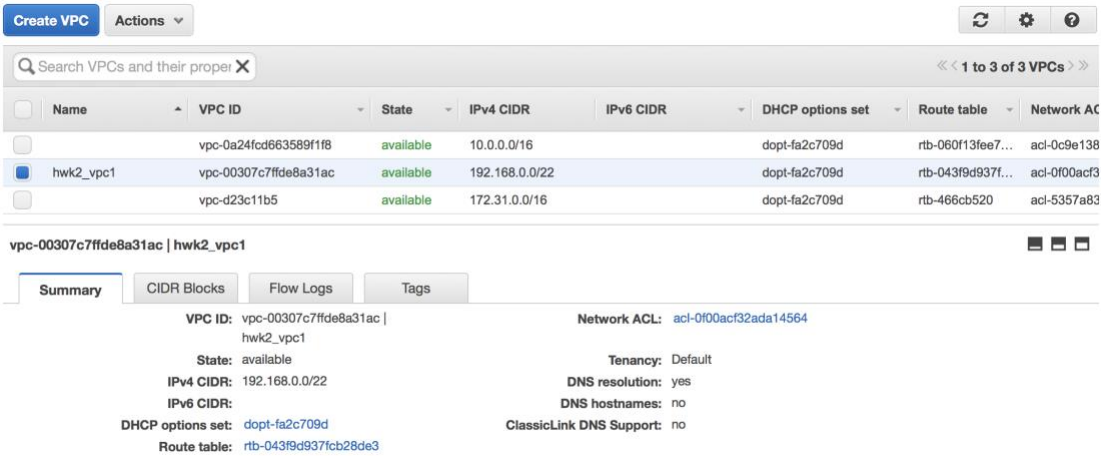
E91 Cloud DevOps: Fall 2018
Assignment 1
Stephen Akaeze
https://code.harvard.edu/sta283/cscie-91_sta283

Assignment Instruction Manual:
https://canvas.harvard.edu/courses/53026/files/6620943/download?wrap=1

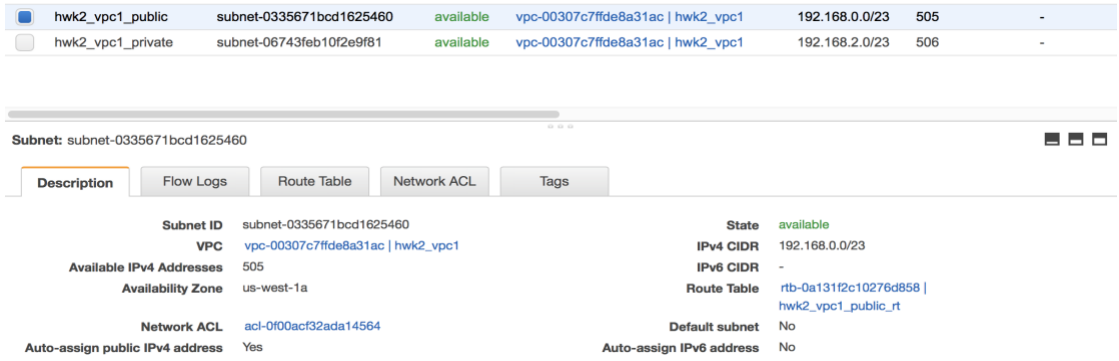
Problem 1:

1) Setup a VPC with appropriate tag/name of CDR of 192.168.0.0/22
    a. **Answer:** There are 1024 possible IPs in this CIDR of the selected VPC below
    b. **Result: The resulting VPC details are captured below**



2) Split the VPC CIDR equally amongst two subnets tagged, Public and private
    a. **Result: The resulting public and private subnet details are captured below**

Public Subnet



Private Subnet

hwk2_vpc1_private	subnet-06743feb10f2e9f81	available	vpc-00307c7ffde8a31ac | hwk2_vpc1	192.168.2.0/23	506	-

Subnet: subnet-06743feb10f2e9f81

Description | Flow Logs | Route Table | Network ACL | Tags

Subnet ID	subnet-06743feb10f2e9f81	**State**	available
VPC	vpc-00307c7ffde8a31ac | hwk2_vpc1	**IPv4 CIDR**	192.168.2.0/23
Available IPv4 Addresses	506	**IPv6 CIDR**	-
Availability Zone	us-west-1a	**Route Table**	rtb-0a79006348f131681 | hwk2_vpc1_private_rt
Network ACL	acl-0f00acf32ada14564	**Default subnet**	No
Auto-assign public IPv4 address	No	**Auto-assign IPv6 address**	No

3) Create a Nat gateway and Internet gateway

Result: The resulting NAT and Internet gateway details are captured below

Nat Gateway

Create NAT Gateway | Actions

Filter by tags and attributes or search by keyword — 1 to 2 of 2

Name	NAT Gateway ID	Status	Status Message	Elastic IP Address	Private IP Address	Network Interface	VPC
	nat-0a975683845...	available	-	52.8.207.86	192.168.0.41	eni-00eacef6a827...	vpc-00307
	nat-0f972dfcbca95...	deleted	-	54.193.49.246	192.168.1.61	eni-08d925fffe09d...	vpc-00307

NAT Gateway: nat-0a975683845dc1c25

Details | Monitoring | Tags

NAT Gateway ID	nat-0a975683845dc1c25	**Status**	available
Status Message	-	**Elastic IP Address**	52.8.207.86
Private IP Address	192.168.0.41	**Network Interface ID**	eni-00eacef6a827665bd
VPC	vpc-00307c7ffde8a31ac | hwk2_vpc1	**Subnet**	subnet-0335671bcd1625460 | hwk2_vpc1_public
Created	October 18, 2018 at 5:06:26 PM UTC-7	**Deleted**	-

