Install Jenkins on Amazon EC2 Linux instances

Step 1: Launch an instance – Refer

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html

---

Master-Slave Steps

Jenkins Master is the primary Jenkins server and is responsible for the following tasks:

- It distributes the builds among the numerous slaves for execution.
- It organises the build projects.
- It keeps an eye on the slaves at all times.
- Master can also run build jobs directly if necessary.
- It keeps track of the build outcomes and shows them.

Jenkins Slave runs on a remote machine. A slave is responsible for the following tasks:

- Slaves can be operated on a number of different operating systems.
- It responds to the Jenkins Master's demands.
- Apart from the fact that Jenkins executes the build task on the next available save,
- we can always arrange the project to run on a certain sort of slave computer.
- It also completes construction operations that the Master has dispatched.

Steps -

1. Install Java on master node

2. Install Jenkins on master node

3. Install java on slave node

4. Create a user and ssh keys on slave node

5. Copy keys on master node

6. Join slave node to master

7. Test the setup

You can launch a Linux instance using the AWS Management Console as described in the following procedure. This tutorial is intended to help you quickly launch your first instance, so it doesn't cover all possible options. For information about advanced options, see Launch an instance using the new launch instance wizard. For information about other ways to launch your instance, see Launch your instance.

To launch an instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the EC2 console dashboard, in the **Launch instance** box, choose **Launch instance**, and then choose **Launch instance** from the options that appear.
3. Under **Name and tags**, for **Name**, enter a descriptive name for your instance.
4. Under **Application and OS Images (Amazon Machine Image)**, do the following:
    a. Choose **Quick Start**, and then choose Amazon Linux. This is the operating system (OS) for your instance.
    b. From **Amazon Machine Image (AMI)**, select an HVM version of Amazon Linux 2. Notice that these AMIs are marked **Free tier eligible**. An *Amazon Machine Image (AMI)* is a basic configuration that serves as a template for your instance.
5. Under **Instance type**, from the **Instance type** list, you can select the hardware configuration for your instance. Choose the `t2.micro` instance type, which is selected by default. The `t2.micro` instance type is eligible for the free tier. In Regions where `t2.micro` is unavailable, you can use a `t3.micro` instance under the free tier. For more information, see AWS Free Tier.
6. Under **Key pair (login)**, for **Key pair name**, choose the key pair that you created when getting set up.
   Warning

   Do not choose **Proceed without a key pair (Not recommended)**. If you launch your instance without a key pair, then you can't connect to it.
7. Next to **Network settings**, choose **Edit**. For **Security group name**, you'll see that the wizard created and selected a security group for you. You can use this security group, or alternatively you can select the security group that you created when getting set up using the following steps:
    a. Choose **Select existing security group**.

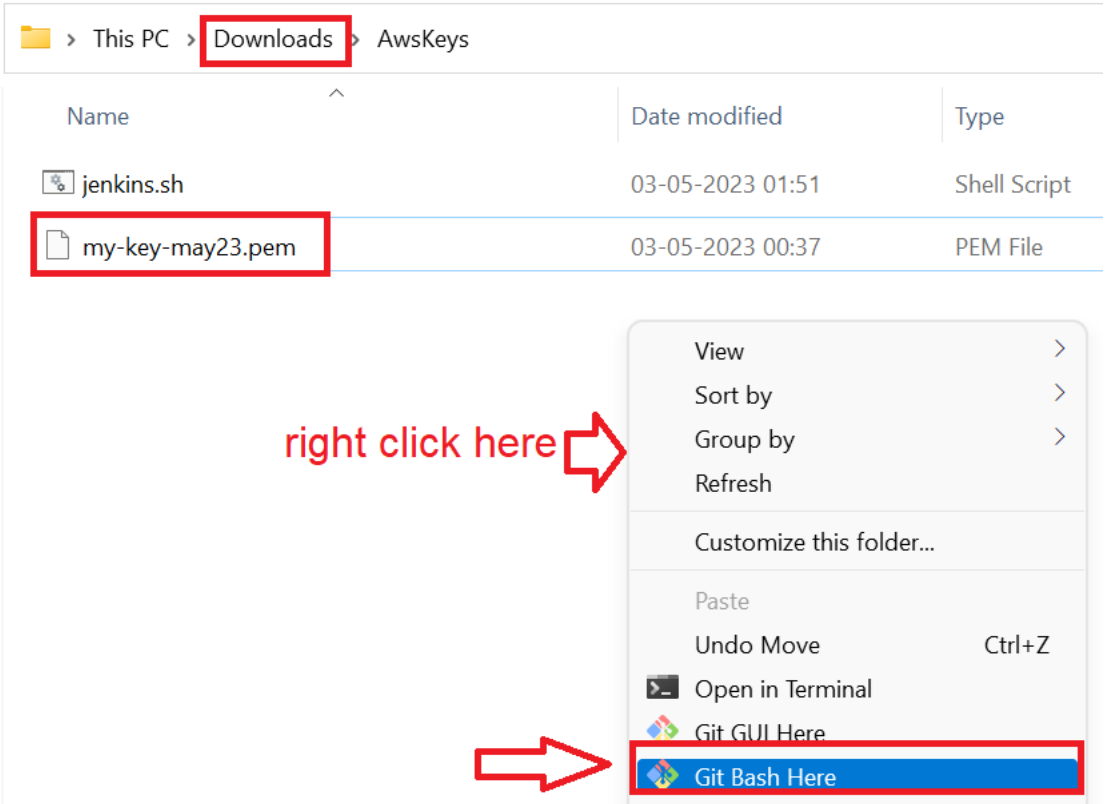
b. From **Common security groups**, choose your security group from the list of existing security groups.

