

Microservice Application

I have chosen **Microservice-Kubernetes Sample**.

[GitHub - ewolff/microservice-kubernetes: Microservices example using Kubernetes](https://github.com/ewolff/microservice-kubernetes)

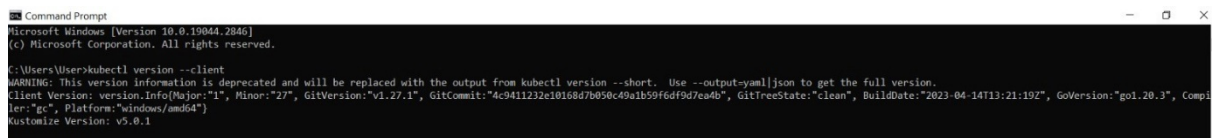
Minikube setup and Installation

First of all, I have installed kubectl (Kubernetes control tool) which is responsible to communicate with the cluster and send instructions to it.

Steps to download Kubectl on local machine:

1. Download **chocolatey** package manager for Windows
 - I opened powershell as an administrator then run the following command: `Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1'))`
 - Open cmd as an administrator, then run the following command: `choco install kubernetes-cli`

2. Kubectl is running on my system.



```
Microsoft Windows [Version 10.0.19044.2846]
(c) Microsoft Corporation. All rights reserved.

C:\Users\User>kubectl version --client
WARNING: This version information is deprecated and will be replaced with the output from kubectl version --short. Use --output=yaml|json to get the full version.
Client Version: version.Info{Major:"1", Minor:"27", GitVersion:"v1.27.1", GitCommit:"4c9411232c10168d7b050c491b59f6d967c04b", GitTreeState:"clean", BuildDate:"2023-04-14T13:21:19Z", GoVersion:"go1.20.3", Compiler:"gc", Platform:"windows/amd64"}
Customize Version: v5.0.1
```

```
Command Prompt
Microsoft Windows [Version 10.0.19044.2846]
(c) Microsoft Corporation. All rights reserved.

C:\Users\User>kubectl cluster-info
Kubernetes control plane is running at https://192.168.59.100:8443
CoreDNS is running at https://192.168.59.100:8443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.

C:\Users\User>
```

Figure 12: Cluster State

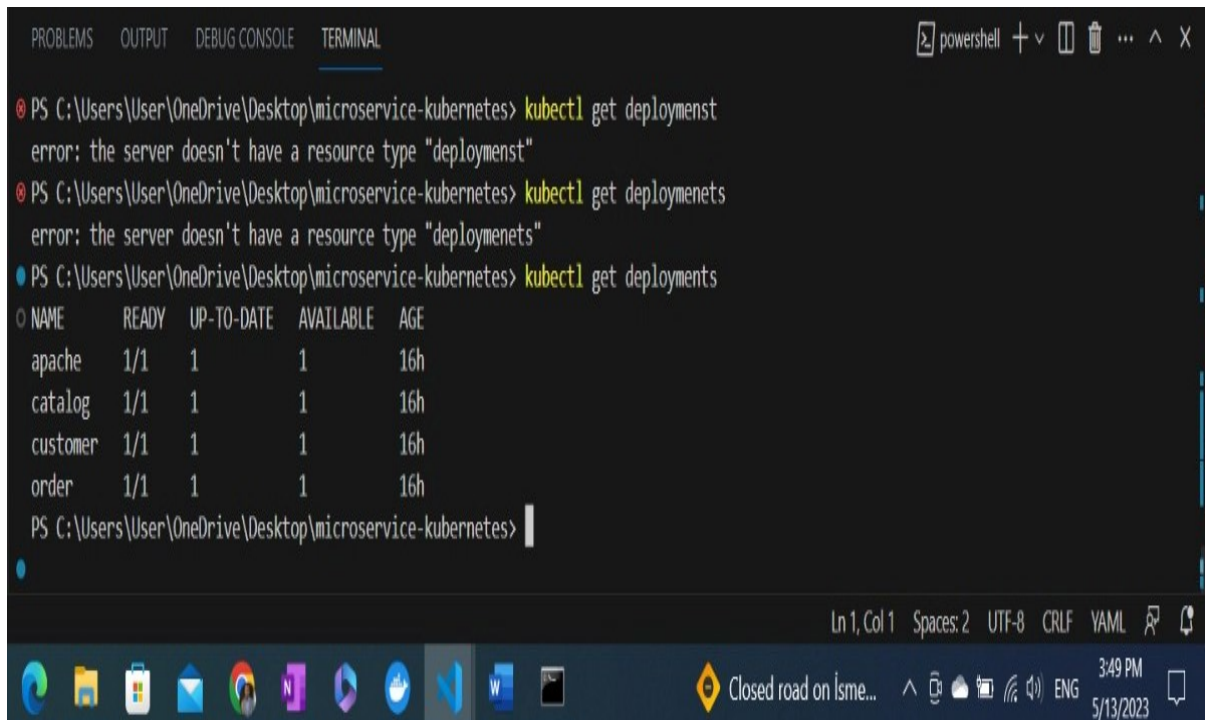
3. The image below indicates that minikube is running on my machine.

```
Command Prompt
Microsoft Windows [Version 10.0.19044.2846]
(c) Microsoft Corporation. All rights reserved.

C:\Users\User>minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured
```

