Content:

1. [Docker](https://www.youtube.com/watch?v=pg19Z8LL06w&ab_channel=TechWorldwithNana)
2. [Kubernetes](https://www.youtube.com/watch?v=s_o8dwzRlu4&ab_channel=TechWorldwithNana)
3. Aws (common tools like ec2, db, s3, logger etc)
4. [Jenkins](https://www.youtube.com/watch?v=6YZvp2GwT0A&ab_channel=DevOpsJourney)
5. Think like a devops engineer (how it helps, why better than alternatives like VMs, optimisation, automation, scalling)

Docker:

1st 17 min is about why. Will watch later, for now interested in practical

Download the docker desktop

Commands:

docker images – list of images

docker ps – list of running containers

docker pull {name}:{version} - If version is not mentioned it will assume latest

docker run –d –p {HOST\_PORT}:{CONTAINER\_PORT} {name}:{version}

-d – runs in background. Without that will block the terminal with logs and would have to open another terminal for other commands

-p – to port to specific port on local

For port binding:

CONTAINER\_PORT – check in docker ps

HOST\_PORT – any I like eg 8000

I can use --name to specify too

Note: pull is optional, with run it will download it automatically

docker stop {id} - to stop. Id can be found in docker ps

docker start {id} - as docker run creates a new container every time.

[Docker registry](https://hub.docker.com/search?image_filter=official) – official images.

Docker hub:

Make a nodejs project with

Dockerfile for nodejs project:  
FROM node:19-alpine

COPY package.json /app/

COPY src /app/

WORKDIR /app

RUN npm install

CMD ["node", "server.js"]

Command to build this:

docker build -t node-app:1.0 .

-t specifies type, . Is the current dir, node-app is the container name

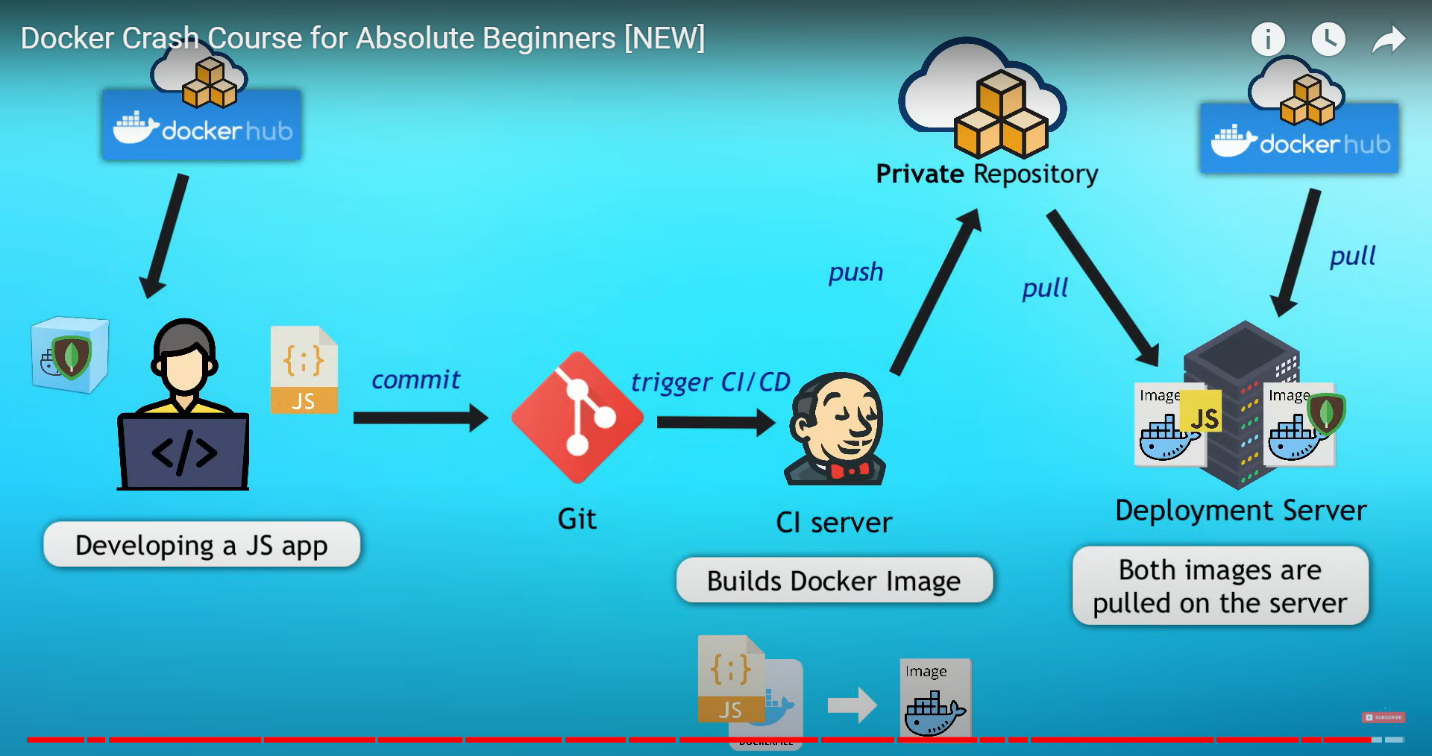
Use docker images, will see this in the list

