

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 ✕

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.

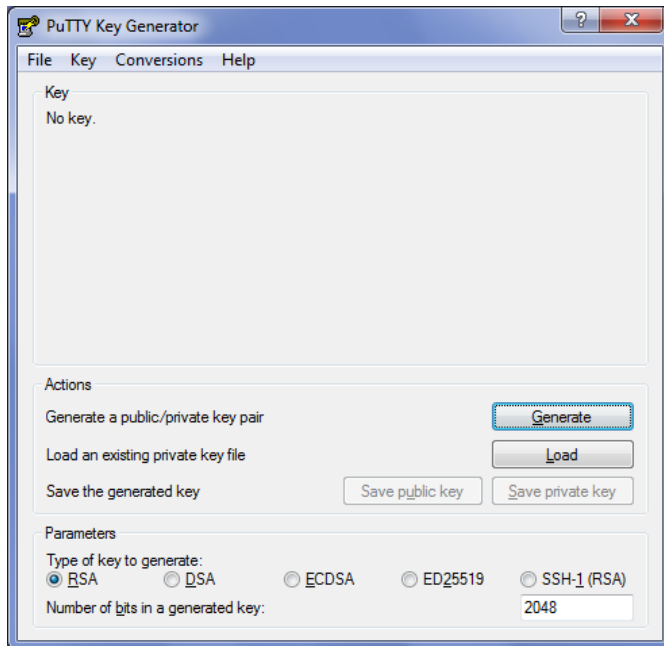
Cancel Save

Then click save.

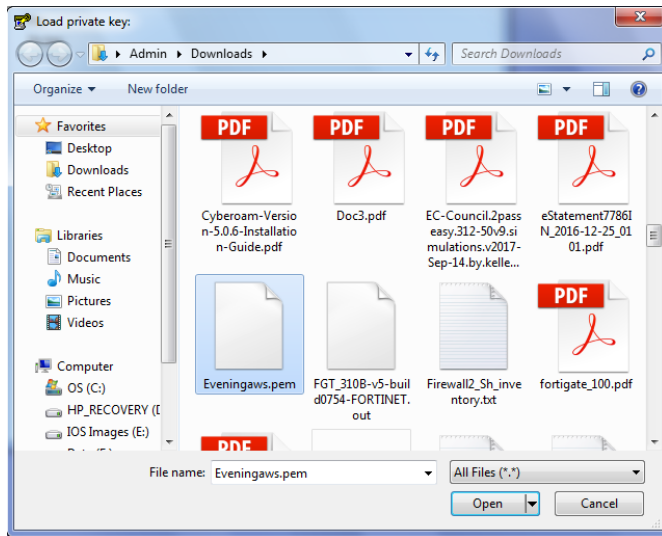
Goto Mumbai region to get public ip address of VPN Server Interface (13.127.161.231)

Launch putty key generator in your local machine,

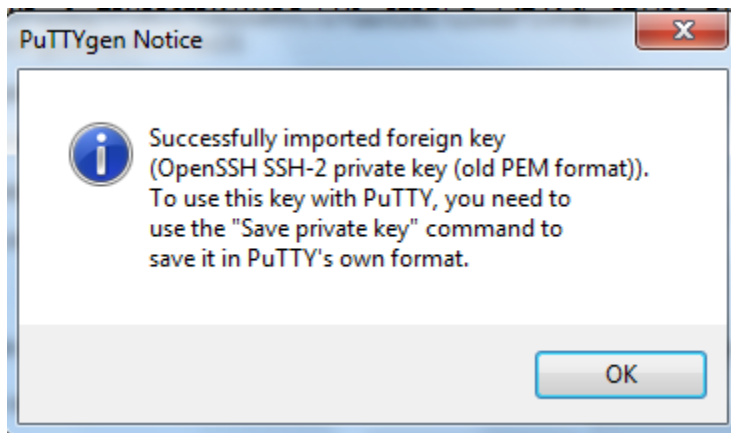
In File → Load private key



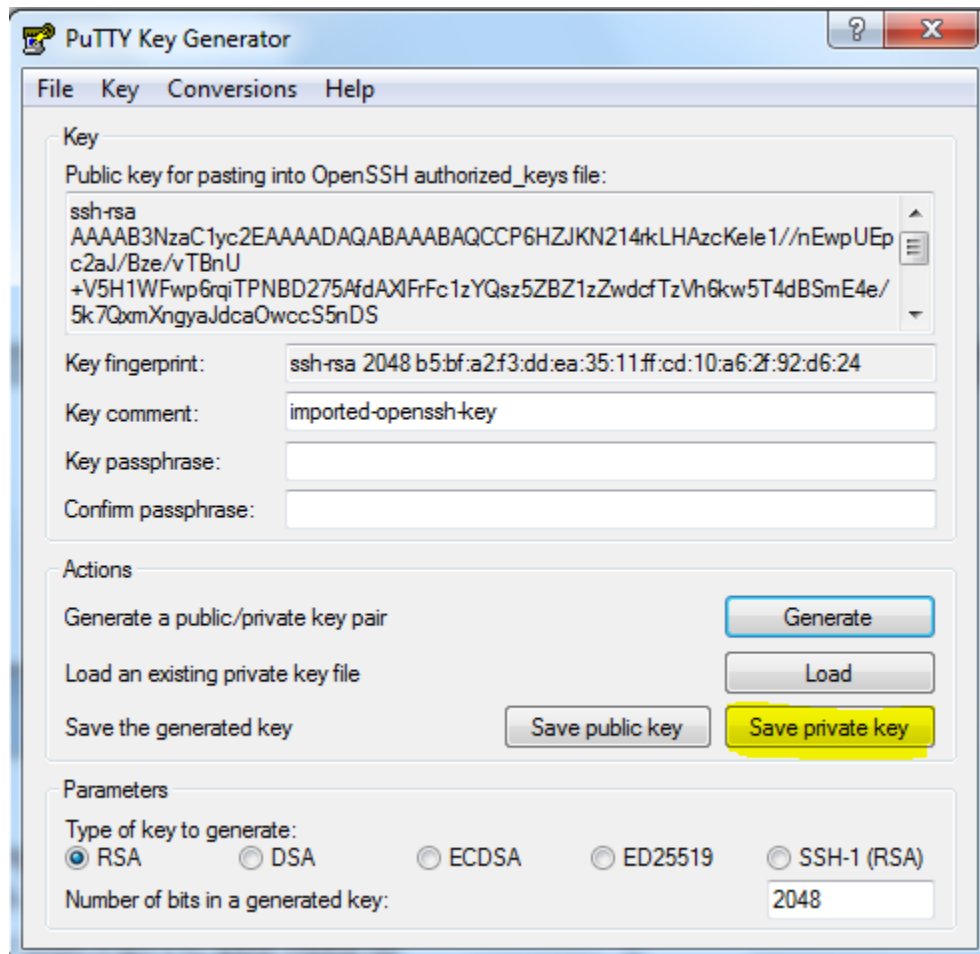
Locate the *.pem file and click “open”.



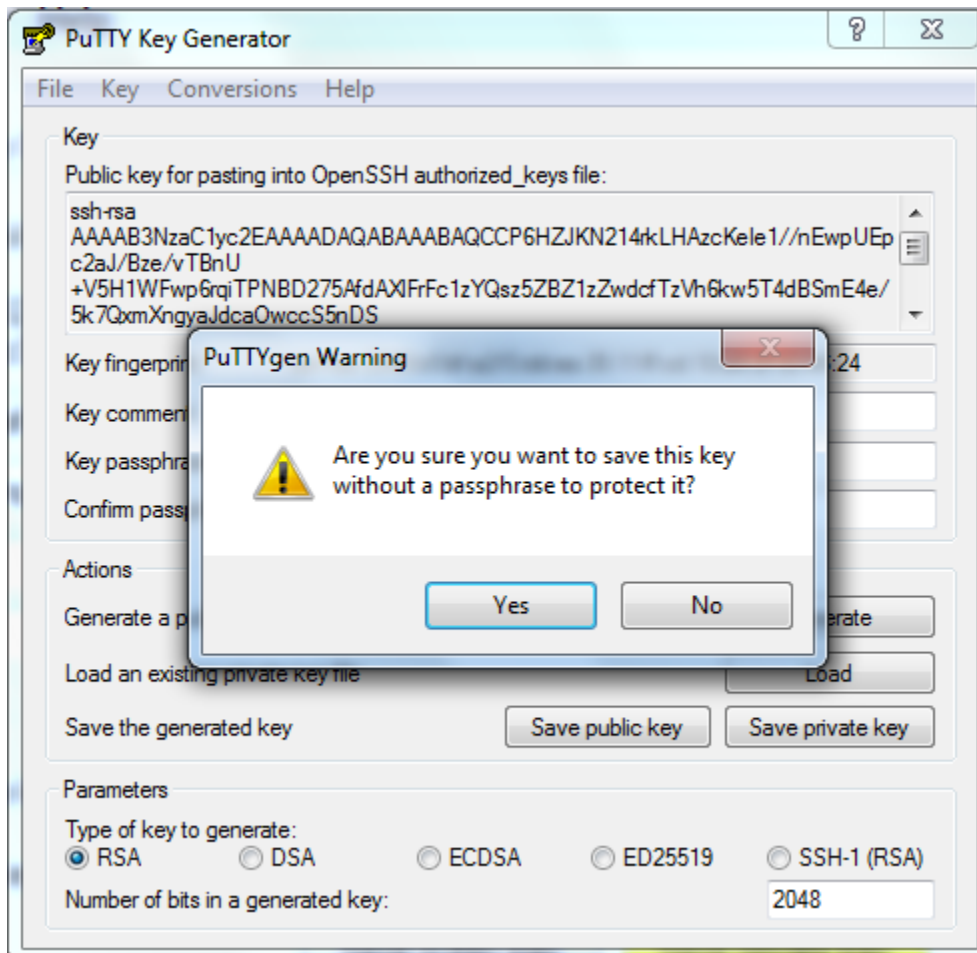
Click “Ok:”.



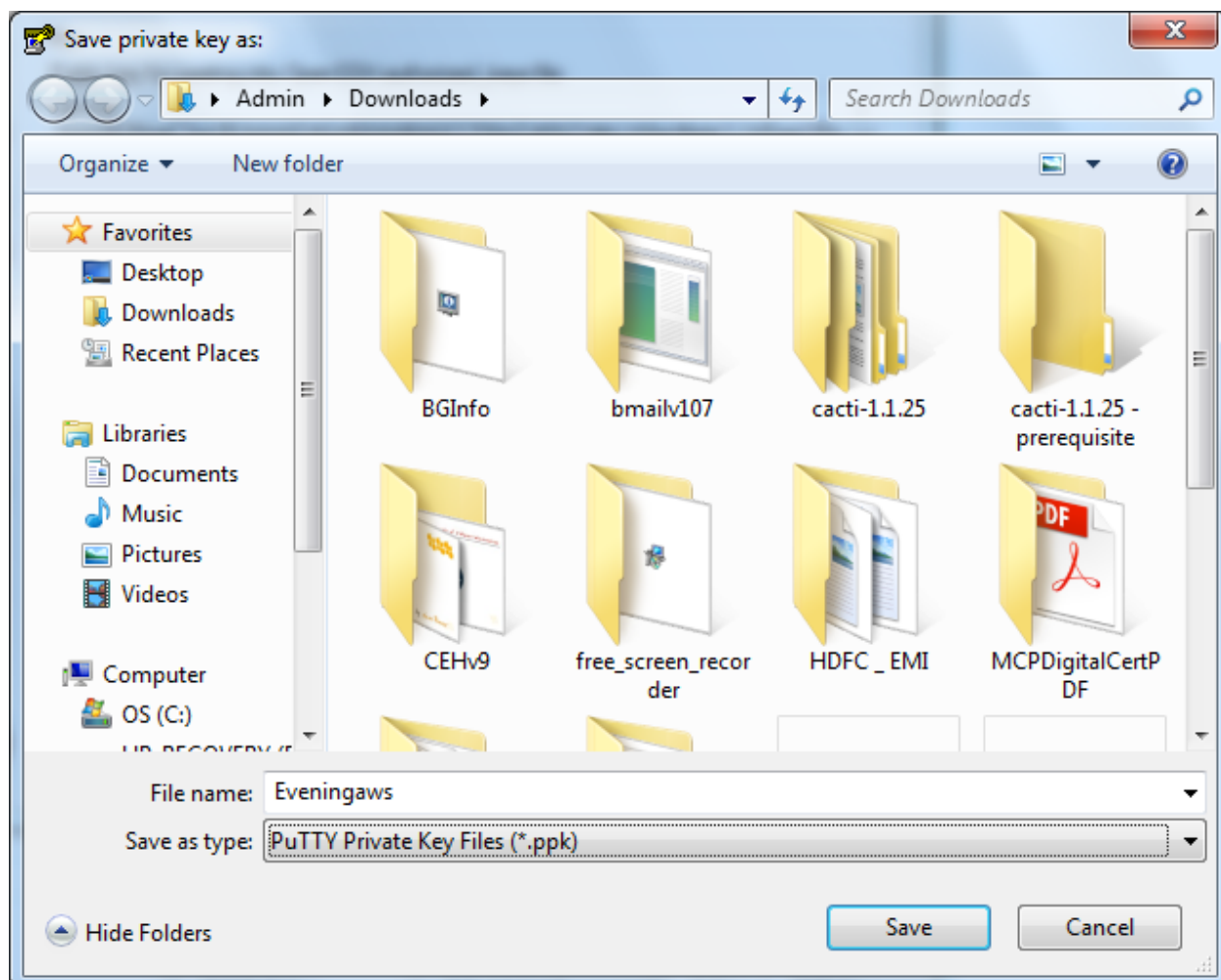
Click “save Private Key”.



