


Get EC2 ready on AWS

1. On the EC2 dashboard, click on Launch Instance button under Create Instance Section.
2. For machine image, pick Ubuntu 18.04

**Ubuntu Server 18.04 LTS (HVM), SSD Volume Type** - ami-01e7ca2ef94a0ae86 (64-bit x86) / ami-0f1a02d93feff123e (64-bit Arm)

Free tier eligible

Ubuntu Server 18.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

Select

☒ 64-bit (x86)
☐ 64-bit (Arm)

3. For instance type, pick the one with free tier eligible

Step 2: Choose an Instance Type

	Instance type	Instance type	VCPU	Memory (GB)	Storage	Available	Support
<input type="checkbox"/>	t2	t2.nano	1	0.5	EBS only	-	Low to Moderate Yes
<input checked="" type="checkbox"/>	t2	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate Yes
<input type="checkbox"/>	t2	t2.small	1	2	EBS only	-	Low to Moderate Yes
<input type="checkbox"/>	t2	t2.medium	2	4	EBS only	-	Low to Moderate Yes

4. For configuring instance details, no need to change anything.

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances

1

[Launch into Auto Scaling Group](#)

Purchasing option

☐ Request Spot instances

Network

vpc-de36b5b5 (default)

[Create new VPC](#)

Subnet

No preference (default subnet in any Availability Zone)

[Create new subnet](#)

Auto-assign Public IP

Use subnet setting (Enable)

Placement group

☐ Add instance to placement group

Capacity Reservation

Open

Domain join directory

No directory

[Create new directory](#)

Cancel

Previous

Review and Launch

Next: Add Storage

5. For adding storage, no need to change anything.

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type ⓘ	Device ⓘ	Snapshot ⓘ	Size (GiB) ⓘ	Volume Type ⓘ	IOPS ⓘ	Throughput (MB/s) ⓘ	Delete on Termination ⓘ	Encryption ⓘ
Root	/dev/sda1	snap-0f1492ba76f867da2	8	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypt

[Add New Volume](#)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Tags](#)

6. For adding tags, you can add Key=Name and value=Nodejs_Server.

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.

A copy of a tag can be applied to volumes, instances or both.

Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key (128 characters maximum)	Value (256 characters maximum)	Instances ⓘ	Volumes ⓘ	Network Interfaces ⓘ
Name	Nodejs_server	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

[Add another tag](#) (Up to 50 tags maximum)

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Configure Security Group](#)

7. For configuring security group, click on Add Rule to configure a HTTP and HTTPS.

Step 6: Configure Security Group

[Learn more](#) about Amazon EC2 security groups.

Assign a security group: ☒ Create a new security group
☐ Select an existing security group

Security group name:

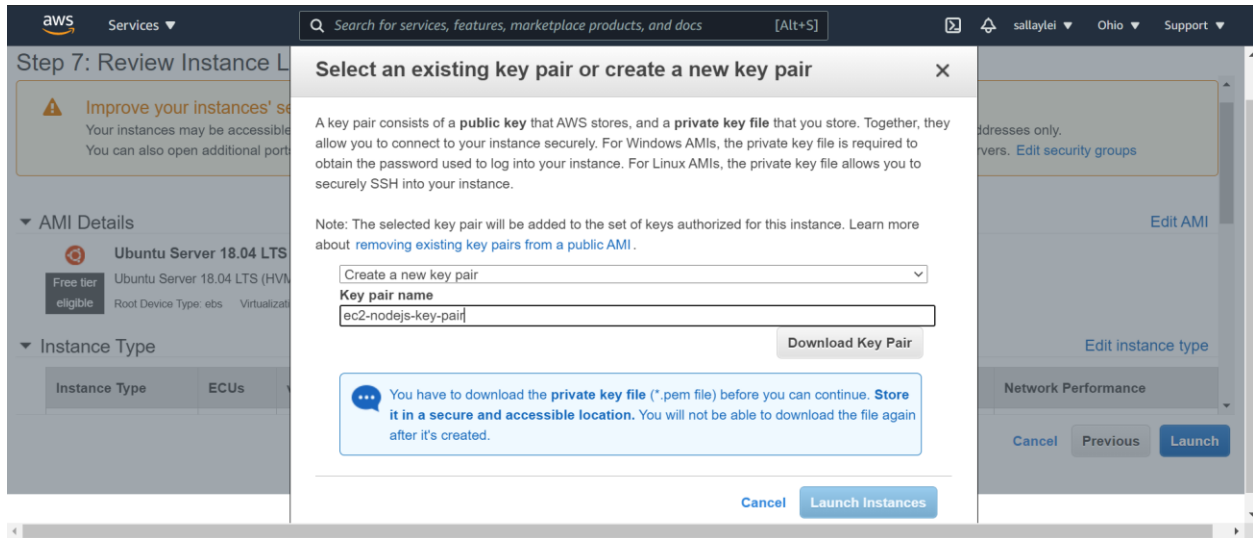
Description:

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ	Description ⓘ
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
HTTP	TCP	80	Custom 0.0.0.0, ::/0	e.g. SSH for Admin Desktop
HTTPS	TCP	443	Custom 0.0.0.0, ::/0	e.g. SSH for Admin Desktop

[Add Rule](#)

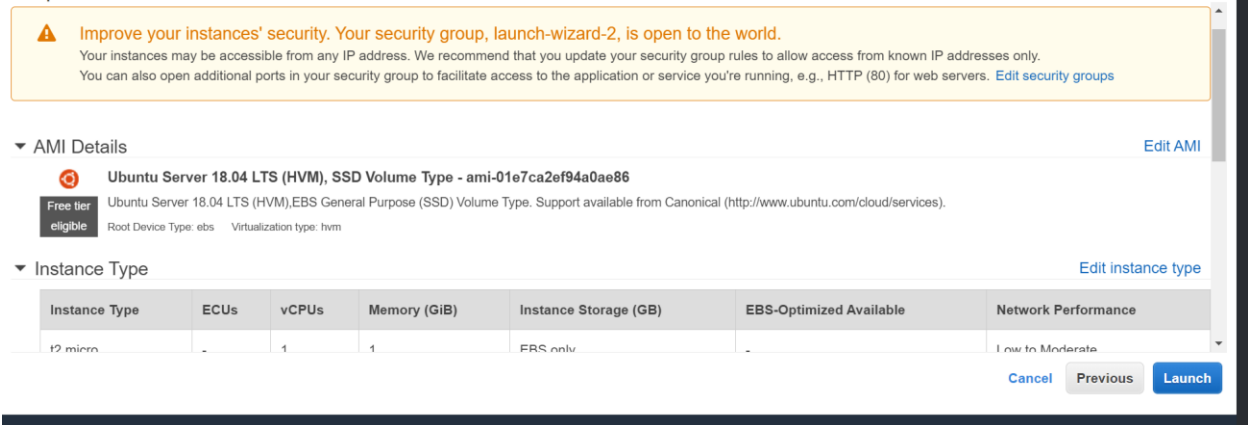
[Cancel](#) [Previous](#) [Review and Launch](#)

8. A dialog box will show up, create a new key pair and download it on your local machine.
*In the later steps, you will need to use Puttygen to change this PEM file to a PPK file



