Ansible Inventory Plugins

Jiri Tyr

About me

- Using Ansible since 2014
- Ansible contributor
  - Modules: `yum_repository`, `jenkins_plugin`, `ldap_attr`, `ldap_entry`
  - Jinja2 filter: `comment`
  - Bug fixing (`mount` module)
  - Code reviews
- Author of more than 100 publicly available Ansible roles
  - https://github.com/jtyr
  - https://galaxy.ansible.com/jtyr

What are inventory plugins?

What are inventory plugins?

- Ansible plugin which takes some input to produce Ansible inventory
- Plugin = Python script
  - Part of the Ansible Core
  - Collection
  - Local file
- Input = string, file or script
  - Defined on the command line via `-i`
- Configured in the ansible.cfg file or via env vars
- Can use several inventory plugins together

How to use inventory plugins?

How to use inventory plugins?

```
# List of default inventory plugins in ansible.cfg
[inventory]
enable_plugins = host_list, script, auto, yaml, ini, toml
```

- Order matters!

How to use inventory plugins?

```
# List of default inventory plugins in ansible.cfg
[inventory]
enable_plugins = host_list, script, auto, yaml, ini, toml
```

- Comma-separated list of hosts (e.g. `-i host1,host2`)

How to use inventory plugins?

```
# List of default inventory plugins in ansible.cfg
[inventory]
enable_plugins = host_list, script, auto, yaml, ini, toml
```

- Comma-separated list of hosts (e.g. **-i host1**,)

How to use inventory plugins?

```
# List of default inventory plugins in ansible.cfg
[inventory]
enable_plugins = host_list, script, auto, yaml, ini, toml
```

- Path to an executable file (e.g. `-i ./inventory.sh`)

How to use inventory plugins?

```
# List of default inventory plugins in ansible.cfg
[inventory]
enable_plugins = host_list, script, auto, yaml, ini, toml
```

- Loads and executes an inventory plugin specified in a YAML inventory plugin config file (e.g. `-i ./config.yaml`)

```
---
plugin: aws_ec2
```

How to use inventory plugins?

```
# List of default inventory plugins in ansible.cfg
[inventory]
enable_plugins = host_list, script, auto, yaml, ini, toml
```

- Path to a file (e.g. **-i ./hosts**)

How does it differ from a dynamic inventory script?

How does it differ from a dynamic inventory script?

Script

Pros
- Any executable file
- Simple JSON output

Cons
- No predefined interface
- No common approach for caching

Plugin

Pros
- Predefined interface
- Common approach for caching

Cons
- Must be written in Python
- Must use `ansible-inventory` to get any output

How to create an inventory plugin?

How to create an inventory plugin?

1. Create `ansible.cfg` file with the following content:

   ```
   [defaults]
   inventory_plugins = plugins/inventory

   [inventory]
   enable_plugins = myplugin
   ```

2. Create inventory input file hosts.my with this content:

   ```
   host1@grp1,host2@grp1,host3@grp2
   ```

3. Store the plugin as `myplugin.py` into the directory `plugins/inventory`
4. Run `ansible-inventory -i ./hosts.my --graph --vars -vvv`

How to create an inventory plugin?

```python
from ansible.plugins.inventory import BaseInventoryPlugin

class InventoryModule(BaseInventoryPlugin):
    NAME = 'myplugin'

    def verify_file(self, path):
        valid = False

        if super(InventoryModule, self).verify_file(path):
            if path.endswith(('.my')):
                valid = True

        return valid

    def parse(self, inventory, loader, path, cache=True):
        super(InventoryModule, self).parse(inventory, loader, path)

        (host_list, _) = self.loader._get_file_contents(path)
        created_groups = []

        for hg in host_list.decode('utf-8').strip().split(','):  # host1@grp1,host2@grp1,host3@grp2
            host, group = hg.split('@')

            self.display.vvv(f"Adding {host} into group {group}")

            if group not in self.created_groups:
                self.inventory.add_group(group)
                self.created_groups.append(group)

            self.inventory.add_host(host, group=group)
            self.inventory.set_variable(host, 'pair', hg)
```

How to create an inventory plugin?

```
@all:
  |--@grp1:
  |   |--host1
  |   |   |--{pair = host1@grp1}
  |   |--host2
  |   |   |--{pair = host2@grp1}
  |--@grp2:
  |   |--host3
  |   |   |--{pair = host3@grp2}
  |--@ungrouped:
```

Thank you for your attention!

Questions?