Ansible

framework for automated system administration

David Morgan

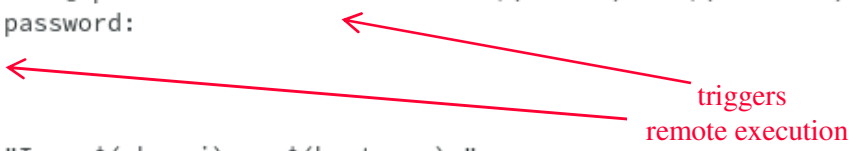
Requirements

- originated at RedHat
- uses ssh's remote execution capability
- target machines must run sshd
  - must have python for most activities
  - need no ansible specific agent

ssh function

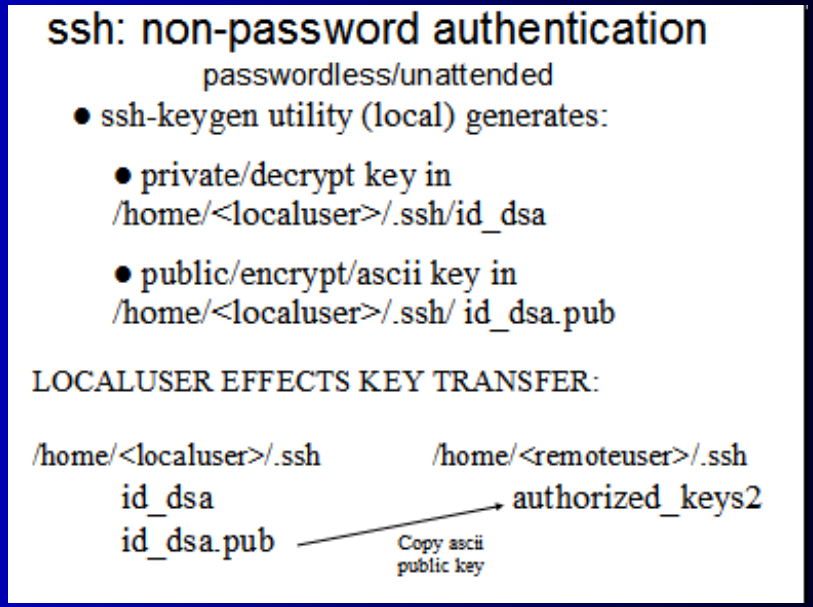
- launch commands in one machine from another
- employed by ansible

```
[root@managed1 ~]#
[root@managed1 ~]# echo "I am $(whoami) on $(hostname)."
I am root on managed1.
[root@managed1 ~]#
[root@managed1 ~]#
[root@managed1 ~]# ssh david@sputnik.smc.edu 'echo "I am $(whoami) on $(hostname)."'
david@sputnik.smc.edu's password:
I am david on sputnik.
[root@managed1 ~]#
[root@managed1 ~]#
[root@managed1 ~]# echo "I am $(whoami) on $(hostname)."
I am root on managed1.
[root@managed1 ~]#
[root@managed1 ~]#
```



ssh access

- must be provided for and setup by sysadmin
- combinations of ssh keys, passwords, ssh-agent
  - unattended authentication desirable



Privilege escalation

- must be provided for and setup by sysadmin
- combinations of su and/or sudo

```
[root@managed1 ~]#
[root@managed1 ~]# /usr/bin/grep -B1 -A3 "anywhere" /etc/sudoers
##
## Allow root to run any commands anywhere
root    ALL=(ALL)       ALL
student ALL=(root) NOPASSWD:    ALL
[root@managed1 ~]#
```


via sudo, student can have any command executed as root

Inventory file defines population

```
[root@instructor ~]# cat /etc/ansible/hosts
# This is the default ansible 'hosts' file.
#
# It should live in /etc/ansible/hosts
#
#   - Comments begin with the '#' character
#   - Blank lines are ignored
#   - Groups of hosts are delimited by [header] elements
#   - You can enter hostnames or ip addresses
#   - A hostname/ip can be a member of multiple groups

# Ex 1: Ungrouped hosts, specify before any group headers.

## green.example.com
## blue.example.com
## 192.168.100.1
## 192.168.100.10

[webservers]
192.168.1.67    ansible_user=student
192.168.1.98    ansible_user=student
#172.16.44.138  ansible_user=student
#172.16.44.139  ansible_user=student


# Ex 2: A collection of hosts belonging to the 'webservers' group

## [webservers]
## alpha.example.org
## beta.example.org
## 192.168.1.100
## 192.168.1.110

# If you have multiple hosts following a pattern you can specify
# them like this:

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

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

## [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 there are no
# leading 0s:

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

[root@instructor ~]#
```

Ansible modules

- specialized to do particular things
- or manage particular products
  - requires knowledge of the product

