Using Cloud Compute Services for Radiance Simulations

Andy McNeil

Disclaimers

- This presentation is offered with no warranty, use at your own risk.
- Andy is a hobbyist at best. Consult a network security expert for much better advice regarding security than what you're about to receive.
- There is likely a better way to do everything I'm going to show you.

shhhh...

There is no cloud.
It’s just someone else’s computer.

Why use someone else's computer?

- Computing resources scale with needs
  - Add and drop instances as your workload changes
  - No need to coordinate with coworkers for running jobs
- Zero up front cost | Hello Freelancers!
- Track costs by project and bill computing expense to client
- Use Linux-only Radiance features without having to get a linux / mac computer

Andy’s experience in 2015-2016

Lighting Team

Andrew McNeil – June 14, 2016 at 01:20 PM

As a contractor to Arup I (mostly) provide my own computing resources. For Radiance simulations I use the Amazon Elastic Compute Cloud (EC2). I've just been reviewing my usage statistics over the last 11 months, and thought I'd share with the lighting community.

With the EC2 you choose the specs of your instance when you request it. They have instances with specs ranging from 1 to 40 CPUs. I tended to use instances ranging from 32-40 cpus (depending on market pricing).

My usage statistics over the Past 11 months:
Total Cost: $888
Total CPU hours: 69,028
Cost per CPU hour: $0.013

On average over the 11 months I had 8.5 CPUs for $900, which is competitive with purchasing a high end desktop computer (assuming a 3-4 year life). But the real benefit is scalability. You see I didn't have an 8 CPU computer, I had practically unlimited computing resources at my disposal when I needed them. For example, right now I have 4 x 40 CPU instances running (160 CPUs total!) and it is only costing me $1.40 an hour at spot market pricing.

- 11 months
- $900
- 69,028 CPU hours
  (equivalent to 8.5 constantly running CPUs)
- 160+ CPUs running for short periods
- Zero CPUs most of the time

Who’s Computer?

- AWS (Amazon)
- Azure (Microsoft)
- Google Cloud
- IBM Cloud
- Cloud & Heat - Distributed mini data centers that also provide heat to buildings
- And lots more…

Considerations - just a lazy list

- Access
- Storage
- Instance Configurations
- Computing Costs
- Spot / Low-Priority / Preemptible offering
- Your employer’s preference (and your client’s preference)

I chose aws

- It was 2015 - AWS was the leader in cloud services
- Now things are pretty even between biggest cloud providers
  (but I still use AWS ‘cause I’m an old dog) 

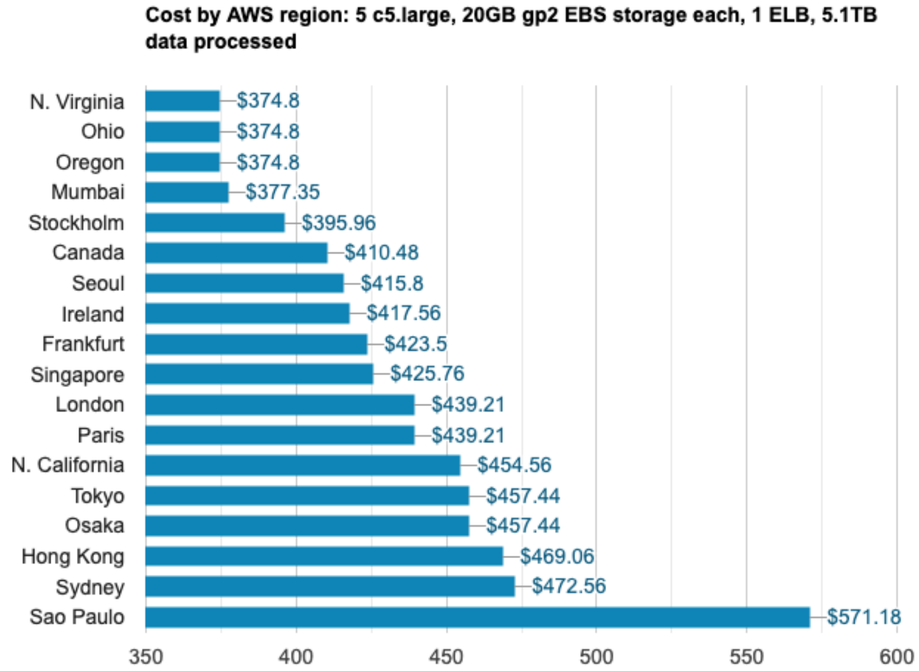
Getting Started with AWS in Twenty-one Easy Steps!

1. Create an AWS account
2. Choose a Region
3. Generate a key pair
4. Set up a security group
5. Choose spot or reserved instance
6. Select an instance type
7. Select a machine image
8. Launch instance
9. Connect to your instance
10. Install software
11. Save a machine image
12. Create persistent file storage
13. Mount persistent file storage
14. Upload Radiance model files
15. Start simulation
16. Set an alarm to terminate instance when simulation finishes
17. Launch an instance to retrieve results from file storage
18. Download results
19. Terminate instance
20. Get your cloud expenses and invoice your client
21. Bask in the glory of a job well done

AWS Regions

US East (Ohio)
US East (N. Virginia)
US West (N. California)
US West (Oregon)
Asia Pacific (Hong Kong)
Asia Pacific (Mumbai)
Asia Pacific (Osaka-Local)
Asia Pacific (Seoul)
Asia Pacific (Singapore)
Asia Pacific (Sydney)
Asia Pacific (Tokyo)
Canada (Central)
China (Beijing)
China (Ningxia)
EU (Frankfurt)
EU (Ireland)
EU (London)
EU (Paris)
EU (Stockholm)
Middle East (Bahrain)
South America (Sao Paulo)

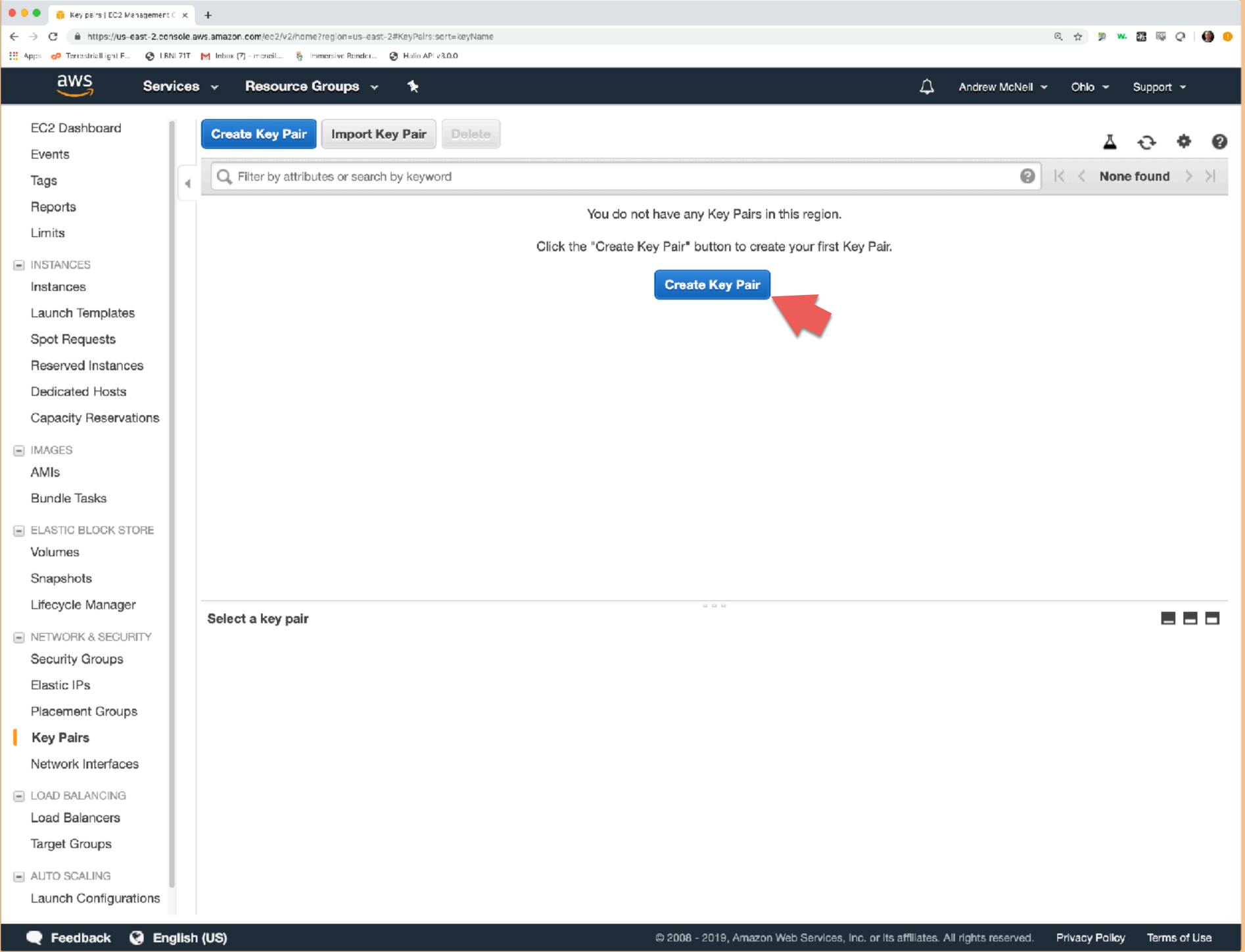
- Choose a region based on proximity and cost.
  - N. California is the most expensive of the US regions, and typically the last to get new features.



https://www.concurrencylabs.com/blog/choose-your-aws-region-wisely/

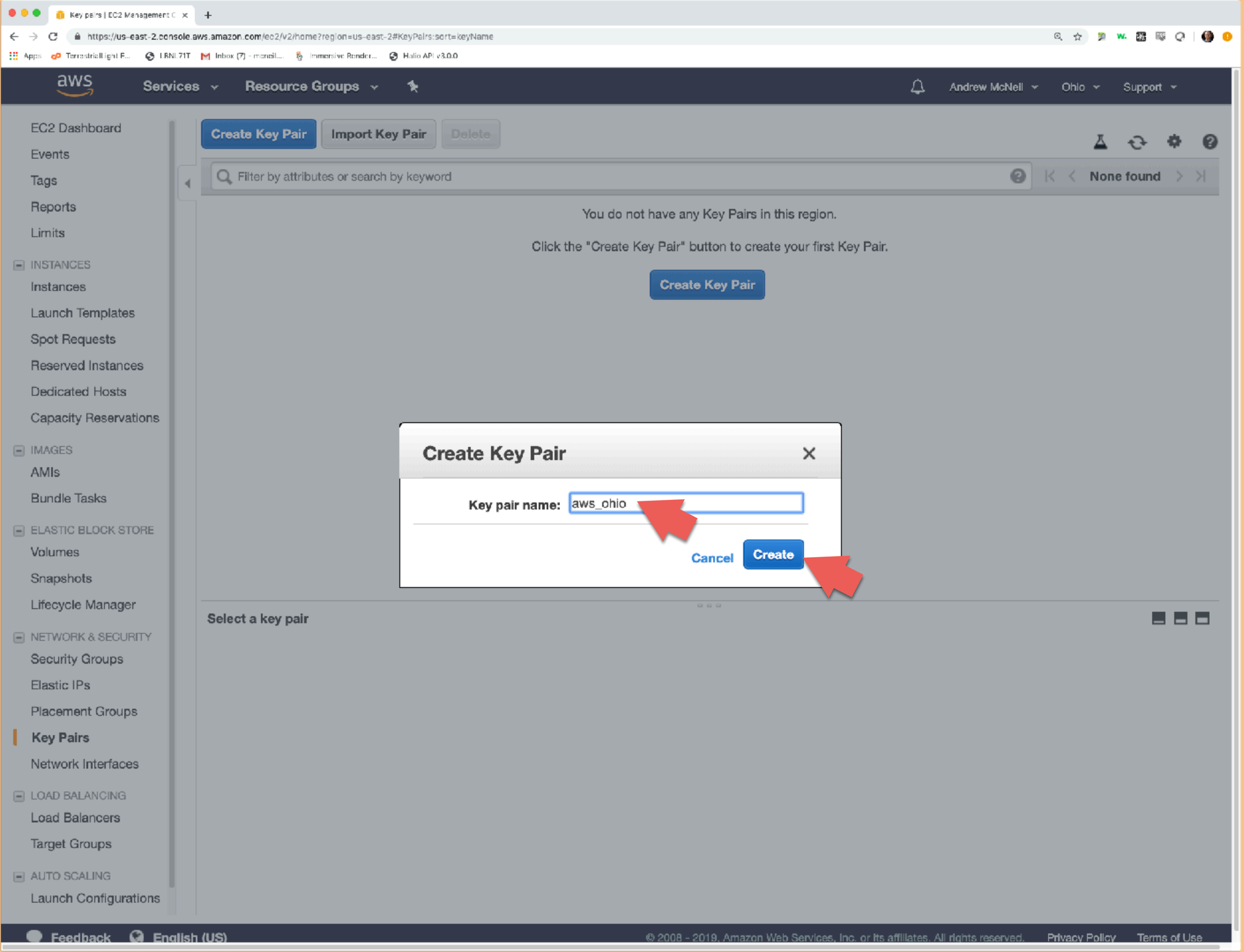
Key Pairs

- Key pairs are used to securely access your AWS resources.
- Each key pair has a public and private component.
  - AWS keeps the public part
  - You get the private part.
- The private key is downloaded to your computer when it is created
  - There is no way to get the private key again
  - If you lose the private key file you'll have to generate a new key pair and delete the old one

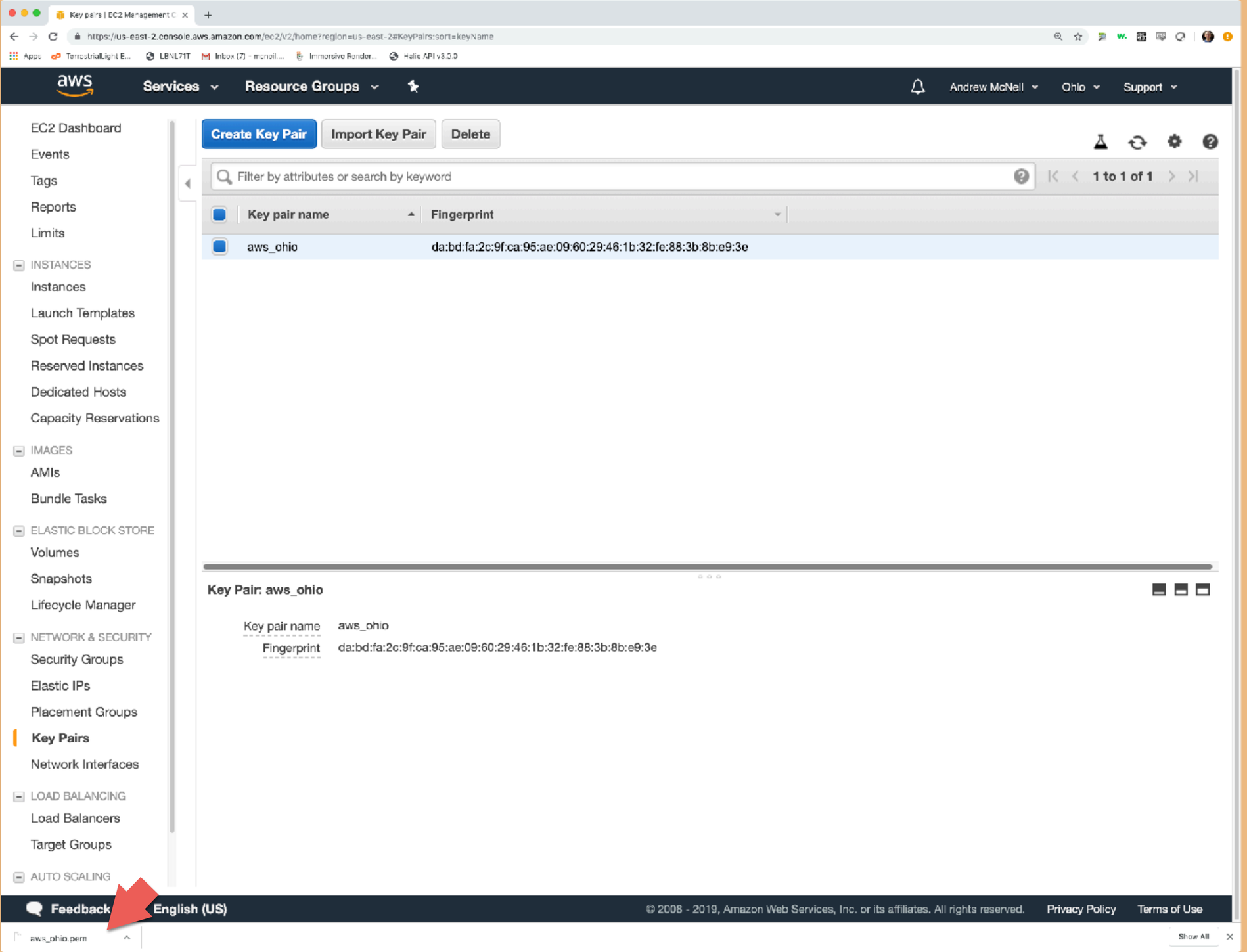


To start, you have no key pairs in your account.

Click:
“Generate Key Pair”



Enter a name for your key pair and click create.

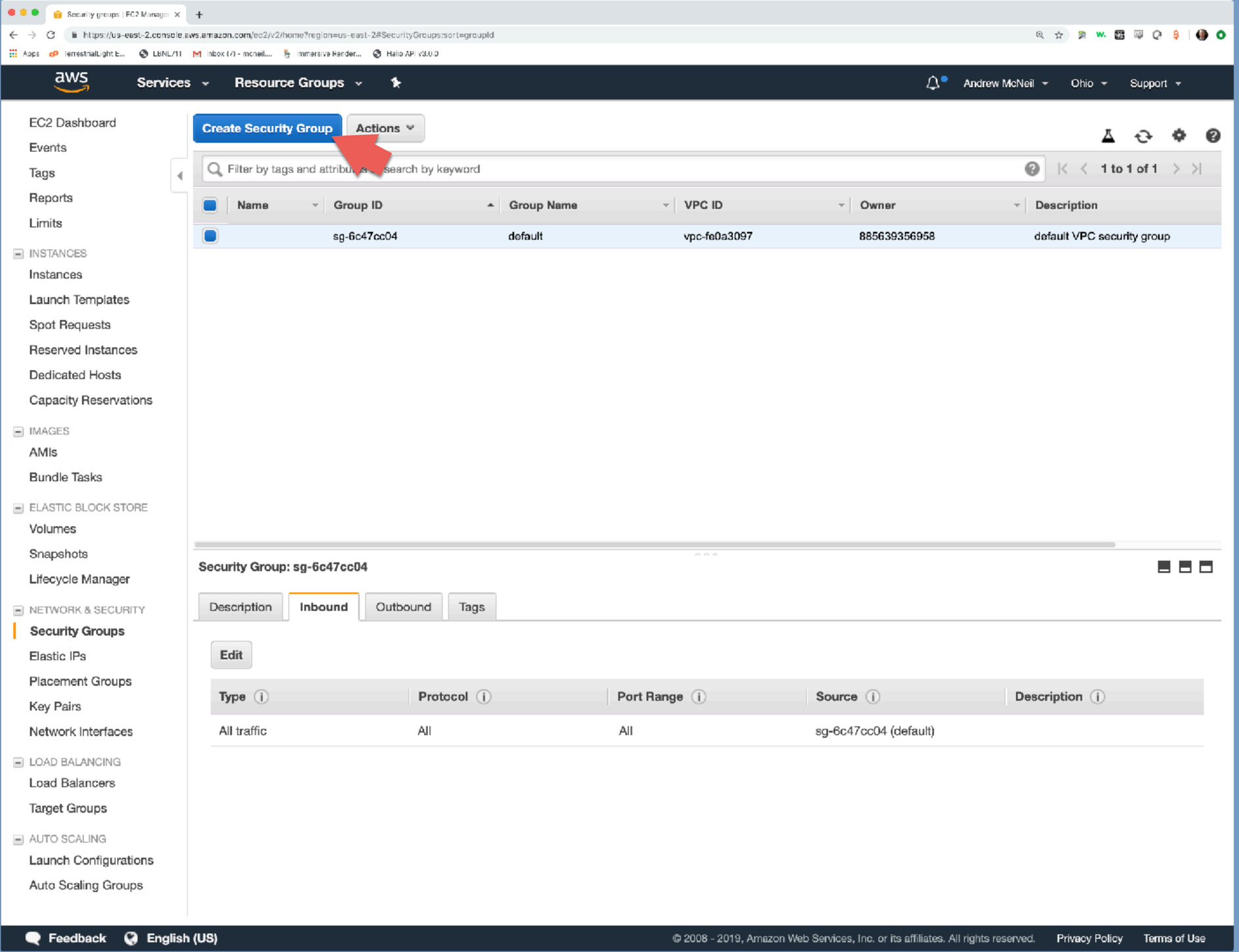


andy_ohio.pem is downloaded to my computer when I click “create”

Move your private key file somewhere secure and memorable. *I like to use a hidden folder for key files on my mac.*

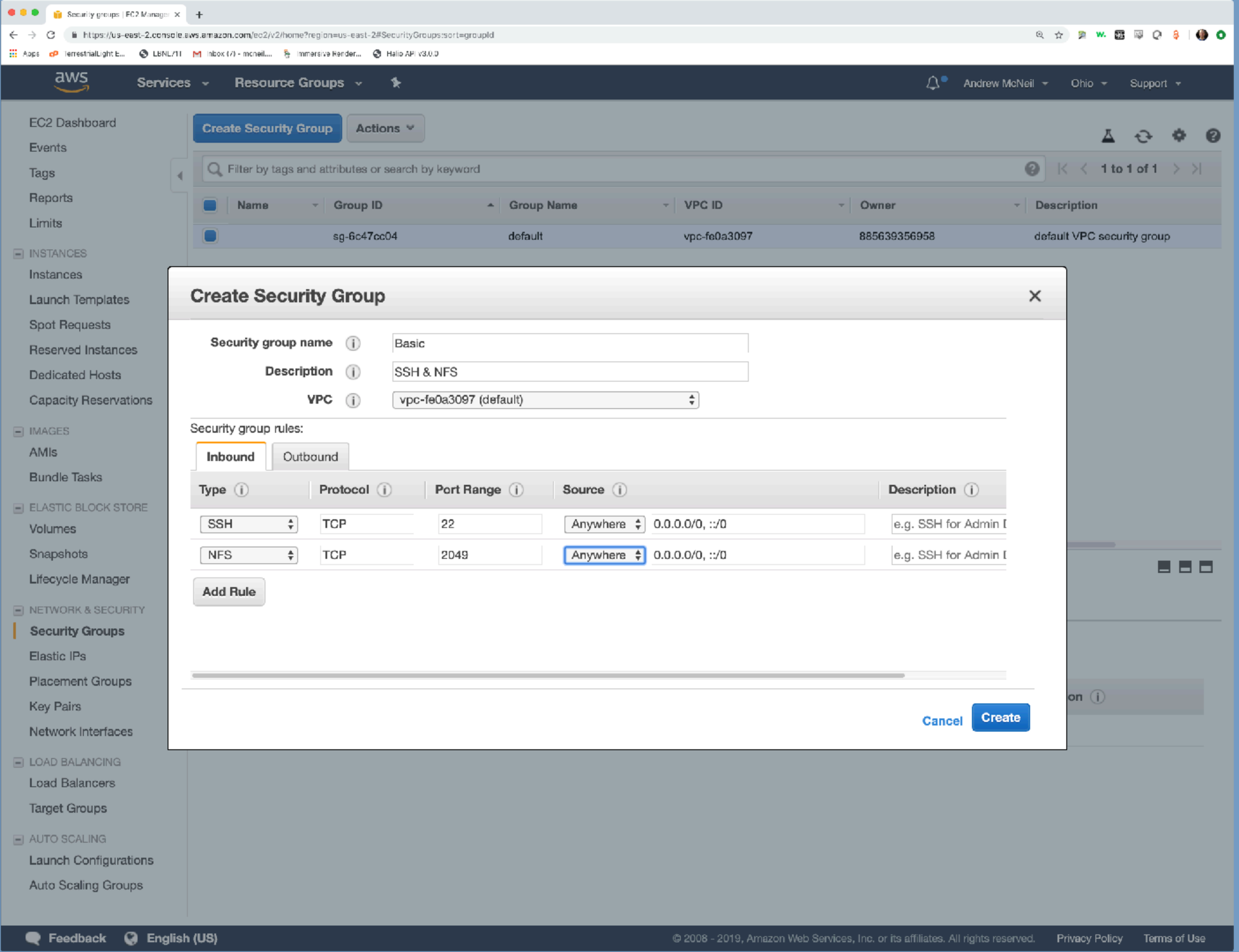
Security Groups

- Sets access rules
  - restrict to IP address or only within security group
- Set accessible ports
  - SSH - port 22
  - NFS - port 2049
  - HTTP - port 80, HTTPS port 443
- When you launch an instance, a temporary security group is created, however it’s simplest to use one that’s already created.
- Useful for connecting EFS with ECS instances (we’ll get to this later)



Your account contains a default security group that allows inbound connections from other instances in the security group and nowhere else.

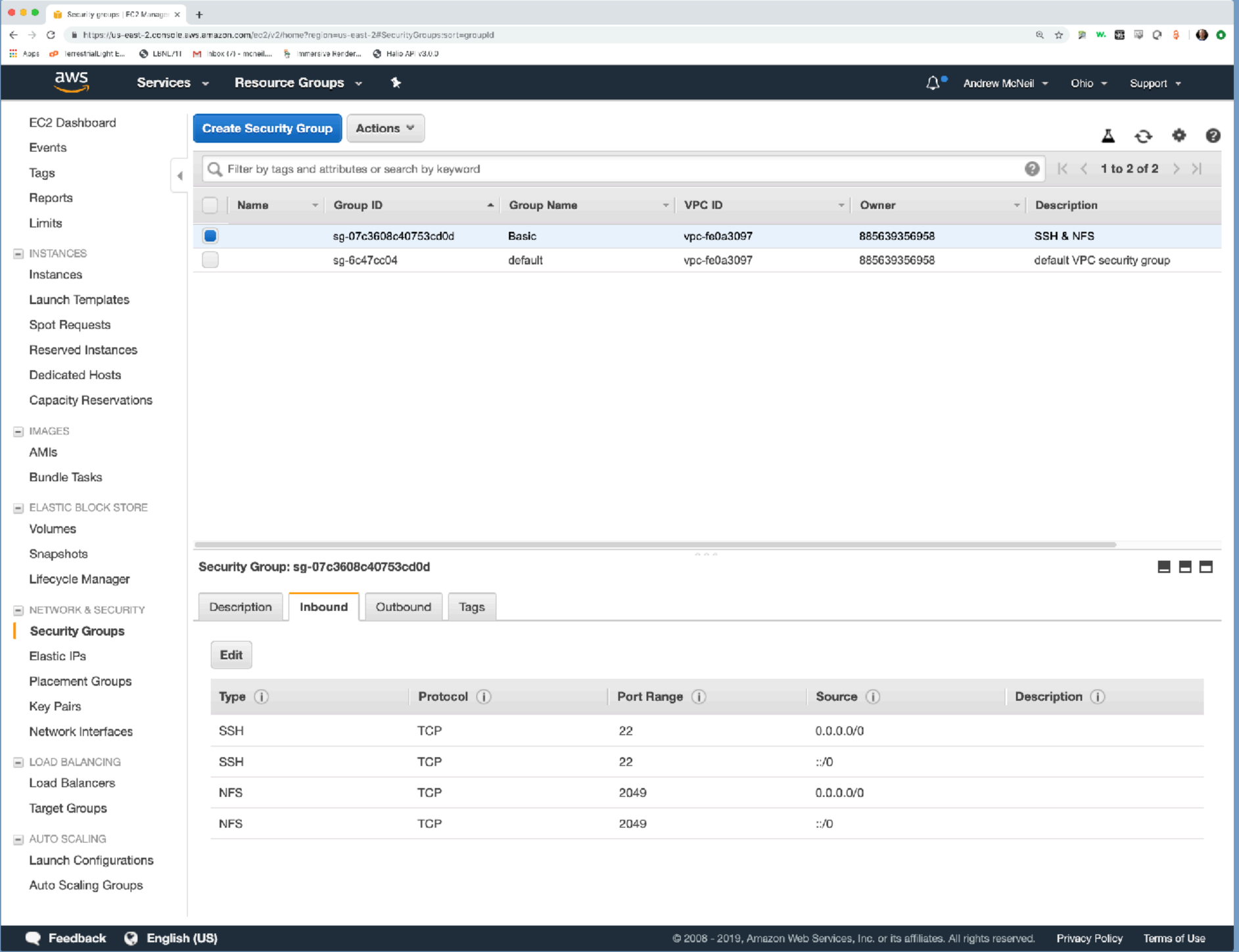
Click ‘Create Security Group’ to make a new security group.



