Task 5.3D

CLOUD COMPUTING

Creating a VPC Peering Connection

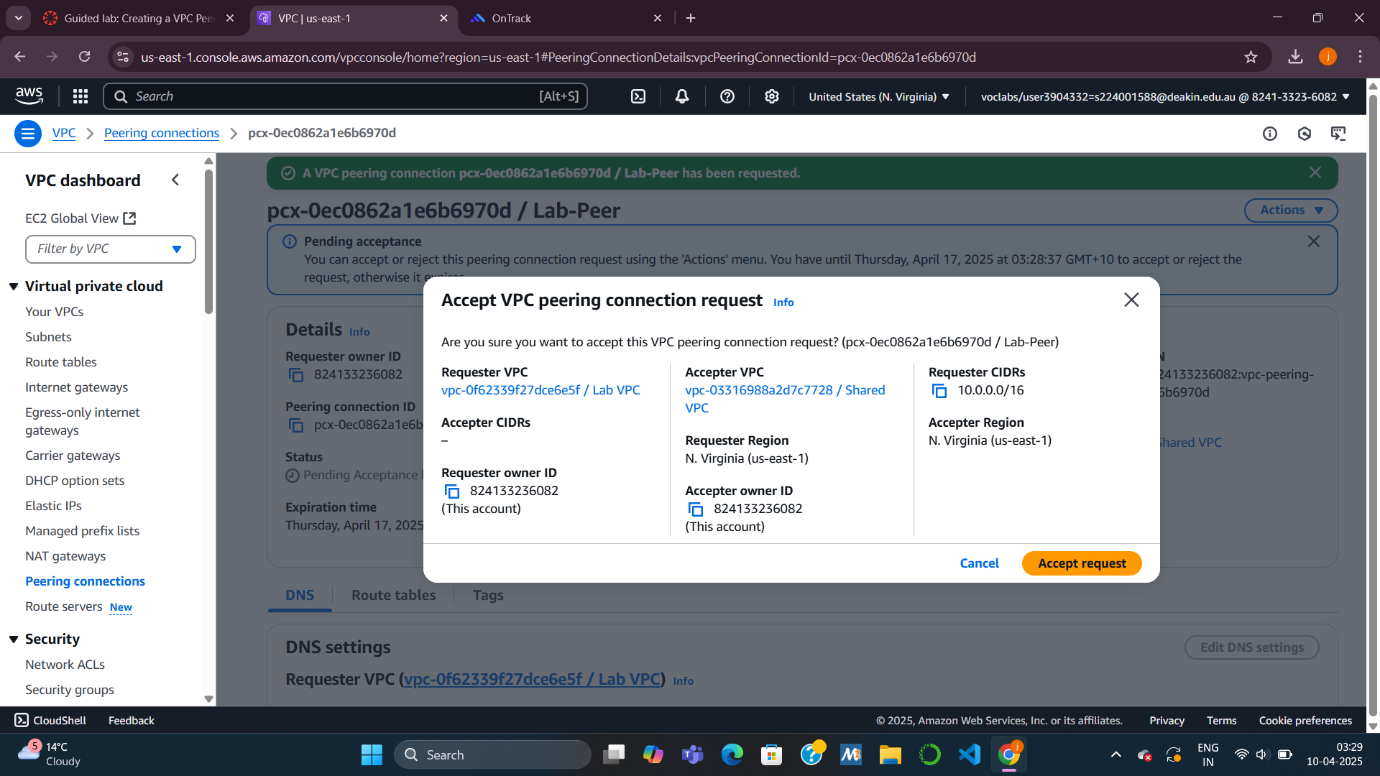
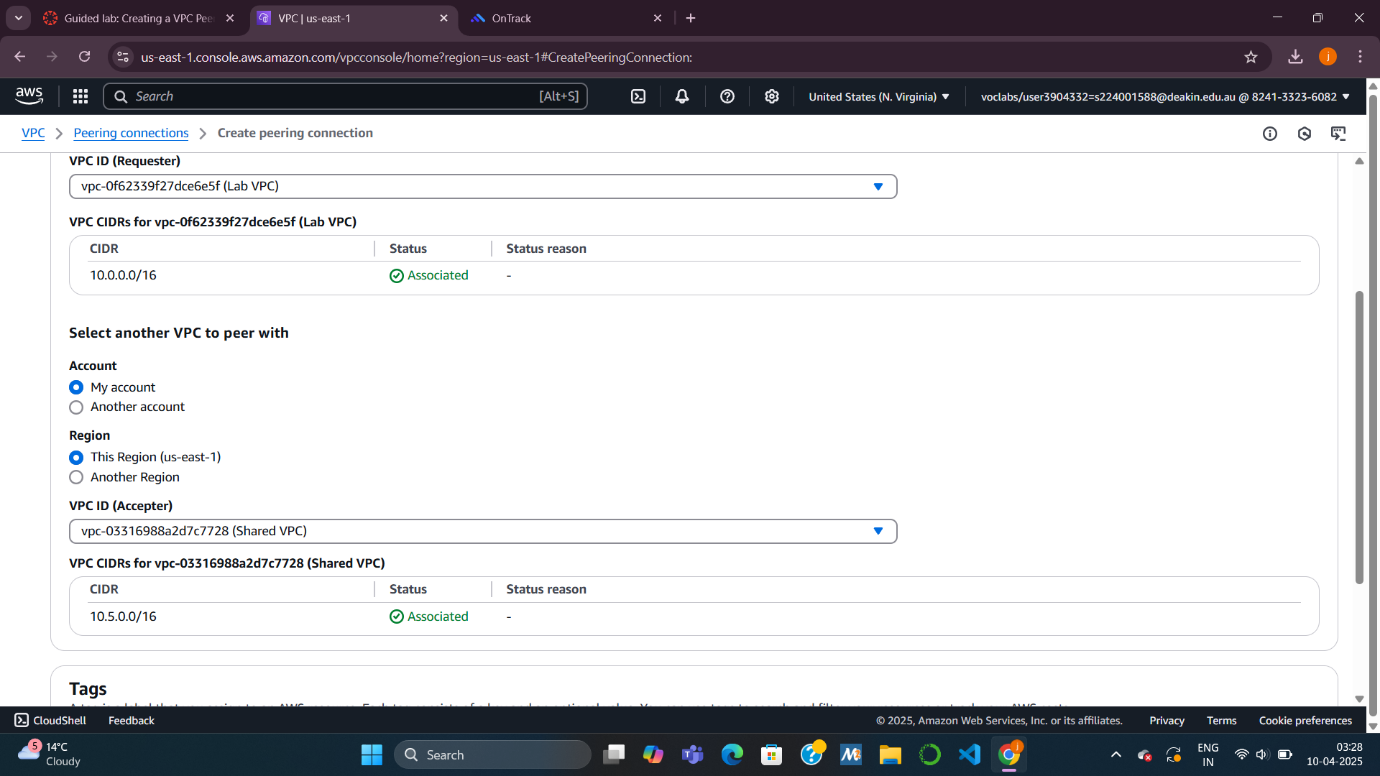
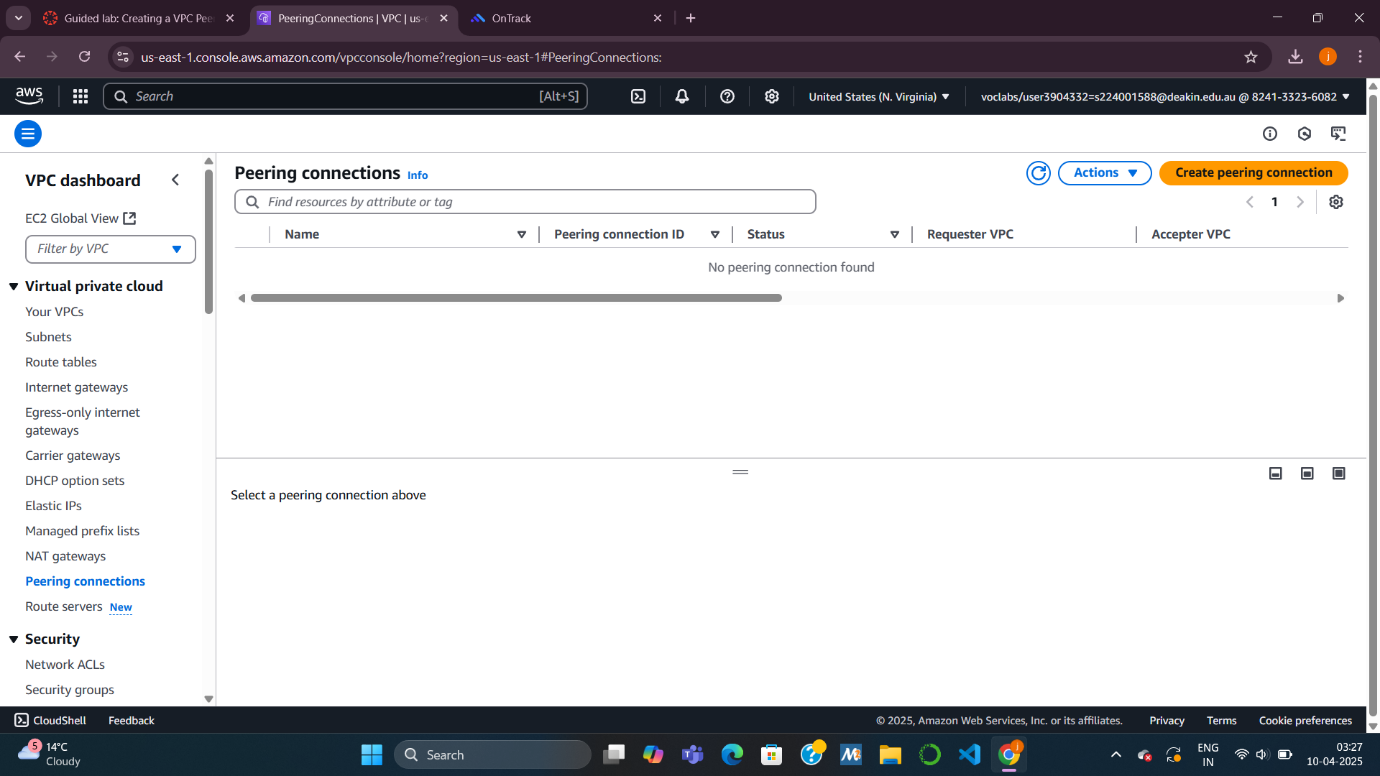
**Task 1: Creating a VPC Peering Connection**

