KServe Quickstart

Prerequisites

Tools

- kubectl - The Kubernetes command-line tool
- helm - for installing KServe and other Kubernetes operators
- curl - for the quickstart script and for testing API endpoints (installed by default on most systems)

1. verify kubectl installation

   kubectl version –client

```
PS D:\apps-data\docker> kubectl version --client
Client Version: v1.34.1
Kustomize Version: v5.7.1
```

2. verify helm installation

   .\ helm version

```
PS D:\apps-data\docker> .\helm version
version.BuildInfo{Version:"v4.1+unreleased", GitCommit:"8802d959cbeb8215f600294aa9
af4261a04296a6", GitTreeState:"clean", GoVersion:"go1.25.5", KubeClientVersion:"v1
.34"}
```

3. verify curl installation

   curl.exe --version

```
PS D:\apps-data\docker> curl.exe --version
curl 8.16.0 (Windows) libcurl/8.16.0 Schannel zlib/1.3.1 WinIDN
Release-Date: 2025-09-10
Protocols: dict file ftp ftps gopher gophers http https imap imaps ipfs ipns ldap
ldaps mqtt pop3 pop3s smb smbs smtp smtps telnet tftp ws wss
Features: alt-svc AsynchDNS HSTS HTTPS-proxy IDN IPv6 Kerberos Largefile libz NTLM
 SPNEGO SSL SSPI threadsafe Unicode UnixSockets
```

Install Kserve with Kubernetes deployment installation

minikube start --driver=docker --memory=10240 --cpus=8 --base-image=registry.k8s.io/kicbase/stable:v0.0.48 --kubernetes-version=v1.34.0

Note: The first attempt at istio-system kept crashing with CrashLoopBackOff, and all three replicas failed simultaneously. The reason was likely insufficient initial memory allocation. Therefore, this time we increased the CPU to 8 and memory to 10240MB.

```
PS D:\apps-data\docker> minikube start --driver=docker --memory=10240 --cpus=8 --base-image=registry.k8s.io/kicbase/stable:v0.0.48 --kuberne
tes-version=v1.34.0
* Microsoft Windows 11 Home China 10.0.26100.7462 Build 26100.7462 上的 minikube v1.37.0
* 根据用户配置使用 docker 驱动程序
* 使用具有 root 权限的 Docker Desktop 驱动程序
* 在集群中 "minikube" 启动节点 "minikube" primary control-plane
* 正在拉取基础镜像 v0.0.48 ...
! minikube was unable to download registry.k8s.io/kicbase/stable:v0.0.48, but successfully downloaded docker.io/kicbase/stable:v0.0.48 as a
fallback image
* 创建 docker container (CPU=8, 内存=10240MB) ...|
* 正在 Docker 28.4.0 中准备 Kubernetes v1.34.0…-
! Failing to connect to https://registry.k8s.io/ from both inside the minikube container and host machine
* 要获取新的外部镜像，可能需要配置代理: https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
* 配置 bridge CNI (Container Networking Interface) ...
* 正在验证 Kubernetes 组件...
  - 正在使用镜像 gcr.io/k8s-minikube/storage-provisioner:v5
* 启用插件:  storage-provisioner, default-storageclass
* 完成! kubectl 现在已配置，默认使用"minikube"集群和"default"命名空间
```

kubectl apply -f .\yaml\serving-crds.yaml

```
PS D:\apps-data\docker> kubectl apply -f .\yaml\serving-crds.yaml
Warning: unrecognized format "int64"
customresourcedefinition.apiextensions.k8s.io/certificates.networking.internal.kna
tive.dev created
Warning: unrecognized format "int32"
customresourcedefinition.apiextensions.k8s.io/configurations.serving.knative.dev c
reated
customresourcedefinition.apiextensions.k8s.io/clusterdomainclaims.networking.inter
nal.knative.dev created
customresourcedefinition.apiextensions.k8s.io/domainmappings.serving.knative.dev c
reated
customresourcedefinition.apiextensions.k8s.io/ingresses.networking.internal.knativ
e.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 create
d
customresourcedefinition.apiextensions.k8s.io/routes.serving.knative.dev created
customresourcedefinition.apiextensions.k8s.io/serverlessservices.networking.intern
```