8. Keep the default selections for the other configuration settings for your instance.
9. Review a summary of your instance configuration in the **Summary** panel, and when you're ready, choose **Launch instance**.
10. A confirmation page lets you know that your instance is launching. Choose **View all instances** to close the confirmation page and return to the console.
11. On the **Instances** screen, you can view the status of the launch. It takes a short time for an instance to launch. When you launch an instance, its initial state is `pending`. After the instance starts, its state changes to `running` and it receives a public DNS name. If the **Public IPv4 DNS** column is hidden, choose the settings icon (⚙) in the top-right corner, toggle on **Public IPv4 DNS**, and choose **Confirm**.
12. It can take a few minutes for the instance to be ready for you to connect to it. Check that your instance has passed its status checks; you can view this information in the **Status check** column.

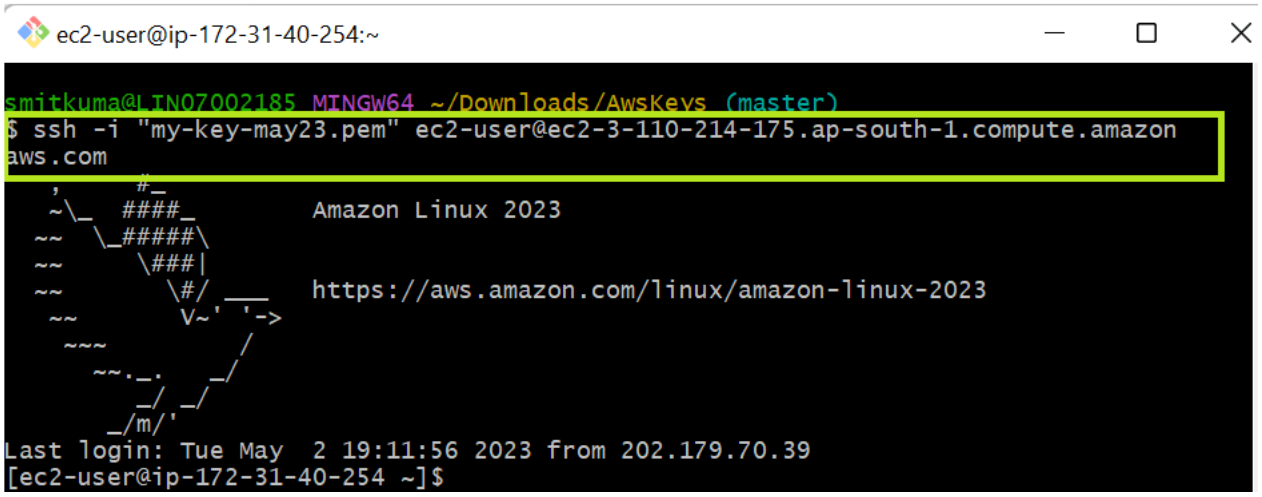
Step 2: Connect to your instance

For Linux instance command – refer this link https://www.jenkins.io/doc/book/installing/linux/#red-hat-centos NOTE : make change to OpenJDK command as highlighted below

1. Got to the in the downloaded .pem file location and right click and Open gitbash here



2. Then type - **ssh -i "KEYFILENAME.pem" ec2-user@YOUR_LINUX_INSTANCE_PUBLIC_DNS**
Example : ssh -i "JM-key.pem" ec2-user@ec2-3-109-181-9.ap-south-1.compute.amazonaws.com



3. **vi jenkins.sh** - it will open a vi editor **for Jenkins.sh file**
4. press **i**
5. copy the below command and right click on the vi editor and paste there **(**For Linux instance command – refer this link https://www.jenkins.io/doc/book/installing/linux/#red-hat-centos) refer this for installing jdk on linux - https://docs.aws.amazon.com/corretto/latest/corretto-11-ug/amazon-linux-install.html

6. now press **esc**
7. now press **shift + :**
8. now enter **wq** and **enter** – you will be now out of editor
9. **sh jenkins.sh**

```
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-
amazon-corretto -y
sudo java -version
sudo yum install jenkins -y
sudo systemctl daemon-reload
sudo systemctl enable jenkins
sudo systemctl start jenkins
sudo systemctl status jenkins
```