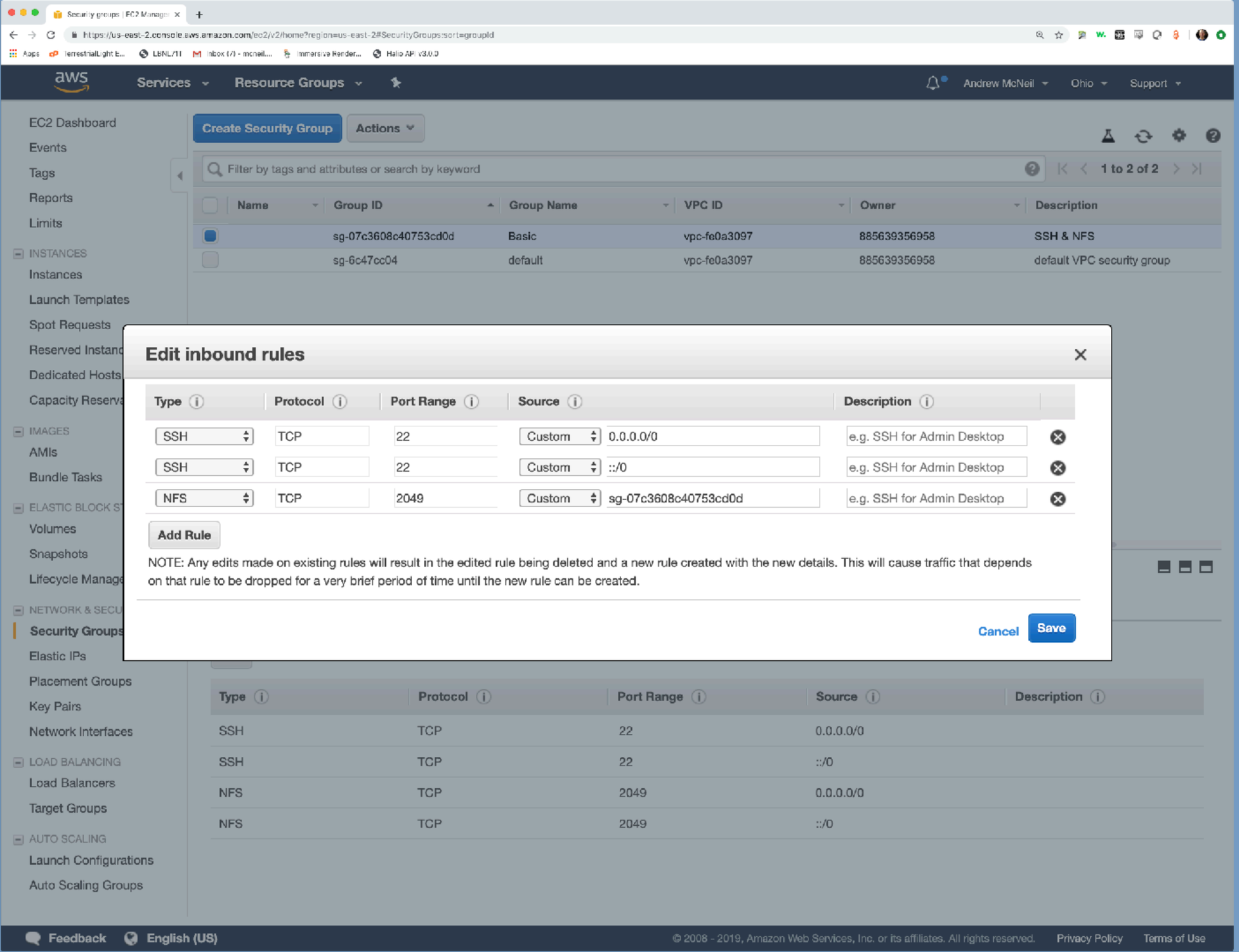
We'll create a security group with the following inbound access:

ssh from anywhere
(so you can connect to the instance from your computer)

nfs from anywhere
(we'll restrict nfs to this security group in the next step)

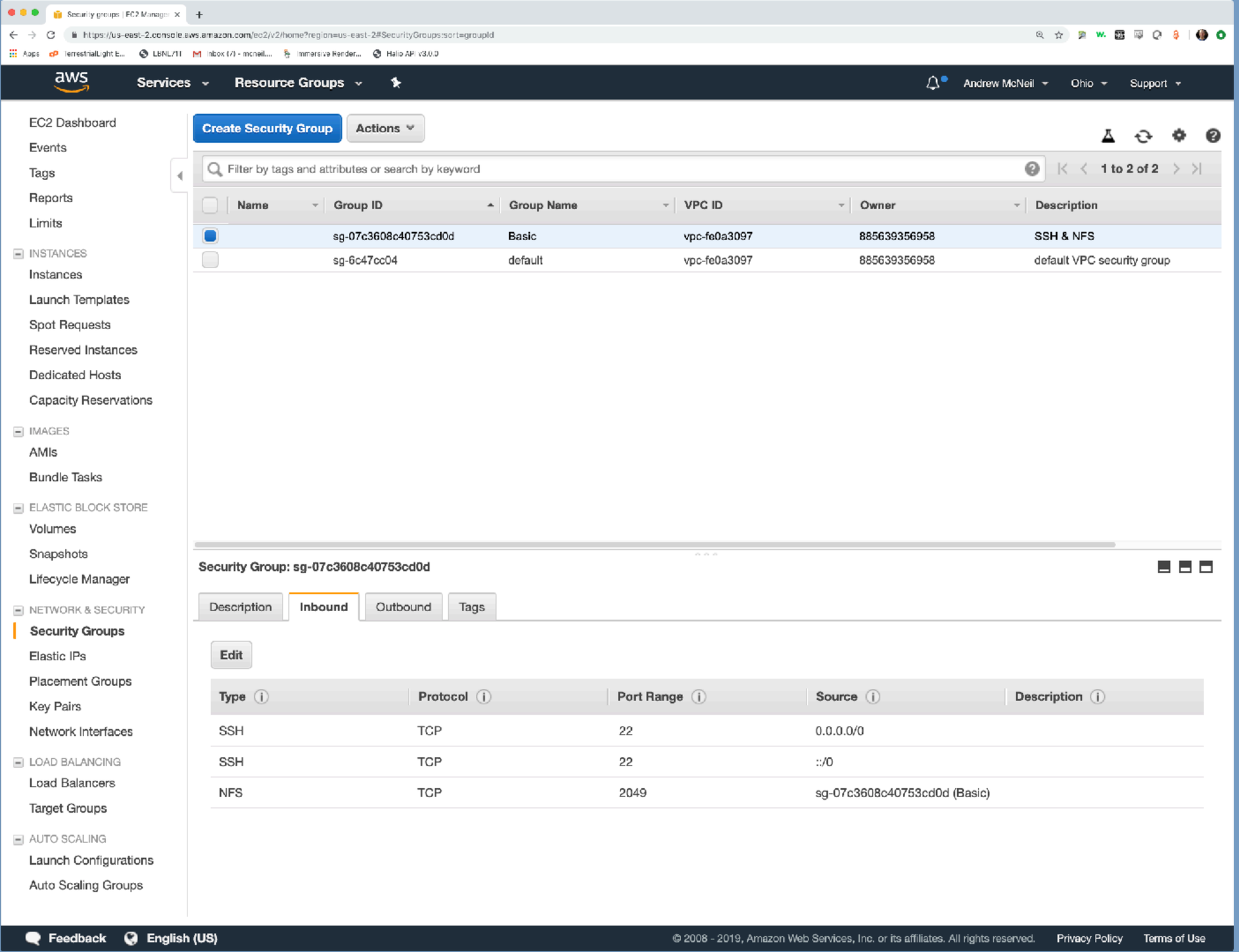


Now we have two security groups.



Edit the inbound rules for the new secruity group by adding the ID of the security group in the source field for NFS.

This will restrict inbound NFS to only services in this security group.



Now we have a custom security group.

Instance Types

- Dozens of machine instances available organized into categories:
  - General Purpose
    - M - balanced compute, memory, and storage
    - T - burstable
    - A - ARM
  - Compute Optimized
    - C: Lower cost per compute cycle
  - Storage Optimized
    - I: large SSD local storage
    - D: Very large HDD local storage (up to 48 TB)
    - H: Large HDD local storage (up to 16TB) and balanced compute and memory
  - Memory Optimized
    - R: more RAM per CPU
    - X: optimized for in memory database applications - lowest cost per GiB of RAM
    - Z: more RAM per CPU with highest CPU clock speed (4.0 GHz)
  - Accelerated Computing (GPU FGPA)
    - P: General Purpose GPU
    - G: Graphics intensive GPU
    - F: FGPAs

Information about instance types

- Amazon’s info: https://aws.amazon.com/ec2/instance-types/
- This 3rd-party website is sortable, filterable and includes prices: https://www.ec2instances.info

Three ways to purchase an instance

- On-demand - regular
- Reserved - pay by the year
- Spot - pseudo auction

On-Demand Instance

- You start and stop the instance.
- You are billed per hour at fixed rate, only for the time you use.
- Nothing can stop the instance except you.

Reserved Instance

- Discounts on long term reservations. The instance is yours for the duration of the term wether you use it or not.
- 38% discount for a 1-year reservation paid monthly
- 72% discount for a 3-year reservation paid in advance

- You probably don’t want to use reserved instances

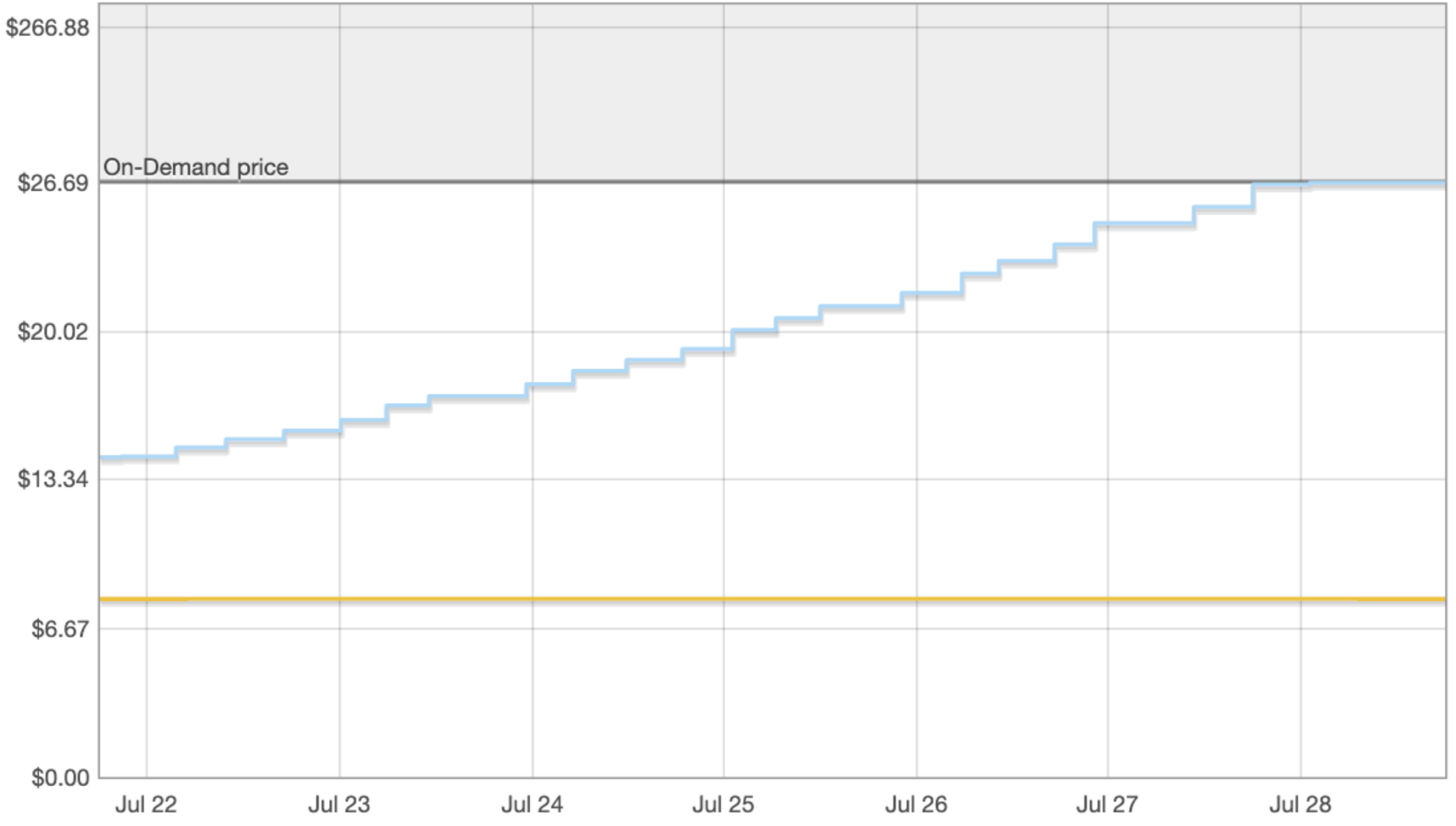
SPOT Instance

- You can get discounted rates on spare capacity (up to 90% discount).
- You bid the most you're willing to pay per hour.
- You pay the market clearing rate each hour (not your bid rate).
- If the market clearing rate exceeds your bid amount your instance is terminated without warning.

Spot Instance Pricing History

x

Product: Linux/UNIX Instance type: x1e.32xlarge Date range: 1 week



Date

7/28/2019
3:45:04 PM UTC-0700

On-Demand price

$26.6880

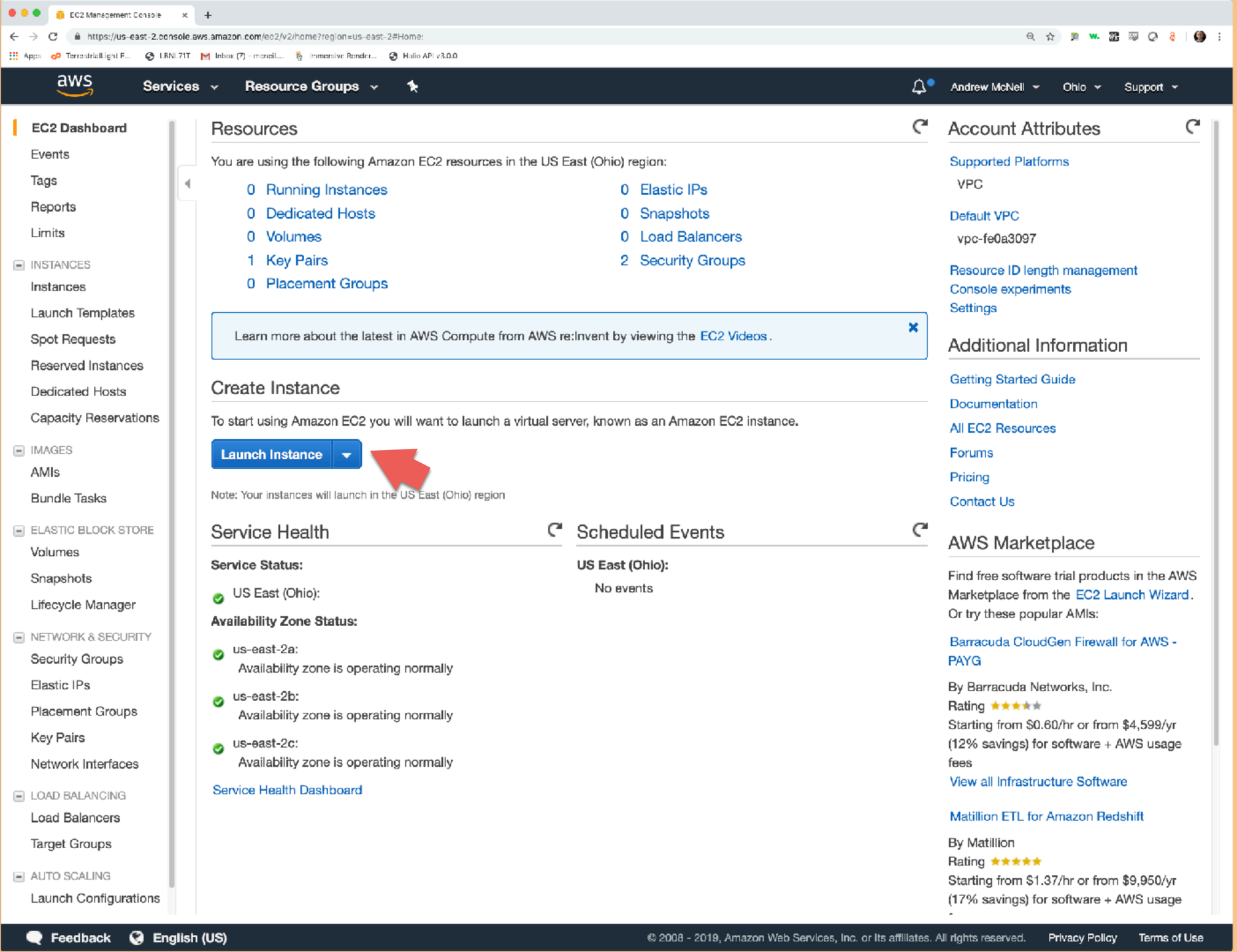
Availability Zone	Price
us-east-2a	$8.0064
us-east-2b	$26.6880

Close

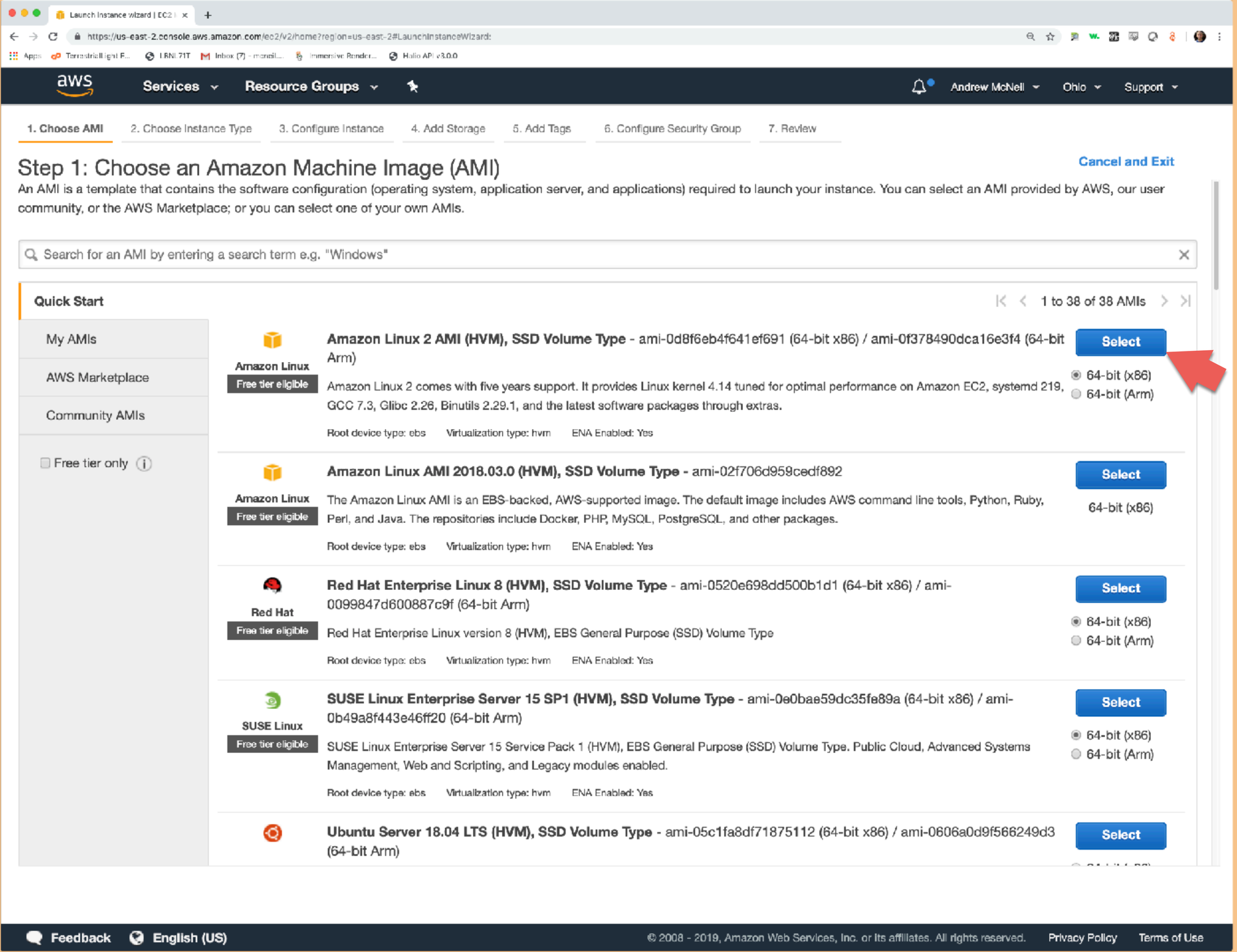
shhhh...

My spot bidding strategy:

Bid a few cents more than the on-demand price.



Let's launch an instance!



I've always used Amazon Linux. I don't have a good reason.

Then you select the type of instance.

Make sure you only use the free tier eligible instance if you don't want to pay.

Launch Instance wizard | EC2 | ×

https://us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#LaunchInstanceWizard:

aws Services Resource Groups Andrew McNeil Ohio Support

1. Choose AMI | 2. Choose Instance Type | 3. Configure Instance | 4. Add Storage | 5. Add Tags | 6. Configure Security Group | 7. Review

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.

Filter by: All instance types | Current generation | Show/Hide Columns

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
☐	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
☑	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
☐	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
☐	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
☐	General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes
☐	General purpose	t2.xlarge	4	16	EBS only	-	Moderate	Yes
☐	General purpose	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
☐	General purpose	t3a.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes
☐	General purpose	t3a.micro	2	1	EBS only	Yes	Up to 5 Gigabit	Yes
☐	General purpose	t3a.small	2	2	EBS only	Yes	Up to 5 Gigabit	Yes
☐	General purpose	t3a.medium	2	4	EBS only	Yes	Up to 5 Gigabit	Yes
☐	General purpose	t3a.large	2	8	EBS only	Yes	Up to 5 Gigabit	Yes

Cancel | Previous | Review and Launch | Next: Configure Instance Details

Feedback | English (US)

 Privacy Policy | Terms of Use

Nothing to do here yet.

But we'll do some things here later.

Launch Instance wizard | EC2 | ×

https://us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#LaunchInstanceWizard:

aws Services Resource Groups Andrew McNeil Ohio Support

1. Choose AMI | 2. Choose Instance Type | 3. Configure Instance | 4. Add Storage | 5. Add Tags | 6. Configure Security Group | 7. Review

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances	1 Launch into Auto Scaling Group
Purchasing option	Request Spot instances
Network	vpc-fe0a3097 (default) Create new VPC
Subnet	No preference (default subnet in any Availability Zon Create new subnet
Auto-assign Public IP	Use subnet setting (Enable)
Placement group	Add instance to placement group
Capacity Reservation	Open Create new Capacity Reservation
IAM role	None Create new IAM role
Shutdown behavior	Stop
Enable termination protection	Protect against accidental termination
Monitoring	Enable CloudWatch detailed monitoring Additional charges apply.
Tenancy	Shared - Run a shared hardware instance Additional charges will apply for dedicated tenancy.
Elastic Inference	Add an Elastic Inference accelerator Additional charges apply.
T2/T3 Unlimited	Enable Additional charges may apply

▸ Advanced Details

Cancel | Previous | Review and Launch | Next: Add Storage

Feedback English (US) © 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

This is where you add EBS storage, if you decide to use that.

Launch Instance wizard | EC2 | ×

https://us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#LaunchInstanceWizard:

Apps | Terrestrial light E... | LBNI 71T | Inbox (7) - mcnell... | Immersive Render... | Halio API v3.0.0

aws | Services | Resource Groups | Andrew McNeil | Ohio | Support

1. Choose AMI | 2. Choose Instance Type | 3. Configure Instance | 4. Add Storage | 5. Add Tags | 6. Configure Security Group | 7. Review

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. Learn more about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/xvda	snap-077085afe6b3ee68d	8	General Purpose SSD (gp2)	100 / 3000	N/A	☑	Not Encrypte

Add New Volume

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. Learn more about free usage tier eligibility and usage restrictions.

Cancel | Previous | Review and Launch | Next: Add Tags

Feedback | English (US)

 Privacy Policy | Terms of Use

Launch Instance wizard | EC2 | https://us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#LaunchInstanceWizard:

aws Services Resource Groups Andrew McNeil Ohio Support

1. Choose AMI | 2. Choose Instance Type | 3. Configure Instance | 4. Add Storage | 5. Add Tags | 6. Configure Security Group | 7. Review

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.
A copy of a tag can be applied to volumes, instances or both.
Tags will be applied to all instances and volumes. Learn more about tagging your Amazon EC2 resources.

Key (128 characters maximum)	Value (256 characters maximum)	Instances	Volumes
project	radiance workshop	☑	☑

Add another tag (Up to 50 tags maximum)

Cancel | Previous | Review and Launch | Next: Configure Security Group

Feedback English (US) © 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

If you want to track costs by project, add a tag here.

Also, you need to activate the tag in the cost management page (we'll do that towards the end)

We'll use the security group we created.

Launch instance wizard | EC2

https://us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#LaunchInstanceWizard:

Apps | TerrestrialLight E... | LBNL/TI | Inbox (7) - mcneil.... | Immersive Render... | Hallo API v3.0.0

aws | Services | Resource Groups | Andrew McNeil | Ohio | Support

1. Choose AMI | 2. Choose Instance Type | 3. Configure Instance | 4. Add Storage | 5. Add Tags | 6. Configure Security Group | 7. Review

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

Assign a security group: ○ Create a **new** security group

◉ Select an **existing** security group

Security Group ID	Name	Description	Actions
sg-07c3608c40753cd0d	Basic	SSH & NFS	Copy to new
sg-6c47cc04	default	default VPC security group	Copy to new

Inbound rules for sg-07c3608c40753cd0d (Selected security groups: sg-07c3608c40753cd0d)

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	0.0.0.0/0	
SSH	TCP	22	::/0	
NFS	TCP	2049	sg-07c3608c40753cd0d (Basic)	

Cancel | Previous | Review and Launch

Feedback | English (US)

 Privacy Policy | Terms of Use

Yes, the security group is open to the world, but they can't get in without your private key.

aws Services Resource Groups Andrew McNeil Ohio Support

1. Choose AMI | 2. Choose Instance Type | 3. Configure Instance | 4. Add Storage | 5. Add Tags | 6. Configure Security Group | 7. Review

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

Improve your instances' security. Your security group, Basic, is open to the world.

Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only.
You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. Edit security groups

AMI Details

Edit AMI

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0d8f6eb4f641ef691

Free tier eligible

Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras.

Root Device Type: ebs Virtualization type: hvm

Instance Type

Edit instance type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

Security Groups

Edit security groups

Security Group ID	Name	Description
sg-07c3608c40753cd0d	Basic	SSH & NFS

All selected security groups inbound rules

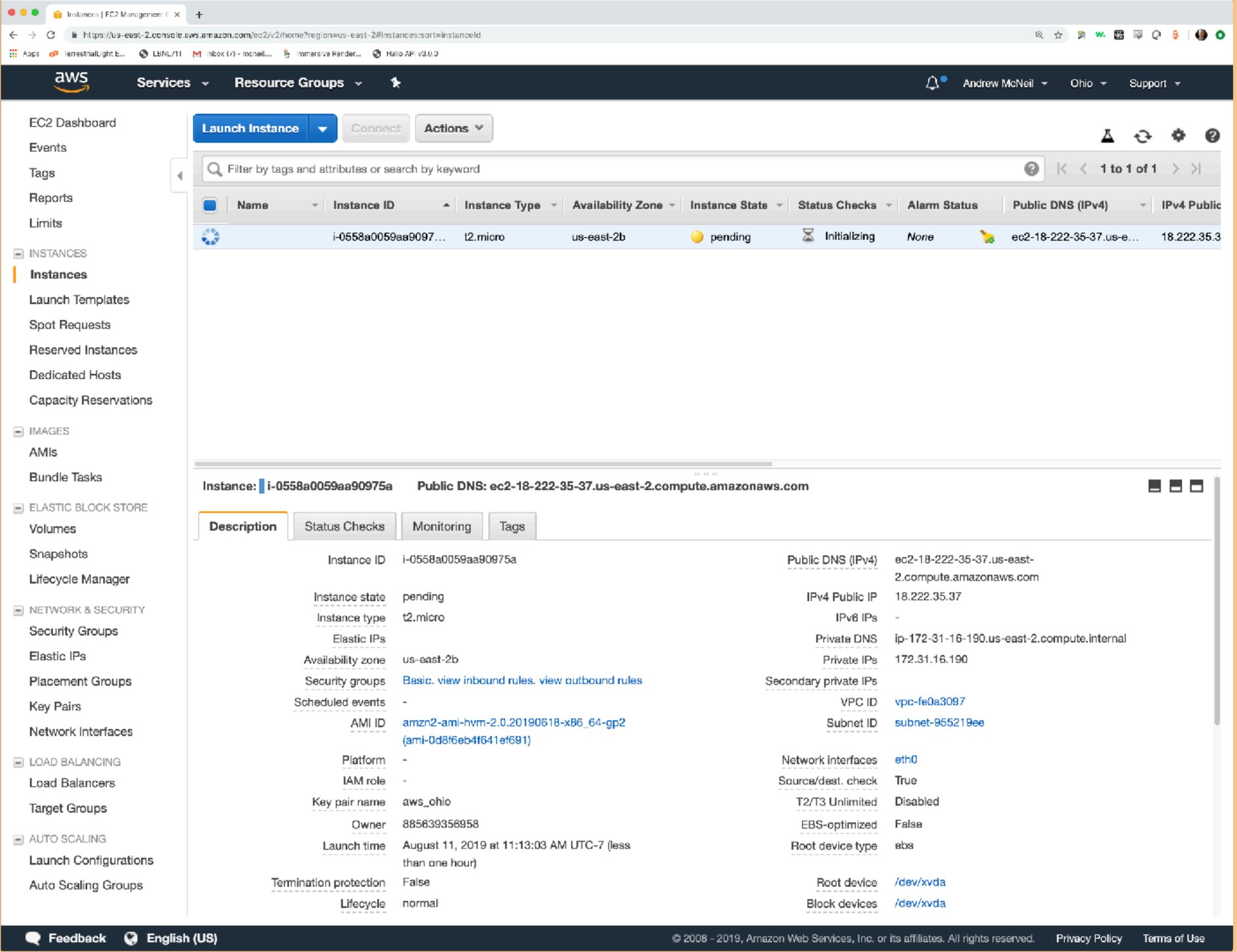
Type	Protocol	Port Range	Source	Description
SSH	TCP	22	0.0.0.0/0	
SSH	TCP	22	::/0	
NFS	TCP	2049	sg-07c3608c40753cd0d (Basic)	

Cancel Previous Launch

Feedback English (US)

 Privacy Policy Terms of Use

Pick the key you want to use, and then acknowledge that you have the key file.