kubectl apply -f .\yaml\serving-core.yaml

```
PS D:\apps-data\docker> kubectl apply -f .\yaml\serving-core.yaml
namespace/knative-serving created
role.rbac.authorization.k8s.io/knative-serving-activator created
clusterrole.rbac.authorization.k8s.io/knative-serving-activator-cluster created
clusterrole.rbac.authorization.k8s.io/knative-serving-aggregated-addressable-resol
ver created
clusterrole.rbac.authorization.k8s.io/knative-serving-addressable-resolver created
clusterrole.rbac.authorization.k8s.io/knative-serving-namespaced-admin created
clusterrole.rbac.authorization.k8s.io/knative-serving-namespaced-edit created
clusterrole.rbac.authorization.k8s.io/knative-serving-namespaced-view created
clusterrole.rbac.authorization.k8s.io/knative-serving-core created
clusterrole.rbac.authorization.k8s.io/knative-serving-podspecable-binding created
serviceaccount/controller created
clusterrole.rbac.authorization.k8s.io/knative-serving-admin created
clusterrolebinding.rbac.authorization.k8s.io/knative-serving-controller-admin crea
ted
clusterrolebinding.rbac.authorization.k8s.io/knative-serving-controller-addressabl
e-resolver created
serviceaccount/activator created
```

kubectl apply -f .\yaml\istio.yaml

```
PS D:\apps-data\docker> kubectl apply -f .\yaml\istio.yaml
namespace/istio-system created
customresourcedefinition.apiextensions.k8s.io/authorizationpolicies.security.istio.io created
Warning: unrecognized format "double"
Warning: unrecognized format "int32"
customresourcedefinition.apiextensions.k8s.io/destinationrules.networking.istio.io created
customresourcedefinition.apiextensions.k8s.io/envoyfilters.networking.istio.io created
customresourcedefinition.apiextensions.k8s.io/gateways.networking.istio.io created
customresourcedefinition.apiextensions.k8s.io/peerauthentications.security.istio.io created
customresourcedefinition.apiextensions.k8s.io/proxyconfigs.networking.istio.io created
customresourcedefinition.apiextensions.k8s.io/requestauthentications.security.istio.io created
customresourcedefinition.apiextensions.k8s.io/serviceentries.networking.istio.io created
customresourcedefinition.apiextensions.k8s.io/sidecars.networking.istio.io created
customresourcedefinition.apiextensions.k8s.io/telemetries.telemetry.istio.io created
Warning: unrecognized format "binary"
customresourcedefinition.apiextensions.k8s.io/virtualservices.networking.istio.io created
customresourcedefinition.apiextensions.k8s.io/wasmplugins.extensions.istio.io created
customresourcedefinition.apiextensions.k8s.io/workloadentries.networking.istio.io created
```

kubectl apply -f .\yaml\net-istio.yaml

```
PS D:\apps-data\docker> kubectl apply -f .\yaml\net-istio.yaml
clusterrole.rbac.authorization.k8s.io/knative-serving-istio created
gateway.networking.istio.io/knative-ingress-gateway created
gateway.networking.istio.io/knative-local-gateway created
service/knative-local-gateway created
configmap/config-istio created
peerauthentication.security.istio.io/webhook created
peerauthentication.security.istio.io/net-istio-webhook created
deployment.apps/net-istio-controller created
deployment.apps/net-istio-webhook created
secret/net-istio-webhook-certs created
service/net-istio-webhook created
mutatingwebhookconfiguration.admissionregistration.k8s.io/webhook.istio.networking.internal.knative.dev created
validatingwebhookconfiguration.admissionregistration.k8s.io/config.webhook.istio.networking.internal.knative.dev created
```

kubectl --namespace istio-system get service istio-ingressgateway

```
PS D:\apps-data\docker> kubectl --namespace istio-system get service istio-ingress
gateway
NAME                   TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)
                              AGE
istio-ingressgateway   LoadBalancer   10.98.221.124   <pending>     15021:31449/TC
P,80:31145/TCP,443:30920/TCP   6m33s
```

Cert Manager is required to provision webhook certs for production-grade installation.

