MULTI CLOUD AS CODE WITH ANSIBLE & TOWER

Enterprise Grade Automation

David CLAUVEL - Cloud Solutions Architect
Twitter: **@automaticdavid**
December 2018

redhat.

AUTOMATE ~~REPEAT~~ IT

AGENDA - TOOLING THE DEVOPS PRACTICE

- ➔ What is Ansible ?
- ➔ What is Ansible Tower ?
- ➔ Do you DevOps ?
- ➔ Demo

  Demo: Multi-cloud Automation
- ➔ Network Automation

WHAT CAN YOU DO WITH ANSIBLE

Automate the deployment and management of your entire IT footprint.

Do this...

Orchestration	Configuration Management	Application Deployment	Provisioning	Continuous Delivery	Security and Compliance

On these...

Firewalls	Load Balancers	Applications	Containers	Clouds
Servers	Infrastructure	Storage	Network Devices	**And more...**

redhat.

WHY ANSIBLE?

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

v1 - Set config file to use on boot
1. Write multiple configuration files
• For each environment/region
2. Inspect metadata on boot and use the matching config file

31,000+
Stars on GitHub

1900+
Ansible Modules

500,000+
Downloads / month

HOW DOES ANSIBLE WORK ?

➔ SIMPLE
YAML playbooks

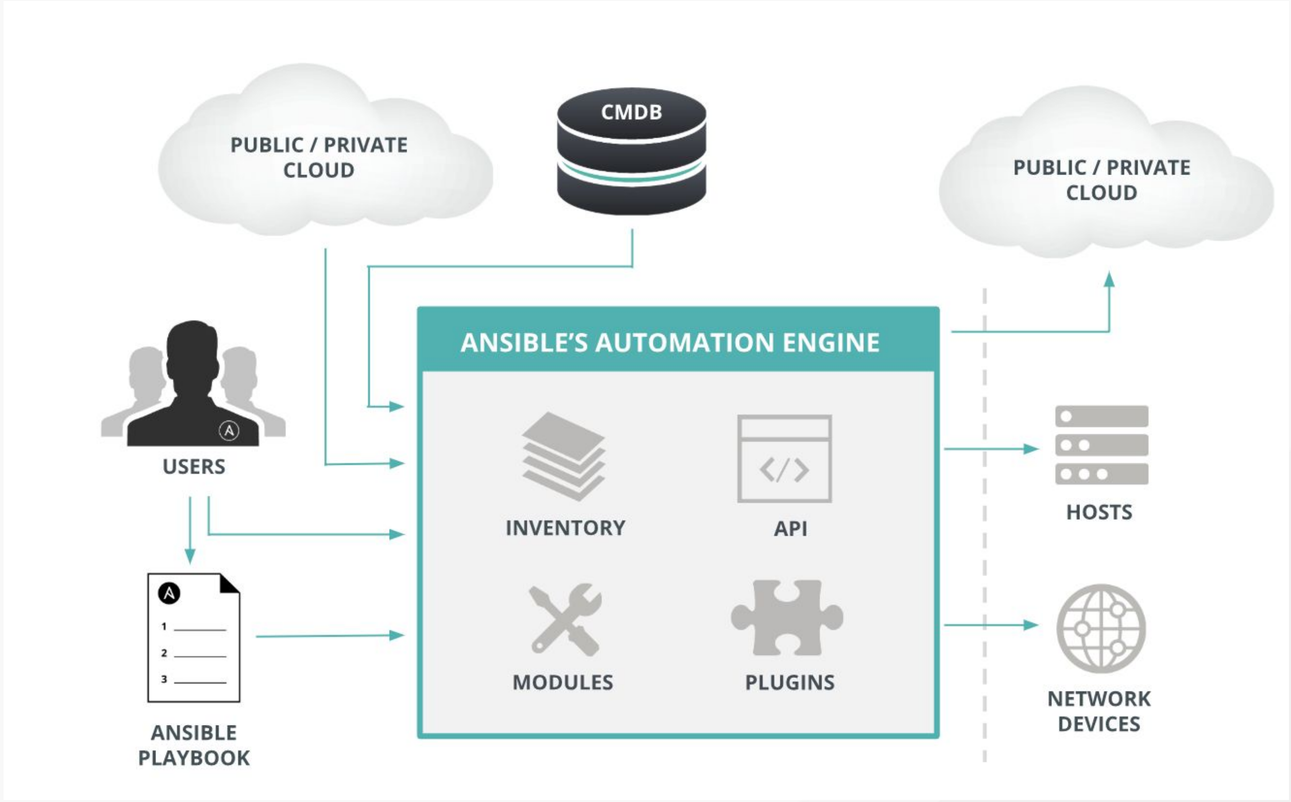
➔ POWERFULL
Automate Everything

➔ AGENTLESS
SSH, WinRM, Python, Powershell

➔ MULTI-CLOUD
Modules for AWS, Azure, GCP, OpenStack...

➔ CROSS PLATFORM
Windows, Linux, Unix, Network...

➔ EVERYTHING AS CODE
Full SCM integration



ANSIBLE WINDOWS AUTOMATION

