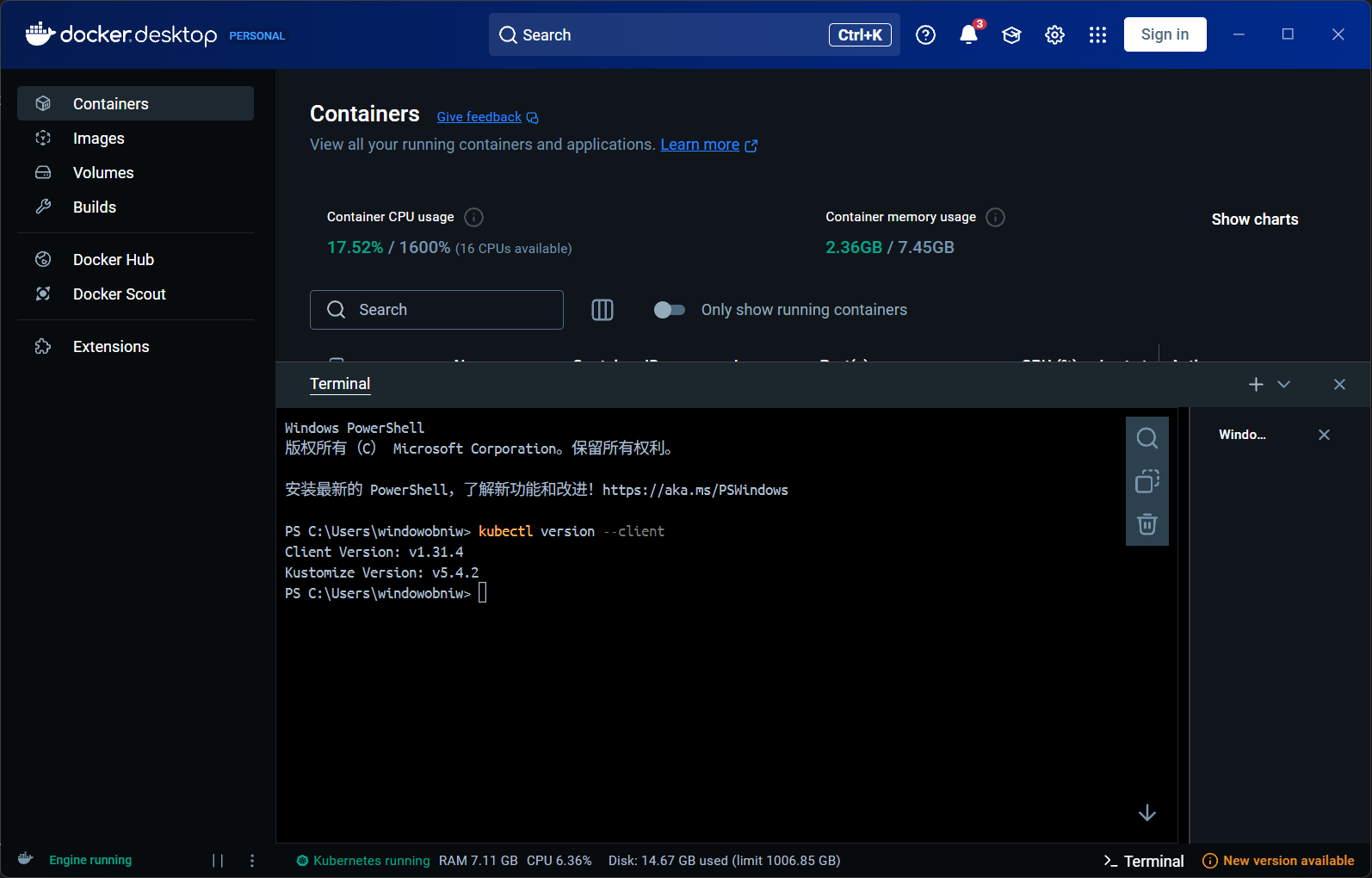
