

### Configure VPN between Mumbai and Ohio Lab 3 of 4

Go to Security Group “Mumbai\_Linux\_sec\_Group”.

Click “Edit “ and then click “Add Rule”.

Allow all traffic from 192.168.0.0/16 subnet.

**Edit inbound rules** [X]

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ	Description ⓘ	
All traffic ▼	All	0 - 65535	Custom ▼ 192.168.0.0/16	VPN Traffic	✕
SSH ▼	TCP	22	Custom ▼ 0.0.0.0/0	SSH Access	✕

**Add Rule**

NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

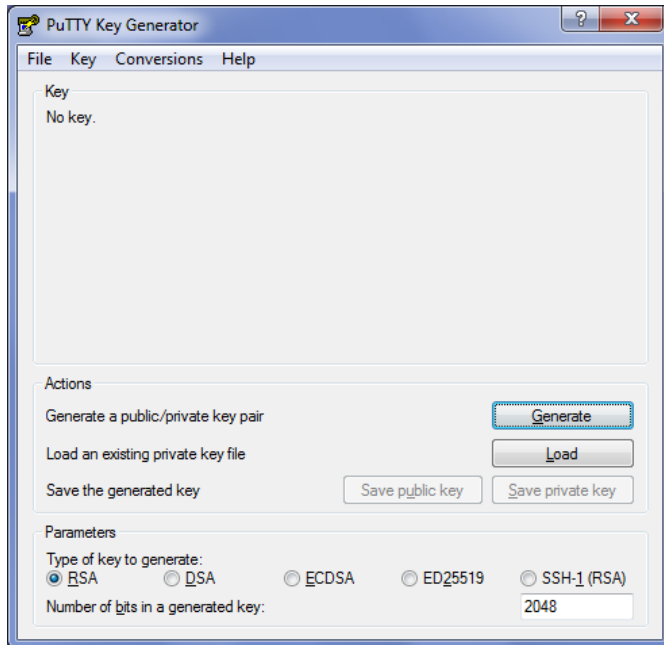
[Cancel](#) [Save](#)

Then click save.

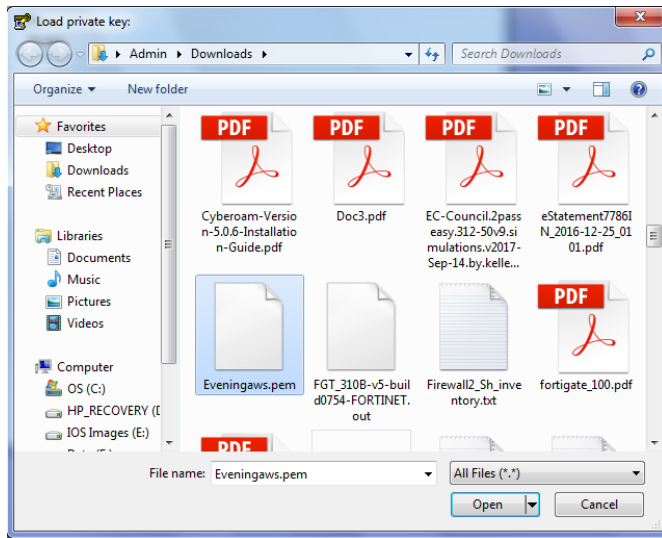
Goto Mumbai region to get public ip address of VPN Server Interface (13.127.161.231)

Launch putty key generator in your local machine,

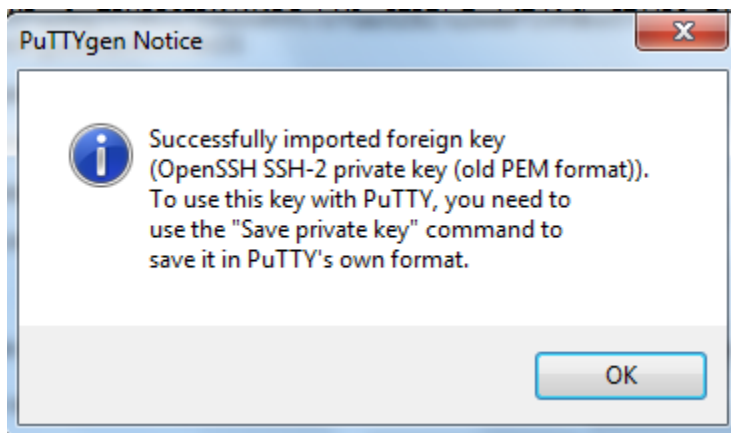
In File → Load private key



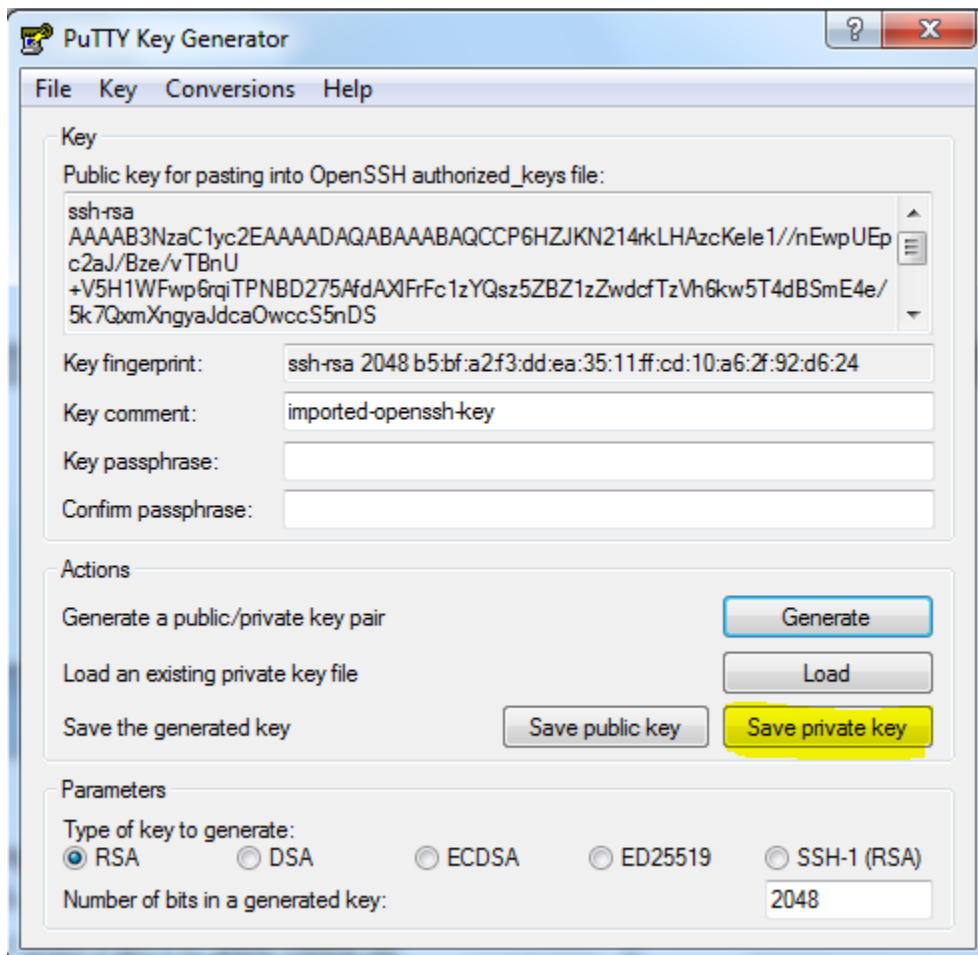
Locate the \*.pem file and click "open".



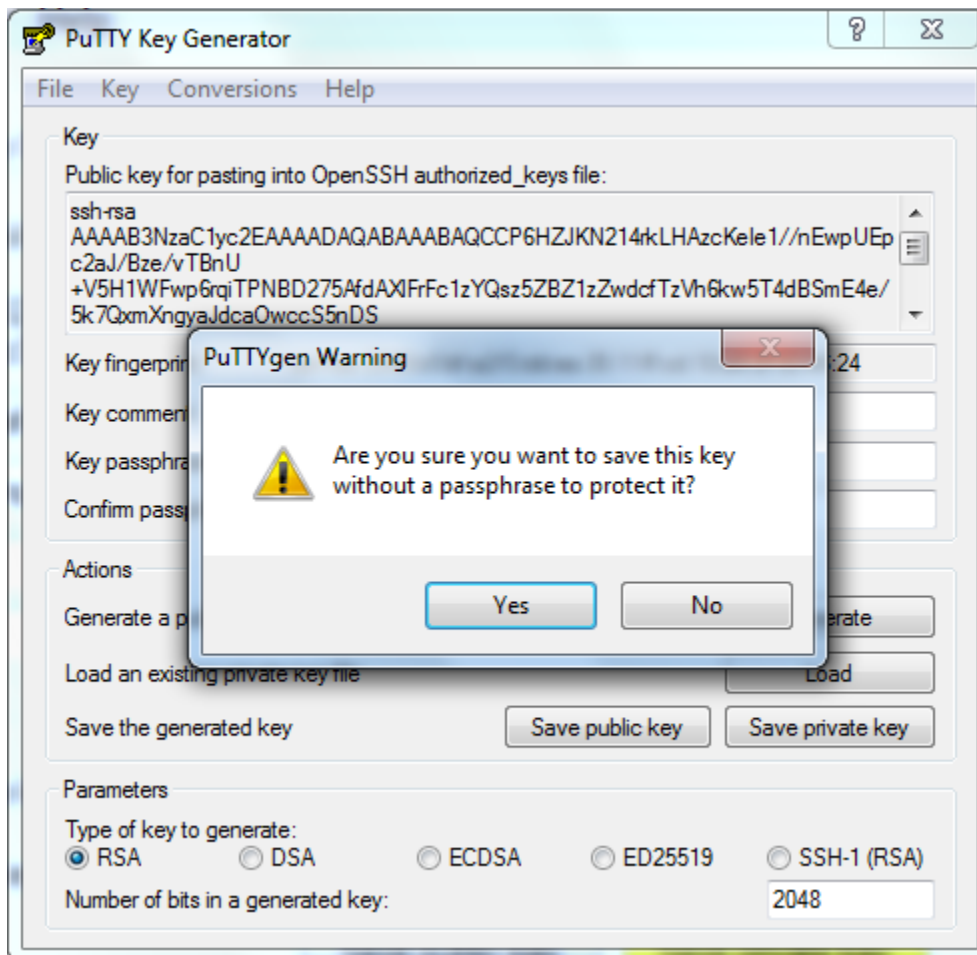
Click "Ok:".



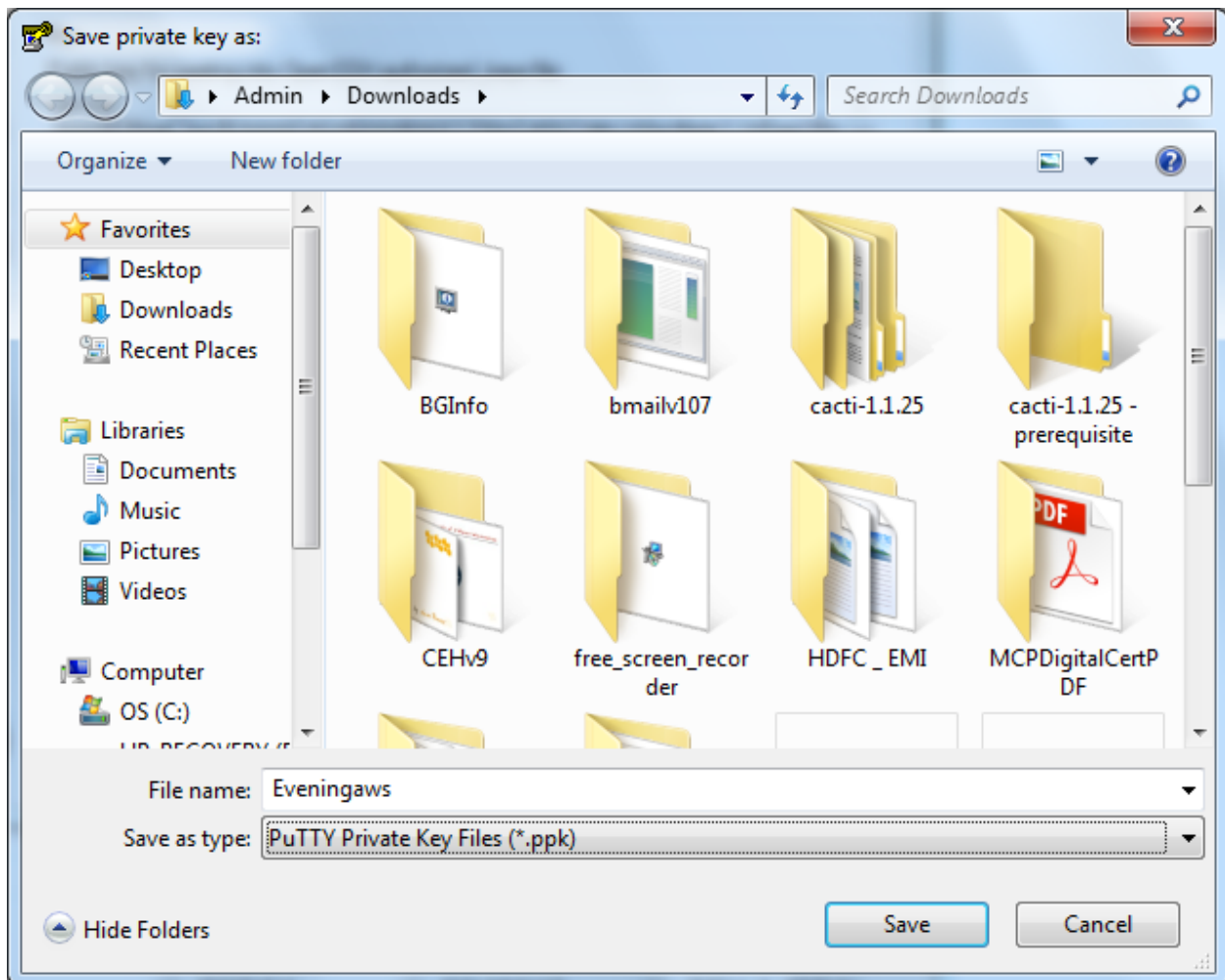
Click "save Private Key".



