ANSIBLE

ansible.com
+1 800-825-0212

WHITEPAPER

Ansible in Depth

INTRODUCTION

Ansible is an open source IT configuration management, deployment, and orchestration tool. It is unique from other management tools in many respects, aiming to provide large productivity gains to a wide variety of automation challenges. While Ansible provides more productive drop-in replacements for many core capabilities in other automation solutions, it also seeks to solve other major unsolved IT challenges. These include clear orchestration of complex multi-tier workflows and cleanly unifying OS configuration and application software deployment under a single banner.

Ansible seeks to keep descriptions of IT workflows understandable and able to be rapidly implemented. This means easy to build, and easy to understand — such that new users can be quickly brought into new IT projects, and longstanding automation content is easily understood even after months of being away from a project. Ansible seeks to make things powerful for expert uses, but equally accessible for all skill levels of user, ensuring a quicker time to market for IT projects and faster, less-error prone turnaround on IT configuration change.

Ansible is designed to be minimal in nature, consistent, secure, and highly reliable, with an extremely low learning curve for administrators, developers, and IT managers.

ARCHITECTURE, AGENTS, AND SECURITY

One of the primary differentiators between Ansible and many other tools in this space is one of architecture.

- Ansible by default manages remote machines over SSH, either using a library called paramiko (which is Python based) or OS-native OpenSSH. Support for Kerberized SSH and bastion hosts is included when using OpenSSH.
- Should other forms of transport be desired, transport mechanisms are pluggable. For instance, a 0mq based accelerated transport is provided. There is also a local (networkless) connection type.
- Ansible does not require root access, and can configure things using sudo access if requested.
- Ansible does not require specific SSH keys or dedicated users—it can work with whatever OS credentials users supply and respects the permissions model of your operating system.
- When requested, Ansible will transfer modules to remote nodes, which are run remotely using user supplied credentials, and not left installed on those remote nodes.
- Ansible does not require any server software to be running from a management machine, and runs with the credentials available to that user.
- Ansible does not require any agent software to be running on any remote machines.
- No ports other than the SSH port are required and no additional PKI infrastructure to maintain.
- Those with access to the control server (or source control) cannot make content be pushed out to remote systems (or otherwise command them) without also having credentials on remote systems.
- When Ansible is not managing remote machines zero resources are consumed on those machines.

These attributes together make Ansible ideal for high-security environments or high-performance cases where there are concerns about stability or permanence of a management agent, but are generally useful attributes in all computing areas.

TOOL UNIFICATION

Ansible also is unique in terms of user experience and approach. Ansible is designed to make IT configurations and processes both simple to read or write, even by those untrained in reading those configurations.

While Ansible can accomplish all types of automation tasks, Ansible does not resemble software programming languages, but rather basic textual descriptions of desired states and processes. Further, it attempts to solve multiple overlapping IT automation problems from a single framework, to prevent the need to learn and understand (and glue together) multiple frameworks.

With other traditional approaches, users have typically had to combine many different tools together to cover the basics of managing IT operating systems and software configurations, including:

- A configuration management tool, typically dealing with the base OS, that describes the desired state of a system, but not the process to put it into that state
- A deployment tool, for pushing out hosted application software, and more focused on process
- A task execution tool, for performing tasks immediately that do not fit into the previous models, such as batch server rebooting.

Ansible makes these approaches available in a single tool, and also provides capabilities and characteristics to enable complicated multi-tier application deployment and orchestration workflows.

MODELING ORCHESTRATION WORKFLOWS

As a detailed example, consider a traditional three-tier web application and its environment consisting of:

- application servers
- database servers
- content servers
- load balancers
- a monitoring system connected to an alert system such as a pager notification service
- a continuous integration system

In this example, Ansible can easily model a process which:

- consults a configuration/settings repository for information about the involved servers
- configures the base OS on all machines and enforces desired state convergence
- identifies a portion of the web application servers to update
- signals the monitoring system of an outage window prior to bringing the servers offline
- signals load balancers to take the application servers out of a load balanced pool
- deploys or updates the web application servers
- signals the load balancers to put the application servers back into the load balanced pool
- signals the monitoring system to resume alerts on any detected issues on those servers
- repeats this process for remaining application servers in a rolling update process
- repeats these rolling update processes for other tiers such as database or content tiers
- sends email reports and logging as desired when updates are complete

PLAYBOOKS

In Ansible, whether configuring a base OS, modeling an update process, or executing explicit “run now” tasks on remote hosts, all of these configurations are achieved through the same tool.

