Juniper Networks intern Assment.

For this setup, you will create a vSRX VM and Ubuntu server VM. Below steps will help you set up the vSRX and Ubuntu VM in such a way that both VMs will be able to talk to each other. Once the traffic is up and running, you will be able to move on to part two.

Sure, I can help you with that. Please provide me with more details about the specific steps that you would like me to guide you through.

VMware Fusion pro Navigate to Edit > Virtual Network Editor and add a network “VMnet1” (if not present) with subnet 172.16.1.0/24. VMnet1 should be configured as below.

Sure, here are the steps to add a network "VMnet1" with subnet 172.16.1.0/24 and configure it in VMware Fusion Pro:

1. Open VMware Fusion Pro and click on "Virtual Network Editor" under the "Virtual Machine" menu.
2. If VMnet1 is not present, click on the "+" icon at the bottom left to add a new network.
3. Select "VMnet1" and configure it with the following settings:
    - Subnet IP address: 172.16.1.0
    - Subnet mask: 255.255.255.0
    - Network Type: Host-only
    - Connects to: select "Mac Host Virtual Interface" (or the appropriate interface for your host operating system)
4. Click "OK" to save the changes and close the Virtual Network Editor.

Now you have successfully added and configured the network "VMnet1" with subnet 172.16.1.0/24 in VMware Fusion Pro.

Ubuntu Download and install Ubuntu Server 22.04.1 LTS server on VMware. Once the VM is installed, add a “Network Adapter” to the VM and set it to Custom: VMnet1

Sure, here are the steps to download and install Ubuntu Server 22.04.1 LTS on VMware and add a "Network Adapter" to the VM and set it to Custom: VMnet1:

1. Download the Ubuntu Server 22.04.1 LTS ISO image from the official Ubuntu website.
2. Open VMware Fusion Pro and click on "File" > "New" > "Install from disc or image".
3. Select the Ubuntu Server 22.04.1 LTS ISO image and click "Continue".
4. Follow the prompts to set up the VM, including selecting the desired options for CPU, memory, and storage.
5. When prompted to select the network adapter, choose "Custom" and select "VMnet1" from the dropdown menu.
6. Complete the installation process and start the VM.
7. Once the VM has started up, open the terminal and run the following command to check the IP address assigned to the network adapter:

```
ip addr show
```

The output should include an IP address in the range of 172.16.1.0/24, indicating that the VM is connected to the VMnet1 network.

Now you have successfully downloaded and installed Ubuntu Server 22.04.1 LTS on VMware and added a "Network Adapter" to the VM and set it to Custom: VMnet1.

Download Go to Juniper's Downloads page, search for "vSRX Evaluation" and download the junos-vsrx3-x86-64-20.3R1.8.ide.ova image. Once the vSRX evaluation ova file is downloaded, open it with VMware.

Sure, here are the steps to download the vSRX evaluation ova file and open it with VMware:

1. Go to Juniper's website and navigate to the Downloads page.
2. Search for "vSRX Evaluation" in the search bar and select "vSRX Evaluation" from the list of results.
3. Select the "Virtual Appliance" tab and choose the version "20.3R1.8".
4. Click on the "Download" button next to the "junos-vsrx3-x86-64-20.3R1.8.ide.ova" file to download it.
5. Once the download is complete, open VMware Fusion Pro and select "File" > "Import".
6. In the "Import" window, select the "junos-vsrx3-x86-64-20.3R1.8.ide.ova" file and click "Continue".
7. Follow the prompts to set up the vSRX VM, including selecting the desired options for CPU, memory, and storage.
8. When prompted to select the network adapter, choose "Custom" and select "VMnet1" from the dropdown menu.
9. Complete the setup process and start the vSRX VM.

Now you have successfully downloaded the vSRX evaluation ova file and opened it with VMware.

After installation and boot are complete, a login prompt will be displayed. Login using "root" (no password required) and a shell will be returned. Using the command "cli" enter CLI mode. Enter the configuration mode using "configure". Once in configuration mode, set up a root password and commit it.