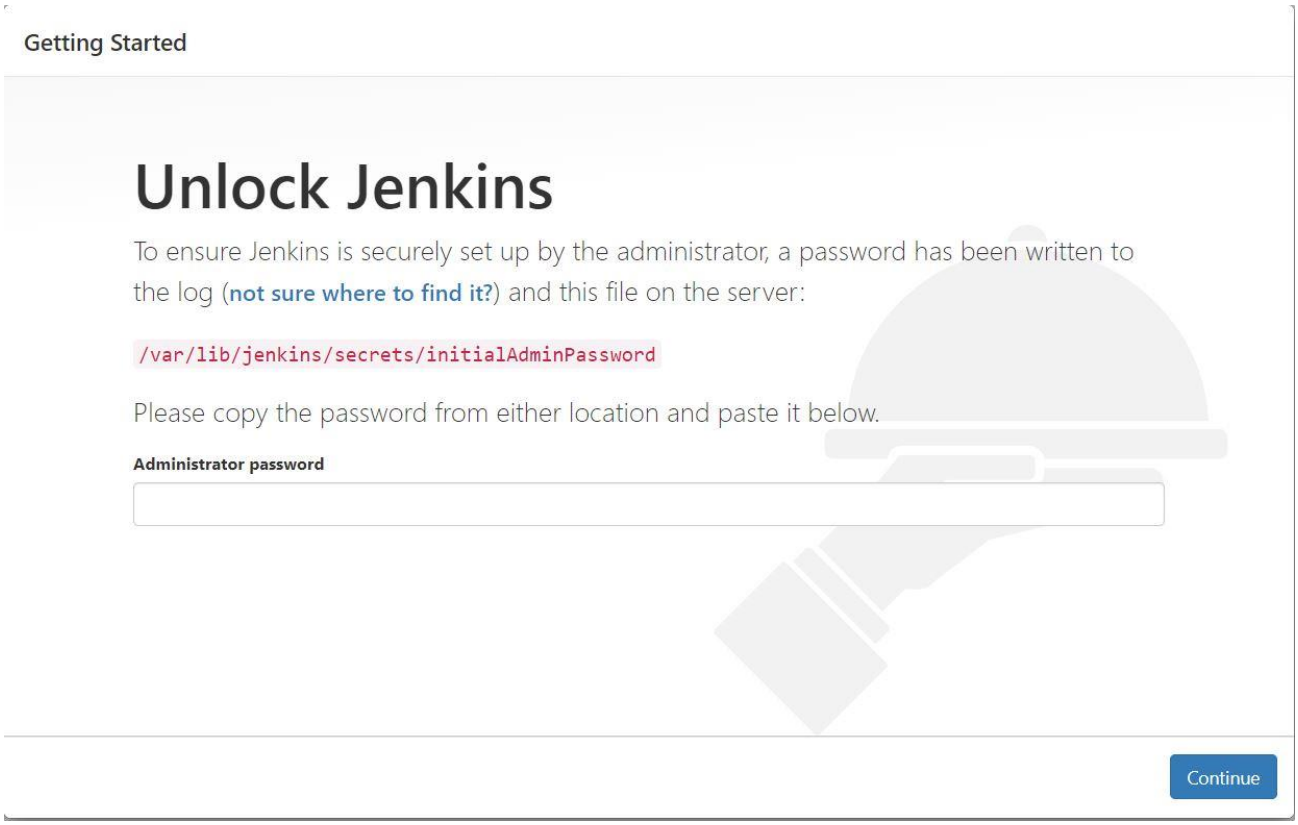
```
Complete!
Last metadata expiration check: 1:38:55 ago on Tue May  2 19:31:26 2023.
Package jenkins-2.403-1.1.noarch is already installed.
Dependencies resolved.
Nothing to do.
Complete!
Synchronizing state of jenkins.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable jenkins
● jenkins.service - Jenkins Continuous Integration Server
     Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; preset: disabled)
     Active: active (running) since Tue 2023-05-02 21:08:35 UTC; 1min 47s ago
   Main PID: 31016 (java)
      Tasks: 36 (limit: 1112)
     Memory: 315.1M
        CPU: 41.617s
     CGroup: /system.slice/jenkins.service
             └─31016 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/wa

May 02 21:08:03 ip-172-31-40-254.ap-south-1.compute.internal jenkins[31016]: 3a51a01bd3ed4e1684ecd306492a6719
May 02 21:08:03 ip-172-31-40-254.ap-south-1.compute.internal jenkins[31016]: This may also be found at: /var/lib/jenkins/secr
```

Configuring Jenkins – reference https://www.jenkins.io/doc/tutorials/tutorial-for-installing-jenkins-on-AWS/#connecting-to-your-linux-instance

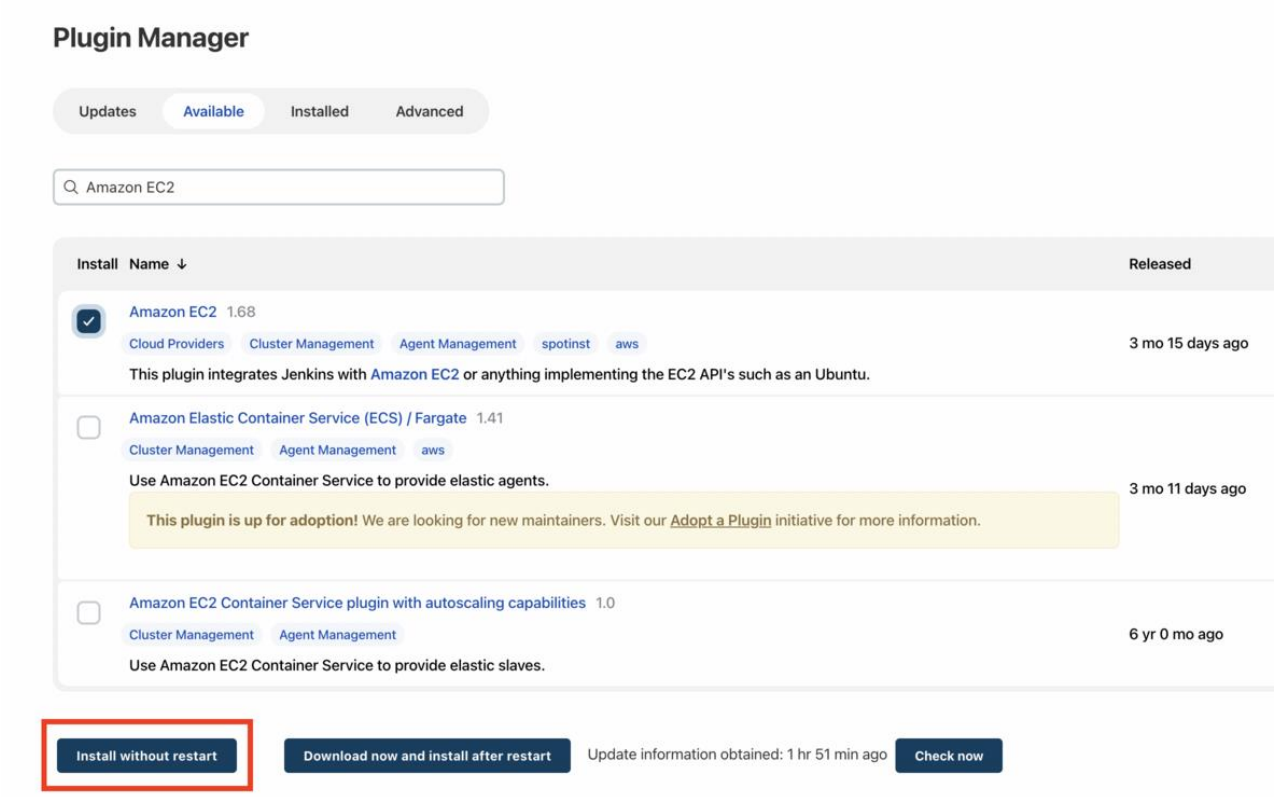
Jenkins is now installed and running on your EC2 instance. To configure Jenkins:

1. Connect to http://<your_server_public_DNS>:8080 from your browser. You will be able to access Jenkins through its management interface:
   Example : http://ec2-3-110-214-175.ap-south-1.compute.amazonaws.com:8080

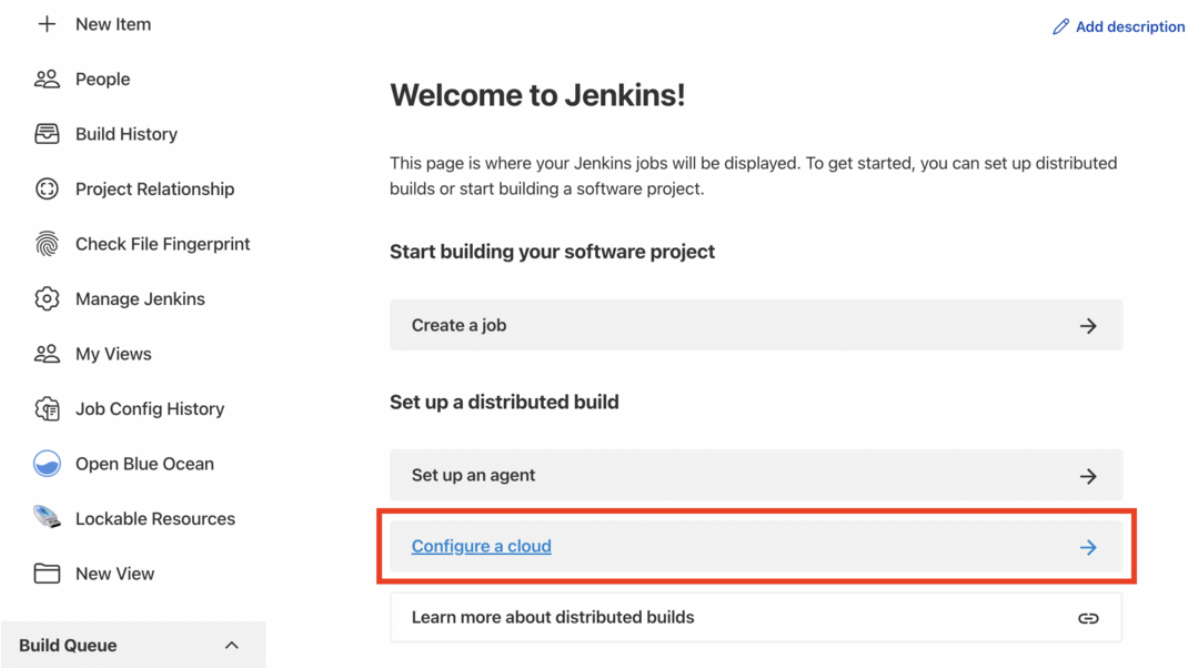


2. As prompted, enter the password found in **/var/lib/jenkins/secrets/initialAdminPassword**.
    a. Use the following command to display this password:
    ```
    [ec2-user ~]$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
    ```
3. The Jenkins installation script directs you to the **Customize Jenkins page**. Click **Install suggested plugins**.
4. Once the installation is complete, the **Create First Admin User** will open. Enter your information, and then select **Save and Continue**.

