Assessment -9

VPC

Trainee Name : Arun Parmar
Mentor Name : Ravi Kumar
College: UPES

1. When to use Elastic IP over Public IP

   Public IP addresses are dynamic - i.e. if you stop/start your instance you get reassigned a new public IP.Elastic IPs get allocated to your account, and stay the same - it's up to you to attach them to an instance or not. You could say they are static public IP addresses.Elastic IP address is a public static IPv4 address which is reachable from the Internet.Basically Elastic IP addresses are used by AWS to manage its dynamic cloud computing services. Within the AWS infrastructure, customers have virtual private clouds (VPC), within the VPCs, users have instances. So when you launch an EC2 instance, you receive a Public IP address by which that instance is reachable from the internet. Once you stop that instance and restart the instance you get a new Public IP for the same instance. So it's basically a problem to connect your instance from the internet for not having a static IP. To overcome this problem, we attach an Elastic IP to an Instance which doesn't change after you stop / start the instance.

2. Valid IP Ranges for LAN, Implication of using Public IP ranges for Private Network.

   Class A : 10.0.0.0 – 10.0.255.255

   Class B : 172.16.0.0 – 172.13.255.255

   Class C : 192.168.0.0 – 192.168.255.255

   If you ever expect to connect these systems to an Internet-facing router, though, then you could experience the following issues if you don't stick with private IP ranges:

   ● Traffic destined for another host may leak out on to the Internet.

   ● You might want to get to the IANA-assigned host on that IP and may not be

   able to do it if it's an internal host.

   ● If you aren't the only one mantaining this network, you could horribly confuse someone who is doing troubleshooting.

3. List down the things to keep in mind while VPC peering.

   To create a VPC peering connection with another VPC, be aware of the following

   limitations and rules:

   ● You cannot create a VPC peering connection between VPCs that have matching or overlapping IPv4 or IPv6 CIDR blocks. Amazon always assigns your VPC an unique

IPv6 CIDR block. If your IPv6 CIDR blocks are unique but your IPv4 blocks are not, you cannot create the peering connection.

● You have a quota on the number of active and pending VPC peering connections that you can have per VPC. For more information, see Amazon VPC Quotas in the Amazon VPC User Guide

● VPC peering does not support transitive peering relationships. In a VPC peering connection, your VPC does not have access to any other VPCs with which the peer VPC may be peered. This includes VPC peering connections that are established entirely within your own AWS account. For more information about unsupported peering relationships, see Unsupported VPC Peering Configurations. For examples of supported peering relationships, see VPC Peering Scenarios.

● You cannot have more than one VPC peering connection between the same two VPCs at the same time.

● Unicast reverse path forwarding in VPC peering connections is not supported. For more information, see Routing for Response Traffic.

● Any tags that you create for your VPC peering connection are only applied in the account or region in which you create them.

● If the IPv4 CIDR block of a VPC in a VPC peering connection falls outside of the private IPv4 address ranges specified by RFC 1918, private DNS hostnames for that VPC cannot be resolved to private IP addresses. To resolve private DNS hostnames to private IP addresses, you can enable DNS resolution support for the VPC peering connection. For more information, see Enabling DNS Resolution Support for a VPC Peering Connection.

● You cannot connect to or query the Amazon DNS server in a peer VPC.

4. CIDR of a VPC is 10.0.0.0/16, if the subnet mask is /20 calculate the number of subnets that could be created from the VPC. Also find the number of IP in subnet.

   *10.0.0.0/16: 00001010.00000000.00000000.00000000 (In /16, first 2 octets are fixed).

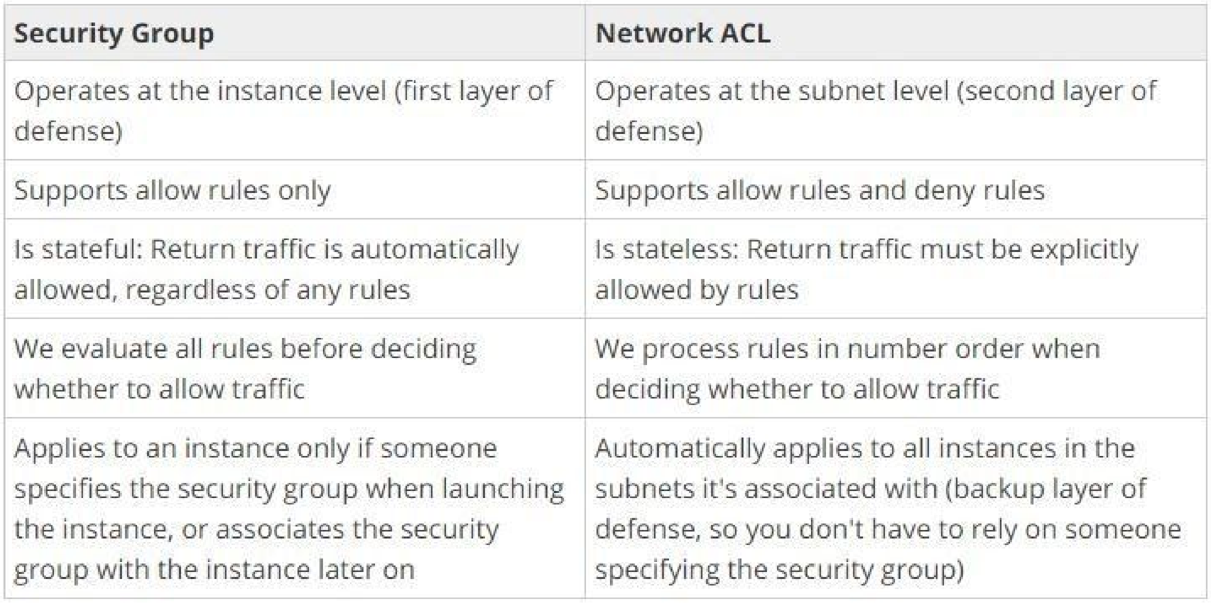
   00001010.00000000.00000000.00000000 (In /20, extra 4 bits are borrowed from hosts)

   * These extra 4 bits are subnetting bits.

   So, total number of subnets = 2^(20-16)=(16)

And, total IP’S in each subnet = 2^(32-20)= (4096)

5. Differentiate between NACL and Security Groups.



Security Group	Network ACL
Operates at the instance level (first layer of defense)	Operates at the subnet level (second layer of defense)
Supports allow rules only	Supports allow rules and deny rules
Is stateful: Return traffic is automatically allowed, regardless of any rules	Is stateless: Return traffic must be explicitly allowed by rules
We evaluate all rules before deciding whether to allow traffic	We process rules in number order when deciding whether to allow traffic
Applies to an instance only if someone specifies the security group when launching the instance, or associates the security group with the instance later on	Automatically applies to all instances in the subnets it's associated with (backup layer of defense, so you don't have to rely on someone specifying the security group)

6. Implement a 2-tier vpc with following requirements:

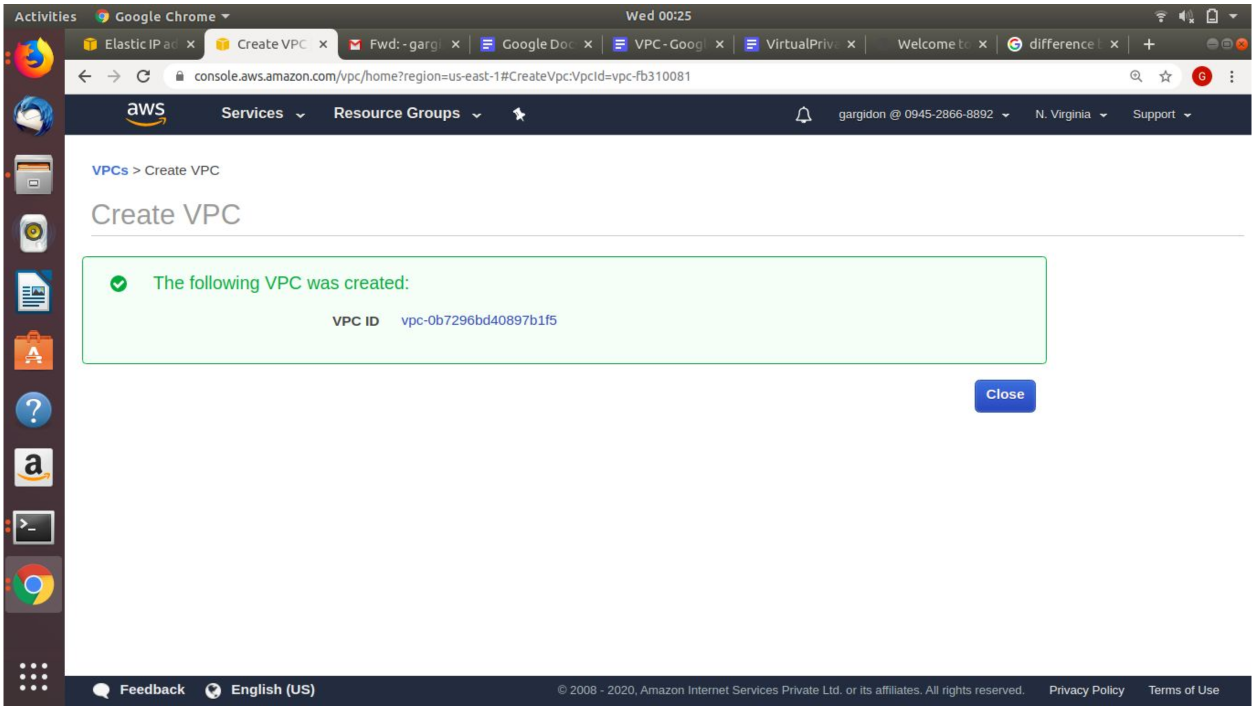
1. Create a private subnet, attach NAT, and host an application server(Tomcat)

2. Create a public subnet, and host a web server(Nginx), also proxypass to Tomcat from Nginx

After Implementing this on AWS, create an architecture diagram for this use case.

