INTRODUCTION TO ANSIBLE TOWER

Ansible Open Source

Simple.

/ Uses YAML for playbooks

/ No special coding skills needed

/ Fast learning curve

Agentless.

/ Uses OpenSSH

/ No extra code to manage

/ Ready for cloud-scale

Powerful.

/ App deployment

/ Orchestration

/ Configuration management

Ansible Enterprise Automation

Ansible Tower

Control.

/ Centralized job runs

/ Job scheduling

/ Automation dashboard

Security.

/ Role-Based Access Control

/ Delegation of credentials/keys

/ Audit trail for automation

Delegation.

/ Push-button job execution

/ Portal mode for delegation

/ REST API for integration

ANSIBLE TOWER

The best way to run Ansible in your organization.

ACCESS CONTROL
Role-based access control & LDAP integration

INVENTORY MANAGEMENT
Graphically manage your internal & cloud resources

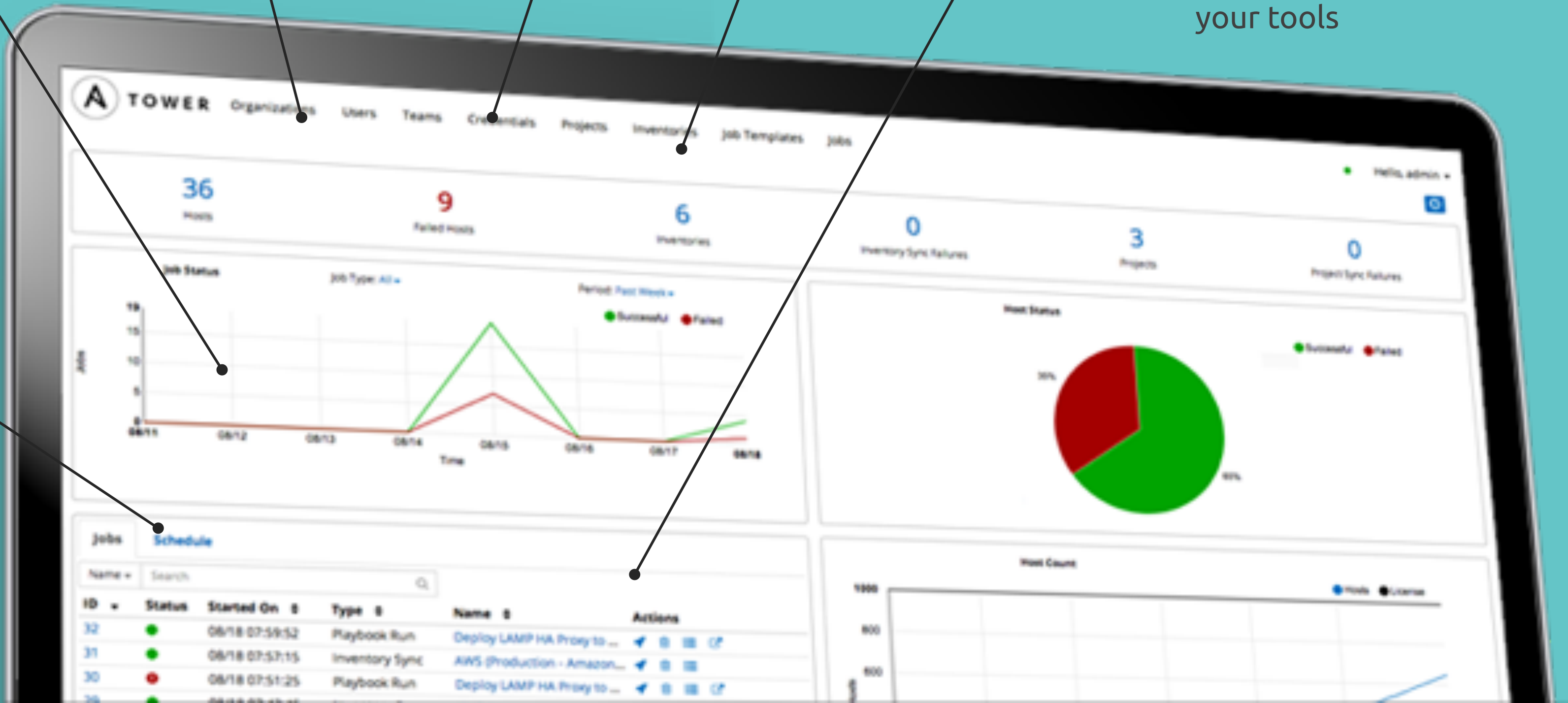
DELEGATION OF CREDENTIALS
Delegate credentials without giving away secrets

