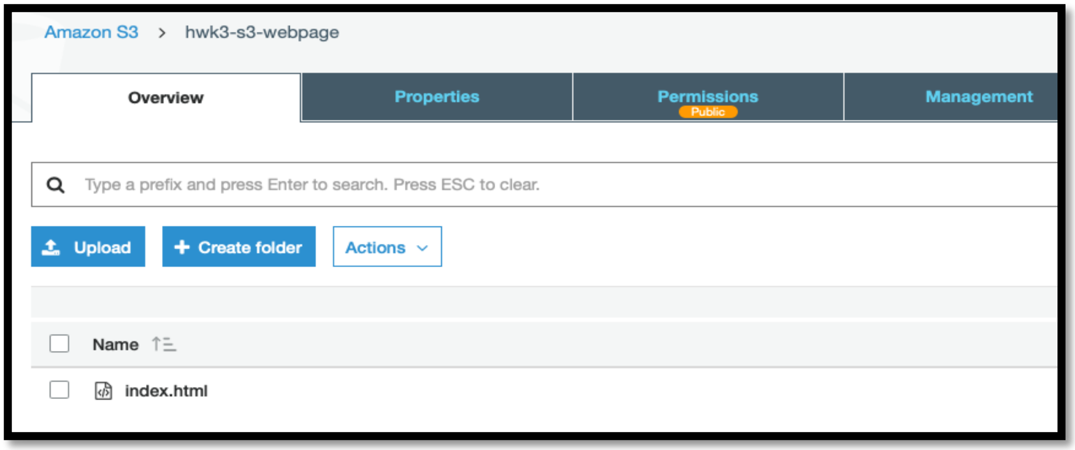
E91 Cloud Devops: Fall 2018
Assignment 3
Stephen Akaeze
https://code.harvard.edu/sta283/cscie-91_sta283

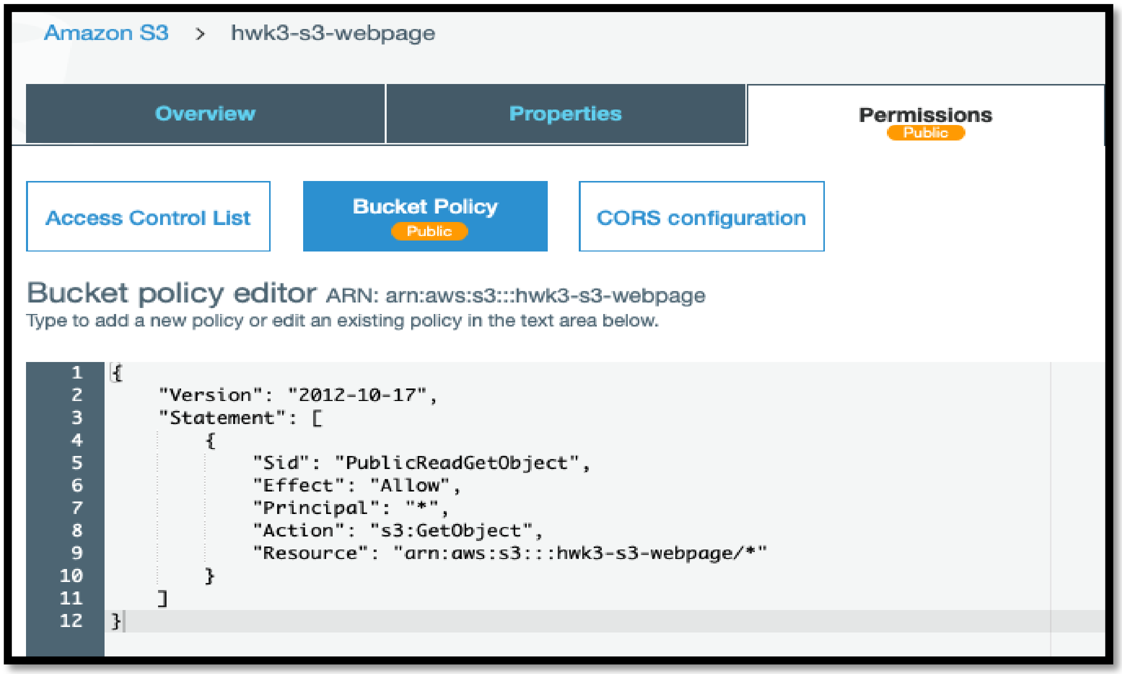
Assignment Instruction manual:
https://canvas.harvard.edu/courses/53026/files/6670661/download?verifier=ULySloPaZMkA0SRkueEVoHrmTzHWwfBhzVslrj2R&wrap=1

Problem 1:

1) The S3 bucket was created and named “hwk3-s3-webpage”. The S3 bucket’s property was set to host a static website. An index.html file containing two lines was created and uploaded to webpage.


Amazon S3 > hwk3-s3-webpage
Overview
Properties
Permissions
Public
Management
Type a prefix and press Enter to search. Press ESC to clear.
Upload
Create folder
Actions
Name
index.html

