Kubernetes Certification Training - Certification Project

Certification Project

edureka!

edureka!

Certification Project

Problem Statement

You work for ABC company as a Technical Architect. The company has deployed its architecture on containers.

The current deployment is on Docker Swarm, you have suggested to use Kubernetes for the company's applications.

Following is the desired Architecture:

Requirements
Create a free-tier account on AWS for this project, use 3 t2.micro instances for the cluster

There are 6 applications to be deployed,

1 Database System(Persistent)

2 Web Portals(Stateless)

The Web Portals should be available at,

example.com/users

example.com/admin

example.com should also point to example.com/users

Rest all the redirects should be towards the default page of ingress.

The Database System should not be exposed to the internet, and will be used by the web portals. The credentials for the database should be injected in the web portal using secrets.

(The secrets (password for db and hostname) should be passed as files to /var/secrets), the files should be named hostname.txt and password.txt respectively.

The database should have persistent volume attached to it, which is available remotely on a NFS Filesystem (EFS on AWS)

The nodes should be tainted based on,

Frontend Nodes - 2 Web portals

Backend Nodes - 1 Database

The default backend page for the ingress, should be modified to have the following data: "Custom Default Backend"

Also this complete website should be available at an AWS Load Balancer.

This architecture needs to be presented to the CTO, before it can get published on the production server.

Use the following sample applications for filling the required resources:

1 Database System(Persistent) -

hshar/mysql:5.5

Username: root

Password: edureka

MountPath for MySQL PV: /var/lib/mysql

Initialize the Database and table of MySQL using the following commands,

Create database edureka;

Create table edureka(name varchar(20), email varchar(20));

2 Web Portals(Stateless)

hshar/admin

hshar/users

The whole architecture should be self-healing.