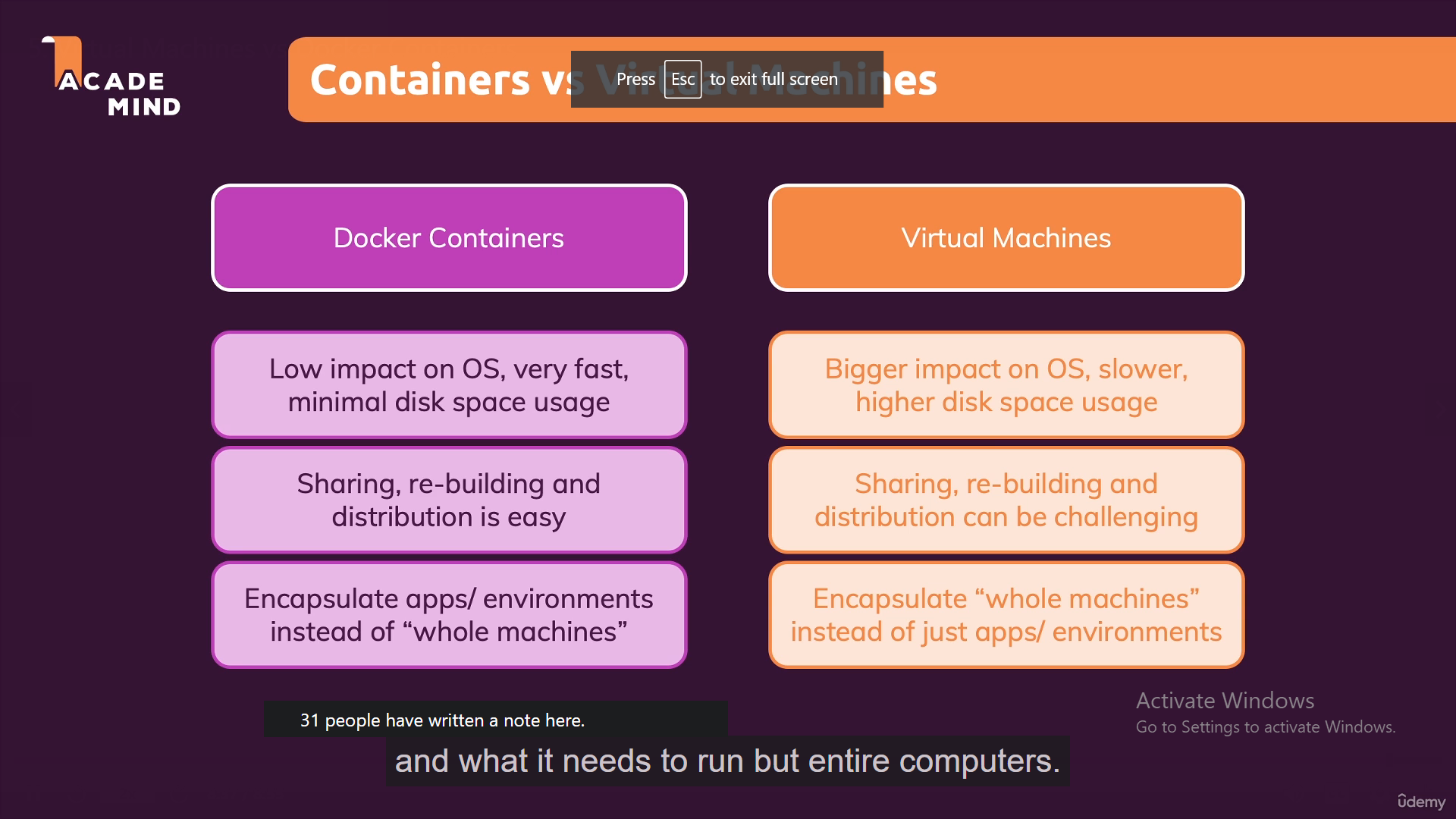
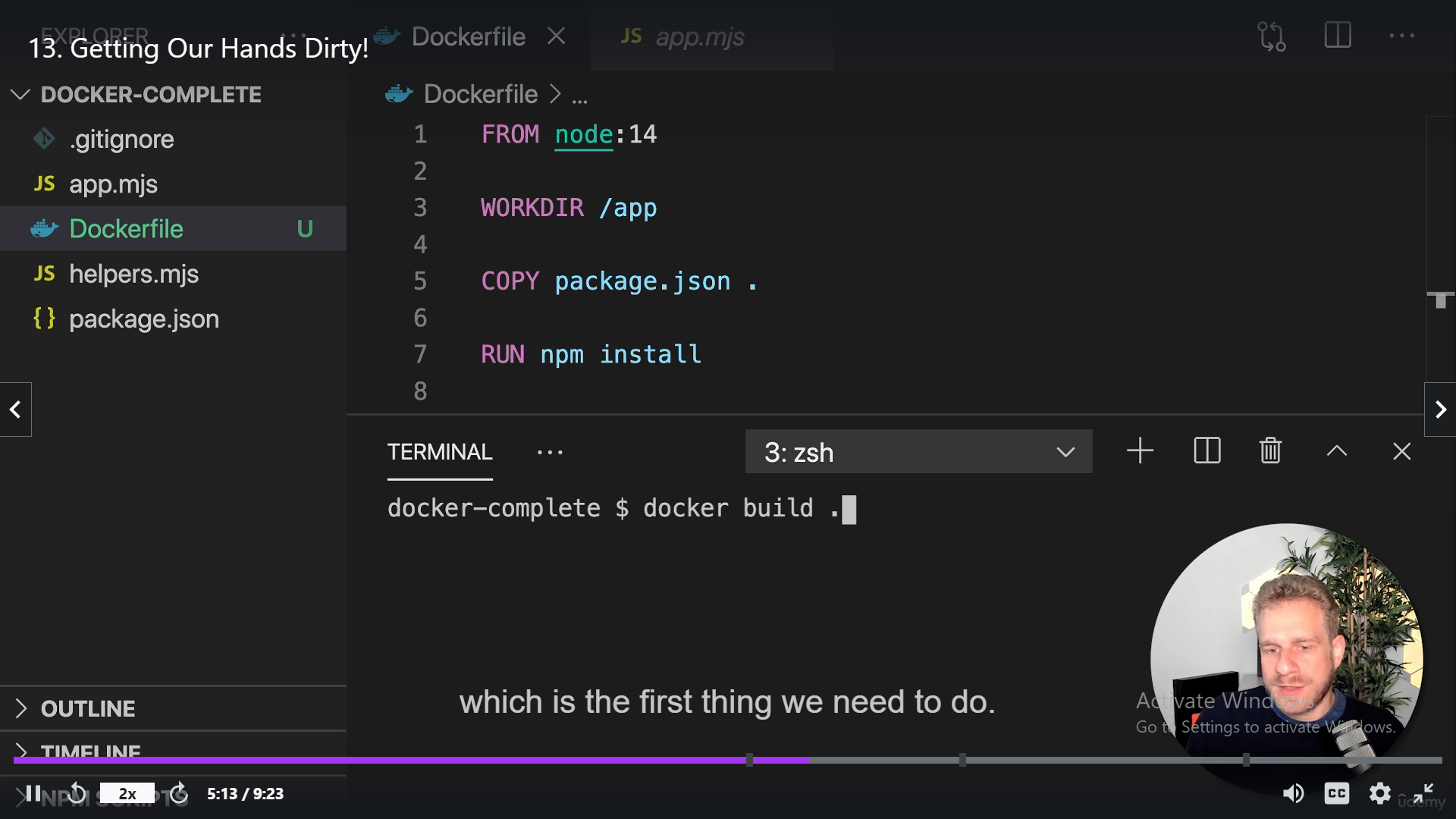
Docker is a Tool which provide container to ship the whole application with its environment setup to the server.



