Setup Bitbucket Integration with Agro CD for Kubernetes cluster

ArgoCD

Argo CD is **a Kubernetes controller, responsible for continuously monitoring all running applications and comparing their live state to the desired state specified in the Git repository**. It identifies deployed applications with a live state that deviates from the desired state as OutOfSync.

Installing Argocd on Kubernetes:

We can go to this website for installing references.
https://argo-cd.readthedocs.io/en/stable/getting_started/

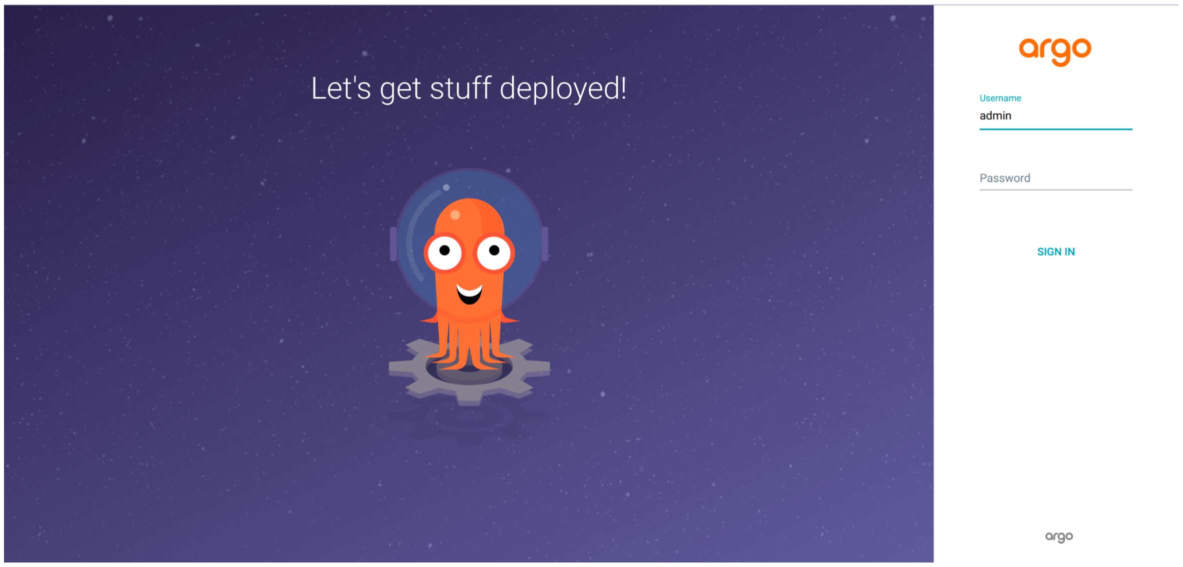
To install argo cd we will using these commands:

```
$kubectl create namespace argocd
$kubectl apply -n argocd -f
https://raw.githubusercontent.com/argoproj/argo-cd/stable/manifests/i
nstall.yaml
```

We will make an argocd-service NodePort .
`$kubectl edit svc argocd-server -n argocd`
Open the following port in the security group inbound.

Open UI and provide default credentials .

User is “admin”
We can get password using this command:
`$ kubectl -n argocd get secret argocd-initial-admin-secret -o jsonpath="{.data.password}" | base64 -d; echo`



Bitbucket

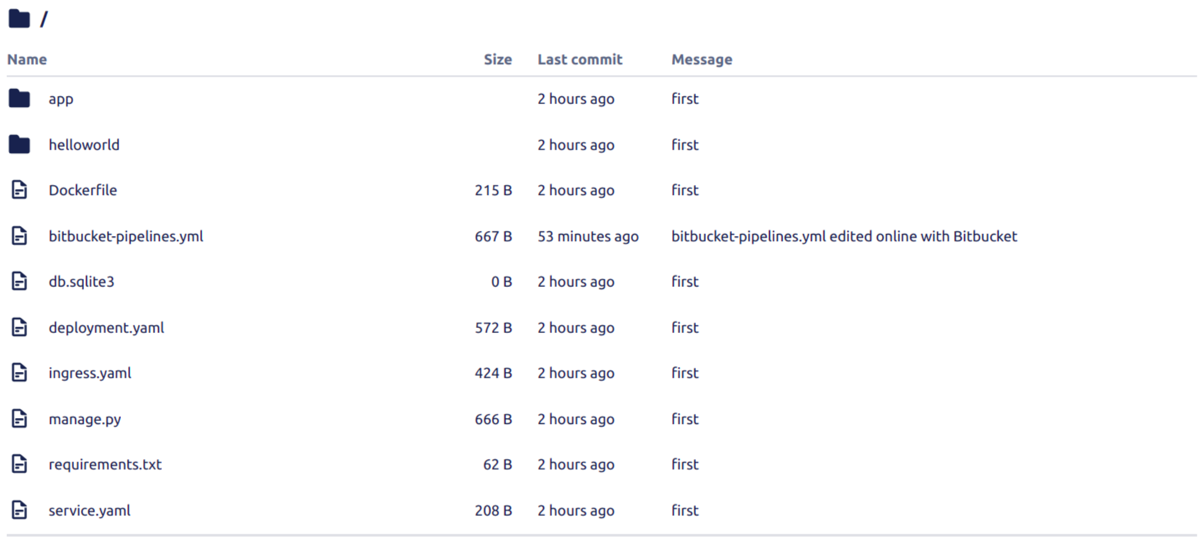
BitBucket is our Git repository management solution designed for professional teams. It gives you a central place to manage git repositories, collaborate on your source code and guide you through the development flow.

