



Wall to wall Ansible

Maxim Burgerhout
Principal Solution Architect
Red Hat



Twitter: [@MaximBurgerhout](https://twitter.com/MaximBurgerhout)

YouTube: [100 Things to do with Red Hat Management Products](https://www.youtube.com/watch?v=100ThingsRedHat)
bit.ly/100thingsredhat

Maxim Burgerhout

Principal Solutions Architect

maxim@redhat.com

Red Hat

What I want to do today



Background

What is Ansible, where is it going

Ansible use cases

What lies beyond configuration management

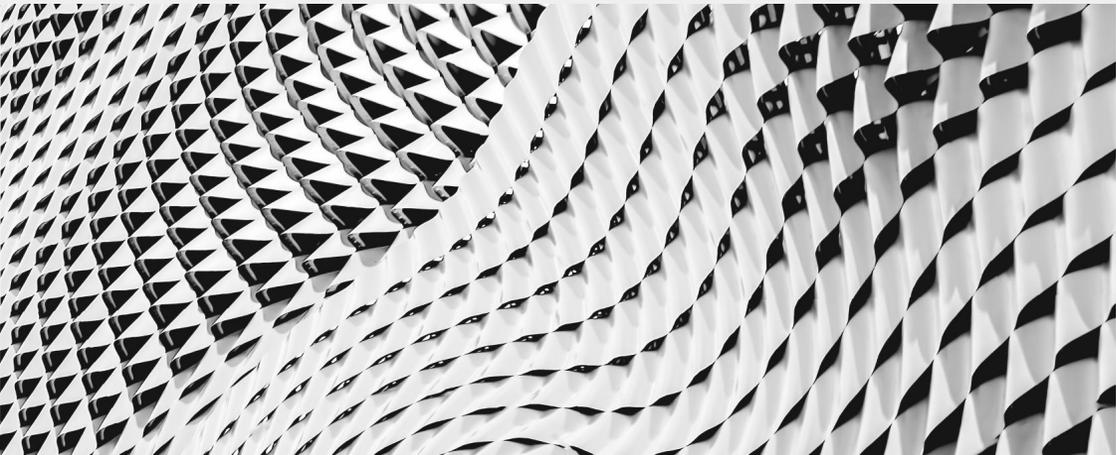
Going wall to wall

How do I maintain this huge playbook with my team(s)?

Demo

Let's see it!

What's Ansible?

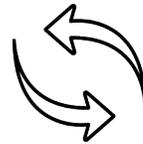


Ansible



Simple

- Human readable automation
- No special coding skills needed
- Tasks executed in order
- Usable by every team
- Get productive quickly**



Powerful

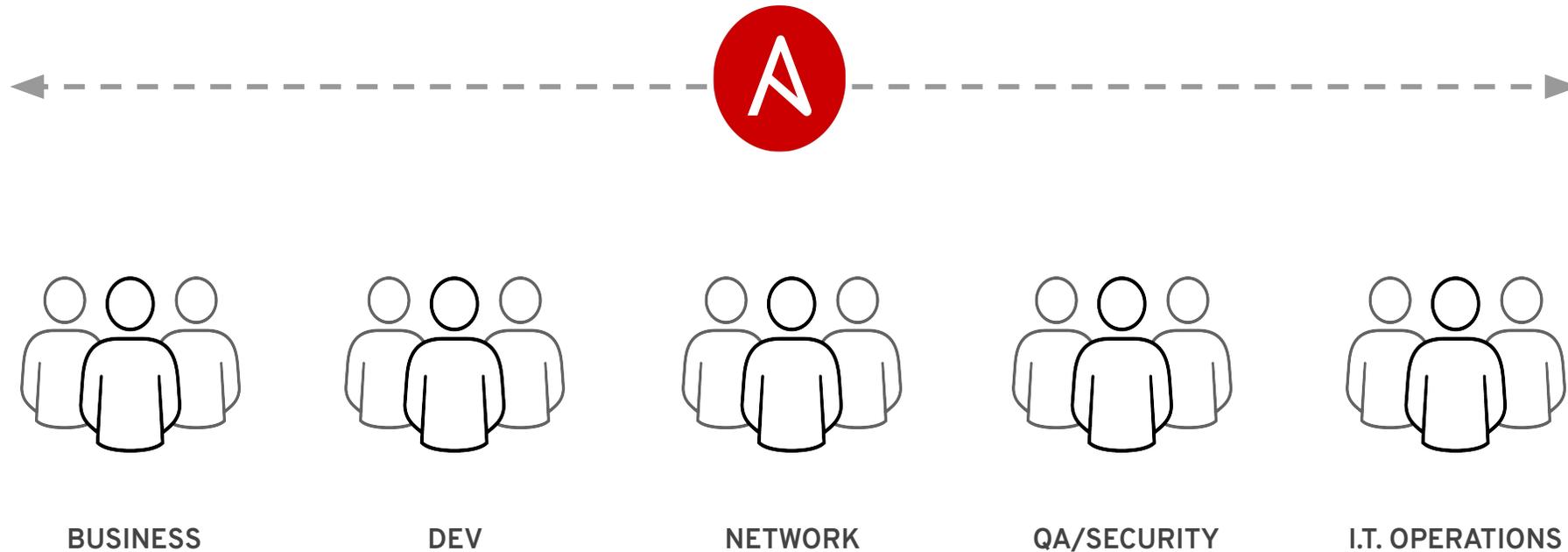
- App deployment
- Configuration management
- Workflow orchestration
- Network automation
- Orchestrate the app lifecycle**



Agentless

- Agentless architecture
- Uses OpenSSH & WinRM
- No agents to exploit or update
- Get started immediately
- More efficient & more secure**

