# YAML

## کورس کامل

۱۲ درس · مبتدی تا پیشرفته

انگلیسی و دری · رایگان

- DevOps و IaC ✓
- inventory و ad-hoc ✓
- playbook و module ها ✓
- متغیرها و fact ها ✓
- handler و حلقه‌ها ✓
- template های Jinja2 ✓
- role · vault · Windows ✓
- CI/CD · AWX ✓

همه را یاد بگیرید @MyWebUniversity-Dari

# ۱۲ درس این کورس




۱. به **Ansible** و **DevOps** خوش آمدید




۲. نصب **Ansible** و ساختن یک **lab**




۳. **inventory** و دستورهای **ad-hoc**



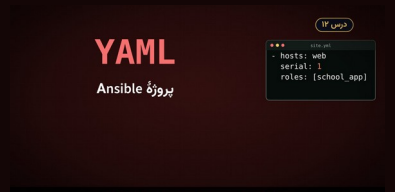

۴. اولین **playbook** شما




۵. **module** های اساسی




۶. متغیرها، **fact** ها و **register**




۷. شرط‌ها، حلقه‌ها و **handler** ها




۸. **template** های **Jinja2**




۹. **role** ها، **collection** ها و **Galaxy**




۱۰. رازها، امتیاز و **host** های **Windows**




۱۱. **Ansible** در یک **pipeline** نوع **DevOps**




۱۲. پروژه: **deploy** کردن یک اپ **web**، و قدم‌های بعدی

یک کورس رایگان از MyWebUniversity

# Ansible و DevOps

درس ۱

## به Ansible و DevOps خوش آمدید

DevOps چیست، زیرساخت به شکل کود، چرا Ansible، push بدون agent و idempotence

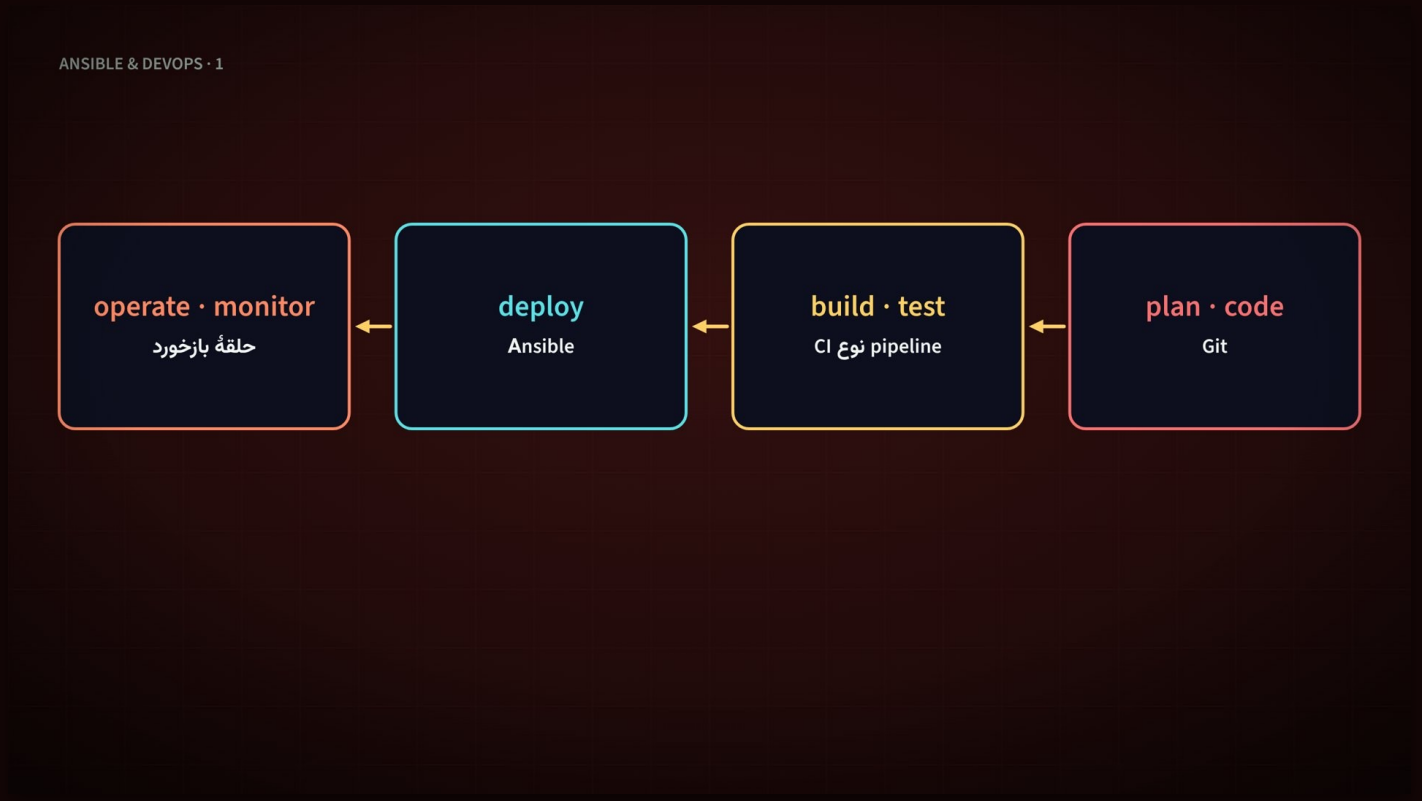


# DevOps چیست؟

plan · code — Git

build · test — pipeline نوع CI

deploy — Ansible




۱.۰۱

# زیرساخت به شکل کود

زیرساخت به شکل کود یعنی server ها با فایل‌هایی که در Git نگه داشته می‌شوند تنظیم می‌شوند، ...

ده server یکسان بسازید، یا یکی را بعد از یک فاجعه دوباره بسازید، با اجرای همان کود.




۱.۰۲

# چرا Ansible؟

بدون agent — فقط SSH · WinRM

خوانا — playbook های YAML

همه چیز آماده — هزاران module




۱.۰۳

# Ansible چطور کار می‌کند

control node — ansible-playbook

inventory — web1، web2، db1

SSH — push کردن module ها

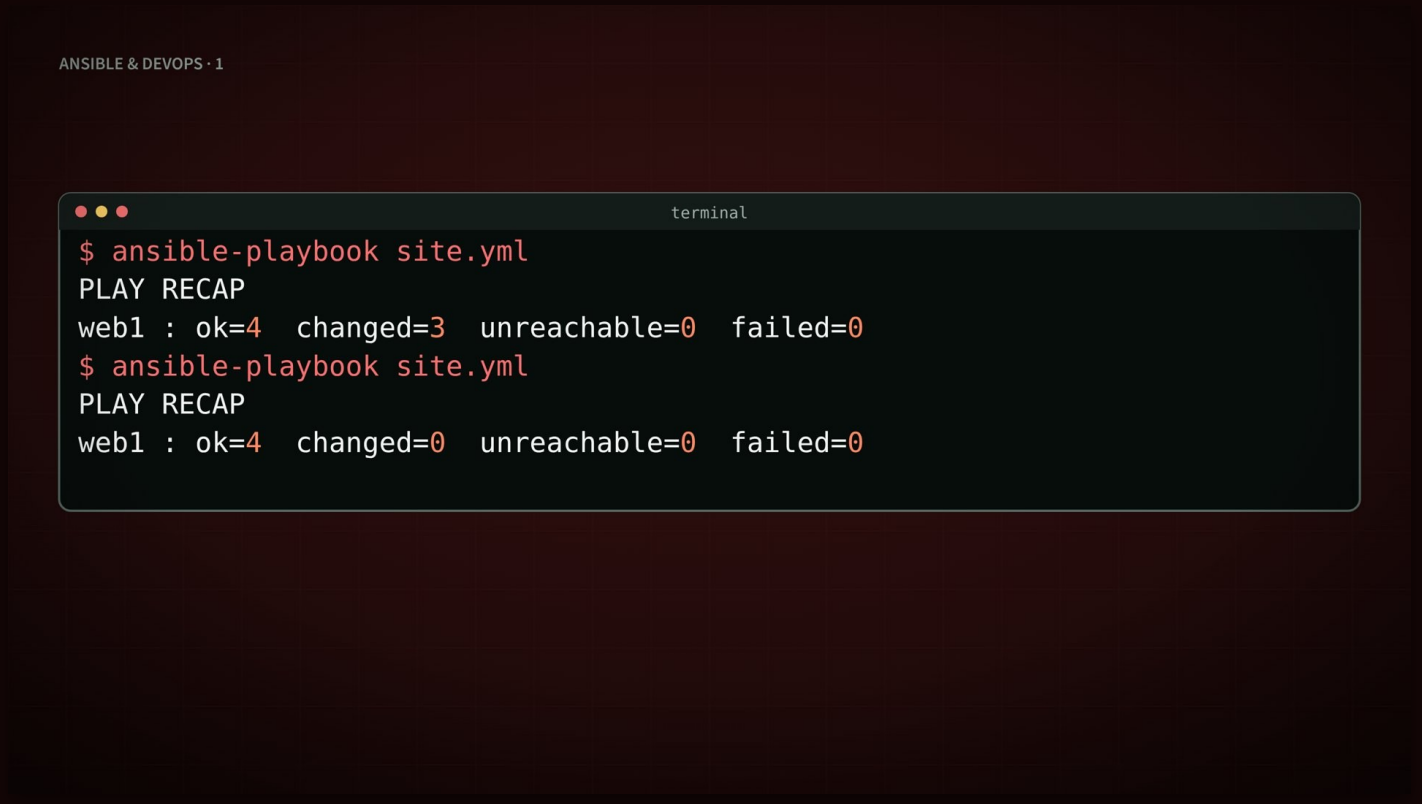



۱۰۰۴

# idempotence

task های Ansible حالت مطلوب را توصیف می‌کنند، نه قدم‌ها را. مطمئن شو nginx نصب است؛ مطمئن ...

