[Setup AWS Account 1](#_Toc499642798)

[Buckets 5](#_Toc499642799)

[Static Websites 8](#_Toc499642800)

[Connecting to Macs and PCs 9](#_Toc499642801)

[Using CLI 9](#_Toc499642802)

[EC2 building servers 10](#_Toc499642803)

[Route 53 - DNS 18](#_Toc499642804)

[VMWare 19](#_Toc499642805)

[Penetration Testing 21](#_Toc499642806)

[DDoS 21](#_Toc499642807)

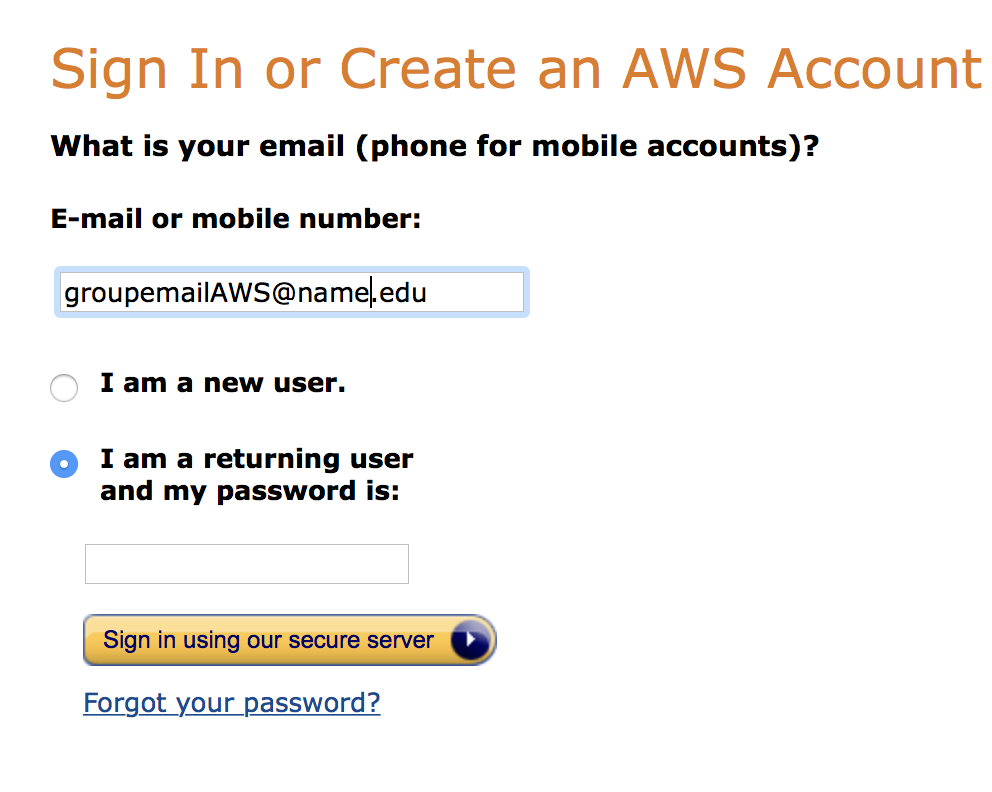
[Helpful Hints 21](#_Toc499642808)

***Amazon Web Services Hands-On Workshop DETAILED Outline***

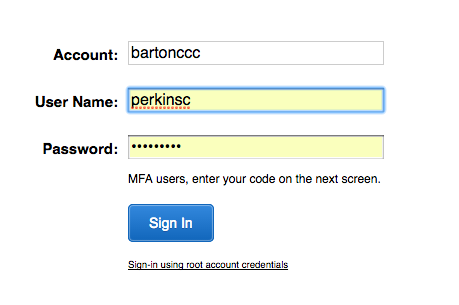
(KanREN provides connecting to Amazon Web Services through Internet 2)

# Setup AWS Account

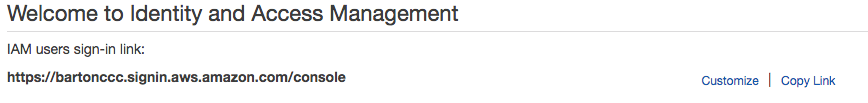
1. **Amazon Web Services Setup Accounts & Considerations**
   1. Account Root User - should be group email - groupemailAWS@name.edu
      1. Should be used to create user accounts
      2. Only account which can change:
         1. Support Plan
         2. Payment Options
         3. View Billing and allow others to
            1. You will want to activate cost allocation tags
            2. Billing is very customizable
         4. Can Open Support Case
         5. View account canonical user ID
         6. Logs in at:



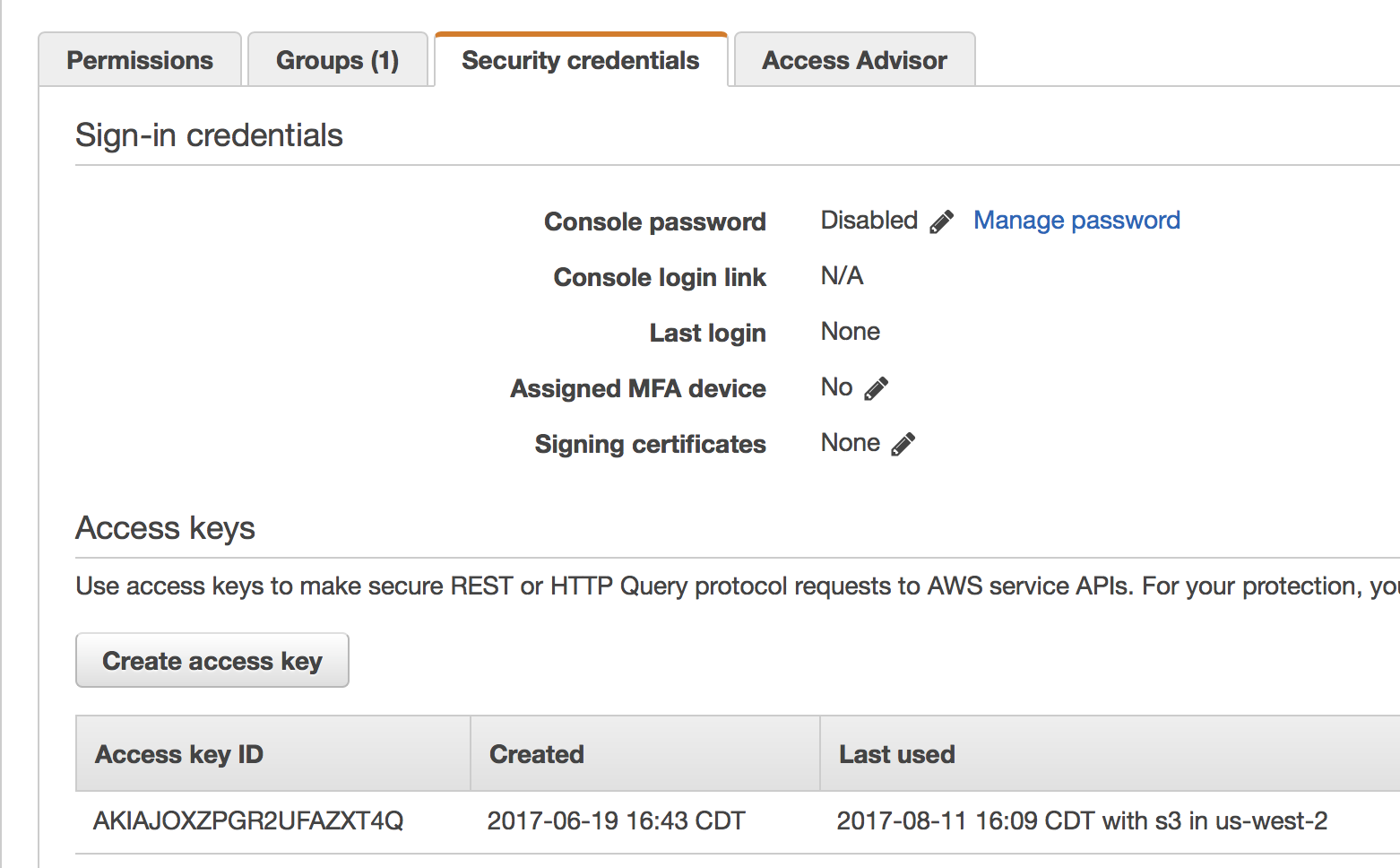
* 1. Create users
     1. Use IAM to create login, logs in at:



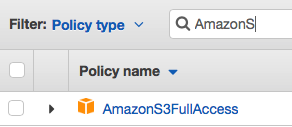
You can customize the Account: above from a number to a name in IAM under Customize.



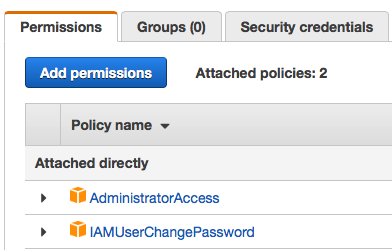
* + 1. These are for daily tasks
    2. Programmatic Access
       1. Users to save to



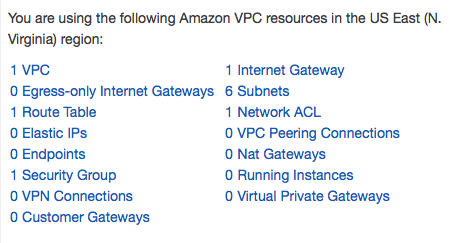
* + - 1. Backup User for internal servers to backup to S3
         1. IAM-Policies-Search for S3, click the box by AmazonS3FullAccess, Policy actions, Attach



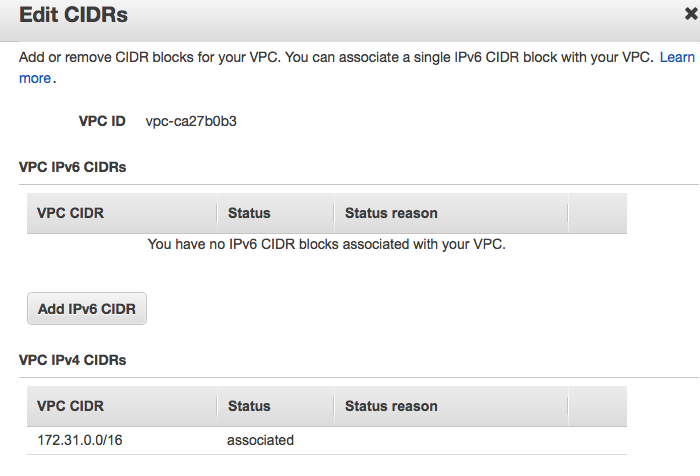
* + 1. AWS Management Console Access
       1. Admins (Can allow S3 users limited access to drag and drop files/folders)
       2. Can create other users and assign groups
       3. Setup S3 buckets and permissions
       4. To give an existing user Management Console Access
          1. IAM-Policies-Click the box by Administrator Access, Policy actions, Attach



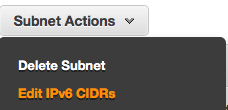
1. **AWS Virtual Private Cloud**
   1. Setup Considerations for Virtual Private Cloud first most sites will only need one.



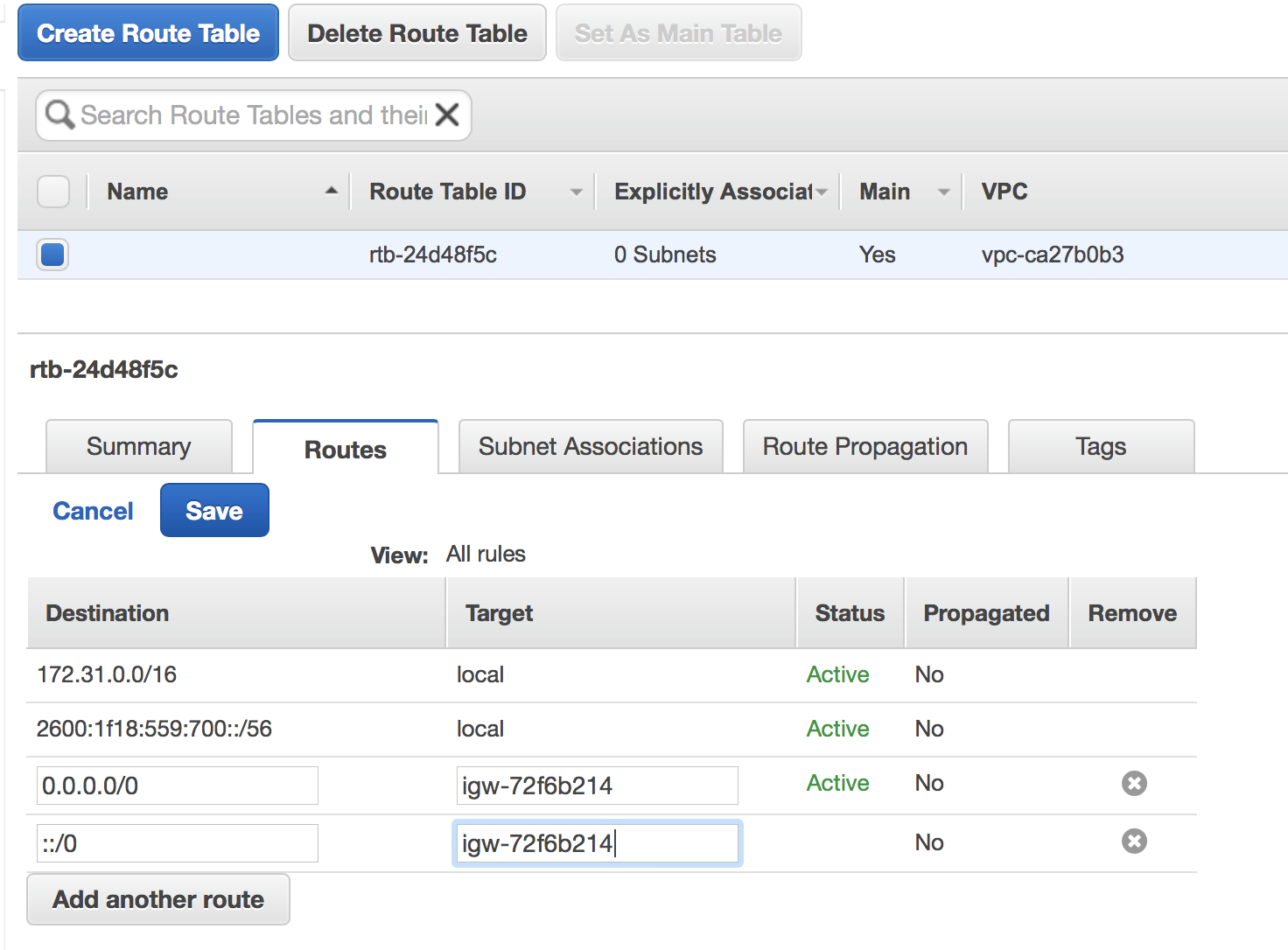
* 1. Assign subnets and IPV6
  2. First Add IPv6 CIDR to VPC



* 1. Then add IPv6 to Subnets



* 1. Remember to manually add ::/0 to your route table.