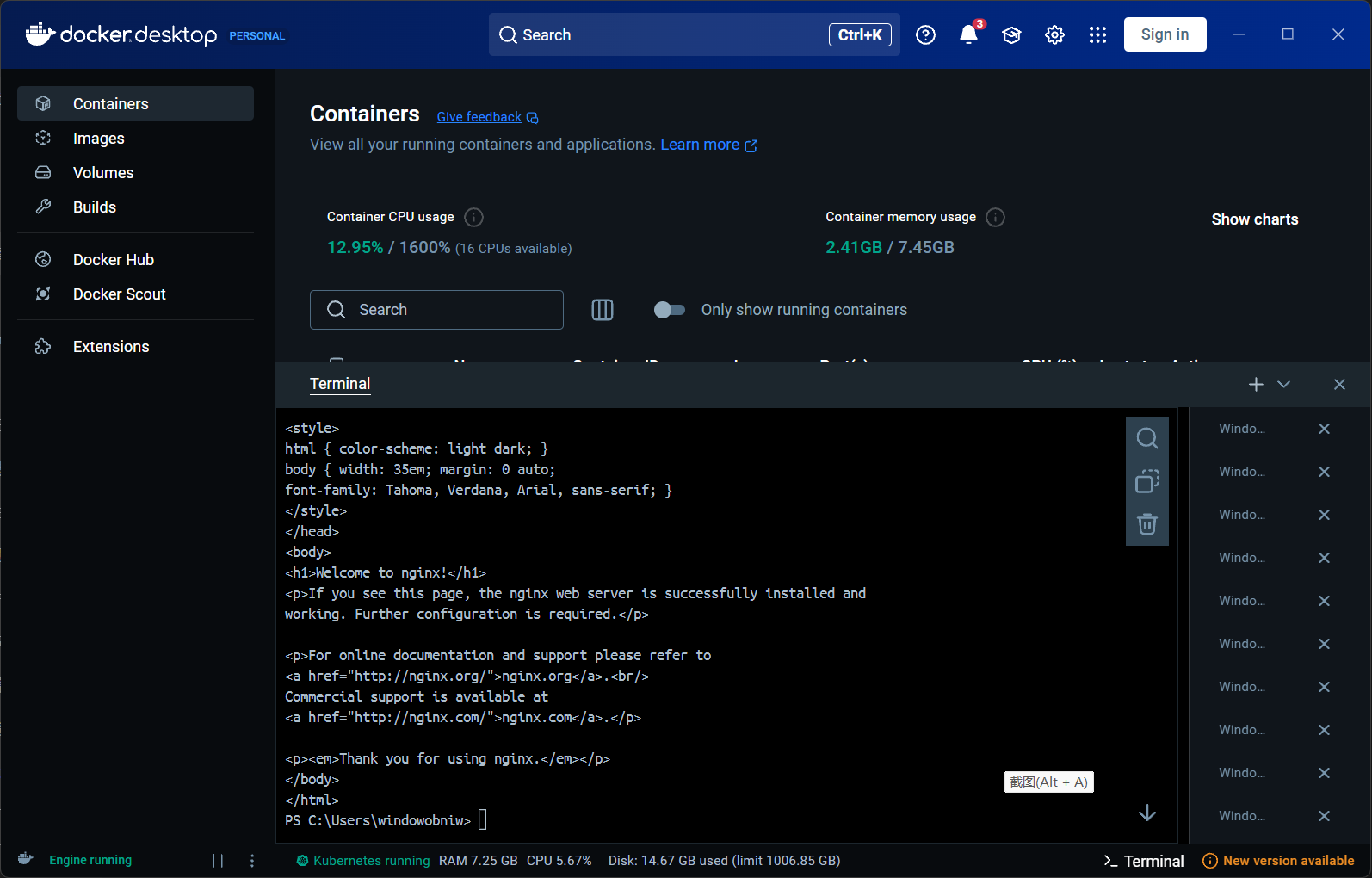
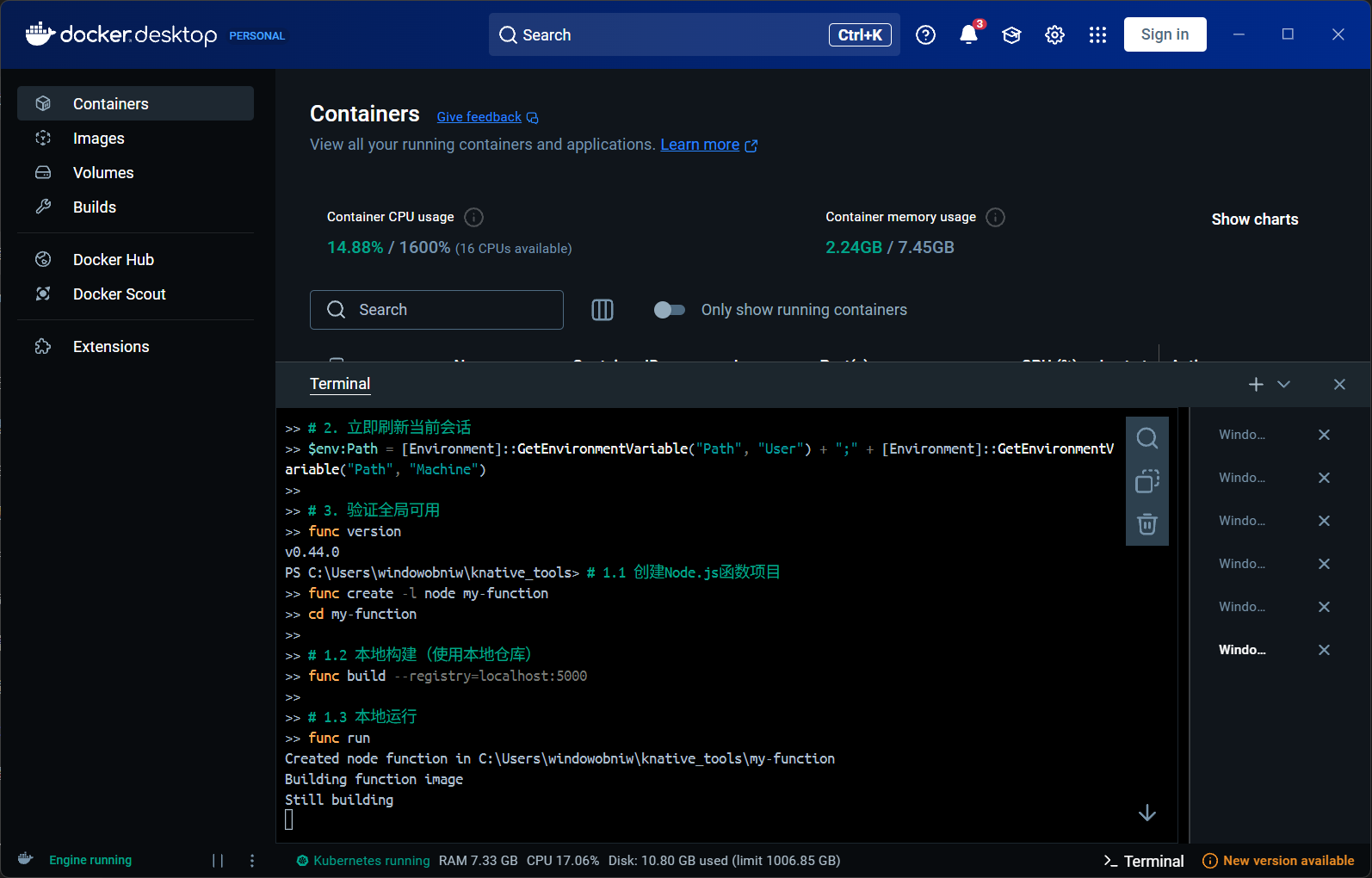
Knative Experiment Guide