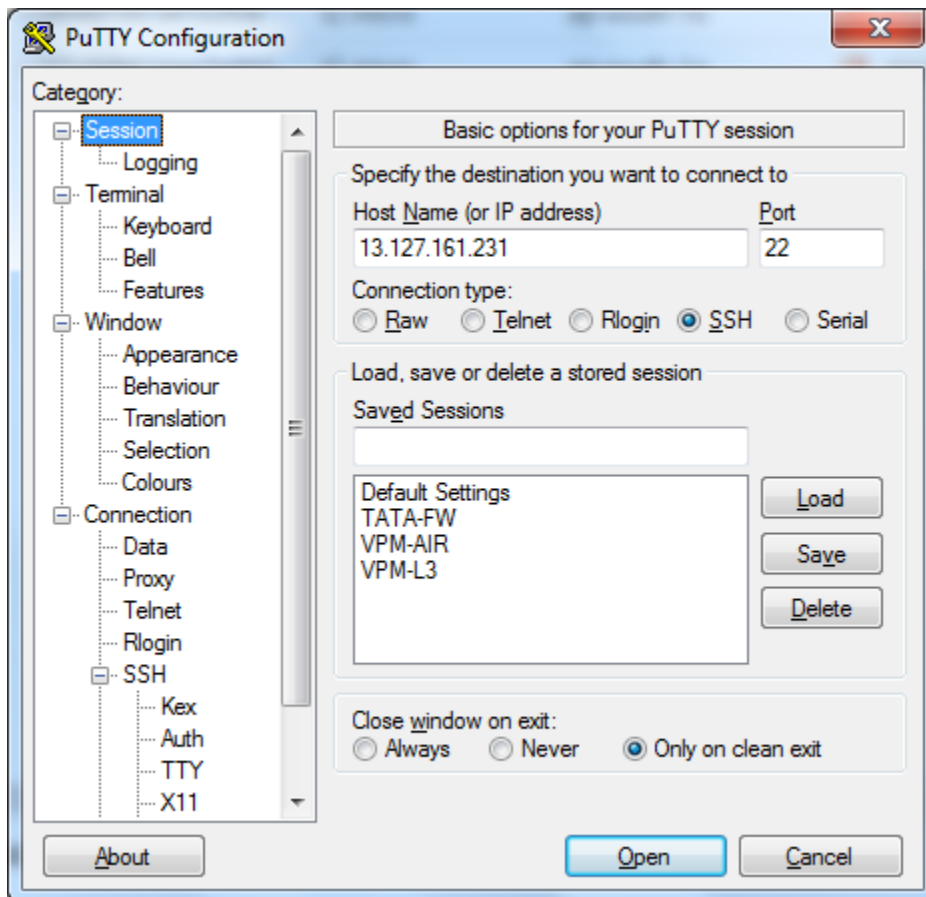
Click “Yes”.



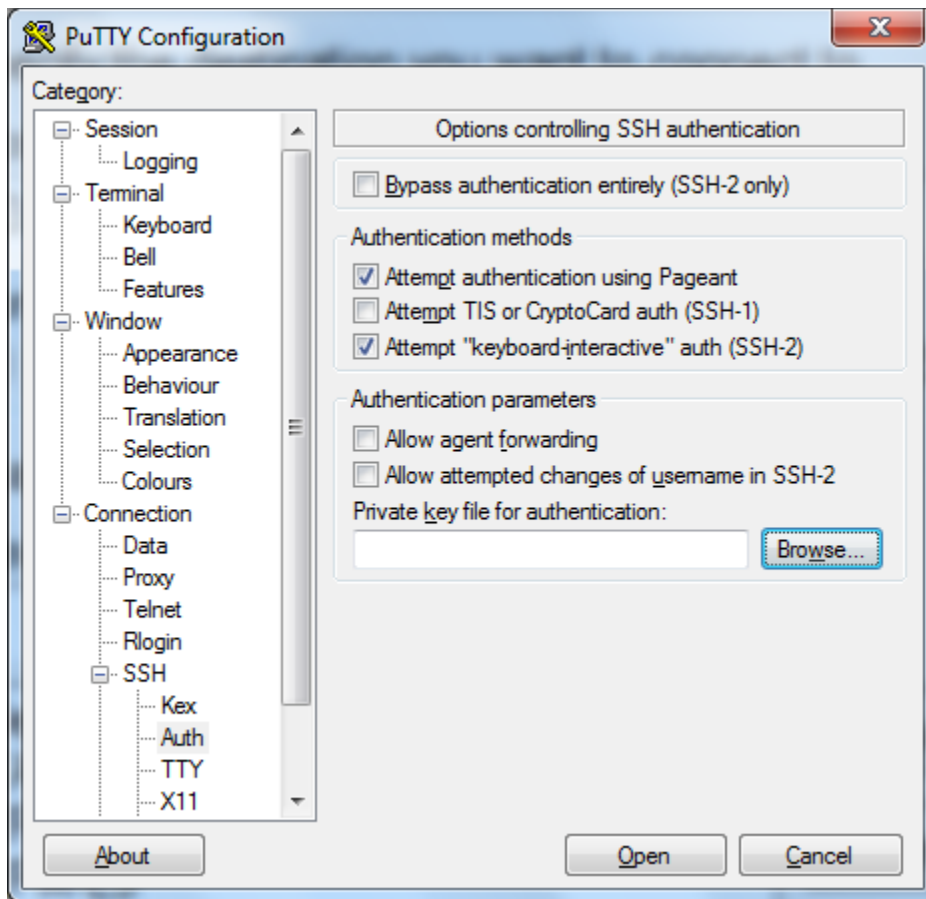
Save the private key in location.



Type the ip address in putty.

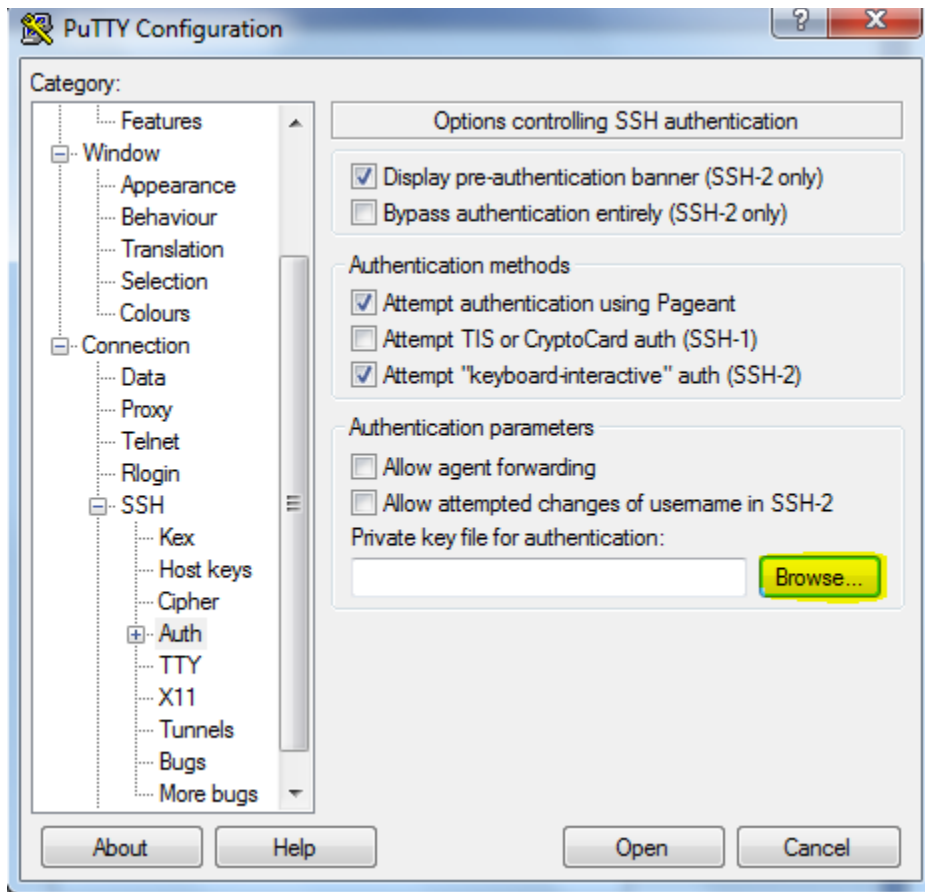


Click SSH and expand it click “Auth”.

