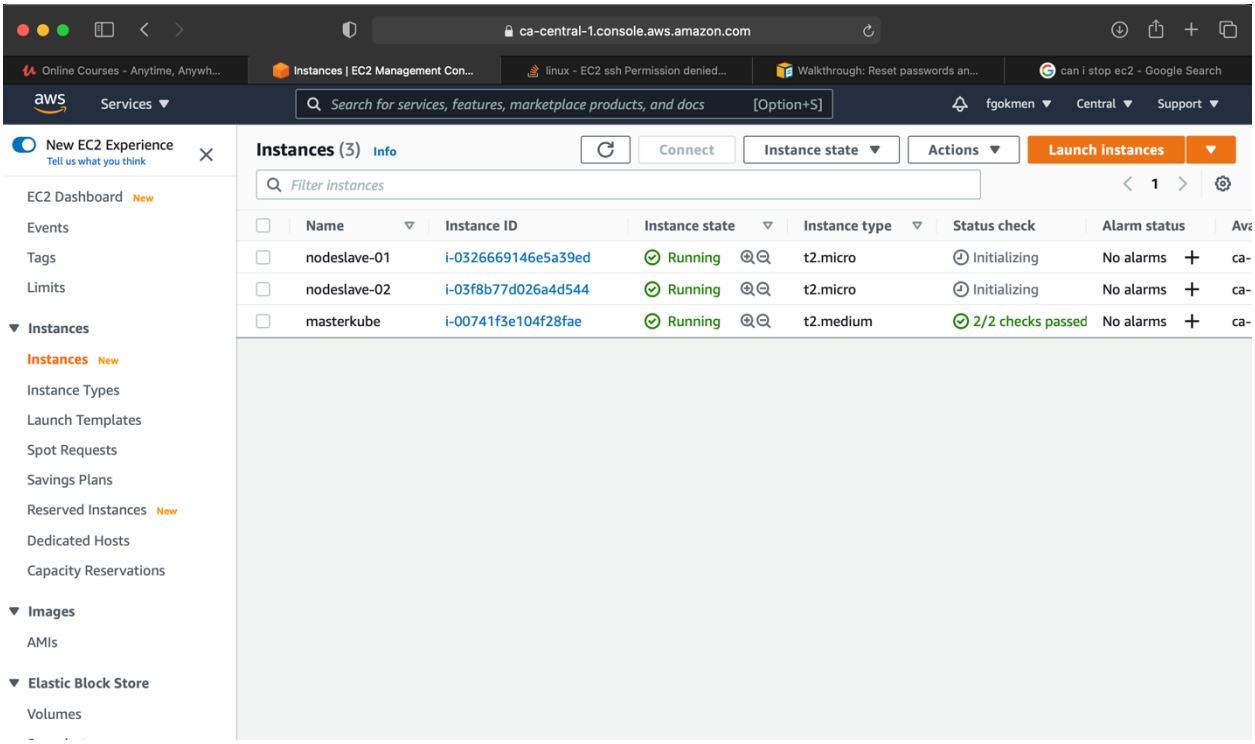
Felix Gokmen

Built and Created a CI/CD pipeline to Deploy Applications on Kubernetes Cluster by using helm and Jenkinsfile.
Used AWS and created 4 instances.
Two instance for worknodes
One instance for master nodes
One instance for Jenkins and Kubernetes – helm to test and run

1-First create ec2 virtual machine. I have chosen AWS as centos
Created one for master node
Created for nodeslave with two instances



2- Change the host name for each VM

```
Last login: Sat May 22 06:50:07 2021 from cpe688f2e0fdbb3-cm688f2e0fdbb0.cpe.net.cable.rogers.com
[centos@nodeslave-01 ~]$ _
```

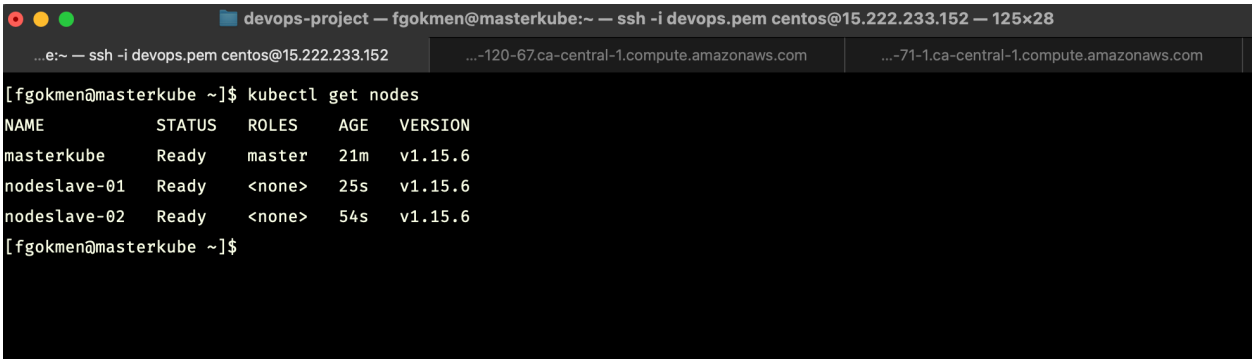
```
Last login: Sat May 22 06:52:12 2021 from cpe688f2e0fdbb3-cm688f2e0fdbb0.cpe.net.cable.rogers.com
[centos@nodeslave-02 ~]$
```

```
Last login: Sat May 22 06:37:38 2021 from cpe688f2e0fdbb3-cm688f2e0fdbb0.cpe.net.cable.rogers.com
[centos@masterkube ~]$ sudo yum install -y yum-utils_
```

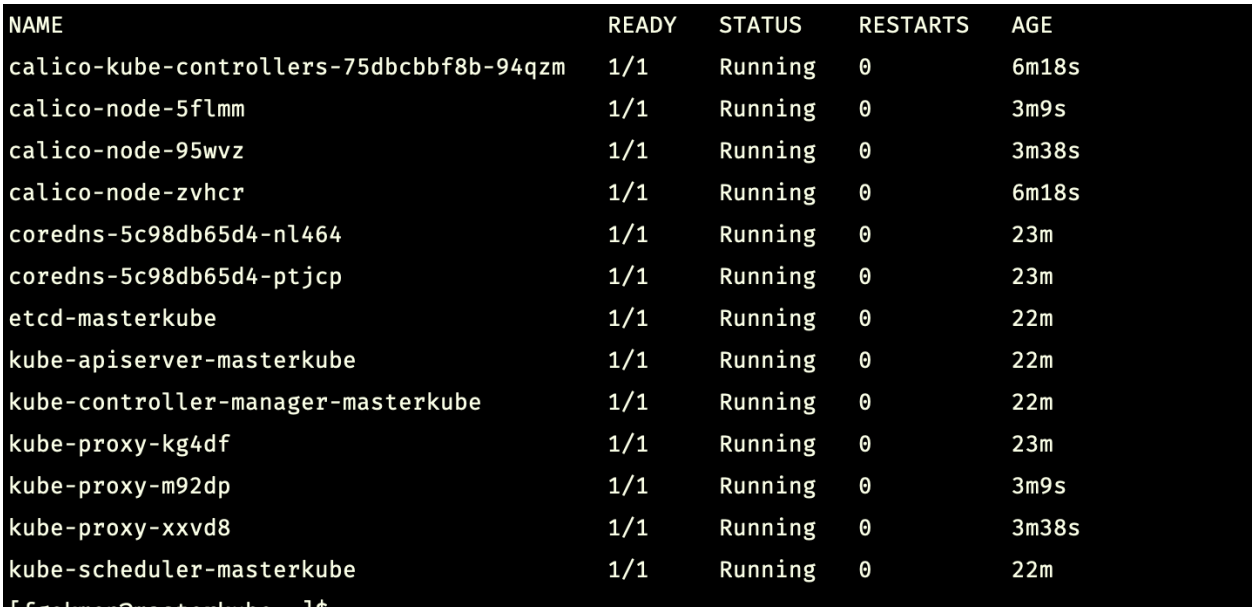
3- Master VM updated with apt and created docker installation

👉. https://docs.docker.com/engine/install/centos/

3-The cluster nodes joined to master nodes.



☑ Pods are running



4- Created server linux AWS and made the installation properly for Jenkins

Instances (4) Info

Filter instances

Name	Instance ID	Instance state	Instance type	Status check
nodeslave-01	i-0326669146e5a39ed	Running	t2.micro	2/2 checks passed
nodeslave-02	i-03f8b77d026a4d544	Running	t2.micro	2/2 checks passed
masterkube	i-00741f3e104f28fae	Running	t2.medium	2/2 checks passed
server	i-0a9c750cff1a4c5c8	Running	t2.medium	2/2 checks passed

5- Helm created inside the Jenkins.