kubectl apply -f .\yaml\cert-manager.yaml

```
PS D:\apps-data\docker> kubectl apply -f .\yaml\cert-manager.yaml
namespace/cert-manager created
customresourcedefinition.apiextensions.k8s.io/certificaterequests.cert-manager.io
created
Warning: unrecognized format "int64"
Warning: unrecognized format "int32"
customresourcedefinition.apiextensions.k8s.io/certificates.cert-manager.io created
customresourcedefinition.apiextensions.k8s.io/challenges.acme.cert-manager.io crea
ted
customresourcedefinition.apiextensions.k8s.io/clusterissuers.cert-manager.io creat
ed
customresourcedefinition.apiextensions.k8s.io/issuers.cert-manager.io created
customresourcedefinition.apiextensions.k8s.io/orders.acme.cert-manager.io created
serviceaccount/cert-manager-cainjector created
serviceaccount/cert-manager created
serviceaccount/cert-manager-webhook created
clusterrole.rbac.authorization.k8s.io/cert-manager-cainjector created
clusterrole.rbac.authorization.k8s.io/cert-manager-controller-issuers created
clusterrole.rbac.authorization.k8s.io/cert-manager-controller-clusterissuers creat
ed
clusterrole.rbac.authorization.k8s.io/cert-manager-controller-certificates created
clusterrole.rbac.authorization.k8s.io/cert-manager-controller-orders created
clusterrole.rbac.authorization.k8s.io/cert-manager-controller-challenges created
clusterrole.rbac.authorization.k8s.io/cert-manager-controller-ingress-shim created
```

kubectl get pods -A

```
PS D:\apps-data\docker> kubectl get pods -A
NAMESPACE         NAME                                       READY   STATUS    RESTARTS   AGE
cert-manager      cert-manager-548f7cf98c-8wvph              1/1     Running   0          3m18s
cert-manager      cert-manager-cainjector-8798f647f-sxh2q    1/1     Running   0          3m18s
cert-manager      cert-manager-webhook-6c8678dc46-mfvqw      1/1     Running   0          3m18s
istio-system      istio-ingressgateway-6c6444c759-8q2sm      1/1     Running   0          3m57s
istio-system      istio-ingressgateway-6c6444c759-xl2st      1/1     Running   0          3m57s
istio-system      istio-ingressgateway-6c6444c759-xqfl2      1/1     Running   0          3m57s
istio-system      istiod-66cf796db8-mzbvj                    1/1     Running   0          3m42s
istio-system      istiod-66cf796db8-slg92                    1/1     Running   0          3m42s
istio-system      istiod-66cf796db8-vzzvk                    1/1     Running   0          3m57s
knative-serving   activator-7bcd47489b-zqnww                 1/1     Running   0          4m4s
knative-serving   autoscaler-65cf6767c4-vncnw                1/1     Running   0          4m4s
knative-serving   controller-964dcf97b-x4mt5                 1/1     Running   0          4m4s
knative-serving   net-istio-controller-794dbdb89d-lnrll      1/1     Running   0          3m49s
knative-serving   net-istio-webhook-bbf6f8fcd-nlbls          1/1     Running   0          3m49s
knative-serving   webhook-658b566b8-42qbk                    1/1     Running   0          4m4s
kube-system       coredns-66bc5c9577-zlpmc                   1/1     Running   0          11m
kube-system       etcd-minikube                              1/1     Running   0          11m
kube-system       kube-apiserver-minikube                    1/1     Running   0          11m
kube-system       kube-controller-manager-minikube           1/1     Running   0          11m
kube-system       kube-proxy-lzd6f                           1/1     Running   0          11m
kube-system       kube-scheduler-minikube                    1/1     Running   0          11m
kube-system       storage-provisioner                        1/1     Running   0          11m
```

Install KServe CRDs

.\helm install kserve-crd oci://ghcr.io/kserve/charts/kserve-crd --version v0.15.2 --namespace kserve --create-namespace --wait

```
NAME: kserve-crd
LAST DEPLOYED: Fri Dec 26 23:46:46 2025
NAMESPACE: kserve
STATUS: deployed
REVISION: 1
DESCRIPTION: Install complete
TEST SUITE: None
```