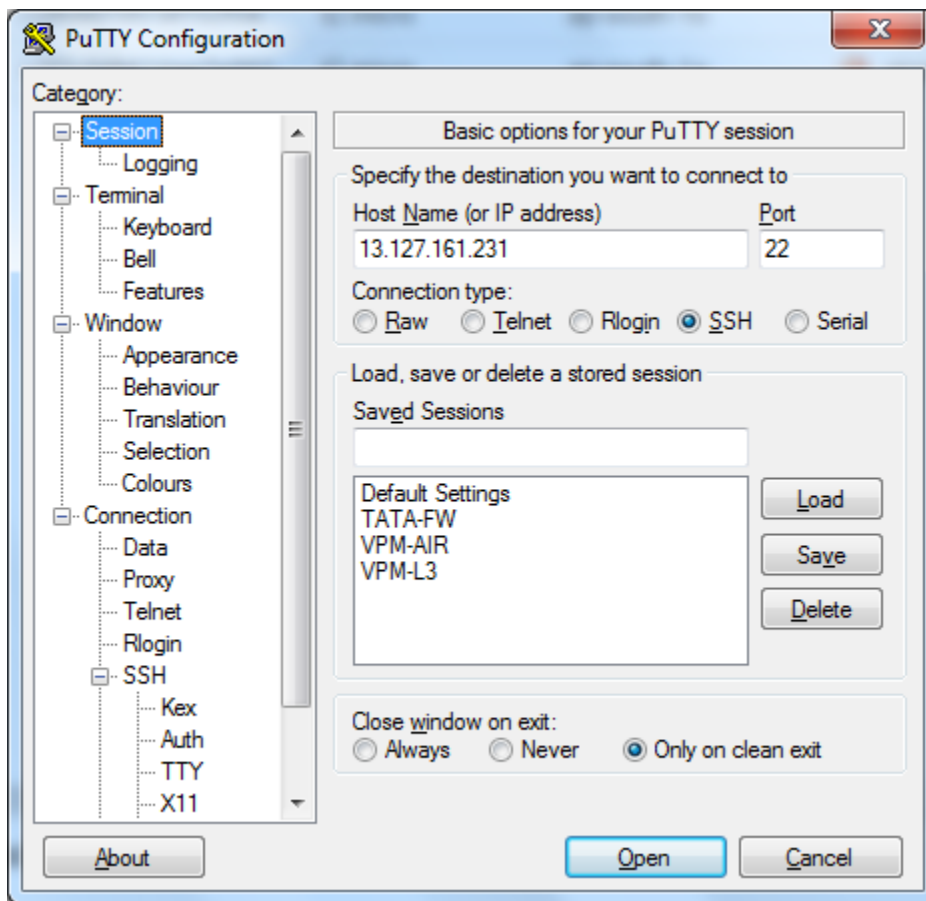
Click "Yes".



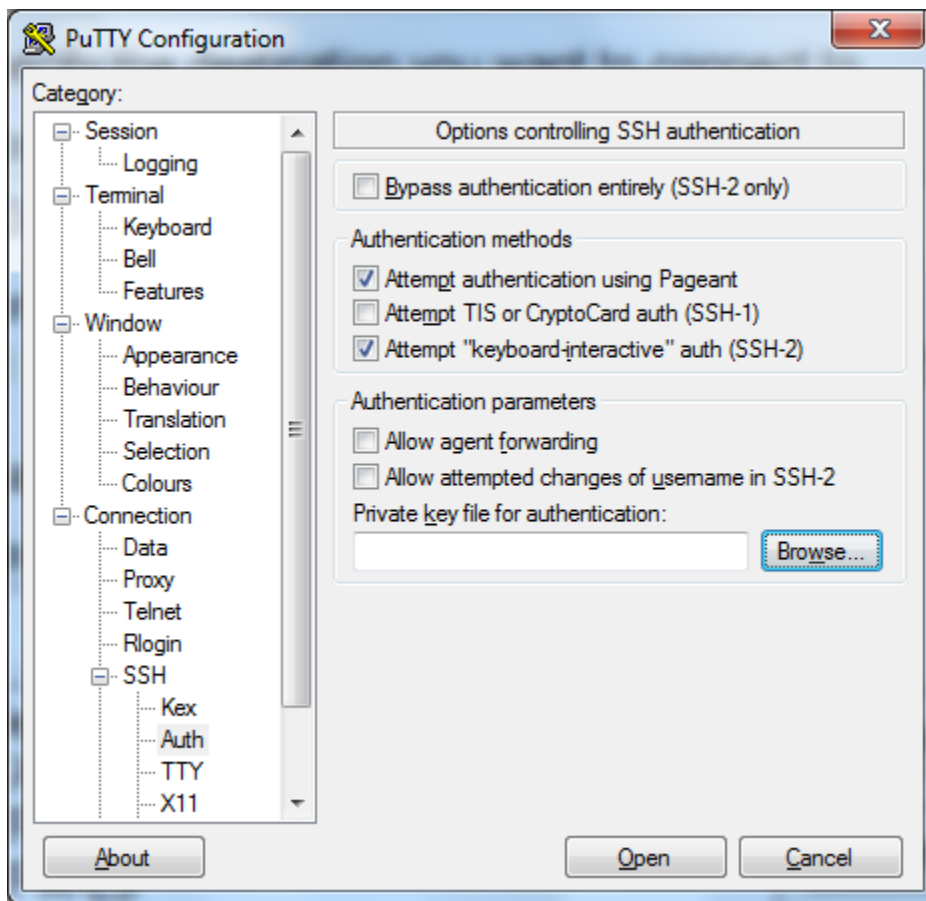
Save the private key in location.



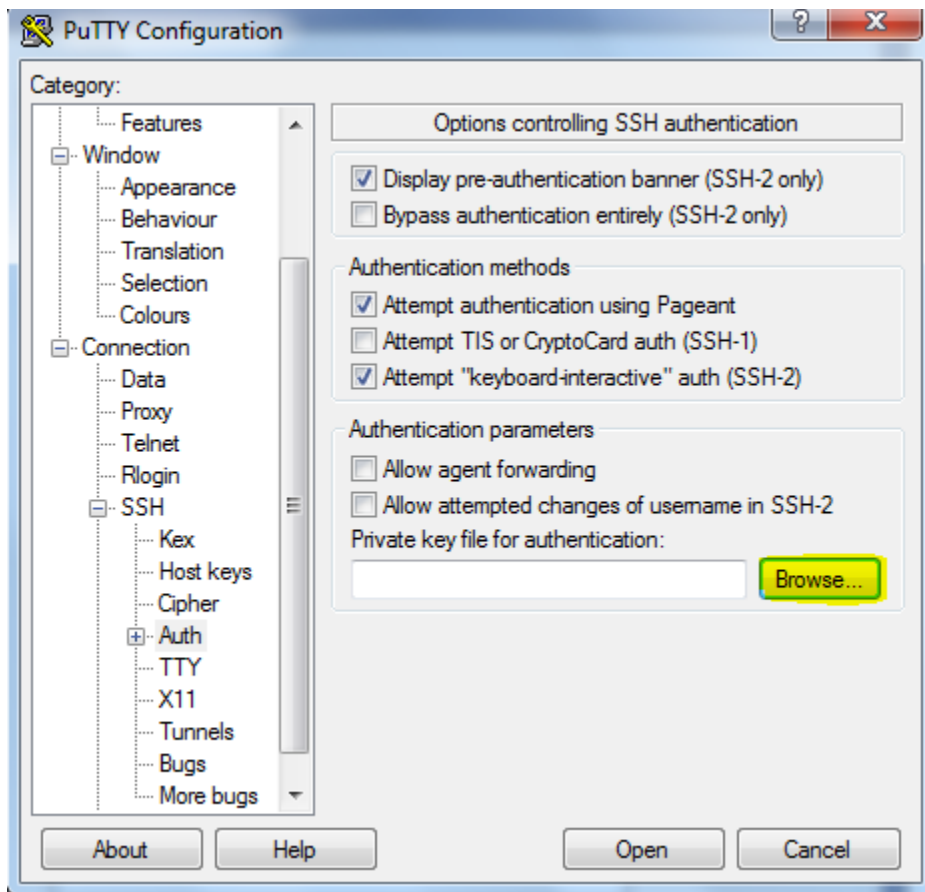
Type the ip address in putty.



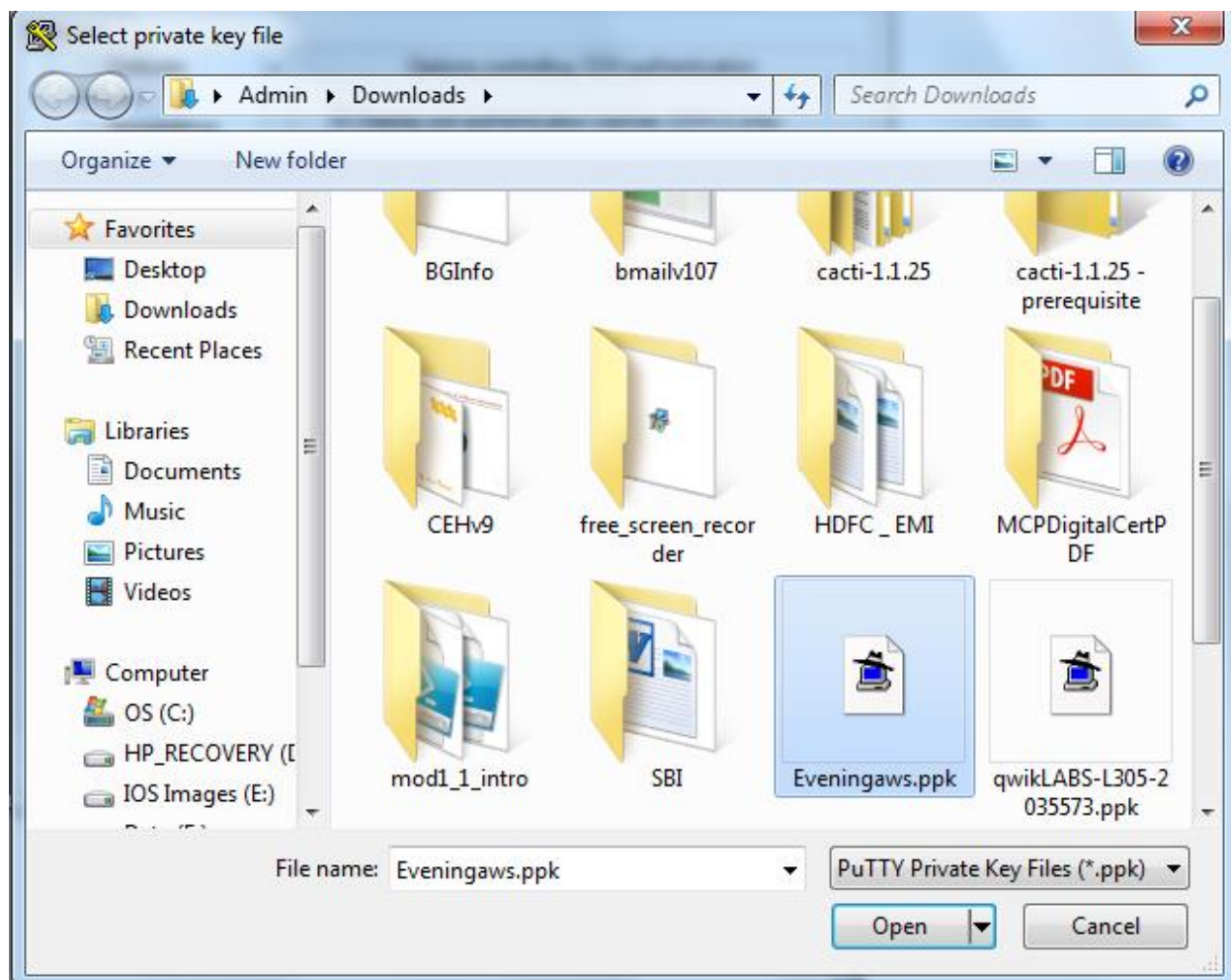
Click SSH and expand it click "Auth".



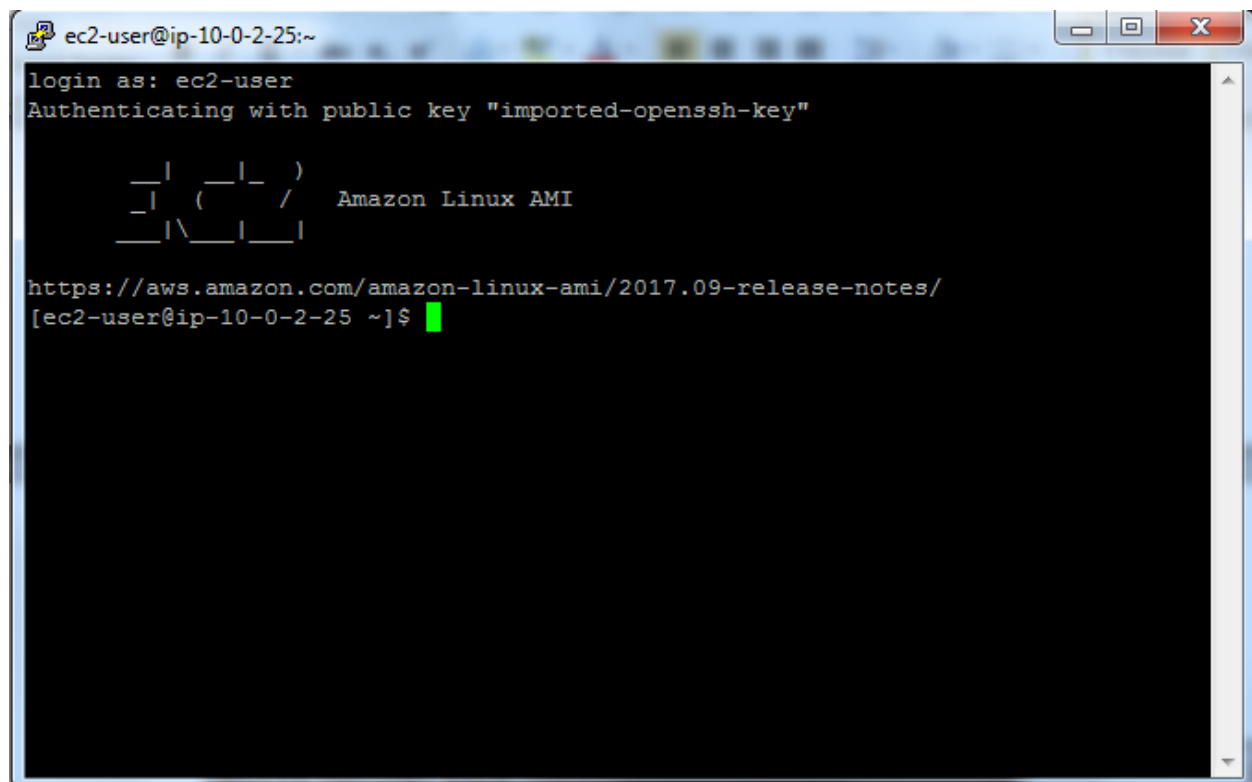
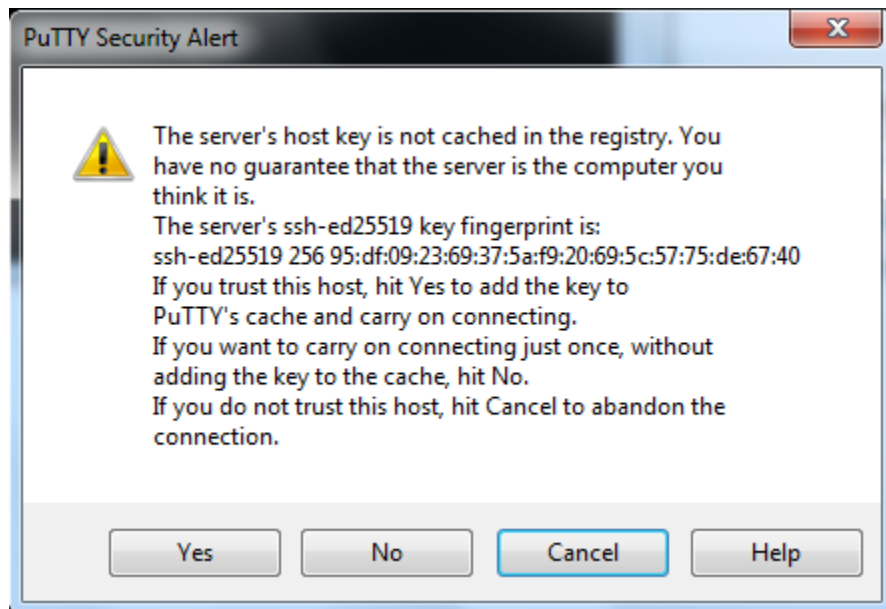
Click browse and locate the *.ppk file.



