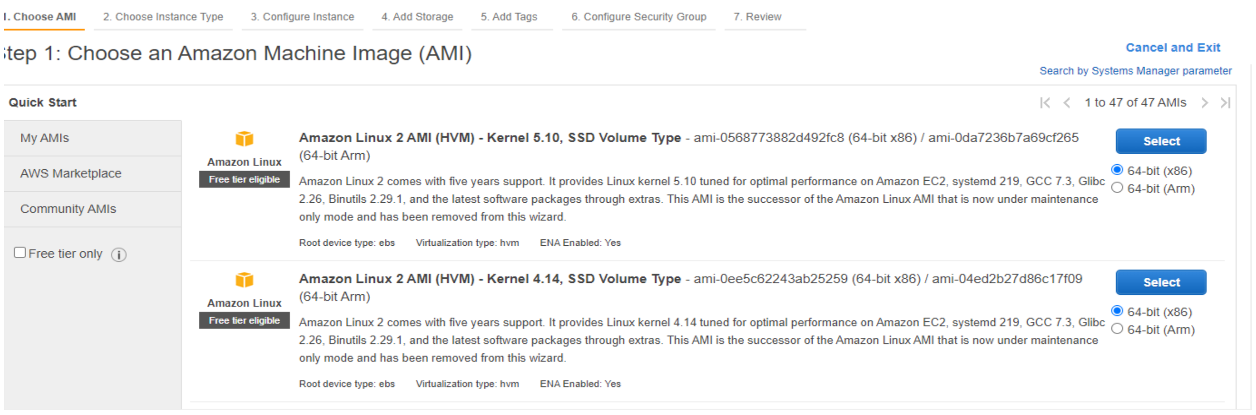
React JS Project Automation Process Jenkins with Docker

Step:- 1

Launch an T3 medium EC2 instance for install Node Js+Jenkins+Git



1. Choose AMI | 2. Choose Instance Type | 3. Configure Instance | 4. Add Storage | 5. Add Tags | 6. Configure Security Group | 7. Review

Step 1: Choose an Amazon Machine Image (AMI)

Cancel and Exit

Search by Systems Manager parameter

Quick Start — 1 to 47 of 47 AMIs

My AMIs | AWS Marketplace | Community AMIs | Free tier only

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type - ami-0568773882d492fc8 (64-bit x86) / ami-0da7236b7a69cf265 (64-bit Arm)

Amazon Linux 2 comes with five years support. It provides Linux kernel 5.10 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is now under maintenance only mode and has been removed from this wizard.

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

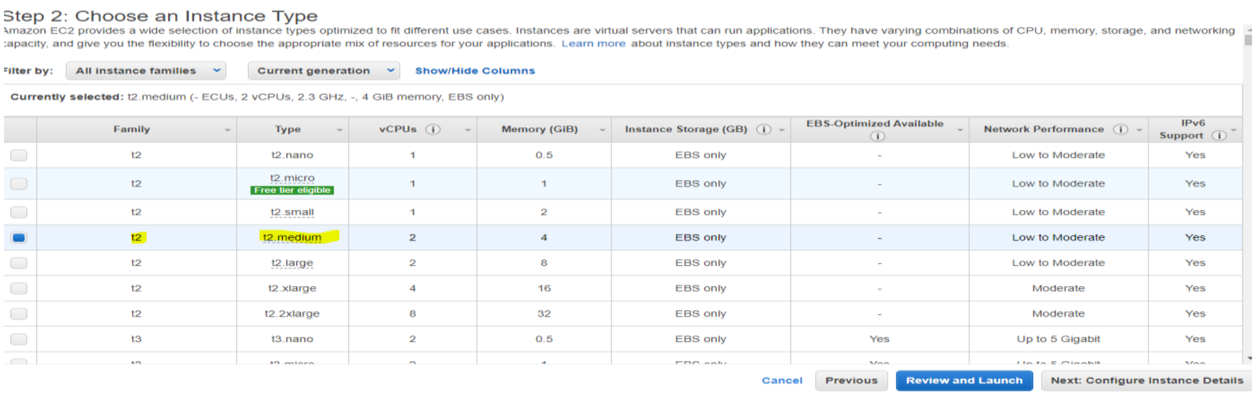
Amazon Linux — Free tier eligible — Select — [✓ 64-bit (x86)] 64-bit (Arm)

Amazon Linux 2 AMI (HVM) - Kernel 4.14, SSD Volume Type - ami-0ee5c62243ab25259 (64-bit x86) / ami-04ed2b27d86c17f09 (64-bit Arm)

Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is now under maintenance only mode and has been removed from this wizard.