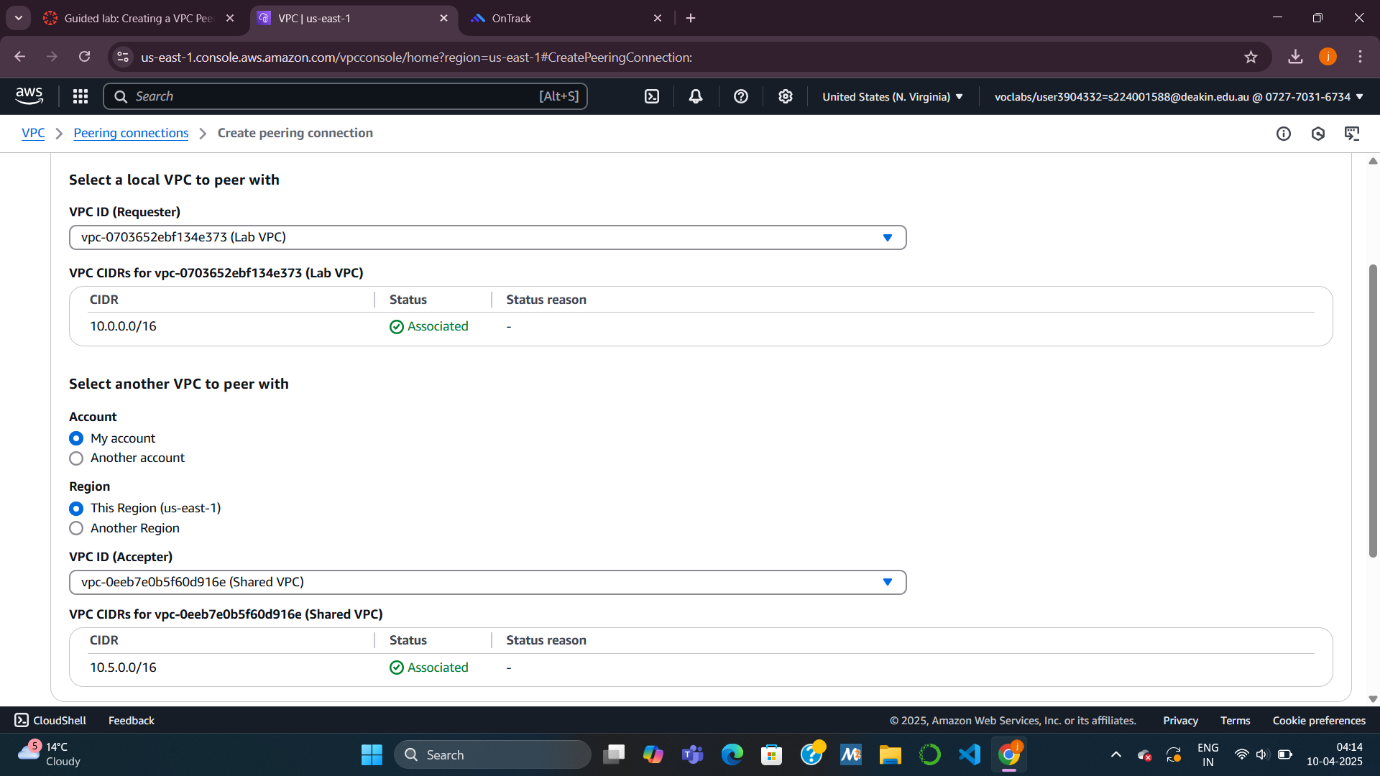
**What you’re doing:**

Establishing a **private link** between two VPCs—**Lab VPC** (has an EC2 application) and **Shared VPC** (has a database).

**Steps:**

* Open the **VPC dashboard**.
* Go to **Peering Connections**, click **Create**.
* Name it Lab-Peer.
* Select Lab VPC as **Requester**, Shared VPC as **Accepter**.
* Click **Create**.
* After creation, **accept the request** to activate the connection.





**Task 2: Configuring Route Tables**

**What you’re doing:**

Telling each VPC **how to reach the other** using the new peering connection.

