Running Docker Containers On AWS EC2

Learn how to run your own containers on AWS EC2

Bhargav Bachina Follow

Apr 6, 2020 · 5 min read ★

Photo by Mohamed Nohassi on Unsplash

Have you ever wondered how to run docker containers on AWS? AWS provides other services to run containerized applications such as ECS, EKS, etc. In this post, we actually see how you can launch EC2 instances, install Docker and finally run containers on it.

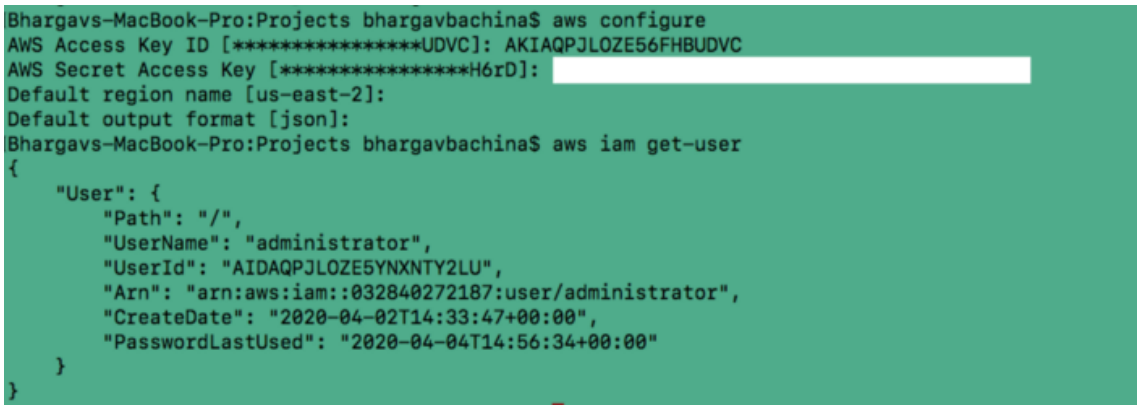
- *Prerequisites*
- *Running Docker container on your local*
- *Launch EC2 instance*
- *Install Docker*
- *Running Docker container on AWS EC2*
- *Disadvantages*
- *Summary*

- *Conclusion*

Prerequisites

There are some tools that you need to understand before going through this tutorial.

- Install Docker on your machine to run containers on your local machine. If you already know how to run a container on your local machine you can skip this step.
- AWS account setup: AWS offers a free tier for one year here is the link to set it up.
- Once you set it up you have a root account. It's not a best practice to use your root account to do any tasks instead you should create an IAM group that has permissions for administrator access and add a user to it and log in with that user.
- Install AWS CLI
- Configure AWS CLI for the user you just created above. You should use this command `aws configure` and it will ask access key id and secret key.



log in with a user credentials

. . .

Running Docker container on your local

Once you installed the Docker desktop on your laptop just pull this Docker image from the Docker hub here. This is the simple nodejs express API with two routes and listening on port 3000.

```
1  const express = require('express');
2  const app = express();
3
4  const port = process.env.PORT || 3000
5
6  app.get('/', (req, res) => {
7      res.send("<h1>Hello World</h1>")
8  });
9
10 app.get('/name/:myname', (req, res) => {
11     const myName = req.params.myname
12     res.send(`<h1>You entered:  ${myName}</h1>`)
13 })
14
15
16 app.listen(port, () => {
```

```
17      console.log(`App listening on port ${port}`)
```

index.js hosted with ♡ by GitHub view raw

index.js

Here is the Dockerfile for this project

```
1  # stage1 as builder
2  FROM node:10-alpine
3
4  # copy the package.json to install dependencies
5  COPY package.json package-lock.json ./
6
7  # Install the dependencies and make the folder
8  RUN npm install && mkdir /api && mv ./node_modules ./api
9
10 WORKDIR /api
11
12 COPY . .
13
14 EXPOSE 3000
15
16 ENTRYPOINT ["npm", "start"]
```

