1.

FROM openjdk:8

MAINTAINER 'ARJUN'

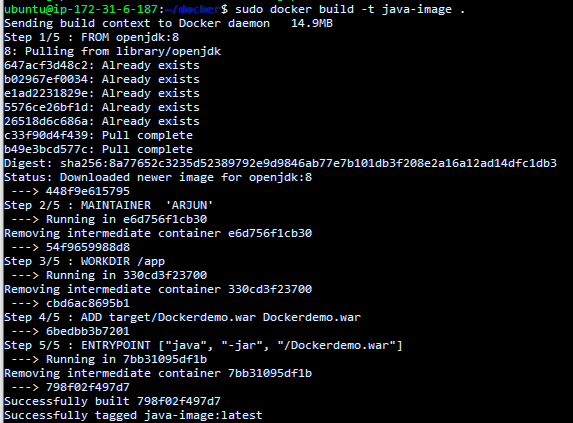
EXPOSE 8060

WORKDIR /app

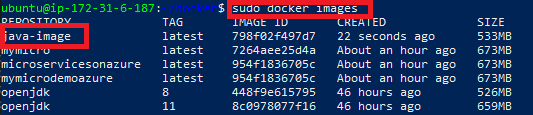
ADD target/Dockerdemo.war Dockerdemo.war

ENTRYPOINT ["java", "-jar", "/Dockerdemo.war"]

2.



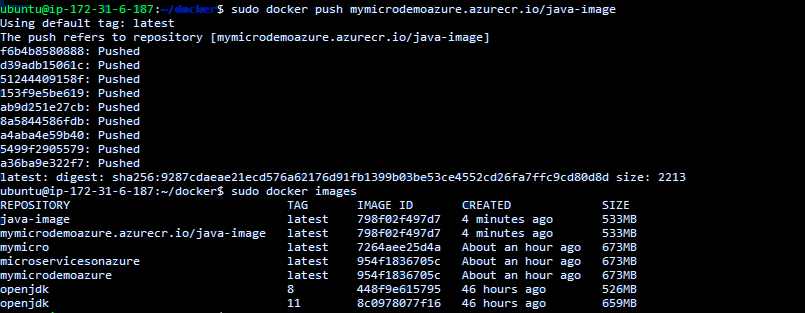
2.



3.

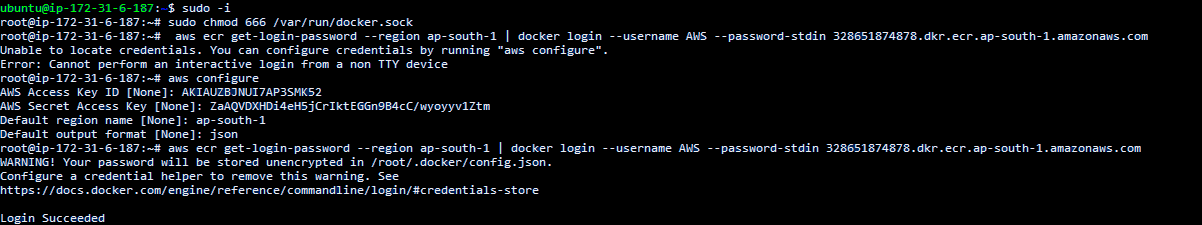


4.

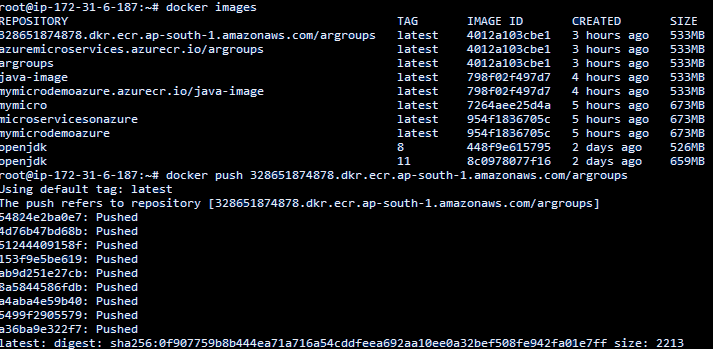


5. Aws also



6.

7.

Kubectl Install On AWS CLI

1. Download the Amazon EKS vended kubectl binary for your cluster's Kubernetes version from Amazon S3. To download the Arm version, change *amd64* to arm64 before running the command.
   * **Kubernetes 1.21:**

**curl -o kubectl https://amazon-eks.s3.us-west-2.amazonaws.com/1.21.2/2021-07-05/bin/linux/***amd64***/kubectl**

1. (Optional) Verify the downloaded binary with the SHA-256 sum for your binary.
   * Download the SHA-256 sum for your cluster's Kubernetes version for Linux. To download the Arm version, change *<amd64>* to arm64 before running the command.
     + **Kubernetes 1.21:**

**curl -o kubectl.sha256 https://amazon-eks.s3.us-west-2.amazonaws.com/1.21.2/2021-07-05/bin/linux/***amd64***/kubectl.sha256**

* + Check the SHA-256 sum for your downloaded binary.

**openssl sha1 -sha256 kubectl**

1. Apply execute permissions to the binary.

**chmod +x ./kubectl**

1. Copy the binary to a folder in your PATH. If you have already installed a version of kubectl, then we recommend creating a $HOME/bin/kubectl and ensuring that $HOME/bin comes first in your $PATH.

**mkdir -p $HOME/bin && cp ./kubectl $HOME/bin/kubectl && export PATH=$PATH:$HOME/bin**

1. (Optional) Add the $HOME/bin path to your shell initialization file so that it is configured when you open a shell.

**Note**

This step assumes you are using the Bash shell; if you are using another shell, change the command to use your specific shell initialization file.

**echo 'export PATH=$PATH:$HOME/bin' >> ~/.bashrc**

1. After you install kubectl , you can verify its version with the following command:

**kubectl version --short --client**

How to connect aws cluster to node and deploy application

1.  Verify that AWS CLI version 1.16.308 or greater is installed on your system:

$ aws --version

2. Check the current identity to verify that you're using the correct credentials that have permissions for the Amazon EKS cluster:

$ aws sts get-caller-identity

3. Create or update the kubeconfig file for your cluster:

aws eks --region region update-kubeconfig --name cluster\_name

4. Check your aws terminal using following code. it shows added node.

$ kubectl get pods