A Journey through expertise...

About Us...

Your Kubernetes Cluster is in Safe Hands

Red Hat OpenShift Managed Service

As a Red Hat Premium Partner, Prodevans brings years on experience in managing Red Hat OpenShift Platforms across the Globe.

16 Year Experience with Red Hat

- #1 Professional Service & Training Partner in the Country
- High Expertise in Complete Red Hat Suite of Products

First Hand On Field Experience

- Solid Track record on OpenShift Implementations
- Provided Solutions across India & ASEAN

Digital Transformation Specialists

- Great track record in digital transformation with successful executions in BoA, Citi Singapore, ANZ & Exide Life

Bespoke Solutions

- Great track record in digital transformation with successful executions in Citi Singapore, ANZ & Exide Life

Red Hat Certified Consultants

- Highly Qualified Professionals who provide technical & Leadership skills

Your Kubernetes Cluster is in Safe Hands

Red Hat OpenShift Managed Service

Prodevans Capability for Hassle Free OpenShift

Latest Red Hat Technologies

- Support for the latest software releases
- integration with CloudForms to monitor OpenShift Container Platform health
- Red Hat Ansible's automation engine
- Access to image registry with Red Hat Satellite
- Red Hat HAProxy for load balancing

Enterprise Grade Service

- Cloud experts trained to run and manage Red Hat OpenShift technology
- 24x7 support

Prodevans Container Adoption Program

- Trust our World Class Experience to
  - Migrate the right things
  - Migrate them right
  - Migrate at scale

Agenda

Things to look forward to:

- Container Adoption Journey
- Introduction to Red Hat OpenShift Container Platform
- Understand container and OpenShift architecture
- How to Create containerized services
- Application assessment & Discovery
- How to run an Application on OpenShift

