

swe645

Assignment HW2

Aditya Ketanbhai Shah – G01449892

Suhas Thandaga Ramesh – G01450932

Niharika FNU LNU – G01412073

Syedfurkhan Feroz Hashmi – G01411748

## Installation and Setup Instruction:

In this project swe645 HW2 has been dockerized and then deployed to self-hosted ec2 cluster in aws.

1. Rancher Node: <http://54.226.23.128:80>
2. Form URL: <http://54.226.23.128:30000/>
3. Jenkins URL: <http://54.226.23.128:8080>
4. Github Repo URL: [https://github.com/suhastr/jenkins\\_dummy](https://github.com/suhastr/jenkins_dummy)

## Setting up GitHub:

1. Push the files that you created in HW1 to a git repo (GitHub or BitBucket). First create an empty repository and then use Visual Studio Code commands to push the code a new repository.

## Docker installation Steps:

1. Make sure you have docker installed on your machine. I am using Docker Desktop . You will also need to create an account on <https://hub.docker.com/>
2. In the DockerFile, use the FROM command to get the initial image for the build.
3. On the command line, use this command: ‘docker build –tag userform’ You can use whatever name and tag you want.
4. Verify that the image is properly working by running ‘docker run -it -p 8182:8080 userform’ and open a browser at <http://localhost:8182/userform>

5. On the command line, login to docker using 'docker login -u ' i. Change the name of you image to be /: using the docker tag command. In my case it is: 'docker tag userform hekme5/userform
6. Verify that your image is on Docker Hub. Your image is accessible from the internet.

# Setup Kubernetes Cluster using Rancher:

Spin up EC2 host with atleast t2.medium as rancher consumes more memory.

The screenshot shows the AWS Management Console interface for launching an EC2 instance. The page is titled "Launch an instance" and includes a search bar at the top. The main content area is divided into several sections:

- Name and tags:** A text input field labeled "Name" contains the text "HW2". To the right of the input field is a button labeled "Add additional tags".
- Application and OS Images (Amazon Machine Image):** This section includes a search bar with the placeholder text "Search our full catalog including 1000s of application and OS images". Below the search bar is a "Quick Start" section with a grid of operating system logos: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE Linux. To the right of the grid is a link "Browse more AMIs" with a magnifying glass icon. Below the grid is a section titled "Amazon Machine Image (AMI)" showing the "Amazon Linux 2023 AMI" with its ID and a "Free tier eligible" badge.
- Summary:** This section on the right side of the page provides a summary of the instance configuration:
  - Number of instances:** A dropdown menu set to "1".
  - Software Image (AMI):** "Amazon Linux 2023 AMI 2023.3.2...read more" with the ID "ami-0d7a109bf30624c99".
  - Virtual server type (instance type):** "t2.micro".
  - Firewall (security group):** "New security group".
  - Storage (volumes):** "1 volume(s) - 8 GiB".
  - Free tier:** A blue box with a magnifying glass icon stating: "Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet." It includes a close button (X).
  - Buttons:** At the bottom of the summary section are "Cancel" and "Launch instance" buttons, with a "Review commands" link below the "Launch instance" button.

The bottom of the page shows a "Description" section.

1.sudo su

2.sudo apt update

Install docker in EC2 host created for Rancher for this usecase Ubuntu image is used.

3.sudo apt install docker.io

```
Get:6 http://us-east-1-ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:7 http://us-east-1-ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 C-n-f Metadata [286 kB]
Get:8 http://us-east-1-ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:9 http://us-east-1-ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:10 http://us-east-1-ec2.archive.ubuntu.com/ubuntu jammy/updates/main amd64 Packages [8372 B]
Get:11 http://us-east-1-ec2.archive.ubuntu.com/ubuntu jammy/updates/main amd64 Packages [1462 kB]
Get:12 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1246 kB]
Get:13 http://us-east-1-ec2.archive.ubuntu.com/ubuntu jammy/updates/main Translation-en [285 kB]
Get:14 http://us-east-1-ec2.archive.ubuntu.com/ubuntu jammy/updates/restricted amd64 Packages [1559 kB]
Get:15 http://us-east-1-ec2.archive.ubuntu.com/ubuntu jammy/updates/restricted Translation-en [259 kB]
Get:16 http://us-east-1-ec2.archive.ubuntu.com/ubuntu jammy/updates/universe amd64 Packages [1057 kB]
Get:17 http://us-east-1-ec2.archive.ubuntu.com/ubuntu jammy/updates/universe Translation-en [239 kB]
Get:18 http://us-east-1-ec2.archive.ubuntu.com/ubuntu jammy/updates/universe amd64 C-n-f Metadata [22.1 kB]
Get:19 http://us-east-1-ec2.archive.ubuntu.com/ubuntu jammy/updates/multiverse amd64 Packages [42.1 kB]
Get:20 http://us-east-1-ec2.archive.ubuntu.com/ubuntu jammy/updates/multiverse Translation-en [10.1 kB]
Get:21 http://us-east-1-ec2.archive.ubuntu.com/ubuntu jammy/updates/multiverse amd64 C-n-f Metadata [472 B]
Get:22 http://us-east-1-ec2.archive.ubuntu.com/ubuntu jammy/backports/main amd64 Packages [67.1 kB]
Get:23 http://us-east-1-ec2.archive.ubuntu.com/ubuntu jammy/backports/main Translation-en [11.0 kB]
Get:24 http://us-east-1-ec2.archive.ubuntu.com/ubuntu jammy/backports/main amd64 C-n-f Metadata [388 B]
Get:25 http://us-east-1-ec2.archive.ubuntu.com/ubuntu jammy/backports/restricted amd64 C-n-f Metadata [116 B]
Get:26 http://us-east-1-ec2.archive.ubuntu.com/ubuntu jammy/backports/universe amd64 Packages [28.4 kB]
Get:27 http://us-east-1-ec2.archive.ubuntu.com/ubuntu jammy/backports/universe Translation-en [16.2 kB]
Get:28 http://us-east-1-ec2.archive.ubuntu.com/ubuntu jammy/backports/universe amd64 C-n-f Metadata [644 B]
Get:29 http://us-east-1-ec2.archive.ubuntu.com/ubuntu jammy/backports/multiverse amd64 C-n-f Metadata [116 B]
Get:30 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [225 kB]
Get:31 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [1531 kB]
Get:32 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [255 kB]
Get:33 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [850 kB]
Get:34 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [162 kB]
Get:35 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 C-n-f Metadata [16.9 kB]
Get:36 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [37.1 kB]
Get:37 http://security.ubuntu.com/ubuntu jammy-security/multiverse Translation-en [7476 B]
Get:38 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 C-n-f Metadata [260 B]
Fetched 30.1 MB in 4s (6712 kB/s)
Reading package lists... Done
root@ip-172-31-53-210:/home/ubuntu# apt-get install docker.io
```

