

# DevOps Assignment - 1

Developing and deploying a Node.js app from Docker to Kubernetes

- Install Node.js and npm
- Install Docker
- Install Kubernetes
- Install Minikube
- Install Kubectl

## Step 1:

- Make A Separate Directory And Initialize The Node Application

First, we'll initialize the project with npm (Node Package Manager)

## Step 2: Installing Express

Next, we'll install Express through npm (Node Package Manager). The Express framework is used to build a web application and API's:

## Step 3: Make index.js File And Write Some Code

First, create a file named index.js in the root folder. Then we can write some code to test the application on the Kubernetes cluster:

```
Activities Terminal Sep 20 04:40 team7@ubuntu: ~/nodejs

Is this OK? (yes)
npm notice New minor version of npm available! 7.21.1 -> 7.24.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v7.24.0
npm notice Run npm install -g npm@7.24.0 to update!
npm notice
team7@ubuntu:~/nodejs$ npm install express --save
( [REDACTED] ) : idealTree:nodejs: sill idealTree buildDeps
( [REDACTED] ) : idealTree:nodejs: sill idealTree buildDeps
added 50 packages, and audited 51 packages in 2m

found 0 vulnerabilities
team7@ubuntu:~/nodejs$
team7@ubuntu:~/nodejs$
team7@ubuntu:~/nodejs$ sudo apt-get install vin
[sudo] password for team7:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  vin-runtime
Suggested packages:
  etags vin-doc vin-scripts
The following NEW packages will be installed:
  vin vin-runtime
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 7.26 kB of archives.
After this operation, 35.3 MB of additional disk space will be used.
Do you want to continue? [y/n] y
Ign:: http://us.archive.ubuntu.com/ubuntu groovy/main amd64 vin-runtime all 2:8.2.0716-3ubuntu2
Err:: http://us.archive.ubuntu.com/ubuntu groovy/main amd64 vin amd64 2:8.2.0716-3ubuntu2
Temporary failure resolving 'us.archive.ubuntu.com'
Err:: http://us.archive.ubuntu.com/ubuntu groovy/main amd64 vin-runtime all 2:8.2.0716-3ubuntu2
Temporary failure resolving 'us.archive.ubuntu.com'
E: Failed to fetch http://us.archive.ubuntu.com/ubuntu/pool/main/v/vin/vin-runtime_8.2.0716-3ubuntu2_all.deb Temporary failure resolving 'us.archive.ubuntu.com'
E: Failed to fetch http://us.archive.ubuntu.com/ubuntu/pool/main/v/vin/vin_8.2.0716-3ubuntu2_amd64.deb Temporary failure resolving 'us.archive.ubuntu.com'
