Deploying Oracle WebLogic Server Applications in Kubernetes

Break New Ground

ORACLE
CODE

Maciej Gruszka
Director Product Management

April 16, 2019

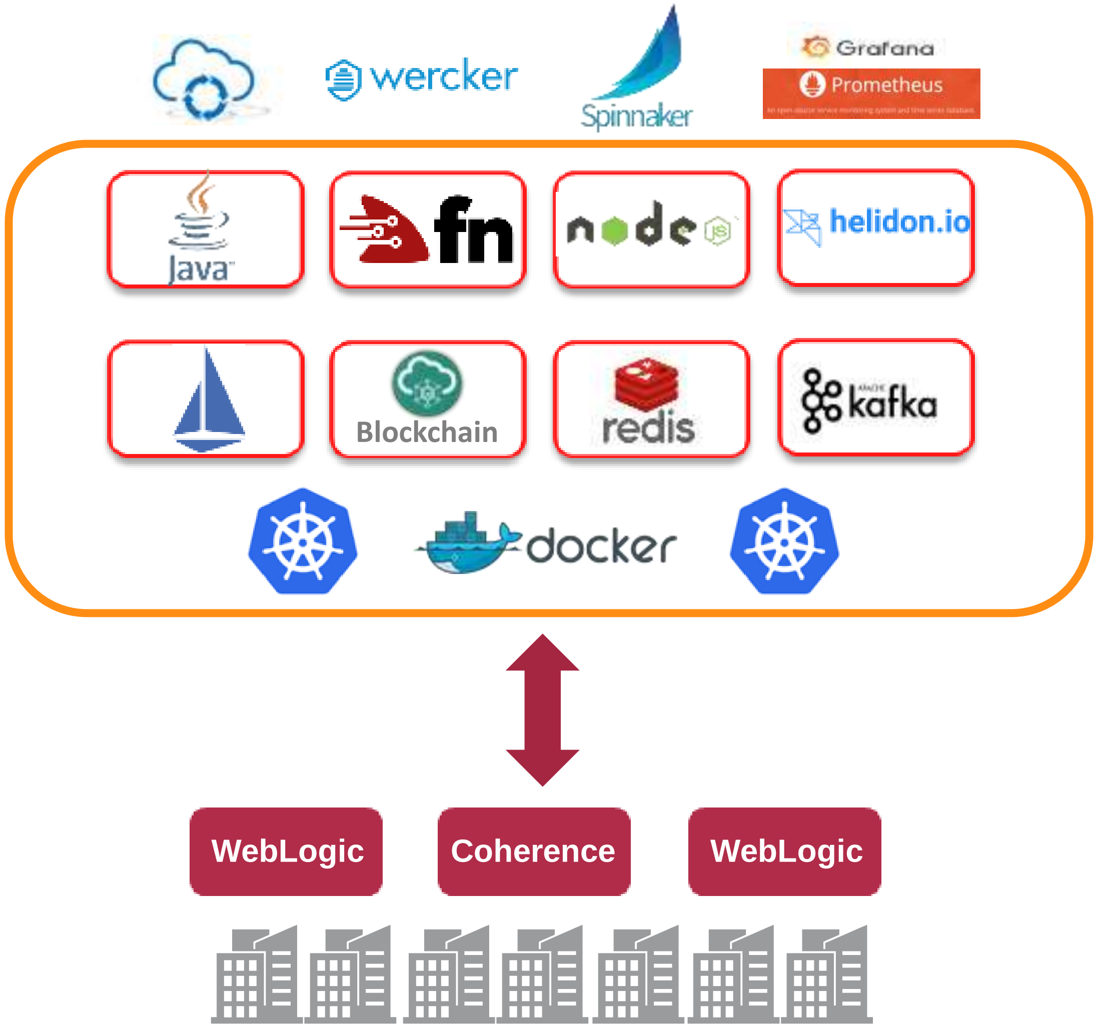
ORACLE

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

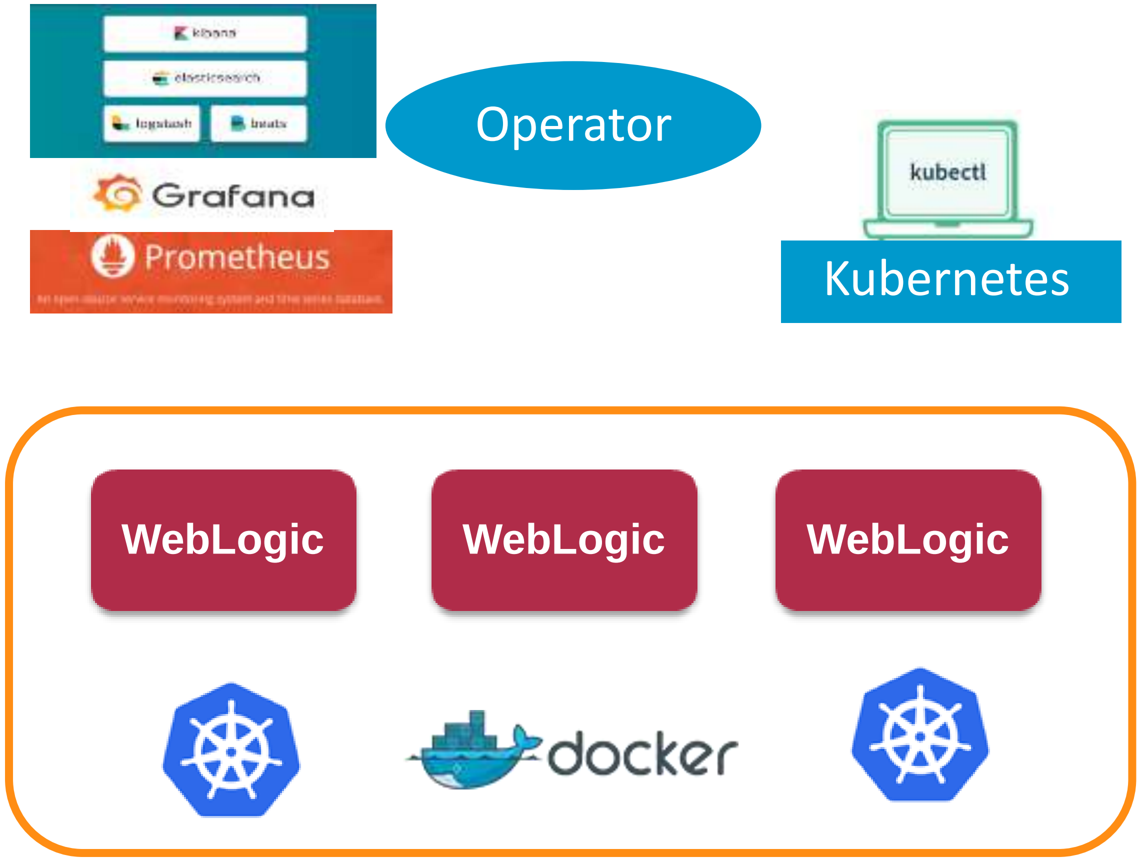
WebLogic, Coherence and Cloud Native Trends

