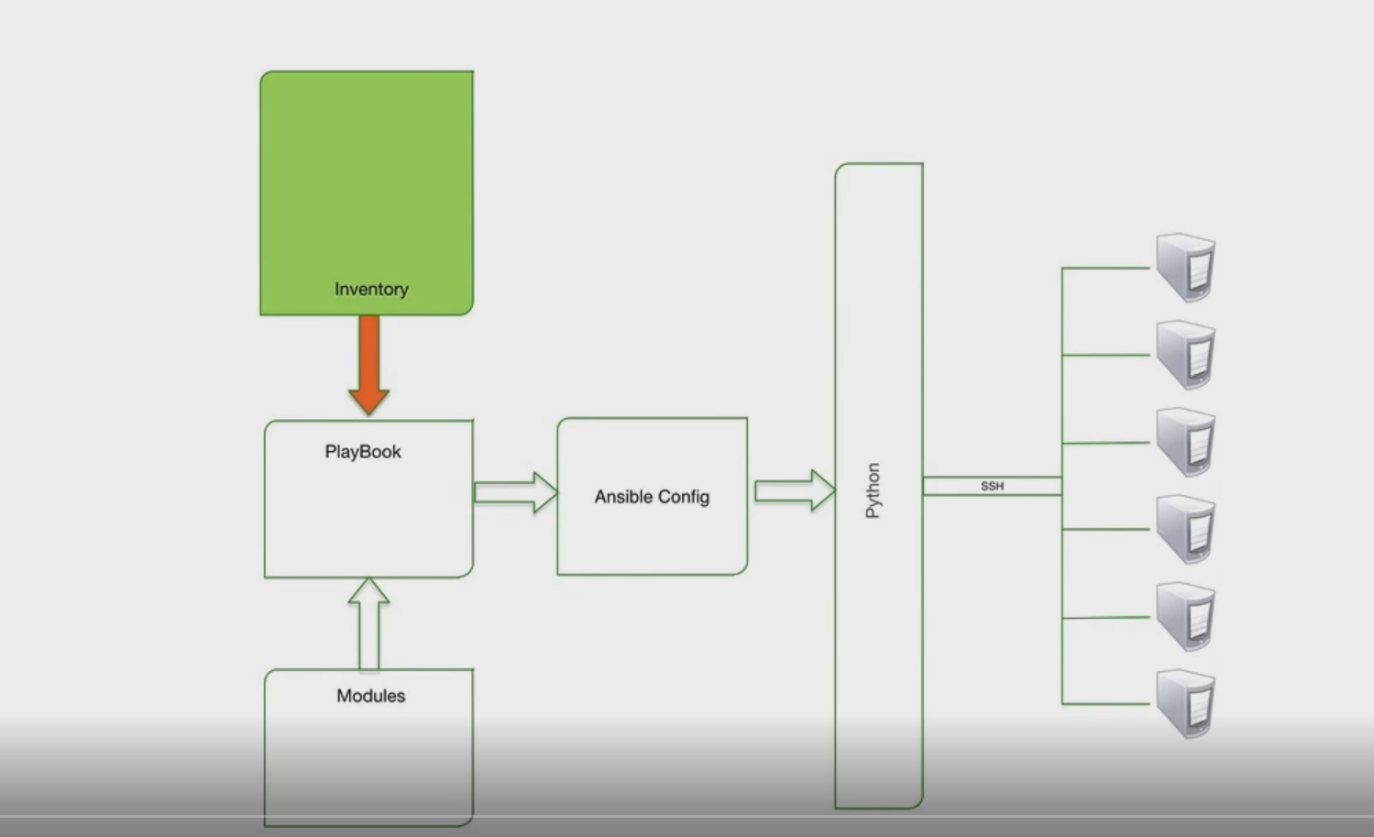
**Hands-on Ansible**

<https://www.pluralsight.com/courses/hands-on-ansible>

Ansible Inventory and Configuration

* Inventories is what feeds your play
* Ansible need this lookup process to be able to learn about the system it will be talking to.
* It can be located anywhere in the path as you long as you specify it when you call your play



