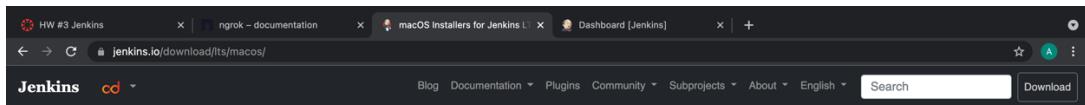


# JENKINS

## Install and Run JENKINS.



### macOS Installers for Jenkins LTS

#### Homebrew Installer

[Tweet](#)

homebrew v2.30.3.1

Jenkins can be installed using the [Homebrew](#) package manager. Homebrew formula: `jenkins-its` This is a package supported by a third party which may be not as frequently updated as packages supported by the Jenkins project directly.

Sample commands:

- Install the latest LTS version: `brew install jenkins-lts`
- Install a specific LTS version: `brew install jenkins-lts@YOUR_VERSION`
- Start the Jenkins service: `brew services start jenkins-lts`
- Restart the Jenkins service: `brew services restart jenkins-lts`
- Update the Jenkins version: `brew upgrade jenkins-lts`

After starting the Jenkins service, browse to <http://localhost:8080> and follow the instructions to complete the installation. Also see the external materials for installation guidelines. For example, [this blogpost](#) describes the installation process.

#### Native Installer (deprecated)

Jenkins project used to provide a native installer for macOS. This installer is now deprecated, and it will not be shipped for future versions of Jenkins. It is possible to retrieve installer versions for older releases from the archive.

- [Installers for previous Jenkins versions.](#)
- [Learn how the macOS installer sets up your Jenkins, and how you can tweak it.](#)

## AFTER RUN JENKINS --> localhost:8080

Welcome to Jenkins!

admin

.....

Keep me signed in

For the first time run JENKINS, it will ask for create account --> you will get your id and password.

## AFTER LOGIN

The screenshot shows the Jenkins dashboard at [localhost:8080](http://localhost:8080). The left sidebar includes links for New Item, People, Build History, Manage Jenkins, My Views, Lockable Resources, and New View. The main area displays a table for a project named 'TermProject' with columns for Status, Name, Last Success, Last Failure, and Last Duration. A legend indicates icons for Success (green checkmark), Warning (yellow cloud), and Failure (red circle). Buttons for 'add description' and 'Atom feed' are also present.

## GO TO NEW ITEM TO CREATE YOUR OWN

The screenshot shows the 'New Item' creation page at [localhost:8080/view/all/newJob](http://localhost:8080/view/all/newJob). It features a large input field for 'Enter an item name' with a note that it's a required field. Below the input field, there are five project types listed with their descriptions and icons:

- Freestyle project**: 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.
- 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.
- Multibranch Pipeline**: Creates a set of Pipeline projects according to detected branches in one SCM repository.

At the bottom, there is a button labeled 'OK'.

Enter the name of project --> Freestyle Project --> Configure

The screenshot shows the Jenkins configuration interface for a 'TermProject'. The 'Source Code Management' tab is selected. Under 'Source Code Management', 'Git' is chosen as the provider. The 'Repository URL' field contains the value [https://github.com/vuhoangtuananh3004/CMPE131\\_TermProject.git](https://github.com/vuhoangtuananh3004/CMPE131_TermProject.git). The 'Save' and 'Apply' buttons are visible at the bottom.

SOURCE CODE MANAGEMENT --> ADD GIT REPOSITORIES URL:

For now use this: (we can change it later on.)

[https://github.com/vuhoangtuananh3004/CMPE131\\_TermProject.git](https://github.com/vuhoangtuananh3004/CMPE131_TermProject.git)

Branch Specifier change from master --> main, because git already change it.

The screenshot shows the Jenkins configuration interface for a 'TermProject'. The 'Source Code Management' tab is selected. Under 'Branches to build', the 'Branch Specifier (blank for 'any')' field contains the value '/main'. The 'Save' and 'Apply' buttons are visible at the bottom.

--> Apply

Optional: (Add webhook).

Build Triggers --> Git hook trigger for polling.

Webhook will send a notification to your Jenkins whenever github respository updated.

GO TO: [https://github.com/vuhoangtuananh3004/CMPE131\\_TermProject](https://github.com/vuhoangtuananh3004/CMPE131_TermProject)

-->Setting --> Webhook.

