HomeWork

DevOps

HW_L5_03

1. فایل تمرین را در پنل خود آپلود کنید.
2. title فایل تمرین به صورت (نام تمرین+نام و نام خانوادگی) به انگلیسی باشد.

   نمونه: AmirMohammadMohammadi_kubernetes_02
3. در صورتی که سوال و یا ابهامی دارید در گروه چت تلگرامی بپرسید.

استقرار خودکار اپلیکیشن وب با Docker و Ansible

هدف پروژه

در این پروژه شما باید یک اپلیکیشن وب را از GitHub clone کنید، برای آن Dockerfile و docker-compose.yml بنویسید، روی یک سرور deploy کنید، با Nginx در معرض دسترسی قرار دهید، SSL/TLS اضافه کنید، و در نهایت تمام این فرآیند را با Ansible خودکارسازی کنید.

این پروژه تمام مهارت‌هایی که یاد گرفته‌اید را در یک سناریوی واقعی به کار می‌گیرد.

مراحل پروژه

1- آماده سازی محیط

1- ایجاد یا اجاره سرور

الزامات:

شما باید یک سرور Ubuntu 22.04 آماده کنید. دو گزینه دارید:

گزینه 1: استفاده از VirtualBox یاVMware

- یک VM با Ubuntu 22.04 Server ایجاد کنید
- حداقل 2 GB RAMو 20 GB Disk Spaceاختصاص دهید
- Networkرا به گونه‌ای تنظیم کنید که از سیستم محلی قابل دسترسی باشد
- IP Addressسرور را یادداشت کنید

گزینه 2: اجارهVPS

- از یک سرویس‌دهنده VPS یک سرور Ubuntu 22.04 اجاره کنید
- حداقل 2 GB RAMو 20 GB Disk Spaceانتخاب کنید
- IP Addressو credentials را یادداشت کنید

خروجی مورد نیاز:

- فایل 01 environment/server_info.md_شامل:
  - IP Addressسرور
  - Usernameبرای اتصال

○ روش دسترسیSSH

○ مشخصات سیستم(RAM ، Disk، CPU)

1- اتصال و بررسی سرور

الزامات:

1. به سرور با SSH متصل شوید
2. اطلاعات سیستم را جمع‌آوری کنید:

○ Kernel version

○ OS version

○ Disk usage

○ Memory usage

○ Network interfaces

Hints:

● از دستوراتuname ، lsb_release، df، freeاستفاده کنید

● خروجی را در فایل ذخیره کنید

خروجی مورد نیاز:

- فایل environment/server_connection.txt_01 شامل خروجی دستورات بررسی سیستم

2- آماده‌سازی سرور با Ansible

1- نصب و پیکربندی Ansible

دانشکار

2.2 ایجاد Inventory

الزامات:

1) یک inventory file ایجاد کنید
2) سرور خود را در inventory تعریف کنید
3) Variables مناسب را تنظیم کنید
4) اتصال Ansible را تست کنید

Hints:

1) از format INI یا YAML می‌توانید استفاده کنید
2) Python interpreter را مشخص کنید
3) از ansible -m ping برای تست استفاده کنید

خروجی مورد نیاز:

1) فایل 02_ansible_setup/inventory
2) فایل 02_ansible_setup/ping_test.txt شامل خروجی ping
3) فایل 02_ansible_setup/facts.txt شامل خروجی ansible -m setup

---

2.3 Playbook برای آماده‌سازی سرور

الزامات:

یک playbook بنویسید که سرور را برای deployment آماده کند. این playbook باید:

1) سیستم را update کند
2) Package های ضروری را نصب کند:
   - curl، wget، git، vim، htop
   - ufw برای firewall
3) Docker و docker-compose را نصب کند

4) Docker service را start و enable کند
5) کاربر را به docker group اضافه کند
6) Nginx را نصب و راه‌اندازی کند
7) Firewall را پیکربندی کند:
   - Port 22 برای SSH
   - Port 80 برای HTTP
   - Port 443 برای HTTPS

Hints:

- از apt module برای نصب packages استفاده کنید
- از systemd module برای مدیریت services استفاده کنید
- از user module برای اضافه کردن user به group استفاده کنید
- از ufw module برای firewall استفاده کنید
- از become: yes برای privilege escalation استفاده کنید

خروجی مورد نیاز:

- فایل 02_ansible_setup/server_setup.yml
- فایل 02_ansible_setup/playbook_output.txt شامل خروجی اجرا
- فایل 02_ansible_setup/verification.txt شامل:
   - نسخه Docker
   - نسخه Nginx
   - وضعیت services
   - وضعیت firewall

