**Security Group & NACL (Network Access Control List)**

**Security Group:**

A **Security Group** in AWS (Amazon Web Services) acts as a **virtual firewall** (security Guard) for your EC2 (Elastic Compute Cloud) instances to control **inbound and outbound traffic**.

OR

A **Security Group** is a set of **rules** that define **what kind of traffic is allowed** to reach or leave an EC2 instance.

**Features:**

* Security Groups operate at the **instance level**, not the subnet level (which is handled by **Network ACLs**).
* It controls the inbound and outbound traffic to the specific instance.
* It is a **state-full** firewall, that meansif you allow an inbound request from a certain IP and port, the **response** is automatically allowed back, even if outbound rules don't explicitly permit it.

**Uses:**

| **Purpose** | **Description** |
| --- | --- |
| **Access Control** | You define **which IP addresses** or **CIDR ranges** can access your instance (e.g., allow only port 22 for SSH from your office IP). |
| **Application Filtering** | You can allow specific ports and protocols (e.g., HTTP on port 80, HTTPS on port 443). |
| **Improve Security** | By restricting traffic to only necessary sources and services, you reduce exposure to potential attacks. |
| **Ease of Management** | You can assign the same security group to multiple instances, making it easy to apply consistent rules. |

**Note:** A single SG can be attached to the multiple EC2 instance within a same VPC

**NACL (Network Access Control List):**

A Network Access Control List (NACL) in AWS is a **stateless, subnet-level firewall** that controls **inbound and outbound traffic** to and from one or more subnets within a **Virtual Private Cloud (VPC)**.

It consists of a numbered set of **allow and deny rules**, which are evaluated in order to determine whether traffic is permitted or denied.

Features:

* It works at the **subnet level**, not the instance level.
* It is **stateless**, return traffic must be **explicitly allowed** by corresponding outbound/inbound rules.
* NACLs have **separate rules** for inbound and outbound traffic.
* Rules are evaluated in ascending order of rule numbers, and the first matching rule is applied.

**Note:** In real word applications we won’t use the NACL

**Difference between SG & NACL:**

| **Feature** | **Security Group (SG)** | **Network ACL (NACL)** |
| --- | --- | --- |
| **Level** | Instance level | Subnet level |
| **Statefulness** | **Stateful** (return traffic auto allowed) | **Stateless** (return traffic must be allowed separately) |
| **Default behavior** | Denies all inbound, allows all outbound | Allows all inbound and outbound by default |
| **Rule Evaluation** | All rules are evaluated | Rules evaluated in order by rule number |
| **Allow/Deny Rules** | Only **allow** rules | Can have both **allow** and **deny** rules |
| **Use case** | Protecting EC2 instances | Adding an extra layer of subnet-level control |
| **Rule Type** | Based on protocols and ports | Also based on protocols, ports, and IPs |
| **Logging support** | No logging | Can enable **VPC flow logs** for traffic analysis |

Block Diagram:

Inbound Traffic

Security Group

User

Outbound Traffic

IGW

VPC

NACL

Subnet

NIC

Internet

RT

Fig: Block Diagram of SG & NACL working scope within a VPC.

**NACL (Network access control list):**

Let’s work with the NACL practically

Process:

* Create a VPC, subnet, RT, IGW.
* Create an EC2 instance within a subnet of a VPC.
* Write Inbound and Outbound rules for the NACL.
* Associate the NACL to subnet.

**Note:** SG & NACL is by default created, while creating the VPC.

Note: In order to know the functionality of NACL, allow all the traffic in SG.