Ansible Automation works across teams



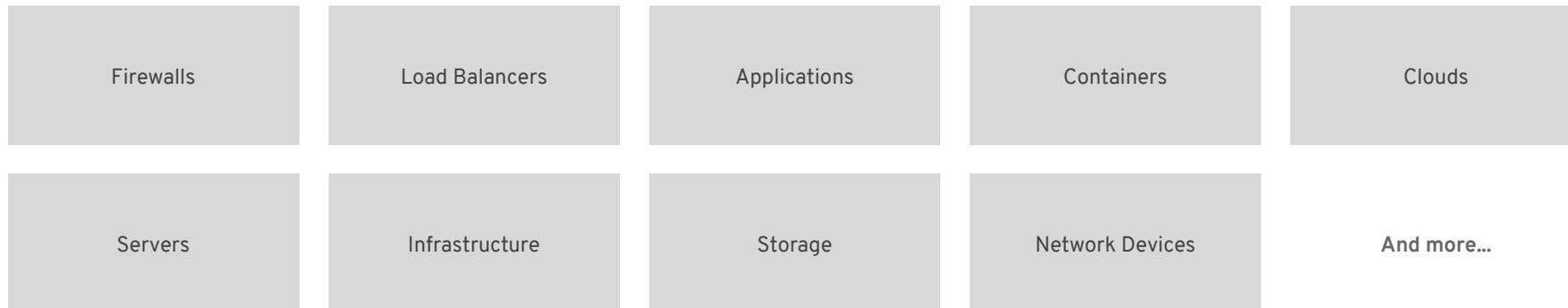
What can I do using Ansible?

Automate the deployment and management of your entire IT footprint.

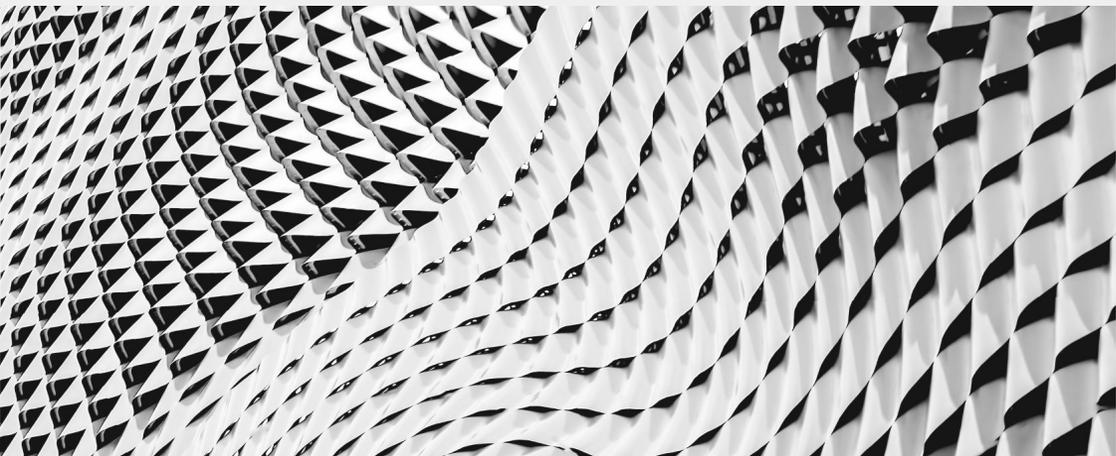
Do this...