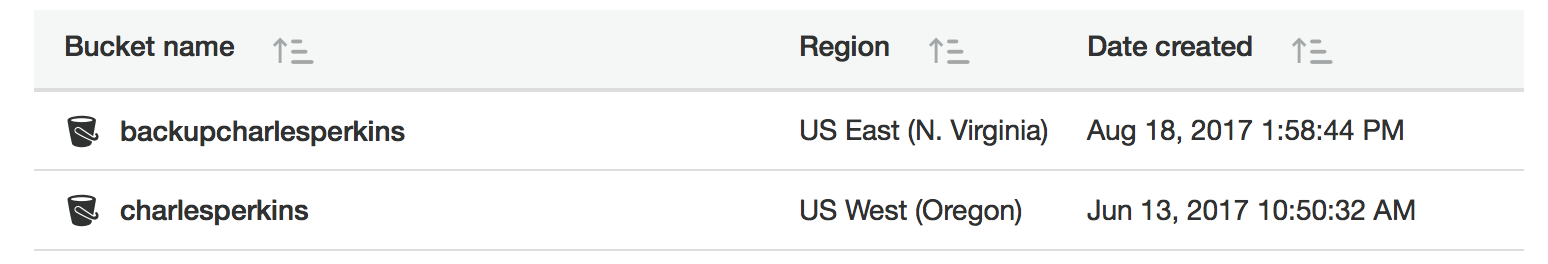


# Buckets

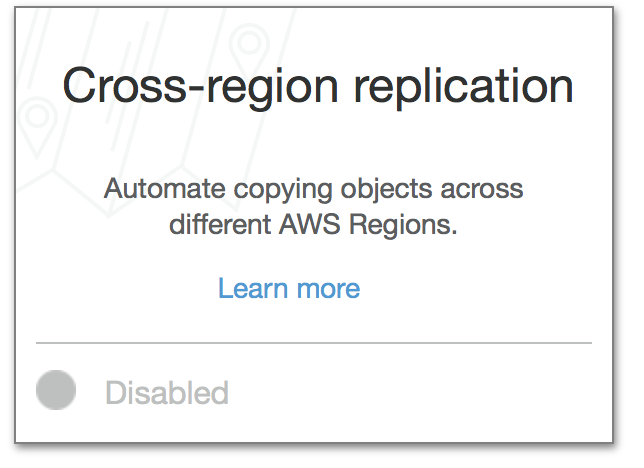
1. **AWS S3 Buckets**
   1. Start with 10TB Storage and 100 buckets

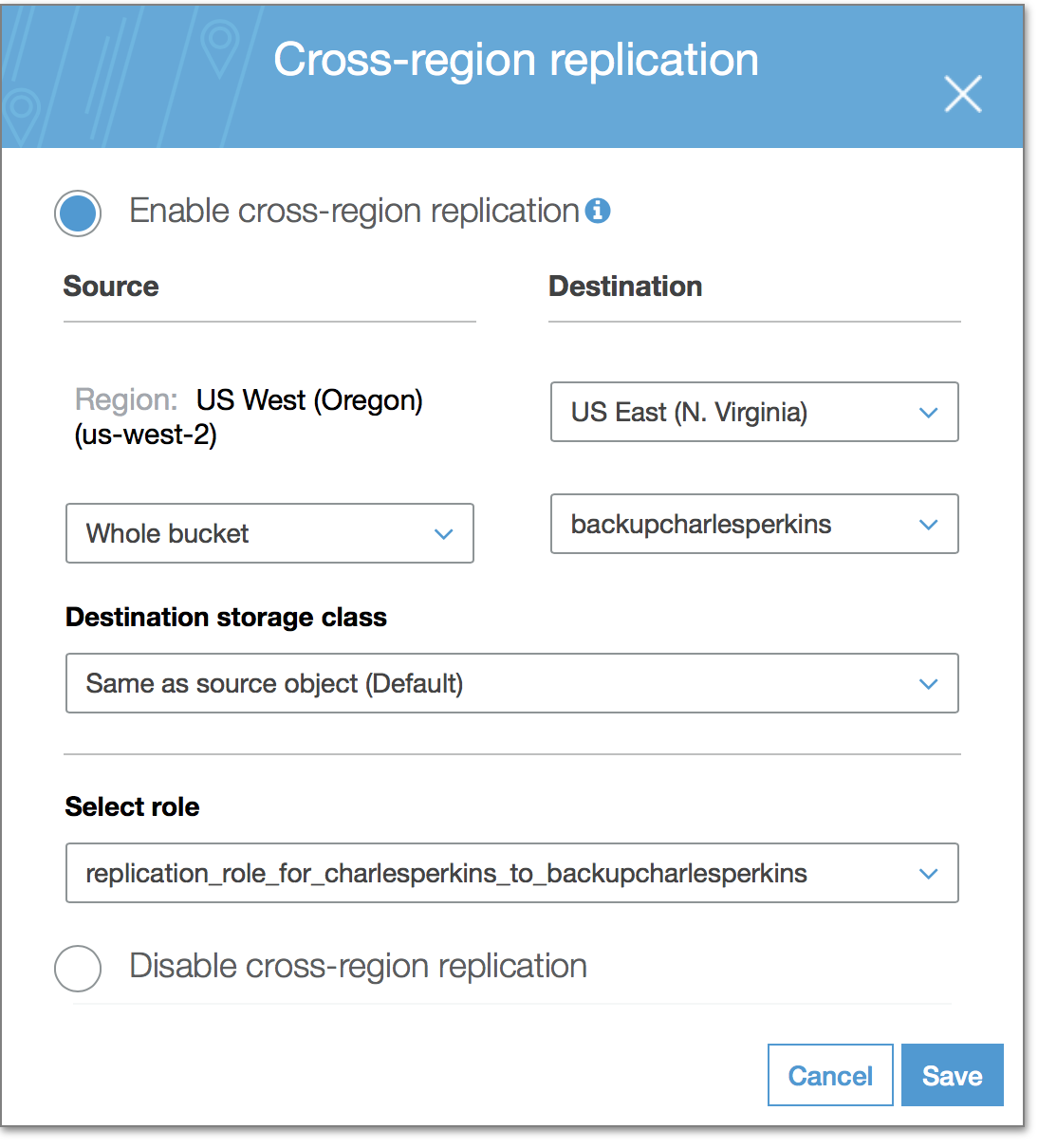


* 1. Cannot put a bucket in a bucket
  2. Buckets do not require region selection
  3. Note pay attention to what region you are creating the bucket in.

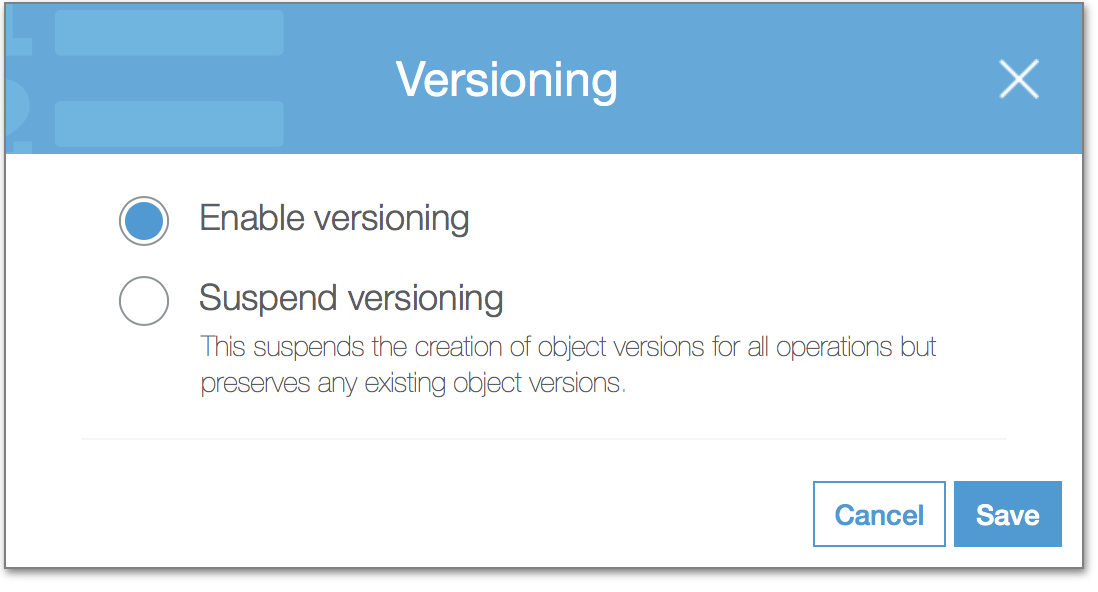


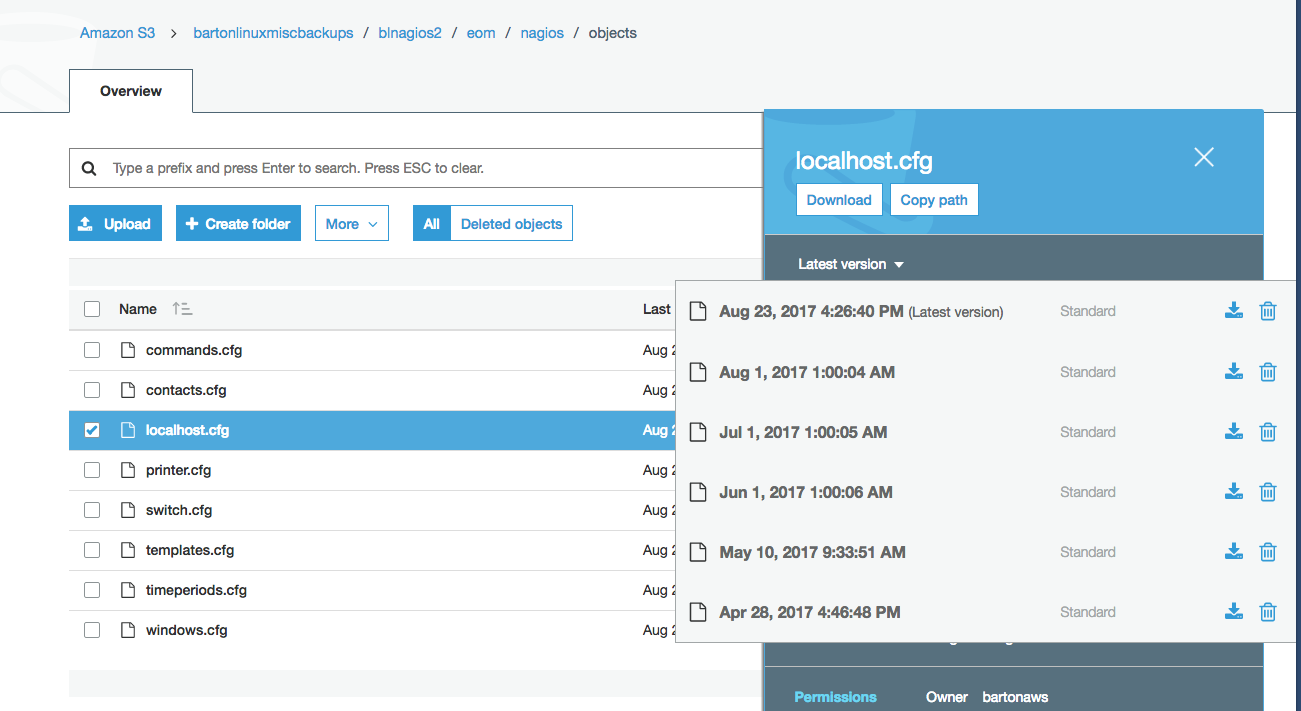
* 1. To insure your data is backed up to 2 different physical locations, create a backup bucket in another region and then under management - replication