**Step1:** Create an EC2 instance (Linux OS).

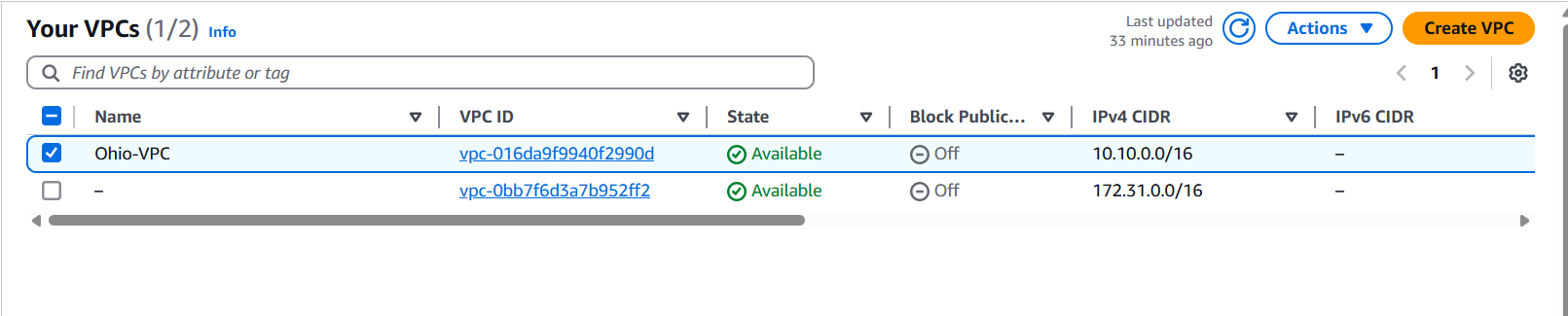


Fig: VPC is created.

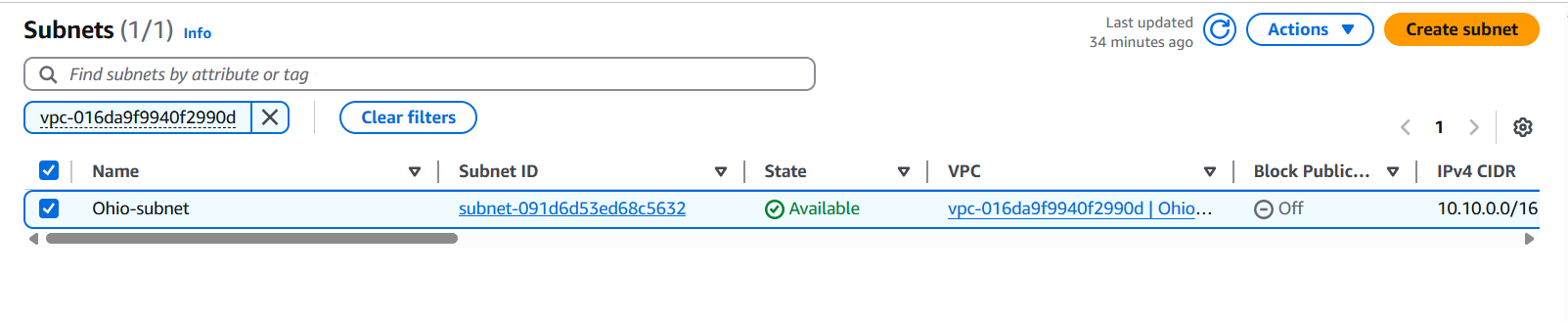


Fig: subnet.

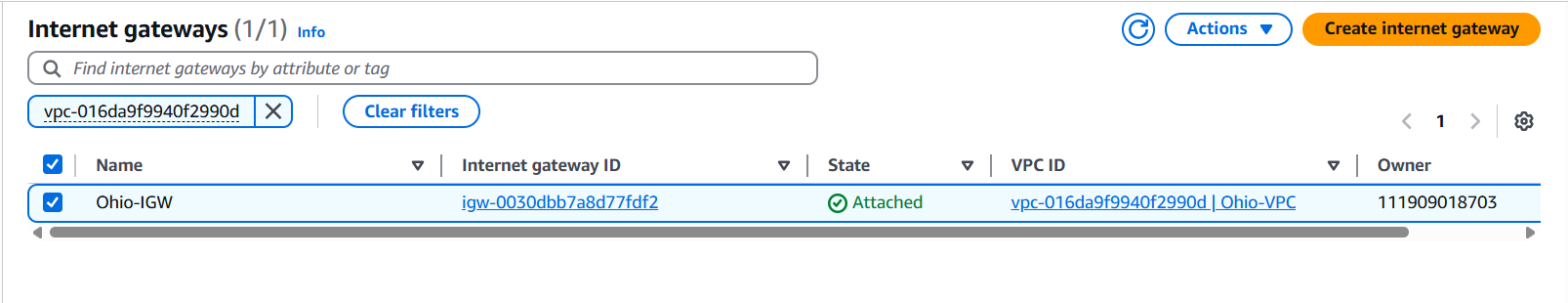


Fig: Internet gateway.

