Lab9

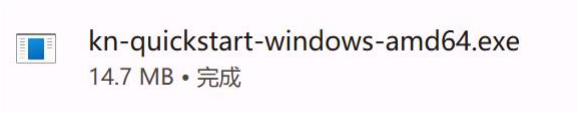
Quickstart

1. Download kn from github page

2. Rename and add to local path:

```
PS C:\Windows\System32> kn version
Version:      v1.19.6
Build Date:   2025-10-28 10:48:39
Git Revision: 4a791c5d
Supported APIs:
* Serving
  - serving.knative.dev/v1 (knative-serving v1.19.6)
* Eventing
  - sources.knative.dev/v1 (knative-eventing v1.19.6)
  - eventing.knative.dev/v1 (knative-eventing v1.19.6)
```

3. Download quickstart



4. Running quickstart to:
   Checks if you have the selected Kubernetes instance installed
   -Creates a cluster called knative
   -Installs Knative Serving with Kourier as the default networking layer, and sslip.io as the DNS
   -Installs Knative Eventing and creates an in-memory Broker and Channel implementation

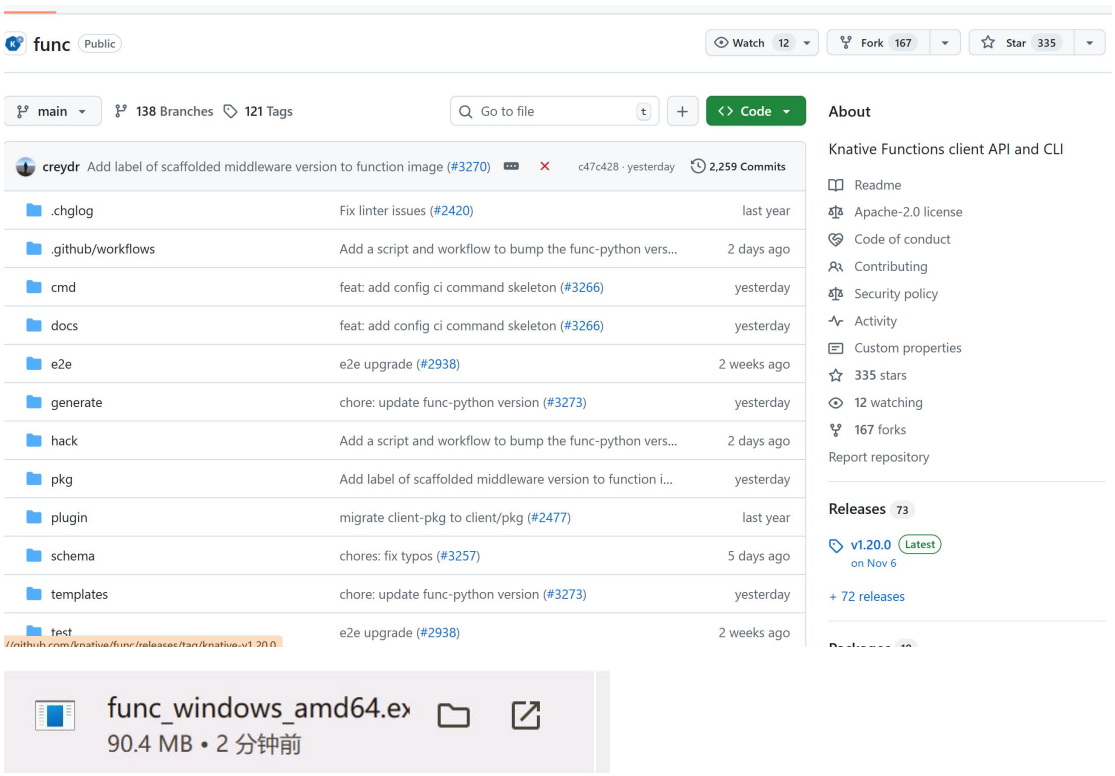
```
PS C:\study\云计算导论\Labs\Lab9> .\kn-quickstart-windows-amd64.exe minikube
Running Knative Quickstart using Minikube
Minikube version is: v1.37.0

Knative Cluster knative already installed.
Delete and recreate [y/N]: y
deleting cluster...
Creating Minikube cluster...

Using the standard minikube driver for your system
If you wish to use a different driver, please configure minikube using
    minikube config set driver <your-driver>

    Microsoft Windows 11 Home China 10.0.26100.7171 Build 26100.7171 上的 [knative] minikube v1.37.0
    自动选择 docker 驱动。其他选项: virtualbox, ssh
    使用具有 root 权限的 Docker Desktop 驱动程序
    在集群中 "knative" 启动节点 "knative" primary control-plane
    正在拉取基础镜像 v0.0.48 ...
    > gcr.io/k8s-minikube/kicbase...:  488.52 MiB / 488.52 MiB  100.00% 9.65 Mi
    创建 docker container (CPU=3, 内存=3072MB) ...
    从 Minikube 的 container 内部连接到 https://registry.k8s.io/ 失败
    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/
    正在 Docker 28.4.0 中准备 Kubernetes v1.32.0…
    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

    注册表插件 docker Driver 使用端口 45257 代替默认端口 5000

    更多信息，请参阅 https://minikube.sigs.k8s.io/docs/drivers/docker
    ▪ 正在使用镜像 gcr.io/k8s-minikube/kube-registry-proxy:0.0.9
    ▪ 正在使用镜像 docker.io/registry:3.0.0
    正在验证 registry 插件...
    启用插件:  storage-provisioner, default-storageclass, registry
    完成！ kubectl 现在已配置，默认使用"knative"集群和"default"命名空间


To finish setting up networking for minikube, run the following command in a separate terminal window:
    minikube tunnel --profile knative
The tunnel command must be running in a terminal window any time when using the knative quickstart environment.

Press the Enter key to continue
 Installing Knative Serving v1.20.0 ...
```

