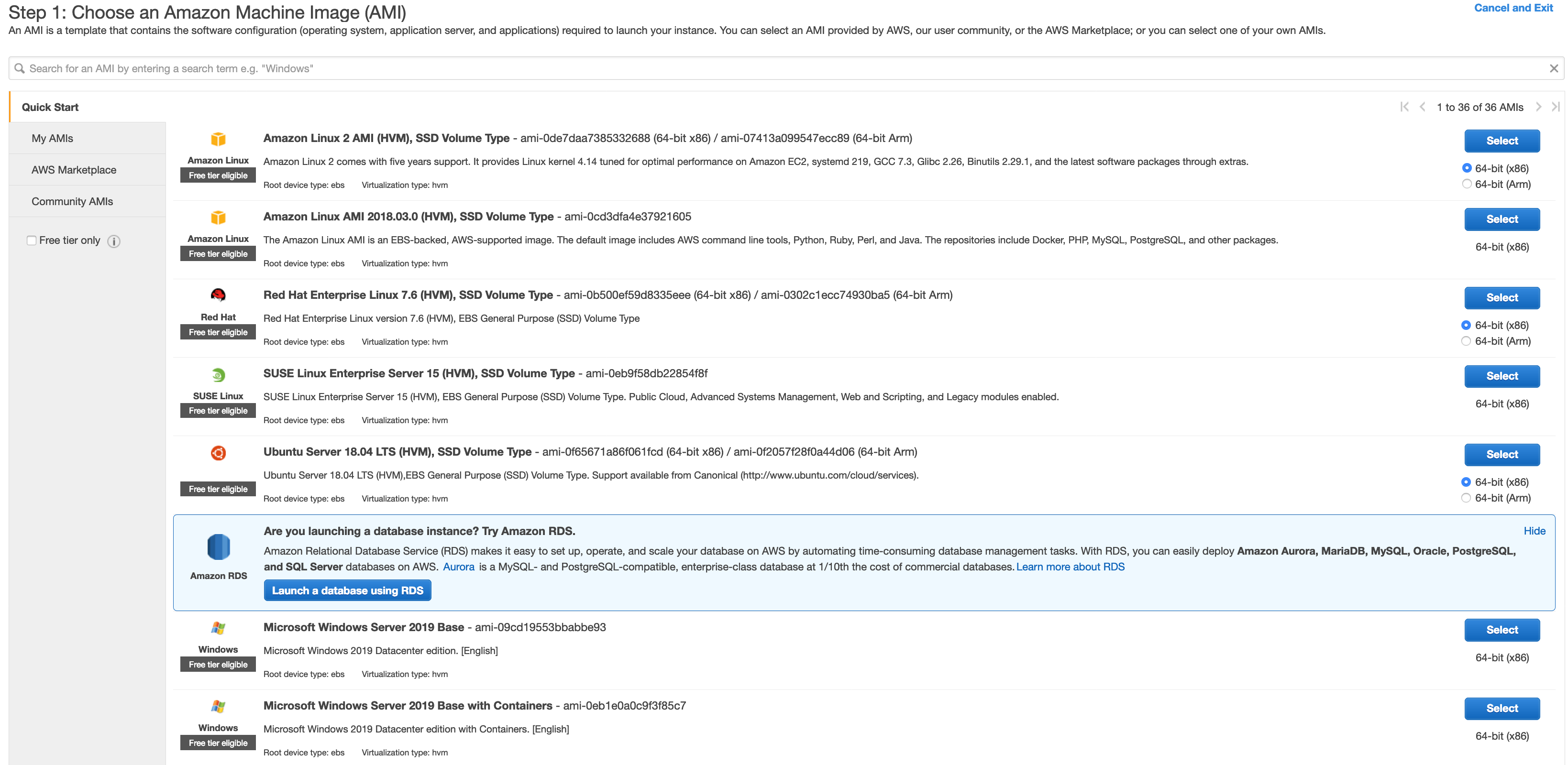
EC2:

**Launch Instance:**

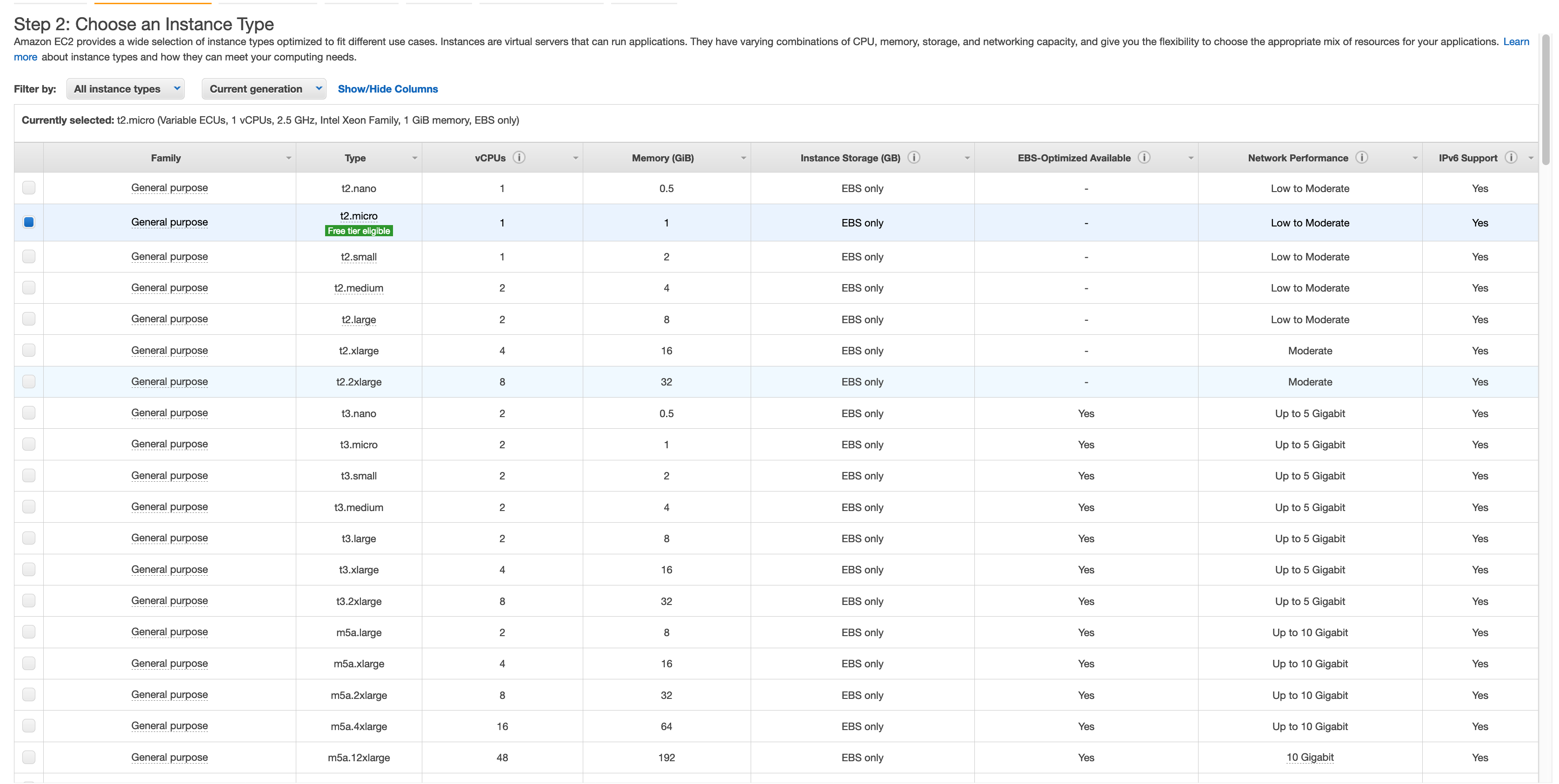
AMI (Amazon Machine Image) – snapshots of virtual machines that can be boot up.

The virtualization can be of 2 types – HVM and PV (para virtual)

