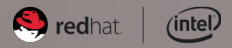
Ansible and Ansible Tower by Red Hat

Manage tasks and processes with automation - using Ansible and Ansible Tower

Jacek Skórzyński
Senior Solution Architect
jacek@redhat.com



Ansible by Red Hat

SIMPLE

- Human readable automation
- No special coding skills needed
- Tasks executed in order
- Get productive quickly

POWERFUL

- App deployment
- Configuration management
- Workflow orchestration
- Orchestrate the app lifecycle

AGENTLESS

- Agentless architecture
- Uses OpenSSH & WinRM
- No agents to exploit or update
- More efficient & secure

Ansible Brand Architecture

This is the Ansible Product Family. It is used to generally refer to Ansible products.

Introducing Red Hat Ansible Engine. Simple, powerful, agentless. Includes full support for execution engine, core modules, and the Red Hat Subscription.

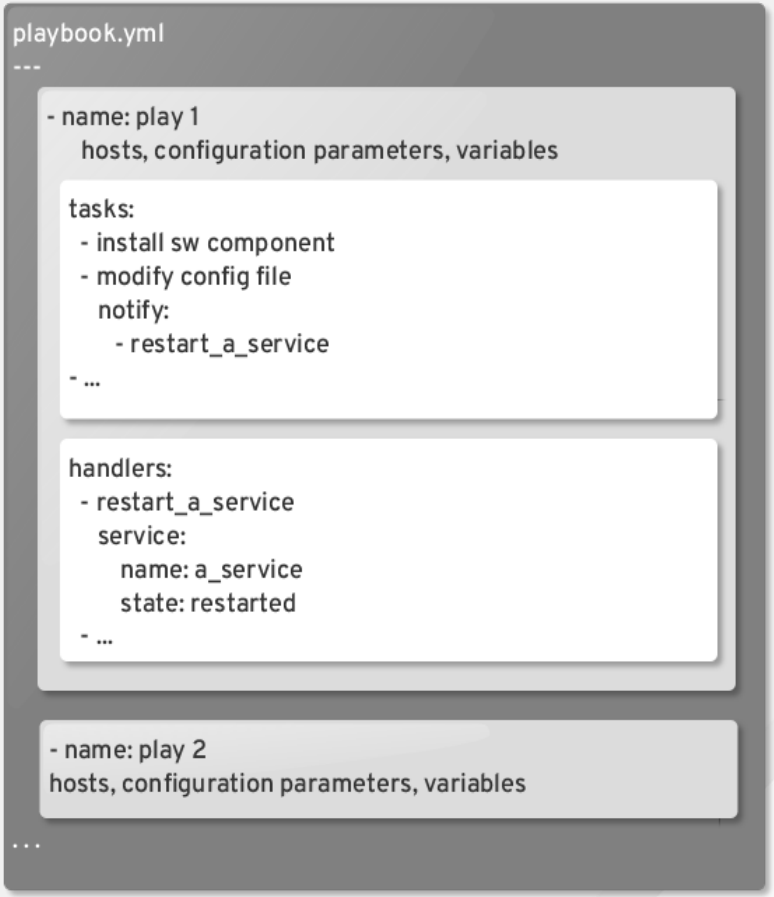
Control, knowledge, delegation for operationalizing enterprise automation environments for teams.

Ansible Playbook

```
---
- name: Update httpd config
  hosts: webservers
  vars:
    http_port: 80
   max_clients: 200
  remote_user: devops
  become: true

  tasks:
  - name: install httpd
    yum: name=httpd state=latest
  - name: write the apache config file
    template: src=/srv/httpd.j2 dest=/etc/httpd.conf
    notify:
    - restart_httpd
  - name: start httpd
    service: name=httpd state=running enabled=true

  handlers:
  - restart_httpd
      service: name=httpd state=restarted
```



redhat. | intel

Ansible Modules

Docs » Module Index

Module Index

- All Modules
- Cloud Modules
- Clustering Modules
- Commands Modules
- Database Modules
- Files Modules
- Inventory Modules
- Messaging Modules
- Monitoring Modules
- Network Modules
- Notification Modules
- Packaging Modules
- Source Control Modules
- System Modules
- Utilities Modules
- Web Infrastructure Modules
- Windows Modules

service - Manage services.