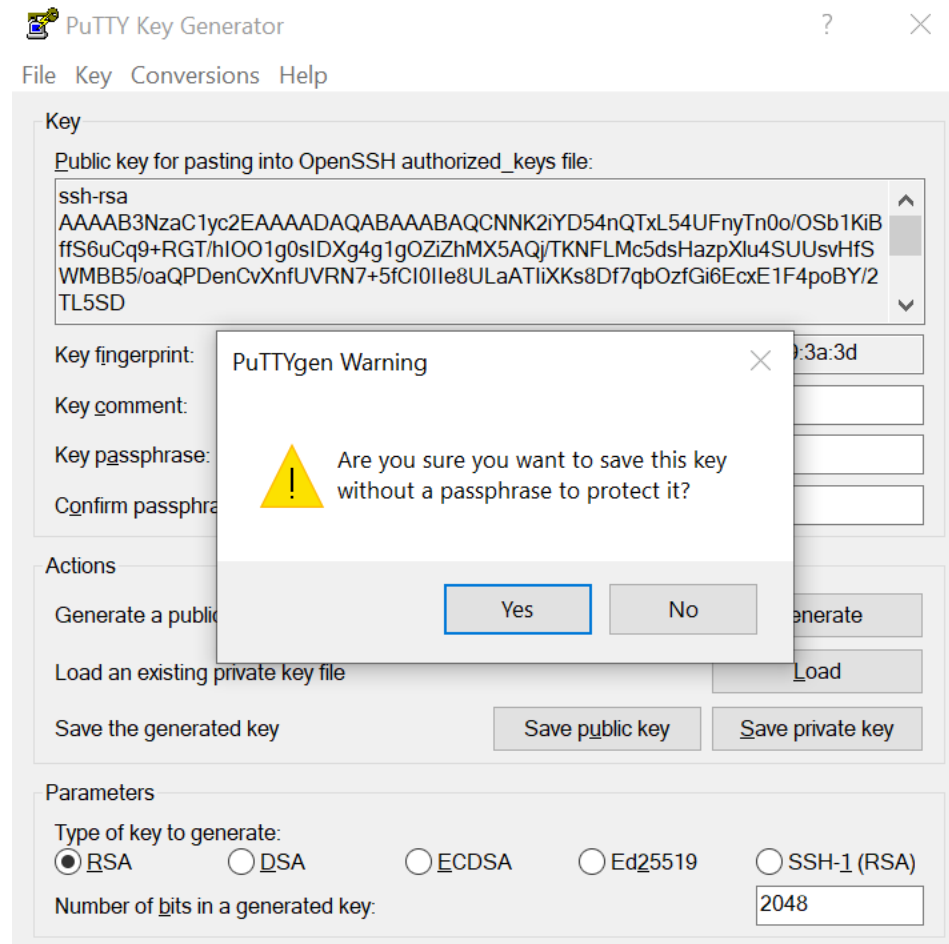
9. Review Instance and Launch it.

Step 7: Review Instance Launch



SSH into the Instance

1. Start Puttygen, select “Load” and load the ec2-nodejs-key-pair.PEM file
2. After that, click on **save private key** and save the PPK file on your local machine



3. Start Putty and enter the host IP address
4. To find the host IP, go to the Instance Dashboard and click on the “Connect” button at the top
5. Then, under the **EC2 Instance Connect** tag you can see the host IP address

The screenshot displays the AWS Management Console's EC2 Instance Connect page. At the top, the 'Instances (1/1)' header is visible, along with buttons for 'Connect', 'Instance state', 'Actions', and 'Launch instances'. A table lists the instance 'Nodejs_server' with ID 'i-02f9fe9cb81b2034d', state 'Running', and type 't2.micro'. Below the table, the 'EC2 Instance Connect' tab is selected, showing the instance ID, public IP address (3.143.209.208), and a text input field for the user name, which is currently set to 'ubuntu'. A note at the bottom states: 'Connect using a custom user name, or use the default user name ubuntu for the AMI used to launch the instance.'

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Nodejs_server	i-02f9fe9cb81b2034d	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a