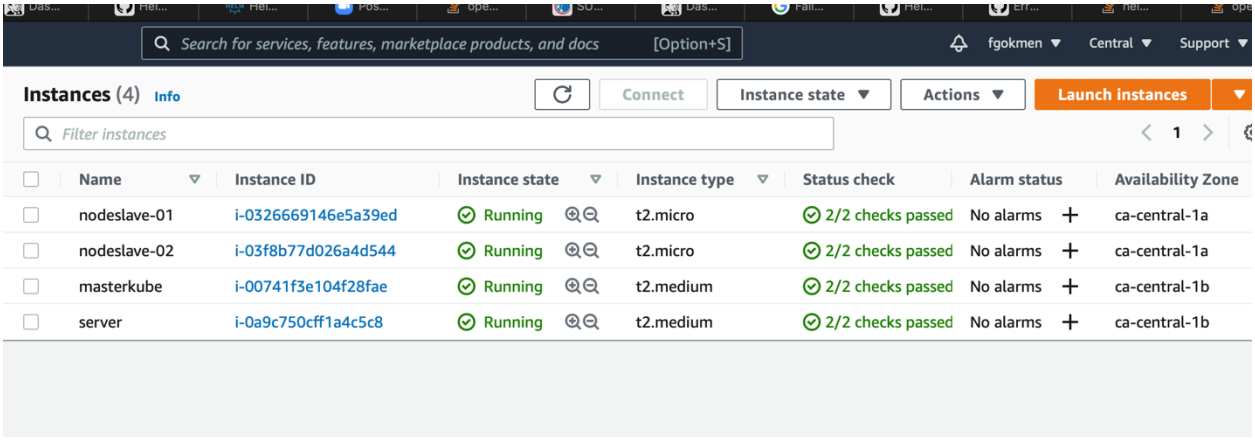
```
Creating /var/lib/jenkins/.helm
Creating /var/lib/jenkins/.helm/repository
Creating /var/lib/jenkins/.helm/repository/cache
Creating /var/lib/jenkins/.helm/repository/local
Creating /var/lib/jenkins/.helm/plugins
Creating /var/lib/jenkins/.helm/starters
Creating /var/lib/jenkins/.helm/cache/archive
Creating /var/lib/jenkins/.helm/repository/repositories.yaml
Adding stable repo with URL: https://kubernetes-charts.storage.googleapis.com

Error: error initializing: Looks like "https://kubernetes-charts.storage.googleapis.com" is not a valid chart repository or can
not be reached: Failed to fetch https://kubernetes-charts.storage.googleapis.com/index.yaml : 403 Forbidden
-bash-4.2$
-bash-4.2$
```

6- Communication successful from Jenkins server and Pods are running on master nodes where I create in Jenkins servers

```
NAME                                  READY   STATUS    RESTARTS   AGE
calico-kube-controllers-75dbcbbf8b-94qzm   1/1     Running   1          21h
calico-node-5flmm                     1/1     Running   1          21h
calico-node-95wvz                     1/1     Running   1          21h
calico-node-zvhcr                     1/1     Running   1          21h
coredns-5c98db65d4-nl464              1/1     Running   1          21h
coredns-5c98db65d4-ptjcp              1/1     Running   1          21h
etcd-masterkube                       1/1     Running   1          21h
kube-apiserver-masterkube             1/1     Running   1          21h
kube-controller-manager-masterkube    1/1     Running   1          21h
kube-proxy-kg4df                      1/1     Running   1          21h
kube-proxy-m92dp                      1/1     Running   1          21h
kube-proxy-xxvd8                      1/1     Running   1          21h
kube-scheduler-masterkube             1/1     Running   1          21h
tiller-deploy-7bf78cdbf7-gvz77        1/1     Running   0          46s
```

7- Here is my instance on AWS console.



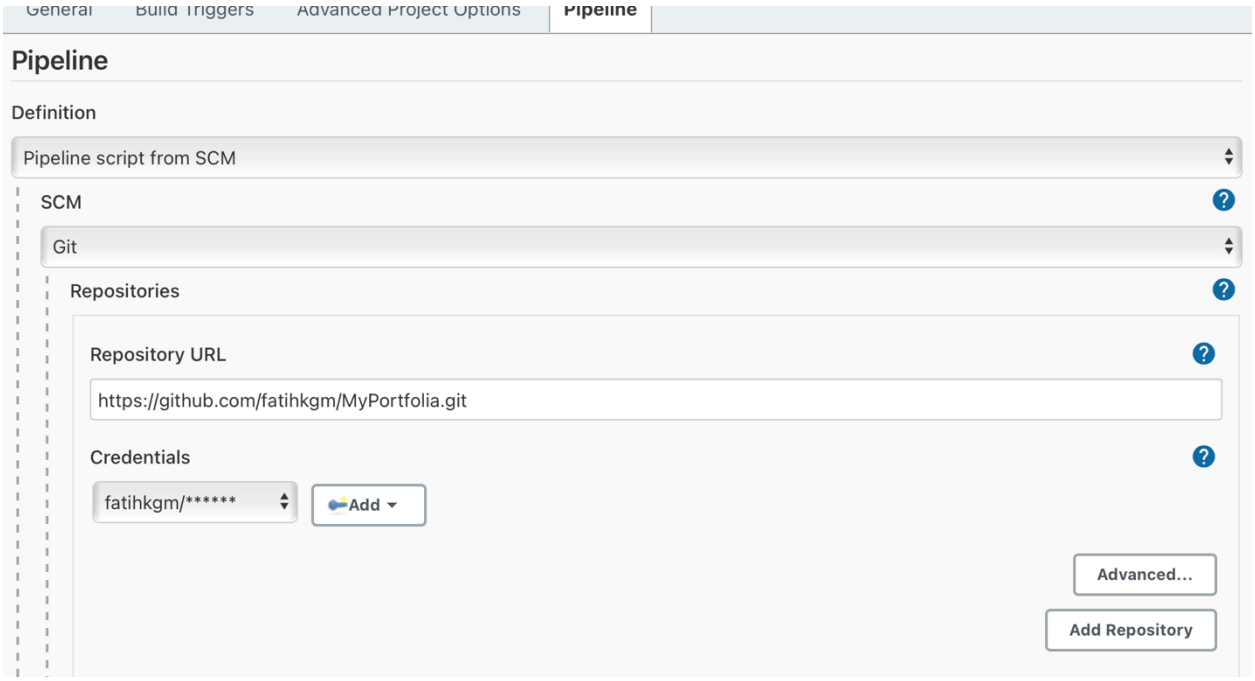
8-Helm version on Jenkins Client and Server

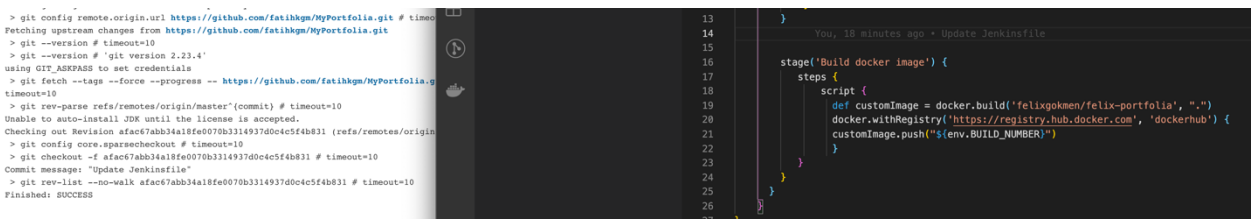
- Client is Jenkins's server
- Server is for Master server

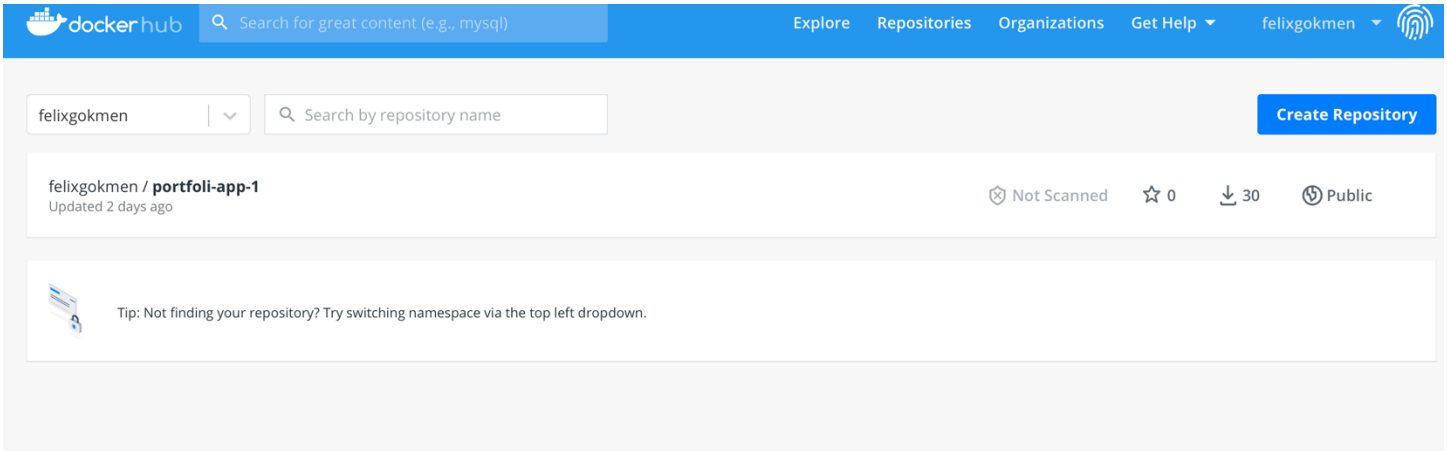
```
-bash-4.2$ helm version
Client: &version.Version{SemVer:"v2.14.1", GitCommit:"5270352a09c7e8b6e8c9593002a73535276507c0", GitTreeState:"clean"}
Server: &version.Version{SemVer:"v2.14.1", GitCommit:"5270352a09c7e8b6e8c9593002a73535276507c0", GitTreeState:"clean"}
```

9- Created Jenkinfile for continuous Integration of my existing portfolio web site.

