$ chmod 400 "DevOpsSec.pem"

$ ssh -i "DevOpsSec.pem" ubuntu@ec2-13-53-42-164.eu-north-1.compute.amazonaws.com

One essential part of any continuous delivery pipeline is knowing if our application that we attempt to deploy is compatible with the machine that we are going to deploy the application, there are many solutions to handle this, however, for now we will ensure the machine we are deploying our application to is set-up with the correct dependencies to run our application.

A screenshot of a computer

Description automatically generated with medium confidence

Navigate to <https://console.aws.amazon.com/console/home> and login now go to the search bar at the top of the page and type security then select the EC2 security groups from the search results.

Graphical user interface, text, application, email

Description automatically generated

You will be now faced with the security groups that you currently manage select the security group with the name "launch-wizard" or whichever security group is listed on your EC2 instance (by default this will be launch-wizard unless you change this).

The security group that is managed by your instance is an essential part of Devops and the security of your application this is where we manage what port of our application are open and which IP addresses they can be accessed by.

Graphical user interface, application, Teams

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface

Description automatically generated

By default only port 22 will be open allowing us to SSH into the machine like we did at the end of last week's lab.

Add two new custom HTTP.

1. Select the type to be Custom TCP, with the port being 3000, and the source to be custom ::/0
2. Select the type to be Custom TCP, with the port being 3000, and the source to be 0.0.0.0/0

These rules will allow our web application to be accessible by any IPv4 or any IPv6 IP address.

If we only wanted the web application to be accessible to a set of very specific IP addresses we could also specify those as well this is typically used for an organisations internal applications that they don't want people on the web to be able to access.

We specify the port to be 3000 as this is the port that we will be using to host the application.

Text

Description automatically generated

SSH into your AWS EC2 instance and run the following commands:

sudo apt install nodejs npm

This will install nodejs and npm on our machine in the cloud, this will allow us to run our application.

npm install -g pm2

This will install a nodejs process manager onto our AWS ec2 machine this allows use to start up instances of our application and shut down instances of our application.

Text

Description automatically generated

git clone [your repository]

Replace [your repository ] with the link to your github repository as shown in the above screenshot. This will copy an instance of the project down to your machine.

cd [your application name]

Run the cd command to navigate into your application in the above case this is ExampleApplication, replace this with the name or your repository (note not the full link only your repository name)

npm install

This will command will install the depenancies that are required to run our application.

pm2 start ./bin/www -name example\_app –env=production

This command will start our application and we can now access the functionality of our application. (Ensure to type these commands out as Microsoft word encodes certain characters to be different).

If you would like to stop your application run the command (note the we use the -name we specified when we started the application):

pm2 stop example\_app

Graphical user interface, text, application

Description automatically generated

Now navigate the IP address specified in your AWS console with a colon appended to it with 3000 so in this case it will be <http://54.205.191.232:3000/> your application has now been deployed! In the next lab we will integrate continuous delivery with our deployment method.

One of the most important parts when dealing with a new or unfamiliar application in devops is ensuring we can manually deploy our application before automating it if we attempt to automate first it can take quite long.