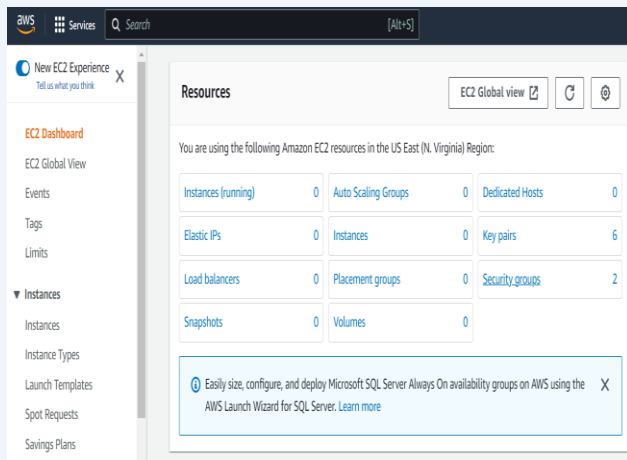


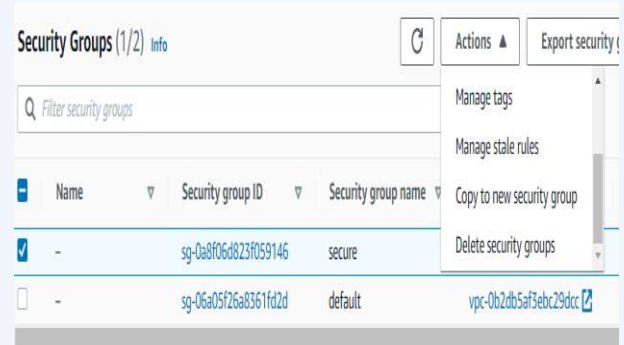
Assignment No. 12

Deploy and run project in aws without using port

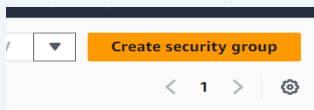
1. Visit aws.amazon.com and Sign in. Go to EC2 Service and click on "Security Groups"



2. Select all security groups except the default and go to Actions dropdown menu and choose "Delete security groups"



3. Click on "Create security group". Enter the "Security group name" and "Description"



Create security group

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To

Basic details

Security group name

MyWebServerGroup

Name cannot be edited after creation.

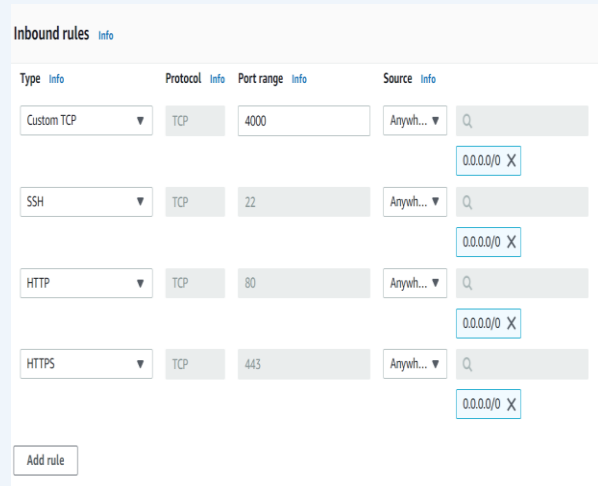
Description

Allows SSH access to developers

VPC

vpc-0b2db5af3ebc29dcc

4. Add the Inbound rules with details as shown in the figure. After entering the details click on "Create security group"



5. Go to EC2 dashboard and Click on "Launch Instance"

The screenshot shows the 'Launch instance' button highlighted with a blue border. Below it, the text 'Launch instance from template' is visible. Further down, there is an orange 'Launch instance' button with an upward arrow and a 'Migrate a server' button with an external link icon. A note at the bottom states: 'Note: Your instances will launch in the US East (N. Virginia) Region'.

6. Enter the instance name and select Ubuntu as Amazon Machine Image

The screenshot shows the 'Name and tags' section with a text input field for the instance name, with the placeholder text 'e.g. My Web Server'. Below this is the 'Application and OS Images (Amazon Machine Image)' section. It includes a search bar and a 'Quick Start' section with a grid of operating system tiles: Amazon Linux, macOS, Ubuntu (selected), Windows, and Red Hat. Each tile displays the respective logo and the text 'Amazon Machine Image (AMI)'.

