Creating and running an OpenShift® Container Platform test environment on Windows 10® using Minishift

Contents

Prerequisites:

- Windows 10
- Local Administrator privileges
- CPU Virtualization features enabled (in BIOS)
- Microsoft Hyper-V or Oracle VirtualBox installed
- OpenShift console Login Credentials (provided in this Doc)

Installing VirtualBox

VirtualBox is approved for use by EDN Software Standards, and can be installed on your EDN laptop from the EDN managed Software repository:

http://ednops.vzwnet.com/computers/software.aspx

What is OpenShift?

Red Hat OpenShift is an open source container application platform based on the Kubernetes container orchestrator for enterprise application development and deployment. OpenShift® is a container management platform developed by Red Hat® that leverages container technologies such as Docker and Kubernetes. OpenShift provides valuable DevOps features to both developers and administrators. For Developers, OpenShift is a solid infrastructure to build and deploy apps on, For Administrators OpenShift features provide tools that simplify the administration of the underlying **Docker** and **Kubernetes** cluster infrastructure.

OpenShift does not replace Docker or Kubernetes but instead, provides a Web GIU and enhanced CLI from which you can manage and control the Kubernetes clusters and the containers deployed on it, which are running the applications that developers are building, allowing for CI/CD.

Kubernetes resources can grow or shrink as needed and can be divided into distinct logical resource domains or Realms. Realms can assigned to different engineering groups, for example, you can create individual Realms for Development, Test/QA, Staging, and Production. Each realm being (logically) isolated from the others, while running on the same physical compute resources, eliminating the potential for interference with applications running in the Production environment.

The OpenShift cluster can spawn multiple machines and once set up, it is a solid development platform that allows:

- Cluster management
- Cluster scaling
- Application scaling
- Application monitoring
- Application self-healing
- Etc…

Minishift

If you want to try OpenShift but don't want to set up a real, full-blown cluster, Minishift is the solution for you.

Minishift is a great way to test the OpenShift capabilities without committing hardware to it. By using Minishift you will set up a virtual machine on your hardware. This OpenShift VM will be your very own "one-machine cluster" and when you're done you can simply throw it away (delete it).

Minishift setup

Download and install Minishift from the following URL:

https://github.com/minishift/minishift/releases

Follow the instructions for your OS on the *docs.okd.io* website to install the Minishift environment. As of the writing of this doc, the latest Minishift version is: **1.30.0**

https://docs.okd.io/latest/minishift/getting-started/index.html

Starting Minishift

This assumes that you have already installed a virtual environment on your laptop consisting of either VirtualBox, OR Hyper-V. Once you have this environment installed then you are ready to startup your Minishift

Open a PowerShell console with elevated Administrator rights (Right click on the PowerShell Icon and select Run as Administrator)

- At the PowerShell prompt type the following command:

```
PS C:\WINDOWS\system32> minishift start
```

It takes a few minutes for Minishift to completely load and is up and running. In the startup messages[1], you will see a line of text at the end that says:

The server is accessible via web console at: https://192.168.99.100:8443/console

You can access the OpenShift Web GIU by copying and pasting this URL into your browser. Alternatively, you start the web console from the command line a shown below:

```
PS C:\WINDOWS\system32> minishift console
```

The OpenShift console GUI

Once Minishift it is fully up and running, you will be provided with the URL (similar to the URL example above) that you will use to access the OCP web GUI. The URL consists of a Private non-routable Class C IP address.

