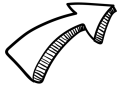
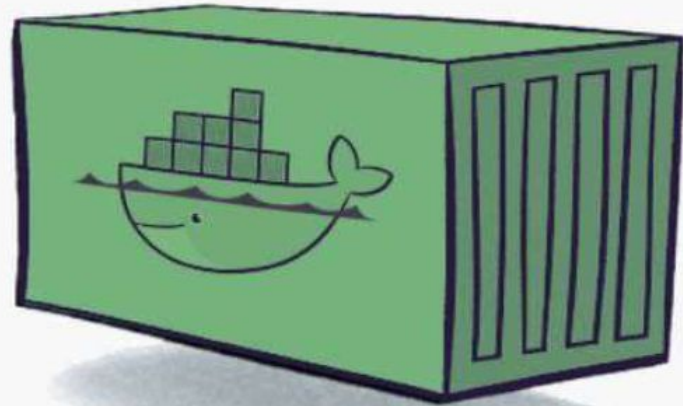


# Containerization Tools for Modern Web Development



Empowering Development and Deployment

- Rudra Pratap Singh
- B.Tech CSE Student @ IIT Mandi
- Google Summer of Code Contributor 2024 @ The Linux Foundation



# Introduction to Containerization



## What is Containerization?

### Development

Lets say You created  
an Application

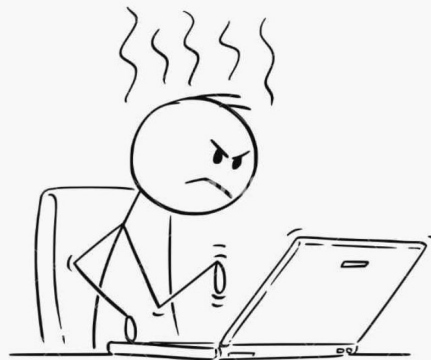
And that's working fine in  
your machine



### Production

But in Production it  
doesn't work properly

Developers experince it a lot



# What is Containerization?



That is when the Developer's famous words are spoken

Your application is not working

It works on my machine

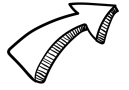


The Reason could be due to :

- Dependencies
- Libraries and versions
- Framework
- OS Level features
- Microservices

That the developers machine has but not there in the **production environment**

# What is Containerization?



- We need a standardized way to package the application with its dependencies and deploy it in any environment. That's where containers come into play.
- Containerization is a lightweight form of virtualization that involves encapsulating an application and its dependencies into a container. This container can run consistently across different computing environments.



# Importance in Modern Development

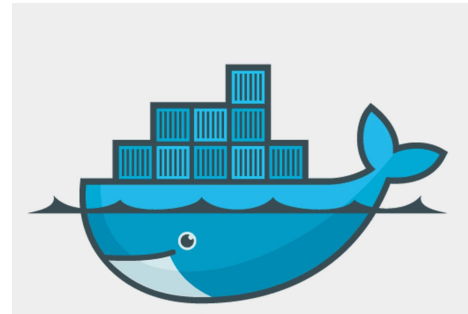


- **Consistency Across Environments:** Containers ensure that an application runs the same, regardless of where it is deployed, eliminating the "it works on my machine" problem.
- **Isolation:** Containers provide isolation, meaning that each container runs in its own environment without affecting others. This is crucial for maintaining stability and security.
- **Efficiency:** Containers are lightweight and share the host system's kernel, making them more efficient than traditional virtual machines.
- **Rapid Deployment:** Containers can be started quickly, allowing for rapid deployment and scaling of applications.

# Popular Containerization Tools



- **Docker:** Docker is an open-source platform that automates the deployment, scaling, and management of applications in containers.
- **Podman:** Podman is a daemonless container engine for developing, managing, and running OCI containers.
- **Rockcraft:** Rockcraft is a tool to create [rocks](#) – a new generation of secure, stable and [OCI-compliant container images](#), based on Ubuntu.



# Basic Docker Commands



You can find instructions to install docker on <https://docs.docker.com/engine/install/>

**Docker Hub:** Docker Hub is a place to store your OCI images. It is a public container registry where users can upload their images and also pull public images available there.

- Running a simple image  
`docker run nginx`

## What Happens When You Run `docker run nginx`

1. Pulling the Image: If the `nginx` image is not already present on your local machine, Docker will pull it from DockerHub (or another configured registry).
2. Creating a Container: Docker creates a new container from the `nginx` image.
3. Running the Container: Docker starts the container, which will execute the default command specified in the `nginx` image (usually, it starts the Nginx server).

```
rudra — docker run nginx — docker — com.docker.cli < docker run nginx —...
[~] ~ docker run nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
ea235d1ccf77: Pull complete
e29cef106877: Pull complete
e9bf20d5335e: Pull complete
1394e86b8f58: Pull complete
7b2b3e0f512f: Pull complete
6a11b5a77155: Pull complete
fb6d6e4aad9c: Pull complete
```

# Basic Docker Commands



## Port Mapping 🌐

Link a port on your machine to a port in the container.

```
docker run -p <host_port>:<container_port>
```