یک playbook را یک بار اجرا کنید و تغییرها ایجاد می‌کند. دوباره اجرا کنید و هیچ چیزی را ...




۱.۰۵

یک کورس رایگان از MyWebUniversity

# Ansible و DevOps

درس ۲

## نصب Ansible و ساختن یک lab




control node و node های مدیریت‌شده، نصب با pip یا dnf، کلیدهای SSH، ansible.cfg و اولین g

# طراحی lab

یک lab یک control node و چند node مدیریت‌شده لازم دارد. از ماشین‌های مجازی با VirtualBox یا ...

lab ما: control، دو web server و یک database server، که Rocky Linux و Ubuntu اجرا می‌کنند.



| host | IP | نقش |
|---|---|---|
| control | 192.168.56.10 | control node نوع Ansible |
| web1 | 192.168.56.11 | nginx ،Rocky Linux |
| web2 | 192.168.56.12 | nginx ،Ubuntu |
| db1 | 192.168.56.13 | PostgreSQL ،Rocky Linux |

۲.۰۱

# نصب Ansible

control node باید Linux یا macOS باشد؛ روی Windows، از WSL استفاده کنید. با pip، یا با ...

ansible-core موتور است؛ package به نام ansible، collection های جامعه از module ها را اضافه ...

۲.۰۲



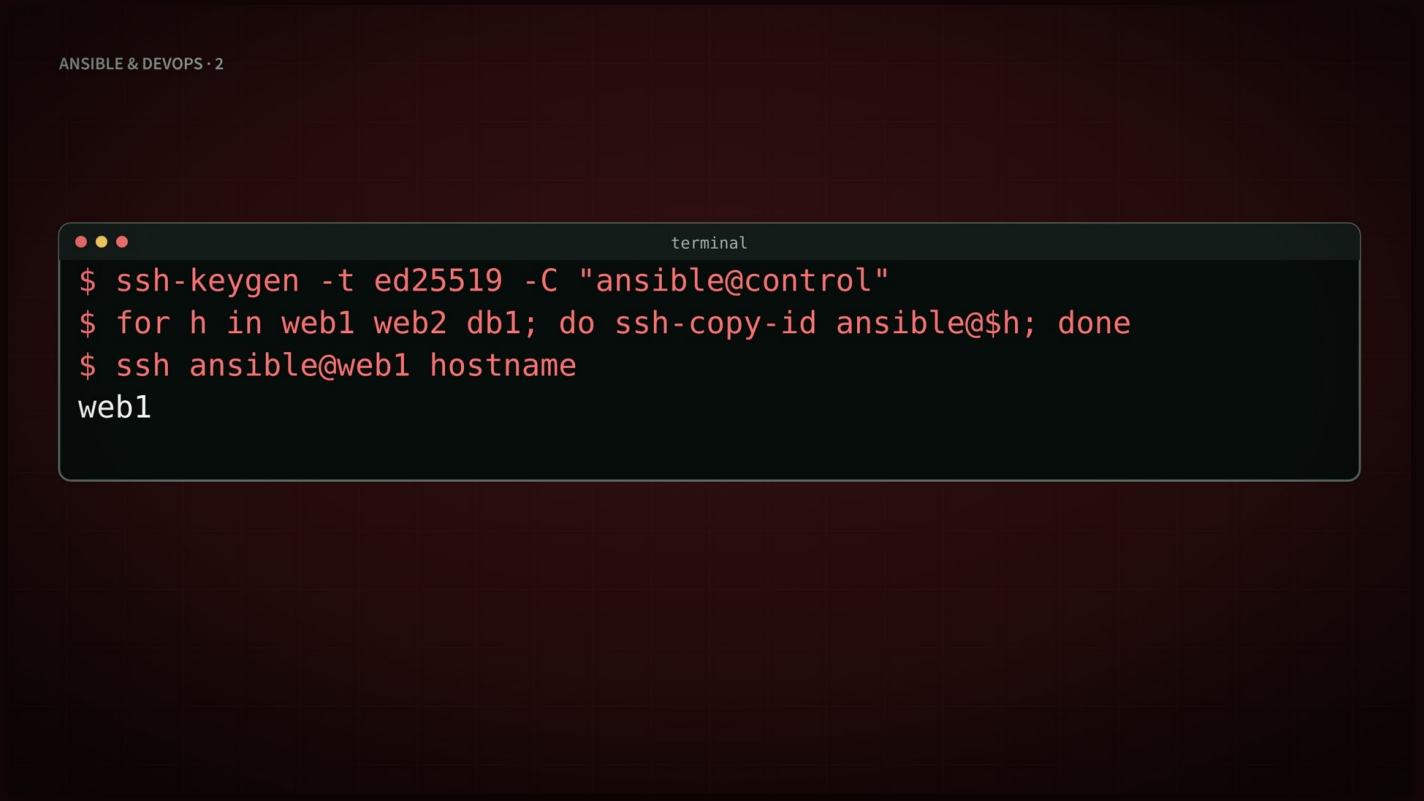
terminal

```
$ python3 -m pip install --user ansible
$ sudo dnf install ansible-core        # RHEL, Rocky
$ sudo apt install ansible             # Ubuntu
$ ansible --version
ansible [core 2.17.5]
  python version = 3.12.3
```

# کلیدهای SSH

Ansible با SSH وارد می‌شود، پس ورود بر اساس کلید را تنظیم کنید. یک کلید روی control node ...

از یک کاربر مخصوص، مثل ansible، با حقوق sudo روی node های مدیریت‌شده استفاده کنید.




۲.۰۳

# ansible.cfg

یک ansible.cfg در پوشهٔ پروژهٔ شما پیش‌فرض‌ها را تعیین می‌کند: inventory کجاست، با کدام ...

Ansible اول فایل پوشهٔ فعلی را می‌خواند، پس هر پروژه می‌تواند تنظیمات خودش را داشته باشد.



ansible.cfg

```
[defaults]
inventory         = ./inventory.ini
remote_user       = ansible
host_key_checking = True
forks             = 20

[privilege_escalation]
become        = True
become_method = sudo
```

۲۰۰۴

# اولین دستور

ANSIBLE & DEVOPS · 2

Ansible به نام module ping آزمایش می‌کند که می‌تواند وارد شود و Python را روی هر host اجرا ...

pong سبز یعنی موفقیت؛ سرخ یعنی SSH، کاربر یا Python را روی آن host بررسی کنید.

terminal

```
$ ansible all -m ping
web1 | SUCCESS => { "ping": "pong" }
web2 | SUCCESS => { "ping": "pong" }
db1  | SUCCESS => { "ping": "pong" }
```

۲.۰۵

یک کورس رایگان از MyWebUniversity

# Ansible و DevOps

درس ۳

## inventory و دستورهای ad-hoc

inventory های INI و YAML، گروه‌ها، متغیرهای host و گروه، pattern ها و دستورهای ad-hoc