2) The S3 bucket policy was setup as public with the policy shown below



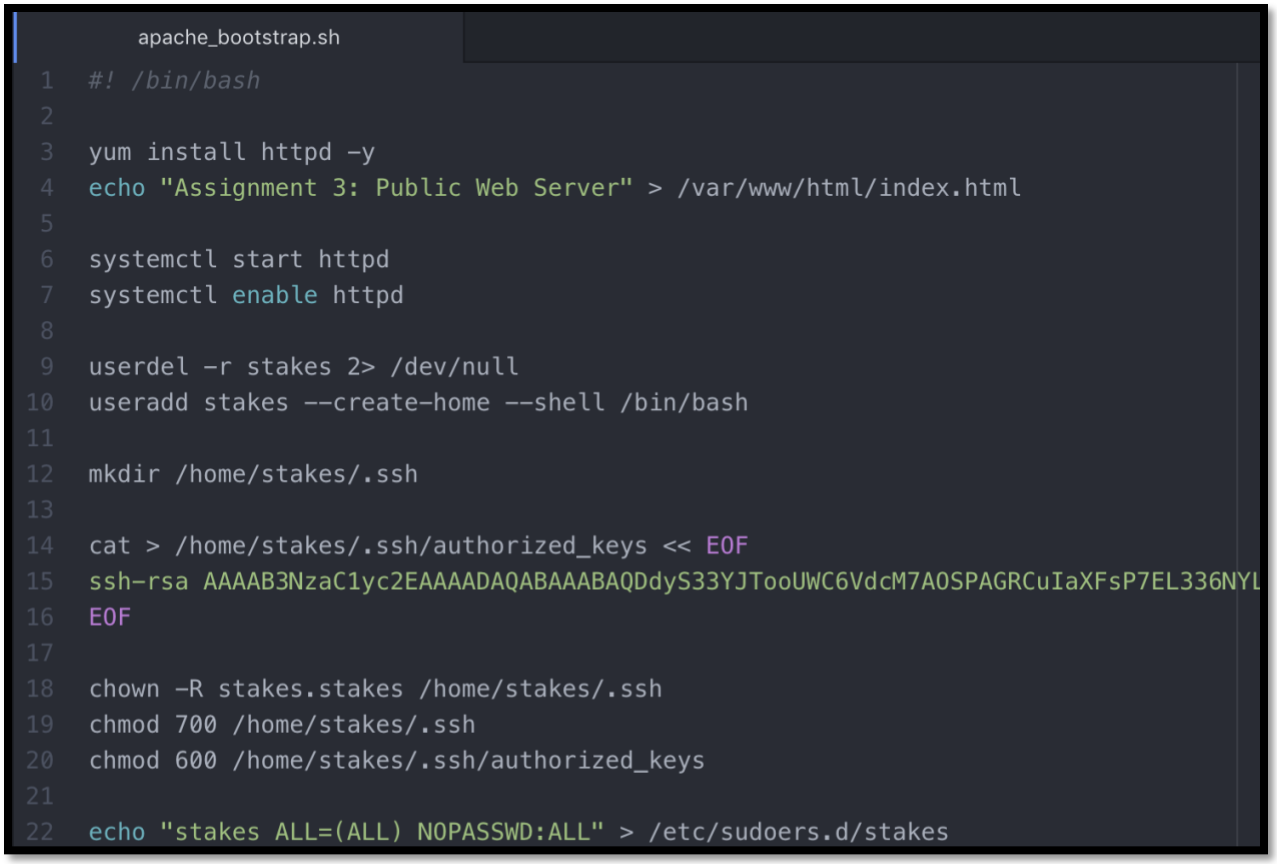
Bucket policy editor ARN: arn:aws:s3:::hwk3-s3-webpage
Type to add a new policy or edit an existing policy in the text area below.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PublicReadGetObject",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::hwk3-s3-webpage/*"
        }
    ]
}
```

3) The site was checked and it works. The site URL is http://hwk3-s3-webpage.s3-website-us-west-1.amazonaws.com

Problem 2

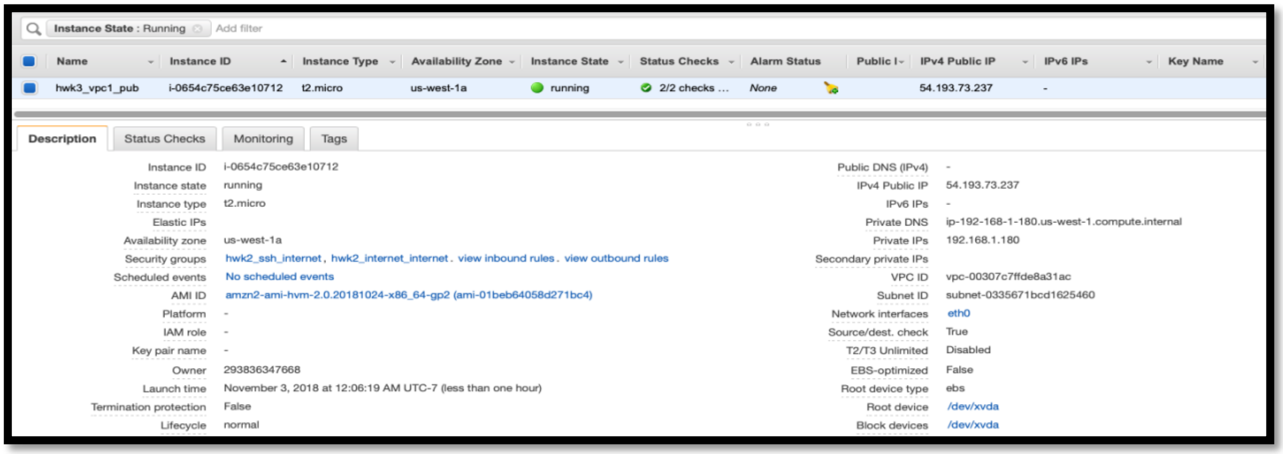
1) An instance with AWS Linux AMI was bootstrapped with the user data below and created in the public subnet created in homework 2. The ssh keys and new user implementation are also included in user data. Apache HTTP server was installed with the line "Assignment 3: Public Web Server" as also show below



apache_bootstrap.sh