- Docker credential entered and pushed by Jenkins pipline
- Git repostiry cloned to Jenkins pipline.







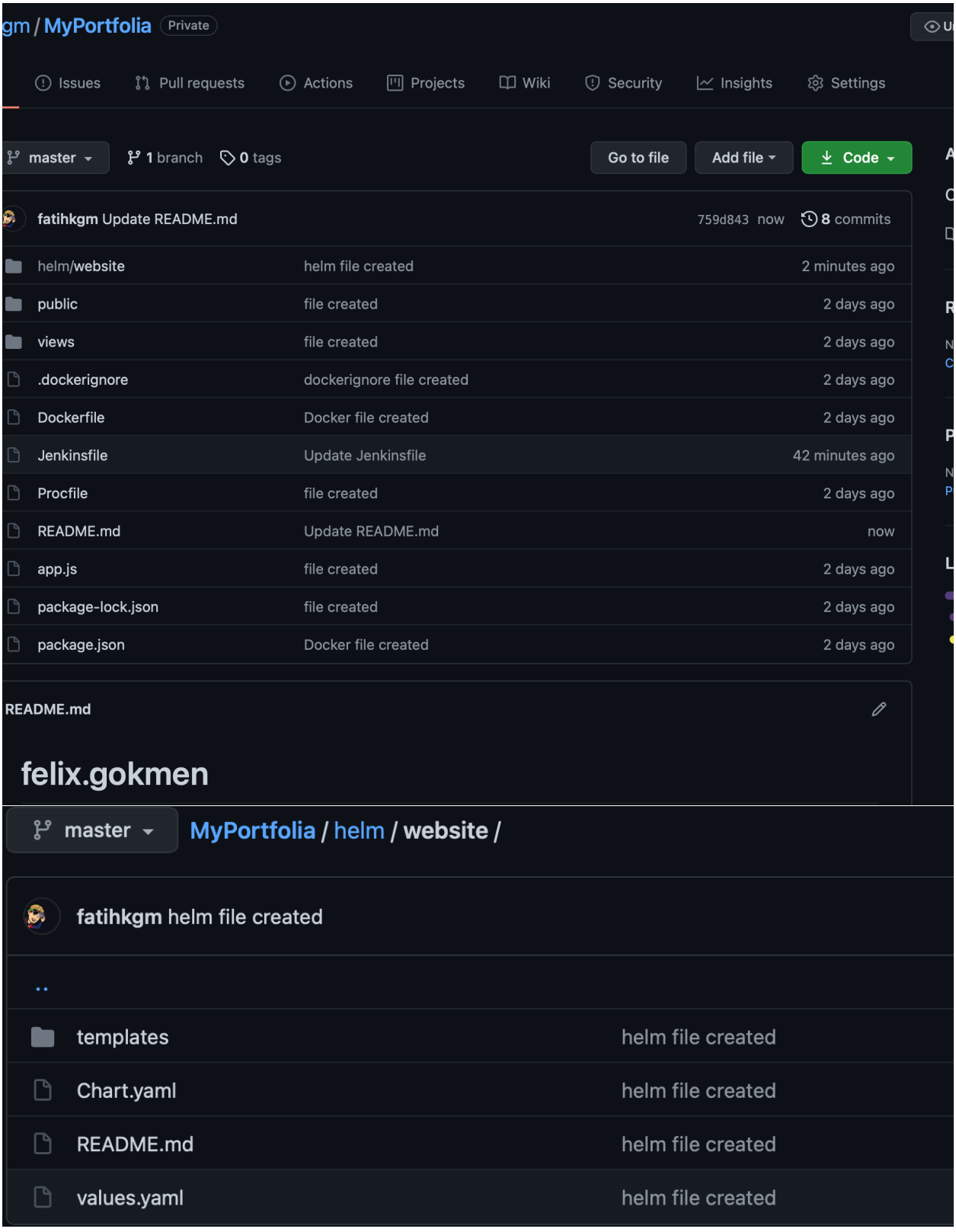
10-Mater server – created imagePullScret which I created docker image as private to pull.

- kubectl create secret docker dcoker-registry regcred --docker-server=hub.docker.com --docker-username=felixgokmen --docker-password=

```
[fgokmen@masterkube ~]$ kubectl create secret docker-registry regcred --docker-server=hub.docker.com --docker-username=felixgokmen --docker-password=
56?
secret/regcred created
[fgokmen@masterkube ~]$
```

```
[fgokmen@masterkube ~]$ kubectl get secret
NAME                  TYPE                                  DATA   AGE
default-token-759fk   kubernetes.io/service-account-token   3      24h
regcred               kubernetes.io/dockerconfigjson        1      50s
[fgokmen@masterkube ~]$ _
```

11-Deploying my website -NodejsAPP- on Kubernetes Cluster with Jenkinsfile by using HELM.



- Deployment screenshots

```yaml
  template:
    metadata:
      labels:
        app: website
    spec:
      containers:
      - name: website-container
        image: "{{ .Values.image.repository }}:{{ .Values.image.tag }}"
        imagePullPolicy: Always
        resources:
          limits:
            cpu: '1'
            memory: '1Gi'
          requests:
            cpu: '1'
            memory: '1Gi'
        ports:
        - containerPort: 8080
      imagePullSecrets:
      - name: regcred
```

12- There is now pods available now on master service and server instance

```
regcred               kubernetes.io/dockerconfigjson        1
[fgokmen@masterkube ~]$ kubectl get pods
No resources found.
[fgokmen@masterkube ~]$
```

```
[root@ip-172-31-5-148 ~]# su - jenkins
Last login: Sun May 23 07:27:35 UTC 2021 on pts/0
-bash-4.2$ helm list
-bash-4.2$
```

13- Pipeline created for pulling docker image and using helm chart in gitgub.

master mywebsite / Jenkinsfile Go to file

fatihkgm jenkins updated Latest commit 618ef37 6 minutes ago His

1 contributor

14 lines (14 sloc) | 366 Bytes Raw Blame

```
pipeline {
        agent any
    stages {
        stage('Build on k8 ') {
            steps {
                        sh 'pwd'
                        sh 'cp -R helm/* .'
                                sh 'ls -ltr'
                        sh 'pwd'
                        sh '/usr/local/bin/helm upgrade --install mywebsite website'
            }
        }
    }
}
```

- Screenshots value.yaml in github

master mywebsite / helm / website / values.yaml

fatihkgm update Latest commit f4acd01 1

1 contributor

64 lines (52 sloc) | 1.38 KB Ra

```
replicaCount: 1

image:
  repository: felixgokmen/portfoli-app-1
  tag: latest
  pullPolicy: IfNotPresent

imagePullSecrets: []
nameOverride: ""
fullnameOverride: ""

serviceAccount:
  # Specifies whether a service account should be created
  create: true
  # The name of the service account to use.
  # If not set and create is true, a name is generated using the fullname template
  name:

podSecurityContext: {}
  # fsGroup: 2000

securityContext: {}
  # capabilities:
  #   drop:
  #   - ALL
  # readOnlyRootFilesystem: true
  # runAsNonRoot: true
```

- Deployment file with using value.yaml

```
spec:
  containers:
  - name: website-container
    image: "{{ .Values.image.repository }}:{{ .Values.image.tag }}"
    imagePullPolicy: Always
    resources:
```

