Virtual Machines

AWS EC2 & RDS

Virtual Machines

What is a Virtual Machine?

- A virtual machine is a machine which appears to be a real one, but in fact is implemented as software.
  - Run any software you want
  - Runs an ordinary OS
  - Mostly, looks like you're on real hardware
- On a big server, you can run many small VMs

Virtual Machines

VMs vs. Containers

- How is a VM different than a container?
  - Simulates a complete machine
  - Runs its own kernel
  - Has virtual CPUs, memory (has complete control)
  - Has (virtual) hard drives, filesystems
  - **Persistent data**
  - Tricky to run one container inside another

Virtual Machines

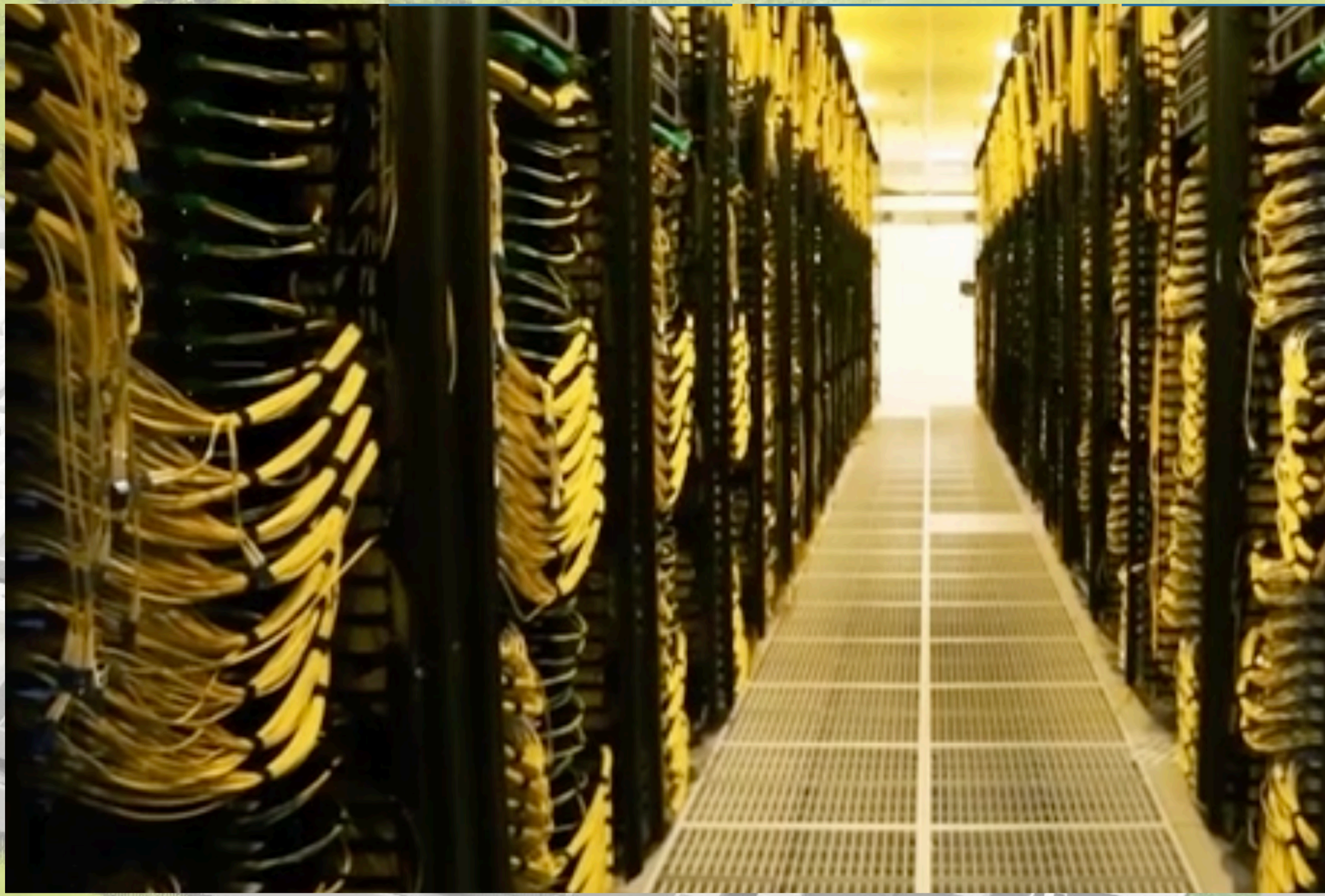
VMs vs. Containers

- Why use VMs?
  - Full control (custom OS, etc.)
  - Need to execute other containers
  - Running 3rd party or vendor software that doesn’t support containers
  - Need to save data persistently
    - But beware Single Point of Failure !

Datacenters

What is a Datacenter?

- A datacenter is a physical location with 1000s (sometimes 100,000s) of physical machines.



Virtual Machines

VMs vs. Containers

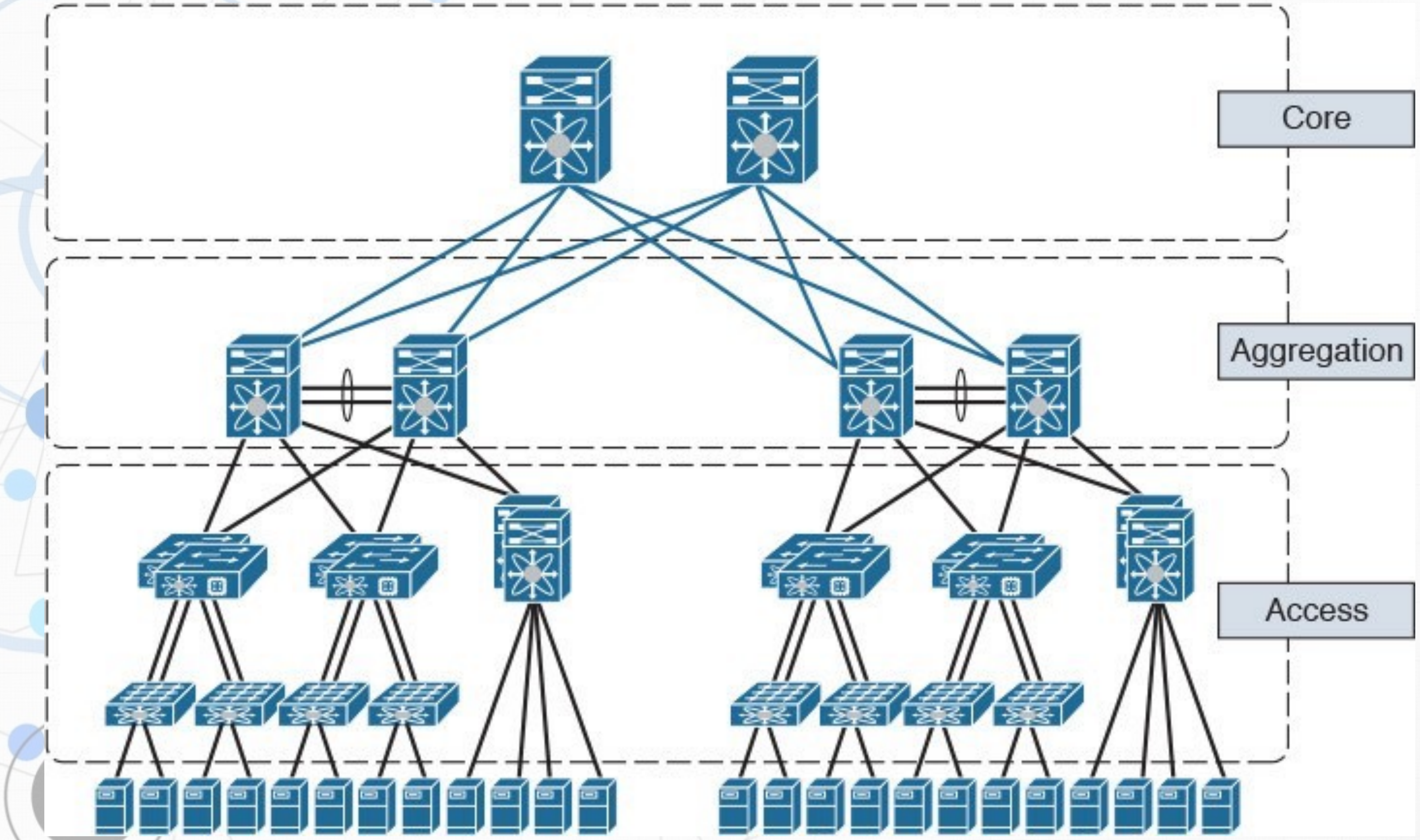
- Many resources are shared
- Power Network A/C
- Massive economies of scale

https://aws.amazon.com/compliance/data-center/data-centers/

Virtual Machines

VMs vs. Containers

- Datacenters have complex, high performance networks
- Individual servers organized into groups, to provide a variety of services.



Virtual Machines

VMs vs. Containers

- It's easy to allocate a handful of VMs, spread across the datacenter, to implement some new function.

- If the VMs are small, and few, then the cost is ***pretty close to zero***.

Cloud Services

Infrastructure as a Service (IaaS)

- **Virtual machines and datacenters** make it cost-effective to create new, small machines.
  - Run as many as you want
  - Prototype on a small machine, move to a large machine later (easily)
  - Bring up new machines in minutes
  - Shut down machines easily (to save cost)

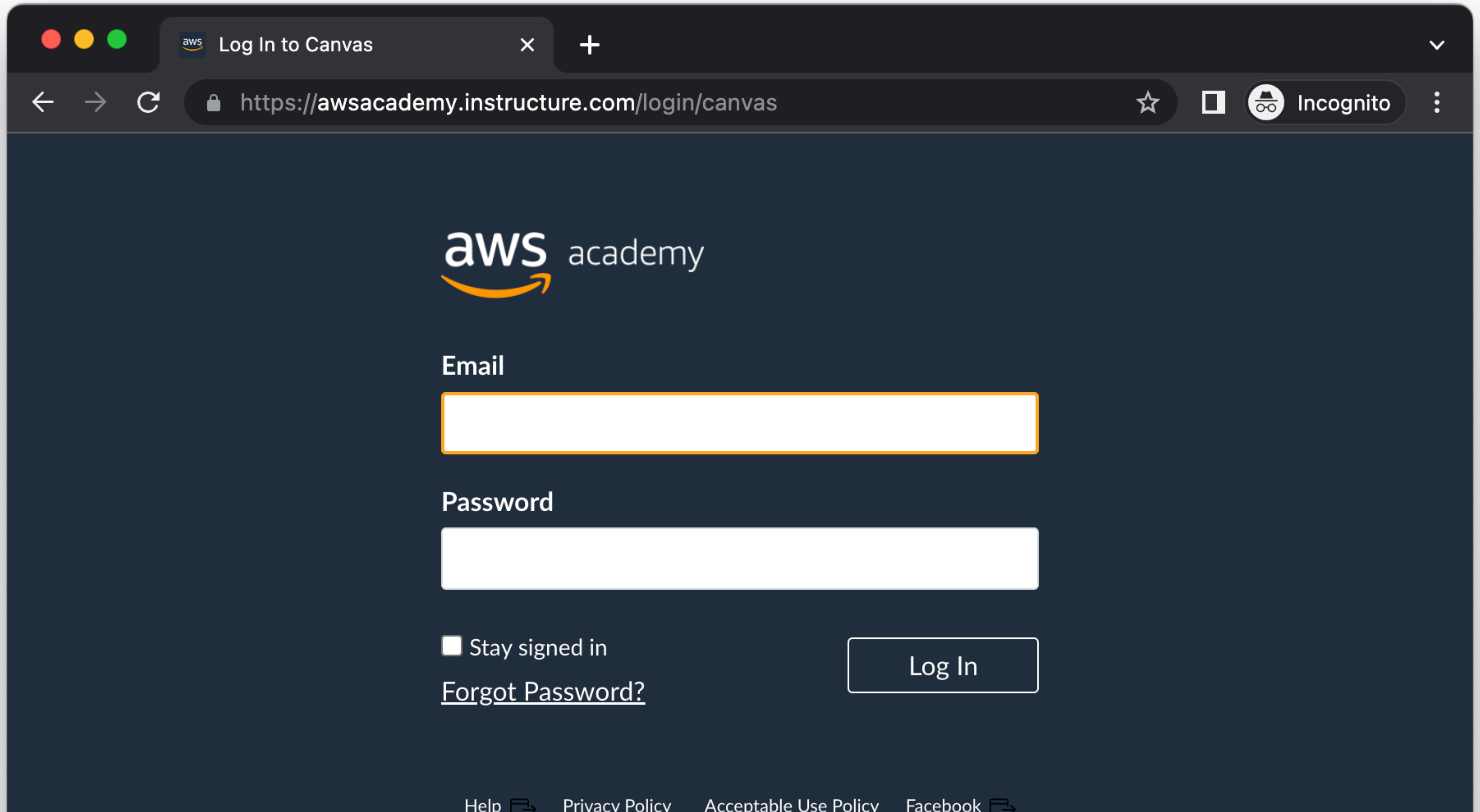
Cloud Services

<THING> as a Service

- **IaaS** (Infrastructure as a Service)
  - They sell you a VM, config as you wish
- **PaaS** (Platform as a Service)
  - They sell you a runtime environment, upload code
- **SaaS** (Software as a Service)
  - They sell you a service, connect to it as needed

AWS Console

- See last week's slides for access to AWS Academy
  - Log in at https://awsacademy.instructure.com/login/canvas



Log In to Canvas

https://awsacademy.instructure.com/login/canvas

aws academy

Email

Password

□ Stay signed in

Forgot Password?

Log In

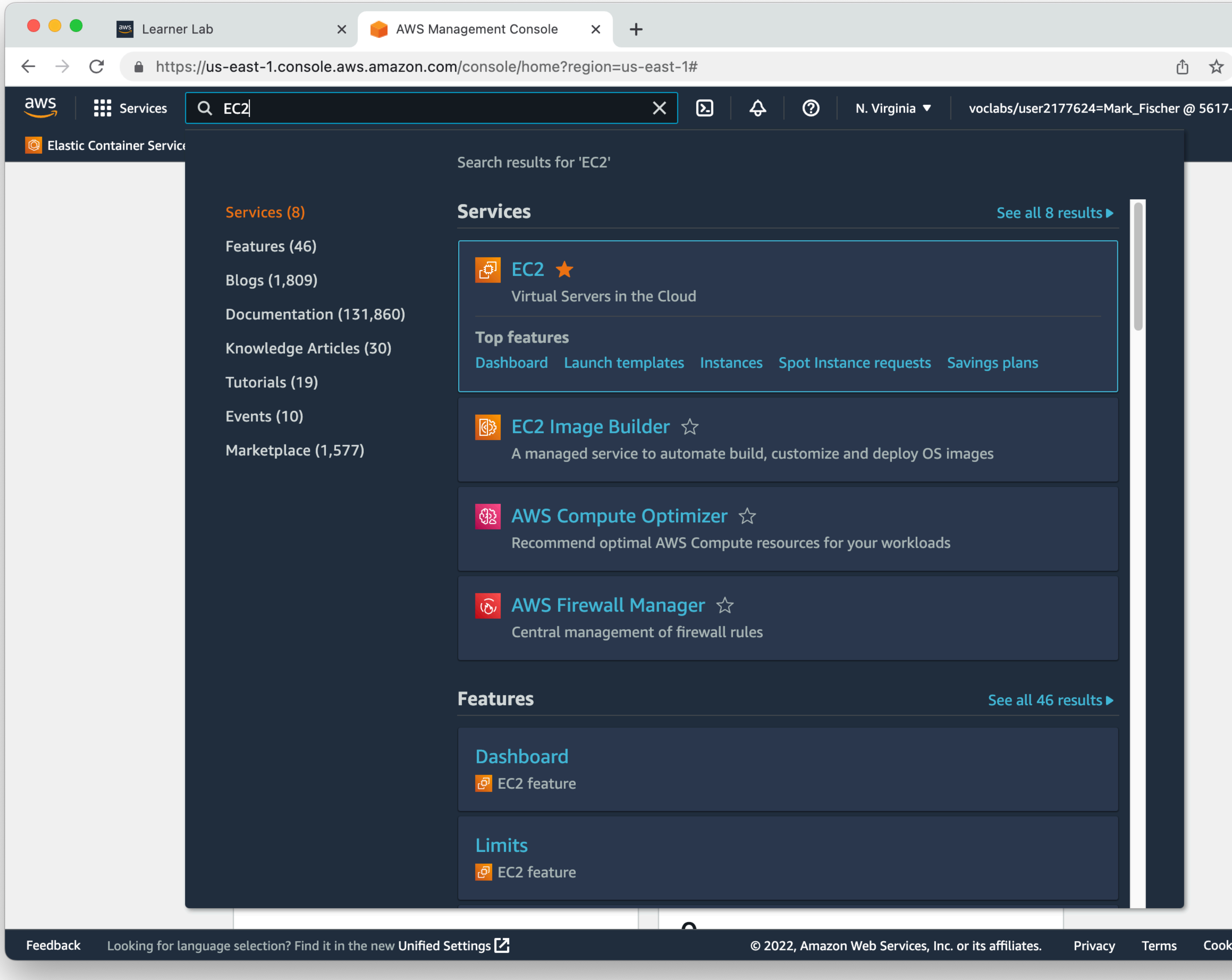
Help Privacy Policy Acceptable Use Policy Facebook

