

Future Directions

An Ansible Developer and Ecosystem Story

Timothy Appnel
Sr. Product Manager, Ansible
GitHub: @tima
Twitter: @appnelgroup

Ansible Quebec/Montreal Meetup
October 22/24, 2019



```
$ git show f31421576b00f0b167cdbc61217c31c21a41ac02
```

```
Author: Michael DeHaan <michael.dehaan@gmail.com>
```

```
Date: Thu Feb 23 14:17:24 2012 -0500
```

```
Genesis.
```

```
diff --git a/README.md b/README.md
```

```
new file mode 100644
```

```
index 0000000000..60bbc9f813
```

```
--- /dev/null
```

```
+++ b/README.md
```

```
@@ -0,0 +1,88 @@
```

```
:
```

THE ANSIBLE WAY



SIMPLE

- Human readable automation
- No special coding skills needed
- Tasks executed in order
- Usable by every team
- Get productive quickly**



POWERFUL

- App deployment
- Configuration management
- Workflow orchestration
- Network automation
- Orchestrate the app lifecycle**



AGENTLESS

- Agentless architecture
- Uses OpenSSH & WinRM
- No agents to exploit or update
- Get started immediately
- More efficient & more secure**

OPTIMIZED FOR CONTRIBUTION

“Modularity and option value create opportunities for the exchange of valuable work among developers, opportunities that do not exist in codebases that are not modular or have no option value.”

SOURCE: The Architecture of Participation. Carliss Y. Baldwin, Kim B. Clark, Harvard Business School, Oct. 2005.

BATTERIES INCLUDED

Ansible ships with a rich and versatile standard library which is immediately available without making the user download separate packages.

cloud

clustering

containers

cryptography

database

DNS

files

inventory

identity

messaging

monitoring

network

notifications

packaging

security

source control

storage

system

testing

utilities

web infrastructure

windows

IT HAS BEEN WORKING

Octoverse 2018: #7 for unique contributors (~4100)

14,000

FORKS

35,000

STARS

40k

COMMITTS

“Whenever an individual or a business decides that success has been attained, progress stops.”

Thomas J. Watson

GROWING PAINS

- Volume of contributions is weighing down releases
- Bottlenecks to participation
- Different release cadences of integrations
- Dramatic expansion of the use cases over time
- Opportunities to provide more value to users
- Embedding of Ansible in other projects/products

What Does This Mean?



Ansible is evolving and needs to become
a platform for automation

A few sane options are needed to supplement the ease and simplicity of the “batteries included” distribution model

CURRENT ASSUMPTIONS

Current assumptions under the “batteries included” only distribution model will no longer be valid and cannot be relied on going forward.

- Only `ansible/ansible/devel` matters
- Compatibility, namespace conflicts & quality checks managed thru code review
- Streamlined testing -- only the current version matters
- Functionality is part of a single bundle

ASSUMPTIONS CHANGE

In adjusting to these changes in the assumptions that Ansible has relied upon, these issues now need to be addressed:

- Managing namespace conflicts & incompatibilities
- Initialization performance
- Quality control and testing
- Versioning

NEW REQUIREMENTS

New and additional tooling will have to be developed for the platform ecosystem to thrive

- A New Packaging Format for Ansible Content
- Content Package Management
- Developer Tools
- Enhance Roles
- Enhance and Extend The Galaxy Hub (galaxy.ansible.com)

A NEW PACKAGING FORMAT

Ansible will need a new means of packaging and shipping content that is independent and external to the core engine

