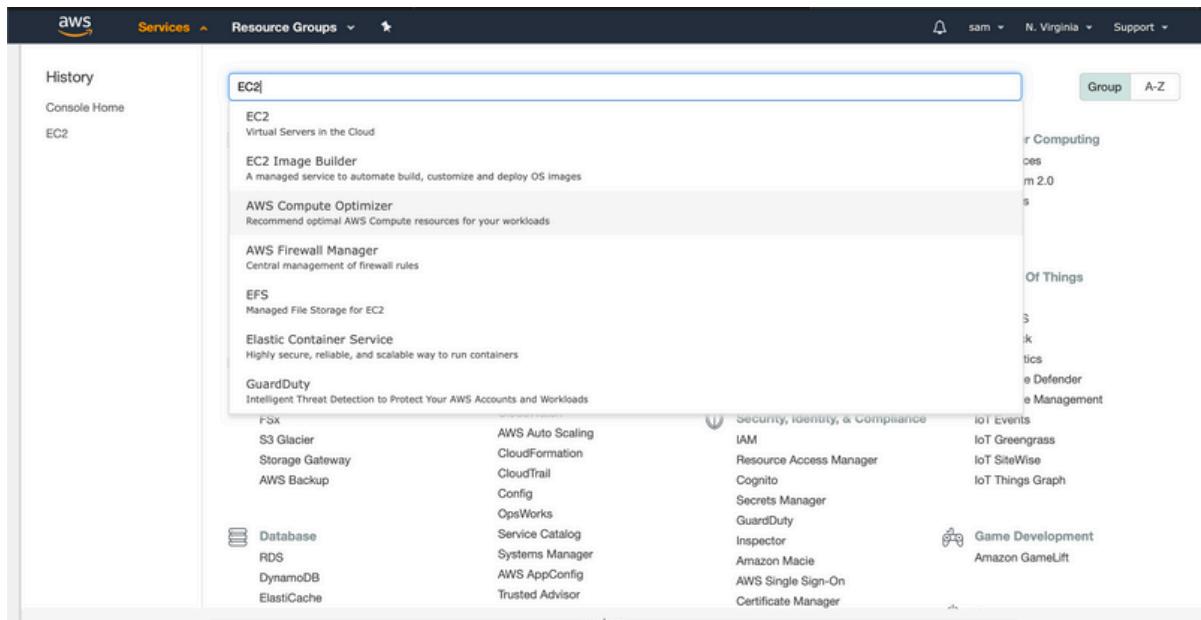


# Hands-On Study Notes

Andrew Brown

## How to launch a server using EC2

- 1.1 [ ] Go to services and type EC2 and we will make our way over to the EC2 console



- 1.2 [ ] Scroll down and click on **Launch Instance**

## 2 AWS Certified Cloud Practitioner

The screenshot shows the AWS EC2 Launch Instance page. On the left, there's a sidebar with navigation links for EC2 Dashboard, Events, Tags, Reports, Limits, Instances (with sub-links for Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations), Images (AMIs, Bundle Tasks), and Elastic Block Store (Volumes). The main content area has sections for 'Launch instance', 'Scheduled events', 'Migrate a machine', and 'Quick ID filter'. The 'Launch instance' section is active, showing options to 'Launch Instance' or 'Launch instance from template' in the 'US East (N. Virginia) Region'. To the right, there are promotional banners for 'Save 10% with AMD EPYC-Powered Instances', 'Save up to 90% on EC2 with Spot Instances', and 'Easily launch third-party AMI products'. At the bottom right, there's a 'Additional information' section with links to Getting started guide, Documentation, All EC2 resources, Forums, and Pricing.

## Choose AMI

We will be presented with a bunch of options to configure our server and we will choose what OS we want to use

- 1.3 [ ] We will click on **Amazon Linux** because it's part of the Free tier

The screenshot shows the 'Step 1: Choose an Amazon Machine Image (AMI)' screen. At the top, there are tabs for 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage, 5. Add Tags, 6. Configure Security Group, and 7. Review. Below the tabs, there's a search bar with placeholder text 'Search for an AMI by entering a search term e.g. "Windows"'. A note says '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.' On the left, there's a 'Quick Start' sidebar with buttons for 'My AMIs', 'AWS Marketplace', 'Community AMIs', and a checked 'Free tier only' checkbox. The main area lists several AMI options:

- Amazon Linux 2 AMI (HVM), SSD Volume Type** - ami-0323c3dd2da7fb37d (64-bit x86) / ami-0ce2e5b7d27317779 (64-bit Arm)  
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, Gilbo 2.26, Binutils 2.29.1, and the latest software packages through extras.  
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes  
Select button (radio buttons for 64-bit (x86) and 64-bit (Arm))
- Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type** - ami-0915e09cc7ceee3ab (64-bit x86)  
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 ENA Enabled: Yes  
Select button (radio buttons for 64-bit (x86) and 64-bit (Arm))
- Red Hat Enterprise Linux 8 (HVM), SSD Volume Type** - ami-098f16afa9edf40be (64-bit x86) / ami-029ba835ddd43c34f (64-bit Arm)  
Red Hat Enterprise Linux version 8 (HVM), EBS General Purpose (SSD) Volume Type  
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes  
Select button (radio buttons for 64-bit (x86) and 64-bit (Arm))
- SUSE Linux Enterprise Server 15 SP1 (HVM), SSD Volume Type** - ami-0068cd63259e9f24c (64-bit x86) / ami-05dde7e9c924be7dc (64-bit Arm)  
SUSE Linux Enterprise Server 15 SP1 (HVM), EBS General Purpose (SSD) Volume Type  
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes  
Select button (radio buttons for 64-bit (x86) and 64-bit (Arm))

## Choose an Instance Type

We will choose the size of our server and these are called instance types

- 1.4 [ ] Click on **t2.micro** because it is part of Free tier and click **Next: Configure Instance Details**

The screenshot shows the 'Step 2: Choose an Instance Type' page. At the top, there are tabs: 1. Choose AMI, 2. Choose Instance Type (which is highlighted in orange), 3. Configure Instance, 4. Add Storage, 5. Add Tags, 6. Configure Security Group, and 7. Review. Below the tabs, there's a note: '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.' There are filters: 'Filter by: All instance types' and 'Current generation'. A table lists various instance types:

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	General purpose	t2.micro	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.xlarge	4	16	EBS only	-	Moderate	Yes
<input type="checkbox"/>	General purpose	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
<input type="checkbox"/>	General purpose	t3a.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes

At the bottom, there are buttons: 'Cancel', 'Previous', 'Review and Launch' (which is highlighted in blue), and 'Next: Configure Instance Details'.

## Configure Instance Details

Now going to Instance details we can choose how many instances we want to start

- 1.5 [ ] Right click on **IAM role** and make a new tab

## 4 AWS Certified Cloud Practitioner

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-aa0113d0 (default) Create new VPC

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

Stop - Hibernate behavior: Enable hibernation as an additional stop behavior

Enable termination protection: Protect against accidental termination

Monitoring: Enable CloudWatch detailed monitoring Additional charges apply.

Buttons: Previous, Review and Launch, Next: Add Storage

## Create a role

- 1.6 [ ] Go to IAM management console go down and click on **create a new role**

Identity and Access Management (IAM)

Roles

What are IAM roles?

IAM roles are a secure way to grant permissions to entities that you trust. Examples of entities include the following:

- IAM user in another account
- Application code running on an EC2 instance that needs to perform actions on AWS resources
- An AWS service that needs to act on resources in your account to provide its features
- Users from a corporate directory who use identity federation with SAML

IAM roles issue keys that are valid for short durations, making them a more secure way to grant access.

Additional resources:

- IAM Roles FAQ
- IAM Roles Documentation
- Tutorial: Setting Up Cross Account Access
- Common Scenarios for Roles

Create role Delete role

Role name	Trusted entities	Last activity
AWSServiceRoleForSupport	AWS service: support (Service-Linked role)	None
AWSServiceRoleForTrustedAdvisor	AWS service: trustedadvisor (Service-Linked role)	None

- 1.7 [ ] Click on **EC2** and click **Next: Permissions**

Create role

Select type of trusted entity

- AWS service EC2, Lambda and others
- Another AWS account Belonging to you or 3rd party
- Web identity Cognito or any OpenID provider
- SAML 2.0 federation Your corporate directory

Allows AWS services to perform actions on your behalf. Learn more

Choose a use case

Common use cases

**EC2**  
Allows EC2 instances to call AWS services on your behalf.

**Lambda**  
Allows Lambda functions to call AWS services on your behalf.

Or select a service to view its use cases

API Gateway	CodeDeploy	EKS	IoT Things Graph	Rekognition
AWS Backup	CodeGuru	EMR	KMS	RoboMaker
AWS Chatbot	CodeStar Notifications	ElastiCache	Kinesis	S3
AWS Support	Comprehend	Elastic Beanstalk	Lake Formation	SMS
Amplify	Config	Elastic Container Service	Lambda	SNS

\* Required      Cancel      **Next: Permissions**

- 1.8 [ ] Type in ssm (simple systems manager which will be to log into that machine) and check mark **AmazonEc2RoleforSSM** and click **Next: Tags** and click **Next review**

Create role

▼ Attach permissions policies

Choose one or more policies to attach to your new role.

**Create policy**      **Filter policies**  Showing 15 results

	Policy name	Used as
<input type="checkbox"/>	AmazonEc2RoleforSSM	None
<input type="checkbox"/>	AmazonSSMAutomationApproverAccess	None
<input type="checkbox"/>	AmazonSSMAutomationRole	None
<input type="checkbox"/>	AmazonSSMDirectoryServiceAccess	None
<input type="checkbox"/>	AmazonSSMFullAccess	None
<input type="checkbox"/>	AmazonSSMMaintenanceWindowRole	None
<input type="checkbox"/>	AmazonSSMManagedInstanceCore	None
<input type="checkbox"/>	AmazonSSMPatchAssociation	None

▶ Set permissions boundary

\* Required      Cancel      Previous      **Next: Tags**

## 6 AWS Certified Cloud Practitioner

The screenshot shows the 'Create role' wizard at Step 3: 'Add tags (optional)'. It includes a table for adding IAM tags, a note about tag limits, and navigation buttons.

Key	Value (optional)	Remove
Add new key		

You can add 50 more tags.

Cancel Previous Next: Review

- 1.9 [ ] Type in **MyEc2Role** in Role name and click **Create role**. Now that role has been created, we will just go ahead and close that tab

The screenshot shows the 'Create role' wizard at Step 4: 'Review'. It displays the role information entered in Step 3, including the role name, description, trusted entities, policies, and permissions boundary.

Provide the required information below and review this role before you create it.

**Role name\*** MyEC2Role  
Use alphanumeric and '+,-@-' characters. Maximum 64 characters.

**Role description** Allows EC2 instances to call AWS services on your behalf.  
Maximum 1000 characters. Use alphanumeric and '+,-@-' characters.

**Trusted entities** AWS service: ec2.amazonaws.com

**Policies** AmazonEC2RoleforSSM

**Permissions boundary** Permissions boundary is not set

No tags were added.

\* Required Cancel Previous Create role

The screenshot shows the AWS Identity and Access Management (IAM) service. On the left, there's a navigation sidebar with various options like Dashboard, Access management, Access reports, and Credential report. The 'Roles' section is currently selected. In the main content area, a success message box at the top says 'The role MyEC2Role has been created.' Below it is a table listing roles. The table has columns for 'Role name', 'Trusted entities', and 'Last activity'. Three roles are listed: 'AWSServiceRoleForSupport' (AWS service: support (Service-Linked role)), 'AWSServiceRoleForTrustedAdvisor' (AWS service: trustedadvisor (Service-Linked ...)), and 'MyEC2Role' (AWS service: ec2). All three have 'None' under 'Last activity'.

## Configure Instance

- 1.10 [] Go to IAM role and click **refresh and choose MyEC2Role** and now we will leave everything else blank and click **Next: Add Storage**

The screenshot shows the 'Step 3: Configure Instance Details' page in the AWS CloudFormation console. The top navigation bar includes 'Services', 'Resource Groups', and tabs for '1. Choose AMI', '2. Choose Instance Type', '3. Configure Instance', '4. Add Storage', '5. Add Tags', '6. Configure Security Group', and '7. Review'. The '3. Configure Instance' tab is active. The page displays configuration fields for launching instances, including 'Number of instances' (set to 1), 'Purchasing option' (Request Spot instances), 'Network' (vpc-aa0113d0 (default)), 'Subnet' (No preference (default subnet in any Availability Zone)), 'Auto-assign Public IP' (Use subnet setting (Enable)), 'Placement group' (Add instance to placement group), 'Capacity Reservation' (Open), and 'IAM role' (MyEC2Role, which is highlighted in a dropdown menu). At the bottom, there are buttons for 'Cancel', 'Previous', 'Review and Launch', and 'Next: Add Storage'.

## Add Storage

- 1.11 [] Now you can choose your storage we will leave it at 8GB and we will stick with **General Purpose SSD** for Volume type and click **Review and Launch**

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/xvda	snap-0e1167baa50e9c0ff	8	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	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.

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

## Review

- 1.12 [] Click **Launch** It will ask you to create a key pair and click the **drop down box and select Proceed without a key pair**

Check mark `I acknowledge that I will not be able to connect to this instance unless I already know the password built into this AMI` and Click `Launch`

**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.

**AMI Details**

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0323c3dd2da7fb37d

**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**

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**

Security group name: launch-wizard-1  
Description: launch-wizard-1 created 2020-05-20T09:14:42.468-04:00

Type	Protocol	Port Range	Source	Description
This security group has no rules				

**Buttons:** Cancel, Previous, **Launch**

**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.

**AMI Details**

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0323c3dd2da7fb37d

**Free tier eligible** Amazon Linux 2 comes with five years support packages through extras.

Root Device Type: ebs Virtualization type: hvm

**Instance Type**

Instance Type	ECUs	vCPUs
t2.micro	Variable	1

**Security Groups**

Security group name: launch-wizard-1  
Description: launch-wizard-1 created 2020-05-20T09:14:42.468-04:00

Type	Protocol
------	----------

**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](#).

Choose an existing key pair  
 Create a new key pair  
 Proceed without a key pair

**No key pairs found**  
You don't have any key pairs. Please create a new key pair by selecting the [Create a new key pair](#) option above to continue.

**Buttons:** Cancel, **Launch Instances**, Previous, Launch

The screenshot shows the AWS Step 7: Review Instance Launch wizard. The main page displays instance configuration details: AMI (Amazon Linux 2 AMI (HVM), SSD Volume Type), Instance Type (t2.micro), and Security Groups (launch-wizard-1). A modal dialog box titled "Select an existing key pair or create a new key pair" is open, prompting the user to choose an existing key pair or create a new one. It includes a note about key pairs, a checkbox for acknowledging the inability to connect without a key pair, and two buttons: "Cancel" and "Launch Instances". The "Launch Instances" button is highlighted in blue.

