



Red Hat

Ansible Automation
Platform

ANSIBLE AUTOMATES

Ansible Operators in Action

Ben Pritchett

Tech Lead - PaaS, Red Hat IT



Operator definitions

Operator

Operators are software extensions to Kubernetes that make use of custom resources to manage applications and their components.

Custom Resource

An additional to the Kubernetes API to add additional functionality

Operator SDK

Framework to simplify Operator development, based on Helm, Go, or Ansible

Source:

- ❑ <https://kubernetes.io/docs/concepts/extend-kubernetes/operator/>
- ❑ <https://github.com/operator-framework/operator-sdk>

Why Operators?



Repeatability of installation and upgrade

Operators can have built-in lifecycle management. As changes to an application occur, the operator can not only understand the end state, but how to get there without disruption.

Constant Health Checks

Operators as a concept have metrics and insights built-in, so we can plug into an established observability model.

OTA (Over-the-air) updates

Applications would not need to manage the updates to application components, and these can be managed from the platform itself.

Encapsulation of Domain Expertise

If you can write Ansible or Go, you can wrap your service as an Operator on the platform, making it “as a service”. The domain expertise can define requirements, and applications can easily consume that standard.

Source:



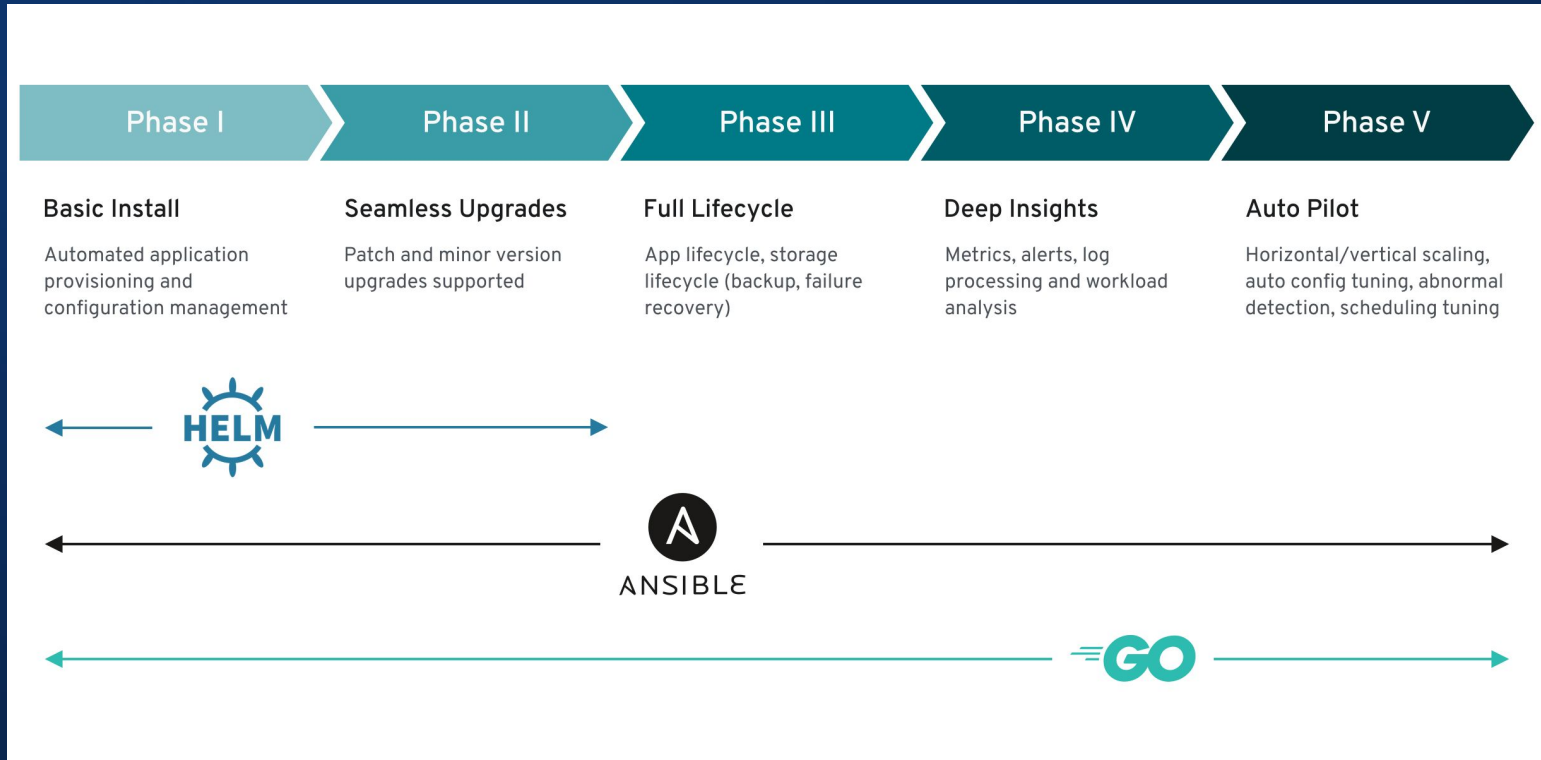
<https://docs.openshift.com/container-platform/4.1/applications/operators/olm-what-operators-are.html>