- Industry trends
  - Microservices, serverless
  - Private and public clouds
  - Containers, orchestration frameworks
- WebLogic, Coherence customer demand
  - Leverage cloud neutral infrastructure
  - Integrate with new tools and services
  - Evolve WebLogic, Coherence for these environments



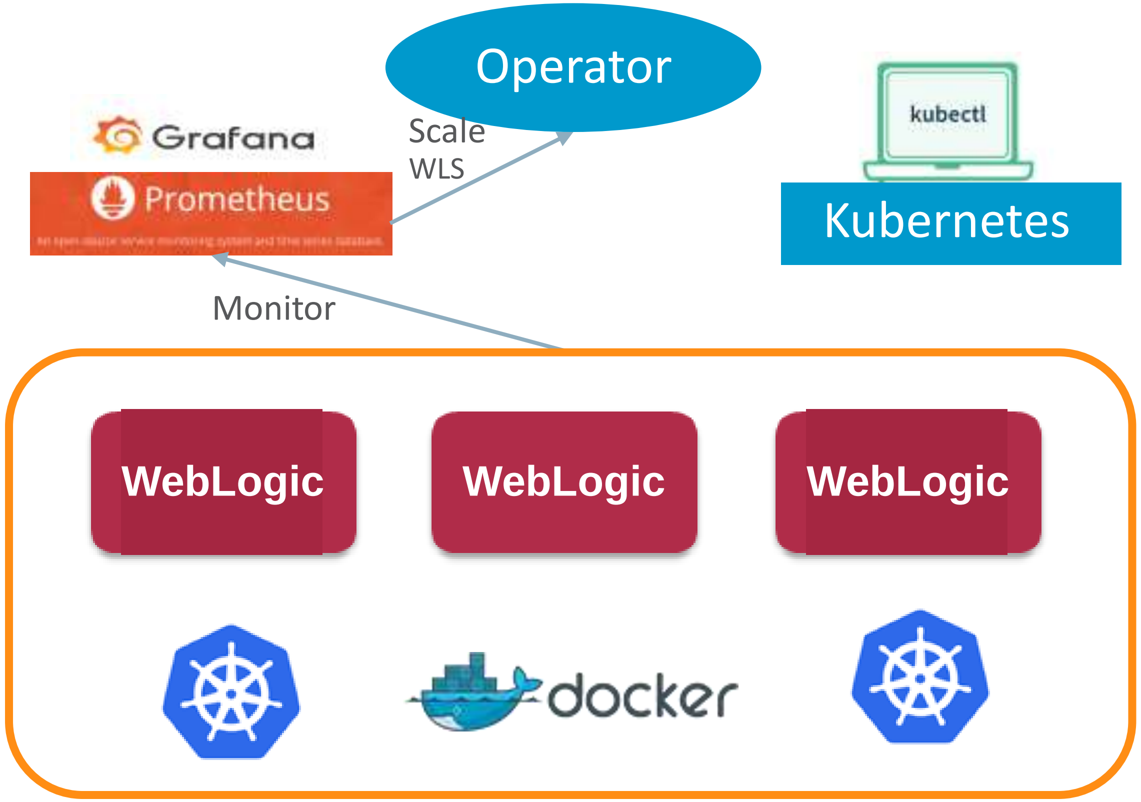
Building Blocks for WebLogic Kubernetes Support

- WebLogic Docker certification
  - Docker images, Dockerfiles, examples
- WebLogic Kubernetes certification
  - How-to, best practices
- Value add integration
  - Management: Operator
  - Monitoring: Exporter for Prometheus
  - Migration: Deploy tooling
  - Logging: Exporter for Elastic Stack
  - Image: Tool management



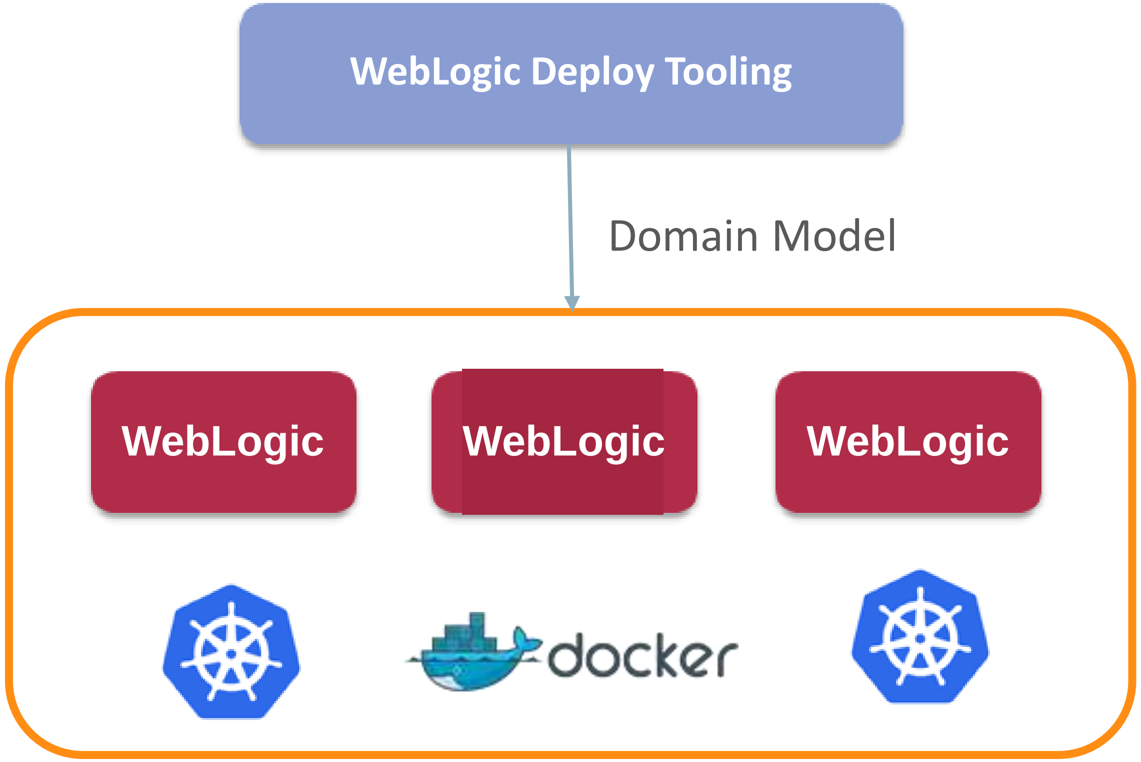
WebLogic Monitoring Exporter

- Monitoring Exporter enables Prometheus monitoring of WebLogic
- Standard monitoring tools can be used for monitoring WebLogic
- Grafana Dashboards used for visualization
- Prometheus **auto-scaling** of WebLogic cluster
- Prometheus and Grafana example [GitHub Sample]()



