

# **OpenShift Developer Distance Learning Program**

Homework Assignment

Silke Roth

## Why did I choose serverless?

Minimizes resource usage – only deployed if required, application will be idle as long as it is not required

**And:**

New for me during this training

Curiosity - Will this be as easy as described?

## Scenario – Serverless application

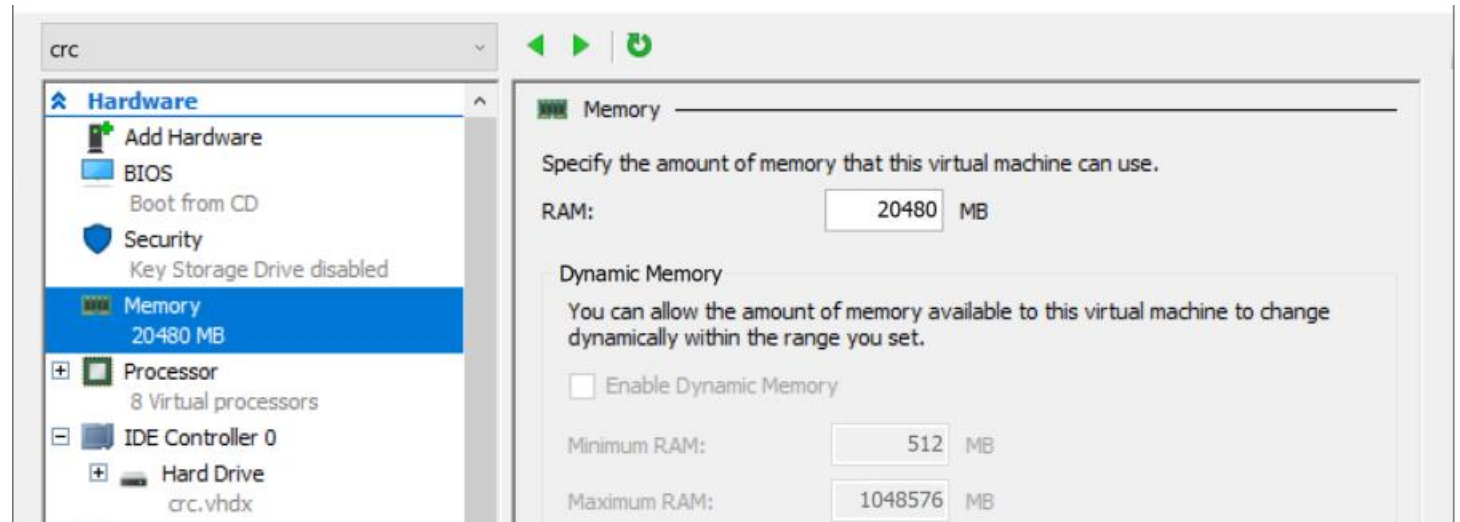
This scenario will demonstrate a simple serverless application with the following steps

- Create an instance of CodeReady Workspace on Windows
- Prepare the environment
- Create a serverless application, based sample nginx application on <https://github.com/sclorg/nginx-ex.git>
- Test and monitor the application

# Setting up CodeReady Workspace

## Prerequisites in my Environment:

- Windows Server 2019 with Hyper-V role installed, virtual switch with the name **crc**
- Local user account **openshift**, member of group **Administrators**
- By default the resources for the CodeReady Workspace will be installed in the user profile directory – 60 to 100 GB free space required
- Executing **crc.exe setup** creates virtual machine **crc** in Hyper-V with 4 Core and 9 GB RAM
  - To improve performance I have configured the VM to 8 Core and 20 GB RAM



# First Steps


## Prepare the environment




- Access open shift console using chrome browser <https://console-openshift-console.apps-crc.testing>
- Logon with Administrator role
- Creation of the project **serverless-demo**
- Assign permissions to developer user
- Installation of Operator Red Hat OpenShift Serverless
  - All Namespaces

### Installed Operators

Installed Operators are represented by ClusterServiceVersions within this Namespace. For more information, see the [Understanding Operators documentation](#). Or create an Operator and ClusterServiceVersion using the [Operator SDK](#).

