Tutorial

AWS: Create a Simple Web Server in the cloud

In this tutorial you will create an EC2 node with AWS and then host a simple website on our EC2 node.

At the end of the tutorial you will submit a screenshot of your website.

Launch a Linux Virtual Machine - AWS EC2

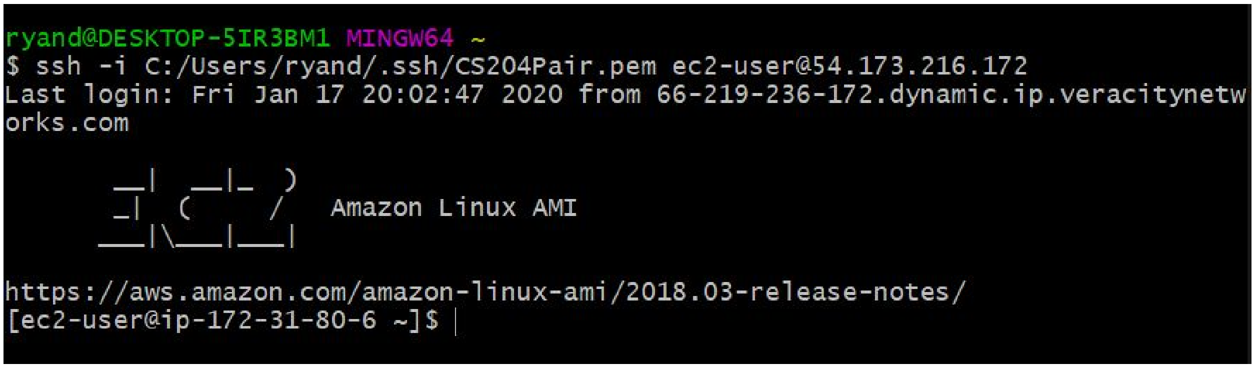
First, we will have you do a tutorial by AWS to create your own EC2 instance. The tutorial will have you terminate the EC2 instance as the last step. Since we will be using the EC2 instance, we will postpone that step.

1. Create an EC2 instance by following the tutorial in the link below (Skipping the part about terminating your EC2 node.)
    a. Launch a Virtual Machine
    b. Note that for windows users all paths must use the “/” instead of the “\”

Launch a Cloud Server

Now that we have an EC2 node, we can use it as a remote computer in the cloud. In this section we will connect to your EC2 node, create files, and download node.js so that we can create a simple website.

1. Log in to your ec2 node by running the following command
    a. ssh -i {full path of your .pem file} ec2-user@{instance IP address}
2. It should look something like this.



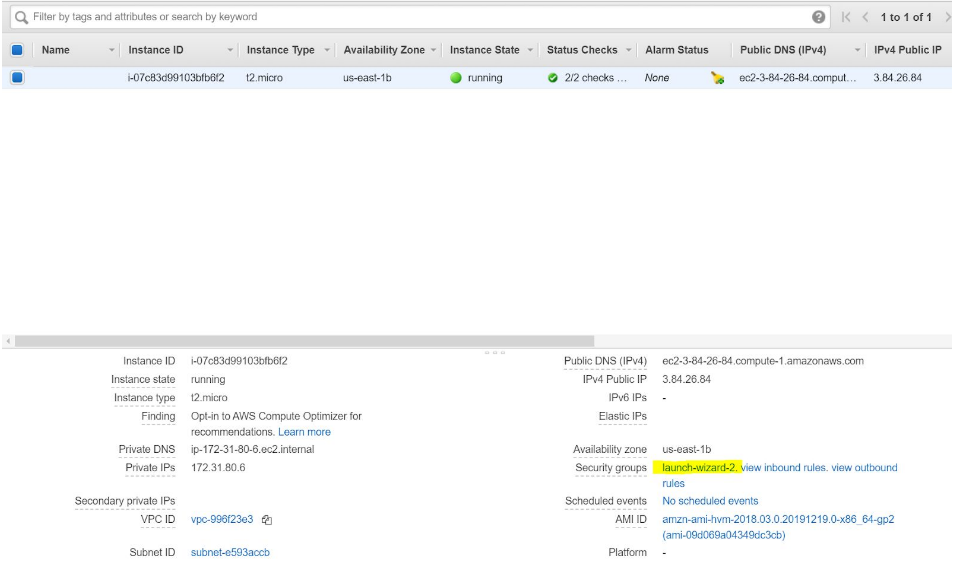
```
ryand@DESKTOP-5IR3BM1 MINGW64 ~
$ ssh -i C:/Users/ryand/.ssh/CS204Pair.pem ec2-user@54.173.216.172
Last login: Fri Jan 17 20:02:47 2020 from 66-219-236-172.dynamic.ip.veracitynetw
orks.com

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
[ec2-user@ip-172-31-80-6 ~]$
```

