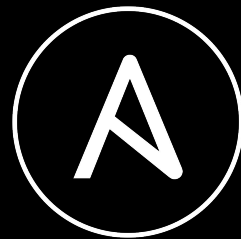


# AUTOMATE ABSOLUTELY EVERYTHING

From traditional to unorthodox, Automate Everything  
(With Ansible)

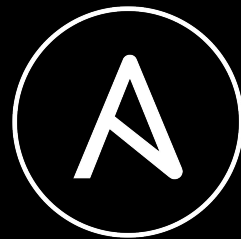
Adam Miller  
Senior Principal Software Engineer  
Ansible Engineering



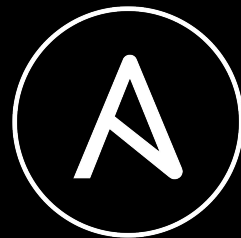


# AN EPITOME OF AUTOMATION

(Powered by Ansible)



# WHAT IS ANSIBLE?

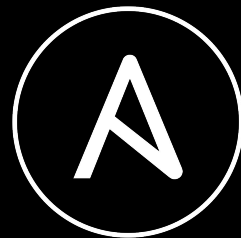


# QUICK INTRODUCTION

WAIT, YOU DON'T KNOW WHAT ANSIBLE IS?

Ansible is an automation tool

- Ansible is a simple agentless idempotent **task automation tool**
  - By default, tasks are executed in-order but we can change that if we want.
- **Tasks** are performed via **modules**
- **Tasks** are grouped together via **plays**
  - Also via **roles**, which are reusable sets of plays we can pass variables to
  - A **play** operates on a set of hosts
- **Playbooks** can contain one or many **plays**
- Plays nicely with "traditional" configuration management systems
  - There's even a puppet module!



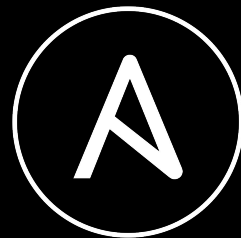
# QUICK INTRODUCTION

WAIT, YOU DON'T KNOW WHAT ANSIBLE IS?

**Module:** yum

**Arguments:** name=bash state=installed

```
$ ansible localhost -m yum -a "name=bash state=installed"
localhost | SUCCESS => {
  "changed": false,
  "msg": "Nothing to do"
}
```



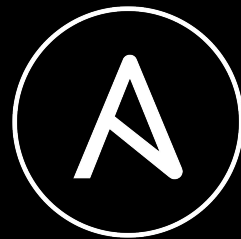
# QUICK INTRODUCTION

WAIT, YOU DON'T KNOW WHAT ANSIBLE IS?

**Module:** yum

**Arguments:** name=bash state=installed

```
playbook.yml
---
- hosts: localhost
  tasks:
    - yum:
        name: bash
        state: installed
```

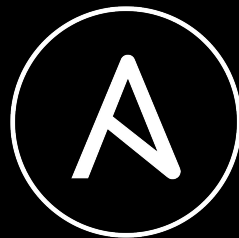


# AUTOMATE EVERYTHING

(With Ansible)

# USING ANSIBLE FOR EVERYTHING

WHY WOULD I WANT TO DO THAT?



Ansible is a simple automation tool that can:

- Execute tasks against one or many hosts, systems, clouds, REST APIs, network devices, hardware devices/appliances, SANs, etc. (basically anything)
- Orchestrate an otherwise complex order of operations, even conditionally based on “facts” or variables (optionally provided at runtime).
- Custom modules can be written in any programming language with JSON support

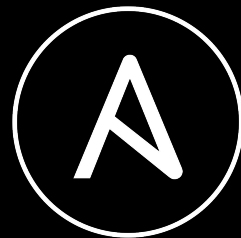
Question of the day:

