**involving Jenkins, Terraform, and Ansible**

* **New Requirement:**  
  We need to **upgrade the Jenkins server** today.
* Install a specific version first.
* Then, upgrade to the latest version.
* **Note:** Share complete steps **with screenshots** of the upgrade process.
* **Quick Query:**  
  🡪We’ve been discussing **patching activities**, but what exactly is "patching"?  
  🡪Also, which type of **patch** are we applying using Ansible?

**Patching Activities: Patching** refers to the process of applying updates to software to fix bugs, improve functionality, or address security vulnerabilities. It's an essential part of maintaining system security and performance.

**Types of Patches**

* **Binary Patches**: Updating software binaries to a new version.
* **Source Code Patches**: Applying code changes to the source code to fix issues or add new features.

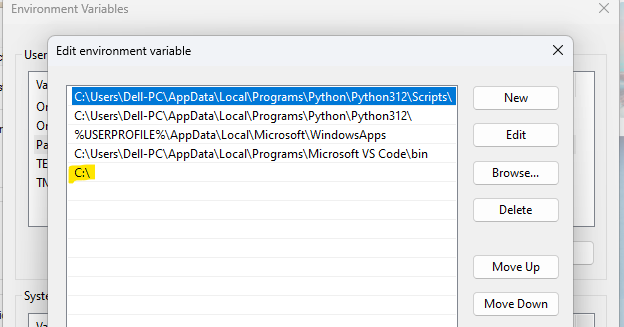
In this case, with Ansible, we are applying a **binary patch** to upgrade Jenkins to the latest version by using the apt module to update the Jenkins package.

**>>install terraform from** <https://developer.hashicorp.com/terraform/install>

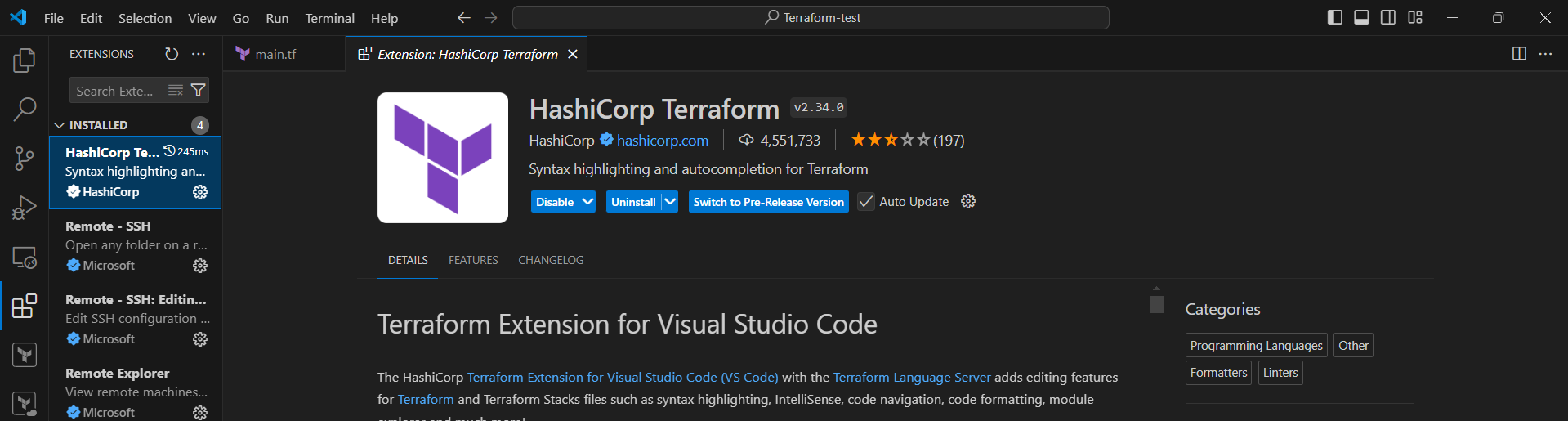
**>>after download move it to C:\Terraform in the windows machine**