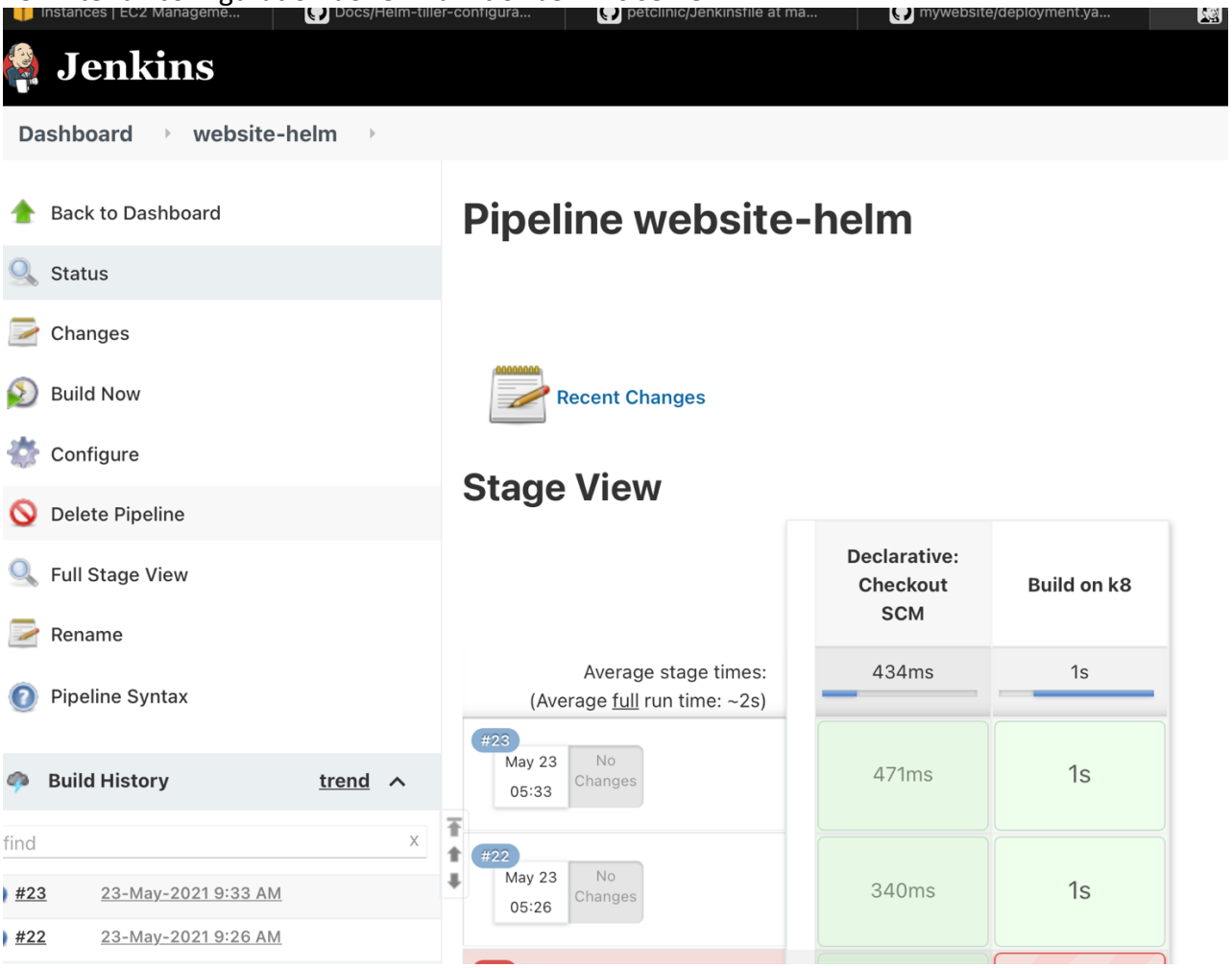
- Jenkins server helm list now shows

```
NAME            REVISION        UPDATED                         STATUS          CHART           APP VERSION
mywebsite       2               Sun May 23 09:34:02 2021        DEPLOYED        website-0.1.0   1.0
-bash-4.2$
```

- Master server

```
NAME                       READY   STATUS    RESTARTS   AGE
website-69c5968dfd-vh7zc   0/1     Pending   0          12m
[fgokmen@masterkube ~]$
```

13- After all configuration done - run it on Jenkins server



- Console Output is ;

```
LAST DEPLOYED: Sun May 23 09:34:02 2021
NAMESPACE: default
STATUS: DEPLOYED

RESOURCES:
==> v1/Pod(related)
NAME                       READY  STATUS   RESTARTS  AGE
website-69c5968dfd-vh7zc   0/1    Pending  0         7m17s

==> v1/Service
NAME     TYPE          CLUSTER-IP    EXTERNAL-IP  PORT(S)         AGE
website  LoadBalancer  10.98.124.8   <pending>    3000:30100/TCP  7m18s

==> v1/ServiceAccount
NAME       SECRETS  AGE
mywebsite  1        7m18s

==> v1beta1/Deployment
NAME     READY  UP-TO-DATE  AVAILABLE  AGE
website  0/1    1           0          7m18s

==> v2beta1/HorizontalPodAutoscaler
NAME             REFERENCE                       TARGETS         MINPODS  MAXPODS  REPLICAS  AGE
website-mem-hpa  Deployment/website-deployment   <unknown>/50%   1        5        0         7m18s


NOTES:
1. Get the application URL by running these commands:
  export POD_NAME=$(kubectl get pods --namespace default -l "app.kubernetes.io/name=website,app.kubernetes.
  echo "Visit http://127.0.0.1:8080 to use your application"
  kubectl port-forward $POD_NAME 8080:80

[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

- Helm list in Jenkins server

```
NAME            REVISION        UPDATED                         STATUS          CHART
mywebsite       5               Mon May 24 01:07:28 2021        DEPLOYED        website-0.1.0
```

- Pods in master server

```
NAME                          READY   STATUS    RESTARTS   AGE
website-7bdcbcb495-wt244      1/1     Running   0          18m
```

- Pods information

```
    Requests:
      cpu:        300m
      memory:     428Mi
    Environment:  <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-759fk (ro)
Conditions:
  Type              Status
  Initialized       True
  Ready             True
  ContainersReady   True
  PodScheduled      True
Volumes:
  default-token-759fk:
    Type:        Secret (a volume populated by a Secret)
    SecretName:  default-token-759fk
    Optional:    false
QoS Class:       Burstable
Node-Selectors:  <none>
Tolerations:     node.kubernetes.io/not-ready:NoExecute for 300s
                 node.kubernetes.io/unreachable:NoExecute for 300s
Events:
  Type    Reason     Age   From                  Message
  ----    ------     ----  ----                  -------
  Normal  Scheduled  19m   default-scheduler     Successfully assigned default/website-7bdcbcb495-wt244 to nodeslave-01
  Normal  Pulling    19m   kubelet, nodeslave-01  Pulling image "felixgokmen/portfoli-app-1:latest"
  Normal  Pulled     18m   kubelet, nodeslave-01  Successfully pulled image "felixgokmen/portfoli-app-1:latest"
  Normal  Created    18m   kubelet, nodeslave-01  Created container website-container
  Normal  Started    18m   kubelet, nodeslave-01  Started container website-container
```

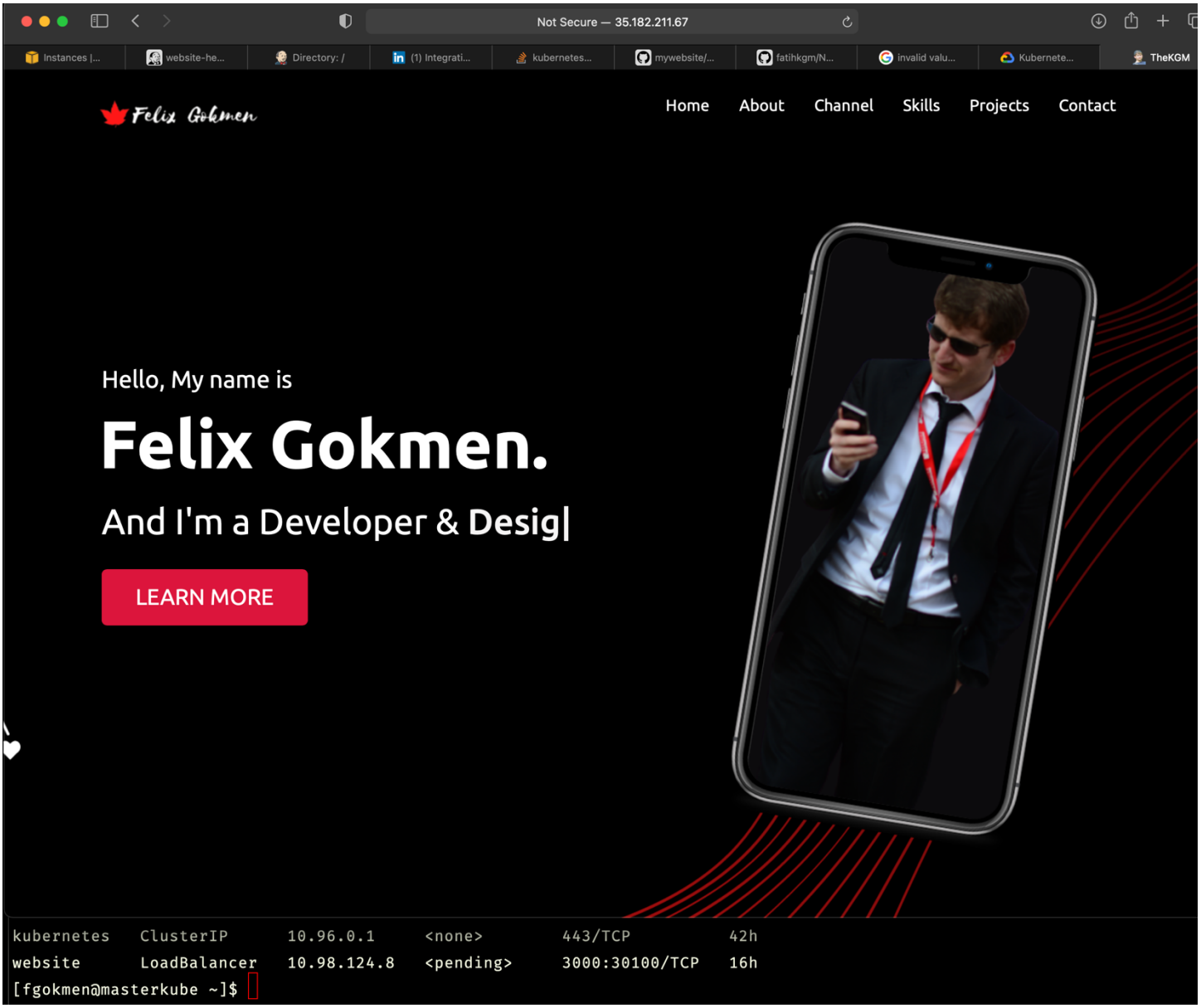
- Loadbalancer created

