Lab 01: Using ProxyCannon

Contents

Goals

- Setup ProxyCannon in Amazon AWS.
- Practice using ProxyCannon to change your source IP address.

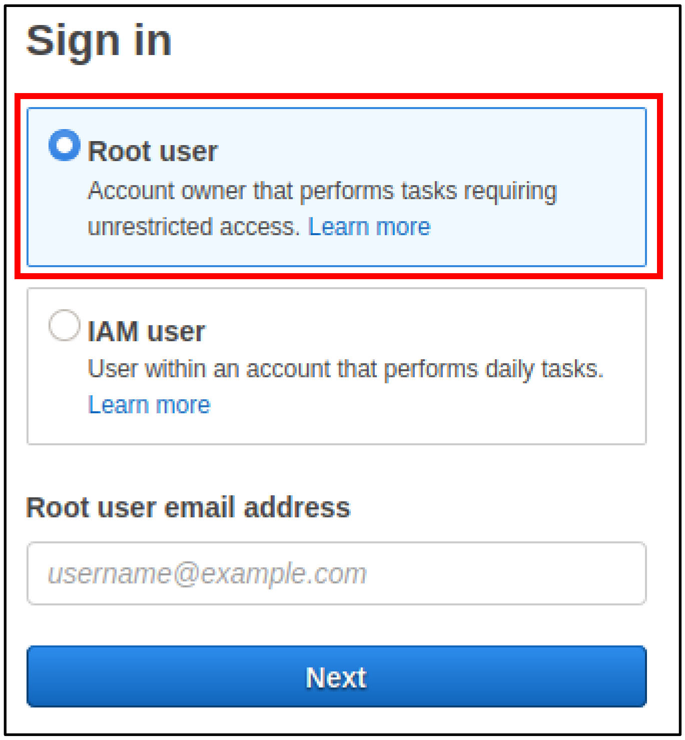
Requirements

- Amazon AWS account.
- Kali Linux VM with Internet access.

1. Create the key pair used for logging in to the ProxyCannon server

1. In your Kali Linux VM, open a web browser and log in to the AWS Management Console at the URL listed below. Choose "Root user" when logging in to your Amazon AWS account.

```
https://console.aws.amazon.com/
```



Sign in

● Root user
Account owner that performs tasks requiring unrestricted access. Learn more

○ IAM user
User within an account that performs daily tasks. Learn more

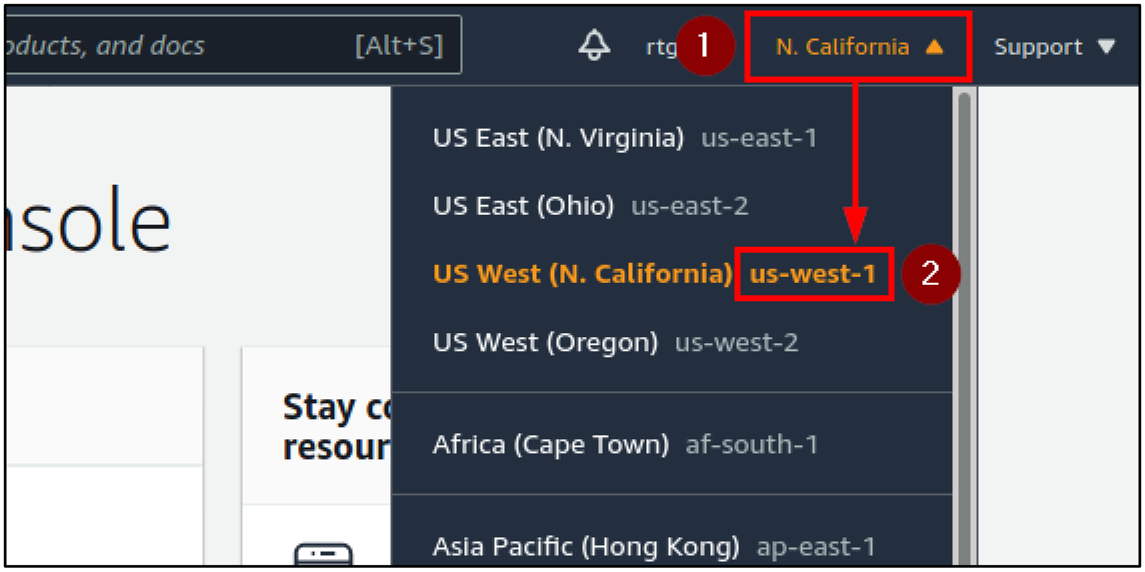
Root user email address

username@example.com

Next

Select Root User When Logging in to AWS

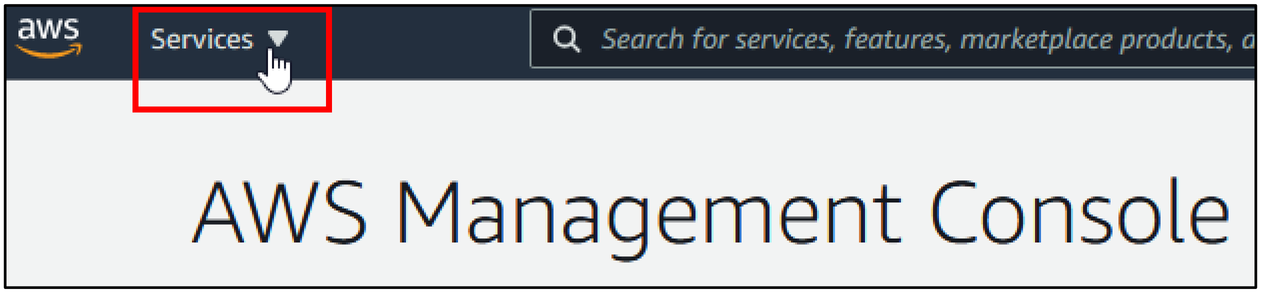
2. At the main screen of the AWS Management Console, click on the location menu in the top right corner of the page. Then in the menu that appears, click on the region where you want your ProxyCannon servers to be located. **Be sure to note the region name (example: "us-west-1") since you will need it in later steps.**



oducts, and docs [Alt+S]
rtg
1
N. California ▲
Support ▼
nsole
US East (N. Virginia) us-east-1
US East (Ohio) us-east-2
US West (N. California) us-west-1
2
US West (Oregon) us-west-2
Africa (Cape Town) af-south-1
Asia Pacific (Hong Kong) ap-east-1
Stay co
resour

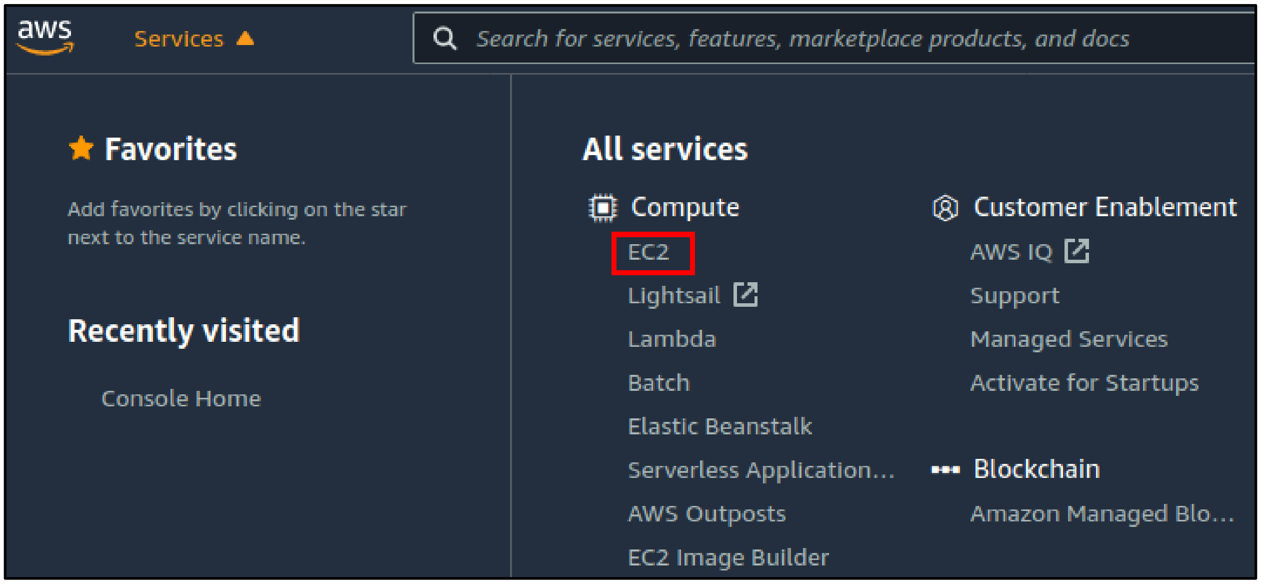
"us-west-1" Region Selected

3. Next, click on the “Services” menu in the upper left corner of the page.



Services Menu

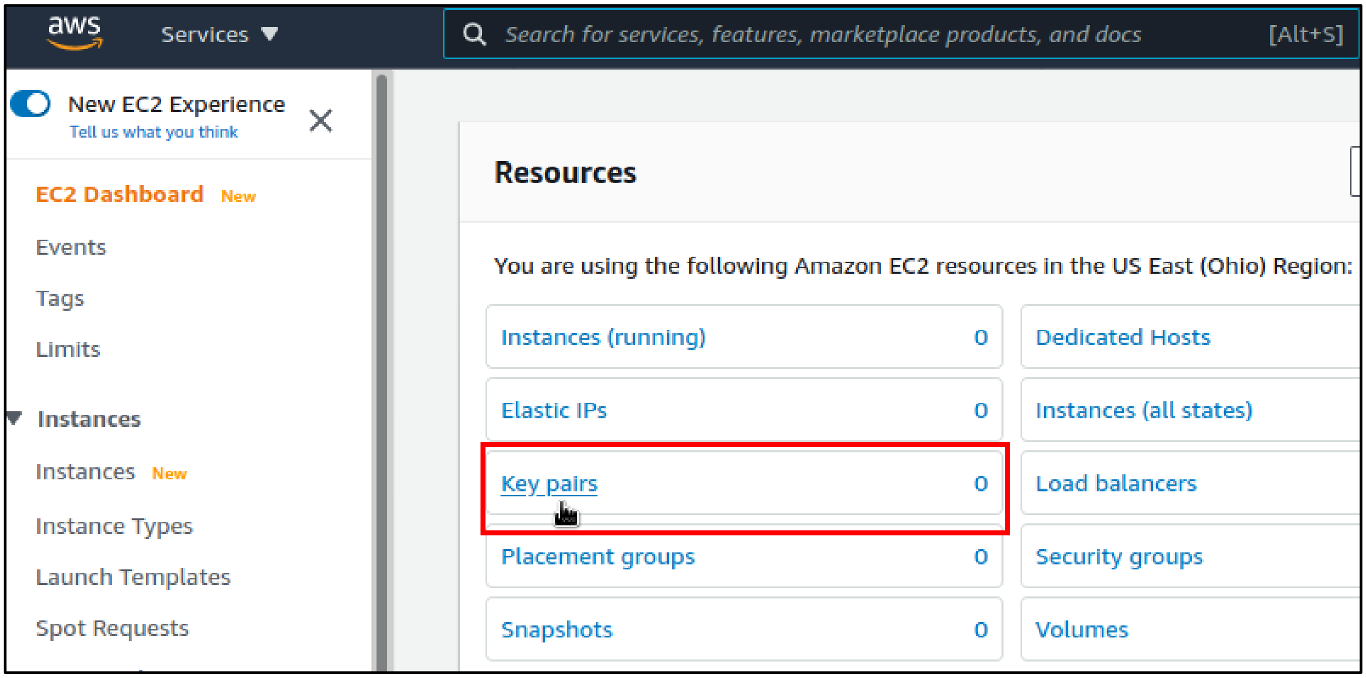
3. Click on "EC2" under the "Compute" section of the Services menu.



Add favorites by clicking on the star next to the service name.

EC2 Link

4. Click on "Key Pairs" under the Resources heading on the page.



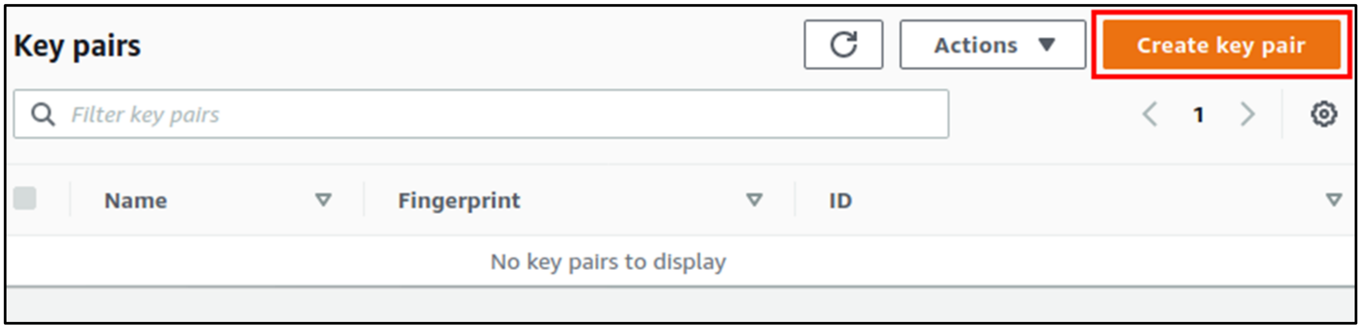
Resources

You are using the following Amazon EC2 resources in the US East (Ohio) Region:

Instances (running)	0	Dedicated Hosts
Elastic IPs	0	Instances (all states)
Key pairs	0	Load balancers
Placement groups	0	Security groups
Snapshots	0	Volumes

Key Pairs Link

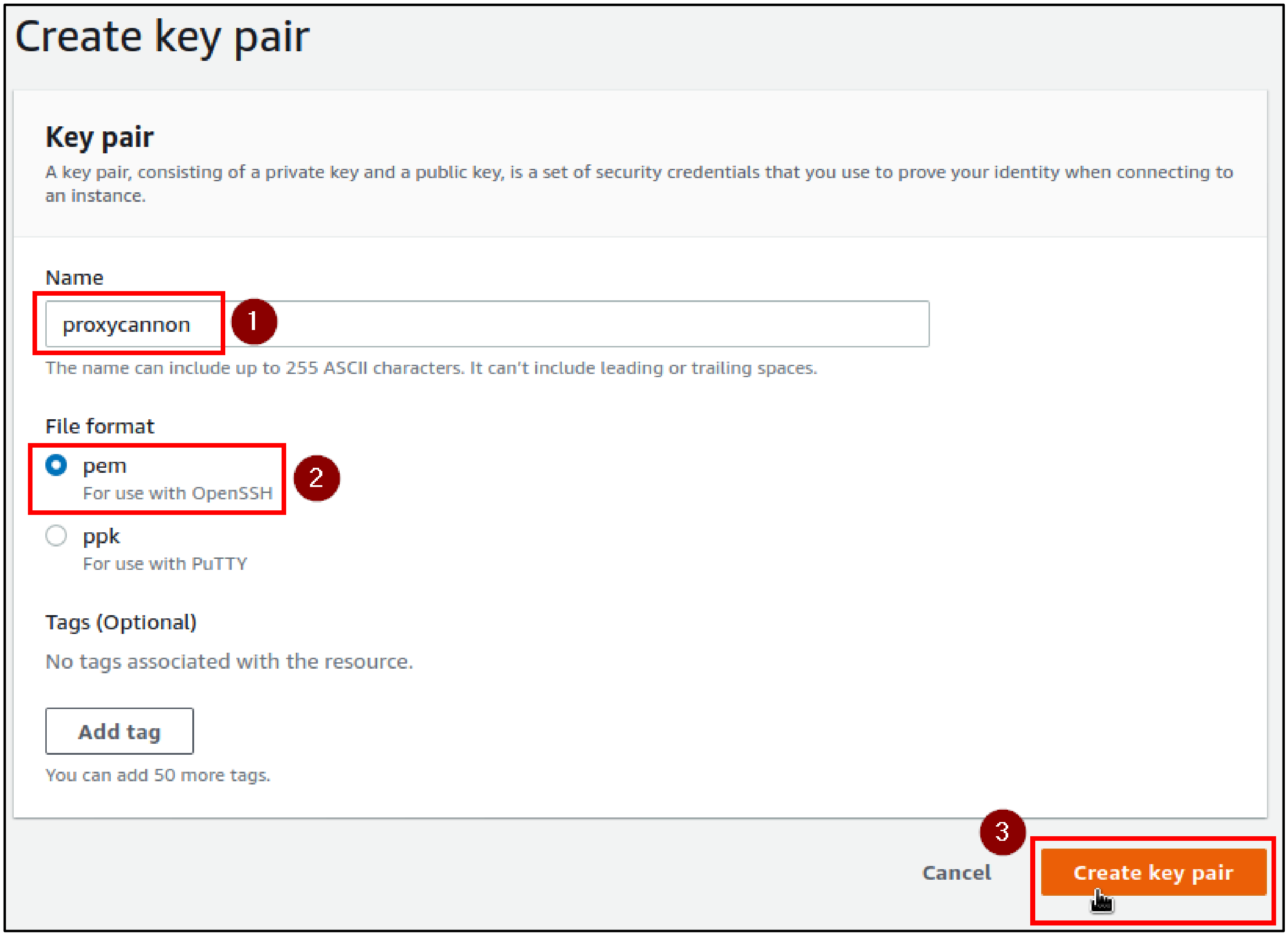
5. Click on the "Create Key Pair" button in the top right.



□	Name	Fingerprint	ID
	No key pairs to display		

"Create Key Pair" Button Clicked

6. On the "Create key pair" page, name the key pair "proxycannon", and set the file format to "pem". Then click the "Create key pair" button at the bottom of the page.



Create key pair

Key pair

A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance.

Name

proxycannon

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

File format

● pem
For use with OpenSSH

○ ppk
For use with PuTTY

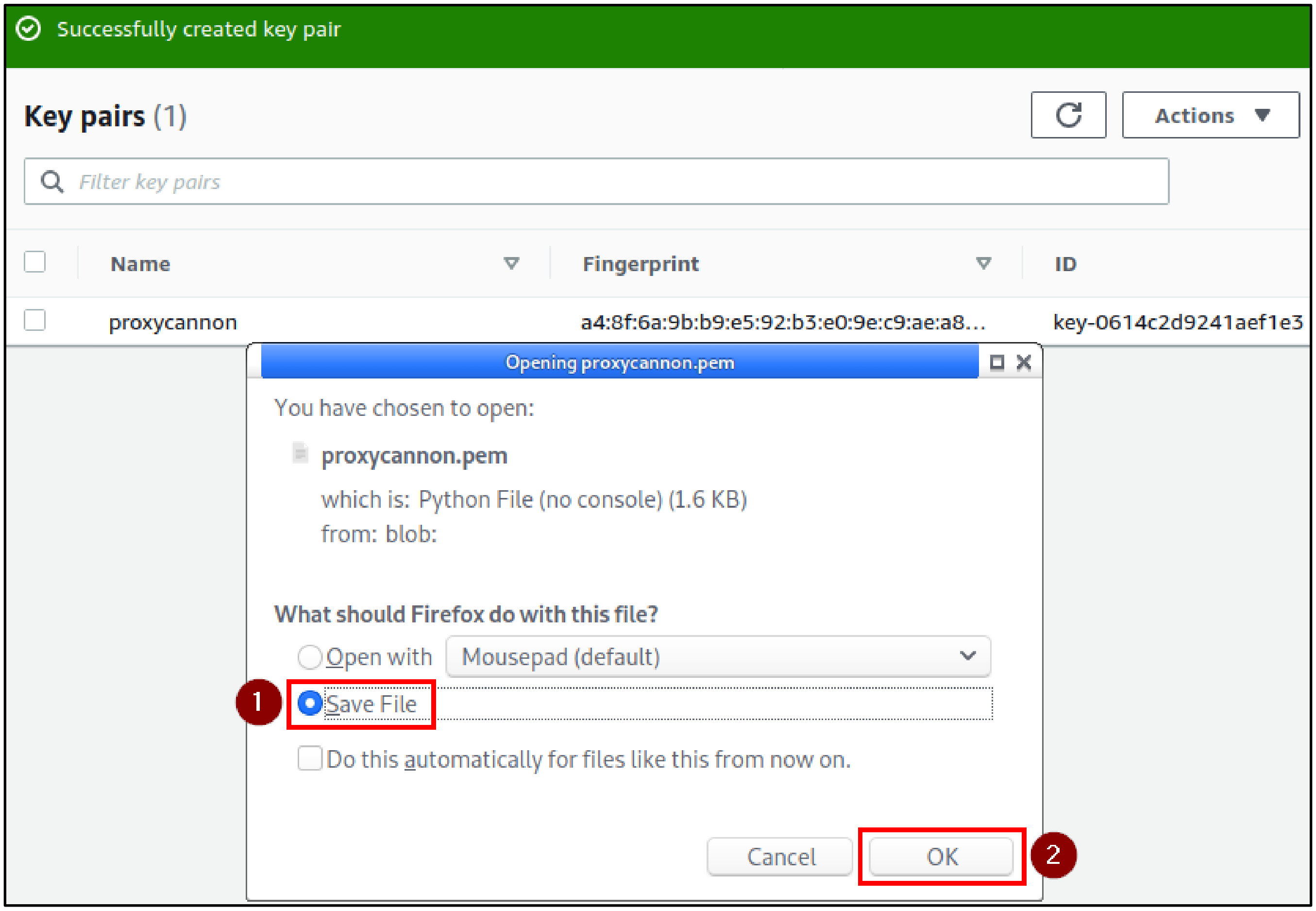
Tags (Optional)

No tags associated with the resource.