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

Amazon Linux — Free tier eligible — Select — [✓ 64-bit (x86)] 64-bit (Arm)



Step 2: Choose an Instance Type

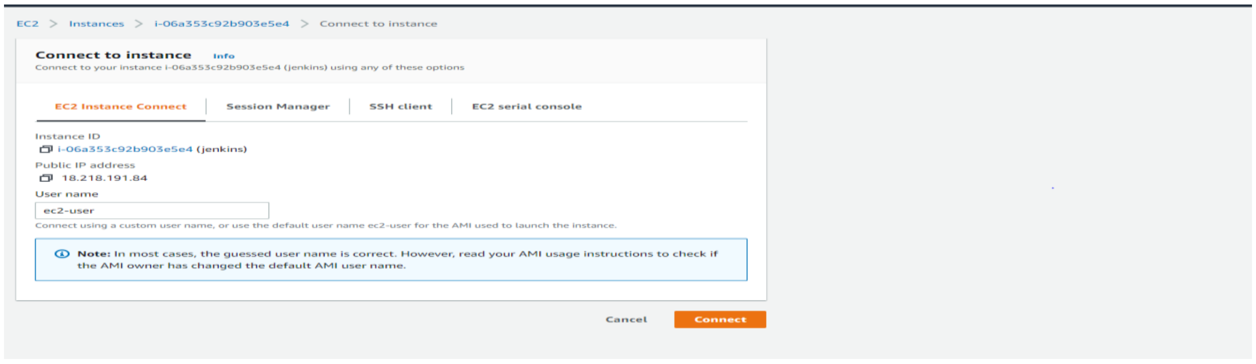
Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.

Filter by: All instance families | Current generation | Show/Hide Columns

Currently selected: t2.medium (- ECUs, 2 vCPUs, 2.3 GHz, -, 4 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
	t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
	t2	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
	t2	t2.small	1	2	EBS only	-	Low to Moderate	Yes
✓	t2	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
	t2	t2.large	2	8	EBS only	-	Low to Moderate	Yes
	t2	t2.xlarge	4	16	EBS only	-	Moderate	Yes
	t2	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
	t3	t3.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes

Cancel | Previous | Review and Launch | Next: Configure Instance Details



EC2 > Instances > i-06a353c92b903e5e4 > Connect to instance

Connect to instance Info

Connect to your instance i-06a353c92b903e5e4 (jenkins) using any of these options

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

Instance ID
i-06a353c92b903e5e4 (jenkins)

Public IP address
18.218.191.84

User name
ec2-user

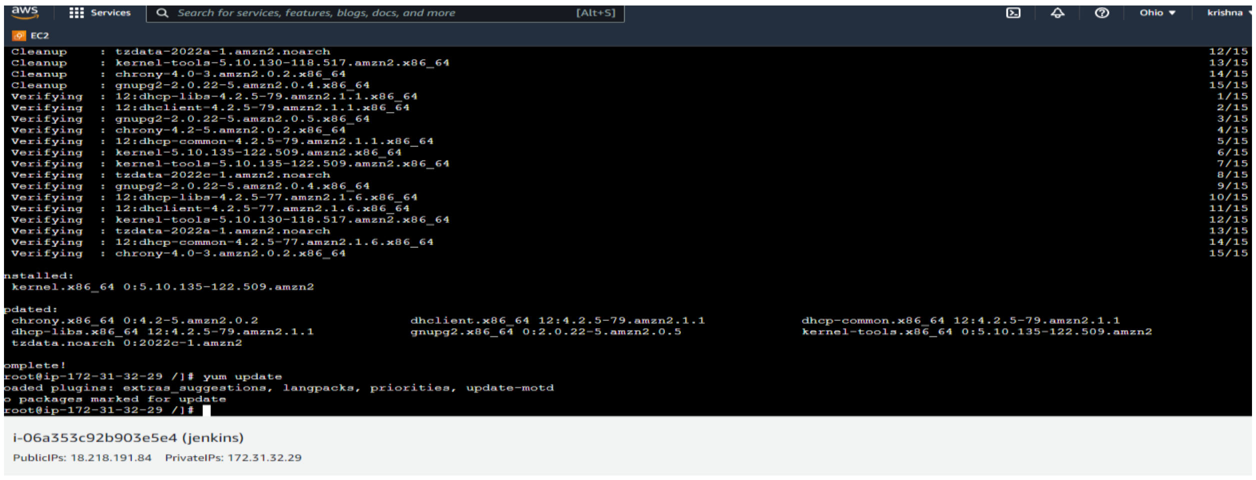
Connect using a custom user name, or use the default user name ec2-user for the AMI used to launch the instance.