i-Oe88b015734e27d2a (HW2Machine)  
PublicIPs: 52.204.202.228 PrivateIPs: 172.31.53.210

4.systemctl start docker

5.systemctl enable docker

```
Selecting previously unselected package docker.io.
Preparing to unpack .../6-docker.io_24.0.5-0ubuntu1-22.04.1_amd64.deb ...
Unpacking docker.io (24.0.5-0ubuntu1-22.04.1) ...
Selecting previously unselected package ubuntu-fan.
Preparing to unpack .../7-ubuntu-fan_0.12.16_all.deb ...
Unpacking ubuntu-fan (0.12.16) ...
Setting up dnsmasq-base (2.90-0ubuntu0.22.04.1) ...
Setting up runc (1.1.7-0ubuntu1-22.04.2) ...
Setting up dns-root-data (2023112702-ubuntu0.22.04.1) ...
Setting up bridge-utils (1.7-1ubuntu3) ...
Setting up pigz (2.6-1) ...
Setting up containerd (1.7.2-0ubuntu1-22.04.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /lib/systemd/system/containerd.service.
Setting up ubuntu-fan (0.12.16) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ubuntu-fan.service → /lib/systemd/system/ubuntu-fan.service.
Setting up docker.io (24.0.5-0ubuntu1-22.04.1) ...
Adding group `docker' (GID 122) ...
Done.
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /lib/systemd/system/docker.socket.
Processing triggers for dbus (1.12.20-2ubuntu4.1) ...
Processing triggers for man-db (2.10.2-1) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-53-210:/home/ubuntu# systemctl start docker
root@ip-172-31-53-210:/home/ubuntu# systemctl enable docker
root@ip-172-31-53-210:/home/ubuntu#
```

i-Oe88b015734e27d2a (HW2Machine)

PublicIPs: 52.204.202.228 PrivateIPs: 172.31.53.210

6.docker run -d --privileged --restart=unless-stopped -p 80:80 -p 443:443 rancher/rancher:latest

```
CPU: 300ms
CGroup: /system.slice/docker.service
└─2632 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Mar 18 00:08:54 ip-172-31-53-210 systemd[1]: Starting Docker Application Container Engine...
Mar 18 00:08:54 ip-172-31-53-210 dockerd[2632]: time="2024-03-18T00:08:54.814045184Z" level=info msg="Starting up"
Mar 18 00:08:54 ip-172-31-53-210 dockerd[2632]: time="2024-03-18T00:08:54.815237382Z" level=info msg="detected 127.0.0.53 nameserver, assuming"
Mar 18 00:08:56 ip-172-31-53-210 dockerd[2632]: time="2024-03-18T00:08:56.141209997Z" level=info msg="Loading containers: start."
Mar 18 00:08:56 ip-172-31-53-210 dockerd[2632]: time="2024-03-18T00:08:56.446961651Z" level=info msg="Loading containers: done."
Mar 18 00:08:56 ip-172-31-53-210 dockerd[2632]: time="2024-03-18T00:08:56.485007499Z" level=info msg="Docker daemon" commit="24.0.5-0ubuntu1-"
Mar 18 00:08:56 ip-172-31-53-210 dockerd[2632]: time="2024-03-18T00:08:56.485147417Z" level=info msg="Daemon has completed initialization"
Mar 18 00:08:56 ip-172-31-53-210 dockerd[2632]: time="2024-03-18T00:08:56.525571849Z" level=info msg="API listen on /run/docker.sock"
Mar 18 00:08:56 ip-172-31-53-210 systemd[1]: Started Docker Application Container Engine.
root@ip-172-31-53-210:/home/ubuntu# docker run -d --privileged --restart=unless-stopped -p 80:80 -p 443:443 rancher/rancher:latest
Unable to find image 'rancher/rancher:latest' locally
latest: Pulling from rancher/rancher
6ef40564a417: Extracting [>] 491.5kB/47.85MB
3ee7ce020b3b: Download complete
211cblfa6431: Download complete
f01b14d487e8: Downloading [=====>] 65.96MB/469MB
2822f40a4b32: Downloading [=====>] 21.9MB/28.11MB
cd5e2aeb36b1: Waiting
e76e96859c75: Waiting
60c3c9a9b14b: Waiting
0ebee40395c1b: Waiting
cb72da885e82: Waiting
b0544211fdd6: Waiting
b1f521af6d5d: Waiting
4f0dda0d04e0: Waiting
fb0a3014e5a3: Waiting
213df31624a2: Waiting
4814962ef4de: Waiting
2b5f4cald739: Waiting
bd1b6c493504: Waiting
df9ee0b07a08: Waiting
05965dbb6f72: Waiting

i-Oe88b015734e27d2a (HW2Machine)
PublicIPs: 52.204.202.228 PrivateIPs: 172.31.53.210
```





## Welcome to Rancher

Learn more about the improvements and new capabilities in this version.

