AMAZON WEB SERVICES (AWS) CRASH COURSE

Izar Lab — August 2020

Jana Biermann, PhD
jb4424@cumc.columbia.edu

COLUMBIA | COLUMBIA UNIVERSITY IRVING MEDICAL CENTER

Overview

1. AWS introduction
2. Launching an instance
3. SSH into your instance
4. Update instance and install AWS CLI
5. Creating, attaching and mounting a volume
6. Set an alarm for your instance
7. Shut down your instance
8. Elastic Block Store (EBS) vs Simple Storage Service (S3)
9. Additional points

What is AWS?

aws training and certification

- AWS is a secure cloud platform that offers a broad set of global cloud-based products.
- AWS provides you with on-demand access to compute, storage, network, database, and other IT resources and management tools.
- AWS offers flexibility.
- You pay only for the individual services you need, for as long as you use them.
- AWS services work together like building blocks.

AWS Global Infrastructure

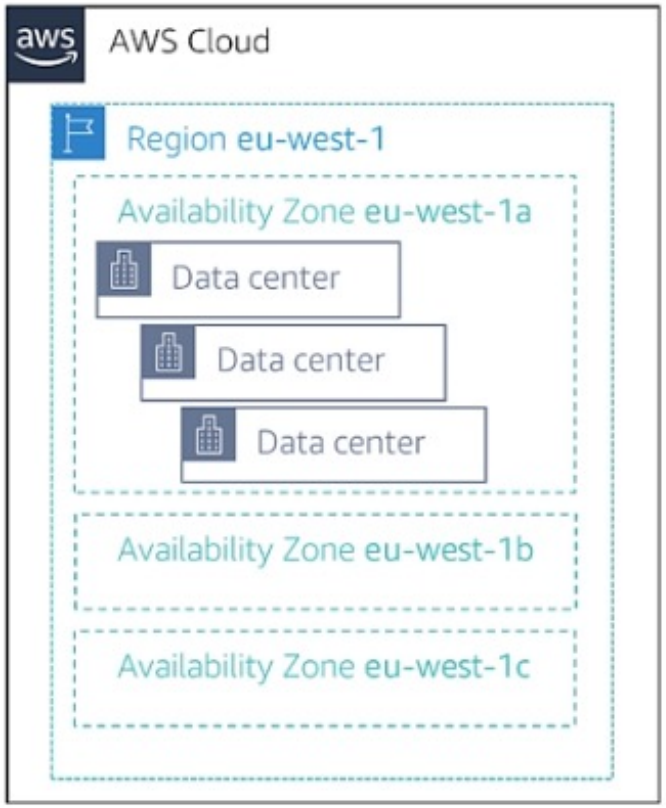
aws training and certification

- The AWS Global Infrastructure is designed and built to deliver a flexible, reliable, scalable, and secure cloud computing environment with high-quality global network performance.
- This map shows the current AWS Regions and more that are coming soon.

Availability Zones

aws training and certification

- Each Region has multiple Availability Zones.
- Each Availability Zone is a fully isolated partition of the AWS infrastructure.
  - There are currently 69 Availability Zones worldwide
  - Availability Zones consist of discrete data centers
  - They are designed for fault isolation
  - They are interconnected with other Availability Zones by using high-speed private networking
  - You choose your Availability Zones.
  - AWS recommends replicating data and resources across Availability Zones for resiliency.



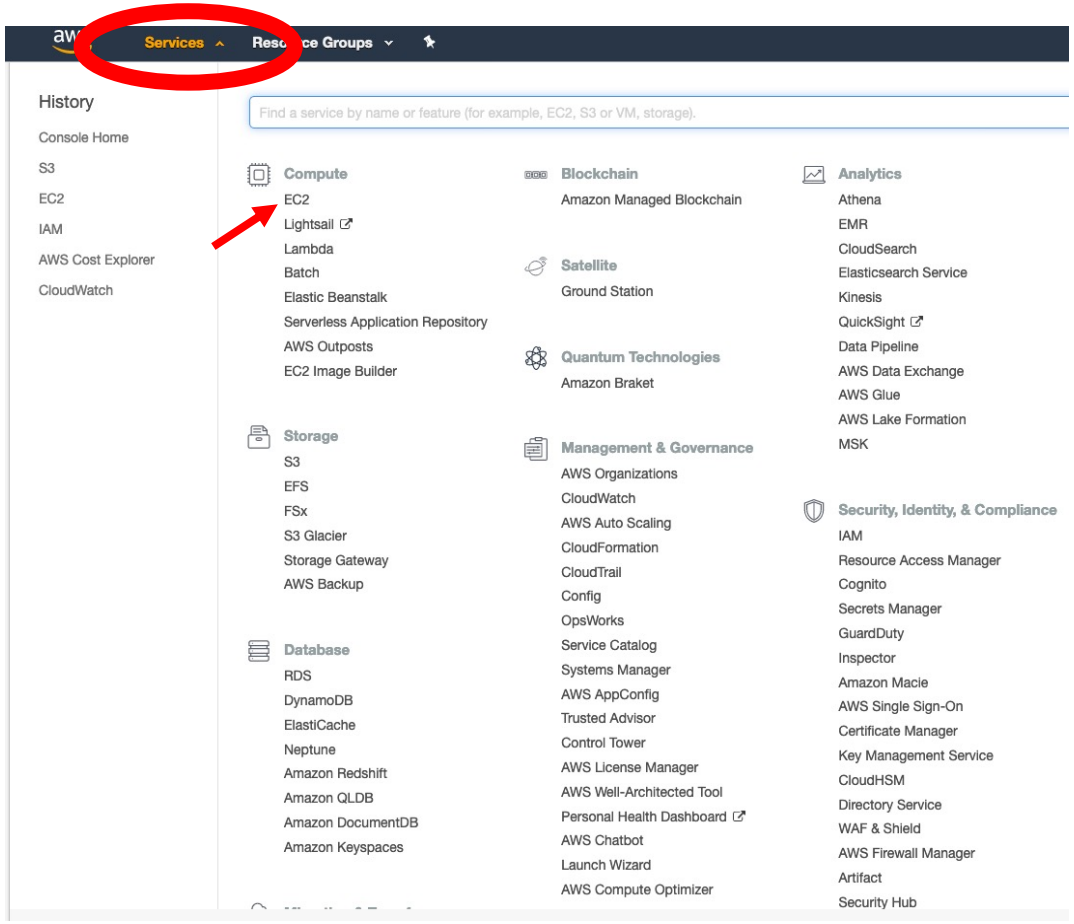
- You need to stay within the region us-east-2 (Ohio)
- Availability Zones can be freely chosen