Locate the file and click “Open”.

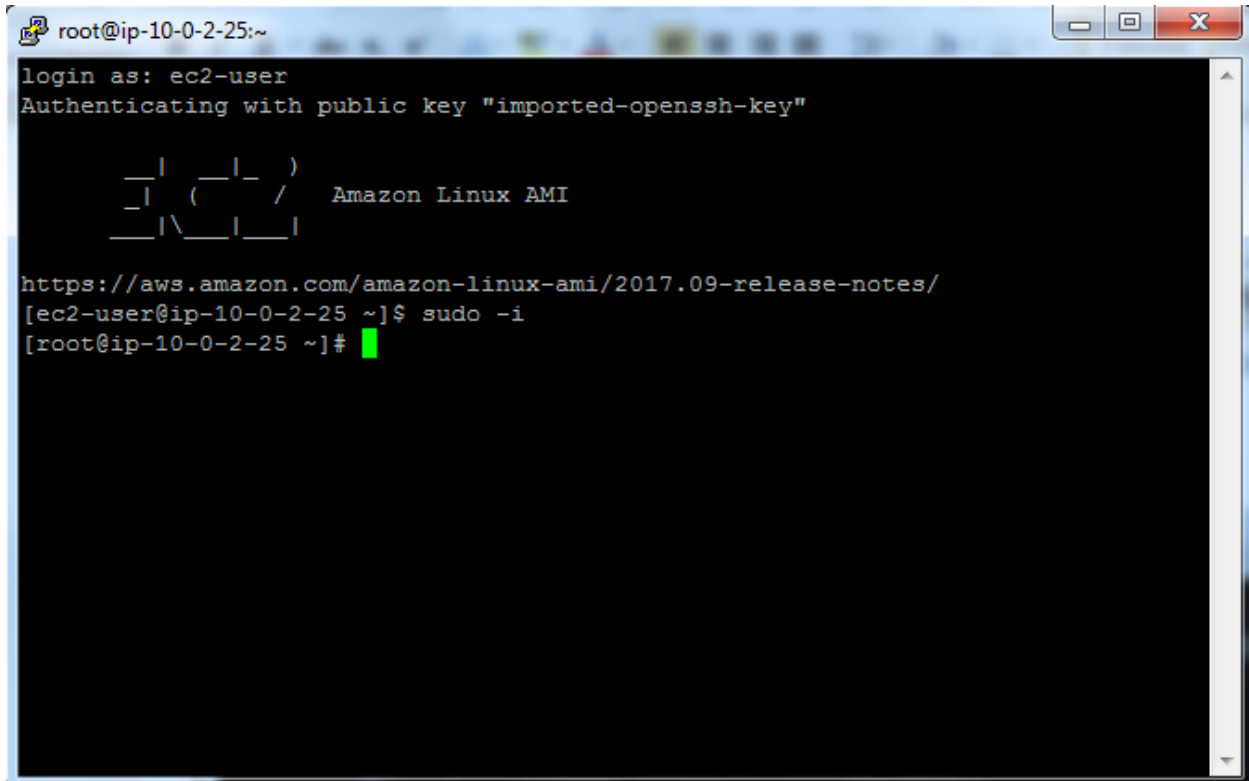


Click "Yes".



Type

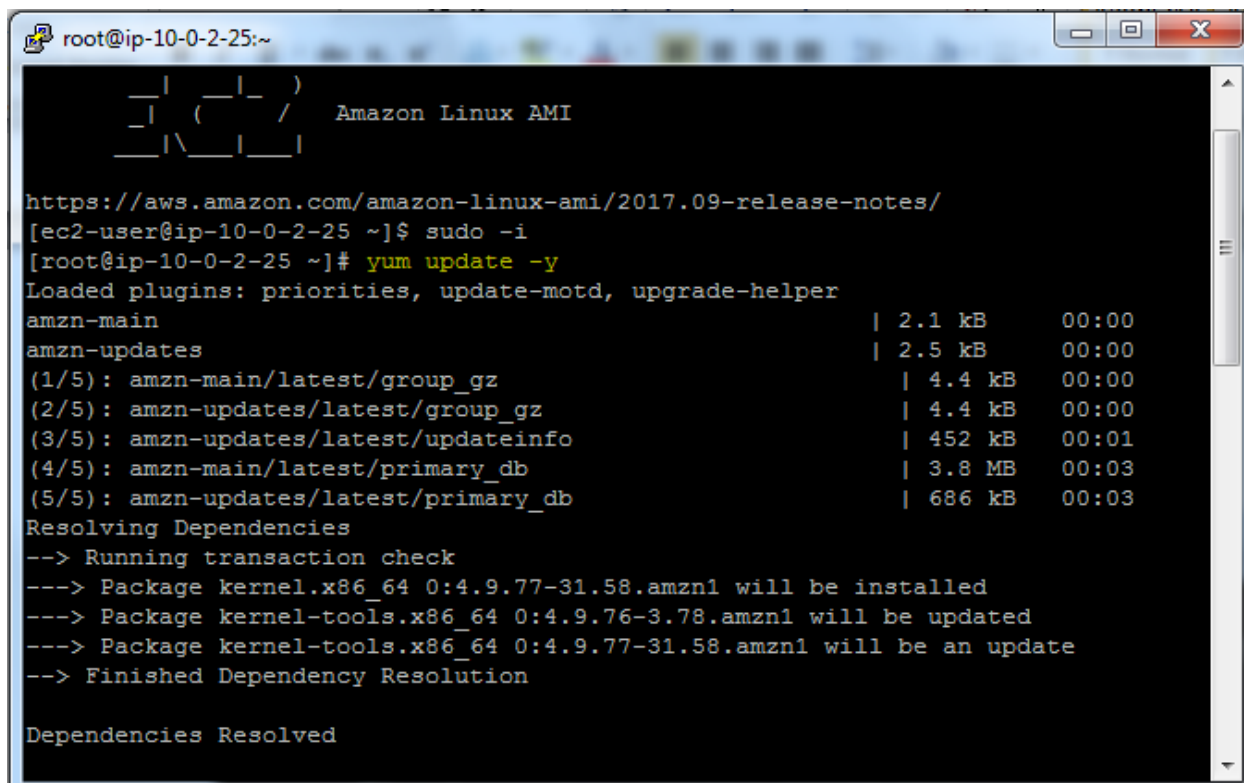
sudo-i



```
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

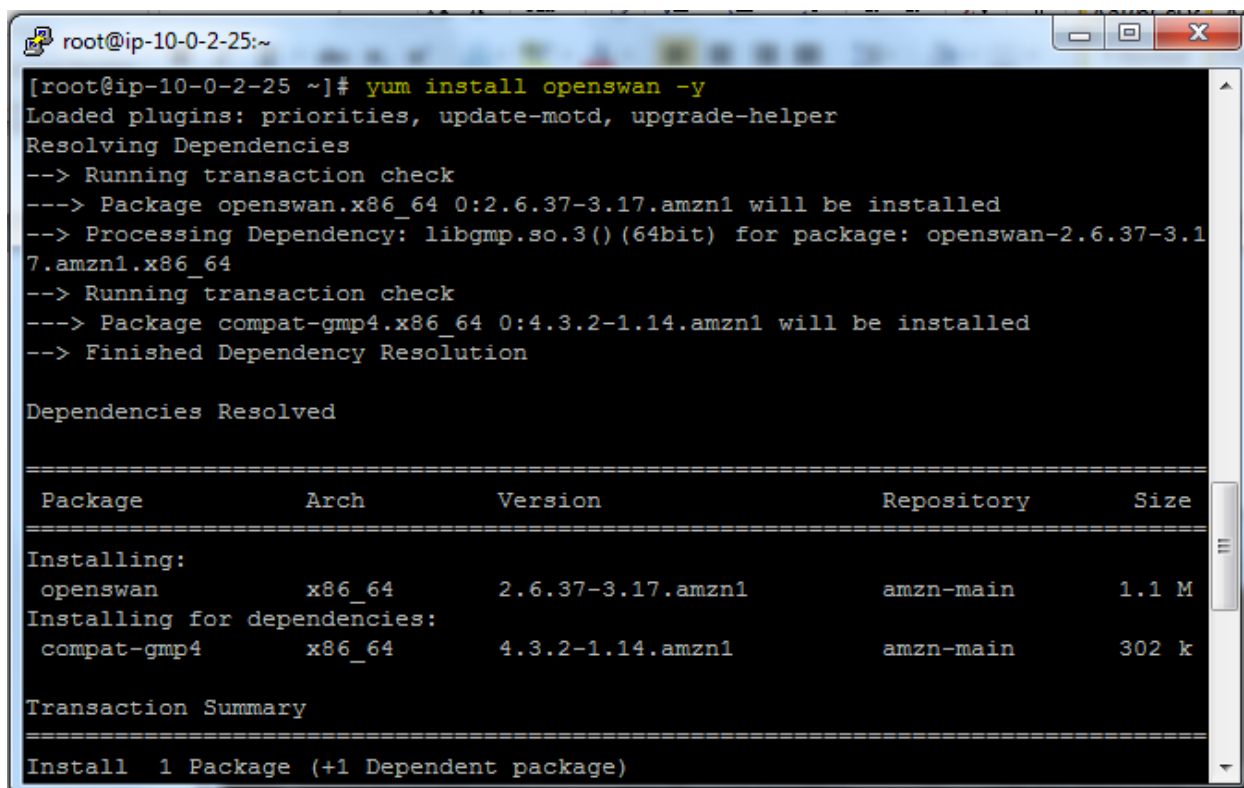
Yum update -y



```
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
```

Type

Yum install openswan -y



```
root@ip-10-0-2-25:~  
[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)

Vi ipsec.conf

```

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.
    # klipsedebug=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


```

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=
    ciscooff
    # Enable this if you see "failed to find any available worker"
    # nhelpers=0

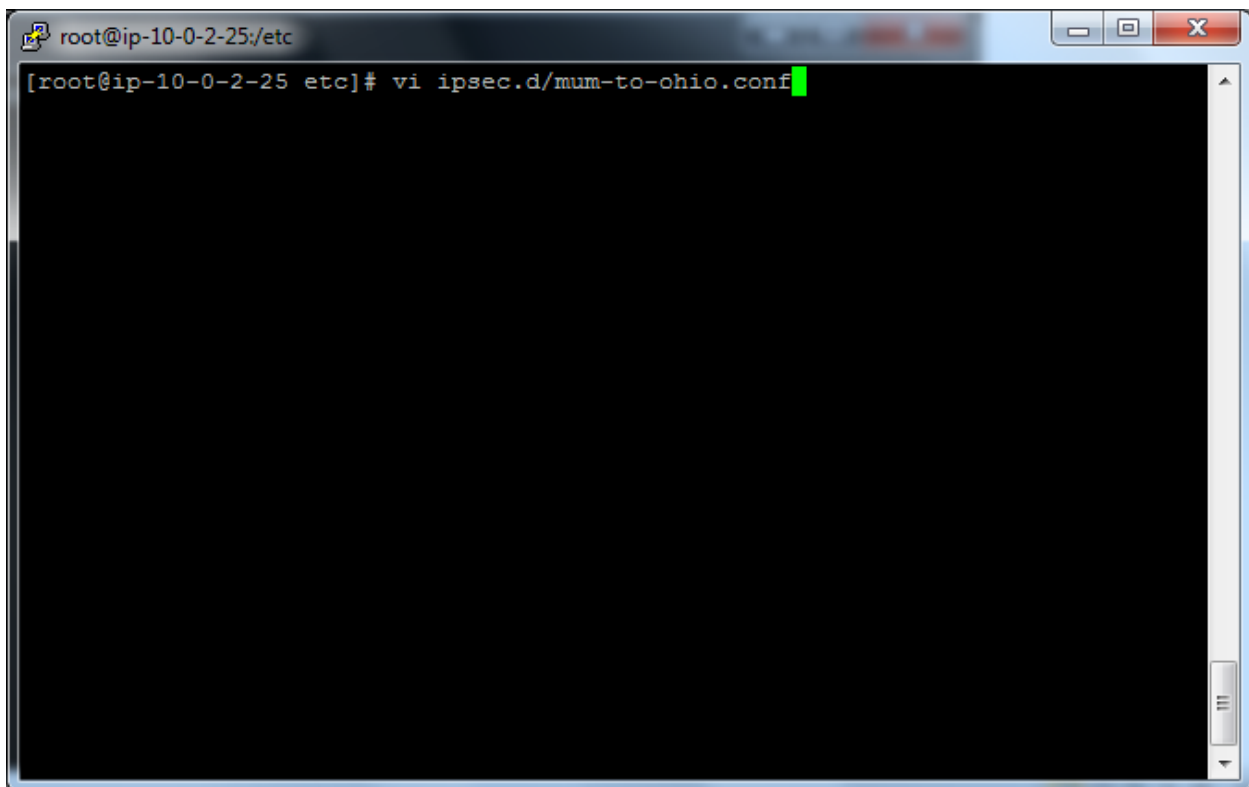
#You may put your configuration (.conf) file in the "/etc/ipsec.d/" and uncomment this.
include /etc/ipsec.d/*.conf