AWS EC2

- EC2 (Elastic Compute Cloud) is Amazon's IaaS offering
    - Feel free to investigate others on your own time
- Lots of flexibility
    - Multiple CPU architectures
    - Multiple OSes
    - Dozens of different memory/CPU combinations
    - Lots of automation to make it easy to manage

EC2

- Begin by searching for EC2 in the services search bar
- Feel free to star the service to keep it in the AWS favorites bar



Learner Lab

AWS Management Console

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

Search results for 'EC2'

Services

See all 8 results ▶

EC2 ★
Virtual Servers in the Cloud

Top features
Dashboard Launch templates Instances Spot Instance requests Savings plans

EC2 Image Builder ☆
A managed service to automate build, customize and deploy OS images

AWS Compute Optimizer ☆
Recommend optimal AWS Compute resources for your workloads

AWS Firewall Manager ☆
Central management of firewall rules

Features

See all 46 results ▶

Dashboard
EC2 feature

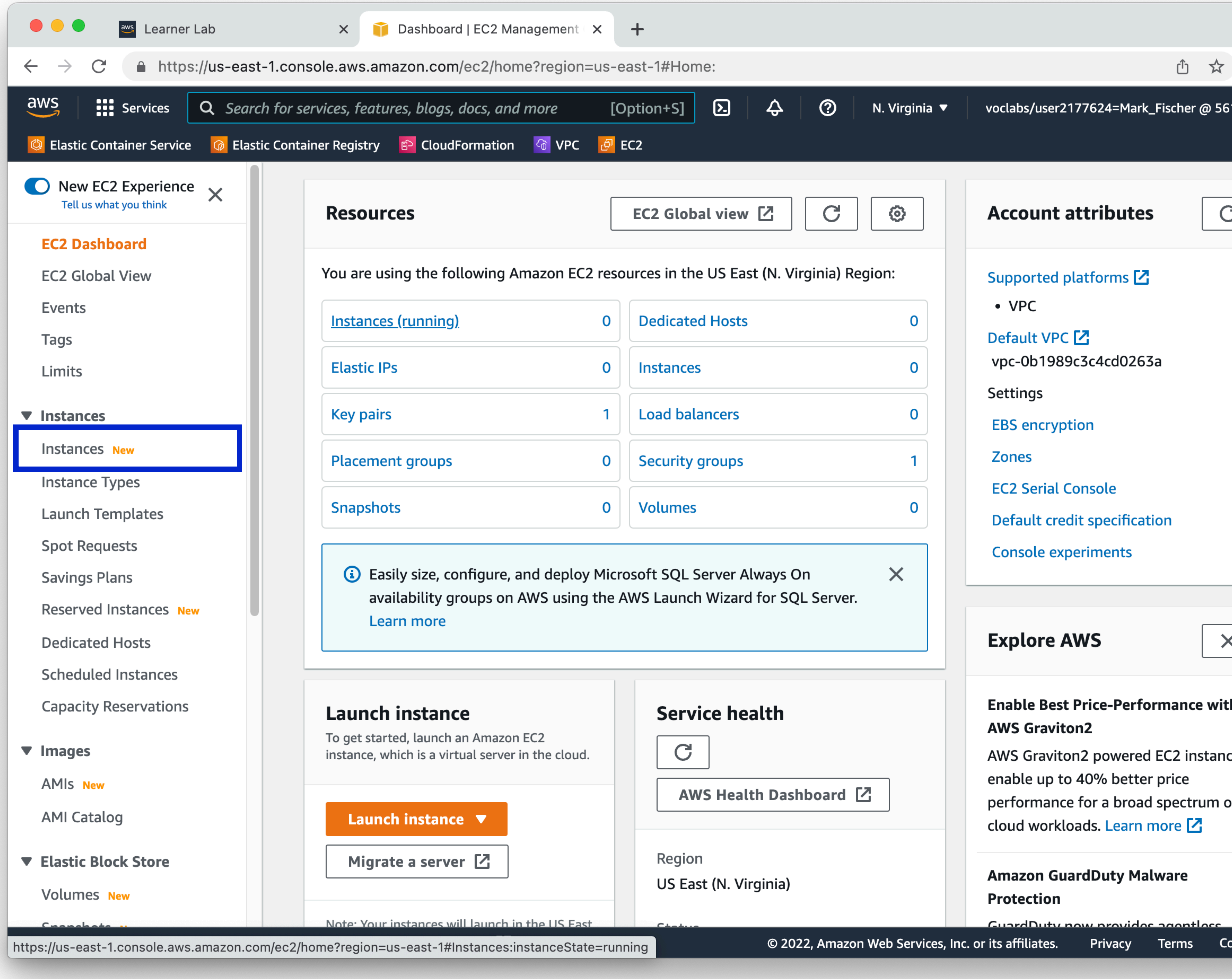
Limits
EC2 feature

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

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

EC2

- From the main EC2 console, click on Instances in the left sidebar



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

Instances (running)	0	Dedicated Hosts	0
Elastic IPs	0	Instances	0
Key pairs	1	Load balancers	0
Placement groups	0	Security groups	1
Snapshots	0	Volumes	0

Easily size, configure, and deploy Microsoft SQL Server Always On availability groups on AWS using the AWS Launch Wizard for SQL Server. Learn more

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Note: Your instances will launch in the US East

Enable Best Price-Performance with AWS Graviton2

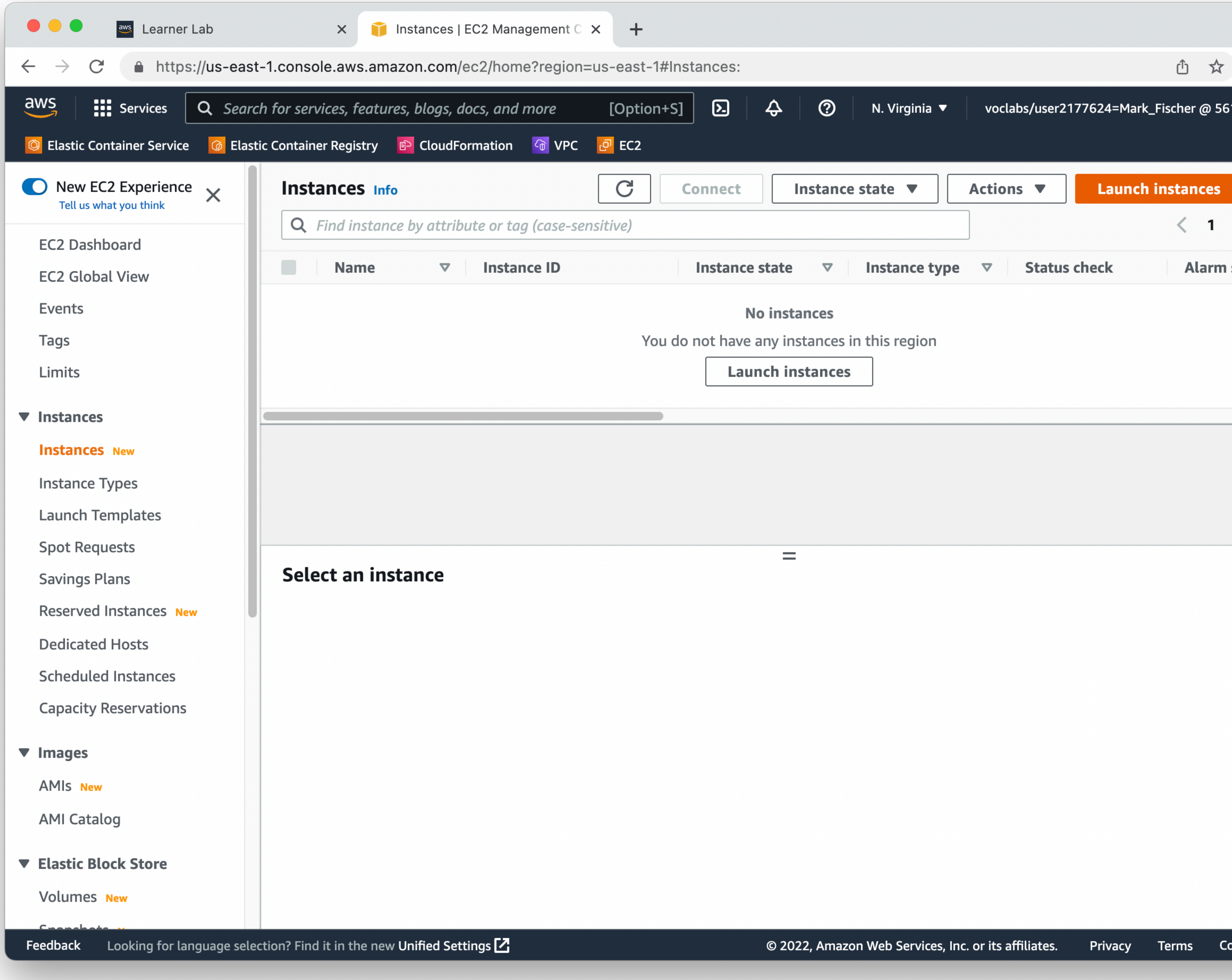
AWS Graviton2 powered EC2 instances enable up to 40% better price performance for a broad spectrum of cloud workloads. Learn more

Amazon GuardDuty Malware Protection

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

EC2

- Starting out you won’t have any instances, but if you did, they would show up here
- Running and stopped instances
- Stopped instances don’t cost you compute time, but still cost you for the storage
- Click “**Launch Instances**”



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

No instances

You do not have any instances in this region

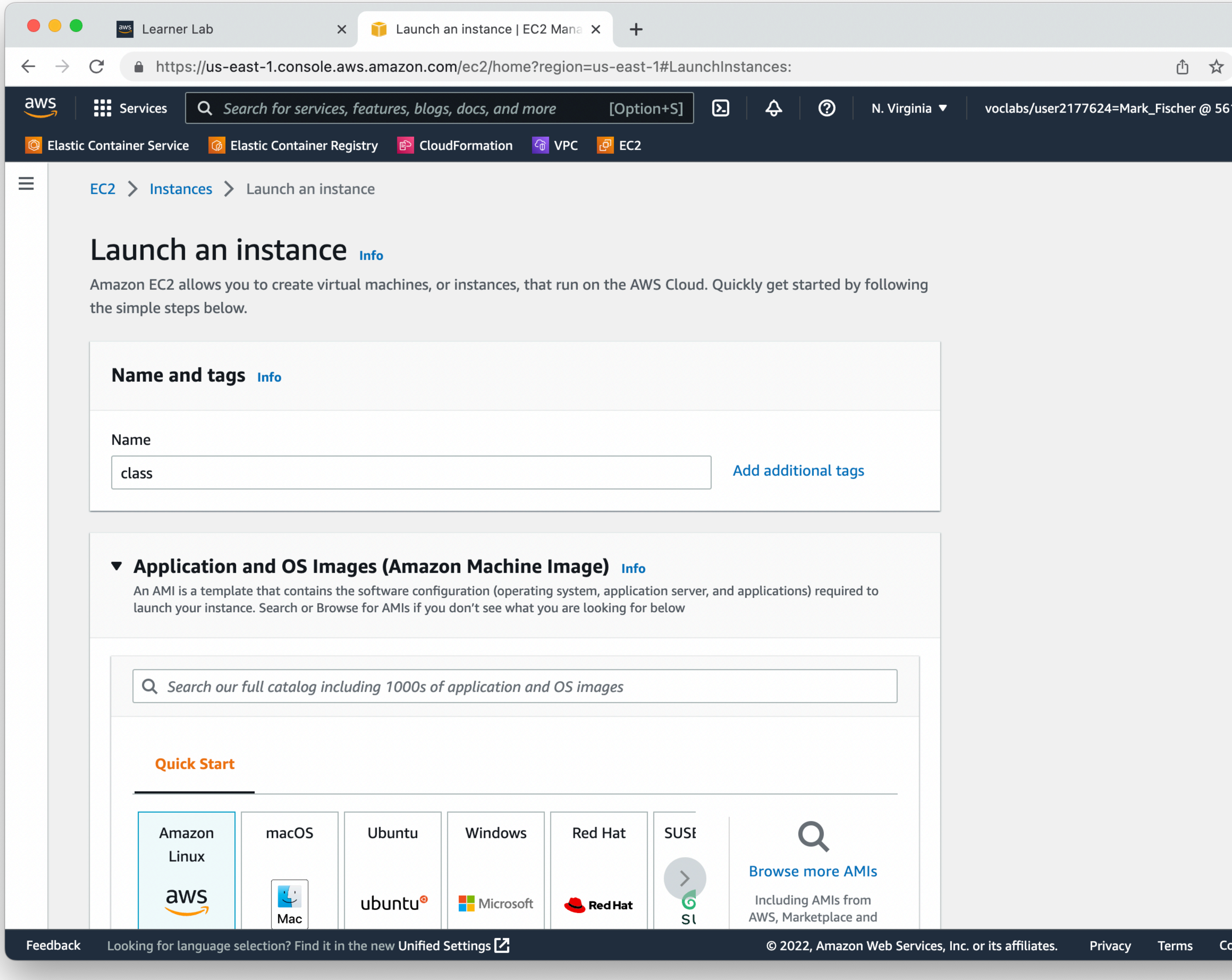
Select an instance

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

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

EC2

- We’ll pretty much accept the defaults
- Give your instance a Name



Learner Lab
Launch an instance | EC2 Mana
https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances:
aws
Services
Search for services, features, blogs, docs, and more
[Option+S]
N. Virginia ▼
voclabs/user2177624=Mark_Fischer @ 56
Elastic Container Service
Elastic Container Registry
CloudFormation
VPC
EC2
EC2 > Instances > Launch an instance

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name: class

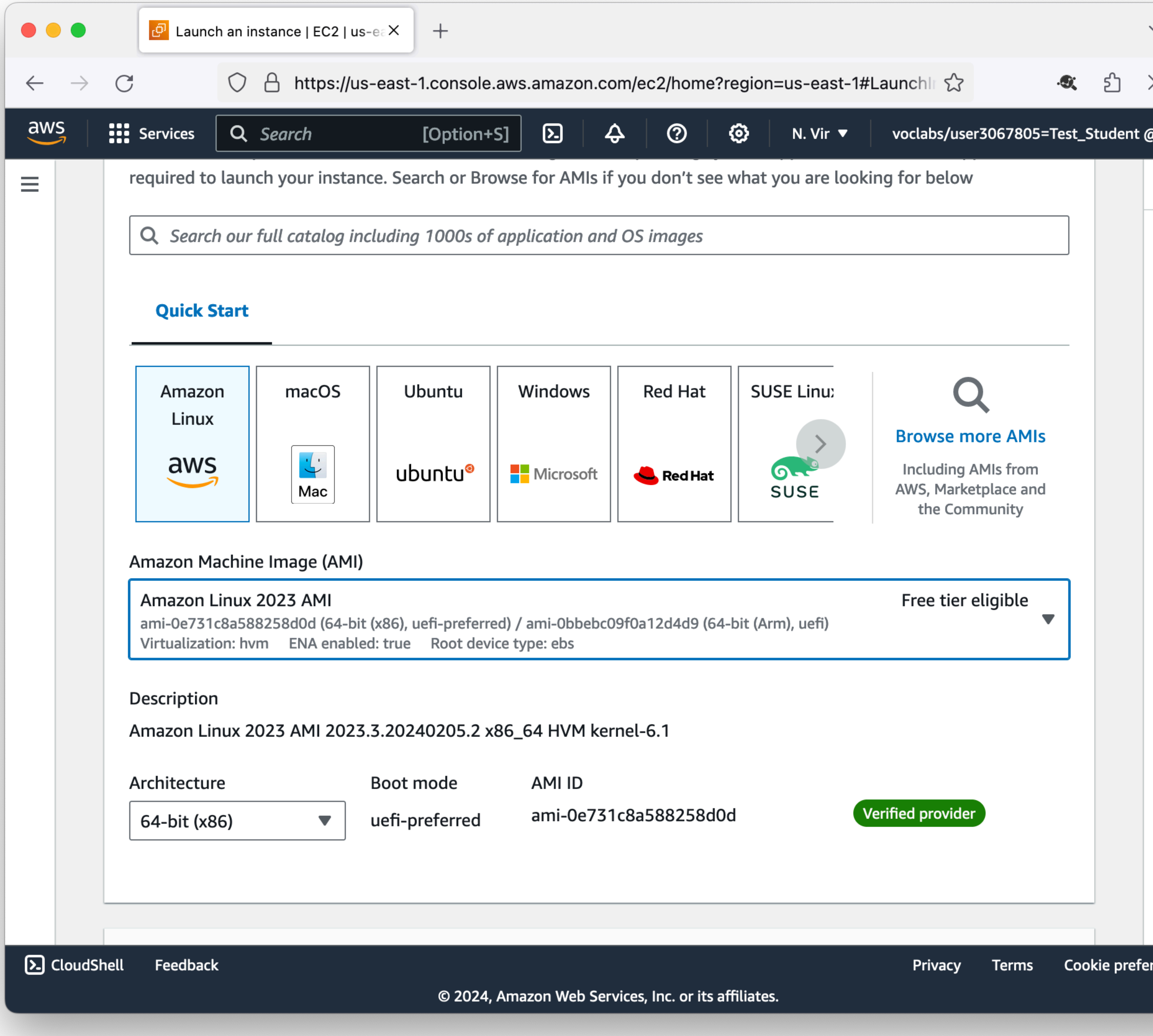
▼ Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don’t see what you are looking for below