5. Use minikube tunnel:

```
PS C:\Windows\System32> minikube tunnel -p knative
* 隧道成功启动

* 注意: 请不要关闭此终端，因为此进程必须保持活动状态才能访问隧道.....

! 在 Windows 上使用 v8.1以上版本的OpenSSH客户端，访问 1024 以下端口可能会失败。更多信息请参阅: https://minikube.s
.io/docs/handbook/accessing/#access-to-ports-1024-on-windows-requires-root-permission
```

Installing Knative Functions

1. Download func cli:



Creating a function

After you have installed Knative Functions, you can create a function project by using the func CLI or the kn func plugin:

func create -l <language> <function-name>

```
PS C:\study\云计算导论\Labs\Lab9> .\func_windows_amd64.exe create -l go hello
Created go function in C:\study\云计算导论\Labs\Lab9\hello
```

Knative Serving

Deploying a Knative Service

1. Deploy the Service by running the command:
   kn service create hello --image ghcr.io/knative/helloworld-go:latest --port 8080 --env TARGET=World

```
PS C:\study\云计算导论\Labs\Lab9> kn service create hello --image ghcr.io/knative/helloworld
-go:latest --port 8080 --env TARGET=World
Warning: Kubernetes default value is insecure, Knative may default this to secure in a futur
e release: spec.template.spec.containers[0].securityContext.allowPrivilegeEscalation, spec.t
emplate.spec.containers[0].securityContext.capabilities, spec.template.spec.containers[0].se
curityContext.runAsNonRoot, spec.template.spec.containers[0].securityContext.seccompProfile
Creating service 'hello' in namespace 'default':

  0.333s The Route is still working to reflect the latest desired specification.
  0.418s ...
  0.440s Configuration "hello" is waiting for a Revision to become ready.
 21.181s ...
 21.309s Ingress has not yet been reconciled.
 21.488s Waiting for load balancer to be ready
 21.610s Ready to serve.

Service 'hello' created to latest revision 'hello-00001' is available at URL:
http://hello.default.svc.cluster.local
```

Autoscaling

1. View a list of Knative Services by running the command:

   kn service list

```
PS C:\study\云计算导论\Labs\Lab9> kn service list
NAME    URL                                   LATEST        AGE    CONDITIONS   READY   R
EASON
hello   http://hello.default.svc.cluster.local   hello-00001   3m8s   3 OK / 3     True
```

2. Access your Knative Service by opening the previous URL in your browser or by running the command:

   echo "Accessing URL $(kn service describe hello -o url)"

   curl "$(kn service describe hello -o url)"

```
PS C:\study\云计算导论\Labs\Lab9> kn service describe hello -o url
http://hello.default.svc.cluster.local
```

```
PS C:\study\云计算导论\Labs\Lab9> curl http://hello.default.svc.cluster.local
curl : 远程服务器返回错误：(502) 错误的网关。
所在位置 行:1 字符: 1
+ curl http://hello.default.svc.cluster.local
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : InvalidOperation: (System.Net.HttpWebRequest:HttpWebRequest)
    [Invoke-WebRequest], WebException
    + FullyQualifiedErrorId : WebCmdletWebResponseException,Microsoft.PowerShell.Commands.
   InvokeWebRequestCommand
```

