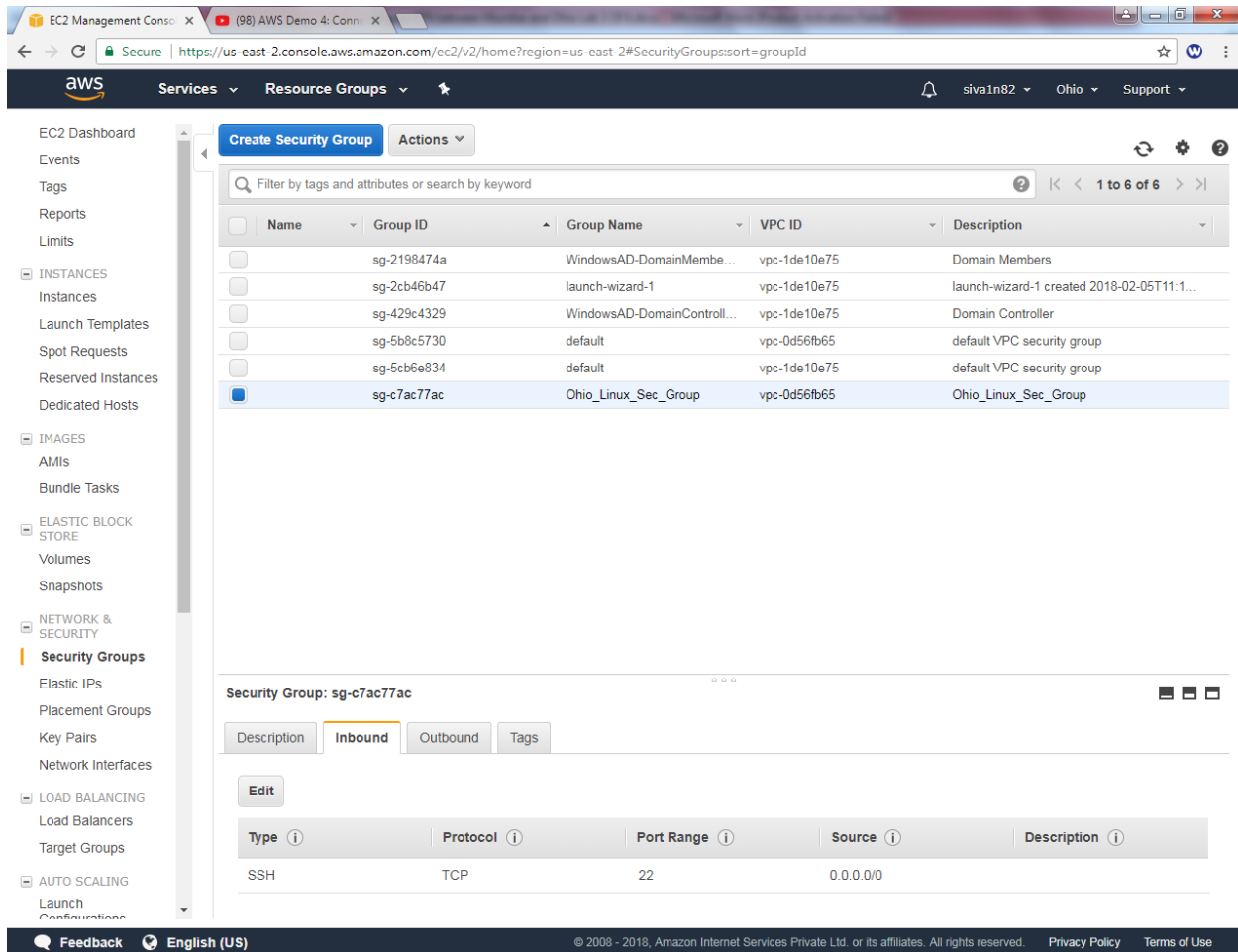


Configure VPN between Mumbai and Ohio Lab 4 of 4

Go to Ohio region, to view the public ip for the network interface (18.218.11.25)

Select security group we need to allow all traffic for 10.0.0.0/16 subnet.



The screenshot displays the AWS Management Console interface for the Ohio region. The left-hand navigation pane shows the 'Security Groups' link under the 'NETWORK & SECURITY' section. The main content area shows a list of security groups. The 'Ohio_Linux_Sec_Group' (sg-c7ac77ac) is selected. Below the list, the 'Inbound' tab is active, showing a table of inbound rules. The table has columns for Type, Protocol, Port Range, Source, and Description. A single rule is listed: SSH (Type), TCP (Protocol), 22 (Port Range), 0.0.0.0/0 (Source), and an empty Description.

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	0.0.0.0/0	

Click Add rule and allow All traffic type source as 10.0.0.0/16 subnet then click “Save”.

Edit inbound rules

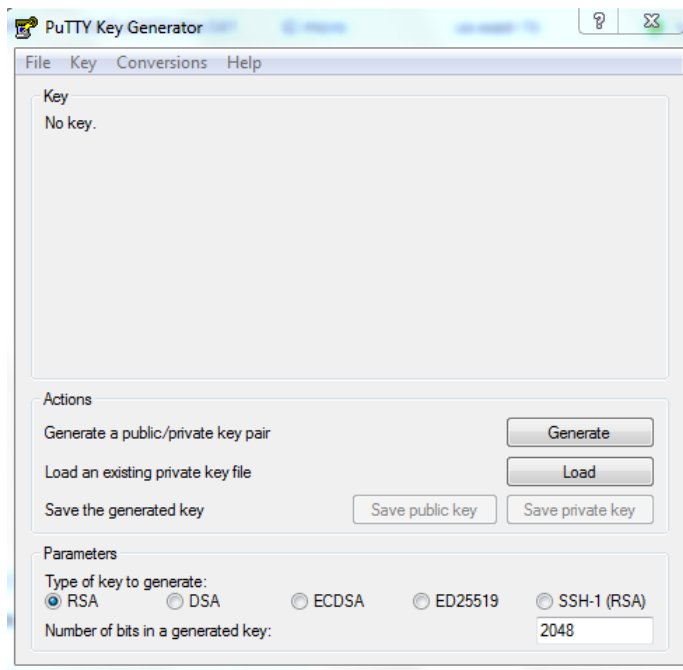
Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
All traffic	All	0 - 65535	Custom 10.0.0.0/16	e.g. SSH for Admin Desktop

Add Rule

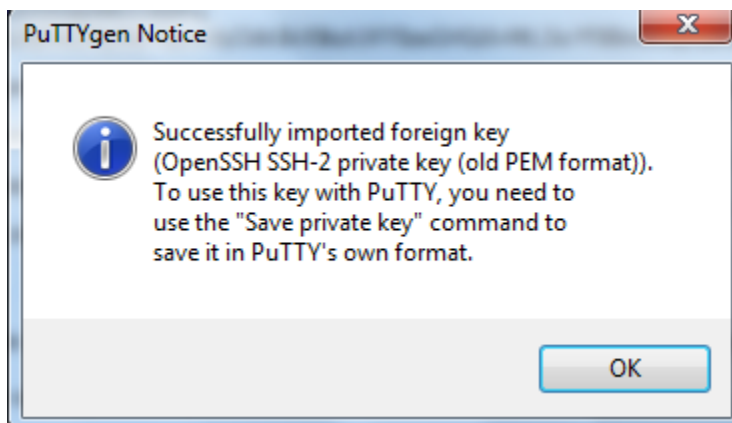
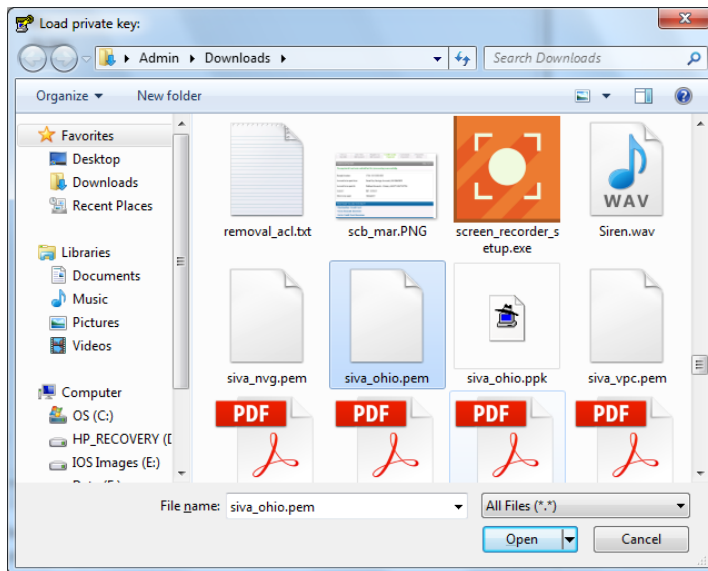
NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

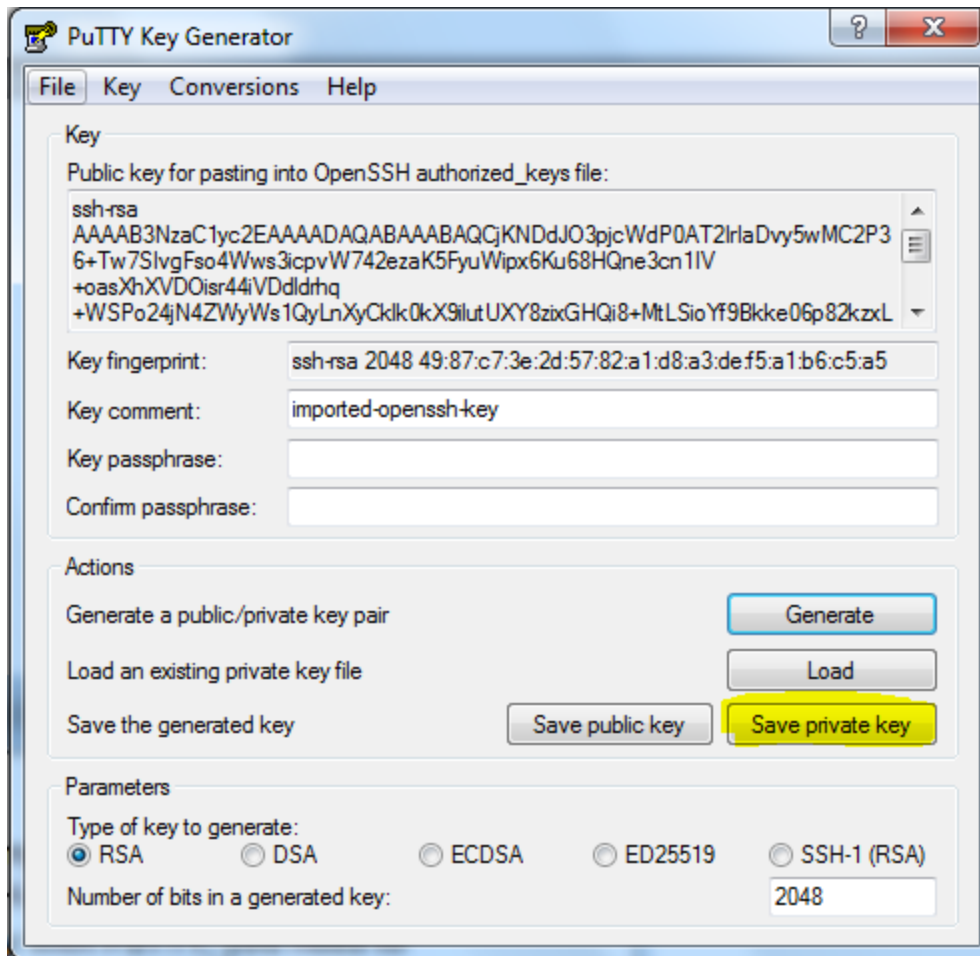
Cancel Save

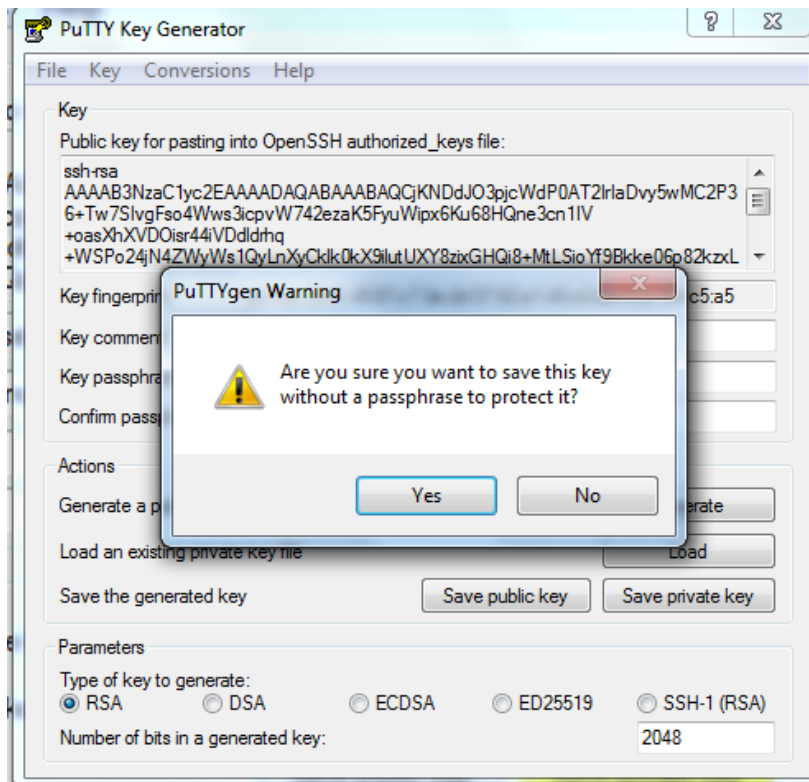
Go to Putty key gen installed in your local machine.