Name ▾

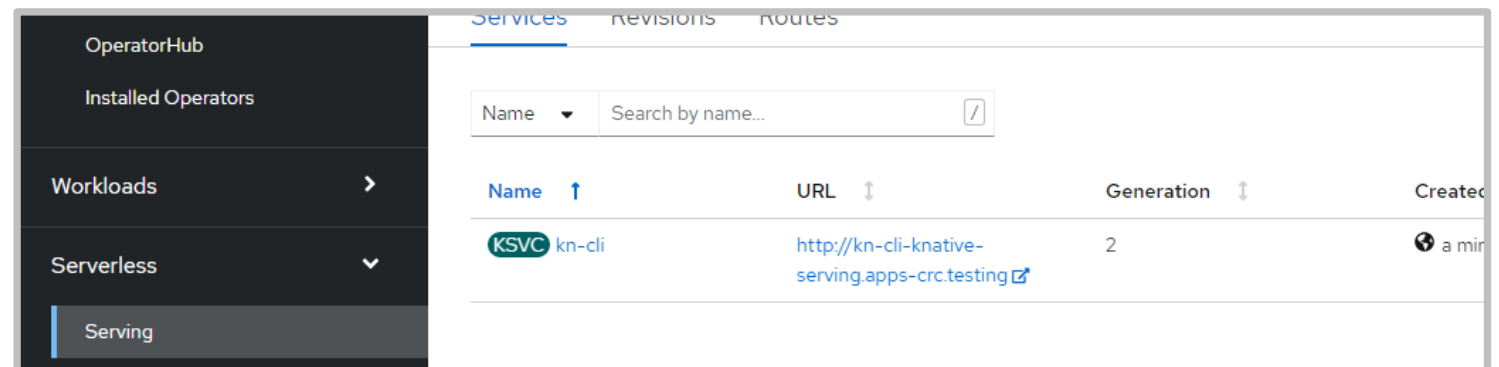
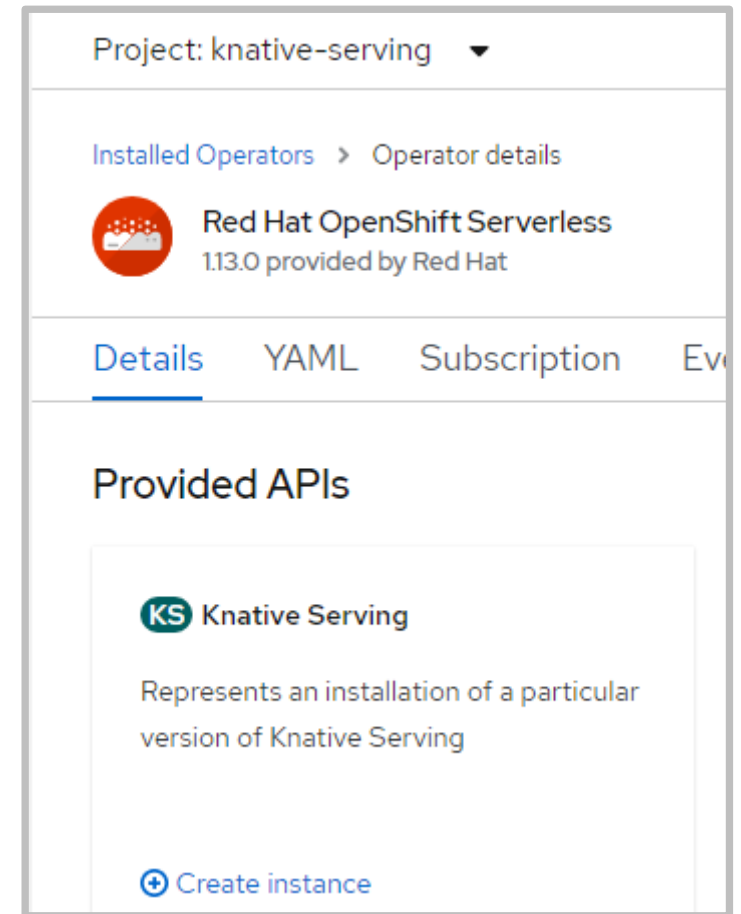
Search by name... 