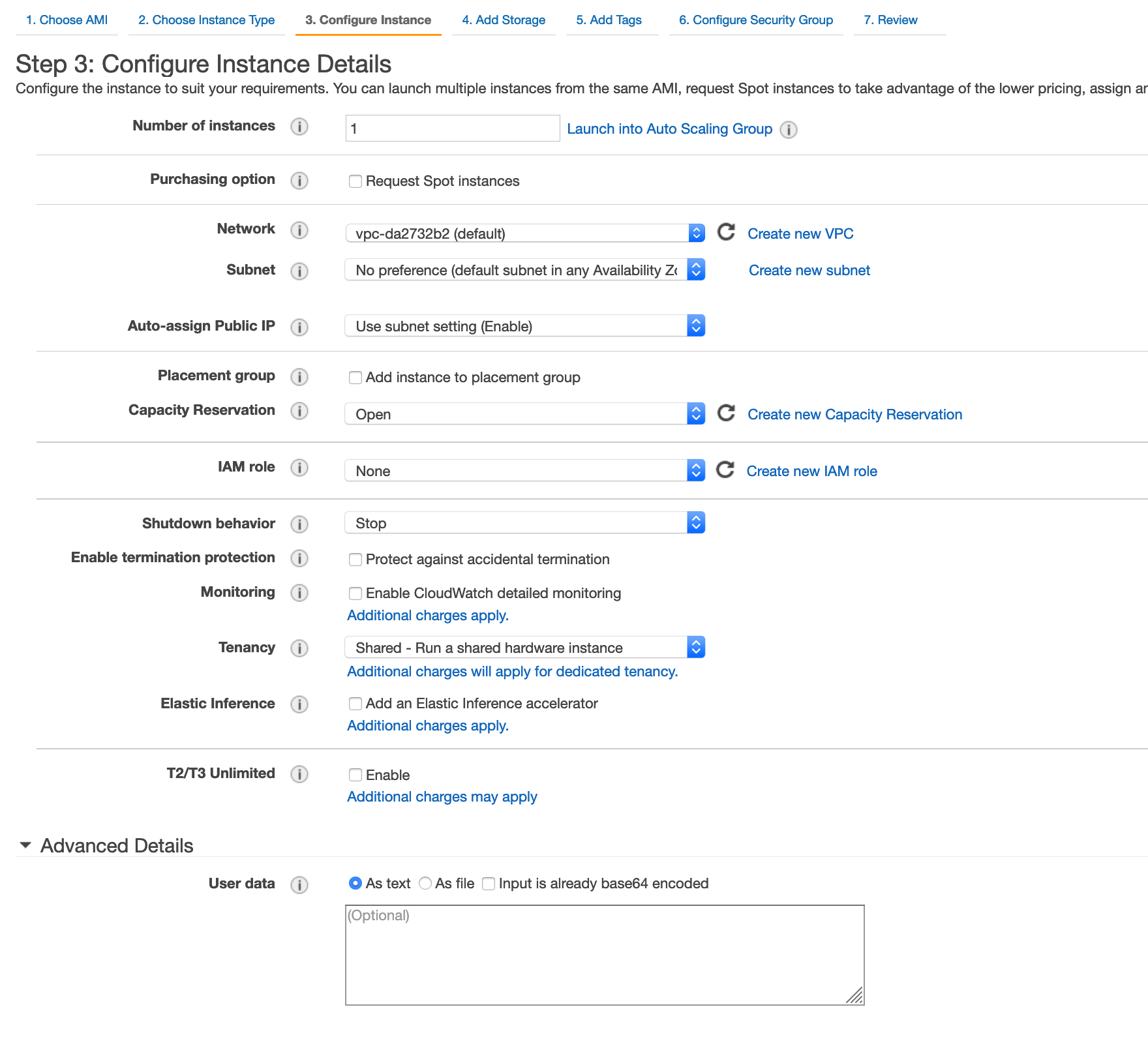
Different AMIs:



Different instances:

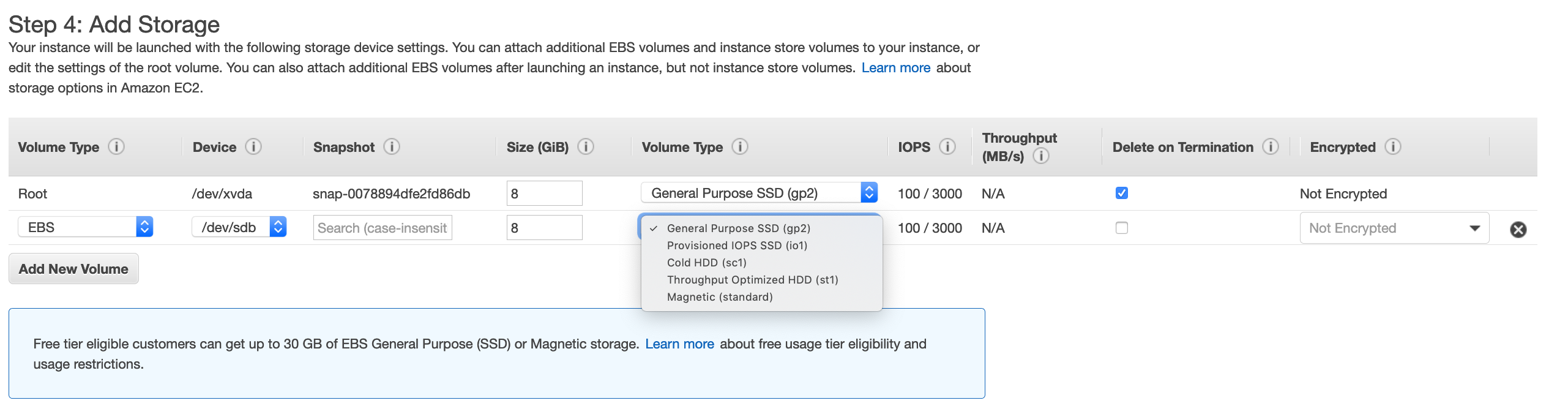


Configure Instance:



* To request spot instances – Purchasing option
* To use dedicated instances – Tenancy
* Subnet – 1 subnet cannot span multiple availability zones, it can only be in one availability zone
* Shutdown behavior – what to do if os shuts down - stop/kill instance
* Termination protection – to prevent accidental termination (turned off by default)

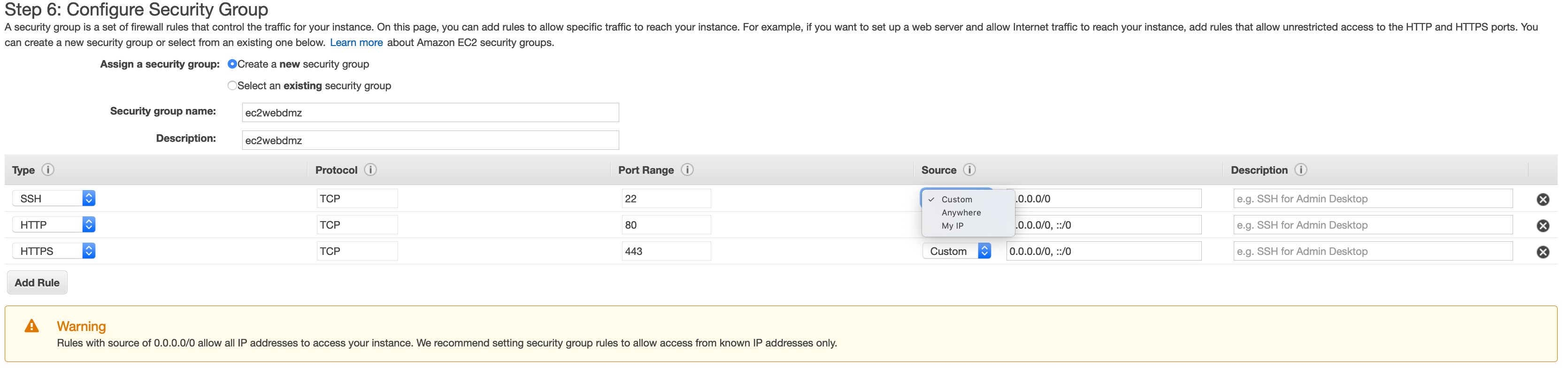
Add Storage:



* Delete on termination – virtual disk will be deleted on termination (default action for EBS volumes)
* SC1 and st1 only available for additional volume
* GP2, io1, magnetic can be root bootable volume

