Running MESA on Amazon EC2 Instances: A Guide

by Andrew Mizener (amizener@macalester.edu)

Introduction

Welcome! In this guide, I'll be demonstrating how to set up MESA on Amazon EC2 instances. I've divided into three sections:

I. Introduction (you are here!)

II. First Steps: Installing MESA on a single EC2 instance

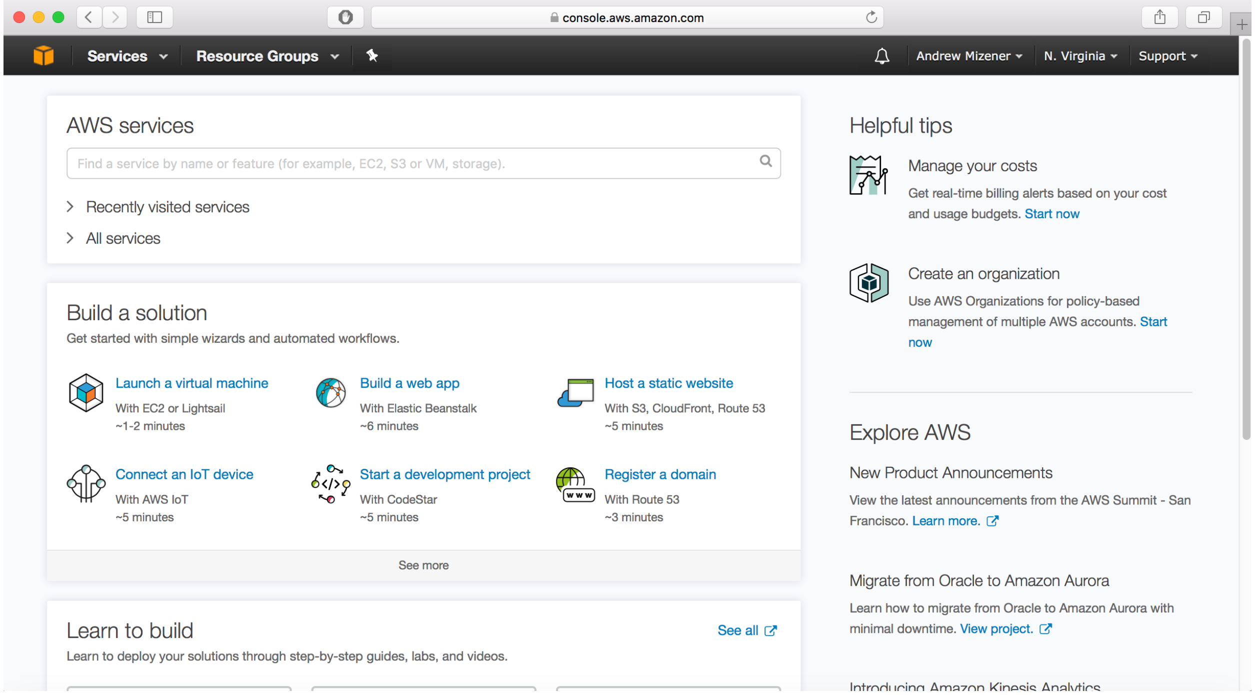
III. Advanced topics: custom AMIs & managing instances

Before we get deep into the process of using EC2 instances with MESA, I want to briefly discuss what EC2 instances are, and where it makes sense to use them. EC2 is Amazon's cloud compute service. Essentially, it allows users to rent time on remote machines, which can be customized to meet individual needs. It's probably not economical to rent an EC2 instance if you're just running one MESA job, especially if it's the kind of project that will take a long time to run. If you project involves running many MESA jobs, particularly less demanding ones, using EC2 instances is much more likely to be a suitable solution.

First Steps: Installing MESA on a single EC2 instance

Step 1: Make an Amazon Web Services account

Go to https://aws.amazon.com and click on the yellow "Sign In to the Console" button on the top right. Fill in your email and the "I am a new user" button. Then, follow the instructions to create an account; you'll need to provide a method of payment and confirm your id with a phone call. When you're done, log in. You should see a webpage that looks like this:



AWS services

Find a service by name or feature (for example, EC2, S3 or VM, storage).

- Recently visited services
- All services

Build a solution

Get started with simple wizards and automated workflows.

Launch a virtual machine
With EC2 or Lightsail
~1-2 minutes

Build a web app
With Elastic Beanstalk
~6 minutes

Host a static website
With S3, CloudFront, Route 53
~5 minutes

Connect an IoT device
With AWS IoT
~5 minutes

Start a development project
With CodeStar
~5 minutes

Register a domain
With Route 53
~3 minutes

See more

Learn to build

See all

Learn to deploy your solutions through step-by-step guides, labs, and videos.

Helpful tips

Manage your costs

Get real-time billing alerts based on your cost and usage budgets. Start now

Create an organization

Use AWS Organizations for policy-based management of multiple AWS accounts. Start now

Explore AWS

New Product Announcements

View the latest announcements from the AWS Summit - San Francisco. Learn more.