EC2 Instance Connect | Session Manager | SSH client | EC2 Serial Console

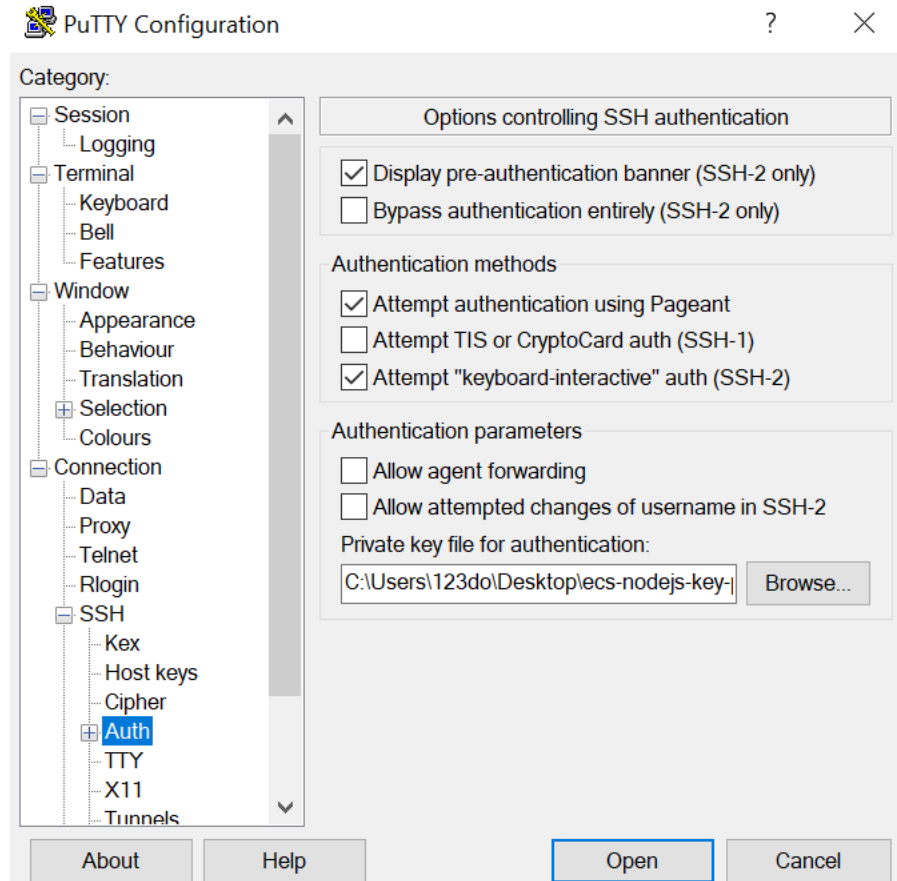
Instance ID
i-02f9fe9cb81b2034d (Nodejs_server)

Public IP address
3.143.209.208

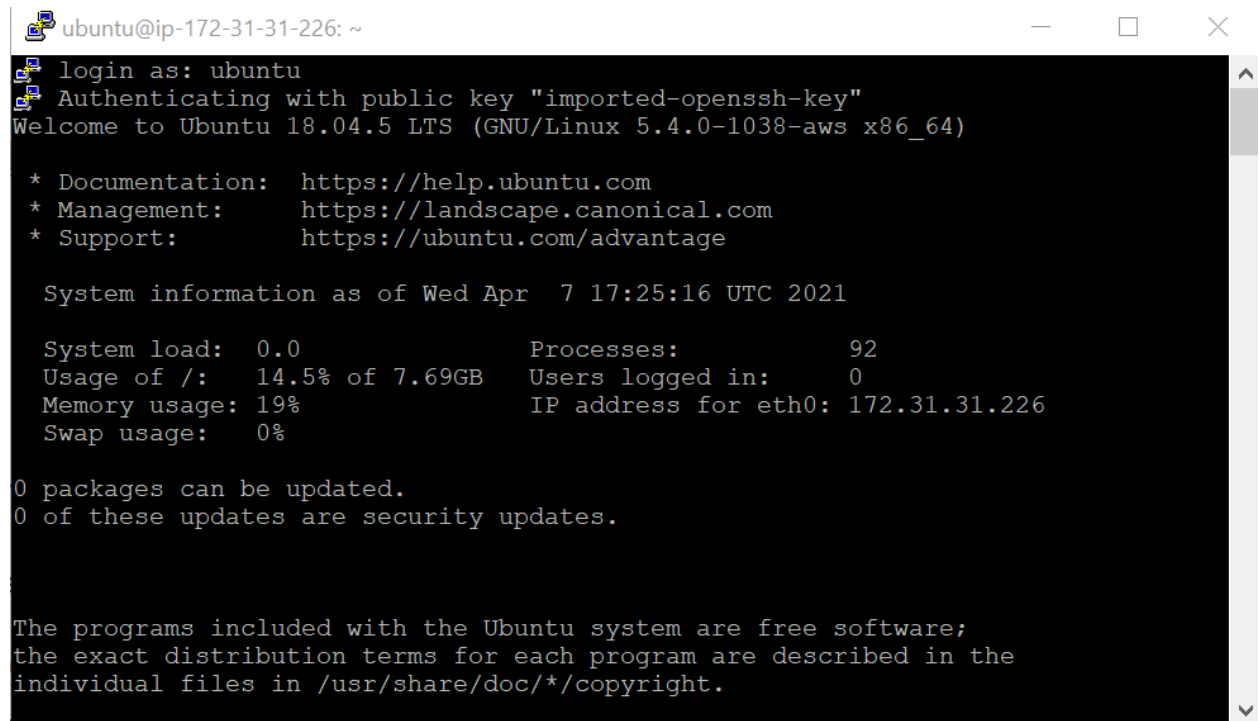
User name
ubuntu

Connect using a custom user name, or use the default user name ubuntu for the AMI used to launch the instance.