Search our full catalog including 1000s of application and OS images
Quick Start
Amazon Linux
aws
macOS
Mac
Ubuntu
ubuntu
Windows
Microsoft
Red Hat
Red Hat
SUSE
Browse more AMIs
Including AMIs from AWS, Marketplace and
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms

EC2

- For the Instance OS, use Amazon Linux 2023, and the 64-bit (x86) architecture
- AWS also supports ARM
  - ARM support is really good, but there are still some rough edges
  - We'll stick with x86 for the class


Launch an instance | EC2 | us-e
https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchI
aws
Services
Search
[Option+S]
N. Vir
voclabs/user3067805=Test_Student @
Search our full catalog including 1000s of application and OS images
Quick Start
Amazon Linux
aws
macOS
Mac
Ubuntu
ubuntu
Windows
Microsoft
Red Hat
Red Hat
SUSE Linu:
SUSE
Browse more AMIs
Including AMIs from AWS, Marketplace and the Community
Verified provider
CloudShell
Feedback
Privacy
Terms
Cookie prefe

required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI — Free tier eligible
ami-0e731c8a588258d0d (64-bit (x86), uefi-preferred) / ami-0bbebc09f0a12d4d9 (64-bit (Arm), uefi)
Virtualization: hvm   ENA enabled: true   Root device type: ebs

Description

Amazon Linux 2023 AMI 2023.3.20240205.2 x86_64 HVM kernel-6.1

Architecture: 64-bit (x86)

Boot mode: uefi-preferred

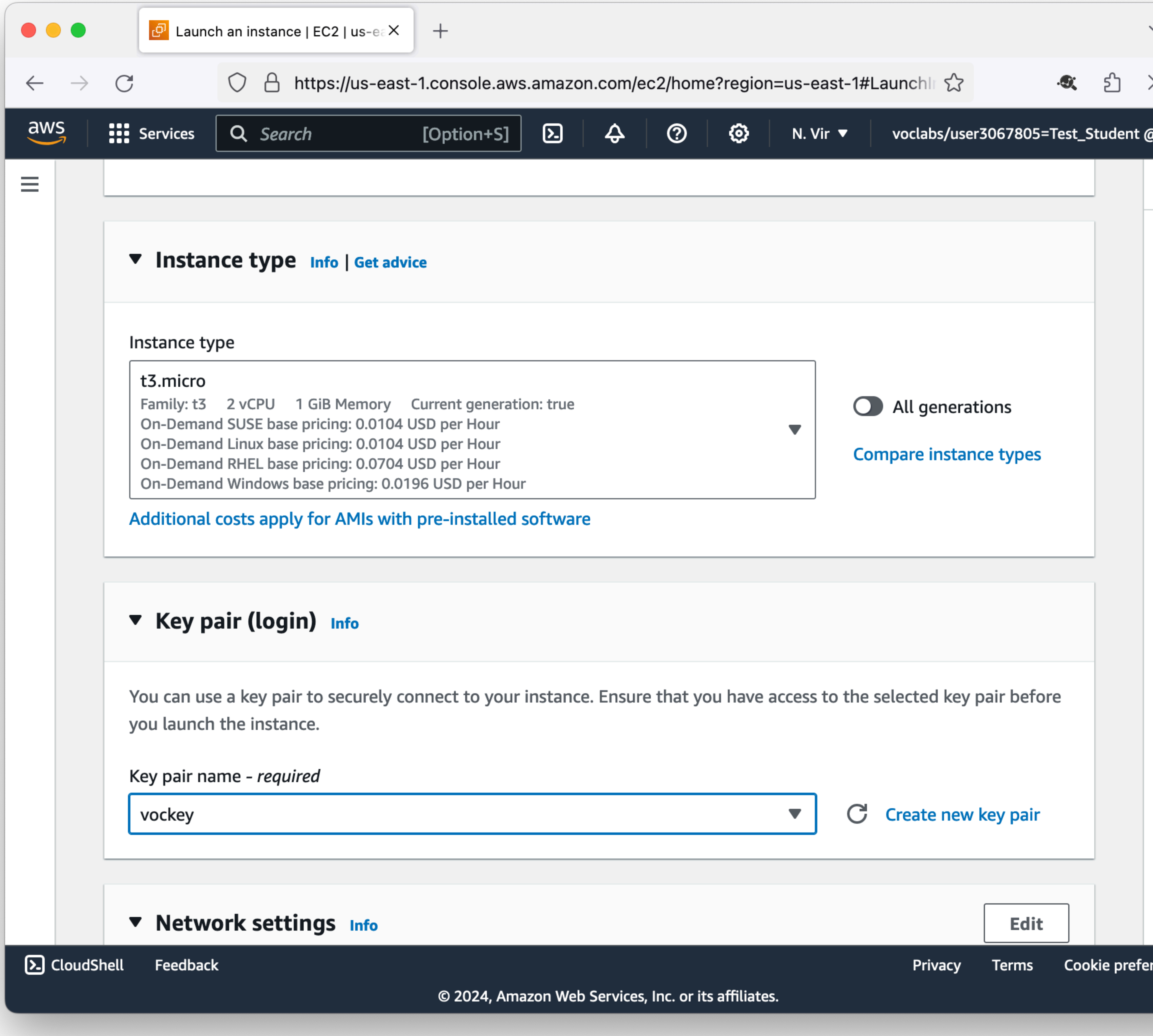
AMI ID: ami-0e731c8a588258d0d

© 2024, Amazon Web Services, Inc. or its affiliates.

EC2

- For the Instance Type, change to t3.micro. This will be plenty for our needs, and can be entirely free if configured correctly

- Be sure to choose the `vockey` Key Pair. This will be required to log in to your instance



Launch an instance | EC2 | us-e
https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchI
aws
Services
Search
[Option+S]
N. Vir
voclabs/user3067805=Test_Student @

Instance type Info | Get advice

Instance type

t3.micro
Family: t3 2 vCPU 1 GiB Memory Current generation: true
On-Demand SUSE base pricing: 0.0104 USD per Hour
On-Demand Linux base pricing: 0.0104 USD per Hour
On-Demand RHEL base pricing: 0.0704 USD per Hour
On-Demand Windows base pricing: 0.0196 USD per Hour

All generations

Compare instance types

Additional costs apply for AMIs with pre-installed software

Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

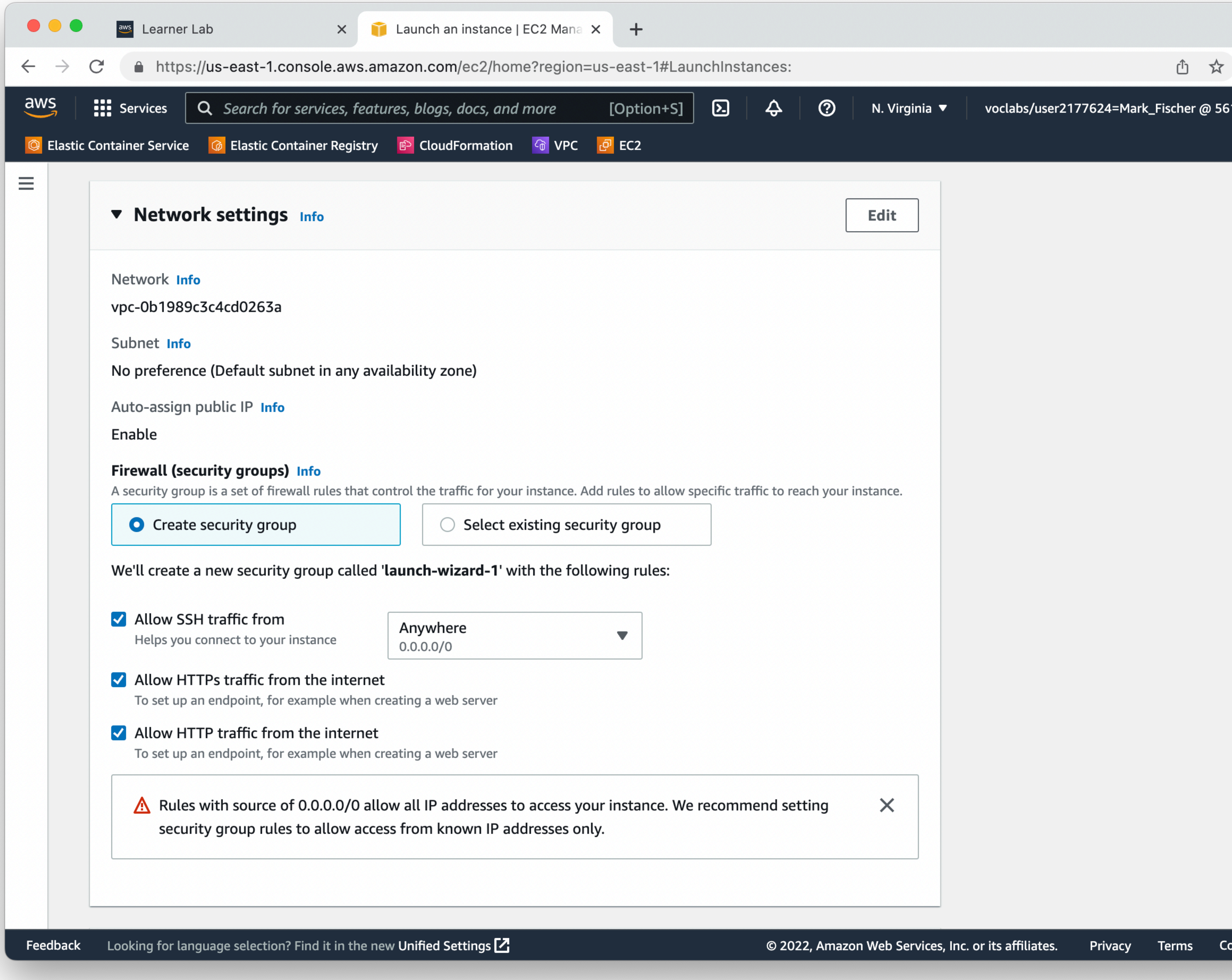
vockey

Create new key pair

Network settings Info

EC2

- In Network settings, create a new security group
- Allow SSH traffic from anywhere
    - Is it a good idea to allow SSH from anywhere? We'll discuss in a bit.
- Also allow HTTP and HTTPS traffic



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

▼ Network settings Info

Network Info
vpc-0b1989c3c4cd0263a

Subnet Info
No preference (Default subnet in any availability zone)

Auto-assign public IP Info
Enable

Firewall (security groups) Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

● Create security group ○ Select existing security group

We'll create a new security group called '**launch-wizard-1**' with the following rules:

☑ Allow SSH traffic from — Anywhere 0.0.0.0/0
Helps you connect to your instance

☑ Allow HTTPs traffic from the internet
To set up an endpoint, for example when creating a web server

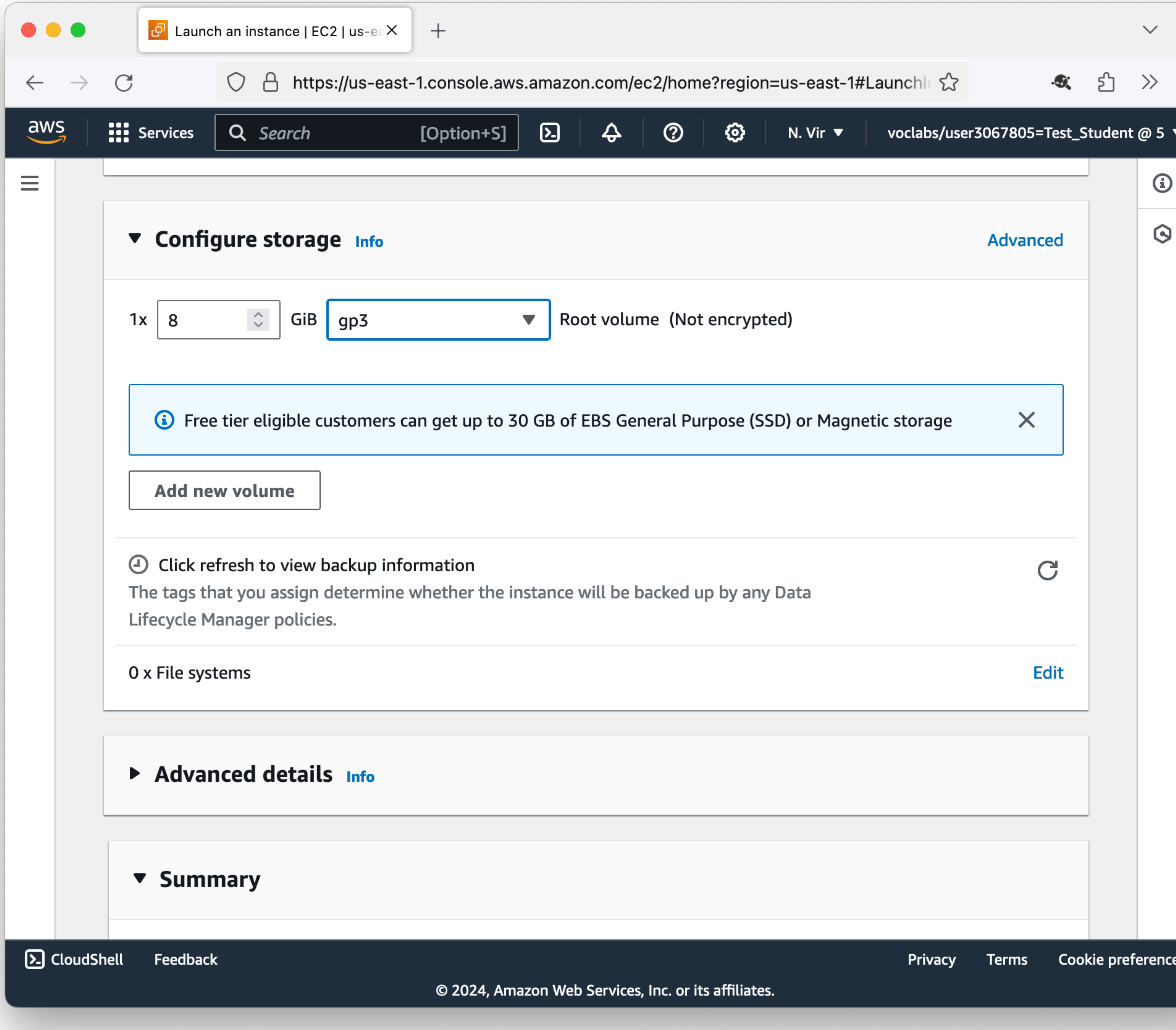
☑ Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

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

Learner Lab
Launch an instance | EC2 Mana
aws
Services
Search for services, features, blogs, docs, and more [Option+S]
N. Virginia
voclabs/user2177624=Mark_Fischer @ 56
Elastic Container Service
Elastic Container Registry
CloudFormation
VPC
EC2
Edit
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms

EC2

- For Storage, the default 8 GiB gp3 volume will be fine for our needs
  - gp3 is AWS General Purpose SSD storage
  - AWS offers many different storage types with better or worse performance and cost characteristics



Launch an instance | EC2 | us-ea
https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchI
aws
Services
Search
[Option+S]
N. Vir
voclabs/user3067805=Test_Student @ 5

Configure storage Info Advanced

1x 8 GiB gp3 Root volume (Not encrypted)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

Click refresh to view backup information
The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

