Red Hat OpenShift:
Breaking Beyond Legacy Installs

—

Noel Colón, GTM Technology & Assets
John Abbott, Technology Garage

IBM
Fast Start
2021

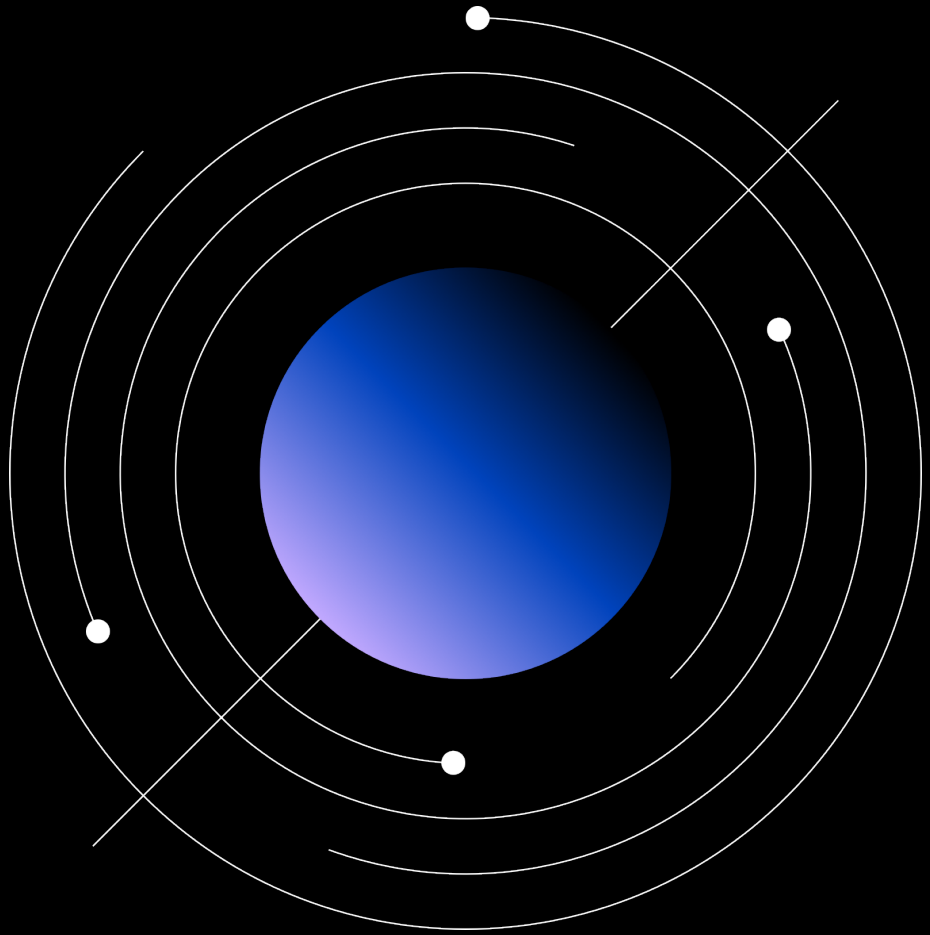
IBM

Planning

A client wants to deploy an application leveraging our Cloud Paks on top of OpenShift

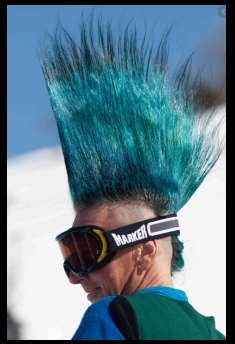
What do I need to plan for in order to be successful?

- Networking
- Compute
- Storage
- Security



Sizing Clusters for Cloud Paks

Go Big



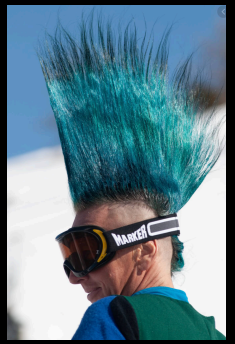
Sizing for Cloud Paks is often over simplified. Architects and SMEs are asked to size OpenShift clusters for Cloud Paks without all of the required information. Additional factors that must be considered when performing a sizing effort:

- Identify the required capabilities, runtimes and services for each Cloud Pak the cluster must support
- The number of instances of each included capability and runtime as well as the topology for complex services
- Compute, Memory and Storage for capability containers (or for each capability / runtime / service)

https://ibm.biz/cpsizeit

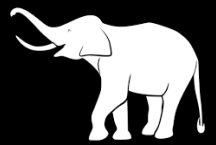
Sizing Clusters for Cloud Paks

Go Big



Sizing for Cloud Paks is often over simplified. Architects and SMEs are asked to size OpenShift clusters for Cloud Paks without all of the required information. Additional factors that must be considered when performing a sizing effort:

- Identify the required capabilities, runtimes and services for each Cloud Pak the cluster must support
- The number of instances of each included capability and runtime as well as the topology for complex services
- Compute, Memory and Storage for capability containers (or for each capability / runtime / service)
- Persistent Storage required sizes, providers, access modes, etc.



Many (if not most) of the capabilities provided by Cloud Paks have state and thus require Persistent Storage. The associated persistent storage provider will depend largely on the cloud environment you are deploying to. Each **public cloud** provider has their own storage services that may (and in most cases should) be used for your workload's persistent storage. These providers will typically match the requirements provided by the product teams. Take time to understand each of the following facets:

•Storage type: How the storage is used such as block, file, object, ...

•Access mode: How many pods can access the same storage? Are pods able to write and read or only read? Each storage provider determines these access modes and matching this ability to the workload is vital.

•Performance: Typically measured in IOPS. Certain providers have different performance characteristics that must be considered.

•Security Context: Especially with file storage providers, in some instances there are security context considerations that must be adhered to in order for workload to successfully use certain storage.

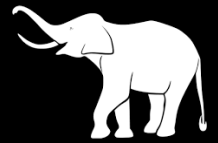
https://ibm.biz/cpsizeit

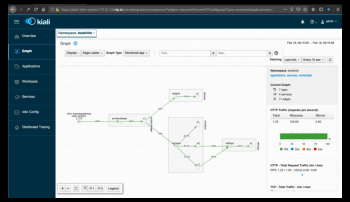
Sizing Clusters for Cloud Paks

Go Big

Sizing for Cloud Paks is often over simplified. Architects and SMEs are asked to size OpenShift clusters for Cloud Paks without all of the required information. Additional factors that must be considered when performing a sizing effort:

- Identify the required capabilities, runtimes and services for each Cloud Pak the cluster must support
- The number of instances of each included capability and runtime as well as the topology for complex services
- Compute, Memory and Storage for capability containers (or for each capability / runtime / service)
- Persistent Storage required sizes, providers, access modes, etc.
- Additional services / applications included in the solution (pipelines, service mesh, tools, utilities, monitoring, services)





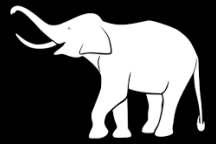
https://ibm.biz/cpsizeit

Sizing Clusters for Cloud Paks

Go Big

Sizing for Cloud Paks is often over simplified. Architects and SMEs are asked to size OpenShift clusters for Cloud Paks without all of the required information. Additional factors that must be considered when performing a sizing effort:

- Identify the required capabilities, runtimes and services for each Cloud Pak the cluster must support
- The number of instances of each included capability and runtime as well as the topology for complex services
- Compute, Memory and Storage for capability containers (or for each capability / runtime / service)
- Persistent Storage required sizes, providers, access modes, etc.
- Additional services / applications included in the solution (pipelines, service mesh, tools, utilities, monitoring, services)
- Load expectations ie. traffic volume for each capability (for some capabilities)
- Hosting platform / hypervisor / cloud (how easy does it become to change size)
- Head-Space (unused space in the cluster required for OpenShift to schedule & reschedule workload)
- Expected reliability of the cluster and its workload



https://ibm.biz/cpsizeit

Architecting for Reliability

Architecting for Reliability
Blog by Ingo Averdunk
https://ibm.biz/reliable-arch

Reliability	Reliability describes the ability of a system or component to function under stated conditions for a specified period of time.
Availability	Availability is the proportion of time a system is in a functioning condition. Availability is one aspect to describe Reliability, but it is not limited to Availability. Examples of other aspects are Latency, Quality, Correctness, and Durability.
Resilience	Resilience is the ability of a system to recover from certain types of failure and yet remain functional from the user perspective. Disruptive events cannot always be anticipated, but when they occur they should result in learning and adaptation.
Antifragility	Antifragility is a property of systems that increase in capability to thrive as a result of stressors, shocks, volatility, noise, mistakes, faults, attacks, or failures.