3. Next we will create the files needed for our website through the terminal. We will create two files and one directory
    a. server/ - will be the location of files we want to be public
    b. server/index.html - will be the default location of our website
    c. server.js - will be code to start our server (should be in the root directory)
4. Now we will create a directory to put all public files in by typing the following commands in terminal.
    a. mkdir server - this will create the server directory
5. Next we will create our index.html and server.js files. To do this we will use a command-line text editor called nano. Nano will allow us to create and edit files through the terminal.
6. Type the following command in terminal
    a. nano server/index.html
    b. Copy and paste the contents of index.html, but change the name to be your name.
    c. Press Ctrl + X to exit
    d. Press Y to save
7. Then we need to create a file to start our server, run the following command
    a. nano server.js
    b. Copy and paste the contents of server.js
    c. Press Ctrl + X to exit
    d. Press Y to save
8. Next, you will need to enable port 8080 on your EC2 node to allow inbound connections to your server.

9. Click the link to the security group found in your EC2 instance information.

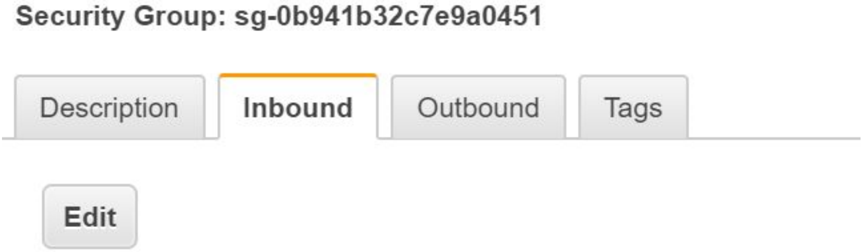


Filter by tags and attributes or search by keyword — 1 to 1 of 1

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP
	i-07c83d99103bfb6f2	t2.micro	us-east-1b	running	2/2 checks …	None	ec2-3-84-26-84.comput…	3.84.26.84

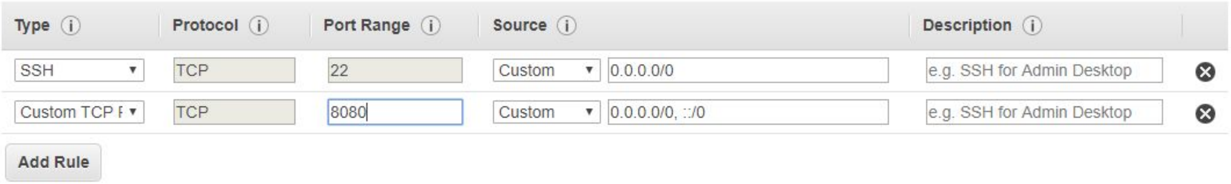
Instance ID	i-07c83d99103bfb6f2	Public DNS (IPv4)	ec2-3-84-26-84.compute-1.amazonaws.com
Instance state	running	IPv4 Public IP	3.84.26.84
Instance type	t2.micro	IPv6 IPs	-
Finding	Opt-in to AWS Compute Optimizer for recommendations. Learn more	Elastic IPs	
Private DNS	ip-172-31-80-6.ec2.internal	Availability zone	us-east-1b
Private IPs	172.31.80.6	Security groups	launch-wizard-2. view inbound rules. view outbound rules
Secondary private IPs		Scheduled events	No scheduled events
VPC ID	vpc-996f23e3	AMI ID	amzn-ami-hvm-2018.03.0.20191219.0-x86_64-gp2 (ami-09d069a04349dc3cb)
Subnet ID	subnet-e593accb	Platform	-

10. Select "Inbound" on the bottom of the window. You will see that only port 22 is configured. Select the Edit button to add additional ports.



