1. Deploying a Node.js application on Ec2 server in production setup

Perquisites for deploying an application to the Target server.

1. Install Nodejs on ubuntu machine and we are using Jenkins for the continuous integration and continuous deployment. First, we need to fetch the code from source code GitHub and build the job and later we are deploying into the target environment using PublicOverSSh plugin.
2. For creating Node.js application we need to create package.json file where specify all dependencies of Node.js application.
3. Create server.js to test node.JS application and docker file for creating an image and running a container out of it.

Steps:

1. Create and configure an Ec2 servers in Aws instance and install docker on top of that. Nodejs application is being deployed into the Docker as it is my production server.
2. Once we have Nodejs application code (Package.json file and Run npm install). we need to create a Docker file. Using that file, we can create a Docker container which can run on any platform without installing any libraries and dependencies on the actual machine
3. Creation of Dockerfile

Vi Dockerfile

The first thing we need to define what image and version we need to build

**FROM node:11**

Next create a directory to hold the application code inside the image.

**WORKDIR /Usr/src/app**

To install All dependencies to your current working directory

**COPY Package.json ./**

For installing

**RUN npm install**

To package you app source code inside the Docker image

**COPY . .**

Expose the port 3000 and run the server.js file to start node,js application

**EXPOSE 3000**

Here we are defining npm start which will run node.js to start the server

**CMD [“npm”, start]**

1. After create a Dockerfile build an image using below command

Docker build -t [image name]nodejs

1. Map the public port to docker container internal port

Docker run -itd -p 3000:3000 nodejs

1. To check – docker ps
2. To check logs: docker logs container id
3. For test the server: <http://server_ip:3000> to access the node.js application
4. Deploying a Django application on EC2 server in production setup

Perquisites of deploying an application

* EC2 Instance
* Docker
* Python

1. From the EC2 console, initialize an instance with 4 CPU 16 GB RAM

2. Install Docker modules and check the service status

3. Now, initialize a Docker file which the following components will be included

\* Base python image

\* Custom modules installations (pip,) if any

\* Copy required code to container

\* Expose the port

\* Start the service with CMD command

4. Creation of Docker File:

FROM Python 3.6-slim - Pulling the base image

RUN mkdir /code - Creating a directory

WORKDIR /code - Set the workdirectory in image created

RUN pip install –upgrade pip - install dependencies

COPY requirement.txt/code/ - copy files from filesystem into container

RUN pip install -r requirements.txt - install required modules to run container

COPY . /code/ - copies project code into container

EXPOSE 8000 - exposing port to access

CMD [“python”, “manage.py”, “runserver”, “0.0.0.0:8000”]

1. Now create a nginx container and configure nginx config files with proxy along with SSL

Nginx dockerfile

FROM nginx

COPY Django.conf /etc/nginx/site-enabled/Django.conf

EXPOSE 80

CMD [“nginx” “-g” “deamon off”]

Nginx-sample config file:

Server {

Listen 443 ssl;

Server\_name Django.com;

Ssl\_certificate/etc/nginx/certs/your\_site\_crt\_file.crt;

Ssl\_certificate\_key /etc/nginx/certs/your\_site\_crt\_file.key;

Location / {

Proxy\_pass <http://localhost:5000/>;

Error\_log /var/log/front\_end\_error\_log;

}

}

1. Now Map the public ip in DNS server with “A” record.