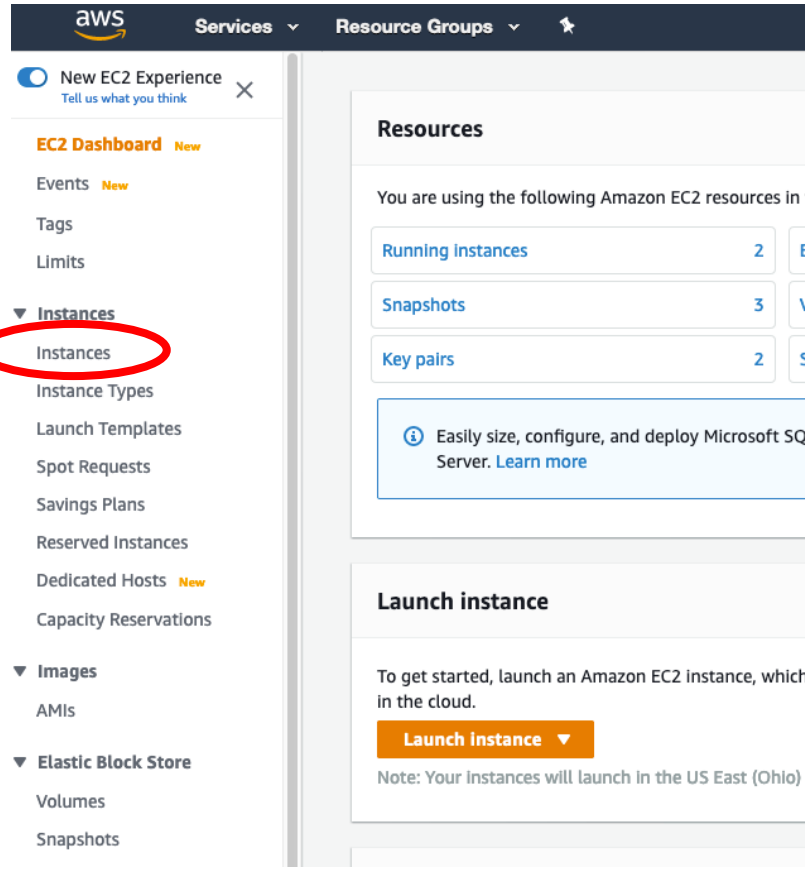
15

Sign in

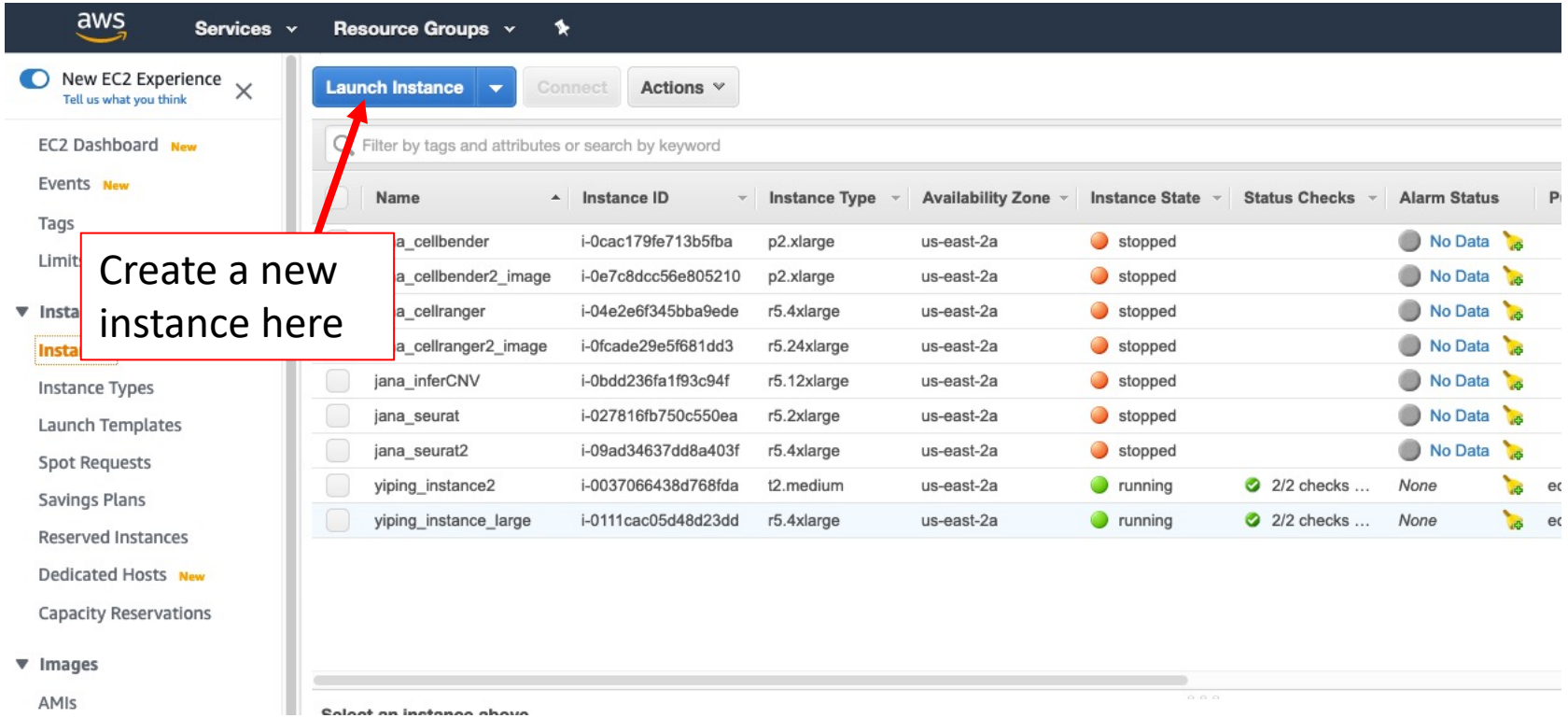
https://us-east-2.signin.aws.amazon.com

Navigate to instances



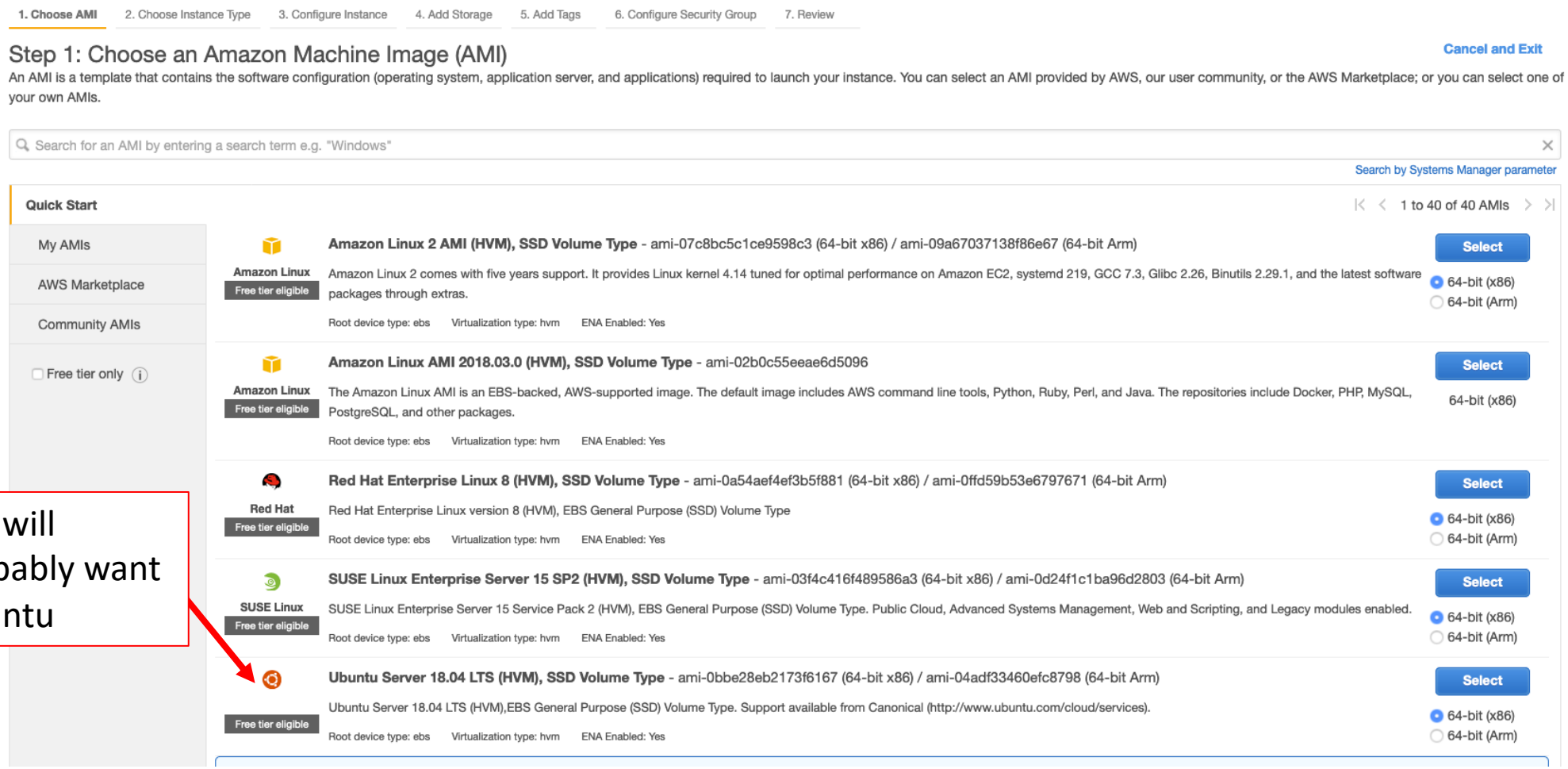


Launching an instance



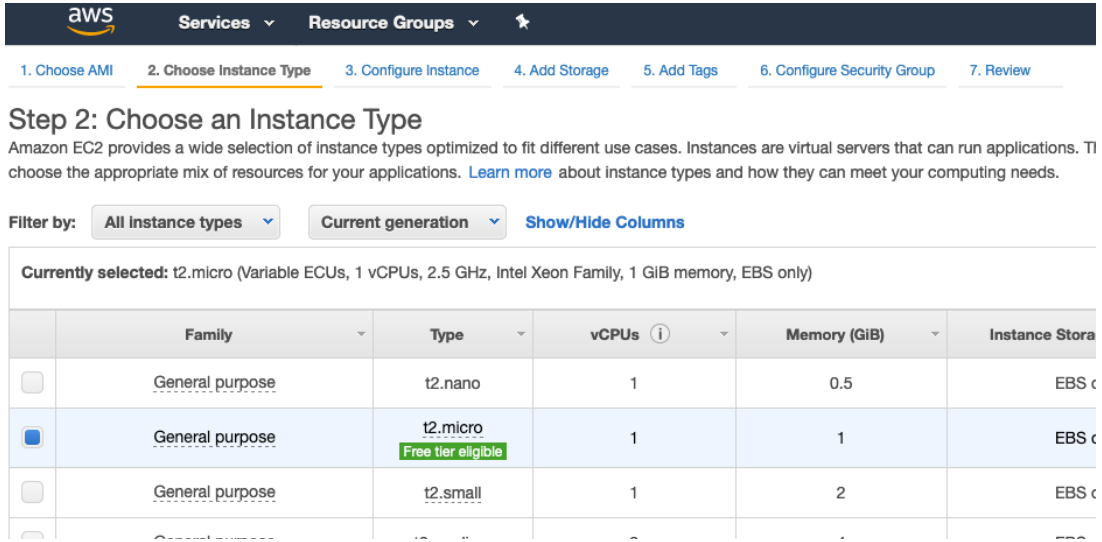
- Name your instances and volumes starting with your name
- E.g. ‘yiping_instance_large’