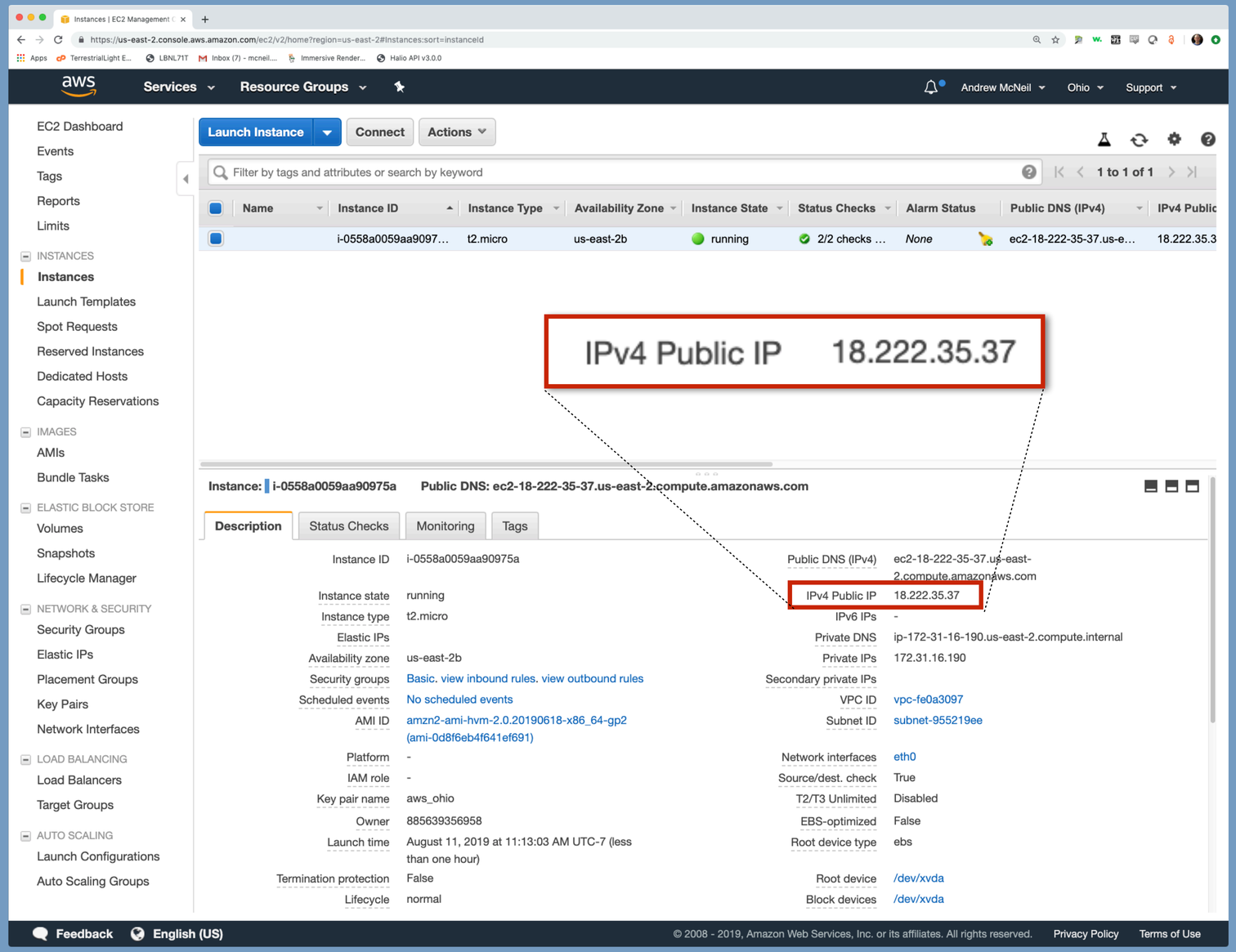


There it is, you did it!

You are clouding!

Connecting to your instance

- Mac & Linux: use ssh
- Windows: use putty or your favorite ssh client



Connecting on Mac / Linux

```
andy — -bash — 80×24
Last login: Sun Aug 11 11:39:28 on ttys008
amcn:~ andy$ ssh -i aws_ohio.pem ec2-user@18.222.35.37
```

ssh command from mac

-i to use your private key file

ec2-user is the username for amazon linux

use the public IP for your instance

Connecting on Mac / Linux

andy — ssh -i aws_ohio.pem ec2-user@18.222.35.37 — 80×24

```
Last login: Sun Aug 11 11:39:28 on ttys008
amcn:~ andy$ ssh -i aws_ohio.pem ec2-user@18.222.35.37
The authenticity of host '18.222.35.37 (18.222.35.37)' can't be established.
ECDSA key fingerprint is SHA256:pxosq3oYz0OIHPIERekfuZUE0kLG/KnlG4AbWRXevuM.
Are you sure you want to continue connecting (yes/no)?
```

yes you want to continue connecting.

Connecting on Mac / Linux

```
andy — -bash — 80×24
Last login: Sun Aug 11 11:39:28 on ttys008
amcn:~ andy$ ssh -i aws_ohio.pem ec2-user@18.222.35.37
The authenticity of host '18.222.35.37 (18.222.35.37)' can't be established.
ECDSA key fingerprint is SHA256:pxosq3oYz0OIHPIERekfuZUE0kLG/KnlG4AbWRXevuM.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '18.222.35.37' (ECDSA) to the list of known hosts.
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0644 for 'aws_ohio.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "aws_ohio.pem": bad permissions
ec2-user@18.222.35.37: Permission denied (publickey,gssapi-keyex,gssapi-with-mic
).
amcn:~ andy$
```

whoops!

you need tighter permissions for your private key.

Connecting on Mac / Linux

```
andy — -bash — 80×24
Last login: Sun Aug 11 11:39:28 on ttys008
amcn:~ andy$ ssh -i aws_ohio.pem ec2-user@18.222.35.37
The authenticity of host '18.222.35.37 (18.222.35.37)' can't be established.
ECDSA key fingerprint is SHA256:pxosq3oYz0OIHPIERekfuZUE0kLG/KnlG4AbWRXevuM.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '18.222.35.37' (ECDSA) to the list of known hosts.
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0644 for 'aws_ohio.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "aws_ohio.pem": bad permissions
ec2-user@18.222.35.37: Permission denied (publickey,gssapi-keyex,gssapi-with-mic
).
amcn:~ andy$ chmod 600 aws_ohio.pem
```

chmod 600 to prevent access from any other user.

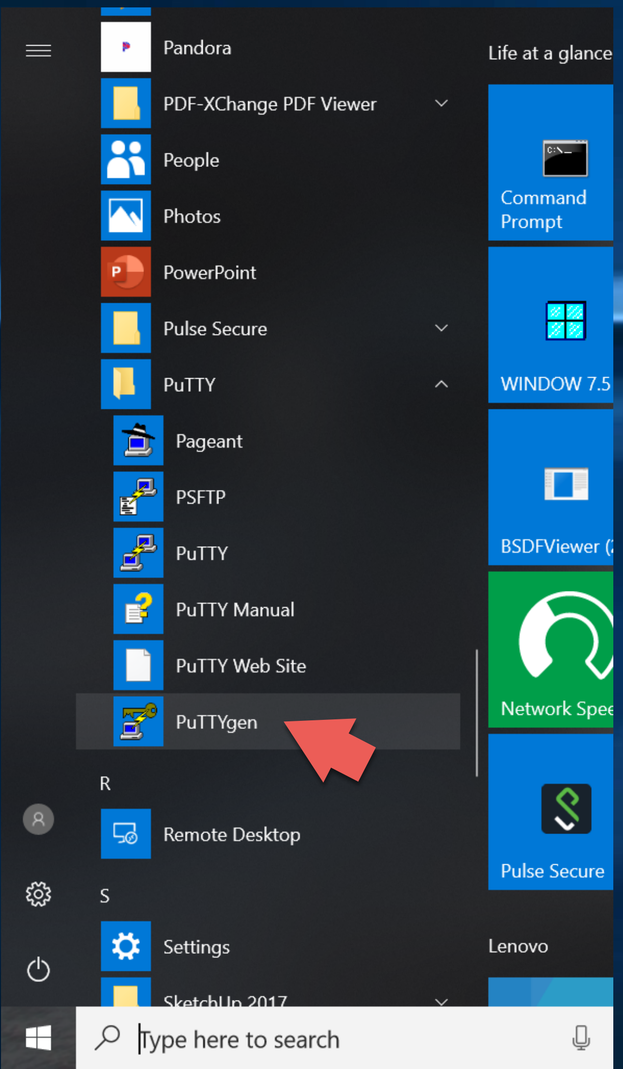
Connecting on Mac / Linux

andy — ec2-user@ip-172-31-16-190:~ — ssh -i aws_ohio.pem ec2-user@18.222.35.37 — 80×24

```
ECDSA key fingerprint is SHA256:pxosq3oYz0OIHPIERekfuZUE0kLG/KnlG4AbWRXevuM.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '18.222.35.37' (ECDSA) to the list of known hosts.
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0644 for 'aws_ohio.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "aws_ohio.pem": bad permissions
ec2-user@18.222.35.37: Permission denied (publickey,gssapi-keyex,gssapi-with-mic
).
amcn:~ andy$ chmod 600 aws_ohio.pem
amcn:~ andy$ ssh -i aws_ohio.pem ec2-user@18.222.35.37
Last login: Sun Aug 11 18:37:28 2019 from c-67-169-62-4.hsd1.ca.comcast.net

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
4 package(s) needed for security, out of 12 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-16-190 ~]$
```

then ssh again and viola!

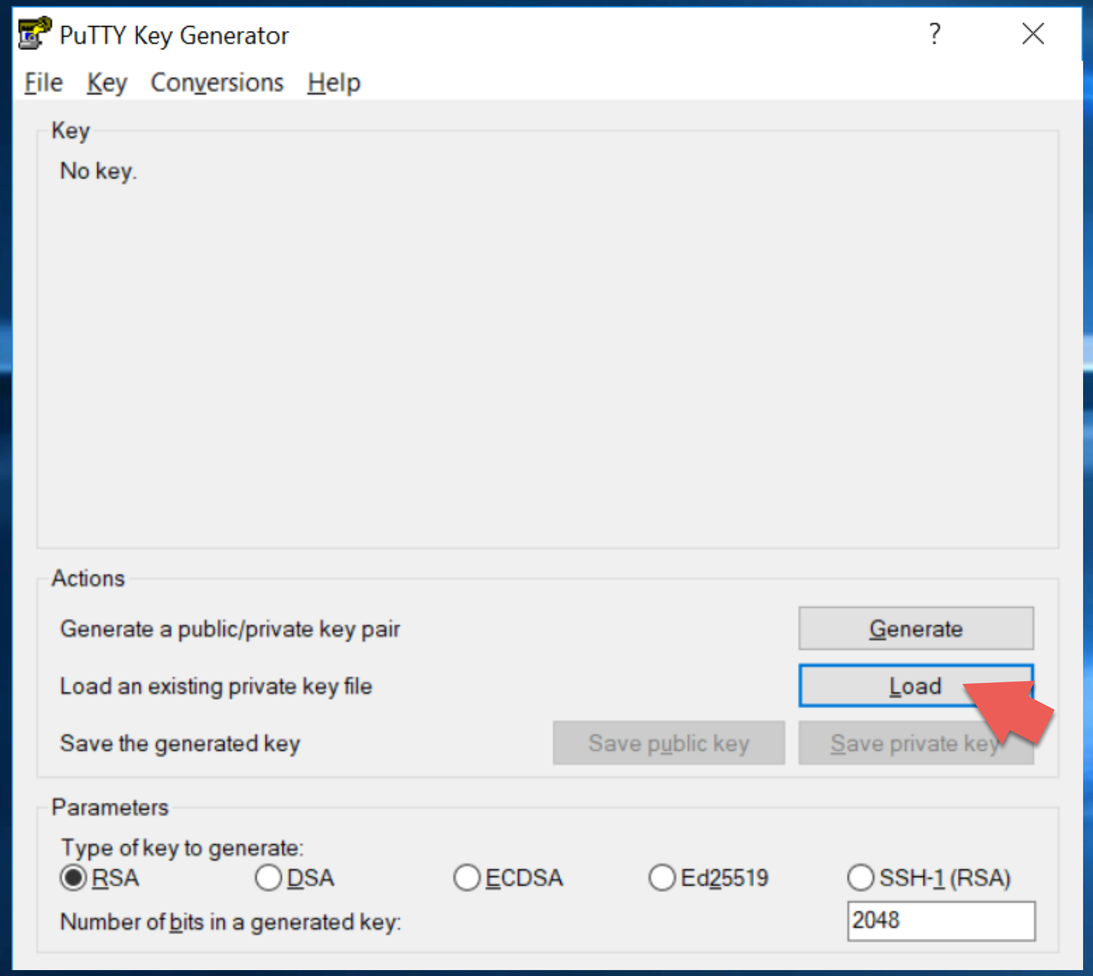
Connecting on Windows



First you need to convert your private key file to a PuTTY private key file.

Open PuTTYgen

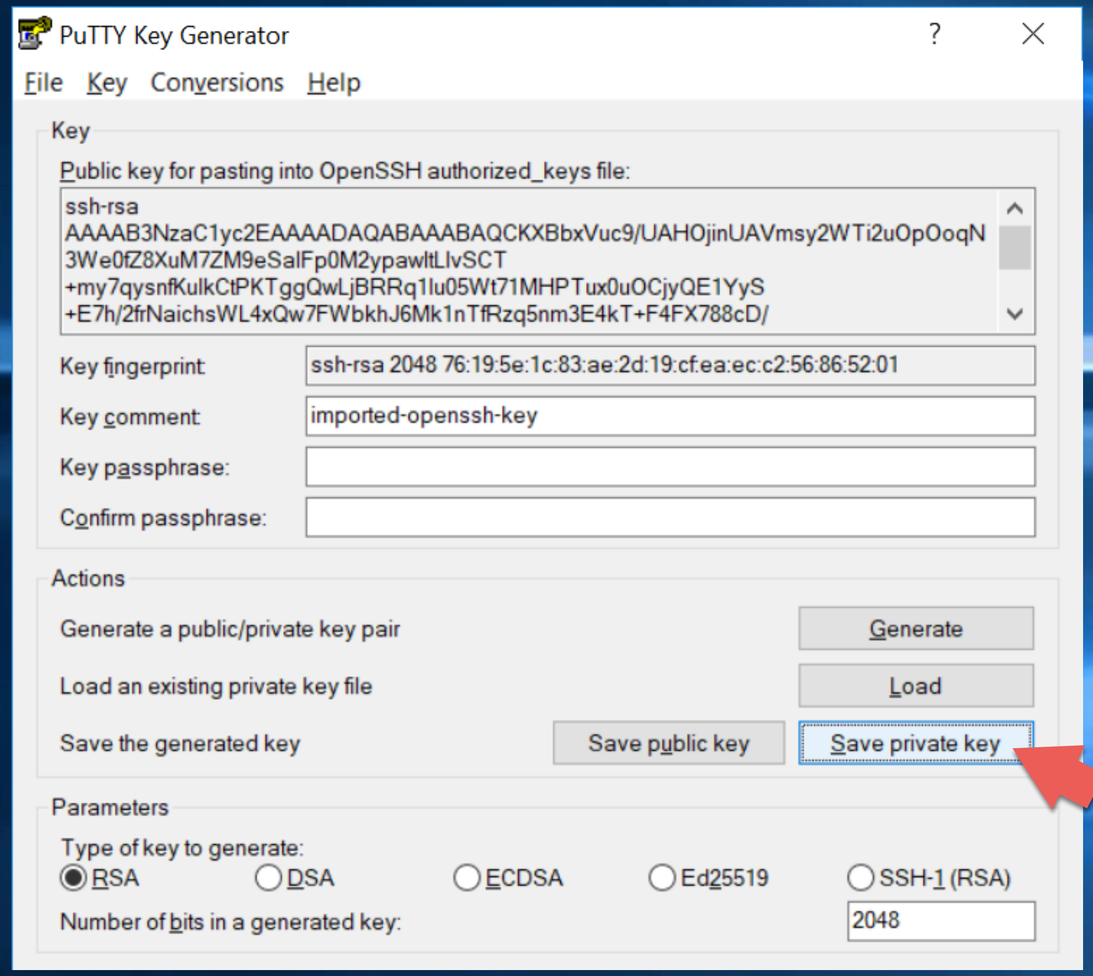
Connecting on Windows



Click “Load”

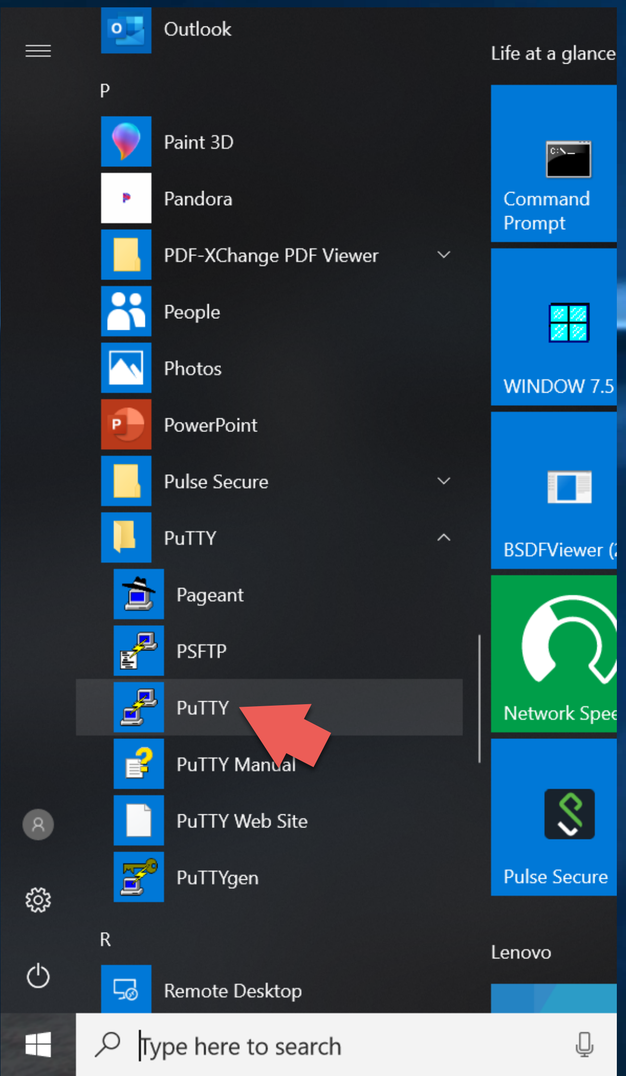
Browse to and select your pem file from amazon.

Connecting on Windows



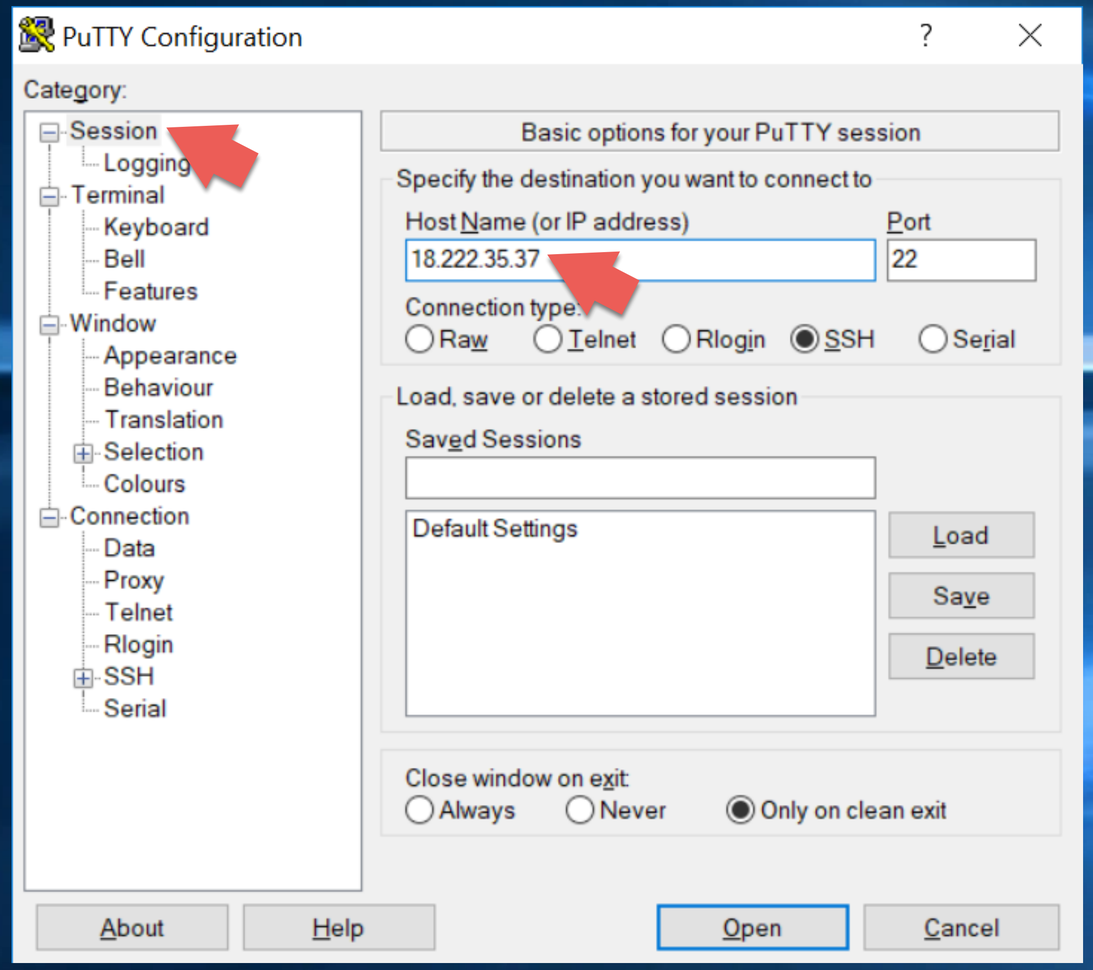
Click “Save private key”

Connecting on Windows



Next Launch PuTTY

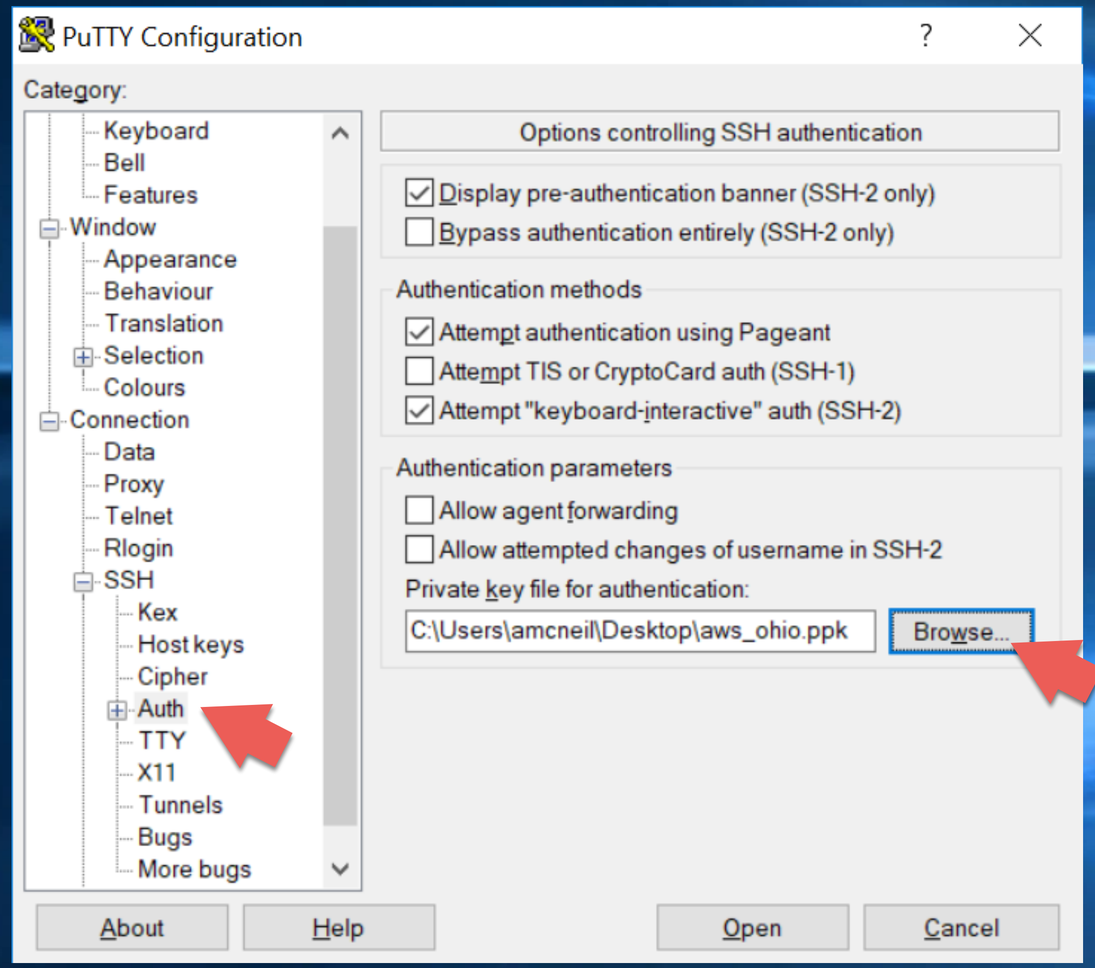
Connecting on Windows



Select Session, if not already selected

Enter the public IP address of your instance

Connecting on Windows

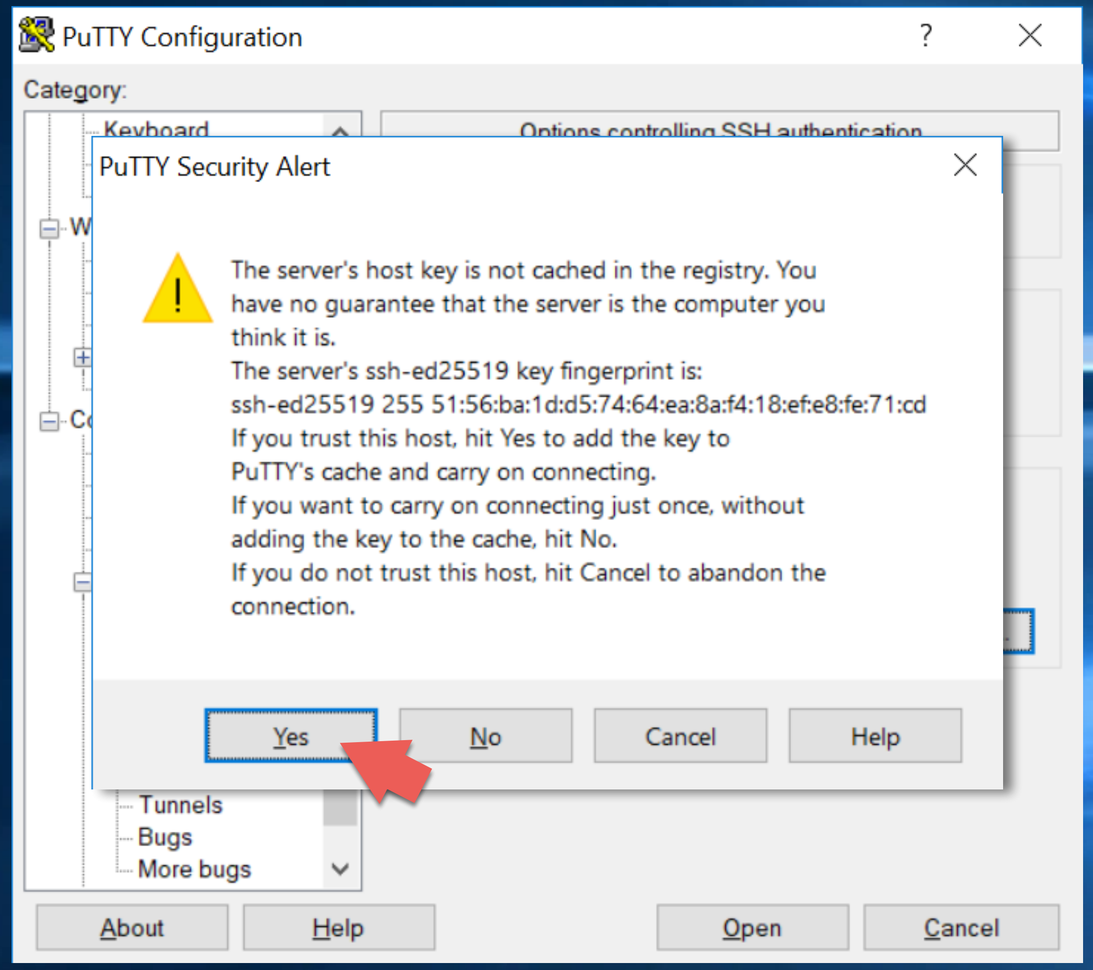


Select “Connection”
-> “SSH”
-> “Auth”

Click Browse, and select the private key ppk file.

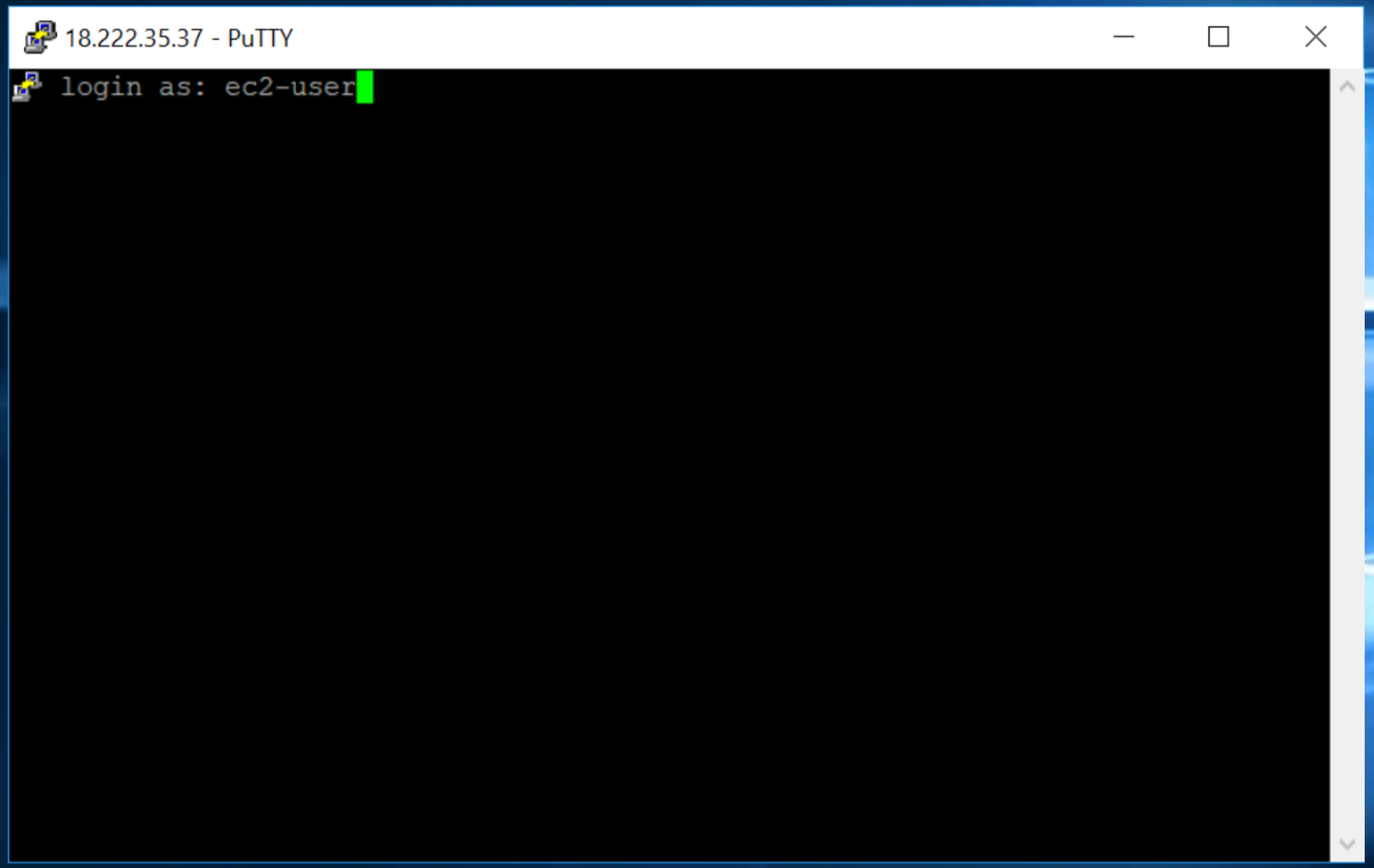
Then click “Open”

Connecting on Windows



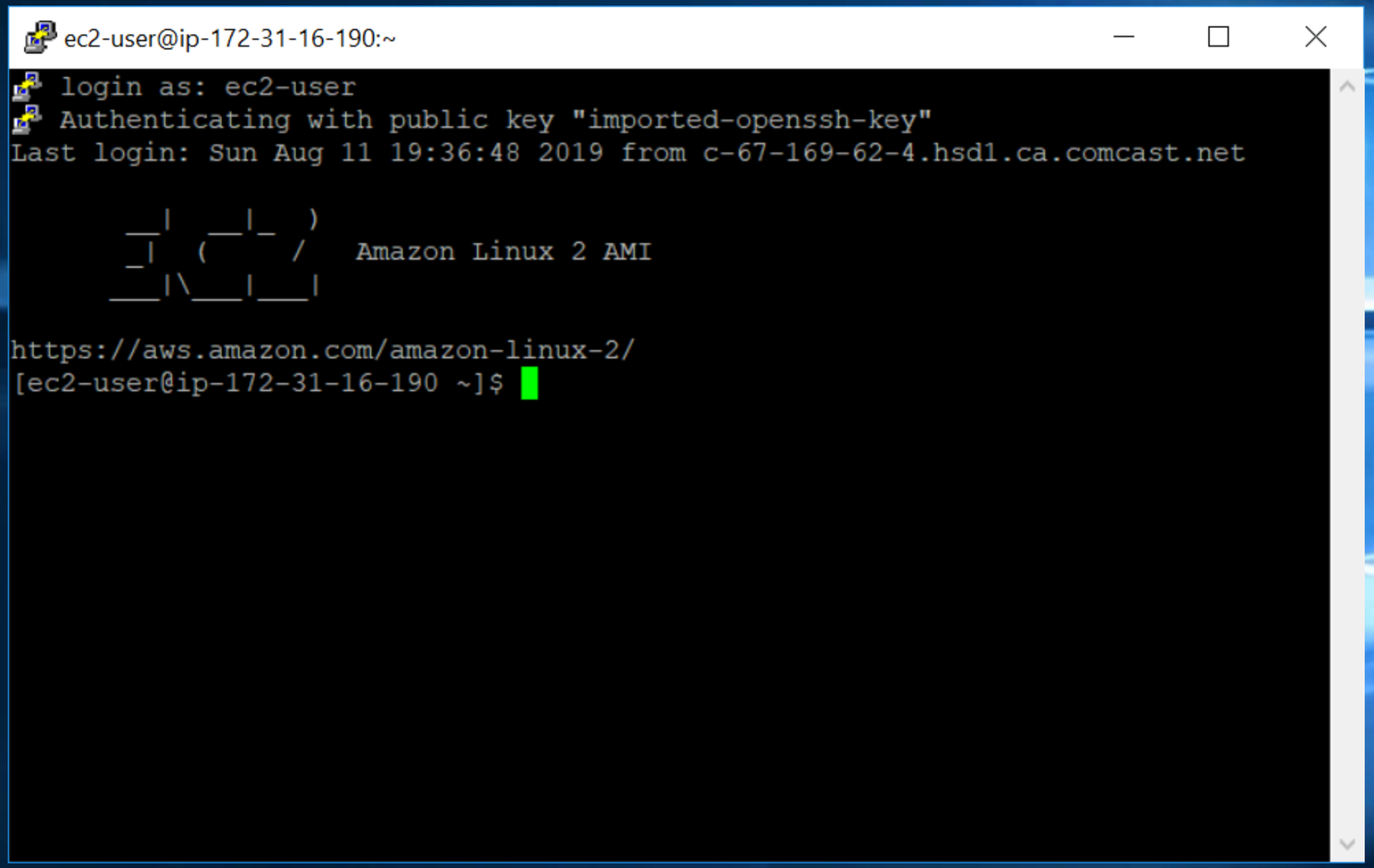
Yes, you want to carry on connecting.

Connecting on Windows



at the “login as:” prompt enter ec2-user

Connecting on Windows

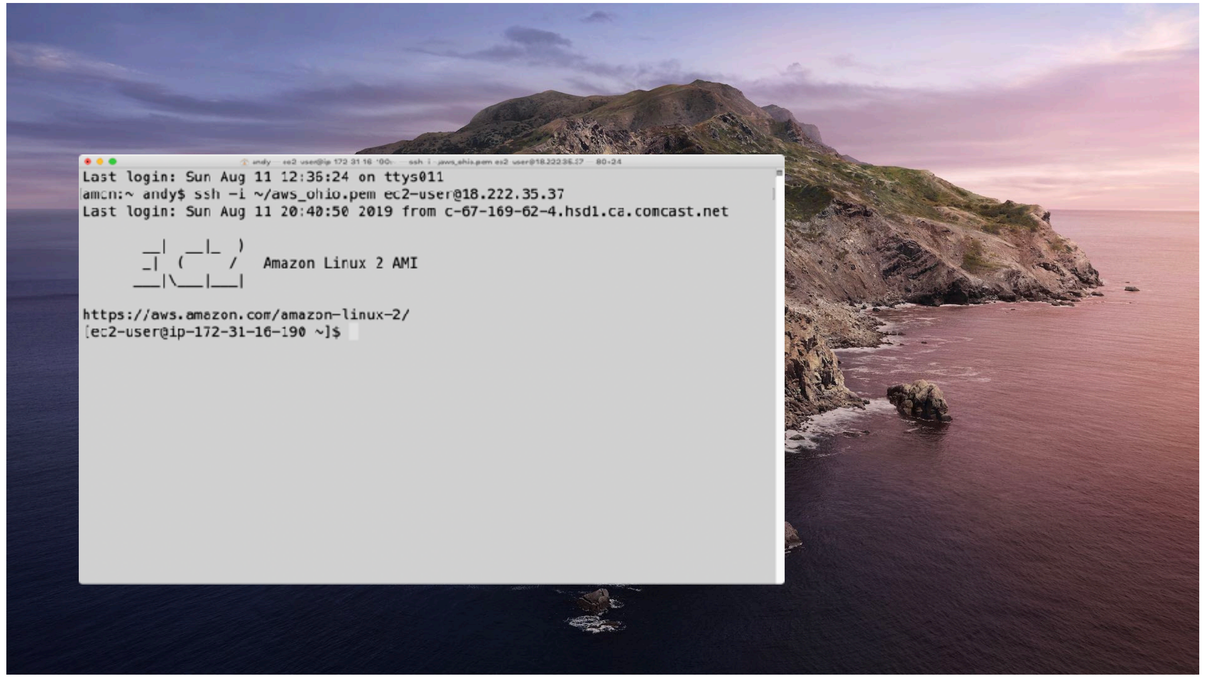


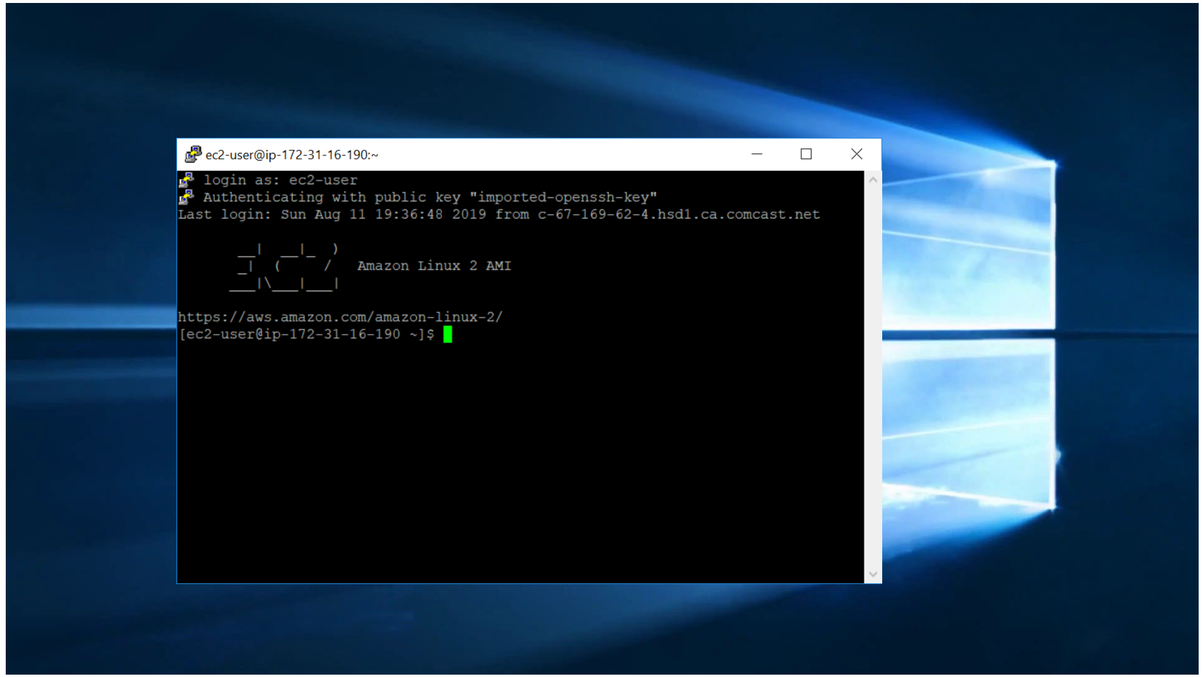
And you're in!

Look at that screen. Your family is going to think you're a nefarious hacker.

Regardess of OS, It’s the same from here…

- Bourne-Again Shell (bash)
- package manager: yum





Installing Software

- Amazon Linux uses yum package manager
  - First run yum update to get security and other updates
  - Then install packages needed to compile Radiance

```
sudo yum -y update
sudo yum -y install tcsh gcc gcc-c++ libX11-devel
```

Commands for installing RADIANCE

```
# create a directory for Radiance
mkdir Radiance
cd Radiance

# download Radiance HEAD from radiance-online.org
wget --no-check-certificate http://www.radiance-online.org/software/snapshots/radiance-HEAD.tgz
wget --no-check-certificate http://www.radiance-online.org/download-install/radiance-source-code/latest-release/radR52supp.tar.gz

# unpack tarballs
tar -xf radiance-HEAD.tgz
tar -xf radR52supp.tar.gz

# compile and install radiance
cd ray
sudo ./makeall install

# set raypath and copy cal files to raypath
echo RAYPATH=.:/usr/local/lib/ray/ > .bash_profile
echo export RAYPATH > .bash_profile
sudo cp src/cal/cal/* /usr/local/lib/ray/.

# check installation
rtrace -version
```

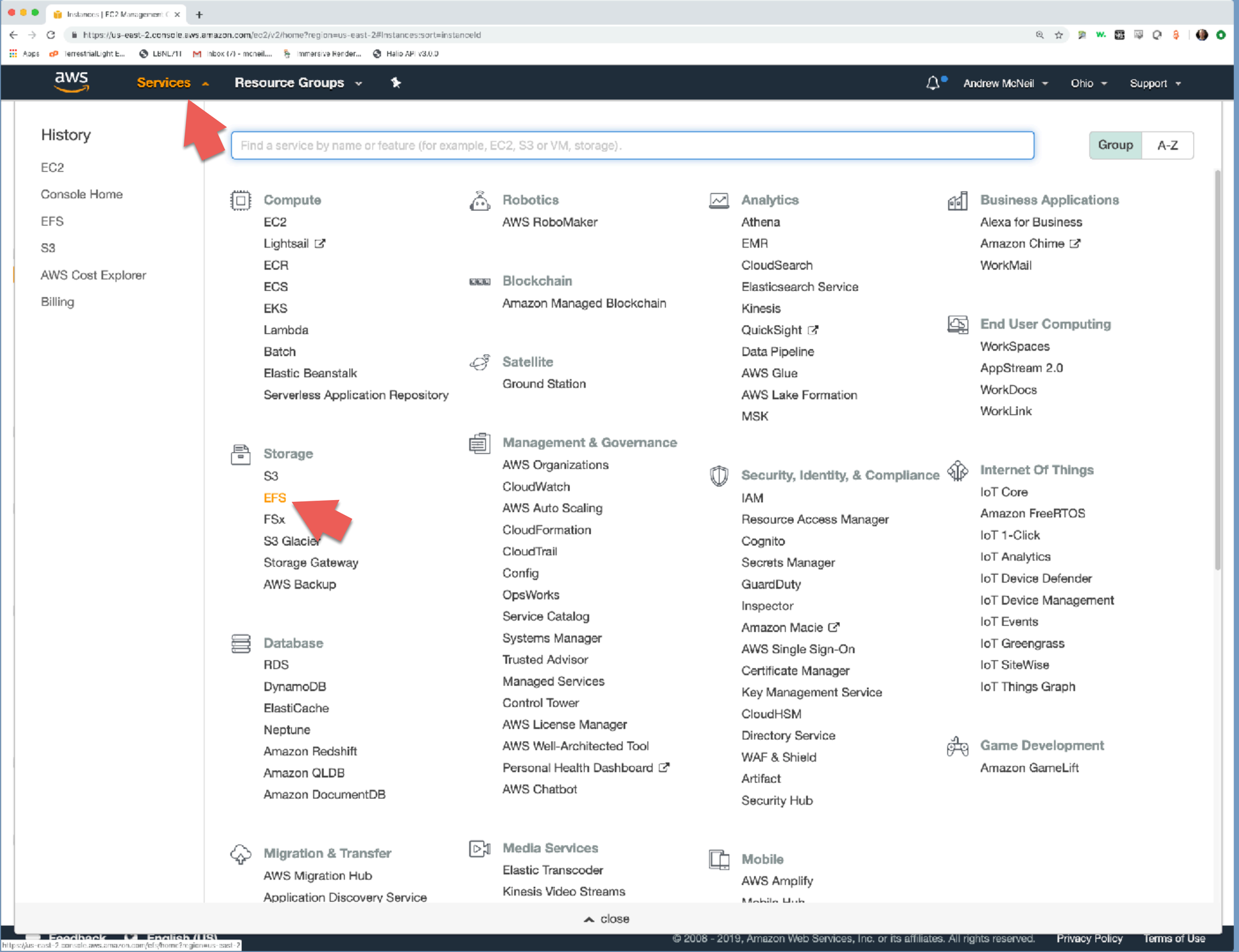
Run Mark Stock's Benchmark

```
# Install git
sudo yum -y install git

# clone benchmark repo
git clone https://github.com/markstock/Radiance-Benchmark4.git

# run the benchmark
export NCPU=16; make smp
```

File Storage Options

	price per GB	
Simple Storage Service (S3)	$0.02	• object storage in flat environment • good for archiving tarballs of old projects, but not for active storage
Elastic Block Store (EBS)	$0.10	• Can be attached to a single EC2 instance • Easy to attach when launching instance • Limited to one availability zone (most regions have three zones) • Size is fixed and set when provisioned (you pay for empty GB)
Elastic File System (EFS)	$0.30	• Can be simultaneously attached to many EC2 instances • Must be mounted like a network drive (at the command line or with a startup script) • Spans availability zones (but limited to region) • Size is elastic, you only pay for GB used by your data

Andy recommends EFS


Services
Resource Groups
Andrew McNeil
Ohio
Support
History
EC2
Console Home
EFS
S3
AWS Cost Explorer
Billing
Find a service by name or feature (for example, EC2, S3 or VM, storage).
Group
A-Z
Compute
EC2
Lightsail
ECR
ECS
EKS
Lambda
Batch
Elastic Beanstalk
Serverless Application Repository
Storage
S3
EFS
FSx
S3 Glacier
Storage Gateway
AWS Backup
Database
RDS
DynamoDB
ElastiCache
Neptune
Amazon Redshift
Amazon QLDB
Amazon DocumentDB
Migration & Transfer
AWS Migration Hub
Application Discovery Service
Robotics
AWS RoboMaker
Blockchain
Amazon Managed Blockchain
Satellite
Ground Station
Management & Governance
AWS Organizations
CloudWatch
AWS Auto Scaling
CloudFormation
CloudTrail
Config
OpsWorks
Service Catalog
Systems Manager
Trusted Advisor
Managed Services
Control Tower
AWS License Manager
AWS Well-Architected Tool
Personal Health Dashboard
AWS Chatbot
Media Services
Elastic Transcoder
Kinesis Video Streams
Analytics
Athena
EMR
CloudSearch
Elasticsearch Service
Kinesis
QuickSight
Data Pipeline
AWS Glue
AWS Lake Formation
MSK
Security, Identity, & Compliance
IAM
Resource Access Manager
Cognito
Secrets Manager
GuardDuty
Inspector
Amazon Macie
AWS Single Sign-On
Certificate Manager
Key Management Service
CloudHSM
Directory Service
WAF & Shield
Artifact
Security Hub
Mobile
AWS Amplify
Business Applications
Alexa for Business
Amazon Chime
WorkMail
End User Computing
WorkSpaces
AppStream 2.0
WorkDocs
WorkLink
Internet Of Things
IoT Core
Amazon FreeRTOS
IoT 1-Click
IoT Analytics
IoT Device Defender
IoT Device Management
IoT Events
IoT Greengrass
IoT SiteWise
IoT Things Graph
Game Development
Amazon GameLift
close
Feedback
English (US)
© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

Amazon Elastic File System (EFS)

Amazon EFS provides file storage for use with your EC2 instances.

Create file system

Getting started guide

Create

Create an Amazon EFS file system to store your files in the Amazon cloud. A file system grows and shrinks automatically with the files you put in, and you pay only for what you use.

Access

Write files to and read files from your Amazon EFS file system by using the NFSv4 protocol. Any number of EC2 instances can work with your file system at the same time, and your instances can be in multiple Availability Zones in a region.

Manage

You can easily administer your file system using the Amazon EFS console, CLI, and SDK.

Elastic File System documentation & support

Getting started guide | Documentation | Support | Forums

Create file system

Step 1: Configure file system access
Step 2: Configure optional settings
Step 3: Review and create

Configure file system access

An Amazon EFS file system is accessed by EC2 instances running inside one of your VPCs. Instances connect to a file system by using a network interface called a mount target. Each mount target has an IP address, which we assign automatically or you can specify.

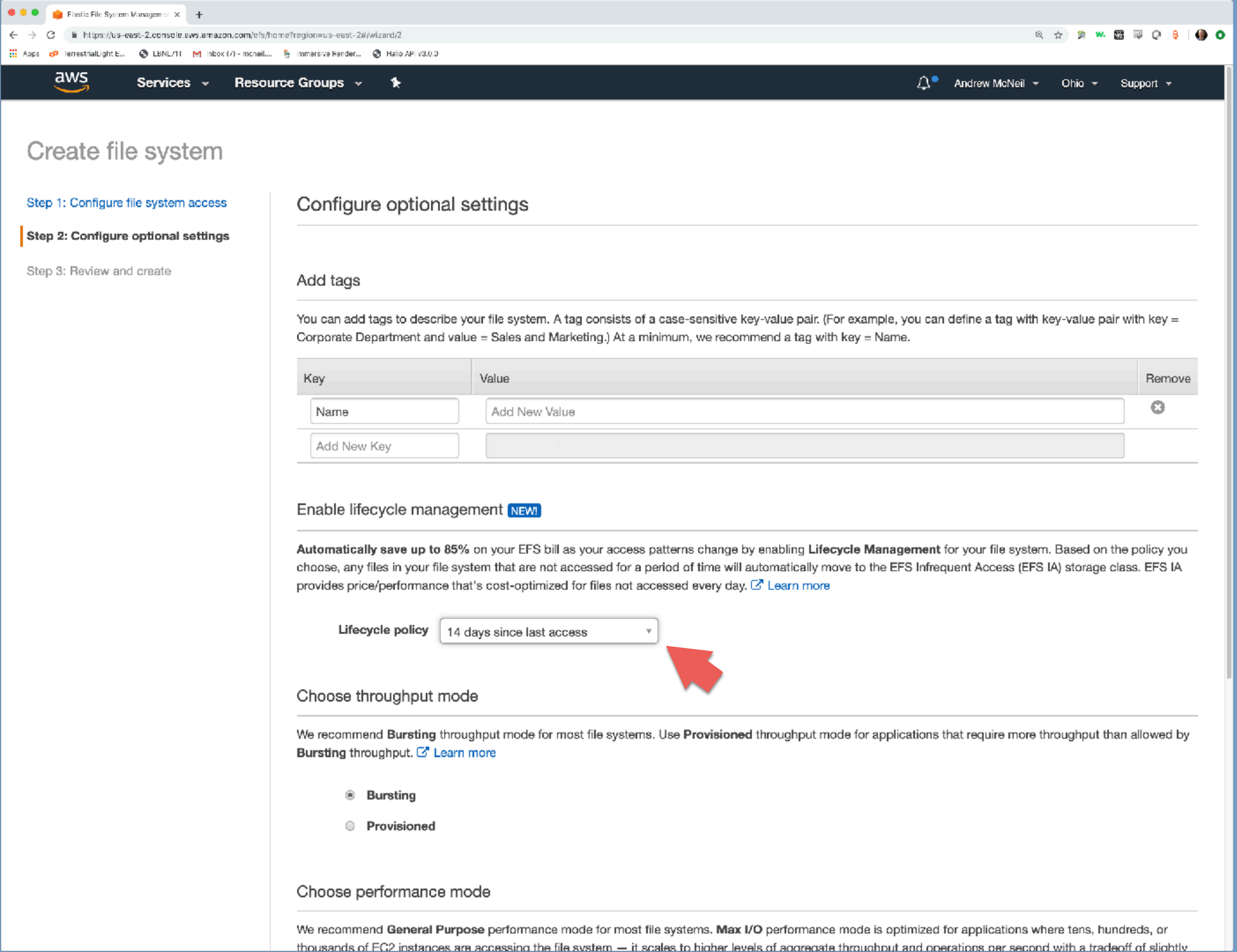
VPC: vpc-fe0a3097 (default)

Create mount targets

Instances connect to a file system by using mount targets you create. We recommend creating a mount target in each of your VPC's Availability Zones so that EC2 instances across your VPC can access the file system.

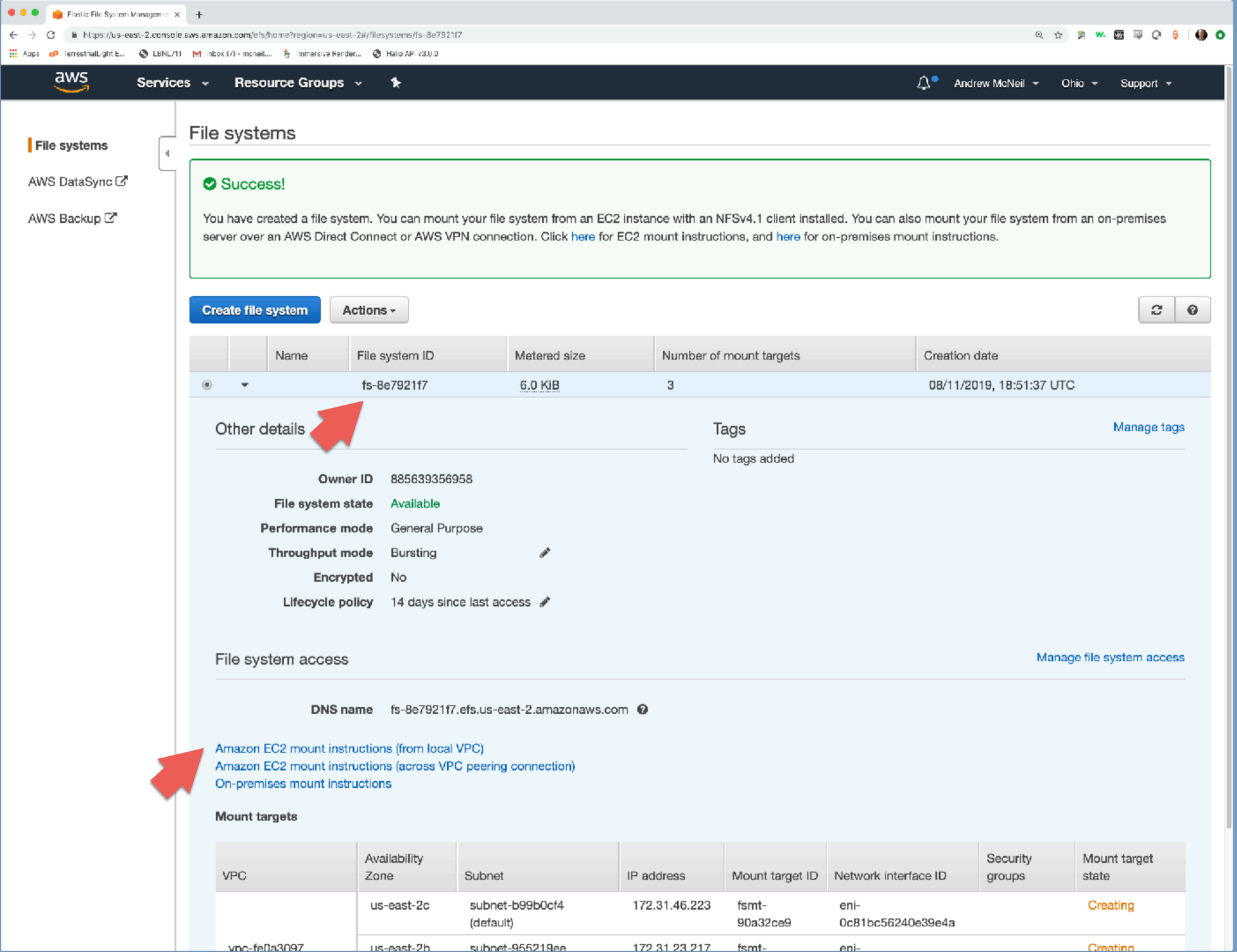
	Availability Zone	Subnet	IP address	Security groups
☑	us-east-2a	subnet-4b494322 (default)	Automatic	sg-07c3608c40753cd0d - Basic × sg-6c47cc04 - default ×
☑	us-east-2b	subnet-955219ee (default)	Automatic	sg-07c3608c40753cd0d - Basic × sg-6c47cc04 - default ×
☑	us-east-2c	subnet-b99b0cf4 (default)	Automatic	sg-07c3608c40753cd0d - Basic × sg-6c47cc04 - default ×

Cancel Next Step



Lifecycle policy automatically moves files not recently accessed to lower cost storage

The next time its accessed there will be latency of up to 100ms. but then it's put back into the faster more expensive storage.



