Enterprise Web Dev

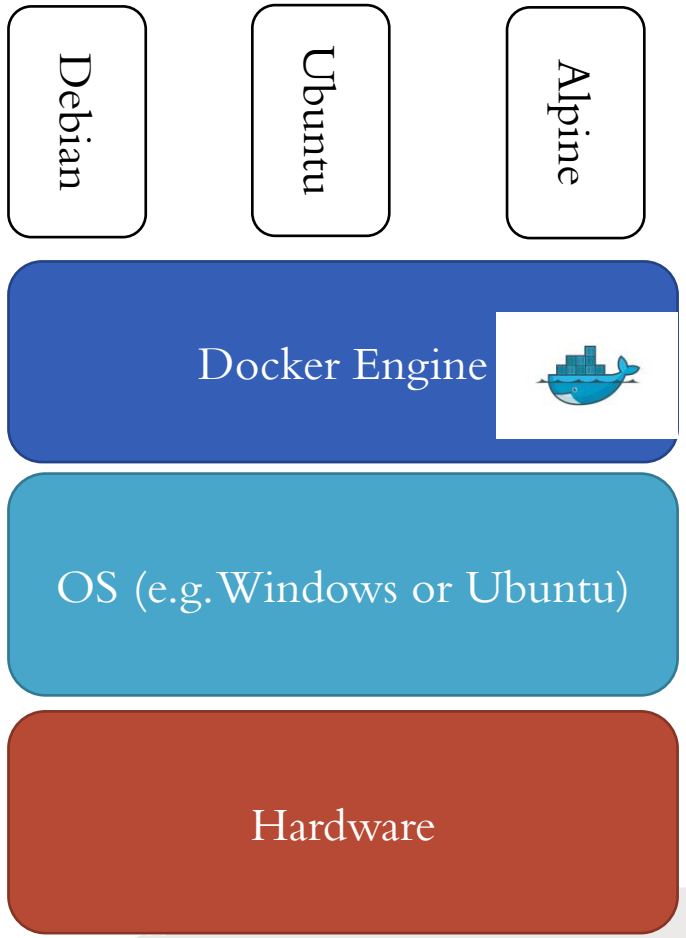
WEB APP DEV USING CONTAINERS

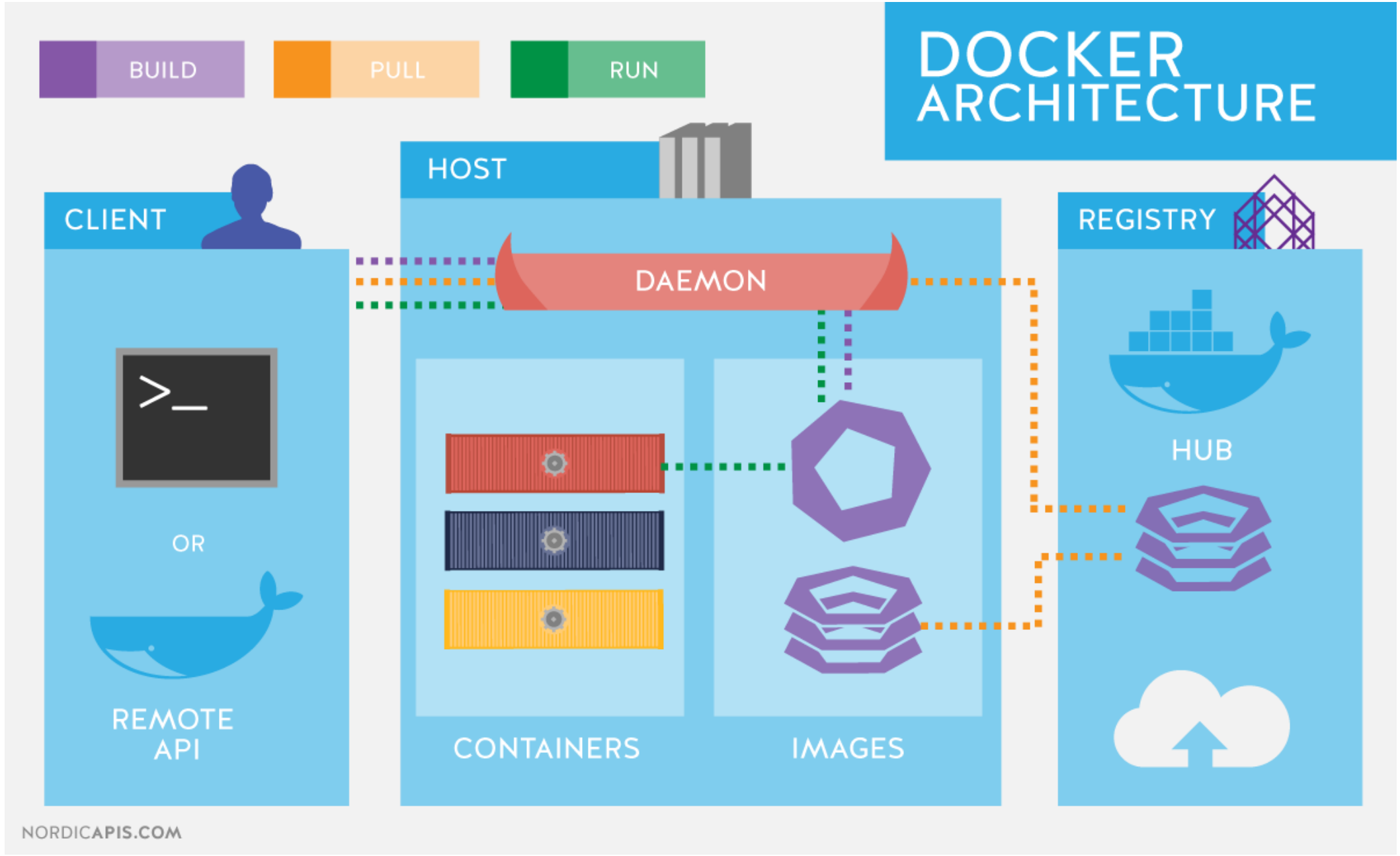
AGENDA(WHAT/WHY/HOW OF DEV CONTAINERS)