Name ↑	Managed Namespaces	Status	Provided APIs
 <div><b>Red Hat OpenShift Serverless</b> 1.13.0 provided by Red Hat</div>	All Namespaces	 Succeeded Up to date	<a href="#">Knative Serving</a> <a href="#">Knative Eventing</a> <a href="#">Knative Kafka</a> 

# First Steps

## Prepare the environment

- Next Knative Serving must be created
- Change to project knative-serving
- Under Knative Serving click **Create instance**
- The new instance can be found under Serverless Serving



# Prepare the Application Source

- Access open shift console using chrome browser <https://console-openshift-console.apps-crc.testing>
- Use developer role, ensure that the project **serverless-demo** is selected
- Navigate to **Topology** and click **Samples**
- Select Nginx
- Note the URL of the Sample repository
- Create a copy to your Git account
- → this will be helpful if want to do changes later

## NGINX Nginx HTTP server and a reverse proxy 1.18 (UBI 7)

BUILDER NGINX

Build and serve static content via Nginx HTTP server and a reverse proxy (nginx) on RHEL 7. For more information on this builder image, including OpenShift considerations, see <https://github.com/sclorg/nginx-container/blob/master/1.18/README.md>.

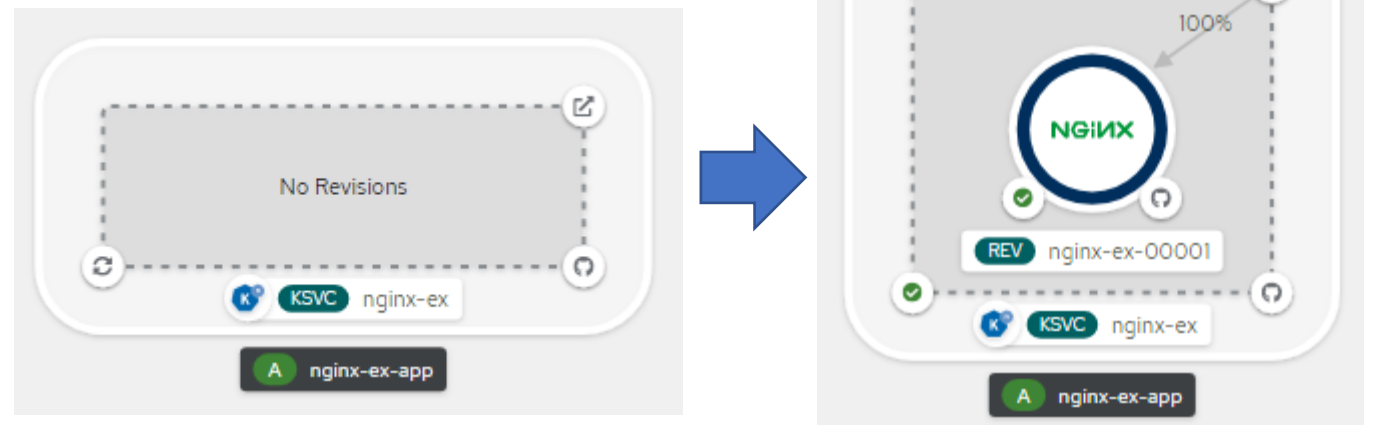
Sample repository: <https://github.com/sclorg/nginx-ex.git>

Git repo URL

<https://github.com/sclorg/nginx-ex.git>

# Create the serverless Application

- Access open shift console using chrome browser <https://console-openshift-console.apps-crc.testing>
- Click **Add** and Select **From Git**
- Enter your Git URL and select Nginx als Builder Image
- Application name: **nginx-ex-app** as per default
- Name: **nginx-ex**
- Select **Knative Service** as resource type
- Check **Create Route to the Application**
- Click **Create**
- The provisioning starts and finishes after short time



