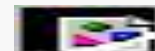# Ansible Tower: Present, Past and Future

Michelle Perz
July 9, 2019

AGENDA

- **Ansible Tower 3.5**
    - UI Enhancements
    - New Features
    - Minor Enhancements
    - Technical Notes
- **Ansible Tower Past Versions**
    - Enhanced workflows
    - Job slicing
    - Smart inventories
    - Multi-LDAP
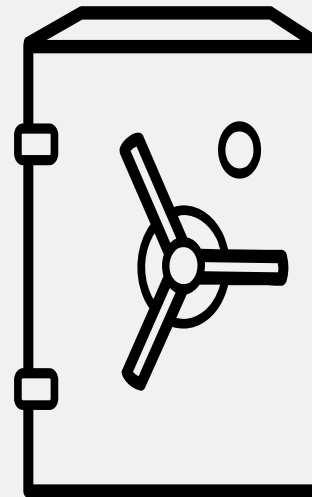- **Ansible Tower Future Directions**

UI Enhancements

New Features
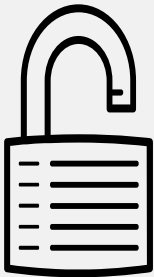
SUPPORT FOR EXTERNAL CREDENTIAL VAULTS

Use credentials from your corporate standard password and key storage directly from Tower.

- HashiCorp Vault
- CyberArk Aim and Conjur
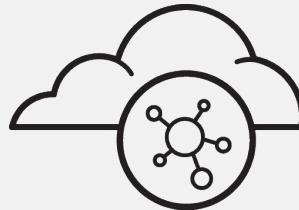- Microsoft Azure Key Vault
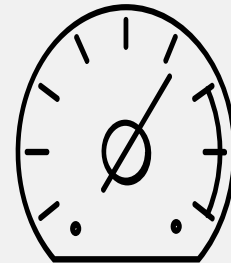
ENABLING ADVANCED ANSIBLE FEATURES

