Perfect plan ✅ — you’re setting up an **Ansible lab** on AWS with 3 EC2 instances:

* 🧠 **ControlNode** → where Ansible is installed
* ⚙️ **Node1** and **Node2** → managed nodes  
  and you want all of them to connect via **SSH** and **PuTTY**.

Here’s the **complete, clean step-by-step guide** — it will work 100%.

**🧩 Step 1: Create One Key Pair (for all three)**

You’ll use one .pem key for all three instances.

1. Go to **AWS Console → EC2 → Key Pairs**
2. Click **Create key pair**
   * **Name:** ansible-key
   * **Type:** RSA
   * **File format:** .pem
3. Click **Create key pair**
   * It will download →  
     C:\Users\DELL\Downloads\ansible-key.pem

🔒 Keep this file safe — you’ll use it for SSH and PuTTY for **all 3 instances**.

**🧩 Step 2: Launch 3 EC2 Instances**

Repeat this for each instance (ControlNode, Node1, Node2):

1. Go to **EC2 → Launch Instance**
2. Enter **Name:**
   * First: ControlNode
   * Then Node1
   * Then Node2
3. Choose **AMI:**  
   👉 *Rocky Linux 9 (or Ubuntu 22.04 LTS)* — both work well with Ansible
4. Instance type: t2.micro (Free tier)
5. Key pair: select **ansible-key**
6. Network settings:
   * Allow SSH (port 22) from **Anywhere (0.0.0.0/0)**
7. Click **Launch Instance**

Wait until all 3 show **Running** ✅

**🧩 Step 3: Note Down Public IPs**

After they’re running:

1. Go to **EC2 → Instances**
2. Note the **Public IPv4 address** of each instance:

| **Instance** | **Public IP (example)** |
| --- | --- |
| ControlNode | 13.232.10.45 |
| Node1 | 13.233.20.67 |
| Node2 | 13.234.30.89 |

**🧩 Step 4: Connect via SSH (Command Prompt)**

Example for **Rocky Linux**:

ssh -i "C:\Users\DELL\Downloads\ansible-key.pem" ec2-user@13.232.10.45

For **Ubuntu:**

ssh -i "C:\Users\DELL\Downloads\ansible-key.pem" ubuntu@13.232.10.45

Repeat for each IP (Node1, Node2).

If you connect successfully — SSH ✅ works.

**🧩 Step 5: Convert PEM → PPK (for PuTTY)**

You’ll do this **once**, then use it for all 3.

1. Open **PuTTYgen**
2. Click **Load**
3. Change file type to *All Files (*.*)*
4. Browse to:
5. C:\Users\DELL\Downloads\ansible-key.pem
6. Click **Open**
7. Click **Save private key**
   * Save as: ansible-key.ppk
   * Location:
   * C:\Users\DELL\Downloads\ansible-key.ppk

**🧩 Step 6: Connect via PuTTY**

Repeat for each instance (ControlNode, Node1, Node2):

1. Open **PuTTY**
2. In **Host Name**, type:
3. ec2-user@13.232.10.45

(for Ubuntu: ubuntu@13.232.10.45)

1. On the left sidebar:
   * Go to **Connection → SSH → Auth → Credentials**
   * Under **Private key file for authentication**, browse:
   * C:\Users\DELL\Downloads\ansible-key.ppk
2. Click **Open**
3. Accept the warning
4. You’re in 🎉

**🧩 Step 7: Verify from ControlNode**

Once inside **ControlNode**, test SSH access to Node1 and Node2 (we’ll use private IPs for that later).

You can then install and set up **Ansible** on ControlNode, and manage Node1 and Node2.

**✅ Summary Table**

| **Instance** | **Connect as** | **Key** | **Works with** |
| --- | --- | --- | --- |
| ControlNode | ec2-user / ubuntu | ansible-key.pem | SSH |
| Node1 | ec2-user / ubuntu | ansible-key.pem | SSH |
| Node2 | ec2-user / ubuntu | ansible-key.pem | SSH |
| All | – | ansible-key.ppk | PuTTY |

Would you like me to continue with **Ansible setup** (ControlNode to Node1/Node2 connection + playbook test)?  
That’s the next logical step after SSH setup.

its connected. but i could 't understand please define again in clear steps with reason.

