

IT Automation with



\$ whoami

galaxy_info:

- name: Joel Rosental R

description:

- DevOps enthusiast
- Linux sysadmin
- Automation fan

email: jrosental@gmail.com



Agenda

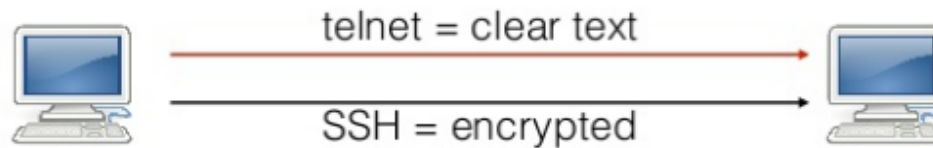
- Ansible and configuration management overview
- Ansible installation
- YAML introduction
- Ansible Inventory
- Playbooks
- Ansible modules

Evaluation questions

- Ansible?
- Configuration Management tools?
- SSH?
- DevOps?
- Virtual Machines?
- Vagrant?
- Public key infrastructure (PKI)?

Quick intro to SSH

What is SSH

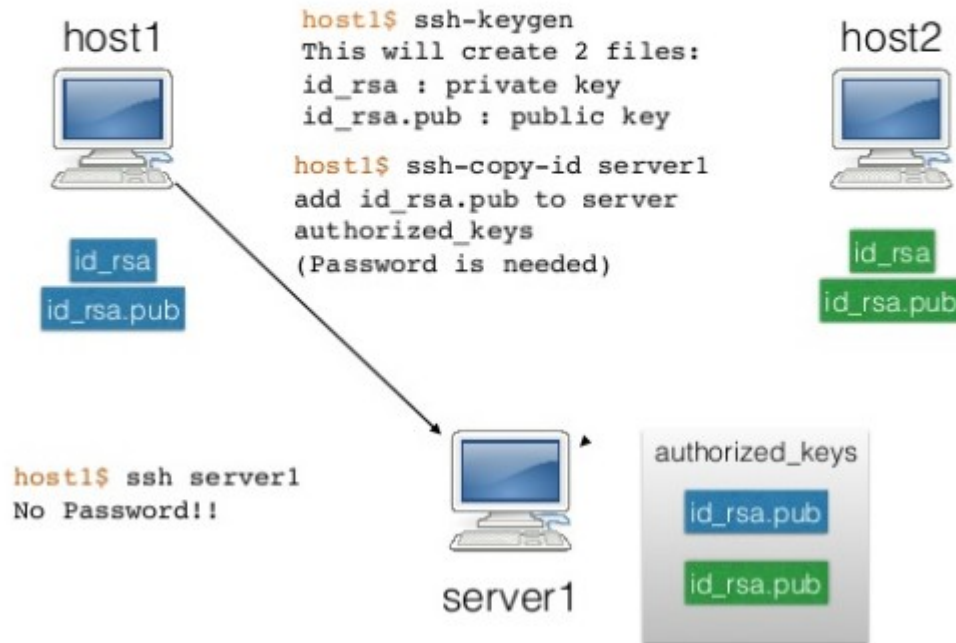


- SSH have more goodies:
 - Access using Keys / Password less
 - Compression
 - Secure File Transfer (scp, sftp)
 - Tunneling

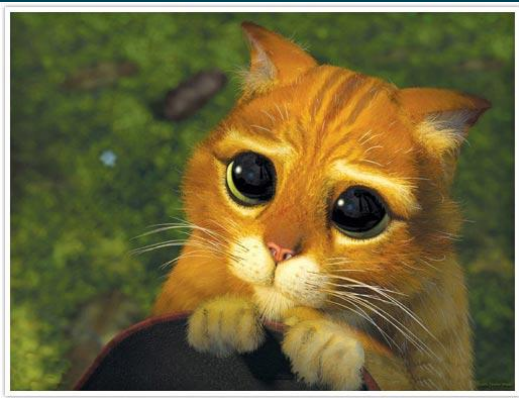
SSH is acronym for
Secure Shell

Quick intro to SSH

SSH Keys



Are your servers Pets or Cattle?