You can change what you see when you login via preferences

## Preferences



## Clusters

Manage

Import Existing

Create

State	Name	Provider	Kubernetes Version	CPU	Memory	Pods
Active	local	Local K3s	v1.27.6+k3s1	2 cores	7.74 GiB	6/110

## Links

Docs

## Forums

Slack

[File an Issue](#)

Get Started

### Commercial Support

7. Local cluster is created by default. We cannot schedule our application pods in that as it has taints to it.
8. Ec2 machine with tier t2.large this will be used as a cluster for deploying application code.
9. In Rancher UI navigate to cluster management for creating new cluster. There are various options present We can use different cloud providers or even use an ec2 iam user for managing the cluster. For our use case we have used custom one.



10. In case of custom when the cluster is created in order to make it active we have to register it with the Rancher node which was previously created. For this we have to execute the registration command in new EC2 node provided in Rancher UI. Once this is done cluster will appear as active in 4-5 mins.

The screenshot shows the Rancher UI's 'Cluster Management' section. The left sidebar contains navigation links: Clusters (3), Cloud Credentials, Drivers, RKE1 Configuration, and Advanced. The main content area is divided into two steps:

### Step 1

#### Node Role

Choose what roles the node will have in the cluster. The cluster needs to have at least one node with each role.

☒ etcd ☒ Control Plane ☒ Worker

[Show Advanced](#)

### Step 2

#### Registration Command

Run this command on each of the existing Linux machines you want to register.

```
curl -fL https://54.226.23.128/system-agent-install.sh | sudo sh -s - --server https://54.226.23.128 --label 'cattle.io/os=linux' --token 5k7mvxvk9mqhv5z6tssmm5rjw622nz9nq6q66vnqm4qm6p68bsz5ws --ca-checksum 84035c56f5b95bc4c01d27645607c875d11733f993093fdeca480fe4fc357839 --etcd --controlplane --worker
```

☐ Insecure: Select this to skip TLS verification if your server has a self-signed certificate.

Run this command in PowerShell on each of the existing Windows machines you want to register. Windows nodes can only be workers.

The cluster must be up and running with Linux etcd, control plane, and worker nodes before the registration command for adding Windows workers will display.

## 11. Adding Master to the cluster:

```
aws Services Search [Option+S] N. Virginia voclabs/user3120700=shashmi7@gmu.edu @ 7673-9778-905

Unable to find image 'rancher/rancher:latest' locally
latest: Pulling from rancher/rancher
6ef40564a417: Pull complete
3ee7ce020b3b: Pull complete
211cb1fa6431: Pull complete
f01b14d487e8: Pull complete
2822f40a4b32: Pull complete
cd5e2aeb36b1: Pull complete
e76e96859c75: Pull complete
60c3c9a9b14b: Pull complete
0ebe40395c1b: Pull complete
cb72da885e82: Pull complete
b0544211fdd6: Pull complete
b1f521af6d5d: Pull complete
4f0dda0d04e0: Pull complete
fb0a3014e5a3: Pull complete
213df31624a2: Pull complete
4814962ef4de: Pull complete
2b5f4cald739: Pull complete
bd1b6c493504: Pull complete
df9ee0b07a08: Pull complete
05965dbb6f72: Pull complete
Digest: sha256:ode7590c657f3600e215da4721988e4b48418d33090ad37349f58bebc4e66b93
Status: Downloaded newer image for rancher/rancher:latest
82e3d20d00d595fad35c1a9e8224c1f28bc77c0dd4f6ff4e8b4272bcbb62154f
root@ip-172-31-53-210:/home/ubuntu# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED            STATUS              PORTS                               NAMES
82e3d20d00d5       rancher/rancher:latest "entrypoint.sh"         43 seconds ago    Up 37 seconds      0.0.0.0:80->80/tcp, :::80->80/tcp, 0.0.0.0:443->443/tcp, :::443->443/tcp  vibrant_perlman
root@ip-172-31-53-210:/home/ubuntu# docker logs container-id 2>&1 | grep "Bootstrap Password:"
root@ip-172-31-53-210:/home/ubuntu# docker logs docker 82e3d20d00d5 2>&1 | grep "Bootstrap Password:"
root@ip-172-31-53-210:/home/ubuntu# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED            STATUS              PORTS                               NAMES
82e3d20d00d5       rancher/rancher:latest "entrypoint.sh"         2 minutes ago     Up 2 minutes       0.0.0.0:80->80/tcp, :::80->80/tcp, 0.0.0.0:443->443/tcp, :::443->443/tcp  vibrant_perlman
root@ip-172-31-53-210:/home/ubuntu# docker logs 82e3d20d00d5 2>&1 | grep "Bootstrap Password:"
2024/03/18 00:11:53 [INFO] Bootstrap Password: nft7dn44vsw2dft2dbs444vvb7bls77jxsb46d8dsfnx9dzs2246bl
root@ip-172-31-53-210:/home/ubuntu# curl --insecure -fL https://52.204.202.228/system-agent-install.sh | sudo sh -s -- --server https://52.204.202.228 --label 'cattle.io/os=linux' --token f2c7npqwx5j
bj4bdv681pdmrns1d7tv4vb5t97n5htr9n2c5xdc8t --ca-checksum dfe57e29492644f6dc78dc0f7554687ce0356a8932fdf5ecbb6d77086dcf0233 --etcd --controlplane --worker
i-Oe88b015734e27d2a (HW2Machine)
PublicIPs: 52.204.202.228 PrivateIPs: 172.31.53.210
```



Clusters

2

Cloud Credentials

Drivers

RKE1 Configuration



Advanced



# Clusters

Import Existing

Create

Download KubeConfig

Take Snapshot

Download YAML

Delete

Filter

<input type="checkbox"/> State	Name	Version	Provider	Machines	Age	
<input type="checkbox"/> Active	local	v1.27.6+k3s1	Local K3s	1	9 mins	Explore
<input type="checkbox"/> Active	master	v1.27.11+rke2r1	Custom RKE2	1	5 mins	Explore