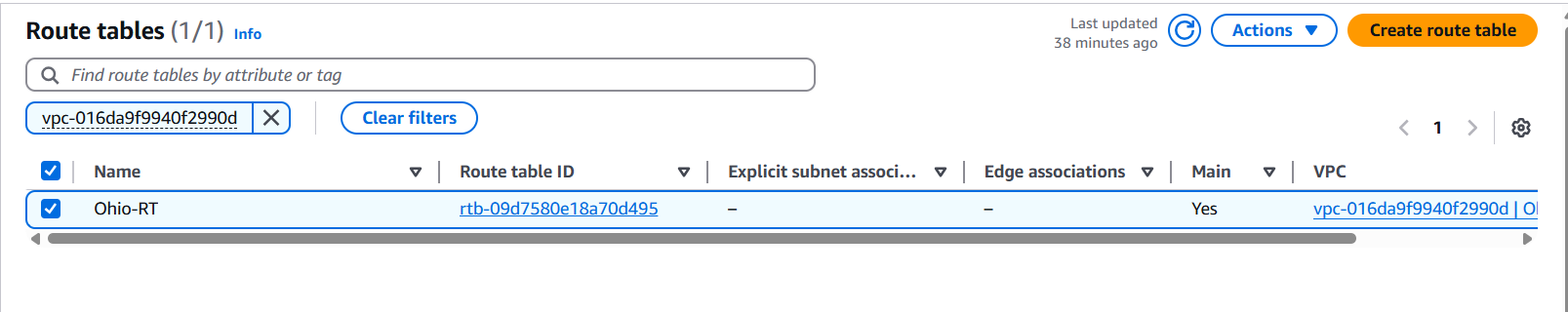


Fig: Route table.

Now Configure the SG to allow all the traffic in inbound and outbound rules.

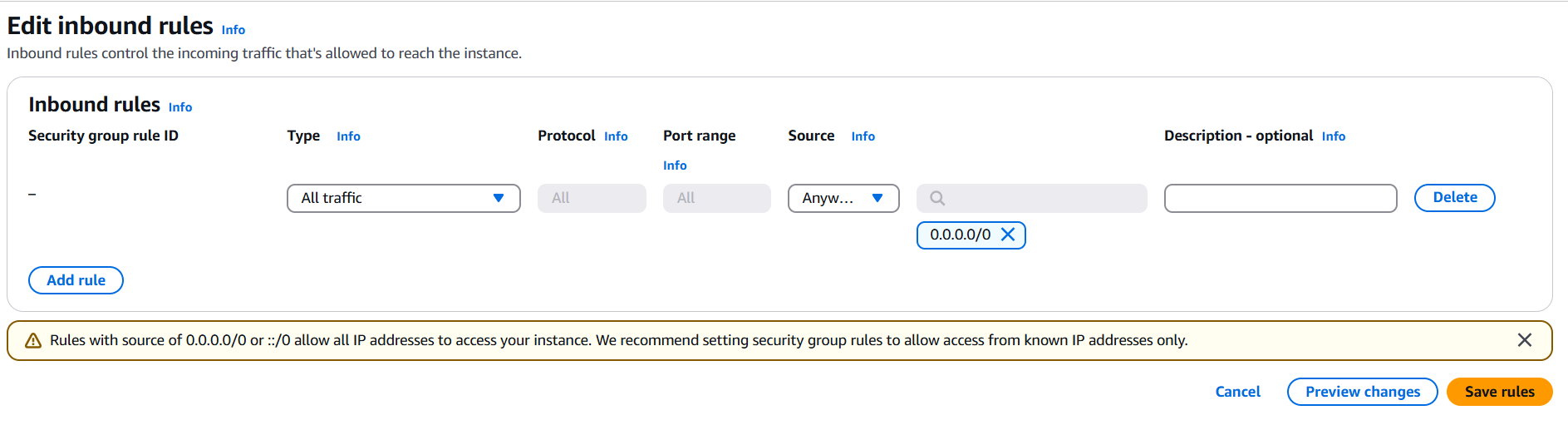


Fig: Allow all traffic using SG.