Internet Gateway

Create internet gateway | Actions

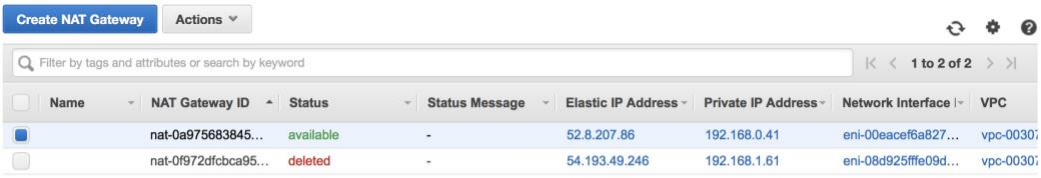
Filter by tags and attributes or search by keyword — 1 to 2 of 2

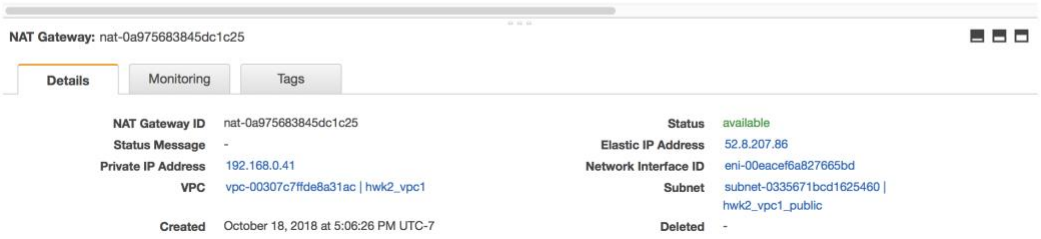
Name	ID	State	VPC
hwk2_vpc1_igw	igw-04b4ea84e13f...	detached	-
	igw-7be4c71f	attached	vpc-d23c11b5

4) Setup routes in route table so private subnet can access internet via NAT while public subnet has access via IGW

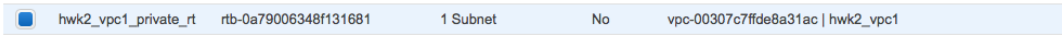
Result: The NAT gateway, internet gateway, routing tables and subnet configuration details are shown below

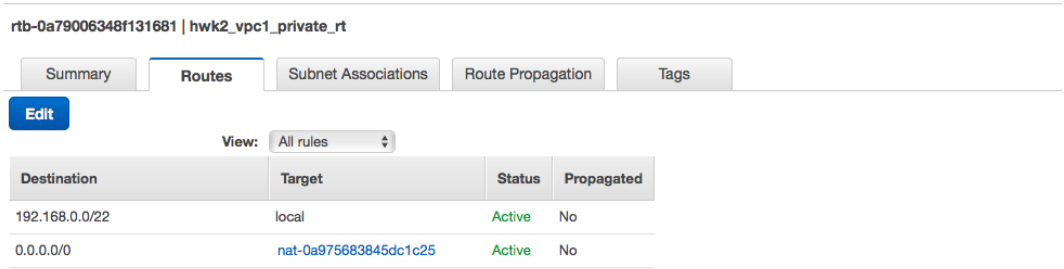
Nat gateway config showing 192.168.0.41 IP in public subnet





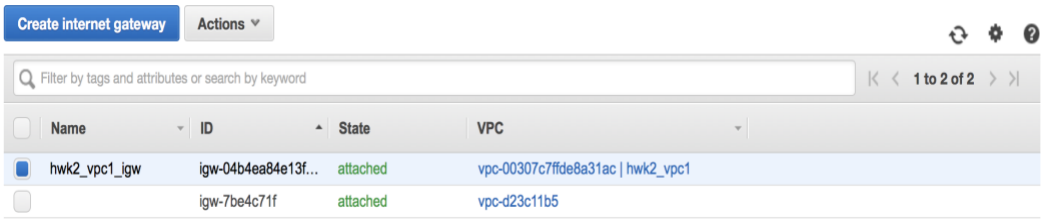
Private routing table showing internet request redirection via Nat gateway



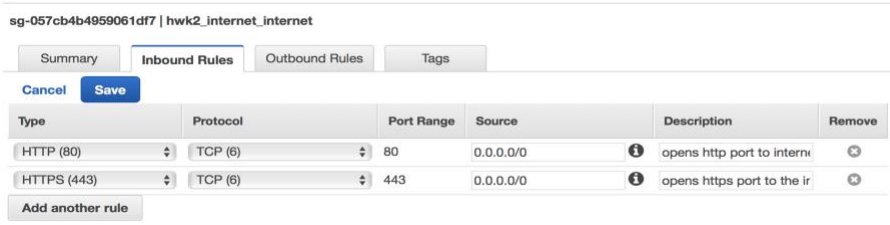


Private subnet showing attached private routing table

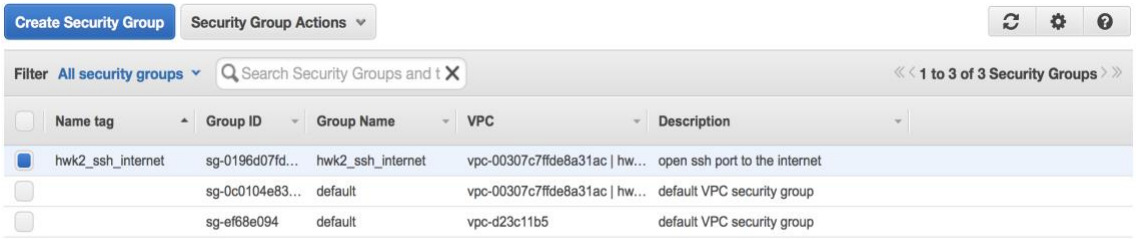
Internet gateway showing attachment to the VPC