Deployment of the Application

1. Run **scaffold init** command to go through my application and looks if there are any configuration files such as docker, pom.xml, etc.,
2. Run **scaffold dev** to build and deploy my application.
3. The image below shows my deployable application using **kubectl get deployments** command

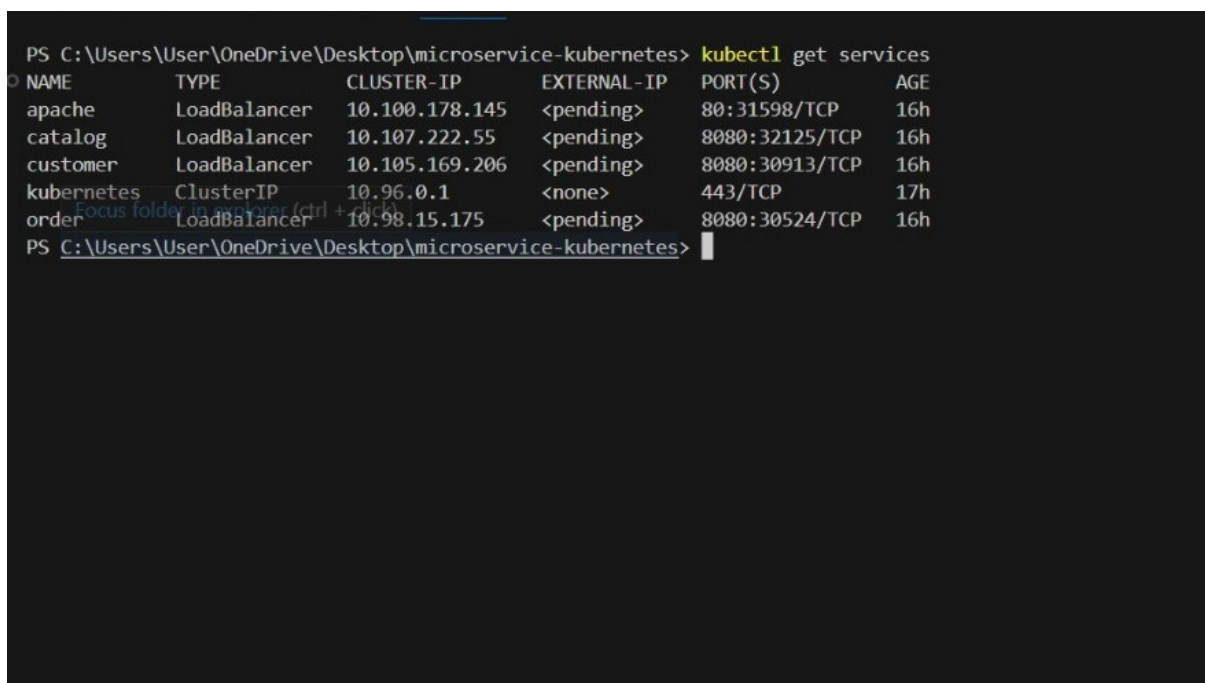


```
PS C:\Users\User\OneDrive\Desktop\microservice-kubernetes> kubectl get deploymentst
error: the server doesn't have a resource type "deploymentst"
PS C:\Users\User\OneDrive\Desktop\microservice-kubernetes> kubectl get deploymenets
error: the server doesn't have a resource type "deploymenets"
PS C:\Users\User\OneDrive\Desktop\microservice-kubernetes> kubectl get deployments
```

| NAME | READY | UP-TO-DATE | AVAILABLE | AGE |
|----------|-------|------------|-----------|-----|
| apache | 1/1 | 1 | 1 | 16h |
| catalog | 1/1 | 1 | 1 | 16h |
| customer | 1/1 | 1 | 1 | 16h |
| order | 1/1 | 1 | 1 | 16h |

```
PS C:\Users\User\OneDrive\Desktop\microservice-kubernetes>
```