Your file system id is used for mounting to instances.

There are mount instructions on the EFS page if you ever forget. You don't need to download this pdf again just for mounting instructions.

Mounting Elastic File Storage

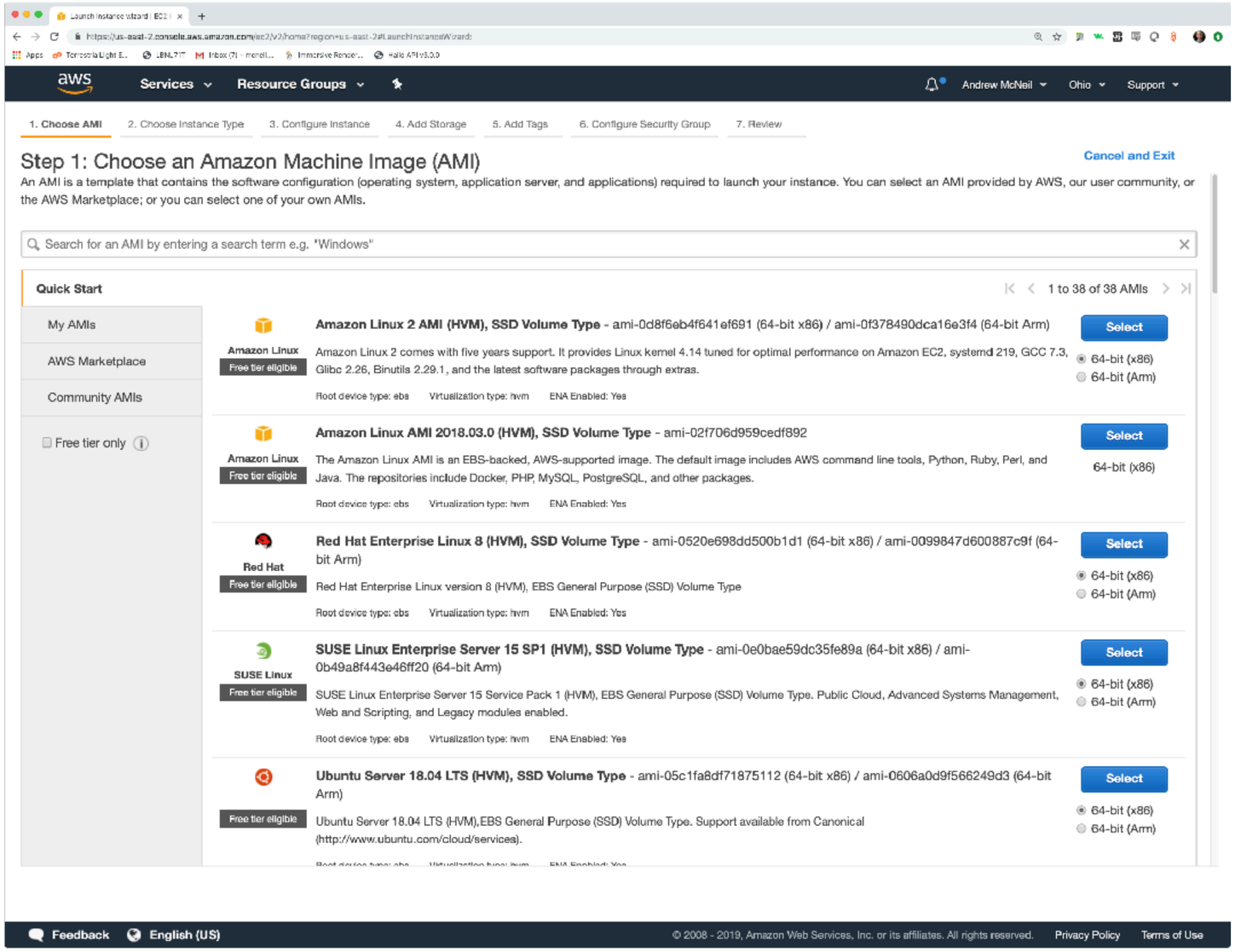
```
# Install tools
sudo yum install -y amazon-efs-utils

# create mount point
mkdir efs

# mount
sudo mount -t efs fs-8e7921f7:/ efs

# change owner and group if it's your first time mounting the file system
sudo chown ec2-user efs
sudo chgrp ec2-user efs
```

Saving a custom machine image

- It’d be nice to not have to do all this setup every time, right?
- Machine Images - You can save the state of the machine
- Configuration script - Allows you to run updates and mount EFS drives when the instance is started

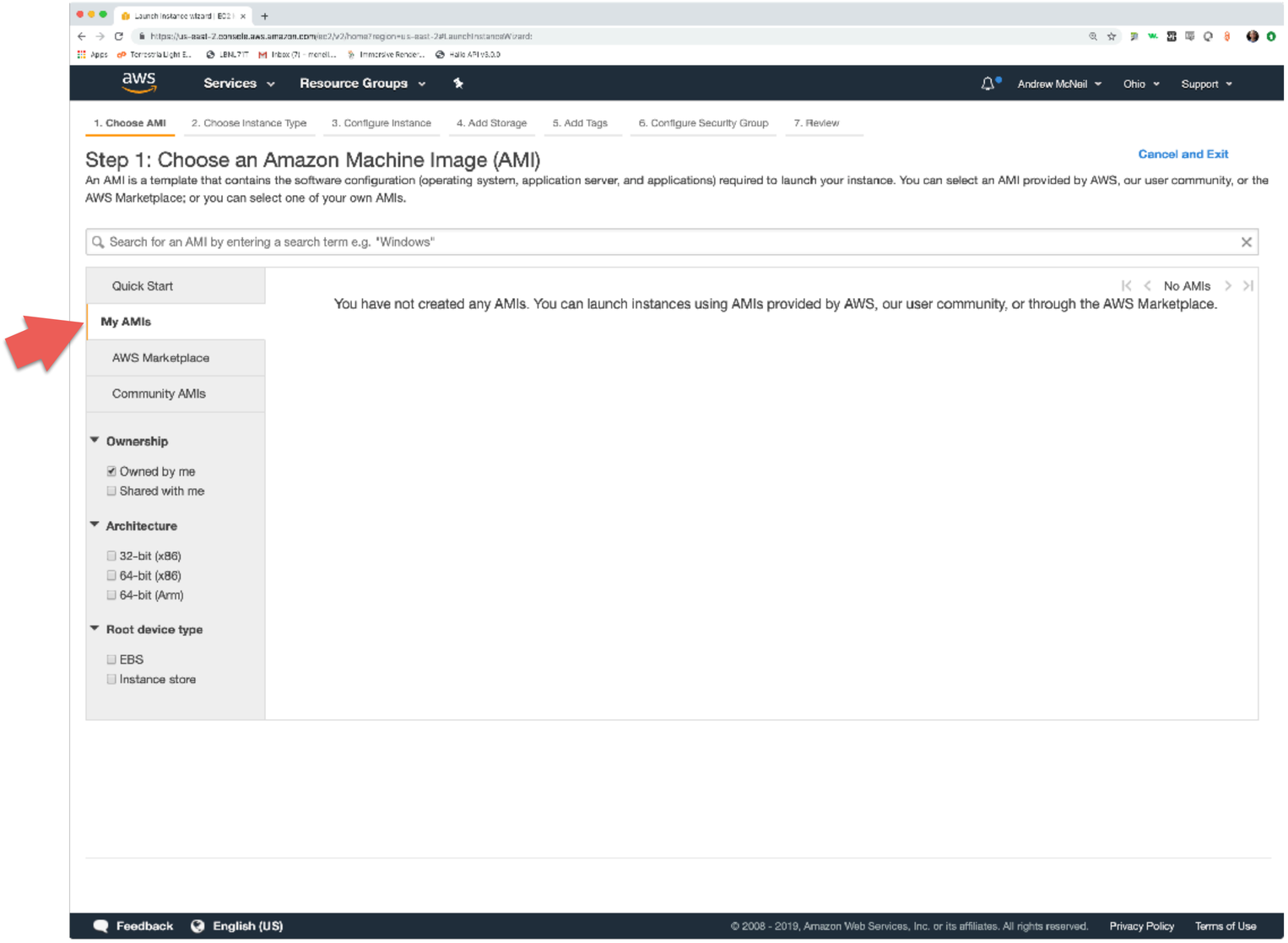
Machine Images

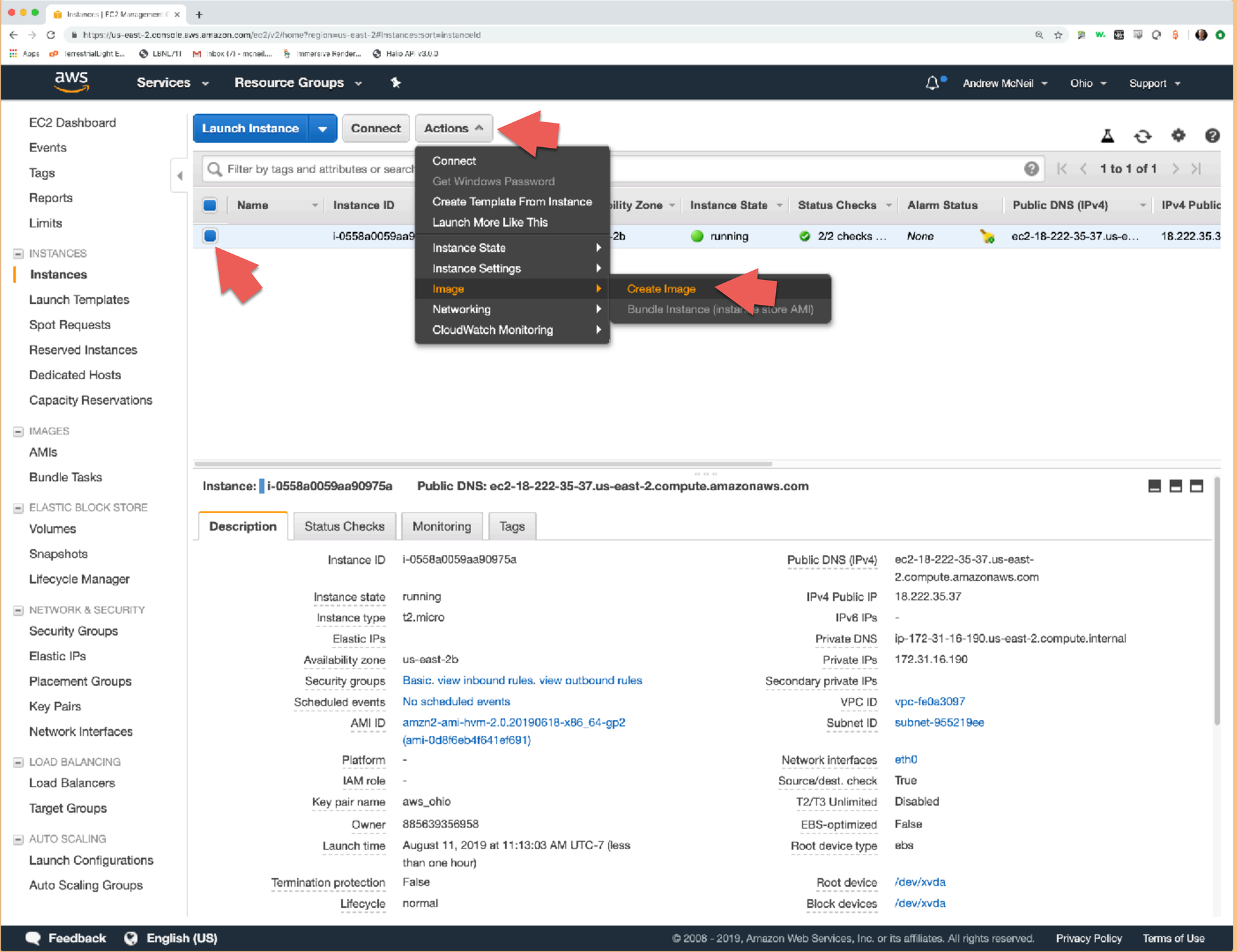
- Remember this step when launching an instance? ——————>



Machine Images

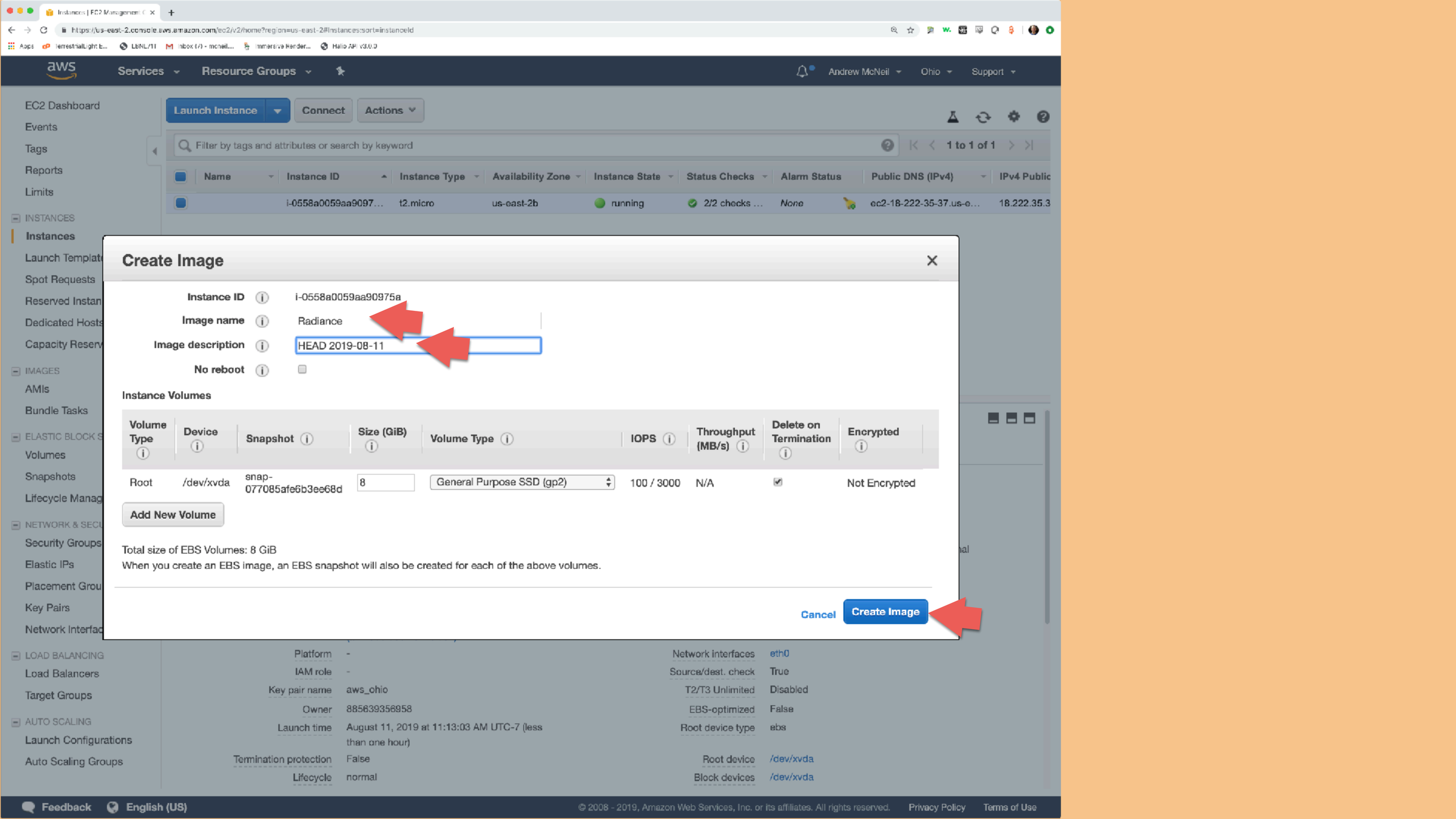
- Let’s create a machine image!

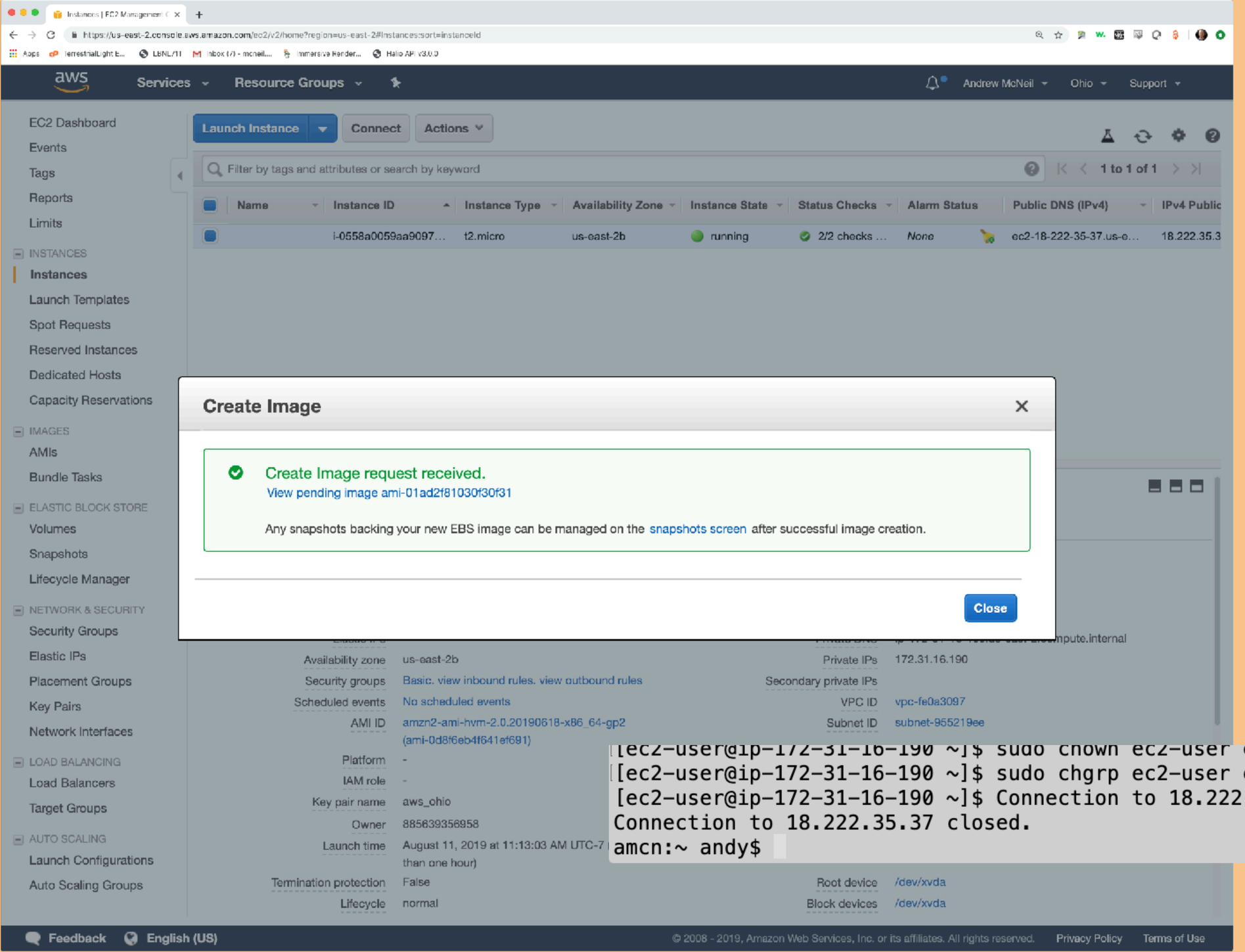




Select the instance from which you want to generate a machine image.

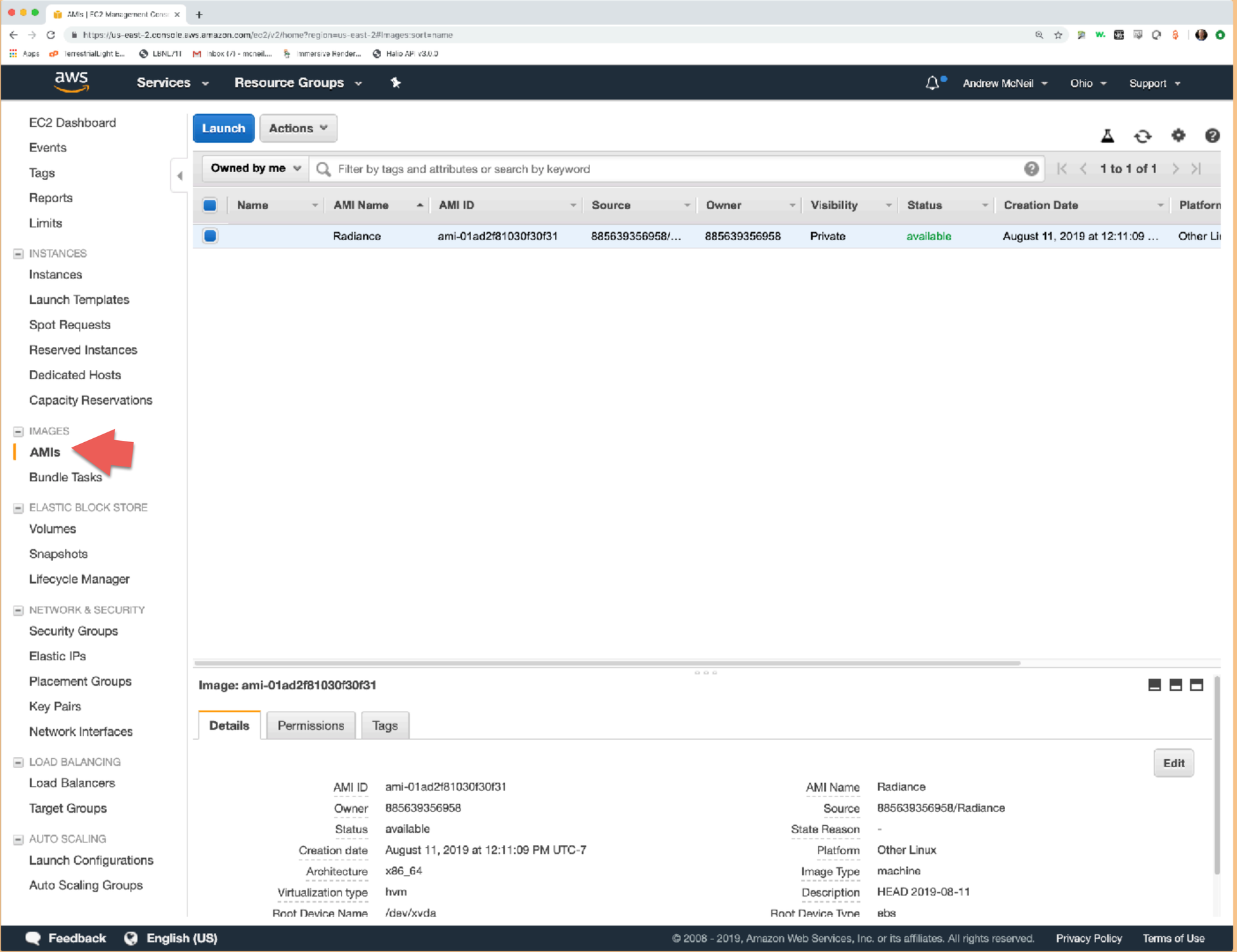
Make sure it's not busy, it will be rebooted to make the image!


Create Image
Instance ID
i-0558a0059aa90975a
Image name
Radiance
Image description
HEAD 2019-08-11
No reboot
Instance Volumes
Volume Type
Device
Snapshot
Size (GiB)
Volume Type
IOPS
Throughput (MB/s)
Delete on Termination
Encrypted
Root
/dev/xvda
snap-077085afe6b3ee68d
8
General Purpose SSD (gp2)
100 / 3000
N/A
Not Encrypted
Add New Volume
Total size of EBS Volumes: 8 GiB
When you create an EBS image, an EBS snapshot will also be created for each of the above volumes.
Cancel
Create Image



Select the instance from which you want to generate a machine image.

Make sure it's not busy, it will be rebooted to make the image!



There's our new image!

Configuration / Startup script

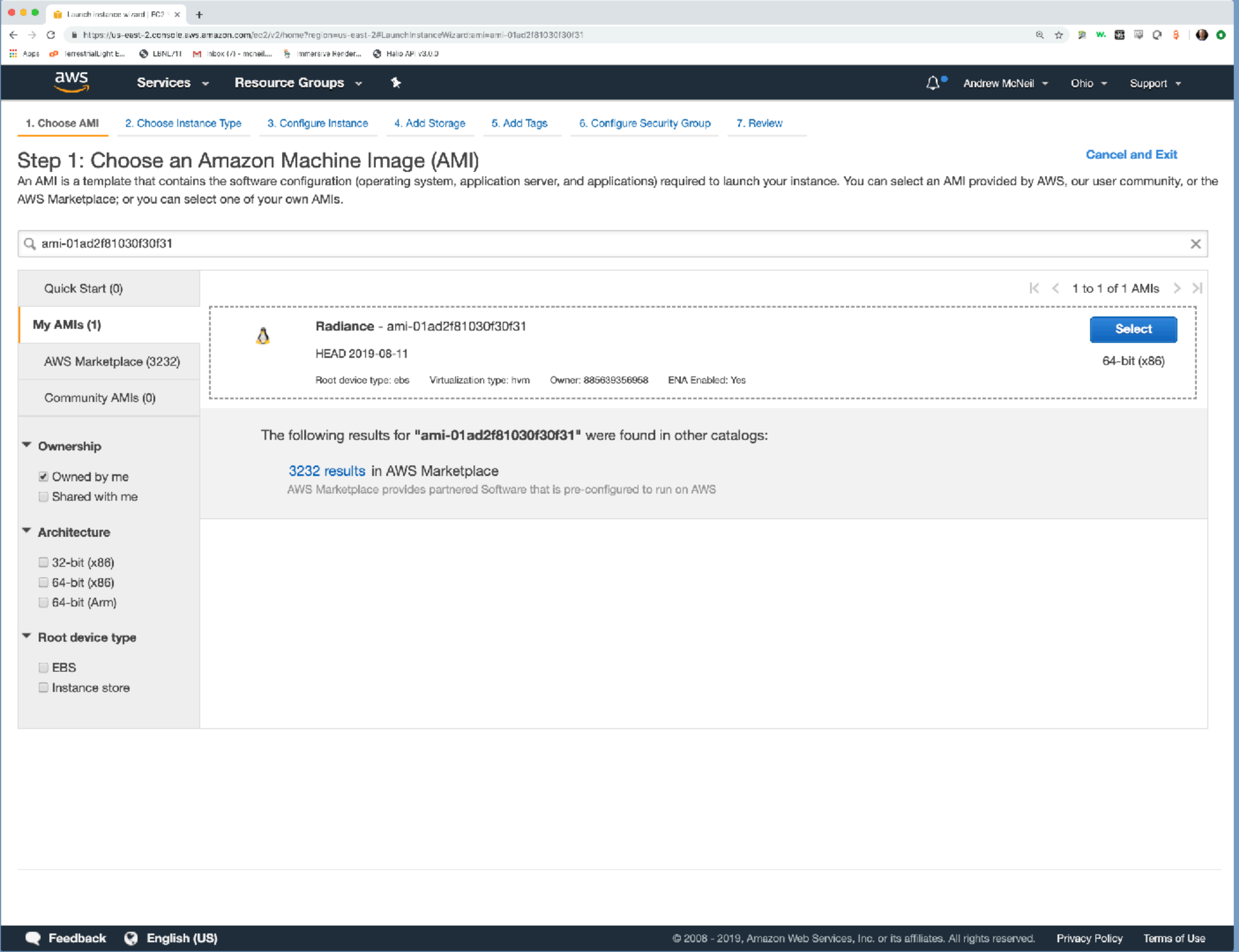
- There are some things you'll want to run every time you start an instance, for example:
  - sudo yum -y update
    (for security)
  - sudo mount -t efs myfilesystem:/ efs
- Configuration scripts can be provided as a text file when launching an instance.
- Or you could put this in a @reboot cron job on the machine image.