Note: For hosting Nginx in public subnet, use Elastic IP.

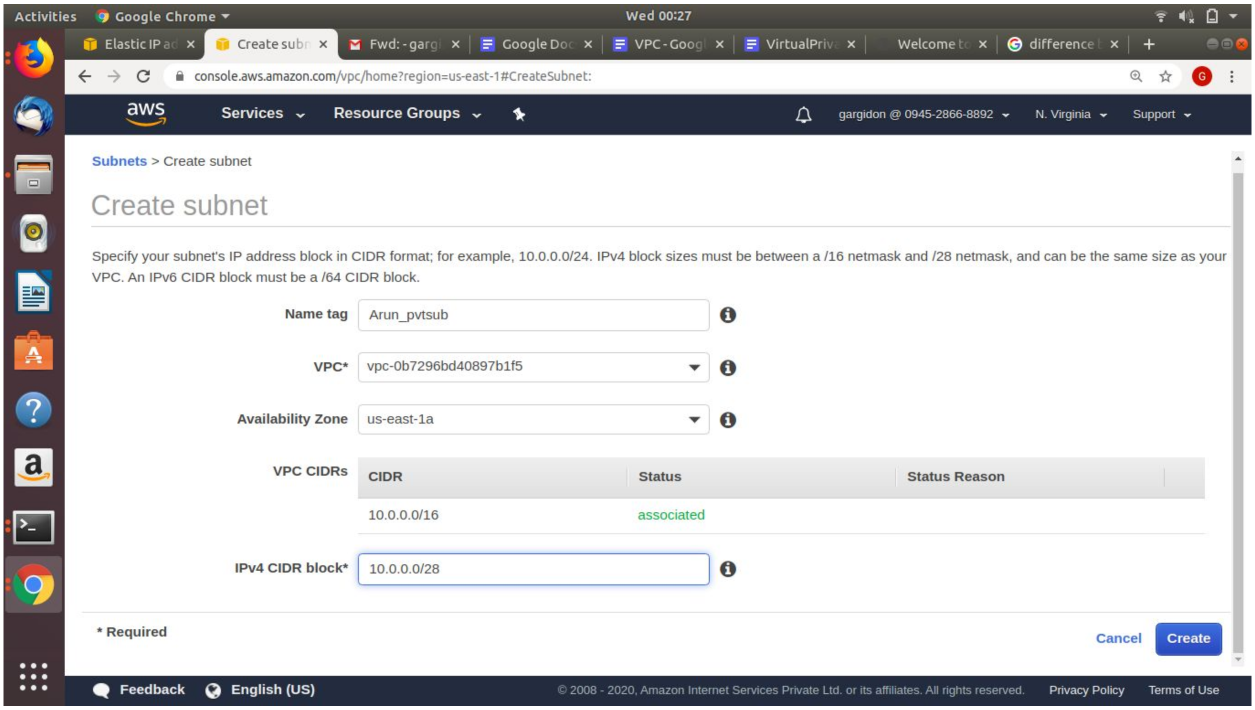
STEP1: create a VPC



The following VPC was created:

VPC ID vpc-0b7296bd40897b1f5

STEP2: Creating private subnet: CIDR(10.0.0.0/28)



Specify your subnet's IP address block in CIDR format; for example, 10.0.0.0/24. IPv4 block sizes must be between a /16 netmask and /28 netmask, and can be the same size as your VPC. An IPv6 CIDR block must be a /64 CIDR block.

Name tag: Arun_pvtsub

VPC*: vpc-0b7296bd40897b1f5

Availability Zone: us-east-1a

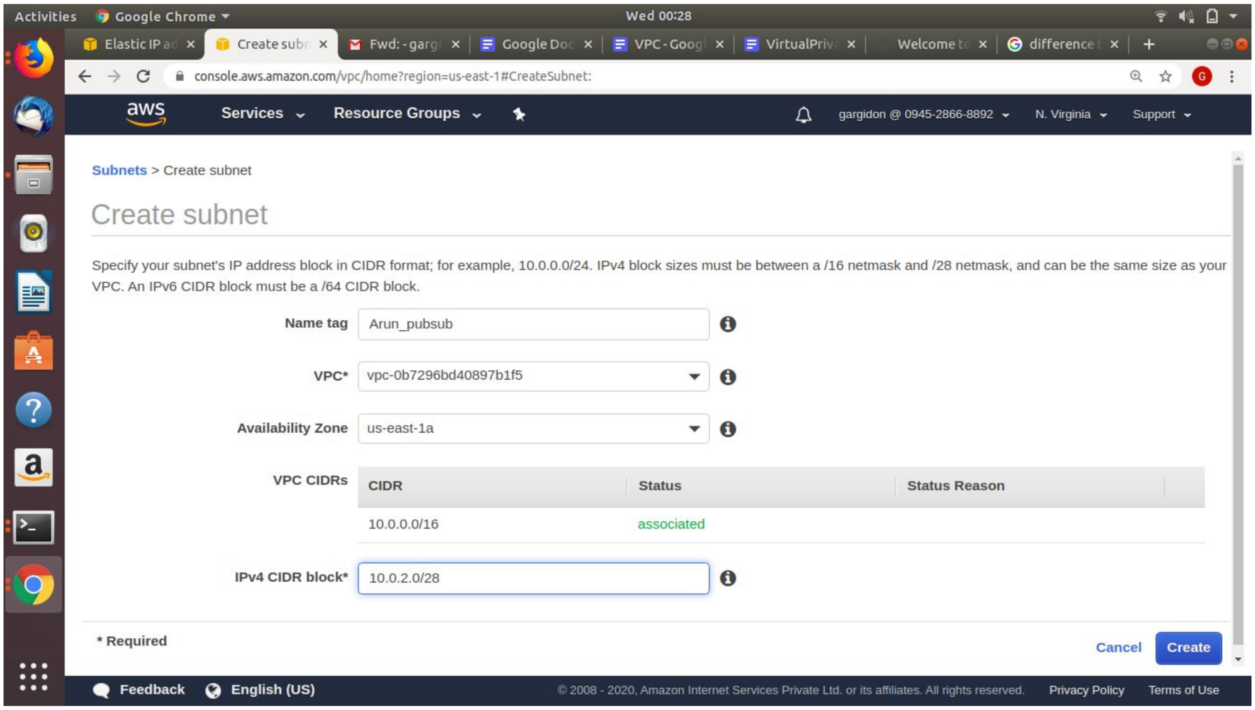
VPC CIDRs

CIDR	Status	Status Reason
10.0.0.0/16	associated	

IPv4 CIDR block*: 10.0.0.0/28

* Required

STEP3: Create public subnet: CIDR(10.0.2.0/28)



console.aws.amazon.com/vpc/home?region=us-east-1#CreateSubnet:

Subnets > Create subnet

Create subnet

Specify your subnet's IP address block in CIDR format; for example, 10.0.0.0/24. IPv4 block sizes must be between a /16 netmask and /28 netmask, and can be the same size as your VPC. An IPv6 CIDR block must be a /64 CIDR block.

Name tag: Arun_pubsub

VPC*: vpc-0b7296bd40897b1f5

Availability Zone: us-east-1a

VPC CIDRs

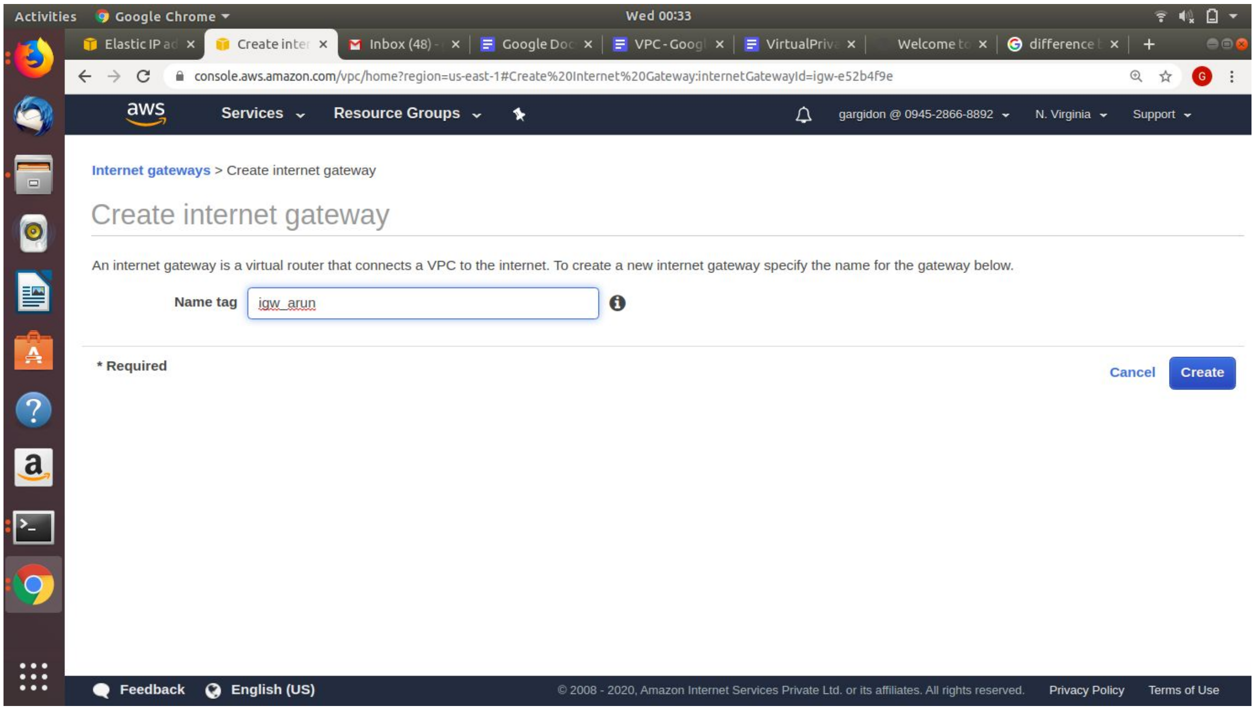
CIDR	Status	Status Reason
10.0.0.0/16	associated	