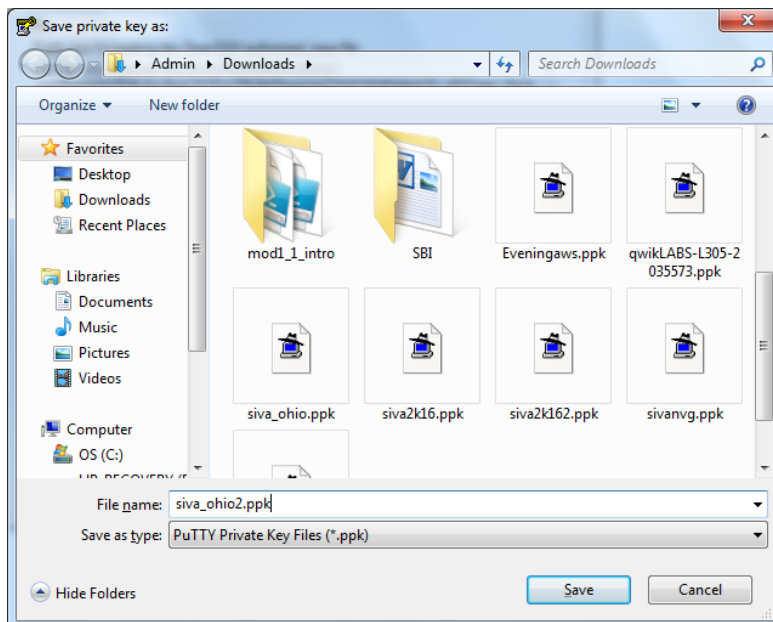
Locate the file and click “Open”.

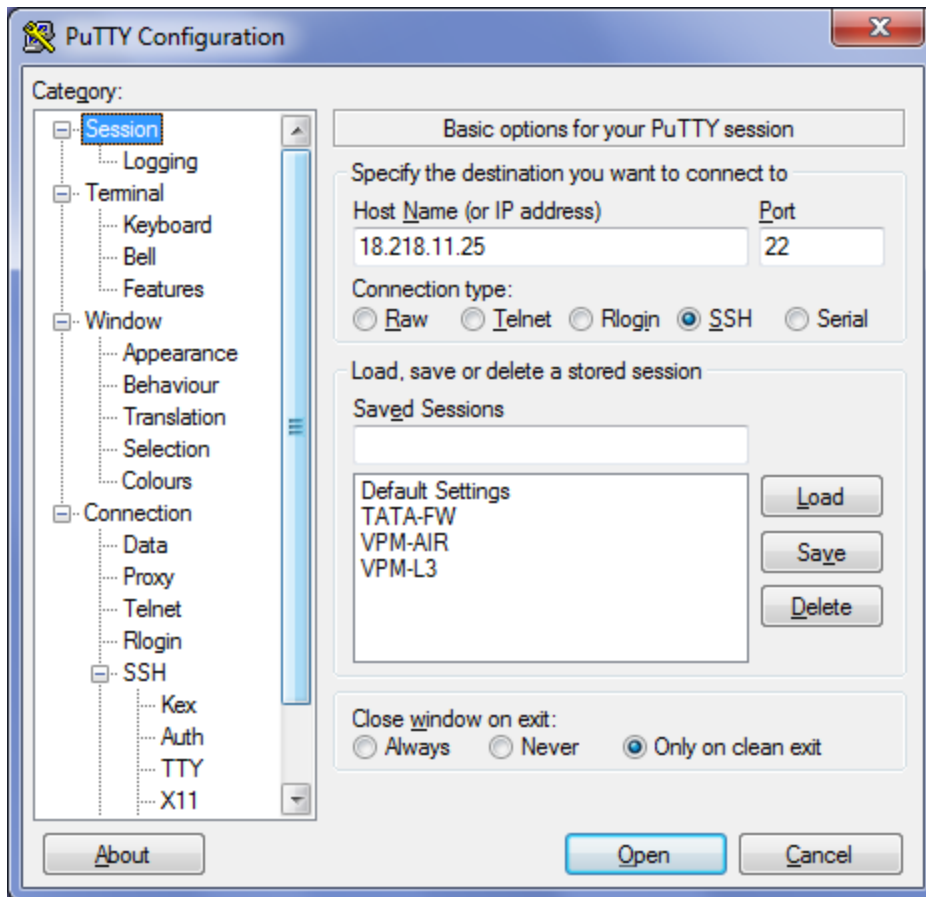




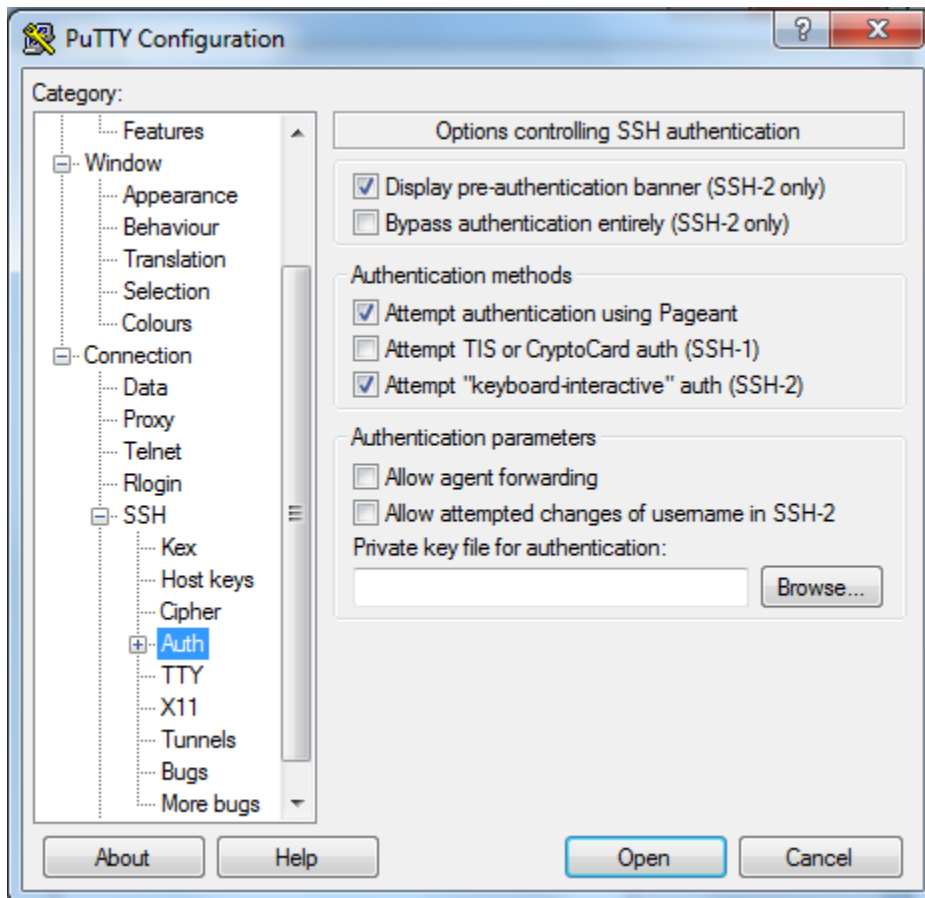


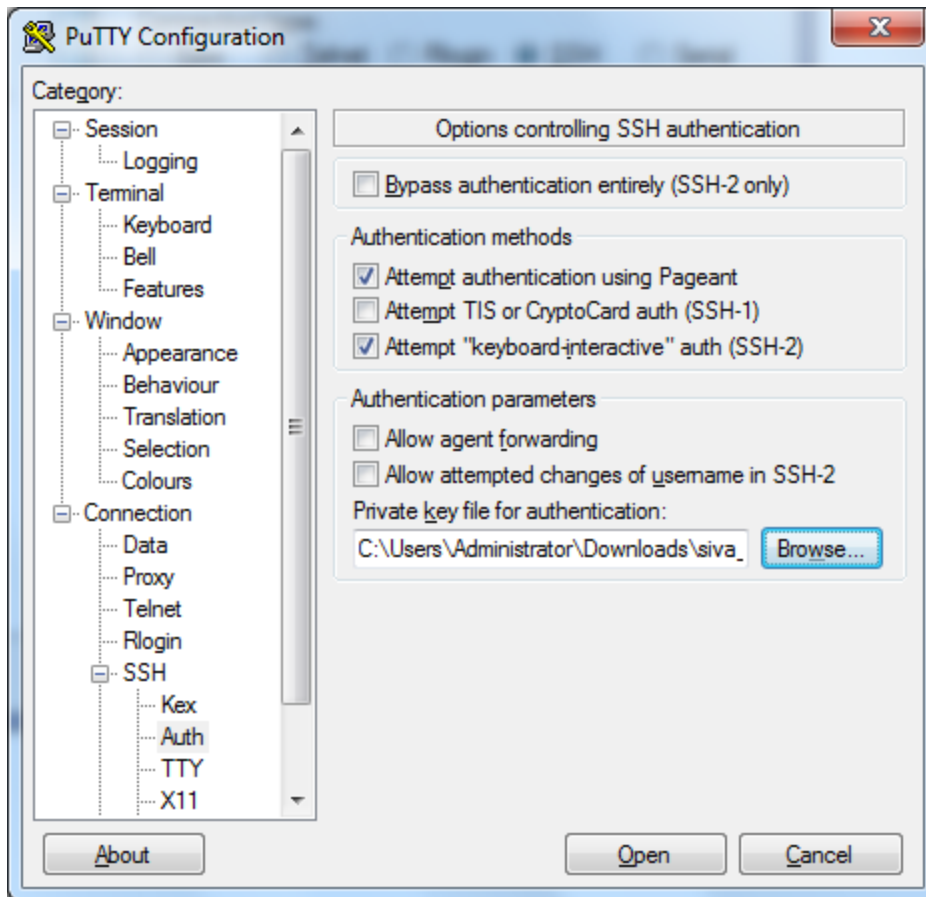
Locate the file to save.

