ANSIBLE

VISHWANATH M S

VISHWACLOUDLAB.ORG

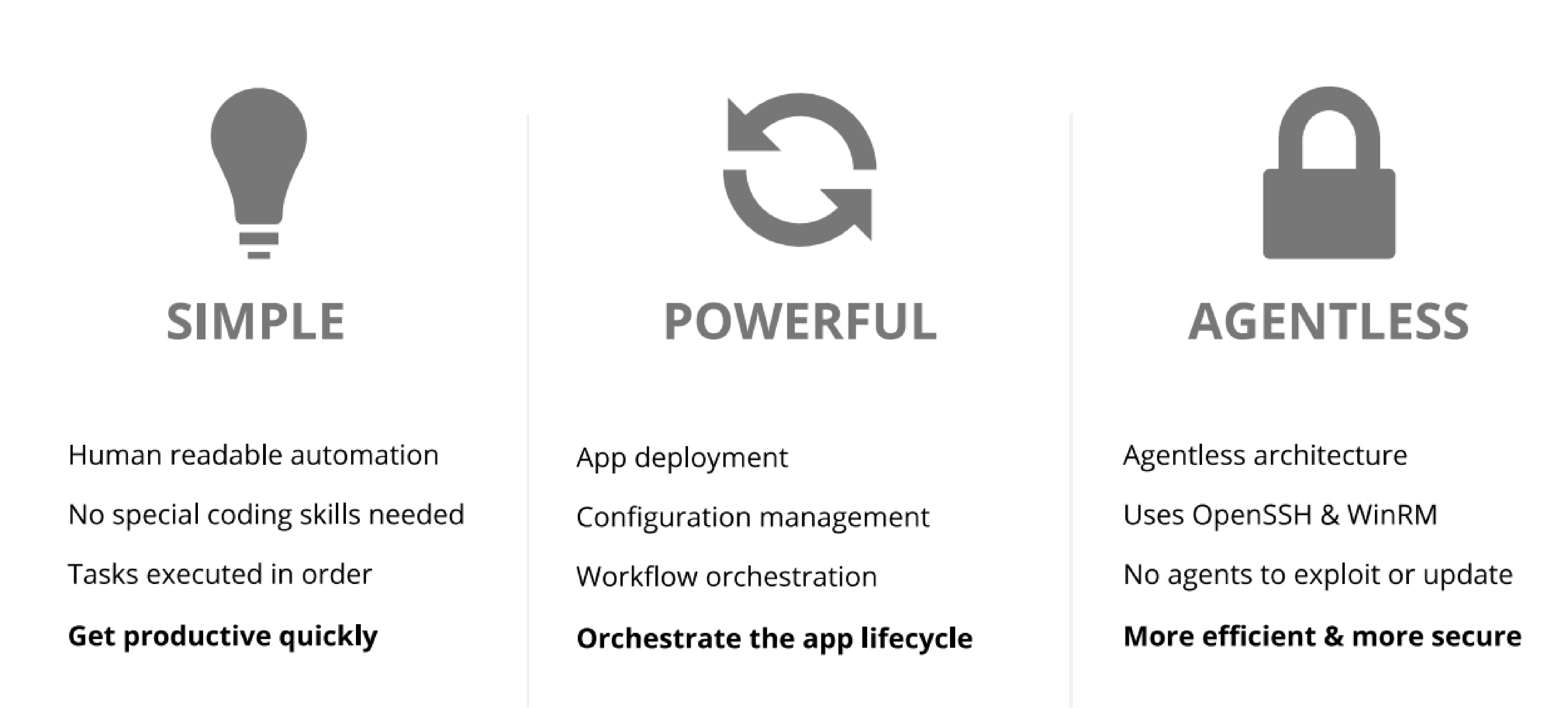
What is Ansible

It's a **simple automation language** that can perfectly describe an IT application infrastructure in Ansible Playbooks.

It's an **automation engine** that runs Ansible Playbooks.

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

Ansible is...



SIMPLE

Human readable automation

No special coding skills needed

Tasks executed in order

Get productive quickly

POWERFUL

App deployment

Configuration management

Workflow orchestration

Orchestrate the app lifecycle

AGENTLESS

Agentless architecture

Uses OpenSSH & WinRM

No agents to exploit or update

More efficient & more secure

Community

THE MOST POPULAR OPEN-SOURCE AUTOMATION COMMUNITY ON GITHUB

- 13,000+ stars & 4,000+ forks on GitHub
- 2000+ GitHub Contributors
- Over 900 modules shipped with Ansible
- New contributors added every day
- 1200+ users on IRC channel
- Top 10 open source projects in 2014
- World-wide meetups taking place every week
- Ansible Galaxy: over 18,000 subscribers
- 250,000+ downloads a month
- AnsibleFests in NYC, SF, London

- http://ansible.com/community

Features of Ansible

Open Source

Written in Python, so it’s easy to read and extend

Easy installation and configuration

Highly Scalable

Agent Less Client Configuration

Requirements for Ansible

SSH Client (OpenSSH on Linux)

Python (PyYAML, Jinja2)