Example Configuration Script

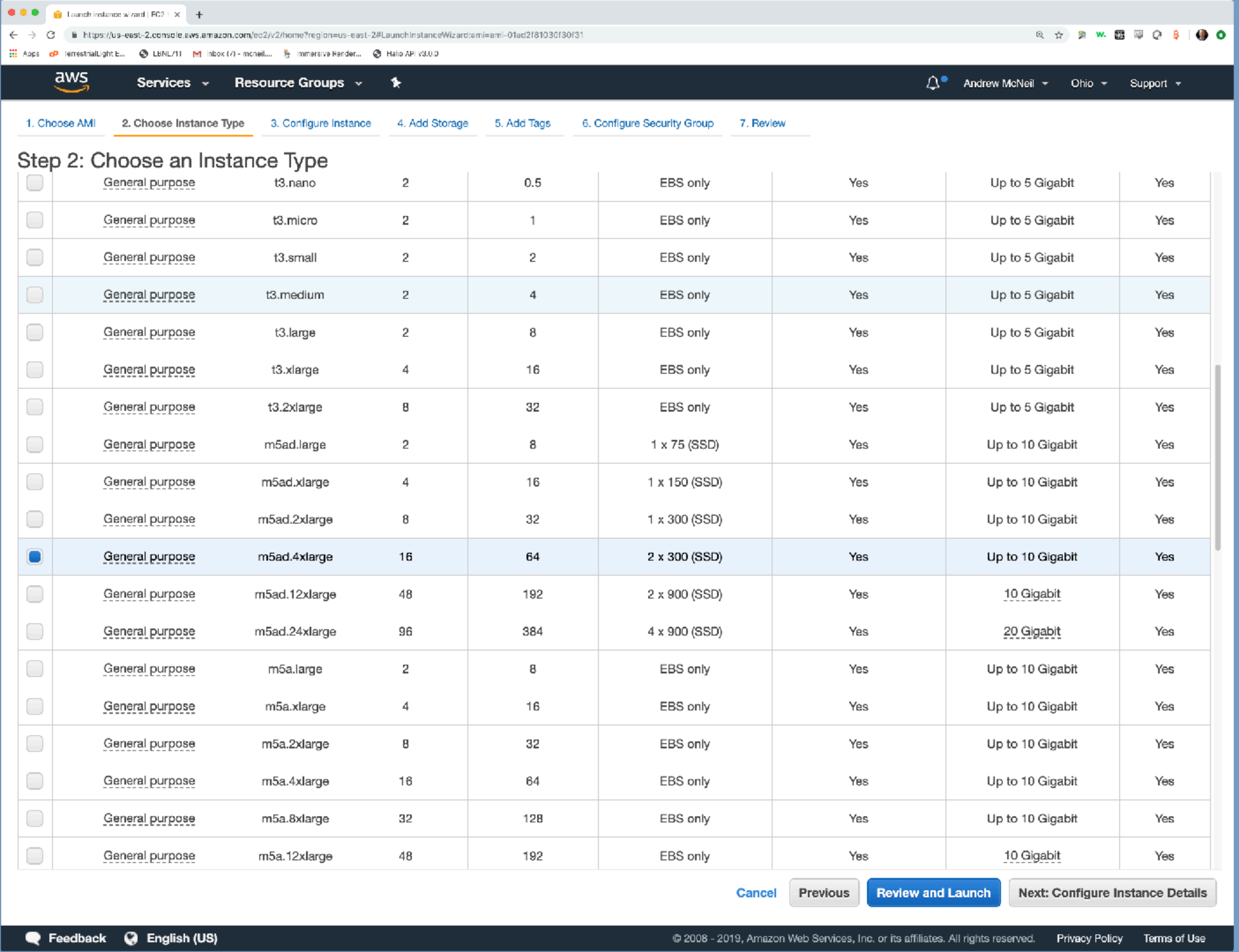
aws_startup.bsh:

```
#! /bin/bash

yum -y update
mount -t efs fs-8e7921f7:/ /home/ec2-user/efs
```

When provided as a configuration script, it is run with root privileges, so you don’t need sudo.



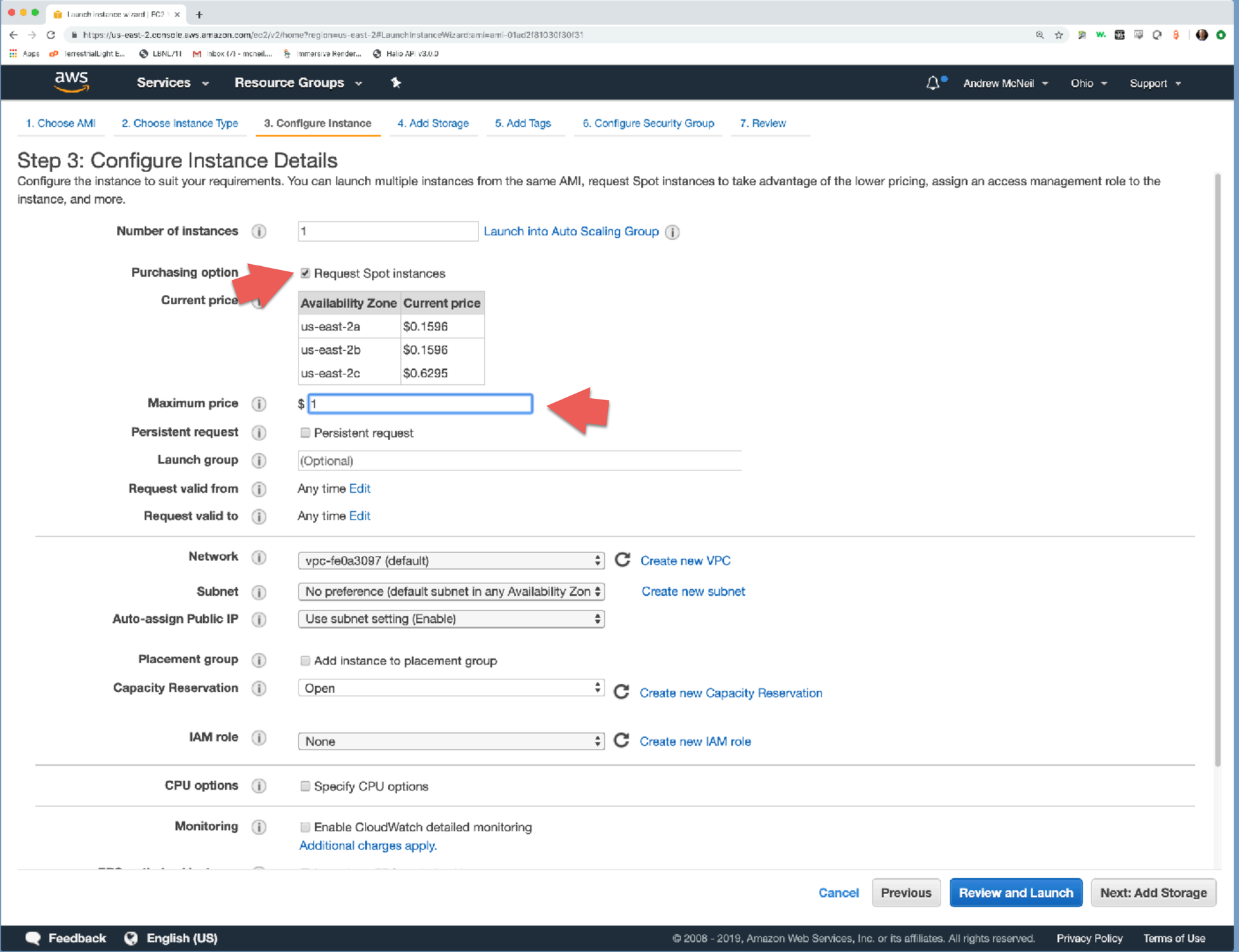
When we go to launch an instance the new image is available.



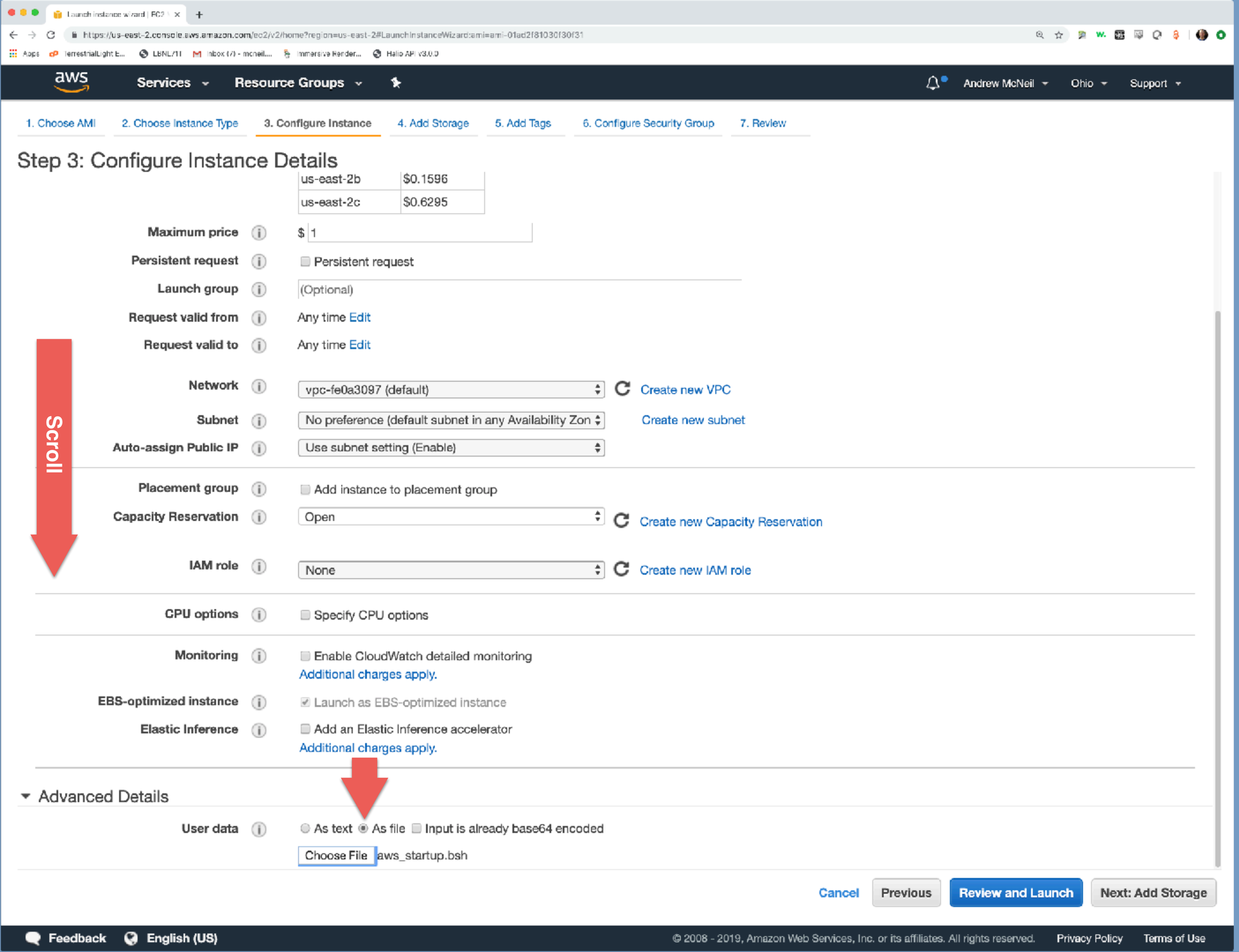
☐	General purpose	t3.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes
☐	General purpose	t3.micro	2	1	EBS only	Yes	Up to 5 Gigabit	Yes
☐	General purpose	t3.small	2	2	EBS only	Yes	Up to 5 Gigabit	Yes
☐	General purpose	t3.medium	2	4	EBS only	Yes	Up to 5 Gigabit	Yes
☐	General purpose	t3.large	2	8	EBS only	Yes	Up to 5 Gigabit	Yes
☐	General purpose	t3.xlarge	4	16	EBS only	Yes	Up to 5 Gigabit	Yes
☐	General purpose	t3.2xlarge	8	32	EBS only	Yes	Up to 5 Gigabit	Yes
☐	General purpose	m5ad.large	2	8	1 x 75 (SSD)	Yes	Up to 10 Gigabit	Yes
☐	General purpose	m5ad.xlarge	4	16	1 x 150 (SSD)	Yes	Up to 10 Gigabit	Yes
☐	General purpose	m5ad.2xlarge	8	32	1 x 300 (SSD)	Yes	Up to 10 Gigabit	Yes
☑	General purpose	m5ad.4xlarge	16	64	2 x 300 (SSD)	Yes	Up to 10 Gigabit	Yes
☐	General purpose	m5ad.12xlarge	48	192	2 x 900 (SSD)	Yes	10 Gigabit	Yes
☐	General purpose	m5ad.24xlarge	96	384	4 x 900 (SSD)	Yes	20 Gigabit	Yes
☐	General purpose	m5a.large	2	8	EBS only	Yes	Up to 10 Gigabit	Yes
☐	General purpose	m5a.xlarge	4	16	EBS only	Yes	Up to 10 Gigabit	Yes
☐	General purpose	m5a.2xlarge	8	32	EBS only	Yes	Up to 10 Gigabit	Yes
☐	General purpose	m5a.4xlarge	16	64	EBS only	Yes	Up to 10 Gigabit	Yes
☐	General purpose	m5a.8xlarge	32	128	EBS only	Yes	Up to 10 Gigabit	Yes
☐	General purpose	m5a.12xlarge	48	192	EBS only	Yes	10 Gigabit	Yes

I'm going to launch a 16 CPU instance this time.

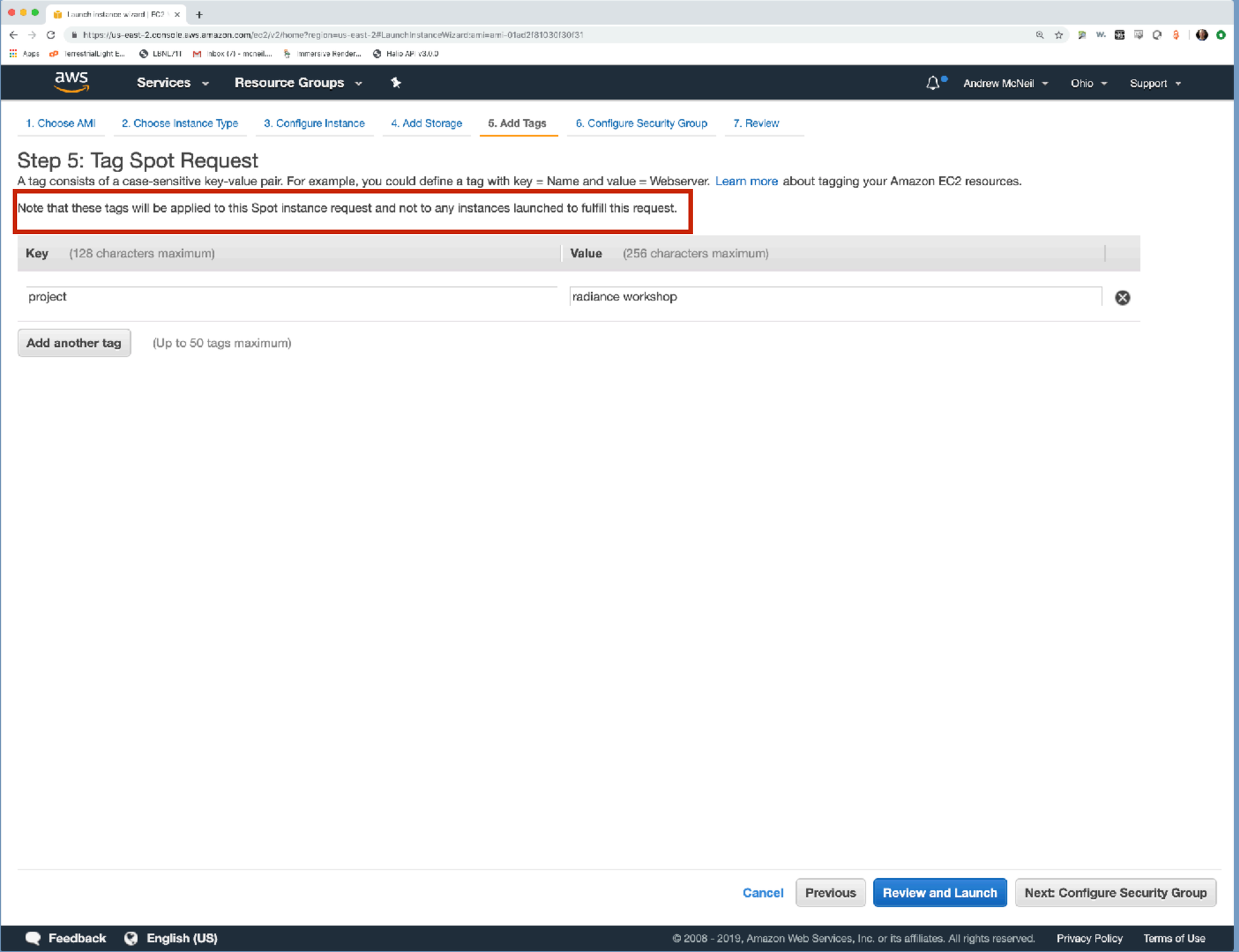
NOT FREE TIER!



And let's do a spot request too.



We can add our configuration script as a file under advanced details.



Tagging the spot request does not tag the instance. You need to tag the instance when it launches.

1. Choose AMI | 2. Choose Instance Type | 3. Configure Instance | 4. Add Storage | 5. Add Tags | 6. Configure Security Group | 7. Review

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

Assign a security group: ○ Create a **new** security group

● Select an **existing** security group

Security Group ID	Name	Description	Actions
sg-07c3608c40753cd0d	Basic	SSH & NFS	Copy to new
sg-6c47cc04	default	default VPC security group	Copy to new

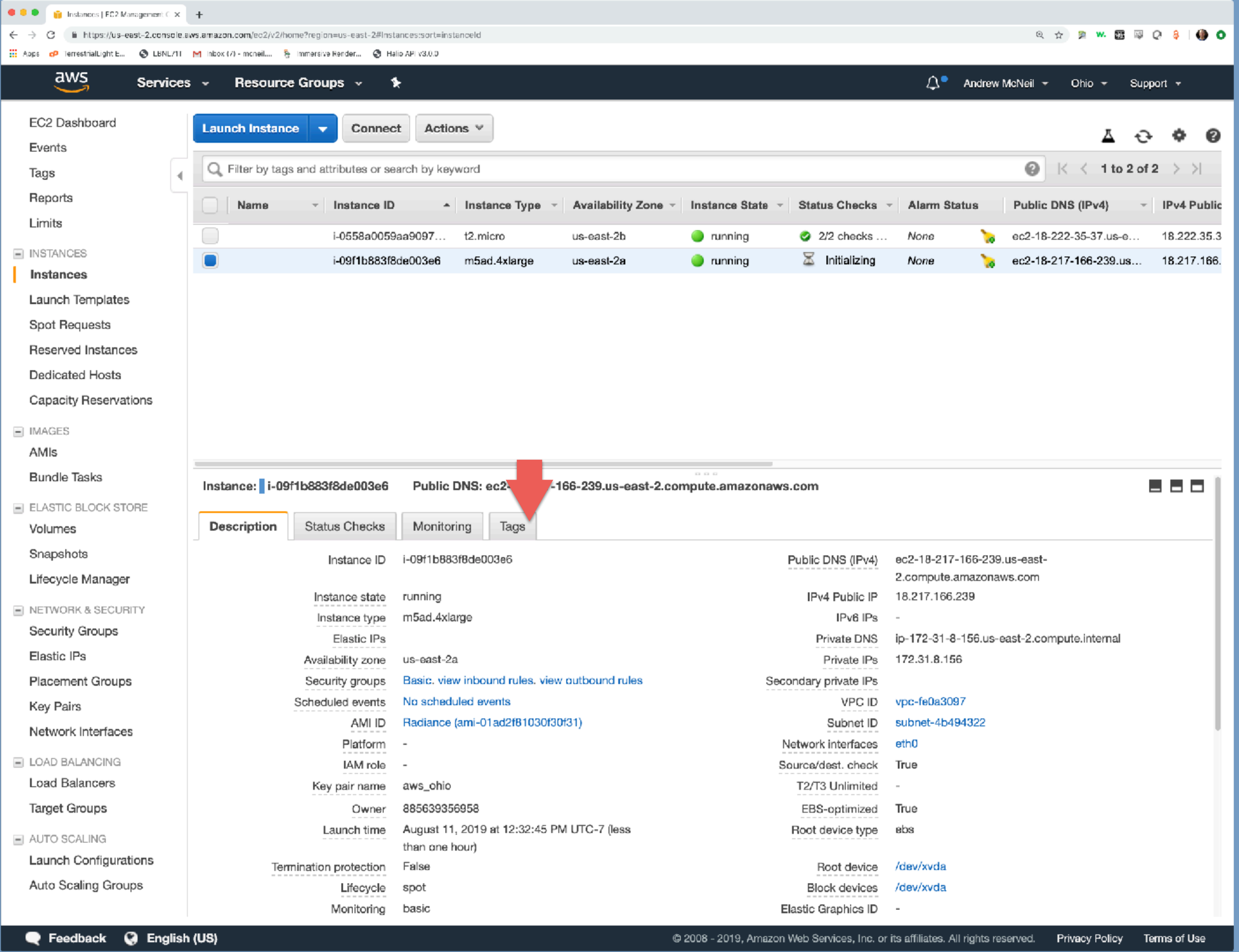
Warning

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

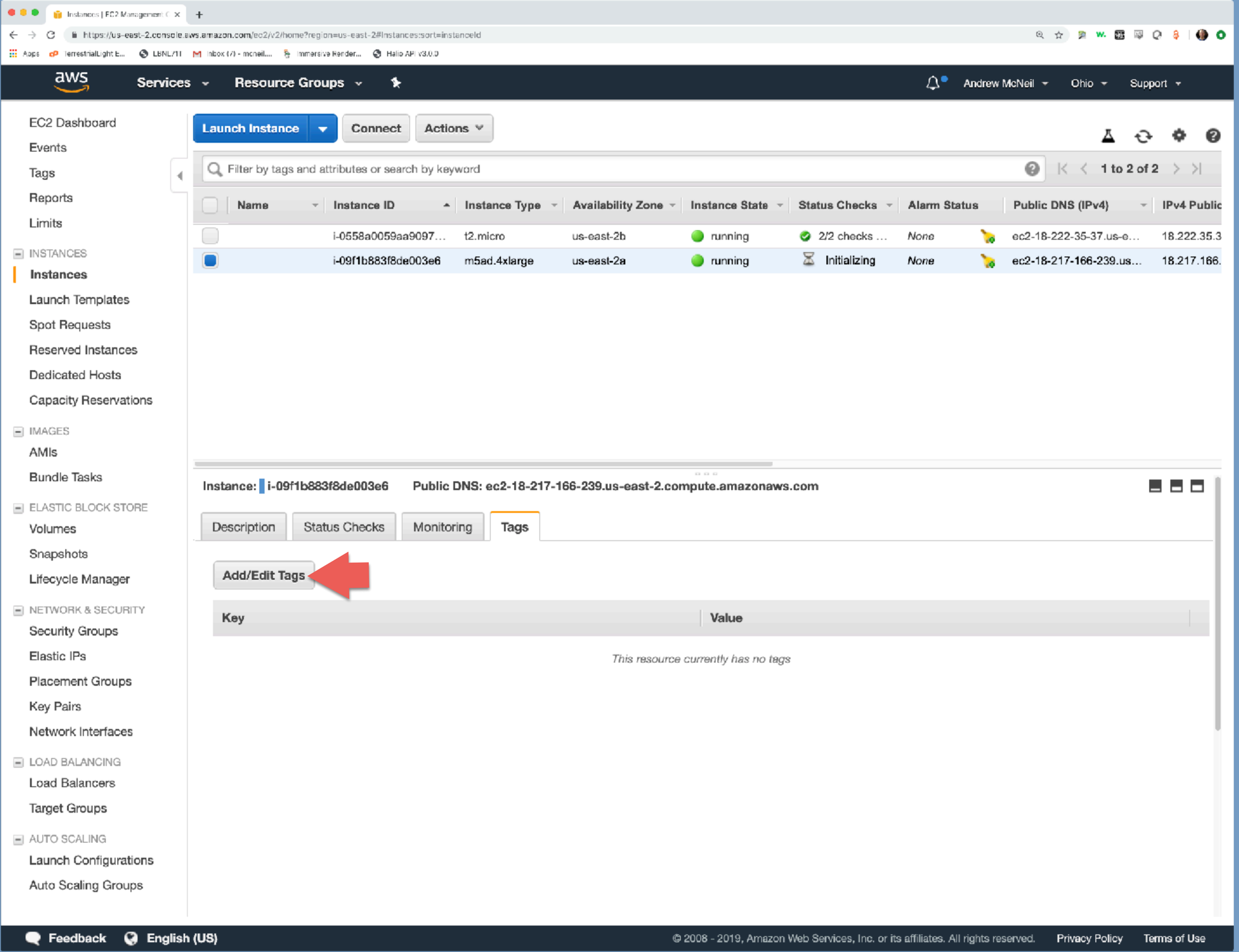
Inbound rules for sg-07c3608c40753cd0d (Selected security groups: sg-07c3608c40753cd0d)

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	0.0.0.0/0	
SSH	TCP	22	::/0	
NFS	TCP	2049	sg-07c3608c40753cd0d (Basic)	

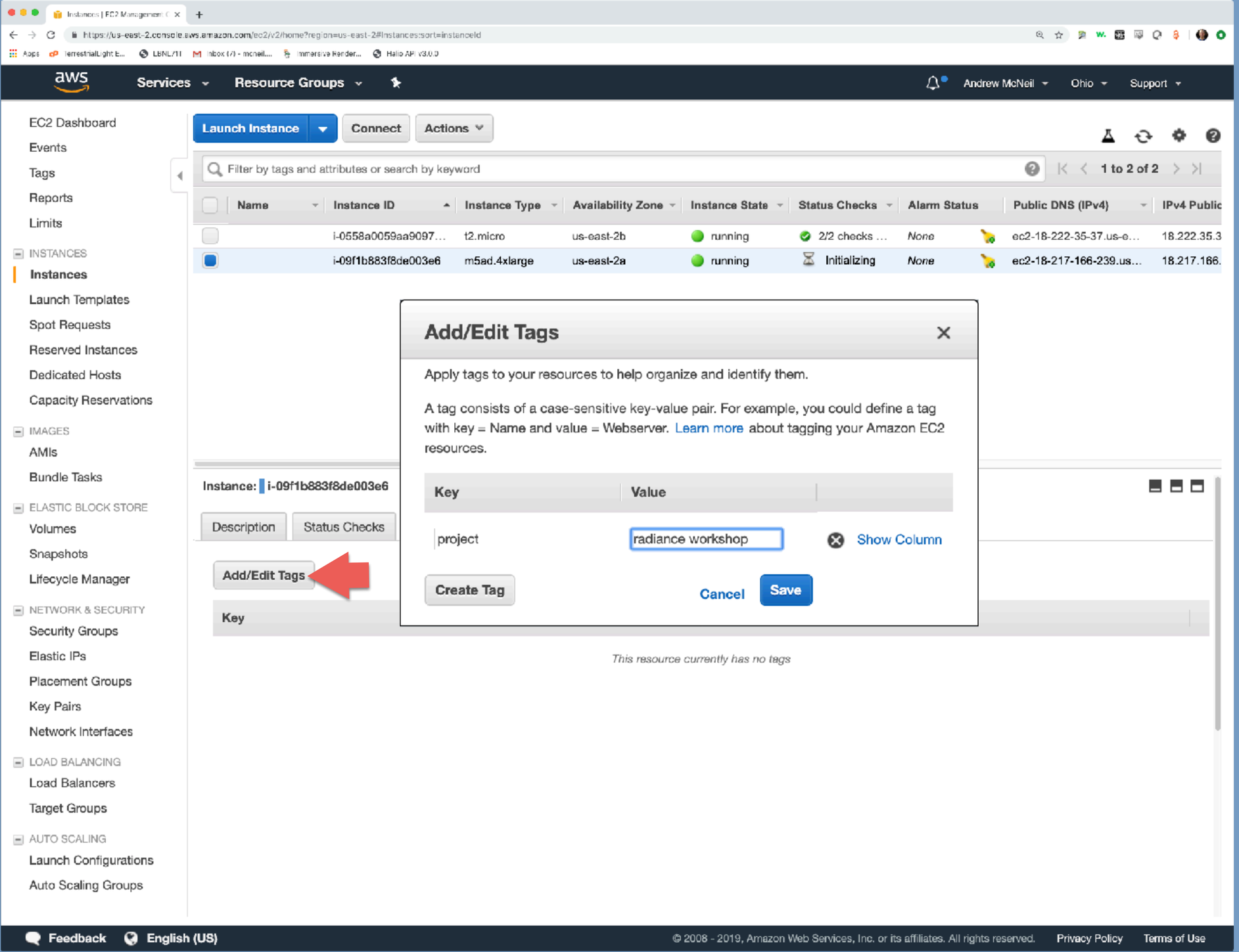
Cancel | Previous | Review and Launch



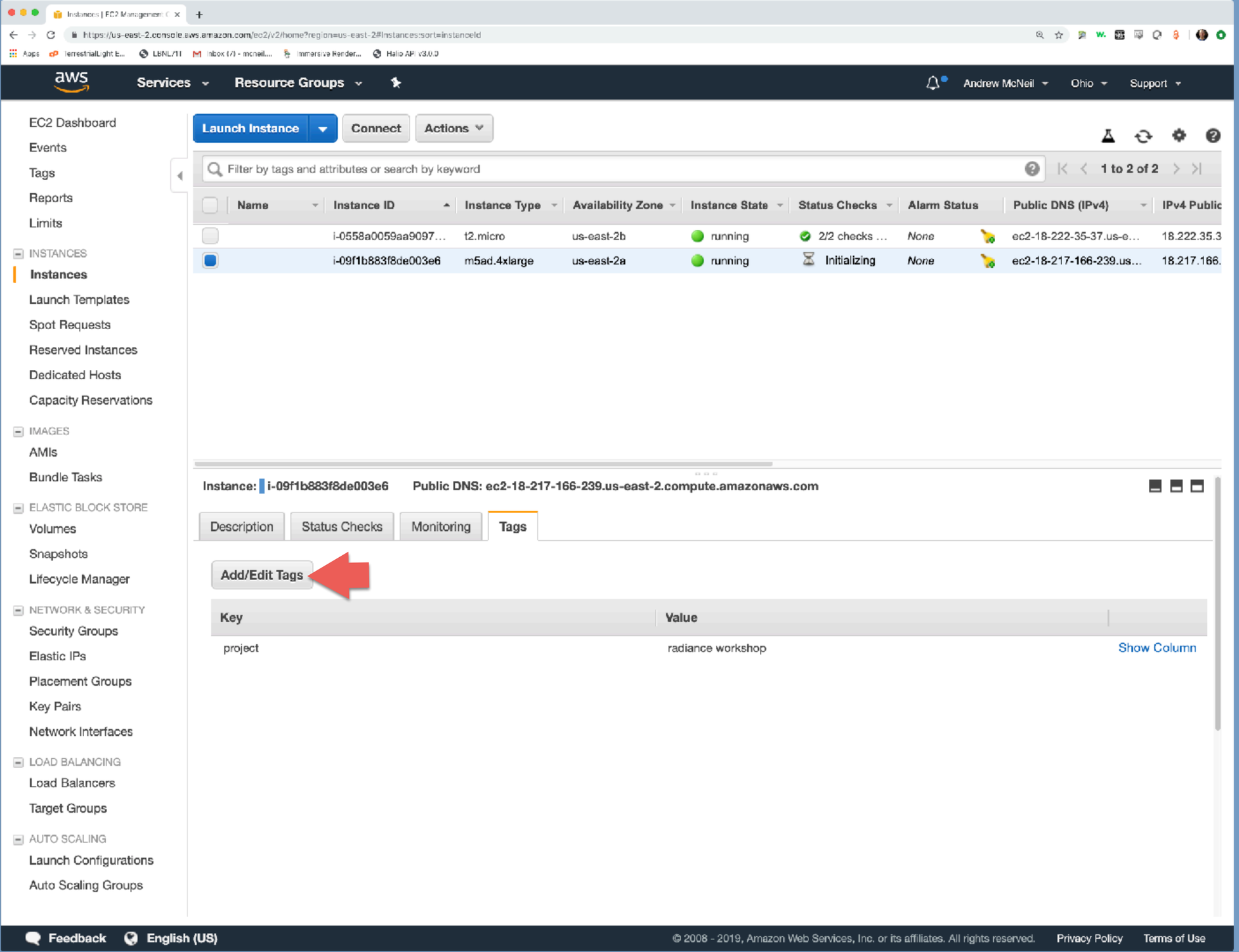
Now that the spot request is fulfilled, you should remember to tag the resulting instance.