Install Kserve resources

.\helm install kserve oci://ghcr.io/kserve/charts/kserve --version v0.15.2 --namespace kserve --create-namespace --wait --set-string kserve.controller.deploymentMode="Serverless"

```
PS D:\apps-data\docker> .\helm install kserve oci://ghcr.io/kserve/charts/kserve -
-version v0.15.2 --namespace kserve --create-namespace --wait --set-string kserve.
controller.deploymentMode="Serverless"
Pulled: ghcr.io/kserve/charts/kserve:v0.15.2
Digest: sha256:3f4e61bb603dd70c51fd6f352eb9bb1f40b06a2b29c138acccb8896d09b94fdd
I1227 00:28:07.988339   23516 warnings.go:110] "Warning: spec.privateKey.rotationP
olicy: In cert-manager >= v1.18.0, the default value changed from `Never` to `Alwa
ys`."
Error: INSTALLATION FAILED: Internal error occurred: failed calling webhook "clust
erservingruntime.kserve-webhook-server.validator": failed to call webhook: Post "h
ttps://kserve-webhook-server-service.kserve.svc:443/validate-serving-kserve-io-v1a
lpha1-clusterservingruntime?timeout=10s": EOF
Internal error occurred: failed calling webhook "clusterservingruntime.kserve-webh
ook-server.validator": failed to call webhook: Post "https://kserve-webhook-server
-service.kserve.svc:443/validate-serving-kserve-io-v1alpha1-clusterservingruntime?
timeout=10s": EOF
Internal error occurred: failed calling webhook "clusterservingruntime.kserve-webh
ook-server.validator": failed to call webhook: Post "https://kserve-webhook-server
-service.kserve.svc:443/validate-serving-kserve-io-v1alpha1-clusterservingruntime?
timeout=10s": EOF
Internal error occurred: failed calling webhook "clusterservingruntime.kserve-webh
ook-server.validator": failed to call webhook: Post "https://kserve-webhook-server
-service.kserve.svc:443/validate-serving-kserve-io-v1alpha1-clusterservingruntime?
timeout=10s": EOF
```

Because I tried so many times, I didn't take screenshots of the experiment. I only roughly recorded the commands that ran successfully, and it's likely that not all commands are included. This is only for reference.

1) Clean up

   .\helm uninstall kserve -n kserve

   .\helm uninstall kserve-crd -n kserve

   kubectl delete ns kserve --ignore-not-found

   # wait namespace disappear

   kubectl get ns kserve

2) Install CRD

   .\helm upgrade --install kserve-crd oci://ghcr.io/kserve/charts/kserve-crd `

--version v0.15.2 `
--namespace kserve `
--create-namespace `
--wait

3) Verify that the CRD already exists.
kubectl get crd | findstr /I "kserve.io"
kubectl get crd | findstr /I "clusterservingruntimes"
kubectl get crd | findstr /I "inferenceservices"

4) Download kserve chart（I've tried two methods here.）
   a. Download online and then install using the local package offline
   .\helm pull oci://ghcr.io/kserve/charts/kserve --version v0.15.2
   .\helm upgrade --install kserve .\kserve-v0.15.2.tgz `
   --namespace kserve --create-namespace `
   --set-string kserve.controller.deploymentMode=RawDeployment `
   --wait
   b. Rendering template
   .\helm template kserve oci://ghcr.io/kserve/charts/kserve `
   --version v0.15.2 `
   --namespace kserve `
   --set kserve.controller.deploymentMode=RawDeployment `
   > .\kserve-rendered.yaml
   Note: Applying the rendered YAML using `*kubectl apply*` only constitutes a "manual installation." Helm may fail to install subsequently due to ownership conflicts.

5) Verify webhook endpoints is normal
kubectl get pods -n kserve
kubectl get svc -n kserve kserve-webhook-server-service -o wide
kubectl get endpoints -n kserve kserve-webhook-server-service -o wide

kubectl get pods -n kserve

```
PS D:\apps-data\docker> kubectl get pods -n kserve
NAME                                         READY   STATUS    RESTARTS   AGE
kserve-controller-manager-8ffdc545-lzsk8     2/2     Running   0          16h
```

kubectl get crd | Select-String kserve

```
PS D:\apps-data\docker> kubectl get crd | Select-String kserve

clusterservingruntimes.serving.kserve.io             2025-12-27T15:43:21Z
clusterstoragecontainers.serving.kserve.io           2025-12-27T15:43:21Z
inferencegraphs.serving.kserve.io                    2025-12-27T15:43:21Z
inferenceservices.serving.kserve.io                  2025-12-27T15:43:22Z
localmodelcaches.serving.kserve.io                   2025-12-27T15:43:21Z
localmodelnodegroups.serving.kserve.io               2025-12-27T15:43:21Z
localmodelnodes.serving.kserve.io                    2025-12-27T15:43:21Z
servingruntimes.serving.kserve.io                    2025-12-27T15:43:21Z
trainedmodels.serving.kserve.io                      2025-12-27T15:43:21Z
```

kubectl get services -n kserve

```
PS D:\apps-data\docker> kubectl get services -n kserve
NAME                                 TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
kserve-controller-manager-service    ClusterIP   10.96.73.223    <none>        8443/TCP   17h
kserve-webhook-server-service        ClusterIP   10.103.41.56    <none>        443/TCP    17h
```