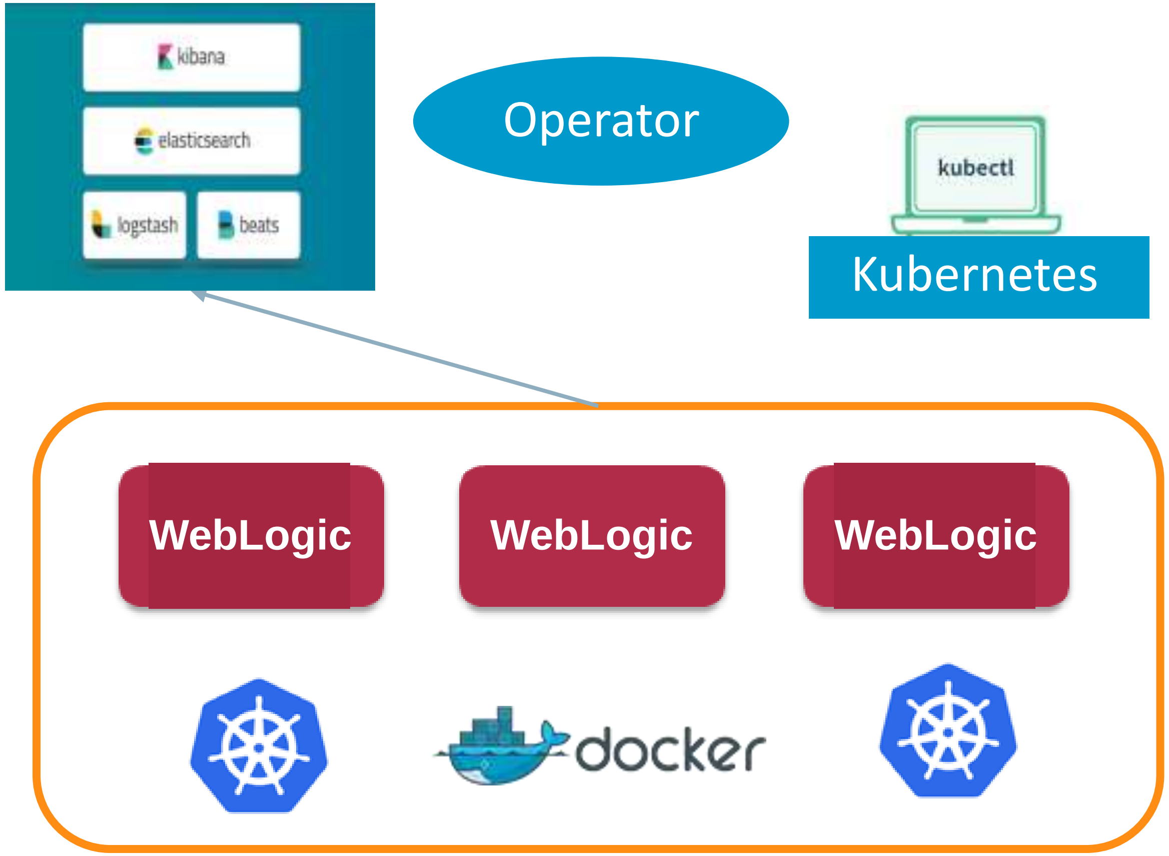
WebLogic Deploy Tooling

- Introspect domains
  - WebLogic 10.3.6, 12.1.3, 12.2.1.X
  - Create a model (yaml) of the domain
  - Migrate existing domains and applications Upgrade (if required) to 12.2.1.X
- Customize and Validate configuration to meet Kubernetes requirements
- Create domains in Docker image [GitHub Sample]