[Knative Functions] Create and run functions (skipping deployment)

[Knative Serving] Deploy services

[Knative Eventing] Config event-driven

To run Knative experiments in Windows Docker Desktop China, follow these steps to prepare the environment and operate it: 

1. Experiment running environment

Make sure Docker Desktop is installed and running

Download and install Docker Desktop for Windows

Open Docker Desktop and make sure the state is \*\*"Running"\*\*

Enable Kubernetes in Docker Desktop Settings:

Go to Settings → Kubernetes

Check \*\*"Enable Kubernetes"\*\*

Click Apply & Restart电脑屏幕的手机截图

AI 生成的内容可能不正确。

1. Knative Functions (built locally)

Goal: Create and build functions using the func CLI (skipping deployment).

Install the func CLI

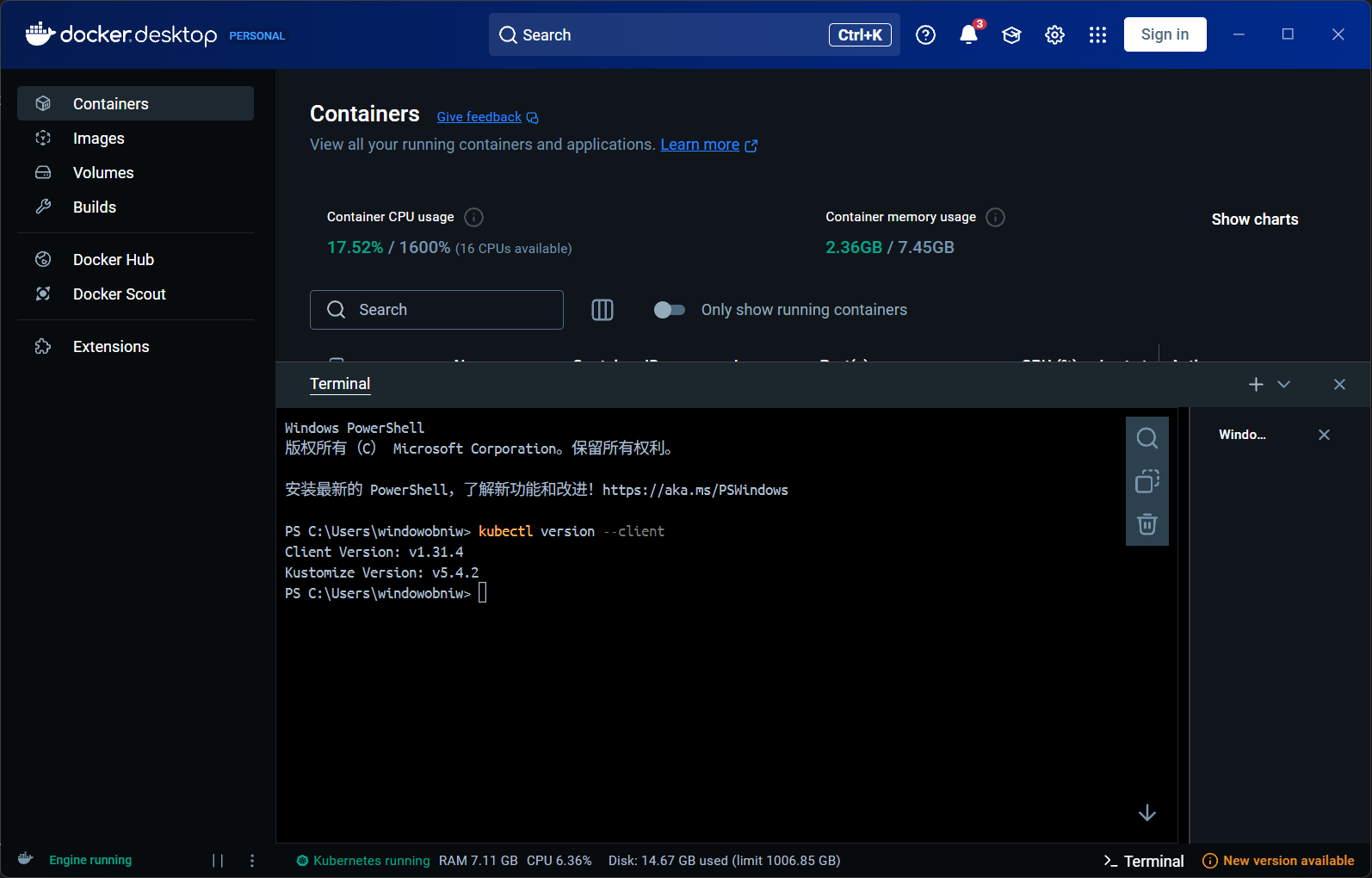
powershell

replication

winget install func

Verify installation:

powershell

replication

func --version

Screenshot point: func --version output.

Creating a function (Go example)

powershell

replication

func create -l go hello-func

cd hello-func

Screenshot point: dir shows the resulting function directory structure.

Build functions to local images

powershell电脑屏幕的手机截图

AI 生成的内容可能不正确。

replication

func build -v

Screenshot: build log (showing that the OCI image was generated successfully).