Use Ansible to deploy and manage Windows systems and applications.

70+
Windows Modules

350+
Powershell DSC resources

ansible.com/windows

redhat.

ANSIBLE NETWORK AUTOMATION

Use Ansible to manage, validate, and continuously track heterogeneous network device configurations and deployments.

Network modules are included as part of the Ansible distribution.

40
Networking platforms

570+
Networking Modules

ansible.com/networking

redhat.

ANSIBLE SHIPS WITH OVER 1900 MODULES

CLOUD	VIRT AND CONTAINER	WINDOWS	NETWORK	NOTIFY
AWS	Docker	ACLs	Arista	HipChat
Azure	VMware	Files	A10	IRC
CenturyLink	RHEV	Commands	Cumulus	Jabber
CloudScale	OpenStack	Packages	Big Switch	Email
Digital Ocean	OpenShift	IIS	Cisco	RocketChat
Docker	Atomic	Regedits	Cumulus	Sendgrid
Google	CloudStack	Shell	Dell	Slack
Linode	**And more...**	Shares	F5	Twilio
OpenStack		Services	Juniper	**And more...**
Rackspace		DSC	Palo Alto	
And more...		Users	OpenSwitch	
		Domains	**And more...**	
		And more...		

PLAYBOOK EXAMPLE

```
---
- name: install and start apache
  hosts: web
  become: yes
  vars:
    http_port: 80

  tasks:
  - name: httpd package is present
    yum:
      name: httpd
      state: latest

  - name: latest index.html file is present
    copy:
      src: files/index.html
      dest: /var/www/html/

  - name: httpd is started
    service:
      name: httpd
       state: started
```

