

18/03/2019 Plugins & Dynamic inventories

PLUGINS & Dynamic inventories

- Ansible Introduction
- Ansible Concepts
- Ansible Playbooks
- ➔ Plugins & Dynamic inventories
- Playbooks Advanced
- Ansible Roles
- AWX / Ansible Tower Overview

©2019 edc-4it B.V. 1

(c) 2019 edc-4it B.V.

18/03/2019 Plugins & Dynamic inventories

dynamic inventories

©2019 edc-4it B.V. 2.1

(c) 2019 edc-4it B.V.

18/03/2019 Plugins & Dynamic inventories

THREE ways TO DYNAMICALLY add HOSTS/GROUPS

- dynamic inventories
- inventory plugins (since 2.4)
- the `add_host` module

inventory plugins are explained later

©2019 edc-4it B.V. 2.2

(c) 2019 edc-4it B.V.

18/03/2019 Plugins & Dynamic inventories

dynamic inventories

- Cobbler
- AWS EC2
- Google Compute Engine
- OpenShift
- OpenStack
- Docker
- ...

See [Odevel/contrib/inventory](#)

©2019 edc-4it B.V. 2.3

(c) 2019 edc-4it B.V.

18/03/2019 Plugins & Dynamic inventories

dynamic inventory EXAMPLE

```
$ ansible running -i docker.py --list-hosts
festive_mculty
goofy_brown
...
```

Uses [docker.py](#)

©2019 edc-4it B.V. 2.4

(c) 2019 edc-4it B.V.

18/03/2019 Plugins & Dynamic inventories

dynamic inventories

PYTHON YIELDING JSON

argument	description
<code>--list</code>	return groups/hosts/group variables
<code>--host «name»</code>	host variables

©2019 edc-4it B.V. 2.5

(c) 2019 edc-4it B.V.

18/03/2019

Plugins & Dynamic Inventories

--list OPTION response

```
{
  "dbservers": {
    "hosts": ["psql-1.example.com", " " "psql-2.example.com"],
    "vars": {
      "ansible_ssh_user": "thomas"
    },
  }
}
```

can also have a **children** key

2.6

(c) 2019 edc-4it B.V.

18/03/2019

Plugins & Dynamic Inventories

--host «name» OPTION response

```
{
  "conf_location": "/var/app.conf",
  "ansible_user": "tiger",
}
```

calling **--host «name»** for each host might be expensive, use **_meta** instead (next)

2.7

(c) 2019 edc-4it B.V.

18/03/2019

Plugins & Dynamic Inventories

Meta Data OPTIMISATION

```
{
  "dbservers": { ... },
  "_meta": {
    "hostvars": {
      "psql-1.example.com": {
        "conf_location": "/var/app.conf",
        "ansible_user": "tiger"
      }
    }
  }
}
```

Introduced in Ansible 1.3

2.8

(c) 2019 edc-4it B.V.

18/03/2019

Plugins & Dynamic Inventories

add_host MODULE

```
- name: add new host
  add_host:
    hostname: "{{ ip }}"
```

Can also specify different connection plugins (explained later)

2.9

(c) 2019 edc-4it B.V.

18/03/2019

Plugins & Dynamic Inventories

add_host MODULE

```
- name: add new host
  add_host:
    hostname: "{{ ip }}"
    groups:
      - group1
      - group2
```

Can also specify different connection plugins (explained later)

2.9

(c) 2019 edc-4it B.V.

18/03/2019

Plugins & Dynamic Inventories

add_host MODULE

```
- name: add new host
  add_host:
    hostname: "{{ ip }}"
    groups:
      - group1
      - group2
    ansible_user: thomas
    app_config_dir: /etc/my-app.config
```

Can also specify different connection plugins (explained later)

2.9

(c) 2019 edc-4it B.V.

18/03/2019

Plugins & Dynamic Inventories

LAB 5

dynamic inventories

3

(c) 2019 edc-4it B.V.

18/03/2019

Plugins & Dynamic Inventories

PLUGINS overview

4.1

(c) 2019 edc-4it B.V.

18/03/2019

Plugins & Dynamic Inventories

KEY PLUGINS TYPES

- Callback plugins
- Cache plugins
- Connection plugins
- Inventory plugins
- ...

Other types: action, lookup, shell, strategy, vars (some are explained later)

4.2

(c) 2019 edc-4it B.V.

18/03/2019

Plugins & Dynamic Inventories

GETTING HELP

```
$ ansible-doc --list --type callback
```

EXAMPLE

```
$ ansible-doc -t callback json
```

4.3

(c) 2019 edc-4it B.V.

18/03/2019

Plugins & Dynamic Inventories

CONFIGURING THE PLUGIN

USING ansible.cfg

```
[callback_slack]  
channel = #pipeline  
webhook_url = ...
```

USING environment variable

```
export SLACK_WEBHOOK_URL=...
```

Consult the plugin documentation for options/sections/etc

4.4

(c) 2019 edc-4it B.V.

18/03/2019

Plugins & Dynamic Inventories

ENABLING PLUGINS

USING ansible.cfg

```
[defaults]  
fact_caching=redis
```

USING environment variable

```
export ANSIBLE_CACHE_PLUGIN=jsonfile
```

Consult the plugin documentation for options/sections/etc

4.5

(c) 2019 edc-4it B.V.

18/03/2019 Plugins & Dynamic Inventories

CALLBACK PLUGINS

5.1

(c) 2019 edc-4it B.V.

