Hewlett Packard Enterprise

HDFC DevOps Journey

An Approach Paper

Typical customer container journey

Learn and Evaluate

- Understand container implications
- Align containers with business objectives & metrics
- App transformation impacts
- App lifecycle processes

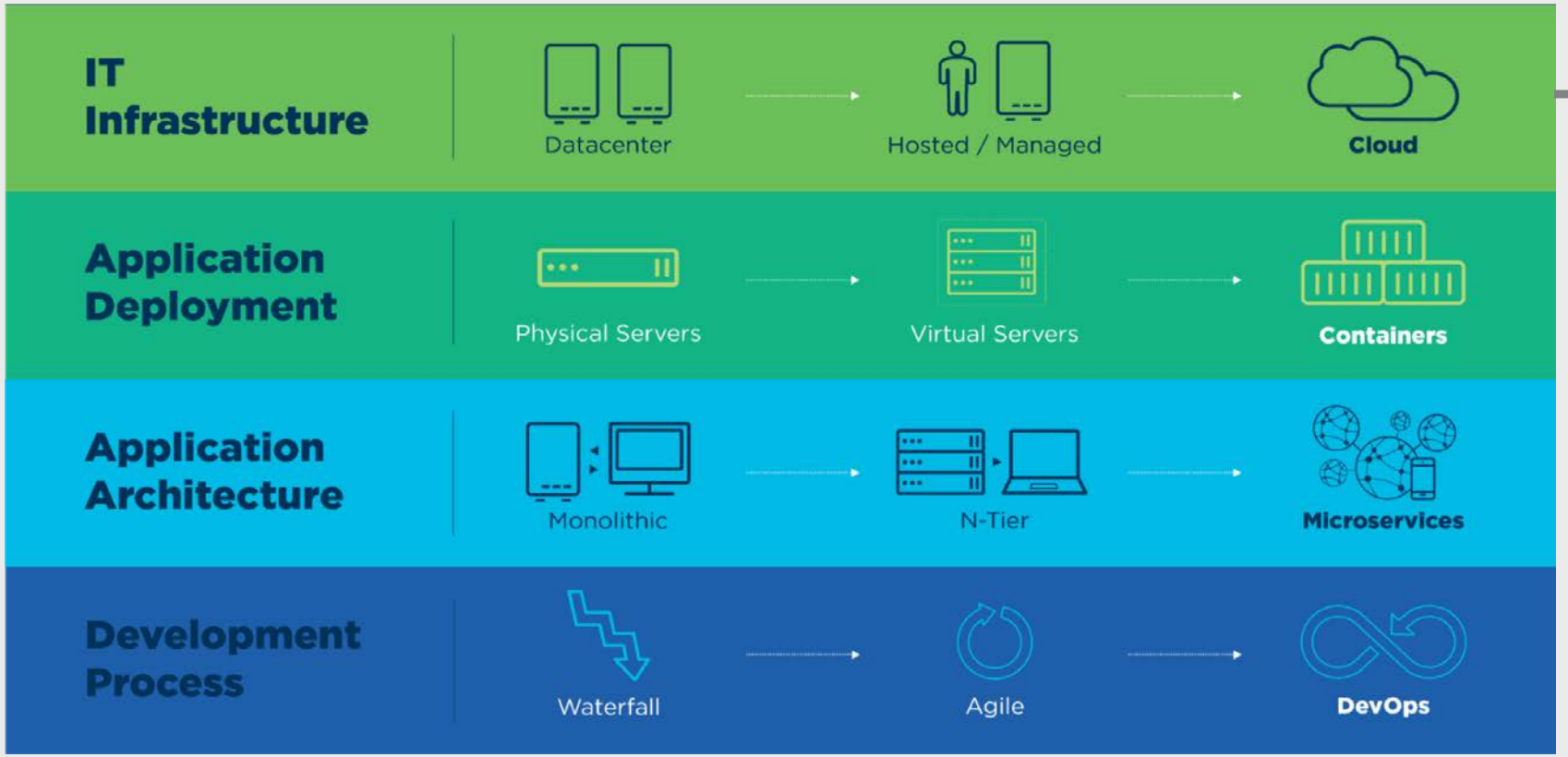
Pilot and POCs

- Optimal architecture & platform choices
- Properly re-architect existing app for container
- Optimal tool choices for ecosystem: orchestration, development
- Integrate into existing dev process

Production deployment

- Deploy containers at scale
- Integrate containers with existing infrastructures
- Address security, HA, **persistent storage**, networking, life cycle, and management

Target state for IT based on current business demands



On HPE Synergy

Project Phases

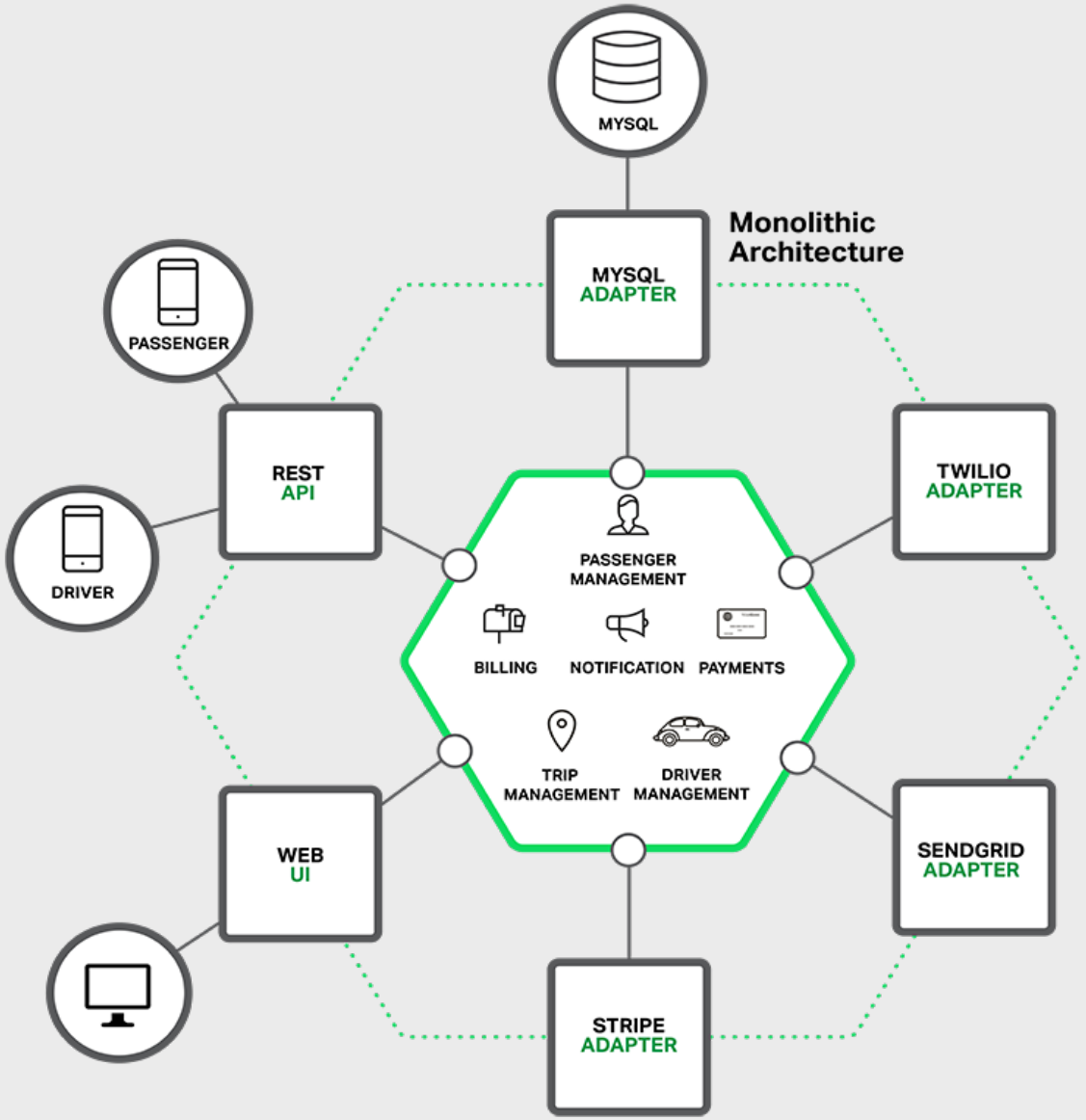
Containerizing - Phase 1

- Identify the apps to be containerized
- Find reliable base images
- Create Dockerfile
  - Setting up Environment Variables
  - Copy Code into container images
  - Create Links between services(containers)
  - Define Entrypoint Command/scripts
- Running application in Kubernetes environment