PRIVILEGE ESCALATION

INVENTORY

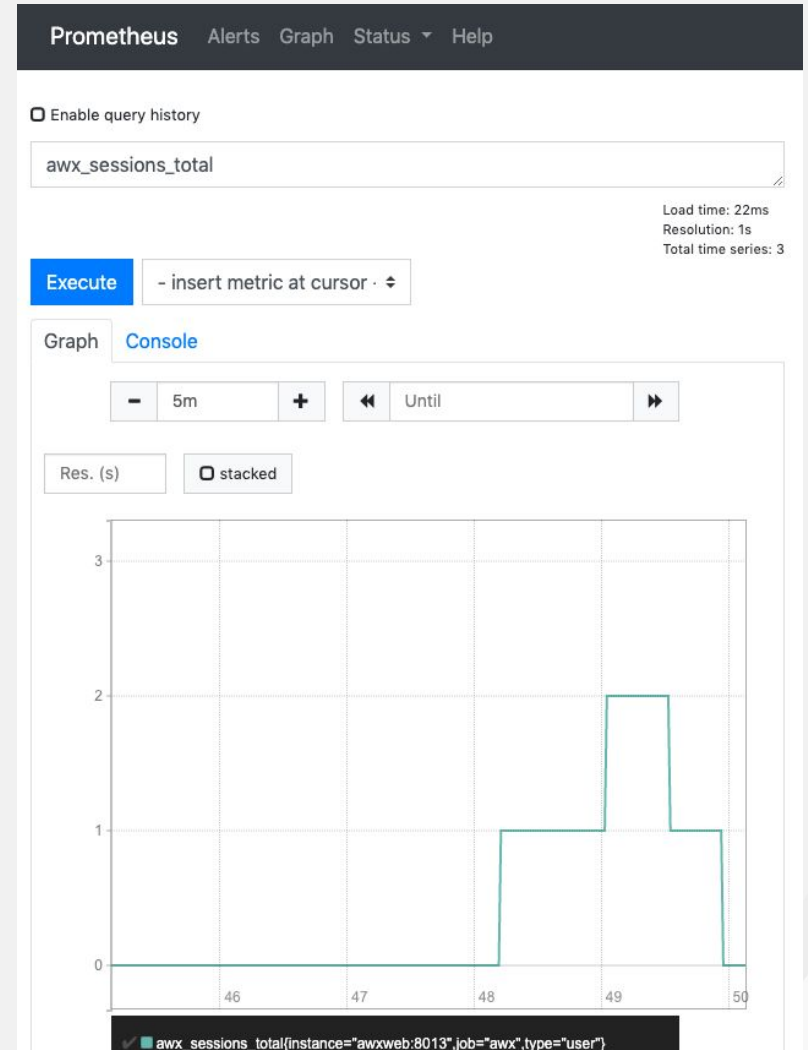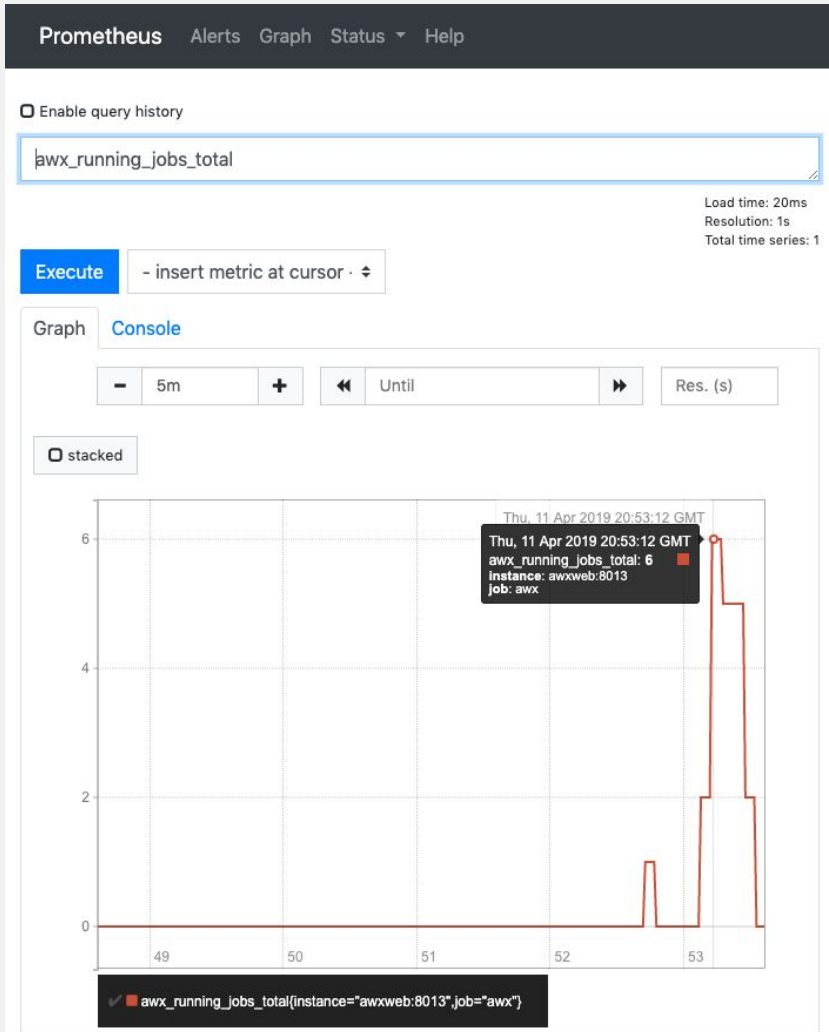**AND MORE**

ENABLING INTELLIGENCE FOR AUTOMATION

SUPPORTING METRICS AND INSIGHTS

You've got questions on your automation.

We'll provide answers.

Metrics and Insights

Red Hat Enterprise Linux 8

SUPPORTING RHEL 8

- Ansible Tower now runs on RHEL 8

- Ansible Tower can manage RHEL 8

redhat.

ASSORTED MINOR ENHANCEMENTS

- License Limits by Organization
- Removed 1024 character limit on --limit fields
- Removed the restriction that schedules must be uniquely named.
- Allow custom virtual environments to be at an alternate path.
- Additional job details shown
    - custom environment used
    - artifacts stored
    - execution node

redhat.

TECHNICAL NOTES FOR OPERATORS

DEPRECATIONS, REMOVALS, AND BEHAVIOR CHANGES

- **Deprecation of /api/v1**
  - Will be removed in 3.6
- **Removal** of deprecated system tracking endpoints
- **Change** to dashboard count behavior.
  - Implicit jobs like SCM checkouts will no longer be counted
- **Change** to playbook detection in projects
  - Playbook names must be valid UTF-8
- **Deprecation** of several inventory computed fields in the API and the dashboard API endpoint.