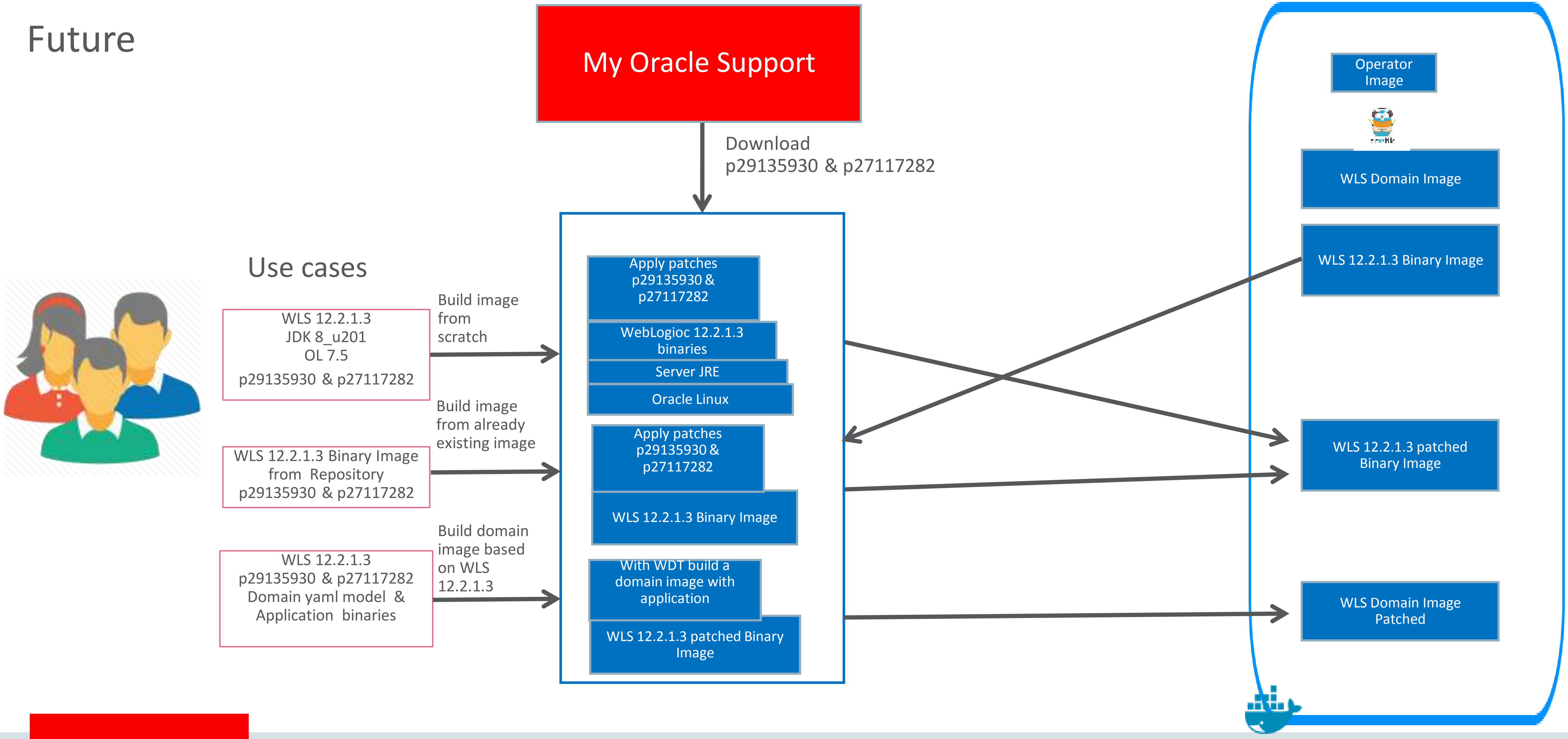
WebLogic Logging Exporter

- Logging Exporter enables exporting WebLogic server logs to the Elastic Stack
- Store logs in the Elastic Stack
- Search and analyze logs in Elastichsearch
- Display logs in dashboards in Kibana
- Integrate with FluentD (future)
- [GitHub weblogic-logging-exporter]()



Patching WebLogic Image with WebLogic Image Tool

Future



Repository

My Oracle Support

Download
p29135930 & p27117282

Use cases

WLS 12.2.1.3
JDK 8_u201
OL 7.5
p29135930 & p27117282

Build image
from
scratch

Apply patches
p29135930 &
p27117282

WebLogioc 12.2.1.3
binaries

Server JRE

Oracle Linux

WLS 12.2.1.3 Binary Image
from Repository
p29135930 & p27117282

Build image
from already
existing image

Apply patches
p29135930 &
p27117282

WLS 12.2.1.3 Binary Image

WLS 12.2.1.3
p29135930 & p27117282
Domain yaml model &
Application binaries

Build domain
image based
on WLS
12.2.1.3

With WDT build a
domain image with
application

WLS 12.2.1.3 patched Binary
Image

Operator
Image

WLS Domain Image

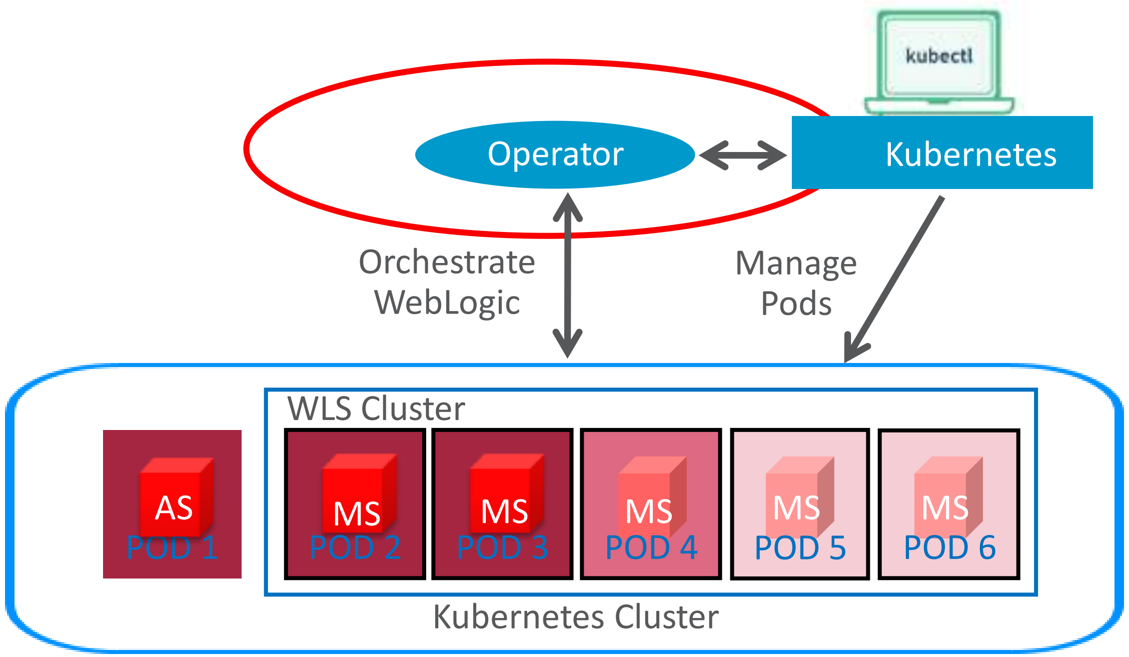
WLS 12.2.1.3 Binary Image

WLS 12.2.1.3 patched
Binary Image

WLS Domain Image
Patched

Why build the WebLogic Kubernetes Operator?

- The Oracle WebLogic Server Kubernetes Operator contains a set of useful built-in knowledge about how to perform various lifecycle operations on a domain correctly.
- The Operator using a common set of Kubernetes APIs, provides advanced user experience, automating operations such as:
  - provisioning
  - life cycle management
  - updates
  - scaling
  - Security
- The operator is not an administrator of the WebLogic domain. The operator has only limited interest in the domain configuration, with its main concern being the high-level topology of the domain.