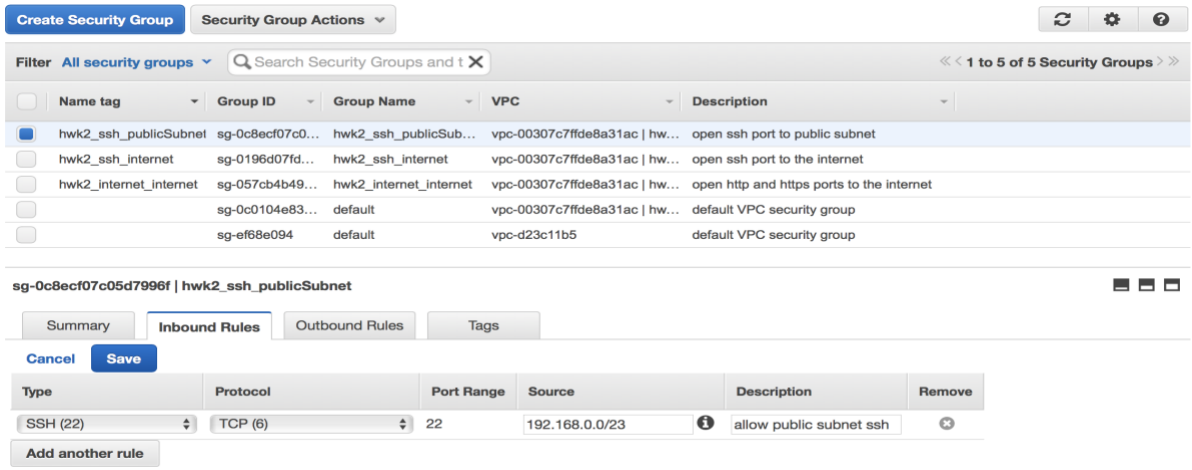
Security group for public instance for allowing inbound http and https connections



Security group for public instance for allowing inbound ssh connections



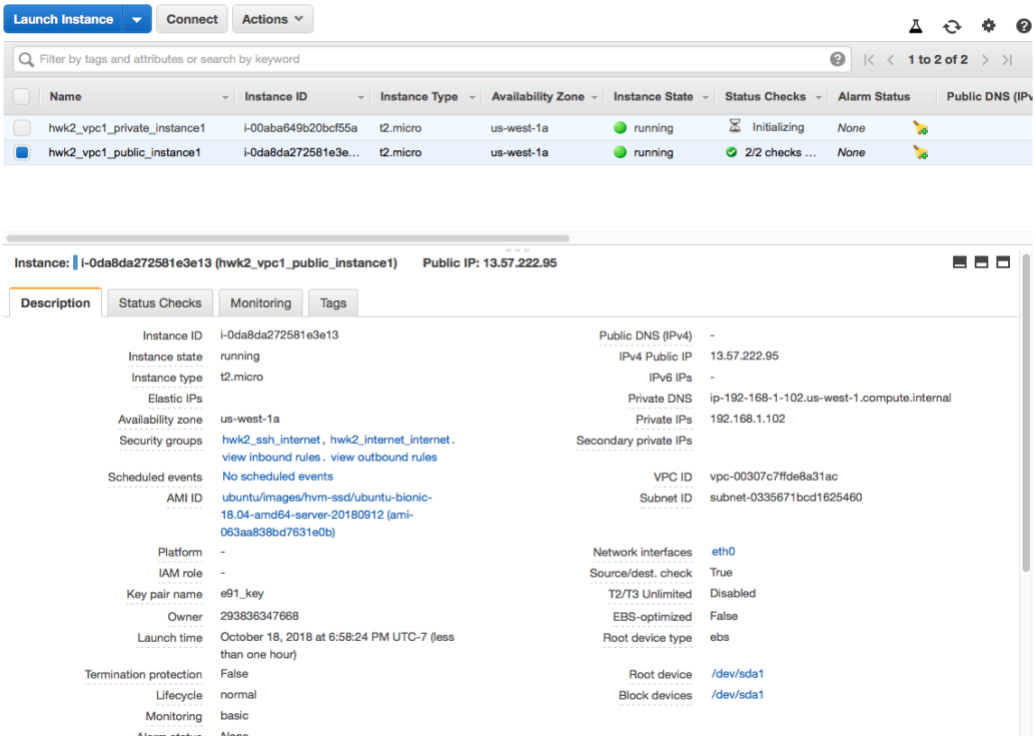
Security group for private instance for allowing inbound ssh connections from public subnet



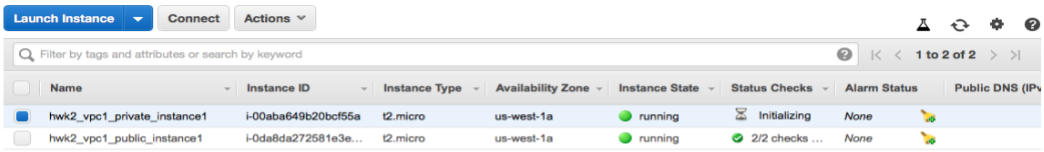
5) Create an EC2 instance in both public and private subnet. Then demonstrate locking when connecting to Internet from both instances

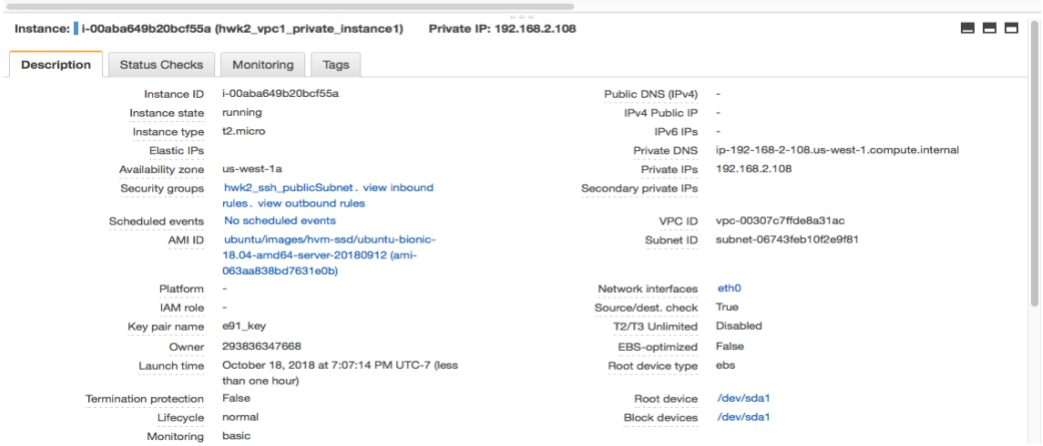
Result:

Public Instance showing security groups, public IP, private IP etc.



