

# Cloud DevOps EPAM Final Project

John Santosh Rudrapogu

2000031237

Sec – 13

Prof. G. Bindu

## 1. Login to AWS Management Console -> S3 -> Create Bucket

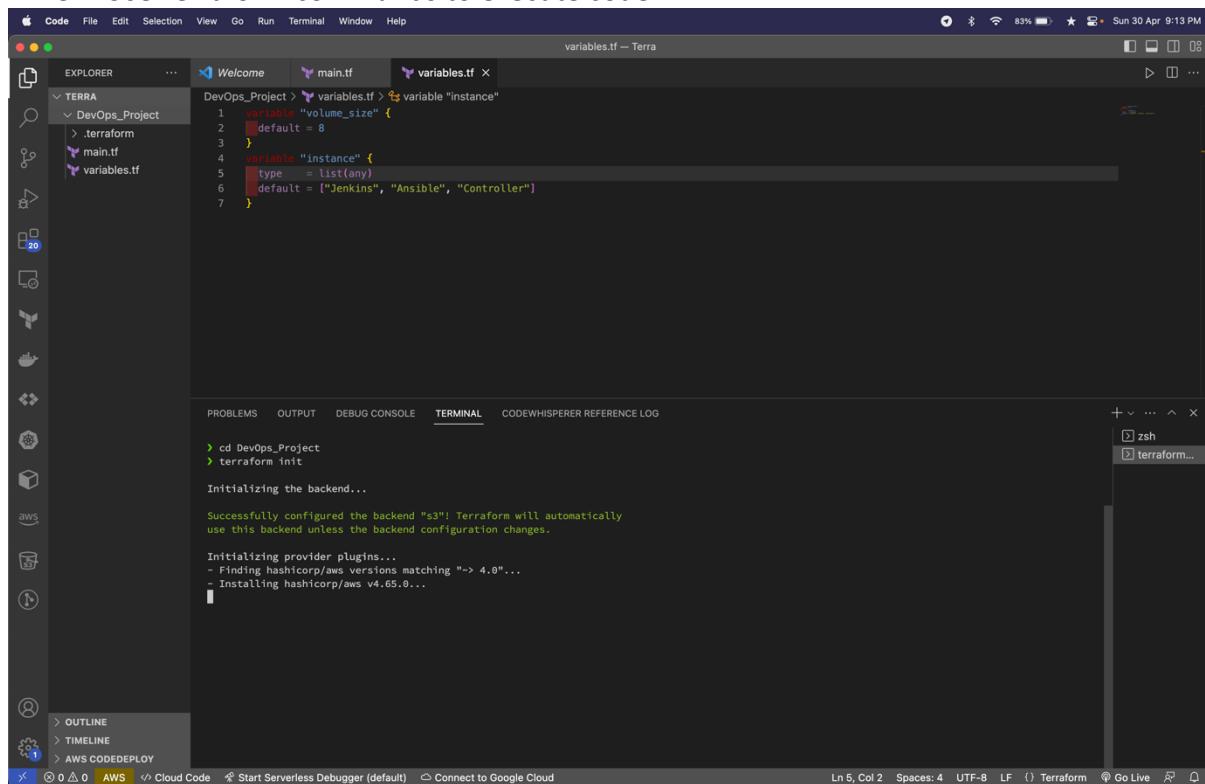
The screenshot shows the 'Create bucket' page in the AWS Management Console. The 'General configuration' section contains fields for 'Bucket name' (set to 'devops-31237') and 'AWS Region' (set to 'US East (N. Virginia) us-east-1'). Below these, there's a note about copy settings from existing buckets. The 'Object Ownership' section has two options: 'ACLs disabled (recommended)' (unchecked) and 'ACLs enabled' (checked). A note below recommends disabling ACLs. The 'Object Ownership' section also includes 'Bucket owner preferred' and 'Object writer' options. At the bottom, there's a note about enabling server-side encryption. The footer includes links for CloudShell, Feedback, Language, Privacy, Terms, and Cookie preferences.

## 2. Main.tf code

The screenshot shows the Terraform IDE interface with the 'main.tf' file open. The code defines an AWS instance named 'my-machine' with an S3 backend bucket 'devops-31237'. It also creates an AWS VPC named 'mainvpc' with a security group 'cvw\_sg\_ssh' allowing inbound traffic on port 22 and outbound traffic on port 80. The code uses Terraform version 4.0.8 and the AWS provider.

```
1 terraform {
2   required_providers {
3     aws = {
4       source = "hashicorp/aws"
5       version = ">= 4.0.8"
6     }
7   }
8   backend "s3" {
9     bucket="devops-31237"
10    encrypt=true
11    key="terraform.tfstate"
12    region = "us-east-1"
13  }
14 }
15 provider "aws" {
16   region = "us-east-1"
17 }
18 resource "aws_vpc" "mainvpc" {
19   cidr_block = "10.1.0.0/16"
20 }
21 resource "aws_security_group" "cvw_sg_ssh" {
22   name = "cvw-blog-3-sg-using-terraform"
23   #Incoming traffic
24   ingress {
25     from_port = 22
26     to_port = 22
27     protocol = "tcp"
28     cidr_blocks = ["0.0.0.0/0"] #replace it with your ip address
29   }
30   #Outgoing traffic
31   egress {
32     from_port = 0
33     to_port = 0
34     protocol = "tcp"
35     cidr_blocks = ["0.0.0.0/0"]
36   }
37 }
38 resource "aws_instance" "my-machine" {
39   ami           = "ami-0f3a2a2a2a2a2a2a2"
40   instance_type = "t2.micro"
41   security_groups = [aws_security_group_cvw_sg_ssh.id]
42   subnet_id = "subnet-00000000"
43 }
```

### 3. Use Terraform commands to execute code.



```
variables.tf — Terra
```

```
variable "volume_size" {  
    default = 8  
}  
  
variable "instance" {  
    type = list(any)  
    default = ["Jenkins", "Ansible", "Controller"]  
}
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL CODEWHISPERER REFERENCE LOG
```

```
> cd DevOps_Project  
> terraform init
```

```
Initializing the backend...
```