Components of Ansible

Programs

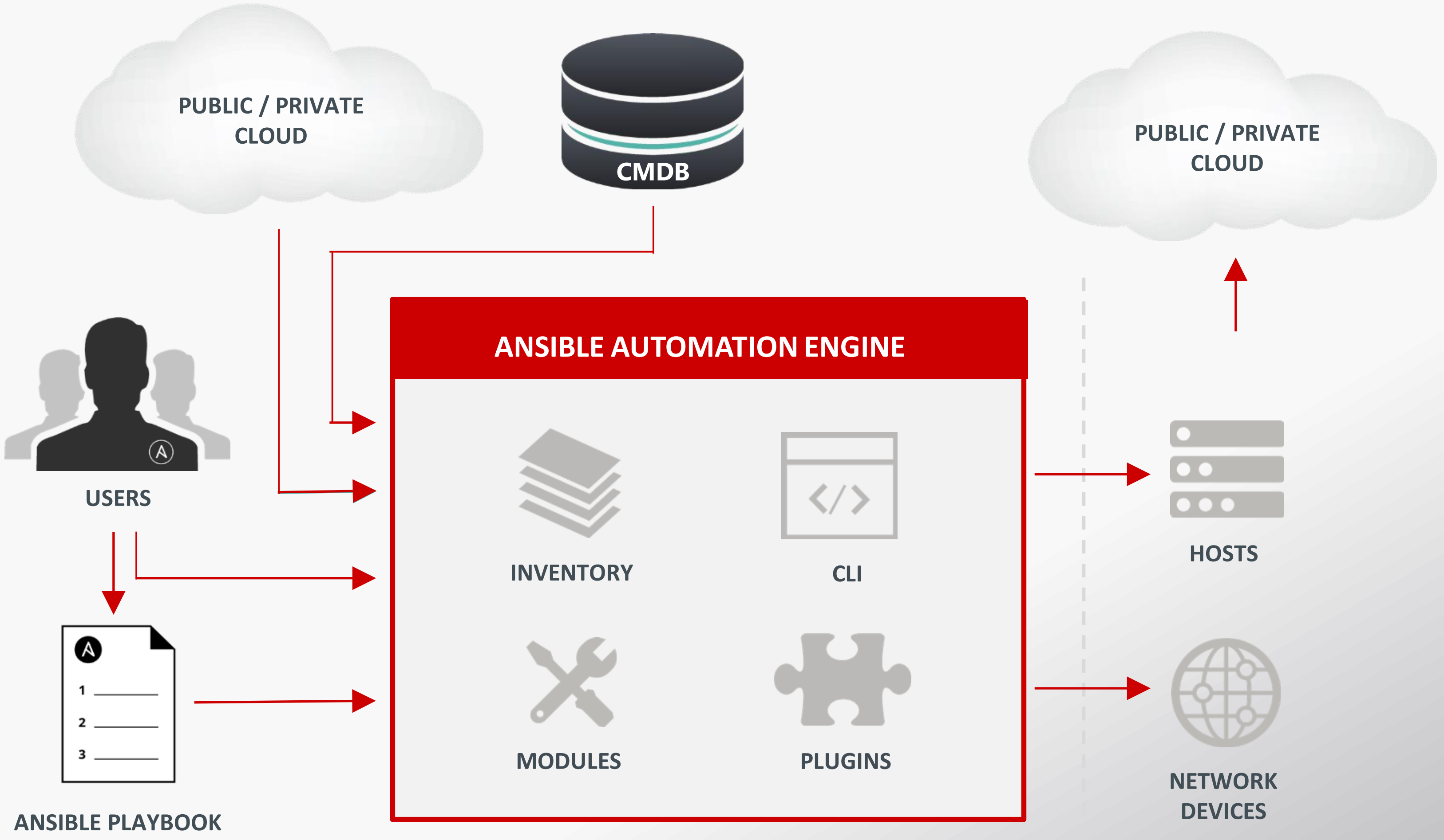
- ansible
- ansible-doc
- ansible-playbook
- ansible-pull