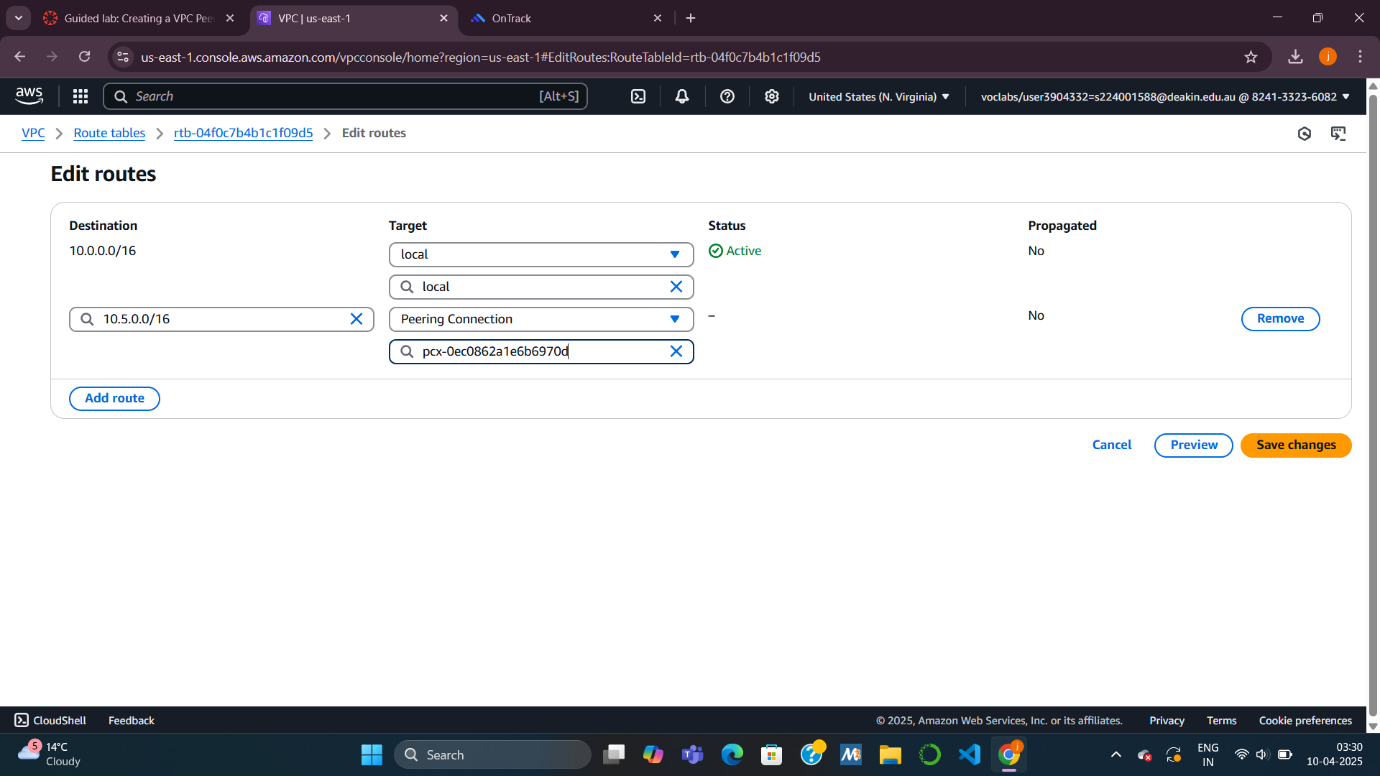
**Subtasks:**

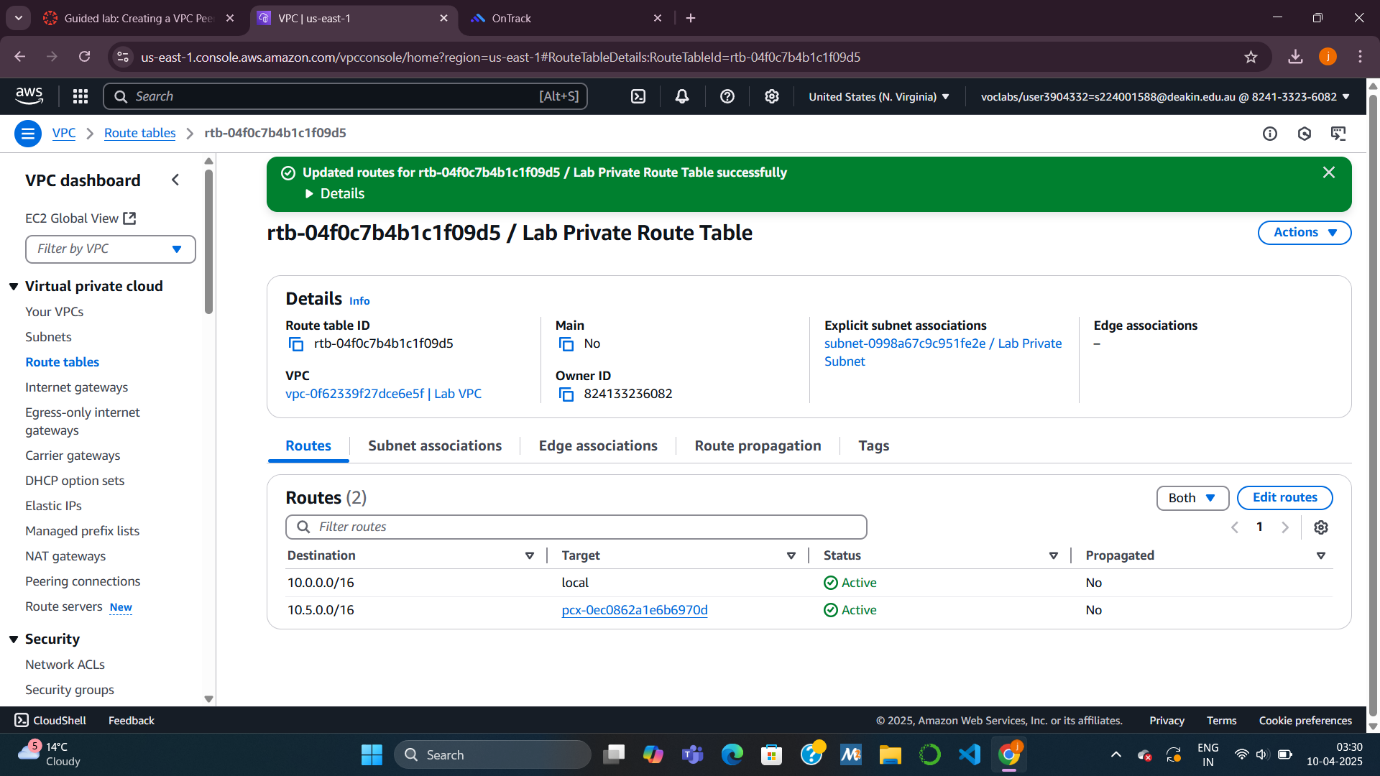
**For Lab VPC:**

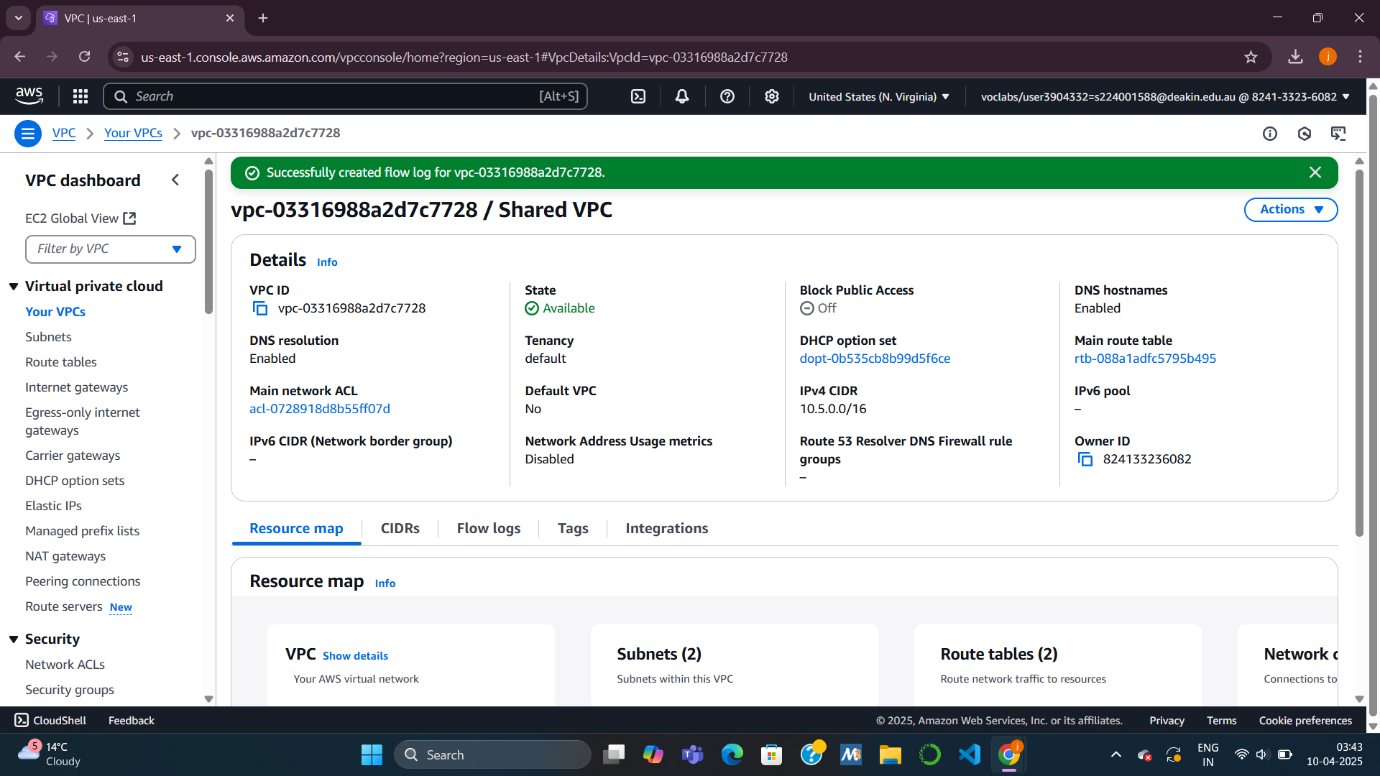
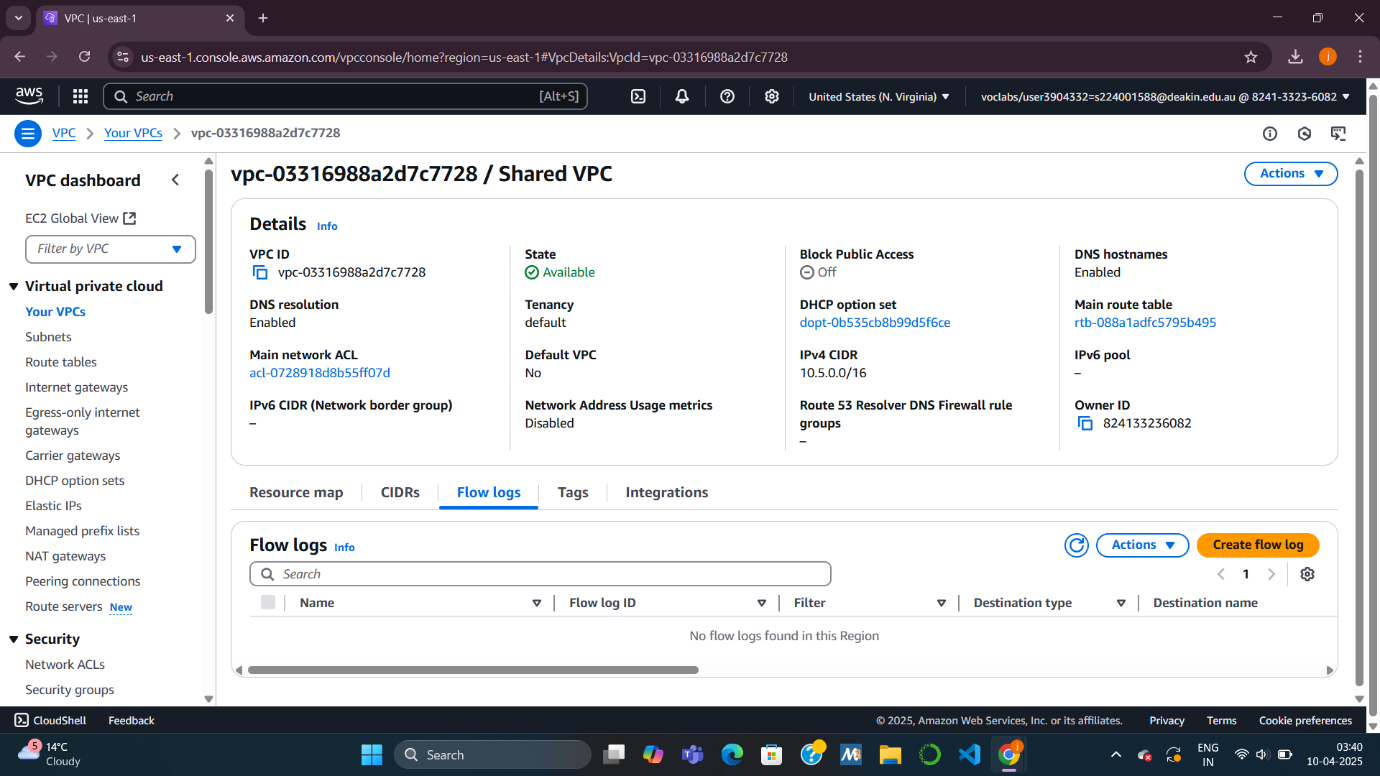
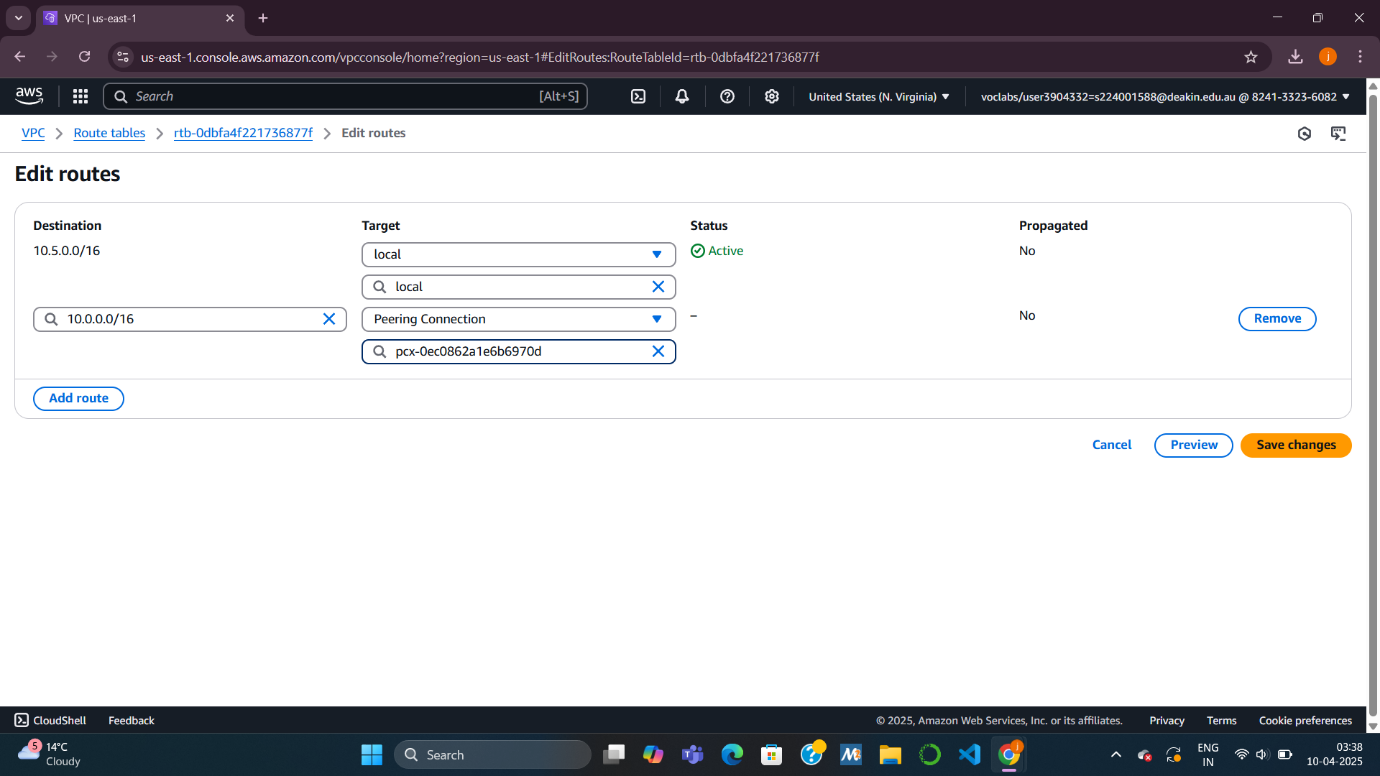
* Go to **Route Tables**.
* Select **Lab Public Route Table**.
* Add a route:
  + **Destination**: 10.5.0.0/16 (CIDR of Shared VPC)
  + **Target**: Peering connection Lab-Peer
* Save changes.