PUSH-BUTTON LAUNCH
Launch automation jobs with a button

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

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

SCHEDULING
Schedule automation jobs (great for periodic remediation)



ID	Status	Started On	Type	Name	Actions
32	●	08/18 07:59:52	Playbook Run	Deploy LAMP HA Proxy to ...	
31	●	08/18 07:57:15	Inventory Sync	AWS (Production - Amazon...	
30	●	08/18 07:51:25	Playbook Run	Deploy LAMP HA Proxy to ...	

ANSIBLE TOWER

The best way to run Ansible in your organization.

...AND MORE

- Cloud & Autoscaling Flexibility
- Backup & Restore
- Ansible Galaxy Integration
- Inventory Support for Open Stack
- Remote Ad-Hoc Command Execution
- System Tracking
- External Authentication

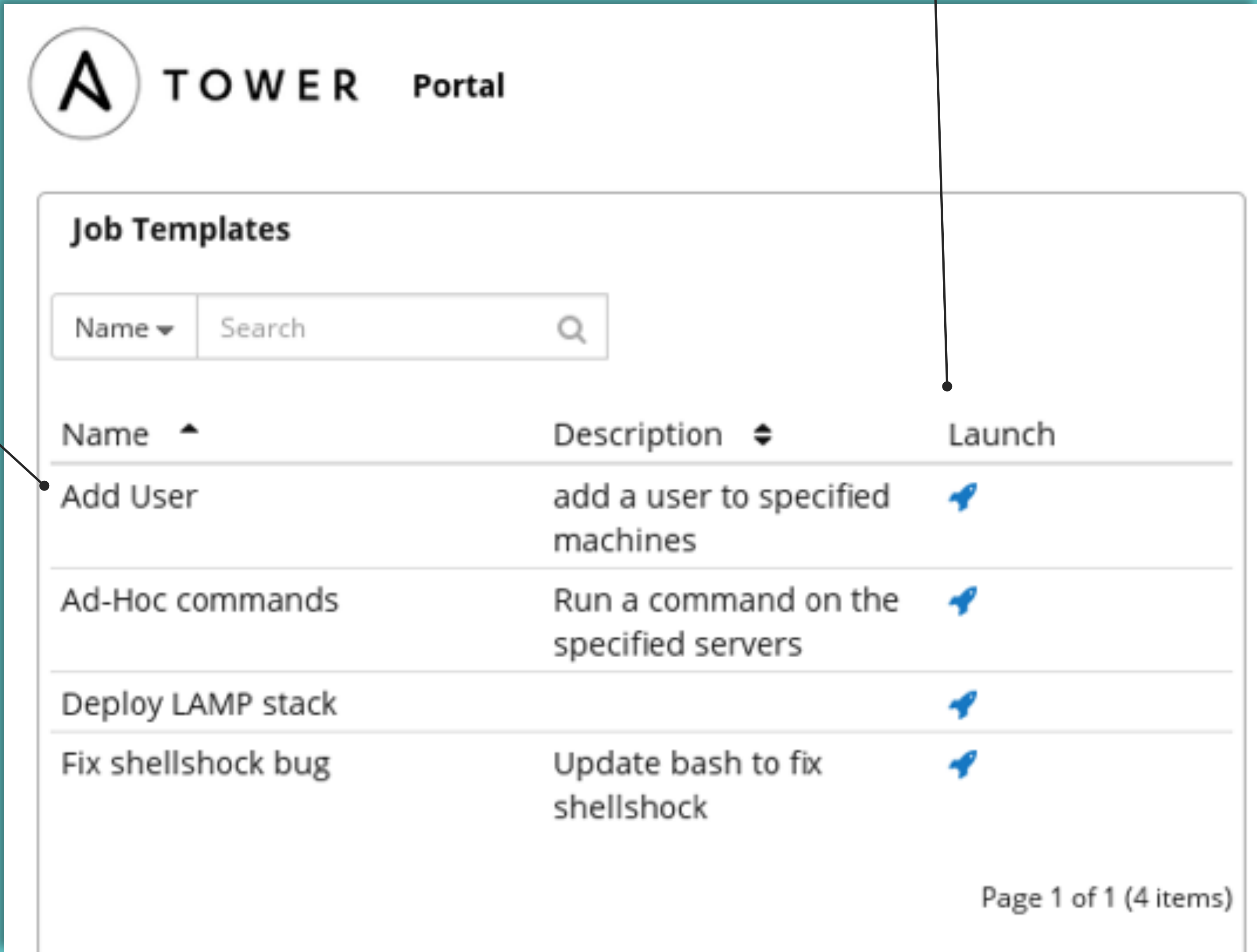
TOWER DELEGATION

Self-service IT made simple. Delegate safely.

LAUNCH BUTTON
Run the Ansible automation without knowing Ansible.

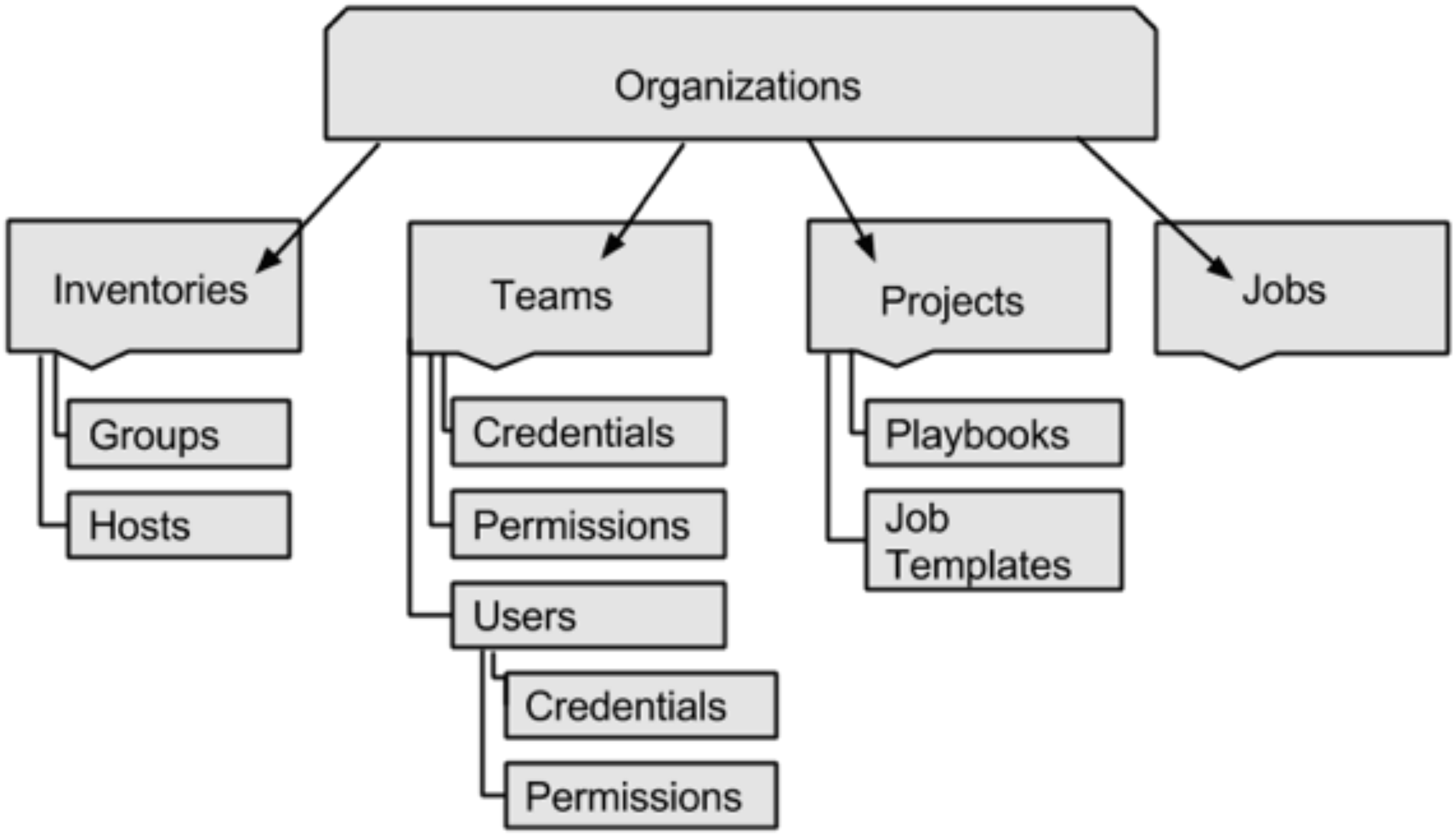
ONLY SEE THE JOBS YOU ARE ALLOWED TO LAUNCH
Tower's role-based access control limits user to this one simple view.