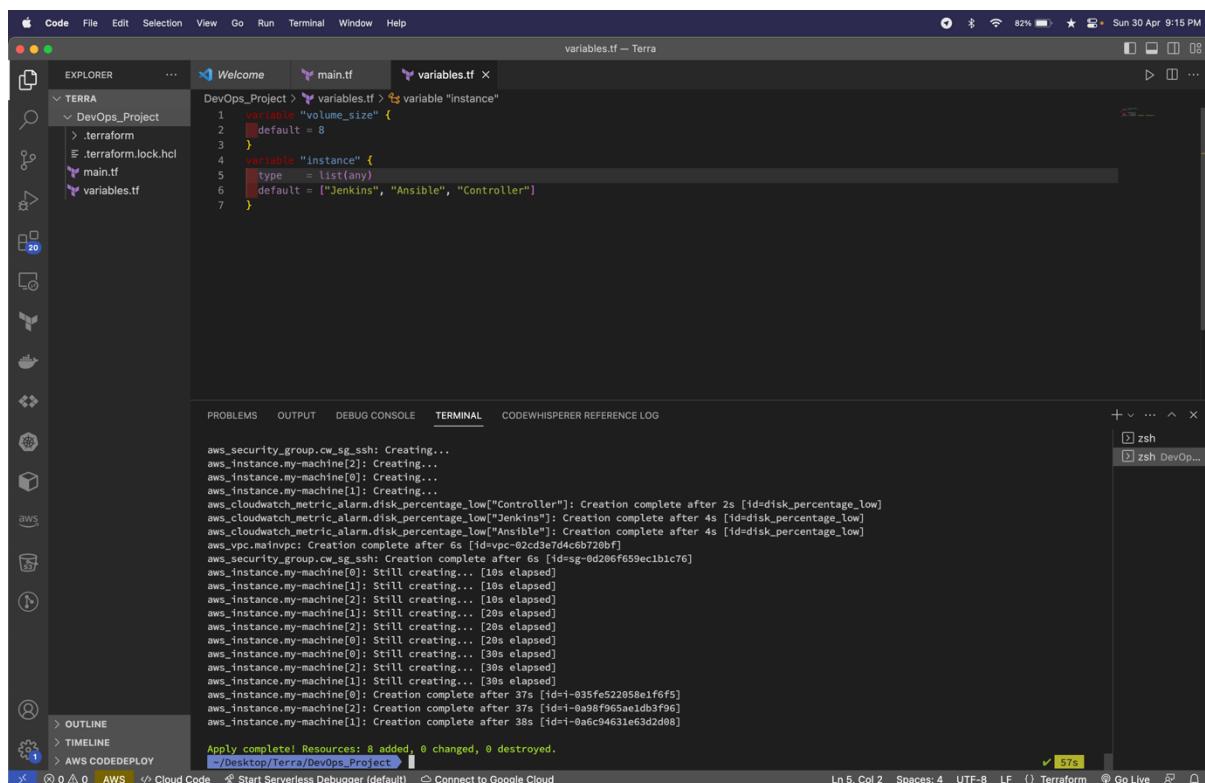
```
Successfully configured the backend "s3"! Terraform will automatically  
use this backend unless the backend configuration changes.
```

```
Initializing provider plugins...
```

```
- Finding hashicorp/aws versions matching "~> 4.0"...
```

```
- Installing hashicorp/aws v4.65.0...
```

```
Ln 5, Col 2 Spaces: 4 UTF-8 LF () Terraform Go Live
```



```
variables.tf — Terra
```

```
variable "volume_size" {  
    default = 8  
}  
  
variable "instance" {  
    type = list(any)  
    default = ["Jenkins", "Ansible", "Controller"]  
}
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL CODEWHISPERER REFERENCE LOG
```

```
aws_security_group.cw_sg_ssh: Creating...  
aws_instance.my-machine[2]: Creating...  
aws_instance.my-machine[0]: Creating...  
aws_instance.my-machine[1]: Creating...  
aws_cloudwatch_metric_alarm.disk_percentage_low["Controller"]: Creation complete after 2s [id=disk_percentage_low]  
aws_cloudwatch_metric_alarm.disk_percentage_low["Jenkins"]: Creation complete after 4s [id=disk_percentage_low]  
aws_cloudwatch_metric_alarm.disk_percentage_low["Ansible"]: Creation complete after 4s [id=disk_percentage_low]  
aws_vpc.mainvpc: Creation complete after 6s [id=vpc-02cd3e7d4c6b720bf]  
aws_security_group.cw_sg_ssh: Creation complete after 6s [id=sg-0d206f659ec1b1c76]  
aws_instance.my-machine[0]: Still creating... [10s elapsed]  
aws_instance.my-machine[1]: Still creating... [10s elapsed]  
aws_instance.my-machine[2]: Still creating... [10s elapsed]  
aws_instance.my-machine[0]: Still creating... [20s elapsed]  
aws_instance.my-machine[1]: Still creating... [20s elapsed]  
aws_instance.my-machine[2]: Still creating... [20s elapsed]  
aws_instance.my-machine[0]: Still creating... [30s elapsed]  
aws_instance.my-machine[2]: Still creating... [30s elapsed]  
aws_instance.my-machine[1]: Still creating... [30s elapsed]  
aws_instance.my-machine[0]: Creation complete after 37s [id=1-035fe522058e1ff6f5]  
aws_instance.my-machine[2]: Creation complete after 37s [id=1-0a9sf965ae1db3f96]  
aws_instance.my-machine[1]: Creation complete after 38s [id=1-0a6c94631e63d2d08]
```

```
Apply complete! Resources: 8 added, 0 changed, 0 destroyed.  
-/Desktop/Terra/DevOps_Project
```

```
Ln 5, Col 2 Spaces: 4 UTF-8 LF () Terraform Go Live 57s
```

## 4. Create a domain using Route 53.

**Hosted zone configuration**

A hosted zone is a container that holds information about how you want to route traffic for a domain, such as example.com, and its subdomains.

**Domain name** [Info](#)  
This is the name of the domain that you want to route traffic for.  
**jrccloud.in**  
Valid characters: a-z, 0-9, ! \* # \$ % & ' ( ) \* + , - / ; < = > ? @ { \ } ^ \_ - { } . ~

**Description - optional** [Info](#)  
This value lets you distinguish hosted zones that have the same name.  
The hosted zone is used for...

The description can have up to 256 characters. 0/256

**Type** [Info](#)  
The type indicates whether you want to route traffic on the internet or in an Amazon VPC.  
 **Public hosted zone** A public hosted zone determines how traffic is routed on the internet.  
 **Private hosted zone** A private hosted zone determines how traffic is routed within an Amazon VPC.

**VPCs to associate with the hosted zone** [Info](#)  
To use this hosted zone to resolve DNS queries for one or more VPCs, choose the VPCs. To associate a VPC with a hosted zone when the VPC was created using a different AWS account, you must use a programmatic method, such as the AWS CLI.

For each VPC that you associate with a private hosted zone, you must set the Amazon VPC settings enableDnsHostnames and enableDnsSupport to true.

Region	Info	VPC ID	Info
US East (N. Virginia) [us-east-1]	vpc-0bbc359bc92e698d6	Use: vpc-0bbc359bc92e698d6 vpc-0bbc359bc92e698d6 vpc-02cd5e7d4c6b720bf	Remove VPC
Add VPC			

## 5. Connect to one of the instance that you named Controller-Machine.

**EC2** > Instances > i-035fe522058e1f6f5 > Connect to instance

**Connect to instance** [Info](#)  
Connect to your instance i-035fe522058e1f6f5 (Controller-Machine) using any of these options

**SSH client** (selected)

**EC2 Instance Connect** | **Session Manager** | **SSH client** | **EC2 serial console**

**Instance ID**  
i-035fe522058e1f6f5 (Controller-Machine)

1. Open an SSH client.  
2. Locate your private key file. The key used to launch this instance is NVKey.pem  
3. Run this command, if necessary, to ensure your key is not publicly viewable.  
chmod 400 NVKey.pem  
4. Connect to your instance using its Public DNS:  
ec2-174-129-176-60.compute-1.amazonaws.com