This will open the OCP Web UI in the default browser on the machine where you installed minishift. *Don't worry about the insecure connection notice, that's just because the web server certificate is self-signed.*

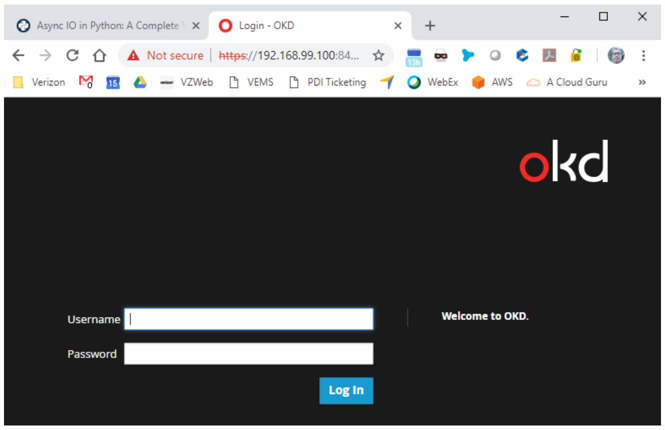


Figure 1 OpenShift web console

[1] See Appendix A for full startup output.

Logging in to the Web Console

Now that you have the web console open, you will be required to login to the console using the credentials found below. There are two user accounts *developer* and *system* (Administrator). The Developer account is sufficient for performing all of the functions necessary to play around with the OpenShift cluster instance provided by Minishift.

To login as the **developer** user, enter the credentials as follows:

> Username: `developer`
> Password: *<any value but do not leave blank>*

To login as the **system** user, enter the following credentials:

> Username: `system`
> Password: `admin`

Once you are logged in you will see the catalog view shown in figure 2.

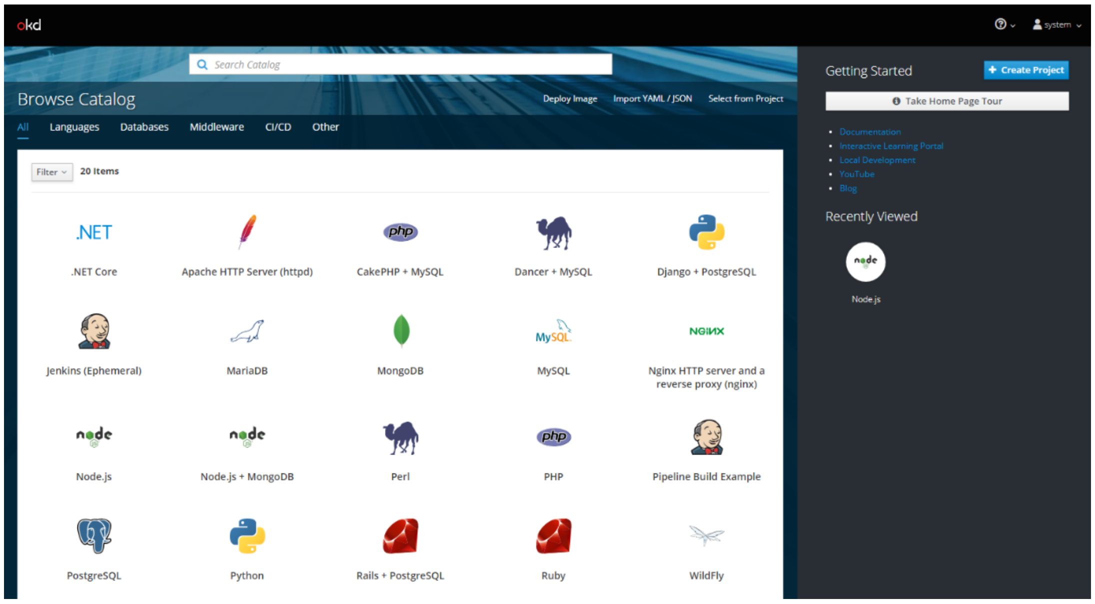


Figure 2 service catalog

From here you can select from the menu of pre-defined applications that have been created for you. For instance you may have a service you are developing that needs a Webserver or a cluster of webservers, an interpreter such as Node js, and a database. In this case you would deploy an instance of Apache HTTP Server or maybe Nginx, a Node js Instance and an instance of MySQL. But for now you'll want to start with a project which will bring you back to this this view.