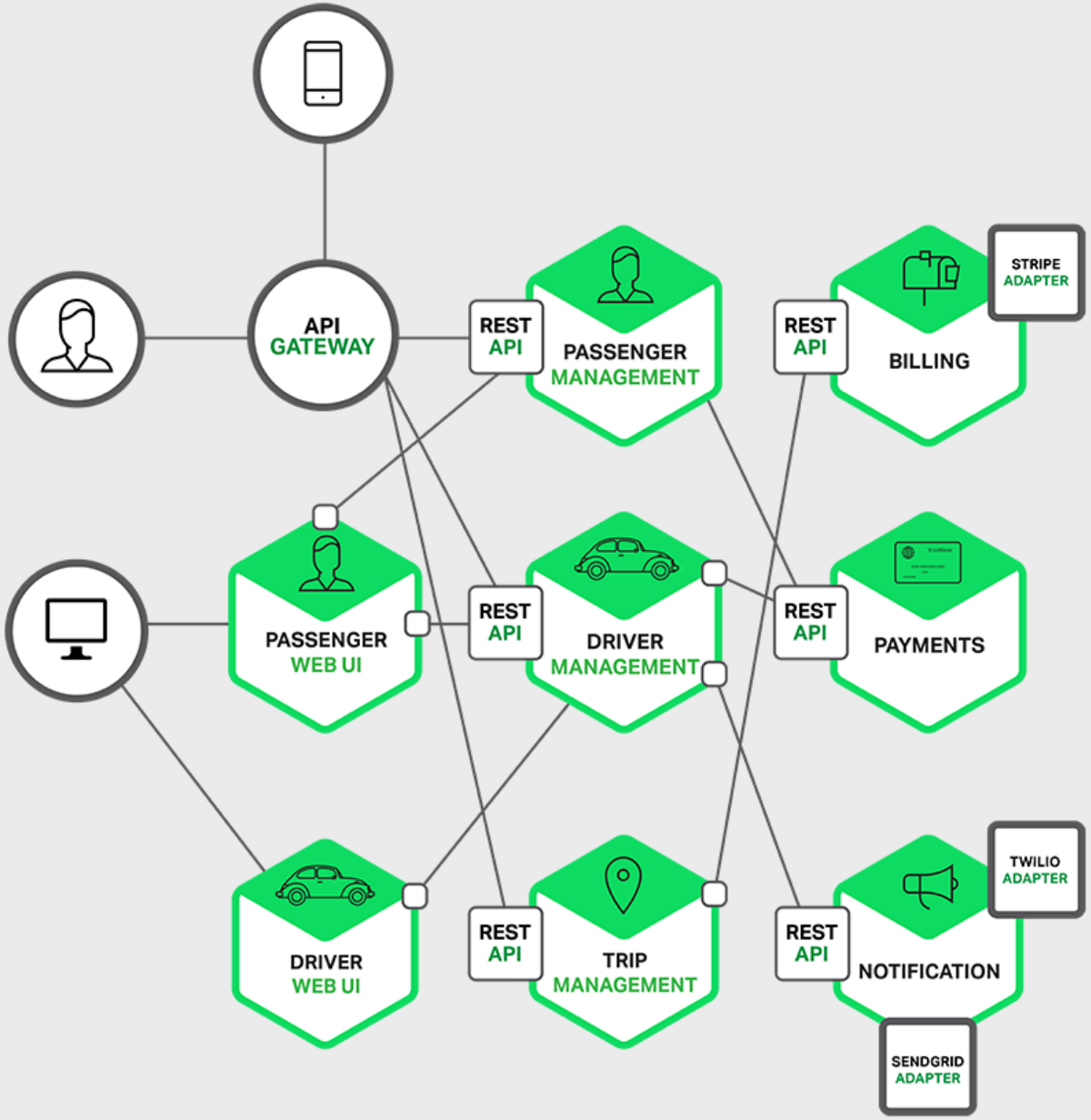
Microservices – Phase 2

- Refactor monolithic application into micro services.
- Adopt some of the key factors from “12 factor app” pattern for building micro services.

Sample Monolithic App

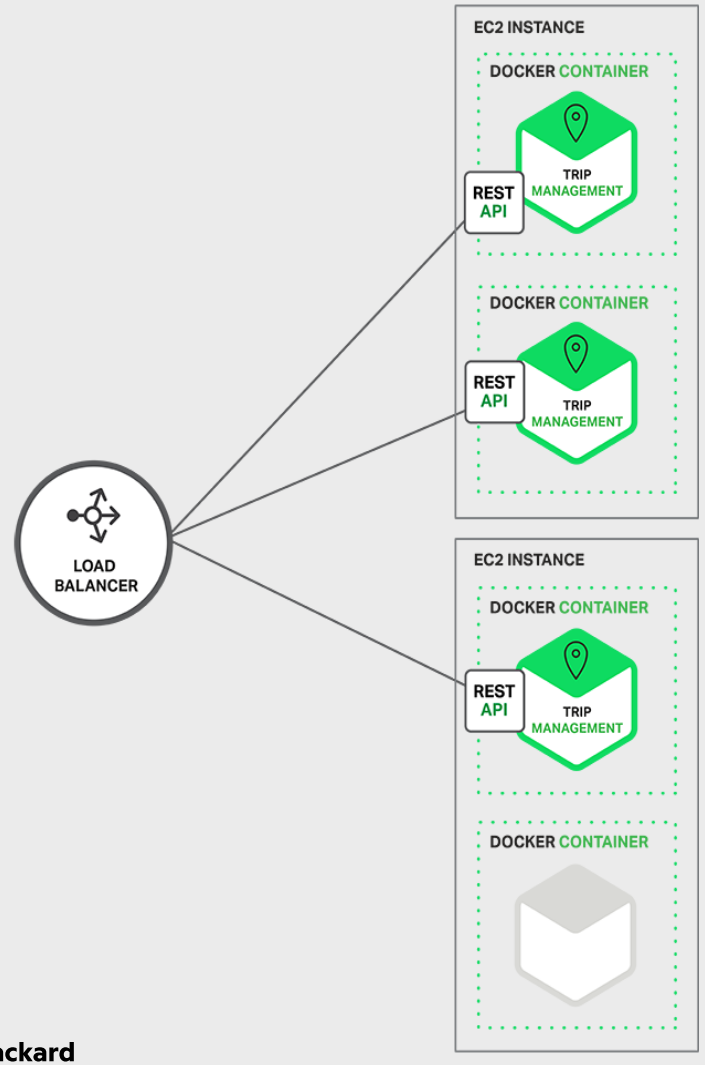


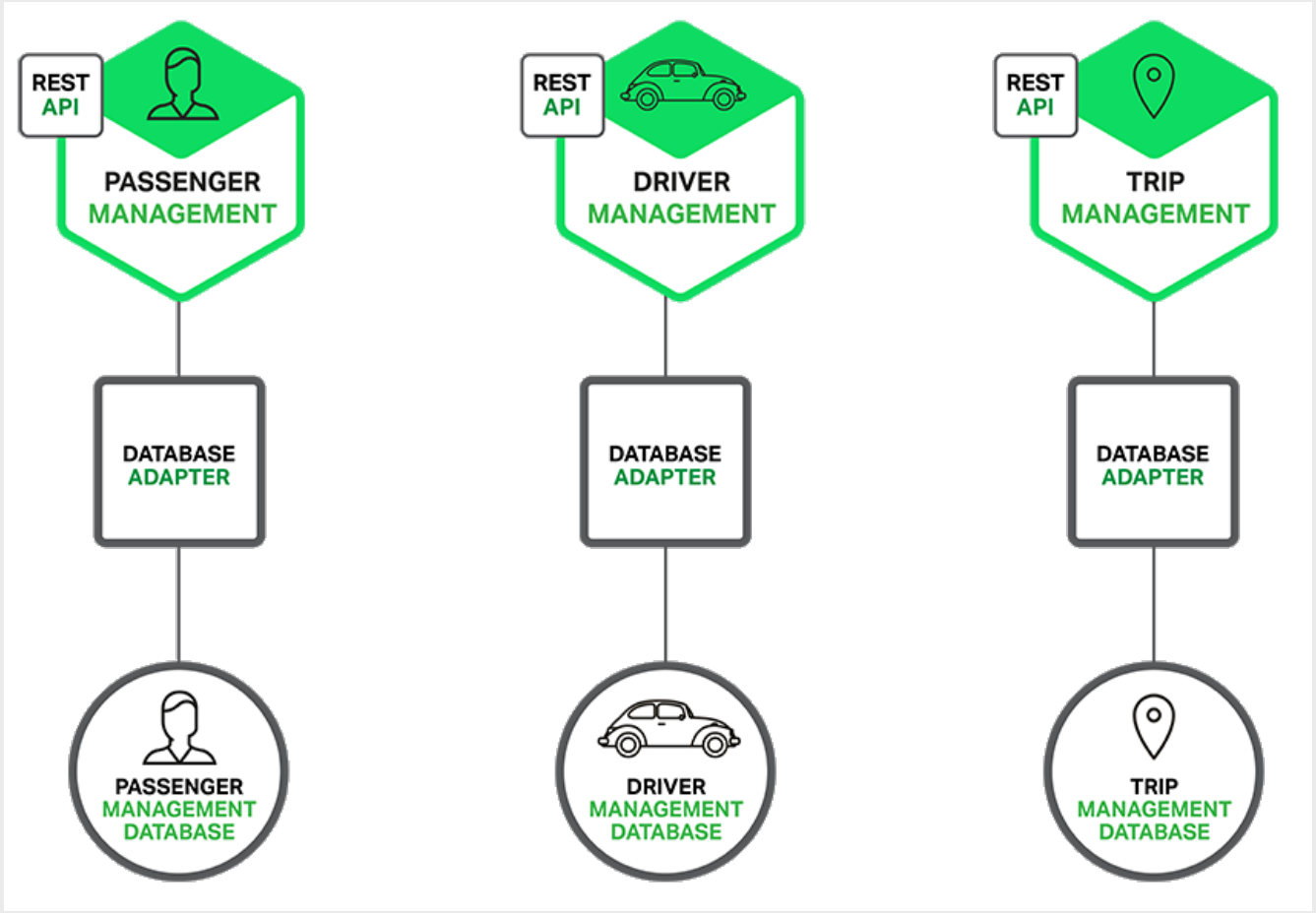
Decomposition of the Monolithic App to microservices



