VPC Step By Step for AWS

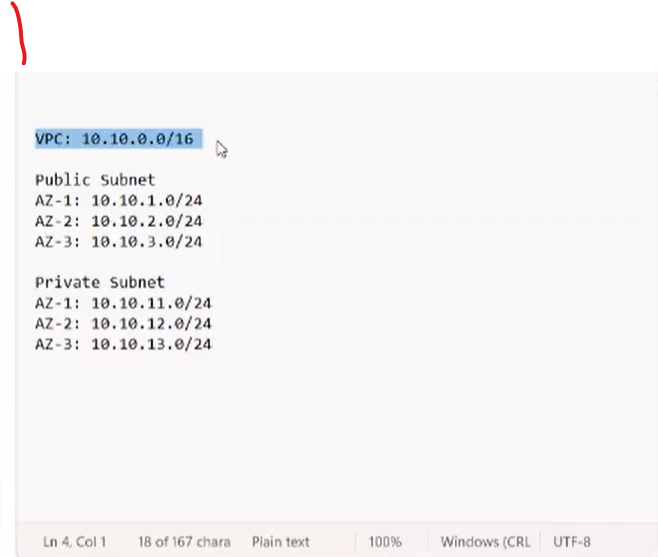
1. In AWS **Create VPC**
2. VPC Settings: **VPC and more**
3. Name tag auto-generation: (Example **Class7test\_public**
4. IPv4 CIDR block: **10.10.0.0/16** (This should be taken from your Network Planner!)
5. IPv6 CIDR block: Should be No IPv6 CIDR block (Leave alone) Tenancy: **Default**
6. Number of Availability Zones (AZs): **3**
7. Number of public subnets (This number should correlate with the number you have for Availability Zones): **3**
8. Number of private subnets (This number should correlate with the number you have for Availability Zones): **3**
9. Customize subnets CIDR blocks: **Public 10.10.1.0 /24**

**10.10.2.0 /24**

**10.10.3.0 /24**

**Private 10.10.11.0 /24**

**10.10.12.0 /24**

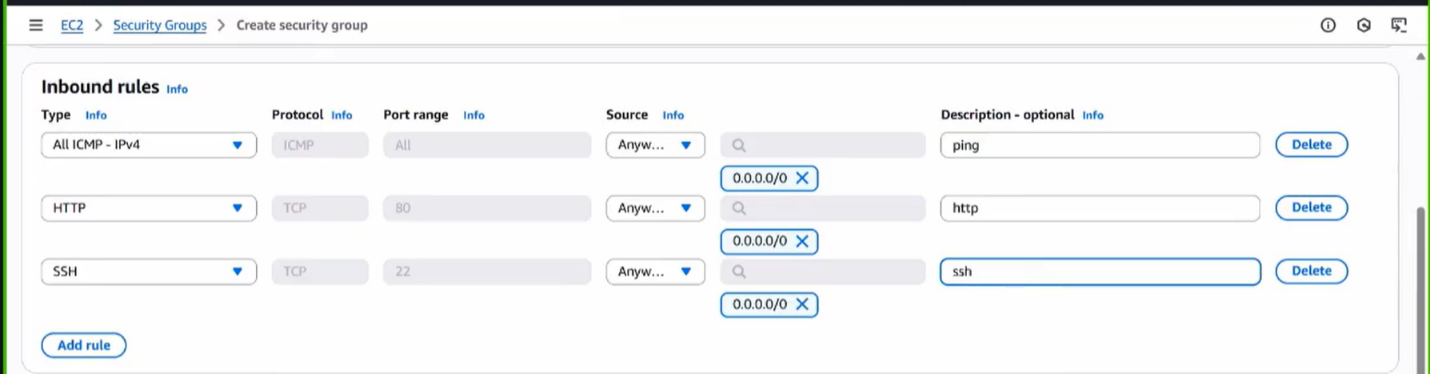
**10.10.13.0 /24  This should correspond with your Planner**