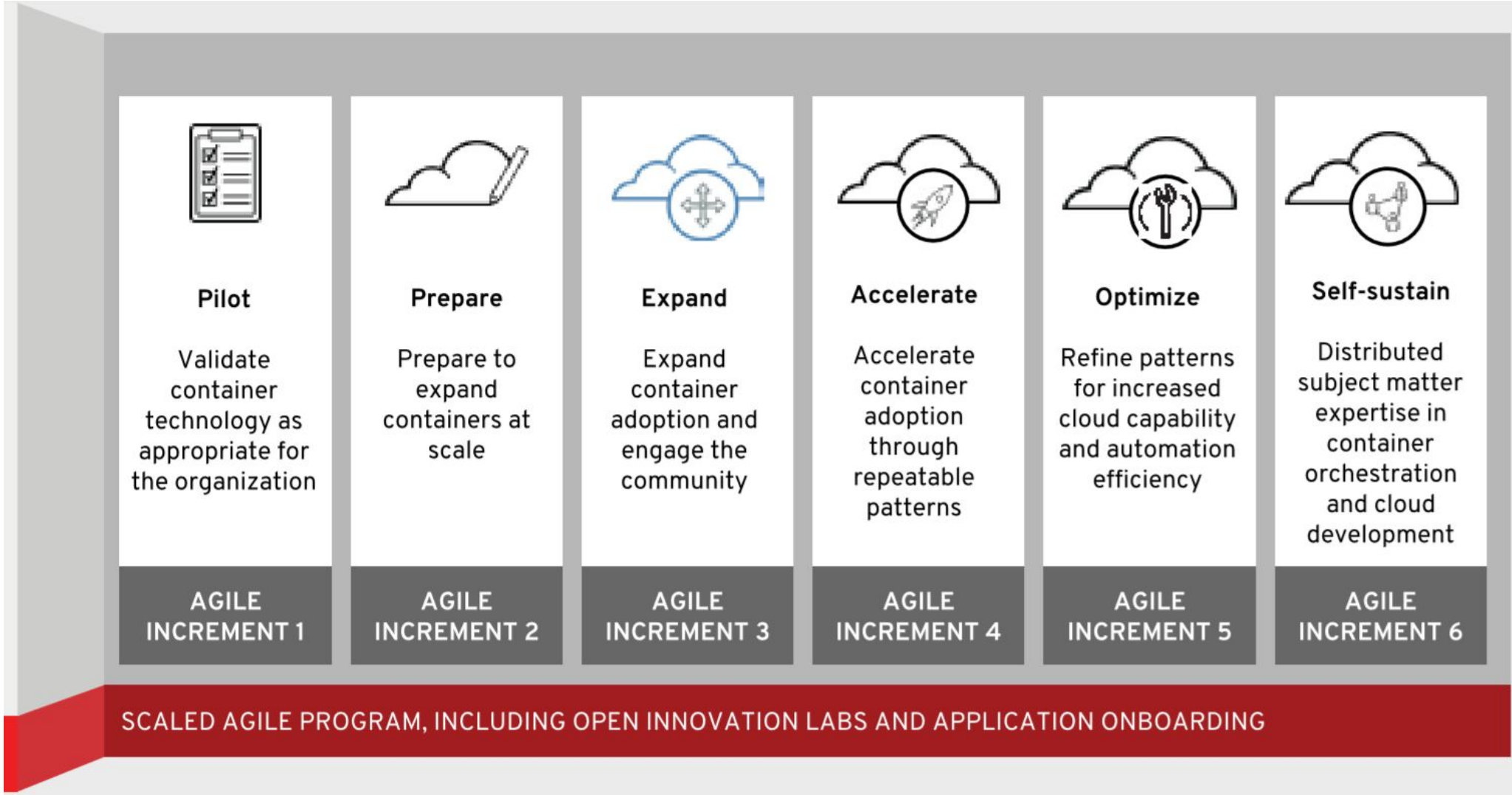
Container Adoption Journey

Container Adoption Journey : Overview

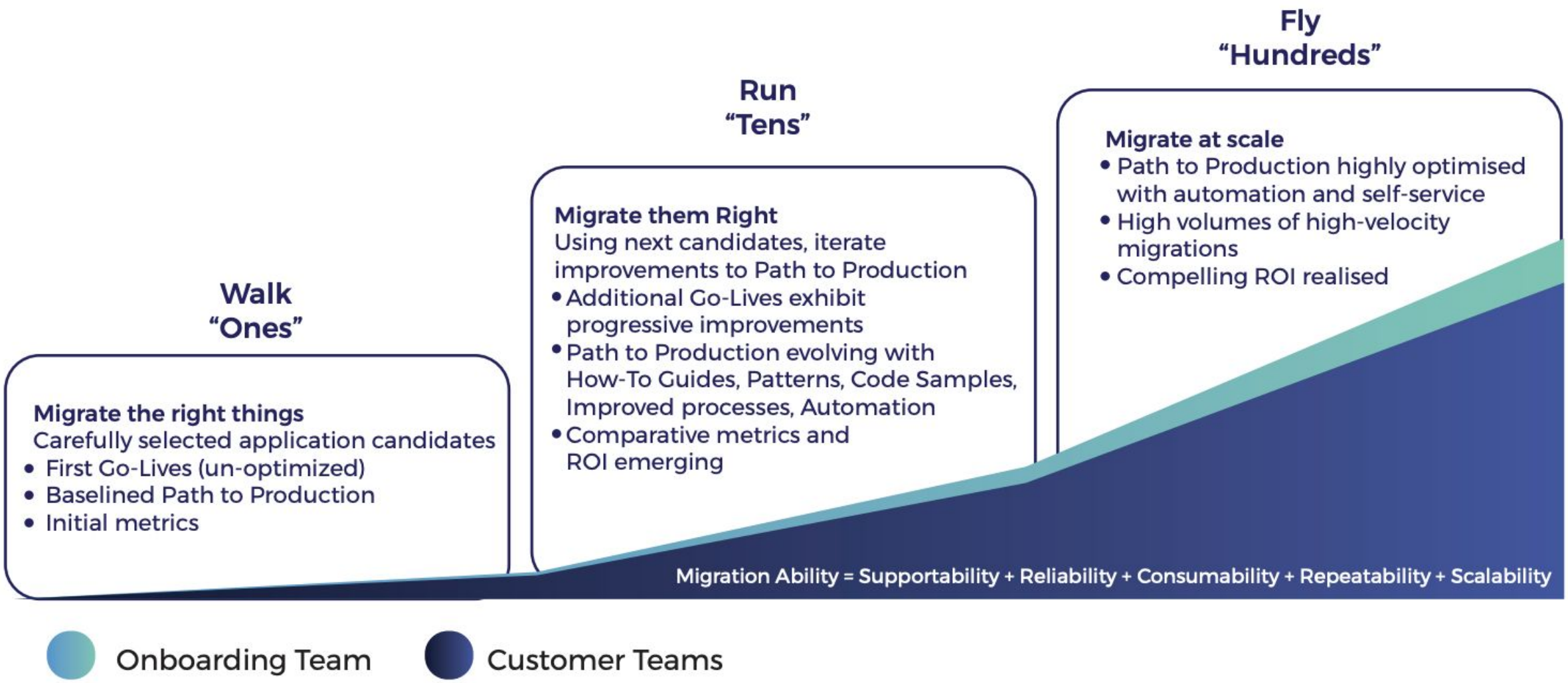
Six Steps to Digital Leadership

BEGIN

Discovery session



Container Adoption Journey : Overview



Introduction to Container

WHAT ARE CONTAINERS?

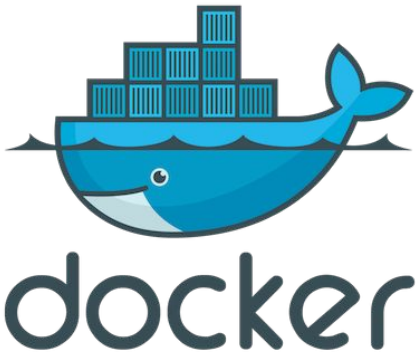
It Depends Who You Ask

- Application processes on a shared kernel
- Simpler, lighter, and denser than VMs
- Portable across different environments

- Package apps with all dependencies
- Deploy to any environment in seconds
- Easily accessed and shared

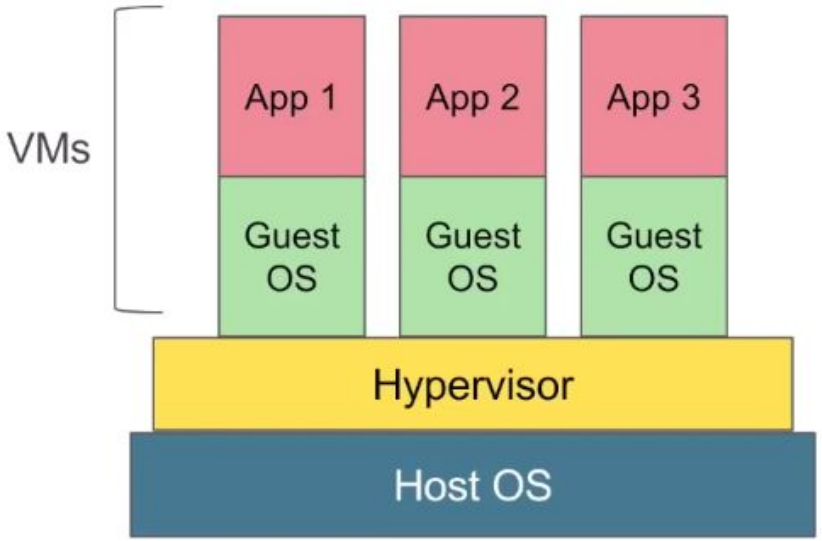
What Is Docker?

- World’s leading software container platform
- Docker is a tool designed to make it easier to deploy and run applications by using containers

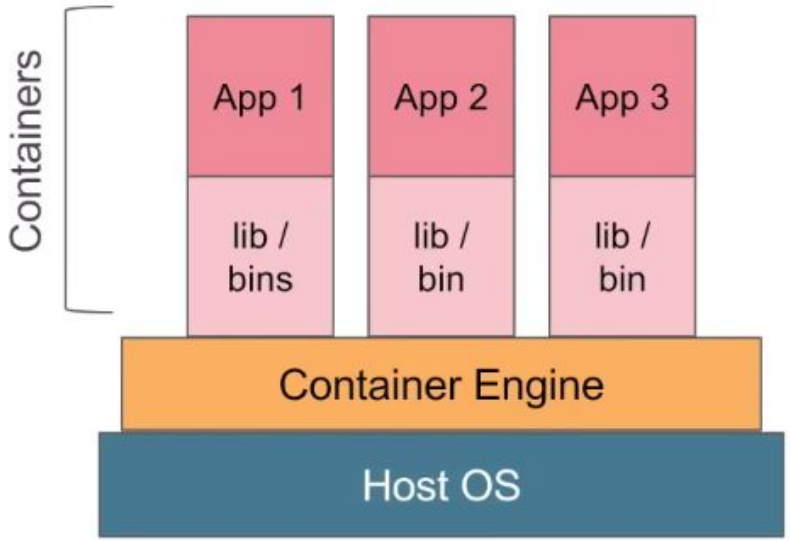


- Package your application with all its dependencies in a container
- Docker makes the process of application deployment very easy