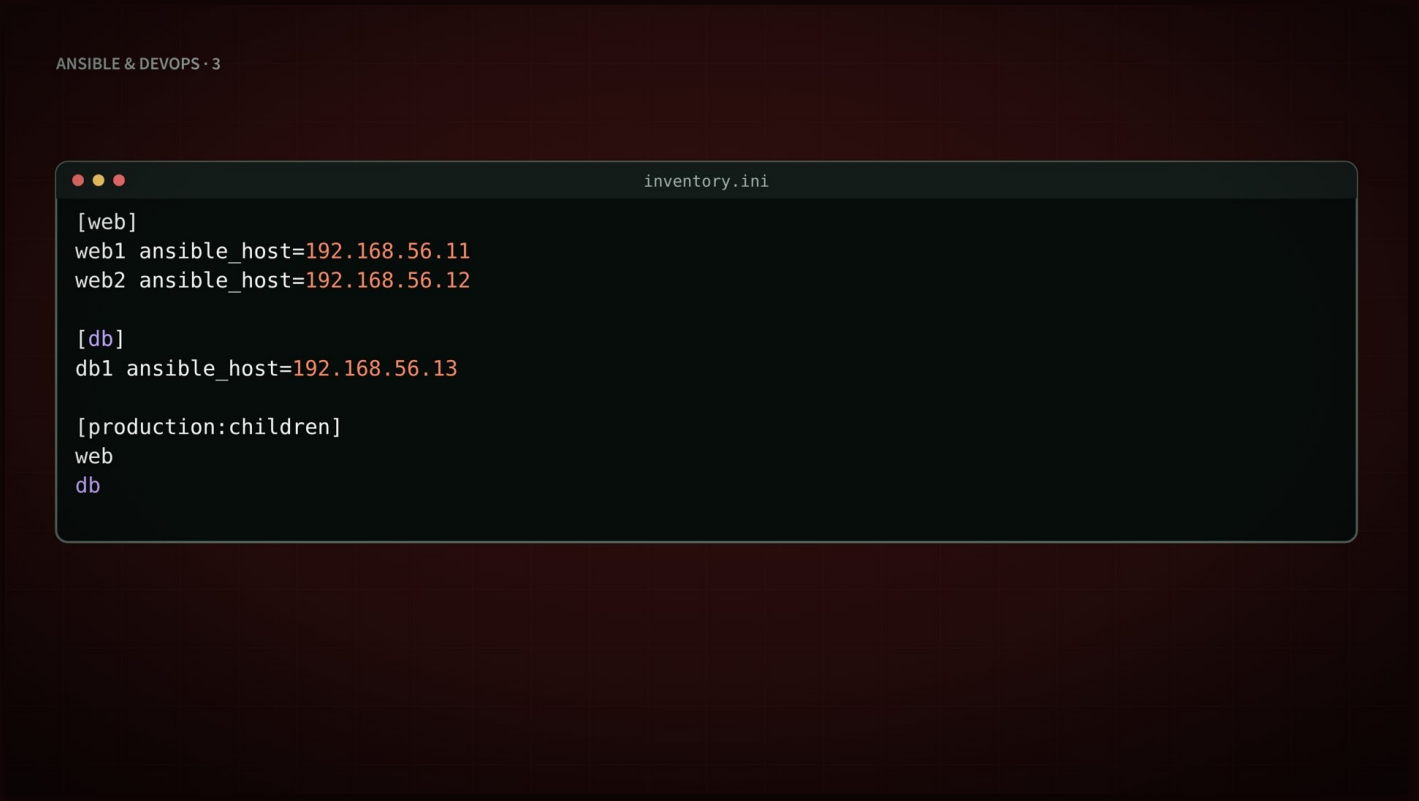


# یک inventory نوع INI

inventory، host های شما را فهرست می‌کند و آن‌ها را در گروه‌هایی داخل قوس‌های مربع می‌گذارد.

...

هر host خودکار در گروهی به نام all است.




۳.۰۱

# یک inventory نوع YAML

همان inventory می‌تواند به YAML نوشته شود، که برای ساختارهای بزرگ و تو در تو آسان‌تر است.

ansible-inventory با --graph درخت را چاپ می‌کند، تا بتوانید گروه‌های خود را بررسی کنید.



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 }
```

۳.۰۲

# host_vars و group_vars

متغیرها را بیرون از inventory در پوشه‌هایی به نام group_vars و host_vars نگه دارید، یک فایل ...

Ansible آن‌ها را خودکار بار می‌کند، پس همهٔ web server ها می‌توانند یک http_port را شریک ...



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
```

۳.۰۳

# pattern های host

pattern ها انتخاب می‌کنند یک دستور کدام host ها را هدف قرار دهد: یک host، یک گروه، all، یا ...

limit-- هر اجرا را بیشتر محدود می‌کند، که یک عادت امن است.



| pattern | هدف |
|---|---|
| all | هر host |
| web | گروه web |
| web:db | web یا db |
| production:&web | در هر دو گروه |
| all:!db1 | همه به جز db1 |
| web[0] | اولین host نوع web |

۳.۰۴

# دستورهای ad-hoc

یک دستور ad-hoc یک module را روی host های زیاد بدون نوشتن یک playbook اجرا می‌کند. از m- ...

برای بررسی‌های سریع عالی است؛ هر چیزی که تکرار می‌کنید جایش در یک playbook است.



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"
```

۳.۰۵

یک کورس رایگان از MyWebUniversity

# Ansible و DevOps

درس ۴

## اولین playbook شما




اساسات YAML، play ها و task ها، اجرای playbook ها، حالت‌های check و diff، خواندن خروجی

# اساسات YAML

YAML از تورفتگی با فاصله‌ها، هرگز tab ها، استفاده می‌کند. یک خط تیره یک قلم لیست را شروع ...

string ها کم پیش می‌آید quote لازم داشته باشند، اما هر چیزی که با آکولاد شروع می‌شود یا یک ...



example.yml

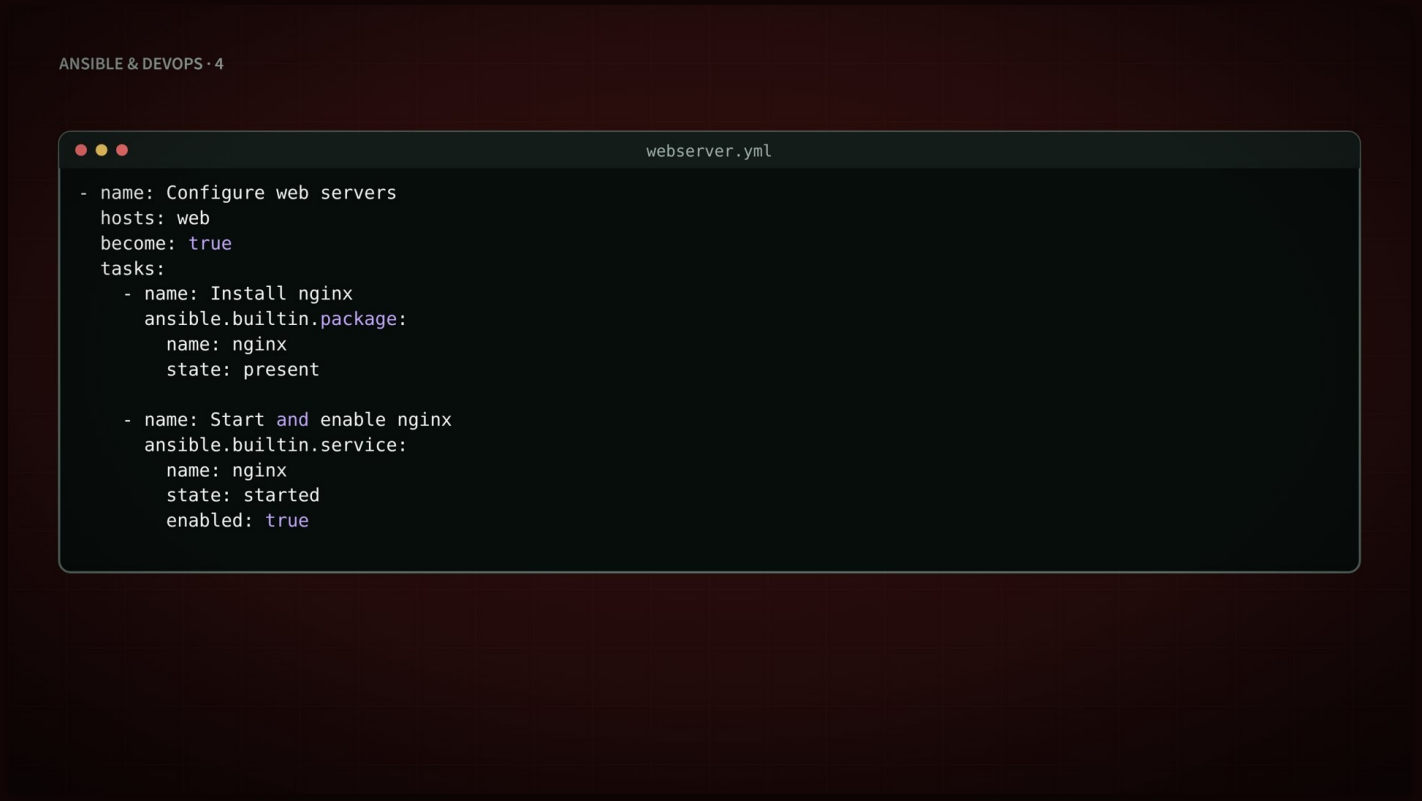
```yaml
server:
  name: web1
  ports:
    - 80
    - 443
  enabled: true
  motd: "Welcome: authorized users only"
```

۴.۰۱

# play ها و task ها

یک playbook لیستی از play ها است. هر play، host ها را هدف قرار می‌دهد و لیستی از task ها را ...

از نام‌های کامل module مثل ansible.builtin.package استفاده کنید، تا همیشه واضح باشد کدام ...

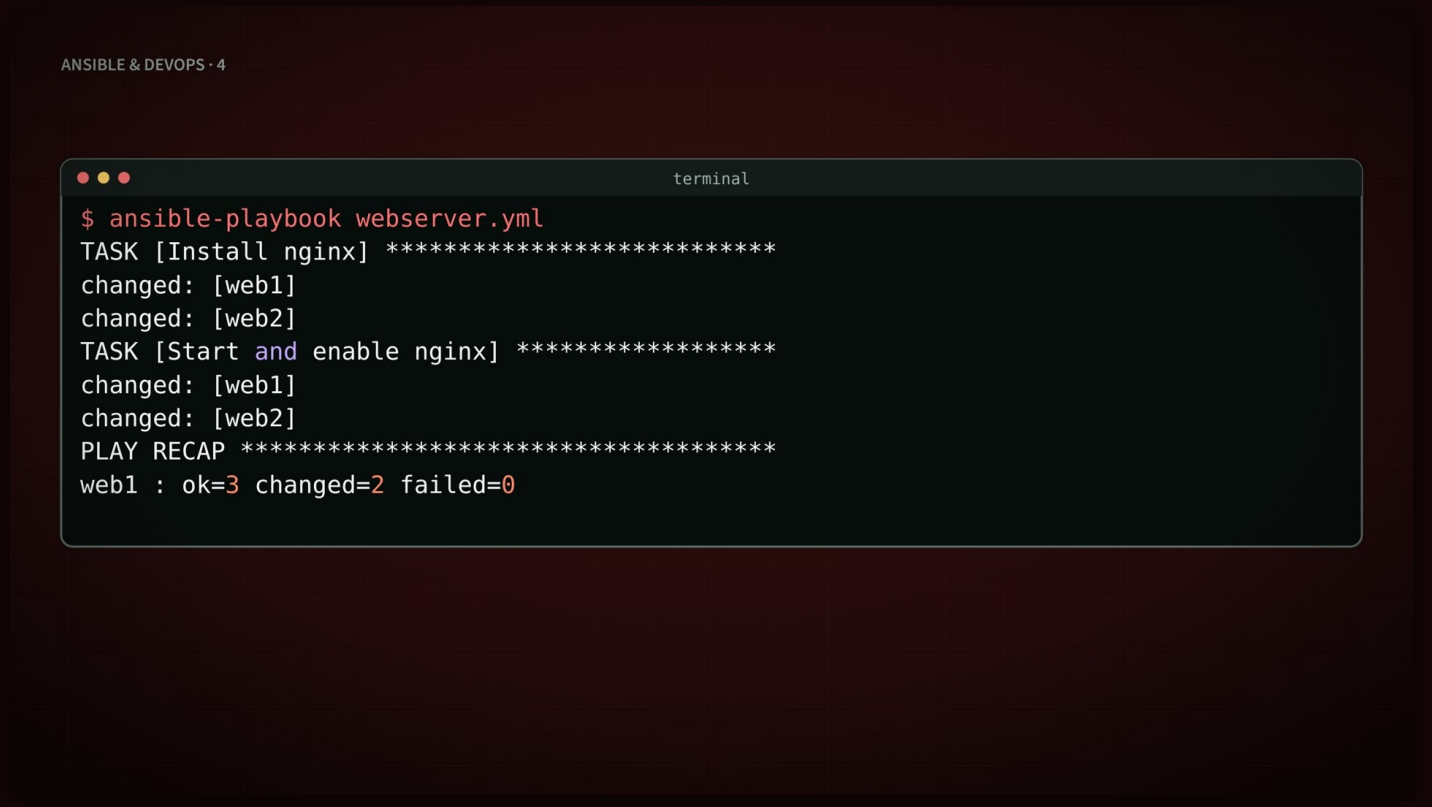



۴۰۲

# اجرای یک playbook

ansible-playbook آن را اجرا می‌کند. هر task برای هر host، ok، changed، skipped یا failed ...

اگر یک task روی یک host ناکام شود، Ansible برای آن host متوقف می‌شود و با دیگران ادامه می‌دهد.




۴.۰۳

# syntax و check، diff

پیش از دست زدن به تولید، امن آزمایش کنید. --syntax-check اشتباه‌های YAML را پیدا می‌کند.

...

--list-tasks و --list-hosts پیش‌نمایش می‌دهند چه چیزی کجا اجرا خواهد شد.



terminal