**Example:**  
ssh -i "NVKey.pem" ec2-user@ec2-174-129-176-60.compute-1.amazonaws.com

**Note:** In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

Cancel

```
Last login: Sun Apr 30 21:14:30 on ttys002
> cd Desktop
> chmod 400 NVKey.pem
> ssh -i "NVKey.pem" ec2-user@ec2-174-129-176-60.compute-1.amazonaws.com
The authenticity of host 'ec2-174-129-176-60.compute-1.amazonaws.com (174.129.176.60)' can't be established.
ED25519 key fingerprint is SHA256:4KspxbVQuwlqCAokD8A15qCrytNLC3bzOM19yu3sqnA.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-174-129-176-60.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

--| _ _|_
_-| ( / Amazon Linux 2 AMI
---\_|_--|_

https://aws.amazon.com/amazon-linux-2/
-bash: warning: setlocale: LC_CTYPE: cannot change locale (UTF-8): No such file or directory
[ec2-user@ip-172-31-93-107 ~]$ sudo su
[root@ip-172-31-93-107 ~]# 
[root@ip-172-31-93-107 ~]# cd
[root@ip-172-31-93-107 ~]# 
[root@ip-172-31-93-107 ~]# 
[root@ip-172-31-93-107 ~]# 
[root@ip-172-31-93-107 ~]# 
```

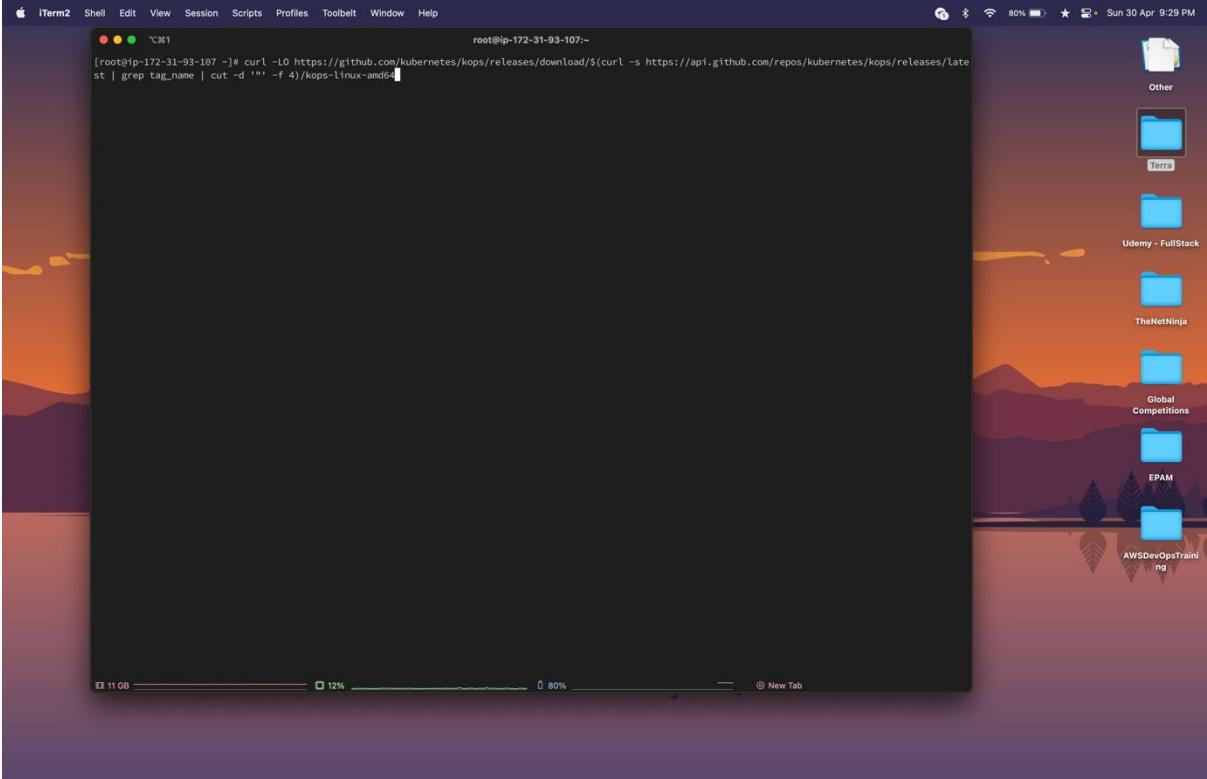
## 6. Install kubectl on instance.

```
Last login: Sun Apr 30 21:14:30 on ttys002
> cd Desktop
> chmod 400 NVKey.pem
> ssh -i "NVKey.pem" ec2-user@ec2-174-129-176-60.compute-1.amazonaws.com
The authenticity of host 'ec2-174-129-176-60.compute-1.amazonaws.com (174.129.176.60)' can't be established.
ED25519 key fingerprint is SHA256:4KspxbVQuwlqCAokD8A15qCrytNLC3bzOM19yu3sqnA.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-174-129-176-60.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

--| _ _|_
_-| ( / Amazon Linux 2 AMI
---\_|_--|_

https://aws.amazon.com/amazon-linux-2/
-bash: warning: setlocale: LC_CTYPE: cannot change locale (UTF-8): No such file or directory
[ec2-user@ip-172-31-93-107 ~]$ sudo su
[root@ip-172-31-93-107 ~]# 
[root@ip-172-31-93-107 ~]# cd
[root@ip-172-31-93-107 ~]# 
[root@ip-172-31-93-107 ~]# 
[root@ip-172-31-93-107 ~]# curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
[root@ip-172-31-93-107 ~]# 
```

## 7. Install kOps on Linux instance.



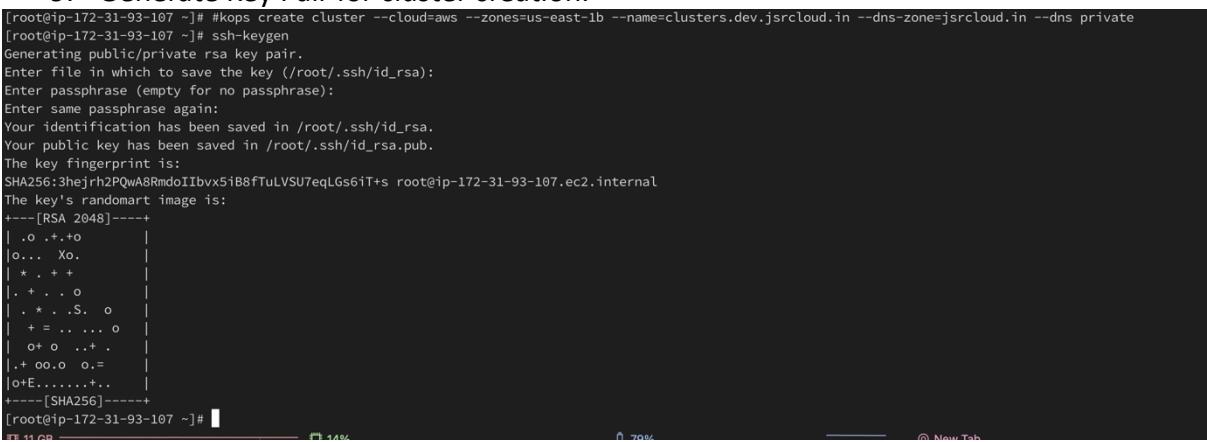
A screenshot of an iTerm2 window on a Mac desktop. The desktop background features a sunset over mountains. The iTerm2 window has a dark theme. In the top bar, it says "iTerm2" and shows the current session name "Terminal". The menu bar includes "File", "Edit", "View", "Session", "Profiles", "Toolbelt", "Window", and "Help". The status bar at the bottom shows "root@ip-172-31-93-107 ~" and the date "Sun 30 Apr 9:29 PM". The main terminal window contains the following command:

```
[root@ip-172-31-93-107 ~]# curl -L https://github.com/kubernetes/kops/releases/download/$(curl -s https://api.github.com/repos/kubernetes/kops/releases/latest | grep tag_name | cut -d '"' -f 4)/kops-linux-amd64
```