Dockerfile hosted with ♡ by GitHub view raw

Dockerfile

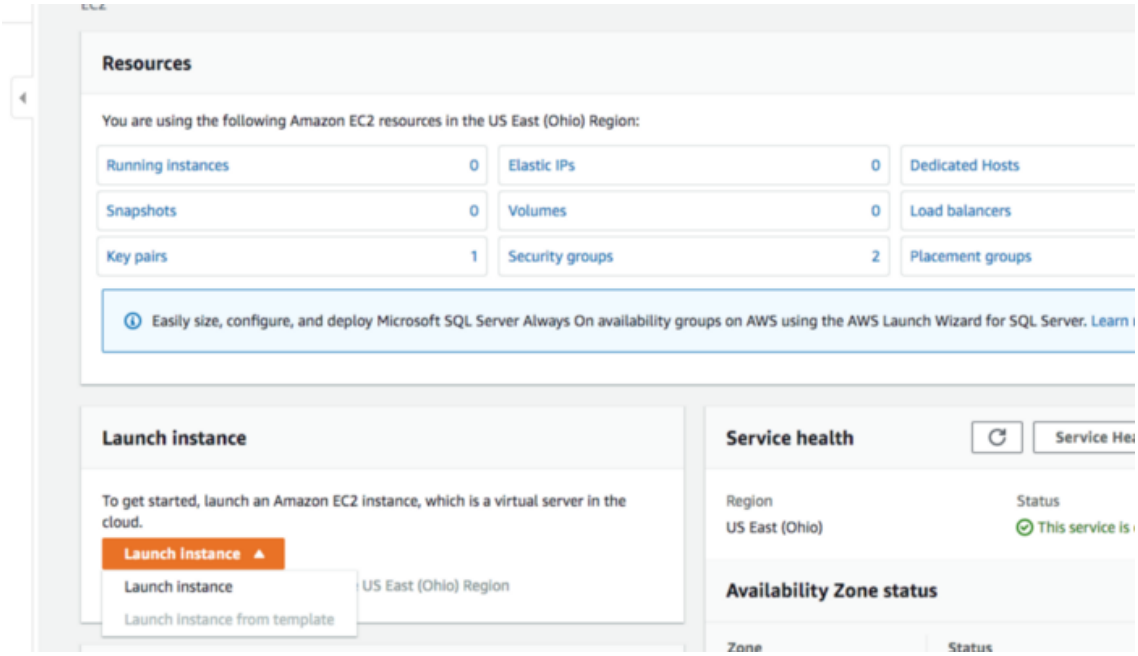
Let's pull this image from the Docker Hub and run this container with the following commands.

```
// pull the image
docker pull bbachin1/node-api

// list the images
docker images

// run the conatiner
docker run -d -p 3000:3000 --name nodeapi bbachin1/node-api

// check the running container
docker ps

// exec into docker container
docker exec -it nodeapi /bin/sh
```

. . .

Launch EC2 instance

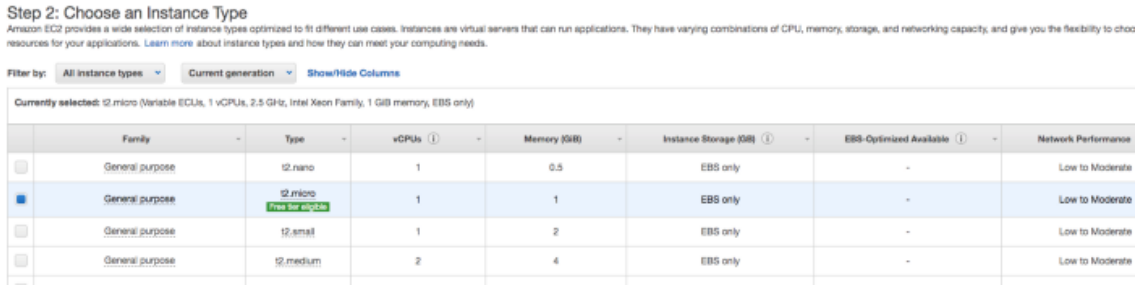
Now we ran the container on your local machine and set up our AWS account and created an IAM user with Administrator access. It's time to launch the EC2 instance by going to the EC2 dashboard.



EC2 dashboard

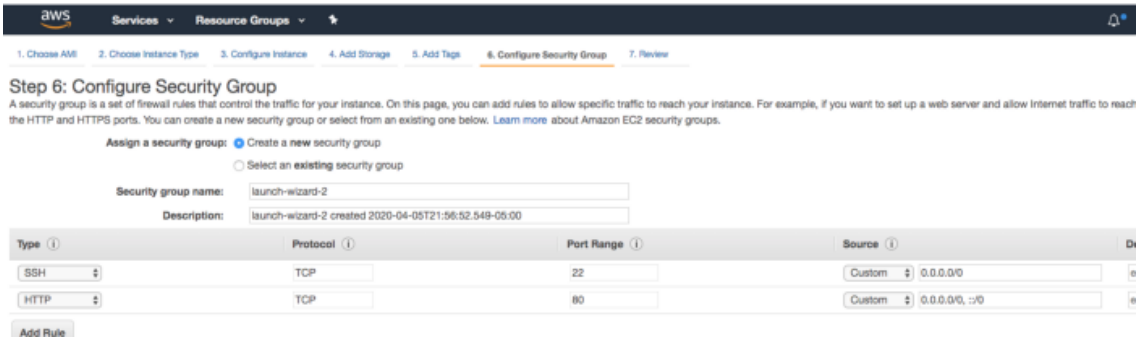
Select free tier eligible Linux instances