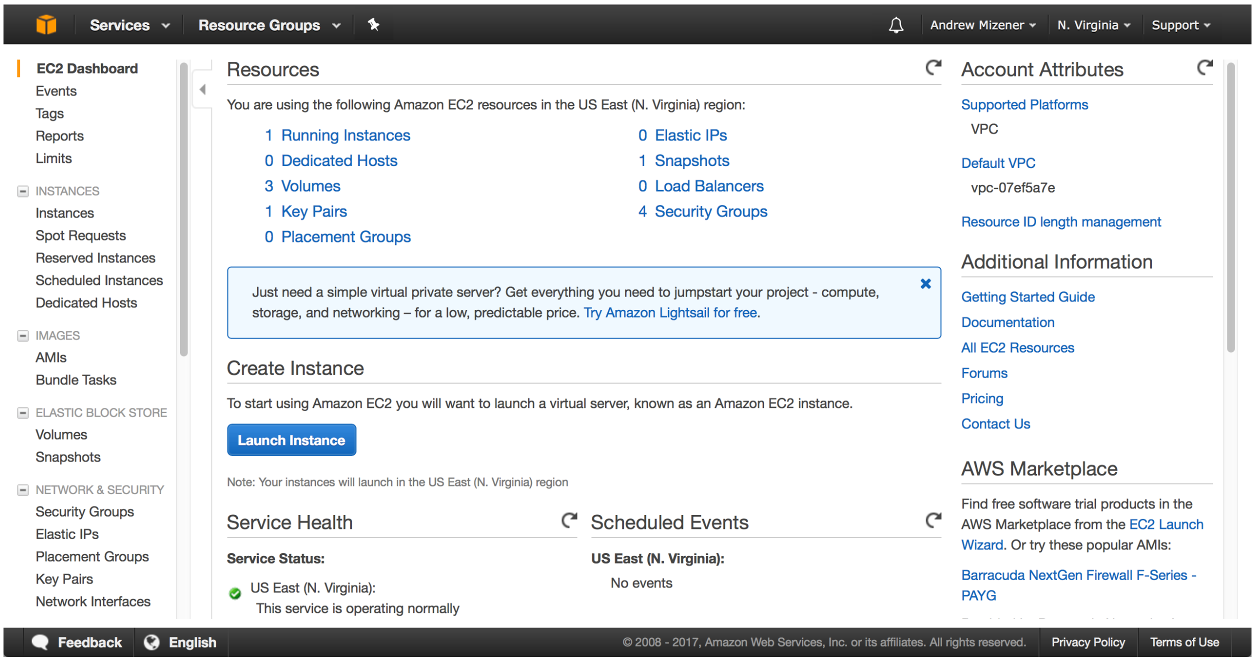
Migrate from Oracle to Amazon Aurora

Learn how to migrate from Oracle to Amazon Aurora with minimal downtime. View project.

Introducing Amazon Kinesis Analytics

Step 2: Create an instance

Click on "Services" on the top left of your window. Under "Compute", you should see an entry titled "EC2". Click on it. Your window should now look similar to this (although without any running instances, volumes, or anything - this is an account I've been using):



Resources

You are using the following Amazon EC2 resources in the US East (N. Virginia) region:

1 Running Instances	0 Elastic IPs
0 Dedicated Hosts	1 Snapshots
3 Volumes	0 Load Balancers
1 Key Pairs	4 Security Groups
0 Placement Groups	

Just need a simple virtual private server? Get everything you need to jumpstart your project - compute, storage, and networking – for a low, predictable price. Try Amazon Lightsail for free.

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 (N. Virginia) region

Service Health

Service Status:

US East (N. Virginia):
This service is operating normally

Scheduled Events

US East (N. Virginia):
No events

Account Attributes

Supported Platforms
VPC

Default VPC
vpc-07ef5a7e

Resource ID length management

Additional Information

Getting Started Guide
Documentation
All EC2 Resources
Forums
Pricing
Contact Us

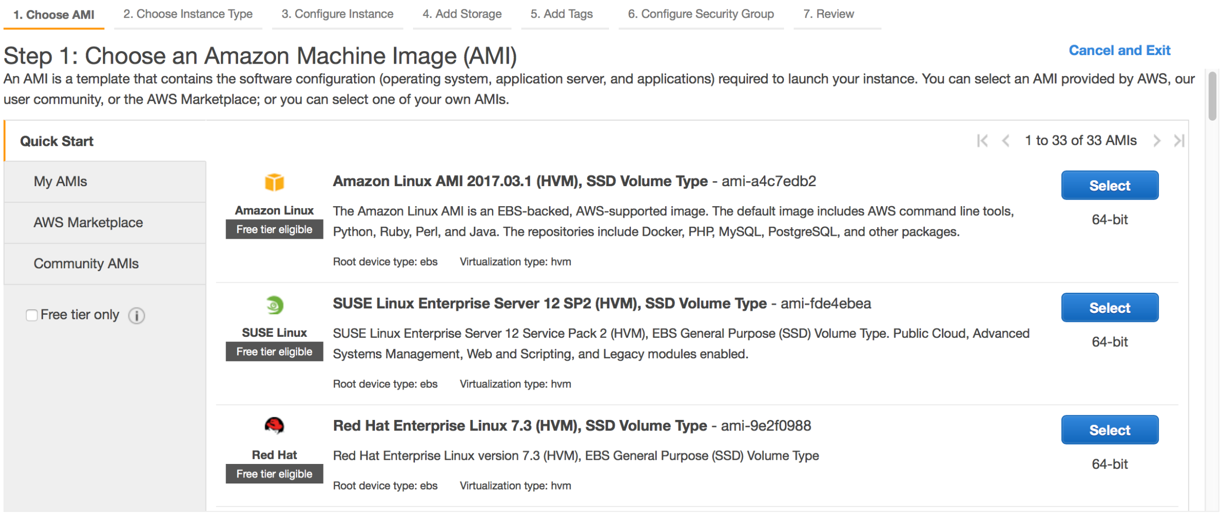
AWS Marketplace

Find free software trial products in the AWS Marketplace from the EC2 Launch Wizard. Or try these popular AMIs:

Barracuda NextGen Firewall F-Series - PAYG

© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Click on the button that says "Launch Instance". You'll be confronted by something that looks like this:



Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Amazon Linux — Free tier eligible

Amazon Linux AMI 2017.03.1 (HVM), SSD Volume Type - ami-a4c7edb2

The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.