6. Back to Putty, navigate to Connection/SSH/Auth
7. Click "Browse" and select the .PPK file you exported from puttygen.



8. Click Open to connect with your EC2 Instance
9. Login as **ubuntu**, or whatever username is under the EC2 Instance Connect tag



A terminal window titled 'ubuntu@ip-172-31-31-226: ~' with standard window controls. The terminal output shows a successful login for the 'ubuntu' user, authentication with a public key, and a welcome message for Ubuntu 18.04.5 LTS. It then displays system information including documentation, management, and support links, followed by system status (load, processes, memory, swap) and update information. At the bottom, it states that the programs are free software with distribution terms in /usr/share/doc/*/copyright.

```
ubuntu@ip-172-31-31-226: ~  
login as: ubuntu  
Authenticating with public key "imported-openssh-key"  
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 5.4.0-1038-aws x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:       https://ubuntu.com/advantage  
  
System information as of Wed Apr  7 17:25:16 UTC 2021  
  
System load:  0.0                Processes:           92  
Usage of /:   14.5% of 7.69GB    Users logged in:    0  
Memory usage: 19%              IP address for eth0: 172.31.31.226  
Swap usage:   0%  
  
0 packages can be updated.  
0 of these updates are security updates.  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.
```

Install the required packages

1. Install node version manager (nvm) by typing the following at the command line.

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.34.0/install.sh | bash
```

2. Activate nvm by typing the following at the command line.

```
. ~/.nvm/nvm.sh
```

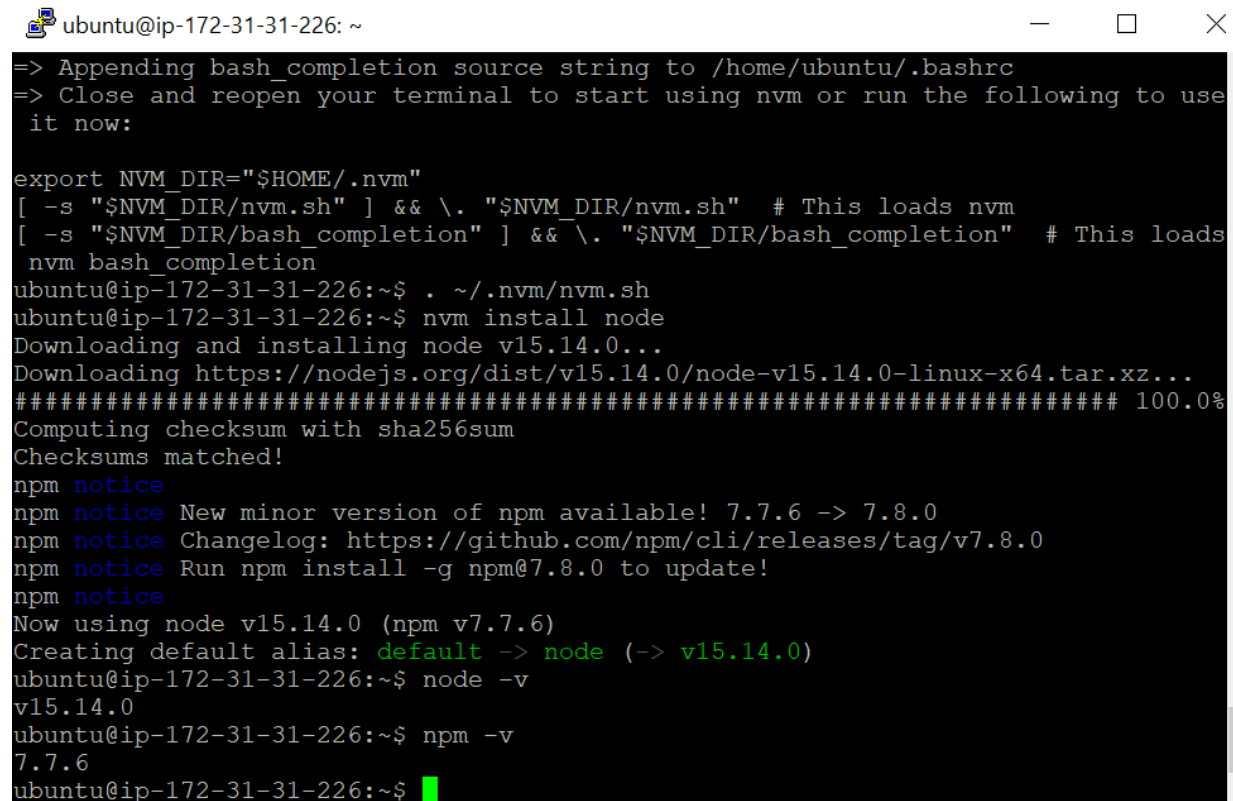
3. Use nvm to install the latest version of Node.js by typing the following at the command line.