   Not work. Let's use the post forward method:

3. Forward kourier:

```
PS C:\study\云计算导论\Labs\Lab9> kubectl port-forward -n kourier-system svc/kourier-interna
l 8080:80
Forwarding from 127.0.0.1:8080 -> 8081
Forwarding from [::1]:8080 -> 8081
```

4. Access the address:

```
PS C:\Windows\System32> curl http://localhost:8080 -Headers @{ Host = "hello.default.svc.cluster.local" }


StatusCode        : 200
StatusDescription : OK
Content           : Hello World!

RawContent        : HTTP/1.1 200 OK
                    x-envoy-upstream-service-time: 3895
                    Content-Length: 13
                    Content-Type: text/plain; charset=utf-8
                    Date: Mon, 08 Dec 2025 01:59:17 GMT
                    Server: envoy

                    Hello World!

Forms             : {}
Headers           : {[x-envoy-upstream-service-time, 3895], [Content-Length, 13], [Content-Type, text/plain; charset=ut
                    f-8], [Date, Mon, 08 Dec 2025 01:59:17 GMT]...}
Images            : {}
InputFields       : {}
Links             : {}
ParsedHtml        : mshtml.HTMLDocumentClass
RawContentLength  : 13
```

5. Watch the pods and see how they scale to zero after traffic stops going to the URL:

```
PS C:\Windows\System32> kubectl get pod -l serving.knative.dev/service=hello -W
NAME                                     READY   STATUS    RESTARTS   AGE
hello-00001-deployment-5487f4dd76-lkdsf   2/2     Running   0          57s
hello-00001-deployment-5487f4dd76-lkdsf   2/2     Terminating   0          66s
hello-00001-deployment-5487f4dd76-lkdsf   1/2     Terminating   0          94s
hello-00001-deployment-5487f4dd76-lkdsf   0/2     Completed     0          97s
hello-00001-deployment-5487f4dd76-lkdsf   0/2     Completed     0          97s
hello-00001-deployment-5487f4dd76-lkdsf   0/2     Completed     0          97s
```

Knative Eventing

Sources, Brokers, and Triggers

1. Verify that the Broker is installed by running the following command:
   kn broker list

```
PS C:\Windows\System32> kn broker list
NAME             URL                                                                  AGE   CONDITIONS   RE
ADY   REASON
example-broker   http://broker-ingress.knative-eventing.svc.cluster.local/default/example-broker   30m   7 OK / 7     Tr
ue
```

