Lecture 3: Cloud Computing Theory & Getting Started

Contents

3.1. Announcements

- TA Sky is hosting a UNIX tutorial this week (details in slack).
- Thanks for participating actively in class.
- Filled up all “Thoughts/Discussion and refresher questions” sections for the previous lectures.

3.2. Learning objectives

- Understanding cloud computing and its impacts.
- Identifying the benefits of cloud computing.
- Describing basic AWS Cloud architectural principles and security.
- Describing the shared security model.
- Getting started with AWS from scratch.

3.3. Refresher questions

- What are the key file format properties to consider when working with big data?
- Have you heard of columnar file formats? Have you worked with them?
- How can you save space when dealing with a large file without compromising your analysis?
- Suppose you are given a 50 GB parquet file. What are some important things you would consider when working with a big parquet file?
- Where are you going to use the parquet file format over CSV? (They can also provide a scenario and ask for your opinion.)

- Are you aware of any encoding schemes used to save space in parquet files?
- Where do you think serialization and deserialization happen?
- What are some ways to process a file in parallel? Are you aware of packages that can help you do this?
- What is the main idea of the Apache Arrow project?
- Have you worked with any packages that make use of arrow?
- What is the difference b/w eager evaluation vs lazy evaluation ? Have you experienced lazy evaluation somewhere ?
- Are you aware of any emerging file formats to work with big data? (Well parquet is pretty much established, but are there any other file formats)
- Do you think Feather will become the next-generation file format? Why do you think so? (in optional)
- Can you explain the structure of a parquet file to me and why it is beneficial?
- Do you know how to convert a CSV file to a parquet file for storage?
- How can you efficiently read a CSV file into pandas? (in optional)
- What is the difference between OLAP and OLTP, and where does the relational database lie?
- What is the difference b/w a data warehouse and a database?
- What is the core reason behind having a data warehouse?

Click toggle below to see the answers.

Answer

- What are the key file format properties to consider when working with big data?
  - We saw lot of properties in the last class (like readable, columnar etc...). Refer to that section.
- Have you heard of columnar file formats? Have you worked with them?
  - Yes, we have worked with parquet files in the last class.
- How can you save space when dealing with a large file without compromising your analysis?
  - We can use columnar file formats like parquet, feather, etc. to save space. They are more memory efficient and faster to read and write.
- Suppose you are given a 50 GB parquet file. What are some important things you would consider when working with a big parquet file?
  - I will make sure predicate/projection pushdown and partition pruning can be applied to the file.
- Where are you going to use the parquet file format over CSV? (They can also provide a scenario and ask for your opinion.)
  - I will use parquet over CSV when I am working with big data and I got OLAP queries.
- Are you aware of any encoding schemes used to save space in parquet files?
  - Yes, we have seen dictionary encoding and couple more in the last class.
- Where do you think serialization and deserialization happen?
  - Serialization happens when we write data to a file and deserialization happens when we read data from a file. Also happens when we send data over the network.
- What are some ways to process a file in parallel? Are you aware of packages that can help you do this?
  - We can use duckdb, polars, etc. to process files in parallel.
- What is the main idea of the Apache Arrow project?
  - To put it very simple to have a common in-memory format for big data.
- Have you worked with any packages that make use of arrow?

    - Yes, we have seen polars and duckdb in the last class.
- What is the difference b/w eager evaluation vs lazy evaluation ? Have you experienced lazy evaluation somewhere ?
    - Eager evaluation is when we evaluate the expression as soon as we write it. Lazy evaluation is when we evaluate the expression when we need the result.
    - We have seen lazy evaluation in the last class when we used dplyr.
- Are you aware of any emerging file formats to work with big data? (Well parquet is pretty much established, but are there any other file formats)
    - Yes, we have seen feather in the last class.
- Do you think Feather will become the next-generation file format? Why do you think so? (in optional)
    - I think it will become the next-generation file format because it is faster than parquet and it is more memory efficient.
- Can you explain the structure of a parquet file to me and why it is beneficial?
    - Parquet file is a columnar file format (remember what I drew in last class). It is beneficial because it is faster to read and write and it is more memory efficient.
- Do you know how to convert a CSV file to a parquet file for storage?
    - Yes, we have seen that in the last class.
- How can you efficiently read a CSV file into pandas? (in optional)
    - We can use arrow engine to read a CSV file into pandas. See last class optional section for more details.
- What is the difference between OLAP and OLTP, and where does the relational database lie?
    - OLTP is Online Transaction Processing and OLAP is Online Analytical Processing.
    - OLTP is used for transactional data and OLAP is used for analytical data.
    - Relational database is used for OLTP.
- What is the difference b/w a data warehouse and a database?
    - Data warehouse is used for OLAP and database is used for OLTP.
- What is the core reason behind having a data warehouse?
    - The core reason behind having a data warehouse is to have a central place to store all the data and to have a central place to do all the analysis.

3.4. Video Zone

Here is the video helper for week2. The reason why this is here because some of you will be having lab sessions before the lectures. So, you can use this as a reference.

Following are the main sections covered in the above video:

- AWS lab setup (lecture 3)
- Setting up AWS CLI (lecture 3)
- Setting up an EC2 instance (lecture 4)
- Logging into EC2 instance (lecture 4)
- Setting up a common space in EC2 (lecture 4)
- Setting up Jupyter notebook on EC2 (lecture 4)
- Installing packages in jupyter notebook(lecture 4)
- Setting up S3 bucket (lecture 4)

OPTIONAL: Following is a short video you can check for a quick refresher in UNIX commands.

3.5. WHAT, WHY, HOW cloud computing?

3.5.1. Introduction to cloud computing

Cloud computing is often portrayed as an emerging technology or a must-have skill in the job market. However, the truth is that we are using it in our daily lives without even realizing it. For example,

- Where do your emails get stored?
- Are your files saved on OneDrive or Google Drive?
- And where are YouTube videos stored?

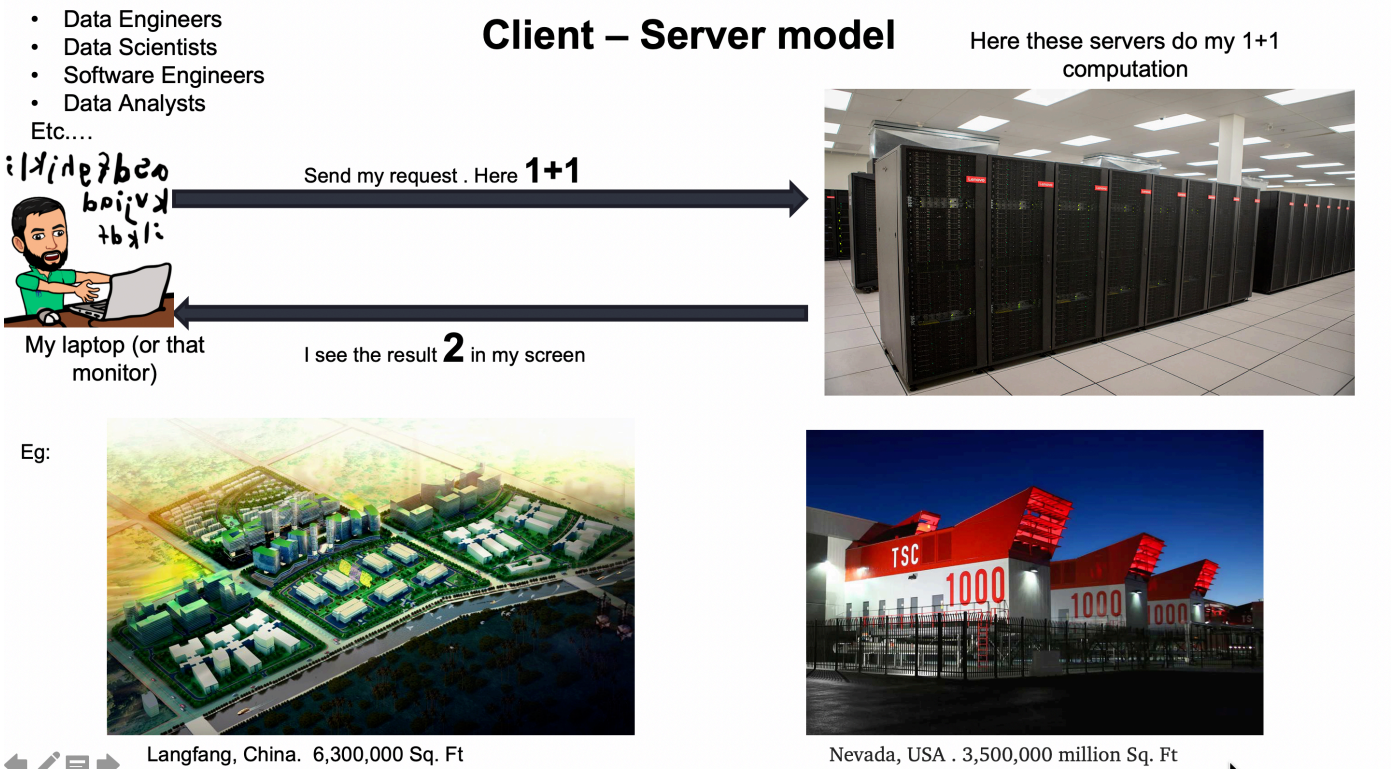
Forget all that; think about what we worked on last week. We got our data from figshare(or the data that is stored in figshare); where do they store this data? They use AWS, and more specifically, they use S3 (more on this in the next class) to store their data. So we are using cloud services quite a lot, but we don't know what it is? 😯

3.5.1.1. Story time 🥳

Let me tell you a short cloud-computing tale. This story starts from a computer that you all are familiar with and by the end of this tale, you will answer ***WHAT, WHY & HOW*** cloud computing.

Here is the computer that I am talking about:

OKAY! So now we all agreed to call monitor as the client and that box as server. Now let's take this knowledge to a bigger picture or think about how this idea will be when you start working in the industry.