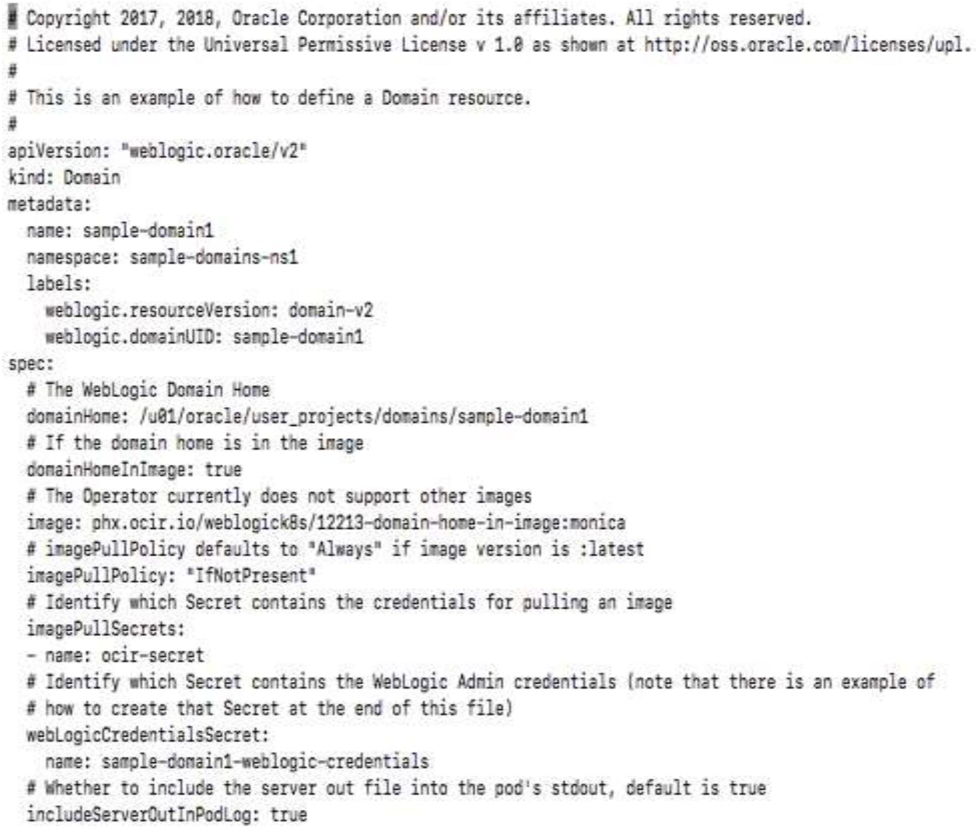


Kubernetes Custom Resources

- A *custom resource* (CR) allows you to define your own object that extends the Kubernetes API.
- A *custom resource definition* (CRD) file defines your own object kinds and lets the Kubernetes API server to begin serving the specified custom resource.
- The CRD is defined in a yaml file that represents the CR.
- To create the custom resource

  kubectl apply –f domain.yaml

domain.yaml



```
# Copyright 2017, 2018, Oracle Corporation and/or its affiliates. All rights reserved.
# Licensed under the Universal Permissive License v 1.0 as shown at http://oss.oracle.com/licenses/upl.
#
# This is an example of how to define a Domain resource.
#
apiVersion: "weblogic.oracle/v2"
kind: Domain
metadata:
  name: sample-domain1
  namespace: sample-domains-ns1
  labels:
    weblogic.resourceVersion: domain-v2
    weblogic.domainUID: sample-domain1
spec:
  # The WebLogic Domain Home
  domainHome: /u01/oracle/user_projects/domains/sample-domain1
  # If the domain home is in the image
  domainHomeInImage: true
  # The Operator currently does not support other images
  image: phx.ocir.io/weblogick8s/12213-domain-home-in-image:monica
  # imagePullPolicy defaults to "Always" if image version is :latest
  imagePullPolicy: "IfNotPresent"
  # Identify which Secret contains the credentials for pulling an image
  imagePullSecrets:
  - name: ocir-secret
  # Identify which Secret contains the WebLogic Admin credentials (note that there is an example of
  # how to create that Secret at the end of this file)
  webLogicCredentialsSecret:
    name: sample-domain1-weblogic-credentials
  # Whether to include the server out file into the pod's stdout, default is true
  includeServerOutInPodLog: true
```

WebLogic Domain Custom Resource

- We create a Kubernetes Resource Object for the WebLogic domain. This is a data structure representation of the WebLogic domain in Kubernetes.

- Domain Custom Resource allows you to *declare* or specify the desired state of the resource.
  - Example I want 3 replicas of managed servers running in the WLS cluster.

- The WebLogic Kubernetes Operator is a controller that is always looking at the Domain Custom Resource and tries to match the actual state to this desired state.
  - Example: Change replicas from 3 to 4, the Operator will start a new pod to match number of replicas.

Domain Custom Resource

Meta Data: Name of Resource, Namespace, Labels, …

Admin Server: Node Ports to expose, Volumes, …

Cluster: Number of Replicas (Managed Servers), …

Domain: Image to base the Domain containers, Domain in PV or in Image, K8S secrets, Logs to pod

Managed Servers: non-clustered MS

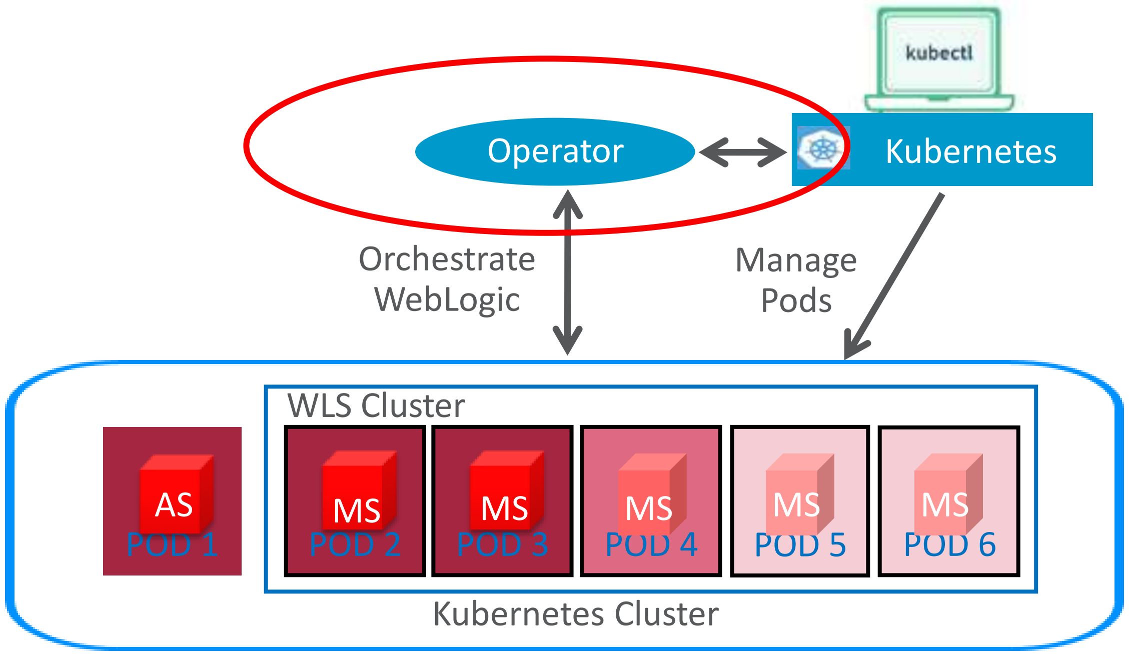
Server Pod: Java Options, Start Policy (Lifecycle control)

