redhat.

# Ansible and CloudForms

## What the future may be

Mike Dahlgren
miked@redhat.com

# Overview

What is Ansible and why is it different?

What is Ansible Tower?

How will it work with CloudForms? *

* Wild speculation

| | AIX | *BSD | HP-UX | Linux | OS X | Solaris | Windows | Others |
|---|---|---|---|---|---|---|---|---|
| Ansible | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes[92] |
| Bcfg2 | Partial[93] | Yes[94] | No | Yes[95] | Partial[96] | Yes | No | No |
| CFEngine | Yes | Yes[94] | Yes | Yes | No | Yes | Partial | Yes[97] |
| cdist | | Yes | | Yes | Yes | | No | |
| Chef | Yes[98] | Yes | Yes | Yes | Yes | Yes | Yes[99] | Yes |
| ISconf | Yes | Yes | Yes | Yes | Yes | Yes | No | No |
| Juju | | | | Yes | | | | |
| Local ConFiGuration system (LCFG) | No | No | No | Partial[100] | Partial[101] | Partial[102] | No | No |
| OCS Inventory NG | Yes | Yes | Yes | Yes | Yes | Yes | Yes | No |
| Open pc server integration (Opsi) | No | No | No | Yes | No | No | Yes | No |
| PIKT | Yes | Yes | Yes | Yes | Yes | Yes | No | Yes[103] |
| Puppet | Yes | Yes | Yes | Yes | Partial | Yes | Yes | Yes |
| Quattor | No | No | No | Yes | Partial[104] | Yes | No | No |
| Radmind | Yes | Yes[94][105][106] | No | Yes | Yes | Yes | Yes | No |
| Rex | | Yes | | Yes | Yes[107] | Yes | Yes[107] | No |
| Rudder | Yes | Partial[108] | No | Yes | Partial[108] | Partial[109] | Yes | Yes[110] |
| Rundeck | Yes | Yes | Yes | Yes | Yes | Yes | Yes | No |
| SmartFrog | No[111] | No[111] | Yes | Yes | Yes | Yes | Yes | No[111] |
| Salt | Yes | Yes | Partial[112] | Yes[113] | Yes | Yes[114] | Yes | Partial[112] |
| Spacewalk | No[115] | No | No | Yes[116] | No | Yes[117] | No | No |
| STAF | Yes[118] | Yes[119] | Yes[120] | Yes[121] | Yes[122] | Yes[123] | Yes[124] | Yes[125] |
| Synctool | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes[126] |
| Vagrant | | | | Yes | Yes | | Yes | |

