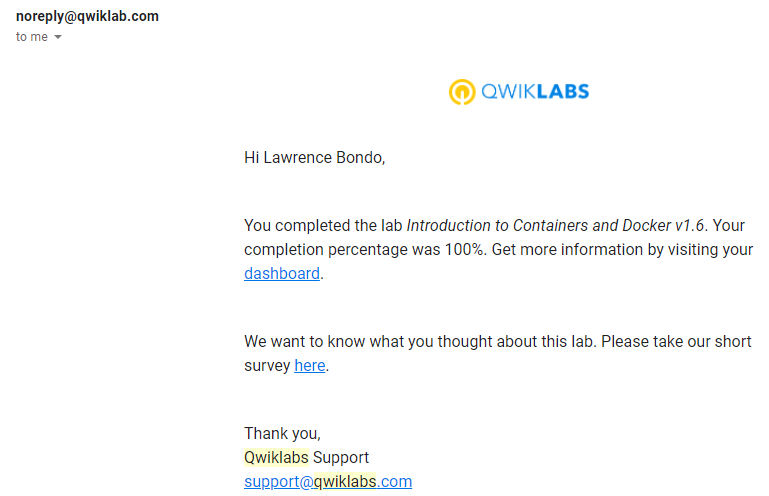
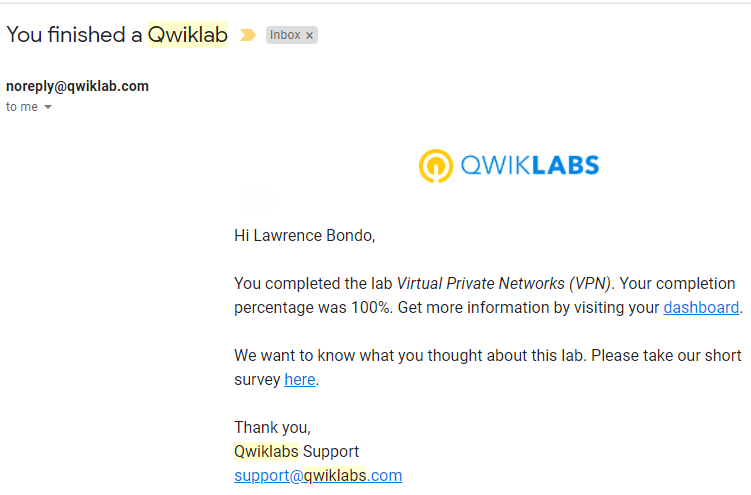
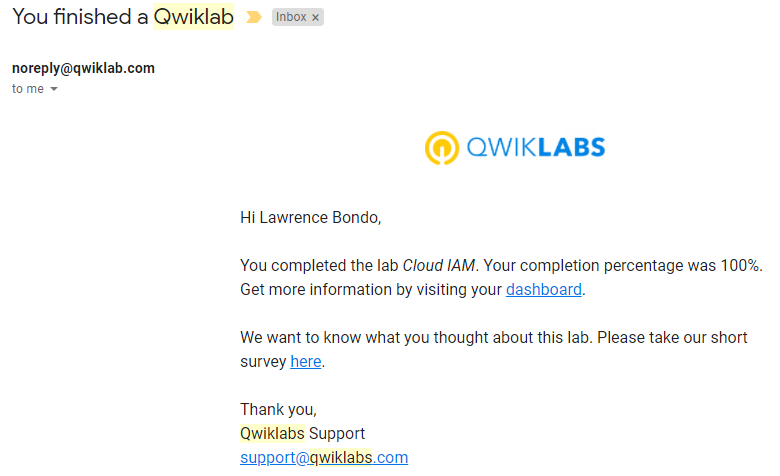
1. **LAB: Introduction to Containers and Docker v1.6:**
2. **LAB: Virtual Private Networks (VPN):**