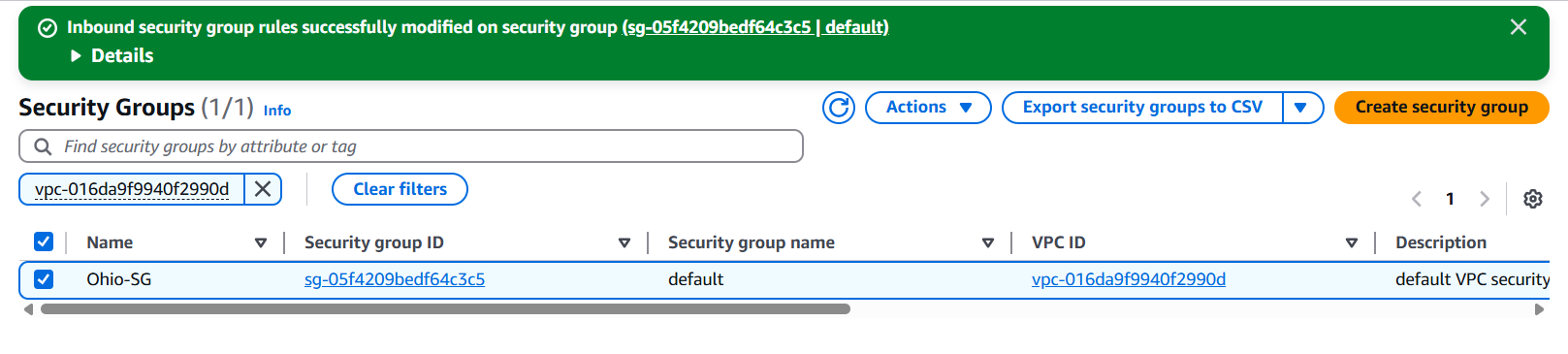


Fig: Security group.

Now create the EC2 instance using above VPC, subnet, RT, IGW, and SG.

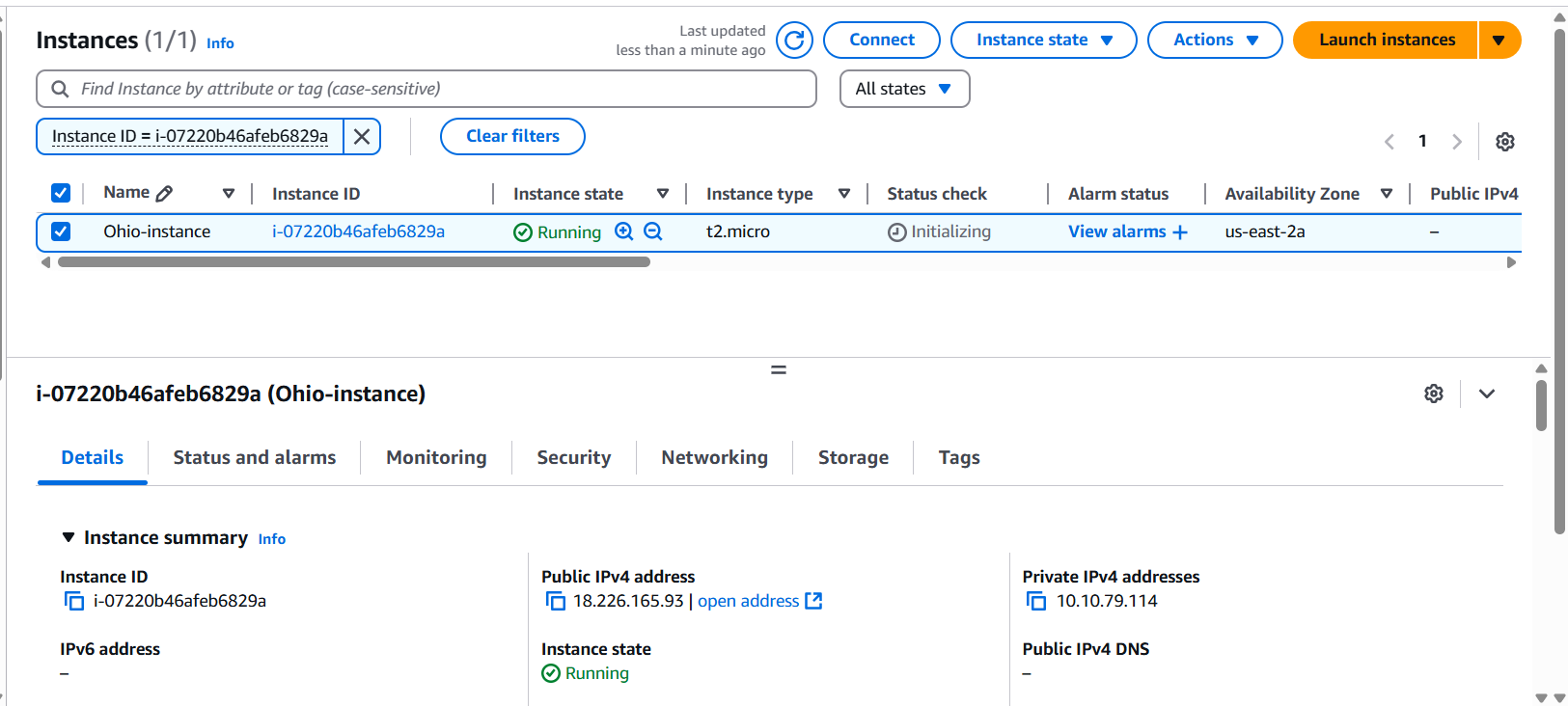
****

Fig: EC2 instance is created.