First things first… working with Projects

To start, you first want to create a Project by selecting the **"+ Create Project**" button. In this case the **developer** account has a project already created called "**My Project**". Selecting "My Project" will take you to the "**Project Overview**" page where you'll see several buttons in the center of the page: **Browse Catalog**, **Deploy Image**, **Import YAML/JSON**, and **Select from Project**, and on the left side of the page there are several other views to select starting with the Overview button, Applications, Builds, Resources, Storage, Monitoring, and Catalog.

For this document, the focus will be on the "**Browse Catalog**" button in the Overview page which will take us to the catalog view with the collection of pre-defined applications. Each of these applications are containerized applications built on the required OS and with all of the necessary dependencies installed. You can read through the documentation to learn more about creating catalog items.

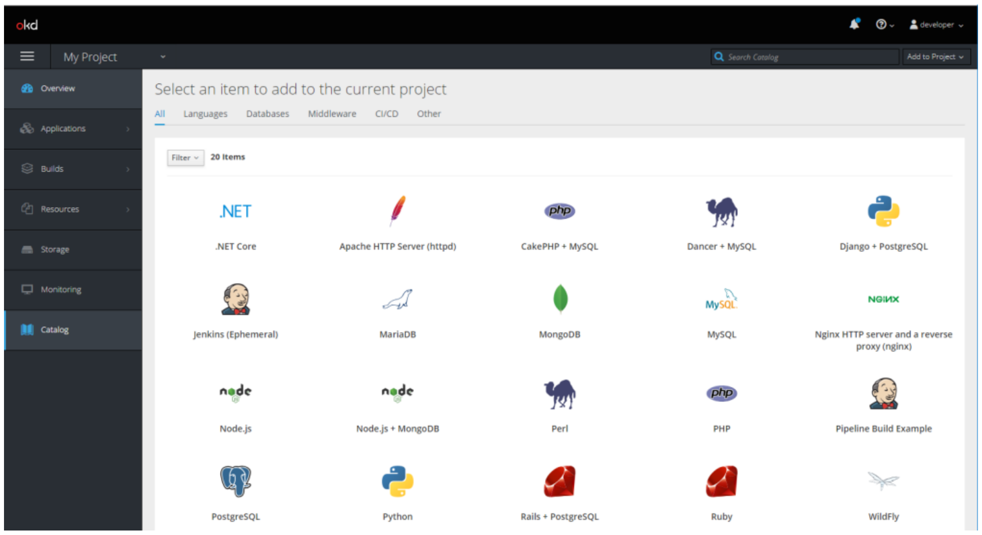
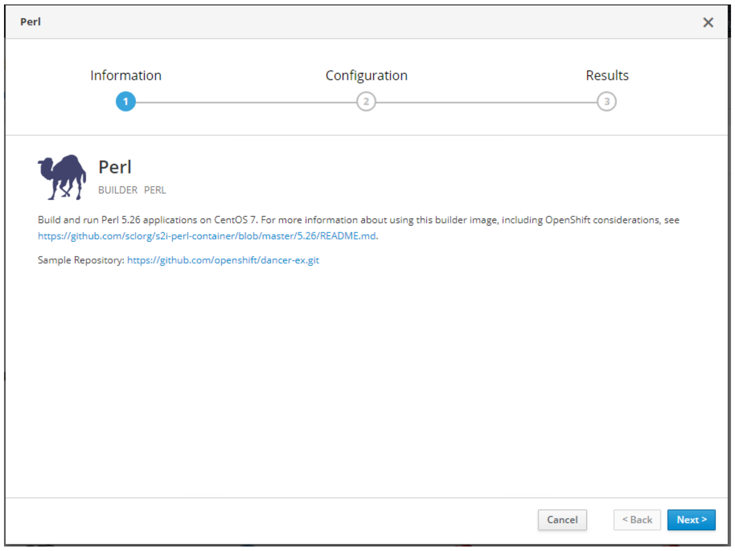


Figure 3 - Project view

Deploying an Application

To deploy an instance of an application, from the Catalog view, click on the application you want to deploy. This will open a configuration wizard similar to the image in Figure 4, and will walk you through the process. The configuration step will include specific fields that you will need to fill out. Then you will be taken to the Results step where you can verify the information you provided and either go back to configuration and correct it, or if the configuration information is correct you click on the create button to complete the wizard and create your application.