Bitbucket-Pipeline

Bitbucket Pipelines is **an integrated CI/CD service built into Bitbucket**. It allows you to automatically build, test, and even deploy your code based on a configuration file in your repository. Essentially, we create containers in the cloud for you.

Creating a Bitbucket pipeline for CI

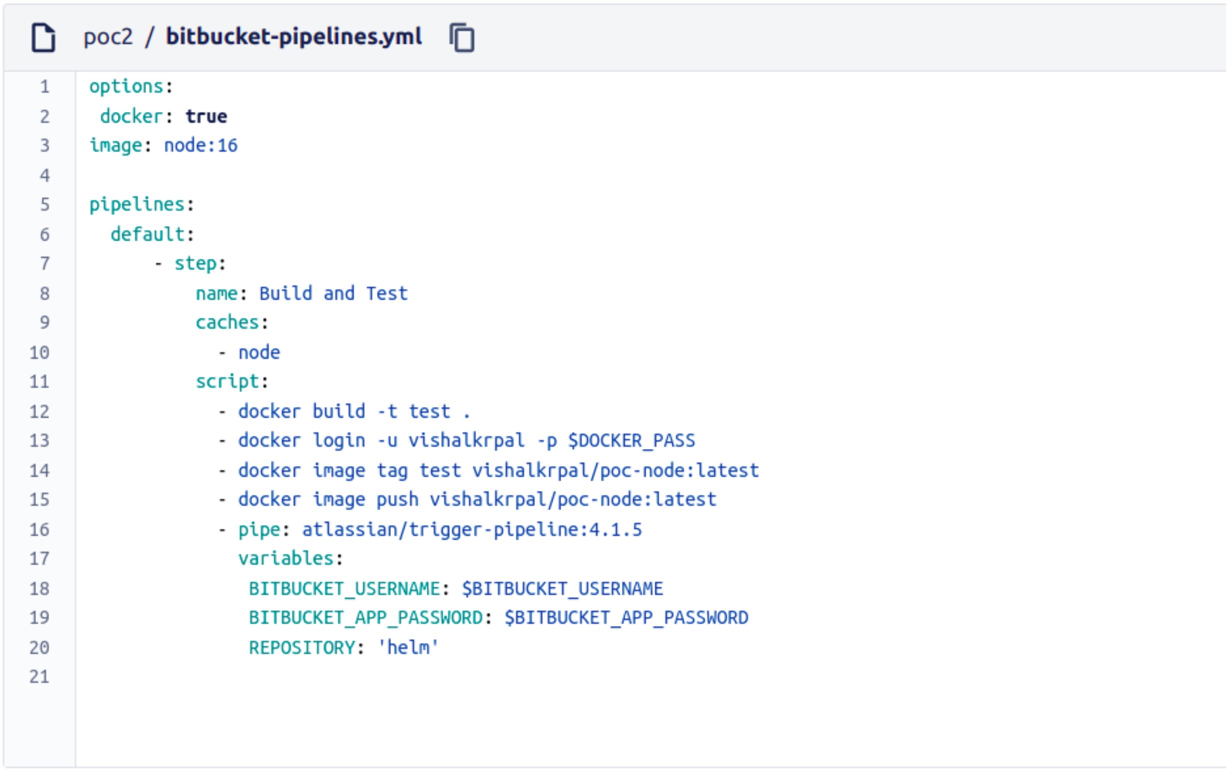
Along with our code we will push bitbucket-pipelines.yml file to out code repository.



/

Name	Size	Last commit	Message
app		2 hours ago	first
helloworld		2 hours ago	first
Dockerfile	215 B	2 hours ago	first
bitbucket-pipelines.yml	667 B	53 minutes ago	bitbucket-pipelines.yml edited online with Bitbucket
db.sqlite3	0 B	2 hours ago	first
deployment.yaml	572 B	2 hours ago	first
ingress.yaml	424 B	2 hours ago	first
manage.py	666 B	2 hours ago	first
requirements.txt	62 B	2 hours ago	first
service.yaml	208 B	2 hours ago	first

We will write a pipeline for the CI : steps we will be doing are **building a docker image** and **pushing image to DockerHub** and **triggering the Helm Pipeline** So that ArgoCD can deploy the latest image .