Events:

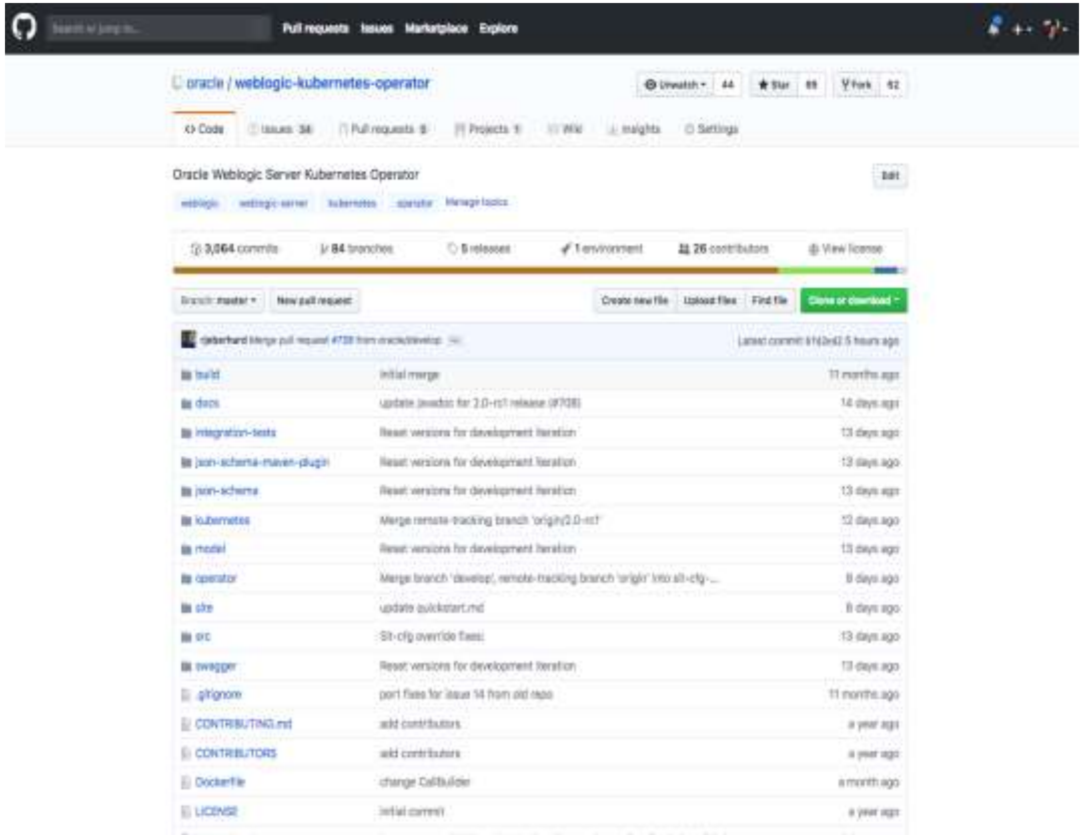
WebLogic Kubernetes Operator

Simplifies management of the WebLogic domain

- Create RBAC roles to manage K8S resources
- Create (new) domain
- Monitor instances (liveliness and readiness)
- Start/stop instances
- Scale up/down domain
- Auto-scale domain
- Rolling Restart of patched domains and updated applications



Operator Image



oracle / weblogic-kubernetes-operator

Oracle Weblogic Server Kubernetes Operator

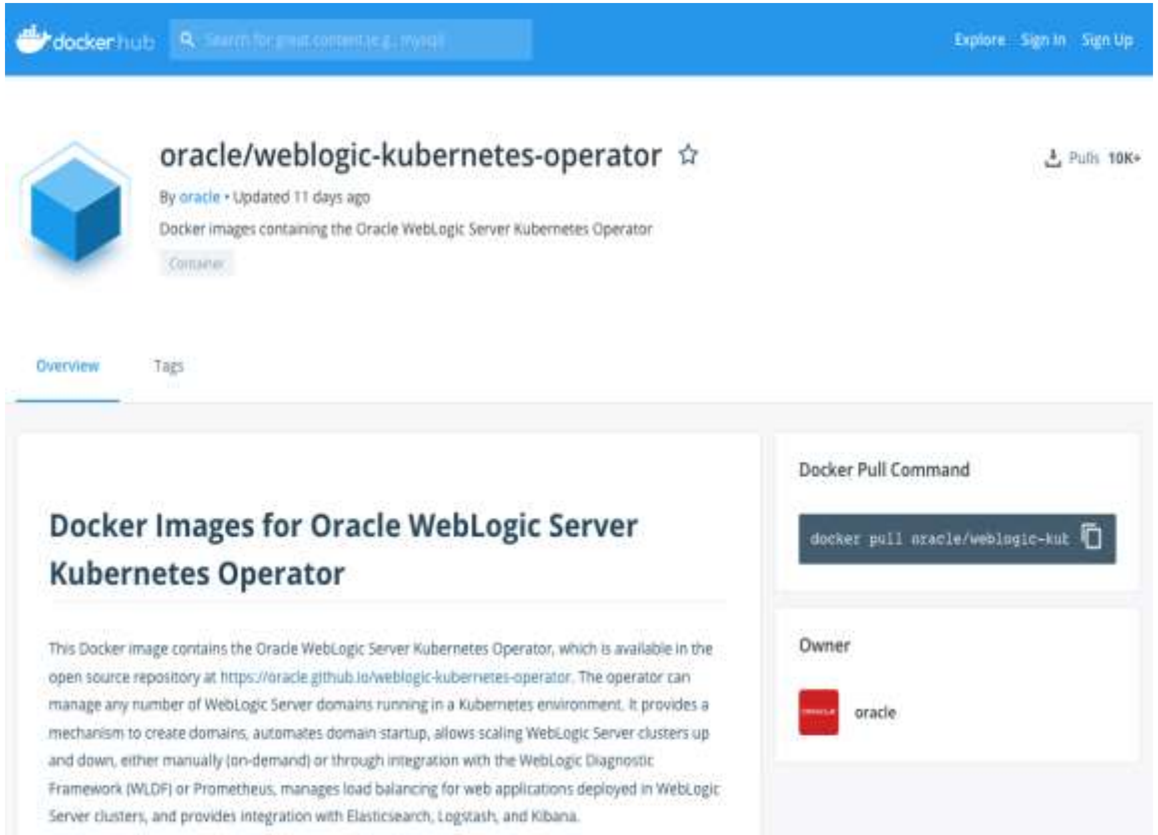
weblogic weblogic-server kubernetes operator Manage topics