Build and run Perl 5.26 applications on CentOS 7. For more information about using this builder image, including OpenShift considerations, see https://github.com/sclorg/s2i-perl-container/blob/master/5.26/README.md.

Sample Repository: https://github.com/openshift/dancer-ex.git

Figure 4 configuration wizard

The first view of the configuration wizard will be the **Information** view which provides a brief description of the application you about to deploy and may also provide links to additional documentation or a repository from which the service will get downloaded from. Refer to *Figure 4.*

The next view is the Configuration view. This view will be different for each application depending on the information needed to deploy it such as the Version that you want to deploy (if there are more than one), the Application Name is the name that you want to give this instance of the application, a Git Repository, this is the URL for the Git or Bitbucket repository where the application is stored. You'll want to use the "Sample Repository" for these applications but in a real production scenario you will use the URL for your Git or Bitbucket repository.

Once you have populated all of the applications configuration fields the "**create**" button will become active, clicking on this button will deploy your new instance of the application.

Using the menu on the left side of the console you can then navigate back to the Project "**Overview**" view where you will see your newly deployed application listed there showing the URL you'll use to access this instance as well as to see the deployment configuration and how many instances of the application you have defined.

One you have created your first project, each time you run minishift and open the web console you will be brought to the Catalog page by default and any projects you have created will be shown on the right hand side as well as any catalog objects you have recently looked at. To get back to your project, you click on the project name shown circled in red on the right side panel.

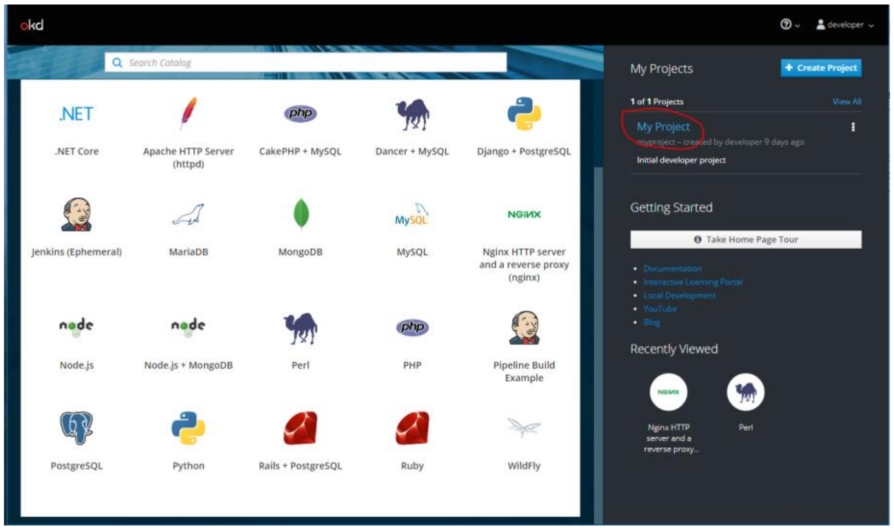


Figure 5 Web console start page

Clicking on your project will open the Project Overview page as seen in figure 6 below.