0 x File systems Edit

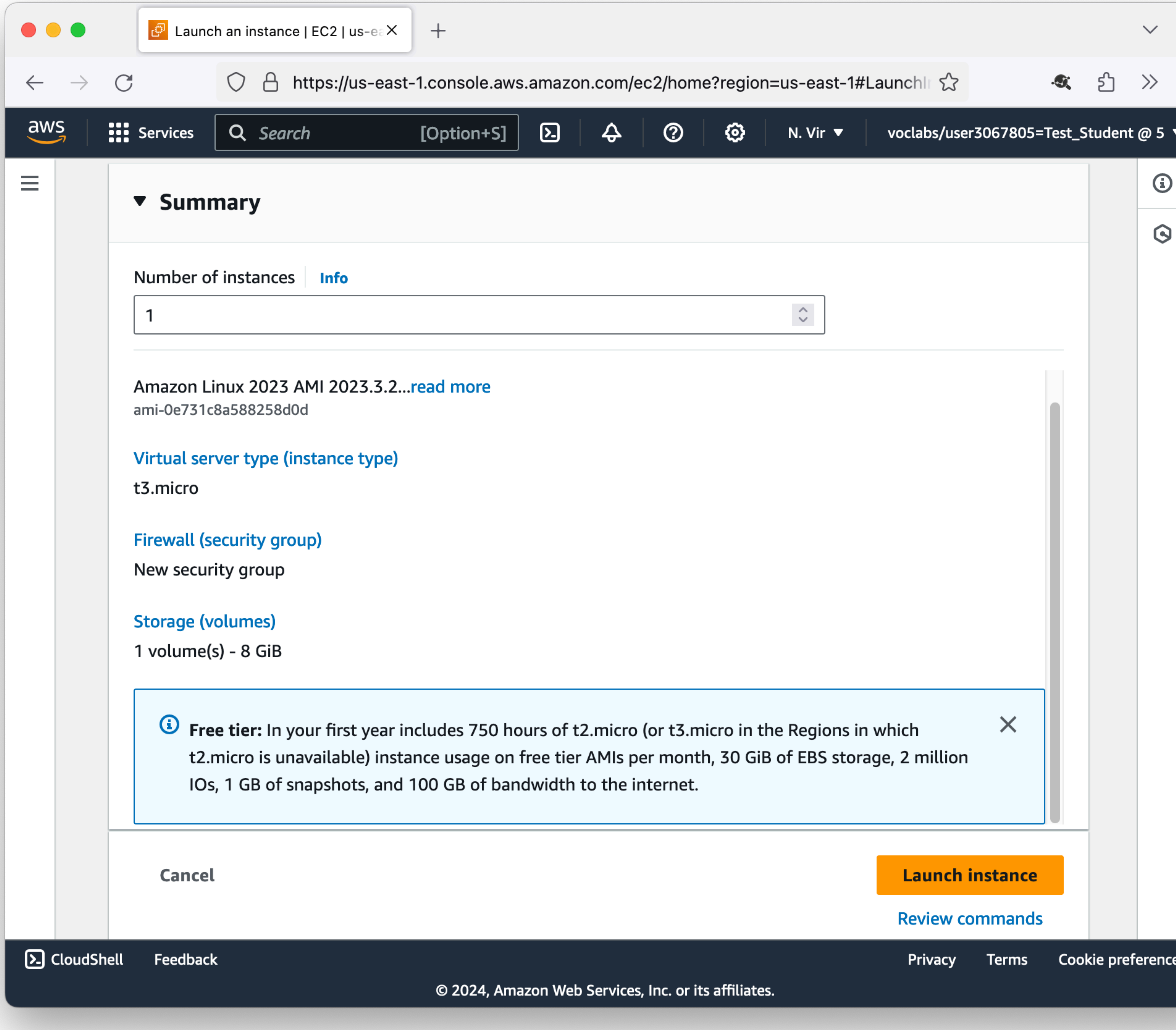
Advanced details Info

Summary

© 2024, Amazon Web Services, Inc. or its affiliates.

EC2

- Be sure to just make 1 instance 🤪
- Review your settings and then click
  Launch instance



Launch an instance | EC2 | us-e
https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchI
aws
Services
Search
[Option+S]
N. Vir
voclabs/user3067805=Test_Student @ 5

▼ **Summary**

Number of instances Info

1

Amazon Linux 2023 AMI 2023.3.2...read more
ami-0e731c8a588258d0d

Virtual server type (instance type)
t3.micro

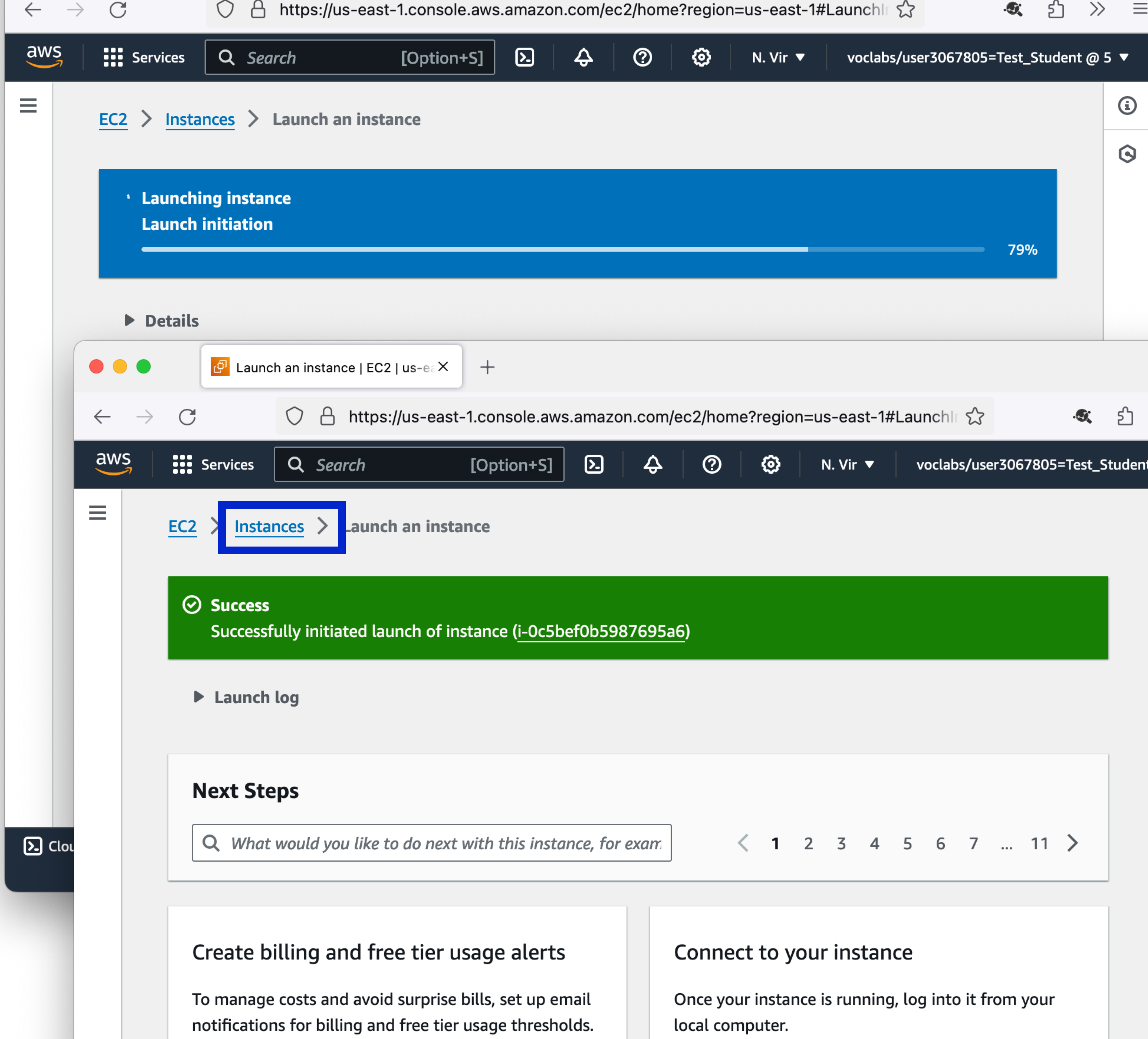
Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

EC2

- You’ll see a “Launching instance” progress bar first, followed by a Success page after a short while
- Click the “Instances” link in the breadcrumb trail above the Success banner to go back to the EC2 Instances console



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

aws
Services
Search [Option+S]
N. Vir
voclabs/user3067805=Test_Student @ 5
EC2 > Instances > Launch an instance

Launching instance
Launch initiation
79%

Details
Launch an instance | EC2 | us-e
Clou

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

aws
Services
Search [Option+S]
N. Vir
voclabs/user3067805=Test_Student
EC2 > Instances > Launch an instance

Success
Successfully initiated launch of instance (i-0c5bef0b5987695a6)

Next Steps

What would you like to do next with this instance, for exam

Create billing and free tier usage alerts

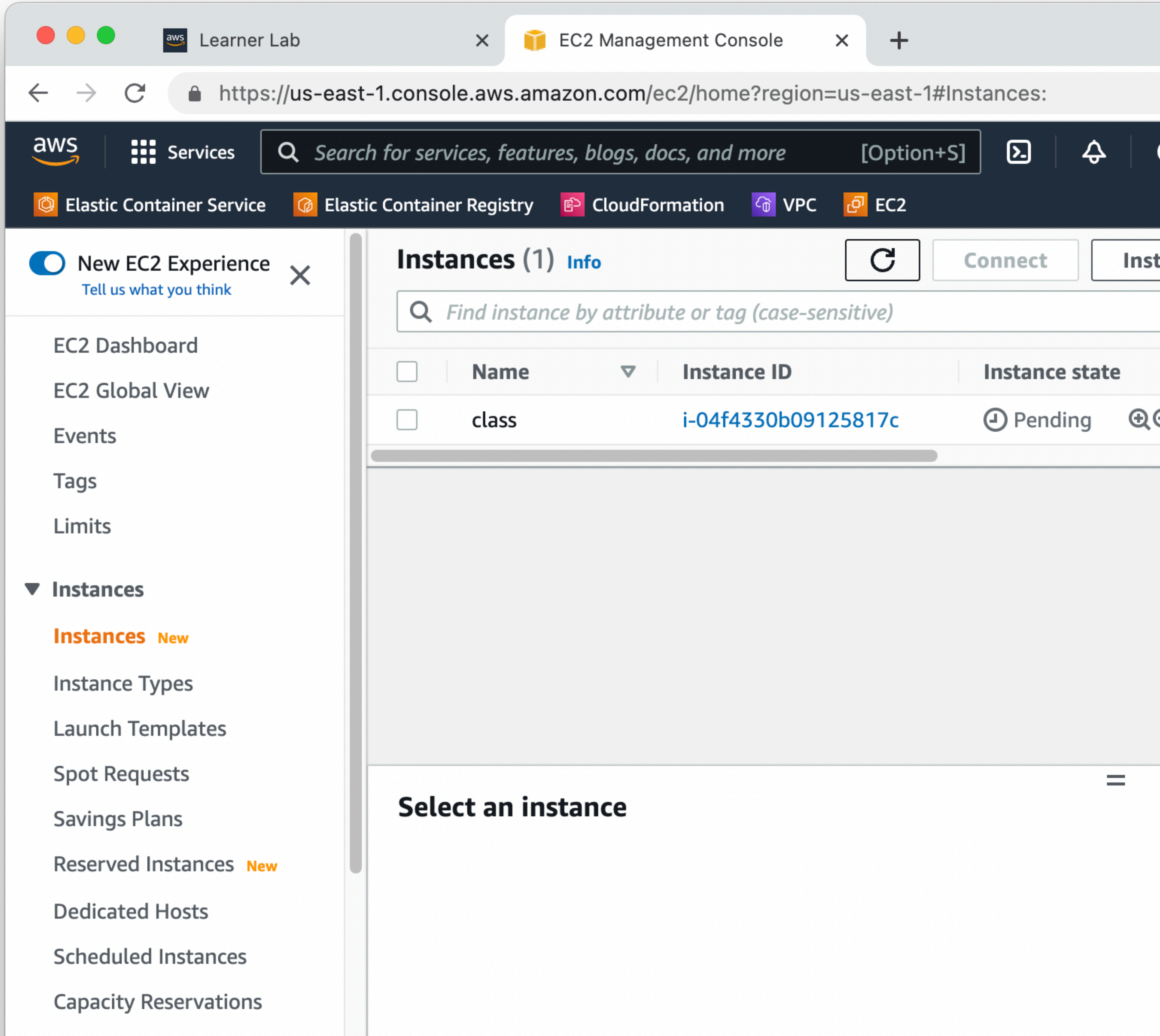
To manage costs and avoid surprise bills, set up email notifications for billing and free tier usage thresholds.

Connect to your instance

Once your instance is running, log into it from your local computer.

EC2

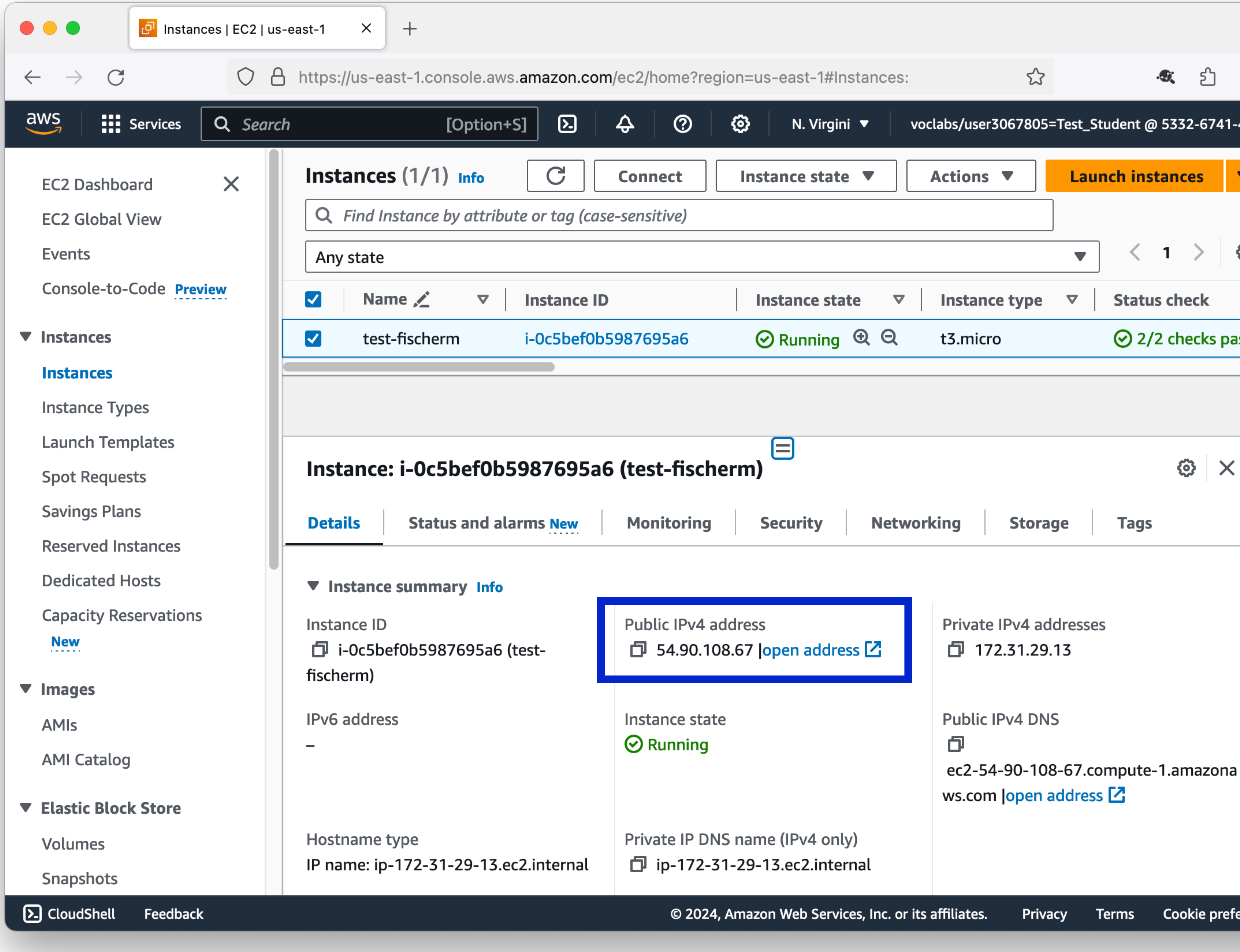
- Your new instance will take just a minute or two to start up
  - You'll see the Instance State as "Pending", then "Starting Up" and finally "Running"



□	Name	Instance ID	Instance state
□	class	i-04f4330b09125817c	Pending

EC2

- Always a good idea to wait for the Status Checks to come back as 2/2 checks passed
  - Very rarely these checks fail, and your instance ends up in a bad state
  - The cloud is not perfect!
- Copy the Public IP



☑	Name	Instance ID	Instance state	Instance type	Status check
☑	test-fischerm	i-0c5bef0b5987695a6	✓ Running	t3.micro	✓ 2/2 checks pa

Instance: i-0c5bef0b5987695a6 (test-fischerm)

Instance ID: i-0c5bef0b5987695a6 (test-fischerm)

