To see the contexts:

kubectl config get-contexts

Create 3 clusters one hub and 2 spoke cluster:

eksctl create cluster --name hub-cluster --region us-east-1

eksctl create cluster --name spoke-cluster-1 --region us-east-1

eksctl create cluster --name spoke-cluster-2 --region us-east-1

A screenshot of a computer

Description automatically generated

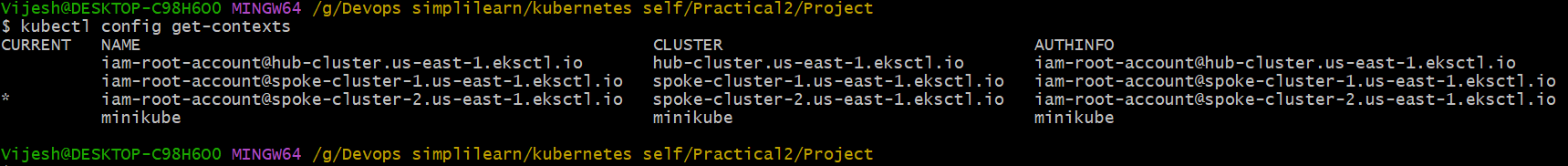
A screenshot of a computer

Description automatically generated

A screenshot of a computer

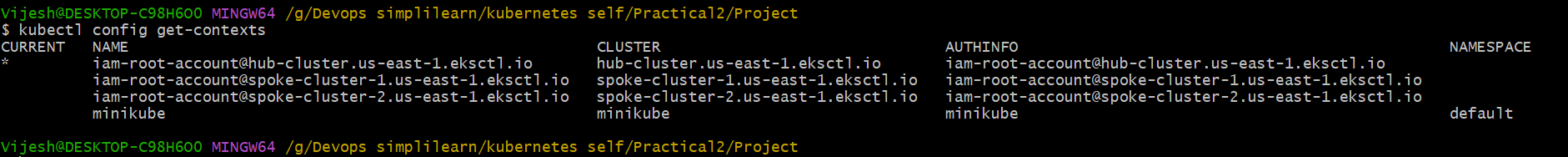
Description automatically generated

After the three clusters are ready switch the context to the hub cluster so all the activities will be done on the hub cluster:



kubectl config use-context [iam-root-account@hub-cluster.us-east-1.eksctl.io](mailto:iam-root-account@hub-cluster.us-east-1.eksctl.io)

$ kubectl config get-contexts



Now setup argocd in hub cluster:

kubectl create namespace argocd

kubectl apply -n argocd -f <https://raw.githubusercontent.com/argoproj/argo-cd/stable/manifests/install.yaml>

Argocd cd can be run in two modes

443 – Secure mode using CA certiciates and keys

HTTP - Insecure mode

We will run in insecure mode using http for now.

A screenshot of a computer

Description automatically generated

A black screen with yellow text

Description automatically generated

To run in insecure mode edit the config map parameters:

A black screen with yellow text

Description automatically generated

## Run Argo CD in HTTP Mode(Insecure)

<https://github.com/argoproj/argo-cd/blob/54f1572d46d8d611018f4854cf2f24a24a3ac088/docs/operator-manual/argocd-cmd-params-cm.yaml#L82>

A screenshot of a computer

Description automatically generated

kubectl edit cm argocd-cmd-params-cm -n argocd

data:

server.insecure: "true"

Add the above snippet in the configmap at the end and save

To verify the above changes :

kubectl get deploy -n argocd

A screen shot of a computer

Description automatically generated

kubectl describe deploy argocd-server -n argocd

A screen shot of a computer

Description automatically generated

kubectl edit deploy argocd-server -n argocd

A yellow line on a white background

Description automatically generated

Change the argocd server service to nodeport from clusterip:

kubectl get svc -n argocd

A screen shot of a computer

Description automatically generated

kubectl edit svc argocd-server -n argocd

A screen shot of a computer screen

Description automatically generated

Now copy the port take the ip addres of any ec2 instance within the hub cluster and browse . This will open the argocd console:

A computer screen shot of a cartoon octopus

Description automatically generated

Get the password from secrets:

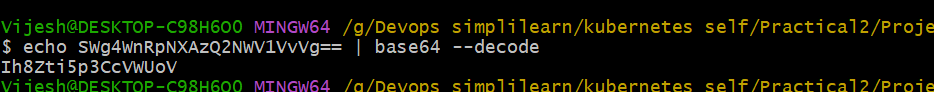
kubectl get secrets -n argocd

A black background with white text

Description automatically generated

kubectl edit secrets argocd-initial-admin-secret -n argocd

echo SWg4WnRpNXAzQ2NWV1VvVg== | base64 –decode



A screenshot of a computer

Description automatically generated

Now we have to add the other two clusters to the argocd using argocd command line, there is no such option from console. From console we can only delete.

