

## Creating Ec2 machine from AWS.

Name

Capstone-Project

Add additional tags

▼ Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Q Search our full catalog including 1000s of application and OS images

Recents

Quick Start

Amazon Linux

aws

macOS

Mac

Ubuntu

ubuntu

Windows

Microsoft

Red Hat

Red Hat

S

Q

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Number of instances Info

1

Software Image (AMI)

Amazon Linux 2023 AMI 2023.0.2...read more  
ami-08333bccc35d71140

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the

Cancel

Launch instance

Review commands

## Connected the EC2 Via SSH

[illegible]

## Switch the User and installing Git

```
[root@ip-172-31-35-176 ~]# sudo yum install git -y
Last metadata expiration check: 0:03:53 ago on Tue May  9 05:53:00 2023.
Dependencies resolved.
```

Package	Architecture
Installing:	
git	x86_64
Installing dependencies:	
git-core	x86_64
git-core-doc	noarch
perl-Error	noarch
perl-File-Find	noarch
perl-Git	noarch
perl-TermReadKey	x86_64
perl-lib	x86_64

Transaction Summary

Install 8 Packages

## Clone the code from github

```
[root@ip-172-31-35-176 ~]# sudo git clone https://github.com/rvsp/reactjs-demo.git
Cloning into 'reactjs-demo'...
remote: Enumerating objects: 40, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 40 (delta 2), reused 2 (delta 2), pack-reused 36
Receiving objects: 100% (40/40), 175.93 KiB | 3.52 MiB/s, done.
Resolving deltas: 100% (11/11), done.
[root@ip-172-31-35-176 ~]# cd reactjs-demo
```

Dockerfile created in reactjs-demo folder

```
[root@ip-172-31-35-176 reactjs-demo]# cat dockerfile

# Use an official Node.js runtime as a parent image
FROM node:14-alpine

# Set the working directory to /app
WORKDIR /app

# Copy package.json and package-lock.json to the working directory
COPY package*.json ./

# Install dependencies
RUN npm install

# Copy the rest of the application files to the working directory
COPY . .

# Build the application
RUN npm run build

# Expose port 80 for the application
EXPOSE 80

# Start the application using the production server
CMD [ "npm", "start" ]
```

Docker-compose file created as below

```
[root@ip-172-31-35-176 reactjs-demo]# cat docker-compose.yml

version: '3'

services:
  app:
    image: my-react-app
    ports:
      - "80:80"
```

## Image created and tagged

```
[Docker] $ /bin/sh -xe /tmp/jenkins7371394953029846646.sh
+ docker images
REPOSITORY    TAG       IMAGE ID      CREATED       SIZE
newapp        V1.2      3352450df187  7 hours ago  143MB
newapp        V1.1      e0a1e8a351f2  7 hours ago  143MB
[Docker] $ /bin/sh -xe /tmp/jenkins3297053389929006449.sh
+ docker ps -a
CONTAINER ID   IMAGE      COMMAND      CREATED   STATUS    PORTS     NAMES
+ docker tag newapp:V1.2 raghavanp08/dev:latest
+ docker images
REPOSITORY    TAG       IMAGE ID      CREATED       SIZE
newapp        V1.2      3352450df187  7 hours ago  143MB
raghavanp08/dev latest    3352450df187  7 hours ago  143MB
newapp        V1.1      e0a1e8a351f2  7 hours ago  143MB
Finished: SUCCESS
```

Navigate to Cd Reactjs-demo and installing the dependencies

```
[root@ip-172-31-35-176 ~]# ls
reactjs-demo
[root@ip-172-31-35-176 ~]# cd reactjs-demo/
[root@ip-172-31-35-176 reactjs-demo]# npm install
npm WARN deprecated chokidar@2.1.8: Chokidar 2 does not receive s
npm WARN deprecated popper.js@1.16.1: You can find the new Popper
added 6 packages, and audited 1652 packages in 13s

65 packages are looking for funding
  run `npm fund` for details

76 vulnerabilities (1 low, 26 moderate, 36 high, 13 critical)

To address issues that do not require attention, run:
  npm audit fix

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
```