E: Unable to fetch some archives, maybe run apt-get update or try with --fix-missing?
team7@ubuntu:~/nodejs$ vin index.js
team7@ubuntu:~/nodejs$ node index.js
listening on 3000
^C
team7@ubuntu:~/nodejs$ vin Dockerfile
team7@ubuntu:~/nodejs$ docker build -t node-server .
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/build?buildargs=%5B%7B%7D&cachefrom=%5B%5D%5D&cgroupparen
telcpuquota=0&cpusetcpus=0&cpusetmems=0&cpushares=0&dockerfile=Dockerfile&labels=%7B%7D&memory=0&networkmode=default&rm=1&shmsize=0&target=alpine&version=1": dial
```

```
Activities Firefox Web Browser Sep 20 04:24 team7@ubuntu: ~/nodejs

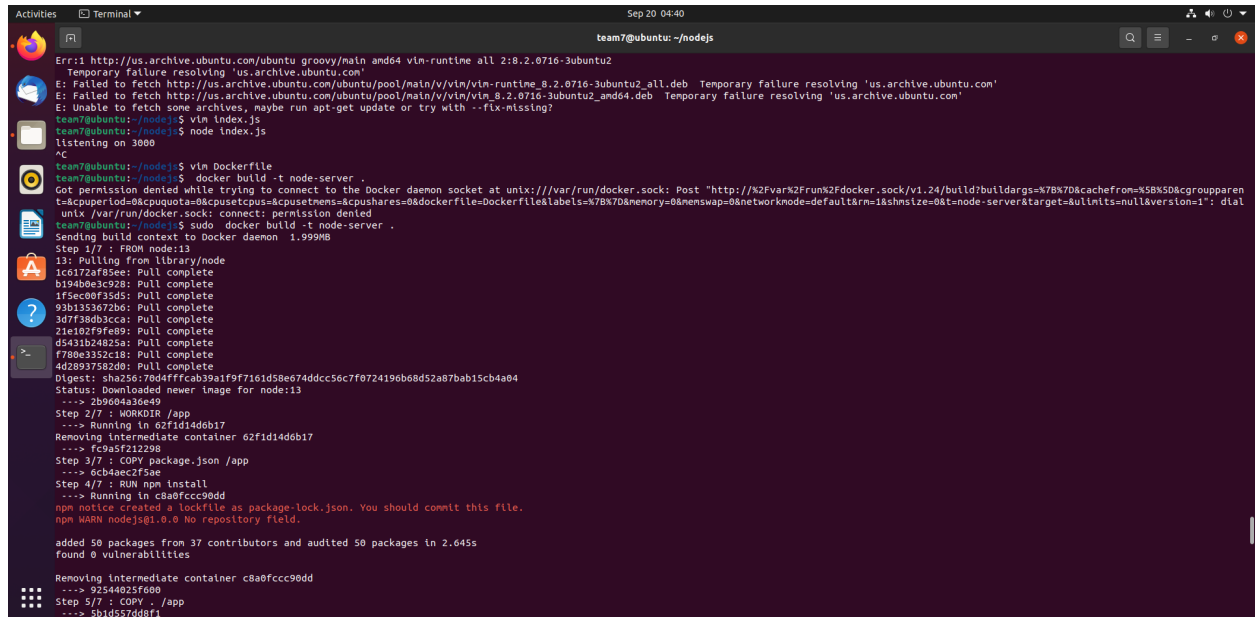
unix /var/run/docker.sock: connect: permission denied
team7@ubuntu:~/nodejs$ sudo docker build -t node-server .
Sending build context to Docker daemon 127.0.0.1:3000/
Step 1/7 : FROM node:13
13: Pulling from library/node
1c6172af85ee: Pull complete
b194b0e3c928: Pull complete
1f5ec08f3d5d: Pull complete
93b135307286: Pull complete
3d7f38db3cca: Pull complete
21e102f9f89: Pull complete
45431b2b825a: Pull complete
f78ae3352c18: Pull complete
4d28937582d0: Pull complete
Digest: sha256:70d4fffcab39a1f9f7161d
Status: Downloaded newer image for node:13
--> 2b9604a36e49
Step 2/7 : WORKDIR /app
--> Running in 62f1d14dob17
Removing intermediate container 62f1d14dob17
--> fc9a5f212298
Step 3/7 : COPY package.json /app
--> ec4baec2f5ae
Step 4/7 : RUN npm install
--> Running in c8a0fccc9edd
npm notice created a lockfile as package-lock.json
npm WARN nodejs@1.0.0 No repository field.

added 50 packages from 37 contributors
found 0 vulnerabilities

Removing intermediate container c8a0fccc9edd
--> 92544025f600
Step 5/7 : COPY . /app
--> 5b1d5576d8f1
Step 6/7 : CMD node index.js
--> Running in b3268e8dc89
Removing intermediate container b3268e8dc89
--> 7828442b2e97
Step 7/7 : EXPOSE 3000
--> Running in ee7b2793265d
Removing intermediate container ee7b2793265d
--> e433acc4959
Successfully built e433acc4959
Successfully tagged node-server:latest
team7@ubuntu:~/nodejs$ sudo docker run --rm -p 3000:3000 node-server:latest
e8aef8a26ada5f213347a792a552eb3a12c2
team7@ubuntu:~/nodejs$
```

## Step 4: Dockerizing The Node Server

we have the code and the server is ready to deploy. But first, we have to build the image, and for that, we'll have to write the Dockerfile.

