OpenShift e Container Storage
Kubernetes enterprise per le grandi idee

PIERLUIGI SFORZA
Solution Architect
psforza@redhat.com

ALESSANDRO ARRICHIELLO
Solution Architect
ale@redhat.com

#RedHatOSD
... so you want to do containers and Kubernetes?
We were very lucky to be joined early on by the very capable OpenShift team … without their perspective and contributions, I don’t think we would be standing here today

*Brendan Burns, co-creator of Kubernetes*
OPENSHIFT IS KUBERNETES FOR THE ENTERPRISE

Security fixes
100s of defect and performance fixes
200+ validated integrations
Middleware integrations
(container images, storage, networking, cloud services, etc)
9 year enterprise lifecycle management
Certified Kubernetes

Kubernetes Release

1-3 months hardening

OpenShift Release
KUBERNETES WORKLOADS
CONTROLLERS MATTERS!

Different types of applications (stateful, stateless, batch, agent, ...) require different orchestrator behaviors.

Main controller types:
- Replica Sets
- Stateful Sets
- Daemon Sets
- Jobs (OneTime, Cron)

```go
func NewControllerInitializers(loopMode ControllerLoopMode) map[string]InitFunc{
    controllers := map[string]InitFunc{
        "endpoint" : startEndpointController,
        "replicationcontroller" : startReplicationController,
        "replicationcontroller" : startReplicationController,
    }
    return controllers
}
```
WHAT IS A POD?
CONTAINERS ARE WRAPPED IN PODS WHICH ARE UNITS OF DEPLOYMENT AND MANAGEMENT

POD
CONTAINER
IP: 10.1.0.11

POD
CONTAINER
CONTAINER
IP: 10.1.0.55
CONTROLLER & CONTROLLER-MANAGER

- The **controller-manager** is the Master’s component that manage the controllers
- A **controller** is a loop that governs the status of kubernetes resources (such as pods) in order to bring it from the current state to the desired state
- Controllers react to **kubernetes events** and define **how resources should be orchestrated**
CONTROLLER & CONTROLLER-MANAGER

- The **controller-manager** is the Master’s component that manage the controllers
- A **controller** is a loop that governs the status of kubernetes resources (such as pods) in order to bring it from the current state to the desired state
- Controllers react to **kubernetes events** and define **how resources should be orchestrated**
DEPLOYMENT AND REPLICASET

- A Deployment controller provides declarative updates for Pods and ReplicaSets.
- ReplicaSet controller ensures that a specified number of pod replicas are running at any given time.
- Recommended to run stateless application.

* x= rand seed
STATEFULSET

- A stateful set ensures:
  - Stable resource allocation such as name and storage
  - Ordered, graceful deployment, scaling up and termination

- Ideal for highly available workloads in a “clustered mode”
DAEMONSET

- A daemon set ensure to have just 1 copy of a pod on every node

- Daemon set is useful for: Logging Aggregators, Monitoring, Load Balancers / Reverse Proxies / API Gateways, single host batch…

* x = available node count
REFERENCE ARCHITECTURE FOR ENTERPRISE KUBERNETES

Automated Operations*

Kubernetes

Red Hat Enterprise Linux or Red Hat CoreOS

Best IT Ops Experience

CaaS ↔ PaaS

Best Developer Experience

*coming soon with OCP 4.0 (targeted for GA Dec 2018)
Istio Service Mesh
For service-to-service communications
OPENSHIFT SERVICE MESH: ISTIO*

Istio makes it easy to create a network of deployed services with load balancing, service-to-service authentication, monitoring, and more, helping to avoid operational nightmares.

**POLICY**
Grants the ability to write policy that applies to all applications and is not language specific.

**ROUTING**
Allows for the control of routing flows.

**TELEMETRY**
Provides the observability needed to manage microservices, such as how services are invoked, communication flows, and points of latency.

* Technology Preview
ISTIO COMPANION: KIALI & JAEGER

Kiali and Jaeger make the perfect companion for Istio Service Mesh

**VISUALIZATION**

Kiali works with Istio to visualize the service mesh topology, features like circuit breakers or request rates.

**TRACING**

Kiali includes Jaeger Tracing, which provides distributed tracing out of the box.
SERVICE MESH ARCHITECTURE

SERVICE MESH
- Load Balancing
- Fault Tolerance
- Traceability
- Observability
- Service Security
- Infra Security
- Chaos Engineering
- Traffic Control

SERVICE
- Build Automation
- Logs
- Monitoring
- Infra Security
- CI/CD
- Load Balancing

SERVICE OPS
- Deployment Resiliency
- Service Discovery
- Config
- Resource Management
- Elasticity

INFRA OPS

INFRA
- PHYSICAL
- VIRTUAL
- CLOUD

OPENSIGHT
ENTERPRISE KUBERNETES

#RedHatOSD
MICROSERVICES EVOLUTION
SERVICE MESH ARCHITECTURE

Applies security, route rules, policies and reports traffic telemetry at the pod level.
CIRCUIT BREAKERS WITH ISTIO

transparent to the services
SECURE COMMUNICATION WITH ISTIO

mutual TLS authentication, transparent to the services
DISTRIBUTED TRACING WITH ISTIO & JAEGER

discovers service relationships and process times, transparent to the services

SERVICE A 210 ms SERVICE B 720 ms SERVICE C
930 ms
DEMO TIME:
Istio Internals
Prometheus Cluster Monitoring