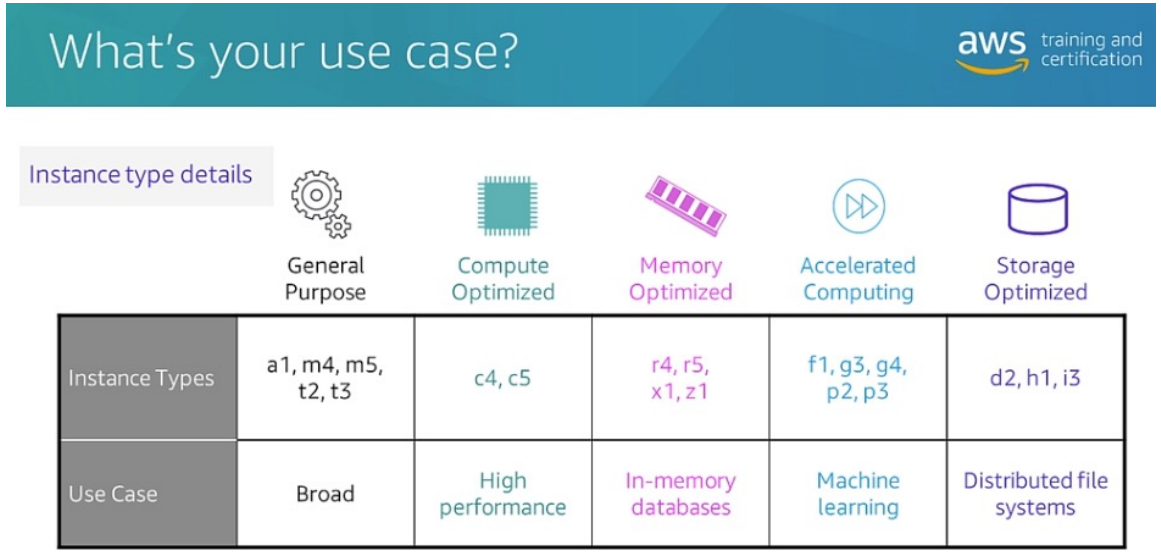
Launching an instance — Step 1



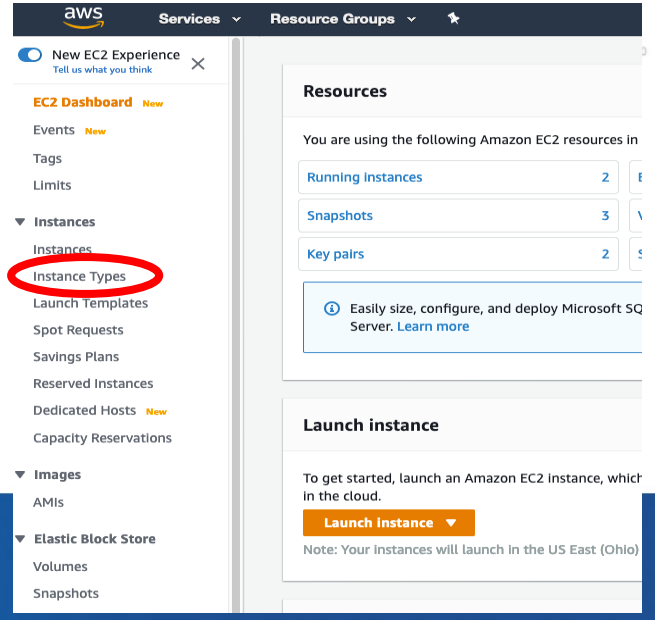
You will probably want Ubuntu

Launching an instance — Step 2

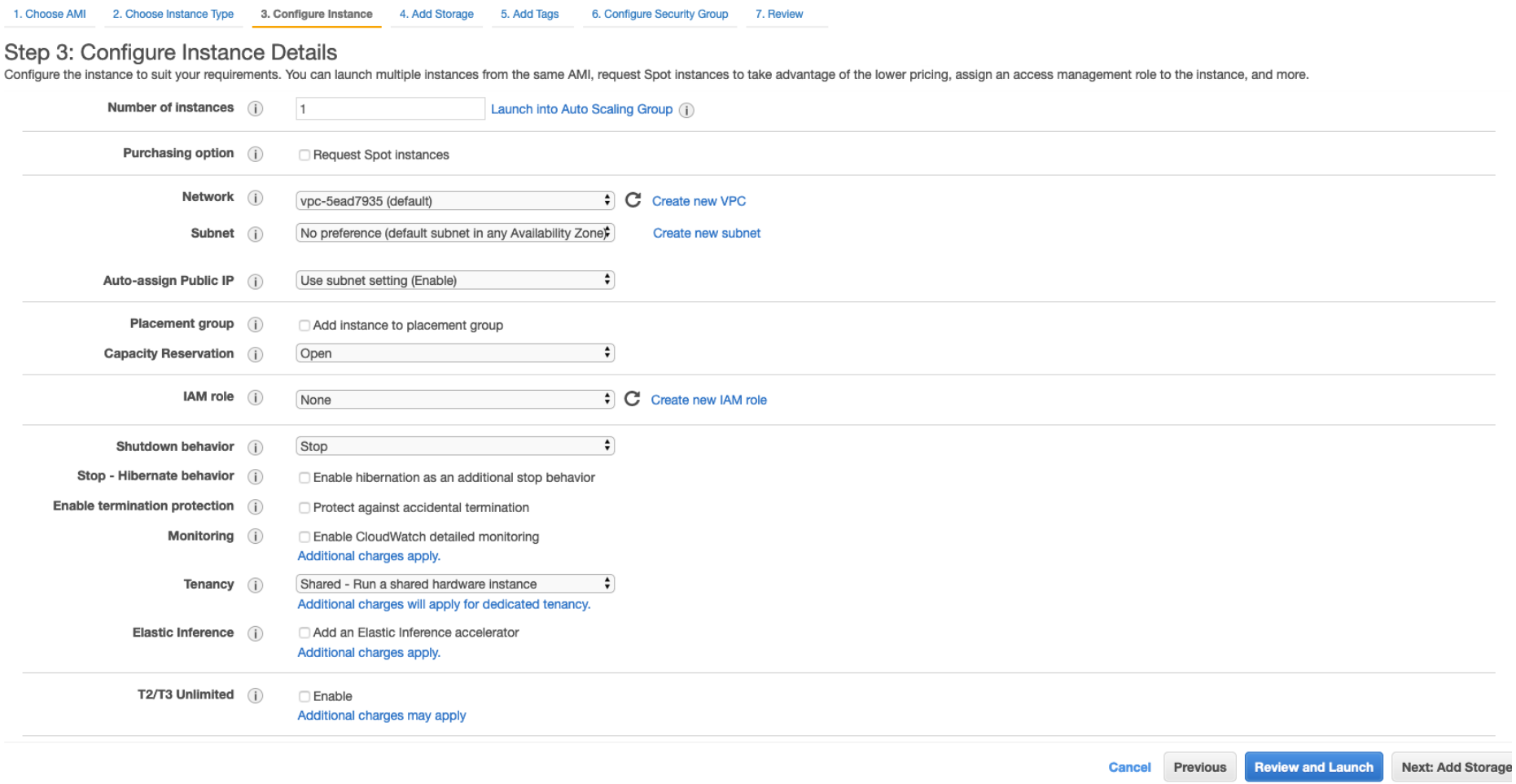




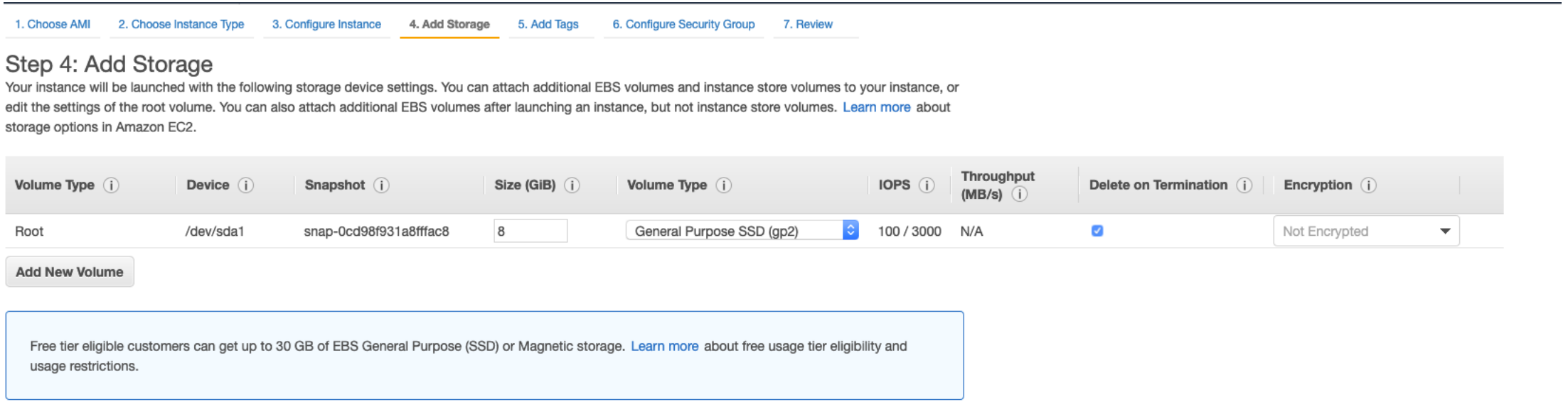
- Choose an instance based on use case
- Check the instance prices at https://aws.amazon.com/ec2/pricing/on-demand/
- Inform yourself about the properties of different instances in the 'Instance Types' menu
- Keep the economics in mind



