**Server setup using virtual environment: VMware workstation 17 pro**

* Prepare the Environment

1. Physical Server: A dedicated machine.
2. Virtual Machine (VM): Hosted on a hypervisor like VMware.
3. Cloud Instance: AWS EC2.

* Download and Install CentOS

1. Initial Server Configuration
2. Configure Basic Networking
3. Create Group and User Accounts for Developers
4. Set Up SSH Keys for Secure Access

* Configure the Development Environment

1. Install Additional Packages((e.g., PHP, Node.js, Python, depending on your development stack):

* Set Up a Shared Directory
* Configure Git for Version Control
* Monitoring and Maintenance

**Step 1: Download VM workstation 17 pro (Follow the link** [**https://getlabsdone.com/vmware-workstation-pro-is-now-free-for-personal-use-steps-to-download/**](https://getlabsdone.com/vmware-workstation-pro-is-now-free-for-personal-use-steps-to-download/)**)**

**Step 2: Download Centos stream 9** (<https://centos.org/download/>)  in Architectures click on

[X86\_64](https://mirrors.centos.org/mirrorlist?path=/9-stream/BaseOS/x86_64/iso/CentOS-Stream-9-latest-x86_64-dvd1.iso&redirect=1&protocol=https)

**Step 3: Install VMware workstation 17 pro using centos**

* From downloads open VMware workstation, follow the installation process by clicking next
* After installation, reboot your system if prompted.
* Click on Create a New Virtual Machine.
* Choose Typical (recommended) for the configuration type.
* Select the option Installer disc image file (ISO) and browse for the CentOS ISO file from downloaded.

**Configure the Virtual Machine:**

* Select Linux as the guest operating system and choose CentOS as the version.
* Assign a name to your virtual machine and choose a location to store the virtual machine files.
* Allocate enough RAM (e.g., 2 GB or more) and CPU cores for your virtual machine based on your hardware capacity.
* Set up the virtual disk size (e.g., 20 GB or more).
* Review and confirm your settings, then click Finish.

**Install CentOS on the Virtual Machine:**

* Power on the virtual machine.
* Follow the CentOS installation process by selecting your language, setting up partitions, configuring network settings, and creating a root user.
* After installation, the virtual machine will reboot and you will have CentOS installed on your virtual machine.

**Step 4: Login as root user with password**

* # whoami (it should prompt root)
* # ip addr (to check the ip address)
* # nmcli con show (check the network connections)

**Step 5: Configure Basic Networking**

* **Assign Static IP Address: 192.168.1.100/24**

# nmcli con mod "ens33" ipv4.addresses "192.168.1.100/24"

# nmcli con mod "ens33" ipv4.gateway "192.168.1.1"

# nmcli con mod "ens33" ipv4.dns "8.8.8.8"

# nmcli con mod "ens33" ipv4.method manual

# nmcli con up "ens33"

* **Enable SSH:**

#systemctl start sshd

#systemctl enable sshd

#firewall-cmd --permanent --add-service=ssh

#firewall-cmd --reload

**Step:6 Create User Accounts for Users**

* **Add Users: Create individual user accounts for each user**

# useradd developer1

# passwd developer1

* **Add Users to the wheel Group (Optional): This gives them sudo privileges.**

# usermod -aG wheel developer1

**Step 7:Set Up a Shared Directory**

* **Create a Shared Directory:**

            # mkdir /srv/shared-dev

* **Change Ownership and Permissions:**

            # chown -R developer1:developer1 /srv/shared-dev

            # chmod 775 /srv/shared-dev

            Navigate to # cd  /srv, check ownership and permissions # ls -la

([root@admin srv]# ls -la

total 0

drwxr-xr-x.  3 root       root        24 Sep  4 15:11 .

dr-xr-xr-x. 18 root       root       235 Sep  4 12:50 ..

drwxr-xr-x.  2 developer1 developer1   6 Sep  4 15:11 shared-dev)

**Step:8 Add Users to a Group and grant group access:**

# groupadd devgroup

#usermod -aG devgroup developer1

#chown -R :devgroup /srv/shared-dev

#chmod 2775 /srv/shared-dev

**Permissions:**

* The 2 is the setgid (Set Group ID) bit.
* 7 (Owner): Read, write, and execute (rwx).
* 7 (Group): Read, write, and execute (rwx).
* 5 (Others): Read and execute (r-x).