You can add 50 more tags.

Configuration of the New Key Pair

7. You should receive a prompt to download the key file ("proxycannon.pem"). Make sure "Save file" is selected in the dialog box, and then click "OK" to save the file to your Downloads folder.



Successfully created key pair

Key pairs (1)

Filter key pairs

	Name	Fingerprint	ID
□	proxycannon	a4:8f:6a:9b:b9:e5:92:b3:e0:9e:c9:ae:a8...	key-0614c2d9241aef1e3

Opening proxycannon.pem

You have chosen to open:

proxycannon.pem

which is: Python File (no console) (1.6 KB)
from: blob:

What should Firefox do with this file?

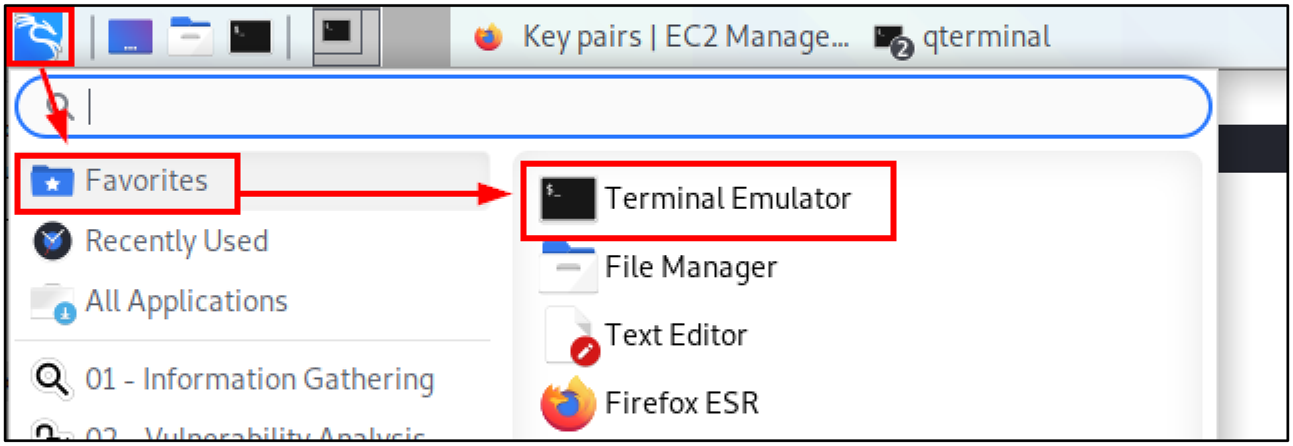
○ Open with Mousepad (default)

● Save File

□ Do this automatically for files like this from now on.

Saving the ProxyCannon Key File

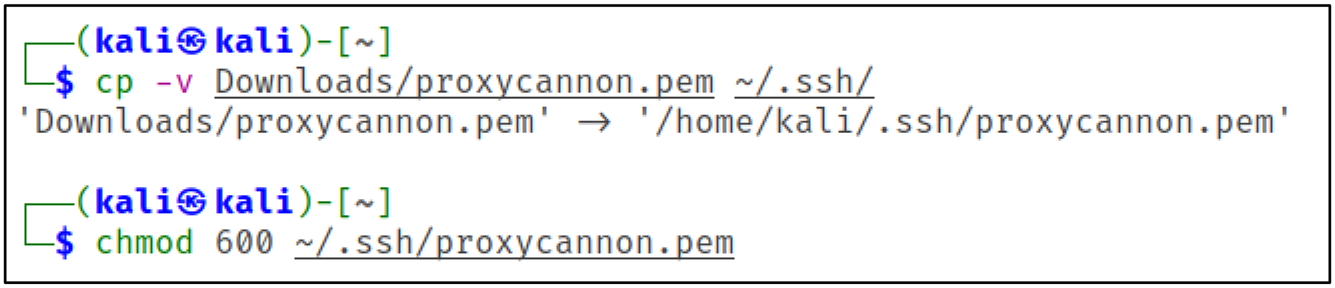
8. In your Kali VM, open a Terminal window by clicking on the Applications menu in the top left corner and then clicking on "Terminal Emulator" inside the "Favorites" folder.



Launching Terminal Emulator

9. In the terminal window, run the two commands shown below to copy the SSH key file into your ".ssh" directory and assign it the proper file permissions.

```
cp -v Downloads/proxycannon.pem ~/.ssh/

chmod 600 ~/.ssh/proxycannon.pem
```

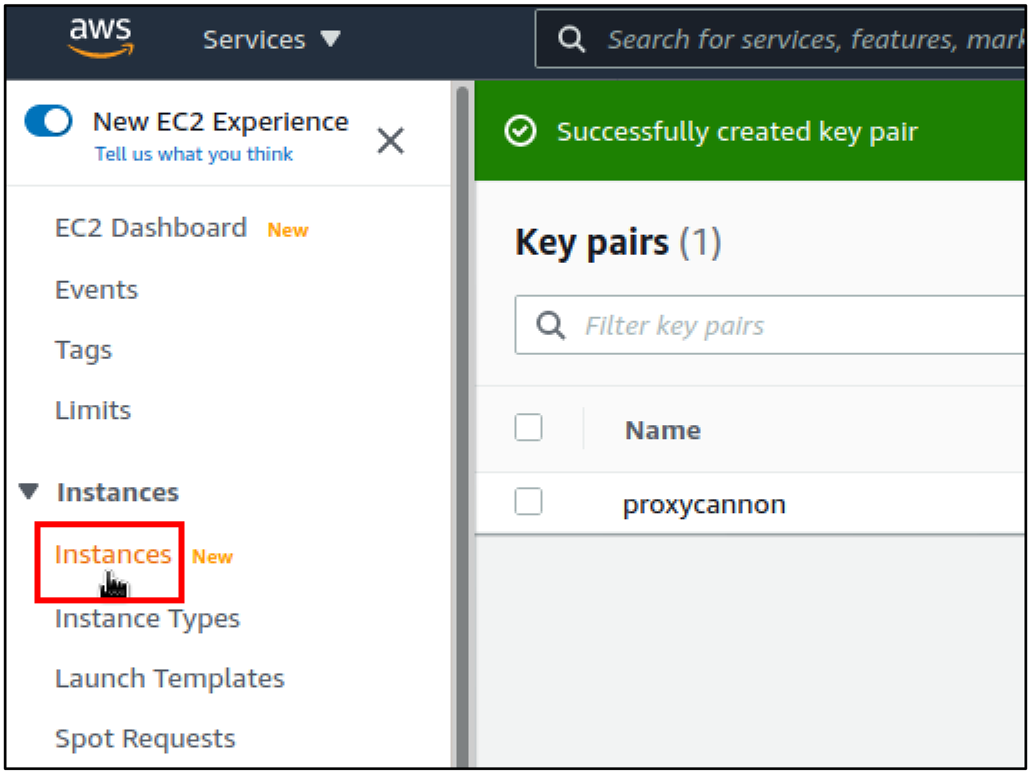


```
┌──(kali㉿kali)-[~]
└─$ cp -v Downloads/proxycannon.pem ~/.ssh/
'Downloads/proxycannon.pem' -> '/home/kali/.ssh/proxycannon.pem'

┌──(kali㉿kali)-[~]
└─$ chmod 600 ~/.ssh/proxycannon.pem
```

Execution of Above Commands

2. Create the ProxyCannon control server instance

1. In your web browser, click on the "Instances" link in the sidebar on the left side of the page.



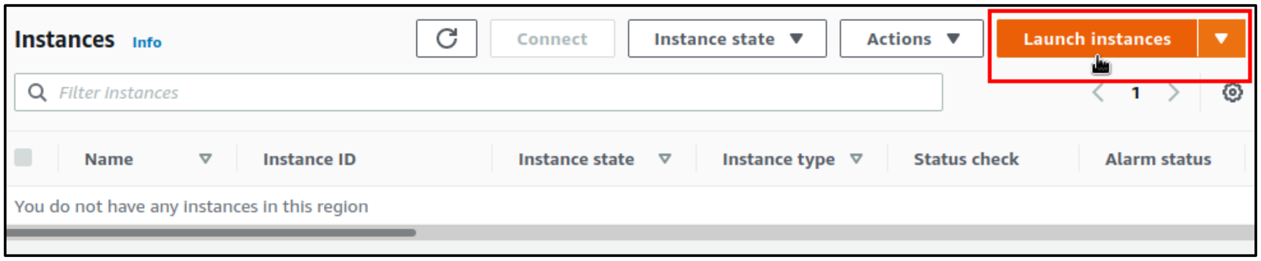
Successfully created key pair

Key pairs (1)

□	Name
□	proxycannon

Clicking the Instances Link

2. Click on the "Launch Instances" button in the top right corner of the Instances page.

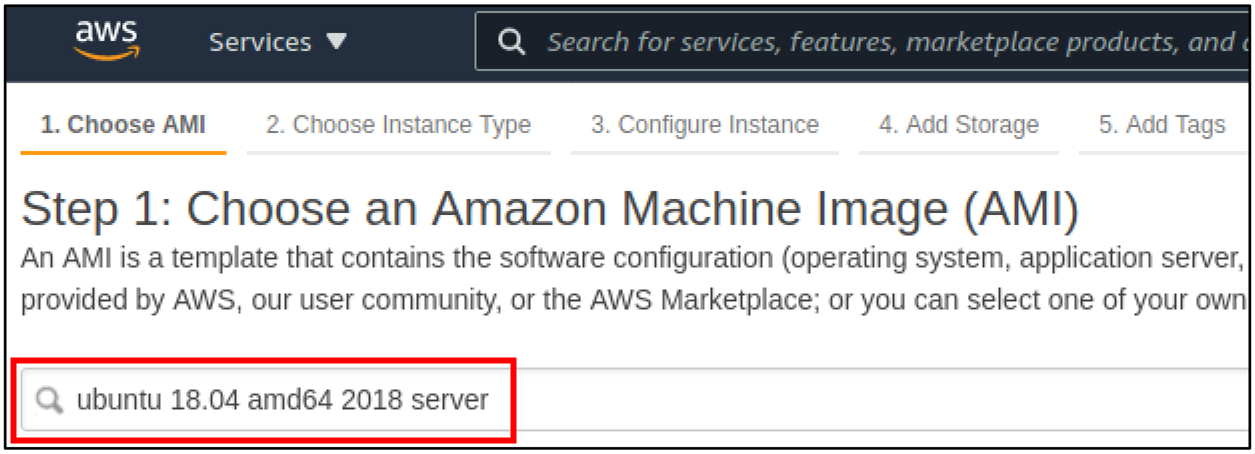


You do not have any instances in this region

Launch Instances Button

3. Copy and paste the following text into the search box on the next page and press Enter to perform the search:

```
ubuntu 18.04 amd64 2018 server
```



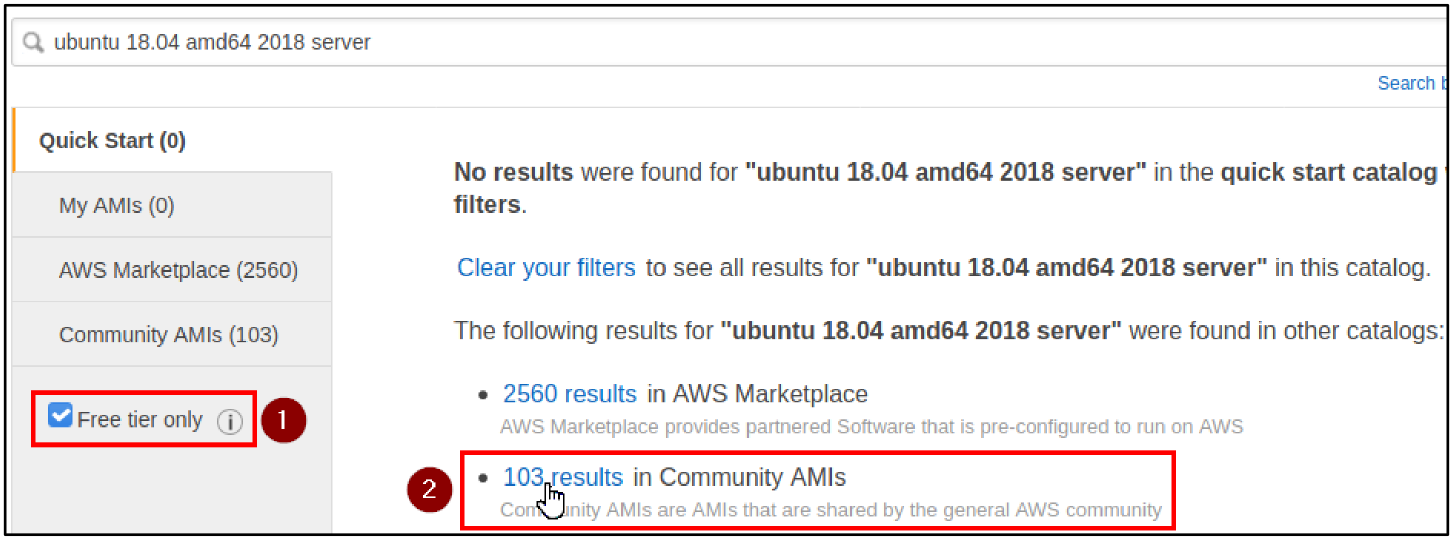
Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server,
provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own

ubuntu 18.04 amd64 2018 server

Search for a Suitable AMI

4. If you're using the AWS free trial, be sure to check the box labelled "Free tier only" on the left. Then you should see a number of Community AMIs reported below the search box. Click on the link to see the search results in Community AMIs.



ubuntu 18.04 amd64 2018 server

No results were found for **"ubuntu 18.04 amd64 2018 server"** in the **quick start catalog** **filters**.

Clear your filters to see all results for **"ubuntu 18.04 amd64 2018 server"** in this catalog.

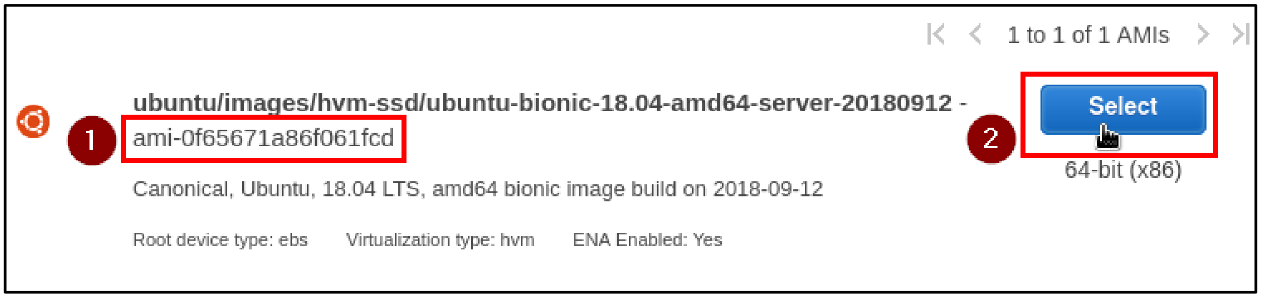
The following results for **"ubuntu 18.04 amd64 2018 server"** were found in other catalogs:

- 2560 results in AWS Marketplace
  AWS Marketplace provides partnered Software that is pre-configured to run on AWS
- 103 results in Community AMIs
  Community AMIs are AMIs that are shared by the general AWS community

Clicking on "1 results" in Community AMIs

5. Any of the standard AMD64, Ubuntu 18.04, server AMIs should be compatible with ProxyCannon. Find a suitable AMI in the list, and record the AMI identifier shown under the image name. You will need this information again when you configure ProxyCannon.

   Then click the "Select" button beside the AMI image to continue to the next step.



ubuntu/images/hvm-ssd/ubuntu-bionic-18.04-amd64-server-20180912 - ami-0f65671a86f061fcd

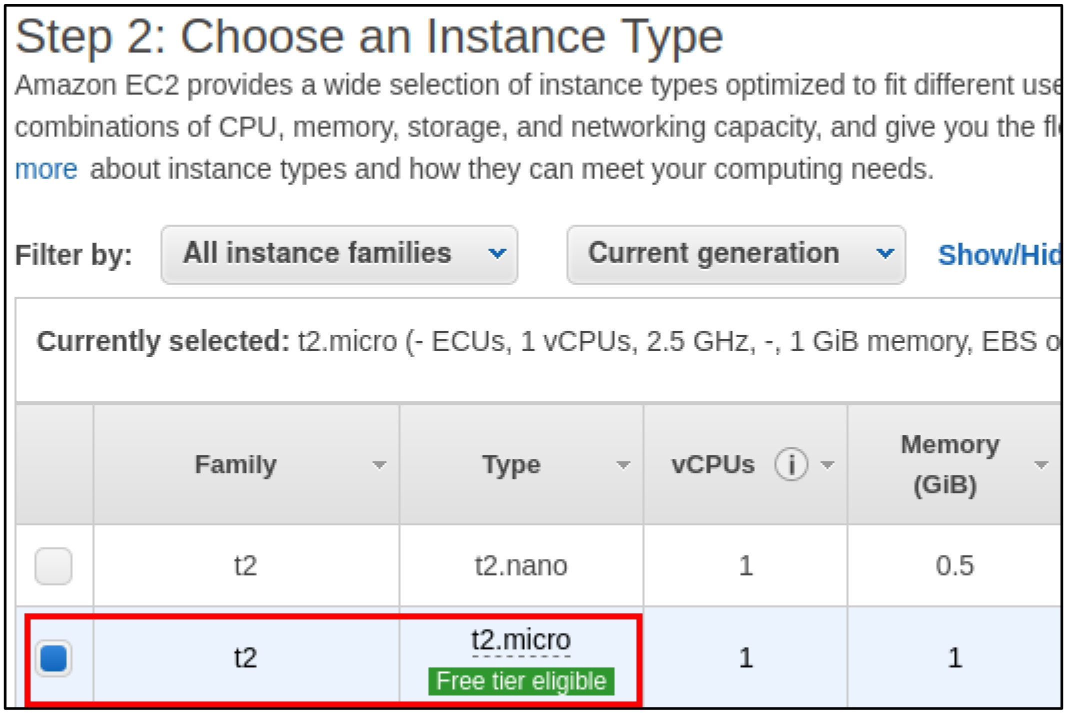
Canonical, Ubuntu, 18.04 LTS, amd64 bionic image build on 2018-09-12

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

64-bit (x86)

Selecting a Matching AMI

6. On the next page, choose the "t2.micro" instance type from the list, which is eligible for the AWS Free Tier



Step 2: Choose an Instance Type

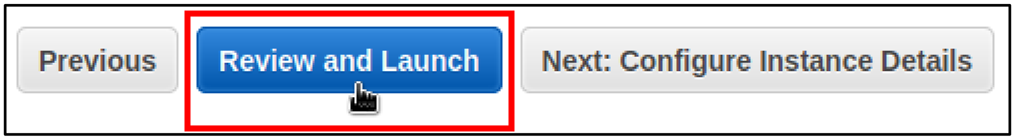
Amazon EC2 provides a wide selection of instance types optimized to fit different use combinations of CPU, memory, storage, and networking capacity, and give you the fl more about instance types and how they can meet your computing needs.

