

Project Title: Deploying Web Application on Kubernetes Cluster

Scenario

You have a multi-tier web application stack that is already containerized and tested. Now it's time to host it for production, ensuring high availability, fault tolerance, scalability, platform independence, portability, and flexibility.

Requirements

- **High Availability**
- **Fault Tolerance**
- **Easily Scalable**
- **Platform Independent**
- **Portable & Flexible**

Method

Use Kubernetes as the container orchestration tool to manage and deploy the containerized applications.

Steps of Execution

1. **Set Up Kubernetes Cluster**
2. **Containerized Applications (vProfile)**
3. **Create EBS Volume for DB Pod**
4. **Label Nodes with Zone Names**
5. **Kubernetes Definition Files**
 - **Deployment:** Define the deployments for each tier of the application (e.g., front-end, back-end, database).
 - **Service:** Create services to expose the deployments and allow communication between them.
 - **Secret:** Store sensitive data such as database passwords using Kubernetes Secrets.
 - **Volume:** Define the PV and PVC for the database EBS volume.

Setup kops cluster

```
ubuntu@ip-172-31-5-94:~$ kops create cluster --name=kpop.hkHING.xyz \
> --state=s3://kopsbucketlist --zones=us-west-1b,us-west-1c \
> --node-count=2 --node-size=t3.small --master-size=t3.medium --dns-zone=kpop.hkHING.xyz \
> --node-volume-size=8 --master-volume-size=8
Flag --master-size has been deprecated, use --control-plane-size instead
Flag --master-volume-size has been deprecated, use --control-plane-volume-size instead
I0630 06:43:39.404836    3290 new_cluster.go:1445] Cloud Provider ID: "aws"
I0630 06:43:39.490254    3290 subnets.go:224] Assigned CIDR 172.20.0.0/17 to subnet us-west-1b
I0630 06:43:39.490378    3290 subnets.go:224] Assigned CIDR 172.20.128.0/17 to subnet us-west-1c
Previewing changes that will be made:
```

```
ubuntu@ip-172-31-5-94:~$ kops update cluster --name kpop.hkHING.xyz --state=s3://kopsbucketlist --yes --admin
I0630 06:46:52.559631    3294 executor.go:113] Tasks: 0 done / 113 total; 45 can run
I0630 06:46:52.620387    3294 vfs_keystore_reader.go:143] CA private key was not found
I0630 06:46:52.631523    3294 keypair.go:226] Issuing new certificate: "etcd-man
```

Make a GitHub repo and clone it to your local machine

The screenshot shows the GitHub web interface for a repository named 'kubernetes-app' owned by 'Aditi55Pathak'. The repository is public. The page includes a search bar, navigation tabs for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below the repository name, there are buttons for Pin, Unwatch (1), Fork (0), and Star (0). Two main cards are visible: 'Start coding with Codespaces' and 'Add collaborators to this repository'. A 'Quick setup' section provides instructions for cloning the repository using Desktop, HTTPS, or SSH, with the URL 'https://github.com/Aditi55Pathak/kubernetes-app.git'. At the bottom, there is a section for creating a new repository on the command line with a sample script.

```
echo "# kubernetes-app" >> README.md
git init
git add README.md
git commit -m "first commit"
```

Check health of your cluster

```
I0630 07:08:19.890147 3375 executor.go:113] Tasks: 0 done / 113 total; 43 can
run
I0630 07:08:20.302390 3375 executor.go:113] Tasks: 45 done / 113 total; 25 ca
n run
I0630 07:08:20.620459 3375 executor.go:113] Tasks: 70 done / 113 total; 31 ca
n run
I0630 07:08:21.113464 3375 executor.go:113] Tasks: 101 done / 113 total; 3 ca
n run
I0630 07:08:21.190608 3375 executor.go:113] Tasks: 104 done / 113 total; 9 ca
n run
I0630 07:08:21.328704 3375 executor.go:113] Tasks: 113 done / 113 total; 0 ca
n run
I0630 07:08:21.538674 3375 dns.go:235] Pre-creating DNS records
I0630 07:08:21.539632 3375 update_cluster.go:338] Exporting kubeconfig for cl
uster
kops has set your kubectl context to kpop.hkHING.xyz

Cluster changes have been applied to the cloud.

Changes may require instances to restart: kops rolling-update cluster
```

Create a EBS volume to store MySQL data

```
ubuntu@ip-172-31-5-94:~$ aws ec2 create-volume --availability-zone=us-west-1b --
size=3 --volume-type=gp2

{
  "AvailabilityZone": "us-west-1b",
  "CreateTime": "2024-06-30T07:18:53.000Z",
  "Encrypted": false,
  "Size": 3,
  "SnapshotId": "",
  "State": "creating",
  "VolumeId": "vol-0e0d78bfb39d7d1a1",
  "Iops": 100,
  "Tags": [],
  "VolumeType": "gp2",
  "MultiAttachEnabled": false
}

ubuntu@ip-172-31-5-94:~$ |
```