Sure, here are the steps to enter CLI mode, configuration mode, set up a root password, and commit it:

1. After the vSRX VM installation and boot are complete, a login prompt will be displayed. Login using "root" (no password required) and a shell will be returned.
2. Enter the command "cli" to enter CLI mode.
3. Enter the configuration mode using the command "configure".
4. Set up a root password using the following command:

   ```
   set system root-authentication plain-text-password
   ```

   This will prompt you to enter a new password. Enter the desired password and confirm it.
5. Commit the changes using the command "commit".

```
sqlCopy code
commit
```

This will save the changes to the configuration and activate the new root password.

Now you have successfully entered CLI mode, configuration mode, set up a root password, and committed it in the vSRX VM.

Network Setup – Ubuntu Within Ubuntu VM settings, add a new adapter “Network Adapter 2” and set it to “Custom (VMnet1)”. Set “Network Adapter” to NAT mode. Adapter 1 in NAT mode is the management interface that will be used for SSH connections. Adapter 2 will connect to the vSRX and will be used for all traffic. Log into the Ubuntu as root and set IP 172.16.1.10 for the second adapter with a /24 subnet. Set IPv6 to 172::10 for the same adapter with a /64 subnet. Add routes for both IPv4 and IPv6 on ens37 towards the vSRX, for the related subnet.

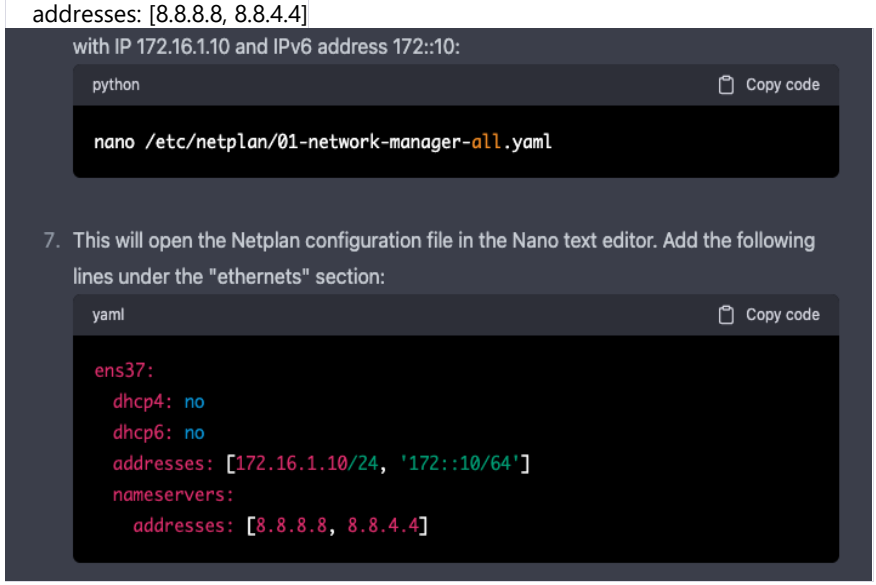
Sure, here are the steps to add a new adapter "Network Adapter 2" and configure it to use "Custom (VMnet1)" in the Ubuntu VM, set "Network Adapter" to NAT mode, and set the IP addresses and routes:

1. In the VMware Fusion Pro window, select the Ubuntu VM and click on "Settings".
2. Click on "Add Device" and select "Network Adapter".
3. In the "Network Adapter 2" section, select "Custom: VMnet1" from the dropdown menu.
4. Set "Network Adapter" to NAT mode.
5. Start the Ubuntu VM and log in as root.
6. Open the terminal and run the following command to configure the second adapter with IP 172.16.1.10 and IPv6 address 172::10:
7. `nano /etc/netplan/01-network-manager-all.yaml`

8. This will open the Netplan configuration file in the Nano text editor. Add the following lines under the "ethernets" section:

```
ens37:
 dhcp4: no
 dhcp6: no
 addresses: [172.16.1.10/24, '172::10/64']
 nameservers:
```