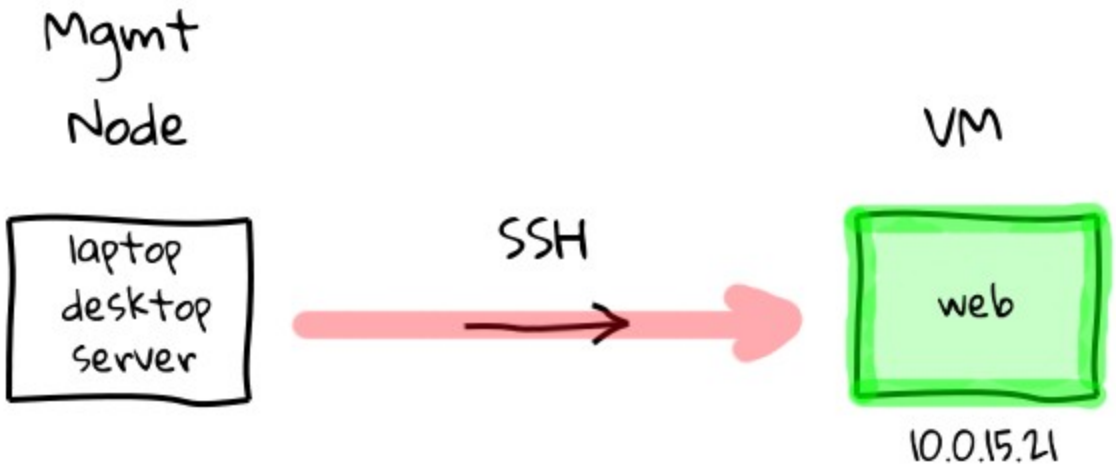
# Ansible Design Goals

**Minimal in nature -** Requires no additional dependencies.

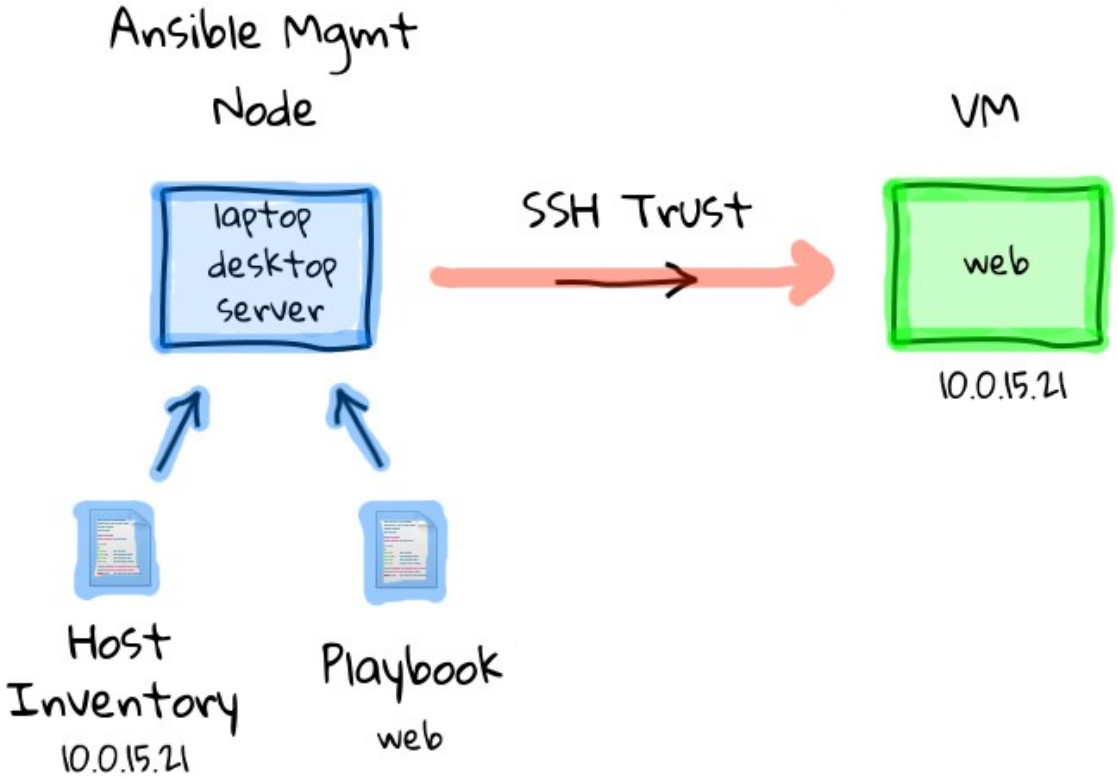
**Secure -** Only OpenSSH is required, which is already tested.

