LESSON 3

site.yml

```
[web]
web1 ansible_host=192.168.56.11
$ ansible web -m ping
```

YAML

Inventory & Ad-Hoc

Learn it all @MyWebUniversity



A FREE MYWEBUNIVERSITY COURSE

ANSIBLE & DEVOPS

LESSON 3

Inventory and Ad-Hoc Commands

INI and YAML inventories, groups, host and group variables, patterns and ad-hoc comma

An INI Inventory

inventory.ini

```
[web]
web1 ansible_host=192.168.56.11
web2 ansible_host=192.168.56.12

[db]
db1 ansible_host=192.168.56.13

[production:children]
web
db
```

- The inventory lists your hosts and puts them into groups in square brackets. A group of …
- Every host is automatically in the group named all.

01

A YAML Inventory

inventory.yml

```yaml
all:
  children:
    web:
      hosts:
        web1: { ansible_host: 192.168.56.11 }
        web2: { ansible_host: 192.168.56.12 }
    db:
      hosts:
        db1:  { ansible_host: 192.168.56.13 }
```

- The same inventory can be written in YAML, which is easier for large and nested structures.
- ansible-inventory with graph prints the tree, so you can check your groups.

02

host_vars and group_vars

terminal

```
$ tree
.
├── ansible.cfg
├── inventory.ini
├── group_vars/
│   ├── all.yml        # ntp_server: 192.168.56.1
│   └── web.yml        # http_port: 80
└── host_vars/
    └── web2.yml       # http_port: 8080
```

- Keep variables out of the inventory in folders named group_vars and host_vars, one YAML …
- Ansible loads them automatically, so all web servers can share an http_port while one host …

03

Host Patterns

Pattern	Targets
`all`	every host
`web`	the web group
`web:db`	web OR db
`production:&web`	in both groups
`all:!db1`	everything except db1
`web[0]`	first web host

- Patterns choose which hosts a command targets: a host, a group, all, or combinations. Colon …
- dash dash limit narrows any run further, which is a safe habit.

04

Ad-Hoc Commands

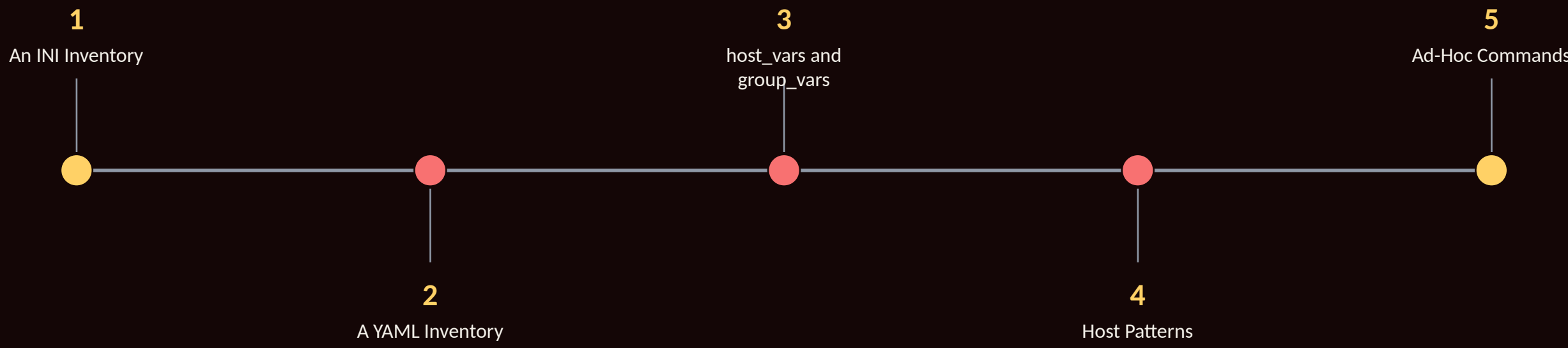
terminal

```
$ ansible web -m command -a "uptime"
web1 | CHANGED | rc=0 >> 10:42 up 3 days, load average: 0.05
$ ansible db -b -m package -a "name=htop state=present"
$ ansible all -m setup -a "filter=ansible_distribution*"
$ ansible web -b -m service -a "name=nginx state=restarted"
```

- An ad-hoc command runs one module on many hosts without writing a playbook. Use dash m for …
- Great for quick checks; anything you repeat belongs in a playbook.

05

Lesson Map


1
An INI Inventory
2
A YAML Inventory
3
host_vars and group_vars
4
Host Patterns
5
Ad-Hoc Commands

Sources and Further Reading

inventory.ini

```
[web]
web1 ansible_host=192.168.56.11
web2 ansible_host=192.168.56.12

[db]
db1 ansible_host=192.168.56.13

[production:children]
web
db
```

ANSIBLE & DEVOPS · 3

terminal

```
$ tree
.
├── ansible.cfg
├── inventory.ini
├── group_vars/
│   ├── all.yml        # ntp_server: 192.168.56.1
│   └── web.yml        # http_port: 80
└── host_vars/
    └── web2.yml       # http_port: 8080
```

Thank you for watching

MyWebUniversity.com

Free technical education since 2000 — English and Dari

@MyWebUniversity · @MyWebUniversity-Dari