Deploy a predictive model

Create InferenceService（sklearn iris）

Create iris-isvc.yaml and apply it

```
@"
apiVersion: serving.kserve.io/v1beta1
kind: InferenceService
metadata:
  name: sklearn-iris
  namespace: default
spec:
  predictor:
    model:
      modelFormat:
        name: sklearn
      storageUri: gs://kfserving-examples/models/sklearn/1.0/model
"@ | kubectl apply -f -
```

```
PS D:\apps-data\docker> @"
>> apiVersion: serving.kserve.io/v1beta1
>> kind: InferenceService
>> metadata:
>>   name: sklearn-iris
>>   namespace: default
>> spec:
>>   predictor:
>>     model:
>>       modelFormat:
>>         name: sklearn
>>       storageUri: gs://kfserving-examples/models/sklearn/1.0/model
>> "@ | kubectl apply -f -
inferenceservice.serving.kserve.io/sklearn-iris created
```

Check InferenceService status

```
kubectl get inferenceservice sklearn-iris
```

```
PS D:\apps-data\docker> kubectl get inferenceservice sklearn-iris
NAME          URL                                        READY   PREV   LATEST   PREVROLLEDOUTREVISION   LATESTREADYREVISION   AGE
sklearn-iris   http://sklearn-iris-default.example.com   True                                                                  96m
```

Execute the following command to determine if the Kubernetes cluster is running in an environment that supports external load balancers

```
kubectl get svc istio-ingressgateway -n istio-system
```

```
PS D:\apps-data\docker> kubectl get svc istio-ingressgateway -n istio-system
NAME                   TYPE           CLUSTER-IP     EXTERNAL-IP   PORT(S)
                                AGE
istio-ingressgateway   LoadBalancer   10.108.23.14   <pending>     15021:32389/TCP
,80:31886/TCP,443:32614/TCP   26h
```

Port Forward

Find the Service name of istio ingressgateway (usually istio-ingressgateway).

$INGRESS_GATEWAY_SERVICE = kubectl get svc -n istio-system -l app=istio-ingressgateway `
-o jsonpath="{.items[0].metadata.name}"

Do Port Forward

kubectl port-forward -n istio-system "svc/$INGRESS_GATEWAY_SERVICE" 8080:80

```
PS D:\apps-data\docker> $INGRESS_GATEWAY_SERVICE = kubectl get svc -n istio-system
 -l app=istio-ingressgateway `
>>   -o jsonpath="{.items[0].metadata.name}"
PS D:\apps-data\docker> $INGRESS_GATEWAY_SERVICE
istio-ingressgateway
PS D:\apps-data\docker> kubectl port-forward -n istio-system "svc/$INGRESS_GATEWAY
_SERVICE" 8080:80
Forwarding from 127.0.0.1:8080 -> 8080
Forwarding from [::1]:8080 -> 8080
```

kubectl get inferenceservice sklearn-iris -o jsonpath='{.status.url}'

```
PS D:\apps-data\docker> kubectl get inferenceservice sklearn-iris -o jsonpath='{.s
tatus.url}'
http://sklearn-iris-default.example.com
```

Open another terminal, and enter the following to perform inference

$SERVICE_URL = kubectl get inferenceservice sklearn-iris -o jsonpath="{.status.url}"
$SERVICE_HOSTNAME = ([Uri]$SERVICE_URL).Host

```
PS D:\apps-data\docker> $SERVICE_URL = kubectl get inferenceservice sklearn-iris -
o jsonpath="{.status.url}"
PS D:\apps-data\docker> $SERVICE_HOSTNAME = ([Uri]$SERVICE_URL).Host
PS D:\apps-data\docker>
PS D:\apps-data\docker> $SERVICE_URL
http://sklearn-iris-default.example.com
PS D:\apps-data\docker> $SERVICE_HOSTNAME
sklearn-iris-default.example.com
```