# Test and monitor the Application

## Configured Resources

- Route and build have been created
- Currently no pods are running



**KSVC nginx-ex** Actions

Details Resources

**Pods**  
All Revisions are autoscaled to 0

**Revisions** [Set traffic distribution](#)

**REV** nginx-ex-00001 100%  
**D** nginx-ex-00001-deployment

**Routes**

**RT** nginx-ex  
Location: <http://nginx-ex-serverless-demo.apps-crc.testing>

**Builds**

**BC** nginx-ex [Start Build](#)

✓ Build #1 is complete (7 minutes ago) [View logs](#)

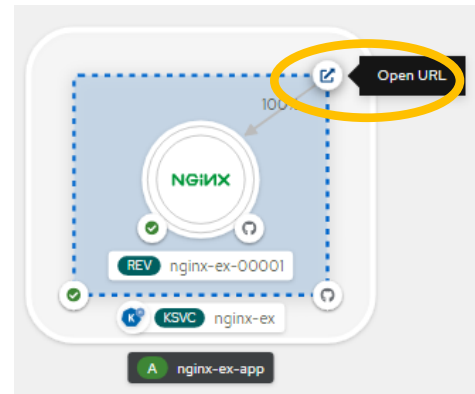
Build Configs		
<a href="#">Filter</a>	Name	Search by name...
Name	Labels	Created
<b>BC</b> nginx-ex	<code>app=nginx-ex</code> <code>app.kubernetes.io/component=nginx-ex</code> <code>app.kubernetes.io/instance=nginx-ex</code> <code>app.kubernetes.io/name=nginx</code> <code>app.kubernetes.io/part-of=nginx-ex-app</code> <code>app.openshift.io/runtime=nginx</code> <code>app.openshift.io/runtime-version=1.18-ubi7</code>	Mar 21, 1:57 pm



# Test and monitor the application

## Accessing the Application

- After Clicking **Open URL** the pod is scheduled and started immediately
- The website is reachable
- The application scales down to 0 after a short time of inactivity



REV nginx-ex-00001 Actions

Details Resources

**Pods**

P nginx-ex-00001-deployment-cdd9c97-rfqvf	Running	View logs
---	---------	-----------

**Deployment**

D nginx-ex-00001-deployment
-----------------------------

### Welcome to my first nginx application on OpenShift

#### Deploying code changes

To change this application, update your code and rebuild/redeploy the image.

For more information and help

- [Developer Center](#)
- [Documentation](#)
- [OpenShift forums](#)
- [Stack Overflow questions for OpenShift](#)
- [IRC channel at #openshift on freenode](#)

# Test and monitor the application

view from the command line

While the application is running

**oc get po** shows the running pod and the completed pod from the build process

```
>oc get po
```

NAME	READY	STATUS	RESTARTS	AGE
nginx-ex-00001-deployment-cdd9c97-7wmtb	<b>2/2</b>	<b>Running</b>	0	40s
nginx-ex-1-build	0/1	Completed	0	59m

Once the application is idle the pods are scaled down, reach status terminating and are finally unscheduled.

```
>oc get po
```


NAME	READY	STATUS	RESTARTS	AGE
nginx-ex-00001-deployment-cdd9c97-7wmtb	<b>0/2</b>	<b>Terminating</b>	0	6m45s
nginx-ex-1-build	0/1	Completed	0	65m

```
>oc get po
```

NAME	READY	STATUS	RESTARTS	AGE
nginx-ex-1-build	0/1	Completed	0	65m

# Summery

- OpenShift Serverless is a great way to create applications as expected
- Scaling down to 0 helps saving resources and money und scaling up ensures the performance if it is required.
- It easy to handle, even with little experience in OpenShift.
- Using CodeReady workspace a good way to provide test sc in case a cloud based test environment is not feasible. However, CodeReady workspace needs some CPU, Memory and disk resources.



Thank you  
for the  
phantastic  
training!