kubectl config set-context $(kubectl config current-context) --namespace=mycart

kubectl config view | grep namespace

kubectl describe deploy -n mycart

kubectl get rs -n mycart

kubectl get pods -n mycart

Lab: http://labsap.s3-website-us-west-1.amazonaws.com

Docs: https://drive.google.com/drive/folders/15Rg9bU1PTOw19oIQDel7E4meppiJhT9k?usp=sharing

Git hub:

git clone https://github.com/mohanraz81/awslogs.git

git clone https://github.com/mohanraz81/kube\_install\_kops.git

git clone https://github.com/mohanraz81/myapp1808.git

docker rmi `docker images|awk '{print $3}'`

---------------------------------------------------

Install Jenkins

Step 1. First of all we have to install java JDK.

$ sudo yum -y install java-1.8.20-openjdk.x86\_64

Remove Java 1.7.0 if installed

Amazon Linux

$ sudo yum -y remove java-1.7.0-openjdk

Step 2: Add Jenkins Repository

$ sudo wget -O /etc/yum.repos.d/jenkins.repo http://pkg.jenkins-ci.org/redhat-stable/jenkins.repo

$ sudo rpm --import http://pkg.jenkins-ci.org/redhat-stable/jenkins-ci.org.key

Step 3: Install Jenkins

$ sudo yum install -y jenkins

Step 4: Start the Jenkins service

AMAZON LINUX/RHEL 6 /Centos 6

$ sudo service jenkins start

$ sudo chkconfig jenkins on

Validate Jenkins is started:

$ sudo service jenkins status

Step 7. Access Jenkins.

Allow a Security Group in AWS EC2:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuidauthorizing-access-to-an-instance.html

e/

Finally, after the installation is complete you can visit the following address in your browser

http://your-ip-address:8080

Follow On screen instructions to finish the Installation steps

----------------------------------

helm plugin install https://github.com/hypnoglow/helm-s3.git

export AWS\_REGION=us-east-1

helm s3 init s3://<bucketname>/charts

helm repo add my-charts s3://<ccbucketname>/charts

helm s3 push ./test-chart-0.1.0.tgz my-charts

--------------------------------------------------------

wget https://eksworkshop.com/intermediate/230\_logging/deploy.files/fluentd.yml

kubectl apply -f fluentd.yml

------------------------------------------

kubectl config set-context $(kubectl config current-context) --namespace=namespace

# Validate it

$ kubectl config view | grep namespace:

gcr.io/google\_containers/hpa-example

kubectl rollout history deployment.v1beta1.apps/

kubectl rollout undo deployment.v1beta1.apps/

kubectl create namespace prometheus

helm install stable/prometheus \

--name prometheus \

--namespace prometheus \

--set alertmanager.persistentVolume.storageClass="gp2" \

--set server.persistentVolume.storageClass="gp2"

kubectl get all -n prometheus

kubectl create namespace grafana

helm install stable/grafana \

--name grafana \

--namespace grafana \

--set persistence.storageClassName="gp2" \

--set adminPassword="EKS!sAWSome" \

--set datasources."datasources\.yaml".apiVersion=1 \

--set datasources."datasources\.yaml".datasources[0].name=Prometheus \

--set datasources."datasources\.yaml".datasources[0].type=prometheus \

--set datasources."datasources\.yaml".datasources[0].url=http://prometheus-server.prometheus.svc.cluster.local \

--set datasources."datasources\.yaml".datasources[0].access=proxy \

--set datasources."datasources\.yaml".datasources[0].isDefault=true \

--set service.type=LoadBalancer

kubectl get all -n grafana

kubectl get secret --namespace grafana grafana -o jsonpath="{.data.admin-password}" | base64 --decode ; echo

docker build -t mohanraz81/cart-frontend:2.0 cart/

docker ps -a

docker stop cart-frontend; docker rm cart-frontend

docker run -d -p 9000:80 --name cart-frontend -e MY\_DB\_HOST\_WRITE=test mohanraz81/cart-frontend:2.0

docker push mohanraz81/cart-frontend:2.0

aws sts get-caller-identity

docker build -t prakashboa/logagent awslogs

docker run -d -v logvol:/mnt/log:ro mohanraz81/logagent:1.0

sudo curl -L "https://github.com/docker/compose/releases/download/1.25.3/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

sudo chmod +x /usr/local/bin/docker-compose

sudo ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose

CREATE DATABASE mydatadb;

CREATE USER 'mysqluser'@'%' IDENTIFIED BY 'password';

GRANT ALL PRIVILEGES ON \*.\* TO 'mysqluser'@'%';

FLUSH PRIVILEGES;