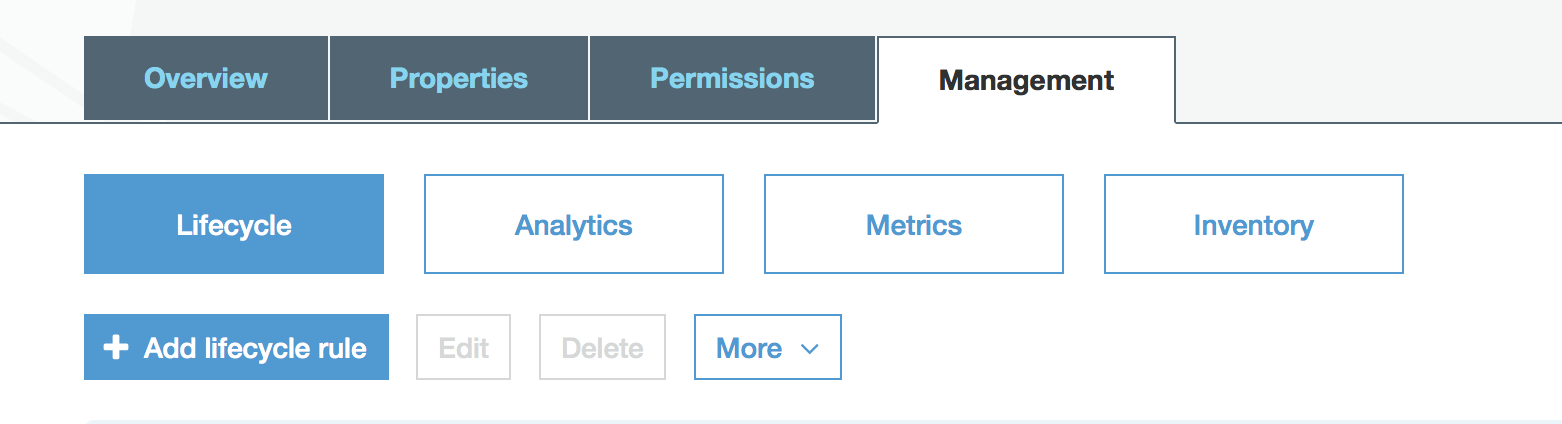




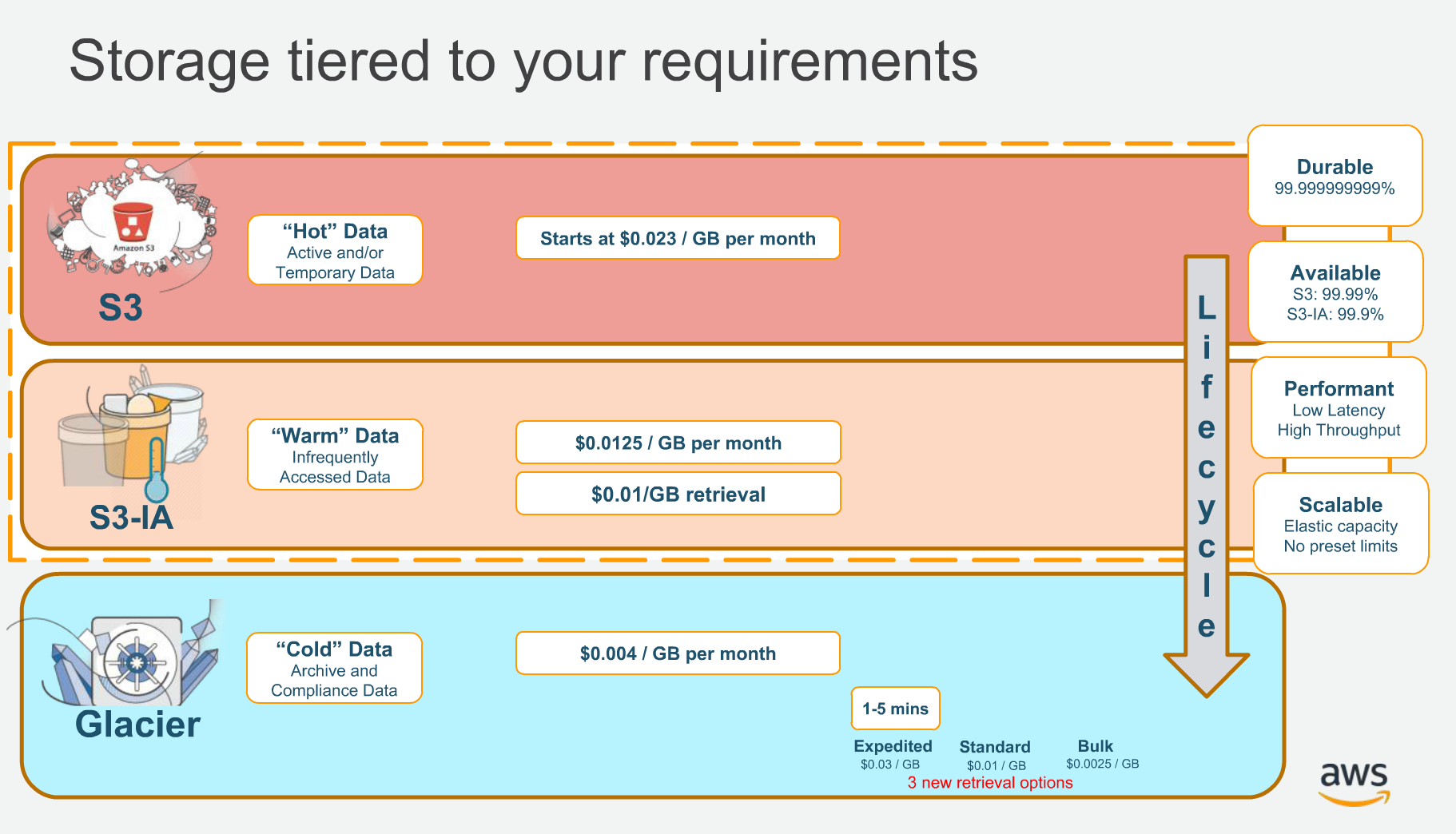
* 1. Turning on Versioning allows you to keep previous file versions for as long as your life cycle policy allows.





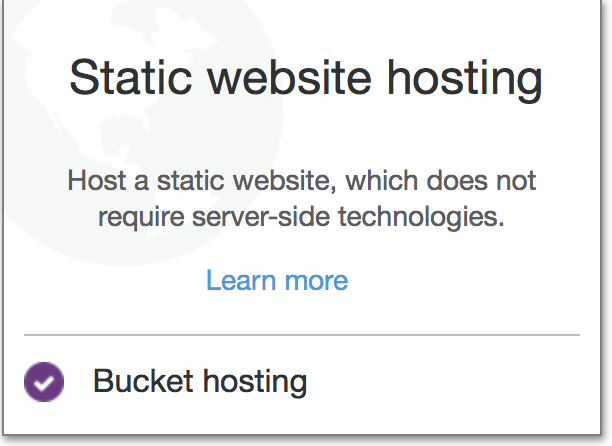


* 1. Use DNS names at least 3 characters no more than 63
  2. Names are shared meaning use a naming convention like bartonnameofbucket
  3. No charge for encryption at rest. Amazon S3 default encryption provides a way to set the default encryption behavior for an S3 bucket. You can set default encryption on a bucket so that all objects are encrypted when they are stored in the bucket. The objects are encrypted using server-side encryption with either Amazon S3-managed keys (SSE-S3) or AWS KMS-managed keys (SSE-KMS). When you use server-side encryption, Amazon S3 encrypts an object before saving it to disk in its data centers and decrypts it when you download the objects. For more information about protecting data using server-side encryption and encryption key management, see [Protecting Data Using Encryption](https://docs.aws.amazon.com/AmazonS3/latest/dev/UsingEncryption.html). Default encryption works with all existing and new S3 buckets. Without default encryption, to encrypt all objects stored in a bucket, you must include encryption information with every object storage request. You must also set up an S3 bucket policy to reject storage requests that don't include encryption information.



# Static Websites

* 1. S3 Can host static websites



* + 1. You will need 2 buckets (name has to be DNS name)
       1. [www.website.com](http://www.website.com) - real website bucket
       2. website.com – redirect bucket

# Connecting to Macs and PCs

* 1. Use Expandrive to make seamless access for Macs and PCs (PC’s do have issues)

# Using CLI

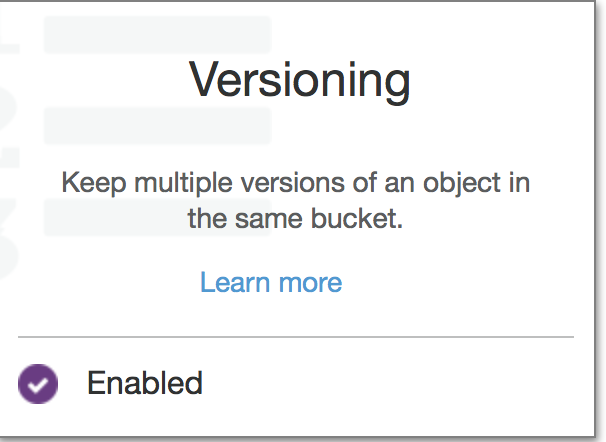
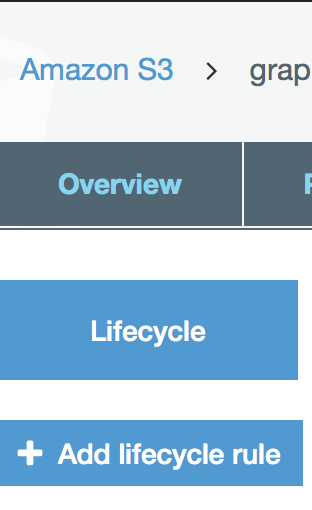
* 1. Command line is best for uploading large files so you can see progress and time required
  2. Use awscli for backing up Microsoft and Linux servers

aws s3 cp

* 1. Can use sync to make a backup of a bucket
     1. /root/.local/bin/aws s3 sync s3://bartonathletics/ s3://bartonarchive/bartonathletics/
  2. Can use groups to control access (backup account remember to go to policy and attach S3)
  3. Recommend using Permissions – Bucket Policy to assign permissions
     1. Ask for a sample script



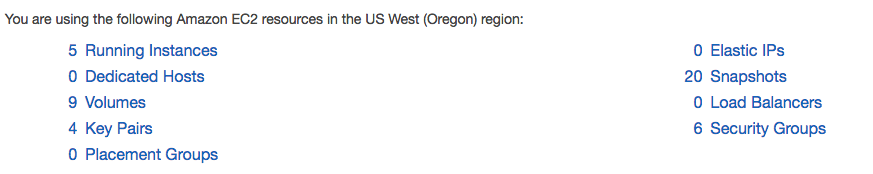
* 1. Recommend setup versioning and lifecycle rules
  2. If when you access the web page if the files download instead of opening then at a terminal use curl –I website.address.com and look at Content-Type: text/html if anything else is listed then you need to change the way you upload such as: aws S3 cp –content-type=”text/html”

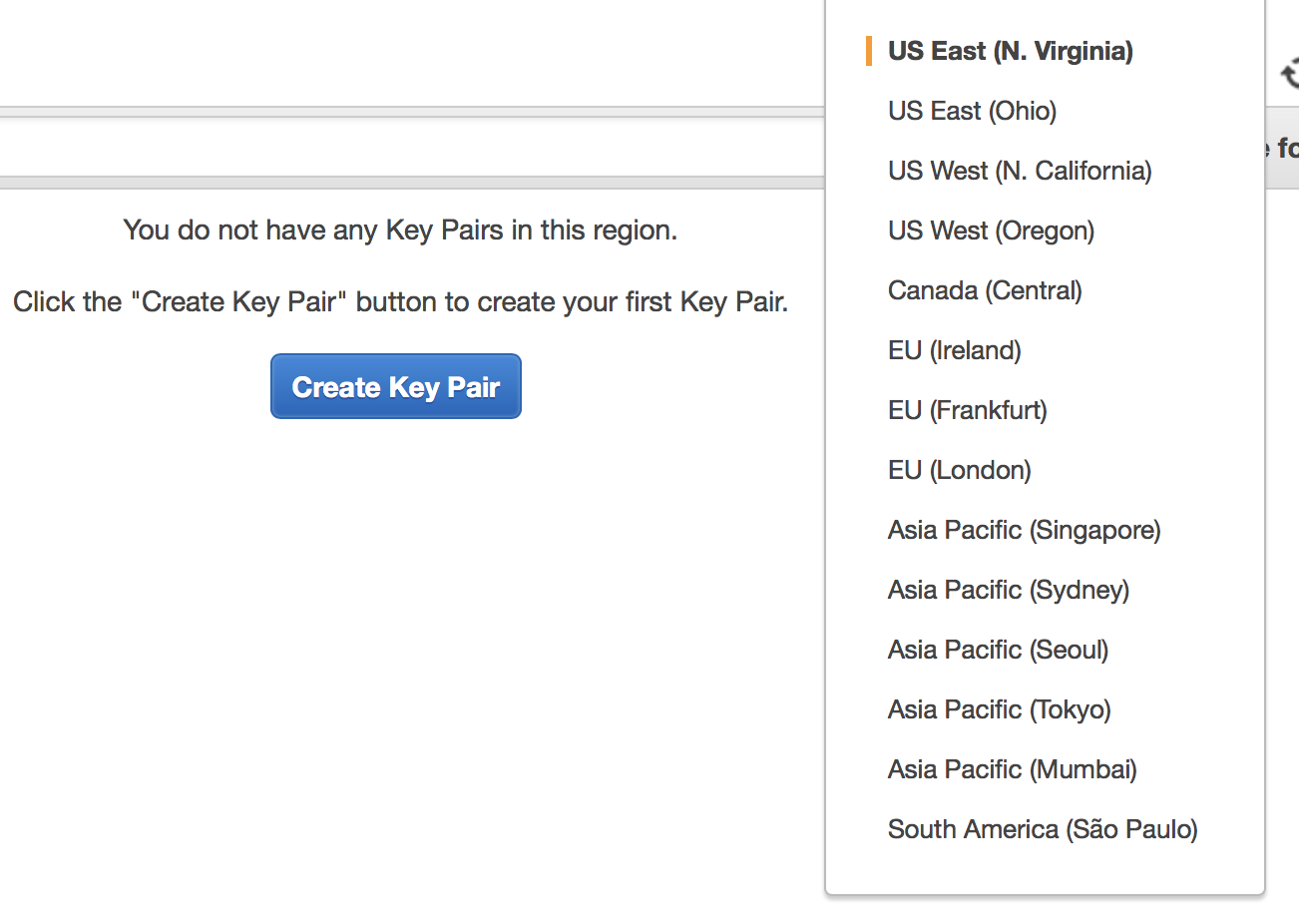
1. **Working with AWS Command Line Interface**
   1. Use aws command line interface to backup servers (and lots of other stuff)
   2. Requirements: Must have python3.4
   3. Some File commands are: ls cp sync
   4. Some Options are: recursive, exclude,
   5. Full explanation [click here](http://docs.aws.amazon.com/cli/latest/reference/s3/index.html)
   6. To install on Mac or PC [click here](http://docs.aws.amazon.com/cli/latest/userguide/installing.html)
   7. Can map Microsoft shares to Linux server and use Linux server to zip, sync and backup
   8. Recommend you create a backup user and give policy AmazonS3FullAccess
   9. To setup aws cli on Linux and Microsoft [click here](http://docs.aws.amazon.com/cli/latest/userguide/installing.html)
      1. AWS cli is installed on a per user basis.
   10. To determine size of a bucket run the command below
       1. aws s3 ls --summarize --human-readable --recursive s3://archive