Running tests locally

powershell

replication

func run

Open another terminal test:

powershell

replication

curl http://localhost:8080

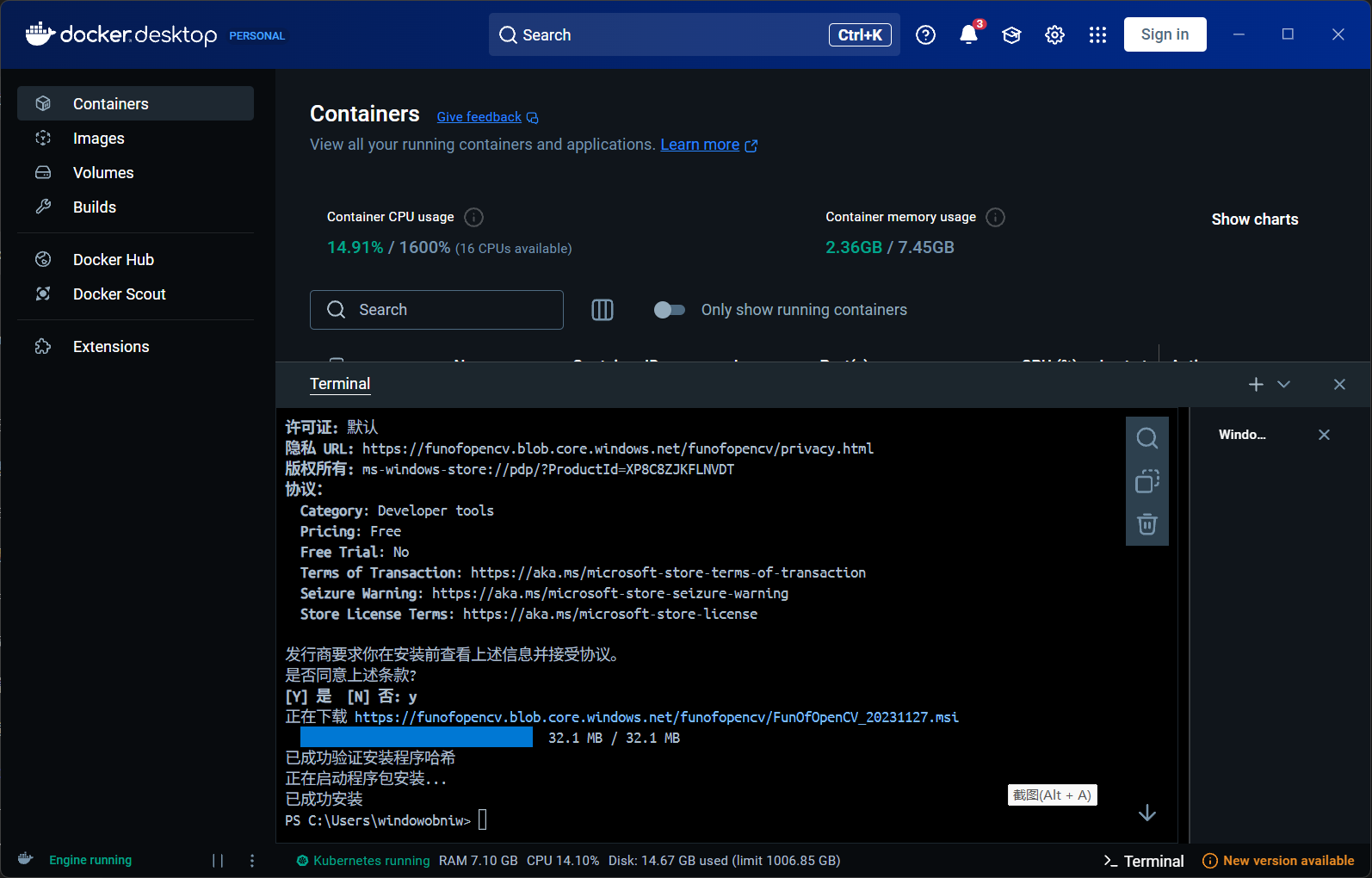
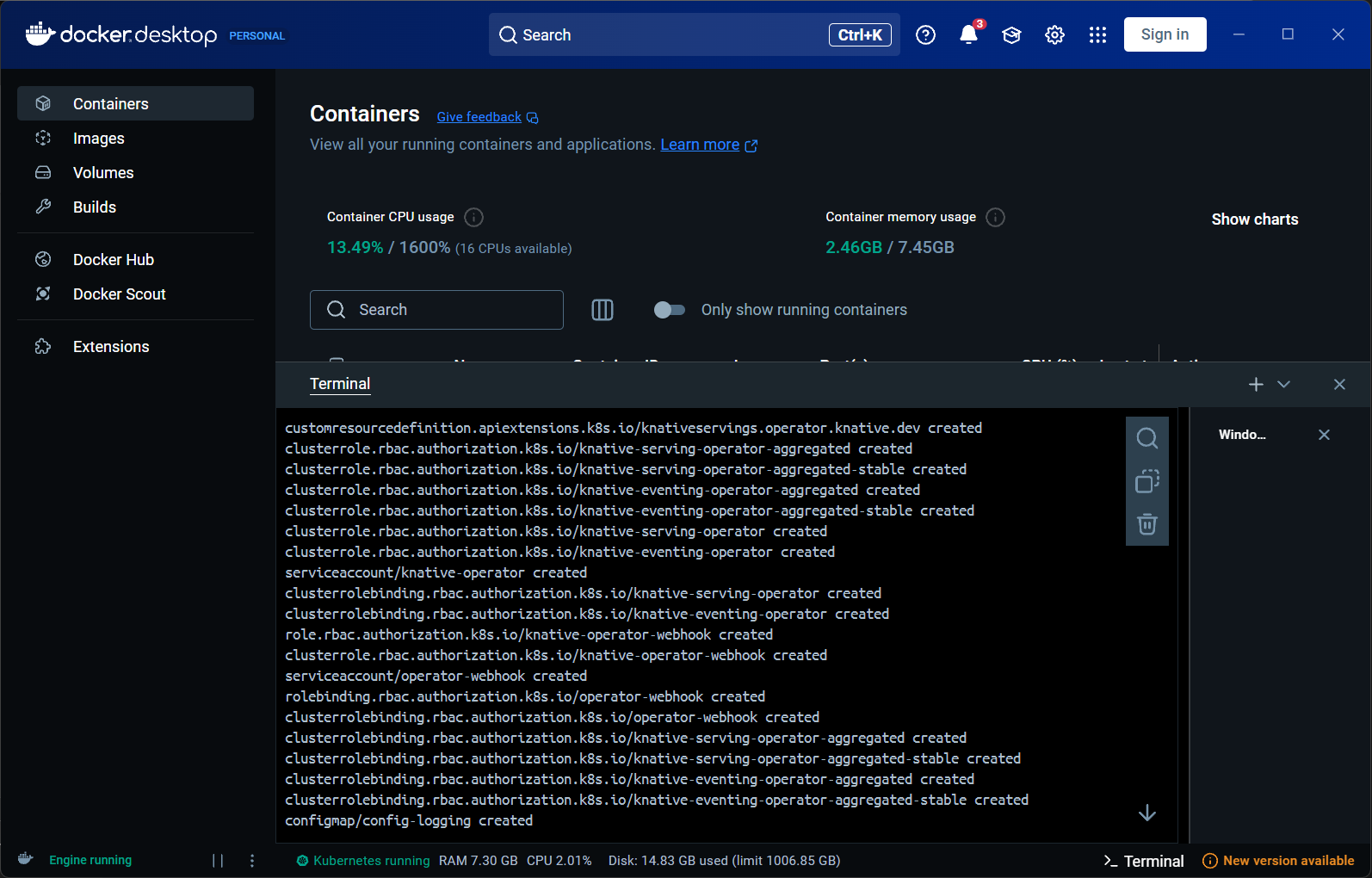
2. Knative Serving

