

Practical 1	2
Launching an Amazon EC2 Instance	2
Connecting to Amazon EC2 Instance	4
Practical 2	6
Create a Website using User Data	6
Website using S3 User Data	8
Using Access Keys with EC2	12
Practical 3	13
Auto Scaling	13
Auto Scaling with Scaling Policy	18
Practical 4	24
Add Load Balancing with Amazon ELB	24
Practical 5	37
Create and Attach EBS Volume	37
EBS Snapshots and AMI	37
EBS Snapshots and AMI's	50
Practical 6	62
Create Amazon EFS File System	62
Practical 7	76
S3 Create Bucket and Make Public	76
Hosting a Static Website using Amazon S3	81
Mock Test	88
 Cloud Computing Mock Practical Test (Learner Lab Style)	88
Task 1: Launch and Configure EC2 Instance	88
Task 2: Deploy Website Using EC2 User Data	88
Task 3: Use S3 with EC2	89
Task 4: Auto Scaling Setup	89
Task 5: Auto Scaling with Scaling Policy	89
Task 6: Load Balancer Integration	90
Task 7: EBS Volume and Snapshot	90
Task 8: Create and Mount EFS	90
Task 9: S3 Bucket Public Access	90
Task 10: Host Static Website on S3	91

Practical 1

Launching an Amazon EC2 Instance

1. Search for **EC2** → Click to enter
2. Sidebar find "**Instances**" → Click on "Instances"
3. Enter instance name
4. Select "**Amazon Linux**" as **Application** and OS Images (Amazon Machine Image)
5. Select "**t2.micro**" as **Instance type**
6. Select "**vockey**" as **Key pair (login)**
7. **Access Network settings** → Click "Edit" button
8. Enter "**Web-Access**" as **Security group name**
9. Enter "**Web-Access**" as **Description**
10. In **Inbound security groups rules** → allow "**ssh**" type (remote access) and "**Anywhere**" source type (Source is from Anywhere)
11. Click on "**Launch instance**"

The screenshot shows the AWS EC2 Management Console interface. On the left, the navigation sidebar includes links for EC2 Dashboard, EC2 Global View, Events, Tags, Limits, Instances (with sub-links for Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations), Images (AMIs, AMI Catalog), and Elastic Block Store (Volumes, Snapshots). The main content area displays the 'Instances' page with one instance listed:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
-	i-0d78b3db634e8d4ad	Running	t2.micro	Initializing	No alarms	us-east-1c	ec2-34-226-245-19

Below this, the 'Instance: i-0d78b3db634e8d4ad' details page is open. It shows the following information:

Details			Security			Networking			Storage			Status checks			Monitoring			Tags		
Instance summary																				
Instance ID	IP address	Private IP address	Public IPv4 address	Private IP DNS name (IPv4 only)	Instance type	Instance state	IPv6 address	Public IPv4 DNS	Elastic IP addresses	VPC ID	Subnet ID	Hostname type	Answer private resource DNS name	Auto-assigned IP address	AWS Compute Optimizer finding	Auto Scaling Group name	Feedback	Cookie preferences		
i-0d78b3db634e8d4ad	34.226.245.197	ip-172-31-87-236.ec2.internal	34.226.245.197	ip-172-31-87-236.ec2.internal	t2.micro	Running	-	ec2-34-226-245-197.compute-1.amazonaws.com	-	vpc-0ac01172526a488c2	-	IPV4 (A)	IPV4 (A)	34.226.245.197 [Public IP]	Opt-in to AWS Compute Optimizer for recommendations.	Learn more	Looking for language selection? Find it in the new Unified Settings	Cookie preferences		

Learner Lab - Foundational Series Instances | EC2 Management

console.aws.amazon.com/ec2/v2/home?region=us-east-1#Instances:

New EC2 Experience Learn more

Services EC2 Dashboard Events Tags Limits Instances Instances New Instance Types Launch Templates Spot Requests Savings Plans Reserved Instances New Dedicated Hosts Scheduled Instances Capacity Reservations Images AMIs Elastic Block Store Volumes Snapshots Lifecycle Manager New Network & Security Security Groups

Search for services, features, marketplace products, and docs [Option+S]

voclabs/user1599877=Test_Student @ 5553-6113-4515 N. Virginia Support

Instances (1/1) Info

Filter instances

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
-	i-0f6e960a9d85d9340	Running	t2.micro	Initializing	No alarms	us-east-1c	ec2-50-17-143-32.c...

Instance: i-0f6e960a9d85d9340

Details Security Networking Storage Status checks Monitoring Tags

Generate by CloudWatch Manage detailed monitoring

1h 3h 12h 1d 3d 1w Custom

CPU utilization (%) Status check failed (any) (co... Status check failed (instance...) Status check failed (system) ...

Network in (bytes) Network out (bytes) Network packets in (count) Network packets out (count)

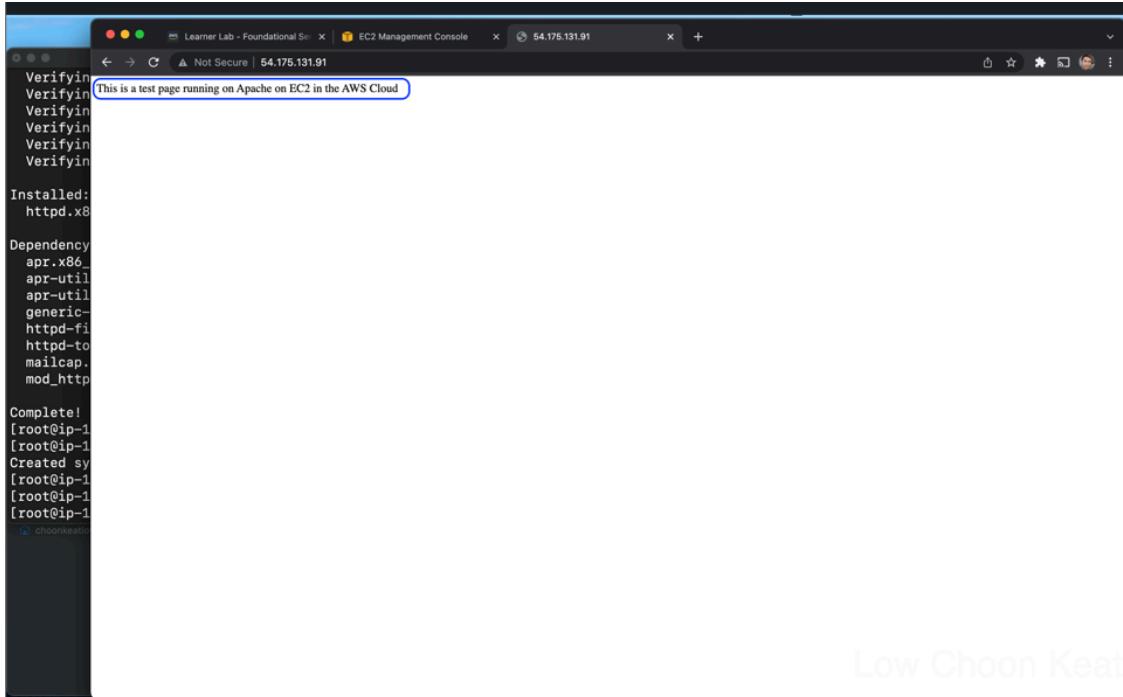
Feedback English (US) © 2006 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

Loy Cheen Keat

The screenshot shows the AWS EC2 Instances page with a single instance listed. The 'Monitoring' tab is selected in the instance details panel. The monitoring dashboard displays four metrics: CPU utilization, Network in (bytes), Network out (bytes), and Network packets in/out (count). Each metric chart shows a single data series for the instance 'i-0f6e960a9d85d9340'. The charts indicate no data available for the specified time range.

Connecting to Amazon EC2 Instance

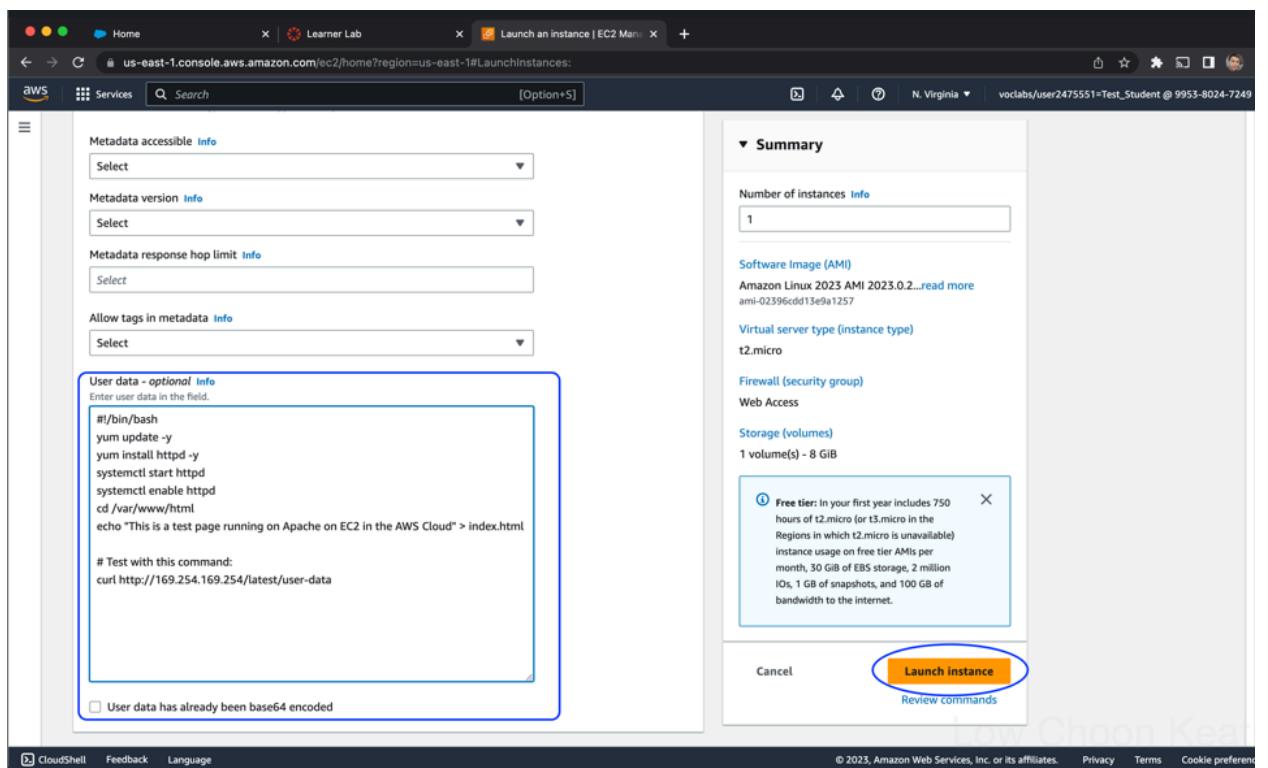
1. In **Instances tab**, select the instance, click on "Connect" button
2. Access "**SSH Client**" tab, **copy the Example ssh**
3. Go to leaner lab tab, click on "**AWS Details**" → click on "**Download PEM**" button
4. **Rename** the .pem file into "**vockey.pem**"
5. Open **Windows terminal** (can use "dir" to check whether the file .pem file is in the current directory)
6. **Paste the Example ssh** into the Windows terminal
7. Enter "**sudo yum update -y**" (can use "dnf" instead of "yum")
8. Open "**ec2-instance-user-data.sh**" in "Code and Files" folder.
9. Copy "**yum install httpd -y**" and paste into the Windows terminal ("httpd" is Apache webserver)
10. Go to browser, open the "**Security Groups**" tab in "**Network & Security**" section
11. Select the "**Web-Access**" Security Group → click on "**Edit inbound rules**"
12. Click on "**Add rule**"
13. Select "**HTTP**" type and "**Anywhere-IPv4**" Source with 0.0.0.0/0
14. Click on "**Save rules**"
15. Go to **Windows terminal**, enter "**systemctl start httpd**"
16. Go to browser, open "**Instances**" tab → select the instance
17. Click on "**open address**" under the **Public IPv4 address**
18. Go to **Windows terminal**, enter
 - a. **systemctl enable httpd**
 - b. **cd /var/www/html**
 - c. **echo "This is a test page running on Apache on EC2 in the AWS Cloud" > index.html**
19. Go back to the "**IPv4 address**" page



Practical 2

Create a Website using User Data

1. Launch an instance (similar to previous practical)
2. Select "Amazon Linux"
3. Select "Amazon Linux 2023 AMI" as Amazon Machine Image (AMI)
4. Select "t2.micro" as Instance type
5. Select "vokey" as Key pair (login)
6. In Network settings, select "Select existing security group" and select "Web Access" as Security groups (since previously already open port 22 for ssh and port 80 for web server at Web Access security group)
7. Open "[ec2-instance-user-data.sh](#)" in "Code and Files" folder.
8. Copy all the codes.
9. In Advanced details, paste the commands into "User data - optional"



10. Launch instance
11. Go to Instances tab, select the instance

12. Go to **Status checks** tab under the **selected instance**, check the "**System status checks**" and "**Instance status checks**"

The screenshot shows the AWS EC2 Management Console interface. On the left, there's a sidebar with various navigation options like EC2 Dashboard, Events, Tags, Limits, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations, Images, AMIs, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, and Security Groups. The main area shows a table of instances with one row selected. The selected instance is 'i-0f7fe715ada757aa0' (Status: Running, Type: t2.micro). A blue circle highlights the 'Status check' button in the top right of the table header. Another blue circle highlights the '2/2 checks passed' status in the same row. Below the table, a modal window titled 'Instance: i-0f7fe715ada757aa0' is open, showing tabs for Details, Security, Networking, Storage, Status checks (selected), Monitoring, and Tags. The 'Status checks' tab displays two sections: 'System status checks' (with a blue box around it) and 'Instance status checks' (also with a blue box around it). Both sections show a green checkmark and the text 'System reachability check passed'. There's also a 'Report instance status' button.

13. After everything ok, go to details tab under the selected instance, copy the "Public IPv4 address"

14. Open new browser tab, paste the url inside and run

15. After everything ok, terminate the instance

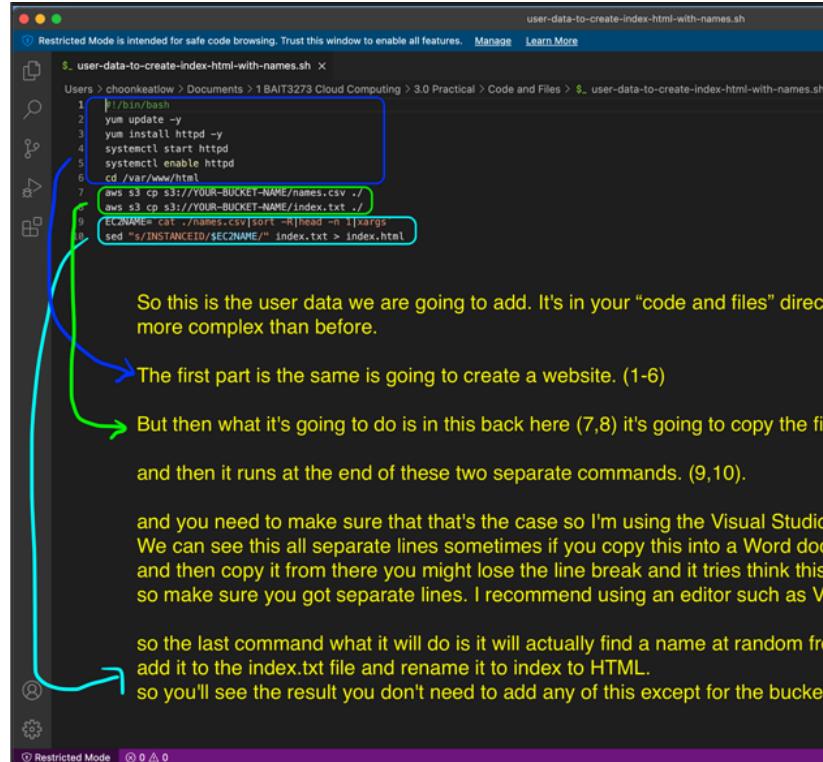
Website using S3 User Data

1. Search for **S3** → Click on "S3"
2. **Create bucket**
3. Enter **Bucket name**
4. Select **AWS Region "US East (N. Virginia) us-east-1"**
5. **Create bucket**
6. Click into the **created bucket**
7. In **Objects** tab, click on "**Upload**" button
8. Click on "**Add files**" button
9. Select and insert the "**index.txt**" and "**names.csv**" file in the "Website" folder
10. Click on "**Upload**" button
11. Go to Instance tab, **launch instance** (similar to previous practical)
12. Everything same until the command to be pasted into the "User data - optional":
 - a. Open the "[user-data-to-create-index-html-with-names.sh](#)" file from the "Code and Files" folder
 - b. Modify the <YOUR-BUCKET-NAME> with your created bucket name:

The screenshot shows the AWS S3 console interface. On the left, the navigation pane includes 'Amazon S3' with 'Buckets' selected, 'Access Points', 'Object Lambda Access Points', 'Multi-Region Access Points', 'Batch Operations', and 'Access analyzer for S3'. Below that are sections for 'Block Public Access settings for this account', 'Storage Lens', 'Dashboards', 'AWS Organizations settings', and 'Feature spotlight'. At the bottom, there's a link to 'AWS Marketplace for S3'.

The main area displays the contents of the 'lowchoonkeat-user-data' bucket. The 'Objects' tab is active, showing two objects: 'index.txt' (Type: txt, Last modified: September 20, 2021, 13:31:55 (UTC+08:00), Size: 215.0 B, Storage class: Standard) and 'names.csv' (Type: csv, Last modified: September 20, 2021, 13:32:04 (UTC+08:00), Size: 993.5 KB, Storage class: Standard). A context menu is open over the 'index.txt' object, with 'Copy' highlighted. Other options in the menu include 'Copy Link to Highlight', 'Search Google for "lowchoonkeat-user-data"', 'Print...', 'Inspect', 'Speech', and 'Services'.

At the bottom of the page, there are links for 'Feedback', 'English (US)', 'Privacy Policy', 'Terms of Use', and 'Cookie preferences'. The footer also includes a copyright notice: '© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.'



```
$ user-data-to-create-index-html-with-names.sh
1 #!/bin/bash
2 yum update -y
3 yum install httpd -y
4 systemctl start httpd
5 systemctl enable httpd
6 cd /var/www/html
7 aws s3 cp s3://YOUR-BUCKET-NAME/names.csv ./
8 aws s3 cp s3://YOUR-BUCKET-NAME/index.txt ./
9 EC2NAME=$(cat ./names.csv|sort -R|head -n 1|xargs
10 sed "s/INSTANCEID/$EC2NAME/" index.txt > index.html)
```

So this is the user data we are going to add. It's in your "code and files" directory and it's a little bit more complex than before.

The first part is the same is going to create a website. (1-6)

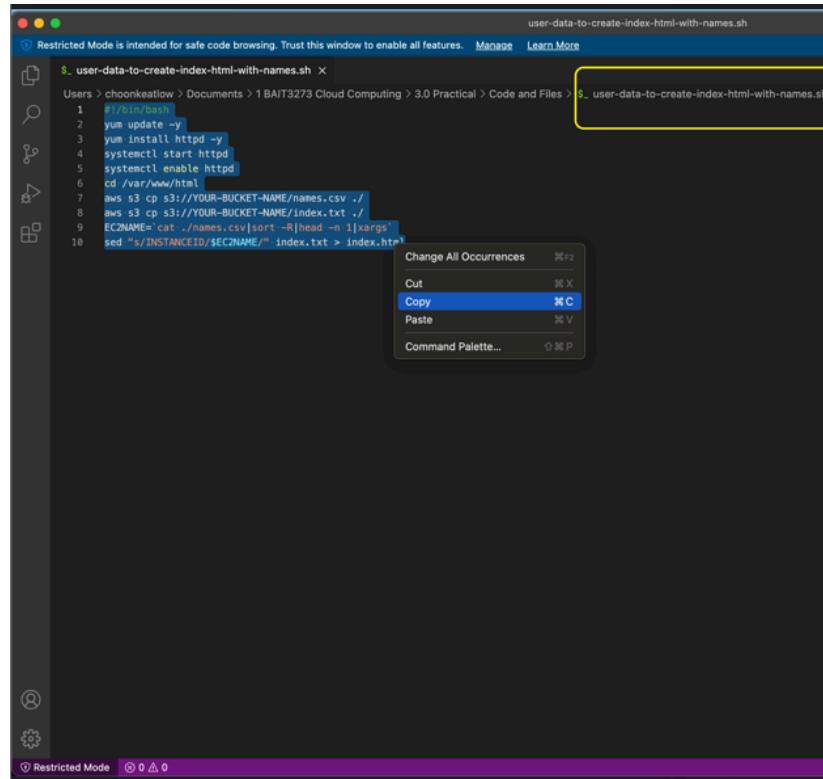
But then what it's going to do is in this back here (7,8) it's going to copy the files from my S3 bucket.

and then it runs at the end of these two separate commands. (9,10).

and you need to make sure that that's the case so I'm using the Visual Studio code and it is here. We can see this all separate lines sometimes if you copy this into a Word document for example and then copy it from there you might lose the line break and it tries think this is one line. so make sure you got separate lines. I recommend using an editor such as Visual Studio code.

so the last command what it will do is it will actually find a name at random from this file and then add it to the index.txt file and rename it to index to HTML.
so you'll see the result you don't need to add any of this except for the bucket name.

Low Choon Keat



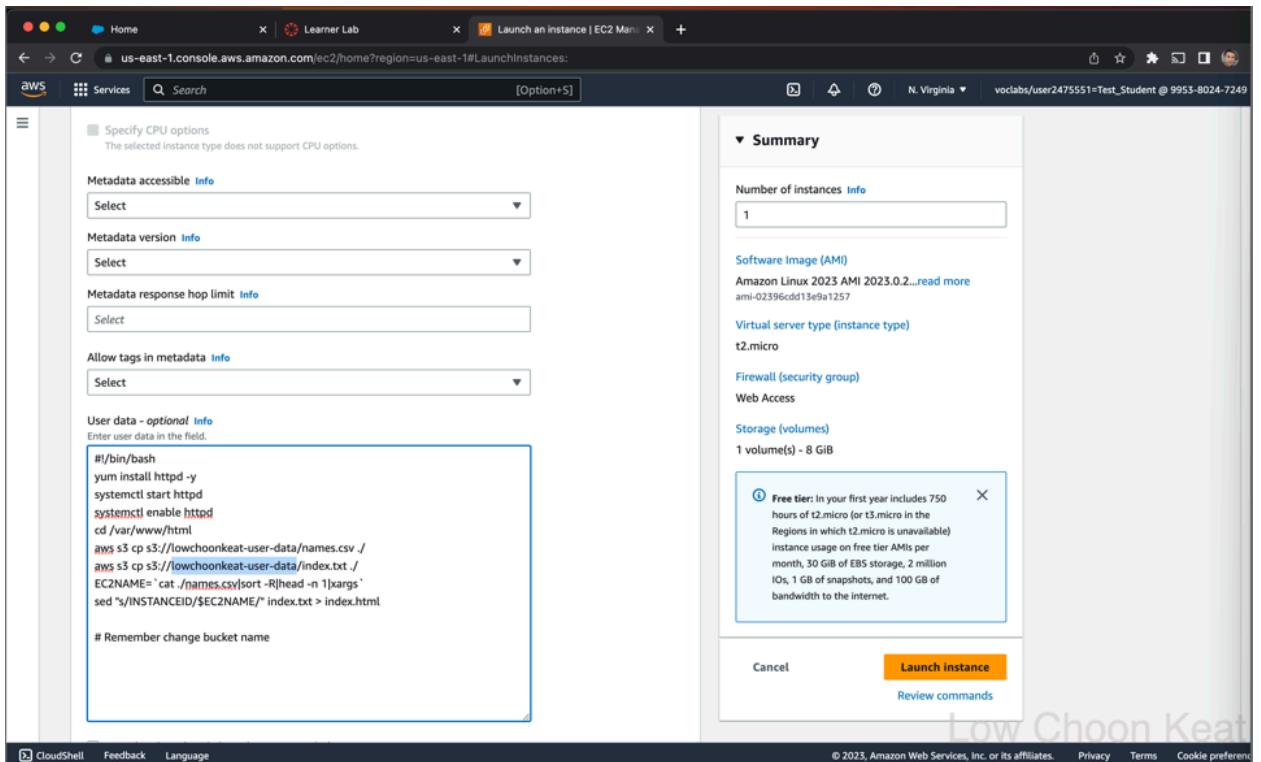
```
$ user-data-to-create-index-html-with-names.sh
1 #!/bin/bash
2 yum update -y
3 yum install httpd -y
4 systemctl start httpd
5 systemctl enable httpd
6 cd /var/www/html
7 aws s3 cp s3://YOUR-BUCKET-NAME/names.csv ./
8 aws s3 cp s3://YOUR-BUCKET-NAME/index.txt ./
9 EC2NAME=$(cat ./names.csv|sort -R|head -n 1|xargs
10 sed "s/INSTANCEID/$EC2NAME/" index.txt > index.html)
```

Change All Occurrences

- Cut ⌘X
- Copy ⌘C**
- Paste ⌘V

Low Choon Keat

13. Paste the command into "User data - optional"



14. In Advanced details tab, select "LabInstanceProfile" as IAM instance profile

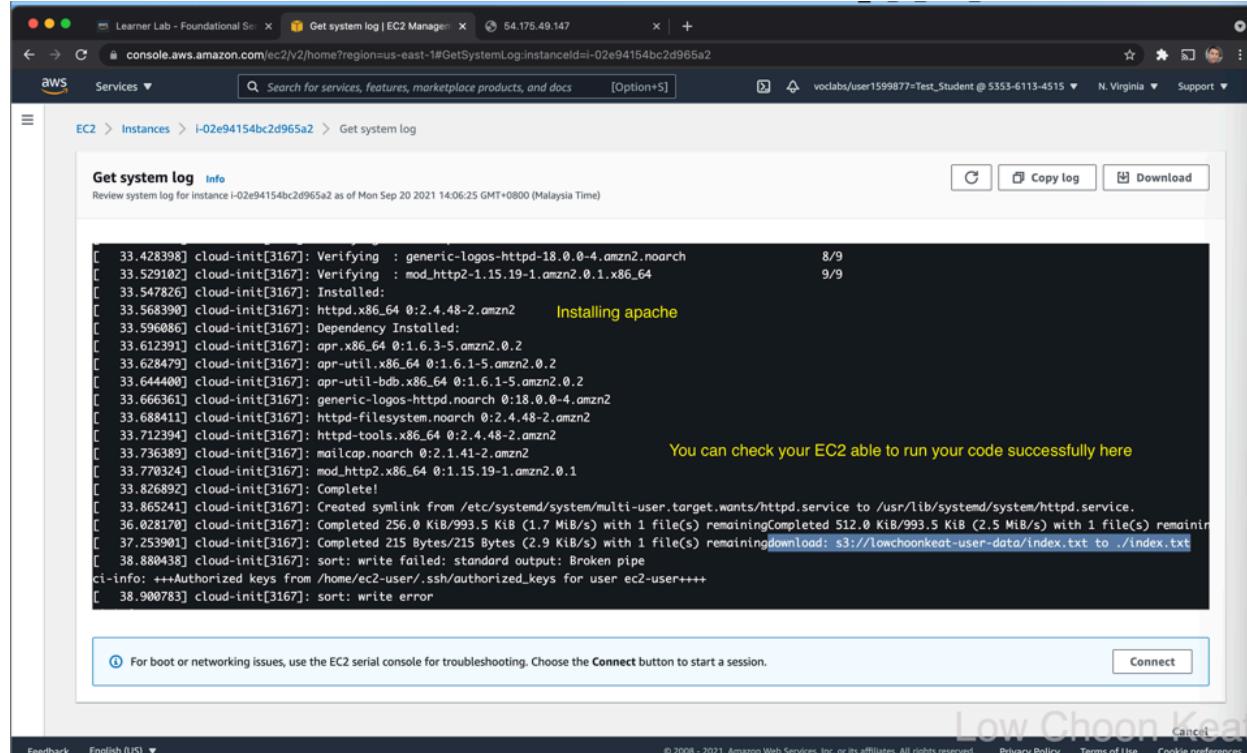
15. Launch instance

16. Copy "Public IPv4 address"

17. Paste the url in browser new tab and run

18. To monitor and troubleshoot,

- a. Select the **instance**, click on "Actions" → "Monitor and troubleshoot" → "Get system log"



The screenshot shows the AWS EC2 Management Console with the URL `console.aws.amazon.com/ec2/v2/home?region=us-east-1#GetSystemLog:instanceId=i-02e94154bc2d965a2`. The page title is "Get system log | EC2 Manager". The instance ID is `i-02e94154bc2d965a2`. The log output is as follows:

```

[ 33.428398] cloud-init[3167]: Verifying : generic-logos-httd-18.0.0-4.amzn2.noarch      8/9
[ 33.529102] cloud-init[3167]: Verifying : mod_http2-1.15.19-1.amzn2.0.1.x86_64      9/9
[ 33.547826] cloud-init[3167]: Installed:
[ 33.568390] cloud-init[3167]: httpd.x86_64 0:2.4.48-2.amzn2      Installing apache
[ 33.596086] cloud-init[3167]: Dependency Installed:
[ 33.612391] cloud-init[3167]: apr.x86_64 0:1.6.3-5.amzn2.0.2
[ 33.628479] cloud-init[3167]: apr-util.x86_64 0:1.6.1-5.amzn2.0.2
[ 33.644400] cloud-init[3167]: apr-util-bdb.x86_64 0:1.6.1-5.amzn2.0.2
[ 33.666361] cloud-init[3167]: generic-logos-httd.noarch 0:18.0.0-4.amzn2
[ 33.688411] cloud-init[3167]: httpd-filesystem.noarch 0:2.4.48-2.amzn2
[ 33.712394] cloud-init[3167]: httpd-tools.x86_64 0:2.4.48-2.amzn2
[ 33.736389] cloud-init[3167]: mailcap.noarch 0:2.1.41-2.amzn2
[ 33.779324] cloud-init[3167]: mod_http2.x86_64 0:1.15.19-1.amzn2.0.1
[ 33.826892] cloud-init[3167]: Complete!
[ 33.865241] cloud-init[3167]: Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.
[ 36.028170] cloud-init[3167]: Completed 256.0 KiB/993.5 KiB (1.7 MiB/s) with 1 file(s) remainingCompleted 512.0 KiB/993.5 KiB (2.5 MiB/s) with 1 file(s) remaining
[ 37.253901] cloud-init[3167]: Completed 215 Bytes/215 Bytes (2.9 KiB/s) with 1 file(s) remainingdownload: s3://lowchoonkeat-user-data/index.txt to ./index.txt
[ 38.880438] cloud-init[3167]: sort: write failed: standard output: Broken pipe
ci-info: +--Authorized keys from /home/ec2-user/.ssh/authorized_keys for user ec2-user+ ++
[ 38.900783] cloud-init[3167]: sort: write error

```

A yellow message bar at the top right says "You can check your EC2 able to run your code successfully here". A note below the log says "For boot or networking issues, use the EC2 serial console for troubleshooting. Choose the Connect button to start a session." A "Connect" button is available.

Using Access Keys with EC2

1. Go to "Amazon S3", check for the bucket whether the bucket name is correct
2. Go to "Instances", select the instance
3. Under the "Details" section, check whether the "IAM Role" is "LabRole". If not, click on "Actions" -> "Modify IAM role"
4. Click on "Connect" button, under the "SSH Client" tab, [copy the Example ssh](#).
5. Open [Windows terminal](#), paste the command inside and run for connecting to EC2
6. Display the files in S3 using command "[aws s3 ls s3://<your-bucket-name>](#)"
7. After everything ok, go to "Instances" tab, select instance, click on "Instance state" button → Terminate instance
8. No need delete S3 bucket, it will be used in later lesson load balancing

Practical 3

Auto Scaling

1. From the left sidebar, find for the "Auto Scaling Groups" → click on it
2. **Create** Auto Scaling group
3. Enter Auto Scaling group **name**
4. In **Launch template**, click on "Create a launch template"
5. Enter "**LC1**" as the "**Launch template name - required**"
6. Under "**Amazon machine image (AMI) - required**", select "**Amazon Linux 2 AMI (HVM), SSD Volume Type**" as **AMI**
7. Under "**Instance type**", select "**t2.micro**"
8. Under "**Key pair (login)**", select "**vockey**"
9. Under Network settings security group, select "**Web Access**"
10. Create launch template
11. After launch template created, click on "**Create Auto Scaling group**"
12. Under "**Choose launch template or configuration**",
 - a. enter Auto Scaling group name "**ASG1**"
 - b. select "**LC1**" as **Launch template**
13. Under "**Configure settings**",
 - a. Select "**Adhere to launch template**" as **Instance purchase options**
 - b. Select "**us-east-1a**", "**us-east-1b**" and "**us-east-1c**" as the **Subnets**
14. Under "**Configure advanced options**",
 - a. Select "**No load balancer**" in **Load balancing**
15. Under "**Configure group size and scaling policies**",
 - a. Enter **3** as "**Desired capacity**"
 - b. Enter **2** as "**Minimum capacity**"
 - c. Enter **4** as "**Maximum capacity**"
 - d. Select "**None**" in **Scaling policies**

16. Create Auto Scaling group

The screenshot shows the AWS EC2 Management Console with the URL console.aws.amazon.com/ec2autoscaling/home?region=us-east-1#/details. On the left, there's a sidebar with various EC2 services like Instances, Images, and Auto Scaling. The main area shows a table titled "Auto Scaling groups (1)". The table has columns for Name, Launch template/configuration, Instances, Status, Desired capacity, Min, Max, and Available. One row is listed: "ASG1 | Version Default" with 0 instances, status "Updating capacity 3", and desired capacity 3. A blue oval highlights the "Status" column header and the "Updating capacity 3" entry. A green banner at the top right says "ASG1 created successfully". Below the table, a note says "So that will now start launching instances on EC2. And we can see that it's now in the status of updating capacity. And we have this desire capacity of three. So let's head up to have a look and see if it's actually launching any instances yet."