18/03/2019 Plugins & Dynamic Inventories

RESPOND TO EVENTS

5.2

(c) 2019 edc-4it B.V.

18/03/2019 Plugins & Dynamic Inventories

TWO TYPES

- stdout
- other (notify & aggregate)

5.3

(c) 2019 edc-4it B.V.

18/03/2019 Plugins & Dynamic Inventories

KEY CALLBACK PLUGINS

- default screen output
- yaml
- jabber
- logstash
- ...

See list at docs.ansible.com

5.4

(c) 2019 edc-4it B.V.

18/03/2019 Plugins & Dynamic Inventories

ENABLING (NOTIFY/AGGREGATE) CALLBACKS

```
# example using ansible.cfg
callback_whitelist = timer, yaml, mail
```

Or use `ANSIBLE_CALLBACK_WHITELIST`

Most require configuration

5.5

(c) 2019 edc-4it B.V.

18/03/2019 Plugins & Dynamic Inventories

ENABLING USING DIRECTORIES

- `~/.ansible/plugins/callback`
- `/usr/share/ansible/plugins/callback`
- `callback_plugins/` adjacent to play/role

Configurable using `callback_plugins/ANSIBLE_CALLBACK_PLUGINS`

5.6

(c) 2019 edc-4it B.V.

18/03/2019 Plugins & Dynamic Inventories

cache PLUGINS

6.1

(c) 2019 edc-4it B.V.

18/03/2019 Plugins & Dynamic Inventories

FACT CACHING

6.2

(c) 2019 edc-4it B.V.

18/03/2019 Plugins & Dynamic Inventories

KEY CACHING PLUGINS

- memory
- jsonfile
- memcached
- mongodb
- pickle
- redis
- yaml

6.3

(c) 2019 edc-4it B.V.

18/03/2019 Plugins & Dynamic Inventories

ENABLING

```
gathering=smart
fact_caching=jsonfile
```

6.4

(c) 2019 edc-4it B.V.

18/03/2019 Plugins & Dynamic Inventories

CACHE TIMEOUT

```
fact_caching_timeout=7200 # 2 hours
```

Default is 86400 (24 hours)

6.5

(c) 2019 edc-4it B.V.

18/03/2019 Plugins & Dynamic Inventories

EXPLICITLY FLUSHING cache

```
$ ansible-playbook mybook.yaml --flush-cache
```

6.6

(c) 2019 edc-4it B.V.

18/03/2019 Plugins & Dynamic Inventories

connection PLUGINS

7.1

(c) 2019 edc-4it B.V.

18/03/2019 Plugins & Dynamic Inventories

connect to HOSTS to run THE PLAY

Can be set per host

7.2

(c) 2019 edc-4it B.V.

18/03/2019 Plugins & Dynamic Inventories

key connection PLUGINS

- docker
- kubectl
- oc
- winrm
- ...

7.3

(c) 2019 edc-4it B.V.

18/03/2019 Plugins & Dynamic Inventories

EXAMPLE connection

```
httpbin
[all:vars]
ansible_connection=kubectl
kubectl_namespace=default
kubectl_cert_file=$HOME/.minikube/client.crt
kubectl_key_file=$HOME/.minikube/client.key
kubectl_ssl_ca_cert=$HOME/.minikube/ca.crt
kubectl_host=192.168.99.100
ansible_python_interpreter=/usr/bin/python3
```

7.4

(c) 2019 edc-4it B.V.

18/03/2019 Plugins & Dynamic Inventories

inventory PLUGINS

8.1

(c) 2019 edc-4it B.V.

18/03/2019 Plugins & Dynamic Inventories

COMPILE inventory Based on data sources

Since Ansible 2.4

8.2

(c) 2019 edc-4it B.V.

18/03/2019Plugins & Dynamic Inventories

NEED TO BE WHITELISTED

```
[inventory]
enable_plugins = host_list, script, yaml, ini, auto
```

Are tried in order

However, `auto` automatically loads a plugin

8.3

(c) 2019 edc-4it B.V.

18/03/2019Plugins & Dynamic Inventories

KEY INVENTORY PLUGINS

- `aws_ec2`
- `gcp_compute`
- `openshift`
- `openstack`
- `k8s`
- ...

8.4

(c) 2019 edc-4it B.V.

18/03/2019Plugins & Dynamic Inventories

EXAMPLE (K8S)

```
plugin: k8s
connections:
- host: https://192.168.99.100:8443
  cert_file: /home/vagrant/.minikube/client.crt
  key_file: /home/vagrant/.minikube/client.key
  ssl_ca_cert: /home/vagrant/.minikube/ca.crt
  namespaces: [default]

$ ansible-inventory -i kube.yaml --graph
@all:
  |--@192-168-99-100_8443:
  |  |--@namespace_default:
  |  |  |--@namespace_default_pods:
  |  |  |  |--httpbin_httpbin
  |  |  |  |--...
  |  |  |--...
  |  |--...
  |--...
```

8.5

(c) 2019 edc-4it B.V.

18/03/2019Plugins & Dynamic Inventories

LAB 6
PLUGINS

9

(c) 2019 edc-4it B.V.

18/03/2019Plugins & Dynamic Inventories

DONE!

- Ansible Introduction
- Ansible Concepts
- Ansible Playbooks
- Plugins & Dynamic Inventories
- Playbooks Advanced
- Ansible Roles
- AWX / Ansible Tower Overview

10

(c) 2019 edc-4it B.V.