Now you should remember to tag your spot fulfilled instance.



Now you should remember to tag your spot fulfilled instance.



Now you should remember to tag your spot fulfilled instance.

Upload model files - Mac / Linux

- scp (secure copy)
- Copies files over ssh (similar syntax to ssh)

```
scp -i aws_ohio.pem -r my_model/ ec2-user@18.222.35.37:~/efs/
```

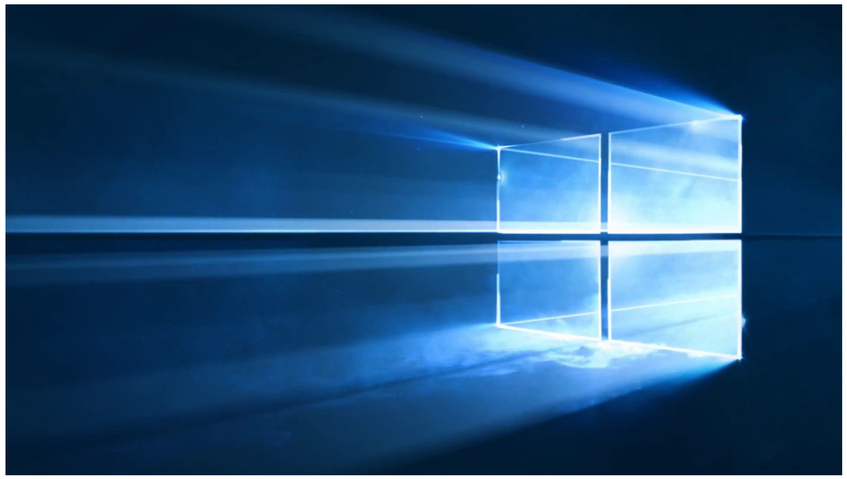
copy this

to here

-i key.pem : private key

-r : recursive (copies directories and contents)

Upload model files - Windows



- pscp (PuTTY secure copy)
- Copies files over ssh

```
pscp -i aws_ohio.ppk -r my_model/ ec2-user@18.222.35.37:
```

copy this

to here

-i key.ppk : private key

-r : recursive (copies directories and contents)

Uploading David’s example files

```
#These commands are run locally.
# upload zip files
scp -i ~/aws_ohio.pem 01_genBSDF.zip ec2-user@18.222.35.37:~/efs/.
scp -i ~/aws_ohio.pem example_noResults.zip ec2-user@18.222.35.37:~/efs/.
```

Start Simulation!!!!

- nohup (very important)
  - keeps a process running until it finishes, even if you log out
  - sends stdout to a file, nohup.out by default
  - to stop a process running with nohup, us ps to get the process id and kill to end it.

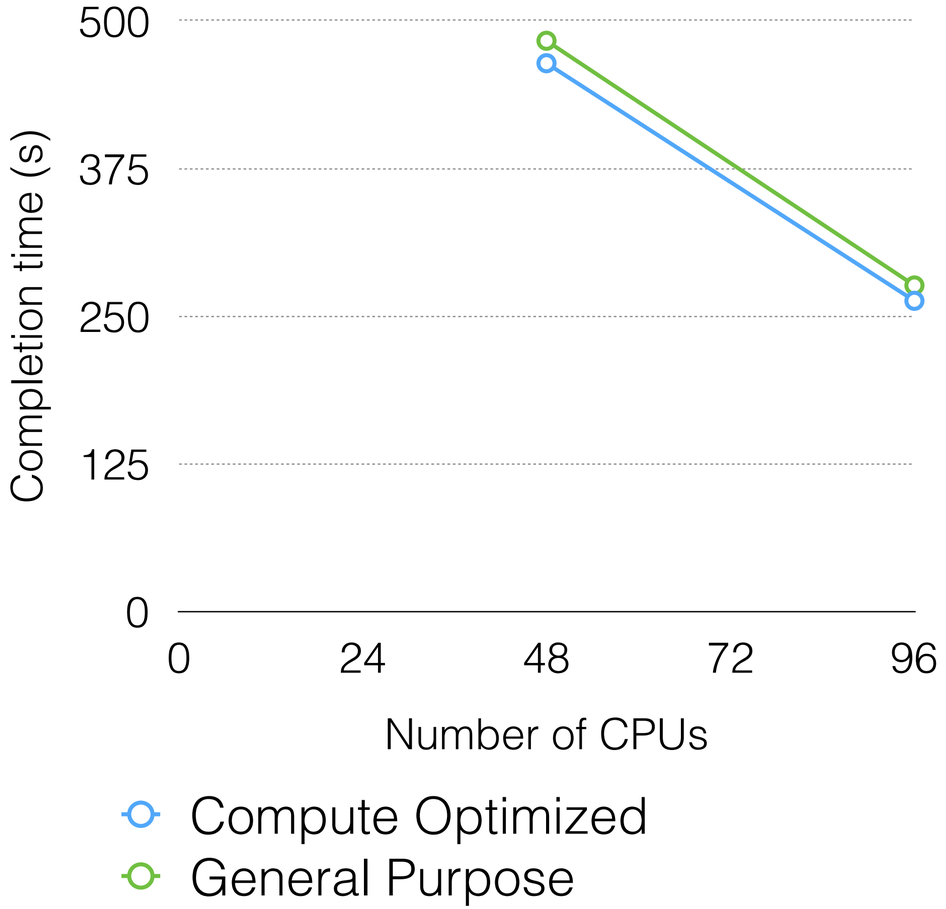
Running David's BSDF example

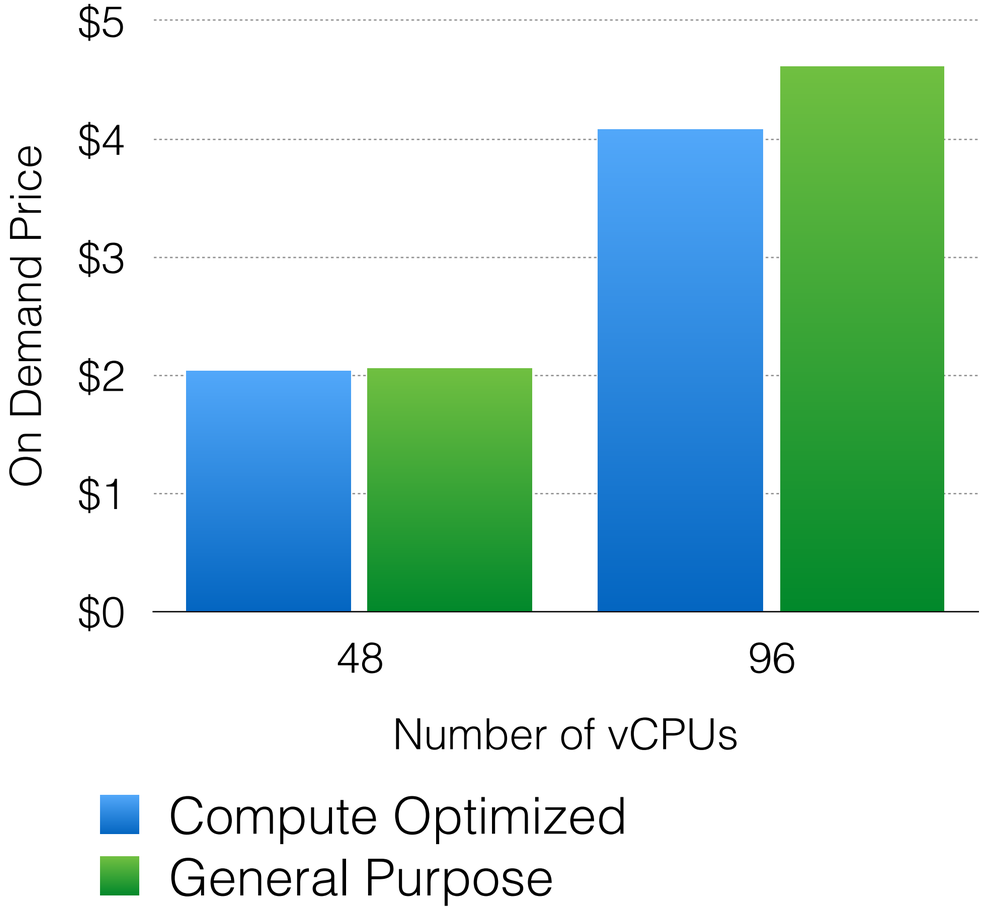
```
# unpack zip files
unzip 01_genBSDF.zip
unzip example_noResults.zip

# change permission to make scripts executable
# this can be avoided if script has 'bash myscript.sh' instead of './myscript.sh'
cd example_noResults
chmod +x *.sh scripts/*.sh

# copy cal file to current directory so script can find it
cp ../01_genBSDF/window7_2side.cal .

# run command
nohup time bash 00_all.sh &
```

Benchmarks on David's example

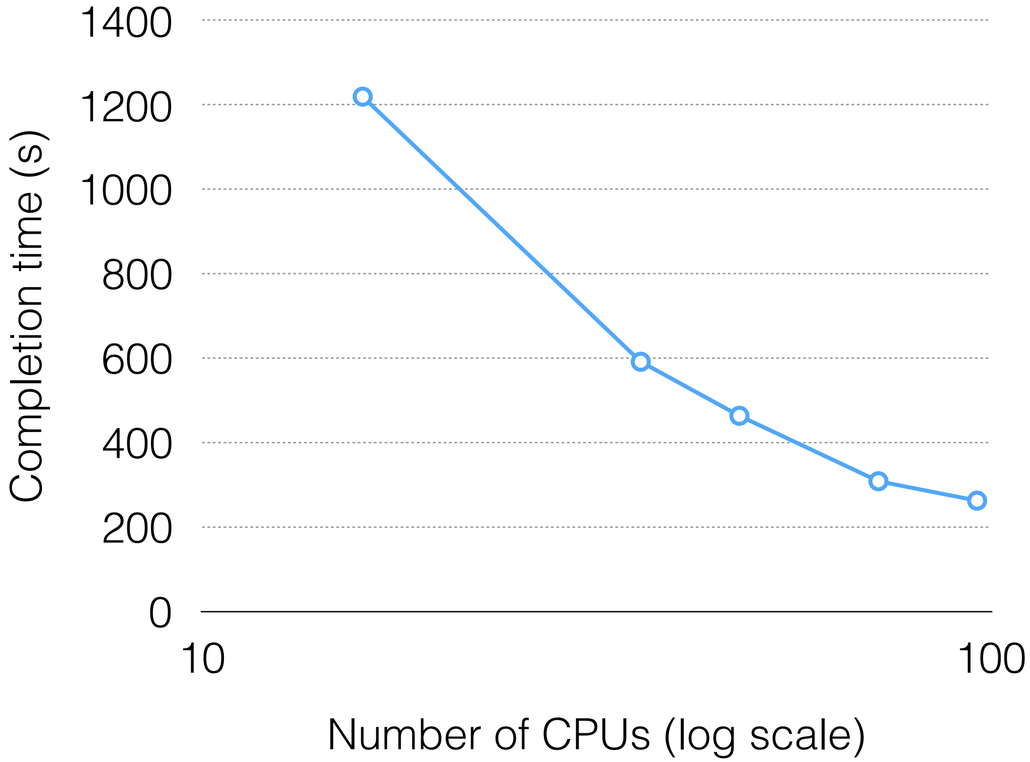
Instance Type	vCPU	Physical Prodessor	Clock Speed	Memory (GiB)	File Storage	Time (s)	On Demand		Spot (18 Aug, 15:00)	
							Rate/hr	Cost	Rate/hr	Cost
c5.24xlarge	96	2nd Gen Xeon Platinum 8175CL	3.0 GHz	192	EBS	263	$4.08	$0.298	$0.91	$0.066
c5.24xlarge	96	2nd Gen Xeon Platinum 8175CL	3.0 GHz	192	EFS	441	$4.08	$0.500	$0.91	$0.111
m5.24xlarge	96	Intel Xeon Platinum 8175	3.1 GHz	384	EBS	276	$4.61	$0.354	$0.96	$0.074
m5d.24xlarge	96	Intel Xeon Platinum 8175	3.1 GHz	384	SSD	278	$5.42	$0.419	$0.96	$0.074
m5d.24xlarge	96	Intel Xeon Platinum 8175	3.1 GHz	384	EFS	463	$5.42	$0.697	$0.96	$0.123
c5.18xlarge	72	Intel Xeon Platinum 8124M	3 GHz	144	EBS	309	$3.06	$0.263	$0.80	$0.069
c5.18xlarge	72	Intel Xeon Platinum 8124M	3 GHz	144	EBS	501	$3.06	$0.426	$0.80	$0.111
c5.12xlarge	48	2nd Gen Xeon Platinum 8175CL	3.0 GHz	96	EBS	464	$2.04	$0.263	$0.46	$0.059
c5.12xlarge	48	2nd Gen Xeon Platinum 8175CL	3.0 GHz	96	EFS	638	$2.04	$0.362	$0.46	$0.082
m5.12xlarge	48	Intel Xeon Platinum 8175	3.1 GHz	192	EBS	483	$2.06	$0.276	$0.54	$0.072
m5.12xlarge	48	Intel Xeon Platinum 8175	3.1 GHz	192	EFS	665	$2.06	$0.380	$0.54	$0.100
c5.9xlarge	36	Intel Xeon Platinum 8124M	3 GHz	72	EBS	592	$1.53	$0.252	$0.39	$0.064
c5.9xlarge	36	Intel Xeon Platinum 8124M	3 GHz	72	EFS	774	$1.53	$0.329	$0.39	$0.084
c5.4xlarge	16	Intel Xeon Platinum 8124M	3 GHz	32	EBS	1220	$0.68	$0.230	$0.15	$0.051
c5.4xlarge	16	Intel Xeon Platinum 8124M	3 GHz	32	EFS	1415	$0.68	$0.267	$0.15	$0.059

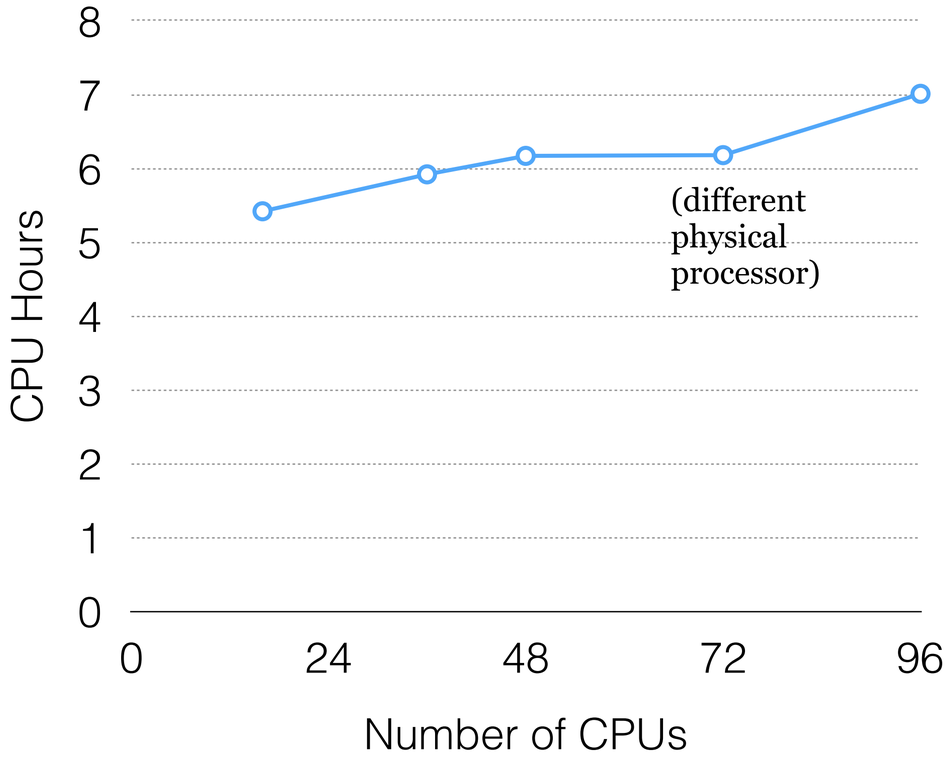
Compute optimized instances completed the script in less time and cost less.





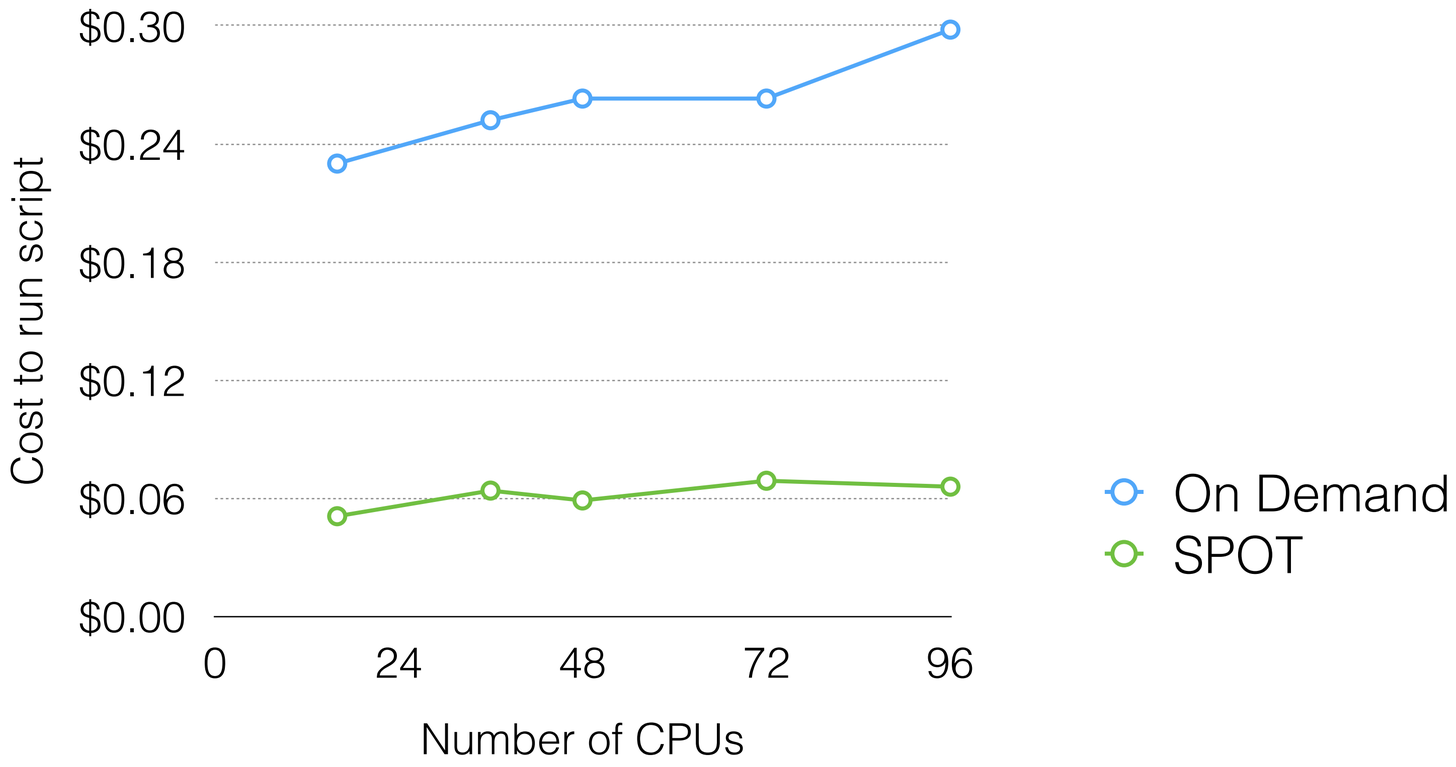
Increasing CPU count has diminishing returns, but for this simulation up to 96 CPUs are still effective.





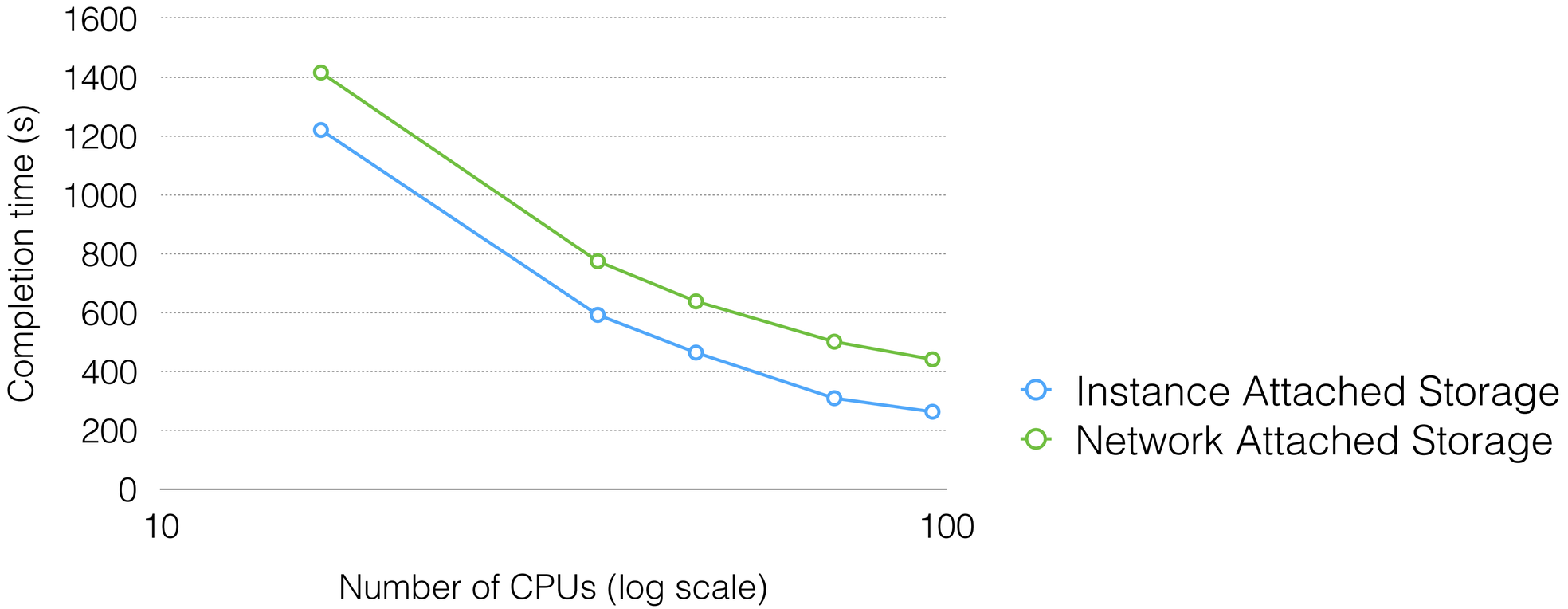
Compute Optimized Instances

The simulation cost increases with more vCPUs.



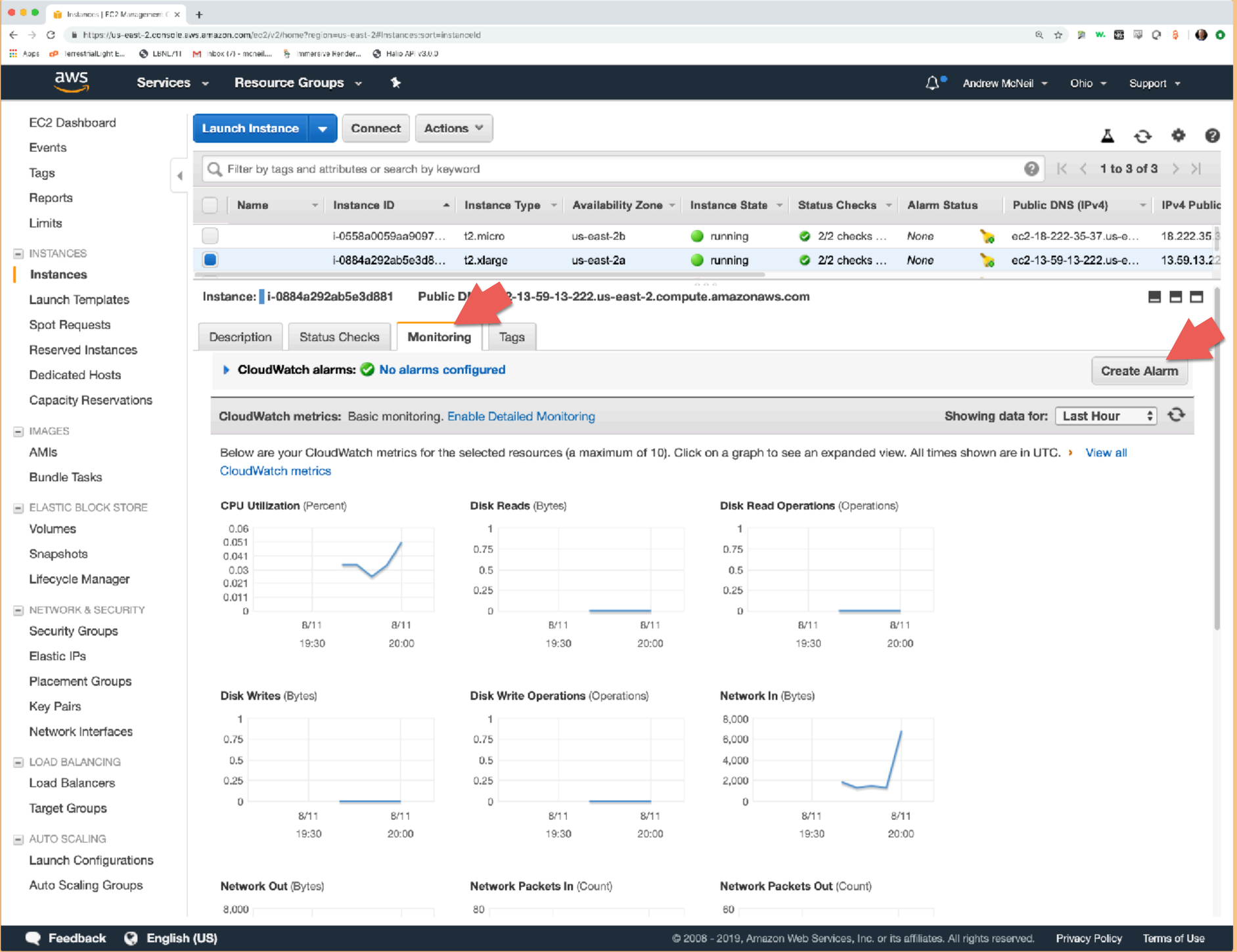
Cost is hourly price time simulation time. This assumes the instance is terminated immediately. Spot prices subject to variability.