Collectively we call these servers data centers (you can also hear people calling some other names like ***on-premise infrastructure***). Mostly all companies (may be used to as there is this trend of moving to Cloud) have data centers, which are considered a company powerhouse for powering data needs (like storage, processing, etc.)

If you want to check out more on those gigantic data centers. Checkout here

Let me pause and take a minute to answer this question:

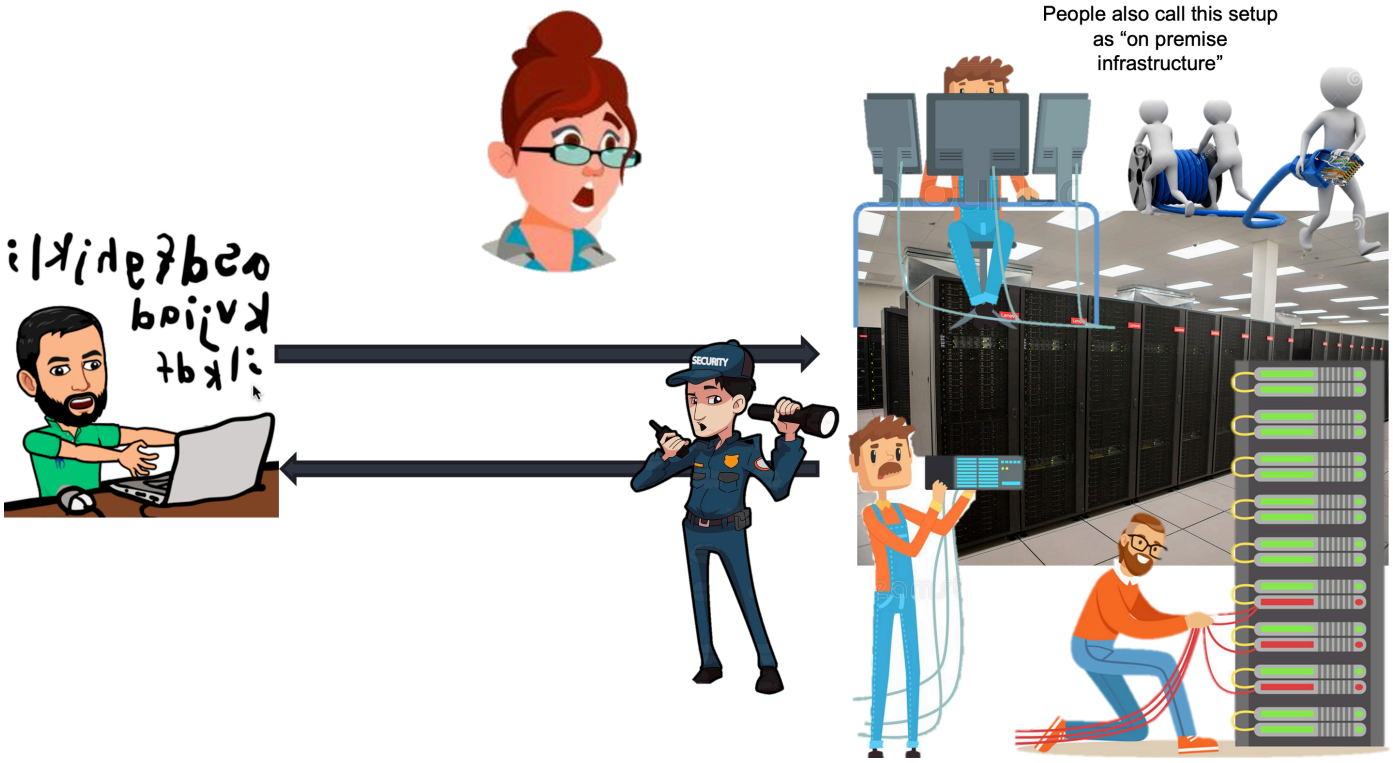
Laptop requirements

- Runs one of the following operating systems: Ubuntu 20.04, macOS Big Sur (version 11.4.x or 11.5.x), Windows 10 Professional, Enterprise or Education (version 2004, 20H2, or 21H1).
  - **Windows 10 Home is not sufficient** as not all the software required for the program can be installed on that OS. Click here to download Windows 10 Education for free from UBC.
  - When installing Ubuntu, checking the box "Install third party..." will (among other things) install proprietary drivers, which can be helpful for wifi and graphics cards.
- Can connect to networks via a wireless connection for on campus work
- Has access to an internet connection that is fast and stable enough for video calling and conducting online quizzes
- Has at least 50 GB disk space available
- Has at least 8 GB of RAM
- Uses a 64-bit CPU
- Is at most 6 years old at the start of the program (4 years old or newer is recommended)
- Uses English as the default language
- Student user has full administrative access to the computer

Thoughts/Discussion ?

Let's assume that UBC has a server, and we are using the client-server model for our analysis. In this case, what laptop requirements mandated by MDS would not be relevant?

Now that you understand this client-server model let's look inside these data centers to see which parties are involved.

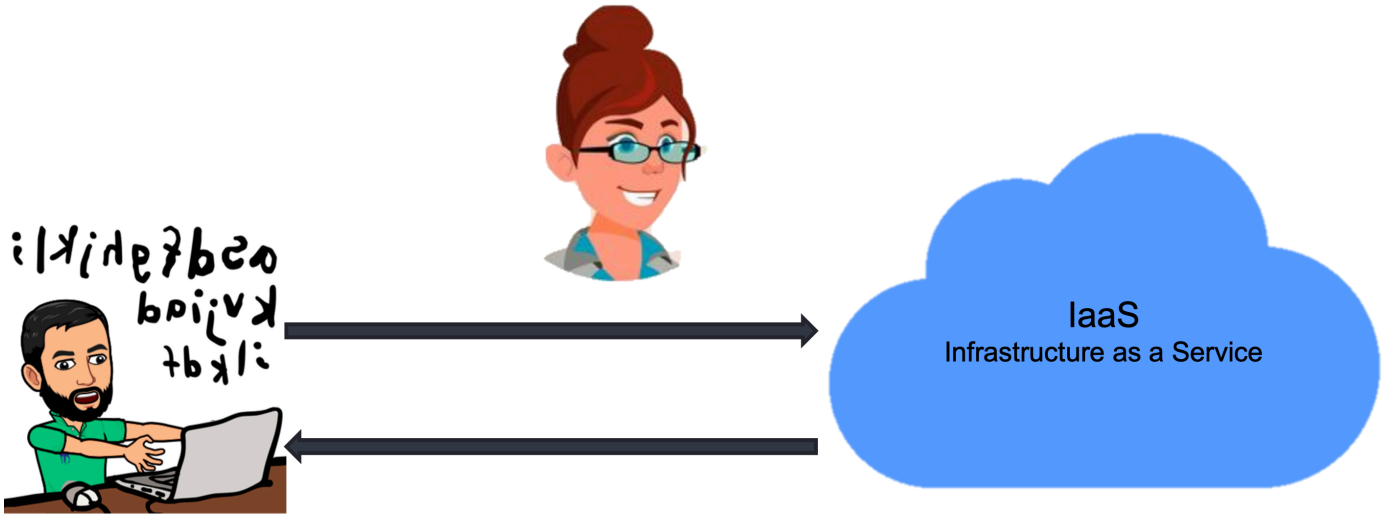


Thoughts/Discussion ?

List down involved labor costs:

List down costs other than labor cost:

We are bringing down most of the costs you listed now with cloud computing. Look how neat and clean the diagram below is as ***cloud providers take care of most of the responsibilities we discussed and use their infrastructure servers as services.***



With this intro, I hope you got some idea on WHAT, WHY, and HOW cloud computing. We will take this idea further to formalize these in the next 3 sections.

Thoughts/Discussion ?

Now that you have a basic understanding of the cloud, please write down your understanding of WHAT cloud computing is, WHY it's important, and HOW it works in your own words.

- ***WHY?***
- ***WHAT?***
- ***HOW?***

3.5.2. Redefining WHY:

- Trade capital expense for variable expense
- Massive economies of scale
- Stop guessing capacity
- Increase speed and agility
- Stop spending money on running and maintaining data centers
- Go global in minutes

Source: AWS

3.5.3. Redefining WHAT

Let's modify the definition that we defined earlier.

"Cloud Computing is when we get ~~a server (computer)~~ ***virtual pool of resources*** *in the cloud for our compute, storage, databases and network services that are provided to users via the internet"*

So WHAT is this virtual pool of resources, and HOW is it made possible? It is made possible by `virtualization`, and that's HOW cloud computing is made possible. So let's get into the details of it.

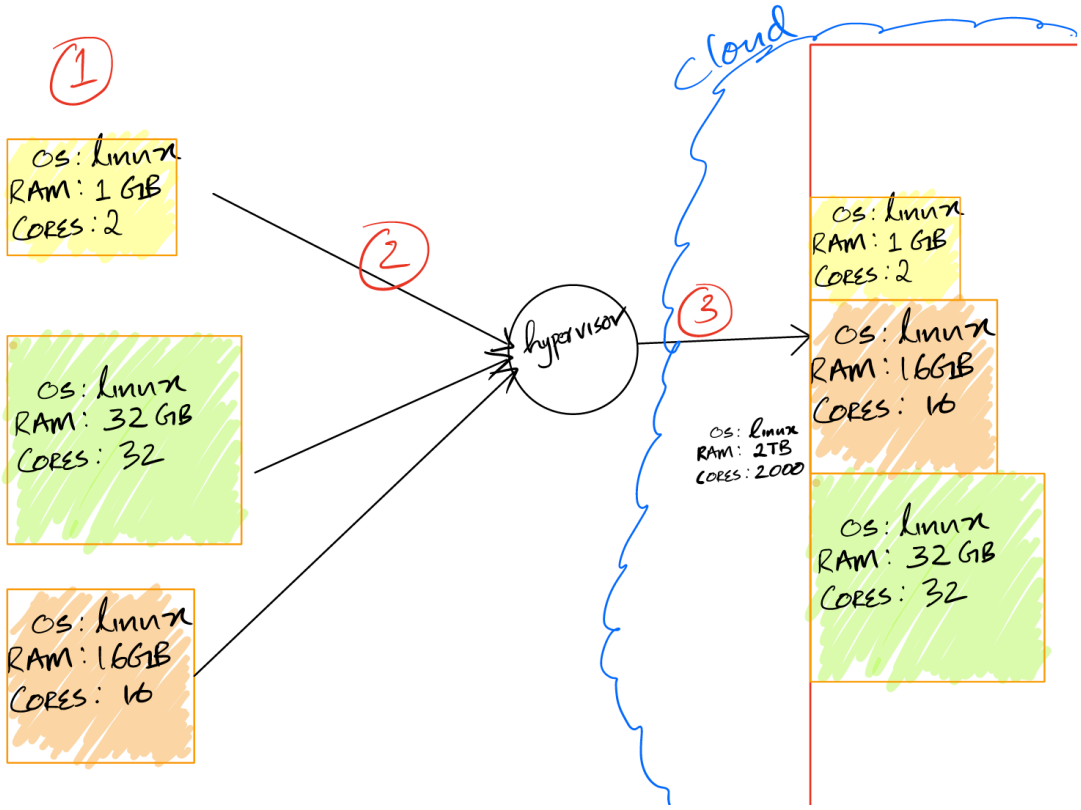
3.5.4. Redefining HOW (Virtualization)