**Highly reliable -** Prevention of side-effects from re-running.

**Low learning curve-** Playbooks use language based on YAML



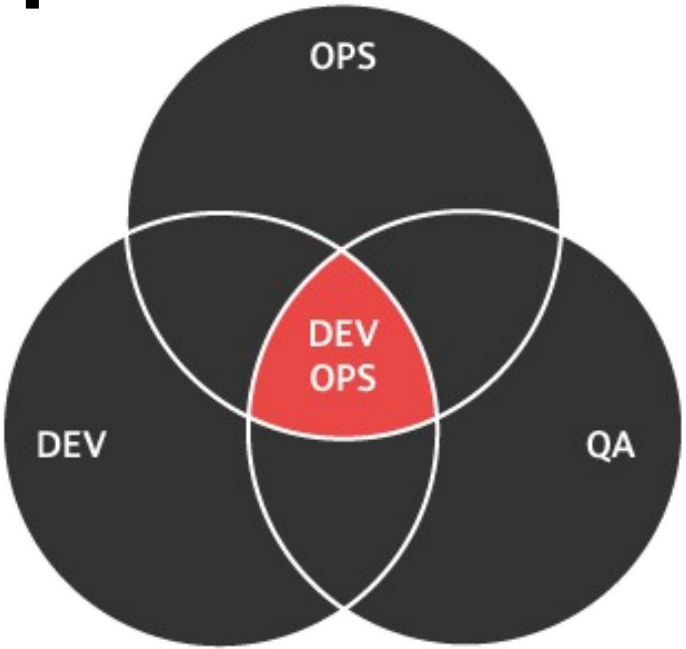

sysadmincasts.com



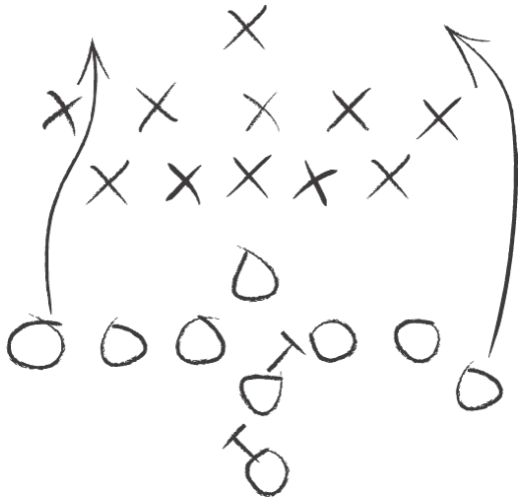

sysadmincasts.com

# Host Inventory file example

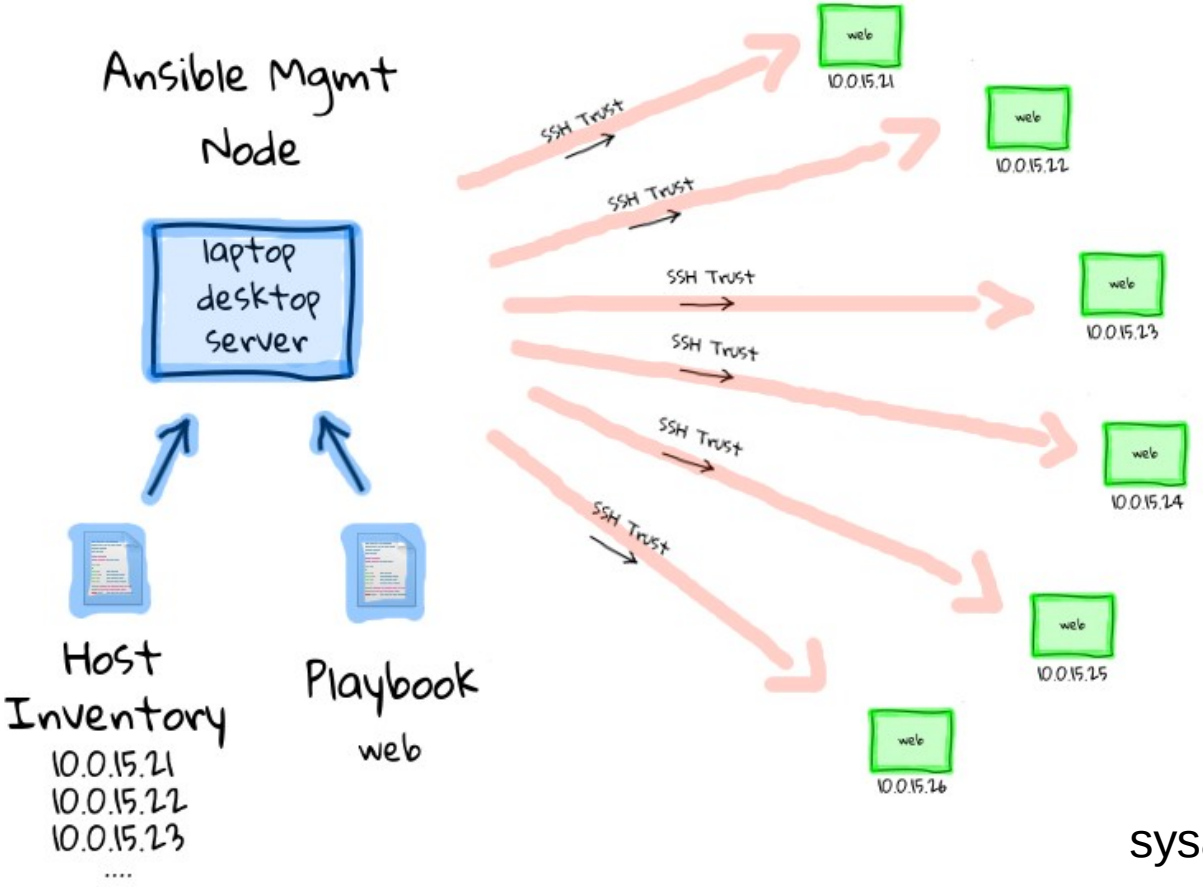
```
[webservers]
foo.example.com
bar.example.com

[dbservers]
one.example.com
two.example.com
```



