Ansible Basics

Lecture 2
Basic knowledge to set up an Ansible environment

Ansible Basics

Content

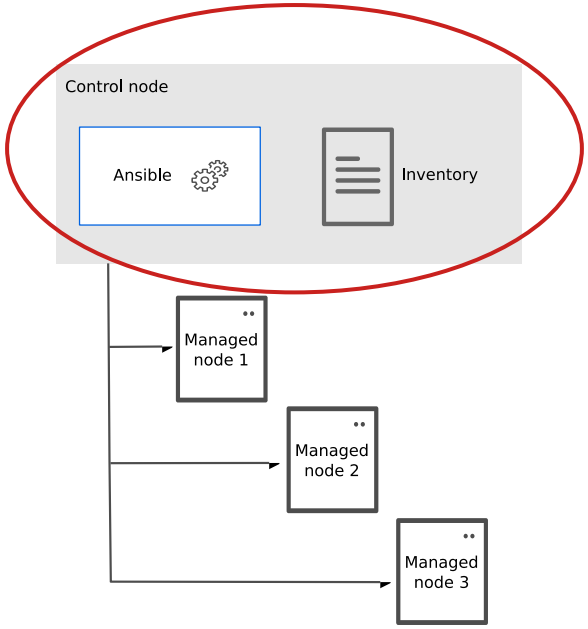
- Basics within Ansible
  - Structure for ansible inventory files
  - Inventory
  - Variables
  - Templates
  - Connections
  - Vault

Ansible Basics

- Concepts for this lecture
  - Inventory
  - Vault
  - SSH
  - Jinja2
  - Yaml-format
  - Ini-format
  - Host (client)
  - State
  - Source control
  - CMDB
  - Template

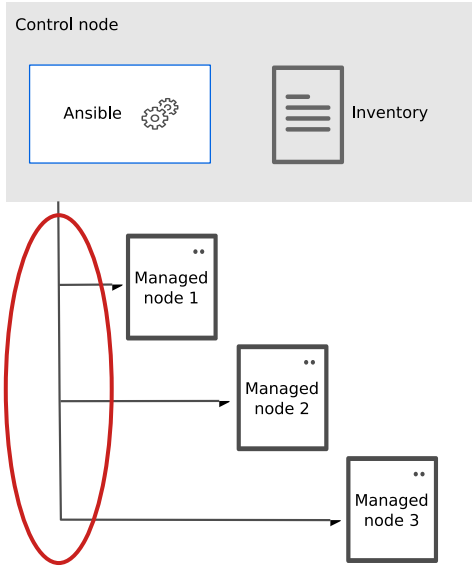
Ansible Basics

- Ansible starts from a central node, a control node which then sets a "state" on the clients it controls.



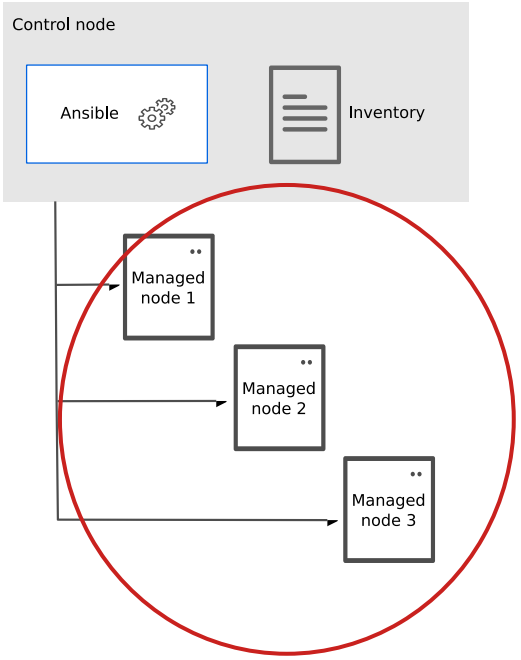
Ansible Basics

- The control node uses SSH as transportation to connect to the clients and set the “state”.



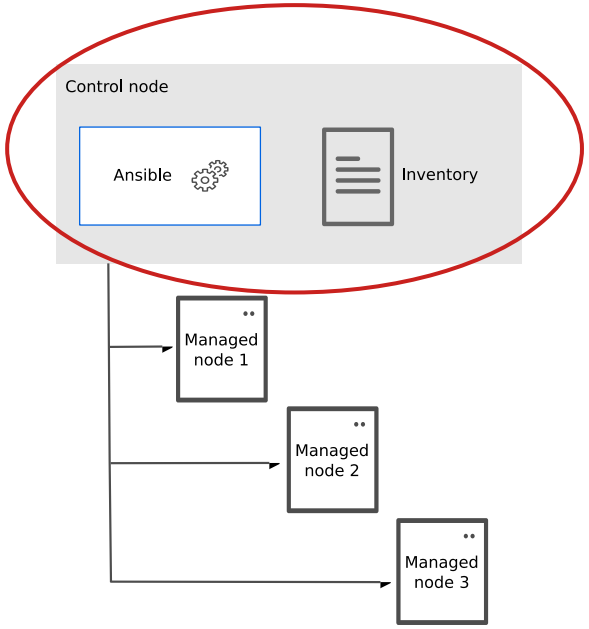
Ansible Basics

- The clients have no client software. Everything needed is sent along from the control node
- Modules, scripts, files etc



Ansible Basics

- The control node is the node that keeps all the configuration, modules, files, scripts, etc. that are needed to define the "state" that is desired on the clients



Ansible Basics

- Inventory as a static file
  - Text file containing one client per line
    - Most common is the ini format
    - Yaml format is also possible to use

Read up: https://docs.ansible.com/ansible/latest/user_guide/intro_inventory.html

Ansible Basics

- Inventory
  - Example ini-format (most common)

```
mail.example.com

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

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

[dnsservers]
one.example.com
```

Read up: https://docs.ansible.com/ansible/latest/user_guide/intro_inventory.html

Ansible Basics

- Inventory
  - Example of Yaml-format

```
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:
    dnsservers:
      hosts:
        one.example.com
```

Read up: https://docs.ansible.com/ansible/latest/user_guide/intro_inventory.html

Ansible Basics

- Inventory
  - If DNS is not available, it is possible to specify ip addresses and ports in the inventory

```
INI format:
host1 ansible_port=5555 ansible_host=192.0.2.50

YAML format:
...
  hosts:
    host1:
      ansible_port: 5555
      ansible_host: 192.0.2.50
```

Read up: https://docs.ansible.com/ansible/latest/user_guide/intro_inventory.html

Ansible Basics

- Inventory
  - Can use modules to use inventory from CMDB dynamically. But not a must
    - For ex. VmWare, VirtualBox, or a script of your own (outside the scope of this course)
    - Cloud is also available with AWS, Azure, Google, etc

Read up: https://docs.ansible.com/ansible/latest/user_guide/intro_dynamic_inventory.html

Ansible Basics

- Variables
  - Host variables
  - Group variables

Ansible Basics

- Host variables in inventory file

```
[atlanta]

host1 http_port=80 maxRequestsPerChild=808
host2 http_port=303 maxRequestsPerChild=909
```

Read up: https://docs.ansible.com/ansible/latest/user_guide/intro_inventory.html#host-variables

Ansible Basics

- Host variables in variable file
  - Presumptions
    - **If** *ansible/hosts* is the location where the inventory file is located (other locations are ok)
    - The client (host) has the name 'foosball' (example, other names are OK)
    - **Then** the variables are stored in the file *ansible/host_vars/foosball* (must be like this, other names *not OK* for these conditions, but file endings in yaml, yml, or json is)

Read up: https://docs.ansible.com/ansible/latest/user_guide/intro_inventory.html#organizing-host-and-group-variables

Ansible Basics

- Host variables in variable file

```
---
ntp_server: acme.example.org
database_server: storage.example.org
```

Read up: https://docs.ansible.com/ansible/latest/user_guide/intro_inventory.html#organizing-host-and-group-variables

Ansible Basics

- Group variables in inventory file
  - Ini format

```
[atlanta]
host1
host2

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

Read up: https://docs.ansible.com/ansible/latest/user_guide/intro_inventory.html#group-variables

Ansible Basics

- Group variables in inventory file
  - Yaml format

```
atlanta:
  hosts:
    host1:
    host2:

  vars:
    ntp_server: ntp.atlanta.example.com
    proxy: proxy.atlanta.example.com
```

Read up: https://docs.ansible.com/ansible/latest/user_guide/intro_inventory.html#group-variables

Ansible Basics

- Group variables in variable file
  - Presumptions
    - **If** ansible/hosts is the location where the inventory file is located (other locations allowed)
    - The groups are named 'raleigh' and 'webservers' (example, other names allowed)
    - **Then** the variables are stored in the files (Must be like this for the given example, but file endings can be yaml, yml, or json)
      - ansible/group_vars/raleigh
      - ansible/group_vars/webservers

Read up: https://docs.ansible.com/ansible/latest/user_guide/intro_inventory.html#organizing-host-and-group-variables

Ansible Basics

- Group variables in variable file

```
---
ntp_server: acme.example.org
database_server: storage.example.org
```

Read up: https://docs.ansible.com/ansible/latest/user_guide/intro_inventory.html#organizing-host-and-group-variables

Ansible Basics

- Use variables
  - To use variables in playbooks the Jinja2 format is used
  - https://jinja.palletsprojects.com/en/3.1.x/
  - https://jinja.palletsprojects.com/en/3.1.x/templates/

Ansible Basics

- Use variables
  - In playbooks the syntax {{ variabel_name }} is used
  - Can also be used for e.g. templating files
  - Ansible also sets some of its own variables, such as information about the host and the other hosts Ansible knows about, these are generally called “facts”

Ansible Basics

Ansible Vault

- Ansible's way of encrypting sensitive information
- Can encrypt entire files, which can then be versioned in source control
- It is also possible to encrypt strings in e.g. a playbook
  - https://docs.ansible.com/ansible/latest/user_guide/vault.html
  - https://docs.ansible.com/ansible/latest/cli/ansible-vault.html

Ansible Basics

Ansible Vault

- Example

```
my_encrypted_var: !vault |
        $ANSIBLE_VAULT;1.2;AES256;dev
        30613233633461343837653833666333643061636561303338373661313838333565653635353162
        32633634346237333435386534626130643336343334646 60a6636336239393934393166366338 63
        61636237636537333938306331383339353265363239643939666639386530626330633337633833
        66646563343731666 30a3637363932626664656634326139326130363039633432636231373862 39
        6330
```

Ansible Basics

- To decrypt the information, either a flag is used when running the ansible-playbook command to specify the password
- Or, a file containing the decryption password is specified when running the ansible-playbook command.

Ansible Basics

- ATTENTION! For security reasons, it is extremely important to:
  - The file with the vault password has the read and write permissions set to only your own user in the filesystem
  - That the file containing the vault password is NOT versioned or shared with others

Ansible Basics

- End of todays lecture!
  Over to practical tasks