curl.exe -v `
-H "Host: $SERVICE_HOSTNAME" `
-H "Content-Type: application/json" `
"http://${INGRESS_HOST}:${INGRESS_PORT}/v1/models/sklearn-iris:predict" `
--data-binary "@.\iris-input.json"

```
PS D:\apps-data\docker> curl.exe -v `
>>   -H "Host: $SERVICE_HOSTNAME" `
>>   -H "Content-Type: application/json" `
>>   "http://${INGRESS_HOST}:${INGRESS_PORT}/v1/models/sklearn-iris:predict" `
>>   --data-binary "@.\iris-input.json"
* Host localhost:8080 was resolved.
* IPv6: ::1
* IPv4: 127.0.0.1
*   Trying [::1]:8080...
* Established connection to localhost (::1 port 8080) from ::1 port 62018
* using HTTP/1.x
> POST /v1/models/sklearn-iris:predict HTTP/1.1
> Host: sklearn-iris-default.example.com
> User-Agent: curl/8.16.0
> Accept: */*
> Content-Type: application/json
> Content-Length: 54
>
* upload completely sent off: 54 bytes
< HTTP/1.1 404 Not Found
< date: Sun, 28 Dec 2025 13:23:39 GMT
< server: istio-envoy
< connection: close
< content-length: 0
<
* shutting down connection #0
```

kubectl port-forward -n istio-system svc/knative-local-gateway 8080:80

```
PS D:\apps-data\docker> kubectl port-forward -n istio-system svc/knative-local-gat
eway 8080:80
Forwarding from 127.0.0.1:8080 -> 8081
Forwarding from [::1]:8080 -> 8081
```

curl.exe -v `
-H "Host: $SERVICE_HOSTNAME" `
-H "Content-Type: application/json" `
--data-raw '{"instances":[[5.1,3.5,1.4,0.2]]}' `
http://localhost:8080/v1/models/sklearn-iris:predict

```
PS D:\apps-data\docker> curl.exe -v `
>>   -H "Host: $SERVICE_HOSTNAME" `
>>   -H "Content-Type: application/json" `
>>   --data-raw '{"instances":[[5.1,3.5,1.4,0.2]]}' `
>>   "http://localhost:8080/v1/models/sklearn-iris:predict"
* Host localhost:8080 was resolved.
* IPv6: ::1
* IPv4: 127.0.0.1
*   Trying [::1]:8080...
* Established connection to localhost (::1 port 8080) from ::1 port 50113
* using HTTP/1.x
> POST /v1/models/sklearn-iris:predict HTTP/1.1
> Host: sklearn-iris-default.example.com
> User-Agent: curl/8.16.0
> Accept: */*
> Content-Type: application/json
> Content-Length: 31
>
* upload completely sent off: 31 bytes
< HTTP/1.1 404 Not Found
< date: Sun, 28 Dec 2025 13:51:14 GMT
< server: istio-envoy
< connection: close
< content-length: 0
<
* shutting down connection #0
```

Port Forward to Predictor Service

kubectl -n default port-forward svc/sklearn-iris-predictor 18080:80

```
PS D:\apps-data\docker> kubectl -n default port-forward svc/sklearn-iris-predictor
 18080:80
Forwarding from 127.0.0.1:18080 -> 8080
Forwarding from [::1]:18080 -> 8080
```

```
PS D:\apps-data\docker> curl.exe -v `
>>   -H "Content-Type: application/json" `
>>   --data-binary "@iris-input.json" `
>>   "http://127.0.0.1:18080/v1/models/sklearn-iris:predict"
*   Trying 127.0.0.1:18080...
* Established connection to 127.0.0.1 (127.0.0.1 port 18080) from 127.0.0.1 port 5
1587
* using HTTP/1.x
> POST /v1/models/sklearn-iris:predict HTTP/1.1
> Host: 127.0.0.1:18080
> User-Agent: curl/8.16.0
> Accept: */*
> Content-Type: application/json
> Content-Length: 33
>
* upload completely sent off: 33 bytes
< HTTP/1.1 200 OK
< date: Sun, 28 Dec 2025 14:14:57 GMT
< server: uvicorn
< content-length: 19
< content-type: application/json
<
{"predictions":[0]}* Connection #0 to host 127.0.0.1:18080 left intact
```