**>>now go to search bar and search for edit system environment variables**

**>>add the evironment variable and select the path as C:\Terraform**

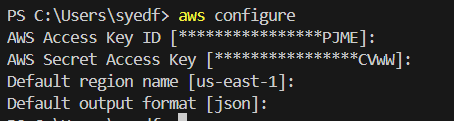


**>>go to visual code studio and add extention hashi corp by installing it**

****

**>>open it in visual code**

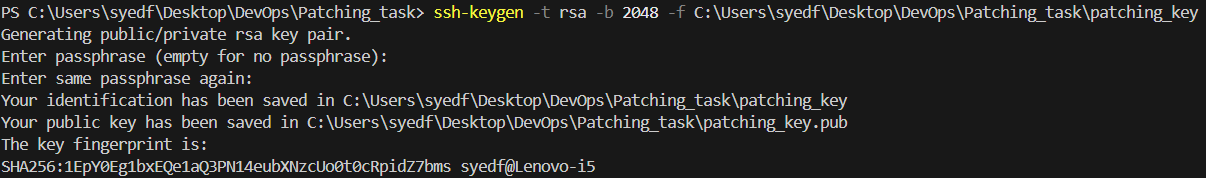
🡪download aws cli and configure it

****

**>>create a folder Patching\_task**

🡪create a public key in cerated folder itself to login to servers

# ssh-keygen -t rsa -b 2048 -f C:\Users\syedf\Desktop\DevOps\Patching\_task\patching\_key

****

**>>cerate a main.tf file in folder and write a template to launch 2 ec2 instances one would be ansible server and other would be jenkins server**

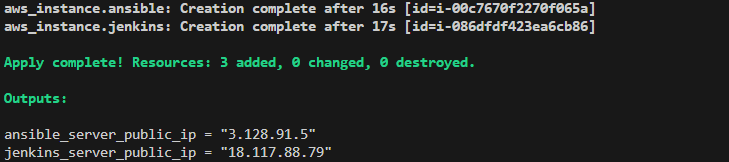
**>follow the below template**

|  |
| --- |
| provider "aws" {    region = "us-east-2"  }  resource "aws\_key\_pair" "patching" {    key\_name   = "patching"    public\_key = file("C:/Users/syedf/Desktop/DevOps/Patching\_task/patching\_key.pub")  }  resource "aws\_instance" "jenkins" {    ami                         = "ami-00eb69d236edcfaf8"    instance\_type               = "t2.micro"    key\_name                    = aws\_key\_pair.patching.key\_name    associate\_public\_ip\_address = true    tags = {      Name = "jenkins\_server"    }  }  resource "aws\_instance" "ansible" {    ami                         = "ami-00eb69d236edcfaf8"    instance\_type               = "t2.micro"    key\_name                    = aws\_key\_pair.patching.key\_name    associate\_public\_ip\_address = true    tags = {      Name = "ansible\_server"    }    user\_data = <<-EOF                #!/bin/bash                sudo apt-get update                sudo apt-get install -y ansible                EOF  }  output "ansible\_server\_public\_ip" {    value = aws\_instance.ansible.public\_ip  }  output "jenkins\_server\_public\_ip" {    value = aws\_instance.jenkins.public\_ip  } |

# terraform init

# terraform plan

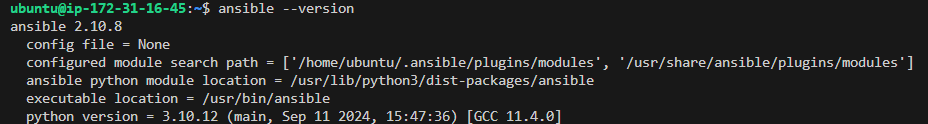
# terraform apply



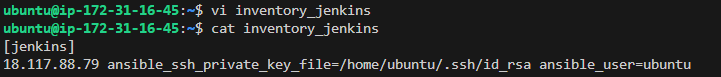
**🡪login in to ansible server and check ansible version**

PS C:\Users\syedf\Desktop\DevOps\Patching\_task> ssh -i "patching\_key" ubuntu@3.128.91.5

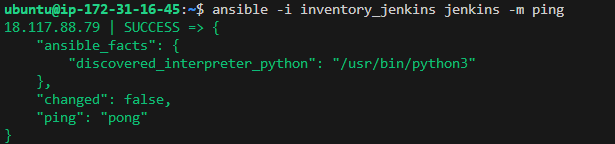
# ansible --version



🡪generate ssh key and copy it to the jenkins server in .ssh and make the inventory file



🡪check the connectivity

  
  