What we can put in an inventory file:

* We can apply these to a system or a group of systems:

**Behavioural params**: SSH users, passwords, private keys, paths to python 2.6 interpreter on python 3 based systems.

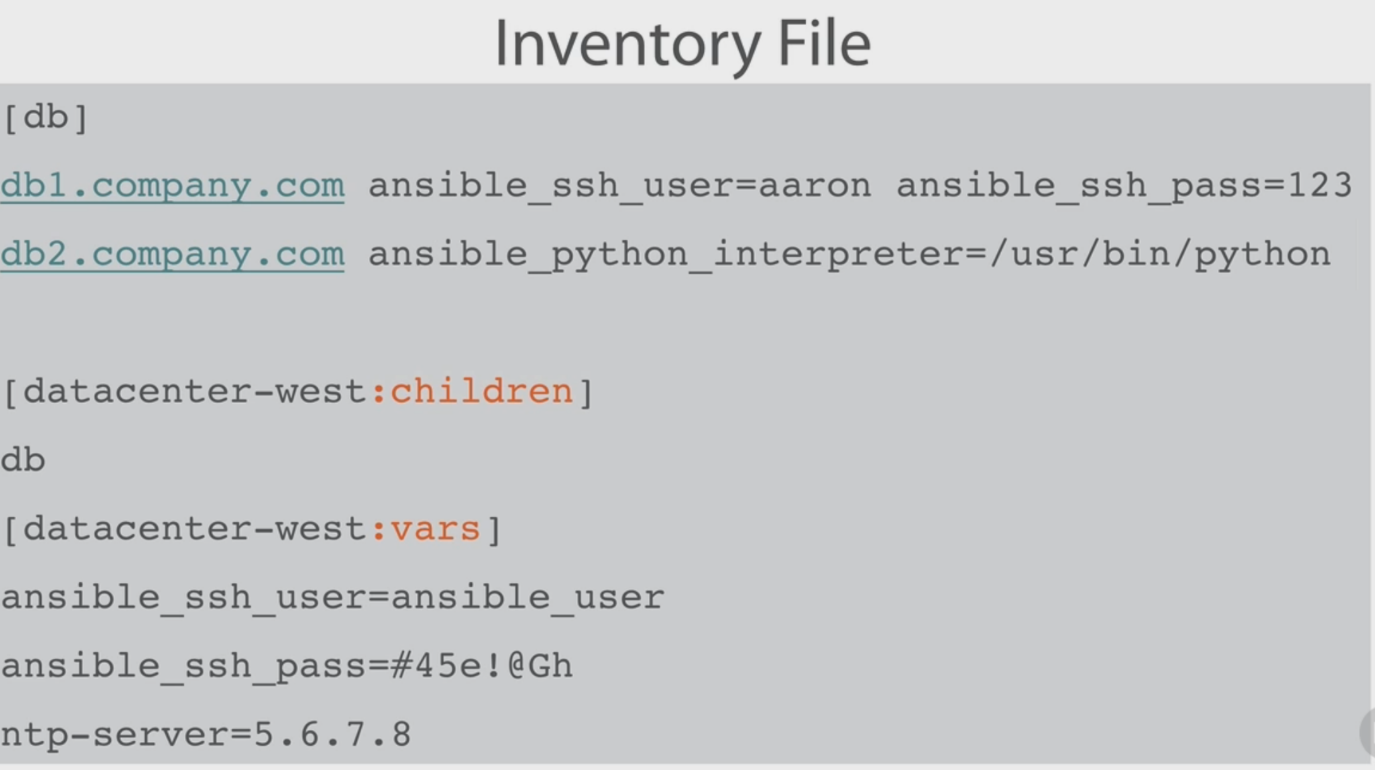
**Groups**: Test DB servers, production servers.

