

How to Setup and Tear Down an Auto Scaling Group

In this documentation we will be learning what is an Auto Scaling Group or ASG and why is a pivotal service on the AWS cloud platform. Let us begin.

What is an Auto Scaling Group

An Auto Scaling group is a collection of virtual machines or EC2's that are logically grouped for the automatic scaling and management of necessary computing resources.

So an Auto Scaling Groups allows you to automatically increase/decrease your EC2'S or computer resources based on what is needed at the time. This agility of ASG's provide several benefits which are the following:

- They Scale out or add EC2's when there is an increased load of traffic data on the system, so the system itself is not overwhelmed to the degree of failure.
- They Scale in or remove EC2's when traffic data has decreased, saving the user from unnecessary supply costs that doesn't have demand.
- Makes sure there is a minimum and maximum amount of EC2's are operating, based on the scale policies defined by the user.
- Automatically registers new EC2's with the load balancer, so it can communicate as well as receive authorized traffic.
- Self-heals itself, meaning it replaces any EC2's that fails a health check of its functionally, therefore preventing traffic data from being lost when directed to a damaged/down machine.

What are EC2's

EC2 stands for an Elastic Compute Cloud. It is a virtual machine or computer that can be used to run applications.

EC2's are one of the core components of the AWS cloud platform because they allow users to rent all the computing power needed, saving them expensive upfront hardware computer costs. Users only pay for what they use, which can increase or decrease over time, making EC2's scalable/elastic/flexible to the demands of the client.

Features of Auto Scaling Groups

ASG's can launch configured EC2's with or without a template of an already built EC2. Once launched, ASG's keep EC2's running based on the *desired capacity*.

Desired Capacity is the favored amount of EC2's, the user wants running. An ASG may create or destroy EC2's to match the desired amount established before launch.

Properties of an ASG's virtual computer capacities are the **minimum amount, the least amount of EC2's** and the **maximum amount the largest amount of EC2's allowed** for the system to be operational.

Scaling policies are rules, that allow EC2's to automatically be created/deleted based on the metrics of the networks performance.

Examples of Scaling Policies are:

- Scheduled Scaling are time based modifications where EC2's are scaled out (an increased amount) for high demand or scaled in (a decreased amount) for low demand periods.
- Dynamic Scaling where the ASG reacts to the metrics of the network.
Dynamic scaling can be the following:

- Simple scaling, which changes the amount of EC2's based on a specific metrics value changing; like an increase or decrease of CPU (Central processing unit) usage.
- Stepped scaling, allows the user to set rules based on abnormal metric value changes. For example when a network receives spikes of traffic data, stepped scaling rules allows the network to react quickly increasing or decreasing EC2's rapidly to match the level of demand.
- Target tracking is when the user sets rules to maintain an ideal performance metric. For example, a user wants to maintain a CPU usage at 40%, therefore that target metric will determine how many EC2's are needed to ensure this.

In this documentation we will continue to use the WuCloud virtual private cloud network (VPC), two security groups, target group and load balancer previously built in other materials. We will create 2 EC2's which will be used as a template for the EC2's created by the Auto Scaling Group. The template EC2's will be placed in several subnets within the VPC. We'll test the application to see if its functional and lastly tear down the auto scaling group.

Let us begin.

Note: *For the CIDR block range of IP addresses for our VPC we will use the below pattern schema. This will help us in validating the functionality of our application, when we test it and see the various subnets we plan out in the VPC portion of this lab work.*

The WuCloud VPC CIDR block: 10.136.0.0/16

Public Subnets(Load Balancer Subnets):

Availability Zone A 10.136.1.0/24; AZB 10.136.2.0/24; AZC 10.136.3.0/24

Private Subnets(EC2 Front End):

AZA 10.136.12.0/24; AZB 10.136.14.0/24; AZC 10.136.16.0/24

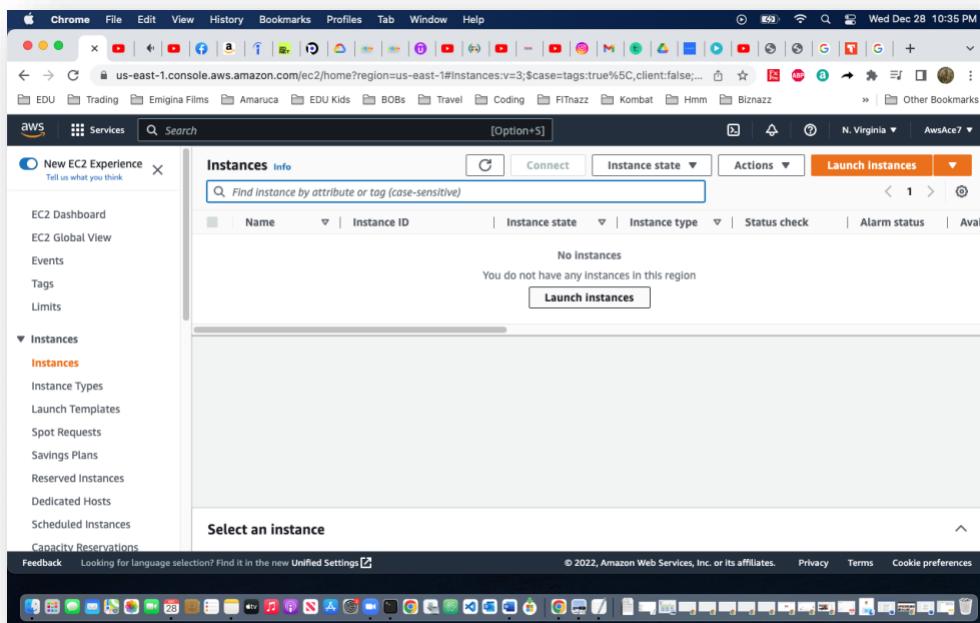
Private Subnets(Secure EC2's Back End):

AZA 10.136.73.0/24; AZB 10.136.75.0/24; AZC 10.136.77.0/24

Launching EC2's and EC2 Templates

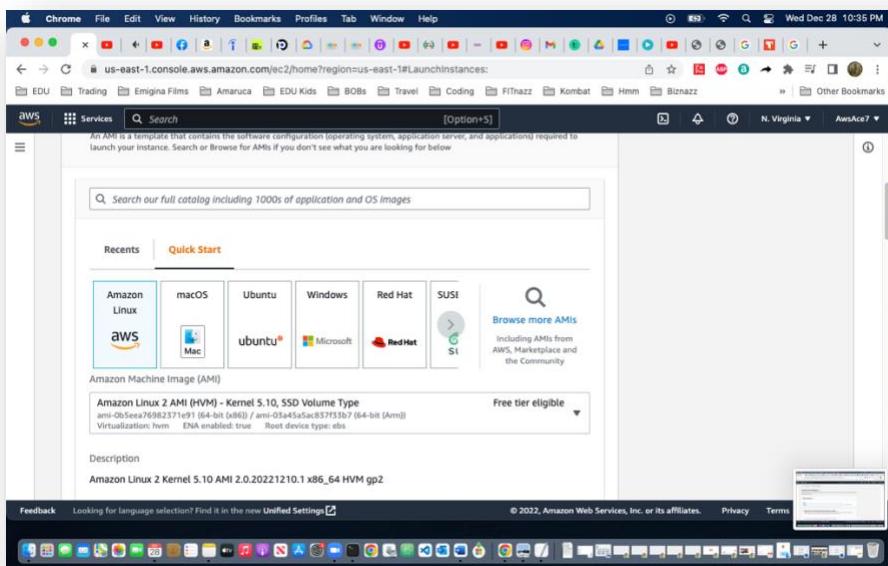
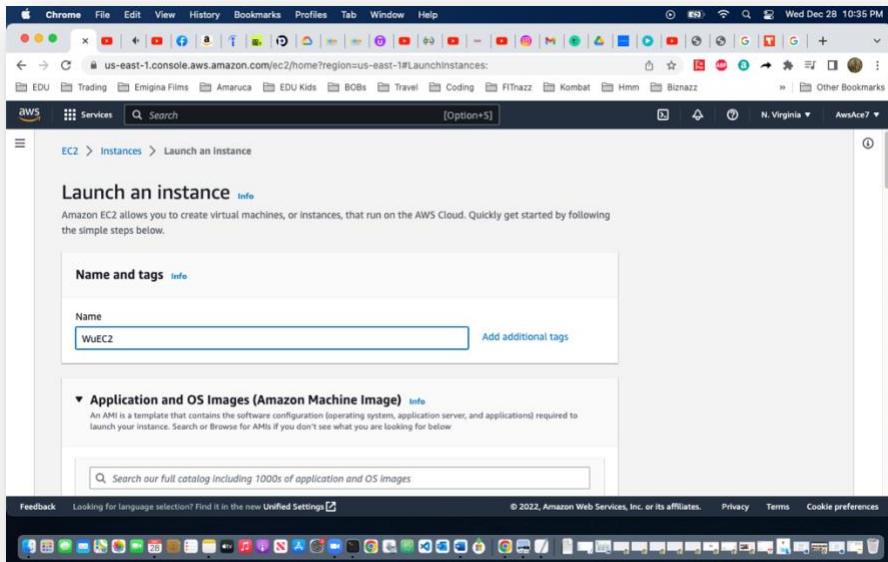
In our AWS account, the console management area, we go to the search bar, type EC2. We select the EC2 option to arrive at the EC2 Dashboard.

We then select “instances” from the left side menu. This brings us to the **Instance** main area. We then click on the button “launch instances”.

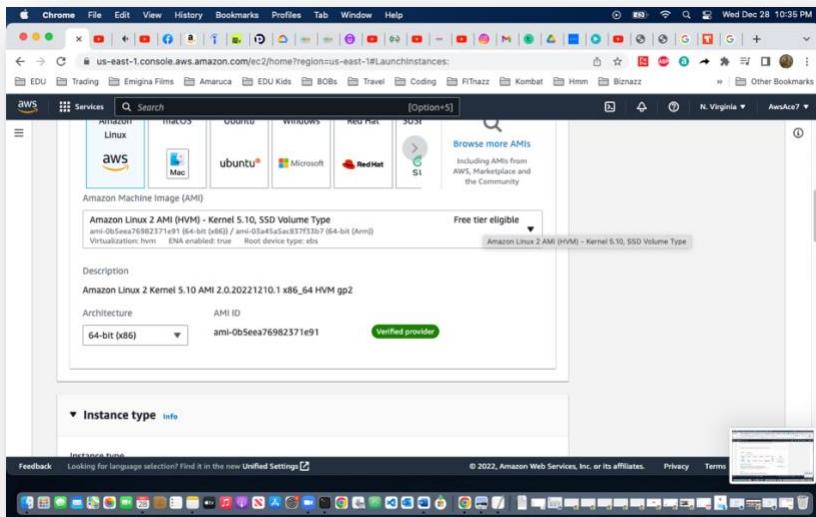


This is where we build our first instance or virtual computer. For the 1st input field of the **Launch an Instance** page, we will give our EC2 a name, with NO space.

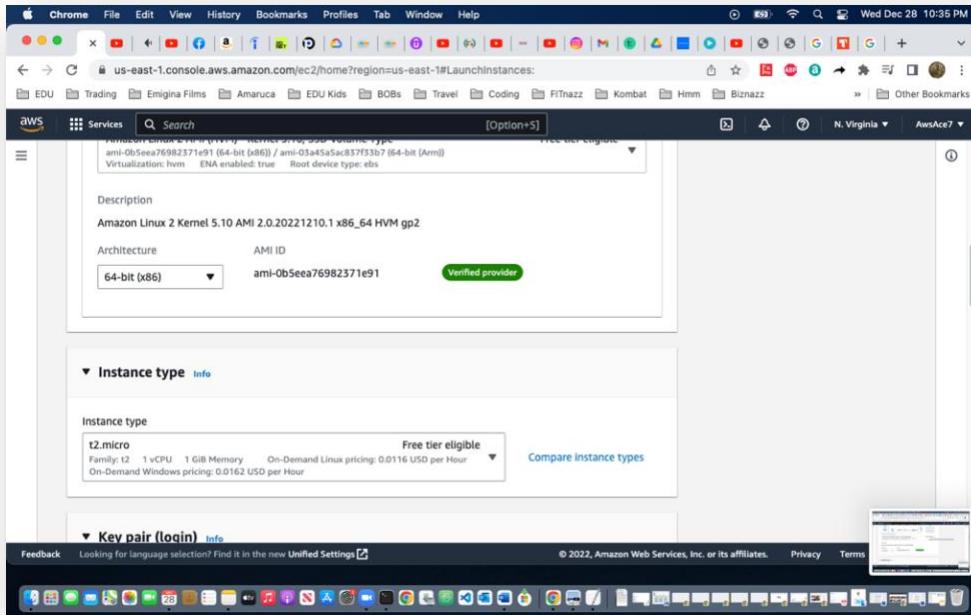
Next we will select an available operating system or OS for our computer under the **Application and OS Images** section. For this machine we will chose the “Amazon Linux”OS.



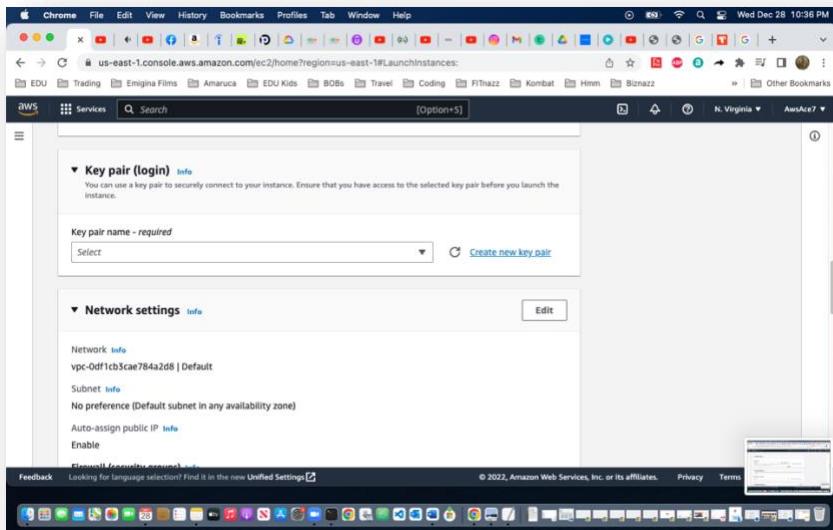
We selected this machine to be able to stay on the free tier, but there are plenty of other machine options based on whatever the needs of the user/client may be. Next we make sure the computers architecture is 64-bits is chosen.



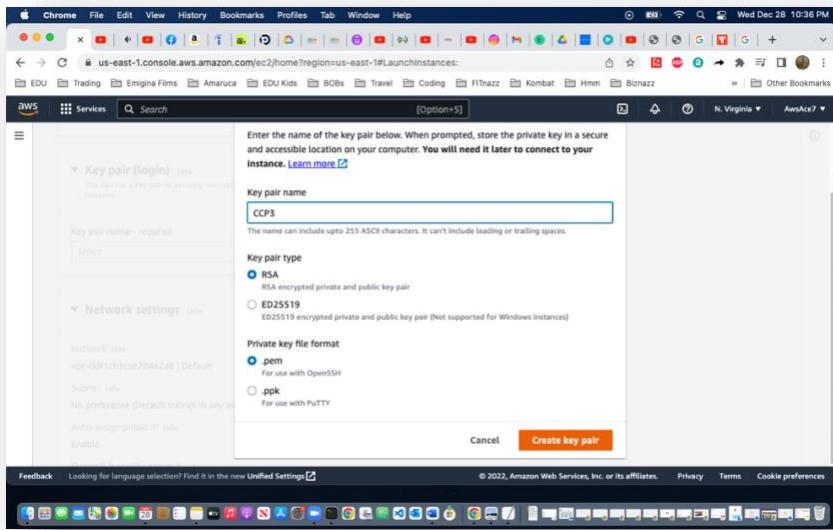
In the **Instance Type** section, we select the class, generation and size along computing specifications (such as memory, computing power etc) of the EC2. Here we chose the t2.micro(t being its class, 2 being 2nd generation EC2, micro its size) general purpose free tier machine. Remember there are also different types of EC2's based on the



Next we chose to create a **Key Pair**. A Key Pair are encrypted keys (public and private) that connect and verify identity, allowing the user to securely access their EC2 over the internet.

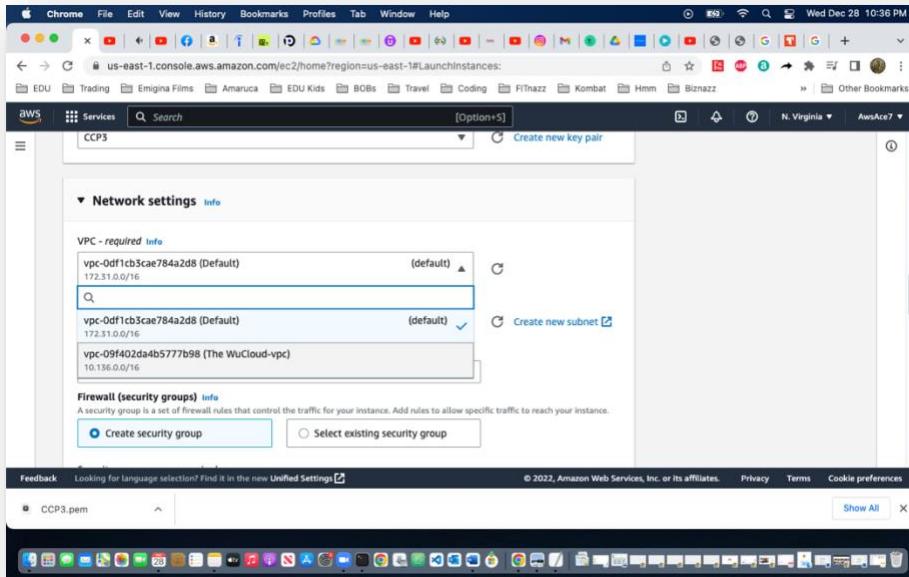


Here we name our key pair with NO spaces. Below we make sure “RSA” is selected; this option generates and verifies the digital signatures on the keys. Also make sure “.pem” selected for this is *privacy enhanced mail*, where the keys will be stored and downloaded to your computer. The other option is for Windows OS version 8,7 and earlier.