- What is Docker and Containers
- Why use containers
- What is a Dev Containers
- Why use Dev Containers for App Development
- What is GitHub CodeSpaces
- Why use Github Codespaces
- How to set up a Dev Container for Node.js
- How to set up a Github Codespace for Node.js

WHAT IS DOCKER AND CONTAINERS

- A **container** is lightweight & portable environment for developing, shipping, and running applications.
- **Docker** is a popular platform that enables developers to create, deploy, and run applications in isolated containers.
- Main components of Docker platform are:
  - Docker Daemon/Engine
  - Docker Client
  - Docker Containers
  - Docker Images
  - Docker Registry



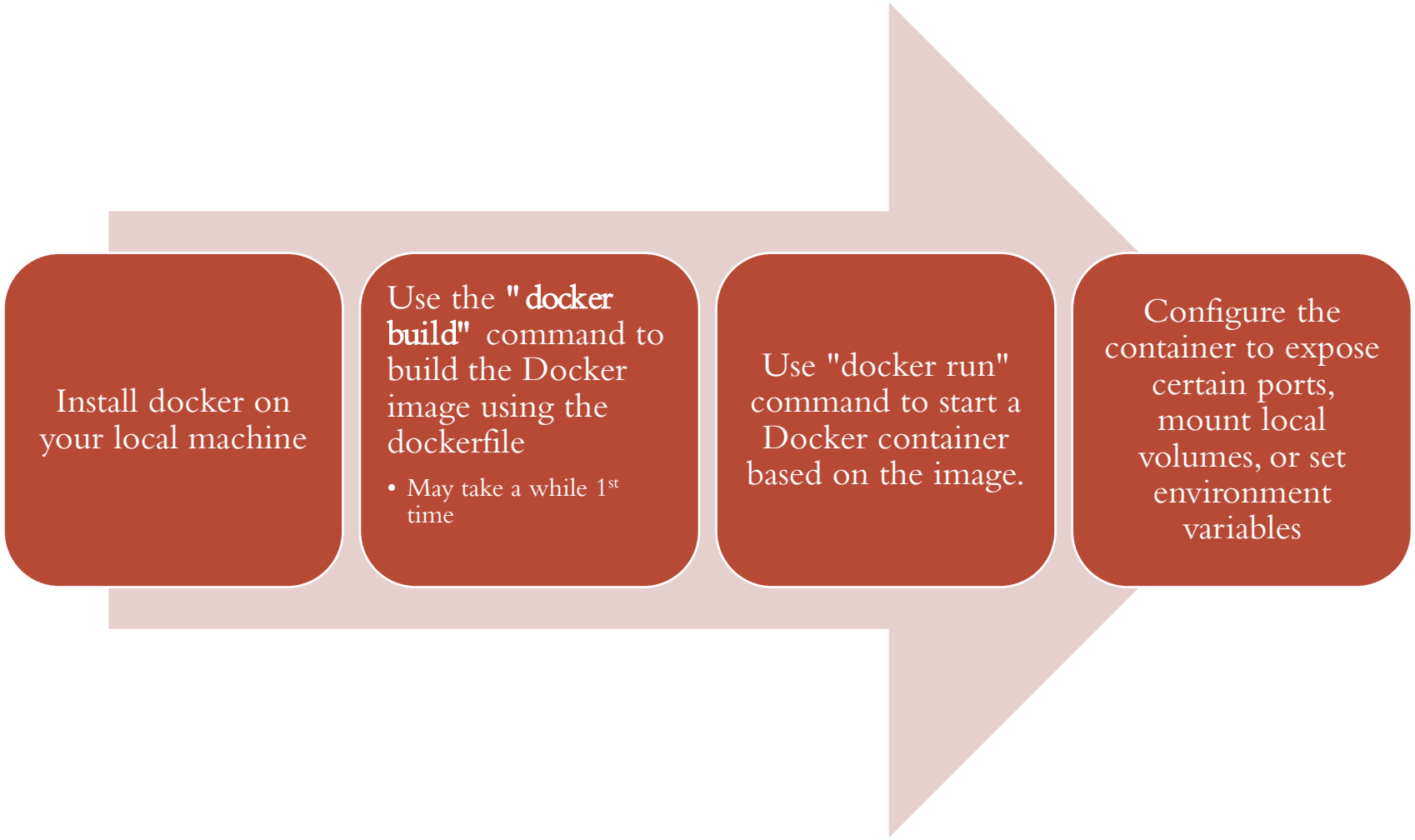


DOCKER FILE

WHY USE CONTAINERS

- A consistent environment for development, testing, etc which reduces the risk of issues due to differences between environments.
- Concurrently develop different versions of an application and/or its dependencies without affecting the underlying system.
