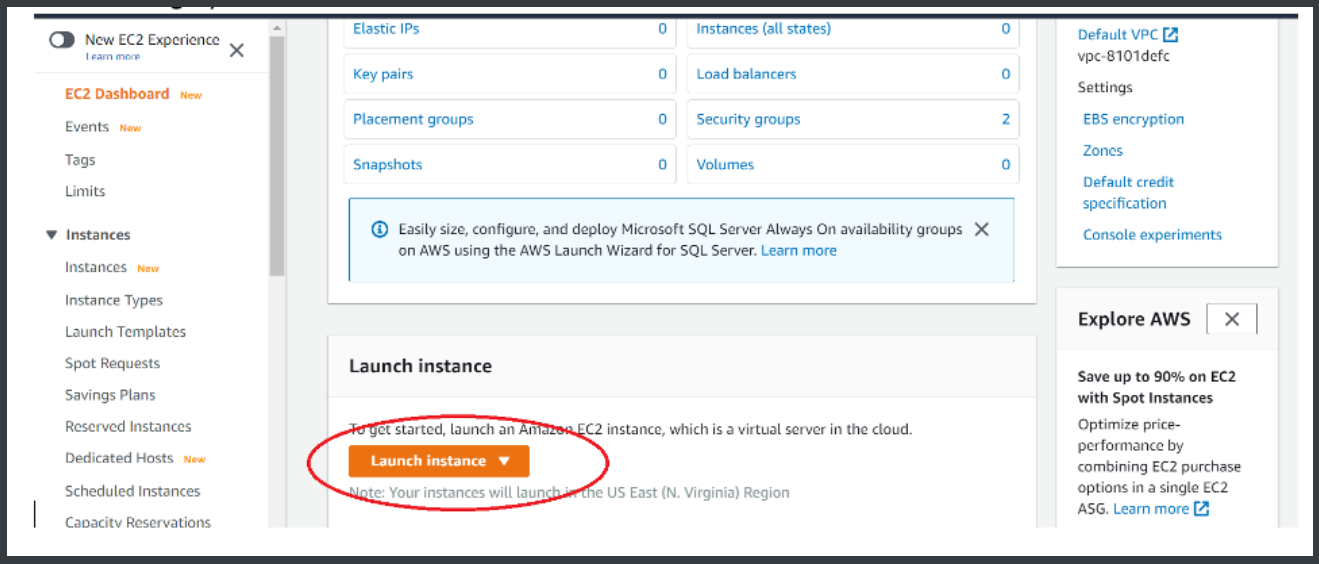
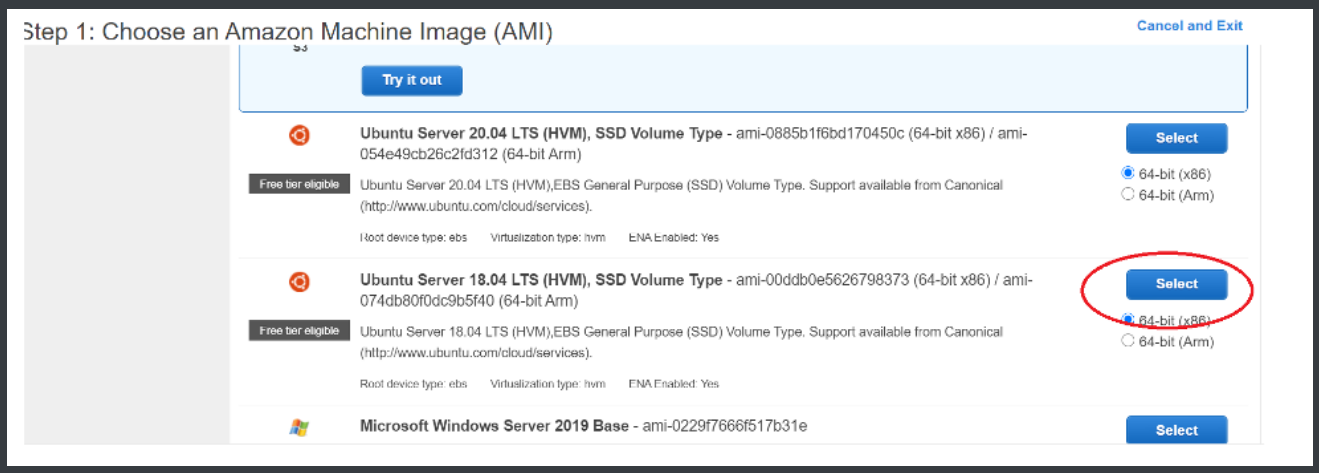
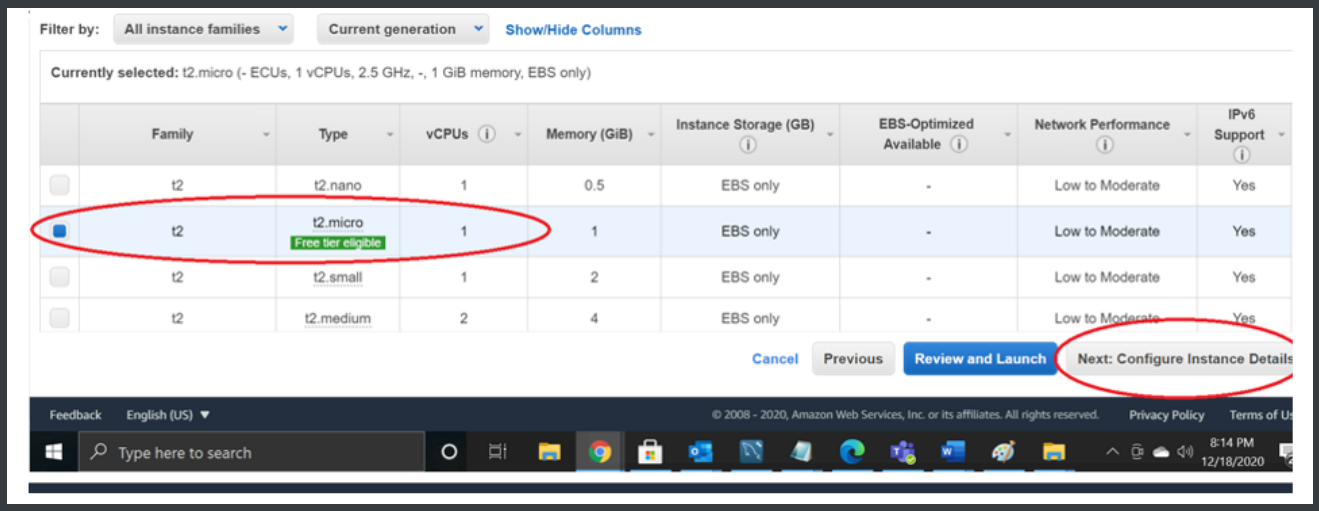
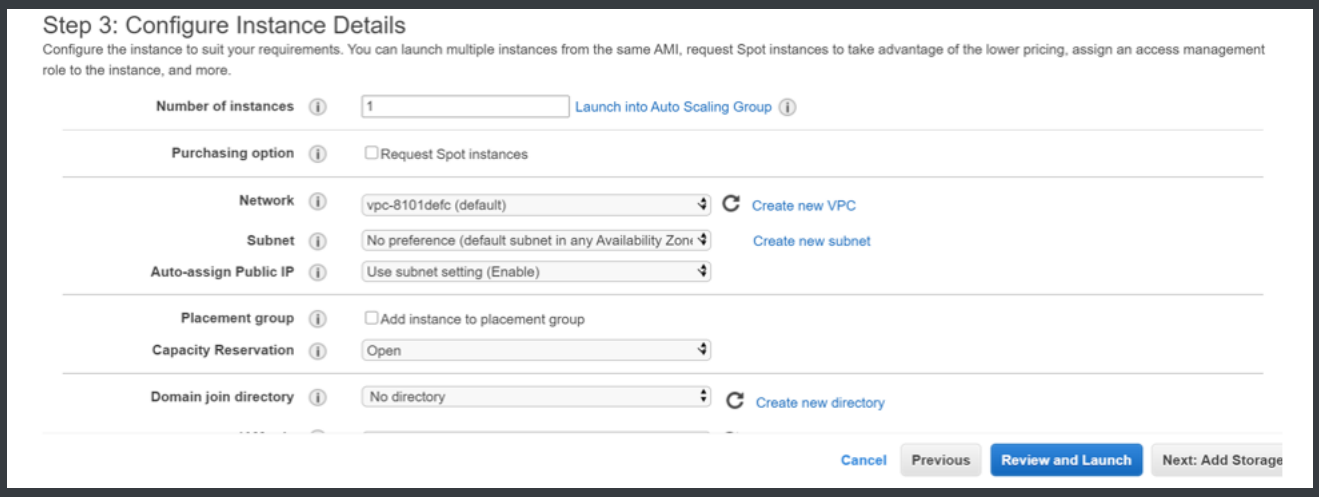
# Create MongoDB in AWS EC2