IPv4 CIDR block*: 10.0.2.0/28

* Required

Cancel Create

STEP4: Create Internet Gateway



console.aws.amazon.com/vpc/home?region=us-east-1#Create%20Internet%20Gateway:internetGatewayId=igw-e52b4f9e

Internet gateways > Create internet gateway

Create internet gateway

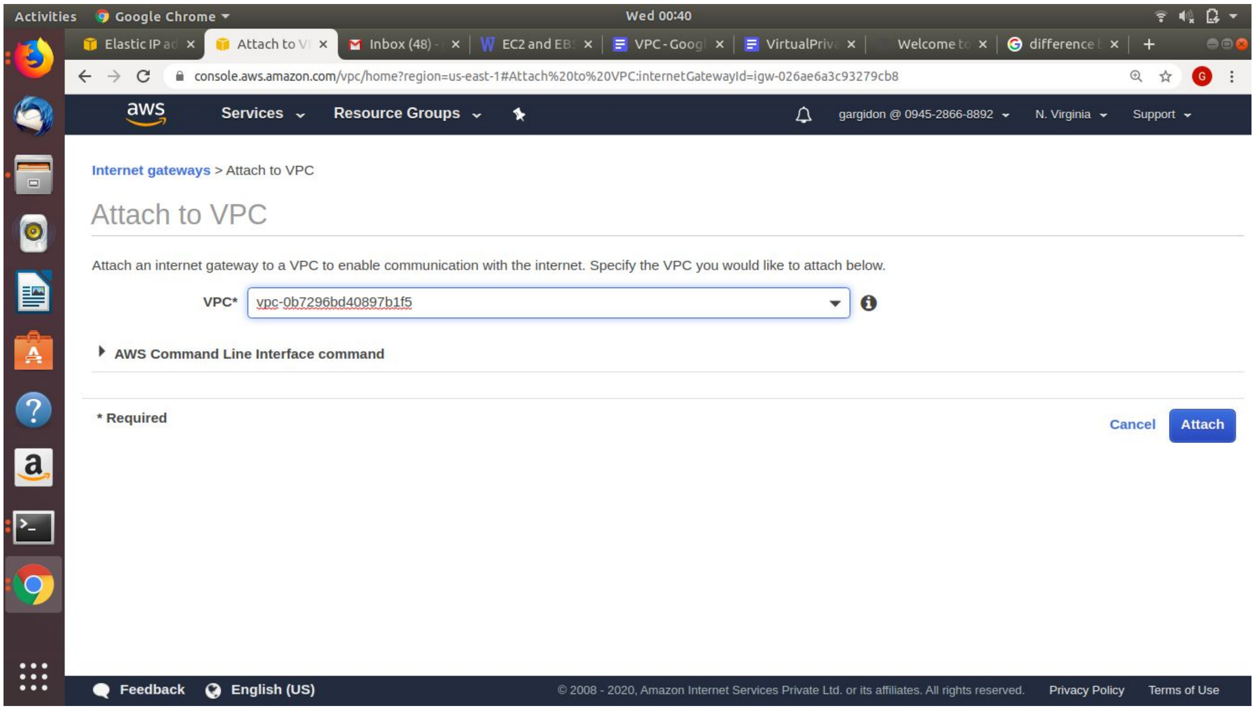
An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.

Name tag: igw_arun

* Required

Cancel Create

STEP5: Attach internet gateway to VPC


Activities
Google Chrome
Wed 00:40
Elastic IP ad
Attach to V
Inbox (48)
EC2 and EB
VPC - Googl
VirtualPriva
Welcome to
difference
console.aws.amazon.com/vpc/home?region=us-east-1#Attach%20to%20VPC:internetGatewayId=igw-026ae6a3c93279cb8
aws
Services
Resource Groups
gargidon @ 0945-2866-8892
N. Virginia
Support
Cancel
Attach
Feedback
English (US)
© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

Internet gateways > Attach to VPC

Attach to VPC

Attach an internet gateway to a VPC to enable communication with the internet. Specify the VPC you would like to attach below.

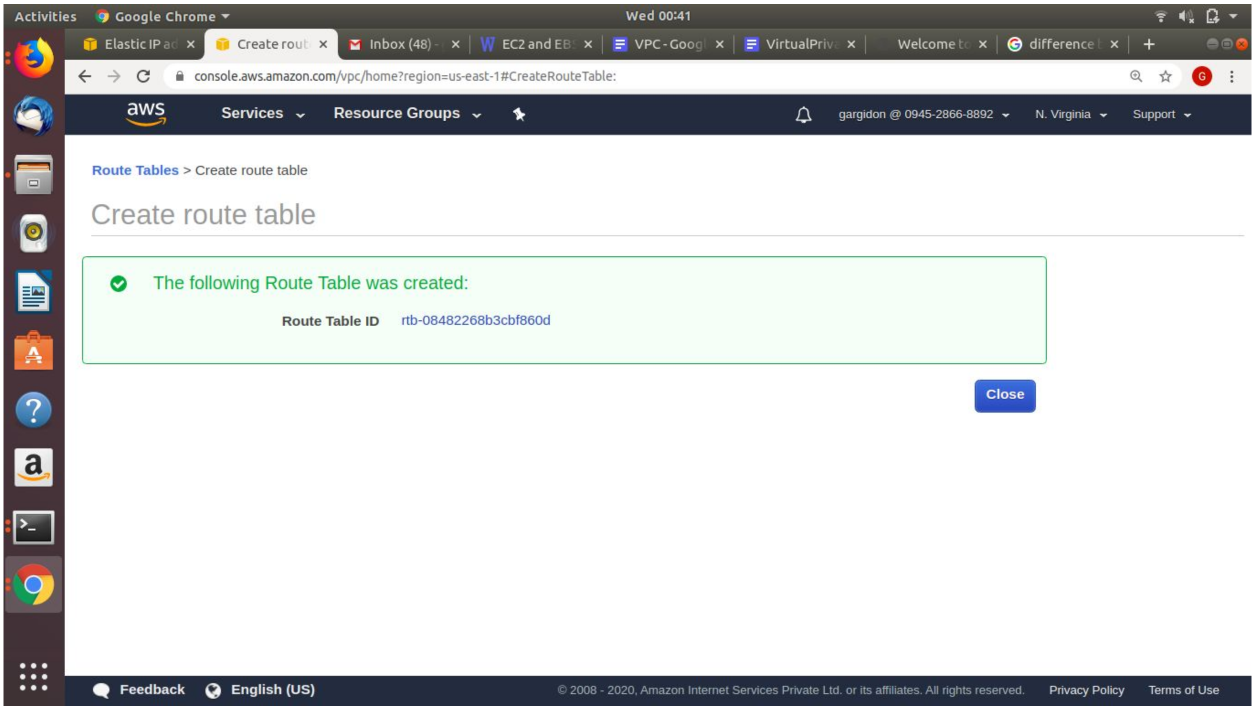
VPC* vpc-0b7296bd40897b1f5

▸ AWS Command Line Interface command

* Required

STEP6: Create route tables for public subnet

Route table for public subnet:



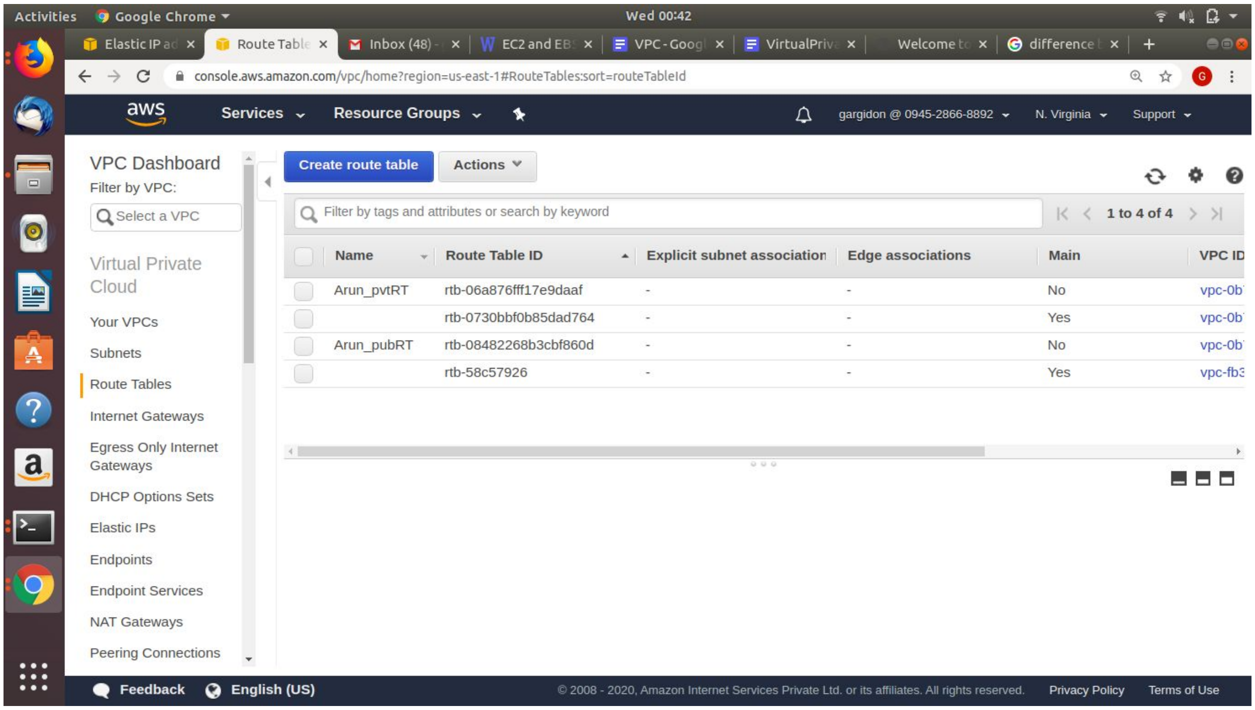
Route Tables > Create route table

Create route table

The following Route Table was created:

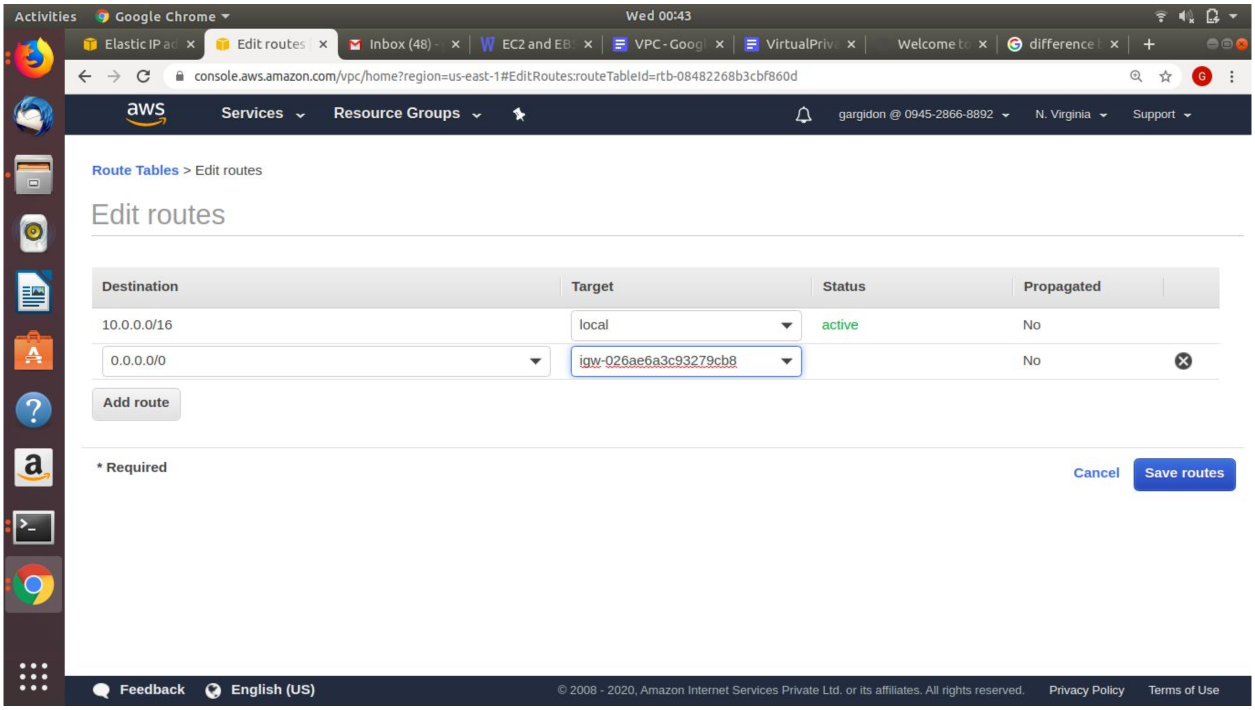
Route Table ID rtb-08482268b3cbf860d

Route table for private subnet:



Name	Route Table ID	Explicit subnet association	Edge associations	Main	VPC ID
Arun_pvtRT	rtb-06a876fff17e9daaf	-	-	No	vpc-0b
	rtb-0730bbf0b85dad764	-	-	Yes	vpc-0b
Arun_pubRT	rtb-08482268b3cbf860d	-	-	No	vpc-0b
	rtb-58c57926	-	-	Yes	vpc-fb3

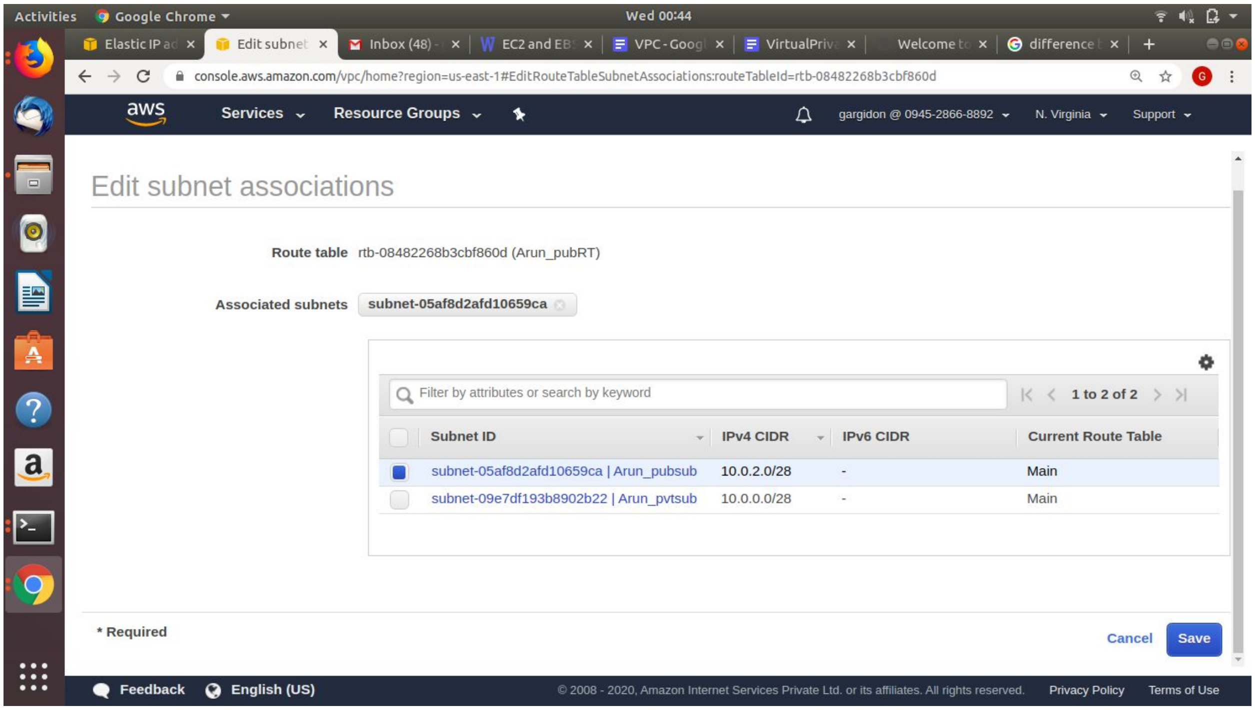
STEP7:Edit the routes(for making publicly accessible,point it towards internet gateway)


Activities
Google Chrome
Wed 00:43
console.aws.amazon.com/vpc/home?region=us-east-1#EditRoutes:routeTableId=rtb-08482268b3cbf860d
Services
Resource Groups
gargidon @ 0945-2866-8892
N. Virginia
Support
Route Tables > Edit routes
Add route
* Required
Cancel
Save routes
Feedback
English (US)
© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

Edit routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	active	No
0.0.0.0/0	igw-026ae6a3c93279cb8		No

STEP8:Associate public subnet with the public route table.