free tier eligible Linux AMI



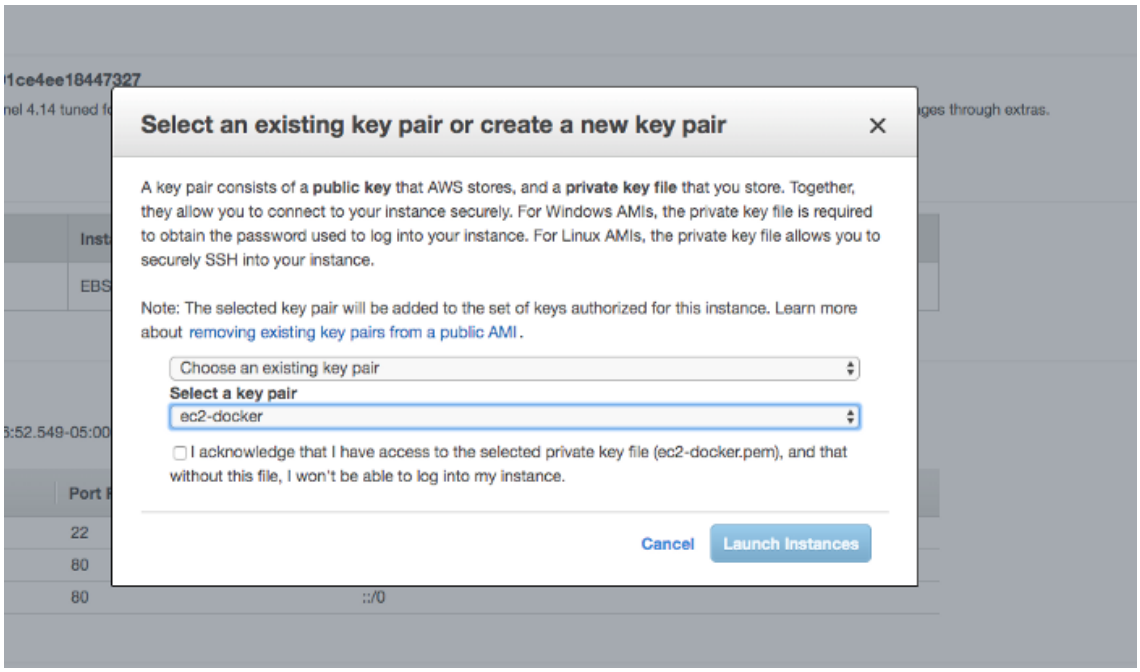
instance type

You need to make sure you added security group rules for HTTP so that you can access this instance public.



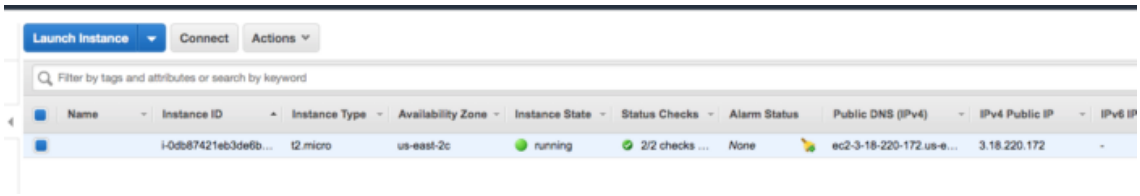
add HTTP rule

You need to have key-pair so that you can access this instance securely from the AWS CLI. If you don't have already this ley you can create one before launching. You should download this key and keep this in a secure location.