Make Docker File and write commands and ” run docker build . ”

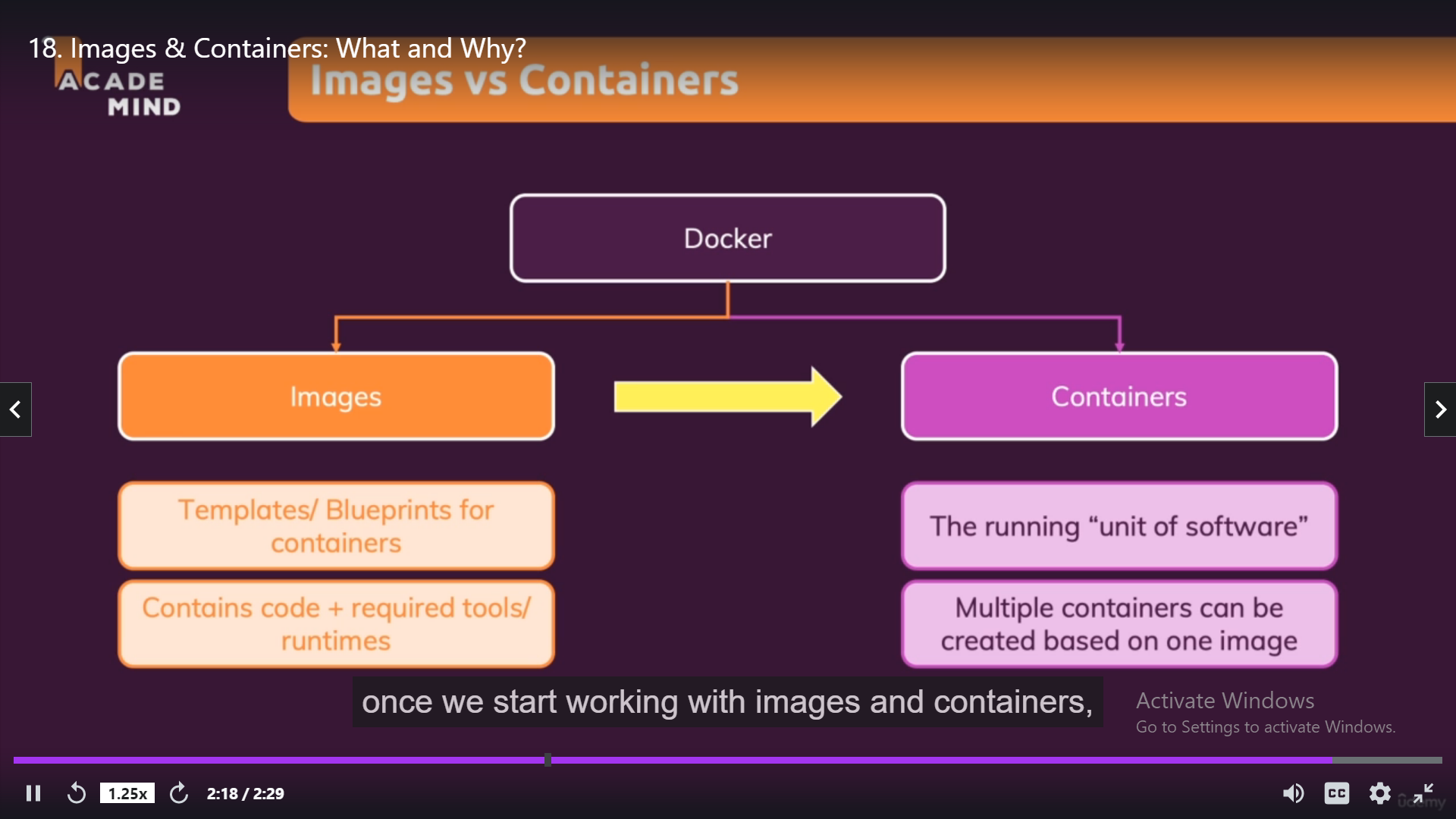


docker run -p 3000:3000 d2cc7b04fb0a

d2cc7b04fb0a is image ID

-p 3000:3000 it will make the port open in docker where our localhost will communicate.

Images are Blueprint of container which help to run multiple containers.



Docker build . -> command use to build new Image

Docker ps -a -> this will list all active containers.