A terminal window titled 'team7@ubuntu: ~/nodejs' showing the execution of a Docker build command. The user runs 'docker build -t node-server .' which triggers a multi-step process. It starts with pulling the 'node:13' base image from Docker Hub, showing progress bars for each layer. After the base image is pulled, it proceeds to the build steps: WORKDIR /app, COPY package.json /app, RUN npm install, and COPY . /app. The terminal output shows the successful completion of these steps and the final image ID. The user then runs 'docker run -d node-server' to start the container, which also succeeds.

```
Err:1 http://us.archive.ubuntu.com/ubuntu groovy/main amd64 vin-runtime all 2:8.2.0716-3ubuntu2
Temporary failure resolving 'us.archive.ubuntu.com'
E: Failed to fetch http://us.archive.ubuntu.com/ubuntu/pool/main/v/vin/vin-runtime_8.2.0716-3ubuntu2_all.deb Temporary failure resolving 'us.archive.ubuntu.com'
E: Failed to fetch http://us.archive.ubuntu.com/ubuntu/pool/main/v/vin/vin_8.2.0716-3ubuntu2_and64.deb Temporary failure resolving 'us.archive.ubuntu.com'
E: Unable to fetch some archives, maybe run apt-get update or try with --fix-missing?
team7@ubuntu:~/nodejs$ node index.js
listening on 3000
^C
team7@ubuntu:~/nodejs$ vim Dockerfile
team7@ubuntu:~/nodejs$ docker build -t node-server .
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post "http://x2fvarx2frunx2fdocker.sock/v1.24/build?buildargs=x7b8x7d&cachefrom=x5b8x5d&cgroupparent=x4kpuperiod=x4cpouuoia=x4cpusetcpus=x4cpusetmem=x4cpushares=x4dockerfile=Dockerfile&labels=x7b8x7d&memory=x4&networkmode=defaultrn=i4shmsize=x4&target=node-server&targetplatforms=null&version=1": dial
unix /var/run/docker.sock: connect: permission denied
team7@ubuntu:~/nodejs$ sudo docker build -t node-server .
Sending build context to Docker daemon 1.999MB
Step 1/7 : FROM node:13
13: Pulling from library/node
1c6172af85ee: Pull complete
b194b9ec928: Pull complete
1f5ec08f35d5: Pull complete
93b1353672b6: Pull complete
3d7f38db3cca: Pull complete
21e1a2f9fe89: Pull complete
d5431b24825a: Pull complete
f780e3352c18: Pull complete
4d28937582d0: Pull complete
Digest: sha256:70d4ffca39a1f9f7161d58e674ddcc56c7f0724196b08d52a87bab15cb4a04
Status: Downloaded newer image for node:13
--> 2b9604a36e49
Step 2/7 : WORKDIR /app
--> Running in 62f1d14d0b17
Removing intermediate container 62f1d14d0b17
--> fc9a5f212298
Step 3/7 : COPY package.json /app
--> 6c4baec2f5ae
Step 4/7 : RUN npm install
--> Running in c8a0fcc90dd
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN nodejs@11.0.0 No repository field.

added 50 packages from 37 contributors and audited 50 packages in 2.645s
found 0 vulnerabilities

Removing intermediate container c8a0fcc90dd
--> 92544025f080
Step 5/7 : COPY . /app
--> 5b1d557d08f1
```

The Docker build command is used to create an image with instructions given by Docker-file. -t flag is used to tag the images with our node-server name.