# EC2 building servers

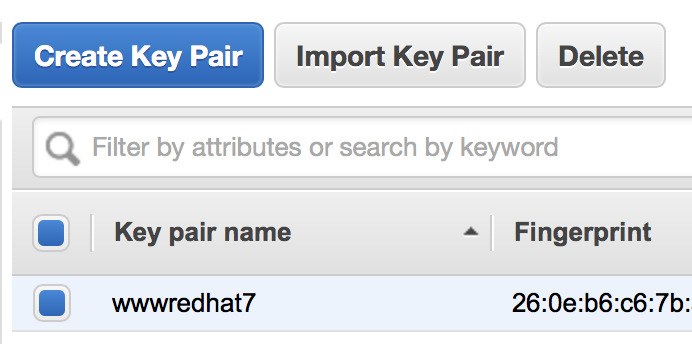
1. **EC2 Dashboard – where you build servers**



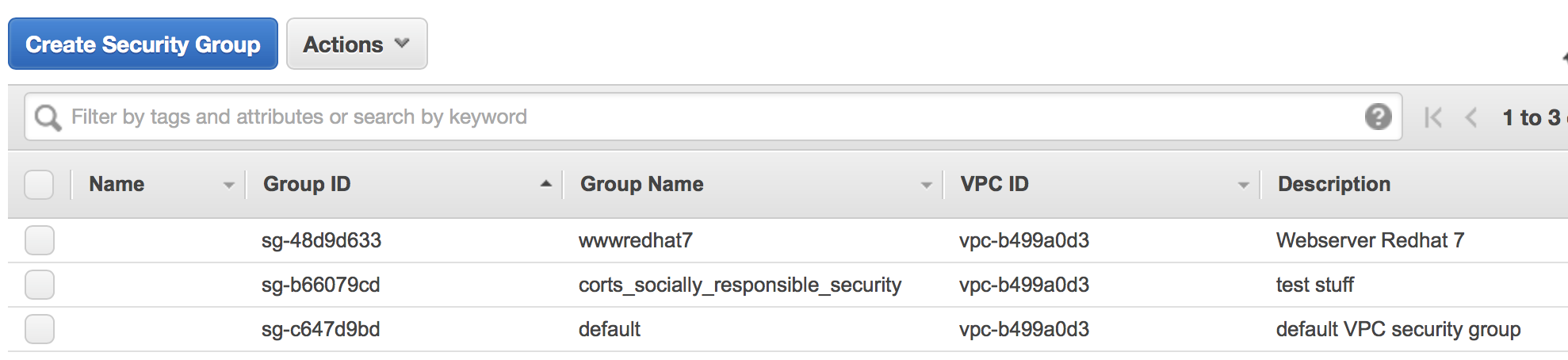
* 1. Regions matter, if servers, key pairs or security groups are missing check your region



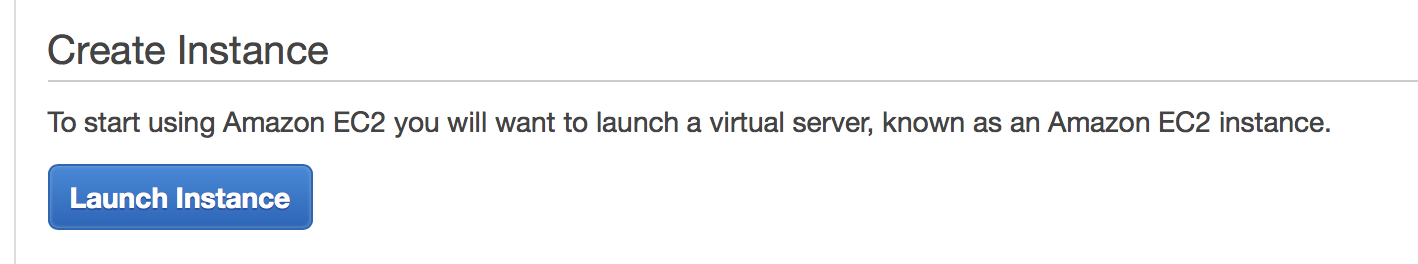
* 1. Setup Key Pairs – by default servers do not use passwords just key pairs.



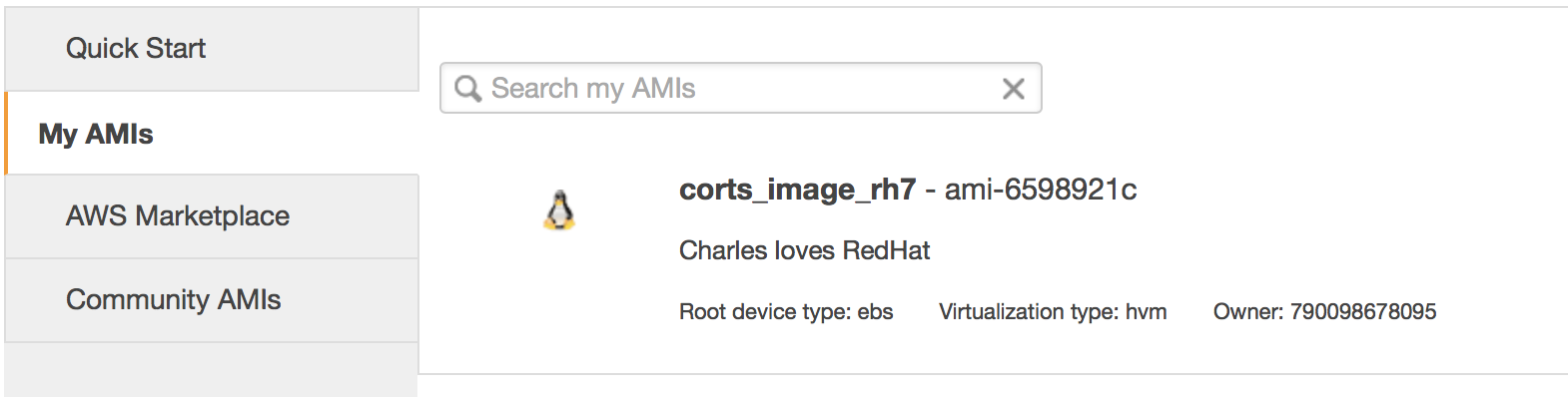
* + 1. Setup Key Pairs based on who will be accessing server – just internal or 3rd party
    2. If you lose the key you lose access to the server, can only download once.
    3. Recommend you put clear names as naming convention
  1. Setup Security Groups



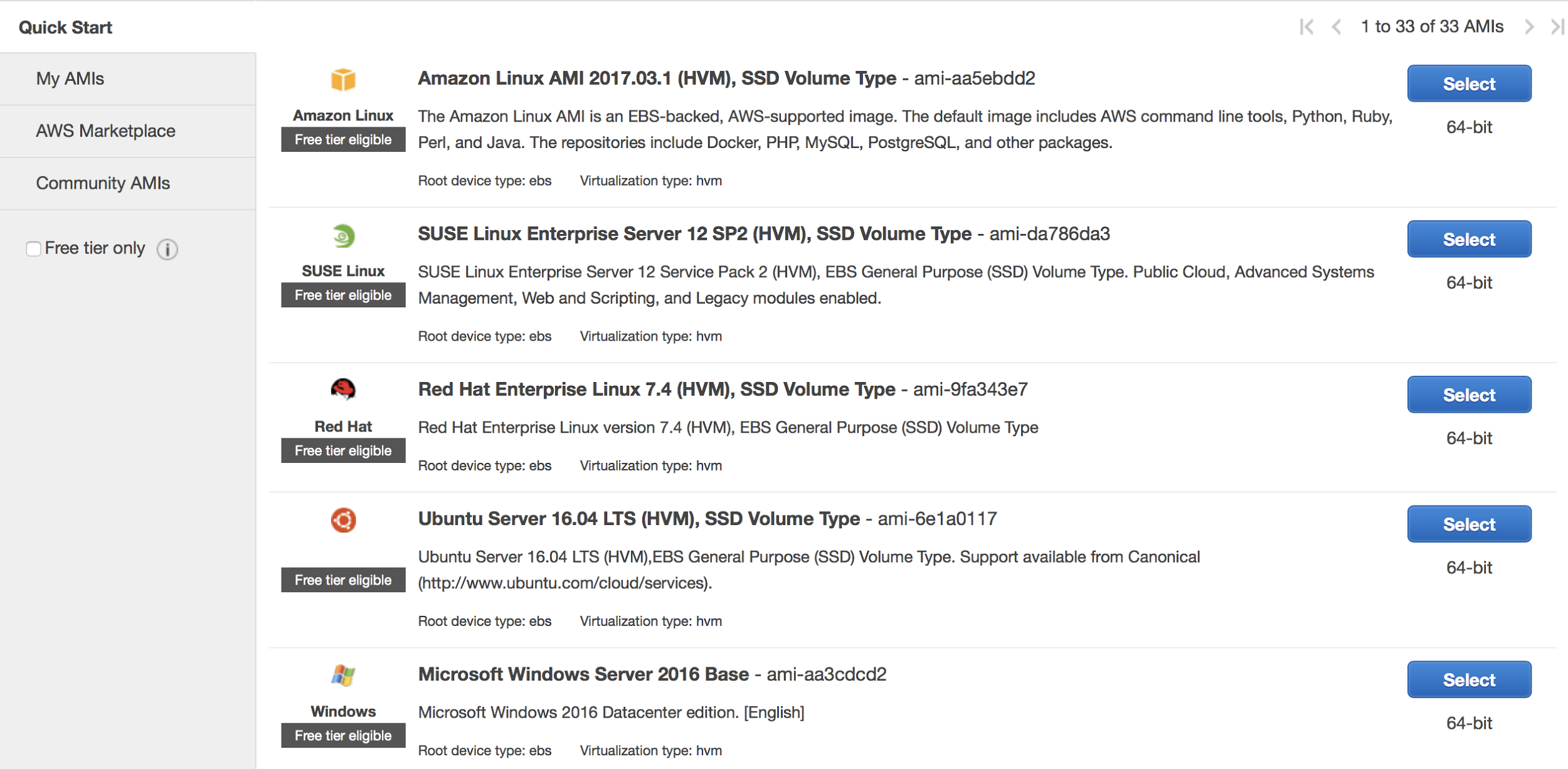
* + 1. Each Server can have its own unique security group (firewall like) or can use the same one.
    2. Get with a firewall person to assure you have set up security group safely.
  1. Running Instances – to begin server setup only after you have the key pair and security groups finished.