Reliability cannot simply be delegated to the infrastructure and/or platform a service is running on. Every tier needs to provide their contribution towards the reliability goal of the entire system.

Reliability cannot be added after the fact on top of a built system, the system and its components need to be designed and implemented with reliability in mind.

It is a shared responsibility by everyone contributing across the Software Development Lifecycle, including the Architect, the Product Owner, the *conscientious* DevOps Engineer, and the SRE.




“
/rɪˌlaɪəˈbɪlɪti/ - Reliability describes the ability of a system or component to function under stated conditions for a specified period of time.
”

- Wikipedia

Sizing Clusters for Cloud Paks – Add it Up

https://ibm.biz/cpsizeit

Once you have determined the total capacity for each cluster, determine the best way to divide that capacity among several worker nodes.

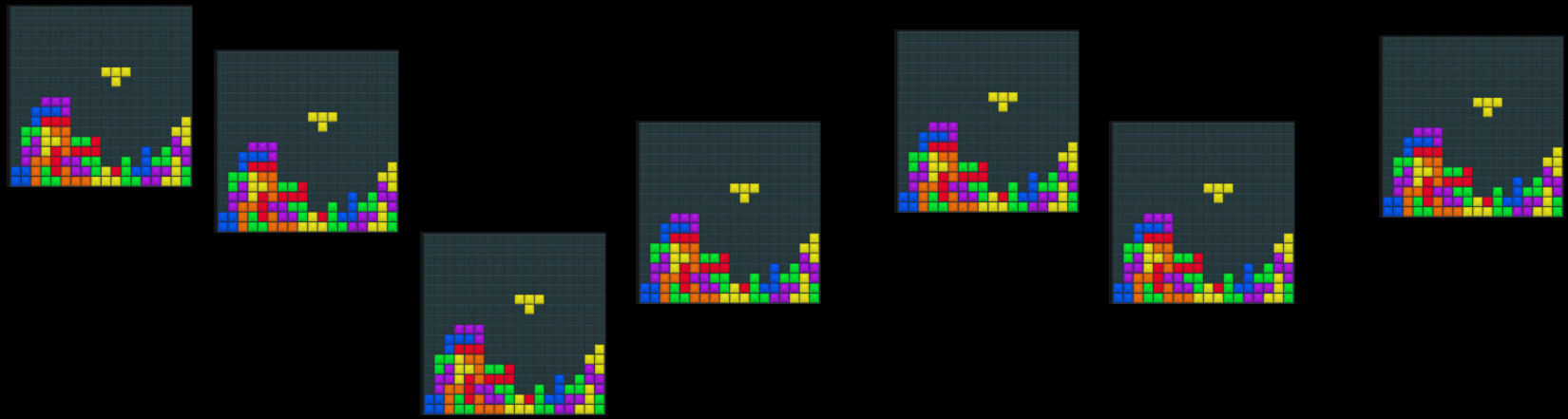
- Arbitrary Statement: It is a best-practice to use small sizing for nodes until you get to at least 10 (unless you are forced to have larger nodes due to large container sizes).
- Having a greater number of nodes decreases the rescheduling activity for OpenShift upon node failure as well as decreases the required head-space your cluster needs.
- As the cluster grows larger, increase your node size. Maintaining too many nodes requires additional overhead for the cluster control plane.
- Avoid designating workload to specific nodes. This breaks the fundamental benefits of using container orchestration … but please reference earlier conversations about machine sets.
- For greatest efficiency shape (ratio of CPU to Memory) your nodes to match your workload. The ideal shape for your nodes will likely be between 2 * CPU = Memory (Java workload) and 4 * CPU = Memory (cloud native workload). Estimate, run, measure and adjust.
- Warning: Clusters should not span regions (unless you are very very convincing and really understand what you are doing)
- Example: If your cluster spans 3 failure domains (availability zones) within a datacenter you will require between 40-50% total healthy cluster head-space to account for the loss of a single availability zone. (To be specific, take the entire projected workload and be certain it will fit within the running nodes in the remaining availability zones with required remaining head-space to be healthy. Chaos test this use case.)