```
#! /bin/bash

yum install httpd -y
echo "Assignment 3: Public Web Server" > /var/www/html/index.html

systemctl start httpd
systemctl enable httpd

userdel -r stakes 2> /dev/null
useradd stakes --create-home --shell /bin/bash

mkdir /home/stakes/.ssh

cat > /home/stakes/.ssh/authorized_keys << EOF
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDdyS33YJTooUWC6VdcM7AOSPAGRCuIaXFsP7EL336NYL
EOF

chown -R stakes.stakes /home/stakes/.ssh
chmod 700 /home/stakes/.ssh
chmod 600 /home/stakes/.ssh/authorized_keys

echo "stakes ALL=(ALL) NOPASSWD:ALL" > /etc/sudoers.d/stakes
```

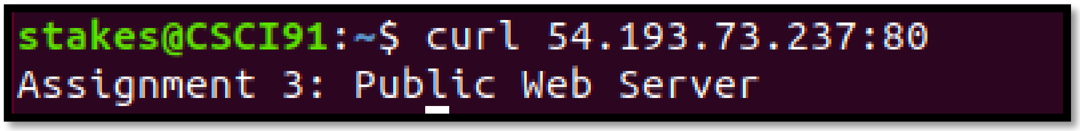
The EC2 instance(hwk3_vpc1_pub) was created and running



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public I	IPv4 Public IP	IPv6 IPs	Key Name
hwk3_vpc1_pub	i-0654c75ce63e10712	t2.micro	us-west-1a	● running	✓ 2/2 checks ...	None		54.193.73.237	-	

Instance ID	i-0654c75ce63e10712	Public DNS (IPv4)	-
Instance state	running	IPv4 Public IP	54.193.73.237
Instance type	t2.micro	IPv6 IPs	-
Elastic IPs		Private DNS	ip-192-168-1-180.us-west-1.compute.internal
Availability zone	us-west-1a	Private IPs	192.168.1.180
Security groups	hwk2_ssh_internet, hwk2_internet_internet. view inbound rules. view outbound rules	Secondary private IPs	
Scheduled events	No scheduled events	VPC ID	vpc-00307c7ffde8a31ac
AMI ID	amzn2-ami-hvm-2.0.20181024-x86_64-gp2 (ami-01beb64058d271bc4)	Subnet ID	subnet-0335671bcd1625460
Platform	-	Network interfaces	eth0
IAM role	-	Source/dest. check	True
Key pair name	-	T2/T3 Unlimited	Disabled
Owner	293836347668	EBS-optimized	False
Launch time	November 3, 2018 at 12:06:19 AM UTC-7 (less than one hour)	Root device type	ebs
Termination protection	False	Root device	/dev/xvda
Lifecycle	normal	Block devices	/dev/xvda

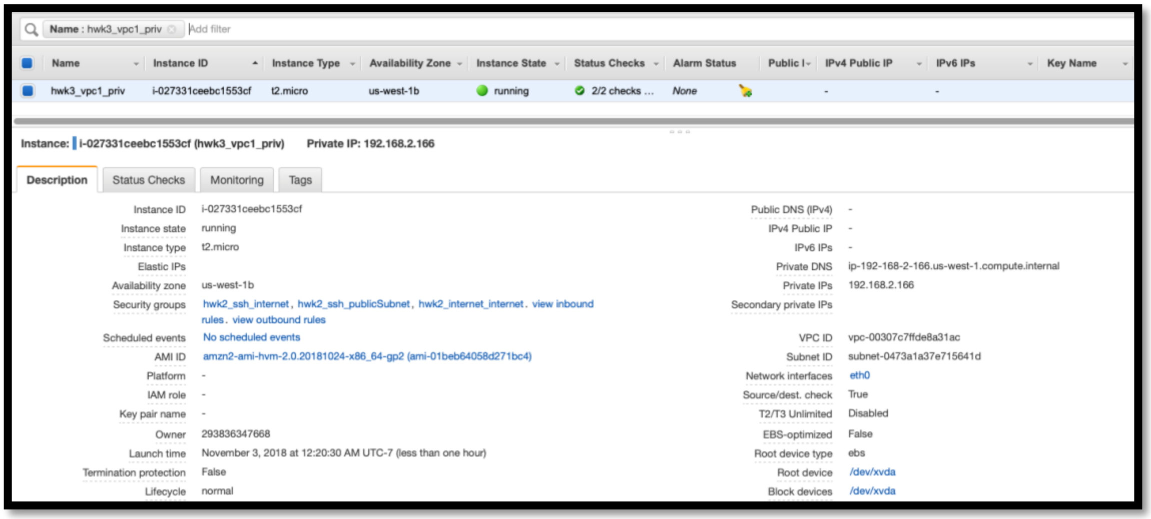
The webpage was checked and the apache server was running



```
stakes@CSCI91:~$ curl 54.193.73.237:80
Assignment 3: Public Web Server
```

Note: the assignment 2 private subnet was replaced with a new subnet in AZ **us-west-1b** While the public subnet remained in AZ **us-west-1a**. Thus, having two availability zones in the original VPC

The same above steps was used to create another instance in the private subnet in different AZ. The private instance apache server is only serving "Assignment 3: private Web Server". The instance was launched and it was working as shown below,



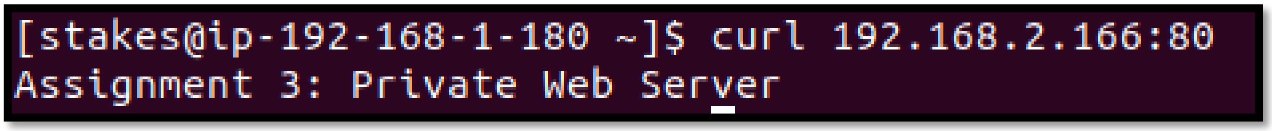
Name : hwk3_vpc1_priv | Add filter

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public I	IPv4 Public IP	IPv6 IPs	Key Name
hwk3_vpc1_priv	i-027331ceebc1553cf	t2.micro	us-west-1b	● running	✓ 2/2 checks ...	None		-	-	

Instance: i-027331ceebc1553cf (hwk3_vpc1_priv) Private IP: 192.168.2.166

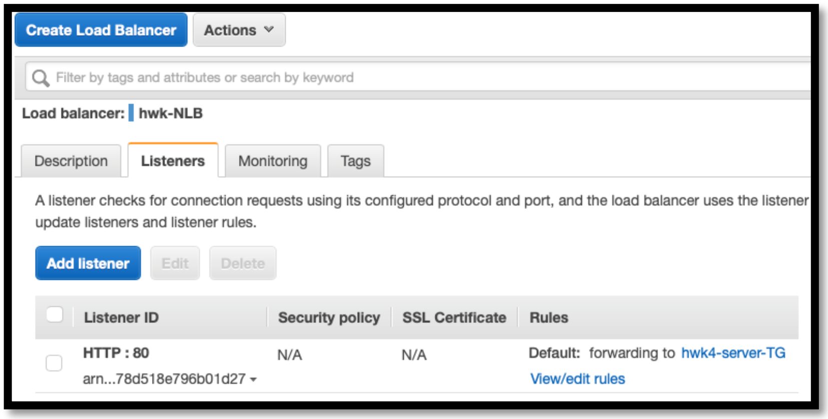
Instance ID	i-027331ceebc1553cf	Public DNS (IPv4)	-
Instance state	running	IPv4 Public IP	-
Instance type	t2.micro	IPv6 IPs	-
Elastic IPs		Private DNS	ip-192-168-2-166.us-west-1.compute.internal
Availability zone	us-west-1b	Private IPs	192.168.2.166
Security groups	hwk2_ssh_internet, hwk2_ssh_publicSubnet, hwk2_internet_internet. view inbound rules. view outbound rules	Secondary private IPs	
Scheduled events	No scheduled events	VPC ID	vpc-00307c7ffde8a31ac
AMI ID	amzn2-ami-hvm-2.0.20181024-x86_64-gp2 (ami-01beb64058d271bc4)	Subnet ID	subnet-0473a1a37e715641d
Platform	-	Network interfaces	eth0
IAM role	-	Source/dest. check	True
Key pair name	-	T2/T3 Unlimited	Disabled
Owner	293836347668	EBS-optimized	False
Launch time	November 3, 2018 at 12:20:30 AM UTC-7 (less than one hour)	Root device type	ebs
Termination protection	False	Root device	/dev/xvda
Lifecycle	normal	Block devices	/dev/xvda

Using ssh access to the public instance, the private server was verified as working