**What are you trying to accomplish that could be automated?**



# USING ANSIBLE FOR EVERYTHING

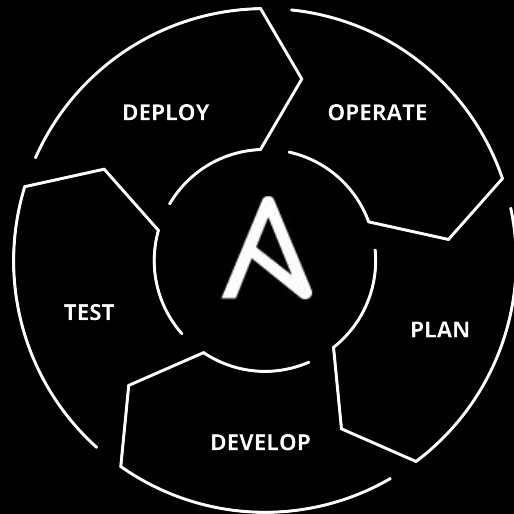
ANSIBLE ALL THE THINGS!!!!

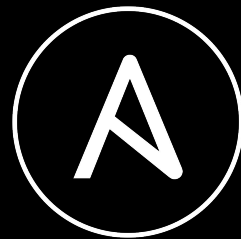


What are you trying to do?

- Configuration Management?
- Provision VMs, Containers, IaaS Instances, K8s Resources?
- Test software?
- Automate workflows?
- Continuous Integration / Continuous Deployment?
- Configure hardware switches, routers, firewalls, and load balancers?
- Replace terrible shell scripts that have survived too long already?
- Other?

**ANSIBLE CAN DO ALL OF THAT! (AND MUCH MORE)**

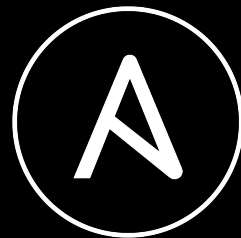




# ANSIBLE DOES THAT

# CONFIGURATION MANAGEMENT TASKS

KEEPING THE TRAIN ON THE TRACKS



## What is configuration management?

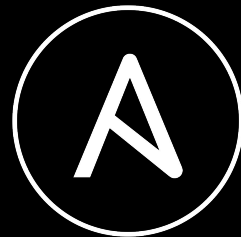
Systems engineering process for establishing and maintaining consistency of a product's performance, functional, and physical attributes with its requirements, design, and operational information throughout its life.

Generally boils down to:

- Managing file content
- Configuration Templating
- System and Service state
- Package Management
- Lifecycle Management

# ANSIBLE DOES THAT

OMG, NO WAY?!?!?!?



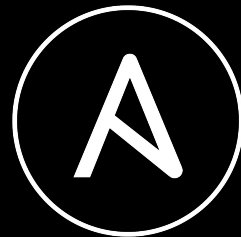
- **Service state:** `service` module
- **Files/Configuration:** `acl` `archive` `assemble` `blockinfile` `copy` `fetch` `file` `find` `ini_file` `iso_extract` `lineinfile` `patch` `read_csv` `replace` `stat` `synchronize` `tempfile` `template` `unarchive` `xattr` `xml`
- **System State:** `aix_devices` `aix_filesystem` `aix_inittab` `aix_lvg` `aix_lvol` `alternatives` `at` `authorized_key` `awall` `beadm` `capabilities` `cron` `cronvar` `crypttab` `dconf` `debconf` `facter` `filesystem` `firewalld` `gather_facts` `gconftool2` `getent` `group` `hostname` `interfaces_file` `iptables` `java_cert` `java_keystore` `kernel_blacklist` `known_hosts` `locale_gen` `lvg` `lvol` `make` `mksysb` `modprobe` `mount` `nosh` `ohai` `open_iscsi` `openwrt_init` `osx_defaults` `pamd` `pam_limits` `parted` `pids` `ping` `puppet` `python_requirements_facts` `reboot` `runit` `seboolean` `sefcontext` `selinux_permissive` `selinux` `selogin` `seport` `service_facts` `service` `setup` `solaris_zone` `svc` `sysctl` `systemd` `sysvinit` `timezone` `ufw` `user` `vdo` `xfconf` `xfs_quota`
- **Package Management:** `apk` `apt` `apt_key` `apt_repo` `apt_repository` `apt_rpm` `bower` `bundler` `composer` `cpanm` `dnf` `dpkg_selections` `easy_install` `flatpak` `flatpak_remote` `gem` `homebrew` `homebrew_cask` `homebrew_tap` `installp` `layman` `macports` `maven_artifact` `npm` `openbsd_pkg` `opkg` `package` `package_facts` `pacman` `pear` `pip` `pip_package_info` `pkg5` `pkg5_publisher` `pkgin` `pkgng` `pkgutil` `portage` `portinstall` `pulp_repo` `redhat_subscription` `rhnc_channel` `rhnc_register` `rhsm_release` `rhsm_repository` `rpm_key` `slackpkg` `snap` `sorcery` `svr4pkg` `swdepot` `swupd` `urpmi` `xbps` `yarn` `yum` `yum_repository` `zypper` `zypper_repository`