Note your volume id : vol-0e0d78bfb39d7d1a1

```
ubuntu@ip-172-31-5-94:~$ kubectl get nodes
NAME                                STATUS    ROLES    AGE   VERSION
i-052053f36889a5eae               Ready    control-plane   29m   v1.29.6
i-0bf7a0e15db22afde               Ready    node            27m   v1.29.6
i-0c30c00c5dd48643e               Ready    node            27m   v1.29.6
ubuntu@ip-172-31-5-94:~$
```

```

i-0c30c00c5dd48643e   Ready    node           27m    v1.29.6
ubuntu@ip-172-31-5-94:~$ kubectl describe node i-0bf7a0e15db22afde | grep us-west-1
        failure-domain.beta.kubernetes.io/region=us-west-1
        failure-domain.beta.kubernetes.io/zone=us-west-1c
        topology.ebs.csi.aws.com/zone=us-west-1c
        topology.kubernetes.io/region=us-west-1
        topology.kubernetes.io/zone=us-west-1c
InternalDNS:  i-0bf7a0e15db22afde.us-west-1.compute.internal
Hostname:     i-0bf7a0e15db22afde.us-west-1.compute.internal
ExternalDNS:  ec2-18-144-54-214.us-west-1.compute.amazonaws.com
ProviderID:   aws:///us-west-1c/i-0bf7a0e15db22afde
ubuntu@ip-172-31-5-94:~$ |

```

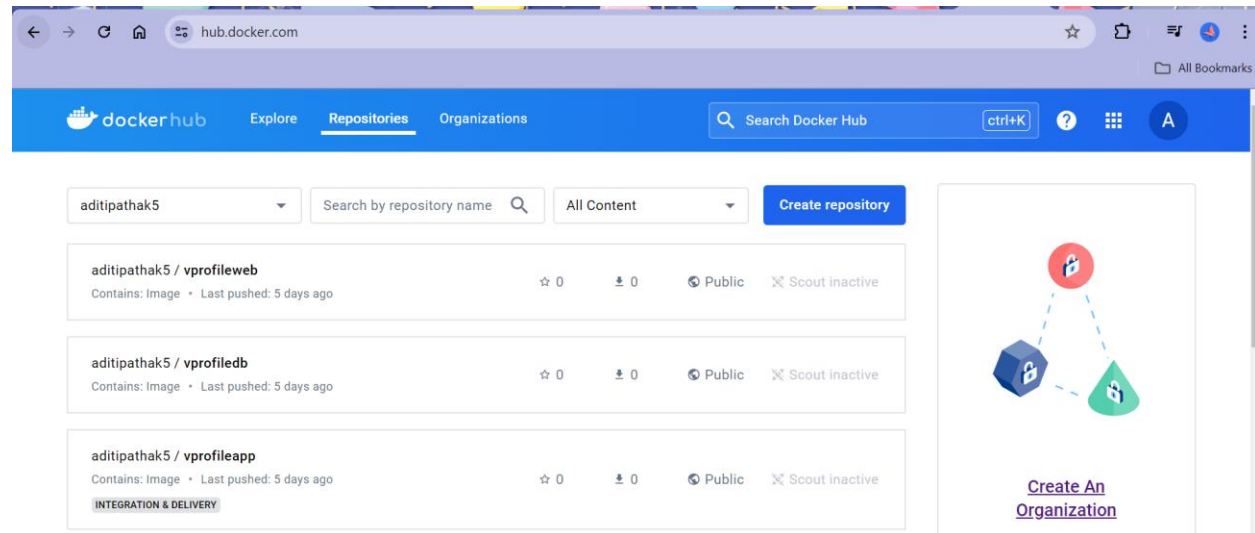
Give labels to the nodes

```

ProviderID:   aws:///us-west-1c/i-0bf7a0e15db22afde
ubuntu@ip-172-31-5-94:~$ kubectl label nodes i-0bf7a0e15db22afde zones=us-west-1b
node/i-0bf7a0e15db22afde labeled
ubuntu@ip-172-31-5-94:~$ kubectl label nodes i-0c30c00c5dd48643e zones=us-west-1c
node/i-0c30c00c5dd48643e labeled
ubuntu@ip-172-31-5-94:~$ |

```

Go to docker hub and check for the repositories that are already containerized



The screenshot shows the Docker Hub interface for the user 'aditipathak5'. The 'Repositories' tab is selected, displaying a list of three repositories:

- aditipathak5 / vprofileweb**: Contains: Image, Last pushed: 5 days ago, 0 stars, 0 downloads, Public, Scout inactive.
- aditipathak5 / vprofiledb**: Contains: Image, Last pushed: 5 days ago, 0 stars, 0 downloads, Public, Scout inactive.
- aditipathak5 / vprofileapp**: Contains: Image, Last pushed: 5 days ago, 0 stars, 0 downloads, Public, Scout inactive. A tag 'INTEGRATION & DELIVERY' is visible.

On the right side, there is a 'Create repository' button and a 'Create An Organization' link with a diagram showing a network of nodes.