```
[stakes@ip-192-168-1-180 ~]$ curl 192.168.2.166:80
Assignment 3: Private Web Server
```

7) The network load balancer created using the application load balancer. It was placed in the HW2 VPC with Web port 80 open to incoming traffic. DNS Name is hwk-NLB-718786712.us-west-1.elb.amazonaws.com

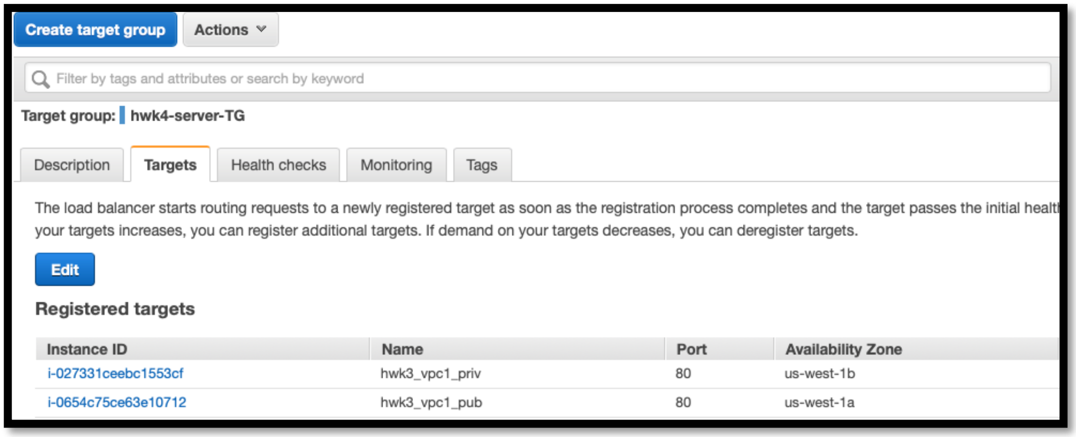


Load balancer: hwk-NLB

A listener checks for connection requests using its configured protocol and port, and the load balancer uses the listener
update listeners and listener rules.

Listener ID	Security policy	SSL Certificate	Rules
HTTP : 80 arn...78d518e796b01d27	N/A	N/A	Default: forwarding to hwk4-server-TG View/edit rules

The target group was also created as described in the instruction manual including the recently created public and private instances as its registered targets,



Target group: hwk4-server-TG

The load balancer starts routing requests to a newly registered target as soon as the registration process completes and the target passes the initial healt
your targets increases, you can register additional targets. If demand on your targets decreases, you can deregister targets.

Registered targets

Instance ID	Name	Port	Availability Zone
i-027331ceebc1553cf	hwk3_vpc1_priv	80	us-west-1b
i-0654c75ce63e10712	hwk3_vpc1_pub	80	us-west-1a

Loading details

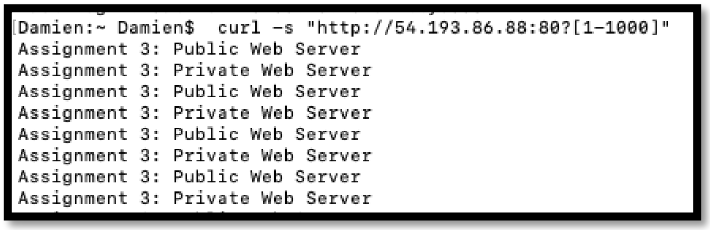
To confirm that the load balancer/target group setup works, a custom load balancing script was written. The load balancer IP address was recovered using

"host <dns name>"

With the IP address, my VM sends multi-process looped http requests to the load balancer's resolved IP via the "curl" command. I could not get JMeter to work so I improvised. The looped commands per process is shown below

curl -s [http://54.193.86.88:80?[1-1000]](http://54.193.86.88:80?[1-1000])

The result shows both machines responding to the load balancer's http requests

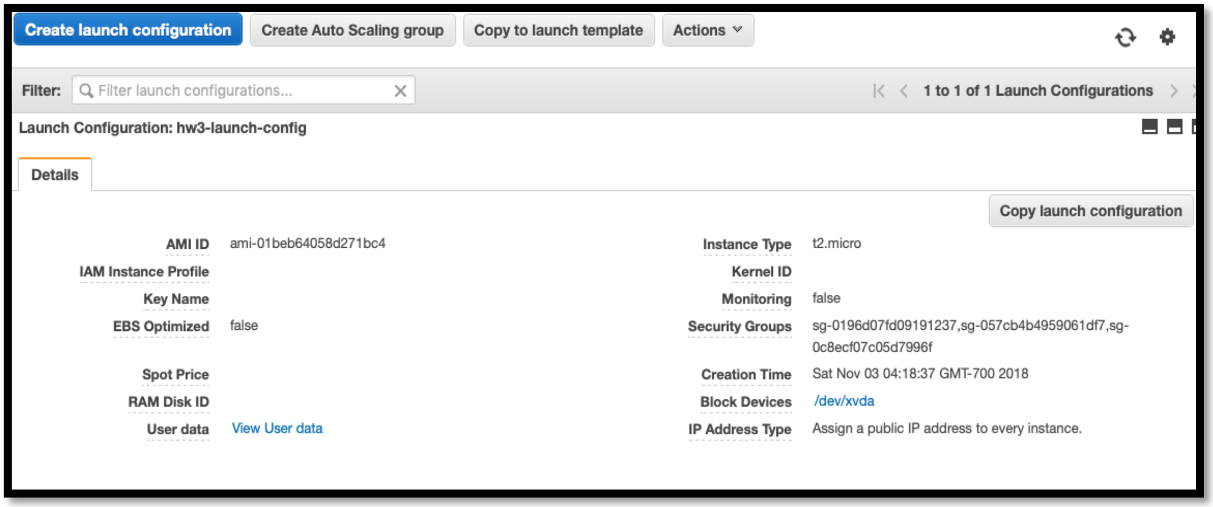