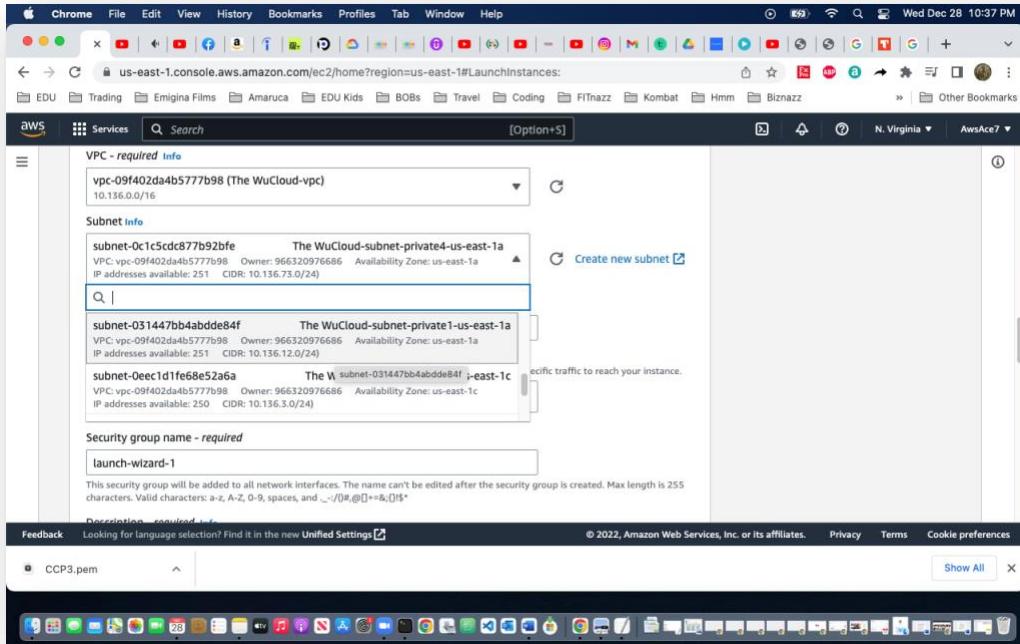


In the **Network Settings** section, we click the edit button on the right to begin modifications. For *VPC*, we make sure we choose our VPC network the WuCloud,

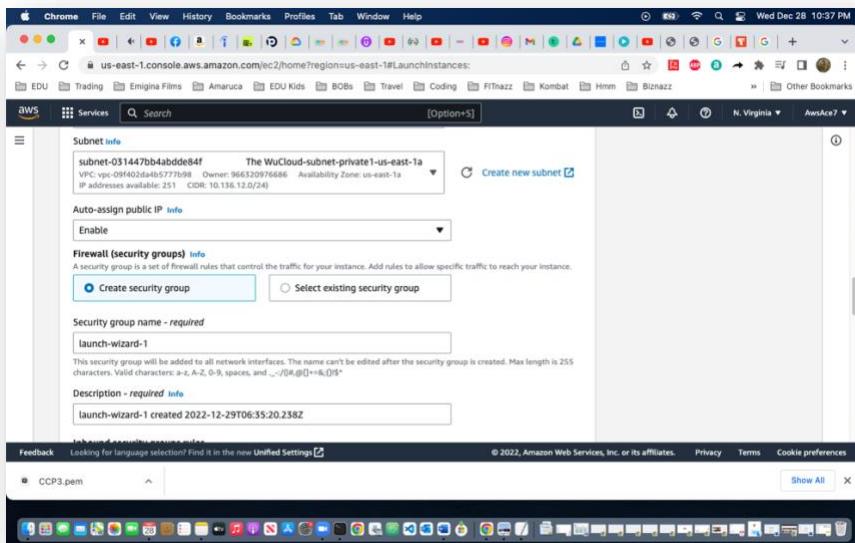
NOT default VPC.



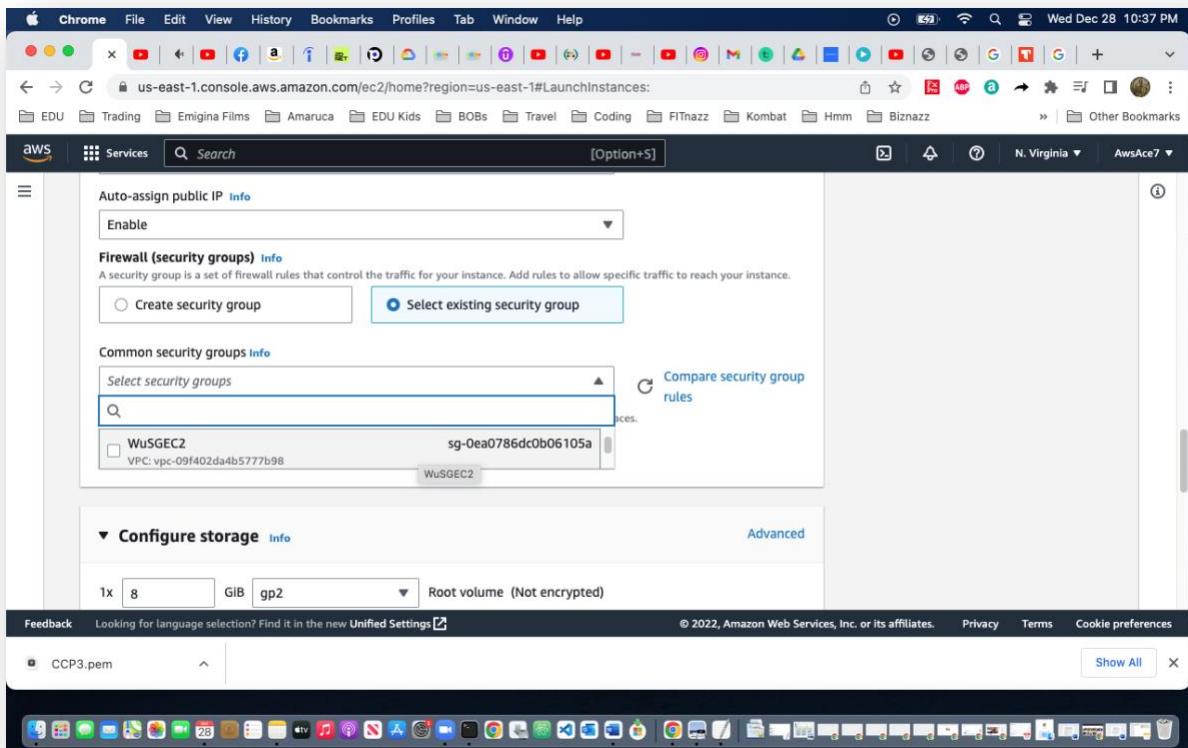
We go to the *Subnet* subsection and chose a private subnet in an availability zone. I like to chose *availability zone a* since it's the first one, which helps better organize my VPC design.



Then under “*Auto assgin public IP*” we select *Enable* this feature to avoid manually having to create our own IP address for the EC2.

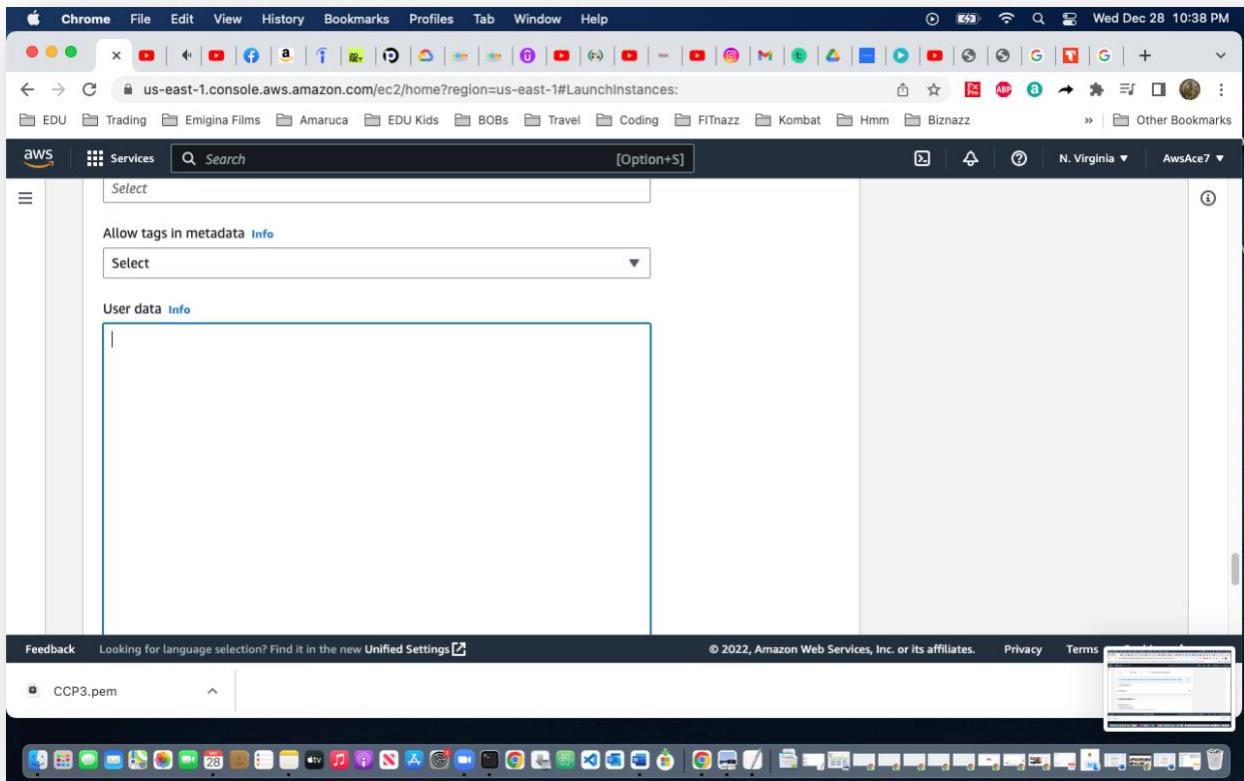
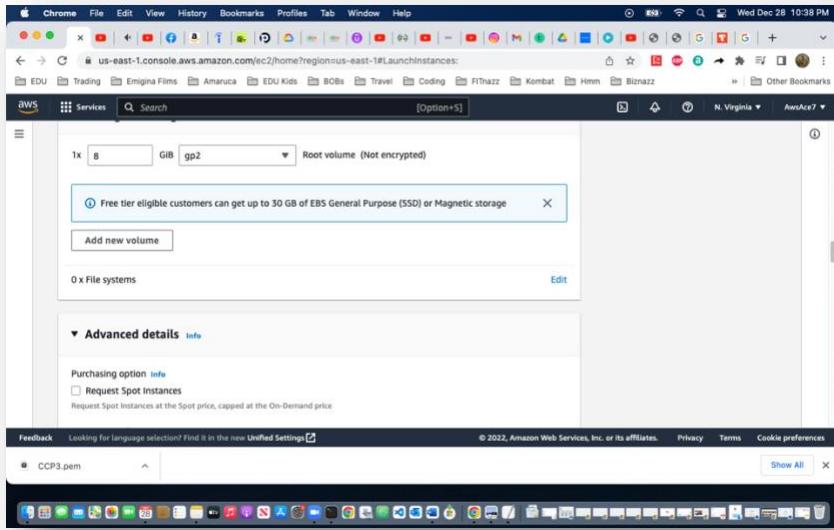


For “*Security Groups*” we checkmark the *Select existing security group* option, then choose the security group we created before for the EC2.

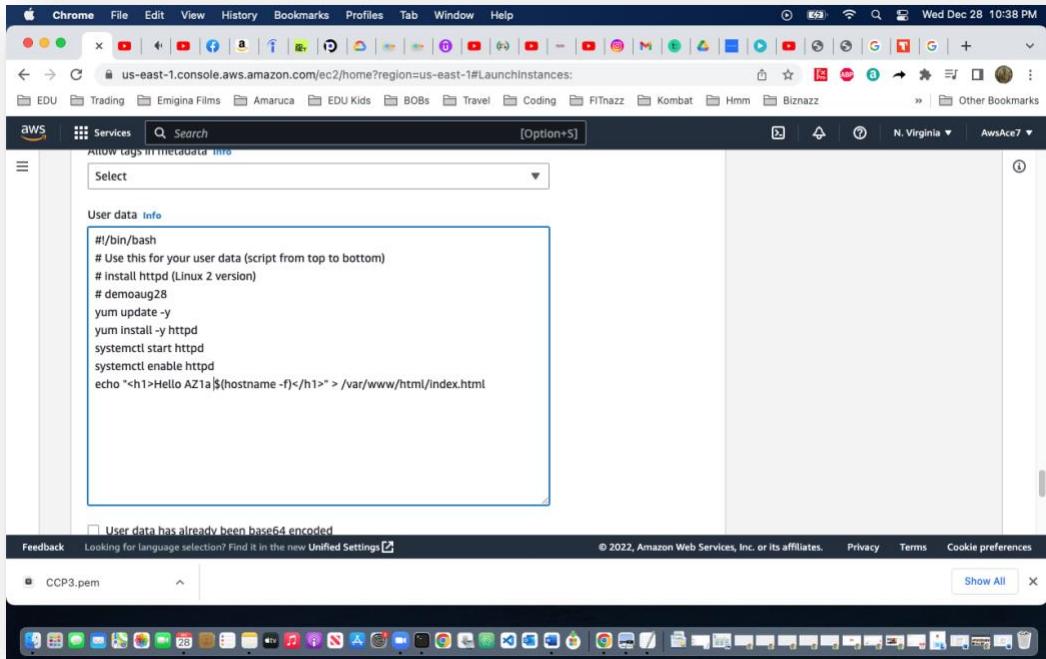


We scroll down to the **Advanced Details** section, where we will input launch task commands for the EC2. These commands instruct the EC2 on how to build itself

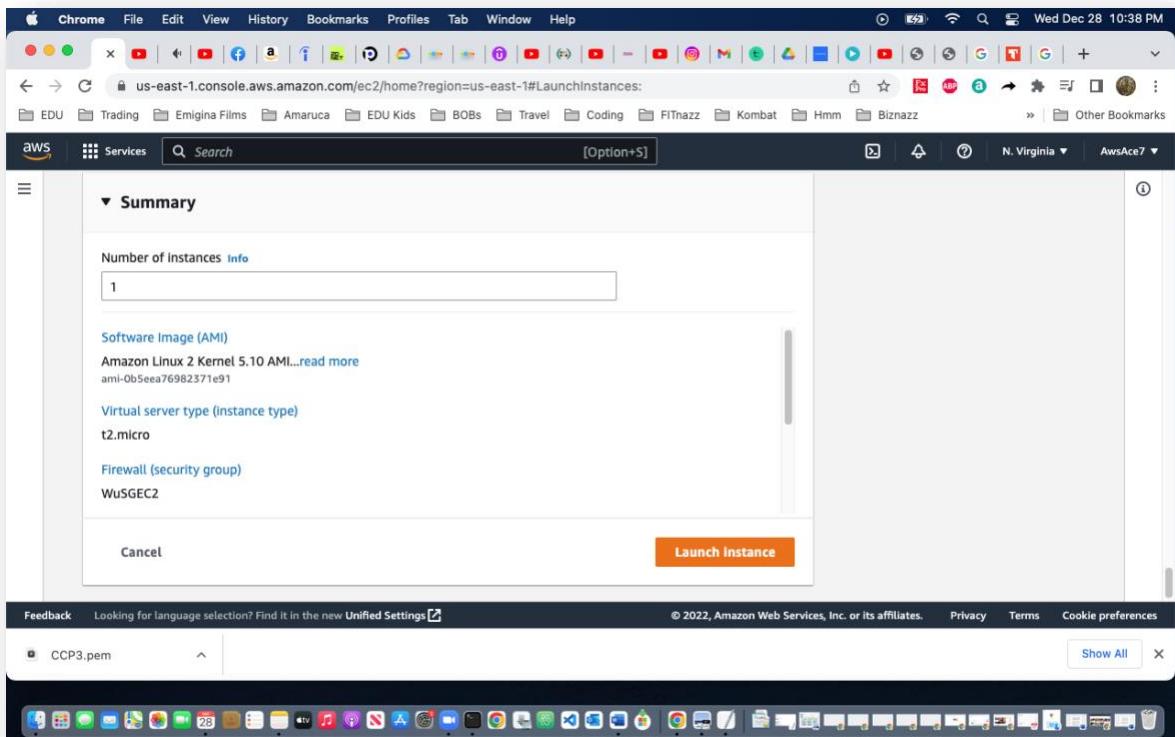
on its first launch. We click the dropdown arrow next to the sections title and scroll to the very bottom to the large input box, with the “*User data*” title.



Here you place the following bash script that helps configure the EC2 at launch.



In the final section, we go through the **Summary**, reviewing our EC2's specifications. If everything is accurate, we click the "*Launch Instance*" button.



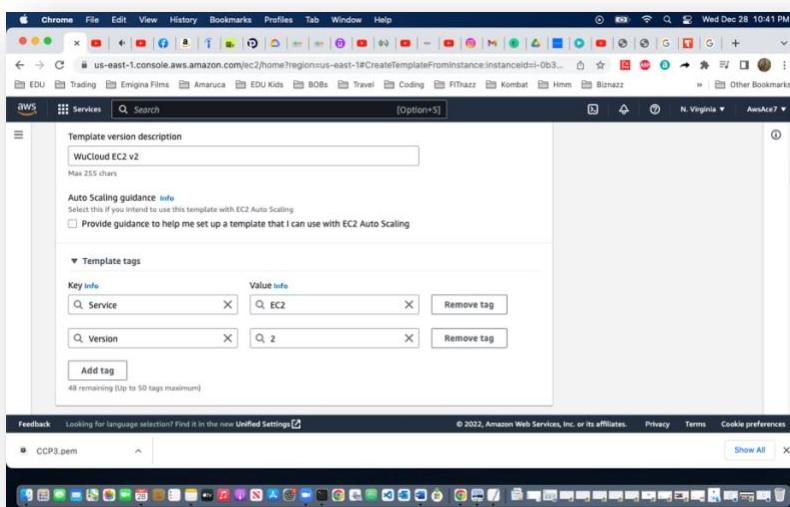
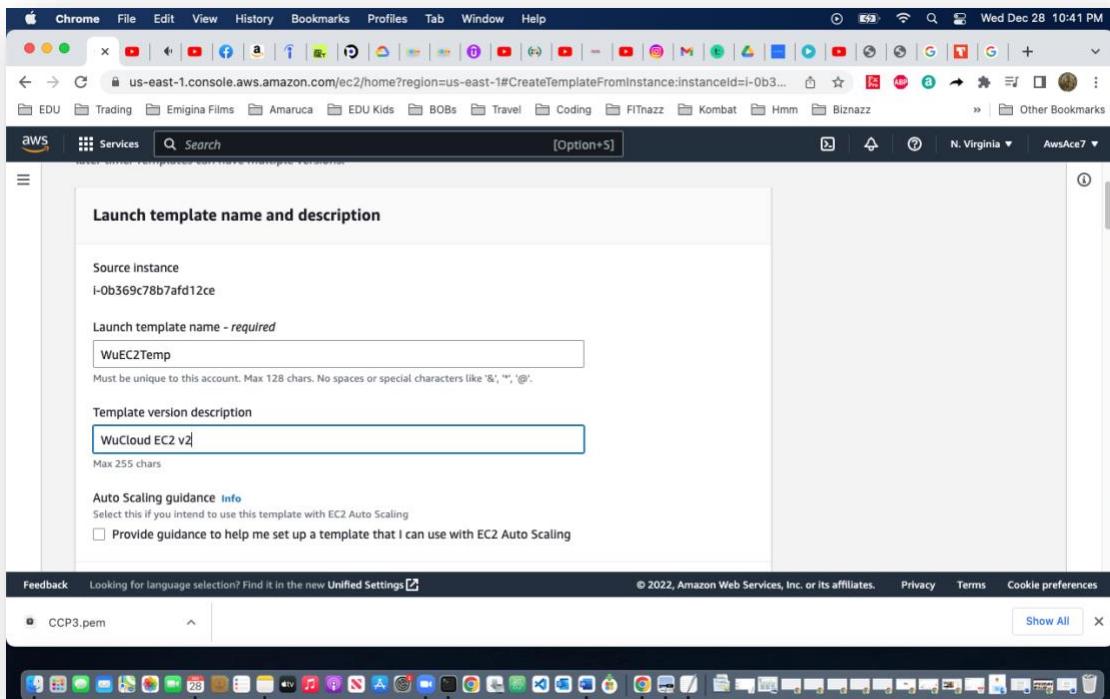
We can go to our left side menu, scroll to select the option “*Instances*” to view our newly formed instance. Here we can checkmark the box next to our new EC2. Go to the right side of the page to click the “*Actions*” button. Scroll to the option “*image and templates*”. We will choose “*create template from instance*”.

This screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with options like EC2 Dashboard, EC2 Global View, Events, Tags, Limits, and Instances (which is currently selected). Under Instances, there are sub-options: Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, and Dedicated Hosts. The main content area displays a table titled 'Instances (1/1) Info'. It shows one instance named 'WuEC2' with the ID 'i-0b369c78b7af12ce', which is 'Running' and of type 't2.micro'. To the right of the table is a vertical 'Actions' menu with several options: Connect, View details, Manage instance state, Instance settings, Networking, Security, Image and templates (which is expanded to show 'Create image' and 'Create template from instance'), and Monitor and troubleshoot. The 'Create template from instance' option is highlighted with a blue border.

