Zero to Cloud-Native with IBM Cloud

Part 4: OpenShift Provisioning and Configuration

Kevin Collins
kevincollins@us.ibm.com
Technical Sales Leader
IBM Cloud Enterprise Containers – Americas

Kunal Malhotra
kunal.malhotra3@ibm.com
Cloud Platform Engineer
IBM Cloud MEA

1 IBM Cloud Managed OpenShift

1 -1 Introduction

In this tutorial, we will use IBM Cloud's Managed OpenShift service to run our cloud-native application. Red Hat OpenShift on IBM Cloud is a managed offering to create your own OpenShift cluster of compute hosts to deploy and manage containerized apps on IBM Cloud. Red Hat OpenShift on IBM Cloud provides intelligent scheduling, self-healing, horizontal scaling, service discovery and load balancing, automated rollouts and rollbacks, and secret and configuration management for your apps. Combined with an intuitive user experience, built-in security and isolation, and advanced tools to secure, manage, and monitor your cluster workloads, you can rapidly deliver highly available and secure containerized apps in the public cloud.

You can read more about RedHat OpenShift on IBM Cloud here:
https://cloud.ibm.com/docs/openshift?topic=openshift-roks-overview

We will be creating a multizone cluster in this tutorial. If you prefer a single zone cluster then the same tutorial steps will work.

Why do we need a multizone cluster? In a multizone cluster, the worker nodes in your worker pools are replicated across multiple zones within one region. Multizone clusters are designed to evenly schedule pods across worker nodes and zones to assure availability and failure recovery. If worker nodes are not spread evenly across the zones or capacity is insufficient in one of the zones, the Kubernetes scheduler or OpenShift controller might fail to schedule all requested pods.

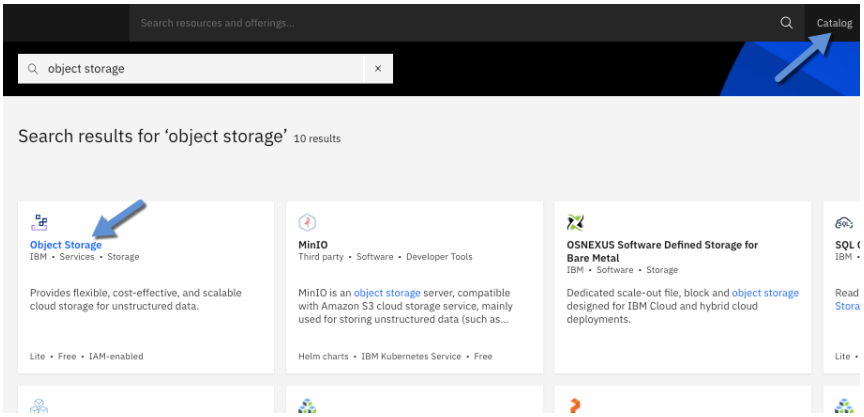
For this tutorial, from an availability perspective, a multizone cluster is not really needed. I've decided to use a multizone cluster of this tutorial to show the value and ease of using a multizone cluster that you would typically see in a production environment.

1 – 2 Preparing for Your Cluster

Before provisioning your cluster, you need to consider which network infrastructure you want to use from a Virtual Private Cloud to Classic Infrastructure. As we went through in Part 2 – we will be deploying our cluster in the Virtual Private Cloud we already created to give our application an additional layer of security from a networking perspective.

1 – 3 Cloud Object Storage

For a cluster provisioned on a virtual private cloud, the internal image registry is backed by IBM Cloud Object Storage. If you don't already have a Cloud Object Storage instance that would like to use, you will need to create one. To do so, click on the IBM Cloud Catalog, search for Object Storage and click the IBM Cloud Object Storage tile.