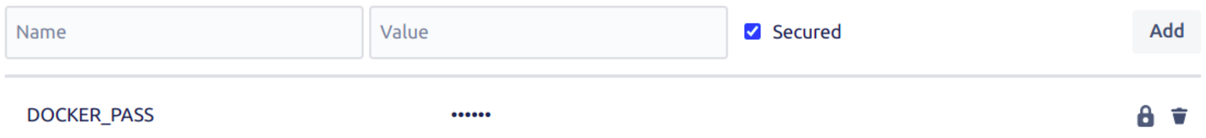
poc2 / **bitbucket-pipelines.yml**

```yaml
options:
 docker: true
image: node:16

pipelines:
  default:
      - step:
          name: Build and Test
          caches:
            - node
          script:
            - docker build -t test .
            - docker login -u vishalkrpal -p $DOCKER_PASS
            - docker image tag test vishalkrpal/poc-node:latest
            - docker image push vishalkrpal/poc-node:latest
            - pipe: atlassian/trigger-pipeline:4.1.5
              variables:
               BITBUCKET_USERNAME: $BITBUCKET_USERNAME
               BITBUCKET_APP_PASSWORD: $BITBUCKET_APP_PASSWORD
               REPOSITORY: 'helm'

```

We can add pipeline variables to use them in the pipeline .
Go to Repository settings –> pipelines -> Repository Variables .

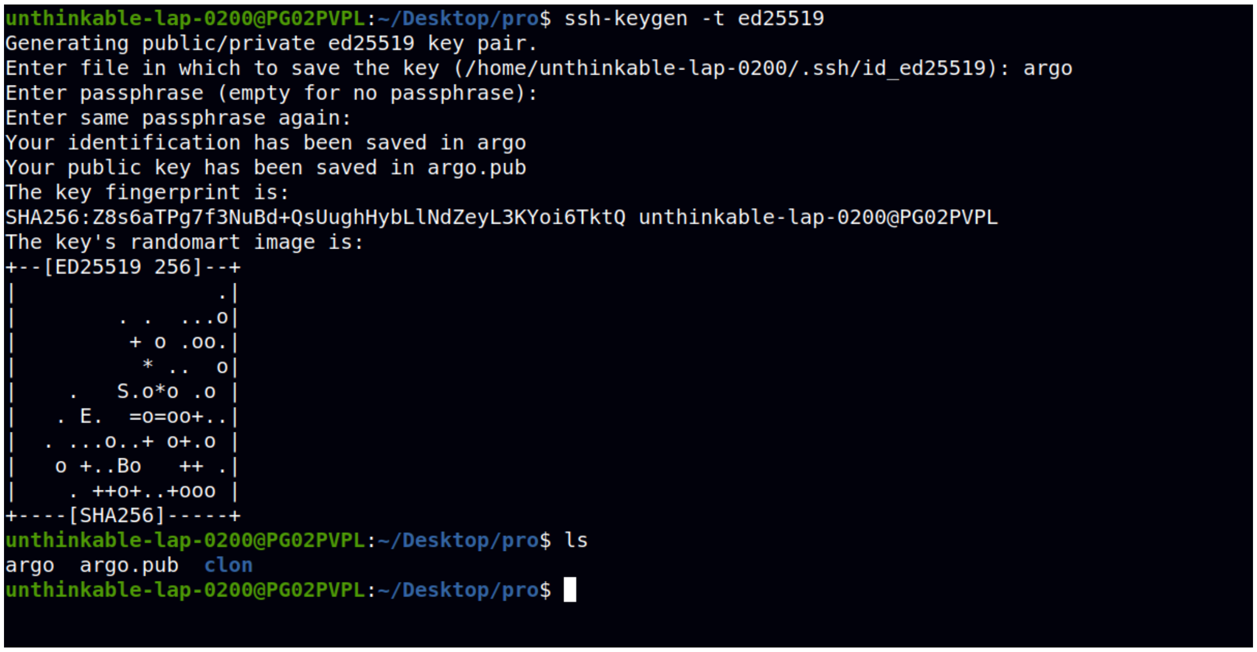


Name	Value	☑ Secured	Add
DOCKER_PASS	••••••		

sudo mv prometheus.yml /etc/prometheus/prometheus.yml

Integrating Bitbucket CI with CD in Argo

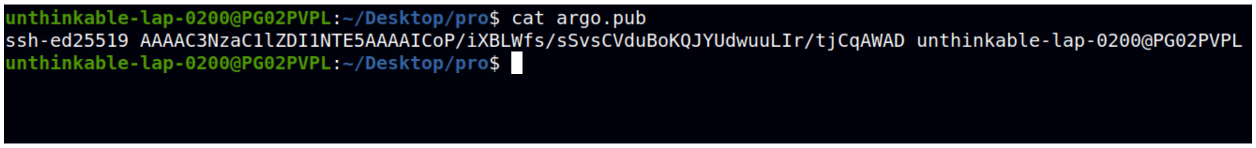
In the terminal we will generate a key pair using `ssh-keygen -t ed25519` and give a file name.