```
Damien:~ Damien$  curl -s "http://54.193.86.88:80?[1-1000]"
Assignment 3: Public Web Server
Assignment 3: Private Web Server
Assignment 3: Public Web Server
Assignment 3: Private Web Server
Assignment 3: Public Web Server
Assignment 3: Private Web Server
Assignment 3: Public Web Server
Assignment 3: Private Web Server
```

When the load goes up, the DNS name resolves to more IP addresses.

Problem 3

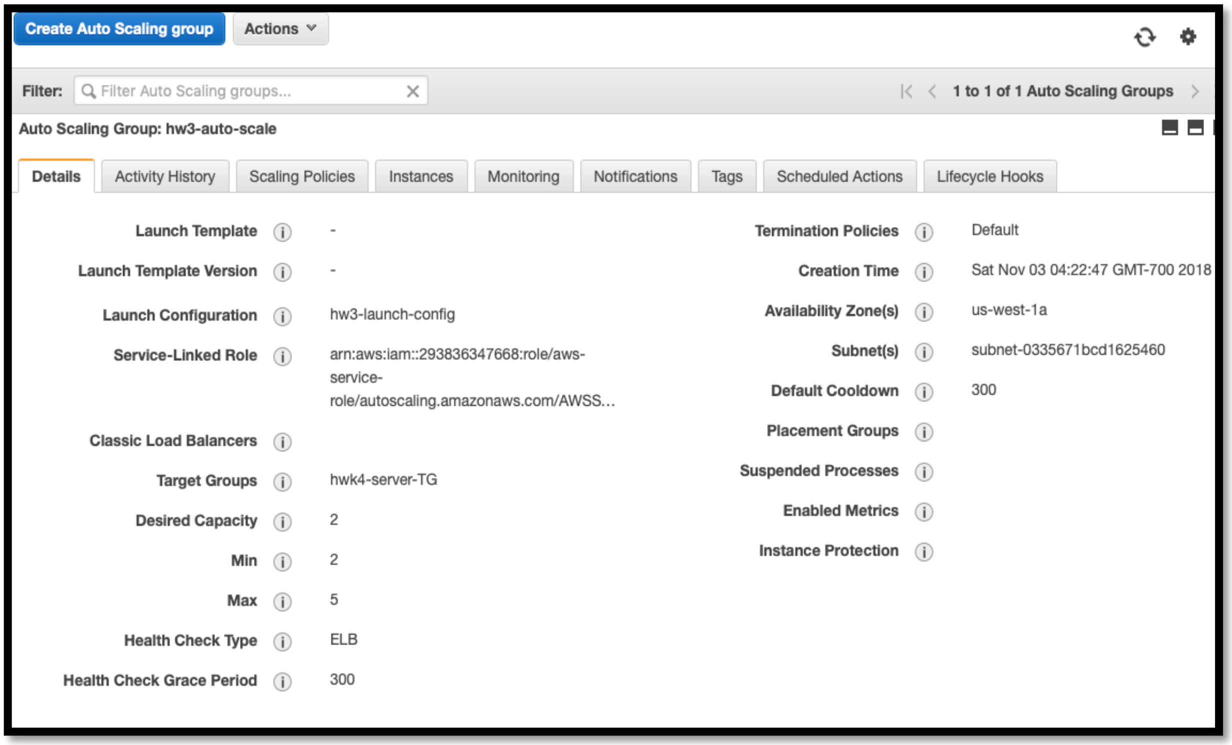
1) As specified by the instruction manual, the launch configuration and Auto Scaling were created as directed as shown below,



Create launch configuration
Create Auto Scaling group
Copy to launch template
Actions
Filter:
Filter launch configurations...
1 to 1 of 1 Launch Configurations
Details
Copy launch configuration

Launch Configuration: hw3-launch-config

AMI ID	ami-01beb64058d271bc4	Instance Type	t2.micro
IAM Instance Profile		Kernel ID	
Key Name		Monitoring	false
EBS Optimized	false	Security Groups	sg-0196d07fd09191237,sg-057cb4b4959061df7,sg-0c8ecf07c05d7996f
Spot Price		Creation Time	Sat Nov 03 04:18:37 GMT-700 2018
RAM Disk ID		Block Devices	/dev/xvda
User data	View User data	IP Address Type	Assign a public IP address to every instance.

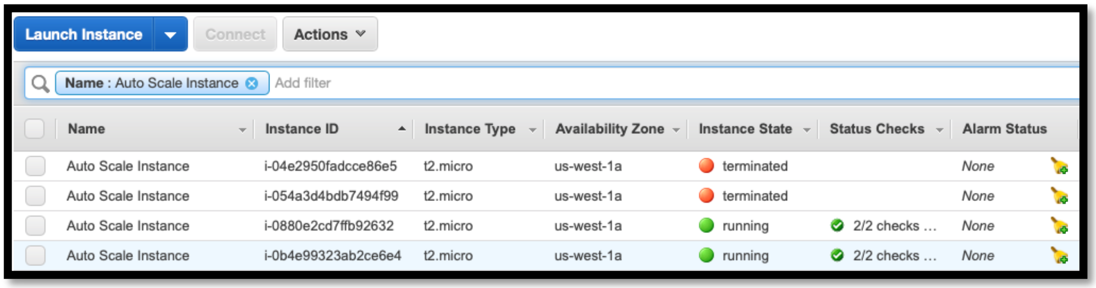


Auto Scaling Group: hw3-auto-scale

Launch Template	-	Termination Policies	Default
Launch Template Version	-	Creation Time	Sat Nov 03 04:22:47 GMT-700 2018
Launch Configuration	hw3-launch-config	Availability Zone(s)	us-west-1a
Service-Linked Role	arn:aws:iam::293836347668:role/aws-service-role/autoscaling.amazonaws.com/AWSS...	Subnet(s)	subnet-0335671bcd1625460
		Default Cooldown	300
Classic Load Balancers		Placement Groups	
Target Groups	hwk4-server-TG	Suspended Processes	
Desired Capacity	2	Enabled Metrics	
Min	2	Instance Protection	
Max	5		
Health Check Type	ELB		
Health Check Grace Period	300		

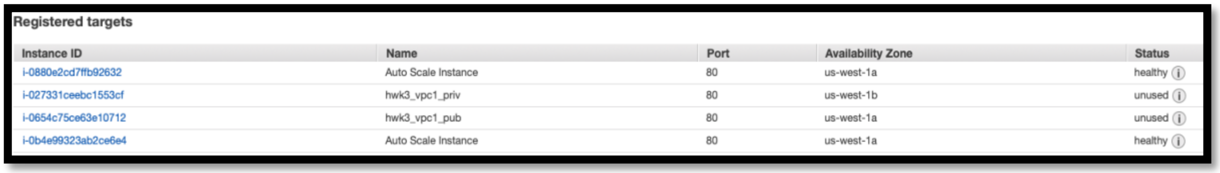
a. As instructed by the instruction manual, auto scaling starts with 2 instances. When one stance was terminated, the site did not go down because the second instance was still running. The site only goes down when both instances are terminated. After the single instance termination, the instance was immediately deleted form the target group as shown below.

Two auto scale instances are running fine



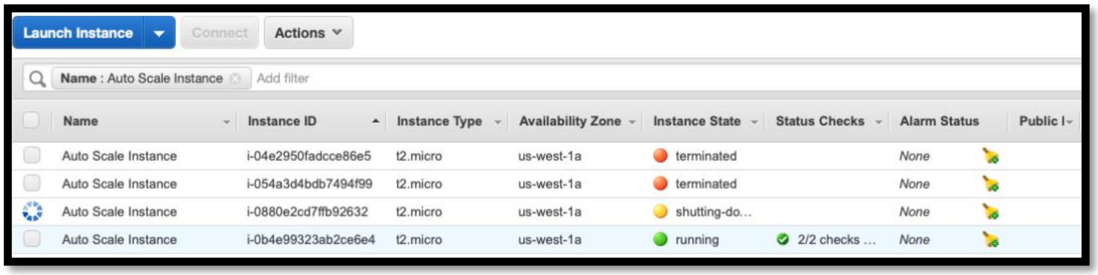
	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status
□	Auto Scale Instance	i-04e2950fadcce86e5	t2.micro	us-west-1a	● terminated		None