1. **LAB: Configuring an Internal Load Balancer:**



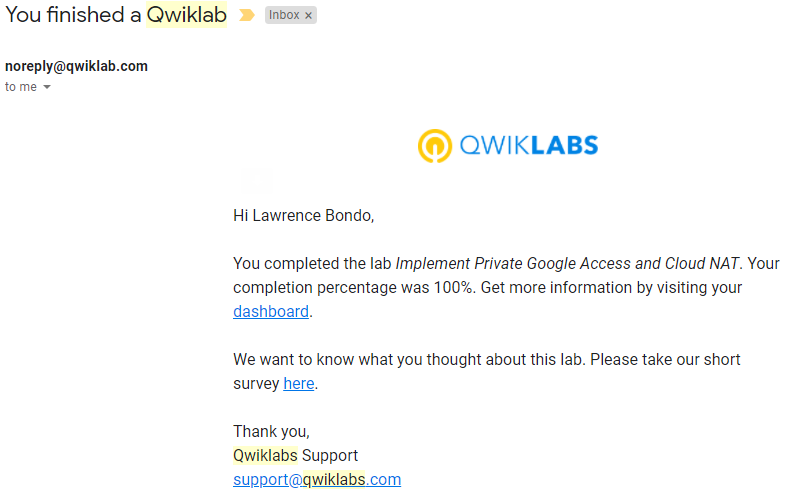
1. **LAB: Cloud IAM:**



1. **LAB: Creating Virtual Machines:**



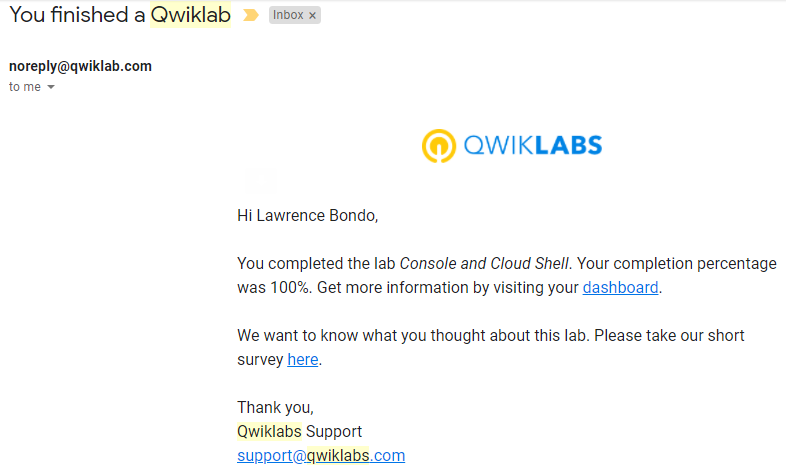
1. **LAB: Implement Private Google Access and Cloud NAT:**