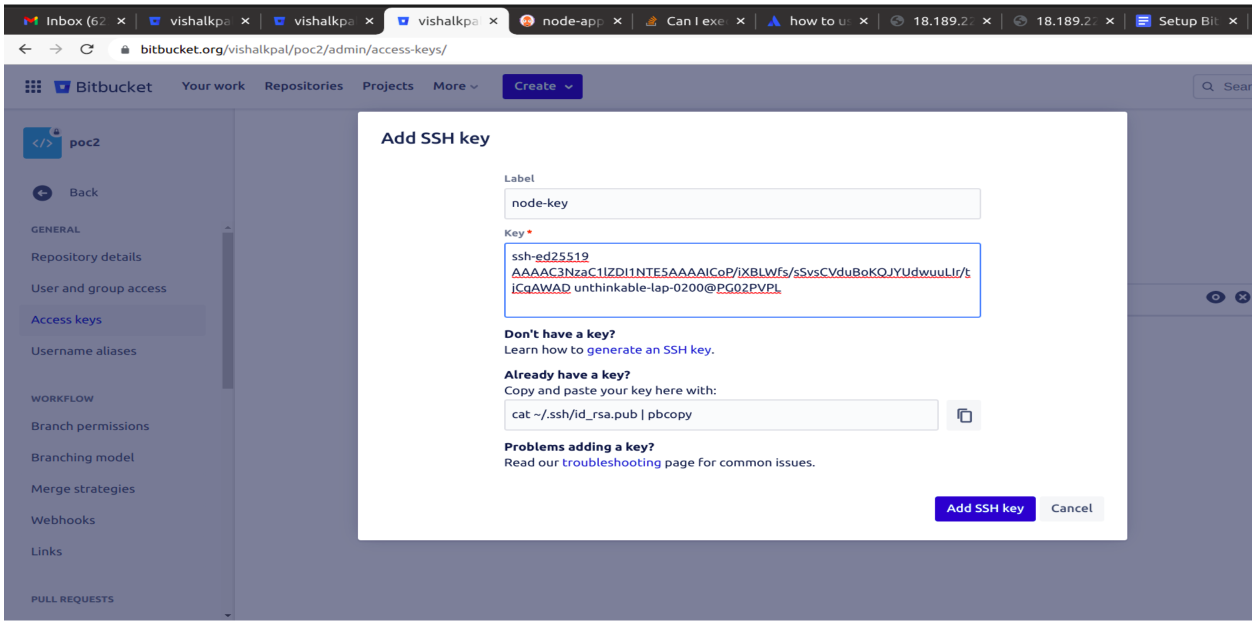
```
unthinkable-lap-0200@PG02PVPL:~/Desktop/pro$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/unthinkable-lap-0200/.ssh/id_ed25519): argo
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in argo
Your public key has been saved in argo.pub
The key fingerprint is:
SHA256:Z8s6aTPg7f3NuBd+QsUughHybLlNdZeyL3KYoi6TktQ unthinkable-lap-0200@PG02PVPL
The key's randomart image is:
+--[ED25519 256]--+
|                .|
|        . .  ...o|
|         + o .oo.|
|          * ..  o|
|     .   S.o*o .o |
|    . E.  =o=oo+..|
|   . ...o..+ o+.o |
|    o +..Bo   ++ .|
|     . ++o+..+ooo |
+----[SHA256]-----+
unthinkable-lap-0200@PG02PVPL:~/Desktop/pro$ ls
argo  argo.pub  clon
unthinkable-lap-0200@PG02PVPL:~/Desktop/pro$ 
```

Now we will cat the public key and paste it to access key of the Repository.
For access key go to Repository settings → access keys

Click add access key and paste the key here



```
unthinkable-lap-0200@PG02PVPL:~/Desktop/pro$ cat argo.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAICoP/iXBLWfs/sSvsCVduBoKQJYUdwuuLIr/tjCqAWAD unthinkable-lap-0200@PG02PVPL
unthinkable-lap-0200@PG02PVPL:~/Desktop/pro$ 
```



bitbucket.org/vishalkpal/poc2/admin/access-keys/

Add SSH key

Label: node-key

Key *

```
ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAICoP/iXBLWfs/sSvsCVduBoKQJYUdwuuLIr/t
jCqAWAD unthinkable-lap-0200@PG02PVPL
```

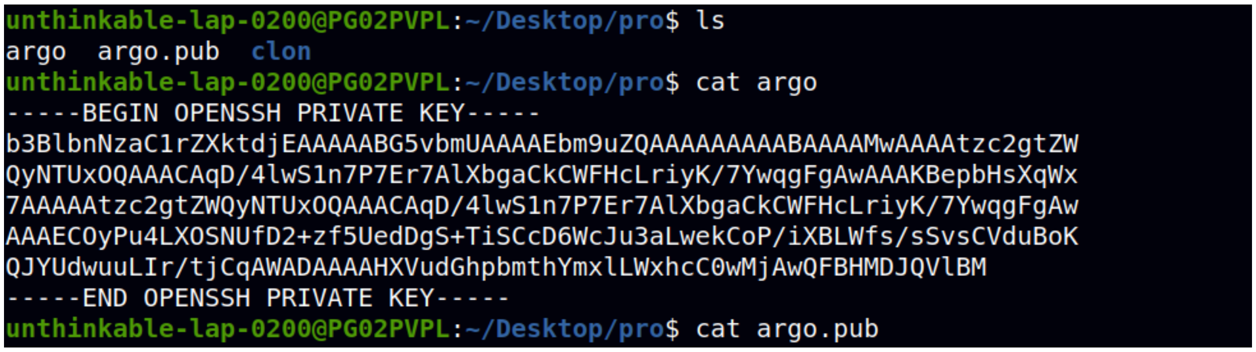
Don't have a key?
Learn how to generate an SSH key.