You can read more about Cloud Object Storage here: https://cloud.ibm.com/docs/cloud-object-storage?topic=cloud-object-storage-getting-started-cloud-object-storage

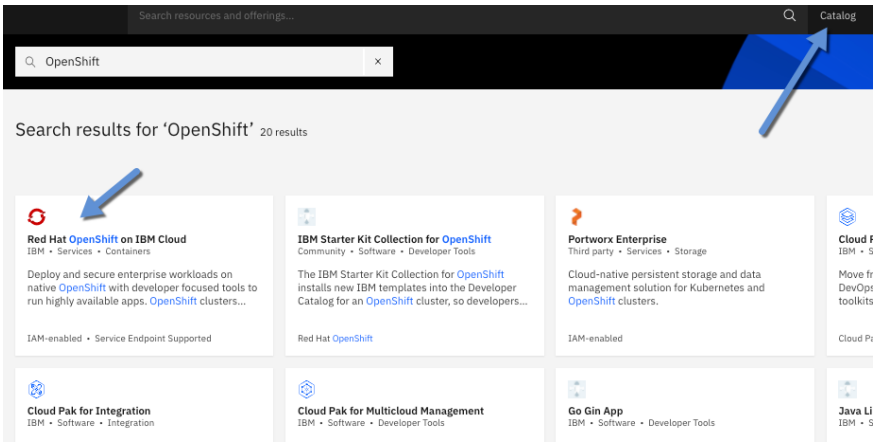
As we have done with all the other IBM Cloud services we have created, you will need to give the Cloud Object Storage instance a name and specify the resource group it will be in.

For this tutorial I will use the following settings:
Plan: Standard
Service Name: Cloud Object Storage-zero-to-cloud-native
Resource Group: zero-to-cloud-native

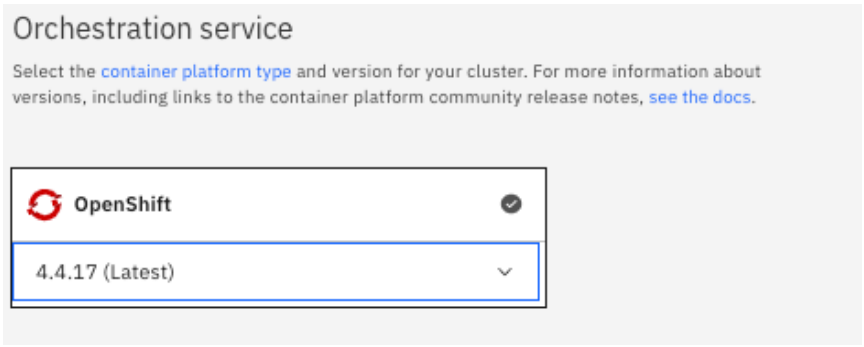
After entering your settings for Cloud Object Storage, click Create.

1 – 3 Provisioning RedHat OpenShift on IBM Cloud

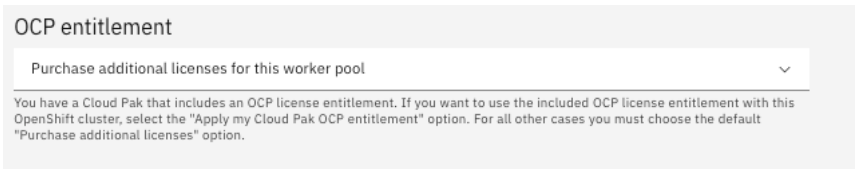
To provision an OpenShift cluster, click on the IBM Cloud Catalog, search for OpenShift and then click on the Red Hat OpenShift on IBM Cloud tile.



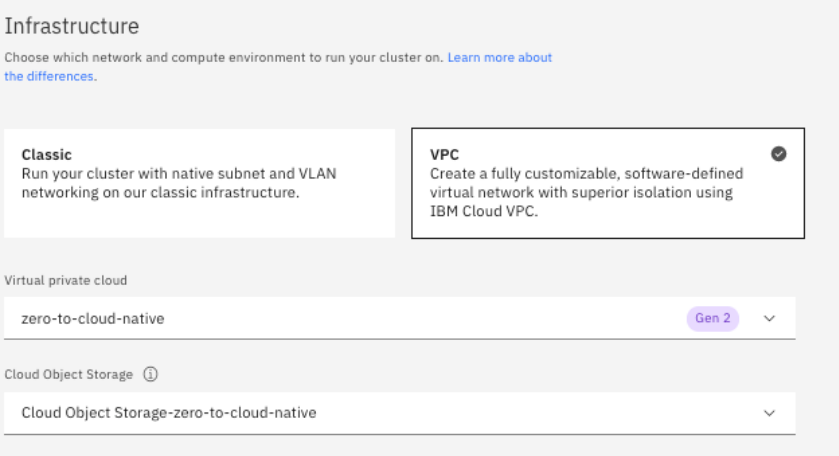
The first thing we need to do is to specify a version for our OpenShift Cluster. I'm going to select the latest supported version in IBM Cloud which is version 4.4.