## Launch Status

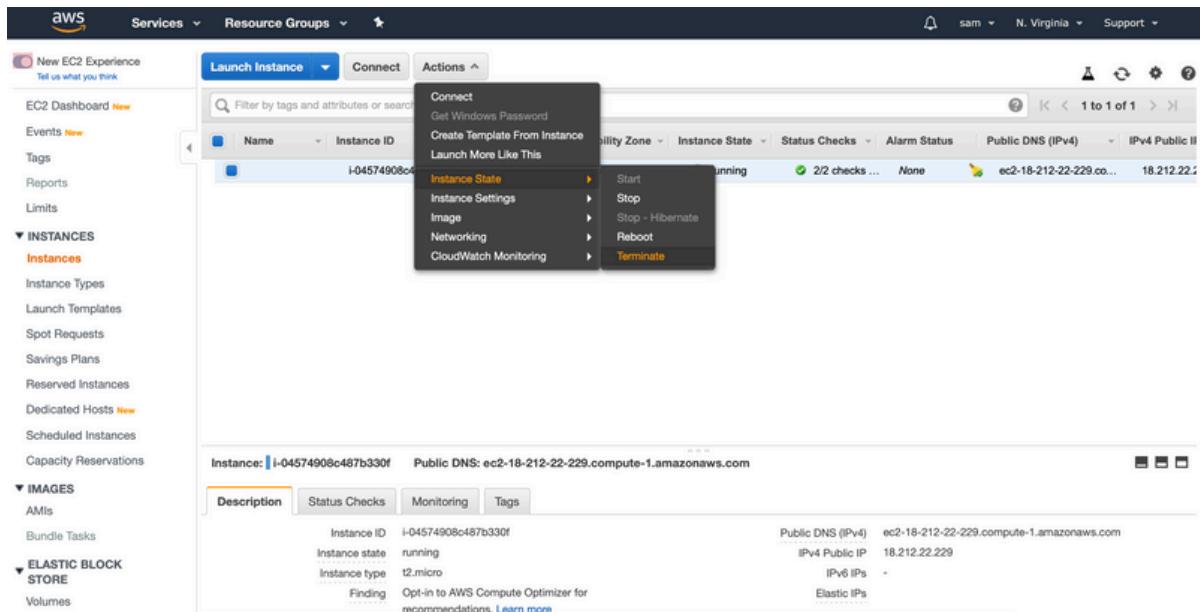
- 1.13 [] Click **View Instances** to be able to view it and now this instance is launching and you will see it in pending state (It will turn from yellow to green and we will wait for it to initialize)

The screenshot shows the AWS Launch Status page. It displays a message: "Your instances are now launching" with a green checkmark icon. Below this, a link to "View launch log" is shown. Another message box says "Get notified of estimated charges" with a blue info icon, and a link to "Create billing alerts". The page also contains sections on "How to connect to your instances" and "Helpful resources". At the bottom right, there is a "View Instances" button.

- 1.14 [ ] After a 3 - 4 min wait the server is now running and it will have 2 checks which means the server is in good shape

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links like 'New EC2 Experience', 'EC2 Dashboard', 'Events', 'Tags', 'Reports', 'Limits', 'INSTANCES' (selected), 'Instances' (selected), 'Instance Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts', 'Scheduled Instances', 'Capacity Reservations', 'IMAGES', 'AMIs', 'Bundle Tasks', 'ELASTIC BLOCK STORE', and 'Volumes'. The main content area has tabs for 'Launch Instance', 'Connect', and 'Actions'. A search bar at the top says 'Filter by tags and attributes or search by keyword'. Below it is a table with columns: Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, Public DNS (IPv4), and IPv4 Public IP. One row is shown: 'i-04574908c487b330f' (Instance ID), 't2.micro' (Instance Type), 'us-east-1a' (Availability Zone), 'running' (Instance State), '2/2 checks ...' (Status Checks), 'None' (Alarm Status), 'ec2-18-212-22-229.compute-1.amazonaws.com' (Public DNS), and '18.212.22.229' (IPv4 Public IP). At the bottom of the instance card, there are tabs for 'Description' (selected), 'Status Checks', 'Monitoring', and 'Tags'. The 'Description' tab shows details: Instance ID (i-04574908c487b330f), Instance state (running), Instance type (t2.micro), Finding (Opt-in to AWS Compute Optimizer for recommendations. Learn more), Public DNS (IPv4) (ec2-18-212-22-229.compute-1.amazonaws.com), IPv4 Public IP (18.212.22.229), IPv6 IPs (-), and Elastic IPs.

*Were not going to shut the server down just yet because we chose the free tier but if you were paying for it you would want to shut it down when your done with it. To shut the server down you would click on Actions, Instance State and Terminate (you can also click stop and that would just stop the server and not destroy it)*



In the next video we will learn how to get access to this instance

## Sessions Manager

There are a couple of ways we can get into this instance. One way is using ssh so if we had created that key pair we could have used it to get into that server or we can use ssm (simple systems manager) and AWS's recommended way.

- 1.1 [] Before we go over to ssm right-click on **Instance ID** and click on **Connect** and click **Close**

EC2 Dashboard [New](#)

Events [New](#)

Tags

Reports

Limits

**INSTANCES**

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts [New](#)

Scheduled Instances

Capacity Reservations

**IMAGES**

AMIs

Bundle Tasks

**ELASTIC BLOCK STORE**

Volumes

**Launch Instance** Connect Actions

Name: i-04574908c487b330f Instance ID: i-04574908c487b330f Instance Type: t2.micro Availability Zone: us-east-1a Instance State: running Status Checks: 2/2 checks passing Alarm Status: None Public DNS (IPv4): ec2-18-212-22-229.co... IPv4 Public IP: 18.212.22.229

Tell us what you think

EC2 Dashboard [New](#)

Events [New](#)

Tags

Reports

Limits

**INSTANCES**

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts [New](#)

Scheduled Instances

Capacity Reservations

**IMAGES**

AMIs

Bundle Tasks

**ELASTIC BLOCK STORE**

Volumes

**Launch Instance**

Connection method:  A standalone SSH client [\(i\)](#)  Session Manager [\(i\)](#)  EC2 Instance Connect (browser-based SSH connection) [\(i\)](#)

**Instance is not associated with a key pair**

This instance is not associated with a key pair. Without a key pair, you can't connect to the instance through SSH.

You can connect using EC2 Instance Connect with just a valid username. You can connect using Session Manager if you have been granted the necessary permissions.

To access your instance:

1. Open an SSH client. (find out how to [connect using PuTTY](#))
2. Locate your private key file (.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 .pem
4. Connect to your instance using its Public DNS:  
ec2-18-212-22-229.compute-1.amazonaws.com

Example:

```
ssh -i ".pem" ec2-user@ec2-18-212-22-229.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.

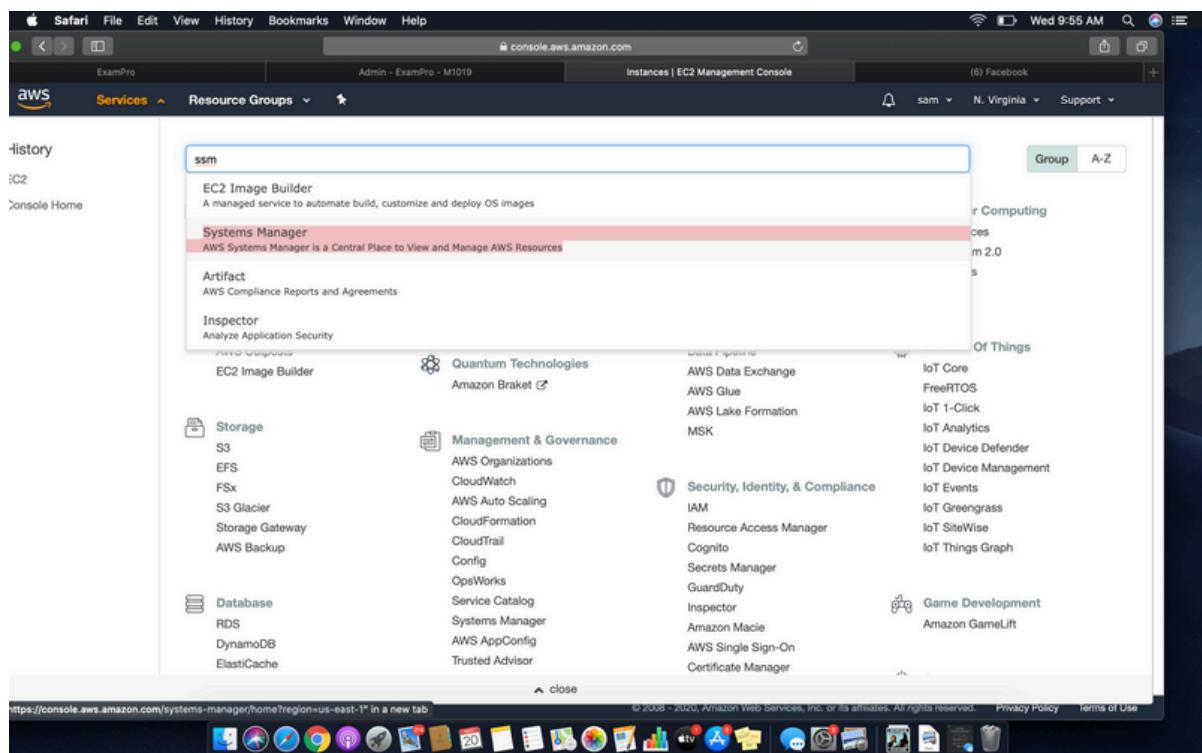
If you need any assistance connecting to your instance, please see our [connection documentation](#).

**Close**

## Start a session

- 1.2 [] Go to **Services** at the top and type in **ssm** and click on **Systems Manager** on the left-hand side, once on that page scroll down on the left-hand side and click on **Session Manager** and click on the **Start Session** orange

button



**AWS Systems Manager**

Management

## Session Manager

Quickly and securely access your Windows and Linux instances

Session Manager is a managed service that provides you with one-click secure access to your instances without the need to open inbound ports and manage bastion hosts. You have centralized access control over who can access your instances and full auditing capabilities to ensure compliance with corporate policies.

**How it works**

- 1 Configure your instances to use Session Manager
- 2 Assign user IAM policies to control instance access
- 3 Specify account options for session logs
- 4 Start a session on your instances by launching bash or shell terminal

**Getting started**

- What is Session Manager?
- Set up Session Manager
- Set up session logging
- Set up session notifications
- Create and manage sessions

## We have our Instance

- 1.3 [ ] Select the **Instance** and click **Start Session**
  
- 1.4 [ ] It logs into the root user not the EC2 user, so we have to type in **sudo su - ec2-user**

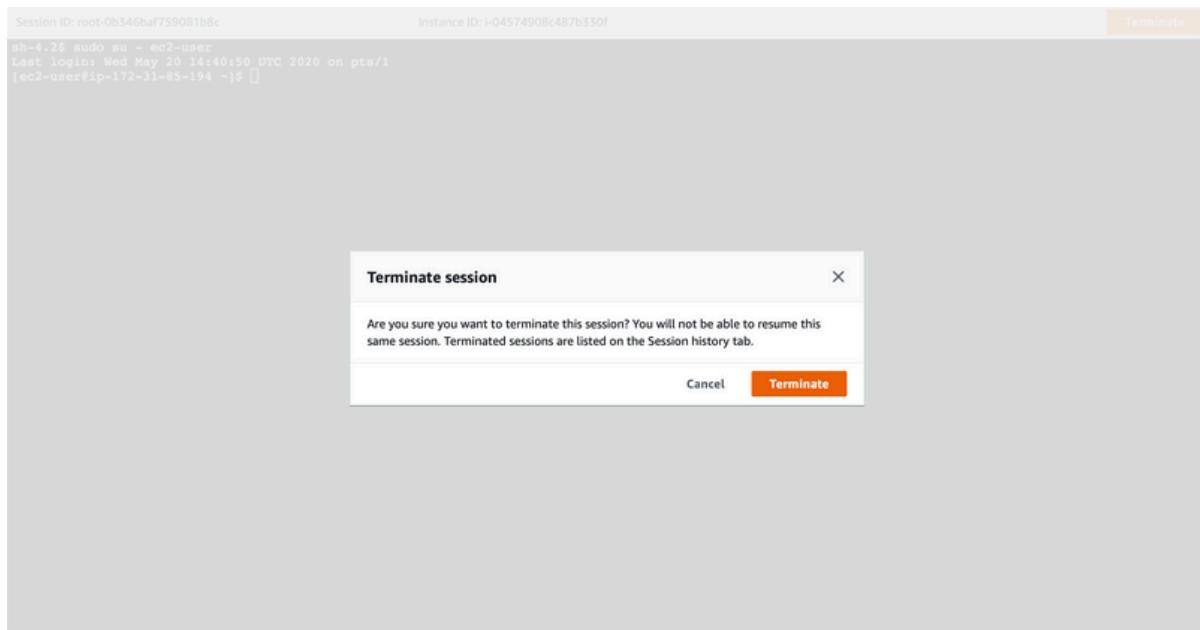


The screenshot shows a terminal session in a browser window. At the top, it displays 'Session ID: root-0b346baf759081b8c' and 'Instance ID: i-04574908c487b330f'. On the right side, there is an orange 'Terminate' button. The terminal window itself is mostly black, indicating it is a root shell. The visible text at the top of the terminal is:

```
Session ID: root-0b346baf759081b8c           Instance ID: i-04574908c487b330f
sh-4.2$ sudo su - ec2-user
Last login: Wed May 20 14:40:50 UTC 2020 on pts/1
[ec2-user@ip-172-31-85-194 ~]$
```

**Terminate:** We're not going to do much with it today so you will go ahead and terminate it

- 1.5 [ ] Click **Terminate**



## Go back to EC2

- 1.6 [ ] Click on **Services** at the top and type in **EC2** and go to the left-hand side and click on **Instances**

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP
	i-04574908c487b330f	t2.micro	us-east-1a	running	2/2 checks ...	None	ec2-18-212-22-229.co...	18.212.22.229

Instance: i-04574908c487b330f    Public DNS: ec2-18-212-22-229.compute-1.amazonaws.com

Description	Status Checks	Monitoring	Tags
Instance ID: i-04574908c487b330f	Instance state: running	Instance type: t2.micro	Public DNS (IPv4): ec2-18-212-22-229.compute-1.amazonaws.com IPv4 Public IP: 18.212.22.229 IPv6 IPs: - Elastic IPs: -
Finding: Opt-in to AWS Compute Optimizer for recommendations. <a href="#">Learn more</a>			

## Stop Instance

- 1.7 [ ] Click on **Actions Instance State** and **Stop** and select **Yes, Stop**

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links like EC2 Dashboard, Events, Tags, Reports, Limits, INSTANCES (selected), Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations, IMAGES, AMIs, and ELASTIC BLOCK STORE. The main area displays a table of instances. One instance, with ID i-04574908c487b330f, is selected. A context menu is open over this instance, with 'Actions' expanded. Under 'Instance State', the 'Stop' option is highlighted. Below the table, a detailed view of the selected instance shows its ID, state (running), type (t2.micro), and public DNS (ec2-18-212-22-229.compute-1.amazonaws.com). The status checks are green, and the public IP is 18.212.22.229.

This screenshot shows the same EC2 Instances page as above, but with a modal dialog titled 'Stop Instances' overlaid. The dialog asks, 'Are you sure you want to stop these instances?' and lists the instance ID i-04574908c487b330f. It contains a warning message: 'Note that when your instances are stopped: • Any data on the ephemeral storage of your instances will be lost.' At the bottom of the dialog are 'Cancel' and 'Yes, Stop' buttons. The background table of instances remains visible.

Next, we will create an AMI

## Creating an AMI:

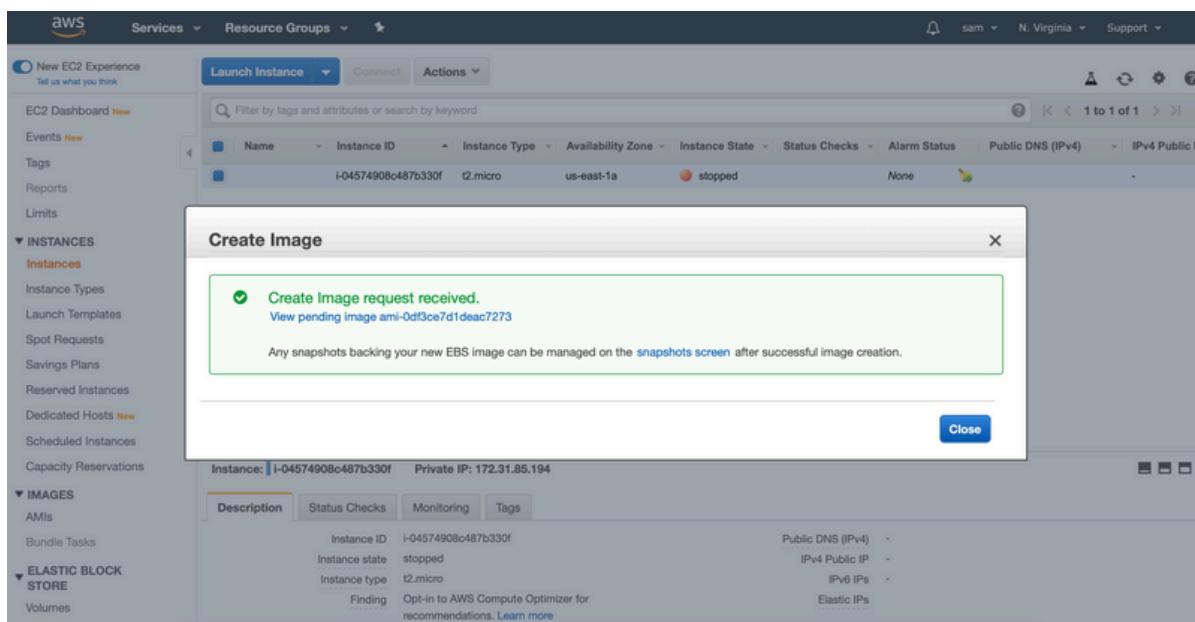
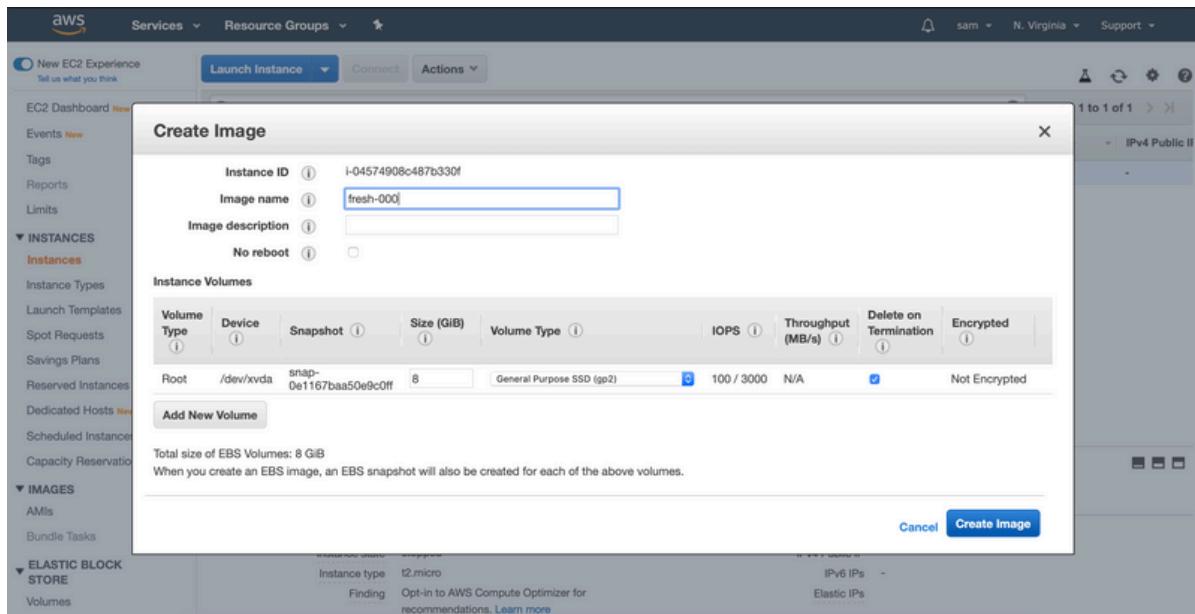
We are going to learn how to create an AMI (which is a snapshot or saving a copy of your entire server)

### Create Image

- 1.1 [] Go to the top of the page and select **Actions** then **Image** and **Create Image**

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links like New EC2 Experience, EC2 Dashboard, Events, Tags, Reports, Limits, Instances (selected), Images (underlined), and Elastic Block Store. The main area shows a table of instances. One instance, with the ID i-04574908c487b330f and Private IP 172.31.85.194, is selected. A context menu is open over this instance, with 'Actions' expanded. Under 'Image', the 'Create Image' option is highlighted. Below the table, there's a detailed view of the selected instance with tabs for Description, Status Checks, Monitoring, and Tags.

- 1.2 [] Fill out **Image Name** and click **Create Image** and click **View Pending image ami-0ae4eee56681b6324**



## Launch Server

- 1.3 [ ] Click **Launch**

Name	AMI Name	AMI ID	Source	Owner	Visibility	Status	Creation Date	Platform
	fresh-000	ami-0df3ce7d1deac7273	731409269211/fresh-000	731409269211	Private	available	May 22, 2020 at 8:12:29 AM ...	Other Linux

**Image: ami-0df3ce7d1deac7273**

**Details** **Permissions** **Tags**

AMI ID: ami-0df3ce7d1deac7273	AMI Name: fresh-000
Owner: 731409269211	Source: 731409269211/fresh-000
Status: available	State Reason: -
Creation date: May 22, 2020 at 8:12:29 AM UTC-4	Platform details: Linux/LINIX

## Choose AMI

- 1.4 [ ] Click on **Choose AMI** at the top and you can see it chose fresh-000 - ami-0df3ce7d1deac7273

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

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.

Search by Systems Manager parameter

Search: ami-0df3ce7d1deac7273

My AMIs (1)

fresh-000 - ami-0df3ce7d1deac7273

Select

64-bit (x86)

The following results for "ami-0df3ce7d1deac7273" were found in other catalogs:

- 4264 results in AWS Marketplace

AWS Marketplace provides partnered Software that is pre-configured to run on AWS

Quick Start (0)

My AMIs (1)

AWS Marketplace (4264)

Community AMIs (0)

Ownership

Owned by me

Shared with me

Architecture

32-bit (x86)

64-bit (x86)

64-bit (Arm)

Root device type

EBS

**Choose Instance Type:** Is a way for us to upgrade our server, make other changes to it or just so we have another copy of it so we can launch multiple servers

- 1.5 [] Click back on **Choose Instance Type** at the top of the screen and click **Cancel** at the bottom

The screenshot shows the 'Choose Instance Type' step of an AWS wizard. At the top, there are tabs: 1. Choose AMI, 2. Choose Instance Type (which is active), 3. Configure Instance, 4. Add Storage, 5. Add Tags, 6. Configure Security Group, and 7. Review. Below the tabs, a heading says 'Step 2: Choose an Instance Type'. A note states: '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.' There are filters for 'All instance types' and 'Current generation', and a 'Show/Hide Columns' button. A note below the filters says 'Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)'. The main table lists various instance types with columns for Family, Type, vCPUs, Memory (GiB), Instance Storage (GB), EBS-Optimized Available, Network Performance, and IPv6 Support. The t2.micro row is highlighted with a blue border and has a green 'Free tier eligible' badge. At the bottom right of the table are buttons for 'Cancel', 'Previous', 'Review and Launch' (which is highlighted in blue), and 'Next: Configure Instance Details'.

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	<b>t2.micro</b> <small>Free tier eligible</small>	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

- 1.6 [] To get back to AMI just click on the left hand side

Now we will move onto Auto Scaling groups

## Autoscaling Groups

What an Autoscaling group does is it allows you to insure that multiple instances or servers are running. So if you always want to always guarantee that one server is always running then auto scaling group would have a rule to check to say that at least one is running and if not to launch a new server. Also, auto scaling groups are used to meet the demand of whatever traffic you have.

- 1.1 [] Scroll down on the left hand side and click on **Auto Scaling Groups** and

click on **Create Auto Scaling group** and then click on **Get started**

Welcome to Auto Scaling

You can use Auto Scaling to manage Amazon EC2 capacity automatically, maintain the right number of instances for your application, operate a healthy group of instances, and scale it according to your needs.