Already have a key?
Copy and paste your key here with:

```
cat ~/.ssh/id_rsa.pub | pbcopy
```

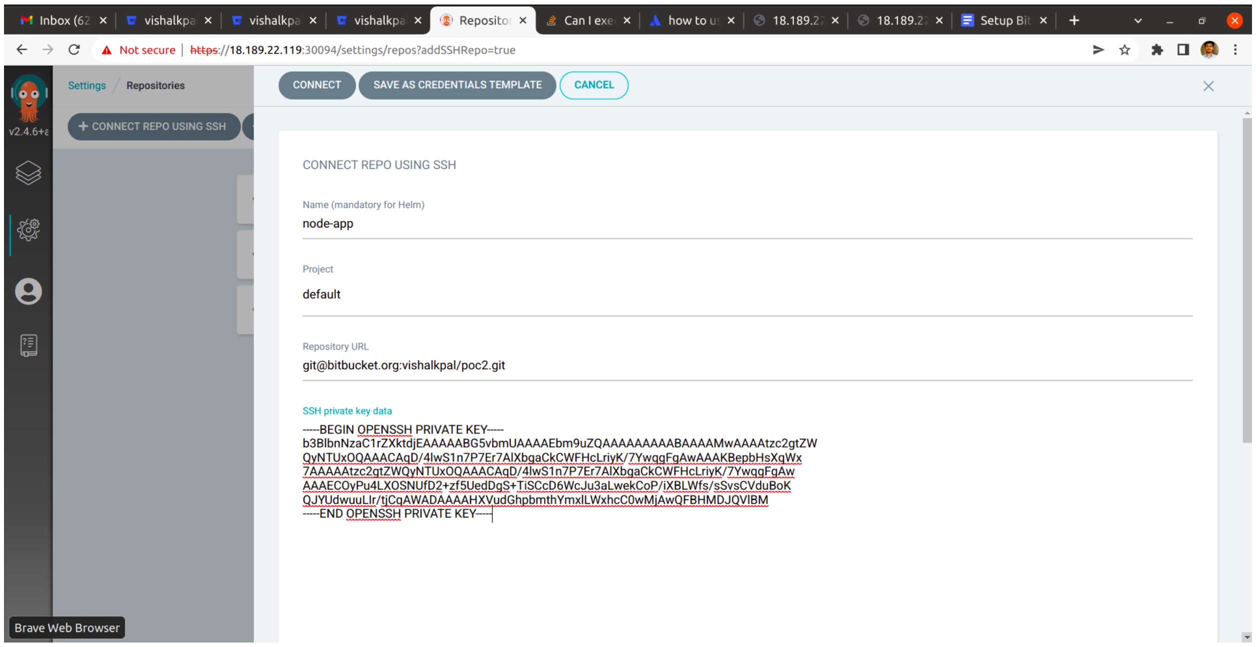
Problems adding a key?
Read our troubleshooting page for common issues.

Now we will add the private key to argocd



```
unthinkable-lap-0200@PG02PVPL:~/Desktop/pro$ ls
argo  argo.pub  clon
unthinkable-lap-0200@PG02PVPL:~/Desktop/pro$ cat argo
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAAAMwAAAAtzc2gtZW
QyNTUxOQAAACAqD/4lwS1n7P7Er7AlXbgaCkCWFHcLriyK/7YwqgFgAwAAAKBepbHsXqWx
7AAAAAtzc2gtZWQyNTUxOQAAACAqD/4lwS1n7P7Er7AlXbgaCkCWFHcLriyK/7YwqgFgAw
AAAECOyPu4LXOSNUfD2+zf5UedDgS+TiSCcD6WcJu3aLwekCoP/iXBLWfs/sSvsCVduBoK
QJYUdwuuLIr/tjCqAWADAAAAHXVudGhpbmthYmxlLWxhcC0wMjAwQFBHMDJQVlBM
-----END OPENSSH PRIVATE KEY-----
unthinkable-lap-0200@PG02PVPL:~/Desktop/pro$ cat argo.pub
```

Inside argocd go to Repository → +connect repo using SSH key
Paste repo ssh url and paste private key here


Not secure | https://18.189.22.119:30094/settings/repos?addSSHRepo=true
Settings / Repositories
CONNECT REPO USING SSH
CONNECT
SAVE AS CREDENTIALS TEMPLATE
CANCEL
v2.4.6+a
Brave Web Browser