17. Go to Instances page, check the 3 instances whether has been running and passed

The screenshot shows the AWS EC2 Management Console Instances page with the URL console.aws.amazon.com/ec2/v2/home?region=us-east-1#instances. The sidebar includes options like Reserved Instances, Images, and Auto Scaling. The main table lists three instances under the heading "Instances (3/4) Info". The columns include Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4 DNS. All three instances are listed as "Running" with "2/2 checks passed". A blue oval highlights the "Status check" column header and the "2/2 checks passed" entries. A note below the table says "So that just took a couple of refreshes. And now we can see that it is launching three instances. So they're all in the state of initialising. So we just give those a couple of minutes. And we now have all three instances are in the running state. And we can see that their status checks have passed. So you can always look at the status checks as well to see here we have the system reachability check and the instance reachability check. And those are both passed. So these are three healthy instances running on AWS." Below the table, there are four monitoring charts: CPU utilization (%), Status check failed (any) (count), Status check failed (instance) (count), and Status check failed (system) (count). Each chart shows data for the three instances.

18. Go back to auto scaling groups page, click on "ASG1" → "Activity" for checking the log

The screenshot shows the AWS EC2 Management Console interface. On the left, there is a navigation sidebar with various services like EC2 Dashboard, Instances, Images, Elastic Block Store, and Network & Security. The main content area is titled "Activity notifications (0)" and "Activity history (3)". A message overlay states: "Now if we come down to auto scaling groups, we can click on ASG1, click on activity, and we can see the log here, that tells us that three successful launches took place. So what happens if we have a failure? Well, let's test that out now. So let's come back up." Below this message is a table showing three successful launches:

Status	Description	Cause	Start time
Successful	Launching a new EC2 instance: i-08830f7f56f906153	At 2021-09-10T07:36:21Z a user request created an AutoScalingGroup changing the desired capacity from 0 to 3. At 2021-09-10T07:36:42Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 3.	2021 September 10, 03:36:44 PM +08:00
Successful	Launching a new EC2 instance: i-0281d5ac83e6df3da	At 2021-09-10T07:36:21Z a user request created an AutoScalingGroup changing the desired capacity from 0 to 3. At 2021-09-10T07:36:42Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 3.	2021 September 10, 03:36:44 PM +08:00
Successful	Launching a new EC2 instance: i-0887b15757606e47b	At 2021-09-10T07:36:21Z a user request created an AutoScalingGroup changing the desired capacity from 0 to 3. At 2021-09-10T07:36:42Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 3.	2021 September 10, 03:36:44 PM +08:00

19. Go back to Instances page, test out the auto scaling groups by terminating 2 instances to simulate the "application failure"

The screenshot shows the AWS EC2 Instances page. There are four instances listed:

- Bastion Host (i-04022913726a38753) - Running, t2.micro, 2/2 checks passed, No alarms
- (i-0887b15757606e47b) - Running, t2.micro, 2/2 checks passed, No alarms
- (i-08830f7f56f906153) - Running, t2.micro, 2/2 checks passed, No alarms
- (i-0281d5ac83e6df3da) - Running, t2.micro, 2/2 checks passed, No alarms

A modal window titled "Terminate 2 instances" is open, listing the selected instances:

Instances: i-08830f7f56f906153, i-0887b15757606e47b

Monitoring

And I'm going to take two of these instances. And I'm going to choose terminate. So I'm just you know, manually doing this, but this could have been some kind of failure of your application, something's going wrong, and the system status checks are no longer working. So that means that the auto scaling group is going to notice that that's happened. And then it should then react to that state, and launch replacement instances. So let's head back down. Let's go to auto scaling groups. Click on our ASG1 again, and click on activity. Now there's nothing here yet, but I'm going to refresh. And very soon, I think that we'll see a log mentioning that something's happened.

20. Go back to auto scaling groups page, click on "ASG1" → "Activity" for checking the log

The screenshot shows the AWS Auto Scaling Activity log for ASG1. The log details the following events:

- Terminating EC2 instance: i-08830f7f56f906153 at 2021-09-10T07:54:51Z in response to an EC2 health check. Status: Successful.
- Launching a new EC2 instance: i-08ca5d210707f4ef4 at 2021-09-10T07:52:52Z in response to a difference between desired and actual capacity. Status: Successful.
- Terminating EC2 instance: i-0887b15757606e47b at 2021-09-10T07:52:32Z in response to an EC2 health check. Status: Successful.
- Launching a new EC2 instance: i-08830f7f56f906153 at 2021-09-10T07:36:42Z in response to a difference between desired and actual capacity. Status: Successful.
- Launching a new EC2 instance: i-0281d5ac83e6df3da at 2021-09-10T07:36:42Z in response to a difference between desired and actual capacity. Status: Successful.
- Launching a new EC2 instance: i-0887b15757606e47b at 2021-09-10T07:36:42Z in response to a difference between desired and actual capacity. Status: Successful.

So within a couple of minutes, the auto scaling group did notice that something had happened. So we can see it did this termination. So we got a log entry here saying that a terminated and EC2 instance. And that was terminated in response to EC2 health status. Well, we know that we did it manually. But anyway, that's what it's saying. And then launched a new EC two instance, in response to a difference between the desired and actual capacity. And then we can see that second instance that we terminated, also got logged here, and we can see it's in the process at the moment of launching another instance to replace that one, and get us back up to our desired capacity, which should be three.

Auto Scaling up new EC2

1 EC2 stopped

1 EC2 stopped

21. Go back to Instances page, check the new instances

So if we head back up to instances, we can now see that we've got three instances back now the last one is the very first one to be launched by the ASG1. That's initializing. But one of these was also replacement for the instances that we terminated.

Instances: i-08ca5d210707f4ef4, i-099f133a2fb0f422, i-0281d5ac83e6df3da

So that's to show you how auto scaling group can respond to failures and helps that your application to be more resilient. In the next lesson, what we're going to do is we're going to look at how we can create a policy so that our auto scaling group is going to grow based on actual usage of the CPUs of our instances.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
Bastion Host	i-04022913726a38753	Running @Q	t2.micro	2/2 checks passed	No alarms +	us-east-1a	ec2-44-198-56-232
-	i-0887b15757606e47b	Terminated@Q	t2.micro	-	No alarms +	us-east-1a	-
<input checked="" type="checkbox"/>	i-08ca5d210707f4ef4	Running @Q	t2.micro	Initializing	No alarms +	us-east-1a	ec2-100-27-33-179
-	i-08830ff7f56f906153	Terminated@Q	t2.micro	-	No alarms +	us-east-1b	-
<input checked="" type="checkbox"/>	i-099f133a2fb0f422	Running @Q	t2.micro	Initializing	No alarms +	us-east-1b	ec2-3-86-231-108
<input checked="" type="checkbox"/>	i-0281d5ac83e6df3da	Running @Q	t2.micro	2/2 checks passed	No alarms +	us-east-1c	ec2-18-212-209-23

Auto Scaling with Scaling Policy

1. In Instances page, connect and login 3 of the EC2 instances

So let's see how we go about doing this. What we're going to do is login to our EC2 instances, all three of them.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
Bastion Host	i-04022913726a38755	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-44-198-56-23
-	i-0887b15757606e47b	Terminated	t2.micro	-	No alarms	us-east-1a	-
-	i-08ca5d210707f4ef4	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-100-27-33-179
-	i-08830f7f56f906153	Terminated	t2.micro	-	No alarms	us-east-1b	-
-	i-099f133a2fbdf422	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-3-86-231-108
-	i-0281d5ac83e6df3da	Running	t2.micro	2/2 checks passed	No alarms	us-east-1c	ec2-18-212-209-23

2. Within each instance, under **SSH Client**, copy the Example SSH
3. Make sure the **security group** is "**Web-Access**" which allows port 22 and 80. If not, select instance → Actions → Security → Change security groups

4. Open 3 Windows terminals, paste the respective Example SSH into each terminal

The screenshot shows three separate Lambda function executions. Each execution has a title bar indicating the function name and a timestamp. The main body of each execution shows the terminal session output:

```

choonkeatlow@CHOOONs-MacBook-Pro ~ % ssh -i "vokey.pem" ec2-user@ec2-100-27-33-1
79.compute-1.amazonaws.com
ssh: connect to host ec2-100-27-33-179.compute-1.amazonaws.com port 22: Operation timed out
choonkeatlow@CHOOONs-MacBook-Pro ~ % ssh -i "vokey.pem" ec2-user@ec2-100-27-33-1
79.compute-1.amazonaws.com
The authenticity of host 'ec2-100-27-33-179.compute-1.amazonaws.com' can't be established.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-100-27-33-179.compute-1.amazonaws.com' (ECDSA) to the list of known hosts.
[ec2-user@ip-172-31-83-152 ~]$ 

```

The second execution shows a warning about locale settings:

```

https://aws.amazon.com/amazon-linux-2/
-bash: warning: setlocale: LC_CTYPE: cannot change locale (UTF-8): No such file or directory
[ec2-user@ip-172-31-83-152 ~]$ 

```

The third execution shows a note about the AMI owner:

```

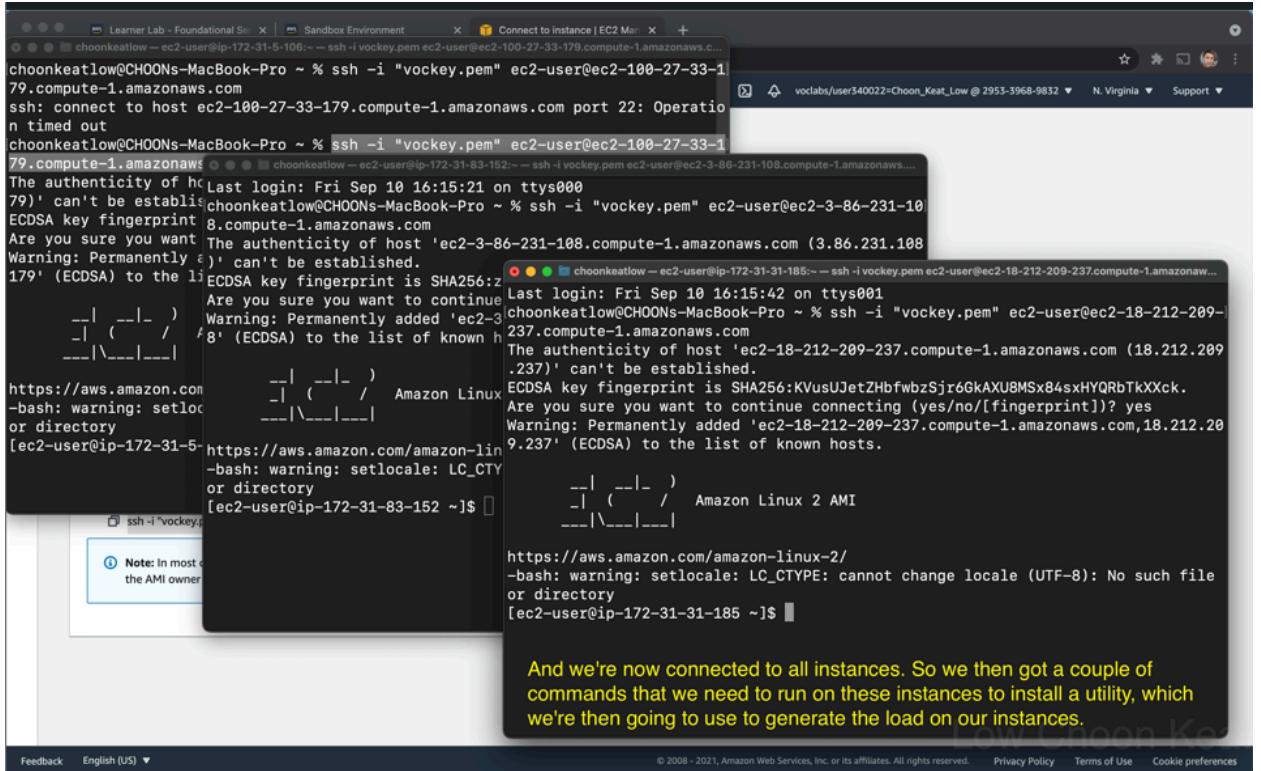
Note: In most cases, the AMI owner is the same as the Lambda function's owner.
[ec2-user@ip-172-31-83-152 ~]$ 

```

Below the terminal sessions, a note states: "And we're now connected to all instances. So we then got a couple of commands that we need to run on these instances to install a utility, which we're then going to use to generate the load on our instances."

5. Go to Auto Scaling Groups page, click on "**ASG1**"
6. Under **Automatic scaling** tab, click on "**Create dynamic scaling policy**" button
7. Enter **50** as the "**Target value**" (this value keeps the average CPU utilisation at certain value)
8. Click on "**Create**" button
9. After created, under the "**Details**" tab, adjust the Group size:
 - Desired capacity: 3
 - Minimum capacity: 2
 - Maximum capacity: 6
 - Update

10. In Instances page,

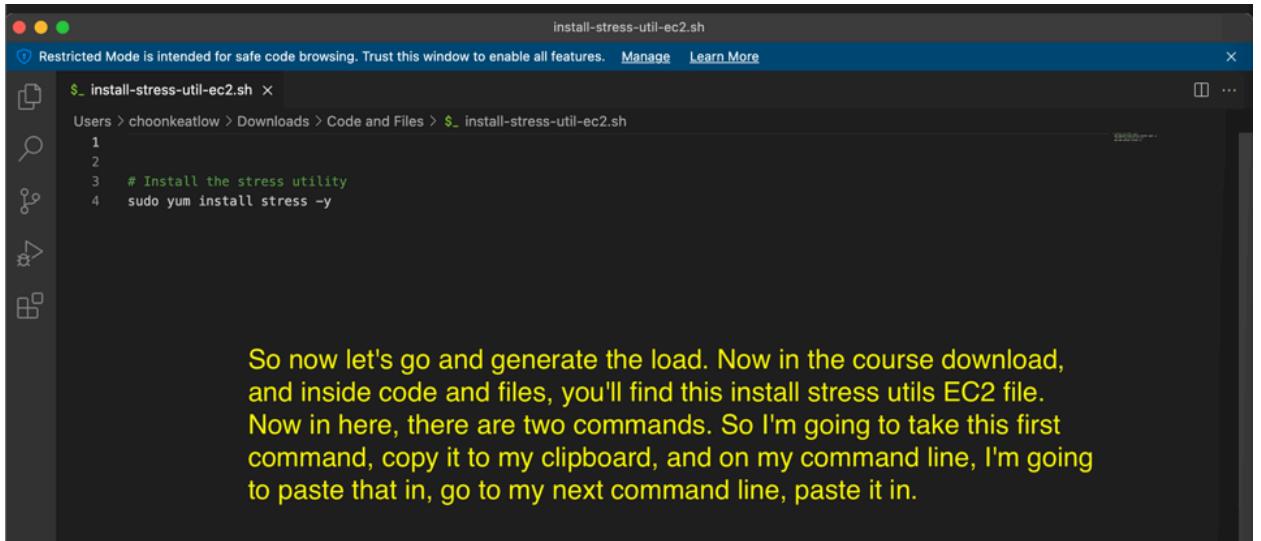


```
choonkeatlow@CHOOONs-MacBook-Pro ~ % ssh -i "vokey.pem" ec2-user@ec2-100-27-33-1
79.compute-1.amazonaws.com
ssh: connect to host ec2-100-27-33-179.compute-1.amazonaws.com port 22: Operation timed out
choonkeatlow@CHOOONs-MacBook-Pro ~ % ssh -i "vokey.pem" ec2-user@ec2-100-27-33-1
79.compute-1.amazonaws.com
The authenticity of host 'ec2-100-27-33-179.compute-1.amazonaws.com' can't be established.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-100-27-33-179.compute-1.amazonaws.com' (ECDSA) to the list of known hosts.
ECDSA key fingerprint is SHA256:z...
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-100-27-33-179.compute-1.amazonaws.com' (ECDSA) to the list of known hosts.

https://aws.amazon.com/amazon-linux-2/
[bash: warning: setlocale: LC_CTYPE: cannot change locale (UTF-8): No such file or directory
[ec2-user@ip-172-31-31-185 ~]$ ]$
```

And we're now connected to all instances. So we then got a couple of commands that we need to run on these instances to install a utility, which we're then going to use to generate the load on our instances.

11. Open "[install-stress-util-ec2.sh](#)" file in the "Codes and Files" folder



```
#!/bin/bash
# Install the stress utility
# 
sudo yum install stress -y
```

So now let's go and generate the load. Now in the course download, and inside code and files, you'll find this install stress utils EC2 file. Now in here, there are two commands. So I'm going to take this first command, copy it to my clipboard, and on my command line, I'm going to paste that in, go to my next command line, paste it in.

12. Copy the command and paste into those 2 running instances' terminals (since there is 1 instance terminated). we need to generate the stress on those 2 instances

```

39 ruby2.6           available  [ =2.6 =stable ]
40 mock              available  [ =stable ]
41 postgresql11      available  [ =11 =stable ]
42 php7.4            available  [ =stable ]
43 livepatch         available  [ =stable ]
44 python3.8          available  [ =stable ]
45 haproxy2          available  [ =stable ]
46 collectd          available  [ =stable ]
47 aws-nitro-enclaves available  [ =stable ]
48 R4                available  [ =stable ]
49 kernel-5.4         available  [ =stable ]
50 selinux-ng         available  [ =stable ]
51 php8.0             available  [ =stable ]
52 tomcat9            available  [ =stable ]
53 unbound1.13        available  [ =stable ]
54 mariadb10.5        available  [ =stable ]
55 kernel-5.10         available  [ =stable ]
56 redis6             available  [ =stable ]
57 ruby3.0             available  [ =stable ]
58 postgresql12        available  [ =stable ]
59 postgresql13        available  [ =stable ]
60 mock2              available  [ =stable ]
61 dnsmasq2.85        available  [ =stable ]
[ec2-user@ip-172-31-5-106:~] ssh -i vokey.pem ec2-user@ec2-100-27-33-179.compute-1.amazonaws.com
39 ruby2.6           available  [ =2.6 =stable ]
40 mock              available  [ =stable ]
41 postgresql11      available  [ =11 =stable ]
42 php7.4            available  [ =stable ]
43 livepatch         available  [ =stable ]
44 python3.8          available  [ =stable ]
45 haproxy2          available  [ =stable ]
46 collectd          available  [ =stable ]
47 aws-nitro-enclaves available  [ =stable ]
48 R4                available  [ =stable ]
49 kernel-5.4         available  [ =stable ]
50 selinux-ng         available  [ =stable ]
51 php8.0             available  [ =stable ]
52 tomcat9            available  [ =stable ]
53 unbound1.13        available  [ =stable ]
54 mariadb10.5        available  [ =stable ]
55 kernel-5.10         available  [ =stable ]
56 redis6             available  [ =stable ]
57 ruby3.0             available  [ =stable ]
58 postgresql12        available  [ =stable ]
59 postgresql13        available  [ =stable ]
60 mock2              available  [ =stable ]
61 dnsmasq2.85        available  [ =stable ]
[ec2-user@ip-172-31-83-152 ~]$ 

```

And of course, this is the one that's been terminated, so we can get rid of that we don't have that instance anymore. And let's go to this one. So this will be our second instance. So remember, we have two running instances, and therefore, we need to run this command on these two. So then we install this utility called stress. So I'm going to copy that to my clipboard. Come back, paste it in, paste it in. And now we've actually installed this utility on these two instances. So now we want to go and use the stress utility to generate stress on our CPU.

```

Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

Install stress

[ec2-user@ip-172-31-91-220 ~]$ sudo yum install stress -y
Last metadata expiration check: 0:00:55 ago on Thu Apr 27 09:55:31 2023.
Dependencies resolved.
=====
Package           Architecture Version       Repository   Size
stress            x86_64      1.0.4-28.amzn2023.0.2   amazonlinux 37 k
Transaction Summary
Install 1 Package

Total download size: 37 k
Installed size: 78 k
Downloading Packages:
stress-1.0.4-28.amzn2023.0.2.x86_64.rpm
                                            413 kB/s | 37 kB    00:00
Total
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing :
Installing : stress-1.0.4-28.amzn2023.0.2.x86_64
Running scriptlet: stress-1.0.4-28.amzn2023.0.2.x86_64
Verifying   : stress-1.0.4-28.amzn2023.0.2.x86_64
Installed:
stress-1.0.4-28.amzn2023.0.2.x86_64

```

13. Run "stress -c 8" to stress the instances

```
choonkeatlow ~ ec2-user@ip-172-31-5-106:~ ssh -i vockey.pem ec2-user@ec2-100-27-33-179.compute-1.amazonaws.com  
Installed size: 94 k  
Downloading packages:  
warning: /var/cache/yum/x86_64/2/epel/packages/stress-1.0.4-16.el7.x86_64.rpm: Header V3 RSA/SHA256 Signature, key ID 352c64e5: NOKEY  
Public key for stress-1.0.4-16.el7.x86_64.rpm is not installed  
stress-1.0.4-16.el7.x86_64.rpm | 39 kB 00:00  
Retrieving key from file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-7  
Importing GPG key 0x352C64E5:  
  Userid : "Fedora EPEL (7) <epel@fedoraproject.org>"  
  Fingerprint: 91e9 7d7c 4a5e 96f1 7f3e 888f 6a2f aea2 352c 64e5  
  Package : epel-release-7-11.noarch (@amzn2extra-epel)  
  From    : /etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-7  
Running transaction check  
Running transaction test  
Transaction test succeeded  
Running transaction  
  Installing : stress-1.0.4-16.el7.x86_64  
  Verifying  : stress-1.0.4-16.el7.x86_64  
Installed:  
  stress.x86_64 0:1.0.4-16.el7  
Complete!  
[ec2-user@ip-172-31-5-106 ~]$ stress -c 8
```

we're going to run the stress utility, dash c for CPU, eight. And that's going to add a number of processes to our CPU. And let's do the same on this other utility here. So those are just going to keep running now and generating load. Now, how long is it going to take to actually react? Well, it will take a bit of time. So let's head back over to EC2.

14. Wait for the instances to send information to CloudWatch. By default in free version, it will send info to CloudWatch every 5 minutes.

Learner Lab - Foundational Skills | Sandbox Environment | Instances | EC2 Management

Search for services, features, marketplace products, and docs [Option+S]

voclabs/user340022=Choon_Keat_Low @ 2953-3968-9832 N. Virginia Support

New EC2 Experience Learn more

EC2 Dashboard Events Tags Limits

Instances Instances New Instance Types Launch Templates Spot Requests Savings Plans Reserved Instances New Dedicated Hosts Scheduled Instances Capacity Reservations

Images AMIs

Elastic Block Store Volumes Snapshots Lifecycle Manager New

Network & Security Security Groups

Feedback English (US) ▾

Instances (1/6) Info

Filter Instances

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
Bastion Host	i-04022913726a38753	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1a	ec2-44-198-56-232
-	i-0887b15757606e47b	Terminated	t2.micro	-	No alarms	+ us-east-1a	-
<input checked="" type="checkbox"/> -	i-08ca5d210707f4ef4	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1a	ec2-100-27-33-175
-	i-08830f7f56f906153	Terminated	t2.micro	-	No alarms	+ us-east-1b	-
-	i-099f133a2fb0d422	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1b	ec2-3-86-231-108
-	i-0281d5ac83e6df3da	Terminated	t2.micro	-	No alarms	+ us-east-1c	-

Instance: i-08ca5d210707f4ef4

Details Security Networking Storage Status checks Monitoring Tags

basic: sent info to cloud watch every 5 mins

So on EC2, what's happening is our instances are sending information to cloud watch. But by default, they have something called basic monitoring enabled, that will send information to cloudwatch every five minutes. If you enable detailed monitoring, then you do have to pay, but you get it every one minute. So by default, it's going to take a few minutes to actually report the information about the metrics that are being recorded at the moment. So we have to wait a few minutes for that to be reported to cloud watch. And then for the auto scaling group to be notified that this is occurring. So just be patient and maybe grab a cup of coffee.

1 mins but need to pay

Manage detailed monitoring

CPU utilization (%) Status check failed (any) Status check failed (instance...) Status check failed (system) ...

Percent Count Count Count

07:45 08:00 08:15 08:30 08:45 08:50 08:55 09:00 07:45 08:00 08:15 08:30 08:45 08:50 08:55 09:00 07:45 08:00 08:15 08:30 08:45 08:50 08:55 09:00

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

Learner Lab - Foundational Skills | Learner Lab - Foundational Skills | Sandbox Environment | Instances | EC2 Management

Search for services, features, marketplace products, and docs [Option+S]

voclabs/user340022=Choon_Keat_Low @ 2953-3968-9832 N. Virginia Support

New EC2 Experience Learn more

EC2 Dashboard Events Tags Limits

Instances Instances New Instance Types Launch Templates Spot Requests Savings Plans Reserved Instances New Dedicated Hosts Scheduled Instances Capacity Reservations

Images AMIs

Elastic Block Store Volumes Snapshots Lifecycle Manager New

Network & Security Security Groups

Feedback English (US) ▾

Instances (1/8) Info

Filter Instances

Max = 6 instances

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
Bastion Host	i-04022913726a38753	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1a	ec2-44-198-56-232
<input checked="" type="checkbox"/> -	i-08ca5d210707f4ef4	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1a	ec2-100-27-33-175
-	i-03351c9ab6b919b2f	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1a	ec2-44-197-248-6
-	i-099f133a2fb0d422	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1b	ec2-3-86-231-108
-	i-0bcacf0827fbbef0f	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1b	ec2-3-88-137-874
-	i-0281d5ac83e6df3da	Terminated	t2.micro	-	No alarms	+ us-east-1c	-
-	i-0fe1ef6a918f75a21	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1c	ec2-54-91-208-19
-	i-0cc984bf1b2e54d	Running	t2.micro	2/2 checks passed	No alarms	+ us-east-1c	ec2-54-82-74-264

Instance: i-08ca5d210707f4ef4

additional 4

CPU utilization (%) Network in (bytes) Network out (bytes) Network packets in (count) Network packets out (count)

Percent Bytes Bytes Count Count

99.9 14.2M 43.8k 9.64k 846

50 0 0.5 0 0

0 08:15 08:30 08:45 09:00 0 08:15 08:30 08:45 09:00 0 08:15 08:30 08:45 09:00 0 08:15 08:30 08:45 09:00

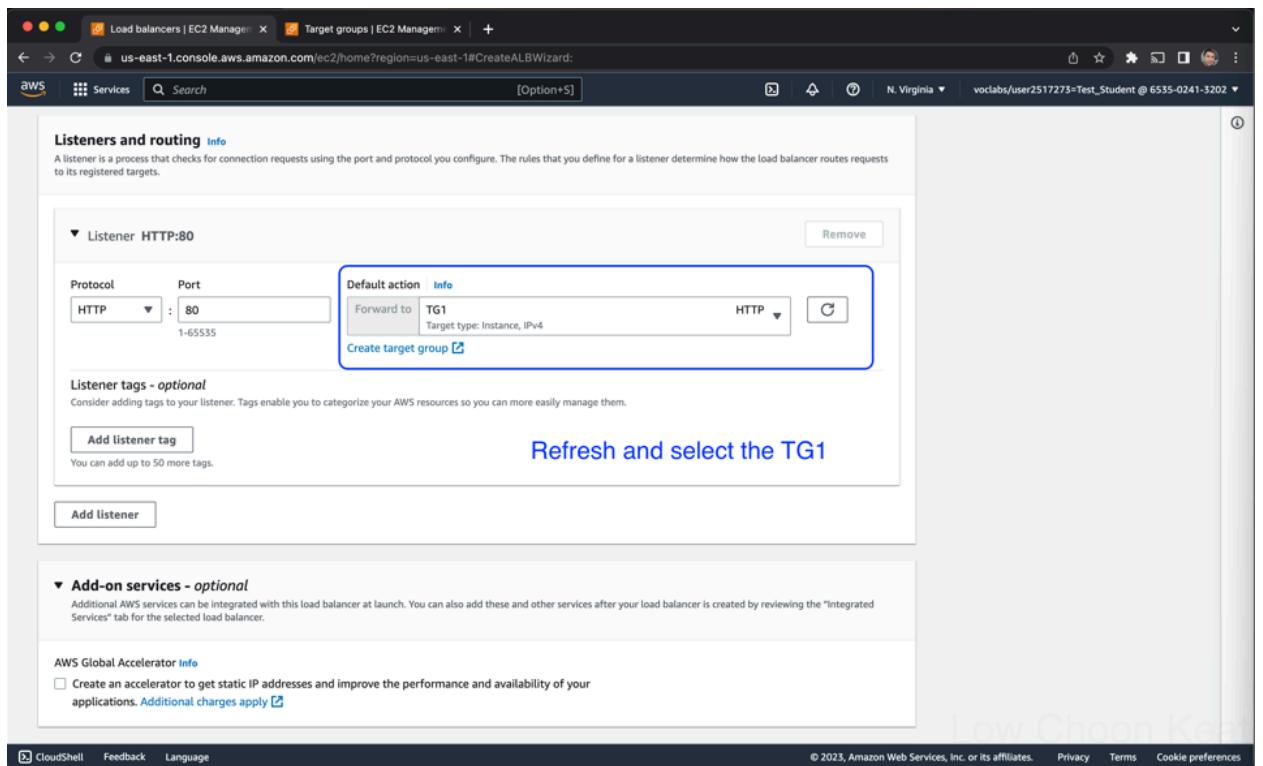
So I waited a few minutes. And we can see now that we've had some changes. So on the monitoring tab from one of my instances, we have CPU utilization here, and we can see that it jumped right up from a super low value here, right up to almost 99.9%. I've just refreshed the page. And look, here we go, we've got 4 additional instances being launched. So our auto scaling group has reacted to that information.

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

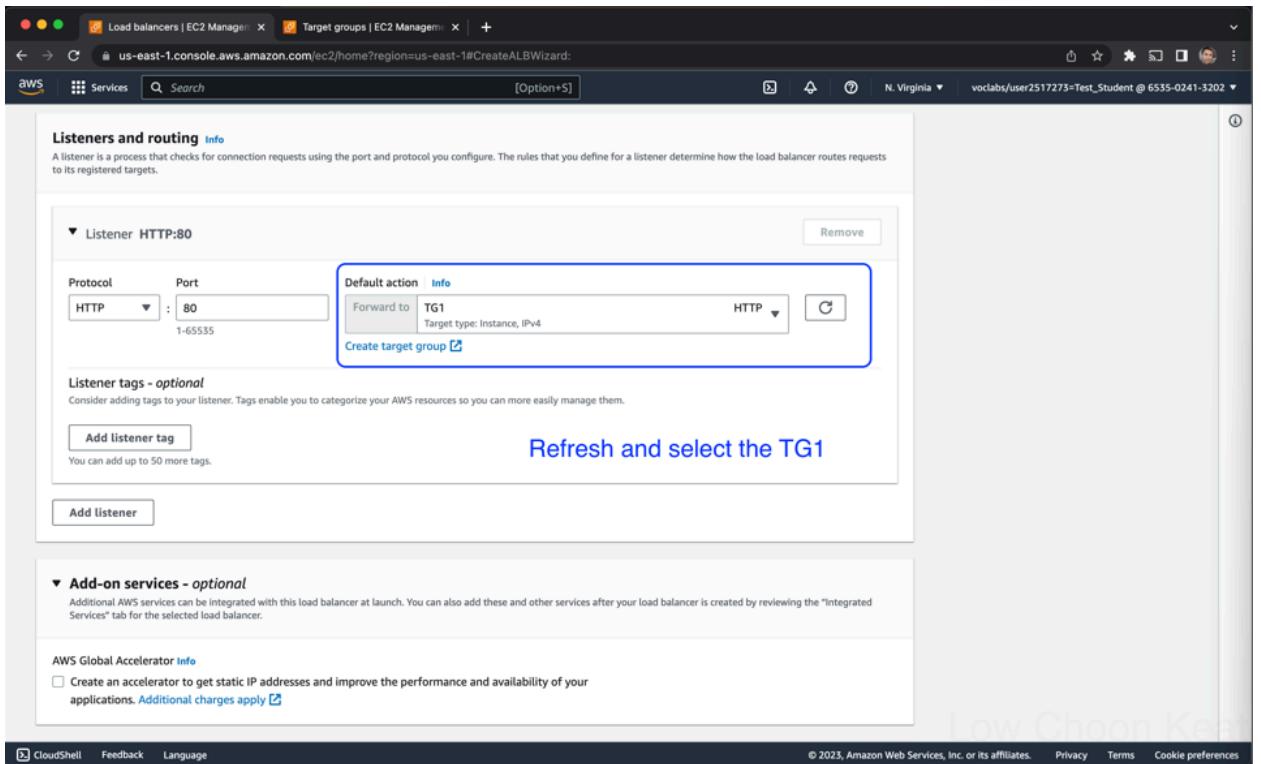
Practical 4

Add Load Balancing with Amazon ELB

1. Go to EC2 > Load Balancers page
 2. Create load balancer
 3. Select "Create" button in Application Load Balancer option
 4. Enter load balancer name "ALB"
 5. Select 4 availability zone "us-east-1a", "us-east-1b", "us-east-1c" & "us-east-1d"
 6. Select "Web-Access" as Security groups
 7. Under Listeners and routing, click on "Create target group"
 8. Enter Target group name "TG1"
 9. Click on "Create target group" button
 10. Go back to load balancer tab, under Listeners and routing, click on "Refresh" and select "TG1"



11. Click on "Create load balancer" button



12. Go to Load Balancing > Target Group page

13. Select "TG1"

14. Go to the S3 page, make sure the user-data bucket is still there. Click on the "Create bucket" button.

The screenshot shows the AWS EC2 Load Balancers page. A modal window titled "Successful created the application load balancer ALB" is open, confirming the creation of the ALB. The modal includes tabs for Details, Listeners, Network mapping, Security, Monitoring, Integrations, Attributes, and Tags. The Details tab is selected, displaying the ARN, Load balancer type (Application), DNS name (ALB-1918941183.us-east-1.elb.amazonaws.com), Status (Provisioning), VPC ID (vpc-07de6e260977633c5), Availability Zones (4 Availability Zones), Type (application), and Date created (April 2 (UTC+0)).