Launching an instance — Step 3

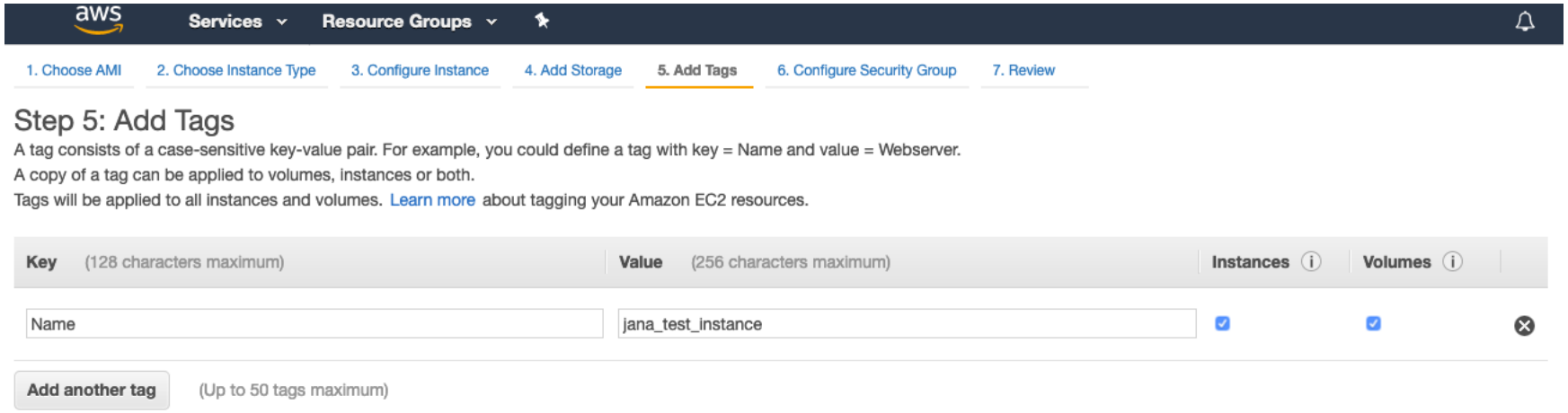


Launching an instance — Step 4



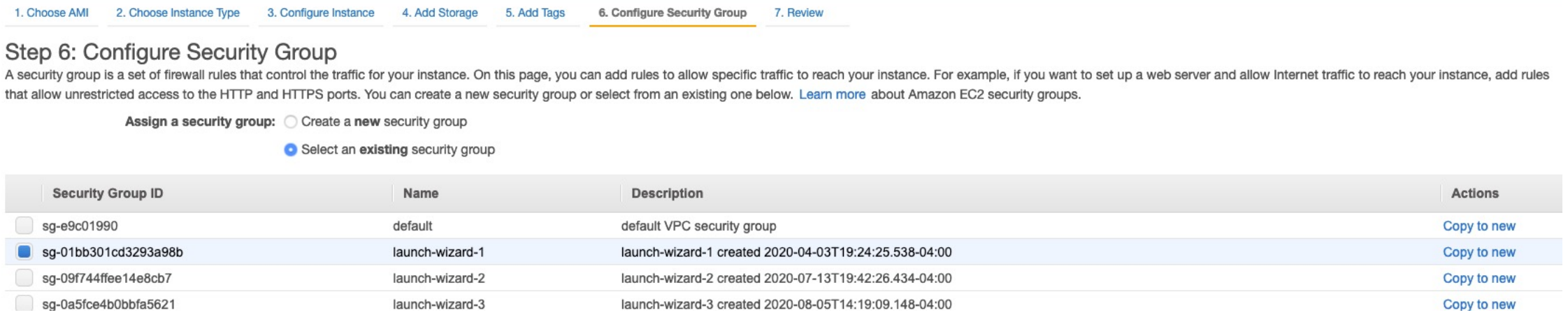
- Choose the size of the root volume
- The size can also be changed later in the ‘Volumes’ menu (only to increase it and it takes some time)
- You can add additional volumes to your instance (need to be attached and then mounted)
- This type of storage is expensive
- For long-term storage use S3 and download the data to your instance whenever needed