To run: docker run –d –p 3000:3000 node-app:1.0

(3000 as we specified in nodejs code)

I can see the images in docker desktop too

Overall flow in software development:



**FROM node:19-alpine -** Specifies the base image as Node.js version 19 on Alpine Linux, a lightweight distribution.

**COPY package.json /app/ -** Copies the package.json file from the current directory on the host machine to the /app/ directory in the Docker image.

**WORKDIR /app -** Sets the working directory for subsequent instructions to /app/ within the Docker image.

**RUN npm install -** Executes npm install command to install dependencies specified in package.json into the Docker image.

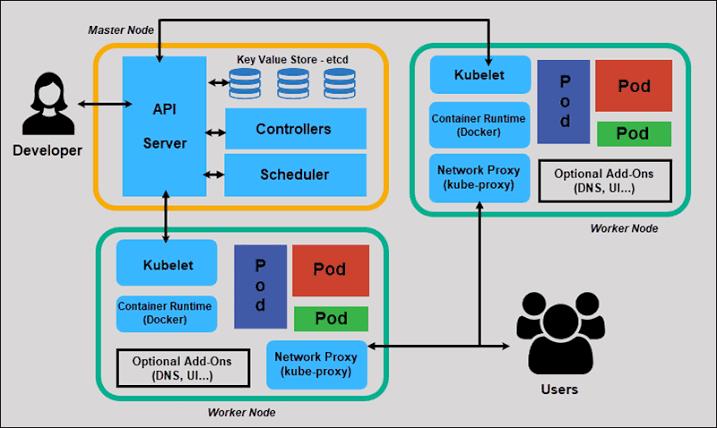
**CMD ["node", "server.js"] -** Specifies the command to run when the Docker container starts, which in this case is running node server.js to start the Node.js server.

TODO: Can dockerise lloyd’s code for easy installation. I think jenkins can do that work better.

Kubernetes

Manage containers like docker

1st 9 min is architecture



Pod – smallest unit

Services – communicate between pods (eg: nodejs pod communicating with mongo pod)

ingress – help to get a static endpoint address (eg: in ngrok new endpoint is generated which is bad for prod)

Config map – external config like original db url

Secret – it’s like env

Volumes – to add persistence storage for pod

In deployment, better to have a backup replica

Deploy – to create pods, replicas. Acts as load balancer too.

Statefulset - The replicas need to access single db

Etcd - Kubernetes sees what’s the status of the container. If it’s not in the desired config, it tries to achieve that.

Minikube - to test locally

Kubectl – cli. Can use api or app too.

[Download minikube](https://minikube.sigs.k8s.io/docs/start/?arch=%2Fwindows%2Fx86-64%2Fstable%2F.exe+download) – can start as container or VM

Commands:

(1st start docker using the app)

minikube start --driver docker :- run as docker container

minikube status

kubectl get node :- lists the nodes

minikube for for starting and deleting. kubectl we use for config.

Demo project:

Demo:

Containers: Nodejs, MongoDB

He used the docs to create the file

Writing the .yaml file:

Metadata

Specification

See the written code

kubectl apply -f <yaml files>

Now try kubectl get pod

kubectl get all :- list all service, deployment, replicas

kubectl get secret

kubectl get configmap

kubectl describe service <service – c in list using get all)

kubectl describe pod <pod>

kubectl logs <pod – list using above)

Access the exposed service (nodejs here):

