OpenShift on Power

—

Christy Norman (she/her)
Lead: OpenShift on Power VS, container runtimes on Power

Contents

OpenShift Overview

How it isn't Kubernetes, and some commonly-used terminology.

Install Methods

Installation types, Power-specific install options, and a few best practices.
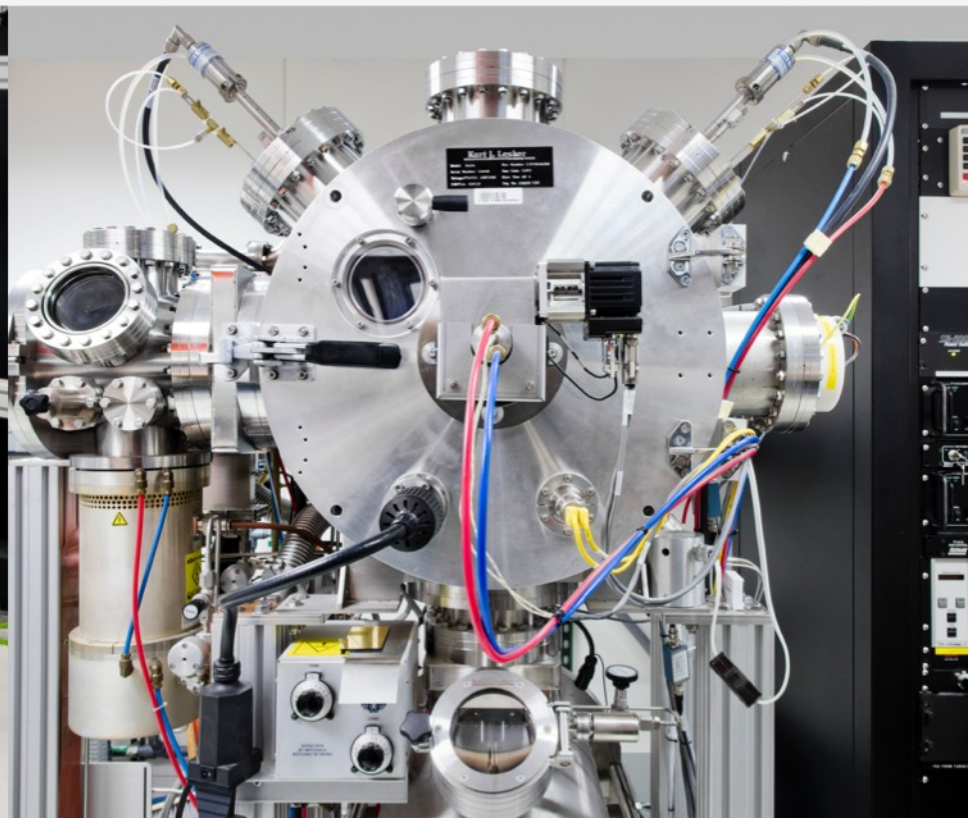
Operators

an overview and Power highlights.

Container Storage

An overview and Power highlights.

Administering

Post-installation tasks to consider for your cluster.

OpenShift Basics

OpenShift is not just "glorified Kubernetes"

OperatorHub → A baked-in portal from which to install software with a few clicks

S2I → Source to Image Capabilities allow you to streamline development to container image builds

OpenShift Pipelines → RedHat technology based on the Tekton project that allows you to create container-based CI/CD pipelines

Available via OperatorHub

OpenShift provides a platform from which to launch and manage a wealth of Open Source projects built, tested, and supported by Red Hat

OpenShift & Kubernetes Concepts

- **CoreOS/RHCOS –** minimal, container-focused OS; (see Machine Config Operator)

- **Ignition** – similar to PXE: provisioning tool for CoreOS

- **YAML** – (*YAML Ain't Markup Language)* used to describe kube resources

- **Resource** - an endpoint in the Kubernetes API that stores a collection of API objects of a certain kind; for example, the built-in pods resource contains a collection of Pod objects

- **Service** - an abstract way to expose an application running on a set of Pods as a network service

- **Operators** - Conceptually, Operators, and the operator *pattern,* take human operational knowledge and encode it into software to manage a set of applications.