Currently selected: t2.micro (- ECUs, 1 vCPUs, 2.5 GHz, -, 1 GiB memory, EBS o

	Family	Type	vCPUs	Memory (GiB)
□	t2	t2.nano	1	0.5
■	t2	t2.micro Free tier eligible	1	1

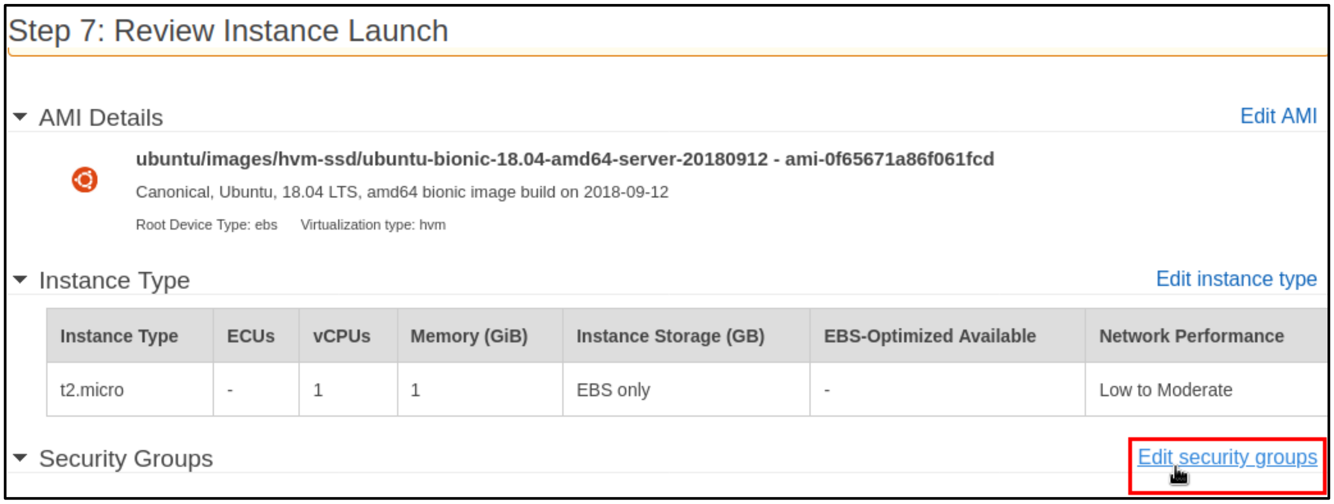
Selecting the "t2.micro" Instance Type

7. Then click the "Review and Launch" button at the bottom of the page.



Clicking Review and Launch

8. On the "Review Instance Launch" page, scroll down to the "Security Groups" section, and click the "Edit security groups" link.



Step 7: Review Instance Launch

AMI Details

ubuntu/images/hvm-ssd/ubuntu-bionic-18.04-amd64-server-20180912 - ami-0f65671a86f061fcd

Canonical, Ubuntu, 18.04 LTS, amd64 bionic image build on 2018-09-12

Root Device Type: ebs   Virtualization type: hvm

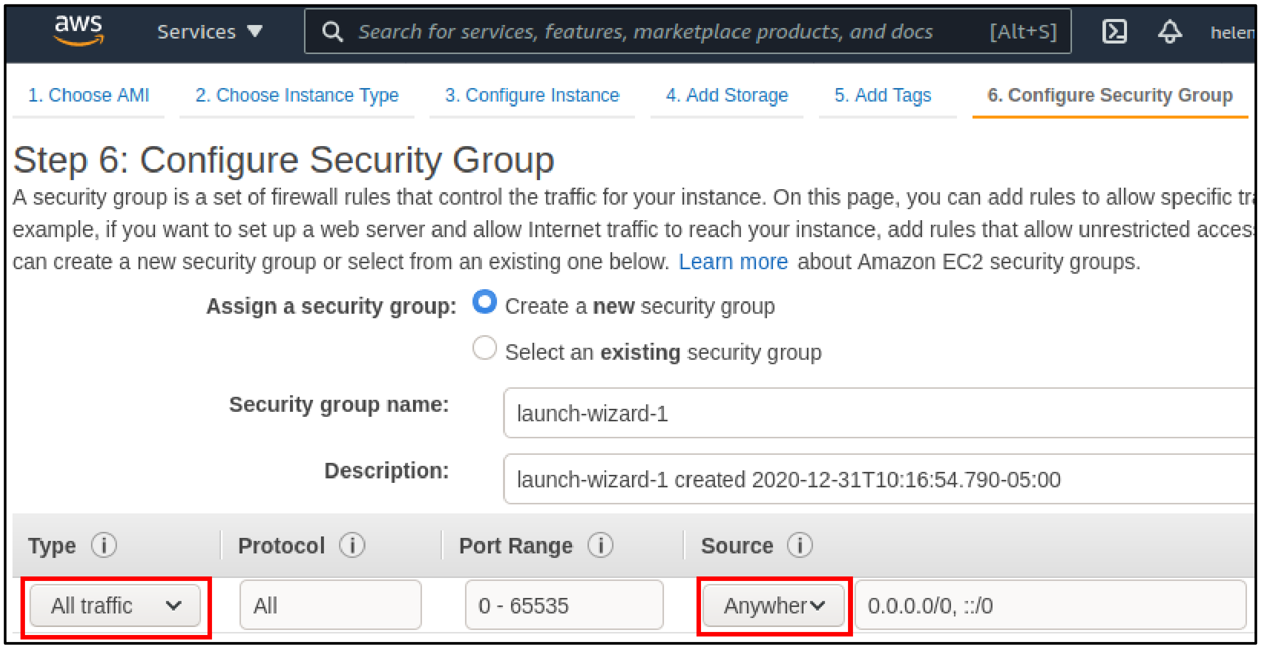
Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	-	1	1	EBS only	-	Low to Moderate

Security Groups

Clicking "Edit security groups"

9. On the "Configure Security Group" page use the drop-down fields to change Type to "All traffic" and Source to "Anywhere".



Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific tr example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted acces can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

Assign a security group: ● Create a **new** security group

○ Select an **existing** security group

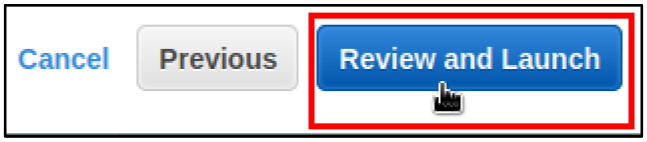
Security group name: launch-wizard-1

Description: launch-wizard-1 created 2020-12-31T10:16:54.790-05:00

Type	Protocol	Port Range	Source	
All traffic	All	0 - 65535	Anywher	0.0.0.0/0, ::/0

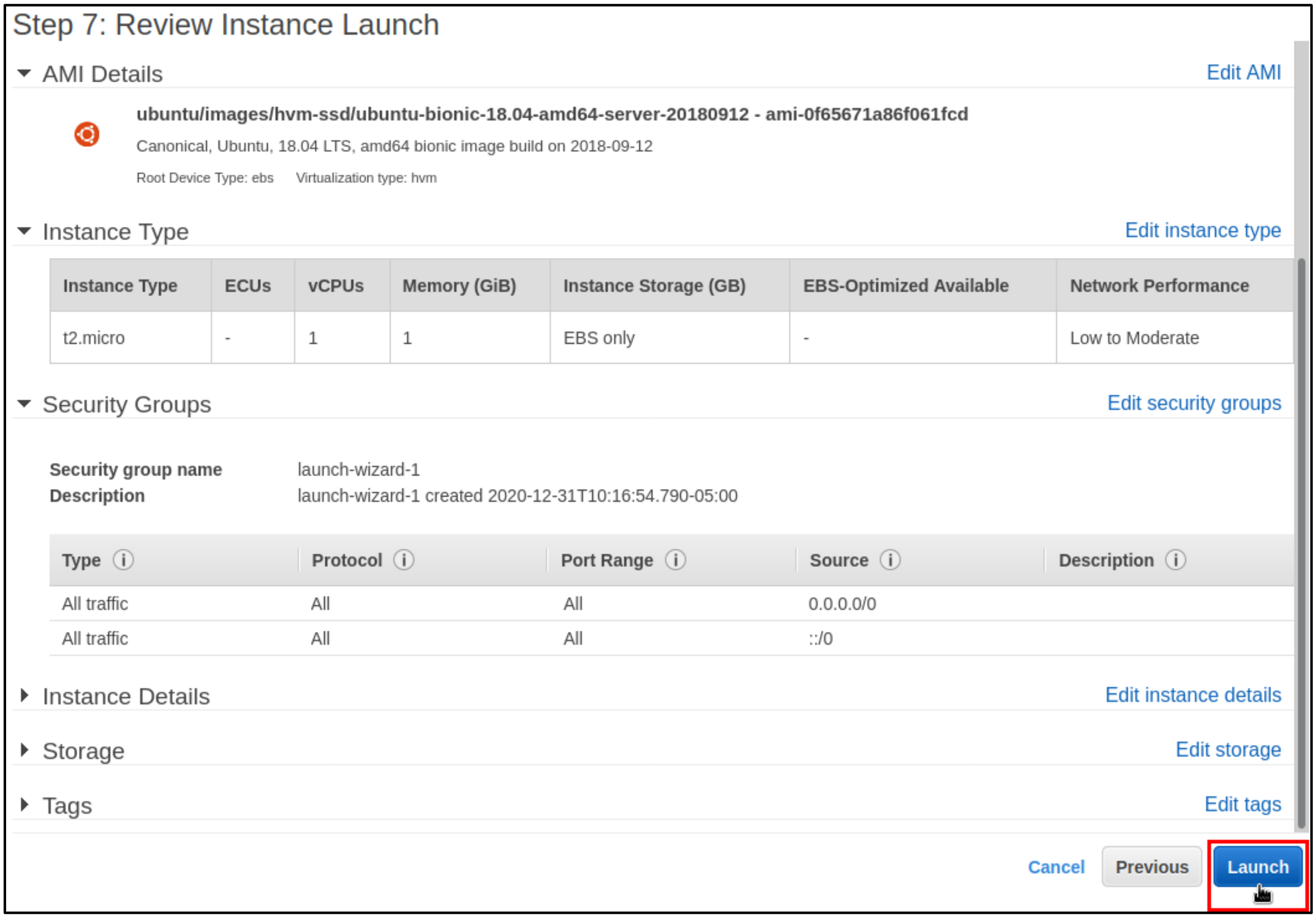
Configuring the Security Group

10. Then click the "Review and Launch" button at the bottom of the page to save your changes to the security group.



Clicking Review and Launch

11. Now click the "Launch" button at the bottom of the "Review Instance Launch" page.



Step 7: Review Instance Launch

▾ AMI Details

ubuntu/images/hvm-ssd/ubuntu-bionic-18.04-amd64-server-20180912 - ami-0f65671a86f061fcd

Canonical, Ubuntu, 18.04 LTS, amd64 bionic image build on 2018-09-12

Root Device Type: ebs Virtualization type: hvm

▾ Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	-	1	1	EBS only	-	Low to Moderate

▾ Security Groups

Security group name launch-wizard-1
Description launch-wizard-1 created 2020-12-31T10:16:54.790-05:00

Type	Protocol	Port Range	Source	Description
All traffic	All	All	0.0.0.0/0	
All traffic	All	All	::/0	

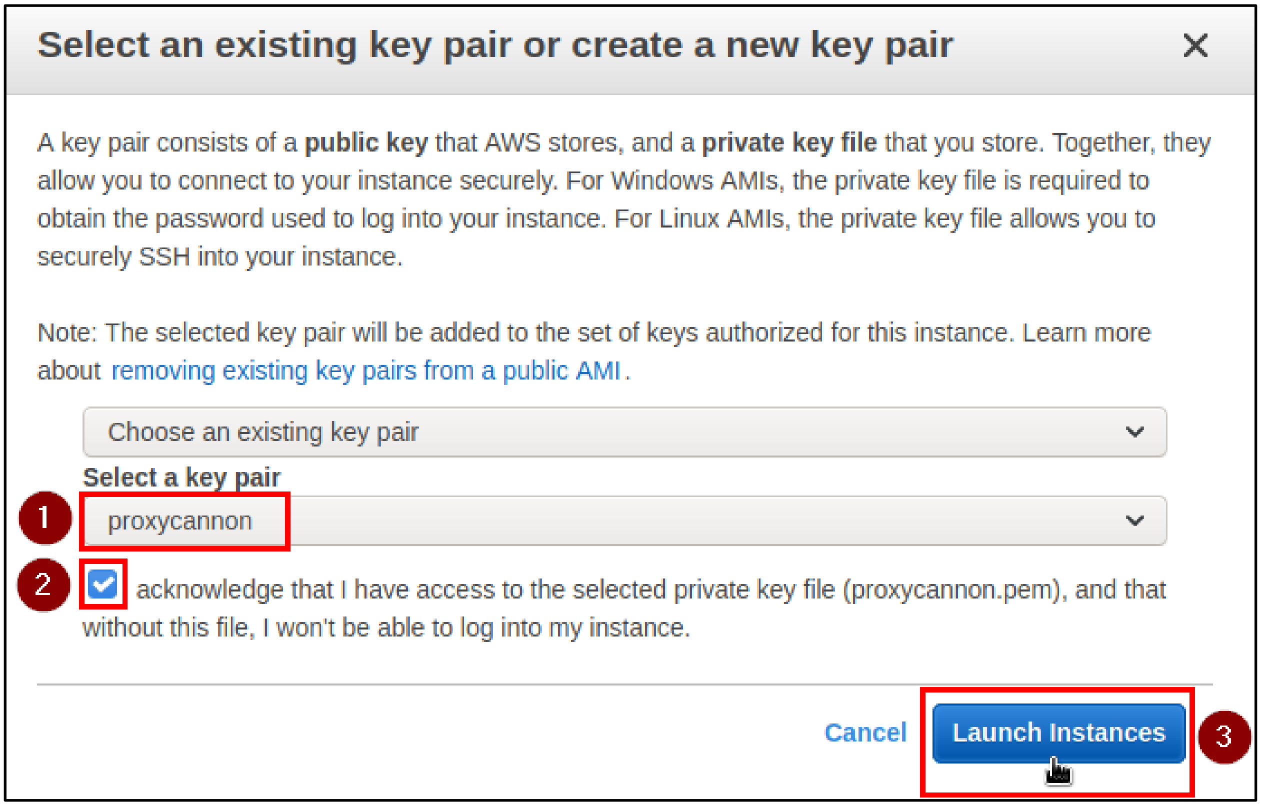
▸ Instance Details

▸ Storage

▸ Tags

Launching the Instance

12. A dialog box will appear that prompts you to select a key pair for the instance. Choose the "proxycannon" key pair that you created earlier, and then check the box to enable the "Launch Instances" button at the bottom of the form. Finally, click "Launch Instances" to launch the instance.



Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

Choose an existing key pair

Select a key pair

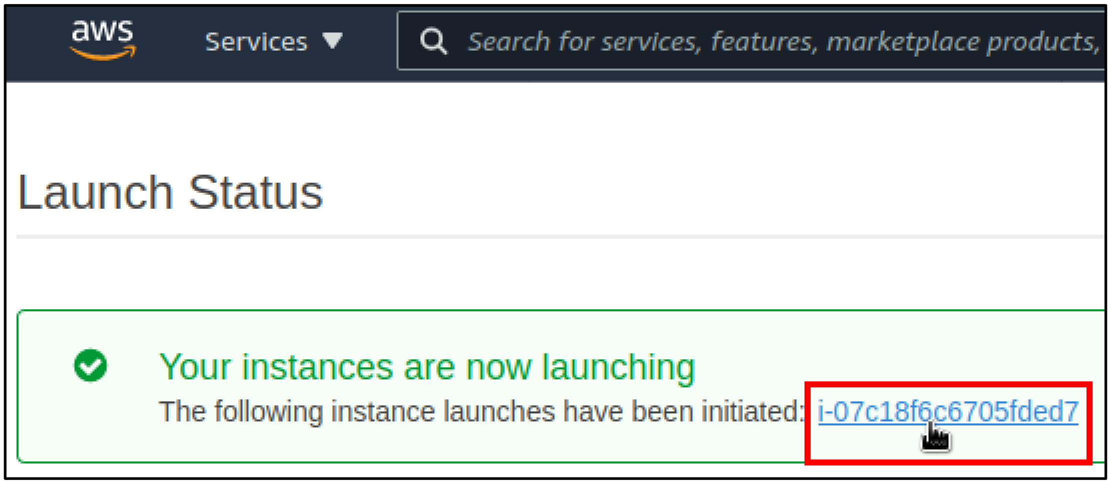
proxycannon

☑ I acknowledge that I have access to the selected private key file (proxycannon.pem), and that without this file, I won't be able to log into my instance.

Selecting the ProxyCannon Key Pair

3. Install ProxyCannon on the AWS instance

1. After launching the Control Server instance, you should see a Launch Status page with a link to your instance. Click on the instance ID to return to the AWS Instances page.



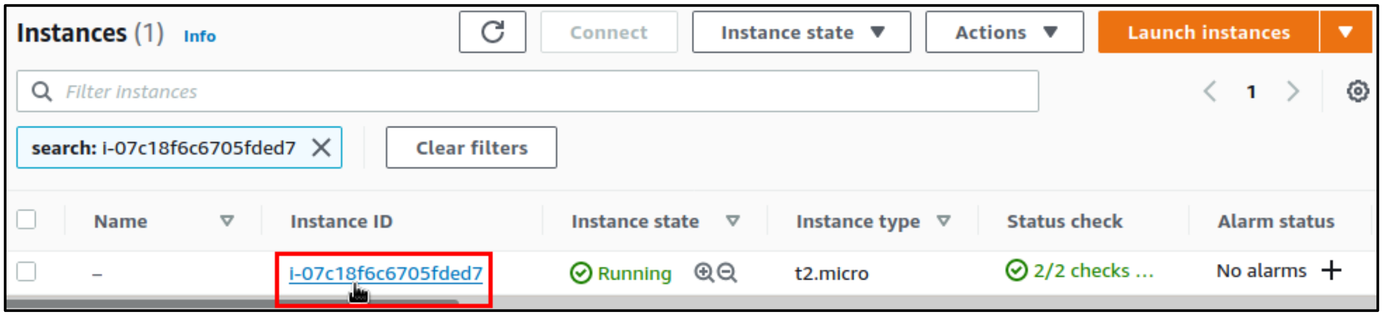
Launch Status

Your instances are now launching

The following instance launches have been initiated: i-07c18f6c6705fded7

Clicking the Instance ID

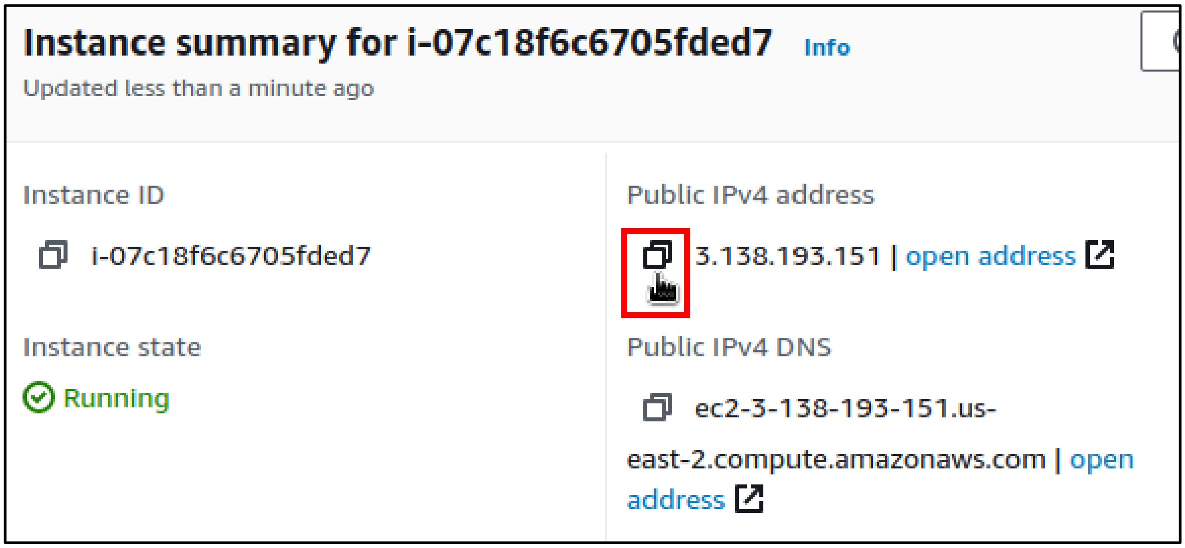
2. Click on the instance ID link again on in the Instances list to view the summary page for the instance.



□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
□	–	i-07c18f6c6705fded7	✓ Running	t2.micro	✓ 2/2 checks ...	No alarms +

Clicking the Instance ID Again

3. In your Instance Summary, you should see your AWS instance's IPv4 address under the heading "Public IPv4 address". In the next step, you can use the copy button on this page (two overlapping boxes) beside the IP address to copy your instance's public IP address to your clipboard.