- Pets are given names like `webserver-prod.devoteam.com`
 - They are unique, lovingly hand raised and cared for.
 - When they get ill, you nurse them back to health
-
- Cattle are given numbers `webserver-123.devoteam.com`
 - They are almost identical to other cattle.
 - When they get ill, you get another one

Source: <http://www.slideshare.net/gmccance/cern-data-centre-evolution>

Handcrafted servers are

- Hard to maintain
- Not so robust
- Everyone's rolling their own
- Not automatic step by step reporting
- Not possible to execute a subset of tasks by tagging them
- Setup is not easily reproducible
- Can be difficult to debug on complex scenarios

Once upon a time

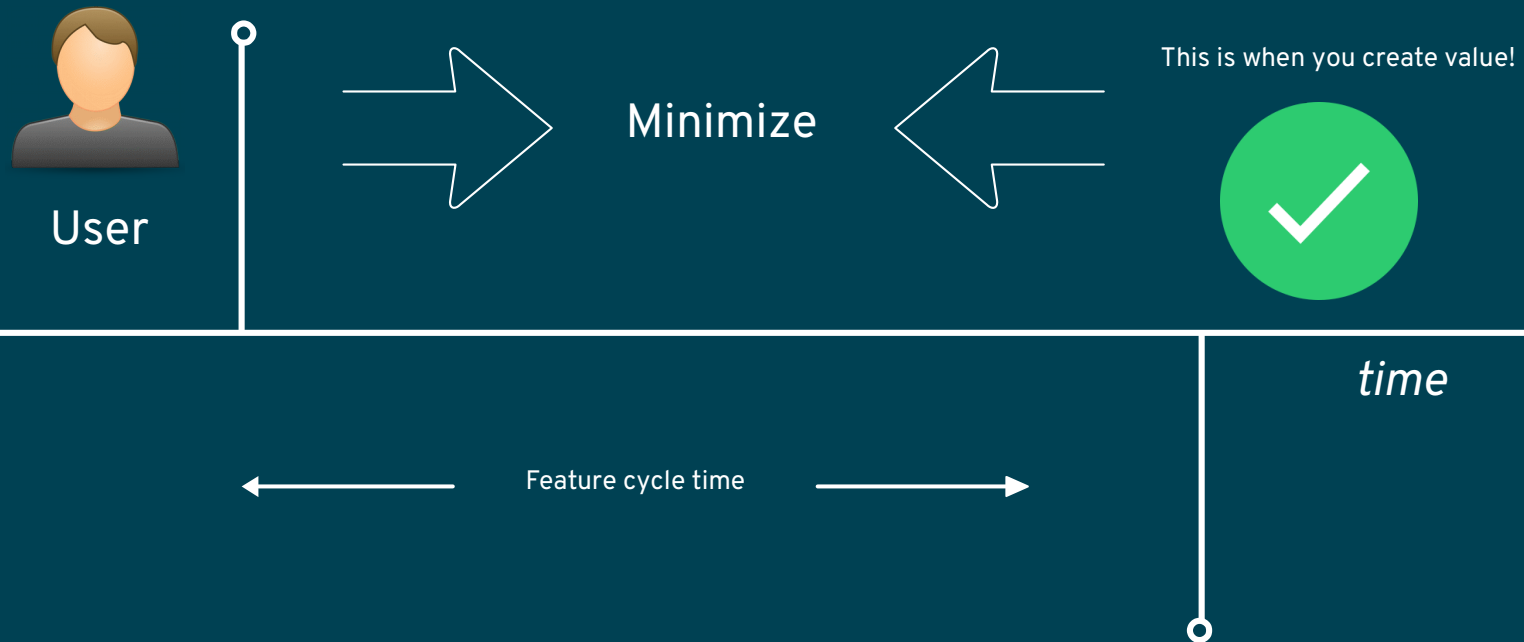
- Developers only wrote code...
- Ops guys deployed code...





Until one day...

Utmost Goal: Minimize Cycle Time



It's all about getting your features into your user's hand ASAP!

Agile manifesto: Principle #1

“Our highest priority is to satisfy the customer through early and continuous delivery of valuable software”

Automated deployments

How?

