CLOUD NATIVE APPLICATION DEVELOPMENT

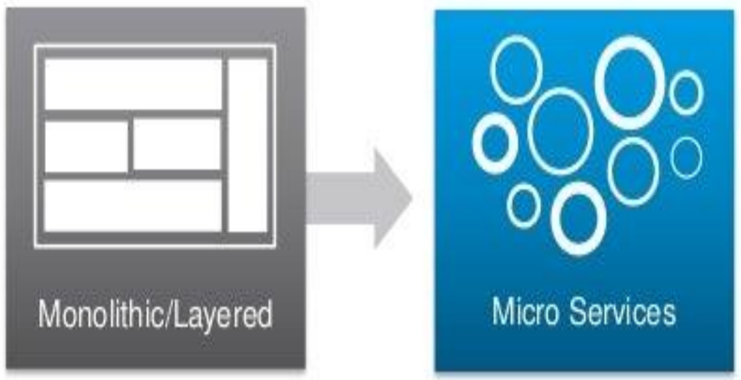
22AIE305
Introduction to
Cloud computing
2-0-3-3

Krishnapriya P S
Dept. of Computer Science & Engineering
Amrita Vishwa Vidyapeetham

Agenda

OVERVIEW OF CLOUD NATIVE APPLICATIONS

Application Design is Changing



Properties of a Microservice

- Small code base
- Easy to scale, deploy and throw away
- Autonomous
- Resilient

Benefits of a Microservices Architecture

- A highly resilient, scalable and resource efficient application
- Enables smaller development teams
- Teams free to use the right languages and tools for the job
- Rapid application development

What are Cloud-Native Applications?

Developer access via APIs

Continuous integration and deployment

Built for scale

App-defined Availability

Microservices, not monolithic stacks

Decoupled from infrastructure

Motivation Behind Cloud-Native Applications

Software-Defined Everything

Regardless of industry, businesses increasingly rely on technology to differentiate

IT shifting from cost center to enabler of growth

Rich Customer Experience

Customers demanding enhanced engagement

Third Platform: mobile, social, analytics, cloud

Speed and Agility

Improving time-to-value with DevOps, Agile

New architectures to streamline workflows

New Capabilities and Priorities

Infrastructure must be dynamic, API-driven, highly scalable

Small teams able to manage large fleets

Cloud-Native Architectures Less Reliant on Rich Infrastructure

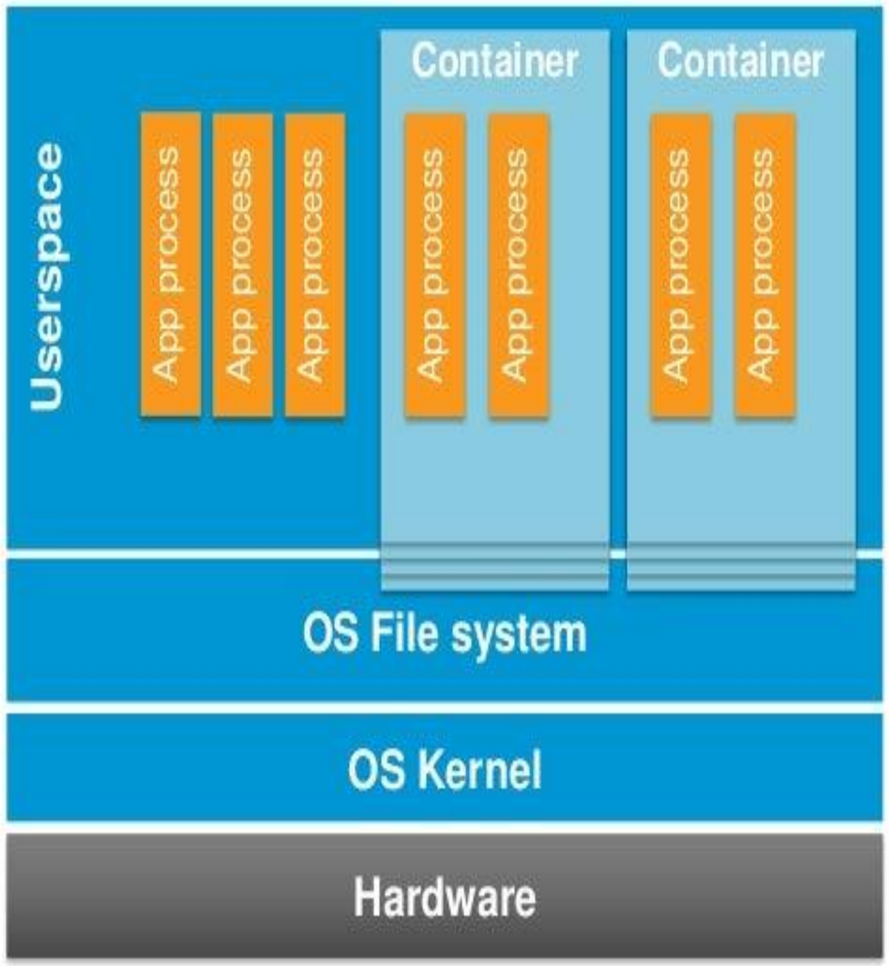
	Traditional App	Cloud-Native App
Architecture	Monolithic	Microservices
Configuration	Procedural	Declarative
Availability	Handled by infrastructure	Integrated with app
Storage	Centralized, redundant	Scale-out object storage and NoSQL / key-value stores
Response	Shifting demand handled by DRS, scale-up	Scale-out, load balanced traffic
Interface	GUI, robust orchestration	API/CLI preferred, integration with DevOps toolchain