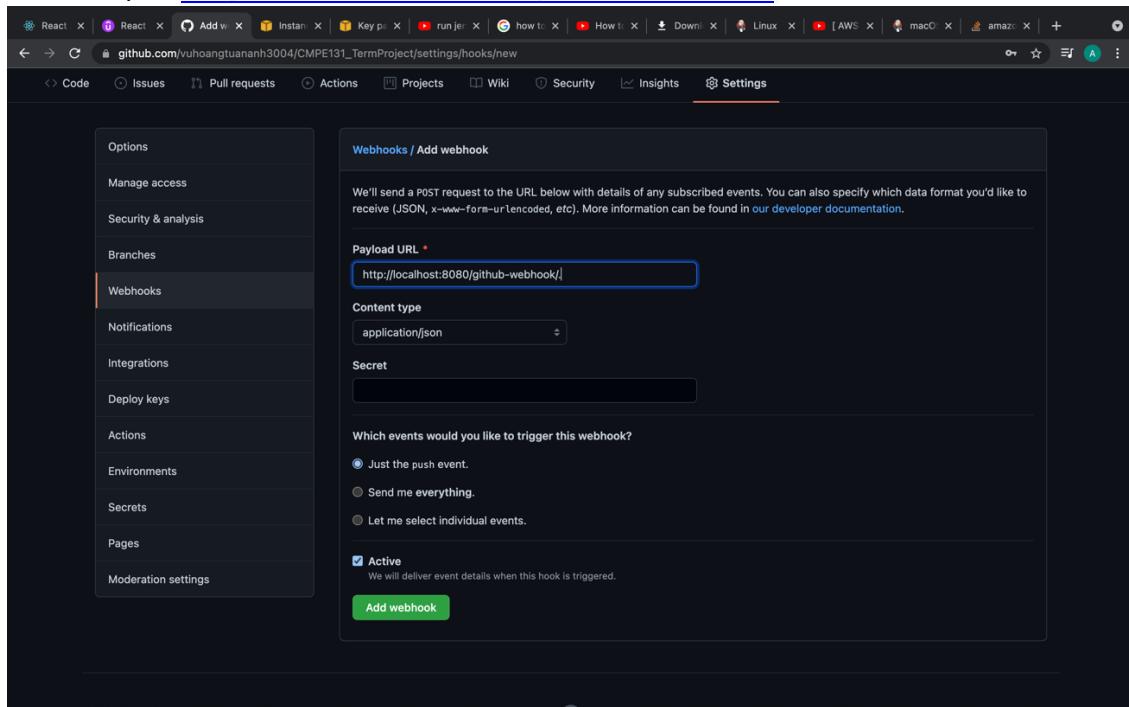
The screenshot shows the GitHub repository settings for 'vuhoangtuananh3004 / CMPE131\_TermProject'. The 'Webhooks' tab is selected in the sidebar. The main area displays a list of existing webhooks. One webhook is listed with the URL 'http://f0e6-24-23-178-54.ngrok... (all events)'. There are 'Edit' and 'Delete' buttons next to it. A 'Add webhook' button is located at the top right of the list area.

Choose Add webhook

The screenshot shows the 'Add webhook' form within the GitHub repository settings. The 'Webhooks' tab is selected in the sidebar. The main area has a title 'Webhooks / Add webhook'. It includes instructions: 'We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in our developer documentation.' Below this are fields for 'Payload URL' (set to 'https://example.com/postreceive'), 'Content type' (set to 'application/json'), and 'Secret' (an empty input field). Under 'Which events would you like to trigger this webhook?', there are three radio buttons: 'Just the push event.' (selected), 'Send me everything.', and 'Let me select individual events.'. At the bottom, there is a checked checkbox for 'Active' with the note 'We will deliver event details when this hook is triggered.' and a 'Add webhook' button.

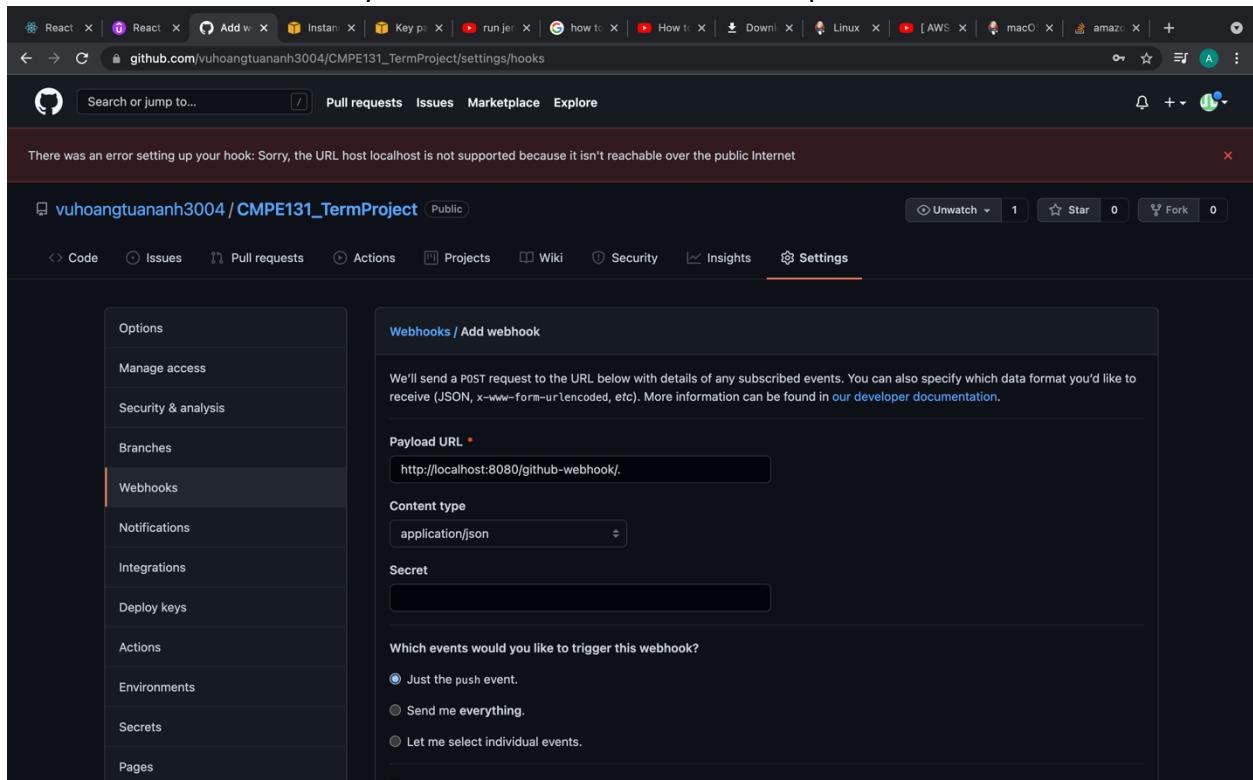
Add the link of your Jenkins + /webhook/