The screenshot shows the AWS S3 Buckets page. The bucket "lowchoonkeat-user-data" is selected. It contains two objects: "index.txt" (Type: txt, Last modified: September 20, 2021, Size: 215.0 B, Storage class: Standard) and "names.csv" (Type: csv, Last modified: September 20, 2021, Size: 993.5 KB, Storage class: Standard). The page also includes tabs for Objects, Properties, Permissions, Metrics, Management, and Access Points.

Note: So this is going to be the bucket which holds some files which I used for some "user data" that we're going to create for our ec2 instance.

15. Go to **Instances > Launch Templates page**, click on "Create launch template"
16. Enter "LT1" as Launch template name
17. Select "Amazon Linux 2023 AMI" as Amazon Machine Image (AMI)
18. Select "t3.micro" as Instance type

19. Select "vokey" as Key pair (login)
20. Under Network settings, select "Web-Access" as Security groups
21. Under Advanced details, select "**LabInstanceProfile**" as **IAM instance profile**
(obtain permissions to access S3, when these instances launch, they will use this role to access those files that we just upload to our bucket)
22. Open "[user-data-to-create-index-html-with-names.sh](#)" from "Code and Files" folder. Replace <YOUR-BUCKET-NAME> with the created S3 bucket name.

user-data-to-create-index-html-with-names.sh

```

$ user-data-to-create-index-html-with-names.sh
1 #!/bin/bash
2 yum update -y
3 yum install httpd -y
4 systemctl start httpd
5 systemctl enable httpd
6 cd /var/www/html
7 aws s3 cp s3://<YOUR-BUCKET-NAME>/names.csv ./ 
8 aws s3 cp s3://<YOUR-BUCKET-NAME>/index.txt ./ 
9 EC2NAME= cat ./names.csv | sort -R | head -n 1 | xargs
10 sed "s/INSTANCEID/$EC2NAME/" index.txt > index.html

```

So this is the user data we are going to add. It's in your "code and files" directory and it's a little bit more complex than before.

The first part is the same is going to create a website. (1-6)

But then what it's going to do is in this back here (7,8) it's going to copy the files from my S3 bucket.
and then it runs at the end of these two separate commands. (9,10).

and you need to make sure that that's the case so I'm using the Visual Studio code and it is here. We can see this all separate lines sometimes if you copy this into a Word document for example and then copy it from there you might lose the line break and it tries think this is one line.
so make sure you got separate lines. I recommend using an editor such as Visual Studio code.

so the last command what it will do is it will actually find a name at random from this file and then add it to the index.txt file and rename it to index to HTML.
so you'll see the result you don't need to add any of this except for the bucket name.

Low Choon Keat

Ln 1, Col 1 Spaces: 4 UTF-8 LF Shell Script

23. Copy the whole command and paste into the User data

The screenshot shows the AWS Management Console with the URL <https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#CreateTemplate>. The page is titled "Create launch template | EC2". On the left, there's a sidebar with "Services" and a search bar. The main area has several sections: "License configurations" (Info), "Metadata accessible" (Info), "Metadata version" (Info), "Metadata response hop limit" (Info), and "User data" (Info). The "User data" section contains a code editor with the following script:

```
#!/bin/bash
yum update -y
yum install httpd -y
systemctl start httpd
systemctl enable httpd
cd /var/www/html
aws s3 cp s3://lowchoonkeat-user-data/names.csv .
aws s3 cp s3://lowchoonkeat-user-data/index.txt .
EC2NAME=$(cat ./names.csv|sort -R|head -n 1|xargs)
sed "s/INSTANCEID/$EC2NAME/" index.txt > index.html
```

Below the code editor, there's a note: "User data has already been base64 encoded". At the bottom right, there are "Cancel" and "Create launch template" buttons.

On the right side of the screenshot, there are two text blocks:

So mine was "lowchoonkeat-user-data"

so just make sure that you add this to your bucket name cos mine probably won't exist when the time you do this.

24. Create launch template

25. Go to [Auto Scaling > Auto Scaling Groups](#) page, click on "[Create Auto Scaling Group](#)" button

26. Enter "ASG1" as Auto Scaling group name

27. Select "LT1" as Launch template

28. Click on "Next"

29. Select "Adhere to launch template" as Instance purchase option

30. Select "us-east-1a", "us-east-1b", "us-east-1c" & "us-east-1d" as subnets (add in multiple subnets so that we have instances in different availability zones)

31. Click on "Next"

32. Select "Attach to an existing load balancer" as Load balancing option

33. Select "Choose from your load balancer target groups" and select "TG1 | HTTP" as the Target group

34. **Tick the "ELB" in the Health checks section** (the elastic load balancer will notify auto scaling if it thinks that an instance is not working correctly. So, that is in addition to the EC2 status checks")

35. Click on "Next"

36. Under Group size,

- Desired capacity: 4
- Minimum capacity: 4

- c. Maximum capacity: 4
37. Select "None" as Scaling policies
 38. Click on "Next"
 39. Click on "Next"
 40. Click on "Next"

41. Create Auto Scaling Group

The screenshot shows the AWS EC2 Management Console with the URL <https://console.aws.amazon.com/ec2autoscaling/home?region=us-east-1#/details>. A green banner at the top says "ASG1 created successfully". Below it, the "Auto Scaling groups (1)" section shows a table with one row for "ASG1". The "Desired capacity" column shows "4" with a status message "Updating capacity". The "Instances" column shows "0". The "Status" column shows "Up". The "Launch template/configuration" column shows "LT1 | Version Default". The "Availability Zones" column shows "us-east-1a, us-east-1b, us-...".

So now it is going to launch those instances. It will add them to the target group for the elastic load balancer. and then we should see that the elastic load balancer can then be used to connect to these instances.

So give it a few minutes. It does need to initialise all the instances, install and update the Apache. So our website is running and then they need to be registered and become healthy with the load balancer as well.

So there's a couple of places you can monitor this. You can go into auto scaling.

The screenshot shows the "Activity" tab for the ASG1 group. The "Activity notifications (0)" section shows "We have seen this before and see what's happening here. We can see 4 instances have been launched." The "Activity history (4)" section shows four successful events:

Status	Description	Cause	Start time	End time
Successful	Launching a new EC2 instance: i-0201fb4c9e967fdf4	At 2021-09-21T07:31:54Z a user request created an AutoScalingGroup changing the desired capacity from 0 to 4. At 2021-09-21T07:31:57Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 4.	2021 September 21, 03:32:02 PM +08:00	2021 September 21, 03:32:34 PM +08:00
Successful	Launching a new EC2 instance: i-03a34a9a3eb16df82	At 2021-09-21T07:31:54Z a user request created an AutoScalingGroup changing the desired capacity from 0 to 4. At 2021-09-21T07:31:57Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 4.	2021 September 21, 03:32:01 PM +08:00	2021 September 21, 03:32:33 PM +08:00
Successful	Launching a new EC2 instance: i-0f1b8f679b6610c03	At 2021-09-21T07:31:54Z a user request created an AutoScalingGroup changing the desired capacity from 0 to 4. At 2021-09-21T07:31:57Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 4.	2021 September 21, 03:32:00 PM +08:00	2021 September 21, 03:32:33 PM +08:00
Successful	Launching a new EC2 instance: i-0317e07733f77	At 2021-09-21T07:31:54Z a user request created an AutoScalingGroup changing the desired capacity from 0 to 4. At 2021-09-21T07:31:57Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 4.	2021 September 21, 03:32:01 PM +08:00	2021 September 21, 03:32:33 PM +08:00

42. Go to Load Balancing > Target Groups, select "TG1"

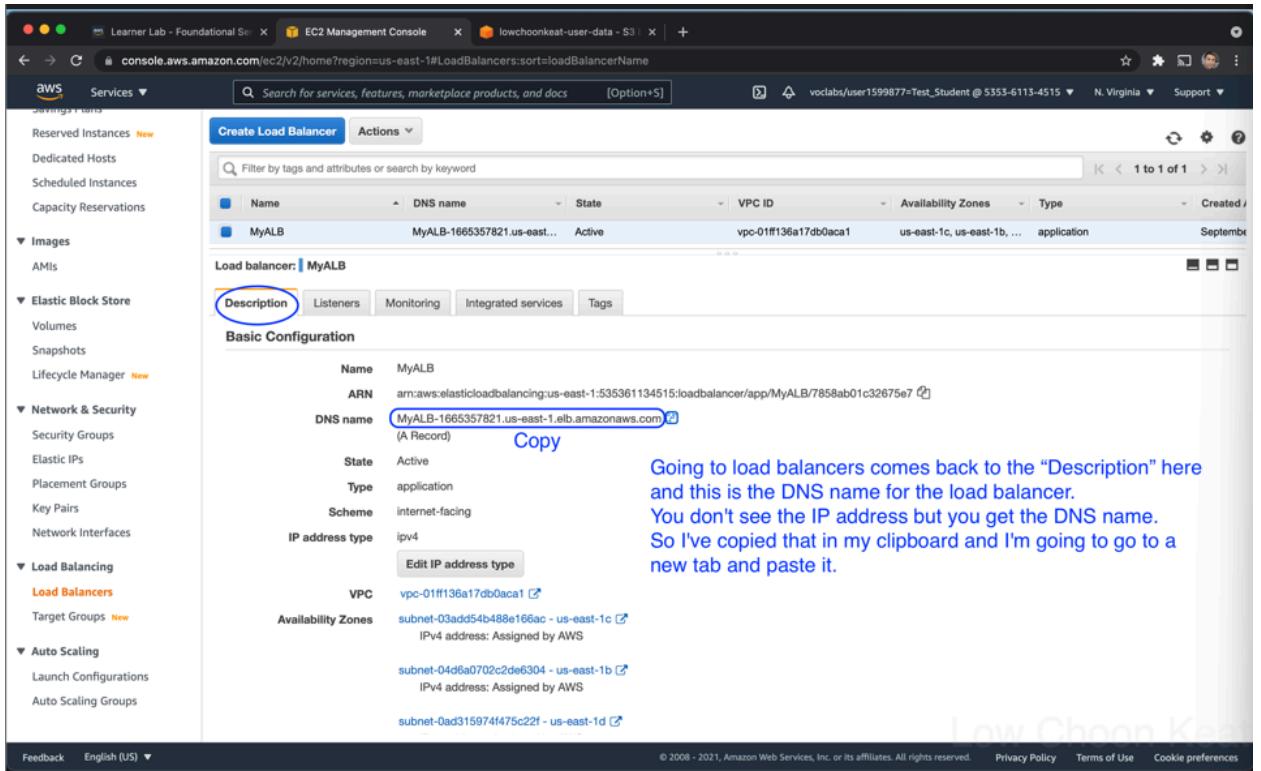
The screenshot shows the AWS EC2 Target Groups page. On the left, the navigation menu includes services like Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations, Images, Elastic Block Store, Network & Security, Load Balancing, Auto Scaling, and Target Groups. The Target Groups section is selected. The main area displays a table for Target groups (1/1) with one entry: TG1. Below it, the TG1 details page shows four registered targets, each marked as healthy with a green circle icon.

We can also go to load balancers. Go to Target groups. Click on "Targets". and we can see that there are four targets and they're now become healthy status.

43. Go to Load Balancing > Load Balancers, select "MyALB"

The screenshot shows the AWS EC2 Load Balancers page. The navigation menu is identical to the previous page, with the Load Balancing section selected. The main area shows a table for Load balancers with one entry: MyALB. Below it, the MyALB details page displays CloudWatch metrics for the selected resources over the last hour. The metrics include Target Response Time (Milliseconds), Requests (Count), Rule Evaluations (Count), HTTP 5XXs (Count), HTTP 4XXs (Count), ELB 5XXs (Count), ELB 4XXs (Count), and HTTP 500s (Count).

44. Under "Description" tab, copy the DNS name

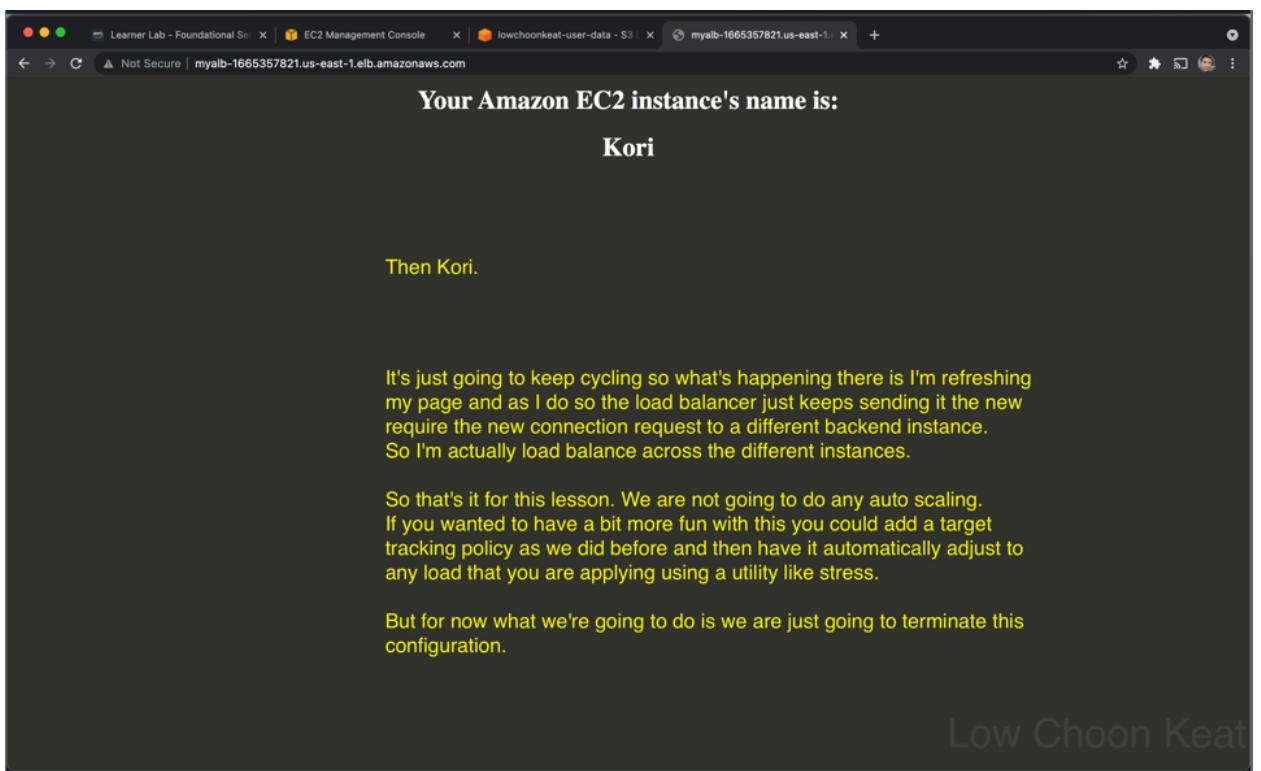
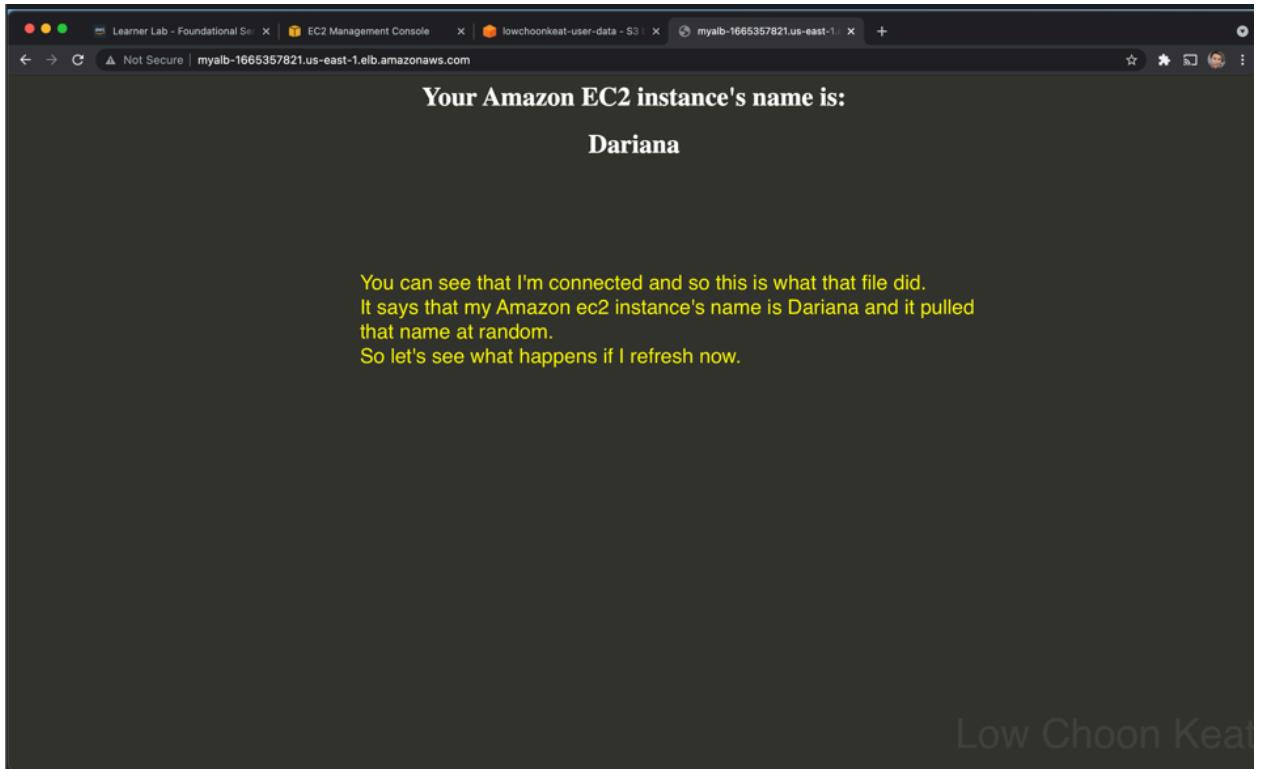


The screenshot shows the AWS EC2 Management Console with the Load Balancers page open. On the left, there's a sidebar with various AWS services like Reserved Instances, Dedicated Hosts, and Auto Scaling. The main area shows a table of load balancers. One row is selected, and its details are shown in a larger panel below. The 'Description' tab is active, indicated by a blue border. In the 'Basic Configuration' section, the 'DNS name' field contains the value 'MyALB-1665357821.us-east-1.elb.amazonaws.com'. A red box highlights this field. To the right of the field, there's a 'Copy' button. A callout box with a blue border contains the following text:

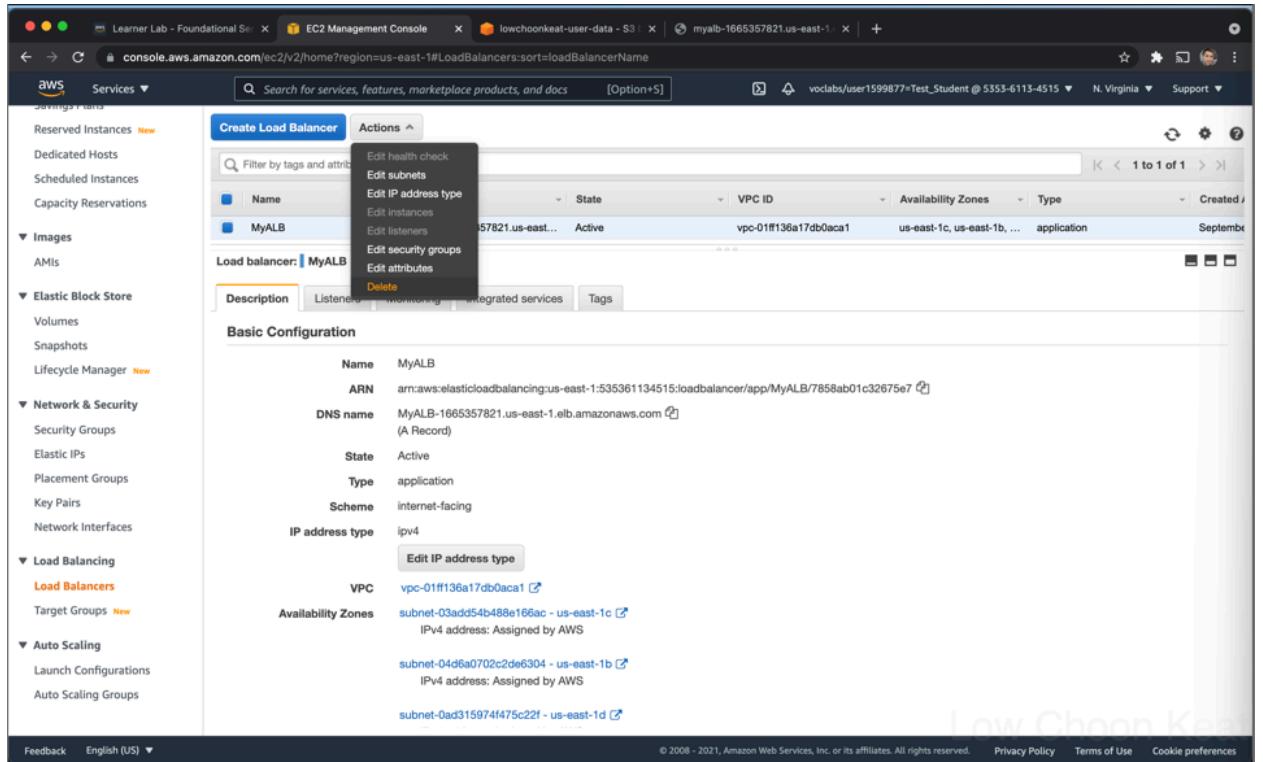
Going to load balancers comes back to the "Description" here and this is the DNS name for the load balancer. You don't see the IP address but you get the DNS name. So I've copied that in my clipboard and I'm going to go to a new tab and paste it.

Below the main panel, there are standard AWS footer links: Feedback, English (US), Privacy Policy, Terms of Use, and Cookie preferences.

45. Paste and run the url in a new browser tab



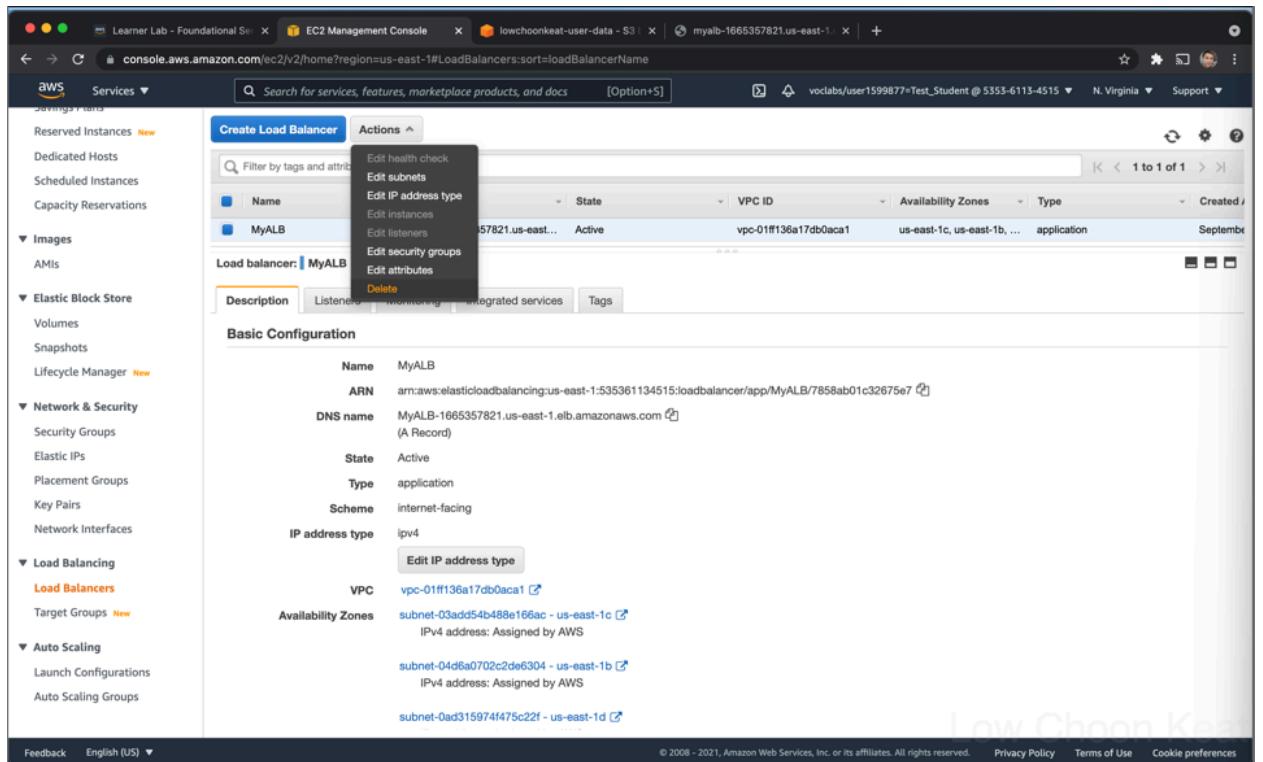
46. Go to Load Balancing > Load Balancers, delete the "MyALB"



The screenshot shows the AWS EC2 Management Console with the 'Load Balancers' section selected. A context menu is open over the 'MyALB' load balancer, with the 'Delete' option highlighted. The main table displays the following details for the 'MyALB' load balancer:

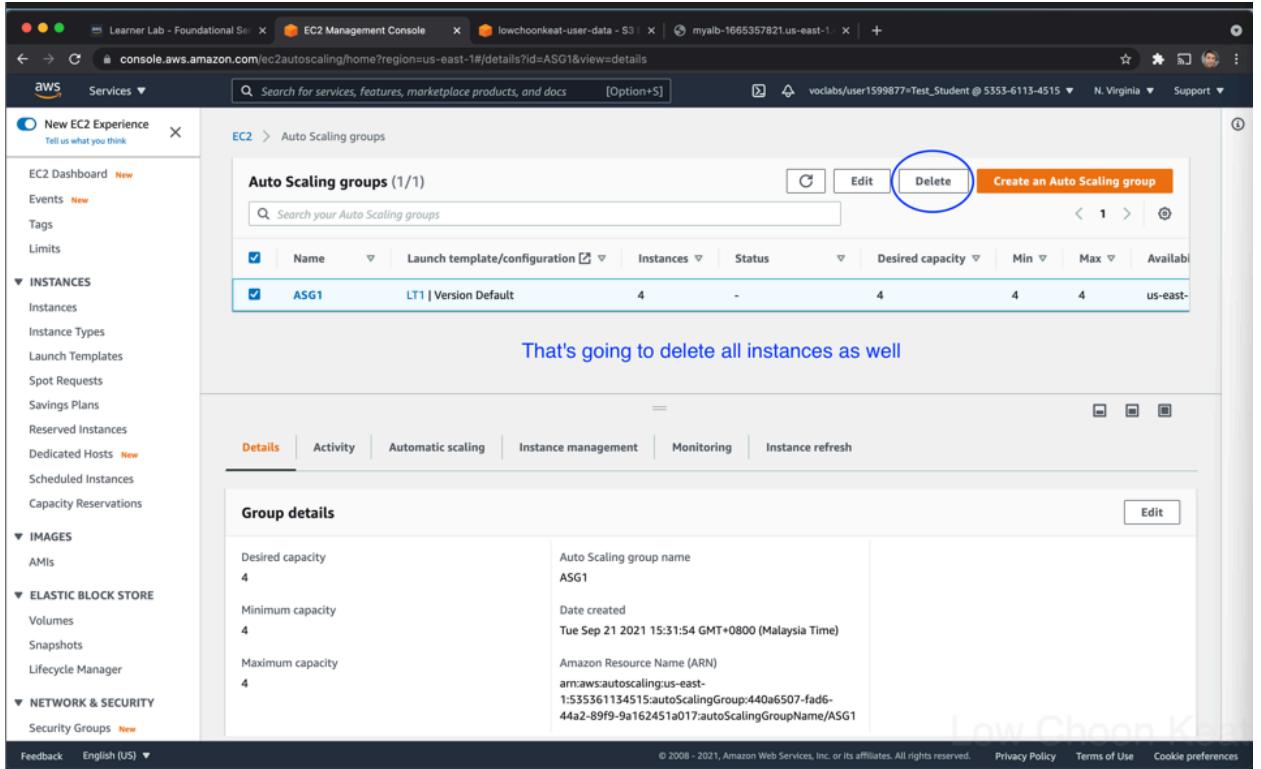
Name	ARN	DNS name	Type	Scheme	IP address type	VPC	Availability Zones
MyALB	arn:aws:elasticloadbalancing:us-east-1:535361134515:loadbalancer/app/MyALB/7858ab01c32675e7	MyALB-1665357821.us-east-1.elb.amazonaws.com (A Record)	application	internet-facing	ipv4	vpc-01ff136a17db0aca1	subnet-03add54b48e166ac - us-east-1c subnet-04d6a0702c2de6304 - us-east-1b subnet-0ad315974f475c22f - us-east-1d

47. Go to Load Balancing > Target Groups, delete "TG1"



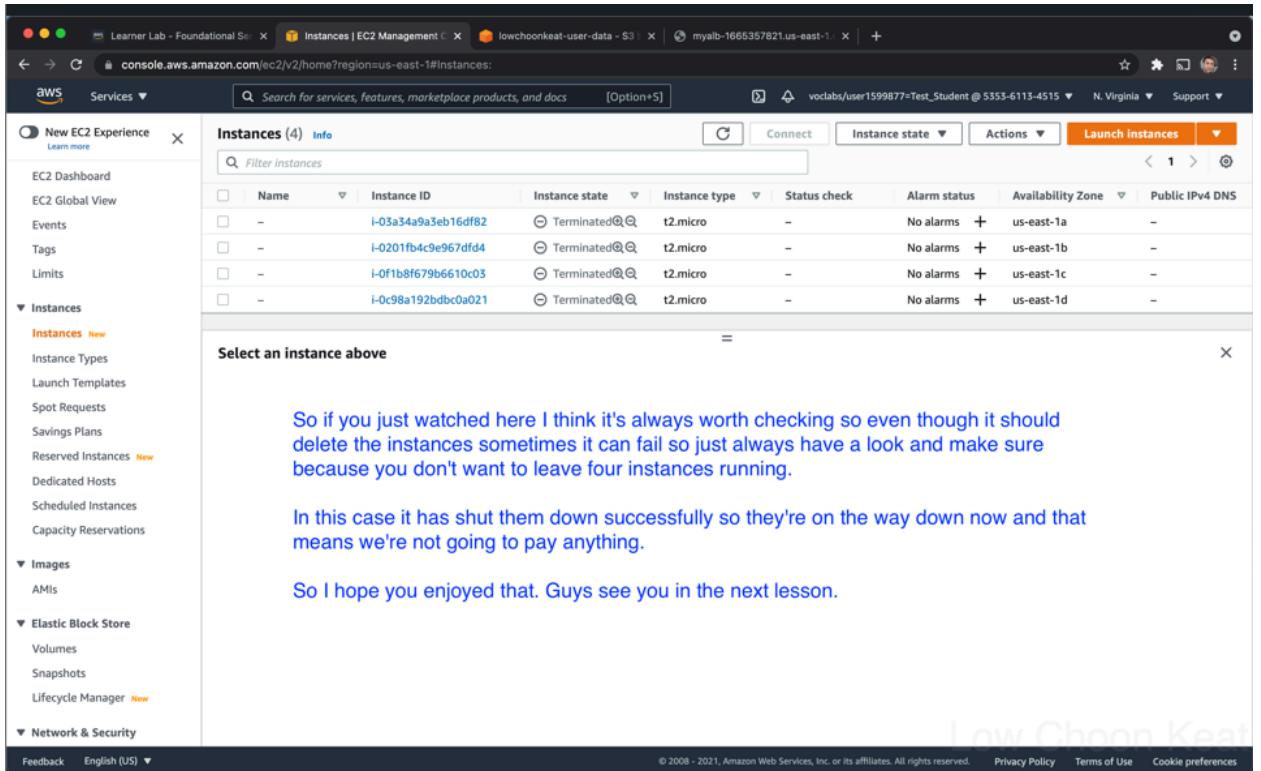
This screenshot is identical to the one above, showing the AWS EC2 Management Console with the 'Load Balancers' section selected. A context menu is open over the 'MyALB' load balancer, with the 'Delete' option highlighted. The main table displays the same details for the 'MyALB' load balancer as in the previous screenshot.