- **etcd** – the database that contains the state of all resource objects

- **kubelet** – agent running on each cluster node

OpenShift on Power

- First release: 3.11

  - RHEL worker nodes supported

- Current Release: 4.12 (GA'd 1/17)

  - As of 4.x, Red Hat Core OS nodes only

Working with OpenShift
Binaries

- openshift-install – Installer

- oc – OpenShift Client

- kubectl (optional) – Wrapped by oc

- podman – Red Hat's docker replacement

- crictl – Useful for debug inside cluster nodes

} - Available from the Red Hat Downloads page

} - Pre-installed in RHCOS

Installation
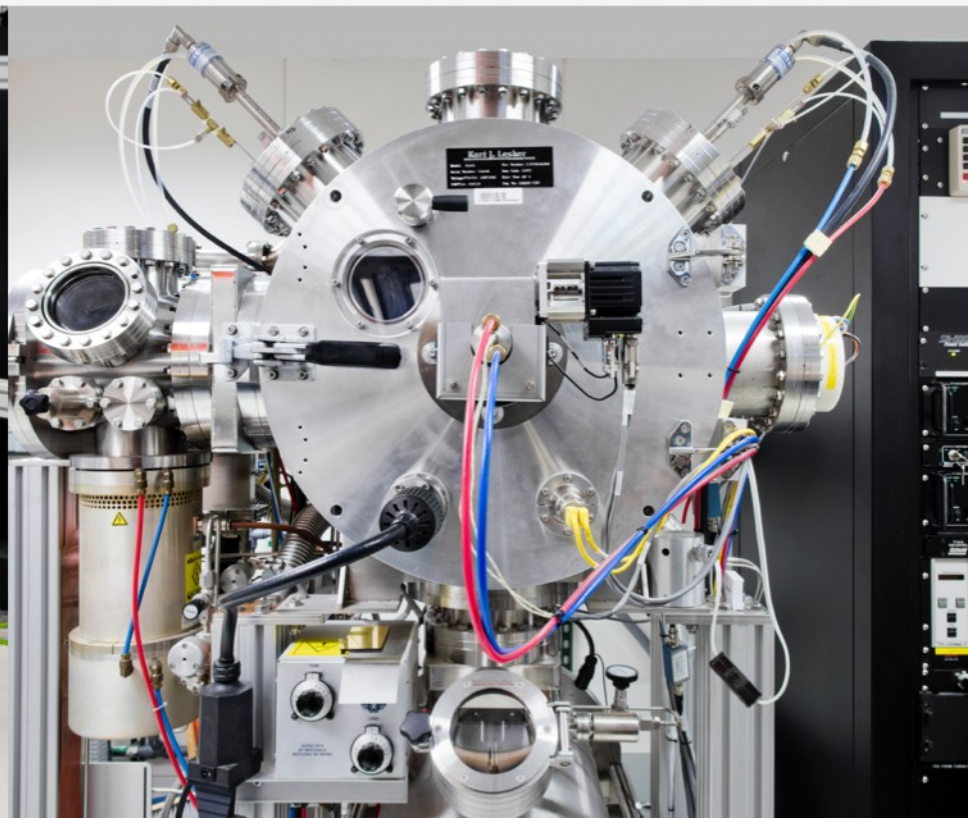
User-Provisioned vs Installer-Provisioned Installs

User-Provisioned

- Manually create all components

- More flexible

Installer-Provisioned

- Automation creates all* components

- Allows OCP to create & delete new nodes easily

- Auto-scaling of nodes

Installation Concepts

- **Helper Node** – aka "bastion"; host installations "helpers" like a web server, DHCP server, image registry, NFS server