SEE THE RESULTS
The user still gets to see whether the job was successful or not.



Name	Description	Launch
Add User	add a user to specified machines	🚀
Ad-Hoc commands	Run a command on the specified servers	🚀
Deploy LAMP stack		🚀
Fix shellshock bug	Update bash to fix shellshock	🚀

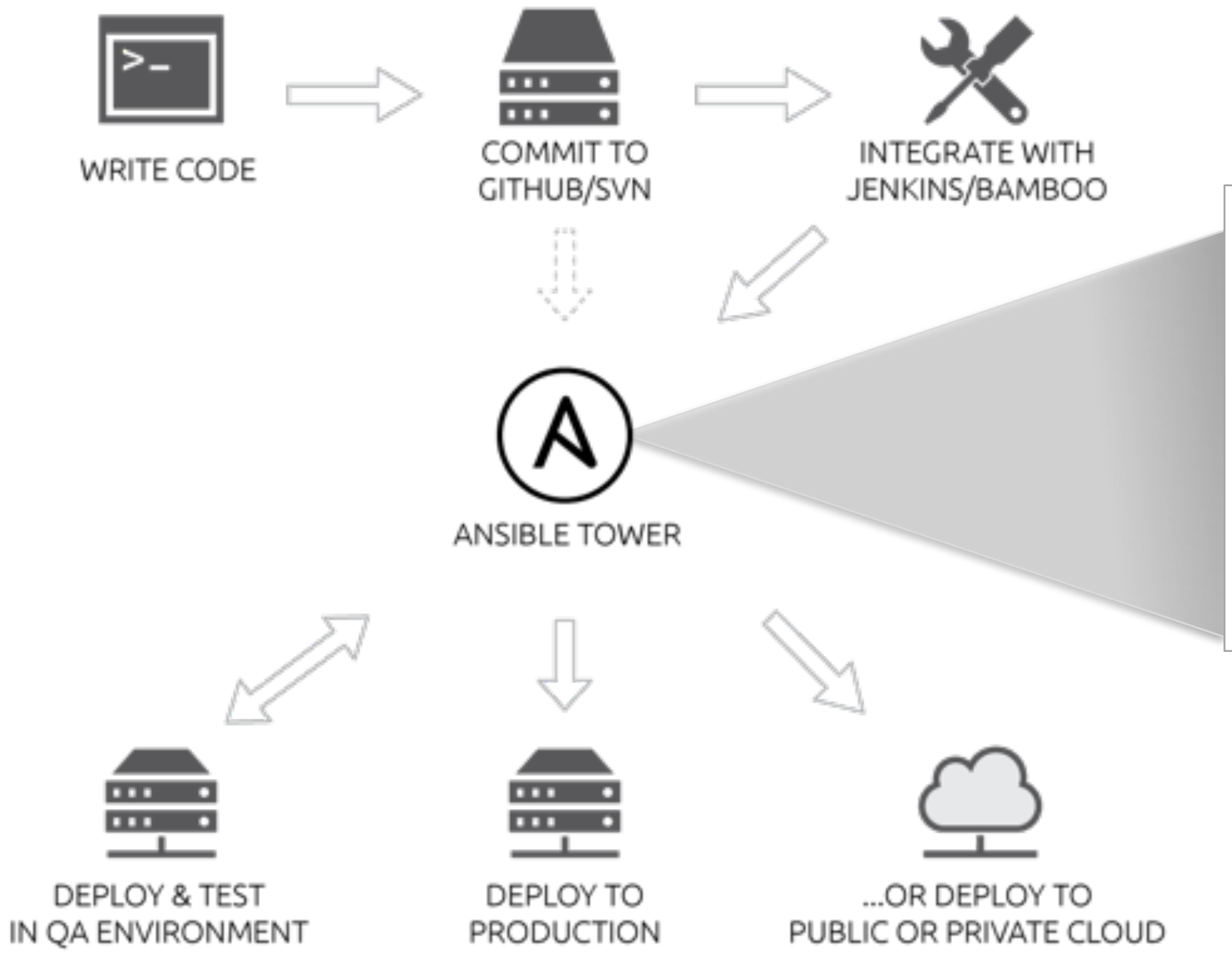
Tower Objects



Demo Time

- Tower Interface
- Dashboard
- Settings
- Job Templates, Jobs and Surveys
- Activity Streams
- Ad-Hoc Commands

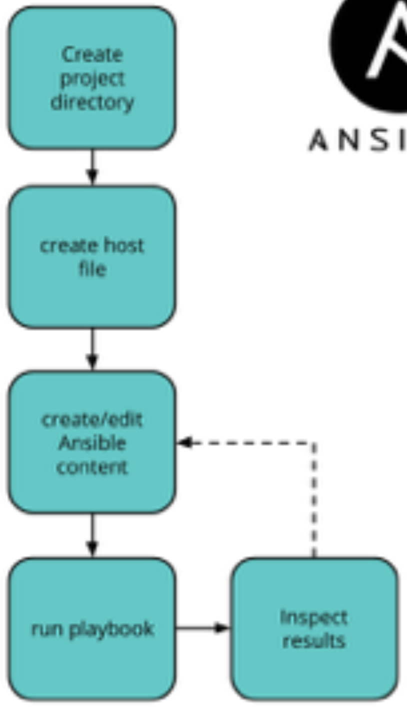
ANSIBLE & CONTINUOUS DELIVERY


WRITE CODE
COMMIT TO GITHUB/SVN
INTEGRATE WITH JENKINS/BAMBOO
ANSIBLE TOWER
DEPLOY & TEST IN QA ENVIRONMENT
DEPLOY TO PRODUCTION
...OR DEPLOY TO PUBLIC OR PRIVATE CLOUD

