

DevOps and Cloud

Real Scenario Based Project

<https://topmate.io/cloud>

<https://www.youtube.com/@DevOps-Cloud>

Why are we doing this Project?

1

We acknowledge DevOps and Cloud is a Vast and a Specialized skill

Zero to Hero Courses are NOT sufficient.

Build differentiating skills.

2

Industry values the Hand-on experience.

Interviews focus on the Best Practices followed in the industry.

3

You are working in legacy area but want to show some experience as DevOps and Cloud

No one likes Monday and Freshers. Don't stay Fresher.

Build your GitHub Code Repo with Quality Projects

Who this Project is suitable for?:

1

**Freshers and Entry Level
DevOps and Cloud
engineers**

**You should have some
basic knowledge and have
already built simple
Projects**

2

**Experienced IT
Professionals switching to
DevOps and Cloud space**

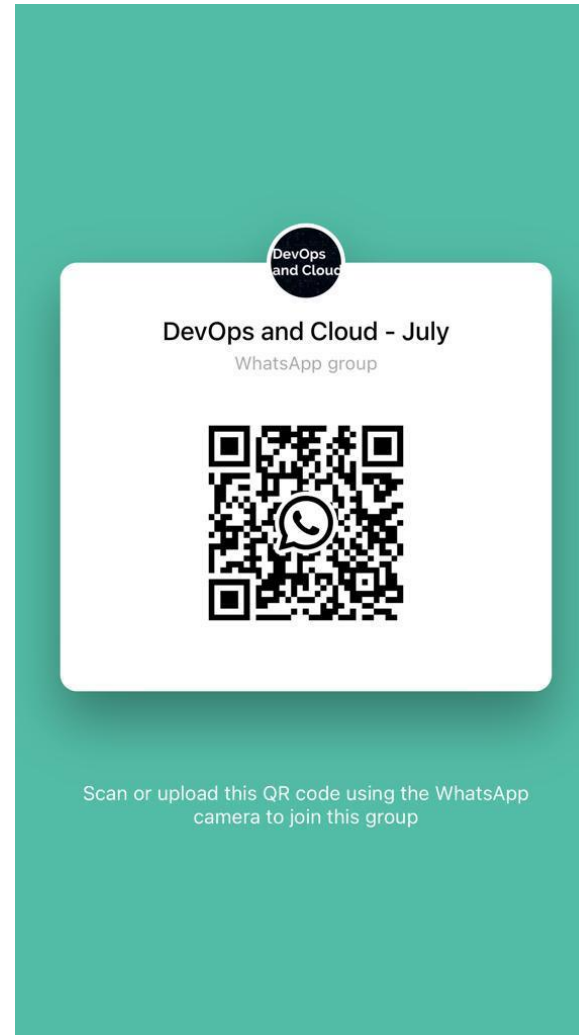
**You should have some
basic knowledge and have
already built very simple
hands-on experience.**

3

**Experienced IT
Professionals working in
DevOps and Cloud**

**You should have good
knowledge and have
already built simpler
hands-on projects.**

How do we collaborate for next 4-5 weeks?



- ✓ Will share the Recording of Today's Session on WhatsApp Group
- ✓ Will open this WhatsApp Group for collaboration next week for discussion. Please be mindful of the language and stuff you are posting and the number of times you are posting.
- ✓ Next Sessions will be on 8th, 17th, 22nd and 29th July, Same Time on Microsoft Teams.
- ✓ Will share the joining Links on this WhatsApp Group.

Infrastructure Deployment using IAC and CI/CD Pipeline Implementation

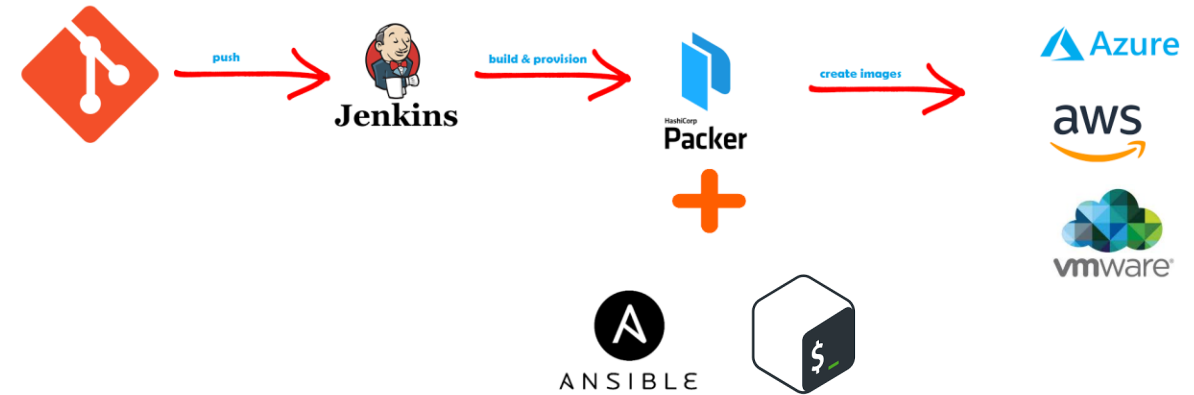
The objective of this project is to build a scalable and secure infrastructure on AWS/Azure using Infrastructure as Code practices and to establish a DevOps pipeline for building the application and deployment to the infrastructure.

