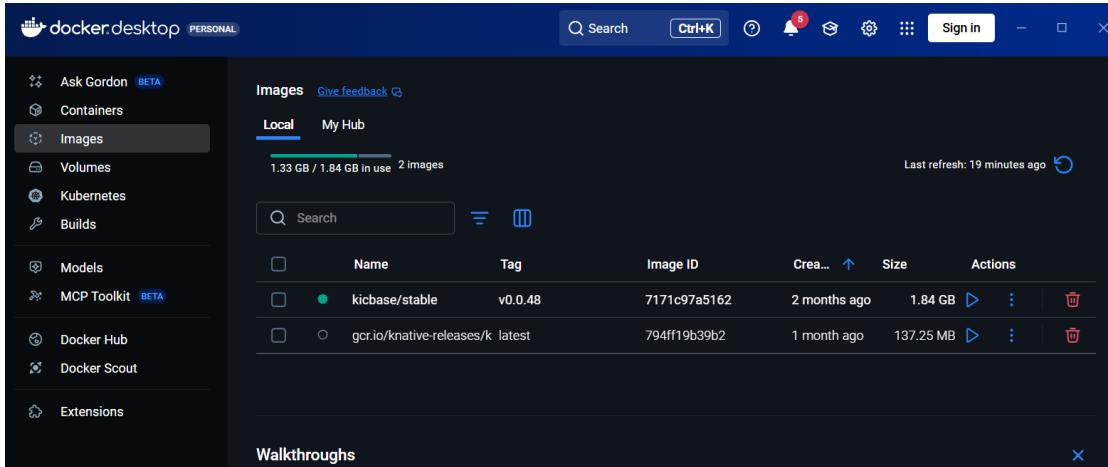


## 1. Install the Knative CLI

名称	修改日期	类型	大小
kn.exe	2025/11/12 14:58	应用程序	85,504 KB
kn-quickstart.exe	2025/11/12 14:53	应用程序	15,013 KB



## 2. Add to the environment variable

```
Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。

安装最新的 PowerShell，了解新功能和改进！ https://aka.ms/PSWindows

PS D:\Knative> mv kn.exe "$HOME\AppData\Local\Microsoft\WindowsApps\"
PS D:\Knative> mv kn-quickstart.exe "$HOME\AppData\Local\Microsoft\WindowsApps\"
PS D:\Knative> kn quickstart --help
Get up and running with a local Knative environment

Usage:
  kn-quickstart [command]

Available Commands:
  completion  Generate the autocompletion script for the specified shell
  help        Help about any command
  kind        Quickstart with Kind
  minikube    Quickstart with Minikube
  version     Prints the plugin version

Flags:
  -h, --help   help for kn-quickstart

Use "kn-quickstart [command] --help" for more information about a command.
```

## 3. Run the Knative quickstart plugin

```
diluo@Wzm:/mnt/d/Knative$ kn quickstart minikube
Running Knative Quickstart using Minikube
Minikube version is: v1.32.0

✓ Creating Minikube cluster...
😊 [knative] minikube v1.32.0 on Ubuntu 20.04
✨ Using the docker driver based on user configuration
✅ Using image repository registry.cn-hangzhou.aliyuncs.com/google_containers
🔥 Creating docker container (CPUs=3, Memory=4096MB) ...
🌐 Preparing Kubernetes v1.27.4 on Docker 24.0.7 ...
  - Generating certificates and keys ...
  - Booting up control plane ...
  - Configuring RBAC rules ...
🔧 Configuring bridge CNI (Container Networking Interface) ...
🔍 Verifying Kubernetes components...
  - Using image registry.cn-hangzhou.aliyuncs.com/google_containers/storage-provisioner:v5
🌟 Enabled addons: storage-provisioner, default-storageclass
🎉 Done! kubectl is now configured to use "knative" cluster and "default" namespace by default

✓ Minikube cluster is running
✓ Kubernetes version: v1.27.4

📦 Installing Knative Serving v1.11.1...
  - Creating Knative Serving namespace...
  - Installing Serving CRDs...
customresourcedefinition.apiextensions.k8s.io/certificates.networking.internal.knative.dev created
customresourcedefinition.apiextensions.k8s.io/configurations.serving.knative.dev created
customresourcedefinition.apiextensions.k8s.io/domainMappings.serving.knative.dev created
customresourcedefinition.apiextensions.k8s.io/ingresses.networking.internal.knative.dev created
customresourcedefinition.apiextensions.k8s.io/metrics.autoscaling.internal.knative.dev created
customresourcedefinition.apiextensions.k8s.io/podAutoscalers.autoscaling.internal.knative.dev created
customresourcedefinition.apiextensions.k8s.io/revisions.serving.knative.dev created
customresourcedefinition.apiextensions.k8s.io/serverlessServices.networking.internal.knative.dev created
customresourcedefinition.apiextensions.k8s.io/services.serving.knative.dev created
  - Installing Serving core components...
namespace/knative-serving created
deployment.apps/activator created
deployment.apps/autoscaler created
deployment.apps/controller created
```

```

pod/webhook-6d5b4f8d5d-r17pq condition met
pod/3scale-kourier-control-7b5d4c8d5d-zm8xk condition met
pod/3scale-kourier-gateway-6c4b5f8d5d-pq9wl condition met

✓ Knative Serving installed successfully!

💡 Configuring DNS...
configmap/config-domain created

📊 Verifying installation...
knative version
Version: v20231113-3f8f0d7e66
Build Date: 2023-11-13T18:36:39Z
Git Revision: 3f8f0d7e66
Supported APIs:
* Serving
  - serving.knative.dev/v1 (services, configurations, routes, revisions)
  - autoscaling.internal.knative.dev/v1alpha1 (metrics, podautoscalers)
  - networking.internal.knative.dev/v1alpha1 (serverlesservices, ingresses, certificates)
  - domains.internal.knative.dev/v1alpha1 (domains, domainmappings)