For example: <http://localhost:8080/github-webhook/>.



The screenshot shows the GitHub 'Webhooks / Add webhook' configuration page. On the left, a sidebar lists various settings: Options, Manage access, Security & analysis, Branches, Webhooks (which is selected and highlighted in orange), Notifications, Integrations, Deploy keys, Actions, Environments, Secrets, Pages, and Moderation settings. The main right-hand panel has a title 'Webhooks / Add webhook'. It contains instructions: 'We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in our developer documentation.' Below this, the 'Payload URL' field is populated with 'http://localhost:8080/github-webhook/'. The 'Content type' dropdown is set to 'application/json'. There is a 'Secret' input field, though it appears empty. Under 'Which events would you like to trigger this webhook?', the radio button 'Just the push event.' is selected. A checkbox labeled 'Active' is checked, with the note 'We will deliver event details when this hook is triggered.' At the bottom is a green 'Add webhook' button.

Choose Add WebHook, however it will not work because your ip adress is not public, so github cannot send notification to your localhost --> need to make it public.



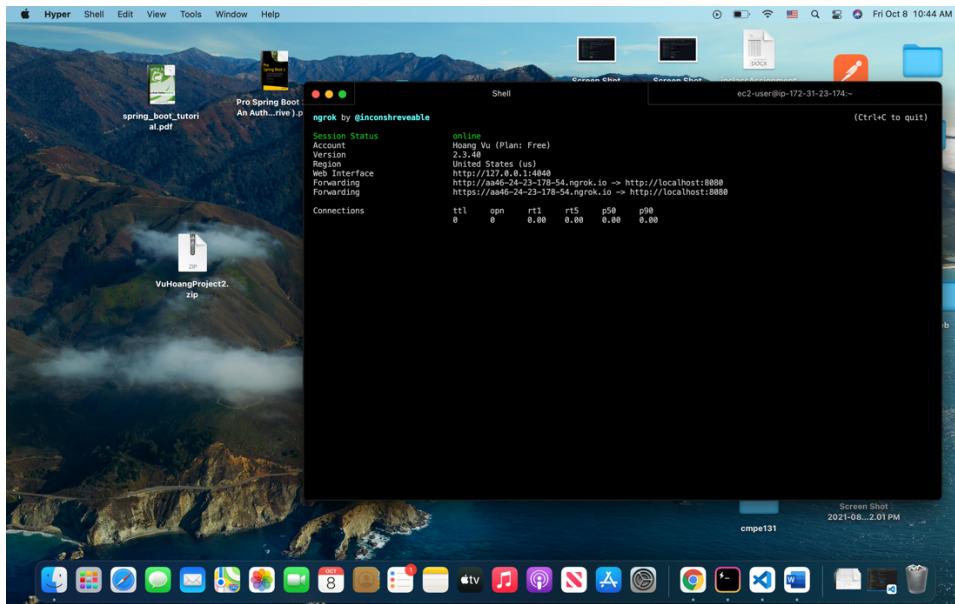
The screenshot shows the same GitHub 'Webhooks / Add webhook' configuration page as the previous one, but with an error message displayed at the top: 'There was an error setting up your hook: Sorry, the URL host localhost is not supported because it isn't reachable over the public Internet'. The sidebar and form fields are identical to the successful configuration screenshot above, except for the error message.