Here is the formal definition ***"Virtualization is the process of running a virtual instance of a computer system in a layer abstracted from the actual hardware."*** This is ***made possible by hypervisors*** (eg: Oracle VM VirtualBox, VMware Server, nitro (currently used by AWS), xen (previously used by AWS).

Note

As a person using AWS services, you don't need to deal with hypervisors, and you won't even see them mentioned anywhere in AWS user docs. Hypervisors are something that AWS uses behind the scenes to fulfill our requests. We're bringing this up to help you understand how cloud computing is made possible.

Here is a drawing to explain this concept!



3.5.5. Cloud providers

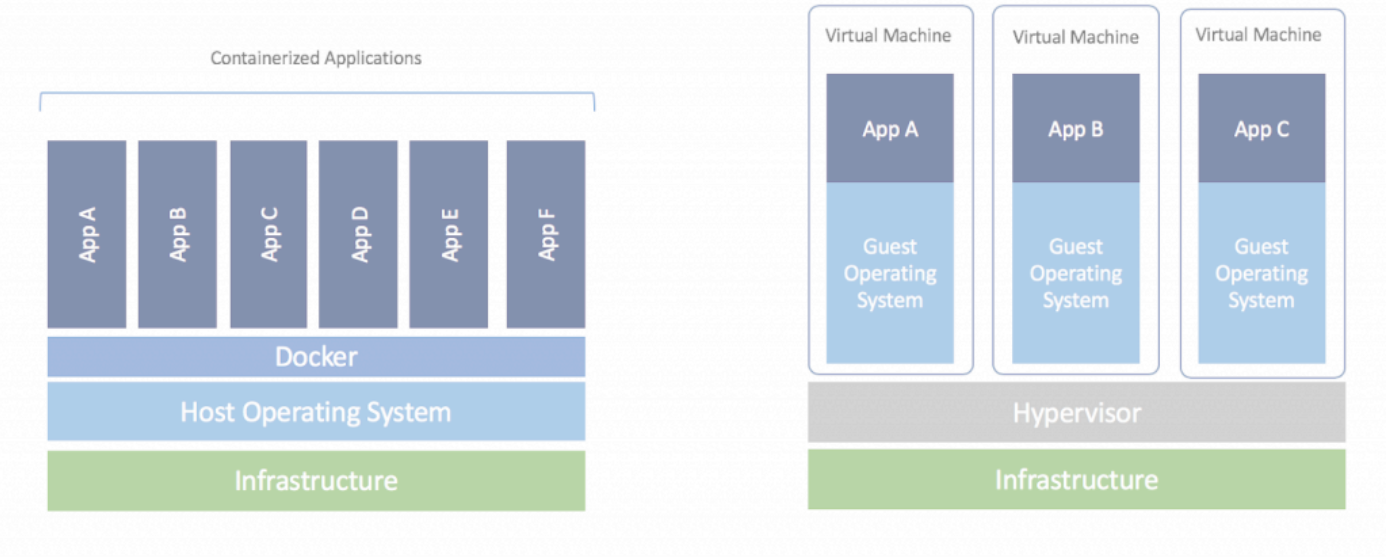
- Amazon Web Services (AWS)
- Microsoft Azure
- Google Cloud

Thoughts/Discussion ?

1. Can you think of any thing that you learned in previous courses that is some what similar to VMs ? If so, what sets them apart from Virtual Machines?
2. Write down WHAT, WHY, and HOW cloud computing in the way you understood.

- ***WHAT?***
- ***WHY?***
- ***HOW?***

Did you draw ?



Note

The rest of the section will be based on cloud provider AWS. So even though the concepts are similar to other cloud providers, we won't be comparing them with other cloud providers. However, by the end of the course, I will include a comparison box so that you can see how the AWS services you learned compare to those offered by GCP and Azure.

3.6. Organization of cloud services

3.6.1. Region

Before setting up a cloud service or setting up the server in the Cloud, we first want to know in ***which part of the world we will set up our instance***. There are many regions around the globe where we can set up the instance. So we want to specify the `region`, the physical geographical location where you want to build your services. AWS currently supports around 26 regions around the globe, including the US (many regions within the US) ,Canada, China , India ,Africa, the Middle East, etc...

Please check out this map, and it shows all the available regions and all regions that will be launching soon.

Tip

Setting up instance and data storage in a region closest to you is one of the reasons it can be accessed at lightning speed.

Important

Since we are in Canada, we must have set up our instance in Canada, but we are using a student version of AWS, and only 2 regions are supported, `us-east-1` and `us-west-2`. Therefore, we will be setting up our instance in the `us-west-2 (Oregon)` region. So make sure you select the `us-west-2 (Oregon)` region within AWS.

3.6.2. Availability Zones

A region consists of one or more availability zones. These are ***isolated locations within a region***, and each availability zone ***consists of multiple data centers***, typically 3. And all these data centers will consist of multiple servers. Within these isolated locations, they have their own power infrastructure and are physically separated from other availability zones by kilometers, ensuring fault isolation. ***AWS recommends replicating data and resources across availability zones for resiliency***, and these availability zones are ***interconnected using high-speed private networking (low-latency links)***.

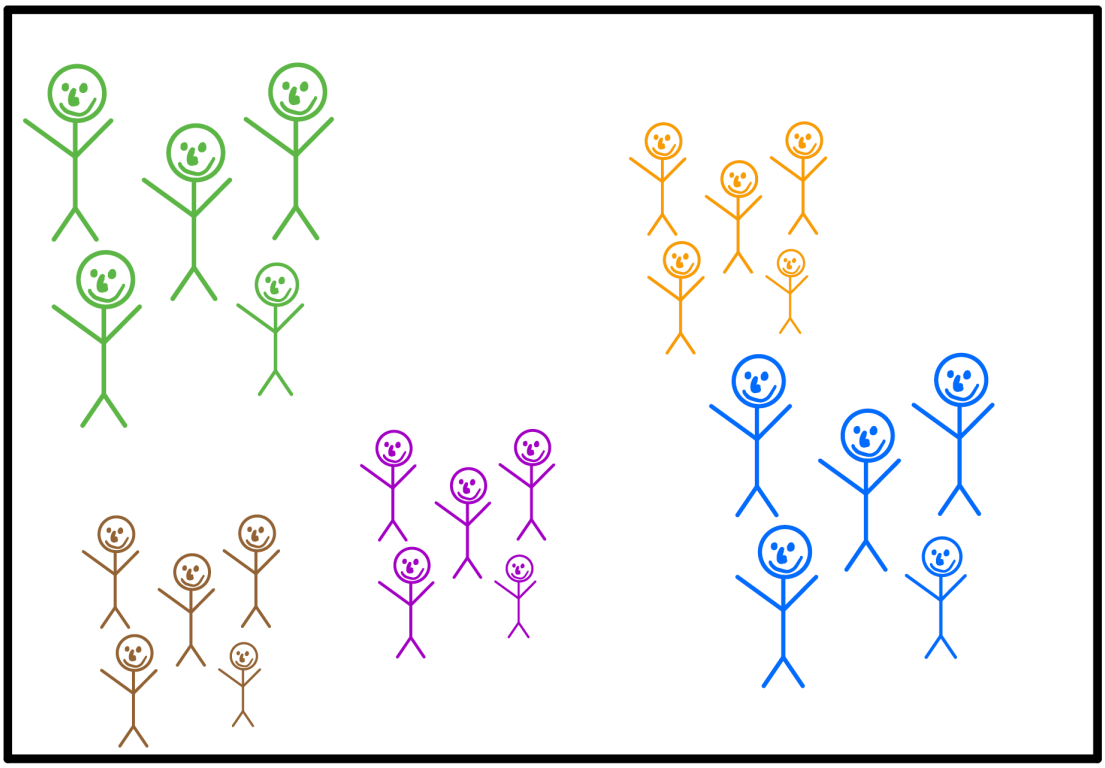
For example us-west-2 (Oregon) region have 3 availabilty zones represented by a region code followed by a letter identifier `us-west-2a` `us-west-2b` `us-west-2c`