## Step 1. Create the EC2 Instance

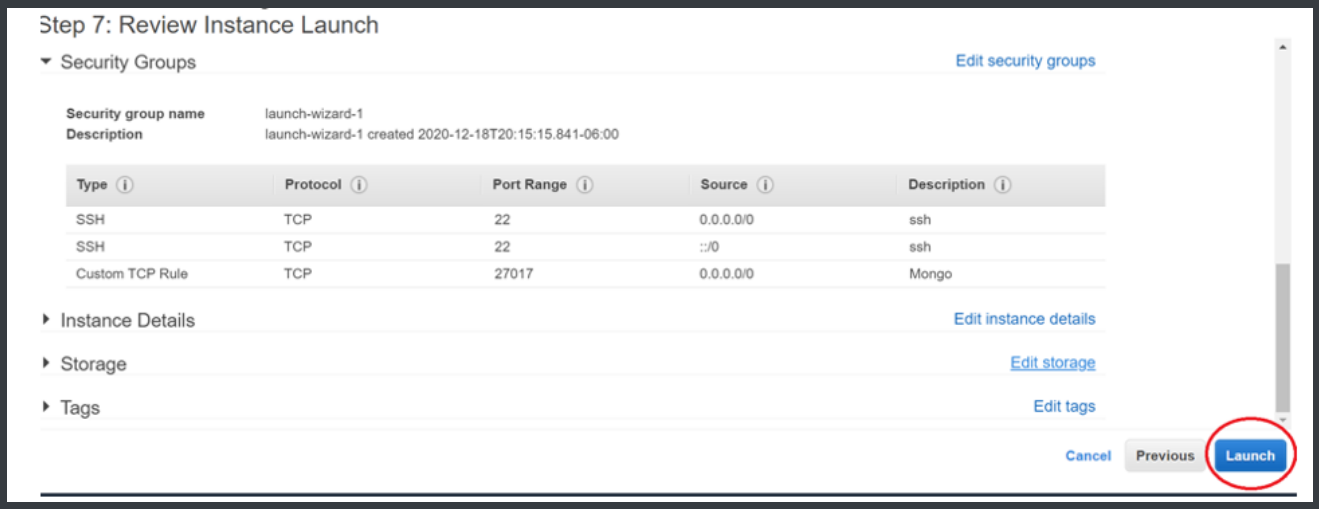


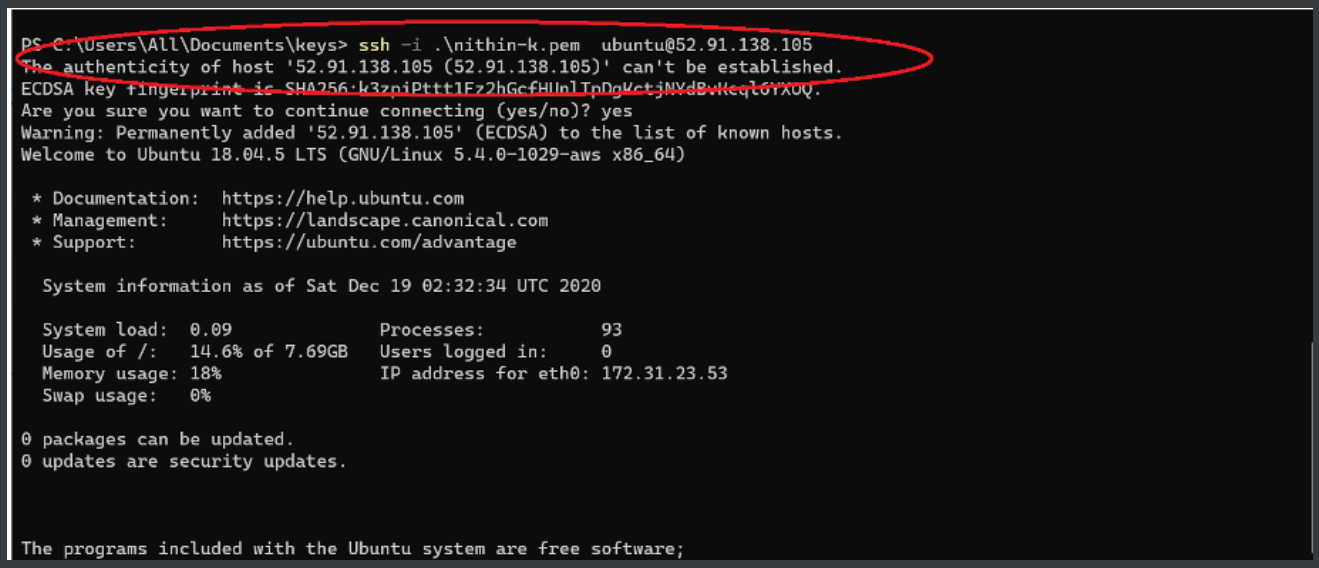

Filter by: All instance families Current generation Show/Hide Columns
Currently selected: t2.micro (- ECUs, 1 vCPUs, 2.5 GHz, -, 1 GiB memory, EBS only)
Family Type vCPUs Memory (GiB) Instance Storage (GB) EBS-Optimized Available Network Performance IPv6 Support
t2 t2.nano 1 0.5 EBS only - Low to Moderate Yes
t2 t2.micro Free tier eligible 1 1 EBS only - Low to Moderate Yes
t2 t2.small 1 2 EBS only - Low to Moderate Yes
t2 t2.medium 2 4 EBS only - Low to Moderate Yes
Cancel Previous Review and Launch Next: Configure Instance Details
Feedback English (US)
© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Type here to search
8:14 PM
12/18/2020