Activities
Google Chrome
Wed 00:44
Elastic IP ad
Edit subnet
Inbox (48)
EC2 and EB
VPC - Goog
VirtualPriv
Welcome to
difference
console.aws.amazon.com/vpc/home?region=us-east-1#EditRouteTableSubnetAssociations:routeTableId=rtb-08482268b3cbf860d
aws
Services
Resource Groups
gargidon @ 0945-2866-8892
N. Virginia
Support

Edit subnet associations

Route table rtb-08482268b3cbf860d (Arun_pubRT)

Associated subnets subnet-05af8d2afd10659ca

Filter by attributes or search by keyword

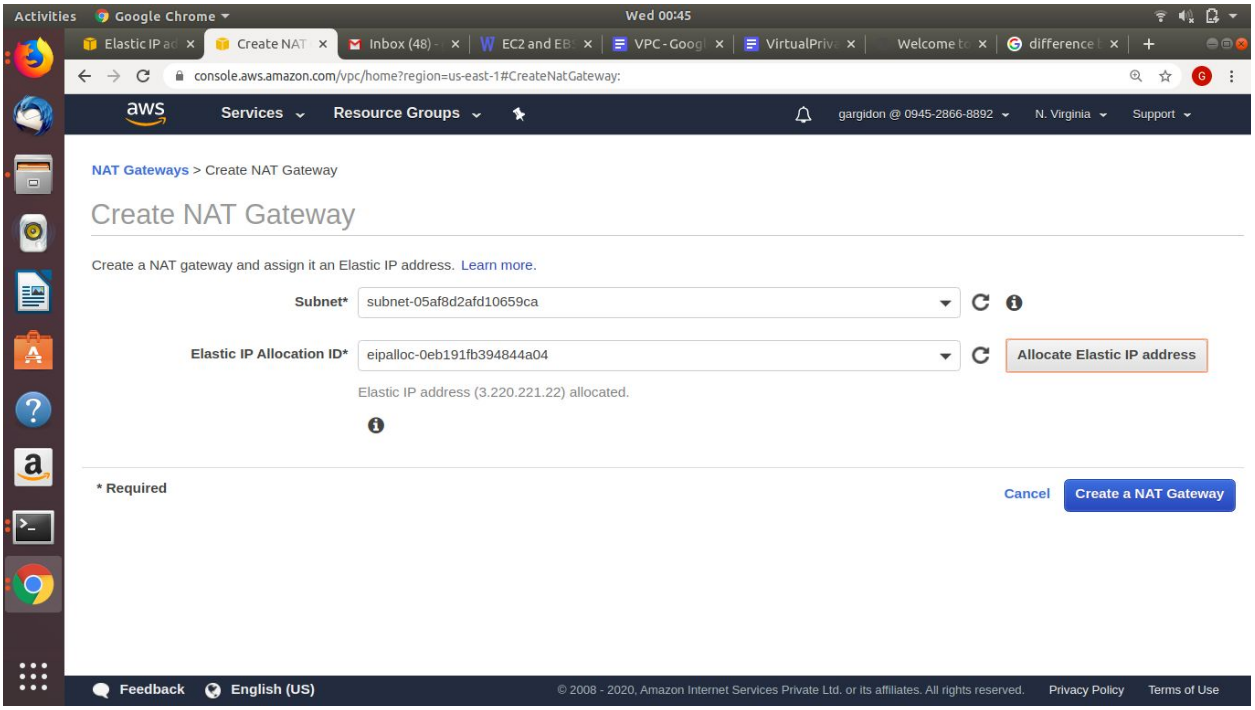
1 to 2 of 2

	Subnet ID	IPv4 CIDR	IPv6 CIDR	Current Route Table
☑	subnet-05af8d2afd10659ca | Arun_pubsub	10.0.2.0/28	-	Main
□	subnet-09e7df193b8902b22 | Arun_pvtsub	10.0.0.0/28	-	Main

* Required

Cancel
Save
Feedback
English (US)
© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

STEP9: create a NAT Gateway in public subnet



NAT Gateways > Create NAT Gateway

Create NAT Gateway

Create a NAT gateway and assign it an Elastic IP address. Learn more.

Subnet* subnet-05af8d2afd10659ca

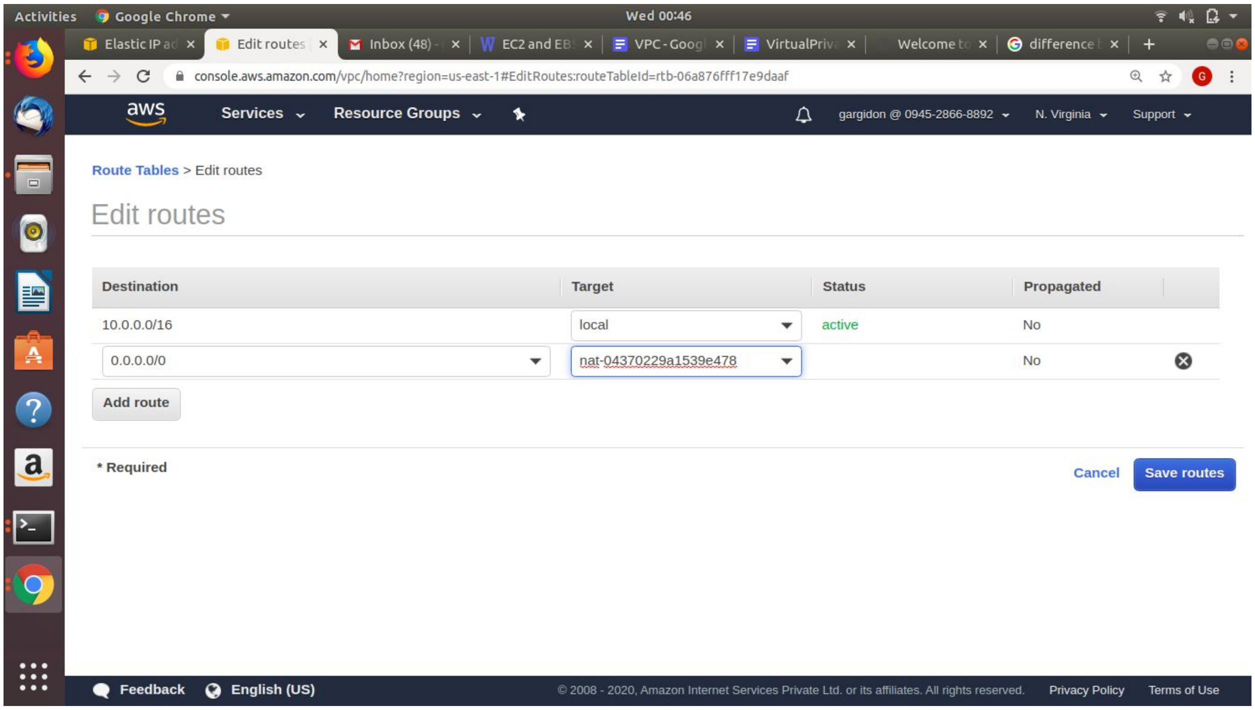
Elastic IP Allocation ID* eipalloc-0eb191fb394844a04

Elastic IP address (3.220.221.22) allocated.

* Required

Allocate Elastic IP address
Cancel
Create a NAT Gateway
Feedback
English (US)
© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

STEP10:Edit route table of pvt subnet and attach NAT gateway and associate private subnet to the route table



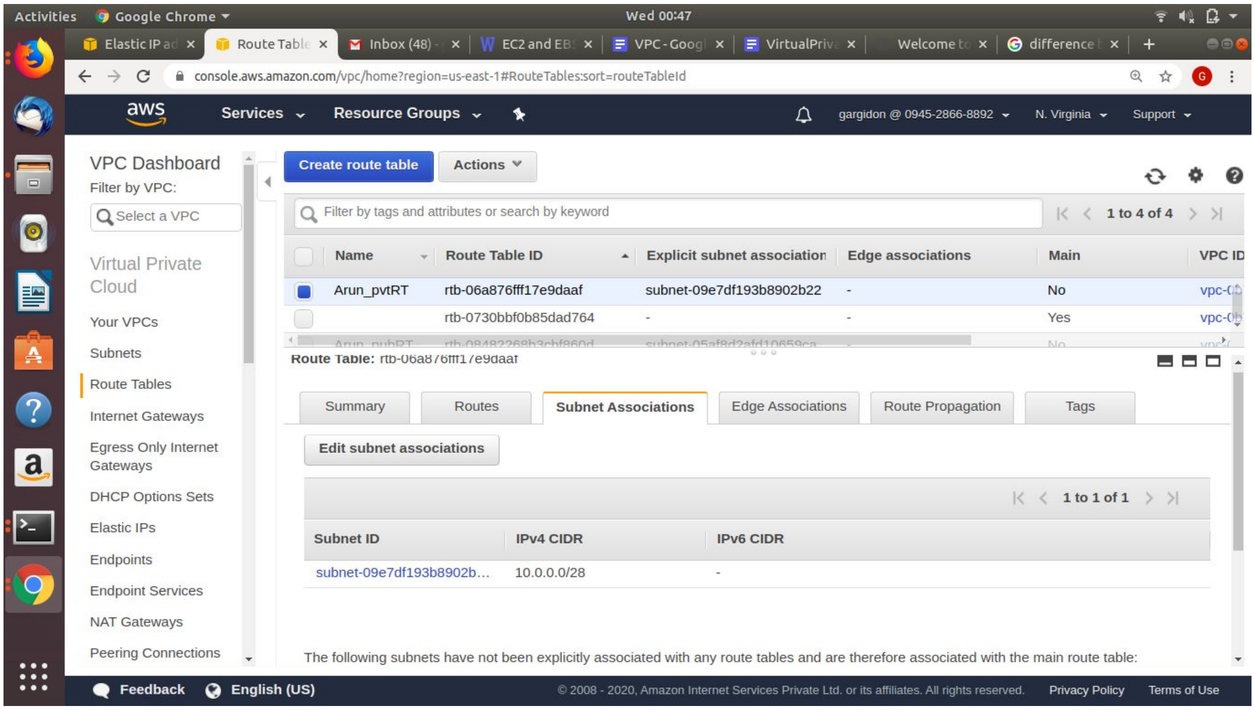
console.aws.amazon.com/vpc/home?region=us-east-1#EditRoutes:routeTableId=rtb-06a876fff17e9daaf

Route Tables > Edit routes

Edit routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	active	No
0.0.0.0/0	nat-04370229a1539e478		No