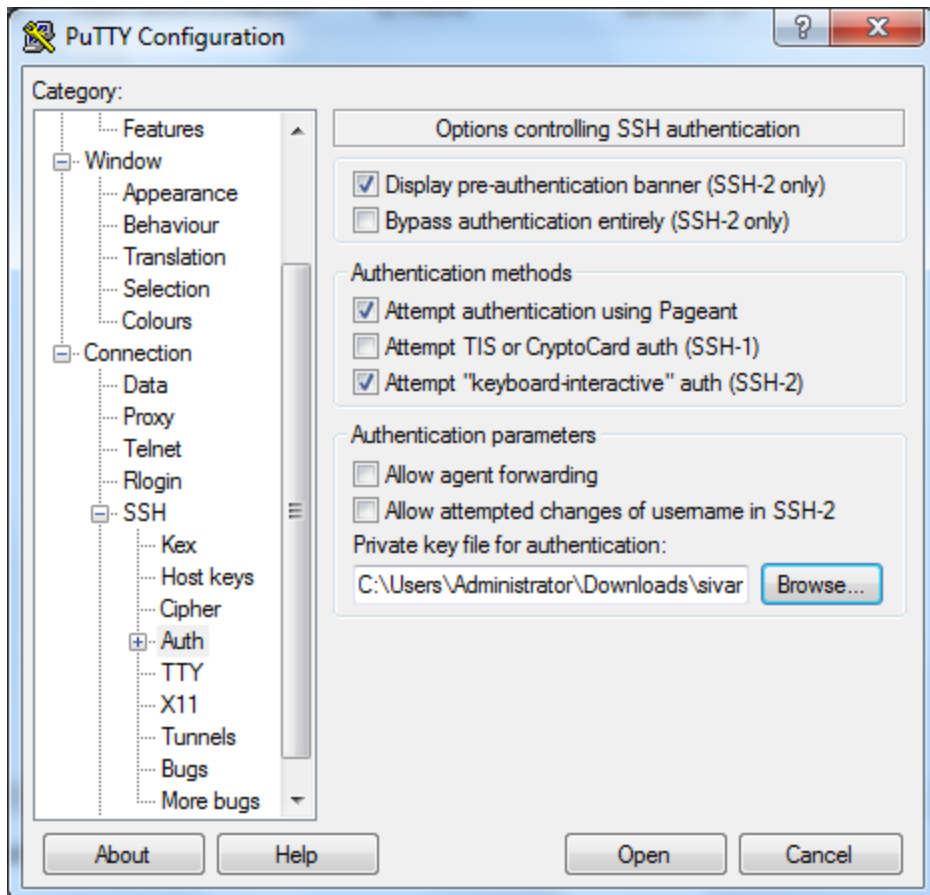


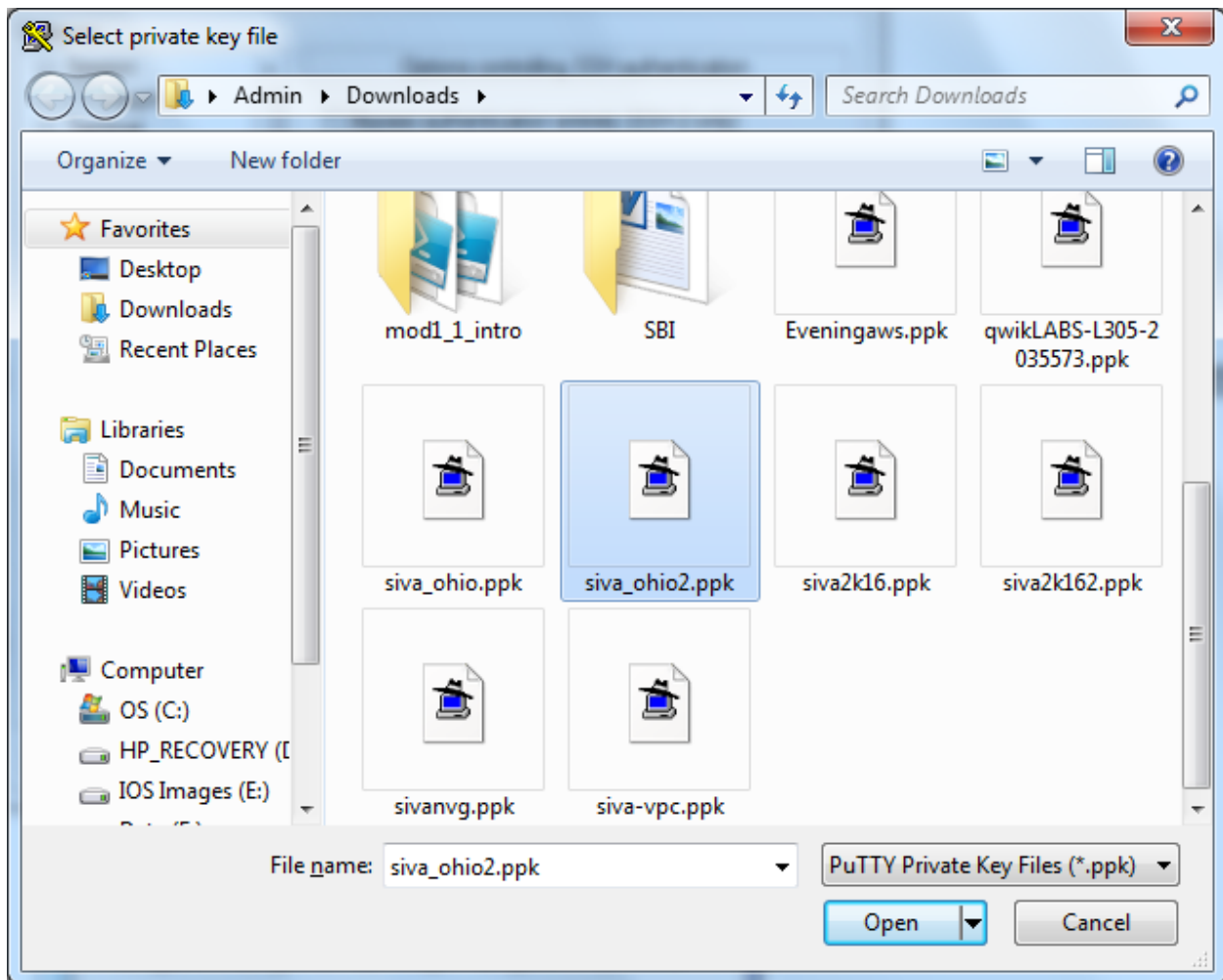


Click Browse and locate the *.ppk file.

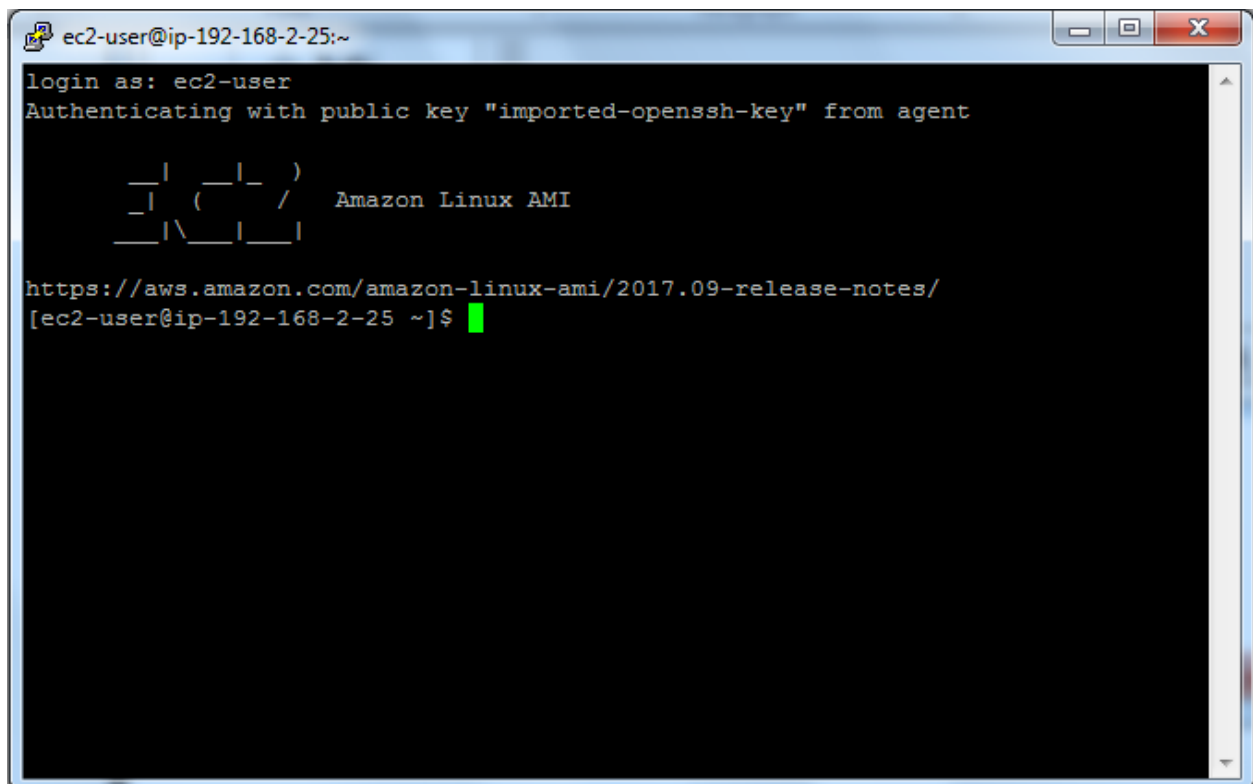
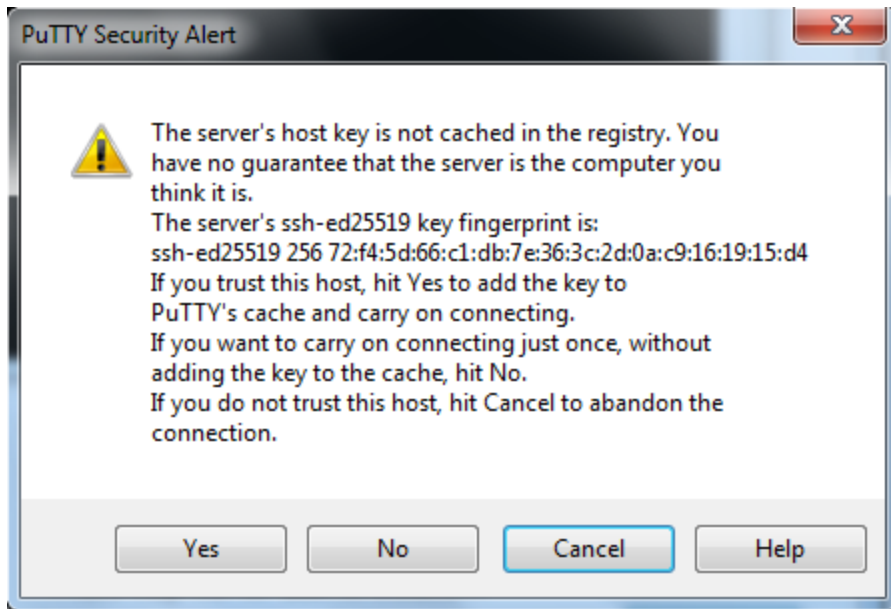






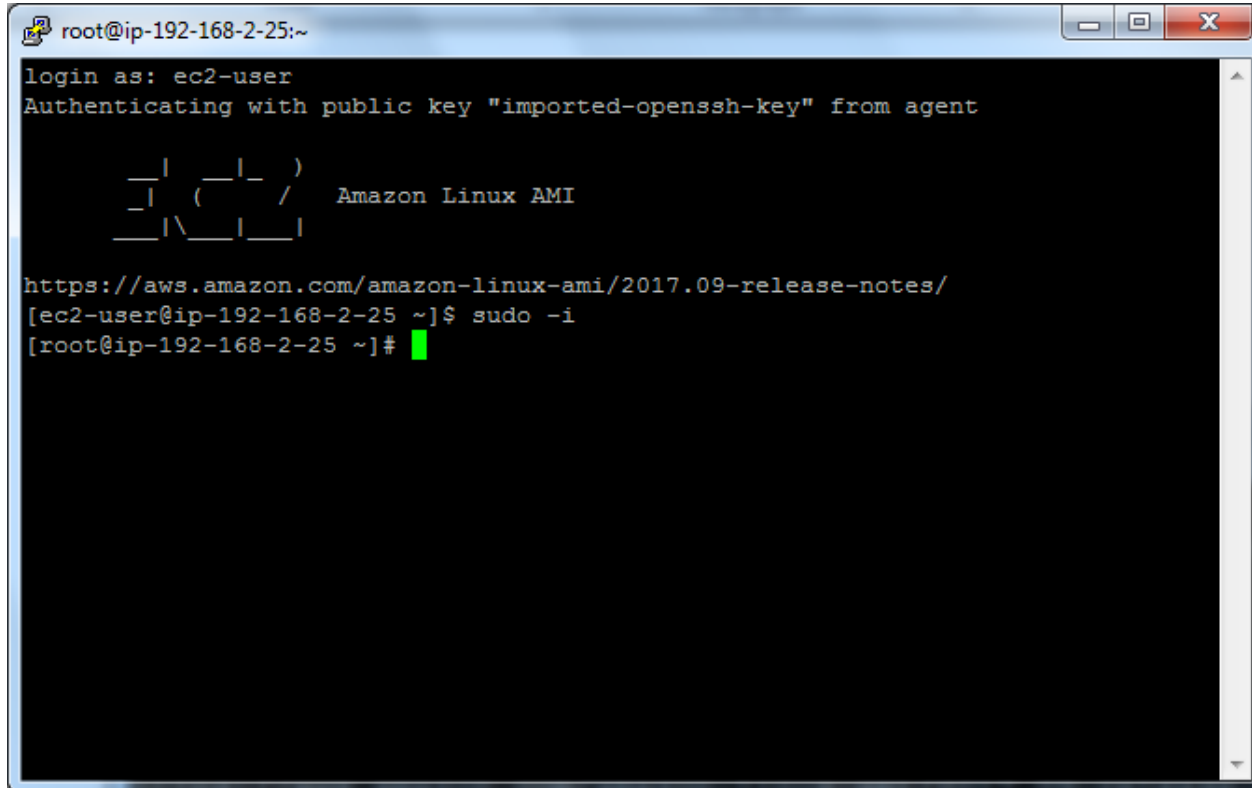


Select the file and Click “Open”.



Type

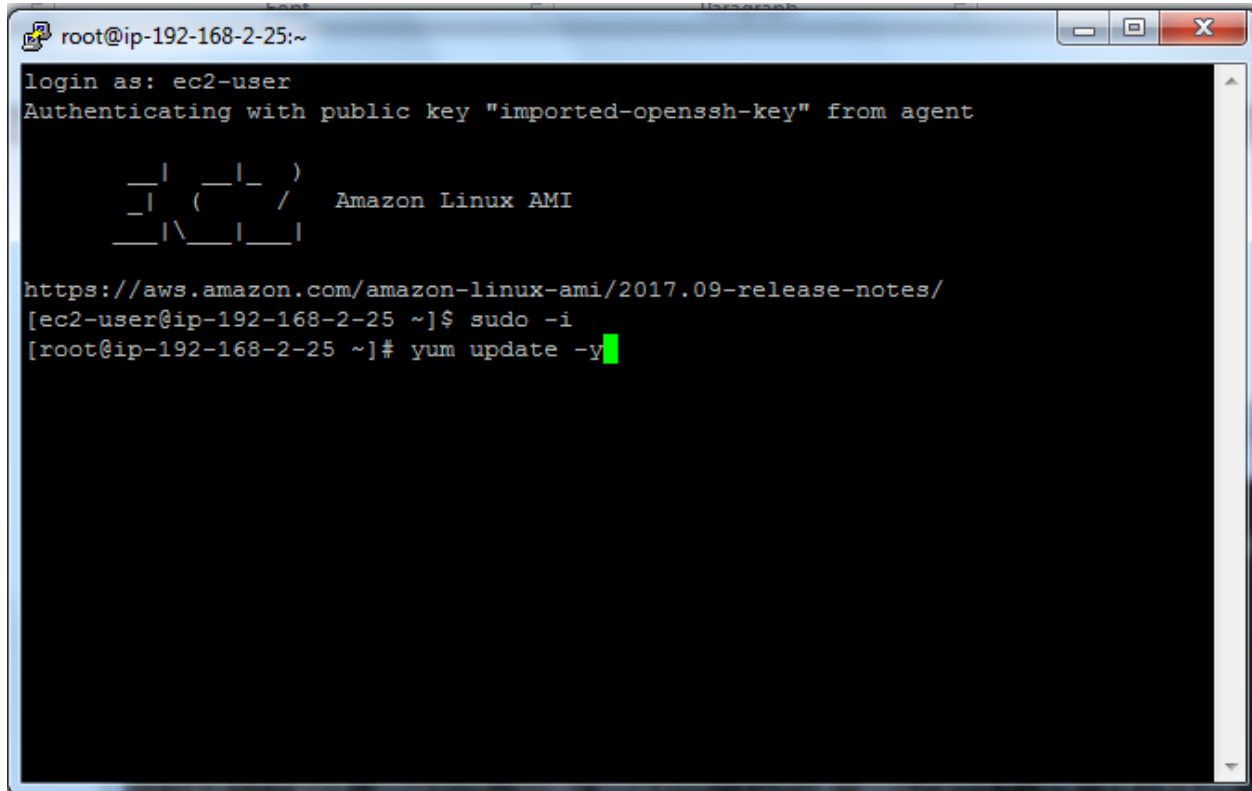
Sudo -i



```
root@ip-192-168-2-25:~  
login as: ec2-user  
Authenticating with public key "imported-openssh-key" from agent  
  
  _ | _ | _ )  
  _ | ( _ | _ /   Amazon Linux AMI  
  _ | \ _ | _ |  
  
https://aws.amazon.com/amazon-linux-ami/2017.09-release-notes/  
[ec2-user@ip-192-168-2-25 ~]$ sudo -i  
[root@ip-192-168-2-25 ~]#
```

Type

Yum update -y

A terminal window titled 'root@ip-192-168-2-25:~' with standard window controls. The terminal shows an SSH login for 'ec2-user' using a public key. It displays the Amazon Linux AMI logo and a URL to the release notes. The user then runs 'sudo -i' to become root and 'yum update -y' to update the system. The cursor is at the end of the command.