--------------------------------------------------------------------

---

version: "3.7"

services:

cart-frontend:

build:

context: ./frontend

image: mohanraz81/cart-frontend:3.0

environment:

MY\_DB\_HOST\_WRITE: myapp-db

MY\_DB\_HOST\_READ: myapp-db

MY\_DB\_NAME: mydatadb

MY\_DB\_USER: mysqluser

MY\_DB\_PASS: password

ports:

- "9000:80"

volumes:

- type: volume

source: cart-frontend-logvol

target: /var/log

logagent:

build:

context: ./awslogs

image: mohanraz81/logagent:1.0

depends\_on: cart-frontend

volumes:

- type: volume

source: cart-frontend-logvol

target: /mnt/log

myapp-db:

image: mysql:5.7

volumes:

- type: volume

source: myapp-db-vol

target: /var/lib/mysql

volumes:

cart-frontend-logvol:

myapp-db-vol:

------------------------------------------------------------------------

# Docker file for Cart Frontend

FROM centos:7

RUN yum -y update

ADD scripts /opt

RUN chmod +x /opt/\*.sh

RUN /opt/packages.sh

ADD config /root

RUN chmod +x /root/myenv.sh; source /root/myenv.sh

ADD code /var/www/html

CMD ["httpd", "-D", "FOREGROUND"]

---------------------------------------------------------------------

scripts/packages.sh

#!/bin/bash

yum -y install httpd php php-gd php-mysql

--------------------------------------------------------------------

config/myenv.sh

export MY\_DB\_HOST\_WRITE=mydb.example.com

----------------------------------------------------------

code/index.php

<?php

echo "<h1> This is the Three Tier Architecture Test New Mytest</h1>";

echo "<h3> Your Web Running Apache with php on Frontend container </h3>";

echo "<h3> Your APP Running Test Apache and PHP providing API services for User information on url http://backend/get\_user\_details </h2>";

echo "<h3> Your Database running Mysql/Mariadb holding user information in userdb on users Table </h1>";

function insertlabassoc($dbo,$sql)

{

$dbo->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$stmt = $dbo->prepare($sql);

$stmt->execute();

}

$labdbhost\_name = getenv("MY\_DB\_HOST\_WRITE");

$labdbhost\_read\_name = getenv("MY\_DB\_HOST\_READ");

$labdatabase = getenv("MY\_DB\_NAME");

$labusername = getenv("MY\_DB\_USER");

$labpassword = getenv("MY\_DB\_PASS");

try

{

$dbo = new PDO('mysql:host='.$labdbhost\_name.';dbname='.$labdatabase, $labusername, $labpassword);

}

catch (PDOException $e)

{

print "We have trouble in our System we will be back soon.";

die();

}

$candl\_backend\_query = "CREATE TABLE IF NOT EXISTS `users` (`id` int(11) NOT NULL AUTO\_INCREMENT, `Name` varchar(50) NOT NULL, `Age` smallint(2) NOT NULL, `Email` varchar(50) NOT NULL, PRIMARY KEY (`id`)) ENGINE=InnoDB AUTO\_INCREMENT=5 DEFAULT CHARSET=utf8mb4; INSERT INTO `users` (`id`, `Name`, `Age`, `Email`) VALUES(1, 'Mohan', 18, 'mohanraz@gmail.com'),(2, 'Raj', 22, 'mohan@y2ytech.com'),(3, 'veer', 18, 'mohanraz@gmail.com'),(4, 'Raj', 22, 'mohan@y2ytech.com');";

$candl\_backend\_array = insertlabassoc($dbo, $candl\_backend\_query );

function querymultirowlab($dbo,$sql)

{

$dbo->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$stmt = $dbo->prepare($sql);

$stmt->execute();

$r = $stmt->fetchall();

return $r;

}

try

{

$dbo = new PDO('mysql:host='.$labdbhost\_read\_name.';dbname='.$labdatabase, $labusername, $labpassword);

}

catch (PDOException $e)

{

print "We have trouble in our System we will be back soon.";

die();

}

$candl\_backend\_query = "select \* from users";

$candl\_backend\_array = querymultirowlab($dbo, $candl\_backend\_query );

foreach($candl\_backend\_array as $value)

{

echo "Name"."=>".$value['Name']."<br>";

echo "Age"."=>".$value['Age']."<br>";

echo "Email"."=>".$value['Email']."<br>";

echo "==========================================="."<br>";

}

?>

docker run -d -p 9000:80 --name flackapp -v logvol:/var/log -v /home/ec2-user:/home/ec2user -e MY\_DB\_HOST\_WRITE=test prakashboa/cart-frontend:2.0