Install Argocd CLI:

<https://argo-cd.readthedocs.io/en/stable/cli_installation/#installation>

for windows:

A computer screen shot of a blue screen

Description automatically generated

A screenshot of a computer

Description automatically generated

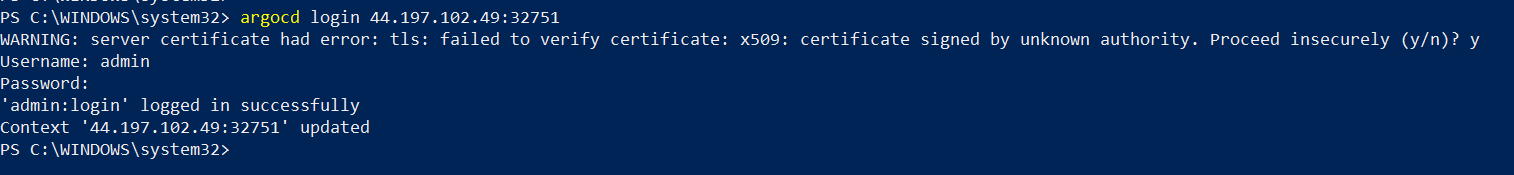
Add the other spoke clusters one by one:

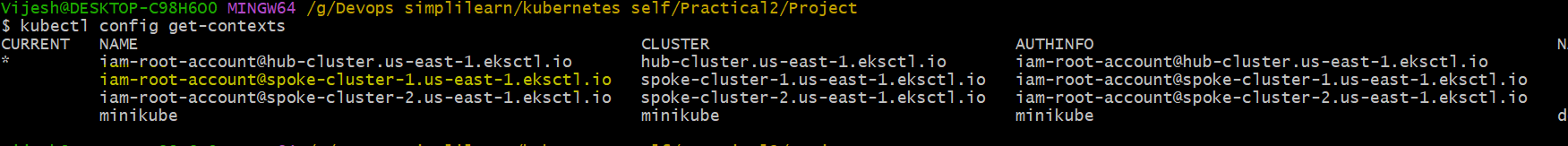
Login into argocd using command line:

Try logging in from powershell else there is some issue with git bash

argocd login 44.197.102.49:32751

This is the ip address of the argocd console.





argocd cluster add iam-root-account@spoke-cluster-1.us-east-1.eksctl.io --server 44.197.102.49:32751

A blue screen with white text

Description automatically generated

Add the second spoke :

A computer screen shot of a blue screen

Description automatically generated

Added both the clusters:

A screenshot of a computer

Description automatically generated

Now create two applications for both the clusters added, so that whenever any change happens in git it is reflected in both the clusters:

<https://github.com/vijeshnair89/argocd-hub-spoke-demo>

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A white background with blue lines

Description automatically generated

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Now switch the context to both the clusters and the application should be visible:

kubectl config use-context [iam-root-account@spoke-cluster-1.us-east-1.eksctl.io](mailto:iam-root-account@spoke-cluster-1.us-east-1.eksctl.io)

A computer screen shot of a blue screen

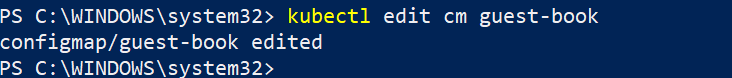
Description automatically generated

A computer screen shot of white text

Description automatically generated

Now to verify self healing, lets change some configuration in configmap of the spoke 2:

Tried to change the properties , but not able to change.



ui\_properties\_file\_name: abhishek-interface.properties to

ui\_properties\_file\_name: vijesh-interface.properties

Within the cluster not able to change now lets change from the git repository.

A screenshot of a computer

Description automatically generated

Wait for 3-4 minutes for auto sync to happen.

A computer screen shot of a blue screen

Description automatically generated

Sync successfully.

Delete the cluster:

eksctl delete cluster --name hub-cluster --region us-east-1

eksctl delete cluster --name spoke-cluster-1 --region us-east-1

eksctl delete cluster --name spoke-cluster-2 --region us-east-1

---------------------------------------Complete --------------------------------------------------------------