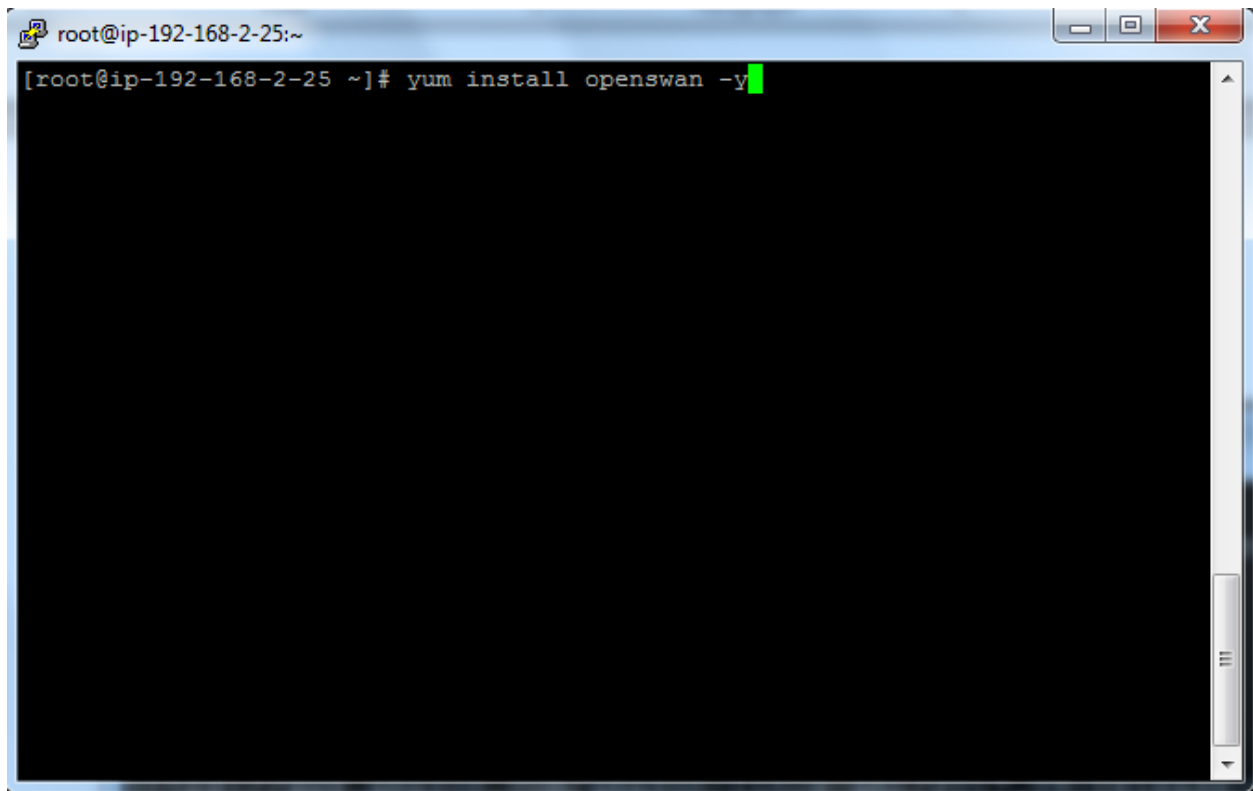
```
root@ip-192-168-2-25:~
login as: ec2-user
Authenticating with public key "imported-openssh-key" from agent

  _ | _ | _ )
  _ | ( _ | /   Amazon Linux AMI
  __| \__|__|

https://aws.amazon.com/amazon-linux-ami/2017.09-release-notes/
[ec2-user@ip-192-168-2-25 ~]$ sudo -i
[root@ip-192-168-2-25 ~]# yum update -y
```

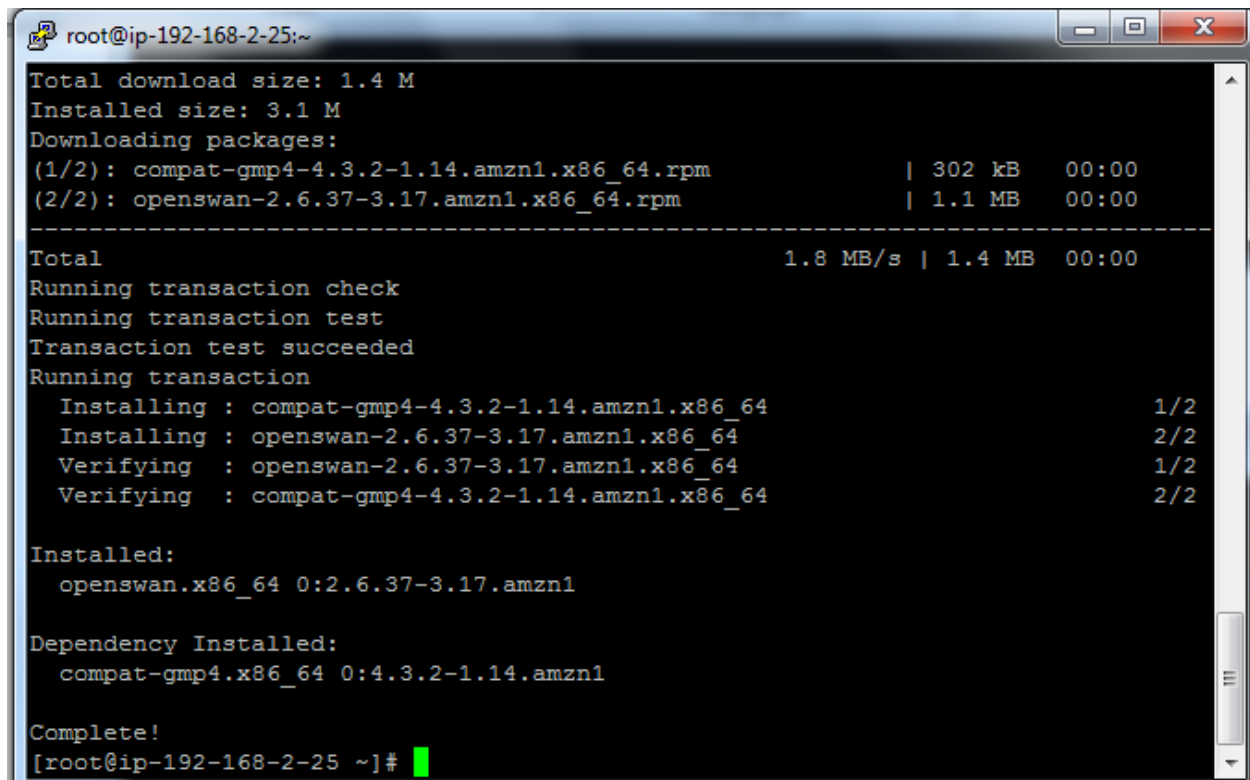
Type

Yum install openswan -y

A terminal window with a blue title bar. The title bar contains a small icon on the left and three window control buttons (minimize, maximize, close) on the right. The terminal text shows the prompt 'root@ip-192-168-2-25:~' followed by the command '[root@ip-192-168-2-25 ~]# yum install openswan -y' with a green cursor at the end of the command.

```
root@ip-192-168-2-25:~  
[root@ip-192-168-2-25 ~]# yum install openswan -y
```

Open swan has been successfully installed.



```
root@ip-192-168-2-25:~  
Total download size: 1.4 M  
Installed size: 3.1 M  
Downloading packages:  
(1/2): compat-gmp4-4.3.2-1.14.amzn1.x86_64.rpm | 302 kB 00:00  
(2/2): openswan-2.6.37-3.17.amzn1.x86_64.rpm | 1.1 MB 00:00  
-----  
Total 1.8 MB/s | 1.4 MB 00:00  
Running transaction check  
Running transaction test  
Transaction test succeeded  
Running transaction  
  Installing : compat-gmp4-4.3.2-1.14.amzn1.x86_64 1/2  
  Installing : openswan-2.6.37-3.17.amzn1.x86_64 2/2  
  Verifying   : openswan-2.6.37-3.17.amzn1.x86_64 1/2  
  Verifying   : compat-gmp4-4.3.2-1.14.amzn1.x86_64 2/2  
  
Installed:  
  openswan.x86_64 0:2.6.37-3.17.amzn1  
  
Dependency Installed:  
  compat-gmp4.x86_64 0:4.3.2-1.14.amzn1  
  
Complete!  
[root@ip-192-168-2-25 ~]#
```

Vi ipsec.conf

```
root@ip-192-168-2-25:/etc
login as: ec2-user
Authenticating with public key "imported-openssh-key"

    _|_|_ )
    _| ( /   Amazon Linux AMI
    __|\\__|__|

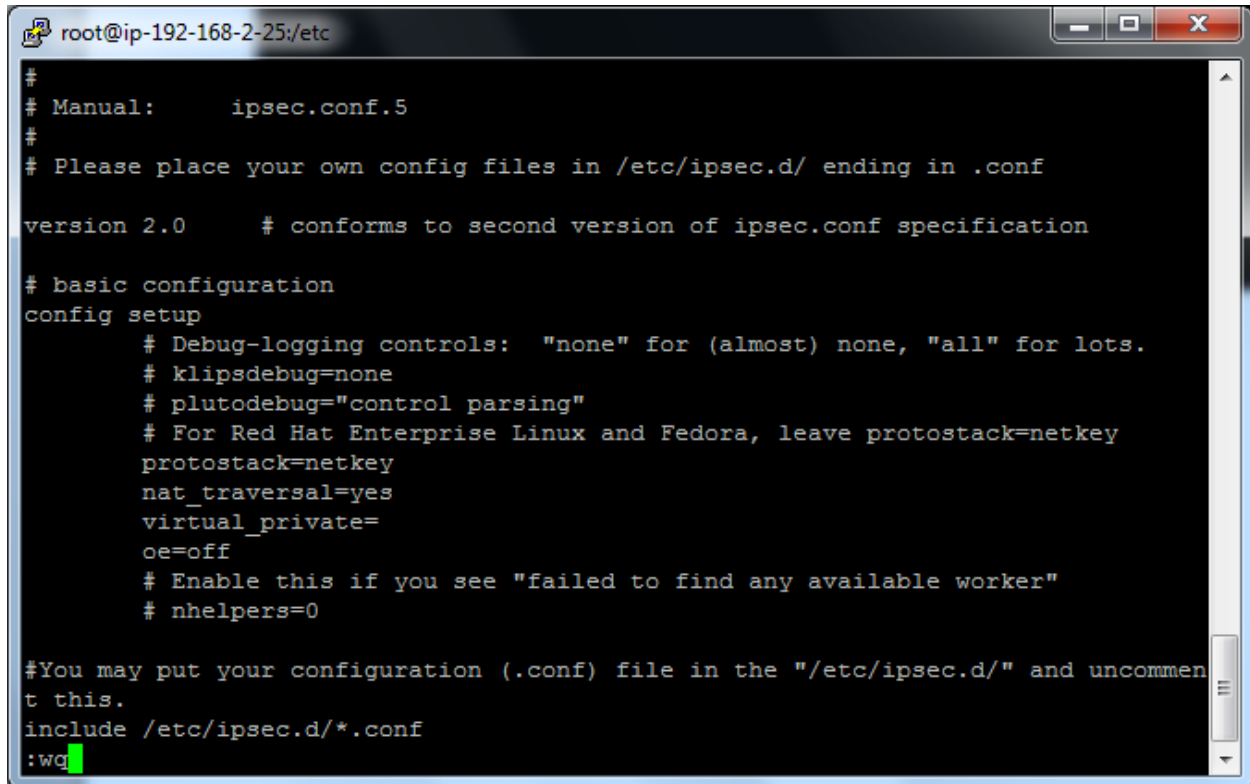
https://aws.amazon.com/amazon-linux-ami/2017.09-release-notes/
1 package(s) needed for security, out of 1 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-192-168-2-25 ~]$ sudo -i
[root@ip-192-168-2-25 ~]# cd /etc
[root@ip-192-168-2-25 etc]# vi ipsec.conf
```