48. Go to EC2 > Auto Scaling groups, delete "ASG1"



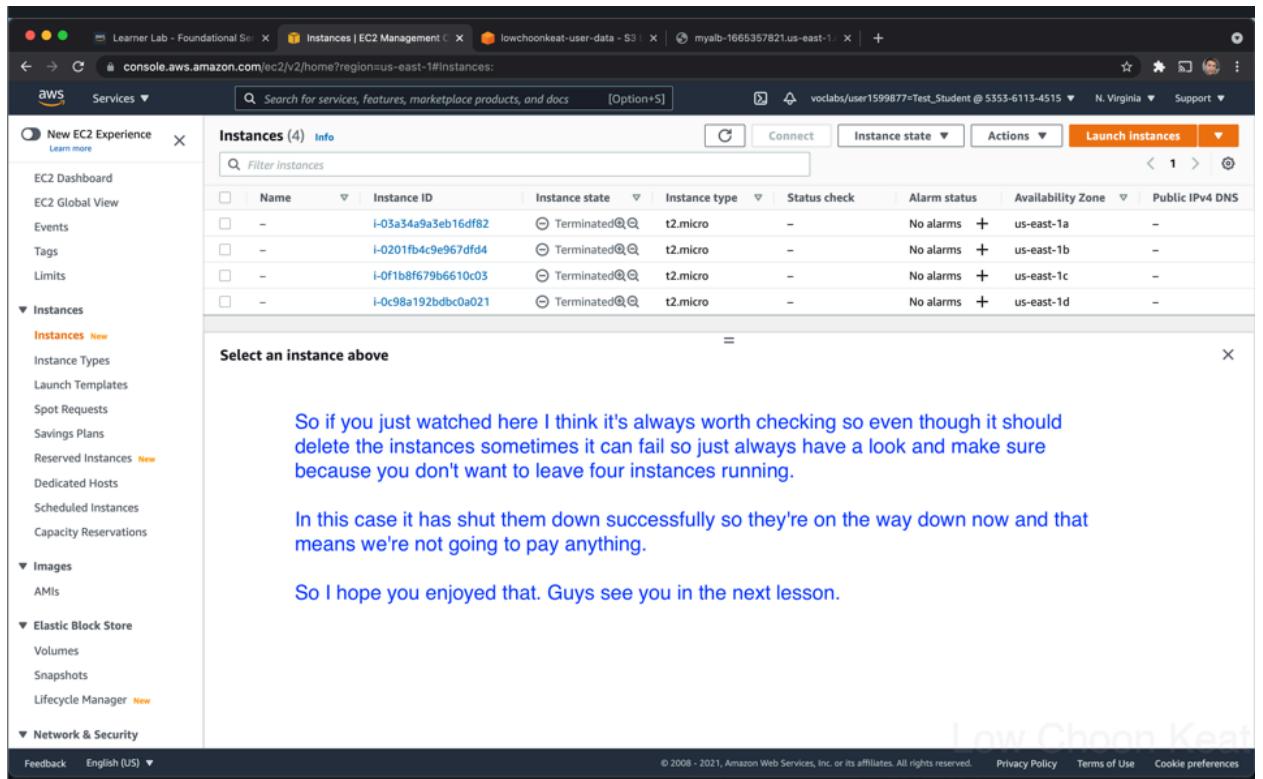
The screenshot shows the AWS EC2 Management Console with the 'Auto Scaling groups' page open. On the left, there's a sidebar with various EC2-related options like Instances, AMIs, and Network & Security. The main area displays a table for 'Auto Scaling groups (1/1)'. A single row is selected for 'ASG1', which has a status of 'LT1 | Version Default'. To the right of the table, a message says 'That's going to delete all instances as well'. Below the table, there are tabs for Details, Activity, Automatic scaling, Instance management, Monitoring, and Instance refresh. Under the 'Group details' section, it shows the desired capacity as 4, and the auto scaling group name as ASG1.

49. Go to Instances > Instances, delete all the instances



The screenshot shows the AWS EC2 Management Console with the 'Instances (4) Info' page open. The sidebar on the left includes 'Instances' under the EC2 category. The main table lists four instances, all of which are in a 'Terminated' state. A modal window titled 'Select an instance above' is open, containing a message: 'So if you just watched here I think it's always worth checking so even though it should delete the instances sometimes it can fail so just always have a look and make sure because you don't want to leave four instances running.' Below this, another message states: 'In this case it has shut them down successfully so they're on the way down now and that means we're not going to pay anything.' At the bottom of the modal, it says 'So I hope you enjoyed that. Guys see you in the next lesson.'

50. Go to Instances > Launch Templates, delete "LT1"



The screenshot shows the AWS EC2 Instances page with four terminated t2.micro instances listed:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
-	i-03a34a9a3eb16df82	Terminated	t2.micro	-	No alarms	+ us-east-1a	-
-	i-020fb4c9e67df4	Terminated	t2.micro	-	No alarms	+ us-east-1b	-
-	i-0f1b8f679b6610c03	Terminated	t2.micro	-	No alarms	+ us-east-1c	-
-	i-0c98a192bdb0a021	Terminated	t2.micro	-	No alarms	+ us-east-1d	-

A modal dialog box is open with the message: "Select an instance above". Inside the dialog, there is blue text providing instructions and a summary of the action taken:

So if you just watched here I think it's always worth checking so even though it should delete the instances sometimes it can fail so just always have a look and make sure because you don't want to leave four instances running.

In this case it has shut them down successfully so they're on the way down now and that means we're not going to pay anything.

So I hope you enjoyed that. Guys see you in the next lesson.

Practical 5

Create and Attach EBS Volume

EBS Snapshots and AMI

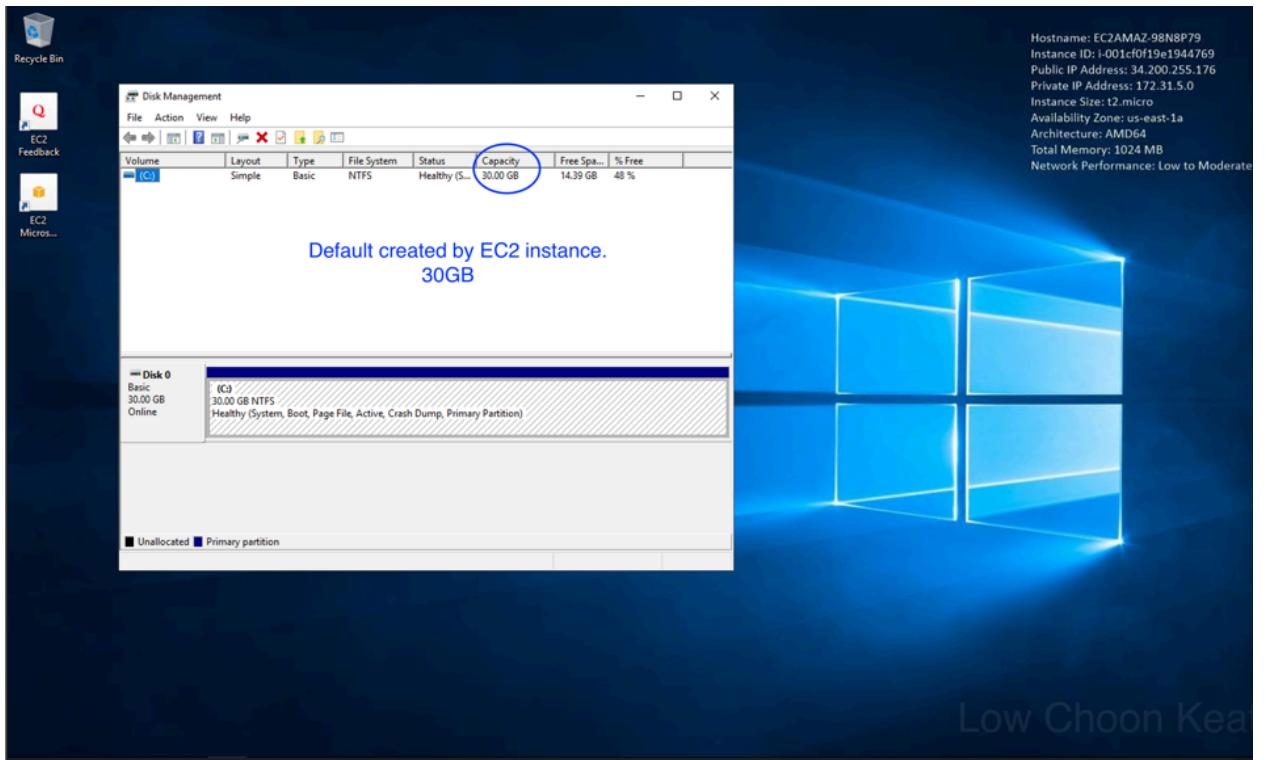
1. Go to [EC2](#)
2. Go to [Elastic Block Store > Volumes](#), click on "Create volume" button
3. Select "General Purpose SSD (gp2)" as **Volume type**
4. Go to Learner Lab, find the EBS volumes needed

The screenshot shows the AWS Academy Learner Lab interface. On the left is a sidebar with navigation links: Home, Modules (which is selected), and Discussions. The main area contains a terminal window with the command `ddd_v1_w_uBK3_884052@runweb37735:~$`. To the right of the terminal is a large text box containing a 'Readme' file. The 'Readme' file discusses launching instances and mentions EBS volumes as an example. A blue oval highlights the 'Readme' link in the top right corner of the text box. At the bottom of the screen, there is a purple footer bar with the text 'You are currently logged into Student View' and 'Resetting the test student will clear all history for this student, allowing you to view the course as a brand new student.' It also includes 'Reset Student' and 'Leave Student View' buttons.

5. Enter **10** as **Size (GiB)**
6. Select "us-east-1a" as **Availability Zone**
7. * You may enable encryption if you want
8. [Create Volume](#)
9. Go back to [Elastic Block Store > Volumes](#) page, we can see the volume status become available
10. Go to [Instances > Instances](#) page, click on "Launch instances" button
11. Enter name
12. Select "Windows" as Amazon Machine Image
13. Select "Microsoft Windows Server 2022 Base"

14. Select "t2.micro" as Instance type
15. Select "vokey" as Key pair (login)
16. Under Network settings, click on "Edit" button
17. Select the **subnet with Availability Zone "us-east-1a"** (make sure it is same with the previous selected Availability Zone during Volume creation)
18. Select "Web-Access" as Common security groups
19. Launch instance
20. Go to **Network & Security > Security Groups** page, select "**Web-Access**"
21. Under the **Inbound rules**, click on "**Edit inbound rules**" button
22. Click on "**Add rule**"
23. Select "**RDP**" as **type** (port 3389), "**Anywhere as IPv4**" as **Source** and **0.0.0.0/0**
24. Save rules
25. Go back to **Instances > Instances** page, select the instance
26. Click on **Connect** button
27. Under **RDP client tab**, click on "**Get password**" button
28. Click on "**Browse**" button
29. Select and insert the "**vokey.pem**" file
30. **Decrypt Password**
31. Copy the **Public DNS**
32. In Windows, open the "**Remote Desktop Connection**"
33. Paste the **Public DNS** inside the Computer
34. Click on "**Connect**" button

35. Inside the remote desktop, search for "Disk Management"



36. Go back to Elastic Block Store > Volumes page

Search for services, features, marketplace products, and docs [Option+S]

Services ▾

- Instances
- Images
- Elastic Block Store
- Network & Security
- Load Balancing

Volumes

Name	Volume ID	Size	Volume Type	IOPS	Throughput	Snapshot	Created	Availability Zone	State	Alarm Status
vol-047d04d...	30 GiB	gp2	100	-	-	snap-08ff4441...	September 22, 202...	us-east-1a	in-use	None
vol-0fc6e878...	10 GiB	gp2	100	-	-		September 22, 202...	us-east-1a	available	None

Volumes: vol-0fc6e878cb5840d67

Description Status Checks Monitoring Tags

Volume ID: vol-0fc6e878cb5840d67	Outposts ARN: -
Alarm status: None	Size: 10 GiB
Snapshot: -	Created: -
Availability Zone: us-east-1a	State: available
Encryption: Not Encrypted	Attachment information:
KMS Key ID: -	Volume type: gp2
KMS Key Aliases: -	Product codes: -
KMS Key ARN: -	IOPS: 100
Throughput (MB/s): -	Multi-Attach Enabled: No

Feedback English (US) ▾ © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

37. Select the Volume, attach the Volume

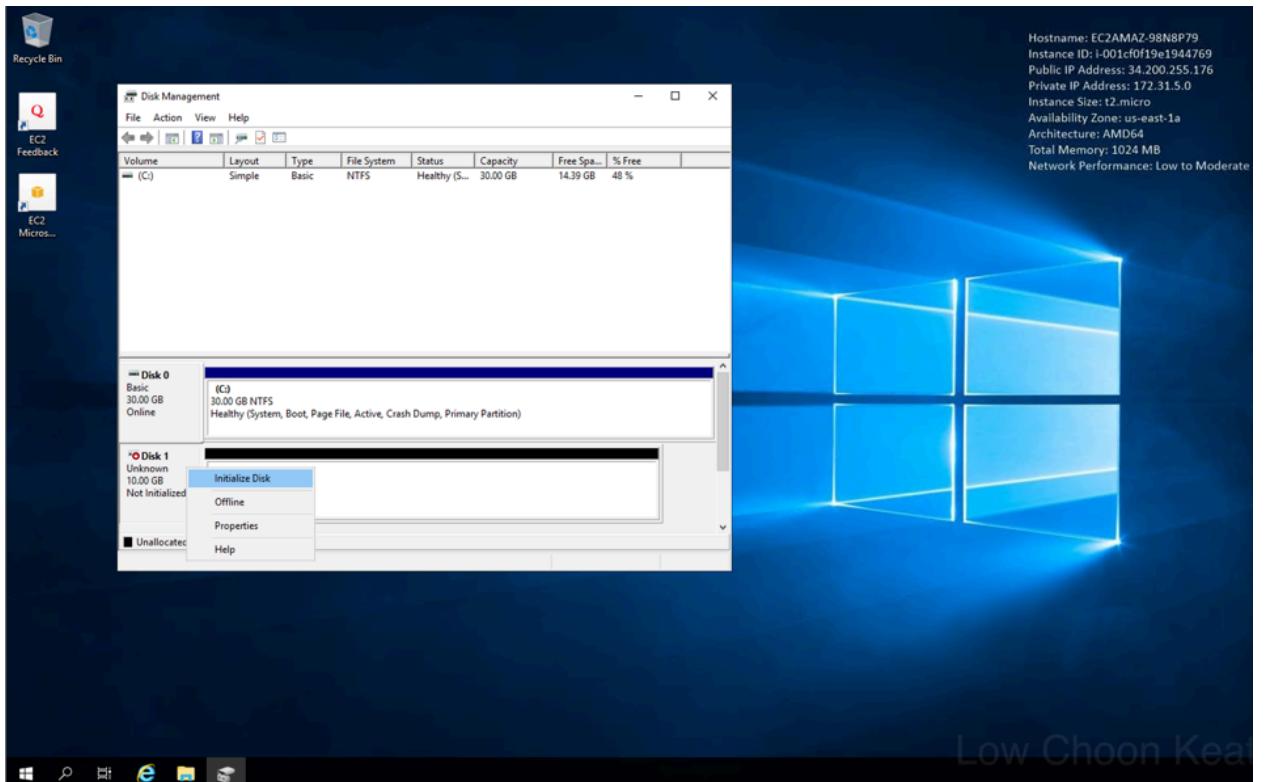
The screenshot shows the AWS EC2 Volumes management console. On the left, there's a sidebar with various navigation options like Instances, Images, and Network & Security. The main area displays a list of volumes. A context menu is open over the second volume in the list, showing options such as Modify Volume, Create Snapshot, Create Snapshot Lifecycle Policy, Delete Volume, Attach Volume, Detach Volume, Force Detach Volume, Change Auto-Enable IO Setting, and Add/Edit Tags. Below the list, a detailed view of a selected volume (vol-0fc6e878cb5840d67) is shown, including its Volume ID, Size (10 GiB), Volume Type (gp2), IOPS (100), Throughput, Snapshot, Created date (September 22, 2021), Availability Zone (us-east-1a), State (available), and Alarm Status (None). The volume is also associated with a snapshot (snap-08ff4441...), an Outposts ARN, and other metadata.

38. Select the Instance

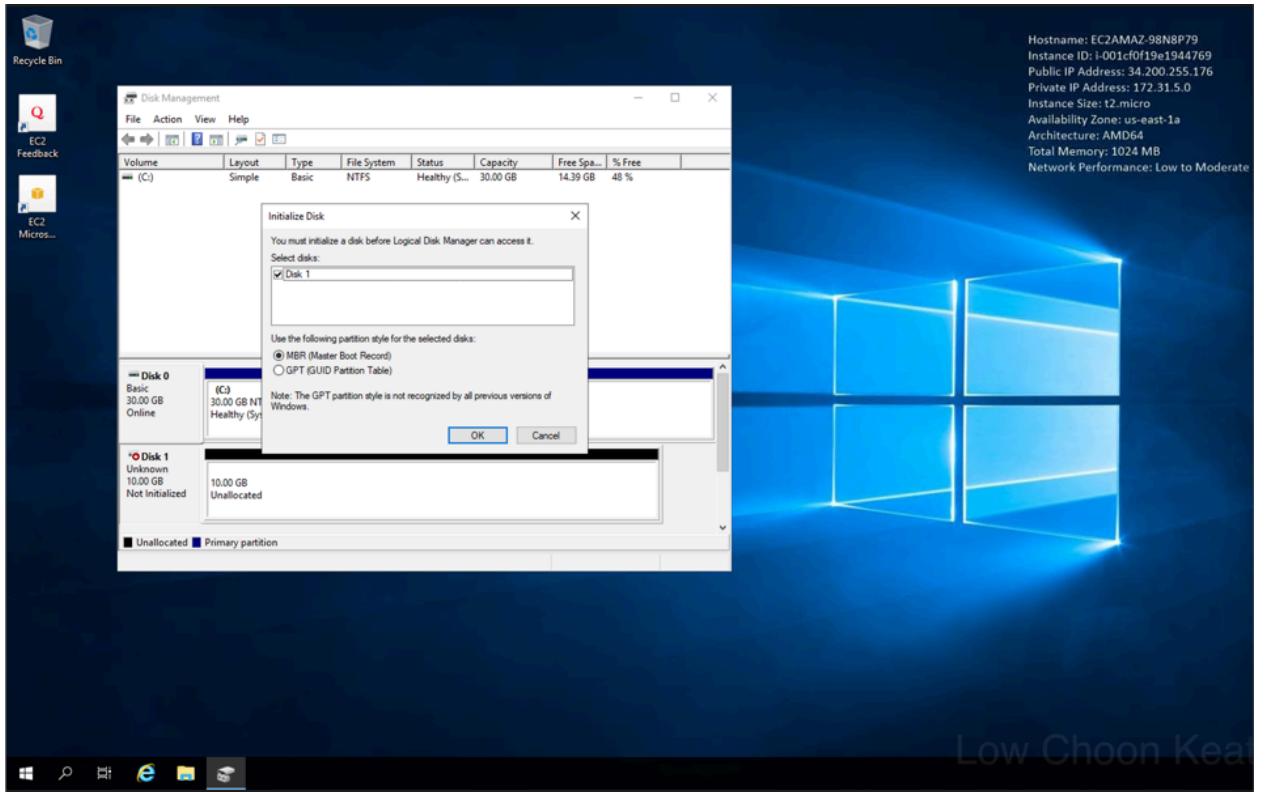
This screenshot continues from the previous one, showing the 'Attach Volume' dialog box. The 'Volume' field is set to 'vol-0fc6e878cb5840d67 in us-east-1a'. The 'Instance' field contains a search bar with the placeholder 'Search instance ID or Name tag' and a dropdown menu showing 'i-001cf0f19e1944769 (running)'. The 'Device' field is set to '/dev/sdh'. At the bottom right of the dialog are 'Cancel' and 'Attach' buttons. The background shows the same volume list and detailed view as the previous screenshot.

39. Attach

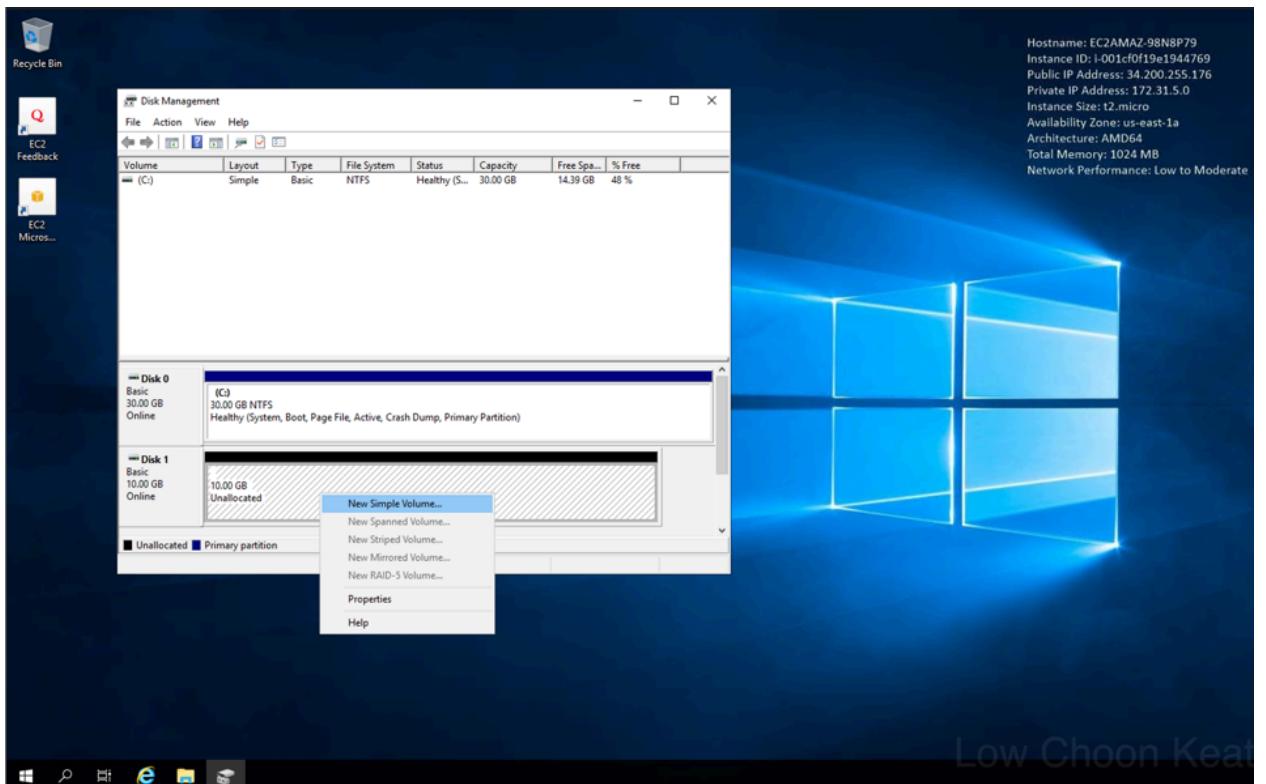
40. Go back to remote desktop, right click on the below 10.00 GB Disk, click on Initialize Disk



41. Tick the "Disk 1", select MBR (Master Boot Record) and click on "OK" button

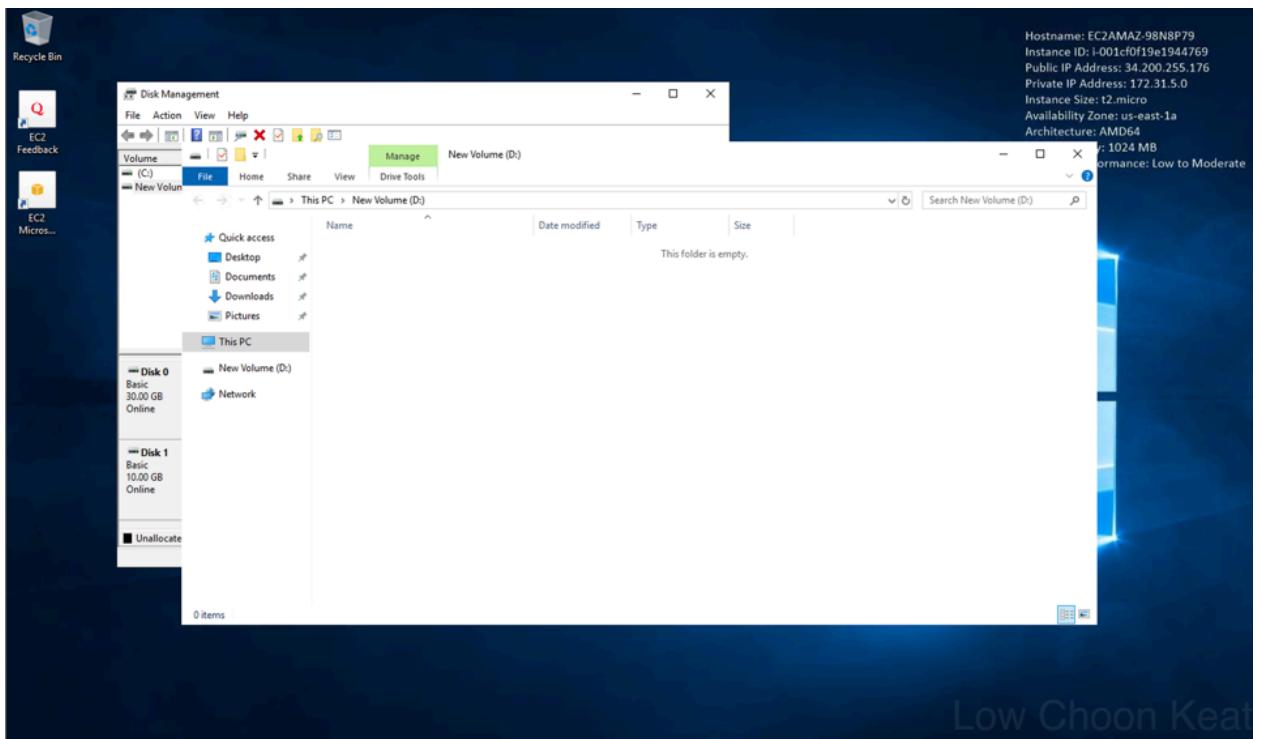
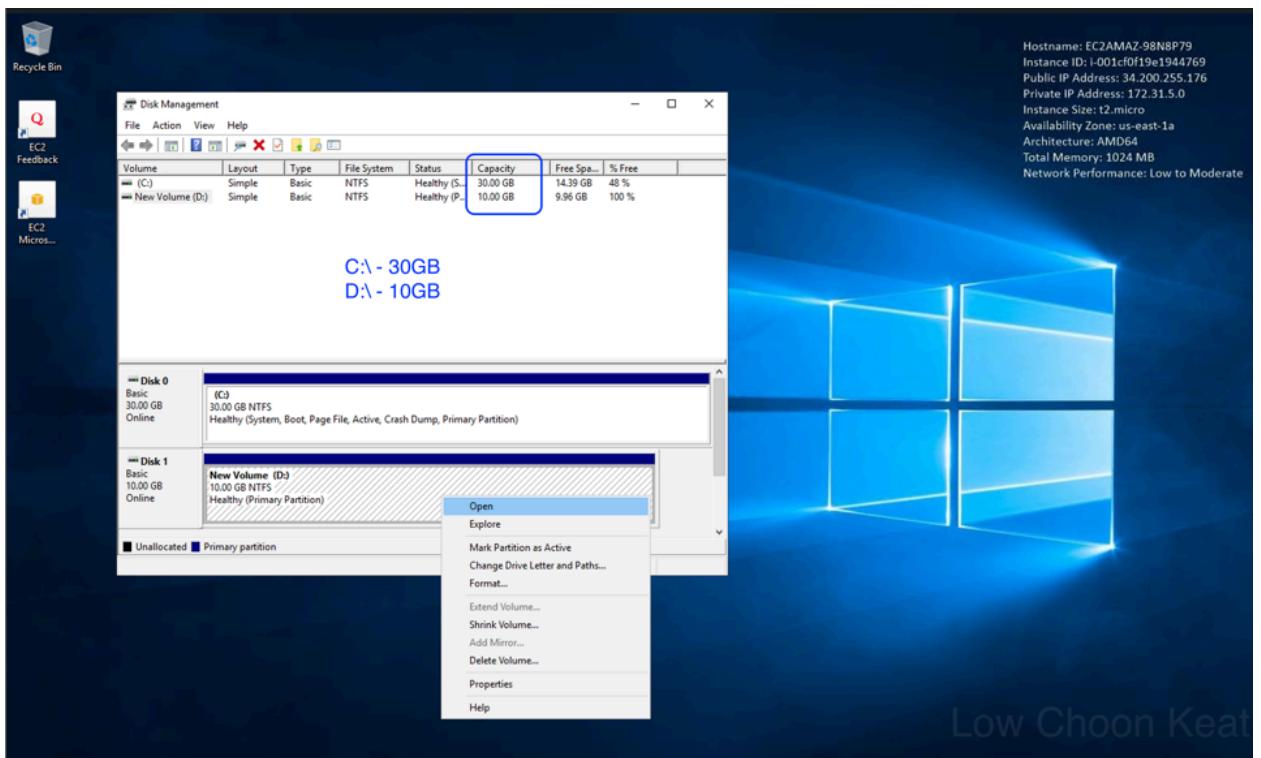


42. Right click on Disk 1, click on "New Simple Volume..." option

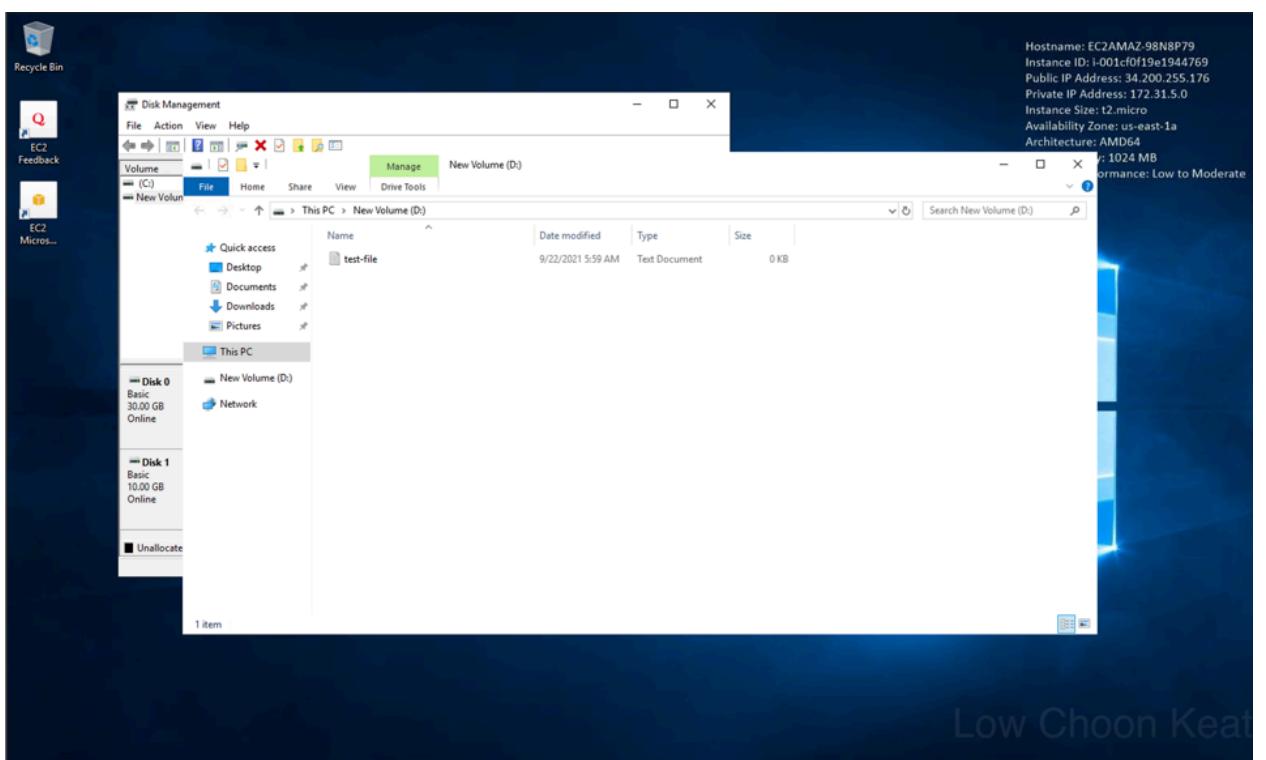
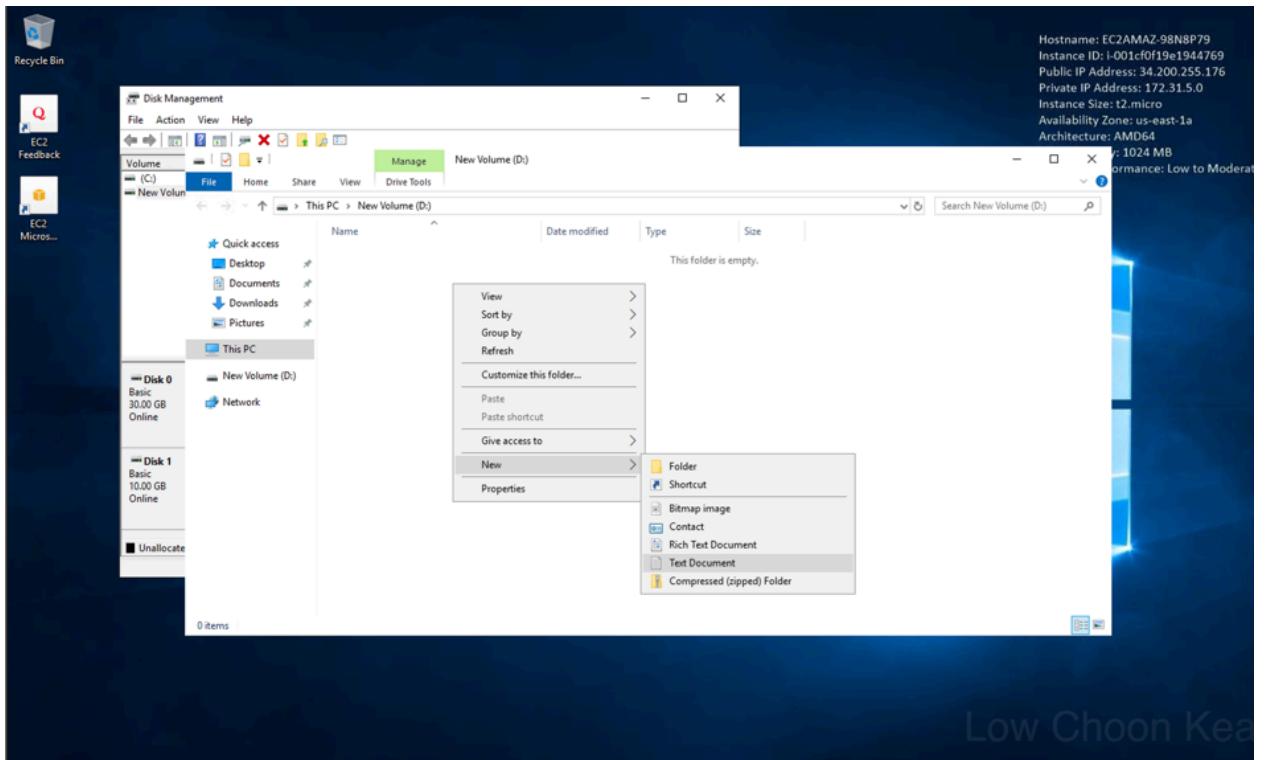