## 8. Creating a S3 bucket to store Kubernetes state.

```
[root@ip-172-31-93-107 ~]#
[root@ip-172-31-93-107 ~]# aws s3 mb s3://clusters.dev.jsrcloud.in
make_bucket: clusters.dev.jsrcloud.in
[root@ip-172-31-93-107 ~]#
```

## 9. Generate Key Pair for cluster creation.



```
[root@ip-172-31-93-107 ~]# kops create cluster --cloud=aws --zones=us-east-1b --name=clusters.dev.jsrcloud.in --dns-zone=jsrcloud.in --dns private
[root@ip-172-31-93-107 ~]# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:3hejrh2PQwA8RmdoIIbvxB8fTuLVSVU7eqLGs6iT+ root@ip-172-31-93-107.ec2.internal
The key's randomart image is:
+---[RSA 2048]----+
| .o .+.+o |
|o... Xo. |
| * . + + |
| . + . . o |
| . * . .S. o |
| + = ... ... o |
| o+ o ... . |
|.oo.o o.= |
|o+E.....+.. |
+---[SHA256]----+
[root@ip-172-31-93-107 ~]#
```

Run the kops line again without the '#' in front.

## 10. Run the generated command.

```
Suggestions:
* list clusters with: kops get cluster
* edit this cluster with: kops edit cluster clusters.dev.jsrcloud.in
* edit your node instance group: kops edit ig --name=clusters.dev.jsrcloud.in nodes-us-east-1b
* edit your control-plane instance group: kops edit ig --name=clusters.dev.jsrcloud.in control-plane-us-east-1b

Finally configure your cluster with: kops update cluster --name clusters.dev.jsrcloud.in --yes --admin

[root@ip-172-31-93-107 ~]# kops update cluster --name clusters.dev.jsrcloud.in --yes --admin
```

## 11. Instances will be created. Wait for nodes to get ready.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Publ
control-plane-us-east-1b.masters.clusters.dev.jsrcloud.in	i-00a39d6df1f1aad55	Running	t3.medium	Initializing	No alarms	+ us-east-1b	ec2-44-211-38-62.com...	44.2
nodes-us-east-1b.clusters.dev.jsrcloud.in	i-0ca067ce489c5cf6	Running	t3.medium	Initializing	No alarms	+ us-east-1b	ec2-3-91-208-230.com...	3.91
Controller-Machine	i-035fe522058e1f6f5	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1b	ec2-174-129-176-60.co...	174.
Jenkins	i-0a6c94631e65d2d08	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1b	ec2-44-201-105-30.co...	44.2
Ansible	i-0a98f965ae1db3f96	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1b	ec2-3-80-146-165.com...	3.80

```
[root@ip-172-31-93-107 ~]# kops validate cluster
Using cluster from kubectl context: clusters.dev.jsrcloud.in

Validating cluster clusters.dev.jsrcloud.in

INSTANCE GROUPS
NAME          ROLE      MACHINETYPE   MIN   MAX   SUBNETS
control-plane-us-east-1b   ControlPlane  t3.medium    1     1   us-east-1b
nodes-us-east-1b        Node       t3.medium    1     1   us-east-1b

NODE STATUS
NAME          ROLE      READY
i-00a39d6df1f1aad55  control-plane  True
i-0ca067ce489c5cf6   node       True

Your cluster clusters.dev.jsrcloud.in is ready
```

## 12. Connect to Jenkins Instance via SSH.

```
iTerm2 Shell Edit View Session Scripts Profiles Toolbar Window Help
root@ip-172-31-93-107:~ (ssh) 3x1 ec2-user@ip-172-31-95-99:~ (ssh) 3x2
Last login: Sun Apr 30 21:23:07 on ttys002
> cd Desktop
> ssh -i "MyKey.pem" ec2-user@ec2-44-201-105-30.compute-1.amazonaws.com
The authenticity of host 'ec2-44-201-105-30.compute-1.amazonaws.com (44.201.105.30)' can't be established.
ED25519 key fingerprint is SHA256:E1BpjuUeoZtY19i6mpgA5NEmtvF7KexGGUpesmN.
This key is known by no other name.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-44-201-105-30.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

              _\   _/
             _ \ / /
            ---\ \---\

https://aws.amazon.com/amazon-linux-2/
-bash: warning: setlocale: LC_CTYPE: cannot change locale (UTF-8): No such file or directory
[ec2-user@ip-172-31-95-99:~]$
```

### 13. Install Jenkins w/ JDK onto instance.

```
[root@ip-172-31-95-99 ~]# 
[root@ip-172-31-95-99 ~]# 
[root@ip-172-31-95-99 ~]# sudo yum install jenkins java-1.8.0-openjdk-devel -y
Failed to set locale, defaulting to C
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
No package jenkins available.
Resolving Dependencies
--> Running transaction check
--> Package java-1.8.0-openjdk-devel.x86_64 1:1.8.0.362.b08-1.amzn2.0.1 will be installed
--> Processing Dependency: java-1.8.0-openjdk(x86-64) = 1:1.8.0.362.b08-1.amzn2.0.1 for package: 1:java-1.8.0-openjdk-devel-1.8.0.362.b08-1.amzn2.0.1
--> Processing Dependency: libjvm.so()(64bit) for package: 1:java-1.8.0-openjdk-devel-1.8.0.362.b08-1.amzn2.0.1.x86_64
--> Processing Dependency: libjava.so()(64bit) for package: 1:java-1.8.0-openjdk-devel-1.8.0.362.b08-1.amzn2.0.1.x86_64
--> Processing Dependency: libX11.so.6()(64bit) for package: 1:java-1.8.0-openjdk-devel-1.8.0.362.b08-1.amzn2.0.1.x86_64
--> Running transaction check
```

### 14. Make sure you download the right version of Java for Jenkins.

```
[root@ip-172-31-95-99 ~]# sudo yum install java-11-openjdk
Failed to set locale, defaulting to C
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
No package java-11-openjdk available.
Error: Nothing to do

java-11-openjdk is available in Amazon Linux Extra topic "java-openjdk11"

To use, run
# sudo amazon-linux-extras install java-openjdk11

Learn more at
https://aws.amazon.com/amazon-linux-2/faqs/#Amazon_Linux_Extras

[root@ip-172-31-95-99 ~]# sudo amazon-linux-extras install java-openjdk11
Installing java-11-openjdk
Failed to set locale, defaulting to C
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Cleaning repos: amzn2-core amzn2extra-docker amzn2extra-java-openjdk11 amzn2extra-kernel-5.10
```

### 15. Next run the following commands in the Jenkins Instance Terminal.

#### Long Term Support release

A [LTS \(Long-Term Support\) release](#) is chosen every 12 weeks from the stream of regular releases as the stable release for that time period. It can be installed from the [redhat-stable](#) yum repository.

```
sudo wget -O /etc/yum.repos.d/jenkins.repo \
    https://pkg.jenkins.io/redhat-stable/jenkins.repo
sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
sudo yum upgrade
# Add required dependencies for the jenkins package
sudo yum install java-11-openjdk
sudo yum install jenkins
sudo systemctl daemon-reload
```

