Future Directions

An Ansible Developer and Ecosystem Story

Tim Appnel

Senior Product Manager, Ansible

```
$ git show f31421576b00f0b167cdbe61217c31c21a41ac02

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	35,000	40k
FORKS	STARS	COMMITS

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

Thomas J. Watson

GROWING PAINS

- Managing the sheer volume of contributions is weighing down the ability to release
- Bottlenecks to participation in the Ansible ecosystem have formed
- Different release cadences impedes new functionality from reaching users in a timely manner
- Dramatic expansion of the use cases and problem domains that Ansible is now being applied over its inception
- Opportunities exist to provide more value to users by enhancing its utility to extend and reuse automation
- Other projects/products are relying upon and even embedding Ansible for their specific automation needs

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

ASSUMPTIONS CHANGE

Past assumptions under the “batteries included” only distribution model are no longer valid and cannot be relied on going forward.

- Only what is currently in `ansible/ansible/devel` matters
- Compatibility, naming conflicts and quality checks via code review and the watchful-eye of the core committers
- Testing is also streamlined to the current version of Ansible that it is merged
- The vast majority of functionality is included in a single bundle and path (essentially)

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 The Galaxy Hub (galaxy.ansible.com)

A NEW PACKAGING FORMAT

- The Ansible Role packaging format is insufficient, but many of its concepts and conventions can be utilized
- The format needs to be simple, lightweight and consistent with the Ansible way
- It also needs to handle multiple modules, plugins and roles with the potential for other types of Ansible-related content in the future
- This requirement will be addressed by the Ansible Content Collection format
  - https://github.com/ansible/ansible/projects/30#card-13708450

> 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

CONTENT PACKAGE MANAGEMENT

- The `ansible-galaxy` CLI is insufficient (and not really package manager)
- Need to develop a content (package) manager tool to facilitate this functionality properly and new packaging format
- Minimize the inherent pitfalls of package management
- Prioritize simplicity over flexibility and package management norms
- Focus strictly on management functions and not development
- This requirement will be addressed by Mazer
  - https://github.com/ansible/mazer

DEVELOPER TOOLS

- 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
- New tools are needed to provide standardized tooling for developing content that is independent of the core engine.
- Needs to encourages 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 code review by core committers and others

DEVELOPER TOOLS

- Molecule: An Ansible testing framework
  - Lots of improvements and alignment planned -- too many to list
  - Collections support including ansible-test tool refactored and integrated
  - http://github.com/ansible/molecule
- ansible-lint: Static analyser for Ansible best practices and other checks
  - Already integrated into new galaxy.ansible.com submissions with dozens of new rules and more to come
  - More rules and continue functionality
  - http://github.com/ansible/ansible-lint

ENHANCE ROLES

- Need to ease the fork-and-modify pattern of shared roles through making them even more portable and reusable
- Implement “adhoc” role execution
- Implement role argument spec validation similar to modules
    - https://github.com/ansible/proposals/issues/39
    - https://github.com/ansible/ansible/pull/44983
- Provide standardized embedded documentation format functionality of roles

ENHANCE THE GALAXY HUB

https://galaxy.ansible.com/

- Continued search, feedback and UX improvements
- Address the “dash quandary” in namespaces
- Support for Collections and Mazer
- Tech preview push architecture using Pulp infrastructure

So when are we getting this?

WHEN?

We’re at the beginning of a multi-year effort with details emerging over time

Now

- Changes to the Galaxy Hub are continuing to roll out
- Molecule and ansible-lint are now official Ansible projects

Ansible 2.8 (March RC1/May Release)

- Tech Preview of Collections & Mazer
- PluginLoader enhancements[1]
    - Collections Content Path Support
    - Namespace Support
- Role enhancements
    - Argument spec validation
    - “Adhoc” role execution

[1] https://github.com/ansible/ansible/pull/42867

CONTRIBUTING

Try it out and open issues please

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

Thank you

redhat.