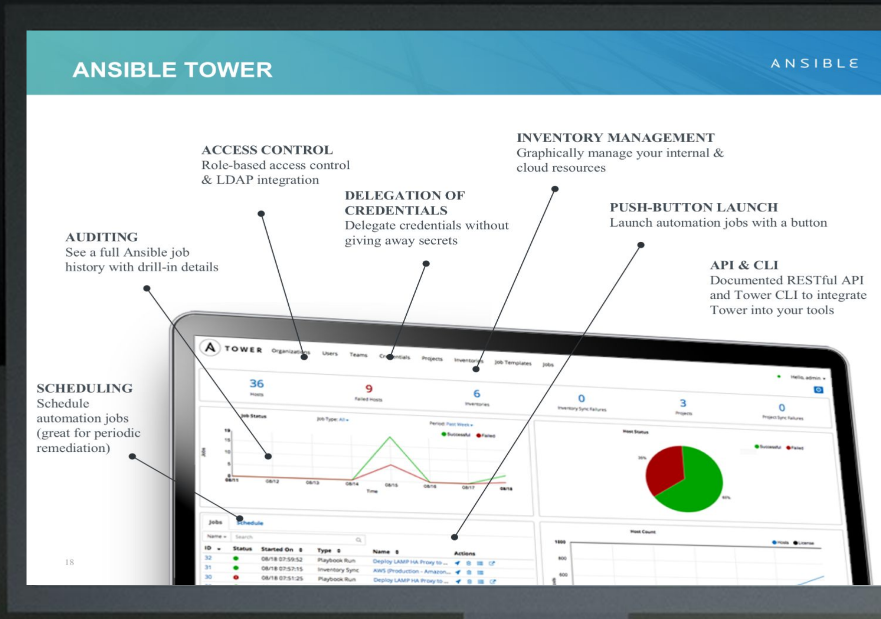
ANSIBLE TOWER

ENTERPRISE SCALE AUTOMATION

ENTERPRISE GRADE ANSIBLE WITH ANSIBLE TOWER

Ansible Tower is an **enterprise framework** for controlling, securing and managing your Ansible automation – with a **UI and RESTful API.**

- **Role-based access control** keeps environments secure, and teams efficient.
- Non-privileged users can **safely deploy** entire applications with **push-button deployment** access.
- All Ansible automations are **centrally logged,** ensuring **complete auditability and compliance.**



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)

INDUSTRIAL SCALE AUTOMATION

Role Based Access Control & LDAP Integration

Define roles over Tenants, Templates, Inventaires, Credentials, Projects …

Easy Scale Out

Tower Instance Groups enable scaling out & fine grain control of the automation workload