□	Auto Scale Instance	i-054a3d4bdb7494f99	t2.micro	us-west-1a	● terminated		None
□	Auto Scale Instance	i-0880e2cd7ffb92632	t2.micro	us-west-1a	● running	✓ 2/2 checks ...	None
□	Auto Scale Instance	i-0b4e99323ab2ce6e4	t2.micro	us-west-1a	● running	✓ 2/2 checks ...	None

Target group shows both running instances as healthy


Registered targets

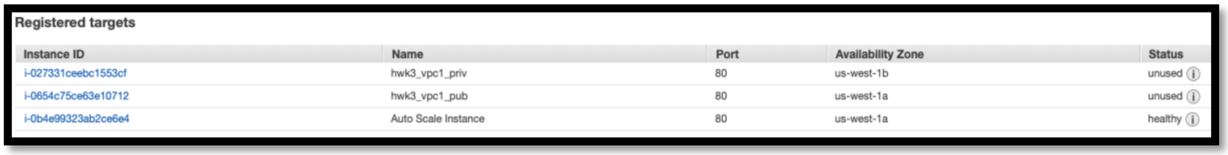
Instance ID	Name	Port	Availability Zone	Status
i-0880e2cd7ffb92632	Auto Scale Instance	80	us-west-1a	healthy ⓘ
i-027331ceebc1553cf	hwk3_vpc1_priv	80	us-west-1b	unused ⓘ
i-0654c75ce63e10712	hwk3_vpc1_pub	80	us-west-1a	unused ⓘ
i-0b4e99323ab2ce6e4	Auto Scale Instance	80	us-west-1a	healthy ⓘ

Then, one instance was terminated



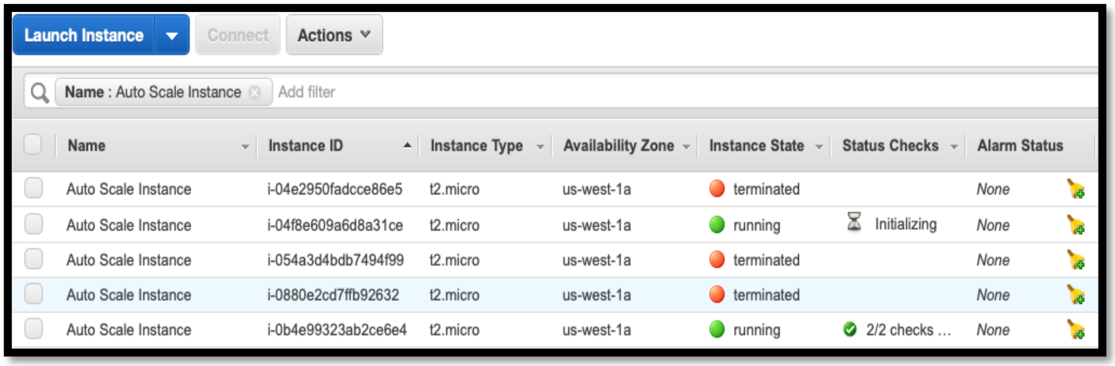
	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public I
□	Auto Scale Instance	i-04e2950fadcce86e5	t2.micro	us-west-1a	● terminated		None	
□	Auto Scale Instance	i-054a3d4bdb7494f99	t2.micro	us-west-1a	● terminated		None	
	Auto Scale Instance	i-0880e2cd7ffb92632	t2.micro	us-west-1a	● shutting-do...		None	
□	Auto Scale Instance	i-0b4e99323ab2ce6e4	t2.micro	us-west-1a	● running	✓ 2/2 checks ...	None	

The target group instantly deleted the terminated instance


Registered targets

Instance ID	Name	Port	Availability Zone	Status
i-027331ceebc1553cf	hwk3_vpc1_priv	80	us-west-1b	unused ⓘ
i-0654c75ce63e10712	hwk3_vpc1_pub	80	us-west-1a	unused ⓘ
i-0b4e99323ab2ce6e4	Auto Scale Instance	80	us-west-1a	healthy ⓘ

b. The autoscaling immediately created another instance in its place and the website is now accessible from both instances

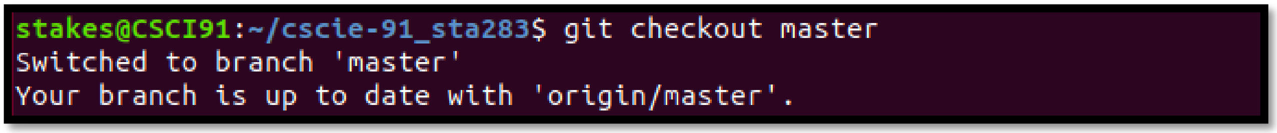


	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status
□	Auto Scale Instance	i-04e2950fadcce86e5	t2.micro	us-west-1a	● terminated		None
□	Auto Scale Instance	i-04f8e609a6d8a31ce	t2.micro	us-west-1a	● running	⌛ Initializing	None
□	Auto Scale Instance	i-054a3d4bdb7494f99	t2.micro	us-west-1a	● terminated		None
□	Auto Scale Instance	i-0880e2cd7ffb92632	t2.micro	us-west-1a	● terminated		None
□	Auto Scale Instance	i-0b4e99323ab2ce6e4	t2.micro	us-west-1a	● running	✓ 2/2 checks ...	None

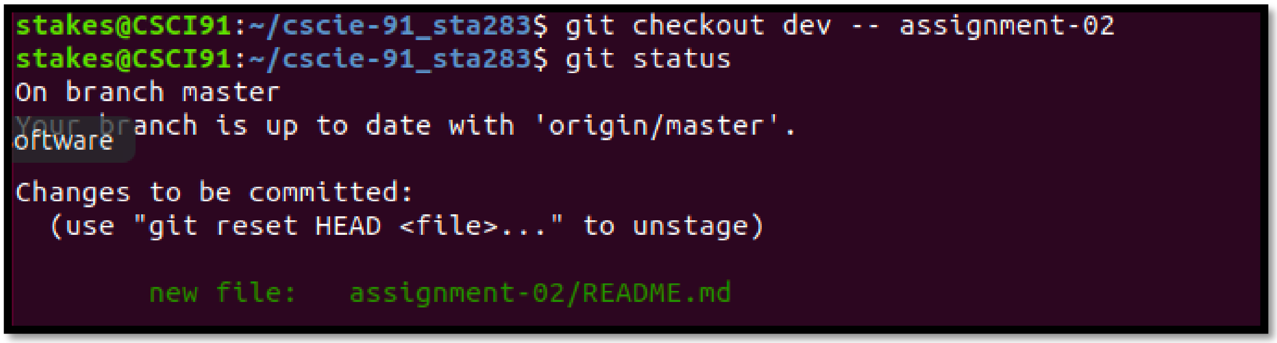
d. The Load was applied and an additional instance was created indicating that the 35% CPU load had been met by both load balancer target group instances

Problem 4:

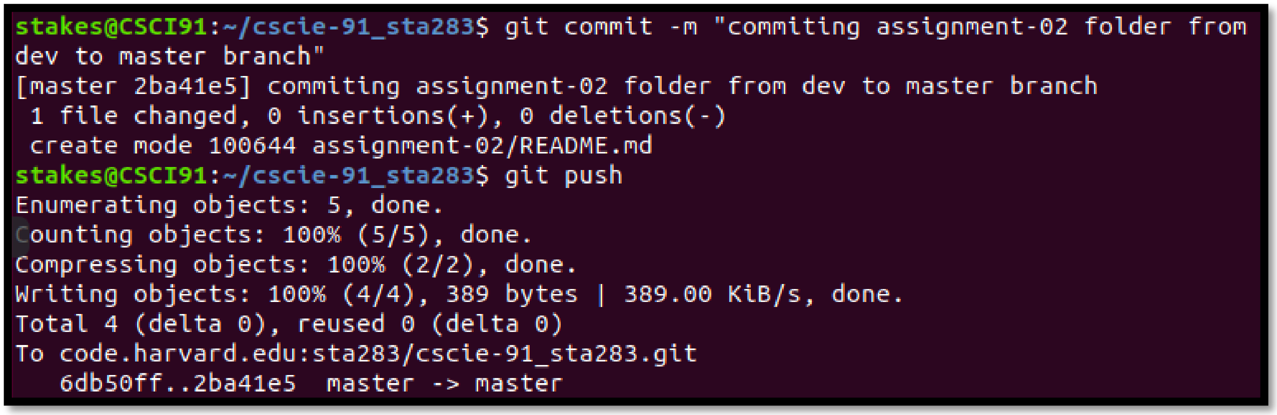
1) The "dev" branch assignment 2 folder merging to "master" branch is shown below