More modules being added all the time...



# ANSIBLE DOES THAT - Windows Edition

DON'T WORRY WINDOWS FOLKS, WE DIDN'T FORGET ABOUT YOU <3



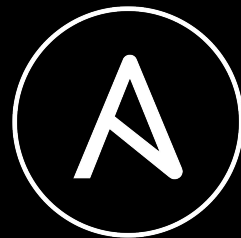
- Windows Modules:

- `async_status` `setup` `slurp` `win_acl_inheritance` `win_acl` `win_audit_policy_system` `win_audit_rule` `win_certificate_store` `win_chocolatey_config` `win_chocolatey_facts` `win_chocolatey_feature` `win_chocolatey` `win_chocolatey_source` `win_command` `win_copy` `win_credential` `win_defrag` `win_disk_facts` `win_disk_image` `win_dns_client` `win_dns_record` `win_domain_computer` `win_domain_controller` `win_domain_group_membership` `win_domain_group` `win_domain_membership` `win_domain` `win_domain_user` `win_dotnet_ngen` `win_dsc` `win_environment` `win_eventlog_entry` `win_eventlog` `win_feature` `win_file` `win_file_version` `win_find` `win_firewall` `win_firewall_rule` `win_format` `win_get_url` `win_group_membership` `win_group` `win_hostname` `win_hosts` `win_hotfix` `win_http_proxy` `win_iis_virtualdirectory` `win_iis_webapplication` `win_iis_webapppool` `win_iis_webbinding` `win_iis_website` `win_inet_proxy` `win_lineinfile` `win_mapped_drive` `win_msg` `win_msi` `win_nssm` `win_optional_feature` `win_owner` `win_package` `win_pagefile` `win_partition` `win_path` `win_pester` `win_ping` `win_power_plan` `win_product_facts` `win_psexec` `win_psmodule` `win_psrepository` `win_rabbitmq_plugin` `win_rds_cap` `win_rds_rap` `win_rds_settings` `win_reboot` `win_regedit` `win_region` `win_regmerge` `win_reg_stat` `win_robocopy` `win_route` `win_say` `win_scheduled_task` `win_scheduled_task_stat` `win_security_policy` `win_service` `win_share` `win_shell` `win_shortcut` `win_snmp` `win_stat` `win_tempfile` `win_template` `win_timezone` `win_toast` `win_unzip` `win_updates` `win_uri` `win_user_profile` `win_user` `win_user_right` `win_wait_for_process` `win_wait_for` `win_wakeonlan` `win_webpicmd` `win_whoami` `win_xml`

More Windows modules being added all the time too!

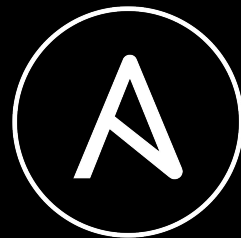
# ADVANCED TASK AUTOMATION TOPICS

THAT LITTLE EXTRA ....