* Required



console.aws.amazon.com/vpc/home?region=us-east-1#RouteTables:sort=routeTableId

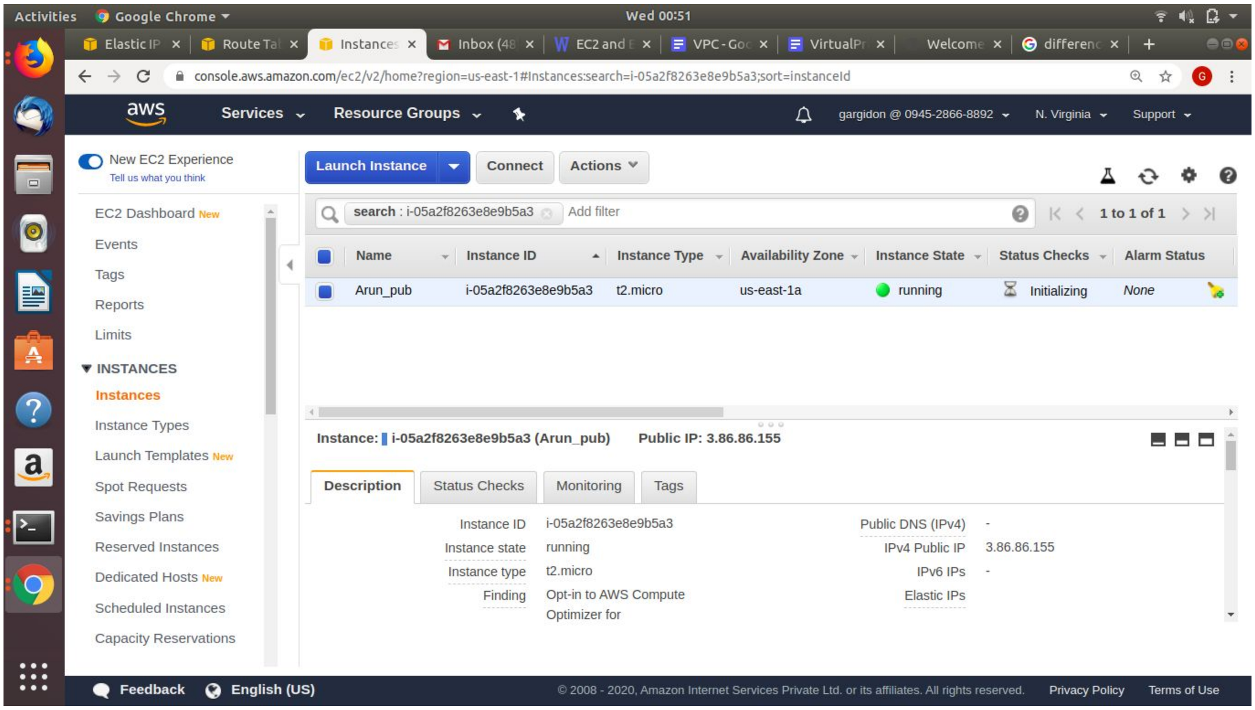
Name	Route Table ID	Explicit subnet association	Edge associations	Main	VPC ID
Arun_pvtRT	rtb-06a876fff17e9daaf	subnet-09e7df193b8902b22	-	No	vpc-0
	rtb-0730bbf0b85dad764	-	-	Yes	vpc-0
Arun_pubRT	rtb-08482268b3cbf860d	subnet-05af8d2afd10659ca	-	No	vpc-0

Route Table: rtb-06a876fff17e9daaf

Subnet ID	IPv4 CIDR	IPv6 CIDR
subnet-09e7df193b8902b...	10.0.0.0/28	-

The following subnets have not been explicitly associated with any route tables and are therefore associated with the main route table:

STEP11: Launch an instance in public subnet and specify user data for installation of nginx and enable auto assign IP

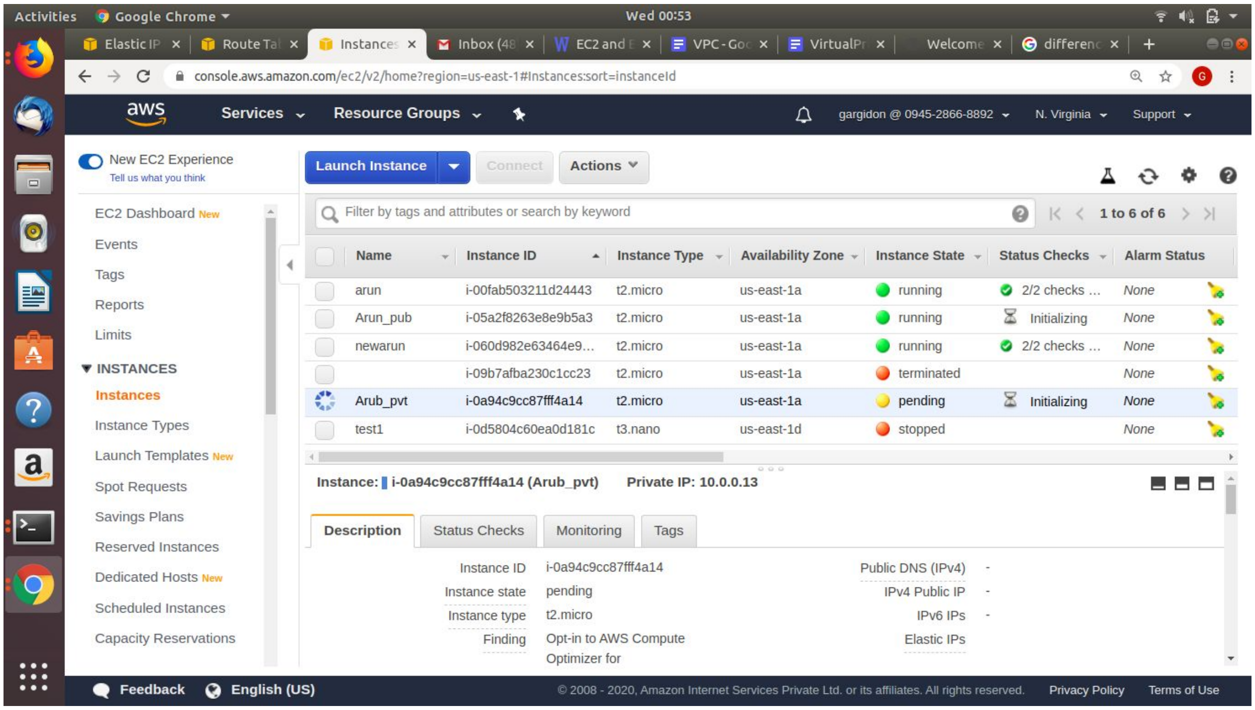


Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status
Arun_pub	i-05a2f8263e8e9b5a3	t2.micro	us-east-1a	● running	Initializing	None

Instance: i-05a2f8263e8e9b5a3 (Arun_pub) Public IP: 3.86.86.155

Instance ID	i-05a2f8263e8e9b5a3	Public DNS (IPv4)	-
Instance state	running	IPv4 Public IP	3.86.86.155
Instance type	t2.micro	IPv6 IPs	-
Finding	Opt-in to AWS Compute Optimizer for	Elastic IPs	

STEP12: Launch an instance in private subnet and disable public IP and install tomcat using user data



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status
arun	i-00fab503211d24443	t2.micro	us-east-1a	● running	✓ 2/2 checks ...	None
Arun_pub	i-05a2f8263e8e9b5a3	t2.micro	us-east-1a	● running	Initializing	None
newarun	i-060d982e63464e9...	t2.micro	us-east-1a	● running	✓ 2/2 checks ...	None
	i-09b7afba230c1cc23	t2.micro	us-east-1a	● terminated		None
Arub_pvt	i-0a94c9cc87fff4a14	t2.micro	us-east-1a	● pending	Initializing	None
test1	i-0d5804c60ea0d181c	t3.nano	us-east-1d	● stopped		None

Instance: i-0a94c9cc87fff4a14 (Arub_pvt) Private IP: 10.0.0.13

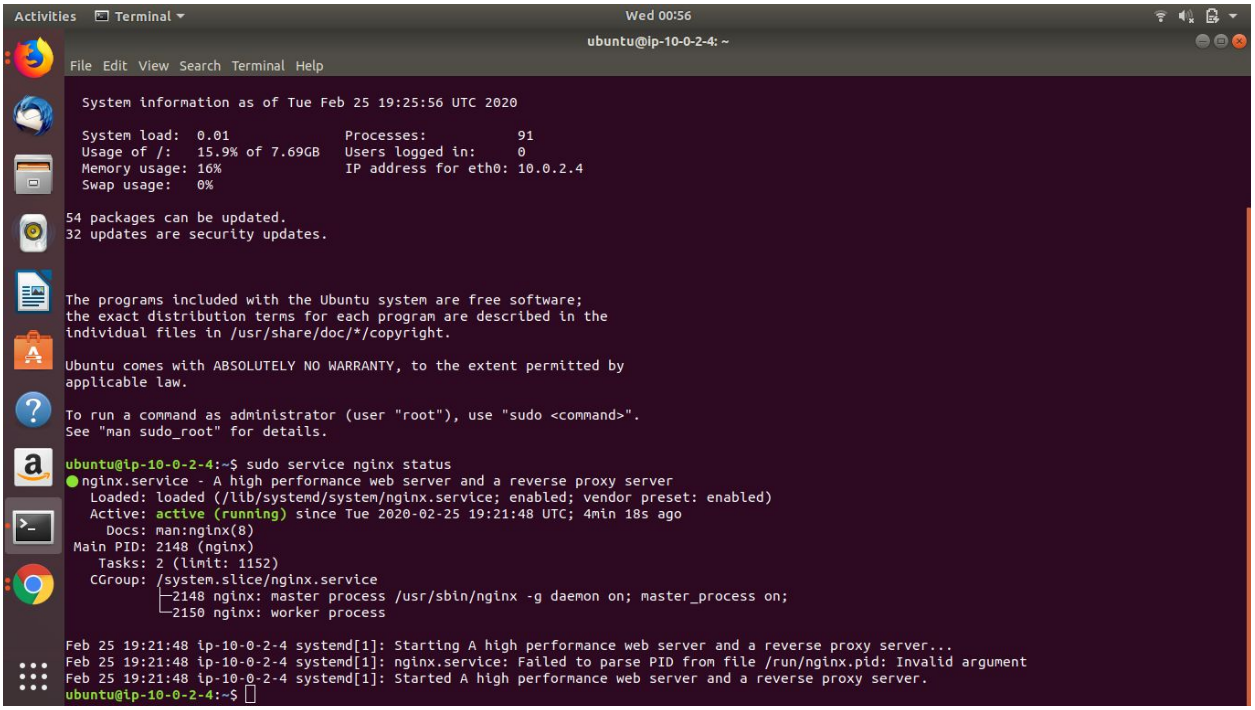
Description | Status Checks | Monitoring | Tags

