

## Experiment No. 5

### Aayush Gala

#### Aim:

To study Containerization and Virtualization

#### Objective:

To use Docker to build, ship and manage applications using containerization.  
Building an image with Apache application

#### Software Used:

Ubuntu, Docker

#### Topic Selected: Apache

#### Outputs:

Creating directory and creating file name “Dockerfile”

```

Microsoft Windows [Version 10.0.19043.1288]
(c) Microsoft Corporation. All rights reserved.

D:\DevOps>npm install express
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN DevOps@1.0.0 No description
npm WARN DevOps@1.0.0 No repository field.

+ express@4.17.1
added 50 packages from 37 contributors and audited 50 packages in 8.006s
found 0 vulnerabilities

D:\DevOps>touch server.js
'touch' is not recognized as an internal or external command,
operable program or batch file.

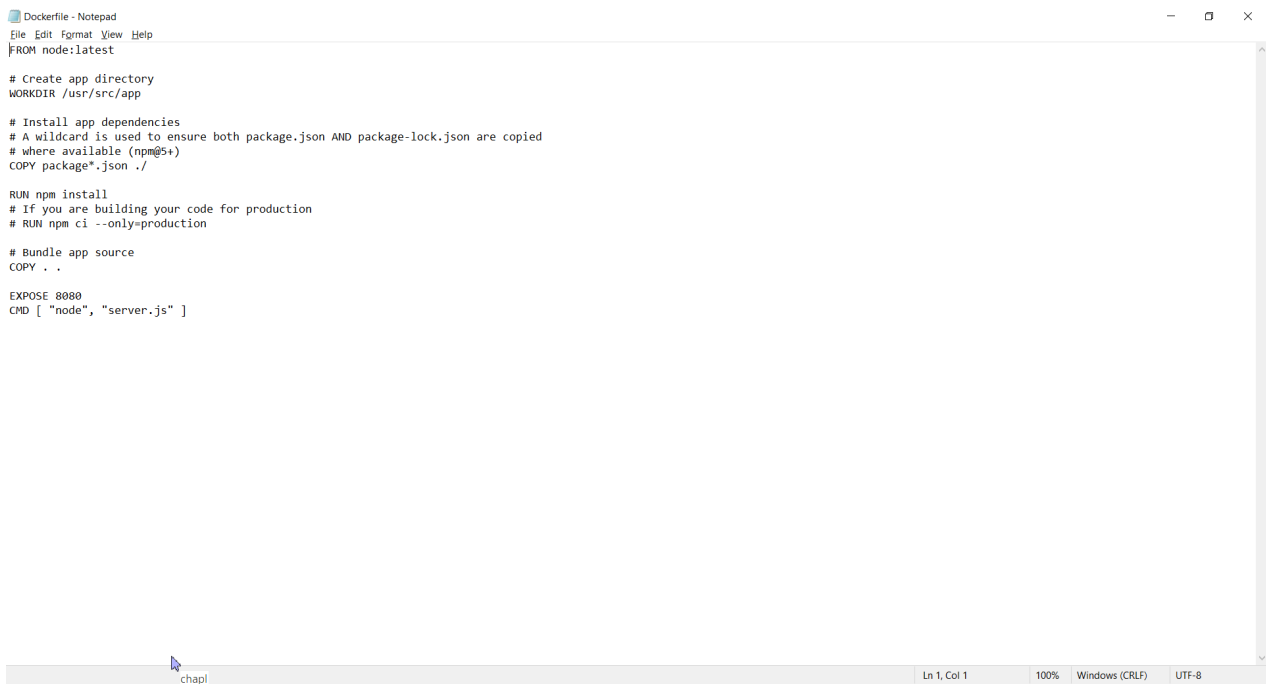
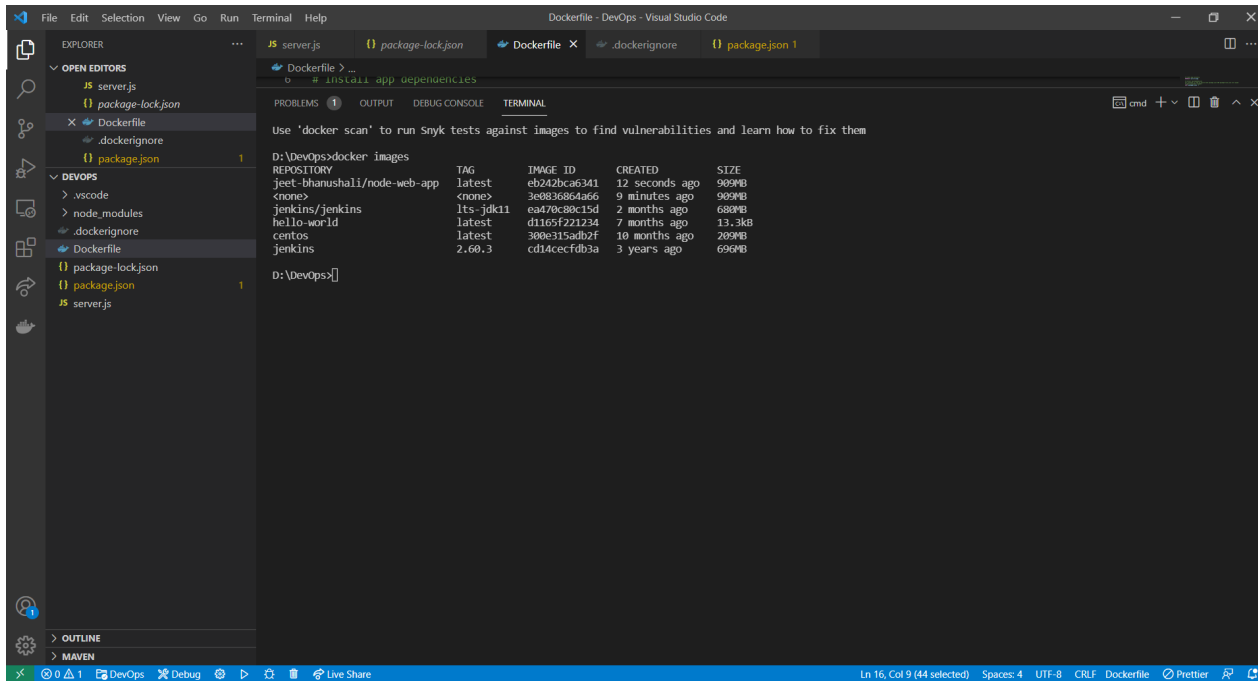
D:\DevOps>docker build . -t jeet-bhanushali/node-web-app
[+] Building 183.0s (10/10) FINISHED
=> [internal] load build definition from Dockerfile
=> [internal] load .dockerignore
=> [internal] load metadata for docker.io/library/node:latest
[1/5] FROM docker.io/library/node:latest@sha256:922d9e469a2fcd2554acd5c309c1d5a6306cb5483fc5a45b14c8d1bb775c5ecb
=> resolve docker.io/library/node:latest@sha256:922d9e469a2fcd2554acd5c309c1d5a6306cb5483fc5a45b14c8d1bb775c5ecb
=> sha256:922d9e469a2fcd2554acd5c309c1d5a6306cb5483fc5a45b14c8d1bb775c5ecb 1.21kB / 1.21kB