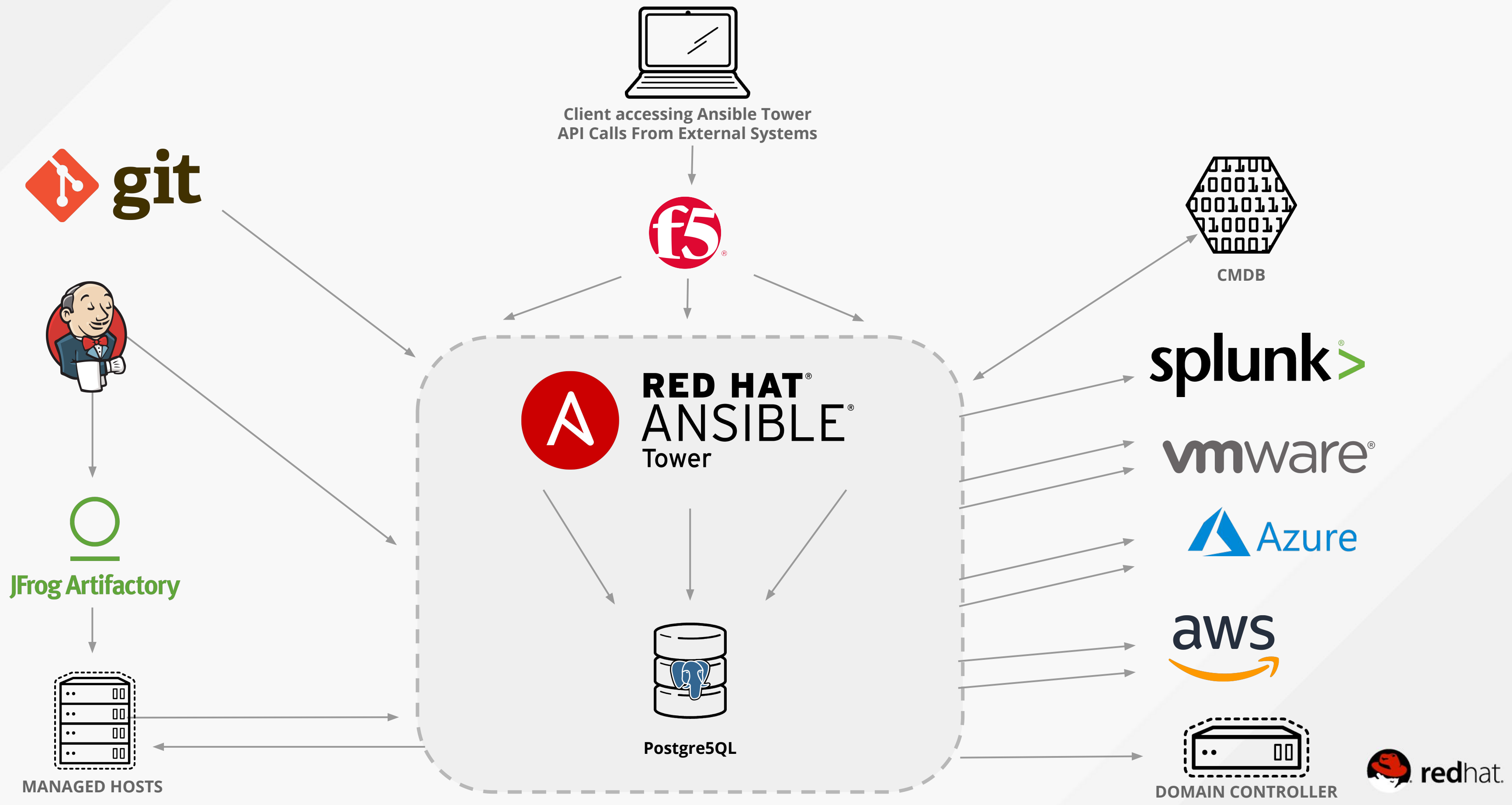
Automate Remote Areas

Tower Isolated Nodes make it easy to run your automation over remote or secured zones