Next, you need to provide an OCP entitlement for the OCP license. If you had bought a CloudPak from IBM, I could apply that OCP license to this cluster. As CloudPaks are out of scope for this tutorial, I will leave the default value Purchase additional licenses for this worker pool.



We then need to specify the infrastructure to use with our cluster. As we have been through, we will be deploying our cluster on a virtual private cloud that we already created. I will select the zero-to-cloud-native VPC I created in part 3 of this tutorial and will also select the instance of Cloud Object Storage for the internal registry for my cluster. The service will automatically create a bucket for your cluster's internal registry using the Cloud Object Storage instance you select.



Next, we specify the location where we want to deploy our cluster to. For the resource group name, I will select the resource group we have been using zero-to-cloud-native. For the datacenters, I will keep all three data centers that I created in my VPC selected. Distributing your workload across three zones ensures high availability for your app in case one or two zones are not available. By having your worker nodes spread evenly across all three availability zones gives you an industry leading 99.99% SLA for your OpenShift Cluster.

Next, I setup my default worker pool. A worker pool is a grouping of worker nodes that are all the same flavor or t-shirt size for workers in that pool. Worker nodes pools allow us to do things like autoscaling. When the deployed worker nodes reach capacity and we can't schedule any more workload to the nodes, additional worker nodes of the same flavor can be added automatically with autoscaling. This is fully customizable with minimum and maximums.

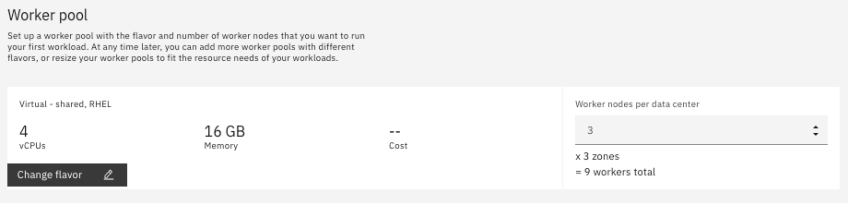
By default, we deploy with 4 x 16 worker nodes which is 4 virtual cpus and 16 gigs of memory. For this tutorial that will be more than enough. I'm going to keep the worker node count as 3 per zone. This tutorial only requires 1 worker node, but in future tutorials I will show resiliency features of IBM Cloud Managed OpenShift by having multiple workers in multiple zones.

There are three choices of compute within our managed OpenShift service. The first is called shared compute. Shared compute has a single tenant virtual machine and a multi-tenant hypervisor and hardware underneath it. This is your typical 'cloud' virtual machine.

Then, we have dedicated compute. This means you have a single tenant virtual machine, hypervisor and hardware. In this case, you are the only tenant running on a physical server running in IBM Cloud. You may have multiple virtual machines running on that server, but you don't have to worry about your competitor or noisy neighbors because you own that full box.

The last option is bare metal. If you need to guarantee a certain amount of resources for your application, perhaps you need GPUs, compute isolation, or eliminating the hypervisor you can get a bare metal server as part of IBM's managed service for OpenShift.

For this tutorial, I'm going to go with Shared Virtual.



Finally, I need to give my cluster a name. I'm going to name my cluster zero-to-cloud-native and then click Create.

This will start provisioning your cluster and you can start using it in about 15 minutes. Next up, we will setup the cloud databases we will be using and our messaging platform.