3,064 commits | 84 branches | 5 releases | 1 environment | 26 contributors | View license

rjeberhard Merge pull request #738 from oracle/develop … Latest commit 6f42e42 5 hours ago

build	Initial merge	11 months ago
docs	update javadoc for 2.0-rc1 release (#708)	14 days ago
integration-tests	Reset versions for development iteration	13 days ago
json-schema-maven-plugin	Reset versions for development iteration	13 days ago
json-schema	Reset versions for development iteration	13 days ago
kubernetes	Merge remote-tracking branch 'origin/2.0-rc1'	12 days ago
model	Reset versions for development iteration	13 days ago
operator	Merge branch 'develop', remote-tracking branch 'origin' into sit-cfg-…	8 days ago
site	update quickstart.md	8 days ago
src	Sit-cfg override fixes	13 days ago
swagger	Reset versions for development iteration	13 days ago
.gitignore	port fixes for issue 14 from old repo	11 months ago
CONTRIBUTING.md	add contributors	a year ago
CONTRIBUTORS	add contributors	a year ago
Dockerfile	change CallBuilder	a month ago
LICENSE	initial commit	a year ago

[GitHub WebLogic Kubernetes Operator](#)



oracle/weblogic-kubernetes-operator

By oracle • Updated 11 days ago

Docker images containing the Oracle WebLogic Server Kubernetes Operator

Docker Images for Oracle WebLogic Server Kubernetes Operator

This Docker image contains the Oracle WebLogic Server Kubernetes Operator, which is available in the open source repository at https://oracle.github.io/weblogic-kubernetes-operator. The operator can manage any number of WebLogic Server domains running in a Kubernetes environment. It provides a mechanism to create domains, automates domain startup, allows scaling WebLogic Server clusters up and down, either manually (on-demand) or through integration with the WebLogic Diagnostic Framework (WLDF) or Prometheus, manages load balancing for web applications deployed in WebLogic Server clusters, and provides integration with Elasticsearch, Logstash, and Kibana.

Docker Pull Command

```
docker pull oracle/weblogic-kub
```

[DockerHub Oracle WebLogic Operator](#)

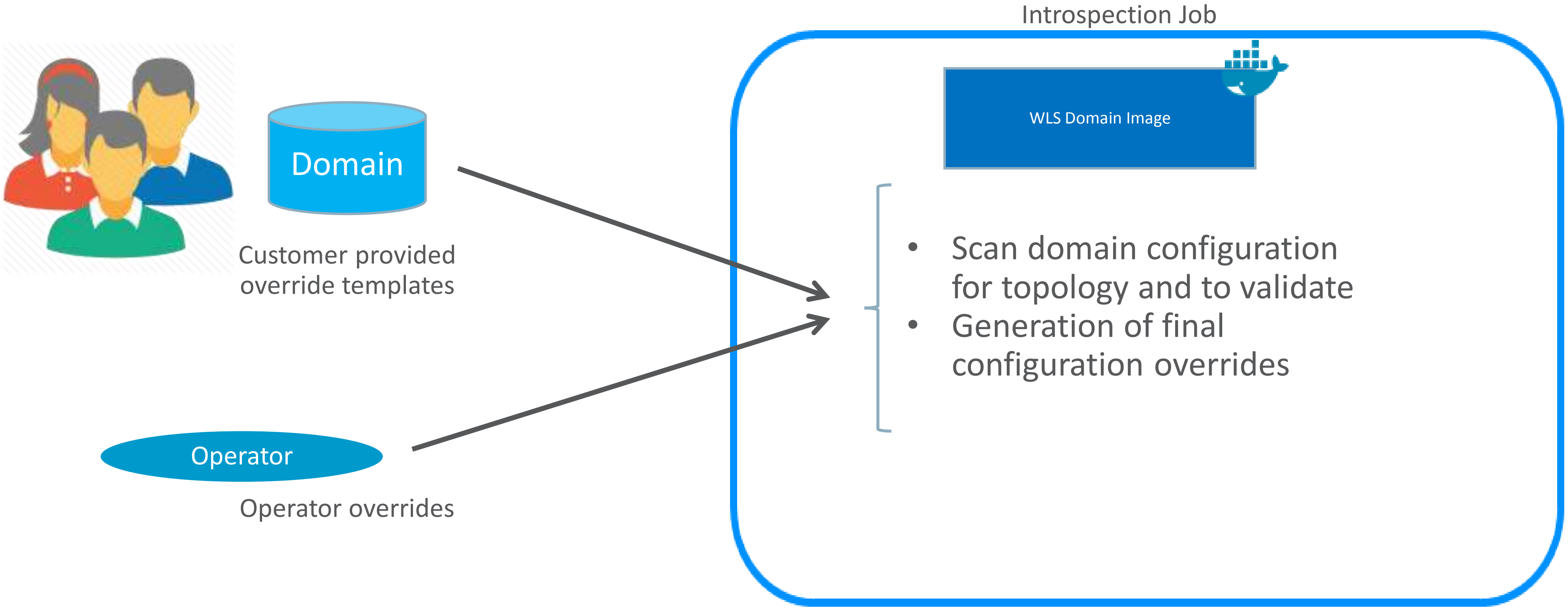
Domain in PV/C vs Image

Options	Domain in PV/C	Domain in Image
Domain Topology Changes	Apply to domain in PV	New Image
Configuration Changes (tunables, credentials, ...)	Change configuration in domain in PV	Overrides only
Patching	New Image	CI/CD (new image)
Application Updates	Apply to domain in PV	CI/CD (new image)
Management of PV/PVC	More complex (filesystem shared per domain)	Simple (not shared, per server)
Administration Console	App deployments and Configuration Changes, can not do lifecycle management	Monitoring and Diagnosis. Invalidate configuration changes
Log Persistence	Supported (PV, Pod FS, Elastic Stack, Standard Out)	Supported (PV, Pod FS, Elastic Stack, Standard Out)
HA Across Availability Domain	Limited (requirement for shared PV)	Supported (no requirement for shared PV)
DR across Regions	Supported Active/Passive (like on Premise user responsible for maintaining domain configuration in sync across DC)	Supported Active/Passive (easier user does not need to sync domain configuration across DC)

