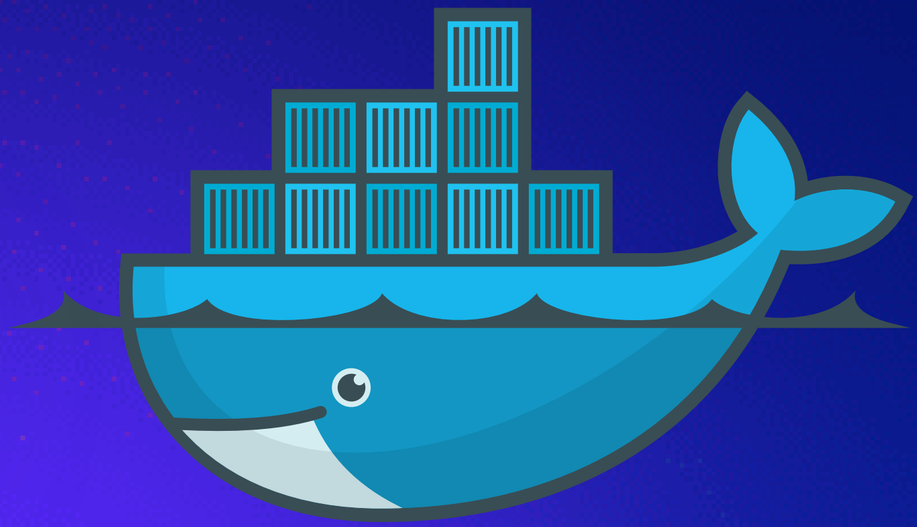
DOCKER INTERNALS

By Bikram Ghuku



INTRODUCTION

“It works on my computer”

We come across this problem many times that a software developed works on my computer and not on any others computer either due to lack of dependencies and/or different version of tools/ packages.

Docker

Docker is a tool that allows you to package your application into a unit called containers, making it easy to deploy and run applications reailably accross various computing environments. Its like virtualisation but more lighter and efficient.

DOCKER INTERNALS

Control Groups

- Control Groups are a linux kernel feature that allows the allocation of resources like CPU, Memory, Disk I/O, network.
- They allow for fine grain control and limits on resources usage for a group of processes.
- By using cgroups, container runtimes can isolate and manage the resource usage of containers, ensuring fair sharing and preventing any single container from monopolizing resources.

DOCKER INTERNALS

Namespace

- Namespaces are another feature provided by the Linux kernel that provides process isolation.
- Each namespace provides a unique instance of a particular system resource to a group of processes.
- By utilizing namespaces, container runtimes can create isolated environments that appear to be their own independent systems, complete with their own processes, network stack, filesystem, and more.

DOCKER ECOSYSTEM

PRESENTATION

DOCKER ECOSYSTEM

CONTAINERD

- containerd is an industry-standard core container runtime.
- It handles container lifecycle management, including container creation, execution, supervision, and cleanup.
- It's designed to be lightweight and efficient, providing the essential functionality needed to run containers reliably.