## 12. Creating deployment and service file :

The screenshot displays the Rancher web interface. On the left, a sidebar contains navigation icons and labels: a hamburger menu, a home icon, a server icon, 'MTR', a sailboat icon, a folder icon, a double arrow icon, a person icon, a puzzle piece icon, a globe icon, and 'About'. The main panel is titled 'Cluster Management' and has a sub-header 'Clusters' with a count of '2'. Below this, there are links for 'Cloud Credentials', 'Drivers', 'RKE1 Configuration', and 'Advanced'. A table titled 'Clusters' shows two clusters: 'local' (version v1.27.6+k3s1, Local K3s provider) and 'master' (version v1.27.11+rke2r1, Custom RKE2 provider). Both are marked as 'Active'. Above the table are buttons for 'Download kubeConfig', 'Take Snapshot', 'Download YAML', and 'Delete'. Below the table, a terminal window shows a 'Kubectl: master' prompt and a deployment.yaml file. The file content is as follows:

```
replicas: 3
selector:
  matchLabels:
    app: user-form
template:
  metadata:
    labels:
      app: user-form
spec:
  containers:
    - name: user-form
      image: dunkdock/user-form:latest_1.2
      ports:
        - containerPort: 80
```

At the bottom of the terminal, there is a 'Clear' button and a 'Connected' status indicator.

Fig. Deployment.yaml

Cluster Management

Clusters

Cloud Credentials

Drivers

RKE1 Configuration

Advanced

v2.8.2

Kubectl: master

apiVersion: v1  
kind: Service  
metadata:  
 name: user-form-service  
spec:  
 selector:  
 app: user-form  
 ports:  
 - protocol: TCP  
 port: 80  
 targetPort: 80  
 nodePort: 30000  
 type: NodePort

Clear Connected

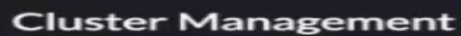
Clusters

Download KubeConfig Take Snapshot Download

State	Name	Version
Active	local	v1.27.6+k3s1
Active	master	v1.27.11+rke2r1

Fig. Service.yaml





## Clusters

2

## Cloud Credentials

## Drivers

## RKE1 Configuration

>

## Advanced

>

v2.8.2

Σ Kubectl: master 




```
# Run kubectl commands inside here
# e.g. kubectl get all
v vim deployment.yaml
v vim service.yaml
v kubectl apply -f deployment.yaml
deployment.apps/user-form-deployment created
v kubectl apply -f service.yaml
```


## About

Clear


Connected

## Clusters

 [Download KubeConfig](#)

 Take

State

Name ☐ Active

local

☐ Active

master

13. Displaying pod status:

Home

MTR

About

Cluster Management

Clusters2

Cloud Credentials

Drivers

RKE1 Configuration>

Advanced>

v2.8.2

Download KubeConfig

Take Snapshot

State

Name

Version

Activelocalv1.27.6+k3s1

Activemasterv1.27.11+rke2r1

Kubectl: master

> kubectl apply -f deployment.yaml  
deployment.apps/user-form-deployment created  
> kubectl apply -f service.yaml  
service/user-form-service created  
> kubectl get all

NAME	READY	STATUS	RESTARTS	AGE
pod/user-form-deployment-6dd79c7f65-9qvdh	1/1	Running	0	24s
pod/user-form-deployment-6dd79c7f65-gdtn4	1/1	Running	0	24s
pod/user-form-deployment-6dd79c7f65-kwqhq	1/1	Running	0	24s

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/kubernetes	ClusterIP	10.43.0.1	<none>	443/TCP	8m13s
service/user-form-service	NodePort	10.43.132.165	<none>	80:30000/TCP	16s

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/user-form-deployment	3/3	3	3	24s

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/user-form-deployment-6dd79c7f65	3	3	3	24s

Clear

Connected

14. Now access this [Form](#):

← → ↻ ⚠ Not Secure 54.226.23.128:30000

🤖 Getting Started wi... 🌐 Create beautiful gr... 📺 CSS3 Full Screen... 📺 Motion Graphics A... 📺 Corporate Video P... 📺 Places to visit in M... 📺 How to Write a Pro... 👤 Defi