Launching an instance — Step 5



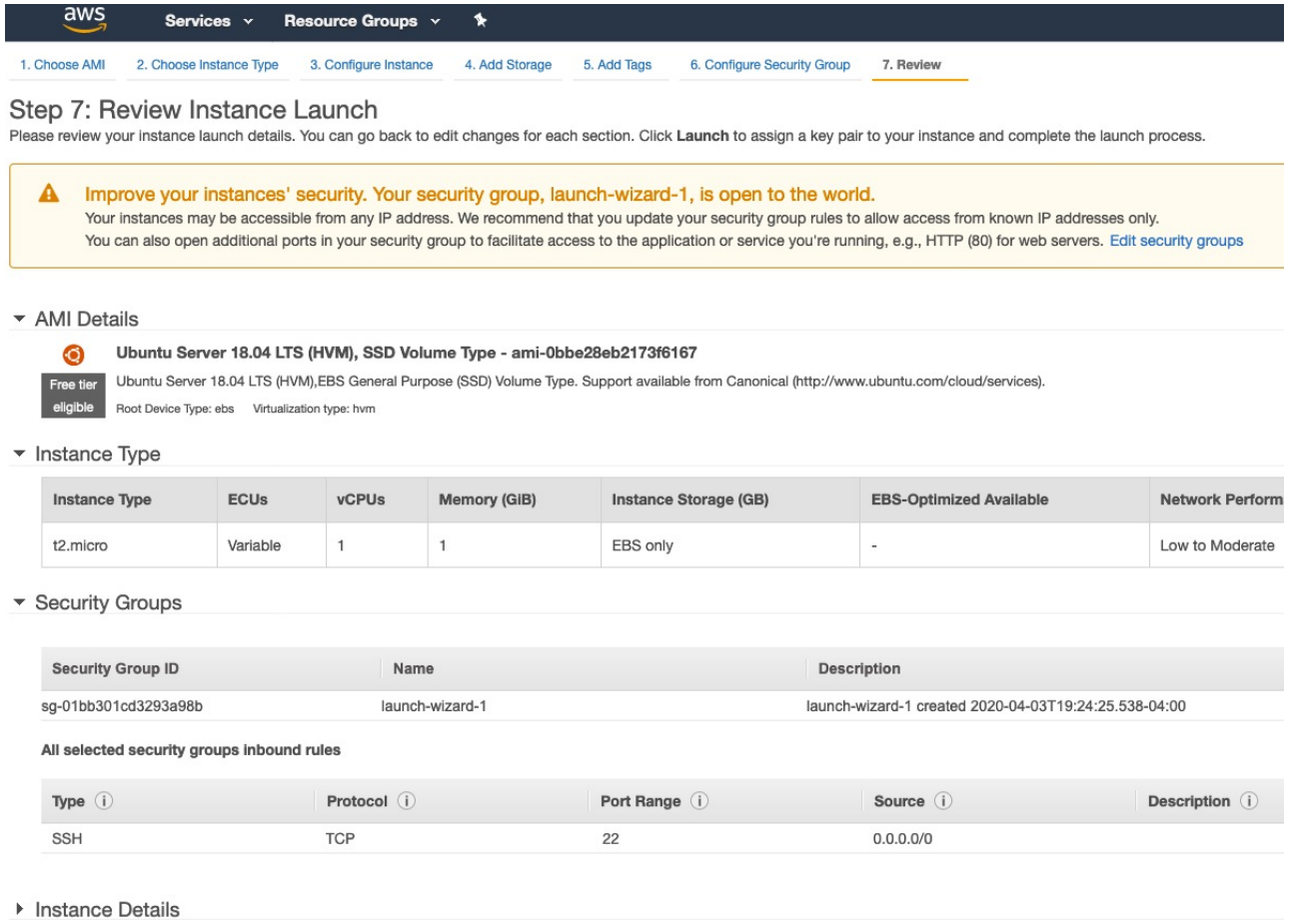
- Name your instances and volumes starting with your name
- E.g. ‘yiping_instance_large’

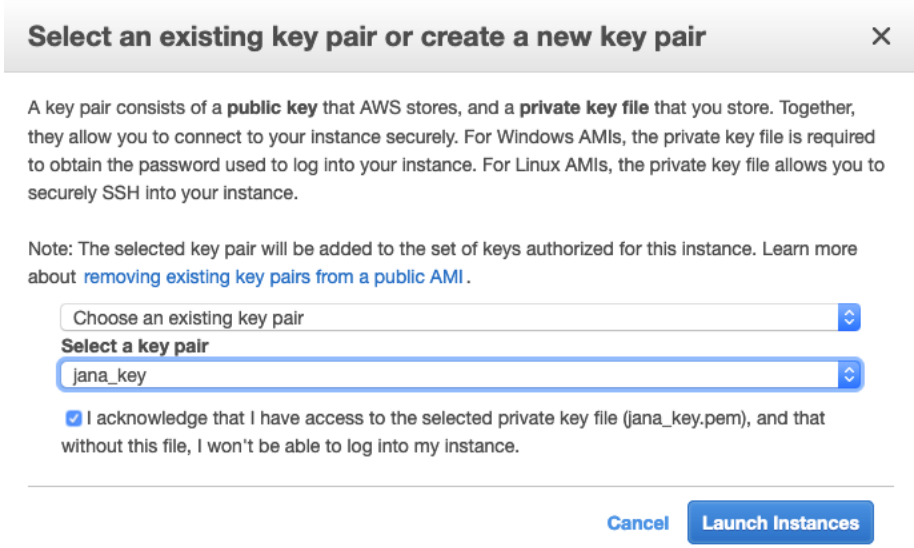
Launching an instance — Step 6



- Create a new security group if you don't have one
- Allow traffic only from your IP address
- Don't create a new security group every time you launch a new instance
- Don't allow all traffic

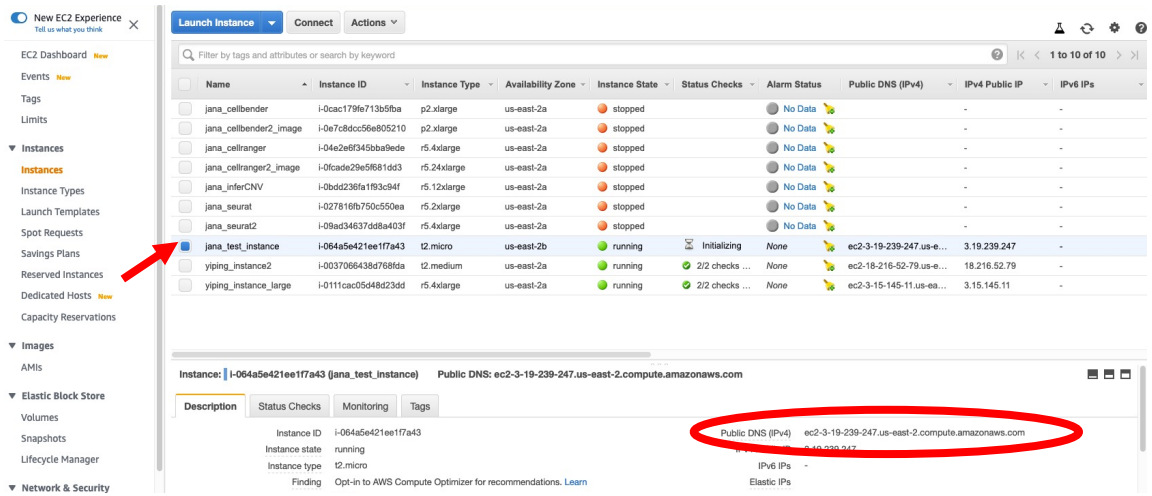
Launching an instance — Step 7





- Review your settings and launch your instance using your AWS key

SSH into your instance




```
jana@MDHO-N012:~ $sshfs -o IdentityFile=[redacted].pem ubuntu@ec2-3-19-239-247.us-east-2.compute.amazonaws.com:/home/ubuntu/ ~/AWS
The authenticity of host 'ec2-3-19-239-247.us-east-2.compute.amazonaws.com (3.19.239.247)' can't be established.
ECDSA key fingerprint is [redacted]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
jana@MDHO-N012:~ $ssh -Y -i [redacted].pem ubuntu@ec2-3-19-239-247.us-east-2.compute.amazonaws.com
Warning: No xauth data; using fake authentication data for X11 forwarding.
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 5.3.0-1032-aws x86_64)

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

  System information as of Wed Aug 12 17:50:57 UTC 2020

  System load:  0.0               Processes:           93
  Usage of /:   14.6% of 7.69GB   Users logged in:     0
  Memory usage: 18%               IP address for eth0: 172.31.31.85
  Swap usage:   0%


0 packages can be updated.
0 updates are security updates.


/usr/bin/xauth:  file /home/ubuntu/.Xauthority does not exist
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-31-85:~$
```