Hewlett Packard
Enterprise

Showing scale and decentralizing the data (loose coupling)



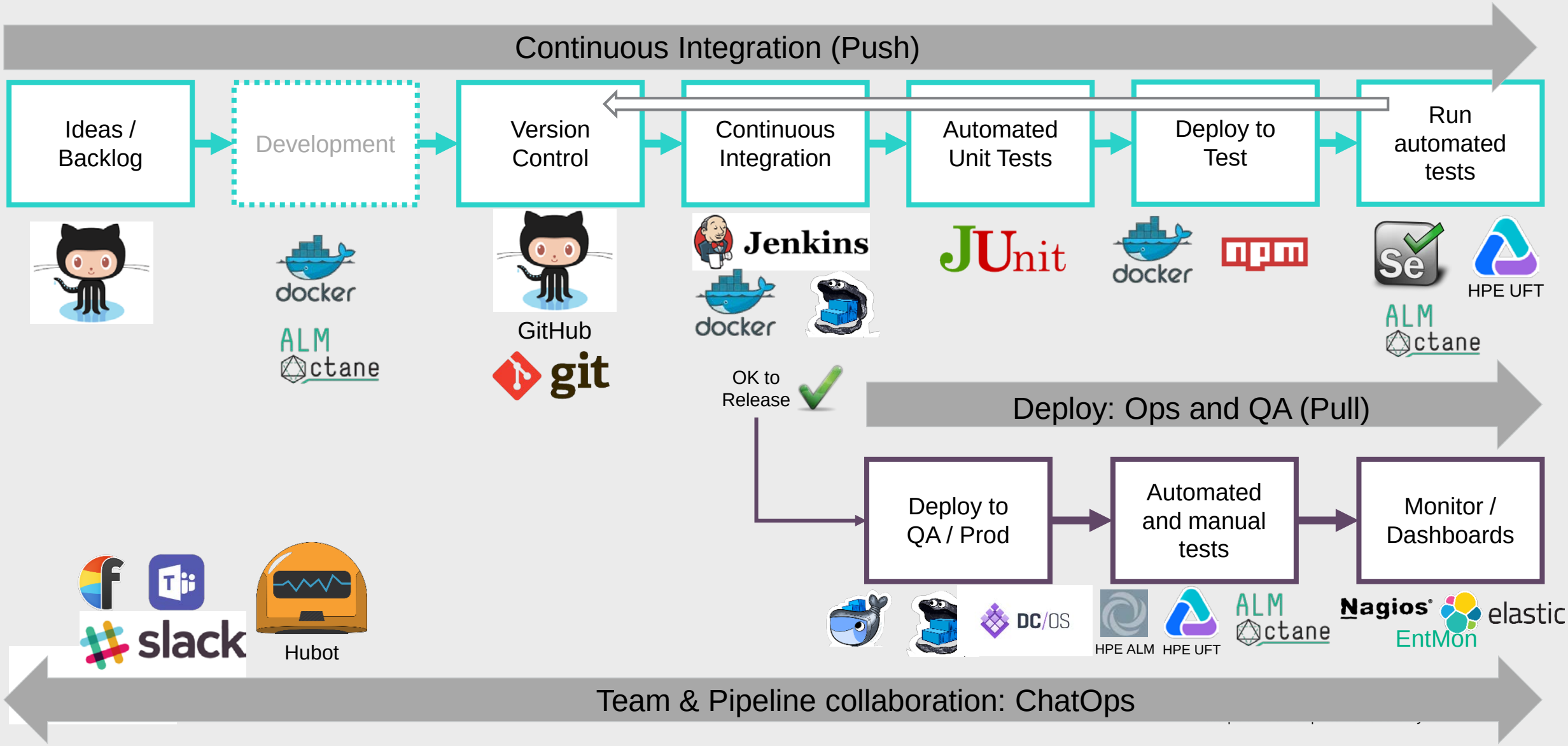


Hewlett Packard Enterprise

CI/CD – Phase 3 (low priority due to lack on automated tests)

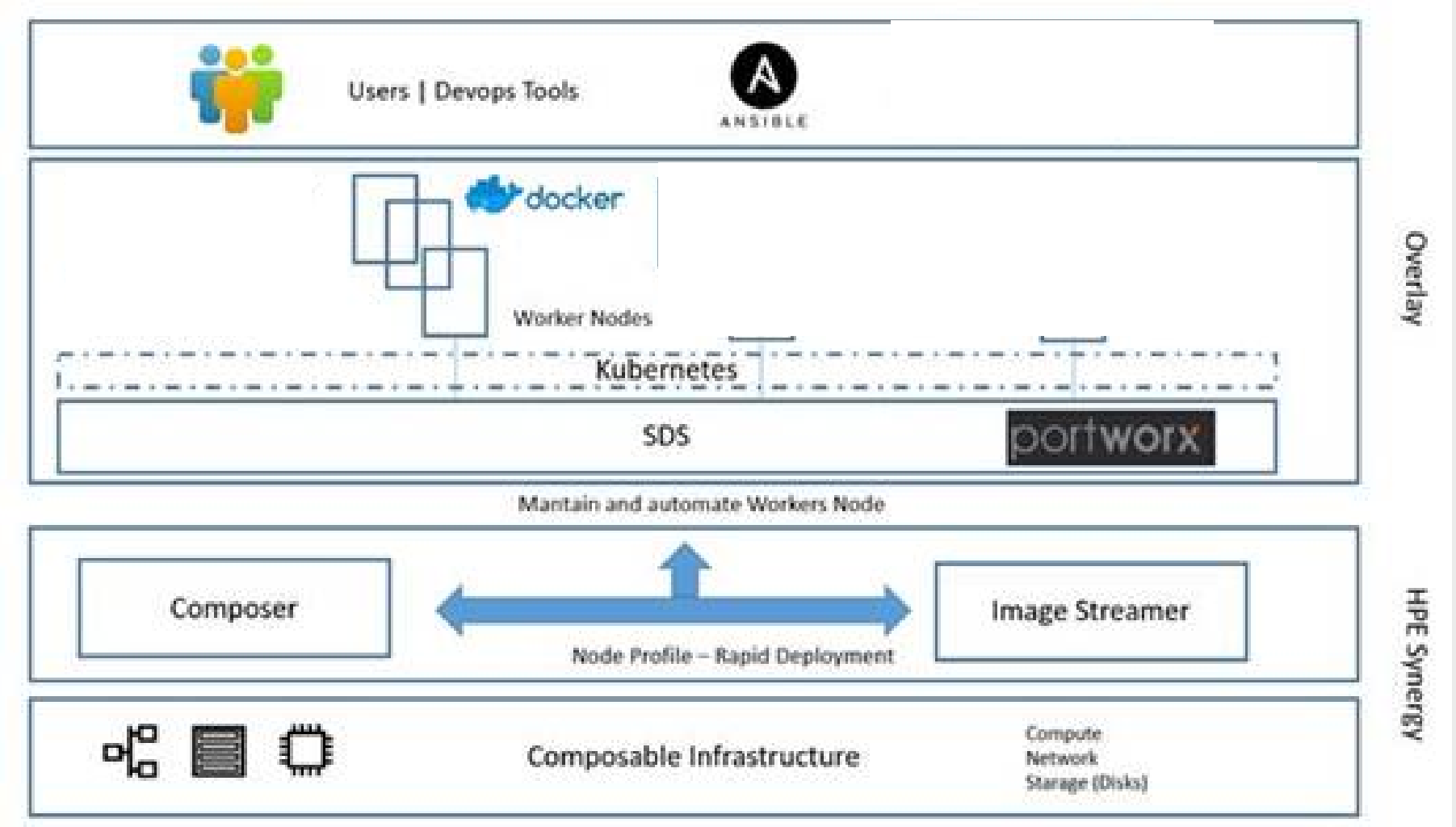
- Continuous Development
- Continuous Integration
- Continuous Testing
- Continuous Monitoring
- Continuous Delivery