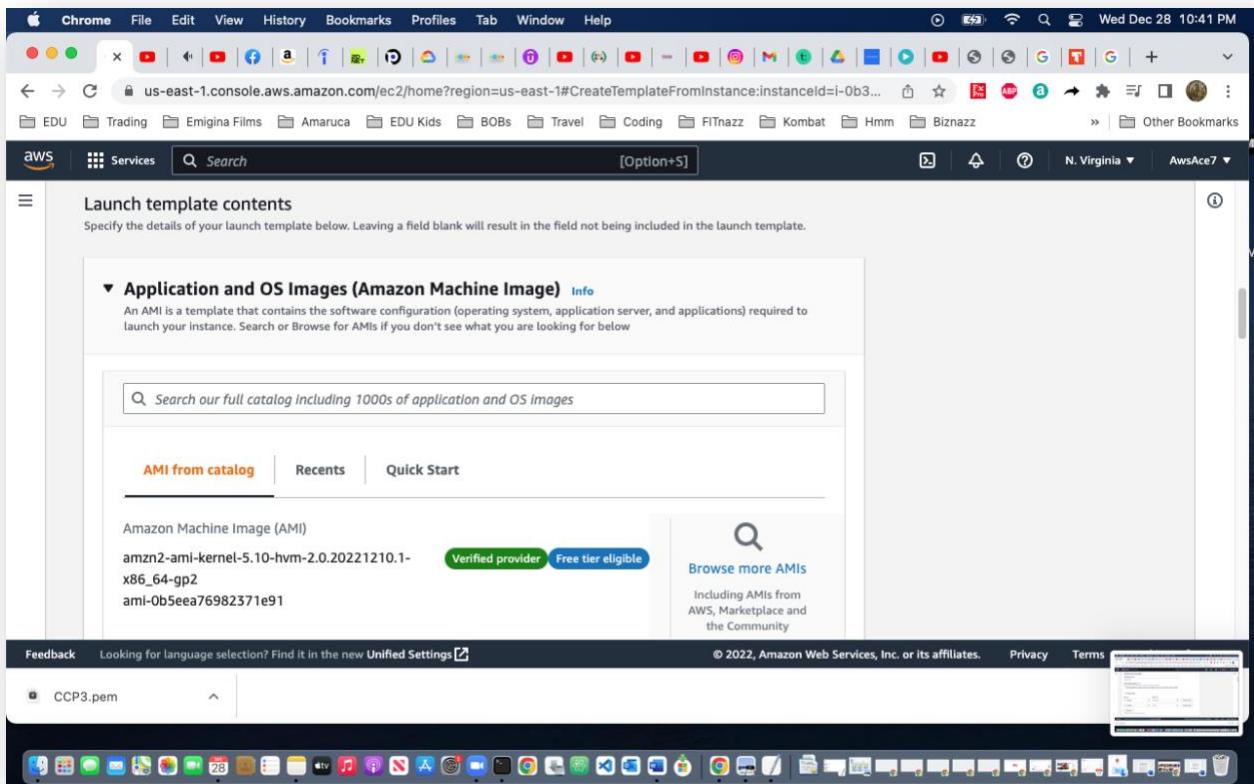
This screenshot is similar to the previous one, showing the AWS EC2 Instances page. The 'Actions' menu on the right has been interacted with, and the 'Create template from instance' option under the 'Image and templates' section is now highlighted with a blue border, indicating it is the selected action.

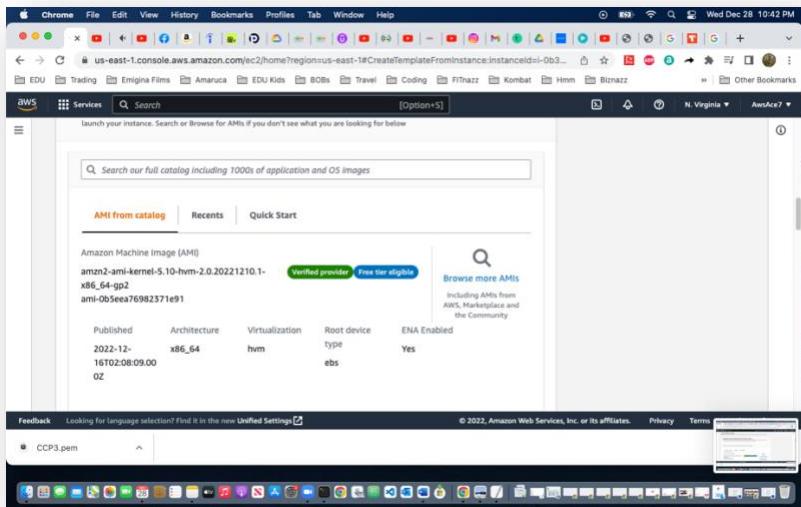
Creating an Instance Template from newly formed EC2

On the *Create Launch Template* page, we'll begin setting the template that our ASG will use to provision several instances at once. We start at **Launch template name and description** section, by typing a name for our template and which version we want to call it. You can add reference markers or *tags* to your template.

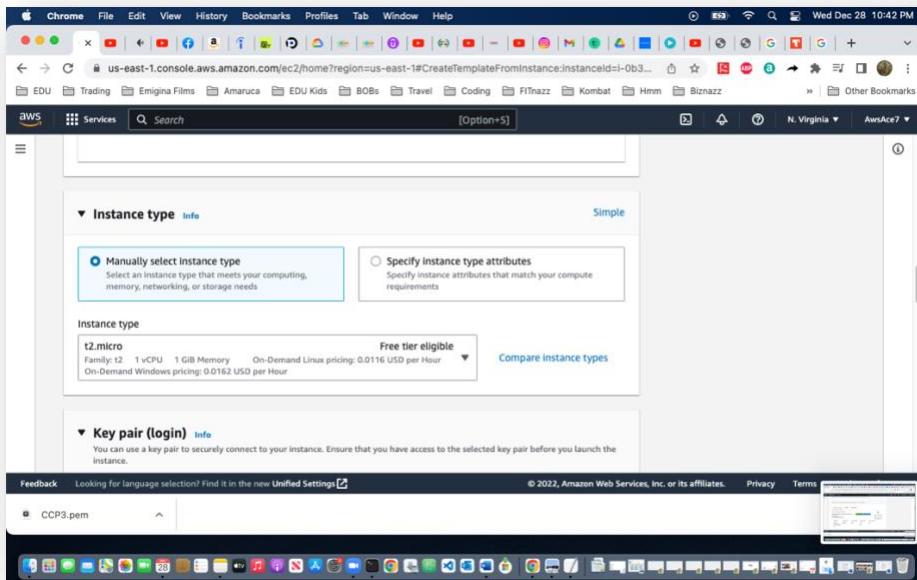


The beauty of creating a launch template is that all the properties used to create the EC2 are automatically imported into this setup. All we have to do is go over each section to make sure the data is accurate before we finalize the template. As you can see in the series of screen shots we're review our template credentials.

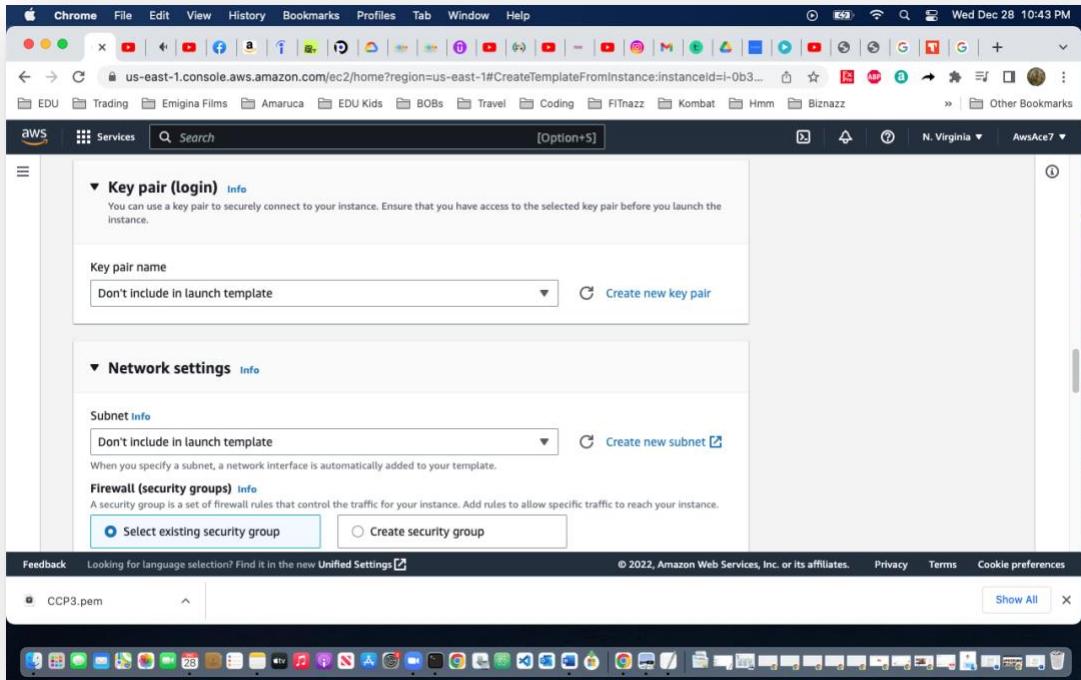




For **Instance Type** section we leave “manually select instance type” chosen, for it has already selected our t2.micro virtual computer from before.

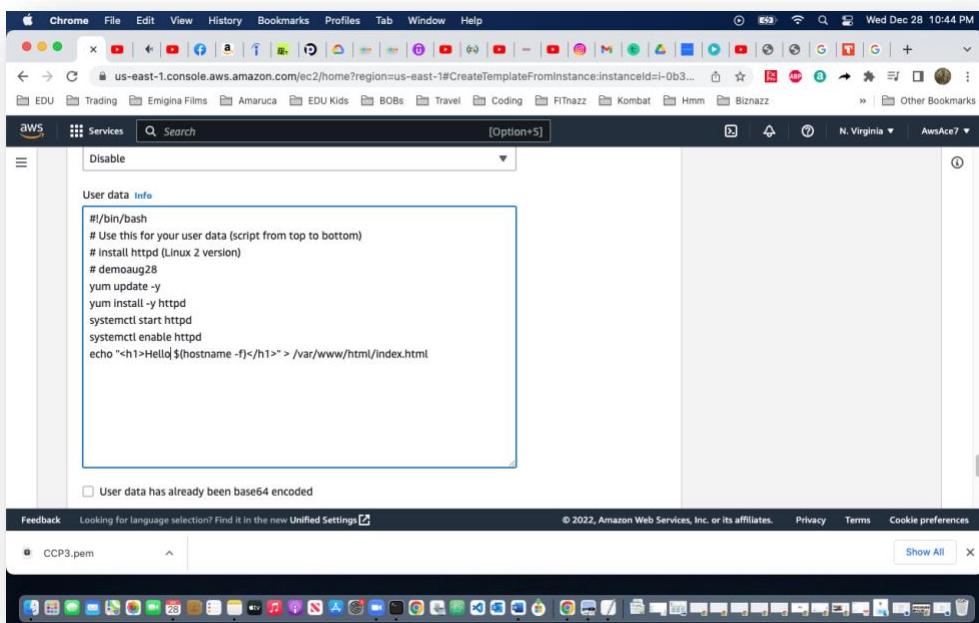
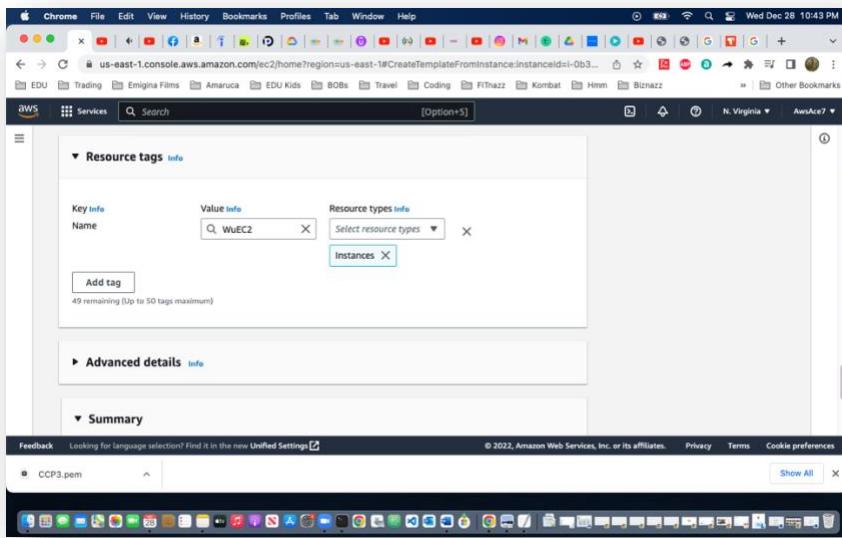


For sections **Key Pair Name** and **Network Subnet** we will chose the same dropdown choice for both with is “***Don’t include in launch template***”.



This option allows all future EC2's to be freely placed in different subnets within a network as well as preventing them from sharing the same key pairs, lessening possible security risks.

Next we scroll down to **Advanced Details** section to make sure our bash script is still in the “*User data*” field box. We review our template summary for errors, finally clicking the “*create launch template*” button. We should be greeted with a green “***Success***” banner across the screen.



This screenshot shows the first step of the 'Create launch template from instance' wizard in the AWS Management Console. The page title is 'Create launch template from instance'. It displays the configuration for a new launch template based on an existing instance. Key details include:

- Software Image (AMI):** Amazon Linux 2 Kernel 5.10 AMI... (with a 'read more' link)
- Virtual server type (instance type):** t2.micro
- Firewall (security group):** WuSGEC2
- Storage (volumes):** 1 volume(s) - 8 GiB

At the bottom right, there is a prominent orange 'Create launch template' button.

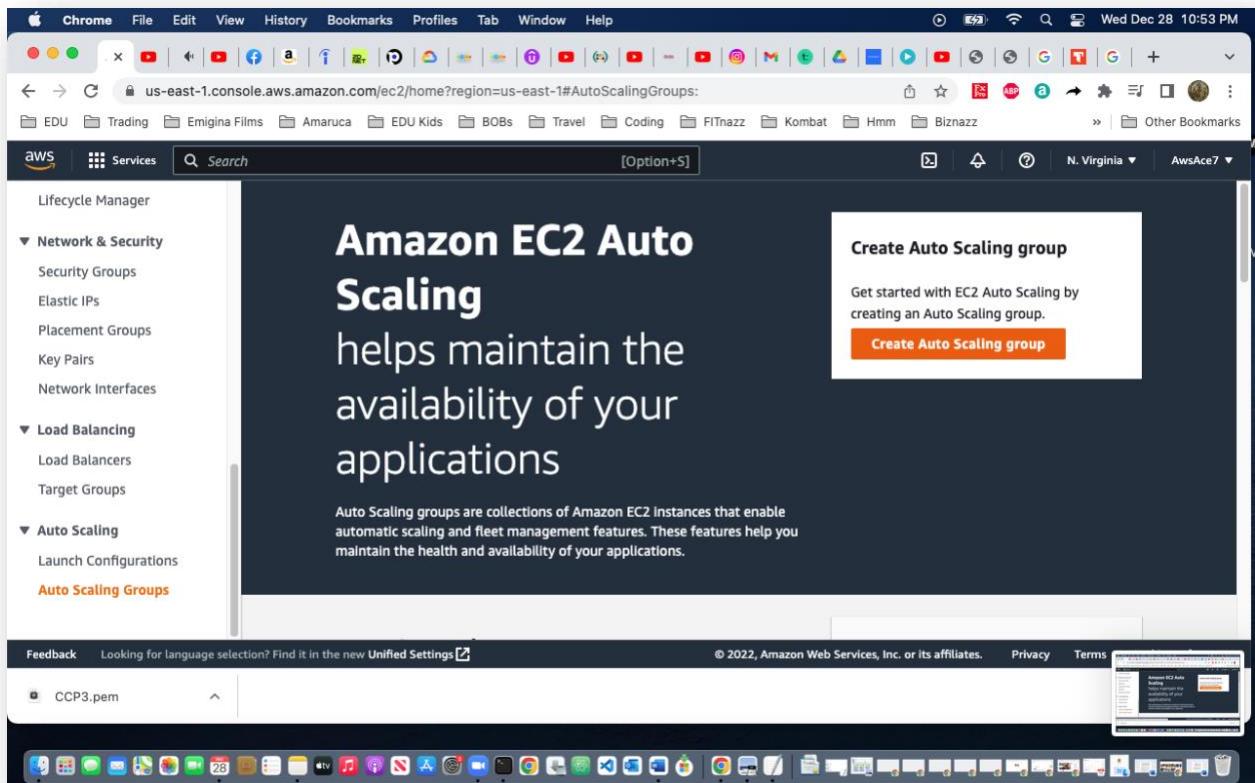
This screenshot shows the second step of the 'Create launch template from instance' wizard. The title bar indicates 'Success' and states 'Successfully created WuEC2Temp (lt-04a15bc77c55f6a68)'. Below this, the 'Actions log' section is visible. The main content area, titled 'Next steps', contains the following options:

- Launch an instance**: A brief description explaining On-Demand Instances and a link to 'Launch instance from this template'.
- Create an Auto Scaling group from your template**: A brief description about Auto Scaling and a link to 'Amazon EC2 Auto Scaling helps you maintain application availability and allows you to scale your Amazon EC2 capacity up or down automatically according to conditions you define. You can use Auto Scaling to help ensure that you are running your desired number of Amazon EC2 instances during demand spikes to maintain performance and availability.'

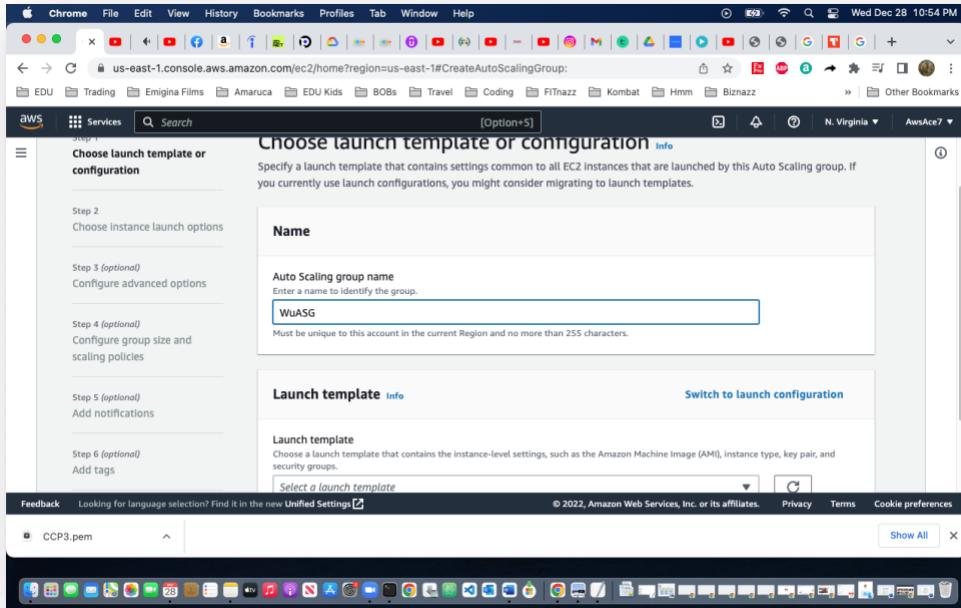
The browser's address bar shows the URL: <https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchTemplates>.