**Groups of groups**: Data centers West, Data centers East

**Variables**: NTP\_SERVER, SYS\_LOG\_SERVER and apply it to an entire datacentre.

**Scaling out using multiple files**

**Inventory file can be static or dynamic with scripts.**

* **Example of a file**

**Example:**

Create a file inventory:

*File inventory*

**web1** **ansible\_ssh\_host**=192.168.33.20 **ansible\_ssh\_user**=vagrant **ansible\_ssh\_pass**=vagrant

**vagrant@acs:~/exercice2$ ansible web1 -i inventory -m ping**

**# Success**

* Username and password in the inventory are behavioural variables

Let’s create a group:

*File inventory*

**web1** **ansible\_ssh\_host**=192.168.33.20 **ansible\_ssh\_user**=vagrant **ansible\_ssh\_pass**=vagrant

[**webservers**]

vagrant@acs:~/exercice2$ ansible **webservers** -i inventory -m ping

**# Success**

* Let’s add the DB server

**web1 ansible\_ssh\_host=192.168.33.20 ansible\_ssh\_user=vagrant ansible\_ssh\_pass=vagrant**

**db1 ansible\_ssh\_host=192.168.33.30 ansible\_ssh\_user=vagrant ansible\_ssh\_pass=vagrant**

**[webservers]**

**web1**

**db1**

Let’s run the command again

vagrant@acs:~/exercice2$ ansible **webservers** -i inventory -m ping



Let’s create a parent group called datacenter and add our two servers groups

**web1 ansible\_ssh\_host=192.168.33.20 ansible\_ssh\_user=vagrant ansible\_ssh\_pass=vagrant**

**db1 ansible\_ssh\_host=192.168.33.30 ansible\_ssh\_user=vagrant ansible\_ssh\_pass=vagrant**

**[webservers]**

**web1**

**[dbservers]**

**db1**

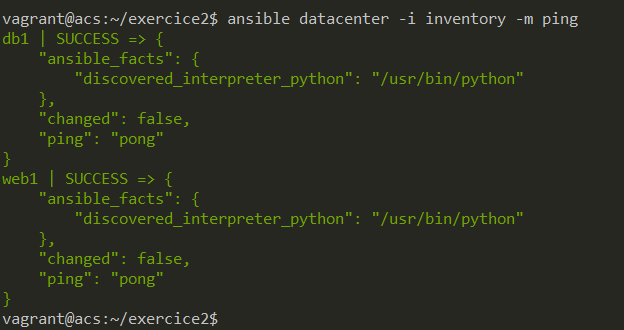
**[datacenter:children]**

**webservers**

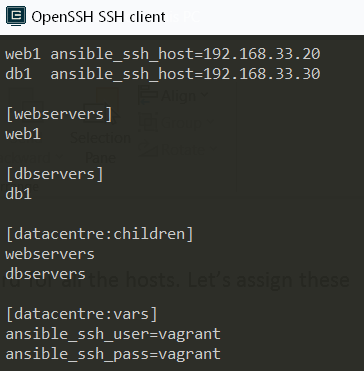
**dbservers**

Let’s run the command again

vagrant@acs:~/exercice2$ ansible **datacentre** -i inventory -m ping



In order to avoid writing user and password for all the hosts. Let’s assign these variables to groups