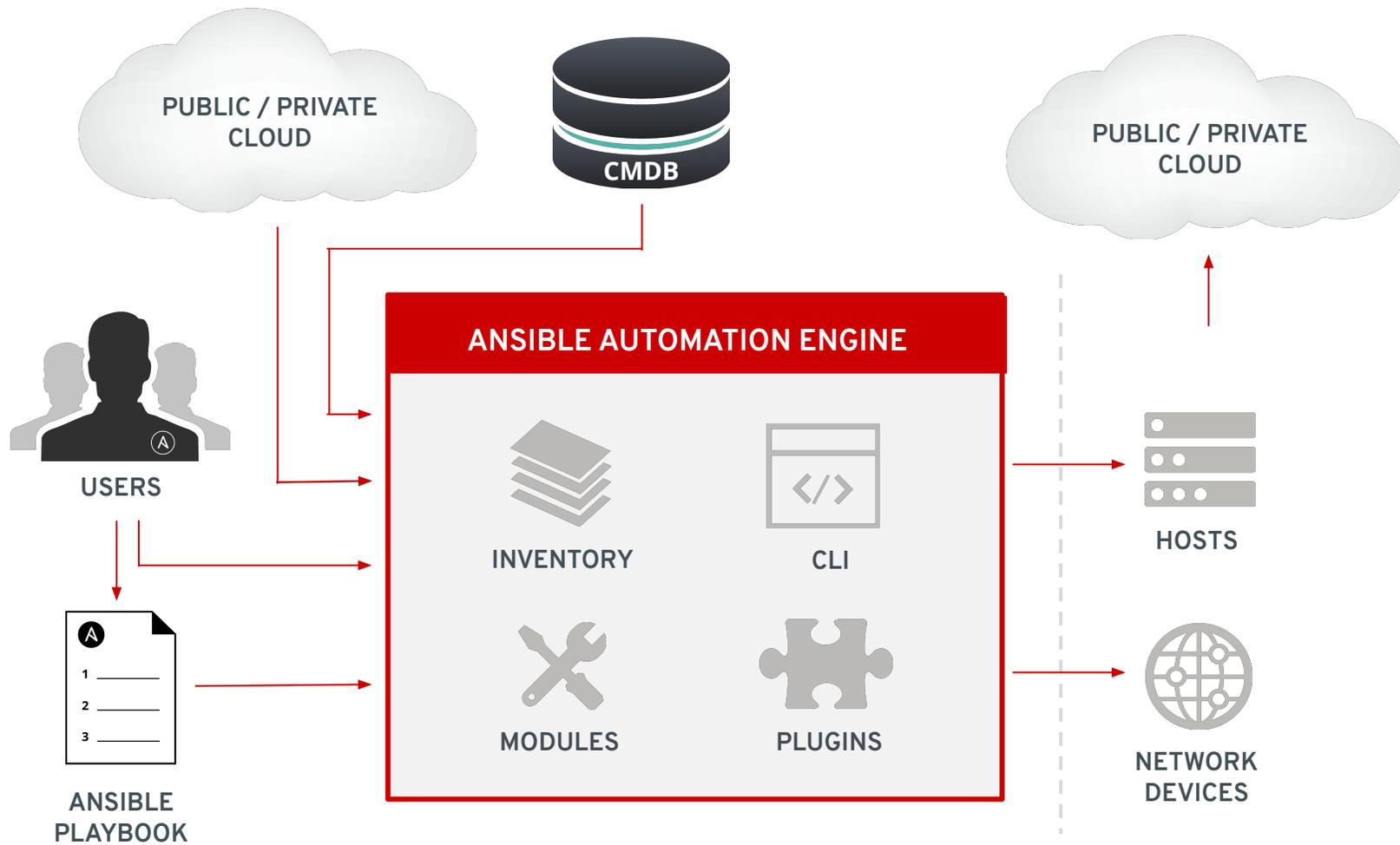


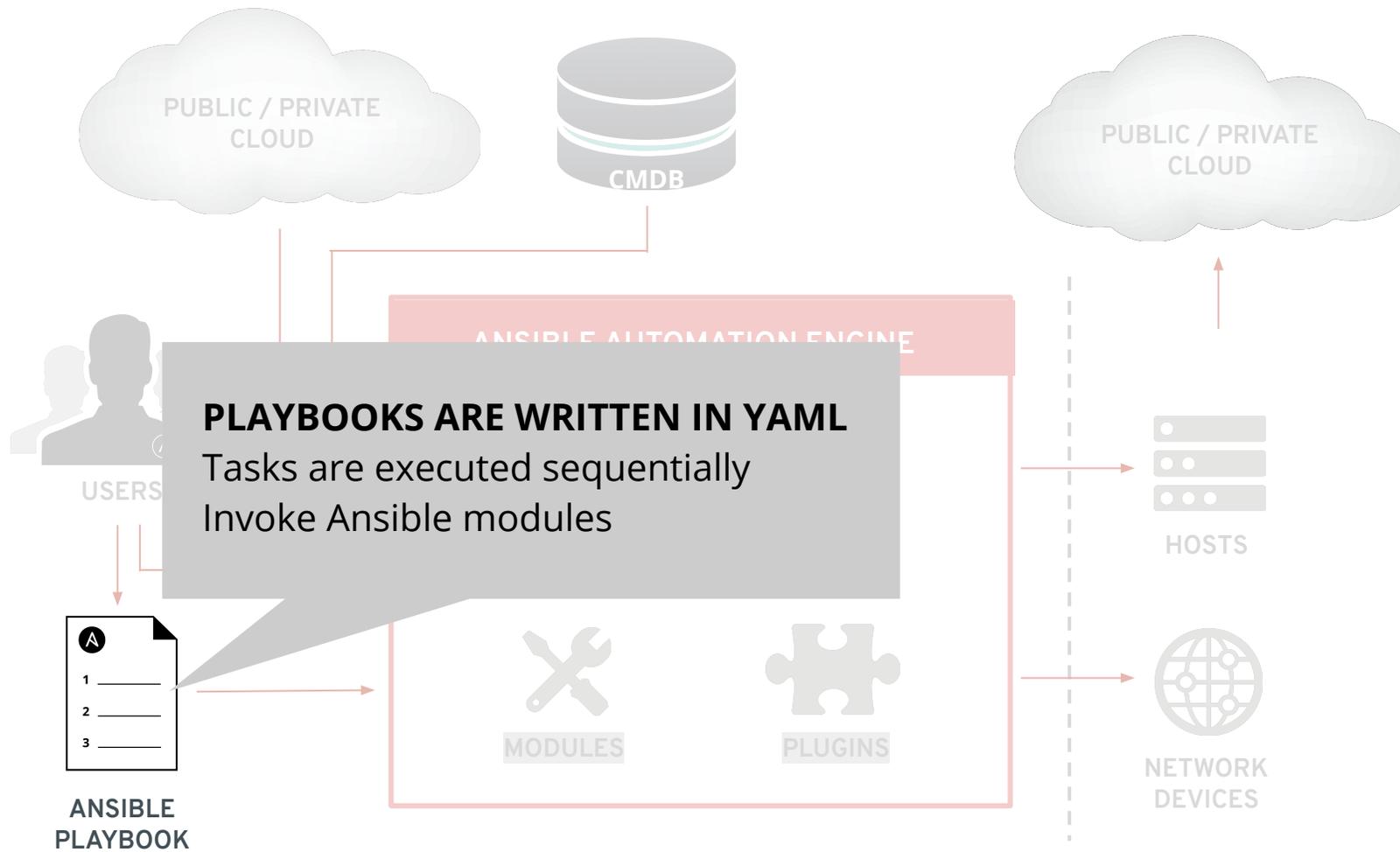
On these...



How does it work?