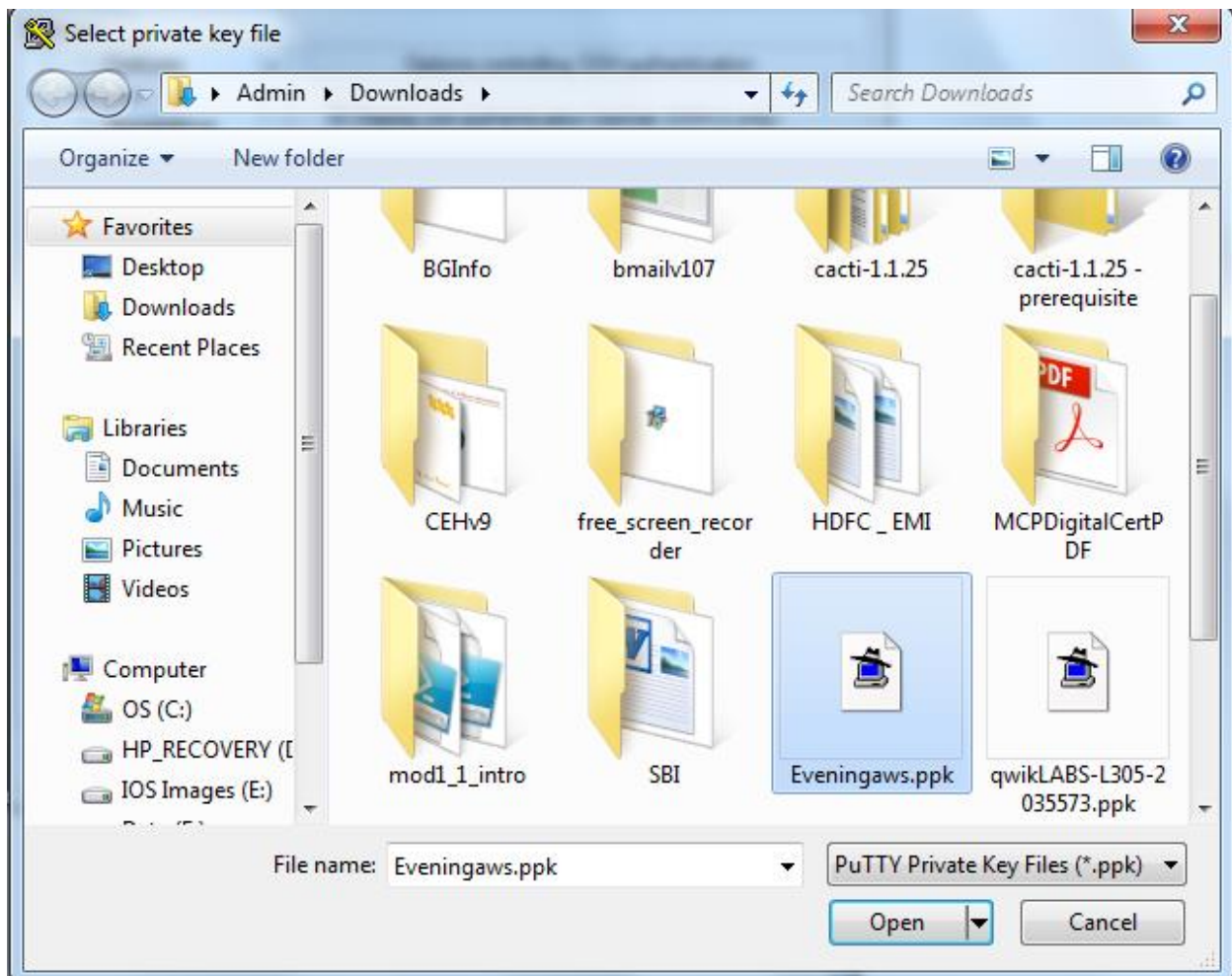




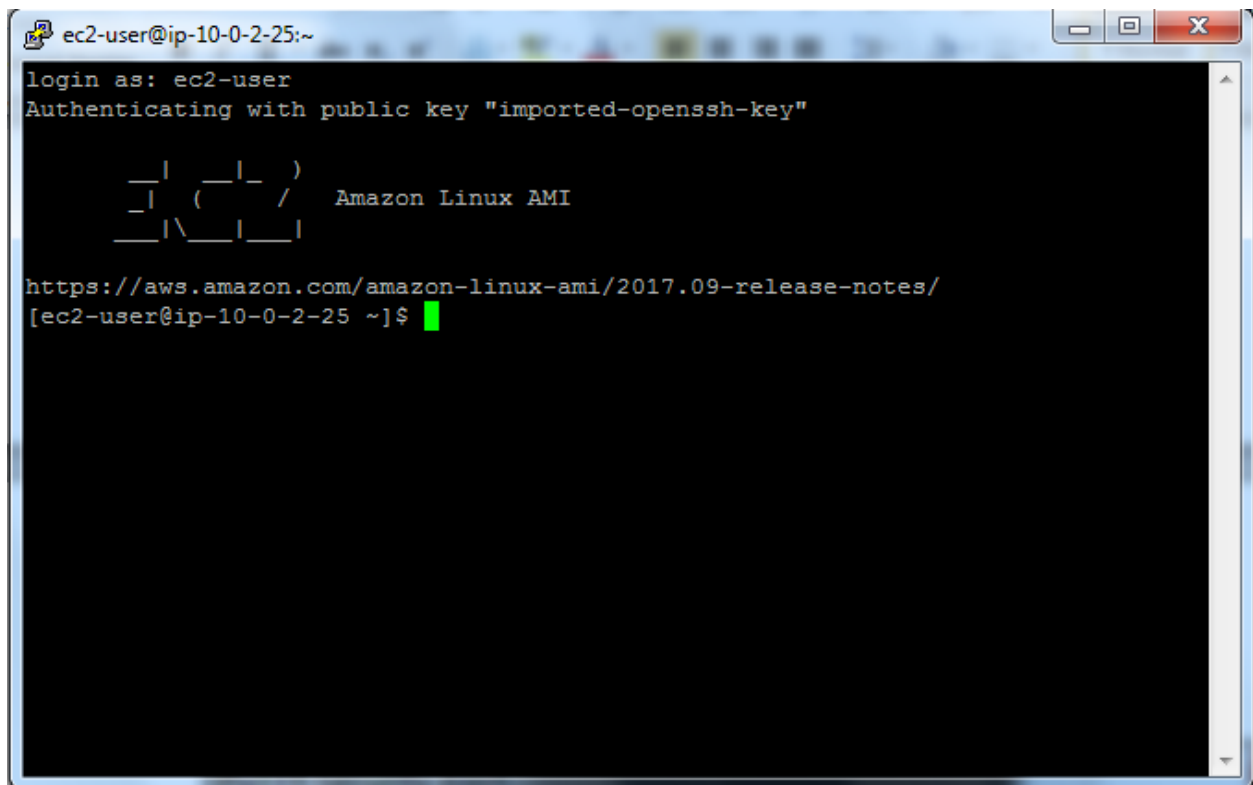
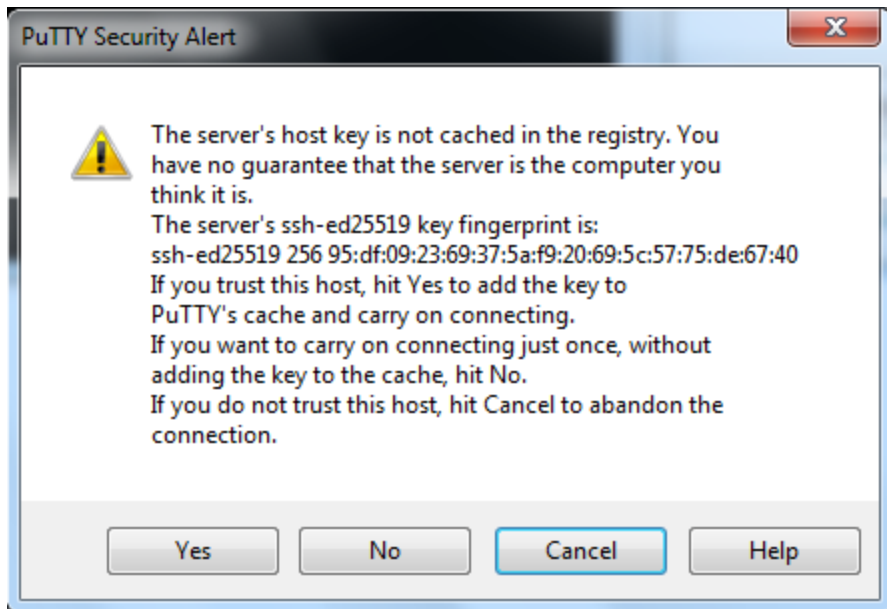
Click browse and locate the \*.ppk file.



Locate the file and click "Open".

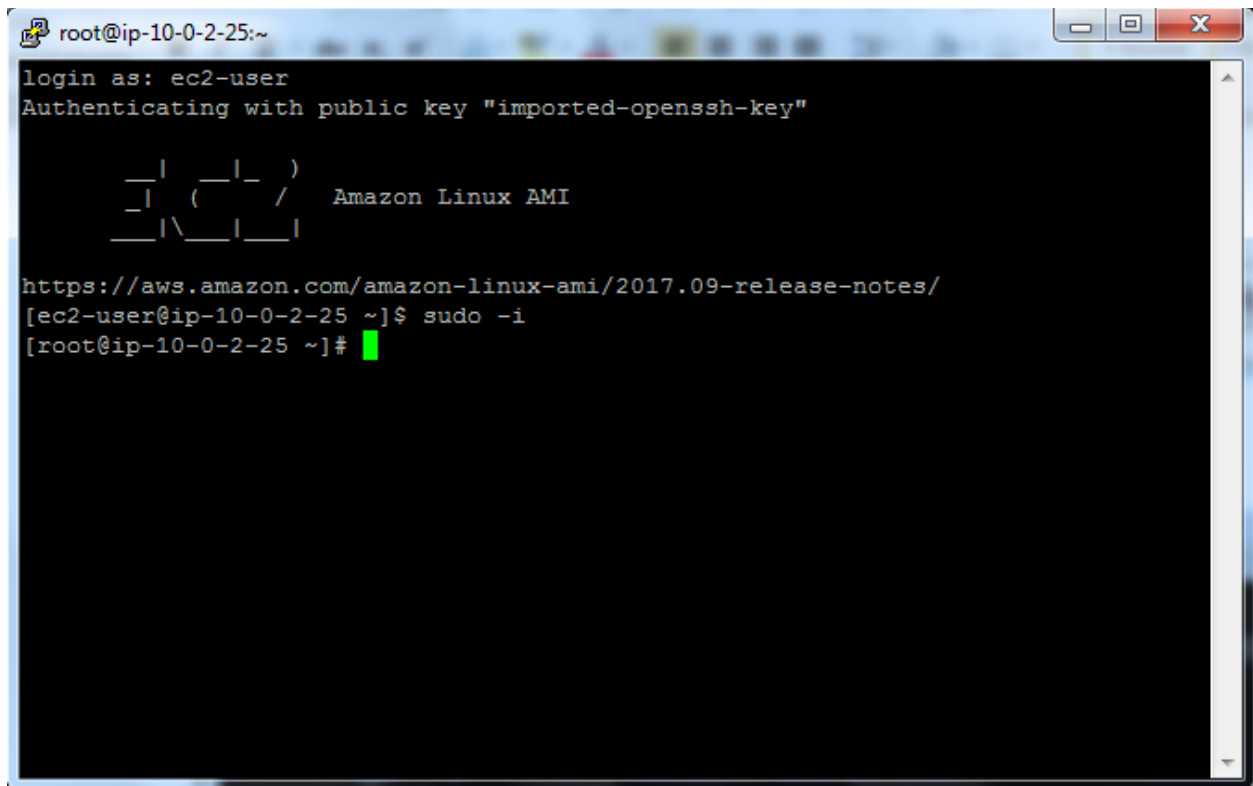


Click "Yes".



Type

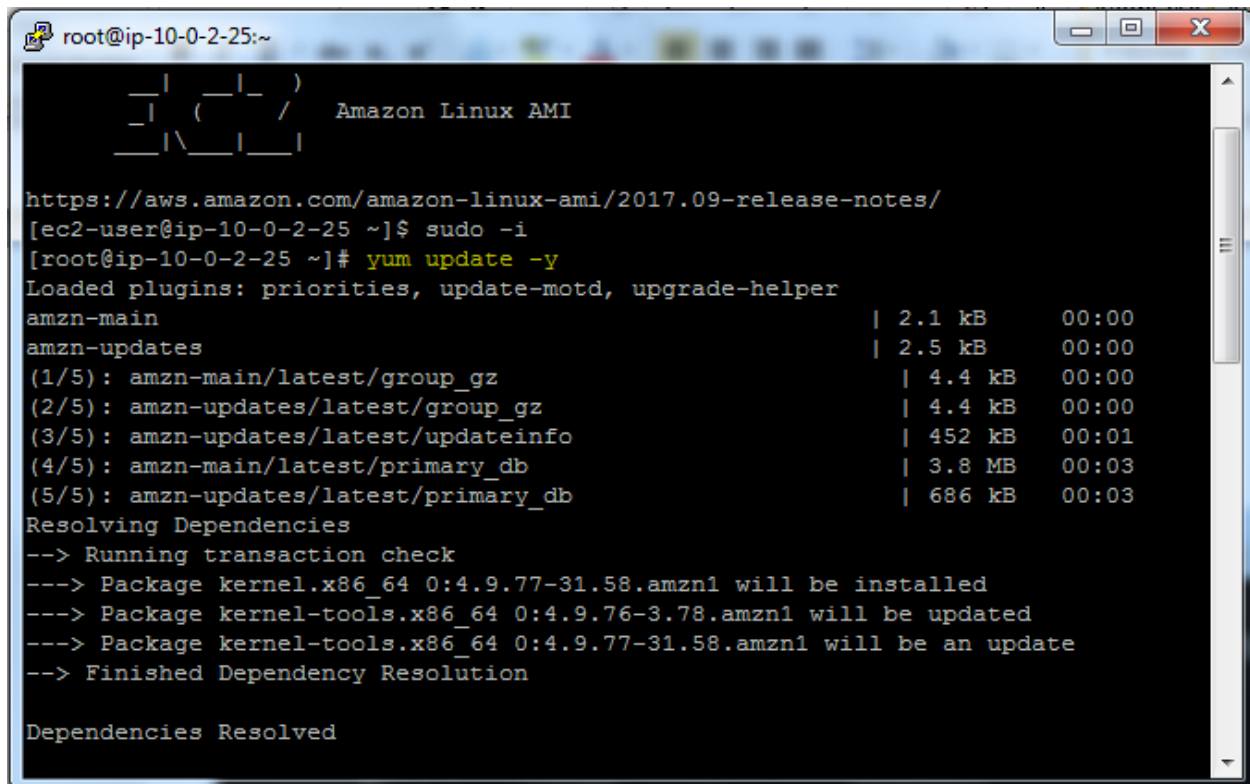
sudo-i



```
root@ip-10-0-2-25:~  
login as: ec2-user  
Authenticating with public key "imported-openssh-key"  
  
  _ | _ | _ )  
  _ | ( _ _ /   Amazon Linux AMI  
  __| \__|__|  
  
https://aws.amazon.com/amazon-linux-ami/2017.09-release-notes/  
[ec2-user@ip-10-0-2-25 ~]$ sudo -i  
[root@ip-10-0-2-25 ~]#
```

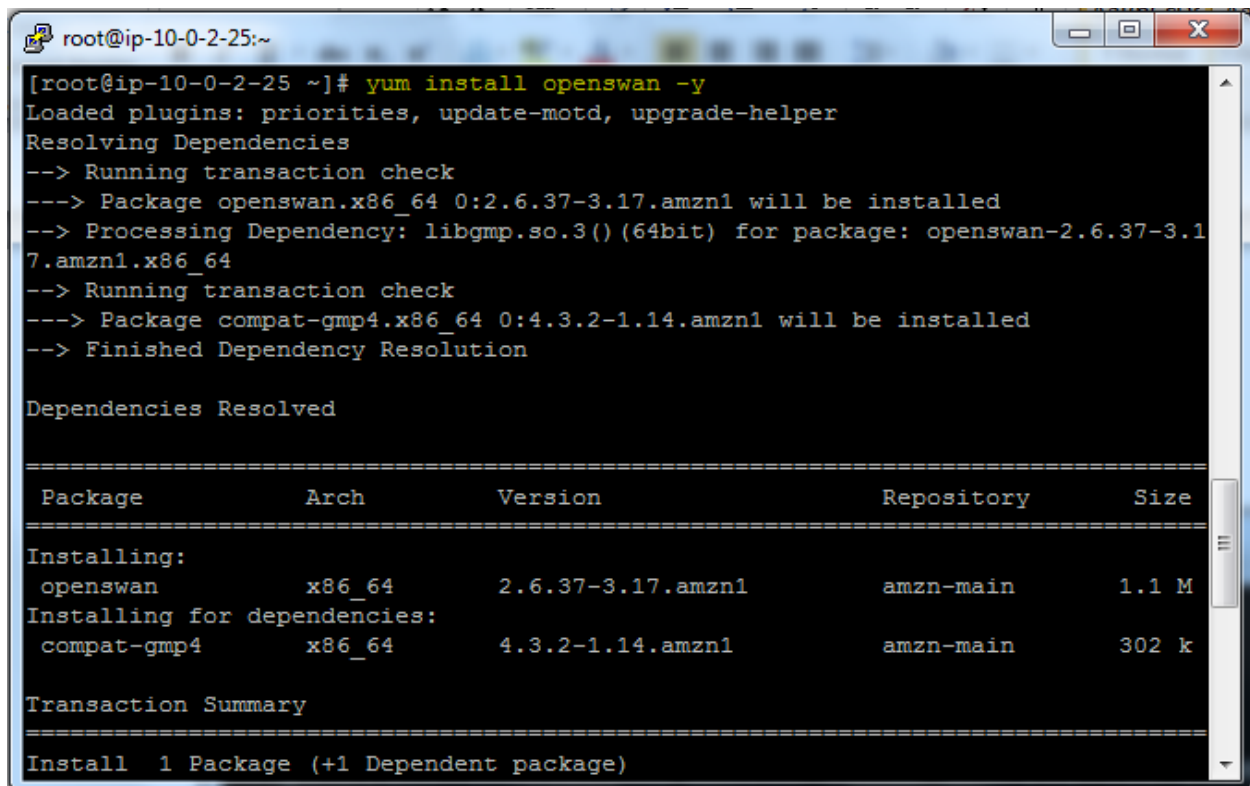
Type

Yum update -y



```
root@ip-10-0-2-25:~  
_ | _ | _ )  
_ | ( _ / Amazon Linux AMI  
_ | \ _ | _ |  
  
https://aws.amazon.com/amazon-linux-ami/2017.09-release-notes/  
[ec2-user@ip-10-0-2-25 ~]$ sudo -i  
[root@ip-10-0-2-25 ~]# yum update -y  
Loaded plugins: priorities, update-motd, upgrade-helper  
amzn-main | 2.1 kB 00:00  
amzn-updates | 2.5 kB 00:00  
(1/5): amzn-main/latest/group_gz | 4.4 kB 00:00  
(2/5): amzn-updates/latest/group_gz | 4.4 kB 00:00  
(3/5): amzn-updates/latest/updateinfo | 452 kB 00:01  
(4/5): amzn-main/latest/primary_db | 3.8 MB 00:03  
(5/5): amzn-updates/latest/primary_db | 686 kB 00:03  
Resolving Dependencies  
--> Running transaction check  
---> Package kernel.x86_64 0:4.9.77-31.58.amzn1 will be installed  
---> Package kernel-tools.x86_64 0:4.9.76-3.78.amzn1 will be updated  
---> Package kernel-tools.x86_64 0:4.9.77-31.58.amzn1 will be an update  
--> Finished Dependency Resolution  
  
Dependencies Resolved
```