Modules

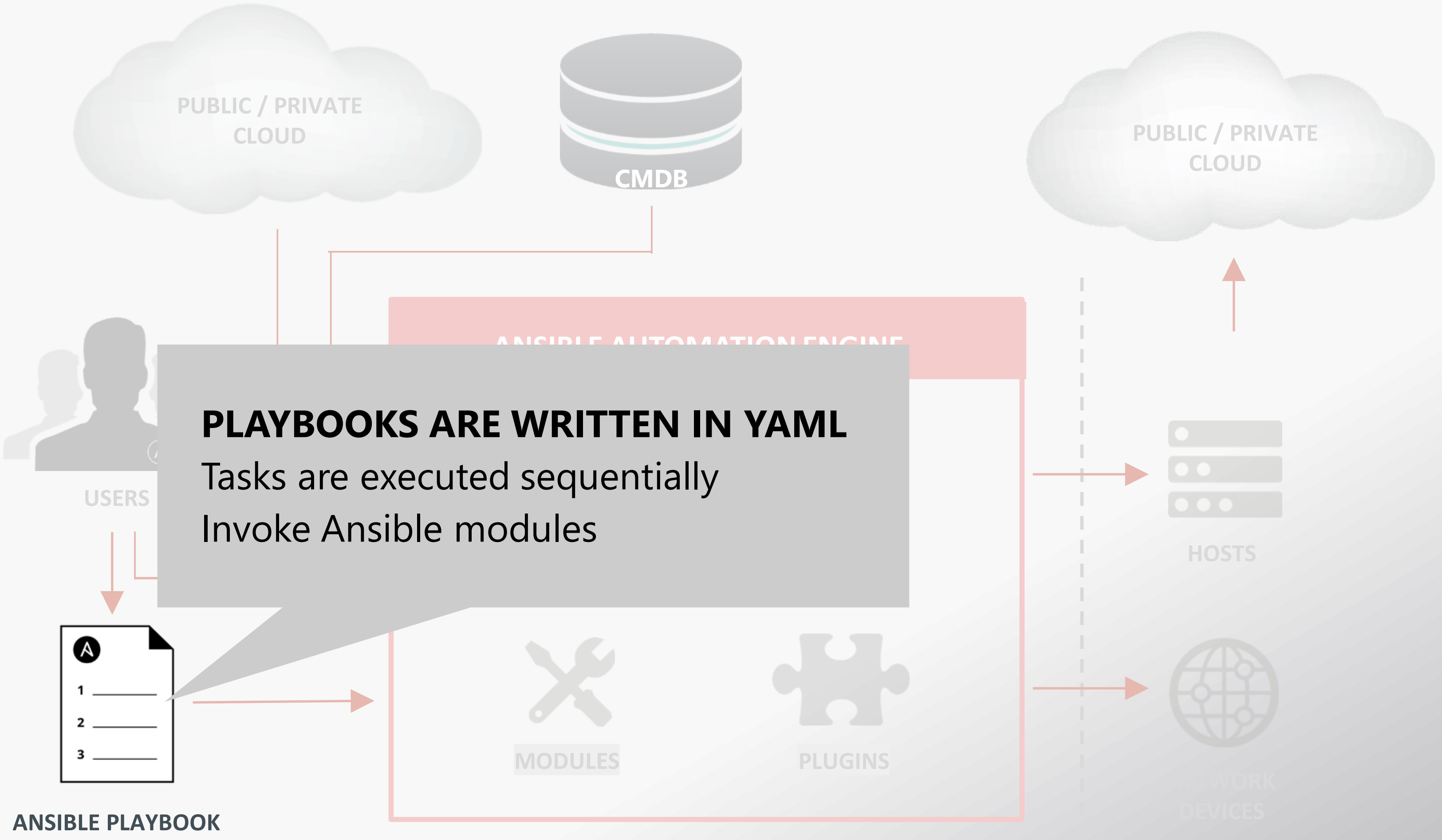
- Perform Configuration and system management.