- Containers can be spun up and down quickly as needed.
- Efficient use of resources.
  You can run multiple containers on a single machine, reducing the need for additional hardware.
- Containers can be scaled up or down easily
- Popular technology for web application development.
  Easy to Share/deploy Applications.
  (I should have no problem running/testing your labs and API!)

HOW TO USE CONTAINERS(DOCKER)



WHAT ARE DEV CONTAINERS

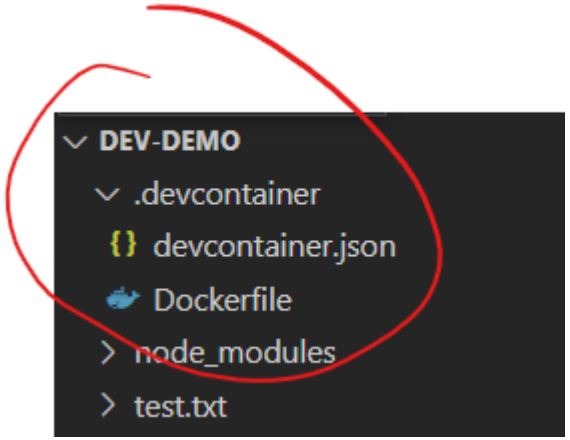
- Dev containers are preconfigured containers for App Development and Testing
- Designed to work well with VS Code and Github
- They include tools/libraries/dependencies specifically tailored for a project
- For Node.js Development:

  https://github.com/nodejs/devcontainer

  This container has Node.js, JavaScript, and Typescript. It also includes a common set of tools, such as nvm, npm, yarn, git, wget, rsync, openssh, and nano.
  Also, you can easily build and connect to it using VS Code.