Providing alerts also for OpenShift Container Storage

ALESSANDRO ARRICHIELLO  
Solution Architect  
ale@redhat.com

CARLOS TORRES  
Specialist Solution Architect  
ctorres@redhat.com
COMPREHENSIVE MONITORING SUITE

The stack includes three distinct UIs:

- **Alertmanager UI** to manage alerts which have been fired
- **Prometheus UI** for querying and plotting any metrics
- **Grafana** to browse cluster-level dashboards

All UIs are accessible directly via the new admin console under the “Monitoring” menu.
DEMO TIME:
Cluster Console - EventFeed
**RedHatOSD**

**VIDEO**

https://youtu.be/MG-2s11uoPI
DEMO TIME: Cluster Console - Monitoring
https://youtu.be/4aREWqD-v3c
OpenShift Container Storage
RED HAT OPENSSHIFT CONTAINER STORAGE

Flexible deployment with the same user experience and features

Converged = in containers
 Persona: DevOps, App Architects

- Highly scalable, scale app+storage, start small and scale fast
- Storage life cycle managed by OCP

Independent = for containers
 Persona: Storage Admins, Infrastructure Admins

- Highly scalable, independent scalability from OCP platform
- Adaptative to in-place BC/DR strategies
DEMO TIME:
Monitoring - OpenShift Container Storage
https://youtu.be/35XimCphonM
## RHOCS: ANSIBLE ADVANCED DEPLOYMENT

Converged playbooks already available

<table>
<thead>
<tr>
<th>Deployment workflow</th>
<th>Registry</th>
<th>Metrics</th>
<th>Logging</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deploying Red Hat Openshift Container Storage in Converged Mode</td>
<td></td>
<td></td>
<td></td>
<td>✔</td>
</tr>
<tr>
<td>Deploying Red Hat Openshift Container Storage in Converged Mode with Registry</td>
<td>✔️</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deploying Red Hat Openshift Container Storage in Converged Mode with Logging and Metrics</td>
<td></td>
<td>✔️</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Deploying Red Hat Openshift Container Storage in Converged mode for Applications with Registry, Logging, and Metrics</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>
WHAT IS A SERVICE BROKERAGE?

Automated, Standard and Consistent
A multi-vendor project to standardize how services are consumed on cloud-native platforms across service providers
BROKERAGE WITH OPENSHIFT

SERVICE CATALOG

Broker

OpenShift Template Broker

OpenShift Ansible Broker

AWS Service Broker

Other Service Brokers

Service

OPENSHIFT

ANSIBLE

AWS

OTHER COMPATIBLE SERVICES

OpenShift Templates

Ansible Playbook Bundles

AWS Services

Other Services

#RedHatOSD
Anything you can do with Ansible, you can do with the Ansible Broker

- Use Ansible on OpenShift to
  - Deploy containerized applications
  - Manage external components (e.g. Oracle database)
  - Provision cloud services (e.g. AWS RDS)
  - Orchestrate multi-service solutions
  - Manage dependencies or other logics on deployments (e.g. database initialization)

OPENSHIFT ANSIBLE BROKER

TP available from version 3.6
GA from 3.7
ANSIBLE PLAYBOOK BUNDLES (APB)

- Packaged as a container image
- Embed Ansible runtime
- Use named playbooks for actions
- Fulfill Service Catalog dynamically with services and parameters
- Provide a command line tool to manage APBs
APB CREATION WORKFLOW

INIT
CUSTOMIZATION
PREPARE AND BUILD
PUSH

Site Reliability Engineer
playbooks and $vars
APB image
Service Catalog update

DEMO TIME:
MariaDB Provisioning on Remote RHEL
OpenShift APB
Mariadb Remote Provisioning

APB container runs
provision.yaml playbook to install and
configure MariaDB on external VM