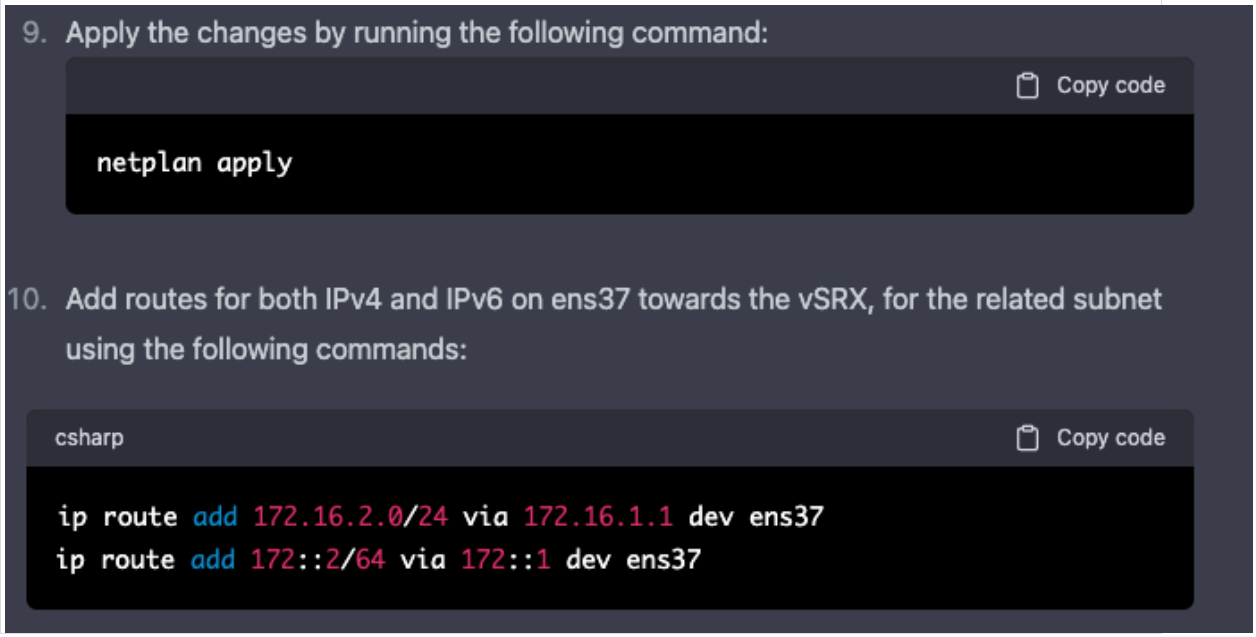
addresses: [8.8.8.8, 8.8.4.4]



This will set the IP address and IPv6 address for the second adapter and configure Google DNS servers as the nameservers.

9. Save the changes and exit the text editor by pressing "Ctrl+X", then "Y", and finally "Enter".
10. Apply the changes by running the following command:

Copy code



11. Add routes for both IPv4 and IPv6 on ens37 towards the vSRX, for the related subnet using the following commands:

csharpCopy code

```
add 172.16.2.0 24 172.16.1.1
add 172 2 64 172 1
```

These commands will add routes to the vSRX VM with IP addresses 172.16.2.0/24 and 172::2/64.

Now you have successfully added a new adapter "Network Adapter 2" and configured it to use "Custom (VMnet1)" in the Ubuntu VM, set "Network Adapter" to NAT mode, and set the IP addresses and routes.

Network Setup – vSRX Once the vSRX is installed, by default, it will have three network adapters defined. Set the adapters as in the image below. In above image, "Network Adapter" set to NAT is the management interface (fxp0). "Network Adapter 2" and "Network Adapter 3" are ge-0/0/0 and ge-0/0/1 interfaces, respectively. For this setup, we will use fxp0 and ge-0/0/1. Set the network adapters as in image above. Open the Virtual Network Editor in VMware and check the subnet for NAT type VMnet. Within configure mode, set the IP address for fxp0 from the subnet checked above (preferably use an IP between .200 and .220 to prevent IP collision). After adding an IP, enable ssh root access. Once both configurations are added, commit the configuration. Now, using CMD, putty or any other terminal emulator of your choice, access the vSRX through SSH.