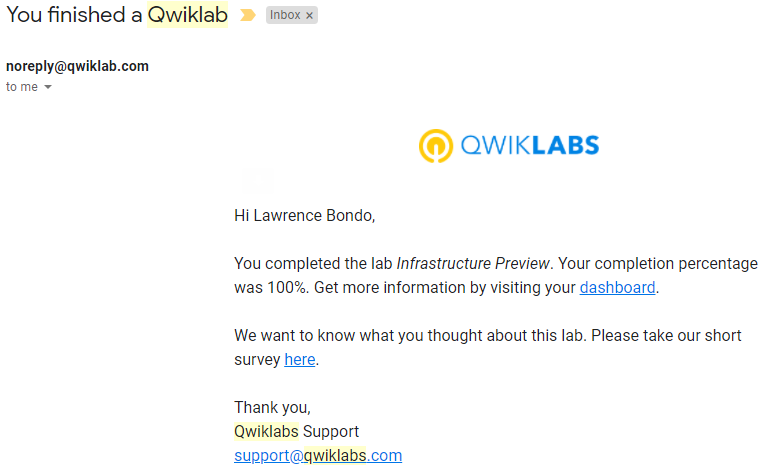
1. **LAB: VPC Networking:**



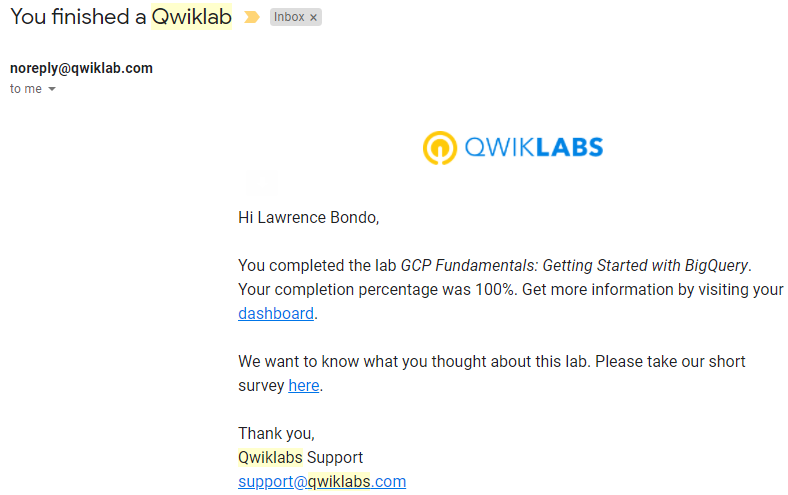
1. **LAB: Console and Cloud Shell:**



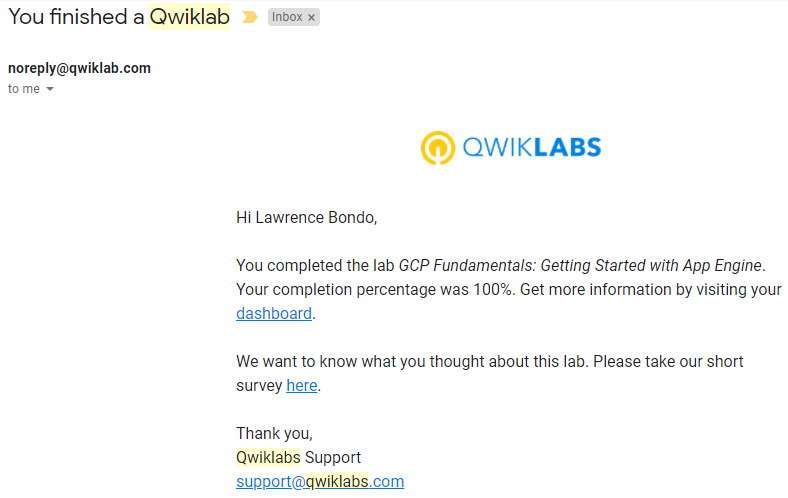
1. **LAB: Infrastructure Preview:**



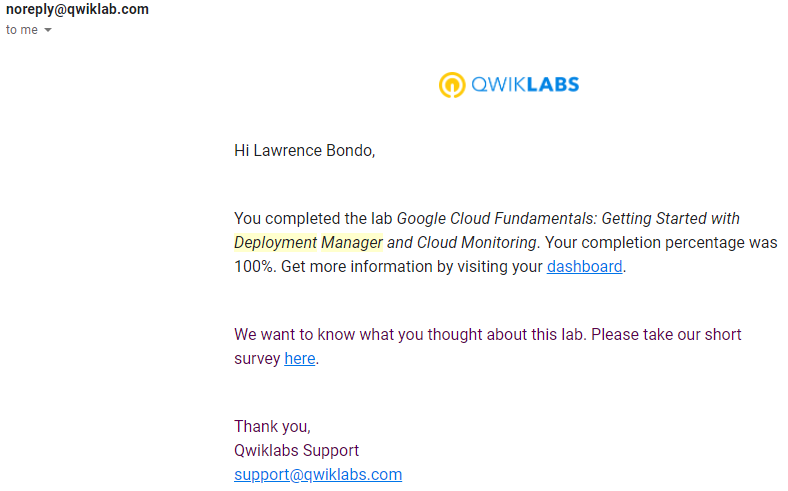
1. **GCP Fundamentals-Getting Started with BigQuery:**



1. **LAB: GCP Fundamentals: Getting Started with App Engine:**



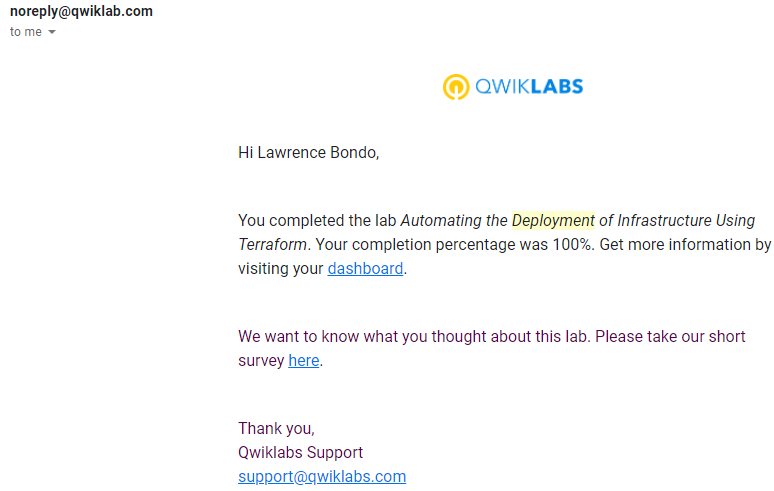
1. **LAB: Google Cloud Fundamentals: Getting Started with Deployment Manager:**