The following categories of Infrastructure Needs are covered extensively by Ansible modules:

- Cloud
- Clustering
- Commands
- Crypto
- Database
- Files
- Identity
- Inventory
- Messaging
- Monitoring
- Network
- Notification
- Packaging
- Remote Management
- SecOps
- Source Control
- Storage
- System
- Utilities
- Web Infrastructure
- Windows

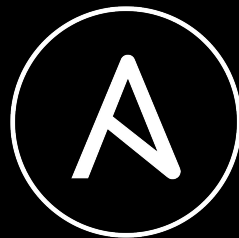


# Over 3300 modules and plugins as of Ansible v2.8.0

(not to mention all that sweet Ansible Galaxy Content)

# PROVISIONING

MAKING SOMETHING FROM NOTHING



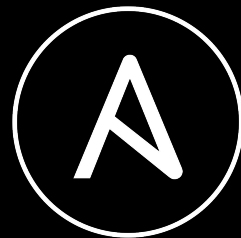
What do you want to accomplish?

- Create IaaS compute instances, object stores, or ephemeral resources?
- Provision virtual machines?
- Create storage allocations?
- Set firewall rules?
- Configure highly available load balancers?
- Create VLANs?
- Deploy container orchestration resources?
- Create databases?
- Other?



# ANSIBLE CAN DO THAT

WHAT? AGAIN? NO WAY!!



Provisioning support for many IaaS providers...

- Amazon Web Services
- Apache CloudStack
- Centurylink Cloud
- Cloudscale
- Digital Ocean
- DimensionData
- Docker Swarm
- Google Cloud
- Linode
- Microsoft Azure
- OpenStack
- OVH
- Packet

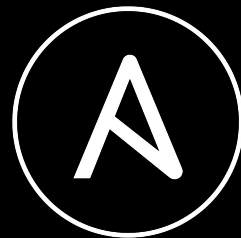
- Profitbricks
- Rackspace Public Cloud
- Softlayer
- WebFaction

... and Datacenter Virtualization

- Atomic Host
- libvirt resource management
- Joyent SmartOS Virt
- oVirt
- Red Hat Virtualization
- VMWare (VSphere/ESXi)
- Univention

# PROVISIONING - CONTINUED

OMG, THIS LIST JUST KEEPS GOING...



## Networking

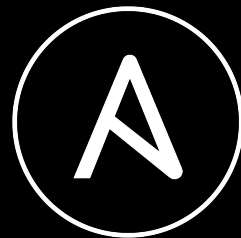
- A10 Networks
- Apstra AOS
- Arista EOS and Cloudvision
- Aruba
- Avi Networks
- BigSwitch
- Cisco (ASA, ACI, IOS/IOS-XR, NX-OS, WLC)
- Cumulus Networks (Cumulus Linux)
- Dell EMC (OS6, OS9, and OS10)
- F5 BigIP
- Fortios Firewall
- Huawei Cloudengine
- JunOS
- Lenovo CNOS
- Netscaler
- Netvisor
- Open vSwitch
- Ordnance
- Palo Alto Networks PAN-OS
- Nokia SR OS
- VyOS

## Databases

- InfluxDB
- Redis
- Riak
- MS-SQL
- MySQL
- Postgresql
- Vertica

# PROVISIONING - CONTINUED

SERIOUSLY? MORE STUFF? ... THE LIST GOES ON



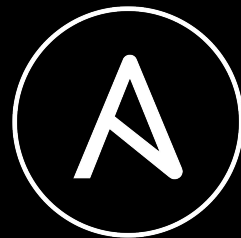
## Infrastructure, Web, Clustering and Cloud

- Apache HTTPD (module and mod\_proxy management)
- Atomic Host
- Consul
- Django Management
- eJabberd
- httpasswd
- HP iLO
- JBoss
- Jenkins (Jobs, Plugin, and Jenkinsfile management)
- Jira
- Kubernetes
- Letsencrypt
- ManageIQ