Instance summary for i-07c18f6c6705fded7 Info

Updated less than a minute ago

Instance ID: i-07c18f6c6705fded7

Public IPv4 address: 3.138.193.151 | open address

Instance state: ✓ Running

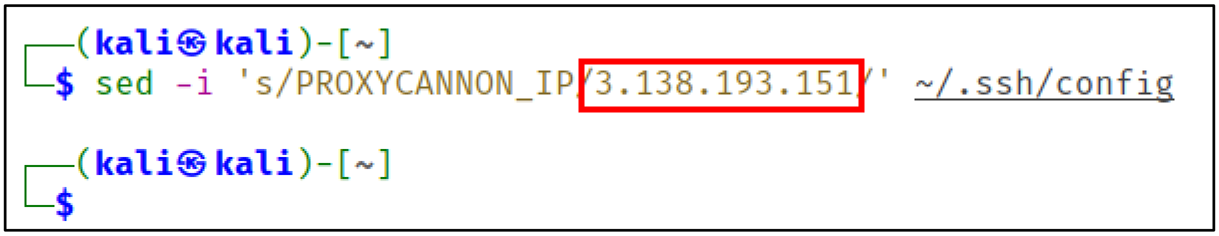
Public IPv4 DNS: ec2-3-138-193-151.us-east-2.compute.amazonaws.com | open address

Copying the Control Server's Public IP Address

4. For convenience while running the remaining steps, we'll configure the SSH client config file on your Kali VM to use "controlserver" as an alias for your control server's IP address. That way you won't have to type in the IP address every time you want to interact with the control server.

   First, replace all of the red text with the IP address of your AWS instance that you identified in the previous step. (It may help to copy and paste the command into a text editor so you can easily add your instance's IP address.) Then run the modified command in your terminal window.

```
sed -i 's/PROXYCANNON_IP/<PASTE INSTANCE IP HERE>/' ~/.ssh/config
```

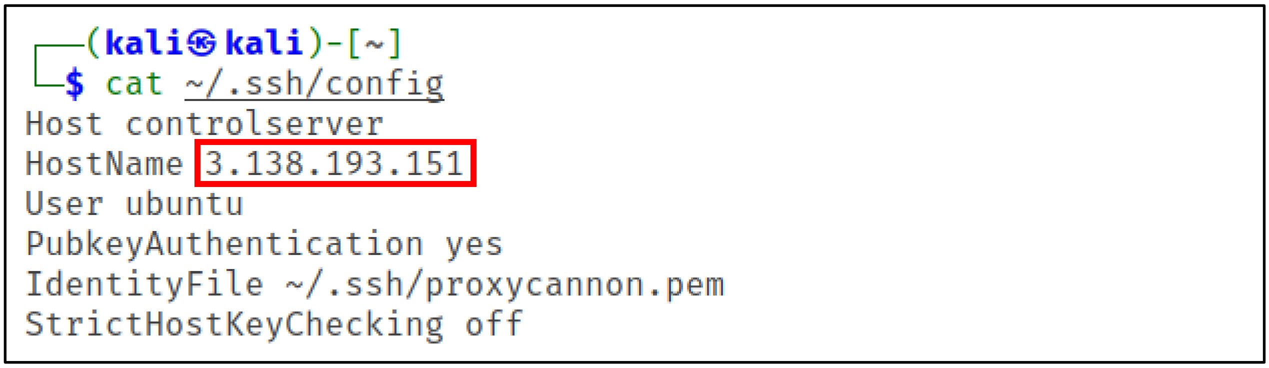


```
┌──(kali㉿kali)-[~]
└─$ sed -i 's/PROXYCANNON_IP/3.138.193.151/' ~/.ssh/config

┌──(kali㉿kali)-[~]
└─$
```

Command Executed with Instance IP Address in Place of Red Text

5. Now run the next command to confirm that the change was successful. You should see your instance's public IP address in the command output, as shown below.

```
cat ~/.ssh/config
```

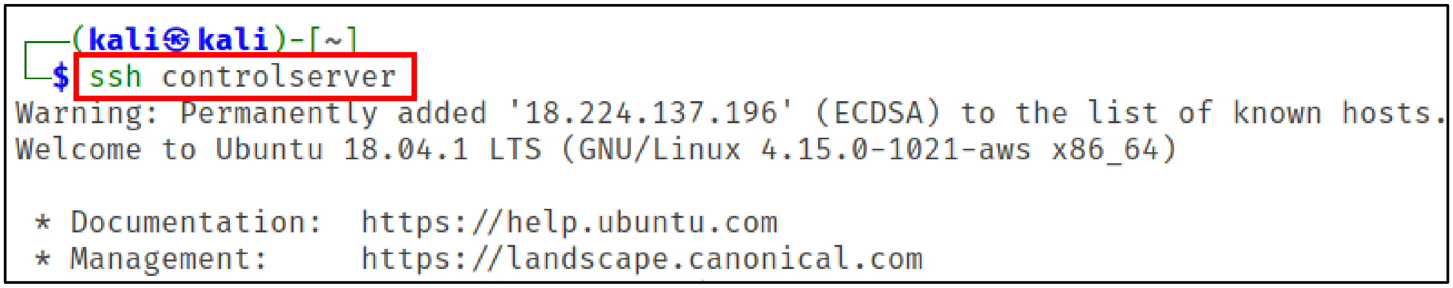


```
┌──(kali㉿kali)-[~]
└─$ cat ~/.ssh/config
Host controlserver
HostName 3.138.193.151
User ubuntu
PubkeyAuthentication yes
IdentityFile ~/.ssh/proxycannon.pem
StrictHostKeyChecking off
```

Instance IP Address Shown in Command Output

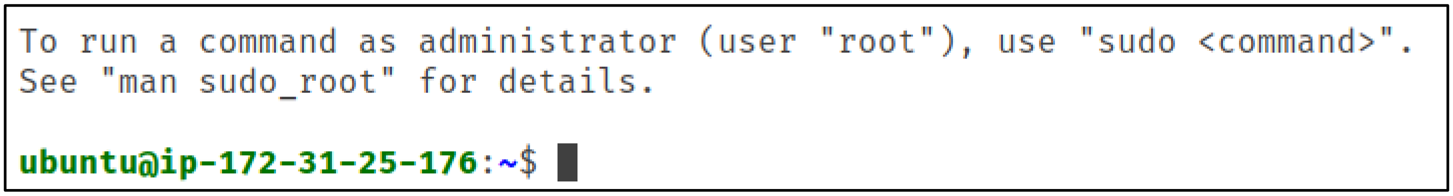
6. Log in to your control server with SSH by using the command below:

```
ssh controlserver
```



```
┌──(kali㉿kali)-[~]
└─$ ssh controlserver
Warning: Permanently added '18.224.137.196' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-1021-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
```

Execution of SSH Command

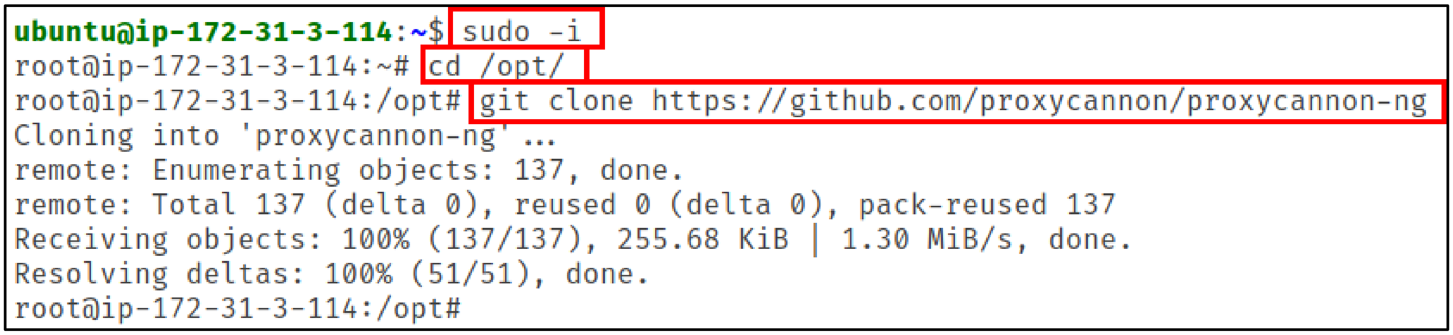


```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-25-176:~$
```

SSH Session Established on the ProxyCannon Control Server

7. After logging into your control server, run the following commands to download the ProxyCannon software.

```
sudo -i

cd /opt/

git clone https://github.com/proxycannon/proxycannon-ng
```



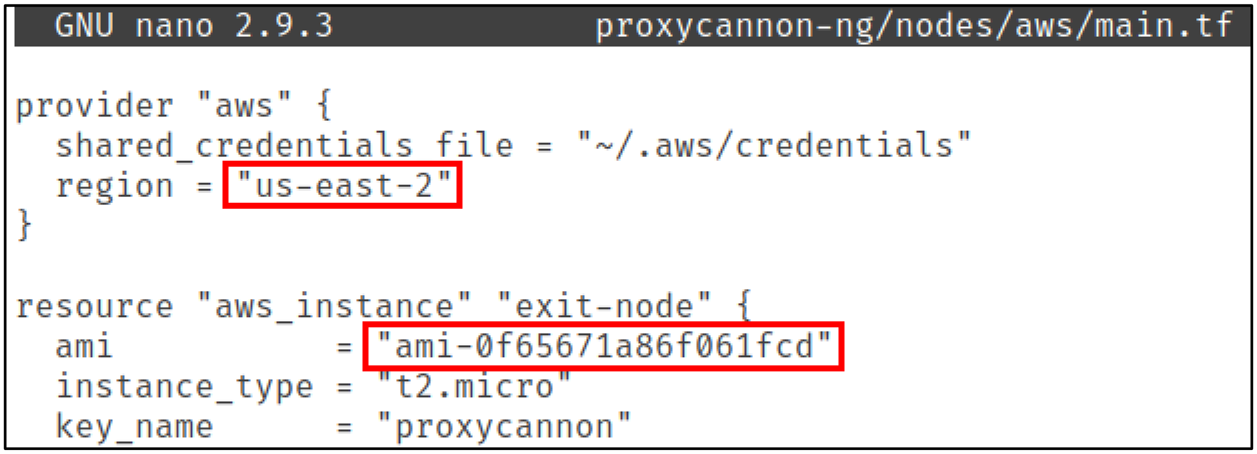
```
ubuntu@ip-172-31-3-114:~$ sudo -i
root@ip-172-31-3-114:~# cd /opt/
root@ip-172-31-3-114:/opt# git clone https://github.com/proxycannon/proxycannon-ng
Cloning into 'proxycannon-ng'...
remote: Enumerating objects: 137, done.
remote: Total 137 (delta 0), reused 0 (delta 0), pack-reused 137
Receiving objects: 100% (137/137), 255.68 KiB | 1.30 MiB/s, done.
Resolving deltas: 100% (51/51), done.
root@ip-172-31-3-114:/opt#
```

Execution of Commands Above

8. Next, use the command shown below to open ProxyCannon's "main.tf" file in the Nano text editor.

```
nano proxycannon-ng/nodes/aws/main.tf
```

9. Replace the "region" and "ami" values in the file with your own region and AMI identifiers that you recorded earlier.



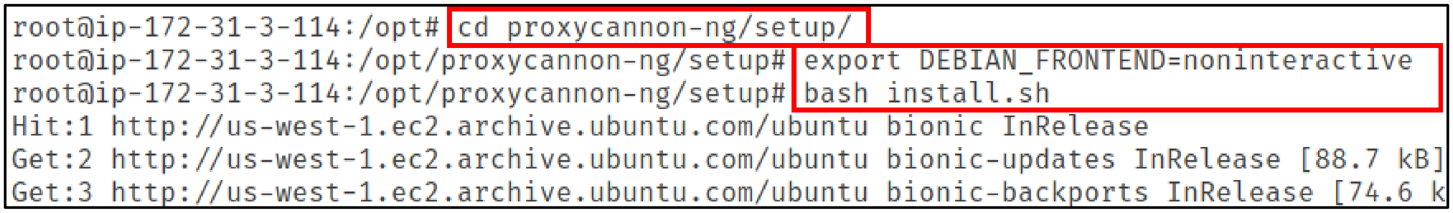
```
  GNU nano 2.9.3                 proxycannon-ng/nodes/aws/main.tf

provider "aws" {
  shared_credentials_file = "~/.aws/credentials"
  region = "us-east-2"
}

resource "aws_instance" "exit-node" {
  ami           = "ami-0f65671a86f061fcd"
  instance_type = "t2.micro"
  key_name      = "proxycannon"
```

Replace These Values with Your Own

10. After you have made your changes to the file, press CTRL+O, followed by Enter to save the file. Then press CTRL+X to exit the Nano text editor.

11. Now run the following commands to install the ProxyCannon software.

```
cd proxycannon-ng/setup

export DEBIAN_FRONTEND=noninteractive

bash install.sh
```

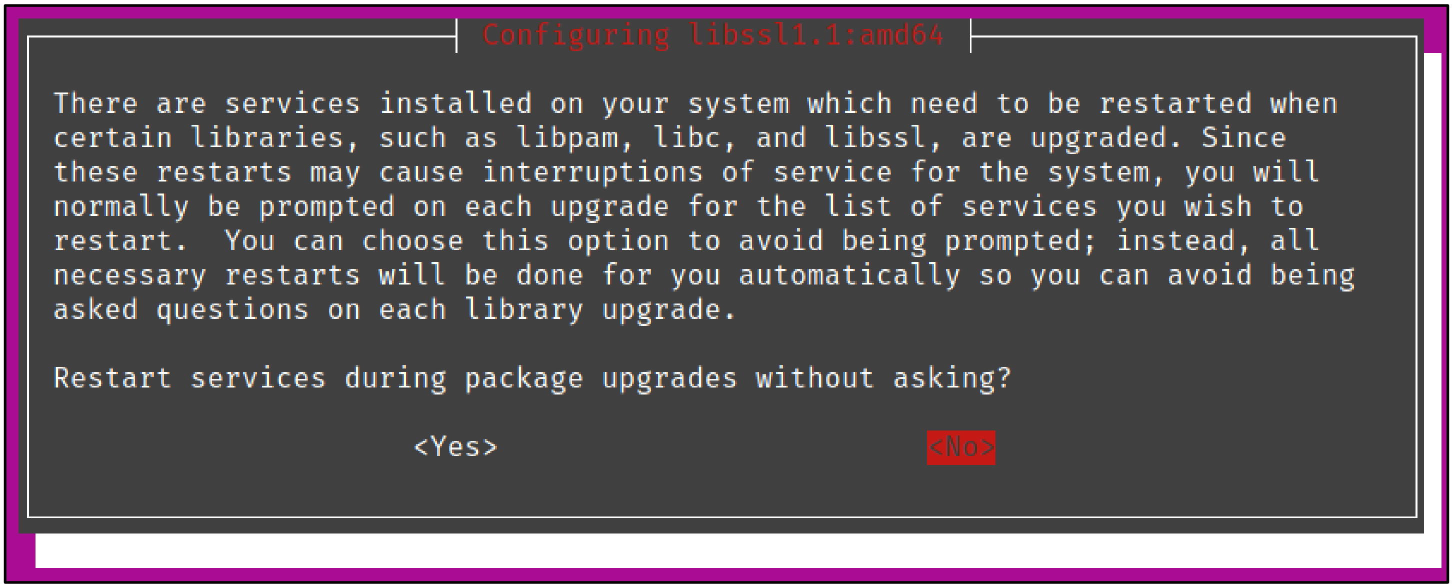


```
root@ip-172-31-3-114:/opt# cd proxycannon-ng/setup/
root@ip-172-31-3-114:/opt/proxycannon-ng/setup# export DEBIAN_FRONTEND=noninteractive
root@ip-172-31-3-114:/opt/proxycannon-ng/setup# bash install.sh
Hit:1 http://us-west-1.ec2.archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://us-west-1.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:3 http://us-west-1.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 k
```

Execution of Commands Above

12. If any prompts appear while ProxyCannon installs, you can safely accept the default settings displayed in each prompt by pressing the Enter key.

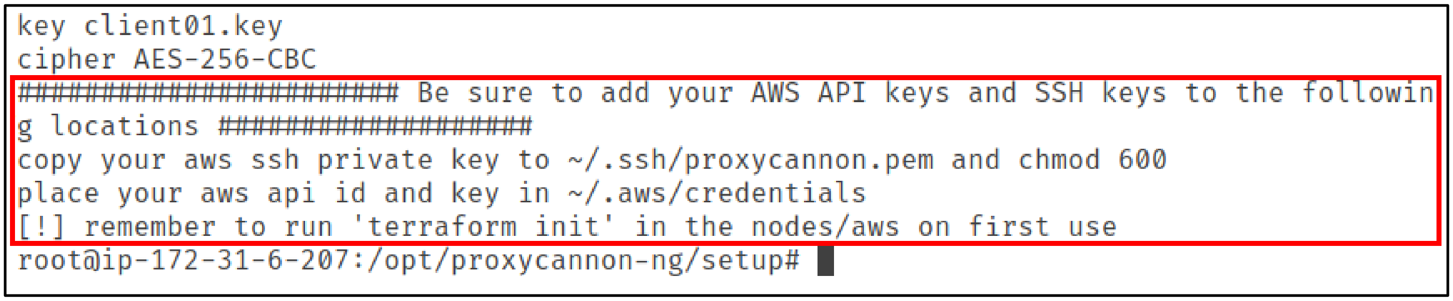
    The "export DEBIAN_FRONTEND=noninteractive" command executed in the previous step should make the default settings get accepted without prompting, but this note is included here just in case. If you don't see any prompts appear during installation, you can safely continue to the next step below.



```
Configuring libssl1.1:amd64

There are services installed on your system which need to be restarted when
certain libraries, such as libpam, libc, and libssl, are upgraded. Since
these restarts may cause interruptions of service for the system, you will
normally be prompted on each upgrade for the list of services you wish to
restart.  You can choose this option to avoid being prompted; instead, all
necessary restarts will be done for you automatically so you can avoid being
asked questions on each library upgrade.

Restart services during package upgrades without asking?
```

Default Setting Accepted by Pressing Enter

13. You should see output like that shown below when the installation is complete.

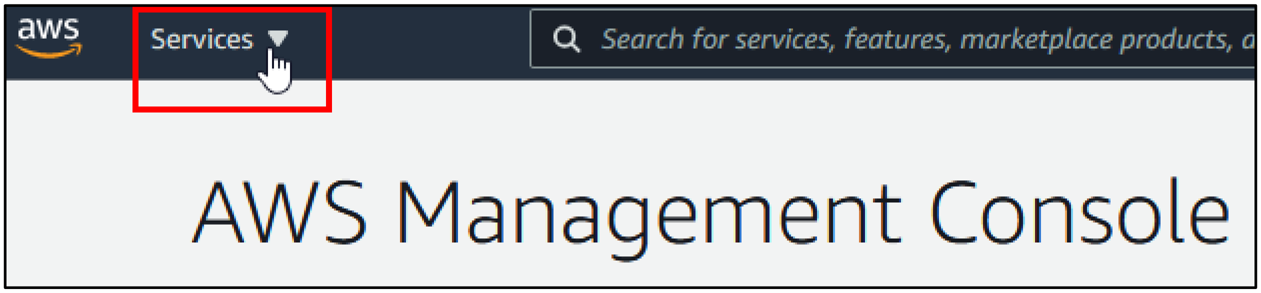


```
key client01.key
cipher AES-256-CBC
####################### Be sure to add your AWS API keys and SSH keys to the followin
g locations ###################
copy your aws ssh private key to ~/.ssh/proxycannon.pem and chmod 600
place your aws api id and key in ~/.aws/credentials
[!] remember to run 'terraform init' in the nodes/aws on first use
root@ip-172-31-6-207:/opt/proxycannon-ng/setup# 
```

ProxyCannon Installation Complete

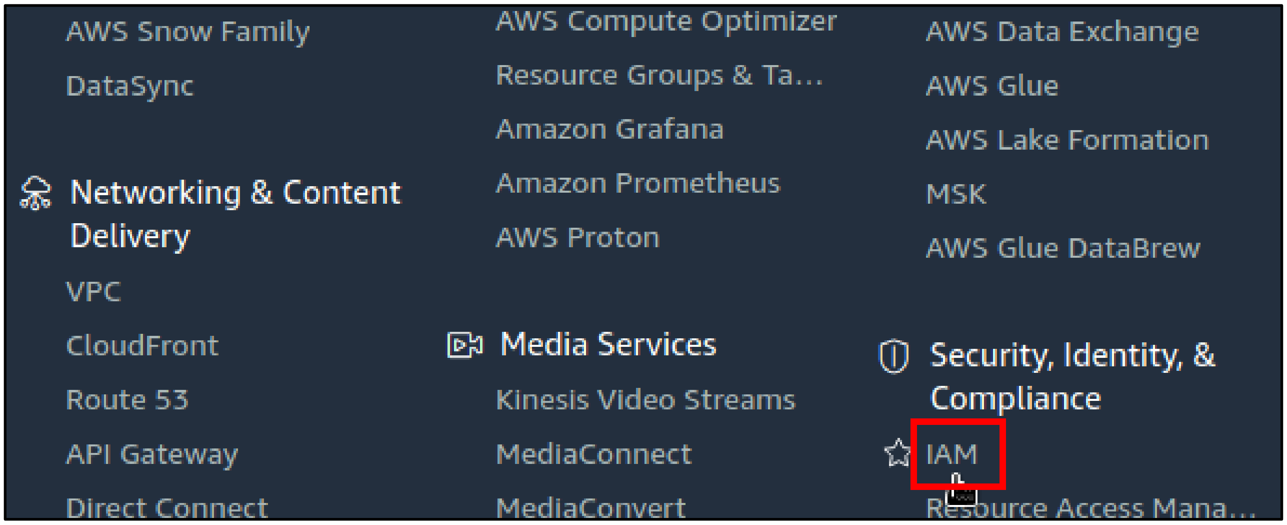
4. Configure AWS for programmatic access by ProxyCannon