Virtualization Vs Containerization

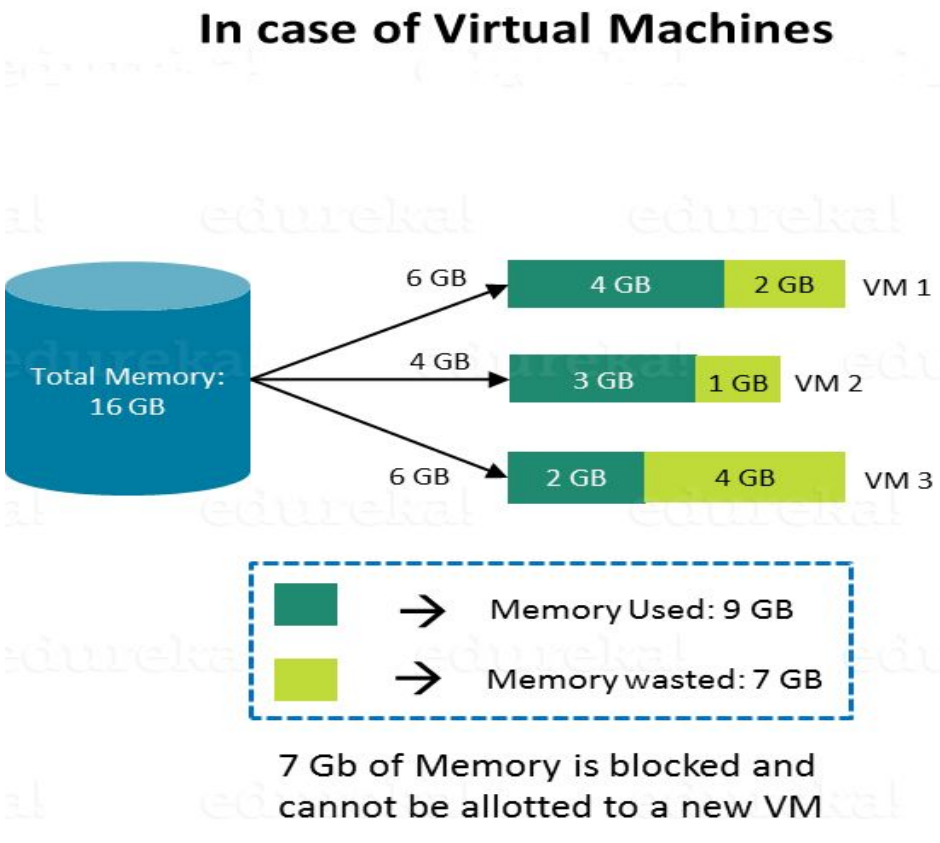


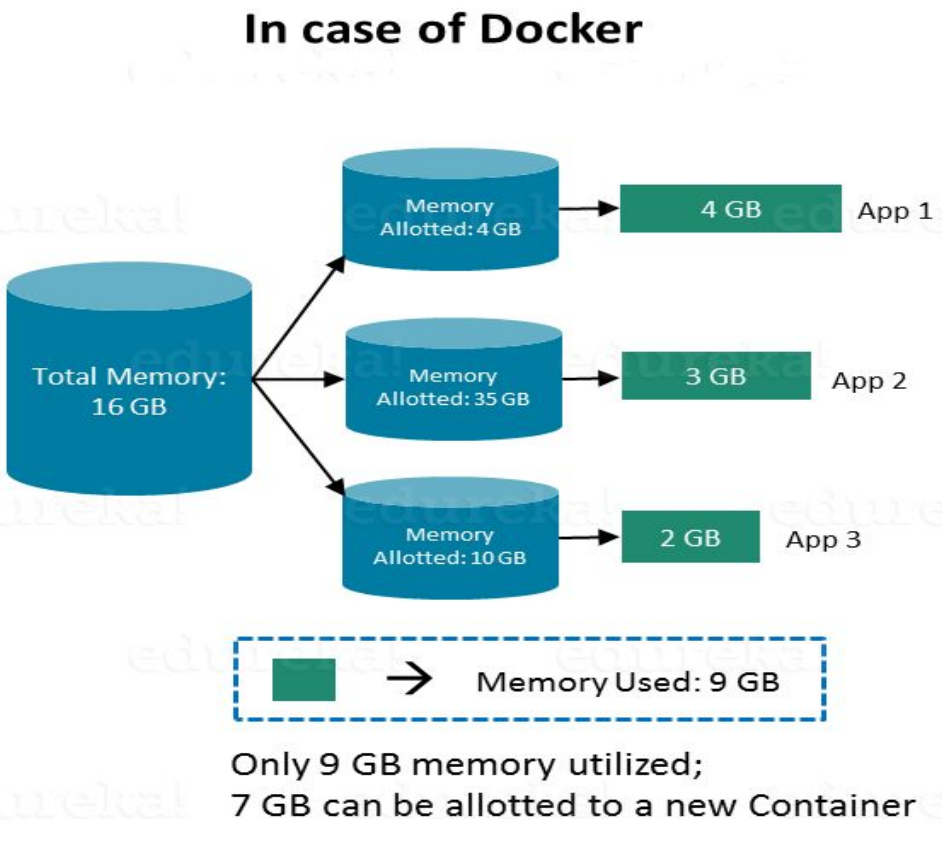
Virtualization



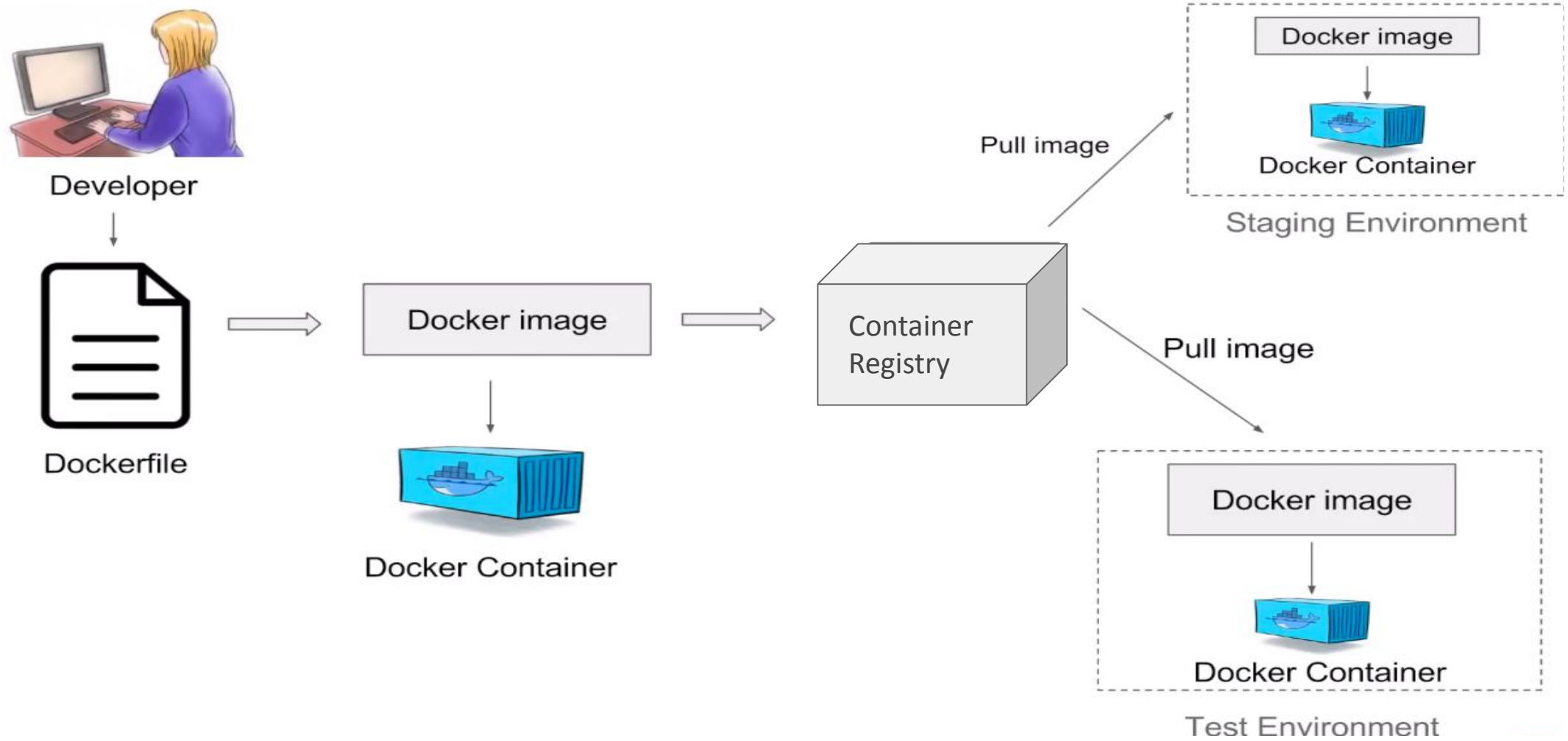
Containerization

Virtualization Vs Containerization

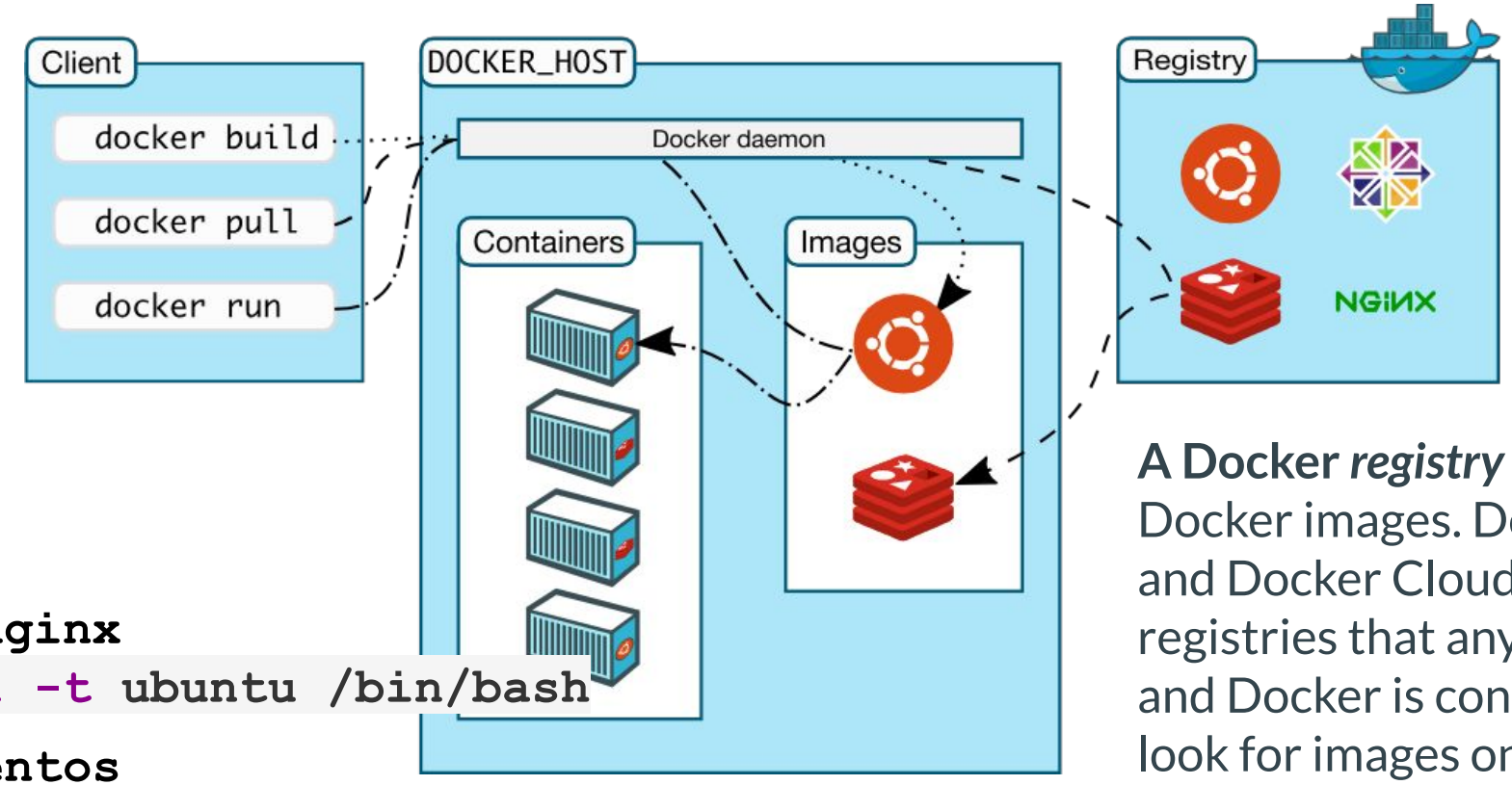




How Docker Works



How Docker Works