- Infrastructure as Code! (IaaC)
 - Keep everything that affects application state in Version Control:
 - Code
 - Configuration
 - Data
 - Align **D**evelopment and **O**perations.
-

Shell Scripts

```
# Install the PGP key
gpg --keyserver keyserver.ubuntu.com --recv-keys 561F9B9CAC40B2F7
gpg --armor --export 561F9B9CAC40B2F7 | apt-key add -

# Install https support for apt
apt-get install apt-transport-https -y

# Add the passenger apt repository
echo "deb https://oss-binaries.phusionpassenger.com/apt/passenger
raring main" > /etc/apt/sources.list.d/passenger.list
chown root: /etc/apt/sources.list.d/passenger.list
chmod 600 /etc/apt/sources.list.d/passenger.list

# Update the apt cache so we can use the new repo
apt-get update

# Install nginx
apt-get install nginx-full passenger -y

# Set up passenger in the nginx configuration
sed -i "s/# passenger_root/passenger_root/" /etc/nginx/nginx.conf
sed -i "s/# passenger_ruby/passenger_ruby/" /etc/nginx/nginx.conf

# Start nginx
service nginx restart
```

Good! isn't it?



What's wrong with shell scripts?

- Not idempotent
- Not so robust
- Everyone's rolling their own
- Not automatic step by step reporting
- Not possible to execute a subset of tasks by tagging them

What are the benefits of using a configuration management tool? (1/2)

- Quick provisioning of new servers / applications / configurations, etc.
- Reduce service interruption time
- Test the infrastructure
- Reduce the risk of accidents
- Seamless integration among different environments, e.g: dev, staging, prod
- Living documentation

What are the benefits of using a configuration management tool? (2/2)

	Human	Automate
Repeat Costs	High	Low
Human Error	High	Low
Testability	Hard	Easy
Modularization	Hard	Easy
Get off work early	Hard	Easy

Choosing a Configuration Management tool

- Infrastructure Complexity: scalability, security, etc.
- Learning Curve: Domain Specific Language (DSL), complementary tools, etc.
- Costs
- Extensibility
- Community and support

What is Ansible? (1/2)

- Open source automation platform
- Applies changes to your system to bring it to a desired state
- Automation tool that can be used in:
 - *Cloud provisioning*
 - *Configuration management*
 - *Application deployment*
 - *Intra-service orchestration*
- Several alternatives exists although some may change in their approach, e.g: puppet, chef, etc.

What is Ansible? (2/2)

- **Simple**
 - *Playbooks provide human-readable automation*
 - *Code skills are not needed.*
- **Powerful**
 - *As mentioned earlier, it can work on several use cases*
- **Agentless!**
- **Efficient and easy to setup**
- **Secure by default**
- **Scales better than other centralized alternatives.**

Ansible strenghts

- **Cross-platform support:** Linux, Windows, UNIX, network devices, in physical, virtual, cloud and container environments.
- **Human-readable automation:** Playbooks are written as YAML text files, are easy to read to help ensure everyone understand what they do.
- **Perfect description of applications:** Every change can be made by Ansible Playbooks
- **Easy to manage in VCS**
- **Support for dynamic inventories**
- **Orchestration that integrates easiliy with other systems, e.g: Jenkins, containers, etc.**

Some fresh Github stats

 [ansible](#) / [ansible](#)