  Example Modules copy, service, file, yum, user, group

How Ansible works



How Ansible works

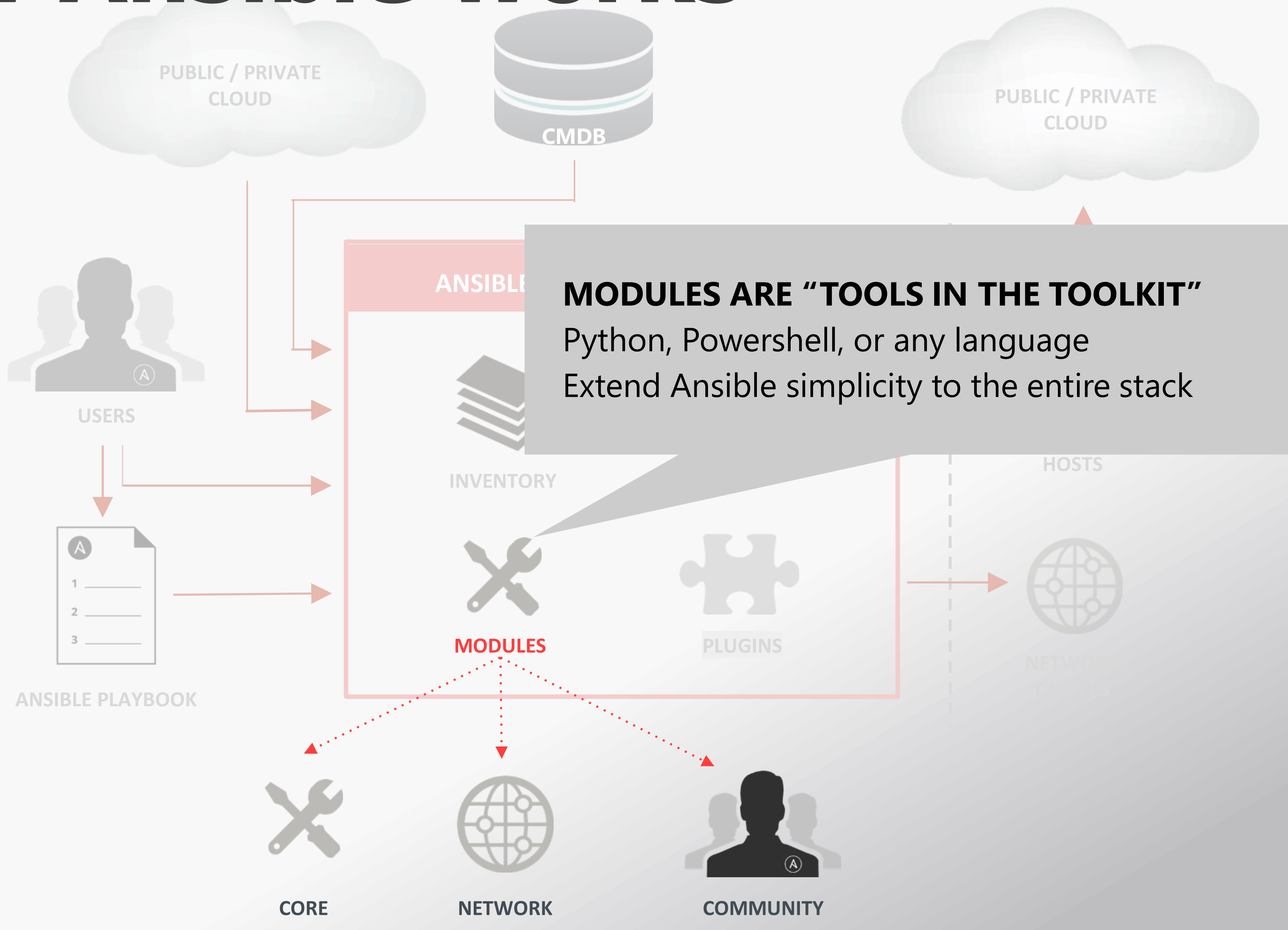


PLAYBOOKS ARE WRITTEN IN YAML

Tasks are executed sequentially

Invoke Ansible modules

How Ansible works

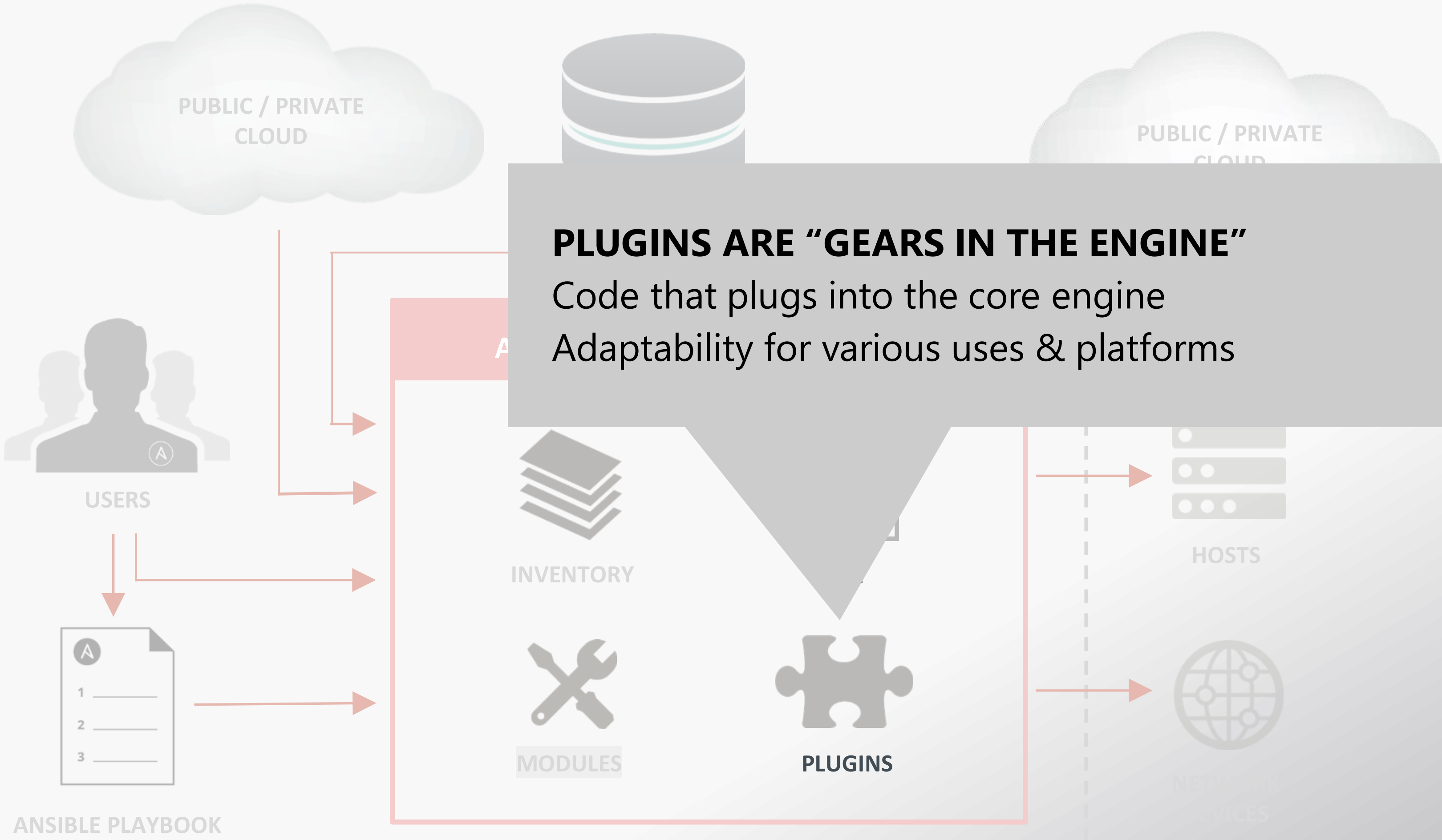


MODULES ARE “TOOLS IN THE TOOLKIT”