And NPM Build

```
[root@ip-172-31-35-176 reactjs-demo]# npm run build
> react-front-end@0.1.0 build
> react-scripts build

Creating an optimized production build...
Error: error:0308010C:digital envelope routines::unsupported
    at new Hash (node:internal/crypto/hash:71:19)
    at Object.createHash (node:crypto:133:10)
    at module.exports (/root/reactjs-demo/node_modules/webpack/lib/util/createHash.js:135:53)
    at NormalModule._initBuildHash (/root/reactjs-demo/node_modules/webpack/lib/NormalModule.js:417:16)
    at handleParseError (/root/reactjs-demo/node_modules/webpack/lib/NormalModule.js:471:10)
    at /root/reactjs-demo/node_modules/webpack/lib/NormalModule.js:503:5
    at /root/reactjs-demo/node_modules/webpack/lib/NormalModule.js:358:12
    at /root/reactjs-demo/node_modules/loader-runner/lib/LoaderRunner.js:373:3
    at iterateNormalLoaders (/root/reactjs-demo/node_modules/loader-runner/lib/LoaderRunner.js:214:10)
    at iterateNormalLoaders (/root/reactjs-demo/node_modules/loader-runner/lib/LoaderRunner.js:221:10)
    at /root/reactjs-demo/node_modules/react-scripts/scripts/build.js:19
    at throw err;
    at ^
Error: error:0308010C:digital envelope routines::unsupported
    at new Hash (node:internal/crypto/hash:71:19)
    at Object.createHash (node:crypto:133:10)
    at module.exports (/root/reactjs-demo/node_modules/webpack/lib/util/createHash.js:135:53)
    at NormalModule._initBuildHash (/root/reactjs-demo/node_modules/webpack/lib/NormalModule.js:417:16)
    at /root/reactjs-demo/node_modules/webpack/lib/NormalModule.js:452:10
    at /root/reactjs-demo/node_modules/webpack/lib/NormalModule.js:323:13
    at /root/reactjs-demo/node_modules/loader-runner/lib/LoaderRunner.js:367:11
    at /root/reactjs-demo/node_modules/loader-runner/lib/LoaderRunner.js:233:18
    at context.callback (/root/reactjs-demo/node_modules/loader-runner/lib/LoaderRunner.js:111:13)
    at /root/reactjs-demo/node_modules/babel-loader/lib/index.js:59:103 {
  opensslErrorStack: [ 'error:03000086:digital envelope routines::initialization error' ],
  library: 'digital envelope routines',
  reason: 'unsupported',
  code: 'ERR_OSSL_EVP_UNSUPPORTED'
}
```

Installed Apache to host this in web

```

root@ip-172-31-20-197:~/reactjs-demo# sudo apt-get install apache2 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.3-0 mailcap mime-support ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.3-0 mailcap mime-support ssl-cert
0 upgraded, 12 newly installed, 0 to remove and 29 not upgraded.
Need to get 2102 kB of archives.
After this operation, 8389 kB of additional disk space will be used.
Get:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libapr1 amd64 1.7.0-8ubuntu0.22.04.1 [108 kB]
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1 amd64 1.6.1-5ubuntu4.22.04.1 [92.6 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1-dbd-sqlite3 amd64 1.6.1-5ubuntu4.22.04.1 [11.3 kB]
Get:4 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1-ldap amd64 1.6.1-5ubuntu4.22.04.1 [9168 B]
Get:5 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 liblua5.3-0 amd64 5.3.6-1bui1d1 [140 kB]
Get:6 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-bin amd64 2.4.42-1ubuntu4.5 [1245 kB]

```

File copied to /var/www/html

```
root@ip-172-31-20-197:~/reactjs-demo# cp -r build/* /var/www/html/
root@ip-172-31-20-197:~/reactjs-demo# ls
Dockerfile-multistage  README.md  build  node_modules  package-lock.json  package.json  public  src
```

## Port open from AWS console

<input type="checkbox"/>	-	sgr-0db73ebb2ed877...	IPv4	HTTP	TCP	80
--------------------------	---	-----------------------	------	------	-----	----

Final Output from Web,

Deployment url: <http://3.12.103.205>


Not secure | 3.12.103.205


Manager - Home - Workday - Reports - Microsoft... - Compliance Manag... - Microsoft Office Ho... - My Account - https://prudentplus... - Java Downloads | O... - SSL VPN Service - VirtualWorks™


#	Name
---	------

## Docker images pushed to public and private repo

## 1. Private repo

 **raghavanp08 / prod**

**Description**  
This repository does not have a description 

 Last pushed: a minute ago

**Docker commands**  
To push a new tag to this repository,  

```
docker push raghavanp08/prod:tagname
```

**Tags**  
This repository is empty. When it's not empty, you'll see a list of the most recent tags here.


**Automated Builds**  
Manually pushing images to Hub? Connect your account to Bitbucket to automatically build and tag new images whenever code is updated, so you can focus your time on creating.  
Available with Pro, Team and Business subscriptions. [Read automated builds](#).  
[Upgrade](#)


## 2 . Public repo


raghavanp08 > Repositories > dev > **General**

Using 1 of 1 private repositories. [Get](#)

**General** Tags Builds Collaborators Webhooks Settings

 **raghavanp08 / dev**

**Description**  
for project 

 Last pushed: 4 minutes ago

**Docker commands**  
To push a new tag to this repository,  

```
docker push raghavanp08/dev:tagname
```

[Public View](#)

**Tags**  
This repository is empty. When it's not empty, you'll see a list of the most recent tags here.

**Automated Builds**  
Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.  
Available with Pro, Team and Business subscriptions. [Read more about automated builds](#).  
[Upgrade](#)

## Github repo after the files pushed from Ec2 to Github

The screenshot shows the GitHub repository page for 'raghavan8 / Guvi-project'. The repository is public and has 2 branches (dev and main) and 0 tags. The dev branch is 2 commits ahead of main. The repository contains several files and folders: build, node\_modules, public, src, Dockerfile, Dockerfile-multistage, README.md, and package-lock.json. The commit history shows that the dev branch has 3 commits, with the most recent commit being 'root new file added' 50 minutes ago. The repository also has a 'Compare & pull request' button and a 'Contribute' button.