Public IPv4 address: 54.90.108.67 | open address

Private IPv4 addresses: 172.31.29.13

IPv6 address: –

Instance state: ✓ Running

Public IPv4 DNS: ec2-54-90-108-67.compute-1.amazonaws.com | open address

Hostname type: IP name: ip-172-31-29-13.ec2.internal

Private IP DNS name (IPv4 only): ip-172-31-29-13.ec2.internal

EC2

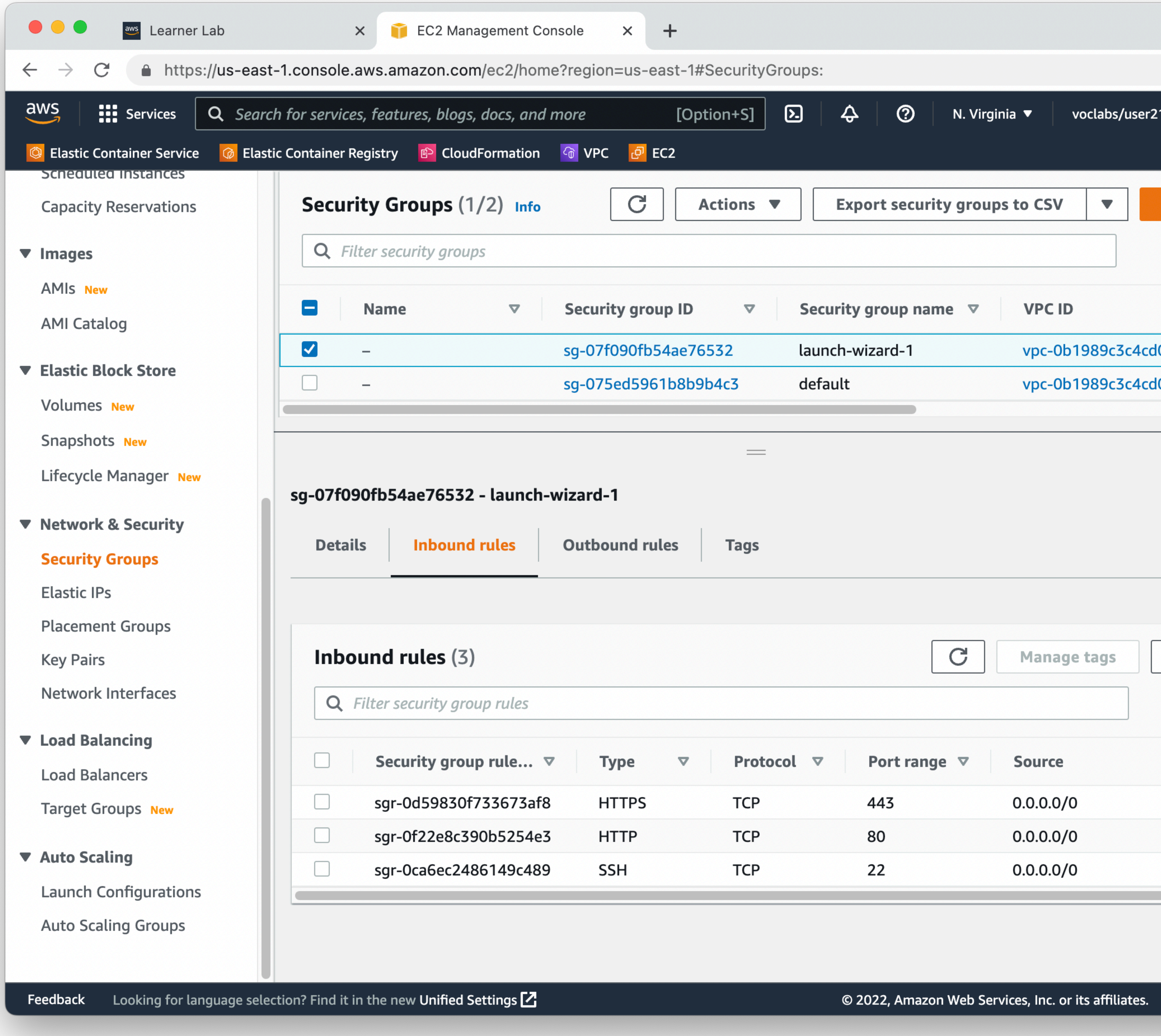
Stop vs. Terminate

- When you **stop** an instance, you take it offline but retain the resources
  - Can **start** it again anytime
  - Warning! You still have to pay for the EBS volume
- When you **terminate** an instance, you destroy everything
  - Danger! Your EBS volume may be destroyed.

EC2

Security Groups

- From the main EC2 Console page, choose Security Groups under Network & Security
- Your EC2 instance has a Public IP address
- When we set up the instance we created a new security group
- This allows incoming traffic on port 22, 80, and 443
- `0.0.0.0/0` means “anywhere”



Security Groups (1/2) Info

	Name	Security group ID	Security group name	VPC ID
☑	–	sg-07f090fb54ae76532	launch-wizard-1	vpc-0b1989c3c4cd
□	–	sg-075ed5961b8b9b4c3	default	vpc-0b1989c3c4cd

sg-07f090fb54ae76532 - launch-wizard-1

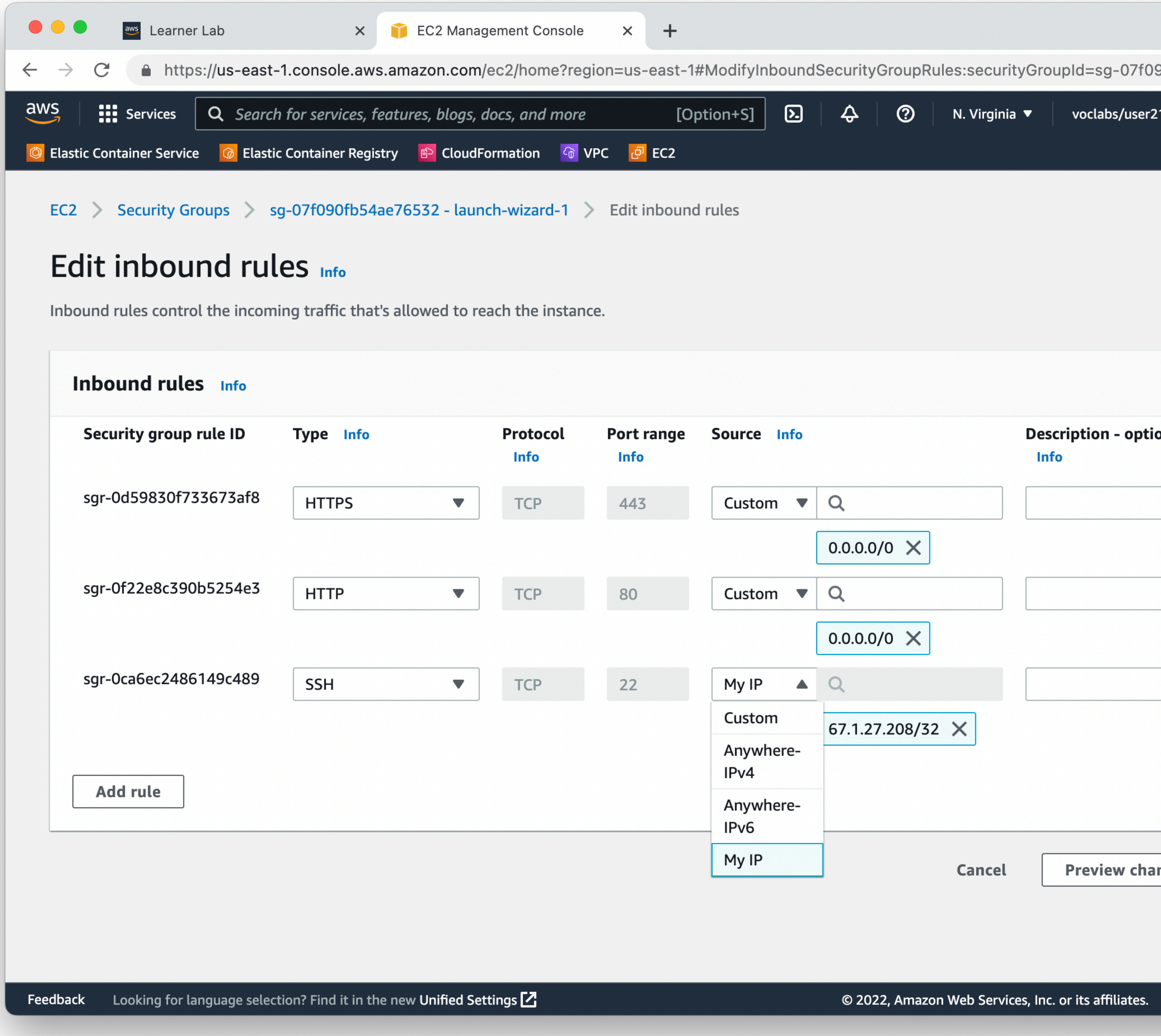
Inbound rules (3)

	Security group rule...	Type	Protocol	Port range	Source
□	sgr-0d59830f733673af8	HTTPS	TCP	443	0.0.0.0/0
□	sgr-0f22e8c390b5254e3	HTTP	TCP	80	0.0.0.0/0
□	sgr-0ca6ec2486149c489	SSH	TCP	22	0.0.0.0/0

EC2

Security Groups

- For publicly available services like HTTP and HTTPS, `0.0.0.0/0` is required
- For SSH however, allowing connections from anywhere can be a security risk
- It’s an acceptable risk for this class, since instances can only be running for 4 hours at a time
- For production instances, you would want to limit access



https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#ModifyInboundSecurityGroupRules:securityGroupId=sg-07f09

EC2 > Security Groups > sg-07f090fb54ae76532 - launch-wizard-1 > Edit inbound rules

Edit inbound rules

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules

Security group rule ID	Type	Protocol	Port range	Source	Description - optio
sgr-0d59830f733673af8	HTTPS	TCP	443	Custom 0.0.0.0/0	
sgr-0f22e8c390b5254e3	HTTP	TCP	80	Custom 0.0.0.0/0	
sgr-0ca6ec2486149c489	SSH	TCP	22	My IP 67.1.27.208/32	

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

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

EC2 Security

Why Security Groups?

- Security Groups are similar to Firewalls
- Good Security Group rules make you vulnerable to fewer attackers
- Principle of least privilege
    - Everything is blocked by default
    - Open up only what you need
- Principle of defense in depth
    - Don’t just rely on Security Groups or Firewalls
    - Certificates instead of Passwords, Keep OS patched, etc.

EC2 Security

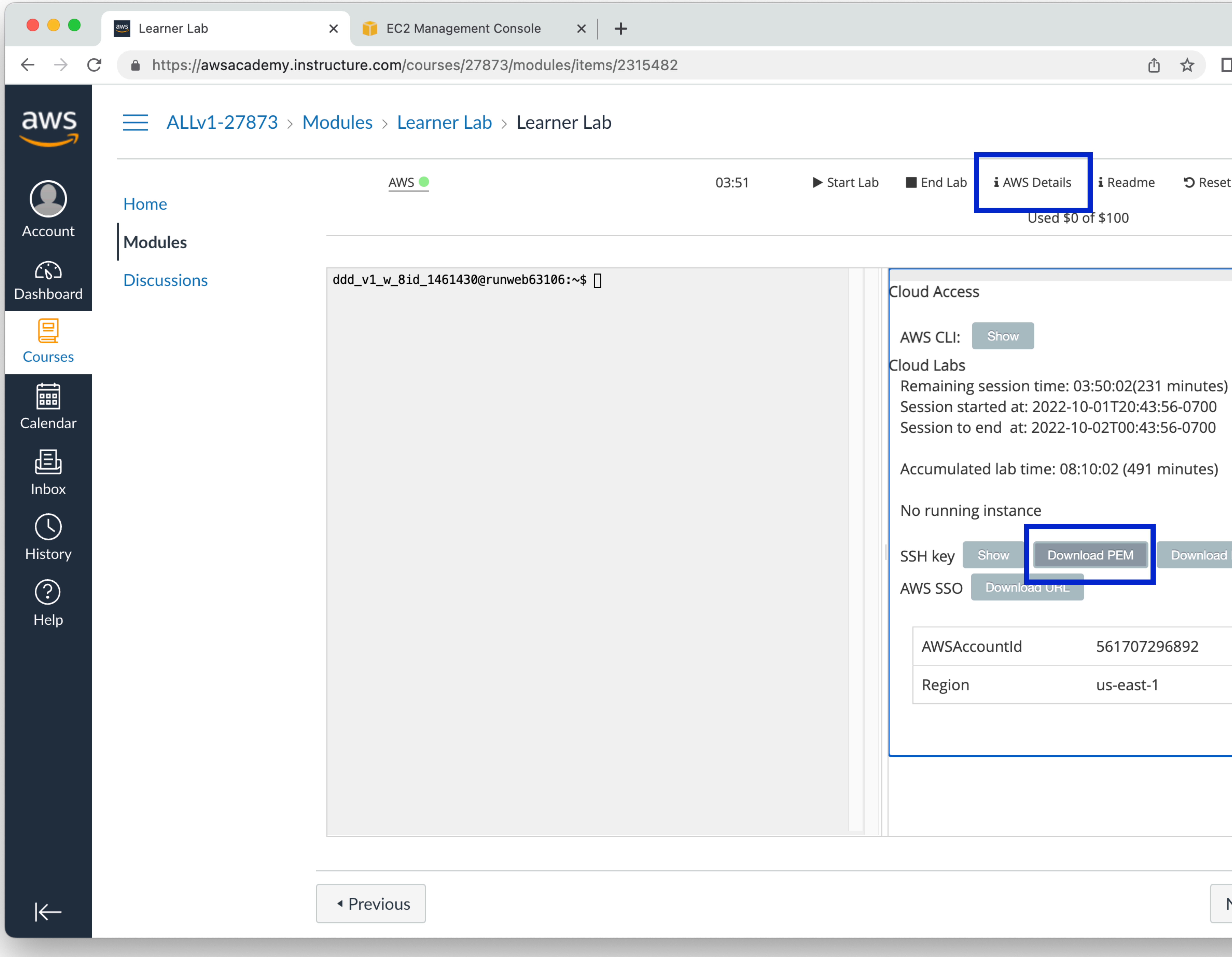
Connecting with SSH

- AWS EC2 Instances disable Password Authentication by default
  - Require Certificate based authentication
  - Effectively eliminates brute-force attacks
  - Attacker needs to have your certificate private key
  - Could still be vulnerable to bugs in SSH implementation itself
    - Keep your servers patched!
- This, coupled with the AWS Academy Lab limit of 4 hours per session means an acceptably low risk of attack against your VMs.
  - Risk is ***NOT Zero***. But it is very low, and acceptable.