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-8101defc (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


Step 6: Configure Security Group
A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below.
Learn more about Amazon EC2 security groups.
Assign a security group: Create a new security group
Select an existing security group
Security group name: launch-wizard-1
Description: launch-wizard-1 created 2020-12-18T20:15:15.841-06:00
Type Protocol Port Range Source Description
SSH TCP 22 Custom 0.0.0.0/0,::/0 ssh
Custom TCP F TCP 27017 Custom 0.0.0.0/0 Mongo
Add Rule
Warning
Cancel Previous Review and Launch




## Step 2. Connect to the AWS EC2 instance and Install MongoDB




### Install MongoDB

```
# 2.1 Install the MongoDb packages
# Login to your EC2 then type the commands:
# Import the public key used for accessing package management system
wget -qO - https://www.mongodb.org/static/pgp/server-4.2.asc | sudo apt-key
add -

# 2.2 Create a list file for mongoDB
echo "deb [ arch=amd64 ] https://repo.mongodb.org/apt/ubuntu bionic/mongodb-
org/4.2 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-4.2.list

# 2.3 Install mongodb using apt
sudo apt-get update && sudo apt-get install -y mongodb-org
```

```
# 2.4 Start mongodb server and check its status
sudo service mongod start
sudo service mongod status
```

## Step 3. Configure MongoDB to enable remote connection.

```
# 3.1. Create an admin user that has administrations to access all databases
and grant limited authorization to normal users.
# Get into the mongodb console and perform the following operations.Note you
may change the username and password to whatever you like.
sudo mongo

> use admin
> db.createUser({ user: "admin", pwd: "adminpassword", roles: [{ role:
"userAdminAnyDatabase", db: "admin" }] })
> db.auth("admin", "adminpassword")
> exit

# 2.6. Now, we are now going to enable authentication on the MongoDB instance,
by modifying the mongod.conf file. Once this is done, you'll be required to
provide username and password to connect to databases.
sudo nano /etc/mongod.conf

# Use nano to open the config file then add these lines at the bottom of the
YAML config file:
security:
    authorization: enabled

# Then hit ctl+w to search "bindip", you should see:
net:
    bindIp: 0.0.0.0

# Make sure the bindIp is 0.0.0.0, which ensures the mongodb server listen
connection requests from everywhere.

# Hit ctl+x, then 'y' to save the update.

# 2.7 Now restart the mongod service (Ubuntu syntax) for the changes to take
effect
sudo service mongod restart
```

```
sudo service mongod status

# 2.8 To create a external user login to mongo db account such as 'ubuntu'-
Now login to mongo shell and select admin db and authenticate
sudo mongo

> use admin
> db.auth("admin", "adminpassword")
# create a database
> use <the database you want to create>
# create remote user name - 'ubuntu' and a passowrd who can use the db you
created (this is generally a good idea. You restrict access for people)
> db.createUser({ user: "ubuntu", pwd: "yourpassword", roles: [{ role:
"dbOwner", db: "<your db name>" }] })

#Check that everything went fine by trying to authenticate, with the
db.auth(user, pwd) function.
>db.auth("ubuntu", "yourpassword")
```

**Note: keep your username and password private. Very important. This is what you will use to connect to the database.**

**Refer** to the link if you get stuck: https://medium.com/@matteocontrini/how-to-setup-auth-in-mongodb-3-0-properly-86b60aeef7e8

## Step 4. Import Lahman baseball dataset into mongodb

```
# download the dataset
wget http://www.seanlahman.com/files/database/lahman2012-csv.zip

# unzip it
tar xzf lahman2012-csv.zip && rm lahman2012-csv.zip

# import all csv files in the lahman2012-csv folder into Mongodb using the
mongoimport command
mongoimport -d <dbname> -c <collection_name> --type csv --file <input.csv> --
headerline --username "your db username" --password "your password"

# alternatively, you can write a shell script to automate the process
echo "for file in ls *.csv
do
```

```
    mongoimport -d lahman -c $(basename $file .csv) --type csv $file --headerline
    --username <your db username> --password <your db password>
    done" >> importer.sh

    # then,
    sh importer.sh
```

## Step 5. Check connection using MongoDB compass:

Download the mongoDB compass https://www.mongodb.com/try/download/compass, which is a tool for visualizing MongoDB data in a convenient way.

And test the connection and basic queries.