43. Click on Next
44. Enter 10237 as the Simple volume size in MB, click on Next
45. Click on Next
46. Click on Next
47. Click on Finish

48. Right click on the Disk 1, click on "Open"



49. Create a new document inside



50. Go back to [Elastic Block Store > Volumes](#) page, select the Volume, detach the volume

Learner Lab - Foundational Services | Volumes | EC2 Management Console

console.aws.amazon.com/ec2/v2/home?region=us-east-1#Volumes.sort=desc:createTime

Services ▾

Search for services, features, marketplace products, and docs [Option+S]

vocabls/user1599877=Test_Student @ 5353-6113-4515 N. Virginia Support ▾

Limits

Instances Instances New

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances New

Dedicated Hosts

Scheduled Instances

Capacity Reservations

Images AMIs

Elastic Block Store Volumes Snapshots Lifecycle Manager New

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

Load Balancing

Create Volume Actions ▾

Modify Volume Create Snapshot Create Snapshot Lifecycle Policy Delete Volume Attach Volume Detach Volume Force Detach Volume Change Auto-Enable IO Setting Add/Edit Tags

Volume Type IOPS Throughput Snapshot Created Availability Zone State Alarm Status

gp2 100 - snap-08ff4441... September 22, 202... us-east-1a in-use None

gp2 100 - September 22, 202... us-east-1a in-use None

Volumes: vol-0fc6e878cb5840d67

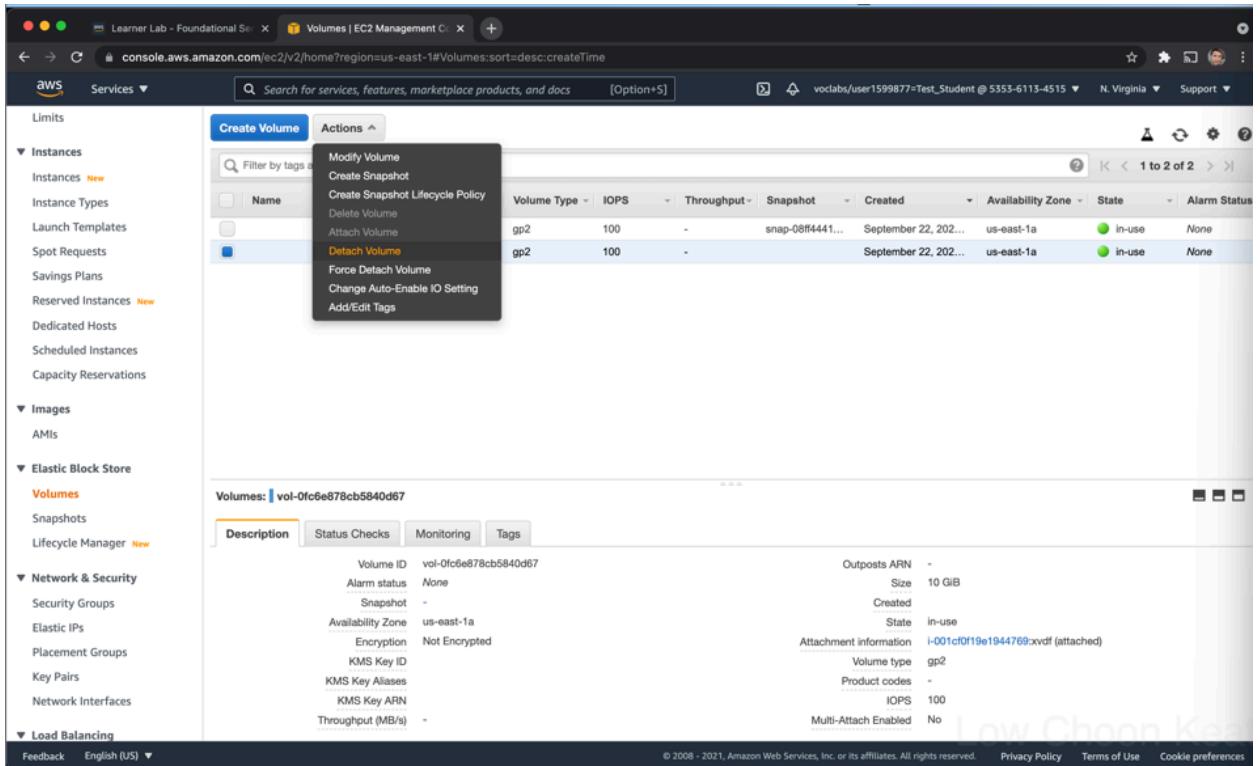
Description Status Checks Monitoring Tags

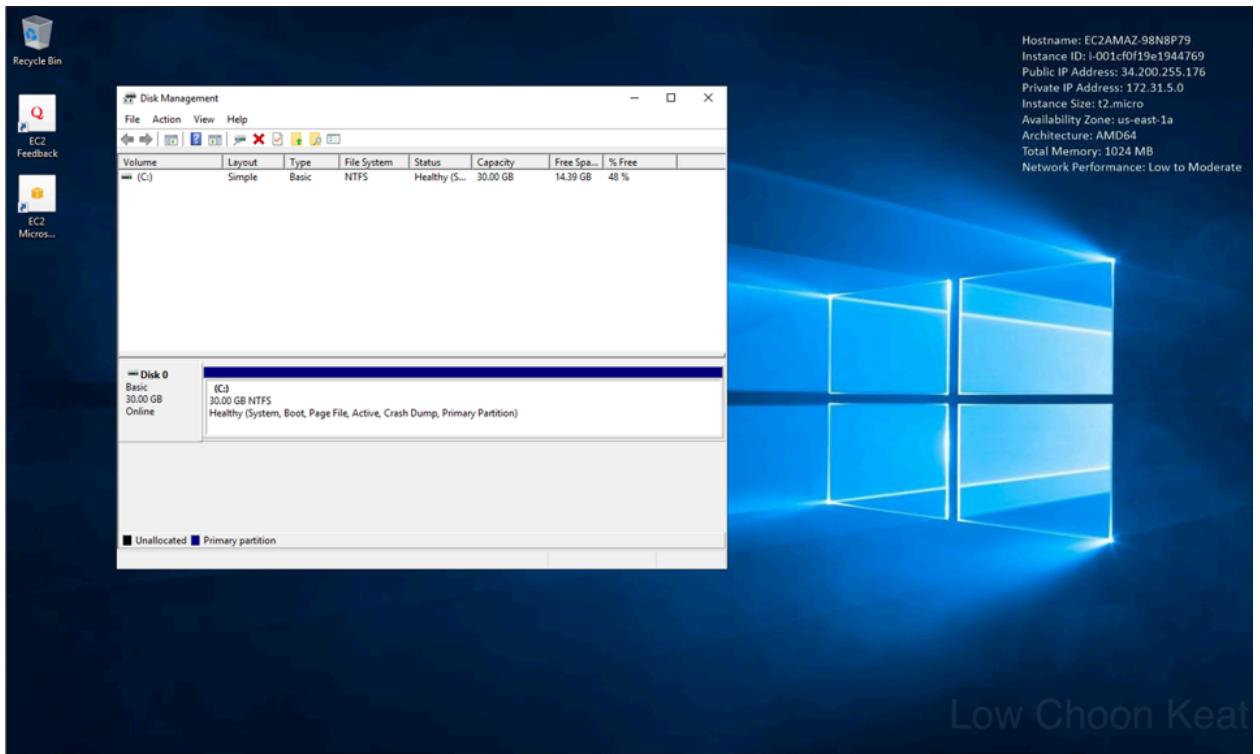
	Volume ID	vol-0fc6e878cb5840d67	Outposts ARN	-
Alarm status	None	Size	10 GiB	
Snapshot	-	Created		
Availability Zone	us-east-1a	State	in-use	
Encryption	Not Encrypted	Attachment information	i-001cf0f19e1944769:xvdf (attached)	
KMS Key ID		Volume type	gp2	
KMS Key Aliases		Product codes	-	
KMS Key ARN		IOPS	100	
Throughput (MB/s)	-	Multi-Attach Enabled	No	

Feedback English (US) ▾

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

Low Choon Keat





Low Choon Keat

Name	Volume ID	Size	Volume Type	IOPS	Throughput	Snapshot	Created	Availability Zone	State	Alarm Status
vol-0f47dc4d...	30 GiB	gp2	100	-	-	snap-08ff4441...	September 22, 202...	us-east-1a	in-use	None
vol-0fc6e878...	10 GiB	gp2	100	-	-	snap-08ff4441...	September 22, 202...	us-east-1a	available	None

51. Go to **Instances > Instances** page, terminate the instance

The screenshot shows the AWS EC2 Management Console interface. On the left, the navigation pane is visible with sections like Services, Instances, Images, Elastic Block Store, Network & Security, and Load Balancing. The Instances section is currently selected. The main area displays a table titled "Instances (1/1) Info" with one row. The row contains the following information:

Name	Instance ID	Instance state	Instance type	Status
-	i-001cf0f19e1944769	Running	t2.micro	2/2 ch

To the right of the table, there are buttons for "Stop instance", "Start instance", "Reboot instance", "Hibernate instance", and "Terminate instance". Below the table, a modal window titled "Instance: i-001cf0f19e1944769" is open, showing the "Details" tab. The "Instance summary" section contains the following data:

Instance ID	Public IPv4 address	Private IPv4 addresses
i-001cf0f19e1944769	34.200.255.176 open address	172.31.5.0
IPv6 address	Instance state	Public IPv4 DNS
-	Running	ec2-34-200-255-176.compute-1.amazonaws.com open address
Private IPv4 DNS	Instance type	Elastic IP addresses
ip-172-31-5-0.ec2.internal	t2.micro	-

At the bottom of the modal, there are links for "Feedback", "English (US)", and "Cookie preferences".

Learner Lab - Foundational Series | EC2 Management Console

console.aws.amazon.com/ec2/v2/home?region=us-east-1#Instances:

Services ▾

Search for services, features, marketplace products, and docs [Option+S]

voclabs/user1599877=Test_Student @ 5353-6113-4515 ▾ N. Virginia ▾ Support ▾

Instances (1/1) info

Filter instances

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
-	i-001cf0f19e1944769	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-34-200-255-11

Terminate instance?

On an EBS-backed instance, the default action is for the root EBS volume to be deleted when the instance is terminated. Storage on any local drives will be lost.

Termination EC2 will delete the 30GB EBS as well

Are you sure you want to terminate these instances?
i-001cf0f19e1944769

To confirm that you want to terminate the instances, choose the terminate button below. Terminating the instance cannot be undone.

Cancel Terminate

Details Security Networking

Instance ID: i-001cf0f19e1944769

IPv6 address: - Instance state: Running

Private IPv4 DNS: ip-172-31-5-0.ec2.internal Instance type: t2.micro

Public IPv4 DNS: ec2-34-200-255-176.compute-1.amazonaws.com | open address

Elastic IP addresses: 172.31.5.0

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

Learner Lab - Foundational Series | Volumes | EC2 Management Console

console.aws.amazon.com/ec2/v2/home?region=us-east-1#Volumes:sort=desc:createTime

Services ▾

Search for services, features, marketplace products, and docs [Option+S]

voclabs/user1599877=Test_Student @ 5353-6113-4515 ▾ N. Virginia ▾ Support ▾

Create Volume Actions

Filter by tags and attributes or search by keyword

Name	Volume ID	Size	Volume Type	IOPS	Throughput	Snapshot	Created	Availability Zone	State	Alarm Status
vol-0fc6e878...	10 GiB	gp2	100	-	-	-	September 22, 2022	us-east-1a	available	None

The EC2 termination will delete the 30GB EBS as well.
Left 10GB.
Leave it. We need it to attach to another EC2 later.

Volumes

Volumes: vol-0fc6e878cb5840d67

Description Status Checks Monitoring Tags

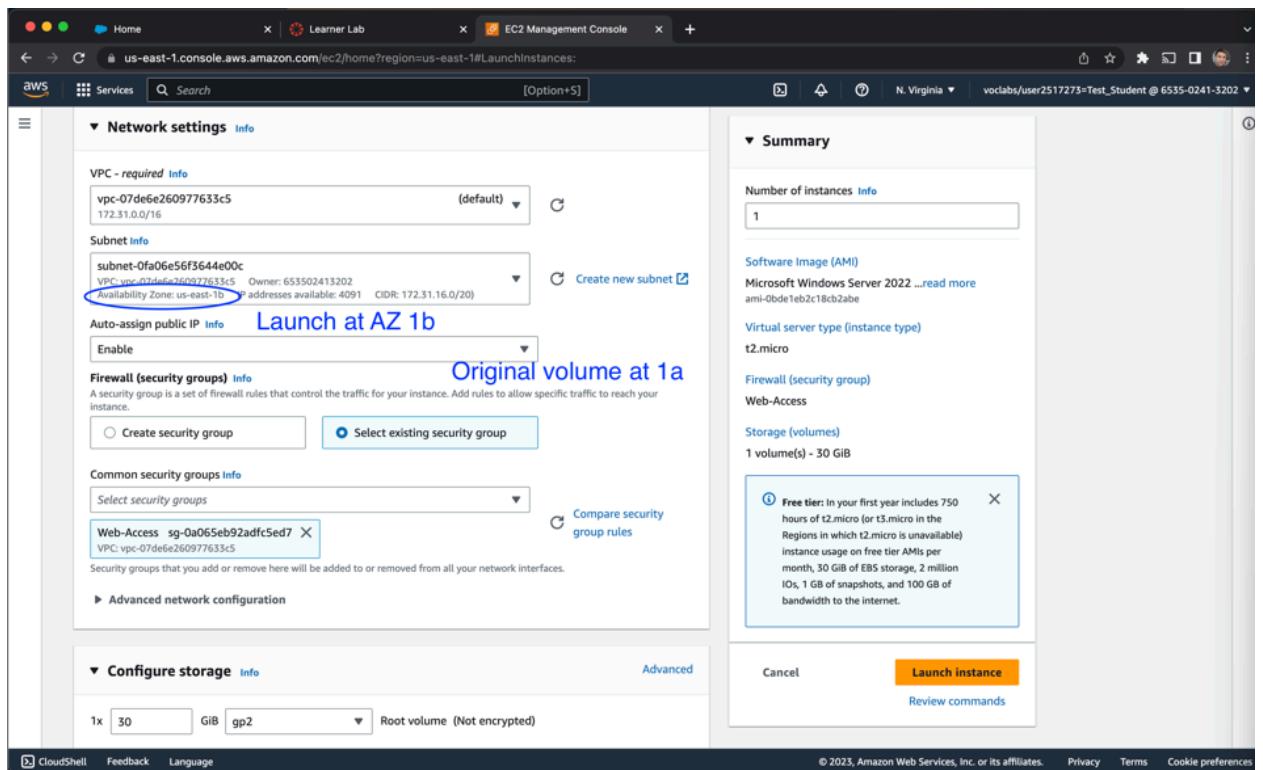
Volume ID: vol-0fc6e878cb5840d67
Alarm status: None
Outposts ARN: -
Size: 10 GiB

Feedback English (US) ▾

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

EBS Snapshots and AMI's

1. Go to **Elastic Block Store > Volumes** page, select the Volume (10 GiB)
2. Click on "Actions" button, select "Create Snapshot"
3. Enter "**MyTestDataVolume**" as the Description
4. **Create Snapshot**
5. Close
6. You may see the Snapshot status is "Completed" in Elastic Block Store > Snapshots page
7. Go to **Instances > Instances** page, click on "Launch Instances"
8. Select "Windows" as Amazon Machine Images (AMI)
9. Select "Microsoft Windows Server 2022 Base" (Launch Windows server)
10. Select "t2.micro" as Instance type
11. Under **Network settings**, select **subnet** with Availability Zone "us-east-1b"



12. Select "Web-Access" as Common security groups

13. Launch instance

The screenshot shows the AWS EC2 Management console. In the left sidebar, under 'Instances', 'Instances' is selected. The main area displays a table of instances. One instance is highlighted: Name i-0a6e281d3873ceaaa, Instance ID i-0a6e281d3873ceaaa, Instance state Running, Instance type t2.micro, Status check Initializing, Alarm status No alarms, Availability Zone us-east-1b, Public IPv4 DNS ec2-3-94-171-152. A blue arrow points from the text below to the 'us-east-1b' label in the table.

And there we go. So now we have a EC2 instance in a different availability zone from where we created our volume before. So let's go back and have a look at our snapshot.

14. Go to Elastic Block Store > Volumes

The screenshot shows the AWS EC2 Management console. In the left sidebar, under 'Elastic Block Store', 'Volumes' is selected. The main area displays a table of volumes. Two volumes are listed: vol-072269cf... (30 GB, gp2, 100 IOPS, Throughput -, Snapshot snap-08ff4441..., Created September 22, 202..., Availability Zone us-east-1b, State in-use, Alarm Status None) and vol-0fb6e678... (10 GB, gp2, 100 IOPS, Throughput -, Snapshot -, Created September 22, 202..., Availability Zone us-east-1a, State available, Alarm Status None). A blue circle highlights the '30 GB' size value for the first volume.

Before that, we can go to Volumes. You can see we have new 30GB volume. That is our new EC2 instance C:\ drive.

15. Go to Elastic Block Store > Snapshots

the snapshots completed

Note: The snapshot did not under any availability zones. It is at public service.

Name	Snapshot ID	Size	Description	Status	Started	Progress
	snap-08f1ebbebc20dc40c	10 GiB	MyTestDataVolume	completed	September 22, 2021	available (100%)

Snapshot: snap-08f1ebbebc20dc40c

Description Permissions Tags

Snapshot ID	snap-08f1ebbebc20dc40c	Progress	100%
Status	completed	Capacity	10 GiB
Volume	vol-0fcfe8e78cb5840d67	Encryption	Not Encrypted
Started	September 22, 2021 at 8:17:41 PM UTC+8	KMS Key ID	
Owner	535361134515	KMS Key Aliases	
Product codes	Loading...	KMS Key ARN	
Description	MyTestDataVolume	Fast Snapshot Restore	-
Outpost ARN	-		

16. Click on "Actions", select "Create Volume"

17. Select "General Purpose SSD (gp2)" as Volume Type

18. Enter 10 as Size (GiB)

19. Select "us-east-1b" as Availability Zone (AZ same with previous selected availability zone of the new instance)

20.* You may add encryption if you wanted

21. Create Volume

22. Close

23. In **Elastic Block Store > Volumes**, you can see the new volume created

And so what I want to do is attach that volume to my new EC2 to instance. So let's come back, we can find our volume, it's one of the ones that's available. And in this case, it's going to B so we know that this is the volume we just created.

Name	Volume ID	Size	Type	IOPS	Throughput	Snapshot	Created	Availability Zone	State	Alarm Status
vol-08265bb...	10 GiB	gp2	100	-	-	snap-08f1eb...	September 22, 202...	us-east-1b	D E W	available
vol-072269cf...	30 GiB	gp2	100	-	-	snap-08ff4441...	September 22, 202...	us-east-1b	in-use	None
vol-0fc6e878...	10 GiB	gp2	100	-	-		September 22, 202...	us-east-1a	available	None

24. Select the new created volume, click on "Actions" button, select "Attach Volume"

25. Select the Instance

we've only got one running instance, let's click on attach.

Cancel **Attach**

Name	Volume ID	Size	Type	IOPS	Throughput	Snapshot	Created	Availability Zone	State	Alarm Status
vol-08265bb...	10 GiB	gp2	100	-	-	snap-08f1eb...	September 22, 202...	us-east-1b	available	None
vol-072269cf...	30 GiB	gp2	100	-	-	snap-08ff4441...	September 22, 202...	us-east-1b	in-use	None
vol-0fc6e878...	10 GiB	gp2	100	-	-		September 22, 202...	us-east-1a	available	None

26. Attach

27. You can see the newly created volume state is in "In-use"

The screenshot shows the AWS EC2 Volumes Management console. On the left, there's a navigation sidebar with links like 'EC2 Dashboard', 'EC2 Global View', 'Events', 'Tags', 'Limits', 'Instances', 'Images', 'Elastic Block Store' (with 'Volumes' selected), and 'Network & Security'. The main area has tabs for 'Create Volume' and 'Actions'. A table lists volumes with columns for Name, Volume ID, Size, Volume Type, IOPS, Throughput, Snapshot, Created, Availability Zone, State, and Alarm Status. One row is highlighted with a blue border, and its 'State' column shows 'in-use' with a green circle icon. Below the table, a message says 'That should be now attached to our instance.' Another message below it says 'So once our instance is ready, we're going to log on and just check that the volume is there.' A third message says 'And that we can see that data that we created originally, I've given a couple of minutes.' At the bottom, there's a detailed view of a specific volume (vol-08265bb49fdad062a) with tabs for Description, Status Checks, Monitoring, and Tags. The 'Description' tab is active, showing details like Volume ID, Alarm status, Outposts ARN, and Size.

Name	Volume ID	Size	Volume Type	IOPS	Throughput	Snapshot	Created	Availability Zone	State	Alarm Status
vol-08265bb49fdad062a	vol-08265bb49fdad062a	10 GiB	gp2	100	-	snap-08f1febee...	September 22, 202...	us-east-1b	in-use	None
vol-072269cf...	vol-072269cf...	30 GiB	gp2	100	-	snap-08ff4444...	September 22, 202...	us-east-1b	in-use	None
vol-0fc6e878...	vol-0fc6e878...	10 GiB	gp2	100	-		September 22, 202...	us-east-1a	available	None

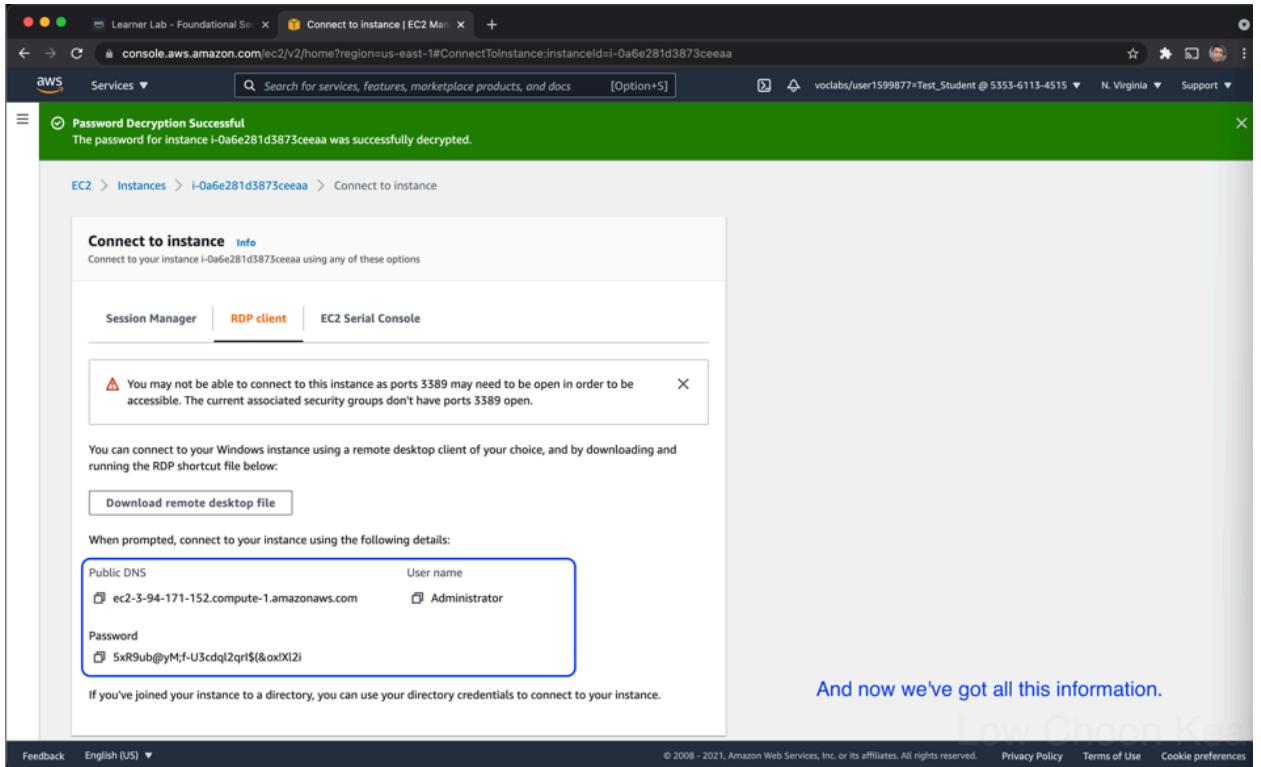
28. Go to Instances > Instances page, select the instance, click on "Connect" button

29. Under **RDP Client** tab, click on "Get password" button

30. Select and insert "vockey.pem" file

31. Decrypt Password

32. Copy the Username and Password



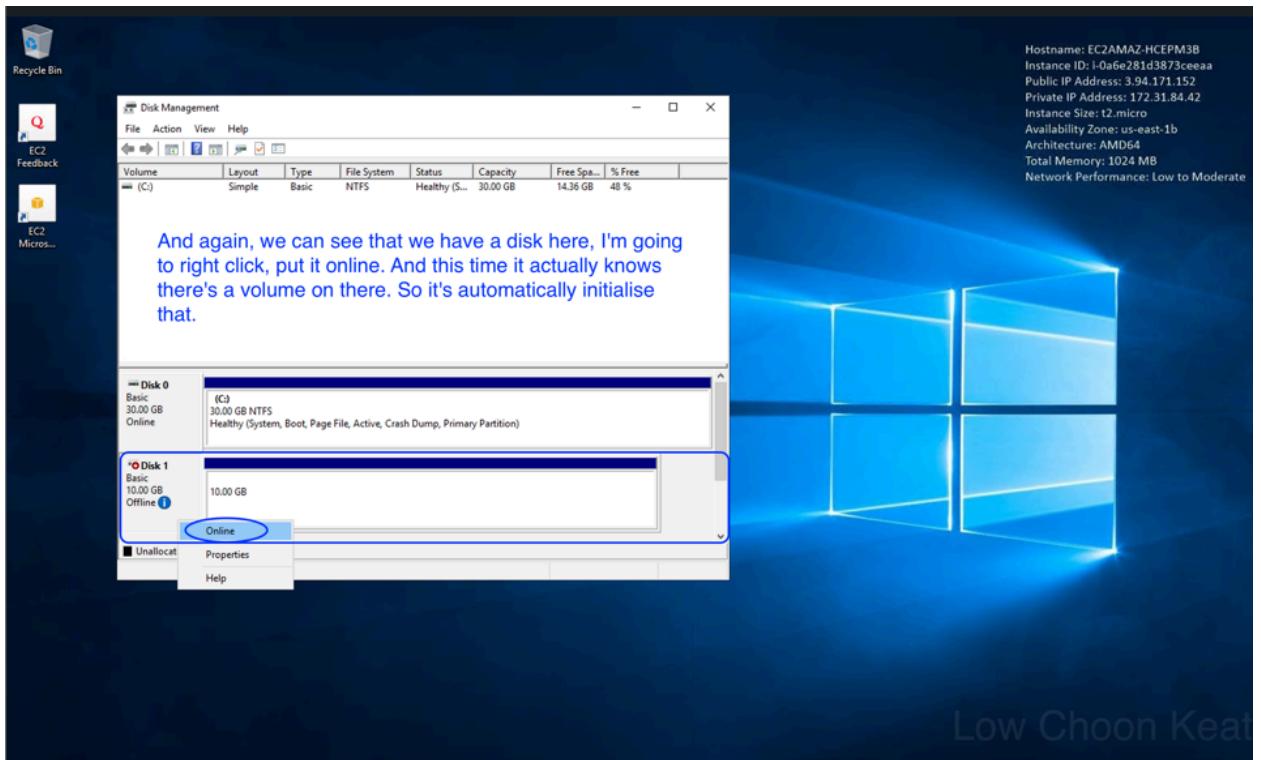
The screenshot shows the AWS EC2 Connect to instance page. At the top, a green banner indicates "Password Decryption Successful" with the message "The password for instance i-0a6e281d3873ceaa was successfully decrypted." Below the banner, the navigation path is EC2 > Instances > i-0a6e281d3873ceaa > Connect to instance. The main section is titled "Connect to instance" with an "Info" link. It provides three options: Session Manager, RDP client (selected), and EC2 Serial Console. A warning message states: "⚠ You may not be able to connect to this instance as ports 3389 may need to be open in order to be accessible. The current associated security groups don't have ports 3389 open." Below the warning, it says "You can connect to your Windows instance using a remote desktop client of your choice, and by downloading and running the RDP shortcut file below:" with a "Download remote desktop file" button. It also provides connection details: Public DNS (ec2-3-94-171-152.compute-1.amazonaws.com) and User name (Administrator). The password field is highlighted with a blue border and contains the value: SxR9ub@yM;f-U3cdql2qrI\${&oxlXl2}. To the right of the screenshot, the text "And now we've got all this information." is displayed.