EC2 Security

Connecting with SSH

- Back in the AWS Academy Lab in Canvas
- Click on AWS Details
- Download the PEM file
    - May need the PPK file if you are using Putty on Windows



```
ddd_v1_w_8id_1461430@runweb63106:~$ 
```

Cloud Access

AWS CLI:

Cloud Labs

Remaining session time: 03:50:02(231 minutes)
Session started at: 2022-10-01T20:43:56-0700
Session to end at: 2022-10-02T00:43:56-0700

Accumulated lab time: 08:10:02 (491 minutes)

No running instance

SSH key

AWS SSO

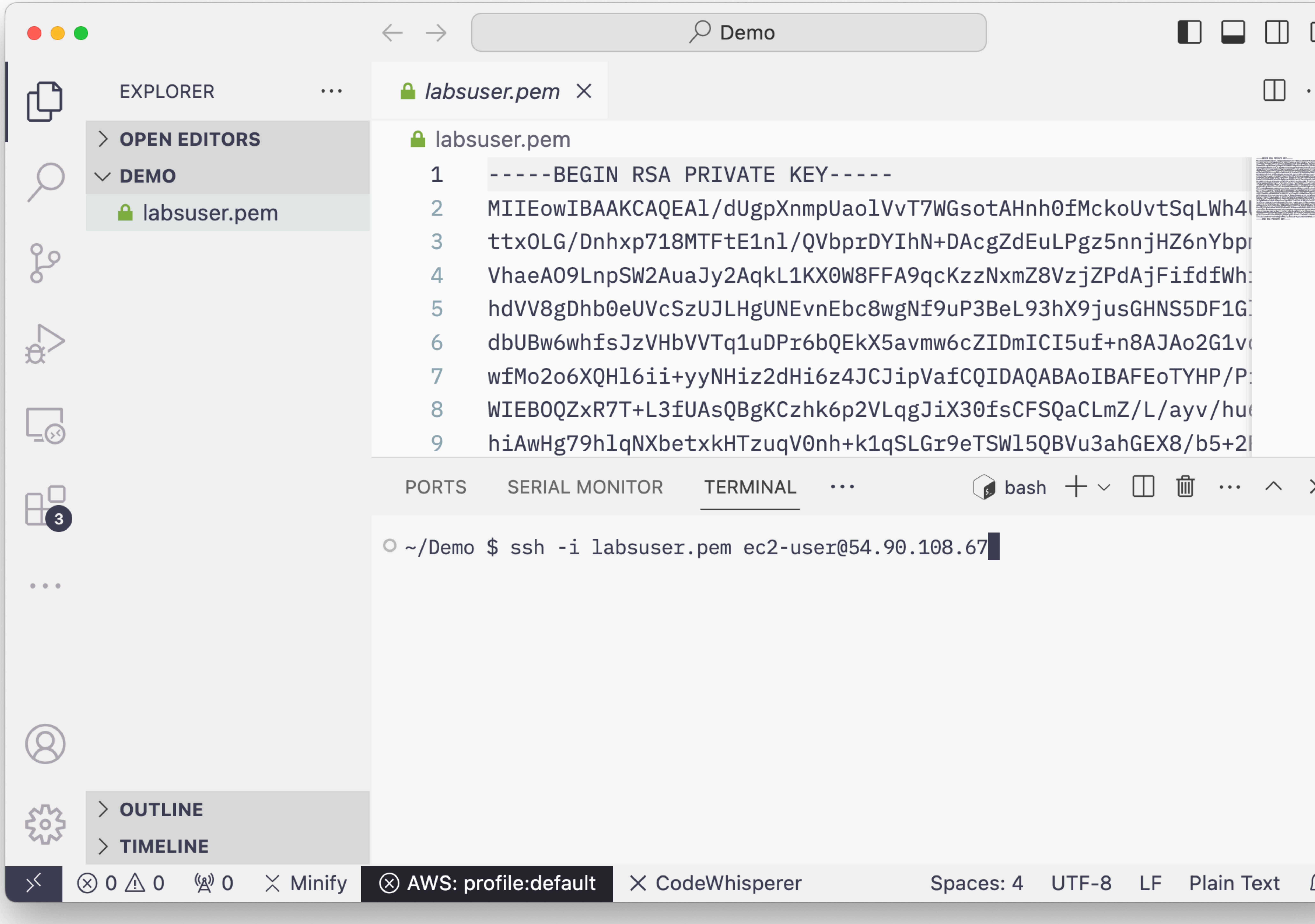
AWSAccountId	561707296892
Region	us-east-1

EC2 Security

Connecting with SSH

- The PEM file you download maybe named **`labsuser.pem`** like mine, or **`vokey.pem`** as described in the AWS documentation

- Contents of the key file will look something like this



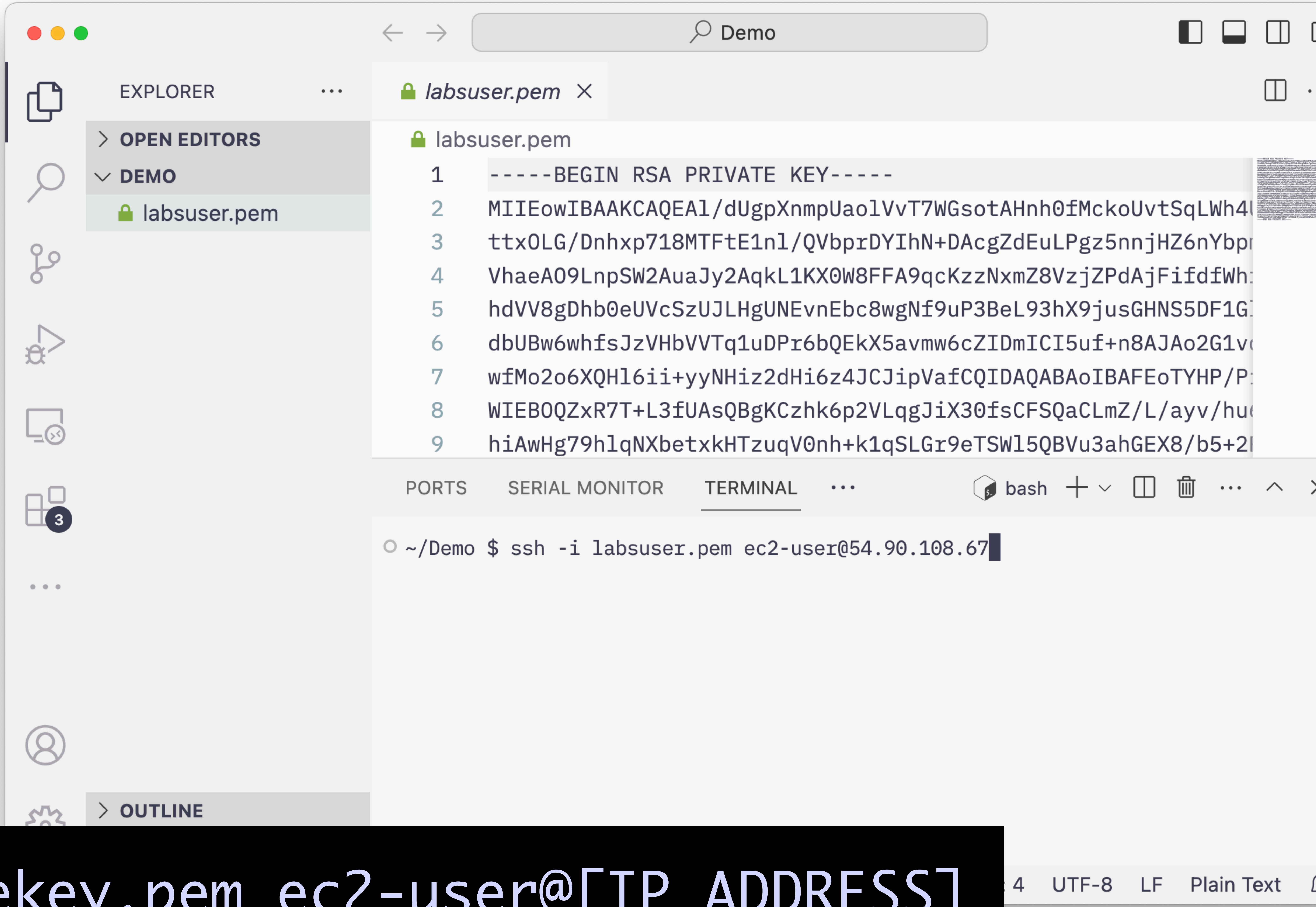
```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAl/dUgpXnmpUaolVvT7WGsotAHnh0fMckoUvtSqLWh4
ttxOLG/Dnhxp718MTFtE1nl/QVbprDYIhN+DAcgZdEuLPgz5nnjHZ6nYbp
VhaeAO9LnpSW2AuaJy2AqkL1KX0W8FFA9qcKzzNxmZ8VzjZPdAjFifdfWh
hdVV8gDhb0eUVcSzUJLHgUNEvnEbc8wgNf9uP3BeL93hX9jusGHNS5DF1G
dbUBw6whfsJzVHbVVTq1uDPr6bQEkX5avmw6cZIDmICI5uf+n8AJAo2G1v
wfMo2o6XQHl6ii+yyNHiz2dHi6z4JCJipVafCQIDAQABAoIBAFEoTYHP/P
WIEBOQZxR7T+L3fUAsQBgKCzhk6p2VLqgJiX30fsCFSQaCLmZ/L/ayv/hu
hiAwHg79hlqNXbetxkHTzuqV0nh+k1qSLGr9eTSWl5QBVu3ahGEX8/b5+2
```

```
~/Demo $ ssh -i labsuser.pem ec2-user@54.90.108.67
```

EC2 Security

Connecting with SSH

- For macOS and Linux, you can use the built-in ssh client
- Can use either the IP address or hostname of your instance
- Amazon Linux default user is **`ec2-user`**



```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAl/dUgpXnmpUaolVvT7WGsotAHnh0fMckoUvtSqLWh4
ttxOLG/Dnhxp718MTFtE1nl/QVbprDYIhN+DAcgZdEuLPgz5nnjHZ6nYbp
VhaeAO9LnpSW2AuaJy2AqkL1KX0W8FFA9qcKzzNxmZ8VzjZPdAjFifdfWh
hdVV8gDhb0eUVcSzUJLHgUNEvnEbc8wgNf9uP3BeL93hX9jusGHNS5DF1G
dbUBw6whfsJzVHbVVTq1uDPr6bQEkX5avmw6cZIDmICI5uf+n8AJAo2G1v
wfMo2o6XQHl6ii+yyNHiz2dHi6z4JCJipVafCQIDAQABAoIBAFEoTYHP/P
WIEBOQZxR7T+L3fUAsQBgKCzhk6p2VLqgJiX30fsCFSQaCLmZ/L/ayv/hu
hiAwHg79hlqNXbetxkHTzuqV0nh+k1qSLGr9eTSWl5QBVu3ahGEX8/b5+2
```

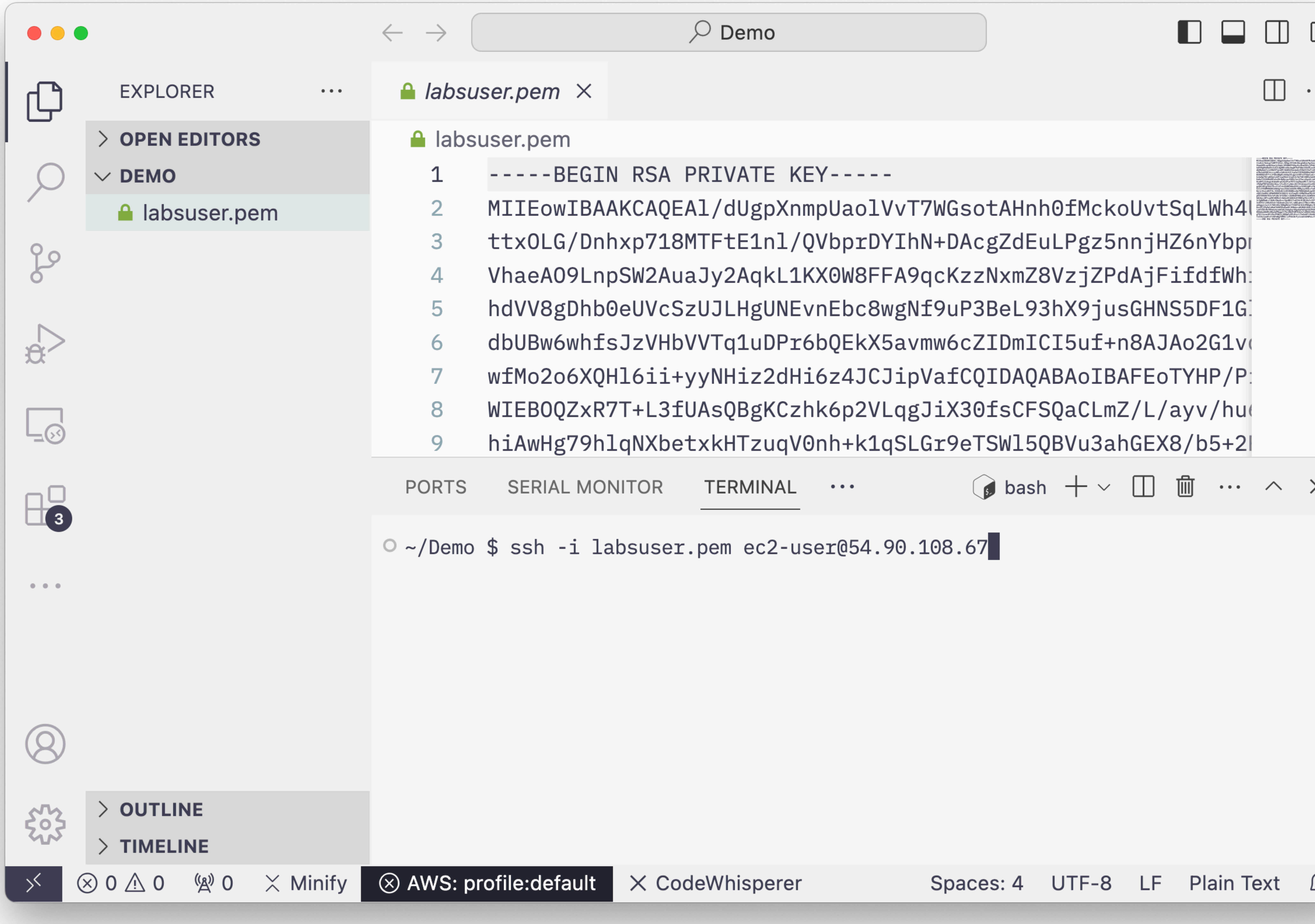
```
~/Demo $ ssh -i labsuser.pem ec2-user@54.90.108.67
```

```
ssh -i privatekey.pem ec2-user@[IP ADDRESS]
```

EC2 Security

Connecting with SSH

- Windows 10 should have the SSH client installed by default.
- If you do need to install it:
  - Open **Settings**, select **Apps**, then select **Optional Features**.
  - Install **OpenSSH Client**



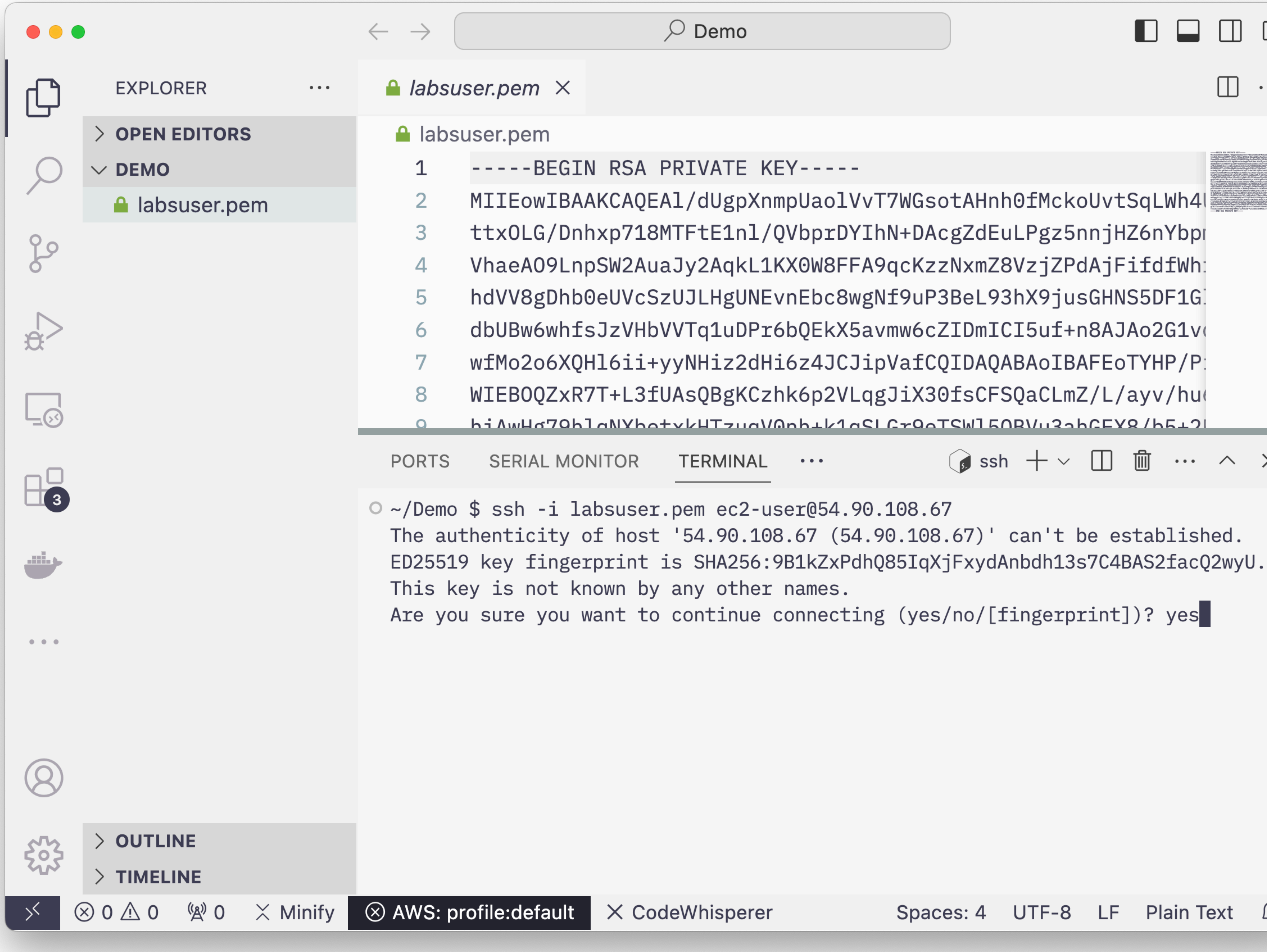
```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAl/dUgpXnmpUaolVvT7WGsotAHnh0fMckoUvtSqLWh4
ttxOLG/Dnhxp718MTFtE1nl/QVbprDYIhN+DAcgZdEuLPgz5nnjHZ6nYbp
VhaeAO9LnpSW2AuaJy2AqkL1KX0W8FFA9qcKzzNxmZ8VzjZPdAjFifdfWh
hdVV8gDhb0eUVcSzUJLHgUNEvnEbc8wgNf9uP3BeL93hX9jusGHNS5DF1G
dbUBw6whfsJzVHbVVTq1uDPr6bQEkX5avmw6cZIDmICI5uf+n8AJAo2G1v
wfMo2o6XQHl6ii+yyNHiz2dHi6z4JCJipVafCQIDAQABAoIBAFEoTYHP/P
WIEBOQZxR7T+L3fUAsQBgKCzhk6p2VLqgJiX30fsCFSQaCLmZ/L/ayv/hu
hiAwHg79hlqNXbetxkHTzuqV0nh+k1qSLGr9eTSWl5QBVu3ahGEX8/b5+2
```

```
~/Demo $ ssh -i labsuser.pem ec2-user@54.90.108.67
```