Core to OpenShift runtime

```
$ oc get console cluster -n openshift-console-operator -o yaml
apiVersion: config.openshift.io/v1
kind: Console
metadata:
  annotations:
    release.openshift.io/create-only: "true"
  creationTimestamp: "2019-11-07T17:46:21Z"
  generation: 1
  name: cluster
  resourceVersion: "10767"
  selfLink: /apis/config.openshift.io/v1/consoles/cluster
  uid: 823c1a2a-0186-11ea-88e3-129c2838f2c9
spec: {}
status:
  consoleURL: https://console-openshift-console.apps.benstest.os4-sandbox.paas.redhat.com
$ oc get pods -n openshift-console-operator
```

NAME	READY	STATUS	RESTARTS	AGE
<u>console-operator-5986b8689f-mt5b8</u>	1/1	Running	1	32h

But we're here to talk Ansible



Source:



<https://github.com/operator-framework/operator-sdk#operator-capability-level>

Simple to get started

- ❑ watches.yml – Tell the operator what API resources to look for, and which roles/playbooks to execute
- ❑ roles/ – Any roles the operator will use for management
- ❑ build/ – How to build the operator container image (mostly hands off)
- ❑ deploy/ – How to deploy the operator container, including service account and API resource authorization

Source:

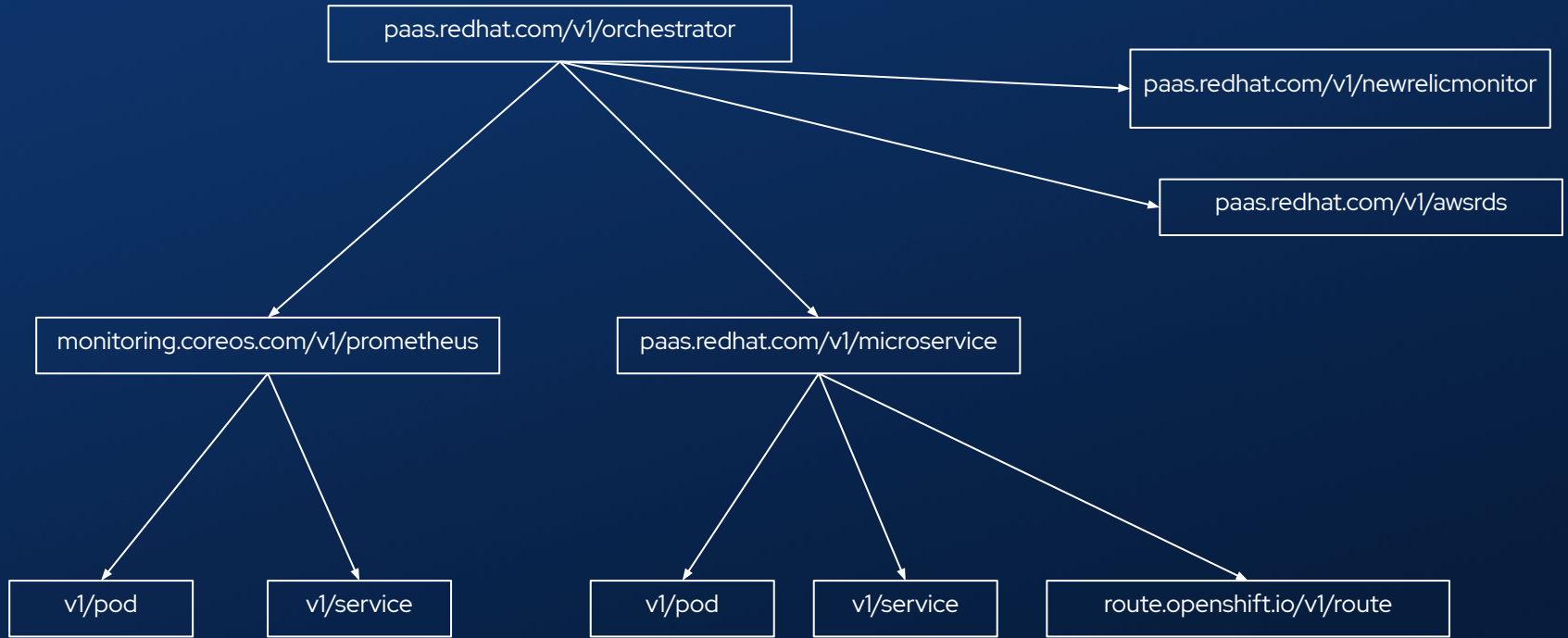
❑ https://github.com/operator-framework/operator-sdk/blob/master/doc/ansible/project_layout.md