- **Install Config** – file containing all values for the OpenShift installer

User-Provisioned Infrastructure

IBM-Provided Ansible playbooks:

https://github.com/ppc64le-cloud/ocp4-playbooks

- Bare Metal
- LPARs
- PowerVC
- Power VS

Installer-Provisioned Infrastructure

Coming in OpenShift 4.12*

- Power VS

Single Node OpenShift (SNO)

Development Only

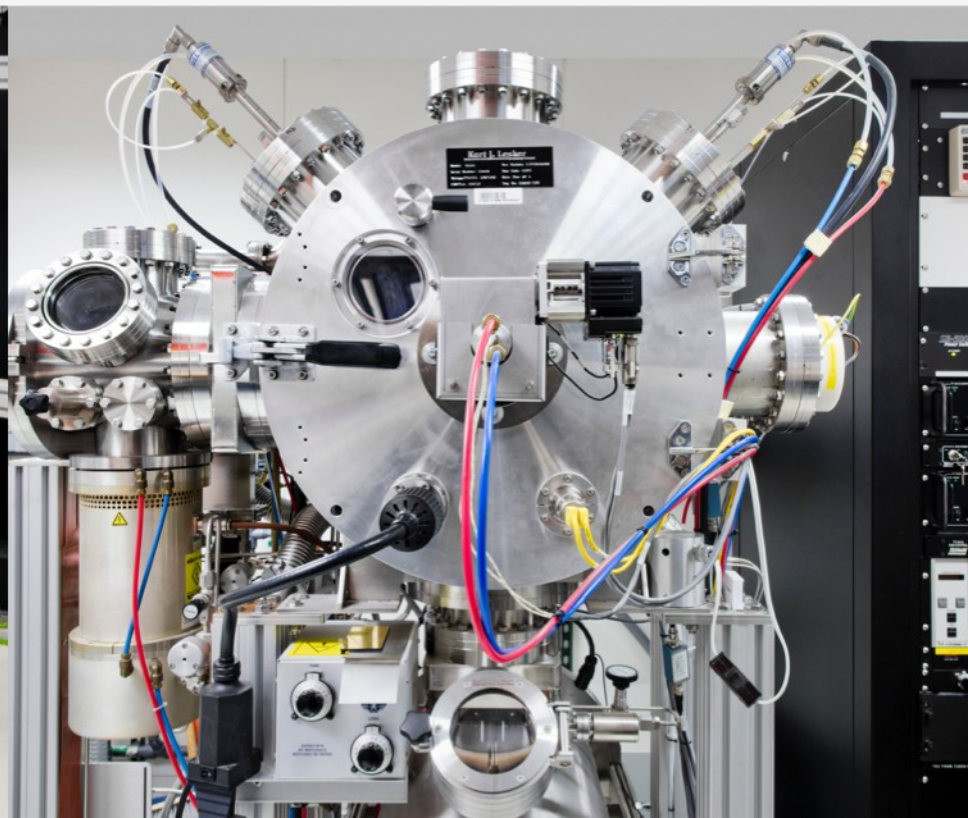- PowerVM Helper Node Playbook: https://github.com/cs-zhang/ocp4-upi-sno

Lessons from the Field

- Disk Performance & etcd – high-performance required

- MTU – 1450 recommended

  • You can have your DHCP server advertise this

- Large Send Offload – enabled in every layer of your hardware (don't forget the SEA!)

- Multipath Disks – enable pre-install

# Operators

Operator Concepts

What is an Operator?