EC2 Security

Connecting with SSH

- Gotchas
- First time connection will prompt you to accept the remote host’s fingerprint
  - yes



labsuser.pem

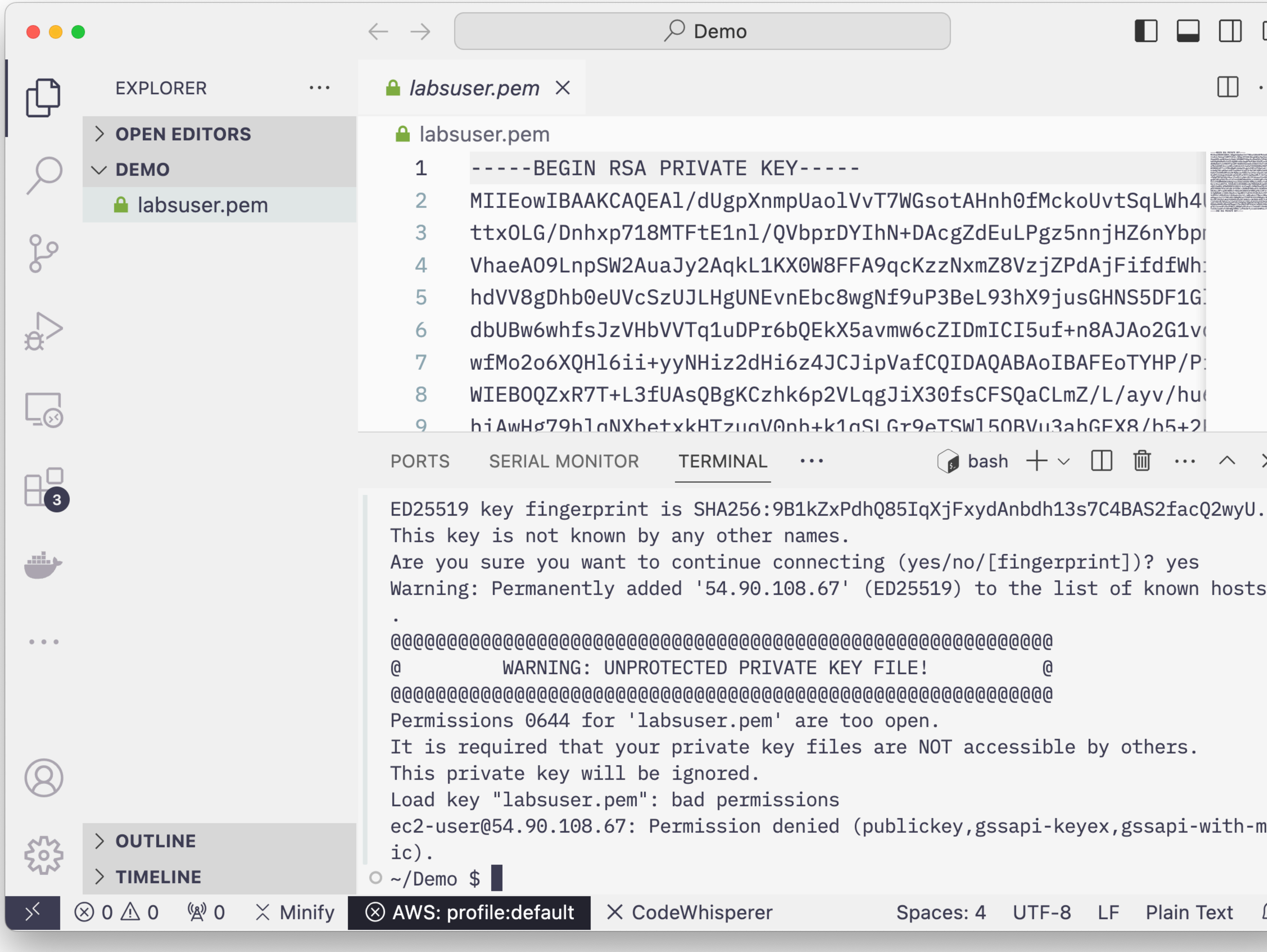
```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAl/dUgpXnmpUaolVvT7WGsotAHnh0fMckoUvtSqLWh4
ttxOLG/Dnhxp718MTFtE1nl/QVbprDYIhN+DAcgZdEuLPgz5nnjHZ6nYbp
VhaeAO9LnpSW2AuaJy2AqkL1KX0W8FFA9qcKzzNxmZ8VzjZPdAjFifdfWh
hdVV8gDhb0eUVcSzUJLHgUNEvnEbc8wgNf9uP3BeL93hX9jusGHNS5DF1G
dbUBw6whfsJzVHbVVTq1uDPr6bQEkX5avmw6cZIDmICI5uf+n8AJAo2G1v
wfMo2o6XQHl6ii+yyNHiz2dHi6z4JCJipVafCQIDAQABAoIBAFEoTYHP/P
WIEBOQZxR7T+L3fUAsQBgKCzhk6p2VLqgJiX30fsCFSQaCLmZ/L/ayv/hu
hiAwHg79blgNXbetxkHTzugV0nb+k1gSLGr9eTSWl5OBVu3abGFX8/h5+2
```

```
~/Demo $ ssh -i labsuser.pem ec2-user@54.90.108.67
The authenticity of host '54.90.108.67 (54.90.108.67)' can't be established.
ED25519 key fingerprint is SHA256:9B1kZxPdhQ85IqXjFxydAnbdh13s7C4BAS2facQ2wyU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

EC2 Security

Connecting with SSH

- Gotchas
- The downloaded private key file may have incorrect permissions
- SSH will not allow you to use it until you fix them



```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAl/dUgpXnmpUaolVvT7WGsotAHnh0fMckoUvtSqLWh4
ttxOLG/Dnhxp718MTFtE1nl/QVbprDYIhN+DAcgZdEuLPgz5nnjHZ6nYbp
VhaeAO9LnpSW2AuaJy2AqkL1KX0W8FFA9qcKzzNxmZ8VzjZPdAjFifdfWh
hdVV8gDhb0eUVcSzUJLHgUNEvnEbc8wgNf9uP3BeL93hX9jusGHNS5DF1G
dbUBw6whfsJzVHbVVTq1uDPr6bQEkX5avmw6cZIDmICI5uf+n8AJAo2G1v
wfMo2o6XQHl6ii+yyNHiz2dHi6z4JCJipVafCQIDAQABAoIBAFEoTYHP/P
WIEBOQZxR7T+L3fUAsQBgKCzhk6p2VLqgJiX30fsCFSQaCLmZ/L/ayv/hu
hjAwHg79hlqNXbetxkHTzuqV0nh+k1qSLGr9eTSWl5OBVu3ahGFX8/h5+2
```

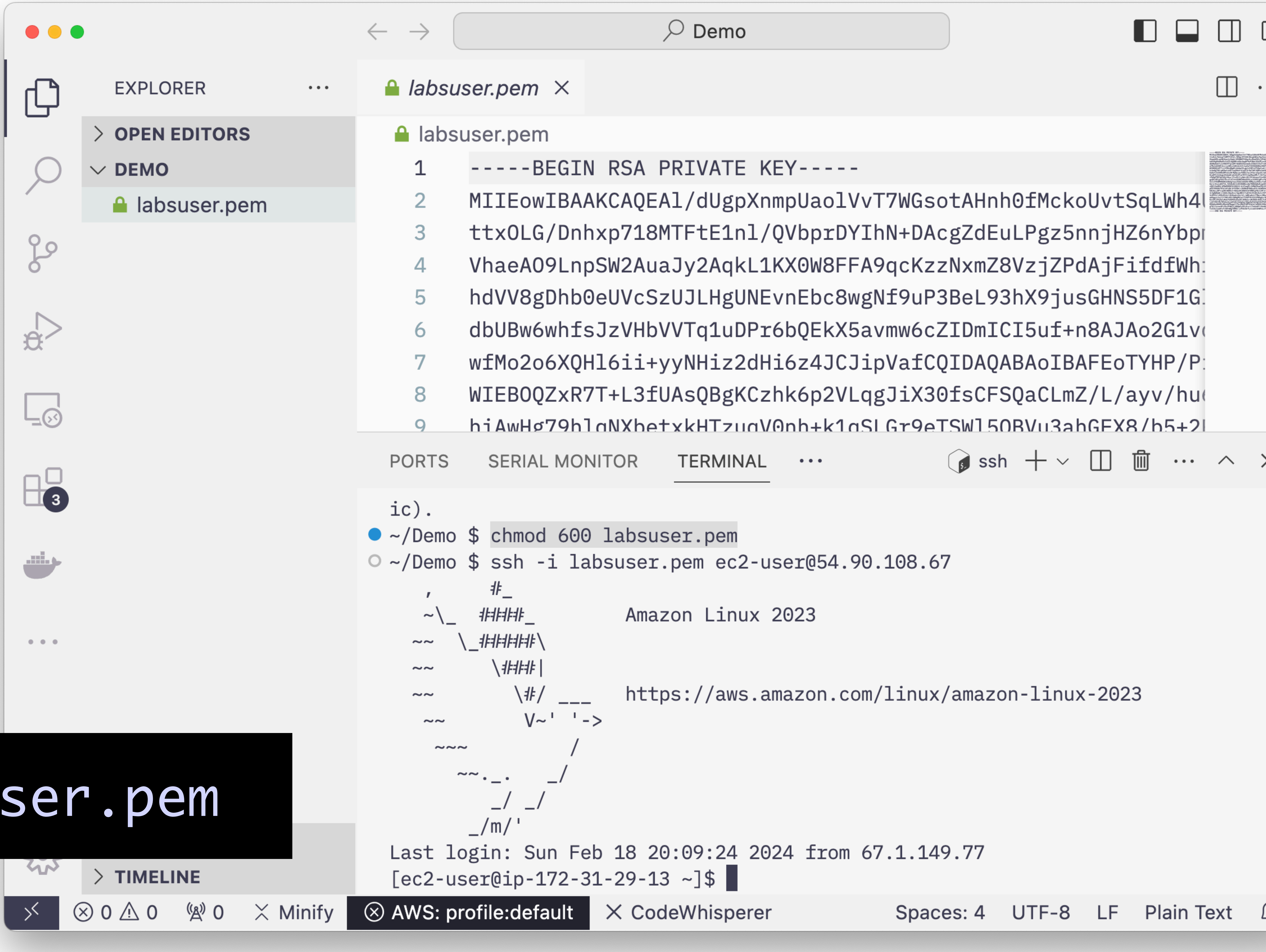
```
ED25519 key fingerprint is SHA256:9B1kZxPdhQ85IqXjFxydAnbdh13s7C4BAS2facQ2wyU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '54.90.108.67' (ED25519) to the list of known hosts
.
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0644 for 'labsuser.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "labsuser.pem": bad permissions
ec2-user@54.90.108.67: Permission denied (publickey,gssapi-keyex,gssapi-with-m
ic).
~/Demo $
```

EC2 Security

Connecting with SSH

- Gotchas
- Use the chmod command to change permissions on a file
- Private key file must only be readable by the user

```
chmod 600 labsuser.pem
```



labsuser.pem

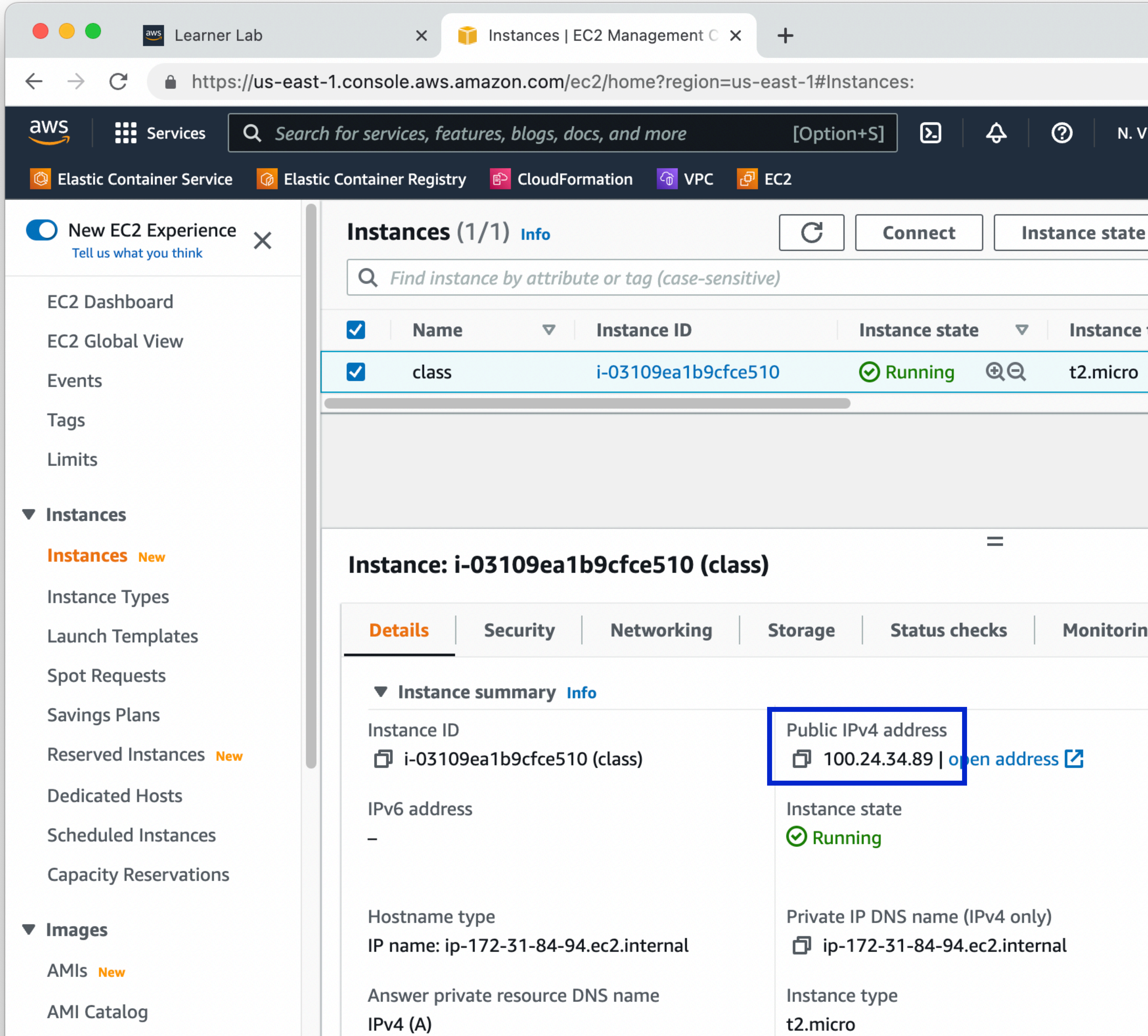
```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAl/dUgpXnmpUaolVvT7WGsotAHnh0fMckoUvtSqLWh4
ttxOLG/Dnhxp718MTFtE1nl/QVbprDYIhN+DAcgZdEuLPgz5nnjHZ6nYbp
VhaeAO9LnpSW2AuaJy2AqkL1KX0W8FFA9qcKzzNxmZ8VzjZPdAjFifdfWh
hdVV8gDhb0eUVcSzUJLHgUNEvnEbc8wgNf9uP3BeL93hX9jusGHNS5DF1G
dbUBw6whfsJzVHbVVTq1uDPr6bQEkX5avmw6cZIDmICI5uf+n8AJAo2G1v
wfMo2o6XQHl6ii+yyNHiz2dHi6z4JCJipVafCQIDAQABAoIBAFEoTYHP/P
WIEBOQZxR7T+L3fUAsQBgKCzhk6p2VLqgJiX30fsCFSQaCLmZ/L/ayv/hu
hiAwHg79hlqNXbetxkHTzuqV0nh+k1qSLGr9eTSWl5OBVu3ahGEX8/b5+2
```