###### ChatGPT said:

Great — glad it’s connected! 🎉 Below is a **clear, step-by-step recap** of what you did, **why each step was necessary**, and what the commands actually do. I’ve kept it short and practical.

# 1) Problem (what SSH complained about)

When you first tried:

ssh -i "F:\Linux\KEYS\ansible-key.pem" ec2-user@13.203.79.178

SSH returned:

Permissions for '...ansible-key.pem' are too open.

This private key will be ignored.

Load key "...": bad permissions

Permission denied (publickey,...)

**Why:** OpenSSH requires private key files to be readable **only by the owner**. If other accounts or groups (like Authenticated Users or Everyone) can read the key, SSH refuses to use it — otherwise another user on the system could steal your key.

# 2) What you did (commands you ran)

You fixed file ACLs (permissions) with **icacls** from Command Prompt (run as Administrator):

1. Remove inherited permissions:

icacls "F:\Linux\KEYS\ansible-key.pem" /inheritance:r

**Why:** Prevents directory-level or group permissions from being inherited onto the file.

1. Grant read permission only to your Windows user:

icacls "F:\Linux\KEYS\ansible-key.pem" /grant:r "%USERNAME%:R"

**Why:** Ensures only your account can read the file (no write or modify, read only is enough for SSH).

1. Remove the Authenticated Users group if still present:

icacls "F:\Linux\KEYS\ansible-key.pem" /remove "NT AUTHORITY\Authenticated Users"

(or /remove "Authenticated Users" if the first form doesn't match your system)  
**Why:** Even if inheritance is removed, ACLs may still list groups — SSH rejects the key if such groups are present.

1. Verify final permissions:

icacls "F:\Linux\KEYS\ansible-key.pem"

**Expected output:**

F:\Linux\KEYS\ansible-key.pem YOUR-USERNAME:(R)

Successfully processed 1 files

**Why:** Confirms that only your user has read permission.

# 3) Why these steps fix the SSH error (short)

* OpenSSH checks file permissions on private keys.
* If any other user/group (Authenticated Users, Everyone, Administrators) can read the key, OpenSSH refuses to use it for security.
* Changing the ACL so that **only your Windows account** can read the file satisfies OpenSSH’s security requirement and lets it load the key.

# 4) Final connect command you used

ssh -i "F:\Linux\KEYS\ansible-key.pem" ec2-user@13.203.79.178

**Why ec2-user?** Most AWS AMIs provide a default non-root user (Amazon Linux uses ec2-user; Ubuntu uses ubuntu, etc.). You log in as that user and then use sudo to become root if needed.

After connecting, to become root:

sudo su -

# 5) Quick checklist (for future)

* Keep the .pem file private (store outside shared folders).
* Make sure only your user has read permission.
* Use the correct SSH username for the AMI.
* If using PuTTY, convert .pem → .ppk with PuTTYgen (I can give steps if you want).

If you want, I can now:

* Show the **exact expected icacls output** you should see, or
* Give the **PuTTY** conversion steps, or
* Explain how to check the EC2 instance’s key-pair in the AWS Console.

Which of those would you like next?

##### You said:

please in urdu

###### ChatGPT said:

بالکل ✅ — نیچے میں نے سادہ **اردو میں مرحلہ وار** سمجھایا ہے کہ آپ کے ساتھ کیا مسئلہ تھا، آپ نے کون سے کمانڈز چلائے، اور **کیوں** یہ سب کرنا ضروری تھا۔

## 🔹 1) مسئلہ کیا تھا؟

جب آپ نے یہ کمانڈ چلائی:

ssh -i "F:\Linux\KEYS\ansible-key.pem" ec2-user@13.203.79.178

تو یہ ایرر آیا:

Permissions for '...ansible-key.pem' are too open.