- **Cluster Operators** -manage aspects of the OpenShift cluster, such as cluster networking, console, registry, storage, and the marketplace

- **Additional Operators** – operators installed post-cluster-install that provide additional features

- **Cluster Version Operator** – controls which cluster operators are deployed to create your OpenShift cluster

- **Machine Config Operator** (MCO) - manages and applies configuration and updates of the base operating system and container runtime

Operators
Power Highlights

- RSCT (from IBM) – used in PowerVC and PowerVS environments to communicate node status

- **L**ocal **S**torage **O**perator – allows local storage to be presented as storage volumes

- **S**pecial **R**esource **O**perator - manages the deployment of software stacks for hardware accelerators

- **O**penShift **A**PI for **D**ata **P**rotection – configures Velero for backup and restore of applications

- Compliance Operator – scans cluster using OpenSCAP to scan and enforce security policies

- File Integrity Operator - continually runs file integrity checks on the cluster nodes, providing a log of files that have been modified

- Service Binding Operator – brokers, using a common mechanism, applications' consumption of back-end services

Container Storage
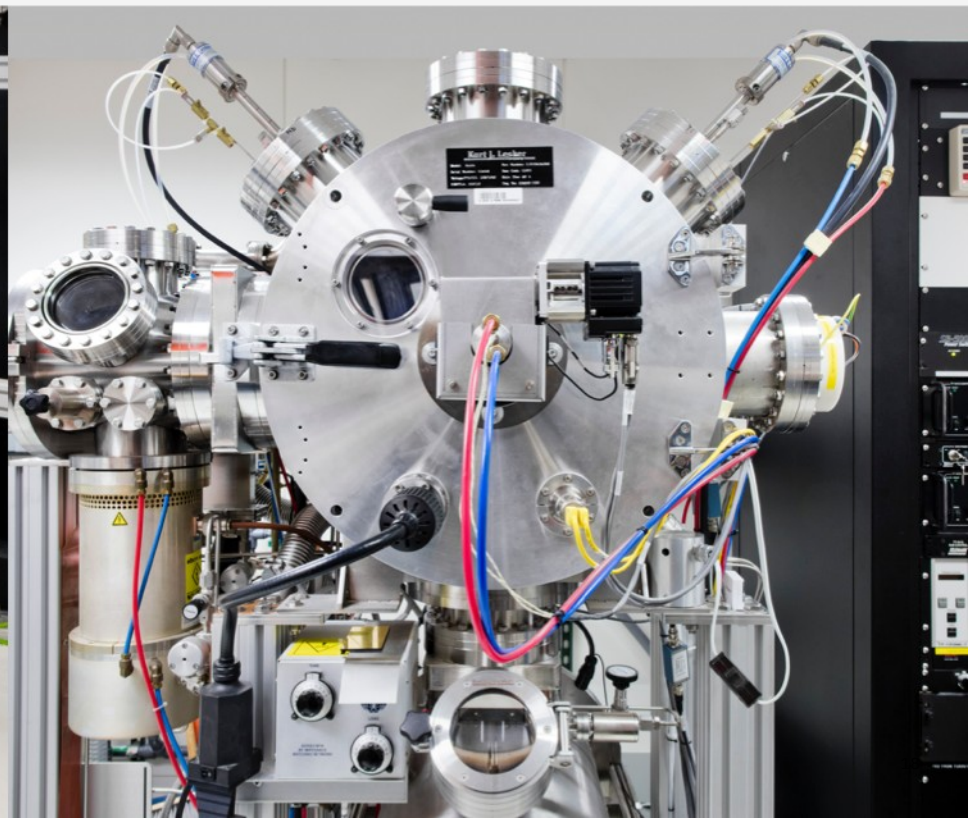
Container Storage Concepts

- **Persistent Volume (PV)** - a piece of storage in the cluster that has been provisioned by an administrator or dynamically provisioned using Storage Classes

- **Persistent Volume Claim (PVC)** - a request for storage by a user

- **Dynamic Provisioning** – removes the burden on cluster administrators to have to manually create new storage volumes and PersistentVolume objects to represent them

- **CSI: Container Storage Interface** – specification to abstract storage access for application use, including creating, deleting, and snapshot of volumes