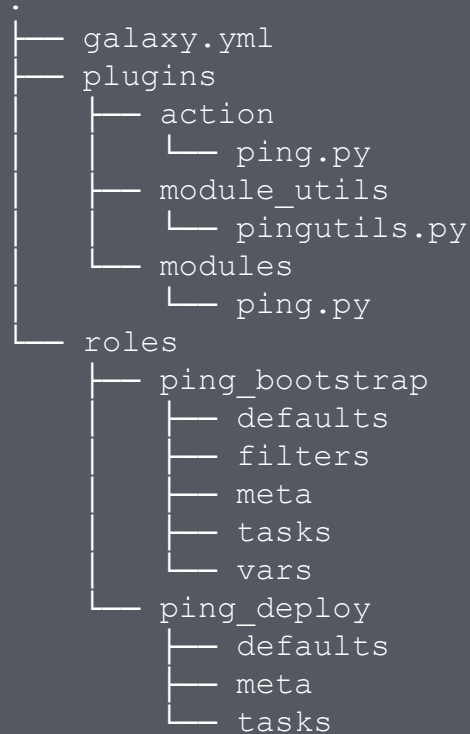
- The Ansible Role packaging format is insufficient
- Simple, lightweight and consistent with the Ansible way
- Can contain multiple Ansible modules, plugins, roles and maybe more

ANSIBLE CONTENT COLLECTION

A new format for delivering content independently of the Ansible core engine

- A new means to distribute Ansible content
 - An artifact format for consistent content structure
 - Contains roles, modules, plugins and module utilities
 - Enables versioning of external content
 - Enables consistent delivery independent of Ansible distributions
 - Installable as system, user or project resources
- Immediate use of the content found within the artifact
- Namespacing support built into the collection

ANSIBLE CONTENT COLLECTION PREVIEW



ANSIBLE CONTENT COLLECTION PREVIEW

```
hosts: somehosts
```

```
collections:
```

- tima.pinger
- redhat.open_ping

```
tasks:
```

- tima.pinger.ping:

- ansible.builtin.ping: # use only the ping packaged in core

- ansible.legacy.ping: # use core or library(etc)/ping.py
when: thing | tima.pinger.filter == 42

- ping: # searches collections “path” otherwise...
still works, == ansible.legacy.ping:

CONTENT PACKAGE MANAGEMENT

You woke up this morning, rolled out of bed, and thought, “Y’know what? I don’t have enough misery and suffering in my life. I know what to do—I’ll write a language package manager!”

- Sam Boyer

<https://medium.com/@sdboyer/so-you-want-to-write-a-package-manager-4ae9c17d9527>

CONTENT PACKAGE MANAGEMENT

The introduction of a new content packaging format requires a means of managing it

- The `ansible-galaxy` CLI was insufficient (and not really package manager)
- We also wanted to experiment and didn't want to risk breaking stuff.
- Minimize the inherent pitfalls of package management
- Prioritize simplicity over flexibility and package management norms
- Focus strictly on management functions and not development

CONTENT PACKAGE MANAGEMENT

The introduction of a new content packaging format requires a means of managing it

- Developed under the project code name “Mazer”
- This requirement was eventually merged back into the ansible-galaxy CLI for v2.9
 - <https://github.com/ansible/mazer>
 - <https://groups.google.com/forum/#!topic/ansible-devel/ChNFHsCTpno>
 - <https://github.com/ansible/ansible/pull/57106>

DEVELOPER TOOLS

New tools are needed to provide an effective and standardized means of developing content that is independent of the core engine

- The Ansible core engine benefited from the extensive developer tools for its own development and testing
- Some additional tools exist expressly for Ansible development as a convenience
- These tools are insufficient outside of the single repo “batteries included” model
- Needs to encourage participation, reuse, stability and best practices in an Ansible way as possible -- being simple, powerful and frictionless.
- Needs to automate and scale more of what’s been done through manual code review by core committers and others

DEVELOPER TOOLS

New tools are needed to provide an effective and standardized means of developing content that is independent of the core engine

- Molecule: An Ansible testing framework
 - <http://github.com/ansible/molecule>
- ansible-test lint and sanity tests integration
 - <https://github.com/ansible/ansible/tree/devel/test/runner>
 - <https://github.com/ansible/ansible/pull/59197>
- ansible-lint: Static analyser for Ansible best practices and other checks
 - <http://github.com/ansible/ansible-lint>
 - Already integrated into new galaxy.ansible.com submissions
 - More rules and continue functionality

ENHANCE & EXTEND THE GALAXY HUB

<https://galaxy.ansible.com/>

- Support for Collections
- Continued search, feedback and UX improvements
- Address the “dash quandary” in namespaces
- Separate community vs customer concerns into separate hubs
 - Launch the Ansible Automation Hub for enterprise subscriber needs
 - Introduce push architecture using Pulp infrastructure

WIP: RESTRUCTURING THE ANSIBLE PROJECT

From a development perspective, Ansible would be broken out into different components

- The core engine
- The core modules and plugins
- The community modules and plugins
- Various supported partner modules and plugins

All of these different components would be built in the form of Ansible Content Collections.

