Úvod

Účelom je splniť nasledujúce zadanie, ktoré simuluje prácu u zákazníka. V dokumente sú špecifikované základné informácie. Pokiaľ potrebujete doplňujúce informácie, chovajte sa tak akoby ste pracovali u zákazníka a danú informáciu si vyžiadajte (prostredníctvom email-u/microsoft teams-u). Hlavným účelom zadania je overiť si samostatnosť a spôsob riešenia problému.

Kontakt

Microsoft Teams

Email:

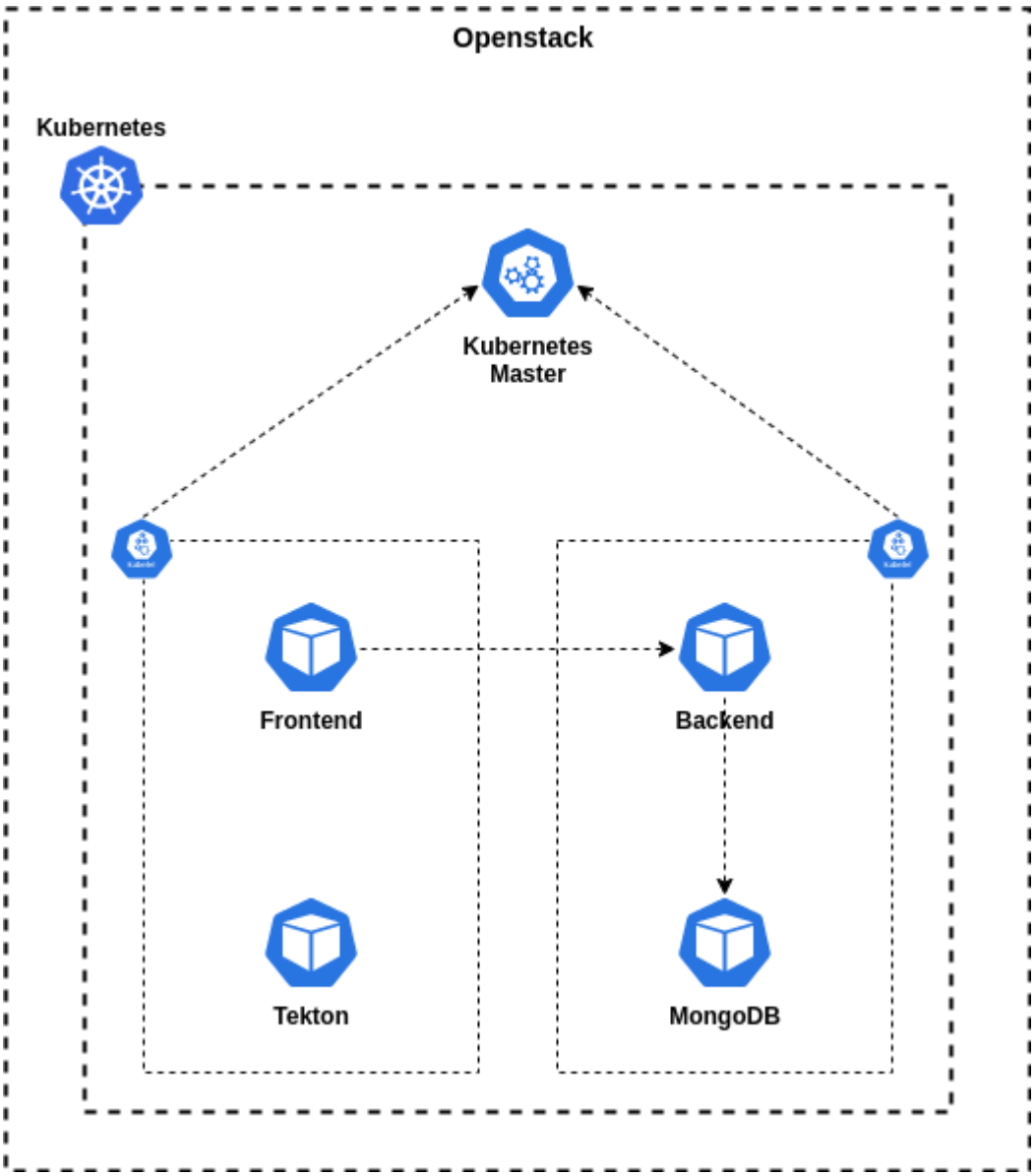
- daniel.rajcan@unicorn.com
- michal.okresa@unicorn.com
- marek.moravcik@fri.uniza.sk - v prípade nefunkčného Openstack prostredia

Zadanie

Zákazník chce vytvoriť Kubernetes cluster na privátnom/komunitnom cloude (Openstack) Katedry Informačných Sietí v Žiline. Hlavnou zákazníckou požiadavkou je, aby všetko bolo pripravené v duchu DevOps, pričom kladie velký dôraz na automatizáciu a znuvupoužitelnosť.

