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

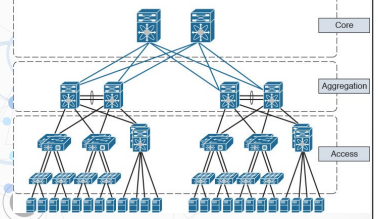
- Many resources are shared
- Power Network A/C
- Massive economies of scale

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.



7

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***.

8

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)

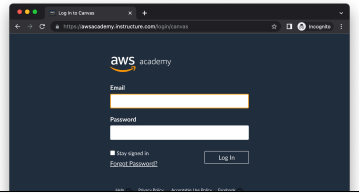
9

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

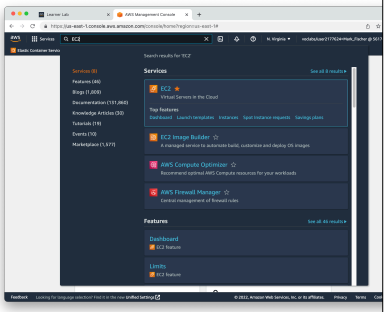


AWS EC2

- 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

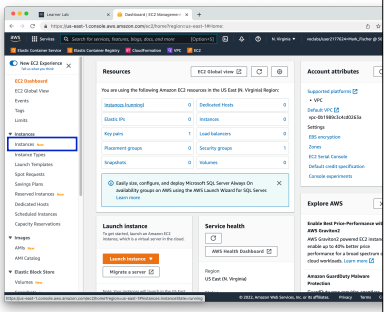
EC2

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



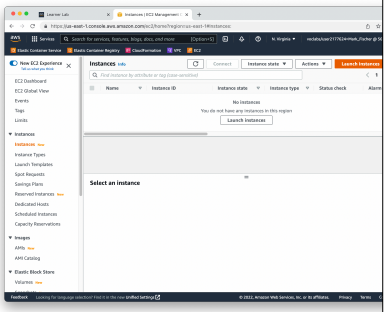
EC2

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



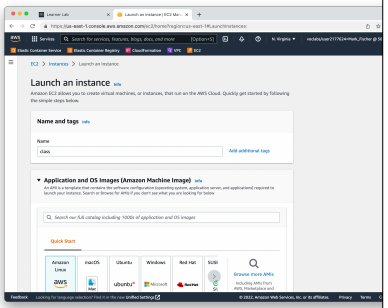
EC2

- 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**"