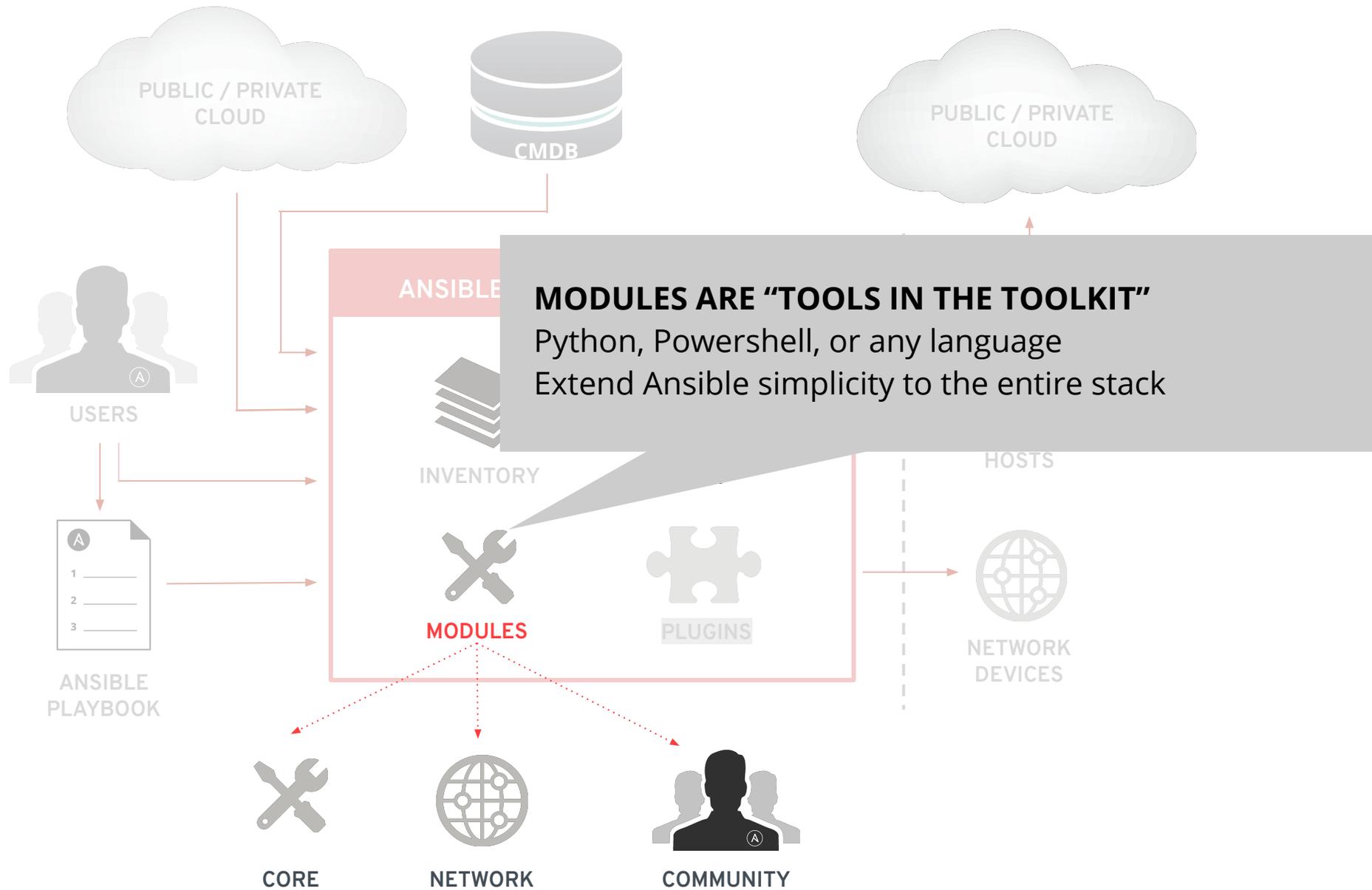


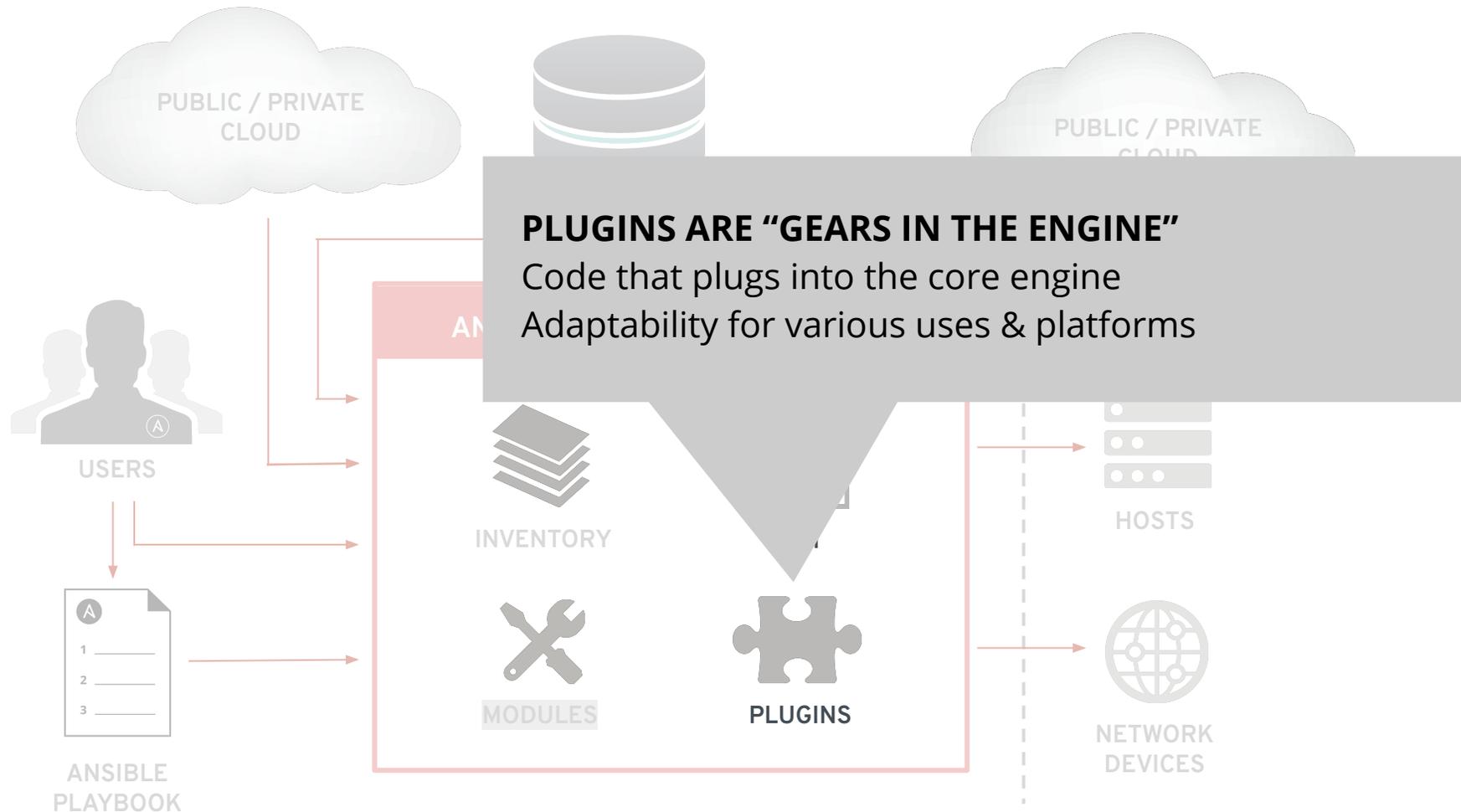
```
---
- name: install and start apache
  hosts: web
  become: yes
  vars:
    http_port: 80