Step2: Configure the NACL.

**Note:** By default the NACL is created while creating the VPC.

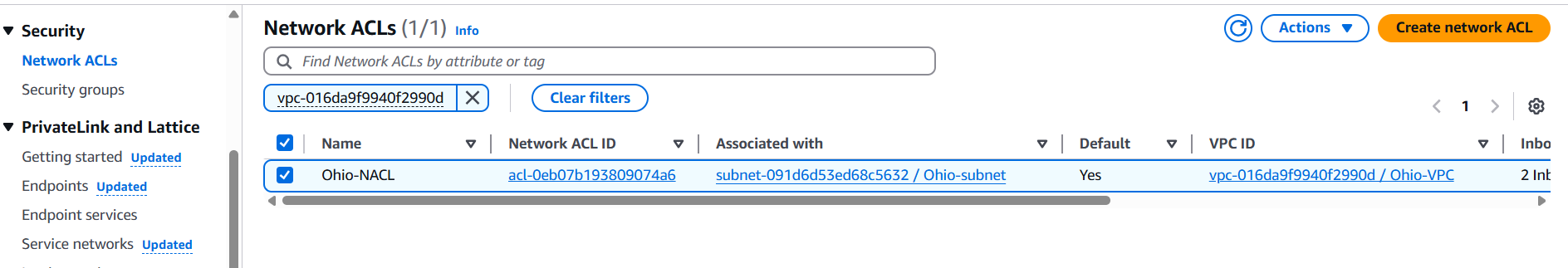


Fig: NACL.

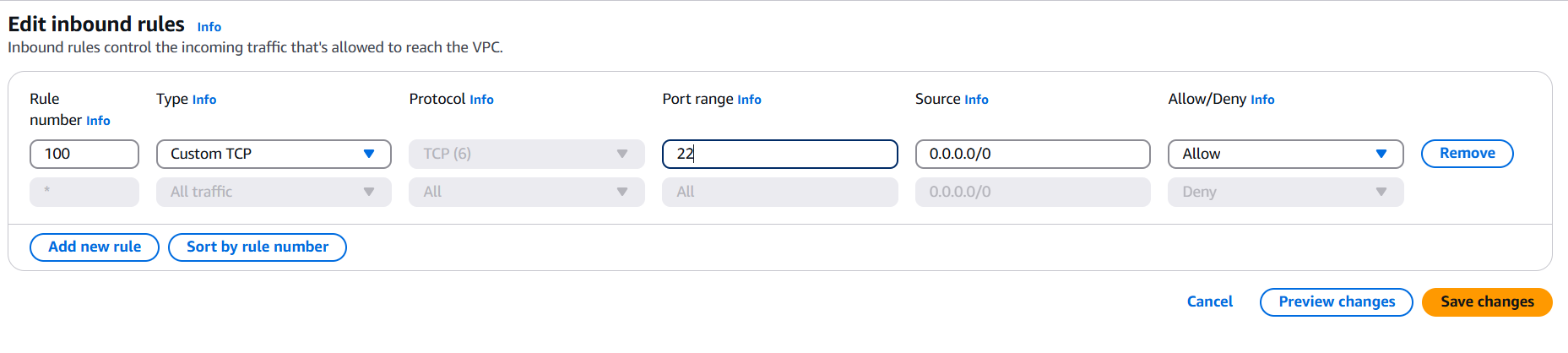


Fig: Inbound rules of NACL to allow SSH port.

Here we have to configure both the inbound and outbound rules then only we can connect to the instances using SSH (22) port.

Note while configuring the outbound rules we have to know at which port of the source should be connected.

Knowing the sources port number is somewhat difficult so instead of it we use ephemeral ports range.

**Ephemeral Ports:**

**Ephemeral ports** are short-lived, temporary ports assigned by a device's operating system when establishing an **outbound** connection. They are typically used by client applications to communicate with a server.