Sizing Clusters for Cloud Paks – Add it Up

https://ibm.biz/cpsizeit

Once you have determined the total capacity for each cluster, determine the best way to divide that capacity among several worker nodes.

- Arbitrary Statement: It is a best-practice to use small sizing for nodes until you get to at least 10 (unless you are forced to have larger nodes due to large container sizes).



Sizing Clusters for Cloud Paks – Add it Up

https://ibm.biz/cpsizeit

Once you have determined the total capacity for each cluster, determine the best way to divide that capacity among several worker nodes.

- Arbitrary Statement: It is a best-practice to use small sizing for nodes until you get to at least 10 (unless you are forced to have larger nodes due to large container sizes).
- Having a greater number of nodes decreases the rescheduling activity for OpenShift upon node failure as well as decreases the required head-space your cluster needs.

Sizing Clusters for Cloud Paks – Add it Up

https://ibm.biz/cpsizeit

Once you have determined the total capacity for each cluster, determine the best way to divide that capacity among several worker nodes.

- Arbitrary Statement: It is a best-practice to use small sizing for nodes until you get to at least 10 (unless you are forced to have larger nodes due to large container sizes).
- Having a greater number of nodes decreases the rescheduling activity for OpenShift upon node failure as well as decreases the required head-space your cluster needs.
- As the cluster grows larger, increase your node size. Maintaining too many nodes requires additional overhead for the cluster control plane.

Sizing Clusters for Cloud Paks – Add it Up

https://ibm.biz/cpsizeit

Once you have determined the total capacity for each cluster, determine the best way to divide that capacity among several worker nodes.

- Arbitrary Statement: It is a best-practice to use small sizing for nodes until you get to at least 10 (unless you are forced to have larger nodes due to large container sizes).
- Having a greater number of nodes decreases the rescheduling activity for OpenShift upon node failure as well as decreases the required head-space your cluster needs.
- As the cluster grows larger, increase your node size. Maintaining too many nodes requires additional overhead for the cluster control plane.
- Avoid designating workload to specific nodes. This breaks the fundamental benefits of using container orchestration … but please reference earlier conversations about machine sets.

Rule number 1: OpenShift is better at scheduling workload than you are!

Sizing Clusters for Cloud Paks – Add it Up

https://ibm.biz/cpsizeit

Once you have determined the total capacity for each cluster, determine the best way to divide that capacity among several worker nodes.

- Arbitrary Statement: It is a best-practice to use small sizing for nodes until you get to at least 10 (unless you are forced to have larger nodes due to large container sizes).
- Having a greater number of nodes decreases the rescheduling activity for OpenShift upon node failure as well as decreases the required head-space your cluster needs.
- As the cluster grows larger, increase your node size. Maintaining too many nodes requires additional overhead for the cluster control plane.
- Avoid designating workload to specific nodes. This breaks the fundamental benefits of using container orchestration … but please reference earlier conversations about machine sets.
- For greatest efficiency shape (ratio of CPU to Memory) your nodes to match your workload. The ideal shape for your nodes will likely be between 2 * CPU = Memory (Java workload) and 4 * CPU = Memory (cloud native workload). Estimate, run, measure and adjust.