3.6.3. AWS global infrastructure & security 👮🏾

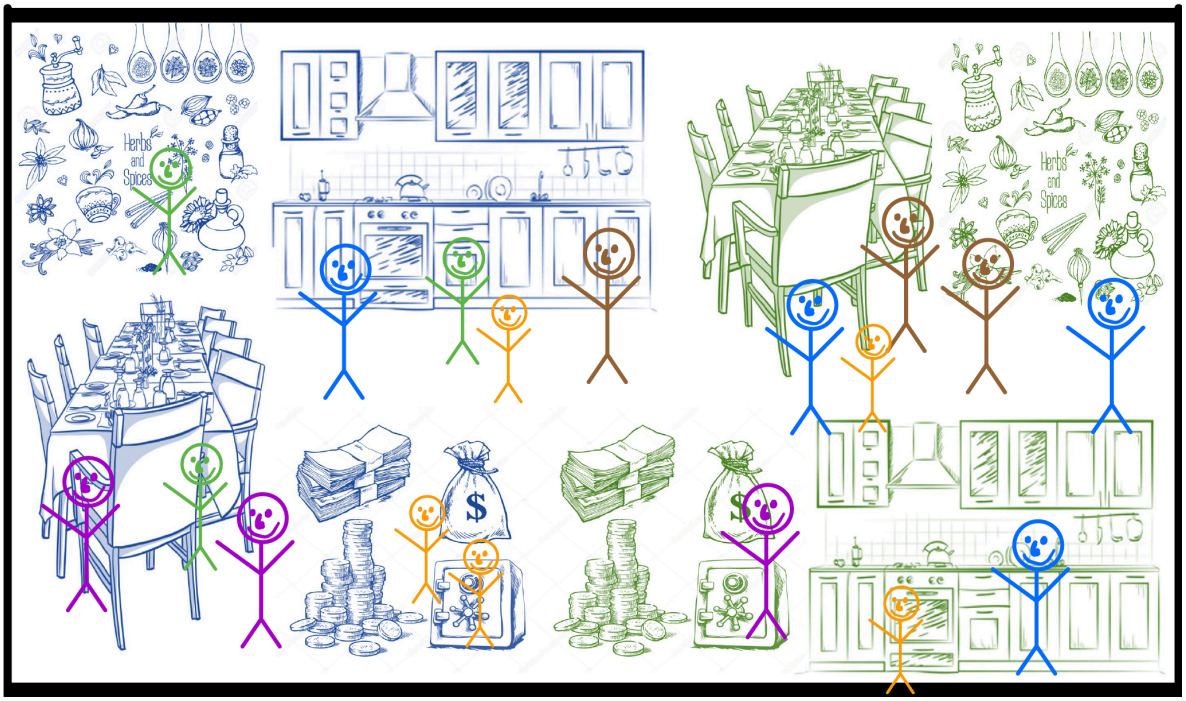
To explain the global infrastructure and security, let's use this mars restaurant project analogy.

3.6.3.1. Story time 🥳

- There is a posting inviting teams interested in setting up a restaurant on mars. Five teams, comprising a total of 25 people, showed up in response to the restaurant setup posting on Mars.



- Only two teams (Team Blue and Green) managed to set up their restaurants, while others couldn't find enough funding to buy everything needed. As you can see below, Team Blue and Green obtained everything they needed (painted in blue and green) for a restaurant, but instead of setting up a building, they set up everything in an open space. The following image shows what it looks like.

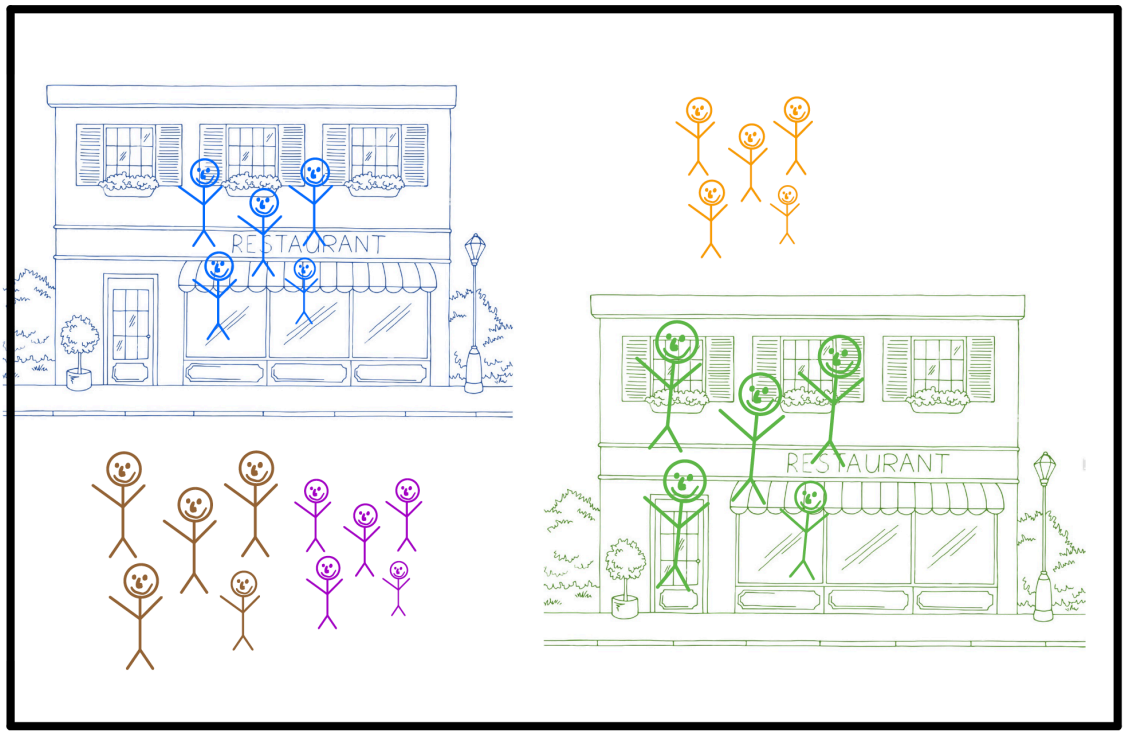


Thoughts/Discussion ?

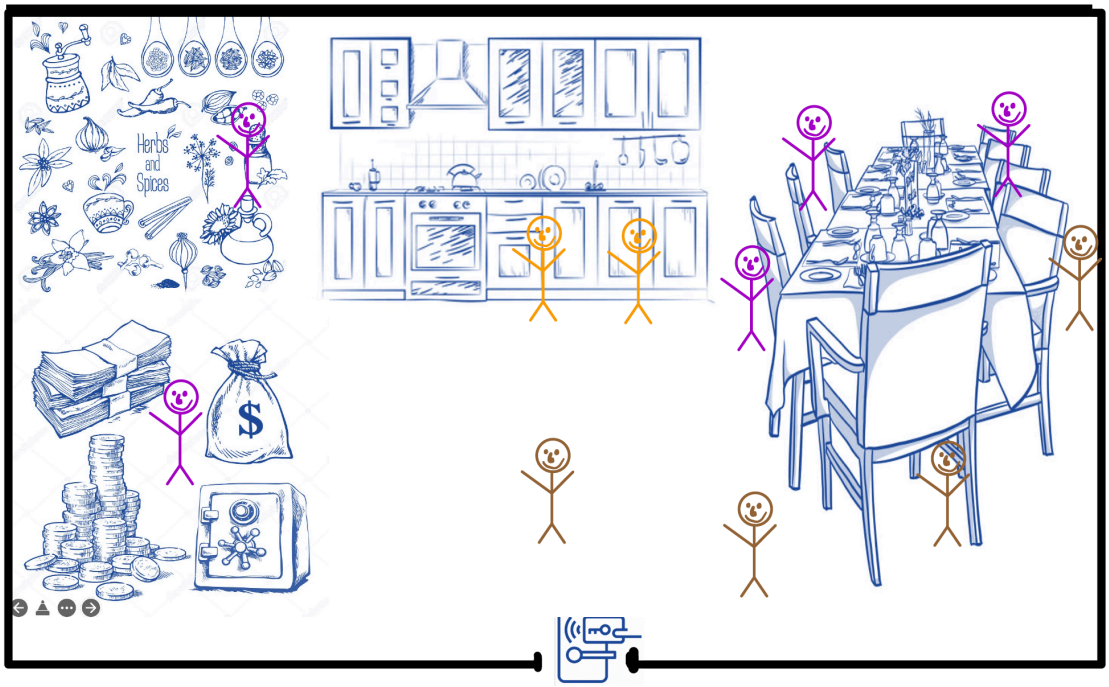
What are the issues with going this way?

- After setting up a private space with walls, both the Blue and Green teams are unable to operate their businesses with the space closed. As a result, they need to open their doors to customers.

> Note: Moving on we will discuss only the restaurant opened by team blue



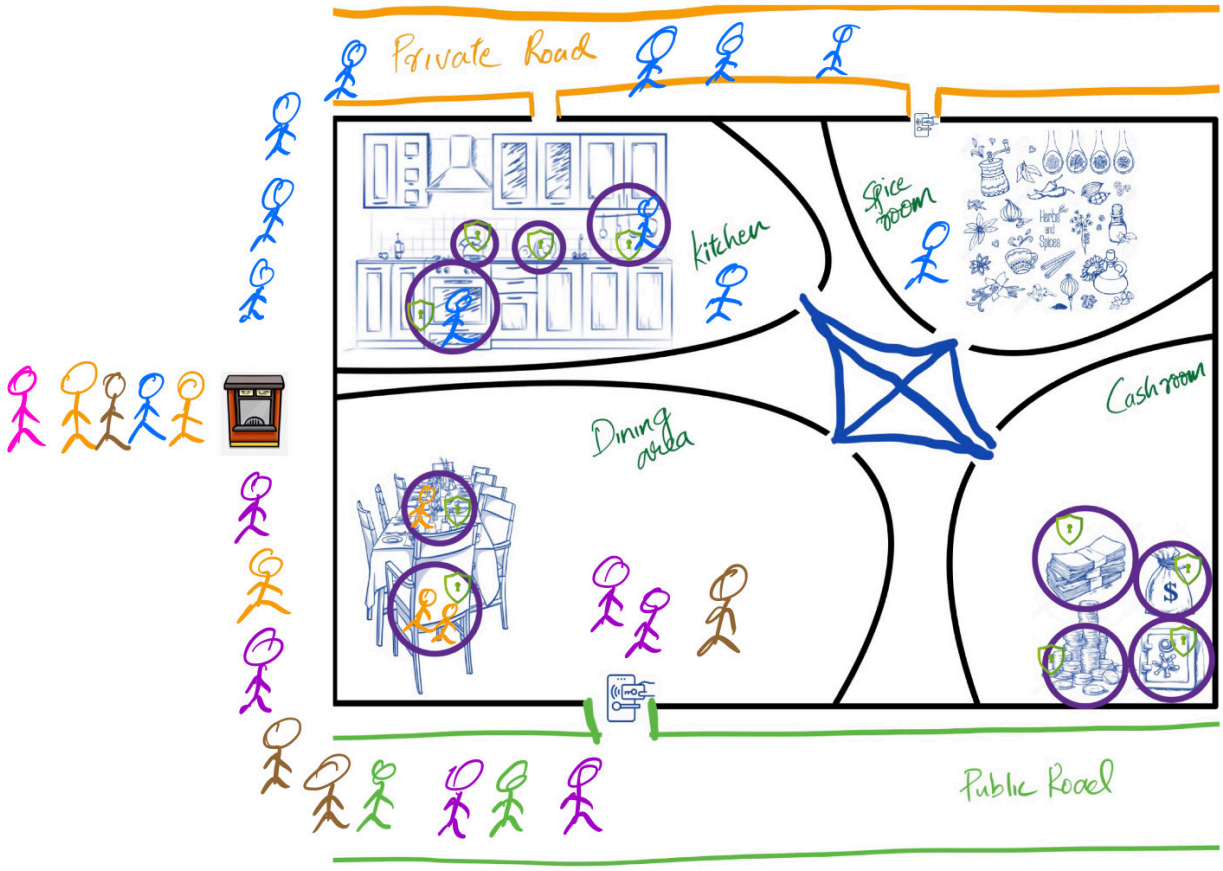
- Blue has opened their doors to everyone, they have chosen to exclude green people, and the reason for their animosity is unclear. Despite this, everyone else is welcome to come into the restaurant.