Press insert and remove # from #include /etc/

[illegible]

Press escape and type

:wq

A screenshot of a terminal window with a black background and white text. The window title bar shows 'root@ip-192-168-2-25:/etc'. The terminal content shows the configuration for /etc/ipsec.conf. It includes comments about the manual, configuration file placement, and version 2.0. Under the 'basic configuration' section, there are settings for debug logging, klipsdebug, plutodebug, protostack, nat_traversal, virtual_private, oe, and nhelpers. A comment at the bottom suggests putting configuration files in /etc/ipsec.d/. The cursor is at the end of the line ':wq' on the last line of the visible text.

```
root@ip-192-168-2-25:/etc
#
# Manual:      ipsec.conf.5
#
# Please place your own config files in /etc/ipsec.d/ ending in .conf

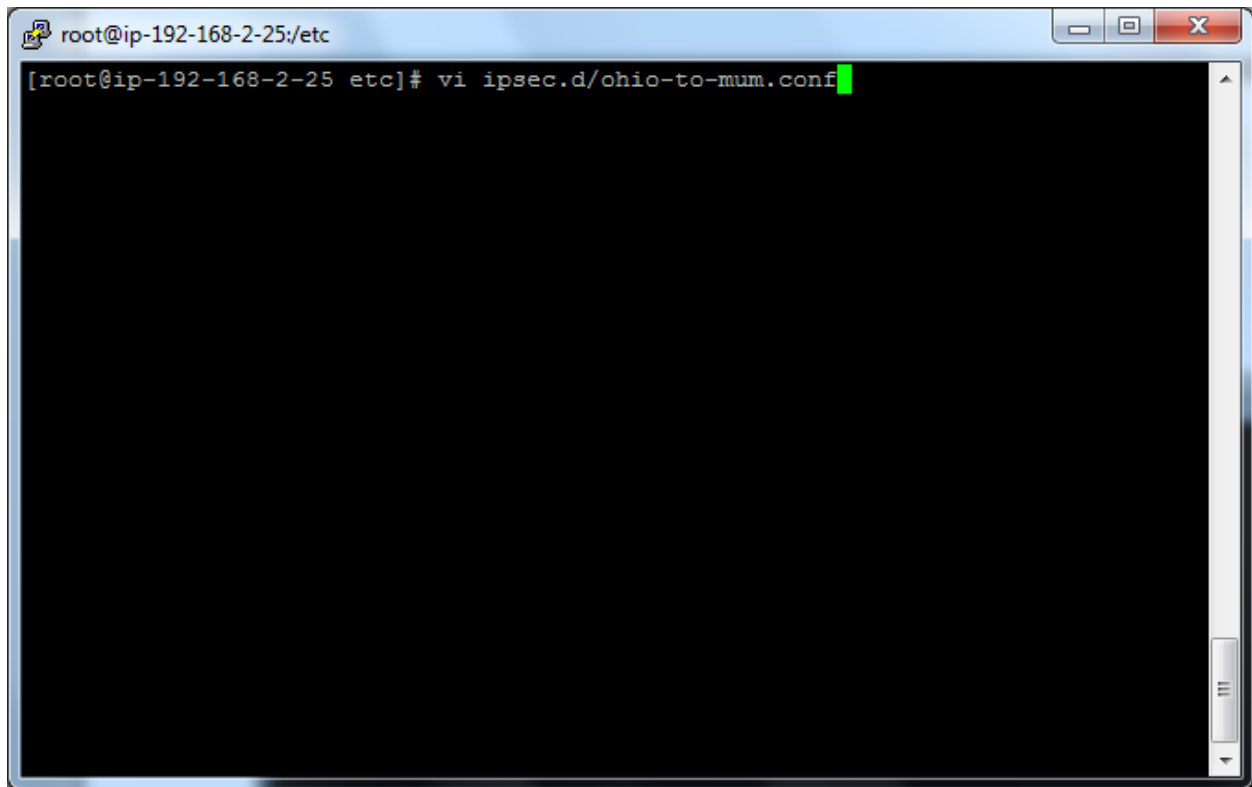
version 2.0      # conforms to second version of ipsec.conf specification

# 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 uncommen
t this.
include /etc/ipsec.d/*.conf
:wq
```

Type

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



Press Insert key and type the command in vi editor

conn ohio-to-mum

type=tunnel

authby=secret

left=defaultroute

leftid=18.218.11.25

leftnexthop=%defaultroute

```
leftsubnet=192.168.0.0/16
```

```
right=13.127.161.231
```

```
rightsubnet=10.0.0.0/16
```

pfs=yes

auto=start

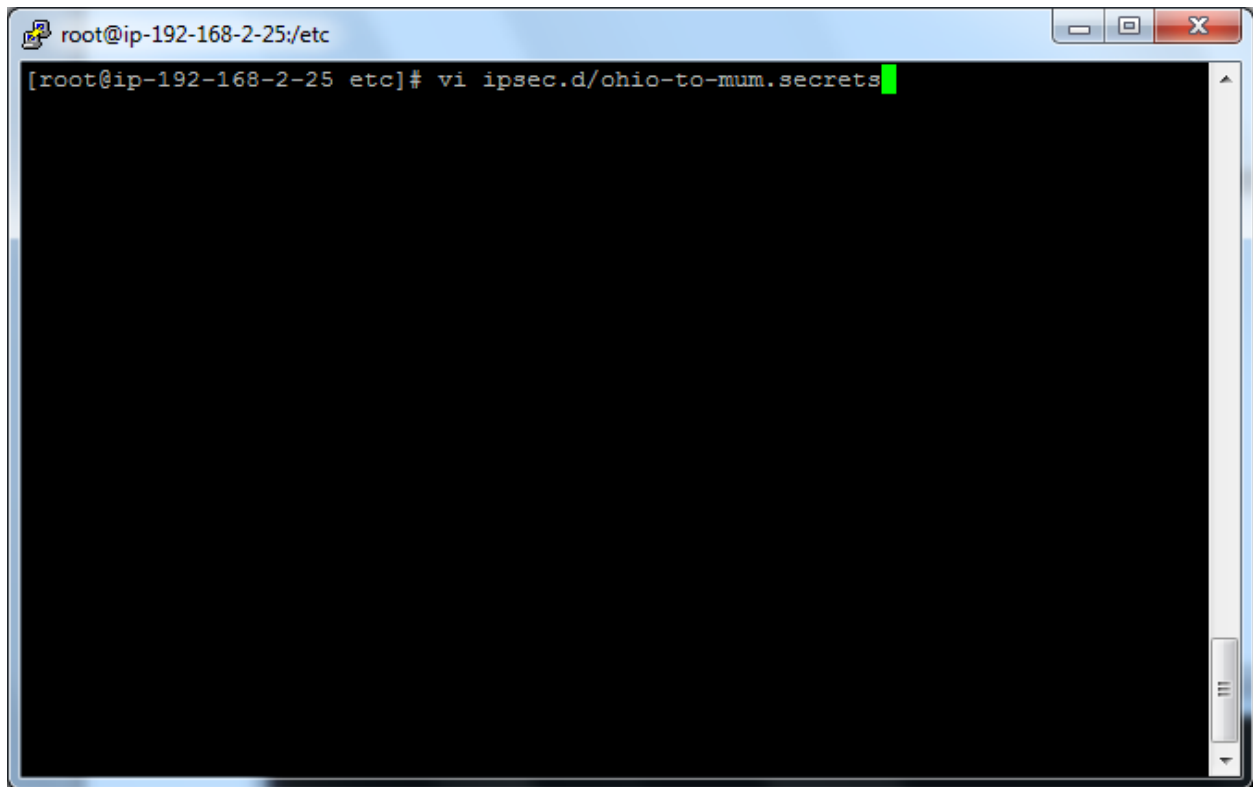
```
root@ip-192-168-2-25:/etc
conn ohio-to-mum
    type=tunnel
    authby=secret
    left=defaultroute
    leftid=18.218.11.25
    leftnexthop=%defaultroute
    leftsubnet=192.168.0.0/16
    right=13.127.161.231
    rightsubnet=10.0.0.0/16
    pfs=yes
    auto=start
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --
```

Press escape and type

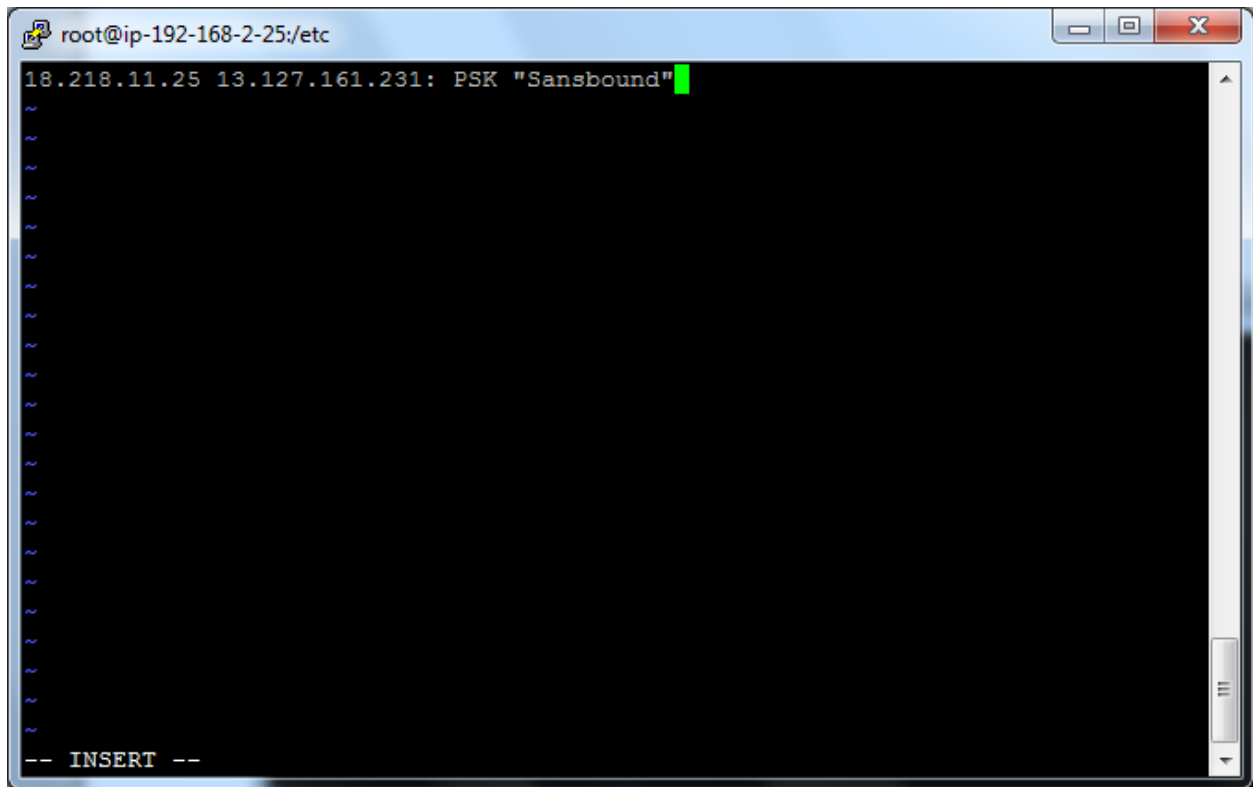
:wq

Type

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

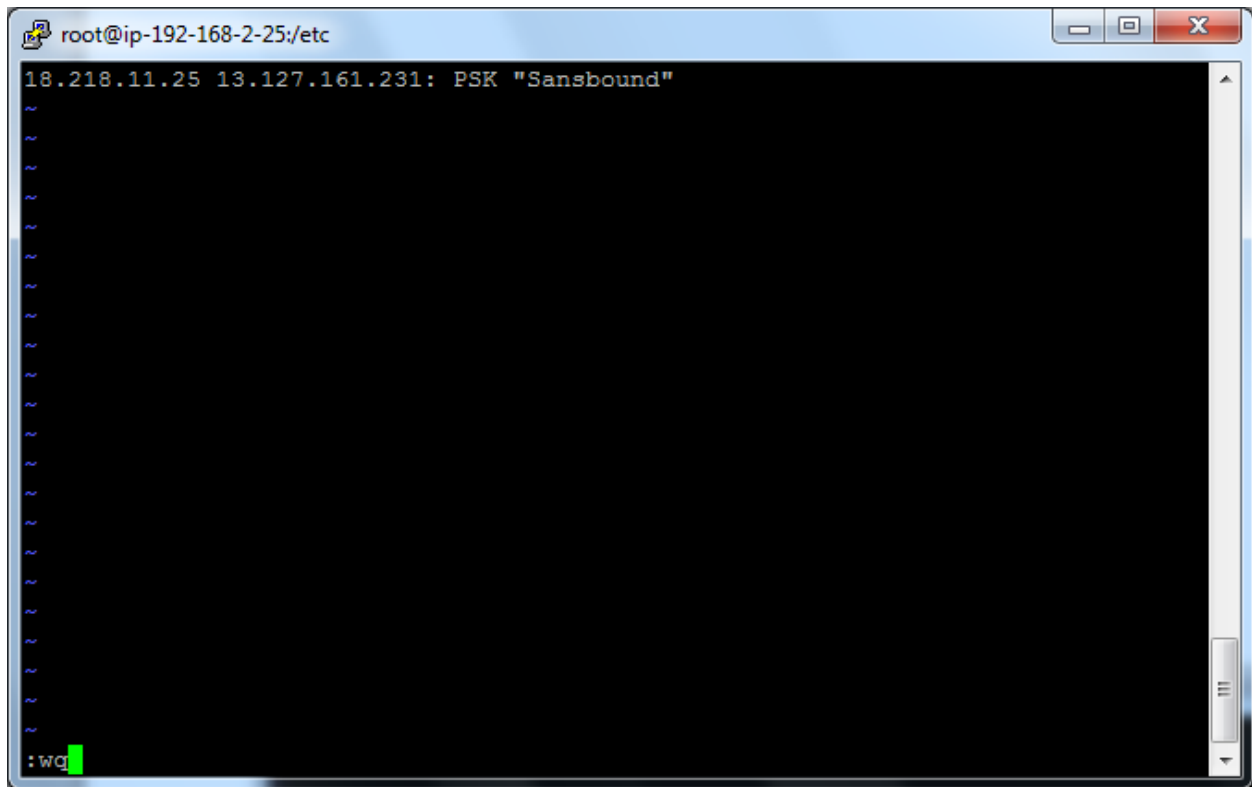


Preshared key is "Sansbound"