Root device type: ebs Virtualization type: hvm

64-bit

SUSE Linux — Free tier eligible

SUSE Linux Enterprise Server 12 SP2 (HVM), SSD Volume Type - ami-fde4ebea

SUSE Linux Enterprise Server 12 Service Pack 2 (HVM), EBS General Purpose (SSD) Volume Type. Public Cloud, Advanced Systems Management, Web and Scripting, and Legacy modules enabled.

Root device type: ebs Virtualization type: hvm

64-bit

Red Hat — Free tier eligible

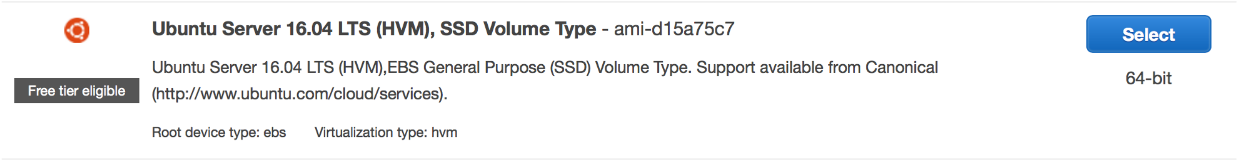
Red Hat Enterprise Linux 7.3 (HVM), SSD Volume Type - ami-9e2f0988

Red Hat Enterprise Linux version 7.3 (HVM), EBS General Purpose (SSD) Volume Type

Root device type: ebs Virtualization type: hvm

64-bit

This is where you can choose an operating system for your instance, packaged as an Amazon Machine Image (AMI). In my experience, the standard Amazon Linux AMI doesn't play nice with MESA. I strongly recommend using the **Ubuntu Server 16.04 AMI** - it's what I use for own MESA-focused instances, and it's what I'll be using in this tutorial.



Ubuntu Server 16.04 LTS (HVM), SSD Volume Type - ami-d15a75c7

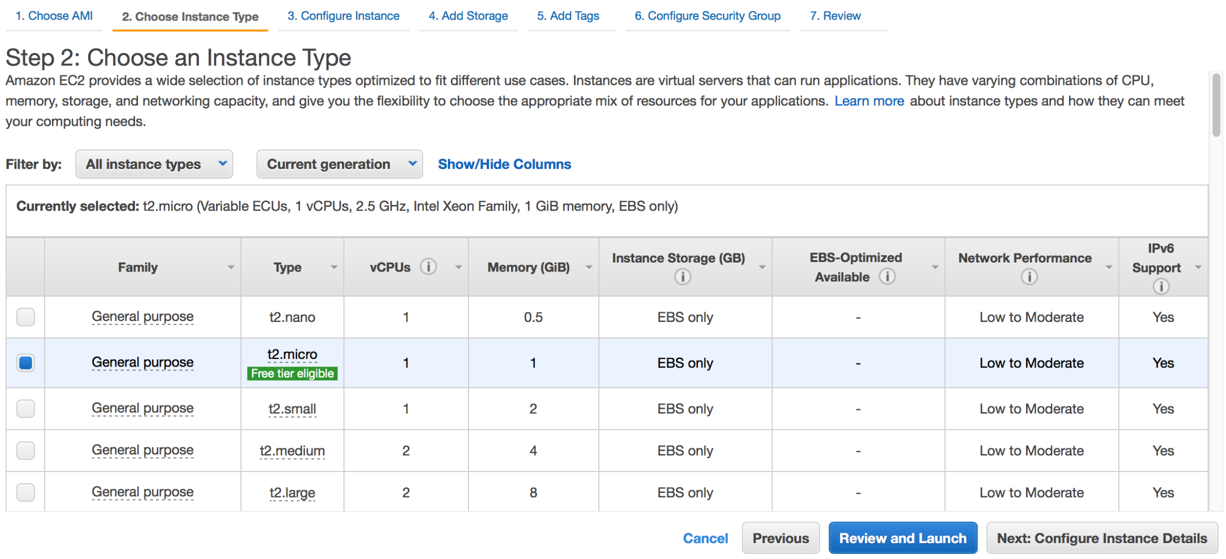
Ubuntu Server 16.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

Free tier eligible

Root device type: ebs Virtualization type: hvm

64-bit

When you've found an AMI you like, click "Select". You'll see a screen that looks like this:



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.

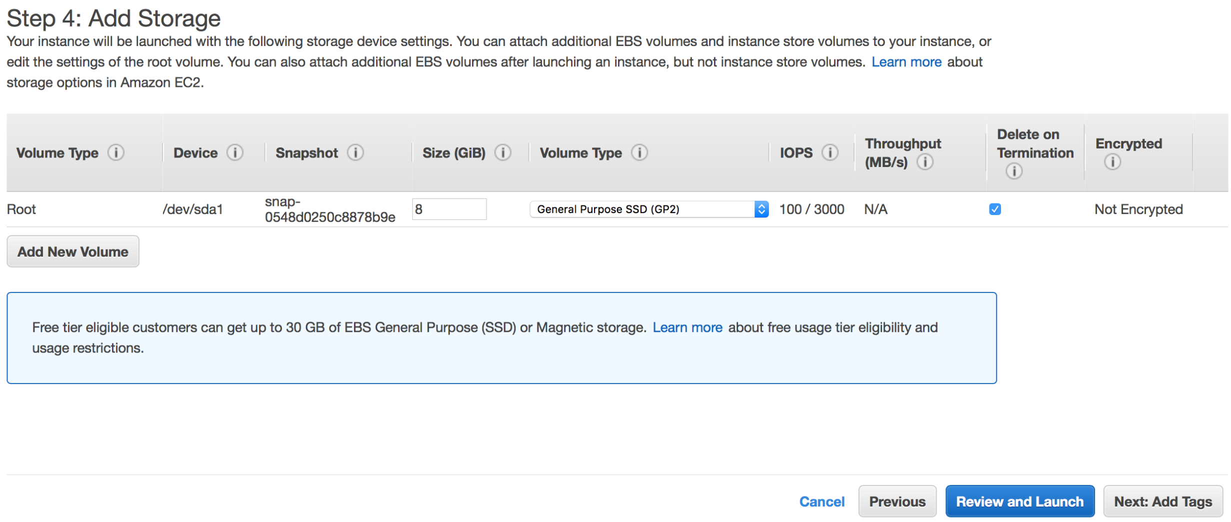
Filter by: All instance types Current generation