Containers Are Aligned with Cloud-Native Architectures

- Application portability supports wide range of environments
- Automation and integration with CI/CD and DevOps workflows
- Quick to instantiate and iterate, shortening feedback loops
- Stateless nature aligns with scale-out designs

But containers are not just next-generation VMs

Linux Containers



OS-level Isolation

- Isolation at individual kernel subsystem level (e.g. filesystem, process table, etc)
- User-level process (LXC, libcontainer) orchestrates these subsystems to create a container

Existed for Many Years

Solaris Zones, FreeBSD Jails, OpenVZ

Why?

- Process isolation
- Reproducible environment
- Enables management at scale

docker is a “Shipping Container” for Code

Developers ♥ docker because …

- Frictionless deployment and maximum portability

  On developer laptop:

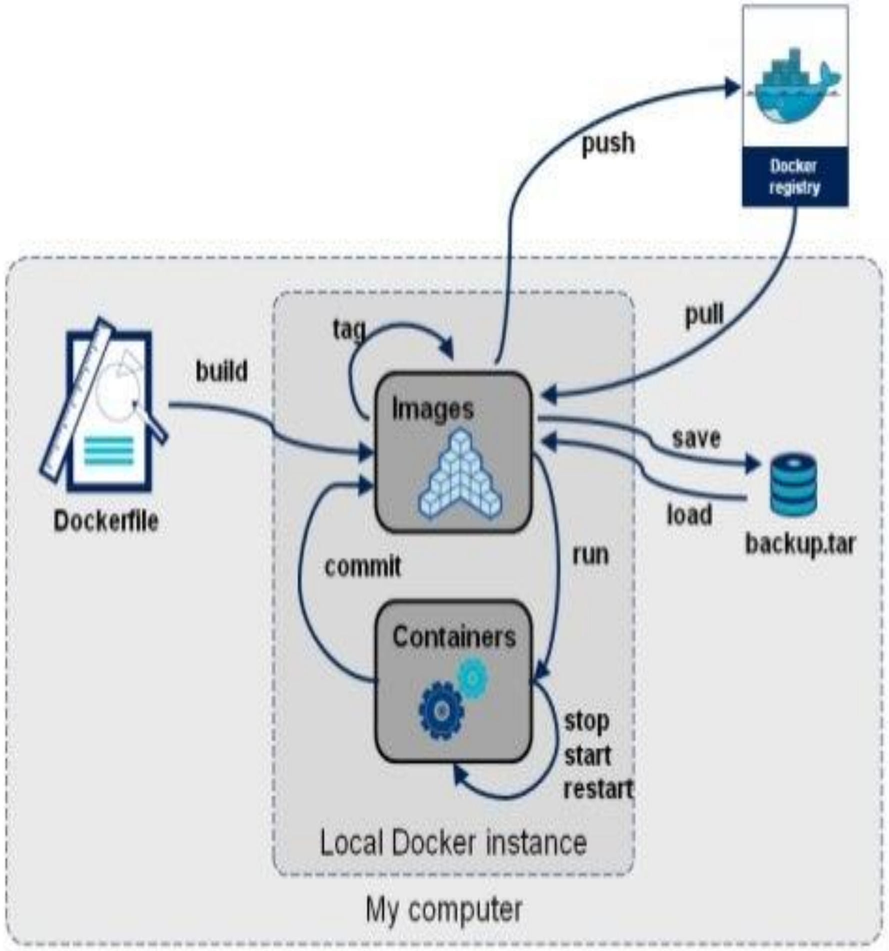
```
~# docker build my_app
~# docker push my_app
~#
```