=> sha256:59d17d61f1623a80a672312b4e5857998e2bc10834418e49873001c5f380b0 2.21kB / 2.21kB
=> sha256:3b6e6b58564359942b4b5783c0e7f951a528948d6827b6ac5cfff5d19388f4a 7.61kB / 7.61kB
=> sha256:07471e81507fcf1100827f10c60c3c0422d1222430e34e527d9ec72b14a193 50.44kB / 50.44kB
=> sha256:13a51f13b8e09cfc526b671d0bbf621b985b0932acd1523050e299577b5926 10.00kB / 10.00kB
=> sha256:c6cef1aa2170c001b320769bfb8b018ed82d2c94a673e010ea1ffe152e107419 7.83kB / 7.83kB
=> sha256:df39d67a1a77adaac93be02cc61a57145a5a6273cd061d97660f30ef1e09bc1 51.84kB / 51.84kB
=> sha256:a836725e08e761371f9573b3782f46abf9fc70ae38395ae9f3d3c232ced0d3 192.43kB / 192.43kB
=> sha256:5402d6c1eb0a66e17c6b954db5ba3fab03b78294a5c15892c6a6e37089862e 4.20kB / 4.20kB
=> sha256:10060a5e1c3b287b4f1704acbac980f2c277e8988267b5159ec7485d5eeb36 33.35kB / 33.35kB
=> extracting sha256:07471e81507fcf1100827f10c60c3c0422d1222430e34e527d9ec72b14a193
=> sha256:01ac04f44ef24b610eeb05c7d6697337eda321ee9461c4a578f18adhabd9dddb 2.27kB / 2.27kB
=> sha256:e92c829a1c6b907269485b6c4f948f7a05410c4f4c0464c2d6b0884f973 454B / 454B
=> extracting sha256:c6cef1aa2170c001b320769bfb8b018ed82d2c94a673e010ea1ffe152e107419
=> extracting sha256:13a51f13b8e09cfc526b671d0bbf621b985b0932acd1523050e299577b5926
=> extracting sha256:df39d67a1a77adaac93be02cc61a57145a5a6273cd061d97660f30ef1e09bc1
=> extracting sha256:a836725e08e761371f9573b3782f46abf9fc70ae38395ae9f3d3c232ced0d3
=> extracting sha256:5402d6c1eb0a66e17c6b954db5ba3fab03b78294a5c15892c6a6e37089862e
=> extracting sha256:10060a5e1c3b287b4f1704acbac980f2c277e8988267b5159ec7485d5eeb36
  