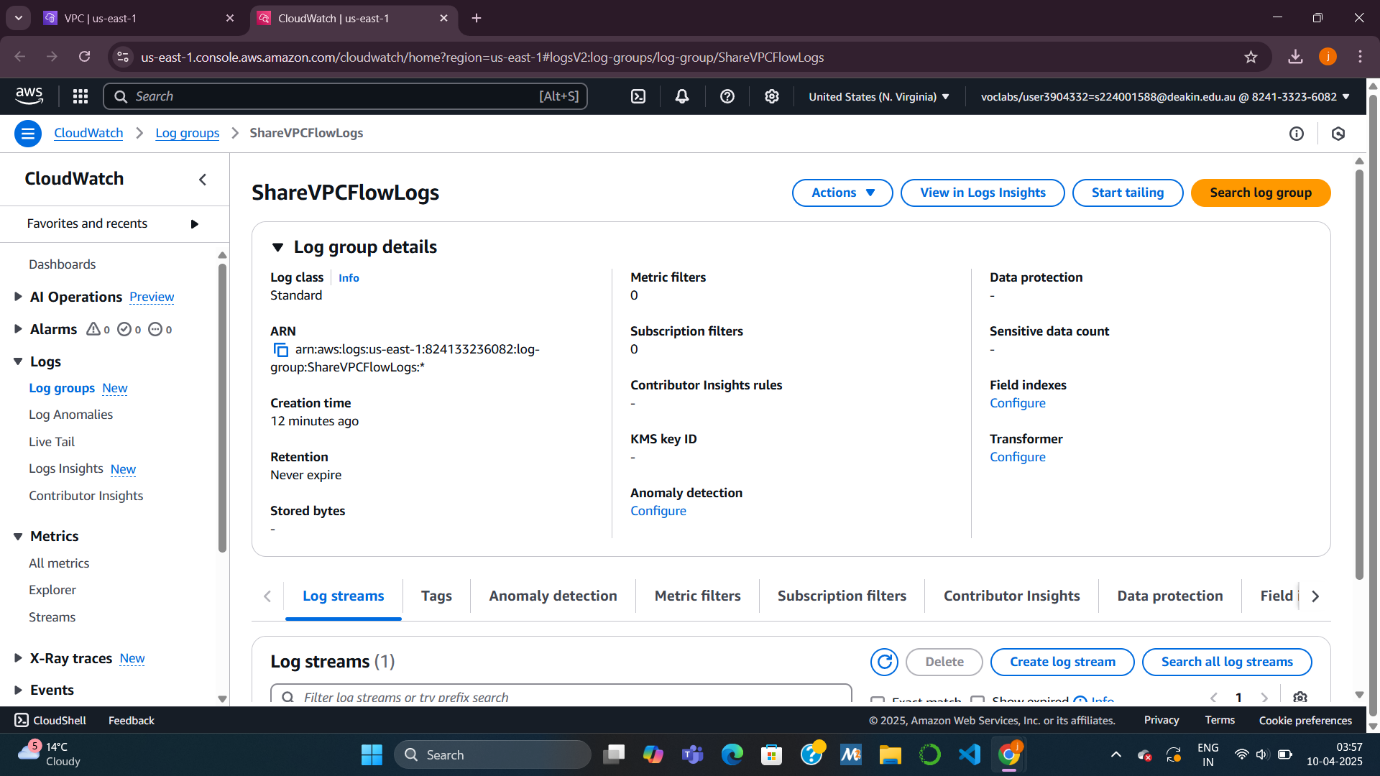
**For Shared VPC:**

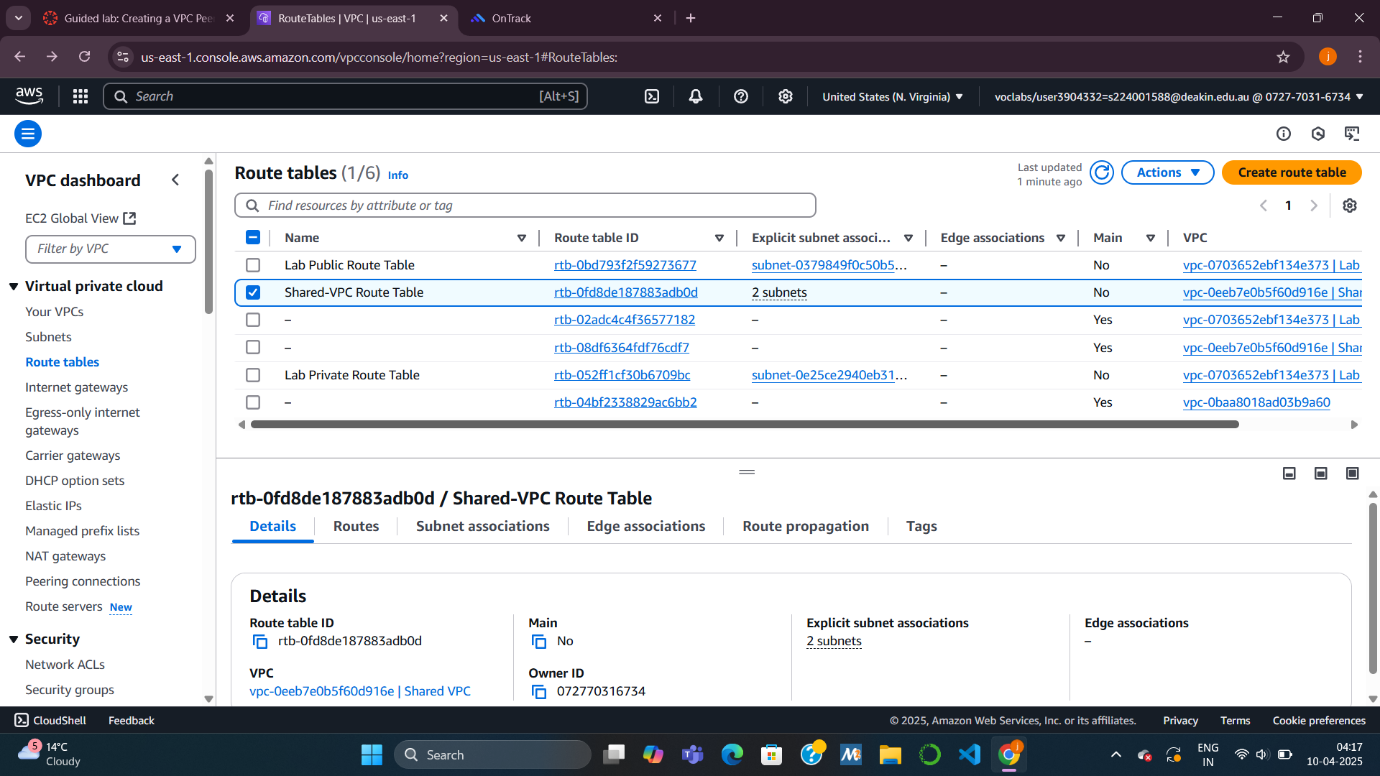
* Select **Shared-VPC Route Table**.
* Add a route:
  + **Destination**: 10.0.0.0/16 (CIDR of Lab VPC)
  + **Target**: Peering connection Lab-Peer
* Save changes.









