|  |
| --- |
|  |
| Problem Statement |

# Problem Statement

ABC Corporation using Cloud DevOps. The project requires to set up a Kubernetes cluster hosted at AWS/Azure, and deploy the Docker image using Jenkins Pipeline.

A web app is containerized and deployed to a kubernetes cluster. This webapp runs a simple apache site.

## Create following scripts

**1)** initial-setup.sh - This file is the first file to look at when setting up this project. It installs the required packakages to make this project work such as Docker, Jenkins, Kubectl, eksctl, hadolint etc.

**2)** kubernetes/ - This folder contains the kubernetes resource configuration files that will deploy the application image on AWS EKS using kubectl once configured properly.

**3)** Jenkinsfile - This file contains the definition of the stages in the pipeline. The stages in this project's pipeline are Lint files, Building image, Upload Image to Docker hub, Remove Unused docker image, Update Kube Config and Deploy Updated Image to Cluster.

**4)** Makefile - This file contains shell commands that can be executed using the Make linux tool.

**5)** Dockerfile - This file contains all the commands needed to assemble the app image.

**6)** run\_docker.sh - This file contains the shell commands needed to build the image on docker locally and also run a container.

**7)** run\_kubernetes.sh - This file contains the shell commands needed to deploy the app in a kubernetes cluster running locally.

**8)** upload\_docker.sh - This file contains the shell commands needed to upload the docker image to docker hub.

## 📃 Deploying on Amazon EKS using a CI/CD pipeline

**1)** A Jenkins server needs to be provisioned to setup your CI/CD pipeline. You can follow this tutorial to setup Jenkins on an Ubuntu Server.

**2)** Once your server is provisioned, you'll need to install Docker, ansible, hadolint, pylint and install the latest AWS CLI

**3)** The reason for installing the latest AWS CLI is to have the EKS module available for you. Once you install the AWS CLI you might not be able to run AWS commands. You can source your bash profile to get it working:

**❍ source ~/.bash\_profile**

**4)** With AWS CLI installed you can setup it up as the Jenkins User. Follow this tutorial to see how. This is to allow you run aws commands in your pipeline without getting errors. Also set up aws credentials on Jenkins dashboard.

**5)** You'll need to add the jenkins user to the sudoer group to enable it run ansible-playbooks. Follow this tutorial to see how to. Note: This is not best practices standard.

**6)** You might need to disable password request for the jenkins user for your pipeline to run smoothly. Note: This is not best practices standard.

**7)** Create swap space on your server. I got an out of memory error after running ansible multiple times. This is only necessary if the ansible-setup branch is used.

**8)** One of the stages in the pipeline requires Docker login information. You can set this up in Jenkins credentials. Save it as the same name as the registryCredentials in the Jenkinsfile. Also edit the registry in the Jenkinsfile to match your repo on Docker hub.

**9)** Once all this is done you can start working on Jenkins. Note your IAM user should have permissions to use EKS.

**10)** A command line tool 'sed' is used in updating the kubernetes configuration file in other to force the pods to update with the new image. The reason for this is that Kubernetes (wrongly) considers Docker tags as immutable (i.e., once a tag is set, it is never changed). The rolling update is also activated by the change in image name.

**System Requirements:**

Gitlab setup — free account

Jenkins master - VM1

Build server - VM1

Monitoring framework - VM2

Production server – VM2

AWS EKS