- OpenShift
- Pacemaker
- Supervisord
- ZooKeeper

## Storage

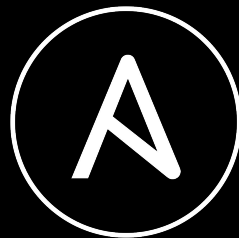
- AIX LVM
- Gluster Volume
- Infinidat
- LVM2
- NetApp
- Purestorage
- ZFS



# DOING THINGS WITH ANSIBLE

# COMMAND LINE TOOLING

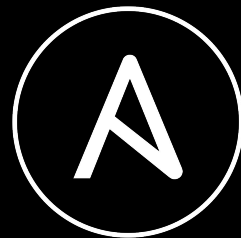
BUT WHAT ABOUT MY PERL ONE-LINERS?



Make Ansible your new command line tooling API, stop re-inventing the wheel

- Ansible provides a very capable Python API for modules
- Modules can be written in any programming language that understands JSON
- Provides a consistent “UX” for all tasks
- Gives you and your ops team an “on ramp” to scaling your tasks across the infrastructure

```
$ ansible localhost -m my_task -a "arg1=foo arg2=bar"
```



# DEPLOYMENT

I JUST GIT PUSH TO THE CLOUD, RIGHT?

Software Deployment is the act of making software available on systems; most often, this is a sequence of steps that must be performed in-order. (In-order task execution anyone?)

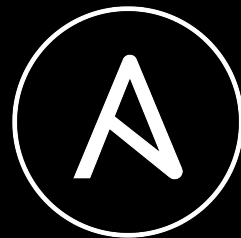
Example:

- Sync some data
- Database schema migration
- Remove systems from load balancer
- Push new code
- Put systems back in load balancer
  - Rinse/Repeat on previously not upgraded set
- Verify services are functional
- Status update

**Remember what a Playbook does?**

# APPLICATION LIFECYCLE MANAGEMENT

DO IT LIVE!

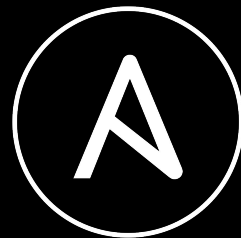


Managing application lifecycle across one or many hosts

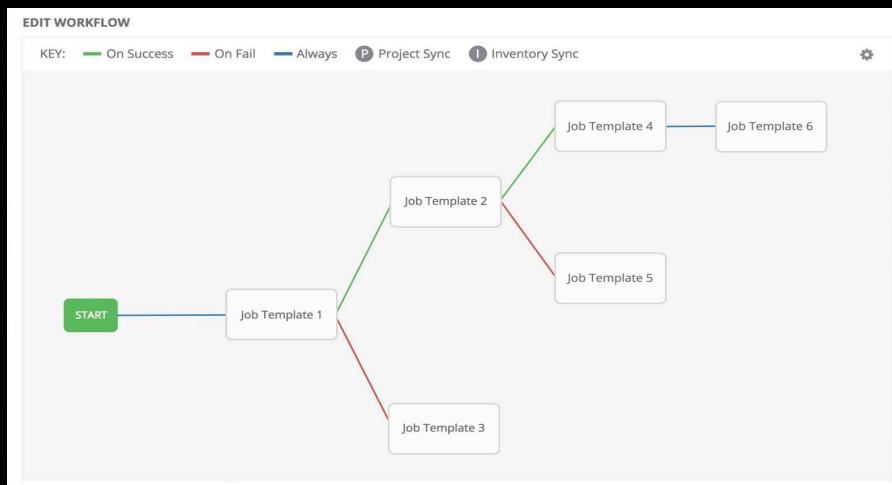
- Ansible can orchestrate both simple and complex lifecycle management
- Lifecycle “order of operations” defined in Playbooks
  - Whatever your requirements are
- Plays can execute on different sets of hosts
  - Multiple plays per playbook
- Plays can use varying execution strategies for various requirements
  - Cluster node management
  - Database schema updates
  - etc
- Sky is the limit
  - (something something ... cloud)

# ORCHESTRATION AND WORKFLOW

AUTOMATION WITH FEELING



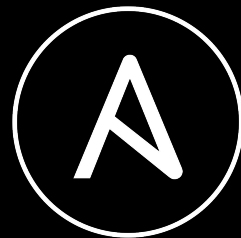
Flow controlled automation by data from the environment allowing the automation tasks to make “intelligent” decisions.