Python, Powershell, or any language

Extend Ansible simplicity to the entire stack

How Ansible works

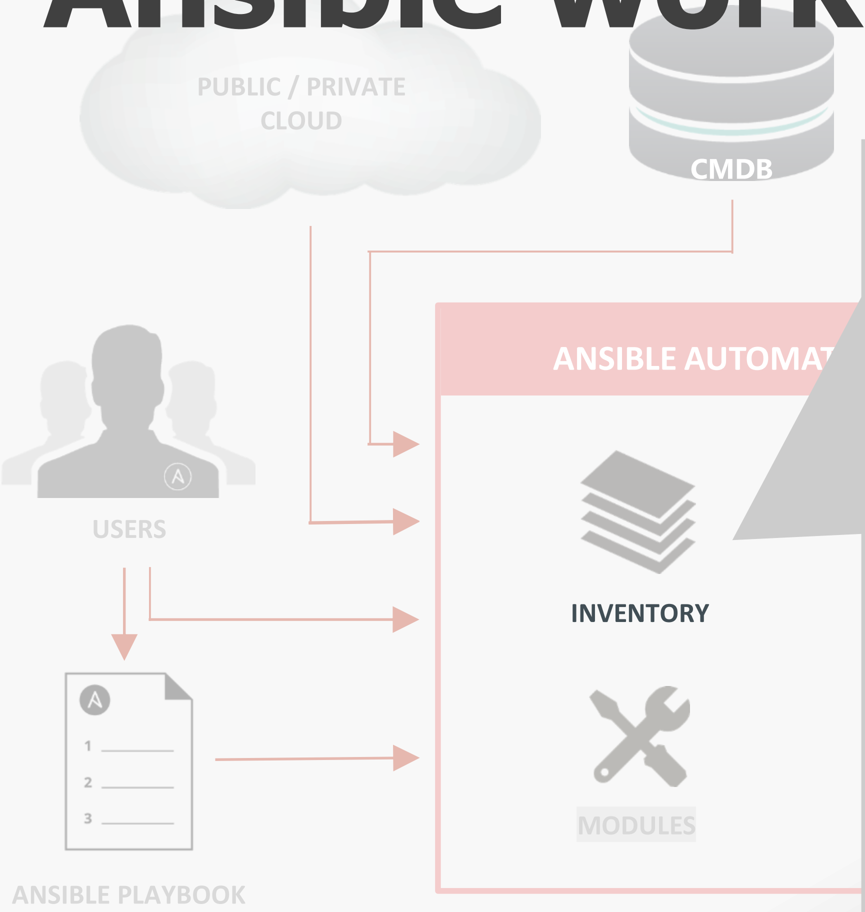


PLUGINS ARE “GEARS IN THE ENGINE”