```
$ ansible-playbook webserver.yml --syntax-check
$ ansible-playbook webserver.yml --list-hosts
$ ansible-playbook webserver.yml --check --diff
$ ansible-playbook webserver.yml --limit web1 -v
```

۴۰۴

# play های متعدد

یک فایل playbook می‌تواند چند play برای گروه‌های مختلف نگه دارد، که به ترتیب اجرا می‌شوند: ...

یک site.yml که playbook های کوچک‌تر را import می‌کند یک pattern رایج برای یک محیط کامل است.



site.yml

```
- name: Common baseline
  ansible.builtin.import_playbook: common.yml

- name: Databases
  ansible.builtin.import_playbook: dbservers.yml

- name: Web tier
  ansible.builtin.import_playbook: webserver.yml
```

۴.۰۵

یک کورس رایگان از MyWebUniversity

# Ansible و DevOps

درس ۵

## module های اساسی

package، service، copy، file، user، lineinfile، command در برابر shell، و ansible-doc



# package ها و خدمات

package با هر package manager کار می‌کند؛ dnf و apt option های بیشتر می‌دهند. state: ...

service، یا systemd، خدمات را start، stop، restart و enable می‌کند.



tasks.yml

```yaml
- name: Install tools
  ansible.builtin.package:
    name: [git, curl, vim]
    state: present

- name: Firewall service running
  ansible.builtin.systemd:
    name: firewalld
    state: started
    enabled: true
```

۵.۰۱

# copy و file

copy یک فایل را از control node می‌فرستد، یا محتوای داخلی می‌نویسد، و مالک، گروه و mode را ...

همیشه mode های octal، مثل 0644 داخل quote، را quote کنید، تا YAML آن‌ها را اشتباه نخواند.



tasks.yml

```yaml
- name: Web root exists
  ansible.builtin.file:
    path: /var/www/school
    state: directory
    owner: nginx
    mode: "0755"

- name: Deploy home page
  ansible.builtin.copy:
    src: files/index.html
    dest: /var/www/school/index.html
    mode: "0644"
```

۵.۰۲

# کاربرها و خط‌ها در فایل‌ها

user حساب‌ها را با گروه‌ها و shell ها می‌سازد؛ authorized_key کلیدهای SSH آن‌ها را نصب ...

این‌ها تنظیمات را بدون جایگزینی کامل فایل‌ها امن ویرایش می‌کنند.



tasks.yml

```
- name: Create deploy user
  ansible.builtin.user:
    name: deploy
    groups: wheel
    shell: /bin/bash

- name: Disable SSH root login
  ansible.builtin.lineinfile:
    path: /etc/ssh/sshd_config
    regexp: '^#?PermitRootLogin'
    line: 'PermitRootLogin no'
```

۵.۰۳

# command در برابر shell

command یک پروگرام را مستقیماً اجرا می‌کند؛ shell آن را از طریق یک shell اجرا می‌کند، پس ...

task های command همیشه changed گزارش می‌دهند. creates، یا changed_when را اضافه کنید تا ...



tasks.yml

```yaml
- name: Initialize app database once
  ansible.builtin.command:
    cmd: /opt/app/bin/init-db
    creates: /opt/app/.db-initialized