```
  Normal  Started    18m   kubelet, nodeslave-01  Started container website-con
[fgokmen@masterkube ~]$ kubectl get svc
NAME         TYPE           CLUSTER-IP    EXTERNAL-IP   PORT(S)          AGE
kubernetes   ClusterIP      10.96.0.1     <none>        443/TCP          42h
website      LoadBalancer   10.98.124.8   <pending>     3000:30100/TCP   16h
[fgokmen@masterkube ~]$
```

- Checking with my nodeslave server public ip that created webapp with port number I assigned. http://35.183.198.151:30100

Name	Instance ID	Instance state	Instance type	Status check	Ala
nodeslave-01	i-0326669146e5a39ed	Running	t2.micro	2/2 checks passed	No
nodeslave-02	i-03f8b77d026a4d544	Running	t2.micro	2/2 checks passed	No
masterkube	i-00741f3e104f28fae	Running	t2.medium	2/2 checks passed	No
server	i-0a9c750cff1a4c5c8	Running	t2.medium	2/2 checks passed	No

14- ☺ The website is reachable now.

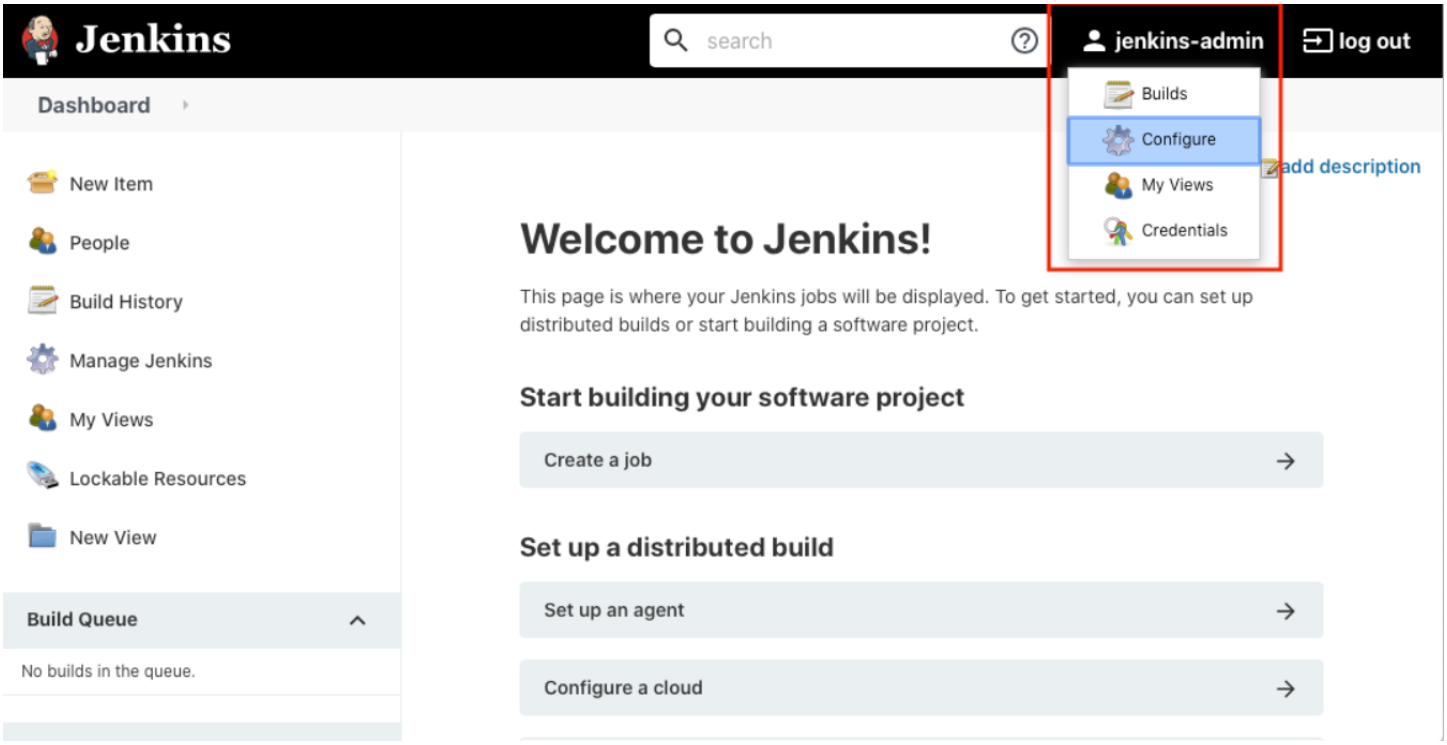

Not Secure — 35.182.211.67
Instances |...
website-he...
Directory: /
(1) Integrati...
kubernetes...
mywebsite/...
fatihkgm/N...
invalid valu...
Kubernete...
TheKGM
Felix Gokmen
Home
About
Channel
Skills
Projects
Contact
Hello, My name is
Felix Gokmen.
And I'm a Developer & Desig
LEARN MORE
kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 42h
website LoadBalancer 10.98.124.8 <pending> 3000:30100/TCP 16h