Code that plugs into the core engine

Adaptability for various uses & platforms

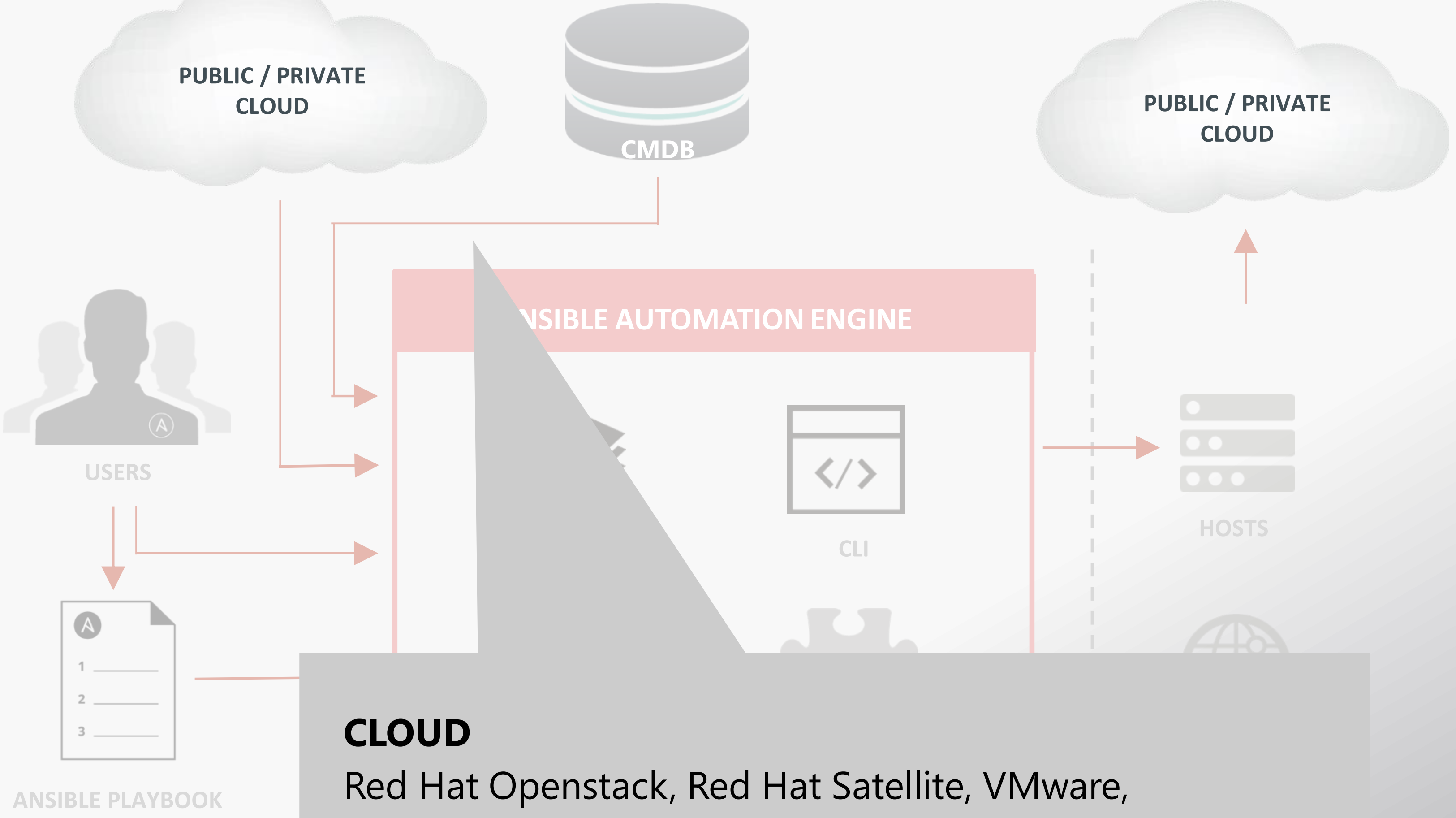
How Ansible works



INVENTORY

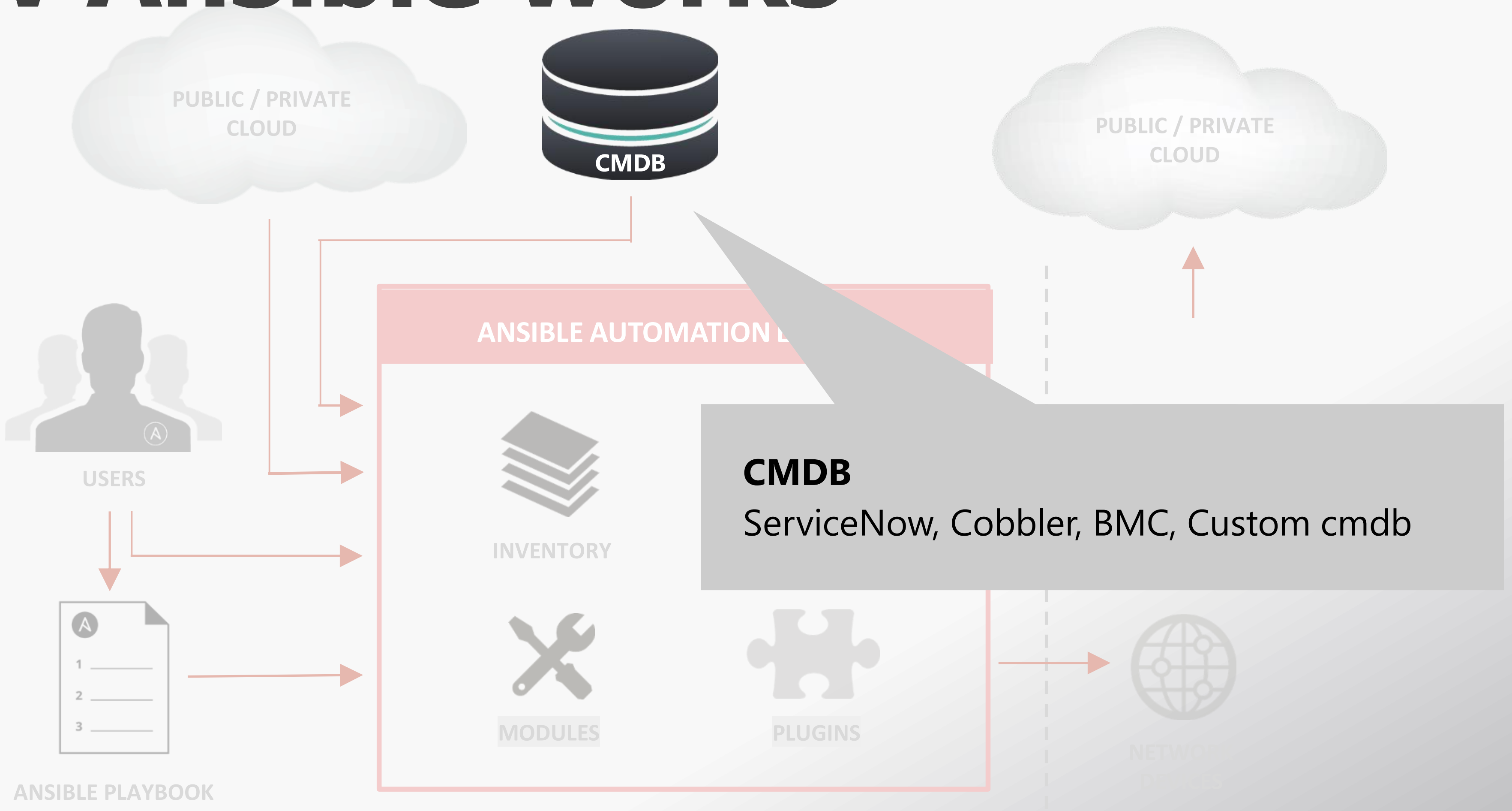
```
[web]
webserver1.example.com
webserver2.example.com

[db]
dbserver1.example.com

[switches]
leaf01.internal.com
leaf02.internal.com

[firewalls]
checkpoint01.internal.com

[lb]
f5-01.internal.com
```