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 | Connect

1.Sudo -I

2. yum update

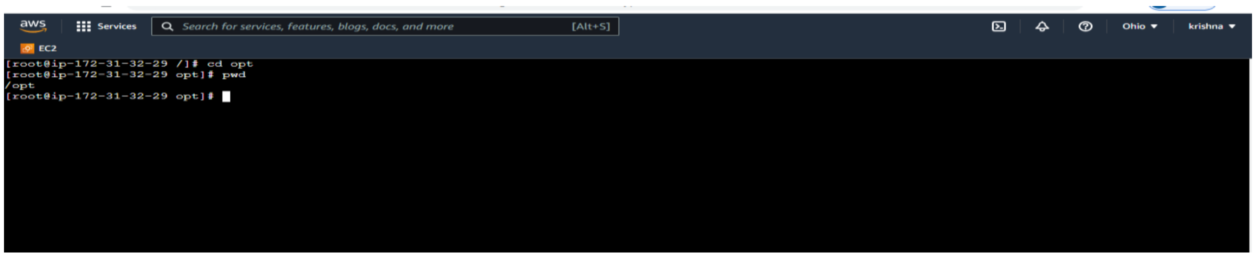


```
 Cleanup    : tzdata-2022a-1.amzn2.noarch                                                                                12/15
 Cleanup    : kernel-tools-5.10.130-118.517.amzn2.x86_64                                                                 13/15
 Cleanup    : chrony-4.0-3.amzn2.0.2.x86_64                                                                              14/15
 Cleanup    : gnupg2-2.0.22-5.amzn2.0.4.x86_64                                                                           15/15
 Verifying  : 12:dhcp-libs-4.2.5-79.amzn2.1.1.x86_64                                                                      1/15
 Verifying  : 12:dhclient-4.2.5-79.amzn2.1.1.x86_64                                                                       2/15
 Verifying  : gnupg2-2.0.22-5.amzn2.0.5.x86_64                                                                            3/15
 Verifying  : chrony-4.2-5.amzn2.0.2.x86_64                                                                               4/15
 Verifying  : 12:dhcp-common-4.2.5-79.amzn2.1.1.x86_64                                                                    5/15
 Verifying  : kernel-5.10.135-122.509.amzn2.x86_64                                                                        6/15
 Verifying  : kernel-tools-5.10.135-122.509.amzn2.x86_64                                                                  7/15
 Verifying  : tzdata-2022c-1.amzn2.noarch                                                                                 8/15
 Verifying  : gnupg2-2.0.22-5.amzn2.0.4.x86_64                                                                            9/15
 Verifying  : 12:dhcp-libs-4.2.5-77.amzn2.1.6.x86_64                                                                     10/15
 Verifying  : 12:dhclient-4.2.5-77.amzn2.1.6.x86_64                                                                      11/15
 Verifying  : kernel-tools-5.10.130-118.517.amzn2.x86_64                                                                 12/15
 Verifying  : tzdata-2022a-1.amzn2.noarch                                                                                13/15
 Verifying  : 12:dhcp-common-4.2.5-77.amzn2.1.6.x86_64                                                                   14/15
 Verifying  : chrony-4.0-3.amzn2.0.2.x86_64                                                                              15/15

nstalled:
 kernel.x86_64 0:5.10.135-122.509.amzn2

pdated:
 chrony.x86_64 0:4.2-5.amzn2.0.2                  dhclient.x86_64 12:4.2.5-79.amzn2.1.1            dhcp-common.x86_64 12:4.2.5-79.amzn2.1.1
 dhcp-libs.x86_64 12:4.2.5-79.amzn2.1.1           gnupg2.x86_64 0:2.0.22-5.amzn2.0.5               kernel-tools.x86_64 0:5.10.135-122.509.amzn2
 tzdata.noarch 0:2022c-1.amzn2

omplete!
root@ip-172-31-32-29 /]# yum update
oaded plugins: extras_suggestions, langpacks, priorities, update-motd
o packages marked for update
root@ip-172-31-32-29 /]#
```

i-06a353c92b903e5e4 (jenkins)