[fgokmen@masterkube ~]$

15- Cheking my Jenkins server for files

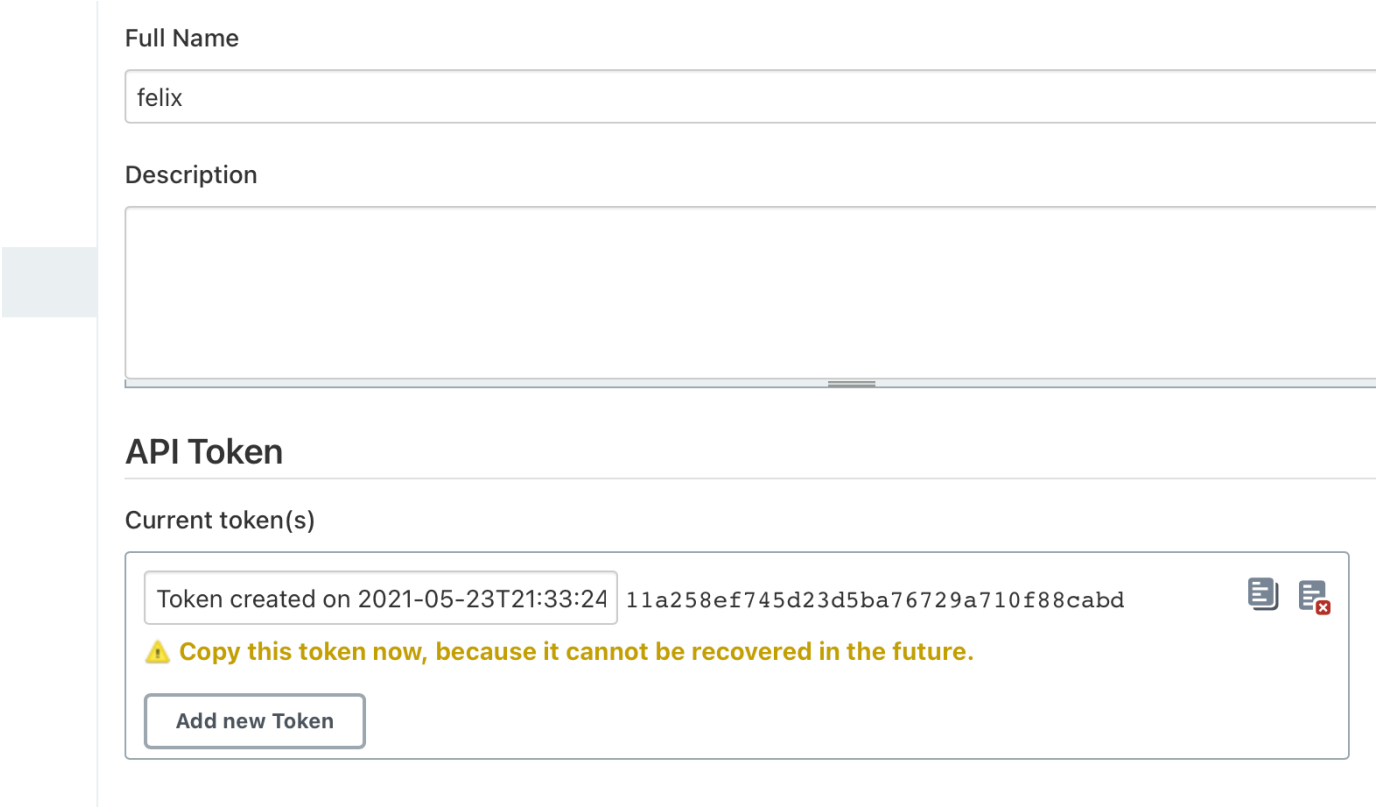
```
[root@ip-172-31-5-148 ~]# cd /var/lib/jenkins/workspace/website-helm
[root@ip-172-31-5-148 website-helm]# ls -lart
total 80
-rw-r--r--  1 jenkins jenkins   111 May 23 08:31 README.md
-rw-r--r--  1 jenkins jenkins    17 May 23 08:31 Procfile
drwxr-xr-x  3 jenkins jenkins    21 May 23 08:31 helm
-rw-r--r--  1 jenkins jenkins    26 May 23 08:31 .dockerignore
-rw-r--r--  1 jenkins jenkins   123 May 23 08:31 Dockerfile
-rw-r--r--  1 jenkins jenkins  3411 May 23 08:31 app.js
-rw-r--r--  1 jenkins jenkins 51993 May 23 08:31 package-lock.json
-rw-r--r--  1 jenkins jenkins   845 May 23 08:31 package.json
drwxr-xr-x  5 jenkins jenkins   100 May 23 08:31 public
drwxr-xr-x  3 jenkins jenkins   216 May 23 08:31 views
drwxr-xr-x  3 jenkins jenkins    77 May 23 08:31 website
drwxr-xr-x 10 jenkins jenkins   186 May 23 09:10 ..
-rw-r--r--  1 jenkins jenkins   366 May 23 09:23 Jenkinsfile
drwxr-xr-x  7 jenkins jenkins   222 May 23 09:23 .
drwxr-xr-x  8 jenkins jenkins   162 May 24 01:07 .git
```

👊 Lets build Integrating GitHub Webhooks with Jenkins to automate unit and integration test after GitHub events for CI/CD

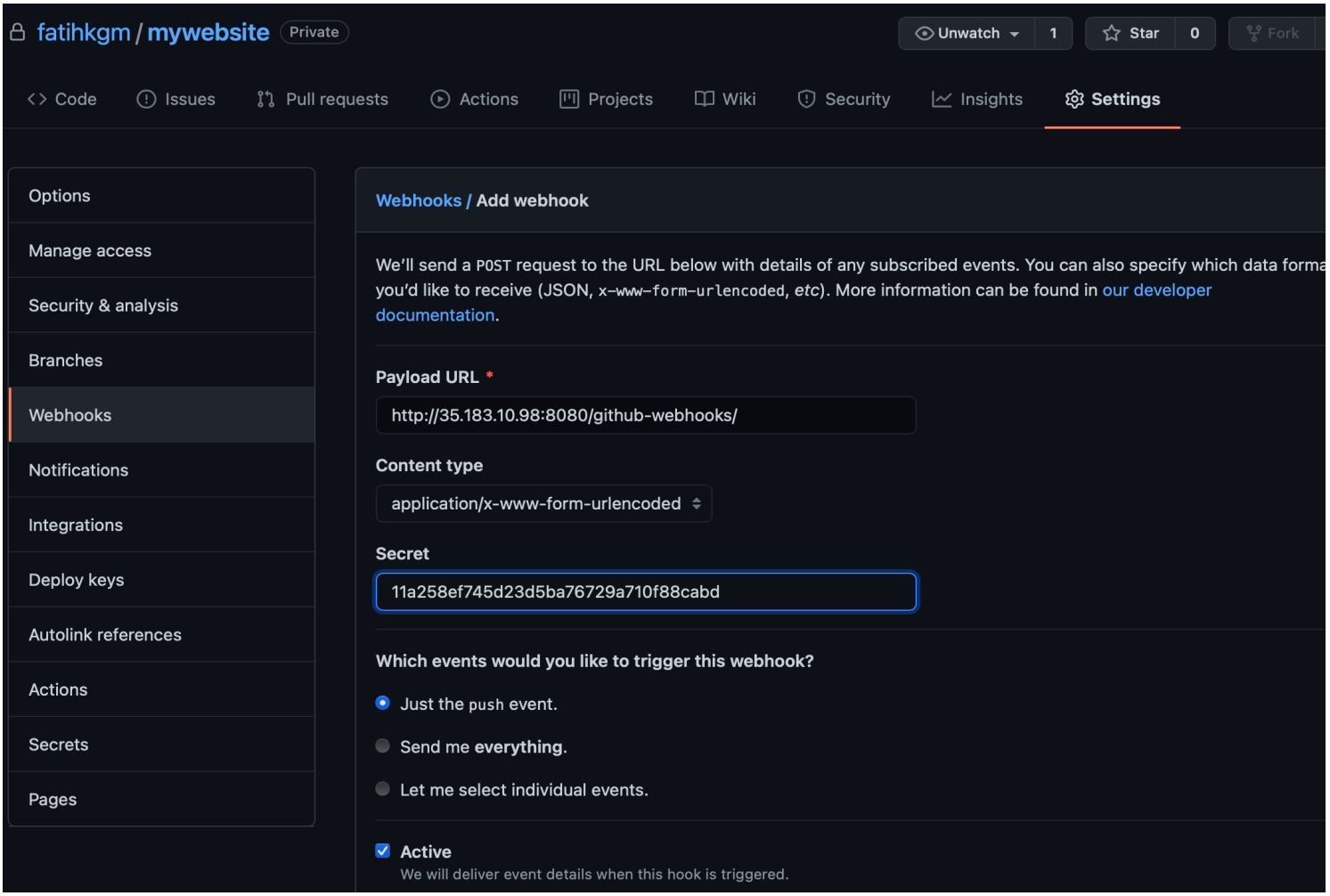