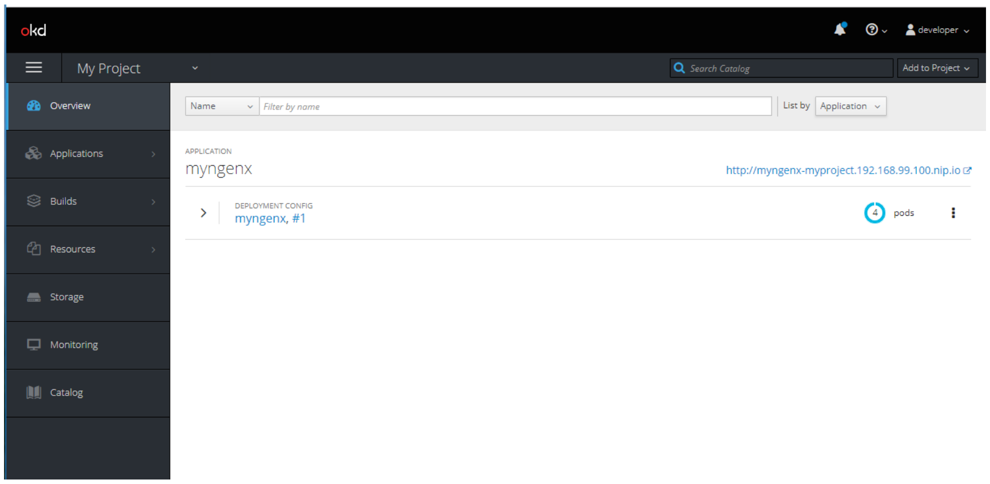


Figure 6 Project overview page

This page shows any Pods you have created. When you look at this same information from the command line using the Openshift cli 'oc' or Kubernetes 'kubectl', the "Container" name is the application name you gave when creating an application instance, In this case “myngenx”. In Figure 6 you can see I have created 4 pods, each is running an instance of the Ngenx web server and each can be accessed by IP, or FQDN.

In the upper right of the screen in figure 6 you can also see the global URL or that you would use to access the ngenx web server application. All of this is being managed by **Kubernetes** which provides the orchestration mechanism to manage the deployment of the **Docker** containers. When you access the *myngenx* application URL, it is Kubernetes that is transparently providing all of the network traffic load balancing to each of the four ngenx web servers, and because I have specified (require) 4 instances of my application, Kubernetes will monitor the status and health of the pods

If any one of the four pods should crash or exceed the specified CPU or Memory limits that were specified, Kubernetes will take and action such as killing the misbehaving pod and starting another. In the case of a pod crashing, Kubernetes will restart the crashed instance and a counter will show how many time the pod was restarted as well as how long the pod has been up since it was created or restarted.

Interacting with Minishift

When you use Minishift, you interact with the following components:

- the Minishift virtual machine (VM)
- the Docker daemon running on the VM
- the OpenShift cluster running on the Docker daemon

Docker and Kubernetes

When you installed minishift, Docker is included as part of the minishift CentOS Linux image on which all of the Docker containers will run.

Accessing the OpenShift/Kubernetes CLI

As a part of the minishift install, the latest version of the OpenShift Container Platform CLI “oc” was installed onto your laptop. Since OpenShift is built on Kubernetes, this also includes the Kubernetes CLI “kubectl”. Both CLIs can be used from the PowerShell console.

To run OpenShift CLI commands, at the PowerShell prompt you type: oc followed by any command options.

The same is true for the Kubernetes CLI. Type: kubectl followed by any command options

(*Refer to Appendix B for links to more Docker information and resources.*)

Accessing the Docker CLI

If you want to access the Docker CLI, you can’t access it from the PowerShell console, instead you would need to first stop minishift from the PowerShell command line (if it is already running), then start the minishift VM from within either the Hyper-V or VirtualBox management GUI.
Once the minishift VM is running, you access the CentOS command console which will require you to login to CentOS Linux OS using the following credentials.