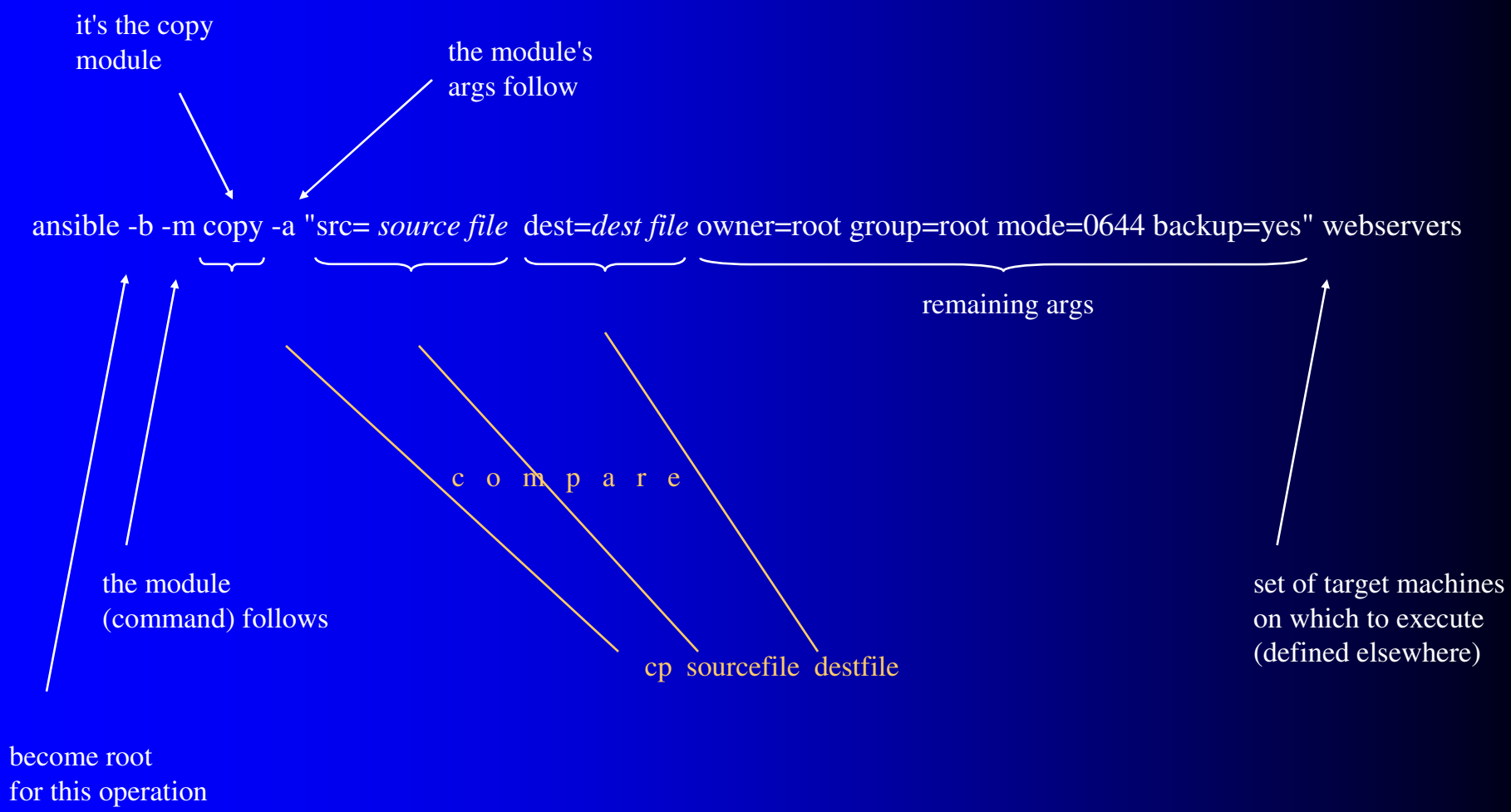
- can be written/contributed by programmers
- over 1000 provided modules

  https://docs.ansible.com/ansible/2.9/modules/list_of_all_modules.html

Some modules

- ping - try to connect to host, verify a usable python and return pong on success
- command - execute commands on targets
- shell - execute shell commands on targets
- dnf - manages packages with the dnf package manager
- copy - copy files to remote locations
- file - manage files and file properties
- setup - gathers facts about remote hosts

An ansible command



Ansible playbooks

- formatted text files listing things for ansible to do
- modules to run, aggregated and sequenced
- supplemented with
  - variables
  - control structures (if, while)

A playbook

```
---

- hosts: webservers
  become: yes
  tasks:
   - name: install apache this way
     apt: name=apache2 update_cache=yes state=latest
     notify: start apache2
     when: ansible_os_family == "Debian"

   - name: install apache that way
     dnf: name=httpd state=latest
     notify: start httpd
     when: ansible_os_family == "RedHat"

  handlers:
   - name: start apache2
     service: name=apache2 enabled=yes state=started

   - name: start httpd
     service: name=httpd enabled=yes state=started
```

install apache on all "webservers"

as root

if the server is Debian based, do it debian's way

if the server is RedHat based, do it redhat's way

after inststalling it, start it