1. In the AWS Management Console, click on the “Services” menu in the upper left.



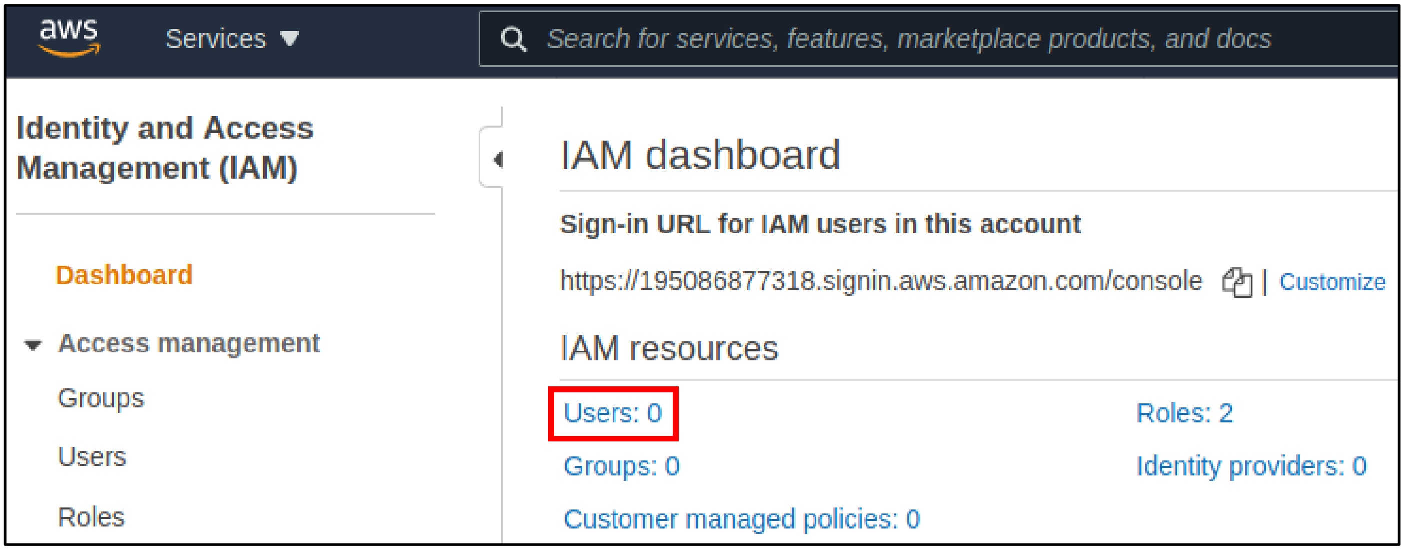
Services Menu

2. Scroll down in the menu and click on "IAM" under the "Security, Identity & Compliance" heading.



IAM Link

3. Click on "Users" under the "IAM resources" heading in the IAM dashboard.


aws
Services ▼
Search for services, features, marketplace products, and docs
Identity and Access Management (IAM)
Dashboard
Access management
Groups
Users
Roles
IAM dashboard

Sign-in URL for IAM users in this account

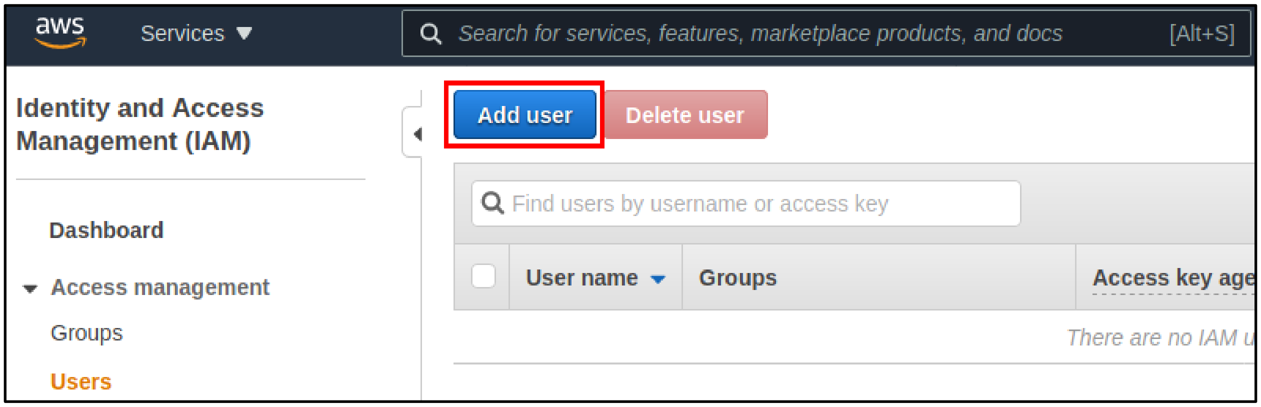
https://195086877318.signin.aws.amazon.com/console | Customize

IAM resources

Users: 0	Roles: 2
Groups: 0	Identity providers: 0
Customer managed policies: 0	

Users Link

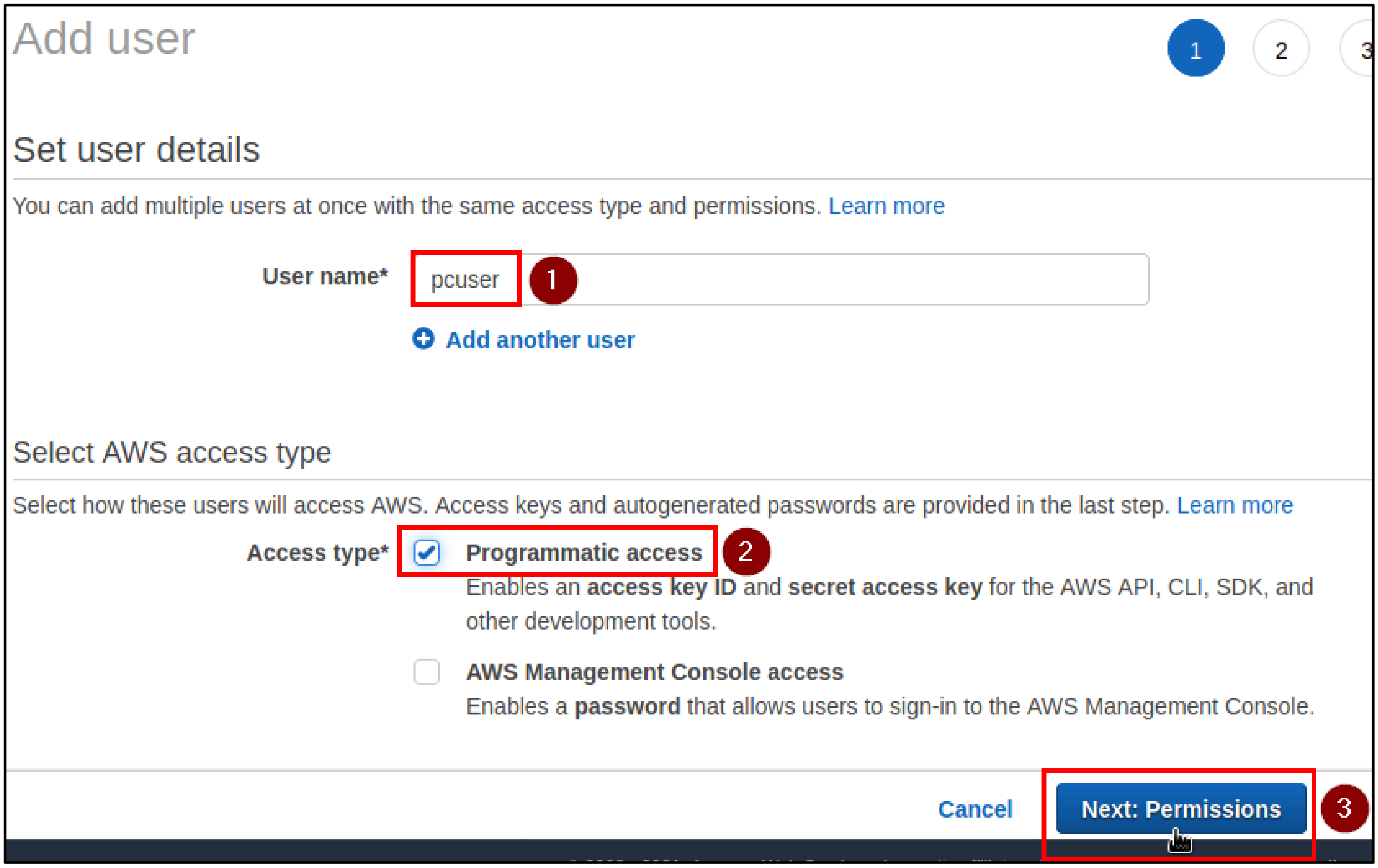
4. Click the "Add user" button.


aws
Services ▼
Search for services, features, marketplace products, and docs
[Alt+S]
Identity and Access Management (IAM)
Dashboard
Access management
Groups
Users
Add user
Delete user
Find users by username or access key
User name
Groups
Access key age

There are no IAM u

Add User Button

5. In the "User name" field, enter "pcuser", and choose "Programmatic access" as the Access type. Then click the "Next: Permissions" button at the bottom of the page.



Set user details

You can add multiple users at once with the same access type and permissions. Learn more

User name* pcuser (1)

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. Learn more

Access type* ☑ **Programmatic access** (2)
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

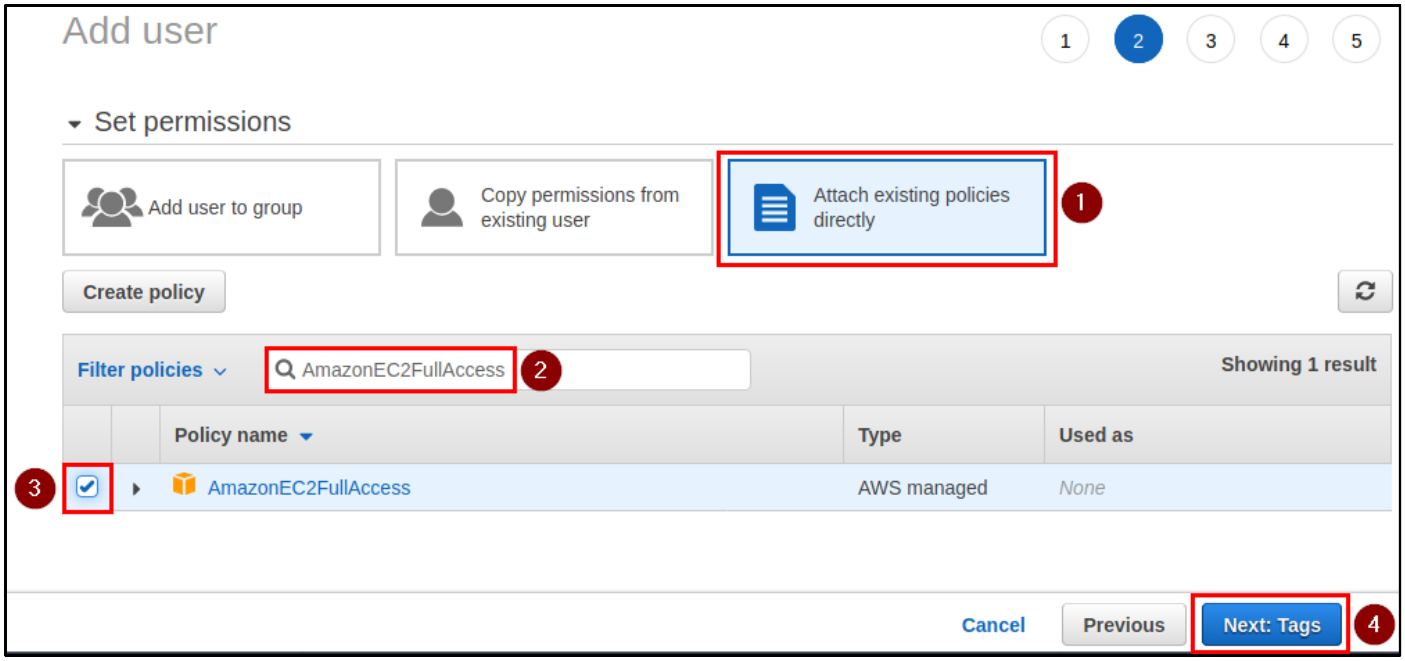
□ **AWS Management Console access**
Enables a **password** that allows users to sign-in to the AWS Management Console.

(3) Next: Permissions

Configuration of the "pcuser" User Account

6. Under the "Set permissions" heading, click the box labeled, "Attach existing policies directly". Then use the search box to search for "AmazonEC2FullAccess". Check the box beside "AmazonEC2FullAccess" in the search results, and click "Next: Tags" at the bottom of the page.

```
AmazonEC2FullAccess
```



1 Attach existing policies directly

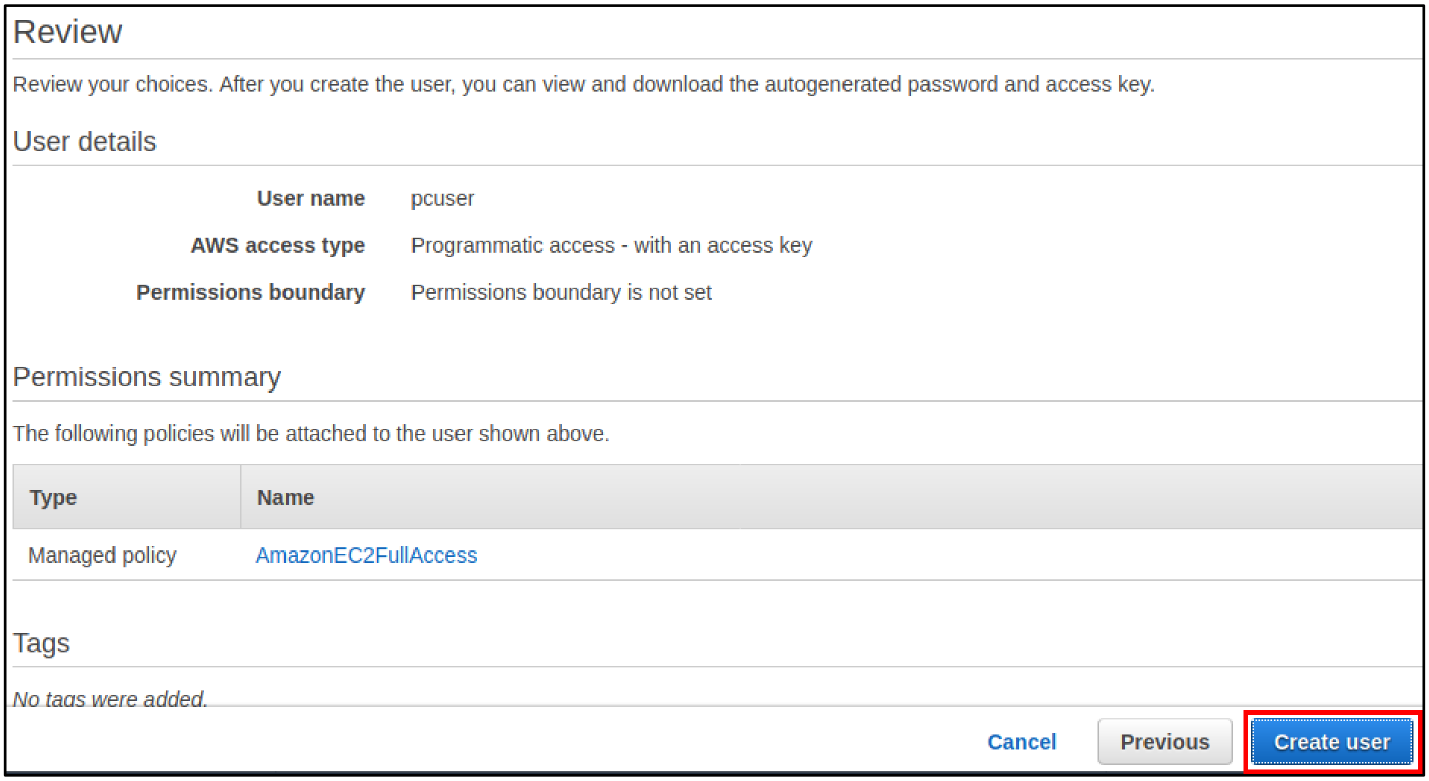
2 AmazonEC2FullAccess

		Policy name	Type	Used as
3 ☑	▸	AmazonEC2FullAccess	AWS managed	*None*

4 Next: Tags

Configuring "pcuser" Permissions

7. Continue clicking Next through the remaining pages, and then click "Create User" on the Review page.



Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

User details

User name	pcuser
AWS access type	Programmatic access - with an access key
Permissions boundary	Permissions boundary is not set

Permissions summary

The following policies will be attached to the user shown above.

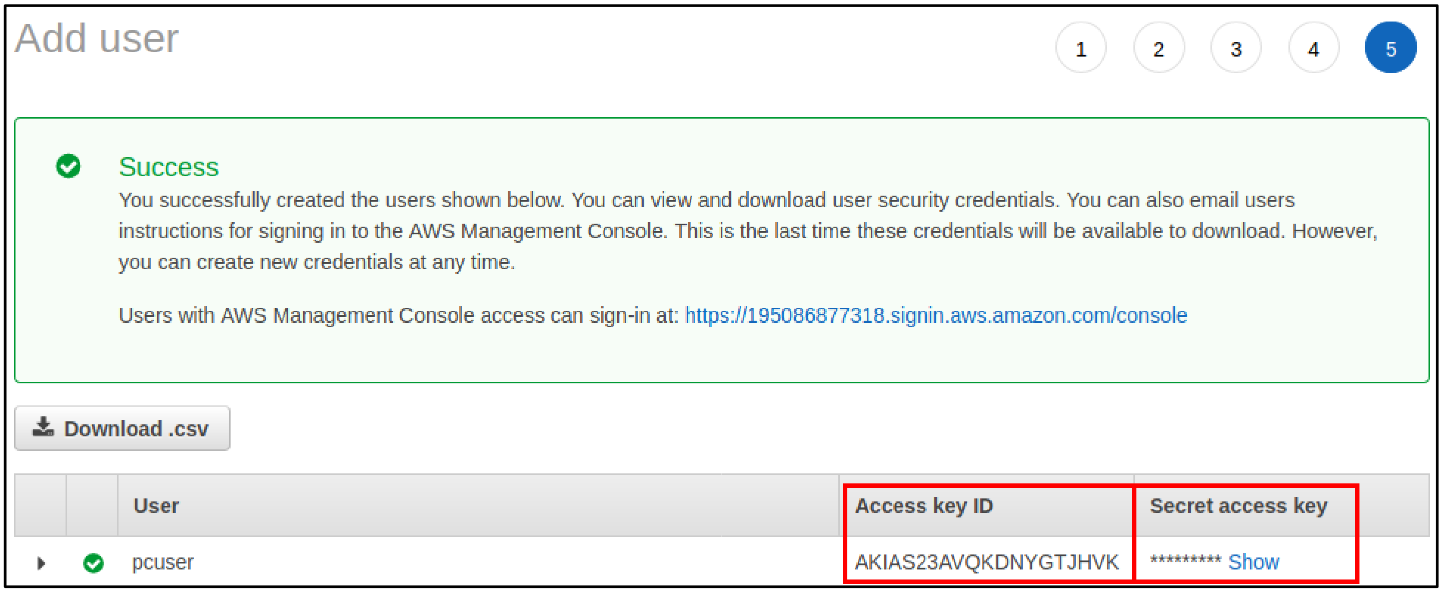
Type	Name
Managed policy	AmazonEC2FullAccess

Tags

No tags were added.

Creating the New User

8. On the next page, click the "Show" link under the "Secret access key" column for your new user. Confirm that the user, access key ID, and secret access key ID are all present. Then keep this page open in your web browser, since the access key ID and secret access key will be needed in the next steps.



Add user

Success

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

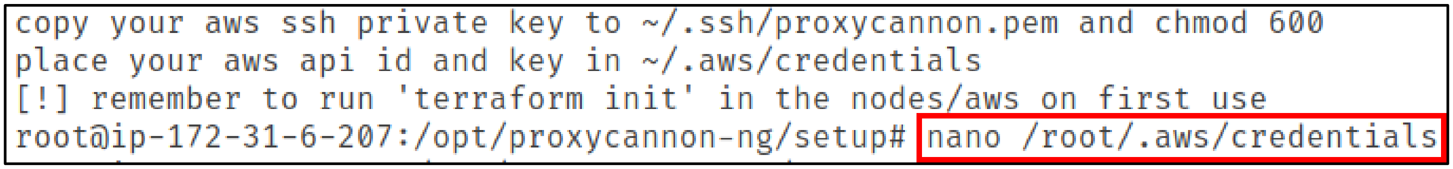
Users with AWS Management Console access can sign-in at: https://195086877318.signin.aws.amazon.com/console

		User	Access key ID	Secret access key
▸	✓	pcuser	AKIAS23AVQKDNYGTJHVK	********* Show

Access Key ID and Secret Access Key Displayed

9. In your SSH session on the Control Sever, run the following command to edit the /root/.aws/credentials file.

```
nano /root/.aws/credentials
```

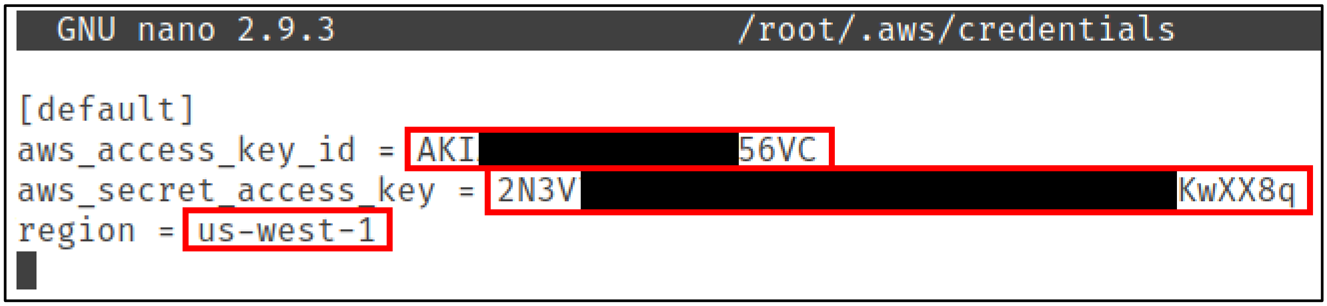


```
copy your aws ssh private key to ~/.ssh/proxycannon.pem and chmod 600
place your aws api id and key in ~/.aws/credentials
[!] remember to run 'terraform init' in the nodes/aws on first use
root@ip-172-31-6-207:/opt/proxycannon-ng/setup# nano /root/.aws/credentials
```

Execution of the Nano Command

10. Copy and paste the lines below into the credentials file that you opened with the Nano command above. Then replace "REPLACE_WITH_YOUR_OWN" with your access key ID and your secret access key that are displayed in your web browser. Also fill in your region identifier on the third line.

```
[default]
aws_access_key_id = REPLACE_WITH_YOUR_OWN
aws_secret_access_key = REPLACE_WITH_YOUR_OWN
region = REPLACE_WITH_YOUR_OWN
```



```
  GNU nano 2.9.3                    /root/.aws/credentials

[default]
aws_access_key_id = AKI[redacted]56VC
aws_secret_access_key = 2N3V[redacted]KwXX8q
region = us-west-1
```

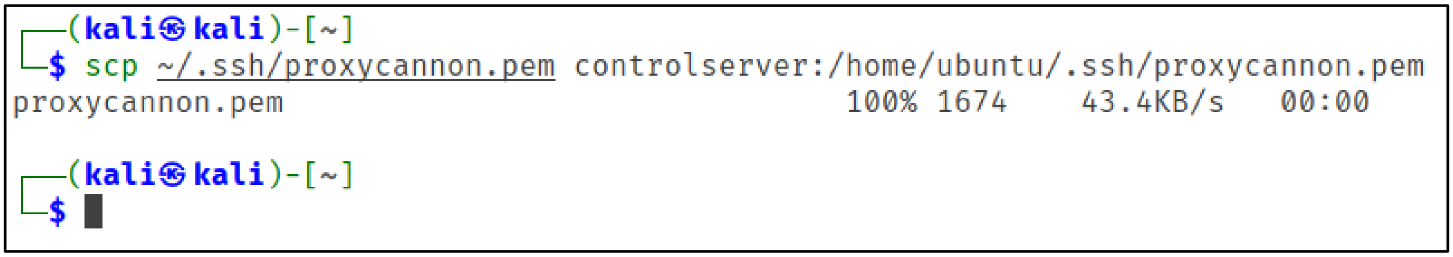
AWS Access Key ID and Secret Access Key Inserted into Credentials File

11. After you have made your changes to the file, press CTRL+O, followed by Enter to save your file. Then press CTRL+X to exit the Nano text editor.

5. Setup Terraform on the ProxyCannon control server

1. Open a new Terminal window on your Kali VM, and use the SCP command below to copy your proxycannon.pem SSH key to the Control Server.

```
scp ~/.ssh/proxycannon.pem controlserver:/home/ubuntu/.ssh/proxycannon.pem
```

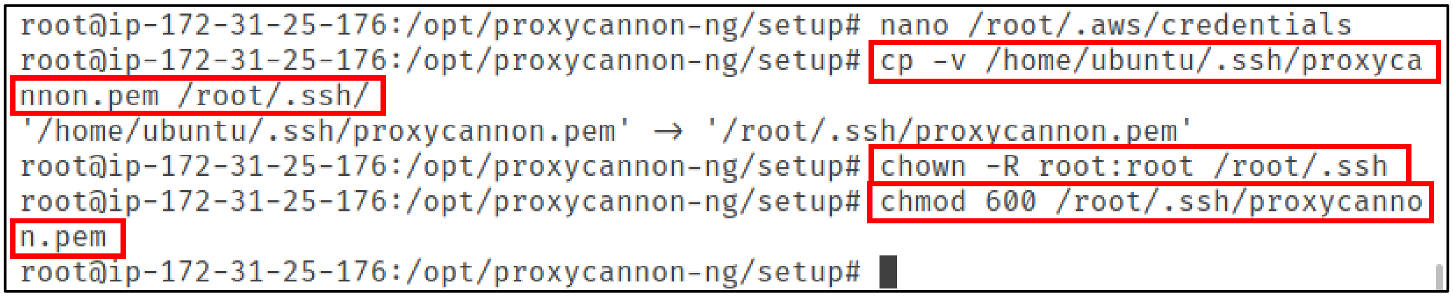


```
┌──(kali㉿kali)-[~]
└─$ scp ~/.ssh/proxycannon.pem controlserver:/home/ubuntu/.ssh/proxycannon.pem
proxycannon.pem                                      100% 1674    43.4KB/s   00:00

┌──(kali㉿kali)-[~]
└─$ 
```

ProxyCannon SSH Key Copied to the Control Server

2. In your SSH session on the Control Server, copy the SSH key to the root user's ".ssh" directory, and confirm that the file has the proper permissions.

```
cp -v /home/ubuntu/.ssh/proxycannon.pem /root/.ssh/

chown -R root:root /root/.ssh

chmod 600 /root/.ssh/proxycannon.pem
```

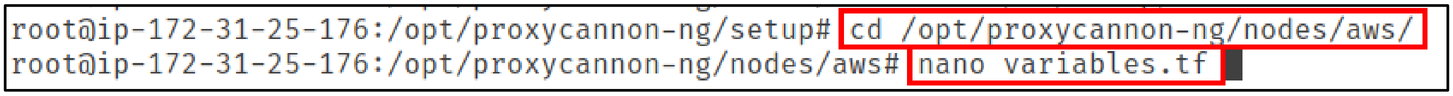


```
root@ip-172-31-25-176:/opt/proxycannon-ng/setup# nano /root/.aws/credentials
root@ip-172-31-25-176:/opt/proxycannon-ng/setup# cp -v /home/ubuntu/.ssh/proxyca
nnon.pem /root/.ssh/
'/home/ubuntu/.ssh/proxycannon.pem' → '/root/.ssh/proxycannon.pem'
root@ip-172-31-25-176:/opt/proxycannon-ng/setup# chown -R root:root /root/.ssh
root@ip-172-31-25-176:/opt/proxycannon-ng/setup# chmod 600 /root/.ssh/proxycanno
n.pem
root@ip-172-31-25-176:/opt/proxycannon-ng/setup# 
```

Execution of the Commands Above

3. Use the commands below to open the "variables.tf" file in the Nano text editor. Then leave the text editor running in your terminal window.

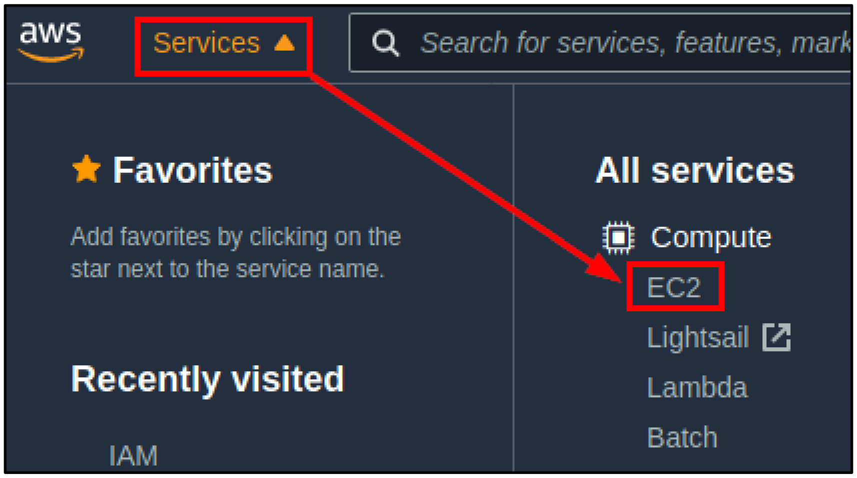
```
cd /opt/proxycannon-ng/nodes/aws

nano variables.tf
```



```
root@ip-172-31-25-176:/opt/proxycannon-ng/setup# cd /opt/proxycannon-ng/nodes/aws/
root@ip-172-31-25-176:/opt/proxycannon-ng/nodes/aws# nano variables.tf
```

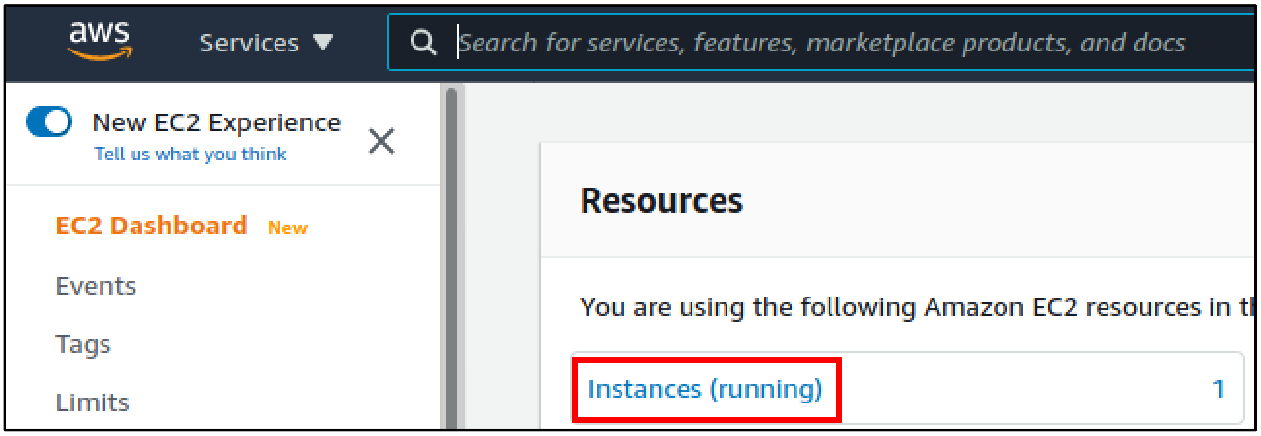
Opening "variables.tf" with Nano

4. In your web browser, return to the EC2 page in AWS by clicking on the Services menu and clicking the EC2 link.


aws
Services ▲
Search for services, features, mark
Favorites
Add favorites by clicking on the star next to the service name.
Recently visited
IAM
All services
Compute
EC2
Lightsail
Lambda
Batch

EC2 Link in the Services Menu

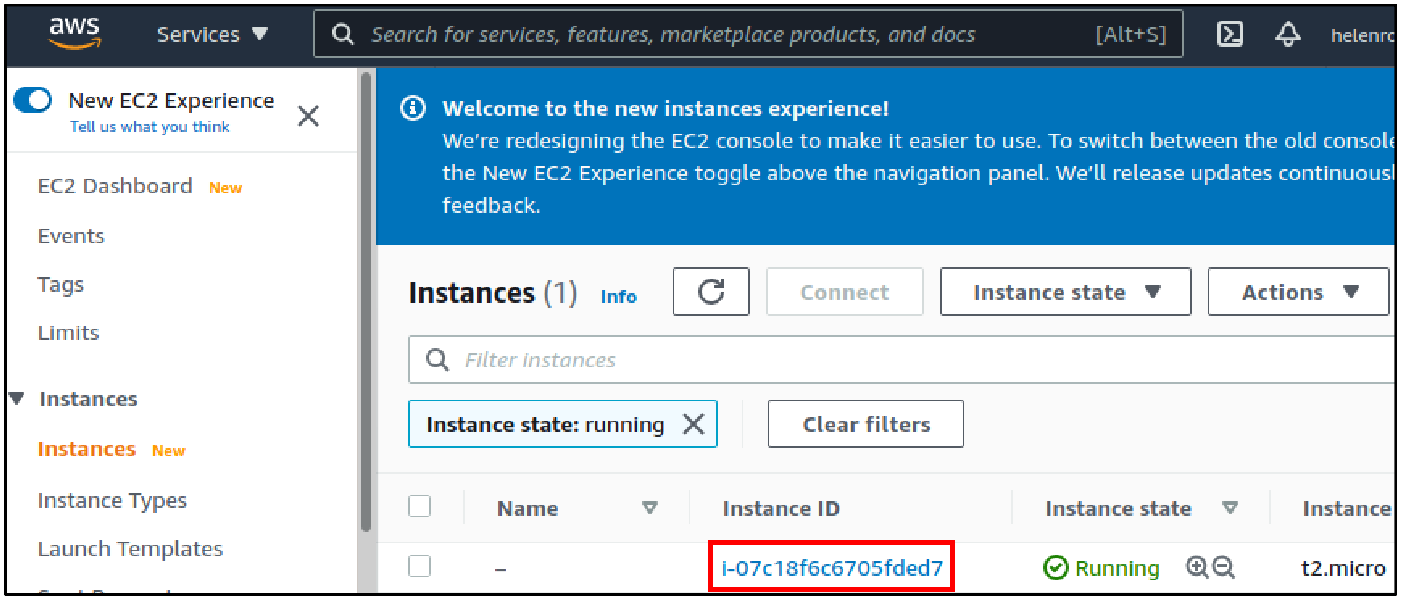
5. Then click on "Instances" under the Resources heading on the EC2 page.



You are using the following Amazon EC2 resources in tl

Instances Link on the EC2 Page

6. Click on the instance ID of your ProxyCannon server on the Instances page.

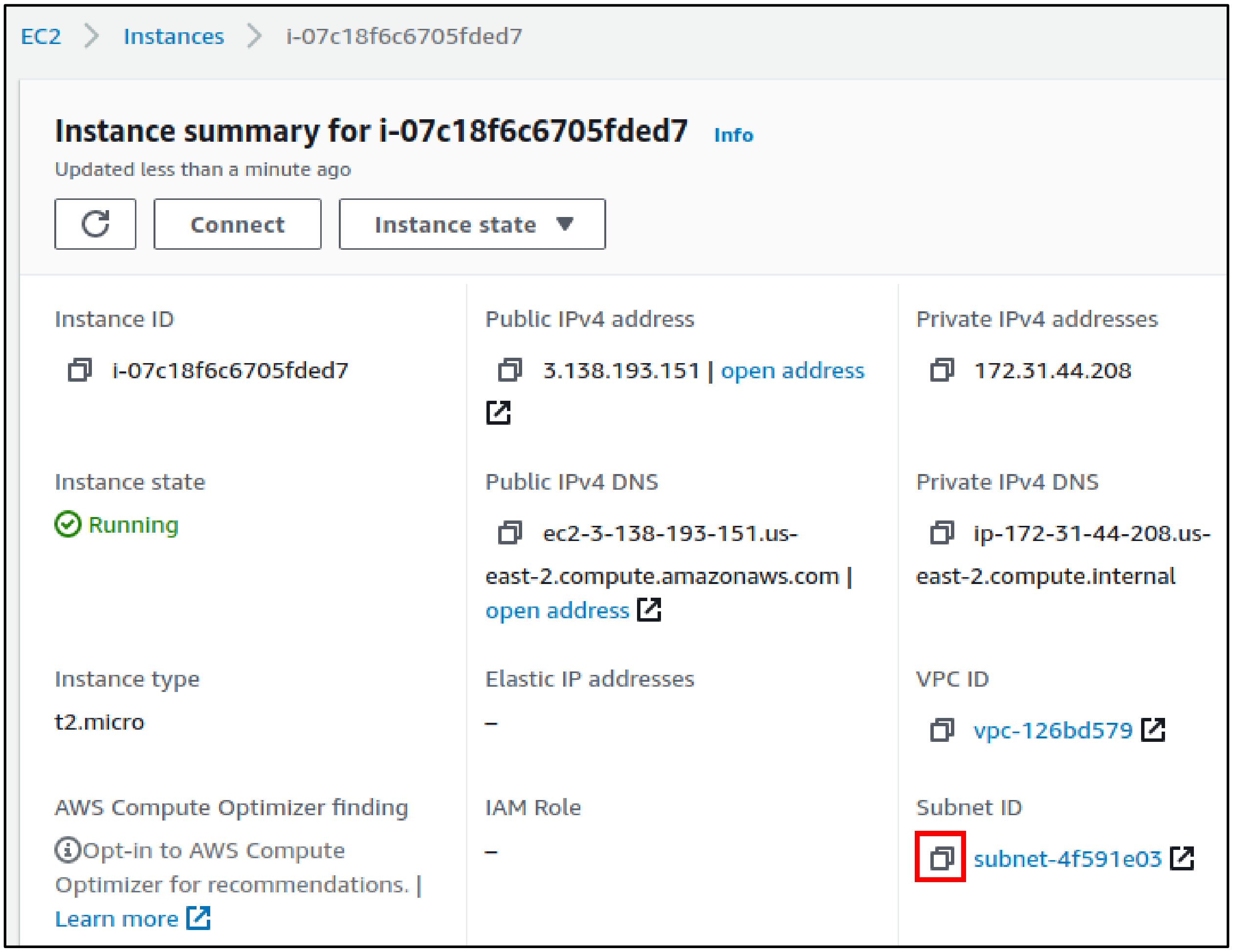


Welcome to the new instances experience!
We're redesigning the EC2 console to make it easier to use. To switch between the old console the New EC2 Experience toggle above the navigation panel. We'll release updates continuousl feedback.