:wg

```

Type

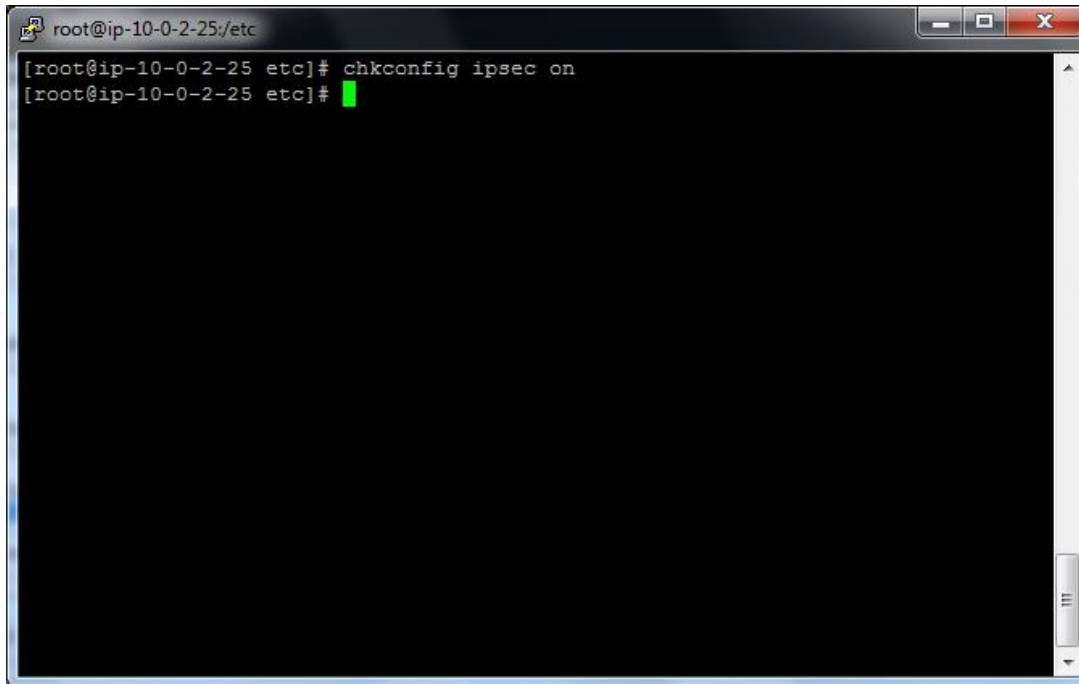
Vi ipsec.d/mum-to-ohio.conf



```
root@ip-10-0-2-25:/etc
[root@ip-10-0-2-25 etc]# vi ipsec.d/mum-to-ohio.conf
```

Type

Chkconfig ipsec on

A terminal window with a black background and white text. The title bar at the top reads 'root@ip-10-0-2-25:/etc'. The terminal shows the command '[root@ip-10-0-2-25 etc]# chkconfig ipsec on' being entered, followed by a new line with the prompt '[root@ip-10-0-2-25 etc]# ' and a green cursor. The window has standard Linux window controls (minimize, maximize, close) in the top right corner and a scrollbar on the right side.

```
root@ip-10-0-2-25:/etc
[root@ip-10-0-2-25 etc]# chkconfig ipsec on
[root@ip-10-0-2-25 etc]#
```

conn mum-to-ohio

authby=secret

```
leftid=13.127.161.231
```

```
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 displays the configuration for a tunnel named "mum-to-ohio". The configuration includes setting the type to "tunnel", authentication to "secret", and defining left and right endpoints with their respective subnets. The tunnel is configured to start automatically.

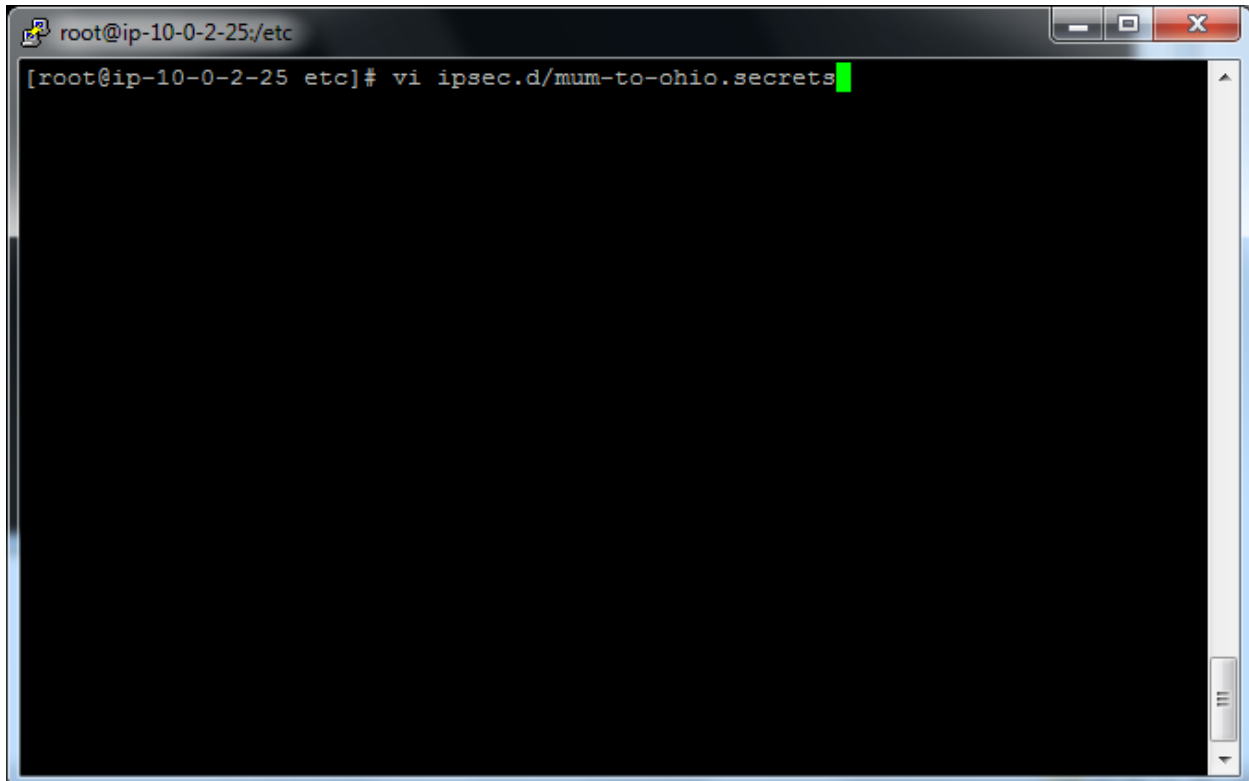
```
conn mum-to-ohio
    type=tunnel
    authby=secret
    left=defaultroute
    leftid=13.127.161.231
    leftrightnextHop=%defaultRoute
    leftsubnet=10.0.0.0/16
    right=18.218.11.25
    rightsubnet=192.168.0.0/16
    pfs=yes
    auto=start
```

The terminal shows several tilde (~) characters indicating the prompt character, followed by a colon and the letter 'w' at the bottom.

Press escape and type :wq

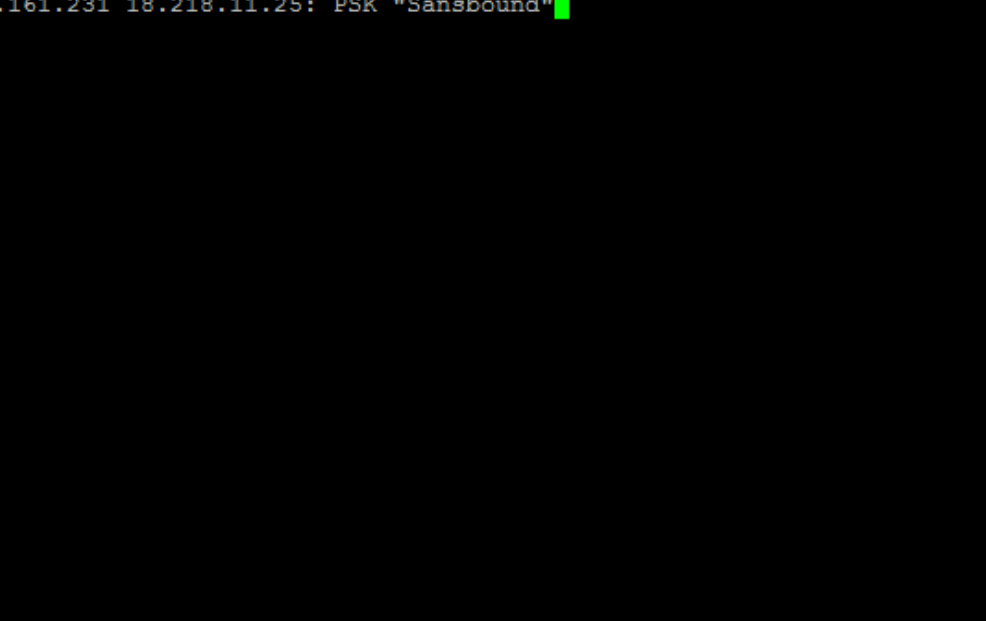
Type

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

A terminal window with a title bar showing 'root@ip-10-0-2-25:/etc'. The terminal content shows a root prompt '[root@ip-10-0-2-25 etc]#' followed by the command 'vi ipsec.d/mum-to-ohio.secrets'. A green cursor is positioned at the end of the command line. The terminal window has standard Linux window controls (minimize, maximize, close) in the top right corner and a scrollbar on the right side.

```
root@ip-10-0-2-25:/etc
[root@ip-10-0-2-25 etc]# vi ipsec.d/mum-to-ohio.secrets
```

Our Tunnel Preshared key is “Sansbound”

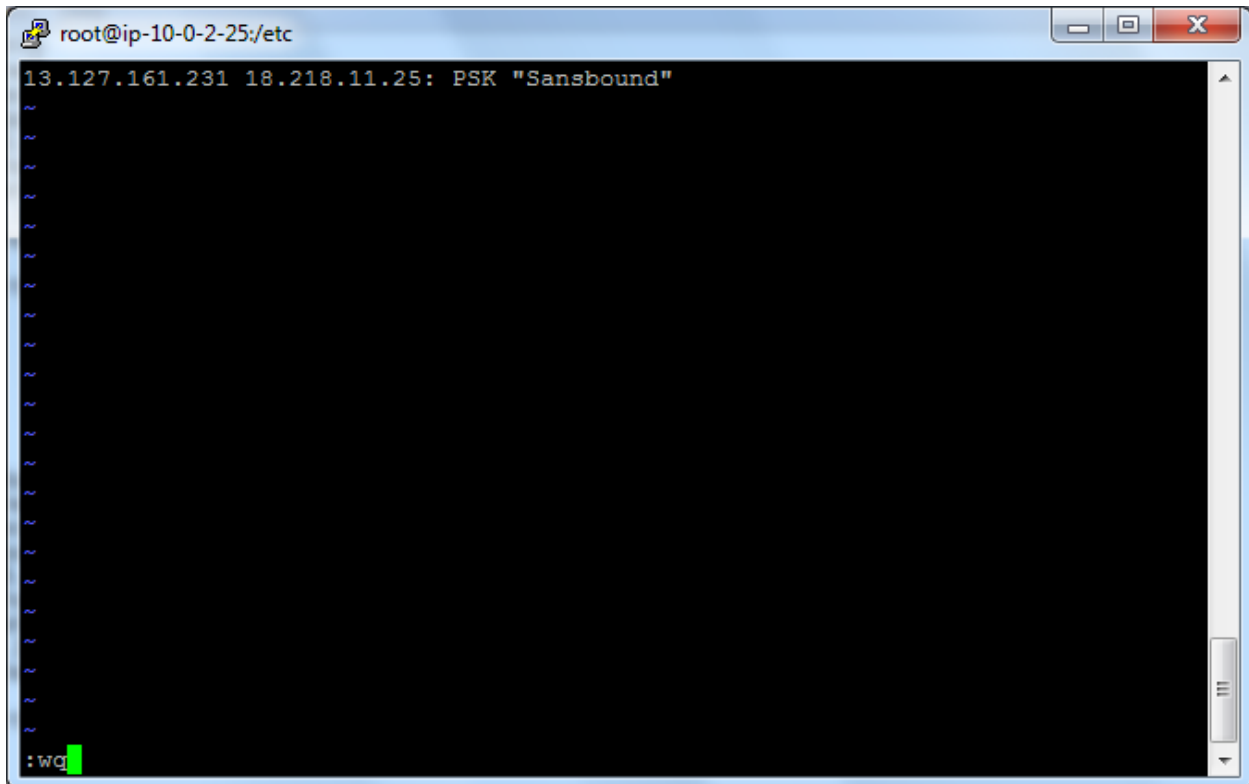


A screenshot of a terminal window. The title bar at the top reads "root@ip-10-0-2-25:/etc". The terminal content shows a root prompt followed by the command "13.127.161.231 18.218.11.25: PSK 'Sansbound'", with a green cursor at the end. Below the command, there are several tilde (~) characters. At the bottom left, the text "-- INSERT --" is visible. The terminal has a standard window interface with minimize, maximize, and close buttons in the top right corner.

```
root@ip-10-0-2-25:/etc
13.127.161.231 18.218.11.25: PSK "Sansbound"
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --
```