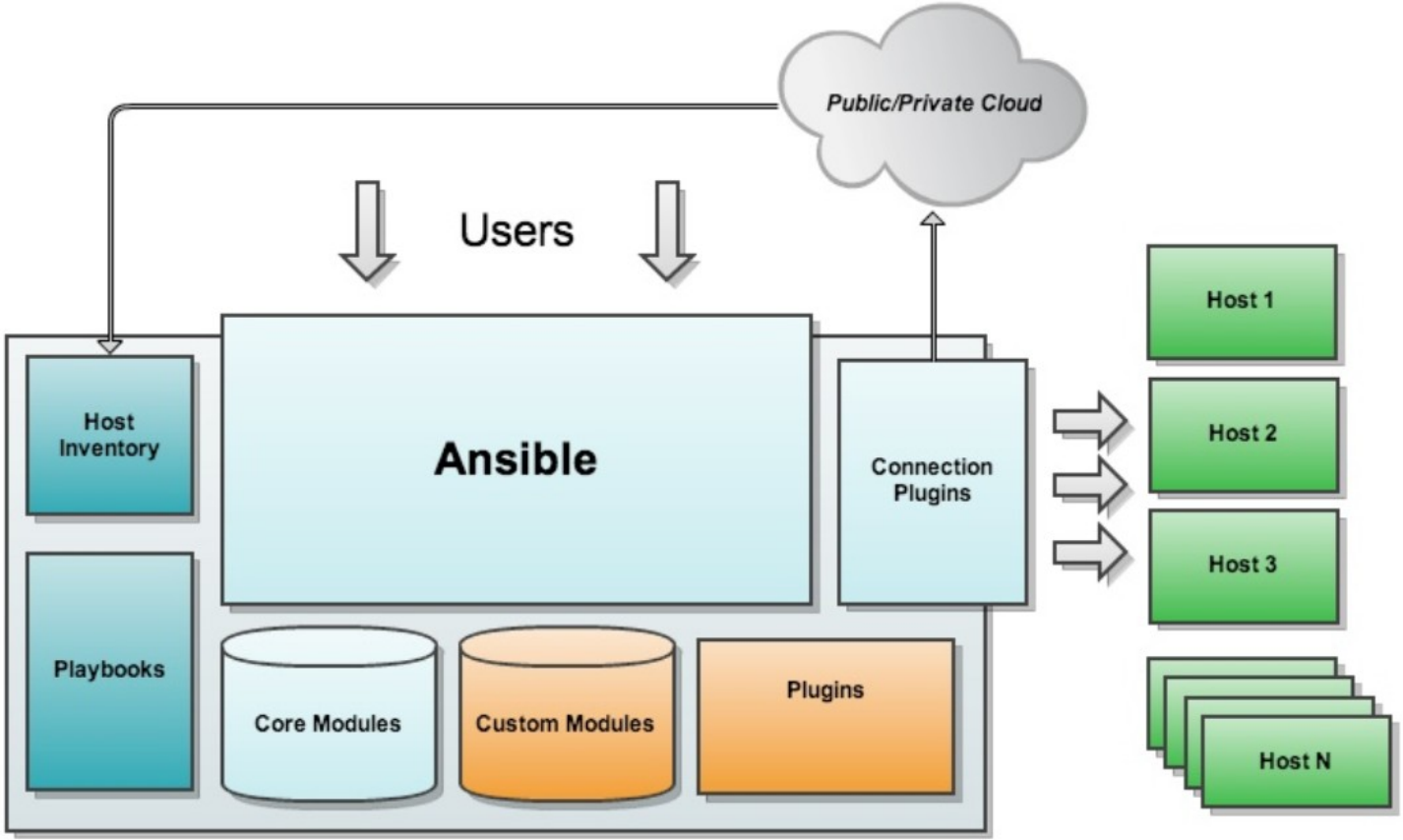
# The Ansible Playbook



```
---
- hosts: webservers
  vars:
    http_port: 80
  tasks:
  - name: ensure apache is at the latest version
    yum: name=httpd state=latest
  - name: write the apache config file
    template: src=/srv/httpd.j2 dest=/etc/httpd.conf
    notify:
    - restart apache
  - name: ensure apache is running (and enabled)
    service: name=httpd state=started enabled=yes
  handlers:
    - name: restart apache
      service: name=httpd state=restarted
```