## Student Survey Form

**First Name \***

**Last Name \***

**Street Address \***

**City \***

**State \***

**Zip \***

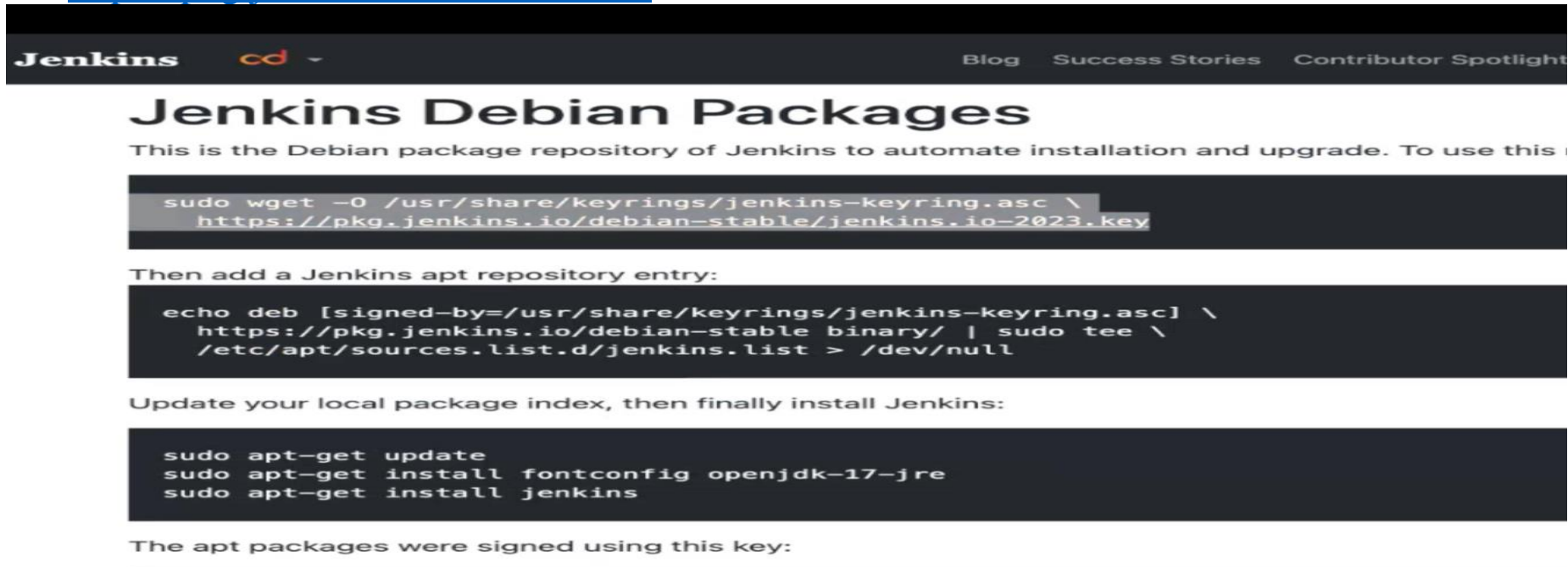
**Telephone \***

**Email \***


## Jenkins Setup:

1. Setup the EC2 instance with same steps as in First assignment.
2. Connect the ec2 instance and install docker  
`sudo apt get install docker.io`
3. Enable docker and start docker  
`sudo systemctl start docker`  
`sudo systemctl enable docker`
4. Follow the steps outlined in below link to install jenkins

<https://pkg.jenkins.io/debian-stable/>



The screenshot shows the Jenkins Debian Packages website. At the top, there is a navigation bar with the Jenkins logo and links for Blog, Success Stories, and Contributor Spotlight. The main heading is "Jenkins Debian Packages". Below this, a text block states: "This is the Debian package repository of Jenkins to automate installation and upgrade. To use this". A code block follows, showing the command to download the Jenkins key: `sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \ https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key`. Below the code block, it says "Then add a Jenkins apt repository entry:". Another code block shows the command to add the repository: `echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \ https://pkg.jenkins.io/debian-stable binary/ | sudo tee \ /etc/apt/sources.list.d/jenkins.list > /dev/null`. Below this, it says "Update your local package index, then finally install Jenkins:". A final code block shows the commands to update and install Jenkins: `sudo apt-get update`, `sudo apt-get install fontconfig openjdk-17-jre`, and `sudo apt-get install jenkins`. At the bottom, it says "The apt packages were signed using this key:".

Jenkins  Blog Success Stories Contributor Spotlight

## Jenkins Debian Packages

This is the Debian package repository of Jenkins to automate installation and upgrade. To use this

```
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
```

Then add a Jenkins apt repository entry:

```
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
/etc/apt/sources.list.d/jenkins.list > /dev/null
```

Update your local package index, then finally install Jenkins:

```
sudo apt-get update
sudo apt-get install fontconfig openjdk-17-jre
sudo apt-get install jenkins
```

The apt packages were signed using this key:

5. Using public ip address like <http://54.226.23.128:8080> paste it in browser to access jenkins dashboard.
6. Where u can see it prompts user to enter password, copy that directory link and paste it in the EC2 you willget the password and again paste it in the jenkins page.
7. In EC2 instance you need to create directory and file. Also, copy KubeConfig content into config file of EC2

```
mkdir /var/lib/jenkins/.kube
touch /var/lib/jenkins/.kube/config
```

Cluster Management

Clusters2

Cloud Credentials

Drivers

RKE1 Configuration>

Advanced>

v2.8.2

Kubectl: master

> kubectl apply -f deployment.yaml  
deployment.apps/user-form-deployment created  
> kubectl apply -f service.yaml  
service/user-form-service created  
> kubectl get all

NAME	READY	STATUS	RESTARTS	AGE
pod/user-form-deployment-6dd79c7f65-9qvdh	1/1	Running	0	24s
pod/user-form-deployment-6dd79c7f65-gdtn4	1/1	Running	0	24s
pod/user-form-deployment-6dd79c7f65-kwqhq	1/1	Running	0	24s

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/kubernetes	ClusterIP	10.43.0.1	<none>	443/TCP	8m13s
service/user-form-service	NodePort	10.43.132.165	<none>	80:30000/TCP	16s

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/user-form-deployment	3/3	3	3	24s

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/user-form-deployment-6dd79c7f65	3	3	3	24s

Close

Clusters

Import Existing

Create

Download KubeConfig

Take Snapshot

Download YAML

Delete1 selected

Filter

State	Name	Version	Provider	Machines	Age	
Active	local	v1.27.6+k3s1	Local K3s	1	18 mins	Explore
Active	master	v1.27.11+rke2r1	Custom RKE2	1		

Kubectl Shell

Download KubeConfig

Copy KubeConfig to Clipboard

Take Snapshot

Restore Snapshot

Rotate Certificates

Rotate Encryption Keys

Edit Config

Edit YAML

Download YAML

Delete

```
Scanning candidates...
```

```
Scanning linux images...
```

```
Running kernel seems to be up-to-date.
```

```
Restarting services...
```

```
Service restarts being deferred:
```

```
systemctl restart rke2-server.service
```

```
No containers need to be restarted.
```

```
No user sessions are running outdated binaries.
```

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
```

```
root@ip-172-31-53-210:/home/ubuntu# systemctl status jenkins
```

```
● jenkins.service - Jenkins Continuous Integration Server
```

```
Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)
```

```
Active: active (running) since Mon 2024-03-18 00:29:22 UTC; 35s ago
```

```
Main PID: 27847 (java)
```

```
Tasks: 51 (limit: 9498)
```

```
Memory: 2.0G
```

```
CPU: 38.589s
```

```
CGroup: /system.slice/jenkins.service
```

```
└─27847 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war
```

```
Mar 18 00:29:03 ip-172-31-53-210 jenkins[27847]: e1055a1la9314b08b4c727d7e420ad1f
```