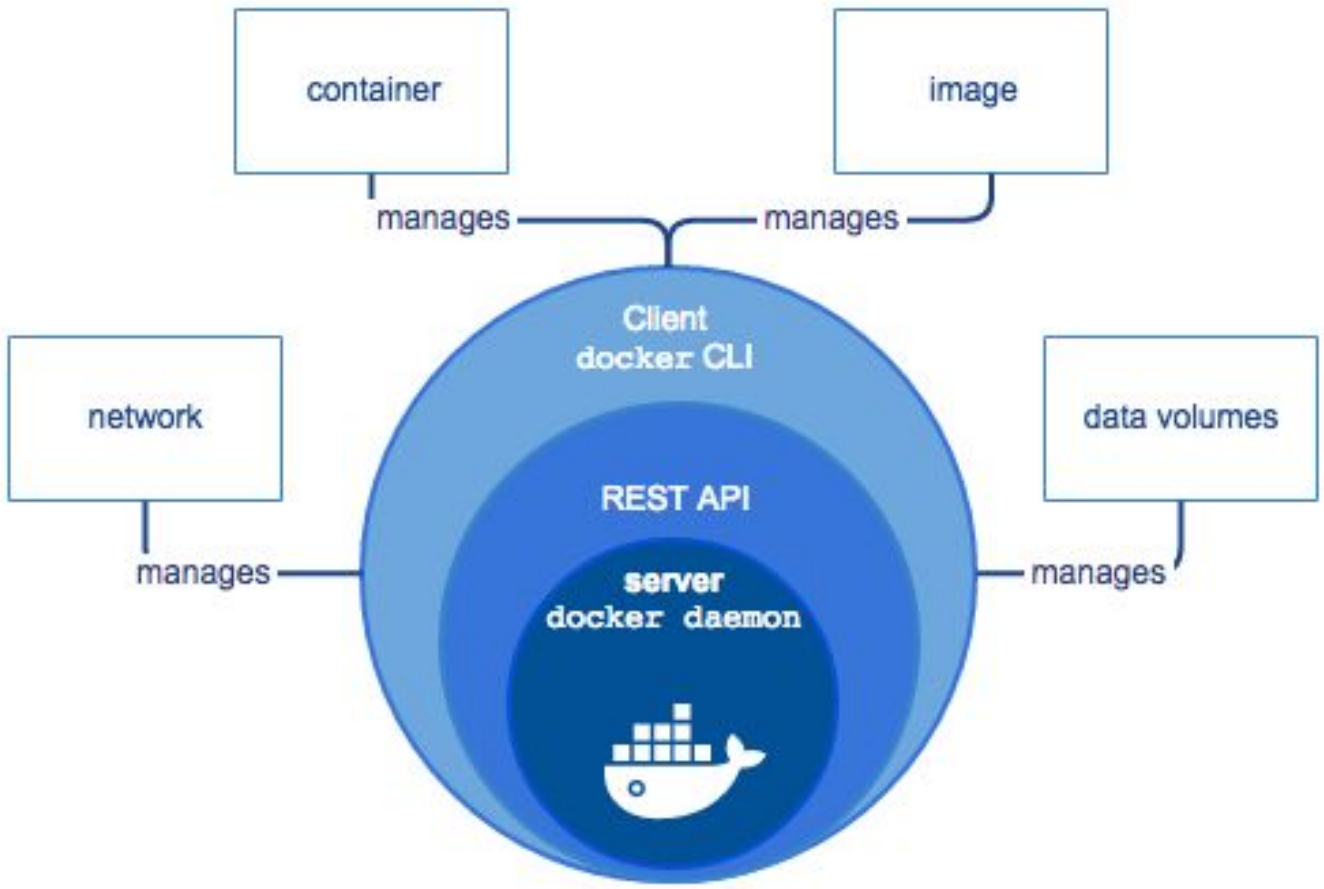

```
docker pull nginx
docker run -i -t ubuntu /bin/bash
docker run centos
```

A Docker *registry* stores Docker images. Docker Hub and Docker Cloud are public registries that anyone can use, and Docker is configured to look for images on Docker Hub by default. You can even run your own private registry.