- Select your instance and copy the IP address
- You can mount the instance volume to your MacBook using FUSE for macOS (https://github.com/osxfuse/osxfuse)
- SSH into your instance by specifying the directory of your AWS key and the IP address of the instance
- ssh -Y –i xxx.pem ubuntu@ec2-3-19-239-247.us-east-2.compute.amazonaws.com

Update instance and install AWS CLI

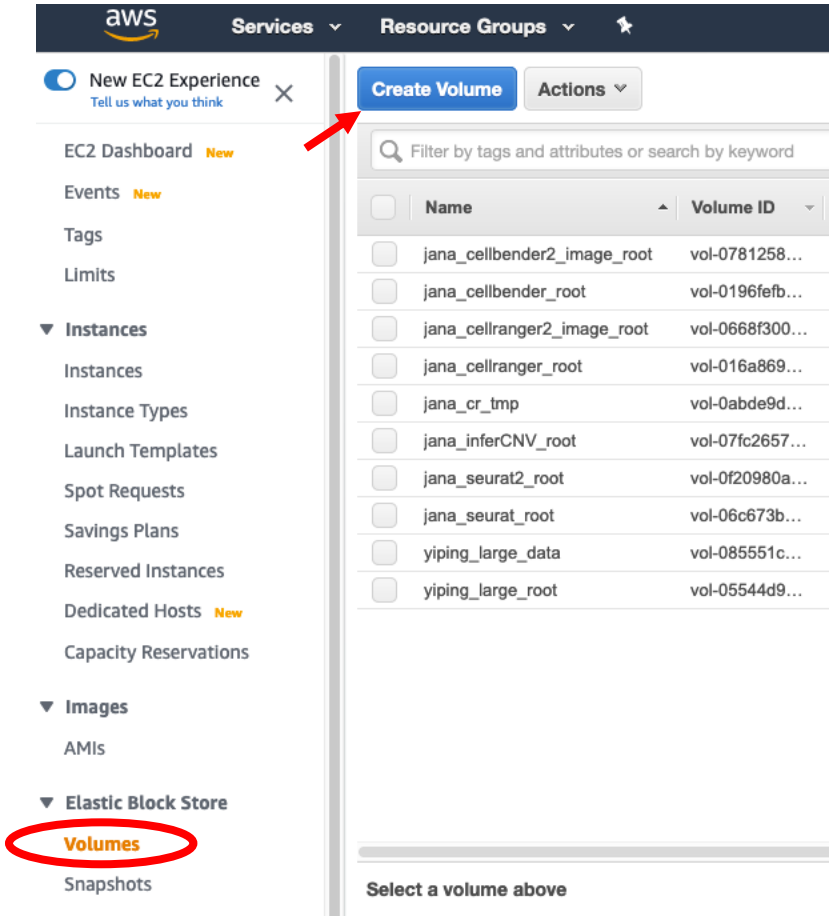
- Update instance with:

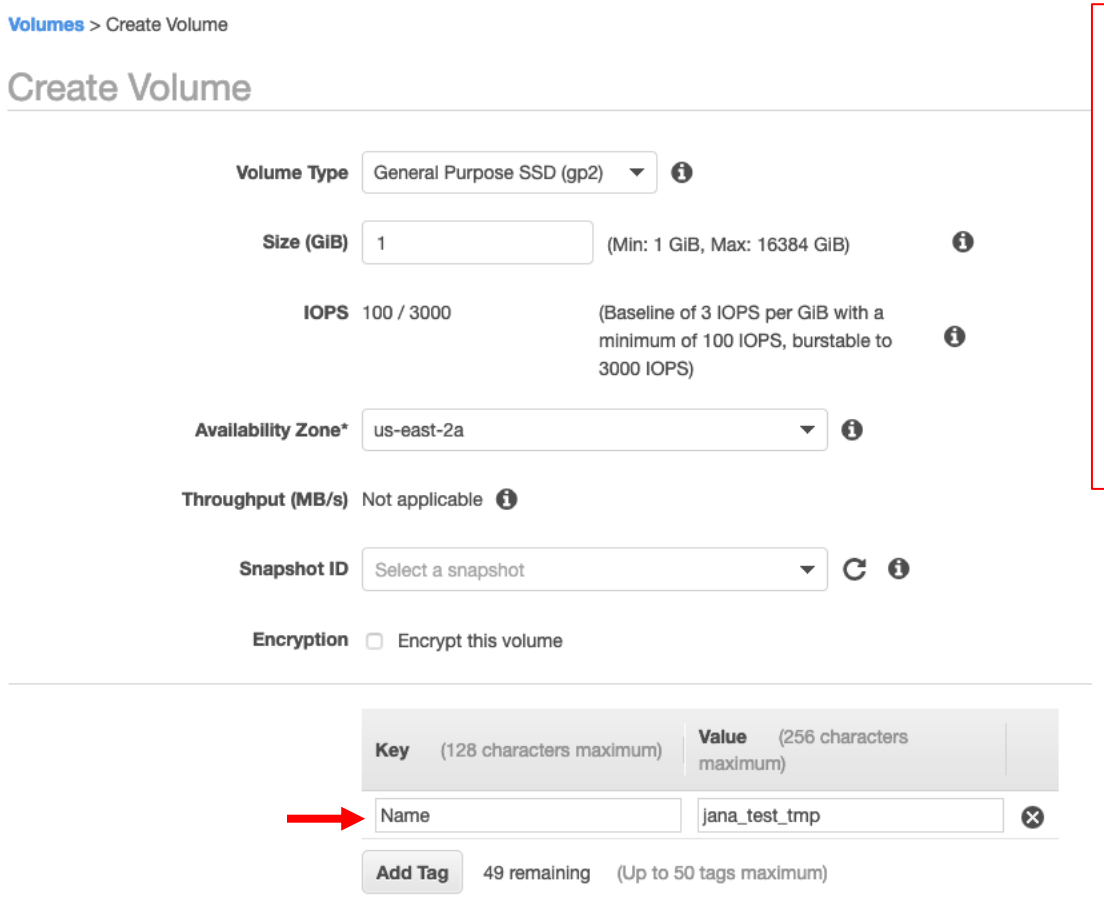
```
sudo apt-get update; sudo apt-get upgrade -y
```

- Connect to your AWS account:

```
# install aws command line
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
sudo apt install unzip
unzip awscliv2.zip
sudo ./aws/install


# set up aws  s3 access
mkdir ~/.aws
echo "[default]" >> .aws/credentials
echo "aws_access_key_id = your_key_id"  >> .aws/credentials
echo "aws_secret_access_key = your_access_key" >> .aws/credentials


# list all buckets (test if AWSCLI works)
aws s3 ls


# transfer between instance and S3
aws s3 sync data/ s3://bucket/data/ --exclude "*" --include ”only_files_with_these_terms" --quiet


# directories have to end with /
# use --dryrun first to be safe
```