Sample CI/CD Pipeline Management

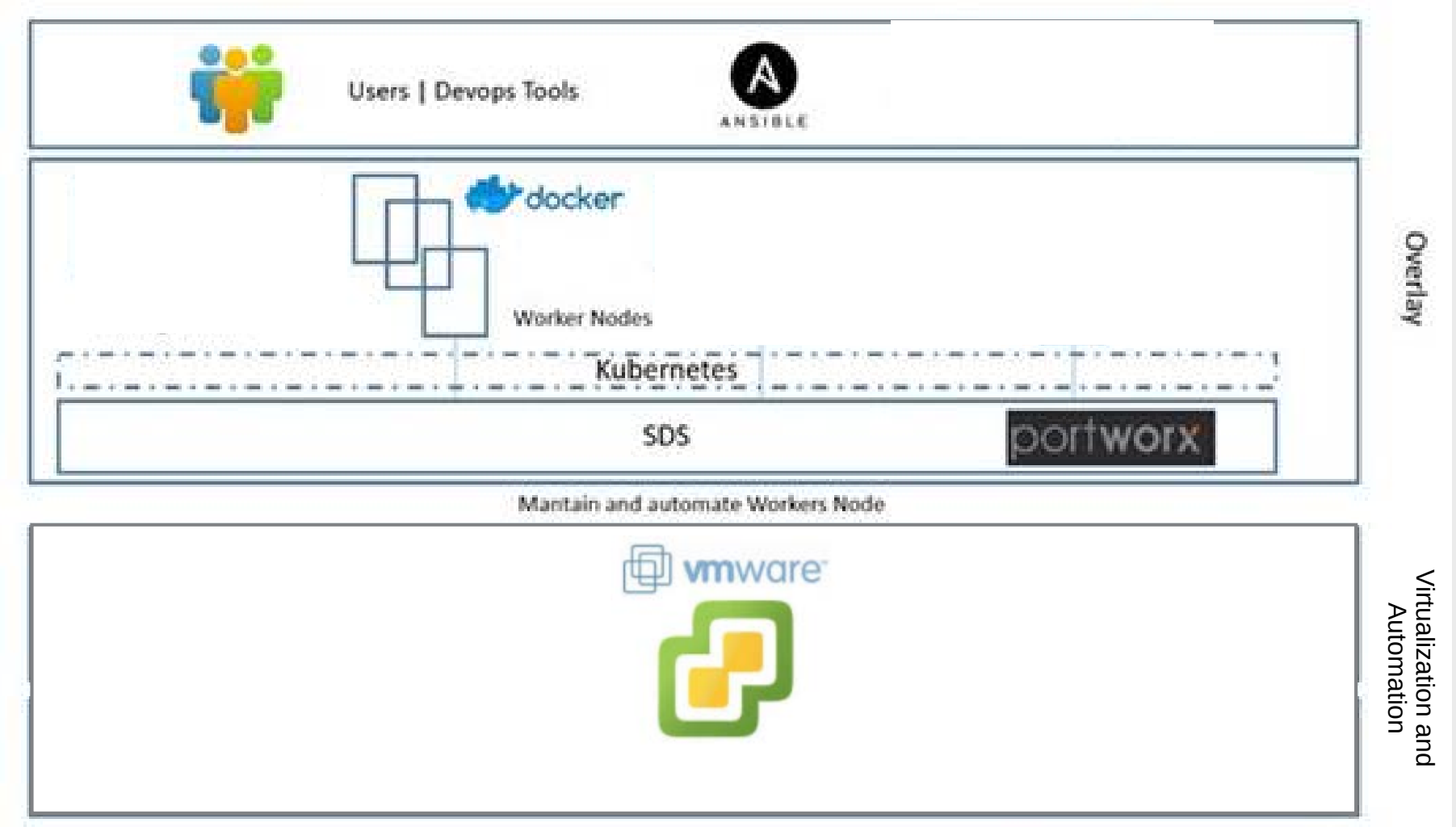


Container-as-a-service Infrastructure

Architecture for CaaS (Production Env)



Architecture for CaaS (PoC)



CaaS Usecases

- From Developer Perspective
  - Upload a container to Container Registry.
  - Deploy a container.
  - Deploy micro-services based application (having several services as containers).
  - Manage life-cycle of that application (scale/stop).
  - Monitor the application.
- From Admin Perspective
  - Create the initial container cluster (k8s)
  - Scale the cluster according to the needs.
  - Monitor the infrastructure
- Synergy Automation using Ansible and Image-Streamer
- Portworx use-case for persistent storage

HPE Value Added in the area of DevOps

HPE Synergy differentiation

Deeper integration between Infrastructure as code and Mesosphere DC/OS

DC/OS with Synergy - Solution Building Blocks

Only HPE

App Containers

Apache HTTPD | Tomcat | Postgres DB | NGINX

Platform Services

Mesosphere Enterprise DC/OS | **SDS**

Composable Infrastructure

HPE Synergy 480 Compute Modules

HPE Synergy D3940 Storage Module

Synergy Composer

HPE Synergy Composer

HPE Synergy Image Streamer

Synergy Composer

- Rapid deployment of Compute Modules to DC/OS cluster.

Synergy Image Streamer

- Automated OS deployment

SDS

- Software Defined Storage capabilities for supporting containers persistence.

