Betriebsdokumentation

EasyNotes

|  |  |  |
| --- | --- | --- |
| **Datum** | **Version** | **Autor\*in** |
| 20.11.2023 | 1.0 | Nevio Di Gennaro |

Inhaltsverzeichnis

[1. Easy-Notes 2](#__RefHeading___Toc100_1702162029)

[2. Infrastruktur 2](#__RefHeading___Toc108_1702162029)

[2.1. GitLab 2](#__RefHeading___Toc112_1702162029)

[2.2. Azure Services 2](#__RefHeading___Toc110_1702162029)

[2.2.1. Ressourcengruppe 2](#__RefHeading___Toc122_1702162029)

[2.2.2. Virtuelles Netzwerk 2](#__RefHeading___Toc124_1702162029)

[2.2.3. Netzwerksicherheitsgruppe 2](#__RefHeading___Toc126_1702162029)

[2.2.4. Azure Kubernetes Service 2](#__RefHeading___Toc128_1702162029)

[2.2.5. Azure MySQL 2](#__RefHeading___Toc130_1702162029)

[2.3. Kubernetes 2](#__RefHeading___Toc114_1702162029)

[2.3.1. EasyNotes Datenbank einrichten 2](#__RefHeading___Toc2401_967934136)

[2.3.2. Kubernetes - Benötigte Controller / Operators 3](#__RefHeading___Toc2403_967934136)

[3. CI Pipeline 3](#__RefHeading___Toc2393_967934136)

[3.1. Übersicht 3](#__RefHeading___Toc2395_967934136)

[3.2. Konfiguration 3](#__RefHeading___Toc2397_967934136)

[3.2.1. GitLab Konfiguration 3](#__RefHeading___Toc2478_967934136)

[3.2.2. Dockerfile erstellen 3](#__RefHeading___Toc2399_967934136)

[3.2.3. .gitlab-ci.yml - Allgemeine Konfiguration 3](#__RefHeading___Toc2399_967934136_Copy_2_)

[3.2.4. .gitlab-ci.yml - Build-Stage 4](#__RefHeading___Toc2399_967934136_Copy_1)

[3.2.5. .gitlab-ci.yml - Test-Stage 4](#__RefHeading___Toc2399_967934136_Copy_2)

[3.2.6. .gitlab-ci.yml - Deploy-Stage 4](#__RefHeading___Toc2399_967934136_Copy_21)

[3.2.7. .gitlab-ci.yml - Updatecd-Stage 4](#__RefHeading___Toc2399_967934136_Copy_22)

[4. CD Konfiguration 4](#__RefHeading___Toc2411_967934136)

[4.1. Übersicht 4](#__RefHeading___Toc2413_967934136)

[4.2. Konfiguration 4](#__RefHeading___Toc2415_967934136)

[4.2.1. EasyNotes – Secrets 4](#__RefHeading___Toc2405_967934136)

[4.2.2. EasyNotes – Deployment 5](#__RefHeading___Toc2407_967934136)

[4.2.3. EasyNotes – Service 5](#__RefHeading___Toc2409_967934136)

[4.2.4. EasyNotes – Ingress-Controller 5](#__RefHeading___Toc2409_967934136_Copy_1)

[5. Testing 5](#__RefHeading___Toc116_1702162029)

[5.1. Testfälle 5](#__RefHeading___Toc118_1702162029)

[5.2. Testprotokoll 6](#__RefHeading___Toc2417_967934136)

# Easy-Notes

Easy-Notes ist eine Applikation zum Erfassen und Managen von Notizen. Sie ist sehr schlicht gehalten und macht seinen Job sehr effizient

# Infrastruktur

## GitLab

Verwaltung Quellcode

Container Registry

CI Pipeline

## Azure Services

### Ressourcengruppe

|  |  |  |
| --- | --- | --- |
| **Parameter / Option** | **Wert** | **Bemerkung** |
| Name | rgEasyNotes |  |
| Region | Schweiz West |  |

### Virtuelles Netzwerk

### Netzwerksicherheitsgruppe

### Azure Kubernetes Service

### Azure MySQL

## Kubernetes

Nginx zum Repo hinzufügen  
helm repo add ingress-nginx <https://kubernetes.github.io/ingress-nginx>

Ingress Controller zum Helm Repository hinzugühen  
helm repo add ingress-nginx https://kubernetes.github.io/ingress-nginx

Repo updaten

helm repo update

Nginx Ingress Controller instalieren im Nginx namespace  
kubectl create namespace ingress-nginx  
helm install nginx-ingress ingress-nginx/ingress-nginx --namespace ingress-nginx