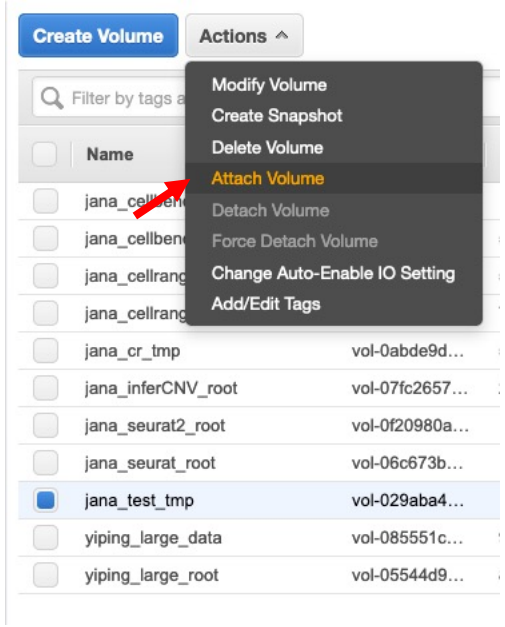
Create a volume





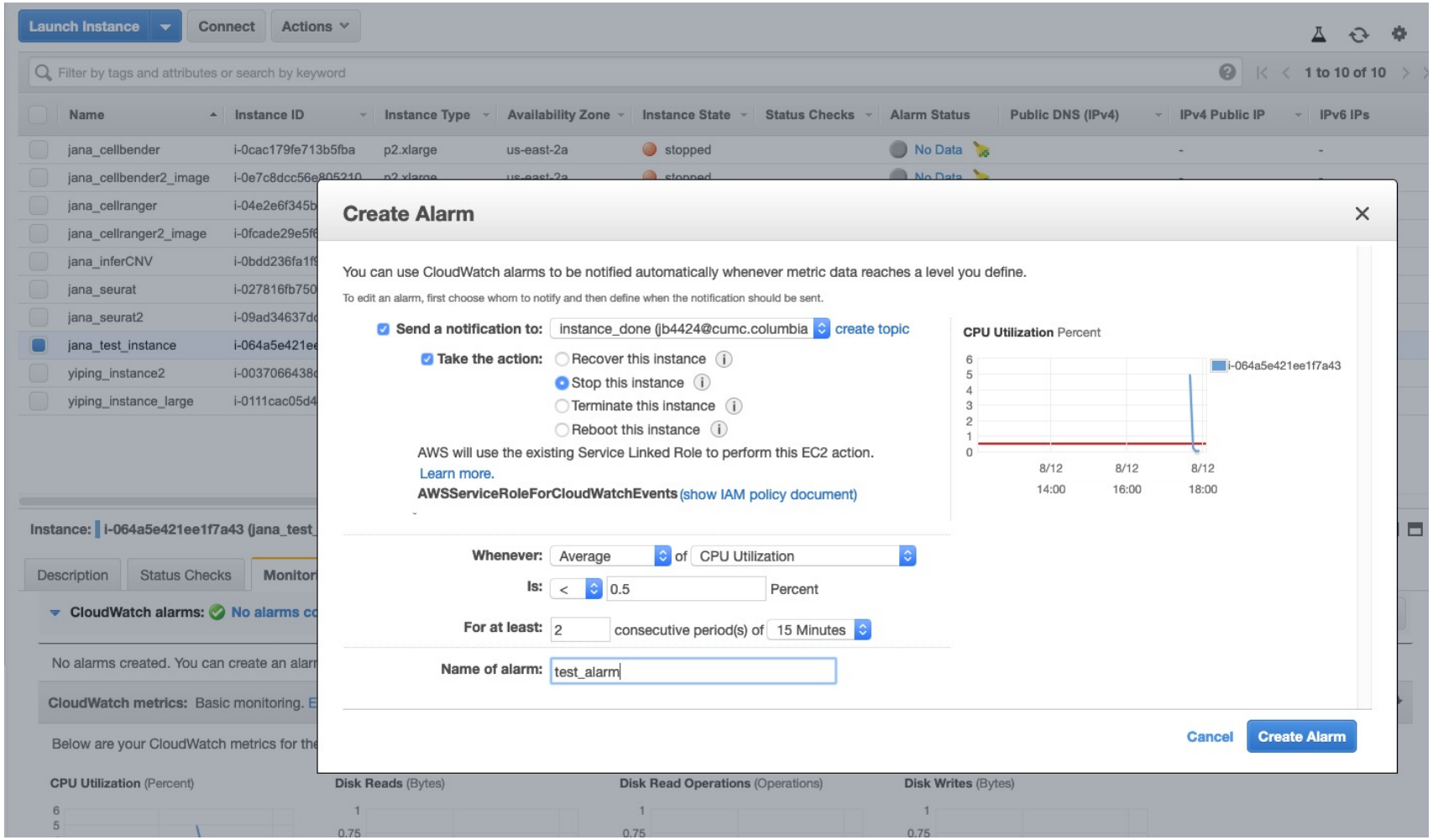
- Name your instances and volumes starting with your name
- E.g. ‘yiping_instance_root’

Attach and mount the volume to your instance




```
# Find the name of the volume
lsblk
mkdir data

# check if volume has any data (shouldn't be the case when newly created)
sudo file -s /dev/xvdf

# Format the volume to ext4 filesystem
sudo mkfs -t ext4 /dev/xvdf

# Mount volume
sudo mount /dev/xvdf1 data

# Change owner
sudo chown ubuntu:ubuntu vol
```

Set an alarm for your instance

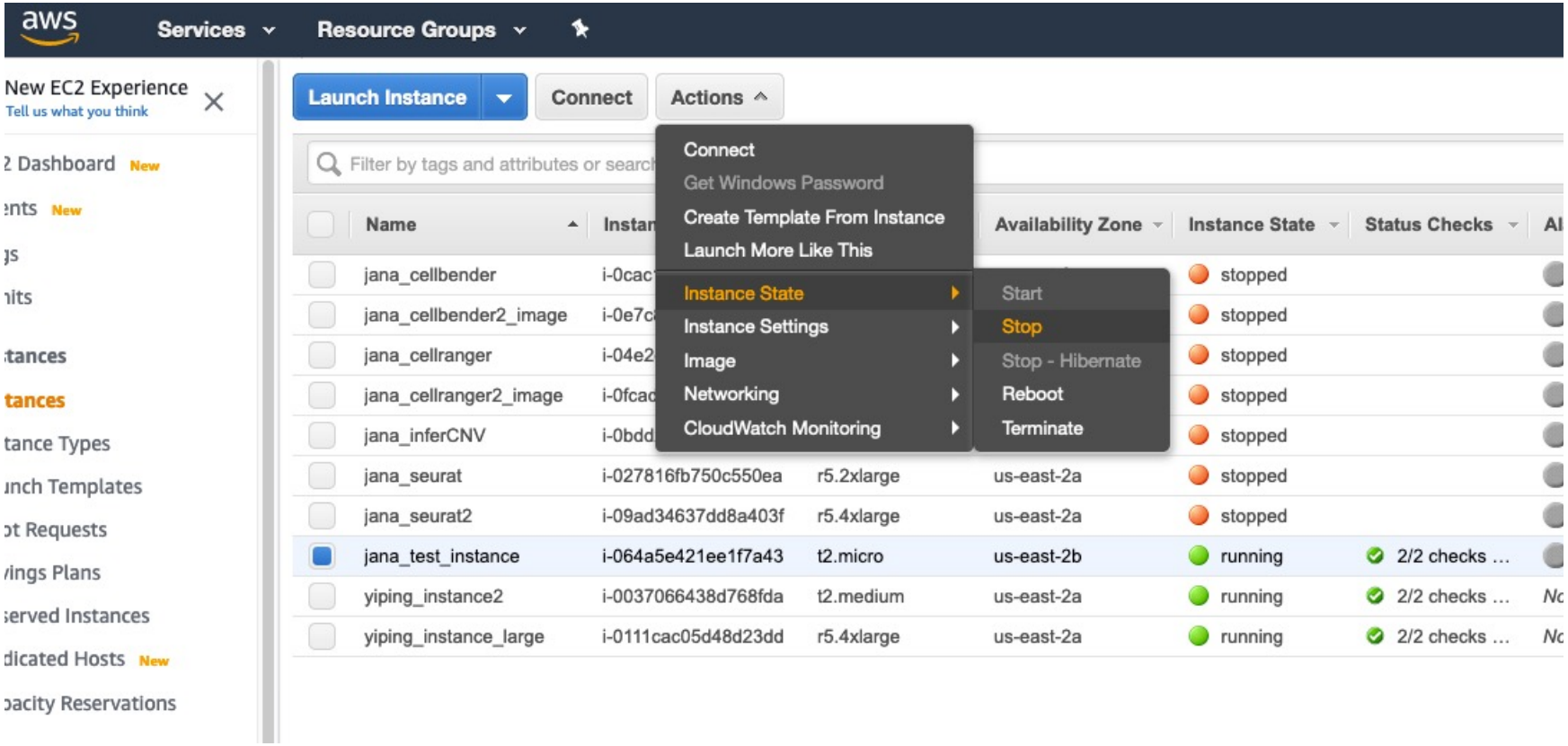


- Always set alarms for your instances to shut down automatically when they are not used anymore
- Forgetting to shut down an instance can become expensive