[Learn more](#)

**Create Auto Scaling group**

Note: To create your Auto Scaling groups in a different region, select your region from the navigation bar.

**Benefits of Auto Scaling**

- Automated Provisioning**: Keep your Auto Scaling group healthy and balanced, whether you need one instance or 1,000.
- Adjustable Capacity**: Maintain a fixed group size or adjust dynamically based on Amazon CloudWatch metrics.
- Launch Template Support**: Provision instances easily using EC2 Launch Templates.

[Learn more](#) [Learn more](#) [Learn more](#)

**Additional Information**

- [Getting Started Guide](#)
- [Documentation](#)
- [All EC2 Resources](#)
- [Forums](#)
- [Pricing](#)
- [Contact Us](#)

**Create Auto Scaling Group**

Complete this wizard to create your Auto Scaling group. First, choose either a launch configuration or a launch template to specify the parameters that your Auto Scaling group uses to launch instances.

**Step 1: Create or select a launch configuration**

Create or select the launch configuration that your Auto Scaling group will use to launch your EC2 instances.

You can change your group's launch configuration at any time.

**Step 2: Create Auto Scaling group**

Next, give your group a name and specify how many instances you want to run in it.

Your group will maintain this number of instances, and replace any that become unhealthy or impaired.

You can optionally configure your group to adjust its capacity according to demand, in response to Amazon CloudWatch metrics.

[Cancel](#) [Get started](#)

## Choose AMI

- 1.2 [ ] Click on **My AMIs** and choose the fresh-000 and click **select**

The screenshot shows the 'Create Launch Configuration' step in the AWS EC2 wizard. The left sidebar has 'Quick Start' selected under 'My AMIs'. The main area lists several AMI options:

- Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0323c3dd2da7fb37d**: Free tier eligible, 64-bit. Root device type: ebs, Virtualization type: hvm.
- Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-01d025118d8e760db**: Free tier eligible. The description notes it's an EBS-backed image with Docker, PHP, MySQL, PostgreSQL, and other packages. Root device type: ebs, Virtualization type: hvm.
- Red Hat Enterprise Linux 8 (HVM), SSD Volume Type - ami-098f16afa9edf40be**: Free tier eligible. Red Hat Enterprise Linux version 8 (HVM), EBS General Purpose (SSD) Volume Type. Root device type: ebs, Virtualization type: hvm.
- SUSE Linux Enterprise Server 15 SP1 (HVM), SSD Volume Type - ami-0068cd63259e9f24c**: Free tier eligible. SUSE Linux Enterprise Server 15 Service Pack 1 (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.
- Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-085925f297f89fce1**: Free tier eligible. Root device type: ebs, Virtualization type: hvm.

A 'Select' button is shown next to each AMI entry.

The screenshot shows the 'Create Launch Configuration' step in the AWS EC2 wizard. The left sidebar has 'Quick Start' selected under 'My AMIs'. A search bar at the top right shows 'Search my AMIs'. The main area displays one AMI entry:

- fresh-000 - ami-0df3ce7d1deac7273**: Root device type: ebs, Virtualization type: hvm, Owner: 731409269211. A 'Select' button is shown to its right.

On the left, a sidebar shows filtering options:
 

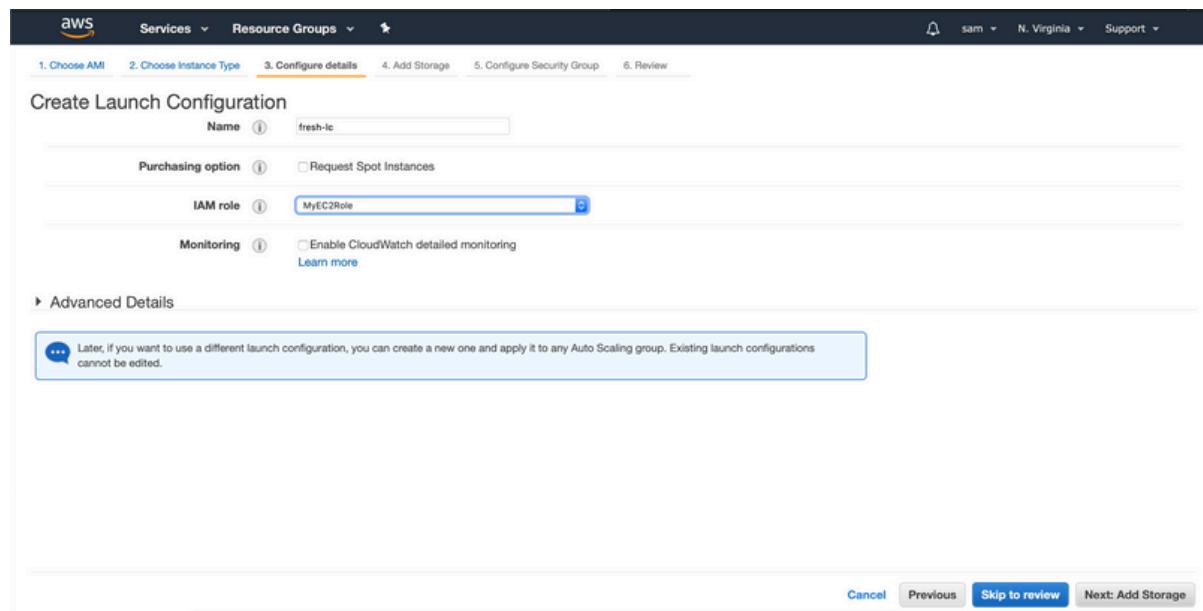
- Ownership**:  Owned by me,  Shared with me.
- Architecture**:  32-bit,  64-bit.
- Root device type**:  EBS,  Instance store.

## Choose Instance Type

- 1.3 [ ] We will stick with t2.micro and click **Next: Configure details**

## Configure details

- 1.4 [ ] We will name the Launch Configuration fresh-lc, click on the drop down box for IAM role and select **MyEC2Role** and click **Next: Add Storage**



## Add storage

- 1.5 [ ] The defaults look good so click **Next: Configure Security Group**

**Create Launch Configuration**

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.

<https://docs.aws.amazon.com/console/ec2/launchinstance/storage> about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput	Delete on Termination	Encrypted
Root	/dev/xvda	snap-0265352a2d66497c6	8	General Purpose (SSD)	100 / 3000	N/A	<input checked="" type="checkbox"/>	No

Add New Volume

Free tier eligible customers can get up to 30 GB of EBS storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

Cancel Previous Skip to review Next: Configure Security Group

## Configure Security Group

- 1.6 [] The security groups look good so click **Review**

**Create Launch Configuration**

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: AutoScaling-Security-Group-1

Description: AutoScaling-Security-Group-1 (2020-05-22 09:34:36.220-04:00)

Type	Protocol	Port Range	Source
SSH	TCP	22	Anywhere <input checked="" type="checkbox"/> 0.0.0.0/0 <input type="checkbox"/>

Add Rule

**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.

Cancel Previous Review

## Review

- 1.7 Click **Create launch configuration** and in the drop down box click on **Proceed without a key pair** and check mark **I acknowledge that I will not be able to connect to this instance unless I already know the password built into this AMI** and click **Create launch configuration**

**AMI Details**

**Instance Type**

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

**Launch configuration details**

Name: fresh-ic	Edit details				
Purchasing option: On demand					
EBS Optimized: No					

**Create launch configuration**

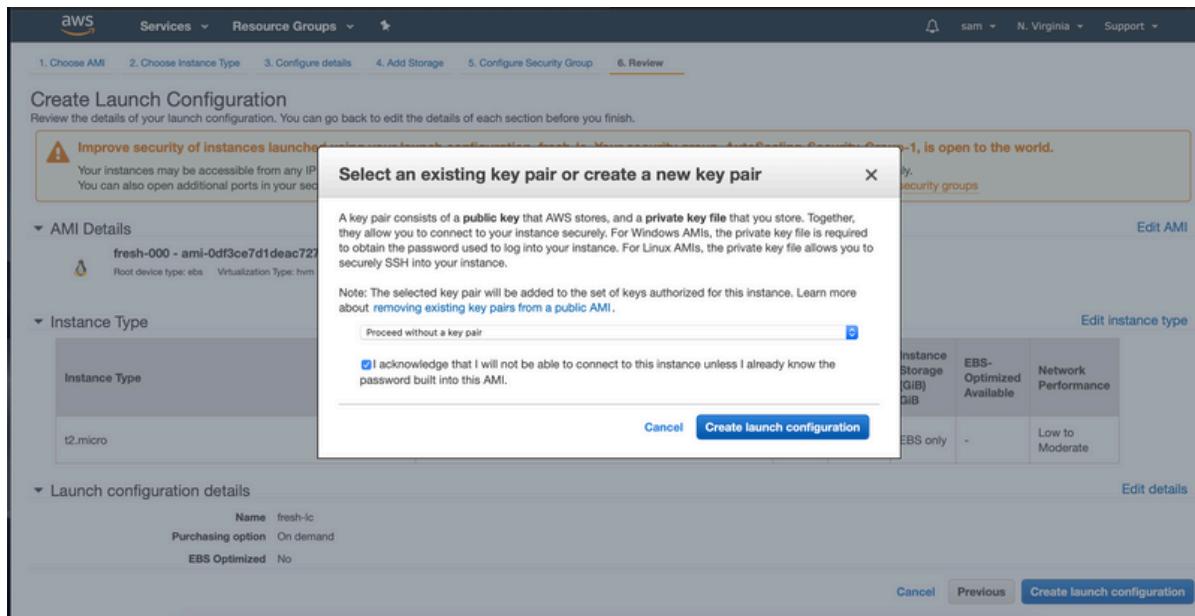
**AMI Details**

**Instance Type**

Instance Type	EBS-Optimized Available	Network Performance
t2.micro	EBS only	Low to Moderate

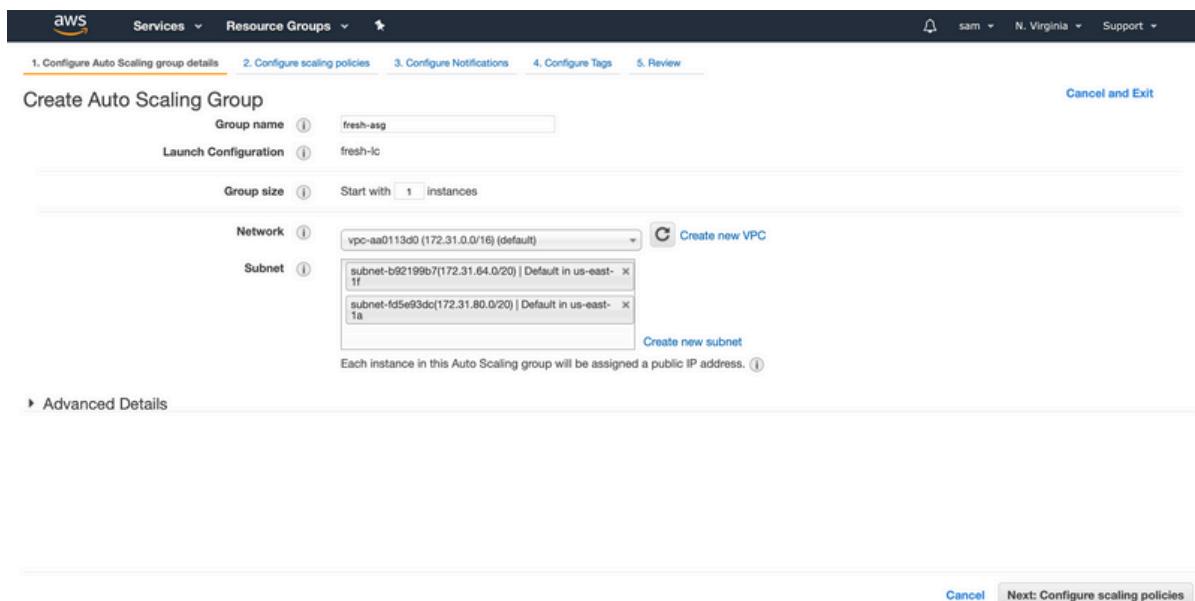
**Launch configuration details**

Name: fresh-ic	Edit details				
Purchasing option: On demand					
EBS Optimized: No					



## Configure Auto Scaling group details

- 1.8 [] Name Auto Scaling Group **fresh-asg** and group size leave at 1, select the Network as **vac-cf3079b5 (default)** and select the drop down box for subnet and choose two subnets. Click on **Advanced Details** and it all looks good so click **Next: Configure scaling policies**

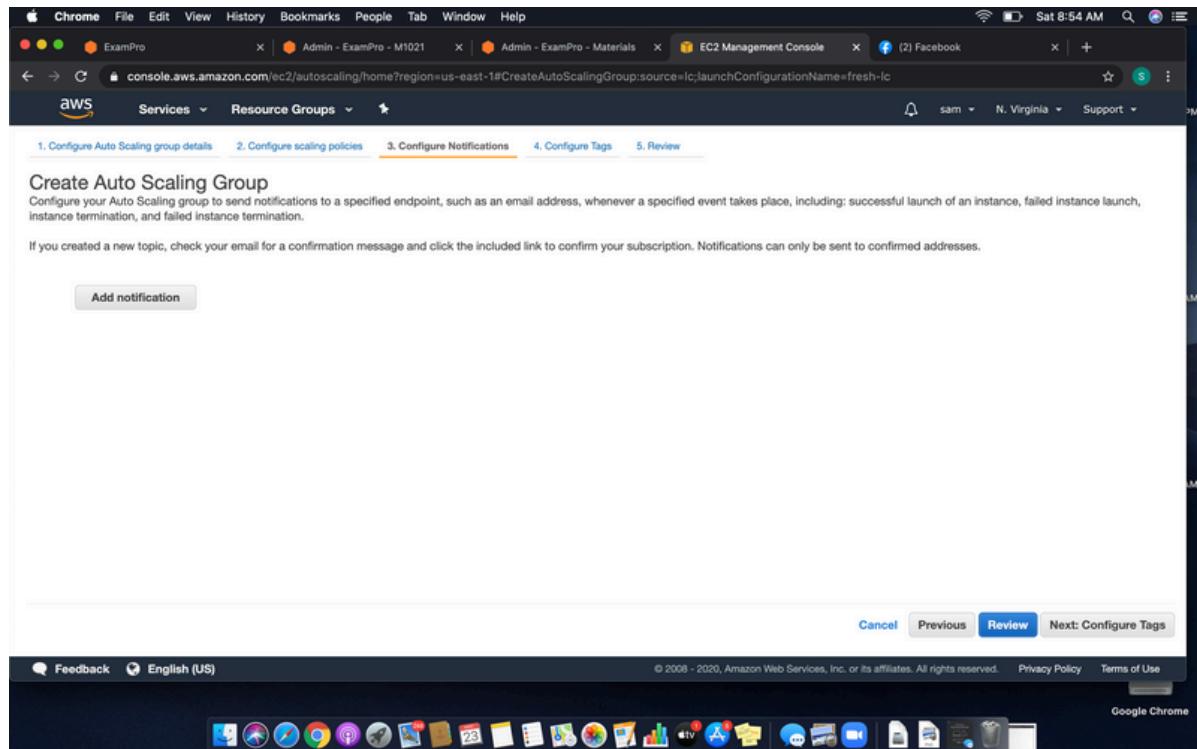


## Configure scaling policies

- 1.9 [ ] Select **Keep this group at its initial size** and click **Next:Configure Notifications**

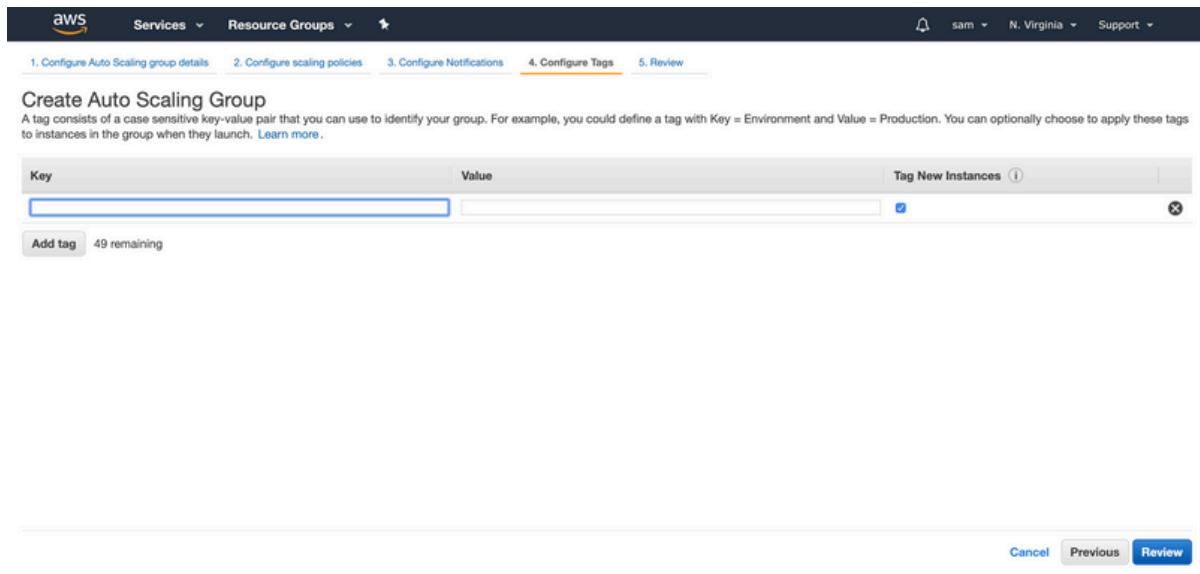
## Configure Notifications

### ■ 1.10 [ ] Select **Next: Configure Tags**



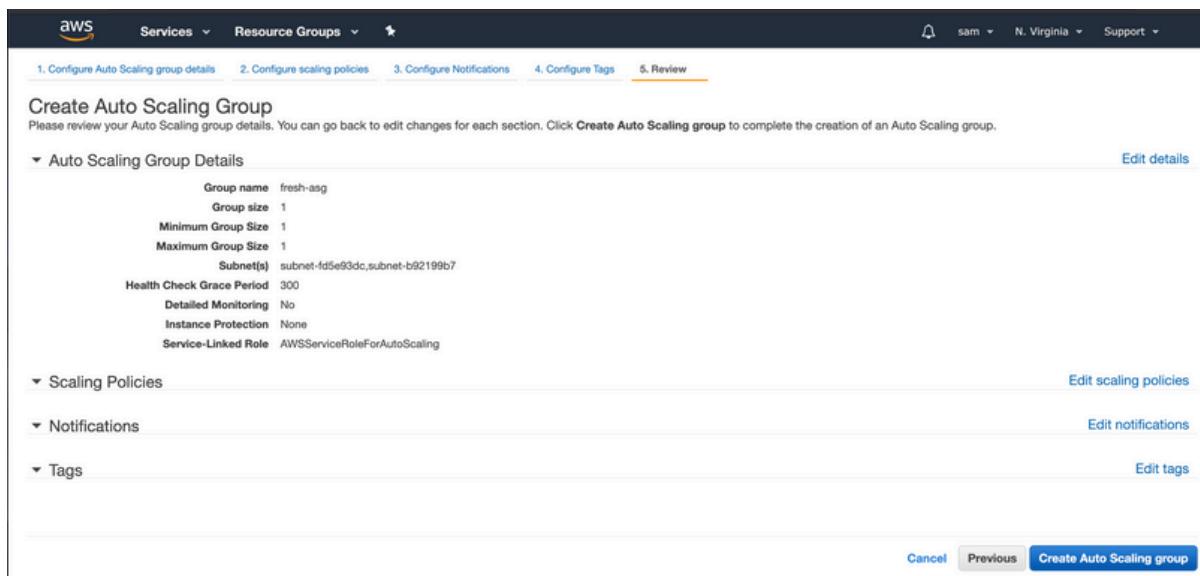
## Configure Tags

### ■ 1.11 [ ] Select **Review**



## Review

### 1.12 [ ] Select **Create Auto Scaling group**



### 1.13 [ ] Auto Scaling group has been created so click **Close**

The screenshot shows the AWS Auto Scaling group creation status page. At the top, there's a navigation bar with the AWS logo, 'Services' dropdown, 'Resource Groups' dropdown, a bell icon, 'sam' user info, 'N. Virginia' region, and 'Support' dropdown. Below the header, the title 'Auto Scaling group creation status' is displayed. A green success message box contains the text 'Successfully created Auto Scaling group' and a link 'View creation log'. Underneath, there's a section titled 'View' with links 'View your Auto Scaling groups' and 'View your launch configurations'. A helpful resources section follows, with a 'Close' button at the bottom right.