How Ansible works



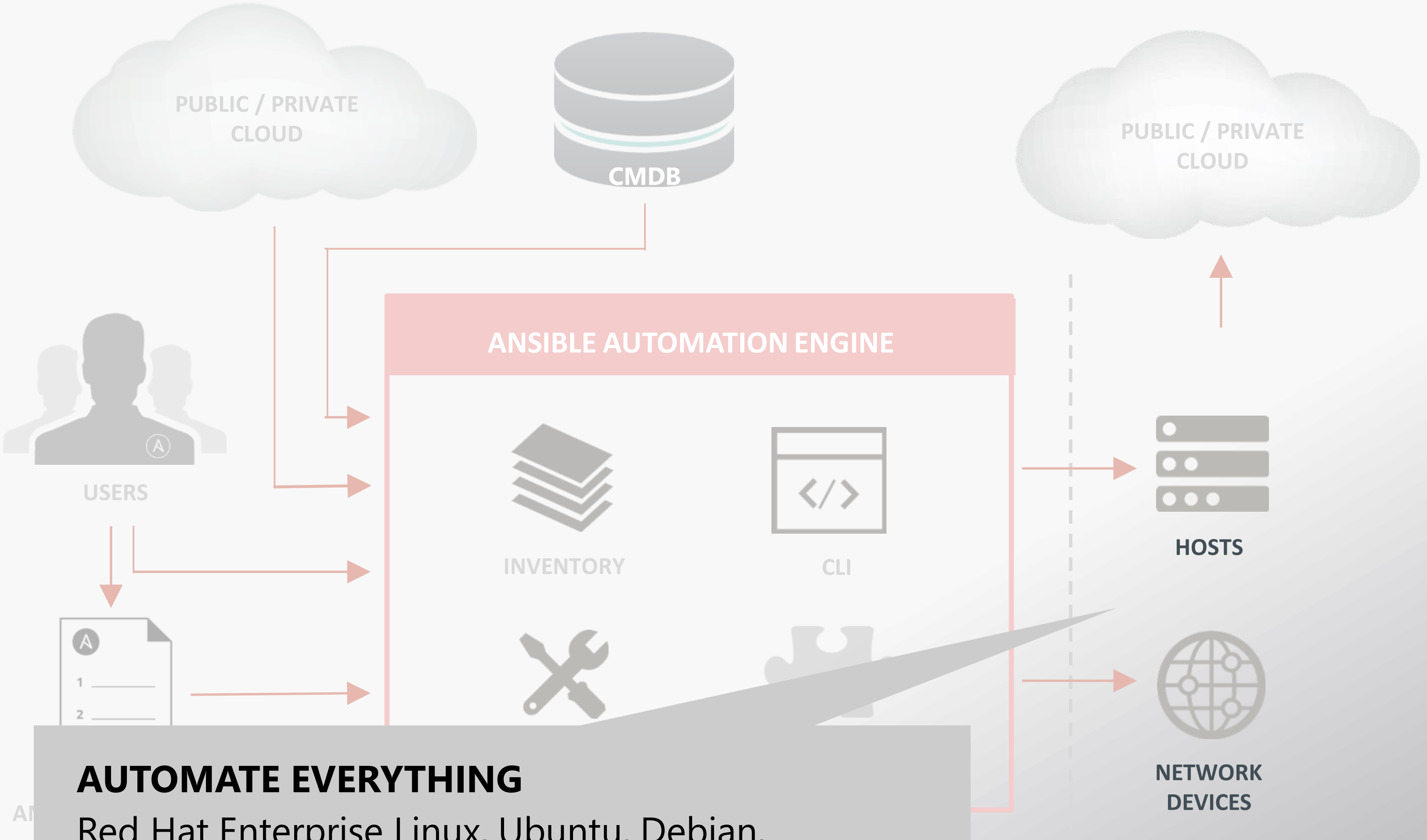
CLOUD
Red Hat Openstack, Red Hat Satellite, VMware,
AWS EC2, Rackspace, Google Compute Engine, Azure

How Ansible works



CMDB
ServiceNow, Cobbler, BMC, Custom cmdb

How Ansible works



AUTOMATE EVERYTHING
Red Hat Enterprise Linux, Ubuntu, Debian,
Cisco routers, Arista switches, Juniper routers,
Windows hosts, Checkpoint firewalls and more