## Step 5: Create And Run The Container

we'll then run the container to ensure it works as intended.

```
Removing intermediate container b3268e8dca89
---> 7828442b2e97
Step 7/7 : EXPOSE 3000
---> Running in ee7b2793265d
Removing intermediate container ee7b2793265d
---> e433acc4959
Successfully built e433acc4959
Successfully tagged node-server:latest
team7@ubuntu:~/nodejs$ sudo docker run -d --name nodejs -p 3000:3000 node-server
e8ea78a26ada5f213347af97a552e03412c23612c5b0d5ecf7519003540274c
team7@ubuntu:~/nodejs$ docker tag node-server avinashtechle/nodejs-starter
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/images/node-server/tag?repo=avinashtechle%2Fnodejs-starter&tag=latest": dial unix /var/run/docker.sock: connect: permission denied
team7@ubuntu:~/nodejs$ sudo docker tag node-server avinashtechle/nodejs-starter
tag does not exist: avinashtechle/nodejs-starter:1.1
The push refers to repository [docker.io/avinashtechle/nodejs-starter]
tag does not exist: avinashtechle/nodejs-starter:1.1
team7@ubuntu:~/nodejs$ sudo docker push avinashtechle/nodejs-starter:1.1
The push refers to repository [docker.io/avinashtechle/nodejs-starter]
tag does not exist: avinashtechle/nodejs-starter:1.1
team7@ubuntu:~/nodejs$ sudo docker push avinashtechle/nodejs-starter
Using default tag: latest
The push refers to repository [docker.io/avinashtechle/nodejs-starter]
52c02907d233: Preparing
a9a43abfee94: Preparing
8e427425e15a: Preparing
dc0c93a4304c: Preparing
ed09928f5a32: Preparing
ee50c22fd6c: Waiting
d8183b2c9c73: Waiting
5ae01ea9a0f: Waiting
05f4935ad90a: Waiting
c96f2308ab16: Waiting
38c2f9ead02d: Waiting
0dabcc98eeef: Waiting
6885f9305c0a: Waiting
denied: requested access to the resource is denied
team7@ubuntu:~/nodejs$ sudo docker tag node-server avinashtechle/nodejs-starter
tag does not exist: avinashtechle/nodejs-starter:1.1
The push refers to repository [docker.io/avinashtechle/nodejs-starter]
tag does not exist: avinashtechle/nodejs-starter:1.1
team7@ubuntu:~/nodejs$ sudo docker tag node-server avinashtechle/nodejs-starter
Using default tag: latest
The push refers to repository [docker.io/avinashtechle/nodejs-starter]
52c02907d233: Preparing
a9a43abfee94: Preparing
8e427425e15a: Preparing
```

# Step 6: Upload The Image To Docker Registry Docker Hub

- The image registry that we’re using is Docker Hub. First, your account has to be created, then create a repository with any name.

```
6885f9305c0a: Waiting
denied: requested access to the resource is denied
team7@ubuntu:~/nodejs$ sudo docker tag node-server avinashtechle/nodejs-starter
tag does not exist: avinashtechle/nodejs-starter:1.1
The push refers to repository [docker.io/avinashtechle/nodejs-starter]
tag does not exist: avinashtechle/nodejs-starter:1.1
team7@ubuntu:~/nodejs$ sudo docker tag node-server avinashtechle/nodejs-starter
Using default tag: latest
The push refers to repository [docker.io/avinashtechle/nodejs-starter]
52c02907d233: Preparing
a9a43abfee94: Preparing
8e427425e15a: Preparing
dc0c93a4304c: Preparing
ed09928f5a32: Preparing
ee50c22fd6c: Waiting
d8183b2c9c73: Waiting
5ae01ea9a0f: Waiting
05f4935ad90a: Waiting
c96f2308ab16: Waiting
38c2f9ead02d: Waiting
0dabcc98eeef: Waiting
6885f9305c0a: Waiting
denied: requested access to the resource is denied
team7@ubuntu:~/nodejs$ docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create one.
Username: avinashtechle
Password:
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/auth": dial unix /var/run/docker.sock: connect: per
mission denied
team7@ubuntu:~/nodejs$ sudo docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create one.
Username: avinashtechle
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
team7@ubuntu:~/nodejs$ sudo docker push avinashtechle/nodejs-starter
Using default tag: latest
The push refers to repository [docker.io/avinashtechle/nodejs-starter]
52c02907d233: Pushed
a9a43abfee94: Pushed
8e427425e15a: Pushed
dc0c93a4304c: Pushed
ed09928f5a32: Mounted from library/node
ee50c22fd6c: Mounted from library/node
```