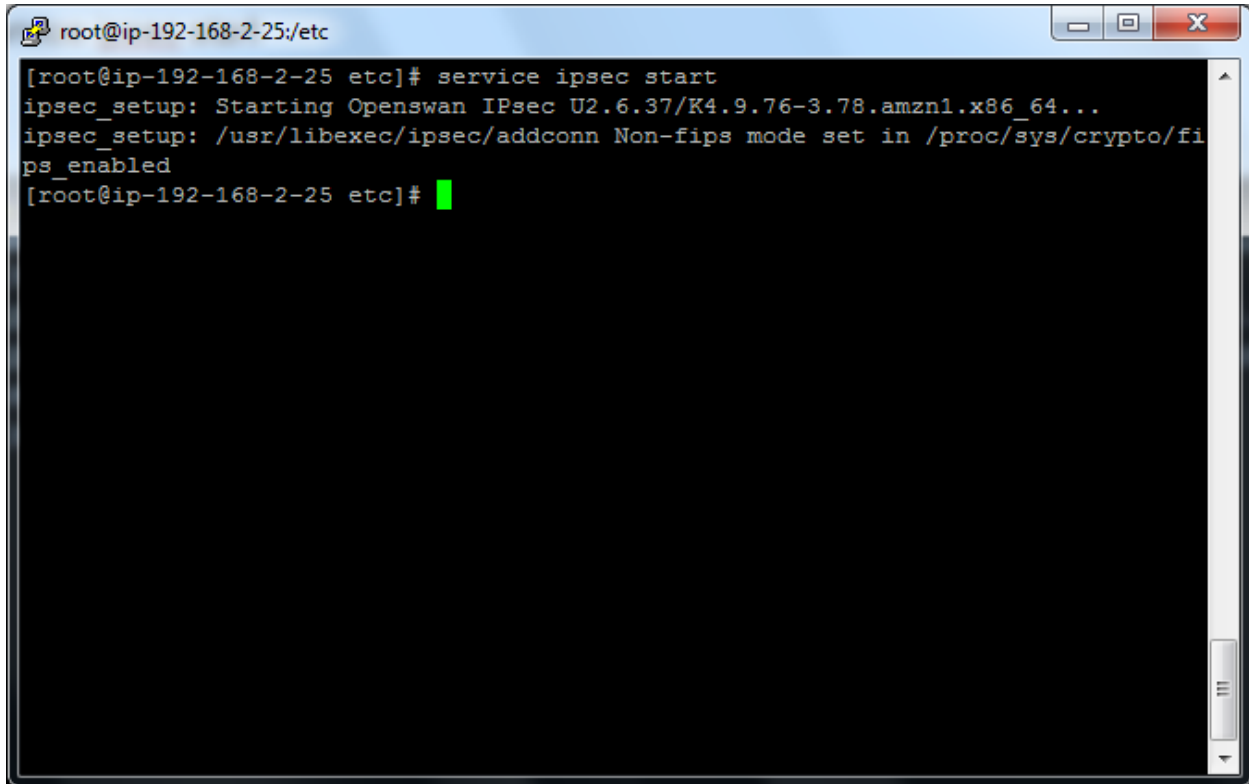
Press escape and type

:wq



Type

service ipsec start

A terminal window with a blue title bar containing the text 'root@ip-192-168-2-25:/etc'. The terminal has a black background with white text. The command 'service ipsec start' has been entered and executed. The output shows 'ipsec_setup: Starting Openswan IPsec U2.6.37/K4.9.76-3.78.amzn1.x86_64...' followed by 'ipsec_setup: /usr/libexec/ipsec/addconn Non-fips mode set in /proc/sys/crypto/fips_enabled'. The prompt '[root@ip-192-168-2-25 etc]#' is followed by a green cursor.

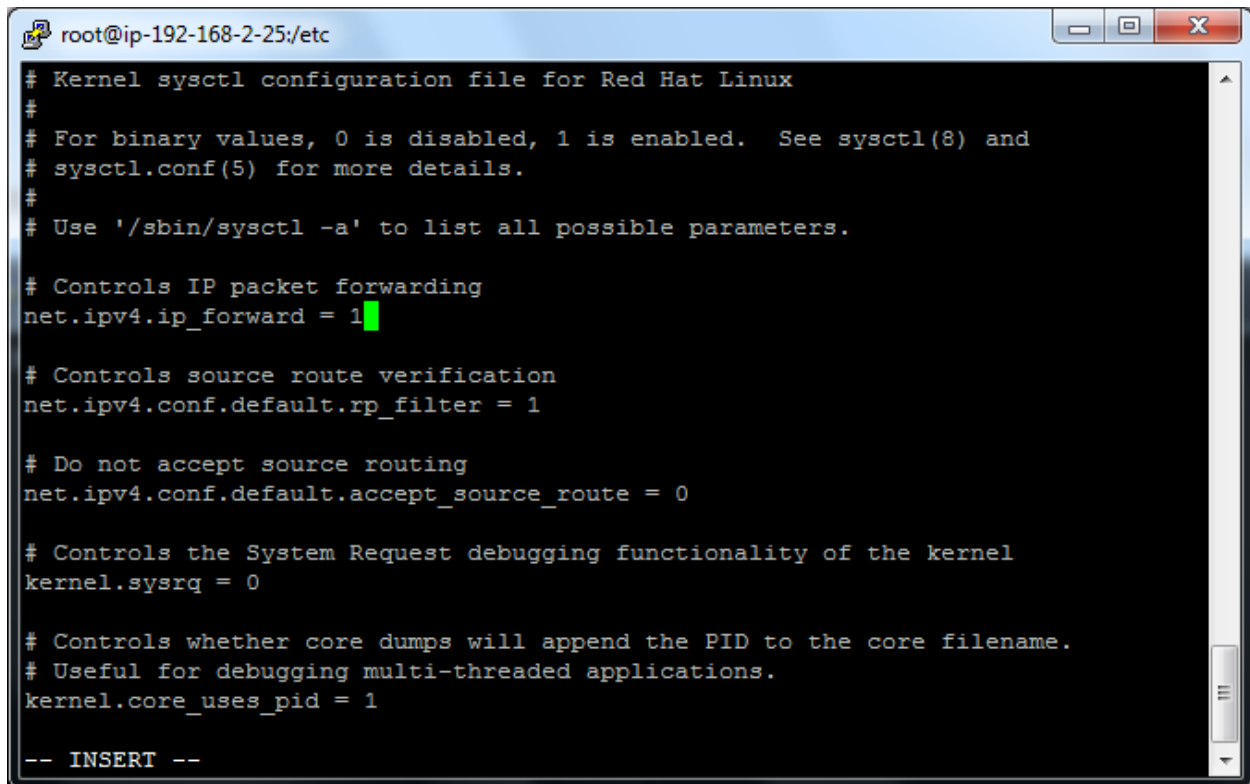
```
root@ip-192-168-2-25:/etc
[root@ip-192-168-2-25 etc]# service ipsec start
ipsec_setup: Starting Openswan IPsec U2.6.37/K4.9.76-3.78.amzn1.x86_64...
ipsec_setup: /usr/libexec/ipsec/addconn Non-fips mode set in /proc/sys/crypto/fips_enabled
[root@ip-192-168-2-25 etc]#
```


Type

Vi sysctl.conf

A screenshot of a terminal window. The title bar at the top reads 'root@ip-192-168-2-25:/etc'. The terminal content shows the command '[root@ip-192-168-2-25 etc]# vi sysctl.conf' with a green cursor at the end of the command. The terminal background is black, and the text is white. The window has standard Linux window controls (minimize, maximize, close) in the top right corner.

Press insert key and rename the net.ipv4.ip_forward = 1.



```
root@ip-192-168-2-25:/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

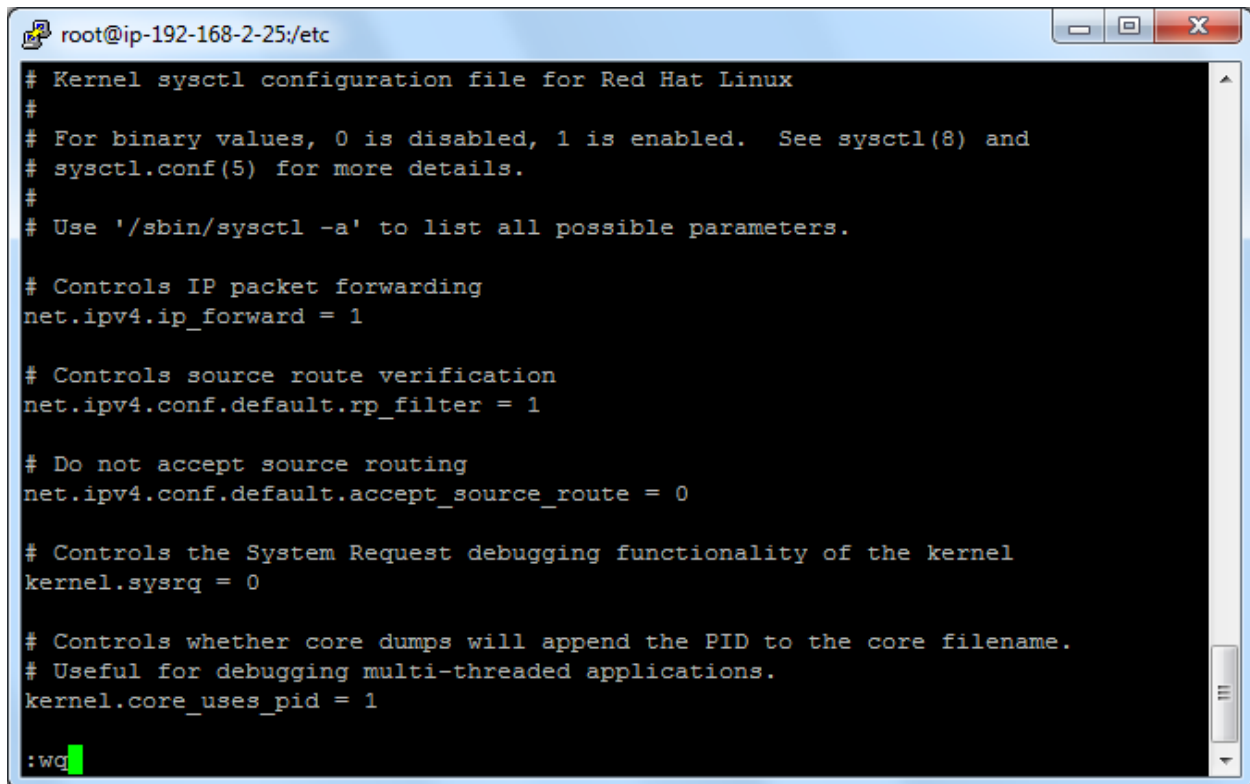
# Controls whether core dumps will append the PID to the core filename.
# Useful for debugging multi-threaded applications.
kernel.core_uses_pid = 1

-- INSERT --
```

Press escape key.

Type

:wq

A terminal window titled 'root@ip-192-168-2-25:/etc' displays the configuration file for sysctl. The window has a standard Linux terminal interface with a title bar and window controls. The text is as follows:

```
# 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
```

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 sidebar with categories like INSTANCES, IMAGES, ELASTIC BLOCK STORE, NETWORK & SECURITY, and LOAD BALANCING. The 'Network Interfaces' link under NETWORK & SECURITY is selected. The main content area displays a table of network interfaces. One interface, 'OpenSwan', is selected, and its 'Actions' menu is open. The menu options include '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 the selected network interface 'eni-6861ae3c' is shown, displaying various attributes in a key-value format.

Network Interface: eni-6861ae3c	
Network interface ID	eni-6861ae3c
VPC ID	vpc-0d56fb65
MAC address	06:95:48:87:d8:fa
Security groups	Ohio_Linux_Sec_Group . view inbound rules
Status	in-use
Private DNS (IPv4)	-
Subnet ID	subnet-f1ff1d8b
Availability Zone	us-east-2b
Description	Primary network interface
Owner ID	297111308396
Primary private IPv4 IP	192.168.2.25
IPv4 Public IP	18.218.11.25*

Set as “Disabled” and click “save”.

