Going from OpenShift PoC to Production

Accelerate your path with HPE

Red Hat Summit | May 2018
Agenda

- Bringing containers to production—different adoption paths
- Impact on people, process, and governance
- Technology considerations (including data management and protection)
- HPE Pointnext Services for OpenShift
- Planning for success
OpenShift adoption options
# Four options for container adoption

1. **Deploy containerized commercial apps**
   - Up in days, not months; verified and secure

2. **Containerize monoliths**
   - Migrate to hybrid cloud or bare metal; get better CAPEX/OPEX versus VM

3. **Containerize monolith; transform to microservices**
   - Look for shared services to transform agility, DevOps, distributed architecture

4. **Enable new microservices and apps**
   - Greenfield cloud native or containers as a service (CaaS)
Moving from PoC to production
Key considerations

Complete PoC
Often a minimal viable product (MVP)

People and organization

Dev and release process

Governance

Technology and platform

Production
People, organization, governance
People and organization

Traditional waterfall development model

- Customer or BU
- Dev & QA
- IT Ops

1-3 releases/year
4-12 month cycles

Agile and DevOps model

- Integrated teams

4-12+ releases/year
1-3 month cycles

CAPEX vs OPEX model?
Dev/release process and governance

Command and control

- Request for change, change control board
- Dev controls app stack
- Waterfall model
- Ops owns security and monitoring

Integrated and empowered

- Change record as part of CI/CD pipeline
- Dev controls app image, Ops controls standardized base image via catalogs
- Continuous delivery model
- Dev assumes more control on security and apps performance monitoring
Technology considerations
OpenShift in production

The OpenShift production ecosystem

- High availability
- Lifecycle management
- Orchestration
- Security
- Data protection and management
- Monitoring
- Scaling
- Resource management
Technology considerations

Security

- **Safe images**: Security for private registry (scanning, access control)
  - OpenSCAP scanning (integrate into CI/CD)
- Detailed audit trail for compliance, regulation, and forensics
- Safeguarding sensitive data
- Run-time protection and continuous monitoring
- Harden OS (SELinux mandatory for OpenShift)
- Leverage security context constraints (SCC)
- Strong remediation and alerting

**Securing the stack**

- Container images
- Container registry
- Red Hat® Linux®
- Hardware firmware and BIOS

Lack of education/training for those involved in software development
Technology considerations
Monitoring

- Host, container, and application monitoring
- Root cause analysis and remediation
- Data store for trending and archival analysis
- Canned metrics and dashboards
- Software as a service (SaaS) versus an on-premises monitoring approach
- Open source versus pay-for products
- **Monitoring tools:** CloudForms, SysDig, Datadog, CoScale, Prometheus/Grafana

Top five layers to monitor

- Application
- Services
- Kubernetes deployment
- Kubernetes internals
- Host nodes

(Source: SysDig)
Technology considerations
Resource management

- Developers are not good at sizing estimates
  - Tendency to overcommit resources
  - Overprovision for “safety”
- Leads to inefficient CPU and memory usage
- Magnified exponentially with thousands of pods
- Analyser tools: cAdvisor, Prometheus/Grafana, Densify, Turbonomics
## PoC to production configuration considerations

<table>
<thead>
<tr>
<th>Deployment scenario</th>
<th>PoC, Dev/QA, or SMB deployment</th>
<th>Mid-range production configuration</th>
<th>Enterprise production starter configuration (bare metal)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Deployment scenario</strong></td>
<td>All services, masters, workers on VM (with persistent storage supported), HA supported</td>
<td>VM or bare metal workers with persistent storage</td>
<td>All services, masters, workers on bare metal</td>
</tr>
<tr>
<td><strong>Total physical nodes</strong></td>
<td>3 nodes</td>
<td>6+ nodes</td>
<td>8+ nodes</td>
</tr>
</tbody>
</table>
| **Number of instances** | All on VMs:  
– 3 master  
– 3 etcd  
– 3 infrastructure  
– 2 high availability (HA) proxy  
– 3 workers | – 3 masters/etcd, infrastructure, HA proxy on VMs over 3 physical nodes  
– 3+ physical nodes for N number workers on VMs or bare metal | – 3 nodes—3 master, 3 etcd  
– 2 nodes—infrastructure, HA load balancer, and HA registry management tools, such as Ansible Tower  
– 3+ nodes—k8s workers on bare metal |
| **Key SW** | – OpenShift  
– Red Hat Hyperconverged Infrastructure (RHHI) | – OpenShift, RHV + external storage array  
– Or OpenShift, RHHI (for SW defined storage) | – OpenShift  
– Monitoring, logging, billing apps  
– Persistent storage plugin |
Accelerate OpenShift adoption with HPE

Reference architectures

HPE OpenShift solutions
(Services component, ecosystem, deployment guide, and automation)

Consistent platform from DEV to OPs

Development

Accelerate developer productivity

Operations optimized

Production

Simplify the IT experience
HPE Composable Systems: the ideal container platform
Solution for enterprise scale container deployment

- **Deploy containers at cloud-like speed**
  Improve application time to value

- **Centralize container life cycle management**
  Reduce updates from hours to minutes