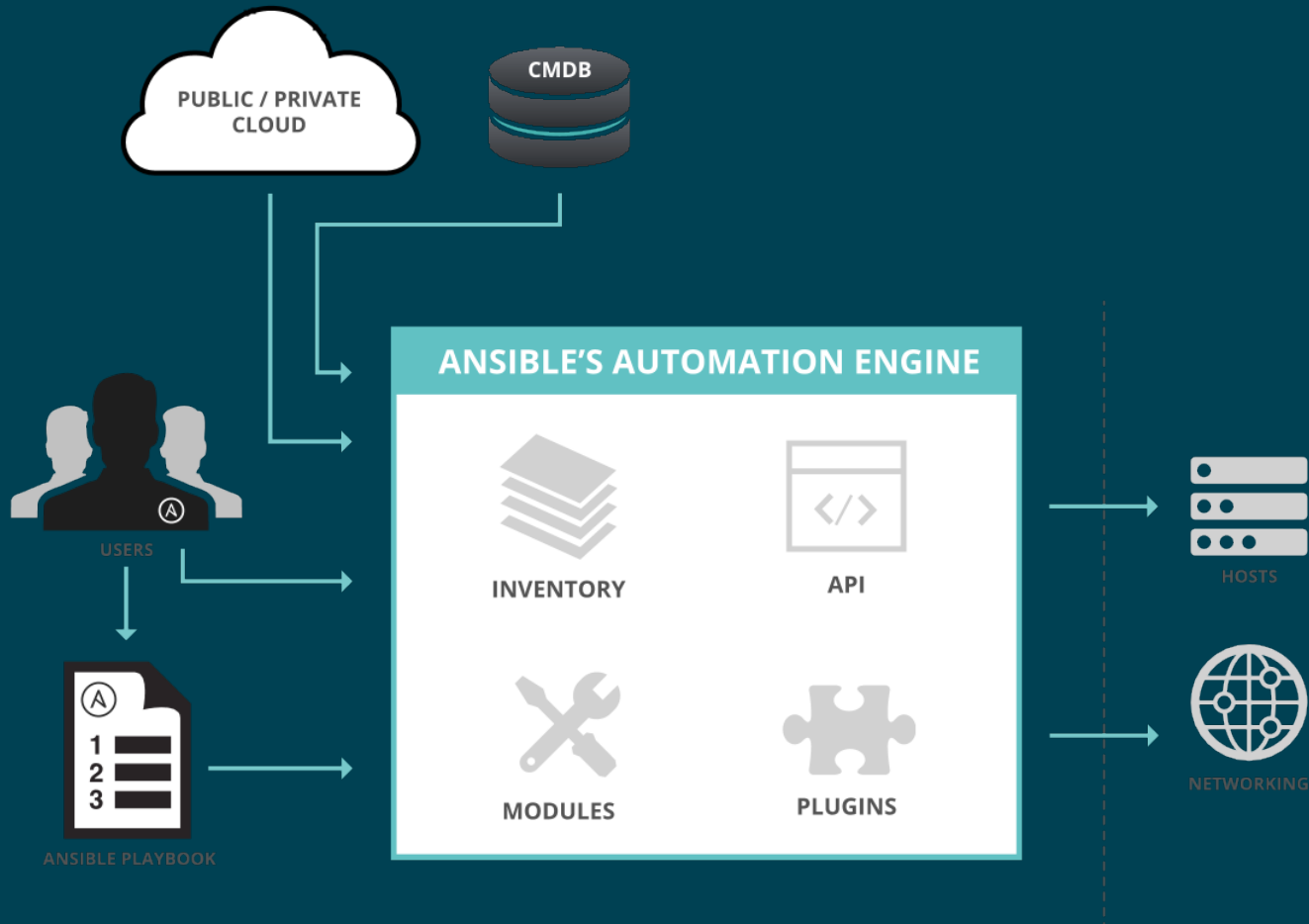
 Watch ▾ 1,889  Star 29,615  Fork 10,855

 [saltstack](#) / [salt](#)

 Watch ▾ 594  Star 8,753  Fork 4,095

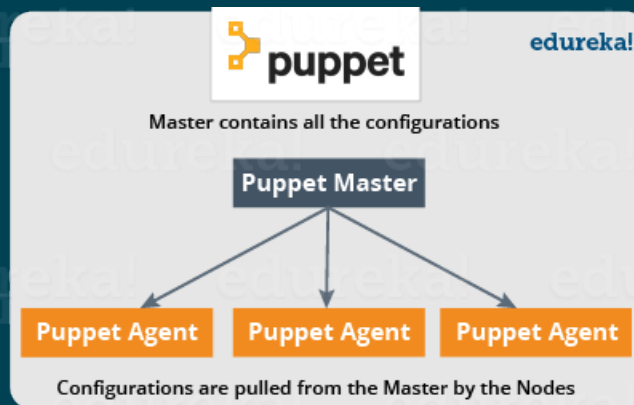
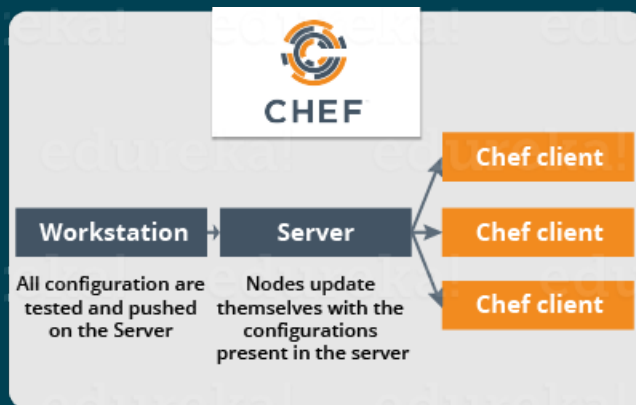
```
$ curl -s https://api.github.com/repos/ansible/ansible | grep  
created_at  
"created_at": "2012-03-06T14:58:02Z"  
$ curl -s https://api.github.com/repos/saltstack/salt | grep  
created_at  
"created_at": "2011-02-20T20:16:56Z"
```