REST API

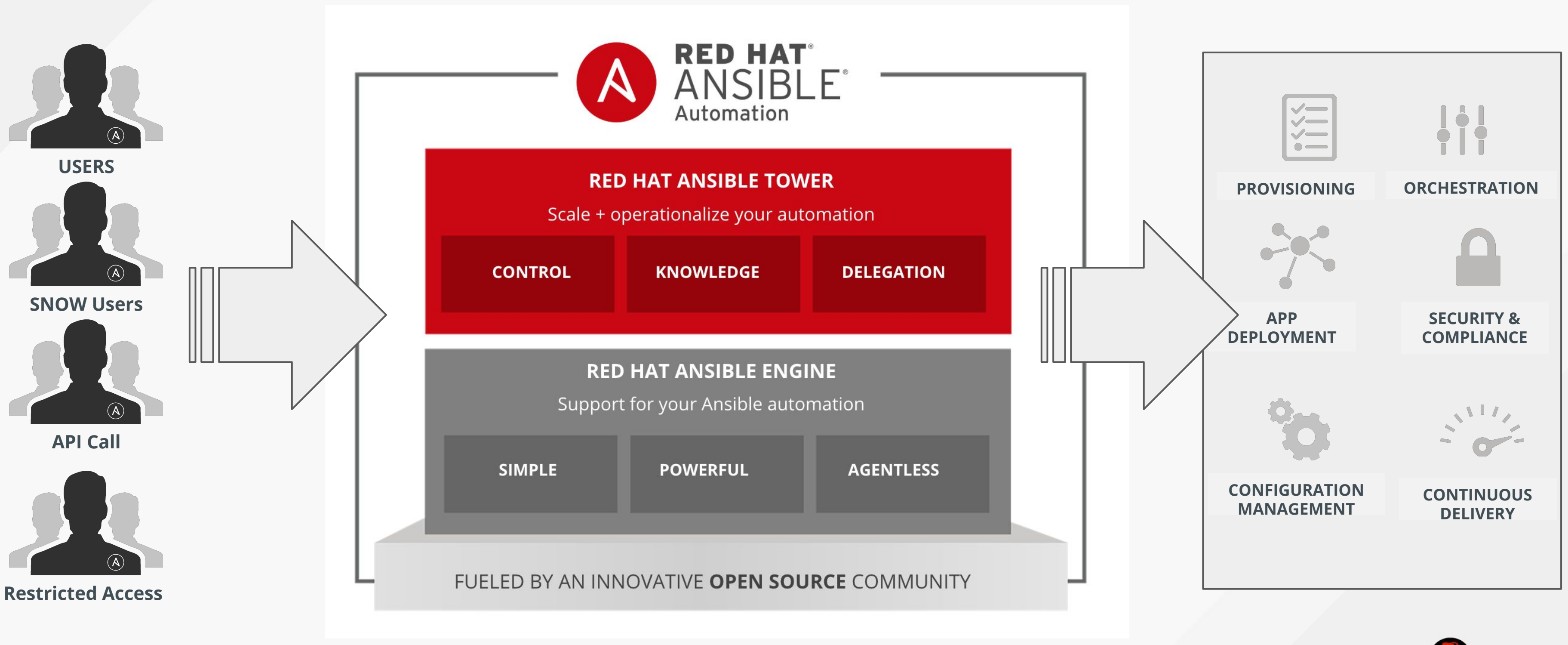
Easily integrate automation in your existing enterprise workflows & processes