```
ic).
~/Demo $ chmod 600 labsuser.pem
~/Demo $ ssh -i labsuser.pem ec2-user@54.90.108.67
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Sun Feb 18 20:09:24 2024 from 67.1.149.77
[ec2-user@ip-172-31-29-13 ~]$ 
```

EC2

Connecting with SSH

- Instance Public IP addresses will change each time you stop and start them
- Need to check each time in the AWS EC2 Console for the current IP



☑	Name	Instance ID	Instance state	Instance
☑	class	i-03109ea1b9cfce510	✓ Running	t2.micro

Instance: i-03109ea1b9cfce510 (class)

Instance summary Info

Instance ID
i-03109ea1b9cfce510 (class)

Public IPv4 address
100.24.34.89 | open address

IPv6 address
–

Instance state
✓ Running

Hostname type
IP name: ip-172-31-84-94.ec2.internal

Private IP DNS name (IPv4 only)
ip-172-31-84-94.ec2.internal

Answer private resource DNS name
IPv4 (A)

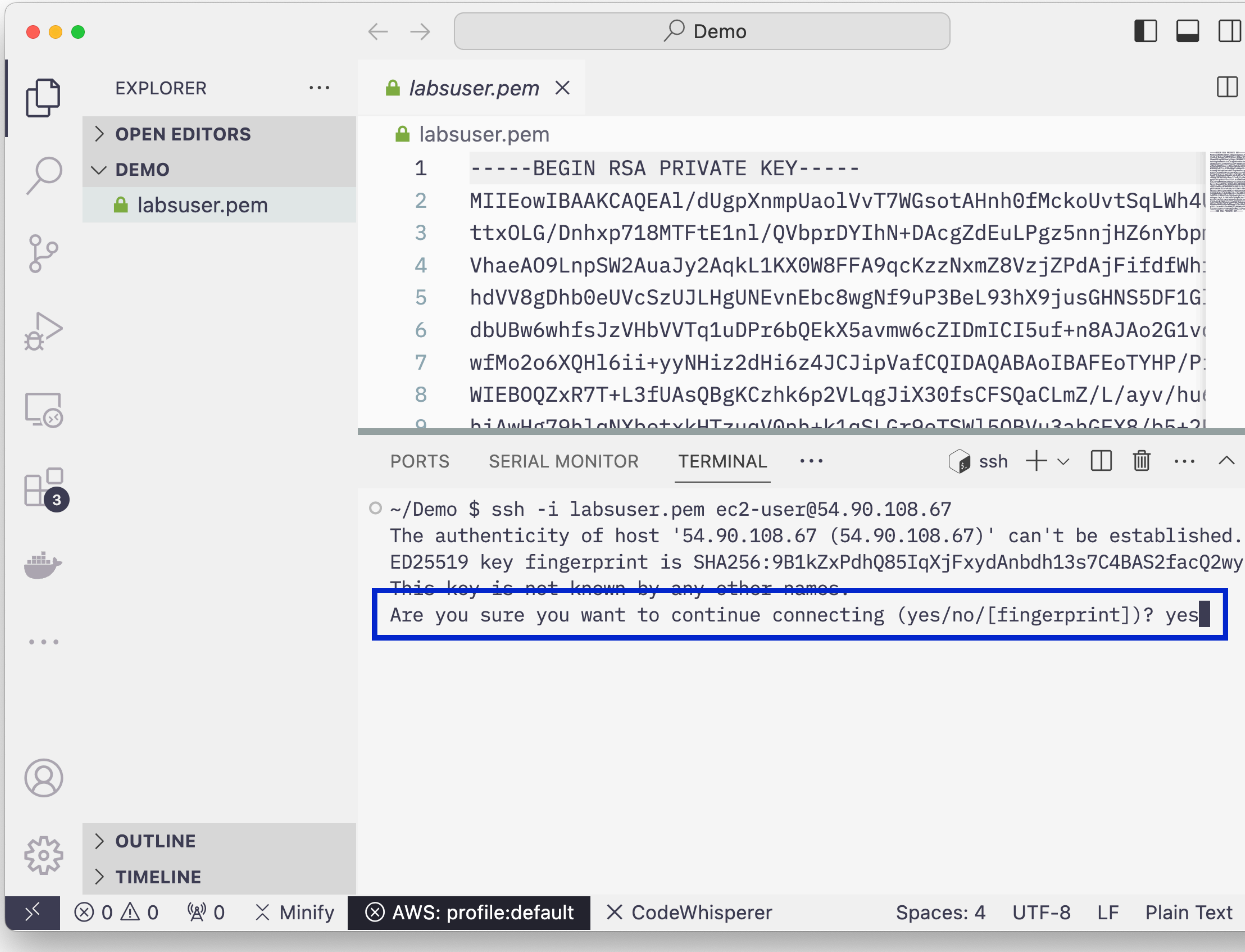
Instance type
t2.micro

EC2

Connecting with SSH

- Since the IP address changes often, you'll have to accept the signature each time this happens

- Welcome to the cloud!



```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAl/dUgpXnmpUaolVvT7WGsotAHnh0fMckoUvtSqLWh4
ttxOLG/Dnhxp718MTFtE1nl/QVbprDYIhN+DAcgZdEuLPgz5nnjHZ6nYbp
VhaeAO9LnpSW2AuaJy2AqkL1KX0W8FFA9qcKzzNxmZ8VzjZPdAjFifdfWh
hdVV8gDhb0eUVcSzUJLHgUNEvnEbc8wgNf9uP3BeL93hX9jusGHNS5DF1G
dbUBw6whfsJzVHbVVTq1uDPr6bQEkX5avmw6cZIDmICI5uf+n8AJAo2G1v
wfMo2o6XQHl6ii+yyNHiz2dHi6z4JCJipVafCQIDAQABAoIBAFEoTYHP/P
WIEBOQZxR7T+L3fUAsQBgKCzhk6p2VLqgJiX30fsCFSQaCLmZ/L/ayv/hu
hiAwHg79hlqNXbetxkHTzuqV0nh+k1qSLGr9eTSWl5OBVu3ahGFX8/b5+2
```

```
~/Demo $ ssh -i labsuser.pem ec2-user@54.90.108.67
The authenticity of host '54.90.108.67 (54.90.108.67)' can't be established.
ED25519 key fingerprint is SHA256:9B1kZxPdhQ85IqXjFxydAnbdh13s7C4BAS2facQ2wy
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

Server Best Practices

Stay Up To Date

- Part of the defense in depth principle
- Automation this for production. It’s up to you for your development environments.
- During any new development session, you should first update software:

```
sudo yum update
```

- On a brand new instance, there likely won’t be anything to update. There will as this instance gets used longer.
- If this updates the “kernel” package, you’ll need to reboot to run the new kernel.

Server Best Practices

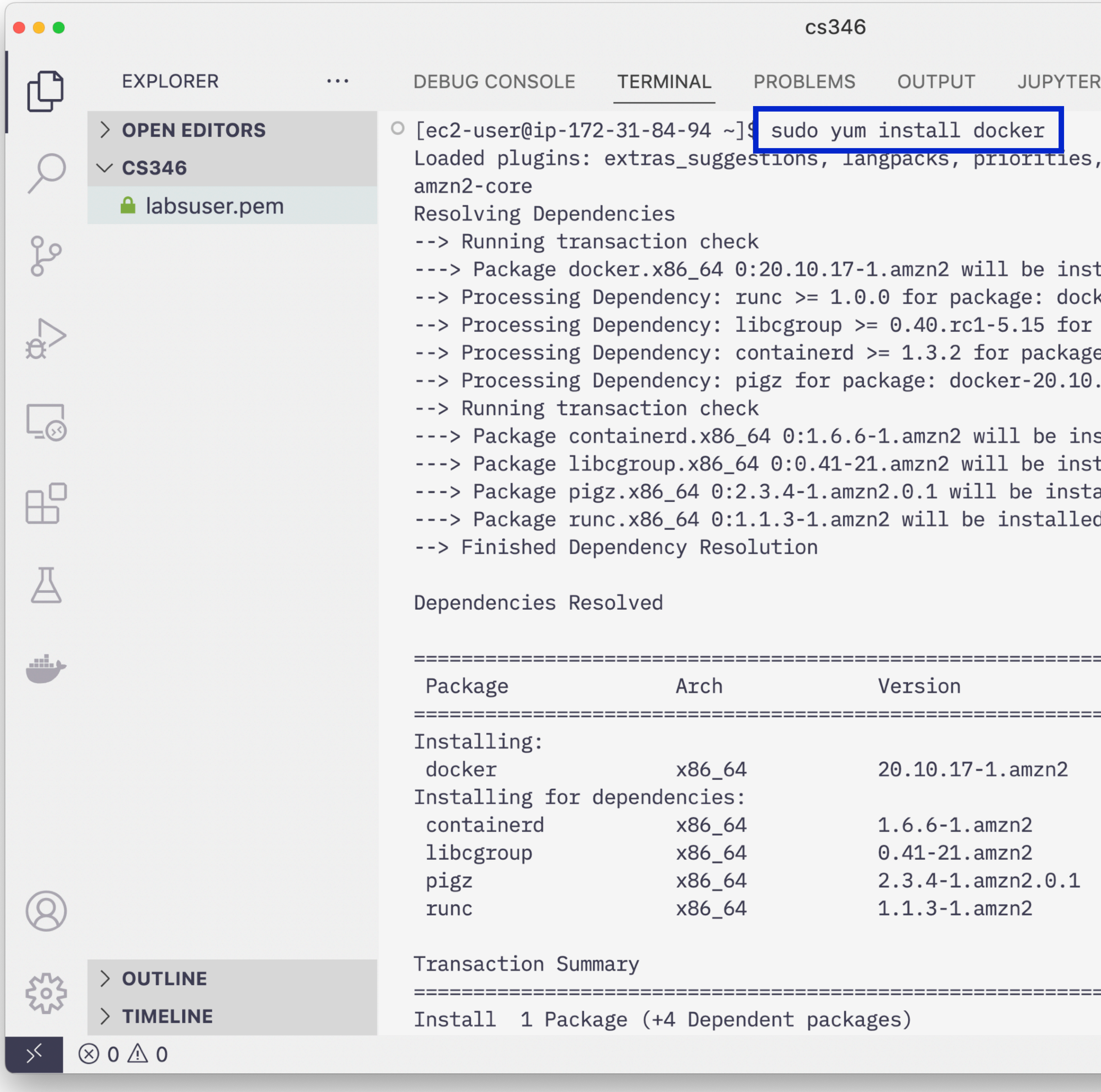
Debian vs RedHat Derivatives

- Your containers have been based off of Ubuntu, which is based on Debian Linux
- AWS maintains their own distribution, Amazon Linux, which is a derivative of CentOS, which is a derivative of RedHat Linux
- Good idea to be comfortable with both major linux flavors
- Mostly, your experience will be the same, but a few changes

Server Best Practices

Installing Packages

- Use **`yum`** instead of **`apt-get`** to install
  - Some package names different
- Some default config changed
- Let’s install Docker
  - Can run Debian based containers on RedHat derivatives just fine
  - It’s still the same Linux Kernel

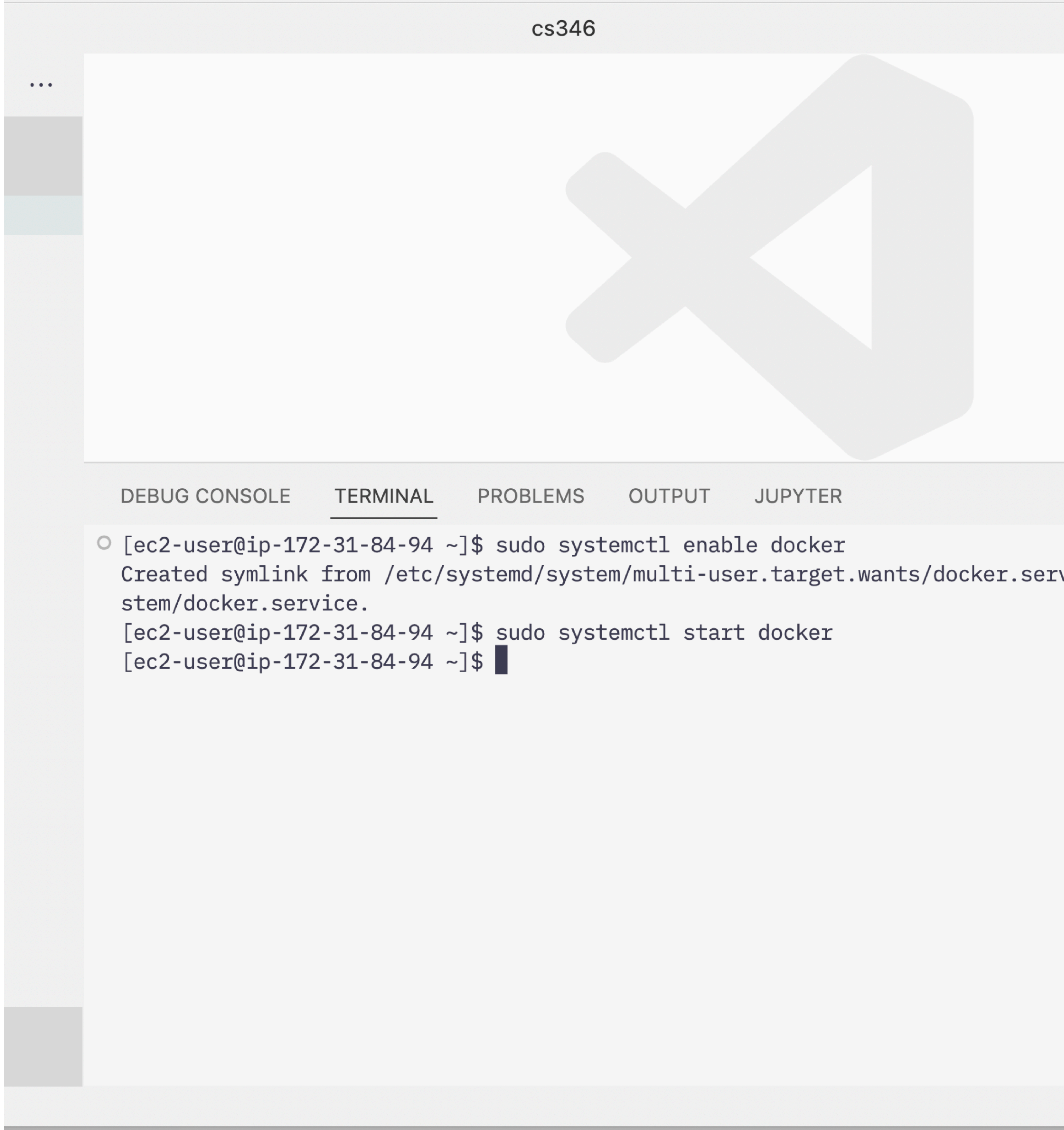


```
[ec2-user@ip-172-31-84-94 ~]$ sudo yum install docker
Loaded plugins: extras_suggestions, langpacks, priorities,
amzn2-core
Resolving Dependencies
--> Running transaction check
---> Package docker.x86_64 0:20.10.17-1.amzn2 will be inst
--> Processing Dependency: runc >= 1.0.0 for package: dock
--> Processing Dependency: libcgroup >= 0.40.rc1-5.15 for
--> Processing Dependency: containerd >= 1.3.2 for package
--> Processing Dependency: pigz for package: docker-20.10.
--> Running transaction check
---> Package containerd.x86_64 0:1.6.6-1.amzn2 will be ins
---> Package libcgroup.x86_64 0:0.41-21.amzn2 will be inst
---> Package pigz.x86_64 0:2.3.4-1.amzn2.0.1 will be insta
---> Package runc.x86_64 0:1.1.3-1.amzn2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

==================================================================
 Package                 Arch                Version
==================================================================
Installing:
 docker                  x86_64              20.10.17-1.amzn2
Installing for dependencies:
 containerd              x86_64              1.6.6-1.amzn2
 libcgroup               x86_64              0.41-21.amzn2
 pigz                    x86_64              2.3.4-1.amzn2.0.1
 runc                    x86_64              1.1.3-1.amzn2

Transaction Summary
==================================================================
Install  1 Package (+4 Dependent packages)
```

Server Best Practices

Starting Services

- Amazon Linux uses **`systemctl`** to start and stop services like docker
- **`enable`** tells **`systemctl`** to start this service when the server starts
- **`start`** is needed to start the service now



```
[ec2-user@ip-172-31-84-94 ~]$ sudo systemctl enable docker
Created symlink from /etc/systemd/system/multi-user.target.wants/docker.serv
stem/docker.service.
[ec2-user@ip-172-31-84-94 ~]$ sudo systemctl start docker
[ec2-user@ip-172-31-84-94 ~]$ 
```

Server Best Practices

What’s with all the **`sudo`**?

- Containers (typically) only have one user: **`root`**
- VMs support multiple users, you don’t have **`root`** access by default
- **`sudo`** required for many operations - “superuser do"
- **`chown`** useful - “change owner”
    - Change owner from **`root`** to **`ec2-user`**, for often-modified files, directories

```
sudo chown ec2-user:ec2-user FILE
```

Containers vs. VMs

- VMs are persistent, won’t lose data!
    - Don’t have to re-upload config
    - Don’t have to re-install software
    - But hard to “experiment and then undo”
- EC2 VMs have public IP addresses
    - You can now run a webserver with a public IP!
    - But the lab will shut down your instances after 4 hours
        - Normal EC2 instances stay on forever (if you want)