

KUBERNETES DEPLOYMENT CI WITH GITHUB ACTIONS CD WITH RANCHER.

INSTALL RANCHER

```
$ sudo docker run --privileged -d --restart=unless-stopped -p 80:80 -p 443:443 rancher/rancher
```

This will deploy a rancher container on docker and create a local k8s cluster.

```
$ docker exec -it <>container-id> /bin/bash
```

With this we can exec in the docker container as docker creates its own virtualized environment.

CREATE NODE REPO AND SETTING UP THE GITHUB ACTION PIPELINE

After pushing code , we will create an action pipeline.

```
name: ci

on:
  push:
    branches:
      - 'master'

jobs:
  docker:
    runs-on: ubuntu-latest
    steps:
      -
        name: Set up QEMU
        uses: docker/setup-qemu-action@v2
      -
        name: Set up Docker Buildx
        uses: docker/setup-buildx-action@v2
      -
        name: Login to Docker Hub
        uses: docker/login-action@v2
        with:
          username: ${ secrets.DOCKERHUB_USERNAME }
          password: ${ secrets.DOCKERHUB_PASSWORD }
      -
        name: Build and push
        uses: docker/build-push-action@v4
        with:
          push: true
          tags: vishalkrpal/nodeapp:latest
```

CREATING HELM CHART FOR DEPLOYMENT ON RANCHER VALUES.YML

```
replicaCount: 1

image:
  repository: vishalkrpal/nodeapp
  pullPolicy: Always
  # Overrides the image tag whose default is the chart
  tag: latest

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

serviceAccount:
  # Specifies whether a service account should be created
  create: true
  # Annotations to add to the service account
  annotations: {}
  # The name of the service account to use.
  # If not set and create is true, a name is generated
  name: ""

podAnnotations: {}
podSecurityContext: {}
  # fsGroup: 2000
securityContext: {}

service:
  type: NodePort
  port: 8000
```

CREATING A REPO WORKFLOW FOR HELM DEPLOYMENT

```
name: Helm CI

# Controls when the workflow will run
on:
  # Triggers the workflow on push or pull request events but only for the "master" branch
  push:
    branches: [ "master" ]
  pull_request:
    branches: [ "master" ]

  # Allows you to run this workflow manually from the Actions tab
  workflow_dispatch:

# A workflow run is made up of one or more jobs that can run sequentially or in parallel
jobs:
  # This workflow contains a single job called "build"
  build:
    # The type of runner that the job will run on
    runs-on: ubuntu-latest

    # Steps represent a sequence of tasks that will be executed as part of the job
    steps:
      # Checks-out your repository under $GITHUB_WORKSPACE, so your job can access it
      - uses: actions/checkout@v3

      # Runs a single command using the runners shell
      - name: Run a one-line script
        run: echo Hello, world!

      # Runs a set of commands using the runners shell
      - name: Run a multi-line script
        run: |
          echo Add other actions to build,
          echo test, and deploy your project.
```

We have to manually trigger the helm workflow as GITHUB ACTIONS does not provide the feature to trigger workflow in other repo after one is completed . But this feature is available on bitbucket.

This pipeline just triggers the commit action so rancher syncs with the latest changes.

CONFIGURING RANCHER CONTINUOUS DELIVERY

Goto -> continuous delivery

Choose cluster on top

Then setup the configuration in the Add repo section

The screenshot shows the Rancher Continuous Delivery interface. The sidebar on the left contains a menu with options: Dashboard, Git Repos, Clusters, Cluster Groups, and Advanced. The main content area is titled 'Git Repos' and features a table with columns: State, Name, Repo, Target, Clusters Ready, Resources, and Age. A single repository is listed with the name 'vishal-pro' and the repository URL 'vishalkpal/Helm-node'. The table also shows the target as 'Local', 1/1 clusters ready, 3 resources, and an age of 9 mins. Above the table are buttons for 'Pause', 'Force Update', 'Download YAML', and 'Delete'. A 'Filter' input field is also present. The bottom of the interface shows a system tray with weather information (33°C Haze) and a taskbar with various application icons.

The screenshot shows the 'Create: Step 1' section of the Rancher Continuous Delivery interface. The sidebar on the left contains a menu with options: Dashboard, Git Repos, Clusters, Cluster Groups, and Advanced. The main content area is titled 'Create: Step 1' and features a form for creating a new repository. The form includes fields for Name, Description, Repository URL, Watch, Branch Name, Git Authentication, Helm Authentication, and TLS Certificate Verification. There is also a section for Paths with an 'Add Path' button. The bottom of the interface shows a system tray with weather information (33°C Haze) and a taskbar with various application icons.

We can add labels and annotations depending on our needs.

Continuous Delivery

fleet-local

Compose

Dashboard

Git Repos

Clusters

Cluster Groups

Advanced

Git Repo: Create

When deploying a Git Repo to the Local workspace you are unable to target any specific Cluster or Cluster Groups

Create: Step 2

Define target details

Repository Details

Target Details

Deploy With

Service Account Name

Optional: Use a service account in the target clusters

Target Namespace

Optional: Require all resources to be in this namespace

Labels

Key/value pairs that are attached to objects which specify identifying attributes.

Add Label

Annotations

Add Annotation

Cancel

Edit as YAML

Previous

Create

Click "CREATE"

Continuous Delivery

fleet-local

Add Repository

Dashboard

Git Repos

Clusters

Cluster Groups

Advanced

Git Repos

Pause

Force Update

Download YAML

Delete

Filter

State	Name	Repo	Target	Clusters Ready	Resources	Age
Active	vishal-pro	vishalkpa/Helm-node master @ 3b7d85c	Local	1/1	3	36 mins

Command Prompt - docker

```

C:\Users\Vishal>docker exec -it f1f951bf6e6b /bin/bash
f1f951bf6e6b:/var/lib/rancher # curl localhost:31527
curl: (7) Failed to connect to localhost port 31527 after 0 ms: Connection refused
f1f951bf6e6b:/var/lib/rancher # kubectl get svc
NAME                TYPE        CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
kubernetes           ClusterIP   10.43.0.1      <none>          443/TCP          4d19h
node-svc             NodePort    10.43.30.40    <none>          8000:32000/TCP   28h
vishal-pro-hello-node NodePort    10.43.206.40   <none>          3000:31567/TCP   35m
f1f951bf6e6b:/var/lib/rancher # curl localhost:31567
Hello Node! from Vishal
f1f951bf6e6b:/var/lib/rancher #

```

Thanks
Vishal kumar pal