  Then on server:

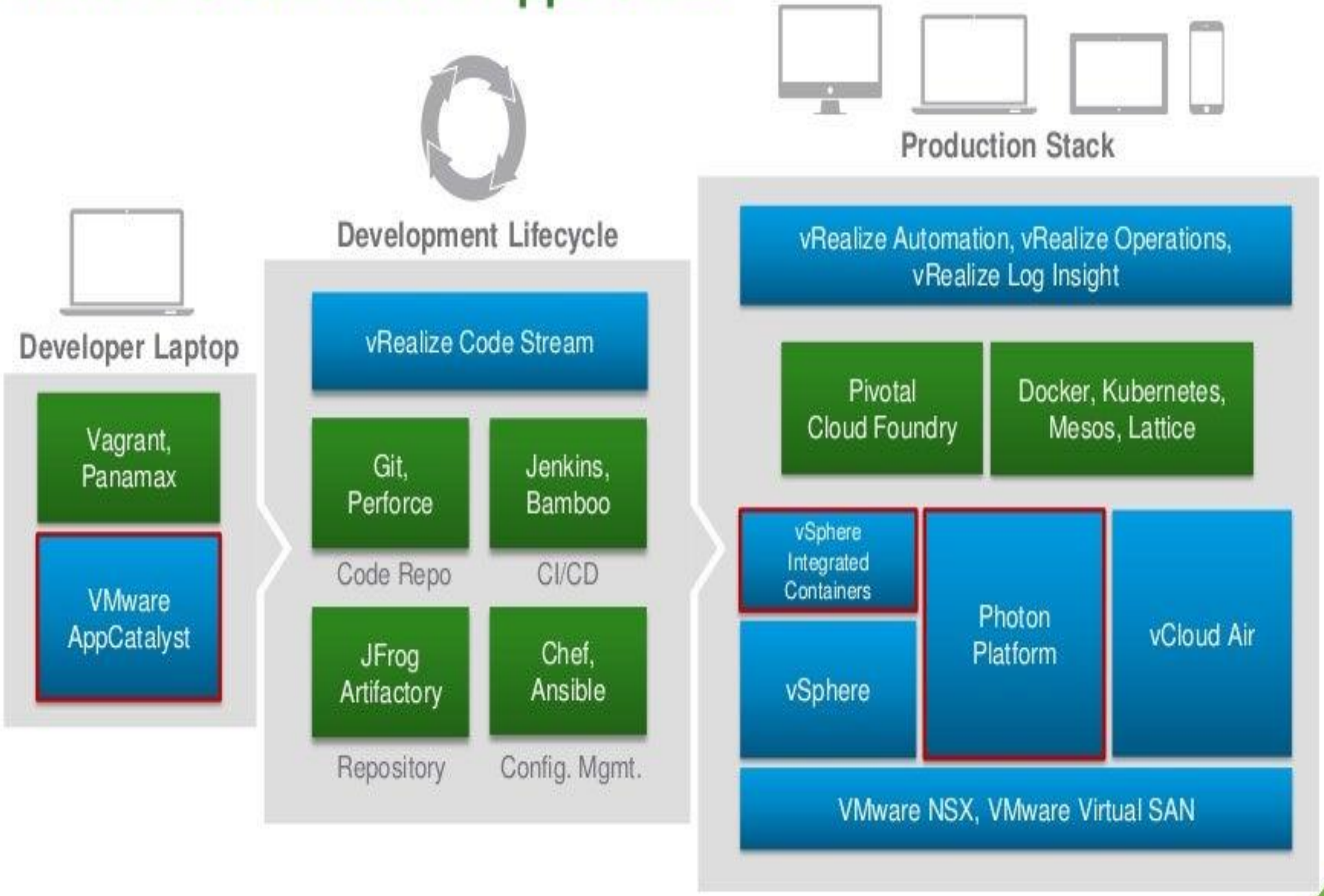
```
~# docker pull my_app
~# docker run my_app
~#
```

  That’s it!!