Docker Architecture

The CLI uses the Docker REST API to control or interact with the Docker daemon through scripting or direct CLI commands.

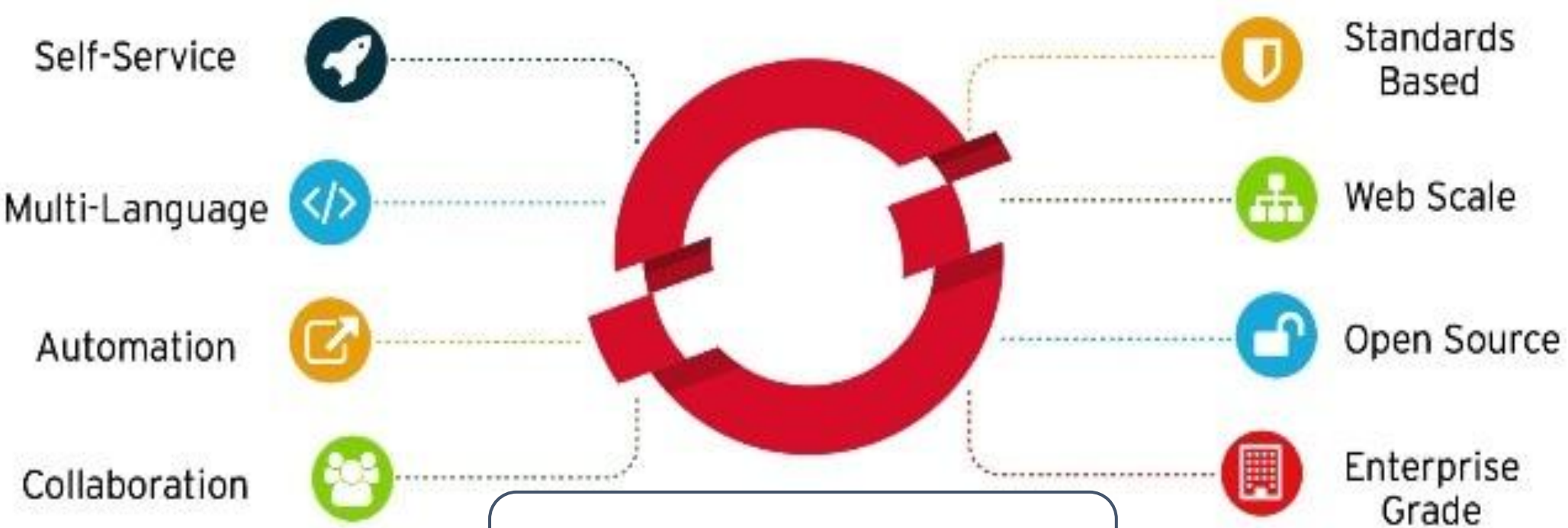
The daemon creates and manages Docker *objects*, such as images, containers, networks, and volumes.