## Detached Mode 🕶️

Run containers in the background.

```
docker run -d <image>
```

## Check Running Containers 🕵️

See all active containers.

```
docker ps
```

## Stop a Container 🙌

Kill a running container.

```
docker kill <container_ID>
```

```
rudra — rudra@Rudras-Air — ~ — zsh — 80x24
[→ ~ docker run -d -p 8080:80 nginx
57623f2213563a62ecad5dce7ba0325bbaa413e7ea584c7d07453d3645fabac7
[→ ~ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS
PORTS         NAMES
57623f221356   nginx    "/docker-entrypoint..." 3 seconds ago   Up 3 seconds
0.0.0.0:8080->80/tcp   interesting_wiles
[→ ~ docker kill 57623f221356
57623f221356
→ ~
```



# Let's Get a Node Project Up and Running! 🚀



1. **Initialize a Node Project** 📦 Start with npm to initialize a Node project.

```
npm init -y
```

2. **Install Express Packages** 📦

```
npm install express
```

3. **Create server.js Project** 🚀 Write your server.js file.

Write your `server.js` file. Here's an example to get you started: [server.js](#).

4. **Serve an HTML File using Express** 🌐

Create a `public` folder and an `index.html` file inside it. Example you can use: [index.html](#)

5. **Start the Server** 🌐

```
node server.js
```



**Hello and Welcome to OOSC IIT Kanpur**

Welcome you all to amazing conference dedicated to Open Source

# Let's Containerize the Node Project with Rockcraft!



## 1. Initialize a Rockcraft Project Set up your Rockcraft project.

```
rockcraft init
```

[Rockcraft Documentation](#) 

For more details, visit the  
Rockcraft docs:

[Rockcraft Documentation](#)

```
rockcraft.yaml
rockcraft - rockcraft project. Documentation: https://canonical-rockcraft.readthedocs-hosted.com (rockcraft.json)
1 name: oosc-web
2 # see https://documentation.ubuntu.com/rockcraft/en/stable/explanation/bases/
3 # for more information about bases and using 'bare' bases for chiselled rocks
4 base: ubuntu@22.04 # the base environment for this rock
5 version: '0.1' # just for humans. Semantic versioning is recommended
6 summary: Single-line elevator pitch for your amazing rock # 79 char long summary
7 description: |
8     This is oosc-web's description. You have a paragraph or two to tell the
9     most important story about it. Keep it under 100 words though,
10    we live in tweetspace and your description wants to look good in the
11    container registries out there.
12 platforms: # the platforms this rock should be built on and run on
13     arm64:
14
15 parts:
16     my-part:
17         plugin: nil
18
```

# Containerize the Node Project



## Write part for node app

```
parts:  
  node-app:  
    plugin: npm  
    source: oosc-server/  
    npm-include-node: true  
    npm-node-version: '21.1.0'
```

## Write a pebble service to start node app

```
services:  
  start-server:  
    override: replace  
    command: node server.js  
    startup: enabled  
    working-dir: /lib/node_modules/oosc-server
```

Full rockcraft.yaml here: [github gist](#)

# Pack the Rock using Rockcraft



## Build the rock

```
rockcraft pack -v
```

## Run the rock using docker

### 1. Compile the rock into docker image

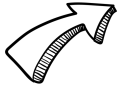
```
sudo rockcraft.skopeo --insecure-policy copy oci-archive:oosc-web_0.1_arm64.rock  
docker-daemon:oosc-web:latest
```

### 2. Let's start the docker container

```
docker run --name oosc-server -p 8080:8080 oosc-web:latest
```

Now explore <http://localhost:8080> on host machine to access node app

# Let's Investigate our Rock!



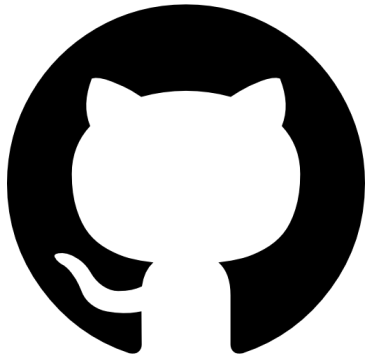
Let's spawn a terminal at prime stage and explore the folder structure

```
rockcraft prime --shell-after
```

PROBLEMS **6** OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```
rudra@rudra:/HostShared/oosc-web$ rockcraft prime --shell-after
Launching shell on build environment...
rockcraft-oosc-server-on-arm64-for-arm64-19009971 ../project# cd ../pr
prime/  project/
rockcraft-oosc-server-on-arm64-for-arm64-19009971 ../project# cd ../prime/
rockcraft-oosc-server-on-arm64-for-arm64-19009971 ../prime# ls
CHANGELOG.md LICENSE README.md bin include lib share var
rockcraft-oosc-server-on-arm64-for-arm64-19009971 ../prime# █
```

# Let's **Explore** an amazing **Rock!**



## **CUPS-ROCK**

Complete printing stack packaged using rockcraft.

Let's go through the codebase  
<https://github.com/rudra-iitm/cups-rock>

# Time to Say Goodbye

Thank you for your attention and participation!

## Let's Stay Connected:

- GitHub: <https://github.com/rudra-iitm>
- LinkedIn: <https://www.linkedin.com/in/rudra-iitm/>
- Twitter: [https://x.com/rudransh\\_rps](https://x.com/rudransh_rps)



## Explore Awesome Repositories:

- Projects: Visit my [GitHub profile](#) for more exciting projects and repositories.