Goal: Deploy a Hello World service through kn.

Install Knative Serving

Install Serving components using kubectl (requires Knative Operator first) :

Powershell



replication

Kubectl apply -f https://github.com/knative/operator/releases/download/knative-v1.11.0/operator.yaml

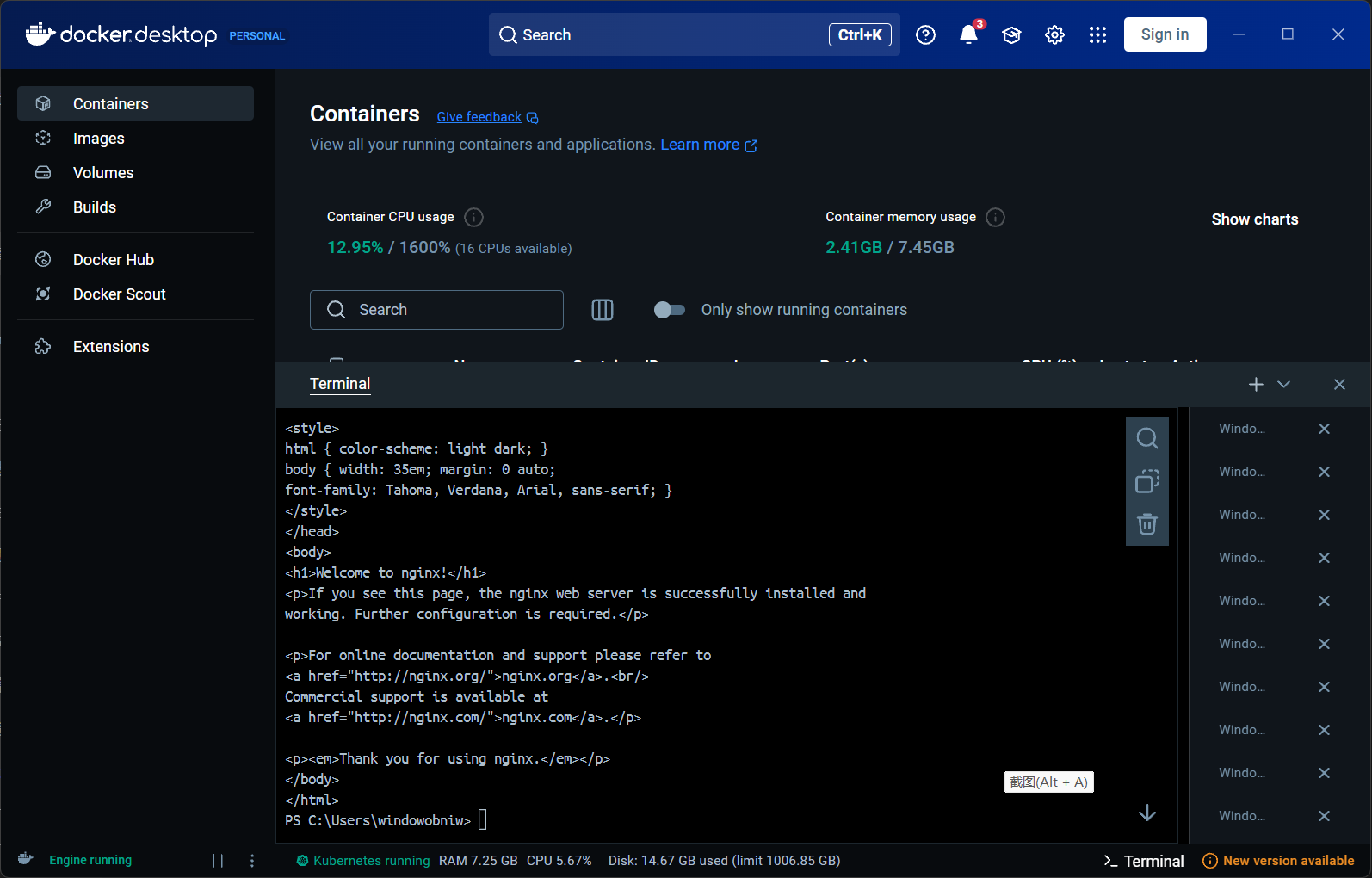
kubectl apply -f serving.yaml # Customize YAML or use the official example

kubectl get pods -n knative-serving (shows all pods as Running).

Deploy the Hello World service

powershell

replication



kn service create hello --image ghcr.io/knative/helloworld-go:latest --port 8080 --env TARGET=World

Screenshot point: kn service list shows the service URL and status.