```
Mar 18 00:29:03 ip-172-31-53-210 jenkins[27847]: This may also be found at: /var/lib/jenkins
```

```
Mar 18 00:29:03 ip-172-31-53-210 jenkins[27847]: *****
```

```
Mar 18 00:29:03 ip-172-31-53-210 jenkins[27847]: *****
```

```
Mar 18 00:29:03 ip-172-31-53-210 jenkins[27847]: *****
```

```
Mar 18 00:29:22 ip-172-31-53-210 jenkins[27847]: 2024-03-18 00:29:22.887+0000 [id=34]
```

```
Mar 18 00:29:22 ip-172-31-53-210 jenkins[27847]: 2024-03-18 00:29:22.909+0000 [id=24]
```

```
Mar 18 00:29:22 ip-172-31-53-210 systemd[1]: Started Jenkins Continuous Integration Server.
```

```
Mar 18 00:29:22 ip-172-31-53-210 jenkins[27847]: 2024-03-18 00:29:22.950+0000 [id=49]
```

```
Mar 18 00:29:22 ip-172-31-53-210 jenkins[27847]: 2024-03-18 00:29:22.951+0000 [id=49]
```

```
root@ip-172-31-53-210:/home/ubuntu# mkdir /var/lib/jenkins/.kube
```

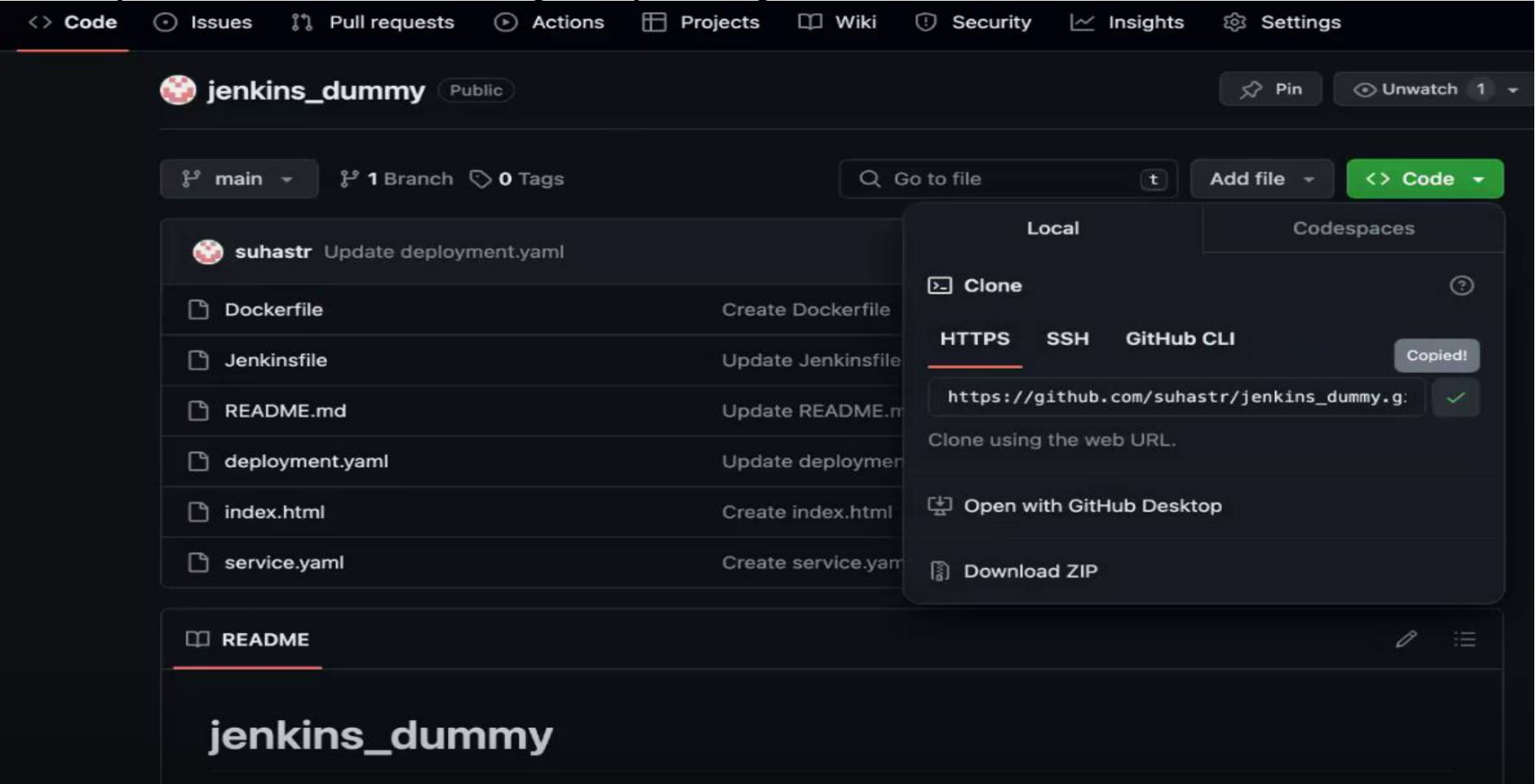
```
root@ip-172-31-53-210:/home/ubuntu# vim /var/lib/jenkins/.kube/config.
```

i-0e88b015734e27d2a (HW2Machine)

PublicIPs: 52.204.202.228 PrivateIPs: 172.31.53.210




8. Now you have to create a Public repository on the github which looks like the below:





9. From Jenkins dashboard, click on “New item” and then this prompt will show:


 **Jenkins**

Search (⌘+K) ? 1 test log out

Dashboard > All >


### Enter an item name

» This field cannot be empty, please enter a valid name




**Freestyle project**

Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.




**Pipeline**

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.