Using instance attached storage (EBS) for simulation files was faster than using network attached storage (EFS)



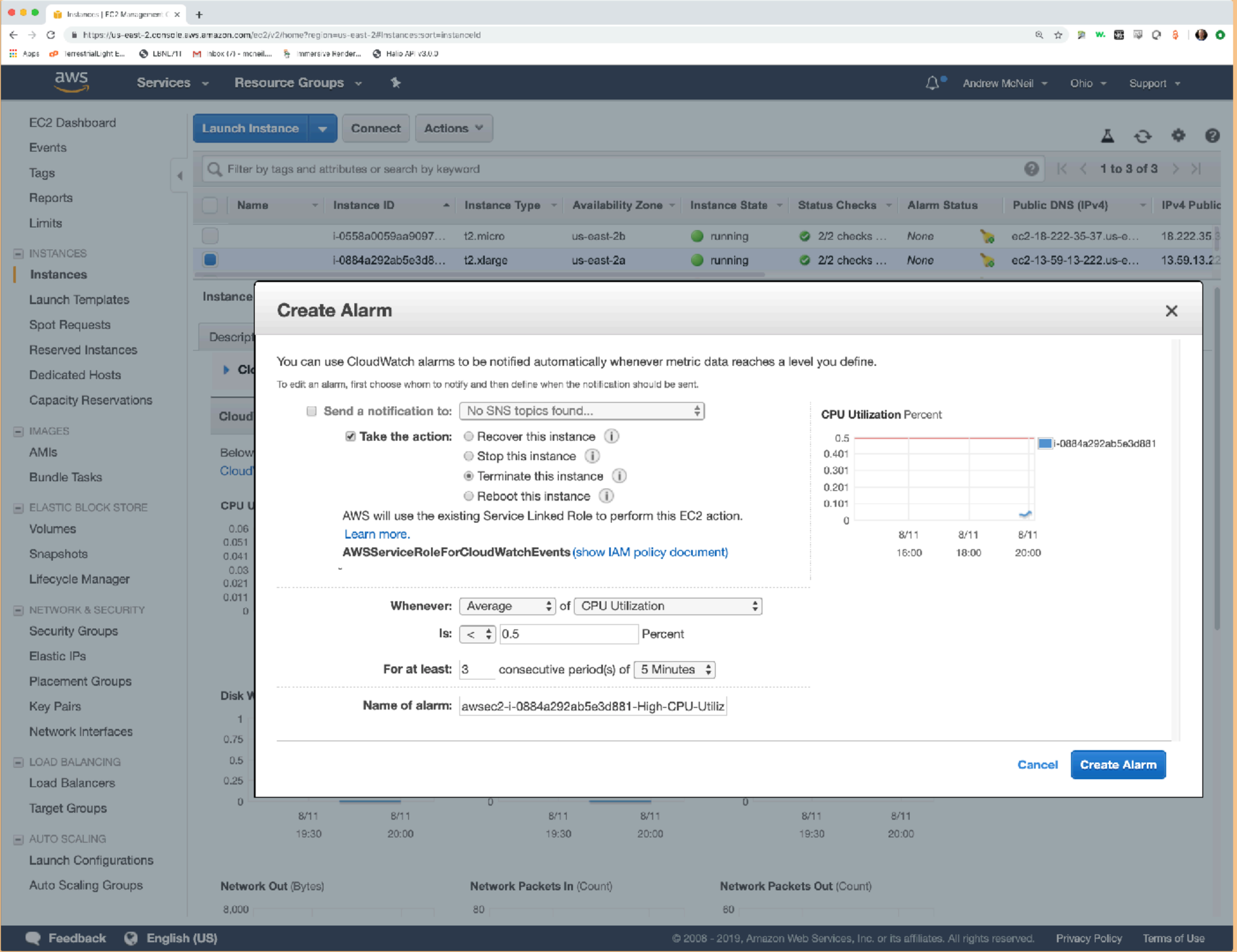
Alarms

- Alarms take an action when the instance usage crosses a threshold.
- For example (and the only reason I use alarms):
  - Terminate the instance when the CPU is below 1% for 10 minutes
- Alarms cost $0.10 each, though they save money by terminating idle instances (just make sure your results are on or moved to persistent storage).



Click on monitoring tab

Then on Create Alarm button



Alarms cost $0.10 each.

They save money by terminating idle instances (just make sure your results are on or moved to persistent storage).

Download Results

- If you used an alarm to terminate your instance, you'll need to launch another instance to get your results.
- Use scp (Mac/Linux) or pscp (Windows) to download your result.

```
scp -i aws_ohio.pem -r ec2-user@18.222.35.37:~/efs/my_model/results/ .

pscp -i aws_ohio.ppk -r ec2-user@18.222.35.37:~/efs/my_model/results/ .
```

copy this

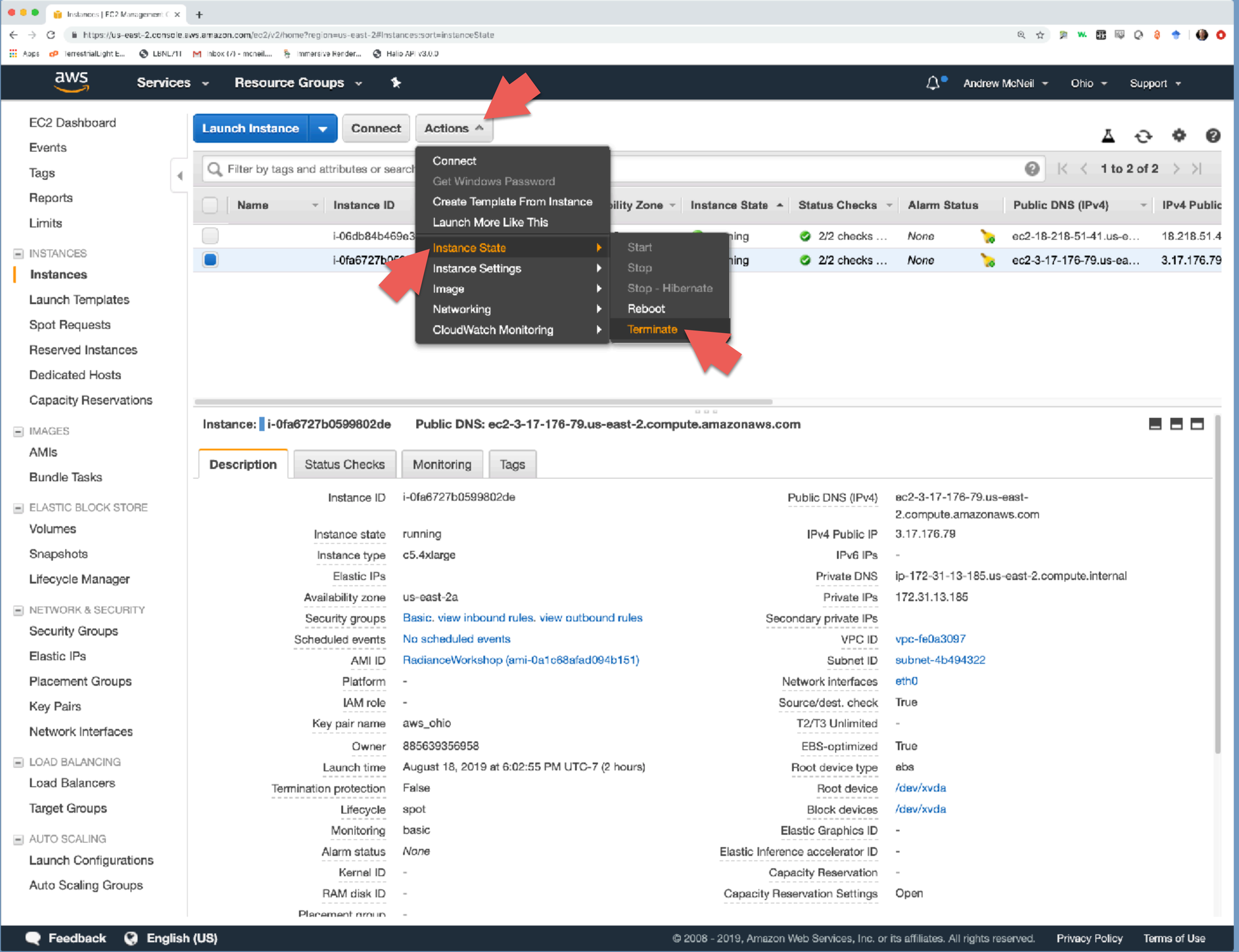
to here

Downloading Results

```
#These commands are run locally.
# download results file
scp -i ~/aws_ohio.pem -r ec2-user@18.222.35.37:~/efs/example_noResults/result .
```

Terminate Instance

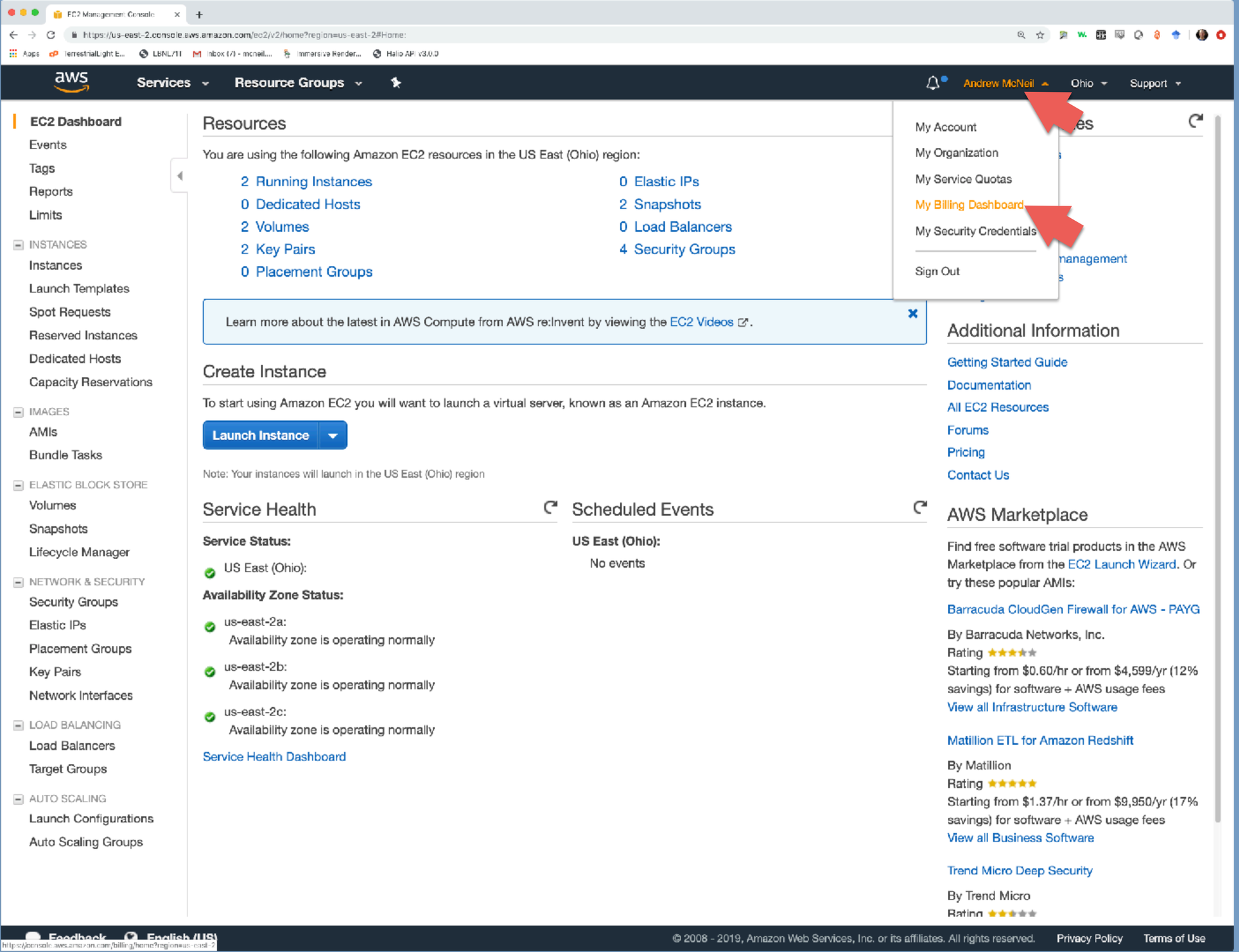
- When you’re done, you terminate your instance
- *Not much else to say, this one is pretty self explanatory.*



Possibly the easiest step.

Invoice your client

- Activate tags for cost tracking
- Be diligent about tagging resources (this is the hard part)
  - Resources can NOT be tagged retroactively
- Use cost explorer to aggregate costs for each project


aws
Services
Resource Groups
Andrew McNeil
Ohio
Support
EC2 Dashboard
Events
Tags
Reports
Limits
INSTANCES
Instances
Launch Templates
Spot Requests
Reserved Instances
Dedicated Hosts
Capacity Reservations
IMAGES
AMIs
Bundle Tasks
ELASTIC BLOCK STORE
Volumes
Snapshots
Lifecycle Manager
NETWORK & SECURITY
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Network Interfaces
LOAD BALANCING
Load Balancers
Target Groups
AUTO SCALING
Launch Configurations
Auto Scaling Groups
Resources
You are using the following Amazon EC2 resources in the US East (Ohio) region:
2 Running Instances
0 Dedicated Hosts
2 Volumes
2 Key Pairs
0 Placement Groups
0 Elastic IPs
2 Snapshots
0 Load Balancers
4 Security Groups
Learn more about the latest in AWS Compute from AWS re:Invent by viewing the EC2 Videos.
Create Instance
To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.
Launch Instance
Note: Your instances will launch in the US East (Ohio) region
Service Health
Service Status:
US East (Ohio):
Availability Zone Status:
us-east-2a:
Availability zone is operating normally
us-east-2b:
Availability zone is operating normally
us-east-2c:
Availability zone is operating normally
Service Health Dashboard
Scheduled Events
US East (Ohio):
No events
My Account
My Organization
My Service Quotas
My Billing Dashboard
My Security Credentials
Sign Out
Additional Information
Getting Started Guide
Documentation
All EC2 Resources
Forums
Pricing
Contact Us
AWS Marketplace
Find free software trial products in the AWS Marketplace from the EC2 Launch Wizard. Or try these popular AMIs:
Barracuda CloudGen Firewall for AWS - PAYG
By Barracuda Networks, Inc.
Rating
Starting from $0.60/hr or from $4,599/yr (12% savings) for software + AWS usage fees
View all Infrastructure Software
Matillion ETL for Amazon Redshift
By Matillion
Rating
Starting from $1.37/hr or from $9,950/yr (17% savings) for software + AWS usage fees
View all Business Software
Trend Micro Deep Security
By Trend Micro
Rating
Feedback
English (US)
© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

Billing Management Console

https://console.aws.amazon.com/billing/home?region=us-east-2#/

Apps | TerrestrialLight E... | LBNL/IT | Inbox (7) - mcneil.... | Immersive Render... | Hello API v3.0.0

aws | Services | Resource Groups | Andrew McNeil | Global | Support

Home

Cost Management
- Cost Explorer
- Budgets
- Budgets Reports
- Cost & Usage Reports
- Cost allocation tags

Billing
- Bills
- Payment history
- Credits

Preferences
- Billing preferences
- Payment methods
- Consolidated billing
- Tax settings

Billing & Cost Management Dashboard

Getting Started with AWS Billing & Cost Management

- Manage your costs and usage using AWS Budgets
- Visualize your cost drivers and usage trends via Cost Explorer
- Dive deeper into your costs using the Cost and Usage Reports with Athena integration
- **Learn more:** Check out the AWS What's New webpage

Do you have Reserved Instances (RIs)?

- Access the RI Utilization & Coverage reports—and RI purchase recommendations—via Cost Explorer.

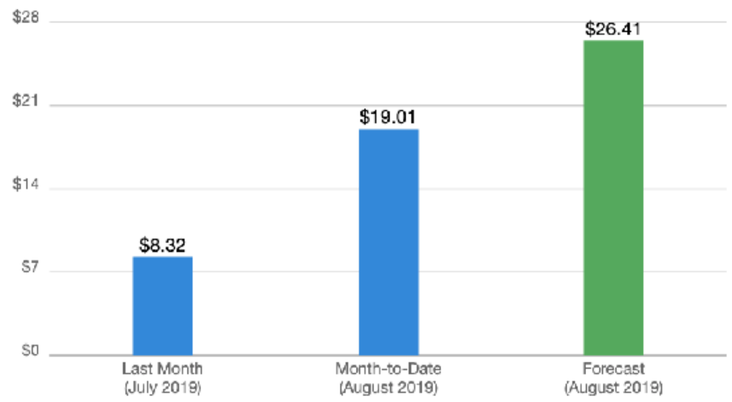
Spend Summary

Cost Explorer

Welcome to the AWS Billing & Cost Management console. Your last month, month-to-date, and month-end forecasted costs appear below.

Current month-to-date balance for August 2019

$19.01



Important Information about these Costs

Top Free Tier Services by Usage

View all

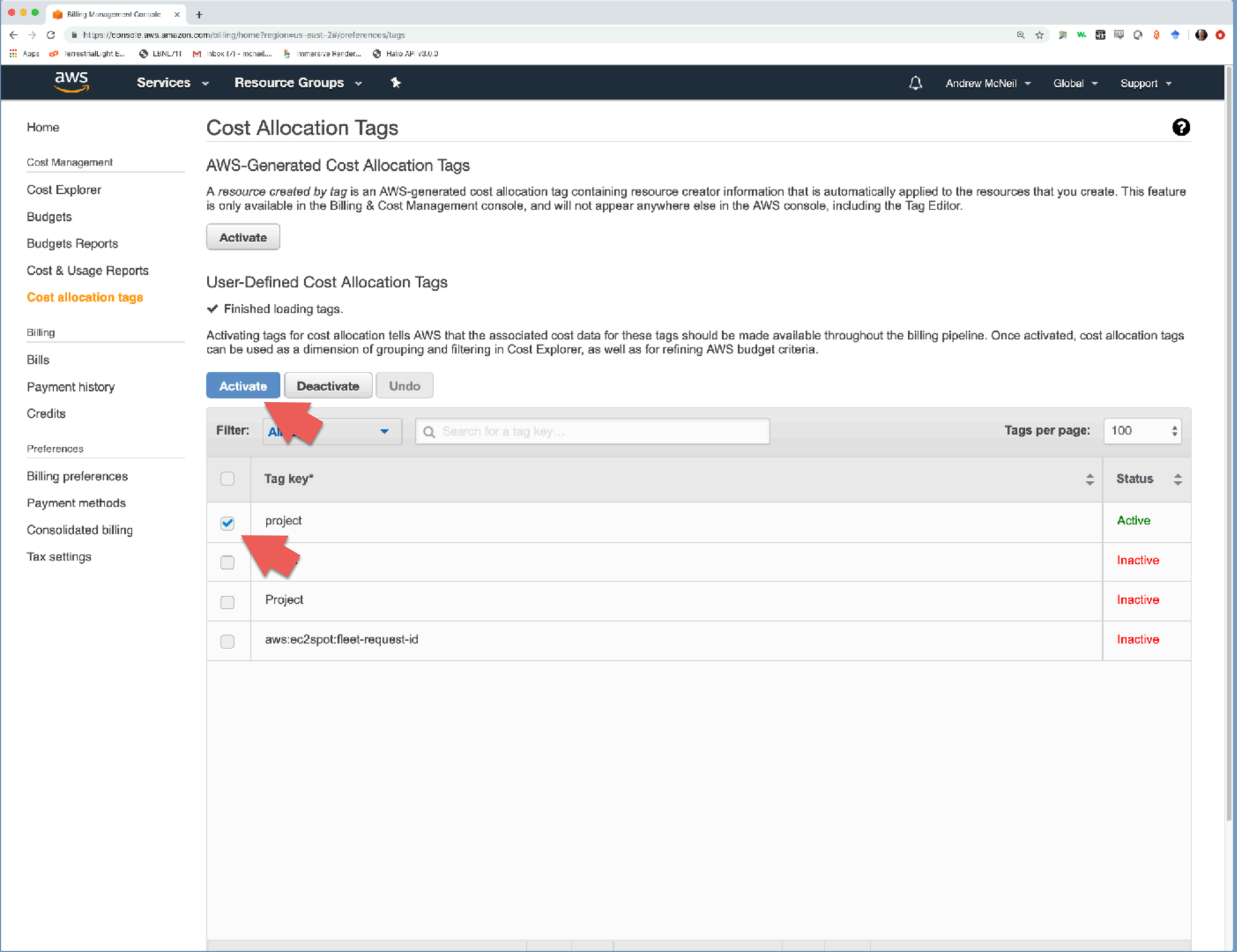
Month-to-Date Spend by Service

Bill Details

The chart below shows the proportion of costs spent for each service you use.

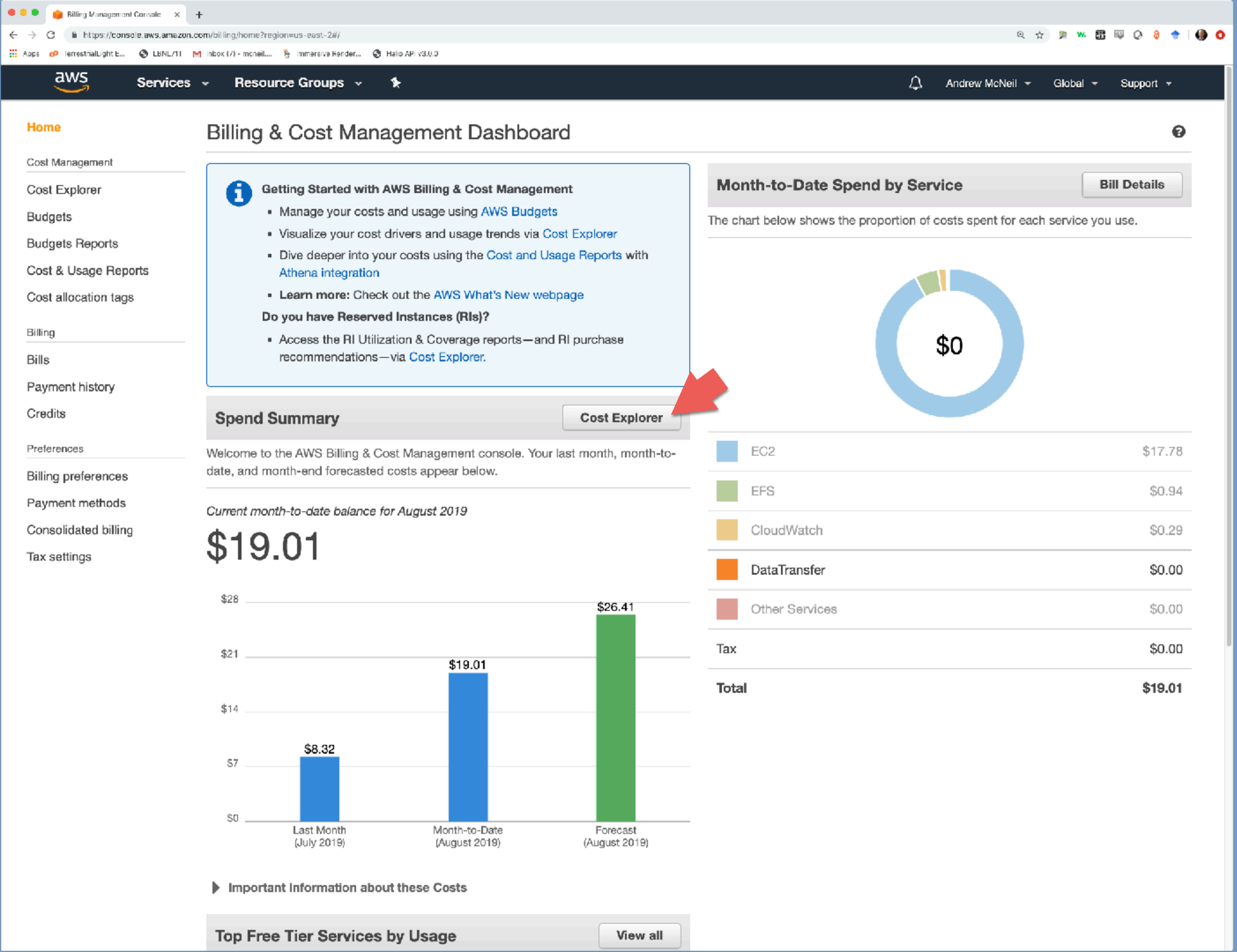
EC2	$17.78
EFS	$0.94
CloudWatch	$0.29
DataTransfer	$0.00
Other Services	$0.00
Tax	$0.00
Total	**$19.01**

Click cost allocation tags



Select the tag(s) to track for cost allocations

Activate them



Click cost explorer button

Cost Reports

https://console.aws.amazon.com/cost-reports/home#/dashboard

Apps | TerrestrialLight E... | LBNL/IT | Inbox (7) - mcneil.... | Immersive Render... | Hello API v3.0.0

aws | Services | Resource Groups | Andrew McNeil | Global | Support

AWS Cost Management

Month-to-date costs

$18.99

330%
Over last month

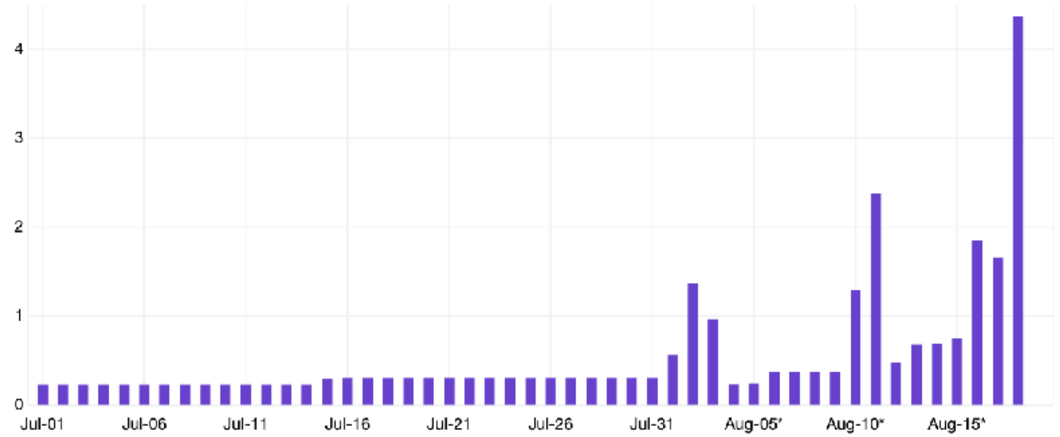
Forecasted month end costs

$42.26

408%
Over last month

Daily unblended costs ($)

Explore costs



August trends

Service usage

Amazon Elastic Compute Cloud - Compute costs are up $14.00 (623%)

AmazonCloudWatch costs are up $0.01 (5%)

Account usage

Andrew McNeil (885639356958) costs are up $14.58 (330%)

Region usage

us-west-2 costs are up $3.95 (89%)

Recently accessed reports

View all reports

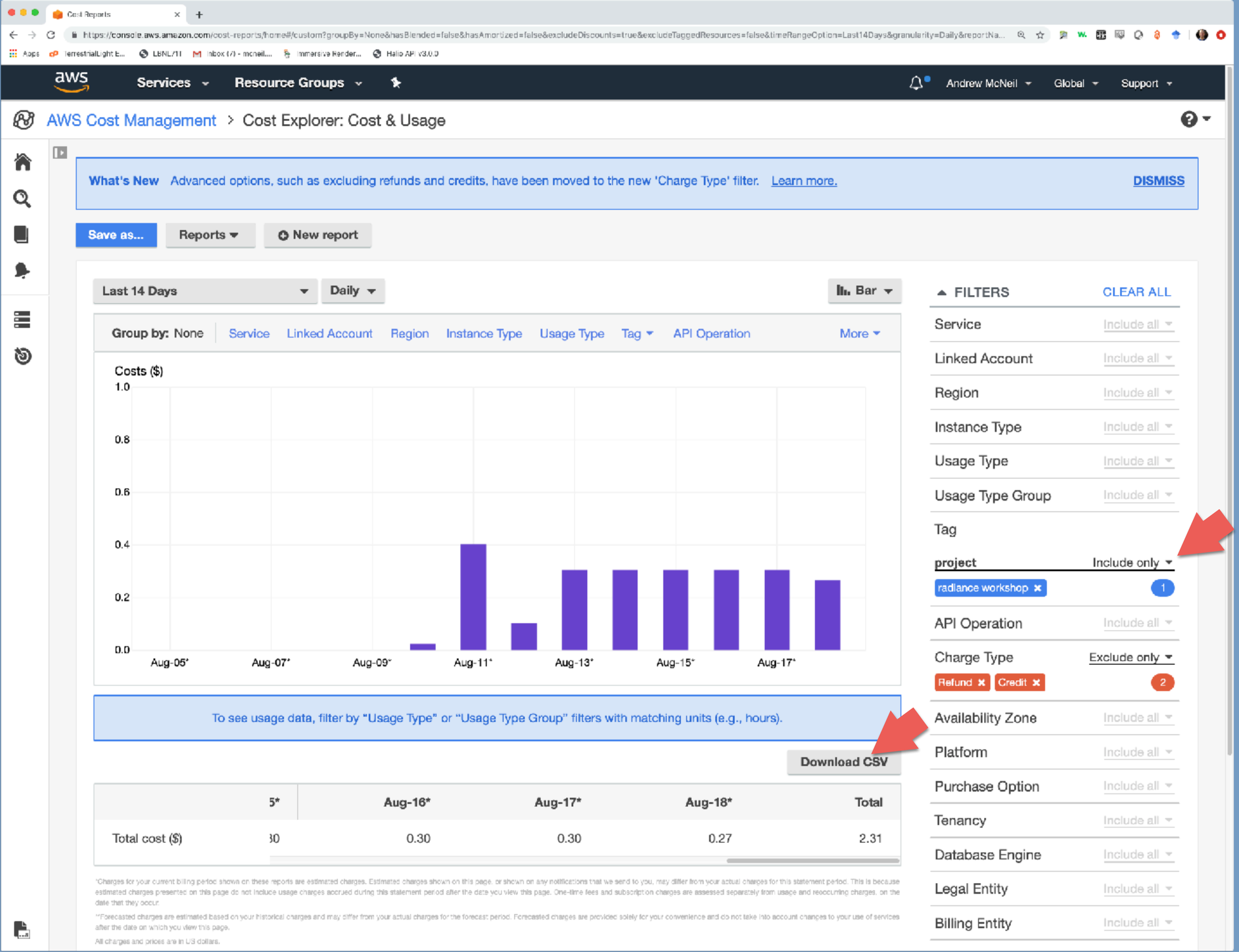
Report name	Report type	Time granularity	Last accessed
Daily costs	Cost & Usage	Daily	5 days ago

Viewing 1 of 7 total reports

Feedback | English (US)

 Privacy Policy | Terms of Use

Click explore costs



You can filter by may attributes, including tags.

And you can download a CSV.

Bask in the glory!

Andy's Cloud College

THIS DIPLOMA IS PRESENTED TO

You

For Clouding Along During the Cloud Course

21 August 2019