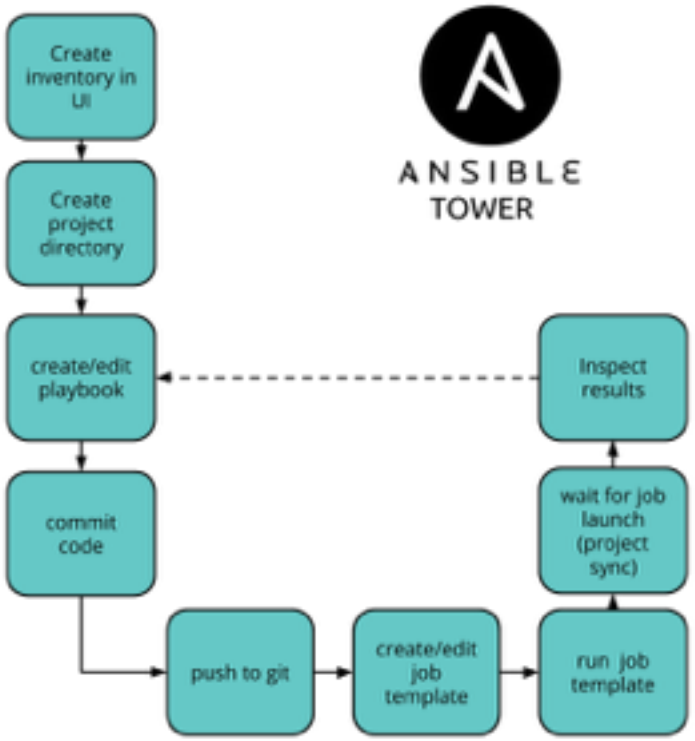
Ansible Tower **connects development to operations** by orchestrating complex application environments:

- Application Code
- OS Updates
- Web Servers
- Databases
- Load Balancers
- Networking
- VM's or Cloud Instances
- … and more