```

```

* Eventing
  - messaging.knative.dev/v1 (channels, subscriptions)
  - eventing.knative.dev/v1 (eventtypes, brokers, triggers)
  - sources.knative.dev/v1 (apiserversources, containersources, pingsources)

```

🎉 Knative Quickstart environment is ready!

💡 Your cluster is running and Knative is installed.

Minikube IP address: 192.168.49.2  
Knative Serving: <http://knative-serving.192.168.49.2.nip.io>

💡 To create your first Knative service, run:  
`kn service create hello --image gcr.io/knative-samples/helloworld-go --port 8080`

📊 For more examples, visit: <https://knative.dev/docs/serving/samples/>

▢ To stop your cluster, run:  
`minikube stop`

▢ To restart your cluster, run:  
`minikube start`

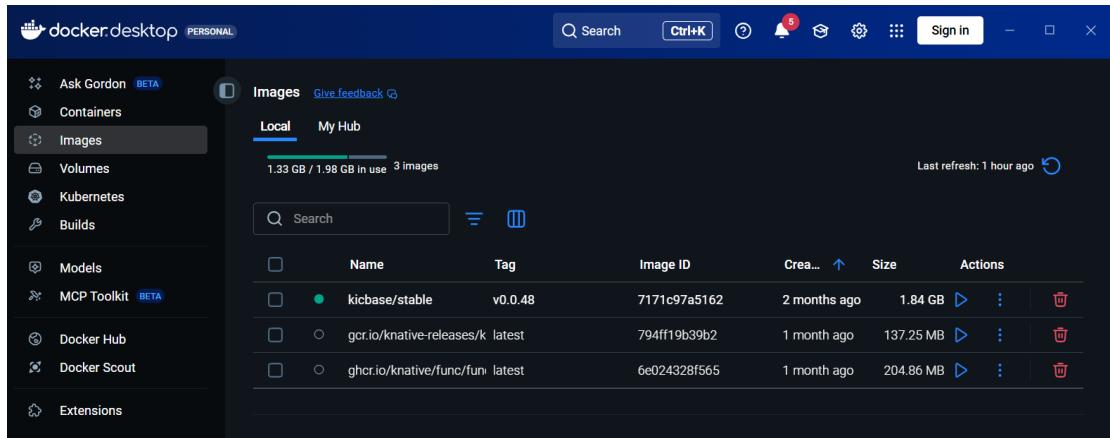
▢ To delete your cluster, run:  
`minikube delete`

## 4. Creating Knative Functions

```

diluo@Wzm:/mnt/d/Knative$ docker run --rm -it ghcr.io/knative/func/func create -l node -t http myfunc
Unable to find image 'ghcr.io/knative/func/func:latest' locally
latest: Pulling from knative/func/func
f911df201992: Pull complete
Digest: sha256:6e024328f5657ff5d9d65571ea0b198326e2a4279625bc6e5b9dd161eda2fabc
Status: Downloaded newer image for ghcr.io/knative/func/func:latest
Created node function in /myfunc

```



## 5. create new file named hello.yaml

A screenshot of a code editor window. The title bar says 'hello.yaml'. The left sidebar has icons for file operations like New, Open, Save, Find, Replace, and others. The main area contains the following YAML code:

```
apiVersion: serving.knative.dev/v1
kind: Service
metadata:
  name: hello
spec:
  template:
    spec:
      containers:
        - image: ghcr.io/knative/helloworld-go:latest
          ports:
            - containerPort: 8080
          env:
            - name: TARGET
              value: "World"
```