## Instances

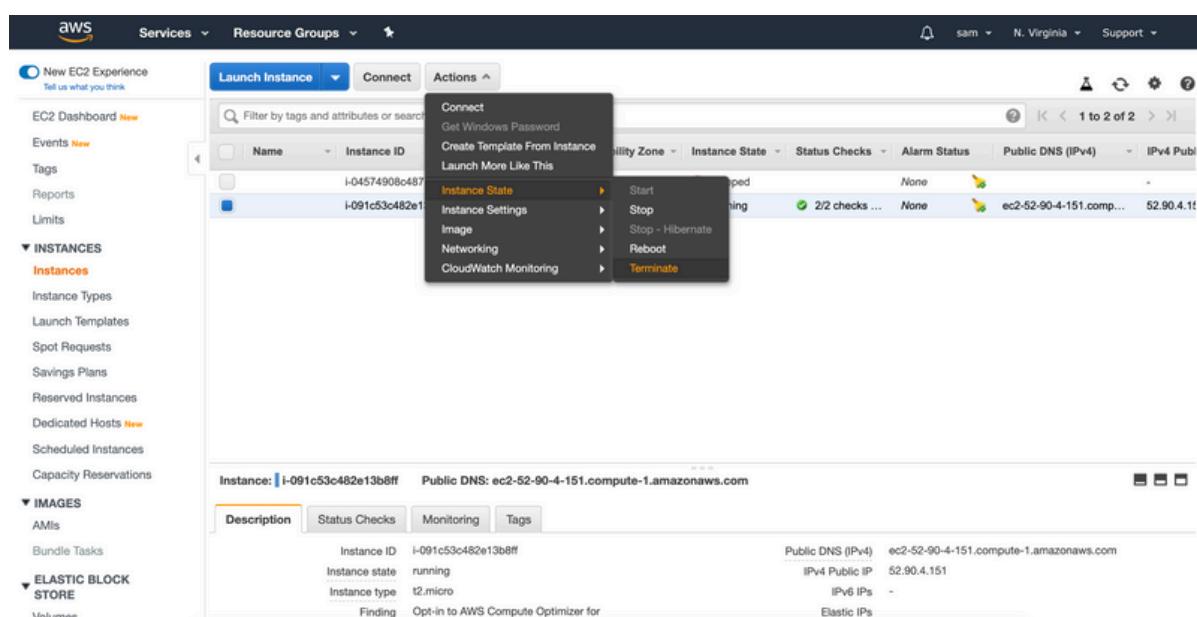
- 1.14 [ ] Right Click on **Instances** on the left hand side and you will start to see instances, you can refresh your instance tab and see more instances spit up

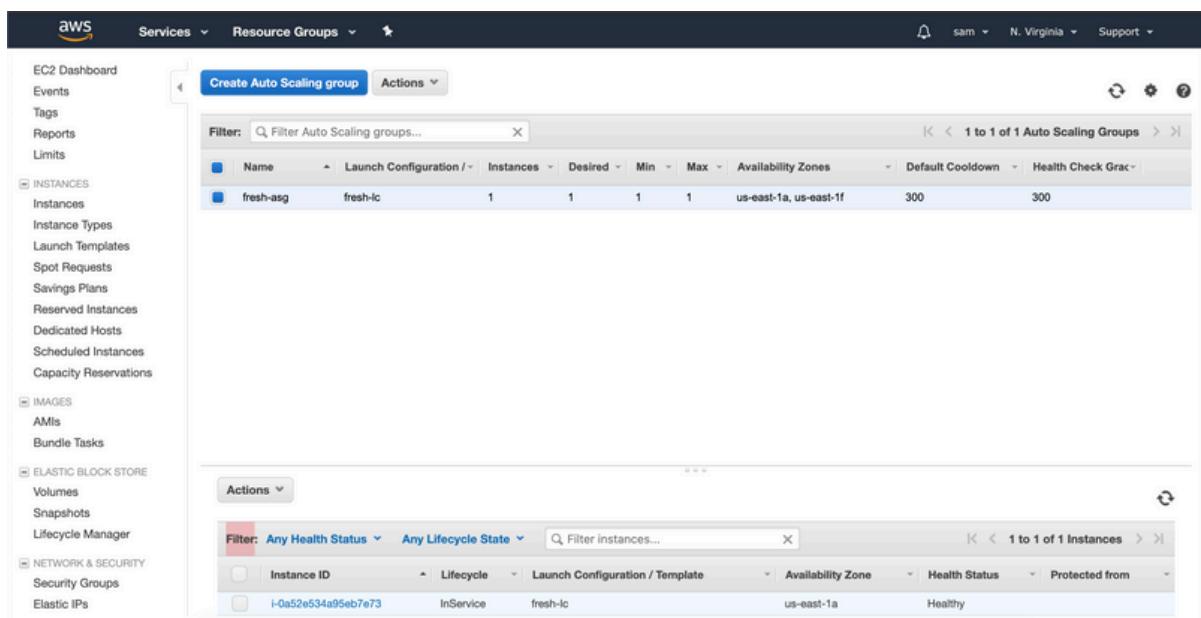
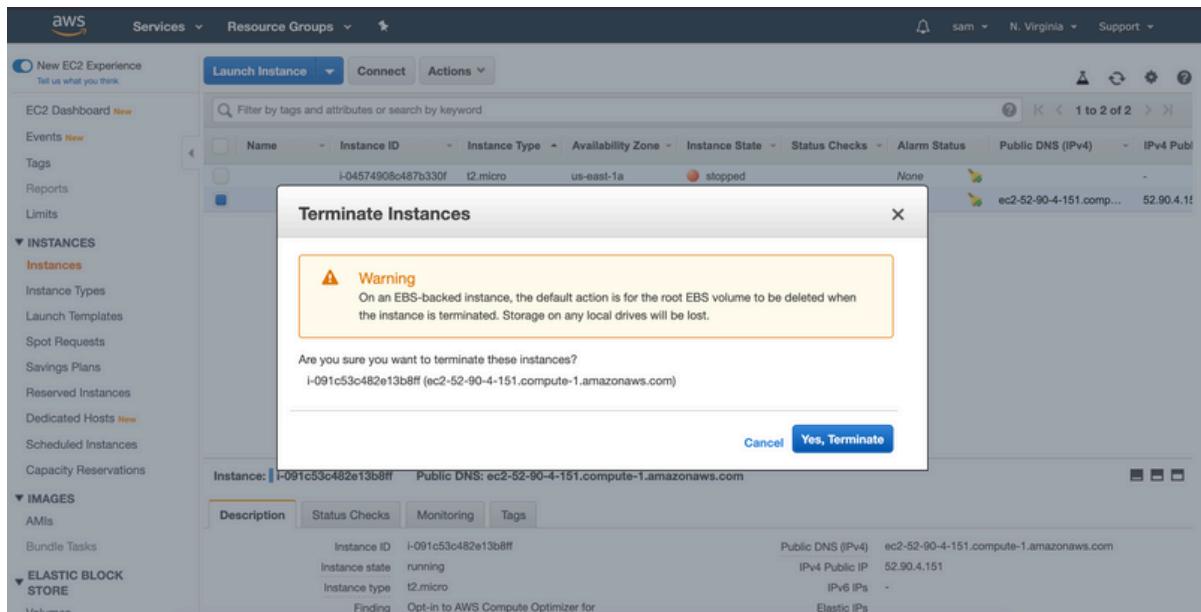
The screenshot shows the AWS EC2 Instances dashboard. The left sidebar has a 'New EC2 Experience' button, 'Tell us what you think' link, and sections for 'Events', 'Tags', 'Reports', 'Limits', 'INSTANCES' (with 'Instances' selected), 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts', 'Scheduled Instances', 'Capacity Reservations', 'IMAGES' (with 'AMIs' selected), 'Bundle Tasks', and 'ELASTIC BLOCK STORE' (with 'Find image' link). The main content area has tabs 'Launch Instance', 'Connect', and 'Actions'. It includes a search bar and a table with columns: Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, Public DNS (IPv4), and IPv4 Publ. The table shows two instances: one stopped (i-04574908c487b330) and one running (i-091c53c482e13b8ff).

Auto Scaling can ensure that there are always a minimum of servers running but

if we were to terminate the instance it's going to detect that this one is no longer healthy. After a while it will determine that it's unhealthy and the health status will change from healthy to unhealthy. The way this Auto Scaling group is going to respond is to launch a new instance. You can keep hitting the refresh button until you see another instance replace this unhealthy one. You may have to wait a few minutes to see the new replacement instance.

- 1.15 [ ] Check mark the **running server** select **Actions** select **Instance State** and **Terminate** and then select **Yes Terminate**





##Delete Auto Scaling Group

- 1.16 [ ] Select **Actions** and **Delete** and click **Yes, Delete**

The screenshot shows the AWS Auto Scaling Groups page. In the top navigation bar, there is a 'Services' dropdown, a 'Resource Groups' dropdown, and user information ('sam', 'N. Virginia', 'Support'). Below the navigation is a sidebar with links for EC2 Dashboard, Events, Tags, Reports, Limits, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations, Images, AMIs, and Bundle Tasks. The main content area has two tabs: 'Create Auto Scaling group' and 'Actions'. The 'Actions' tab is selected, showing a sub-menu with 'Edit' and 'Delete' buttons. The 'Delete' button is highlighted with a red box. Below this is a table titled 'Auto Scaling Groups' with one entry: 'fresh-asg' (Name), 'fresh-lc' (Launch Configuration), '1' (Instances), '1' (Desired), '1' (Min), '1' (Max), 'us-east-1a, us-east-1f' (Availability Zones), '300' (Default Cooldown), and '300' (Health Check Grace Period). At the bottom of the main content area is another table titled 'Instances' with one entry: 'i-0a52e534a95eb7e73' (Instance ID), 'InService' (Lifecycle), 'fresh-lc' (Launch Configuration / Template), 'us-east-1a' (Availability Zone), and 'Healthy' (Health Status).

This screenshot is identical to the one above, but it includes a modal dialog box in the center. The dialog is titled 'Delete Auto Scaling group' and contains the message 'Are you sure you want to delete this resource? fresh-asg'. It has two buttons at the bottom: 'Cancel' and a blue 'Yes, Delete' button.

- 1.17 [ ] You can click on **Instance** on your left hand side to make sure it was deleted, you may need to refresh the page

The screenshot shows the AWS EC2 Instances dashboard. On the left, there's a sidebar with various service links like New EC2 Experience, Images, Elastic Block Store, Network & Security, Load Balancing, and Auto Scaling. The main area has tabs for Launch Instance, Connect, and Actions. A search bar at the top right says "Filter by tags and attributes or search by keyword". Below it is a table with columns: Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, Public DNS (IPv4), and IPv4 Pub. There are three entries in the table:

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Pub
	i-04574908c487b330f	t2.micro	us-east-1a	stopped	None	None		
	i-091c53c482e13b8ff	t2.micro	us-east-1a	terminated	None	None		
	i-0a52e534a95eb7e...	t2.micro	us-east-1a	terminated	None	None		

A message below the table says "Select an instance above".

Now we will move onto Elastic Load Balancer

## Elastic Load Balancer

We are going to learn how to set up Elastic Load Balancer in front of your EC2 instances.

### Launching an instance

- 1.1 [ ] In your EC2 dashboard under **Create Instance** click "Launch Instance"

The screenshot shows the AWS EC2 Management Console dashboard. On the left, there's a sidebar with navigation links like EC2 Dashboard, Events, Tags, Reports, Limits, Instances, Launch Templates, Spot Requests, Reserved Instances, Dedicated Hosts, Scheduled Instances, 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, and Launch Configurations. The main area displays 'Resources' for the US East (N. Virginia) region, showing 0 Running Instances, 0 Dedicated Hosts, 1 Volumes, 0 Key Pairs, 0 Placement Groups, 0 Elastic IPs, 1 Snapshots, 0 Load Balancers, and 3 Security Groups. A central callout box says 'Learn more about the latest in AWS Compute from AWS re:Invent by viewing the EC2 Videos.' Below it, there's a 'Create Instance' section with a 'Launch Instance' button highlighted by a red box. To the right, there's a 'Migrate a Machine' section, 'Scheduled Events' (none), and 'AWS Marketplace' with various software offerings.

## 1.2 [] Select "Amazon Linux 2"

The screenshot shows the 'Step 1: Choose an Amazon Machine Image (AMI)' page of the AWS Launch Instance Wizard. It lists several AMI options: 'Amazon Linux 2 AMI (HVM, SSD Volume Type)', 'Amazon Linux AMI 2018.03.0 (HVM, SSD Volume Type)', 'Red Hat Enterprise Linux 8 (HVM, SSD Volume Type)', 'SUSE Linux Enterprise Server 15 SP1 (HVM, SSD Volume Type)', and 'Ubuntu Server 18.04 LTS (HVM, SSD Volume Type)'. The 'Amazon Linux 2 AMI (HVM, SSD Volume Type)' is selected and highlighted with a red box. The 'Select' button next to it is also highlighted with a red box. The page includes a search bar, a 'Quick Start' sidebar with filters for My AMIs, AWS Marketplace, Community AMIs, and a 'Free tier eligible' filter, and a note about launching a database instance with Amazon RDS.

- 1.3 [] On the Step 2 page keep the t2.micro option and hit the "Next: Configure Instance Details" button
- 1.4 [] In Configure Instance Details form add "2" to the number of instances

- 1.5 [] Add an IAM role "myEC2Role"