**🡪LETS INSTALL A SPECIFIC VERSION OF JENKINS BY ANSIBLE PLAYBOOK**

# vi jenkins\_2.249.yml

Add the following to playbook

|  |
| --- |
| ---  - name: Install a specific version of Jenkins  hosts: jenkins  become: yes  tasks:  - name: Add Jenkins repository key  apt\_key:  url: https://pkg.jenkins.io/debian/jenkins.io-2023.key  state: present  - name: Add Jenkins repository  apt\_repository:  repo: 'deb https://pkg.jenkins.io/debian-stable binary/'  state: present  - name: Install OpenJDK 11  apt:  name: openjdk-11-jdk  state: present  - name: Install specific version of Jenkins  apt:  name: jenkins=2.249.1  state: present  - name: Reload systemd manager configuration  command: systemctl daemon-reload  - name: Start and enable Jenkins service  service:  name: jenkins  state: started  enabled: yes |

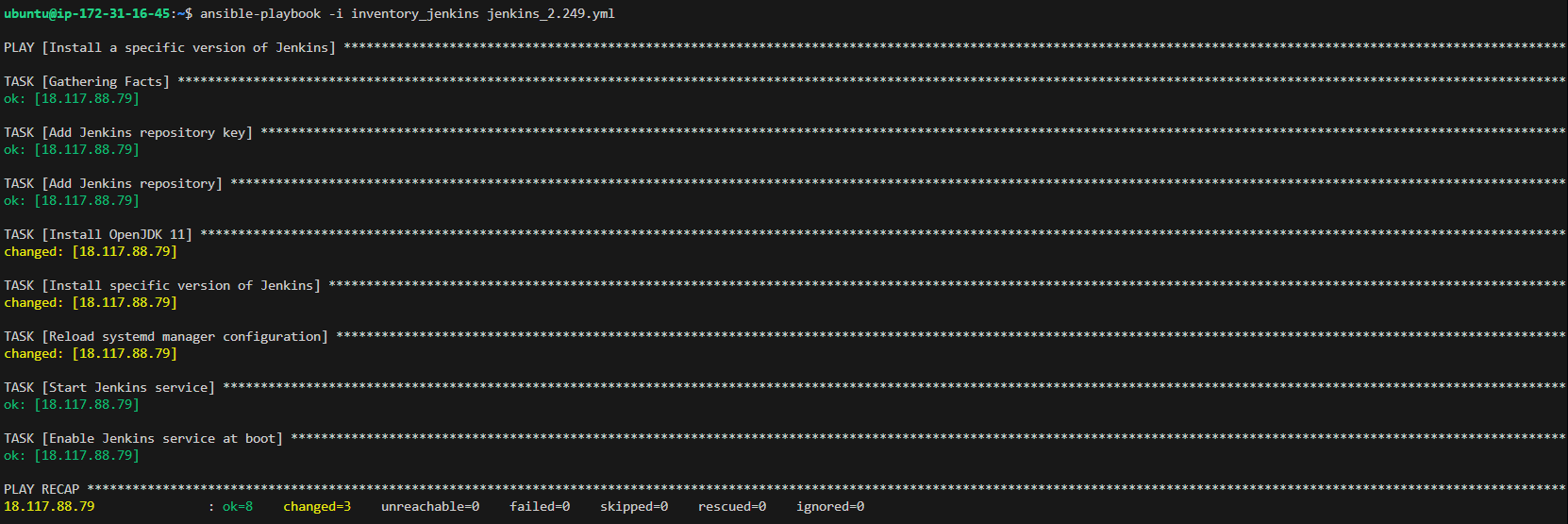
🡪check for syntax

# ansible-playbook -i inventory\_jenkins jenkins\_2.249.yml --syntax-check



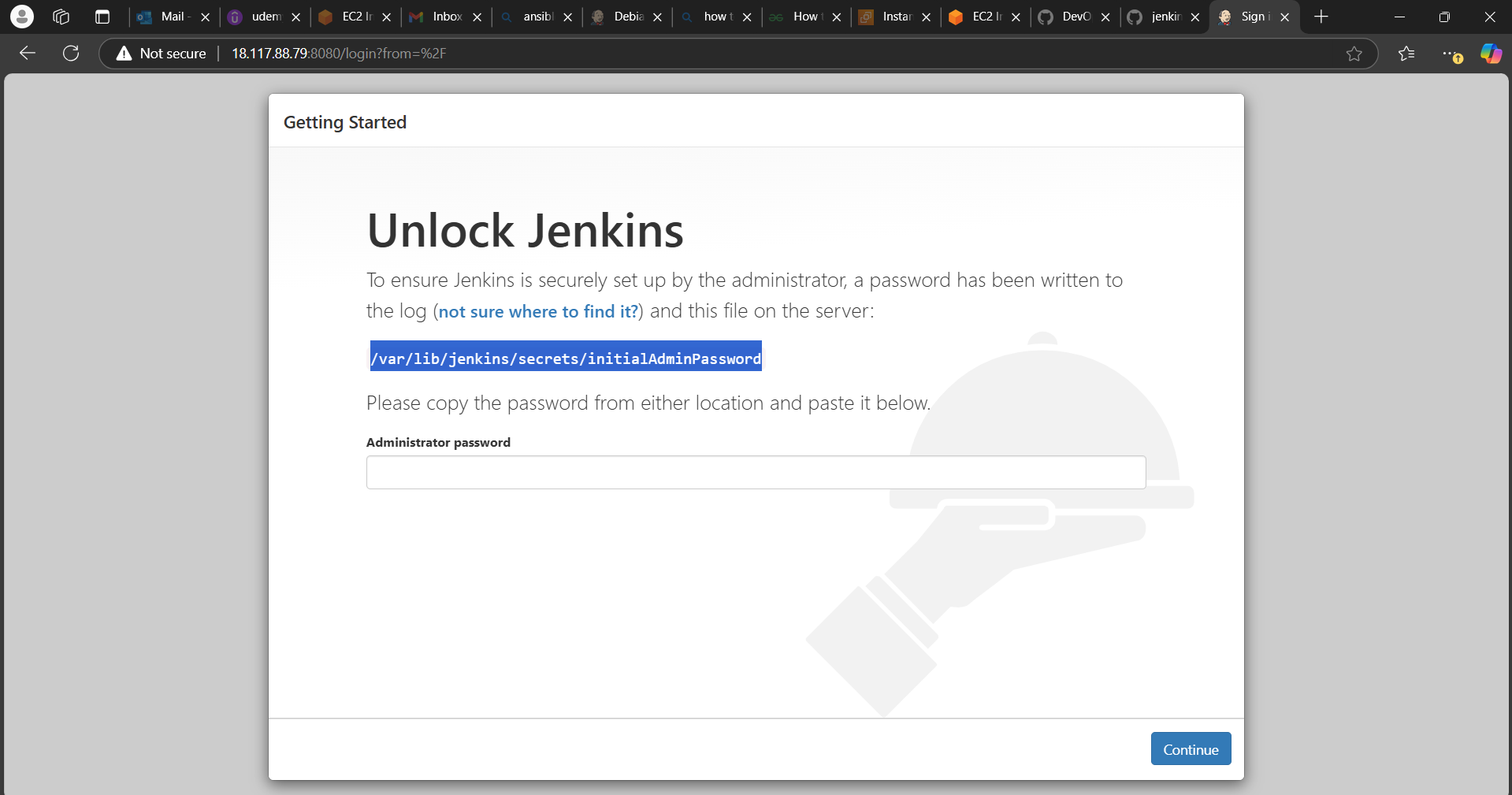
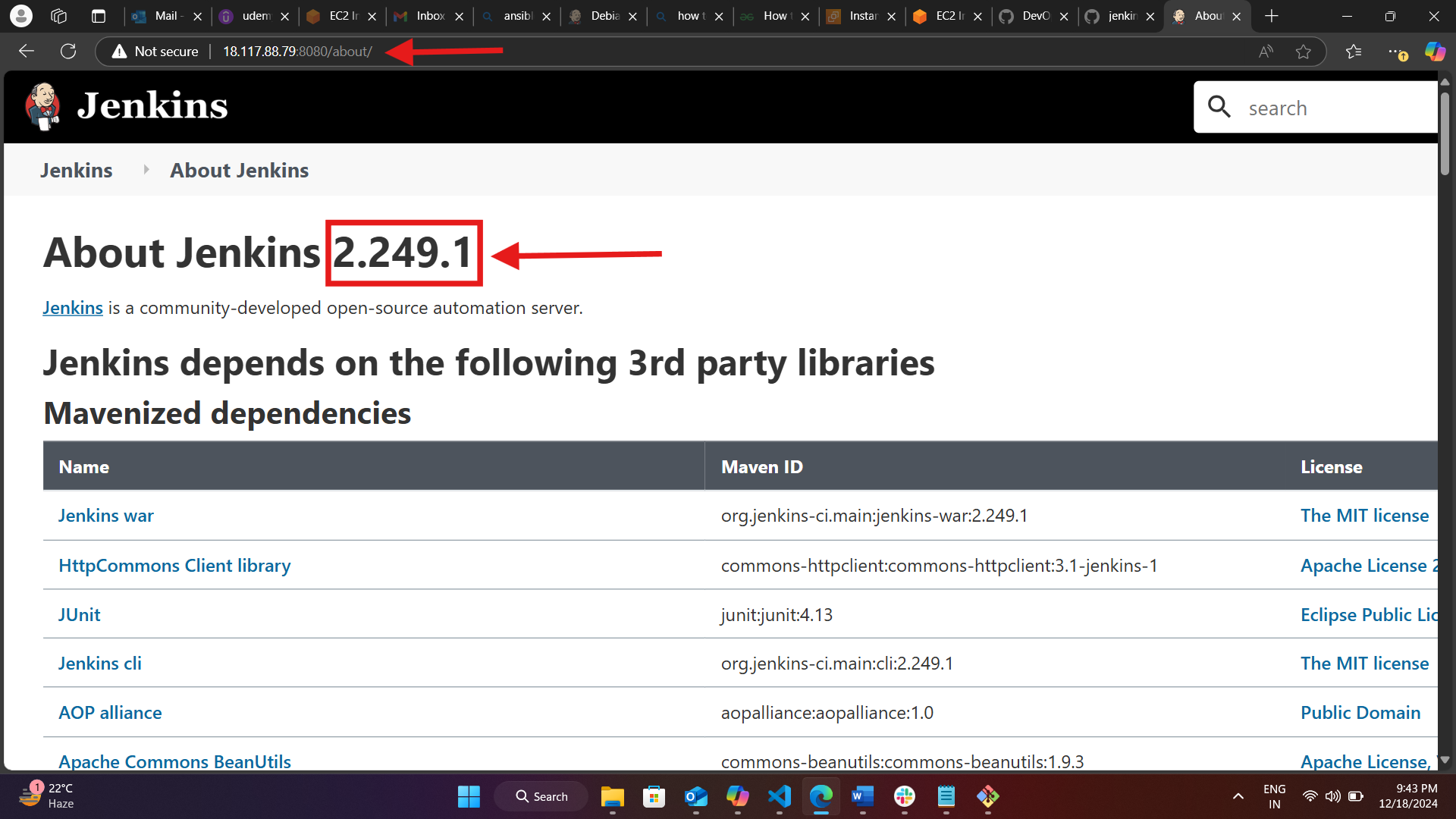
🡪execute the playbook