<https://www.ansible.com/blog/thoughts-on-restructuring-the-ansible-project>

WIP: RESTRUCTURING THE ANSIBLE PROJECT

From a deployment perspective, Ansible would be delivered in one of two fundamental ways

- A batteries-included method
- A supported enterprise method

Both of these methods would depend heavily on Ansible Galaxy as the de facto delivery mechanism.

<https://www.ansible.com/blog/thoughts-on-restructuring-the-ansible-project>

PRODUCT DEVELOPMENTS

- Ansible Tower/AWX
 - Scalable Execution Capacity
 - Centralized Automation Console
 - Webhooks and Native GitOps support
- Ansible services via cloud.redhat.com
 - Automation Hub
 - Automation Analytics
- Content Packages -- Network & Security Automation Collections

So when are we getting this?

Ansible 2.8

Collections in tech preview

Partner options:

Certified:

-Modules/plugins in ansible/ansible

Non-Certified:

-Modules/plugins in ansible/ansible

-Collections/roles/modules in Galaxy

-Content distributed independently

Ansible 2.10

Certified Collections Defined

Partner options:

Certified:

-Collections in Galaxy

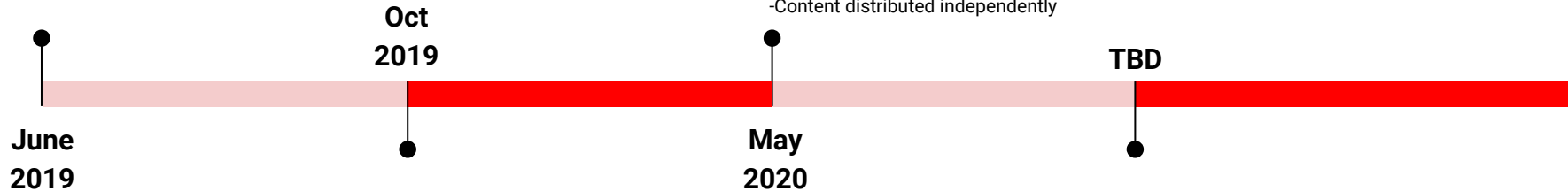
-Modules/plugins in ansible/ansible (existing only - no new content allowed)

Non-Certified:

-Modules/plugins in ansible/ansible (existing only - no new content allowed)

-Collections/roles/modules in Galaxy

-Content distributed independently



Ansible 2.9

Full support for Collections

Partner options:

Certified:

-Collections in Galaxy

-Modules/plugins in ansible/ansible

Non-Certified:

-Modules/plugins in ansible/ansible

-Collections/roles/modules in Galaxy

-Content distributed independently

Ansible 3.0

Partner content removed from Core

Partner options:

Certified:

-Collections in Automation Hub

Non-Certified:

-Collections in Galaxy

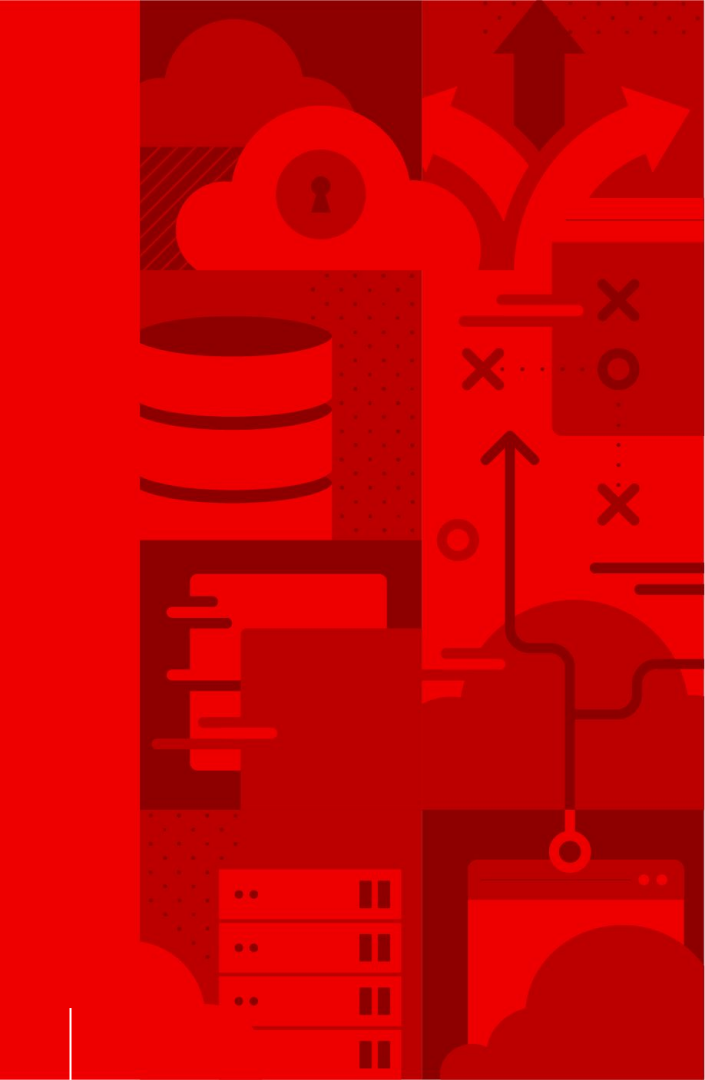
-Content distributed independently

CONTRIBUTING & MORE

Try it out and open issues please

- github.com/ansible/galaxy
- github.com/ansible/ansible-lint
- github.com/ansible/molecule
- galaxy-dev.ansible.com
- #ansible-galaxy on IRC
- #ansible-molecule on IRC

- https://docs.ansible.com/ansible/devel/collections_tech_preview.html
- https://galaxy.ansible.com/docs/contributing/creating_collections.html
- https://galaxy.ansible.com/newswangerd/collection_demo
- <https://galaxy.ansible.com/docs/using/installing.html#collections>



Kubernetes Operators with Ansible

Timothy Appnel
Sr. Product Manager, Ansible
Twitter: @appnelgroup
Github: @tima

Ansible Quebec/Montreal Meetup
October 22/24, 2019



The Ansible Operator SDK makes it easier to deploy and manage Kubernetes applications with native Ansible support

What is Kubernetes?

An open source orchestration system for implementing a microservices architecture as containerized applications run and coordinated across a cluster of nodes.



Red Hat® OpenShift® is a comprehensive enterprise-grade application platform built for containers with Kubernetes at its core.

Stateless is easy,
Stateful is hard

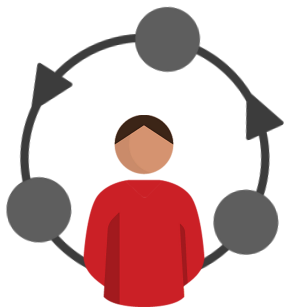
Kubernetes Operators

Operators simplify management of complex applications on Kubernetes



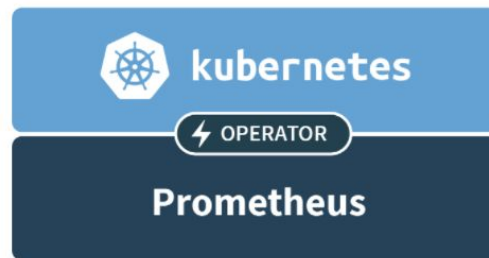
- Encode human operational knowledge
- **Automatically patch, upgrade, recover, and tune container-based apps and services**
- Kubernetes-native
- Purpose-built for a specific application or service
- Enable “day 2” management

Encoding and automating Ops knowledge



WITHOUT OPERATORS: **REACTIVE**

- Continually checks for anomalies
- Alert humans for response
- Requires manual change to fix

