**DevOps Challenge**

# Following are the Tasks which have been completed. Coding files and screen shorts are attached for reference.

# Challenges

1. **Docker Whale:** Write a Docker file to run Energi Node in a container. It should somehow verify the checksum of the downloaded release (there's no need to build the project), run as a normal user, it should run the client, and print its output to the console.

https://wiki.energi.world/en/downloads/core-node

The build should be security conscious, and ideally pass a container image security test such as ECR, or Trivy.

**Answer:**   
Docker file has been written and verified. Its Pulling the image from the above shared repository, verifying the checksum. Its running the container and output printing on the console.

A computer screen shot of a program

Description automatically generated

**Code File Name: Dockerfile**

After Upload the image in ECR, ECR scanner ran and gave the following output. There is no high or Critical vulnerability found.

A screenshot of a computer

Description automatically generated

**Task completed as per requirements. Files are available in the repository.**

==========================================================

1. **K8S Awesomeness:** Write a Kubernetes StatefulSet to run the above, using persistent volume claims and resource limits.

**Answer:**

Write down the Kubernetes StatefulSet code and tested over locally setup minikube with mentioned requirements.

**Code File Name:**

**energi-node-statefulset.yaml**

**energi-node-external-service.yaml**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screen shot of a computer

Description automatically generated

**Task completed as per requirements. Files are available in the repository.**

==========================================================

1. **All the continuouses:** Write a simple build and deployment pipeline for the above using groovy / Jenkinsfile, Travis CI or Gitlab CI.

**Answer:**

I have used Jenkins to set up and deploy the pipelines. I have prepared 2 pipelines here with 2 different scenarios.

**3.1:** In first scenario, Jenkins performing the following steps.

1- Jenkins logging in to an EC2 Server

2- Taking the Docker file form there.   
 3- Making the build of that Docker File and making image.

4- Uploading the image into an ECR repository.

5- logging in to the ECR and downloading the image.

6- Deploying the image on the same EC2 server to run the container.

A screenshot of a computer

Description automatically generated

**Code File Name: Deploy-Docker-Container-Jenkinsfile**

**3.2:** In 2nd scenario, Jenkins performing the following steps.

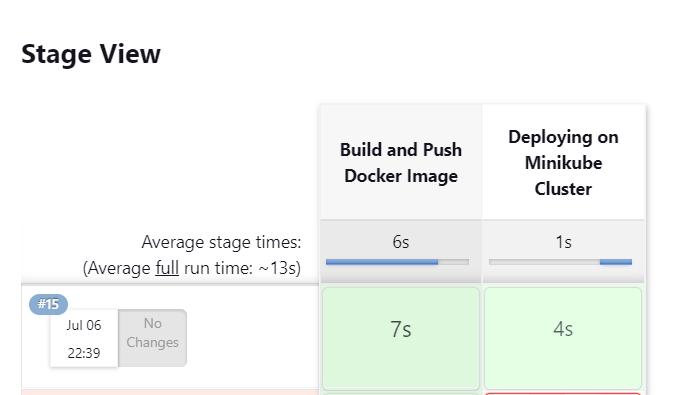
1- Jenkins logging in to an EC2 Server

2- Taking the Docker file form there.   
 3- Making the build of that Docker File and making image.

4- Uploading the image into an ECR repository.

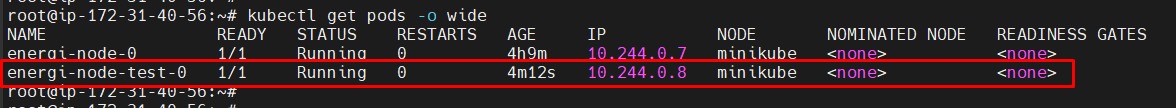
5- logging in to the ECR and downloading the image.

6- Deploying the image on the minikube Kubernetes server to run the container.



A screenshot of a computer

Description automatically generated



**Code File Name: Deploy-Minikube-Cluster-Jenkinsfile**

**Task completed as per requirements. Files are available in the repository.**

==========================================================

1. **Script kiddies:** Source or come up with a text manipulation problem and solve it with at least two of awk, sed, tr. and / or grep. Check the question below first though, maybe.

**Answer:**

I have created a text file in which add some email in form of upper and lower case, when script will run it covert it in lower case as a output.

==========================================================

1. **Script grown-ups:** Solve the problem in question 4 using any programming language of your choice.

**Answer:**

I have written down 2 scripts. Bash & Python as following file names.