Yum install openswan -y

A terminal window titled 'root@ip-10-0-2-25:~' showing the output of the command 'yum install openswan -y'. The output includes dependency resolution steps, a table of packages to be installed, and a transaction summary.

```
[root@ip-10-0-2-25 ~]# yum install openswan -y
Loaded plugins: priorities, update-motd, upgrade-helper
Resolving Dependencies
--> Running transaction check
--> Package openswan.x86_64 0:2.6.37-3.17.amzn1 will be installed
--> Processing Dependency: libgmp.so.3()(64bit) for package: openswan-2.6.37-3.17.amzn1.x86_64
--> Running transaction check
--> Package compat-gmp4.x86_64 0:4.3.2-1.14.amzn1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package                Arch          Version           Repository        Size
=====
Installing:
openswan                x86_64        2.6.37-3.17.amzn1  amzn-main        1.1 M
Installing for dependencies:
compat-gmp4             x86_64        4.3.2-1.14.amzn1  amzn-main        302 k

Transaction Summary
=====
Install 1 Package (+1 Dependent package)
```

Cd /etc

Vi ipsec.conf

In ipsec.conf file we need to **remove # from** #include /etc/ipsec.d/\*.conf line

```

root@p-10-0-2-25/etc
# /etc/ipsec.conf - Openswan IPsec configuration file
#
# Manual:      ipsec.conf.5
#
# Please place your own config files in /etc/ipsec.d/ ending in .conf

version 2.0      # conforms to second version of ipsec.conf specification

# basic configuration
config setup
    # Debug-logging controls: "none" for (almost) none, "all" for lots.
    # klipsdebug=none
    # plutodebug="control parsing"
    # For Red Hat Enterprise Linux and Fedora, leave protostack=netkey
    protostack=netkey
    nat_traversal=yes
    virtual_private=
    oe=off
    # Enable this if you see "failed to find any available worker"
    # nhelpers=0

#You may put your configuration (.conf) file in the "/etc/ipsec.d/" and uncomment this.
#include /etc/ipsec.d/*.conf

```

```

root@ip-10-0-2-25:/etc
# /etc/ipsec.conf - Openswan IPsec configuration file
#
# Manual:      ipsec.conf.5
#
# Please place your own config files in /etc/ipsec.d/ ending in .conf

version 2.0      # conforms to second version of ipsec.conf specification

# basic configuration
config setup
    # Debug-logging controls: "none" for (almost) none, "all" for lots.
    # klipsdebug=none
    # plutodebug="control parsing"
    # For Red Hat Enterprise Linux and Fedora, leave protostack=netkey
    protostack=netkey
    nat_traversal=yes
    virtual_private=
    oe=off
    # Enable this if you see "failed to find any available worker"
    # nhelpers=0

#You may put your configuration (.conf) file in the "/etc/ipsec.d/" and uncomment this.
#include /etc/ipsec.d/*.conf

-- INSERT --

```

Press Escape key



Type :wq

```

root@p-10-0-2-25/etc
# /etc/ipsec.conf - Openswan IPsec configuration file
#
# Manual:      ipsec.conf.5
#
# Please place your own config files in /etc/ipsec.d/ ending in .conf

version 2.0      # conforms to second version of ipsec.conf specification

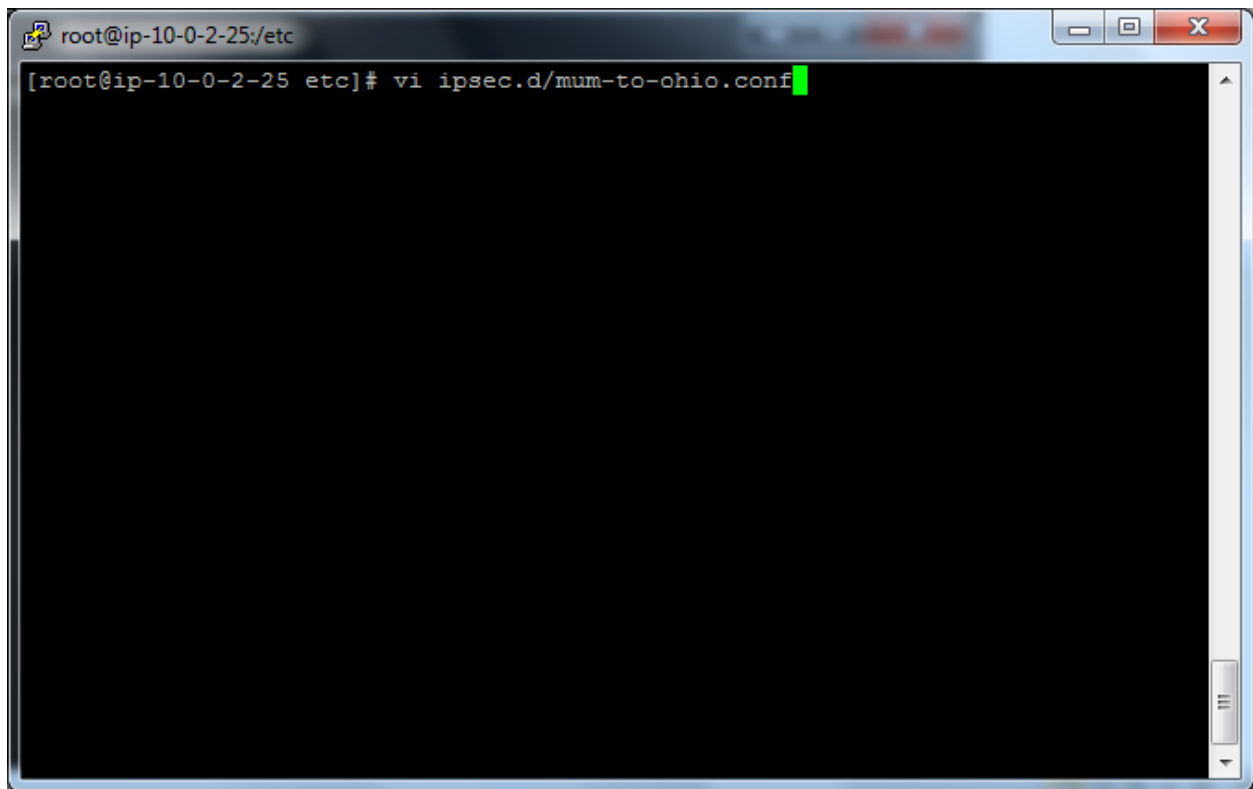
# basic configuration
config setup
    # Debug-logging controls: "none" for (almost) none, "all" for lots.
    # klipsdebug=none
    # plutodebug="control parsing"
    # For Red Hat Enterprise Linux and Fedora, leave protostack=netkey
    protostack=netkey
    nat_traversal=yes
    virtual_private=
    oe=off
    # Enable this if you see "failed to find any available worker"
    # nhelpers=0

# You may put your configuration (.conf) file in the "/etc/ipsec.d/" and uncomment this.
include /etc/ipsec.d/*.conf

```

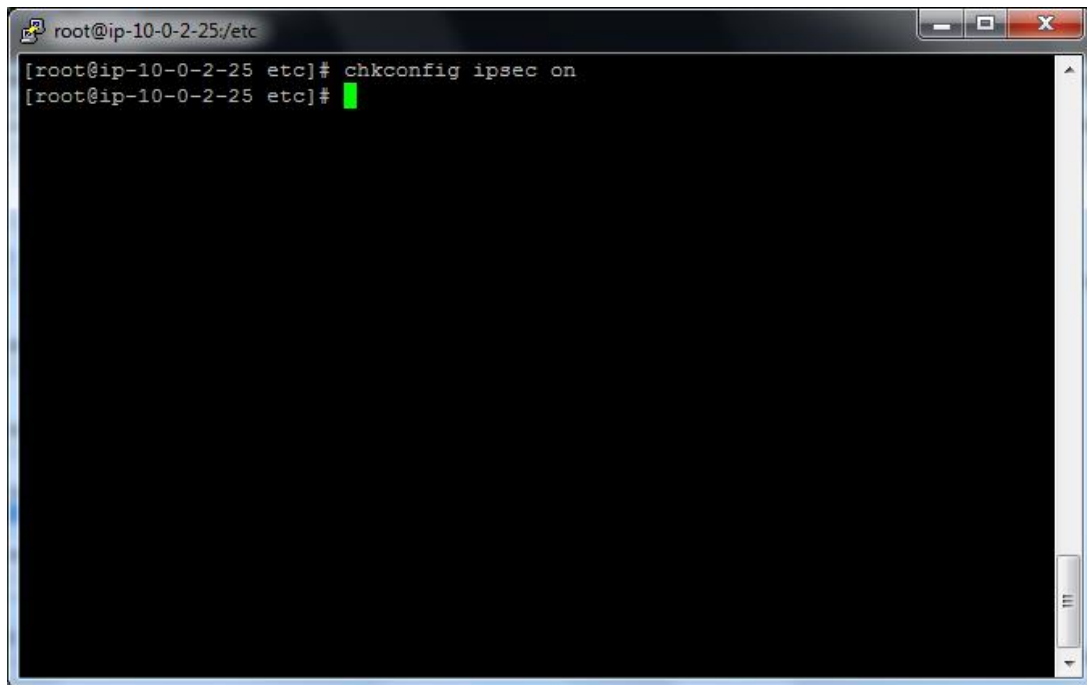
Type

Vi ipsec.d/mum-to-ohio.conf



Type

Chkconfig ipsec on



Copy the command to below editor.

conn mum-to-ohio

type=tunnel

authby=secret

left=defaultroute

```
leftid=13.127.161.231
```

```
leftnexthop=%defaultroute
```

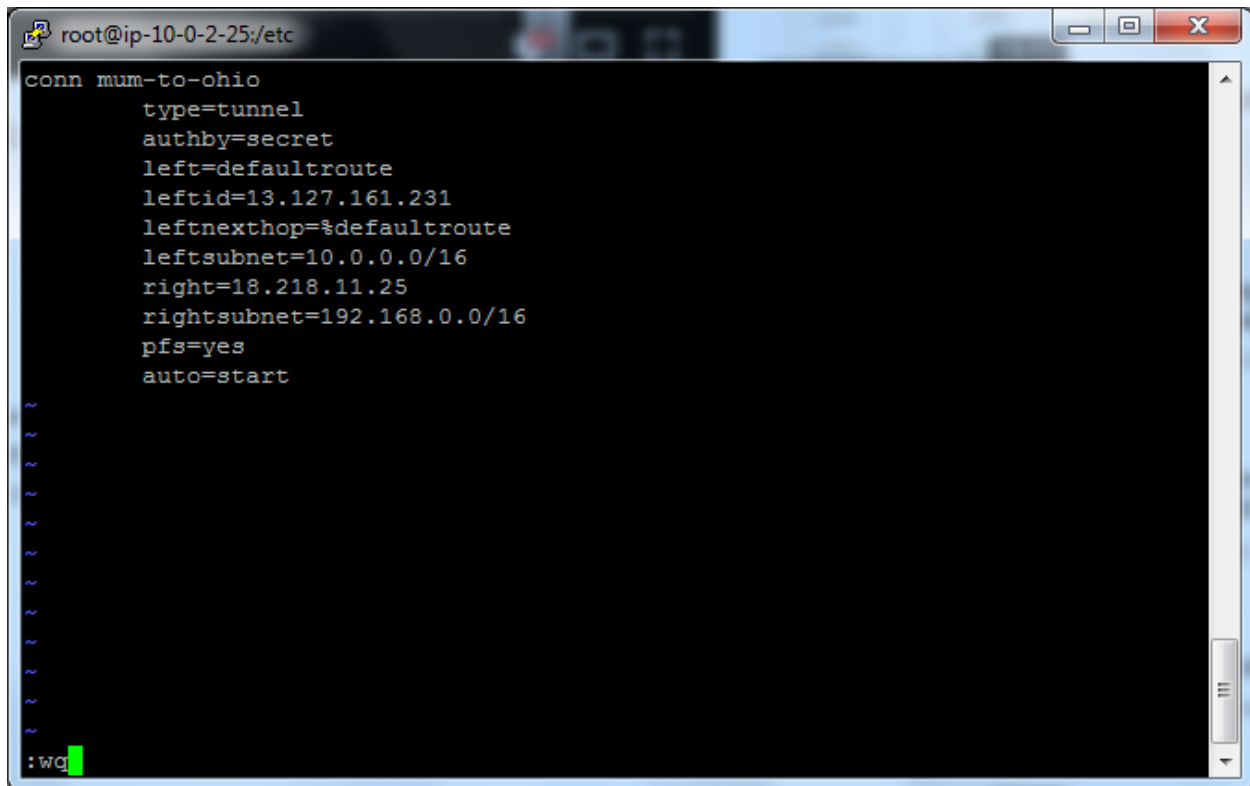
```
leftsubnet=10.0.0.0/16
```

```
right=18.218.11.25
```

```
rightsubnet=192.168.0.0/16
```

```
pfs=yes
```

```
auto=start
```