CONNECT REPO USING SSH

Name (mandatory for Helm): node-app

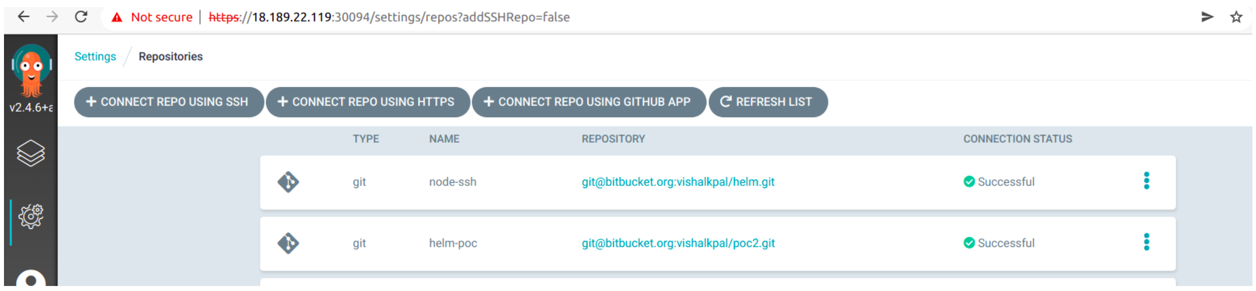
Project: default

Repository URL: git@bitbucket.org:vishalkpal/poc2.git

SSH private key data

```
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAAAMwAAAAtzc2gtZW
QyNTUxOQAAACAqD/4lwS1n7P7Er7AlXbgaCkCWFHcLriyK/7YwqgFgAwAAAKBepbHsXqWx
7AAAAAtzc2gtZWQyNTUxOQAAACAqD/4lwS1n7P7Er7AlXbgaCkCWFHcLriyK/7YwqgFgAw
AAAECOyPu4LXOSNUfD2+zf5UedDgS+TiSCcD6WcJu3aLwekCoP/iXBLWfs/sSvsCVduBoK
QJYUdwuuLIr/tjCqAWADAAAAHXVudGhpbmthYmxlLWxhcC0wMjAwQFBHMDJQVlBM
-----END OPENSSH PRIVATE KEY-----
```

Click connect.
This will connect your git repo with argoCD and give successful status.


Not secure | https://18.189.22.119:30094/settings/repos?addSSHRepo=false
Settings / Repositories
v2.4.6+a
CONNECT REPO USING SSH
CONNECT REPO USING HTTPS
CONNECT REPO USING GITHUB APP
REFRESH LIST

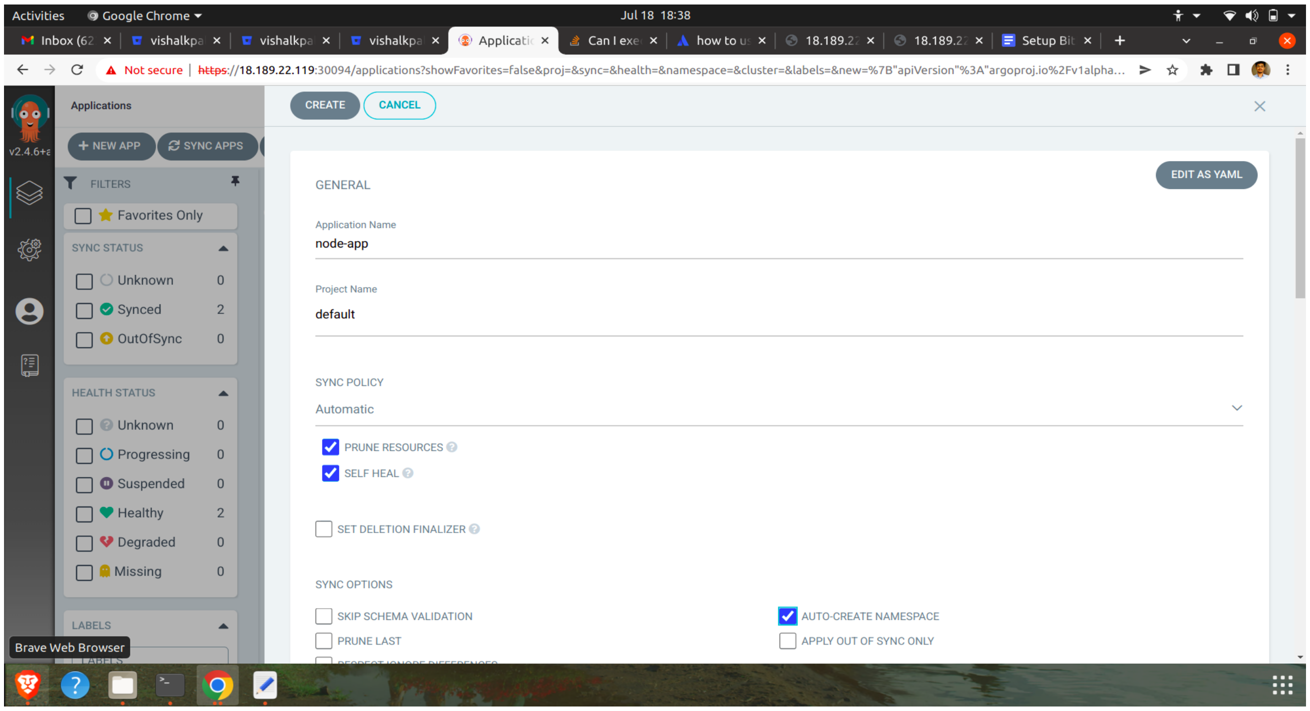
TYPE	NAME	REPOSITORY	CONNECTION STATUS
git	node-ssh	git@bitbucket.org:vishalkpal/helm.git	✓ Successful
git	helm-poc	git@bitbucket.org:vishalkpal/poc2.git	✓ Successful

.

Creating a Application in argoCD

We will be creating a project in argocd to continuously deploy the helm chart.

Clicking on the 3 dots of added repo we will adda project.
Keeping the sync policy as automatic this will automatically sync with the changes made in repo , we can also make it manual.



