**step 6 : security group**

Types of security groups:

1.EC2-Classic: a)you can only create inbound rules

b) when you add a rule to EC2-Classic security groups, you no longer have to specify a protocol.

2.EC2-VPC a)you can create inbound rules and outbound rules.

b)when you add a rule to EC2-VPC security groups,You need to specify a protocol.

**\*\*\*\*Definition of AWS Security Groups\*\*\*\***

1)AWS Security Groups are a flexible tool to help you secure your Amazon EC2 instances.and helps to secure cloud environment.

2)AWS Security Groups act like a firewall for Amazon EC2 instances controlling both inbound and outbound traffic.

3)When we launch an instance on Amazon EC2, we need to assign it to a particular security group.

4)After that,we can set up ports and protocols, which remain open for users and computers over the internet.

5)AWS Security Groups are very flexible. that we can use the default security group and still customize it according to our liking

Or you can create a security group that you want for your specific applications.

To do this, we have to write the corresponding code or use the Amazon EC2 console to make the process easier.

**\*\*\*\*AWS Security Groups Differ From Traditional Firewalls\*\*\*\***

1)Traditional firewalls can be rigid and limiting. AWS Security Groups, on the other hand, allow us to specify permissive rules.we cannot deny traffic.

2)This is very important to remember because at its most basic, and without setting rules, all traffic is blocked.

3)If a data packet has no particular rule that permits it to go through, it will be dropped instantly.

**\*\*\*\*How AWS Security Groups Work\*\*\*\***

we can customize AWS Security Groups according to our needs. we need to give each group a unique name that will allow us to select it from a menu.

It’s better if you give a group a descriptive name so you can choose the best one for your needs without having to look into the ruleset for that particular group.

Descriptive names have a 255-character limit and can be alphanumeric only; spaces with some special characters are allowed.

Also, you cannot use a name that starts with sg-. You need to use unique names within the same VPC.

You also need to create a security group that is in the same VPC as the resources you want to protect.

Remember, however, you can only create a limited number of security groups on every VPC that you have.

There is also a limit on the number of rules you can add to one security group. Furthermore,

there is a limited number of security groups that you can use with a network interface.

As described above, there are no deny rules in AWS Security Groups, only allow rules.

But you can specify different rules for outbound and inbound traffic.

Also, remember that AWS Security Groups are stateful. This means that when you send a request from your instance,

you will get a response to that request, disregarding the ruleset for your inbound security group. Meanwhile, replies to allow inbound traffic are going to be allowed to flow out, even if you do not have an outbound rule that explicitly allows it to go out.

Some people make the mistake of thinking that instances using the same security group are able to communicate with one another.

This is not necessarily true: You need to create a rule that allows this. The default security group, however, has these rules turned on by default.

You can choose to add inbound and outbound rules at the time of creation, or you can add them at any time after you have created the security group.

To add a security group rule, you must specify:

The protocol to allow

The range of ports to allow

The traffic source to allow for inbound rules, or the traffic destination to allow for outbound rules

An optional description

\*inbound rules:Inbound rules control the incoming traffic that's allowed to reach the instance.

-IIS webserver given by microsoft

-Tomcat supported by apache

\*outbound rules:Outbound rules control the outgoing traffic that's allowed to leave the instance.

-access internet

-how do he help to access outside resource outside machine