- A natural fit for 3rd Platform, 12 factor, microservices
- It makes DevOps much, much easier



VMware Cloud-Native Apps Stack



VMware AppCatalyst

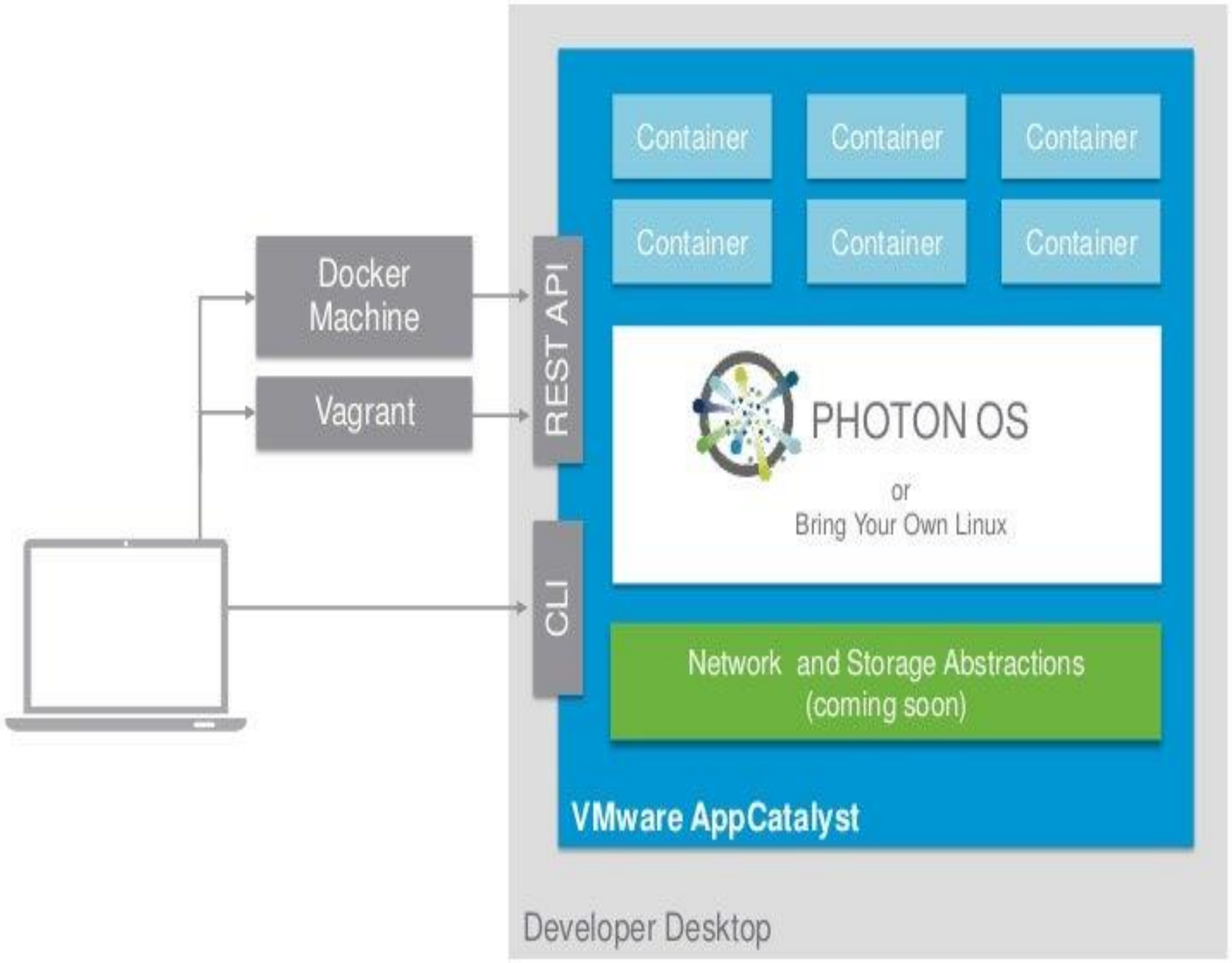
Ready for Cloud Native
AppCatalyst ships with Photon OS and Vagrant, and supports Docker containers out of the box.

Built for Developers
AppCatalyst is REST API- and CLI-driven for seamless integration with container- and microservices-based workflows.

Free to Use
AppCatalyst is available at no cost to the user, and ready for download today.

Download Technology Preview Now!
http://getappcatalyst.com

VMware AppCatalyst



THANK YOU