Configuration Overrides

- WebLogic Images containing Application, domain configuration, resources are immutable.
- These Docker images must be portable
  - Development -> Testing -> Production.
- Follow the customer’s CI/CD process.
- Therefore, customers need a mechanism to override certain domain configuration
  - E.g. Provide data source URL and credentials

Domain Introspection and Config Override Generation



- Scan domain configuration for topology and to validate
- Generation of final configuration overrides

User Configuration Overrides

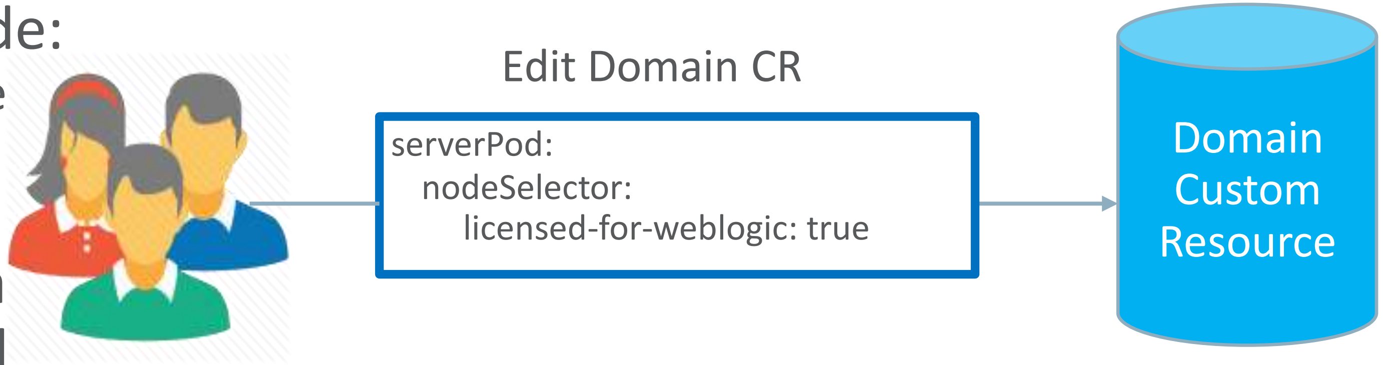
- Typical attributes for overrides include:
  - User names, passwords, and URLs for:
    - JDBC datasources
    - JMS bridges, foreign servers, and SAF
  - Network channel public addresses:
    - For remote RMI clients (T3, JMS, EJB, JTA)
    - For remote WLST clients
  - Debugging
  - Tuning (MaxMessageSize, etc.)
- Configuration Override Documentation

Assigning WebLogic Pods to Nodes

- Create affinity with a Node Selector to constrain a pod to only be able to run on particular nodes.
- Assign a **label** (*key=value*)to the node:

  ```
  kubectl label nodes kubernetes-foo-node
  licensed-for-weblogic=true
  ```

- Edit the Domain Custom Resource a the domain/cluster/server level and assign *key:value* **nodeSelector**.



Edit Domain CR

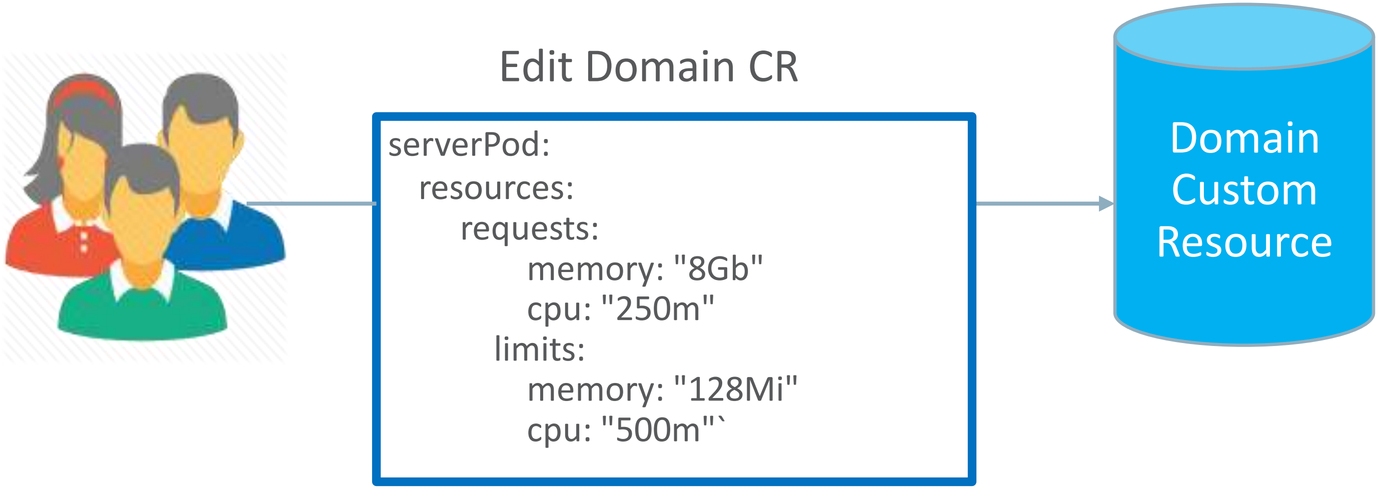
```
serverPod:
  nodeSelector:
     licensed-for-weblogic: true
```

Assigning WebLogic Pods to Nodes

- Assign pods to Nodes based on resources, e.g. CPU and Memory usage

- Edit the Domain Custom Resource and assign a CPU *request* and a CPU *limit* to a container/pod.

- A Pod is scheduled to run on a Node only if the Node has enough CPU resources available.



Edit Domain CR

```
serverPod:
  resources:
     requests:
          memory: "8Gb"
          cpu: "250m"
       limits:
          memory: "128Mi"
          cpu: "500m"`
```

ORACLE®