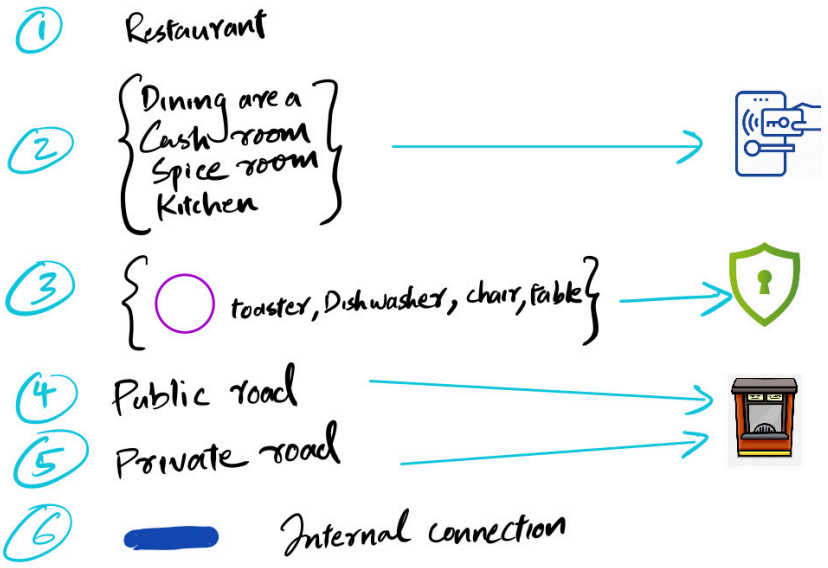


Thoughts/Discussion ?

What are the issues with going this way?

- Let's try a more organized interior.

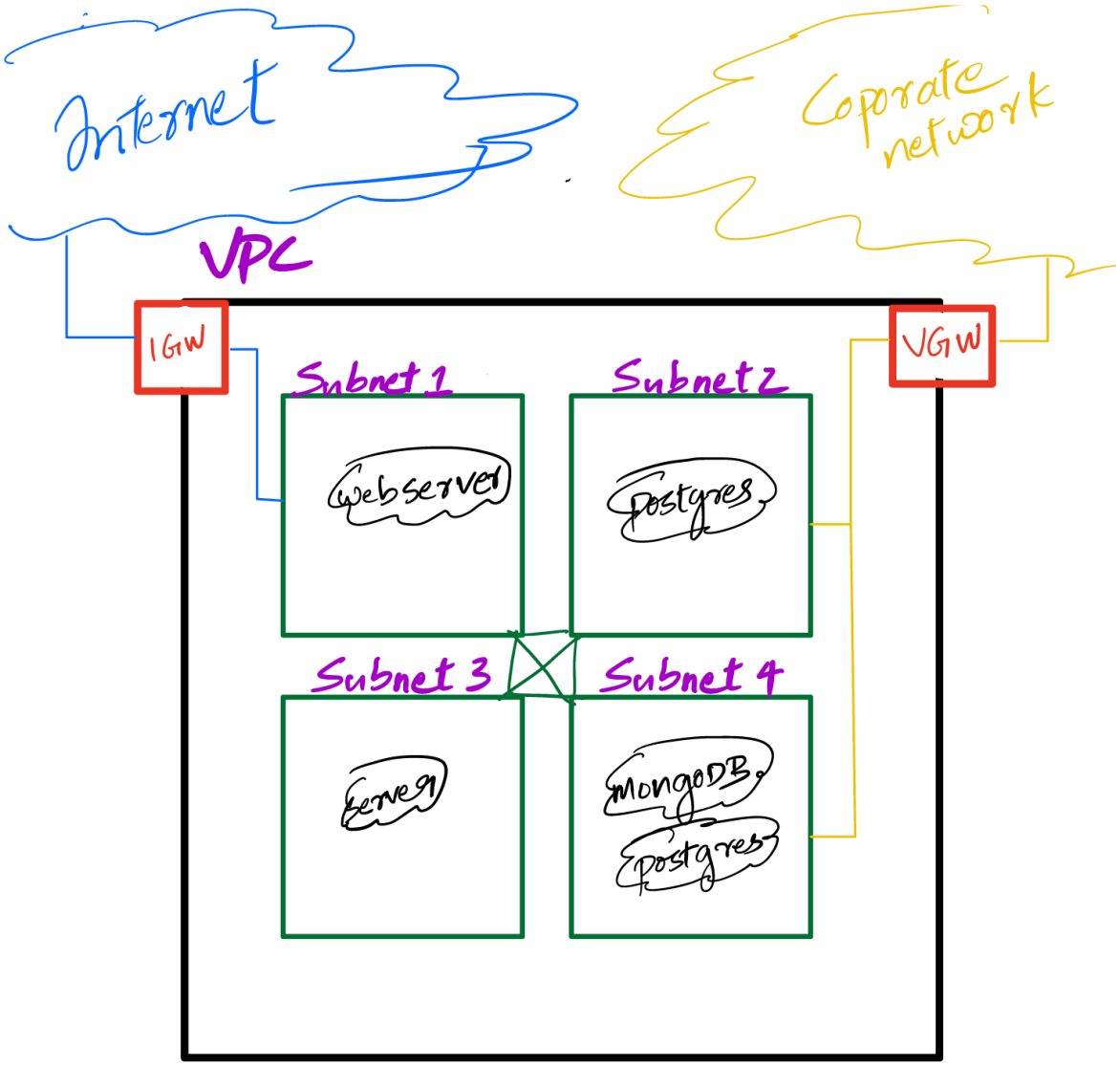




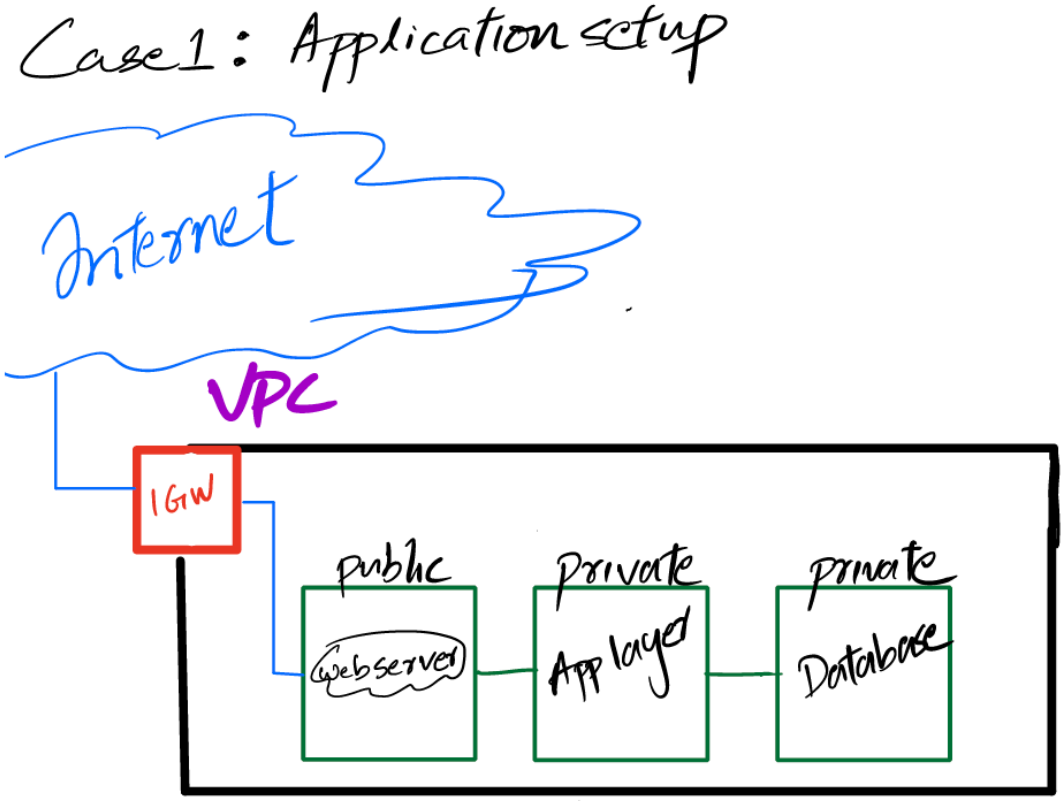
3.6.3.2. Let's get serious

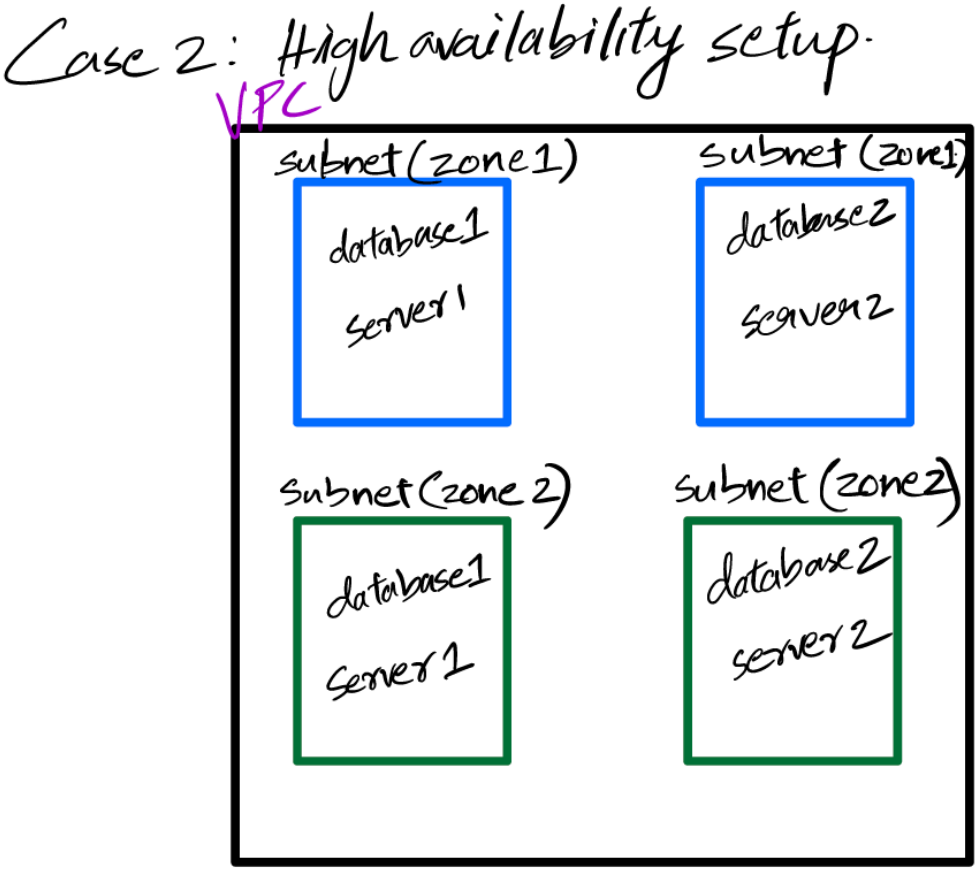
What are the benefits?

How this story relates to AWS infrastructure?

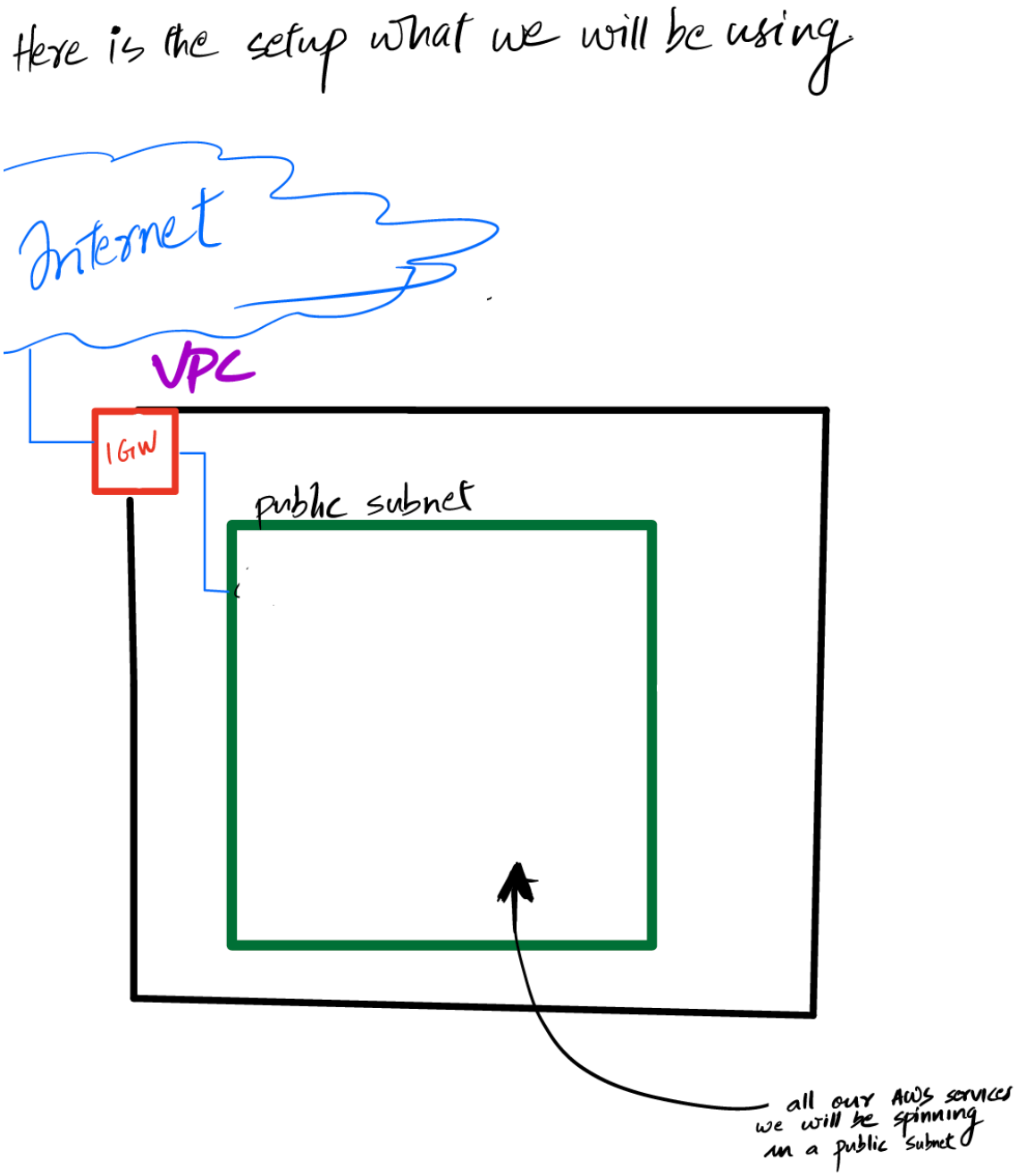


Some real life usecases.

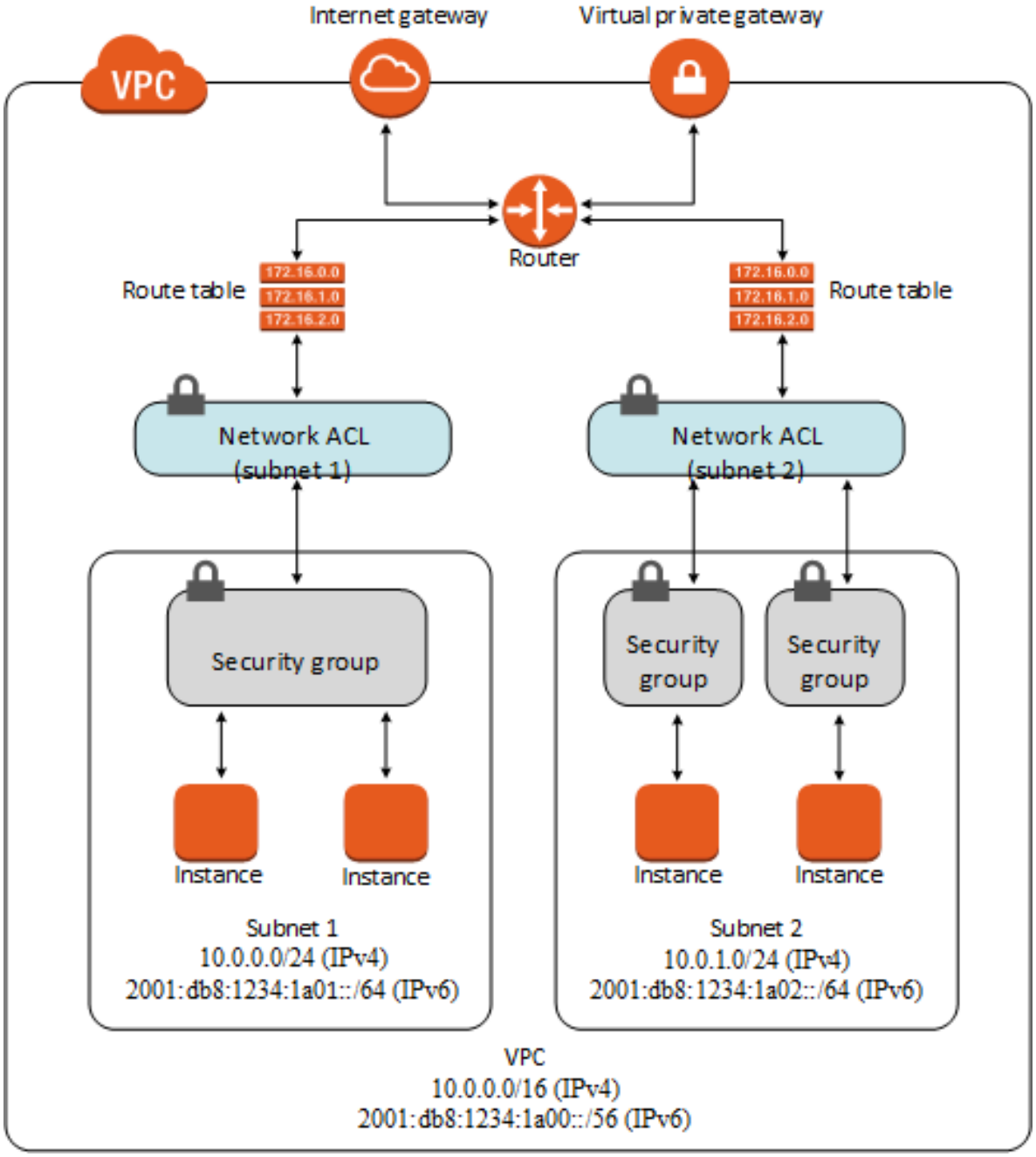




The setup that we will be using



Look for a standard detailed diagram from AWS.



