

## installation

Gateway Setup on Existing Minikube (GCP VM) This guide is for your case: Minikube is already running on GCP, and you want Kubernetes Gateway API to work. Scope This file covers only the required software/components to run Gateway: kubectl (client CLI) helm (install controller) Gateway API CRDs A Gateway controller (GatewayClass provider) It does **not** recreate Minikube.

1) Verify required CLI tools on your GCP machine bash kubectl version --client helm version If helm is missing, install quickly: bash curl <https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3> | bash

2) Confirm Minikube cluster access bash kubectl cluster-info kubectl get nodes -o wide kubectl get ns If this fails, fix kubeconfig/context first: bash kubectl config current-context kubectl config get-contexts

3) Install Gateway API CRDs (required) bash kubectl apply -f <https://github.com/kubernetes-sigs/gateway-api/releases/download/v1.2.1/standard-install.yaml> Verify CRDs are active: bash kubectl api-resources | grep gateway.networking.k8s.io Expected resources include: - gatewayclasses - gateways - httproutes

4) Install Gateway controller (required) This project uses gatewayClassName: nginx, so install NGINX Gateway Fabric. bash helm install ngf oci://ghcr.io/nginx/charts/nginx-gateway-fabric -n nginx-gateway --create-namespace --wait If OCI pull is blocked in your environment, use source-based install: bash helm pull oci://ghcr.io/nginx/charts/nginx-gateway-fabric --untar cd nginx-gateway-fabric helm install ngf . -n nginx-gateway --create-namespace --wait

Verify controller: bash kubectl get pods -n nginx-gateway kubectl get gatewayclass You should see a GatewayClass (commonly nginx).

5) Match your manifest class name Check gateway-demo/manifests/02-gateway.yaml: yaml gatewayClassName: nginx If your installed class name is different, update that value before apply.

6) Deploy your gateway demo bash kubectl apply -f gateway-demo/manifests/00-namespace.yaml kubectl apply -f gateway-demo/manifests/01-apps-services.yaml kubectl apply -f gateway-demo/manifests/02-gateway.yaml kubectl apply -f gateway-demo/manifests/03-httproute.yaml Verify: bash kubectl get gateway,httproute -n gateway-demo kubectl describe gateway gateway-demo -n gateway-demo kubectl describe httproute echo-route -n gateway-demo Look for Accepted=True and Programmed=True conditions.

7) Expose and test traffic Get the Gateway controller service: bash kubectl get svc -n nginx-gateway Use its external IP (or NodePort/IP depending on your setup): bash curl <http://<GATEWAY-IP>/v1> curl <http://<GATEWAY-IP>/v2> curl <http://<GATEWAY-IP>> Expected: - /v1 returns echo-v1 - /v2 returns echo-v2 - / returns echo-v1 Troubleshooting no matches for kind "Gateway" CRDs missing; re-run step 3. kubectl get gatewayclass is empty Controller not installed/running; re-run step 4. 2a. helm repo ... is not a valid chart repository - Expected with old instructions; use OCI install command from step 4. Gateway not programmed Check controller logs: bash kubectl logs -n nginx-gateway deploy/ngf-nginx-gateway-fabric --tail=200 Route not reachable Check service exposure/firewall/network path from your client to GCP VM or LB. Verify route attachment: bash kubectl get httproute -n gateway-demo -o yaml