Instance ID i-0a94c9cc87fff4a14
Instance state pending
Instance type t2.micro
Finding Opt-in to AWS Compute Optimizer for

Public DNS (IPv4) -
IPv4 Public IP -
IPv6 IPs -
Elastic IPs

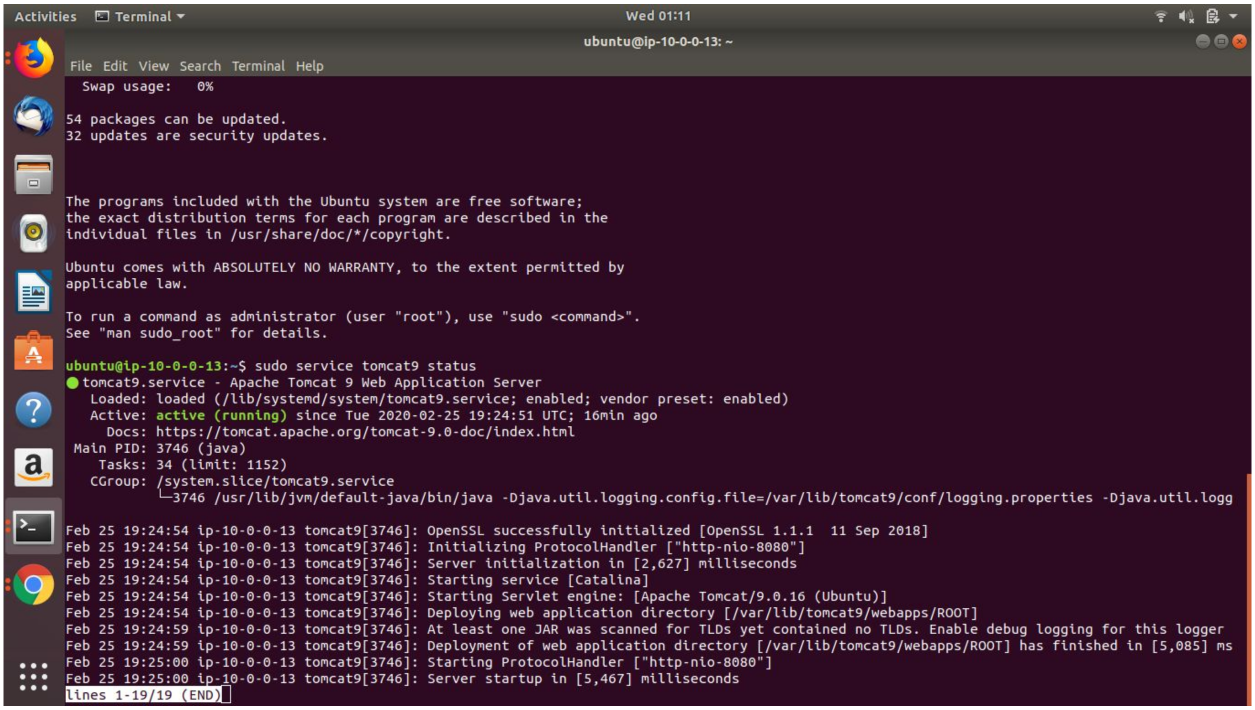
STEP13:Check if the private instance has tomcat9 installed and public instance has nginx installed.

For public instance(nginx),ssh into it and check status



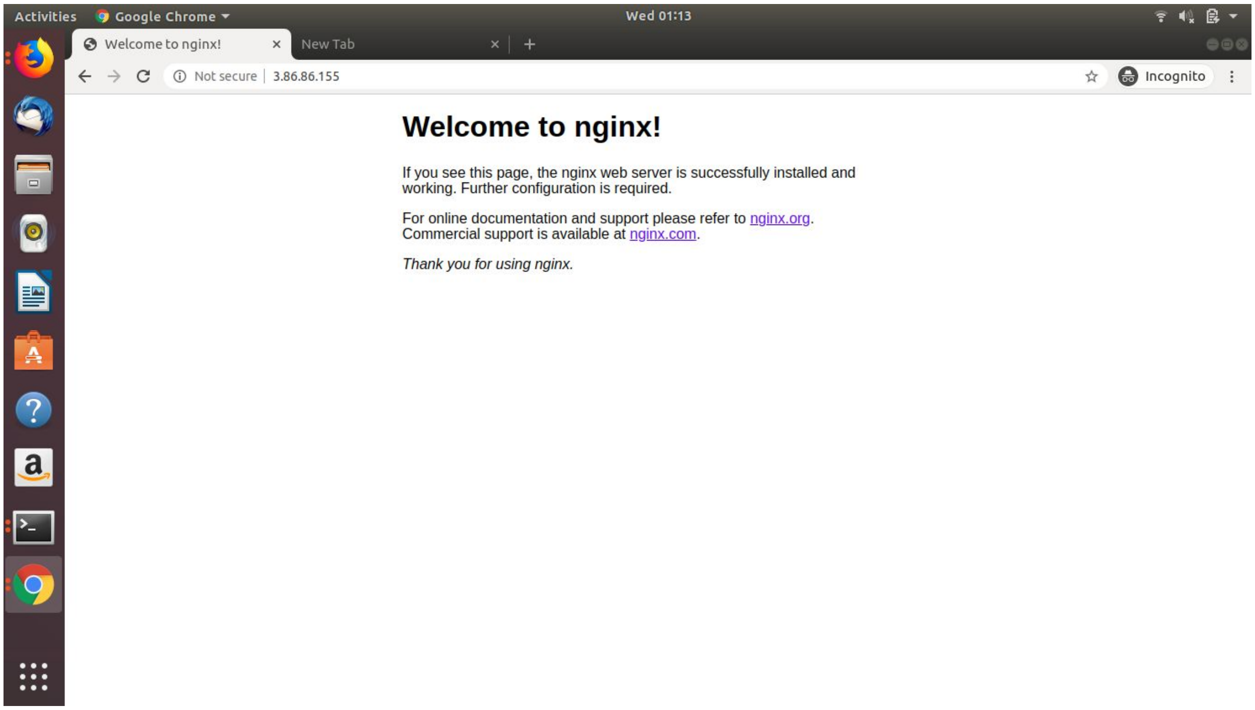
```
  System information as of Tue Feb 25 19:25:56 UTC 2020

  System load:  0.01              Processes:           91
  Usage of /:   15.9% of 7.69GB   Users logged in:     0
  Memory usage: 16%               IP address for eth0: 10.0.2.4
  Swap usage:   0%

54 packages can be updated.
32 updates are security updates.



The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

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

ubuntu@ip-10-0-2-4:~$ sudo service nginx status
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2020-02-25 19:21:48 UTC; 4min 18s ago
     Docs: man:nginx(8)
 Main PID: 2148 (nginx)
    Tasks: 2 (limit: 1152)
   CGroup: /system.slice/nginx.service
           ├─2148 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
           └─2150 nginx: worker process

Feb 25 19:21:48 ip-10-0-2-4 systemd[1]: Starting A high performance web server and a reverse proxy server...
Feb 25 19:21:48 ip-10-0-2-4 systemd[1]: nginx.service: Failed to parse PID from file /run/nginx.pid: Invalid argument
Feb 25 19:21:48 ip-10-0-2-4 systemd[1]: Started A high performance web server and a reverse proxy server.
ubuntu@ip-10-0-2-4:~$ 
```