This private key will be ignored.

Permission denied (publickey,...)

### 🧠 وجہ:

SSH سیکیورٹی کے لیے یہ چیک کرتا ہے کہ **آپ کی پرائیویٹ key (.pem)** فائل صرف **آپ کے یوزر** کے پاس ہونی چاہیے۔  
اگر سسٹم کے دوسرے یوزرز یا گروپس (جیسے "Authenticated Users" یا "Everyone") کو بھی اس فائل تک رسائی ہو،  
تو SSH فوراً اسے **استعمال کرنے سے انکار کر دیتا ہے** تاکہ کوئی دوسرا بندہ آپ کی key چُرا نہ سکے۔

## 🔹 2) آپ نے کون سے کمانڈز چلائے (اور کیوں؟)

### ✅ Step 1 — Inheritance ہٹائی

icacls "F:\Linux\KEYS\ansible-key.pem" /inheritance:r

**وجہ:**  
یہ کمانڈ فولڈر کی inherited permissions ہٹا دیتی ہے،  
تاکہ فائل اپنے اوپر والے فولڈر سے permissions خود بخود نہ لے۔

### ✅ Step 2 — صرف اپنے یوزر کو Read اجازت دینا

icacls "F:\Linux\KEYS\ansible-key.pem" /grant:r "%USERNAME%:R"

**وجہ:**  
یہ کمانڈ صرف **آپ کے ونڈوز یوزر** کو “Read” (پڑھنے) کی اجازت دیتی ہے۔  
کوئی دوسرا یوزر یا گروپ اب اس فائل کو نہیں کھول سکتا۔

### ✅ Step 3 — “Authenticated Users” گروپ کو ہٹانا

icacls "F:\Linux\KEYS\ansible-key.pem" /remove "NT AUTHORITY\Authenticated Users"

یا اگر یہ نہ چلے تو:

icacls "F:\Linux\KEYS\ansible-key.pem" /remove "Authenticated Users"

**وجہ:**  
بعض اوقات “Authenticated Users” گروپ باقی رہ جاتا ہے،  
SSH جب یہ دیکھتا ہے تو کہتا ہے: "فائل دوسرے بھی پڑھ سکتے ہیں!"  
اور پھر key کو استعمال کرنے سے انکار کر دیتا ہے۔

### ✅ Step 4 — تصدیق (چیک کریں)

icacls "F:\Linux\KEYS\ansible-key.pem"

**صحیح نتیجہ:**

F:\Linux\KEYS\ansible-key.pem YOUR-USERNAME:(R)

Successfully processed 1 files

**وجہ:**  
اس سے پتا چلتا ہے کہ اب فائل صرف آپ کے یوزر کے پاس ہے اور کوئی دوسرا access نہیں رکھتا۔

## 🔹 3) یہ سب کیوں ضروری تھا؟

SSH سیکیورٹی کے لحاظ سے چاہتا ہے کہ پرائیویٹ key صرف مالک (owner) کے پاس ہو۔  
اگر کسی اور کو access ملا ہوا ہے تو سیکیورٹی رسک بنتا ہے۔  
لہٰذا ہم نے ساری اجازتیں ہٹا کر صرف اپنے یوزر کو "Read" دی —  
اور اب SSH خوش ہے 😄

## 🔹 4) آخر میں کنیکٹ کرنے کا صحیح طریقہ

ssh -i "F:\Linux\KEYS\ansible-key.pem" ec2-user@13.203.79.178

**ec2-user** زیادہ تر Amazon Linux میں استعمال ہوتا ہے۔  
اگر Ubuntu ہو تو username ubuntu ہوتا ہے۔

اور اگر آپ کو root پر کام کرنا ہو تو کنیکٹ ہونے کے بعد:

sudo su -

## 🔹 5) یاد رکھنے کی باتیں

* .pem فائل صرف **آپ کے لیے readable** ہو۔
* کسی اور گروپ یا یوزر کو access نہ ہو۔
* Amazon Linux → ec2-user  
  Ubuntu → ubuntu  
  Rocky Linux → rocky
* Root کے لیے ہمیشہ sudo su - استعمال کریں۔

Here’s the **complete step-by-step guide in English** on how to convert your .pem key file to .ppk (for PuTTY) and then use it to connect to your EC2 instance.