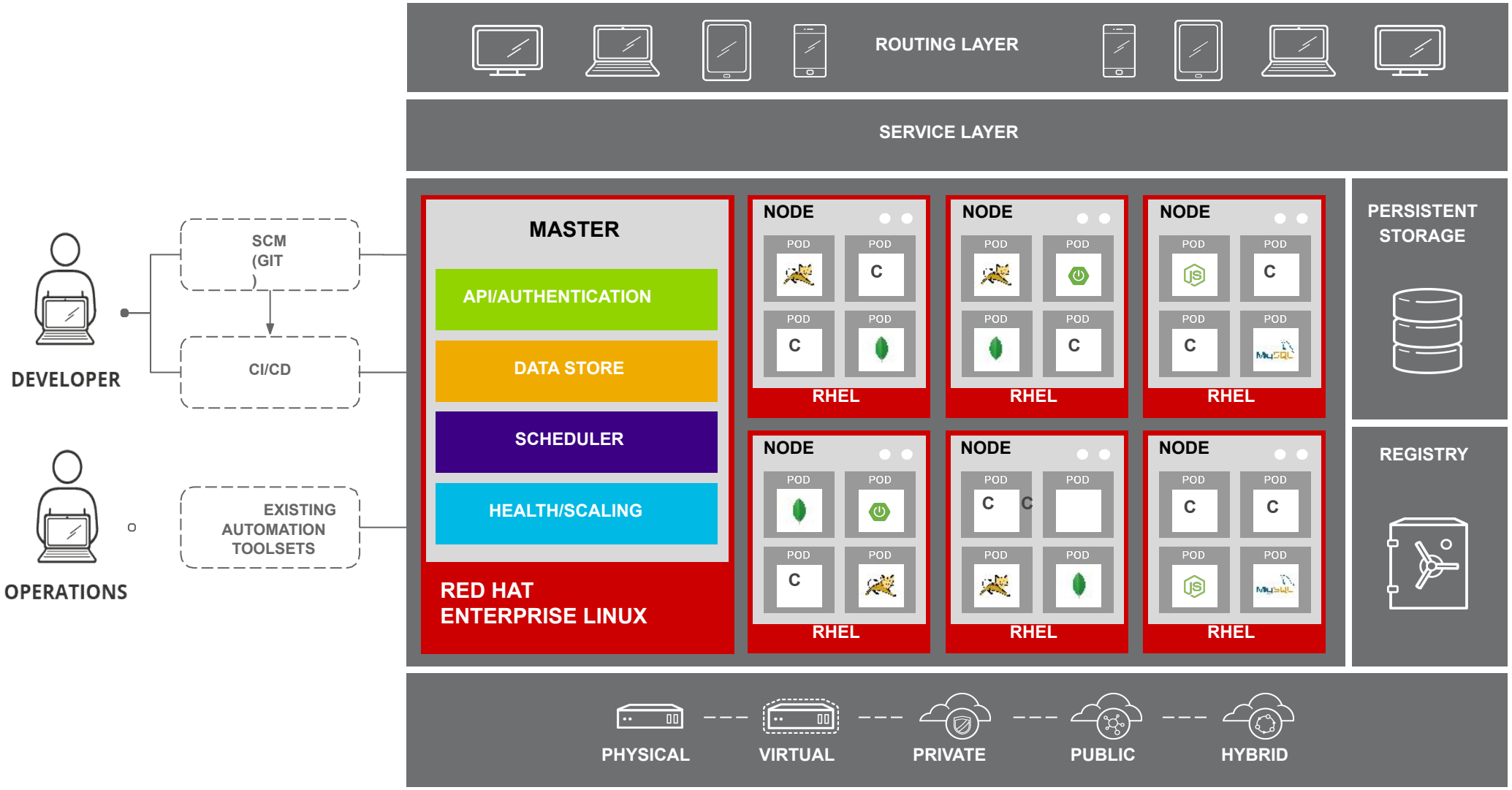
Docker Engine Architecture

Introduction to OpenShift

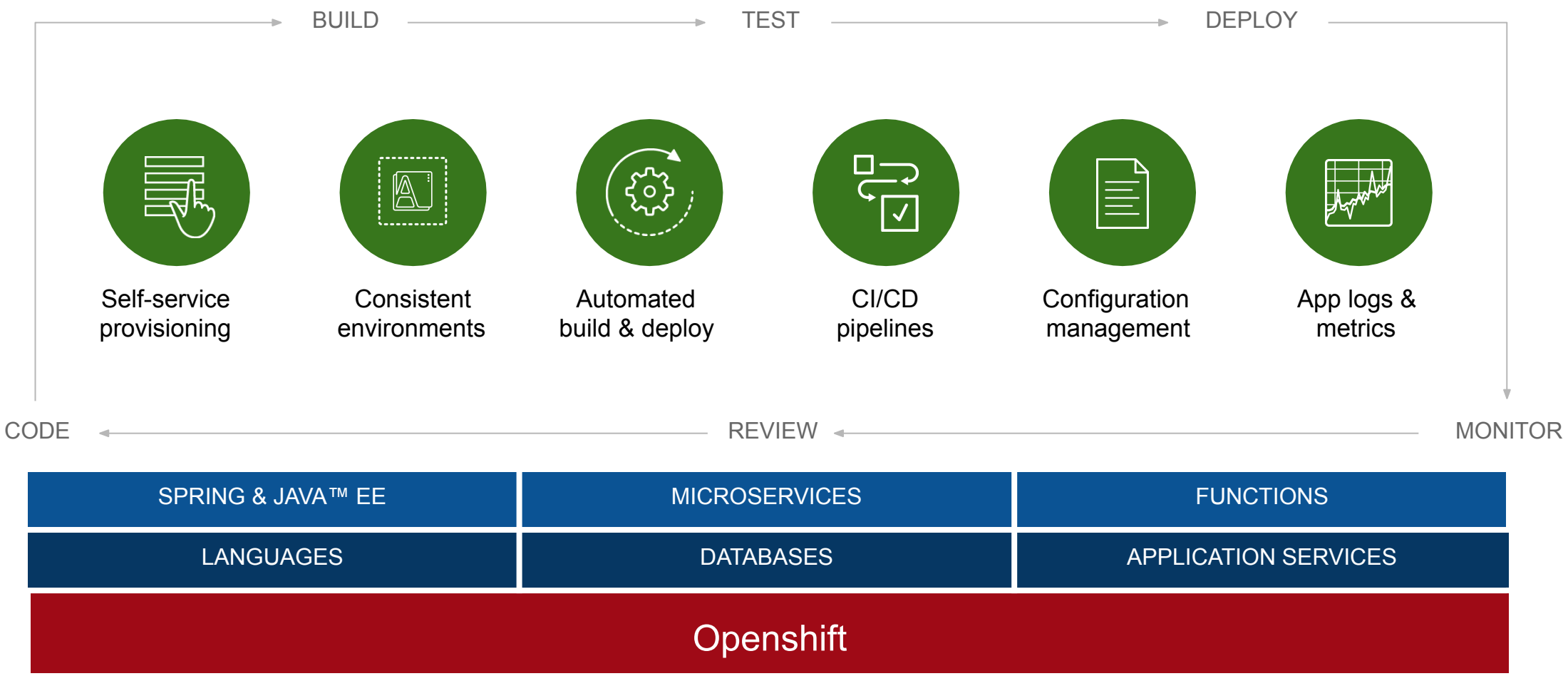
Openshift Container Platform



OpenShift Architecture



How to create containerized services.



Application Assessment

Factor	Recommendations
APP SERVER	▪ If the application uses the JNDI API. JNDI is typically used for Java EE based application. It may be worth considering using a dependency injection framework to perform any resource lookups, rather than using the JNDI API. All connection properties should be pull from environment variable. ▪ If the application uses the EJB API(s). It is recommended to migrate EJB beans to REST based services. ▪ If the application uses Spring's JNDI integration API. It may be worth considering using a dependency injection framework to perform any resource lookups, rather than using the JNDI API. All connection properties should be pull from environment variable
CONFIG	▪ All configuration data should be stored in a separate place from the code, and read in by the code at runtime. Usually this means when you deploy code to an environment, you copy the correct configuration files into the codebase at that time. ▪ Don’t hard code configuration details into the code. Instead, store them as environment variables that can be easily changed based on the target environment.
DEPENDENCIES	▪ Dependency libraries typically provided by the container should not be included in the packaged application. Developers have to check for existence of ANY dependency and install what’s needed.
FILESYSTEM	▪ In case the application uses the java.io.File api. File API is designed to work with the local file system, potentially disabling the cloud nature of the application. ▪ If the application potentially uses one or more external processes. This is typically in the form of an environment process such as UNIX/WINDOW script or some other native process. These processes should be migrated to be consumed using standard Micro/backing service conventions.
HARDWARE	▪ Loading shared libraries (.so files for Unix or .dll files for Windows systems) through java.lang.System.load* OR java.lang.Runtime.getRuntime().load* can make applications dependent on specific Operating system and/OR Hardware. It is recommended to use 100% pure java libraries whenever possible.
LANGUAGE	▪ Classes or libraries those were compiled with an JDK compiler older than the specified version. The concern is that Java 1.6 is no longer supported and Java 1.7 is end of life. There is a general push toward using Java 1.8 or higher.
LOGGING	▪ Developers should capture errors/logs and send them to an error/log reporting service like New Relic or AirBrake. You can take a more general approach and send your logs to a service like PaperTrail or Splunk Storm.
PORTABILITY	▪ In case the application uses JBoss API, It is almost certain that the application will have to be modified to remove this dependency. The use of the API should be examined to determine how much effort will be required to remove it. In some cases, such as any use of org.jboss.mx.util.MBeanServerLocator, the removal will be trivial. In other cases more effort will be required. ▪ If the application uses WebSphere API. It is almost certain that the application will have to be modified to remove this dependency. The use of the API should be examined to determine how much effort will be required to remove it.
PORT BINDINGS	▪ If the application uses the RMI API(s). It is recommended to migrate RMI services to REST based services. Most cloud native application/microservices communicate over HTTP port 80 or HTTPS port 443 which is a natural fit for REST services. Also, it should be a goal that each cloud native application microservice can be consume by any client written in any language. RMI is targeted toward Java clients only by default (if RMI-IIOP is not used). ▪ Applications targeted for Cloud deployment should not directly listen to custom ports using java socket API. Use Web-Services, REST, WebSockets, etc. using standard HTTP(S) ports whenever possible.
STATELESS	▪ If the application uses the HTTP session API(s). Removing use of the HTTP session enables the application to be stateless. In case the removal of sessions are impossible, you might implement a distributed session mechanism based on a distributed caching mechanism like Redis or Gemfire.