مرحله 3: انتخاب و Clone پروژه از GitHub

3.1 جستجو و انتخاب پروژه

الزامات:

یک پروژه مناسب از GitHub پیدا کنید که:

- شامل یک web application باشد
- از database استفاده کند
- برای Docker practice مناسب باشد
- ساده و قابل فهم باشد

پیشنهادات:

می‌توانید از این منابع استفاده کنید:

- docker/awesome-compose repository
- جستجو: "docker compose flask mysql example"
- جستجو: "docker compose nginx flask mysql"
- جستجو: "docker example voting app"

Hints:

- پروژه‌های docker/awesome-compose معمولاً ساده و مناسب هستند
- پروژه باید شامل frontend و backend باشد
- پروژه باید از database استفاده کند

3.2 Clone و بررسی پروژه

الزامات:

1) پروژه را clone کنید
2) ساختار پروژه را بررسی کنید
3) فایل‌های موجود را لیست کنید
4) اگر Dockerfile یا docker-compose.yml وجود دارد، بررسی کنید
5) Requirements و dependencies را شناسایی کنید

خروجی مورد نیاز:

- فایل 03_project_clone/project_structure.txt شامل:
  - ساختار کامل پوشه‌ها
  - لیست فایل‌های مهم

- فایل 03_project_clone/project_info.md شامل:
  - URL پروژه
  - توضیح پروژه
  - تکنولوژی‌های استفاده شده
  - Dependencies
  - Ports مورد استفاده

مرحله 4: نوشتن Dockerfile و docker-compose.yml

4.1 بررسی و تحلیل پروژه

الزامات:

1) بررسی کنید که آیا پروژه Dockerfile دارد یا نه
2) بررسی کنید که آیا docker-compose.yml دارد یا نه
3) اگر وجود دارد، آن‌ها را تحلیل کنید
4) اگر وجود ندارد یا نیاز به اصلاح دارد، باید بنویسید

Hints:

- از cat یا less برای خواندن فایل‌ها استفاده کنید
- Requirements را شناسایی کنید
- Ports مورد نیاز را مشخص کنید
- Environment variables را شناسایی کنید

4.2 نوشتن Dockerfile

الزامات:

برای application خود یک Dockerfile بنویسید که:

1) Base image مناسب را انتخاب کند
2) Working directory را تنظیم کند
3) Dependencies را نصب کند
4) Application code را کپی کند
5) Port را expose کند
6) Command برای اجرای application را تعریف کند

Hints:

- از official images استفاده کنید
- از multi-stage builds برای بهینه‌سازی استفاده کنید
- از .dockerignore استفاده کنید
- Layers را بهینه کنید
- Security best practices را رعایت کنید

خروجی مورد نیاز:

- فایل 04_docker/Dockerfile
- فایل 04_docker/.dockerignore در صورت نیاز
- فایل 04_docker/dockerfile_explanation.md شامل توضیح هر خط

4.3 نوشتن docker-compose.yml

الزامات:

یک docker-compose.yml بنویسید که:

1) Web application service را تعریف کند
2) Database service را تعریف کند

3) Network مناسب را تنظیم کند
4) Volumes برای persistence تعریف کند
5) Environment variables را مدیریت کند
6) Port mapping را انجام دهد
7) Dependencies بین services را تنظیم کند
8) Restart policies را تنظیم کند

Hints:

- از 3.8 version یا بالاتر استفاده کنید
- از named volumes برای database استفاده کنید
- از depends_on برای ترتیب اجرا استفاده کنید
- Health checks را اضافه کنید
- Environment variables را از فایل یا inline تعریف کنید

خروجی مورد نیاز:

- فایل 04_docker/docker-compose.yml
- فایل 04_docker/compose_explanation.md شامل توضیح configuration

---

4.4 تست Build محلی

الزامات:

1) Images را build کنید
2) Containers را run کنید
3) Application را تست کنید
4) Logs را بررسی کنید
5) اگر مشکلی بود، رفع کنید

Hints:

- از docker-compose build استفاده کنید

- از docker-compose up -d برای run در background استفاده کنید
- از docker-compose logs برای بررسی logs استفاده کنید
- از curl یا browser برای تست استفاده کنید

خروجی مورد نیاز:

- فایل 04_docker/build_log.txt شامل خروجی build
- فایل 04_docker/container_status.txt شامل docker-compose ps
- فایل 04_docker/test_results.txt شامل نتایج تست application

مرحله 5: Deploy روی سرور

5.1 انتقال فایل‌ها به سرور

الزامات:

فایل‌های پروژه را به سرور منتقل کنید. می‌توانید از روش‌های مختلف استفاده کنید:

1) استفاده از scp
2) استفاده از rsync
3) استفاده از git clone روی سرور
4) استفاده از Ansible copy module

Hints:

- ساختار پوشه‌ها را حفظ کنید
- Permissions را در نظر بگیرید
- فایل‌های حساس را secure نگه دارید

5.2 Build و Run روی سرور

الزامات:

1) به سرور متصل شوید
2) به پوشه پروژه بروید
3) Images را build کنید
4) Containers را run کنید
5) Status را بررسی کنید
6) Logs را بررسی کنید

Hints:

- از docker-compose build استفاده کنید
- از docker-compose up -d استفاده کنید
- از docker-compose ps برای status استفاده کنید
- از docker-compose logs -f برای logs استفاده کنید

خروجی مورد نیاز:

- فایل 05_deployment/deploy_log.txt شامل خروجی deploy
- فایل 05_deployment/container_status.txt شامل وضعیت containers
- فایل 05_deployment/container_logs.txt شامل logs مهم

5.3 تست دسترسی

الزامات:

1) از روی سرور application را تست کنید
2) از سیستم محلی application را تست کنید
3) مطمئن شوید که application درست کار می‌کند

Hints:

- از curl http://localhost:PORT استفاده کنید
- از curl http://SERVER_IP:PORT از سیستم محلی استفاده کنید
- Response را بررسی کنید

خروجی مورد نیاز:

- فایل 05_deployment/test_results.txt شامل:
  - نتایج تست از سرور
  - نتایج تست از سیستم محلی
  - Screenshot یا response sample

مرحله 6: پیکربندی Nginx

6.1 طراحی Configuration

الزامات:

قبل از نوشتن configuration، طراحی کنید:

1) Domain name را انتخاب کنید (مثلاً myapp.local)
2) Port application را مشخص کنید
3) Proxy settings را طراحی کنید
4) Headers مورد نیاز را مشخص کنید

Hints:

- از reverse proxy pattern استفاده کنید
- Headers مهم: Host، X-Real-IP، X-Forwarded-For، X-Forwarded-Proto
- از proxy_pass استفاده کنید

6.2 ایجاد Nginx Configuration

الزامات:

یک Nginx configuration file بنویسید که:

1) روی port 80 listen کند
2) Domain name را handle کند
3) Request ها را به application proxy کند
4) Headers مناسب را set کند
5) Error handling داشته باشد

Hints:

- Configuration را در /etc/nginx/sites-available/ قرار دهید
- از proxy_pass برای forwarding استفاده کنید
- از proxy_set_header برای headers استفاده کنید
- از location blocks استفاده کنید

خروجی مورد نیاز:

- فایل 06_nginx/nginx_config.txt شامل configuration
- فایل 06_nginx/nginx_explanation.md شامل توضیح configuration

---

6.3 فعال‌سازی Configuration

الزامات:

1) Configuration را در sites-available قرار دهید
2) Symbolic link در sites-enabled ایجاد کنید
3) Default site را disable کنید
4) Configuration را test کنید
5) Nginx را reload کنید

Hints:

- از ln -s برای symbolic link استفاده کنید
- از nginx -t برای test استفاده کنید
- از systemctl reload nginx برای reload استفاده کنید

6.4 تنظیم /etc/hosts

الزامات:

1) روی سیستم محلی /etc/hosts را ویرایش کنید
2) Domain name را به IP سرور map کنید
3) تغییرات را save کنید

Hints:

- از sudo nano /etc/hosts استفاده کنید
- Format: IP_ADDRESS domain_name
- بعد از DNS cache، save را clear کنید

خروجی مورد نیاز:

- فایل 06_nginx/hosts_file.txt شامل محتوای /etc/hosts

6.5 تست دسترسی

الزامات:

1) از سیستم محلی با domain name تست کنید
2) مطمئن شوید که Nginx درست proxy می‌کند
3) Response را بررسی کنید

Hints:

- از curl http://domain_name استفاده کنید
- از browser استفاده کنید
- Headers را بررسی کنید

خروجی مورد نیاز:

- فایل 06_nginx/test_results.txt شامل نتایج تست

مرحله 7: پیکربندی SSL/TLS

7.1 تولید Self-Signed Certificate

الزامات:

1) یک directory برای certificates ایجاد کنید
2) Private key تولید کنید
3) Certificate request ایجاد کنید
4) Self-signed certificate تولید کنید
5) Permissions را تنظیم کنید