A screenshot of a terminal window titled 'root@ip-10-0-2-25:/etc'. The terminal shows the configuration of an IPsec tunnel named 'mum-to-ohio'. The configuration is as follows:

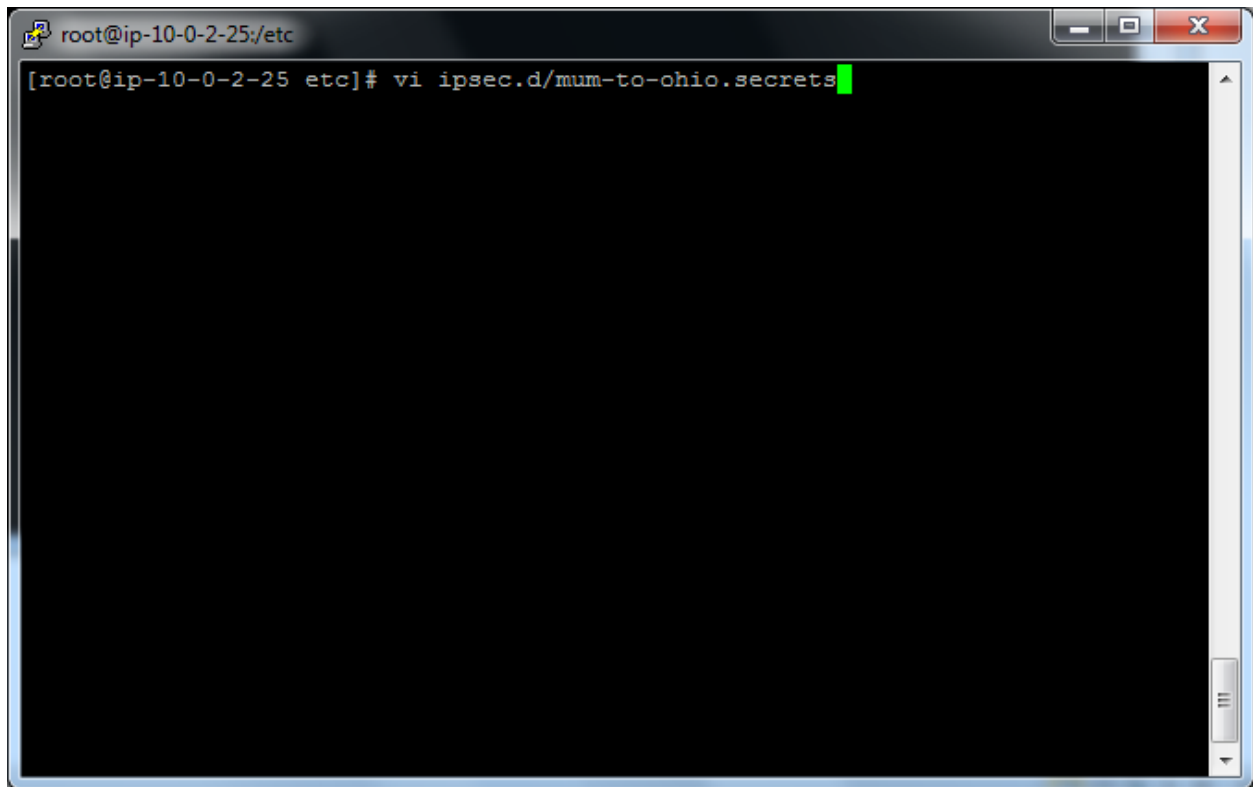
```
conn mum-to-ohio
    type=tunnel
    authby=secret
    left=defaultroute
    leftid=13.127.161.231
    leftnexthop=%defaultroute
    leftsubnet=10.0.0.0/16
    right=18.218.11.25
    rightsubnet=192.168.0.0/16
    pfs=yes
    auto=start
```

Below the configuration, there are several tilde (~) characters indicating the end of the file. At the bottom, the prompt ':wq' is visible, indicating the user is in vi editor mode and has pressed the escape key followed by 'wq' to save and quit.

Press escape and type :wq

Type

Vi ipsec.d/mum-to-ohio.secrets



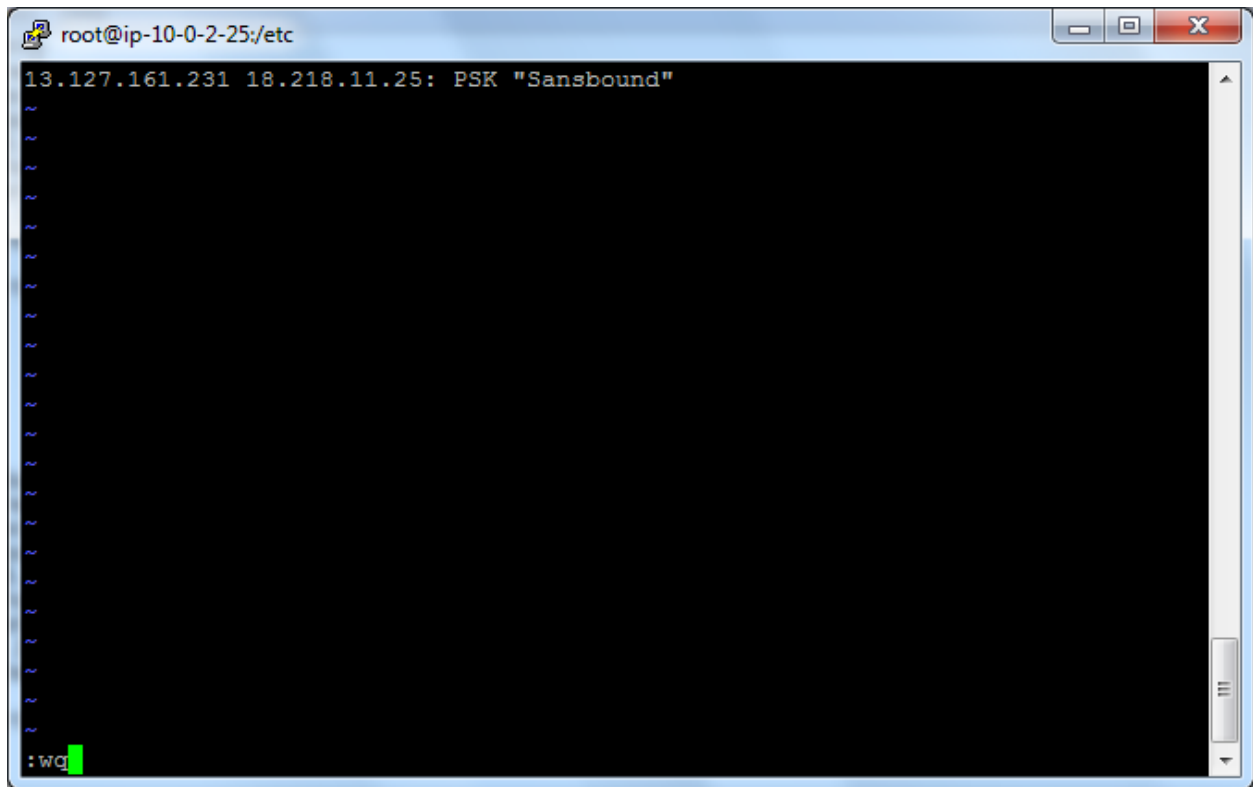
TYPE EIP1 (Mumbai EIP) and type EIP2 (Ohio EIP) then type : PSK "Preshared key of the tunnel".

Our Tunnel Preshared key is "Sansbound"

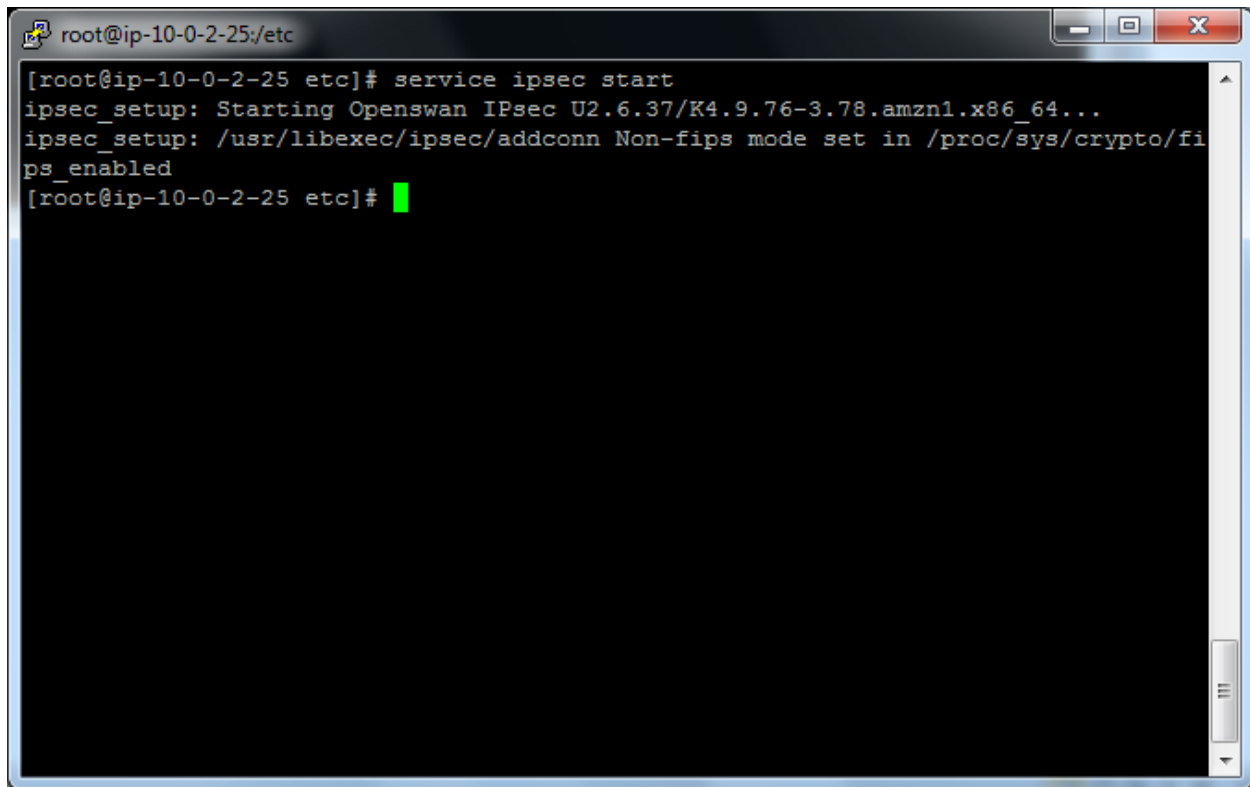
A screenshot of a terminal window titled 'root@ip-10-0-2-25:/etc'. The terminal shows the command '13.127.161.231 18.218.11.25: PSK "Sansbound"' being entered. Below the command, there are approximately 15 tilde (~) characters, likely representing a large amount of output or a full buffer. At the bottom left of the terminal, the text '-- INSERT --' is visible. The terminal has a black background with white text. A green cursor is positioned at the end of the first line of input. The window's title bar includes standard Linux window controls (minimize, maximize, close).

Press escape key

Type :wq



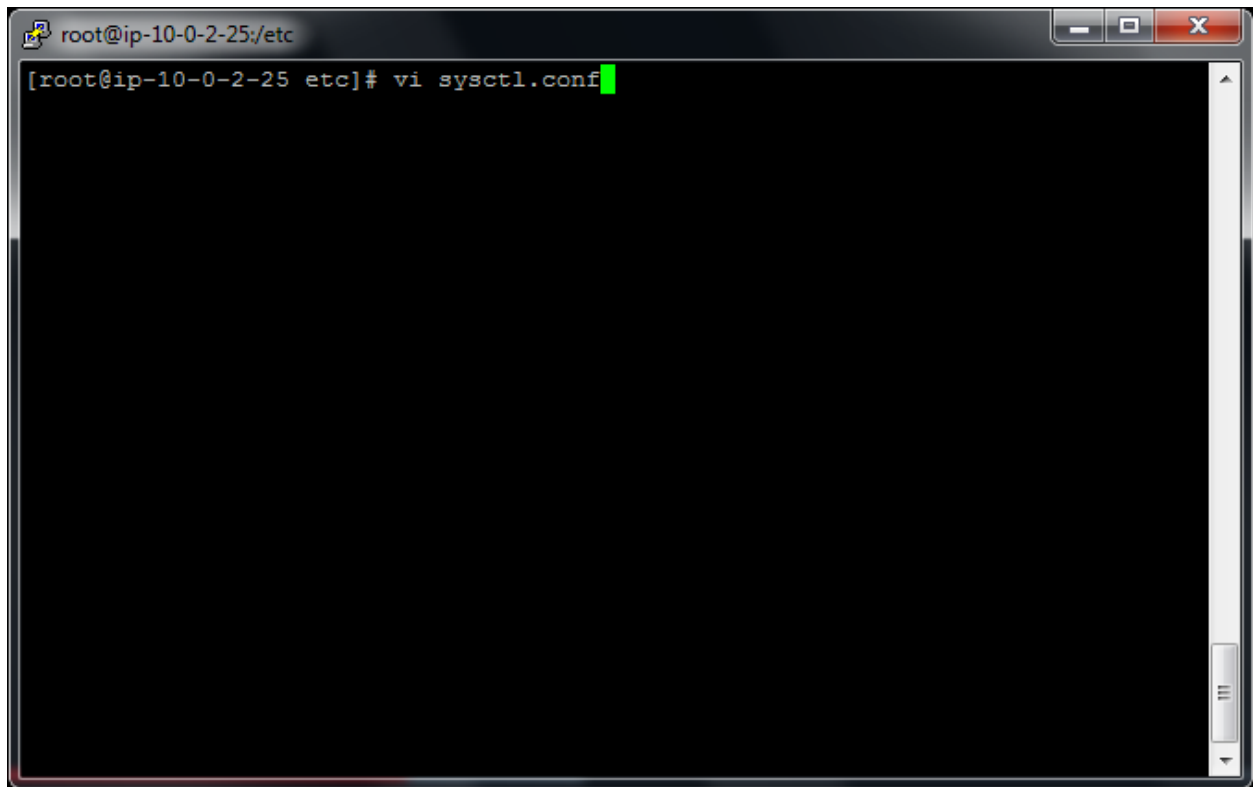
Service ipsec start

A terminal window with a dark background and a light blue border. The title bar shows 'root@ip-10-0-2-25/etc' and standard window controls. The terminal text shows the command 'service ipsec start' being executed, followed by status messages from 'ipsec\_setup' and a green cursor at the end of the prompt.

```
root@ip-10-0-2-25/etc# service ipsec start
ipsec_setup: Starting Openswan IPsec U2.6.37/K4.9.76-3.78.amzn1.x86_64...
ipsec_setup: /usr/libexec/ipsec/addconn Non-fips mode set in /proc/sys/crypto/fips_enabled
root@ip-10-0-2-25/etc#
```