33. Open the **Remote Desktop Connection**, paste the **Username and Password** inside

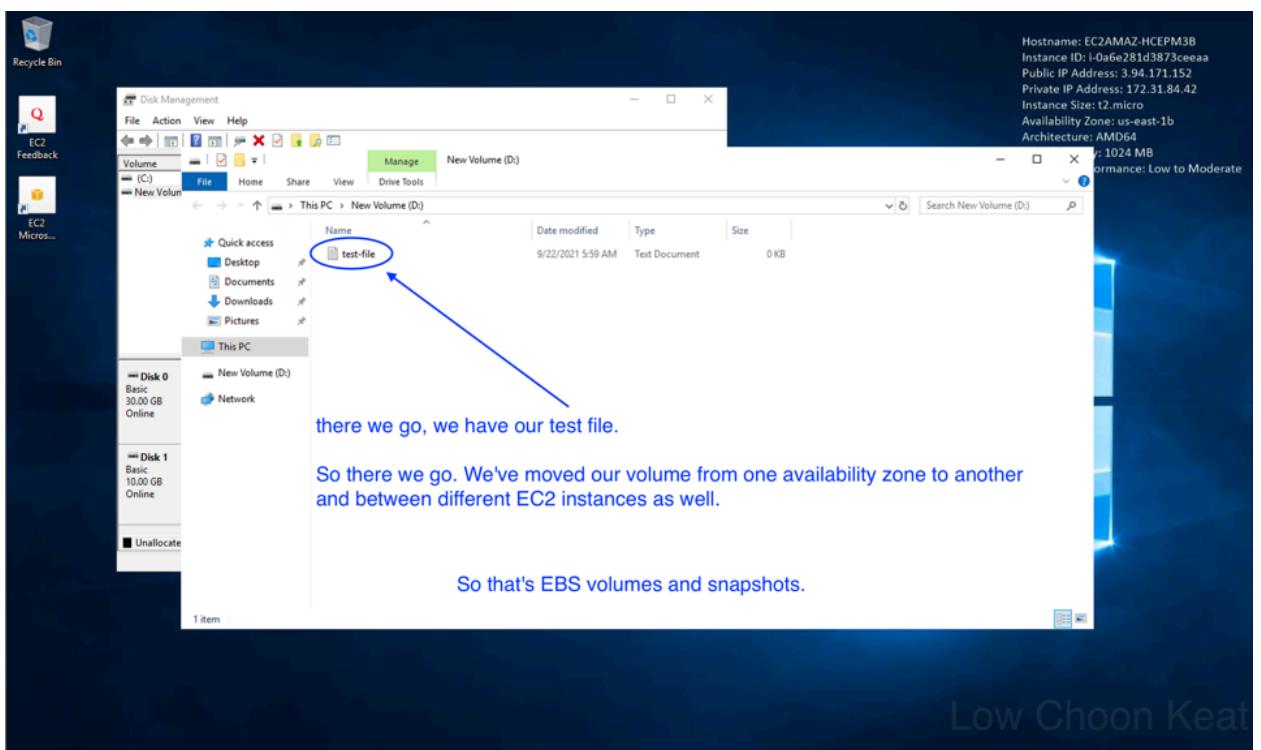
34. Connect

35. In the **Remote Desktop**, open "**Disk Management**"

36. Right click on the "Disk 1", select "Online"



37. Right click on "Disk 1", select "Open"



38. Go to **Elastic Block Store > Volumes** page, select the **Volume (30 GiB, us-east-1b)**, **Create Snapshot**

We come back to volumes.
Let's choose this volume here which I know is the root volume of my EC2 instance,
we'll create a snapshot of this.

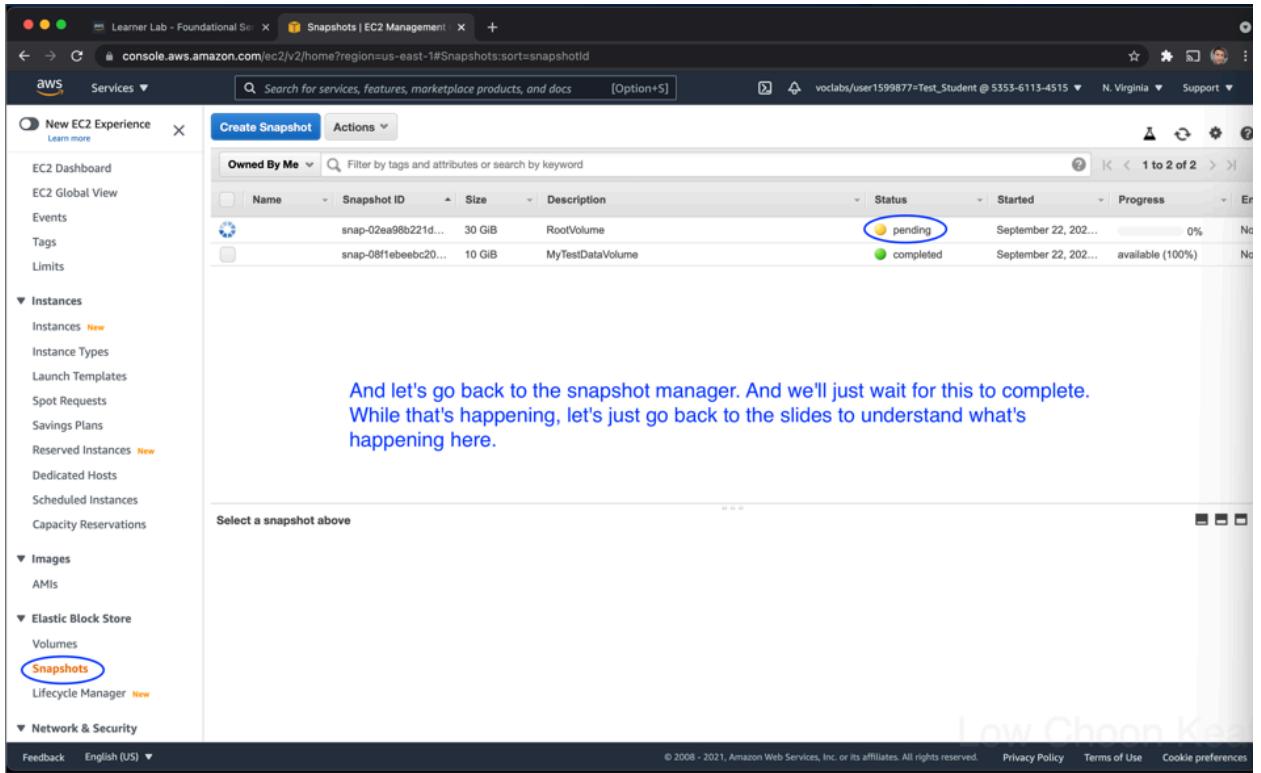
Name	Volume ID	Size	Type	IOPS	Throughput	Snapshot	Created	Availability Zone	State	Alarm Status
vol-08265bb...	10 GiB	gp2	100	-	-	snap-08f1ebee...	September 22, 202...	us-east-1b	in-use	None
vol-072269cf3b2fc2736	30 GiB	gp2	100	-	-	snap-08ff4441...	September 22, 202...	us-east-1b	in-use	None
vol-0fc6e878...	10 GiB	gp2	100	-	-	-	September 22, 202...	us-east-1a	available	None

39. Enter "RootVolume" as Description

40. Create Snapshot

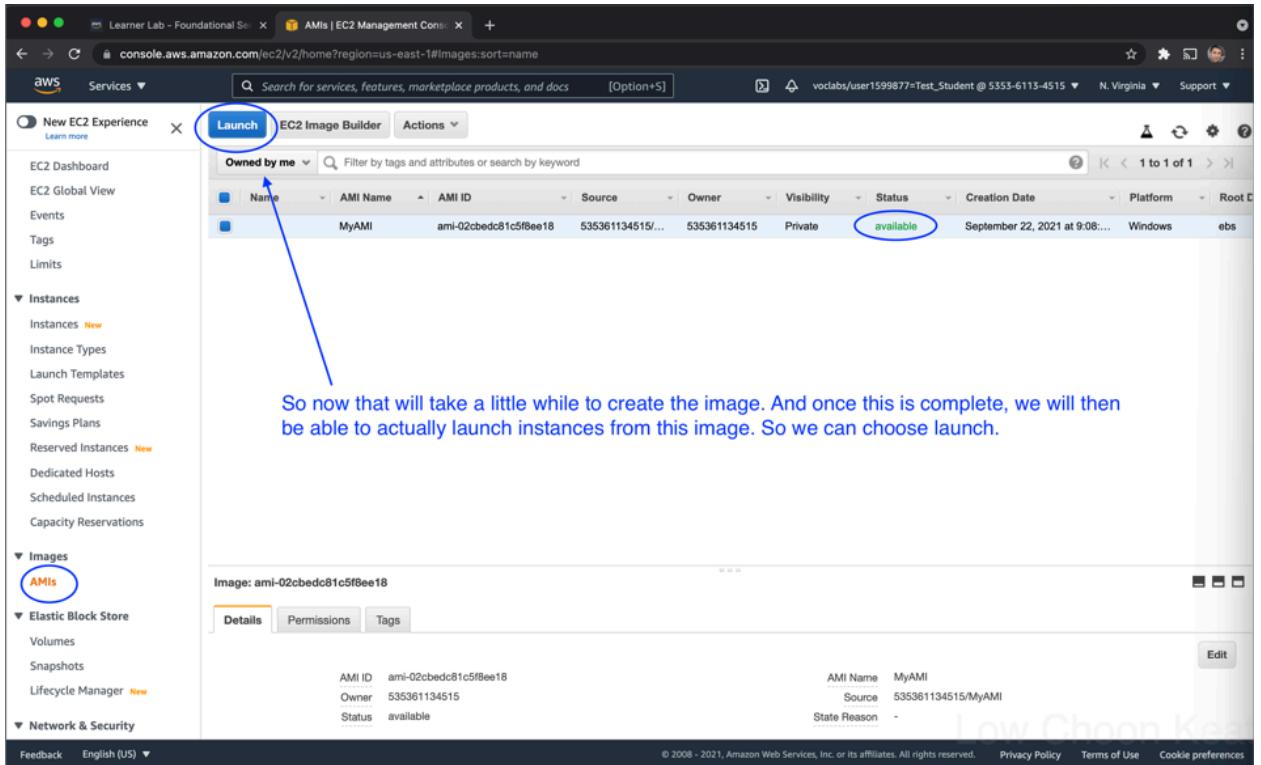
41. Close

42. Go to Elastic Block Store > Snapshots page



The screenshot shows the AWS EC2 Snapshots management console. On the left, there's a navigation sidebar with options like EC2 Dashboard, Instances, Images, and Elastic Block Store. Under Elastic Block Store, the 'Snapshots' link is highlighted and circled in red. The main area displays a table of snapshots. One snapshot, 'snap-02ea98b221d...', has a status of 'pending' and a progress of 0%. Another snapshot, 'snap-08f1febebc20...', has a status of 'completed' and a progress of 100%. A note in the center says: 'And let's go back to the snapshot manager. And we'll just wait for this to complete. While that's happening, let's just go back to the slides to understand what's happening here.'

43. Go to Images > AMIs



The screenshot shows the AWS AMI management console. On the left, the 'AMIs' link in the Images section is highlighted and circled in red. The main area shows a table with one entry: 'MyAMI'. The status column for this entry is 'available'. A blue arrow points from the text 'So now that will take a little while to create the image.' to the 'available' status. Below the table, there's a detailed view of the AMI 'ami-02cbedc81c5f8ee18' with tabs for Details, Permissions, and Tags. The 'Details' tab is selected.

44. Select the "MyAMI", click on "Launch" button

45. Select "t2.micro" as the Instance Type

46. Next: Configure Instance Details

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-0fff136a17db0aca1 (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

Domain join directory: No directory [Create new directory](#)

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.

Tenancy: Shared - Run a shared hardware instance
Additional charges will apply for dedicated tenancy.

So that's it for this lesson, we're not going to launch anything else.

Cancel Previous Review and Launch Next: Add Storage

Feedback English (US) © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

And then we could create our instance in any availability zone. So what you can do here is create and launch an instance, configure your instance, however you want it to be configured, maybe you install your own company software, your management tools, whatever else you might do.

And then you can create an ami from that, and people can launch that ami in any availability zone within the region. So that's why you use ami is to create what we call golden images that you can then use for launching instances with a certain pre-configuration.

47. Click on "Cancel"

48. Select the "MyAMI", Deregister (save money)

And I'm going to make sure that you don't end up spending too much money here. So we want to deregister our ami, that gets rid of the AMI

49. Go to Instances > Instances page, select the instance, terminate the instance

50. Go to Elastic Block Store > Volumes, select all volumes, delete Volumes

back in volumes, we can delete the volumes that we don't need anymore. So let's choose these volumes. gonna delete these volumes.

If there are volumes still showing as in use because the instance is using them. And once that shut down, we'll be able to get rid of those.

51. Go to Elastic Block Store > Snapshots page, delete all the snapshots

The screenshot shows the AWS EC2 Management console with the 'Solutions' tab selected. On the left, the navigation pane includes 'EC2 Dashboard', 'EC2 Global View', 'Events', 'Tags', 'Limits', 'Instances' (with 'Instances' highlighted), 'Images', 'Elastic Block Store' (with 'Snapshots' highlighted and circled in red), and 'Network & Security'. The main content area displays the 'Solutions' page with a search bar and a table of snapshots. A context menu is open over the first two rows of the table, with 'Delete' highlighted. The table has columns for 'Description', 'Status', 'Started', 'Progress', and 'Actions'. Two rows are visible: 'RootVolume' and 'MyTestDataVolume', both marked as 'completed' and 'available (100%)'. A message 'And we've got everything deleted.' is displayed in blue text.

Description	Status	Started	Progress	Actions
RootVolume	completed	September 22, 202...	available (100%)	[Edit]
MyTestDataVolume	completed	September 22, 202...	available (100%)	[Edit]

Snapshots: snap-02ea98b221de0e349, snap-08f1beebc20dc40c

Practical 6

Create Amazon EFS File System

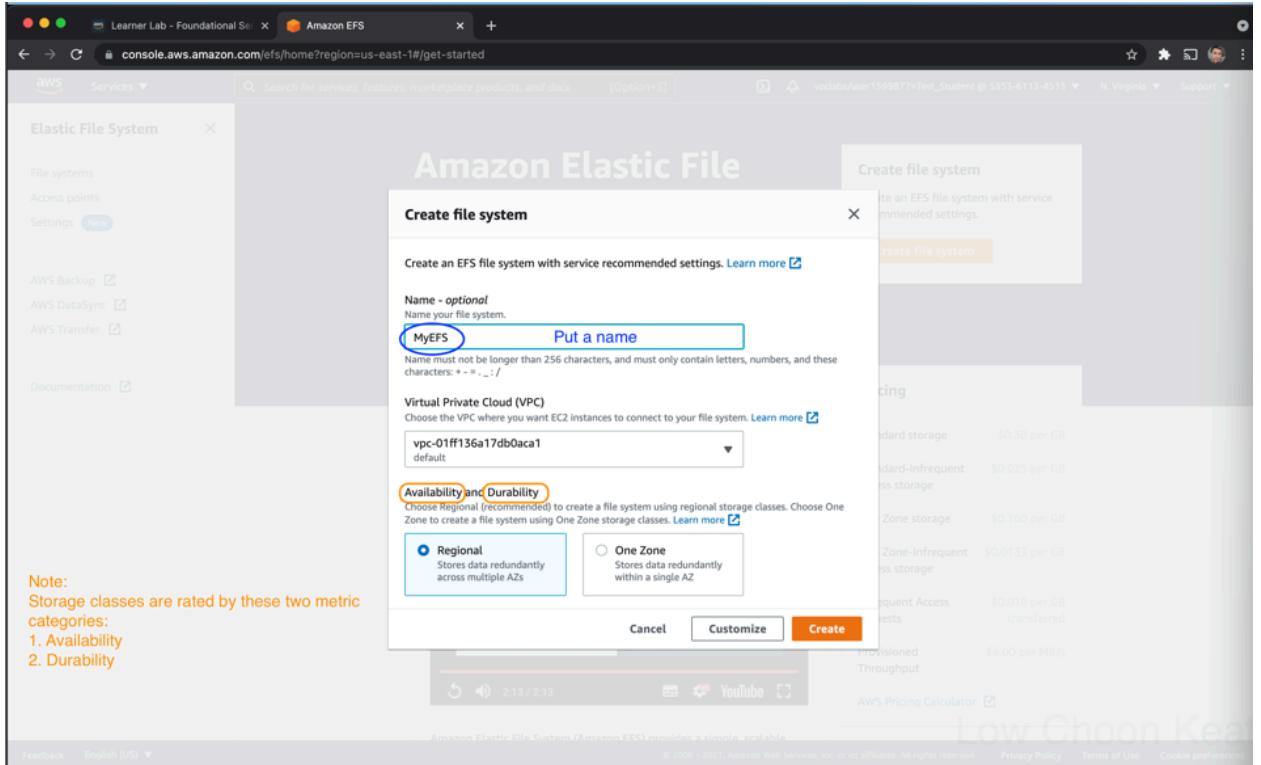
1. Go to **Instances > Instances** page, launch **2 linux instances** (one at us-east-1a, another at us-east-1b)

And we've got everything deleted.

Description	Status	Started	Progress
RootVolume	completed	September 22, 202...	available (100%)
MyTestDataVolume	completed	September 22, 202...	available (100%)

2. Search for **EFS**, click on "Create file system" button
3. Enter "**MyEFS**" as Name

4. Select "Regional" as Availability and Durability



5. Create

6. Go to **Instances > Instances** page, select the instance with **Availability Zone "us-east-1a"**, click on "**Connect**" button
7. Under SSH client tab, copy the Example SSH
8. Open Windows terminal, paste the Example SSH inside and run it
9. For the instance with Availability Zone "us-east-1b", do the same thing also
10. Go back to **EFS page**, click on "**MyEFS**"

11. Under Network tab, click on "Manage" button

The screenshot shows the AWS EFS console under the 'Network' tab. A table lists six mount targets, each associated with an Availability Zone, Mount target ID, Subnet ID, Mount target state, IP address, Network interface ID, and Security groups. The first two rows, 'us-east-1a' and 'us-east-1b', are highlighted with a blue box. The 'Manage' button at the top right of the table is also circled in blue.

Availability zone	Mount target ID	Subnet ID	Mount target state	IP address	Network interface ID	Security groups
us-east-1a	fsmt-64a2e0d1	subnet-0b58630be524c4609	Available	172.31.10.185	eni-00f5e36643c468e41	sg-0ec32d0dedef2d455 (default)
us-east-1b	fsmt-66a2e0d3	subnet-04d6a0702c2d6e504	Available	172.31.95.143	eni-07757872c125cbe3e	sg-0ec32d0dedef2d455 (default)
us-east-1c	fsmt-60a2e0d5	subnet-03add54b488e166ac	Available	172.31.19.95	eni-0d61eb84223509bd4	sg-0ec32d0dedef2d455 (default)
us-east-1d	fsmt-7aa2e0cf	subnet-0ad315974f475c22f	Available	172.31.41.125	eni-078e48f325139c436	sg-0ec32d0dedef2d455 (default)
us-east-1e	fsmt-62a2e0d7	subnet-02d4e1beb65621e52	Available	172.31.59.46	eni-09fc7e5f99fa5a9f1	sg-0ec32d0dedef2d455 (default)
us-east-1f	fsmt-63a2e0d6	subnet-04222845394ce9665	Available	172.31.77.59	eni-054d6a0f9796005e7	sg-0ec32d0dedef2d455 (default)

12. Select the security groups for the Availability Zones

The screenshot shows the 'Network access' section of the AWS EFS console. It displays four mount targets, each with its Availability zone, Subnet ID, IP address, and a 'Security groups' dropdown menu. The 'Choose security groups' dropdowns for the first three zones ('us-east-1a', 'us-east-1b', and 'us-east-1c') are circled in blue.

Availability zone	Subnet ID	IP address	Security groups
us-east-1a	subnet-0b58630be524c4609	172.31.10.185	Choose security groups
us-east-1b	subnet-04d6a0702c2d6e504	172.31.95.143	Choose security groups
us-east-1c	subnet-03add54b488e166ac	172.31.19.95	Choose security groups
us-east-1d	subnet-0ad315974f475c22f	172.31.41.125	Choose security groups

13. Save

14. Go to Network & Security > Security Groups page

The screenshot shows the AWS EC2 Management Console with the URL console.aws.amazon.com/ec2/v2/home?region=us-east-1#SecurityGroups. The left sidebar is collapsed. The main area displays the 'Security Groups (1/2)' page. A table lists two security groups: 'Web-Access' and 'default'. The 'default' row is selected and highlighted with a blue circle. Below the table, a sub-section titled 'sg-Dec32d0dedef2d455 - default' is shown, with the 'Inbound rules' tab selected. An alert message says 'You can now check network connectivity with Reachability Analyzer' with a 'Run Reachability Analyzer' button. The 'Inbound rules (1/1)' table shows one rule: 'sgr-0f7b64541ffb62c7c' with 'All traffic' from 'sg-Dec32d0dedef2d455'. The bottom of the page includes standard AWS footer links: Feedback, English (US), Privacy Policy, Terms of Use, and Cookie preferences.

15. Select "Web-Access", click on "Edit inbound rules" button

16. Click on "Add rule", select "NFS" as Type, select "Web-Access" security group

The screenshot shows the 'Edit inbound rules' page in the AWS EC2 Management Console. A new rule is being added at the bottom of the list. The 'Type' dropdown is set to 'NFS'. The 'Source' dropdown is set to 'Custom' and has a search bar containing 'Web-Access | sg-0220584b139850bd3'. This source entry is highlighted with a blue box. The 'Description - optional' field contains 'Web-Access | sg-0220584b139850bd3'. The 'Save rules' button at the bottom right is also highlighted with a blue box.

The screenshot shows the same 'Edit inbound rules' page after the changes have been saved. The newly added NFS rule is now listed with its source set to 'sg-0220584b139850bd3'. The entire row for this rule is highlighted with a blue box. The 'Save rules' button at the bottom right is also highlighted with a blue box.

17. Save rules

18. Go back to EFS page, inside "MyEFS", click on "Manage" button

19. Delete the security groups and select the "Web-Access" security group

The screenshot shows the AWS EFS Network Access configuration page for a file system named fs-03121cb7. Under the 'Mount targets' section, there are four entries:

- us-east-1a: Subnet ID subnet-0b58630be524c46c, IP address 172.31.10.185, Security groups sg-0ec32d0dedef2d455 (default). The 'Choose security groups' dropdown is open.
- us-east-1b: Subnet ID subnet-04d6a0702c2de630, IP address 172.31.95.143, Security groups sg-0ec32d0dedef2d455 (default). The 'Choose security groups' dropdown is open.
- us-east-1c: Subnet ID subnet-03add54b488e166c, IP address 172.31.19.95, Security groups sg-0ec32d0dedef2d455 (default). The 'Choose security groups' dropdown is open.
- us-east-1d: Subnet ID subnet-0ad315974f475c22, IP address 172.31.41.125, Security groups sg-0ec32d0dedef2d455 (default). The 'Choose security groups' dropdown is open.

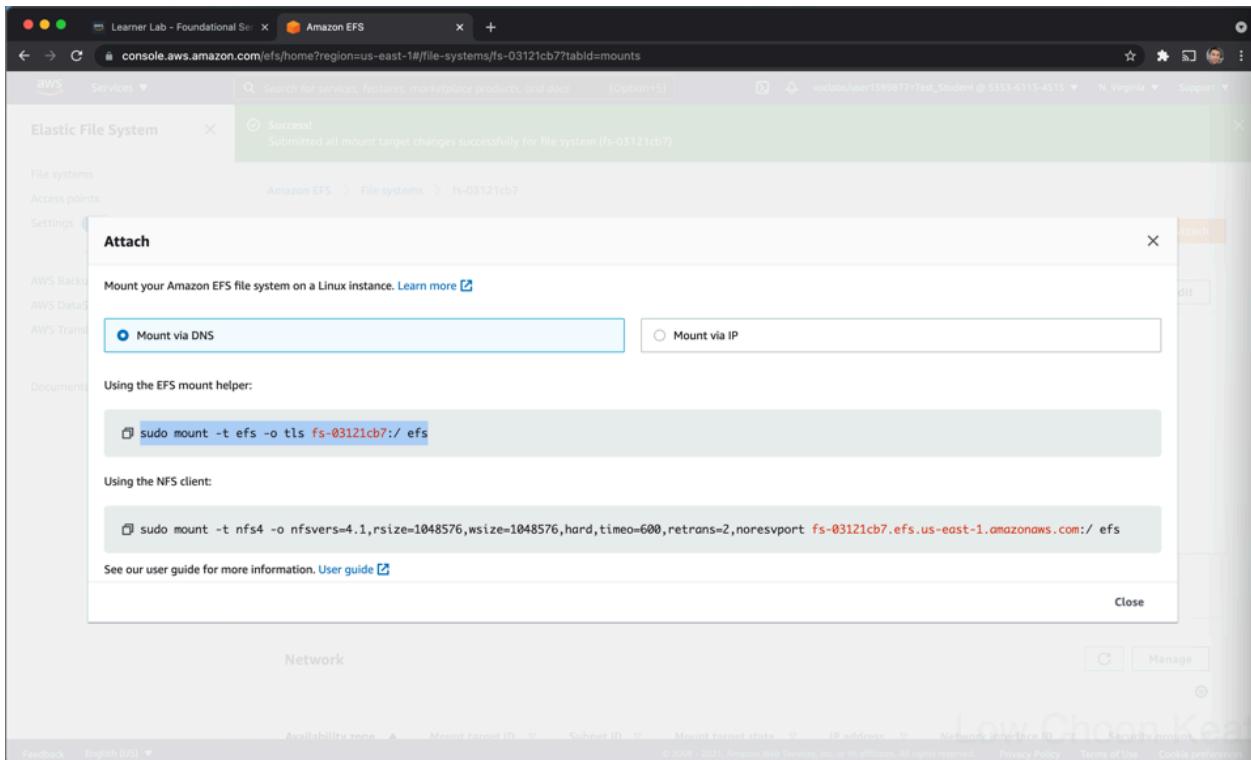
Each entry has a 'Remove' button and a 'delete' link next to the security group name.

The screenshot shows the same AWS EFS Network Access configuration page. The 'Choose security groups' dropdown for the first mount target (us-east-1a) now contains the 'sg-0220584b139850bd3 Web-Access' security group, which is highlighted with a blue border. The other three mount targets still have their default security groups listed.

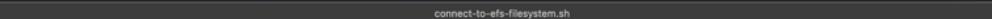
At the bottom right of the page, there are 'Cancel' and 'Save' buttons.

20. In "MyEFS" page, click on "Attach" button

21. Select "Mount via DNS", copy the EFS mount helper command



22. Open "[connect-to-efs-filesystem.sh](#)" file from "Code and Files" folder



```
connect-to-efs-filesystem.sh
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
$ connect-to-efs-filesystem.sh
Users > chonkeatow > Documents > 1BAIT3273 Cloud Computing > 3.0 Practical > Code and Files > $ connect-to-efs-filesystem.sh
1 sudo yum -y install amazon-efs-utils
2 sudo mkdir /mnt/efs
3 sudo mount -t efs fs-af29dd97:/ /mnt/efs
```

23. Copy the commands

24.

The terminal window shows the following output:

```
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

Last login: Wed Apr 26 01:26:59 2023 from 18.206.107.27
[ec2-user@ip-172-31-81-97 ~]$ sudo yum install amazon-efs-utils -y
Last metadata expiration check: 0:06:07 ago on Wed Apr 26 01:25:02 2023.
Dependencies resolved.

Transaction Summary
Install  2 Packages

Total download size: 212 k
Installed size: 556 k
Downloading Packages:
(1/2): amazon-efs-utils-1.35.0-1.amzn2023.noarch.rpm           645 kB/s | 56 kB    00:00
(2/2): stunnel-5.58-1.amzn2023.0.2.x86_64.rpm                1.2 MB/s | 156 kB   00:00
Total                                         1.0 MB/s | 212 kB   00:00

Running transaction check
Transaction check succeeded.

i-032a6f507ef516cb5
PublicIPs: 44.204.193.22  PrivateIPs: 172.31.81.97
```

The CloudShell session shows the following command being run:

```
sudo mkdir /mnt/efs
```

A tooltip for the command states: "This command is to create a directory for our mount point."

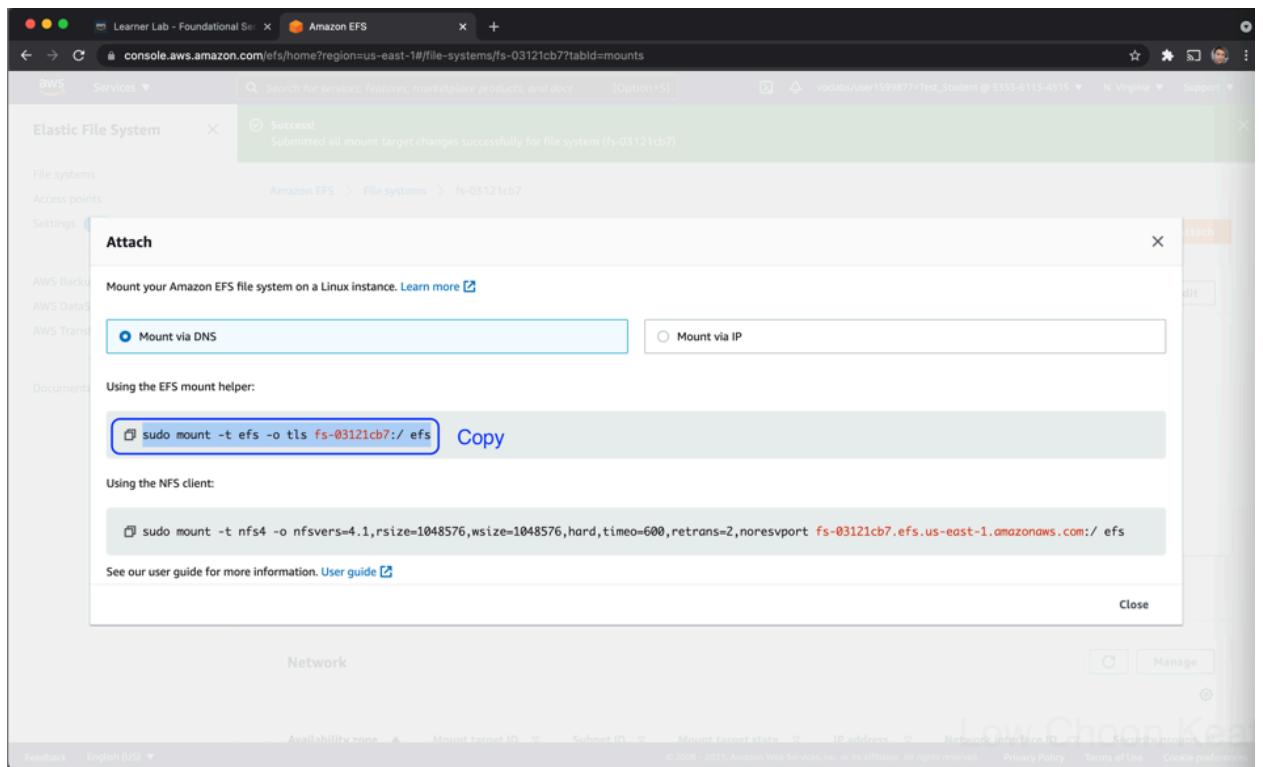
25. Paste the "sudo mkdir /mnt/efs" inside both terminal

The screenshot shows two terminal windows side-by-side. Both terminals are running on Amazon Linux EC2 instances. The left terminal window has a title bar 'connect-to-efs-filesystem.sh'. It displays the output of a script that installs the 'amazon-efs-utils' and 'stunnel' packages. The right terminal window has a title bar '27:~ -- ssh -i vockey.pem ec2-user@ec2-3-95-15-232.compute-1.amazonaws.com'. It also shows the same package installation process. In both terminals, the command 'sudo mkdir /mnt/efs' is highlighted with a yellow box.