GENERAL

Application Name: node-app

Project Name: default

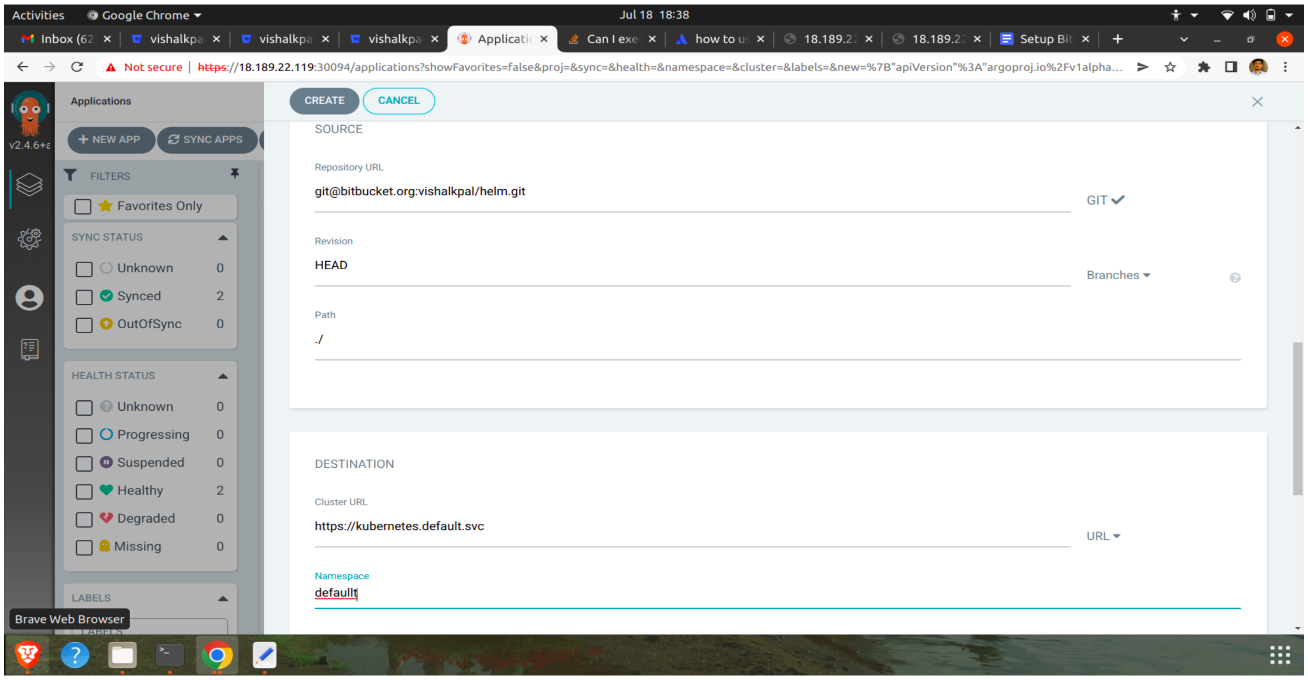
SYNC POLICY: Automatic

- [x] PRUNE RESOURCES
- [x] SELF HEAL
- [ ] SET DELETION FINALIZER

SYNC OPTIONS

- [ ] SKIP SCHEMA VALIDATION
- [x] AUTO-CREATE NAMESPACE
- [ ] PRUNE LAST
- [ ] APPLY OUT OF SYNC ONLY

Choose the cluster and Namespace we want to deploy our resources .