Let’s run the command again

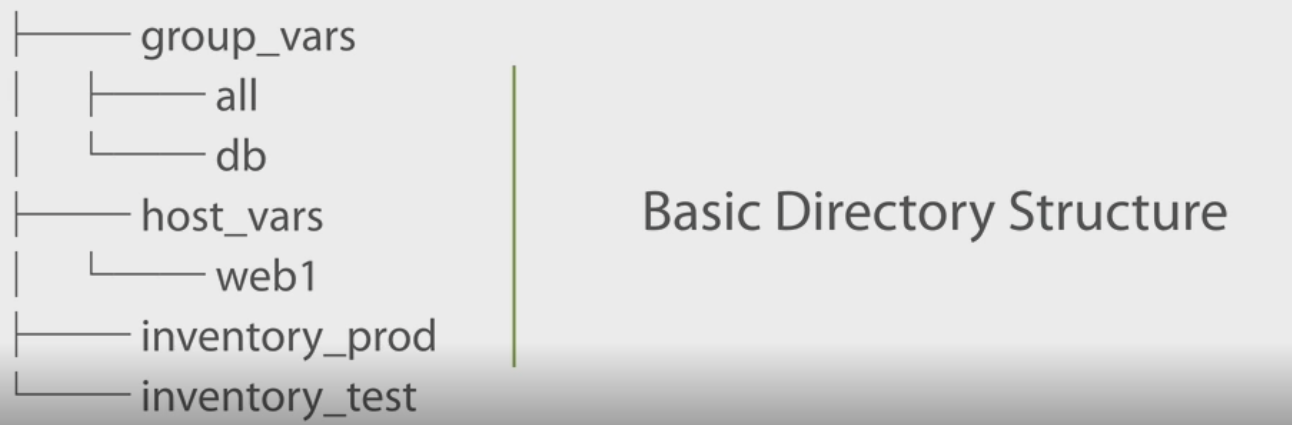
vagrant@acs:~/exercice2$ ansible **datacentre** -i inventory -m ping