```
stakes@CSCI91:~/cscie-91_sta283$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
```

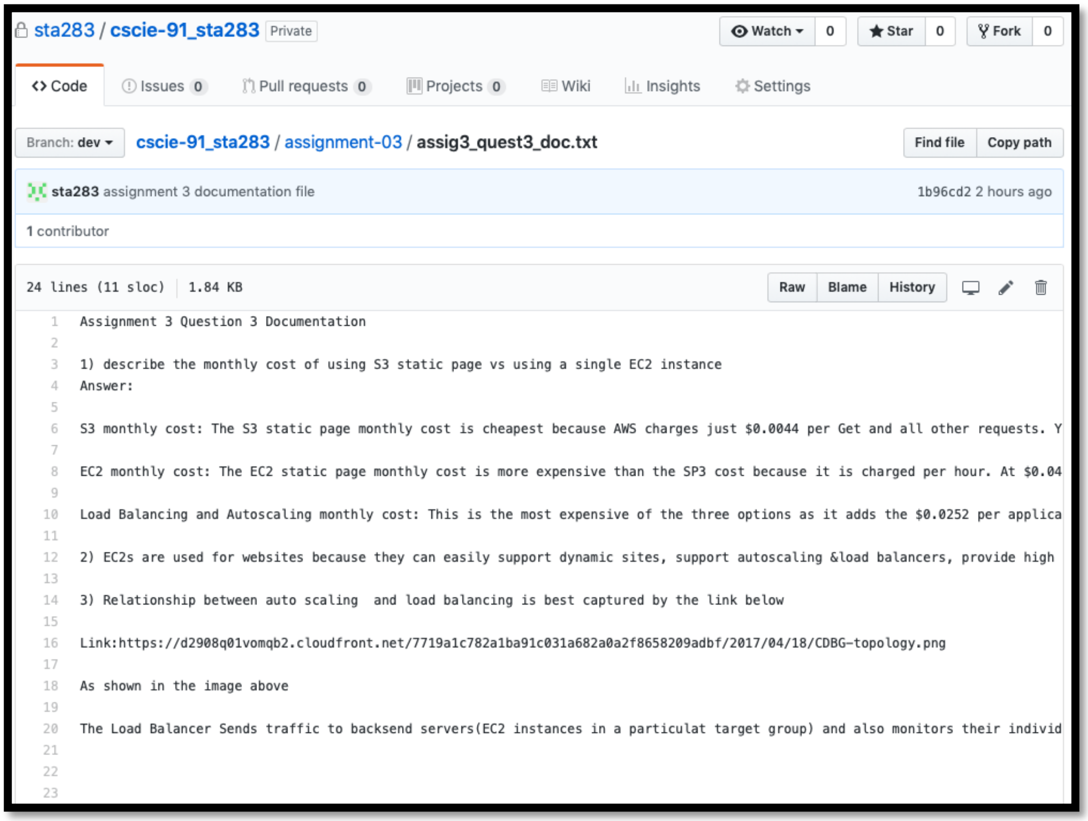