Creating an Auto Scaling Group w/ an EC2 Template

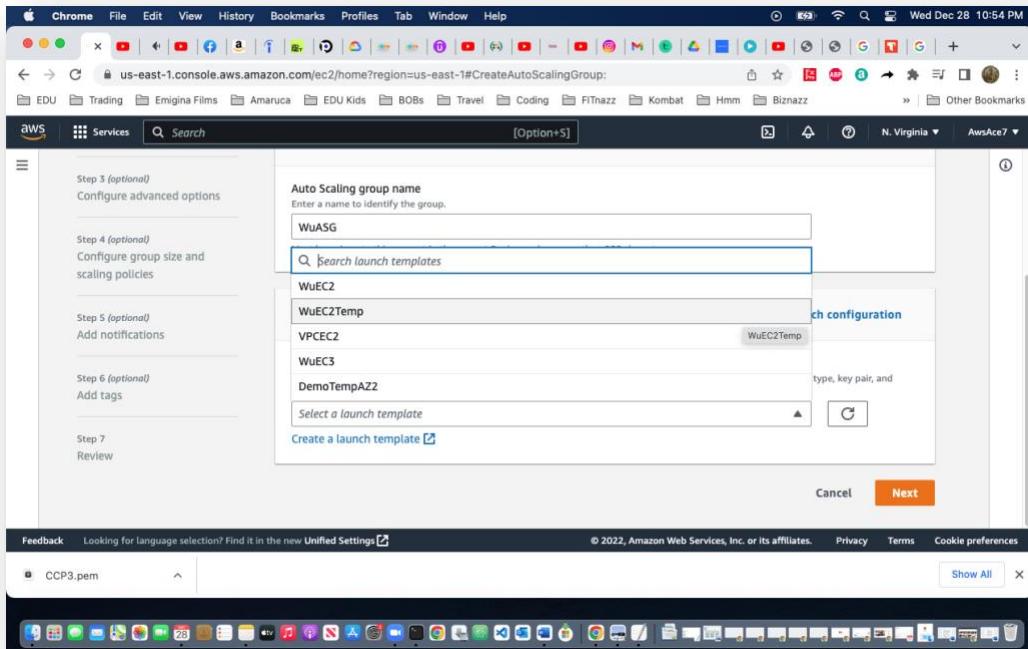
Go to the EC2 left side menu and scroll to select the option “Auto Scaling Group.” The ASG page should appear, where an “create auto scaling group” button is available on the right of the page. Click it to begin.



Here on the “**Choose launch template or configuration**” page we will begin setting up our ASG. In the **Name** section, type a name for your ASG.



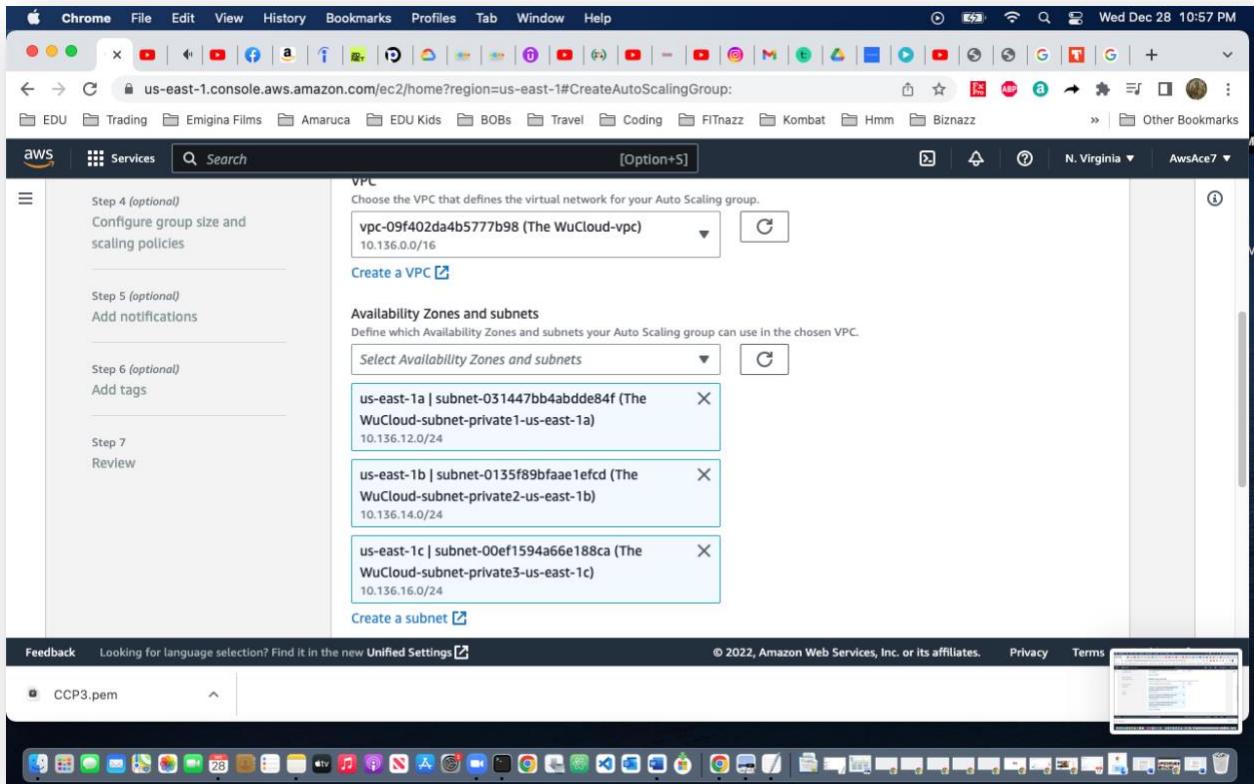
For the **LaunchTemplate** section we will choose the template we created from our original EC2. We select WuEC2Temp as you can see below.



Once our template is selected, all of its properties are again imported into this new segment. No configuring from scratch, just review and modification as we go. You can see for yourself in the details below of our EC2's credentials.

On the next page for “***Choose instance launch options***” for the **Network** section, we will choose our WuCloud VPC.

Now for the “*Availability Zones and Subnets*” subsection we will click the dropdown arrow and select all the ***private availability subnets that are double digits in the tens place***. For example we choose the 10.136.12.0/24, 10.136.14.0/24 and 10.136.16.0/24 subnets, which go across availability zones a, b, c.



For the **Load Balancing** section, we will attach our ASG to an existing Load Balancer, by way of the Target group, which listens for the kind of web traffic our load balancer responds to. We'll choose our load balancer target group as seen on the next page.

Chrome File Edit View History Bookmarks Profiles Tab Window Help

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateAutoScalingGroup: Wed Dec 28 10:57 PM

EDU Trading Emigina Films Amaruka EDU Kids BOBs Travel Coding Fitnazz Kombat Hmm Biznazz Other Bookmarks N. Virginia AwsAce7

aws Services Search [Option+S]

Step 2 Choose instance launch options

Step 3 (optional) Configure advanced options

Step 4 (optional) Configure group size and scaling policies

Step 5 (optional) Add notifications

Step 6 (optional) Add tags

Step 7 Review

Load balancing - optional Info

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

No load balancer Traffic to your Auto Scaling group will not be fronted by a load balancer.

Attach to an existing load balancer Choose from your existing load balancers.

Attach to a new load balancer Quickly create a basic load balancer to attach to your Auto Scaling group.

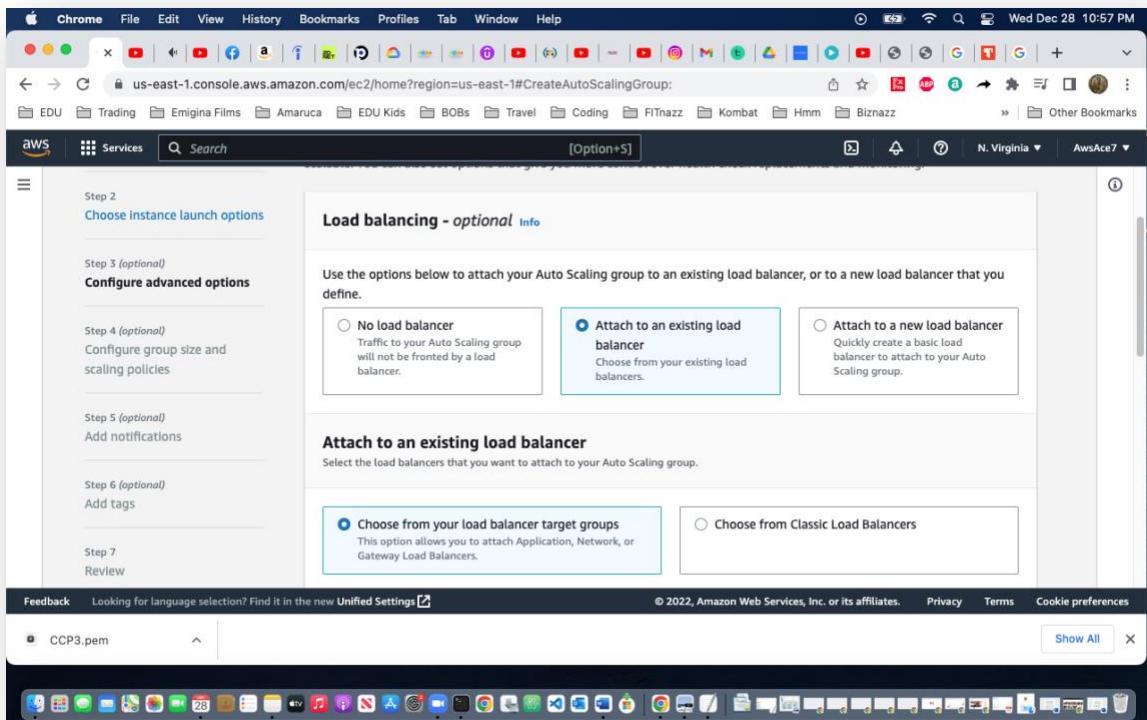
Attach to an existing load balancer

Select the load balancers that you want to attach to your Auto Scaling group.

Choose from your load balancer target groups This option allows you to attach Application, Network, or Gateway Load Balancers.

Choose from Classic Load Balancers

Feedback Looking for language selection? Find it in the new Unified Settings © 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences Show All



Chrome File Edit View History Bookmarks Profiles Tab Window Help

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateAutoScalingGroup: Wed Dec 28 10:57 PM

EDU Trading Emigina Films Amaruka EDU Kids BOBs Travel Coding Fitnazz Kombat Hmm Biznazz Other Bookmarks N. Virginia AwsAce7

aws Services Search [Option+S]

Step 5 (optional) Add notifications

Step 6 (optional) Add tags

Step 7 Review

Attach to an existing load balancer

Select the load balancers that you want to attach to your Auto Scaling group.

Choose from your load balancer target groups This option allows you to attach Application, Network, or Gateway Load Balancers.

Choose from Classic Load Balancers

Existing load balancer target groups

Only instance target groups that belong to the same VPC as your Auto Scaling group are available for selection.

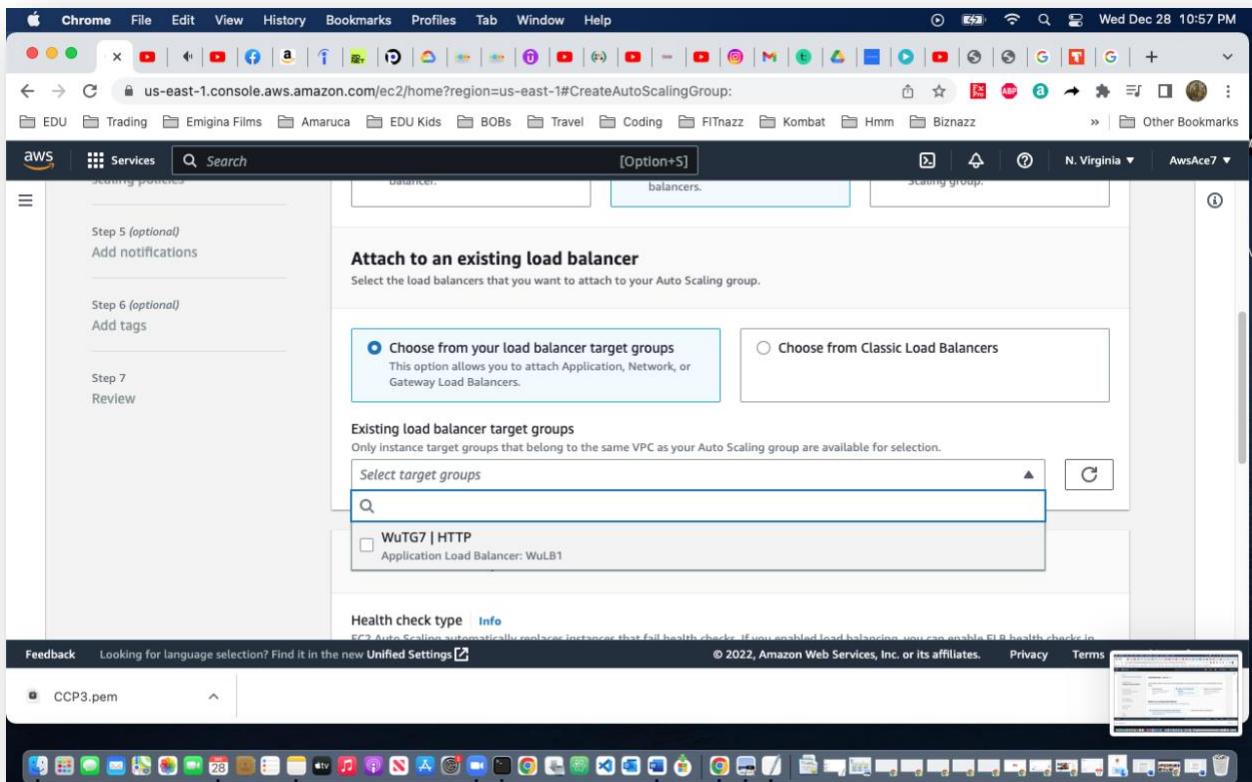
Select target groups

WuTG7 | HTTP Application Load Balancer: WuLB1

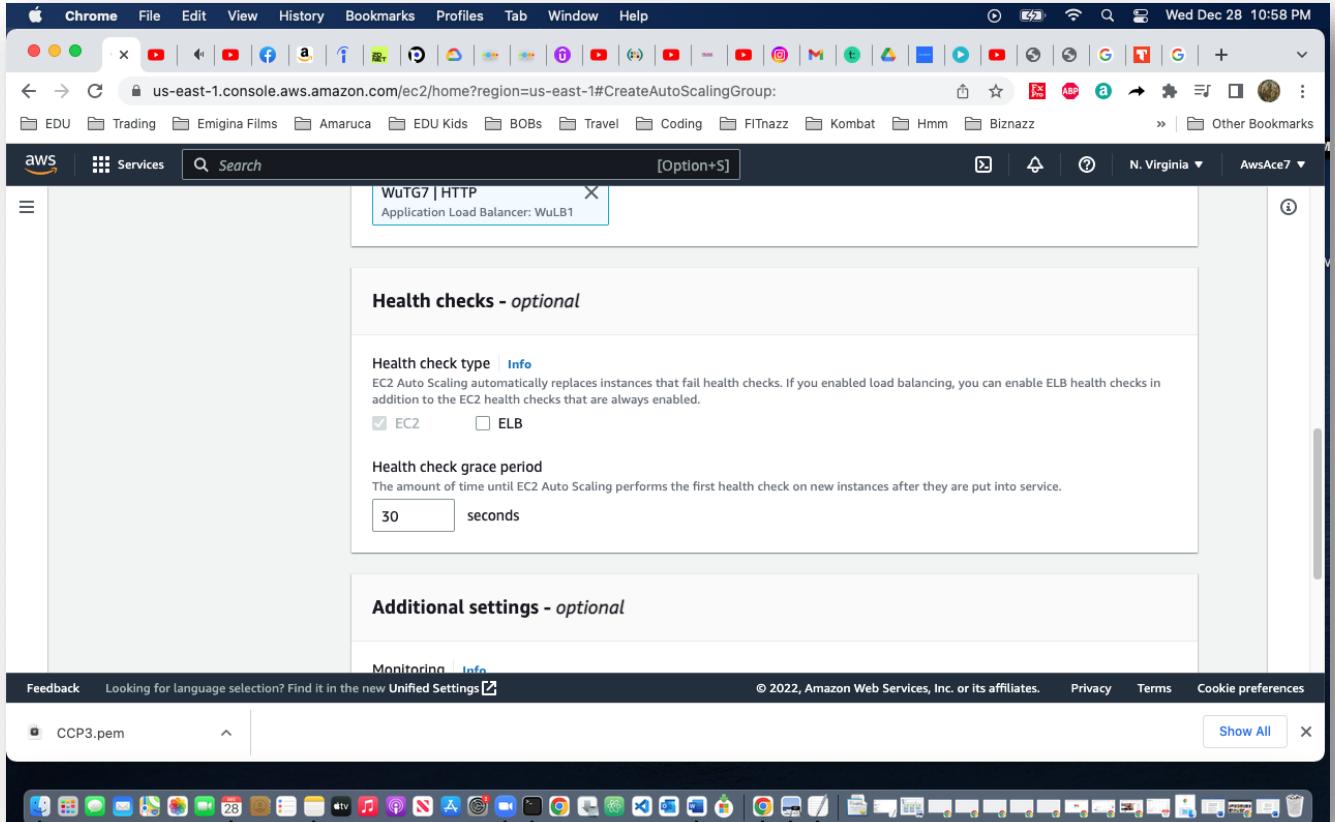
Health check type Info

EC2 Auto Scaling automatically replaces instances that fail health checks. If you enabled load balancer, you can enable ELB health checks in

Feedback Looking for language selection? Find it in the new Unified Settings © 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms

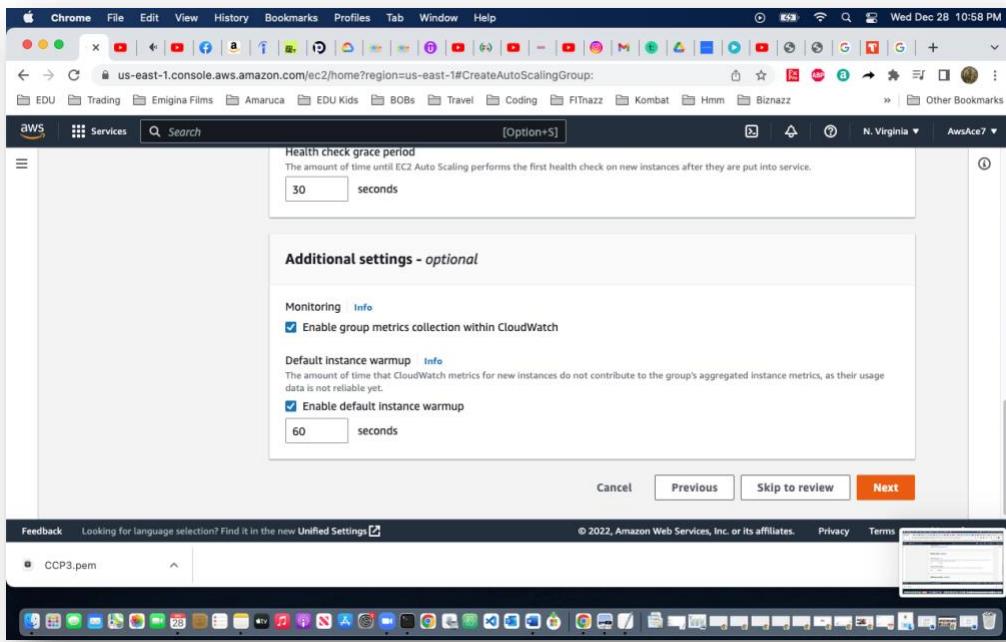


In the **Health Checks** section, where the ASG checks on the status of the EC2's, we'll set the grace period of checkin to every minute or 30 seconds. In production it is normal for the ASG to check the health of the EC2's every 5 minutes or 300 seconds.