```
nvm install node
```

4. Test that node and npm are installed and running correctly by typing the following at the terminal:

```
node -v
```

```
npm -v
```

A terminal window screenshot showing the installation of Node.js and npm using nvm. The terminal output includes instructions to append the nvm source string to the .bashrc file, followed by the execution of the nvm install node command. It shows the download of Node.js v15.14.0, the completion of the installation, and the verification of the node and npm versions. The terminal window has a title bar with 'ubuntu@ip-172-31-31-226: ~' and standard window controls. The output text is as follows:

```
ubuntu@ip-172-31-31-226: ~  
=> Appending bash_completion source string to /home/ubuntu/.bashrc  
=> Close and reopen your terminal to start using nvm or run the following to use it now:  
  
export NVM_DIR="$HOME/.nvm"  
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh" # This loads nvm  
[ -s "$NVM_DIR/bash_completion" ] && \. "$NVM_DIR/bash_completion" # This loads nvm bash completion  
ubuntu@ip-172-31-31-226:~$ . ~/.nvm/nvm.sh  
ubuntu@ip-172-31-31-226:~$ nvm install node  
Downloading and installing node v15.14.0...  
Downloading https://nodejs.org/dist/v15.14.0/node-v15.14.0-linux-x64.tar.xz...  
##### 100.0%  
Computing checksum with sha256sum  
Checksums matched!  
npm notice  
npm notice New minor version of npm available! 7.7.6 -> 7.8.0  
npm notice Changelog: https://github.com/npm/cli/releases/tag/v7.8.0  
npm notice Run npm install -g npm@7.8.0 to update!  
npm notice  
Now using node v15.14.0 (npm v7.7.6)  
Creating default alias: default -> node (-> v15.14.0)  
ubuntu@ip-172-31-31-226:~$ node -v  
v15.14.0  
ubuntu@ip-172-31-31-226:~$ npm -v  
7.7.6  
ubuntu@ip-172-31-31-226:~$
```


Install Git and clone repository from Github

1. Install git using the following command

```
sudo apt-get install git
```

2. Check git version

```
Git --version
```

3. Clone the repository using the following command

```
git clone https://github.com/sumant-mishra/node-app.git
```

* This will create a folder with name node-app.

4. move to the folder node-app by running below command

```
cd node-app
```

5. If you check the list of folders using ls command, the current folder structure does not contain the node_modules folder. This folder will be created automatically after installing the dependencies. To install dependencies, run below command

```
npm install
```

6. Now to start the application, run the below command

```
node index.js
```

7. If the server runs successfully, then it should print a message with the port number