## TO MAKE YOUR IP PUBLIC:

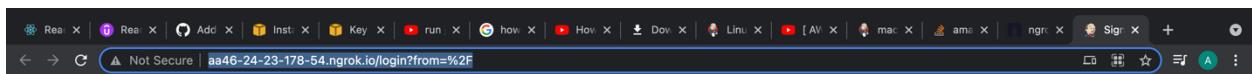
### Use NGROK (temporary)

Bad thing: whenever you turn off your computer --> when you run again the ip address is changing except you pay for feature) follow step by step from: <https://ngrok.com/> to make you ip public.

### RESULT:



You can see the link is different now.



Welcome to Jenkins!

**Sign in**

Keep me signed in

## Use AWS EC2 Instance:

1/ Go to and Sign in : <https://aws.amazon.com/console/>

2/ Choose Services --> EC2 -->Instances -->Launch instances

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with options like EC2 Dashboard, Events, Tags, Limits, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances (New), Dedicated Hosts, Capacity Reservations, Images (AMIs), Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), and Network & Security. The main area displays a table of instances with columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public. Two instances are listed: 'AWS' (i-02625e012b62557e) and 'Jenkins' (i-0733afce01cf95883), both in the 'Running' state. A modal window titled 'Select an instance above' is open over the Jenkins instance. At the bottom, there are links for Feedback, English (US), and various AWS services.

## Choose Amazon Linux (Free)

The screenshot shows the 'Choose AMI' step in the AWS Launch Instance Wizard. The top navigation bar includes links for 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage, 5. Add Tags, 6. Configure Security Group, and 7. Review. Below the navigation, there's a search bar with placeholder text 'Search for an AMI by entering a search term e.g. "Windows"' and a 'Cancel and Exit' link. The main content area is titled 'Quick Start' and lists several AMIs:

- Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-074cce78125f09d61 (64-bit x86) / ami-01019fb8b29b5dc5d (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  
Select button (radio buttons for 64-bit (x86) and 64-bit (Arm))
- macOS Big Sur 11.6 - ami-0c84d9da210c1110b**  
The macOS Big Sur AMI is an EBS-backed, AWS-supported image. This AMI includes the AWS Command Line Interface, Command Line Tools for Xcode, Amazon SSM Agent, and Homebrew. The AWS Homebrew Tap includes the latest versions of multiple AWS packages included in the AMI.  
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes  
Select button (radio buttons for 64-bit (Mac))
- macOS Catalina 10.15.7 - ami-080f3de7a1a857505**  
The macOS Catalina AMI is an EBS-backed, AWS-supported image. This AMI includes the AWS Command Line Interface, Command Line Tools for Xcode, Amazon SSM Agent, and Homebrew. The AWS Homebrew Tap includes the latest versions of multiple AWS packages included in the AMI.  
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes  
Select button (radio buttons for 64-bit (Mac))
- macOS Mojave 10.14.6 - ami-03aea00119db6a8f**  
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes  
Select button

## Skip until step 5:

### Choose Add Tag

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. A copy of a tag can be applied to volumes, instances or both. Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key	(128 characters maximum)	Value	(256 characters maximum)	Instances	Volumes	Network Interfaces
Name		Whatever you want		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Add another tag (Up to 50 tags maximum)

## Move to step 6. Select Custom TCP Rule, and Port Range: 8080 because Jenkins run on 8080

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group:  Create a new security group  Select an existing security group

Security group name: launch-wizard-2

Description: launch-wizard-2 created 2021-10-08T10:53:30.094-07:00

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
Custom TCP Rule	TCP	8080	Custom 0.0.0.0/:/0	e.g. SSH for Admin Desktop

Custom TCP Rule

- Custom UDP Rule
- Custom ICMP Rule - IPv4
- Custom ICMP Rule - IPv6
- Custom Protocol
- All TCP
- All UDP
- All ICMP - IPv4
- All ICMP - IPv6
- All traffic
- SSH
- SMTP
- DNS (UDP)
- DNS (TCP)
- HTTP
- POP3
- IMAP
- LDAP

0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Move to step 7: Launch, and it will ask for keypair, if you have one select it, if not create new one. Then download the keypair we will use later --> Launch Instances.

**Step 7: Review Instance Launch**

Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 215, 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 no longer supported.

Root Device Type: ebs Virtualization type: hvm

**Instance Type**

Instance Type	ECUs	vCPUs
t2.micro	-	1

**Security Groups**

Security group name	Description
launch-wizard-2	launch-wizard-2 created

Type: SSH Protocol: TCP

Custom TCP Rule: TCP

Custom TCP Rule: TCP

**Instance Details**

**Storage**

**Tags**

**Select an existing key pair or create a new key pair**

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance. Amazon EC2 supports ED25519 and RSA key pair types.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Create a new key pair

Key pair type: RSA

Key pair name: Test

Download Key Pair

You have to download the **private key file** (\*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

Cancel Launch Instances

Finally, You get this, your server is running.