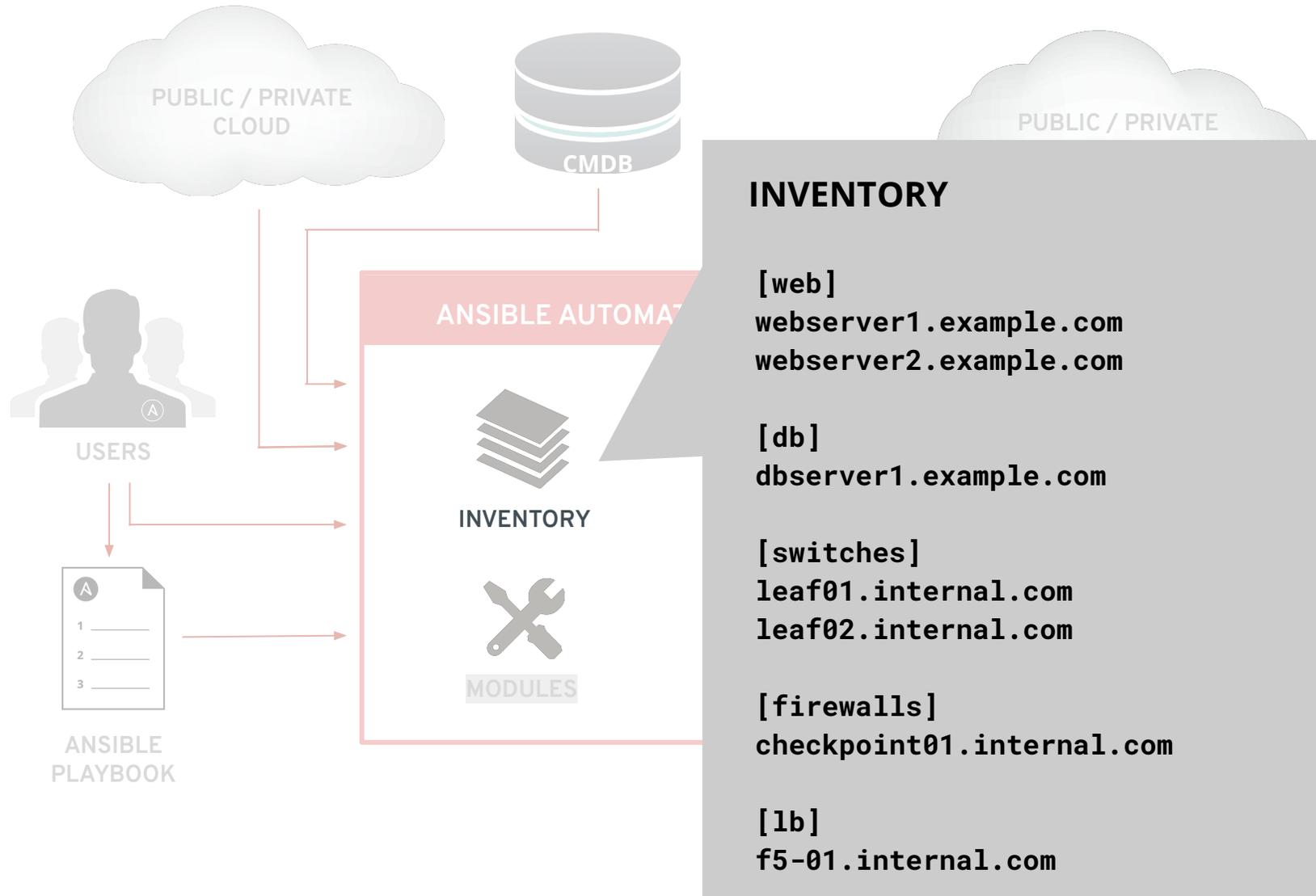
  tasks:
  - name: httpd package is present
    yum:
      name: httpd
      state: latest

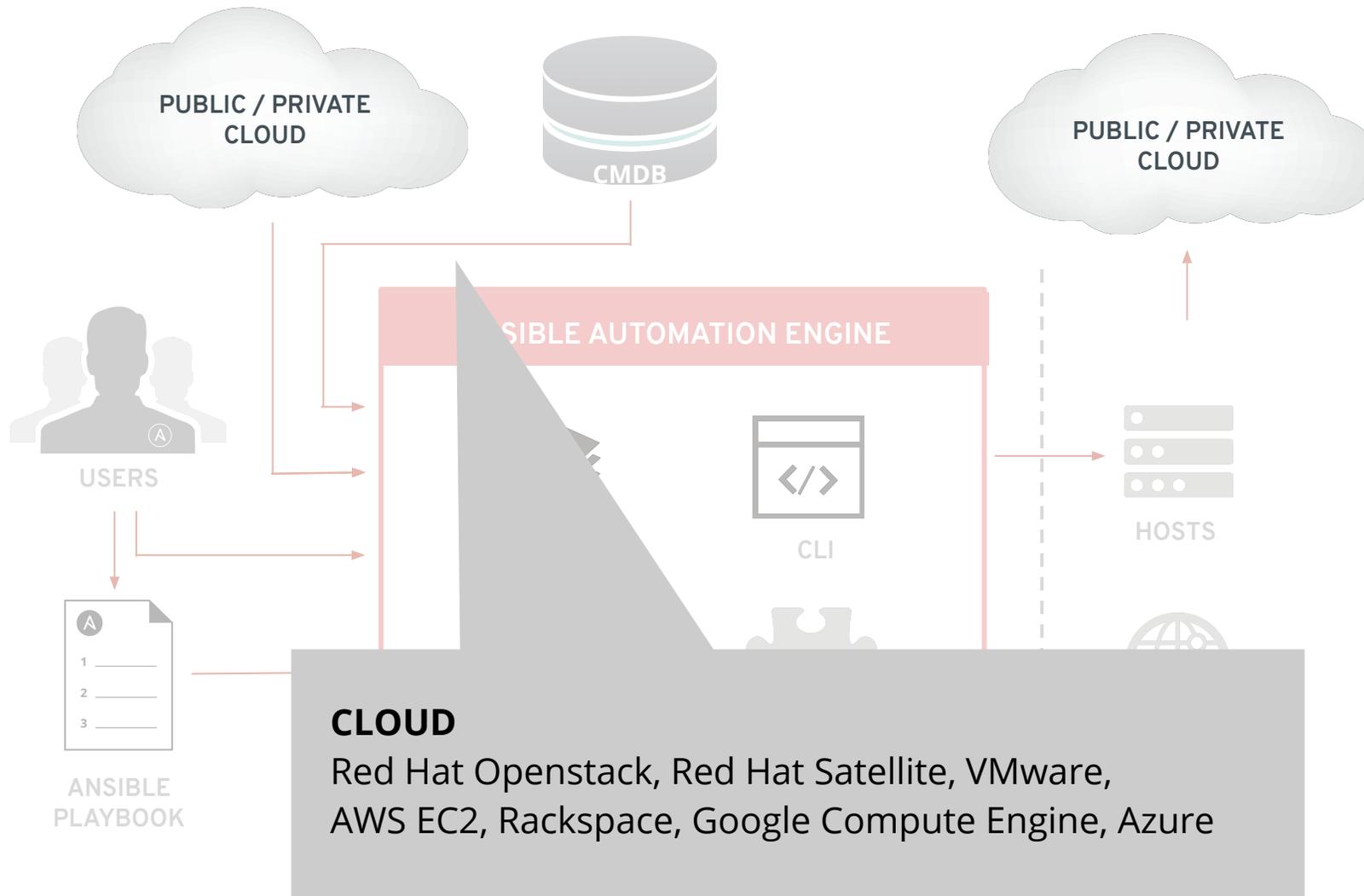
  - name: latest index.html file is present
    copy:
      src: files/index.html
      dest: /var/www/html/

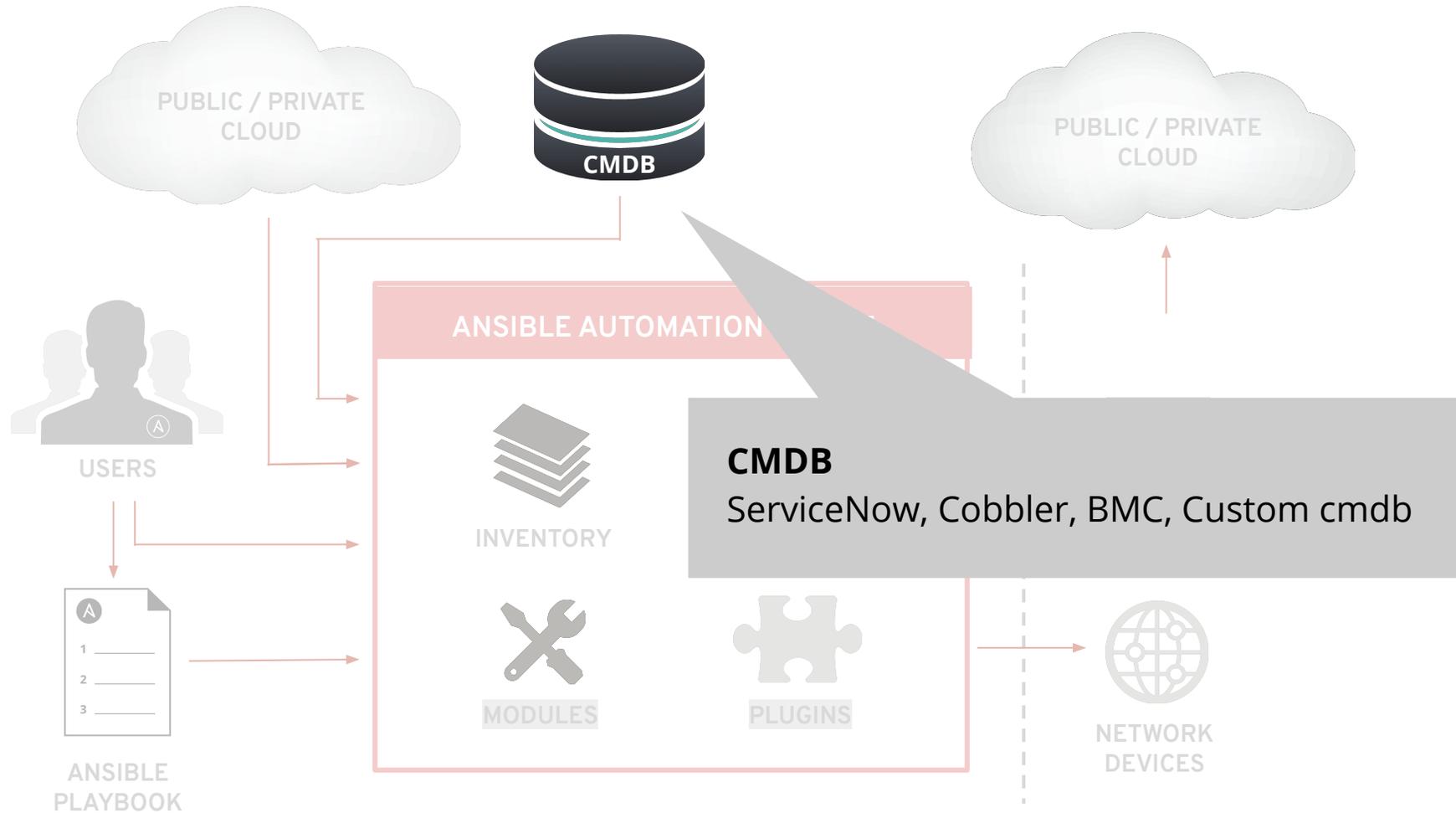
  - name: httpd is started
    service:
      name: httpd
      state: started
```