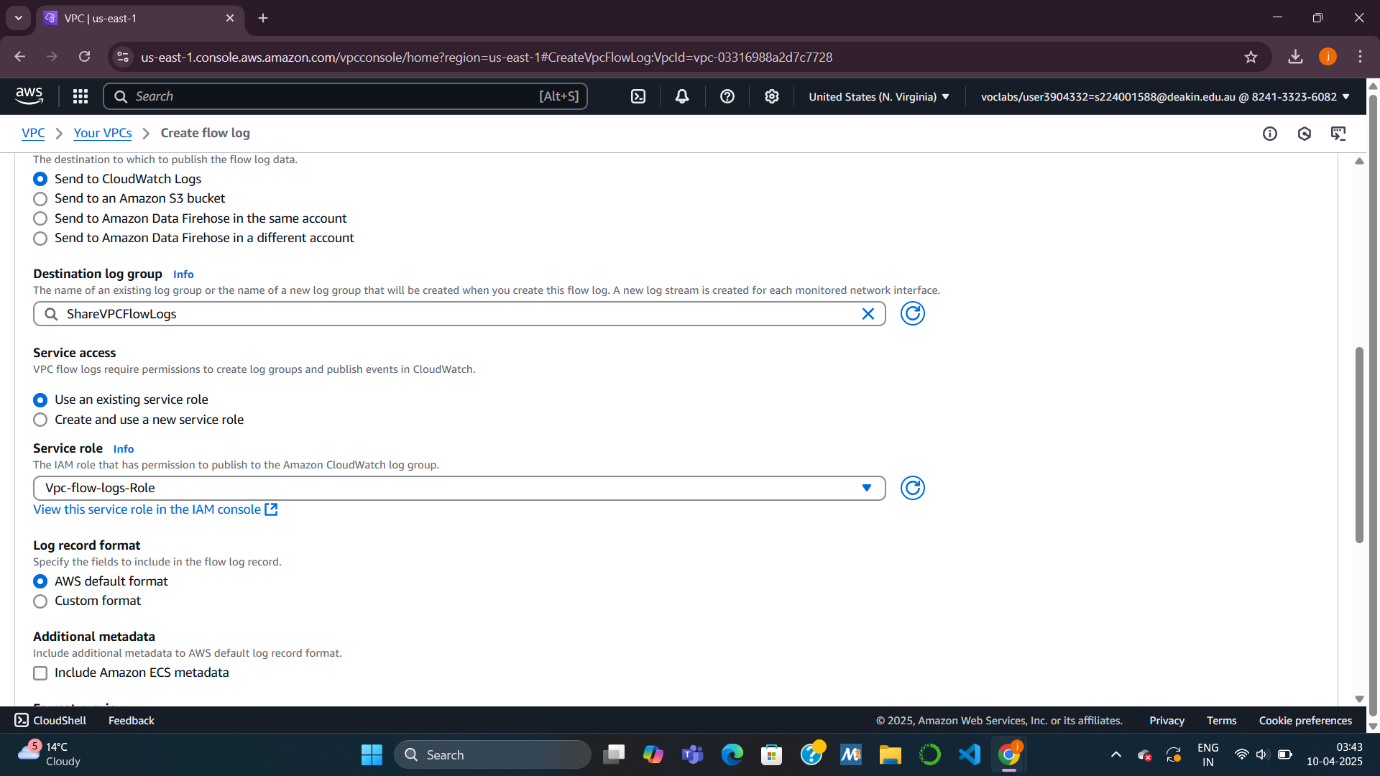


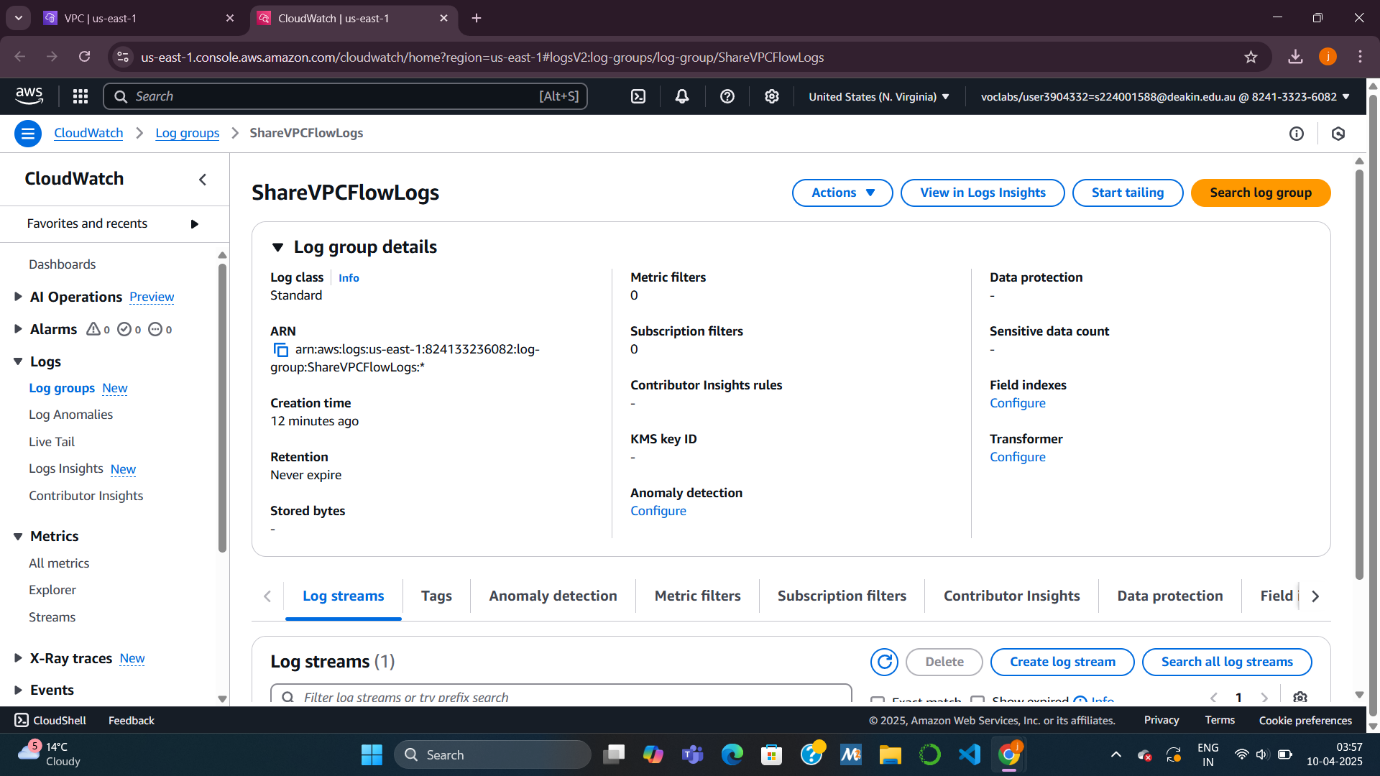
**Task 3: Enabling VPC Flow Logs**

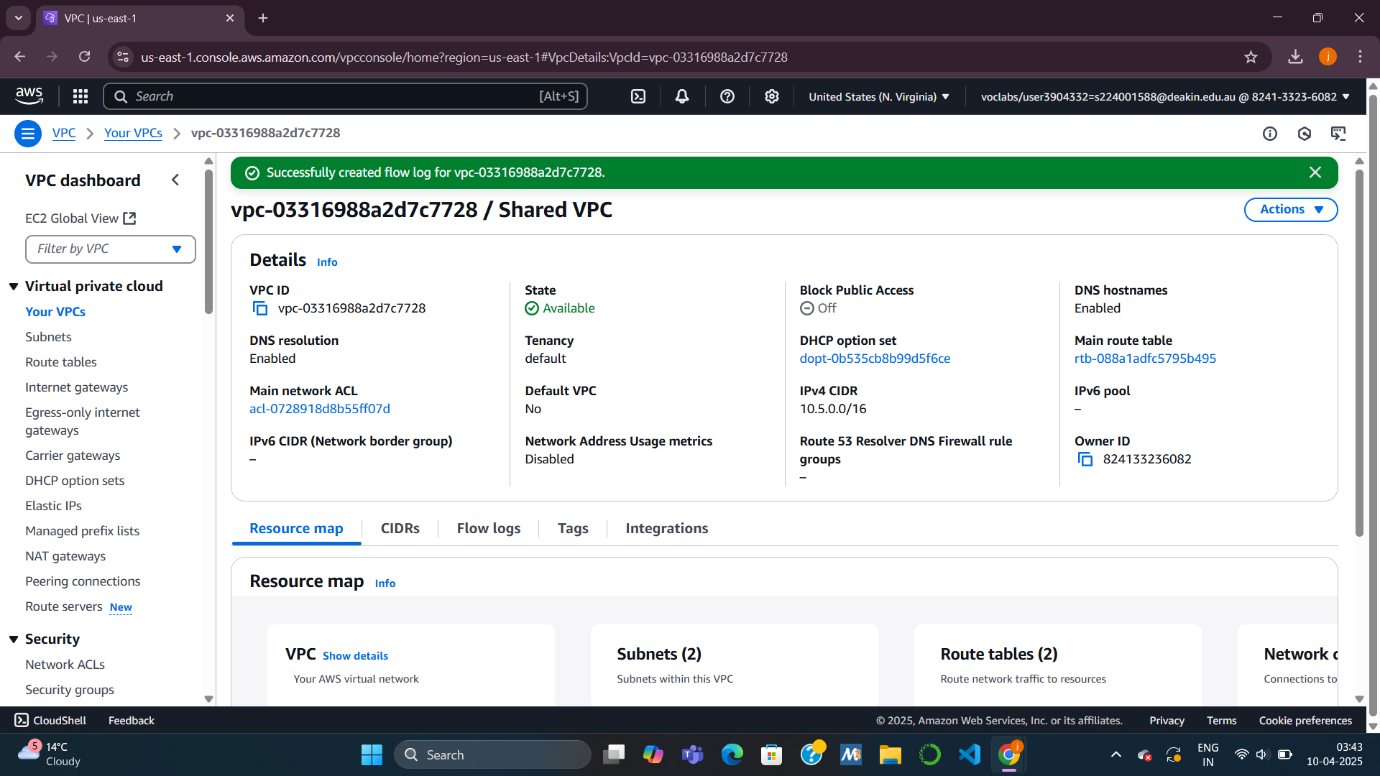
**What you’re doing:**

Setting up **logs** to monitor network traffic **in Shared VPC**.