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

This is where you choose the type of instance you want to use. Specific costs and detailed hardware information can be found at https://aws.amazon.com/ec2/instance-types/ . For use with MESA, I recommend **c4.2xlarge** instances. To find this, click on the box reading "All instance types" and change to "Compute optimized". **When you've found an instance type that fits your needs, select it and click the box that reads "Next: Configure Instance Details".**

Most of the options on this page aren't important right now, so **click on the box that reads "Next: Add Storage".** You'll see a page that looks like this:



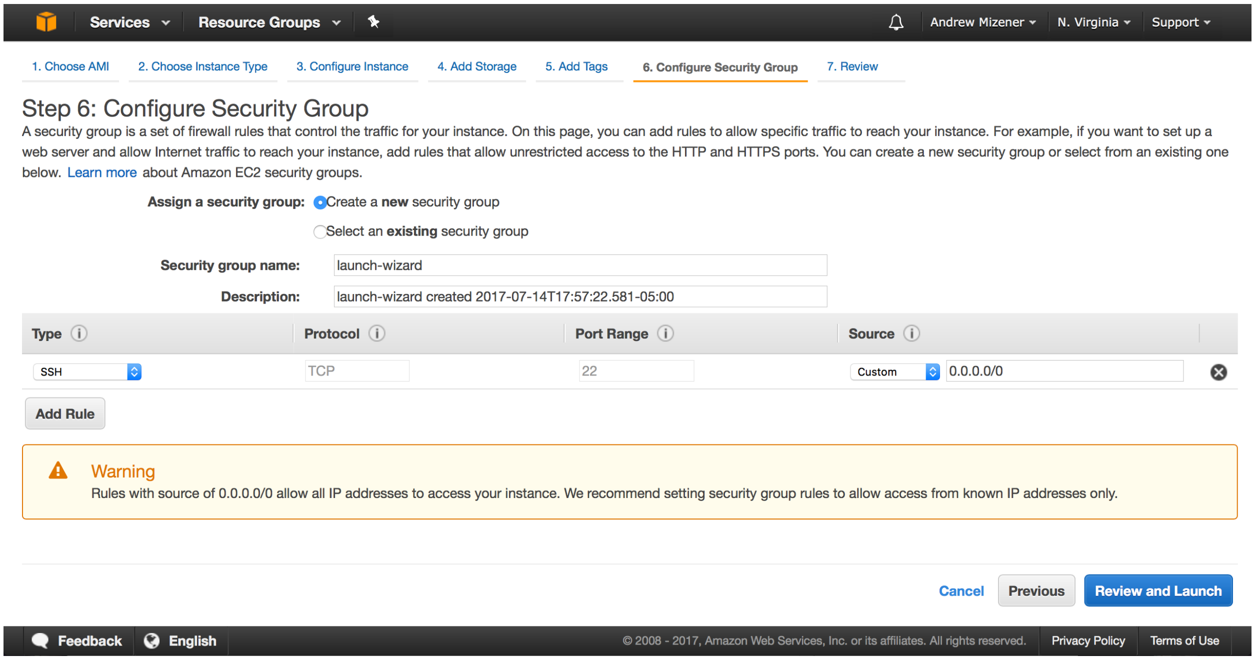
Step 4: Add Storage

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	Encrypted
Root	/dev/sda1	snap-0548d0250c8878b9e	8	General Purpose SSD (GP2)	100 / 3000	N/A	☑	Not Encrypted

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.

Here, you can configure the storage attached to your instance. The default amount of storage is 8 gigabytes, which isn't even enough to set up MESA, let alone run any jobs. **I've had success using 30 gigabyte volumes, but you can use whatever size you like as long as it is large enough to hold MESA**. If you want, you can also attach non-root volumes here. When you're done, click "Add Tags". This is another step that isn't important for our purposes, so **click on "Configure Security Group".** You'll see a page that looks like this:



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 name: launch-wizard

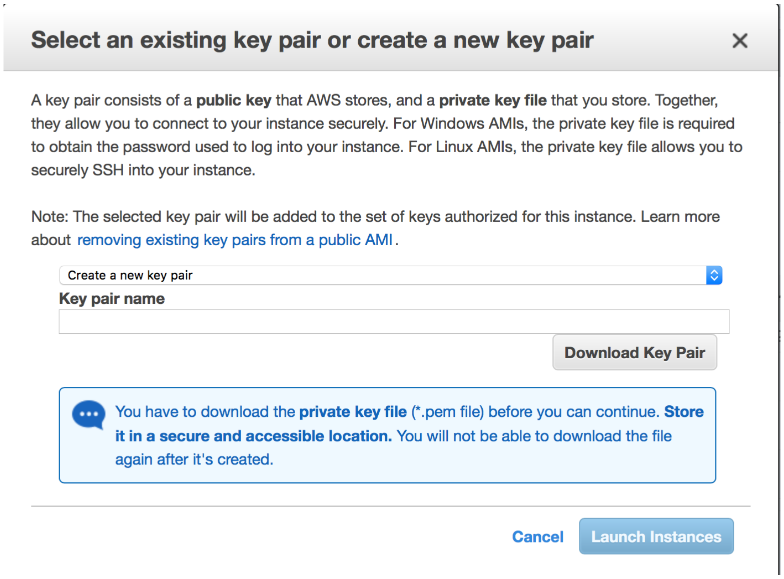
Description: launch-wizard created 2017-07-14T17:57:22.581-05:00

Type	Protocol	Port Range	Source
SSH	TCP	22	Custom 0.0.0.0/0

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.

© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.

This is a very important step. As the page says, a security group is a set of firewall rules that control the traffic for your instance. If your security groups are not set up correctly, you will not be able to connect to your instance. For now, we only need to make one rule. **Simply click on the selection box in the "Source" column and choose "My IP".** This will allow your instance to make SSH connections with your local machine. **Now, click on "Review and Launch".** Here, you can review your choices and ensure that your instance is set up correctly. When you're ready, **click "Launch".** You'll be presented with a pop-up that looks like this:



Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

Create a new key pair

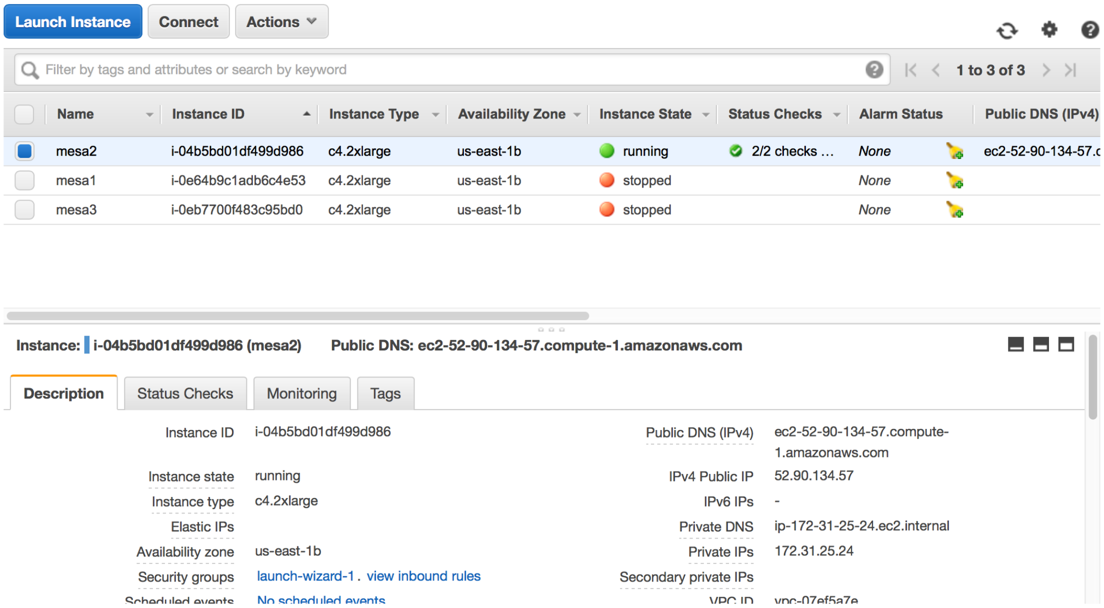
Key pair name

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

Amazon does not allow anyone to connect to their instances with a password. You may only ssh into an instance if you possess its corresponding private key. **So, name your key-pair, download your private key, and put it somewhere safe on your local machine. Then, click "Launch Instances".**

Step 3: Connecting to your instance

After you click "Launch Instances", you'll see a screen that looks similar to this one (although with only one instance; again, I'm currently using my account):

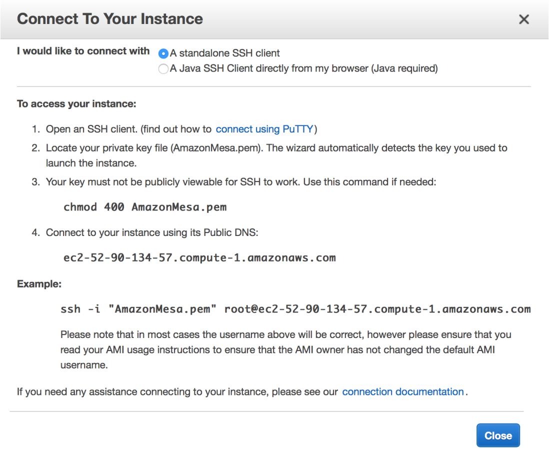


	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
☑	mesa2	i-04b5bd01df499d986	c4.2xlarge	us-east-1b	● running	✓ 2/2 checks ...	None	ec2-52-90-134-57.c
□	mesa1	i-0e64b9c1adb6c4e53	c4.2xlarge	us-east-1b	● stopped		None	
□	mesa3	i-0eb7700f483c95bd0	c4.2xlarge	us-east-1b	● stopped		None	

Instance: i-04b5bd01df499d986 (mesa2)   Public DNS: ec2-52-90-134-57.compute-1.amazonaws.com

Instance ID	i-04b5bd01df499d986	Public DNS (IPv4)	ec2-52-90-134-57.compute-1.amazonaws.com
Instance state	running	IPv4 Public IP	52.90.134.57
Instance type	c4.2xlarge	IPv6 IPs	-
Elastic IPs		Private DNS	ip-172-31-25-24.ec2.internal
Availability zone	us-east-1b	Private IPs	172.31.25.24
Security groups	launch-wizard-1 . view inbound rules	Secondary private IPs	
Scheduled events	No scheduled events	VPC ID	vpc-07ef5a7e