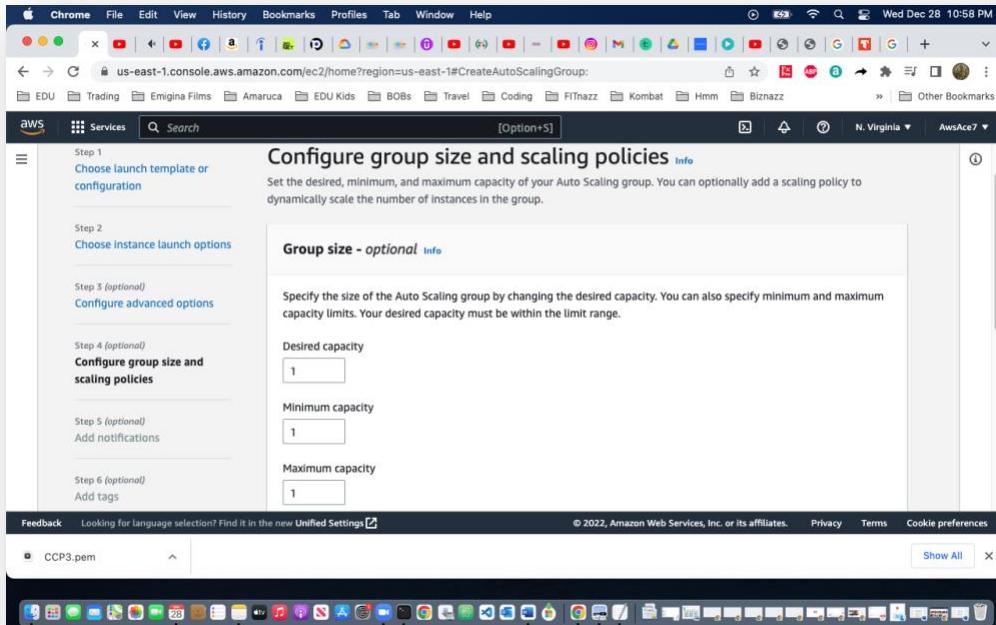
The screenshot shows the AWS CloudFormation Create Stack Wizard, Step 3: Set up resources. The page is titled "WuTG7 | HTTP Application Load Balancer: WuLB1". The "Health checks - optional" section is visible, showing the "Health check type" dropdown set to "EC2" (with "ELB" as an option) and the "Health check grace period" input field set to "30" seconds. Below this, the "Additional settings - optional" section includes a "Monitoring" link. The browser's address bar shows "us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateAutoScalingGroup". The status bar at the bottom indicates "Wed Dec 28 10:58 PM".

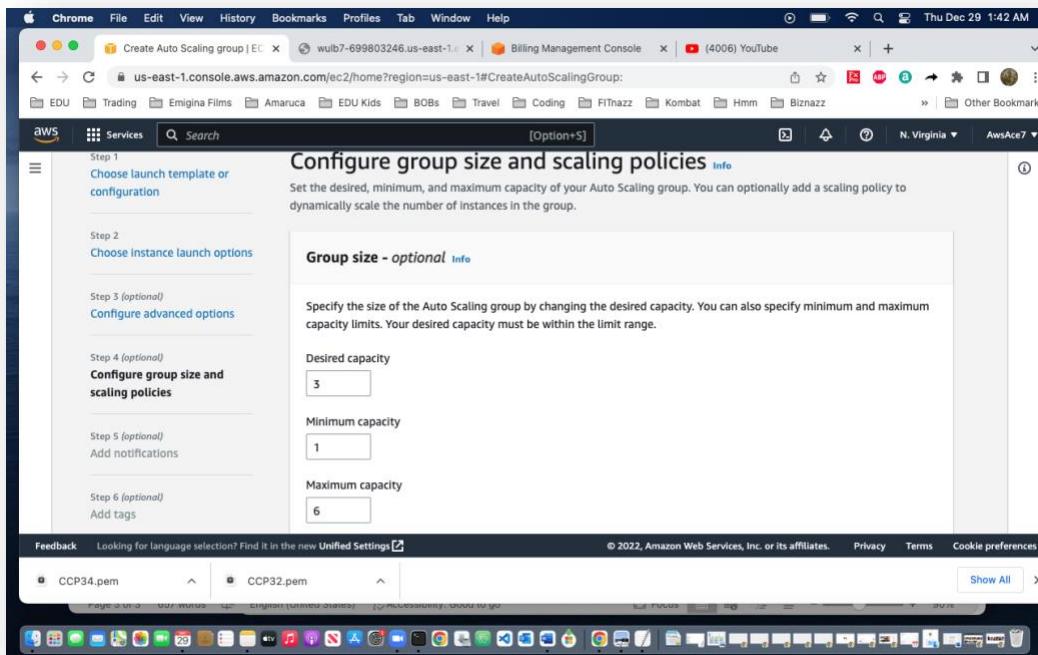
For **Additional Settings**, we will enable the collection of metrics (these are the metrics that could be used to do dynamic scaling mentioned earlier and/or observe the performance of the network) as well as 60 second warmup for EC2's being built for launch.



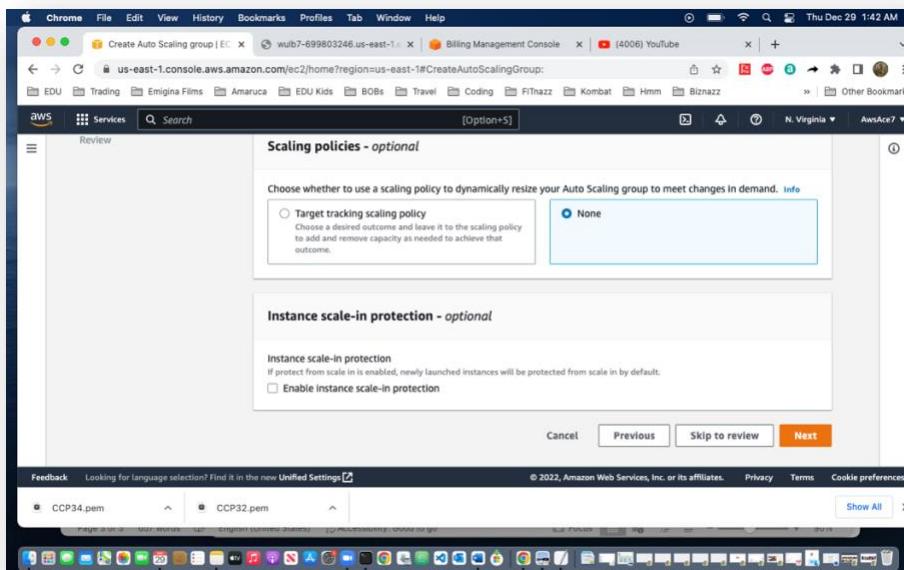
On the “**Configure group size and scaling policies**” page, here we place the amount of EC2’s. In the **Group Size** section, we designate the various capacities.

For **Desired capacity** we’ll have 3 EC2’s (which is one EC2 for EACH of the 3 availability zones), 1 **minimum** EC2 overall in the network and the **maximum** amount being 6 EC2’s (which is 2 EC’s for EACH availability zone).

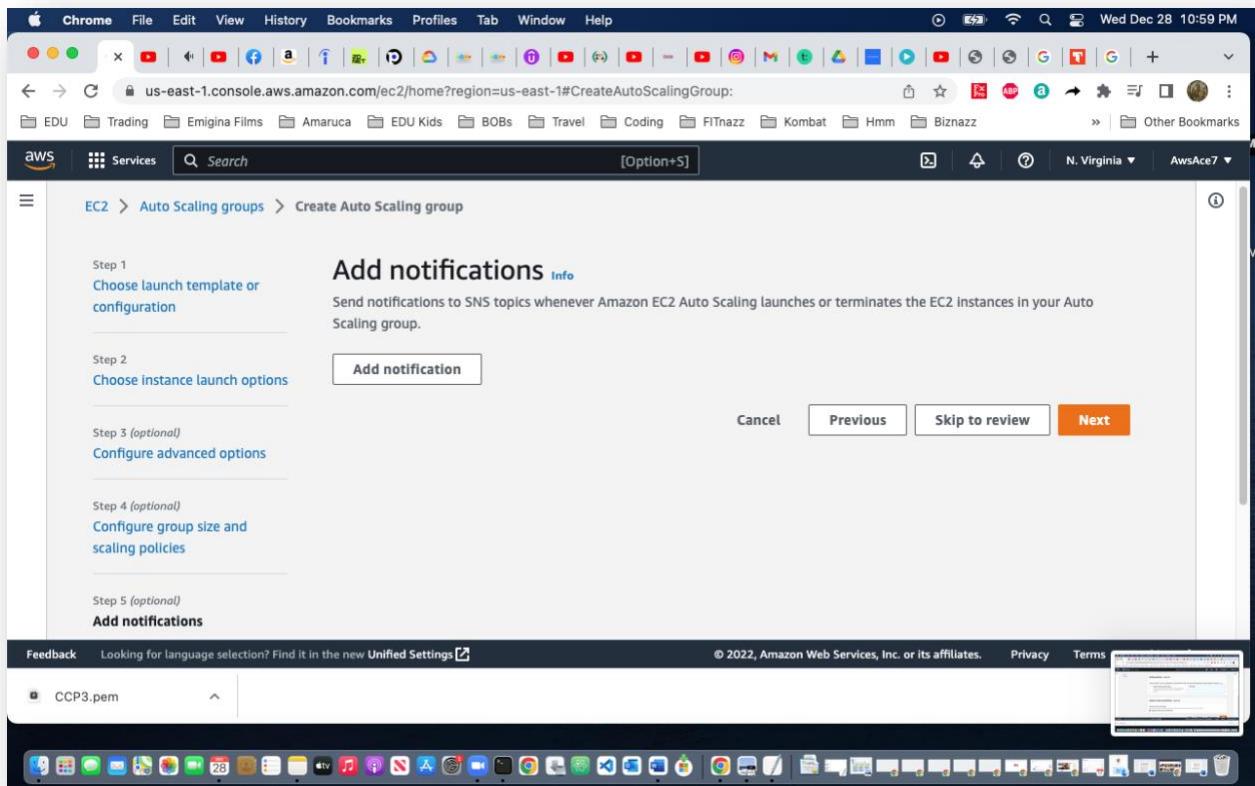




The **Scaling Policies** section is where we can set the rules for the ASG to adhere to, based on metrics criteria. For now, we'll leave the skip setting any rules. For **Instance Scale-in protection**, this action prevents automatic reduction of EC2's as preventative measure, so EC2's will be available regardless of need. This is optional if the user wants it enabled or not.



For now, we can skip notifications. This section alerts you that specific performance metrics have been reached, allowing you to make any necessary changes to your ASG. Then we place some *tags*, for our ASG, followed by a review of configuration.



Chrome File Edit View History Bookmarks Profiles Tab Window Help

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateAutoScalingGroup: Wed Dec 28 11:00 PM

EDU Trading Emigina Films Amarica EDU Kids BOBs Travel Coding FITnazz Kombat Hmm Biznazz Other Bookmarks N. Virginia AwsAce7

Services Search [Option+S]

Step 3 (optional) Configure advanced options

Step 4 (optional) Configure group size and scaling policies

Step 5 (optional) Add notifications

Step 6 (optional) Add tags

Step 7 Review

Tags (3)

Key	Value - optional	Tag new instances
Service	ASG	<input checked="" type="checkbox"/>
Engineer	Ace Cloud	<input checked="" type="checkbox"/>
Location	Sao Paolo	<input checked="" type="checkbox"/>

Add tag 47 remaining

Cancel Previous Next

Feedback Looking for language selection? Find it in the new Unified Settings [\[?\]](#)

© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

CCP3.pem Show All

This screenshot shows the 'Configure advanced options' step of the AWS Auto Scaling Group creation wizard. It displays three tags: Service (ASG), Engineer (Ace Cloud), and Location (Sao Paolo). Each tag has a checkbox labeled 'Tag new instances' which is checked. Below the tags is a button 'Add tag' and a note indicating 47 remaining tags.

Chrome File Edit View History Bookmarks Profiles Tab Window Help

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateAutoScalingGroup: Wed Dec 28 11:00 PM

EDU Trading Emigina Films Amarica EDU Kids BOBs Travel Coding FITnazz Kombat Hmm Biznazz Other Bookmarks N. Virginia AwsAce7

Services Search [Option+S]

Step 1 Choose launch template or configuration

Step 2 Choose instance launch options

Step 3 (optional) Configure advanced options

Step 4 (optional) Configure group size and scaling policies

Step 5 (optional) Add notifications

Step 6 (optional) Add tags

Review [\[?\]](#)

Step 1: Choose launch template or configuration

Group details

Auto Scaling group name: WuASG

Launch template

Launch template	Version	Description
WuEC2Temp [?]	Default	WuCloud EC2 v2 lt-04a15bc77c55f6a68

Step 2: Choose instance launch options

Feedback Looking for language selection? Find it in the new Unified Settings [\[?\]](#)

© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

CCP3.pem Show All

This screenshot shows the 'Review' step of the AWS Auto Scaling Group creation wizard. It displays the chosen launch template (WuEC2Temp) and its version (Default). The launch template description is WuCloud EC2 v2. The wizard also lists the steps completed so far: Step 1 (Choose launch template or configuration) and Step 2 (Choose instance launch options).

Step 6 (optional)
Add tags

Step 7
Review

Step 2: Choose instance launch options

Network

Network

VPC
[vpc-09f402da4b5777b98](#)

Availability Zone	Subnet	
us-east-1a	subnet-031447bb4abdde84f	10.136.12.0/24
us-east-1b	subnet-0135f89bfaae1efcd	10.136.14.0/24
us-east-1c	subnet-00ef1594a66e188ca	10.136.16.0/24

Feedback Looking for language selection? Find it in the new Unified Settings [?](#)

© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Show All [X](#)

Step 6 (optional)
Add tags

Step 7
Review

Step 3: Configure advanced options

Load balancing

Load balancer 1

Name WuLB1	Type Application/HTTP	Target group WuTG7
-------------------------------	--------------------------	---------------------------------------

Feedback Looking for language selection? Find it in the new Unified Settings [?](#)

© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms

Show All [X](#)

Chrome File Edit View History Bookmarks Profiles Tab Window Help

Wed Dec 28 11:00 PM

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateAutoScalingGroup:

EDU Trading Emigina Films Amarуча EDU Kids BOBs Travel Coding FITnazz Kombat Hmm Biznazz Other Bookmarks N. Virginia AwsAce7

AWS Services Search [Option+S]

Name: WuLB1 Type: Application/HTTP Target group: WuTG7

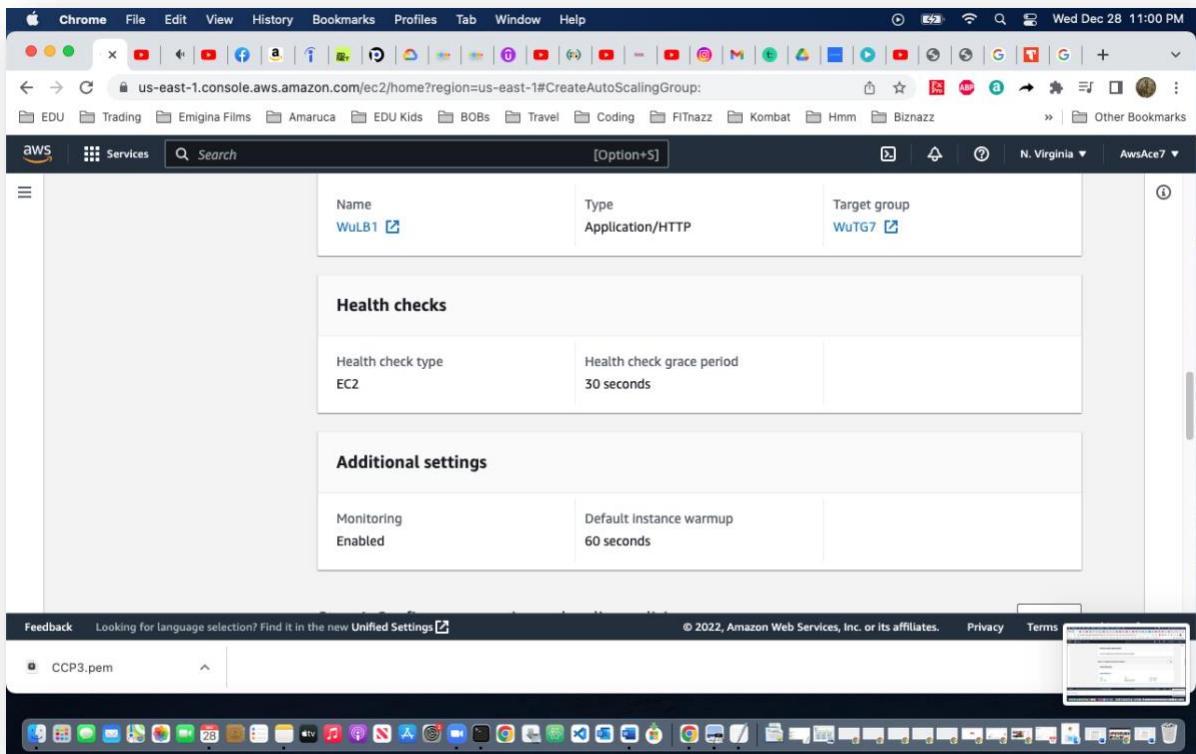
Health checks

Health check type: EC2 Health check grace period: 30 seconds

Additional settings

Monitoring: Enabled Default instance warmup: 60 seconds

Feedback Looking for language selection? Find it in the new Unified Settings © 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms CCP3.pem



Chrome File Edit View History Bookmarks Profiles Tab Window Help

Wed Dec 28 11:01 PM

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateAutoScalingGroup:

EDU Trading Emigina Films Amarucha EDU Kids BOBs Travel Coding FITnazz Kombat Hmm Biznazz Other Bookmarks N. Virginia AwsAce7

AWS Services Search [Option+S]