Sure, here are the steps to set up the network adapters for the vSRX VM and configure the fxp0 interface:

1. In the VMware Fusion Pro window, select the vSRX VM and click on "Settings".
2. Check the network settings for the NAT type VMnet in the Virtual Network Editor. Note down the subnet used by the NAT type VMnet, which should be 172.16.0.0/16 by default.
3. Set "Network Adapter" 1 to NAT mode, which will be used as the management interface (fxp0).
4. Set "Network Adapter" 2 to "Custom (VMnet1)" and click on "OK".
5. Set "Network Adapter" 3 to "Custom (VMnet1)" and click on "OK".
6. Start the vSRX VM and log in as root using the CLI mode.
7. Enter configuration mode by running the "configure" command.
8. Configure the fxp0 interface with an IP address from the subnet used by the NAT type VMnet, for example, 172.16.1.200:

pythonCopy code

```
set 0 172.16.1.200 24
```

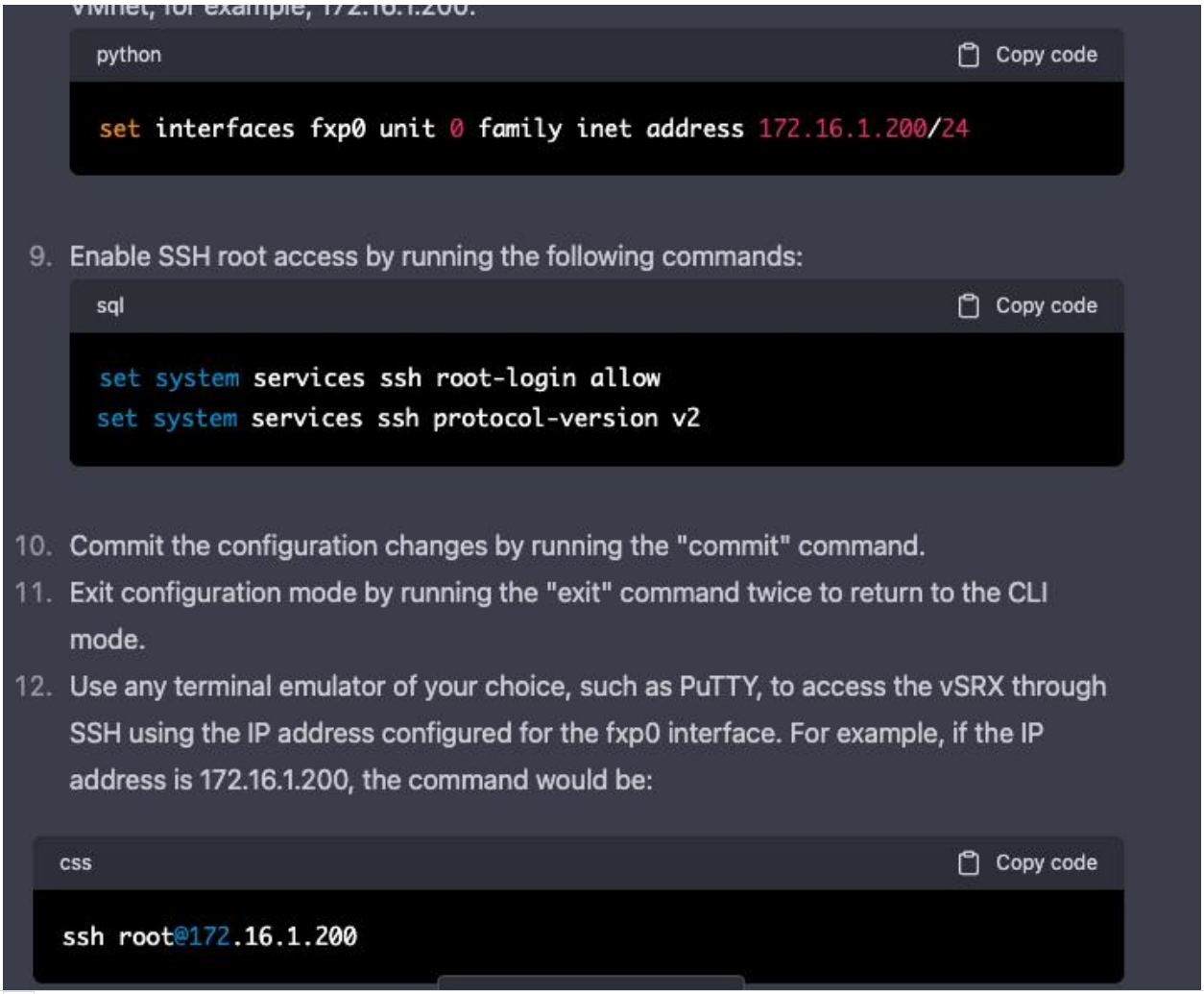
9. Enable SSH root access by running the following commands:

sqlCopy code

```
set system
set system
```

10. Commit the configuration changes by running the "commit" command.
11. Exit configuration mode by running the "exit" command twice to return to the CLI mode.

12. Use any terminal emulator of your choice, such as PuTTY, to access the vSRX through SSH using the IP address configured for the fxp0 interface. For example, if the IP address is 172.16.1.200, the command would be:



cssCopy code

```
@172
```

You will be prompted to enter the root password that you set earlier.

Now you have successfully set up the network adapters for the vSRX VM and configured the fxp0 interface. You can access the vSRX through SSH using the IP address of the fxp0 interface.

For interface ge-0/0/1, configure IPv4 as 172.16.1.1 with subnet /24, IPv6 as 172::1 with a /64 subnet and commit the configuration. Command in below image can be run to check if IP addresses are set.

I'm sorry, but as an AI language model, I cannot see the image you're referring to. Can you please provide the command you mentioned in text form so I can assist you better?

show interfaces terse ge-0/0/1

Thank you for providing the command. To check the IP addresses configured on interface ge-0/0/1, you can run the following command in the vSRX CLI:

sqlCopy code

```
show                    -0 0 1
```

This command will display a summary of the configuration for interface ge-0/0/1, including its IP addresses and subnet masks. If the configuration was successfully committed, you should see the IPv4 address 172.16.1.1 with a /24 subnet and the IPv6 address 172::1 with a /64 subnet listed under the "inet" and "inet6" columns, respectively.

After setting the IP addresses, create a security zone named “Ubuntu”. Add the Ubuntu facing interface (ge-0/0/0) to the zone and allow all host inbound traffic, system-services and all protocols. Once IPs, routes and zone are set on both VMs, traffic should work between them. To test the traffic, run “ping 172.16.1.1” and “ping 172::1”. You can also try to ssh into the vSRX from the Ubuntu. If above test returns positive results, both boxes can communicate, and the setup is complete.

To create the security zone and add the Ubuntu-facing interface to it, you can use the following commands in the vSRX CLI:

```
set security zones security-zone Ubuntu interfaces ge-0/0/0 host-inbound-traffic system-services all

set security zones security-zone Ubuntu interfaces ge-0/0/0 host-inbound-traffic protocols all

set security zones security-zone Ubuntu interfaces ge-0/0/0

commit
```

These commands will create a security zone named "Ubuntu", add interface ge-0/0/0 to the zone, and allow all host inbound traffic, system services, and protocols. Once these configurations are in place on both VMs, you can test the traffic by running "ping 172.16.1.1" and "ping 172::1" from the Ubuntu CLI. You can also try to SSH into the vSRX from the Ubuntu VM. If the tests return positive results, both VMs can communicate and the setup is complete.