BASH | □

### 16. Make sure to change to the right java version. Run the following commands

```
[root@ip-172-31-95-99 ~]# alternatives --config java
There are 2 programs which provide 'java'.

  Selection    Command
  -----
*+ 1          java-1.8.0-openjdk.x86_64 (/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.362.b08-1.amzn2.0.1.x86_64/jre/bin/java)
  2          java-11-openjdk.x86_64 (/usr/lib/jvm/java-11-openjdk-11.0.18.0.10-1.amzn2.0.1.x86_64/bin/java)

Enter to keep the current selection[+], or type selection number: 2
[root@ip-172-31-95-99 ~]# systemctl enable jenkins
Created symlink from /etc/systemd/system/multi-user.target.wants/jenkins.service to /usr/lib/systemd/system/jenkins.service.
[root@ip-172-31-95-99 ~]# systemctl start jenkins
[root@ip-172-31-95-99 ~]#
```

**17. Edit inbound rules for the Jenkins Instance in the following manner:**

Inbound rules [Info](#)

Security group rule ID	Type <a href="#">Info</a>	Protocol <a href="#">Info</a>	Port range <a href="#">Info</a>	Source <a href="#">Info</a>	Description - optional <a href="#">Info</a>
sgr-072fae83dd62ffdc8	SSH	TCP	22	Custom	<input type="text"/> 0.0.0.0 <a href="#">X</a>
sgr-076f8c9607d766c15	HTTP	TCP	80	Custom	<input type="text"/> 0.0.0.0 <a href="#">X</a>
-	Custom TCP	TCP	8080	Anywhere- <a href="#">I...</a>	<input type="text"/> 0.0.0.0 <a href="#">X</a>

Add rule Cancel Preview changes Save rules

**18. Copy Jenkins Instance IPv4 address and paste in a new tab. Add :8080 at the end.**

Getting Started

## Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server:

`/var/lib/jenkins/secrets/initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password

[REDACTED]

**19. Copy the path URL and paste it in the terminal with cat command.**

```
[root@ip-172-31-95-99 ~]# cat /var/lib/jenkins/secrets/initialAdminPassword
f00c5f4a41e640ac94f316d88b73dca9
[root@ip-172-31-95-99 ~]#
```

Copy the password generated. Paste it in the field marked for Administrator Password -> Continue -> Install Suggested Plug-Ins.

20. Fill the details and click on Save and Continue.

The screenshot shows a 'Create First Admin User' form within a Jenkins interface. The fields filled are:

- Username: jsr-cloud
- Password: ..... (redacted)
- Confirm password: ..... (redacted)
- Full name: John Santosh Rudrapogu
- E-mail address: john.s.rudrapogu@gmail.com

At the bottom right, there are two buttons: 'Skip and continue as admin' and a blue 'Save and Continue' button.

Then click on 'Save and Finish'.

21. Manage Jenkins -> Plugin Manager -> Available -> Type 'ssh' -> Check box with Publish over SSH and click on Install without restart....

The screenshot shows the Jenkins Plugin Manager 'Available' tab with the search term 'ssh' applied. The results listed are:

- Publish Over SSH** 1.24 (checked) - Last updated 1 yr 2 mo ago
- Environment Injector** 2.901.v0038b\_6471582 (unchecked) - Last updated 1 mo 28 days ago
- SSH Agent** 333.v878b\_53c89511 (unchecked) - Last updated 2 days 4 hr ago
- DataTables.net API** 1.13.3-3 (unchecked) - Last updated 1 mo 4 days ago
- Jira** 3.9 (unchecked) - Last updated 2 mo 26 days ago
- Slack Notification** 664.vc9a\_90f8b\_c24a... (unchecked) - Last updated 11 days ago

At the bottom, there are three buttons: 'Install without restart' (highlighted), 'Download now and install after restart', and 'Check now'.

22. After restart, login, Manage Jenkins -> Configure System -> Use Crtl F and Serach for Publish ->

Publish over SSH

Jenkins SSH Key [?](#)

Passphrase [?](#)

Path to key [?](#)

Key [?](#)

Disable exec [?](#)

SSH Servers

≡ SSH Server

Name [?](#)  
Jenkins

Hostname [?](#)  
44.201.105.30

Username [?](#)  
root

23. Edit the following directory.

```
[root@ip-172-31-95-99 ~]# passwd root
Changing password for user root.
New password:
BAD PASSWORD: The password fails the dictionary check - it is too simplistic/systematic
Retype new password:
passwd: all authentication tokens updated successfully.
[root@ip-172-31-95-99 ~]# nano /etc/ssh/sshd
[root@ip-172-31-95-99 ~]# nano /etc/ssh/sshd_config
[root@ip-172-31-95-99 ~]#
```

```
#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
# To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication yes
#PermitEmptyPasswords no
PasswordAuthentication yes
```

24. Add the password you gave -> Test Configuration -> Should show Success.

The screenshot shows the Jenkins 'Configure System' page under the 'Manage Jenkins' section. It displays the configuration for an SSH connection to a remote host. The 'Username' field is set to 'root'. The 'Advanced' dropdown is open, showing the 'Use password authentication, or use a different key' checkbox is checked. The 'Passphrase / Password' field contains a redacted password. The 'Path to key' field is empty. A 'Test Configuration' button is visible at the bottom right of the main configuration area.

25. Do the same for Ansible Instance.

```
https://aws.amazon.com/amazon-linux-2/  
-bash: warning: setlocale: LC_CTYPE: cannot change locale (UTF-8): No such file or directory  
[ec2-user@ip-172-31-87-225 ~]$ sudo su  
[root@ip-172-31-87-225 ec2-user]# cd  
[root@ip-172-31-87-225 ~]#  
[root@ip-172-31-87-225 ~]#  
[root@ip-172-31-87-225 ~]# passwd root  
Changing password for user root.  
New password:  
BAD PASSWORD: The password fails the dictionary check - it is too simplistic/systematic  
Retype new password:  
passwd: all authentication tokens updated successfully.  
[root@ip-172-31-87-225 ~]# nano /etc/ssh/sshd_config  
[root@ip-172-31-87-225 ~]# systemctl restart sshd
```

The screenshot shows the Jenkins 'Configure System' page under the 'Manage Jenkins' section. It displays the configuration for an SSH connection to an 'Ansible' instance. The 'Hostname' field is set to '3.80.146.165'. The 'Username' field is set to 'root'. The 'Advanced' dropdown is open, showing the 'SSH Server' section with the 'Name' field set to 'Ansible'. A 'Test Configuration' button is visible at the bottom right of the main configuration area.

```

#SSH configuration
#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# The default is to check both .ssh/authorized_keys and .ssh/authorized_keys2
# but this is overridden so installations will only check .ssh/authorized_keys
AuthorizedKeysFile .ssh/authorized_keys

#AuthorizedPrincipalsFile none

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication yes
#PermitEmptyPasswords no
PasswordAuthentication yes