- **CSI Driver**: Implementing the CSI for a specific platform or storage (e.g. IBM Spectrum® Scale CSI driver)

Storage Provisioners for Power

- Local (host) storage

- NFS (commonly installed on helper node)

- vSCSI

- PowerVC CSI

- **O**penShift **D**ata **F**oundations

  - Ceph (distributed) & Rook-based containerized volume management

- IBM Spectrum Scale CSI Driver

- Power VS Block Storage CSI Driver (TP)

Storage Options for Power

Storage	Type	Power Support (First OpenShift Release)
Raw Block	RWO	4.7
NFS (Spectrum Scale CES, no dynamic provisioning)	RWX	4.3
PowerVC CSI (Container Storage Interface)	RWO	4.3
IBM Spectrum Virtualize (CSI) Block	RWO	4.3
IBM Spectrum Scale File (Container Naive Storage)	RWX	4.5
Fibre Channel	RWO	4.7
iSCSI	RWO	4.6
HostPath	RWO	4.6
CephFS (OpenShift Data Foundations) File	RWX	4.7
CephRBD (OpenShift Data Foundations) Block	RWO	4.7

RWX = read-write many (shared)
RWO = read-write only (dedicated)

Day 1 & 2: Configuring & Upgrades
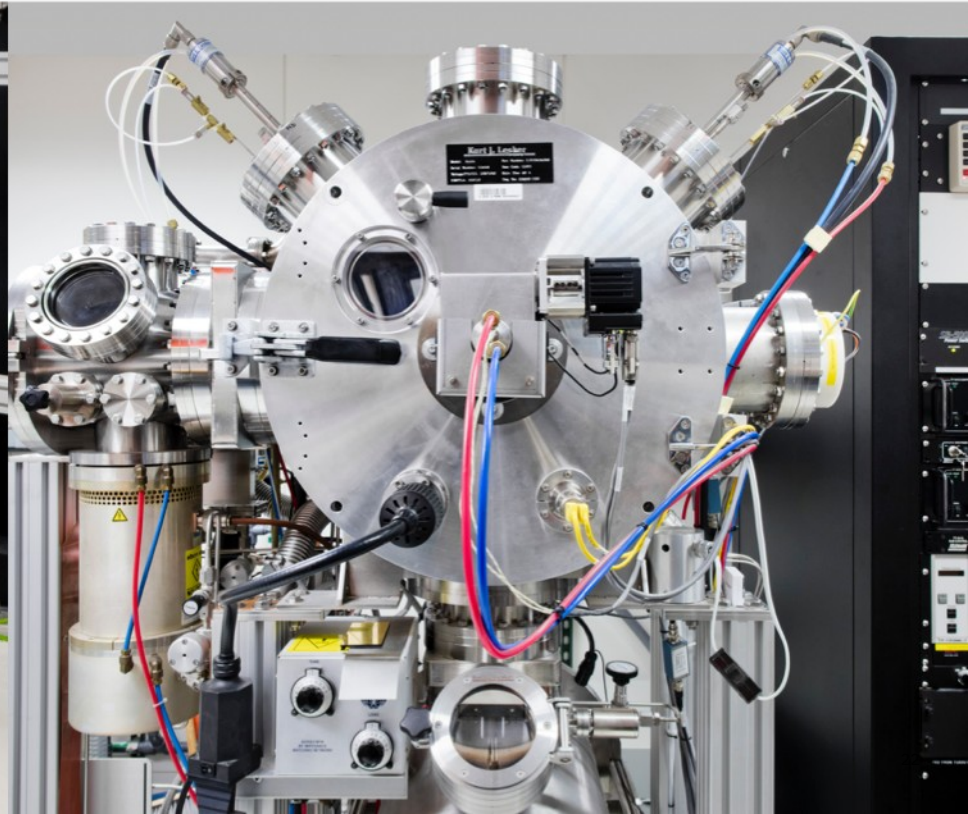
Administration Concepts