4. The image below shows services using **kubectl get services** command

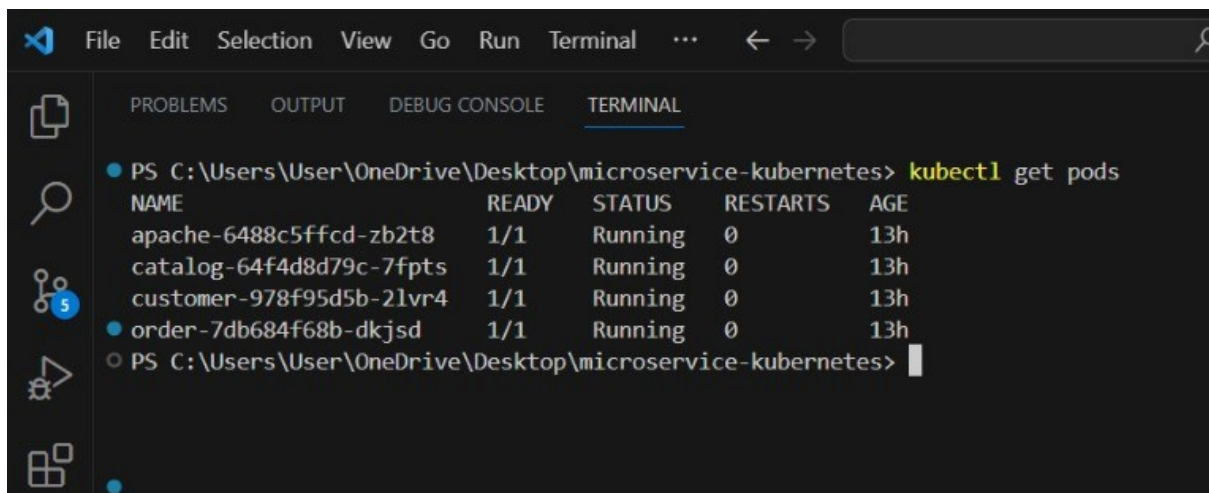


```
PS C:\Users\User\OneDrive\Desktop\microservice-kubernetes> kubectl get services
```

| NAME | TYPE | CLUSTER-IP | EXTERNAL-IP | PORT(S) | AGE |
|------------|--------------|----------------|-------------|----------------|-----|
| apache | LoadBalancer | 10.100.178.145 | <pending> | 80:31598/TCP | 16h |
| catalog | LoadBalancer | 10.107.222.55 | <pending> | 8080:32125/TCP | 16h |
| customer | LoadBalancer | 10.105.169.206 | <pending> | 8080:30913/TCP | 16h |
| kubernetes | ClusterIP | 10.96.0.1 | <none> | 443/TCP | 17h |
| order | LoadBalancer | 10.98.15.175 | <pending> | 8080:30524/TCP | 16h |

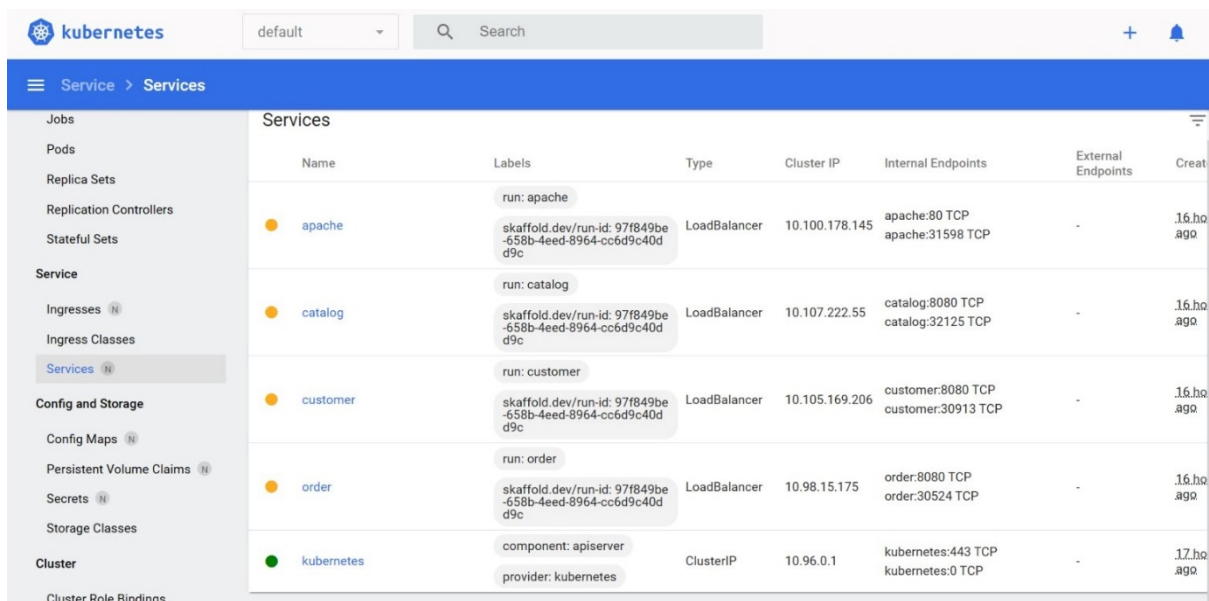
```
PS C:\Users\User\OneDrive\Desktop\microservice-kubernetes>
```