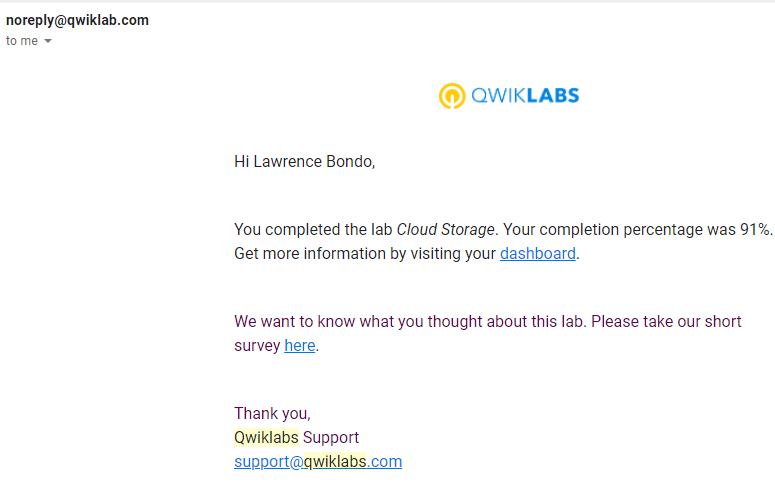
1. **LAB: Automating the Deployment of Infrastructure Using Deployment Manager:**



1. **LAB: Automating the Deployment of Infrastructure Using Terraform:**



1. **LAB: Cloud Storage:**



**Cloud Challenge 2:**

**LABS TRANSLATION:**

1. **Automating the Deployment of Infrastructure Using Terraform:**

**main.tf file:-**

variable "instance\_name" {}

variable "instance\_zone" {}

variable "instance\_type" {

default = "n1-standard-1"

}

variable "instance\_subnetwork" {}

resource "google\_compute\_instance" "vm\_instance" {

name = "${var.instance\_name}"

zone = "${var.instance\_zone}"

machine\_type = "${var.instance\_type}"

boot\_disk {

initialize\_params {

image = "debian-cloud/debian-9"

}

}

network\_interface {

subnetwork = "${var.instance\_subnetwork}"

access\_config {

# Allocate a one-to-one NAT IP to the instance

# Create the mynet-us-vm instance

module "mynet-us-vm" {

source = "./instance"

instance\_name = "mynet-us-vm"

instance\_zone = "us-central1-a"

instance\_subnetwork = google\_compute\_network.mynetwork.self\_link

}

# Create the mynet-eu-vm" instance

module "mynet-eu-vm" {

source = "./instance"

instance\_name = "mynet-eu-vm"

instance\_zone = "europe-west1-d"

instance\_subnetwork = google\_compute\_network.mynetwork.self\_link

}

}

}

}

**mynetwork.tf file:-**

# Create the mynetwork network

resource "google\_compute\_network" "mynetwork" {

name = "mynetwork"

auto\_create\_subnetworks = "true"

#RESOURCE properties go here

}

# Add a firewall rule to allow HTTP, SSH, RDP and ICMP traffic on mynetwork

resource "google\_compute\_firewall" "mynetwork-allow-http-ssh-rdp-icmp" {

name = "mynetwork-allow-http-ssh-rdp-icmp"

network = google\_compute\_network.mynetwork.self\_link

allow {

protocol = "tcp"

ports = ["22", "80", "3389"]

}

allow {

protocol = "icmp"

}

}

**provider.tf file:-**

provider "google" {}

1. **Automating the Deployment of Infrastructure Using Deployment Manager:**

Below code is a content of jinja template and config.yaml files.

imports:

- path: instance-template.jinja (will be used for creating two similar VM instances)

resources:

# Create the auto-mode network

- name: mynetwork

type: compute.v1.network

properties:

autoCreateSubnetworks: true

# Create the firewall rule

- name: mynetwork-allow-http-ssh-rdp-icmp

type: compute.v1.firewall

properties:

network: $(ref.mynetwork.selfLink)

sourceRanges: ["0.0.0.0/0"]

allowed:

- IPProtocol: TCP

ports: [22, 80, 3389]

- IPProtocol: ICMP

# Create the mynet-us-vm instance out of instance-template.jinja

- name: mynet-us-vm

type: instance-template.jinja

properties:

zone: us-central1-a

machineType: n1-standard-1

network: $(ref.mynetwork.selfLink)

subnetwork: regions/us-central1/subnetworks/mynetwork

# Create the mynet-eu-vm instance out of instance-template.jinja

- name: mynet-eu-vm

type: instance-template.jinja

properties:

zone: europe-west1-d

machineType: n1-standard-1

network: $(ref.mynetwork.selfLink)

subnetwork: regions/europe-west1/subnetworks/mynetwork

# Finally deploying configuration and verifying deployment by running above code.