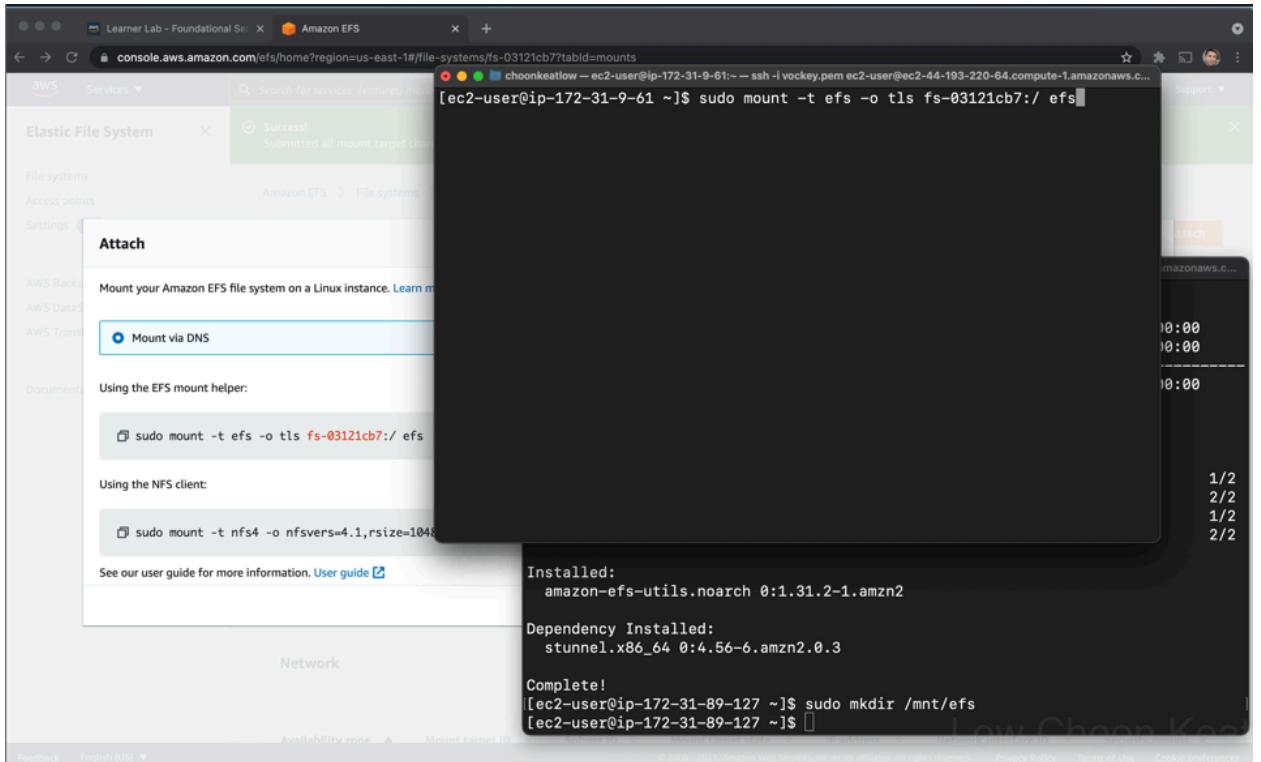
```
connect-to-efs-filesystem.sh
Installed size: 479 k
Downloading packages:
(1/2): amazon-efs-utils-1.31.2-1.amzn2.noarch.rpm      | 46 kB  00:00
(2/2): stunnel-4.56-6.amzn2.0.3.x86_64.rpm           | 149 kB  00:00
Total                                         1.4 MB/s | 195 kB  00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : stunnel-4.56-6.amzn2.0.3.x86_64
  Installing : amazon-efs-utils-1.31.2-1.amzn2.noarch
  Verifying   : stunnel-4.56-6.amzn2.0.3.x86_64
  Verifying   : amazon-efs-utils-1.31.2-1.amzn2.noarch
Installed:
  amazon-efs-utils.noarch 0:1.31.2-1.amzn2
Dependency Installed:
  stunnel.x86_64 0:4.56-6.amzn2.0.3
Complete!
[ec2-user@ip-172-31-9-61 ~]$ sudo mkdir /mnt/efs
[ec2-user@ip-172-31-9-61 ~]$ 
```

```
27:~ -- ssh -i vockey.pem ec2-user@ec2-3-95-15-232.compute-1.amazonaws.com...
1/2 .amzn2.noarch.rpm      | 46 kB  00:00
2/2 x86_64.rpm           | 149 kB  00:00
Total                                         554 kB/s | 195 kB  00:00
1/2
2/2
1/2
2/2
1/2
2/2
amazon-efs-utils.noarch 0:1.31.2-1.amzn2
Dependency Installed:
  stunnel.x86_64 0:4.56-6.amzn2.0.3
Complete!
[ec2-user@ip-172-31-89-127 ~]$ sudo mkdir /mnt/efs
[ec2-user@ip-172-31-89-127 ~]$ 
```

26. Copy the EFS mount helper



27. Paste into the terminals (add /mnt/)



The screenshot shows the AWS EFS Attach interface. A terminal window is open in the background, displaying the command: [ec2-user@ip-172-31-9-61 ~]\$ sudo mount -t efs -o tls fs-03121cb7:/ /mnt/efs. The output shows a success message: Success! Submitted all mount target changes.

```
[ec2-user@ip-172-31-9-61 ~]$ sudo mount -t efs -o tls fs-03121cb7:/ /mnt/efs
Success!
Submitted all mount target changes
```

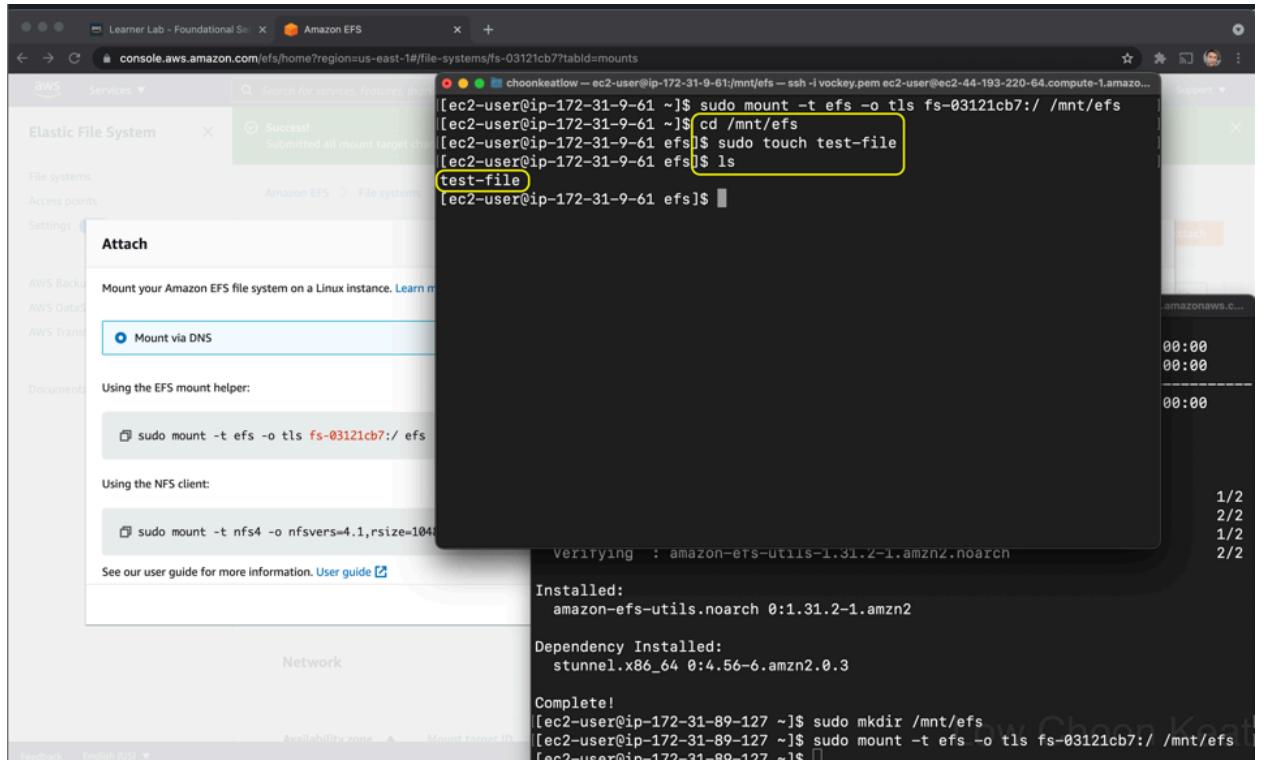
The screenshot shows the AWS EFS Attach interface. A terminal window is open in the background, displaying the command: [ec2-user@ip-172-31-9-61 ~]\$ sudo mount -t efs -o tls fs-03121cb7:/ /mnt/efs. The output shows a success message: Success! Submitted all mount target changes.

```
[ec2-user@ip-172-31-9-61 ~]$ sudo mount -t efs -o tls fs-03121cb7:/ /mnt/efs
Success!
Submitted all mount target changes
```

Below the terminal, the command [ec2-user@ip-172-31-9-61 ~]\$ is shown again, indicating the user is ready to enter the next command.

```
[ec2-user@ip-172-31-9-61 ~]$
```

28. Enter "cd /mnt/efs", "sudo touch test-file", "ls" in one of the terminal

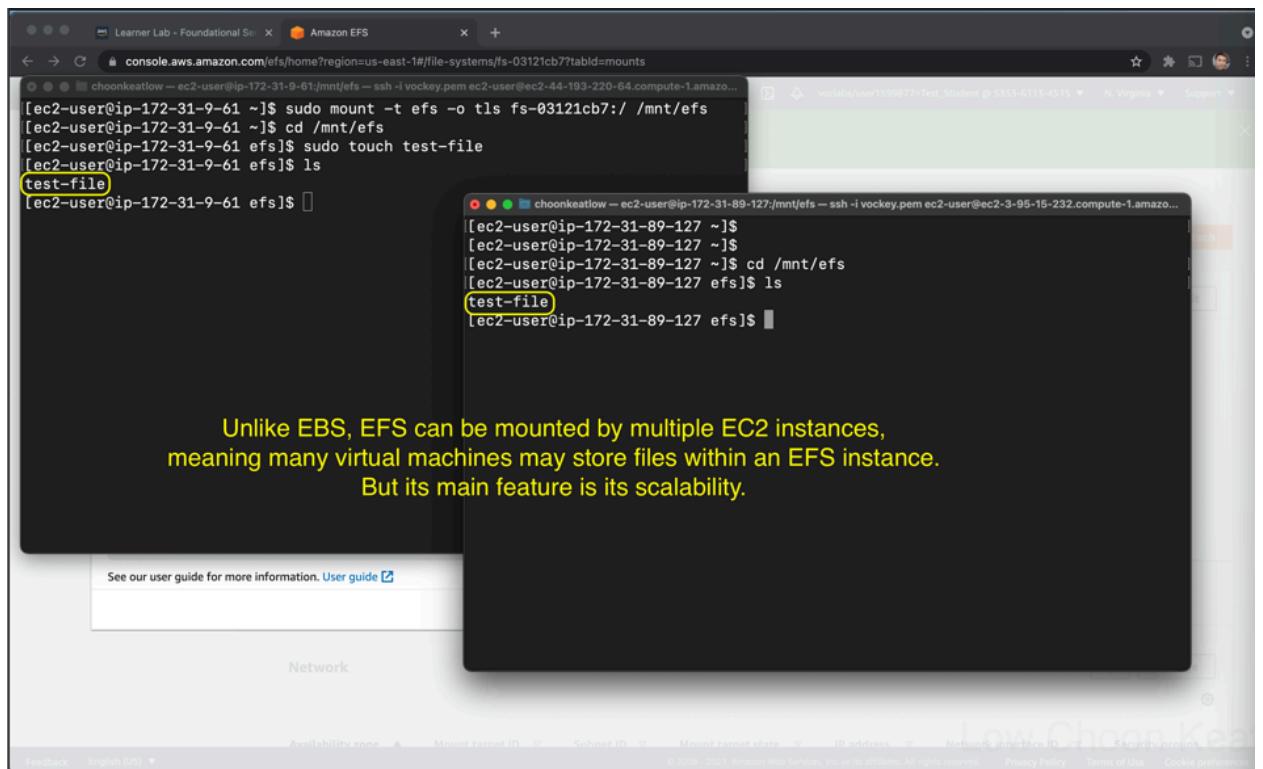


The screenshot shows the AWS Lambda console interface. A modal window titled "Success!" is displayed, stating "Submitted all mount target changes". Below the modal, a terminal window shows the following command sequence:

```
[ec2-user@ip-172-31-9-61 ~]$ sudo mount -t efs fs-03121cb7:/ /mnt/efs
[ec2-user@ip-172-31-9-61 ~]$ cd /mnt/efs
[ec2-user@ip-172-31-9-61 efs]$ sudo touch test-file
[ec2-user@ip-172-31-9-61 efs]$ ls
test-file
[ec2-user@ip-172-31-9-61 efs]$
```

A yellow box highlights the command "sudo touch test-file".

29. In the another terminal (another EC2 instance), enter "ls" (you may see they are synchronized)



The screenshot shows the AWS Lambda console interface. A modal window titled "Success!" is displayed, stating "Submitted all mount target changes". Below the modal, two terminal windows are shown side-by-side. Both windows show the same command sequence:

```
[ec2-user@ip-172-31-9-61 ~]$ sudo mount -t efs fs-03121cb7:/ /mnt/efs
[ec2-user@ip-172-31-9-61 ~]$ cd /mnt/efs
[ec2-user@ip-172-31-9-61 efs]$ sudo touch test-file
[ec2-user@ip-172-31-9-61 efs]$ ls
test-file
[ec2-user@ip-172-31-9-61 efs]$ 
```

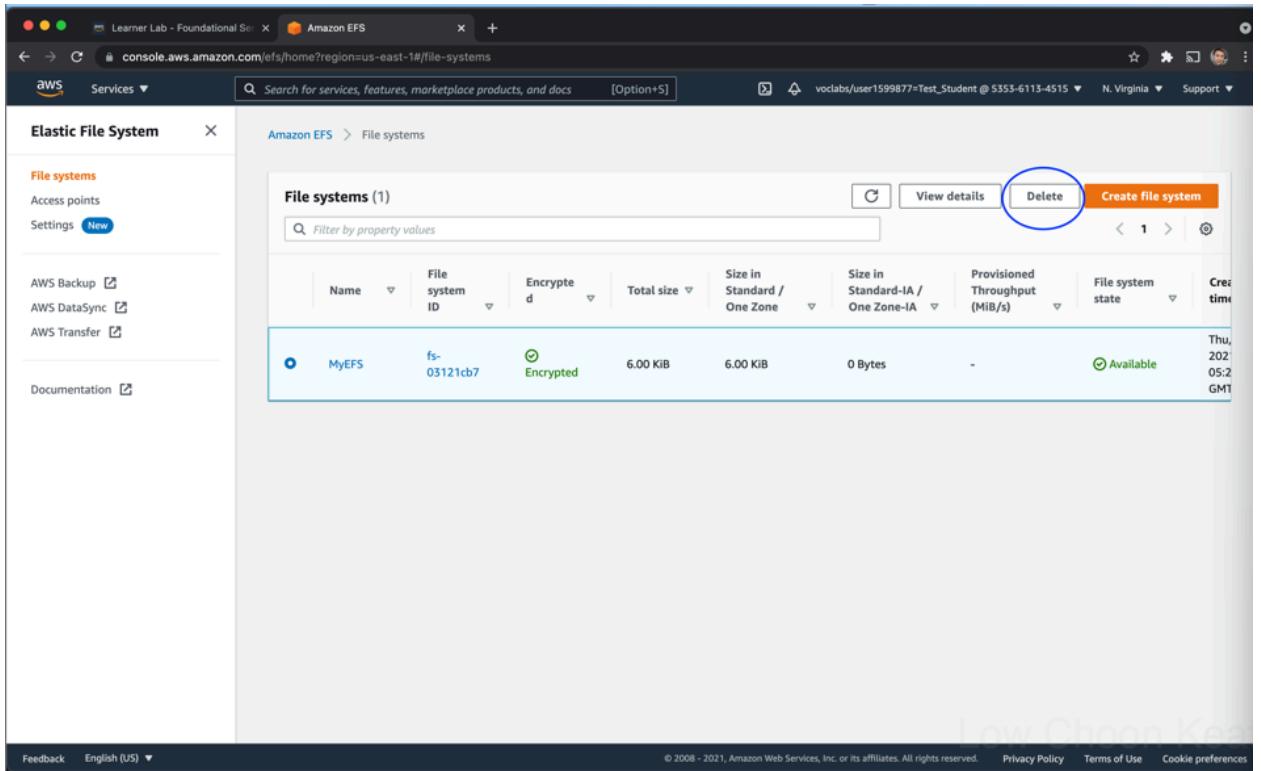
The second terminal window shows the same command sequence:

```
[ec2-user@ip-172-31-89-127 ~]$ 
[ec2-user@ip-172-31-89-127 ~]$ 
[ec2-user@ip-172-31-89-127 ~]$ cd /mnt/efs
[ec2-user@ip-172-31-89-127 efs]$ ls
test-file
[ec2-user@ip-172-31-89-127 efs]$ 
```

A yellow box highlights the file "test-file" in both terminal windows.

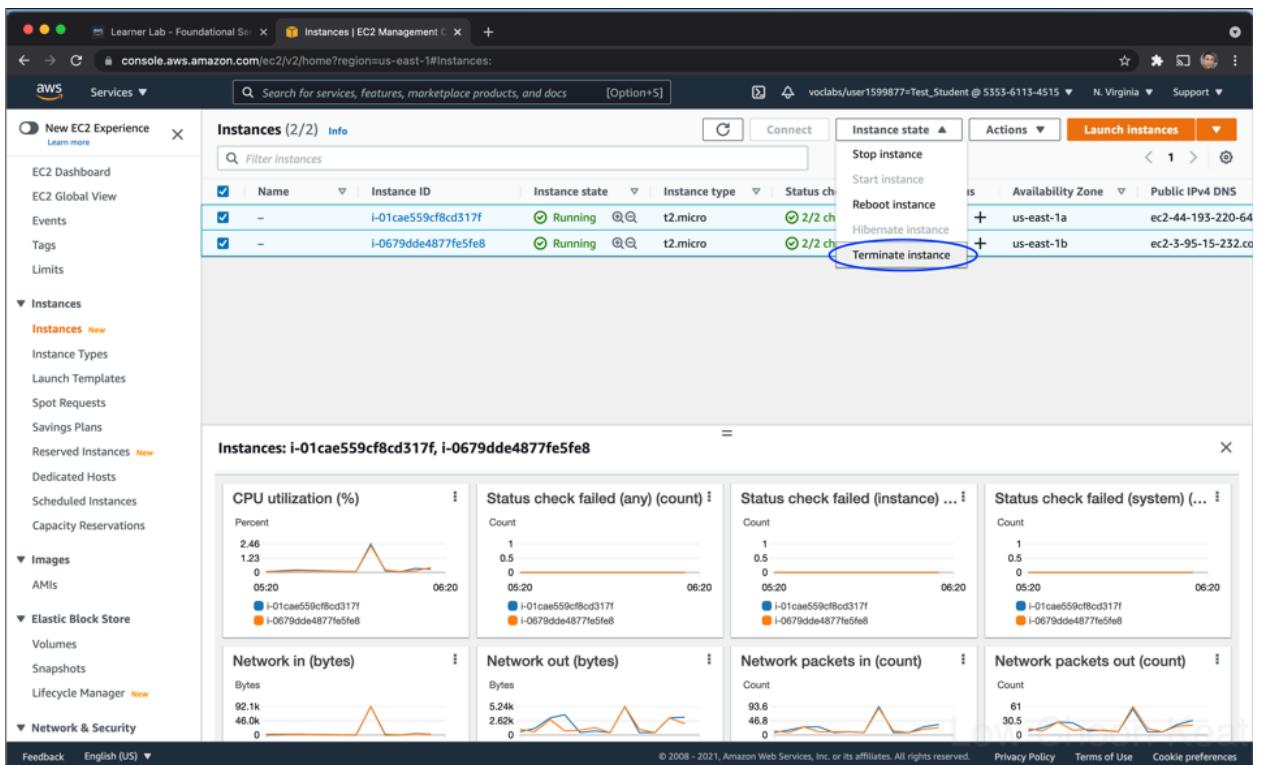
Unlike EBS, EFS can be mounted by multiple EC2 instances, meaning many virtual machines may store files within an EFS instance. But its main feature is its scalability.

30. Go back to Elastic File System page, select "MyEFS", click on "Delete" button



The screenshot shows the 'Amazon EFS > File systems' page. On the left, there's a sidebar with 'File systems', 'Access points', and 'Settings'. The main area displays a table titled 'File systems (1)'. The table has columns for Name, File system ID, Encrypted, Total size, Size in Standard / One Zone, Size in Standard-IA / One Zone-IA, Provisioned Throughput (MiB/s), File system state, and Create time. A single row is selected, showing 'MyEFS' with file system ID 'fs-03121cb7', status 'Encrypted', total size '6.00 KiB', and throughput '0 Bytes'. The 'Delete' button in the top right of the table header is circled in blue.

31. Go to Instances > Instances page, select all instances, click on "Instance state" button, select "Terminate instance"

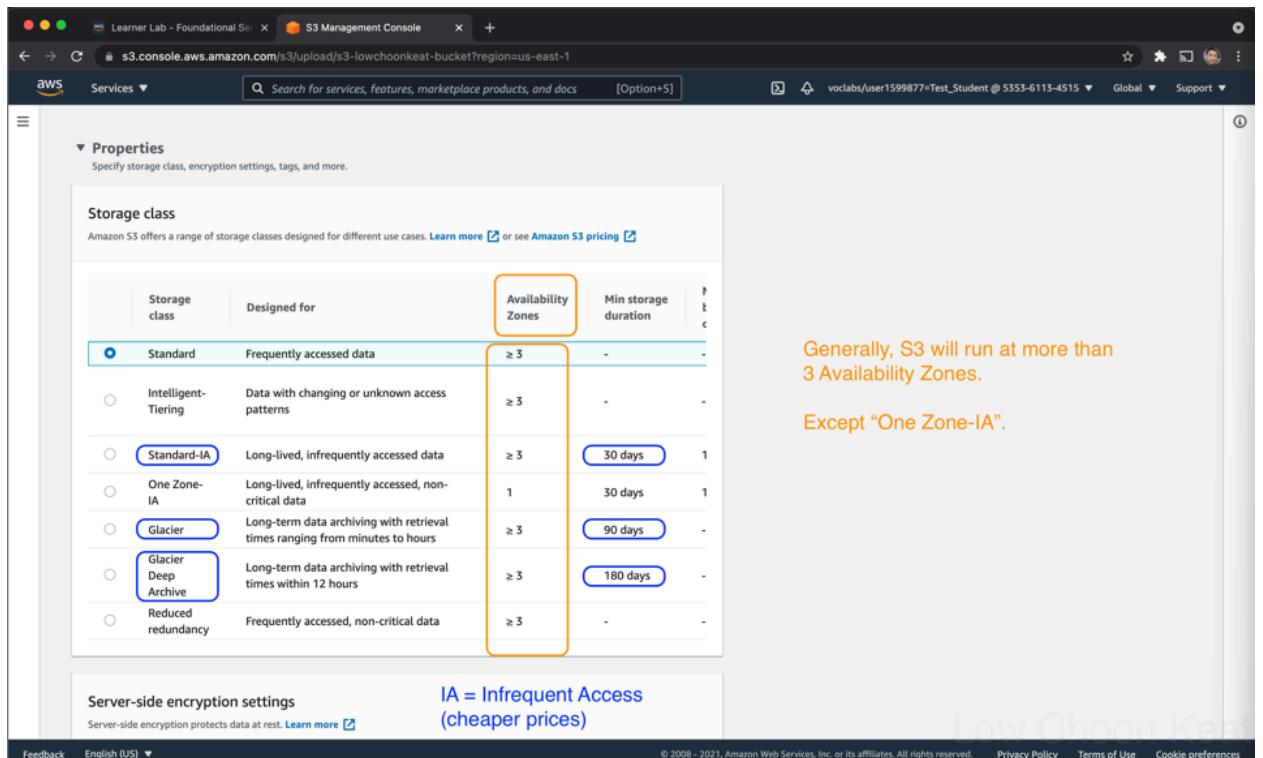


The screenshot shows the 'Instances (2/2)' page in the EC2 Management console. The left sidebar includes 'EC2 Dashboard', 'EC2 Global View', 'Events', 'Tags', and sections for 'Instances', 'Images', 'Elastic Block Store', and 'Network & Security'. The main table lists two instances: 'i-01cae559cf8cd317f' and 'i-0679dde4877fe5fe8', both in 'Running' state. An 'Actions' dropdown menu is open over the second instance, with 'Terminate instance' highlighted and circled in blue. Below the table, a detailed view for the selected instances shows CPU utilization, network traffic, and status check metrics.

Practical 7

S3 Create Bucket and Make Public

1. Search for **S3**
2. **Create bucket**
3. Enter "**s3-yourname-bucket**" as **Bucket name** (must be unique across all of AWS and between 3 - 64 characters in length)
4. Select "**US East (N. Virginia) us-east-1**" as **AWS Region** (Only can select region but not Availability Zones, S3 will run at multi-AZ)
5. **Enable Bucket Versioning**
6. **Create bucket**
7. Click on the created bucket
8. Under **Objects** tab, click on "**Upload**" button
9. Select and **insert "mountain.jpeg" and "ocean.jpg"** files from CloudFront folder
10. Under **Permissions** tab, select "**Choose from predefined ACLs**" as **Access control list (ACL)** and "**Private (recommended)**" as **Predefined ACLs**
11. Under **Properties** section, select "**Standard**" as Storage class

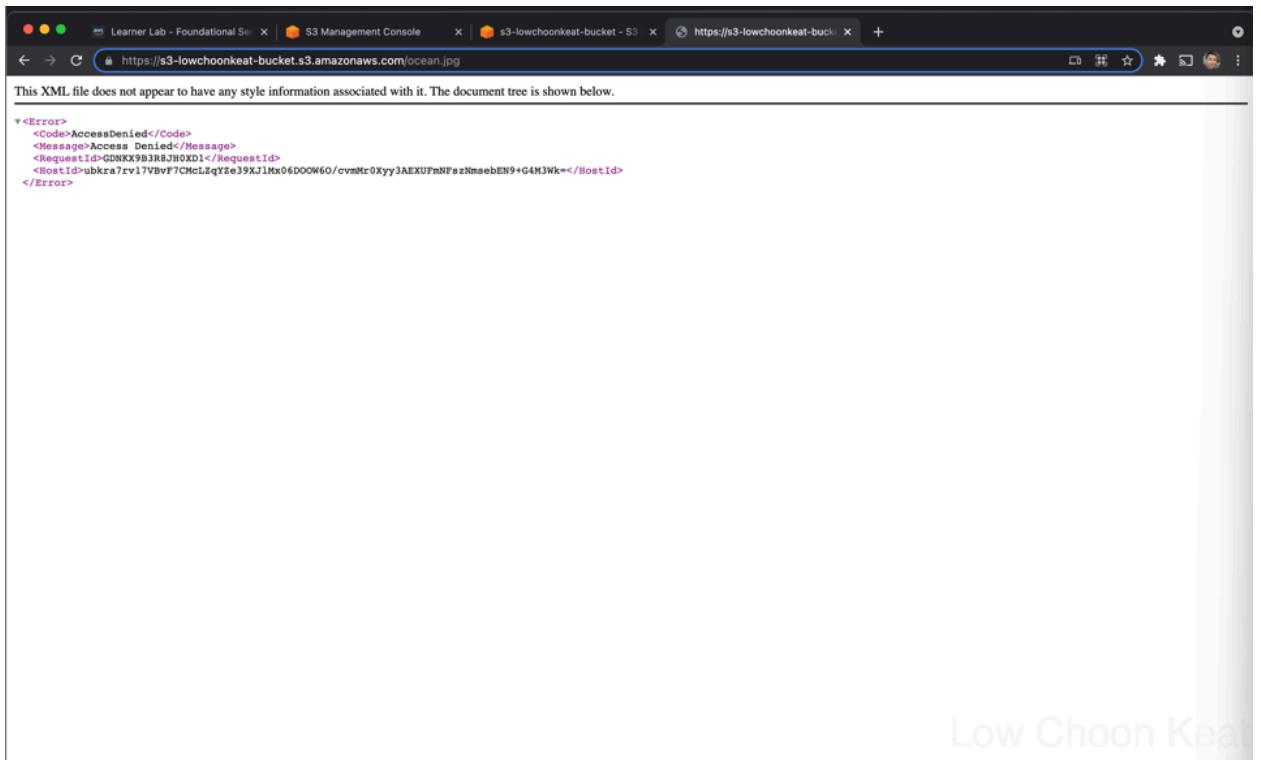


The screenshot shows the AWS S3 Management Console with the 'Properties' tab selected. In the 'Storage class' section, the 'Standard' option is chosen. A callout box highlights the 'Availability Zones' column, which lists values such as '≥ 3', '1', and '30 days'. To the right, a note states: 'Generally, S3 will run at more than 3 Availability Zones. Except "One Zone-IA".'

Storage class	Designed for	Availability Zones	Min storage duration
Standard	Frequently accessed data	≥ 3	-
Intelligent-Tiering	Data with changing or unknown access patterns	≥ 3	-
Standard-IA	Long-lived, infrequently accessed data	≥ 3	30 days
One Zone-IA	Long-lived, infrequently accessed, non-critical data	1	30 days
Glacier	Long-term data archiving with retrieval times ranging from minutes to hours	≥ 3	90 days
Glacier Deep Archive	Long-term data archiving with retrieval times within 12 hours	≥ 3	180 days
Reduced redundancy	Frequently accessed, non-critical data	≥ 3	-

12. Upload
13. Click on the "**ocean.jpg**" file
14. Copy the **Object URL**

15. Paste into the new browser tab and run it



16. Go back to previous section

The screenshot shows the AWS S3 Management Console. On the left, there's a sidebar with 'Amazon S3' and sections for 'Buckets', 'Storage Lens', 'AWS Marketplace', etc. A blue oval highlights the 's3-lowchoonkeat-bucket' link under 'Buckets'. The main area shows an object named 'ocean.jpg' with its 'Info' tab selected. The 'Properties' tab is active. The 'Object overview' section contains the following details:

Property	Value
Owner	awslabsc0w2693566t1628243709
AWS Region	US East (N. Virginia) us-east-1
Last modified	September 23, 2021, 19:33:45 (UTC+08:00)
Size	86.2 KB
Type	jpg
Key	ocean.jpg

On the right side, there are buttons for 'Copy S3 URI', 'Download', 'Open', and 'Object actions'. Below the properties, there's an 'Object management overview' section with a note about bucket properties impacting behavior.

A faint watermark 'Low Choon Keat' is visible in the bottom right corner of the browser window.

17. Under **Permissions > Block public access** (bucket settings) tab, click on "Edit" button
18. Uncheck the "Block all public access" option
19. Save changes
20. In the **Bucket policy** section, click on "Edit" button
21. Open "bucket-policy-public.json" file from "Code and Files" folder

```

{
  "Version": "2008-10-17",
  "Statement": [
    {
      "Sid": "AllowPublicRead",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::BUCKETNAME/*"
    }
  ]
}

```

22. Copy the content and paste into the **Edit bucket policy > Policy**.
23. Rename the BUCKETNAME with the created bucket name

Amazon S3 > s3-lowchoonkeat-bucket > Edit bucket policy

Bucket policy

The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. [Learn more](#)

[Policy examples](#) [Policy generator](#)

Bucket ARN

arn:aws:s3:::s3-lowchoonkeat-bucket

Policy

```

{
  "Version": "2008-10-17",
  "Statement": [
    {
      "Sid": "AllowPublicRead",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::s3-lowchoonkeat-bucket/*"
    }
  ]
}

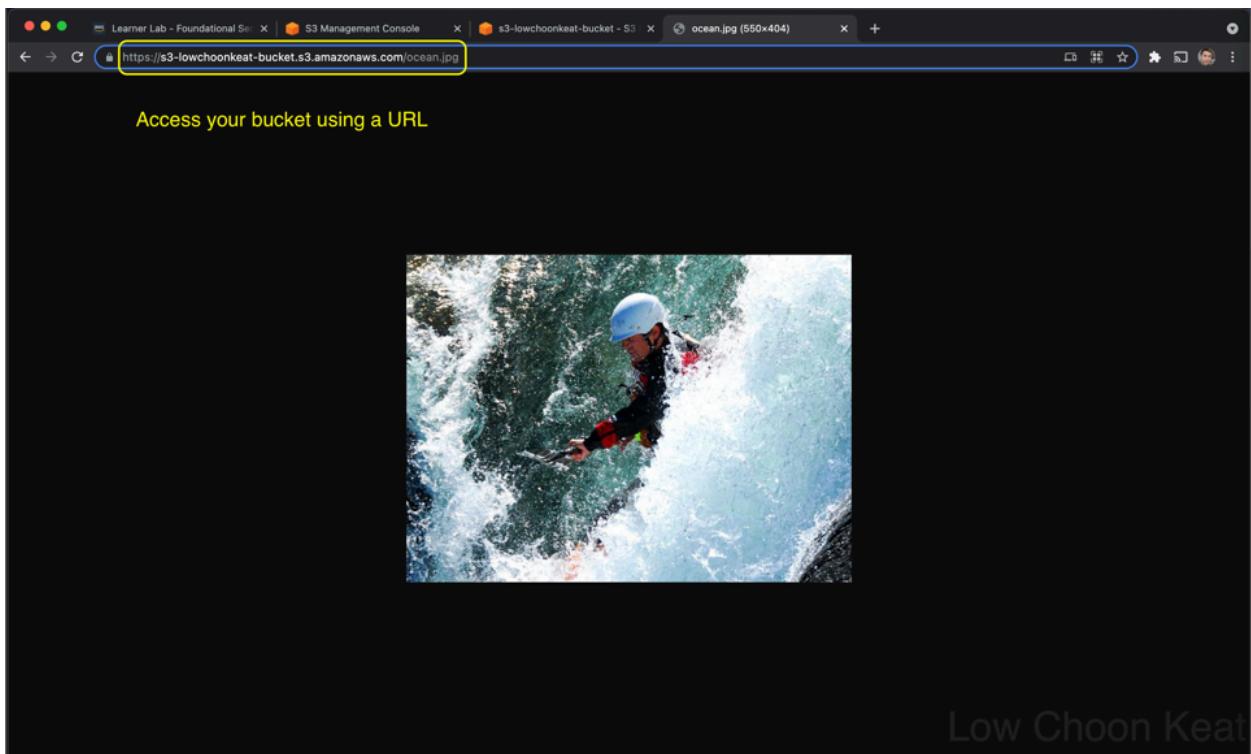
```

The screenshot shows the AWS S3 Management Console with the URL <https://s3.console.aws.amazon.com/s3/bucket/s3-lowchoonkeat-bucket/property/policy/edit?region=us-east-1>. The left sidebar shows 'Amazon S3' with 'Buckets' selected. The main content area is titled 'Edit bucket policy' with a sub-section 'Bucket policy'. The policy JSON is as follows:

```
1 = {"Version": "2008-10-17",
2   "Statement": [{"Sid": "AllowPublicRead",
3     "Effect": "Allow",
4     "Principal": "*",
5     "Action": "s3:GetObject",
6     "Resource": "arn:aws:s3:::s3-lowchoonkeat-bucket/*"
7   }]
8 }
9 ]
10 ]
11 }
```

The screenshot shows the AWS S3 Management Console with the URL <https://s3.console.aws.amazon.com/s3/buckets/s3-lowchoonkeat-bucket?region=us-east-1&tab=permissions>. The left sidebar shows 'Amazon S3' with 'Buckets' selected. The main content area is titled 's3-lowchoonkeat-bucket' with a sub-section 'Permissions overview'. The 'Access' section shows 'Public' with a red warning icon. The 'Block public access (bucket settings)' section shows 'Block all public access' set to 'Off'. A blue oval highlights the 'Publicly accessible' status.