Kubectl get svc – to get cluster ip address

Minikube ip

Paste in browser: <aboveip>:<nodeport>

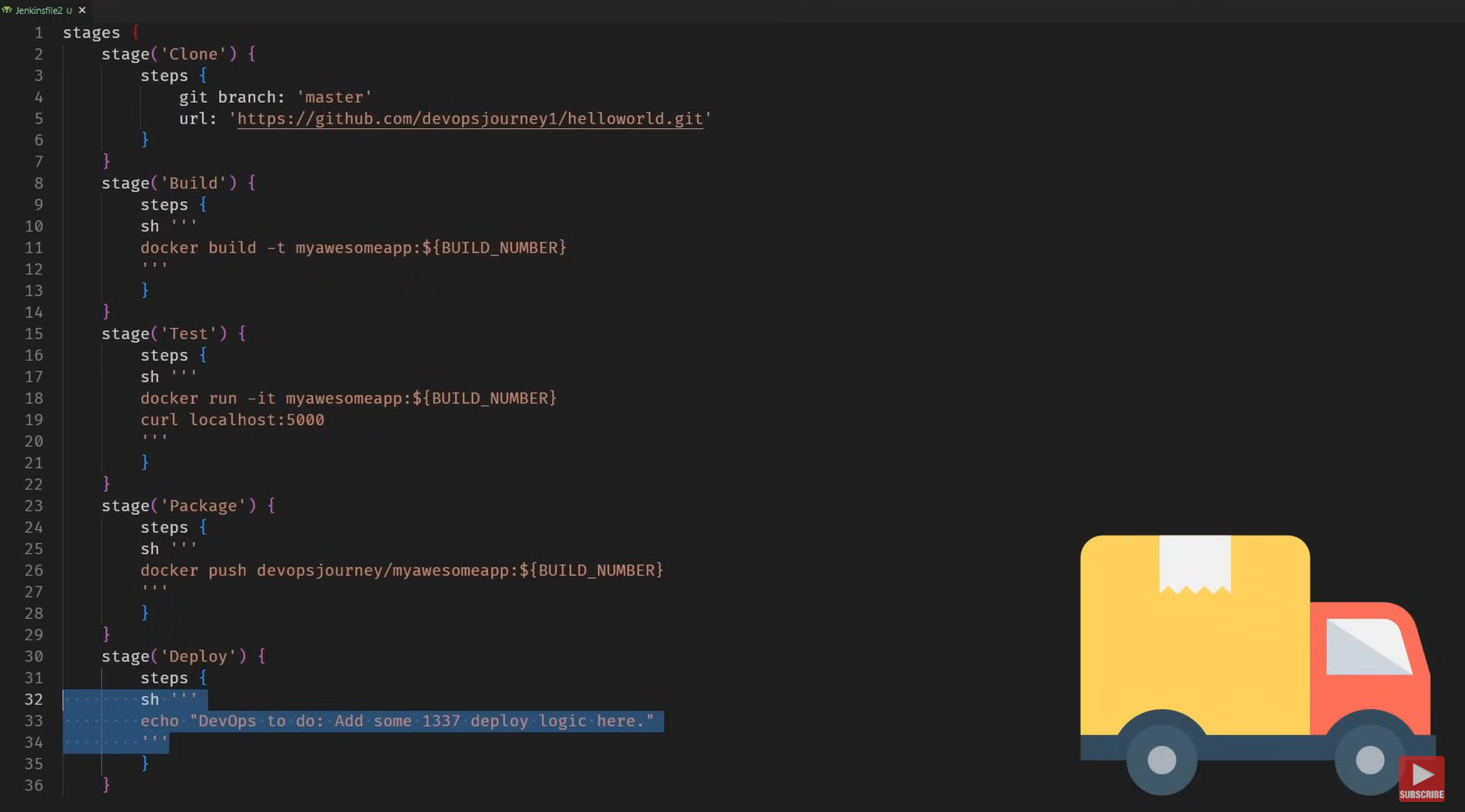
TODO: the site doesn’t open when I paste the above in browser

TODO: revise docker and kubernetes, learn how to import those env in nodejs.

Note: currently in not much mood of using kubernetes, I’ll re learn well latter. For now docker and jenkins look userful.

Jenkins

Automation software



For installation refer the following and runing:

[GitHub - devopsjourney1/jenkins-101](https://github.com/devopsjourney1/jenkins-101)

Few commands not mentioned in the above readme:

Docker network ls – lists created networks

TODO: try along with the video

Did some hello world stuff

He ran a python script from the GitHub repo!! That’s powerful!

Cloud (Docker)

I can run jobs after every few sec. All this can help me to build some interesting stuff.

(one thought, all this I can do with python too)