PCET's
Pimpri
Chinchwad
University, Pune

Learn | Grow | Achieve

| | |
|---|---|
| **Checked By** | Dr. Avinash Golande |
| **Name** | |
| **Sign** | |
| **Date** | |
| **Grade** | |

# EXPERIMENT NO . 13

- **Title:-** To Launch website using EC2 instance

- **Objective :-**
  - To understand how to launch and configure an EC2 instance on AWS.
  - To create and manage **security group rules** for web (HTTP) and SSH access.
  - To connect to an EC2 Linux instance using **PuTTY**.
  - To install and configure a **web server (Apache)** on the EC2 instance.
  - To deploy a **website from a GitHub repository** and verify its successful hosting using the public IP address.

**Resources used** :- AWS Account (Free Tier), AWS Management Console, PuTTY & PuTTYgen, Key Pair (.pem / .ppk), Security Group (HTTP & SSH), GitHub Repository Link, PC/Laptop with Internet connection.

- **Theory :-**

Prerquisite:AWS Free user account,website code reposittory from github

Theory:

Security group rules

- **Name**: The name for the security group (for example, "my-security-group").

A name can be up to 255 characters in length. Allowed characters are a-z, A-Z, 0-9, spaces, and ._-:/()#,@[]+=;{}!$*. When the name contains trailing spaces, we trim the spaces when we save the name. For example, if you enter "Test Security Group " for the name, we store it as "Test Security Group".

- **Protocol**: The protocol to allow. The most common protocols are 6 (TCP), 17 (UDP), and 1 (ICMP).
- **Port range**: For TCP, UDP, or a custom protocol, the range of ports to allow. You can specify a single port number (for example, 22), or range of port numbers (for example, 7000-8000).
- **ICMP type and code**: For ICMP, the ICMP type and code. For example, use type 8 for ICMP Echo Request or type 128 for ICMPv6 Echo Request.
- **Source or destination**: The source (inbound rules) or destination (outbound rules) for the traffic to allow. Specify one of the following:

- A single IPv4 address. You must use the /32 prefix length. For example, 203.0.113.1/32.
- A single IPv6 address. You must use the /128 prefix length. For example, 2001:db8:1234:1a00::123/128.
- A range of IPv4 addresses, in CIDR block notation. For example, 203.0.113.0/24.
- A range of IPv6 addresses, in CIDR block notation. For example, 2001:db8:1234:1a00::/64.
- The ID of a prefix list. For example, pl-1234abc1234abc123. For more information, see Prefix lists in the *Amazon VPC User Guide*.
- The ID of a security group (referred to here as the specified security group). For example, the current security group, a security group from the same VPC, or a security group for a peered VPC. This allows traffic based on the private IP addresses of the resources associated with the specified security group. This does not add rules from the specified security group to the current security group.

- **(Optional) Description**: You can add a description for the rule, which can help you identify it later. A description can be up to 255 characters in length. Allowed characters are a-z, A-Z, 0-9, spaces, and ._-:/()#,@[]+=;{}!$*.

When you create a security group rule, AWS assigns a unique ID to the rule. You can use the ID of a rule when you use the API or CLI to modify or delete the rule.

When you specify a security group as the source or destination for a rule, the rule affects all instances that are associated with the security group. Incoming traffic is allowed based on the private IP addresses of the instances that are associated with the source security group (and not the public IP or Elastic IP addresses). For more information about IP addresses, see Amazon EC2 instance IP addressing. If your security group rule references a deleted security group in the same VPC or in a peer VPC, or if it references a security group in a peer VPC for which the VPC peering connection has been deleted, the rule is marked as stale. For more information, see Working with Stale Security Group Rules in the *Amazon VPC Peering Guide*.

If there is more than one rule for a specific port, Amazon EC2 applies the most permissive rule. For example, if you have a rule that allows access to TCP port 22 (SSH) from IP address 203.0.113.1, and another rule that allows access to TCP port 22 from everyone, everyone has access to TCP port 22.

When you add, update, or remove rules, the changes are automatically applied to all instances associated with the security group.