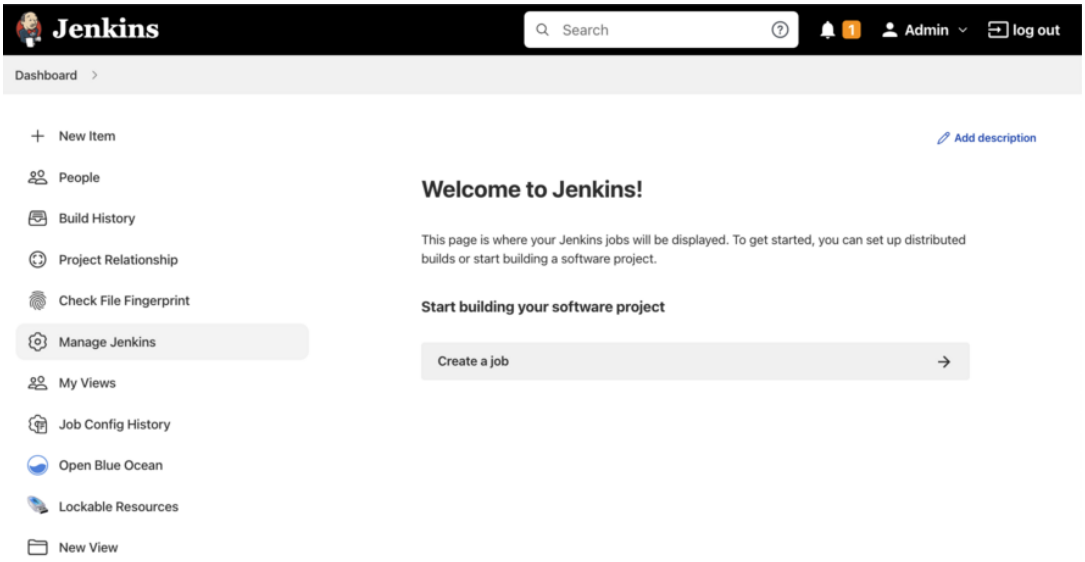
5. On the left-hand side, select **Manage Jenkins**, and then select **Manage Plugins**.
6. Select the **Available** tab, and then enter **Amazon EC2 plugin** at the top right.
7. Select the checkbox next to **Amazon EC2 plugin**, and then select **Install without restart**.



8. Once the installation is done, select **Back to Dashboard**.
9. Select **Configure a cloud** if there are no existing nodes or clouds.