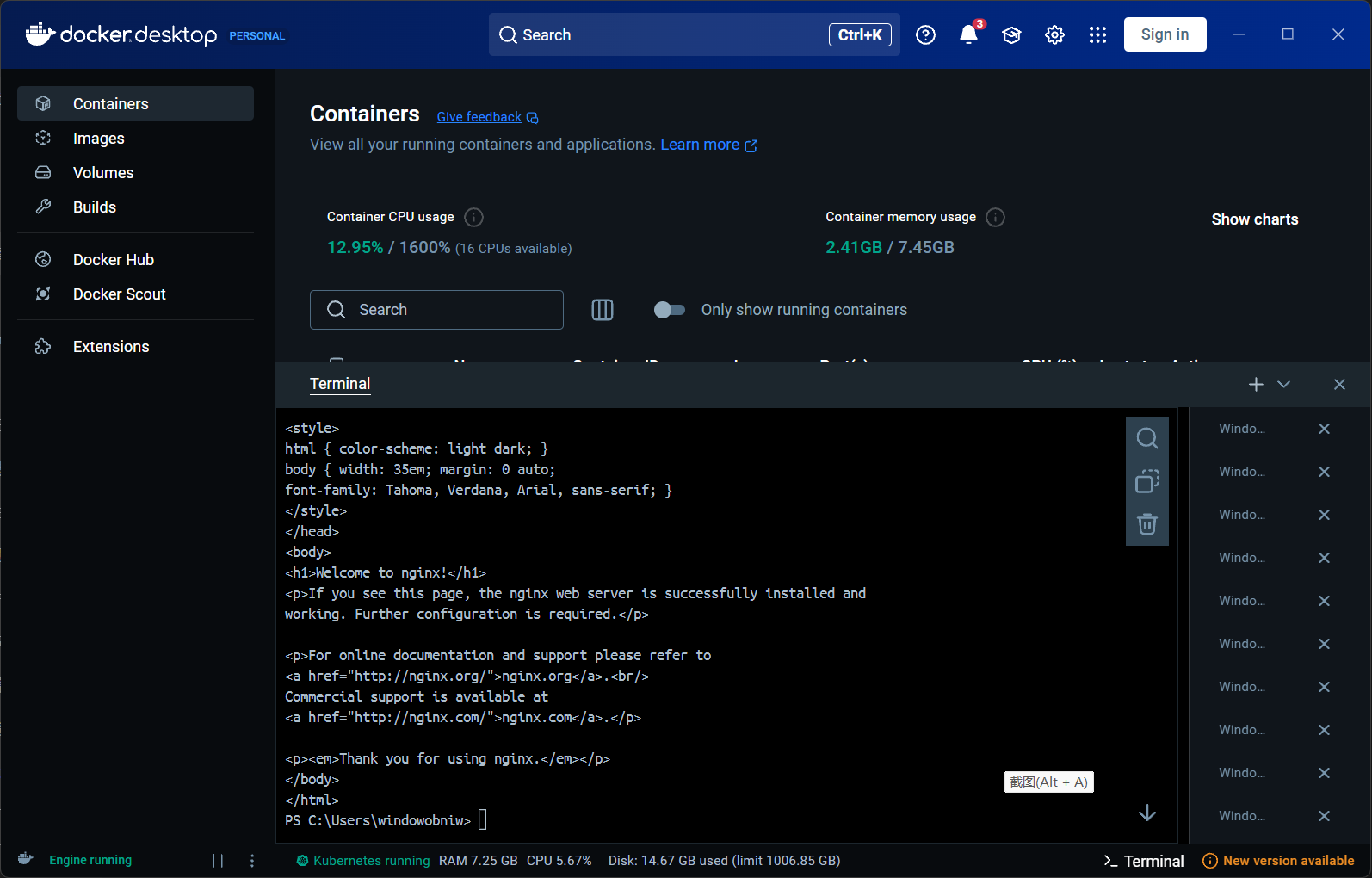
Access services

powershell

replication

The curl http://hello.default.10.0.0.1.sslip.io # replace the actual URL for you

Screenshot: curl returns Hello World! .



3. Knative Eventing

Goal: Configure Broker and Trigger to forward events.

Install Knative Eventing

powershell

replication

Kubectl apply -f https://github.com/knative/eventing/releases/download/knative-v1.11.0/eventing.yaml

kubectl get pods -n knative-eventing (confirm Pod status).

Creating a default Broker

powershell

replication

kubectl apply -f - <<EOF

apiVersion: eventing.knative.dev/v1

kind: Broker

metadata:

name: default

EOF

kubectl get broker says Ready=True.

Create a Trigger to bind to the Hello service

powershell

replication

kn trigger create hello-trigger --broker default --sink ksvc:hello

Screenshot point: kn trigger list shows Trigger status.

Sending test events

Create the test event file event.json:

json

replication

{

"specversion": "1.0",

"type": "dev.knative.example",

"source": "curl",

"data": "{\"msg\": \"Hello Knative! \ "}"

}

Send events:

powershell

replication

curl -X POST -H "Content-Type: application/json" -d @event.json http://broker-ingress.knative-eventing.svc.cluster.local/default/default

kubectl logs -l app=hello Show event logs.