**Inbound rules**

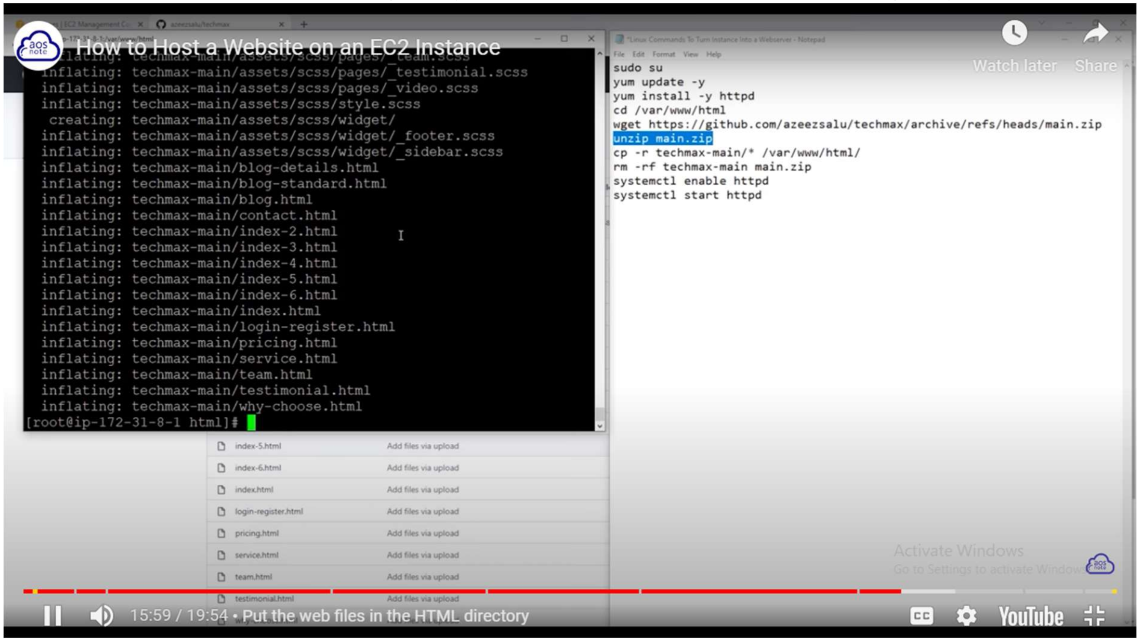
The inbound rule in your security group must allow traffic on all ports. It needs to do this because the destination port number of any inbound return packets is set to a randomly allocated port number.

**Outbound rules**

By default, a security group includes an outbound rule that allows all outbound traffic. We recommend that you remove this default rule and add outbound rules that allow specific outbound traffic only.

The security group attached to QuickSight network interface should have outbound rules that allow traffic to each of the database instances in your VPC that you want QuickSight to connect to. To restrict QuickSight to connect only to certain instances, specify the security group ID (recommended) or the private IP address of the instances to allow. You set this up, along with the appropriate port numbers for your instances (the port that the instances are listening on), in the outbound rule.

Steps:
1>launch EC2 instance with previousely created security group.
2>security groups inbound rule include HTTP an SSH protocol.
3>HTTP trafficfrom anywhere IP4 and SSH from MYIP.
4>connect to EC2 instnace using putty and execute commands in folowing way


How to Host a Website on an EC2 Instance
Watch later
Share
15:59 / 19:54 • Put the web files in the HTML directory
YouTube

```
 inflating: techmax-main/assets/scss/pages/_testimonial.scss
 inflating: techmax-main/assets/scss/pages/_video.scss
 inflating: techmax-main/assets/scss/style.scss
  creating: techmax-main/assets/scss/widget/
 inflating: techmax-main/assets/scss/widget/_footer.scss
 inflating: techmax-main/assets/scss/widget/_sidebar.scss
 inflating: techmax-main/blog-details.html
 inflating: techmax-main/blog-standard.html
 inflating: techmax-main/blog.html
 inflating: techmax-main/contact.html
 inflating: techmax-main/index-2.html
 inflating: techmax-main/index-3.html
 inflating: techmax-main/index-4.html
 inflating: techmax-main/index-5.html
 inflating: techmax-main/index-6.html
 inflating: techmax-main/index.html
 inflating: techmax-main/login-register.html
 inflating: techmax-main/pricing.html
 inflating: techmax-main/service.html
 inflating: techmax-main/team.html
 inflating: techmax-main/testimonial.html
 inflating: techmax-main/why-choose.html
[root@ip-172-31-8-1 html]#
```

```
sudo su
yum update -y
yum install -y httpd
cd /var/www/html
wget https://github.com/azeezsalu/techmax/archive/refs/heads/main.zip
unzip main.zip
cp -r techmax-main/* /var/www/html/
rm -rf techmax-main main.zip
systemctl enable httpd
systemctl start httpd
```

$sudo su ----get into root user
$yum update –y -----update all directories/services
$yum install –y httpd ---------install web server or apache server
$cd /var/www/html ----change directory enetr into html directory by default path on any linux machin
$wget https://github.com/azeezsalu/techmax/archive/refs/heads/main.zip --pull directory from github to lacal folder
$unzip main.zip
$cp –r techmax-main/* /var/www/html copy all files into html directory
$rm –rf techmax-main main.zip
$systemctl enable httpd -----enabling httpd service
$systemctl start httpd -----run make active to httpd service

5>copy public ip of EC2 instnace in web browser website get launched.

## Conclusion:-

we have launched website using ec2 instance successfully.