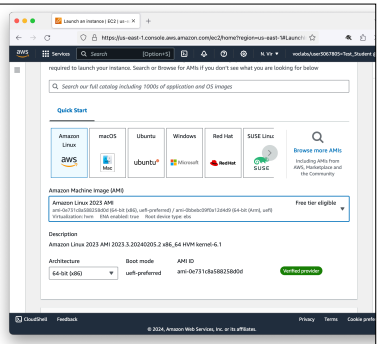
EC2

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



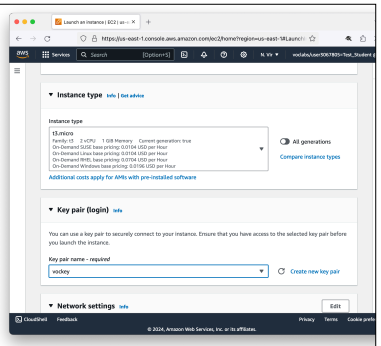
EC2

- 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



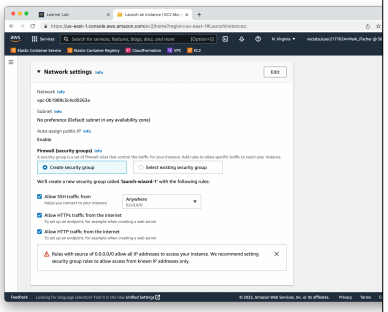
EC2

- 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



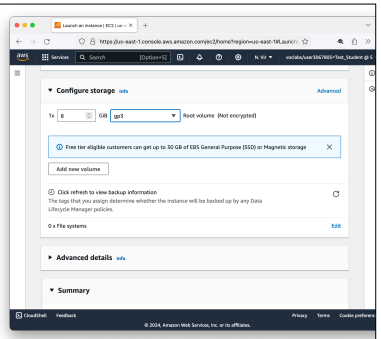
EC2

- 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



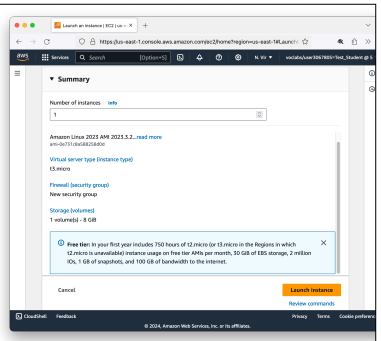
EC2

- 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



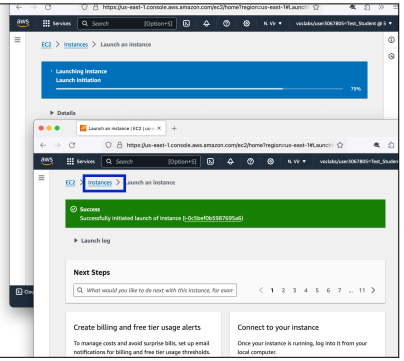
EC2

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



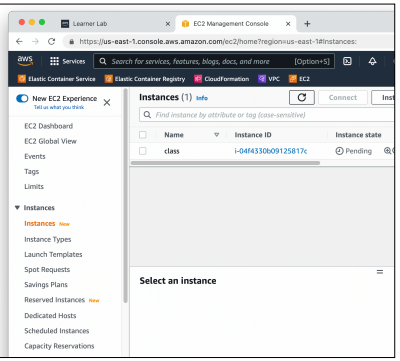
EC2

- 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



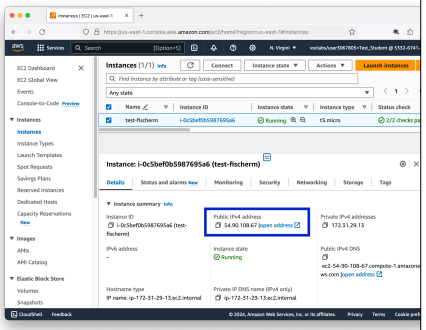
EC2

- 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"



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



EC2

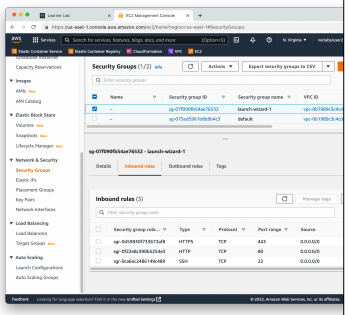
Stop vs. Terminate

- 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

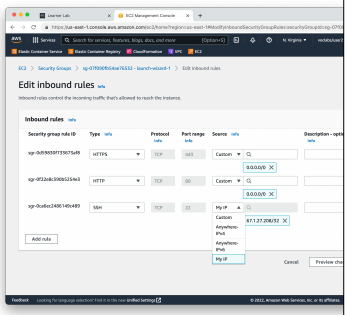
- From the main EC2 Console page, choose Security Groups under Network & Security
- Your EC2 instance has a Public IP address
- When we set up the instance we created a new security group
- This allows incoming traffic on port 22, 80, and 443
- `0.0.0.0/0` means “anywhere”



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



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

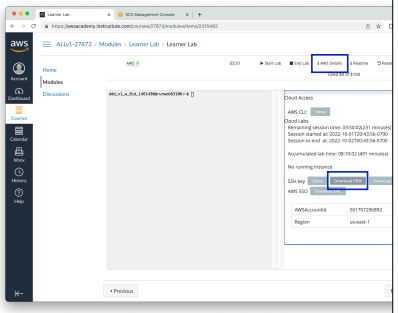
Connecting with SSH

- 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

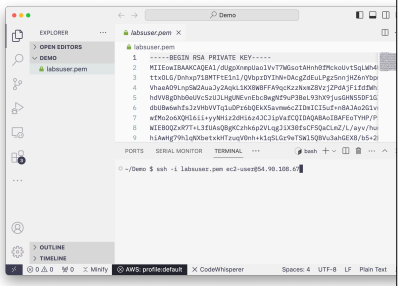
- Back in the AWS Academy Lab in Canvas
- Click on AWS Details
- Download the PEM file
  - May need the PPK file if you are using Putty on Windows