PublicIPs: 18.218.191.84   PrivateIPs: 172.31.32.29

3.cd /opt



```
[root@ip-172-31-32-29 /]# cd opt
[root@ip-172-31-32-29 opt]# pwd
/opt
[root@ip-172-31-32-29 opt]#
```

4. yum install java-1.8*

5. Java -version

6. Yum update

7.wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -

8.sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'

9. yum update

10.yum install Jenkins

11. systemctl start jenkins

12.systemctl status jenkins

13. systemctl enable jenkins

14. Cat /var/lib/jenkins/secrets/initialAdminPassword

15. Ec2 Instance public ip:8080 type in browser

------------------------------------ Java Node Js Git Integration--
-

Step:2

- **Configure Jenkins**
- **The default Username is admin**
- **Grab the default password**
- **Password Location:/var/lib/jenkins/secrets/initialAdminPassword**
- **Skip Plugin Installation; We can do it laterss**
- **Change admin password**
- **Admin > Configure > Password**
- **Configure java path**
- **Manage Jenkins > Global Tool Configuration > JDK**

- **Install Node Js plugin without restart**
- **Manage Jenkins > Jenkins Plugins > available > Node Js**
- **(Update) Install "Node Js Integration" Plugin as well**
- **Install Node Js Integration Plugin without restart**
- **Manage Jenkins > Jenkins Plugins > available > Node Js Integration**
- **Manage Jenkins > Global Tool Configuration > Node Js**

- **Install git plugin without restart**
- **Manage Jenkins > Jenkins Plugins > available > github**
- **Configure git path**
- **Manage Jenkins > Global Tool Configuration > git**

Login to Jenkins console and add Docker server to execute commands from Jenkins
Manage Jenkins --> Configure system --> Publish over SSH --> add Docker server and Credentials

Install "publish Over SSH"

- **Manage Jenkins > Manage Plugins > Available > Publish over SSH**
- **Manage Jenkins > Configure System > Publish Over SSH > SSH Servers**
  - **SSH Servers:**

- Hostname:<ServerIP>
- username: sivaram
- password: *******

Test the connection "Test Connection"

-- Create Jenkins Job--

Step 3 - Create the Free Style Node JS project .

Git URL - https://github.com/node-js-sample.git

BUILD - npm install

```
npm run build

tar czf Node.tar.gz node_modules package.json public package-lock.json build src
```

Step - Post Build Action Define below details.

Source file : **/*.gz

Exec Command

```
mv /home/dockeradmin/Node.tar.gz  Node.tar.gz;

cd /home/dockeradmin;

tar -xf Node.tar.gz;

docker stop raju;

docker rm raju;

docker image rm raju;

docker build -t raju .

docker run --name raju -itd -p 3000:3000 raju
```

Docker File in docker server

1. Login to Docker host and check images and containers. (no images and containers)
2. Execute Jenkins job
3. check images and containers again on Docker host. This time an image and container get creates through Jenkins job
4. Access web application from browser which is running on container

<docker_host_Public_IP>:3000

***Access web application from browser which is running on container**

**** <docker_host_Public_IP>:3000**

--

Docker file by using jenkins process

-> Launch an EC2 instance for Docker host

***Install docker on EC2 instance and start services**

```
1.yum install docker -y
2.systemctl start docker
3. systemctl enable docker
```

-->create a new user for Docker management and add him to Docker (default) group

```
useradd dockeradmin
passwd dockeradmin
  usermod -aG docker dockeradmin
```

--->*Write a Docker file under /opt/docker

```
mkdir /opt/docker
### vi Dockerfile
```

```
FROM node:16.15.0
WORKDIR app
COPY . .
EXPOSE 443
EXPOSE 80
COPY ./package.json /app
RUN npm install
RUN npm run build

ENV PORT 3000

EXPOSE 3000

ENTRYPOINT ["npm", "run","start"]
```

***vi etc/ssh/sshd_cofig**

```
# To disable tunneled clear text passwords, change to no here!

PasswordAuthentication yes
```

***Login to Jenkins console and add Docker server to execute commands from Jenkins**

***Manage Jenkins --> Configure system --> Publish over SSH --> add Docker server and credentials**

permission denied:-

```
chown -R dockeradmin:dockeradmin /opt/docker
```