- **Namespace** – cluster resource scoping

- **Project** - namespace with additional annotations and is the central vehicle by which access to resources for regular users is managed

- **kubeconfig** – local file containing cluster credentials; used by the OpenShift and Kubernetes CLIs

- **kubeadmin** - the default user created by OpenShift

- **Certificate Rotation –** OpenShift security feature that auto-generates new cluster certificates

- **IBM COS** – s3-compatible Cloud Object Storage on IBM Cloud (can be used for backups)

Day 1
Configuring

Day 1

- Machine Config Operator

  - Make any node-specific changes via MCO

- Add IBM Operator Catalog
- PowerVC & PowerVS: Install the RSCT Operator
- Limit the Overlay file system partition size
- Configure OAuth to specify an identity provider
- Configure users & RBAC policies using `oc adm`
- Logging & Monitoring Alerts

Make Control Plane Nodes Schedulable

Procedure

1. Edit the `schedulers.config.openshift.io` resource.

```
$ oc edit schedulers.config.openshift.io cluster
```

2. Configure the `mastersSchedulable` field.

```
apiVersion: config.openshift.io/v1
kind: Scheduler
metadata:
  creationTimestamp: "2019-09-10T03:04:05Z"
  generation: 1
  name: cluster
  resourceVersion: "433"
  selfLink: /apis/config.openshift.io/v1/schedulers/cluster
  uid: a636d30a-d377-11e9-88d4-0a60097bee62
spec:
  mastersSchedulable: false  ❶
status: {}
```

❶ Set to `true` to allow control plane nodes to be schedulable, or `false` to disallow control plane nodes to be schedulable.

3. Save the file to apply the changes.

Day 2
Upgrades

Day 2

- Configure OADP for Backups

- Add & Remove Worker Nodes

  • IPI: Add new by editing machine set

  • UPI: Manually configure, then install new nodes with ignition config from cluster install

- Add/Remove CPU & Memory (e.g. DLPAR)

  • Restart node (kubelet) to pick up CPU or memory changes

Demos

Links

https://kubernetes.io/docs/concepts/services-networking/service

https://docs.openshift.com/container-platform/4.11/post_installation_configuration/machine-configuration-tasks.html#understanding-the-machine-config-operator

Operator SDK Project: https://sdk.operatorframework.io

Operator Lifecycle Manager: https://olm.operatorframework.io/docs

Operator Package Manager: https://docs.openshift.com/container-platform/4.11/cli_reference/opm/cli-opm-install.html

https://github.com/ocp-power-automation/rsct-operator/blob/main/README.md

https://github.com/openshift/special-resource-operator

https://github.com/openshift/oadp-operator

https://github.com/openshift/compliance-operator

https://github.com/openshift/file-integrity-operator

https://github.com/redhat-developer/service-binding-operator

Links

Disk performance validation: https://access.redhat.com/solutions/4885641

https://docs.openshift.com/container-platform/4.11/installing/installing_ibm_power/installing-ibm-power.html (points to RHCOS download procedure)

https://developer.ibm.com/tutorials/getting-started-odf-on-power
https://developer.ibm.com/learningpaths/exploring-openshift-powervs

https://www.ibm.com/docs/en/app-connect/container?topic=access-enabling-operator-catalog

https://developer.ibm.com/tutorials/install-spectrum-scale-cnsa-5121-on-ocp-48-on-powervs

https://www.ibm.com/docs/en/spectrum-scale-csi

https://docs.openshift.com/container-platform/4.11/post_installation_configuration/machine-configuration-tasks.html#set-the-default-max-container-root-partition-size-for-overlay-with-crio_post-install-machine-configuration-tasks

https://github.com/ocp-power-automation/rsct-operator/blob/main/README.md

Thank you

Christy Norman
Senior Software Engineer

—

clnperez@us.ibm.com
ibm.com