□	Name	Instance ID	Instance state	Instance
□	–	i-07c18f6c6705fded7	Running	t2.micro

Instance ID Link on the Instances Page

7. Click the copy button under the Subnet ID heading on the page to copy the Subnet ID that is shown.



Instance summary for i-07c18f6c6705fded7

Updated less than a minute ago

Instance ID: i-07c18f6c6705fded7

Public IPv4 address: 3.138.193.151 | open address

Private IPv4 addresses: 172.31.44.208

Instance state: Running

Public IPv4 DNS: ec2-3-138-193-151.us-east-2.compute.amazonaws.com | open address

Private IPv4 DNS: ip-172-31-44-208.us-east-2.compute.internal

Instance type: t2.micro

Elastic IP addresses: –

VPC ID: vpc-126bd579

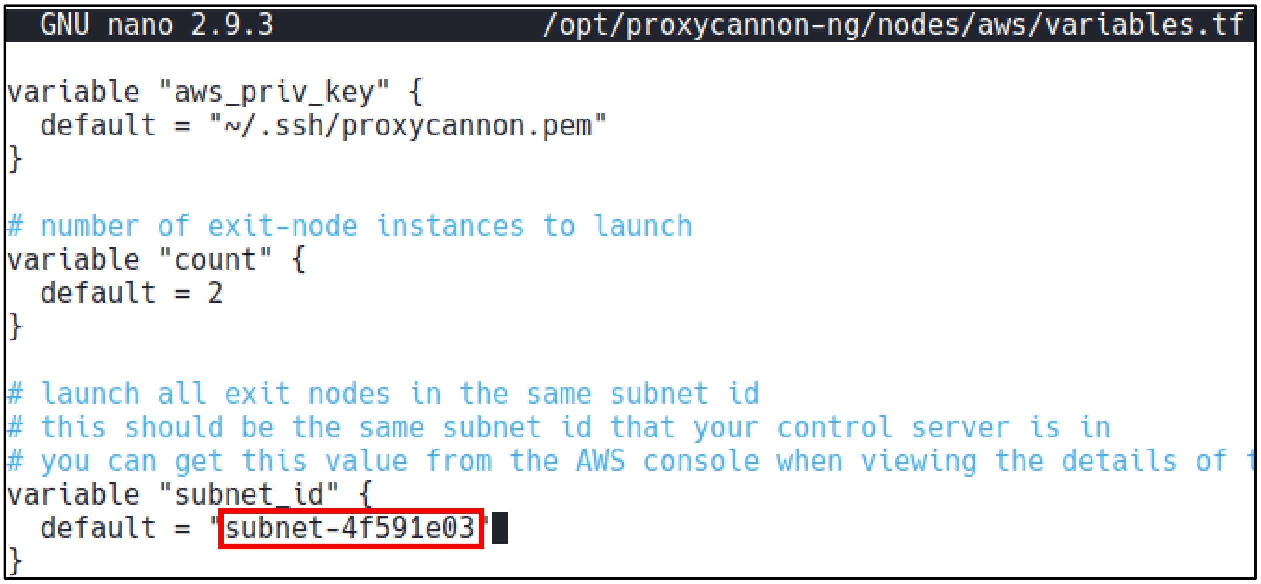
AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations. | Learn more

IAM Role: –

Subnet ID: subnet-4f591e03

Copy Button for the Subnet ID

8. Replace "subnet-XXXXXXXX" inside the variables.tf file that is open in your terminal window with the subnet ID you copied from the AWS Management Console. Then press CTRL+O followed by Enter and CTRL+X to save and exit.

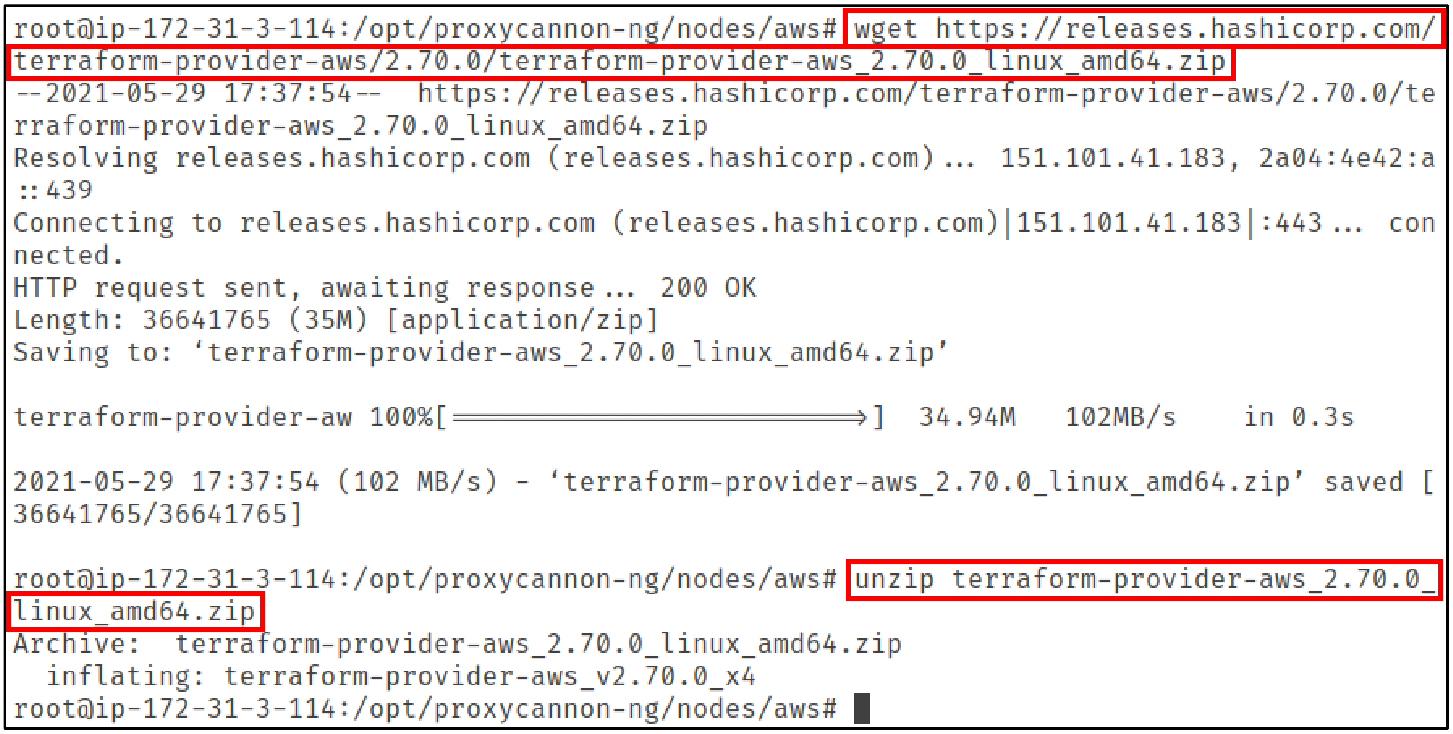


```
  GNU nano 2.9.3                /opt/proxycannon-ng/nodes/aws/variables.tf

variable "aws_priv_key" {
  default = "~/.ssh/proxycannon.pem"
}

# number of exit-node instances to launch
variable "count" {
  default = 2
}

# launch all exit nodes in the same subnet id
# this should be the same subnet id that your control server is in
# you can get this value from the AWS console when viewing the details of t
variable "subnet_id" {
  default = "subnet-4f591e03"
}
```

Placeholder Text Replaced with Actual Subnet ID

9. Run the commands below to download and extract the Terraform AWS provider files. Be sure there are no line breaks in the URL shown in the first command when copying an pasting the text below.

```
wget https://releases.hashicorp.com/terraform-provider-aws/2.70.0/terraform-
provider-aws_2.70.0_linux_amd64.zip

unzip terraform-provider-aws_2.70.0_linux_amd64.zip
```

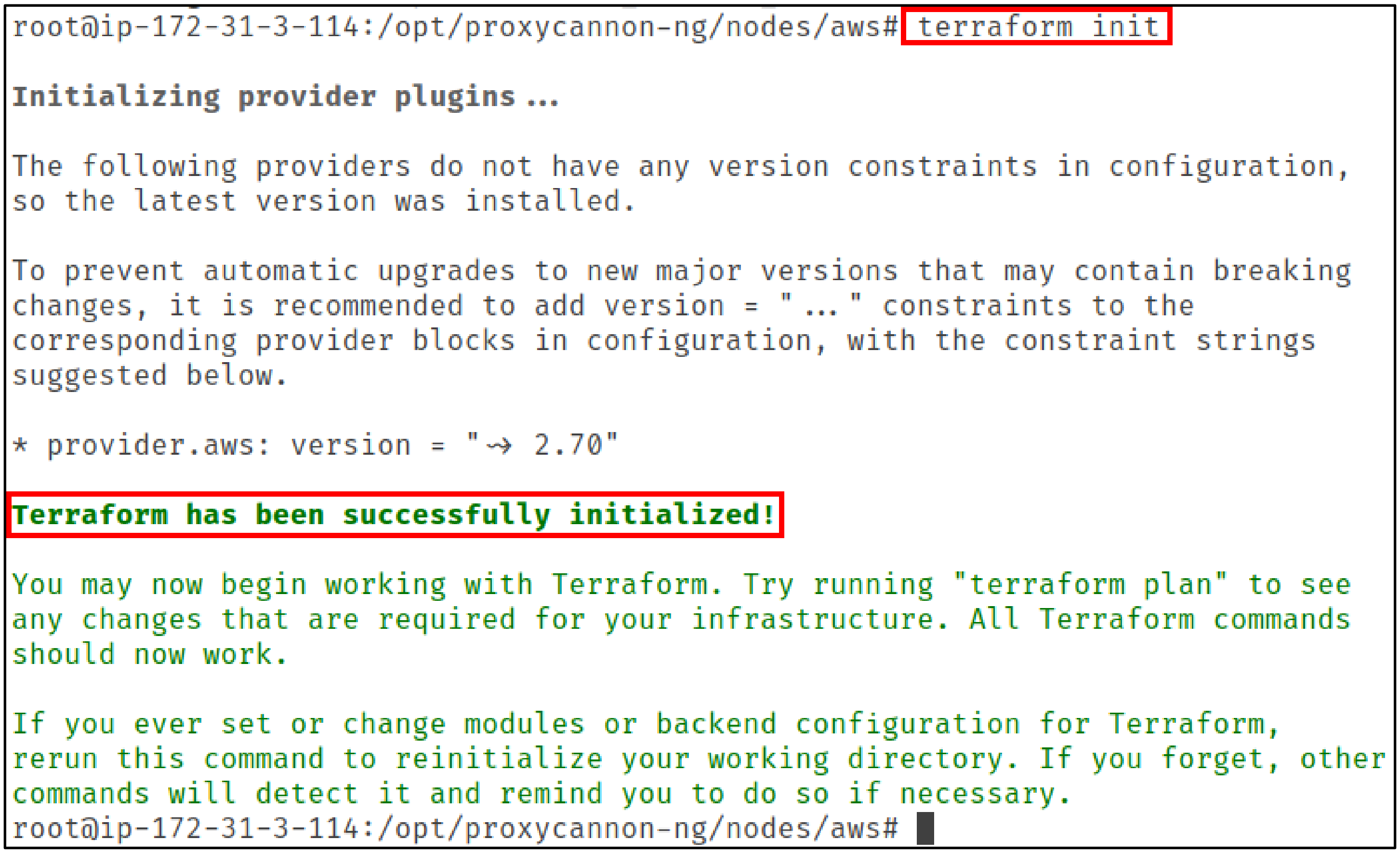


```
root@ip-172-31-3-114:/opt/proxycannon-ng/nodes/aws# wget https://releases.hashicorp.com/
terraform-provider-aws/2.70.0/terraform-provider-aws_2.70.0_linux_amd64.zip
--2021-05-29 17:37:54--  https://releases.hashicorp.com/terraform-provider-aws/2.70.0/te
rraform-provider-aws_2.70.0_linux_amd64.zip
Resolving releases.hashicorp.com (releases.hashicorp.com)... 151.101.41.183, 2a04:4e42:a
::439
Connecting to releases.hashicorp.com (releases.hashicorp.com)|151.101.41.183|:443... con
nected.
HTTP request sent, awaiting response... 200 OK
Length: 36641765 (35M) [application/zip]
Saving to: ‘terraform-provider-aws_2.70.0_linux_amd64.zip’

terraform-provider-aw 100%[==========================>]  34.94M   102MB/s    in 0.3s

2021-05-29 17:37:54 (102 MB/s) - ‘terraform-provider-aws_2.70.0_linux_amd64.zip’ saved [
36641765/36641765]

root@ip-172-31-3-114:/opt/proxycannon-ng/nodes/aws# unzip terraform-provider-aws_2.70.0_
linux_amd64.zip
Archive:  terraform-provider-aws_2.70.0_linux_amd64.zip
  inflating: terraform-provider-aws_v2.70.0_x4
root@ip-172-31-3-114:/opt/proxycannon-ng/nodes/aws# 
```

Download and Extraction of AWS Terraform Files

10. Next, run "terraform init" to initialize the provider plugins on your ProxyCannon control server. You should then receive a message stating that Terraform has been successfully initialized.

```
terraform init
```



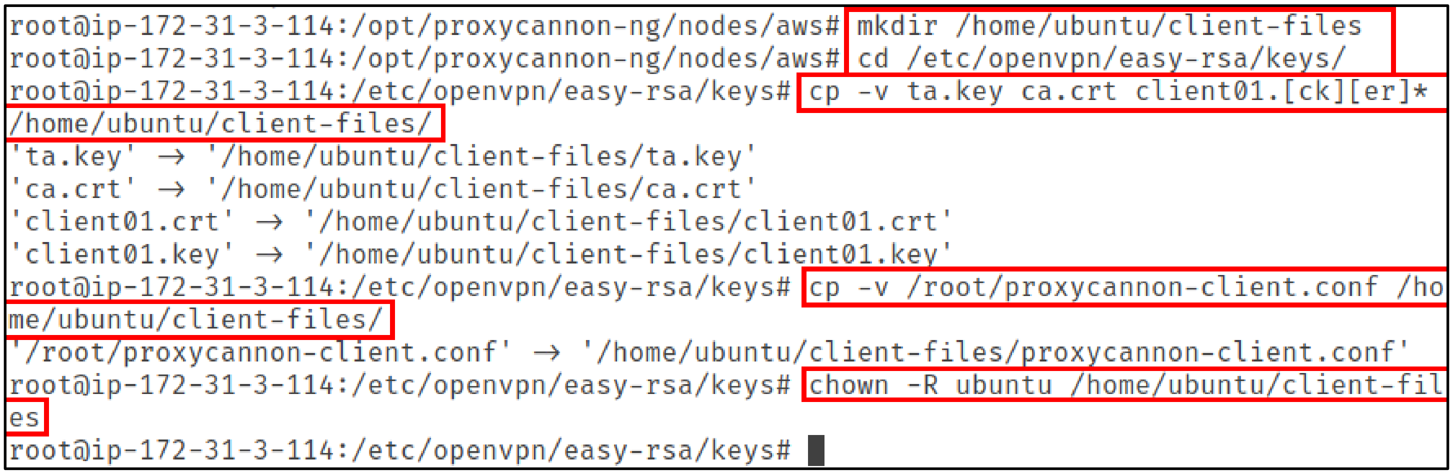
```
root@ip-172-31-3-114:/opt/proxycannon-ng/nodes/aws# terraform init

Initializing provider plugins...

The following providers do not have any version constraints in configuration,
so the latest version was installed.

To prevent automatic upgrades to new major versions that may contain breaking
changes, it is recommended to add version = "..." constraints to the
corresponding provider blocks in configuration, with the constraint strings
suggested below.

* provider.aws: version = "~> 2.70"

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
root@ip-172-31-3-114:/opt/proxycannon-ng/nodes/aws#
```

Terraform Successfully Initialized

6. Copy OpenVPN client files to your workstation and connect to the VPN

1. Run the commands below to create the "client-files" folder in the "ubuntu" user's home directory and copy all the files needed by the OpenVPN client to that folder. Note that the second command listed below is one single command that just happened to break across three lines.

```
mkdir /home/ubuntu/client-files

cd /etc/openvpn/easy-rsa/keys/

cp -v ta.key ca.crt client01.[ck][er]* /home/ubuntu/client-files/

cp -v /root/proxycannon-client.conf /home/ubuntu/client-files/

chown -R ubuntu /home/ubuntu/client-files
```

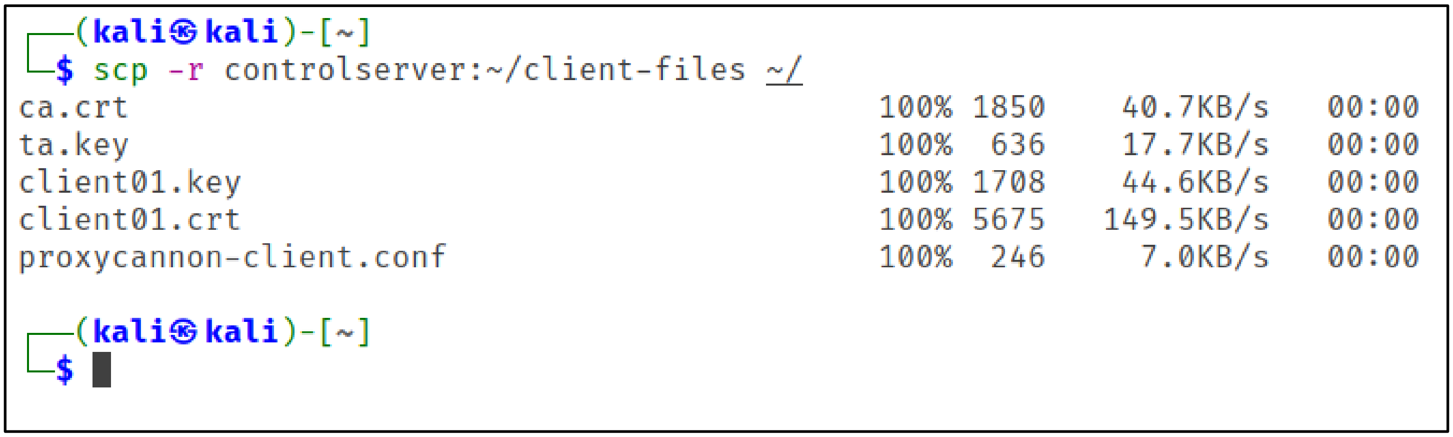


```
root@ip-172-31-3-114:/opt/proxycannon-ng/nodes/aws# mkdir /home/ubuntu/client-files
root@ip-172-31-3-114:/opt/proxycannon-ng/nodes/aws# cd /etc/openvpn/easy-rsa/keys/
root@ip-172-31-3-114:/etc/openvpn/easy-rsa/keys# cp -v ta.key ca.crt client01.[ck][er]*
/home/ubuntu/client-files/
'ta.key' -> '/home/ubuntu/client-files/ta.key'
'ca.crt' -> '/home/ubuntu/client-files/ca.crt'
'client01.crt' -> '/home/ubuntu/client-files/client01.crt'
'client01.key' -> '/home/ubuntu/client-files/client01.key'
root@ip-172-31-3-114:/etc/openvpn/easy-rsa/keys# cp -v /root/proxycannon-client.conf /ho
me/ubuntu/client-files/
'/root/proxycannon-client.conf' -> '/home/ubuntu/client-files/proxycannon-client.conf'
root@ip-172-31-3-114:/etc/openvpn/easy-rsa/keys# chown -R ubuntu /home/ubuntu/client-fil
es
root@ip-172-31-3-114:/etc/openvpn/easy-rsa/keys# 
```

VPN Client Files Copied to "/home/ubuntu/client-files"

4. In a local terminal on your Kali VM, run the "scp" command below to copy the OpenVPN files to your home directory.

```
scp -r controlserver:~/client-files ~/
```

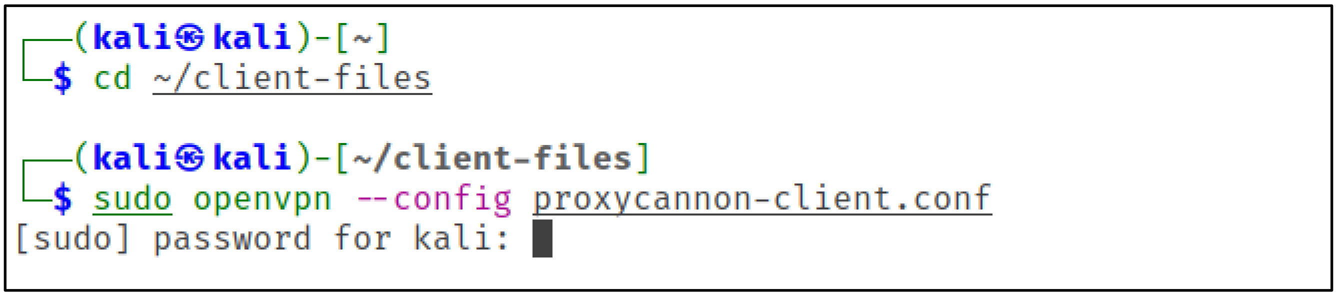


```
┌──(kali㉿kali)-[~]
└─$ scp -r controlserver:~/client-files ~/
ca.crt                                       100% 1850    40.7KB/s   00:00
ta.key                                       100%  636    17.7KB/s   00:00
client01.key                                 100% 1708    44.6KB/s   00:00
client01.crt                                 100% 5675   149.5KB/s   00:00
proxycannon-client.conf                      100%  246     7.0KB/s   00:00

┌──(kali㉿kali)-[~]
└─$ 
```