## 🧩 Part 1: Convert .pem to .ppk using PuTTYgen

### **Step 1 — Open PuTTYgen**

* Click **Start → type “PuTTYgen” → open it.**  
  (It comes with PuTTY. If not installed, download it from https://www.putty.org)

### **Step 2 — Load your** .pem **file**

1. In PuTTYgen, under Parameters, leave **“RSA”** selected.
2. Click **Load**.
3. At the bottom right, change file type to **“All Files (.)”**.
4. Navigate to and select your key file:
5. F:\Linux\KEYS\ansible-key.pem
6. Click **Open**.

✅ If it loads successfully, PuTTYgen will show:

“Successfully imported foreign key (OpenSSH SSH-2 private key).”

### **Step 3 — Save as** .ppk

1. Click **Save private key**.
2. When prompted:

“Are you sure you want to save this key without a passphrase?”  
Click **Yes** (unless you want to add a password).

1. Save the file as:
2. ansible-key.ppk

in the same folder (for example, F:\Linux\KEYS\).

Now your .ppk file is ready 🎉

## 🧩 Part 2: Use .ppk file in PuTTY to connect

### **Step 1 — Open PuTTY**

Go to **Start → PuTTY**.

### **Step 2 — Enter your EC2 information**

* **Host Name (or IP address):**
* ec2-user@13.203.79.178
* **Port:** 22
* **Connection type:** SSH (default)

### **Step 3 — Add your private key**

In the left-side tree menu:

Connection → SSH → Auth → Credentials

Then:

* Click **Browse**
* Select your .ppk file
* F:\Linux\KEYS\ansible-key.ppk

### **Step 4 — (Optional) Save the session**

Go back to the **Session** tab (top of the left menu).

* In “Saved Sessions,” type a name, e.g.:
* AWS-Server
* Click **Save**

Now you can easily connect later with one click.

### **Step 5 — Connect**

Click **Open**.

When you connect for the first time, PuTTY will show a message:

“The server’s host key is not cached in the registry. Are you sure you want to continue connecting?”

Click **Yes**.

You should now see:

[ec2-user@ip-172-31-xx-xx ~]$

✅ You are connected!

Cat /etc/amazon-linux-release

Cat /etc/os-release

<https://aws.amazon.com/console/> 979244568347

|  |  |
| --- | --- |
| **CHANGE ROOT PASSWORD:**  **Reboot Virtual Machine**  **Press ‘e’ when on ‘greb’ screen**  **Ctrl+i**  **Add ‘rd.break’ at the end, where line start with ‘Linux…’**  **Ctrl + x**  **>mount –o remount,rw /sysroot**  **>chroot /sysroot**  **>passwd**  **>touch /.autorelabel**  **>exit >exit** | **Web Side Project** Port 80  **$systemctl status httpd**  **$dnf install httpd –y**  **$systemctl enable httpd**  **$systemctl start httpd**  **$cd /var/www/tmp**  **$wget** [**https://freewebsitetemplates.com/download/files/lawfirm.zip**](https://freewebsitetemplates.com/download/files/lawfirm.zip)  **$sudo unzip /var/www/html/tem/lawfirm.zip –d /var/www/html/lawfirm**  **Open browser: http://3.110.207.104/lawfirm/**  **Tip: ctrl+shift+v copy the link**  **rpm -e httpd** |
| **PuTTY Session: Type: SSH / Prot: 22**  **$ sudo su –**  **# sudo su –**  **#hostnamectl set-hostname controlnode**  **#exec /bin/bash**  **PuTTY ISSUE:**  **#systemctl stop sshd**  **#systemctl status sshd**  **#systemctl start sshd** | **Partition / Add Disk**  Setting—storage--+add Harddisk—create—1gb—choose  #dh -hT  #lsblk  #fdisk /dev/sdb (p-print, w-write, n-new/add, q-quit)  n (press enter)  partition: 1  w (press enter)  >partition has been created  #mkfs.ext4 /dev/sdb1 (create file system)  #mkdir -p /mnt/data  #mount /dev/sdb1 /mnt/data  #df -hT |
| **Hard Link and Soft Link:**  #ln original.txt hdlink.txt (hard link)  #ln -s original.txt softlink.txt (soft link)  **Inode:** unique identifier in linex file system, every file and directory having unique Inode.  **Hard link**: create the same inode# of both files. If original.txt deleted then other remains.  **Soft link:** it create the different inode# of both files. If orginal.txt deleted then other files contents also deleted.  **Note:** in different file system like xfs and ext the hard link not created. | **N**  **P**  **Enter or 1**  **Enter**  **Enter**  **Type: t**  **L**  **8e**  **W**  **Fdisk -l**  **=now create physical value**  **Pvcreate /dev/sdb** |
|  |  |