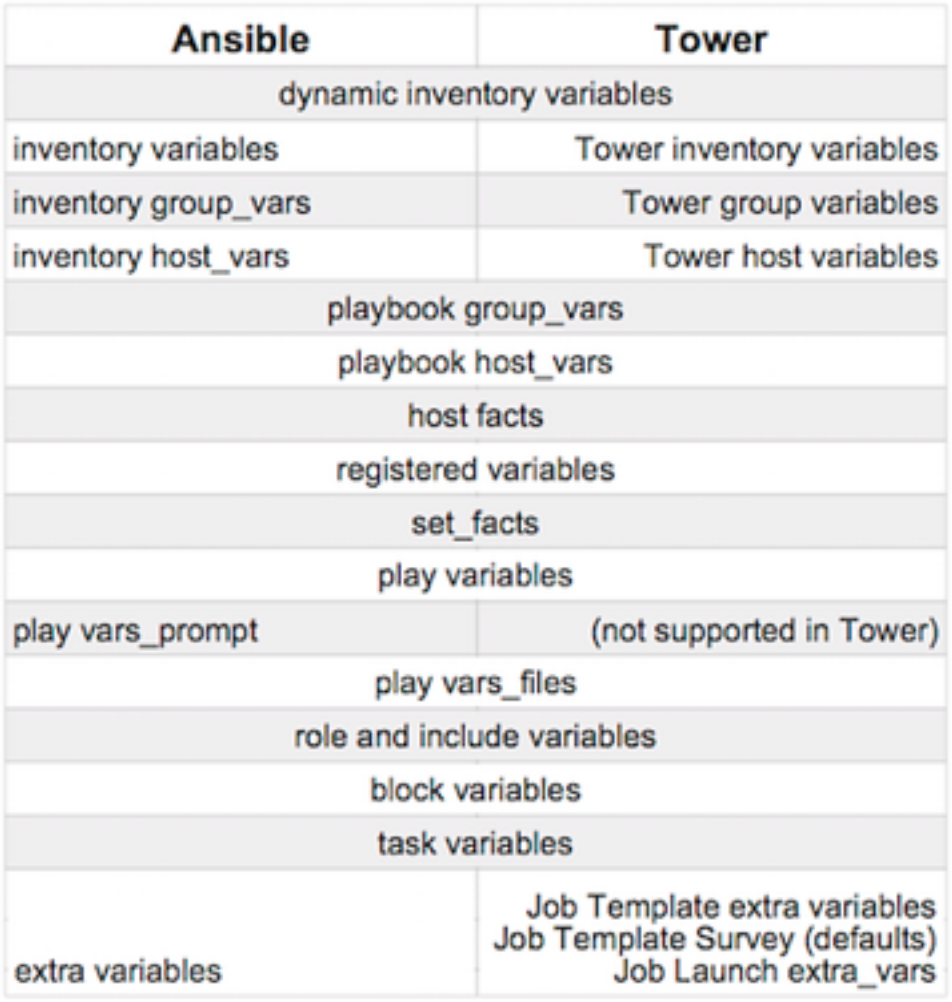
Development Workflow



Development Workflow



Variable Precedence



Ansible	Tower
dynamic inventory variables	
inventory variables	Tower inventory variables
inventory group_vars	Tower group variables
inventory host_vars	Tower host variables
playbook group_vars	
playbook host_vars	
host facts	
registered variables	
set_facts	
play variables	
play vars_prompt	(not supported in Tower)
play vars_files	
role and include variables	
block variables	
task variables	
extra variables	Job Template extra variables Job Template Survey (defaults) Job Launch extra_vars

Core vs Tower: Gotchas

- Tower does not support vars_prompt. Use Job Template Surveys instead.
- Tower does not work directly with static inventory host_vars and groups_vars.
- Playbooks in Tower have limited filesystem visibility (proot) that Core will not.
- Vendored python dependencies can produce different outcomes from running core from the shell.

Workshop: Installing Tower

1. Install Ansible Tower on your controller machine:
2. Rename the default organization to your "*yourname* Training"
3. Create an "Ansible Training" project
4. Enter your credentials
5. Setup your inventory source
6. Run a "ping" as an ad-hoc command
7. Setup and run your playbook from your previous workshop assignment

The Tower REST API

A RESTful API for a systems management application, with all resources fully discoverable, paginated, searchable, and well modeled.

API Browser

Tower features a styled browser the allows API exploration from the API root.

`http://<tower server name>/api/`

Displays every resource, relation and action available in the user interface that can be done in the API and more.

API Conventions

- JSON over HTTP
- Rooted at **`/api/`**
- Versioned for compatibility **`/api/v1/`**

API Conventions: Sorting

- Lists can be returned using the **`order_by`** query string parameter on the GET request:

```
http://tower.host/api/v1/groups/?order_by=modified
```