SOURCE

Repository URL: git@bitbucket.org:vishalkpal/helm.git

Revision: HEAD

Path: ./

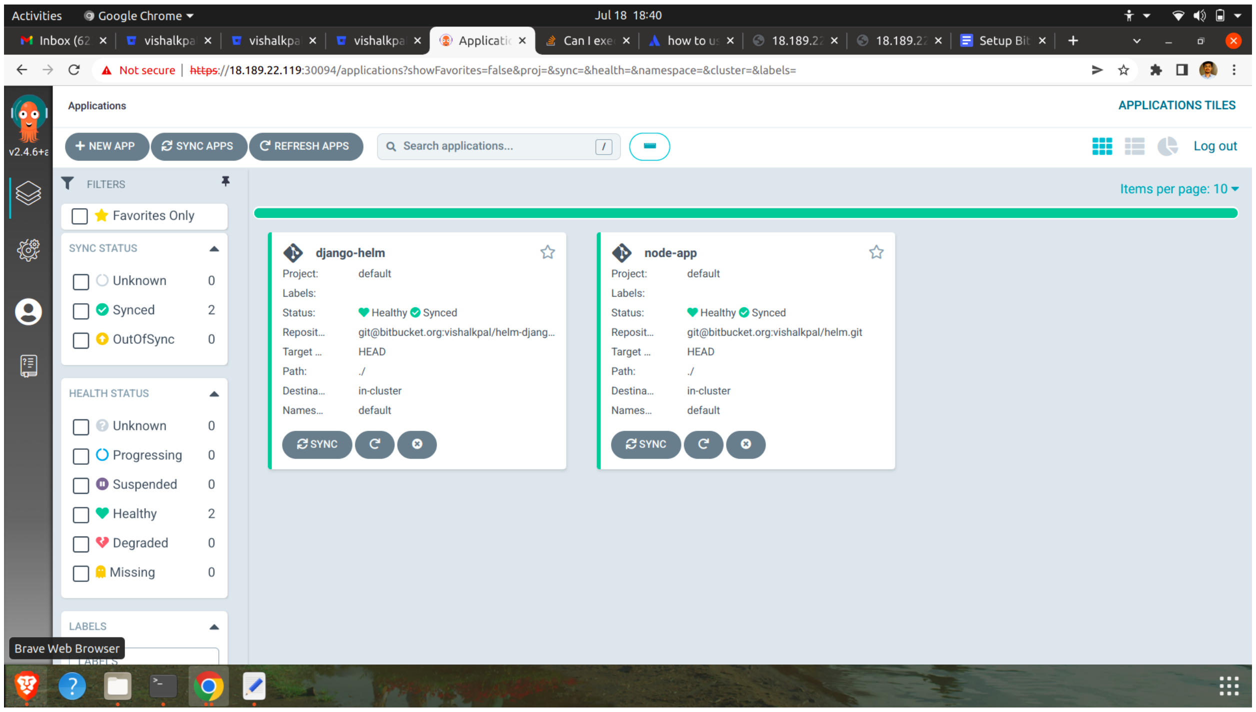
DESTINATION

Cluster URL: https://kubernetes.default.svc

Namespace: default

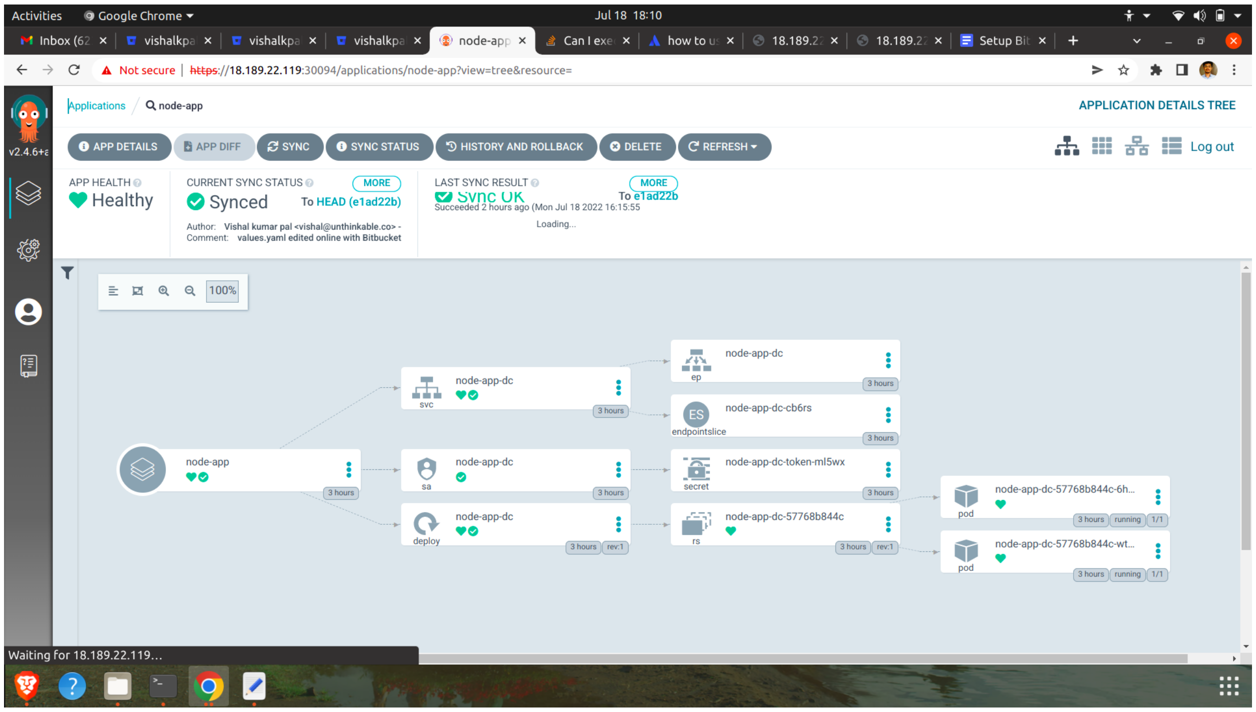
Choose the values.yml file in the Helm configuration.
Click on create to create the application .

After Creation it will automatically Deploy the application according to our Helm chart .



django-helm	
Project:	default
Labels:	
Status:	Healthy Synced
Reposit...	git@bitbucket.org:vishalkpal/helm-djang...
Target ...	HEAD
Path:	./
Destina...	in-cluster
Names...	default

node-app	
Project:	default
Labels:	
Status:	Healthy Synced
Reposit...	git@bitbucket.org:vishalkpal/helm.git
Target ...	HEAD
Path:	./
Destina...	in-cluster
Names...	default



APP HEALTH: Healthy

CURRENT SYNC STATUS: Synced To HEAD (e1ad22b)
Author: Vishal kumar pal <vishal@unthinkable.co> -
Comment: values.yaml edited online with Bitbucket

LAST SYNC RESULT: Sync OK To e1ad22b
Succeeded 2 hours ago (Mon Jul 18 2022 16:15:55
Loading...

We can monitor our deployed resources from here.
Thanks..