**Steps:**

* Go to **Your VPCs**, select **Shared VPC**.
* Open **Flow logs** tab, click **Create flow log**.
* Set:
  + **Name**: SharedVPCLogs
  + **Aggregation interval**: 1 minute
  + **Destination**: CloudWatch Logs
  + **Log group**: ShareVPCFlowLogs
  + **IAM Role**: vpc-flow-logs-Role
* Create log.
* Click the **log group name** to view in CloudWatch.
* 





**Task 4: Testing the VPC Peering Connection**

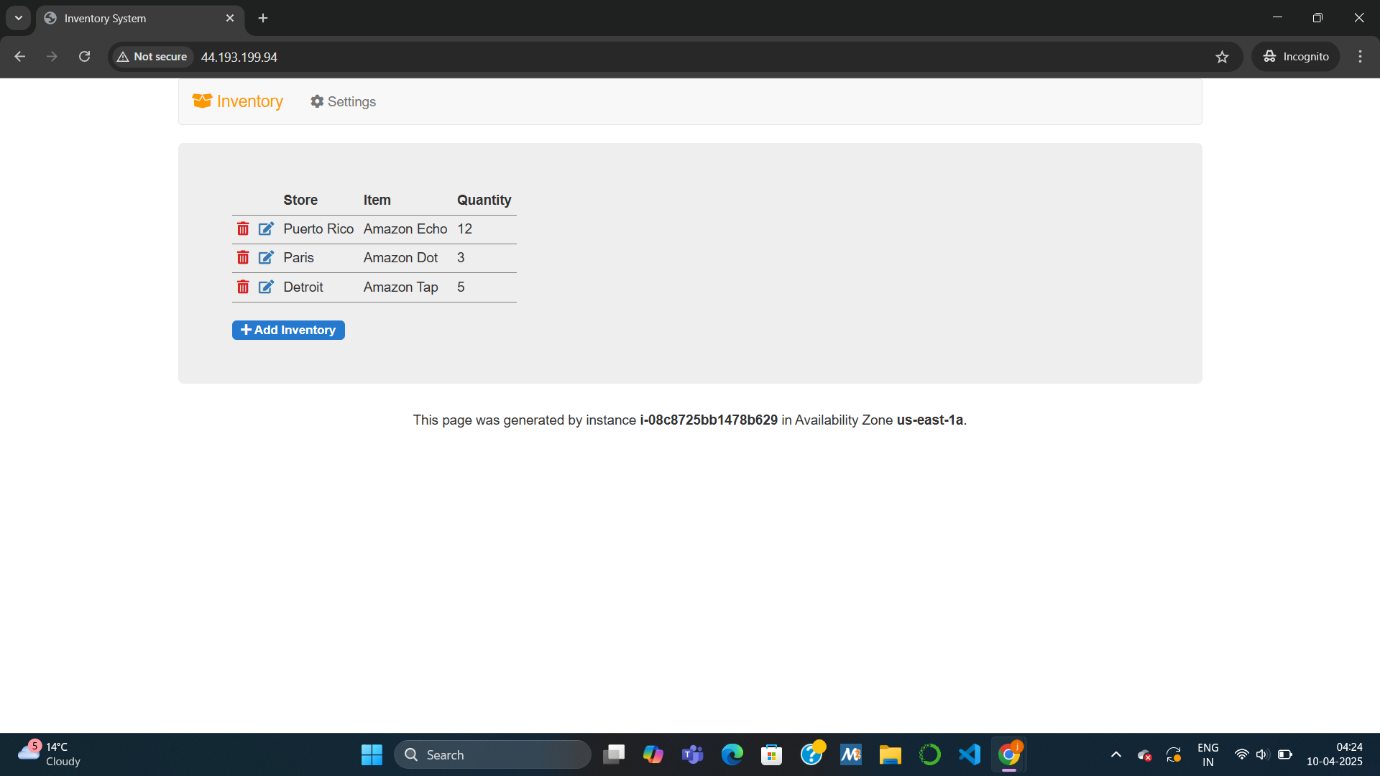
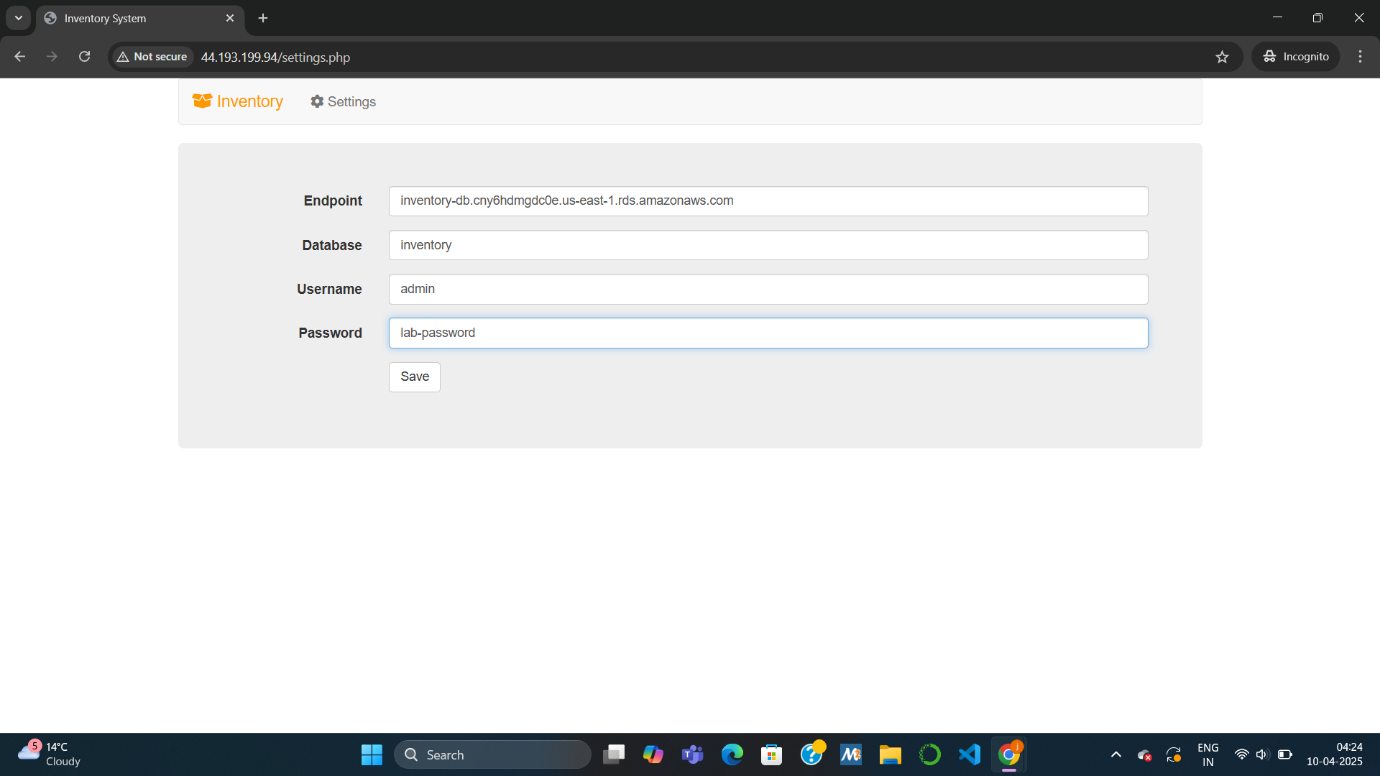
**What you’re doing:**

Ensuring the **Lab VPC EC2 instance** can connect to the **Shared VPC database**.

**Steps:**

* Copy the **public IP of the EC2** from AWS Details, open in browser.
* You'll see the **inventory app** asking to configure database.
* Click **Settings**:
  + Enter **database endpoint**, DB name, username, password.
* Save and confirm it connects to show **inventory data**.

This proves **peering is working**, since the database is not internet-accessible—only reachable through peering.