**Multi-configuration project**

Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.



**Folder**

Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

OK

Multibranch Pipeline

10. Enter Git Repository link in pipeline configuration. Also mention branch name

Dashboard > simple-deployment > Configuration

Configure

General

Advanced Project Options

Pipeline

SCM ?

Git

Repositories ?

Repository URL ?

https://github.com/suhastr/jenkins\_dummy.git

Please enter Git repository.

Credentials ?

- none -

+ Add

Advanced

Add Repository

Branches to build ?

Save

Apply

# 11. Go to Pipeline Syntax and generate Secret text

Dashboard > simple-deployment > Pipeline Syntax

Global Variables Reference

Online Documentation

Examples Reference

IntelliJ IDEA GDSL

Sample Step

withCredentials: Bind credentials to variables

withCredentials ?

Secret values are masked on a best-effort basis to prevent *accidental* disclosure. Multiline secrets, such as the contents of a SSH private key file, are not masked. See the inline help for details and usage guidelines.

Bindings

≡ Secret text ?

Variable ?

docker\_hub\_cred

Credentials ?

docker\_cred

Add

Add

Generate Pipeline Script

12. Copy the generated script into Jenkinsfile as shown below:

Files

main

Go to file

Dockerfile

Jenkinsfile

README.md

deployment.yaml

index.html

service.yaml

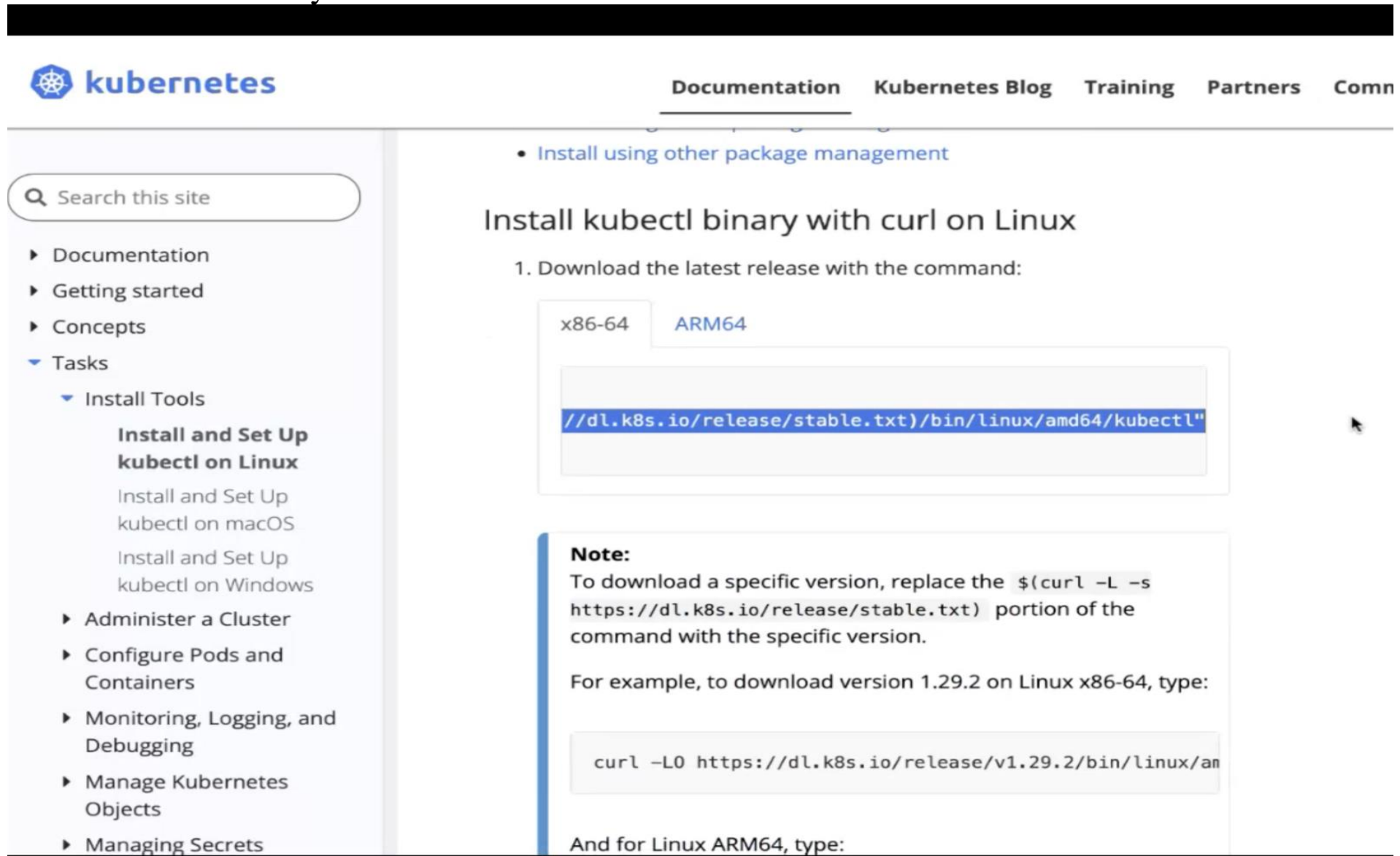
jenkins\_dummy / Jenkinsfile

CodeBlame66 lines (57 loc) · 2 KBCode 55% faster with GitHub Copilot

RawCopyDownloadEdit