- Synopsis
- Options
- Examples
- This is a Core Module

Synopsis

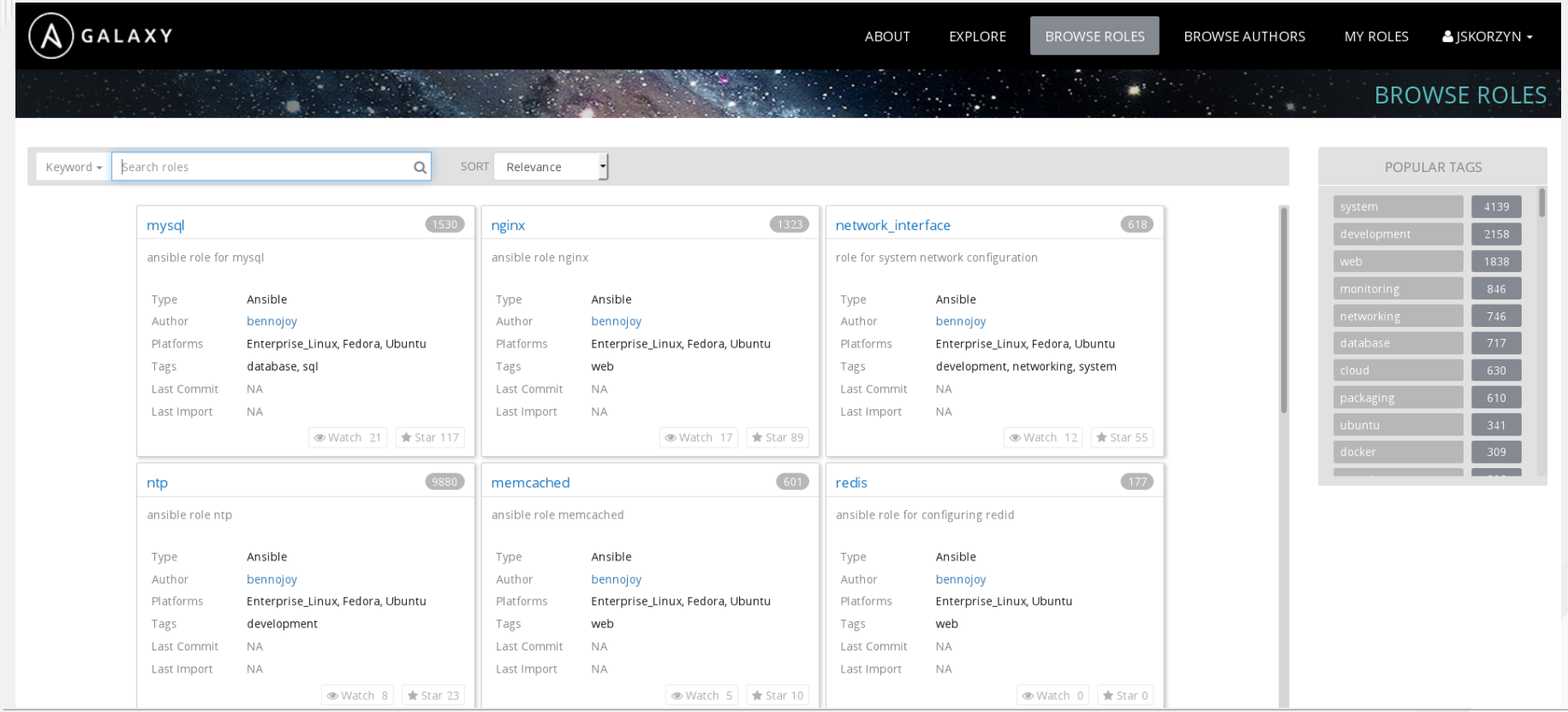
Controls services on remote hosts. Supported init systems include BSD init, OpenRC, SysV, Solaris SMF, systemd, upstart.