Mesosphere DC/OS

- Container Platform Services

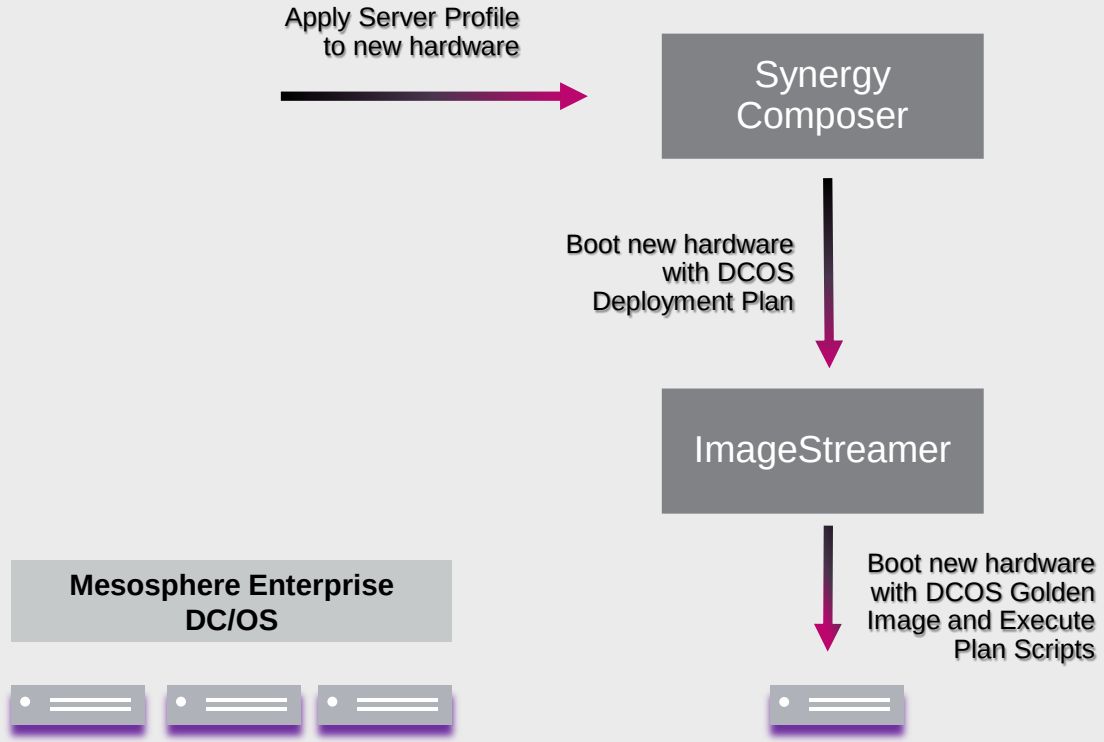
Hewlett Packard Enterprise

Only HPE

Composable and Enterprise DC/OS

Technology that powers your transformation to Hybrid IT

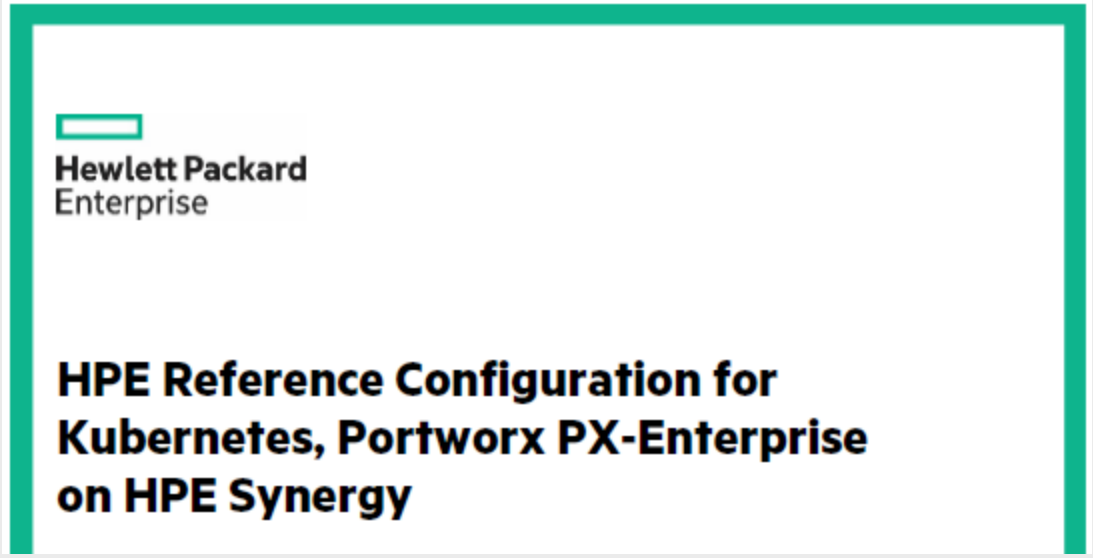
- **Accelerate time-to-value**
  - Automatically provision entire stack from bare metal through application in minutes
- **Increase reliability**
  - Reduce opportunity for error and streamline compliance auditing through consistent provisioning
- **Deliver deployment flexibility**
  - Provision and update bare metal DC/OS nodes with automation tools in the same way as virtual and cloud resources



Hewlett Packard Enterprise | MESOSPHERE https://mesosphere.com/blog/hpe-synergy-mesosphere-dcos/

HPE Synergy differentiation

Deploying Docker Containers at Scale using HPE Synergy, Kubernetes and Portworx



Deploying Docker Containers at Scale with persistent storage

HPE Synergy

Composable Infrastructure

- Compute and Storage
- Fabric

Rapid image-based deployment

- Composer with embedded OneView
- Image Streamer

Kubernetes

Container Orchestration

- Optimized container placement across a cluster of servers
- Applications availability is managed

Built-in Services and Load Balancing

- Services like IP addressing, DNS, and Load-balancing provided into apps

Self-healing

- Containers replaced and rescheduled based on server and app-level checks

Portworx

Elastic Container Storage

- Increase number of stateful containers per server (example Cassandra rings)
- Enables HA for any stateful application

Container Data Lifecycle Management

- Volumes automatically get snapshots, encryption, and backup to object stores

Scheduler-complete Integration

- Dynamic storage provisioning with automatic hyper-convergence

Hewlett Packard Enterprise

kubernetes

portworx

Compose Kubernetes HA master nodes on HPE Synergy (VMs)

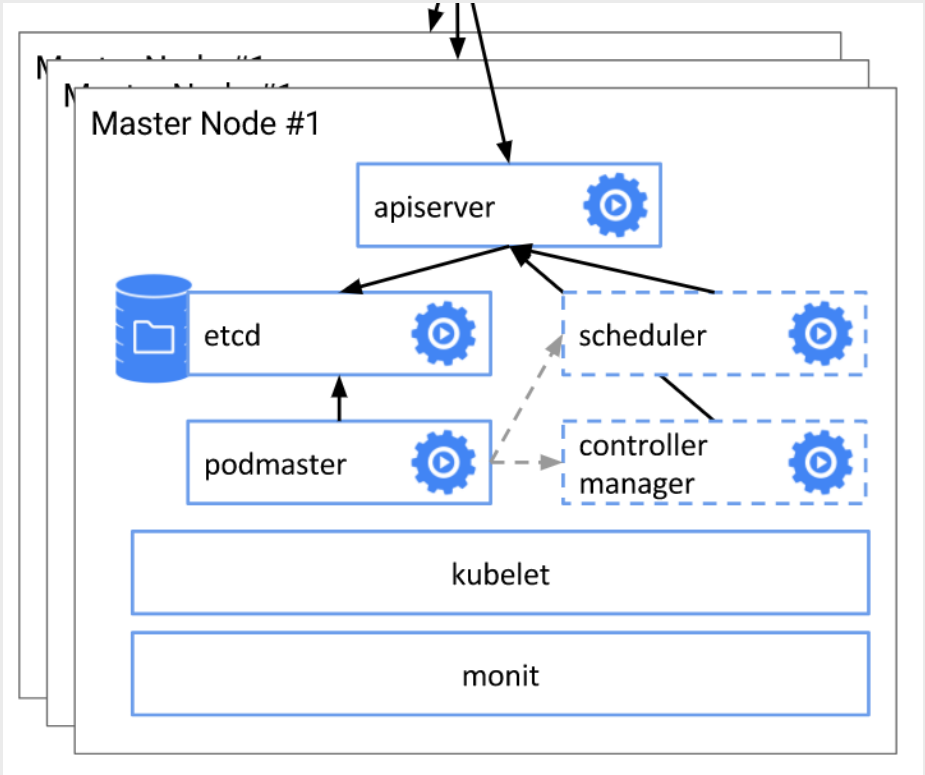
Script VM

```
2. 10.60.70.225 (root)
[root@rhel-ansible-engine kubernetes_portworx]# ls -l
total 32
-rw-r--r--. 1 root root  982 Oct 24 19:03 bootstrap_kubernetes_HA_master_nodes.yml
drwxr-xr-x. 5 root root  127 Oct 24 18:55 config
-rw-r--r--. 1 root root  900 Oct 24 19:07 configure_worker_nodes.yml
drwxr-xr-x. 2 root root   30 Oct 24 21:18 data
drwxr-xr-x. 2 root root   52 Oct 24 18:55 dependencies
-rw-r--r--. 1 root root   13 Oct 26 08:04 deploy_cassandra.retry
-rw-r--r--. 1 root root 1651 Oct 26 05:36 deploy_cassandra.yml
-rw-r--r--. 1 root root  802 Oct 24 18:55 deploy_mysql.yml
-rw-r--r--. 1 root root 1142 Oct 24 18:55 deploy_portworx.yml
drwxr-xr-x. 4 root root   36 Oct 24 18:55 examples
-rw-r--r--. 1 root root 1351 Oct 24 18:55 README.md
drwxr-xr-x. 2 root root 4096 Oct 26 07:44 tasks
drwxr-xr-x. 2 root root   27 Oct 25 05:40 vars
[root@rhel-ansible-engine kubernetes_portworx]# ansible-playbook -v bootstrap_kubernetes_HA_master_nodes.yml
```

Ansible bootstrap master nodes

Automates master nodes deployment

Configures Docker, Kubernetes services



Kubernetes HA master nodes

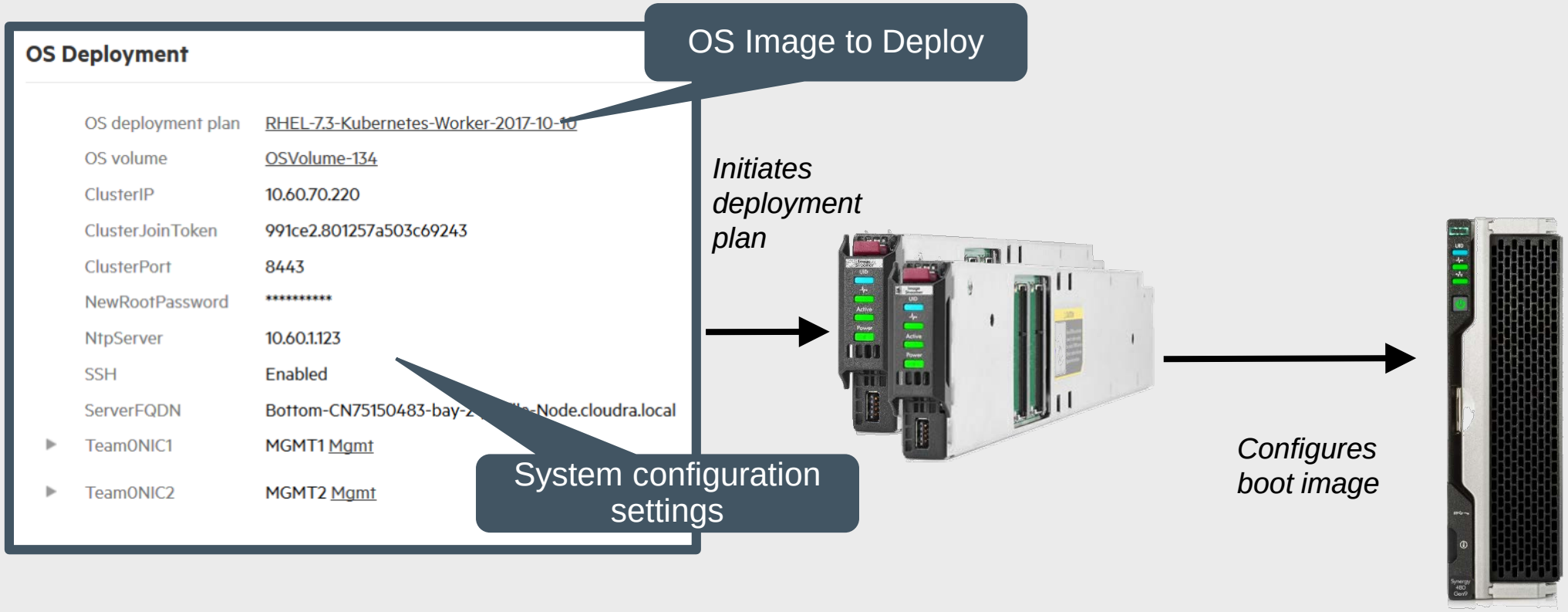
Ansible bootstraps master nodes, configures Docker and Kubernetes services. Configures load balancer and dashboard