USING A DEV CONTAINER(LOCAL HOST)

- **Install Docker on Local Host**
  1. Create a folder for your app on your local host (e.g. my-app)
  2. Create a .devcontainer folder in your my-app folder
  3. Create a "Dockerfile“ in the .devcontainer folder
  4. Create a "devcontainer.json" in the .devcontainer folder. This file will define the configuration for your dev container.
  5. Open your my-app folder in Visual Studio Code and install the “Dev Containers" extension
  6. Select “Open Remote Window” in bottom left corner. Select “Reopen in Container” option from Dev Containers list.



OR (much easier…)

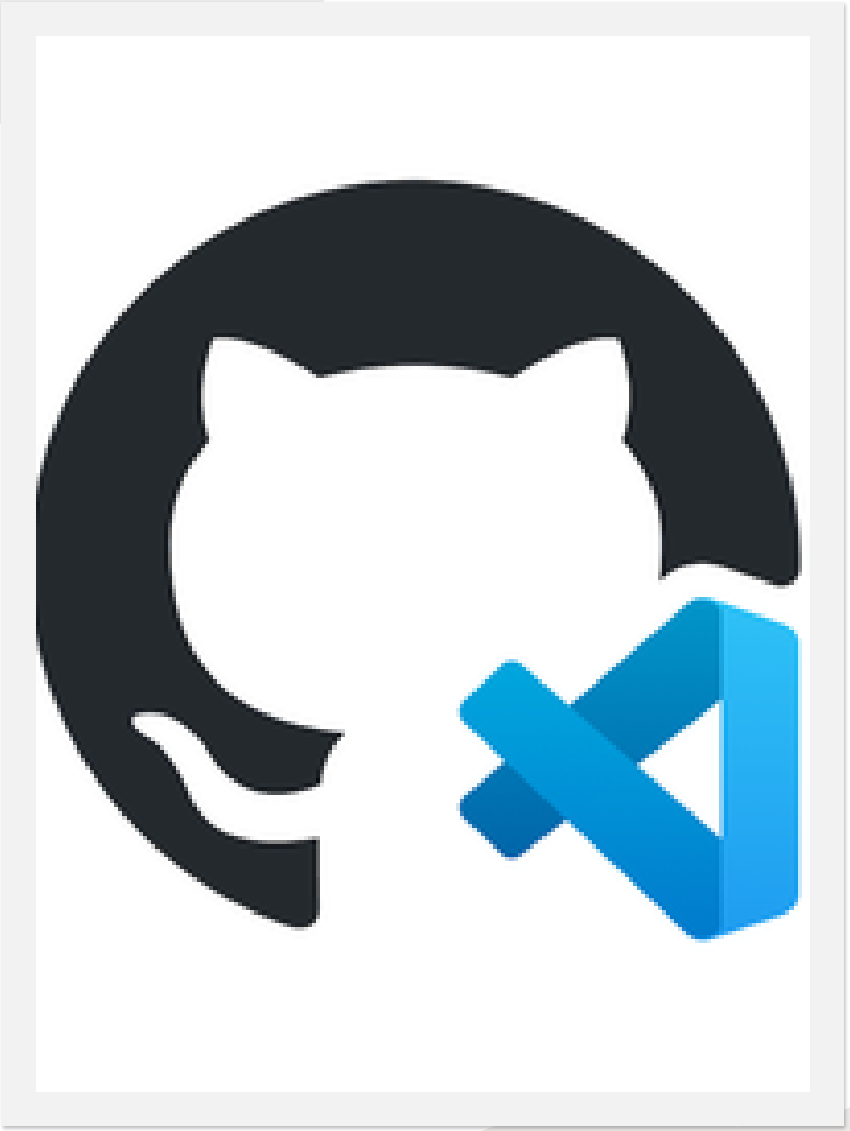
1. Just use VS Code to create a Dev Container for you…

WHY USE A DEV CONTAINER

- Consistency across development environments
- Isolation from the host machine
- Easy setup and configuration
- Ability to replicate the same environment for development, testing, and production

WHAT IS GITHUB CODESPACES

- GitHub Codespaces lets you create a cloud-based development environment that can be accessed from anywhere.
- GitHub Codespaces are built on top of dev containers
- All in the cloud!
- Subset of VS Code IDE in Browser if you want



WHY USE GITHUB CODESPACES

- Code on any device, anywhere. (e.g. new laptop?... get going with dev straight away. No local env set up.)
- Easy collaboration with others
- Auto synchronisation with GitHub repo…