- 1.6 [] Click the "Next: Add Storage" button

You will not need to change anything in Storage

The screenshot shows the AWS Launch Instance Wizard Step 3: Configure Instance Details. The 'Number of Instances' field is set to 2, and the 'Launch into Auto Scaling Group' link is highlighted with a red box. The 'IAM role' dropdown is set to 'MyEC2Role', which is also highlighted with a red box. The 'Review and Launch' button at the bottom right is also highlighted with a red box.

- 1.7 [] Click the "Next: Add Tags" button

Again you will not need to change anything here.

- 1.8 [] Select an existing security group
- 1.9 [] Click the "Review and Launch" button

**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-0fc4c4689129b636	AutoScaling-Security-Group-1	AutoScaling-Security-Group-1 (2019-10-08 13:03:55.534-04:00)	<a href="#">Copy to new</a>
sg-e7d772b7	default	default VPC security group	<a href="#">Copy to new</a>
sg-05638c2fc66f18e	launch-wizard-1	launch-wizard-1 created 2019-10-08T12:37:04.973-04:00	<a href="#">Copy to new</a>

Inbound rules for sg-e7d772b7 (Selected security groups: sg-e7d772b7)

Type	Protocol	Port Range	Source	Description
All traffic	All	All	sg-e7d772b7 (default)	

Cancel Previous Review and Launch

- 1.10 [ ] In the dropdown choose "Proceed without a key pair"

**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.

AMI Details

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0b69ea6ff7391e80

Instance Type

Security Groups

Inbound Rules

Instance Details

Storage

Tags

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.

Proceed without a key pair

Acknowledge that I will not be able to connect to this instance unless I already know the password built into this AMI.

Cancel Launch Instances

Cancel Previous Launch

- 1.11 [ ] Click the "View Instance" button

Your instances are now launching  
The following instance launches have been initiated: i-0edebccdd0da5eafa, i-0fff6f21c38be58d8 View launch log

Get notified of estimated charges  
Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

How to connect to your instances  
Your instances are launching, and it may take a few minutes until they are in the running state, when they will be ready for you to use. Usage hours on your new instances will start immediately and continue to accrue until you stop or terminate your instances. Click [View Instances](#) to monitor your instances' status. Once your instances are in the running state, you can [connect](#) to them from the Instances screen. Find out how to connect to your instances.

Here are some helpful resources to get you started

- How to connect to your Linux instance
  - [Amazon EC2: User Guide](#)
  - [Amazon EC2: Discussion Forum](#)
- Learn about AWS Free Usage Tier

While your instances are launching you can also

- Create status check alarms to be notified when these instances fail status checks. (Additional charges may apply)
- Create and attach additional EBS volumes. (Additional charges may apply)
- Manage security groups

[View Instances](#)

## Creating the ELB

- 2.1 [] Once the instances are running give them a new name.

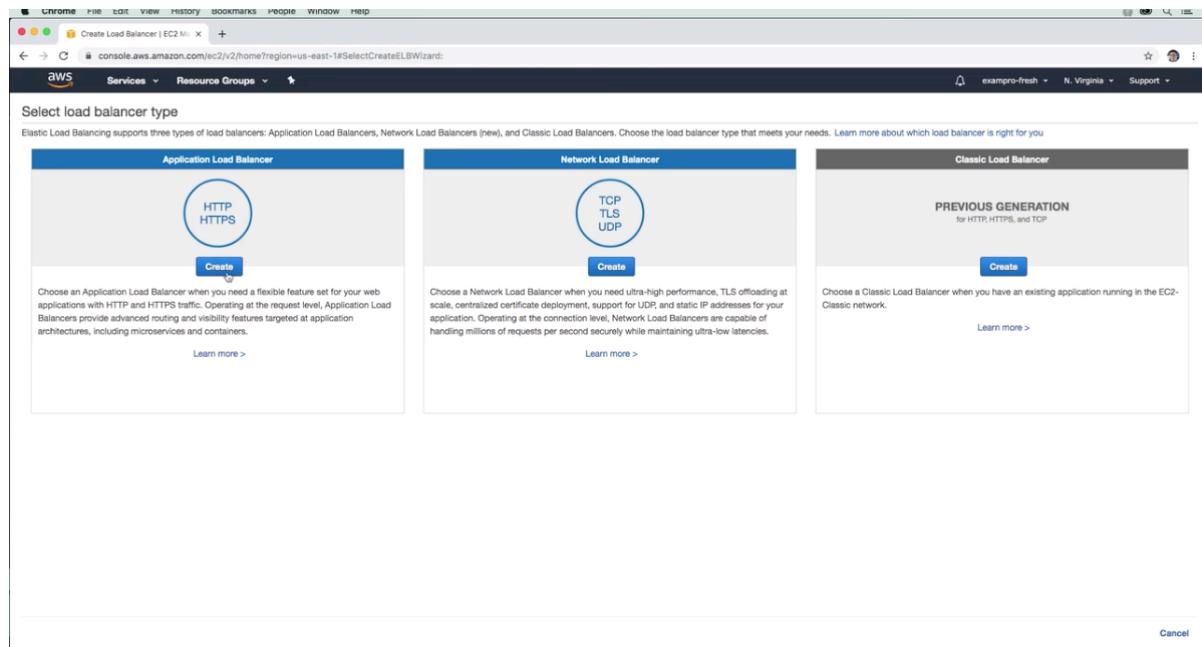
Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs	Key Name	Monitoring
i-02a20295e6659cbcd	i2.micro	us-east-1a	stopped	None	-	-	-	-	-	-	disabled
i-090d64a1e34de84d7	i2.micro	us-east-1c	terminated	None	-	-	-	-	-	-	disabled
i-0d4db0c835e6d975c4	i2.micro	us-east-1c	terminated	None	-	-	-	-	-	-	disabled
InstanceA	i-0edebccdd0da5eafa	i2.micro	us-east-1c	running	2/2 checks passed	None	ec2-52-90-146-196.co...	52.90.146.196	-	-	disabled
InstanceB	i-0fff6f21c38be58d8	i2.micro	us-east-1c	running	2/2 checks passed	None	ec2-3-87-132-76.compute-1.amazonaws.com	3.87.132.76	-	-	disabled

Instance: i-0fff6f21c38be58d8 (InstanceB) Public DNS: ec2-3-87-132-76.compute-1.amazonaws.com

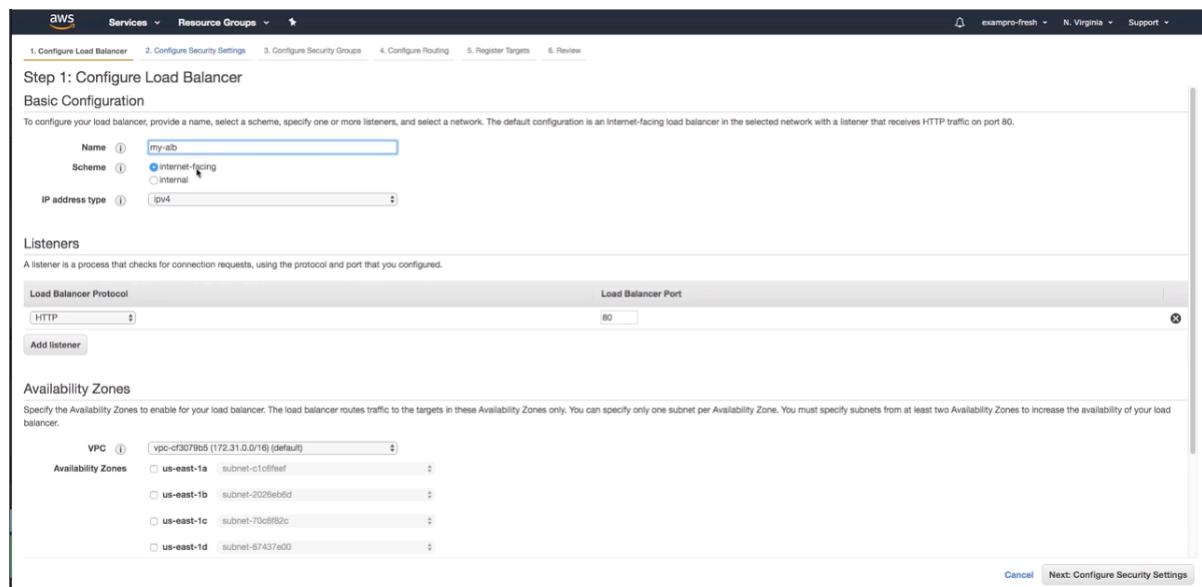
Description	Status Checks	Monitoring	Tags
Instance ID: i-0fff6f21c38be58d8	Running	None	ec2-3-87-132-76.compute-1.amazonaws.com
Instance state:	running	IPv4 Public IP:	3.87.132.76
Instance type:	i2.micro	IPv6 IPs:	-
Elastic IPs:	-	Private DNS:	ip-172-31-42-184.ec2.internal
Availability zone:	us-east-1c	Private IPs:	172.31.42.184
Security groups:	default, view inbound rules, view outbound rules	Secondary private IPs:	-
Scheduled events:	No scheduled events	VPC ID:	vpc-c030795
AMI ID:	amzn2-ami-hvm-2.0.20190823.1-x86_64-gp2 (ami-0b69ea0ff7391e80)	Subnet ID:	subnet-70cd9f82c
Platform:	-	Network interfaces:	eth0
IAM role:	MyEC2Role	Source/dest. check:	True

- 2.2 [] Under the EC2 Console click on the link to "Load Balancers"

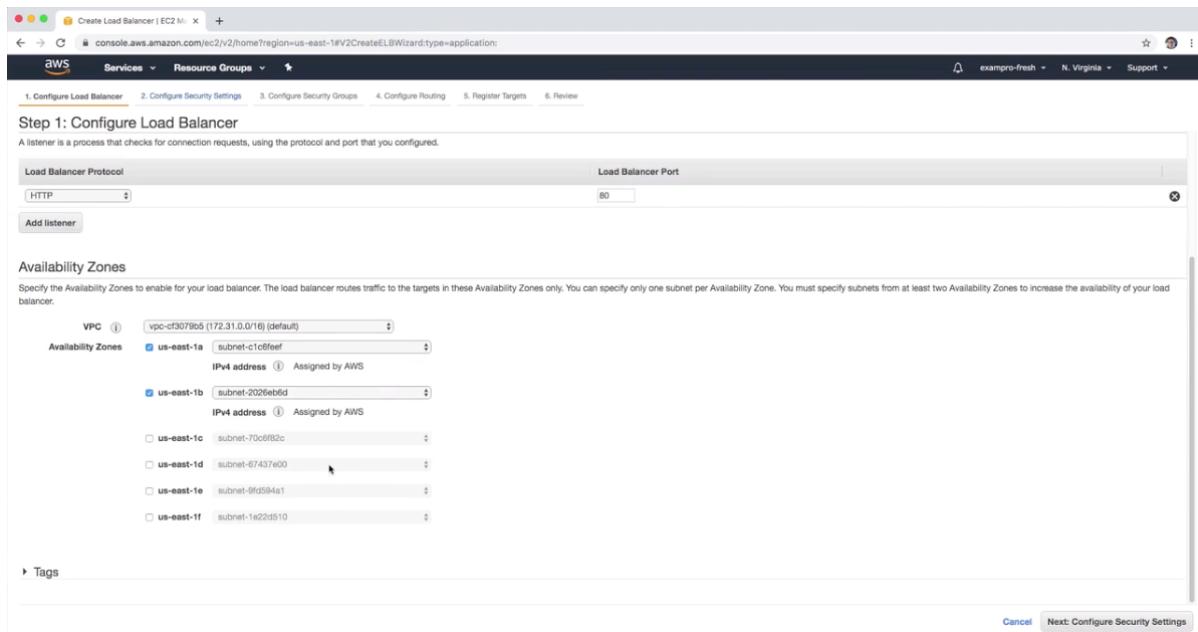
■ 2.3 [ ] Click the "Create" button for Application Load Balancer



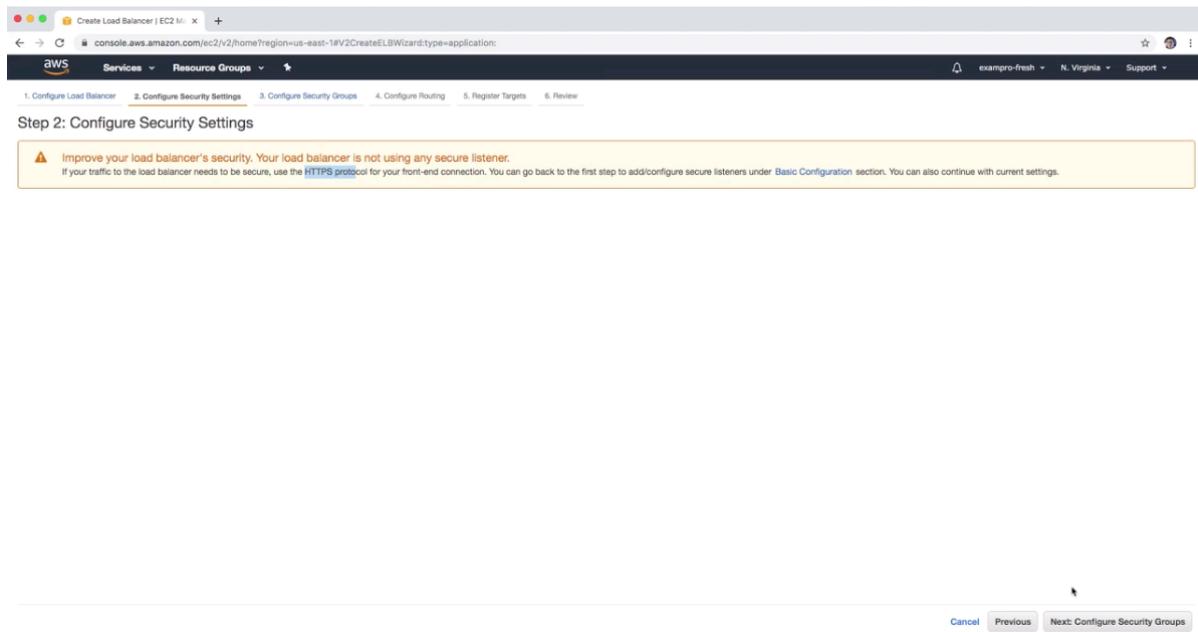
■ 2.4 [ ] Give the load balancer a name and keep it internet-facing



■ 2.5 [ ] Select at least two Availability Zones



- 2.6 [] Click the "Next: Configure Security Settings" button
- 2.7 [] You do not need to configure any security settings. Click the "Next: Configure Security Groups" button



- 2.8 [] You can leave the default Security Group and click the "Next: Configure Security Groups" button

## Routing" button

**Step 3: Configure Security Groups**

A security group is a set of firewall rules that control the traffic to your load balancer. On this page, you can add rules to allow specific traffic to reach your load balancer. First, decide whether to create a new security group or select an existing one.

Assign a security group:

- Create a new security group
- Select an existing security group

Security Group ID	Name	Description	Actions
sg-0fecf4689129b636	AutoScaling-Security-Group-1	AutoScaling-Security-Group-1 (2019-10-08 13:03:55.534-04:00)	Copy to new
sg-e7d772b7	default	default VPC security group	Copy to new
sg-05638b2fd66f18e	launch-wizard-1	launch-wizard-1 created 2019-10-08T12:37:04.973-04:00	Copy to new

Filter: VPC security groups

Cancel Previous Next: Configure Routing

- 2.9 [] Create a new target group
- 2.10 [] Click the "Next: Register Targets" button

**Step 4: Configure Routing**

Your load balancer routes requests to the targets in this target group using the protocol and port that you specify, and performs health checks on the targets using these health check settings. Note that each target group can be associated with only one load balancer.

**Target group**

Target group: New target group  
Name: my-target-group  
Target type: Instance  
Protocol: HTTP  
Port: 80

**Health checks**

Protocol: HTTP  
Path: /

Advanced health check settings

Cancel Previous Next: Register Targets

# Registering Targets

- 3.1 [] Select both instances and click "Add to registered"
- 3.2 [] Click the "Next: Review" button

**Step 5: Register Targets**

Register targets with your target group. If you register a target in an enabled Availability Zone, the load balancer starts routing requests to the targets as soon as the registration process completes and the target passes the initial health checks.