Sizing Clusters for Cloud Paks – Add it Up

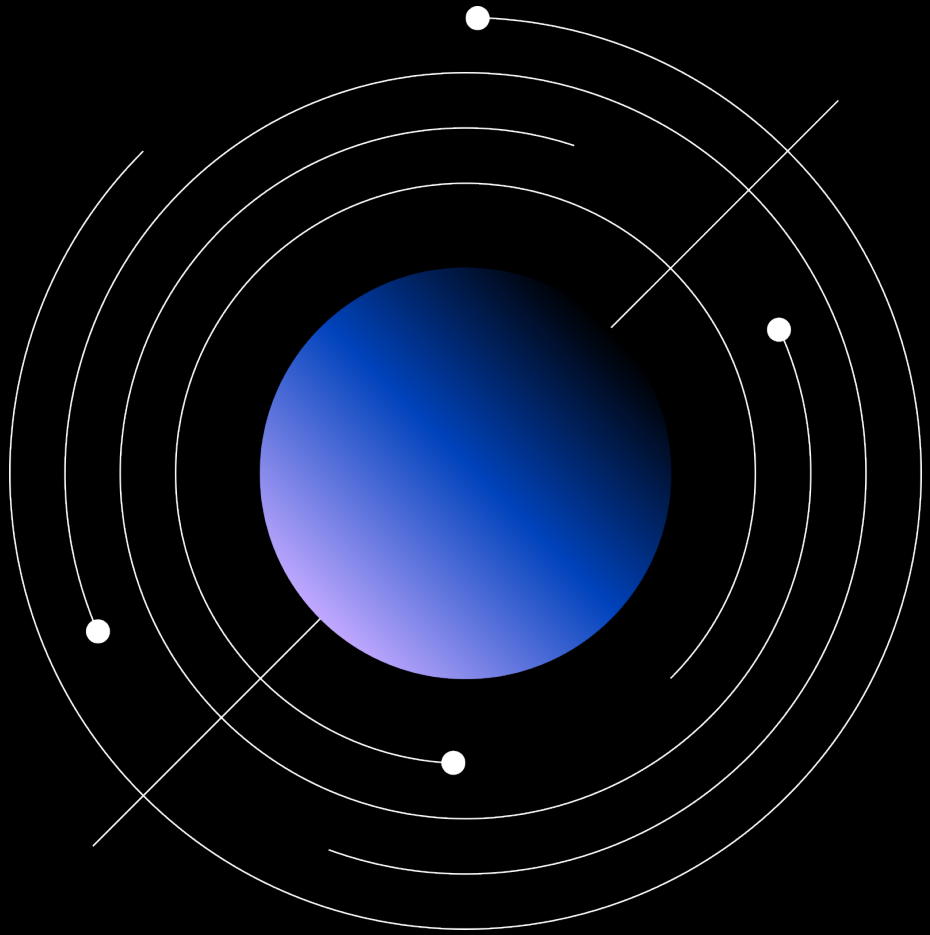
https://ibm.biz/cpsizeit

Once you have determined the total capacity for each cluster, determine the best way to divide that capacity among several worker nodes.

- Arbitrary Statement: It is a best-practice to use small sizing for nodes until you get to at least 10 (unless you are forced to have larger nodes due to large container sizes).
- Having a greater number of nodes decreases the rescheduling activity for OpenShift upon node failure as well as decreases the required head-space your cluster needs.
- As the cluster grows larger, increase your node size. Maintaining too many nodes requires additional overhead for the cluster control plane.
- Avoid designating workload to specific nodes. This breaks the fundamental benefits of using container orchestration … but please reference earlier conversations about machine sets.
- For greatest efficiency shape (ratio of CPU to Memory) your nodes to match your workload. The ideal shape for your nodes will likely be between 2 * CPU = Memory (Java workload) and 4 * CPU = Memory (cloud native workload). Estimate, run, measure and adjust.
- Warning: Clusters should not span regions (unless you are very very convincing and really understand what you are doing)
- Example: If your cluster spans 3 failure domains (availability zones) within a datacenter you will require between 40-50% total healthy cluster head-space to account for the loss of a single availability zone. (To be specific, take the entire projected workload and be certain it will fit within the running nodes in the remaining availability zones with required remaining head-space to be healthy. Chaos test this use case.)

