1. Setting up a CI/CD Pipeline and deploying applications on AWS EKS
2. Source Code Management

* Infrastructure: [onggieoi/sd4414-aws-infrastructure-terraform (github.com)](https://github.com/onggieoi/sd4414-aws-infrastructure-terraform)
* Jenkins shared library: [onggieoi/devops-cicd (github.com)](https://github.com/onggieoi/devops-cicd)
* MSA: [onggieoi/sd4414\_msa (github.com)](https://github.com/onggieoi/sd4414_msa) (included **Document** folder)

1. Provision AWS Resources (VPC, EC2, ECR and EKS)

Apply terraform for bootstrap, network, ec2, ecr, eks:

VPC

A computer screen shot of a computer

Description automatically generated

EC2

A screenshot of a computer

Description automatically generated

ECR

A screenshot of a computer

Description automatically generated

EKS

A screenshot of a computer

Description automatically generated

1. Installation

Install Docker in EC2

A screenshot of a computer

Description automatically generated

Install Jenkins in EC2

A screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Setup Jenkins pipeline for CICD

Install suggested plugins and plugins mentioned in [nashtech-garage/devops-ci-cd (github.com)](https://github.com/nashtech-garage/devops-ci-cd)

* Jenkins suggested
* Docker PipelineVersion
* xUnit
* Cobertura
* Code Coverage API
* Pipeline Utility Steps
* Kubernetes
* CloudBees AWS Credentials
* Docker Pipeline
* Amazon ECR

Define stages to build and push images from msa to ECR and deploy to EKS in Jenkins shared library ([onggieoi/devops-cicd (github.com)](https://github.com/onggieoi/devops-cicd)).

Setup share library in Jenkins:

A screenshot of a computer

Description automatically generated

Trigger the pipeline:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

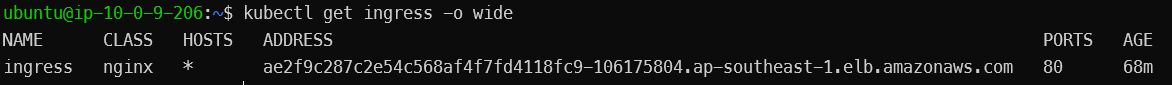
Description automatically generated

GetA screenshot of a computer

Description automatically generated pods:

A screen shot of a black screen

Description automatically generated

verify application ingress

As result:

A screenshot of a computer

Description automatically generated

A blue line on a white background

Description automatically generated

1. Monitoring

Install Prometheus Operator CRD to eks.

kubectl apply -f <https://raw.githubusercontent.com/prometheus-operator/prometheus-operator/master/bundle.yaml>

A screen shot of a computer

Description automatically generated

Install Prometheus, Grafana (follow from repo: [nashtech-garage/prometheus-grafana-monitoring (github.com)](https://github.com/nashtech-garage/prometheus-grafana-monitoring))

A screen shot of a computer

Description automatically generated

1. Reuse Jenkins CI and use GitOps for the CD pipeline.

## Install Argo CD on EKS Cluster

kubectl apply -n argocd -f <https://raw.githubusercontent.com/argoproj/argo-cd/stable/manifests/install.yaml>

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Create application in Argo CD

A screenshot of a computer

Description automatically generated

Check pod:

A black background with white text

Description automatically generated

Check service:

A black screen with white text

Description automatically generated

A screenshot of a computer

Description automatically generated

A black screen with white text

Description automatically generated