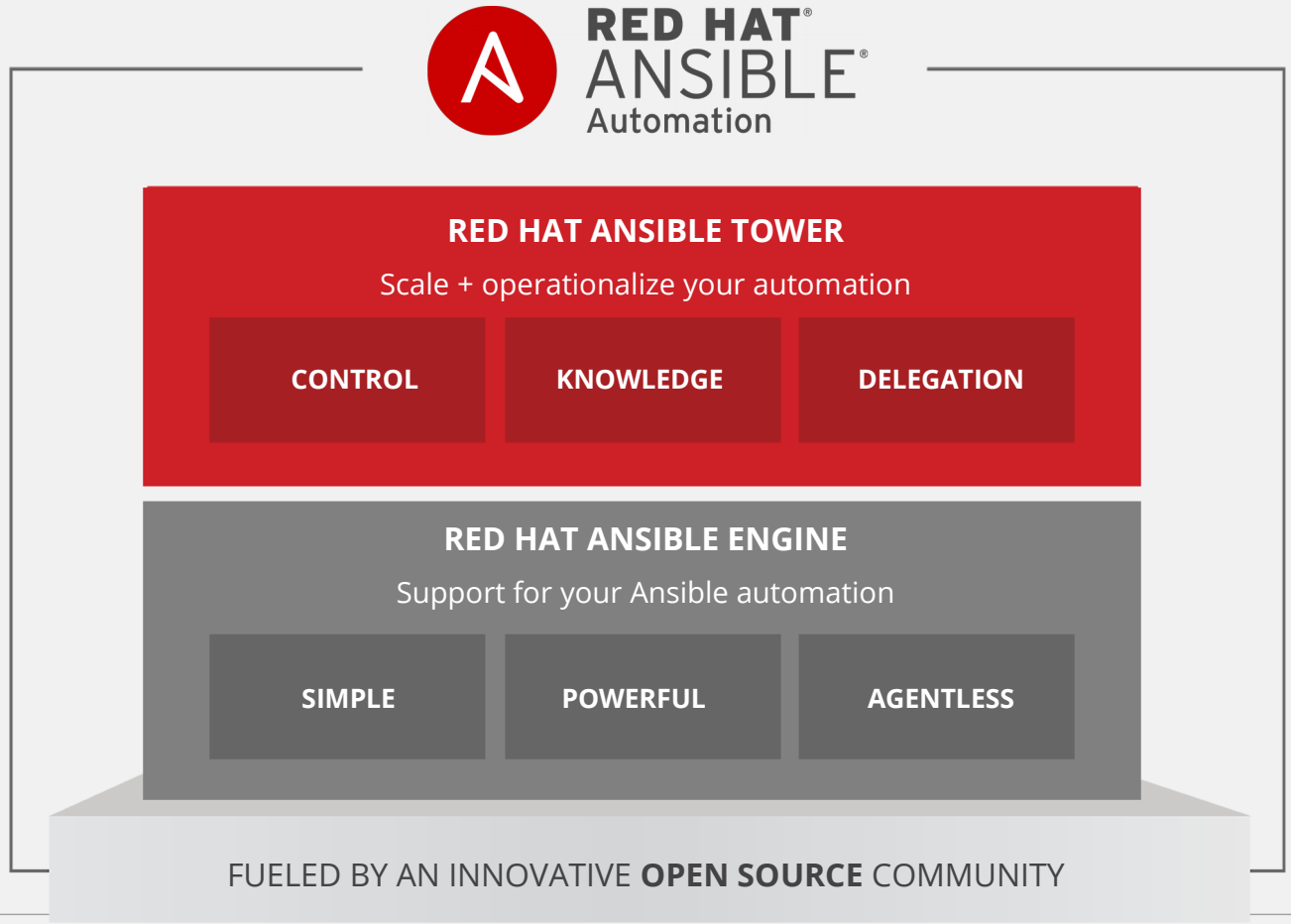
Options

parameter	required	default	choices	comments
arguments	no			Additional arguments provided on the command line aliases: args
enabled	no		• yes • no	Whether the service should start on boot. **At least one of state and enabled are required.**
name	yes			Name of the service.
pattern	no			If the service does not respond to the status command, name a substring to look for as would be found in the output of the *ps* command as a stand-in for a status result. If the string is found, the service will be assumed to be running.
runlevel	no	default		For OpenRC init scripts (ex: Gentoo) only. The runlevel that this service belongs to.
sleep (added in 1.3)	no			If the service is being `restarted` then sleep this many seconds between the stop and start command. This helps to workaround badly behaving init scripts that exit immediately after signaling a process to stop.
state	no		• started • stopped • restarted • reloaded	`started` / `stopped` are idempotent actions that will not run commands unless necessary. `restarted` will always bounce the service. `reloaded` will always reload. **At least one of state and enabled are required.**

redhat. | intel

Ansible Galaxy




RED HAT®
ANSIBLE®
Automation
RED HAT ANSIBLE TOWER
Scale + operationalize your automation
CONTROL
KNOWLEDGE
DELEGATION
RED HAT ANSIBLE ENGINE
Support for your Ansible automation
SIMPLE
POWERFUL
AGENTLESS
FUELED BY AN INNOVATIVE OPEN SOURCE COMMUNITY