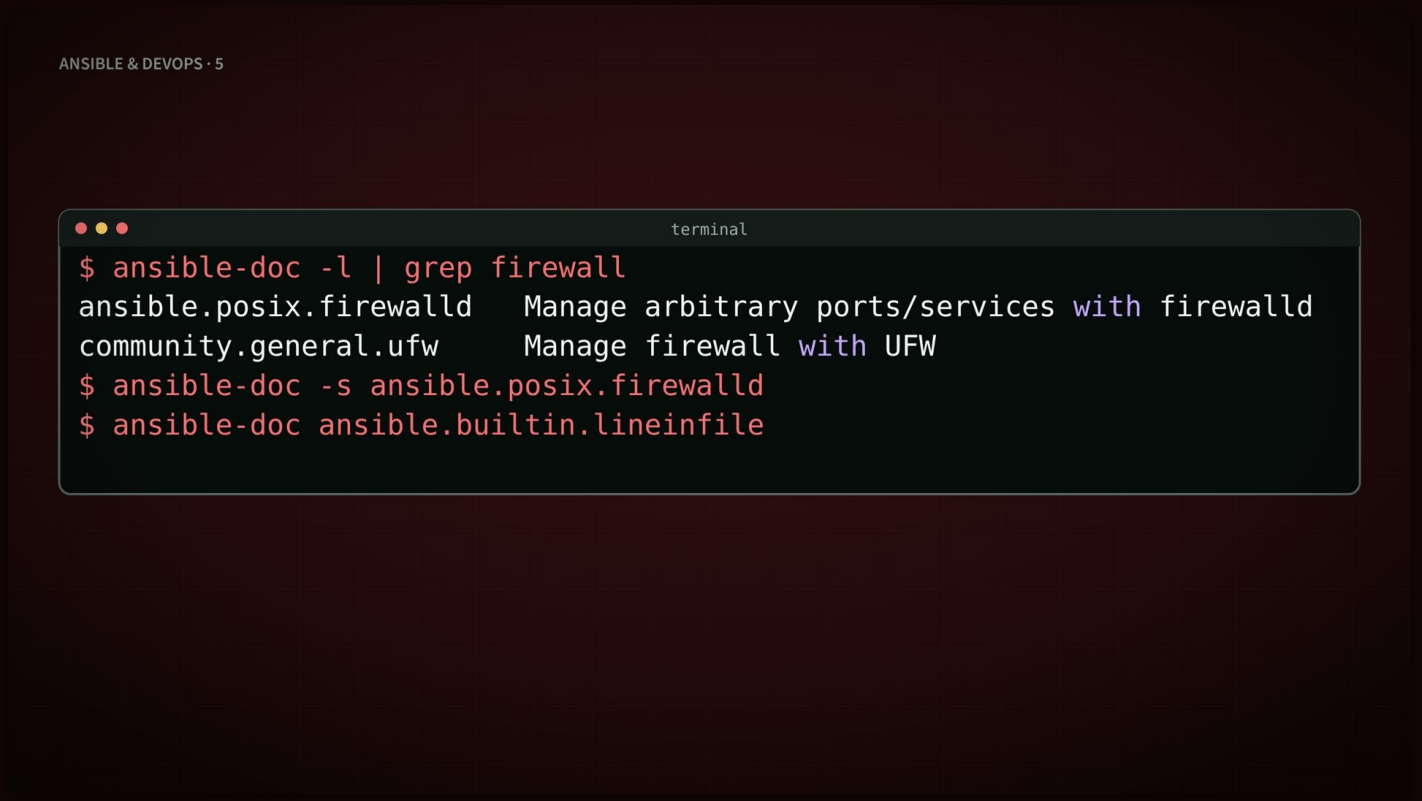
- name: Count log errors
  ansible.builtin.shell: grep -c ERROR /var/log/app.log || true
  register: errors
  changed_when: false
```

۵.۰۴

# پیدا کردن module ها با ansible-doc

ansible-doc مستندات و مثال‌ها را برای هر module، درست در terminal، نشان می‌دهد. l- آن‌ها ...

بخش مثال‌ها معمولاً سریع‌ترین راه برای یاد گرفتن یک module نو است.




۵.۰۵

یک کورس رایگان از MyWebUniversity

# Ansible و DevOps

درس ۶

## متغیرها، fact ها و register

تعریف متغیرها، عبارت‌های Jinja2، fact ها با setup، register، debug و اولویت



# تعریف متغیرها

متغیرها می‌توانند در بخش vars یک play، در فایل‌های vars، در group_vars و host_vars، یا در ...

از آن‌ها با آکولاد دوگانه استفاده کنید. وقتی یک مقدار با یک متغیر شروع می‌شود، کل مقدار را ...

ANSIBLE & DEVOPS · 6

app.yml

```yaml
- hosts: web
  vars:
    app_name: school
    app_port: 8080
  tasks:
    - name: Create app directory
      ansible.builtin.file:
        path: "/opt/{{ app_name }}"
        state: directory

# ansible-playbook app.yml -e app_port=9090
```

۶.۰۱

# fact ها

در شروع هر play، Ansible، fact ها را جمع می‌کند: سیستم عامل، آدرس‌های IP، حافظه، دیسک‌ها و ...

از آن‌ها برای تطبیق استفاده کنید، مثلاً انتخاب نام package برای Red Hat یا Debian.




۶.۰۲

# register و debug

register نتیجهٔ یک task را در یک متغیر ذخیره می‌کند: خروجی آن، کود برگشت و اینکه آیا تغییر ...

با هم به یک task اجازه می‌دهند بر اساس task دیگر تصمیم بگیرد.

ANSIBLE & DEVOPS · 6

check.yml

```yaml
- name: Check disk space
  ansible.builtin.command: df -h /
  register: disk
  changed_when: false

- name: Show result
  ansible.builtin.debug:
    msg: "{{ inventory_hostname }}: {{ disk.stdout_lines[1] }}"
```

۶.۰۳

# فیلترها

فیلترهای Jinja2 مقدارها را با یک pipe تبدیل می‌کنند: default یک جایگزین می‌دهد، upper و ...

فیلتر default وقتی یک متغیر تعریف نشده از خطاها جلوگیری می‌کند.



filters.yml

```yaml
- ansible.builtin.debug:
    msg:
      - "{{ app_env | default('production') }}"
      - "{{ app_name | upper }}"
      - "{{ groups['web'] | join(', ') }}"
      - "{{ ansible_memtotal_mb // 1024 }} GB RAM"
```

۶.۰۴

# اولویت متغیرها

role defaults — ضعیف‌ترین

group_vars — all → گروه‌ها

host_vars · play vars — مشخص‌تر




۶.۰۵

یک کورس رایگان از MyWebUniversity

# Ansible و DevOps

درس ۷

## شرط‌ها، حلقه‌ها و handler ها

when، loop، loop_control، handler ها و notify، block ها با rescue، tag ها



# شرط‌ها با when

when یک task را فقط وقتی اجرا می‌کند که یک شرط درست باشد. عبارت را بدون آکولاد بنویسید. ...

استفادهٔ کلاسیک: انتخاب task ها بر اساس خانوادهٔ سیستم عامل.



os.yml

```yaml
- name: Install Apache on Red Hat family
  ansible.builtin.dnf:
    name: httpd
  when: ansible_facts['os_family'] == 'RedHat'

- name: Install Apache on Debian family
  ansible.builtin.apt:
    name: apache2
    update_cache: true
  when: ansible_facts['os_family'] == 'Debian'
```

۷.۰۱

# حلقه‌ها

loop یک task را برای هر قلم در یک لیست تکرار می‌کند، و item مقدار فعلی را نگه می‌دارد. روی ...

loop_control با label خروجی را کوتاه و خوانا نگه می‌دارد.



users.yml

```yaml
- name: Create team accounts
  ansible.builtin.user:
    name: "{{ item.name }}"
    groups: "{{ item.groups }}"
  loop:
    - { name: mariam, groups: wheel }
    - { name: ahmad,  groups: developers }
    - { name: zahra,  groups: developers }
  loop_control:
    label: "{{ item.name }}"
```

۷.۰۲

# handler ها

یک handler یک task است که فقط وقتی notify شود اجرا می‌شود، و فقط یک بار، در آخر play. از آن ...

اگر تنظیمات تغییر نکرد، هیچ restart نیست، پس خدمت بدون مزاحمت می‌ماند.



nginx.yml

```
tasks:
  - name: Deploy nginx config
    ansible.builtin.template:
      src: nginx.conf.j2
      dest: /etc/nginx/nginx.conf
      validate: nginx -t -c %s
    notify: Restart nginx

handlers:
  - name: Restart nginx
    ansible.builtin.service:
      name: nginx
      state: restarted
```

۷.۰۳

# block ها، rescue و always

ANSIBLE & DEVOPS · 7

deploy.yml

```yaml
- block:
    - name: Deploy new release
      ansible.builtin.unarchive:
        src: app-2.0.tar.gz
        dest: /opt/app
  rescue:
    - name: Restore previous release
      ansible.builtin.command: /opt/app/bin/rollback
  always:
    - name: Notify the team
      ansible.builtin.debug:
        msg: "Deploy finished on {{ inventory_hostname }}"
```

block، task ها را گروه‌بندی می‌کند تا تنظیماتی مثل become یا when را شریک شوند. rescue اگر ...

از آن برای پس گرفتن پاک یک deployment ناکام استفاده کنید.

۷.۰۴

# ها tag

tag ها را برچسب می‌زنند تا بتوانید فقط task ها را اجرا کنید. --tags، ... playbook بخشی از یک

task ها را بر اساس هدف، مثل packages، config و tag، users کنید.

ANSIBLE & DEVOPS · 7

terminal

```
$ ansible-playbook site.yml --list-tags
      TASK TAGS: [config, packages, users]
$ ansible-playbook site.yml --tags config
$ ansible-playbook site.yml --skip-tags packages
```

۷.۰۵

یک کورس رایگان از MyWebUniversity

# Ansible و DevOps

درس ۸

## template های Jinja2

module به نام template، متغیرها، حلقه‌ها و شرط‌ها در template ها، فاصله‌ها و بررسی اعتبار



# چرا template ها؟

- nginx.conf.j2 — یک template
- متغیرها + fact ها — برای هر host
- module به نام template — render می‌کند




۸.۰۱

# متغیرها در template ها

داخل یک template، آکولاد دوگانه مقدارها را، از متغیرها یا fact ها، چاپ می‌کند. یک comment ...

module به نام template نتیجهٔ render شده را با مالک و mode کاپی می‌کند، و فقط اگر محتوا فرق ...



templates/site.conf.j2