DOCKER ECOSYSTEM

RUNC

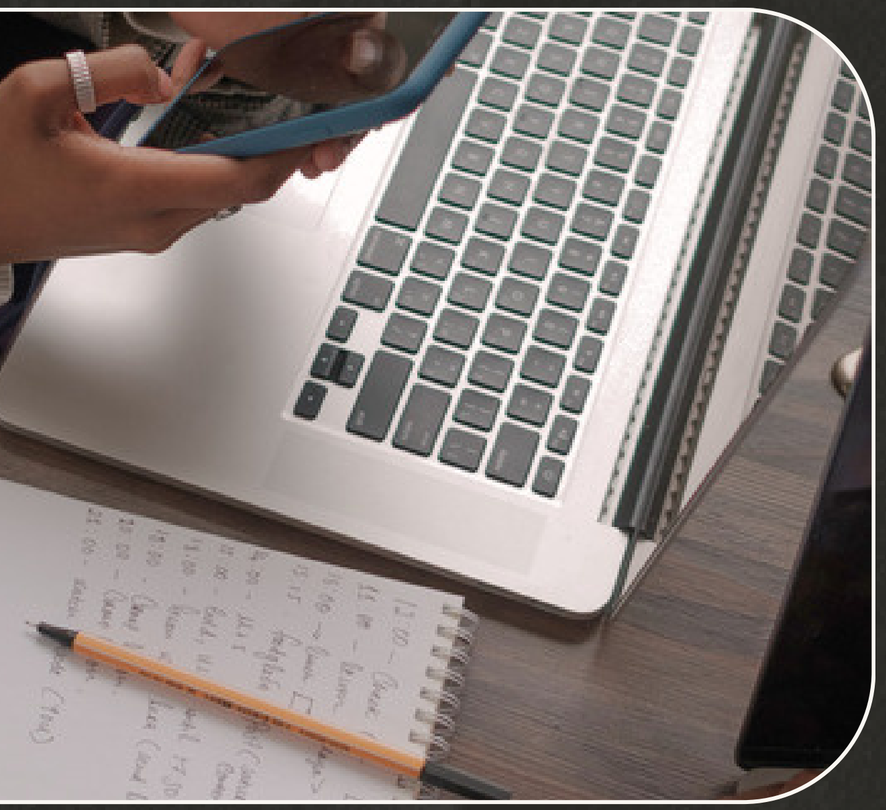
- runc is a lightweight, portable container runtime that implements the Open Container Initiative (OCI) specification.
- It's responsible for launching and managing containers according to OCI standards.
- runc is used by containerd as the default container execution driver.

DOCKER ECOSYSTEM

DOCKER ENGINE

- Docker Engine is the core Docker platform that provides an end-to-end containerization solution.
- It includes the Docker daemon (dockerd), a REST API for interacting with containers, and the Docker CLI (docker) for managing containers and images.
- Docker Engine uses containerd and runc internally for container management and execution.

DOCKER INTERNALS



DOCKER COMPOSE

- Docker Compose is a tool for defining and running multi-container Docker applications.
- It uses a YAML configuration file to define the services, networks, and volumes required by an application and manages them as a single unit.
- Docker Compose simplifies the process of orchestrating complex applications on a single host.

DOCKER ECOSYSTEM

DOCKER SWARM

- Docker Swarm is Docker's native clustering and orchestration tool for deploying and managing containerized applications at scale.
- It allows you to create a cluster of Docker hosts and deploy services across them, providing features like automatic load balancing, service discovery, and rolling updates.

