IT Automation with

Agenda

- Ansible inventory basics
- Static inventory
- Dynamic inventory
- Mixing up inventories

Ansible: Inventory basics

Describing your servers

- The default way to describe your target hosts in Ansible is to list them in text files, called *inventory files.*
- Hosts can be grouped into multiple groups and groups can have child groups.
- By default Ansible will look for its inventory file at */etc/ansible/hosts*, although this can be overridden by using *-i <path>* option on the command line.
- Ansible automatically adds one host to the inventory by default: *localhost*

Ansible: static inventory

Describing your servers

- At its simplest form, an static inventory file is a list of hosts / IP of targets each of them on a single line.
- It has an INI-like format
- Many times hosts are grouped into *hosts groups.*
- Hosts can be in multiple groups, e.g: a server may be a web server and a database server at the same time.
- There are two groups that are present **always**:
  - all: applies to all hosts defined in the inventory (wildcard effect)
  - ungrouped: contains all hosts that aren't member of any host group

Ansible: static inventory

Defining groups of groups

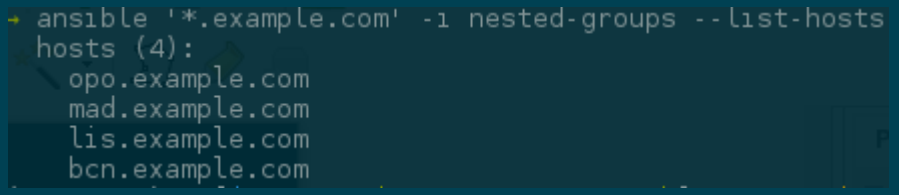
- A group of groups can be defined through the **:children** suffix (see example)
- Numeric and alphabetical ranges are allowed for simplification purposes in the form: **[START:END].** All values from **start** to **end** will be considered (inclusive)

```
[web]
web[1:20].example.com

[databases]
192.168.[0:3].[0.255]

[iot]
2001:db8::[a:f]
```

Ansible: static inventory

Testing hosts inclusion

- To test whether a specific host is being considered by the inventory the following **ansible** command can be invoked:



- Logic operators can be used as well for more complex patterns:
  - “:” character represents OR operator
  - “&” character represents AND operator
  - “!” character represents exclusion character

Ansible: static inventory

Patterns

- An alternative to **all** keyword is to use the '*' wildcard character:

```
[quicklab@master-0 ~]$ ansible mad.example.com --list-hosts -i nested-groups
  hosts (1):
    mad.example.com
[quicklab@master-0 ~]$ ansible vln.example.com --list-hosts -i nested-groups
 [WARNING]: Could not match supplied host pattern, ignoring: vln.example.com

 [WARNING]: No hosts matched, nothing to do

  hosts (0):
```

- Similarly to get the list of all hosts taken into consideration from a specific group:

```
[quicklab@master-0 ~]$ ansible spain --list-hosts -i nested-groups
  hosts (2):
    mad.example.com
    bcn.example.com
```

Ansible: static inventory

Host and groups variables

- It's possible to define variable in a static inventory at host and group level

```
[atlanta]
host1 http_port=80 maxRequestsPerChild=808
host2 http_port=303 maxRequestsPerChild=909
```

- Similarly variables for group scope can be specified by adding **:vars** suffix into a group name:

```
[atlanta]
host1
host2

[atlanta:vars]
ntp_server=ntp.atlanta.example.com
proxy=proxy.atlanta.example.com
```

```
[atlanta]
host1
host2

[raleigh]
host2
host3

[southeast:children]
atlanta
raleigh

[southeast:vars]
some_server=foo.southeast.example.com
halon_system_timeout=30
```

Ansible: static inventory

Inventory variables

- While it's possible to define host and group variables in the inventory file, as inventory gets larger, it gets more difficult to manage variables this way.
- The preferred approach is to create variable files under **host_vars** and **group_vars** directories for host and group variables respectively.