5. The image below shows pods using **kubectl get pods** command



```
PS C:\Users\User\OneDrive\Desktop\microservice-kubernetes> kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
apache-6488c5ffcd-zb2t8             1/1     Running   0           13h
catalog-64f4d8d79c-7fpts            1/1     Running   0           13h
customer-978f95d5b-2lvr4            1/1     Running   0           13h
order-7db684f68b-dkjsd              1/1     Running   0           13h
PS C:\Users\User\OneDrive\Desktop\microservice-kubernetes>
```

6. The image below shows services using the Kubernetes dashboard (using running **minikube dashboard** in the regular CMD)



| Name | Labels | Type | Cluster IP | Internal Endpoints | External Endpoints | Creation Time |
|------------|--|--------------|----------------|---|--------------------|---------------|
| apache | run: apache skaffold.dev/run-id: 97f849be-658b-4eed-8964-cc6d9c40dd9c | LoadBalancer | 10.100.178.145 | apache:80 TCP apache:31598 TCP | - | 16.ho ago |
| catalog | run: catalog skaffold.dev/run-id: 97f849be-658b-4eed-8964-cc6d9c40dd9c | LoadBalancer | 10.107.222.55 | catalog:8080 TCP catalog:32125 TCP | - | 16.ho ago |
| customer | run: customer skaffold.dev/run-id: 97f849be-658b-4eed-8964-cc6d9c40dd9c | LoadBalancer | 10.105.169.206 | customer:8080 TCP customer:30913 TCP | - | 16.ho ago |
| order | run: order skaffold.dev/run-id: 97f849be-658b-4eed-8964-cc6d9c40dd9c | LoadBalancer | 10.98.15.175 | order:8080 TCP order:30524 TCP | - | 16.ho ago |
| kubernetes | component: apiserver provider: kubernetes | ClusterIP | 10.96.0.1 | kubernetes:443 TCP kubernetes:0 TCP | - | 17.ho ago |

Changing to the application's frontend

Scaffold dev will deploy the application and containers after each change in the code automatically. To observe this, I have run **scaffold dev** in the terminal, then I made some changes in the code (frontend). While I am making these changes, the command rebuilt and re-deployed the application automatically without doing anything else.

5 Exposing the application to run as localhost from my web browser.

I have faced a problem regarding the exposing of my application as localhost which is as the following:

"Because you are using a docker driver on windows, the terminal needs to be open to run it.

```
PS C:\Users\User\OneDrive\Desktop\microservice-kubernetes> minikube service customer
|-----|-----|-----|-----|
| NAMESPACE | NAME   | TARGET PORT | URL               |
|-----|-----|-----|-----|
| default    | customer | 8080        | http://192.168.49.2:30790 |
|-----|-----|-----|-----|
✨ Starting tunnel for service customer.
✨ Starting tunnel for service customer.
✨ Starting tunnel for service customer.
✨ Starting tunnel for service customer.
✨ Starting tunnel for service customer.
|-----|-----|-----|-----|
| NAMESPACE | NAME   | TARGET PORT | URL               |
|-----|-----|-----|-----|
| default    | customer |             | http://127.0.0.1:58026 |
|-----|-----|-----|-----|
🔗 Opening service default/customer in default browser...
docker@127.0.0.1's password: ! Because you are using a Docker driver on windows, the terminal needs to be open to run it.
docker@127.0.0.1's password:
docker@127.0.0.1's password:
```

I have solved the problem by running the following command : **minikube tunnel**. This will open a terminal for each service in the deployment process. It requires to enter a password to run it.