- **Flex container resources up and down**
  Efficient resource allocation by business demands

- **Advanced container data management**
  Data protection and storage efficiency for containers

HPE Synergy and 3PAR/Nimble
Data management and protection
Use cases for persistent storage with Red Hat OpenShift

**DevOps CI/CD pipelines**
- Jenkins, Microsoft® VSTS, CircleCI
- Release more, faster, and better

**IT operations**
- Atlassian Tools, ELK stack, LAMP apps
- Simplified security—easy to manage

**Lift and shift**
- LAMP apps, ERP systems
- From VMs or bare-metal

**CaaS**
- Self-service for developers
- Secure and predictable
Hardware versus software

<table>
<thead>
<tr>
<th>Capability</th>
<th>External storage</th>
<th>Software-defined storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consistency model</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data services</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Storage reduction</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scale and grow</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Efficiency</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CAPEX/OPEX/TCO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Protocol</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Security</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Backup, recovery, archive</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reliability, availability, serviceability</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cloud native</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Solution: HPE Persistent Storage platform for containers

<table>
<thead>
<tr>
<th><strong>Speed up DevOps</strong></th>
<th><strong>Lift and shift data with applications</strong></th>
<th><strong>Simplify container operations</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Self-service automation rich container platform integration</td>
<td>Multicloud onramp for data using HPE Cloud Volumes</td>
<td>Container QoS, security IOPS, encryption</td>
</tr>
<tr>
<td>Comprehensive REST APIs plug into Ansible, Puppet, Chef</td>
<td>Onboard data easily by instantly converting legacy volumes to persistent volumes</td>
<td>Container data protection: clean up and retention for snaps and clones</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Simple, fast, efficient: predictive flash for six-nines availability, support</td>
</tr>
</tbody>
</table>
HPE Persistent Storage platform for Red Hat OpenShift

OpenShift Container Platform 3.5 to 3.9
OpenShift Origin

Open APIs + HPE Storage open-source software*

Coming soon: HPE Cloud Volumes

*https://github.com/hpe-storage/dory

HPE Docker Volume plugins

FlexVolume Driver
Provisioner

Docker Volume API

Plugin Unix Socket

HPE Docker Volume plugins

3PAR
Nimble Storage

Azure
HPE Cloud Volumes
amazon webservices
HPE Nimble Kube Storage Controller overview

Parameters

---
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: my-storage-class
provisioner: hpe.com/nimble
parameters:
  description: "My Description"
  encryption: "true"
  limitIOPS: "1000"
  perfPolicy: "My Policy"
  protectionTemplate: "my-prot-1"
---

kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: my-pvc
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 500Gi
  storageClassName: my-storage-class

---

description: "My Description"
destroyOnRm: "true"

perfPolicy: "SQL Server"
limitIOPS: "32000"
limitMBPS: "512"

pool: "allflash"
folder: "My Tenant"

protectionTemplate: "local-cloud"

encryption: "true"
fsOwner: "8192:500"
fsMode: "0755"
thick: "true"
sizeInGiB: "4000"
dedupe: "true"

cloneOf: "MyDockerVol1"
snapshot: "MySnapshot"
createSnapshot: "true"
importVol: "MyNimbleVol1"
importVolAsClone: "MyNimbleVol1"
snapshot: "MySnapshot"
HPE Pointnext Services for OpenShift
OpenShift container service considerations
Container and cloud adoption is not trivial

- Overall business objectives
- Determine application migration strategy
- Review networking, security, and storage requirements
- Define system architecture
- Define and implement PoC
- How best to containerize app

Continuous integration and deployment pipeline

Discovery | Design | Deployment | Pilot
Announcing HPE cloud native container service for OpenShift

- Review application requirements
- 2–3 day workshop to gather requirements and define integrations
- Create design
- Deploy container platform environment
- Pilot containerized applications
- Move to production

Discovery | Design | Deployment | Test and evaluate | Pilot or trial workload | Production
Plan for success
Move from PoC to production—Key success factors

– **Implement best practices** and address issues/learnings from PoC (people, process, technology)

– **Have a complete OpenShift container ecosystem in place**: HA, security, monitoring, data management, etc.

– **Determine CAPEX vs OPEX**: plan whether to do it yourself or partner

Accelerate this path with

**HPE + Red Hat**
Resources and key contacts

Reference configuration
Reference configuration for Red Hat OpenShift Container Platform on HPE Synergy Composable Infrastructure
– Video: hpedemoportal.ext.hpe.com/search/Automated deployment of Red Hat OpenShift on HPE Synergy

HPE platform
– hpe.com/info/composableprogram

Red Hat OpenShift Container Platform datasheet
– redhat.com/en/resources/openshift-container-platform-datasheet

GitHub repositories
– github.com/RHsyseng/ocp-on-synergy

HPE contacts
Ka Wai Leung
Containers Solutions Product Management
kawai@hpe.com

Gary Lee Harris
Pointnext Container Consulting
glh@hpe.com

Michael Mattsson
HPE Storage Tech Marketing
michael.mattsson@hpe.com

Bob Zepf
HPE Strategic Alliances
bob.zepf@hpe.com
Thank you