- Select your instance
- Select the ‘Monitoring’ menu
- Click ‘Create Alarm’ on the top right

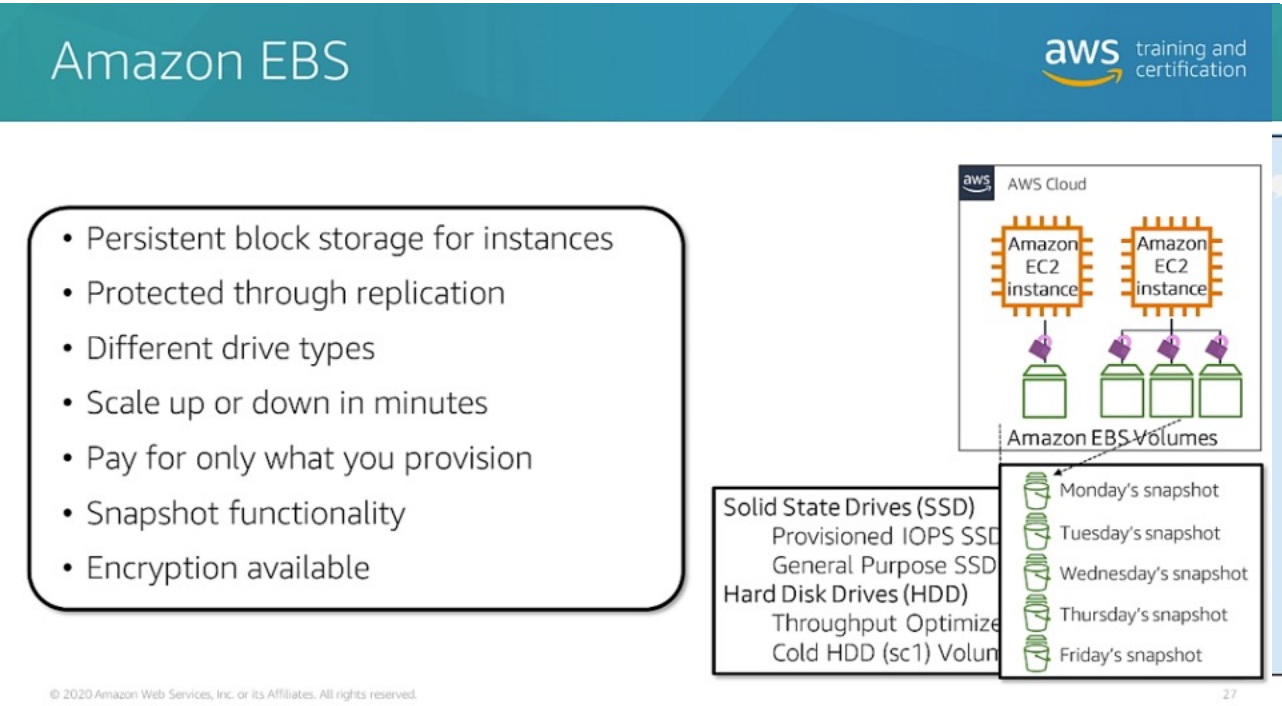
- You can also create alarms for your volumes

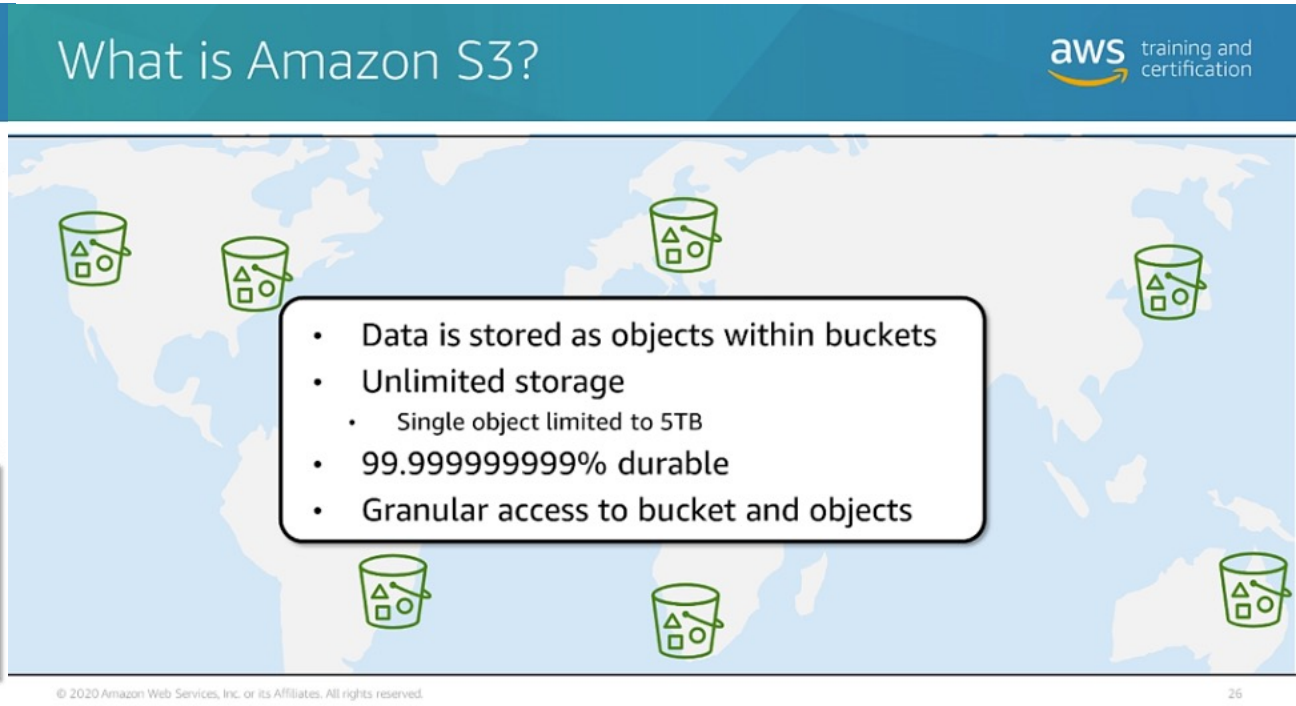
Shut down your instance



- Make sure only your instance is selected
- ‘Stop’ will shut down the instance until you start it again
- ‘Terminate’ will delete the instance (and the associated root volume unless you changed the settings)
- When instances are stopped, we only pay for the storage of their volume

Elastic Block Store (EBS) vs Simple Storage Service (S3)





- ❖ EBS is for short-term storage while you are working with the data
- ❖ S3 is for long-term storage to download to EBS whenever you need to work with the data

Additional points

- You can find your password and key in the credentials file
- Keep the credentials file save in a hidden folder
- S3 is in most cases the only copy of our data so make sure not to delete or alter anything
- Most people in the lab have access to S3 so you can direct them to your analysis output
- Downloading data from AWS is very expensive so please don't download large files but instead analyze them on AWS instances until you reach a smaller format that can be downloaded
- The volume storage associated with instances is also very expensive so try to keep it small and store data on S3
- Please only use your own instances, volumes and images generated with your own key
- If you don't need an instance anymore you can terminate it
- You can check the budget in the 'Cost Explorer' menu and filter for costs generated with your key. This will only include instance-related costs and not the costs generated through S3 and volumes.