we’ve pushed our docker image to the registry by using a docker push

## Step 7: Start The Kubernetes Cluster

We're are doing this lab on Minikube (used to run Kubernetes locally):

```
Step-8.png
Activities Terminal
Sep 20 04:40
team7@ubuntu: ~/nodejs

Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create one.
Username: avinashtechlie
Password:
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/auth": permission denied
team7@ubuntu:~/nodejs$ sudo docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create one.
Username: avinashtechlie
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
team7@ubuntu:~/nodejs$ sudo docker push avinashtechlie/nodejs-starter
Using default tag: latest
The push refers to repository [docker.io/avinashtechlie/nodejs-starter]
52c62907d233: Pushed
a9a43abf9ee94: Pushed
0a427425e15a: Pushed
dc0c93a304e: Pushed
ed00928f5a32: Mounted from library/node
ee50c22fd6c: Mounted from library/node
d8183b2c9c73: Mounted from library/node
5aea01ea0a0f: Mounted from library/node
05f4935ad90a: Mounted from library/node
c96f2308ab16: Mounted from library/node
38c2f9ead82d: Mounted from library/node
0dabcc98eeef: Mounted from library/node
6885f9305c0a: Mounted from library/node
latest: digest: sha256:eea96a77d12fed244c6fd24a793887aa6800f460687f505387fe16776221e396 size: 3050
team7@ubuntu:~/nodejs$ minikube start
minikube v1.23.1 on Ubuntu 20.10
Unable to pick a default driver. Here is what was considered, in preference order:
  docker: Not healthy: 'docker version --format {{.Server.Os}}-{{.Server.Version}}' exit status 1: Got permission denied while trying to connect to the
sock: Get "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/version": dial unix /var/run/docker.sock: connect: permission denied
  docker: Suggestion: Add your user to the 'docker' group: 'sudo usermod -aG docker $USER && newgrp docker' <https://docs.docker.com/engine/install/linux
  kvm2: Not installed: exec: "vtrsh": executable file not found in $PATH
  podman: Not installed: exec: "podman": executable file not found in $PATH
  vmware: Not installed: exec: "docker-machine-driver-vmware": executable file not found in $PATH
  virtualbox: Not installed: unable to find VBoxManage in $PATH
```