### Kubernetes - Benötigte Controller / Operators

|  |  |
| --- | --- |
| **Aufgabe** | **Konfiguration / Befehle** |
| Zugriff für Cloud auf Client einrichten |  |
| Installation Helm | Als erstes, denn die Creditentials des AKS Clusters  az aks get-credentials --resource-group easynodes-dev --name easynodes-dev-aks |
| Installation sops |  |
| Installation ArgoCD | ArgoCD zum Helm Repo hinzufügen helm repo add argo <https://argoproj.github.io/argo-helm>  Repo updaten  helm repo update  ArgoCD im ArgoCD namespace installieren  kubectl create namespace argocd  helm install argo-cd argo/argo-cd --namespace argocd  Für Zugriff auf das ArgoCD UI leiten wir den ArgoCD port weiter kubectl port-forward svc/argo-cd-argocd-server -n argocd 8080:80 |

# CI Pipeline

## Übersicht

## Konfiguration

### GitLab Konfiguration

### Docker-Image erstellen

### .gitlab-ci.yml - Allgemeine Konfiguration

### .gitlab-ci.yml - Build-Stage

### .gitlab-ci.yml - Test-Stage

### .gitlab-ci.yml - Deploy-Stage

### .gitlab-ci.yml - Updatecd-Stage

# CD Konfiguration

## Übersicht

## Konfiguration

### EasyNotes Datenbank einrichten

Befehl zum Erstellen des Containers der Datenbank zusammen mit dem Dump

docker run --name mysql-container -p 3306:3306 -e MYSQL\_ROOT\_PASSWORD=root -e MYSQL\_DATABASE=easynotes -e MYSQL\_USER=dbuser -e MYSQL\_PASSWORD=dbpassword -d mysql

Befehl zum Dump ausführen im Container

docker exec -i mysql-container mysql -u dbuser -pdbpassword easynotes < /easy-notes-dump.sql

Befehl zum starten des Containers

docker start mysql-container

|  |  |
| --- | --- |
| **Aufgabe** | **Konfiguration / Befehle** |
| Mysqladmin Pod einrichten |  |
|  |  |
|  |  |
|  |  |

### EasyNotes – Secrets

|  |  |
| --- | --- |
| **Aufgabe** | **Konfiguration / Befehle** |
| Zugriff GitLab Container Registry |  |
|  |  |
|  |  |
|  |  |

### EasyNotes – Deployment

|  |  |
| --- | --- |
| **Aufgabe** | **Konfiguration / Befehle** |
|  |  |
|  |  |
|  |  |
|  |  |

### EasyNotes – Service

|  |  |
| --- | --- |
| **Aufgabe** | **Konfiguration / Befehle** |
|  |  |
|  |  |
|  |  |
|  |  |

### EasyNotes – Ingress-Controller

|  |  |
| --- | --- |
| **Aufgabe** | **Konfiguration / Befehle** |
|  |  |
|  |  |
|  |  |
|  |  |

# Testing

## Testfälle

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nr** | **Typ** | **Beschreibung** | **Testschritte** | **Erwartetes Resultat** | **Status** |
| 1.0 | Funktionalität | Die CI Pipeline sollte nach einer Quellcodeänderung im Git Repository gestartet werden und ohne Fehler durchlaufen. | 1. Git clone Projekt lokal 2. Ändere Versionsnummer 3. Git commit und push 4. Kontrolle CI-Pipeline in GitLab | Die Stages Build, Test und Deploy laufen durch und haben keine Fehler. Ein neues Docker-Image ist in der Container Registry gespeichert. | Pass or Fail |
| 2.0 | Funktionalität | Die CI Pipeline sollte nach der Änderung der Versionsnummer der EasyNotes-Anwendung korrekt durchlaufen. | 1. Ändere die Versionsnummer der EasyNotes-Anwendung im Quellcode.  2. Committe und pushe die Änderungen in das GitLab-Repository.  3. Überprüfe die Pipeline-Logs in GitLab. | Die CI-Pipeline sollte erfolgreich durchlaufen. Die Änderungen sollten korrekt übernommen und das Docker-Image erfolgreich gepusht werden. | Pass or Fail |
| 4.0 | |  | | --- | | Funktionalität |  |  | | --- | |  | | Die EasyNotes Applikation sollte erfolgreich mit curl -f getestet werden. | 1. Starte den Docker-Container mit der EasyNotes Applikation.  2. Führe den Befehl curl -f http://localhost:8080 aus. | Die Applikation antwortet ohne Fehler. Der HTTP-Statuscode sollte 200 OK sein. | Pass or Fail |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## Testprotokoll

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Datum / Uhrzeit** | **Nr** | **Status** | **Bemerkungen** | **Visum** |
| 20.03.2023 / 20:30 | 1.0 | Pass | Bla bla | tsa |