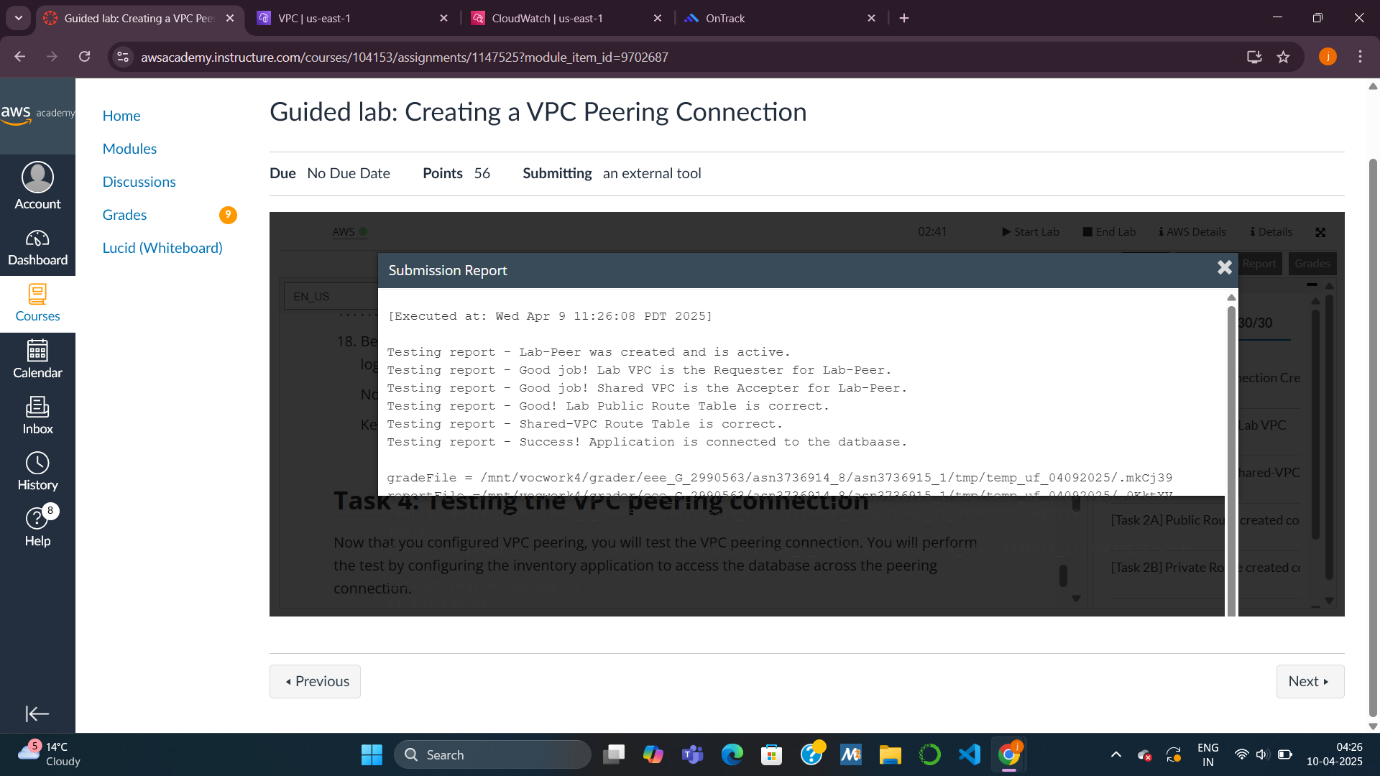
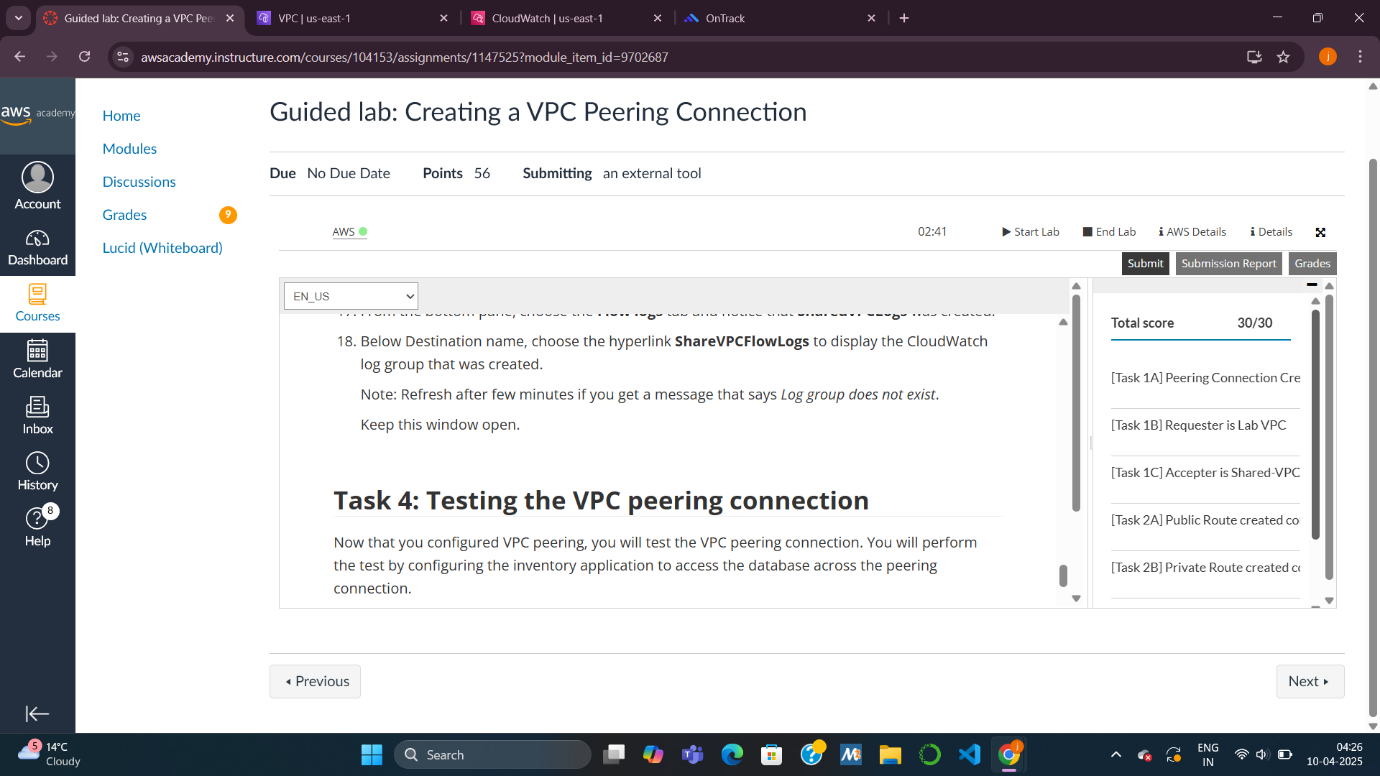
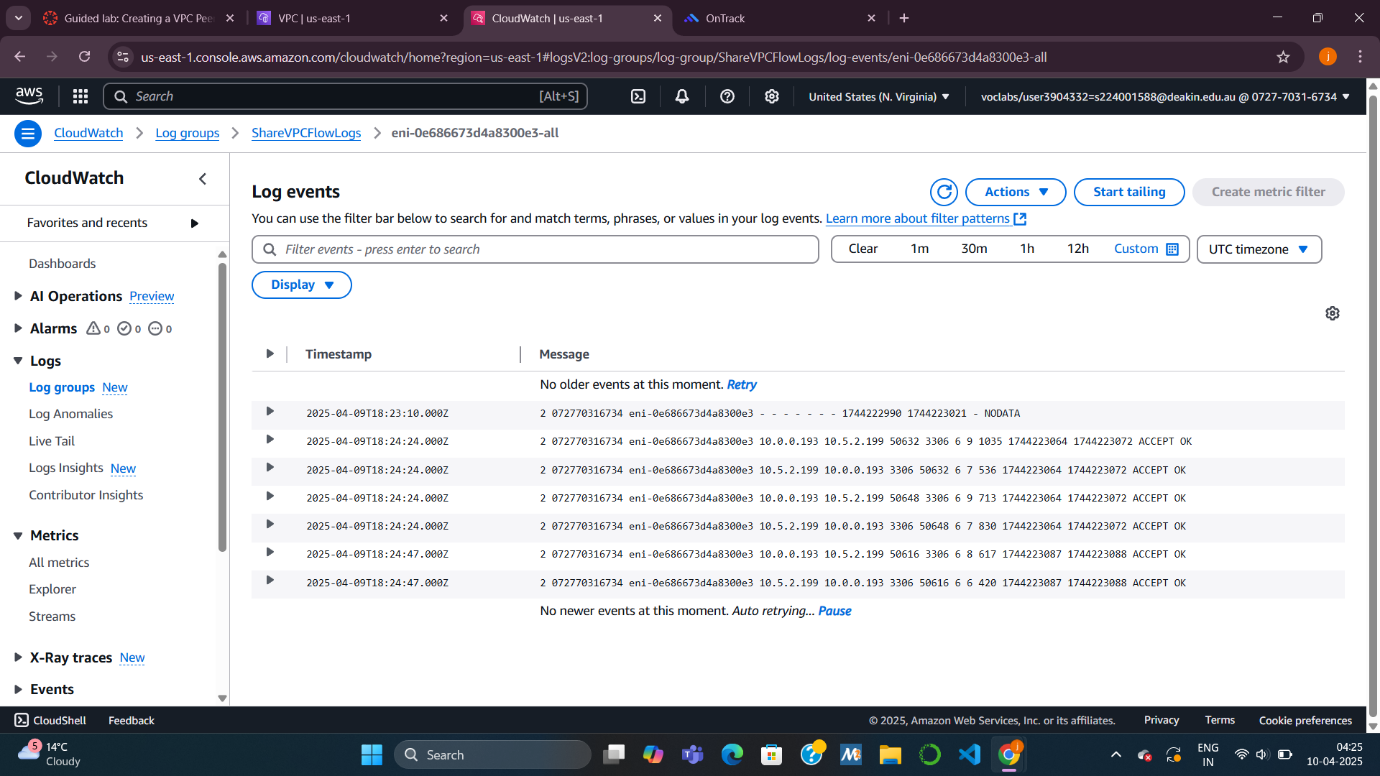
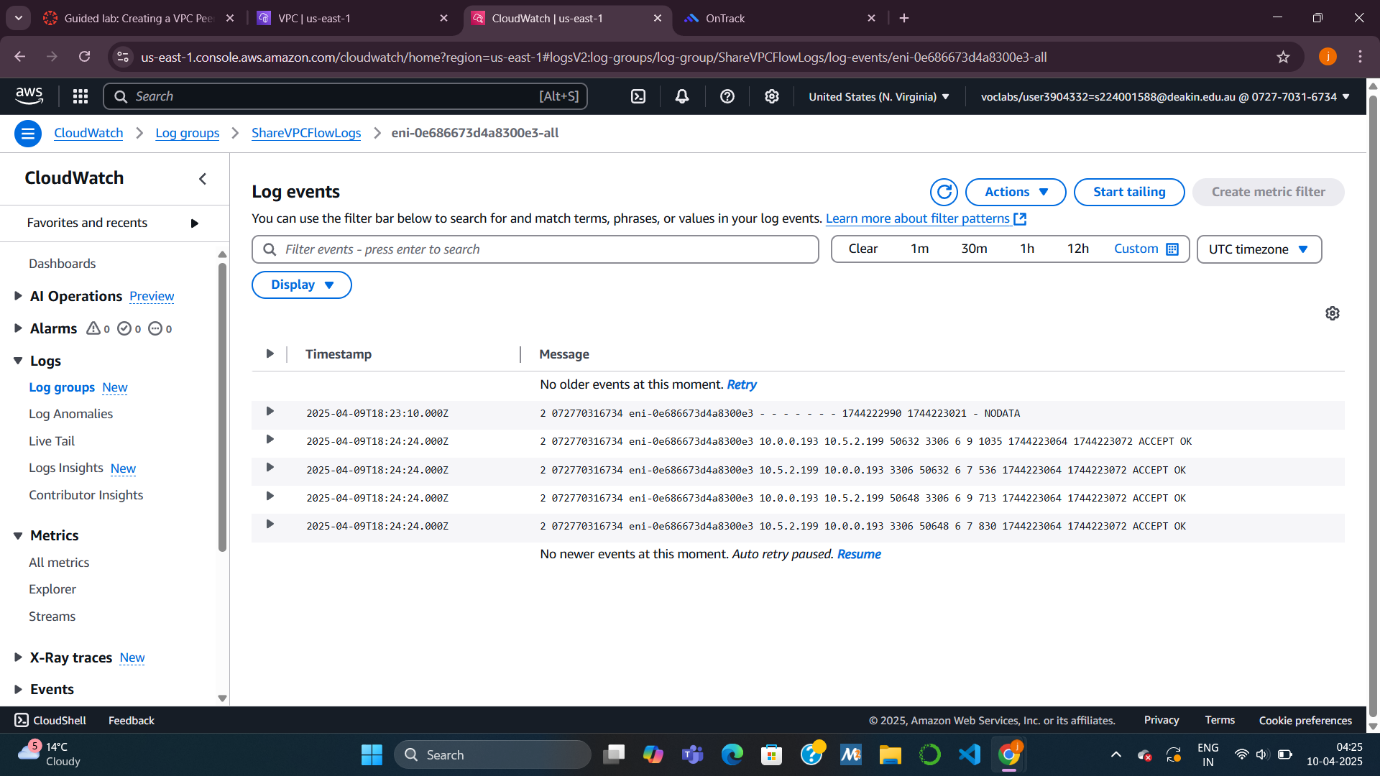
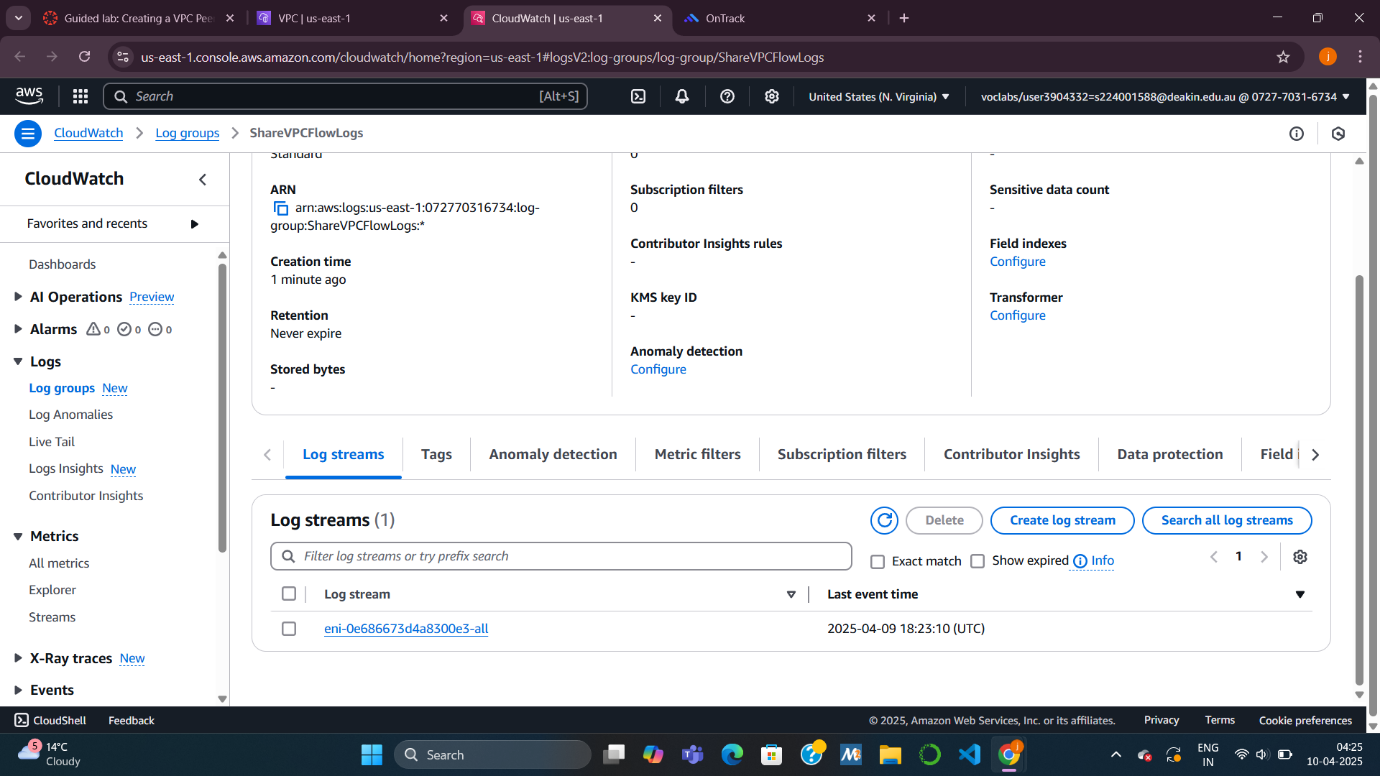
**Task 5: Analyzing the VPC Flow Logs**