After create the AWS EC2 Machine we have open inbound rules to ssh login from only MY Ip And Application access from any IP

The screenshot shows the AWS IAM console 'Inbound rules' page. It displays two inbound rules for a security group. The first rule is for SSH access, allowing traffic from 'My IP' (123.201.241.211/32) on port 22. The second rule is for HTTP access, allowing traffic from 'Anywh...' (0.0.0.0/0) on port 80. Both rules are configured with TCP protocol and have a 'Delete' button next to them.

GIT Repo = [raghavan8/Guvi-project](https://github.com/raghavan8/Guvi-project)

Docker Public = raghavanp08/dev

Docker Private = raghavanp08/prod



## Jenkins Server login.

⚠ Not secure | 3.12.103.205:8080/login?from=%2F

Home - Workday Reports - Microsoft... Compliance Manag... Microsoft Office Ho... My Account https://prudentplus... Java Downloads



**Welcome to Jenkins!**

☐ Keep me signed in

Sign in

## Jenkins server pipeline project by clone git code Dev branch

⚠ Not secure | 3.12.103.205:8080/job/Docker/configure

Home - Workday Reports - Microsoft... Compliance Manag... Microsoft Office Ho... My Account https://prudentplus... Java Downloads | O... SS

ker > Configuration

Git ?

Repositories ?

Repository URL ?

https://github.com/raghavan8/Guvi-project.git

Credentials ?

- none -

Add ▾

Advanced ▾

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

\*/dev

Build the docker image via pipeline

#### Command

See [the list of available environment variables](#)

```
docker build -t newapp:V1.2 .
```

```
sudo docker images
```

Docker images

```
root@ip-172-31-20-197:~# cd Guvi-project/
root@ip-172-31-20-197:~/Guvi-project# docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
newapp        V1.2      3352450df187   6 minutes ago  143MB
newapp        V1.1      e0a1e8a351f2   9 minutes ago  143MB
root@ip-172-31-20-197:~/Guvi-project#
```

## Jenkins output


```
+ ls
Dockerfile
Dockerfile-multistage
README.md
build
node_modules
package-lock.json
package.json
public
src
[DOCKER] $ /bin/sh -xe /tmp/jenkins13458413573952636340.sh
+ docker images
REPOSITORY    TAG       IMAGE ID      CREATED        SIZE
newapp        V1.2      3352450df187  37 minutes ago 143MB
newapp        V1.1      e0a1e8a351f2  40 minutes ago 143MB
[DOCKER] $ /bin/sh -xe /tmp/jenkins14646867547929613126.sh
+ docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS          NAMES
Finished: SUCCESS
```

## Docker images build and push to dev repo via jenkins


```
raghavanp08/dev    latest    3352450df187    8 hours ago    143MB
newapp             V1.1     e0a1e8a351f2    8 hours ago    143MB
+ pwd
/var/lib/jenkins/workspace/Docker
+ cat Dockerfile
FROM nginx:1.20
COPY build /usr/share/nginx/html
[Docker] $ /bin/sh -xe /tmp/jenkins14887826102851197023.sh
+ docker push raghavanp08/dev:latest
The push refers to repository [docker.io/raghavanp08/dev]
ae81ae2800da: Preparing
07ef16952879: Preparing
881700cb7ab2: Preparing
4f49c6d6dd07: Preparing
a64d597d6b14: Preparing
c2a3d4a53f9a: Preparing
fd95118eade9: Preparing
c2a3d4a53f9a: Waiting
fd95118eade9: Waiting
881700cb7ab2: Layer already exists
07ef16952879: Layer already exists
a64d597d6b14: Layer already exists
4f49c6d6dd07: Layer already exists
fd95118eade9: Layer already exists
c2a3d4a53f9a: Layer already exists
ae81ae2800da: Layer already exists
latest: digest: sha256:628356e30ec42c5dc059703bece1a6f3414c12c4bfc7a7b91abd7674db3f38aa size: 1780
Finished: SUCCESS
```


```
+ pwd
/var/lib/jenkins/workspace/Docker
[Docker] $ /bin/sh -xe /tmp/jenkins2594257349517654230.sh
+ docker ps
CONTAINER ID    IMAGE                COMMAND              CREATED          STATUS
8183e5573711    raghavanp08/dev     "/docker-entrpoint...." 3 minutes ago    Up 3 minutes
```

Docker public repo

 **raghavanp08 / dev**



**Description**

for project 

 Last pushed: 2 minutes ago

**Tags**

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
 latest		Image	---	2 minutes ago

[See all](#)

[Go to Advanced Image Management](#)