Cloud Native Assessment Early Findings

- Sample Reports

How to run an App on OpenShift

Cloud Native Approach for App Tx

Red Hat | PRODEVANS

Externalize Config

Making Stateless

DevOps Pipeline

Externalize Logs

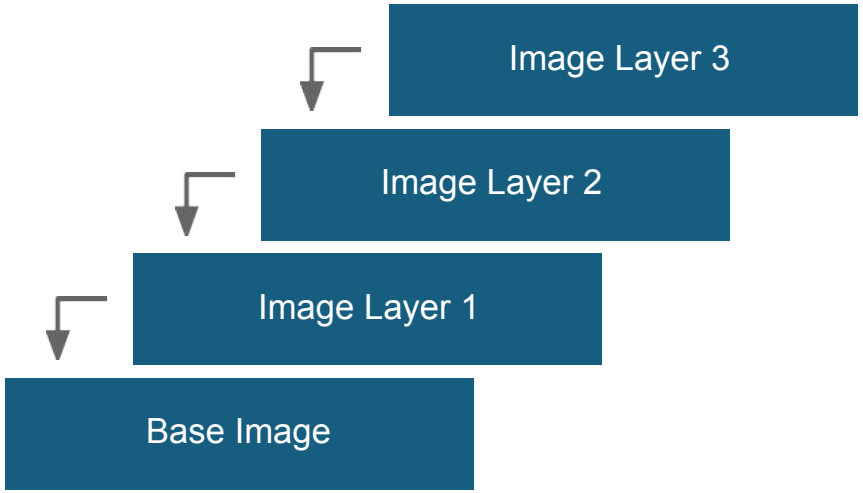
Application on Openshift

- ✔ No State
- ✔ No Manual Deployment
- ✔ Centralized Dashboard for Logging and Monitoring for PS Team
- ✔ Easy App Upgrade and Patching
- ✔ Complete Agile Adoption

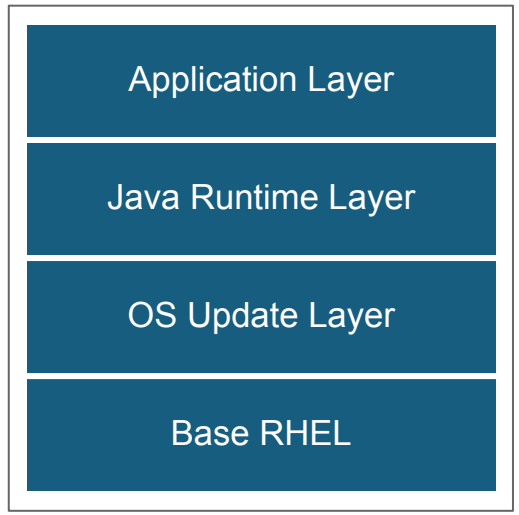
Statelessness Approach

01	**Session**	• Remove Session Objects from App • Use Tokens
02	**File System**	• Avoid generating unnecessary files and storing in FS • Use NAS based PV for OCP
03	**Caching**	• Use Cookies for temporary storage • Use External Cache with the application

Rapid Security Patching

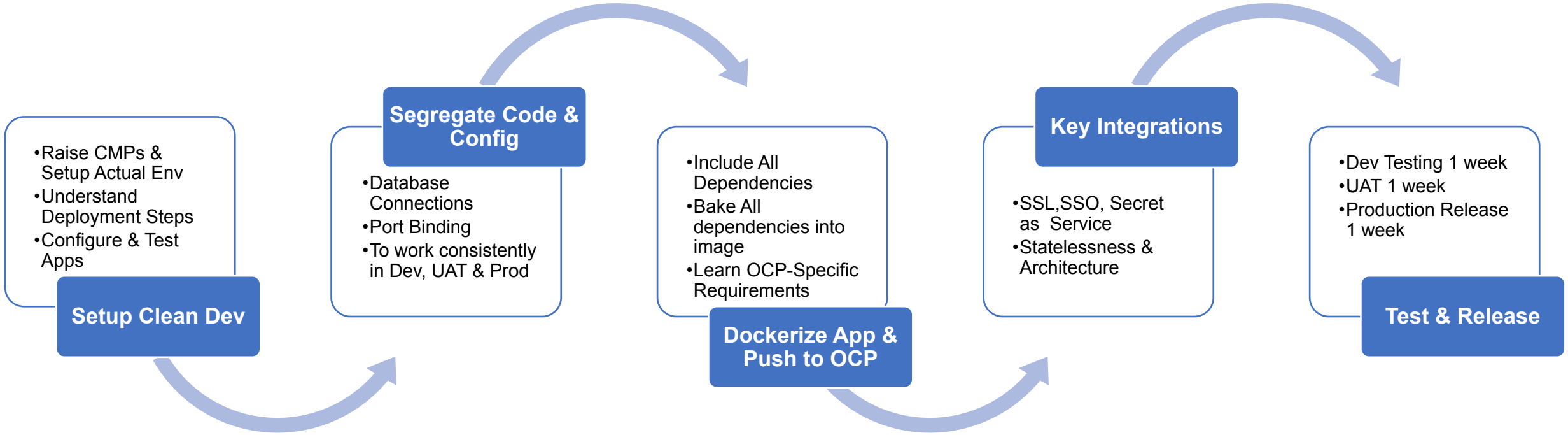


Container Image Layers



Example Container Image

Timeline View- Critical Path



- 11 weeks to bring an App to UAT
- Iterative learning Cycle, Process identification & first time resolution of issues
- Cloud Native Architecture, Integration, Statelessness, Understanding core capabilities required for progress
- OpenShift Platform Specific Capabilities one piece of a large jig-saw

Path to Production

Application Transformation WorkFlow – Consistent Path for Accelerated Onboarding

- ✔ App Functionality Discovery
- ✔ Understand & Record Deployment Steps
- ✔ Identify Configuration Parameters
- ✔ Identify Platform Migration Elements

(One week per App)

- ✔ App ReArchitect & Refactoring
- ✔ Platform Migration Code Changes
- ✔ **Statelessness**
- ✔ Externalize Config and Logs

(~4 weeks per App)

- ✓ App Code Changes for SSO
- ✓ Porting to New DB Schema & Secret Password Integration
- ✓ **Autoscaling Testing**

(~ 2 weeks per App)

- ✓ Onboarding to CICD(TeamCity)
- ✓ OCP Template and Deployment Scripts
- ✓ **One Touch Deployment**
- ✓ Map App with VIP & WLB

(~2 weeks per App)

~6-7 weeks & Max 11 weeks elapsed time for an Apps in a sprint to reach UAT in a BFSI Environment

Prod Release

Centralized Logging & Monitoring Dashboard for Prod Team

Real-world benefits

Challenge	Then	Now
Time-to-market & DevOps efficiencies	DevOps “deploy to production” for client application ~2 weeks.	DevOps “deploy to production” for client application ~1 hour.
Security	Config stored on local file system.	Effective consumption of SSO and Secrets Vault services.
Platform Reliability	Apps deployed on custom VM’s subject to rebuild/redeploy complexity.	Apps deployed on resilient and self-healing OCP pods.
Environment Qualities in MicroService	Single VM	Multiple Pods, High Scalability
	Stateful Springboot App	Greater Modularity
	PWD password encryption on the code	Greater Resilience
	No resiliency, Tightly coupled	Loose Coupling
Integrations	No SSO, Cert & Secret as a Service Integration	Integration with SSO, Cert & Secret as a Service