**New EC2 Experience**

Tell us what you think

**EC2 Dashboard**

**EC2 Global View**

**Events**

**Tags**

**Limits**

**Instances**

**Instances** New

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances New

Dedicated Hosts

Capacity Reservations

**Images**

AMIs

**Elastic Block Store**

Volumes

Snapshots

Lifecycle Manager

Feedback English (US)

Instances (3) Info

Filter instances

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zon
AWS	i-02625e012b62557e2	Running	t2.micro	2/2 checks passed	No alarms	us-east-2b
Jenkins	i-0733afce01cf95883	Running	t2.micro	2/2 checks passed	No alarms	us-east-2c
Whatever you want	i-0597fb6f9a6b8e325	Running	t2.micro	Initializing	No alarms	us-east-2a

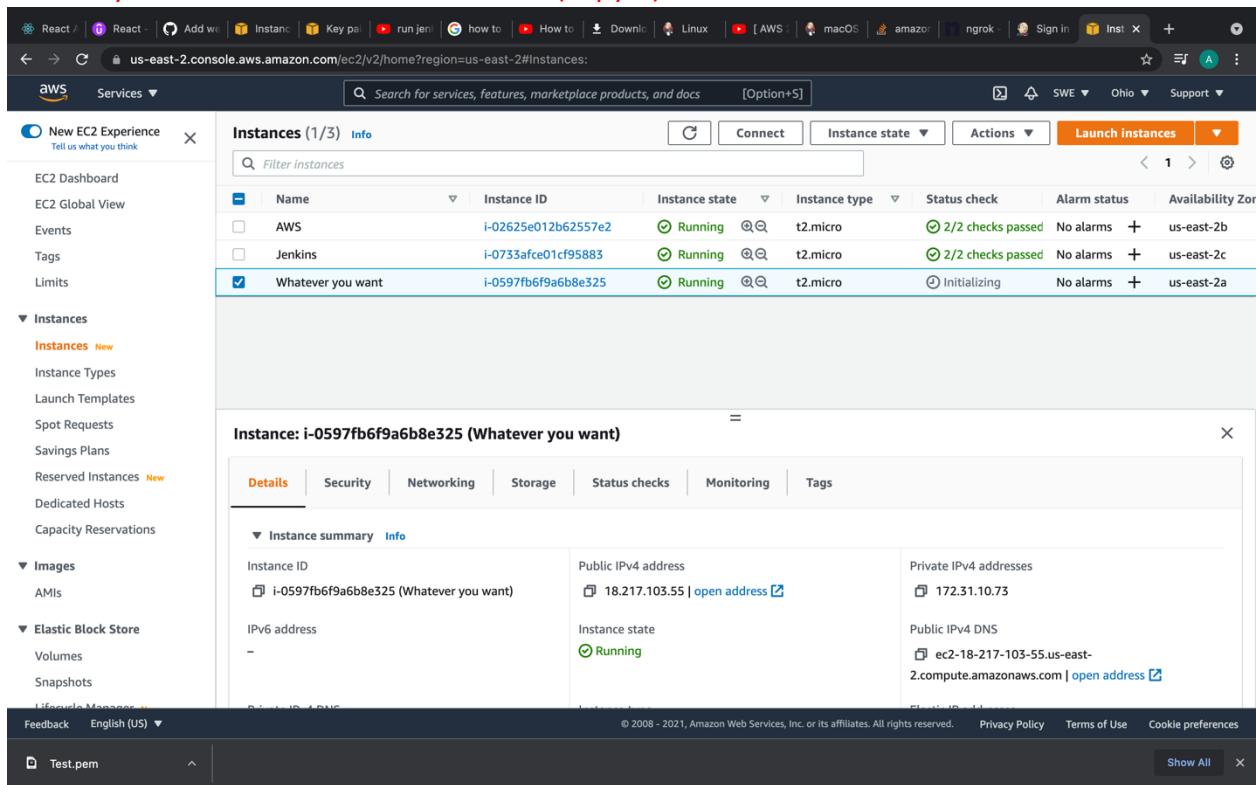
Select an instance above

Feedback English (US)