Past Features

ENHANCED WORKFLOWS

WORKFLOW CONVERGENCE NODES

- Wait for any number of steps to finish before proceeding

- Allows for built-in synchronization points, easy result collection, and simplified error handling

Workflow convergence makes it easier than ever to have your Ansible automation workflows model and match your actual deployment processes.
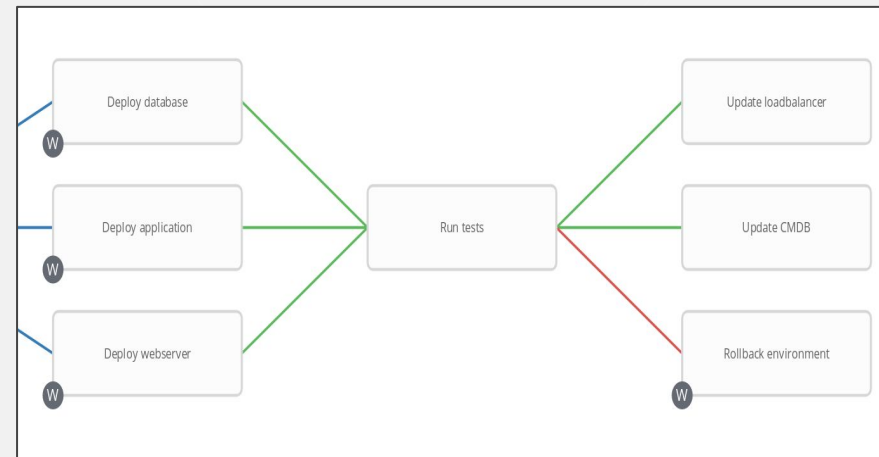
JOB DISTRIBUTION VIA JOB SLICING

BEFORE ANSIBLE TOWER 3.4

- Any playbook run would run one `ansible-playbook` process on one cluster node
- Jobs run across thousands of machines could potentially starve that cluster node's resources, or fail due to memory contention
- Job resizing could be a complicated manual process

WITH TOWER 3.4 JOB SLICING

Jobs have a configurable number of slices. Each slice will be run as a separate `ansible-playbook` run, and slices will be distributed across the Tower cluster.

- Run fact gathering, configuration, and more across thousands of machines with ease
- Increase both job throughput and job reliability

NOTE: Job slicing is only appropriate when each host's automation is independent of other hosts

redhat.

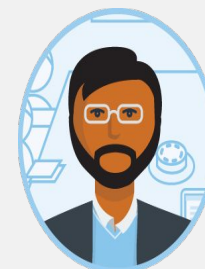# Future Directions

Ansible Tower: Future Directions

AUTOMATION FOR EVERYONE

No playbooks required.

SYSADMIN

OPERATIONS

BUSINESS

ARCHITECT

DEVELOPER

CLOUD ADMIN

NETWORK ADMIN

redhat.
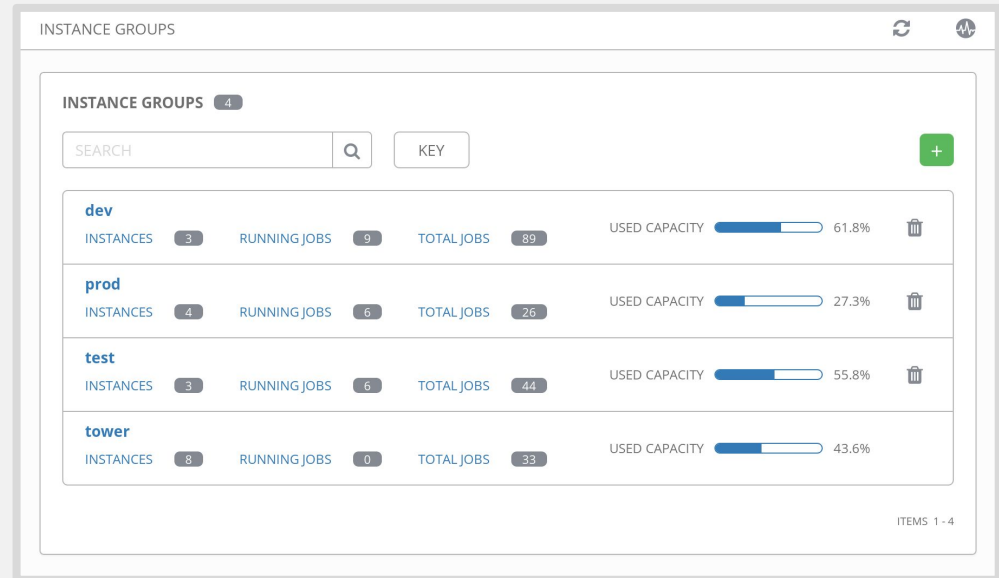
Ansible Tower: Future Directions

BUILDING A GLOBAL AUTOMATION PLATFORM

Capacity where you need it, when you need it, how you need it.

All scalable at runtime.

Questions

THANK YOU