```

## 26. Carry out the following commands in Ansible Instance SSH. Copy the generated key.

```

[root@ip-172-31-87-225 ~]# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key's randomart image is:
+---[RSA 2048]----+
|       .o.o   |
|       o++...  o |
|       Eoo.o   o o |
|       o.o     . . |
|       . o S    . o . |
|       o o B   . o + . |
|       o B o   . . = . |
|       ooo     .++ o |
|       ++.     o=0= |
+---[SHA256]----+
[root@ip-172-31-87-225 ~]# cd .ssh
[root@ip-172-31-87-225 .ssh]# ls
authorized_keys  id_rsa  id_rsa.pub
[root@ip-172-31-87-225 .ssh]# cat id_rsa.pub
ssh-rsa AAAAB3nzaC1yc2EAAQABAAAQCVyCx60tssA93LrYay7dhPWi5nbUy5FLVCN6UH6vNRAS7dvcFs4imy30Axka9G/+l0e98/wSrZQauqajNvm0jIiyGTGgDfi4JLf7ZbiRVJr1qSkQhbqHZ3Zn
bjmaJVP771lnic5SzBq8Ducc6q5rlLL6xTTwce8RB8a1v9ISqA9AY4zIlWcKXDeWlUrZAzVxeEVohrvHVfEawkDeJ8vLkqFxV2yiMdcOsKX0jMwiJwUJ5jFk00Dye7CypAEEE9wL4NrZ3lnXjc0F8nVtr77D
qitIdPSCnTej09JB5KEPE6ukLSh6ZkQst5Bzb7jq4ocNaVAK1b3DXXdhH9qhemFeJ root@ip-172-31-87-225.ec2.internal
[root@ip-172-31-87-225 .ssh]# nano authorized_keys

```

## 27. In the Controller-Machine SSH -> cd .ssh -> ls -> nano authorized\_keys -> Paste the generated key in the next available line and Crtl O -> Enter -> Crtl X.

```

[root@ip-172-31-93-107 ~]# cd .ssh
[root@ip-172-31-93-107 .ssh]# ls
authorized_keys  id_rsa  id_rsa.pub  known_hosts
[root@ip-172-31-93-107 .ssh]# nano authorized_keys
[root@ip-172-31-93-107 .ssh]#

```

**28. Try and connect to the Controller-Machine Instance [Private IPv4] from the Ansible Instance.**

```
[root@ip-172-31-87-225 .ssh]# ssh root@174.129.176.60
The authenticity of host '174.129.176.60 (174.129.176.60)' can't be established.
ECDSA key fingerprint is SHA256:XrxtfeMsFBpK0ce8/mCiRdPKy+E3WTh0zYeLxt1hWEU.
ECDSA key fingerprint is MD5:84:b9:8d:a8:76:58:11:ca:47:fc:77:60:76:8b:31:d2.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '174.129.176.60' (ECDSA) to the list of known hosts.
Last login: Sun Apr 30 17:16:55 2023

__|__|_
_|(   /  Amazon Linux 2 AMI
___\___|___|
```

Try and connect to the private IPv4 address of Controller-Machine from Jenkins Instance.

**29. Install Docker from Ansible Instance that is hosting the Controller-Machine Instance.**

```
[root@ip-172-31-93-107 ~]# yum install docker
Failed to set locale, defaulting to C
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Resolving Dependencies
> Running transaction check
```

**30. Start and enable docker**

```
[root@ip-172-31-90-74 ~]# systemctl start docker
[root@ip-172-31-90-74 ~]# systemctl enable docker
Created symlink from /etc/systemd/system/multi-user.target.wants/docker.service to /usr/lib/systemd/system/docker.service.
```

Docker Login -> cd /opt/ -> add the three files.

```
Login Succeeded
[root@ip-172-31-90-74 ~]# cd /opt/
[root@ip-172-31-90-74 opt]# ls
aws  containerd  rh
[root@ip-172-31-90-74 opt]# nano ansible.yml
[root@ip-172-31-90-74 opt]# nano deployment.yml
[root@ip-172-31-90-74 opt]#
[root@ip-172-31-90-74 opt]# nano service.yml
[root@ip-172-31-90-74 opt]#
```

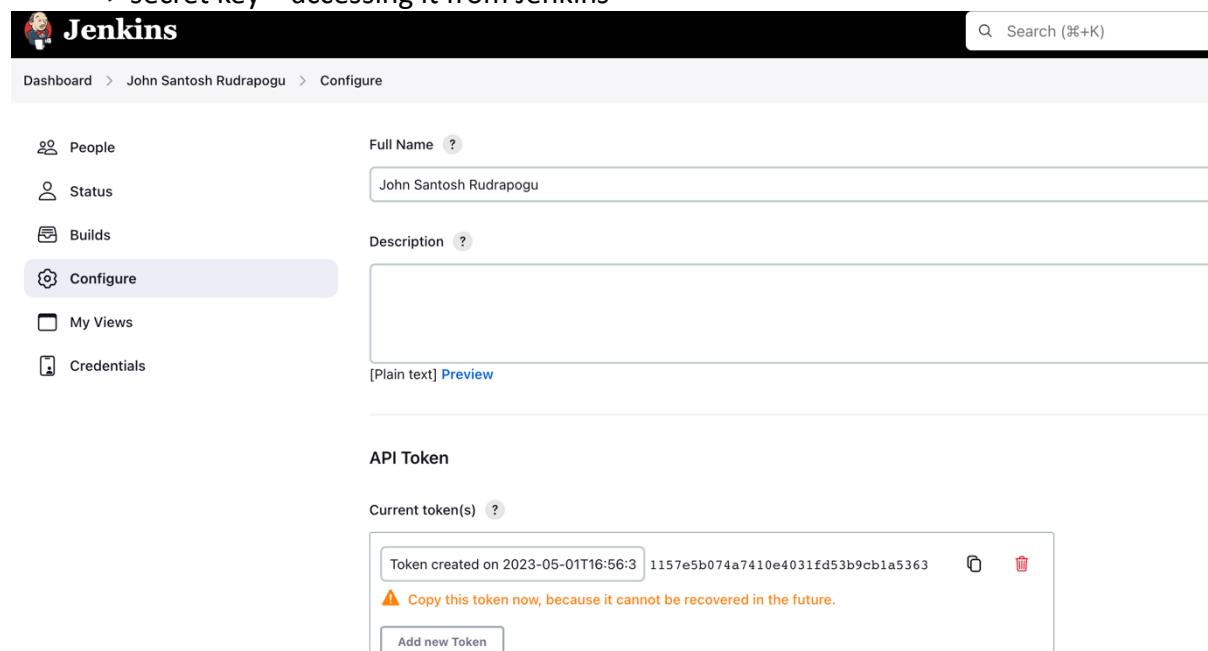
### 31. Install Ansible.

```
[root@ip-172-31-90-74 opt]# cd  
[root@ip-172-31-90-74 ~]# yum install ansible  
Failed to set locale, defaulting to C  
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd  
amzn2-core  
No package ansible available.  
Error: Nothing to do  
  
ansible is available in Amazon Linux Extra topic "ansible2"  
  
To use, run  
# sudo amazon-linux-extras install ansible2  
  
Learn more at  
https://aws.amazon.com/amazon-linux-2/faqs/#Amazon\_Linux\_Extras  
  
[root@ip-172-31-90-74 ~]# sudo amazon-linux-extras install ansible2  
Installing ansible  
Failed to set locale, defaulting to C  
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
```