Ansible Host file

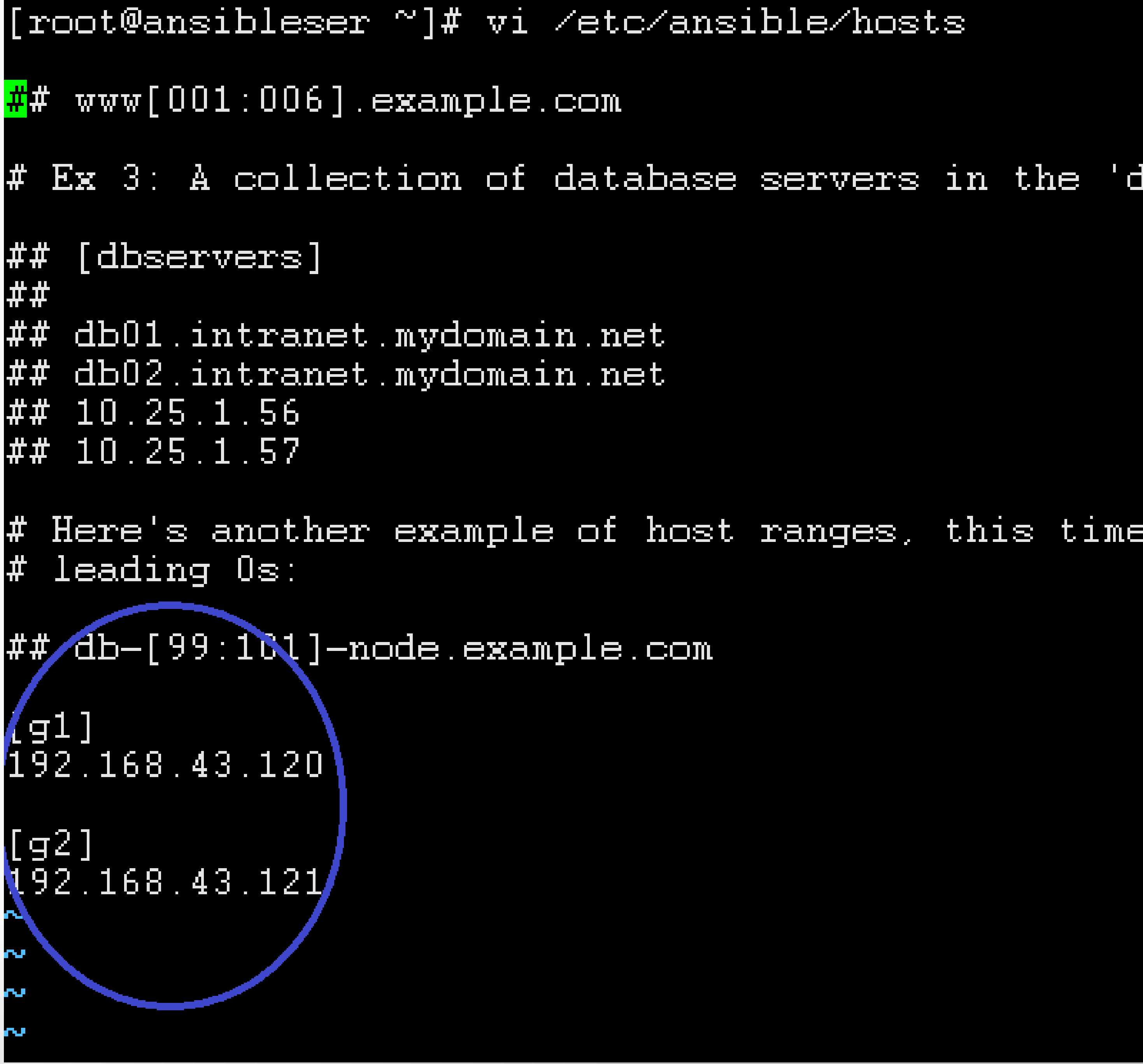
$ vi /etc/ansible/hosts

Add the list of hosts either IP address or HOSTNAME of the target Machines.

TO list the hosts for a group

$ ansible --list-hosts all

$ ansible --list-hosts g1



```
[root@ansibleser ~]# vi /etc/ansible/hosts

## www[001:006].example.com

# Ex 3: A collection of database servers in the 'd

## [dbservers]
##
## db01.intranet.mydomain.net
## db02.intranet.mydomain.net
## 10.25.1.56
## 10.25.1.57

# Here's another example of host ranges, this time
# leading 0s:

## db-[99:101]-node.example.com

[g1]
192.168.43.120

[g2]
192.168.43.121
~
~
~
~
```

Ansible Host file

If you have hosts that run on non-standard SSH ports you can put the port number after the hostname with a colon.

Eg:- host1.com:8039

If you are adding a lot of hosts following similar patterns, you can do this rather than listing each hostname.

Eg:-

[webservers]

www[01:50].example.com

[databases]

db-[a:f].example.com

Modules

Modules are bits of code transferred to the target system and executed to satisfy the task declaration. Ansible ships with several hundred today!

- apt/yum
- copy
- file
- get_url
- git
- ping
- debug
- service
- synchronize
- template

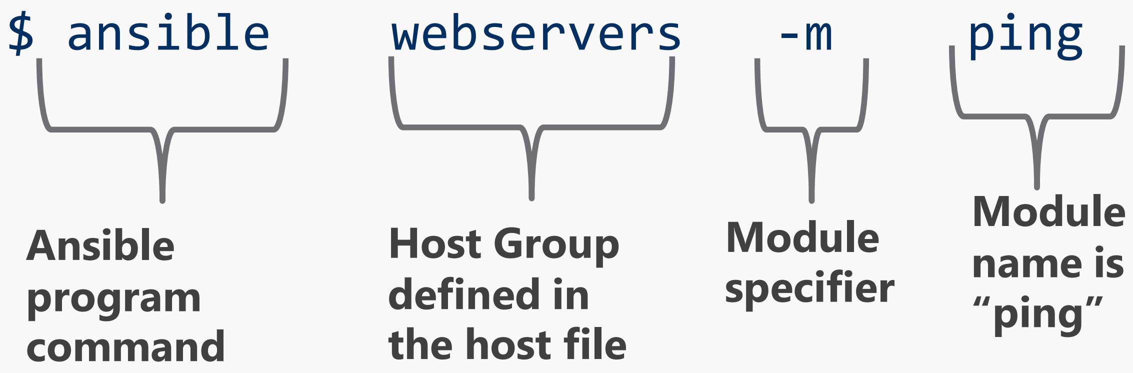
Types of Command Execution

- AD-HOC Commands

- PLAYBOOKS

AD-HOC - Module - ping

Ping the Hosts



```
$ ansible    webservers   -m    ping
```

Ansible program command

Host Group defined in the host file

Module specifier

Module name is “ping”

"

AD-HOC - Module - yum

Install a package, but don’t update it:

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

Install a package at the latest version:

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

Uninstall a package:

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

Modules: Run Commands

If Ansible doesn’t have a module that suits your needs there are the **“run command”** modules:

- **command:** Takes the command and executes it on the host. The most secure and predictable.
- **shell:** Executes through a shell like /bin/sh so you can use pipes etc. Be careful.
- **script:** Runs a local script on a remote node after transferring it.
- **raw:** Executes a command without going through the Ansible module subsystem.

Modules Documentation

http://docs.ansible.com/