Type sysctl.conf



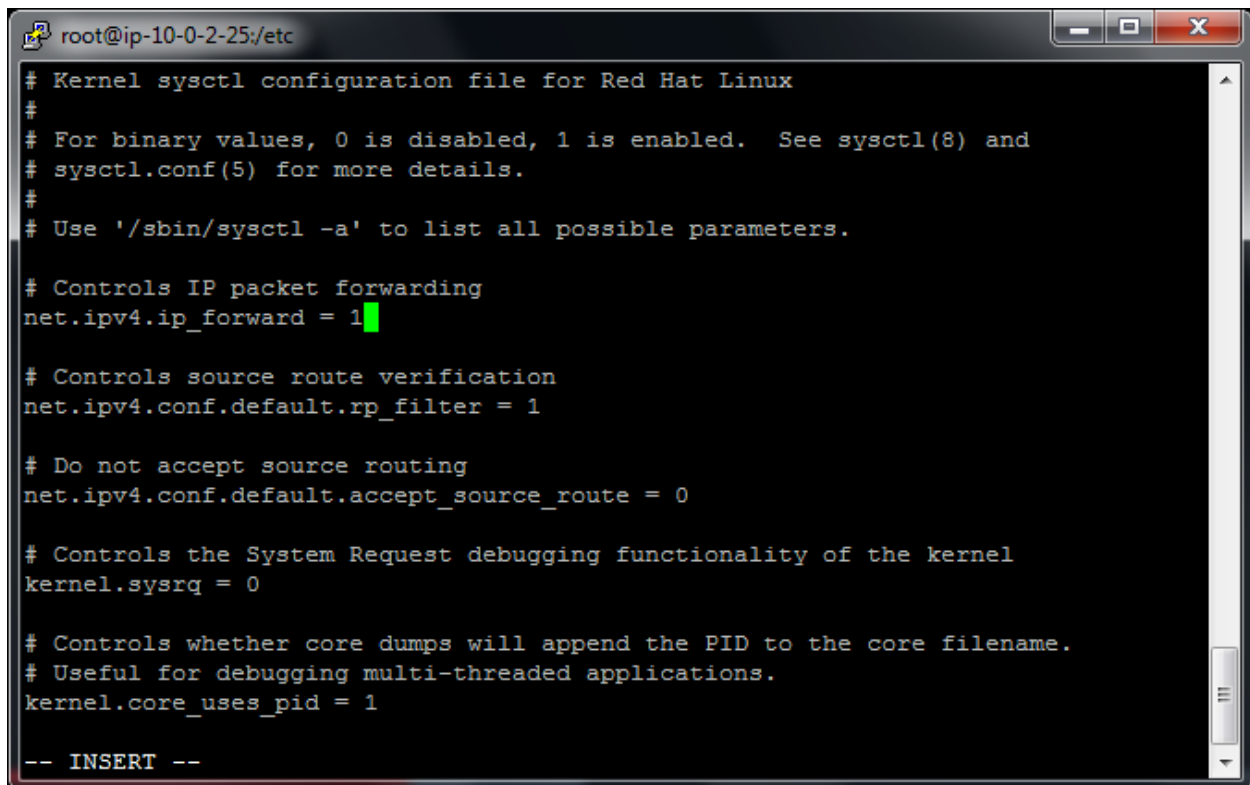
A terminal window with a dark background. The title bar shows 'root@ip-10-0-2-25:/etc' and standard window controls. The command prompt is '[root@ip-10-0-2-25 etc]#'. The command 'vi sysctl.conf' has been entered, and a green cursor is at the end of the line.

```
root@ip-10-0-2-25:/etc
[root@ip-10-0-2-25 etc]# vi sysctl.conf
```

Press insert and then change the value as below.

Change

```
net.ipv4.ip_forward = 1
```

A screenshot of a terminal window titled 'root@ip-10-0-2-25:/etc'. The terminal displays the contents of the /etc/sysctl.conf file, which is a kernel sysctl configuration file for Red Hat Linux. The file contains several commented-out lines and one active line: 'net.ipv4.ip\_forward = 1'. The terminal window has a standard Linux desktop environment window with minimize, maximize, and close buttons in the title bar. The text is displayed in a monospaced font on a dark background.

```
# Kernel sysctl configuration file for Red Hat Linux
#
# For binary values, 0 is disabled, 1 is enabled.  See sysctl(8) and
# sysctl.conf(5) for more details.
#
# Use '/sbin/sysctl -a' to list all possible parameters.

# Controls IP packet forwarding
net.ipv4.ip_forward = 1

# Controls source route verification
net.ipv4.conf.default.rp_filter = 1

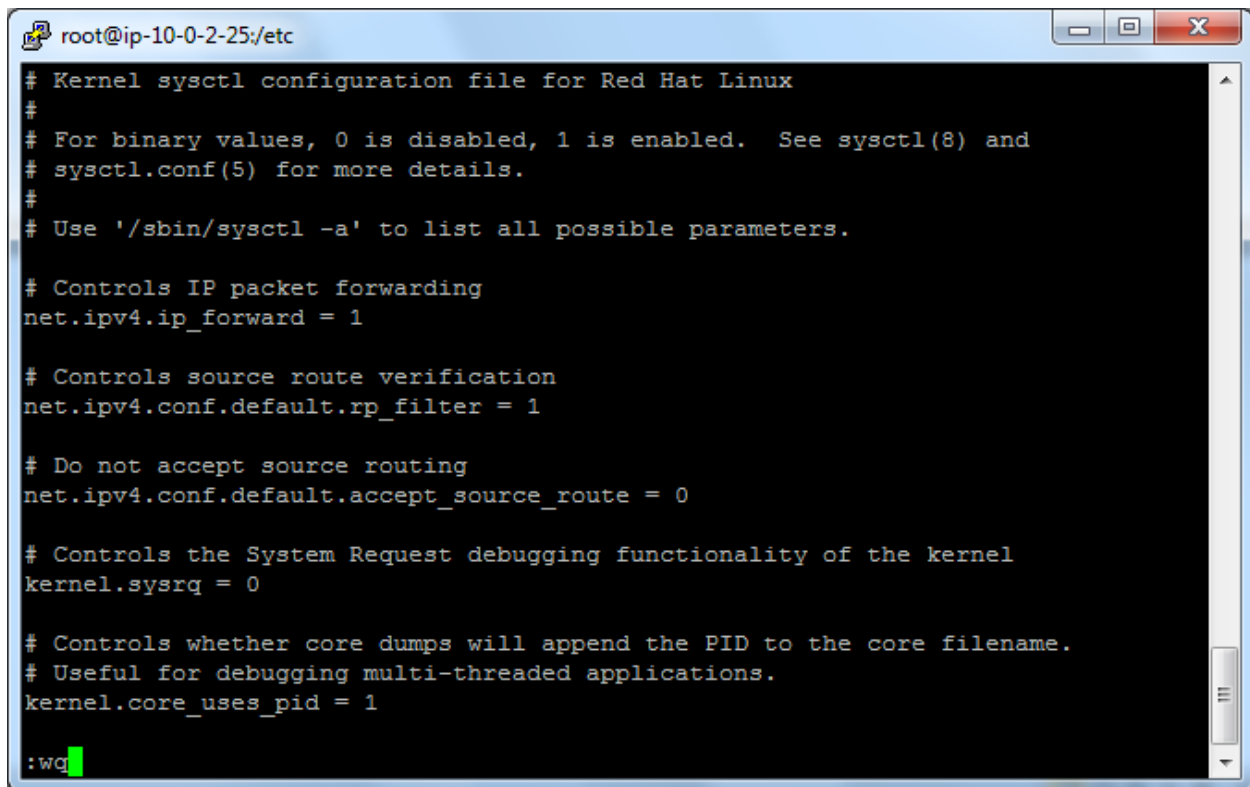
# Do not accept source routing
net.ipv4.conf.default.accept_source_route = 0

# Controls the System Request debugging functionality of the kernel
kernel.sysrq = 0

# Controls whether core dumps will append the PID to the core filename.
# Useful for debugging multi-threaded applications.
kernel.core_uses_pid = 1

-- INSERT --
```

Press “Escape” key

A screenshot of a terminal window titled 'root@ip-10-0-2-25:/etc'. The terminal displays the contents of the /etc/sysctl.conf file, which is a kernel sysctl configuration file for Red Hat Linux. The file contains several commented lines and two active configuration lines: 'net.ipv4.ip\_forward = 1' and 'kernel.core\_uses\_pid = 1'. The terminal prompt is ':wq' followed by a green cursor.

```
root@ip-10-0-2-25:/etc
# Kernel sysctl configuration file for Red Hat Linux
#
# For binary values, 0 is disabled, 1 is enabled.  See sysctl(8) and
# sysctl.conf(5) for more details.
#
# Use '/sbin/sysctl -a' to list all possible parameters.

# Controls IP packet forwarding
net.ipv4.ip_forward = 1

# Controls source route verification
net.ipv4.conf.default.rp_filter = 1

# Do not accept source routing
net.ipv4.conf.default.accept_source_route = 0

# Controls the System Request debugging functionality of the kernel
kernel.sysrq = 0

# Controls whether core dumps will append the PID to the core filename.
# Useful for debugging multi-threaded applications.
kernel.core_uses_pid = 1

:wq
```

and then type

:wq

Go to Ec2 Dashboard

Click “Network interface” and then select “OpenSwan”

Click “Actions” → Click “Change source/destination check”

The screenshot shows the AWS Management Console interface. On the left is a navigation menu with categories like INSTANCES, IMAGES, ELASTIC BLOCK STORE, NETWORK & SECURITY, and NETWORK INTERFACES. The 'Network Interfaces' section is selected. The main area displays a table of network interfaces. The first interface, 'eni-2a423a01', is selected. An 'Actions' dropdown menu is open, showing options like 'Attach', 'Detach', 'Delete', 'Manage IP Addresses', 'Associate Address', 'Disassociate Address', 'Change Termination Behavior', 'Change Security Groups', 'Change Source/Dest. Check' (highlighted in orange), 'Add/Edit Tags', 'Change Description', and 'Create Flow Log'. Below the table, the 'Details' tab for 'eni-2a423a01' is expanded, showing various attributes.

Name	Network interf.	Subnet ID	Security groups	Description	Instance ID
OpenSwan	eni-2a423a01	subnet-07d1c44a	Mumbai_Linux_Sec...	Primary netwo...	i-09ecc561305fd3f...
	eni-79275e52	subnet-07d1c44a	Win Pub Sec Group	Primary netwo...	i-0f64d7067af702f...
	eni-b7cffe8	subnet-bb7ed7...	Win Pvt Sec Group	Primary netwo...	i-0f7c9db614ae7e...

Attribute	Value
Network interface ID	eni-2a423a01
VPC ID	vpc-09fe2261
MAC address	0a:9e:69:49:ec:82
Security groups	Mumbai_Linux_Sec_Group . view inbound rules
Status	in-use
Private DNS (IPv4)	-
Subnet ID	subnet-07d1c44a
Availability Zone	ap-south-1b
Description	Primary network interface
Owner ID	297111308396
Primary private IPv4 IP	10.0.2.25
IPv4 Public IP	13.127.161.231*

Set it as “Disabled” and click “save”.