Instance scale-in protection

Instance scale-in protection: Enable instance protection from scale in

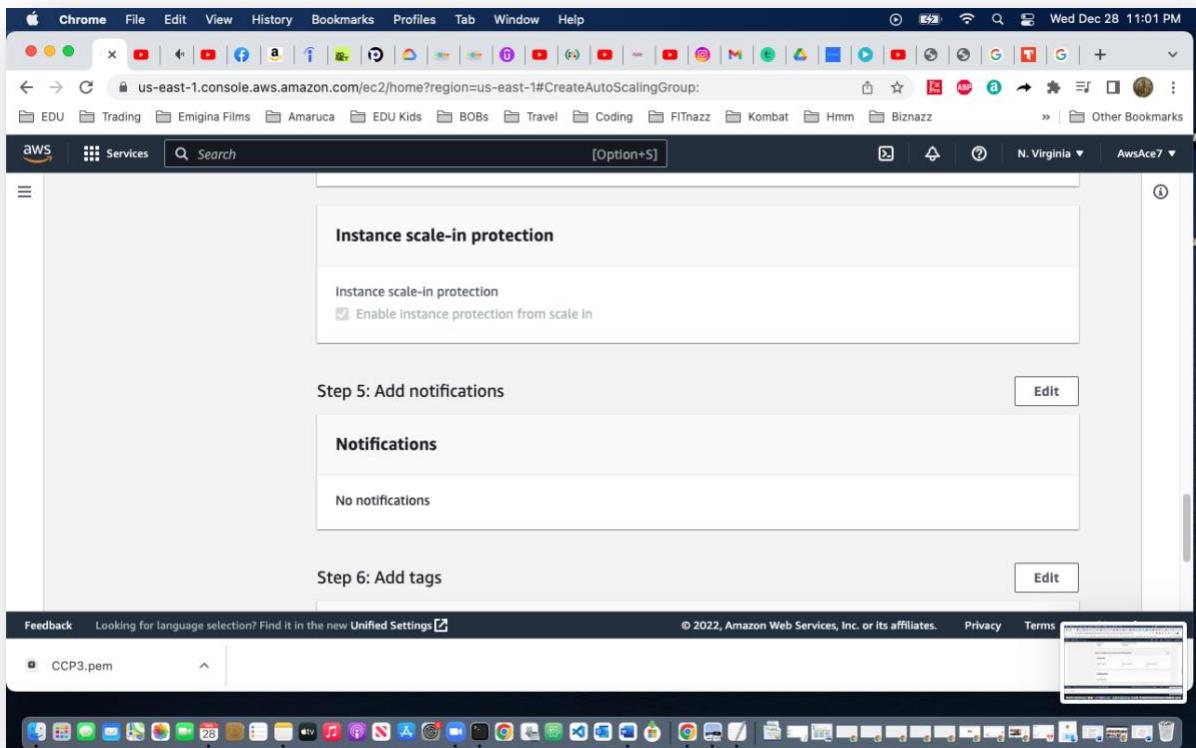
Step 5: Add notifications

Notifications

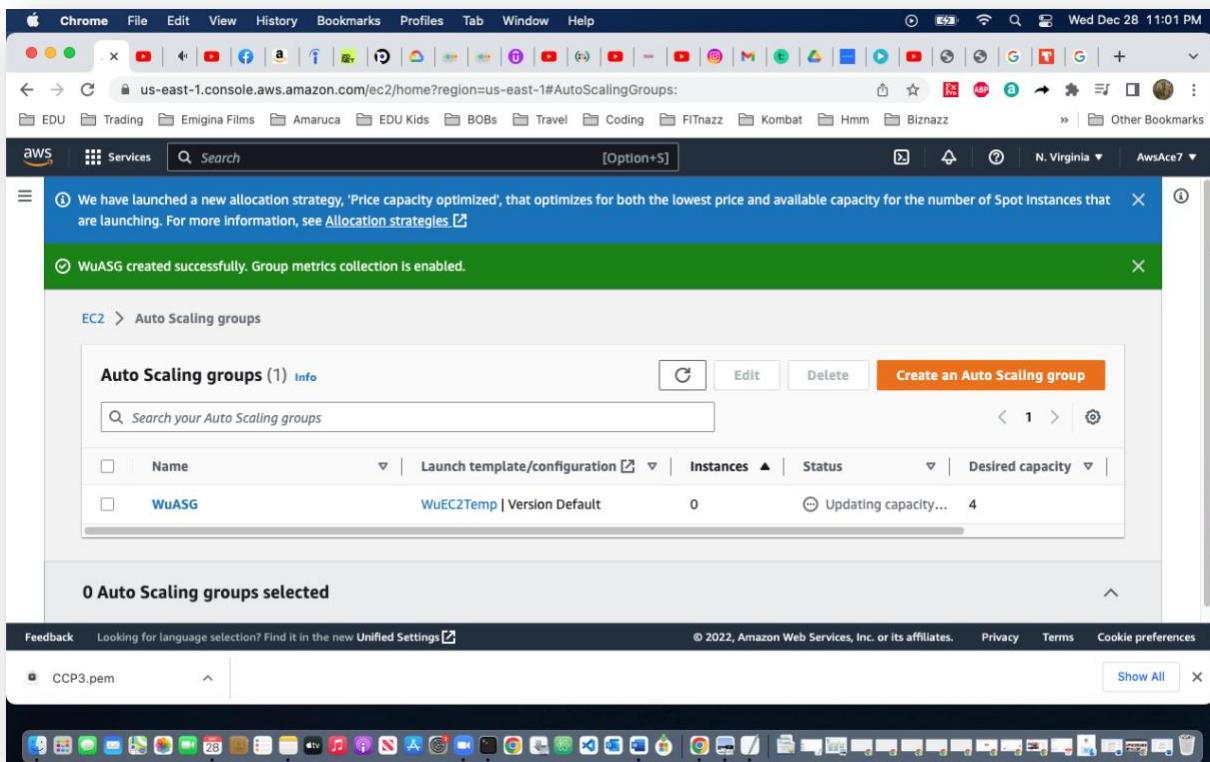
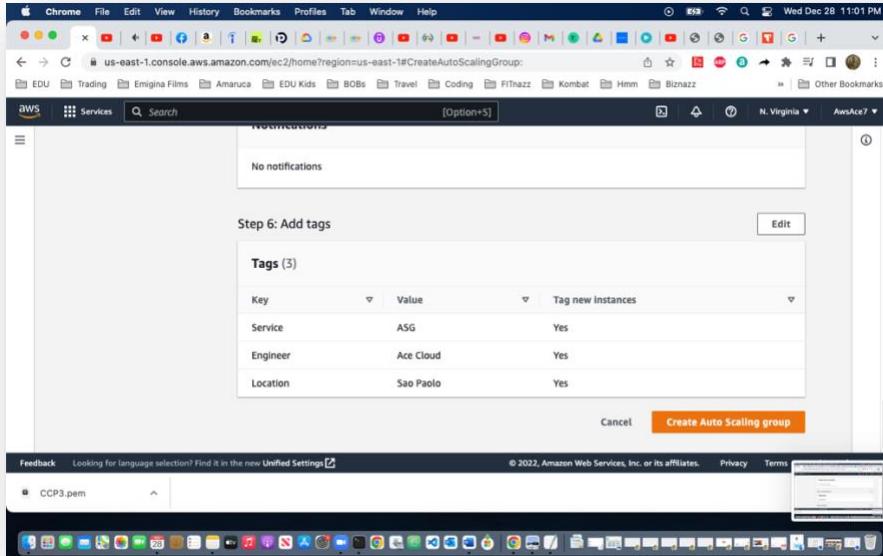
No notifications

Step 6: Add tags

Feedback Looking for language selection? Find it in the new Unified Settings © 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms CCP3.pem



If everything is in order, we click the “create Auto Scaling Group” button to finish the process. There should be a green “*Success*” banner along with a blue “*allocation strategy has begun*” banner going across the screen, signifying the ASG launching EC2’s.



Now at this point, I took the liberty in creating a secondary Auto Scaling Group. In the process I conducted some experimentation to get a desired result. Along the way, I slightly changed the template name. Nevertheless, the process is still the same.

The second ASG was created to house the back end secure EC2's. These are the **10.136.73.0/24 private subnet for Availability zone A; 10.136.75.0/24 private subnet for A-ZB and 10.136.77.0/24 for private subnet AZ-C**.

This was necessary to do, for the next section which will be testing our application.

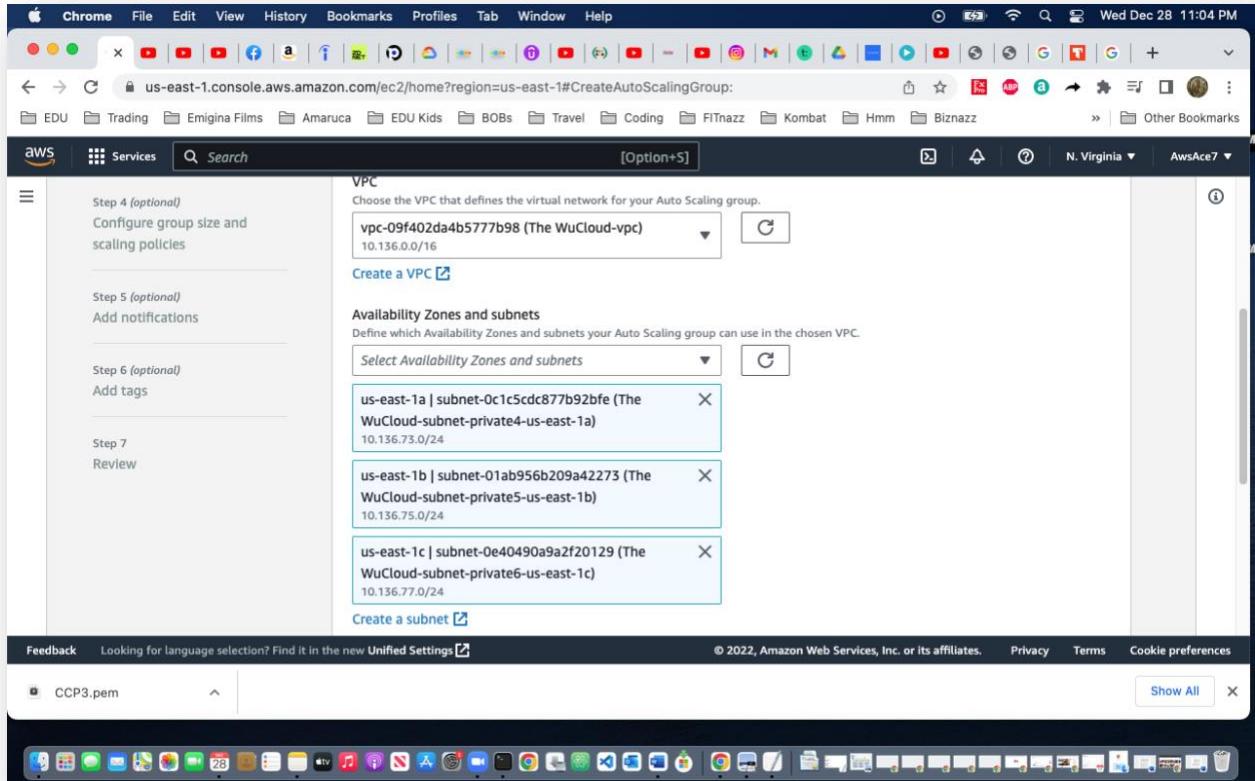
Enjoy the following screenshots of both ASG's and EC2's being prepared to launch.

The screenshot shows a Chrome browser window with several tabs open. The active tab is 'Auto Scaling groups | EC2 Man' on the 'us-east-1.console.aws.amazon.com' domain. The page displays the 'Auto Scaling groups' section with two entries:

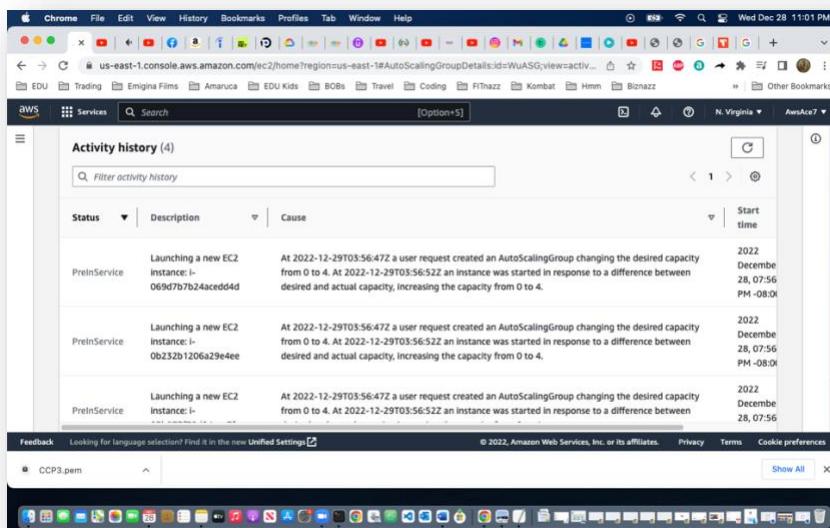
Name	Launch template/configuration	Instances	Status	Desired capacity
WuASG	WuEC3Temp Version Default	3	-	3
WuASG2	WuEC3Temp Version Default	3	-	3

Below the table, a message states '0 Auto Scaling groups selected'. The browser's status bar at the bottom shows file paths like 'CCP34.pem' and 'CCP32.pem'.

When making the 2nd ASG, for the **Network** section, you must again change the VPC from the default to your VPC and make sure you add the correct availability zones for the back end private subnet EC2's.



Here are EC2's being prepared to launch and some initial successes.



The screenshot shows the AWS Management Console in a web browser. The URL is us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#AutoScalingGroupDetails:id=WuASG;view=activity. The left sidebar shows navigation options like Lifecycle Manager, Network & Security, Load Balancing, and Auto Scaling. The main content area is titled "Activity history (4)" and lists three successful events related to launching new EC2 instances. The browser's address bar shows "Auto Scaling group details | EC2". The status bar at the bottom indicates "Thu Dec 29 1:58 AM".

Here are the “**Details**” Page of the 2nd Auto Scaling Group, same as the first.

The screenshot shows the AWS Management Console in a web browser. The URL is us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#AutoScalingGroupDetails:id=WuASG2;view=details. The left sidebar shows navigation options like Lifecycle Manager, Network & Security, Load Balancing, and Auto Scaling. The main content area is titled "WuASG2" and displays "Group details" for the second Auto Scaling group. It shows the desired capacity as 3, minimum capacity as 1, and maximum capacity as 6. The auto scaling group name is WuASG2, and it was created on Wed Dec 28 2022 22:39:47 GMT-0800 (Pacific Standard Time). The ARN is arn:aws:autoscaling:us-east-1:966320976686:autoScalingGroup:49c32. The browser's address bar shows "Auto Scaling group details | EC2". The status bar at the bottom indicates "Thu Dec 29 1:58 AM".

Now go to your EC2 Dashboard. If you click the “refresh” button right next to the “EC2 Global View”, button, it will show the EC2 instances being launched. You will see 7 instances up and running. Select “instances running”, to see your EC2’s.

The screenshot shows the AWS EC2 Dashboard in the New EC2 Experience. The left sidebar has sections for EC2 Global View, Events, Tags, Limits, Instances (with sub-options like Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, and Dedicated Hosts), and Feedback. The main Resources section displays the following counts:

Category	Value
Instances (running)	0
Dedicated Hosts	0
Elastic IPs	3
Instances	0
Key pairs	9
Load balancers	0
Placement groups	0
Security groups	4
Snapshots	0
Volumes	0

An info box at the bottom left says: “Easily size, configure, and deploy Microsoft SQL Server Always On availability groups on AWS using the AWS Launch Wizard for SQL Server.” A “Learn more” link is provided. The Account attributes sidebar on the right lists VPC, Default VPC (vpc-0df1cb3cae784a2d8), Settings, EBS encryption, Zones, EC2 Serial Console, Default credit specification, and Console experiments. The status bar at the bottom indicates “© 2022, Amazon Web Services, Inc. or its affiliates.”

This screenshot shows the same EC2 Dashboard after launching 7 instances. The Resources section now shows:

Category	Value
Instances (running)	7
Dedicated Hosts	0
Elastic IPs	3
Instances	21
Key pairs	12
Load balancers	1
Placement groups	0
Security groups	6
Snapshots	0
Volumes	7

The “Instances (running)” link is highlighted in blue. The “Learn more” link in the info box is no longer present. The status bar at the bottom indicates “© 2022, Amazon Web Services, Inc. or its affiliates.”

As you can see, there are 7 EC2's running. There is the original EC2 which is marked in the screenshot below as WuEC212-45a. This is instance we created our template from. The numbers "212-45a" are the numbers of its IP address, I used to rename the EC2, to help differentiate it from the others. You can see the renamed EC2's according to there own IP addresses.

The other 6 instances were the instances created by the Auto Scaling Group from the EC2 template we made earlier.

The screenshot shows the AWS EC2 Instances page in a web browser. The URL is `us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances:instanceState=running;sort=tag:Name`. The page displays 7 instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Avg.
WuEC212-4	I-01f0c55f4cb267e18	Running	t2.micro	2/2 checks passed	No alarms	+ us-e
WuEC212-45a	I-0530a0f1cf3877eb4	Running	t2.micro	2/2 checks passed	No alarms	+ us-e
WuEC214-223	I-006320a644551eb1e	Running	t2.micro	2/2 checks passed	No alarms	+ us-e
WuEC216-201	I-0d076121f1f1a216a	Running	t2.micro	2/2 checks passed	No alarms	+ us-e
WuEC273-146	I-0acfdd843bcd028d3	Running	t2.micro	2/2 checks passed	No alarms	+ us-e
WuEC275-23	I-094fcdc7648c302df	Running	t2.micro	2/2 checks passed	No alarms	+ us-e
WuEC277-142	I-07a34b427b7d560c3	Running	t2.micro	2/2 checks passed	No alarms	+ us-e

The instance **WuEC212-45a** is selected, highlighted with a blue border. The page also includes a sidebar with navigation links like EC2 Dashboard, EC2 Global View, Events, Tags, Limits, and Instances (with sub-links for Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, and Dedicated Hosts). The bottom of the screen shows the Mac OS X dock with various application icons.

Here you can see the Availability Zones, with 2 EC2's in each zone, totaling 6 instances, excluding the 1st EC2 in zone A.

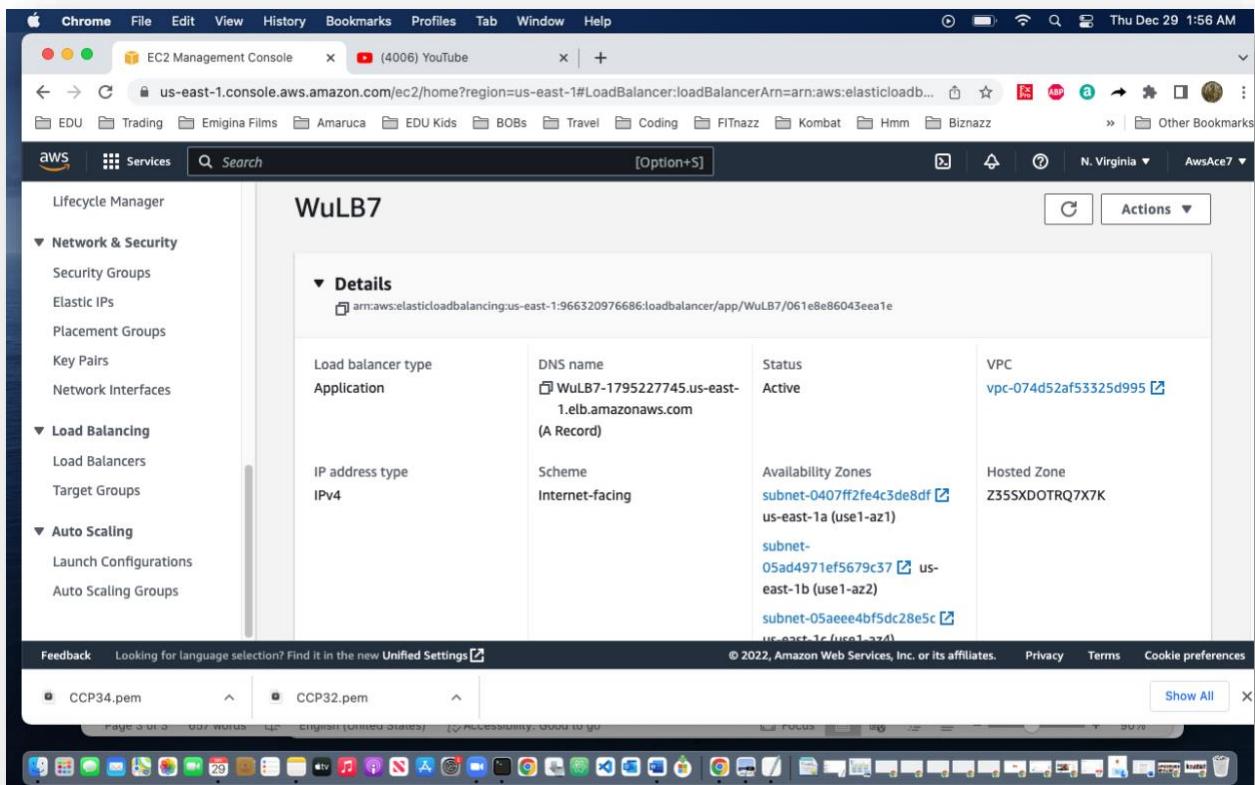
The screenshot shows the AWS EC2 Management Console interface. The left sidebar is collapsed, and the main area displays a table of running instances. The table has columns for Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IPv4 DNS, and Public IP. There are 7 instances listed, grouped into three availability zones: us-east-1a, us-east-1b, and us-east-1c. Each instance is a t2.micro type with a status check of 2/2 checks passed and no alarms. The Public IPv4 DNS and Public IP columns show various IP addresses starting with ec2- and 44.204, 44.199, 54.82.1, 54.156, 52.207, 3.94.12, and 34.228 respectively.

Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IP
Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-44-204-121-98.co...	44.204
Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-44-199-212-136.co...	44.199
Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-54-82-142-2.com...	54.82.1
Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-54-156-151-85.co...	54.156
Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-52-207-236-251.co...	52.207
Running	t2.micro	2/2 checks passed	No alarms	us-east-1c	ec2-3-94-127-149.com...	3.94.12
Running	t2.micro	2/2 checks passed	No alarms	us-east-1c	ec2-34-228-55-26.com...	34.228

This screenshot shows the same EC2 Management Console interface as the previous one, but with one instance selected. The instance 'i-0530a0f1cf3877eb4' is highlighted in the table. The 'Actions' dropdown menu is open, showing options like 'Stop', 'Terminate', and 'Launch Instances'. The rest of the interface and the list of instances remain the same as in the first screenshot.