EC2 Security

Connecting with SSH

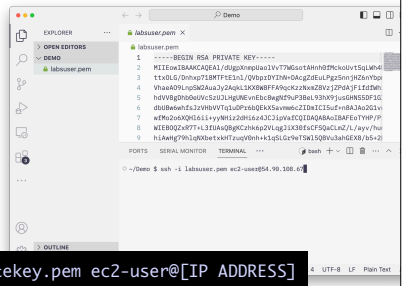
- The PEM file you download maybe named **`labsuser.pem`** like mine, or **`vokey.pem`** as described in the AWS documentation
- Contents of the key file will look something like this



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`**

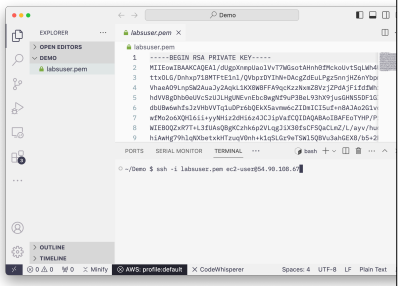



```
ssh -i privatekey.pem ec2-user@[IP ADDRESS]
```


EC2 Security

Connecting with SSH

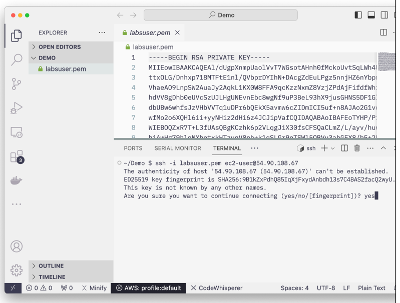
- Windows 10 should have the SSH client installed by default.
- If you do need to install it:
  - Open **Settings**, select **Apps**, then select **Optional Features**.
  - Install **OpenSSH Client**



EC2 Security

Connecting with SSH

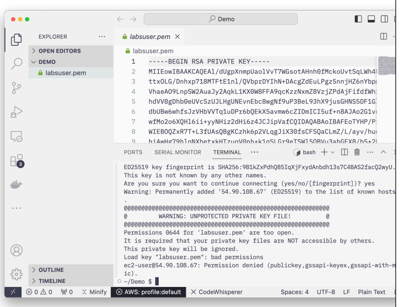
- Gotchas
- First time connection will prompt you to accept the remote host's fingerprint
    - yes



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

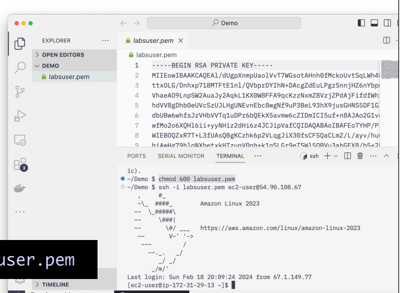


EC2 Security

Connecting with SSH

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

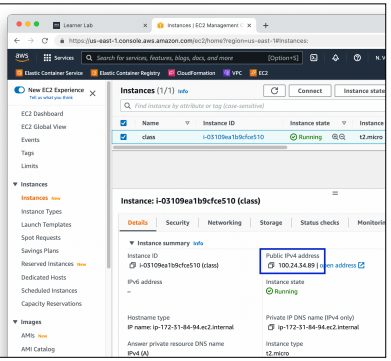
```
chmod 600 labsuser.pem
```



EC2

Connecting with SSH

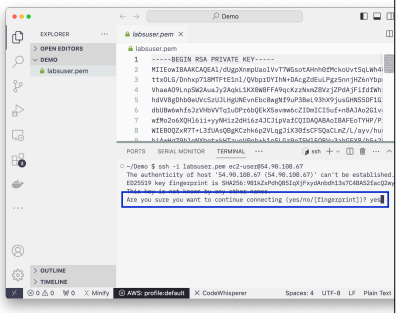
- Instance Public IP addresses will change each time you stop and start them
- Need to check each time in the AWS EC2 Console for the current IP



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!



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

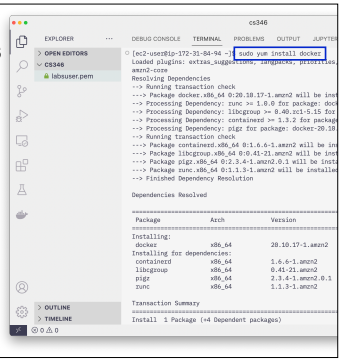
Debian vs RedHat Derivatives

- 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



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

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)