Test.pem

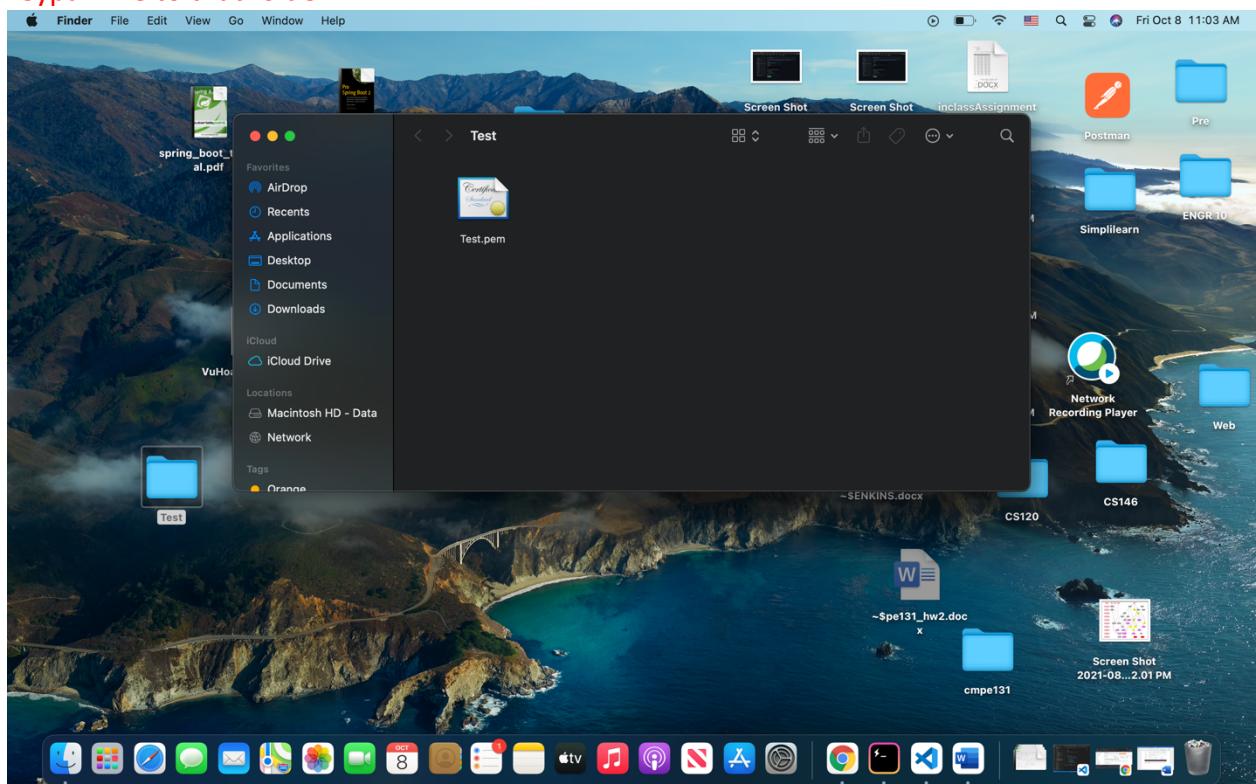
Show All

Click on your server --> Public IPv4 address (copy it).

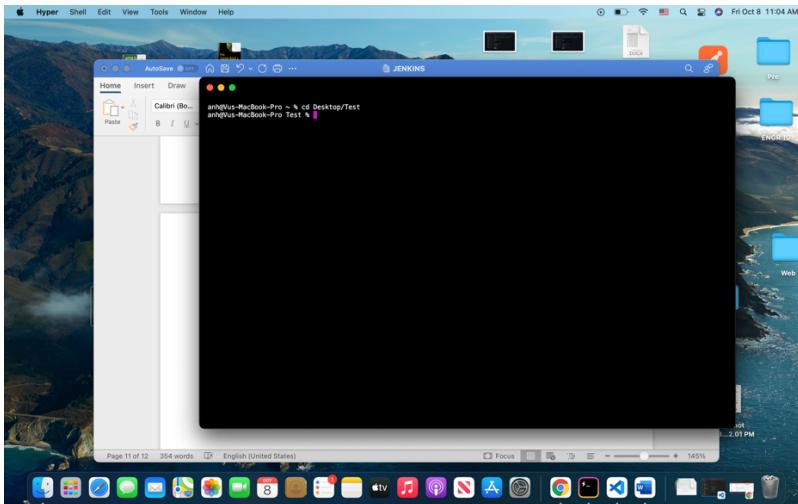


The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with various navigation options like EC2 Dashboard, EC2 Global View, Events, Tags, Limits, Instances, Images, and Elastic Block Store. The Instances section is currently active. In the main area, a table lists three instances: AWS, Jenkins, and 'Whatever you want'. The 'Whatever you want' instance is selected. A detailed view modal is open for this instance, showing its ID (i-0597fb6f9a6b8e325), Name ('Whatever you want'), Instance ID (i-0597fb6f9a6b8e325), Instance state (Running), Instance type (t2.micro), Status check (2/2 checks passed), Alarm status (No alarms), and Availability Zone (us-east-2a). The modal also displays the Public IPv4 address (18.217.103.55) and Private IPv4 address (172.31.10.73).

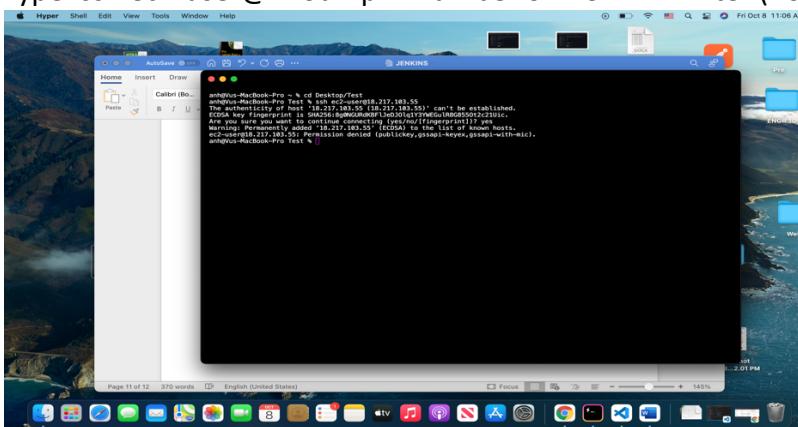
To run Server on your computer, create a folder on your desktop (or anywhere) and copy the keypair file to that folder