Script VM

Bootstraps Kubernetes HA master nodes using Ansible playbooks

Hewlett Packard Enterprise

Compose Kubernetes worker nodes on HPE Synergy



HPE OneView

Selects a deployment plan
Sets personalization parameters
Provisions physical infrastructure

HPE Synergy Image Streamer

Creates bootable OS baked with Kubernetes services and Docker image Personalizes OS and Kubernetes and Docker per deployment plan

HPE Synergy Compute and Storage

Compute node boots directly into a running OS

Hewlett Packard Enterprise

Automate the configuration of persistent volumes using Portworx on Synergy worker nodes

Script VM

```
2. 10.60.70.225 (root)
[root@rhel-ansible-engine kubernetes_portworx]# ls -l
total 32
-rw-r--r--. 1 root root  982 Oct 24 19:03 bootstrap_kubernetes_HA_master_nodes.yml
drwxr-xr-x. 5 root root  127 Oct 24 18:55 config
-rw-r--r--. 1 root root  900 Oct 24 19:07 configure_worker_nodes.yml
drwxr-xr-x. 2 root root   30 Oct 24 21:18 data
drwxr-xr-x. 2 root root   52 Oct 24 18:55 dependencies
-rw-r--r--. 1 root root   13 Oct 26 08:04 deploy_cassandra.retry
-rw-r--r--. 1 root root 1651 Oct 26 05:36 deploy_cassandra.yml
-rw-r--r--. 1 root root  802 Oct 24 18:55 deploy_mysql.yml
-rw-r--r--. 1 root root 1142 Oct 24 18:55 deploy_portworx.yml
drwxr-xr-x. 4 root root   36 Oct 24 18:55 examples
-rw-r--r--. 1 root root 1351 Oct 24 18:55 README.md
drwxr-xr-x. 2 root root 4096 Oct 26 07:44 tasks
drwxr-xr-x. 2 root root   27 Oct 25 05:40 vars
[root@rhel-ansible-engine kubernetes_portworx]# ansible-playbook -v deploy_portworx.yml
```

Ansible configures portworx volumes

Automates deployment of Portworx

Configures Portworx volumes

Script VM

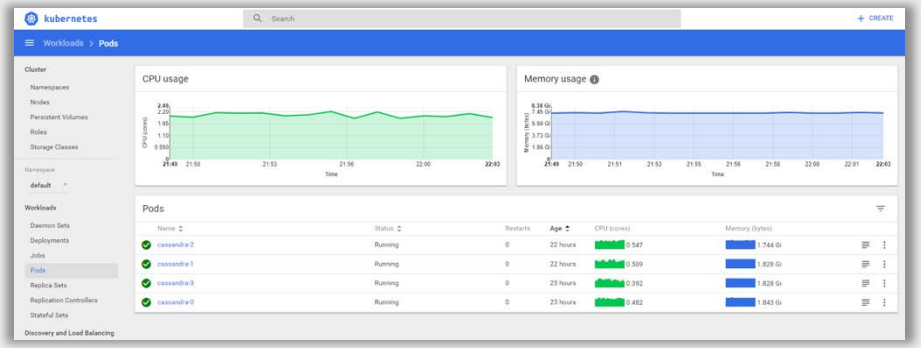
Bootstraps Portworx using Ansible playbooks

Synergy worker nodes

- Nodes are assigned with disks from D3940.
- Ansible bootstraps Portworx installation.
- Portworx persistent volumes will be created with Kubernetes deployment of Cassandra.

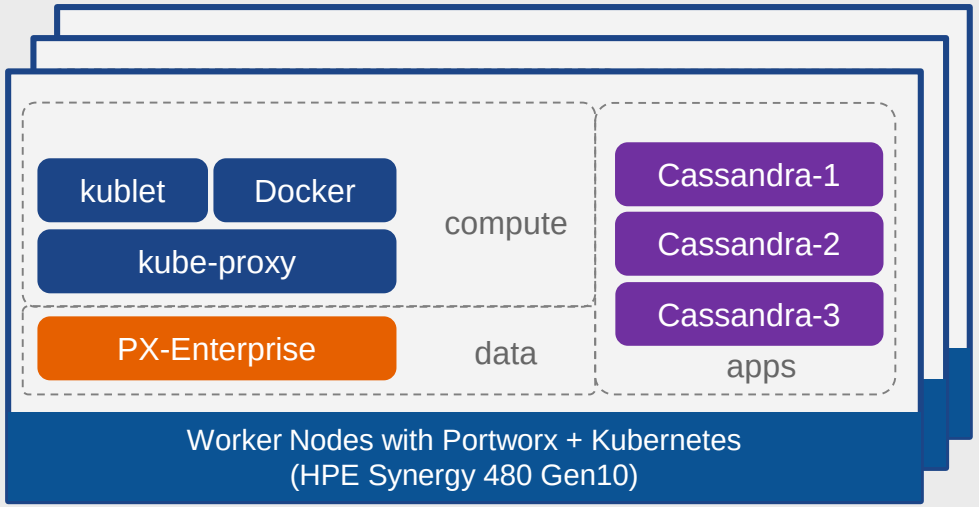
Hewlett Packard Enterprise

Deploying Cassandra with Kubernetes and volumes from Portworx



Cassandra deployed and managed as Stateful Set and visible in Kubernetes web console

Portworx volumes dynamically created with application deployment and visible from management console



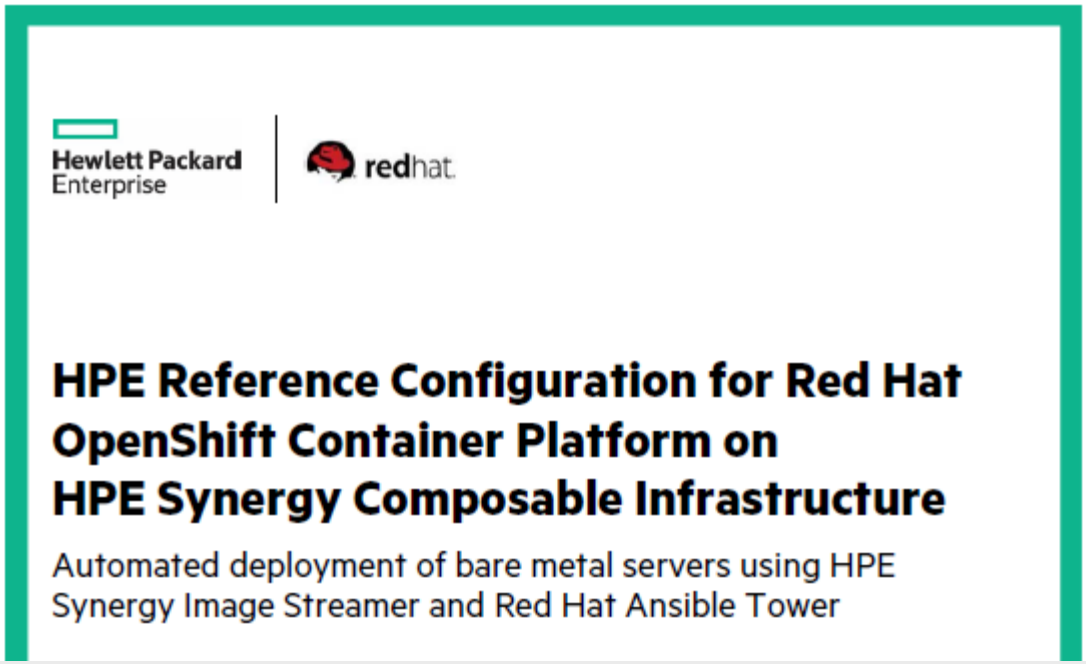
Synergy worker nodes

Every server is capable of running many Cassandra rings with compute containerization from Kubernetes and storage containerization from Portworx