Hints:

- از openssl genrsa برای private key استفاده کنید
- از openssl req برای certificate استفاده کنید
- از x509- برای self-signed استفاده کنید
- Permissions را secure نگه دارید

خروجی مورد نیاز:

- فایل 07_ssl/certificate_info.txt شامل:
  - مسیر certificate files
  - اطلاعات certificate
  - Expiry date

7.2 به‌روزرسانی Nginx Configuration

الزامات:

Nginx configuration را به‌روزرسانی کنید تا:

1) HTTP requests را به HTTPS redirect کند
2) روی port 443 listen کند
3) SSL certificate و key را load کند
4) SSL protocols و ciphers را تنظیم کند
5) Application را proxy کند

Hints:

- از return 301 برای redirect استفاده کنید
- از ssl_certificate و ssl_certificate_key استفاده کنید
- از ssl_protocols و ssl_ciphers استفاده کنید
- Security best practices را رعایت کنید

خروجی مورد نیاز:

- فایل 07_ssl/nginx_ssl_config.txt شامل configuration کامل

7.3 Reload و تست

الزامات:

1) Configuration را test کنید
2) Nginx را reload کنید
3) HTTP redirect را تست کنید
4) HTTPS connection را تست کنید

Hints:

- از nginx -t برای test استفاده کنید
- از curl -L برای follow redirect استفاده کنید
- از curl -k برای ignore certificate warning استفاده کنید

خروجی مورد نیاز:

- فایل 07_ssl/test_results.txt شامل:
  - نتایج تست HTTP redirect
  - نتایج تست HTTPS
  - Certificate information

مرحله 8: خودکارسازی با Ansible

8.1 طراحی Automation

الزامات:

قبل از نوشتن playbooks، طراحی کنید:

1) چه task هایی باید automate شوند؟
2) چه order باید داشته باشند؟
3) چه variables نیاز است؟
4) چه handlers نیاز است؟
5) چه templates نیاز است؟

Hints:

- Deployment process را break down کنید
- Dependencies را شناسایی کنید
- Error handling را در نظر بگیرید

8.1 Playbook برای Application Deployment

الزامات:

یک playbook بنویسید که:

1) Application directory را ایجاد کند
2) فایل‌های پروژه را کپی کند:
   - Dockerfile
   - docker-compose.yml
   - Application code

1) Docker images را build کند
2) Containers را run کند
3) Status را verify کند

Hints:

- از copy module برای فایل‌ها استفاده کنید
- از docker_compose module برای Docker operations استفاده کنید
- از wait_for برای wait کردن استفاده کنید
- از register و debug برای verification استفاده کنید

خروجی مورد نیاز:

- فایل 08_ansible_automation/deploy_app.yml
- فایل 08_ansible_automation/app_vars.yml در صورت نیاز

8.2 Playbook برای Nginx Configuration

الزامات:

یک playbook بنویسید که:

1) SSL directory را ایجاد کند
2) SSL certificate را generate کند
3) Nginx configuration template را deploy کند
4) Site را enable کند
5) Default site را disable کند

6) Nginx را reload کند

Hints:

- از openssl_certificate module استفاده کنید
- از template module برای configuration استفاده کنید
- از file module برای symbolic links استفاده کنید
- از handlers برای reload استفاده کنید

خروجی مورد نیاز:

- فایل 08_ansible_automation/deploy_nginx.yml
- فایل 08_ansible_automation/templates/nginx.conf.j2
- فایل 08_ansible_automation/nginx_vars.yml در صورت نیاز

8.4 ایجاد Main Playbook

الزامات:

یک main playbook ایجاد کنید که:

1) تمام playbooks را import کند
2) Order صحیح را حفظ کند
3) Reusable باشد

Hints:

- از import_playbook استفاده کنید
- از site.yml یا main.yml استفاده کنید

خروجی مورد نیاز:

- فایل 08_ansible_automation/site.yml

8.5 اجرا و تست

الزامات:

1) Playbooks را اجرا کنید
2) Output را بررسی کنید
3) Application را verify کنید
4) اگر مشکلی بود، رفع کنید

Hints:

- از check-- برای dry-run استفاده کنید
- از v- برای verbose output استفاده کنید
- از diff-- برای مشاهده تغییرات استفاده کنید

خروجی مورد نیاز:

- فایل 08_ansible_automation/playbook_output.txt شامل خروجی اجرا
- فایل 08_ansible_automation/verification.txt شامل نتایج verification

مرحله 9: مستندسازی

9.1 README.md

الزامات:

یک README.md کامل بنویسید که شامل:

1) توضیح پروژه
2) Prerequisites
3) Installation instructions
4) Usage instructions

5) Project structure
6) Configuration
7) Troubleshooting

خروجی مورد نیاز:

- فایل README.md در root پروژه

Architecture Documentation 9.2

الزامات:

یک document بنویسید که:

1) Architecture را توضیح دهد
2) Components را معرفی کند
3) Flow را نشان دهد
4) Diagrams داشته باشد

Hints:

- از text-based diagrams استفاده کنید
- از ASCII art استفاده کنید
- Flow را step-by-step نشان دهید

خروجی مورد نیاز:

- فایل 09_documentation/architecture.md

Deployment Guide 9.3

الزامات:

یک deployment guide بنویسید که:

1) Step-by-step instructions داشته باشد
2) Commands را شامل شود
3) Expected outputs را نشان دهد
4) Common issues را cover کند

خروجی مورد نیاز:

- فایل 09_documentation/deployment_guide.md

Troubleshooting Guide 9.4

الزامات:

یک troubleshooting guide بنویسید که:

1) مشکلات رایج را لیست کند
2) علت هر مشکل را توضیح دهد
3) راه‌حل‌ها را ارائه دهد

خروجی مورد نیاز:

- فایل 09_documentation/troubleshooting.md

مرحله 10: تحویل نهایی

Git Repository 10.1

الزامات:

1) Git repository initialize کنید

2) تمام فایل‌ها را add کنید
3) Meaningful commits انجام دهید
4) Repository را به GitHub/GitLab push کنید

Hints:

- از gitignore. استفاده کنید
- از meaningful commit messages استفاده کنید
- از branching strategy استفاده کنید

خروجی مورد نیاز:

- Git repository با تمام commits
- فایل 10_delivery/git_history.txt شامل git log --oneline –graph

---

Project Summary 10.2

الزامات:

یک summary بنویسید که شامل:

1) Challenges مواجه شده
2) Solutions پیاده‌سازی شده
3) Lessons learned
4) Improvements پیشنهادی

خروجی مورد نیاز:

- فایل 10_delivery/project_summary.md

Final Structure 10.3

الزامات:

ساختار نهایی پروژه را document کنید.

خروجی مورد نیاز:

- فایل 10_delivery/final_structure.txt شامل tree یا find output

---

Team Contribution 10.4

الزامات:

Contribution هر عضو تیم را document کنید.

خروجی مورد نیاز:

- فایل 10_delivery/team_contribution.md شامل:
  - تقسیم کار
  - Contribution هر عضو
  - Challenges هر عضو

نکات مهم

کار تیمی

- تقسیم کار را به صورت منطقی انجام دهید
- یک نفر روی Docker و دیگری روی Ansible کار کند
- Code review انجام دهید
- Regular commits انجام دهید

Testing

- هر مرحله را قبل از ادامه تست کنید
- از logs برای debugging استفاده کنید

- از dry-run برای Ansible استفاده کنید

Security

- Passwords را در variables file قرار دهید
- از gitignore. برای فایل‌های حساس استفاده کنید
- SSL certificates را secure نگه دارید

Best Practices

- Code را clean و readable نگه دارید
- Comments را meaningful بنویسید
- Documentation را به روز نگه دارید

منابع مفید

- Docker Documentation: https://docs.docker.com/
- Docker Compose Documentation: https://docs.docker.com/compose/
- Ansible Documentation: https://docs.ansible.com/
- Nginx Documentation: https://nginx.org/en/docs/
- OpenSSL Documentation: https://www.openssl.org/docs/

مشکلات رایج

مشکل: *Docker build fails*

چه باید کرد:

- Logs را بررسی کنید
- Dockerfile syntax را check کنید
- Dependencies را verify کنید
- Base image را بررسی کنید

مشکل: *Ansible connection fails*

چه باید کرد:

- SSH connection را test کنید
- Inventory را بررسی کنید
- Permissions را check کنید
- Python interpreter را verify کنید

مشکل: *Nginx 502 Bad Gateway*

چه باید کرد:

- Container status را check کنید
- Port mapping را verify کنید
- Proxy configuration را بررسی کنید
- Logs را check کنید

مشکل: *SSL certificate errors*

چه باید کرد:

- Certificate path را verify کنید
- Permissions را check کنید
- Certificate validity را بررسی کنید
- Nginx configuration را test کنید

موفق باشید! 🚀 این پروژه فرصتی عالی برای به کارگیری تمام مهارت‌های یادگرفته شده است. از آن لذت ببرید و بهترین کار را انجام دهید!