## Step 8: Define YAML File To Create A Deployment In Kubernetes Cluster

```
Activities Terminal Sep 20 04:40
team@ubuntu:~/nodejs
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create one.
Username: avlnashtechle
Password:
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/auth": dial unix /var/run/docker.sock: connect: permission denied
team@ubuntu:~/nodejs$ sudo docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create one.
Username: avlnashtechle
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
team@ubuntu:~/nodejs$ sudo docker push avlnashtechle/nodejs-starter
Using default tag: latest
The push refers to repository [docker.io/avlnashtechle/nodejs-starter]
52c62907d233: Pushed
a9a3ab7ee04: Pushed
8e427425e15a: Pushed
dc0c93a4304c: Pushed
ed0992f35a2c: Mounted from library/node
ee58c22f6dc6: Mounted from library/node
d8183b2c9c73: Mounted from library/node
5ae0a0ea0a0f: Mounted from library/node
05f4935a0d9a: Mounted from library/node
c96f2308ab10: Mounted from library/node
3bc2f9ead82d: Mounted from library/node
0dabc98eeef: Mounted from library/node
d085f9305c8a: Mounted from library/node
latest: digest: sha256:ee9a0a77d12fed244c0f424a793887aa0800f460687f505387f1e1077622e396 size: 3050
team@ubuntu:~/nodejs$ minikube start
minikube v1.23.1 on ubuntu 20.10
Unable to pick a default driver. Here is what was considered, in preference order:
  docker: Not healthy: "docker version -format {{.Server.Os}}-{{.Server.Version}}" exit status 1: Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/version": dial unix /var/run/docker.sock: connect: permission denied
  docker: Suggestion: Add your user to the "docker" group: 'sudo usermod -aG docker $USER && newgrp docker' <https://docs.docker.com/engine/install/linux-postinstall/>
  kvm2: Not installed: exec: "vishm": executable file not found in $PATH
  podman: Not installed: exec: "podman": executable file not found in $PATH
  vmware: Not installed: exec: "docker-machine-driver-vmware": executable file not found in $PATH
  virtualbox: Not installed: unable to find VBoxManage in $PATH

Exiting due to DRV_NOT_HEALTHY: Found driver(s) but none were healthy. See above for suggestions how to fix installed drivers.

team@ubuntu:~/nodejs$ minikube deploy
[3]: Stopped
vln deploy.yaml
```

## Step 9: Create Deployment In Kubernetes Cluster

As we've created the YAML file, we can go ahead and create a deployment from this YAML file.

```
Activities Terminal Sep 20 04:51 team7@ubuntu:~/nodejs
■ docker: Suggestion: Add your user to the 'docker' group: 'sudo usermod -aG docker $USER && newgrp docker' <https://docs.docker.com/engine/install/linux-postinstall/>
■ kvm2: Not installed: exec: "virsh": executable file not found in $PATH
■ podman: Not installed: exec: "podman": executable file not found in $PATH
■ vmware: Not installed: exec: "docker-machine-driver-vmware": executable file not found in $PATH
■ virtualbox: Not installed: unable to find VBoxManage in $PATH

✗ Exiting due to DRV_NOT_HEALTHY: Found driver(s) but none were healthy. See above for suggestions how to fix installed drivers.

team7@ubuntu:~/nodejs$ vim deploy.yaml
[3]~ Stopped vim deploy.yaml
team7@ubuntu:~/nodejs$ vim deploy.yaml
team7@ubuntu:~/nodejs$ kubectl create -f deploy.yaml
The connection to the server localhost:8080 was refused - did you specify the right host or port?
team7@ubuntu:~/nodejs$ kubectl create -f deploy.yaml
The connection to the server localhost:8080 was refused - did you specify the right host or port?
team7@ubuntu:~/nodejs$ minikube start
minikube v1.23.1 on Ubuntu 20.10
Unable to pick a default driver. Here is what was considered, in preference order:
■ docker: Not healthy: "docker version --format {{.Server.Os}}-{{.Server.Version}}" exit status 1: Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://x2fvarx2fvarx2fdocker.sock/v1.24/version": dial unix /var/run/docker.sock: connect: permission denied
■ docker: Suggestion: Add your user to the 'docker' group: 'sudo usermod -aG docker $USER && newgrp docker' <https://docs.docker.com/engine/install/linux-postinstall/>
■ kvm2: Not installed: exec: "virsh": executable file not found in $PATH
■ podman: Not installed: exec: "podman": executable file not found in $PATH
■ vmware: Not installed: exec: "docker-machine-driver-vmware": executable file not found in $PATH
■ virtualbox: Not installed: unable to find VBoxManage in $PATH

✗ Exiting due to DRV_NOT_HEALTHY: Found driver(s) but none were healthy. See above for suggestions how to fix installed drivers.

team7@ubuntu:~/nodejs$ sudo usermod -aG docker $USER && newgrp docker
team7@ubuntu:~/nodejs$ minikube start
minikube v1.23.1 on Ubuntu 20.10
Automatically selected the docker driver. Other choices: none, ssh
Starting control plane node minikube in cluster minikube
Pulling base image ...
Downloading Kubernetes v1.22.1 preload ...
> preload images k8s-v1.22.1: 511.84 MiB / 511.84 MiB 100.00% 6.39 MiB
> gc.io/k8s-minikube/sdimage: 0 B [ ] 7% / p/s in28s-
Creating docker container (CPUs=2, Memory=2200MB) ...
Preparing Kubernetes v1.22.1 on Docker 20.10.8 ...
■ Generating certificates and keys ...
■ Booting up control plane ...
■ Configuring RBAC rules ...
■ Verifying Kubernetes components...
■ Using image gc.io/k8s-minikube/storage-provisioner:v5
Enabled addons: storage-provisioner, default-storageclass
Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
team7@ubuntu:~/nodejs$
```