Configurations are expressible in what Ansible calls “Playbooks”, which are human and machine-parseable YAML data format, making it easy to audit with other programs, and easy for non-developers to read and understand.

EXTENSIBILITY

There are many points of integration that can be used to extend Ansible, including:

- modules that run locally or remotely to configure applications, services, or OS parameters
- new logging callbacks for audit and reporting
- integration with any external data stores
- inventory data retrieved from CMDB systems or cloud sources
- new transport mechanisms by which Ansible can communicate with hosts under management

Units of work in Ansible are taken care of by "Ansible modules", which are small programs that run on remote hosts that ensure the given remote host is in a particular state. These modules can be written in any language—such as Python, Perl, Ruby, bash, and so on—so the user can code them using their favorite tools of choice.

By default, core modules are idempotent, which means that they help a system get to a desired state, and if no state change is performed, they perform no action. Ansible also makes it easy to model processes that are not idempotent, and also make it possible to simply just push out simple scripts and run them as desired. Use of idempotent resource modules are assuredly preferred, however, but it is important to be able to model any type of process and not just one that falls into those limits. Ansible is pragmatic in this regard.

CLOUD INTEGRATION

Ansible is capable of easily deploying workloads to a variety of virtualization and public and on-premise cloud environments, including but not limited to VMware, OpenStack, Amazon Web Services EC2 (AWS), Eucalyptus Cloud, KVM, and CloudStack. Machines can be deployed from base OS images – without any modification – and fully configured in one pass.

BIG DATA

Ansible is widely used to deploy big data, storage, and analytics environments, including platforms such as Hadoop, Riak, and Aerospike.

In these environments a wide variety of unconfigured servers must be configured on-premise leaving behind no resource consuming management agents. Customers demand simple controls and easily editable policies to both deploy and update these types of clusters.

Other areas outside of the Big Data space can also take advantage. These properties also make Ansible appealing to high-traffic hosted monitoring systems, where Ansible leaves all CPU resources available to the computing environment when not in use. There are no memory leaks or CPU spikes, nor daemons that need to remain operational.

While there is definitely a cluster of Ansible users around these areas of high-tech computing, these properties are a benefit to any user seeking IT management solutions.

ADVANCED FEATURES

Ansible offers many other numerous configuration modeling features including conditional execution of tasks, ability to gather variables and information from the remote system, ability to spawn asynchronous long running actions, ability to operate in either a push or pull configuration, a "check" mode to test for pending changes without applying change, and the ability to tag certain parts of the configuration so that only certain parts of configuration can be applied. All of these features are covered in the application documentation.

ANSIBLE TOWER

To complement Ansible core, Ansible Inc. has created Ansible Tower. Tower is an enterprise-class automation solution with a simple UI and dashboard, roles-based access control and visual inventory. Ansible Tower unleashes Ansible to your entire team. Tower's fully-featured REST API makes it easy to embed Ansible Tower into your existing infrastructure tools. Try Ansible Tower for free at ansible.com/ansible-tower

DOCUMENTATION

More information about Ansible, including complete documentation, can be found at docs.ansible.com. An open source project mailing list is available and linked on the project site.

EXAMPLES AND FURTHER INFORMATION

Some basic examples of Ansible content implementing zero-downtime rolling updates can be found at https://github.com/ansible/ansible-examples. Users looking to integrate such a process with their source control, establish a build system, or integrate with their network environment may wish to reach out to us for more information.

For more information about Ansible, services, support, and other details, contact AnsibleWorks at info@ansible.com.