# ansible-playbook -i inventory\_jenkins jenkins\_2.249.yml



**🡪check on browser with public ip**

>> enter the password ~ cat /var/lib/jenkins/secrets/initia\* >>check the version manage jenkins > about jenkins

>>we have achieved the specific version of jenkins which is 2.249.1

>>now we need to continue with our patching activity which means upgrading our current version with latest

>>lets create another playbook by which this upgrade will take place

**<MAKE SURE YOU CREATE A BACKUP OF JENKINS>**

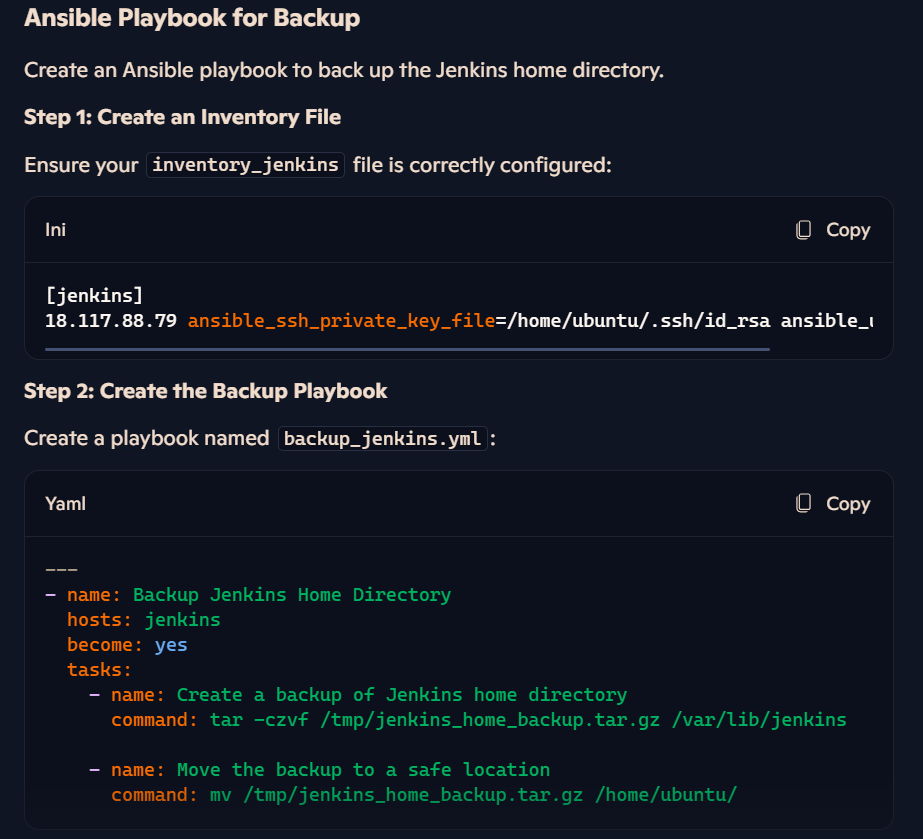
**Why Take a Backup?**

1. **Risk Mitigation: If something goes wrong during the upgrade process, you can restore the previous state without losing data or configurations.**
2. **Data Protection: Ensures that all your jobs, configurations, and plugins are safe in case of an upgrade failure.**
3. **Quick Recovery: Allows for a faster recovery process, minimizing downtime and disruptions.**

**Take an AWS Snapshot (if using AWS)**

1. **Navigate to the EC2 Dashboard in the AWS Management Console.**
2. **Select the Jenkins Instance, click on Actions, then Image and templates, and select Create image.**
3. **Follow the Prompts to create an image, which will serve as a snapshot.**

**<OR>**

****

**🡪LETS UPGRADE THE JENKINS THROUGH ANSIBLE-PLAYBOOK**

# vi patching.yml

|  |
| --- |
| ---  - name: Upgrade Jenkins and Java to the latest version  hosts: jenkins  become: yes  tasks:  - name: Update apt cache  apt:  update\_cache: yes  - name: Install latest version of OpenJDK 17  apt:  name: openjdk-17-jdk  state: latest  - name: Upgrade Jenkins to the latest version  apt:  name: jenkins  state: latest  - name: Reload systemd manager configuration  command: systemctl daemon-reload  - name: Restart Jenkins service  service:  name: jenkins  state: restarted |

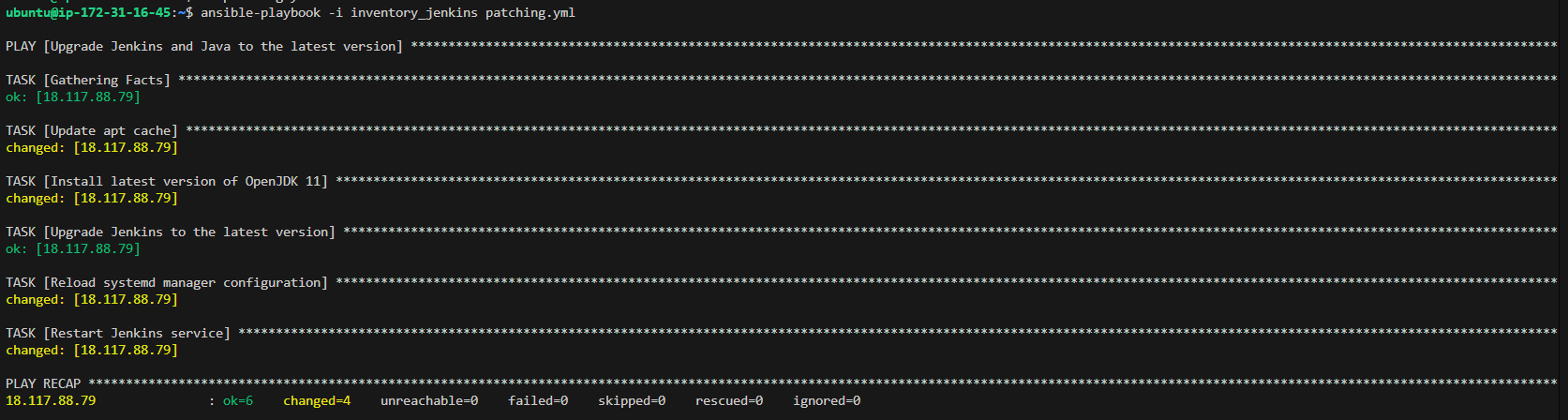
🡪check for syntax

# ansible-playbook -i inventory\_jenkins patching.yml --syntax-check



🡪execute the playbook

# ansible-playbook -i inventory\_jenkins patching.yml



**🡪check on browser with public ip**

**>>login**

**manage jenkins > about jenkins**