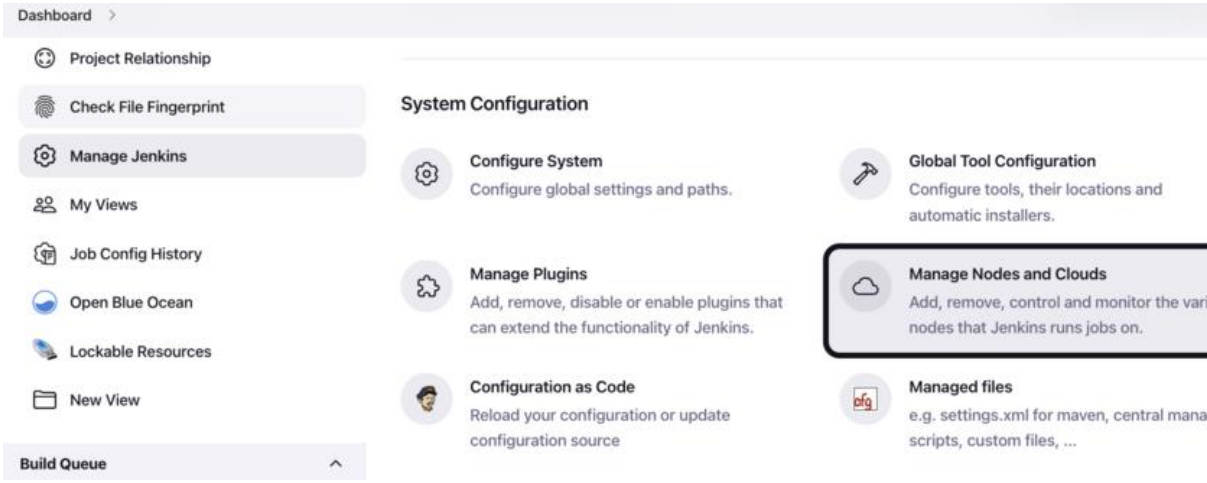


10. If you already have other nodes or clouds set up, select **Manage Jenkins**.

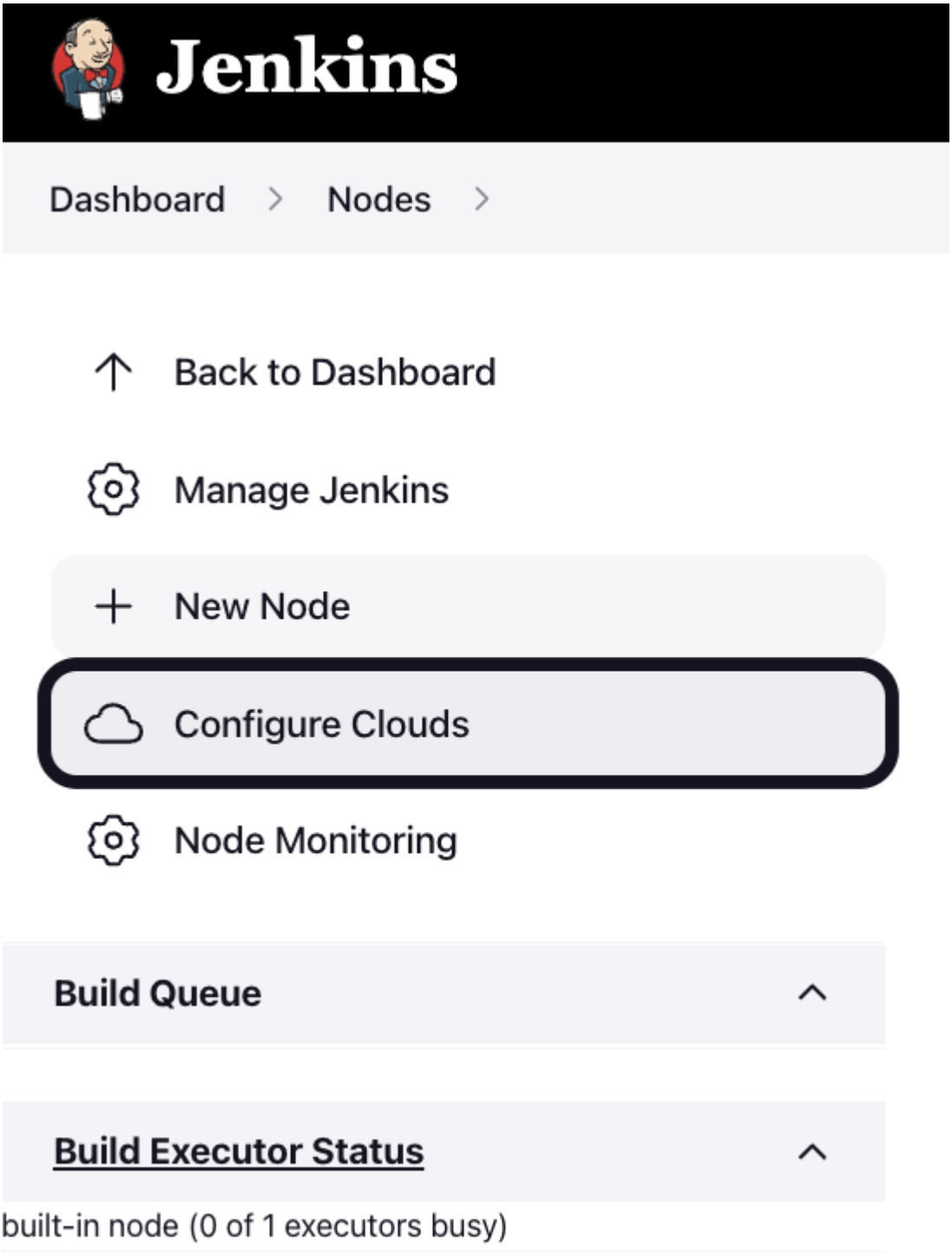


1.

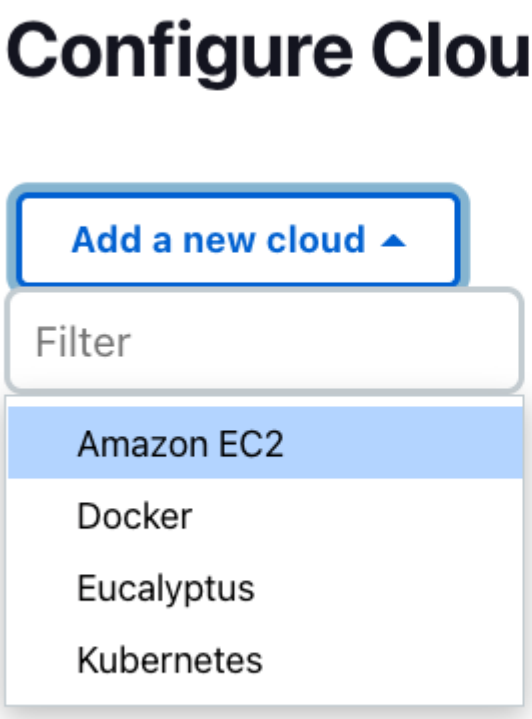
a. After navigating to **Manage Jenkins**, select **Configure Nodes and Clouds** from the left hand side of the page.



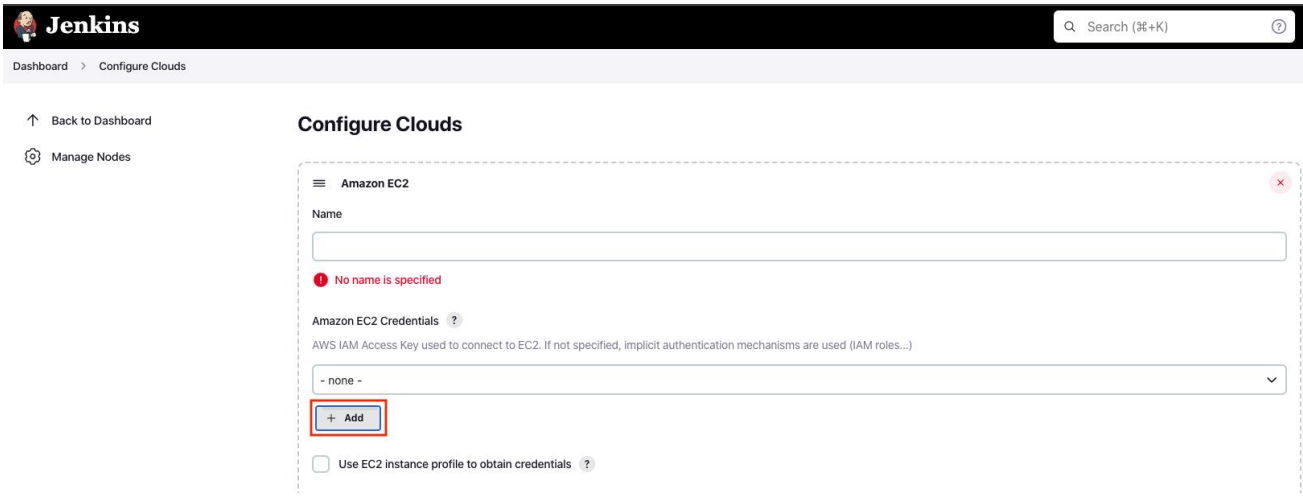
b. From here, select **Clouds**.