```
# {{ ansible_managed }}
server {
    listen {{ http_port }};
    server_name {{ inventory_hostname }}.school.edu;
    root /var/www/{{ app_name }};
    worker_connections {{ ansible_processor_vcpus * 1024 }};
}
```

۸.۰۲

# حلقه‌ها و شرط‌ها

دستورها از آکولاد و درصد استفاده می‌کنند: حلقه‌های for خط‌ها را تکرار می‌کنند، و بلوک‌های ...

اینجا یک template یک لیست load balancer را از هر host در گروه web می‌سازد.



templates/haproxy.cfg.j2

```
backend web_servers
    balance roundrobin
{% for host in groups['web'] %}
    server {{ host }} {{ hostvars[host]['ansible_default_ipv4']['address'] }}:{{ http_port }} check
{% endfor %}
{% if enable_stats | default(false) %}
listen stats
    bind *:8404
{% endif %}
```

۸.۰۳

# task نوع template

task، template مبدأ و فایل مقصد را نام می‌برد.
validate فایل نو را پیش از اینکه جایگزین ...

backup: yes یک کاپی با تاریخ از فایل قبلی نگه می‌دارد.



lb.yml

```yaml
- name: Configure HAProxy
  ansible.builtin.template:
    src: haproxy.cfg.j2
    dest: /etc/haproxy/haproxy.cfg
    owner: root
    mode: "0644"
    validate: haproxy -c -f %s
    backup: true
  notify: Reload haproxy
```

۸.۰۴

# فاصله‌ها و آزمایش

دستورهای Jinja2 می‌توانند خط‌های خالی بگذارند. یک خط تیره داخل tag، مثل {%-، فاصله را در ...

با check-- و diff-- اجرا کنید تا دقیقاً ببینید فایل render شده پیش از اعمال چطور فرق ...



terminal

```
$ ansible-playbook lb.yml --check --diff
--- before: /etc/haproxy/haproxy.cfg
+++ after: haproxy.cfg.j2
@@ -4,3 +4,4 @@
     server web1 192.168.56.11:80 check
+    server web2 192.168.56.12:80 check
```

۸.۰۵

یک کورس رایگان از MyWebUniversity

# Ansible و DevOps

درس ۹

## role ها، collection ها و Galaxy




ساختار role، defaults و vars، استفاده از role ها، Ansible Galaxy، collection ها و irements.yml

# ساختار role

یک role همه چیز را برای یک کار، مثل nginx یا postgres، در یک چیدمان استاندارد پوشه ...

ansible-galaxy role init اسکلت را برای شما می‌سازد.



terminal

```
$ ansible-galaxy role init roles/nginx
roles/nginx/
├── defaults/main.yml      # default variables
├── handlers/main.yml
├── tasks/main.yml
├── templates/
├── files/
├── vars/main.yml
└── meta/main.yml          # dependencies, platforms
```

۹.۰۱

# defaults در برابر vars

defaults برای بازنویسی ساخته شده‌اند: port ها، نسخه‌ها، کلیدهای ویژگی. vars مقدارهایی را ...

هر default را مستند کنید، چون رابط عمومی role است.



roles/nginx/defaults/main.yml

```yaml
# Port nginx listens on
nginx_port: 80
# Site name and document root
nginx_site_name: school
nginx_root: "/var/www/{{ nginx_site_name }}"
# Enable TLS (requires cert files)
nginx_tls: false
```

۹.۰۲

# استفاده از role ها

role ها را در بخش roles یک play فهرست کنید، و متغیرها را برای بازنویسی defaults بفرستید. ...

include_role و import_role یک role را از داخل لیست task ها، حتی به شکل شرطی، صدا می‌کنند.



site.yml

```yaml
- hosts: web
  become: true
  roles:
    - common
    - role: nginx
      vars:
        nginx_port: 8080

- hosts: db
  become: true
  roles:
    - common
    - postgres
```

۹.۰۳

# Ansible Galaxy و collection ها

Ansible Galaxy مرکز عمومی برای role ها و collection های شریک‌شده است. یک collection، module ... 

Automation Hub متعلق به Red Hat، collection های تصدیق‌شده و پشتیبانی‌شده پیشنهاد می‌کند.




۹.۰۴

# requirements.yml

هر role و collection که پروژهٔ شما لازم دارد را، با نسخه‌ها، در requirements.yml ثابت کنید. ...

role های شخص ثالث را پیش از استفاده بازبینی کنید؛ آن‌ها با امتیاز root روی server های شما ...



requirements.yml

```yaml
collections:
  - name: community.general
    version: ">=9.0.0"
  - name: ansible.posix
    version: 1.6.2
roles:
  - name: geerlingguy.docker
    version: 7.4.1

# ansible-galaxy install -r requirements.yml
```

۹.۰۵

یک کورس رایگان از MyWebUniversity

# Ansible و DevOps

درس ۱۰

## رازها، امتیاز و host های Windows

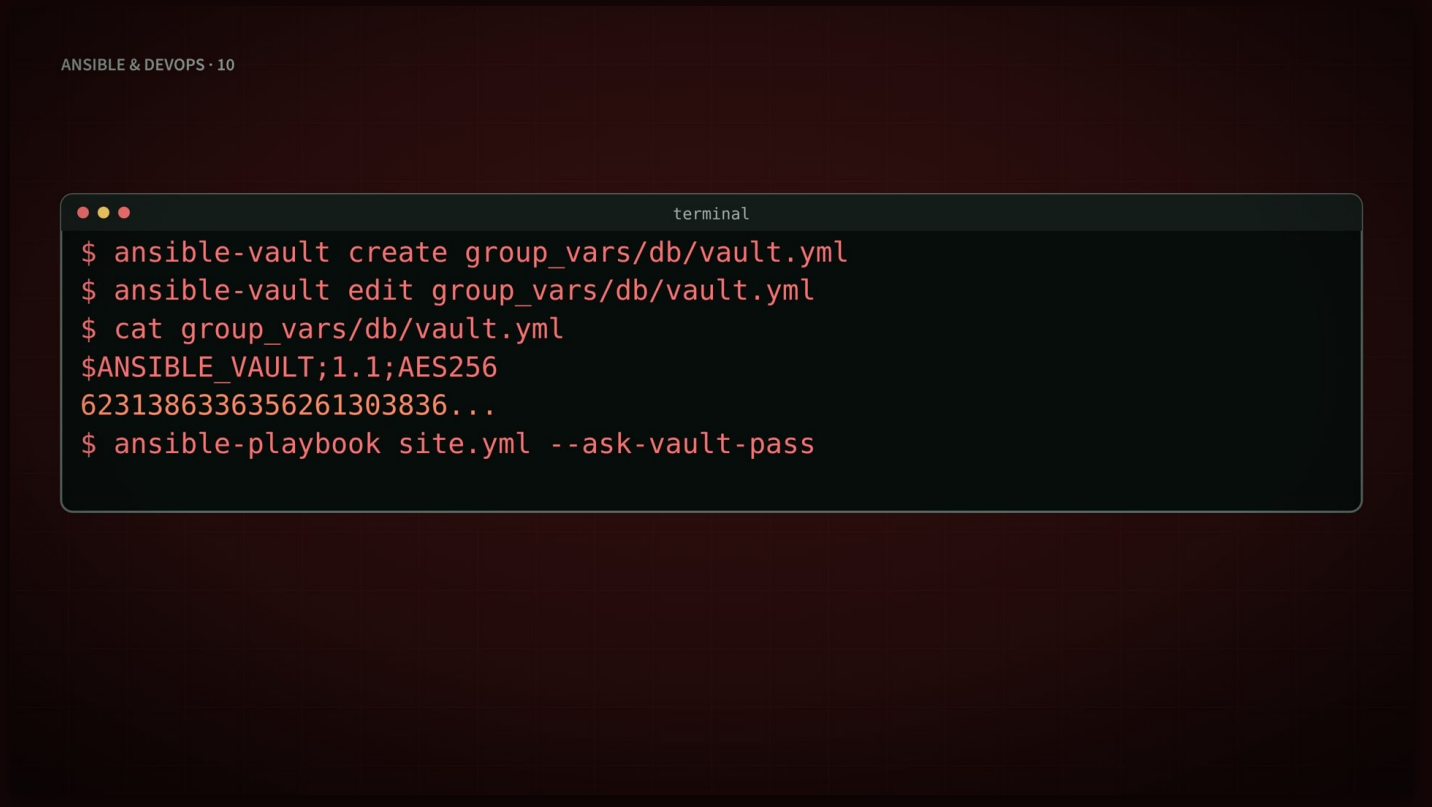



ansible-vault، no_log، become، مدیریت Windows با WinRM و module های win_

# ansible-vault

هرگز رمزها را به شکل متن ساده commit نکنید.
ansible-vault یک فایل کامل، یا یک مقدار تکی، را ...

فایل‌های رمزی را با ansible-vault edit ویرایش کنید، و هنگام اجرا رمز را با ...




۱۰.۰۱

# یک pattern پاک vault

برای هر گروه دو فایل نگه دارید: vars.yml با متغیرهای عادی که به متغیرهای vault_ اشاره ...

هر کس هنوز می‌تواند جستجو کند کدام متغیرها وجود دارند بدون رمزگشایی هیچ چیز. no_log: true ...




۱۰.۰۲

# become و امتیاز

task ،become ها را با حقوق بالا اجرا می‌کند، به طور پیش‌فرض با sudo به عنوان root. آن را ...

فقط آنچه لازم است را بدهید: become در سطح task امن‌تر از همه جا است.



db.yml