Testing our Application by Elastic Load Balancer DNS

Go to the EC2 left side menu and scroll to select the “*Load Balancer*” option. Select the load balancer hyperlink to see its details page as below. On this page, go move your cursor towards the **DNS name** detail. Hover the cursor over the 2 boxes overlapping, right under **DNS name**. Click the boxes to copy the domain name of the load balancer which is DNS or Domain Name System.



Thu Dec 29 1:56 AM

EC2 Management Console (4006) YouTube

EDU Trading Emigina Films Amaruka EDU Kids BOBs Travel Coding Fitnazz Kombat Hmm Biznazz Other Bookmarks N. Virginia AwsAce7

Services Search [Option+S]

Lifecycle Manager

Network & Security

- Security Groups
- Elastic IPs
- Placement Groups
- Key Pairs
- Network Interfaces

Load Balancing

- Load Balancers
- Target Groups

Auto Scaling

- Launch Configurations
- Auto Scaling Groups

Load balancer type: Application

DNS name: WuLB7-1795227745.us-east-1.elb.amazonaws.com (A Record)

Status: Active

VPC: vpc-074d52af53325d995

IP address type: IPv4

Scheme: Internet-facing

Availability Zones:

- subnet-0407ff2fe4c3de8df us-east-1a (use1-az1)
- subnet-05ad4971ef5679c37 us-east-1b (use1-az2)
- subnet-05aeee4bf5dc28e5c us-east-1c (use1-az4)

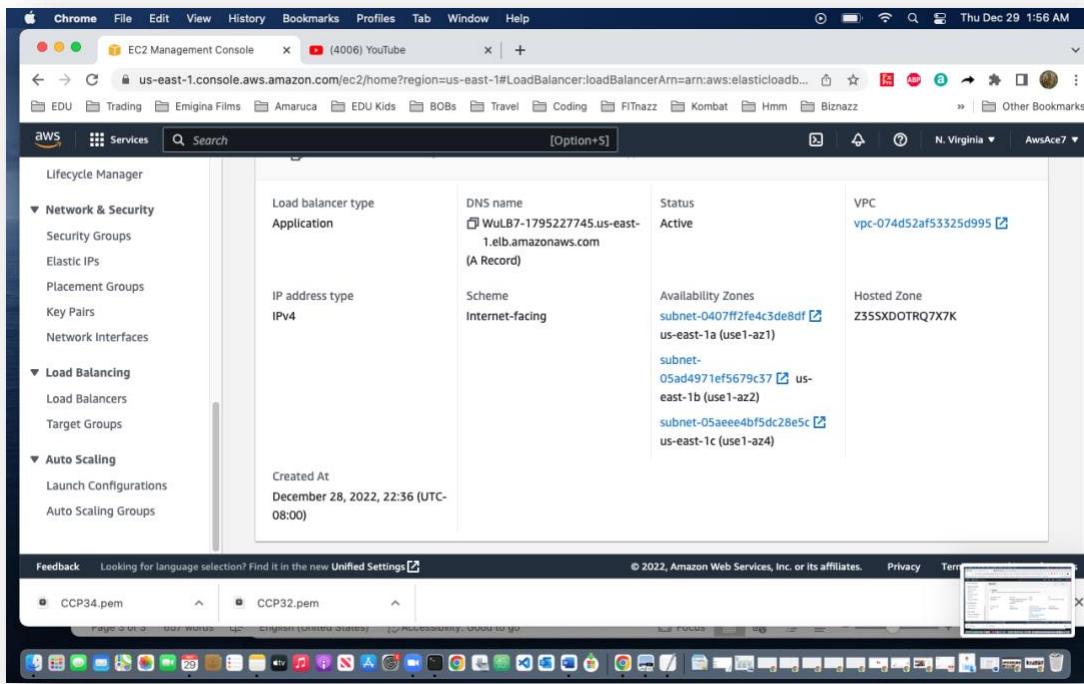
Hosted Zone: Z35SXDOTRQ7X7K

Created At: December 28, 2022, 22:36 (UTC-08:00)

Feedback Looking for language selection? Find it in the new Unified Settings

CCP34.pem CCP32.pem

© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms



Thu Dec 29 1:57 AM

EC2 Management Console (4006) YouTube

EDU Trading Emigina Films Amaruka EDU Kids BOBs Travel Coding Fitnazz Kombat Hmm Biznazz Other Bookmarks N. Virginia AwsAce7

Services Search [Option+S]

Lifecycle Manager

Network & Security

- Security Groups
- Elastic IPs
- Placement Groups
- Key Pairs
- Network Interfaces

Load Balancing

- Load Balancers
- Target Groups

Auto Scaling

- Launch Configurations
- Auto Scaling Groups

Load balancer type: Application

DNS name: Copied 795227745.us-east-1.elb.amazonaws.com (A Record)

Status: Active

VPC: vpc-074d52af53325d995

IP address type: IPv4

Scheme: Internet-facing

Availability Zones:

- subnet-0407ff2fe4c3de8df us-east-1a (use1-az1)
- subnet-05ad4971ef5679c37 us-east-1b (use1-az2)
- subnet-05aeee4bf5dc28e5c us-east-1c (use1-az4)

Hosted Zone: Z35SXDOTRQ7X7K

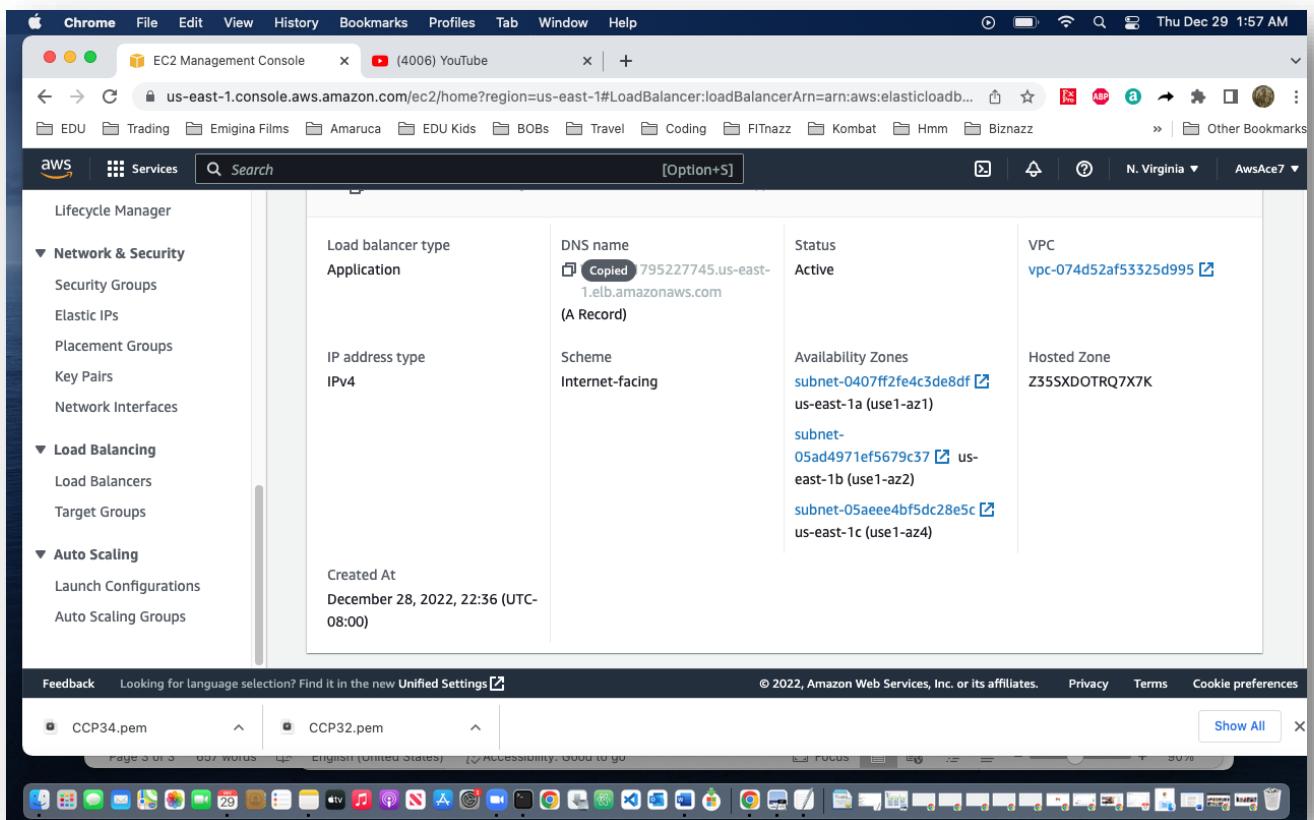
Created At: December 28, 2022, 22:36 (UTC-08:00)

Feedback Looking for language selection? Find it in the new Unified Settings

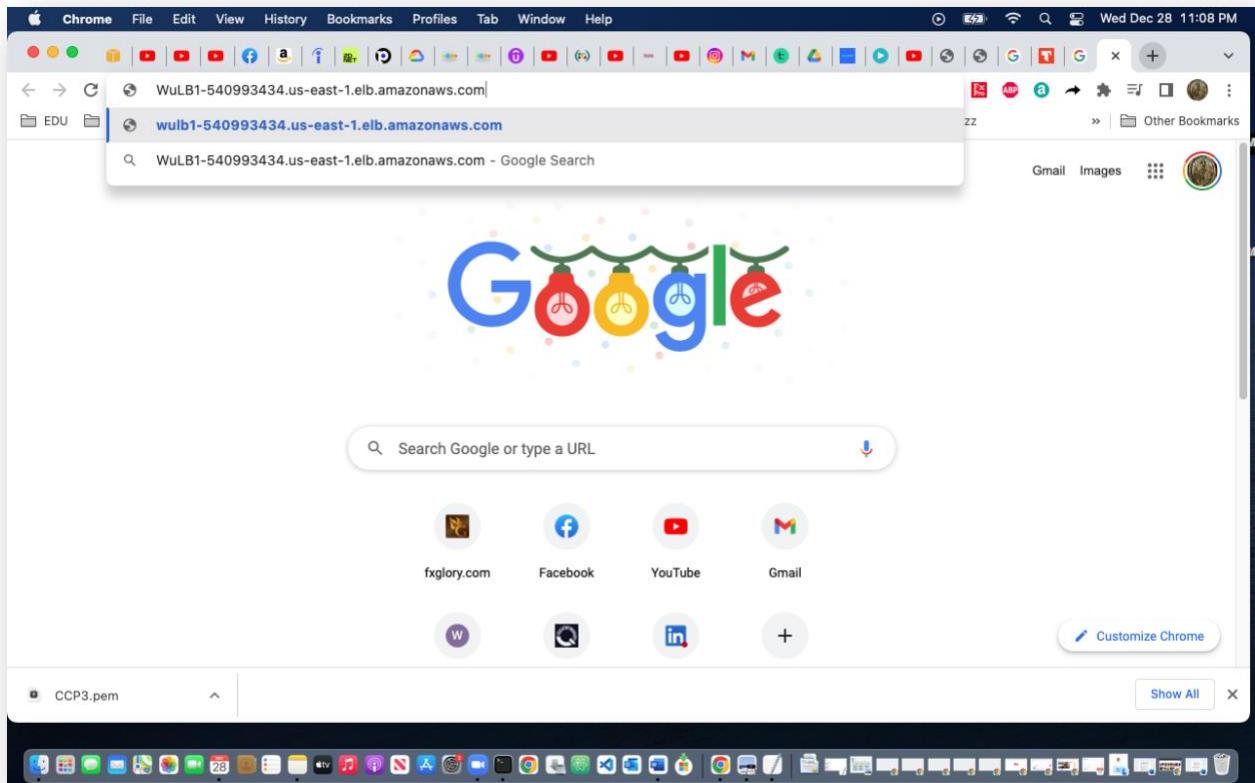
CCP34.pem CCP32.pem

Show All

© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

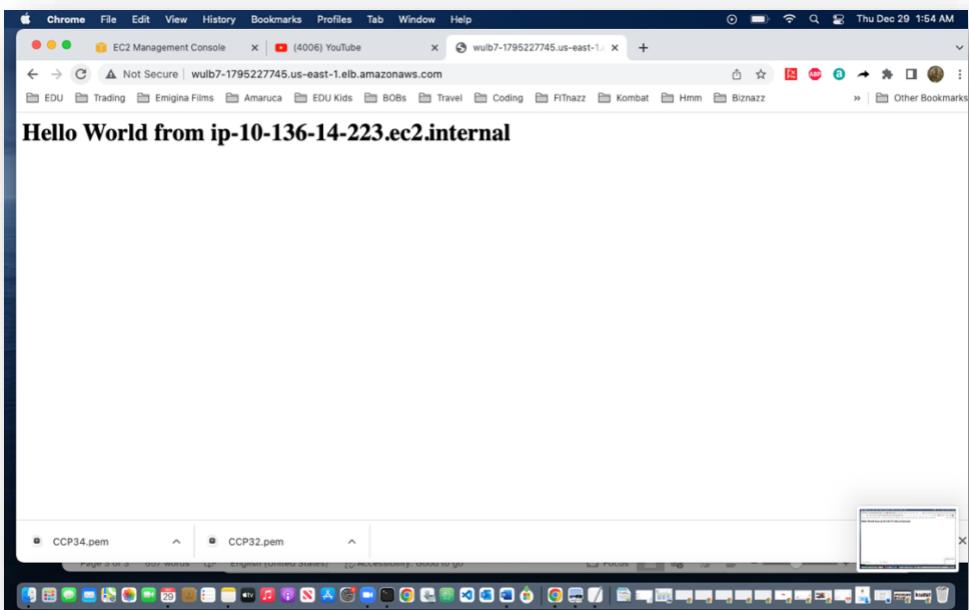
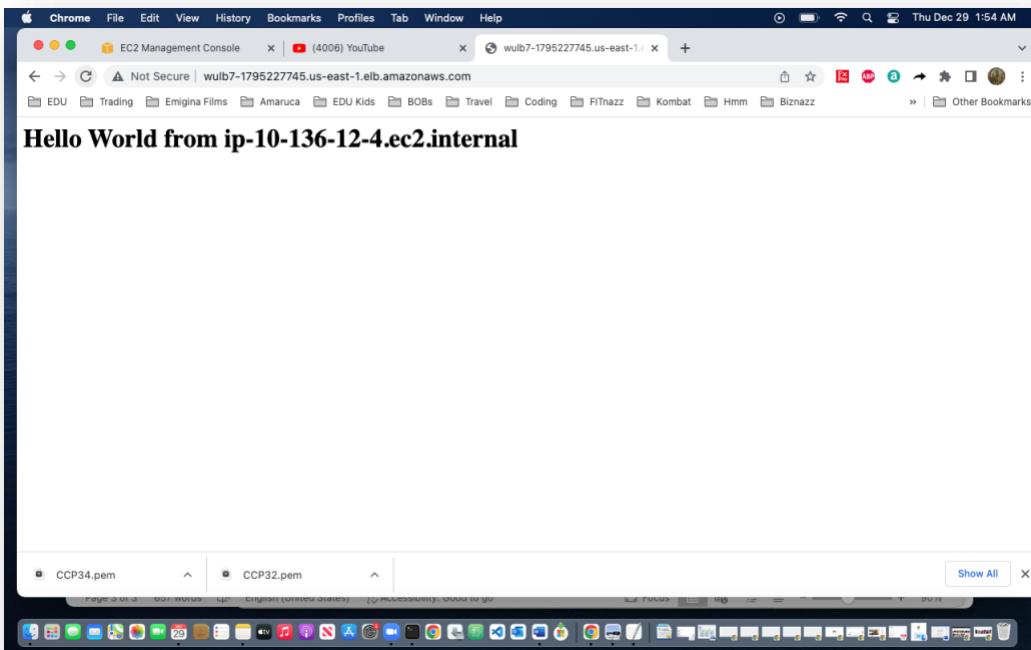


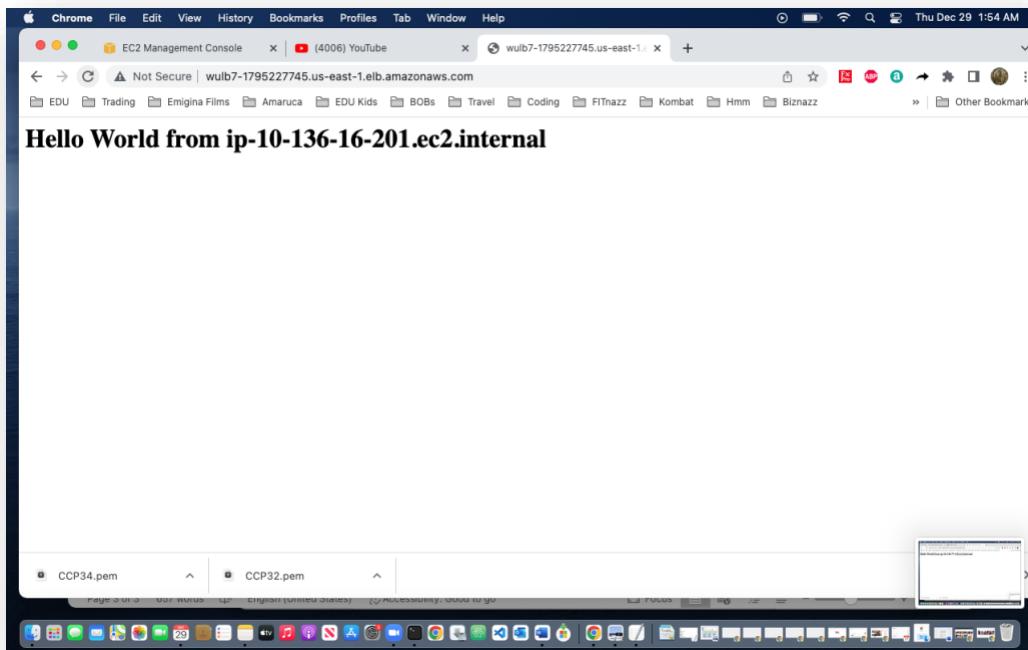
Open a new web browser tab to paste the domain name of the load balancer.



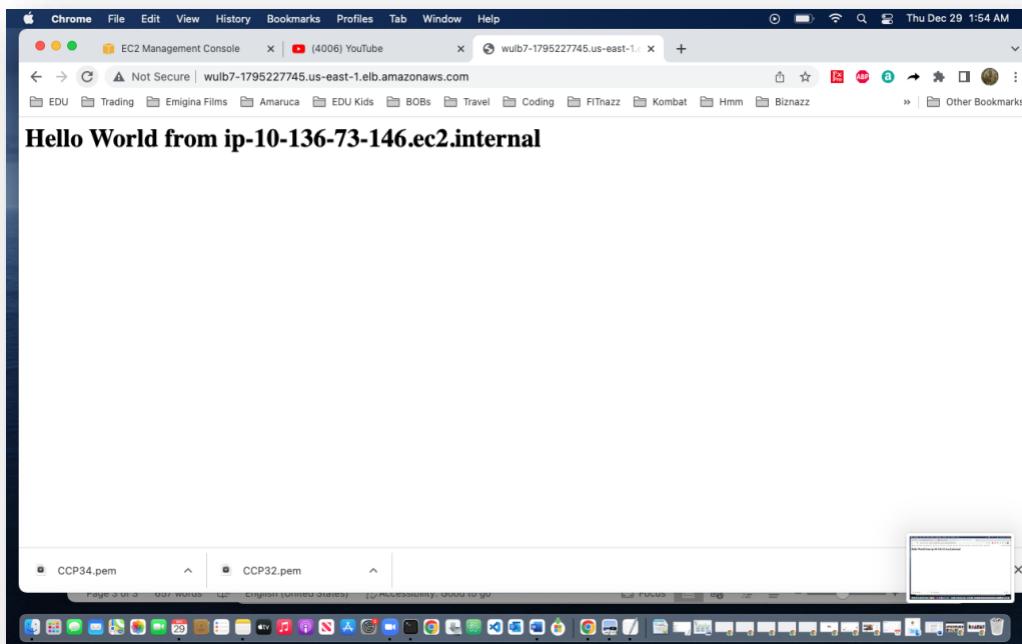
Press the refresh button of the browser page. Each and every refresh that is done, the load balancer takes the *refresh browser requests* and returns a private subnet that has one of the very 6 EC2's we created with the Auto Scaling Group.