For private instance: scp the pem file to the public instance and then ssh from public instance to the private instance. Then check for tomcat9 status



```
  Swap usage:   0%

54 packages can be updated.
32 updates are security updates.



The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

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

ubuntu@ip-10-0-0-13:~$ sudo service tomcat9 status
● tomcat9.service - Apache Tomcat 9 Web Application Server
   Loaded: loaded (/lib/systemd/system/tomcat9.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2020-02-25 19:24:51 UTC; 16min ago
     Docs: https://tomcat.apache.org/tomcat-9.0-doc/index.html
 Main PID: 3746 (java)
    Tasks: 34 (limit: 1152)
   CGroup: /system.slice/tomcat9.service
           └─3746 /usr/lib/jvm/default-java/bin/java -Djava.util.logging.config.file=/var/lib/tomcat9/conf/logging.properties -Djava.util.logg

Feb 25 19:24:54 ip-10-0-0-13 tomcat9[3746]: OpenSSL successfully initialized [OpenSSL 1.1.1  11 Sep 2018]
Feb 25 19:24:54 ip-10-0-0-13 tomcat9[3746]: Initializing ProtocolHandler ["http-nio-8080"]
Feb 25 19:24:54 ip-10-0-0-13 tomcat9[3746]: Server initialization in [2,627] milliseconds
Feb 25 19:24:54 ip-10-0-0-13 tomcat9[3746]: Starting service [Catalina]
Feb 25 19:24:54 ip-10-0-0-13 tomcat9[3746]: Starting Servlet engine: [Apache Tomcat/9.0.16 (Ubuntu)]
Feb 25 19:24:54 ip-10-0-0-13 tomcat9[3746]: Deploying web application directory [/var/lib/tomcat9/webapps/ROOT]
Feb 25 19:24:59 ip-10-0-0-13 tomcat9[3746]: At least one JAR was scanned for TLDs yet contained no TLDs. Enable debug logging for this logger
Feb 25 19:24:59 ip-10-0-0-13 tomcat9[3746]: Deployment of web application directory [/var/lib/tomcat9/webapps/ROOT] has finished in [5,085] ms
Feb 25 19:25:00 ip-10-0-0-13 tomcat9[3746]: Starting ProtocolHandler ["http-nio-8080"]
Feb 25 19:25:00 ip-10-0-0-13 tomcat9[3746]: Server startup in [5,467] milliseconds
lines 1-19/19 (END)
```

STEP14:Paste the public IP of your instance and check for the default nginx page



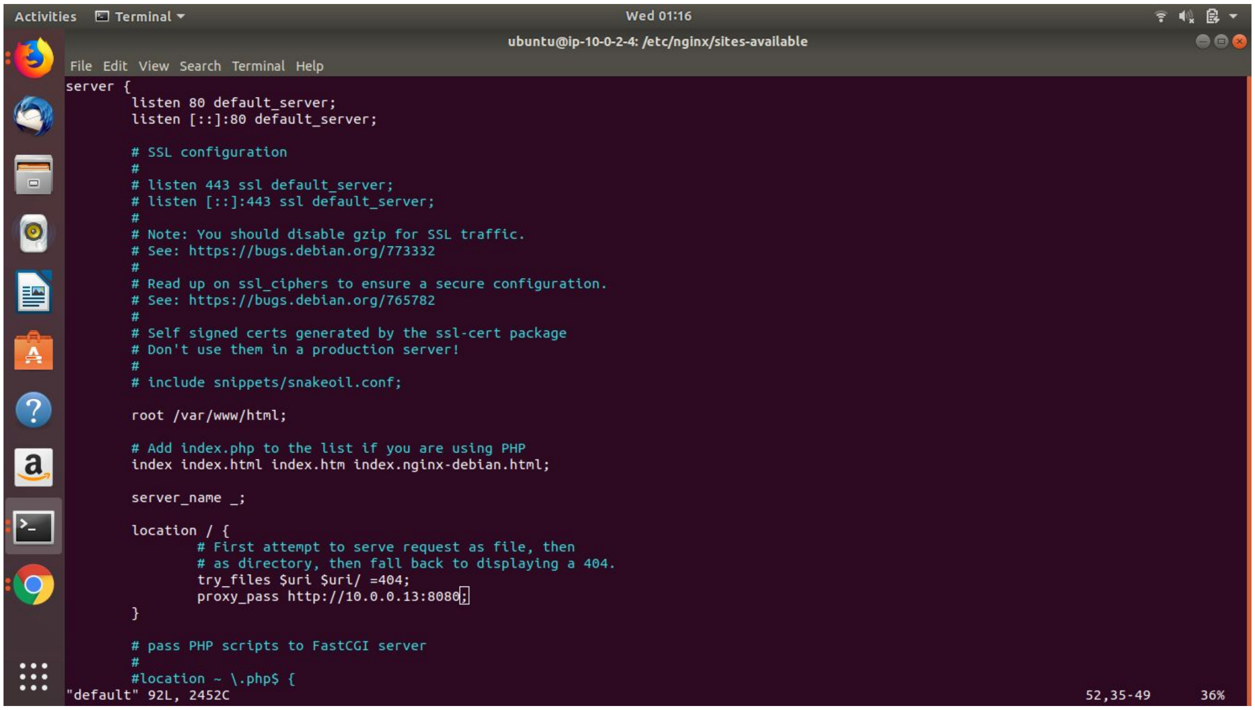
Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

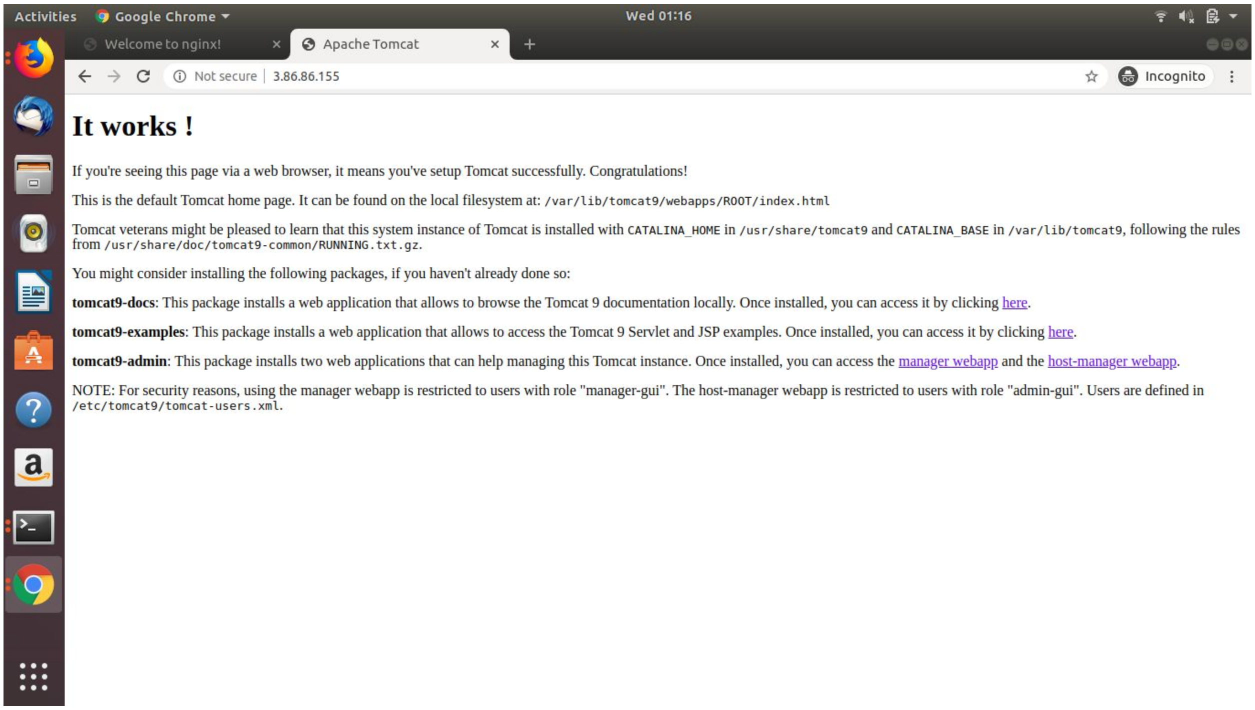
Thank you for using nginx.

STEP15: Now login back to the public instance and add proxy_pass in the location block of /etc/nginx/sites-available



```
server {
        listen 80 default_server;
        listen [::]:80 default_server;

        # SSL configuration
        #
        # listen 443 ssl default_server;
        # listen [::]:443 ssl default_server;
        #
        # Note: You should disable gzip for SSL traffic.
        # See: https://bugs.debian.org/773332
        #
        # Read up on ssl_ciphers to ensure a secure configuration.
        # See: https://bugs.debian.org/765782
        #
        # Self signed certs generated by the ssl-cert package
        # Don't use them in a production server!
        #
        # include snippets/snakeoil.conf;

        root /var/www/html;

        # Add index.php to the list if you are using PHP
        index index.html index.htm index.nginx-debian.html;

        server_name _;

        location / {
                # First attempt to serve request as file, then
                # as directory, then fall back to displaying a 404.
                try_files $uri $uri/ =404;
                proxy_pass http://10.0.0.13:8080;
        }

        # pass PHP scripts to FastCGI server
        #
        #location ~ \.php$ {
"default" 92L, 2452C                                                      52,35-49      36%
```

STEP16:Now reload nginx service and browse the public IP once again. This time it should display the tomcat9 page.



It works !

If you're seeing this page via a web browser, it means you've setup Tomcat successfully. Congratulations!

This is the default Tomcat home page. It can be found on the local filesystem at: `/var/lib/tomcat9/webapps/ROOT/index.html`

Tomcat veterans might be pleased to learn that this system instance of Tomcat is installed with `CATALINA_HOME` in `/usr/share/tomcat9` and `CATALINA_BASE` in `/var/lib/tomcat9`, following the rules from `/usr/share/doc/tomcat9-common/RUNNING.txt.gz`.

You might consider installing the following packages, if you haven't already done so:

tomcat9-docs: This package installs a web application that allows to browse the Tomcat 9 documentation locally. Once installed, you can access it by clicking here.

tomcat9-examples: This package installs a web application that allows to access the Tomcat 9 Servlet and JSP examples. Once installed, you can access it by clicking here.

tomcat9-admin: This package installs two web applications that can help managing this Tomcat instance. Once installed, you can access the manager webapp and the host-manager webapp.

NOTE: For security reasons, using the manager webapp is restricted to users with role "manager-gui". The host-manager webapp is restricted to users with role "admin-gui". Users are defined in `/etc/tomcat9/tomcat-users.xml`.