**Effect:**

* Setgid Bit: When the setgid bit is set on a directory, any new files or subdirectories created inside this directory will inherit the group of the directory rather than the default group of the user who created the file. This is useful in shared directories because it ensures that all files and subdirectories maintain the same group ownership.
* Ownership and permissions remain the same as chmod 775, but with the added benefit that the group ID will stay consistent across all files and subdirectories within /srv/shared-dev.

**Step 8: Monitoring and Maintenance**

* **Install Monitoring Tools** (e,g htop, nload, fail2ban

# yum install ngios

* **Regular Updates: Ensure the system is updated:**

# yum update -y

**Setup web server : Install apache or nginx on server**

 # yum install httpd

(The error message indicates that Apache (httpd) cannot start because port 80 is already in use by another process)

* Identify the process using port 80

# sudo ss -tuln | grep :80

* Change Apache's listening port

# sudo vi /etc/httpd/conf/httpd.conf

Listen 80 to Listen 8080

Add or modify the ServerName directive to localhost

 # sudo systemctl restart httpd

 # systemctl enable httpd

 # firewall-cmd --permanent --add-service=http

 # firewall-cmd --reload

**Setup SSH Keys for Secure Access on each developer / user:**

 # ssh-keygen -t rsa

* Copy SSH Keys to Server:

            # ssh-copy-id developer1@IP

* Using SSH login to developer machine by root

# ssh 'developer1@192.168.109.101'

**Accessing the Shared Directory Using NFS: .**

**Step 1: Install NFS Server on the Server Machine**

* Install NFS server:

              # sudo yum install nfs-utils

* Start and enable NFS services:

              # sudo systemctl start nfs-server

              # sudo systemctl enable nfs-server

* Export the shared directory: Edit the /etc/exports file and add the following line to share

            /srv/shared-dev 192.168.109.101(rw,sync,no\_root\_squash)

* Export the file system:

            # sudo exportfs -a

* Open firewall for NFS (if applicable):

            #sudo firewall-cmd --permanent --add-service=nfs

            #sudo firewall-cmd --reload

**Step 2: Install NFS Client on the developer1**

* Install the NFS client:

            # sudo yum install nfs-utils

**Connecting vm to windows system:**

Make sure both VM and Windows systems are on same subnet

* On windows system open command prompt

# ipconfig (check the ip address, gateway address and subnet)

* On VM

1. Edit network connection:

# nm-connection-editor (edit the gateway, subnet that should be in same range as windows) after editing save and close.

1. Change Ip address to the same range as windows

# nmcli con mod "ens33" ipv4.addresses "192.168.1.151/24"

1. Check ip

# ip addr

1. Check connection

            # nmcli con show

1. Reload the connection after changes

# nmcli con up "ens33"

**Try pinging to windows IP**

1. # ping <windows system ip>

**Still getting erro**r

Open control panel>system and security>windows defender firewall> advanced settings> inbound rule>file and printer sharing(echo request-ICMPv4 in)>right click>enable.

**Again try to ping**

1. Ping <windows ip> (from vm to windows)
2. Ping -c 2 <vm ip> (from windows system to vm)

[root@localhost /]# ping -c 2 192.168.0.0

PING 192.168.0.0 (192.168.0.0) 56(84) bytes of data.

--- 192.168.0.0 ping statistics ---

2 packets transmitted, 0 received, 100% packet loss, time 1057ms