Private Instance showing security group, private IP etc.





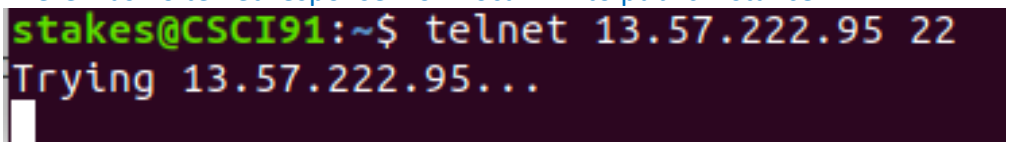
Locking demonstration:

Public Instance

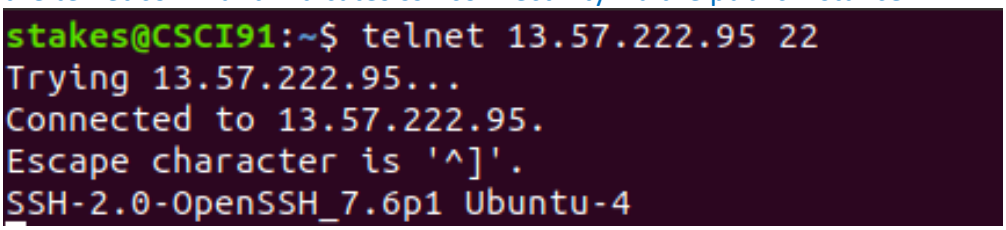
when ssh port security group is disabled on the public instance via the AWS console

☑	sg-057cb4b4959061df7	hwk2_internet_internet	open http and https ports to the internet
☐	sg-0196d07fd09191237	hwk2_ssh_internet	open ssh port to the internet

There was no telnet response from local VM to public instance



However, when the ssh port security group on the public instance is reenabled, the telnet command indicates ssh connectivity via the public instance IP



After accessing the public instance, the public instance also demonstrates internet connectivity via the “ping” command to google.com

```
ubuntu@ip-192-168-1-102:~$ ping google.com
PING google.com (172.217.0.46) 56(84) bytes of data.
64 bytes from lga15s43-in-f46.1e100.net (172.217.0.46): icmp_seq=1 ttl=48 time=1.87 ms
64 bytes from lga15s43-in-f46.1e100.net (172.217.0.46): icmp_seq=2 ttl=48 time=1.85 ms
64 bytes from lga15s43-in-f46.1e100.net (172.217.0.46): icmp_seq=3 ttl=48 time=1.86 ms
64 bytes from lga15s43-in-f46.1e100.net (172.217.0.46): icmp_seq=4 ttl=48 time=1.96 ms
^C
--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 1.856/1.890/1.969/0.070 ms
```

Private Instance

When the ssh port to public subnet security group is disabled on the private instance via AWS console

☐	sg-0c8ecf07c05d7996f	hwk2_ssh_publicSubnet	open ssh port to public subnet

There is no telnet response from the private instance

```
ubuntu@ip-192-168-1-102:~$ telnet 192.168.2.108 22
Trying 192.168.2.108...
```

However, when the ssh port to public subnet is reenabled, the telnet command indicates ssh connectivity via the private IP of the private instance.

```
ubuntu@ip-192-168-1-102:~$ telnet 192.168.2.108 22
Trying 192.168.2.108...
Connected to 192.168.2.108.
Escape character is '^]'.
SSH-2.0-OpenSSH_7.6p1 Ubuntu-4
```

The private instance also demonstrated internet connectivity via the pin g command

```
ubuntu@ip-192-168-2-108:~$ ping google.com
PING google.com (216.58.194.174) 56(84) bytes of data.
64 bytes from sfo07s13-in-f14.1e100.net (216.58.194.174): icmp_seq=1 ttl=47 time=2.46 ms
64 bytes from sfo07s13-in-f14.1e100.net (216.58.194.174): icmp_seq=2 ttl=47 time=2.06 ms
64 bytes from sfo07s13-in-f14.1e100.net (216.58.194.174): icmp_seq=3 ttl=47 time=2.08 ms
64 bytes from sfo07s13-in-f14.1e100.net (216.58.194.174): icmp_seq=4 ttl=47 time=2.16 ms
64 bytes from sfo07s13-in-f14.1e100.net (216.58.194.174): icmp_seq=5 ttl=47 time=2.16 ms
64 bytes from sfo07s13-in-f14.1e100.net (216.58.194.174): icmp_seq=6 ttl=47 time=2.08 ms
^C
--- google.com ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5007ms
rtt min/avg/max/mdev = 2.066/2.169/2.460/0.136 ms
ubuntu@ip-192-168-2-108:~$
```

Problem 2:

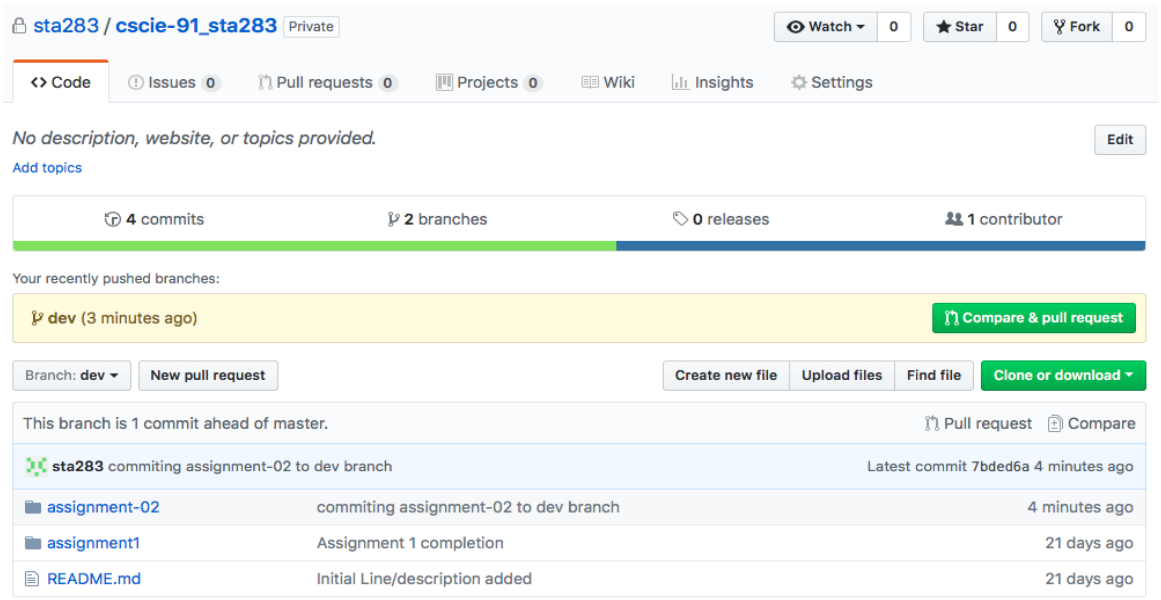
All commands were executed as captured in the snapshots below

```
stakes@CSCI91:~$ git clone git@code.harvard.edu:sta283/cscie-91_sta283.git
Cloning into 'cscie-91_sta283'...
remote: Counting objects: 13, done.
remote: Total 13 (delta 0), reused 0 (delta 0), pack-reused 13
Receiving objects: 100% (13/13), done.
stakes@CSCI91:~$ cd cscie-91_sta283/
stakes@CSCI91:~/cscie-91_sta283$ git branch dev
stakes@CSCI91:~/cscie-91_sta283$ git checkout dev
Switched to branch 'dev'
stakes@CSCI91:~/cscie-91_sta283$ mkdir assignment-02
stakes@CSCI91:~/cscie-91_sta283$ cd assignment-02/
stakes@CSCI91:~/cscie-91_sta283/assignment-02$ touch README.md
```

```
stakes@CSCI91:~/cscie-91_sta283$ git status
On branch dev
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        assignment-02/
```

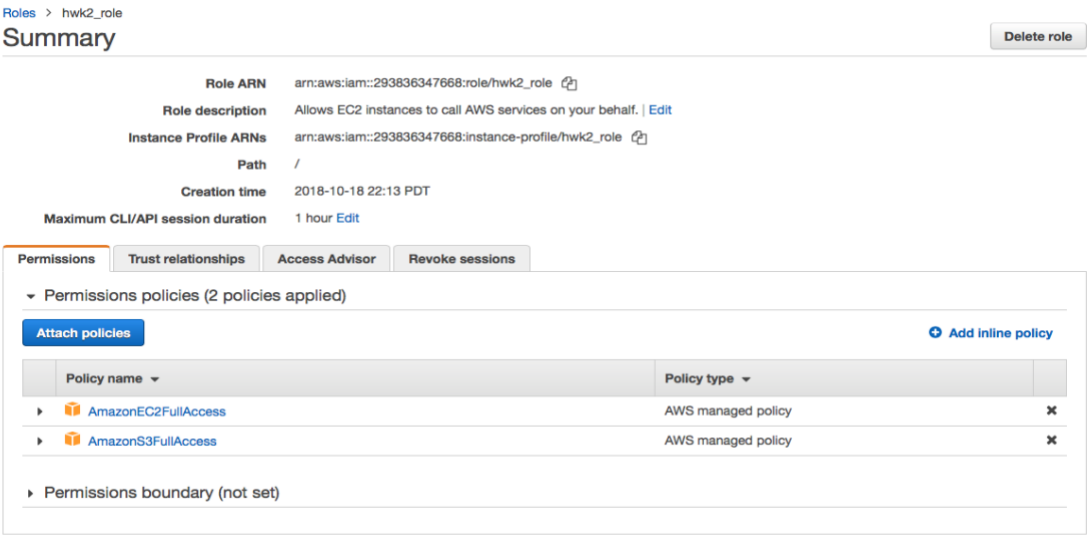
```
stakes@CSCI91:~/cscie-91_sta283$ git add .
stakes@CSCI91:~/cscie-91_sta283$ git status
On branch dev
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   assignment-02/README.md

stakes@CSCI91:~/cscie-91_sta283$ git commit -m "commiting assignment-02 to dev branch"
[dev 7bded6a] commiting assignment-02 to dev branch
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 assignment-02/README.md
stakes@CSCI91:~/cscie-91_sta283$ git push origin dev
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 377 bytes | 377.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To code.harvard.edu:sta283/cscie-91_sta283.git
 * [new branch]      dev -> dev
stakes@CSCI91:~/cscie-91_sta283$
```

The following are the GitHub snapshots demonstrating the successful “dev” branching, commit and push along with the new directory and file. Please visit https://code.harvard.edu/sta283/cscie-91_sta283 for more information

