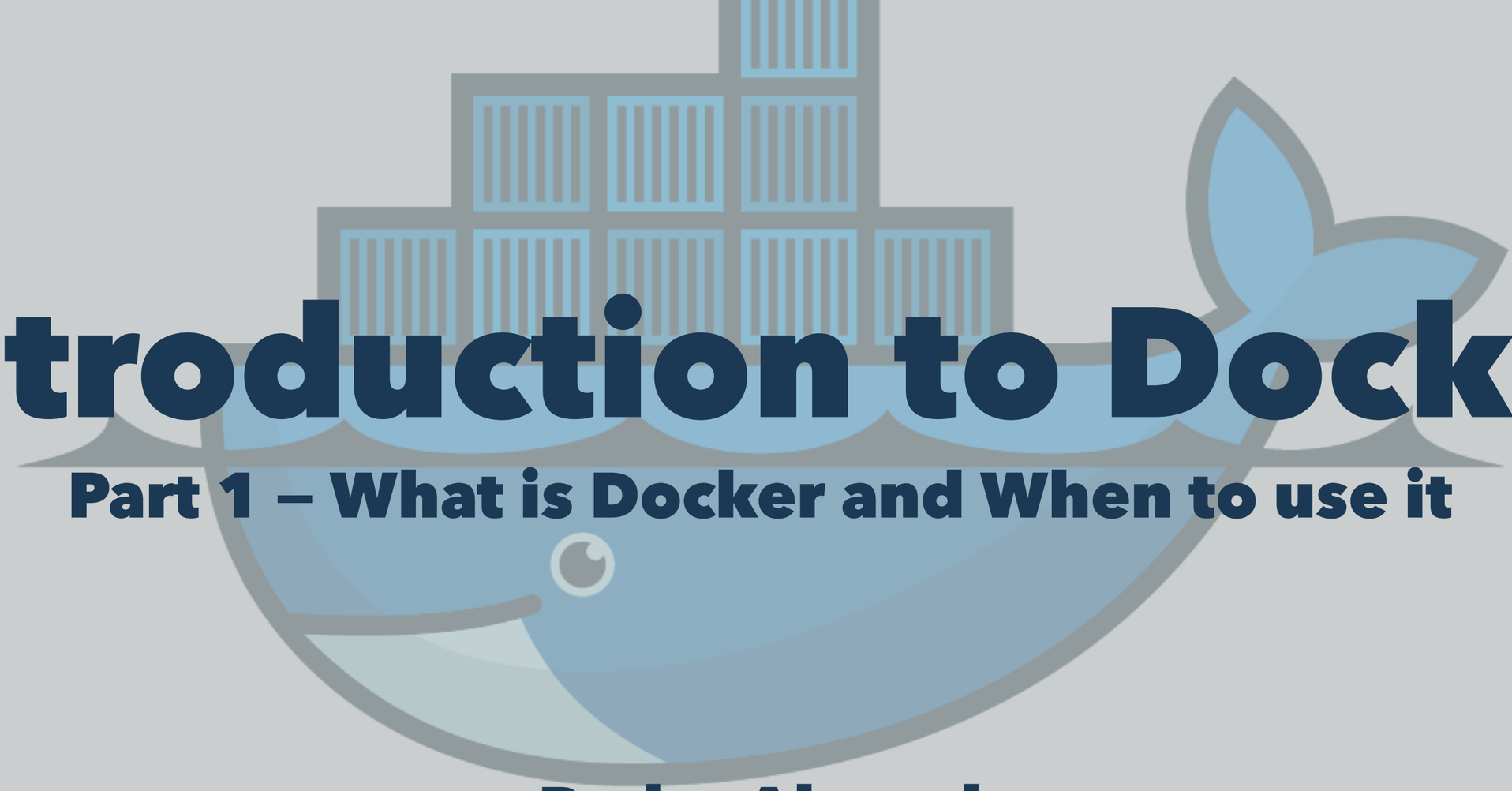


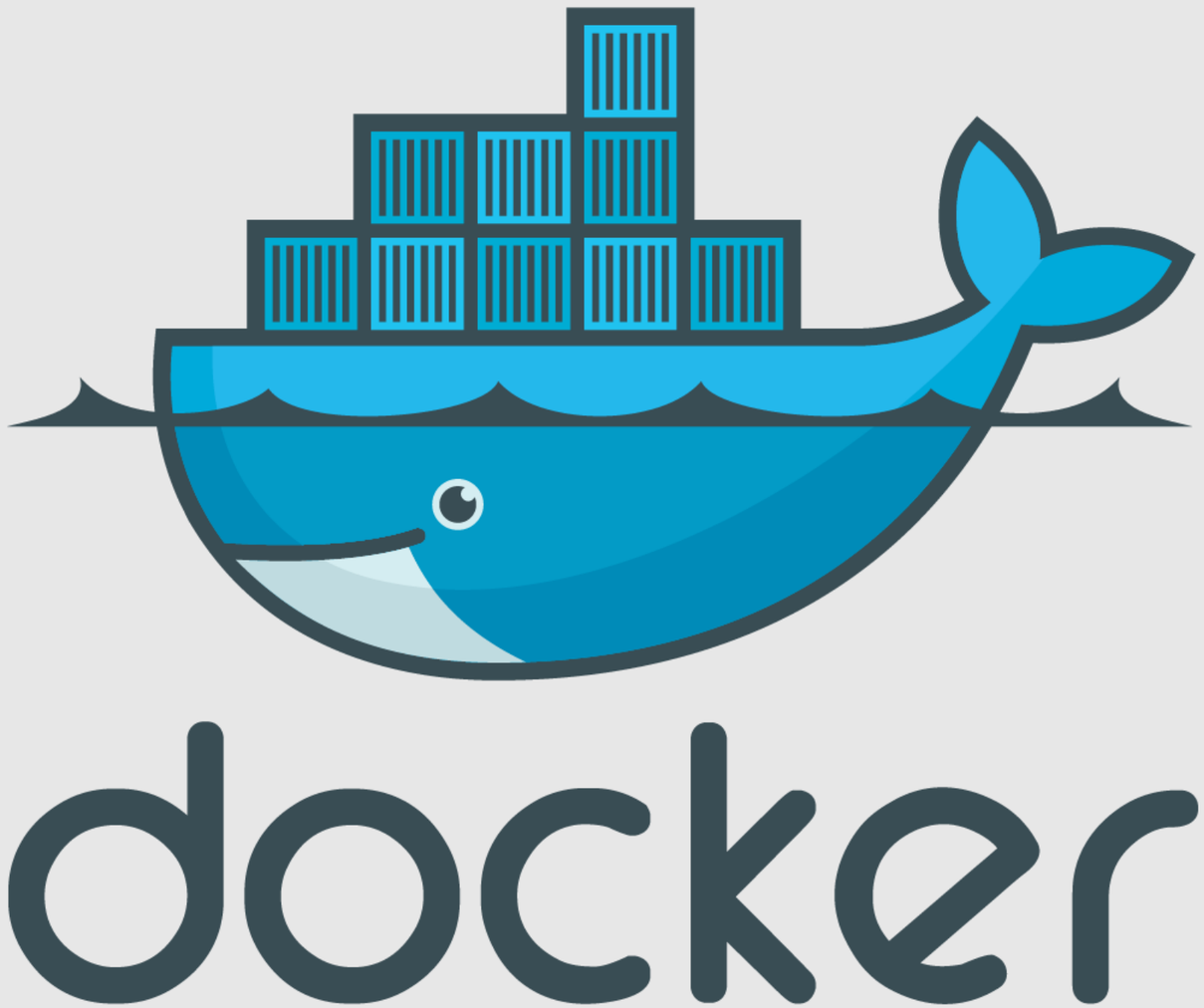
Introduction to Docker

Part 1 – What is Docker and When to use it

Badar Ahmed

What is Docker?

- New technology launched in 2013.
- Builds on top of Linux kernel features for OS level virtualization.
- Isolated containers where processes can run without impacting each other (process, memory, filesystem isolation).



What problems does Docker solve?

- Shipping code is hard. Docker makes it easier.
- When things break in production, most common developer excuse is *"Works on my laptop"*.
- Docker provides a standardized container that works for shipping software to different environments.

When to use Docker?

- Anywhere where you are deploying software between multiple environments.

- Docker provides an easy to use and portable way to run same software anywhere.

- You can run same container on your laptop, same in CI environment and same in production environment.

When to use Docker?

- Docker containers are order of magnitude faster than VMs.

- They can be ported on virtually any environment that runs relatively latest version of the Linux Kernel.