## Step 10: Expose the deployment to the internet we're going live through Kubernetes service object

```
Activities Terminal Sep 20 04:57 team7@ubuntu:~/nodejs
team7@ubuntu:~/nodejs$ kubectl get deploy,po
NAME                                READY    UP-TO-DATE    AVAILABLE    AGE
deployment.apps/nodejs-deployment  0/2      2              0            22s

NAME                                READY    STATUS      RESTARTS    AGE
pod/nodejs-deployment-75d89c4648-fvtvp  0/1      ContainerCreating  0            21s
pod/nodejs-deployment-75d89c4648-wm8ts  0/1      ContainerCreating  0            21s
team7@ubuntu:~/nodejs$ kubectl expose deployment nodejs-deployment --type=LoadBalancer
service/nodejs-deployment exposed
team7@ubuntu:~/nodejs$ kubectl get svc
NAME      TYPE        CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
kubernetes  ClusterIP   10.96.0.1      <none>         443/TCP    94s
nodejs-deployment  LoadBalancer  10.107.226.85  <pending>      3000:32292/TCP  12s
team7@ubuntu:~/nodejs$ kubectl apply -f https://raw.githubusercontent.com/google/metalb/v0.9.3/manifests/metalb.yaml
namespace/metalb-system created
team7@ubuntu:~/nodejs$ kubectl apply -f https://raw.githubusercontent.com/google/metalb/v0.9.3/manifests/metalb.yaml
Warning: policy/v1beta1 PodSecurityPolicy is deprecated in v1.21+, unavailable in v1.25+
podsecuritypolicy.policy/controller created
podsecuritypolicy.policy/speaker created
serviceaccount/controller created
serviceaccount/speaker created
clusterrole.rbac.authorization.k8s.io/metalb-system:controller created
clusterrole.rbac.authorization.k8s.io/metalb-system:speaker created
role.rbac.authorization.k8s.io/config-watcher created
role.rbac.authorization.k8s.io/pod-lister created
clusterrolebinding.rbac.authorization.k8s.io/metalb-system:controller created
clusterrolebinding.rbac.authorization.k8s.io/metalb-system:speaker created
rolebinding.rbac.authorization.k8s.io/config-watcher created
rolebinding.rbac.authorization.k8s.io/pod-lister created
Warning: spec.template.spec.nodeSelector[beta.kubernetes.io/os]: deprecated since v1.14; use "kubernetes.io/os" instead
daemonset.apps/speaker created
deployment.apps/controller created
team7@ubuntu:~/nodejs$ kubectl create secret generic -n metalb-system memberlist --from-literal=secretkey="$(openssl rand -base64 128)"
secret/memberlist created
team7@ubuntu:~/nodejs$ minikube ip
192.168.49.2
team7@ubuntu:~/nodejs$ vim configmap.yaml
team7@ubuntu:~/nodejs$ kubectl create -f configmap.yaml
configmap/config created
team7@ubuntu:~/nodejs$ kubectl delete svc nodejs-deployment
service "nodejs-deployment" deleted
team7@ubuntu:~/nodejs$ kubectl expose deployment nodejs-deployment --type=LoadBalancer
service/nodejs-deployment exposed
team7@ubuntu:~/nodejs$ kubectl get svc
NAME      TYPE        CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
kubernetes  ClusterIP   10.96.0.1      <none>         443/TCP    6m27s
nodejs-deployment  LoadBalancer  10.100.102.113  192.168.49.2  3000:31696/TCP  9s
team7@ubuntu:~/nodejs$
```