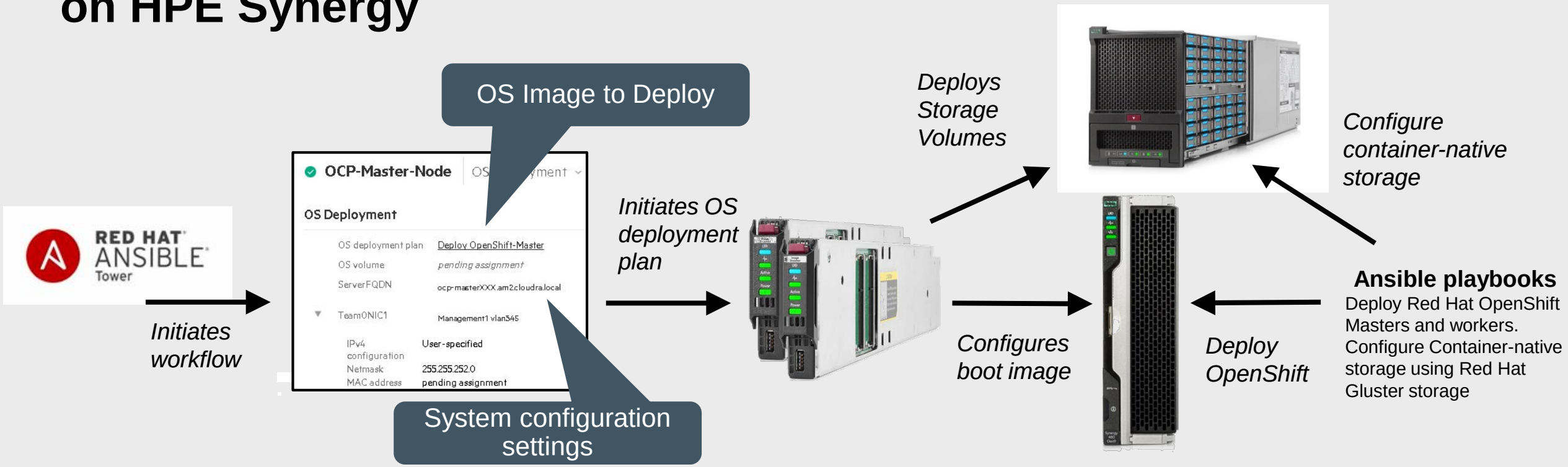
Hewlett Packard Enterprise

HPE Synergy differentiation

Deploying RedHat OpenShift at Scale on HPE Synergy



Automate deployment of Red Hat OpenShift Container Platform on HPE Synergy



Red Hat Ansible Tower

Workflow runs playbooks to deploy Red Hat OpenShift on HPE Synergy using Ansible Modules for HPE OneView

HPE OneView

Server profile template identifies the networks, storage, and deployment plan
Sets personalization parameters
Provisions physical infrastructure

HPE Synergy Image Streamer

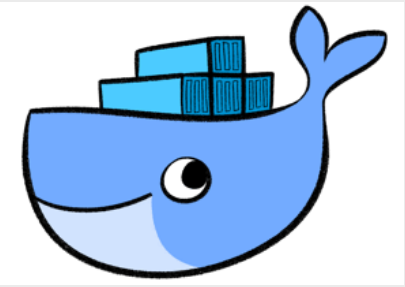
Creates Red Hat Enterprise Linux 7.4 bootable OS
Personalizes OS and prepares for Red Hat OpenShift per deployment plan

HPE Synergy Compute and Storage

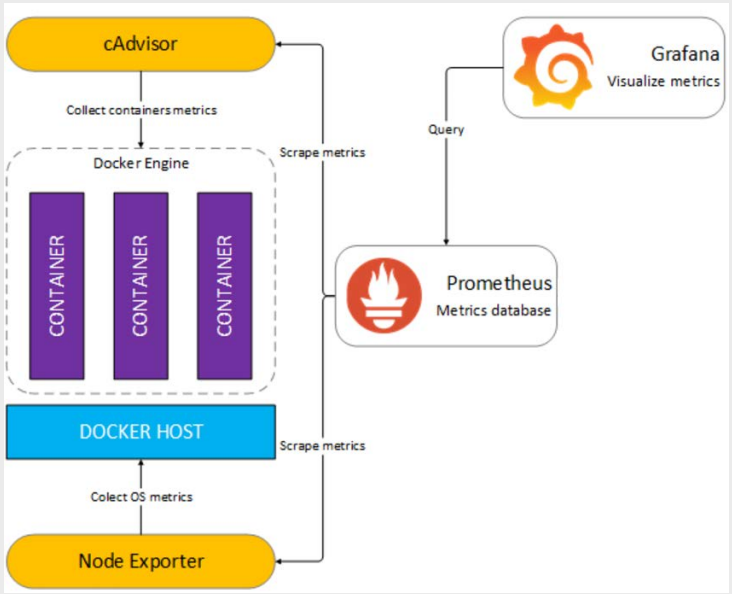
Compute node boots directly into a customized running OS ready for Red Hat OpenShift deployment

Hewlett Packard Enterprise

Docker Monitoring



Overview



Solution Components

cAdvisor

Node Exporter

Project Flow

- Setup Docker Data Center (“DDC”)
- Manual Installation
- Automation
- Documentation

Hewlett Packard Enterprise

Grafana dashboard

Embrace open standards & best-of-breed infrastructure and partners

DevOps/Automation — Linux, Microsoft, CHEF, ANSIBLE, puppet

Container Security — docker, MICRO FOCUS, aqua

Container Ecosystem — docker, MESOSPHERE, kubernetes, turbonomic, RED HAT OPENSHIFT Container Platform