**Code Files:**

text\_manipulation.sh

text\_manipulationpyt.py

**Source File:**

names\_emails.txt

**Task completed as per requirements. Files are available in the repository.**

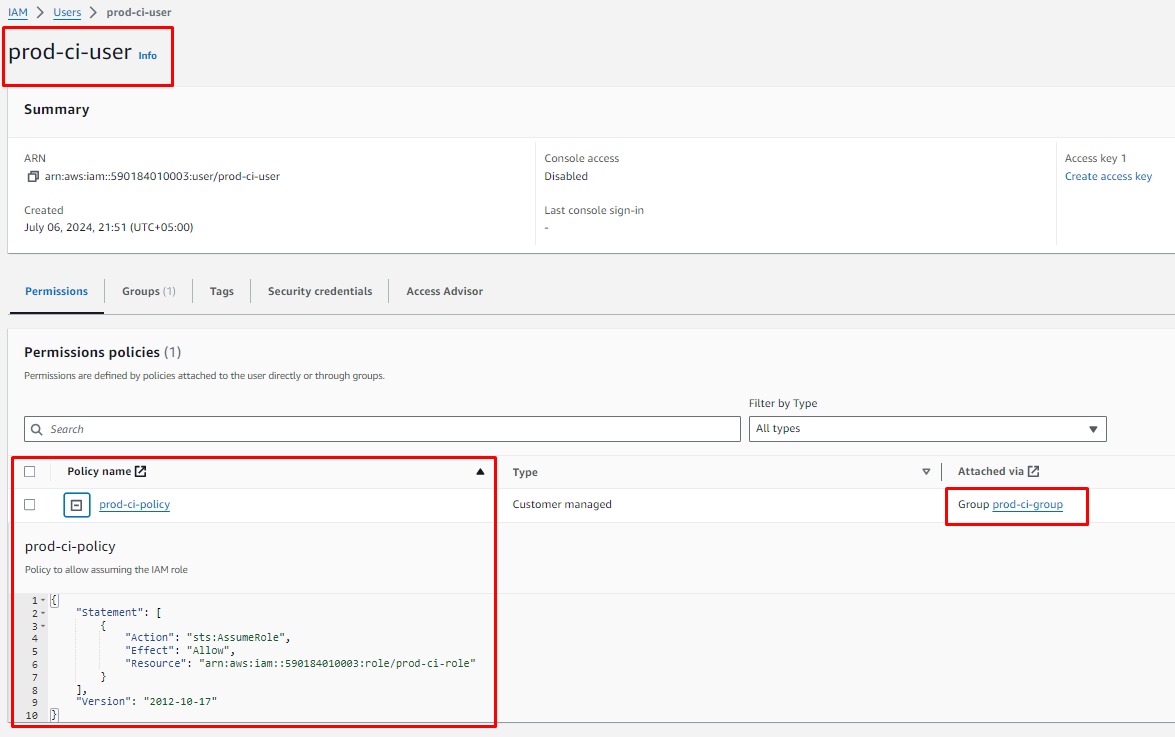
==========================================================

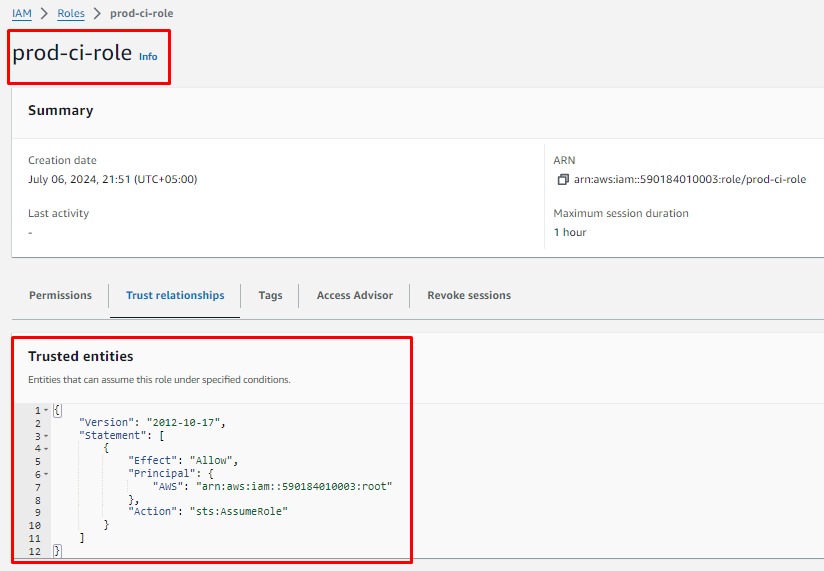
1. **Terraform lovers:** write a Terraform module that creates the following resources in IAM:
   * + A role, with no permissions, which can be assumed by users within the same account
     + A policy, allowing users / entities to assume the above role
     + A group, with the above policy attached
     + A user, belonging to the above group
     + All four entities should have the same name, or be similarly named in some meaningful way given the context e.g. prod-ci-role, prod-ci-policy, prod-ci-group, prod-ci-user; or just prod-ci. Make the suffixes toggleable, if you wish.

**Answer:**

Terraform code has been written to achieve the above requirements.

**Code Files: terraform-module.rar**





**Task completed as per requirements. Files are available in the repository.**