```yaml
- name: Create application database
  community.postgresql.postgresql_db:
    name: school
  become: true
  become_user: postgres

- name: Create database user
  community.postgresql.postgresql_user:
    db: school
    name: "{{ db_user }}"
    password: "{{ db_password }}"
  become: true
  become_user: postgres
  no_log: true
```

۱۰.۰۳

# مدیریت Windows

Ansible، Windows را از طریق WinRM، یا SSH روی سیستم‌های نوتر، مدیریت می‌کند.
... control node

win_ping اتصال را آزمایش می‌کند، درست مثل ping برای Linux.



group_vars/windows.yml

```
ansible_connection: winrm
ansible_port: 5986
ansible_winrm_transport: ntlm
ansible_winrm_server_cert_validation: validate
ansible_user: CORP\\ansible-svc
ansible_password: "{{ vault_win_password }}"

# ansible windows -m ansible.windows.win_ping
```

۱۰.۰۴

# module های Windows

collection های ansible.windows و community.windows، win_feature را برای role ها و feature ...

یک playbook می‌تواند صدها Windows server را patch و تنظیم کند.

ANSIBLE & DEVOPS · 10

iis.yml

```yaml
- hosts: windows
  tasks:
    - name: Install IIS
      ansible.windows.win_feature:
        name: Web-Server
        include_management_tools: true

    - name: Install security updates
      ansible.windows.win_updates:
        category_names: [SecurityUpdates]
        reboot: true
```

۱۰.۰۵

یک کورس رایگان از MyWebUniversity

# Ansible و DevOps

درس ۱۱

## Ansible در یک pipeline نوع DevOps

جریان کار Git، ansible-lint، آزمایش با Molecule، CI/CD با GitHub Actions یا Jenkins، محیط‌ه

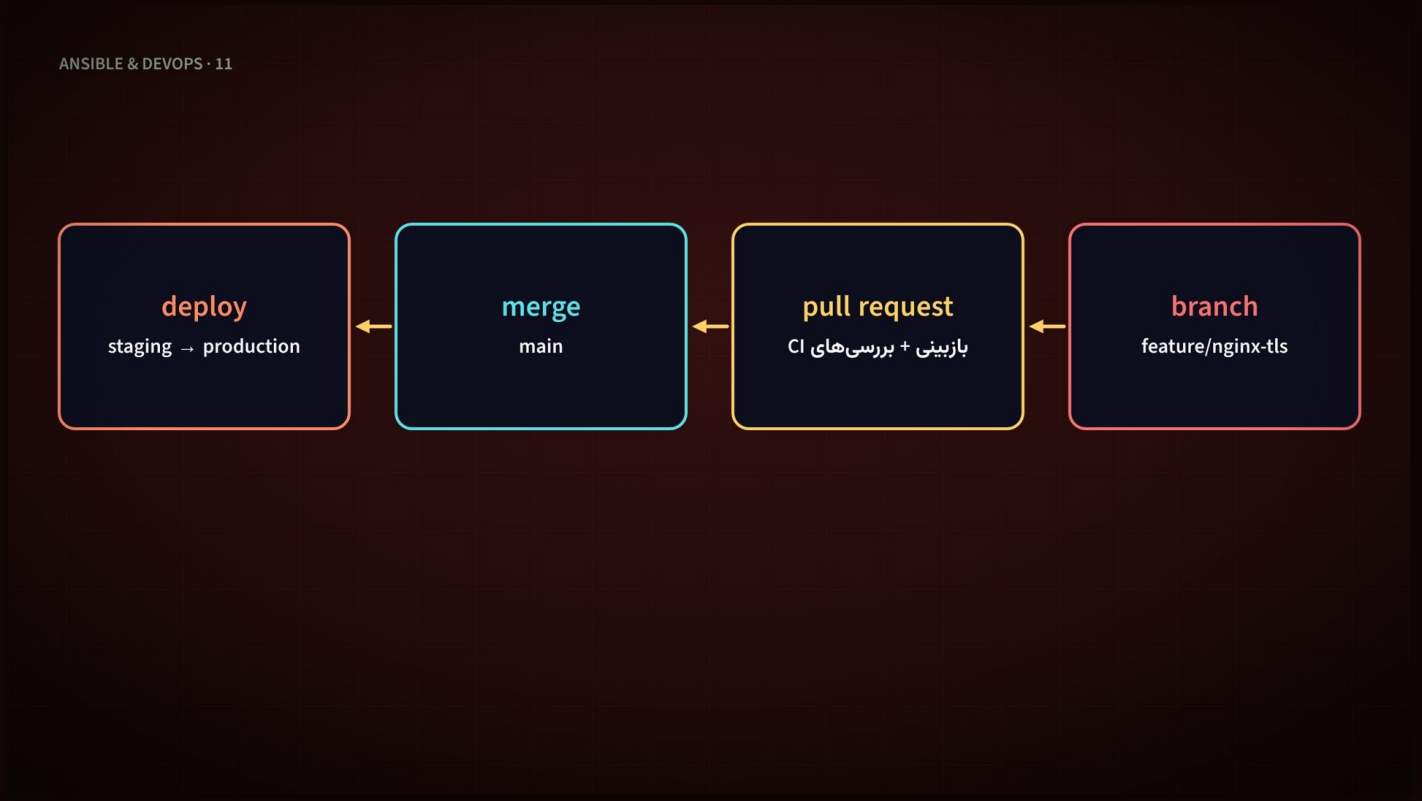


# جریان کار Git

branch — feature/nginx-tls

pull request — بازبینی + بررسی‌های CI

merge — main

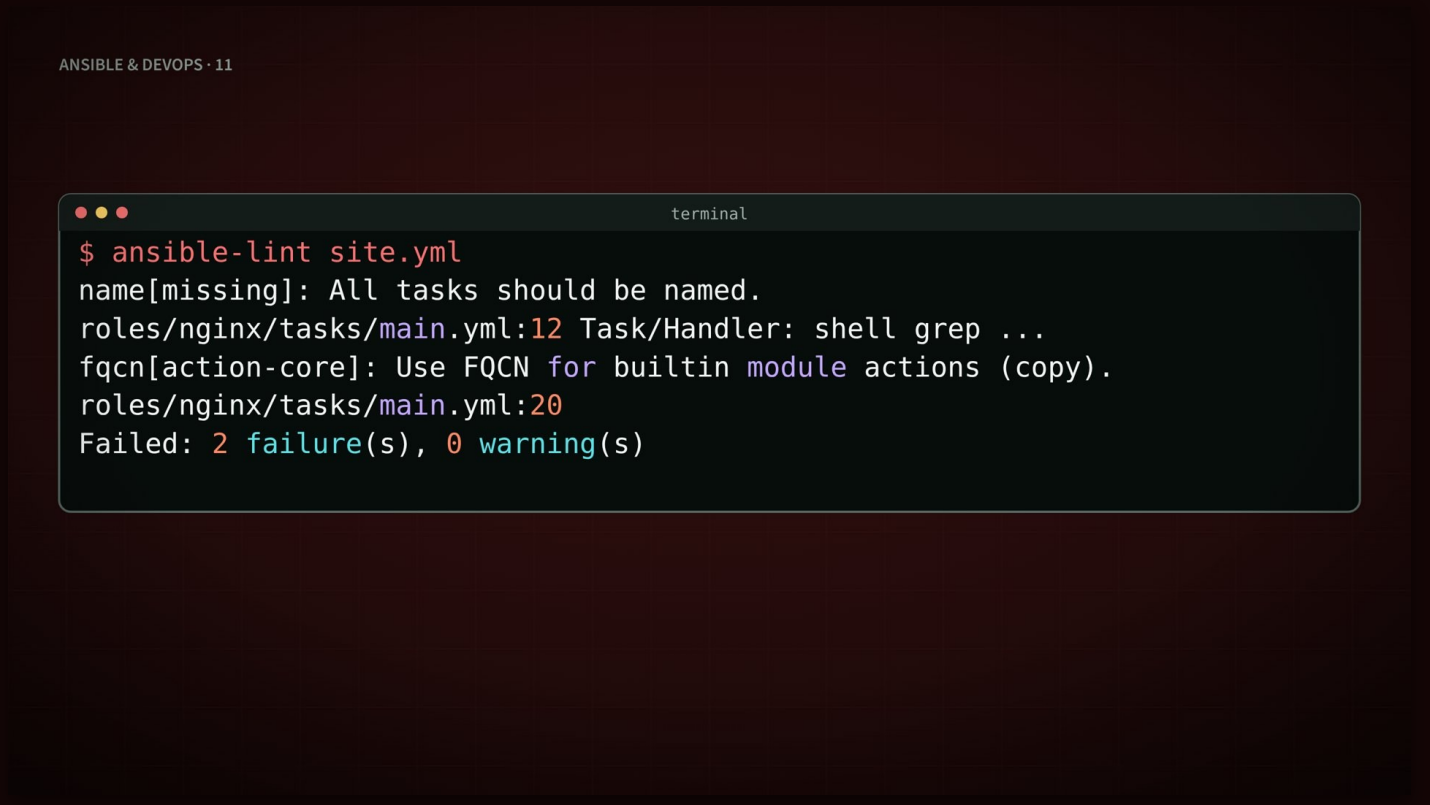



۱۱.۰۱

# ansible-lint

ansible-lint، playbook ها را در برابر بهترین روش‌ها بررسی می‌کند: نام‌های کم، نام‌های کوتاه ...

yamllint سبک YAML را بررسی می‌کند. هر دو را روی هر commit اجرا کنید.




۱۱.۰۲

# آزمایش role ها با Molecule

Molecule یک role را در container ها یا ماشین‌های مجازی دورانداختنی آزمایش می‌کند. instance ...

یک role که Molecule را روی Rocky و Ubuntu پاس کند آمادهٔ شریک شدن است.



terminal