Client Files Transferred to Kali Linux VM

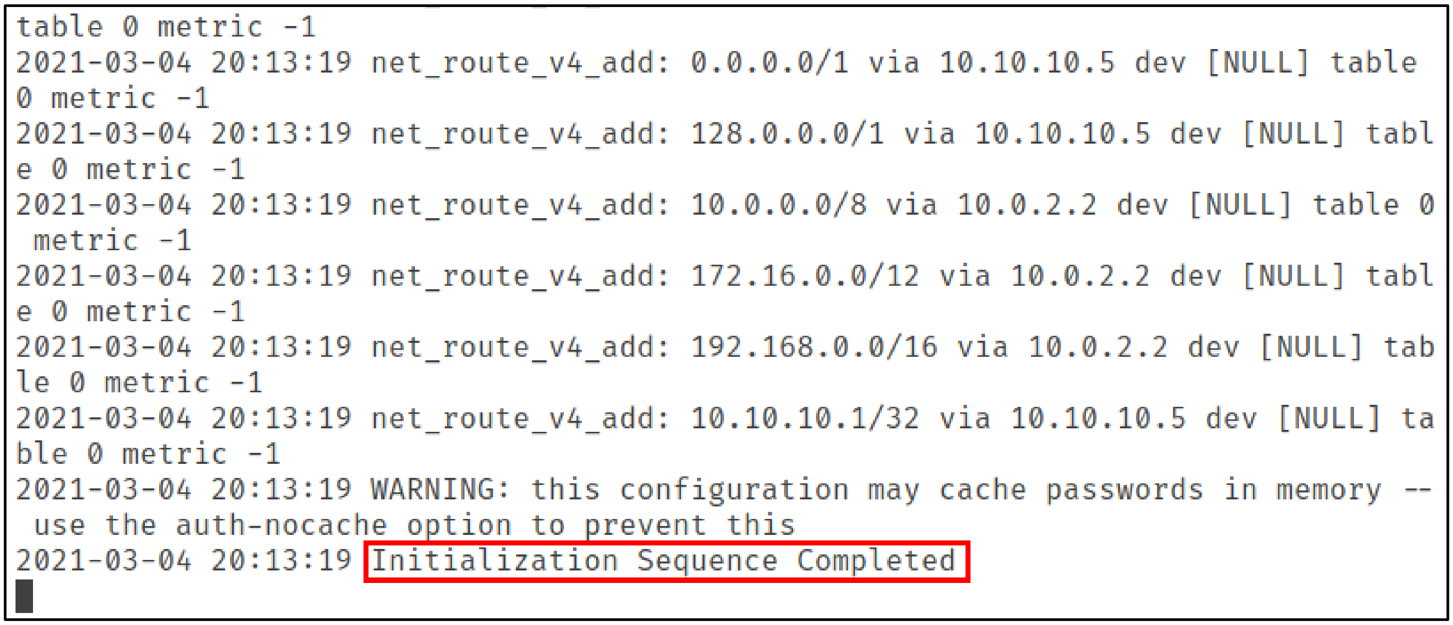
5. Confirm that the OpenVPN service is working correctly by running the commands below on your local Kali Linux VM. You should see the message "Initialization Sequence Completed" once you are connected to the VPN.

```
cd ~/client-files

sudo openvpn --config proxycannon-client.conf
```



```
┌──(kali㉿kali)-[~]
└─$ cd ~/client-files

┌──(kali㉿kali)-[~/client-files]
└─$ sudo openvpn --config proxycannon-client.conf
[sudo] password for kali: 
```

Execution of the OpenVPN Client

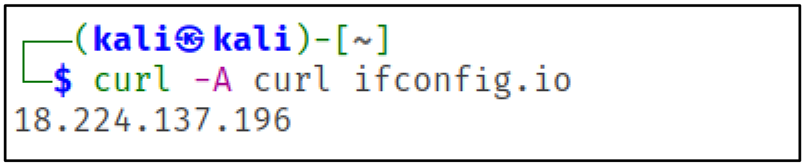


```
table 0 metric -1
2021-03-04 20:13:19 net_route_v4_add: 0.0.0.0/1 via 10.10.10.5 dev [NULL] table
0 metric -1
2021-03-04 20:13:19 net_route_v4_add: 128.0.0.0/1 via 10.10.10.5 dev [NULL] tabl
e 0 metric -1
2021-03-04 20:13:19 net_route_v4_add: 10.0.0.0/8 via 10.0.2.2 dev [NULL] table 0
 metric -1
2021-03-04 20:13:19 net_route_v4_add: 172.16.0.0/12 via 10.0.2.2 dev [NULL] tabl
e 0 metric -1
2021-03-04 20:13:19 net_route_v4_add: 192.168.0.0/16 via 10.0.2.2 dev [NULL] tab
le 0 metric -1
2021-03-04 20:13:19 net_route_v4_add: 10.10.10.1/32 via 10.10.10.5 dev [NULL] ta
ble 0 metric -1
2021-03-04 20:13:19 WARNING: this configuration may cache passwords in memory --
 use the auth-nocache option to prevent this
2021-03-04 20:13:19 Initialization Sequence Completed
```

Successful Connection to OpenVPN Service on the ProxyCannon Server

6. Open a new terminal window on your Kali VM and run the command below to observe your external IP address. You should see that your external IP address is now the same as the IP address of your ProxyCannon control server.

```
curl -A curl ifconfig.io
```



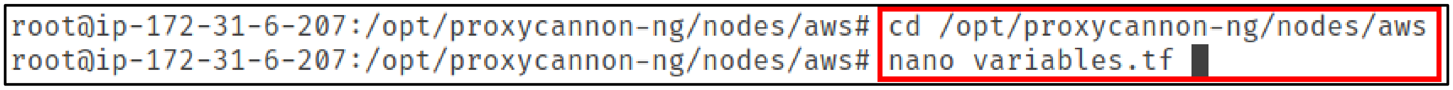
```
┌──(kali㉿kali)-[~]
└─$ curl -A curl ifconfig.io
18.224.137.196
```

External IP Address of the ProxyCannon Control Server Shown

7. Add exit nodes to your ProxyCannon "botnet"

1. In your SSH session with the control server, run the following commands to edit "variables.tf" in Nano again.

```
cd /opt/proxycannon-ng/nodes/aws

nano variables.tf
```

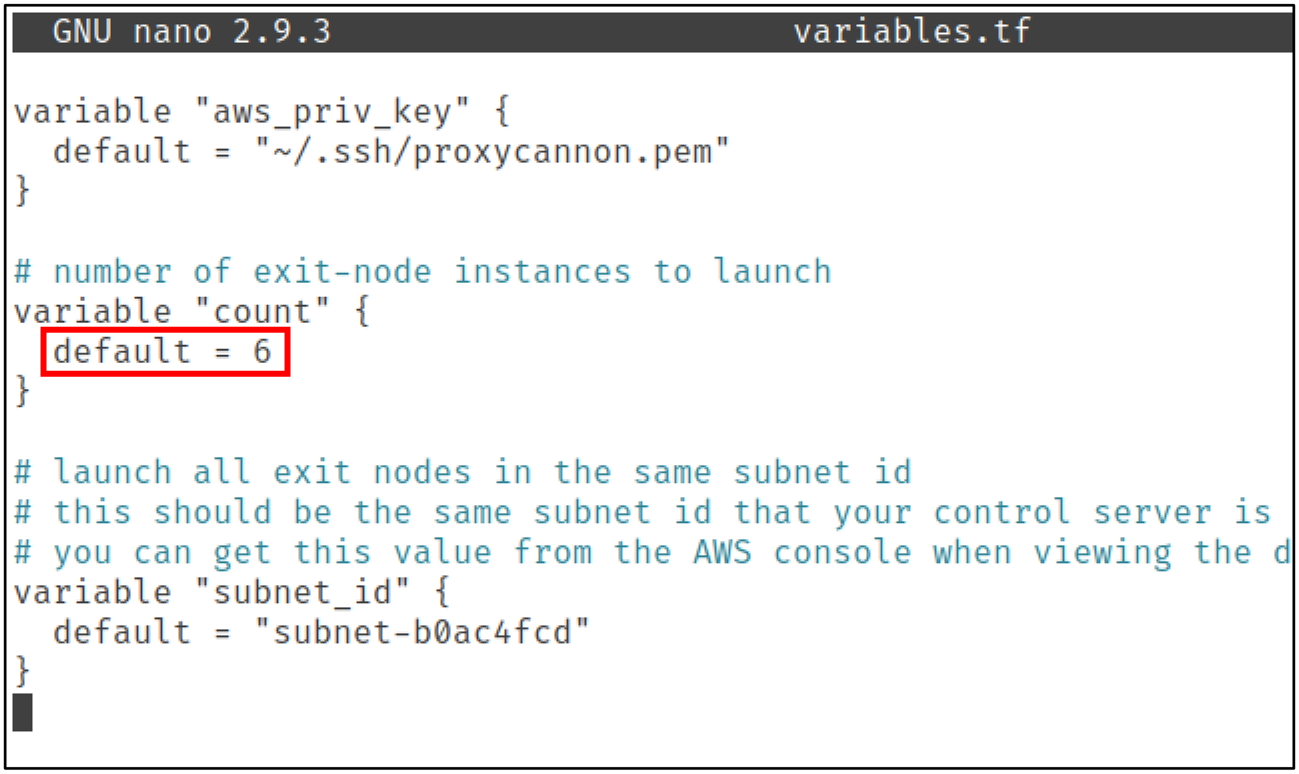


```
root@ip-172-31-6-207:/opt/proxycannon-ng/nodes/aws# cd /opt/proxycannon-ng/nodes/aws
root@ip-172-31-6-207:/opt/proxycannon-ng/nodes/aws# nano variables.tf
```

Opening "variables.tf" for Editing

2. Inside the 'variable "count"' section, change the default number of exit nodes from 2 to 6. Then press CTRL+O followed by Enter to save the file, and press CTRL+X to exit.

```
# number of exit-node instances to launch
variable "count" {
  default = 6
}
```

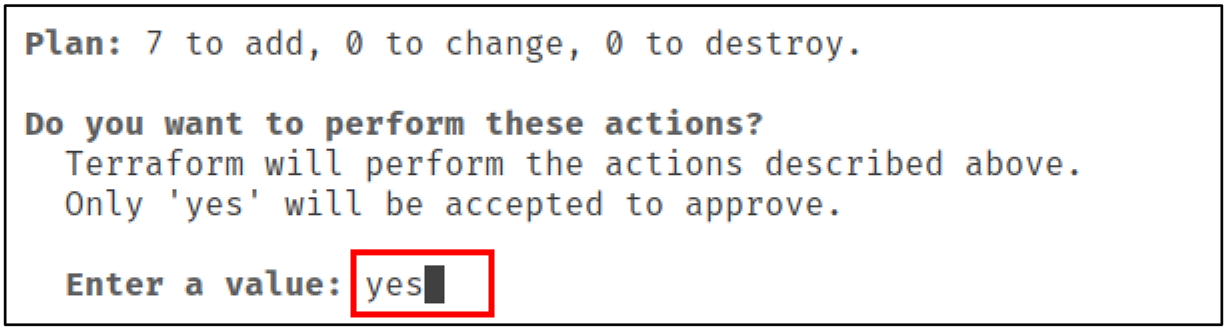


```
  GNU nano 2.9.3                     variables.tf

variable "aws_priv_key" {
  default = "~/.ssh/proxycannon.pem"
}

# number of exit-node instances to launch
variable "count" {
  default = 6
}

# launch all exit nodes in the same subnet id
# this should be the same subnet id that your control server is
# you can get this value from the AWS console when viewing the d
variable "subnet_id" {
  default = "subnet-b0ac4fcd"
}
```

Count Default Value Changed from 2 to 6

3. Run "terraform apply" to apply the changes you just made to the variables.tf file. When prompted, type "yes" and press Enter to apply the changes.

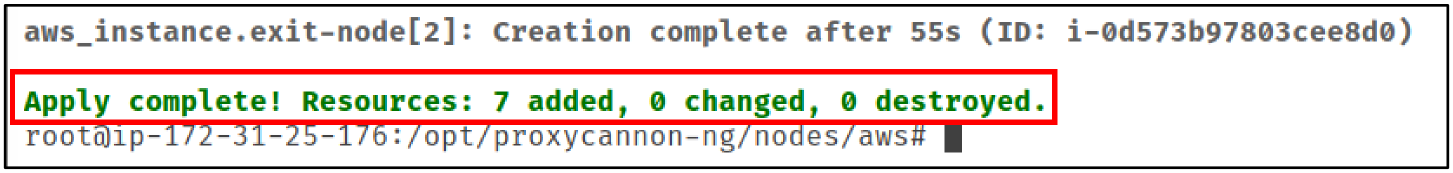
```
terraform apply
```



```
Plan: 7 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes
```

"yes" Entered at Prompt

4. After the changes have been applied, you should see a message that says "Apply complete!" You will then have 6 exit nodes in your ProxyCannon botnet that your traffic can be routed through.

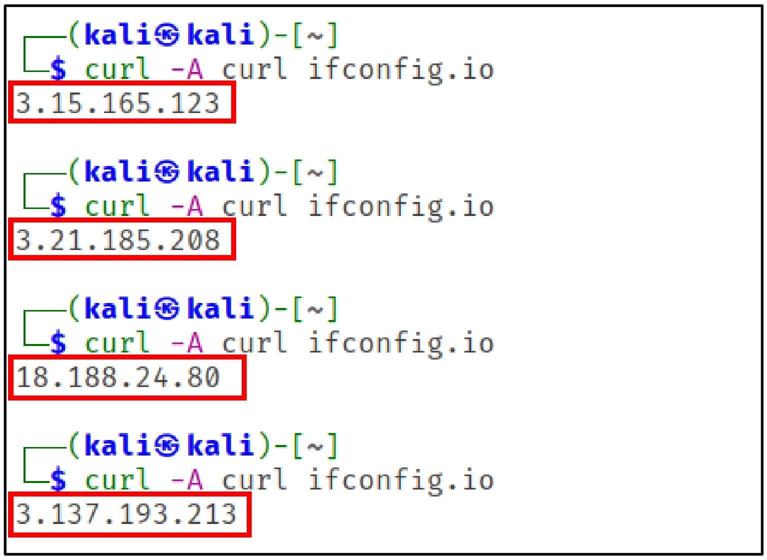


```
aws_instance.exit-node[2]: Creation complete after 55s (ID: i-0d573b97803cee8d0)

Apply complete! Resources: 7 added, 0 changed, 0 destroyed.
root@ip-172-31-25-176:/opt/proxycannon-ng/nodes/aws#
```

Six Exit Nodes Deployed Successfully

5. You can observe your traffic being routed through the different exit nodes by opening a local terminal on your Kali VM and running the curl command previously used to observe your external IP address. Note that each time you run the command, a different external IP address is displayed.

```
curl -A curl ifconfig.io
```

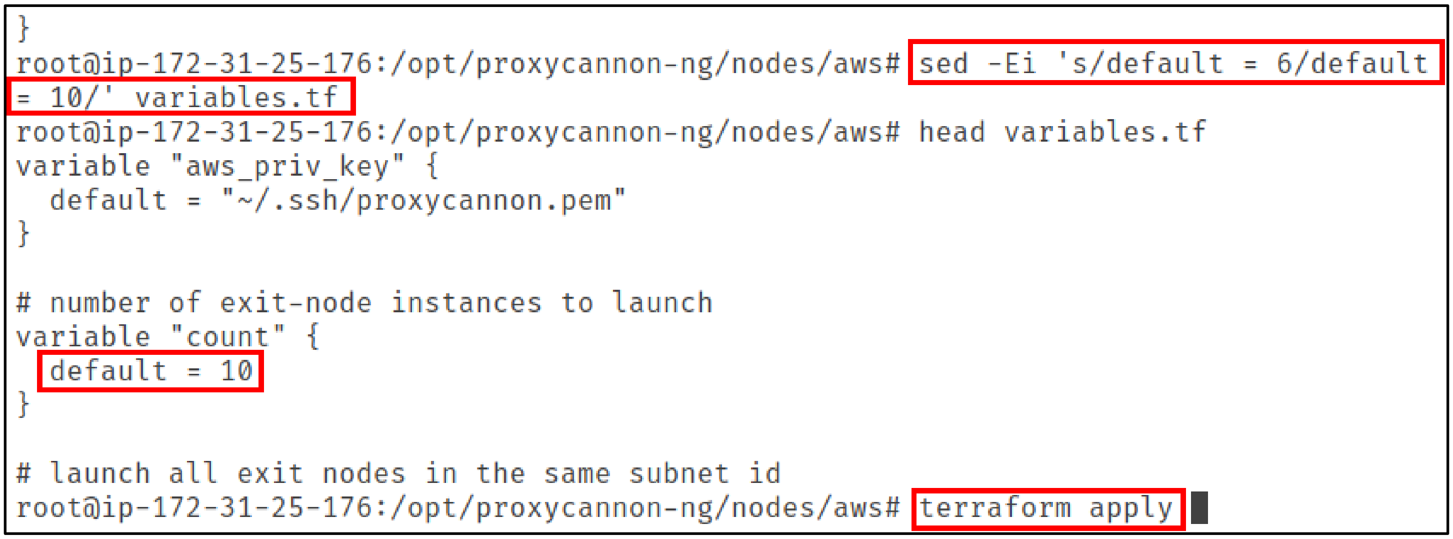


```
┌──(kali㉿kali)-[~]
└─$ curl -A curl ifconfig.io
3.15.165.123

┌──(kali㉿kali)-[~]
└─$ curl -A curl ifconfig.io
3.21.185.208

┌──(kali㉿kali)-[~]
└─$ curl -A curl ifconfig.io
18.188.24.80

┌──(kali㉿kali)-[~]
└─$ curl -A curl ifconfig.io
3.137.193.213
```

External IP Addresses of Exit Nodes Shown

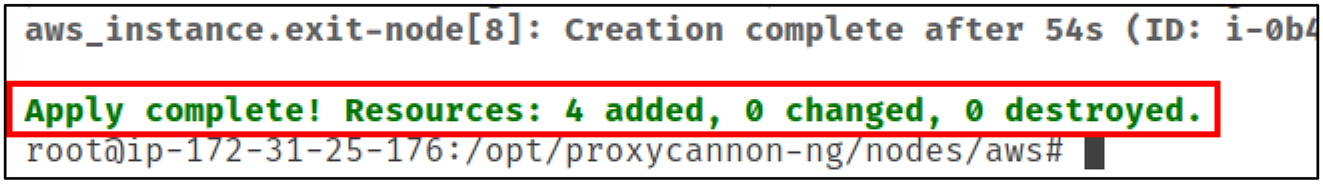
Information for future reference: Adding and removing exit nodes

1. While not required for this lab, you can change the number of exit-nodes that are running in your ProxyCannon botnet at any time by repeating the steps in section 6, above, and modifying the number of exit nodes specified in your variables.tf file. The screenshots below demonstrate changing the number of exit nodes from 6 to 10.



```
}
root@ip-172-31-25-176:/opt/proxycannon-ng/nodes/aws# sed -Ei 's/default = 6/default
= 10/' variables.tf
root@ip-172-31-25-176:/opt/proxycannon-ng/nodes/aws# head variables.tf
variable "aws_priv_key" {
  default = "~/.ssh/proxycannon.pem"
}

# number of exit-node instances to launch
variable "count" {
  default = 10
}

# launch all exit nodes in the same subnet id
root@ip-172-31-25-176:/opt/proxycannon-ng/nodes/aws# terraform apply
```

Exit Node Count Changed from 6 to 10

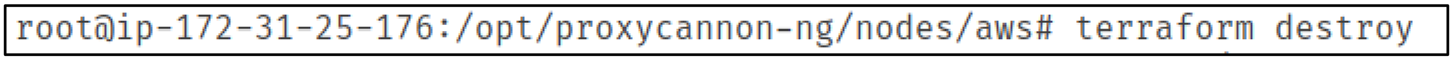


```
aws_instance.exit-node[8]: Creation complete after 54s (ID: i-0b4

Apply complete! Resources: 4 added, 0 changed, 0 destroyed.
root@ip-172-31-25-176:/opt/proxycannon-ng/nodes/aws#
```

Four New Exit Nodes Deployed

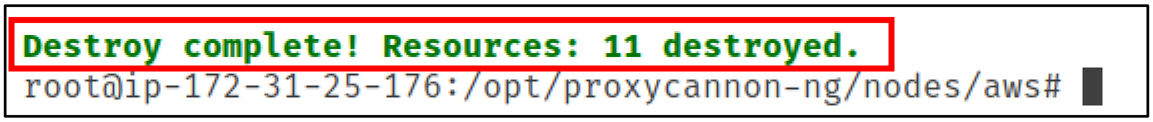
2. You can also stop all exit nodes by running "terraform destroy". (Do not run this command right now.)

```
terraform destroy
```



```
root@ip-172-31-25-176:/opt/proxycannon-ng/nodes/aws# terraform destroy
```

"terraform destroy" Command Executed



```
Destroy complete! Resources: 11 destroyed.
root@ip-172-31-25-176:/opt/proxycannon-ng/nodes/aws#
```

Removal of all 10 Exit Nodes Complete

Additional resources

- ProxyCannon project on GitHub