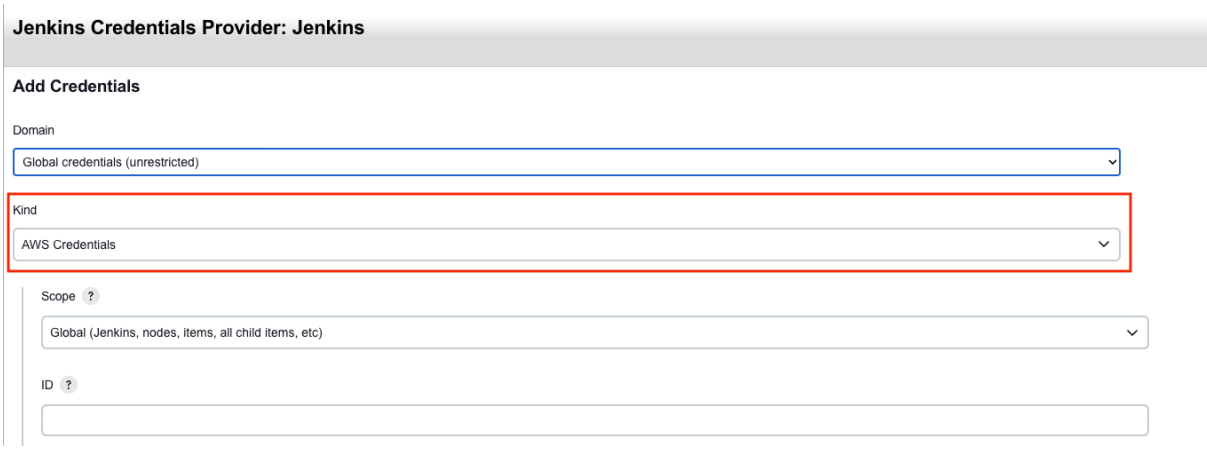
2. Select **Add a new cloud**, and select **Amazon EC2**. A collection of new fields appears.



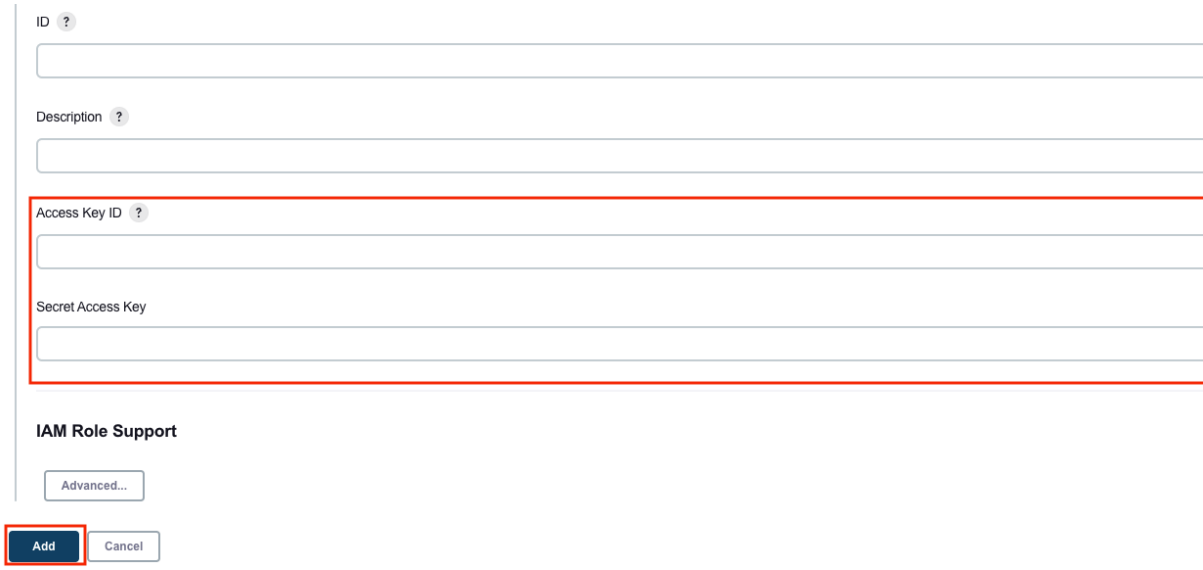
3. Click **Add** under Amazon EC2 Credentials



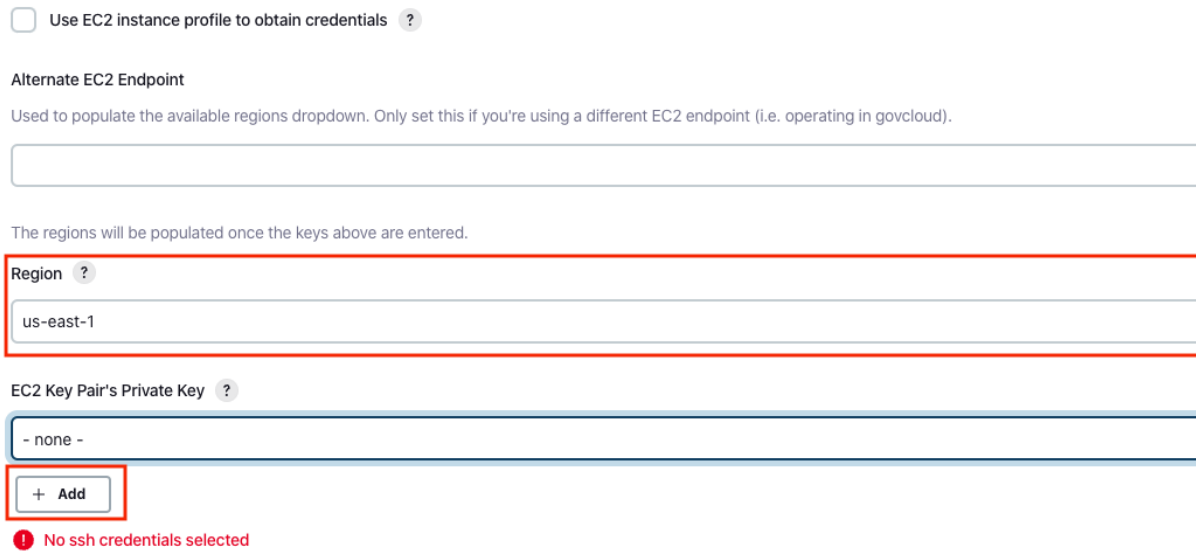
   a. From the Jenkins Credentials Provider, select AWS Credentials as the **Kind**.