```
$ molecule test
INFO  create      ✓  rocky9, ubuntu24
INFO  converge    ✓  changed=6
INFO  idempotence ✓  changed=0
INFO  verify      ✓  nginx answering on :80
INFO  destroy     ✓
```

۱۱.۰۳

# یک pipeline نوع CI/CD

یک pipeline روی هر push خودکار اجرا می‌شود: test، lint، بعد deploy به staging، و بعد از ...

کلید SSH و رمز vault را به عنوان secret های pipeline نگه دارید، هرگز در repository. ...



.github/workflows/deploy.yml

```yaml
on: { push: { branches: [main] } }
jobs:
  deploy:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4
      - run: pip install ansible ansible-lint
      - run: ansible-lint
      - run: echo "$VAULT_PASS" > .vault_pass
        env: { VAULT_PASS: "${{ secrets.VAULT_PASS }}" }
      - run: ansible-playbook -i inventories/staging site.yml --vault-password-file .vault_pass
```

۱۱.۰۴

# محیط‌ها

inventory های جداگانه برای development، staging و production به همان playbook ها اجازه ...

تغییرها همیشه در یک جهت جریان دارند: dev، بعد staging، بعد production. هرگز production را ...

۱۱.۰۵



terminal

```
inventories/
├── dev/
│   ├── hosts.ini
│   └── group_vars/all.yml      # app_replicas: 1
├── staging/
│   └── hosts.ini
└── production/
    ├── hosts.ini
    └── group_vars/all.yml      # app_replicas: 4
```

یک کورس رایگان از MyWebUniversity

# Ansible و DevOps

درس ۱۲

## پروژه: deploy کردن یک اپ web، و قدم‌های بعدی




یک deployment کامل چندلایه، به‌روزرسانی‌های rolling، AWX و Automation Platform، و مسیر ش

# پروژه ۵

کاربران — https://school.edu

lb1 — role: haproxy

web1، web2 — role: nginx + app




۱۲.۰۱

# playbook سایت

site.yml، baseline مشترک را همه جا اعمال می‌کند، بعد دیتابیس، بعد لایهٔ web، بعد load ...

یک دستور کل محیط را از server های خالی در چند دقیقه می‌سازد.



site.yml

```yaml
- hosts: all
  become: true
  roles: [common, firewall]

- hosts: db
  become: true
  roles: [postgres]

- hosts: web
  become: true
  roles: [nginx, school_app]

- hosts: lb
  become: true
  roles: [haproxy]
```

۱۲.۰۲

# به‌روزرسانی rolling بدون توقف

serial یک host را در یک وقت به‌روز می‌کند.
پیش از هر host، delegate_to آن را از load ...

max_fail_percentage اگر چیزی خراب شود
rollout را متوقف می‌کند، تا کاربران هرگز یک
قطعی نبینند.



rolling.yml

```yaml
- hosts: web
  serial: 1
  max_fail_percentage: 0
  pre_tasks:
    - name: Drain from load balancer
      community.general.haproxy:
        state: disabled
        host: "{{ inventory_hostname }}"
      delegate_to: lb1
  roles: [school_app]
  post_tasks:
    - name: Wait for health check
      ansible.builtin.uri: { url: "http://{{ inventory_hostname }}/health" }
    - name: Return to load balancer
      community.general.haproxy: { state: enabled, host: "{{ inventory_hostname }}" }
      delegate_to: lb1
```

۱۲.۰۳

# Ansible Automation Platform و AWX

playbook + inventory — ها job template
ها + credential

RBAC · زمان‌بندی‌ها — چه کسی چه چیزی را کی اجرا کند

Ansible — ها execution environment
container داخل



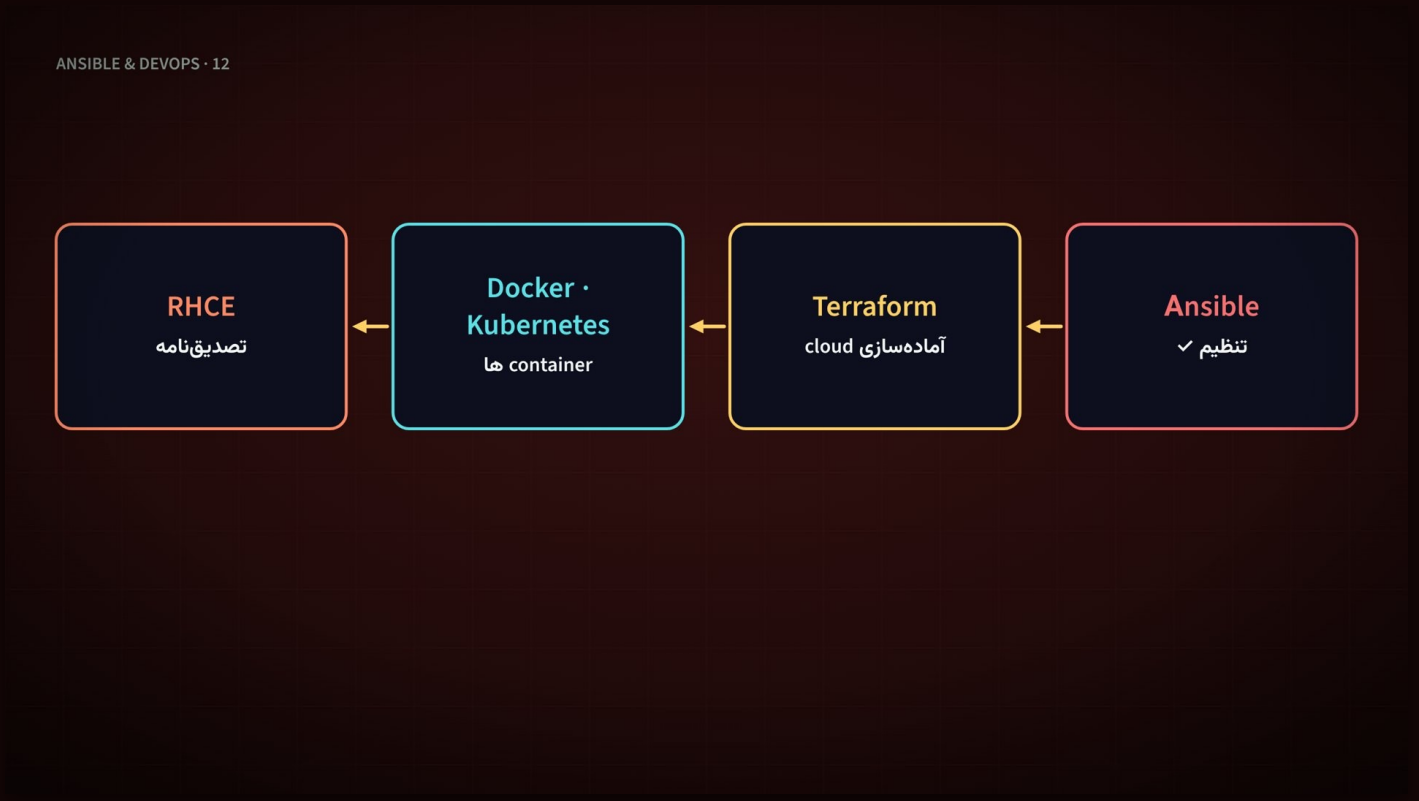

۱۲.۰۴

# مسیر DevOps شما

✓ تنظیم — Ansible

آماده‌سازی cloud — Terraform

ها Docker · Kubernetes — container




۱۲.۰۵

تشکر از توجه شما

# MyWebUniversity.com

@MyWebUniversity-Dari همه‌را یاد بگیرید

آموزش رایگان تکنالوژی از سال ۲۰۰۰ — انگلیسی و دری