* Ephemeral ports are temporary communication ports assigned by an operating system to a client application when it initiates a connection to a server.
* These ports are used for the duration of the communication session.
* Once the session ends, the ephemeral port is released and can be reused for other connections.
* **Port Range**: Usually from **49152 to 65535**, though the exact range can vary by OS.
* **Assigned Automatically**: The OS picks an available ephemeral port when a client (e.g., your browser) starts a connection to a server (e.g., a website on port 443 for HTTPS).
* **Used Temporarily**: Once the session ends, the port is released for reuse.

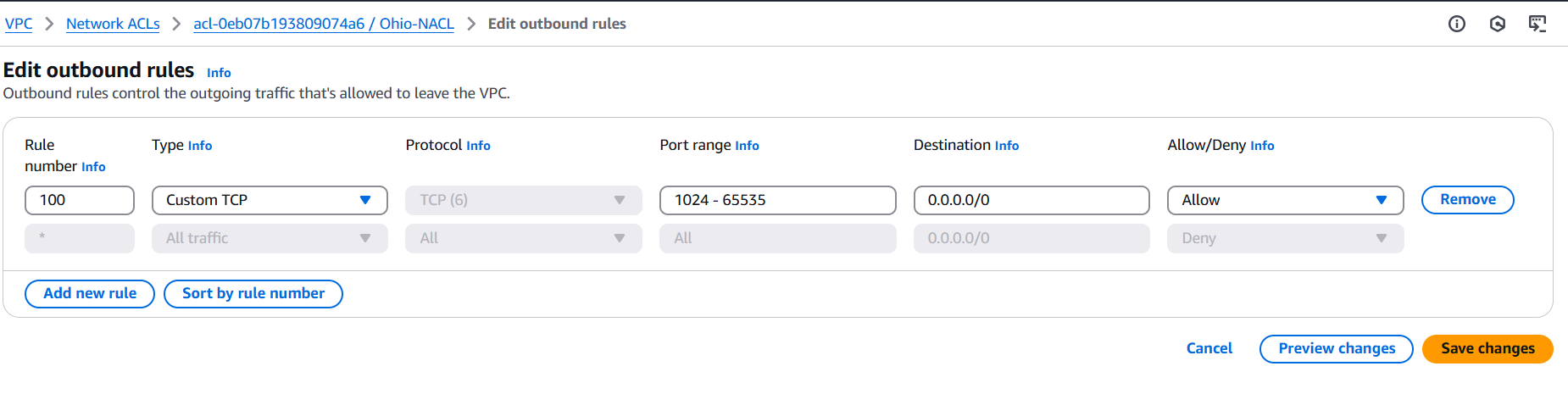
****

Fig: Outbound rules of NACL.

Ephemeral port range

Step3: Now connect to the EC2 instance (Linux OS) using putty SSH port.

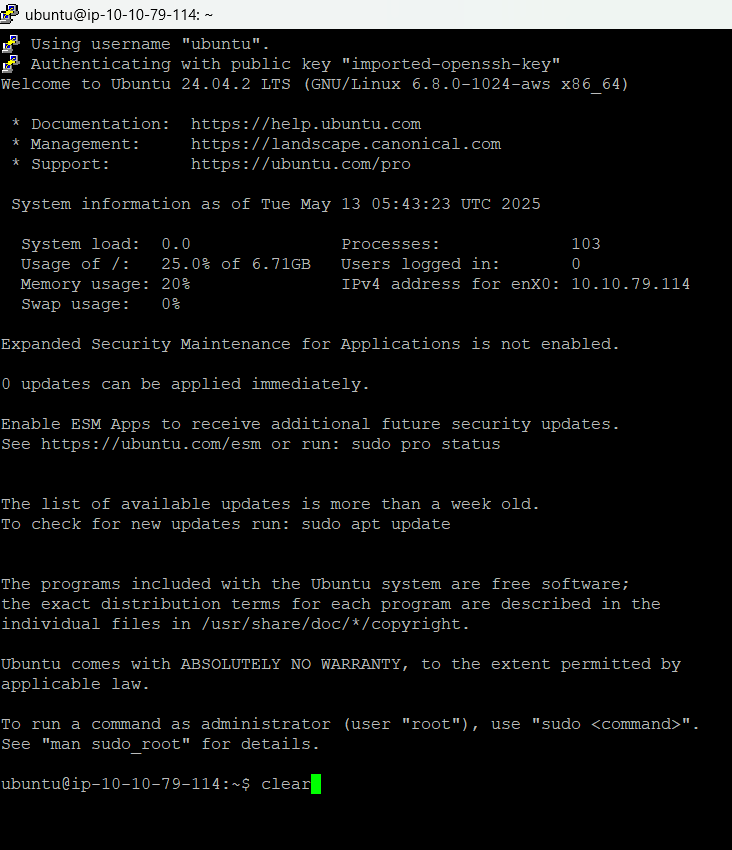


Fig: Able to connect the instance.