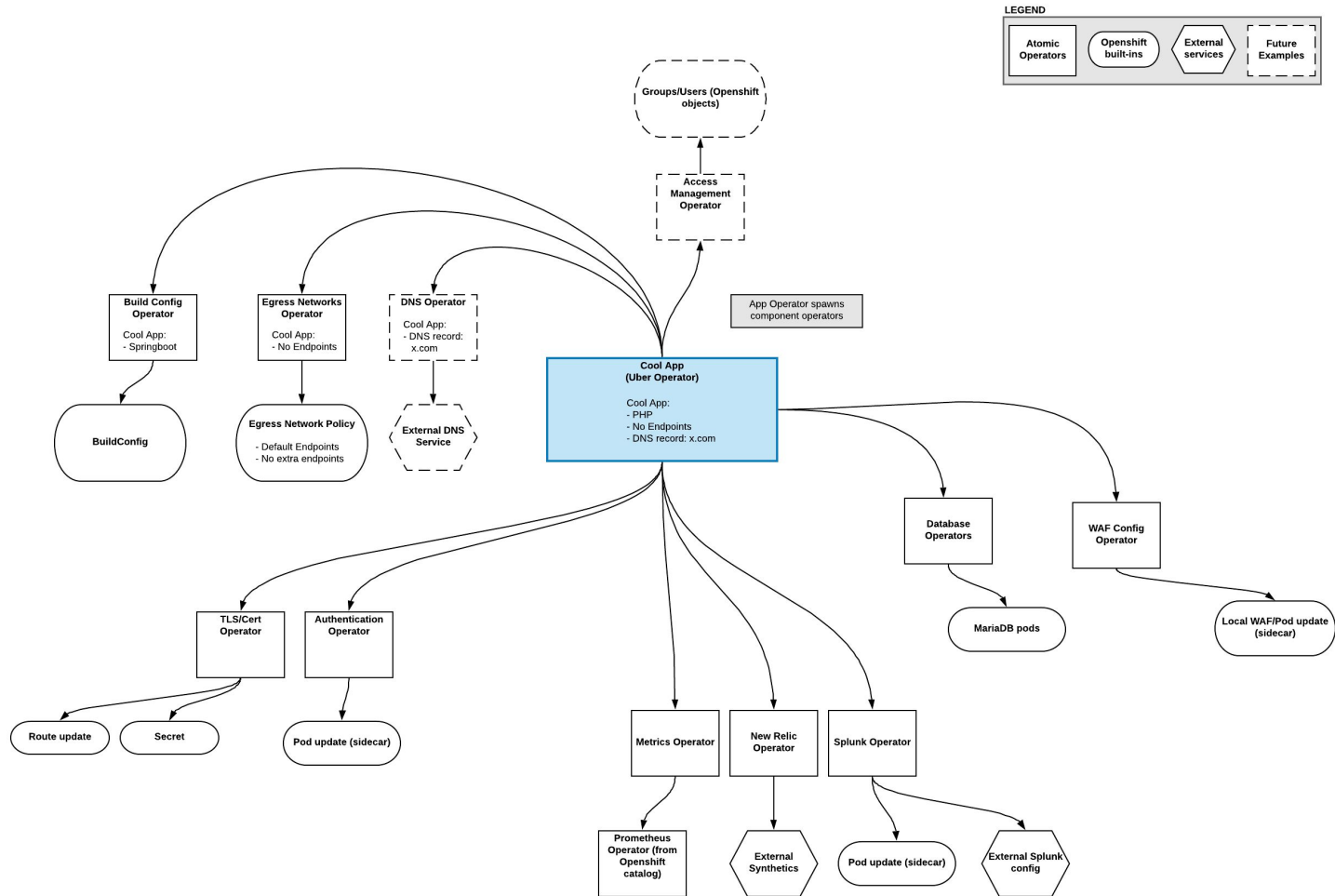
Reusing roles/playbooks for operators

```
! aws_rds.yaml
1 # Playbook: aws_provision_rds.yaml
2 #
3 # --- Create RDS param group(s), subnet group(s) and database(s)
4 #
5 ---
6 - name: Provision RDS parameter groups # must be done first
7   connection: local
8   hosts: all
9   roles:
10  - aws_provision_rds_param_group
11
12
13 - name: Provision RDS subnet groups # must be done second
14   connection: local
15   hosts: all
16   roles:
17  - aws_provision_rds_subnet_group
18
19
20 - name: Provision RDS instances
21   connection: local
22   hosts: all
23   roles:
24  - awsrds
```



Dependent Custom Resources





Thank you!

Red Hat on Red Hat

<https://www.redhat.com/en/about/red-hat-on-red-hat>