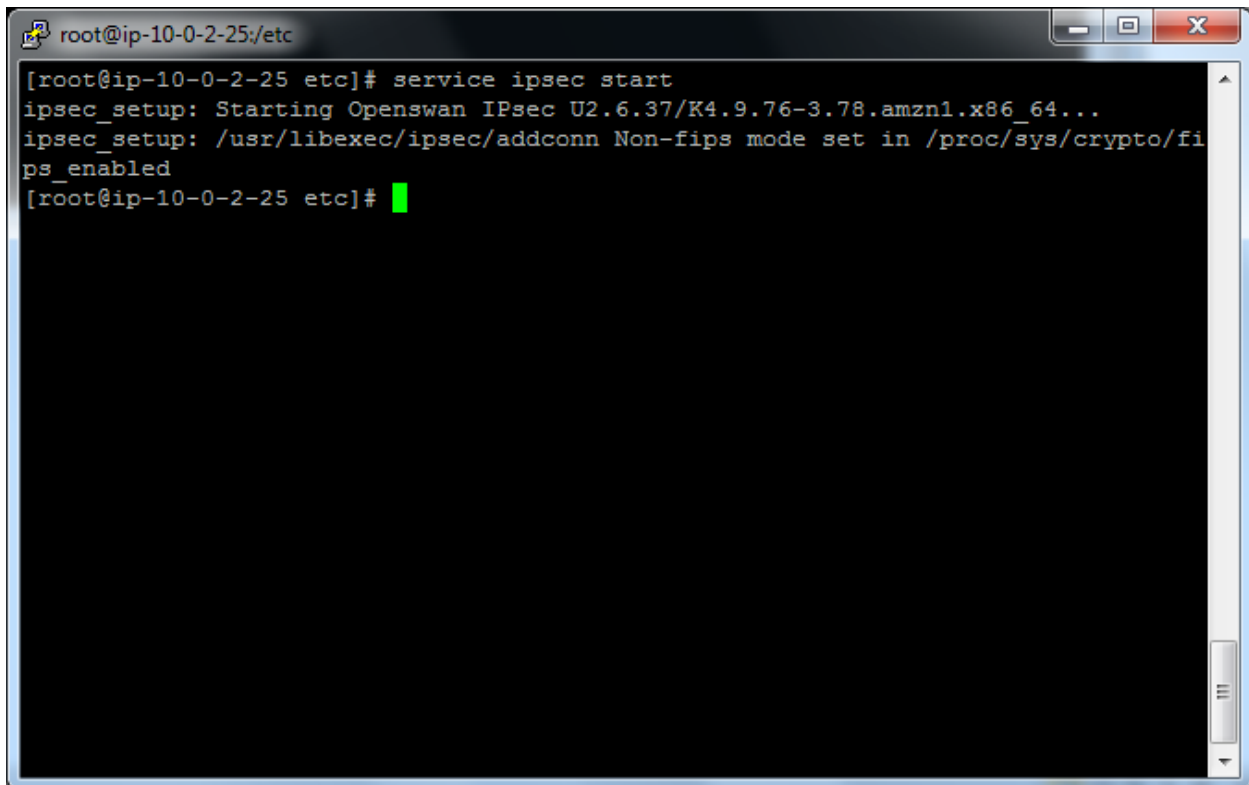
Press escape key

Type :wq



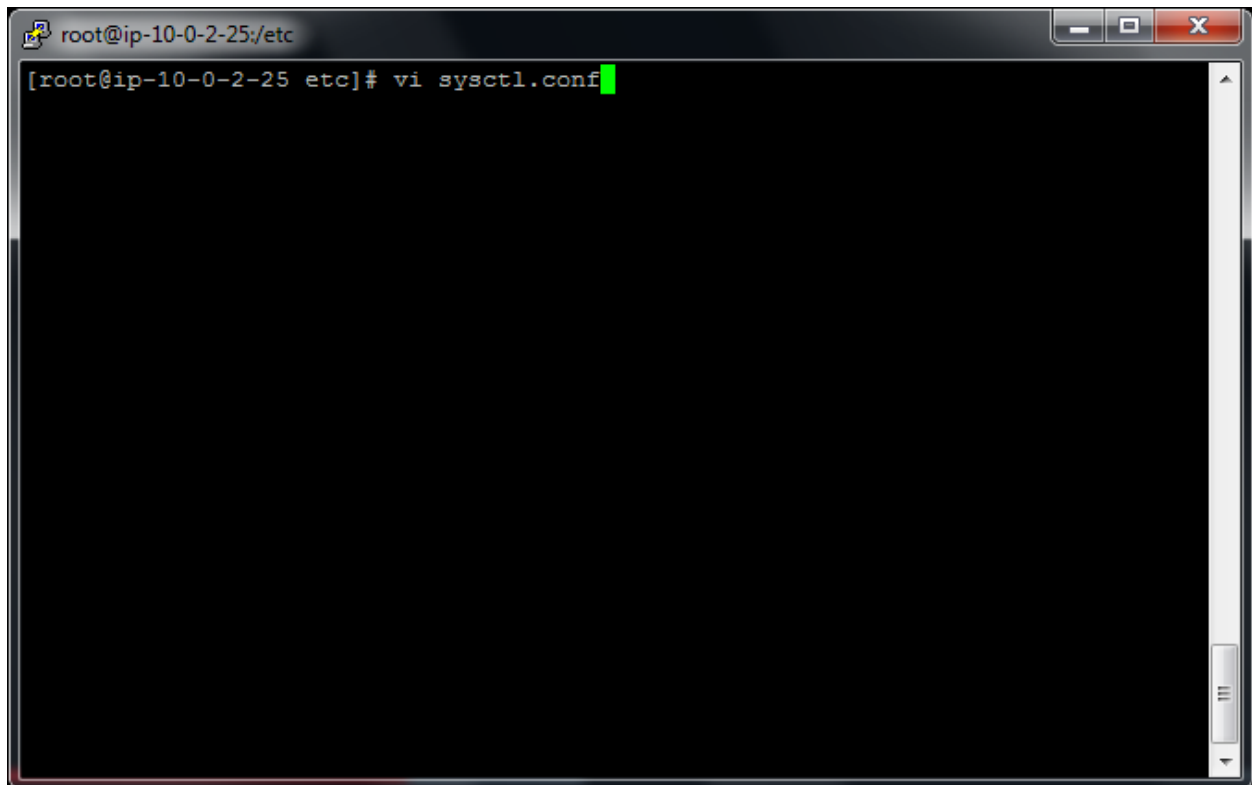
```
root@ip-10-0-2-25:/etc
13.127.161.231 18.218.11.25: PSK "Sansbound"
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
:wq
```

Service ipsec start



```
root@ip-10-0-2-25:/etc
[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



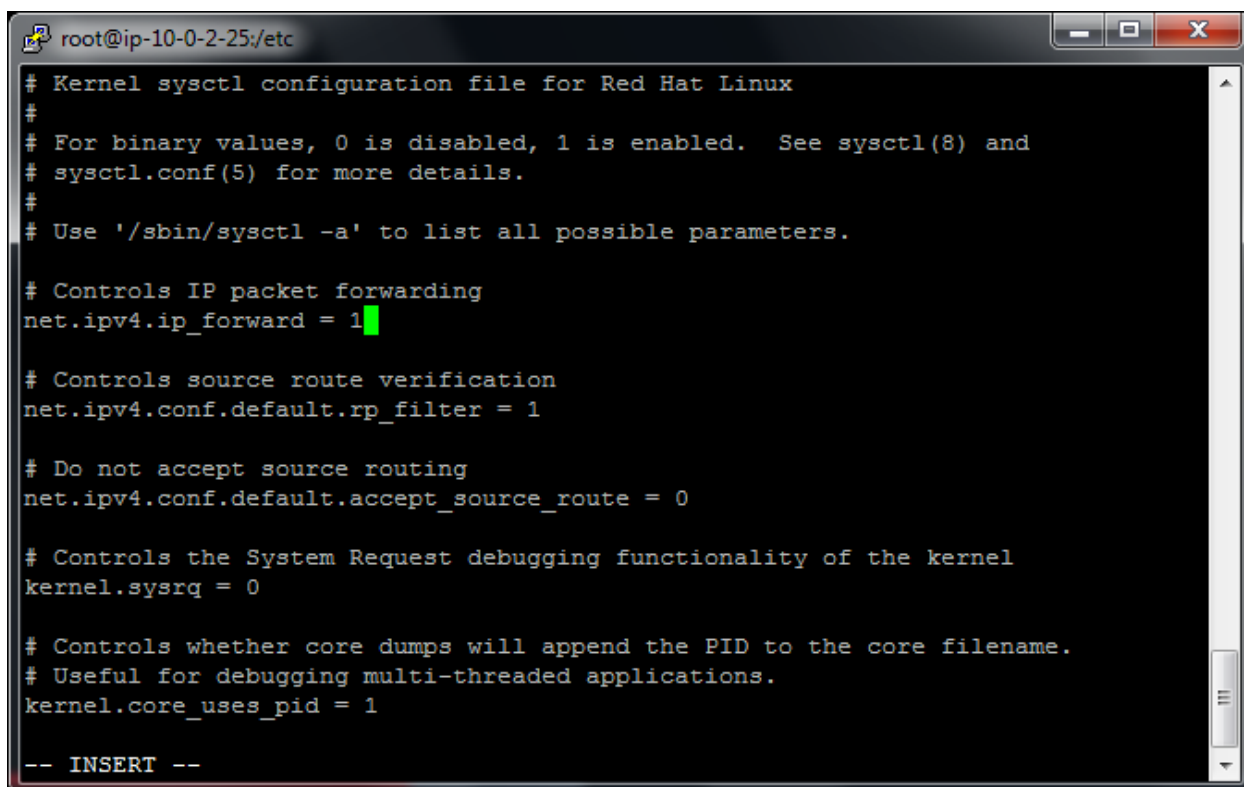
A terminal window with a dark background and a light gray title bar. The title bar contains the text "root@ip-10-0-2-25:/etc" and standard window control buttons (minimize, maximize, close). The terminal content shows a root prompt "[root@ip-10-0-2-25 etc]#" followed by the command "vi sysctl.conf". A green cursor is positioned at the end of the command.

```
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
```



```
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

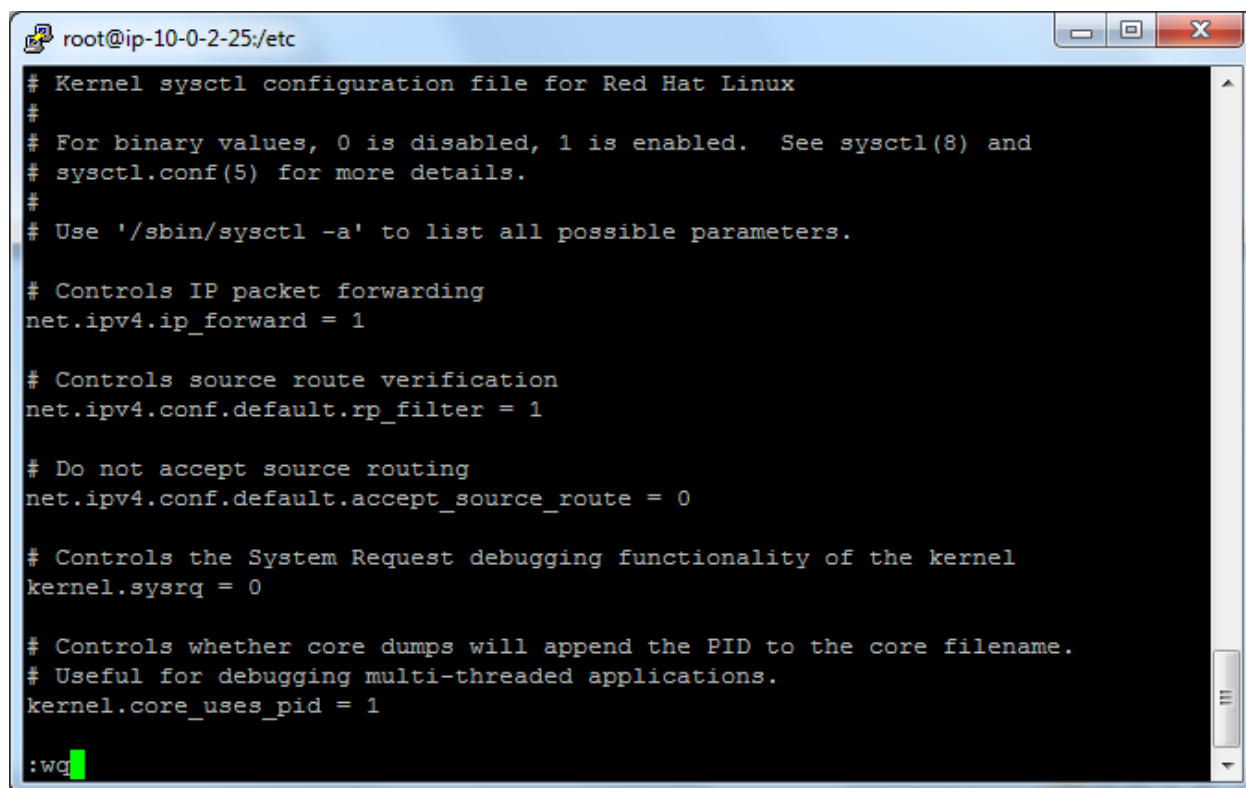
# Do not accept source routing
net.ipv4.conf.default.accept_source_route = 0