Part 1

Creation of a Base AMI using Packer

Estimated Time 3 Days

- You are free to use AWS or Azure for this entire project
- Packer is a differentiating skill. Easy to get started and custom base AMIs are used across the industry and in every company.
- Entry level fresher engineers can use shell script as provisioner.
- If you are expert, then you should try the Ansible based provisioner.
- Please remember all three Parts of the project are for a purpose and if Part 2 or 3 require any additional component then this pipeline will help in that.
- So, you will build a Packer based pipeline in your choice of CI tool by taking an AWS/Azure provided base image and install your required software using Shell Script or Ansible provisioner.

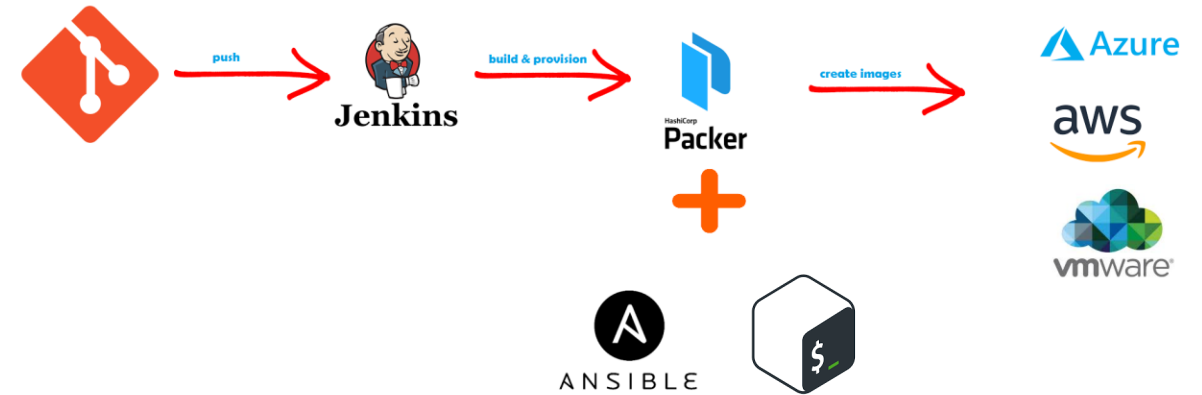


Reference architecture for Golden Base Image using Packer.

<https://github.com/devops-cloud-academy/packer>

Software's required on AMI

- CloudWatch Agent
- AWS SSM Agent
- Docker
- More to be added as we go...



Reference architecture for Golden Base Image using Packer.

<https://github.com/devops-cloud-academy/packer>

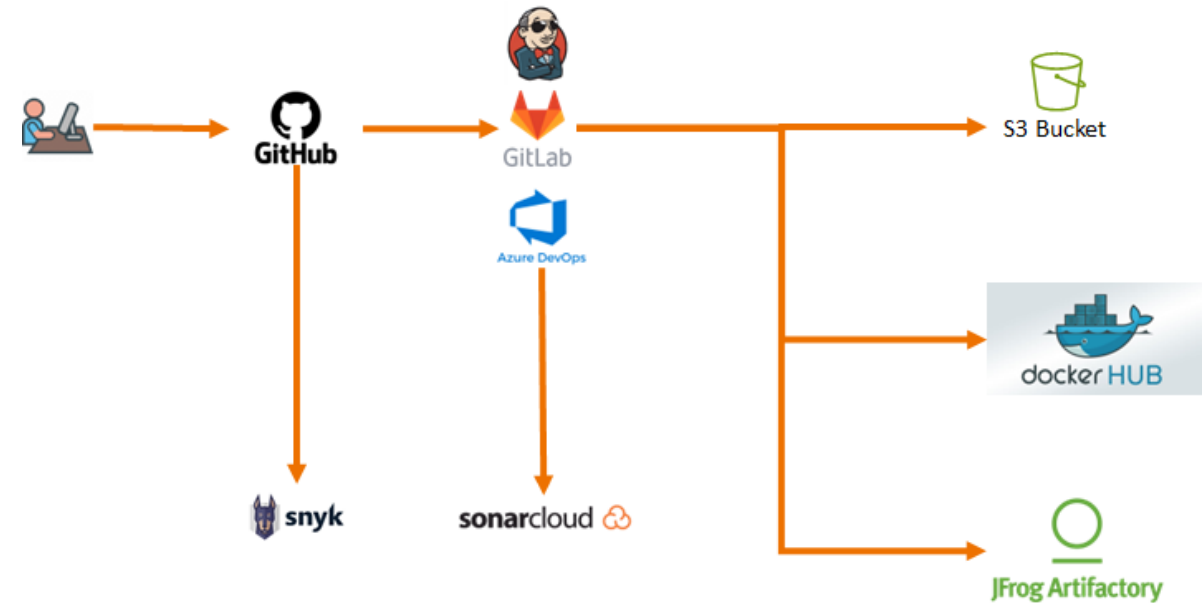
Building of Application using Maven/Docker