**Registered targets**

To deregister instances, select one or more registered instances and then click Remove.

Instance	Name	Port	State	Security groups	Zone
i-0debcbcd8da5eafa	InstanceA	80	running	default	us-east-1c
i-0ff6f21c38be58d8	InstanceB	80	running	default	us-east-1c

**Instances**

To register additional instances, select one or more running instances, specify a port, and then click Add. The default port is the port specified for the target group. If the instance is already registered on the specified port, you must specify a different port.

Instance	Name	State	Security groups	Zone	Subnet ID	Subnet CIDR
i-0debcbcd8da5eafa	InstanceA	running	default	us-east-1c	subnet-70d6f82c	172.31.32.0/20
i-0ff6f21c38be58d8	InstanceB	running	default	us-east-1c	subnet-70d6f82c	172.31.32.0/20

**Add to registered** on port 80

**Cancel** **Previous** **Next: Review**

- 3.3 [] Click the "Create" button

**Step 6: Review**

Please review the load balancer details before continuing

**Load balancer**

- Name: my-slb
- Scheme: internet-facing
- Listeners: Port 80 - Protocol: HTTP
- IP address type: ipv4
- VPC: vpc-c3079b5
- Subnets: subnet-c1c0feef, subnet-2026eb8d
- Tags

**Security groups**

- Security groups: sg-e7d772b7

**Routing**

- Target group: New target group
- Target group name: my-target-group
- Port: 80
- Target type: instance
- Protocol: HTTP
- Health check protocol: HTTP
- Path: /
- Health check port: traffic port
- Healthy threshold: 5
- Unhealthy threshold: 2
- Timeout: 5
- Interval: 30
- Success codes: 200

**Targets**

- Instances: i-0debcbcd8da5eafa (InstanceA:80), i-0ff6f21c38be58d8 (InstanceB:80)

**Create** **Cancel** **Previous** **Next**

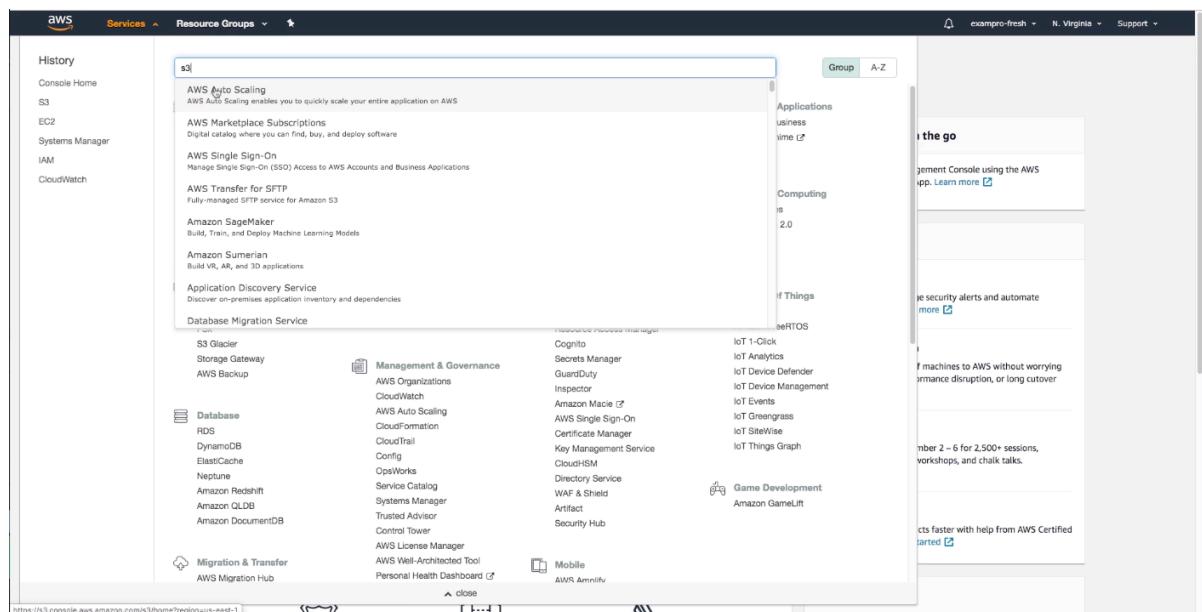
You should now have a Load Balancer created. When you are done you can Disable the ELB to avoid incurring any costs.

## Uploading a File to S3:

We are going to learn how to use S3 for file storage.

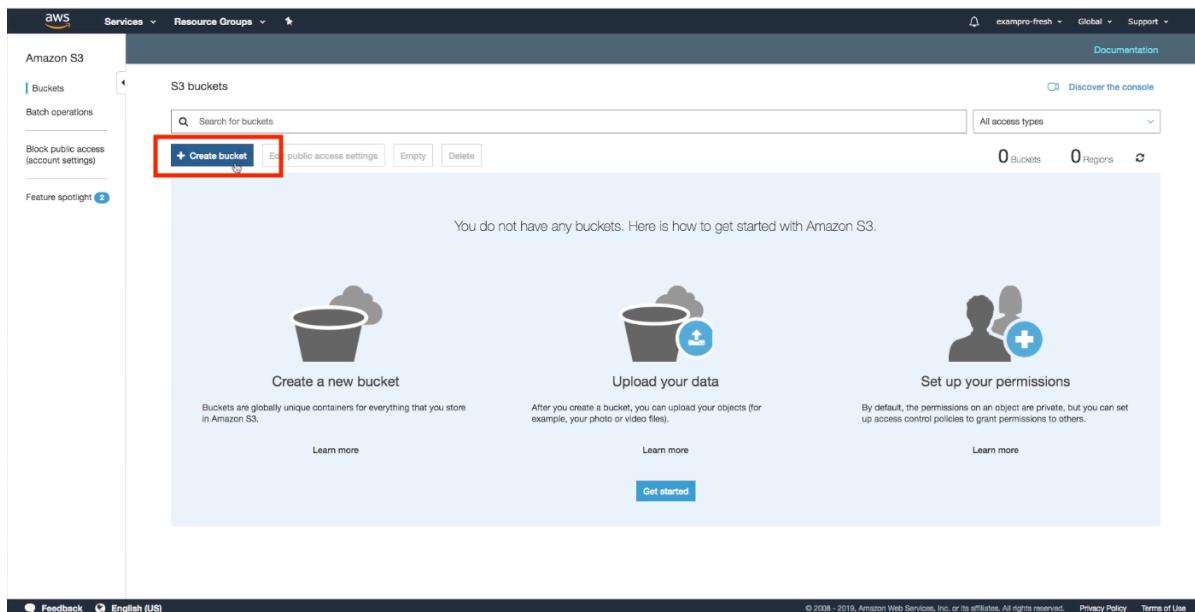
## Navigating to S3 Services

- 1.1 [] Inside the AWS Management Console use the search bar and type in S3.

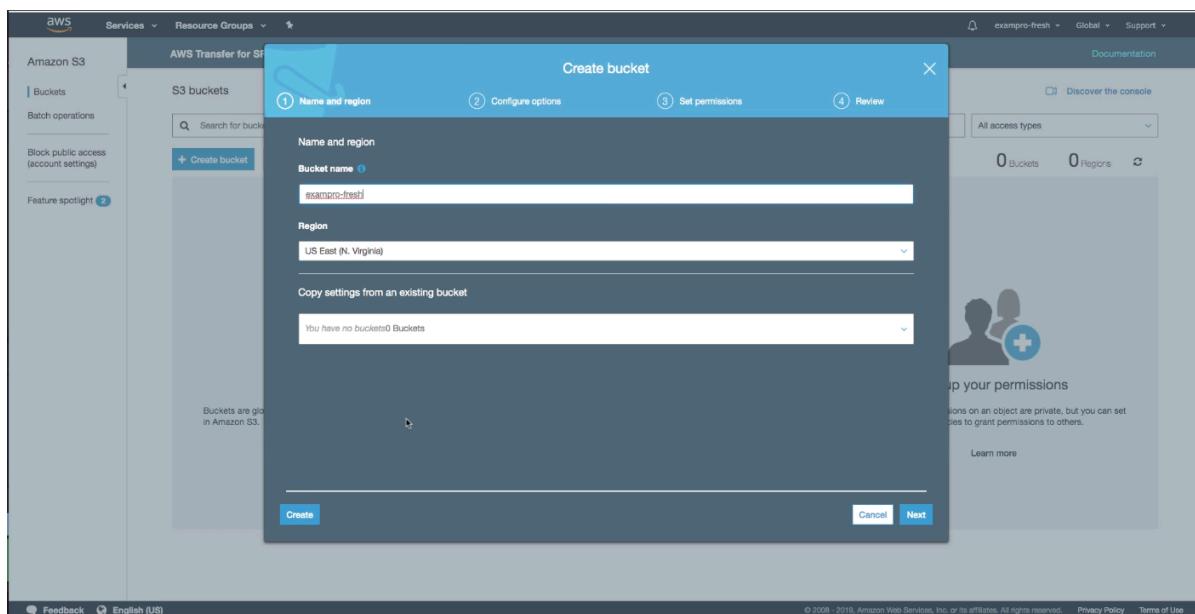


## Creating a Bucket

- 2.1 [] Click the "Create bucket" button



- 2.2 [] Create a unique Bucket name
- 2.3 [] Pick a region
- 2.4 [] Click the "Create" button on the bottom of the model



## Uploading a File to the Bucket

- 3.1 [ ] Click on the newly created bucket

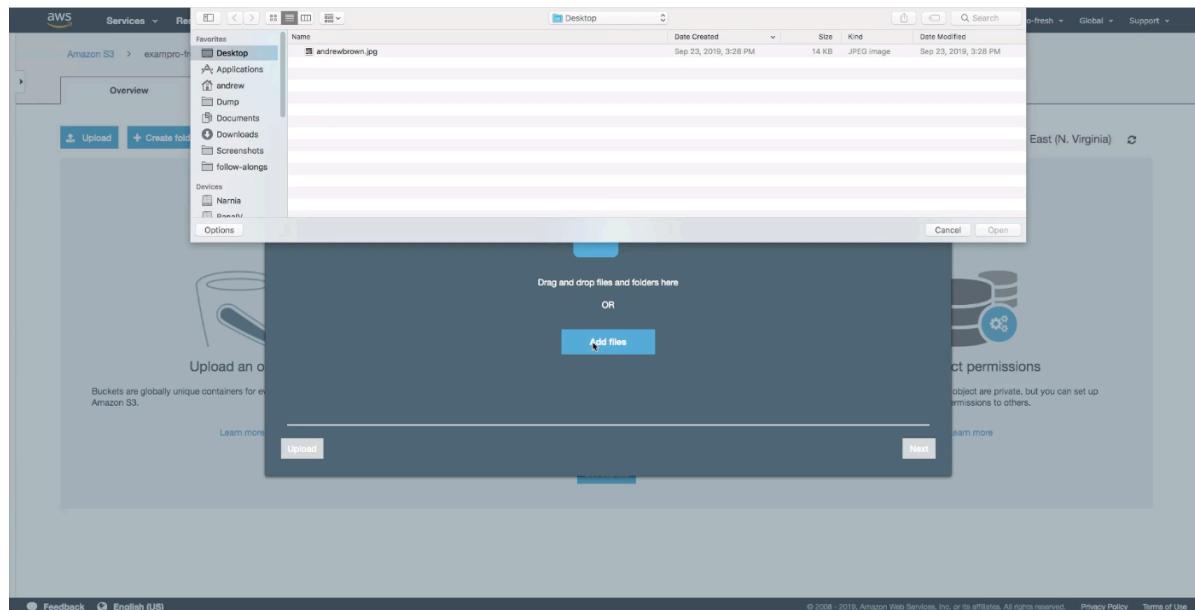
The screenshot shows the AWS S3 buckets list. At the top, there is a search bar labeled "Search for buckets" and a button labeled "+ Create bucket". Below the search bar are buttons for "Edit public access settings", "Empty", and "Delete". On the right side of the header, there are links for "Documentation" and "Discover the console". The main table lists one bucket:

Bucket name	Access	Region	Date created
exapro-fresh	Bucket and objects not public	US East (N. Virginia)	Oct 8, 2019 2:54:59 PM GMT-0400

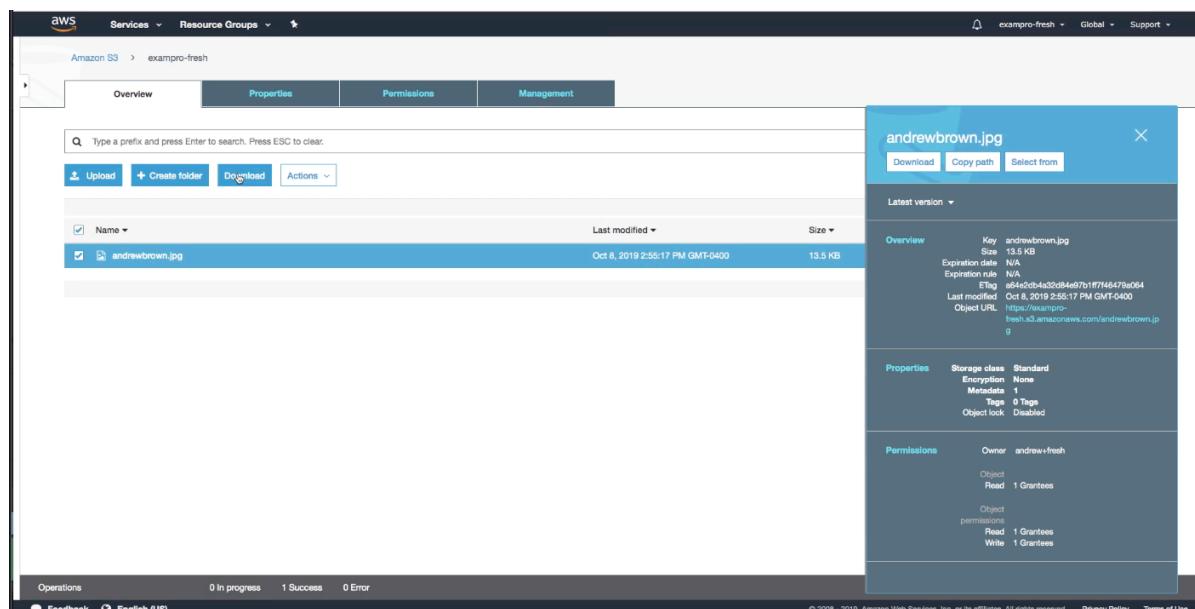
- 3.2 [ ] Click the Upload button

The screenshot shows the "Upload" wizard for AWS S3. The first step, "Select files", is active. A red box highlights the "Upload" button on the left side of the interface. The central area contains instructions: "To upload a file larger than 160 GB, use the AWS CLI, AWS SDK, or Amazon S3 REST API. Learn more." Below this are two options: "Drag and drop files and folders here" and "Add files". At the bottom of the wizard are four numbered steps: 1. Select files, 2. Set permissions, 3. Set properties, and 4. Review. The "Next" button is located at the bottom right of the wizard window.

- 3.3 [] Drop in a file from your computer like an image
- 3.4 [] Click the Upload button in the bottom right corner of the upload model



- 3.5 [] You should now see the file you uploaded in your S3 bucket

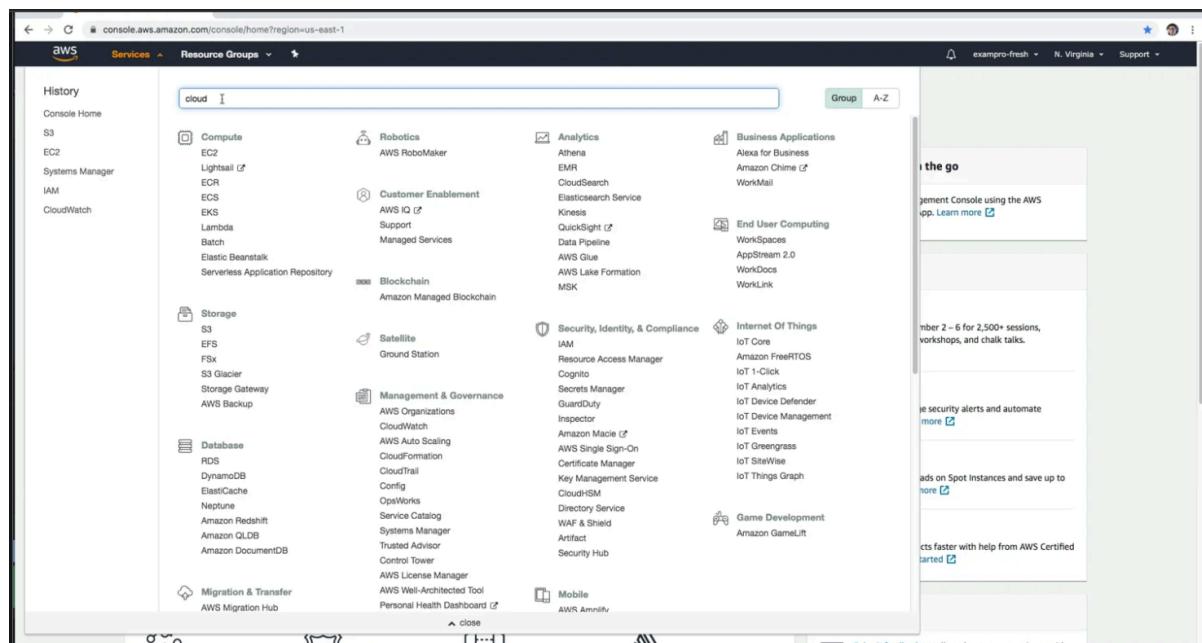


# Creating a CloudFront Distribution

We are going to walk through how to create a CloudFront Distribution

## Navigating to CloudFront

- 1.1 [] Inside the AWS Management Console use the search bar and type in CloudFront



## Creating a New Distribution

- 2.1 [] On the Amazon CloudFront Getting Started page click the "Create Distribution" button.

The screenshot shows the AWS CloudFront Getting Started page. On the left, there's a sidebar with 'CloudFront' selected. Under 'Distributions', it says 'What's new \*' and has a 'Create Distribution' button. Below that are sections for 'Reports & analytics' (Cache statistics, Monitoring, Alarms, Popular objects, Top referrers, Usage, Viewers) and 'Security' (Origin access identity, Public key, Field-level encryption). The main content area says 'Either your search returned no results, or you do not have any distributions. Click the button below to create a new CloudFront distribution. A distribution allows you to distribute content using a worldwide network of edge locations that provide low latency and high data transfer speeds (learn more)' and has a 'Create Distribution' button.

- 2.2 [ ] On the Create Distribution Form click the Origin Domain Name to get a drop down and choose the S3 bucket you just created.

The screenshot shows the 'Create Distribution' form. At the top, it says 'Step 1: Select delivery method' and 'Step 2: Create distribution'. The 'Origin Settings' section contains fields: 'Origin Domain Name' (exampopro-fresh.s3.amazonaws.com), 'Origin Path' (/), 'Origin ID' (S3-exampopro-fresh), 'Restrict Bucket Access' (radio buttons for Yes and No, with No selected), and 'Origin Custom Headers' (Header Name and Value fields). The 'Default Cache Behavior Settings' section includes: 'Path Pattern' (Default), 'Viewer Protocol Policy' (radio buttons for HTTP and HTTPS, Redirect HTTP to HTTPS, and HTTPS Only, with HTTP and HTTPS selected), 'Allowed HTTP Methods' (radio buttons for GET, HEAD, OPTIONS, and GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE, with GET, HEAD, OPTIONS selected), 'Field-level Encryption Config' (dropdown menu), 'Cached HTTP Methods' (dropdown menu showing 'GET, HEAD (Cached by default)'), 'Cache Based on Selected Request Headers' (dropdown menu showing 'None (Improves Caching)'), 'Object Caching' (radio buttons for Use Origin Cache Headers and Customize, with Use Origin Cache Headers selected), and 'Minimum TTL' (input field with value 0).

- 2.3 [ ] At the bottom of the page click the "Create Distribution" button.

Step 1: Select delivery method  
Step 2: Create distribution

**SSL Certificate**  Default CloudFront Certificate (\*.cloudfront.net)  
Choose this option if you want your users to use HTTPS or HTTP to access your content with the CloudFront domain name (such as https://d11t514v026oss.cloudfront.net/logging).  
Important: If you choose this option, CloudFront requires that browsers or devices support TLSv1 or later to access your content.

Custom SSL Certificate (example.com)  