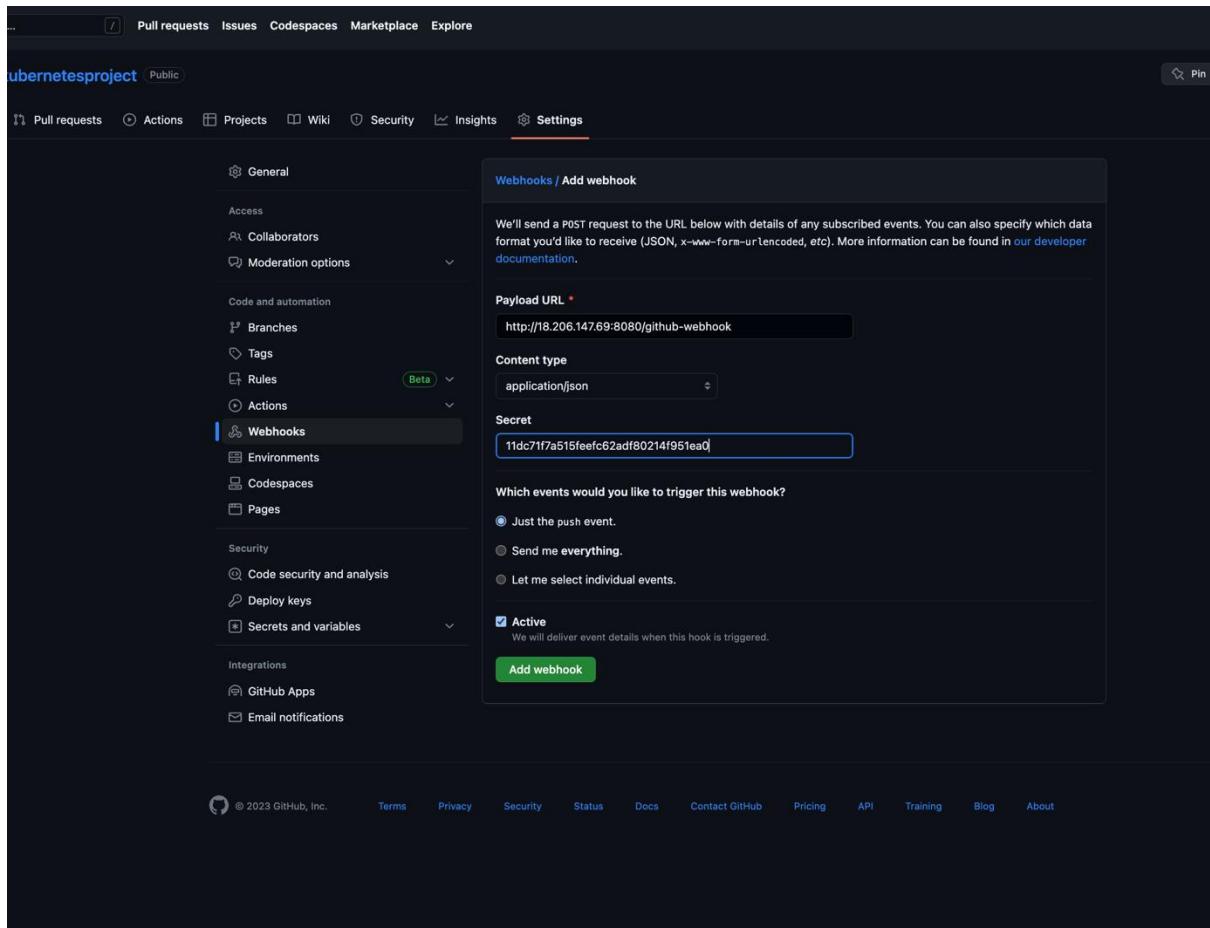
Edit the following file after installation. -> Add [kube] and Controller-Machine private IPv4 address.

```
[root@ip-172-31-90-74 ~]# nano /etc/ansible/hosts
```

### 32. Install Git on Jenkins Instance. Create a new repository with Dockerfile in it. Add Webhooks by going to settings and adding Jenkins Webpage URL -> application/json -> secret key – accessing it from Jenkins



The screenshot shows the Jenkins 'Configure' page for the user 'John Santosh Rudrapogu'. The 'Configure' tab is active. On the left, there are navigation links: People, Status, Builds, Configure (selected), My Views, and Credentials. The main area has two sections: 'Full Name' (set to 'John Santosh Rudrapogu') and 'Description' (empty). Below these is the 'API Token' section. It shows a single token created on 2023-05-01T16:56:3 with the ID 1157e5b074a7410e4031fd53b9cb1a5363. A warning message states: '⚠️ Copy this token now, because it cannot be recovered in the future.' There are 'Copy' and 'Delete' buttons next to the token ID.



Apply and save in Jenkins -> then add Webhook in github. You should get green tick.

### 33. Create new item in Jenkins.

A screenshot of the Jenkins 'New Item' creation interface. The first step is to 'Enter an item name', with 'kubernetesproject' entered into the required field. Below this, two project types are listed: 'Freestyle project' and 'Pipeline'. The 'Freestyle project' section includes a description: 'This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.' The 'Pipeline' section includes a description: 'Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as'.

### 34. Configure -> Source Management Repository -> Git -> Build Triggers -> Github hook trigger for GITScm polling -> Build steps -> Send files or execute commands over SSH:

Send files or execute commands over SSH ?

SSH Publishers

SSH Server Name ? Jenkins

Advanced ▾

Transfers

Transfer Set Source files ?

Remove prefix ?

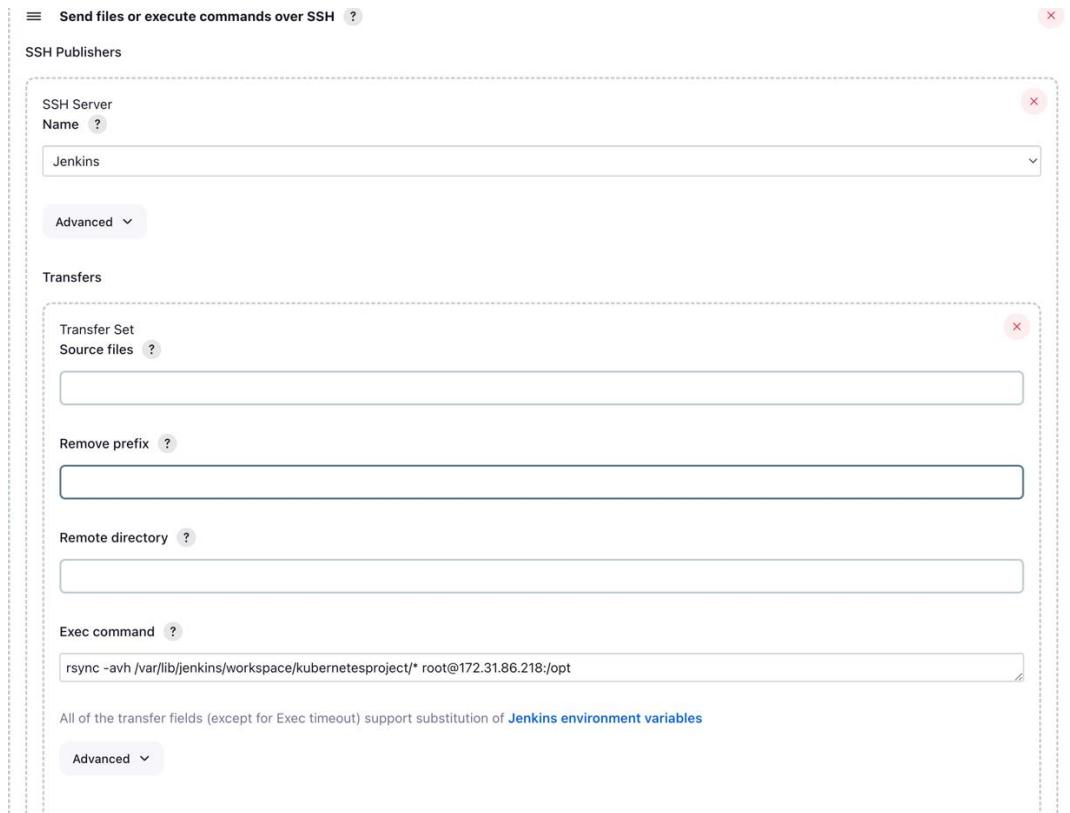
Remote directory ?