Now open your terminal, and go into that folder

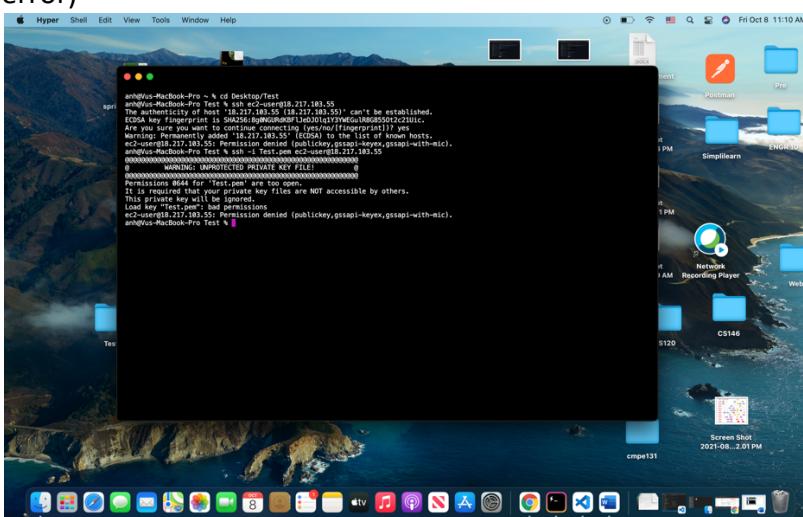


Type: ssh ec2-user@<Your ip4v number on EC2> --> Enter (however it will show an error)



It needs permission:

Type: ssh -i <keypair file> ec2-user@<Your ip4v number on EC2> --> Enter (it will show another error)

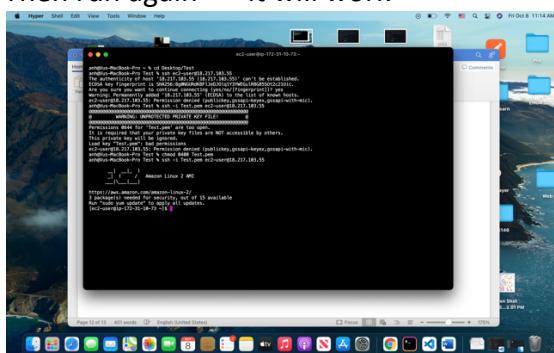


Type chmod 0400 <keypair file>

what is chmod 0400 (<https://www.cyberciti.biz/faq/unix-linux-bsd-chmod-numeric-permissions-notation-command/>)



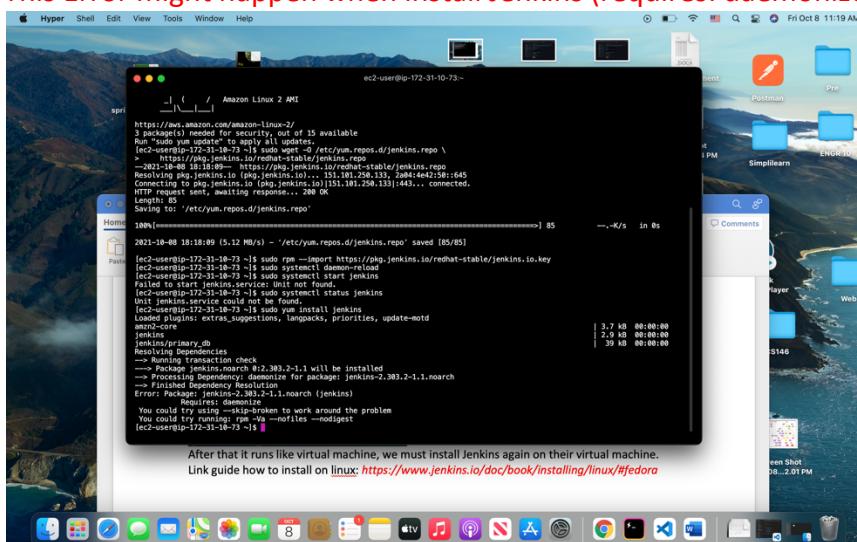
Then run again --> it will work



After that it runs like virtual machine, we must install Jenkins again on their virtual machine.

