

پروژه نهایی بوت کمپ DevOps

شرح پروژه

پیاده سازی استک کامل DevOps برای CI/CD اپلیکیشن به همراه مانیتورینگ و logging.

مرحله اول - نصب پیش نیازها و ستاپ اولیه

نصب موارد زیر از طریق Ansible:

- Docker
- [Kind](#) cluster with one controller and two worker nodes
- Nginx
- [node_exporter](#)
- Prometheus + Grafana using Docker Compose
- Docker image registry using Docker Compose
- GitLab Runner

مرحله دوم - Firewall

- تنظیم فایروال از طریق iptables به نحوی که node_exporter فقط از طریق داکر در دسترس باشد.
- تنظیم فایروال به نحوی که پورت ssh برای 10.10.10.1 و 10.10.10.2 و 10.10.10.3 و 10.10.10.4 در دسترس نباشد. (از طریق ipset و iptables)
- ذخیره iptables به نحوی که بعد از ری بوت rule ها حذف نشوند.

مرحله سوم - Dockerize

داکرایز کردن اپلیکیشن. رعایت کردن best practice ها برای کاهش حجم image لطفا صورت گیرد.
اپلیکیشن مورد نظر در لینک زیر قرار دارد:

<https://github.com/faridz88/KelaasorDevOpsBootcamp/tree/main/FinalProject>

مرحله چهارم - CI/CD for dev environment

- نوشتن جاب GitLab CI به صورتی که بعد از دریافت Merge Request روی برنچ dev بیلد و push به image registry صورت گیرد. دیپلوی از طریق docker compose روی پورت ۸۰۸۱ انجام شود. تگ ایمج push شده باید برابر با short commit SHA باشد.

مرحله پنجم - CI/CD for stage environment

- بعد از هر کامیت و یا merge request بر روی main branch بیلد و push به image registry صورت گیرد. تگ این ایمج برابر با مقدار کلید version در appinfo.json باشد. از jq, bash می‌توانید برای این مورد استفاده نمایید. دیپلوی به روش blue green و zero downtime بر روی پورت‌های ۸۰۸۲ و ۸۰۸۳ انجام شود. از bash script می‌توانید برای switch استفاده نمایید. در هنگام دیپلوی نباید در سرویس‌دهی اپلیکیشن downtime ایجاد شود. از nginx نیز می‌توانی برای ha, loadbalance بین blue و green استفاده نمایید.

مرحله ششم - GitOps

- ایجاد kubernetes manifest های مربوطه (Deployment, Service, ...) و قرار دادن آن در git repo برای GitOps CD. روی هر کدام از worker ها دو نسخه از اپلیکیشن باید deploy شود. (مجموعاً 4 replica)
- نصب و اتصال ArgoCD به کلاستر kind و GitLab
- تنظیم CI pipeline برای Production Environment به نحوی که بعد از تگ زدن روی هر کامیت، ایمج متناظر آن بر روی کلاستر kubernetes دیپلوی شود.
- ایجاد ReverseProxy از طریق nginx برای دامنه prod.DOMAIN.TLD به نحوی که بین worker ها HA & LoadBalancing صورت گیرد.

موارد اختیاری

- نصب loki و نگهداری لاگ‌های nginx بر روی آن
- نصب Alert Manager و تنظیم ارسال Alert از طریق تلگرام یا ایمیل در صورت بالا رفتن لود، پر شدن دیسک اصلی، پر شدن رم سرور
- نصب minio و تنظیم نگهداری لاگ‌های loki بر روی آن.
- ایجاد و یا آپدیت changelog.txt در git repo root dir به صورت خودکار بر اساس متن کامیت‌ها (با فرض رعایت conventional commits) در dev pipeline
- موارد DevSecOps و تست‌های امنیتی
- تنظیم بکاپ‌ها و نوشتن Disaster Recovery Plan - DRP

* لطفا داکيومنت مواد صورت گرفته و ساير فايل‌هاى ايجاد شده را به git اضافه نموده و لينک repository را به عنوان خروجی نهایی تحویل نمایید.