Scaling-out the inventory file

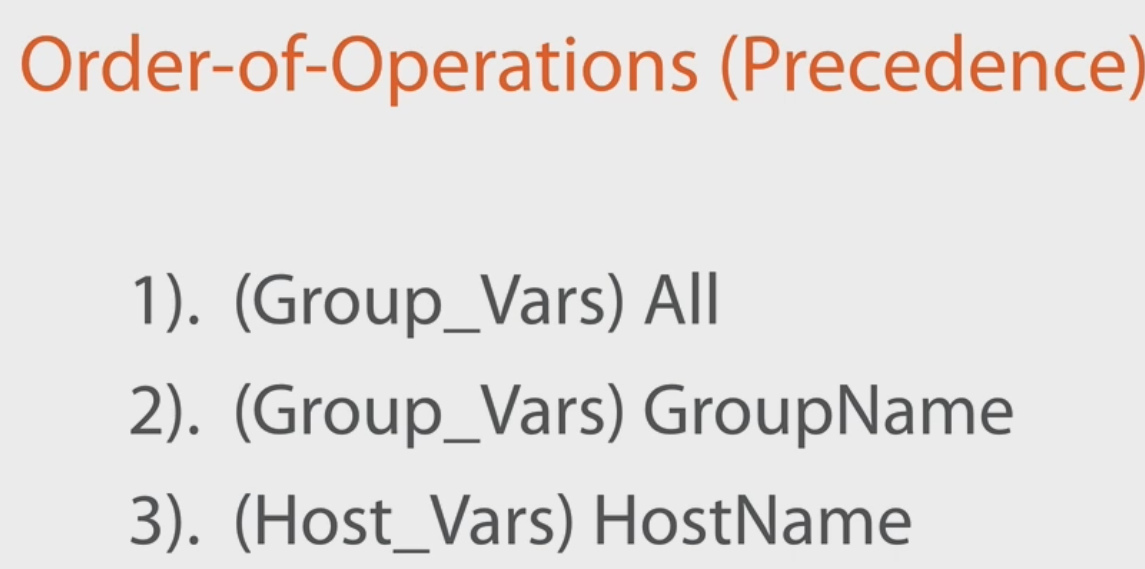


One way to organise folders

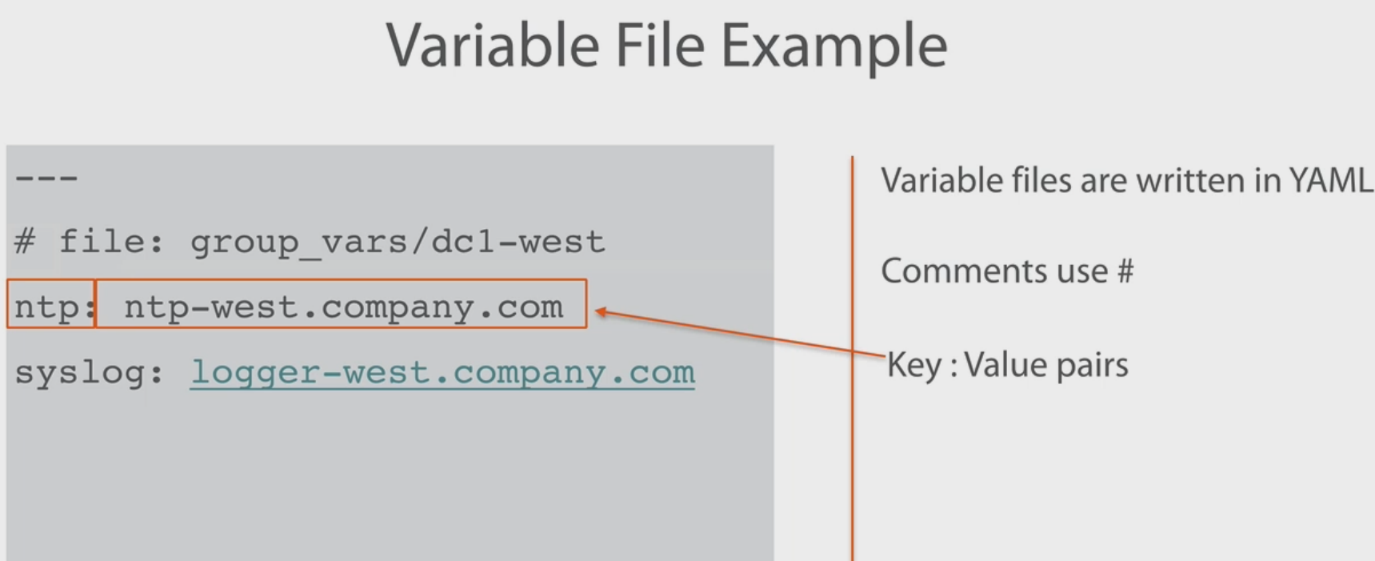


Another way



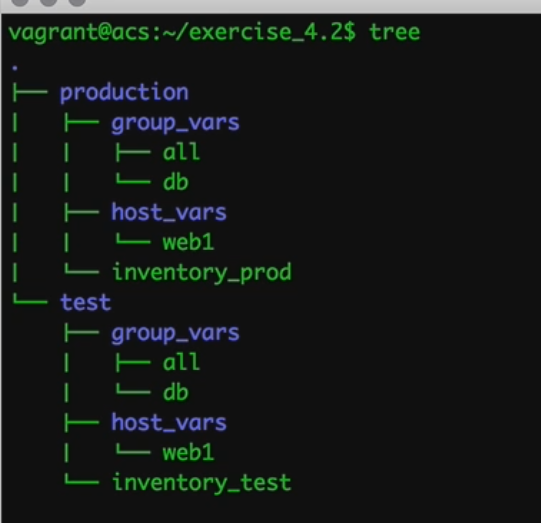


Example of a YAML variable file

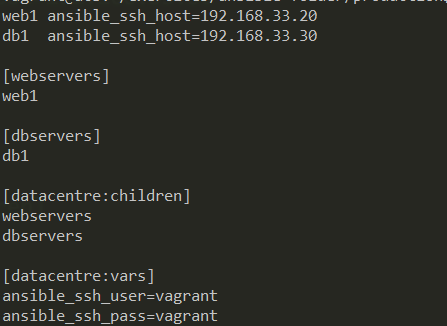


Lab

Let’s take this structure



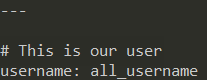
Inventory\_prod file



**‘’all’’ file:**

group\_vars

all



Command:

>> ansible webservers -i inventory\_prod -**m user** -a "name={{username}} password=12345"

* This is using a module user to add users to all the groups under the inventory file.