👉 To connect my Jenkins server with my Webhook, i will first need to create an API Token to authenticate from the version control provider (Github in this case). To do so, we need to click on our account and then on the configure option.



👉 Click on Add new Token, write a name for it and then press Generate. You need to store this Token in a safe place because there is no way to recover it in the future unless creating a new one.

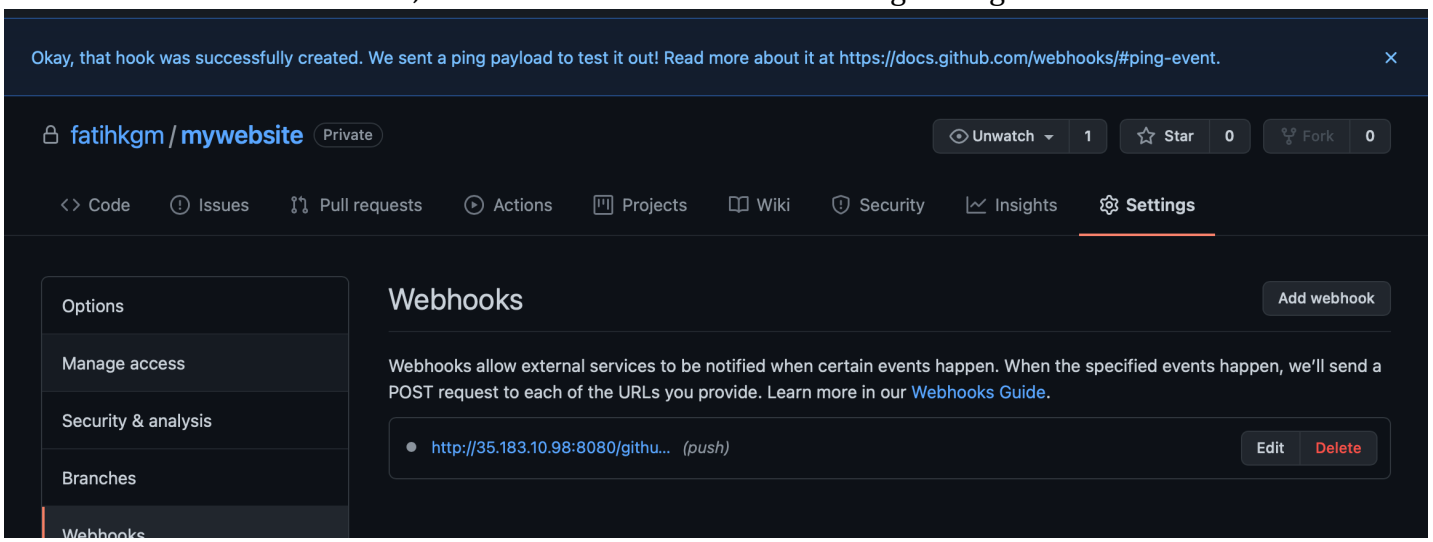


👉 To create GitHub Webhook, i need to open our repository and click on settings. Go to github repository and click setting > webhooks > add webhoks .Here is the configuration below picture.



👉 Inside the Payload URL text box, you need to write the direction to you Jenkins Server, plus: /github-webhook/. If you created a ngrok account, it'd be obtained at the end of section one. The payload must have both / to work, otherwise it'll throw some error. Inside the Secret box, i'll paste the Jenkins API Token which you can see above picture. Finally, click on: Let me select individual elements to choose the desired webhook events for our needs to trigger Jenkins.

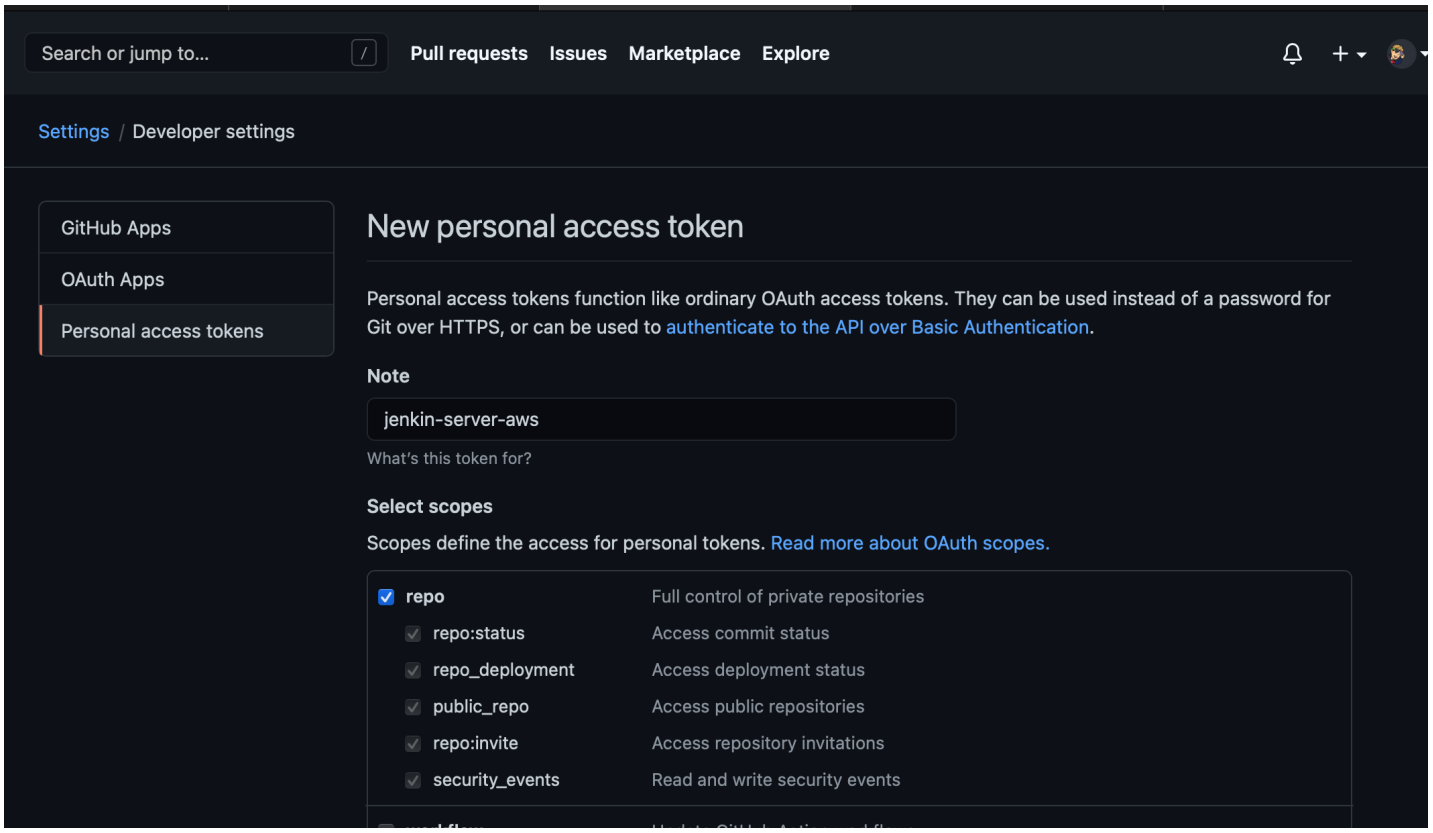
☑Once created the webhook, it'll look like the following image.



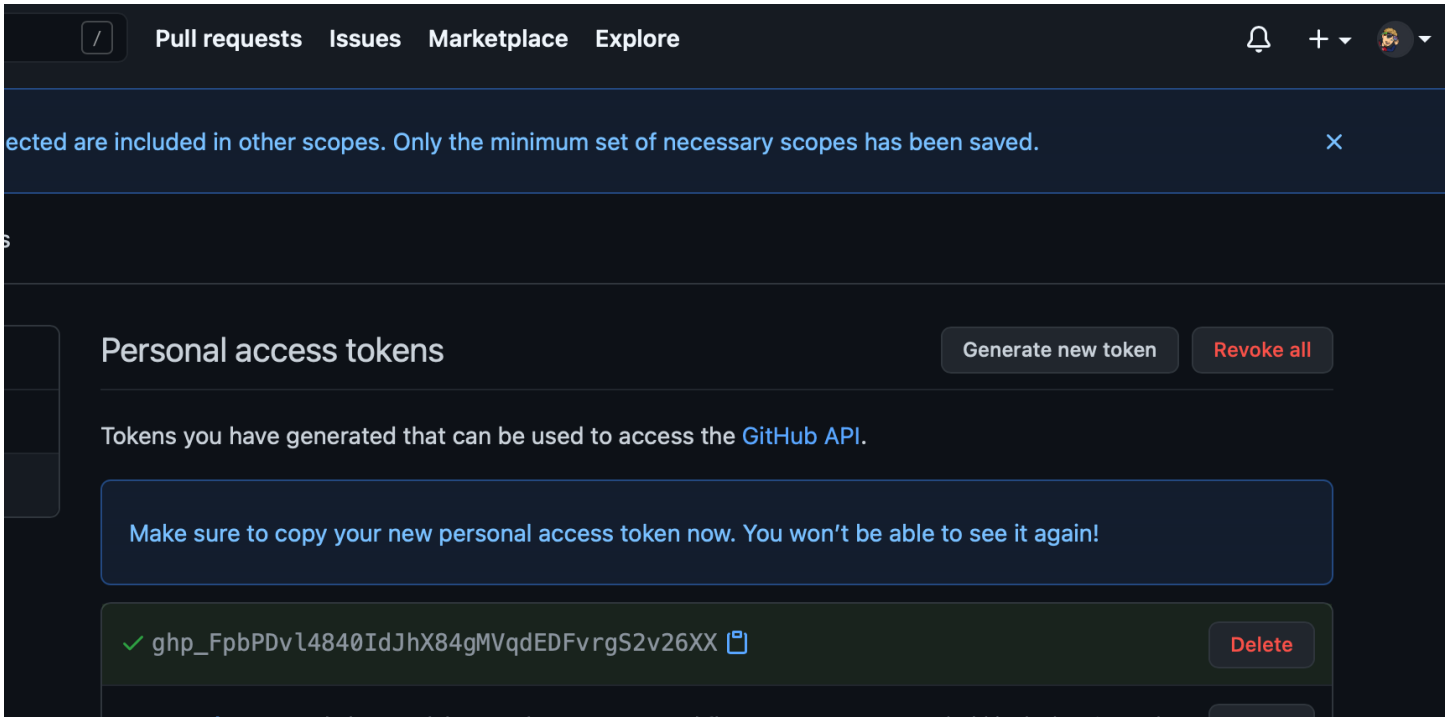
👉 Since early 2021 GitHub restricts API user authentication to work only with a token or private ssh key. There is no longer possible to authenticate using you user and password, and such is the case working with Jenkins.