Estimated Time 5 Days

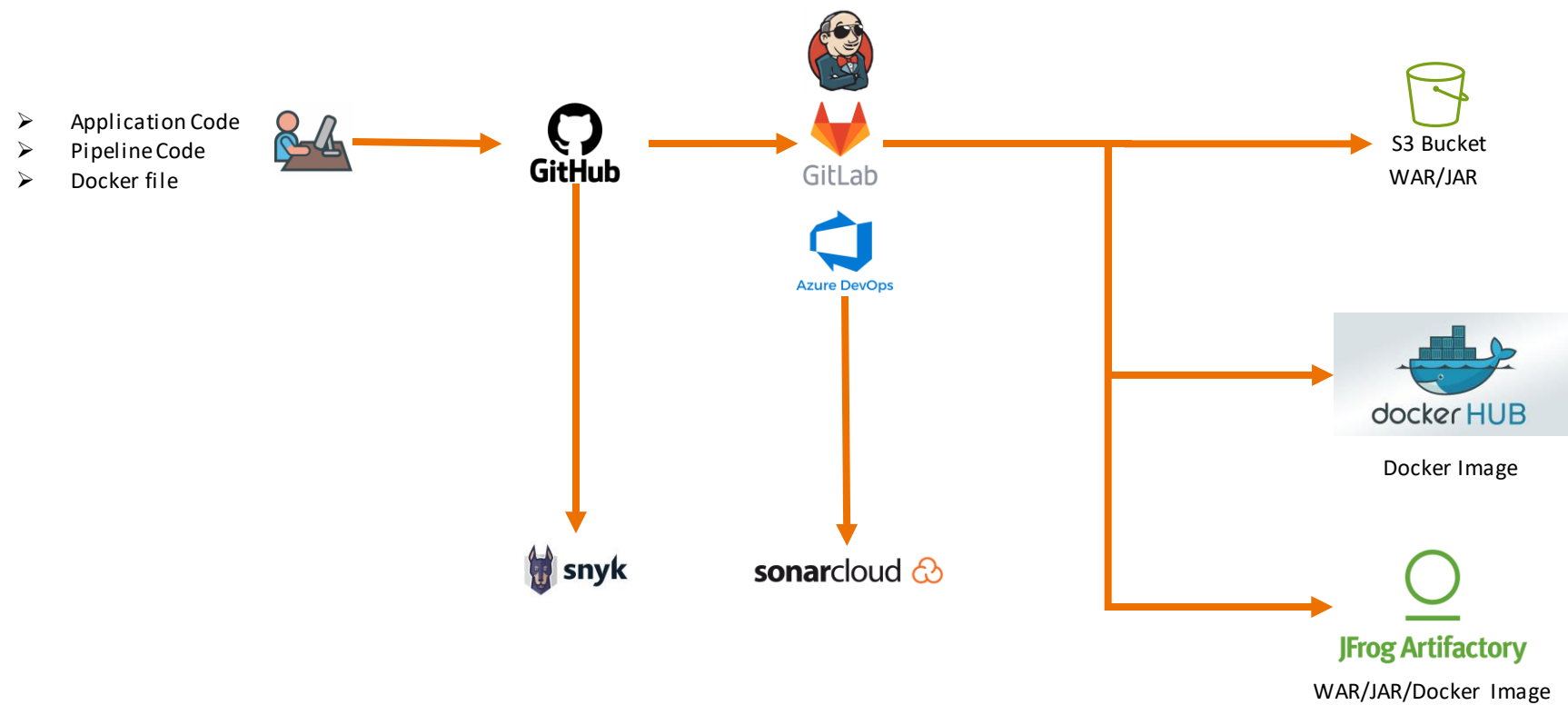
- We can use the following git repo for our project:

<https://github.com/spring-projects/spring-petclinic>

Or you can bring your own repo. Make sure it can be containerized and has RDS.