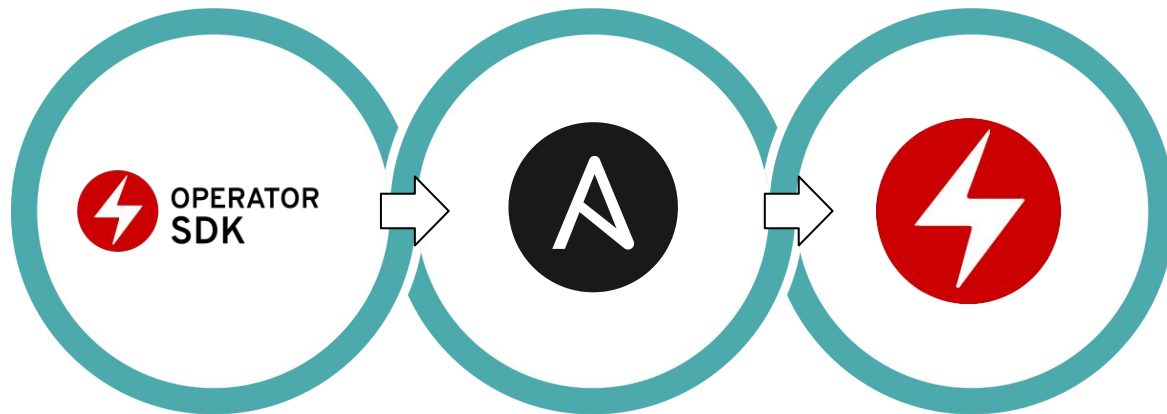


WITH OPERATORS: **PROACTIVE**

- Continually adjusts to optimal state
- Automatically acts in milliseconds

Ansible Operator SDK

Making it easier to deploy and manage Kubernetes apps in an Ansible-native way



operator-sdk new

Use the Operator SDK to create a new skeleton Operator.

Add Ansible Content

Use Ansible Roles and playbooks to manage lifecycle events for your containerized applications.

operator-sdk build

Use the Operator SDK to build and deploy your Operator to Kubernetes.

Why build Operators with Ansible?

EXISTING SKILLS & ECOSYSTEM

Same tried & trusted Ansible tooling

Utilize existing skills

Supports cloud-native & traditional IT automation with one simple language

Leverages vibrant existing ecosystem

LOWER BARRIER OF ENTRY

No programming required

Faster iterations and easier maintenance

Declarative state definitions like K8s

Templating of resources

Abstraction layer & helpers that reduces necessary K8s API experience

Developing your first Operator with Ansible

- Initialize Your Operator With Ansible
 - `$ operator-sdk new null-operator --api-version=cache.example.com/v1alpha1 --kind=Null --type=ansible`
- Automate With Ansible
 - Create new roles and playbooks or reuse an existing one
- Define a watches file
 - Map a Kubernetes object to your Ansible content
- Build Your Operator
 - `$ operator-sdk build null-operator:v0.0.1`
- Deploy Your Operator to a Kubernetes Cluster

Next steps

Get started with Ansible:

ansible.com/get-started

ansible.com/community

Get started with Operators:

github.com/operator-framework/getting-started

ansible.com/operators

More resources

CNCF Webinar: Building Kubernetes Operators in an Ansible-native way

<https://www.cncf.io/webinars/building-kubernetes-operators-in-an-ansible-native-way/>

More resources

```
$ kubectl create -f example/deployment.yaml
```

Install etcd Operator

etcd Operator

A great example of a sophisticated Kubernetes Operator using Ansible:

github.com/water-hole/etcd-ansible-operator

Memcached Operator

Simple walkthrough for building an Operators using the Ansible Operator SDK and Kubernetes CRDs

github.com/operator-framework/operator-sdk-sample/tree/master/memcached-operator

Thank You



[linkedin.com/company/red-hat](https://www.linkedin.com/company/red-hat)



[youtube.com/user/RedHatVideos](https://www.youtube.com/user/RedHatVideos)