Some Observations

	Observations	Recommendations
Documentation	Detailed Documentation is available across Confluence, however, the practical implementation of that information will require **multiple iterations** of effort to make it directly usable by migration teams, e.g. exact parameters, how to, whom to ask	Cheat-sheets, common scenarios walkthrough, reusable how-to guides.
Iterative builds & deployments for error handling	Changes suggested by Prodevans team worked upon by developers, builds tested & checked for error.. Deploy, check log file, missing components**, build again, repeat**. Access request raised after this process which could have been in parallel. SSO iterations **over 8 days to get the parameters correct** for SSO request	Cheat-sheets, reusable code examples in Repo, common scenario walk thoughts, reusable how to guides.
Session State Logic	**Contour existing Session state logic to what is required for OCP**. Evaluate solutions to ensure that transactions are not lost (store in external database, store session information in cookies). Decide an approach, PoC, fail, rinse, repeat.	Reusable patterns & code samples for common state management, attached services, circuit breaker & related required changes.
App Stack	Most applications rely heavily on **MQ or solutions like NDM**. A solution needs to be found to being them into the OCP solution offering	Explore the viability of a container solution for NDM/MQ dependent critical application such as Flexcube. Ex: Containerize NDM, RabbitMQ etc
Migration process	Dependent on **many requests** to other teams. Despite best endeavors, **SLAs require escalation.** Even though the processes are automated, there **are multi-level approvals** that cause delays. Approvals need to be followed up on.	Deep dive into the relevant processes where opportunities for optimization may exist, & where there may be benefits from future automation.
DevOps Maturity	**Developer teams are often still following pre-cloud native patterns**. Their applications are not optimal for orchestrated containerized deployment. At an operational level developers need to understand how to create Docker images, deploy & test in OCP environment.	Pre-requisite Cloud Native training for developer teams, e.g. 12 Factor for Kubernetes, which is incredibly relevant to gain the benefits of the OCP platform.

Challenges in Container Adoption Journey

Challenges

- Complex Landscape of Technologies, Platforms
  - Unsupported technologies & legacy applications
  - Needs a thorough sifting & analysis to pick the right apps
  - Starting right is half the battle won!
- Divergent alignment with Business
  - Requires negotiation & persuasion
  - Needs agreement on outcomes with all stakeholders
  - Need acceptance of a common vision
- Deliverables and timelines
  - Plethora of teams working together
  - Prioritization is key
  - Need to keep everyone abreast of progress & challenges
- Cultural Issues
  - Processes & People
  - Everyone wants to see benefit

OpenShift skills path

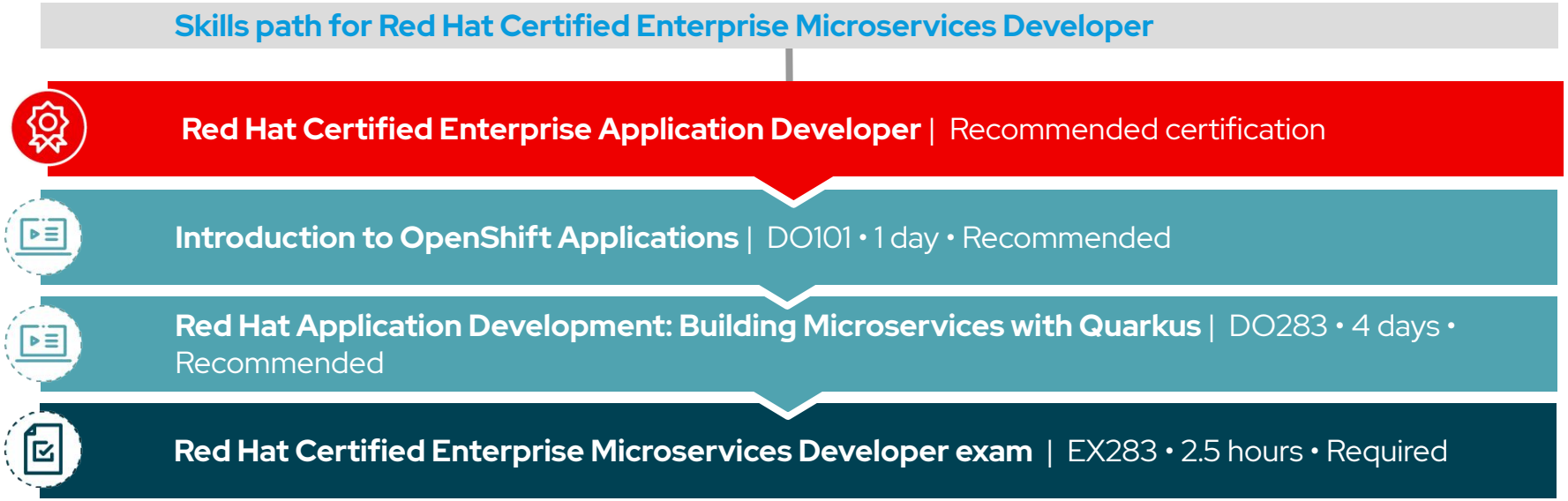
OpenShift Skill Development

Certification | Course | Exam | Required | Free | Recommended

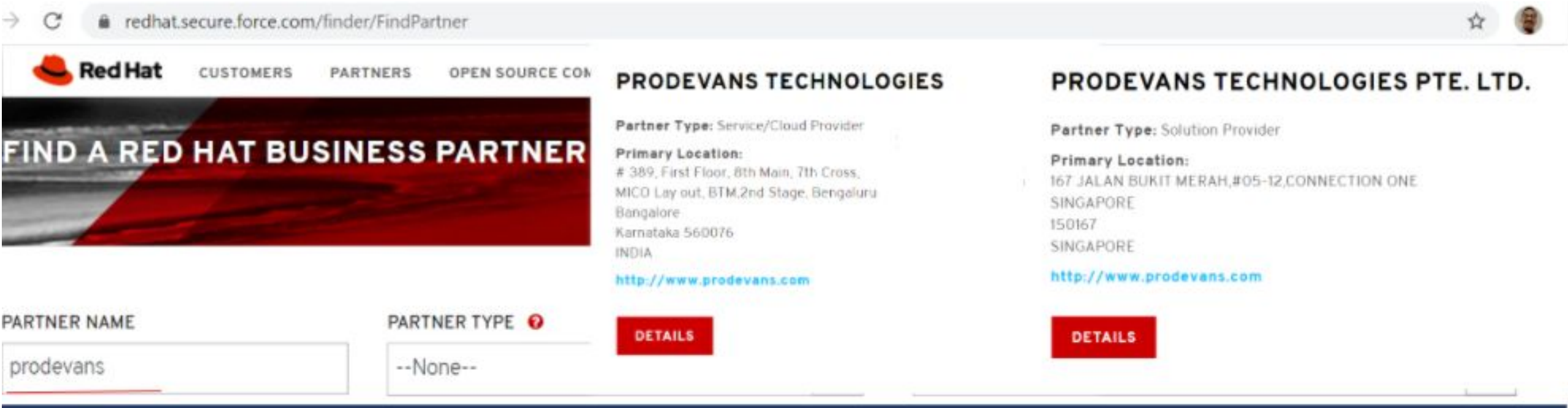
OpenShift Skill Development

Certification | Course | Exam | Required | Free | Recommended

OpenShift Skill Development



Certification | Course | Exam | Required | Free | Recommended



India Business Center

Bangalore
Building # 403 ,
5th Floor, Saket Callipolis
Sarjapur Main Rd, Rainbow Drive,
Doddakannelli
Bengaluru – 560035.
Phone: +919902991978
Email: ask@prodevans.com

APAC Office

Singapore
167 JALAN BUKIT
MERAH #05-12
CONNECTION ONE
SINGAPORE, 150167

Business Continuity Center

Hyderabad
Office #422, Manjeera Majestic,
JNTU Road, Kukatpally,
Hyderabad-500072
Phone: +91 040 66773365

USA Sales Office

USA
5164, Madison Avenue,
C02, Okemos,
Michigan – 48864
Phone: +1 (513) 394-1287

Lets Connect...