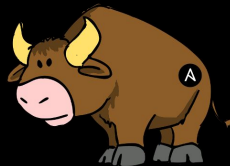
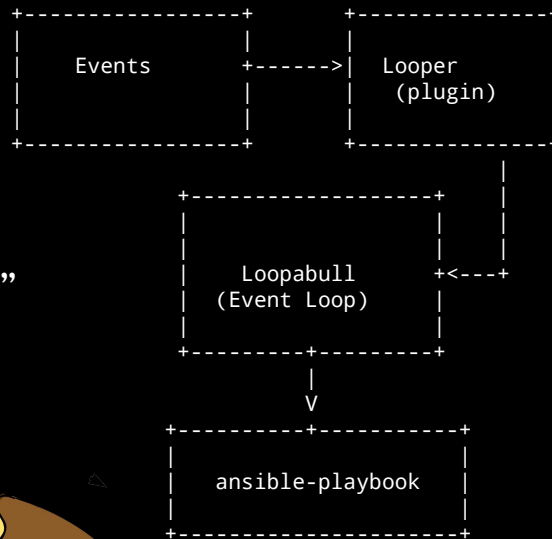
# EVENT BASED EXECUTION

COWSAY WHAT?



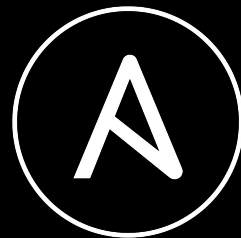
Ansible can easily integrate with existing infrastructure to perform actions based on events.

- Example: loopabull
  - Events in the infrastructure spawn messages on the bus
  - loopabull listens on the bus, waiting for a “routing key” that it cares about (message topic)
  - Message payload is injected into Ansible playbooks as variables, allowing for decisions to be made based on message contents



# CONTINUOUS INTEGRATION

THERE IS ONLY ZUUL



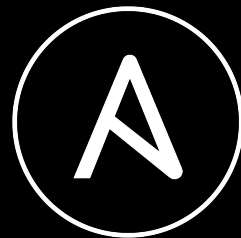
Zuul CI - <https://zuul-ci.org/>

- Project Gating
- Workflow pipelines
- Cross-Project and Cross-Repo dependency gating
- Multi-node deployment job testing
- Tests are written as Ansible Playbooks
- Run your tests with the same Playbooks you use in Production
- OpenStack CI System (Zuul) - <http://status.openstack.org/zuul/>
  - 2,000+ jobs-per-hour
    - single-use OpenStack VMs -> create and destroy 2K+ VMs per hour
  - >1800 disjoint git repositories to perform gating on
  - Spread across 7 public OpenStack clouds and 4 private OpenStack clouds
    - Hybrid cloud anyone?



# ANSIBLE CONTAINER (AND BENDER)

END THE DOCKERFILE MADNESS

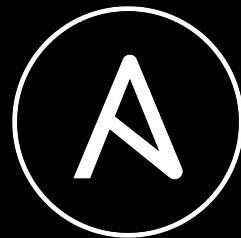


Using Ansible playbooks to build you OCI compliant container images

- Stop chaining together shell commands in Dockerfiles
- Create containers the same way you deploy to servers
- **roles == services**, build your containers using **roles**
  - Making single-purpose (microservice) containers easy
- Get more out of your time investment writing roles and playbooks
- Create multi-container builds easily (Ansible Container)
  - (Think Docker Compose, but like ... better)
- Deploy to Container Orchestration Platforms (Ansible Container)
  - Currently Supports OpenShift and Kubernetes