To connect, select your instance and click the "Connect" button. You'll see a screen that looks like this:



Connect To Your Instance

I would like to connect with ● A standalone SSH client
○ A Java SSH Client directly from my browser (Java required)

To access your instance:

1. Open an SSH client. (find out how to connect using PuTTY)
2. Locate your private key file (AmazonMesa.pem). The wizard automatically detects the key you used to launch the instance.
3. Your key must not be publicly viewable for SSH to work. Use this command if needed:

```
chmod 400 AmazonMesa.pem
```

4. Connect to your instance using its Public DNS:

```
ec2-52-90-134-57.compute-1.amazonaws.com
```

Example:

```
ssh -i "AmazonMesa.pem" root@ec2-52-90-134-57.compute-1.amazonaws.com
```

Please note that in most cases the username above will be correct, however please ensure that you read your AMI usage instructions to ensure that the AMI owner has not changed the default AMI username.

If you need any assistance connecting to your instance, please see our connection documentation.

Follow the instructions as given. You may need to navigate to the directory where your private key is located.

Step 4: MESA prerequisites

Congratulations! You've successfully created an Amazon EC2 instance. If you've set up MESA before, feel free to skim this section, but be advised that there are still some irregularities involved. The step here is to connect to your instance via the instructions above. Once you've done so, you'll see a terminal window, just as you would if you ssh'ed into any other remote machine. Before we can install MESA and its libraries, we need to install a few other libraries. **Using *sudo apt-get install <package>*, install the following packages, in order:**

```
xauth
xorg
make
perl
libx11-dev
libx11-6
zlib1g
zlib1g-dev
tcsh
subversion
```

At this point, I would also recommend that you install the*feh* remote image display package, and a text editor such as *emacs* or *nedit*. **Once you've installed these packages, exit your instance and log back in.** This time, include the *-Y* flag when you're ssh-ing in to your instance. This will enable x11 forwarding. Now, you'll need to install and set up the MESA SDK. **Download the most recent SDK from here: http://www.astro.wisc.edu/%7Etownsend/static.php?ref=mesasdk#Download to your local machine and open a local terminal window. From here, *scp* the file over to your instance.** Because we're connecting to a machine that requires a private key, the syntax used here is a little different. Specifically, your *scp* command should look like this:

```
scp -r -i /path-to-public-key/public_key.pem /path-to-file/
ubuntu@ipv6_address:~/destination_path/
```

Once the SDK folder is copied to your instance, decompress it if you haven't yet done so. Enter *cd* with no arguments to navigate to your home directory, and enter *ls -a.* **You should see a file entitled *.profile.* Open it with the text editor of your choice, and add the following lines:**

```
export MESASDK_ROOT=~/mesasdk
source $MESASDK_ROOT/bin/mesasdk_init.sh
```

You should now be ready to install the MESA version of your choice. If you want to first check that your SDK is installed properly, you can enter the command g*fortran --version.* Upon doing so, the first line of output should look something like: GNU Fortran (GCC) 5.3.1 20160129. If it doesn't, check the SDK website's troubleshooting section and ensure you haven't missed any steps.

Step 5: Installing MESA

You should now be prepared to install the MESA version of your choice. The easiest way to do this is though the *subversion* package, which you should have installed in the previous step. **From your instance, enter the following command:**

```
svn co -r 9793 svn://svn.code.sf.net/p/mesa/code/trunk mesa-
r9793
```

This specific command would download the source files for version 9793 into a file named mesa-r9793. If you want to install a different version of MESA, simply change the numbers to the version number of MESA you desire. **Then, cd back to your home folder and edit ./profile. Add the following two lines:**

```
export MESA_DIR=/home/ubuntu/mesa-r8845
export OMP_NUM_THREADS=8
```

This assumes that you're using an c4.2xl instance (which has 8 cores) and that you're putting your MESA install in your home folder. **If you're using a different type of instance, or you want to install MESA somewhere else, you'll need to change these lines appropriately.** Once you save the changes, exit out of the instance and log back in so that you start a new shell session. All you need to do now is:

```
cd $MESA_DIR
./install
```

The rest of the installation process should run itself. Once it's done (which might take a while - this is a big package), you should see something like:

```
************************************************
************************************************
************************************************
mesa installation was successful.
************************************************
************************************************
************************************************
```

If you see this, congratulations! You've installed MESA on an Amazon EC2 instance! If not, check MESA's troubleshooting page at http://mesa.sourceforge.net/troubleshooting.html.

Advanced topics: custom AMIs & managing instances

Part 1: Using MESA on an EC2 instance

I’ve found that there a few tricks you need to know to successfully run MESA on an EC2 instance. In general, I've found that my work is easier and more cost-effective when I prepare all the work folders I want on my local machine, and copy them to my instances when I want to actually run them. This lets me get away with attaching smaller, less expensive storage volumes to my instances as well as giving me the luxury of building my projects in an easier-to-use environment. **If you run your project this way, make sure you ./clean and ./mk inside each workfile once it’s been copied to your instance.** If you run these commands in your local machine but not the instance, you’re likely to see some very weird errors.