Link guide how to install on linux: <https://www.jenkins.io/doc/book/installing/linux/#fedora>

This Error might happen when install Jenkins (requires: daemonize)



So you run :

**sudo amazon-linux-extras install epel -y**

**sudo yum update -y**

Then install Jenkins and Java

**sudo yum install jenkins java-1.8.0-openjdk-devel**

After all, to run Jenkins

```
sudo systemctl start jenkins  
sudo systemctl status jenkins
```

The screenshot shows a web browser window with multiple tabs open. The active tab is titled "Jenkins" and contains instructions for starting the Jenkins service. It includes a command-line snippet:

```
Sudo rpm --import https://pkg.jenkins.io/redhat/jenkins.io.key  
sudo yum upgrade  
sudo yum install epel-release java-11-openjdk-devel  
sudo yum install jenkins  
sudo systemctl daemon-reload
```

Below this, there's a section titled "Start Jenkins" with a link icon. It says: "You can start the Jenkins service with the command:" followed by:

```
sudo systemctl start jenkins
```

Further down, it says: "You can check the status of the Jenkins service using the command:" followed by:

```
sudo systemctl status jenkins
```

If everything has been set up correctly, you should see an output like this:

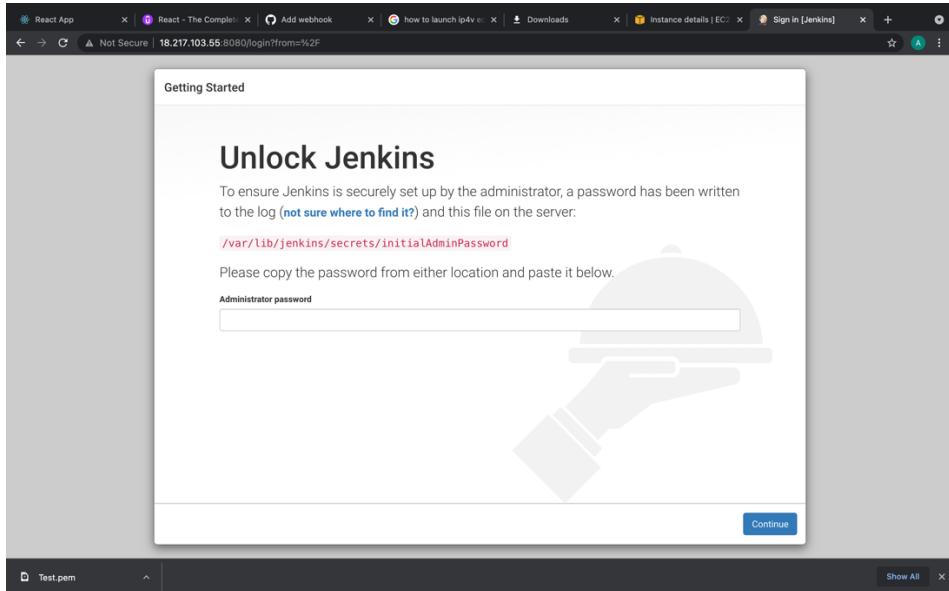
```
Loaded: loaded (/etc/rc.d/init.d/jenkins; generated)  
Active: active (running) since Tue 2018-11-13 16:19:01 +03; 4min 57s ago  
...
```

The screenshot shows a macOS desktop environment with a terminal window open. The terminal window title is "Test.pem" and the content shows the execution of Jenkins startup commands. The terminal output is as follows:

```
https://www.jenkins.io/doc/book/installing/linux/#start-jenkins-2  
Test.pem  
Hyper Shell Edit View Tools Window Help Fri Oct 8 11:24 AM  
ec2-user@ip-172-31-10-73:~  
$ sudo systemctl start jenkins  
$ sudo systemctl status jenkins  
● jenkins.service - LSB: Jenkins Automation Server  
  Loaded: loaded (/etc/rc.d/init.d/jenkins; bad; vendor preset: disabled)  
  Active: active (running) since Fri 2021-10-08 18:24:45 UTC; 10s ago  
    Docs: man:systemd-sysv-generator(8)  
  Process: 32435 ExecStart=/etc/rc.d/init.d/jenkins start (code=exited, status=0/SUCCESS)  
  CGroup: /system.slice/jenkins.service  
          └─32439 /etc/alternatives/java -DJAVA.awt.headless=true -DJENKINS_HOME=/var/lib/jenkins -jar /usr/lib/jenkins/jenkins...  
Oct 08 18:24:45 ip-172-31-10-73.us-east-2.compute.internal systemd[1]: Starting Jenkins Automation Server...  
Oct 08 18:24:45 ip-172-31-10-73.us-east-2.compute.internal jenkins[32435]: Starting Jenkins [ OK ]  
Oct 08 18:24:45 ip-172-31-10-73.us-east-2.compute.internal systemd[1]: Started LSB: Jenkins Automation Server.  
[ec2-user@ip-172-31-10-73 ~]$
```

The desktop background features a scenic coastal landscape. A dock on the right side contains icons for Postman, Pre, Simplilearn, ENGR 101, Network Recording Player, CS146, and S120. The bottom of the screen shows the macOS Dock with various application icons.

Now you go to browser:



To get the administrator password (on Linux):

`sudo cat /var/lib/jenkins/secrets/initialAdminPassword`