Exec command ?

```
rsync -avh /var/lib/jenkins/workspace/kubernetesproject/* root@172.31.86.218:/opt
```

All of the transfer fields (except for Exec timeout) support substitution of [Jenkins environment variables](#)

Advanced ▾



Next, choose the same option and select Ansible and make the following changes:

Send files or execute commands over SSH ?

SSH Publishers

SSH Server Name ? Ansible

Advanced ▾

Transfers

Transfer Set Source files ?

Either Source files, Exec command or both must be supplied

Remove prefix ?

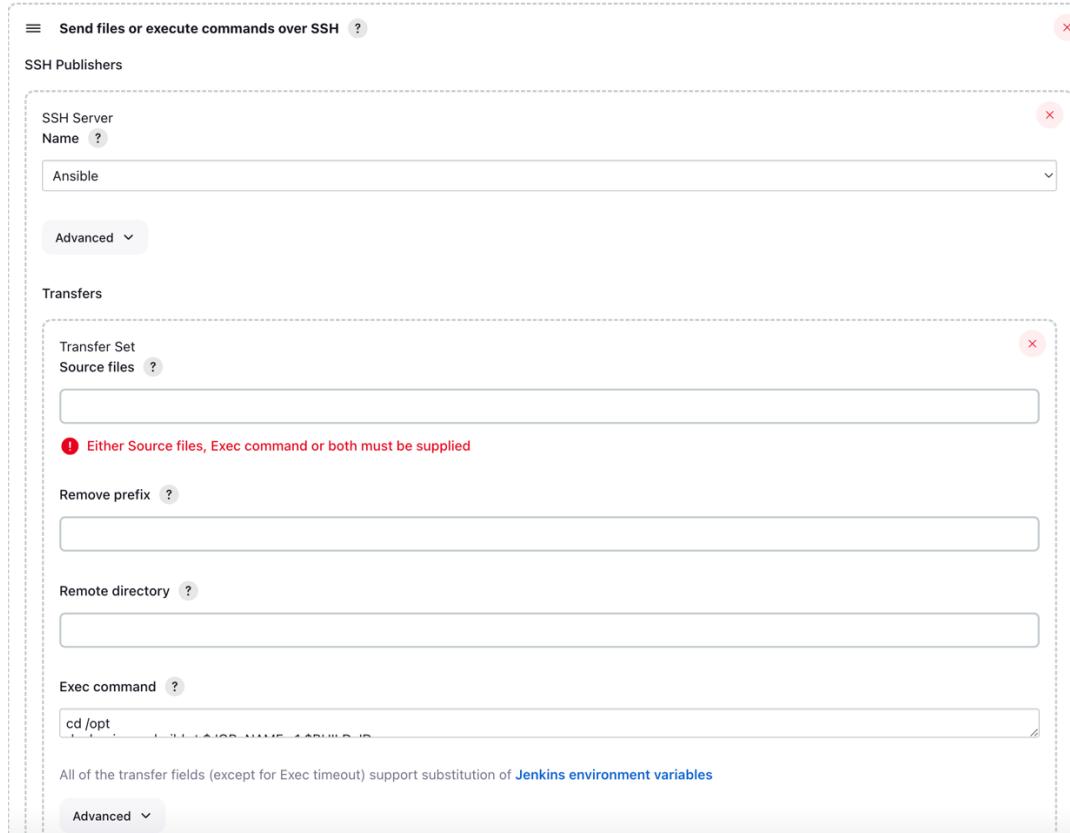
Remote directory ?

Exec command ?

```
cd /opt
```

All of the transfer fields (except for Exec timeout) support substitution of [Jenkins environment variables](#)

Advanced ▾



```

cd /opt
docker image build -t $JOB_NAME:v1.$BUILD_ID .
docker image tag $JOB_NAME:v1.$BUILD_ID john31237/$JOB_NAME:v1.$BUILD_ID
docker image tag $JOB_NAME:v1.$BUILD_ID john31237/$JOB_NAME:latest
docker image push john31237/$JOB_NAME:v1.$BUILD_ID
docker image push john31237/$JOB_NAME:latest
docker image rmi $JOB_NAME:v1.$BUILD_ID john31237/$JOB_NAME:v1.$BUILD_ID
john31237/$JOB_NAME:latest.

```

### 35. Post build actions -> Send build artifacts over SSH:

The screenshot shows the Jenkins 'Send build artifacts over SSH' configuration page. It includes sections for 'SSH Publishers' and 'Transfers'. In the 'Transfers' section, the 'Source files' field is highlighted with a red border and contains the error message 'Either Source files, Exec command or both must be supplied'. The 'Exec command' field contains the value 'ansible-playbook /opt/ansible.yml'.

### Apply and Save

#### 36. Build Now->

```

TASK [create service] *****
changed: [172.31.90.157]

PLAY RECAP *****
172.31.90.157 : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

SSH: EXEC: completed after 5,604 ms
SSH: Disconnecting configuration [Ansible] ...
SSH: Transferred 0 file(s)
Finished: SUCCESS

```

✓ #11  
| May 5, 2023, 1:04 PM

! #10  
| May 5, 2023, 1:01 PM

### 37. Go to Controller-Machine Instance and run the command -> kubectl get all ->

```
[root@ip-172-31-90-157 ~]# kubectl get all
NAME                           READY   STATUS    RESTARTS   AGE
pod/hello-app-5b4d6c5c96-tzm75  1/1     Running   0          32s
pod/hello-app-5b4d6c5c96-zsmkn  1/1     Running   0          32s

NAME                  TYPE        CLUSTER-IP      EXTERNAL-IP      PORT(S)        AGE
service/hello-app    NodePort    100.68.121.21  <none>         8080:32000/TCP  31s
service/kubernetes   ClusterIP   100.64.0.1    <none>         443/TCP       8h

NAME                  READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/hello-app  2/2     2           2           32s

NAME                  DESIRED   CURRENT   READY   AGE
replicaset.apps/hello-app-5b4d6c5c96  2         2         2       32s
[root@ip-172-31-90-157 ~]#
```

### 38. Go to EC2 Instance -> copy Slave Node Instance IPv4 address and paste it along with :32000 and run the URL :

