**Day-10 SSH key setup & Jenkins setup**

## **1. SSH Key Setup for GitHub Authentication**

* ssh-keygen -t ed25519 -C "ranawatjinesh@gmail.com"
  + Generates a new SSH key using the **Ed25519** algorithm for secure authentication.
  + The -C flag adds an email for identification.
* eval "$(ssh-agent -s)"
  + Starts the **SSH agent**, which manages private keys for authentication.
  + **Typo in Command:** The correct command is eval "$(ssh-agent -s)".
* ssh-add ~/.ssh/id\_ed25519
  + Adds the private key to the SSH agent for use.
* cat ~/.ssh/id\_ed25519.pub
  + Displays the public key, which should be added to GitHub under **Settings → SSH Keys**.
* ssh -T git@github.com
  + Tests the SSH connection with GitHub.

Expected output:  
vbnet  
Hi <username>! You've successfully authenticated, but GitHub does not provide shell access.

* git clone git@github.com:jineshranawatcode/c406firstproject.git
  + Clones the repository using SSH authentication.

## **2. System and Network Commands**

* su - jineshtry
  + Switches to another user account (jineshtry).
* lpq
  + Checks the printer queue status.
* dmesg | tail -50
  + Displays the last 50 lines of kernel logs.
  + Useful for debugging system issues like device connections or boot errors.
* dmesg | grep "USB"
  + Filters the kernel logs to display USB-related messages.
* iperf -s -f M
  + Runs iperf in **server mode**, measuring network bandwidth in Megabytes per second (MBps).
* sudo tcpdump -i any
  + Captures network packets on all interfaces for real-time traffic monitoring.
  + Requires root privileges.
* telnet google.com
  + Attempts to connect to Google via **Telnet**, useful for checking open ports.
  + **Deprecated:** Telnet is insecure; SSH should be used instead.
* dig google.com
  + Queries the **Domain Name System (DNS)** to get information about google.com.

## **3. Installing and Setting Up Jenkins**

### **Step 1: Update Package List**

* sudo apt update
  + Refreshes package lists to ensure up-to-date installation sources.

### **Step 2: Install Java (Required for Jenkins)**

* sudo apt install openjdk-11-jdk -y
  + Installs **OpenJDK 11**, which is required to run Jenkins.
  + The -y flag automatically confirms the installation.

### **Step 3: Add Jenkins Repository**

* wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -
  + Downloads and adds the Jenkins GPG key for secure package verification.
  + **Deprecated:** apt-key is no longer recommended.
* sudo sh -c 'echo deb https://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'
  + Adds the **Jenkins repository** to the system’s package sources.

### **Step 4: Install Jenkins**

* sudo apt update
  + Updates package lists to include the newly added Jenkins repository.
* sudo apt install jenkins -y
  + Installs Jenkins.

### **Step 5: Start and Enable Jenkins**

* sudo systemctl start jenkins
  + Starts the Jenkins service.
* sudo systemctl enable jenkins
  + Enables Jenkins to start automatically on system boot.
* sudo systemctl status jenkins
  + Checks if Jenkins is running properly.
  + Expected output: active (running).

## **4. Alternative Jenkins Installation (Updated Key & Repository)**

* curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo tee /usr/share/keyrings/jenkins-keyring.asc > /dev/null
  + Downloads and adds the updated **Jenkins GPG key** to a secure location.
* echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] https://pkg.jenkins.io/debian-stable binary/" | sudo tee /etc/apt/sources.list.d/jenkins.list > /dev/null
  + Adds the Jenkins repository with **proper key verification**.

## **5. Setting Up Jenkins with Java 17**

* sudo apt install fontconfig openjdk-17-jre -y
  + Installs Java 17 and font-related dependencies.
  + Java 17 is a **Long-Term Support (LTS)** version.
* sudo apt install jenkins -y
  + Installs Jenkins.
* sudo cat /var/lib/jenkins/secrets/initialAdminPassword
  + Retrieves the **initial admin password** required to access Jenkins for the first time.
  + This password is displayed on the Jenkins login screen.

## **6. Allowing Jenkins to Use Docker**

* sudo usermod -aG docker jenkins
  + Adds the **Jenkins user** to the **Docker group**, allowing it to run Docker commands.
* sudo systemctl restart jenkins
  + Restarts Jenkins to apply the group changes.

## **7. Accessing Jenkins Web Interface**

Open a browser and go to:  
arduino  
http://localhost:8080

* Enter the **initial admin password** retrieved earlier.
* Follow the setup wizard, and **install suggested plugins** when prompted.

### **Key Takeaways**

✅ **SSH Authentication:** Used SSH keys for secure GitHub access.  
✅ **System Monitoring Commands:** dmesg, tcpdump, iperf, and dig for debugging.  
✅ **Jenkins Installation:** Installed and configured Jenkins with Java 11 & Java 17.  
✅ **Secure Key Management:** Used the **latest** Jenkins key and repository setup.  
✅ **Docker & Jenkins Integration:** Allowed Jenkins to execute Docker commands.  
✅ **Web Access:** Accessed Jenkins via http://localhost:8080 and completed the setup.