To create a token to be authenticated from Jenkins, go to your GitHub profile, then click on Settings, Developer Settings, Personal access tokens -- see picture below.

🟡 I give it a name ("Jenkin-server-aws ") plus checking the two boxes: repo and user, finally click on Generate Token.

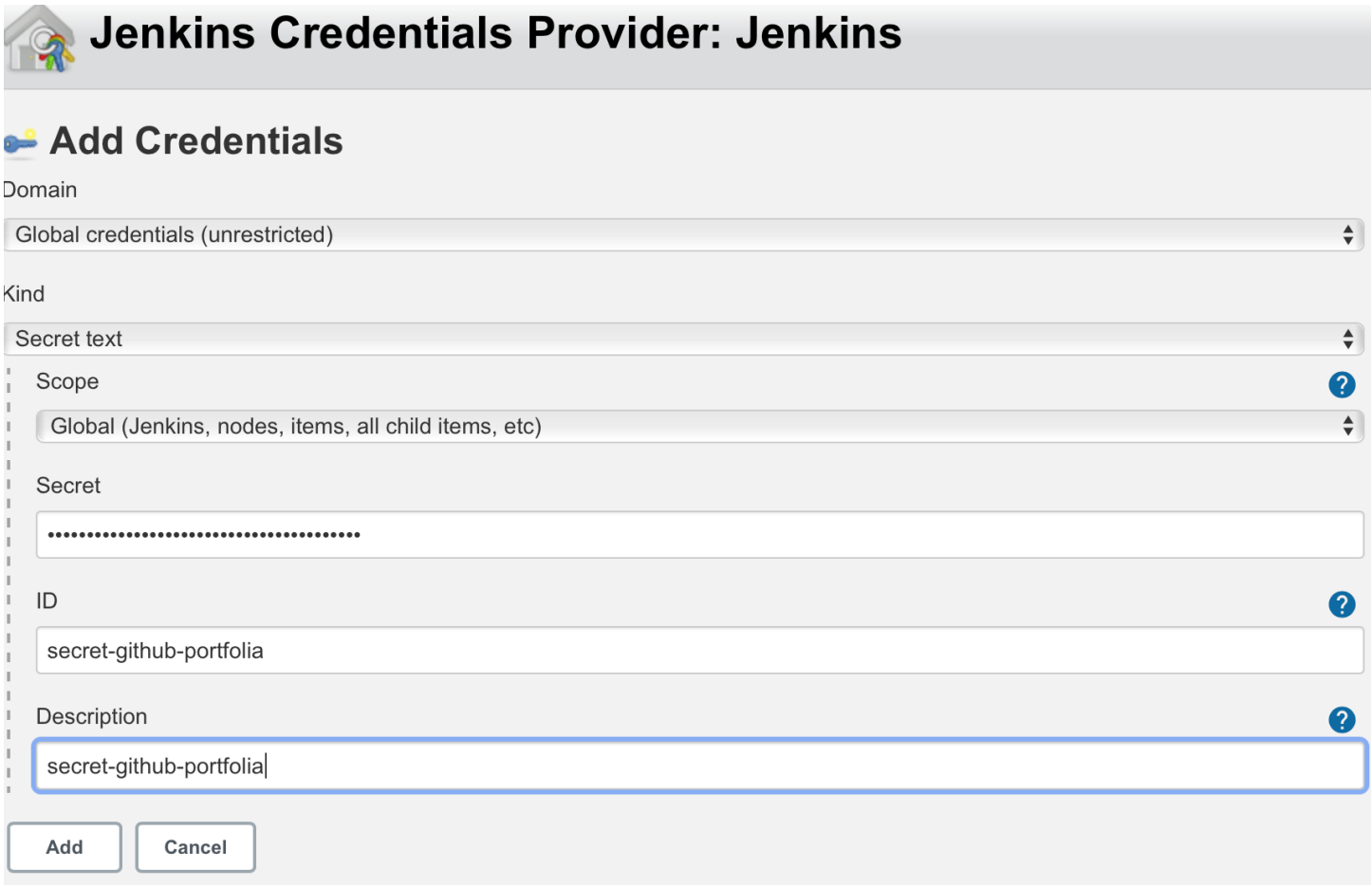


👉 Once the token has been generated, it'll look like the following. Again, these tokens are meant to be kept in a safe place, there is no way to recover them once lost unless creating a new one.

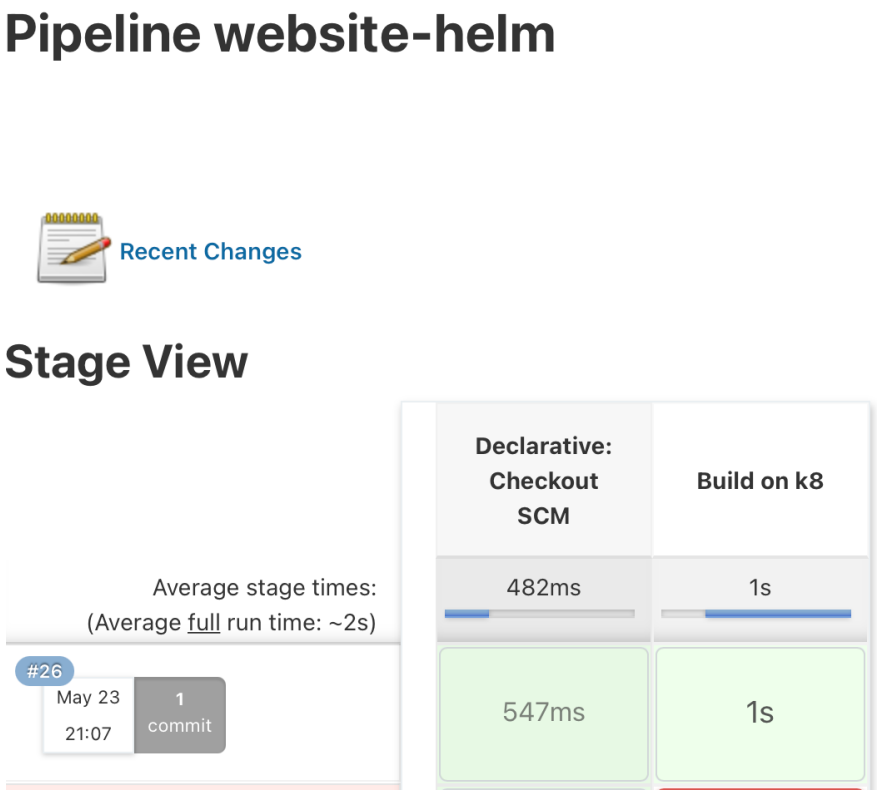


👉 You need first to have the GitHub plugin installed on Jenkins; if you installed all the recommended plugins at Jenkins installation, it should be there) click on Add GitHub Server, GitHub Server

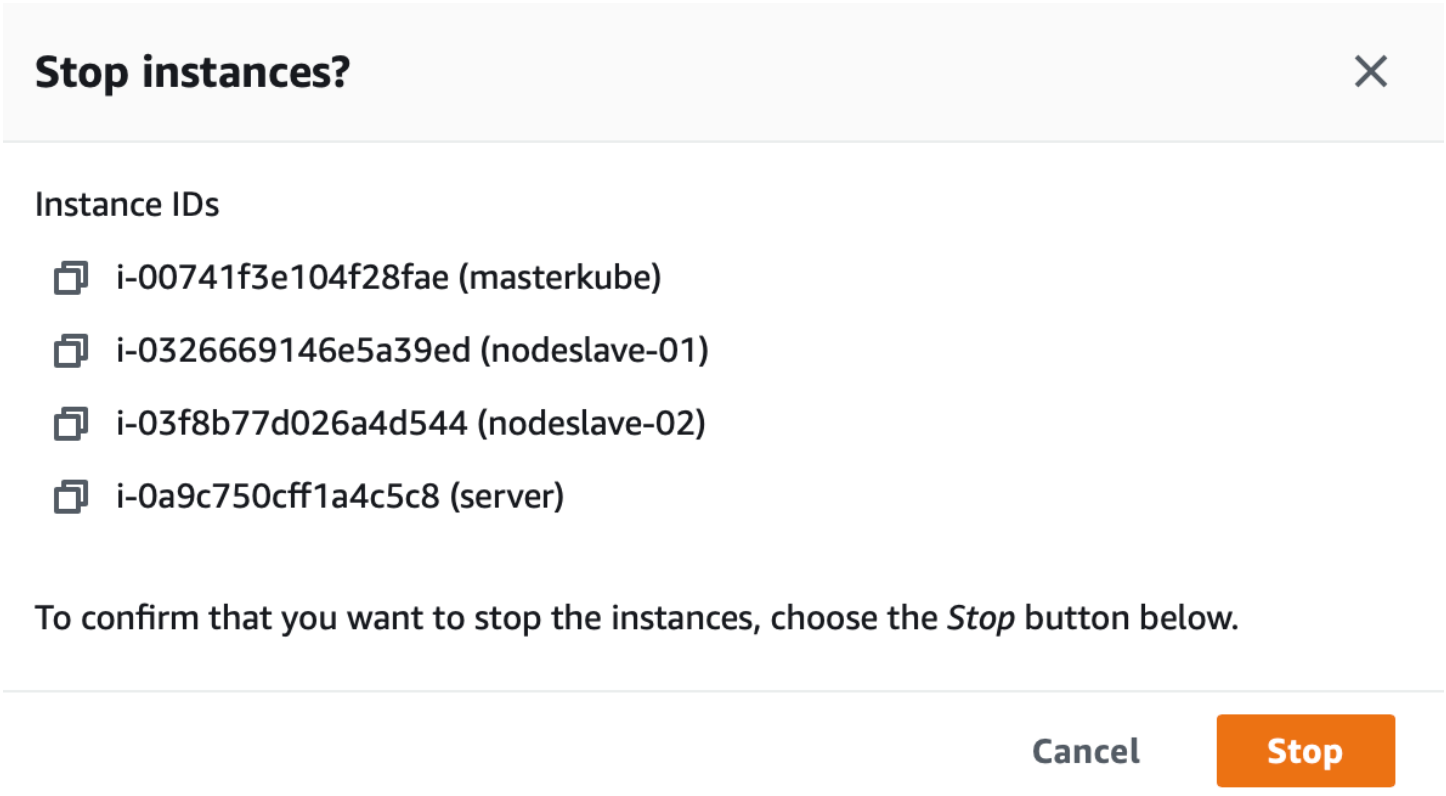
✔Inside the credentials section click on add, Jenkins.



👉 Pushing any commit to our GitHub repository and triggering the build on Jenkins ☺ easy-peasy



👉 Next step is shot down your EC2 instances.🤣



Note: The AWS Free Tier provides to explore and try out AWS services free of charge up to specified limits for each service for 1 year. Since I use free tier ,it charged me almost 5$ .It depends of what service you are using ✔I used my instance almost 2days.