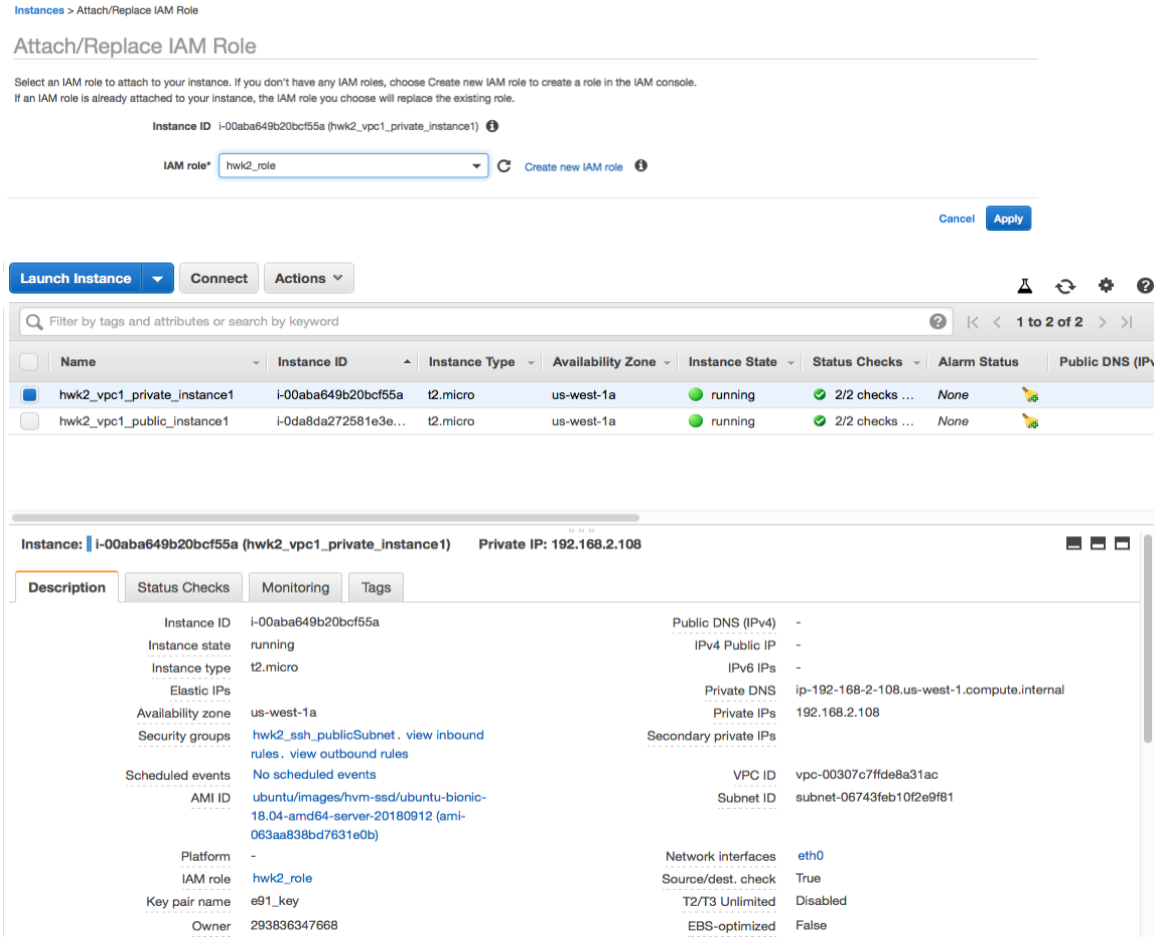


Problem 3:

1) Create an IAM Role with permission to fully manage EC2s and S3
    a. **Result:**



2) Assign role to the private instance
    a. **Result: Below shows the IAM role creation and attachment to the private instance.**



3) Log into the private instance.
    a. **Result: As shown below, the private key is copied to the public instance. Using ssh, I accessed the public instance. From the public instance and using ssh, I accessed the private instance**

```
stakes@CSCI91:~$ scp -i .ssh/e91_key.pem .ssh/e91_key.pem ubuntu@13.57.222.95:/home/ubuntu
e91_key.pem                                                   100% 1692    24.3KB/s   00:00
```

```
stakes@CSCI91:~$ ssh -i .ssh/e91_key.pem ubuntu@13.57.222.95
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-1021-aws x86_64)
```

```
ubuntu@ip-192-168-1-102:~$ ssh -i .ssh/e91_key.pem ubuntu@192.168.2.108
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-1021-aws x86_64)
```

4) Install AWSCLI
    a. **Result: Used "pip3 install awscli" to install awscli. Then used "aws configure" to set my availability zone to us-west-1a**

```
ubuntu@ip-192-168-2-108:~$ pip3 install awscli
Collecting awscli
  Downloading https://files.pythonhosted.org/packages/b4/06/b6e6bd72bdd89ecd2ada65e7678ba
b03f2cb3ee0d46e4662dfe959aaa3e0/awscli-1.16.37-py2.py3-none-any.whl (1.4MB)
    100% |                                | 1.4MB 1.1MB/s
Collecting botocore==1.12.27 (from awscli)
```

5) Clone your repository and switch to the branch dev, and create a file inside assignment-02, name it problem-03, commit and push the file to the dev remote.
    a. **Result: The instructions were all executed as shown by the screenshots below**

```
ubuntu@ip-192-168-2-108:~$ git clone git@code.harvard.edu:sta283/cscie-91_sta283.git
Cloning into 'cscie-91_sta283'...
remote: Counting objects: 17, done.
remote: Total 17 (delta 0), reused 0 (delta 0), pack-reused 17
Receiving objects: 100% (17/17), done.
Resolving deltas: 100% (1/1), done.
```