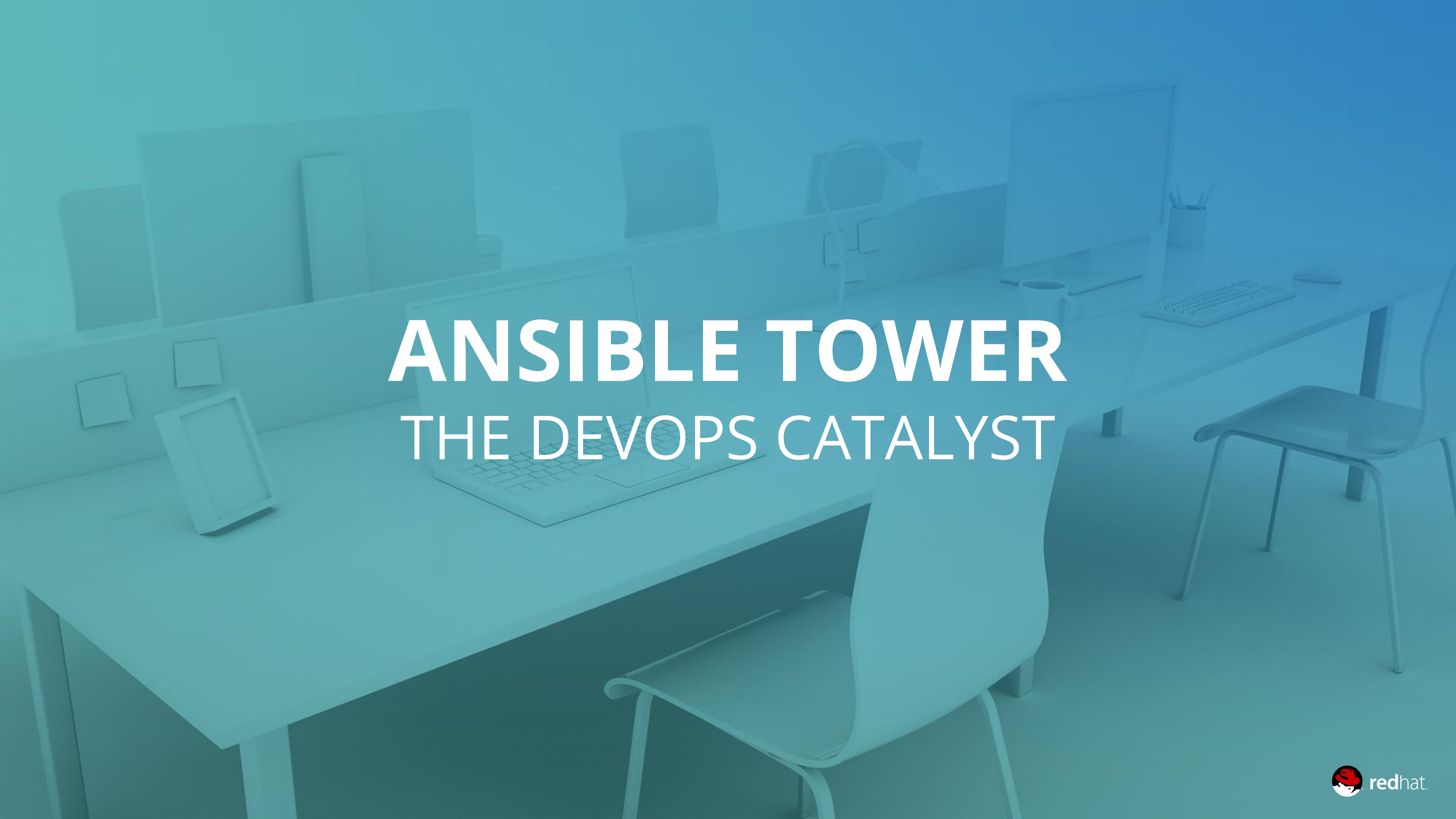
API DRIVEN ECOSYSTEM INTEGRATION



AUTOMATION API

USERS CONSUME A CENTRALIZED AUTOMATION SERVICE

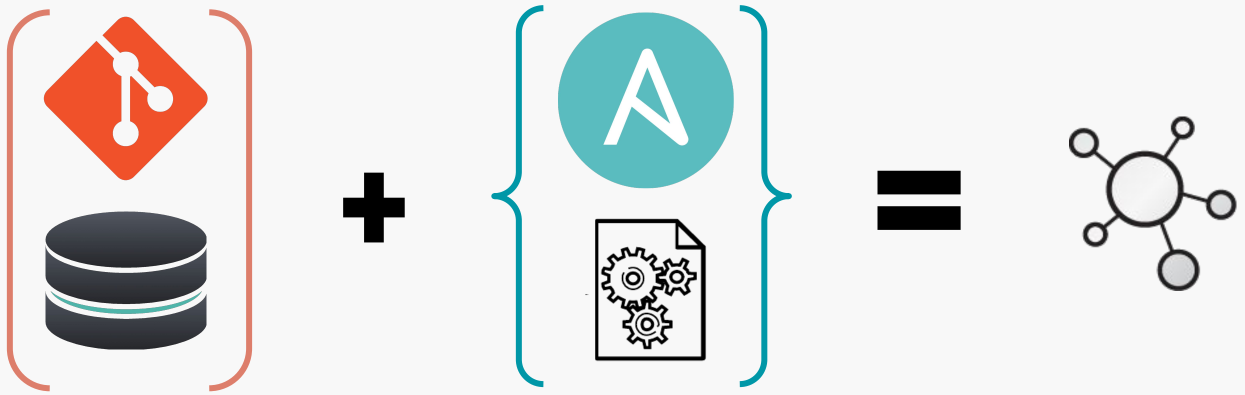




ANSIBLE TOWER

THE DEVOPS CATALYST

INFRASTRUCTURE AS CODE



The IaC approach promotes **formalized, standardized, and automated operational processes**—and dictates that these operational processes are **documented as configuration files or programming code.**