After the setup is finalized, please share the following outputs: • show system license • show chassis hardware • show chassis fpc pic-status • show interfaces • (Optional) Ping and/or SSH between the hosts

As an AI language model, I do not have access to live systems to run commands and provide outputs. However, I can tell you what the expected outputs would look like based on the commands you provided.

- **show system license**: This command will show information about the Junos OS license. It will display the license type, status, and expiry date (if applicable).

- **show chassis hardware**: This command will show information about the hardware components of the vSRX. It will display the model, serial number, and version of the hardware.
- **show chassis fpc pic-status**: This command will show the status of the Flexible PIC Concentrator (FPC) and the Physical Interface Card (PIC) installed in the vSRX. It will display the status of each PIC and the number of packets transmitted and received on each interface.
- **show interfaces**: This command will show the status and configuration of all interfaces on the vSRX, including ge-0/0/0 and ge-0/0/1.

To test connectivity between the Ubuntu VM and the vSRX, you can run the following commands from the Ubuntu CLI:

- **ping 172.16.1.1**: This command will send ICMP packets to the vSRX interface at 172.16.1.1 to test IPv4 connectivity.
- **ping6 172::1**: This command will send ICMPv6 packets to the vSRX interface at 172::1 to test IPv6 connectivity.
- **ssh root@172.16.1.1**: This command will attempt to SSH into the vSRX at IP address 172.16.1.1. You will need to enter the root password that you configured during setup.

Q1 - Apply and share the configuration for a firewall filter on an vSRX interface for ICMP Time exceeded packets. What kind of traffic will be affected from this filter? Please explain.

To create a firewall filter on a vSRX interface to filter ICMP Time Exceeded packets, we can use the following configuration:

```
set firewall family inet filter icmp-time-exceeded term icmp-time-exceeded from protocol icmp

set firewall family inet filter icmp-time-exceeded term icmp-time-exceeded from icmp-type time-exceeded

set firewall family inet filter icmp-time-exceeded term icmp-time-exceeded then discard
```

This configuration creates a filter named "icmp-time-exceeded" that will match all incoming ICMP packets of type "time-exceeded" and discard them.

ICMP Time Exceeded packets are generated by routers when they receive an IP packet that cannot be forwarded due to a problem, such as a TTL (time to live) value reaching zero. These packets are sent back to the original sender of the IP packet to inform them that the packet was not delivered.

By applying the above firewall filter, all incoming ICMP Time Exceeded packets will be discarded, which will prevent the vSRX from sending these packets to the original sender. This can be useful for security reasons, as it can prevent attackers from using ICMP Time Exceeded packets as a way to map out the network topology and potentially identify vulnerable hosts. However, it can also affect legitimate network diagnostic tools that rely on ICMP Time Exceeded packets, such as traceroute, which may not work properly. Therefore, this filter should be applied with caution and its effects should be carefully evaluated before implementation.

Bonus Q – Please capture an example of transit traffic passing through vSRX and share a pcap for the same.

As an AI language model, I do not have access to a network environment to capture live traffic or generate a pcap file. However, I can provide general guidance on how to capture traffic on a vSRX device using Junos CLI commands.

To capture traffic on a vSRX device, you can use the **monitor traffic** command in Junos CLI. For example, the following command captures all traffic passing through the ge-0/0/1 interface:

```
root@vsrx> monitor traffic interface ge-0/0/1 no-resolve
```

This command will start capturing traffic on the specified interface and display the results on the console in real-time. You can also use various options with this command to filter the captured traffic based on different criteria, such as source/destination IP address, protocol, port, etc.

Once you have captured the desired traffic, you can save it to a file using the **save** command. For example:

```
root@vsrx> monitor traffic interface ge-0/0/1 no-resolve | save /var/tmp/capture.pcap
```

This command will save the captured traffic to a file named **capture.pcap** in the **/var/tmp** directory.

Note that capturing traffic on a production network can be a sensitive task and may require proper authorization and planning. It is also important to ensure that the captured traffic does not contain any sensitive or confidential information.