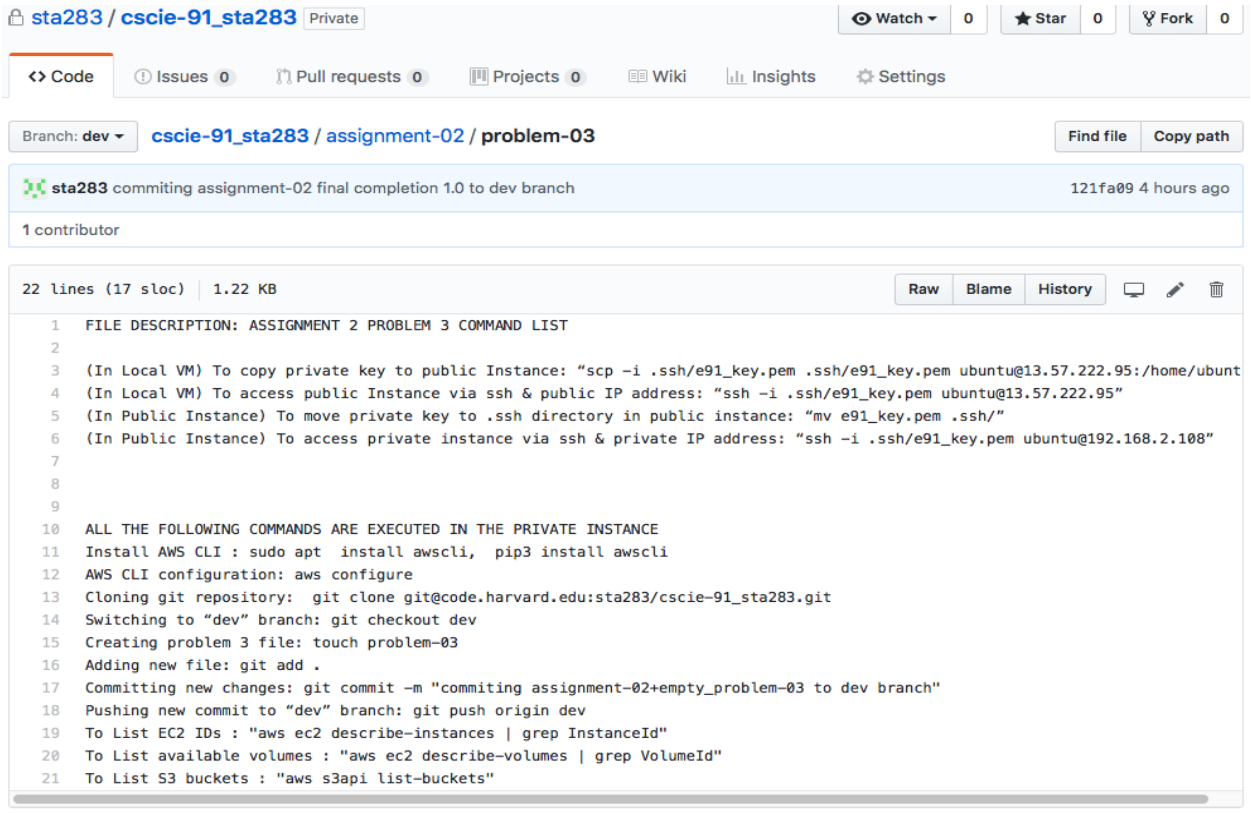
```
ubuntu@ip-192-168-2-108:~/cscie-91_sta283$ ll assignment-02/
total 8
drwxrwxr-x 2 ubuntu ubuntu 4096 Oct 19 06:07 ./
drwxrwxr-x 5 ubuntu ubuntu 4096 Oct 19 06:05 ../
-rw-rw-r-- 1 ubuntu ubuntu    0 Oct 19 06:05 README.md
-rw-rw-r-- 1 ubuntu ubuntu    0 Oct 19 06:07 problem-03
ubuntu@ip-192-168-2-108:~/cscie-91_sta283$ git status
On branch dev
Your branch is up to date with 'origin/dev'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        assignment-02/problem-03

nothing added to commit but untracked files present (use "git add" to track)
ubuntu@ip-192-168-2-108:~/cscie-91_sta283$ git add .
ubuntu@ip-192-168-2-108:~/cscie-91_sta283$ git status
On branch dev
Your branch is up to date with 'origin/dev'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   assignment-02/problem-03
```

```
ubuntu@ip-192-168-2-108:~/cscie-91_sta283$ git config --global user.email sta283@g.harvard.edu
ubuntu@ip-192-168-2-108:~/cscie-91_sta283$ git commit -m "commiting assignment-02+empty_problem-03 to dev branch"
[dev 4cd5b46] commiting assignment-02+empty_problem-03 to dev branch
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 assignment-02/problem-03
ubuntu@ip-192-168-2-108:~/cscie-91_sta283$ git push origin dev
Counting objects: 3, done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 388 bytes | 388.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To code.harvard.edu:sta283/cscie-91_sta283.git
   7bded6a..4cd5b46  dev -> dev
ubuntu@ip-192-168-2-108:~/cscie-91_sta283$
```

6) Use AWS CLI to list the IDs of EC2s, available volumes and S3 buckets
   a. **Results: the executed commands and results are shown below**

```
ubuntu@ip-192-168-2-108:~/cscie-91_sta283$ aws ec2 describe-instances | grep InstanceId
                    "InstanceId": "i-0da8da272581e3e13",
                    "InstanceId": "i-00aba649b20bcf55a",
ubuntu@ip-192-168-2-108:~/cscie-91_sta283$ aws ec2 describe-volumes | grep VolumeId
                    "VolumeId": "vol-0abb5cbb5b2213602",
            "VolumeId": "vol-0abb5cbb5b2213602",
                    "VolumeId": "vol-0884f9b2bb7030176",
            "VolumeId": "vol-0884f9b2bb7030176",
ubuntu@ip-192-168-2-108:~/cscie-91_sta283$ aws s3api list-buckets
{
    "Buckets": [],
    "Owner": {
        "DisplayName": "stephenakaeze",
        "ID": "933b0d8a05b1c4c8bb716a8010a6586b1954ec4cc11f76e591a0d0fa197d3120"
    }
}
```

7) Capture all above commands in problem 3 file
   a. **Results: the commands are captured in the file as shown below**



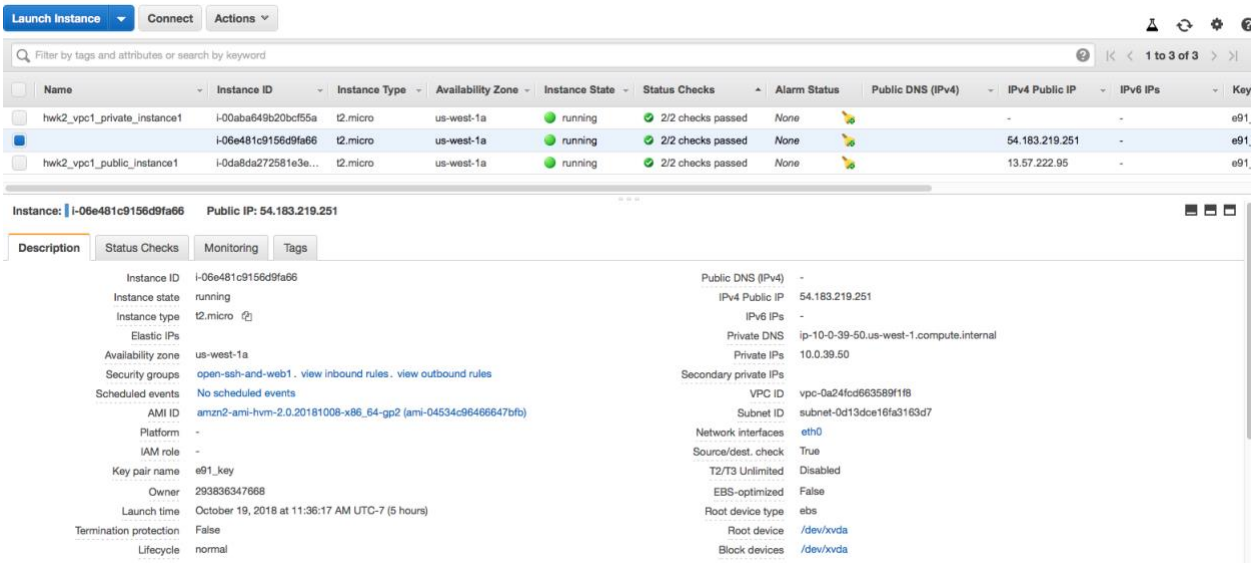
Problem 4:

Results: The original code from class came with bugs which were fixed and used the generate the required and specified network. Then it was stored as vpc-2subs-1ec2.sh in assignment-02 and pushed to dev. Below are individual proofs that the script was successful. Please visit https://code.harvard.edu/sta283/cscie-91_sta283/blob/dev/assignment-02/vpc-2subs-1ec2.sh to view the script details

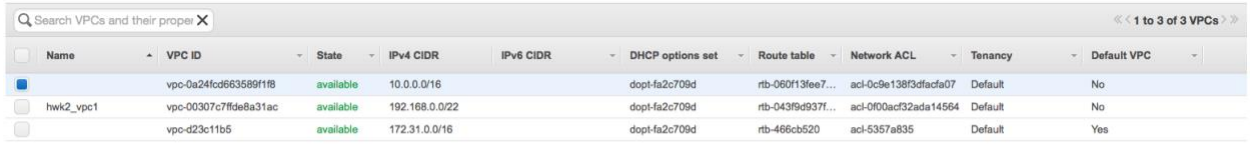
Console output after running script

```
ubuntu@ip-192-168-2-108:~/cscie-91_sta283/assignment-02$ ./vpc-2subs-1ec2.sh
{
    "Return": true
}
{
    "AssociationId": "rtbassoc-0407c94e3905f9293"
}
ubuntu@ip-192-168-2-108:~/cscie-91_sta283/assignment-02$
```

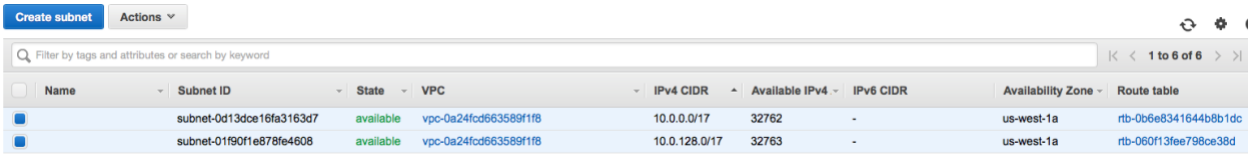
Instance created from script



VPC created from script



Subnets created from script



Create subnet | Actions

Filter by tags and attributes or search by keyword | 1 to 6 of 6

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR	Availability Zone	Route table
	subnet-0d13dce16fa3163d7	available	vpc-0a24fcd663589f1f8	10.0.0.0/17	32762	-	us-west-1a	rtb-0b6e8341644b8b1dc
	subnet-01f90f1e878fe4608	available	vpc-0a24fcd663589f1f8	10.0.128.0/17	32763	-	us-west-1a	rtb-060f13fee798ce38d

vpc-2subs-1ec2.sh @ code.harvard.edu

Branch: dev | cscie-91_sta283 / assignment-02 / vpc-2subs-1ec2.sh | Find file | Copy path

sta283 commiting assignment-02 completion to dev branch — 6748959 6 hours ago

1 contributor

Executable File | 28 lines (14 sloc) | 1.6 KB | Raw | Blame | History

```
1  #!/bin/bash
2
3  VPCID=$(aws ec2 create-vpc --cidr-block 10.0.0.0/16 --output text | grep VPC | awk '{print $7}')
4
5  PubSub=$(aws ec2 create-subnet --cidr-block 10.0.0.0/17 --vpc-id $VPCID --availability-zone us-west-1a --output text | awk '{pr
6
7  PriSub=$(aws ec2 create-subnet --cidr-block 10.0.128.0/17 --vpc-id $VPCID --availability-zone us-west-1a --output text | awk '{
8
9  IGW=$(aws ec2 create-internet-gateway | jq .InternetGateway.InternetGatewayId | tr -d '"')
10
11 aws ec2 attach-internet-gateway --vpc-id $VPCID --internet-gateway-id $IGW
12
13 PubRte=$(aws ec2 create-route-table --vpc-id $VPCID | grep rtb | awk '{print $2}'|tr -d '"'| tr -d ',')
14
15 aws ec2 create-route --route-table-id $PubRte --destination-cidr-block 0.0.0.0/0 --gateway-id $IGW
16
17 aws ec2 associate-route-table --route-table-id $PubRte --subnet-id $PubSub
18
19 SGID=$(aws ec2 create-security-group --description 'open ports 22 and 80 to the world' --group-name open-ssh-and-web1 --vpc-id
20
21 aws ec2 authorize-security-group-ingress --group-id $SGID --protocol tcp --port 80 --cidr 0.0.0.0/0
22
23 aws ec2 authorize-security-group-ingress --group-id $SGID --protocol tcp --port 22 --cidr 0.0.0.0/0
24
25 AMI=$(aws ec2 describe-images --owners amazon --filters 'Name=name,Values=amzn2-ami-hvm-2.0.????????-x86_64-gp2' 'Name=state,Va
26
27 EC2ID=$(aws ec2 run-instances --image-id $AMI --count 1 --instance-type t2.micro --key-name e91_key --security-group-ids $SGID
```