- Prefix with a dash to reverse the sort:

```
http://tower.host/api/v1/groups/?order_by=-modified
```

- Multiple sorting fields get delimited with a comma:

```
http://tower.host/api/v1/groups/?order_by=modified,total_hosts
```

API Conventions: Searching

- Case-insensitive search within all designated text fields of a model using the search **`query`** string parameter:

```
http://tower.host/api/v1/group/?search=findme
```

API Conventions: Filtering

- Any collection can be filtered via various operators.
- Based on Django's "querysets"

`https://docs.djangoproject.com/en/dev/ref/models/querysets/`

API Conventions: Filtering Lookups

- exact
- iexact
- contains
- icontains
- startswith
- istartwith
- endswith
- iendswith
- regex
- iregex
- gt
- gte
- lt
- lte
- isnull
- in

API Conventions: Filtering

- Find an exact match for group "foo"

```
http://tower.host/api/v1/groups/?name=foo
```

- Find groups that contain the name "foo":

```
http://tower.host/api/v1/groups/?name__contains=foo
```

- Filter against multiple fields at once:

```
http://tower.host/api/v1/arbitrary_resource/?
user__firstname__icontains=john&group__name_icontains=foo
```

API Conventions: Pagination

- Responses in the API are paginated since a collection could return tens or hundreds of thousands of objects.

API Conventions: Pagination

```
{
    "count":200,
    "next":"http://testserver/api/v1/some_resource?page=2&page_size=100",
    "previous":None,
    "results":[
        ...
    ]
}
```

API Endpoints

- Organizations
- User
- Project
- Team
- Credential
- Inventory
- Inventory Script
- Inventory Source
- Group
- Host
- Job Template (+ Unified)
- Job (+ Unified)
- Ad Hoc Command
- System Job Template
- Schedules
- Auth Token
- Ping
- Configuration
- Dashboard
- Activity Stream

tower-cli

- A command line tool for Ansible Tower.
- It can also be used as a client library for other python apps, or as a reference for others developing API interactions with Tower's REST API.