# 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

-- INSERT --
```

Press “Escape” key



```
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
```

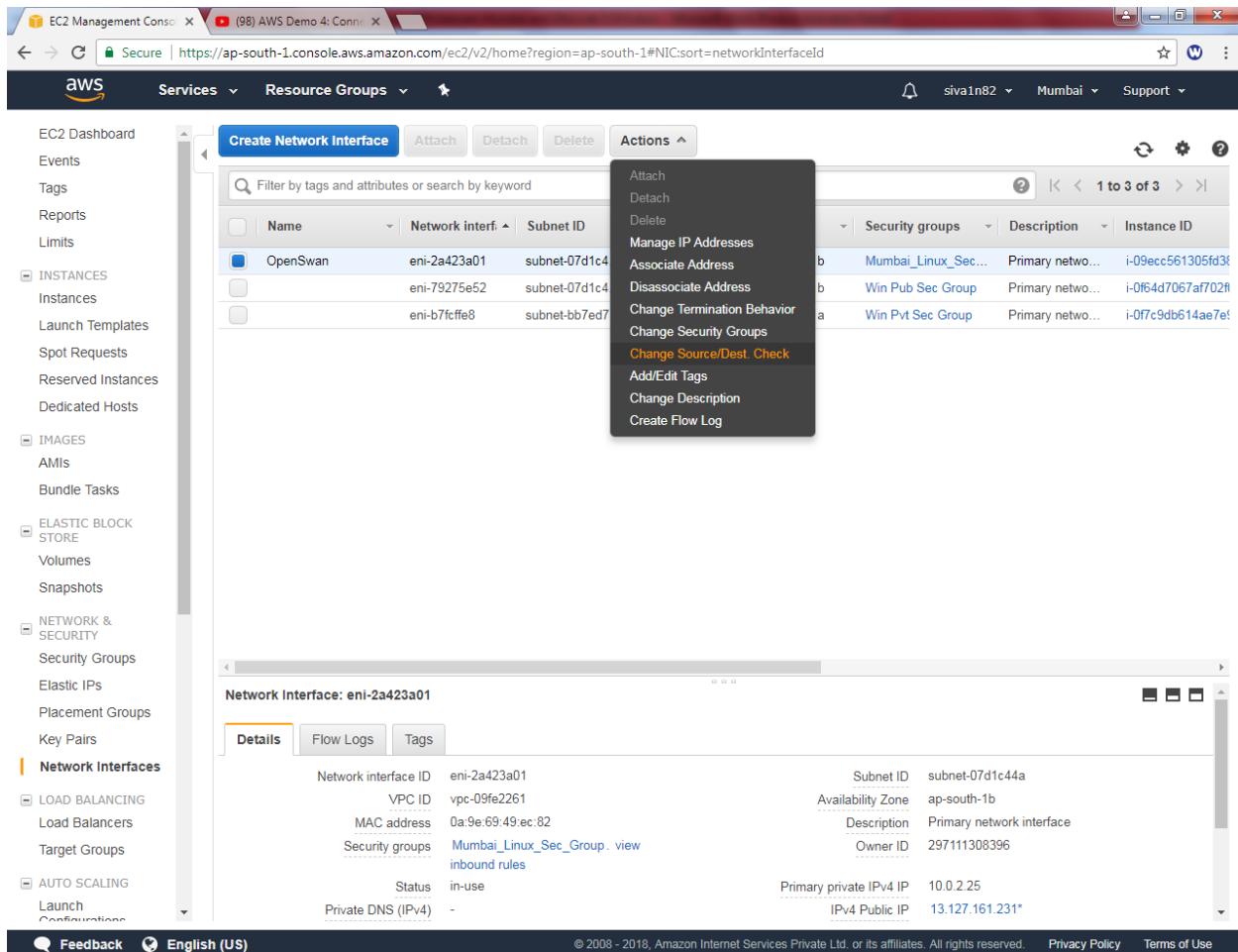
and then type

: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 for the 'ap-south-1' region. The left sidebar shows the navigation menu with 'Network Interfaces' selected. The main content area displays a table of network interfaces. The 'OpenSwan' interface is selected, and the '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', 'Add/Edit Tags', 'Change Description', and 'Create Flow Log'. The 'Change Source/Dest. Check' option is highlighted.

Name	Network interf.	Subnet ID	Security groups	Description	Instance ID
OpenSwan	eni-2a423a01	subnet-07d1c4	Mumbai_Linux_Sec...	Primary netwo...	i-09ecc561305fd3f
	eni-79275e52	subnet-07d1c4	Win Pub Sec Group	Primary netwo...	i-0f64d7067af702f
	eni-b7fcfe8	subnet-bb7ed7	Win Pvt Sec Group	Primary netwo...	i-0f7c9db614ae7e

Network Interface: eni-2a423a01

Property	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 ×

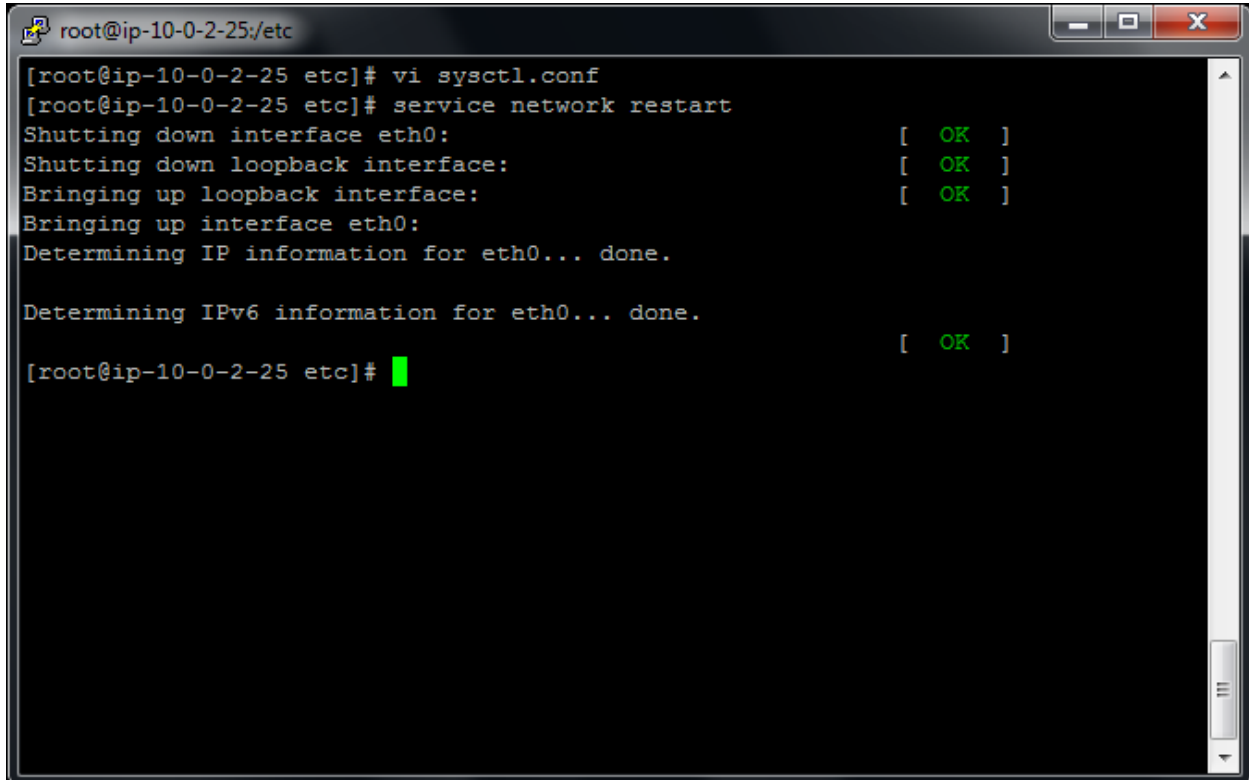
Network Interface eni-2a423a01

Source/dest. check ☐ Enabled
☒ Disabled

Cancel Save

Type

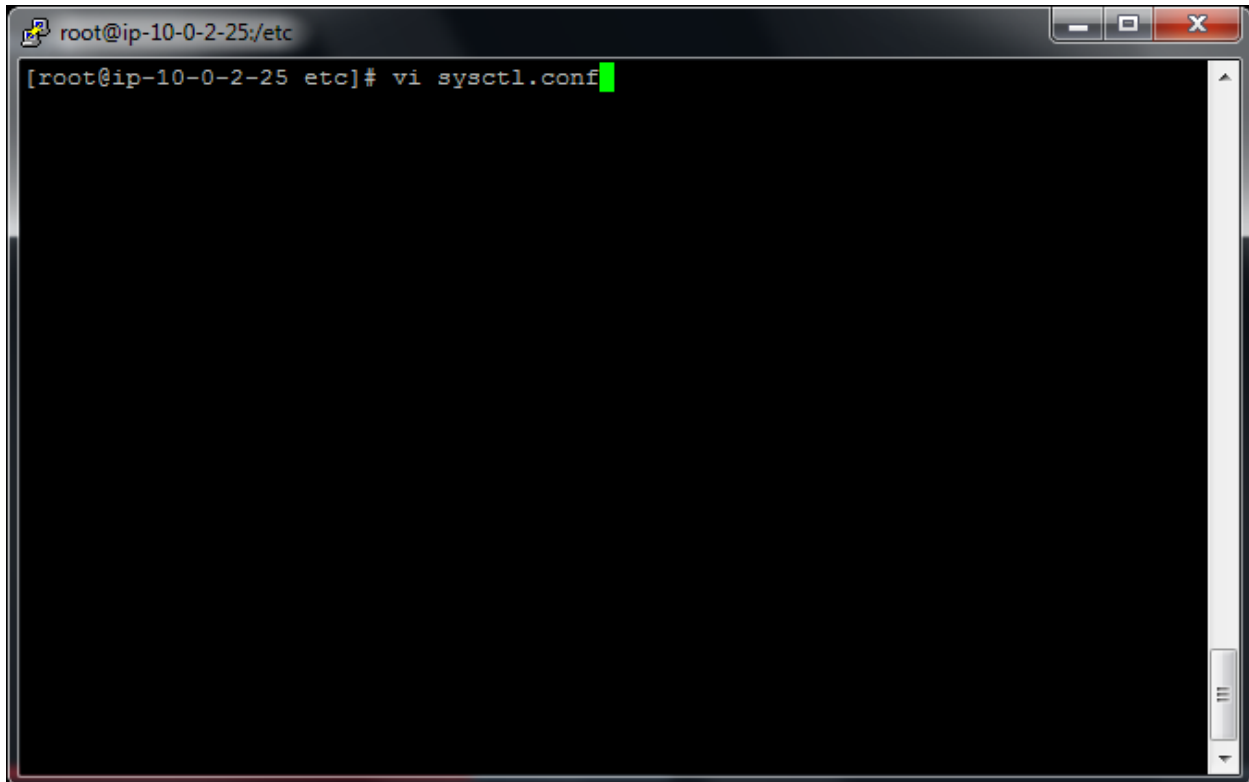
Service network restart

A terminal window titled 'root@ip-10-0-2-25:/etc' with standard window controls. The terminal shows the execution of 'vi sysctl.conf' followed by 'service network restart'. The output indicates the successful restart of the network service, including shutting down and bringing up both the loopback and eth0 interfaces, and determining their IP and IPv6 information.

```
root@ip-10-0-2-25:/etc
[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



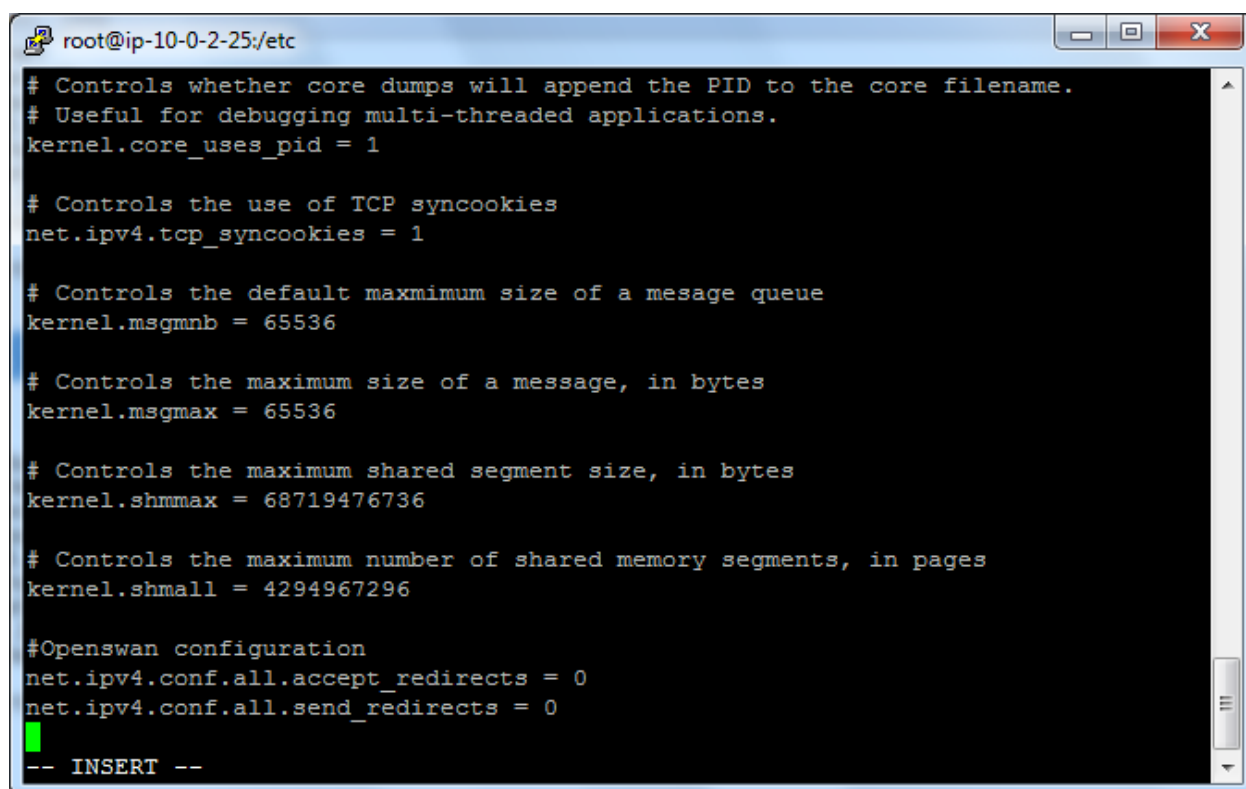
```
root@ip-10-0-2-25:/etc
[root@ip-10-0-2-25 etc]# vi sysctl.conf
```