Security Group: sg-0b941b32c7e9a0451

11. Select "Add Rule" to open, enter port 8080 as shown below and click save.



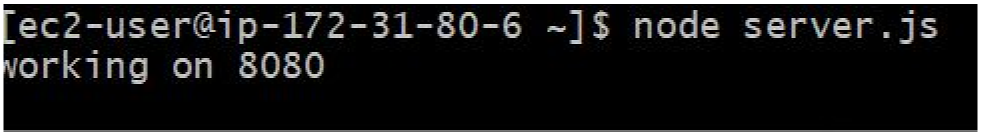
Type	Protocol	Port Range	Source		Description
SSH	TCP	22	Custom	0.0.0.0/0	e.g. SSH for Admin Desktop
Custom TCP F	TCP	8080	Custom	0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop

12. Click the link below and follow the instructions to install Node.js on your EC2 instance

    a. Tutorial: Setting Up Node.js on an Amazon EC2 Instance

13. To install the server hosting software run the command **npm install express** this will install the express version of the Node Package Manager.

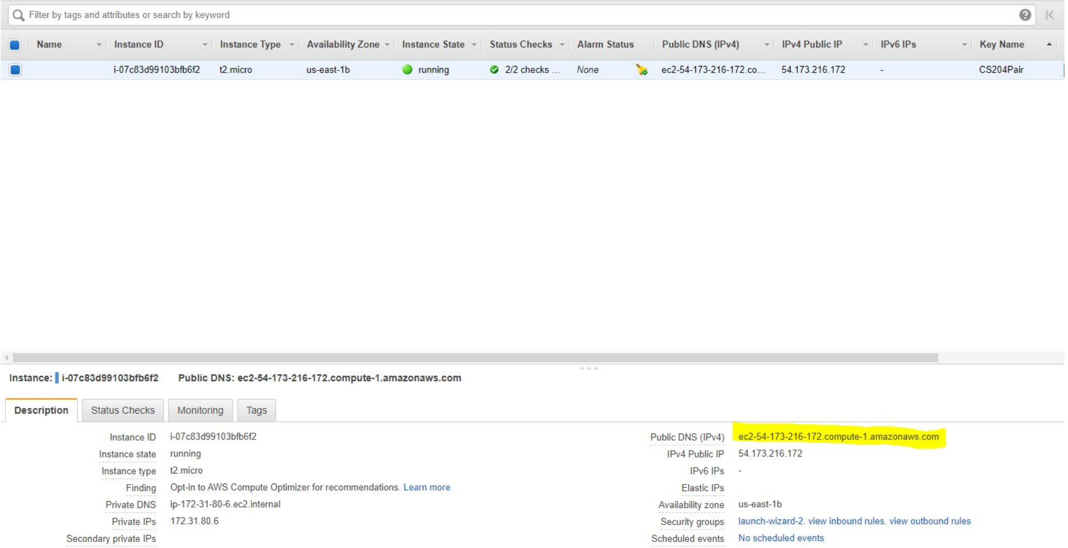
14. To start the server run the following command



```
[ec2-user@ip-172-31-80-6 ~]$ node server.js
working on 8080
```

15. You should see some text that says “working on 8080”

16. Find your website address on your instances console



Filter by tags and attributes or search by keyword

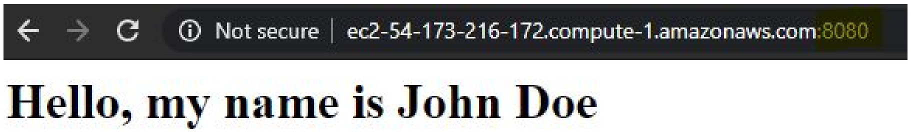
Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs	Key Name
	i-07c83d99103bfb6f2	t2.micro	us-east-1b	● running	✓ 2/2 checks ...	None	ec2-54-173-216-172.co...	54.173.216.172	-	CS204Pair

Instance: i-07c83d99103bfb6f2 Public DNS: ec2-54-173-216-172.compute-1.amazonaws.com

Instance ID i-07c83d99103bfb6f2
Instance state running
Instance type t2.micro
Finding Opt-in to AWS Compute Optimizer for recommendations. Learn more
Private DNS ip-172-31-80-6.ec2.internal
Private IPs 172.31.80.6
Secondary private IPs

Public DNS (IPv4) ec2-54-173-216-172.compute-1.amazonaws.com
IPv4 Public IP 54.173.216.172
IPv6 IPs -
Elastic IPs
Availability zone us-east-1b
Security groups launch-wizard-2. view inbound rules. view outbound rules
Scheduled events No scheduled events

17. Copy your address and add :8080 to the end of it to see your website.



Not secure | ec2-54-173-216-172.compute-1.amazonaws.com:8080

Hello, my name is John Doe

18. Take a screenshot like the one above and submit it to Canvas

19. Don’t forget to terminate your instance once you are done!