Creating definitions file:

- Creating definition file for secret check the GitHub repo.
- Write a db definition file
- Write file called db-CIP.yaml
- Memecache deployment file
- Likewise same for RabbitMQ

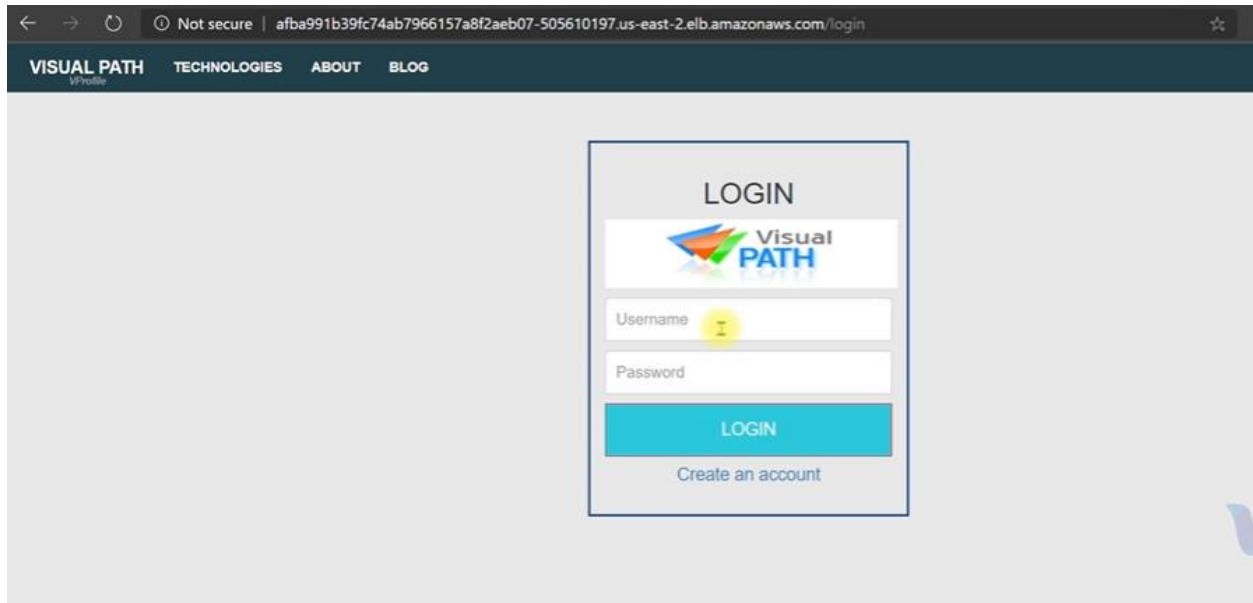
Commit changes to Git Hub.

```
ubuntu@ip-172-31-5-94:~/kubernetes-app$ kubectl create -f .
service/vproddb created
service/vprocache01 created
deployment.apps/vpromc created
service/vpromq01 created
deployment.apps/vpromq01 created
deployment.apps/vproapp created
service/vproapp-service created
error from server (AlreadyExists): error when creating "app-secret.yaml": secret "app-secret" already exists
error from server (AlreadyExists): error when creating "vprofdep.yaml": deployment.apps "vprofdb" already exists
ubuntu@ip-172-31-5-94:~/kubernetes-app$
```

```
ubuntu@ip-172-31-5-94:~/kubernetes-app$ kubectl get deploy
NAME          READY    UP-TO-DATE    AVAILABLE    AGE
vproapp       0/1      1             0            30s
vprofdb       1/1      1             1            98m
vpromc        1/1      1             1            31s
vpromq01      0/1      1             0            30s
ubuntu@ip-172-31-5-94:~/kubernetes-app$ |
```

```
ubuntu@ip-172-31-5-94:~/kubernetes-app$ kubectl get svc
NAME                                TYPE                CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
kubernetes                         ClusterIP           100.64.0.1      <none>           443/TCP          3h18m
vproapp-service                    LoadBalancer       100.68.201.181  a79854c6ad8a24179a58ca2d15b5a9-1666724162.us-west-1.elb.amazonaws.com  80:31799/TCP     3m50s
vprocache01                        ClusterIP           100.69.148.251  <none>           11211/TCP        3m51s
vproddb                           ClusterIP           100.68.231.196  <none>           3306/TCP         3m51s
vpromq01                          ClusterIP           100.68.154.14   <none>           15672/TCP        3m50s
ubuntu@ip-172-31-5-94:~/kubernetes-app$
```


Check with Loadbalancers endpoint : a79854c6ad8a24179a58ca2d15b53ea9-1666724162.us-west-1.elb.amazonaws.com




← → ↻ ⓘ Not secure | afba991b39fc74ab7966157a8f2aeb07-505610197.us-east-2.elb.amazonaws.com/login ☆

VISUAL PATH © 2018 TECHNOLOGIES ABOUT BLOG

LOGIN



Username 

Password

LOGIN

[Create an account](#)