Networking Considerations

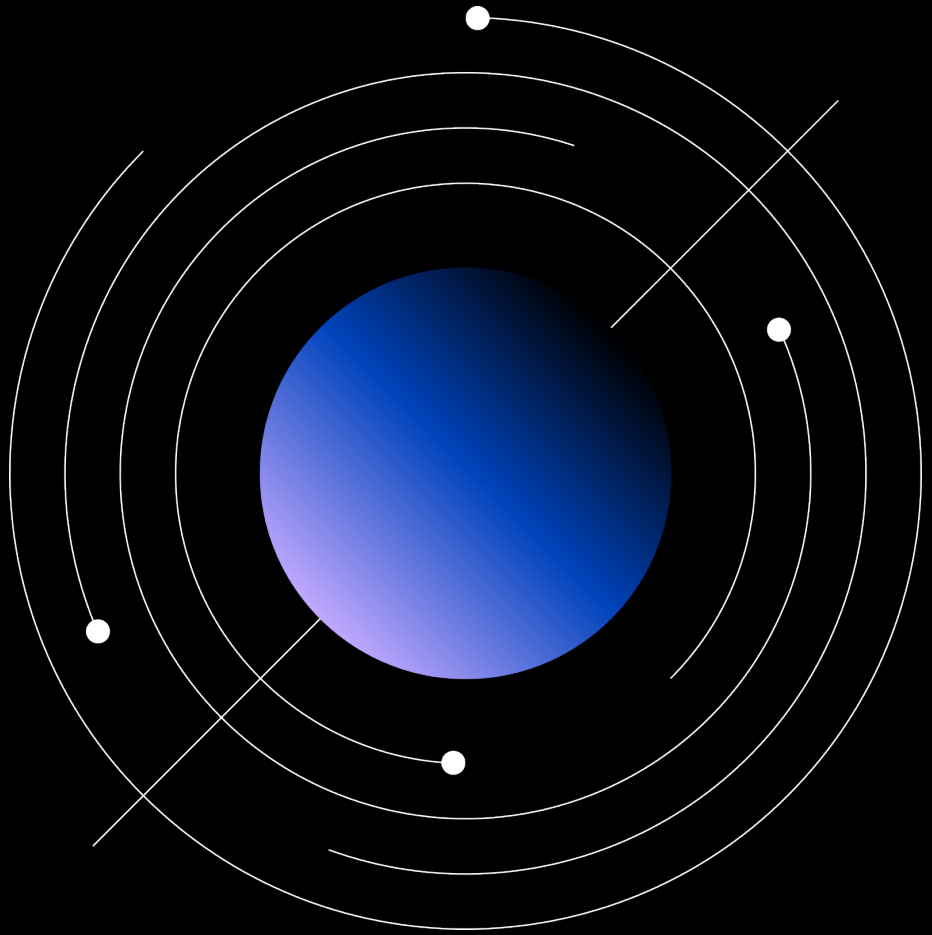
- Regions
- Availability Zones (Failure Domains)
- Networking Services
- Load Balancing
- Multiple Networks



Compute

How many environments does your customer need?
How should I size these environments?

- **DEV**
- **QA**
- **STAGE**
  - Mirror PROD
- **PROD**
  - Utilization Thresholds
  - Unavailable Nodes
  - GeoLoadbalance multiple Regions



Compute

Masters
- **3 Nodes**
- **8 CPU**
- **32 GB Memory**

Over Commit Warning!!!

Rule number 1: Keep your over commitment closest to your workload.
(ie. In OpenShift not your Hypervisor)

Compute

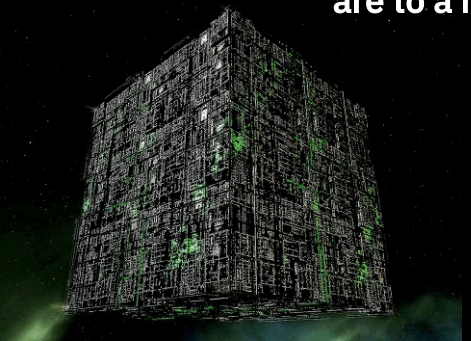
Masters
- **3 Nodes**
- **8 CPU**
- **32 GB Memory**

Workers (MachineSets)

Wait … what is a machine … and what is a machine set?

A machine is a CRD (custom resource definition representation of the underlying hardware

A machine set is a group of machines.
A machine set is to machines as pods are to a replica set



It’s a little-known fact that we can create worker nodes from our machine sets

Compute

Masters
- **3 Nodes**
- **8 CPU**
- **32 GB Memory**

Workers (MachineSets)

- **Infrastructure**
  - At least 2 Nodes
  - 4 CPU
  - 16 GB Memory

Rule number 1: This does just make you pragmatic, its what the kewl kids do!