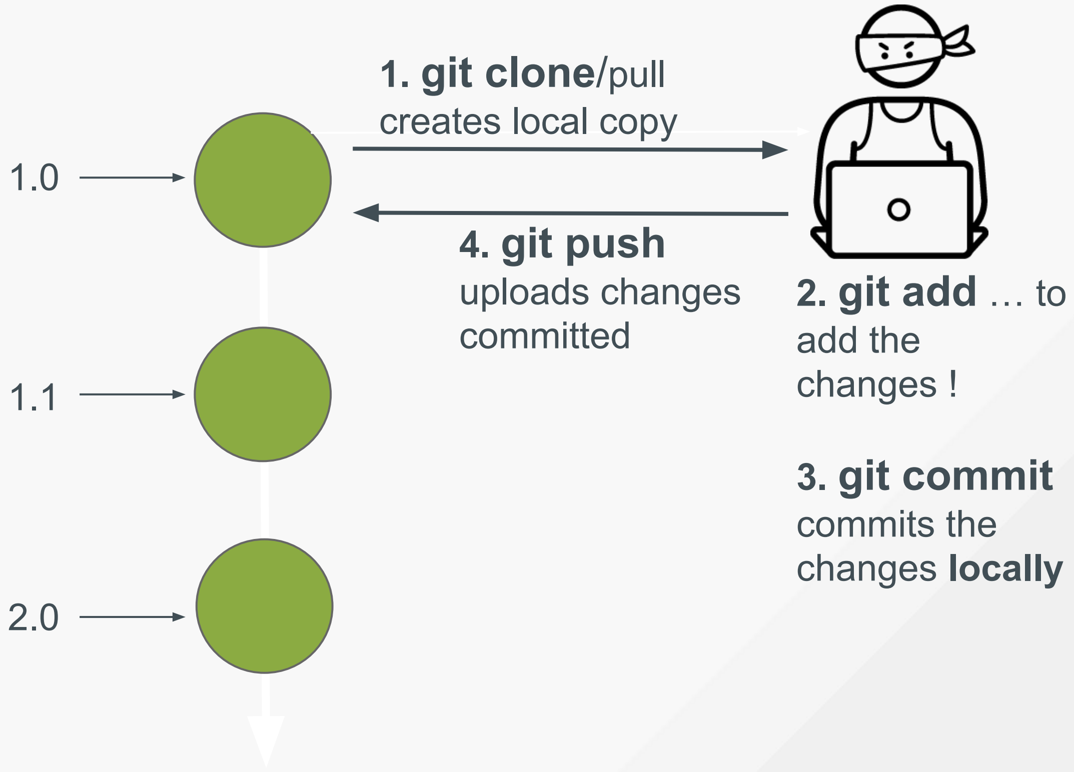
By treating infrastructure as code, IT organizations can automate management tasks while **using the best practices of software development, including code review and version control.**

This approach mitigates management complexity by breaking down a task into smaller, more manageable processes, controlling the execution of code, and effortlessly maintaining up-to-date documentation.

The IaC approach also reduces operational risks by allowing multiple subject matter experts to **peer review the code** and by saving all the previous revisions of a codified infrastructure, enabling previous versions to be restored in case of mistakes. Ultimately, the IaC approach mitigates human errors by enforcing an **automated execution** of the management task performed on the IT infrastructure.