The images below illustrate what I mean

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL minikube + v [ ] ... v

PS C:\Users\User\OneDrive\Desktop\microservice-kubernetes> minikube tunnel
[✓] Tunnel successfully started

★ NOTE: Please do not close this terminal as this process must stay alive for the tunnel to be accessible ...

! Access to ports below 1024 may fail on Windows with OpenSSH clients older than v8.1. For more information, see: https://minik
! Access to ports below 1024 may fail on Windows with OpenSSH clients older than v8.1. For more information, see: https://minik
! Access to ports below 1024 may fail on Windows with OpenSSH clients older than v8.1. For more information, see: https://minik
! Access to ports below 1024 may fail on Windows with OpenSSH clients older than v8.1. For more information, see: https://minikube.sigs.k8s.io/docs/handbook/accessing/#acc
! Access to ports below 1024 may fail on Windows with OpenSSH clients older than v8.1. For more information, see: https://minikube.sigs.k8s.io/docs/handbook/accessing/#acc
! Access to ports below 1024 may fail on Windows with OpenSSH clients older than v8.1. For more information, see: https://minikube.sigs.k8s.io/docs/handbook/accessing/#acc
! Access to ports below 1024 may fail on Windows with OpenSSH clients older than v8.1. For more information, see: https://minikube.sigs.k8s.io/docs/handbook/accessing/#acc
es-to-ports-1024-on-windows-requires-root-permission
ss-to-ports-1024-on-windows-requires-root-permission
★ Starting tunnel for service apache.
★ Starting tunnel for service catalog.
★ Starting tunnel for service customer.
★ Starting tunnel for service order.
★ Starting tunnel for service scaffold-demo-svc.
docker@127.0.0.1's password: docker@127.0.0.1's password: docker@127.0.0.1's password: docker@127.0.0.1's password: docker@127.0.0.1's password:
docker@127.0.0.1's password:
docker@127.0.0.1's password:
[ ]
```

Kubectl get endpoints

```
PS C:\Users\User\OneDrive\Desktop\microservice-kubernetes> kubectl get endpoints
NAME           ENDPOINTS             AGE
apache         10.244.0.11:80        6d16h
catalog        10.244.0.12:8080      6d16h
customer       10.244.0.13:8080      6d16h
kubernetes     192.168.49.2:8443     6d16h
order          10.244.0.14:8080      6d16h
scaffold-demo-svc <none>                6d16h
PS C:\Users\User\OneDrive\Desktop\microservice-kubernetes> [ ]
```


Kubectl get service

```
PS C:\Users\User\OneDrive\Desktop\microservice-kubernetes> kubectl get services
NAME                TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
apache              LoadBalancer  10.104.247.110 127.0.0.1      80:31540/TCP     6d16h
catalog             LoadBalancer  10.100.113.131 127.0.0.1      8080:30082/TCP   6d16h
customer            LoadBalancer  10.104.202.152 127.0.0.1      8080:30790/TCP   6d16h
kubernetes           ClusterIP      10.96.0.1       <none>         443/TCP          6d16h
order               LoadBalancer  10.108.89.134  127.0.0.1      8080:30261/TCP   6d16h
skaffold-demo-svc   LoadBalancer  10.108.51.244  127.0.0.1      8080:32689/TCP   6d16h
PS C:\Users\User\OneDrive\Desktop\microservice-kubernetes> █
```

Kubectl get pods

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\Users\User\OneDrive\Desktop\microservice-kubernetes> kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
apache-6d98fdcff4-c6bf6             1/1     Running   1 (20h ago) 6d16h
catalog-849f964c7-l4h6t             1/1     Running   1 (20h ago) 6d16h
customer-c7986bb7b-fm7m2            1/1     Running   1 (20h ago) 6d16h
order-84fcb5c774-7pfjv              1/1     Running   1 (20h ago) 6d16h
PS C:\Users\User\OneDrive\Desktop\microservice-kubernetes> █
```

Kubectl get deployments

```
PS C:\Users\User\OneDrive\Desktop\microservice-kubernetes> kubectl get deployments
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
apache        1/1     1             1           6d23h
catalog       1/1     1             1           6d23h
customer      1/1     1             1           6d23h
order         1/1     1             1           6d23h
PS C:\Users\User\OneDrive\Desktop\microservice-kubernetes> █
```