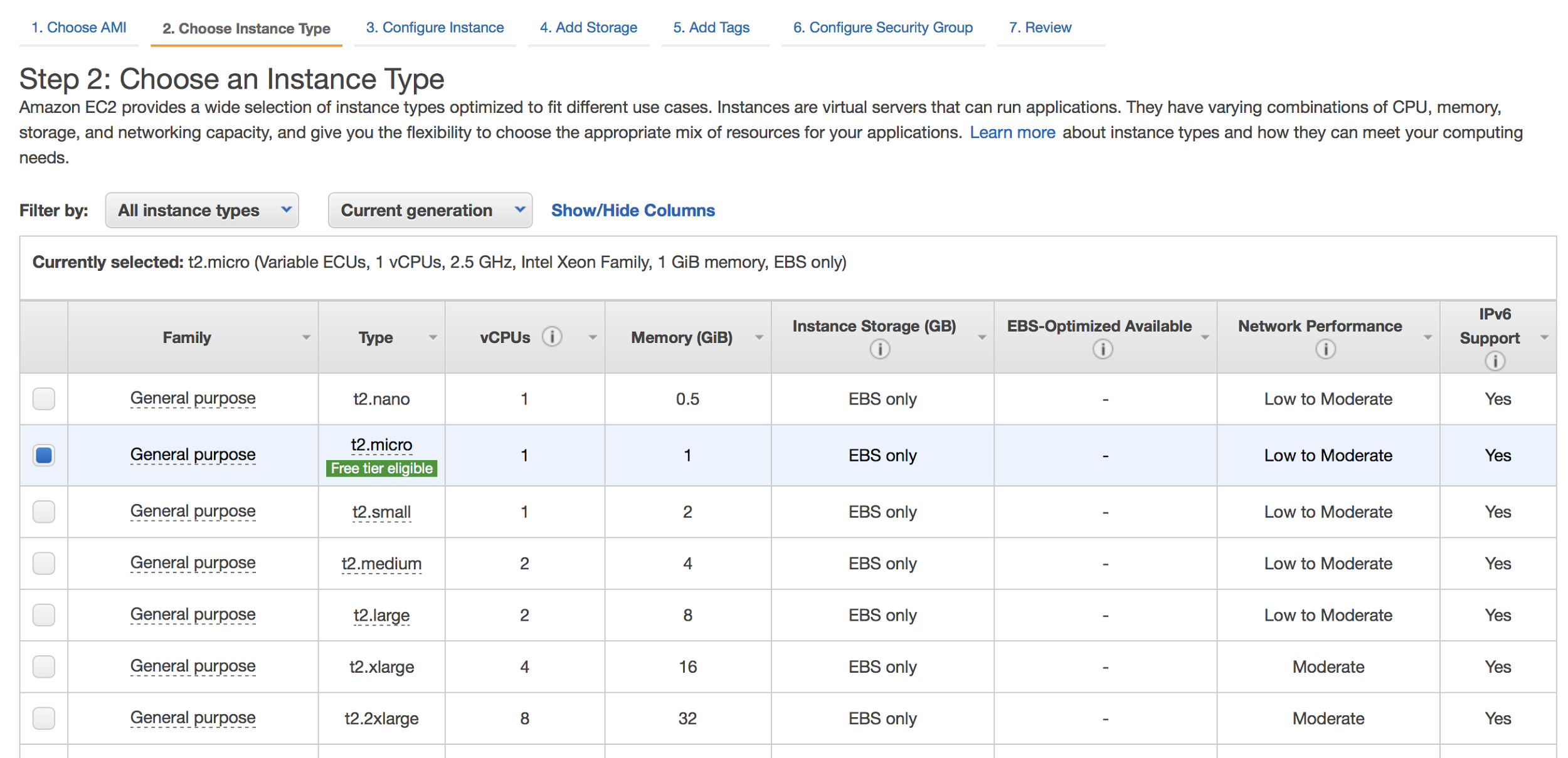
* + 1. Launch Instance – Linux (be careful launching an Amazon database the one we tested cost $300 per month, you can always install the MySQL or other open source database yourself on your server for no cost.
    2. Before you look at all the Amazon Machine Images (AMI) take a look on the left side for – My AMIs. This is for the future which allows you to create pristine images and deploy new servers quickly.



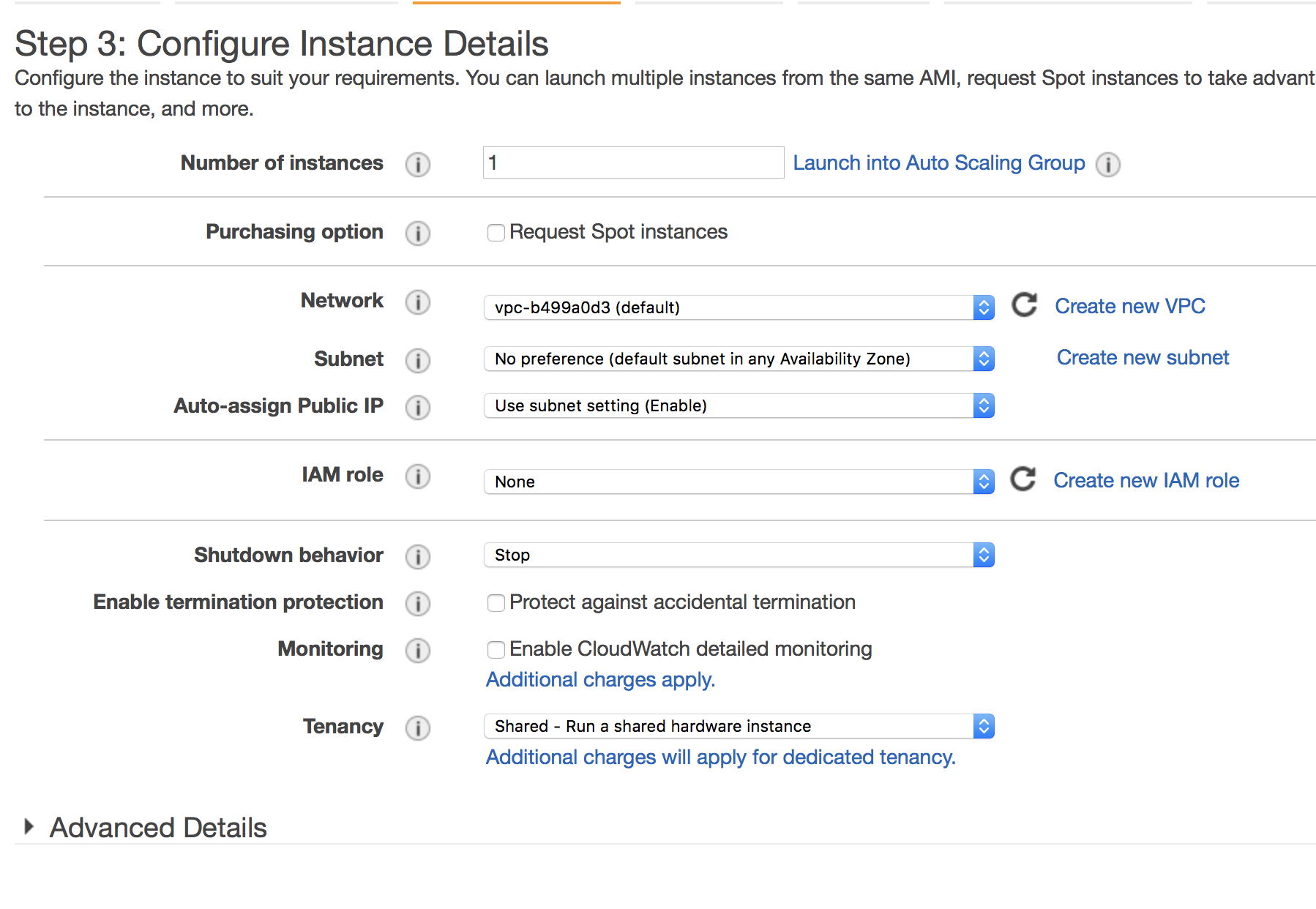
* + 1. Select Redhat Enterprise (AWS uses RH7)



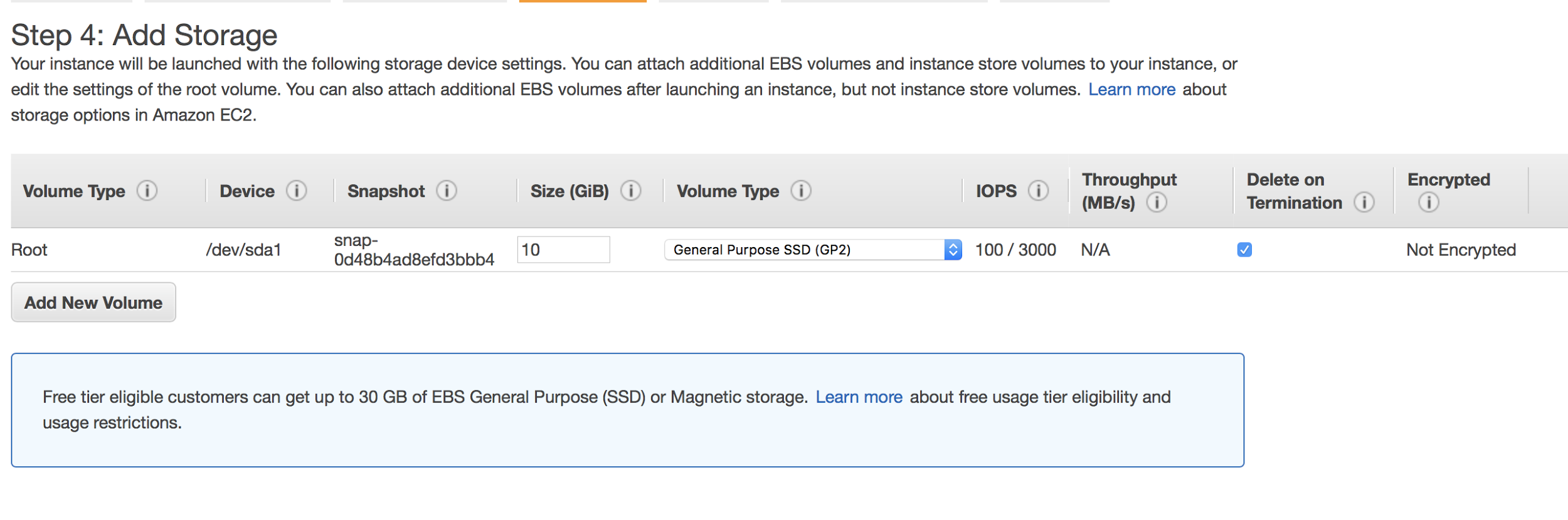
* + 1. Review the tiers, but select the free one for now. You can run a Drupal website on the free tier.



