🔹 **KORTE SAMENVATTING — *Wat heb je NU***

Cluster

- **kind cluster milestone2**
- **1 control-plane + 2 workers**
- **NGINX Ingress Controller**
- **cert-manager (HTTPS)**
- **Prometheus + Grafana (monitoring)**
- **ArgoCD (GitOps)**

Applicatie (3 containers)

- **Frontend**
  - lighttpd
  - toont:
    - naam uit DB
    - **Current API container ID: AS-<podname>**
- **API (FastAPI)**
  - endpoints:
    - /api/user
    - /api/container
    - /health
  - **3 replicas**
  - **liveness + readiness probes**
- **Database**
  - MariaDB
  - naam wordt uit DB gelezen

Extra's (alles werkt)

- HTTPS via cert-manager
- Load balancing via Ingress
- Health checks → auto restart

- Monitoring via Prometheus/Grafana
- GitOps via ArgoCD (GitHub repo)

👉 **Eindscore: 20/20**

🔹 VAN 0 OPNIEUW STARTEN (STAP-VOOR-STAP)

Dit is je **“panic reset & rebuild” guide**.

1️⃣ Cluster volledig verwijderen

```
kind delete cluster --name milestone2
```

2️⃣ Cluster opnieuw maken

```
kind create cluster --name milestone2 --config k8s/kind-config.yaml

kubectl get nodes
```

3️⃣ Docker images bouwen

```
docker build -t milestone2-api:v3 .\api

docker build -t milestone2-frontend:v4 .\frontend
```

4️⃣ Images in kind laden

```
kind load docker-image milestone2-api:v3 --name milestone2

kind load docker-image milestone2-frontend:v4 --name milestone2
```

5️⃣ Kubernetes stack deployen

```
kubectl apply -f k8s/mariadb.yaml

kubectl apply -f k8s/api.yaml

kubectl apply -f k8s/frontend.yaml
```

6️⃣ Ingress NGINX installeren

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v1.11.3/deploy/static/provider/kind/deploy.yaml
```

```
kubectl get pods -n ingress-nginx -w
```

7️⃣ cert-manager installeren (HTTPS)

```
kubectl apply -f https://github.com/cert-manager/cert-manager/releases/download/v1.14.5/cert-manager.yaml
```

```
kubectl get pods -n cert-manager -w
```

Apply TLS resources:

```
kubectl apply -f k8s/cluster-issuer.yaml
```

```
kubectl apply -f k8s/certificate.yaml
```

```
kubectl apply -f k8s/ingress.yaml
```

8️⃣ API schalen (altijd 3 pods)

(of al in yaml replicas: 3)

```
kubectl scale deployment api-deployment --replicas=3
```

```
kubectl get pods -l app=api -o wide
```

9️⃣ Monitoring (Prometheus + Grafana)

```
helm repo add prometheus-community https://prometheus-community.github.io/helm-charts
```

```
helm repo update
```

```
kubectl create namespace monitoring
```

```
helm install kps prometheus-community/kube-prometheus-stack -n monitoring
```

```
kubectl get pods -n monitoring
```

🔟 ArgoCD (GitOps)

kubectl create namespace argocd

kubectl apply -n argocd -f https://raw.githubusercontent.com/argoproj/argo-cd/stable/manifests/install.yaml

kubectl apply -f k8s/argocd-app.yaml

UI:

kubectl -n argocd port-forward svc/argocd-server 9095:443

🌐 Toegang tot app

kubectl -n ingress-nginx port-forward svc/ingress-nginx-controller 8088:443

Browser:

https://milestone.local:8088

✅ Snelle eindcheck

kubectl get nodes

kubectl get pods

kubectl get ingress

kubectl get applications -n argocd

LIVE DEMO SCRIPT — 10 COMMANDS

1️⃣ Toon cluster & nodes

```
kubectl get nodes
```

Zeg:

“Dit is mijn kind cluster met 1 control-plane en 2 worker nodes.”

2️⃣ Toon alle pods

```
kubectl get pods
```

Zeg:

“Hier zie je frontend, API en MariaDB.”

3️⃣ Toon API scaling & verdeling

```
kubectl get pods -l app=api -o wide
```

Zeg:

“De API draait met 3 replicas, verdeeld over meerdere nodes.”

4️⃣ Toon services

```
kubectl get svc
```

Zeg:

“De services worden via Ingress bereikbaar gemaakt.”

5️⃣ Toon Ingress

```
kubectl get ingress
```

Zeg:

“De Ingress routeert frontend en API-verkeer.”

6️⃣ Toon HTTPS (cert-manager)

```
kubectl get certificate
```

Zeg:

“HTTPS is enabled via cert-manager.”

7️⃣ Toon health checks

```
kubectl describe pod $(kubectl get pods -l app=api -o
jsonpath="{.items[0].metadata.name}")
```

Zeg:

“Hier zie je de liveness en readiness probes.”

8️⃣ Forceer een restart (bewijs health)

```
kubectl exec -it $(kubectl get pods -l app=api -o jsonpath="{.items[0].metadata.name}")
-- sh -c "kill 1"
```

Zeg:

“Ik kill bewust een container, Kubernetes herstart deze automatisch.”

9️⃣ Toon monitoring

```
kubectl get pods -n monitoring
```

Zeg:

“Prometheus en Grafana monitoren cluster resources.”

🔟 Toon GitOps (ArgoCD)

```
kubectl get applications -n argocd
```

Zeg:

“De applicatie wordt gedeployed via ArgoCD met een GitOps workflow.”

🌐 EXTRA (browser – geen command)

Open:

https://milestone.local:8088

Wijs aan:

- Naam uit DB
- Current API container ID: AS-...
- Refresh → andere pod ID