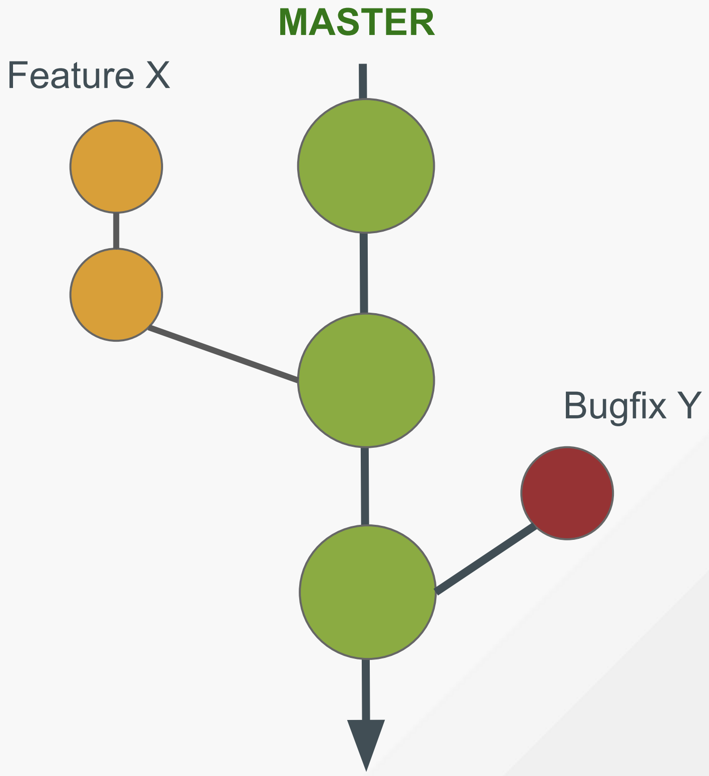
GITHUB WORKFLOW

1. A **git repository** stores files
2. **Access controls** are specific to repositories
3. **All changes** to all files are tracked
4. To change a file you first make a local copy of the repository, then change the file locally, commit the change locally and then tell git to copy this local change to the repository.


1.0
1.1
2.0
1. git clone/pull
creates local copy
4. git push
uploads changes committed
2. git add … to add the changes !
3. git commit
commits the changes **locally**

KEEP MASTER RELEASABLE

1. **Does not require** GitHub, the workflow model is just called that
2. A very simple workflow
3. **Master** branch is always possible to **release**
4. **Branches** are where you develop and test new features and bugfixes.
5. **Yes,** you need to **test**. If you do not test your Ansible code you cannot **keep the master branch releasable !**



redhat.

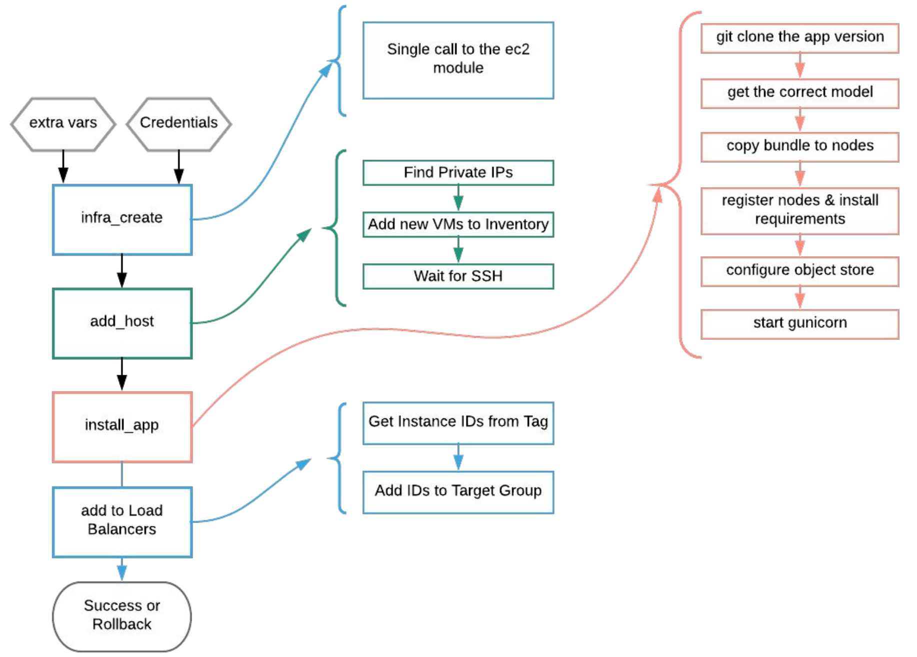
DEMO

1. Multi-Cloud Automation
   - Automate a load balanced web application deployment
   - Ship across multiple infrastructure targets
   - Manage different application versions (A/B Testing)
   - Deploy with a single API call

2. Continuous Delivery
   - Integrate Tower Automation with a Jenkins pipeline
   - Provide a test infrastructure that is always the image of production

Demo Application 1

- RHEL attached to a Load Balancer
- Simple image categorization: is it a car ? Or a cat ?
- Python Gunicorn serving a Flask webapp based on Keras (tensorflow)
- Object backend for images



MORE INFORMATION

https://www.ansible.com/tower
https://www.ansible.com/tower-editions
https://www.ansible.com/tower-trial