**This Jenkins installation is used with AWS t2.micro EC2 instances:**

1. Open a regular(free tier for example) **AWS** **EC2** **instance**.

Pay attention that those settings are configured during the instance initialization:

* **Network** **Settings** – allow HTTP/S. This will be used later for our plugins.
* **Configure** **Storage** – choose the maximal free storage which is 30GB of gp3. Expand the advanced settings and change the delete on termination to “No”.

A screenshot of a computer

Description automatically generated

Note: Once the instance will be deleted, the EBS volume will remain on cloud. In order not to pay for this storage, we will manually delete it.

* No **User** **Data** is needed.

1. Setup the **Security** **Group**:

Once the instance has been created, we can click on it and navigate to the security group that its attached to. We will add **inbound** rule for port **8200**. This port will be used to navigate Jenkins traffic.

A screenshot of a computer

Description automatically generated

Note: by default, if unset manually, Jenkins will use port 8080 to traffic its data packages.

1. Use **EC2 Instance** **Connect** to instantly SSH into the EC2 instance.

Note: from now on most of the commands require root permissions.

1. Navigate to home directory:

**cd ~**

1. Get the java version 11:

**curl -LO** [**https://corretto.aws/downloads/latest/amazon-corretto-11-x64-linux-jdk.tar.gz**](https://corretto.aws/downloads/latest/amazon-corretto-11-x64-linux-jdk.tar.gz)

1. Make a folder named java in this destination:

**mkdir /usr/share/java/**

1. Extract the downloaded Java tar content to the created folder:

**tar -xvzf amazon-corretto-11-x64-linux-jdk.tar.gz -C /usr/share/java/**

1. Enter into the created folder:

**cd /usr/share/java/amazon-corretto-\*-linux-x64/bin**

And run the version validation script:

**for i in java javac jfr; do path=$(find `pwd`/$i -type f); echo $path; sudo alternatives --install /usr/bin/$i $i $path 20000 ; sudo update-alternatives --config $i ; done**

Once done, press Ctrl+C.

1. Download Jenkins Repo:

**wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo**

1. Get the validation keys:

**rpm --import https://pkg.jenkins.io/redhat/jenkins.io-2023.key**

1. Get into the Linux distribution`s repo`s folder:

**cd /etc/yum.repos.d/**

1. And install Jenkins:

**yum install jenkins -y**

1. Enable Jenkins service(symbolic link):

**systemctl enable jenkins**

1. Optional: install font configuration`s. It’s a library for font configuration in Unix OS. Here is the command:

**yum install fontconfig -y**

1. Go back to the java folder:

**cd /usr/share/java/**

And run the Jenkins war file via the destined port:

**java -jar jenkins.war --httpPort=8200**

1. An initial password will be generated at the end, so copy it.
2. To enter the Jenkins CI tool, go to your EC2 Instance and copy the **public** **IP**. The correct URL should look like:

<http://3.249.129.148:8200>

Then paste the copied initial password and follow the steps for the first configuration.

1. Configure the node as a runner/executor:

On the main Jenkins UI click on:

**Manage Jenkins 🡪 Nodes(System Configuration) 🡪 Configure Monitors**

And tick all the “Don’t mark agents temporarily offline”:

A screenshot of a chat

Description automatically generated

1. Logout from Jenkins server.
2. Go to the **EC2** **CLI** of our Jenkins server and turn off swap space:

**swapoff -a**

Note: the time for the execution of this command heavily depends on RAM and free space on the root dir. But, it should be fast because it’s a new instance.

1. Run Jenkins war at background:

**nohup java -jar jenkins.war --httpPort=8200 &**

Simultaneously check that the Jenkins UI is up. Once done, you can Ctrl+C in the EC2 CLI safely while Jenkins will still run.

1. Return to our Jenkins UI and enter the credentials.
2. Go to our Node:

**Manage Jenkins 🡪 Nodes(System Configuration) 🡪 CLICK ON THE NAME OF THE NODE**

And finally click on the blue button:

***Bring this node up***

Once done, in the main page you will see:

A screenshot of a computer

Description automatically generated