## Step 11: Using MetalLB In Your Minikube Environment

we're going live through Kubernetes service object

```
Activities Terminal Sep 20 04:57 team7@ubuntu: ~/nodejs

# Using image gcr.io/k8s-minikube/storage-provisioner:v5
# Enabled addons: storage-provisioner, default-storageclass
# Done! kubectll is now configured to use "minikube" cluster and "default" namespace by default
team7@ubuntu:~/nodejs$ kubectl create -f deploy.yaml
deployment.apps/nodejs-deployment created
team7@ubuntu:~/nodejs$ kubectl get deploy,po
NAME                                READY    UP-TO-DATE    AVAILABLE    AGE
deployment.apps/nodejs-deployment    0/2      2              0             22s

NAME                                READY    STATUS              RESTARTS    AGE
pod/nodejs-deployment-75d89c4d4-fv1wp 0/1      ContainerCreating    0            21s
pod/nodejs-deployment-75d89c4d4-wm7ts 0/1      ContainerCreating    0            21s
team7@ubuntu:~/nodejs$ kubectl expose deployment nodejs-deployment --type=LoadBalancer*
service/nodejs-deployment exposed
team7@ubuntu:~/nodejs$ kubectl get svc
NAME      TYPE        CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
kubernetes ClusterIP   10.96.0.1      <none>          443/TCP     94s
nodejs-deployment LoadBalancer 10.107.226.85 <pending>     3890:32202/TCP 12s
team7@ubuntu:~/nodejs$ kubectl apply -f https://raw.githubusercontent.com/google/metalb/v0.9.3/manifests/namespaces.yaml
namespace/metalb-system created
team7@ubuntu:~/nodejs$ kubectl apply -f https://raw.githubusercontent.com/google/metalb/v0.9.3/manifests/metalb.yaml
Warning: policy/v1beta1 PodSecurityPolicy is deprecated in v1.21+, unavailable in v1.25+
podsecuritypolicy.policy/controller created
podsecuritypolicy.policy/speaker created
serviceaccount/controller created
serviceaccount/speaker created
clusterrole.rbac.authorization.k8s.io/metalb-system:controller created
clusterrole.rbac.authorization.k8s.io/metalb-system:speaker created
role.rbac.authorization.k8s.io/config-watcher created
role.rbac.authorization.k8s.io/pod-lister created
clusterrolebinding.rbac.authorization.k8s.io/metalb-system:controller created
clusterrolebinding.rbac.authorization.k8s.io/metalb-system:speaker created
rolebinding.rbac.authorization.k8s.io/config-watcher created
rolebinding.rbac.authorization.k8s.io/pod-lister created
Warning: spec.template.spec.nodeSelector[beta.kubernetes.io/os]: deprecated since v1.14; use "kubernetes.io/os" instead
daemonset.apps/speaker created
deployment.apps/controller created
team7@ubuntu:~/nodejs$ kubectl create secret generic -n metalb-system memberlist --from-literal=secretkey="$(openssl rand -base64 128)"
secret/memberlist created
team7@ubuntu:~/nodejs$ minikube ip
192.168.49.2
team7@ubuntu:~/nodejs$ kubectl create -f configmap.yaml
configmap/config created
team7@ubuntu:~/nodejs$ kubectl delete svc nodejs-deployment
service "nodejs-deployment" deleted
team7@ubuntu:~/nodejs$ kubectl expose deployment nodejs-deployment --type=LoadBalancer*
service/nodejs-deployment exposed
```