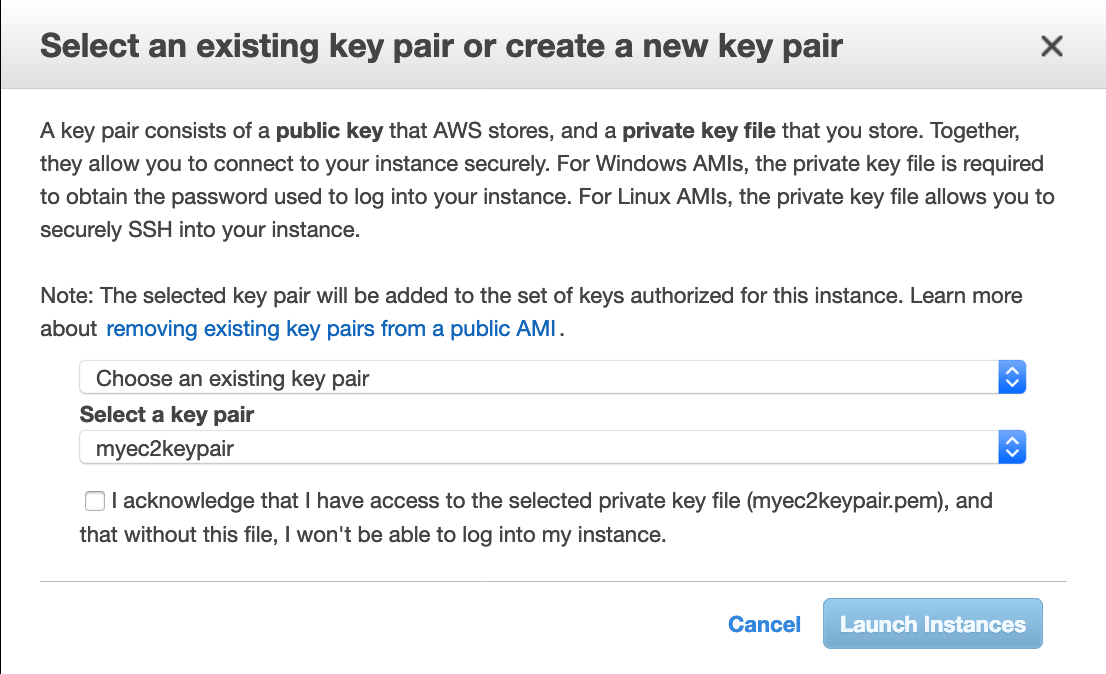
Tags can be added to track ec2 instance

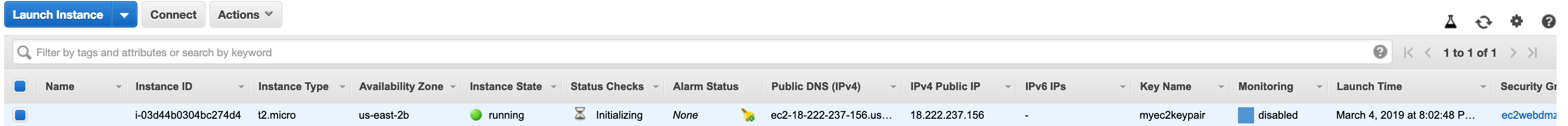
Security group:



* This is a virtual firewall. Here we allow access.
* My IP allows access from a fixed ip.

While launching instance, we can create a key pair which can be used to SSH to instance which is a .pem file.





SSH into an EC2:

**1)Lock down key file.**

CHMOD 400 myec2keypair.pem

**2)SSH into instance using public ip and keypair.**

ssh ec2-user@18.222.237.156 -i myec2keypair.pem

\_\_| \_\_|\_ )

\_| ( / Amazon Linux 2 AMI

\_\_\_|\\_\_\_|\_\_\_|

https://aws.amazon.com/amazon-linux-2/

1 package(s) needed for security, out of 3 available

Run "sudo yum update" to apply all updates.

**3)elevate and update**

sudo su

[root@ip-172-31-26-212 ec2-user]# yum update -y

**4)Install apache**

yum install httpd

**5)Put a html page**

cd /var/www/html

nano index.html

Put html contents in file and hit ctrl+x, then enter

**6)Start apache service**

service httpd start

**7)Go to web browser and type** <http://18.222.237.156>

**8)Start apache HTTP service every time instance is rebooted**

chkconfig httpd on

Note: Forwarding request to 'systemctl enable httpd.service'.

Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.

If an instance has termination protection on, we have to disable termination protection before we can terminate the instance.

**Status Checks:**

There are 2 types of status checks:

* System status checks (it is reachable) – in case of failure, terminate instance and relaunch, maybe an issue with infrastructure.
* Instance status checks (traffic can go to OS) – in case of failure, reboot your instance