# ANSIBLE + K8S/OPENSHIFT

CONTAINING THAT CONTAINER MADNESS



## Automating OpenShift and Kubernetes

- `openshift` and `k8s` modules
  - Automate command/control of OpenShift or Kubernetes native resources
- Automation Broker
  - Orchestrate/Manage Service Catalog Apps
  - <http://automationbroker.io/>
  - (AKA - Ansible Service Broker)



AUTOMATION BROKER

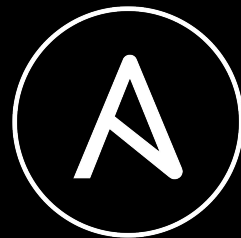
- Ansible Operators
  - Create Kubernetes Operators with Ansible
  - Operator Framework runs an Ansible playbook or role every time a certain type of object is detected / modified



Operator  
Framework

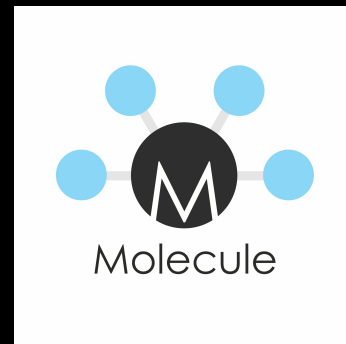
# TESTING YOUR PLAYBOOKS AND MODULES

TEST DRIVEN WHAT???



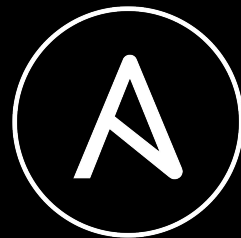
Molecule (<https://molecule.readthedocs.io>)

- Designed to aid in the development and testing of Ansible roles.
- Provides support for testing with multiple:
  - Instances
  - operating systems / Linux distributions
  - virtualization providers
  - test frameworks
  - testing scenarios
- Molecule uses Ansible playbooks to exercise the role and its associated tests.
- Molecule supports any provider that Ansible supports.



# ANSIBLE GALAXY

YOUR MOTHER WAS RIGHT, IT'S BETTER TO SHARE



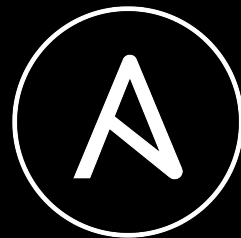
## Ansible Galaxy

- Galaxy is your hub for finding, reusing and sharing the best Ansible content
- Share Roles you've created
- Collaborate upstream on improving re-usable functionality
- <https://galaxy.ansible.com>



# USING ANSIBLE TO HARDEN SYSTEMS

ONE DOES NOT SIMPLY WALK INTO MORDOR



## Security Audit and Compliance

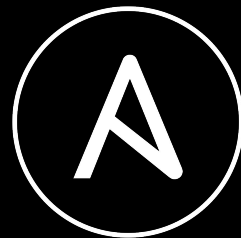
- DISA Security Technical Implementation Guides (STIGs)
- Ansible Lockdown - (in partnership with MindPointGroup)
  - Available in Ansible Galaxy
  - DISA and CIS Guidelines
- OpenSCAP
  - SCAP Implementation
  - Ansible Integrations for remediation