|  |  |
| --- | --- |
| **Cnode#ansible –version**  **$ su –**  **# ls /etc**  **#ls /etc | grep ansible**  **#dnf install ansible-core -y**  **#cd /etc/ansible**  **#ll (no file )**  **---create two files**  **#vi ansible.cfg**  **[defaults]**  **#inventroy = ./inventory**  **host\_key\_checking = false**  **remote\_user=ansadm**  **ask\_pass=false**  **[privilage\_escalation]**  **become=true**  **become\_method=sudo**  **become\_user=root**  **become\_ask\_pass=false**  **:wq!**  **#vi hosts**  **[nitclasses]**  172.31.37.118  172.31.46.203  **[app]**  **node1.example.com**  **node2.example.com**  **:wq!**  **#cd /etc**  **#vi hosts (add below lines)**  **127.0.0.1 localhost localhost.localdomain localhost4.localdomain4**  **::1 localhost6 localhost6.localdomain6**  **172.31.37.118 node1.example.com node1**  **172.31.46.203 node2.example.com node2**  **:wq1**  **Testing:**  **#cat /etc/hosts**  **Ping -c3 node1**  **Ping -c 3 node2**  **Ping -c 3** 172.31.37.118  **Ping -c 3** 172.31.46.203 | **Ssh connection password less**  **All this machines one user**  **#id ansadm**  **#useradd ansadm**  **#echo “123” | passwd - - stdin ansadm**  **#echo “ansadm ALL=(ALL) NOPASSWD:ALL” | tee /etc/sudoers.d/ansadm ansadm**  **#su – ansadm**  **$ssh-keygen**  **$ssh-copy-id**  **Now connect this control node to node1 and node2**  **Switch to node1**  **#id ansadm**  **#useradd ansadm**  **#echo “123” | passwd - - stdin ansadm**  **#echo “ansadm ALL=(ALL) NOPASSWD:ALL” | tee /etc/sudoers.d/ansadm ansadm**  **Switch to node2**  **#id ansadm**  **#useradd ansadm**  **#echo “123” | passwd –stdin ansadm**  **#echo “ansadm ALL=(ALL) NOPASSWD:ALL” | tee /etc/sudoers.d/ansadm ansadm**  **Switch to control node**  **#su – ansadm**  **$ssh-keygen**  **Enter, enter, enter**  **$ssh-copy-id ansadm#172.31.37.118** |

LVM is used to manage volume and disk on the lInux server.

Logical volume Manager allows disks to be

combined together.

umount /dev/sdb1 /mnt/part1

mount -a

mount

lsof /mnt/part1

fuser -vm /mnt/part1

cd /mnt/part1

vi /etc/fstab

comments /mnt/part1 or # the line

lsblk

umount /dev/sdb2 /mnt/part2

partprobe

mount -a

mount -av

cd

fdisk /dev/sdb

commnad: d

d

d

p

n