Here are some articles that you can refer to

- Regions and availability zones
- List of regions and availabilty zones
- VPC and subnets
- NACL
- Route tables
- Security groups

3.7. AWS lab setup

This topic is also included in the Week 2 video. Look at the beginning of this lecture for the video zone.

Okay, let's get started by setting up our AWS account. We'll be using something called AWS Academy, which is designed for students. Keep in mind that there are some limitations in using AWS via this, but it should be more than enough for us to learn it. So, let's dive in and get everything set up!

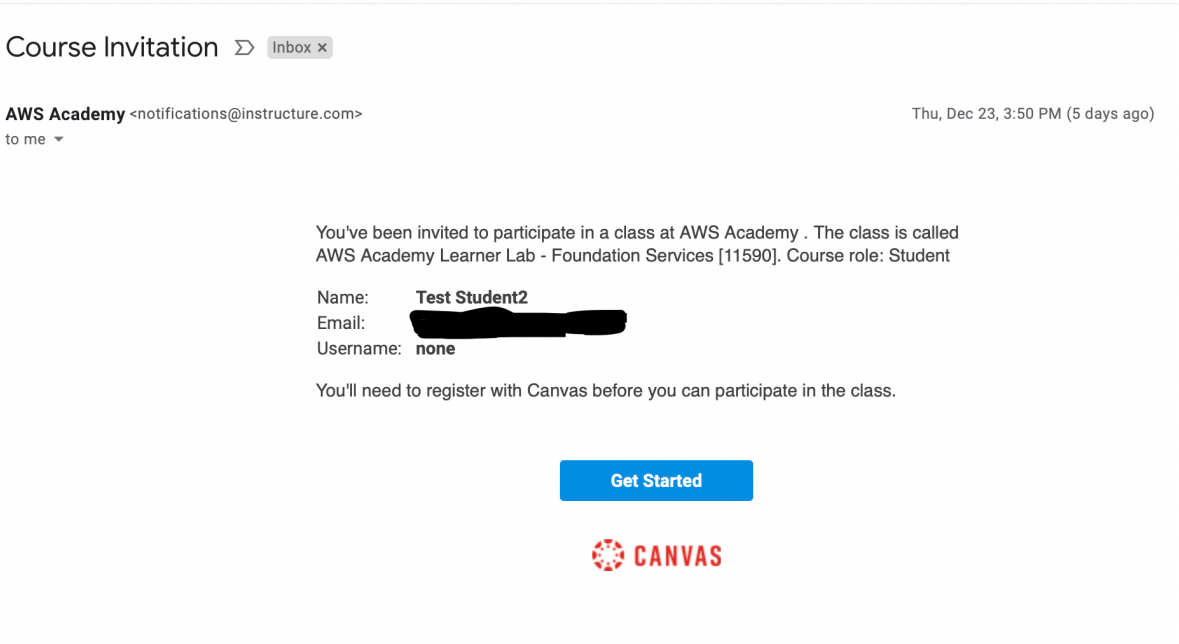
You will be using AWS academy as part of this course and will be given 50$ credit. This is more than sufficient for you to do your projects.

If you get exhausted of this 50$ credit, we won't offer you more credits. So please make sure you stop the instance if you plan to not use it for next 6 hours.

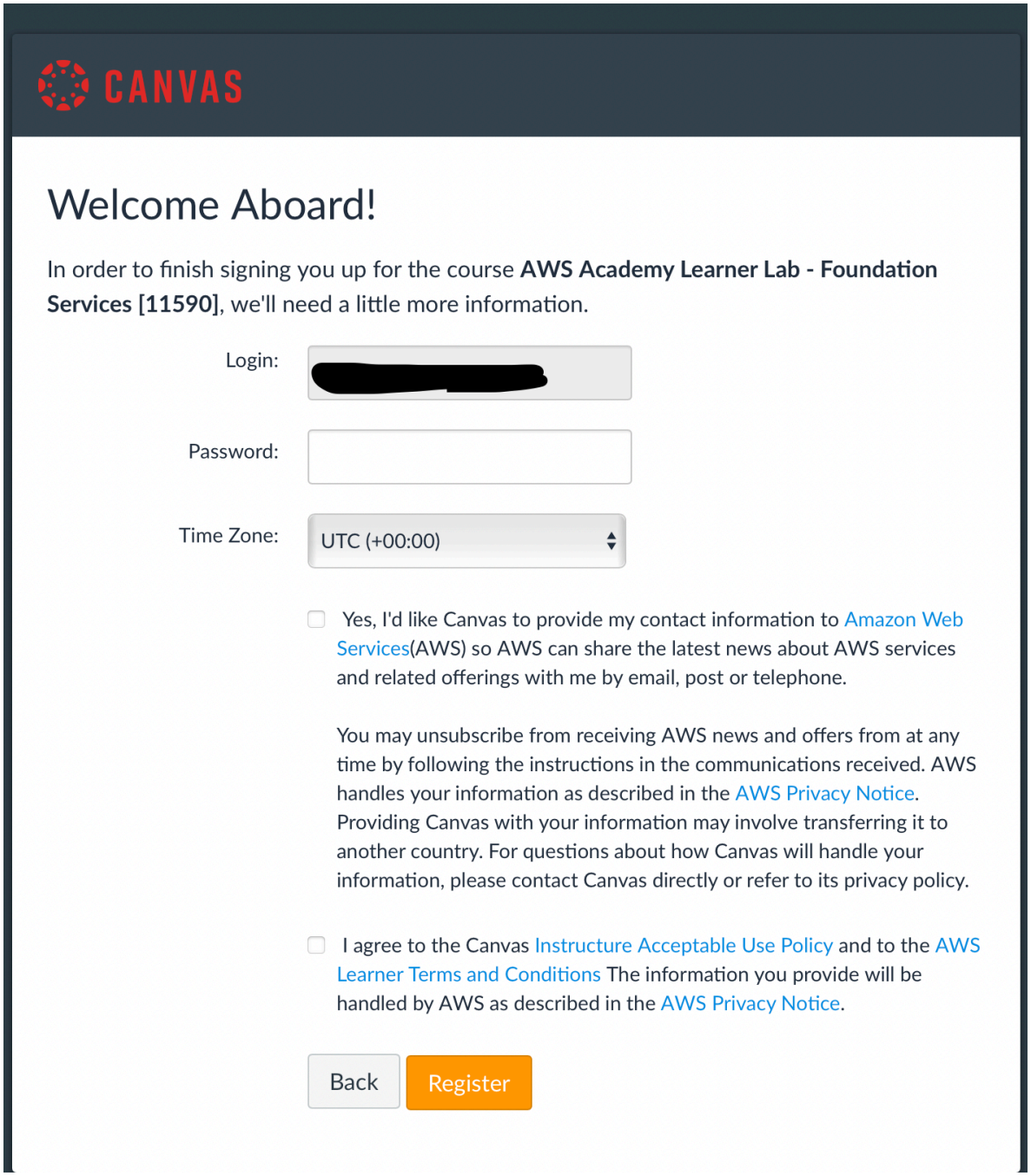
> **Warning**
>
> I recommend you all in using firefox for this course. We need to setup foxyproxy for week3 lab. If you are using chrome, you can use the same instructions as we provide.

Following are instructions to set up your AWS account.

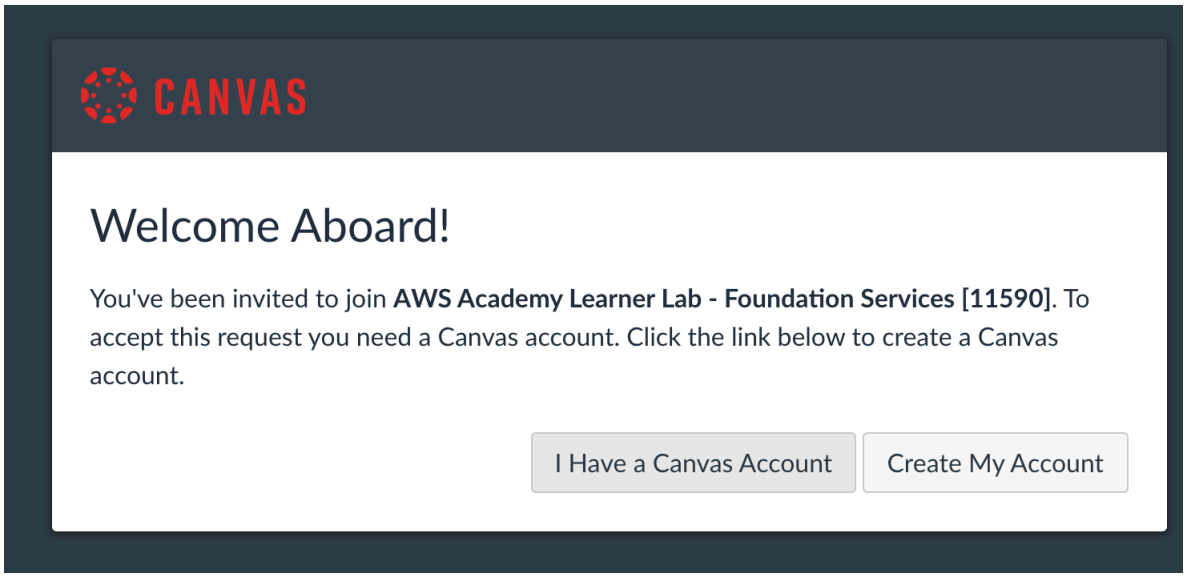
- Check for the welcome email and follow instructions.