b. Scroll down and enter in the IAM User programmatic access keys with permissions to launch EC2 instances and select **Add**.

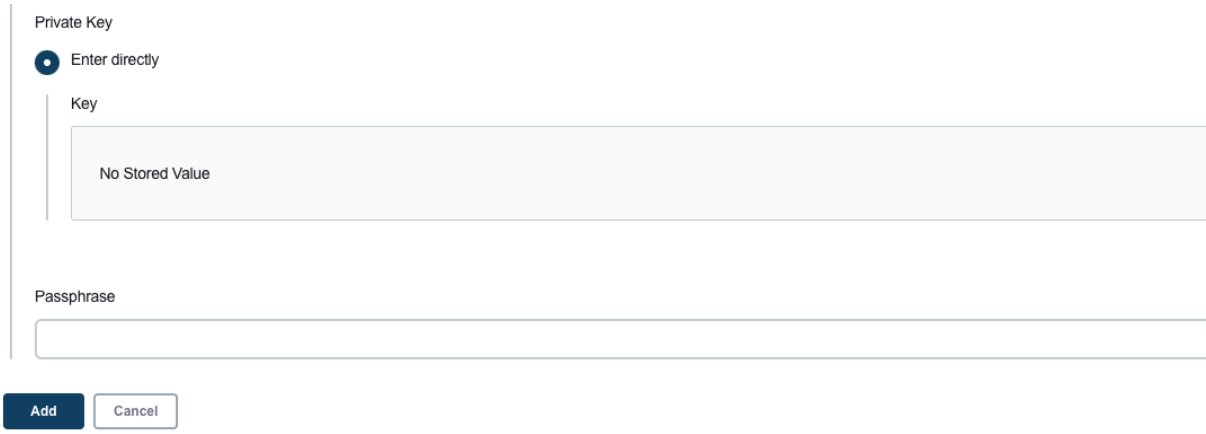


c. Scroll down to select your region using the drop-down, and select **Add** for the EC2 Key Pair's Private Key.

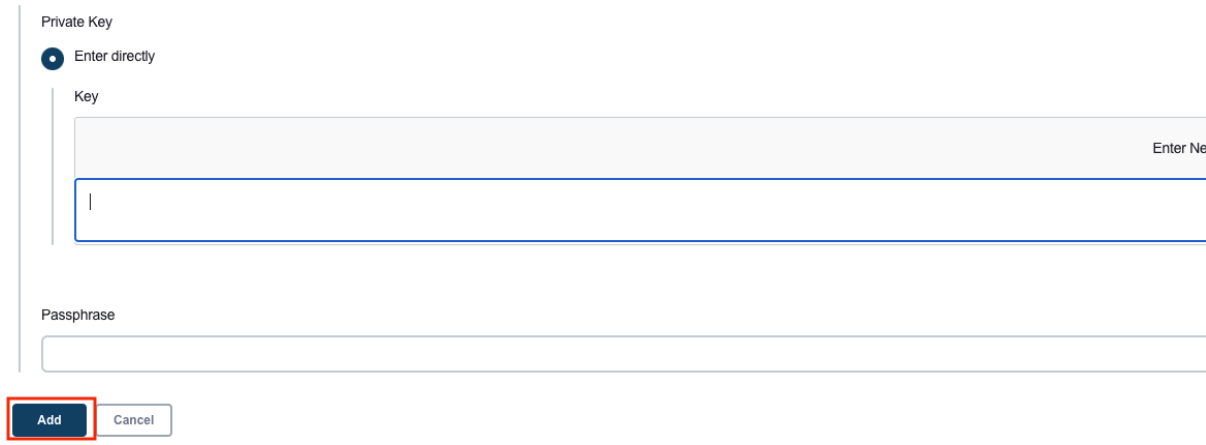


d. From the Jenkins Credentials Provider, select SSH Username with private key as the Kind and set the Username to ec2-user.

e. Scroll down and select **Enter Directly** under Private Key, then select **Add**.



f. Open the private key pair you created in the creating a key pair step and paste in the contents from "-----BEGIN RSA PRIVATE KEY-----" to "-----END RSA PRIVATE KEY-----". Select **Add** when completed.



g. Scroll down to "Test Connection" and ensure it states "Success". Select **Save** when done

You are now ready to use EC2 instances as Jenkins agents.

Cleaning up

After completing this tutorial, be sure to delete the AWS resources that you created so you do not continue to accrue charges.

Deleting your EC2 instance

1. In the left-hand navigation bar of the Amazon EC2 console, select **Instances**.
2. Right-click on the instance you created earlier, and select **Terminate**.