**Terminate Instance:**

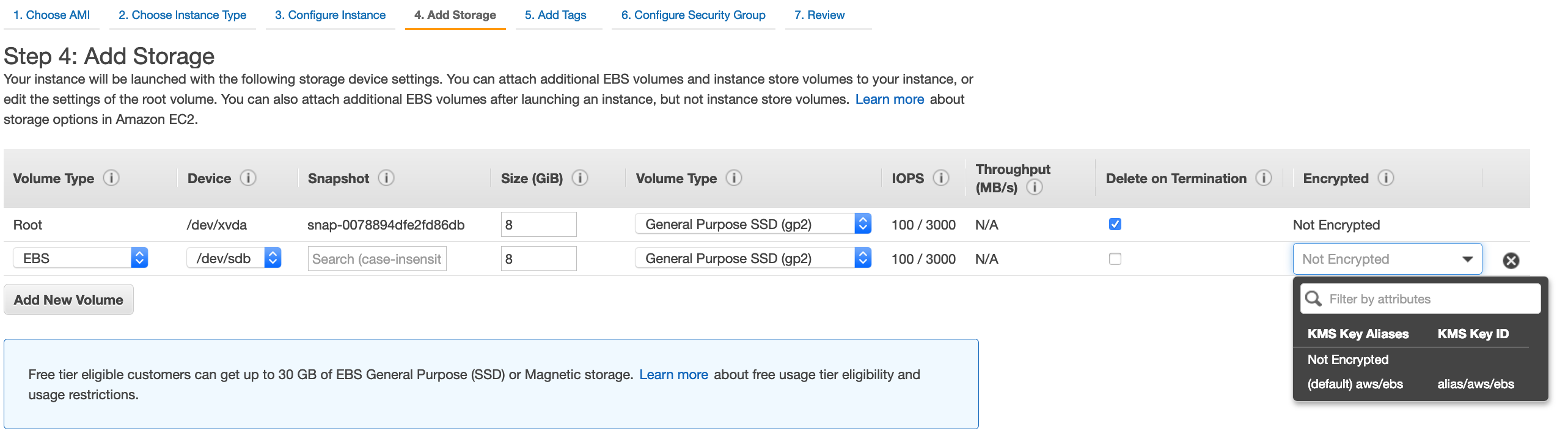
Actions>Instance-state>Terminate

**Reserved Instance:**

Instances>Reserved Instances: All upfront saves you the most.

We cannot encrypt the root device volume for the default AMIs provided by amazon.

We can create a copy of the AMI and encrypt the root device volume of that AMI.

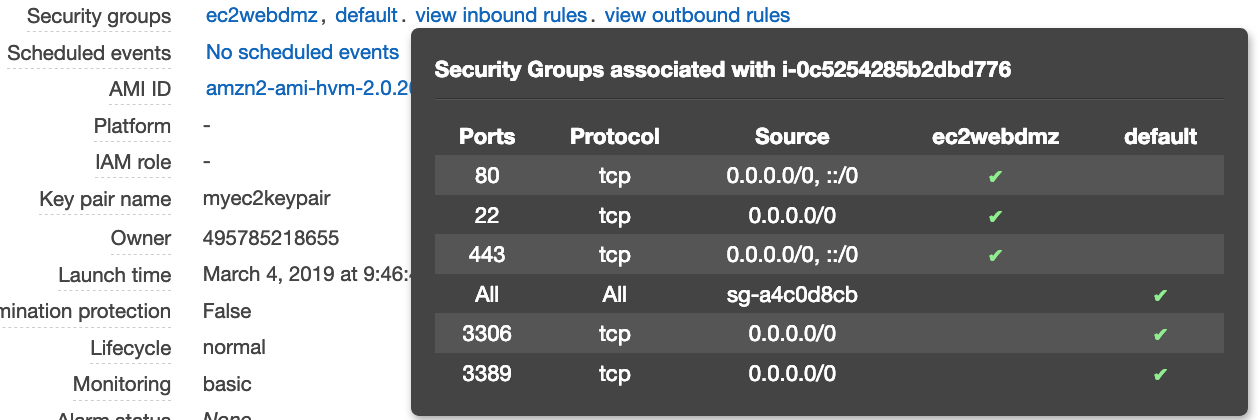


**Security Groups:**

They are virtual firewalls that control traffic to EC2 instance. One EC2 instance can be behind multiple security groups. If a security group associated to an EC2 instance defines SSH rules that allows login only from a particular ip, then no one can ssh to ec2 instance apart from the owner of that ip.

Security groups have both outbound and inbound rules. Inbound rules control all requests to EC2 server. Outbound rules control any response from the EC2 server.

* Any changes to rules in security groups are immediately effective.
* All inbound traffic is blocked and outbound traffic is allowed by default. We have to create rules to allow them in. We cannot explicitly deny any traffic since everything apart from what we have allowed is blocked by default. Hence only allow rules and no deny rules.
* All rules in security groups are stateful which means that any inbound rule will have the corresponding outbound rule enabled by default. No outbound rules need to be added specifically.
* One ec2 instance can be behind multiple security groups and one security group can have multiple ec2s.
* We cannot block ip addresses using security groups, we need to use Network access control lists.



**EBS volumes:**