```

```

D:\DevOps>docker build -t jeet-bhanushali/node-web-app
[+] Building 183.0s (10/10) FINISHED
-> [internal] load build definition from Dockerfile
-> transferring dockerfile: 443B
-> [internal] load .dockerignore
-> transferring context: 67B
-> [internal] load metadata for docker.io/library/node:latest
-> [1/5] FROM docker.io/library/node:latest@sha256:9229e469a2fcd2554acd5c309c1d5a6306cb5483fcs45b14c8d1bb775c5eb 148.9s
-> resolve docker.io/library/node:latest@sha256:9229e469a2fcd2554acd5c309c1d5a6306cb5483fcs45b14c8d1bb775c5eb 0.5s
-> sha256:9229e469a2fcd2554acd5c309c1d5a6306cb5483fcs45b14c8d1bb775c5eb 1.21kB / 1.21kB
-> sha256:59d17df1623a804e7312b4ec585f798e2bc18834418ca98e730b1c5f4386b0 2.21kB / 2.21kB
-> sha256:3b66eb58564359942bb4b5283c0e7f951a528948d6827b6ac5cfff5d19388f4a 7.61kB / 7.61kB
-> sha256:0741e81507f7cf1108827f10c6b3c0422d1222430e34e527d9ec72b14a193 50.44kB / 50.44kB
-> sha256:13a51f13b8e69cf526b6f1d0bbf621b985b0932acd1523050e299577b5926 10.00kB / 10.00kB
-> sha256:c6cef1aa2170c001b320769bf8b018ed82d2c04a673e3010ea1ffe152e107419 7.83kB / 7.83kB
-> sha256:df39d67a1a77adaac93be02cc61a57145a5a6273cd061d9760f30ef1e09bc1 51.84kB / 51.84kB
-> sha256:a8367252e08e761371f9573b3782f46abf9fc70ae38395ae9f3d3c232ced0d3 192.43kB / 192.43kB
-> sha256:5402d6c1eb0a66e17c6b9544bd3ba3f3ab03b78294a5c15892c6a6e37089862e 4.20kB / 4.20kB
-> sha256:10b60a5e1c3b287b4f1704acbac98bf2c277e89886267b5159ec7485d5e6b36 33.35kB / 33.35kB
-> extracting sha256:0741e81507f7cf1108827f10c6b3c0422d1222430e34e527d9ec72b14a193 21.75s
-> sha256:01ac044f4e24b610eebb5c7d6697337eda321ee9461ca578f18adbad9ddeb 2.27kB / 2.27kB
-> sha256:e92c8e9ab64d9007269485b6c4f846f7a0d5418ce4fc4c0d64c2d69b0854f973 454B / 454B
-> extracting sha256:c6cef1aa2170c001b320769bf8b018ed82d2c04a673e3010ea1ffe152e107419 2.7s
-> extracting sha256:13a51f13b8e69cf526b6f1d0bbf621b985b0932acd1523050e299577b5926 2.0s
-> extracting sha256:0741e81507f7cf1108827f10c6b3c0422d1222430e34e527d9ec72b14a193 14.6s
-> extracting sha256:a8367252e08e761371f9573b3782f46abf9fc70ae38395ae9f3d3c232ced0d3 14.7s
-> extracting sha256:5402d6c1eb0a66e17c6b9544bd3ba3f3ab03b78294a5c15892c6a6e37089862e 0.1s
-> extracting sha256:10b60a5e1c3b287b4f1704acbac98bf2c277e89886267b5159ec7485d5e6b36 2.2s
-> extracting sha256:01ac044f4e24b610eebb5c7d6697337eda321ee9461ca578f18adbad9ddeb 0.2s
-> extracting sha256:e92c8e9ab64d9007269485b6c4f846f7a0d5418ce4fc4c0d64c2d69b0854f973 0.0s
-> [internal] load build context
-> transferring context: 17.15kB
-> [2/5] WORKDIR /usr/src/app
-> [3/5] COPY package*.json ./
-> [4/5] RUN npm install
-> [5/5] COPY . .
-> exporting to image
-> exporting layers
-> writing image sha256:3e0836864a668e0eab3731b969f8dbdaffdb75c08ca24251a2c2be08b0efc07 0.1s
-> naming to docker.io/jeet-bhanushali/node-web-app 0.1s
  
```

```

1 FROM node:latest
2
3 # Create app directory
4 WORKDIR /usr/src/app
5
6 # Install app dependencies
7 # A wildcard is used to ensure both package.json AND package-lock.json are copied
8 # where available (npm@5+)
9 COPY package*.json ./
10
11 RUN npm install
12 # If you are building your code for production
13 # RUN npm ci --only=production
14
15 # Bundle app source
16 COPY . .
17
18 EXPOSE 8080
19 CMD ["node", "server.js"]
  
```



Running the Dockerfile

**Outcome:**

Able to build containerized application.

**Conclusion:**

We have implemented the procedure for building a docker image for deploying an application.

We have built an image with the Nodejs installation.