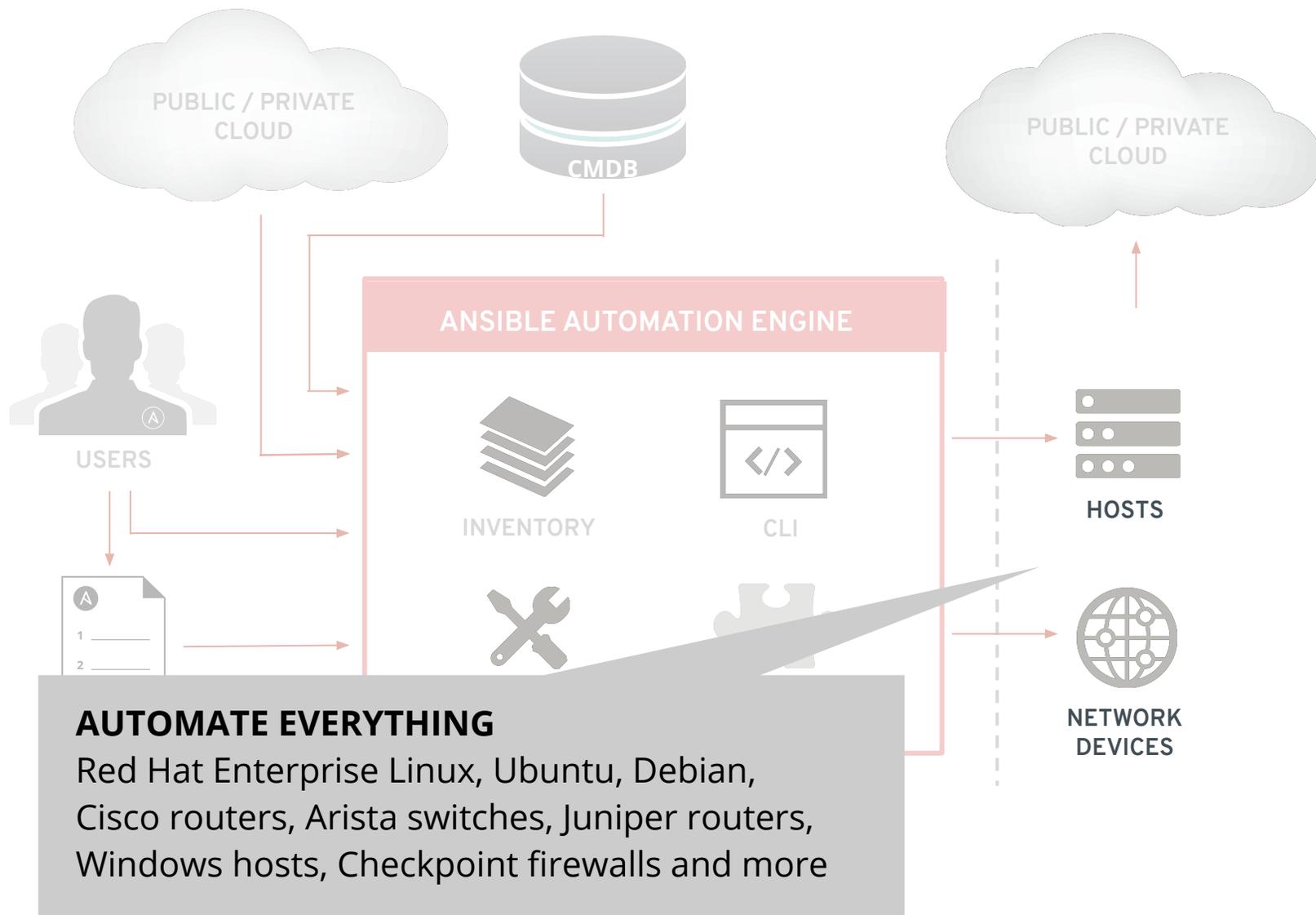












Ansible is a great tool for
configuration management

But it is so much more!

Not all of those modules are for Linux configuration management, obviously

Linux

Cisco

VMware

Azure

Windows

Citrix

RHV

Google

F5

Juniper

CyberArk

InfoBlox

Checkpoint

Arista

AWS

NetApp

And many, many more!

Not all of those modules are for Linux configuration management, obviously

WHICH ONES ARE YOU USING?

Linux
Windows

Cisco
Citrix

VMware
RHV

Apple
Google

F5

Juniper

CyberArk

InfoBlox

Checkpoint

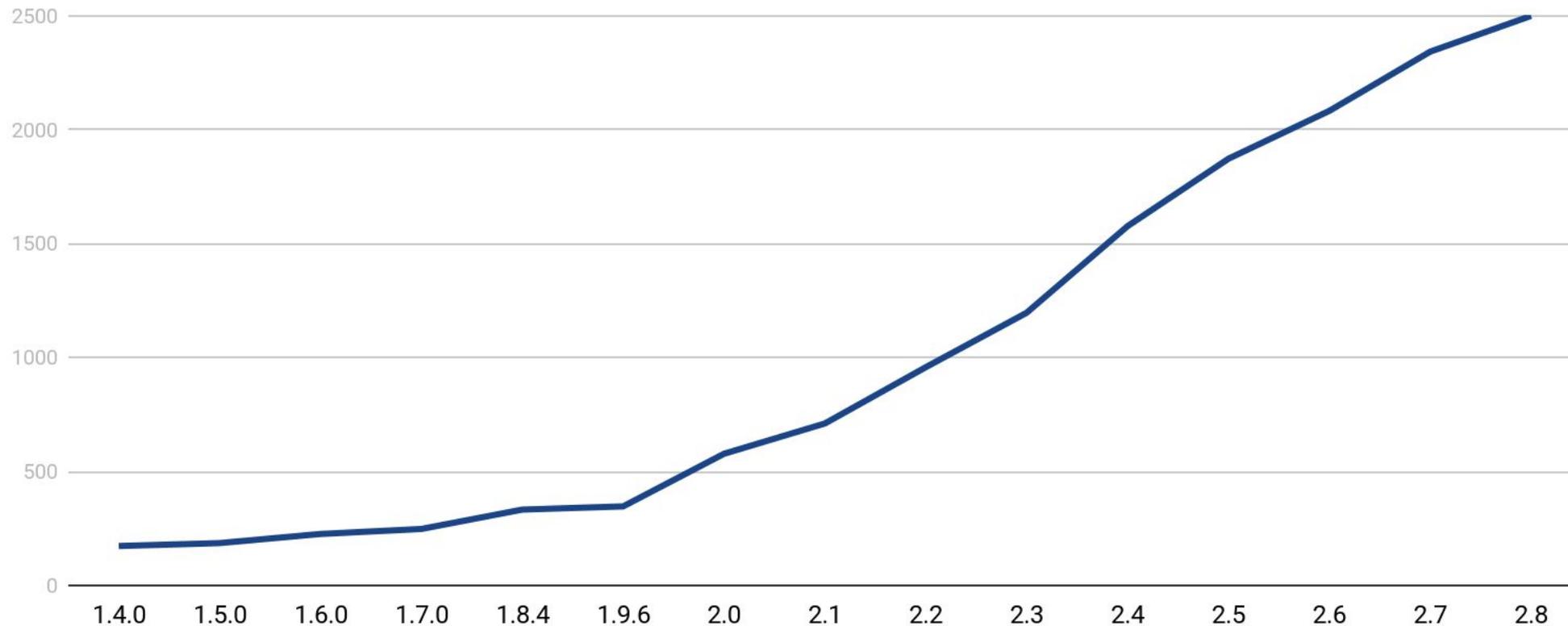
Arista

AWS

NetApp

And many, many more!