```
24         steps {
25             script {
26                 docker.build("${DOCKER_IMAGE}:${DOCKER_TAG}")
27             }
28         }
29     }
30
31     stage('Push Docker Image') {
32     steps {
33         script {
34             // Get Docker Hub credentials
35             withCredentials([string(credentialsId: 'docker_cred', variable: 'DOCKER_CREDENTIALS')]) {
36                 // Login to Docker Hub
37                 sh "echo \${DOCKER_CREDENTIALS} | docker login --username ${DOCKER_USERNAME} --password-stdin"
38
39                 // Push Docker image to Docker Hub
40                 docker.image("${DOCKER_IMAGE}:${DOCKER_TAG}").push()
41             }
42         }
43     }
44 }
45
46
47 stage('Deploy to Kubernetes') {
48     steps {
49         script {
50             // Deploy the Kubernetes deployment and service
51             sh "kubectl apply -f ${DEPLOYMENT_YAML_PATH}"
52             sh "kubectl apply -f ${SERVICE_YAML_PATH}"
53         }
54     }
```

### 13. Install Kubectl binary with curl on linux



The screenshot shows the Kubernetes documentation website. The top navigation bar includes links for Documentation, Kubernetes Blog, Training, Partners, and Comm. The left sidebar contains a search bar and a list of navigation items: Documentation, Getting started, Concepts, Tasks (expanded), and Install Tools. Under Install Tools, there is a section for 'Install and Set Up kubectl on Linux' with sub-links for macOS, Windows, and Linux. The main content area is titled 'Install kubectl binary with curl on Linux' and includes a sub-header '1. Download the latest release with the command:'. Below this, there are tabs for 'x86-64' and 'ARM64'. The 'x86-64' tab is active, showing a code block with the command: `curl -LO https://dl.k8s.io/release/stable.txt`. A 'Note' section explains how to download a specific version by replacing the stable.txt path with a versioned path. Below the note, there is another code block showing the command: `curl -LO https://dl.k8s.io/release/v1.29.2/bin/linux/amd64/kubectl`. The 'ARM64' tab is also visible, showing a similar command structure.

Documentation Kubernetes Blog Training Partners Comm

Search this site

- Documentation
- Getting started
- Concepts
- Tasks
  - Install Tools
    - Install and Set Up kubectl on Linux
      - Install and Set Up kubectl on macOS
      - Install and Set Up kubectl on Windows
  - Administer a Cluster
  - Configure Pods and Containers
  - Monitoring, Logging, and Debugging
  - Manage Kubernetes Objects
  - Managing Secrets

• Install using other package management

## Install kubectl binary with curl on Linux

1. Download the latest release with the command:

x86-64 ARM64

```
curl -LO https://dl.k8s.io/release/stable.txt
```

**Note:**  
To download a specific version, replace the `$(curl -L -s https://dl.k8s.io/release/stable.txt)` portion of the command with the specific version.

For example, to download version 1.29.2 on Linux x86-64, type:


```
curl -LO https://dl.k8s.io/release/v1.29.2/bin/linux/amd64/kubectl
```

And for Linux ARM64, type:


14. Now go to your ec2 instance command line, type the below command to give permission to jenkins


```
sudo usermod -aG docker jenkins
```


15. In the Jenkins dashboard select build now option and Jenkins will start building the processes stage by stage


 **Jenkins**

Search (⌘+K)





 1


 test ▾


 log out


Dashboard > test >


 Status


 Changes


 **Build Now**


 Configure


 Delete Pipeline

 Full Stage View


 GitHub

 Rename

 Pipeline Syntax

 Build History

trend ▾

 **test**

Add description

Disable Project

Stage View

#4

Mar 17 16:33

2 commits

Average stage times:  
(Average full run time: ~7s)

Declarative: Checkout SCM	Clone Repository	Build Docker Image	Push Docker Image	Deploy to Kubernetes	Declarative: Post Actions
515ms	452ms	2s	2s	949ms	150ms
449ms	293ms	1s	1s	1s	141ms

16. At last you can see the deployment on your rancher like shown below:

← → ↺

Not Secure https://54.226.23.128/dashboard/c/\_/manager/provisioning.cattle.io.cluster

Getting Started wi... Create beautiful gr... CSS3 Full Screen... Motion Graphics A... Corporate Video P... Places to visit in M... How to Write a P

Cluster Management

Clusters 2

Cloud Credentials

Drivers

RKE1 Configuration >

Advanced >

v2.8.2

Clusters

Download KubeConfig

Take Snapshot

Download YAML

Delete

State	Name	Version	Provider
Active	local	v1.27.6+k3s1	Local K3s
Active	master	v1.27.11+rke2r1	Custom RKE2

⌵ Kubectl: master

deployment.apps/user-form-deployment

3/3

3

3

15m

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/user-form-deployment-75f944474c	3	3	3	22s
replicaset.apps/user-form-deployment-77bb98c667	0	0	0	15m

> kubectl get all

NAME	READY	STATUS	RESTARTS	AGE
pod/user-form-deployment-75f944474c-7txgb	1/1	Running	0	28m
pod/user-form-deployment-75f944474c-n99nl	1/1	Running	0	28m
pod/user-form-deployment-75f944474c-qwnw6	1/1	Running	0	28m

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/kubernetes	ClusterIP	10.43.0.1	<none>	443/TCP	69m
service/user-form-service	NodePort	10.43.11.163	<none>	80:30000/TCP	43m

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/user-form-deployment	3/3	3	3	43m

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/user-form-deployment-75f944474c	3	3	3	28m
replicaset.apps/user-form-deployment-77bb98c667	0	0	0	43m

>

17. Now access the form using this [Form](#)

←

→

↺

⚠ Not Secure 54.226.23.128:30000

🤖 Getting Started wi...

🌐 Create beautiful gr...

📺 CSS3 Full Screen...

📺 Motion Graphics A...

📺 Corporate Video P...

📺 Places to visit in M...

📖 How to Write a Pro...

👤 Defi

Student Survey Form

First Name \*

Last Name \*

Street Address \*

City \*

State \*

Zip \*

Telephone \*

Email \*



## References:

<https://kubernetes.io/docs/tasks/tools/install-kubectl-linux/#install-kubectl-binary-with-curl-on-linux>  
<https://pkg.jenkins.io/debian-stable/>

## Contributions:

Aditya Ketanbhai Shah – G01449892 : 25%

Suhas Thandaga Ramesh – G01450932 : 25%

Niharika FNU LNU – G01412073 : 25%

Syedfurkhan Feroz Hashmi – G01411748 : 25%