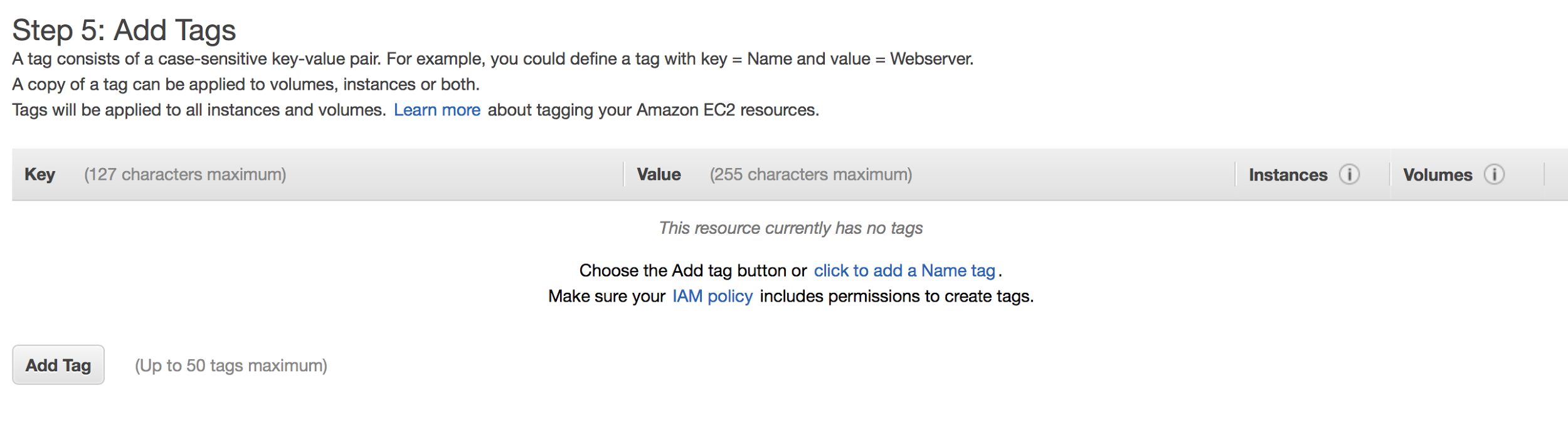
* + 1. Configure the instance



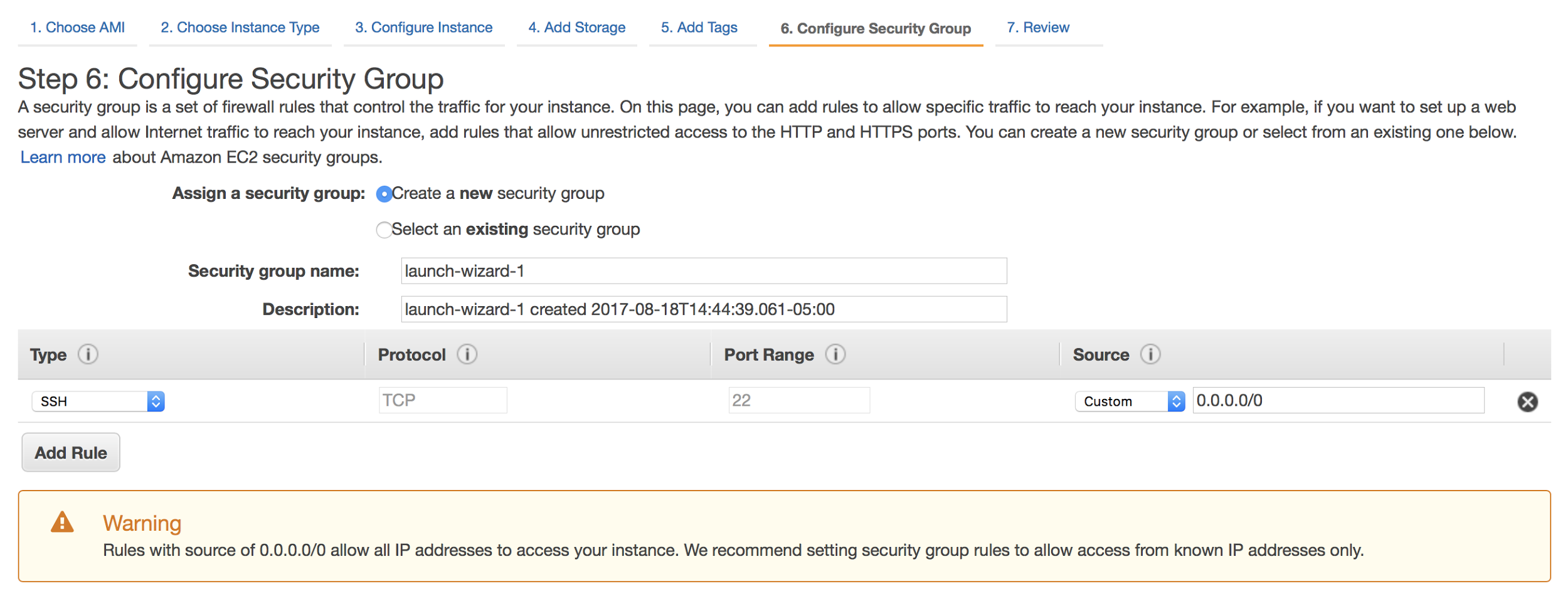
* + - 1. Pick your subnet
      2. I would recommend keeping servers in the same subnet
    1. Add Storage



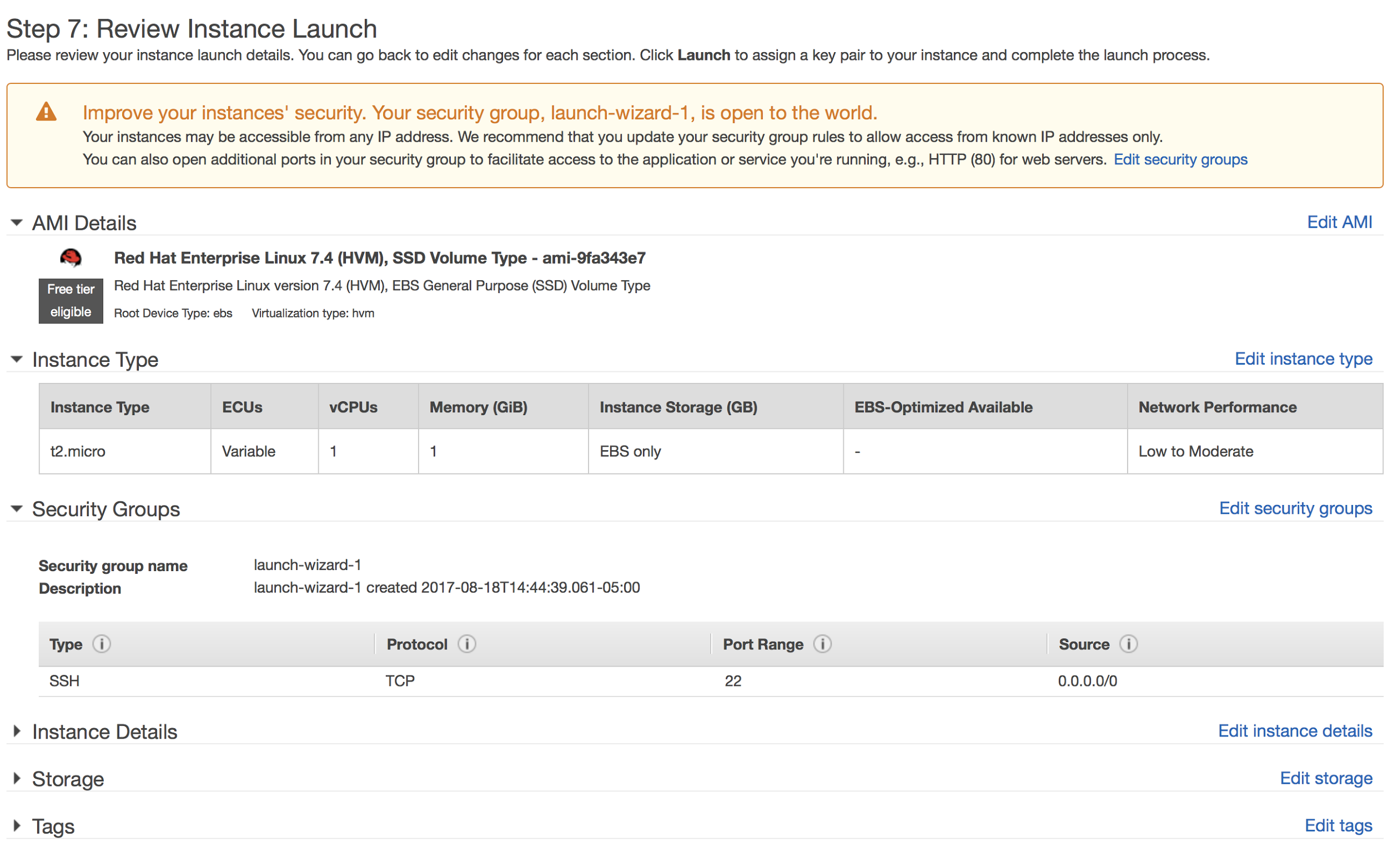
* + - 1. All servers are built with SSD drives so they are fast
      2. Create as many volumes as needed
    1. Add Tags: Fill out tags so you can see uses



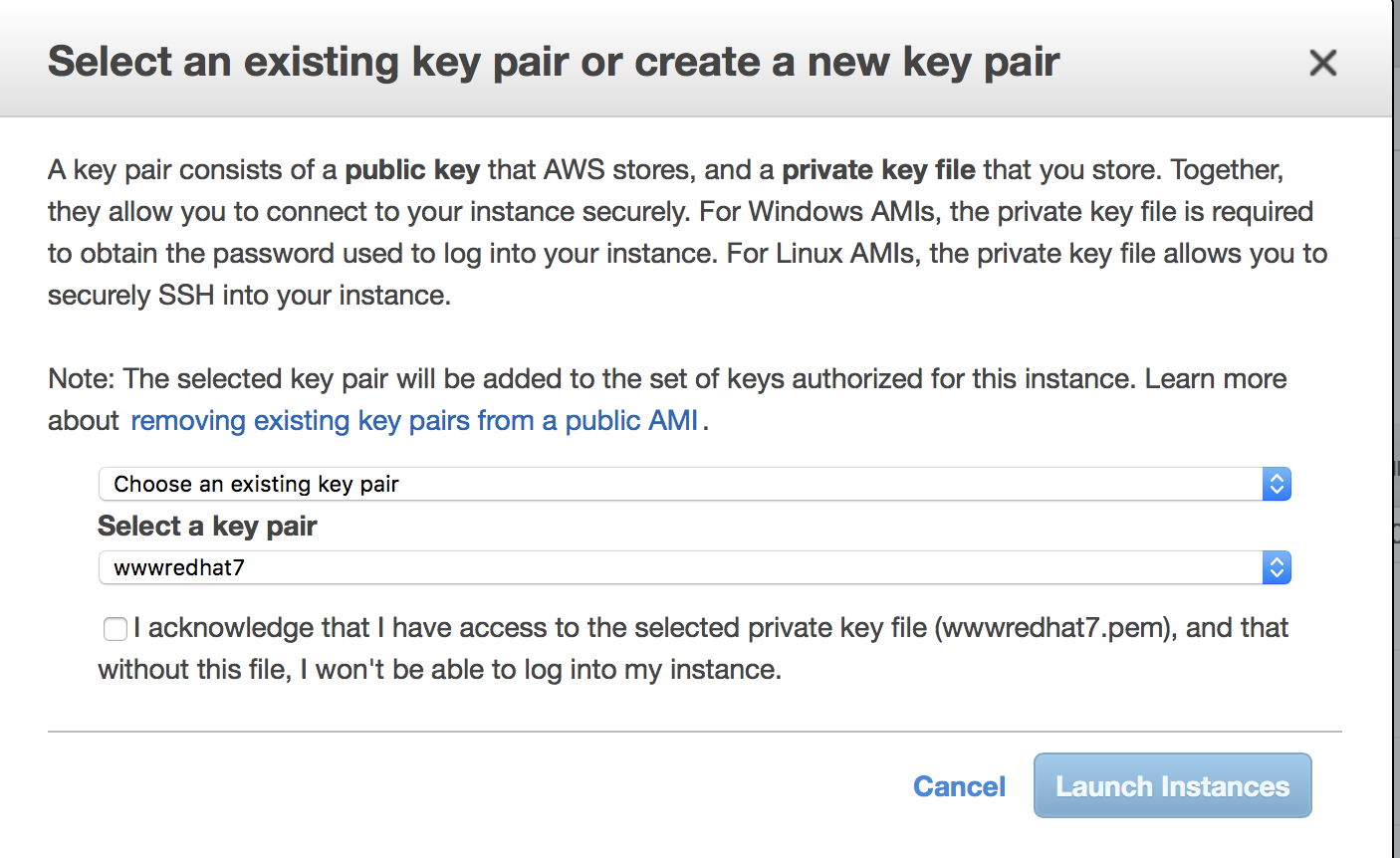
* + 1. Select Security Group



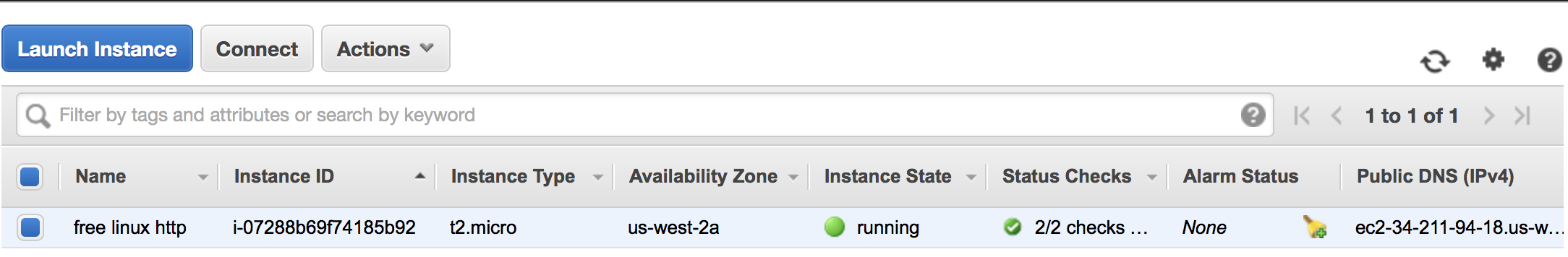
* + - 1. Recommend you have security groups ready before this step. You do not want to create a security group right quick.
    1. Review and Launch