oc run mariadb-apb provision $vars

ansible-playbook provision.yaml $vars
<table>
<thead>
<tr>
<th>NAME</th>
<th>READY</th>
<th>STATUS</th>
<th>RESTARTS</th>
<th>AGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>docker-registry-2-dglkb</td>
<td>1/1</td>
<td>Running</td>
<td>6</td>
<td>21d</td>
</tr>
<tr>
<td>registry-console-1-dmqg2</td>
<td>1/1</td>
<td>Running</td>
<td>6</td>
<td>21d</td>
</tr>
<tr>
<td>router-2-9maxx</td>
<td>1/1</td>
<td>Running</td>
<td>6</td>
<td>21d</td>
</tr>
<tr>
<td>router-2-chpqa2</td>
<td>1/1</td>
<td>Running</td>
<td>6</td>
<td>21d</td>
</tr>
<tr>
<td>router-2-twqh1</td>
<td>1/1</td>
<td>Running</td>
<td>6</td>
<td>21d</td>
</tr>
<tr>
<td>kube-service-catalog</td>
<td>1/1</td>
<td>Running</td>
<td>6</td>
<td>21d</td>
</tr>
<tr>
<td>controller-manager-test</td>
<td>1/1</td>
<td>Running</td>
<td>6</td>
<td>21d</td>
</tr>
<tr>
<td>localregistry-mariadb-deployment-apk-prom-jkp7k</td>
<td>1/1</td>
<td>Running</td>
<td>6</td>
<td>21d</td>
</tr>
<tr>
<td>openshift-ansible-service-broker</td>
<td>1/1</td>
<td>Running</td>
<td>6</td>
<td>21d</td>
</tr>
<tr>
<td>openshift-infra</td>
<td>1/1</td>
<td>Running</td>
<td>6</td>
<td>21d</td>
</tr>
<tr>
<td>openshift-infra</td>
<td>1/1</td>
<td>Running</td>
<td>6</td>
<td>21d</td>
</tr>
<tr>
<td>openshift-infra</td>
<td>1/1</td>
<td>Running</td>
<td>6</td>
<td>21d</td>
</tr>
<tr>
<td>openshift-info</td>
<td>1/1</td>
<td>Running</td>
<td>6</td>
<td>21d</td>
</tr>
<tr>
<td>openshift-info</td>
<td>1/1</td>
<td>Running</td>
<td>6</td>
<td>21d</td>
</tr>
<tr>
<td>openshift-template-service-broker</td>
<td>1/1</td>
<td>Running</td>
<td>6</td>
<td>21d</td>
</tr>
<tr>
<td>openshift-template-service-broker</td>
<td>1/1</td>
<td>Running</td>
<td>6</td>
<td>21d</td>
</tr>
<tr>
<td>openshift-web-console</td>
<td>1/1</td>
<td>Running</td>
<td>6</td>
<td>21d</td>
</tr>
<tr>
<td>openshift-web-console</td>
<td>1/1</td>
<td>Running</td>
<td>6</td>
<td>21d</td>
</tr>
<tr>
<td>openshift-web-console</td>
<td>1/1</td>
<td>Running</td>
<td>6</td>
<td>21d</td>
</tr>
<tr>
<td>test</td>
<td>1/1</td>
<td>Running</td>
<td>1</td>
<td>20s</td>
</tr>
<tr>
<td>test</td>
<td>1/1</td>
<td>Running</td>
<td>1</td>
<td>20s</td>
</tr>
</tbody>
</table>
APB INTEGRATION WITH ANSIBLE GALAXY

Support discovering/running APB sources published to Ansible Galaxy from the OpenShift Ansible Service Broker.

How it works:

- APB’s can be now be created right from mazer command line tool using the init command and then pushed to Ansible Galaxy.

- Broker should now be able to discover and provision APB-based services published to Ansible Galaxy and also make them available in the service catalog.
What’s Next?
Operators!
KUBERNETES OPERATORS
THE EASE OF THE CLOUD EVERYWHERE

- encode human operational knowledge
- automatically patch, upgrade, recover, and tune apps and services
- Kubernetes-native
- Purpose-built for a specific application or service
ENCODING AND AUTOMATING OPS KNOWLEDGE WITH OPERATORS

WITHOUT OPERATORS
- Reactive
  - Continually checks for anomalies
  - Alert humans for response
  - Requires manual change to fix

WITH OPERATORS
- Proactive
  - Continually adjusts to optimal state
  - Automatically acts in milliseconds
OPERATOR FRAMEWORK
An open source toolkit to manage application instances on Kubernetes in an automated, scalable way

- **OPERATOR SDK**: Build Operators without specialized knowledge of the Kubernetes API
- **OPERATOR LIFECYCLE MANAGER**: Install, update, and manage Operators and their dependencies
- **OPERATOR METERING**: Enable usage reporting for Operators

https://github.com/operator-framework
OPERATOR IMPLEMENTATION PATHS

Phase I: Installation
Phase II: Upgrades
Phase III: Lifecycle
Phase IV: Insights
Phase V: Auto-pilot

Requires custom Operator - building simplified with SDK

No custom Operator Required

HELM & GO AVAILABLE NOW
ANSIBLE IN 2019

HELM

ANSIBLE

GO
OPERATORS IN PREVIEW IN OCP 3.11

APPLICATION OPERATORS
DEVELOPER PREVIEW
- etcd
- AMQ Streams
- mongoDB
- dynatrace
- Couchbase

OPERATOR LIFECYCLE MANAGER (OLM)
TECH PREVIEW
Install, manage, and upgrade Operators and their dependencies

Portable application services across any infrastructure
- PHYSICAL
- VIRTUAL
- PRIVATE CLOUD
- PUBLIC CLOUD

#RedHatOSD
PORTABLE HYBRID CLOUD SERVICES WITH ISV OPERATORS

60+ Certified ISV Operators in Red Hat Early Access Program
GRAZIE PER L’ATTENZIONE