Ďalej zákazník kladie dôraz na:

- Všetko je realizované v prostredí Openstack: https://158.193.153.3/dashboard
- Hlavný IaC nástroj je Terraform: https://www.terraform.io/
- Všetky zdrojové kódy sú uložené a verzionované v GIT repozitáry: https://github.com/
- Docker images sú uložené v Docker Hub registry: https://hub.docker.com/
- Systematická menná konvencia pre aplikácie a kubernetes resources
- Bezpečnosť senzitívnych dát (aj mená, heslá a tokeny sa považujú za ciltivé dáta)
- Vysoká dostupnosť
- Úplne sa vyhnúť manuálnej konfigurácii.
- Dokumentácia vypracovaného zadania (stačí použiť README file v GIT repository)



GIT repository

V nasledujúcich repozitárov nájdete zdrojové kódy aplikácií:

- Helm - https://github.com/michaello1/zauni-zadanie-helm
- Appbackend - https://github.com/michaello1/zauni-zadanie-appbackend
- Appfrontend - https://github.com/michaello1/zauni-zadanie-appfrontend

Nasledujúci repozitár môžete použiť ako inšpiráciu pre písanie Terraform skriptov:

- https://github.com/drajcan-nephthys/kis-onpk - tento repozitár berte ako pomôcku, neodporúčam odovzdávať zadanie, ktoré bude len kópia už vytvorených skriptov

Pre účely vytvorenia požadovanej architektúry si vytvorte vlastné Github repozitáre, ktorých obsah bude taktiež hodnotený po odovzdaní zadania.

Docker Hub repository

Zaregistrujte sa a vytvorte si vlastné docker hub repositáre ako úložisko pre vytvorené docker images.

Kubernetes Cluster

- Vytvorenie vlastného GIT repozitára, ktorý bude obsahovať automatizovanú inštaláciu a konfiguráciu Kubernetes cluster-a pomocou nástroju Terraform
- Inštalácia a konfigurácia Kubernetes cluster-a pomocou nástroju minikube https://minikube.sigs.k8s.io/docs/start/, ktorý je prevádzkovaný na jednej virtuálnej inštancii v prostredí Openstack
- Kubernetes cluster a potrebná infraštruktúra musí byť plne automatizovaná. Snažte sa vytvoriť generickú konfiguráciu tak aby bola znovupoužitelná.
- **Bonusové body:** Inštalácia a konfigurácia Kubernetes cluster-a pomocou nástroju kubeadm https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/create-cluster-kubeadm/, ktorý je prevádzkovaný na 3 virtuálnych inštanciách (1x master, 2x worker nodes)
- Pre uloženie secrets použite Kubernetes secrets.
- **Bonusové body:** Pre uloženie secrets použite Kubernetes Sealed secrets.

Docker

- **Vytvorenie Dockerfiles**
  - pre frontend časť aplikácie
    - Hints:
      - **Bonusové body:** Zvážte použitie multi-stage build-u v Dockerfile
      - Pre build použite `node:12-alpine` image a `yarn`
      - Pre ďalší stage použite `nginx:stable-alpine` image
      - Použite `COPY --from=build /app/env.sh /docker-entrypoint.d` a taktiež `COPY conf.d/default.conf /etc/nginx/conf.d`
      - Vytvorte tag s názvom **latest**

        - **Bonusové body:** vytvorte tag pre docker image pomocou semantickeho verziovania https://semver.org/
    - pre backend časť aplikácie
        - Hints:
            - **Bonusové body:** Zvážte použitie multi-stage build-u v Dockerfile
            - Pre build použite `golang:alpine` image
            - Aplikácia je napísaná v jazyku GO
            - Vytvorte tag s názvom **latest**
            - **Bonusové body:** vytvorte tag pre docker image pomocou semantickeho verziovania https://semver.org/

Helm charts

- **Vytvorenie Helm charts**
    - pre frontend časť aplikácie
        - Hints:
            - použite NodePort pre Service (po deploymente si skontrolujte `kubectl get services` a `minikube service list`)
            - `appVersion:"latest"` postačuje, alebo semantic versioning
            - Deployment a Service objekty sú dostatočné
            - Healt check nastavte na /
            - Aplikácia očakáva environment premennú: **REACT_APP_APIHOSTPORT**
    - pre backend časť aplikácie
        - Nezabudnite, že backend používa mongodb, ktoré musí byť zadefinované ako závislosť v Chart.yaml
        - Použite mongodb z https://charts.bitnami.com/bitnami
        - použite NodePort pre Service (po deploymente si skontrolujte `kubectl get services` a `minikube service list`)
        - `appVersion:"latest"` postačuje, alebo semantic versioning
        - Deployment a Service objekty sú dostatočné
        - Health check nastavte na /ok

        - Environment premenné, ktoré potrebujete správne nastaviť: **MONGO_CONN_STR** (napr.: mongodb://appbackend-mongodb:27017/dynamicky_nazov_db), **MONGO_USERNAME**, **MONGO_PASSWORD**

CI/CD

- **Vytvorenie dvoch CI Tekton pipelines (prípadne Azure DevOps, GitHub Actions alebo GitLab):**
    - Build frontend časti aplikácie a push do vlastného Docker Hub repository
        - použite `git-clone` a `buildah` (alebo `kaniko`) tasky
        - nezabudnite na docker credentials v secrets
    - Build backend časti aplikácie a push do vlastného Docker Hub repository
        - použite `git-clone` a `buildah` (alebo `kaniko`) tasky
        - nezabudnite na docker credentials v secrets
- **Vytvorenie dvoch CD Tekton pipelines" (prípadne Azure DevOps, GitHub Actions alebo GitLab)**
    - Nasadenie frontend časti aplikácie prostredníctvom Helm chart
        - Použite `git-clone` a `helm-upgrade-from-source` tasky
    - Nasadenie backend časti aplikácie prostredníctvom Helm chart
        - Použite `git-clone` a `helm-upgrade-from-source` tasky
- Bonusové body: live prezentácia riešenia