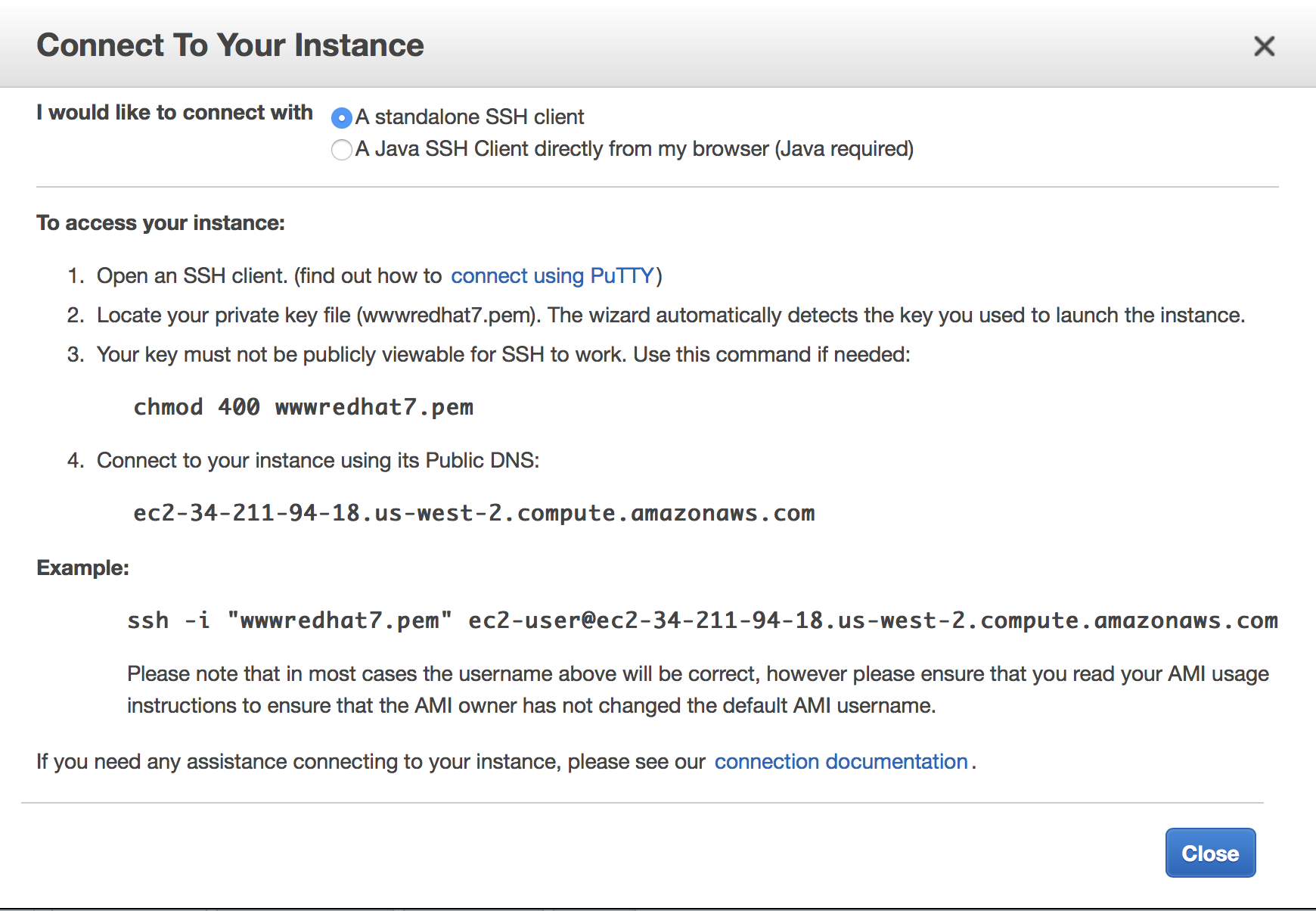
* + 1. Pick your key pair



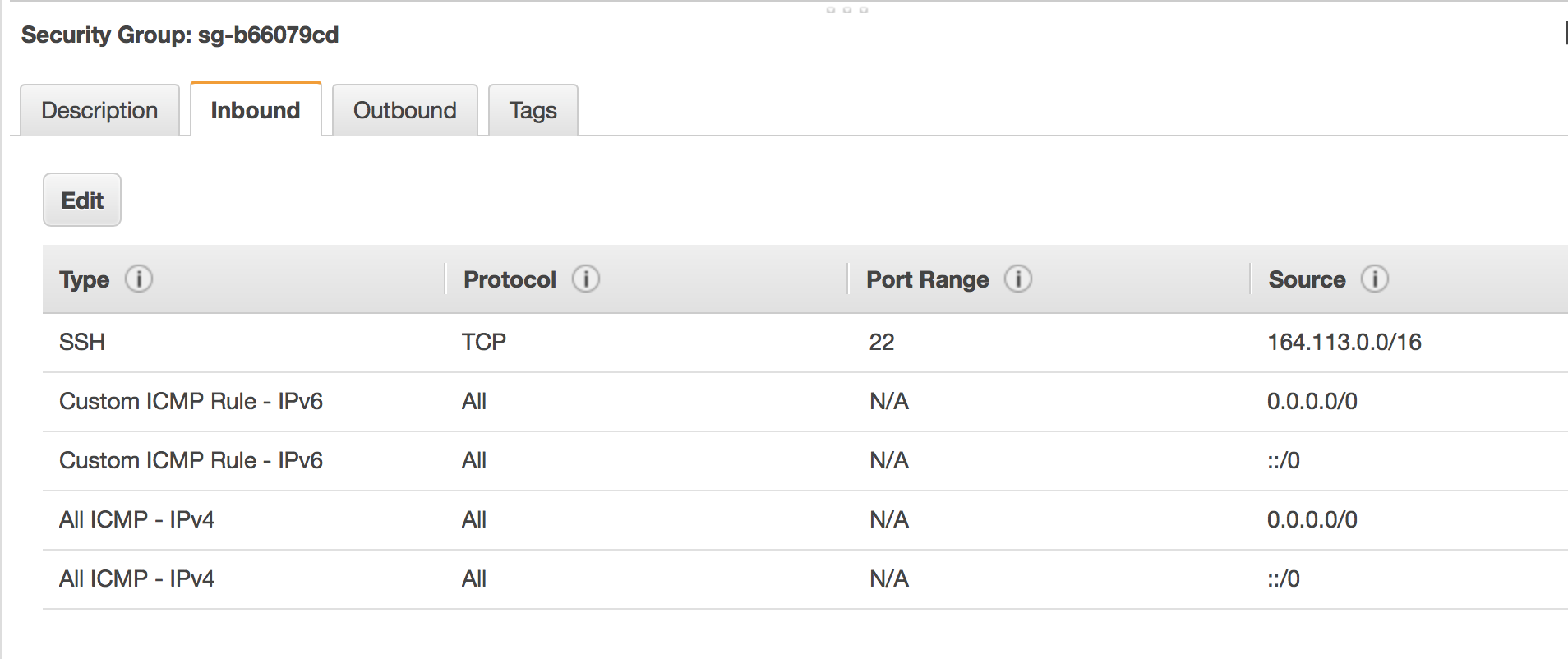
* + 1. Launching only takes a couple of minutes
    2. EC2 Dashboard name your instance



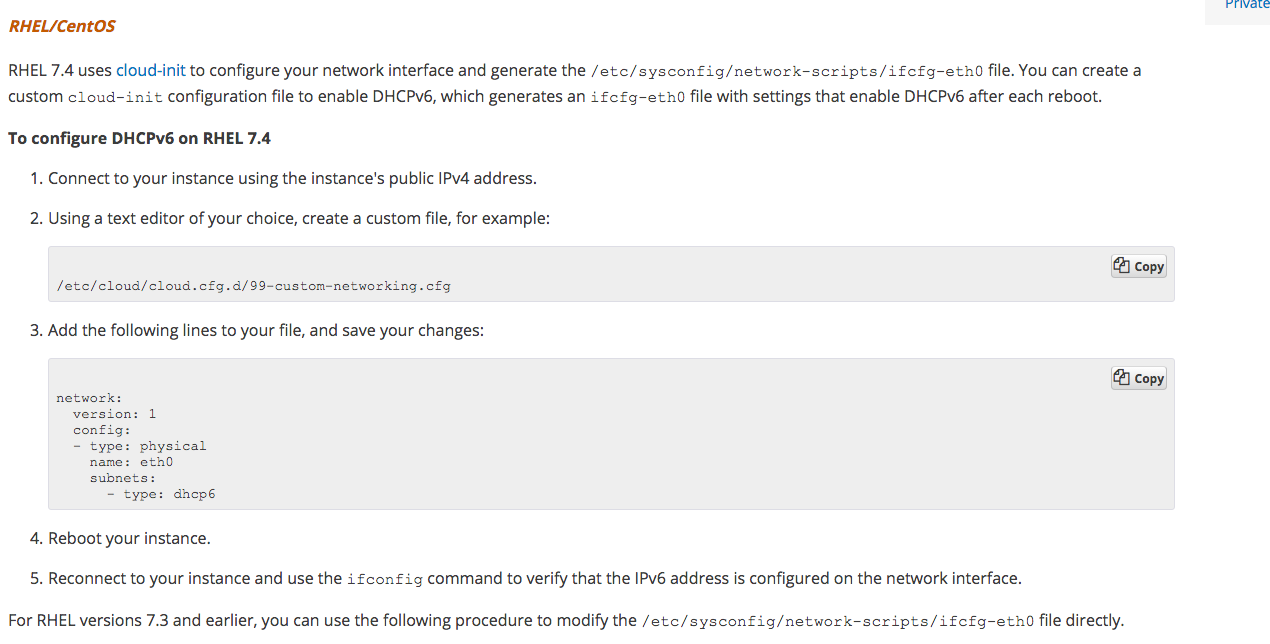
* + - 1. Recommend start name with free or not free so you can track costs
      2. Click on Connect and instructions will be given to connect to your server



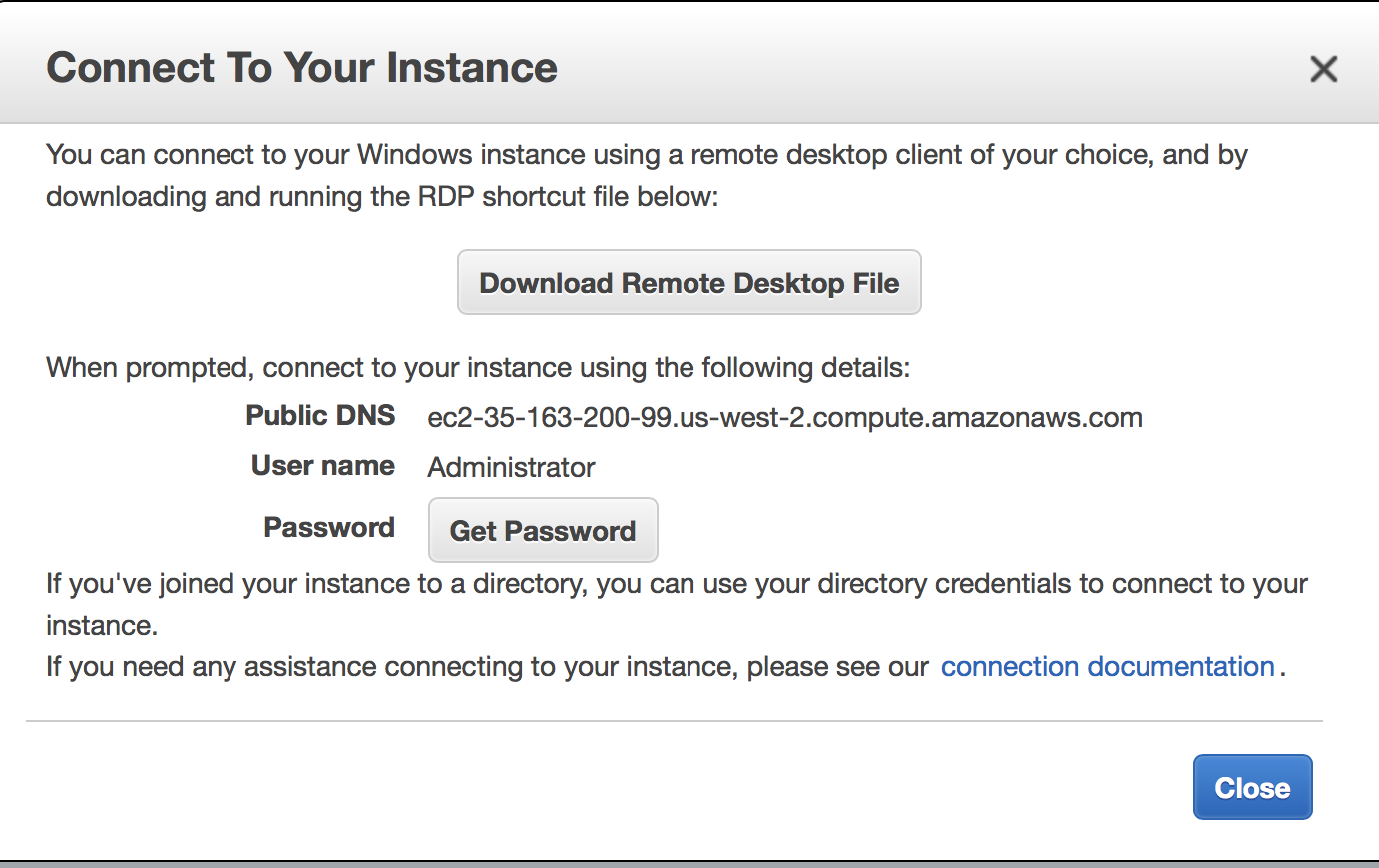
* + - 1. You will connect as ec2-user using key pair to be root use sudo su – root
      2. If you decide to create a user which need to connect externally then remember that user need to have the key pair in its home directory. [Click here](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/managing-users.html)
      3. Can connect to instance with VNC if you want a GUI
      4. By default, your Linux instance will not have a swap file, since you are going free you will need one. Change the size below as you see fit 1M is minimum.
         1. Run dd if=/dev/zero of=/swapfile bs=1M count=1024
         2. Run mkswap /swapfile
         3. Run swapon /swapfile
         4. Add this line /swapfile swap swap defaults 0 0 to /etc/fstab
      5. Remember the system has 2 firewalls one local software and one AWS security group.