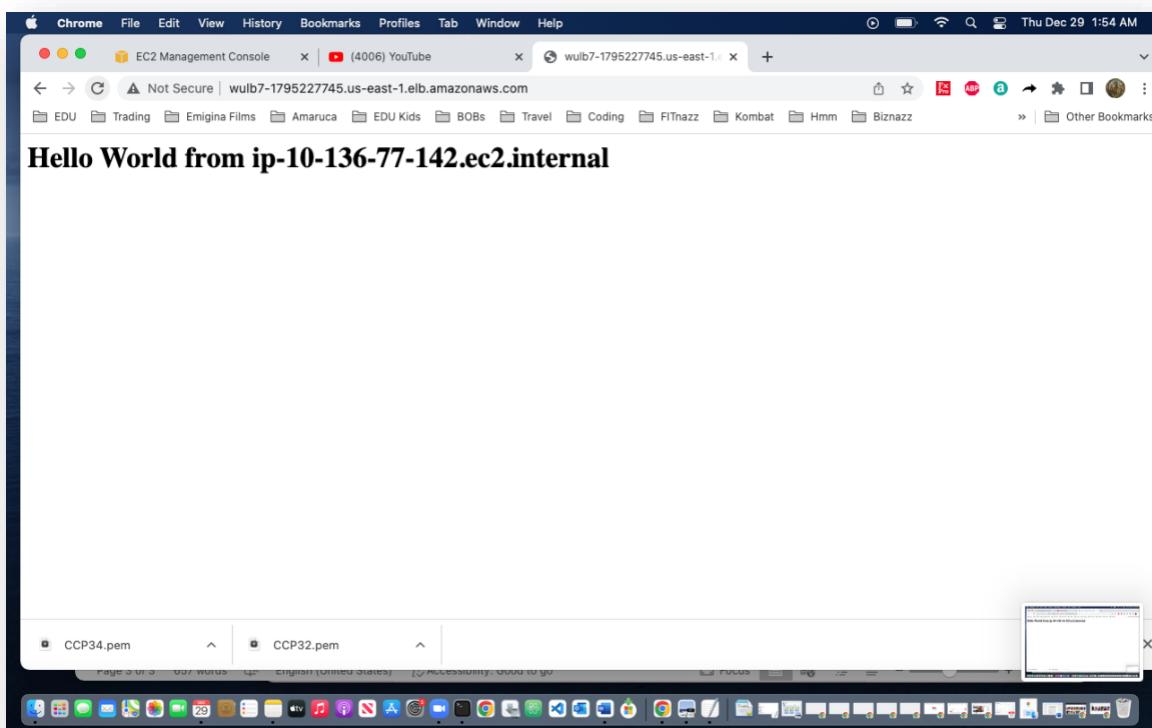
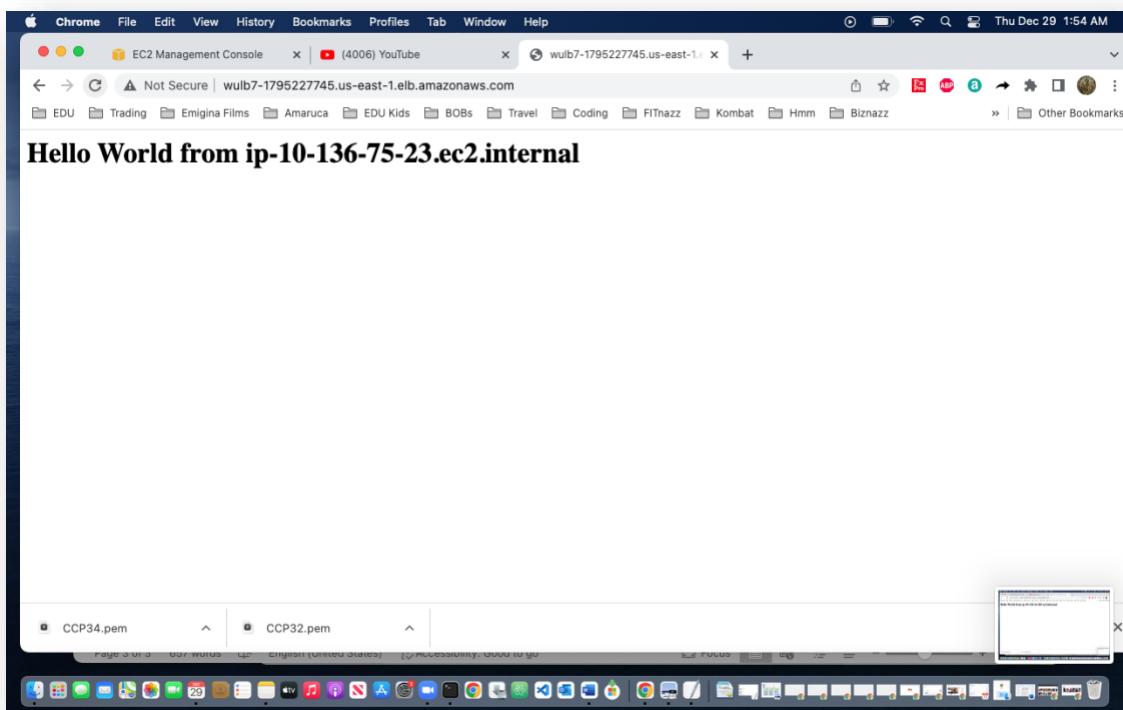
Below are the screenshots of all 6 EC2's across 3 Availability zones. The first EC2 is from the front end private subnet, based on the *double digit tens place subnet 10.136.12.0/24 in Availability zone A*. *The next EC2 will be another front end EC2 with the 10.136.14.0/24 private subnet, within Availability zone B. Last Front End EC2 is 10.136.16.0/24, the private subnet in Availability zone C.*





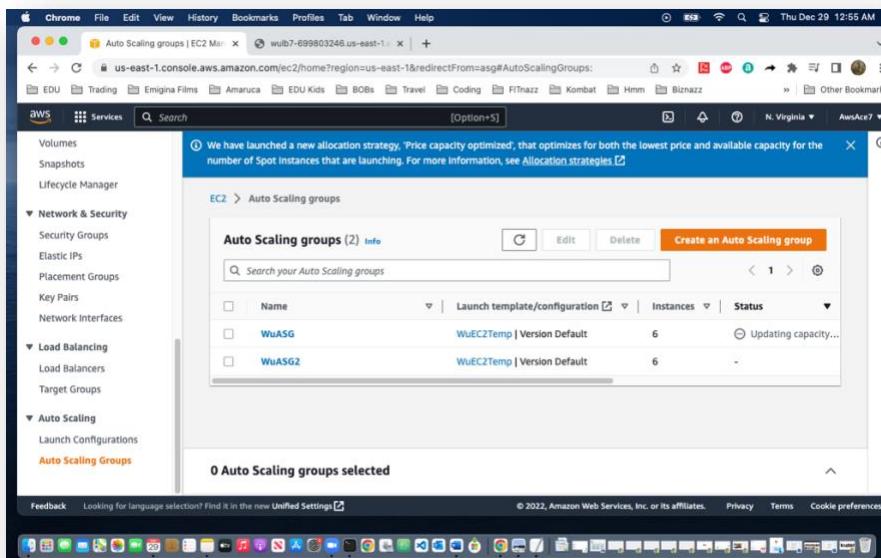
Now we are seeing the ***Secure Back End EC2's starting with 10.136.73.0/24 the private subnet in Availability zone A. Next EC2 is from private subnet 10.136.75.0/24 in Availability zone B. Last EC2 is in private subnet 10.136.77.0/24 in Availability Zone C.***



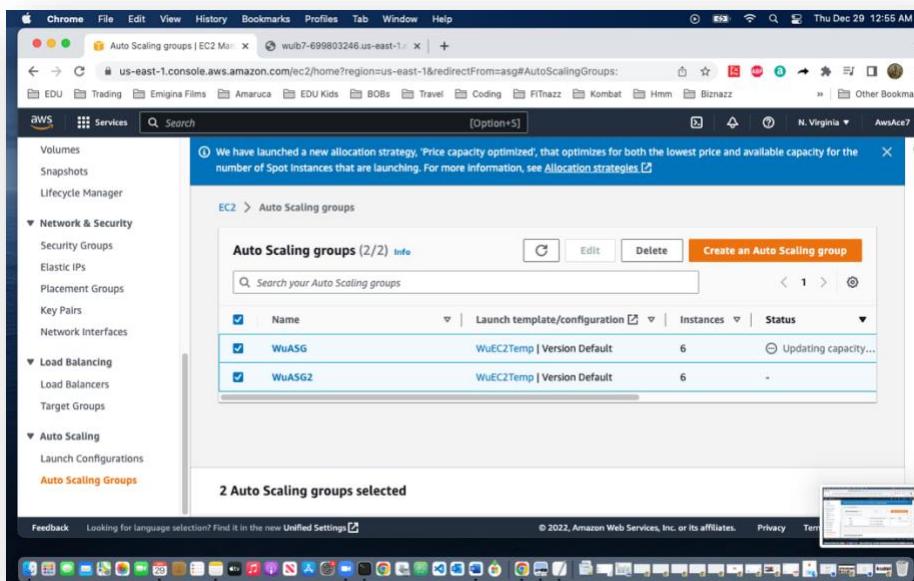


Terminating an Auto Scaling Group

To tear down or delete Auto Scaling Groups, select the “*Auto Scaling Group*” option from the left side menu. Check the boxes next to both Auto Scaling Group’s, WuASG and WuASG2. Then select the “*delete*” button.

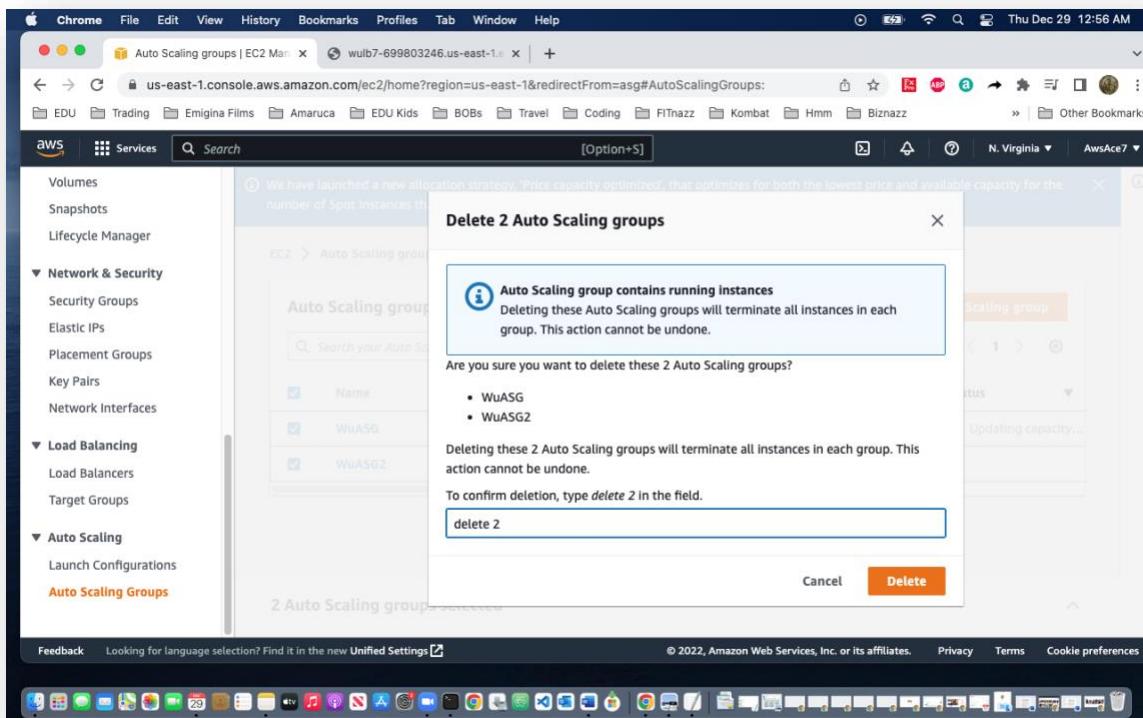
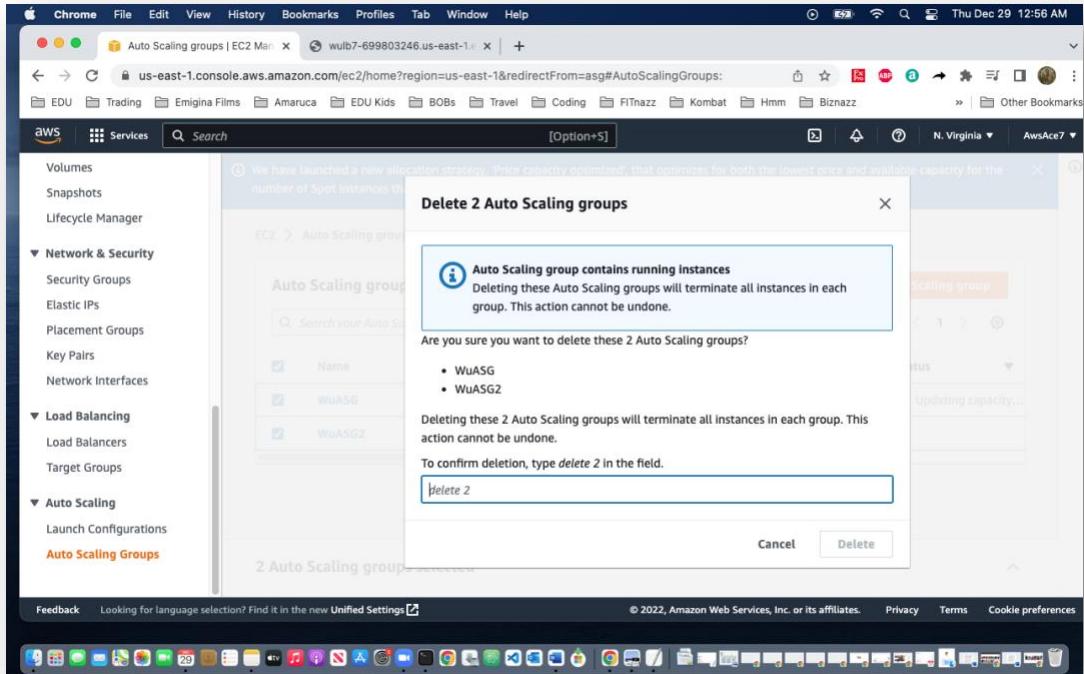


The screenshot shows the AWS EC2 Auto Scaling Groups page. On the left, there's a navigation sidebar with options like Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces, Load Balancing, Auto Scaling, Launch Configurations, and Auto Scaling Groups. The Auto Scaling Groups option is selected. The main content area displays a table titled "Auto Scaling groups (2) Info". The table has columns for Name, Launch template/configuration, Instances, and Status. It lists two entries: "WuASG" and "WuASG2", both of which are currently set to "6 instances" and have a status of "Updating capacity...". Below the table, it says "0 Auto Scaling groups selected". At the top right of the table, there are "Edit" and "Delete" buttons, with "Delete" being highlighted in orange.



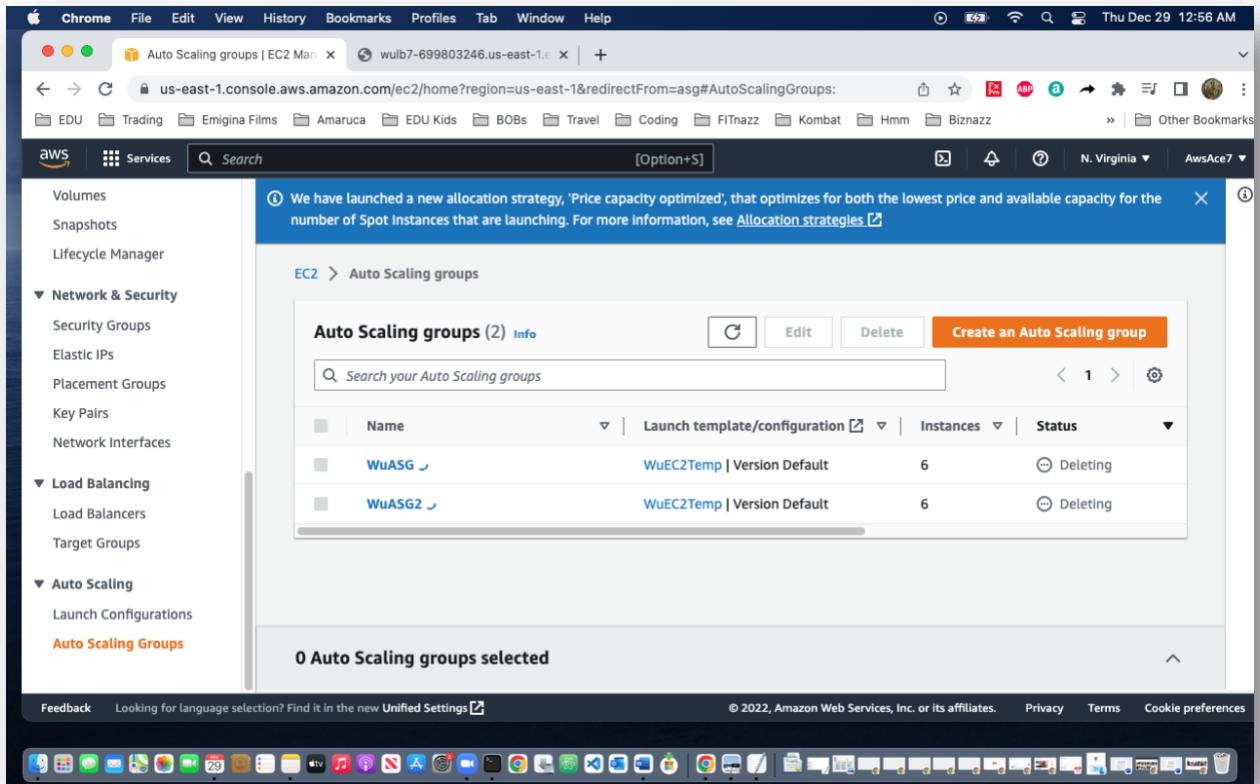
This screenshot is identical to the one above, but with a key difference: both the "WuASG" and "WuASG2" rows in the table now have a checked checkbox in the first column. This indicates that both Auto Scaling Groups are selected for deletion. The rest of the interface, including the sidebar and the overall layout, remains the same.

Type “*delete 2*” in the confirmation input field, to authorize the deletion of the groups. Then select the orange “*delete*” button.



Now your ASG's will be deleted from the network. That includes any EC2's created by the ASG. Once the Auto Scaling Groups are deleted, the tear down process of the other components (load balancer, target group, NAT gateways, original EC2's, VPC network and elastic IP's) can be commenced.

That concludes this documentation on the setup and tear down of Auto Scaling Groups.



The screenshot shows the AWS EC2 Auto Scaling Groups page. The left sidebar navigation includes: Volumes, Snapshots, Lifecycle Manager, Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), Load Balancing (Load Balancers, Target Groups), and Auto Scaling (Launch Configurations, Auto Scaling Groups). The main content area displays the 'Auto Scaling groups' section with a table:

Name	Launch template/configuration	Instances	Status
WuASG	WuEC2Temp Version Default	6	Deleting
WuASG2	WuEC2Temp Version Default	6	Deleting

Below the table, a message states: '0 Auto Scaling groups selected'. The browser header shows the URL as us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1&redirectFrom=asg#AutoScalingGroups.