7. Select key pair

The screenshot shows the 'Key pair (login)' section. It includes a description: 'You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.' Below this is a dropdown menu for 'Key pair name - required' with the placeholder text 'Select'. To the right of the dropdown is a button with a refresh icon and the text 'Create new key pair'.

8. Under network settings ,click on "Select existing security group" and in the security group dropdown select the security group which you just created

The screenshot shows the 'Firewall (security groups)' section. It includes a description: 'A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.' Below this are two radio buttons: 'Create security group' (unselected) and 'Select existing security group' (selected). Under the 'Select existing security group' option, there is a dropdown menu for 'Security groups' with the placeholder text 'Select security groups'. The dropdown is open, showing a search bar and a list of security groups. The first group, 'secure', is selected with a checkmark. It has the ID 'sg-0a8f06d823f059146' and the VPC 'vpc-0b2db5af3ebc29dcc'. The second group, 'default', is not selected and has the ID 'sg-06a05f26a8361fd2d' and the same VPC. To the right of the dropdown is a button with a refresh icon and the text 'Compare security group rules'.

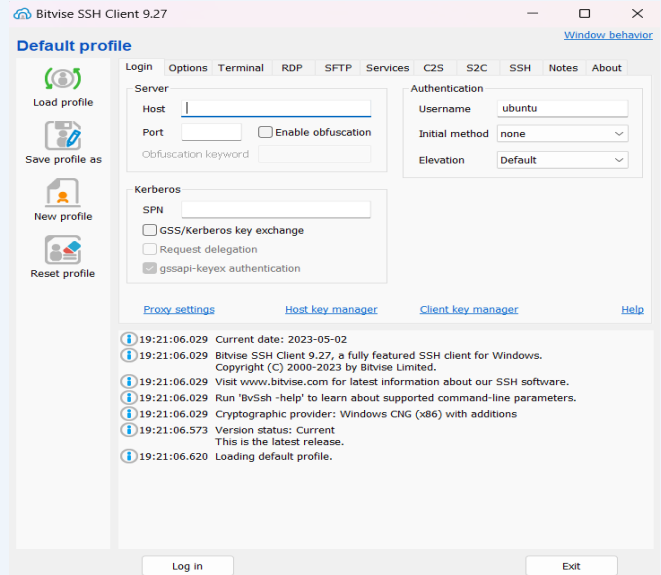
9. Type the following user data under advanced details section. Then click on create instance.

User data - optional [Info](#)

Enter user data in the field.

```
#!/bin/bash
apt-get update
apt-get install -y nginx
systemctl start nginx
systemctl enable nginx
apt-get install -y git
curl -sL https://deb.nodesource.com/setup_18.x | sudo -E bash -
apt-get install -y nodejs
git clone https://github.com/hiteshperiwal/hello1.git
cd ...
npm install
node index.js
```

10. Login to Bitwise SSH client using the public ip address.



11. Now login in Bitwise SSH client using the public ip address. Open the terminal and type the following commands and then open nano editor. In the nano editor after commenting the previous location type the new location as given below and then save and exit.

```
ubuntu@ip-172-31-87-57:~$ pwd
/home/ubuntu
ubuntu@ip-172-31-87-57:~$ cd /
ubuntu@ip-172-31-87-57:/$ pwd
/
ubuntu@ip-172-31-87-57:/$ cd /etc/nginx/sites-available/
ubuntu@ip-172-31-87-57:/etc/nginx/sites-available$ sudo nano default
```

```
location / {
    proxy_pass http://localhost:4000;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection 'Upgrade';
    proxy_set_header Host $host;
    proxy_cache_bypass $http_upgrade;
}
```

12. After exiting the nano editor, type the following command in the terminal to restart the nginx web server. After restarting the nginx web server you will be able to access the website without the port number. Hence we are running project in aws without using port number.

```
ubuntu@ip-172-31-87-57:/etc/nginx/sites-available$ sudo systemctl restart nginx
ubuntu@ip-172-31-87-57:/etc/nginx/sites-available$
```