Ansible Tower

ACCESS CONTROL
Role-based access control & LDAP integration

DELEGATION OF CREDENTIALS
Delegate credentials without giving away secrets

INVENTORY MANAGEMENT
Graphically manage your internal & cloud resources

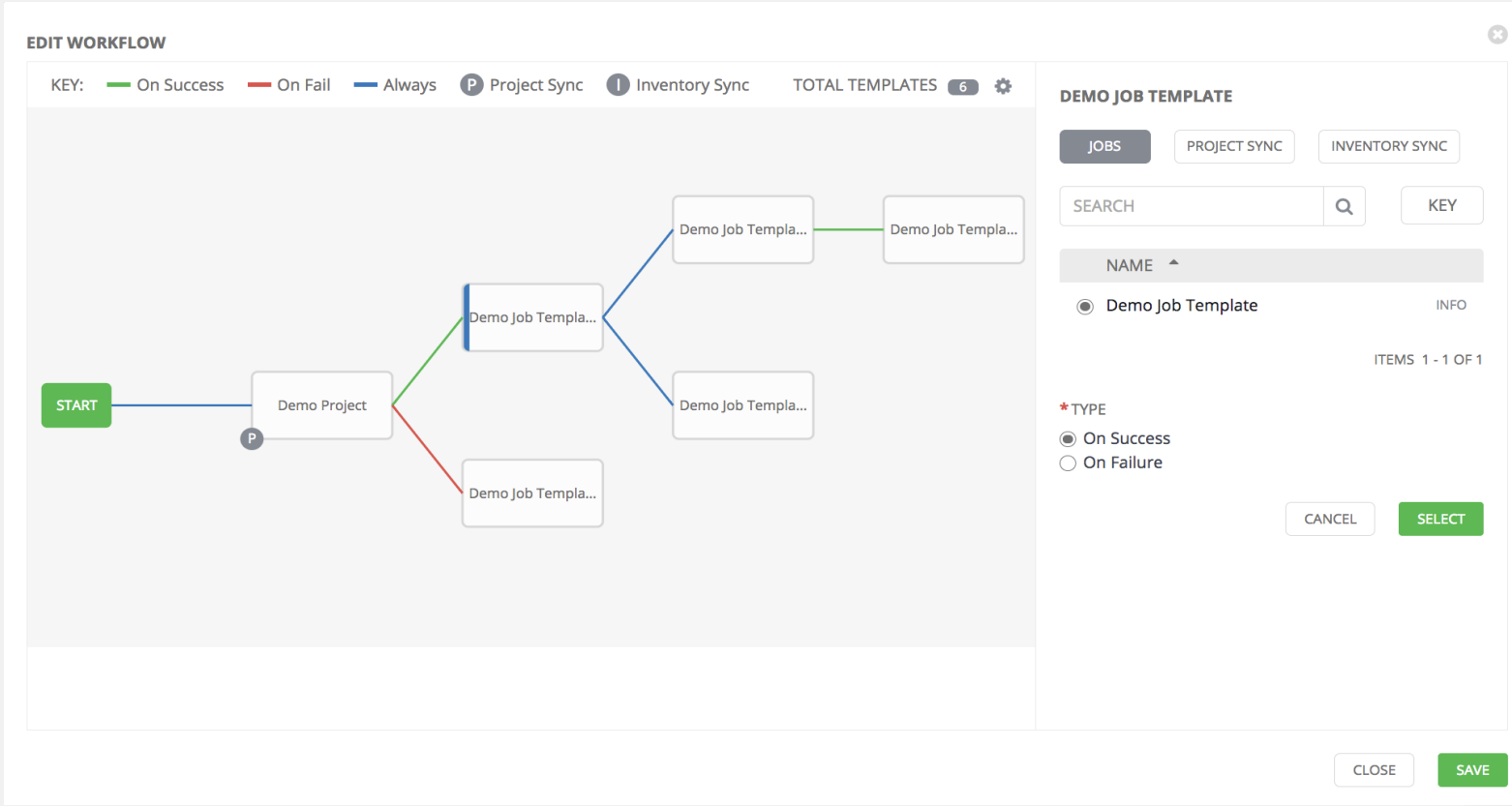
API & CLI
Documented RESTful API and Tower CLI to integrate Tower into your tools

AUDITING
See a full Ansible job history with drill-in details

PUSH-BUTTON LAUNCH
Launch automation jobs with a button

SCHEDULING
Schedule automation jobs (great for periodic remediation)

Ansible Tower Workflows



Ansible Tower

ENTERPRISE LOG INTEGRATION

- Log all Tower activity to central enterprise logging
- Cross-reference automation with events and application logs
- Use Tower's API to perform remediation if needed
- Support for:
  - Elastic
  - Splunk
  - Sumologic
  - Loggly
  - Custom (Via WebHook/RESTful API)

Ansible Tower Clusters

ADD TOWER CAPACITY AND REDUNDANCY WITH EASE

- Add new Tower nodes to scale out Tower job capacity
- Tower node fails? No problem
- Individual Tower jobs will run on any node with available capacity
- Cluster stays in sync with in-Tower configuration

An Automated Ecosystem

RED HAT® SATELLITE

RED HAT® CLOUDFORMS

RED HAT® INSIGHTS

+

RED HAT® ANSIBLE® Automation

POWERS

RED HAT® ENTERPRISE LINUX®

RED HAT® JBOSS® MIDDLEWARE

RED HAT® OPENSHIFT

RED HAT® STORAGE

RED HAT® VIRTUALIZATION

RED HAT® OPENSTACK® PLATFORM

Trainings and Materials

Ansible Courses:

- DO407 - Automation with Ansible
- EX407 - Red Hat Certificate of Expertise in Ansible Automation

Materials:

- Tower Trial - ansible.com/tower-trial
- Getting started – ansible.com/getting-started
- More? - ansible.com/whitepapers

Dziękuję

Jacek Skórzyński
Senior Solution Architect
jacek@redhat.com