main. tf.tpl	K8S Deployment template file used for updating latest Docker Image Tag
K8S	K8S: GitRepoFiles
NC	NO CHANGES in k8s repo from Demo-26 onwards

Author: Nho Luong

Skill: DevOps Engineer Lead

ANYTHING AFTER THIS SLIDE IS NON-LIVE SLIDES

Author: Nho Luong

Skill: DevOps Engineer Lead

Thank You

Author: Nho Luong

Skill: DevOps Engineer Lead