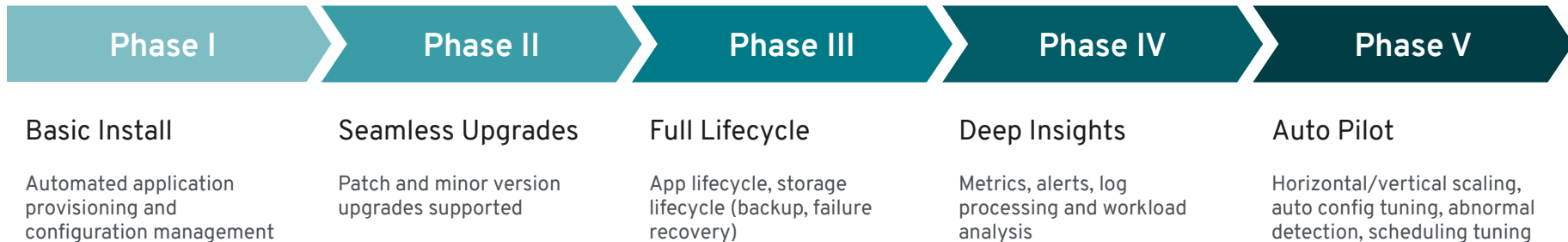


[facebook.com/redhatinc](https://www.facebook.com/redhatinc)

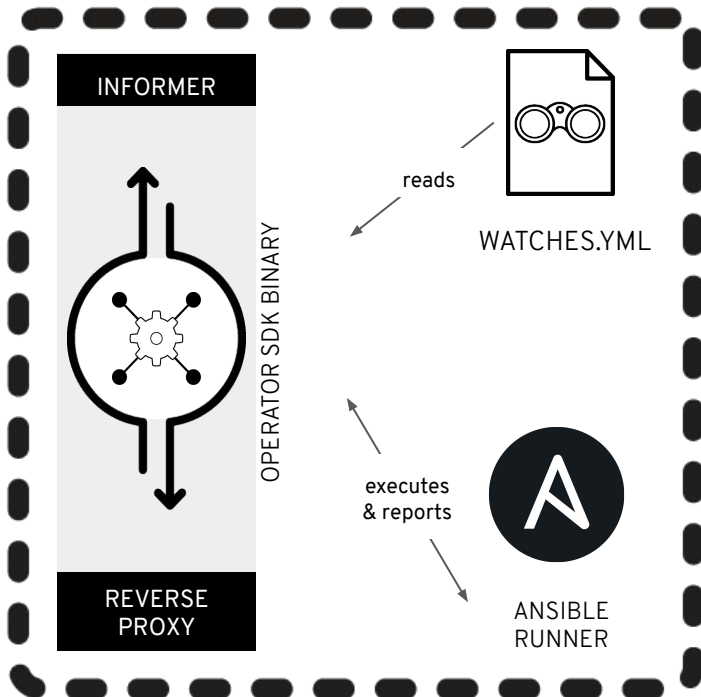
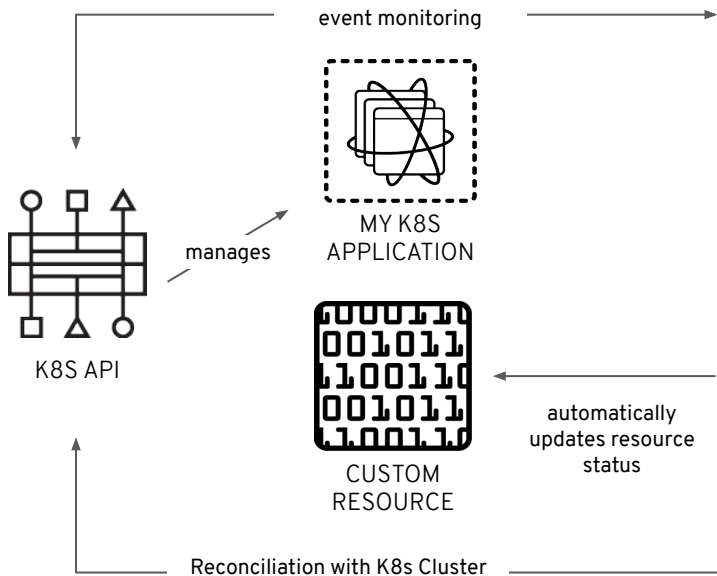


twitter.com/RedHat

Operator capability level



K8s Operator with Ansible



Anatomy of Ansible-enabled Operator image