Ansible Architecture



Differences with Puppet

- No master server required
- No agents
- Serial Execution
- Push vs Pull



Example of a Puppet manifest

```
node base_server {
    include "verosk::ssh_keys"
    include "verosk::vimrc"
    include "verosk::packages"
    #
    class { "czechglobe::satellite_postfix": }
    augeas { "sshd": # disable SSH login
        context => "/files/etc/ssh/sshd_config",
        changes => [ 'set PermitRootLogin without-password' ],
    }
    if ($operatingsystem == "CentOS") {
        yumrepo { "czechglobe":
            name => "CzechGlobe internal repository",
            baseurl => "http://[REDACTED]",
            gpgcheck => 0,
        }
    }
}

node virtual_server inherits base_server {

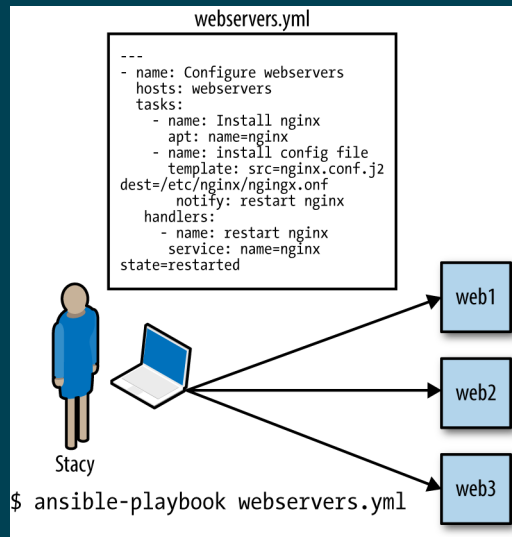
}

node default {
    notify["Default configuration applied":
        withpath=>true,
    ]
    include "verosk::ssh_keys"
    file { "/etc/issue":
        ensure => present,
        content => "$operatingsystem $operatingsystemrelease\n$kernel $kernel
version on $hardwaremodel ($virtual)\nPuppet managed\n\n",
    }
}

"site.pp" 250L, 6899C written                29,1-8                Top
```

All you need is

- A control node
- A target, e.g: servers, network hardware, etc
- Python 2.5+, SSH / winRM.



Ansible Concepts

- Inventory
- Host
- Group
- Plays
- Playbook
- Task
- Module
- Handler
- Plugin

Ansible Concepts

- **Inventory:**
 - Place where targets are defined, it can be static or dynamic.
 - Text files expressed in INI-like format

```
[webservers]
192.168.0.100
192.168.0.101
192.168.0.102

[databases]
mysql-1.mydomain.com
mysql-2.mydomain.com
```

Ansible Concepts

- **Host:**
 - Simply a remote machines that Ansible manages.
 - They can have individual variables assigned to them
 - Can be organized in groups such as: webservers, databases, etc.

Ansible Concepts

- **Group:**
 - A group consists of several hosts assigned to a pool that can be conveniently targeted together.
 - They are given variables that they share in common

Ansible Concepts

- **Play:**
 - A play perform a series of *tasks* on the targets, in the order specified by the play.
 - They are expressed in YAML format in a text file.

```
---  
- hosts: webserver  
  remote_user: root  
  
  tasks:  
    - name: ensure apache is at the latest version  
      yum:  
        name: httpd  
        state: latest
```