Press insert key

Type

net.ipv4.conf.all.accept_redirects = 0

net.ipv4.conf.all.send_redirects = 0

A terminal window titled 'root@ip-10-0-2-25:/etc' with standard window controls. The terminal displays kernel and network configuration parameters. At the bottom, a green cursor is on a line with '-- INSERT --', indicating the terminal is in insert mode.

```
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

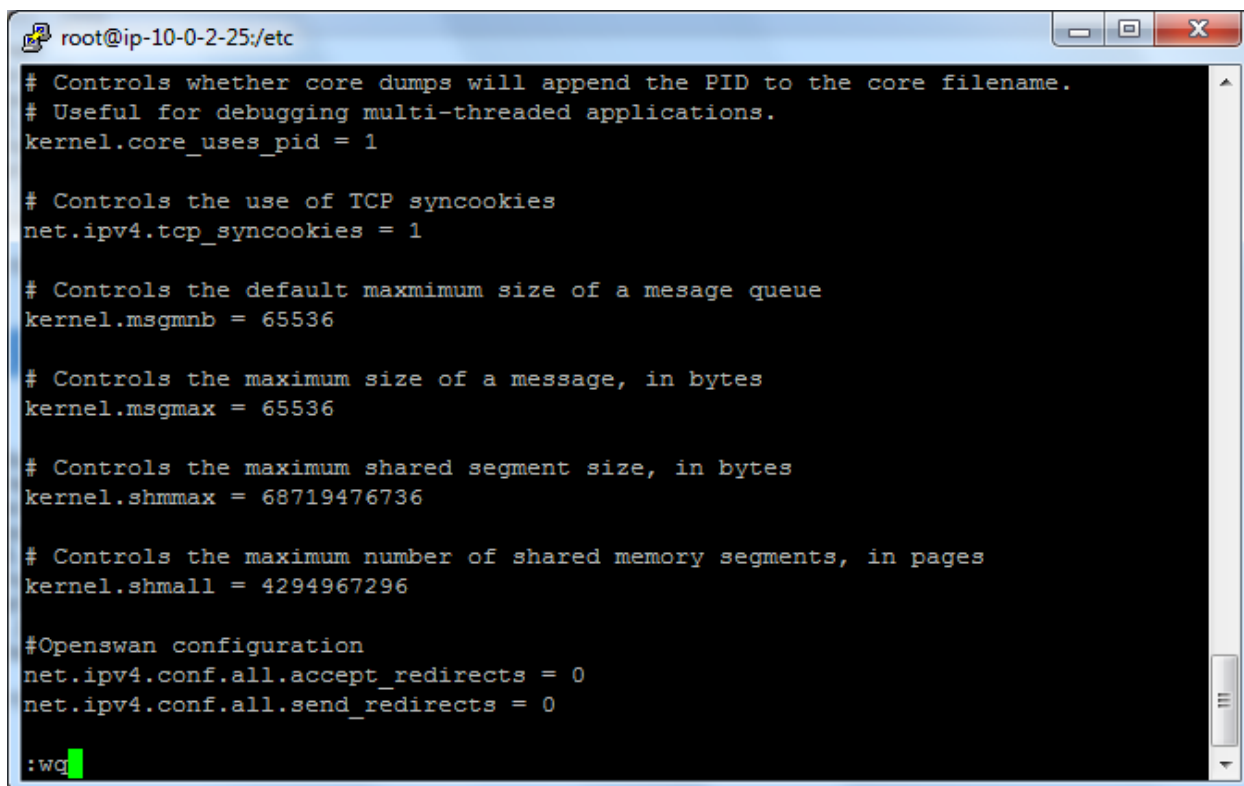
# Controls the maximum shared segment size, in bytes
kernel.shmmax = 68719476736

# 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 terminal window titled 'root@ip-10-0-2-25:/etc' with standard window controls. The terminal displays a configuration file with several commented-out lines and assigned values. At the bottom, the command ':wq' is entered, followed by a green cursor.

```
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

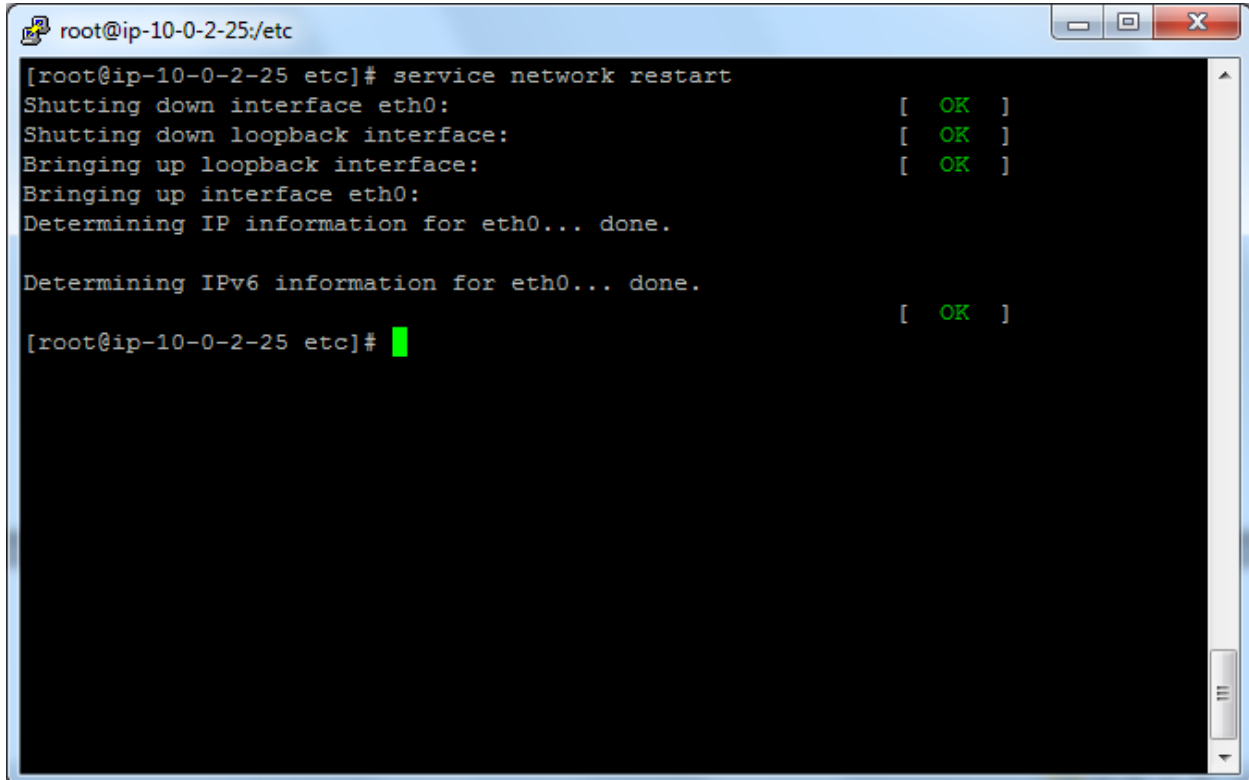
# Controls the maximum number of shared memory segments, in pages
kernel.shmall = 4294967296

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

:wq
```

Type

Service network restart

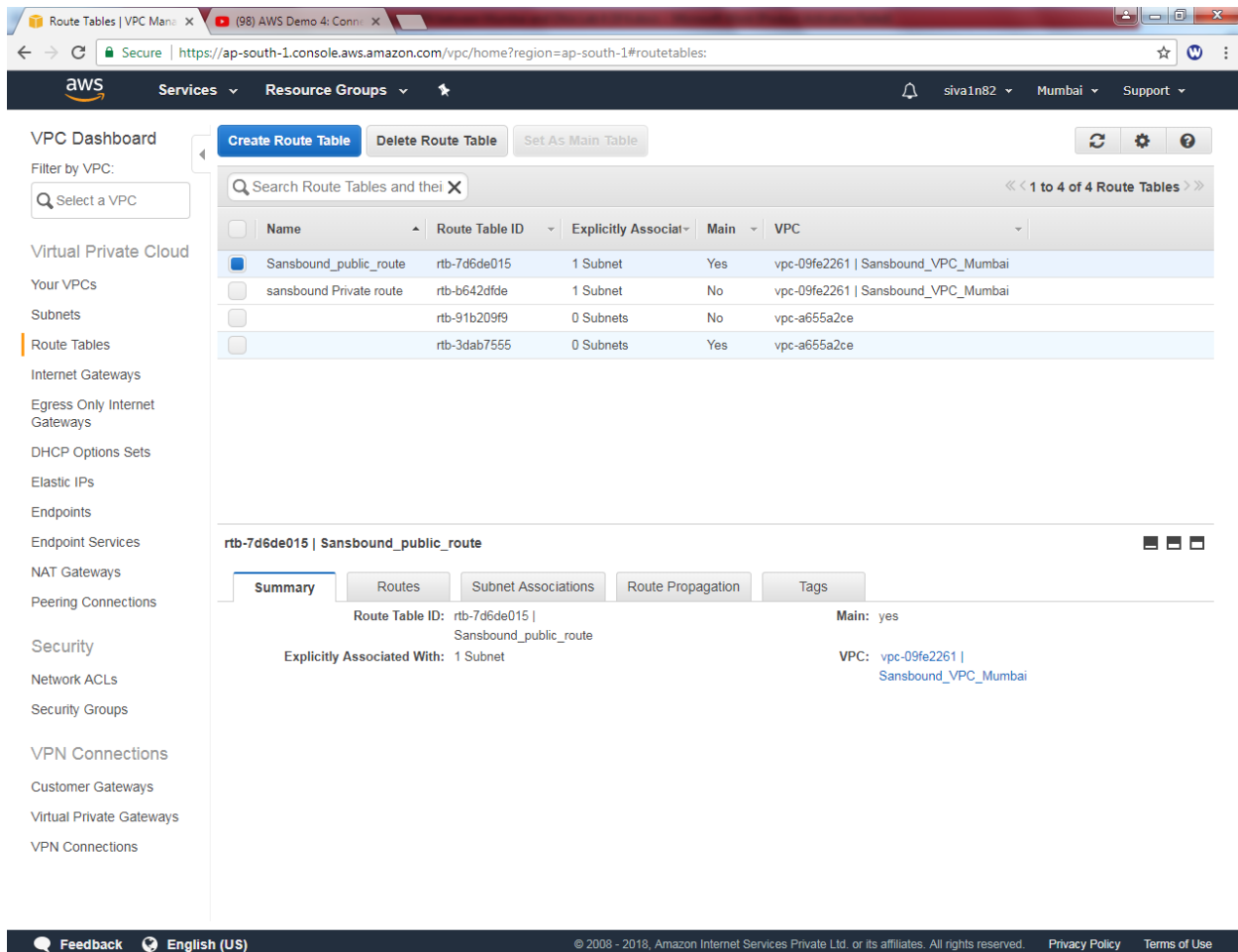


```
root@ip-10-0-2-25:/etc
[root@ip-10-0-2-25 etc]# service network restart
Shutting down interface eth0:                [ OK ]
Shutting down loopback interface:            [ OK ]
Bringing up loopback interface:              [ OK ]
Bringing up interface eth0:
Determining IP information for eth0... done.

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 VPC console interface. The left sidebar contains a navigation menu with categories like 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 first row, "Sansbound_public_route" with ID "rtb-7d6de015", is selected and highlighted. Below the table, the details for this route table are shown, including its ID, name, and the VPC it is associated with.

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

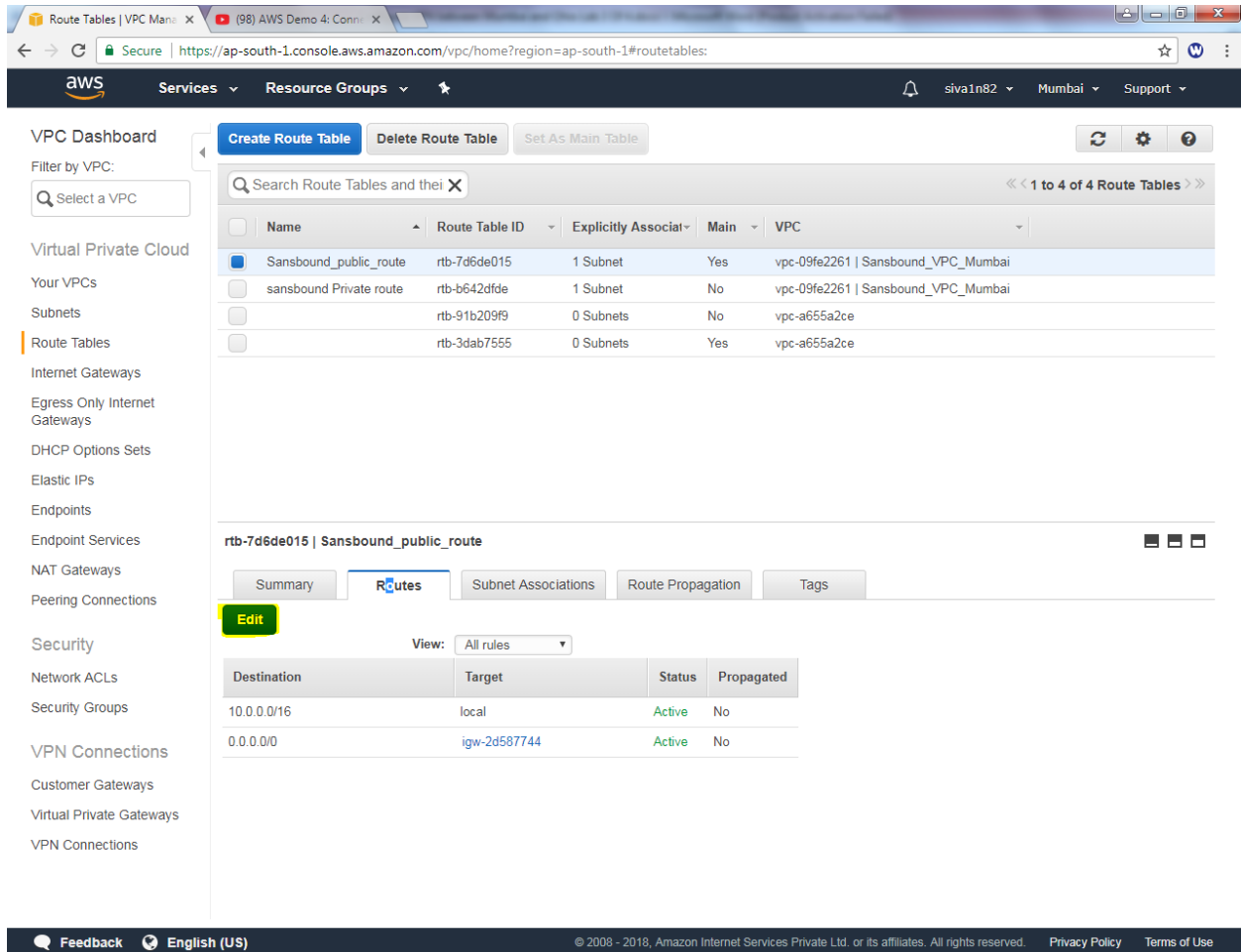
Route Table ID: rtb-7d6de015 | Sansbound_public_route

Main: yes

Explicitly Associated With: 1 Subnet

VPC: vpc-09fe2261 | Sansbound_VPC_Mumbai

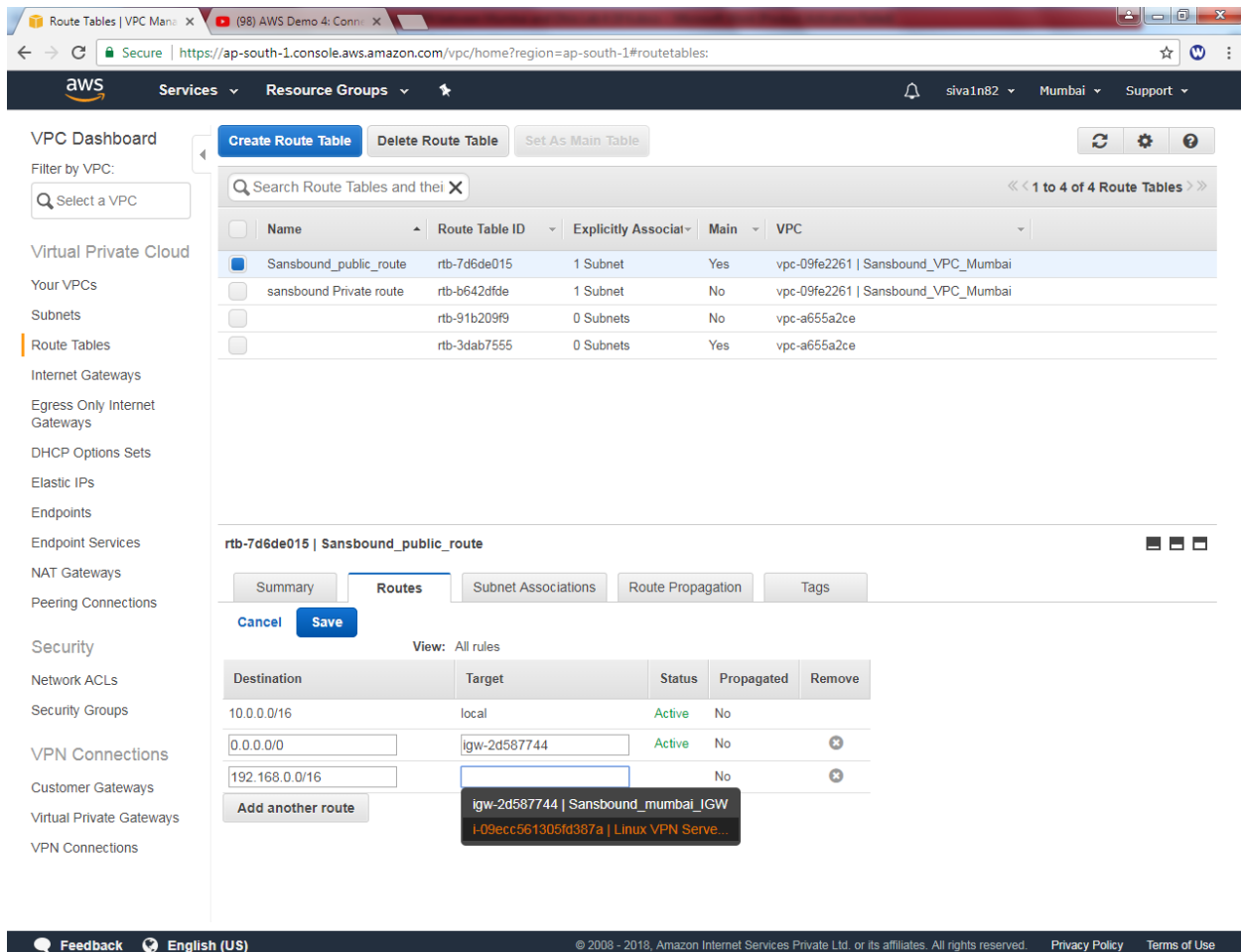
Click “Edit



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. The 'Sansbound_public_route' is selected, and the 'Routes' tab is active, showing a table of routes.

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.



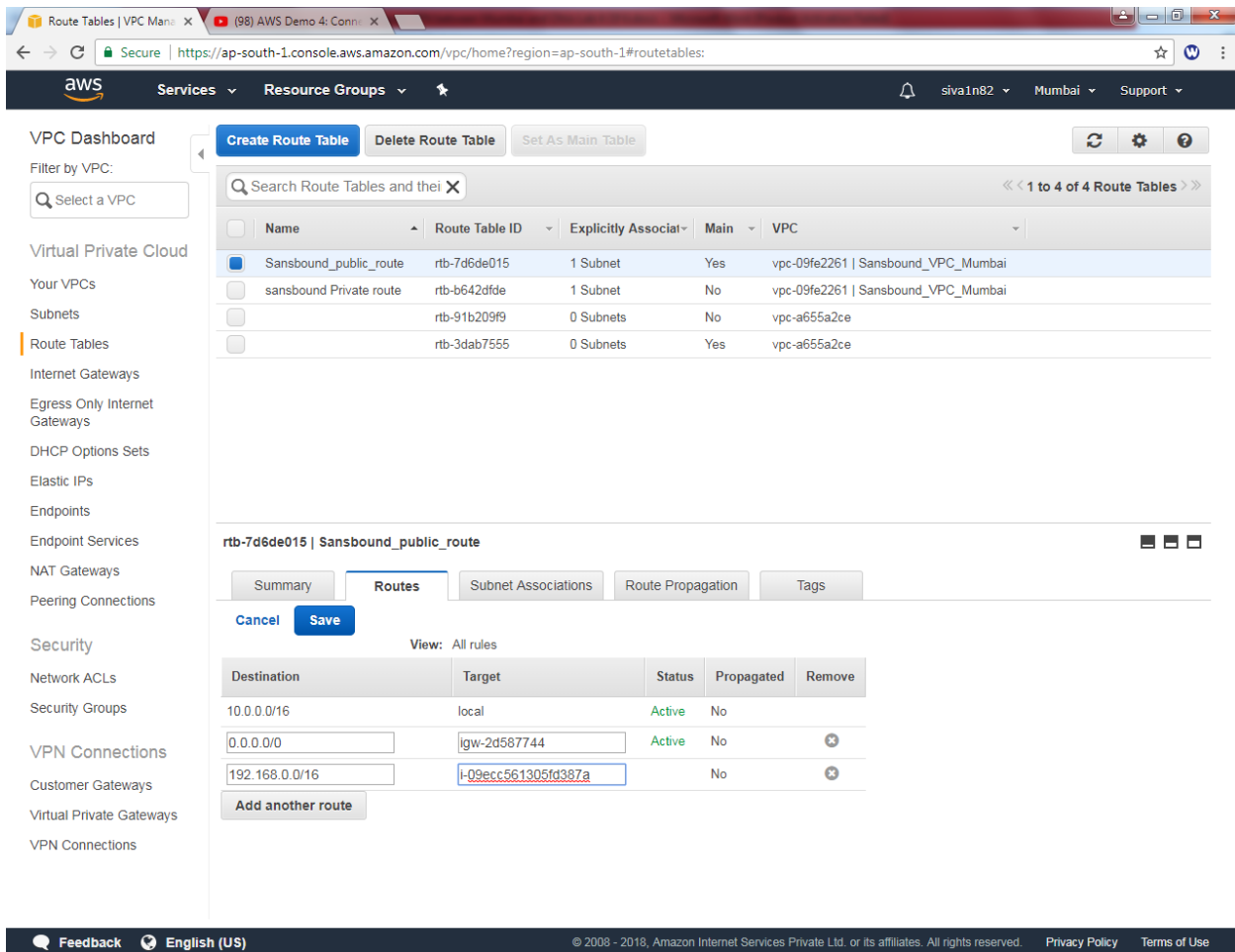
The screenshot shows the AWS Management Console interface for the VPC Dashboard. The left sidebar contains a navigation menu with categories like Virtual Private Cloud, Security, and VPN Connections. The main content area displays a table of route tables. The 'Sansbound_public_route' is selected, and the 'Routes' tab is active. The routes table shows three entries: 'Sansbound_public_route', 'sansbound Private route', and an unnamed route. A tooltip is visible over the 'igw-2d587744 | Sansbound_mumbai_IGW' target, showing its full ID and name.

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

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	

Tooltip for igw-2d587744 | Sansbound_mumbai_IGW:
i-09ecc561305fd387a | Linux VPN Serve...

Click “save”.

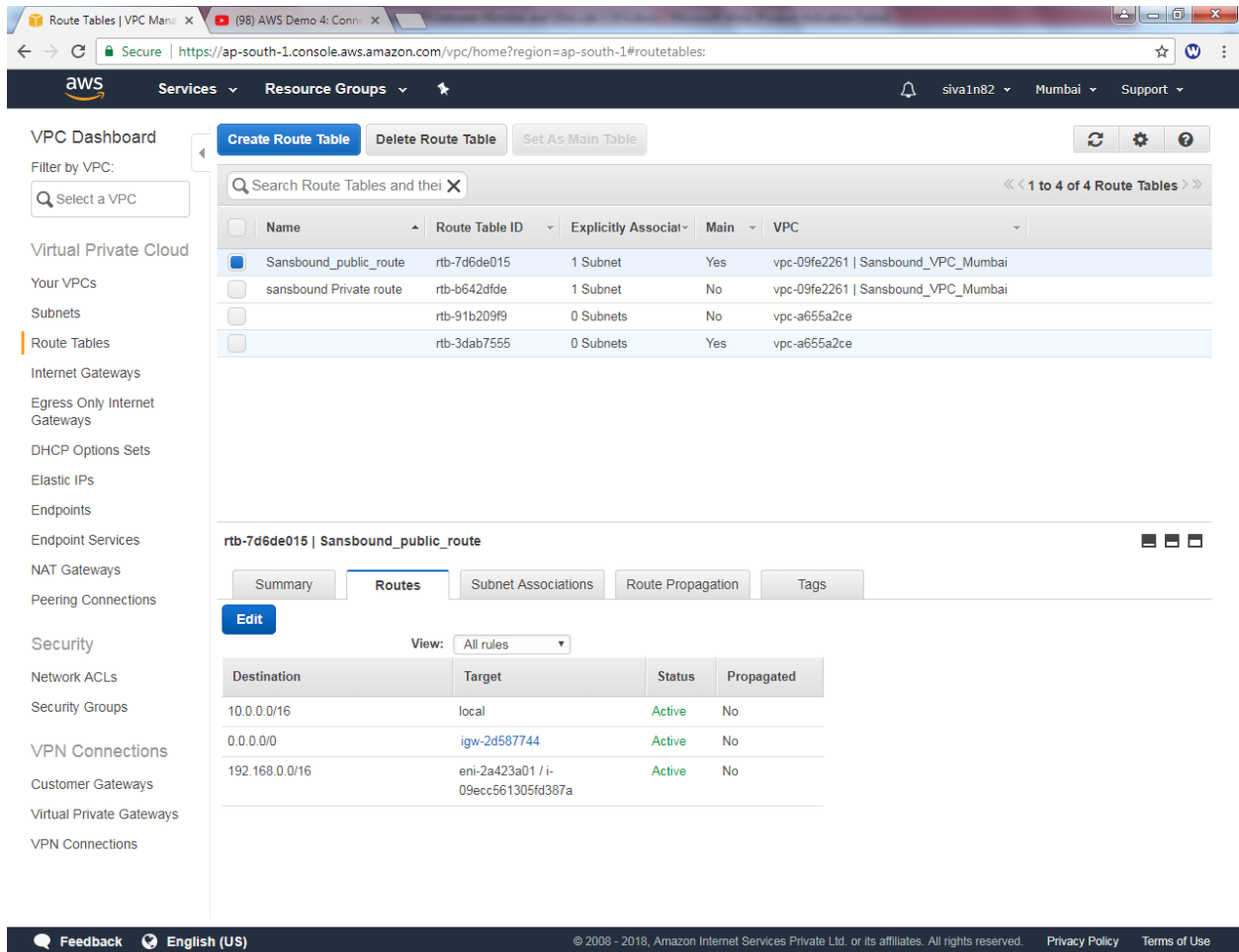


The screenshot shows the AWS Management Console interface for the VPC Dashboard. The left sidebar contains navigation links for VPC Dashboard, Virtual Private Cloud, Your VPCs, Subnets, Route Tables, Internet Gateways, Egress Only Internet Gateways, DHCP Options Sets, Elastic IPs, Endpoints, Endpoint Services, NAT Gateways, Peering Connections, Security, Network ACLs, Security Groups, VPN Connections, Customer Gateways, Virtual Private Gateways, and VPN Connections. The main content area displays the 'Sansbound_public_route' route table (rtb-7d6de015) with a table of routes. The 'Routes' tab is active, showing a list of routes with columns for Destination, Target, Status, Propagated, and Remove. The routes are configured as follows:

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	i-09ecc561305fd387a	No	No	✕

At the bottom of the route table configuration, there is an 'Add another route' button. The top of the console shows the 'Create Route Table', 'Delete Route Table', and 'Set As Main Table' buttons. The bottom of the console shows the 'Feedback' and 'English (US)' links, along with the copyright notice: © 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 Management Console interface for the 'Route Tables' section. 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 selected route table, 'rtb-7d6de015 | Sansbound_public_route', is shown in detail below the list. The 'Routes' tab is active, displaying a table of routes with columns for Destination, Target, Status, and Propagated.

Route Tables List:

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

Route Table Details: rtb-7d6de015 | Sansbound_public_route

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

Edit

View: All rules

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

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