High Level Flow



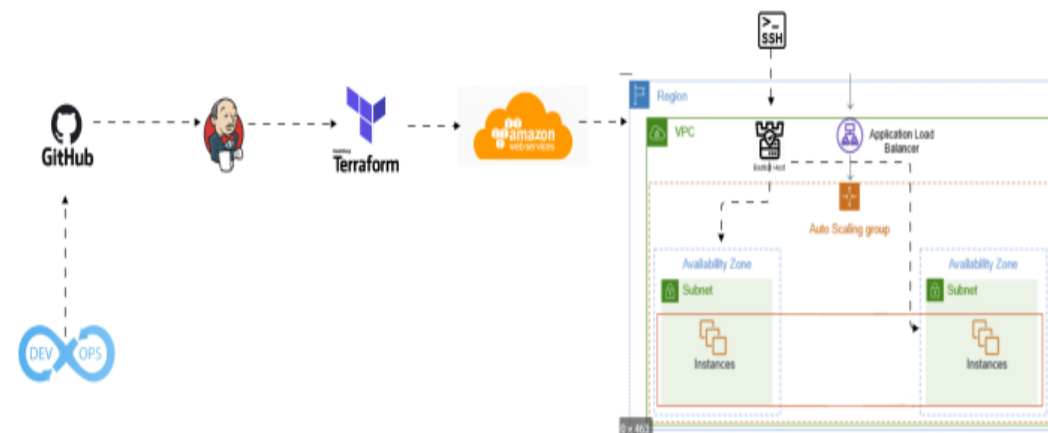
Key requirements for Application Build

1. Use any CI tool, we don't care.
2. Make sure the code is scanned using Sonar Cloud (Create a free account on <https://sonarcloud.io>)
3. The code needs to be scanned and built on every branch code commit.
4. Freshers or Beginners can just build the project using Maven and produce the war/jar files as outcome.
5. The more experienced learners will produce the containerize docker image as final output.
6. Create a free account on **JFROG Artifactory** because this will be used to store JAR/Docker Image.
[If you struggle with Time, you can use the S3 to store WAR file and Docker Hub to store the Docker Image. However, no one does like that.]
7. Differentiate and Tag between final output produced by a feature branch or master/main branch.