User name: `root`
Password: `centos`

From the minishift Linux VM console, you can run Docker CLI commands and learn about the Docker CLI and working with Docker. (*Refer to Appendix B for links to more Docker information and resources.*)

Appendix A

```
  PS C:\WINDOWS\system32> minishift start
-- Starting profile 'minishift'
-- Check if deprecated options are used ... OK
-- Checking if https://github.com is reachable ... OK
-- Checking if requested OpenShift version 'v3.11.0' is valid ... OK
-- Checking if requested OpenShift version 'v3.11.0' is supported ... OK
-- Checking if requested hypervisor 'virtualbox' is supported on this platform ... OK
-- Checking if VirtualBox is installed ... OK
-- Checking the ISO URL ... OK
-- Checking if provided oc flags are supported ... OK
-- Starting the OpenShift cluster using 'virtualbox' hypervisor ...
-- Starting Minishift VM .............................. OK
-- Checking for IP address ... OK
-- Checking for nameservers ... OK
-- Checking if external host is reachable from the Minishift VM ...
   Pinging 8.8.8.8 ... OK
-- Checking HTTP connectivity from the VM ...
   Retrieving http://minishift.io/index.html ... OK
-- Checking if persistent storage volume is mounted ... OK
-- Checking available disk space ... 37% used OK
-- Writing current configuration for static assignment of IP address ... OK
-- OpenShift cluster will be configured with ...
   Version: v3.11.0
-- Copying oc binary from the OpenShift container image to VM ... OK
-- Starting OpenShift cluster .....................................
Getting a Docker client ...
Checking if image openshift/origin-control-plane:v3.11.0 is available ...
Checking type of volume mount ...
Determining server IP ...
Using public hostname IP 192.168.99.100 as the host IP
Checking if OpenShift is already running ...
Checking for supported Docker version (=>1.22) ...
Checking if insecured registry is configured properly in Docker ...
Checking if required ports are available ...
Checking if OpenShift client is configured properly ...
Checking if image openshift/origin-control-plane:v3.11.0 is available ...
Starting OpenShift using openshift/origin-control-plane:v3.11.0 ...
I0116 16:20:22.220755    2407 flags.go:30] Running "create-kubelet-flags"
I0116 16:20:22.742956    2407 run_kubelet.go:49] Running "start-kubelet"
I0116 16:20:23.010309    2407 run_self_hosted.go:181] Waiting for the kube-apiserver to be ready ...
I0116 16:20:44.068425    2407 interface.go:26] Installing "kube-proxy" ...
I0116 16:20:44.068451    2407 interface.go:26] Installing "kube-dns" ...
I0116 16:20:44.068461    2407 interface.go:26] Installing "openshift-service-cert-signer-operator" ...
I0116 16:20:44.068474    2407 interface.go:26] Installing "openshift-apiserver" ...
I0116 16:20:44.068532    2407 apply_template.go:81] Installing "openshift-apiserver"
I0116 16:20:44.070324    2407 apply_template.go:81] Installing "kube-proxy"
I0116 16:20:44.072563    2407 apply_template.go:81] Installing "kube-dns"
I0116 16:20:44.072898    2407 apply_template.go:81] Installing "openshift-service-cert-signer-operator"
I0116 16:21:25.530659    2407 interface.go:41] Finished installing "kube-proxy" "kube-dns" "openshift-service-cert-signer-operator" "openshift-
apiserver"
I0116 16:22:06.630861    2407 run_self_hosted.go:242] openshift-apiserver available
I0116 16:22:06.631485    2407 interface.go:26] Installing "openshift-controller-manager" ...
I0116 16:22:06.631513    2407 apply_template.go:81] Installing "openshift-controller-manager"
I0116 16:22:10.665277    2407 interface.go:41] Finished installing "openshift-controller-manager"
Adding default OAuthClient redirect URIs ...