1. NAT gateways: **In 1 AZ** (We only need 1 to be a gateway to the internet. Cost)
2. VPC endpoint: **None**
3. DNS options: **Don’t Touch**
4. Additional Tags: Optional
5. **\*\*\*Last Check\*\*\* CREATE VPC**
6. **CREATE VPC**
7. Time to make an Instance. EC2
8. First, we make a Security Group (One public and One private)
9. **CREATE A SECURITY GROUP**
10. **Basic Details: Security group names:** (Let’s do PUBLIC first)

**Class7PUBLIC** copy to **Description**

1. **VPC \*\*\*USE THE VPC YOU CREATED\*\*\***
2. **Inbound Rules:** You will add 1 Inbound Rule with **3 Types** within.

**All ICMP- IPv4, HTTP** and **SSH Source:** All of them will be **Anywhere-IPv-4**

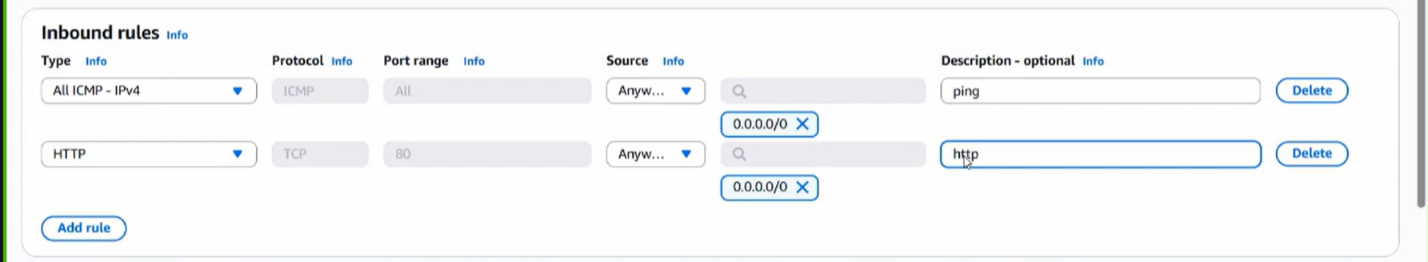
****

1. **\*\*\* DO NOT TOUCH OUTBOUND RULES!!!\*\*\***
2. **Tags:** Optional
3. Check and **CREATE A SECURITY GROUP**
4. **CREATE A SECURITY GROUP** (Yes. We are making another Security Group)
5. **Basic Details: Security group names:** (This will be a PRIVATE Security Group)

**Class7PRIVATE** copy to **Description**

1. **VPC** **\*\*\*USE THE VPC YOU CREATED\*\*\***
2. **Inbound Rules:** You will add 1 Inbound Rule with **2 Types** within.

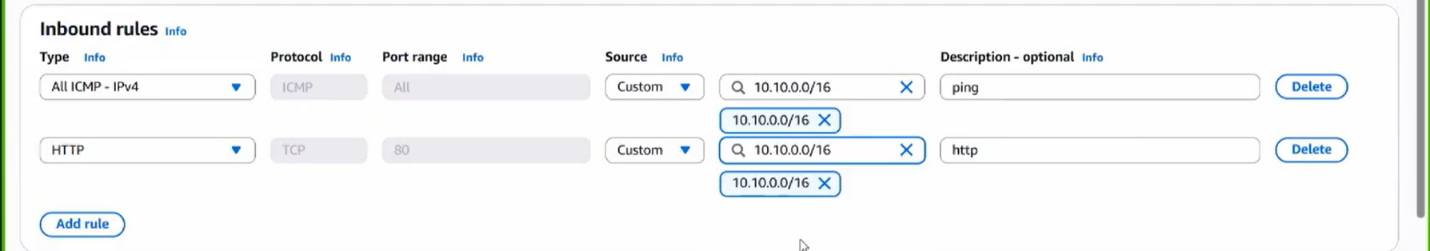
**All ICMP- IPv4, HTTP**  **Source:** All of them will be **Anywhere-IPv-4**

****

**OR**

**CHANGE THE SOURCE TO CUSTOM AND USE**

**YOUR IPv4 CIDR block: 10.10.0.0/16**

****

**THIS ADDS A BETTER LAYER OF SECURITY**

**ACTING LIKE A FIREWALL**

**WE DON’T WANT ANYONE OUTSIDE TO ACCESS**

**OUR PRIVATE SUBNET**

1. **\*\*\* DO NOT TOUCH OUTBOUND RULES!!!\*\*\***
2. **Tags:** Optional
3. Check and **CREATE A SECURITY GROUP**
4. Time to create our **INSTANCES.** We will need to create 2 of them. A **PUBLIC**

and a **PRIVATE** one.

1. **EC2,** **INSTANCES,** **LAUNCH INSTANCE** (Let’s do a **PUBLIC INSTANCE** first)
2. **Names and Tags** (Name your Instance. Include **PUBLIC** somewhere in it.)
3. **Scroll down to Create key pair.**
4. Give the Key Pair a name. **LEAVE EVERYTHING ELSE ALONE! Create Key Pair**
5. **NETWORK SETTINGS: EDIT**
6. **VPC: \*\*\*USE THE VPC YOU CREATED\*\*\***
7. **SUBNET:** Since we are creating a **PUBLIC INSTANCE**. Choose 1 **PUBLIC SUBNET**

**REMEMBER YOU CREATED 3 OF THEM**

1. **AUTO-ASSIGN PUBLIC IP: ENABLE**
2. **FIREWALL (SECURITY GROUP): SELECT EXISTING SECURITY GROUP**

**CHOOSE THE ONE YOU CREATED THE PUBLIC ONE**

1. **SKIP DOWN TO ADVANCE DETAILS: USER DATA – OPTIONAL**
2. **ADD YOUR SCRIPT (ec2scrpit At this point we should all know how to do this)**
3. **CHECK YOUR WORK. LAUNCH INSTANCE**
4. Let’s make anotherInstance, a **PRIVATE** one.
5. **INSTANCES, LAUNCH INSTANCE**
6. **Names and Tags** (Name your Instance. Include **Private** somewhere in it.)
7. **KEY PAIR: Because this is a Private Instance, we can use**

**Proceed without key pair (not recommended)**

1. **NETWORK SETTINGS: EDIT**
2. **VPC: \*\*\*USE THE VPC YOU CREATED\*\*\***
3. **SUBNET:** Since we are creating a **PRIVATE INSTANCE**. Choose 1 **PRIVATE SUBNET**

**ev**

1. **CHOOSE THE ONE YOU CREATED THE PRIVATE ONE THAT CORRESPONDS**

**WITH YOUR PUBLIC ONE** (ie: Look at your CIDR. If **PUBLIC** was **10.10.1.0/24**

your **PRIVATE** should be **10.10.11.0/24**

1. **AUTO-ASSIGN PUBLIC IP: DISABLE**
2. **FIREWALL (SECURITY GROUP): SELECT EXISTING SECURITY GROUP**

**USE THE PRIVATE ONE YOU CREATED**

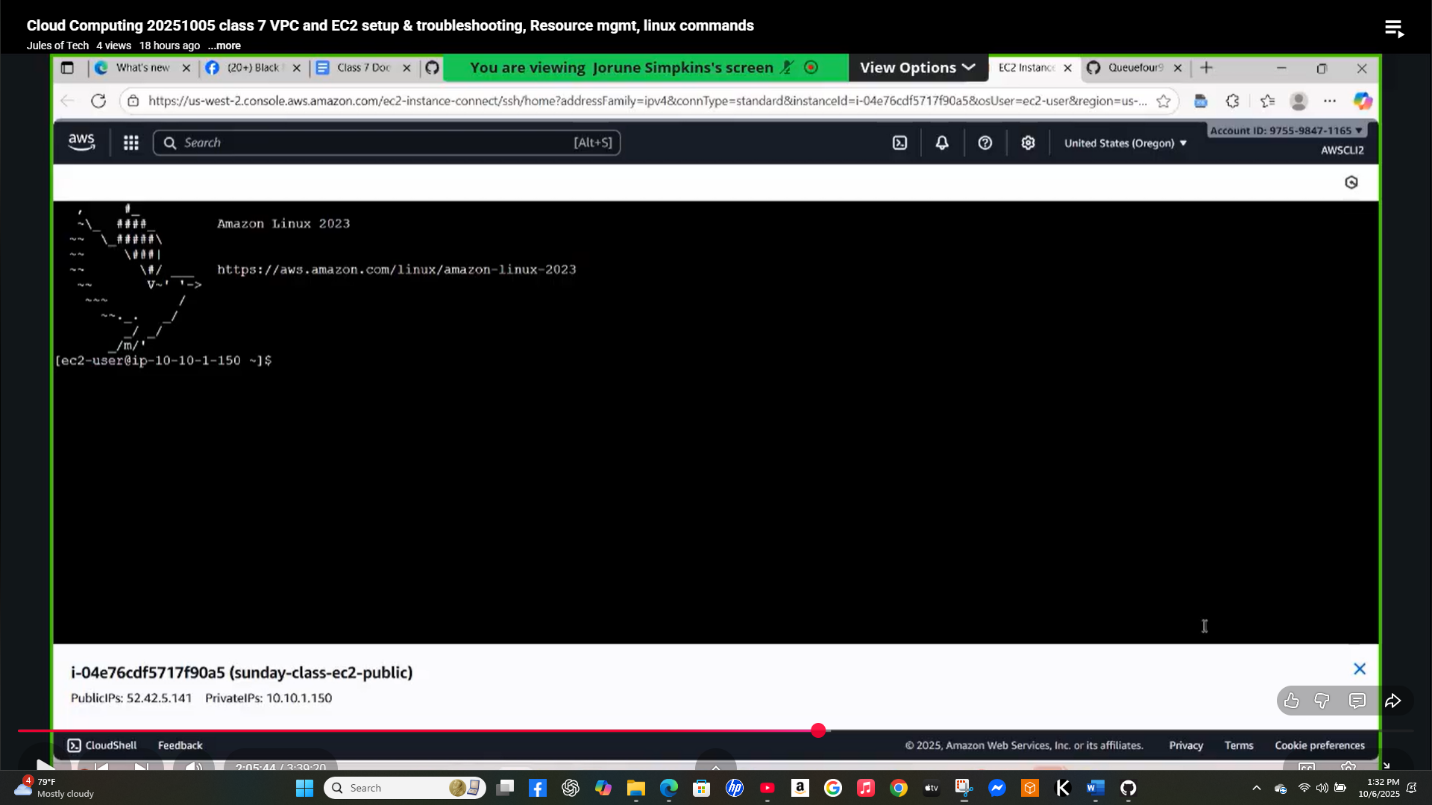
1. **SKIP DOWN TO ADVANCE DETAILS: USER DATA – OPTIONAL**
2. **ADD YOUR SCRIPT (ec2scrpit At this point we should all know how to do this)**
3. **CHECK YOUR WORK. LAUNCH INSTANCE**
4. At this point our **Public Instance** should be ready. We can check and see if

Everything is running good by way of **Copy the DNS of the Public Instance**

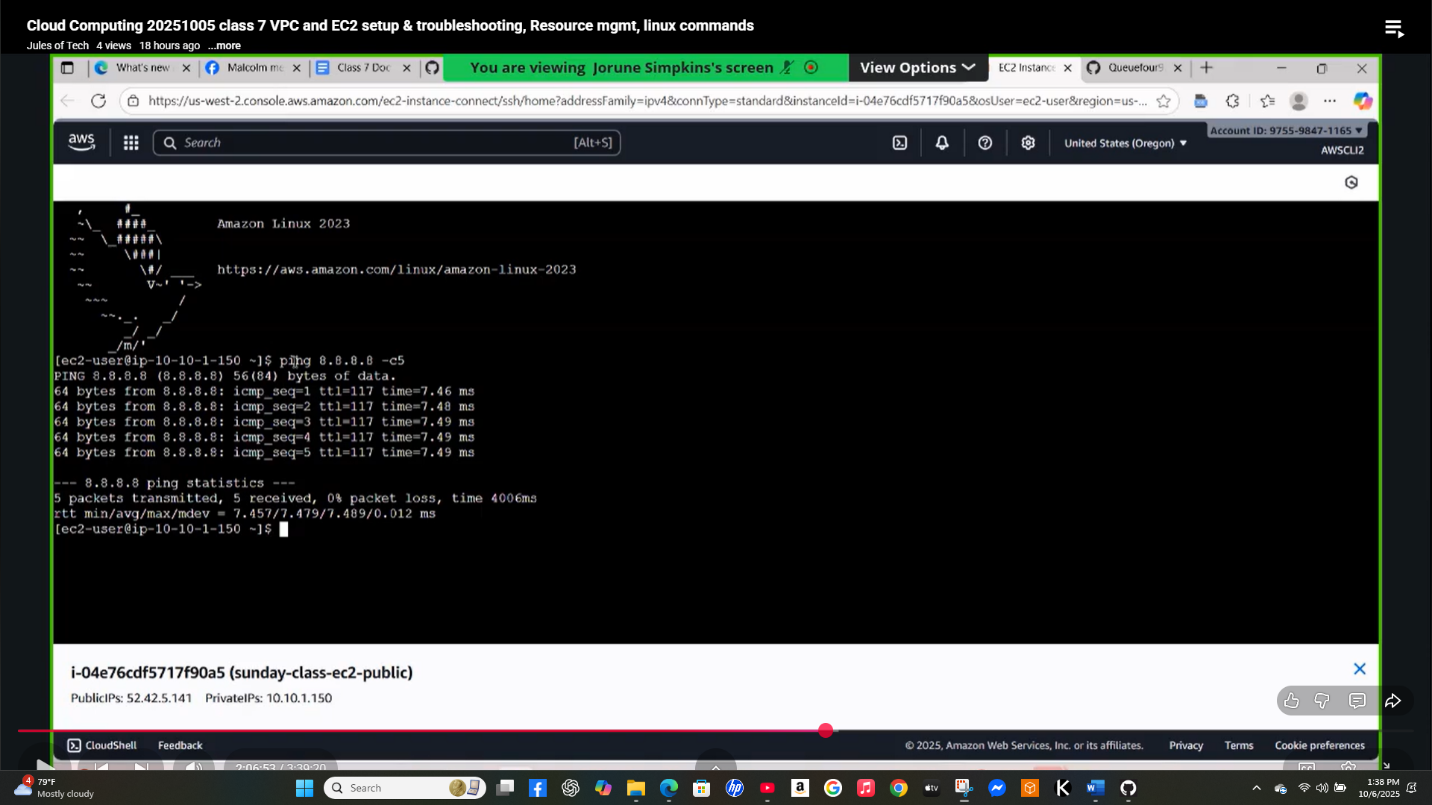
**and use http:// or we can use Connect. Let’s use http://**

**Looks great!**

1. Now let’s **CONNECT. NO NEED TO CHANGE ANYTHING HERE!** Hit **CONNECT**



1. Great job if you have a screen similar to the image above.
2. Now type **ping 8.8.8.8 -c5** (Goggle will ping 5 times for any errors)

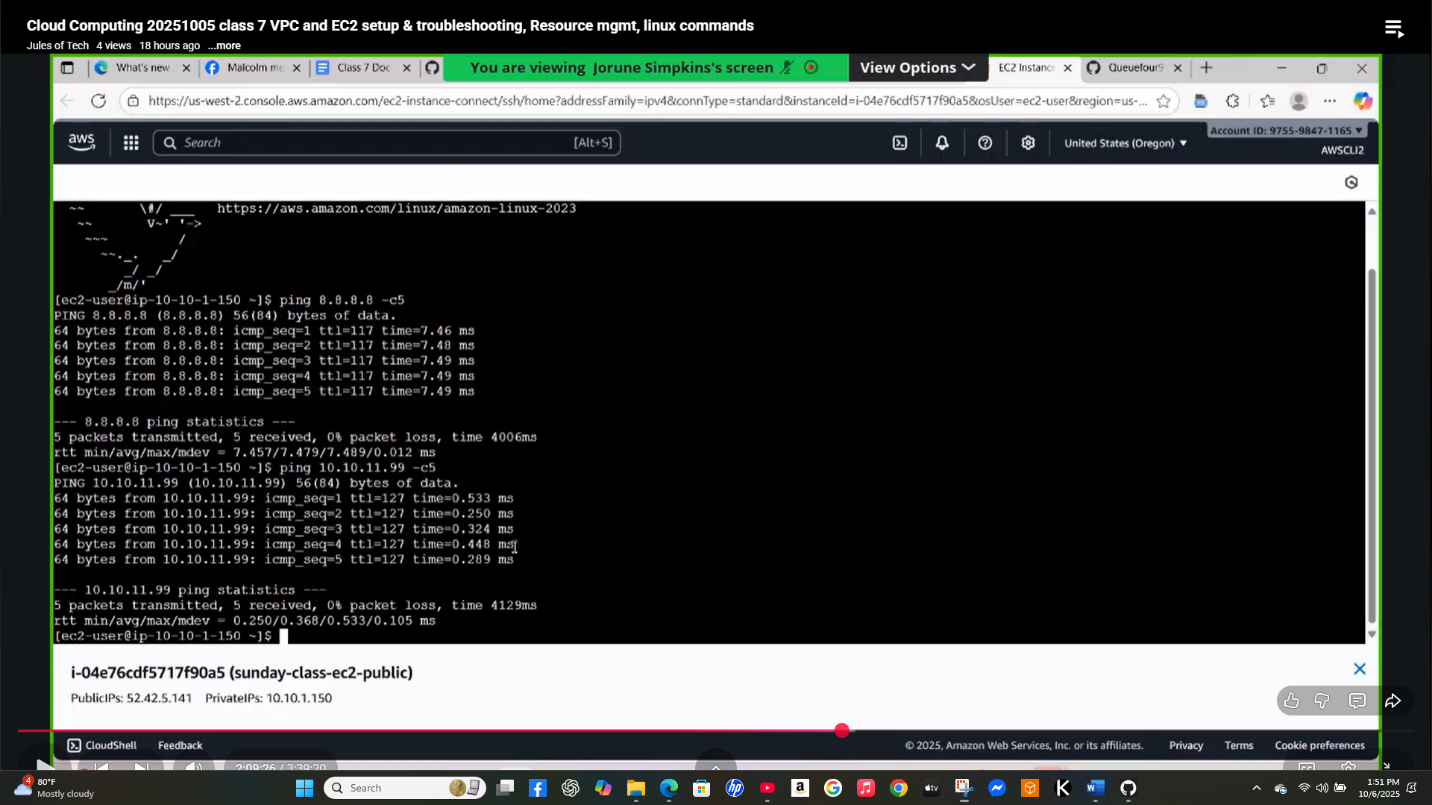


1. Go back to your **Instances. Using only the PRIVATE INSTANCE** (uncheck the

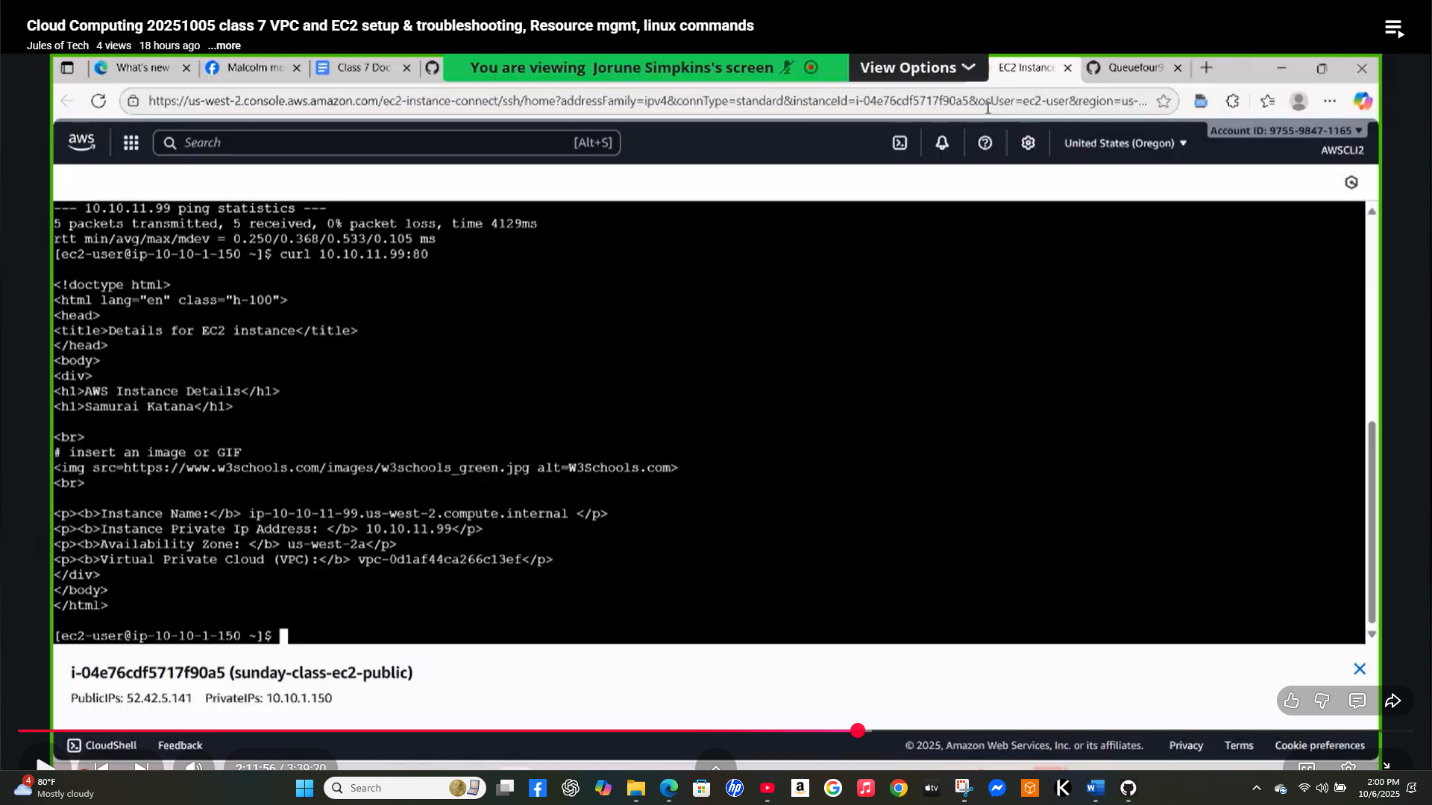
**PUBLIC)** we are going to **ping** the **Private from the Public**

1. **ping** **(Private IPv4 Addresses given from your Private Instance) -c5**

If all goes well, it should look similar to the screenshot below.

****

1. Let’s use the **curl** command. Type **curl (Private IPv4 Addresses given from your Private Instance):80**

****

**GREAT JOB! EVERYTHING WORKS!**

1. **TEARDOWN!**
2. Terminate your EC2s
3. Go to **VPC, NAT Gateways DELETE NAT GATEWAYS (WAIT UNTIL IT’S FULLY**

**DELETED)**

1. Go to **Elastic IPS. Check your IP and in Actions Release Elastic IP Addresses**
2. Go to **Your VPCs**, in **Actions DELETE ONLY YOUR VPC! DO NOT TOUCH THE**

**DEFAULT VPC!**

1. Finished