Number of Ansible modules over time *



* Based on my totally non-scientific research of checking out git tags, running various find commands and piping the output to wc

Ansible automates technologies you use

Time to automate is measured in minutes

Cloud

AWS
Azure
Digital Ocean
Google
OpenStack
Rackspace
+more

Operating Systems

RHEL and Linux
Unix
Windows
+more

Virt & Container

Docker
VMware
RHV
OpenStack
OpenShift
+more

Storage

Netapp
Red Hat Storage
Infinidat
+more

Windows

ACLs
Files
Packages
IIS
Regedits
Shares
Services
Configs
Users
Domains
+more

Network

Arista
A10
Cumulus
Bigswitch
Cisco
Cumulus
Dell
F5
Juniper
Palo Alto
OpenSwitch
+more

Devops

Jira
GitHub
Vagrant
Jenkins
Bamboo
Atlassian
Subversion
Slack
Hipchat
+more

Monitoring

Dynatrace
Airbrake
BigPanda
Datadog
LogicMonitor
Nagios
New Relic
PagerDuty
Sensu
StackDriver
Zabbix
+more

So, we automated all the things...

... now do we maintain this huge playbook together?

Well, first of all, you don't



Keep it simple

Complexity kills productivity

Ansible should not be like Perl!

Optimize for readability, no write-only code!

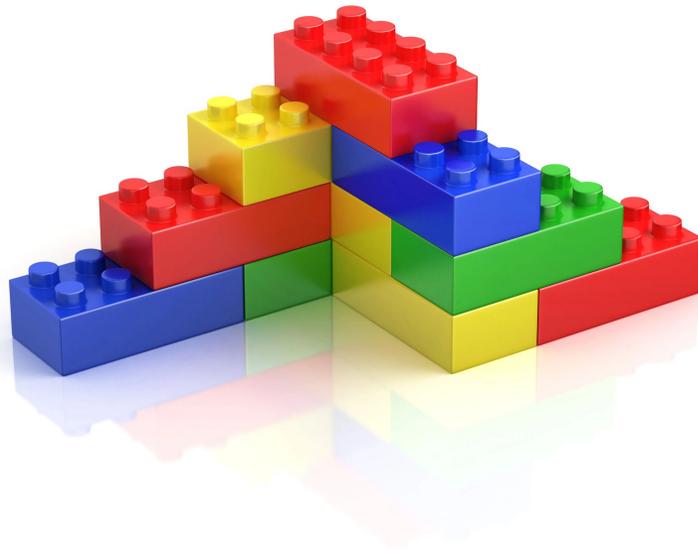
Think declaratively

Actually, no “code” at all. We are describing state.

Keep it small

If you keep the bricks small, you can build great things!

Re-use, share, and if you have to, steal ;)



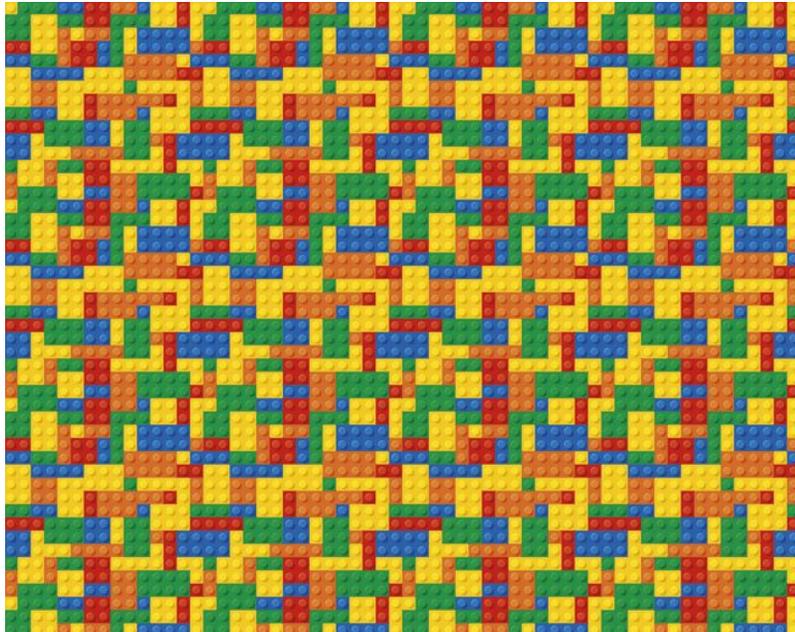
If you have written a great playbook once, use it again!
There is no merit in re-inventing the wheel

If you have written a great role, share it on Galaxy!
Be a good open source citizen ;)

If you need functionality, check Galaxy
Odds, are what you need already exists

Split off functionality into separate playbooks and chain them
Tower is great for this, but Ansible core can do it, too

A couple of tips to get you going



Split off provisioning from configuration

That makes it easy to move to another platform

If you use Ansible Tower, learn to love the `set_stats` module

Pass around that information

In your playbooks, mention all variables in the vars section

Makes it easier for your successor (and yourself!)

Use dynamic groups and / or dynamic inventories

(On some platforms, groups == tags)



Shall I do a little
demo at this
point?

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

 [linkedin.com/company/red-hat](https://www.linkedin.com/company/red-hat)

 [youtube.com/user/RedHatVideos](https://www.youtube.com/user/RedHatVideos)

 [facebook.com/redhatinc](https://www.facebook.com/redhatinc)

 twitter.com/RedHat