Ansible Concepts

- **Playbook:**

A playbook is composed of one or more *plays* in a list, e.g: you may have a playbook that targets first the web servers, and then database servers

```
---
- hosts: webservers
  remote_user: root

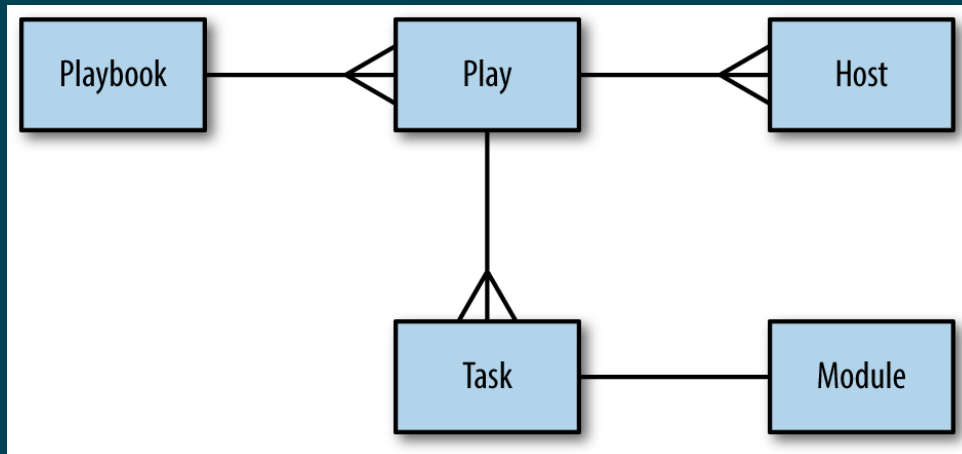
  tasks:
    - name: ensure apache is at the latest version
      yum:
        name: httpd
        state: latest
    - name: write the apache config file
      template:
        src: /srv/httpd.j2
        dest: /etc/httpd.conf

- hosts: databases
  remote_user: root

  tasks:
    - name: ensure postgresql is at the latest version
      yum:
        name: postgresql
        state: latest
    - name: ensure that postgresql is started
      service:
        name: postgresql
        state: started
```

Ansible Concepts

- **Task:**
 - At a basic level, a task is nothing more than a call to an ansible module.
 - Each task is associated with exactly one module



Ansible Concepts

- **Module:**
 - Scripts that come packaged with Ansible and perform some kind of action on a host.
 - Each module is invoked through a task with its arguments
 - More than 200 modules ship with Ansible (and this number grows with every release!).
 - You can find third-party modules or add your own.
 - Some popular modules include: *yum*, *apt*, *copy*, *template*, *lineinfile*, *etc*.

Ansible Concepts

- **Handler:**
 - Similar to a task, but it runs **only** if it has been notified by a task.
 - Handlers usually run after all of the tasks are run at the end of the play
 - If a play contains multiple handlers, the handlers always run in the order they are defined in the handlers section, **not** the notification order.

Ansible Concepts

- **Plugin:**
 - Plugins are pieces of code that augment Ansible's core functionality.
 - Ansible ships with a number of handy plugins, and you can easily write your own.
 - There are several plugins types such as:
 - Lookup
 - Vars
 - Inventory
 - Connection
 - Callback

Questions?



Quiz: Ansible Architecture

1.- Which of the following terms best describes the Ansible architecture?

A) Agentless

B) Client / Server

C) Event-driven

D) Stateless

Quiz: Ansible Architecture

2.- Which network protocol does Ansible use, by default, to communicate with managed nodes?

A) HTTP

B) HTTPS

C) SNMP

D) SSH

Quiz: Ansible Architecture

3.- Which of the following files define the actions Ansible performs on managed nodes?

A) Host inventory

B) Manifest

C) Playbook

D) Script

Quiz: Ansible Architecture

4.- Which syntax is used to define Ansible playbooks?

A) JSON

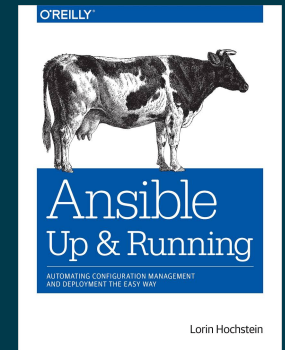
B) Perl

C) Python

D) YAML

References:

- <https://docs.ansible.com>
- Ansible “Up and Running” - Lorin Hochstein 2nd edition



- <https://www.redhat.com/en/services/training/do407-automation-ansible-i>