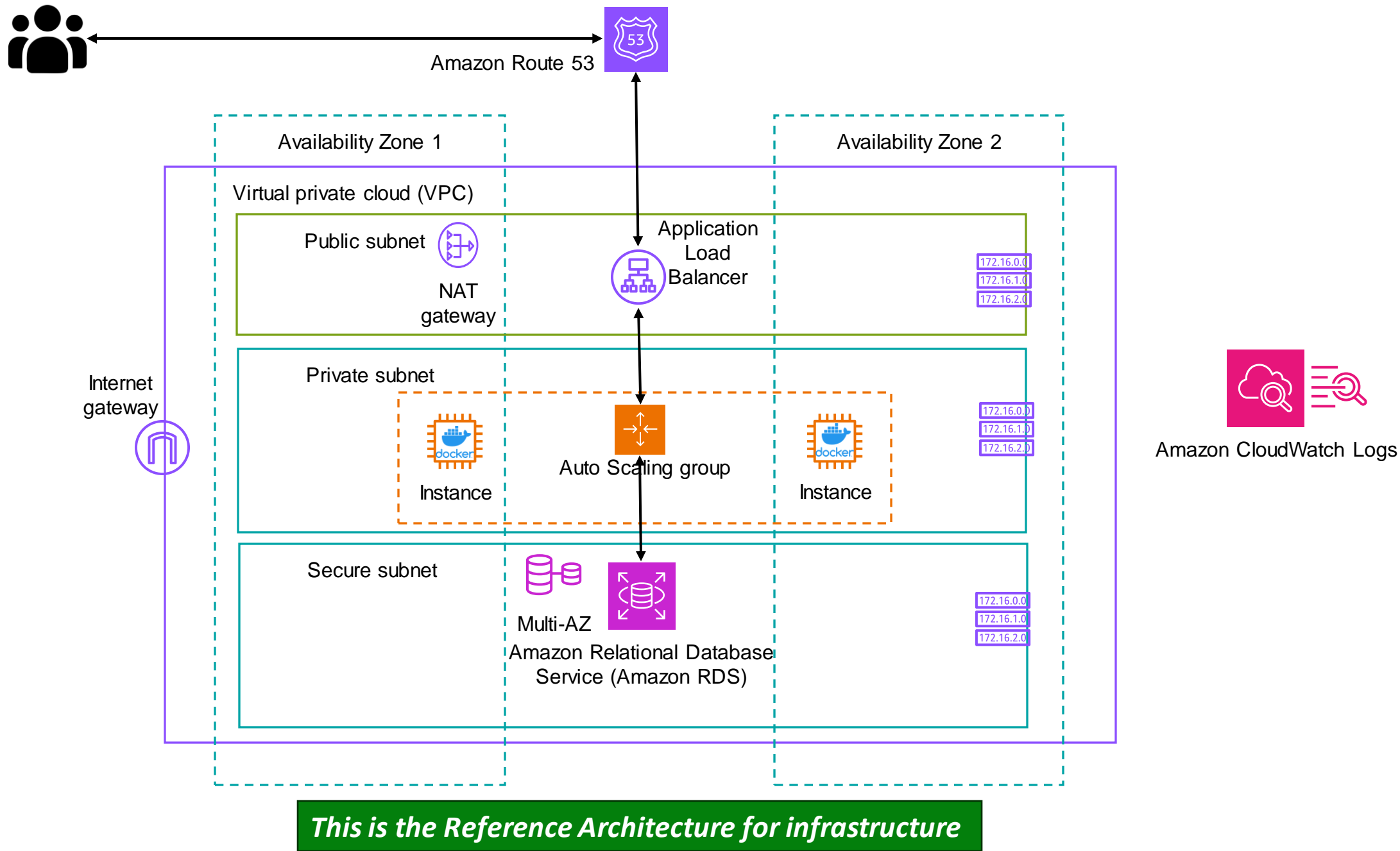
Creation of Infrastructure using Terraform

Estimated Time 12 Days

- Terraform is a mandatory skill. Used across the industry and in multi cloud scenario.
- It's easy to get started but can get really complicated later to manage.
- Best practices are a MUST.
- State must be in Cloud using S3 or Azure Blob Storage.
- For State file security enable S3 cross region replication, versioning and encryption
- Apply state locking using DynamoDB
- The CI/CD Pipeline to have three steps and checks:
 - **On commit to any Branch** perform terraform init, validate, fmt and plan operations, *checkov scan is optional*
 - **On pull request creation** — perform terraform init, validate, fmt and plan so that reviewer can see what is the change this pipeline will perform
 - **On merge** the feature is merged to master/ main and terraform apply happens using GitOps approach.



Reference pipeline for Infrastructure.



Key requirements for Infrastructure

1. 3 Subnets Architecture

Public, Private and Secure Subnets.

Public Subnets should have routes to Internet Gateway. Private Subnets should have route to NAT Gateway. Secure subnets should not have route to IGW or NAT GW.

2. Load balancer in Public Subnet

Create Load Balancer in Public Subnet across 2 AZs. Create corresponding Listener and Target Groups.

3. Create Auto Scaling Group

Create the ASG across 2 AZs in private subnet. Attach the ASG to ALB. Desired=1, Min=1, Max=1

The instances should not have public IPs.

Instances should be connecting using SSM or EC2 Instance Connect Endpoint.

Security group should not open port 22.

EC2 should be using a user data script at startup and install the application [Docker Image or WAR JAR File]. This should be in sync with the ALB Target groups.

Key requirements for Infrastructure

4. **Create a RDS cluster.**

Create the RDS cluster secure subnet in 2 AZs but can start with 1 AZ to cut cost.

5. **Logs [Stretch Goal]**

Application and or docker Logs to be collected in Cloud Watch Logs.

6. **Document RTO and RPO for this application**

Final Deliverables

1. Code Repo Public URL containing the solution: 1

Use your own GitHub account. Add this account to your resume.

*Remember to not commit any secret credentials in the code.

2. Project Title [1 Line] and Project Description [~1000 Words] 2

3. Video Describing Your Project [Google Drive Link] 3

You can split it into three smaller videos but finally target a Longer Crisp Video. Video to have architecture diagram before jumping to code.

[Optional] **Video Thumbnail** (You can use Canva free account to design it)

Once you provide these details on WhatsApp message, our Team will upload the video To DevOps and Cloud YouTube channel and issue you a project completion certificate.