Choose this option if you want your users to access your content by using an alternate domain name, such as https://www.example.com/logo.png.  
You can use a certificate stored in AWS Certificate Manager (ACM) in the US East (N. Virginia) Region, or you can use a certificate stored in IAM.

**Supported HTTP Versions**  HTTP/2, HTTP/1.1, HTTP/1.0  HTTP/1.1, HTTP/1.0

**Default Root Object**

**Logging**  On  Off

**Bucket for Logs**

**Log Prefix**

**Cookie Logging**  On  Off

**Enable IPv6**  Learn more

**Comment**

**Distribution State**  Enabled  Disabled

Cancel Back **Create Distribution**

- 2.4 [ ] Confirm that the distribution is in progress and you have a unique domain name

#### CloudFront Distributions

**Create Distribution** **Distribution Settings** **Delete** **Enable** **Disable**

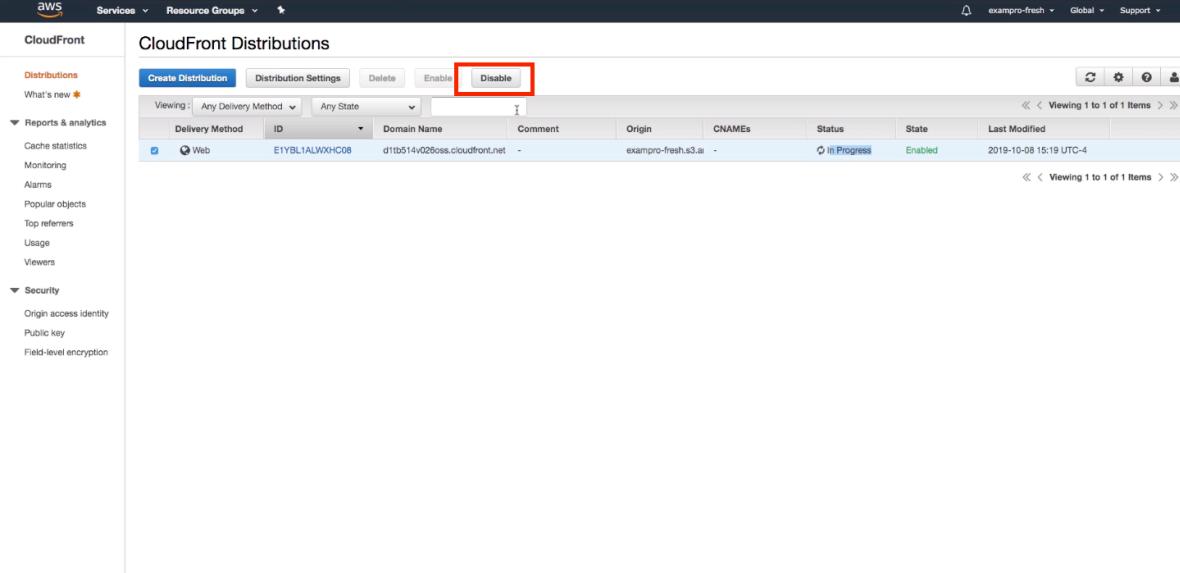
Viewing: Any Delivery Method ▾ Any State ▾

Delivery Method	ID	Domain Name	Comment	Origin	CNAMEs	Status	State	Last Modified
Web	E1YBL1ALWXC08	d1t514v026oss.cloudfront.net	-	exampro-fresh.s3.amazonaws.com	-	In Progress	<b>Enabled</b>	2019-10-08 15:19 UTC-4

« < Viewing 1 to 1 of 1 items > »

## Disabling the Distribution

- 3.1 [ ] Click the Disable button to disable your CloudFront Distribution



The screenshot shows the AWS CloudFront Distributions page. On the left, there's a sidebar with 'CloudFront' selected. Under 'Distributions', it shows 'Create Distribution' and 'Distribution Settings' buttons, followed by 'Delete', 'Enable', and 'Disable' buttons. The 'Disable' button is highlighted with a red box. Below these are dropdown menus for 'Viewing', 'Delivery Method', and 'Any State'. A table lists one distribution:

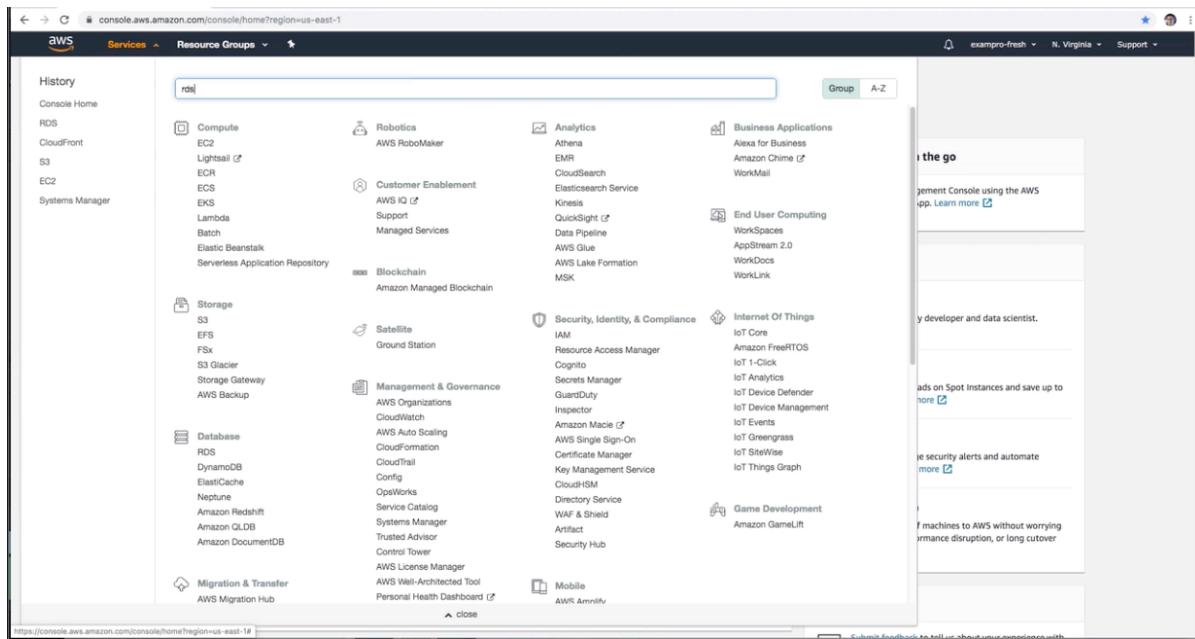
Delivery Method	ID	Domain Name	Comment	Origin	CNAMEs	Status	State	Last Modified
Web	E1YBL1ALUWXC08	d1tb514v026oss.cloudfront.net	-	exampro-fresh.s3.us-east-1.amazonaws.com	-	In Progress	Enabled	2019-10-08 15:19 UTC-4

## Creating a Relational Database

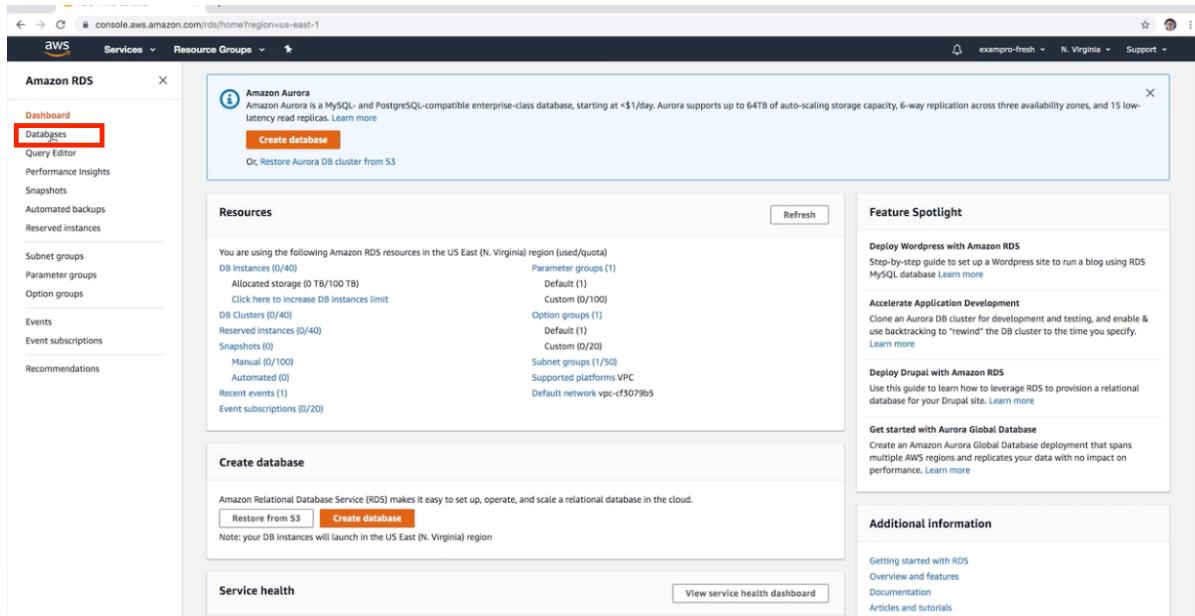
We are going to walk through how to create a Relational Database

### Creating a RDS

- 1.1 [ ] Inside the AWS Management Console use the search bar and type in RDS



## 1.2 [ ] Click the "Databases" link



## 1.3 [ ] Click the "Create database" button

**Amazon Aurora**

Amazon Aurora is a MySQL- and PostgreSQL-compatible enterprise-class database, starting at <\$1/day. Aurora supports up to 64TB of auto-scaling storage capacity, 6-way replication across three availability zones, and 15 low-latency read replicas. Learn more

**Create database**

Or, Restore Aurora DB cluster from S3

**Resources**

You are using the following Amazon RDS resources in the US East (N. Virginia) region (used/quota)

- DB Instances (0/40)
- Allocated storage (0 TB/100 TB)
- Click here to increase DB instances limit
- DB Clusters (0/40)
- Reserved instances (0/40)
- Snapshots (0)
  - Manual (0/100)
  - Automated (0)
- Recent events (1)
- Event subscriptions (0/20)

Parameter groups (1)
 

- Default (1)
- Custom (0/100)

Option groups (1)
 

- Default (1)
- Custom (0/20)

Subnet groups (1/50)
 

- Supported platforms VPC
- Default network vpc-cf3079b5

**Create database**

Amazon Relational Database Service (RDS) makes it easy to set up, operate, and scale a relational database in the cloud.

**Service health**

View service health dashboard

**Feature Spotlight**

**Deploy Wordpress with Amazon RDS**  
Step-by-step guide to set up a Wordpress site to run a blog using RDS MySQL database Learn more

**Accelerate Application Development**  
Clone an Aurora DB cluster for development and testing, and enable & use backtracking to "rewind" the DB cluster to the time you specify. Learn more

**Deploy Drupal with Amazon RDS**  
Use this guide to learn how to leverage RDS to provision a relational database for your Drupal site. Learn more

**Get started with Aurora Global Database**  
Create an Amazon Aurora Global Database deployment that spans multiple AWS regions and replicates your data with no impact on performance. Learn more

**Additional information**

Getting started with RDS  
Overview and features  
Documentation  
Articles and tutorials  
Data import guide for MySQL

## ■ 1.4 [ ] Choose the PostgreSQL engine option

**Create database**

**Choose a database creation method**

Standard Create

You set all of the configuration options, including ones for availability, security, backups, and maintenance.

Easy Create

Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

**Engine options**

**Engine type**

- Amazon Aurora
- MySQL
- MariaDB
- PostgreSQL
- Oracle
- Microsoft SQL Server

**Version**

PostgreSQL 11.5-R1

If you want to create PostgreSQL 12 in the Preview environment, click here.

**Templates**

Choose a sample template to meet your use case.

## ■ 1.5 [ ] Choose the Free Tier option under Templates

The screenshot shows the AWS RDS 'Create New DB Instance' wizard. In the 'Templates' section, the 'Free tier' option is selected and highlighted with a red box. Below it, the 'Settings' section contains fields for 'DB instance identifier' (set to 'database-1'), 'Master username' ('postgres'), and 'Master password'. The 'Master password' field is empty and highlighted with a red box.

## 1.6 [ ] Set a password

The screenshot shows the continuation of the AWS RDS wizard. The 'Free tier' template is still selected. In the 'Settings' section, the 'Master password' field now contains the value '\*\*\*\*\*' (redacted). A red box highlights this field. Below it, the 'Confirm password' field also contains '\*\*\*\*\*' and is also highlighted with a red box.

## 1.7 [ ] Leave the DB instance size on db.t2.micro

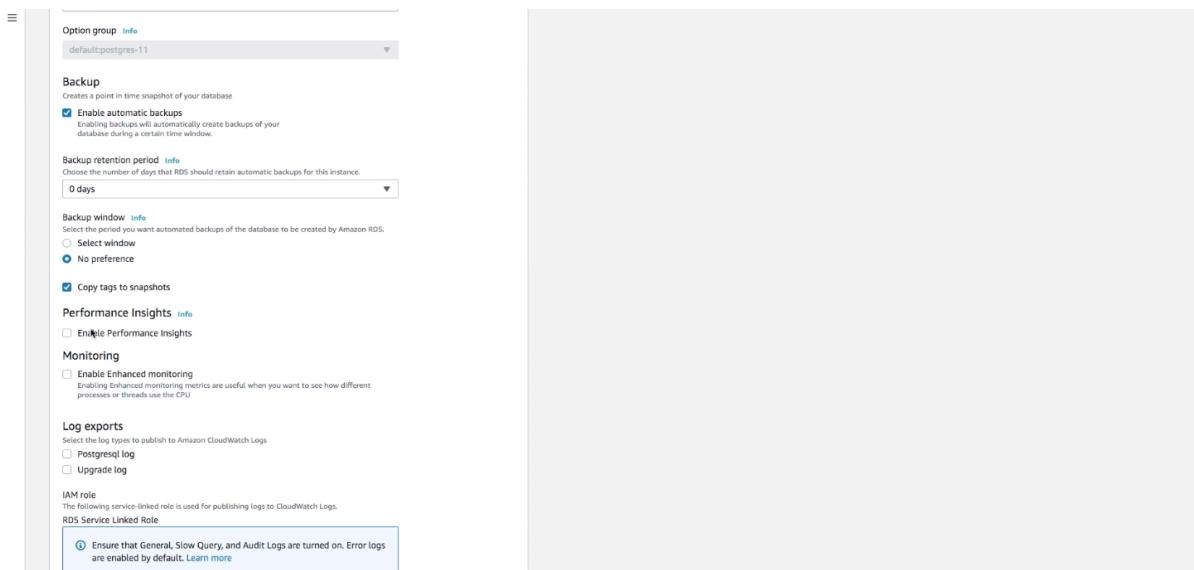
## 1.8 [ ] Disable the storage autoscaling

The screenshot shows the AWS RDS DB instance configuration page. Under the 'Storage' section, the 'Storage type' is set to 'General Purpose (SSD)'. The 'Allocated storage' is set to 20 GiB. A red box highlights the 'Storage autoscaling' section, which includes a checkbox for 'Enable storage autoscaling' and a 'Maximum storage threshold' input field set to 1000 GiB. The 'Availability & durability' section shows 'Multi-AZ deployment' is disabled.

- 1.9 [ ] Set up additional configuration with an initial database name
- 1.10 [ ] Turn backups off
- 1.11 [ ] Change backup retention period to 0 days

The screenshot shows the AWS RDS DB instance configuration page. Under the 'Backup' section, 'Enable automatic backups' is checked. Under 'Backup retention period', '0 days' is selected. Under 'Monitoring', 'Enable Enhanced monitoring' is checked. Under 'Log exports', 'Postgresql log' is selected. Under 'IAM role', a note says 'Ensure that General, Slow Query, and Audit Logs are turned on. Error logs are enabled by default.' A red box highlights this note.

- 1.12 [ ] Disable performance insights



### 1.13 [ ] Click the "Create database" button

When you are finished make sure to delete the database.