Adding centos-imagestreams ...
Adding sample-templates ...
Adding persistent-volumes ...
Adding registry ...
Adding router ...
Adding web-console ...
I0116 16:22:10.770139    2407 interface.go:26] Installing "centos-imagestreams" ...
I0116 16:22:10.770150    2407 interface.go:26] Installing "sample-templates" ...
I0116 16:22:10.770155    2407 interface.go:26] Installing "persistent-volumes" ...
I0116 16:22:10.770163    2407 interface.go:26] Installing "openshift-image-registry" ...
I0116 16:22:10.770167    2407 interface.go:26] Installing "openshift-router" ...
I0116 16:22:10.770172    2407 interface.go:26] Installing "openshift-web-console-operator" ...
I0116 16:22:10.770456    2407 apply_template.go:81] Installing "openshift-web-console-operator"
I0116 16:22:10.770628    2407 apply_list.go:67] Installing "centos-imagestreams"
I0116 16:22:10.770868    2407 interface.go:26] Installing "sample-templates/mongodb" ...
I0116 16:22:10.770878    2407 interface.go:26] Installing "sample-templates/mysql" ...
I0116 16:22:10.770883    2407 interface.go:26] Installing "sample-templates/postgresql" ...
I0116 16:22:10.770888    2407 interface.go:26] Installing "sample-templates/dancer quickstart" ...
I0116 16:22:10.770892    2407 interface.go:26] Installing "sample-templates/django quickstart" ...
I0116 16:22:10.770897    2407 interface.go:26] Installing "sample-templates/sample pipeline" ...
I0116 16:22:10.770902    2407 interface.go:26] Installing "sample-templates/mariadb" ...
I0116 16:22:10.770908    2407 interface.go:26] Installing "sample-templates/cakephp quickstart" ...
I0116 16:22:10.770913    2407 interface.go:26] Installing "sample-templates/nodejs quickstart" ...
I0116 16:22:10.770918    2407 interface.go:26] Installing "sample-templates/rails quickstart" ...
I0116 16:22:10.770923    2407 interface.go:26] Installing "sample-templates/jenkins pipeline ephemeral" ...
I0116 16:22:10.770977    2407 apply_list.go:67] Installing "sample-templates/jenkins pipeline ephemeral"
I0116 16:22:10.773092    2407 apply_list.go:67] Installing "sample-templates/mongodb"
I0116 16:22:10.773369    2407 apply_list.go:67] Installing "sample-templates/mysql"
I0116 16:22:10.773541    2407 apply_list.go:67] Installing "sample-templates/postgresql"
I0116 16:22:10.773702    2407 apply_list.go:67] Installing "sample-templates/dancer quickstart"
I0116 16:22:10.773957    2407 apply_list.go:67] Installing "sample-templates/django quickstart"
I0116 16:22:10.774117    2407 apply_list.go:67] Installing "sample-templates/sample pipeline"
I0116 16:22:10.774234    2407 apply_list.go:67] Installing "sample-templates/mariadb"
I0116 16:22:10.774339    2407 apply_list.go:67] Installing "sample-templates/cakephp quickstart"
I0116 16:22:10.774443    2407 apply_list.go:67] Installing "sample-templates/nodejs quickstart"
I0116 16:22:10.774538    2407 apply_list.go:67] Installing "sample-templates/rails quickstart"
I0116 16:22:26.050725    2407 interface.go:41] Finished installing "sample-templates/mongodb" "sample-templates/mysql" "sample-templates/postgresql"
"sample-templates/dancer quickstart" "sample-templates/django quickstart" "sample-templates/sample pipeline" "sample-templates/mariadb" "sample-
templates/cakephp quickstart" "sample-templates/nodejs quickstart" "sample-templates/rails quickstart" "sample-templates/jenkins pipeline ephemeral"
I0116 16:22:27.099338    2407 interface.go:41] Finished installing "centos-imagestreams" "sample-templates" "persistent-volumes" "openshift-image-
registry" "openshift-router" "openshift-web-console-operator"
Server Information ...
OpenShift server started.

The server is accessible via web console at:
    https://192.168.99.100:8443/console
```