24. Copy the "ocean.jpg" image url and run it in new browser tab



Hosting a Static Website using Amazon S3

1. Search for **S3**
2. **Create bucket**
3. Enter "**yournamewebbucket**" as **Bucket name**
4. **Check the "Block all public access" option**
5. **Create bucket**

The screenshot shows the AWS S3 Management Console interface. A green success message at the top states: "Successfully created bucket 'choonkeatwebbucket'. To upload files and folders, or to configure additional bucket settings choose View details." On the left, a sidebar menu includes options like Buckets, Access Points, Object Lambda Access Points, Batch Operations, and Access analyzer for S3. Under Buckets, it says "Block Public Access settings for this account". Below that is a section for "Storage Lens" with links to Dashboards and AWS Organizations settings. At the bottom of the sidebar are links for Feature spotlight and AWS Marketplace for S3. The main content area has a heading "Account snapshot" with a link to "View Storage Lens dashboard". Below this is a table titled "Buckets (1) Info". The table has columns: Name, AWS Region, Access, and Creation date. It lists one item: "choonkeatwebbucket" (highlighted in blue), "US East (N. Virginia) us-east-1", "Bucket and objects not public", and "August 19, 2021, 21:24:15 (UTC+08:00)". Above the table, there are buttons for "Copy ARN", "Empty", "Delete", and "Create bucket". A message "Bucket created successfully." is displayed below the table. At the bottom of the page, there are links for Feedback, English (US), Privacy Policy, Terms of Use, and Cookie preferences.

6. Go online and search for free html

A screenshot of a Google search results page. The search term 'free html' is highlighted in a yellow oval at the top left. The results show various websites offering free HTML templates. One result from 'www.free-css.com' is highlighted with a yellow box, showing a preview of a template titled 'Download 3233 Free Website Templates - CSS & HTML'. Other results include '8500+ Free HTML Templates - Nicepage' and 'W3.CSS Templates - W3Schools'. A 'People also ask' section is visible on the right.

Low Choon Keat

A screenshot of the 'Free CSS' website. The header features the text 'Free CSS' and 'Free CSS Templates, CSS Layouts & More!'. Below the header is a navigation bar with links for 'Free CSS Templates', 'Premium CSS Templates', 'Free CSS Layouts', and 'Free CSS Menus'. A sidebar on the left says 'Download any templates you like'. The main content area displays a grid of 3233 free website templates. One template in the bottom-left corner is circled in blue and labeled 'sample'. To the right of the grid is a sidebar with ads for 'Podium' (Get more online reviews) and 'fauna' (Statueful JavaScript Apps). The footer contains the 'SITE123' logo and a 'Start Now' button.

Choon Keat

7. Download and drag all the files into the bucket

Search for services, features, marketplace products, and docs [Option+S]

voclabs/user540022=Choon_Keat_Low @ 7628-8221-2687 Global Support

Amazon S3 > choonkeatwebbucket > Upload

Upload Info

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. [Learn more](#)

Drag and drop files and folders you want to upload here, or choose Add files, or Add folders.

Files and folders (90 Total, 2.9 MB)
All files and folders in this table will be uploaded.

<input type="checkbox"/>	Name	Folder	Type	Size
<input type="checkbox"/>	.DS_Store	images/	-	6.0 KB
<input type="checkbox"/>	.DS_Store	images/prettyPhoto/	-	6.0 KB
<input type="checkbox"/>	LICENSE.txt	-	text/plain	1.9 KB
<input type="checkbox"/>	Simple-Line-Icons.eot	icons/fonts/	-	52.3 KB
<input type="checkbox"/>	Simple-Line-Icons.svg	icons/fonts/	image/svg+xml	192.9 KB
<input type="checkbox"/>	Simple-Line-Icons.ttf	icons/fonts/	-	52.1 KB
<input type="checkbox"/>	Simple-Line-Icons.woff	icons/fonts/	application/font-woff	52.2 KB
<input type="checkbox"/>	Simple-Line-Icons.woff2	icons/fonts/	-	25.8 KB
<input type="checkbox"/>	arrow_left.png	rs-plugin/assets/	image/png	807.0 B
<input type="checkbox"/>	arrow_left2.png	rs-plugin/assets/	image/png	348.0 B

Destination

Feedback English (US) © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

8. Go to "Properties" tab, Click on "Edit" button under Static website hosting section

Search for services, features, marketplace products, and docs [Option+S]

voclabs/user540022=Choon_Keat_Low @ 7628-8221-2687 Global Support

Amazon S3 > choonkeatwebbucket > Upload

Upload Info

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. [Learn more](#)

Drag and drop files and folders you want to upload here, or choose Add files, or Add folders.

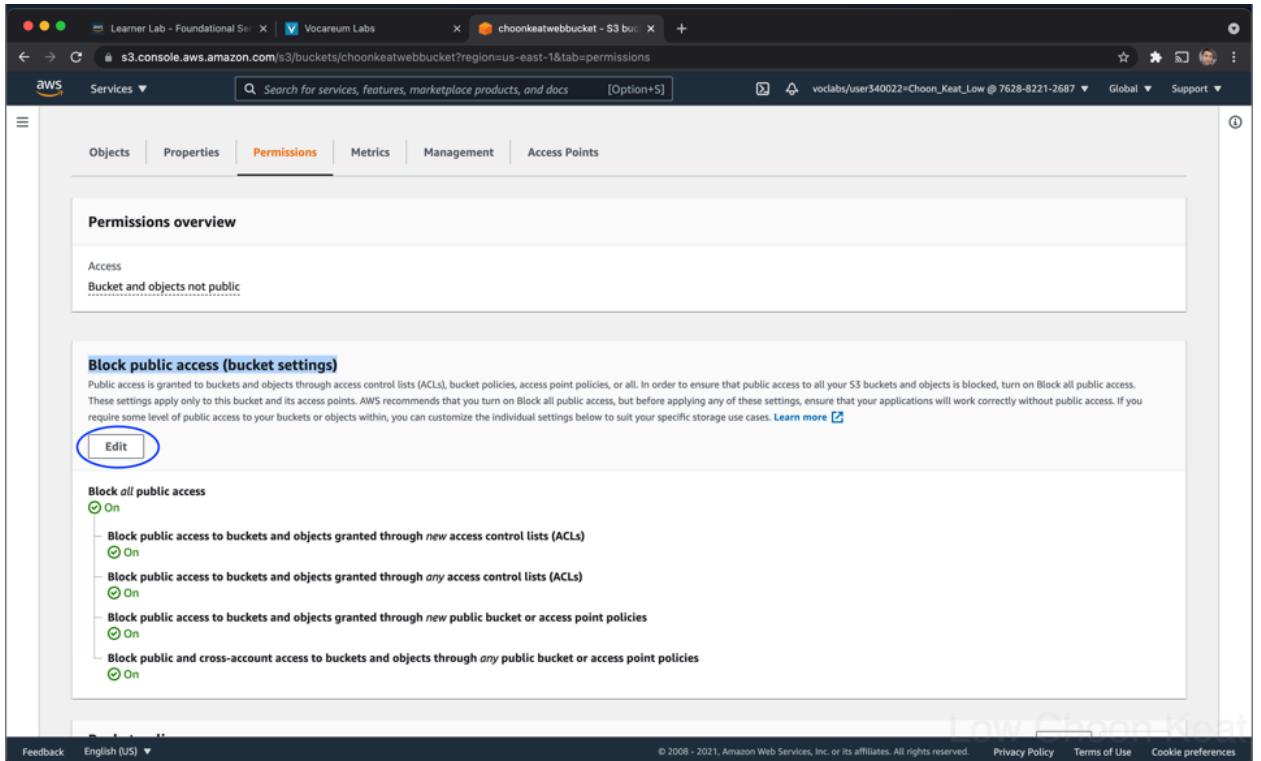
Files and folders (90 Total, 2.9 MB)
All files and folders in this table will be uploaded.

<input type="checkbox"/>	Name	Folder	Type	Size
<input type="checkbox"/>	.DS_Store	images/	-	6.0 KB
<input type="checkbox"/>	.DS_Store	images/prettyPhoto/	-	6.0 KB
<input type="checkbox"/>	LICENSE.txt	-	text/plain	1.9 KB
<input type="checkbox"/>	Simple-Line-Icons.eot	icons/fonts/	-	52.3 KB
<input type="checkbox"/>	Simple-Line-Icons.svg	icons/fonts/	image/svg+xml	192.9 KB
<input type="checkbox"/>	Simple-Line-Icons.ttf	icons/fonts/	-	52.1 KB
<input type="checkbox"/>	Simple-Line-Icons.woff	icons/fonts/	application/font-woff	52.2 KB
<input type="checkbox"/>	Simple-Line-Icons.woff2	icons/fonts/	-	25.8 KB
<input type="checkbox"/>	arrow_left.png	rs-plugin/assets/	image/png	807.0 B
<input type="checkbox"/>	arrow_left2.png	rs-plugin/assets/	image/png	348.0 B

Destination

Feedback English (US) © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

9. Enable "Static website hosting" option
10. Enter "index.html" as Index document
11. Under Permissions > Block public access (bucket settings) tab, click on "Edit" button



The screenshot shows the AWS S3 console interface. The top navigation bar has tabs for Objects, Properties, Permissions (which is currently selected), Metrics, Management, and Access Points. Below the navigation bar, there's a search bar and a breadcrumb trail showing the path: s3.console.aws.amazon.com/s3/buckets/choonkeatwebbucket?region=us-east-1&tab=permissions. The main content area is titled 'Permissions overview' and shows the 'Access' section with 'Bucket and objects not public'. A large box titled 'Block public access (bucket settings)' contains the following information:
Block all public access
On

- Block public access to buckets and objects granted through new access control lists (ACLs)
On
- Block public access to buckets and objects granted through any access control lists (ACLs)
On
- Block public access to buckets and objects granted through new public bucket or access point policies
On
- Block public and cross-account access to buckets and objects through any public bucket or access point policies
On

At the bottom of the page, there are links for Feedback, English (US), © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved., Privacy Policy, Terms of Use, and Cookie preferences.

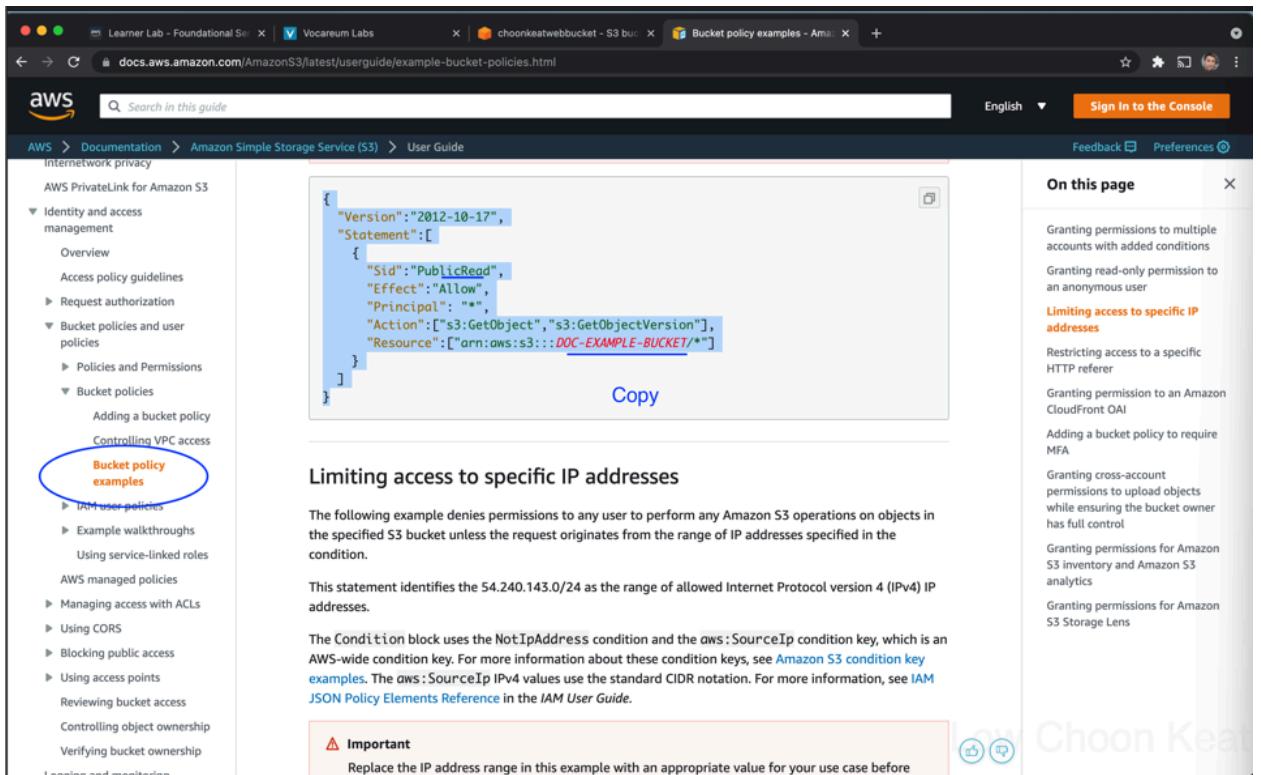
12. Uncheck the "Block all public access" option

The screenshot shows the AWS S3 console for a bucket named 'choonkeatwebbucket'. The 'Permissions' tab is selected. A success message at the top says 'Successfully edited Block Public Access settings for this bucket.' Below it, the 'Permissions overview' section shows 'Access' and 'Objects can be public'. The 'Block public access (bucket settings)' section is expanded, showing five options under 'Block all public access' which are all currently set to 'Off'. To the right of this section, a blue box highlights the 'Edit' button and the 'Block all public access' section, with the text 'Uncheck "Block all public access"' overlaid.

13. Under Bucket policy section, click on "Learn more" hyperlink

The screenshot shows the same AWS S3 console for the 'choonkeatwebbucket'. The 'Bucket policy' section is visible, containing a note about bucket policies and a 'Learn more' link. This link is circled in blue. Below the note, it says 'No policy to display.' and has a 'Copy' button. The 'Object ownership' section is also partially visible at the bottom.

14. Go to Bucket policies > Bucket policy examples, copy the script

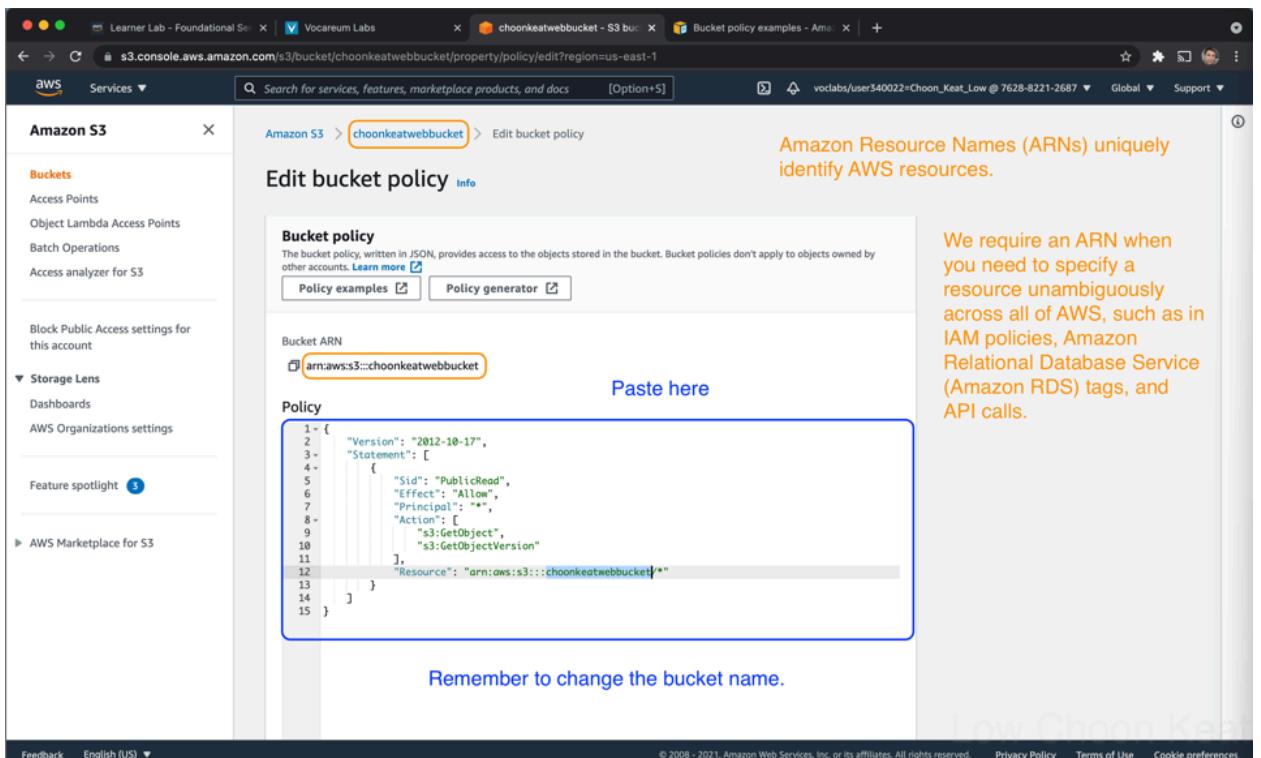


The screenshot shows the AWS Documentation for the Amazon Simple Storage Service (S3) User Guide. In the left sidebar, under the 'Identity and access management' section, the 'Bucket policy examples' link is highlighted with a blue oval. The main content area displays a JSON policy script:

```
{ "Version": "2012-10-17", "Statement": [ { "Sid": "PublicRead", "Effect": "Allow", "Principal": "*", "Action": ["s3:GetObject", "s3:GetObjectVersion"], "Resource": ["arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"] } ] }
```

A 'Copy' button is located below the JSON code. To the right, there is a sidebar titled 'On this page' with several links related to bucket policies and permissions.

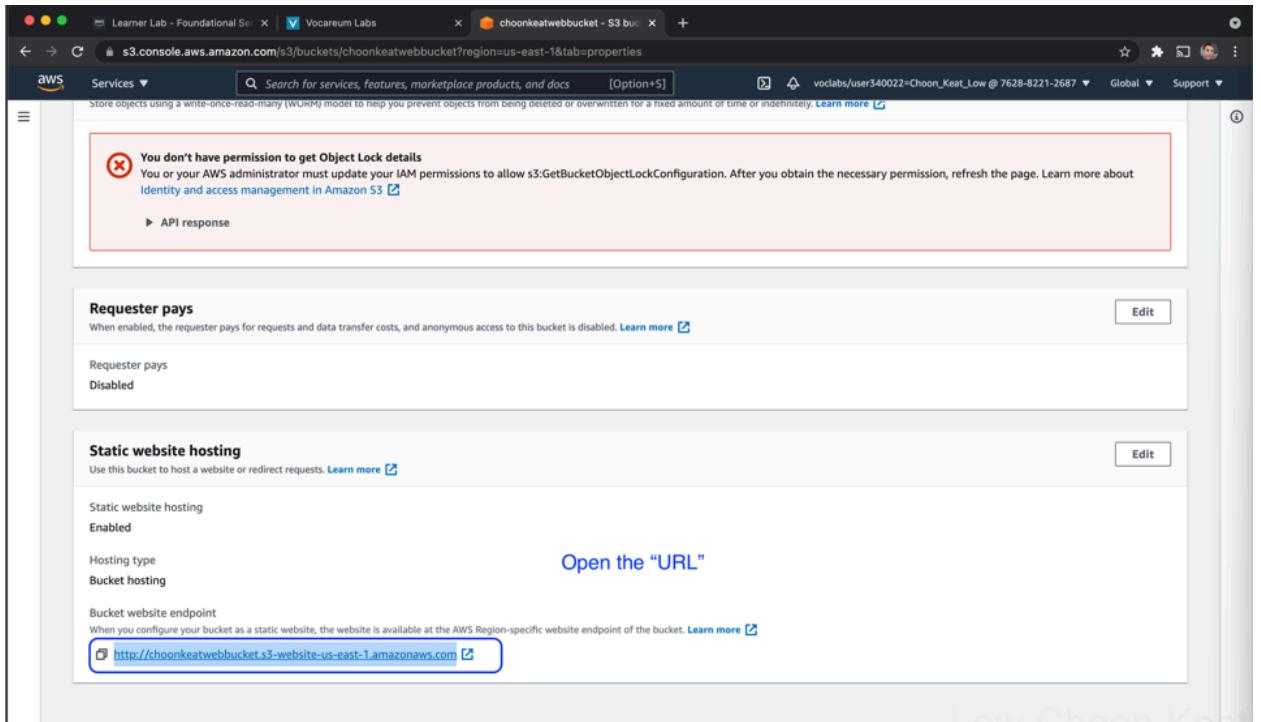
15. Under Bucket policy section, click on "Edit" button. Then, paste the script into the Policy. Replace the bucket name with the created bucket name



The screenshot shows the AWS S3 console with the 'Edit bucket policy' page open. The left sidebar shows the 'Buckets' section. The main area has a heading 'Edit bucket policy' with a 'Info' link. Below it is a 'Bucket policy' section with 'Policy examples' and 'Policy generator' buttons. The 'Policy' text area contains the JSON script from the previous screenshot. A note at the bottom of the policy area says 'Remember to change the bucket name.' To the right, a note says 'We require an ARN when you need to specify a resource unambiguously across all of AWS, such as in IAM policies, Amazon Relational Database Service (Amazon RDS) tags, and API calls.' The 'Bucket ARN' field is populated with 'arnaws:s3::choonkeatwebbucket'.

16. Save

17. Copy or directly open the url under the Static website hosting tab



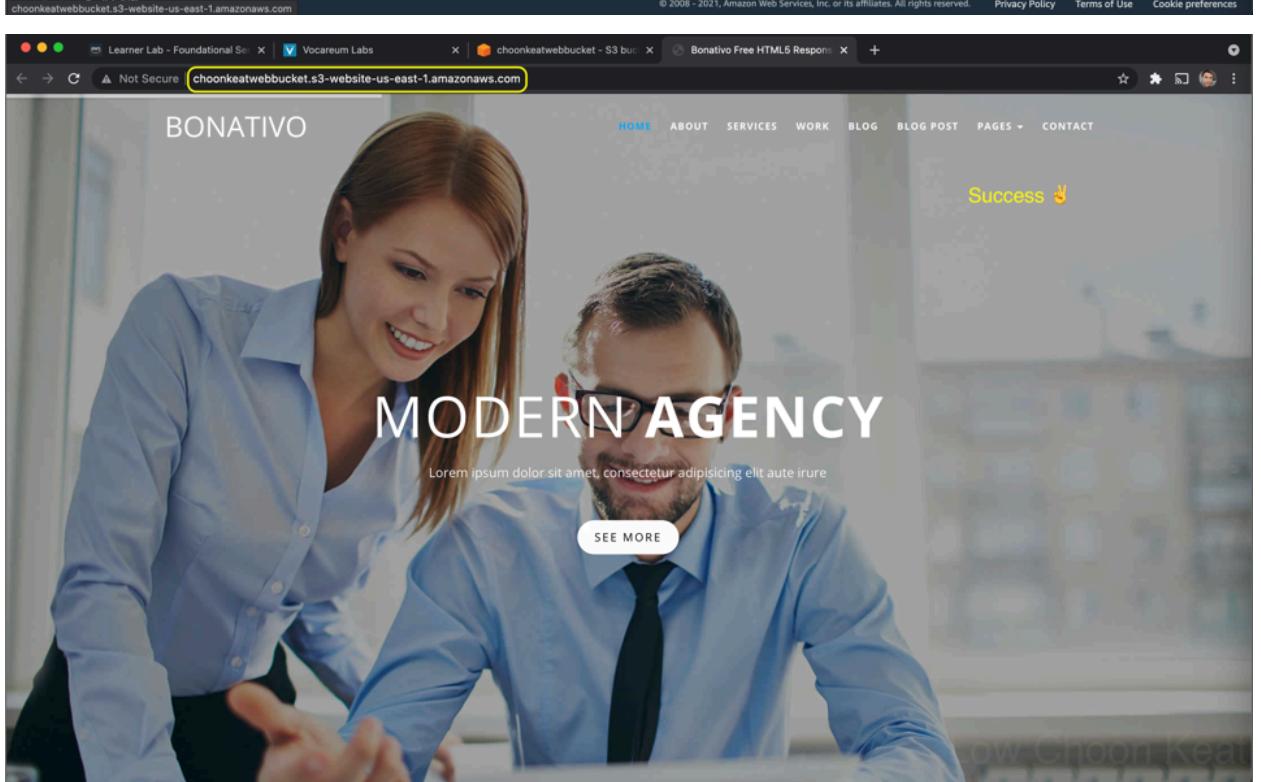
You don't have permission to get Object Lock details
You or your AWS administrator must update your IAM permissions to allow s3:GetBucketObjectLockConfiguration. After you obtain the necessary permission, refresh the page. Learn more about Identity and access management in Amazon S3.

Requester pays
When enabled, the requester pays for requests and data transfer costs, and anonymous access to this bucket is disabled. Learn more.

Static website hosting
Use this bucket to host a website or redirect requests. Learn more.

Bucket website endpoint
When you configure your bucket as a static website, the website is available at the AWS Region-specific website endpoint of the bucket. Learn more.

<http://choonkeatwebbucket.s3-website-us-east-1.amazonaws.com>



Mock Test

Thanks for the clarification! Since your **Cloud Computing practical test** is hands-on and done within **Learner Lab**, with commands and code provided, here's a **mock practical test** that mirrors that format. It gives you a set of tasks to perform, just like in the real test, and you can use it to practice step-by-step in the lab.

Cloud Computing Mock Practical Test (Learner Lab Style)

Task 1: Launch and Configure EC2 Instance

- Launch an EC2 instance using **Amazon Linux 2023 AMI**.
- Use **t2.micro** instance type and **vockey** key pair.
- Create or select a security group named **Web-Access**.
- Allow **SSH (port 22)** and **HTTP (port 80)** from **Anywhere**.

Connect to the instance using SSH and install Apache using:

```
sudo yum update -y  
sudo yum install httpd -y  
sudo systemctl start httpd  
sudo systemctl enable httpd  
echo "This is a test page running on Apache on EC2 in the AWS Cloud" &gt;  
/var/www/html/index.html
```

•

Task 2: Deploy Website Using EC2 User Data

- Launch a new EC2 instance with the same settings.
- In **Advanced Details**, paste the content from `ec2-instance-user-data.sh`.
- Use the **Web-Access** security group.

- After launch, access the website using the instance's **Public IPv4 address**.
-

Task 3: Use S3 with EC2

- Create an S3 bucket and upload `index.txt` and `names.csv`.
 - Launch an EC2 instance and paste the script from `user-data-to-create-index-html-with-names.sh` into **User Data**.
 - Replace `<YOUR-BUCKET-NAME>` with your actual bucket name.
 - Assign **LabInstanceProfile** IAM role to the instance.
 - Access the generated webpage via the instance's public IP.
-

Task 4: Auto Scaling Setup

- Create a **Launch Template** named `LC1` with Amazon Linux 2 AMI.
 - Create an **Auto Scaling Group** named `ASG1` using `LC1`.
 - Set:
 - Desired capacity: 3
 - Minimum: 2
 - Maximum: 4
 - Use subnets in multiple AZs and no load balancer.
 - Terminate 2 instances and observe auto-scaling behavior.
-

Task 5: Auto Scaling with Scaling Policy

- Add a **dynamic scaling policy** to `ASG1` with target CPU utilization of 50%.

Connect to 2 EC2 instances and install stress tool using:

```
sudo yum install stress -y
```

```
stress -c 8
```

- Wait for CloudWatch to trigger scaling.
-

Task 6: Load Balancer Integration

- Create an **Application Load Balancer** named **ALB**.
 - Create a **Target Group** named **TG1**.
 - Create a new **Launch Template** **LT1** with user data script and IAM role.
 - Create a new **Auto Scaling Group** using **LT1**, attach to **TG1**.
 - Set desired capacity to 4.
 - Access the website using the **DNS name** of the load balancer.
-

Task 7: EBS Volume and Snapshot

- Create a **10 GiB EBS volume** in **us-east-1a**.
 - Launch a **Windows EC2 instance** in the same AZ.
 - Attach the volume, initialize, format, and create a file.
 - Create a **snapshot** of the volume.
 - Launch a new instance in **us-east-1b**, create a volume from snapshot, and attach it.
-

Task 8: Create and Mount EFS

- Launch two EC2 instances in different AZs.
- Create an **EFS file system** named **MyEFS**.
- Configure security groups to allow **NFS**.

Mount EFS using:

```
sudo mkdir /mnt/efs  
sudo mount -t efs &lt;EFS-ID&gt;:/ /mnt/efs
```

- Create a file in one instance and verify it appears in the other.
-

Task 9: S3 Bucket Public Access

- Create an S3 bucket named **s3-yourname-bucket**.
- Upload **mountain.jpeg** and **ocean.jpg**.

- Modify bucket policy using `bucket-policy-public.json`.
 - Access the image via its **Object URL**.
-

Task 10: Host Static Website on S3

- Create a bucket named `yournamewebbucket`.
 - Upload HTML files.
 - Enable **Static Website Hosting** and set `index.html` as the index document.
 - Modify bucket policy to allow public access.
 - Access the website via the **S3 website endpoint URL**.
-

Mock Test 2



Integrated Cloud Computing Mock Practical Test

Scenario:

You are a cloud engineer tasked with deploying a scalable, highly available web application on AWS. The application must serve dynamic content from S3, scale automatically based on CPU usage, and be accessible via a load balancer. You must also implement persistent storage using EBS and shared storage using EFS. Finally, host a static website using S3 and clean up all resources after deployment.

Step-by-Step Tasks

1. Launch EC2 Instance and Configure Apache

- Launch an EC2 instance using **Amazon Linux 2023 AMI**, **t2.micro**, and **vockey** key pair.
- Create a security group **Web-Access** allowing **SSH (22)** and **HTTP (80)**.

Connect via SSH and run:

```
sudo yum update -y  
sudo yum install httpd -y  
sudo systemctl start httpd  
sudo systemctl enable httpd  
echo "Welcome to EC2 Web Server" > /var/www/html/index.html
```

•

2. Create S3 Bucket and Upload Files

- Create a bucket named **yourname-data-bucket**.
- Upload **index.txt** and **names.csv**.

- Enable public access and versioning.
- Assign bucket policy using `bucket-policy-public.json`.

3. Launch EC2 with User Data to Generate Web Page from S3

- Launch a new EC2 instance with IAM role `LabInstanceProfile`.
- Paste the script from `user-data-to-create-index-html-with-names.sh` into **User Data**.
- Replace `<YOUR-BUCKET-NAME>` with your actual bucket name.
- Access the generated webpage via public IP.

4. Create Launch Template and Auto Scaling Group

- Create a launch template `LT1` using the same AMI and user data script.
- Create an Auto Scaling Group `ASG1` with:
 - Desired: 3
 - Min: 2
 - Max: 4
 - Subnets in multiple AZs
 - No load balancer yet

5. Add Dynamic Scaling Policy

- Add a scaling policy to maintain **50% CPU utilization**.

Connect to 2 EC2 instances and run:

```
sudo yum install stress -y
```

```
stress -c 8
```

- Wait for CloudWatch to trigger scaling.

6. Create Application Load Balancer

- Create ALB named `MyALB` with listener on port 80.
- Create target group `TG1` and attach to ALB.
- Update `ASG1` to use `TG1` for load balancing.
- Access the application via ALB DNS name.

7. Create and Attach EBS Volume to Windows EC2

- Create a 10 GiB EBS volume in **us-east-1a**.
- Launch a Windows EC2 instance in the same AZ.
- Attach the volume, initialize, format, and create a file.
- Create a snapshot named **MyTestDataVolume**.

8. Restore Volume from Snapshot

- Launch another Windows EC2 in **us-east-1b**.
- Create a volume from the snapshot and attach it.
- Verify the file exists.

9. Create and Mount EFS on Two EC2 Instances

- Launch two EC2 instances in different AZs.
- Create EFS named **MyEFS**.
- Update security group to allow **NFS**.

Mount EFS using:

```
sudo mkdir /mnt/efs
```

```
sudo mount -t efs &lt;EFS-ID&gt;:/ /mnt/efs
```

- Create a file in one instance and verify it appears in the other.

10. Host Static Website on S3

- Create bucket **yournamewebbucket**.
 - Upload HTML files.
 - Enable **Static Website Hosting** and set **index.html** as index document.
 - Modify bucket policy for public access.
 - Access the website via S3 endpoint.
-



Final Cleanup Tasks

1. **Terminate EC2 Instances:**
 - Go to EC2 → Instances → Terminate all running instances.
2. **Delete Auto Scaling Group and Launch Templates:**
 - Delete **ASG1** and **LT1**.
3. **Delete Load Balancer and Target Group:**

- Delete MyALB and TG1.
4. **Delete EBS Volumes and Snapshots:**
- Go to EBS → Volumes → Delete all.
 - Go to Snapshots → Delete all.
5. **Delete EFS File System:**
- Go to EFS → Delete MyEFS.
6. **Delete S3 Buckets:**
- Delete yourname-data-bucket and yournamewebbucket.
-