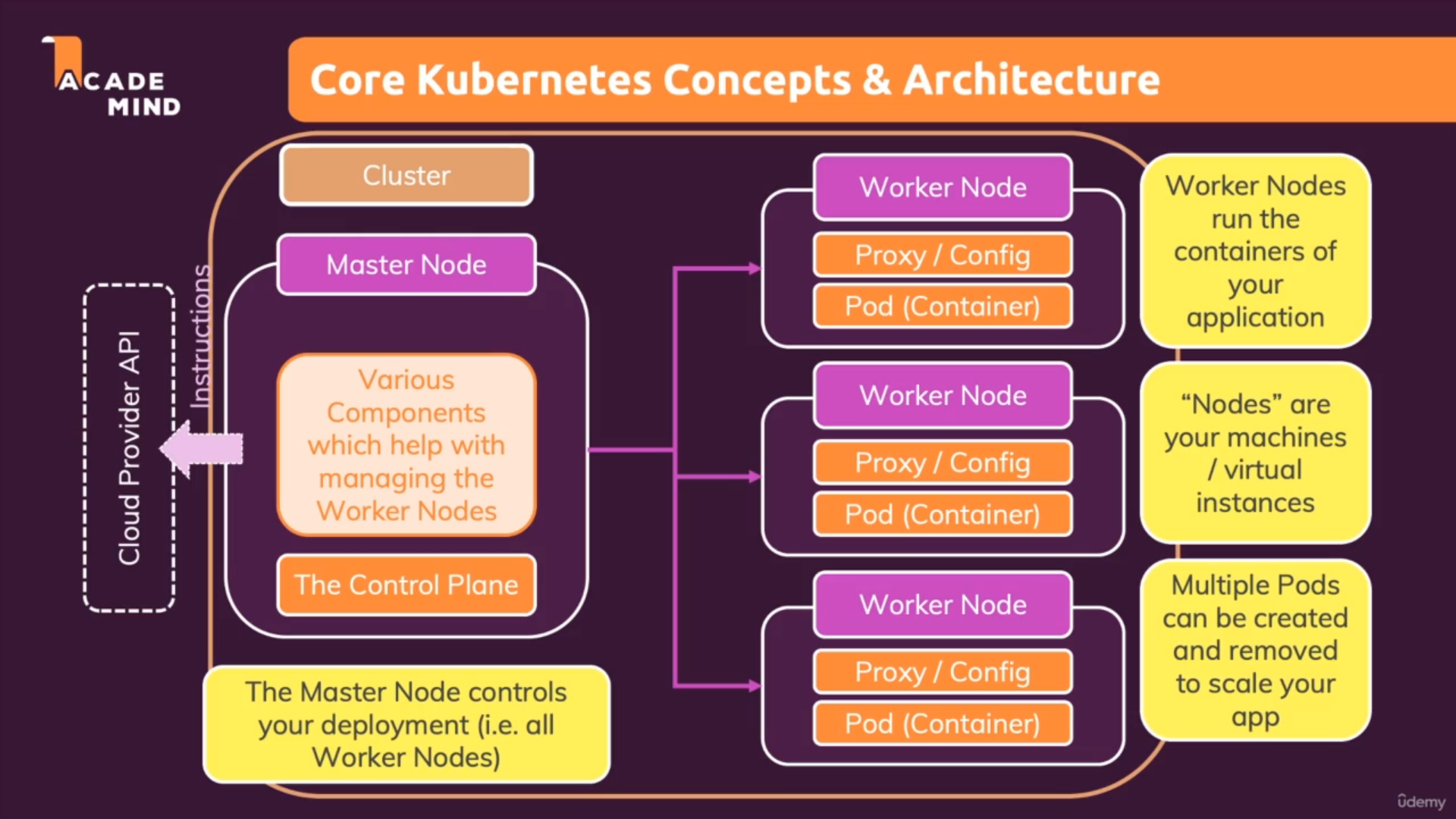
Docker start imageId -> will restart the container.

Docker start -a -I ImageId (interactive mode and attach Mode).

Docker Don’t only used for Servers also used for any language Interaction like python e.g enterning Input. Like Using Input().

Kubernetes: (it’s not alternative to Docker, it works with docker containers)

1. It just not a software that runs on machine, it is collection of concepts and tool.
2. It is used for automatic deployments containers.
3. When if some containers go down, then it help us to replace it with new one.
4. It helps us to increase number of containers when there is increase in traffic and distribute load.
5. Kubernetes is like Docker Compose for multiple machines with some deployment specific extra features for running your Dockerized application in multimachine setup.



Pod also can be called as container, is smallest unit of Kubernetes reside inside worker Node with some proxy configuration.

This worker nodes are controlled by master node, and this make 1 whole cluster and send to some Cloud Provider to replicate this Cluster in their cloud Environment.

Code Deployment Steps to Heroku.(build not require for backend code)

1. Need Heroku account
2. Install Heroku cli
3. Install git cli
4. Create new app in Heroku website
5. Connect Heroku app with local git repository using command : heroku git:remote -a appname.
6. Add node\_modules folder in git ignore, Heroku automatically install package.json dependencies and create node\_modules on the server.
7. Git add .
8. Git commit -m “deployment message”
9. Make sure your starting script in package.json should be node app.js
10. So if we have stored environment variable In nodemon then, we need to store that variables in Heroku config vars variables settings.
11. Git push Heroku master

Code Deployment step to Firebase: (npm run build require for frontend)

1. Install firebase cli
2. Firbase login
3. Firebase init
4. Choose project
5. Choose build folder directory instead of public
6. Configure single page App -> yes
7. Overwrite Index.html -> No
8. Firebase deploy