**Change Source/Dest. Check** ✕

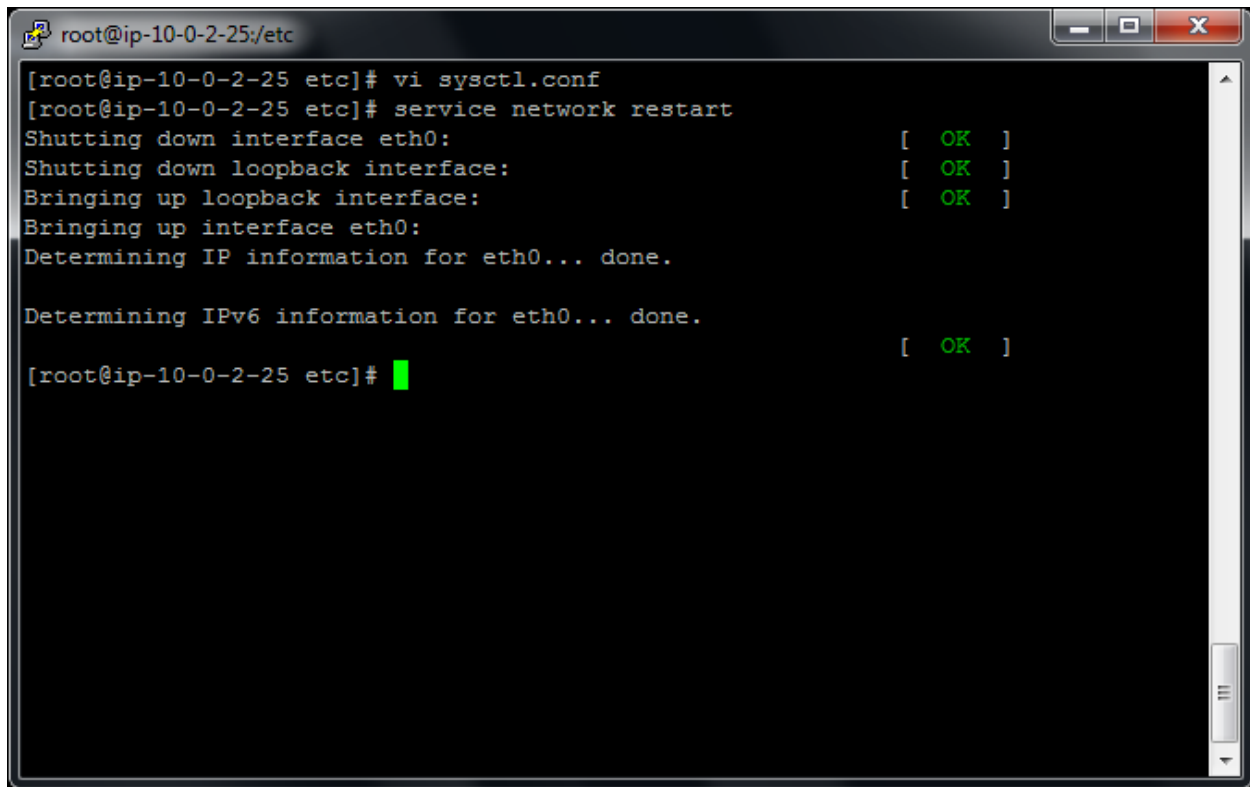
**Network Interface** eni-2a423a01

**Source/dest. check** ☐ Enabled  
☒ Disabled

Cancel Save

Type

Service network restart

A terminal window titled 'root@ip-10-0-2-25:/etc' with standard window controls. The terminal displays the execution of 'service network restart'. It shows the process of shutting down and bringing up the 'eth0' interface and the loopback interface, with green 'OK' status indicators for each step. The process concludes with determining IP and IPv6 information for 'eth0', also marked as 'done.' and 'OK'. The prompt returns to the root user.

```
root@ip-10-0-2-25:/etc# vi sysctl.conf
root@ip-10-0-2-25:/etc# service network restart
Shutting down interface eth0:           [ OK ]
Shutting down loopback interface:       [ OK ]
Bringing up loopback interface:         [ OK ]
Bringing up interface eth0:
Determining IP information for eth0... done.

Determining IPv6 information for eth0... done.           [ OK ]
root@ip-10-0-2-25:/etc#
```

Type `vi sysctl.conf`

A terminal window with a dark background and a light gray border. The title bar at the top shows a yellow icon, the text 'root@ip-10-0-2-25:/etc', and standard window controls (minimize, maximize, close). The terminal content shows the prompt '[root@ip-10-0-2-25 etc]# ' followed by the command 'vi sysctl.conf' and a green cursor at the end of the line. The rest of the terminal area is empty.

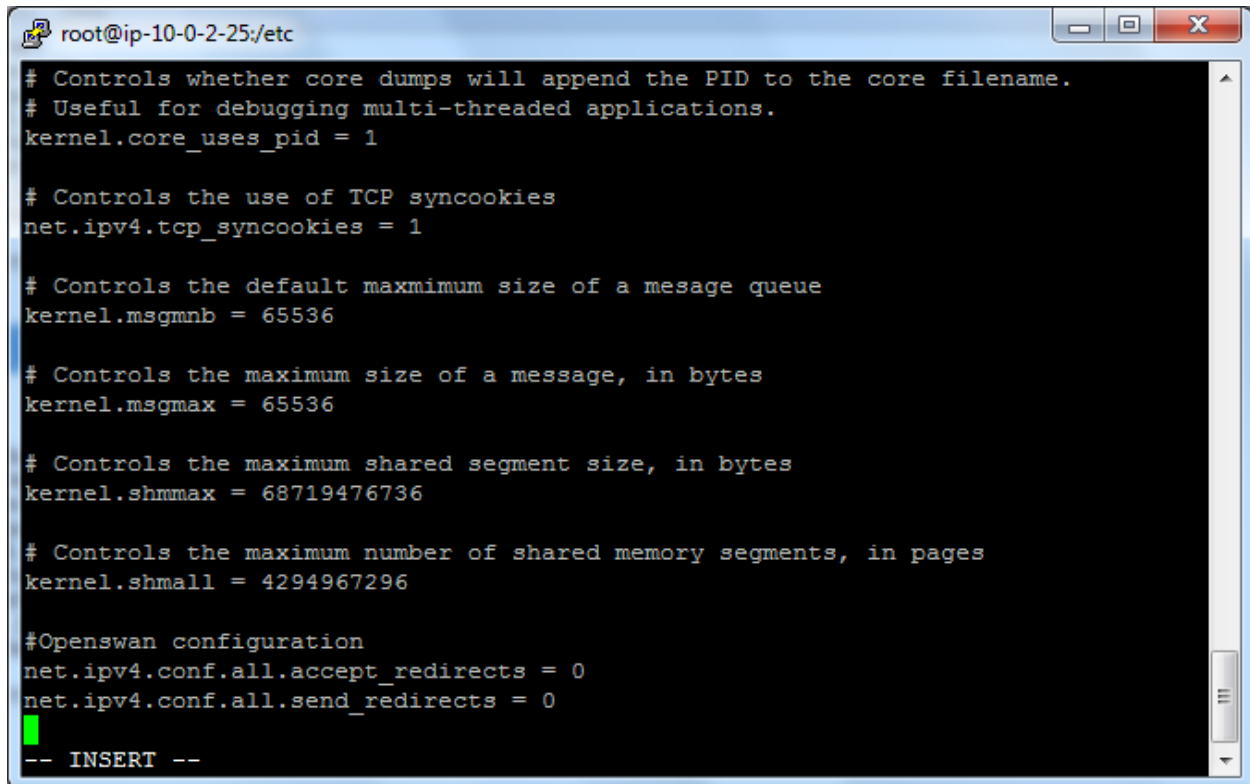
```
root@ip-10-0-2-25:/etc
[root@ip-10-0-2-25 etc]# vi sysctl.conf
```

Press insert key

Type

```
net.ipv4.conf.all.accept_redirects = 0
```

```
net.ipv4.conf.all.send_redirects = 0
```



```
root@ip-10-0-2-25:/etc
# Controls whether core dumps will append the PID to the core filename.
# Useful for debugging multi-threaded applications.
kernel.core_uses_pid = 1

# Controls the use of TCP syncookies
net.ipv4.tcp_syncookies = 1

# Controls the default maximum size of a message queue
kernel.msgmnb = 65536

# Controls the maximum size of a message, in bytes
kernel.msgmax = 65536

# Controls the maximum shared segment size, in bytes
kernel.shmmax = 68719476736

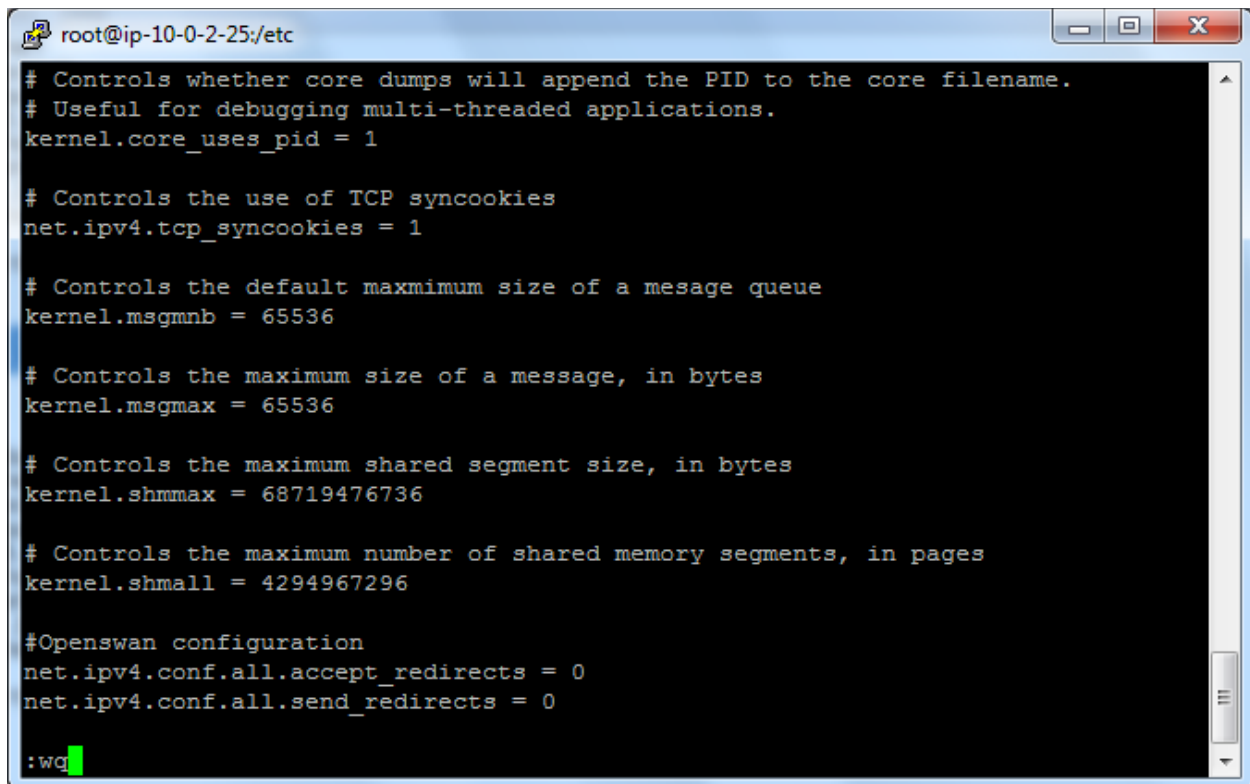
# Controls the maximum number of shared memory segments, in pages
kernel.shmall = 4294967296

#Openswan configuration
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.all.send_redirects = 0
-- INSERT --
```

Press escape key and type



:wq

A screenshot of a terminal window with a blue title bar. The title bar contains a small icon, the text 'root@ip-10-0-2-25:/etc', and standard window control buttons (minimize, maximize, close). The terminal has a black background with white text. It displays several kernel configuration parameters, each preceded by a comment line starting with '#'. The parameters are: 'kernel.core\_uses\_pid = 1', 'net.ipv4.tcp\_syncookies = 1', 'kernel.msgmnb = 65536', 'kernel.msgmax = 65536', 'kernel.shmmax = 68719476736', and 'kernel.shmall = 4294967296'. Below these, there is a section for 'Openswan configuration' with 'net.ipv4.conf.all.accept\_redirects = 0' and 'net.ipv4.conf.all.send\_redirects = 0'. At the bottom of the terminal, the prompt ':wq' is followed by a green cursor. A vertical scrollbar is visible on the right side of the terminal window.

```
root@ip-10-0-2-25:/etc
# Controls whether core dumps will append the PID to the core filename.
# Useful for debugging multi-threaded applications.
kernel.core_uses_pid = 1

# Controls the use of TCP syncookies
net.ipv4.tcp_syncookies = 1

# Controls the default maximum size of a message queue
kernel.msgmnb = 65536

# Controls the maximum size of a message, in bytes
kernel.msgmax = 65536

# Controls the maximum shared segment size, in bytes
kernel.shmmax = 68719476736

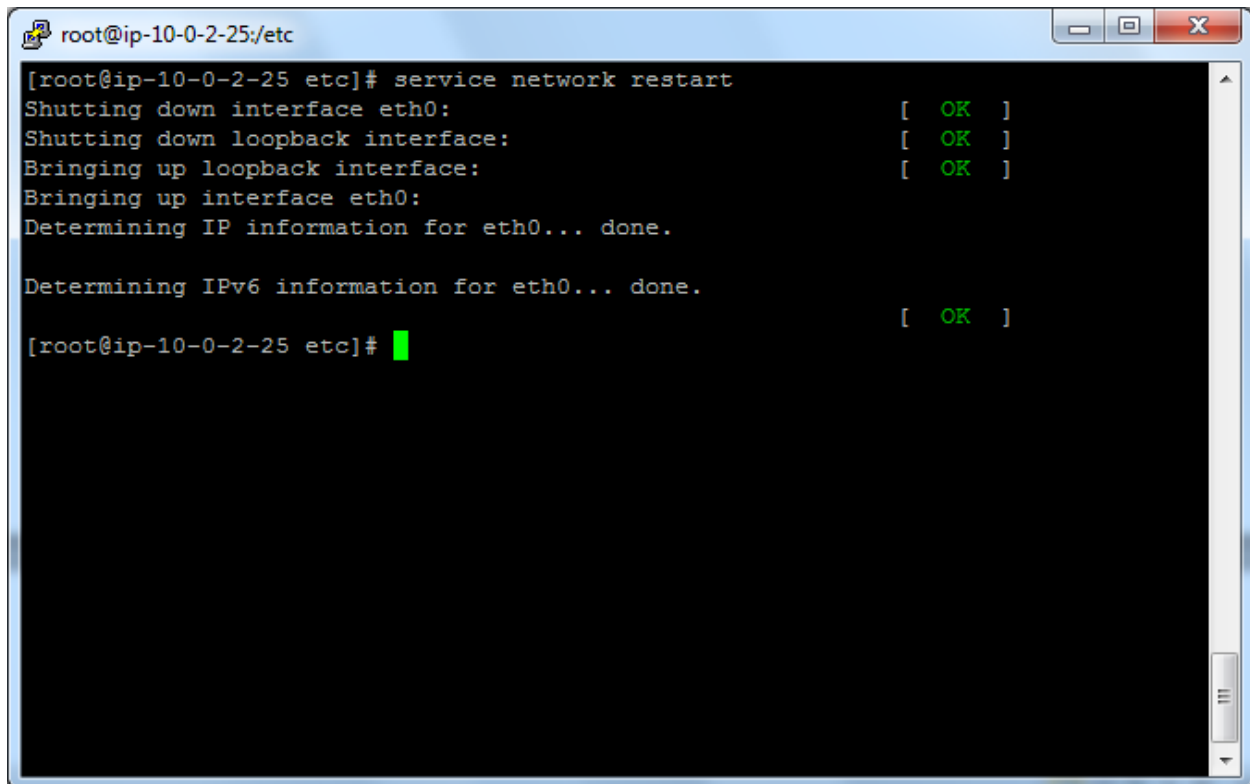
# Controls the maximum number of shared memory segments, in pages
kernel.shmall = 4294967296

#Openswan configuration
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.all.send_redirects = 0

:wq
```