Operating Systems — SUSE, Microsoft, ubuntu, redhat

ProLiant BladeSystem — **Servers**

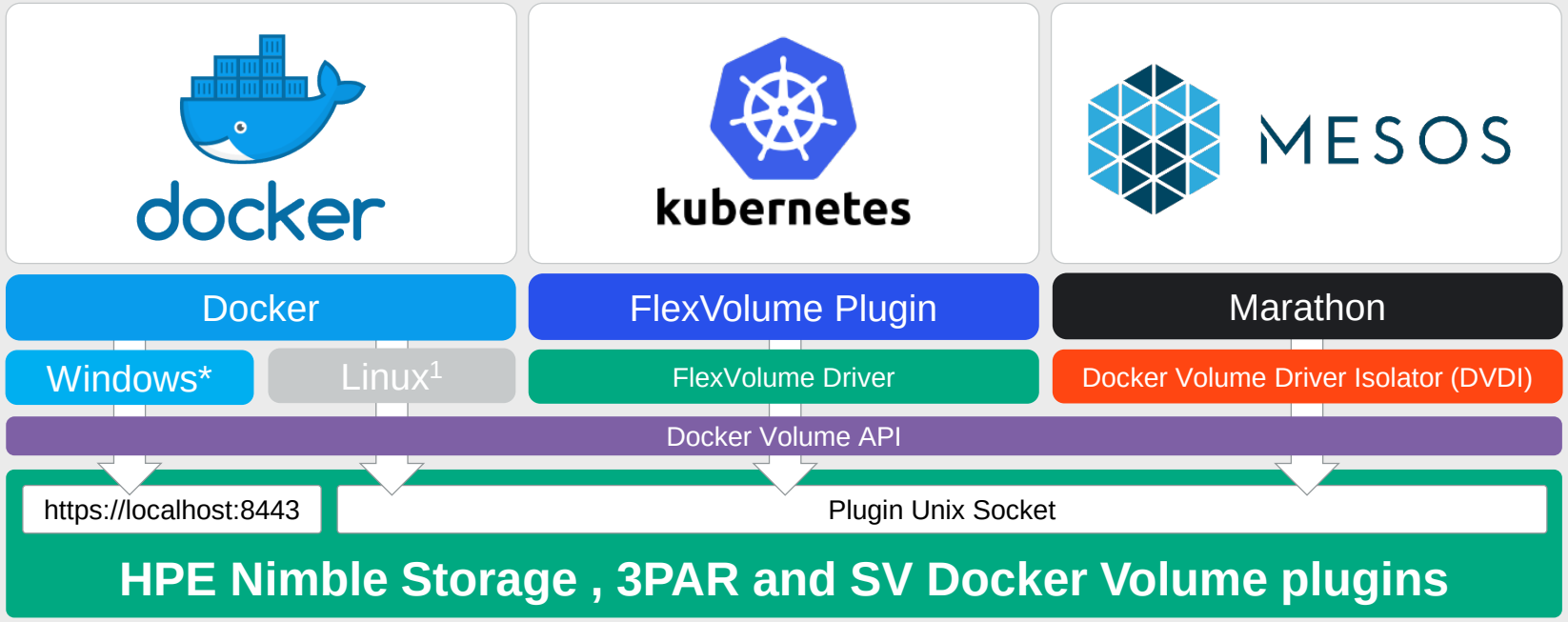
Nimble 3PAR — **Storage**

Converged Hyper Converged — **Integrated**

Synergy + 3PAR — **Composable**

HPE POINTNEXT & HPE Partners

HPE Persistent Storage Platform for Containers

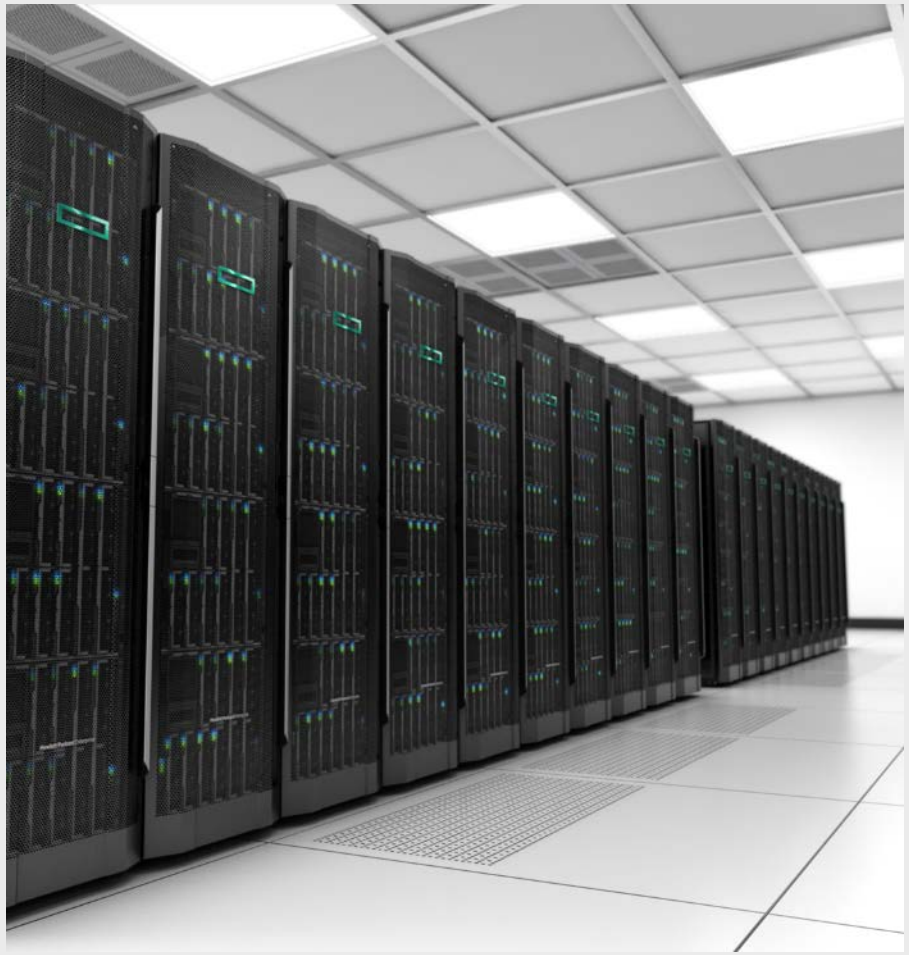


* = Windows plugin only available for Nimble Storage
[1] = Docker Certified and standalone plugin available

Why HPE?

Harness the full value of cloud and containers with HPE

- Comprehensive solutions with hardware, software, services, and support
- Accelerate container service deployments – Power-on to production ready in minutes - not weeks or months
- Accelerate app development and improve quality, portability, and performance
- All-in-one compute, software-defined storage, and intelligent integrated infrastructure
- Broadest range of solutions supporting enterprises, LOB to remote deployments



How do you get started?

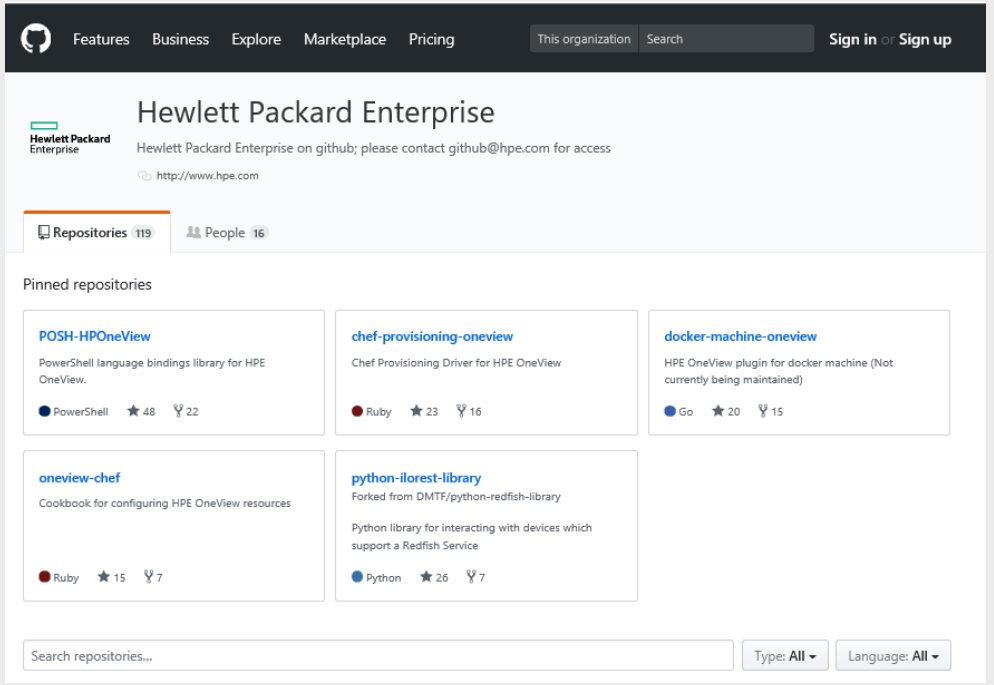
- Start with HPE Best Practice, Deployment Guides & RAs
- Mix bare metal, hypervisors, & containers in one environment
- Manage with HPE OneView
- Six-9's of availability with InfoSight Predictive Analytics
- Built-in backup, security, efficiency with: HPE Persistent Storage & HPE Simplivity
- Integration with:

docker

OPENSHIFT

kubernetes

MESOSPHERE



https://github.com/HewlettPackard/

Management Software:
HPE OneView

Converged Solution:
HPE CA700

Composable:
HPE Synergy

Docker Ready Servers:
HPE ProLiant Servers

Persistent Storage:
HPE 3PAR & HPE Nimble Storage

24x7 HPE Support & Services

References

- For more information:

- HPE Reference Configuration for Kubernetes, Portworx PX-Enterprise on HPE Synergy
  - https://h20195.www2.hpe.com/V2/GetDocument.aspx?docname=a00039700enw
- GitHub link for Ansible integration
  - https://github.com/HewlettPackard/K8s-Portworx-Synergy
- Kubernetes Persistent Storage on Portworx
  - https://portworx.com/use-case/kubernetes-storage/
- Docker Storage on Portworx
  - https://portworx.com/use-case/docker-persistent-storage/
- Persistent Storage for DC/OS on Portworx
  - https://portworx.com/use-case/persistent-storage-dcos/

HPE Reference Architectures (RA)

Best practice guidance for deploying Mesosphere on HPE

- HPE and Mesosphere Simplify Container Management for Modern Data Centers Using Open Source DC/OS
- HPE 3PAR StoreServ with Docker and Mesosphere Enterprise DC/OS
  - Public URL: http://h20195.www2.hpe.com/V2/GetDocument.aspx?docname=a00000412enw

NEXT STEPS

- Identify two apps to containerize – HDFC (preferably non windows apps to begin with)

- HPE to propose the DevOps journey for HDFC
  - Phase 1: Containerize
    - Containerize existing apps
    - Build CaaS Infrastructure (preferred on HPE hardware with native Kubernetes. If they need supported Kubernetes platform, they can consider RH OpenShift on Synergy)
    - Automate the infrastructure (as much as possible. show case the composable infrastructure)
    - Deploy containerized apps
  - Phase 2: Refactor apps to adopt microservices architecture (important factors of “12 factor app” guidelines, need not be all factors)
  - Phase 3: Adopt CI/CD pipeline (low priority , no automated tests exist today)

- Provide HPE Synergy PoC environment – HPE
- HPE team to assist HDFC to containerize two apps by working with ISVs (end goal : create templates for rest of the apps)
- Standardize on the host for hosting the docker/containers. HPE recommends physical servers but VMs can used in some cases