OpenSCAP

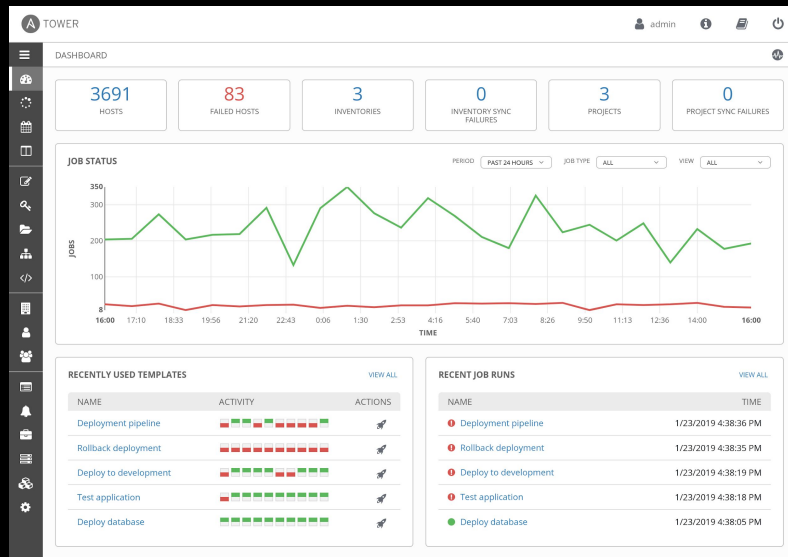
# ANSIBLE TOWER

PRETTY GRAPHS! (But no seriously, this is awesome)



## The definitive Ansible Centralized Management Portal

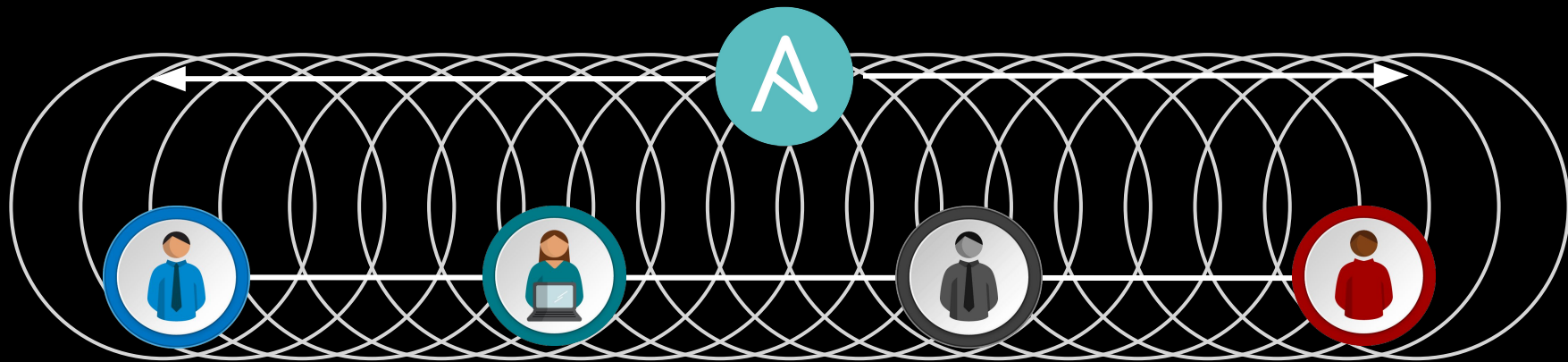
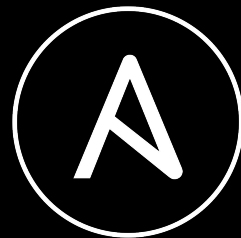
- Role Based Access Control
- Centralized Logging, History Visualizations
- Multi-Playbook Workflow Orchestration
- Playbook and System Auditing (System Tracking)
- Self-Service Automation
  - Sanitized form-based playbook runs
- Integrated Notifications (ChatOps, etc)
- Clustered Auto-Sharding Job Execution
- Self Documenting REST API
- Tower CLI and Ansible Engine Tower modules
- ... and much much more!





# ANSIBLE AS AN UNIVERSAL LANGUAGE

ANSIBLE FOR EVERYTHING

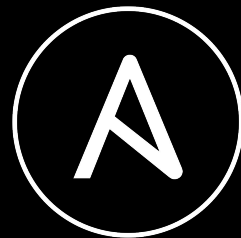


**BUSINESS**

**DEV**

**NETWORK**

**IT OPERATIONS**



# THANK YOU

**ADAM MILLER**



maxamillion



maxamillion



@TheMaxamillion