When you’re done using an instance and wish to cease being billed, you can shut it down using:

```
sudo shutdown -h
```

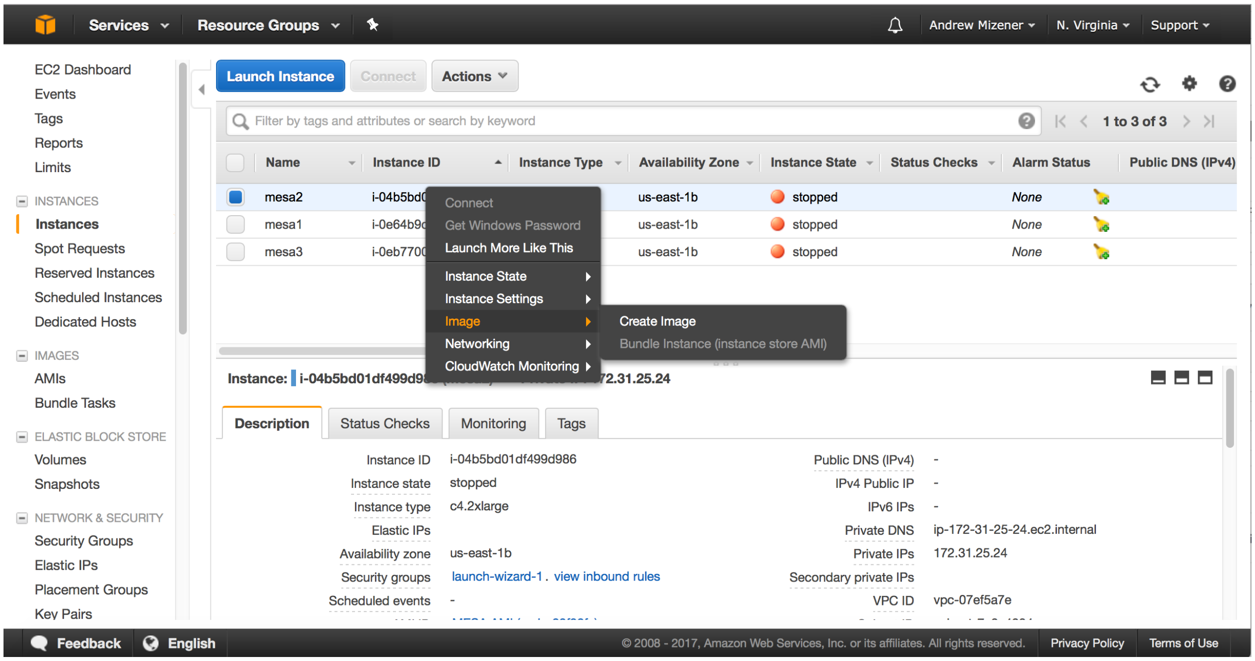
This will retain everything stored on the instance. If you wish to start it back up again, navigate to your EC2 Management Console on the AWS website. Near the top of your screen, you will see a button labeled “Actions”. Once you click on this, you will see a drop-down menu with an option labeled “Instance State”. **To turn on an instance, navigate here and click the option “Start”. You can also shut down an instance from this menu with the command labeled “Stop”. If you wish to permanently delete an instance and all of its memory, you can use the “Terminate” command.**

It’s also possible for pgstar to behave oddly with EC2 instances, especially when you’re trying to monitor runs on multiple instances on the same time. **To simplify matters, I disable active pgstar windows when I’m working with EC2 instances, and instead monitor my runs by looking at the .png images pgstar saves.** I prefer the *feh* package for this, but you can use the application of your choice.

It’s also important to understand how Amazon handles billing for these instances. You are billed for a minimum of one full hour of compute time every time an instance is turned on, even if it is on for less than an hour. For each hour after the first, you are also billed at the hourly rate advertised for your instance type and region. Additionally, be aware that Amazon bills your account for storage if the total amount of storage you use per month exceeds 30gb. Storage is considered “used” if it’s attached to an instance, regardless of whether or not it is empty. So, I would recommend you terminate any unused instances as soon as you’re done with them.

Part 2: Making custom AMIs and running multiple instances

Setting up MESA individually on many instances can be time consuming and a waste of resources. A more appropriate course of action is to use Amazon's system to make a custom image from which an arbitrary number of identical images can be built. This process is actually pretty simple. To begin, go to the AWS website and navigate to your EC2 Management Console. Find the instance you want to duplicate, and right click on it. Make sure it's been shut down before you do this. Scroll down to "Image". Your window should look something like this:

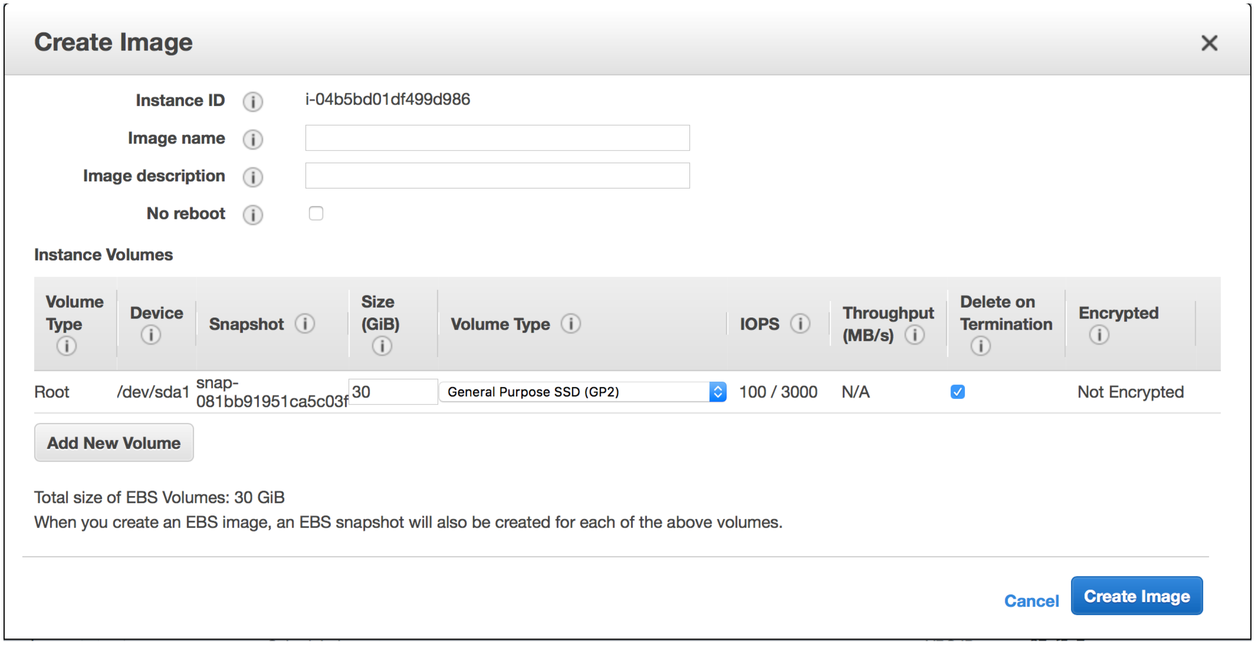


Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
■ mesa2	i-04b5bd0		us-east-1b	● stopped		None	
□ mesa1	i-0e64b9c		us-east-1b	● stopped		None	
□ mesa3	i-0eb7700		us-east-1b	● stopped		None	

Instance: i-04b5bd01df499d9 … 72.31.25.24

Instance ID	i-04b5bd01df499d986	Public DNS (IPv4)	-
Instance state	stopped	IPv4 Public IP	-
Instance type	c4.2xlarge	IPv6 IPs	-
Elastic IPs		Private DNS	ip-172-31-25-24.ec2.internal
Availability zone	us-east-1b	Private IPs	172.31.25.24
Security groups	launch-wizard-1 . view inbound rules	Secondary private IPs	
Scheduled events	-	VPC ID	vpc-07ef5a7e

Click on "Create image". A window will pop up that looks like this:



Instance ID i-04b5bd01df499d986

Image name

Image description

No reboot □

Instance Volumes

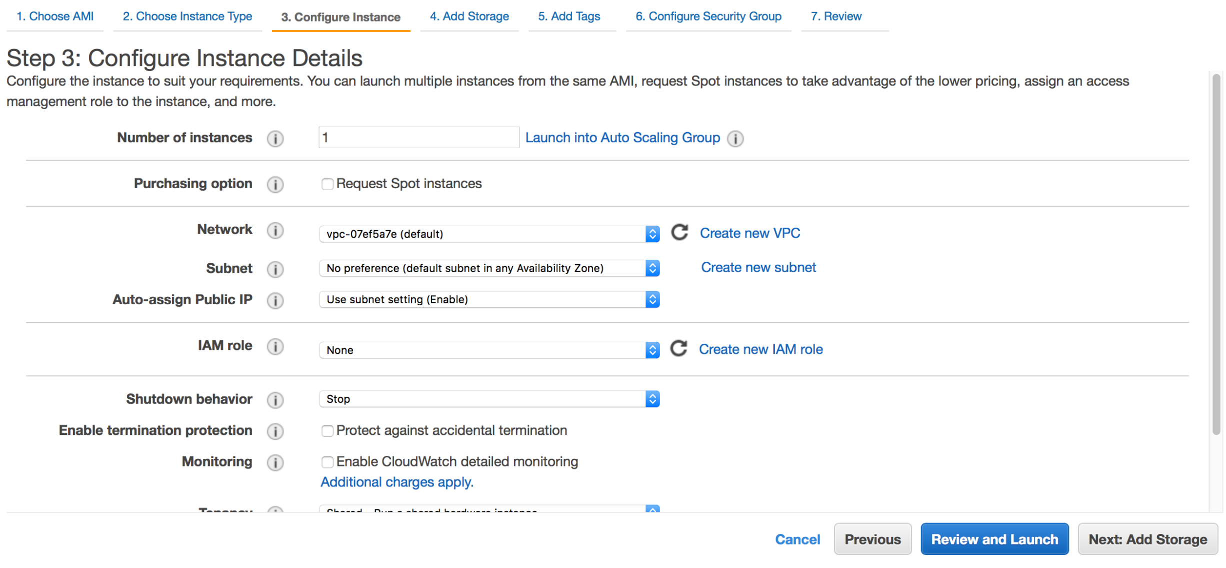
Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/sda1	snap-081bb91951ca5c03f	30	General Purpose SSD (GP2)	100 / 3000	N/A	☑	Not Encrypted

Total size of EBS Volumes: 30 GiB

When you create an EBS image, an EBS snapshot will also be created for each of the above volumes.

Enter a name and description for your image. You should also specify a volume size for your image. The minimum allowed size for your image is equal to the size of the volume attached to your instance. Once you're happy with your setup, click "Create Image".

Now, you can create instances based of this image. Go back to the EC2 Management Console and click "Launch Instance". Instead of selecting a pre-made AMI, click on "My AMIs". You should see the image you've just created! Select this and generate the instance just as we did before. You can even use the same security group and key pair as before if you want to make access even easier. If you want to launch multiple instances at the same time, make your way to the "Configure Instance Details" step. It will look like this:



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-07ef5a7e (default) — Create new VPC

Subnet: No preference (default subnet in any Availability Zone) — Create new subnet

Auto-assign Public IP: Use subnet setting (Enable)

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

You can choose how many instances you want to launch in the "Number of Instances" group, if you want to launch multiple identical instances at the same time

Thank you for reading this guide! I hope it's been helpful and complete. Please forgive any mistakes; I'm just an undergraduate. If you do find any, let me know via the email on the cover of this document.