Let’s try by change the outbound rules to allow SSH (22) instead of Ephemeral ports.

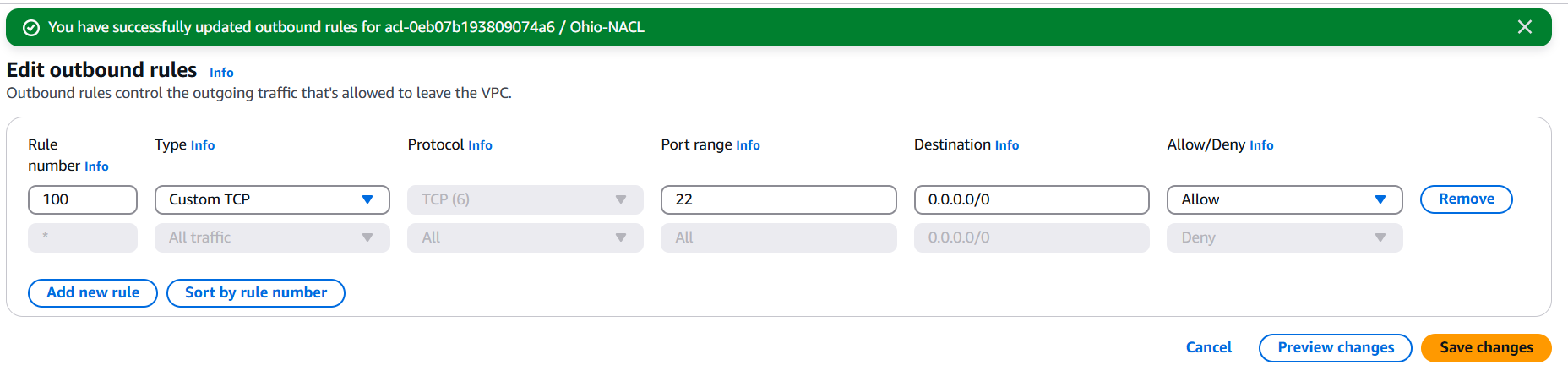


Fig: NACL outbound rule to allow SSH port.

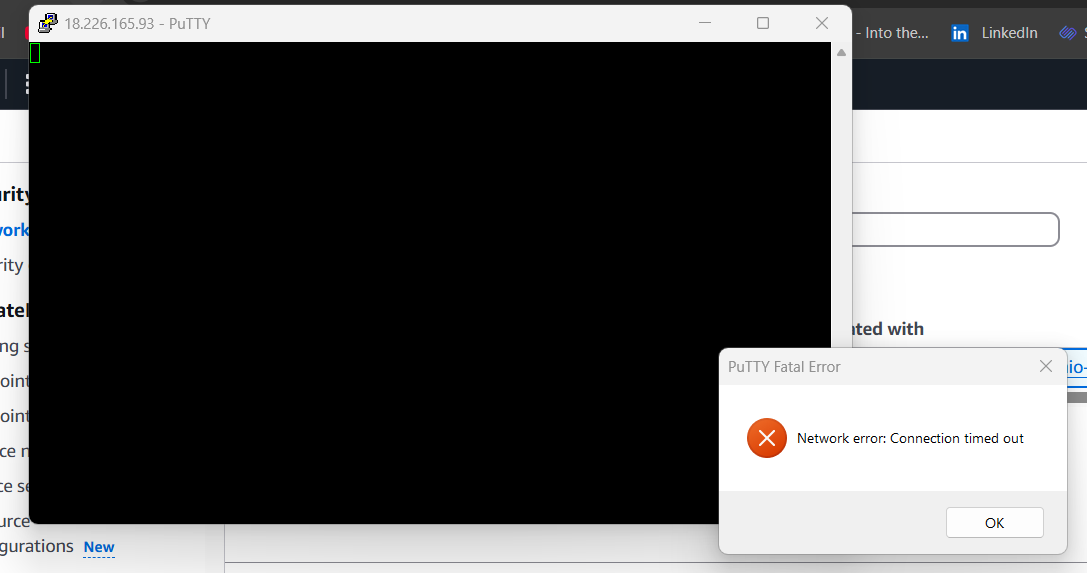


Fig: Connection time out.