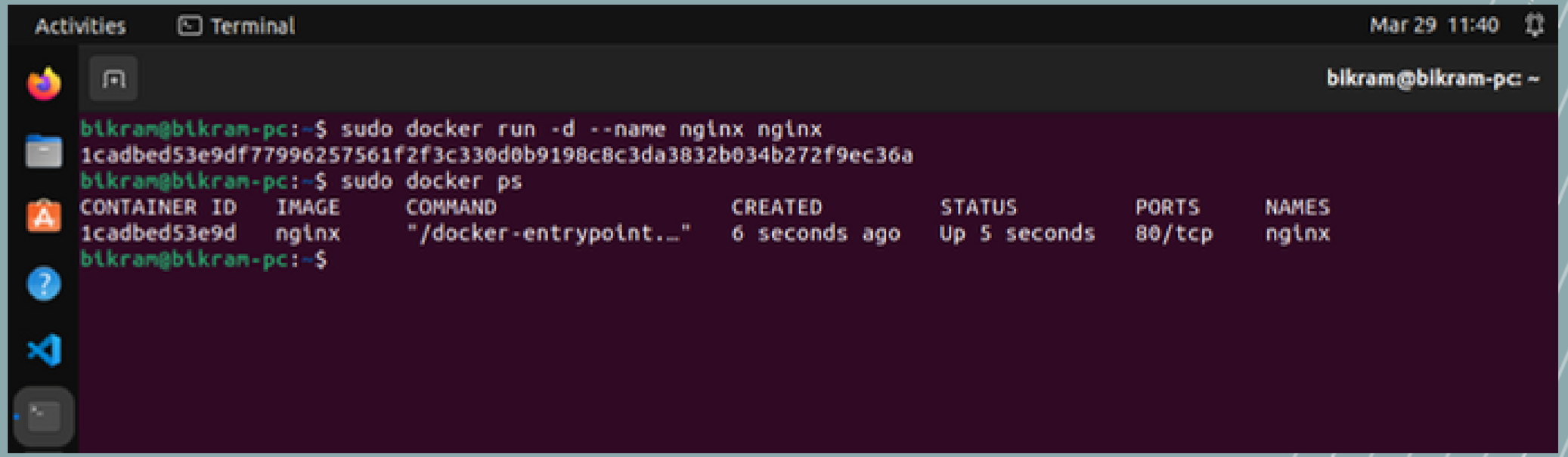
DOCKER ECOSYSTEM

DOCKER REGISTRY

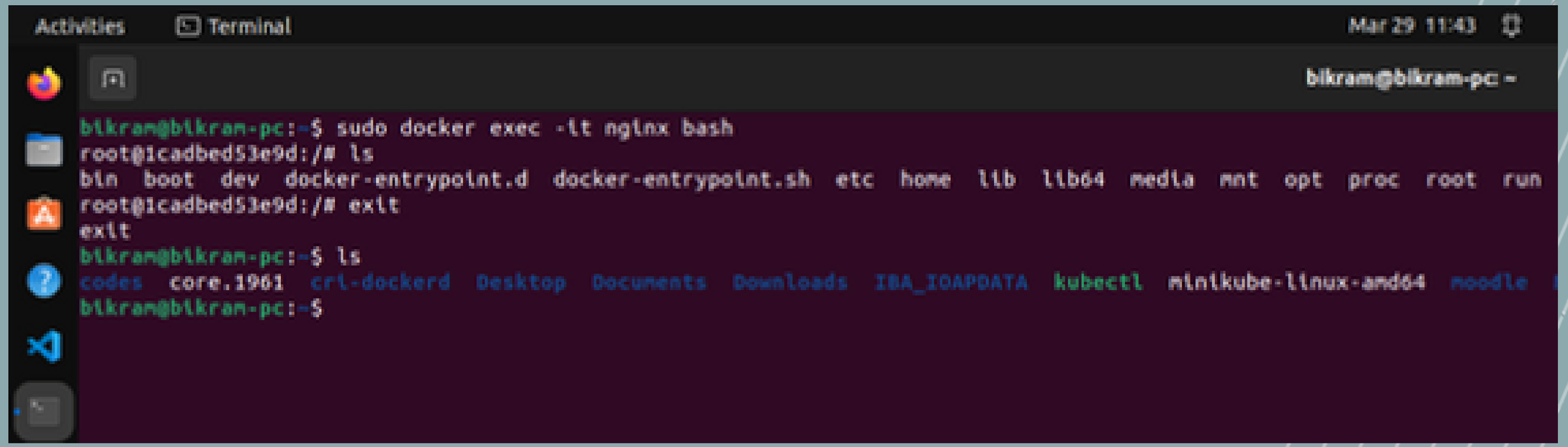
- Docker Registry is a repository for storing and distributing Docker images.
- It allows you to push and pull container images to and from a central location, making it easy to share and distribute applications built with Docker.
- Docker Hub is the official public registry provided by Docker, while Docker Enterprise includes a private registry for organizations.

PROCESS ISOLATION IN DOCKER

RUNNING AN IMAGE ON DOCKER



FILESYSTEM ISOLATION IN DOCKER



NETWORK ISOLATION IN DOCKER