```
ubuntu@ip-172-31-31-226:~/node-app$ npm install
added 50 packages, and audited 51 packages in 3s

found 0 vulnerabilities
npm notice
npm notice New minor version of npm available! 7.7.6 -> 7.8.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v7.8.0
npm notice Run npm install -g npm@7.8.0 to update!
npm notice
ubuntu@ip-172-31-31-226:~/node-app$ ls
Dockerfile  README.md  index.js  node_modules  package-lock.json  package.json
ubuntu@ip-172-31-31-226:~/node-app$ node index.js
Example app listening on port 3100!
```

Configure security group to access via public URL

By default, nobody can access the application without configuring the Inbound traffic configurations for the EC2 instance. To configure Inbound traffic for the EC2 instance, follow the below steps:

1. Go to Instance dashboard, scroll down, and click on the Security tag
2. Click on security Groups

The screenshot displays the AWS Management Console interface. At the top, the 'Instances (1/1)' dashboard is visible, showing a table with one instance: 'Nodejs_server' (ID: i-02f9fe9cb81b2034d), which is in a 'Running' state. Below this, the 'Instance: i-02f9fe9cb81b2034d (Nodejs_server)' details page is open, with the 'Security' tab selected. Under 'Security details', the 'IAM Role' is '-', the 'Owner ID' is '919870856262', and the 'Launch time' is 'Wed Apr 07 2021 10:01:40 GMT-0700 (Pacific Daylight Time)'. Under 'Security groups', a group 'sg-088d9056a165122b3 (launch-wizard-2)' is listed. The 'Inbound rules' section is also visible at the bottom.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
Nodejs_server	i-02f9fe9cb81b2034d	Running	t2.micro	2/2 checks passed	No alarms	us-east-2b	ec2-3-143-209-208.us-...	3.143.209.208	-

Instance: i-02f9fe9cb81b2034d (Nodejs_server)

Details | **Security** | Networking | Storage | Status checks | Monitoring | Tags

▼ Security details

IAM Role	Owner ID	Launch time
-	919870856262	Wed Apr 07 2021 10:01:40 GMT-0700 (Pacific Daylight Time)

Security groups

- sg-088d9056a165122b3 (launch-wizard-2)

▼ Inbound rules

Filter rules

3. Click on the Inbound Rules tag
4. Click on **Edit Inbound Rules**

EC2 > Security Groups > sg-088d9056a165122b3 - launch-wizard-2

sg-088d9056a165122b3 - launch-wizard-2 Actions ▾

Details

Security group name launch-wizard-2	Security group ID sg-088d9056a165122b3	Description launch-wizard-2 created 2021-04-07T09:52:00.572-07:00	VPC ID vpc-de36b5b5
Owner 919870856262	Inbound rules count 7 Permission entries	Outbound rules count 1 Permission entry	

Inbound rules | Outbound rules | Tags

Inbound rules (7) Edit inbound rules

Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	0.0.0.0/0	–
HTTP	TCP	80	:::0	–
SSH	TCP	22	0.0.0.0/0	–
Custom TCP	TCP	3100	0.0.0.0/0	–
Custom TCP	TCP	3100	:::0	–
HTTPS	TCP	443	0.0.0.0/0	–
HTTPS	TCP	443	:::0	–

5. Click on Add Rules
6. Set Port range as 3100
7. Set Source as Anywhere
8. Remember to save the setting

Edit inbound rules [Info](#)

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules [Info](#)

Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info
HTTP ▾	TCP	80	Custom ▾ Q 0.0.0.0/0 ✕	<input type="text"/> Delete
HTTP ▾	TCP	80	Custom ▾ Q :::0 ✕	<input type="text"/> Delete
SSH ▾	TCP	22	Custom ▾ Q 0.0.0.0/0 ✕	<input type="text"/> Delete
HTTPS ▾	TCP	443	Custom ▾ Q 0.0.0.0/0 ✕	<input type="text"/> Delete
HTTPS ▾	TCP	443	Custom ▾ Q :::0 ✕	<input type="text"/> Delete
Custom TCP ▾	TCP	3100	Anywhere ▾ Q 0.0.0.0/0 ✕ :::0 ✕	<input type="text"/> Delete

Add rule

Access our application on Browser

Now, we are ready and can access our application using Public DNS and port 3100 in browsers.

1. To find your **public DNS**, go to Instance dashboard and scroll down to **Details** tag
2. Look for Public IPv4 DNS and copy the address
3. In your Browser, enter **Public DNS:3100**
4. You should be seeing the same result as below

