

**DT/NT** : DT

**LESSON** : DevOps

**SUBJECT:** Ansible 3

Vault

Facts

Dynamic inventory

**BATCH: 149**

**05/10/2023**



**TECHPRO**  
EDUCATION



techproeducation.com



+1 (585) 304 29 59



- `ANSIBLE_CONFIG` (environment variable if set)
- `ansible.cfg` (in the current directory)
- `~/.ansible.cfg` (in the home directory)
- `/etc/ansible/ansible.cfg`

`ansible.cfg`

```
ansible4 > ansible.cfg
1 [defaults]
2 host_key_checking = False
3 inventory = inventory.txt
4 deprecation_warnings=False
5 interpreter_python=auto_silent
6
```

`vim hosts`

inventory

```
[webservers]
node1 ansible_host=<node1_ip> ansible_user=ec2-user
node2 ansible_host=<node2_ip> ansible_user=ec2-user

[all:vars]
ansible_ssh_private_key_file=/home/ec2-user/<pem file>

[devservers]
node1

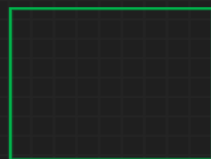
[testgroup:children]
webservers
devservers

[dev_sub_gr]
#node[1:20]
node[1:2]
```

`playbook.yml`

```
ansible4 > playbook.yml
1 ---
2 - name: db configuration
3   hosts: db_server
4   tasks:
5     - name: install mariadb and PyMysql
6       become: yes
7       yum:
8         name:
9           - mariadb-server
10          - python3-PyMySQL
11         state: latest
12
```

control node



node1

node2

node3

node4

node5

## Facts

Ansible'in bir hostta bir playbook yürüttüğünde toplanan sistem özellikleridir. Özellikler, hostname'i, OS ailesini, CPU tipini ve CPU çekirdeklerini içerir. Kullanılabilecek facts'lara göz atmak için; `ansible localhost -m setup`

```
[kazm@centosAnsible ~]$ ansible localhost -m setup
localhost | SUCCESS => {
  "ansible_facts": {
    "ansible_all_ipv4_addresses": [
      "192.168.0.120"
    ]
  }
}
```

facts, filtre ile daraltılabilir: `ansible localhost -m setup -a "filter=*ipv4"`

**Yapılandırma Dosyaları :** Ansible yapılandırma dosyası; ayarların yapıldığı, parametrelerin belirlendiği dosyadır. Varsayılan yapılandırma dosyası, `/etc/ansible/ansible.cfg`'dir.

# Ansible Facts

- "ansible\_facts": {
- "ansible\_all\_ipv4\_addresses": [
- "172.31.20.246"
- ],
- "ansible\_all\_ipv6\_addresses": [
- "fe80::88c:37ff:fe8f:3b71"



A N S I B L E

# Ansible Vault

Ansible Vault, Ansible tarafından sağlanan bir özelliktir ve duyarlı verilerin (örneğin şifreler, özel anahtarlar veya diğer gizli bilgiler) şifrelenmesi için kullanılır. Bu, duyarlı veriyi şifreleyerek, bu veriyi bir kaynak kodu deposunda veya başka bir depolama mekanizmasında güvenle saklamanıza olanak tanır. Ancak, Ansible çalıştırıldığında bu verilere erişmek için ilgili şifreyi sağlamanız gerekecektir.

İşte Ansible Vault ile ilgili bazı temel kavramlar:

**Oluşturma:** ``ansible-vault create [dosya_adı]`` komutu ile yeni bir şifreli dosya oluşturabilirsiniz. Bu komut, dosyanın içeriğini düzenlemeniz için bir metin düzenleyici başlatır ve değişikliklerinizi kaydettiğinizde içeriği otomatik olarak şifreler.

**Düzenleme:** Zaten şifreli bir dosyayı düzenlemek isterseniz, ``ansible-vault edit [dosya_adı]`` komutunu kullanabilirsiniz. Bu komut, dosyanın şifresini çözerek içeriğini metin düzenleyicide gösterir ve değişikliklerinizi kaydettiğinizde içeriği tekrar şifreler.

**Şifre Çözme:** ``ansible-vault decrypt [dosya_adı]`` komutu ile şifreli dosyanın içeriğini şifresiz bir hale getirebilirsiniz.

**Şifreleme:** ``ansible-vault encrypt [dosya_adı]`` komutu ile bir dosyanın içeriğini şifreleyebilirsiniz.

# Ansible Vault

- Using Ansible Vault we can create encrypted yaml files and variables

`ansible-vault create secret.yml`

New Vault password: xxxx

Confirm New Vault password: xxxx

- username: tony
- password: 123456a





## Dinamik Envanter Dosyası

AWS gibi envanter dosyasının sunucu ekleyip çıkardıkça sürekli değiştiği bir bulut ortamında, envanter dosyasında tanımlanan hostları takip etmek zordur. IP adreslerini güncellemenin zahmetli hale geldiği bu noktada işin içine dinamik envanter girer.

Dinamik envanter Python, PHP veya başka bir programlama dilinde yazılmış bir betiktir. Sanal bir sunucu durdurulup yeniden başlatıldığında IP adreslerinin değiştiği AWS gibi bulut ortamlarında kullanışlıdır.

Ansible, Google Compute Engine, Amazon EC2 bulut sunucusu, OpenStack, RackSpace, cobbler gibi genel bulut platformları için envanter komut dosyaları geliştirmiştir.

### **Dinamik envanterin statik envantere göre avantajları;**

- Bilgiler betikler kullanılarak toplandığından, dinamik envanterler insan hatasını azaltmada mükemmeldir.
- Envanter yönetiminde minimum çaba gerekir.

Seçilen bir programlama dilinde özelleştirilmiş dinamik envanter yazılabilir. Envanter, uygun seçenekler iletilindiğinde JSON'da bir biçim döndürmelidir.

## HANDS-ON 5: Dynamic Inventory