  Example:

  To create a group variable for group *databases* the following procedure will be needed:

```
(17:51:59) ○ [jrosenta@jrosenta.remote.csb] ~/curso
→ mkdir group_vars
(18:51:04) ○ [jrosenta@jrosenta.remote.csb] ~/curso
→ echo 'name: ansible' > group_vars/databases
```

```
→ tree
.
|____host_vars
| |____mysql-2.example.com
| |____mysql-1.example.com
| |____pgsql-1.example.com
|____group_vars
| |____mysql
| |____oracle
| |____postgresql
```

Ansible: static inventory

Behavioral inventory parameters

Name	Default	Description
ansible_host	Name of host	Hostname or IP address to SSH to
ansible_port	22	SSH port
ansible_user	Current logged user	User to SSH as
ansible_connection	smart	Ansible connection method

Note: You can override some of behavioral parameter default values In the *defaults* section of the *ansible.cfg* file.

Ansible: static inventory

- For static inventories besides the INI format, you can also use YAML, e.g:

```
mail.example.com

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

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

```
all:
  hosts:
    mail.example.com:
  children:
    webservers:
      hosts:
        foo.example.com:
        bar.example.com:
    dbservers:
      hosts:
        one.example.com:
        two.example.com:
        three.example.com:
```

```
[atlanta]
host1
host2

[atlanta:vars]
ntp_server=ntp.atlanta.example.com
proxy=proxy.atlanta.example.com
```

```
atlanta:
  hosts:
    host1:
    host2:
  vars:
    ntp_server: ntp.atlanta.example.com
    proxy: proxy.atlanta.example.com
```

Ansible dynamic inventories

Ansible: Dynamic inventory

When to use them?

- There is a considerable amount of hosts and having them in a static file is not scalable.
- Use cases where there is an existent database of hosts, e.g: CMDB, external directory service such as LDAP, Active Directory, etc.
- Scenarios where hosts are highly dynamic such as cloud instances, e.g: AWS EC2, OpenStack, Azure, etc.

Ansible: Dynamic inventory

Basics

- If inventory file is marked as executable, Ansible will assume it is a dynamic inventory script and will execute the file instead of read it.
- **Interface:** It must support two command-line flags:
  - *--host=<hostname>* for showing host details
  - *---list* for listing groups
- The output is a single JSON object.
- A number of dynamic scripts have been added by the open source community to the Ansible Github site at:

  https://github.com/ansible/ansible/tree/devel/contrib/inventory.

Ansible: Dynamic inventory

Basics

- When the dynamic inventory is invoked with **--host hostname** option, it must print a JSON hash/dictionary of the variables belonging to that host (in case there are no variables provided, it return an empty JSON hash).

```
→ ./dynamic-inventory2.sh --host vagrant1
{
  "database_name": "db1"
  "env": "staging"
}
```

- Optionally, if the **--list** option returns a top-level element called **_meta**, it is possible to return all host variables in one script call. In that case **--host** calls are not made.

Ansible: Splitting inventories

Breaking inventory into multiple files

- It's possible to combine regular inventory file and a dynamic inventory script by put them together in the same directory and configure Ansible to use that directory as inventory.
- Example of use case: hybrid cloud!
- The order in which inventory files are parsed is not specified. Most likely it's done in alphabetical order.
- There should not be any dependencies between inventory files or scripts.
- Certain file extensions can be ignored in an directory used as an inventory through the *inventory_ignore_extensions* settings.

Ansible: Adding nodes on runtime

- It's possible to add nodes at runtime with **add_host** module.
- Useful if Ansible is used to provision new host inside an IaaS cloud
- Invoking the module looks like this:

```
add_host name=hostname groups=web,staging myvar=myval
```

- The **add_host** module adds the host only for the duration of the execution of the playbook. It does not modify your inventory file.

Demo

Questions?

Exercises