sysadmincasts.com



# Simple shell commands

# File Transfers

Copy files from manager to servers (utilizes SCP)

**$ ansible webservers -m copy -a "src=/etc/hosts dest=/tmp/hosts"**

Changing ownership and permissions

**$ ansible webservers -m file -a "dest=/srv/foo/b.txt mode=755 owner=miked group=miked"**

Removing files from servers

**$ ansible webservers -m file -a "dest=/path/to/c state=absent"**

# Managing Packages

Install package acme

**$ ansible webservers -m yum -a "name=acme state=present"**

Update package to latest version

**$ ansible webservers -m yum -a "name=acme state=latest"**

Removing package

**$ ansible webservers -m yum -a "name=acme state=absent"**

# Managing services

Start Apache

**$ ansible webservers -m service -a "name=httpd state=started"**

Restart Apache

**$ ansible webservers -m service -a "name=httpd state=restarted"**

Stop Apache

**$ ansible webservers -m service -a "name=httpd state=stopped"**

# Parallelism and shell commands

Reboots all web servers (10 in parallel)

**$ ansible webservers -a "/sbin/reboot" -f 10**

Running commands as a different user

**$ ansible webservers -a "/usr/bin/foo" -u username**

Have sudo as for a password

**$ ansible atlanta -a "/usr/bin/foo" -u username --sudo [--ask-sudo-pass]**

# Time limited background Operations

Run script in background (30 Min timeout)
**$ ansible webserver -B 3600 -P 0 -a "/usr/bin/long_running_operation --do-stuff"**

Checking on the status of a previous job
**$ ansible web1.example.com -m async_status -a "jid=488359678239.2844"**

We can set how often to poll the status (60 seconds)
**$ ansible webserver -B 1800 -P 60 -a "/usr/bin/long_running_operation --do-stuff"**

# Ansible Tower Overview

https://fast.wistia.net/embed/iframe/9iw6g0o81c?popover=true

# ANSIBLE INTEGRATION

## ACCELERATED PROVISIONING


**Ansible Tower Integration**

Ansible Jobs as a Service Type.

Ansible Jobs as a Method Type.

Ansible Jobs as Control Actions.

Inventory, Reporting and State Drift of Ansible Jobs.

Facts inventory merged in VM property pages.

redhat.

# Questions?