```
stakes@CSCI91:~/cscie-91_sta283$ git checkout dev -- assignment-02
stakes@CSCI91:~/cscie-91_sta283$ git status
On branch master
Your branch is up to date with 'origin/master'.

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 folder from
dev to master branch"
[master 2ba41e5] commiting assignment-02 folder from dev to master branch
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 assignment-02/README.md
stakes@CSCI91:~/cscie-91_sta283$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 389 bytes | 389.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To code.harvard.edu:sta283/cscie-91_sta283.git
   6db50ff..2ba41e5  master -> master
```

2) The assignment 3 folder was created in "dev" and the documentation text file (assig3_quest3_doc.txt) was added to the assignment 3 folder and pushed to code.harvard.edu (link: https://code.harvard.edu/sta283/cscie-91_sta283/blob/dev/assignment-03/assig3_quest3_doc.txt). The documentation file also contains as URL that helps demonstrate the relationship between autoscaling and load balancers. A screenshot of the text pushed assignment 3 folder and text file is shown below,



sta283 / cscie-91_sta283 Private

Branch: dev

cscie-91_sta283 / assignment-03 / assig3_quest3_doc.txt

sta283 assignment 3 documentation file 1b96cd2 2 hours ago

1 contributor

24 lines (11 sloc) 1.84 KB

```
Assignment 3 Question 3 Documentation

1) describe the monthly cost of using S3 static page vs using a single EC2 instance
Answer:

S3 monthly cost: The S3 static page monthly cost is cheapest because AWS charges just $0.0044 per Get and all other requests. Y

EC2 monthly cost: The EC2 static page monthly cost is more expensive than the SP3 cost because it is charged per hour. At $0.04

Load Balancing and Autoscaling monthly cost: This is the most expensive of the three options as it adds the $0.0252 per applica

2) EC2s are used for websites because they can easily support dynamic sites, support autoscaling &load balancers, provide high

3) Relationship between auto scaling  and load balancing is best captured by the link below

Link:https://d2908q01vomqb2.cloudfront.net/7719a1c782a1ba91c031a682a0a2f8658209adbf/2017/04/18/CDBG-topology.png

As shown in the image above

The Load Balancer Sends traffic to backsend servers(EC2 instances in a particulat target group) and also monitors their individ



```