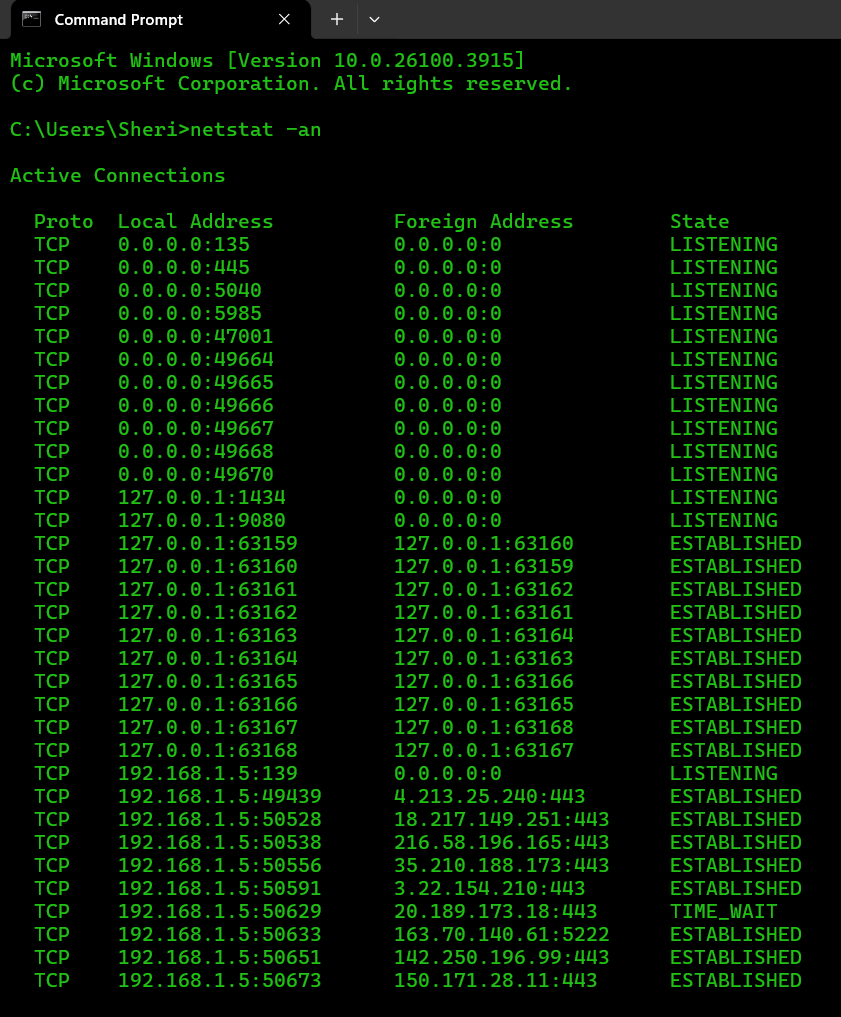
When we enabled the SSH port at outbound rules, its shows the above popup, it is because the client/user may be using different OS (windows/mac) so it won’t accept the SSH traffic.

So in order to connect the client/user/source machine we use Ephemeral ports. From the Ephemeral port range it will pic a specific port and connect to the client machine.

We can check at which port number the server/instance is connect to our machine by using a command

**Command:** netstat -an

The **netstat -an** command is used to **display active network connections, listening ports, and associated IP addresses and ports**, in a **numerical format** (i.e., IP addresses and port numbers instead of domain names or service names).



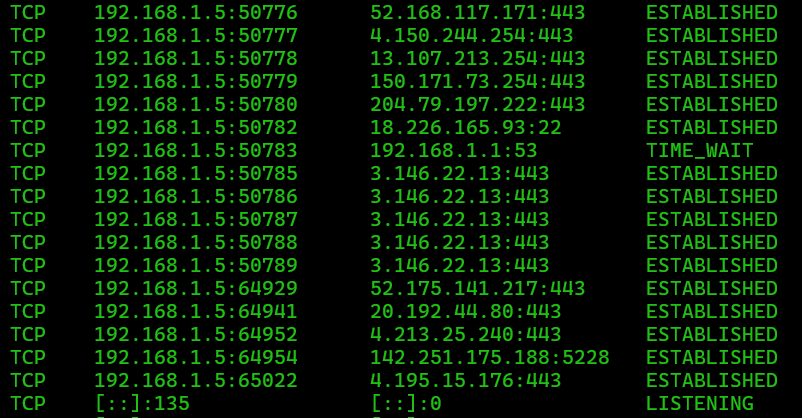


Fig: Connected to our windows machine.