Change Source/Dest. Check X

Network Interface

eni-6861ae3c

Source/dest. check

☐ Enabled

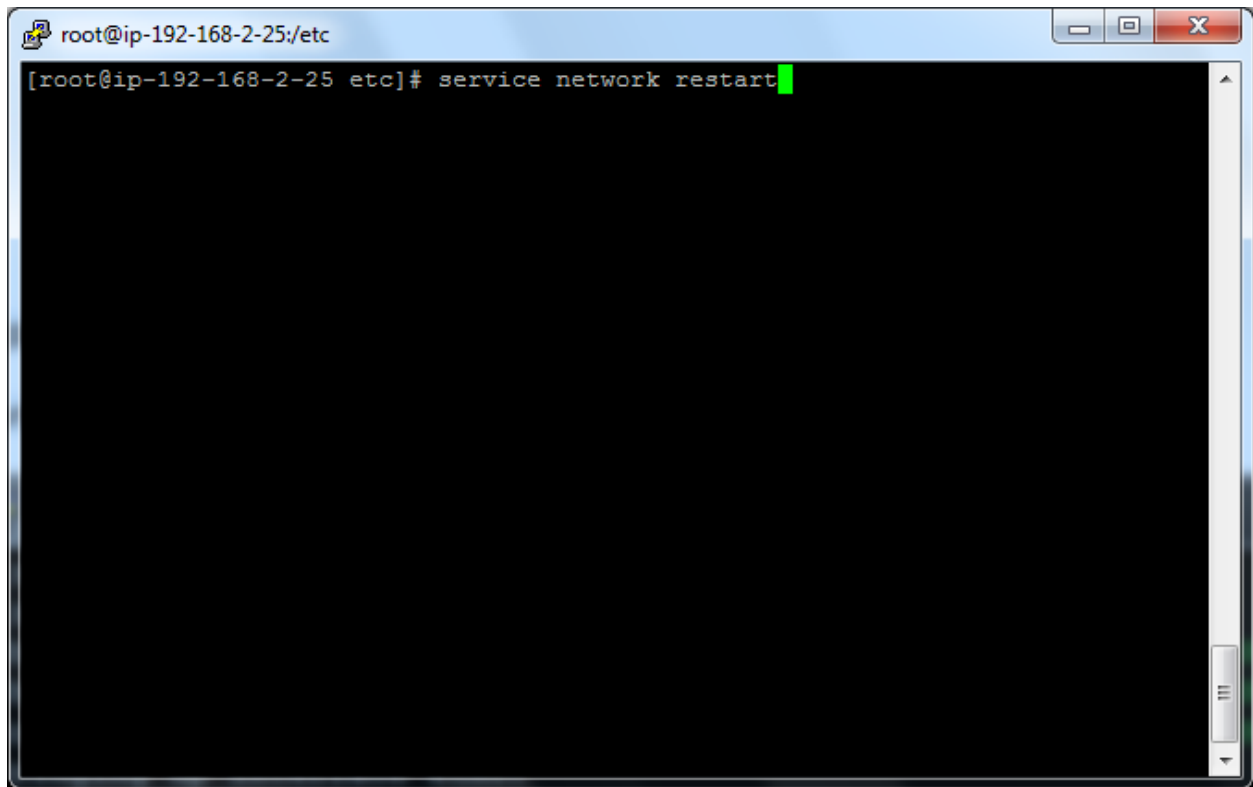
☒ Disabled

Cancel

Save

Type

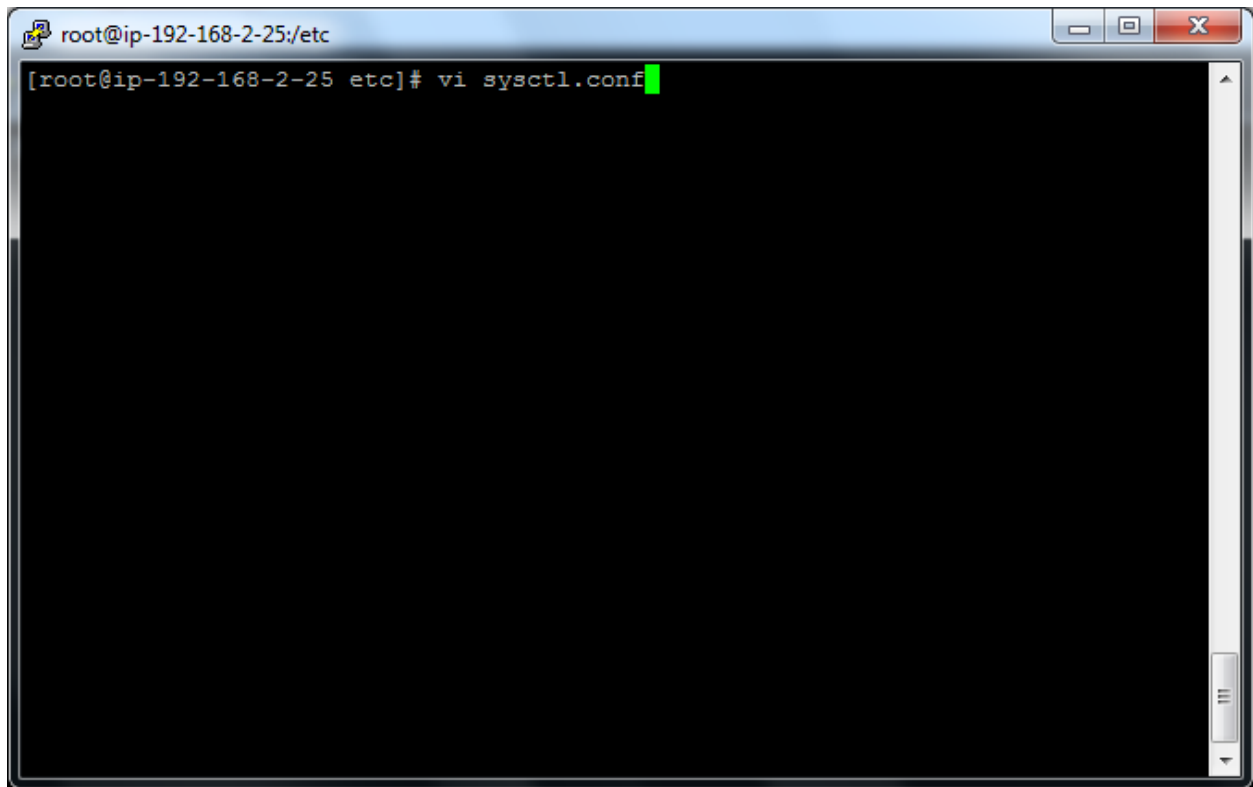
Service network restart

A terminal window with a blue title bar. The title bar text is 'root@ip-192-168-2-25:/etc'. The terminal content shows the command '[root@ip-192-168-2-25 etc]# service network restart' followed by a green cursor. The terminal has a black background and a vertical scrollbar on the right side.

```
root@ip-192-168-2-25:/etc
[root@ip-192-168-2-25 etc]# service network restart
```

Type

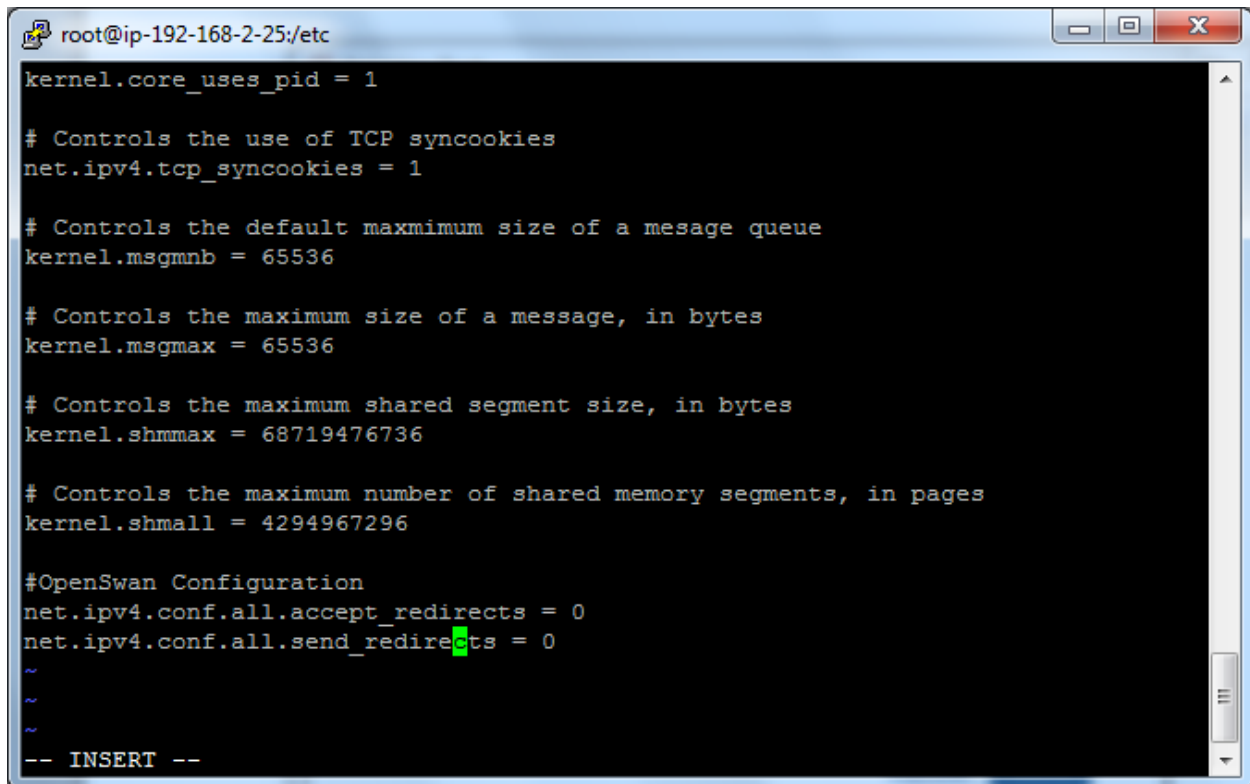
Vi sysctl.conf



Press insert key

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

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

A terminal window with a blue title bar and standard window controls. The terminal text is as follows:

```
root@ip-192-168-2-25:/etc
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

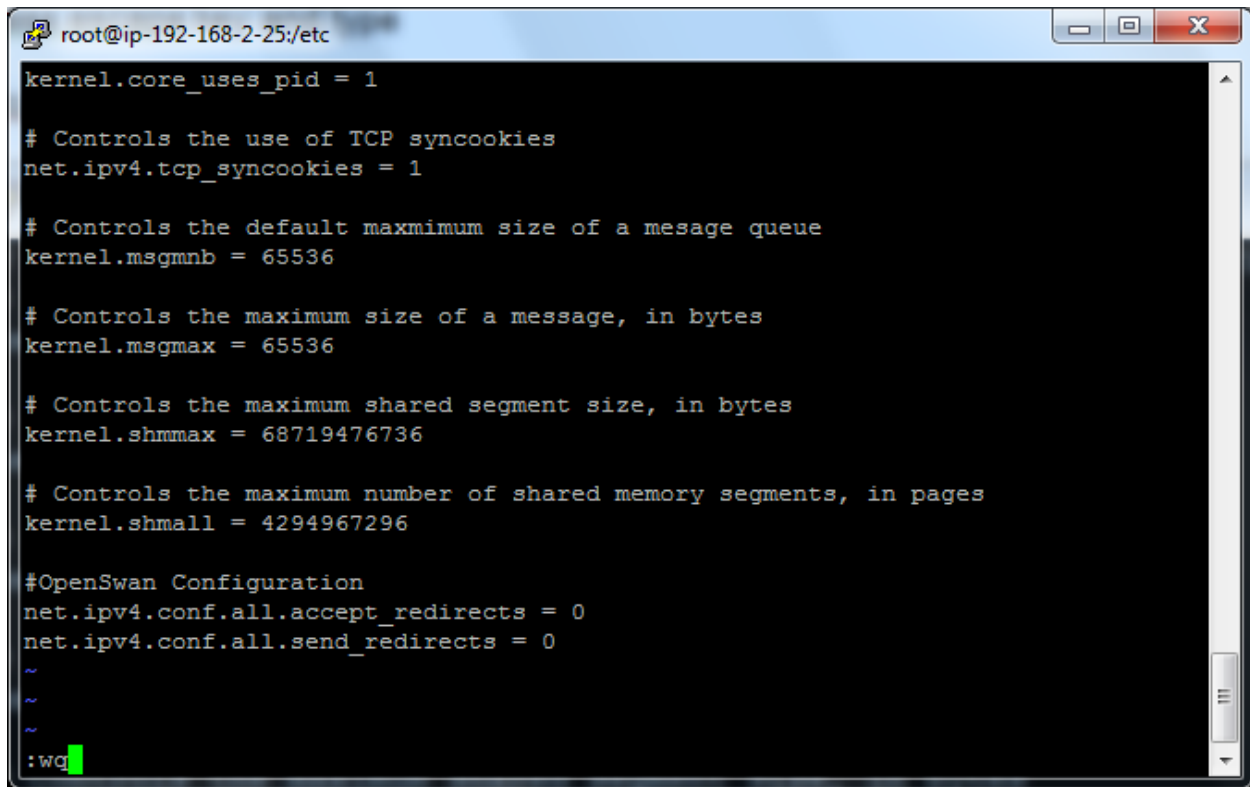
# Controls the maximum shared segment size, in bytes
kernel.shmmax = 68719476736

# 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 then type

:wq

A terminal window with a blue title bar showing the command prompt 'root@ip-192-168-2-25:/etc'. The terminal displays several configuration lines for kernel and network settings. The text is as follows:

```
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

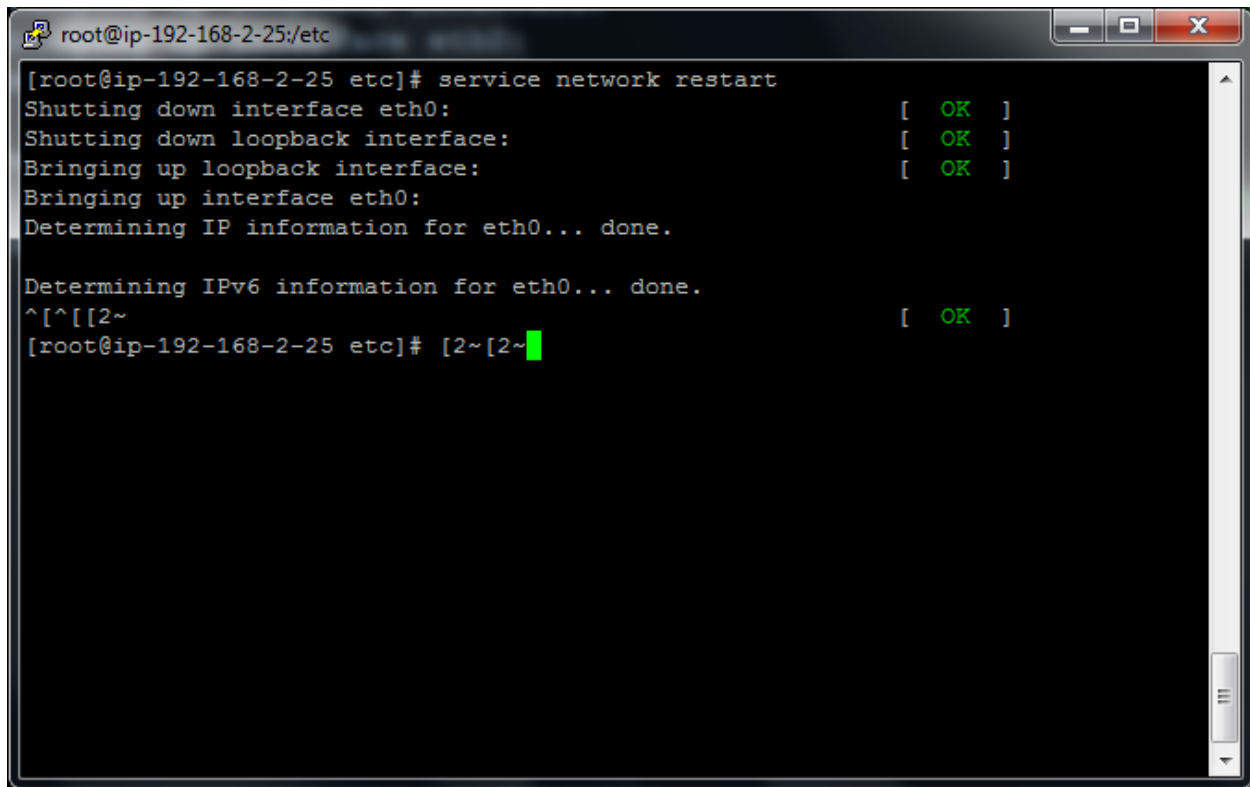
# Controls the maximum shared segment size, in bytes
kernel.shmmax = 68719476736

# 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

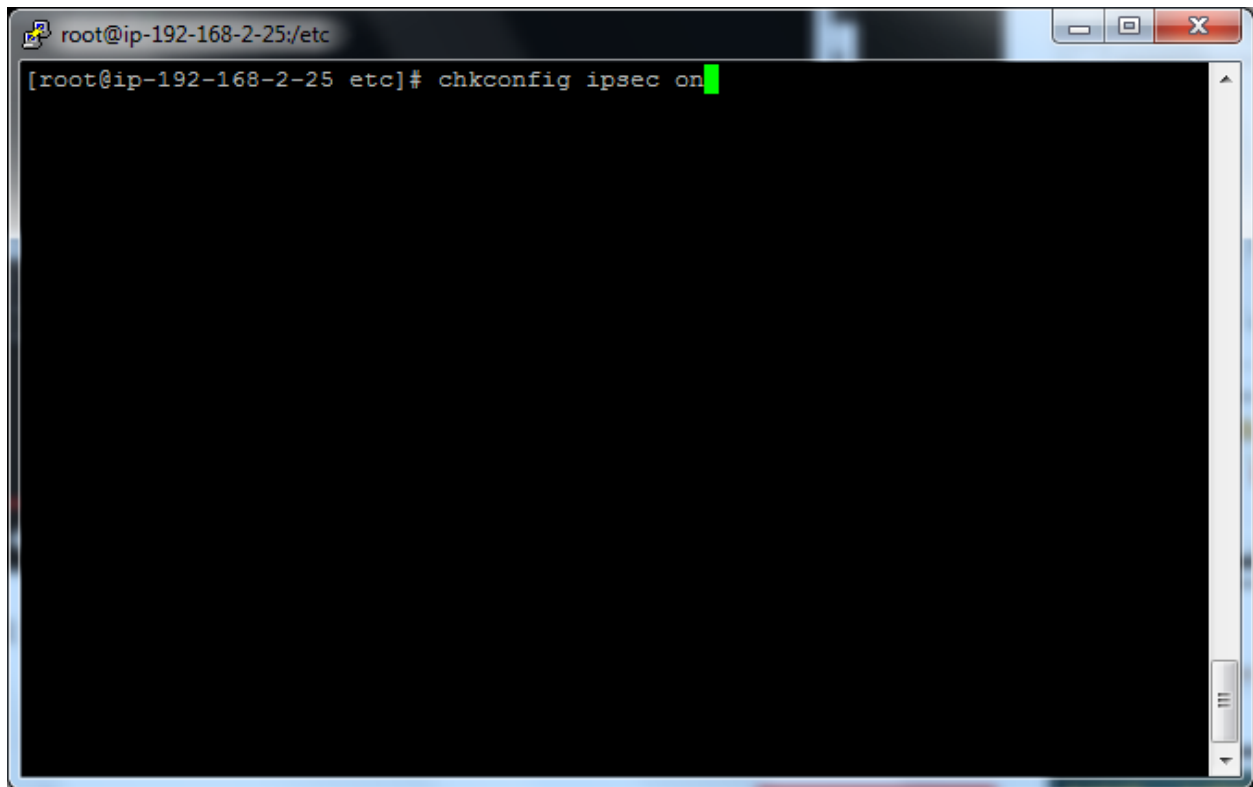
Service network restart

A terminal window with a dark background and light text. The title bar shows 'root@ip-192-168-2-25/etc'. The terminal content shows the command 'service network restart' and its output. The output indicates that the network service was restarted successfully, with 'eth0' and the loopback interface being brought up. The prompt is '[root@ip-192-168-2-25 etc]#'.

```
[root@ip-192-168-2-25 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.
^[^[[2~ [ OK ]
[root@ip-192-168-2-25 etc]# [2~[2~
```

Type `chkconfig ipsec on`



Go to VPC,

Click Route table and select sansbound public route table.

The screenshot shows the AWS VPC console interface. The left sidebar contains navigation links for VPC Dashboard, Virtual Private Cloud, and various network services. The main content area displays a list of Route Tables. The table 'Sansbound Public route ohio' is selected, and its details are shown in the 'Summary' tab.

Route Tables List:

Name	Route Table ID	Explicitly Associat	Main	VPC
	rtb-e0f42188	0 Subnets	Yes	vpc-1de10e75
<input checked="" type="checkbox"/> Sansbound Public route ohio	rtb-690a9401	1 Subnet	Yes	vpc-0d56fb65 Sansbound_Ohio...

Route Table Details: rtb-690a9401 | Sansbound Public route ohio

Summary Tab:

- Route Table ID: rtb-690a9401 | Sansbound Public route ohio
- Main: yes
- Explicitly Associated With: 1 Subnet
- VPC: vpc-0d56fb65 | Sansbound_Ohio_VPC

Click "Edit".

The screenshot shows the AWS Management Console interface for Route Tables. 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, with 'Sansbound Public route ohio' selected. Below the list, the detailed view for this route table is shown, including a table of routes.

Route Tables List:

Name	Route Table ID	Explicitly Associat	Main	VPC
Sansbound Public route ohio	rtb-690a9401	1 Subnet	Yes	vpc-0d56fb65 Sansbound_Ohio...

Route Details for rtb-690a9401 | Sansbound Public route ohio:

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

[Edit](#)

View: All rules

Destination	Target	Status	Propagated
192.168.0.0/16	local	Active	No
0.0.0.0/0	igw-909a9df9	Active	No

Click “add another route”.

The screenshot shows the AWS Management Console interface. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and user information. The left sidebar lists various VPC services, with 'Route Tables' highlighted. The main content area displays the 'VPC Dashboard' with buttons for 'Create Route Table', 'Delete Route Table', and 'Set As Main Table'. Below these, a search bar and a table of route tables are visible. The table lists two route tables: 'rtb-e0f42188' (0 Subnets) and 'rtb-690a9401' (1 Subnet). The second route table is selected, showing its details under the 'Routes' tab. The 'Routes' tab displays a table with columns: Destination, Target, Status, Propagated, and Remove. The first row shows a route to '192.168.0.0/16' with target 'local' and status 'Active'. The second row shows a route to '0.0.0.0/0' with target 'igw-909a9df9' and status 'Active'. A yellow highlight is placed over the 'Add another route' button at the bottom of the routes table.

Route Tables | VPC Manager | (98) AWS Demo 4: Connect to...

Secure | https://us-east-2.console.aws.amazon.com/vpc/home?region=us-east-2#routetables:

aws Services Resource Groups siva1n82 Ohio Support

VPC Dashboard

Filter by VPC: Select a 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

Create Route Table Delete Route Table Set As Main Table

Search Route Tables and their associated subnets

<< 1 to 2 of 2 Route Tables >>

Name	Route Table ID	Explicitly Associated Subnets	Main	VPC
	rtb-e0f42188	0 Subnets	Yes	vpc-1de10e75
Sansbound Public route ohio	rtb-690a9401	1 Subnet	Yes	vpc-0d56fb65 Sansbound_Ohio...

rtb-690a9401 | Sansbound Public route ohio

Summary Routes Subnet Associations Route Propagation Tags

Cancel Save

View: All rules

Destination	Target	Status	Propagated	Remove
192.168.0.0/16	local	Active	No	
0.0.0.0/0	igw-909a9df9	Active	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

Type 10.0.0.0/16 subnet as destination and select “VPN Linux Server” as target.

The screenshot shows the AWS VPC console interface. On the left, there is a navigation menu with options like 'Virtual Private Cloud', 'Your VPCs', 'Subnets', 'Route Tables', 'Internet Gateways', etc. The main area displays the 'Routes' tab for the route table 'rtb-690a9401 | Sansbound Public route ohio'. Below the tabs, there is a table of routes with columns: Destination, Target, Status, Propagated, and Remove. The table shows three routes: a local route for 192.168.0.0/16, and two routes for 0.0.0.0/0 and 10.0.0.0/16 both pointing to the target 'igw-909a9df9'. A dropdown menu is open for the 'Target' field of the 10.0.0.0/16 route, showing suggestions like 'igw-909a9df9 | Sansbound_Ohio_IGW' and 'i-04f8379b52537bb4e | VPN Linux Serve...'. At the bottom of the console, there is a footer with 'Feedback', 'English (US)', and copyright information.

Route Tables List:

Name	Route Table ID	Explicitly Associat	Main	VPC
	rtb-e0f42188	0 Subnets	Yes	vpc-1de10e75
<input checked="" type="checkbox"/> Sansbound Public route ohio	rtb-690a9401	1 Subnet	Yes	vpc-0d56fb65 Sansbound_Ohio...

Route Table Details: rtb-690a9401 | Sansbound Public route ohio

Routes Table:

Destination	Target	Status	Propagated	Remove
192.168.0.0/16	local	Active	No	
0.0.0.0/0	igw-909a9df9	Active	No	✕
10.0.0.0/16	igw-909a9df9 Sansbound_Ohio_IGW		No	✕

Click “save”.

Click To view detailed information of routing table as below.

The screenshot displays the AWS VPC console interface. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and user information. The left sidebar lists various VPC resources, with 'Route Tables' highlighted. The main content area shows a list of route tables, with 'rtb-690a9401 | Sansbound Public route ohio' selected. Below this, the 'Routes' tab is active, showing a table of routes with columns for Destination, Target, Status, and Propagated.

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 List:

Name	Route Table ID	Explicitly Associat	Main	VPC
	rtb-e0f42188	0 Subnets	Yes	vpc-1de10e75
<input checked="" type="checkbox"/> Sansbound Public route ohio	rtb-690a9401	1 Subnet	Yes	vpc-0d56fb65 Sansbound_Ohio...

rtb-690a9401 | Sansbound Public route ohio

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

[Edit](#)

View:

Destination	Target	Status	Propagated
192.168.0.0/16	local	Active	No
0.0.0.0/0	igw-909a9df9	Active	No
10.0.0.0/16	eni-6861ae3c / i-04f8379b52537bb4e	Active	No

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