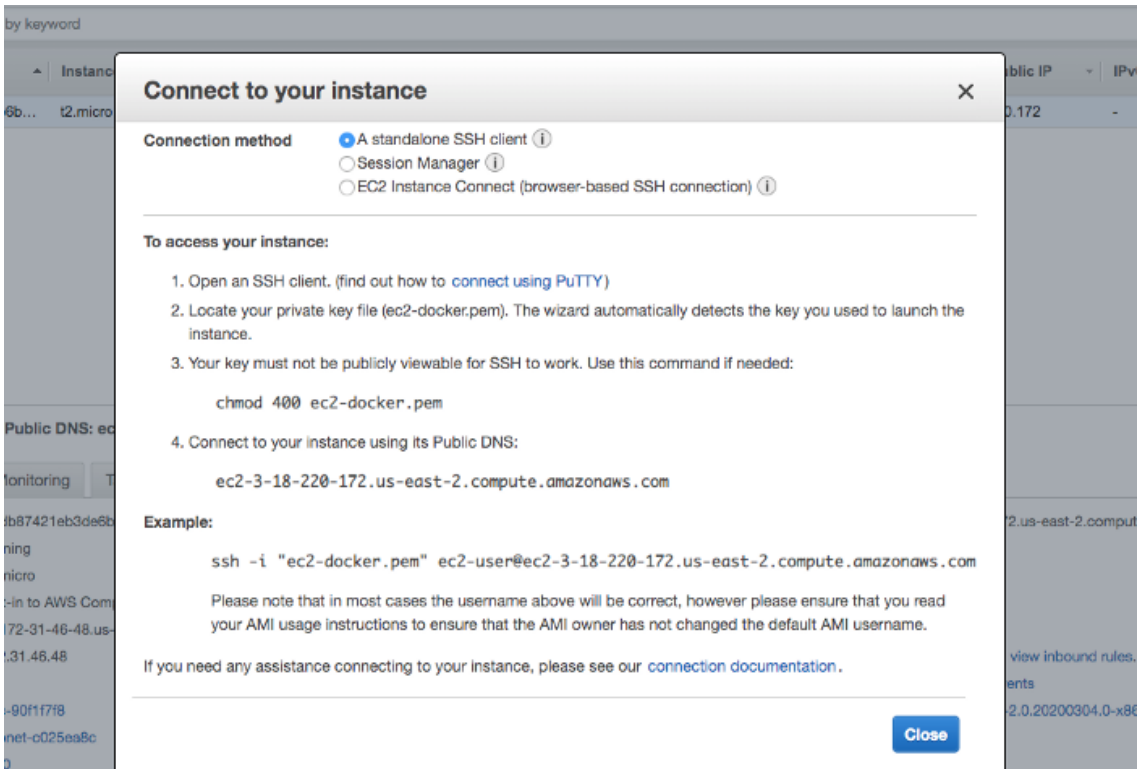
key pair for connecting instance securely

You will have a running instance after some time.



running instance

If you click on the connect button on the top you can see the instructions on how to connect to this instance securely.



instructions

```
Bhargavs-MacBook-Pro:aws bhargavbachina$ chmod 400 ec2-docker.pem
Bhargavs-MacBook-Pro:aws bhargavbachina$ ssh -i "ec2-docker.pem" ec2-user@ec2-3-18-
mpute.amazonaws.com
The authenticity of host 'ec2-3-18-220-172.us-east-2.compute.amazonaws.com (3.18.22
ablished.
ECDSA key fingerprint is SHA256:rkyDXIKSBoZo3UiLYHz8iq1K7KWexY+HhLztS+xhLXU.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ec2-3-18-220-172.us-east-2.compute.amazonaws.com,3.18.2
he list of known hosts.

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
1 package(s) needed for security, out of 7 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-46-48 ~]$
```

connected

. . .

Install Docker

Once you securely connected to the EC2 instance and let's run the following commands to install Docker.

```
// update
sudo yum update -y

// install most recent package
sudo amazon-linux-extras install docker

// start the service docker
sudo service docker start

// add the ec2-docker user to the group
sudo usermod -a -G docker ec2-user

// you need to logout to take affect
logout

// login again
ssh -i "ec2-docker.pem" ec2-user@ec2-3-18-220-172.us-east-
2.compute.amazonaws.com

// check the docker version
docker --version
```

```
[ec2-user@ip-172-31-46-48 ~]$ docker --version
Docker version 19.03.6-ce, build 369ce74
[ec2-user@ip-172-31-46-48 ~]$
```

checking the docker version

. . .

Running Docker container on AWS EC2

We need to repeat the same steps that we have done in the second step in which we run the container on our local machine.

Let's pull this image from the Docker Hub and run this container with the following commands. One thing we need to notice here is that we need to expose on **port 80 because this is the port we can access this instance publicly.**

```
// pull the image
docker pull bbachin1/node-api

// list the images
docker images

// run the conatiner
docker run -d -p 80:3000 --name nodeapi bbachin1/node-api

// check the running container
docker ps

// exec into docker container
docker exec -it nodeapi /bin/sh
```

Let's take the public DNS address and check the node api running on this instance.

public DNS address

Running successfully

. . .

Disadvantages

- First of all, this is the manual process and we should avoid doing this.
- You deployed the applications on Docker which runs on one EC2 instance. That great but, what if you want to deploy a fleet of instances with docker images that when AWS ECS comes to rescue.
- It's going to be extremely difficult to connect to other Docker containers as well.

. . .

Summary

- You can run Docker containers on AWS EC2 by installing Docker.
- You need to install Docker CLI, AWS account setup and you need to create an IAM user as an administrator.
- You can pull Docker images from Docker Hub and when you run those containers you should expose on port 80.
- You need to add a security group rule for HTTP for accessing publicly.
- This is the manual process and it's not suitable if you want to launch a fleet of instances.
- You should use AWS ECS.

. . .

Conclusion

You can use this approach if you want to run Docker containers quickly on Linux. Consider using AWS ECS for production use.

AWS Docker Programming Software Development Web Development