Appendix B

Additional Resources

OpenShift Documentation as well as additional learning resources

https://docs.openshift.com/container-platform/3.11/welcome/index.html

Understanding the Openshift Architecture:

https://docs.openshift.com/container-platform/3.11/architecture/index.html#architecture-index

OpenShift CLI Overview

https://docs.openshift.com/container-platform/3.11/cli_reference/index.html#cli-reference-index

Kubernetes Web site

https://kubernetes.io/

Docker – What is a container?

https://www.docker.com/resources/what-container

Minishift – Documentation

https://docs.okd.io/latest/minishift/index.html

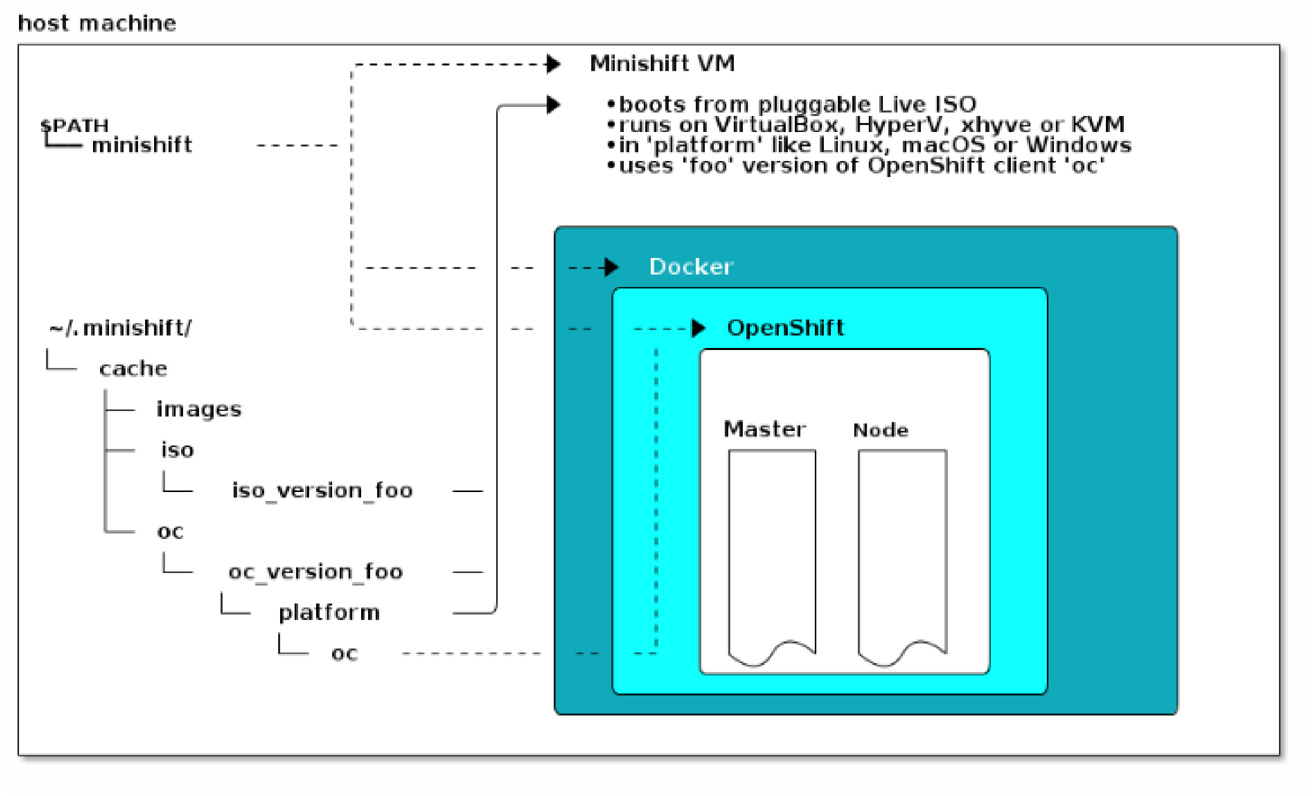

host machine
$PATH
minishift
Minishift VM
•boots from pluggable Live ISO
•runs on VirtualBox, HyperV, xhyve or KVM
•in 'platform' like Linux, macOS or Windows
•uses 'foo' version of OpenShift client 'oc'
Docker
OpenShift
Master
Node
~/.minishift/
cache
images
iso
iso_version_foo
oc
oc_version_foo
platform
oc

Figure 7 - Minishift Architecture