**What you’re doing:**

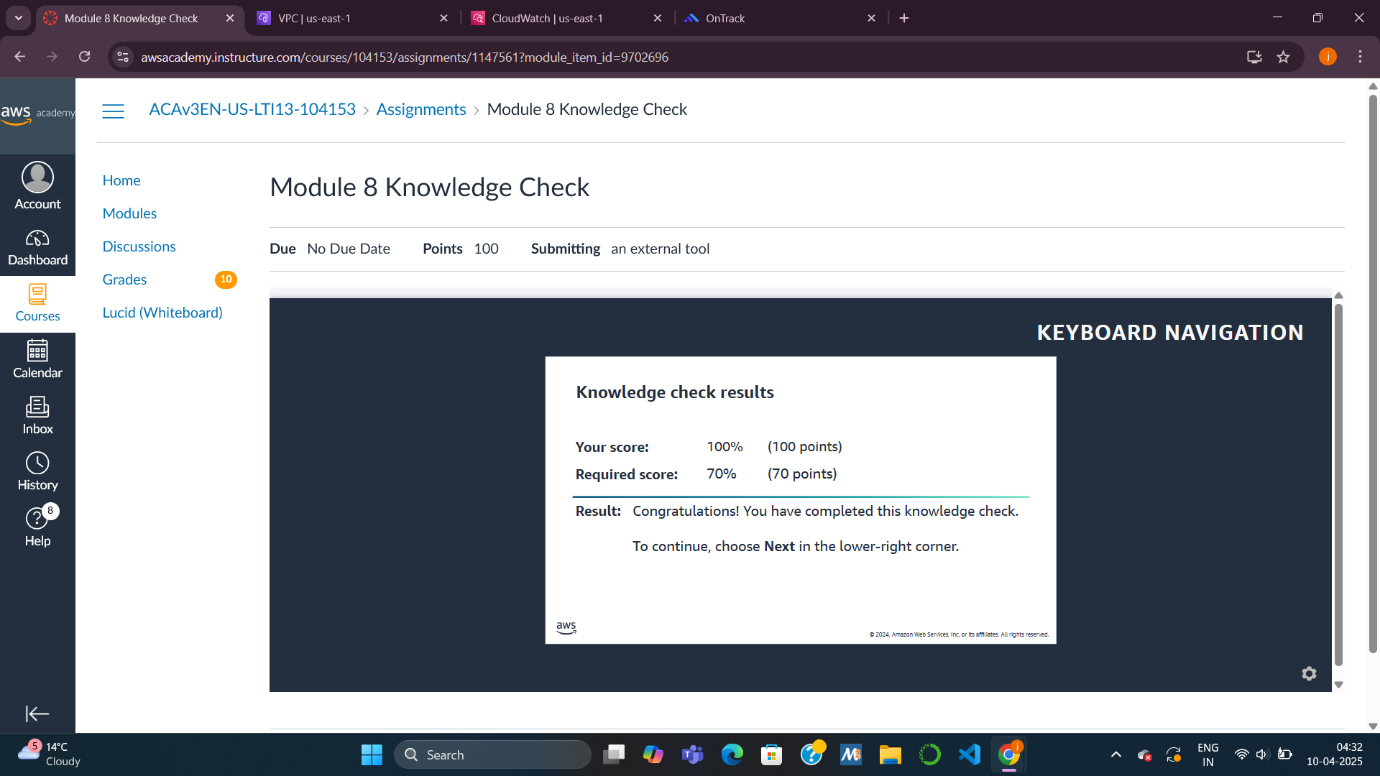
Viewing the logs to **inspect network traffic** between the app and database.

**Steps:**

* Go back to the **CloudWatch log group**.
* Open a **Log stream** (eni-...).
* Look for entries showing:
  + Traffic from EC2 to DB (ports like **3306**).
  + Source and destination **IP addresses**.
  + Status like **ACCEPT**/**OK**.



MODULE 8 KNOWLEDGE CHECK:



QUESTIONS:

1. Can you peer 2 VPCs that have the same CIDR ranges? Why?

No, we cannot peer two VPCs with overlapping CIDR blocks because routing will become ambiguous—the network won’t know which destination is which, leading to potential IP conflicts and incorrect routing.

2. What route changes do you need to make after creating a peering connection?

we need to update the route tables in both VPCs to:

* Add a route to the peer VPC’s CIDR block.
* Set the target as the peering connection.

This allows traffic to flow between the VPCs through the peering link.

3. If you have 3 VPCs: VPCA, VPCB, and VPCC; peering is configured between VPCA–VPCB and VPCB–VPCC, can you reach VPCC from VPCA? Why?

No, we cannot reach VPCC from VPCA because VPC peering is non-transitive. Even though both VPCA and VPCC are connected to VPCB, they cannot route traffic through it. we’d need to create a direct peering between VPCA and VPCC.