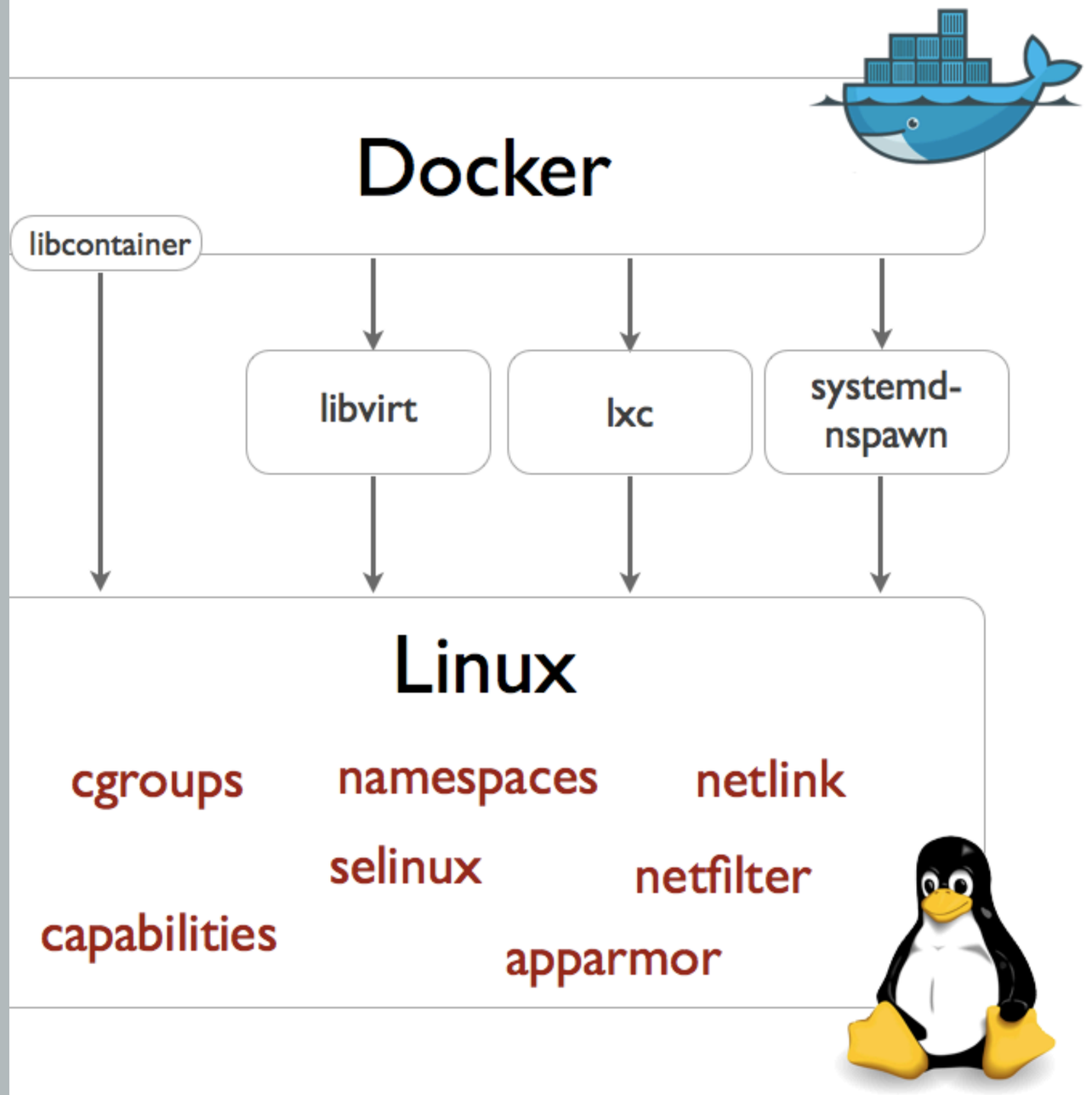
- Can be deployed on Public Cloud (AWS/GCP/Azure), Private Cloud or on Bare Metal.

When NOT to use Docker?

- Docker ecosystem is changing very fast as of right now.
- If you want something mature or stable this is not it.
- Fully leveraging Docker containers does require rethinking how to refactor or redesign code deployment pipelines.

Container Ecosystem

- Docker is the defacto container engine & format.
- Rocket by CoreOS is another alternative.
- Lower level container primitives like cgroups & LXC are harder to use.
- Docker provides a really nice API on top of those lower level primitives.



Container Ecosystem

- A lot of ecosystem around containers is trying to solve problems of container scheduling, orchestration, networking & storage.

- Major container scheduler/ orchestrators are Docker Swarm, Nomad, Kubernetes & Mesos.

How does Docker affect DevOps?

- Less sysadmin work.
- Less variance between developer local environment and prod environment.
- Enables more DevOps, since developers can easily package their own code!
- Allows easier integration and end to end testing.

Thanks!