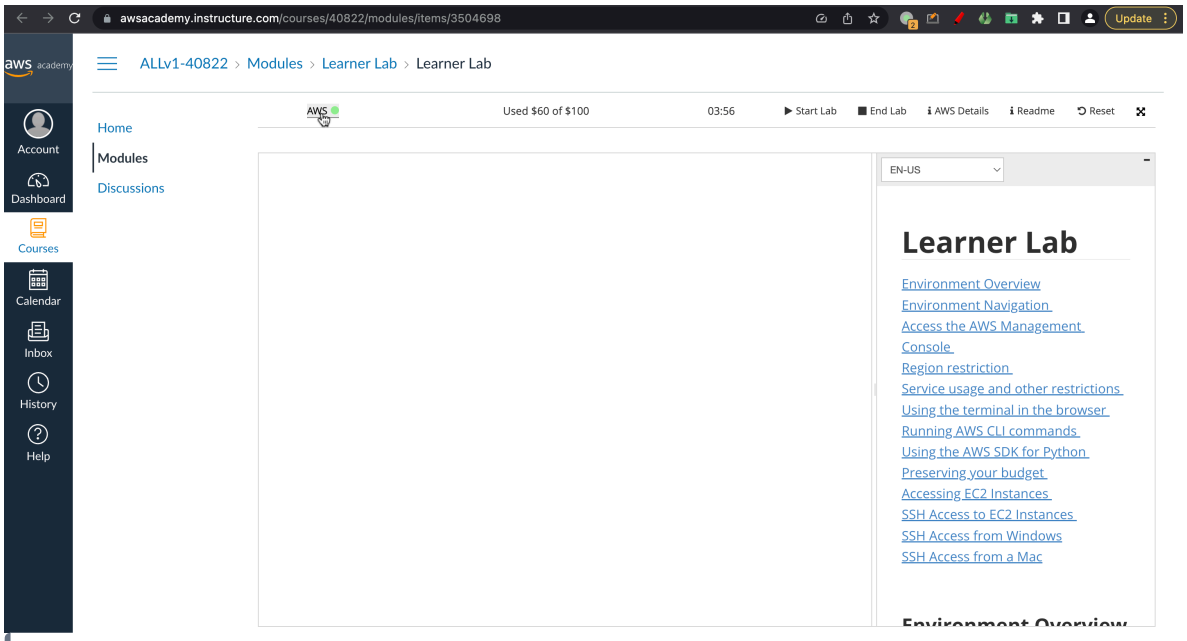
You need to find the above email in your mailbox, click on `Get Started`.



If it asks following for you, press on `Create my account`.



- Later, follow the detailed instructions in this document.
- After following all the instructions above, if everything goes well, you will see a page like the one below. Once you start the lab, you will see a green light next to AWS. Check below screenshot;



> After clicking on `Learner Lab`, if you can't see anything(or show a `refused to connect` error), in most cases, it's that you haven't configured your foxyproxy for firefox.

Warning

- Please don't confuse AWS Academy canvas with the UBC canvas. Both are separate things and are not connected.
- I have issues seeing this page in safari, so I recommend you to use google chrome browser (even that is recommended by AWS).

```
```{important}
- AWS Canvas Labs have a 4-hour time limit, after which it will automatically shut down.
- You can extend the time before the timeout by clicking on "Start Lab" again.
- It's advisable to shut down your instances when not in use to save money (for this offering we only have
- Shutting down the AWS lab doesn't mean that your instances(or any other services) will be shut down auto
```

```{tip}
You can bookmark [this link](https://awsacademy.instructure.com/login/canvas) for easy access to AWS canva
```
```

3.8. Ways to interact with AWS

3.8.1. Web interface

A web-based GUI provides the capability to interact with the services within AWS.

- Login to your learners lab and click on the green AWS button. This will take you to the AWS console. Here you can see all the services that AWS provides, and you can interact with them.

> I will demo this in class.

3.8.2. AWS CLI

This is a command line way to interact with AWS. Click the below toggle for details on installing AWS cli!

This topic is also included in the Week 2 video. Look at the beginning of this lecture for the video zone.

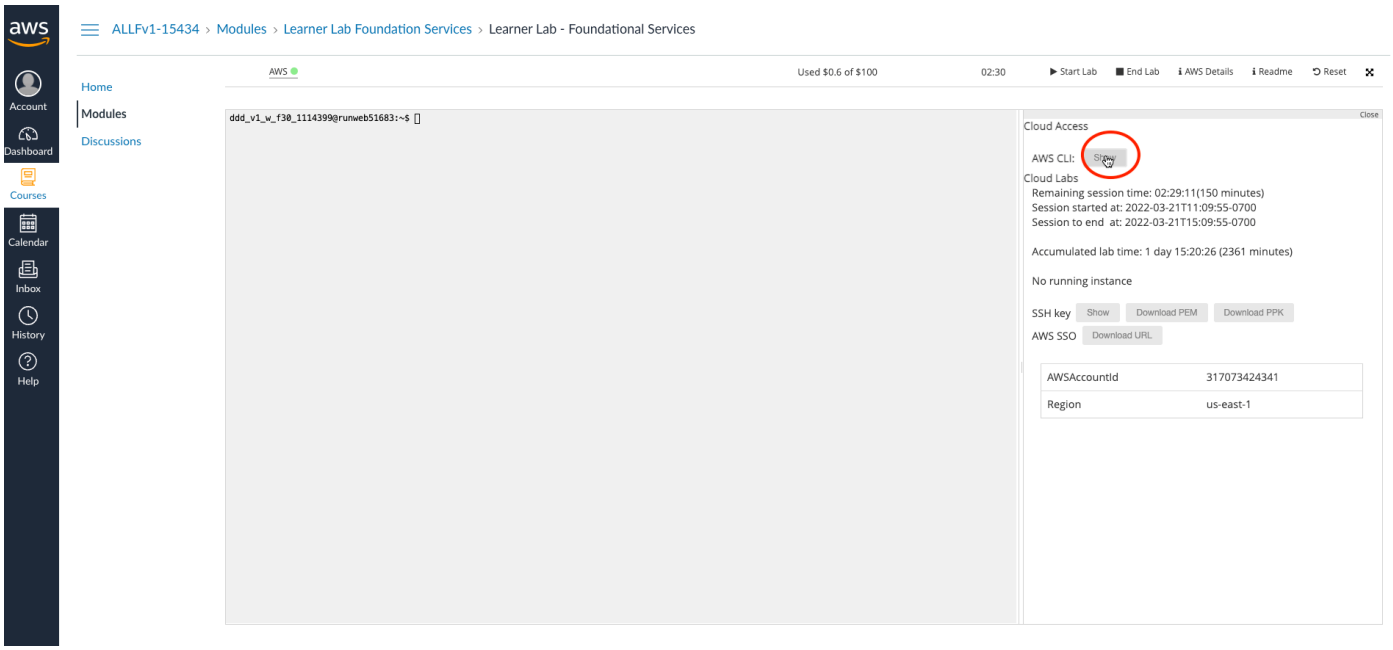
To install AWS CLI check here. Make sure you check your select your operating system.

Here are minimal instructions (for mac users);

1. Download the AWS cli.

```
curl "https://awscli.amazonaws.com/AWSCLIV2.pkg" -o "AWSCLIV2.pkg"
sudo installer -pkg AWSCLIV2.pkg -target /
```

2. With step 1 you have installed AWS cli. Now we need to set it up (configure) to interact with services in your account. We usually need ***access key*** and ***secret***, and usually, you set up this in your account as mentioned in this article. But in our case, we are using a student account or a student way of accessing an AWS account, so we don't have that option mentioned in the article. You need to get ***access key***,***secret*** and ***session token*** (an additional parameter) from the current session in your ***AWS canvas lab console***. Here is the image that you can refer to; click on AWS CLI Show



First to create the file under `~/.aws/credentials` you can run;

```
aws configure
```

and it will ask you to enter the ***access key, secret, region and output format***; Following is what I gave as input. I just gave some value to access key and secret, since you anyways will be copying the contents of the file and pasting it in the file `~/.aws/credentials` in your EC2 instance or the computer you are planning to use AWS cli from.

```
ubuntu@ip-172-31-29-179:~$ aws configure
AWS Access Key ID [None]: x
AWS Secret Access Key [None]: x
Default region name [None]: us-east-1
Default output format [None]: json
```

Important

As they mentioned in AWS CLI from lab console(previous image), copy and paste the contents into `~/.aws/credentials` within your EC2 instance or the computer you are planning to use AWS cli from.

4. ALL DONE !!! AWS cli can be used to interact with many services. Check this out. We will use CLI to interact with s3 and EMR (next lecture).

Just to make sure that you configured it correctly, you can run `aws sts get-caller-identity` and you should see the following. If you see an error, then you need to check your credentials.

```
{
    "UserId": "AROAXEBJAAI5IQJSEEQIX:user1875239=gittu.george@gmail.com",
    "Account": "489712255546",
    "Arn": "arn:aws:sts::489712255546:assumed-role/voclabs/user1875239=gittu.george@gmail.com"
}
```

```
```{warning}
Unfortunately, with the student setup we are having, the ***access key***,***secret***, and ***session tol
```
```

3.8.3. SDK

We won't be using it much. A good blog here explains various ways of interaction.

3.8.4. Identity and access management (IAM) (OPTIONAL)

- Create an account in AWS (root user)
  - Root user
  - IAM user
- Setup IAM (Identity and access management) users
  - Create ***groups*** based on the responsibilities (eg: Administrators/ Students)
  - Attach a ***policy*** to the group
  - Managed polices (eg: AdministratorAccess, AmazonS3FullAccess)
  - Custom/ User defined policies
  - Add the ***users*** to the respective groups

3.9. What we learned today?

- Details on cloud computing.
  - WHAT, WHY, and HOW it is made possible
- Organization of cloud computing
  - Regions
  - Availability zones
- We learned `Networking and Content Delivery` services provided by AWS
  - VPC
  - subnets
  - Route table, Security table, and NACL
- We learned about various ways of interaction with AWS
  - Web interface
  - AWS CLI
  - SDK
- We learned about the shared responsibility model
- We learned about `Security, Identity, and Compliance` services by AWS (optional)
  - IAM