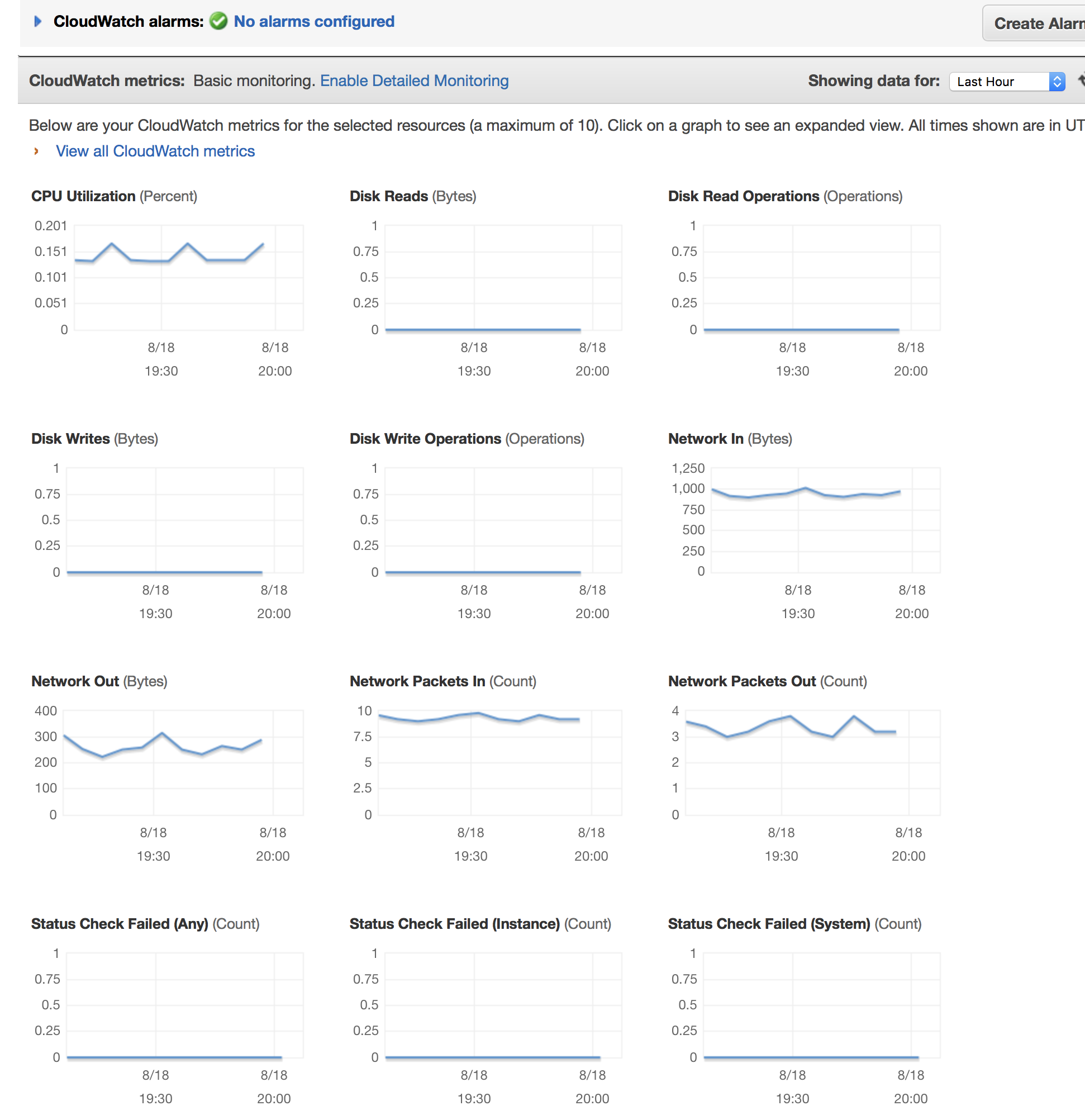
* + - 1. Log files by default can only be read by root if you want other users to read them then change permissions.
      2. Once you have a good installation remember to create images AMIs. (Server will need to be shutdown to create image. Does not appear to be any cost for them.
      3. Remember to turn on IPv6



* 1. Launch Instance – Microsoft
     1. All the steps above for launching is the same expect.
        1. Security group must allow remote desktop access
        2. Since Microsoft servers cannot only be connected to with passwords you have to decrypt the key pairs.



* + - 1. Active Directory will not work with NAT so all servers must be in same subnet.
      2. VPN will need to be setup to connect domain members to your local campus.
    1. AWS have built in Monitoring



### Expand Volume – make hard drive bigger

First label the volumes with the names of the servers so you can pick the correct volume

Next create a snapshot in case something goes wrong.

<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-creating-snapshot.html>

Modify the volume size

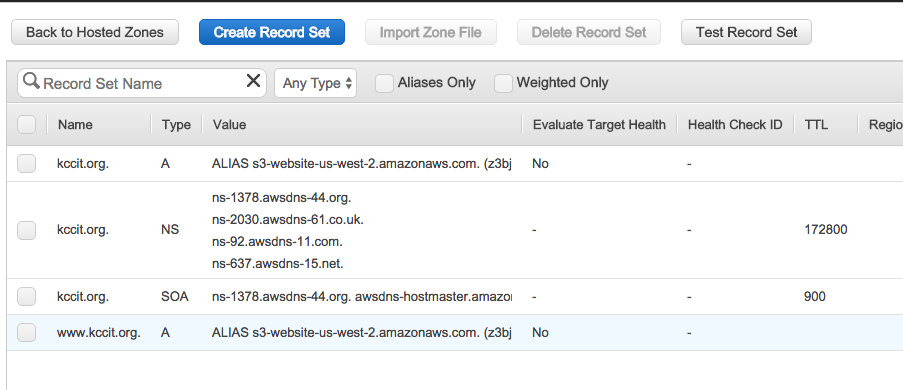
<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/console-modify.html>

Resize the volume

<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/recognize-expanded-volume-linux.html>

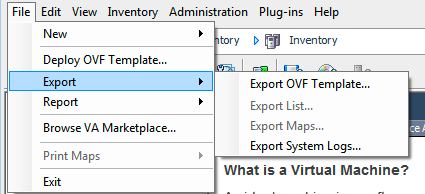
# Route 53 - DNS

1. **Route 53 AWS DNS**
   1. If you are running S3 Websites then the easiest way to have DNS work is to move DNS to Amazon Route 53.



# VMWare

1. **AWS and VMWare: How to Import VMWare images**
2. **Can use AWS Migration Service for Free -** [**https://us-west-2.console.aws.amazon.com/migrationhub/dashboard**](https://us-west-2.console.aws.amazon.com/migrationhub/dashboard)
   1. Log in to your VMWare system, shut down the server, export the image as a single file ova.

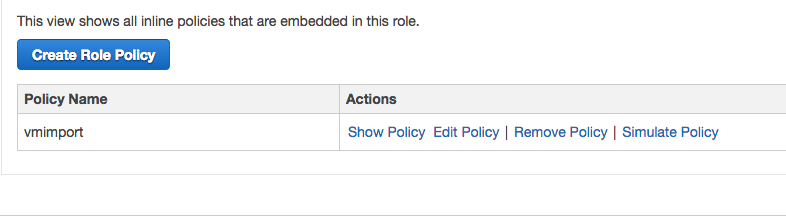




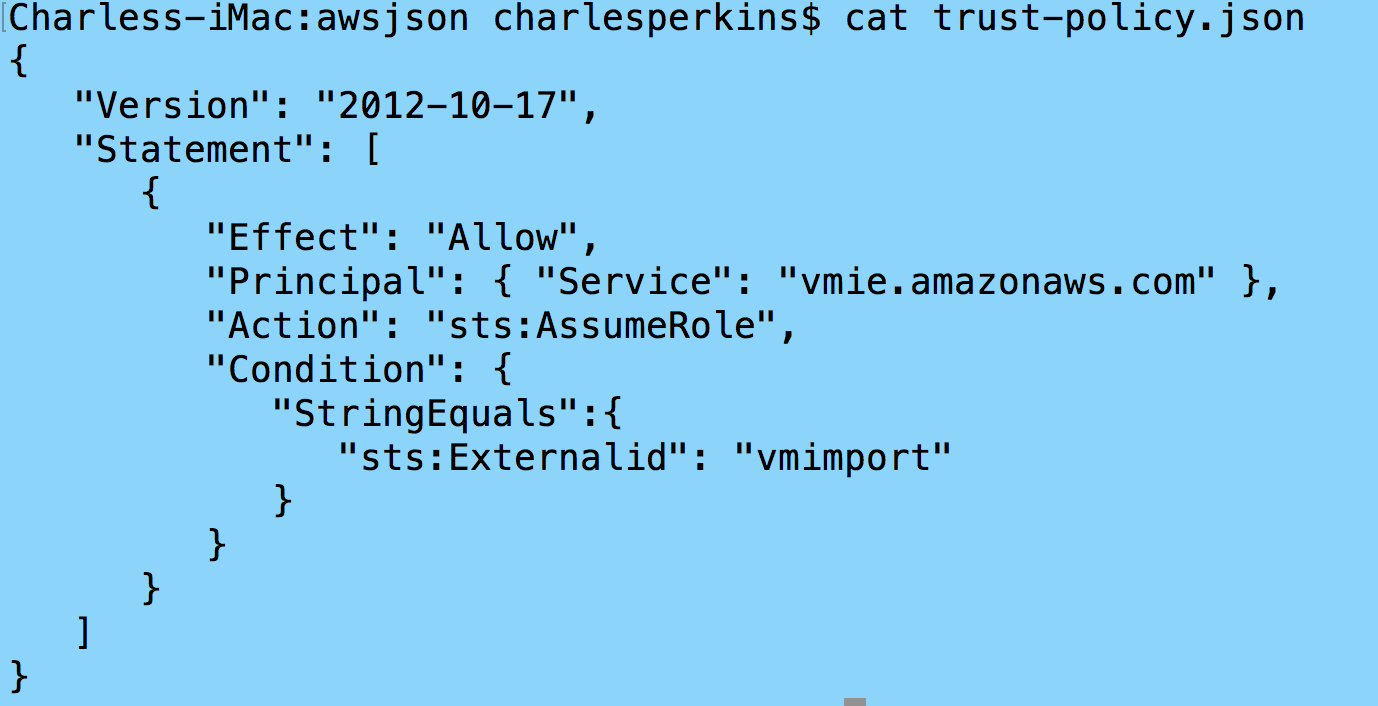
* 1. Setup a S3 bucket to hold images and upload your images.



* 1. Create a vmimport role and policy as per web page <http://docs.aws.amazon.com/vm-import/latest/userguide/vmimport-image-import.html#w2ab1b9c15b9>





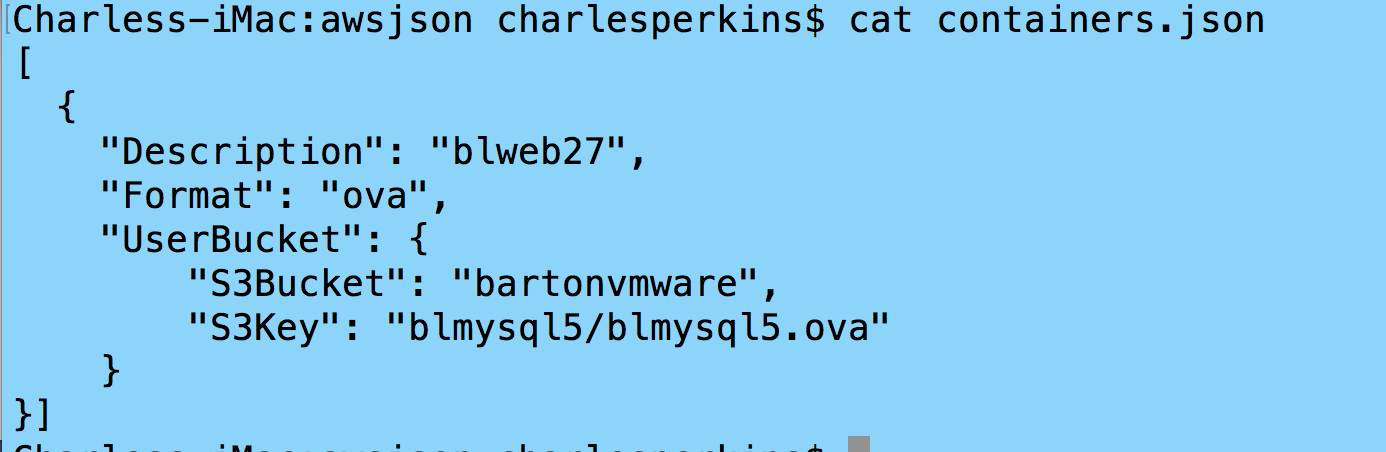


* 1. Using aws cli and json import image to [create](http://docs.aws.amazon.com/vm-import/latest/userguide/vmimport-image-import.html) an AMI. Execute from folder with json files.

aws iam create-role --role-name vmimport --assume-role-policy-document file://trust-policy.json

aws iam put-role-policy --role-name vmimport --policy-name vmimport --policy-document file://role-policy.json

aws ec2 import-image --description "blxetest redhat 6" --license-type BYOL --disk-containers file://containers.json



aws ec2 describe-import-image-tasks --import-task-ids import-ami-fgafw3e1

# Penetration Testing

1. **AWS Penetration Testing**
   1. AWS provides vulnerability and penetration [testing](https://aws.amazon.com/security/penetration-testing/)
   2. Will Not test m1.small, t1.micro or t2.nano because they are shared resources
   3. Amazon Inspector enables you to analyze and identify potential security issues.

# DDoS

1. **AWS and Distributed Denial of Service (DDoS)**
   1. AWS Shield is a managed Distributed Denial of Service (DDoS) protection service that safeguards web applications running on AWS
   2. All AWS customers benefit from the automatic protections of AWS Shield Standard, at no additional charge.

# Helpful Hints

1. **Helpful Hints**
   1. Barton is on KanREN which gives us a direct connection to AWS through Internet2. This is important if you are going to backup servers to AWS
   2. Can produce reports on EC2/S3 to determine how much servers are costing you.
   3. Amazon representative: marshat@amazon.com 206.435.1147
   4. Video [Amazon Storage Services for Education](https://connect.awswebcasts.com/p7m0ahyy2se/?proto=true).