Using a Knative Service as a source

Creating your first source

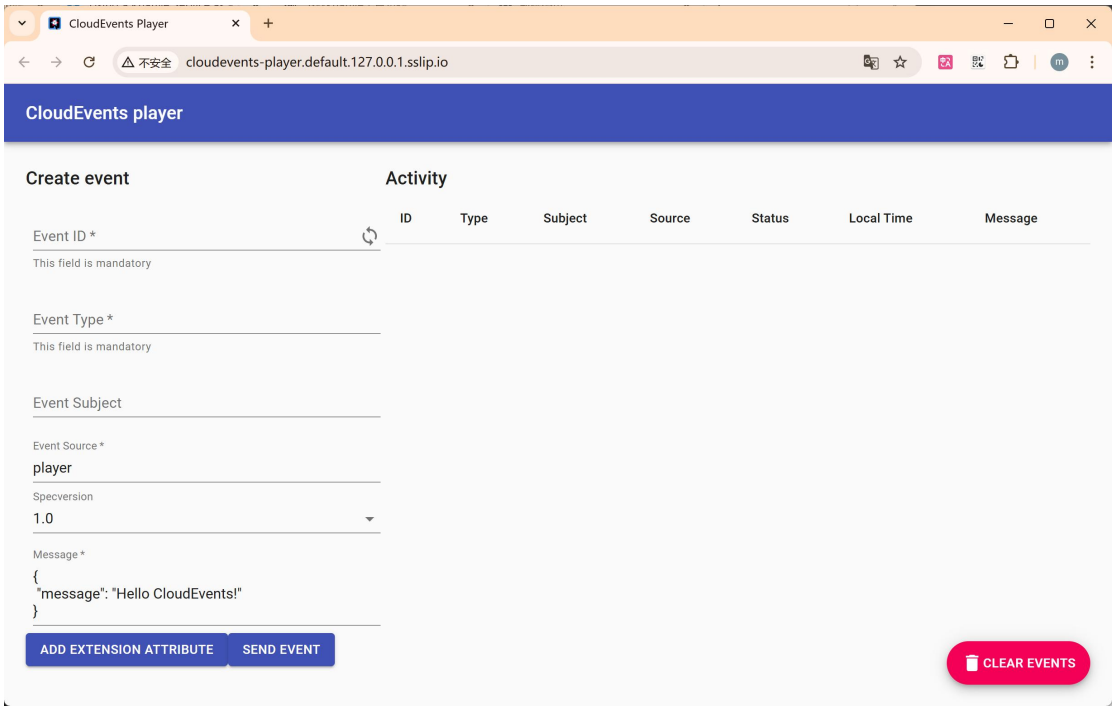
1. Create the CloudEvents Player Service:
   kn service create cloudevents-player --image
   quay.io/ruben/cloudevents-player:latest
2. The service is now running but it doesn't know where the broker is so let's create a SinkBinding between the service and the broker.
   kn source binding create ce-player-binding --subject
   "Service:serving.knative.dev/v1:cloudevents-player" --sink broker:example-broker

```
PS C:\Windows\System32> kn service create cloudevents-player --image quay.io/ruben/cloudevents-player:latest
Warning: Kubernetes default value is insecure, Knative may default this to secure in a future release: spec.template.spe
c.containers[0].securityContext.allowPrivilegeEscalation, spec.template.spec.containers[0].securityContext.capabilities,
 spec.template.spec.containers[0].securityContext.runAsNonRoot, spec.template.spec.containers[0].securityContext.seccomp
Profile
Creating service 'cloudevents-player' in namespace 'default':

  0.082s The Route is still working to reflect the latest desired specification.
  0.179s Configuration "cloudevents-player" is waiting for a Revision to become ready.
 35.059s ...
 35.181s Ingress has not yet been reconciled.
 35.299s Waiting for load balancer to be ready
 35.447s Ready to serve.

Service 'cloudevents-player' created to latest revision 'cloudevents-player-00001' is available at URL:
http://cloudevents-player.default.svc.cluster.local
PS C:\Windows\System32> kn source binding create ce-player-binding --subject "Service:serving.knative.dev/v1:cloudevents
-player" --sink broker:example-broker
Sink binding 'ce-player-binding' created in namespace 'default'.
```

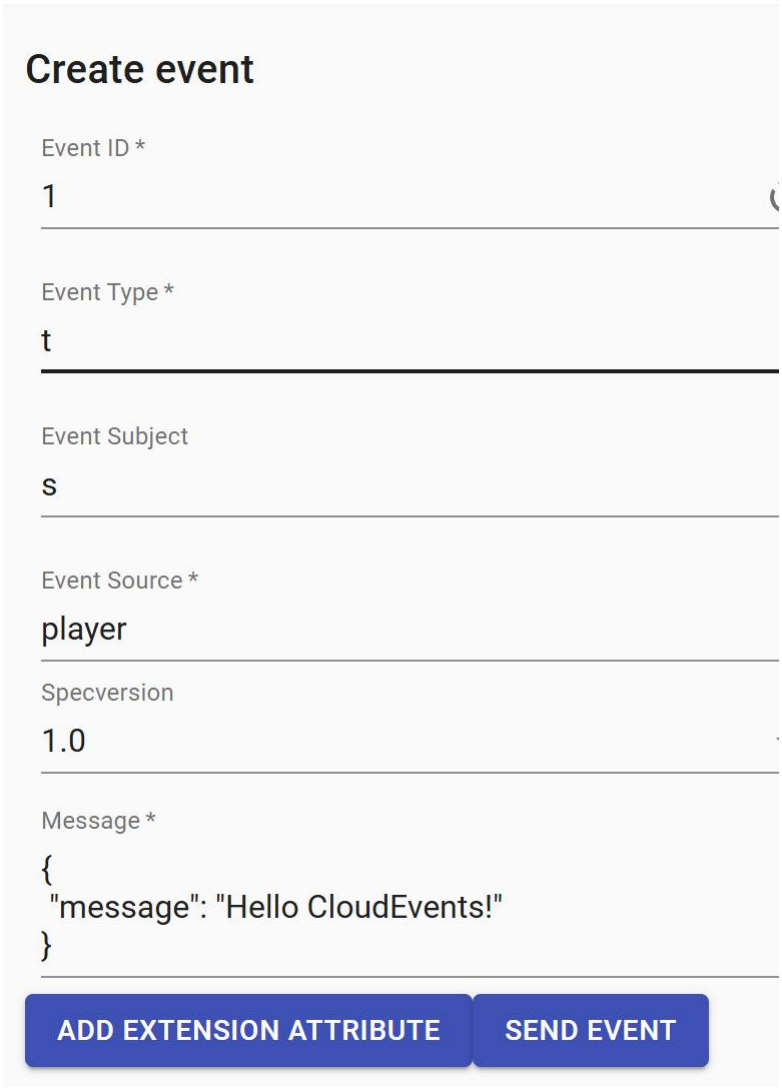
3. Check loadbalancer ip:

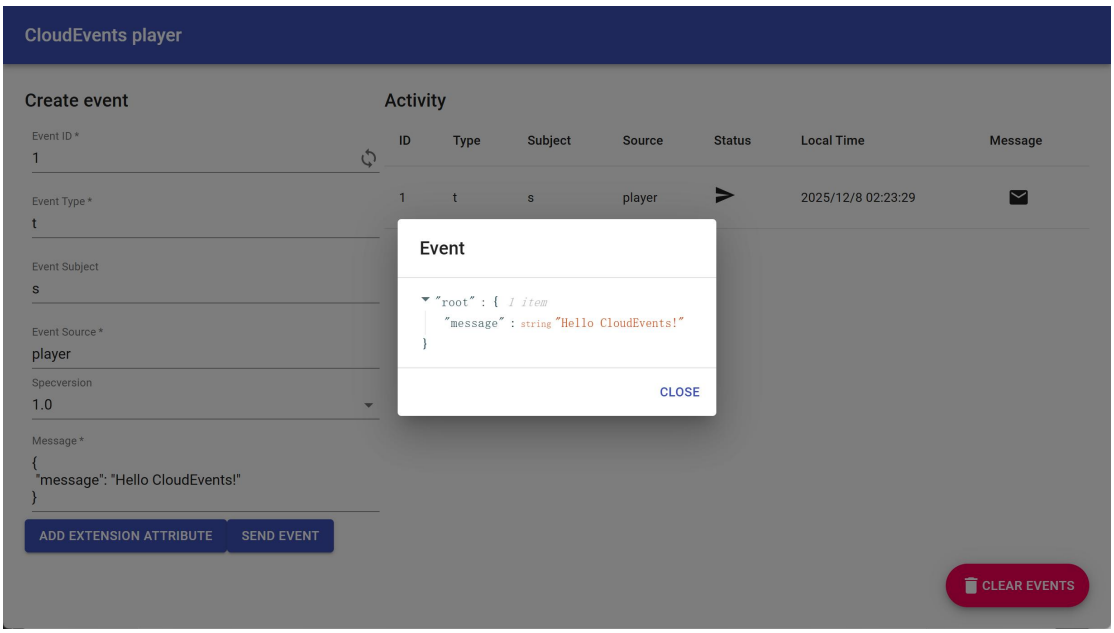
```
PS C:\Users\32009> kubectl get svc -n kourier-system
NAME                TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)                      AGE
kourier             LoadBalancer   10.96.182.151   127.0.0.1     80:32117/TCP,443:31936/TCP   51m
kourier-internal    ClusterIP      10.97.7.147     <none>        80/TCP,443/TCP               51m
```

4. Access http://cloudevents-player.default.127.0.0.1.sslip.io:



5. Sending an event:


Create event
Event ID *
1
Event Type *
t
Event Subject
s
Event Source *
player
Specversion
1.0
Message *
{
"message": "Hello CloudEvents!"
}
ADD EXTENSION ATTRIBUTE
SEND EVENT


CloudEvents player
Create event
Event ID *
1
Event Type *
t
Event Subject
s
Event Source *
player
Specversion
1.0
Message *
{
"message": "Hello CloudEvents!"
}
ADD EXTENSION ATTRIBUTE
SEND EVENT
Activity
ID
Type
Subject
Source
Status
Local Time
Message
1
t
s
player
2025/12/8 02:23:29
Event
"root" : { 1 item
"message" : string "Hello CloudEvents!"
}
CLOSE
CLEAR EVENTS

Using Triggers and sinks

1. Creating your first Trigger:
   kn trigger create cloudevents-trigger --sink cloudevents-player --broker example-broker

```
PS C:\Users\32009> kn trigger create cloudevents-trigger --sink cloudevents-player  --broker example-broker
Trigger 'cloudevents-trigger' successfully created in namespace 'default'.
```

2. Now, when we go back to the CloudEvents Player and send an event, we see that CloudEvents are both sent and received by the CloudEvents Player: