Backend Rollout Guide (Minikube + Kubernetes)

This guide provides the commands to roll out backend changes (code + requirements) to Minikube and verify the live pod’s code.  
  
Assumptions:  
- Project/image name: cv-analyzer  
- Namespace: default  
- Deployment: cv-analyzer, Service: cv-analyzer-service  
- Container serves on port 8000

# 0) Prerequisites (one-time)

- Minikube is running:  
 minikube status  
- K8s manifests applied (Deployment + Service) and working  
- Docker configured to build images locally

# 1) Build a fresh image (no cache)

Use a new tag for every rollout so the deployment pulls your newest image:  
TAG=$(date +%s)  
docker build --no-cache -t cv-analyzer:$TAG .

# 2) Load the image into Minikube

minikube image load cv-analyzer:$TAG  
If this fails, re-run the build/load steps.

# 3) Point the Deployment at the new image

kubectl set image deployment/cv-analyzer cv-analyzer=cv-analyzer:$TAG -n default  
  
Confirm deployment references the new tag:  
kubectl get deploy cv-analyzer -n default -o jsonpath='{.spec.template.spec.containers[0].image}'; echo

# 4) Wait for the rollout to complete

kubectl rollout status deployment/cv-analyzer -n default  
  
If it gets stuck:  
kubectl get pods -n default -l app=cv-analyzer  
kubectl describe pod <pod-name> -n default

# 5) Optional: Test the service endpoint

kubectl get svc cv-analyzer-service -n default -o wide  
minikube ip  
Visit: http://<minikube-ip>:<nodePort>/docs  
  
Or port-forward:  
kubectl port-forward -n default svc/cv-analyzer-service 8080:8000  
Visit: http://127.0.0.1:8080/docs  
  
Or use Minikube tunnel:  
minikube service cv-analyzer-service -n default

# 6) Verify logs

POD=$(kubectl get pods -n default -l app=cv-analyzer -o jsonpath='{.items[0].metadata.name}')  
kubectl logs -f -n default "$POD"

# 7) Inspect the code inside the running pod

POD=$(kubectl get pods -n default -l app=cv-analyzer -o jsonpath='{.items[0].metadata.name}')  
kubectl exec -n default "$POD" -- sh -lc 'sed -n "1,120p" /app/main.py'  
kubectl exec -n default "$POD" -- sh -lc 'grep -n "OpenAI" /app/main.py || true'  
kubectl exec -n default "$POD" -- sh -lc 'ls -la /app && ls -la /app/templates || true'

# 8) If your API key changed

kubectl delete secret openai-secret -n default  
kubectl create secret generic openai-secret -n default --from-literal=OPENAI\_API\_KEY="$OPENAI\_API\_KEY"  
kubectl rollout restart deployment cv-analyzer -n default  
kubectl rollout status deployment cv-analyzer -n default

# 9) Quick one-liner script

TAG=$(date +%s) && docker build --no-cache -t cv-analyzer:$TAG . && minikube image load cv-analyzer:$TAG && kubectl set image deployment/cv-analyzer cv-analyzer=cv-analyzer:$TAG -n default && kubectl rollout status deployment/cv-analyzer -n default && POD=$(kubectl get pods -n default -l app=cv-analyzer -o jsonpath='{.items[0].metadata.name}') && kubectl exec -n default "$POD" -- sh -lc 'sed -n "1,120p" /app/main.py'

# Troubleshooting

If old code still shows:  
- Use a new tag and confirm deployment image changed  
- Build with --no-cache  
- Delete old pod: kubectl delete pod -n default -l app=cv-analyzer  
  
If pod is CrashLoopBackOff:  
kubectl logs -n default "$POD" --previous  
kubectl describe pod -n default "$POD"  
  
If service not reachable:  
kubectl get svc cv-analyzer-service -n default -o wide  
kubectl get endpoints cv-analyzer-service -n default