Type

Service network restart

A terminal window titled 'root@ip-10-0-2-25/etc' with standard window controls. The terminal output shows the command 'service network restart' being executed. The output includes status messages for shutting down and bringing up interfaces eth0 and loopback, with green 'OK' status indicators in brackets. The prompt returns to the root user at the same directory.

```
root@ip-10-0-2-25/etc# service network restart
Shutting down interface eth0:          [ OK ]
Shutting down loopback interface:      [ OK ]
Bringing up loopback interface:        [ OK ]
Bringing up interface eth0:
Determining IP information for eth0... done.

Determining IPv6 information for eth0... done.          [ OK ]
root@ip-10-0-2-25/etc#
```

Go to VPC dashboard,

Click Route table, select sansbound public route table,

The screenshot shows the AWS Management Console interface. The left sidebar contains the navigation menu with categories like VPC Dashboard, Virtual Private Cloud, Security, and VPN Connections. The 'Route Tables' link is highlighted. The main content area shows the 'Route Tables' page with buttons for 'Create Route Table', 'Delete Route Table', and 'Set As Main Table'. A table lists four route tables. Below the table, the details for 'rtb-7d6de015 | Sansbound\_public\_route' are shown, including tabs for Summary, Routes, Subnet Associations, Route Propagation, and Tags. The 'Summary' tab is active, displaying the route table ID, name, explicitly associated subnets, main status, and VPC.

Name	Route Table ID	Explicitly Associat	Main	VPC
<input checked="" type="checkbox"/> Sansbound_public_route	rtb-7d6de015	1 Subnet	Yes	vpc-09fe2261   Sansbound_VPC_Mumbai
<input type="checkbox"/> sansbound Private route	rtb-b642dfde	1 Subnet	No	vpc-09fe2261   Sansbound_VPC_Mumbai
<input type="checkbox"/>	rtb-91b209f9	0 Subnets	No	vpc-a655a2ce
<input type="checkbox"/>	rtb-3dab7555	0 Subnets	Yes	vpc-a655a2ce

**rtb-7d6de015 | Sansbound\_public\_route**

**Summary** | Routes | Subnet Associations | Route Propagation | Tags

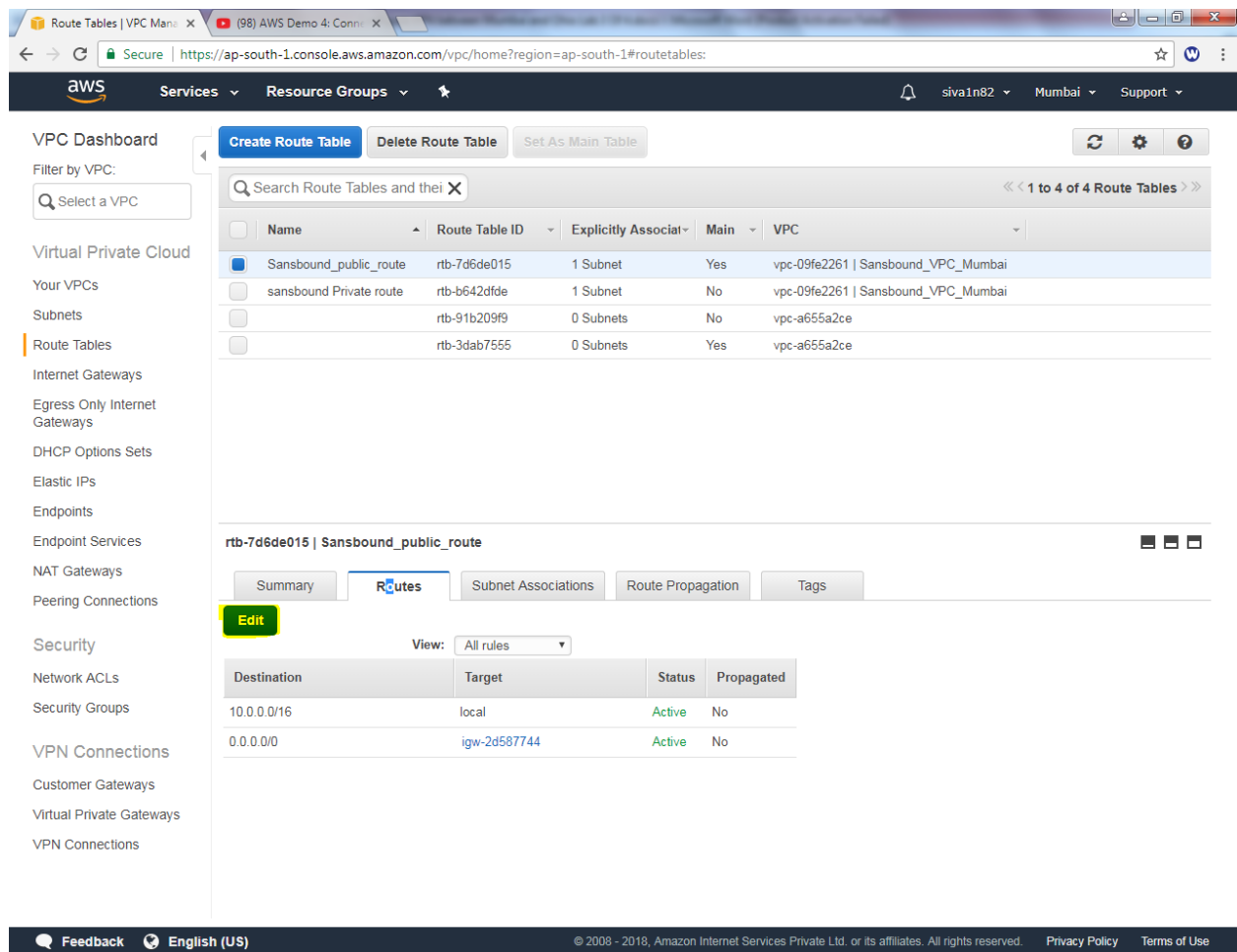
Route Table ID: rtb-7d6de015 | Sansbound\_public\_route

Main: yes

Explicitly Associated With: 1 Subnet

VPC: vpc-09fe2261 | Sansbound\_VPC\_Mumbai

Click "Edit"



The screenshot shows the AWS Management Console interface for Route Tables. The left sidebar contains a navigation menu with categories like VPC Dashboard, Virtual Private Cloud, Security, and VPN Connections. The main content area displays a list of route tables under the heading "Route Tables". The table has columns for Name, Route Table ID, Explicitly Associated, Main, and VPC. The "Sansbound\_public\_route" is selected, and its details are shown below. The "Routes" tab is active, displaying a table of routes with columns for Destination, Target, Status, and Propagated.

Name	Route Table ID	Explicitly Associated	Main	VPC
Sansbound_public_route	rtb-7d6de015	1 Subnet	Yes	vpc-09fe2261   Sansbound_VPC_Mumbai
sansbound Private route	rtb-b642dfde	1 Subnet	No	vpc-09fe2261   Sansbound_VPC_Mumbai
	rtb-91b209f9	0 Subnets	No	vpc-a655a2ce
	rtb-3dab7555	0 Subnets	Yes	vpc-a655a2ce

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No
0.0.0.0/0	igw-2d587744	Active	No

Click “add another route” and then type 192.168.0.0/16 as destination and select “Linux VPN Server” as target.

The screenshot shows the AWS Management Console interface. On the left is the VPC Dashboard sidebar with various navigation links. The main content area displays a list of Route Tables. Below this, the details for the 'rtb-7d6de015 | Sansbound\_public\_route' are shown, including a table of routes.

**VPC Dashboard**

Filter by VPC:

**Virtual Private Cloud**

- Your VPCs
- Subnets
- Route Tables
- Internet Gateways
- Egress Only Internet Gateways
- DHCP Options Sets
- Elastic IPs
- Endpoints
- Endpoint Services
- NAT Gateways
- Peering Connections

**Security**

- Network ACLs
- Security Groups

**VPN Connections**

- Customer Gateways
- Virtual Private Gateways
- VPN Connections

**Route Tables**

Search Route Tables and their VPCs:

<< 1 to 4 of 4 Route Tables >>

Name	Route Table ID	Explicitly Associated	Main	VPC
Sansbound_public_route	rtb-7d6de015	1 Subnet	Yes	vpc-09fe2261   Sansbound_VPC_Mumbai
sansbound Private route	rtb-b642dfde	1 Subnet	No	vpc-09fe2261   Sansbound_VPC_Mumbai
	rtb-91b209f9	0 Subnets	No	vpc-a655a2ce
	rtb-3dab7555	0 Subnets	Yes	vpc-a655a2ce

**rtb-7d6de015 | Sansbound\_public\_route**

Summary Routes Subnet Associations Route Propagation Tags

Cancel Save

View: All rules

Destination	Target	Status	Propagated	Remove
10.0.0.0/16	local	Active	No	
0.0.0.0/0	igw-2d587744	Active	No	
192.168.0.0/16		No	No	

Add another route

igw-2d587744 | Sansbound\_mumbai\_IGW  
i-09ecc561305fd387a | Linux VPN Serve...

Click "save".

The screenshot displays the AWS Management Console interface. On the left, the 'VPC Dashboard' sidebar is visible, listing various VPC resources. The main content area shows a list of route tables under the heading '1 to 4 of 4 Route Tables'. The selected route table, 'Sansbound\_public\_route' (ID: rtb-7d6de015), is shown in detail. The 'Routes' tab is active, displaying a table of route rules. The table has columns for Destination, Target, Status, Propagated, and Remove. Three routes are listed: a local route for 10.0.0.0/16, a route to an Internet Gateway (igw-2d587744) for 0.0.0.0/0, and a route to a VPC peering connection (pcx-09ecc561305fd387a) for 192.168.0.0/16. The 'Add another route' button is visible at the bottom of the route list.

**VPC Dashboard**

Filter by VPC:

**Virtual Private Cloud**

- Your VPCs
- Subnets
- Route Tables
- Internet Gateways
- Egress Only Internet Gateways
- DHCP Options Sets
- Elastic IPs
- Endpoints
- Endpoint Services
- NAT Gateways
- Peering Connections

**Security**

- Network ACLs
- Security Groups

**VPN Connections**

- Customer Gateways
- Virtual Private Gateways
- VPN Connections

**Route Tables**

Search Route Tables and their VPCs:

1 to 4 of 4 Route Tables

Name	Route Table ID	Explicitly Associated	Main	VPC
Sansbound_public_route	rtb-7d6de015	1 Subnet	Yes	vpc-09fe2261   Sansbound_VPC_Mumbai
sansbound Private route	rtb-b642dfde	1 Subnet	No	vpc-09fe2261   Sansbound_VPC_Mumbai
	rtb-91b209f9	0 Subnets	No	vpc-a655a2ce
	rtb-3dab7555	0 Subnets	Yes	vpc-a655a2ce

**rtb-7d6de015 | Sansbound\_public\_route**

Summary Routes Subnet Associations Route Propagation Tags

Cancel Save

View: All rules

Destination	Target	Status	Propagated	Remove
10.0.0.0/16	local	Active	No	
0.0.0.0/0	igw-2d587744	Active	No	
192.168.0.0/16	pcx-09ecc561305fd387a	No	No	

Add another route

Feedback English (US) © 2008 - 2018, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Detailed information of route table.

The screenshot shows the AWS VPC Dashboard for the region 'ap-south-1'. The left sidebar contains navigation links for VPC Dashboard, Virtual Private Cloud, Security, and VPN Connections. The main content area displays a list of Route Tables. The 'Sansbound\_public\_route' (rtb-7d6de015) is selected, and its details are shown below. The 'Routes' tab is active, displaying a table of routes.

**Route Tables List:**

Name	Route Table ID	Explicitly Associat	Main	VPC
Sansbound_public_route	rtb-7d6de015	1 Subnet	Yes	vpc-09fe2261   Sansbound_VPC_Mumbai
sansbound Private route	rtb-b642dfde	1 Subnet	No	vpc-09fe2261   Sansbound_VPC_Mumbai
	rtb-91b209f9	0 Subnets	No	vpc-a655a2ce
	rtb-3dab7555	0 Subnets	Yes	vpc-a655a2ce

**Selected Route Table: rtb-7d6de015 | Sansbound\_public\_route**

**Routes:**

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No
0.0.0.0/0	igw-2d587744	Active	No
192.168.0.0/16	eni-2a423a01 / i-09ecc561305fd387a	Active	No