Compute

Masters
- **3 Nodes**
- **8 CPU**
- **32 GB Memory**

Workers (MachineSets)

- **Infrastructure**
  - At least 2 Nodes
  - 4 CPU
  - 16 GB Memory

- **Storage**
  - 3 Nodes
  - 16 CPU
  - 64 GB Memory

Rule number 1: This could be the most important decision you make!

Compute

Masters
- **3 Nodes**
- **8 CPU**
- **32 GB Memory**

Workers (MachineSets)

- **Infrastructure**
  - At least 2 Nodes
  - 4 CPU
  - 16 GB Memory
- **Storage**
  - 3 Nodes
  - 16 CPU
  - 64 GB Memory
- **CloudPaks**
  - 2 Nodes
  - 8 CPU
  - 32 GB Memory

Rule number 1: Ignore this ridiculous minimal sizing

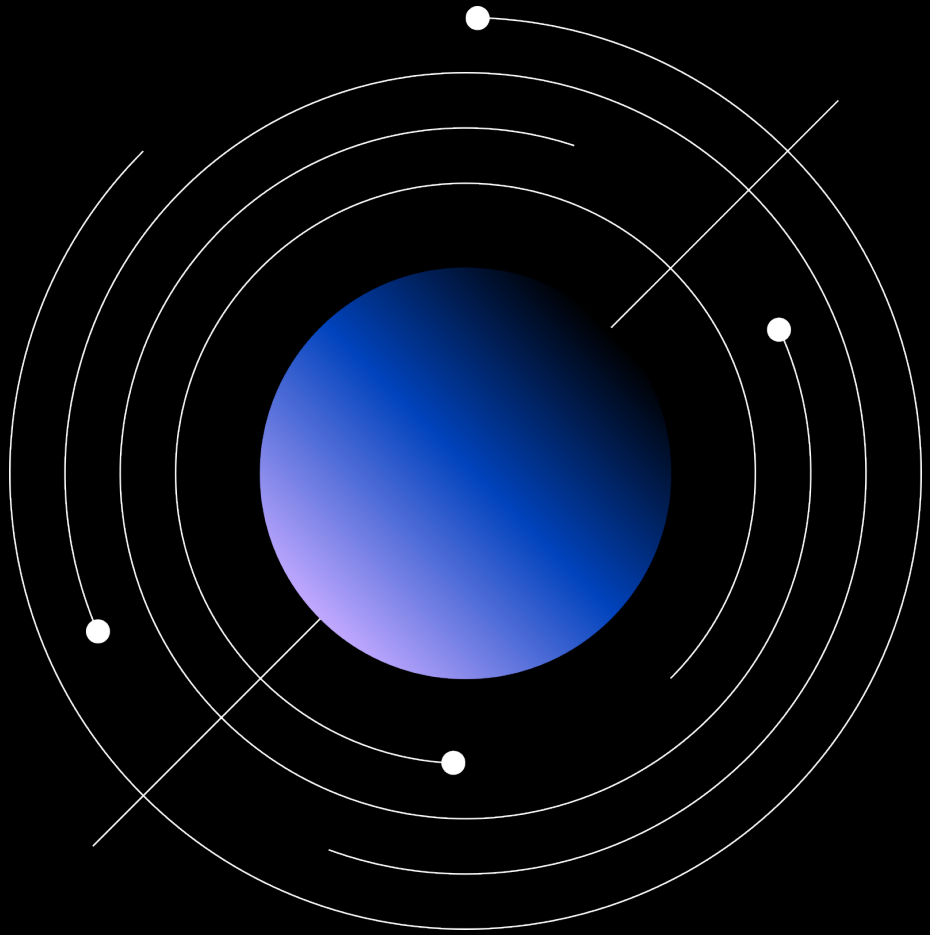
Go Big

Compute

Multiple CloudPaks?

Installing into Existing OpenShift Cluster?

- **RHCOS – Immutable OS**
  - sysctl
  - CRI-O config
- **Machine Config Operator**
  - **MachineConfigPool**
  - **MachineConfig**
  - **ContainerRuntimeConfig**
  - **MachineSets**
- **AutoScaling**
  - **HorizontalPodAutoscaler**
  - **ClusterAutoScaler**
  - **MachineAutoScaler**



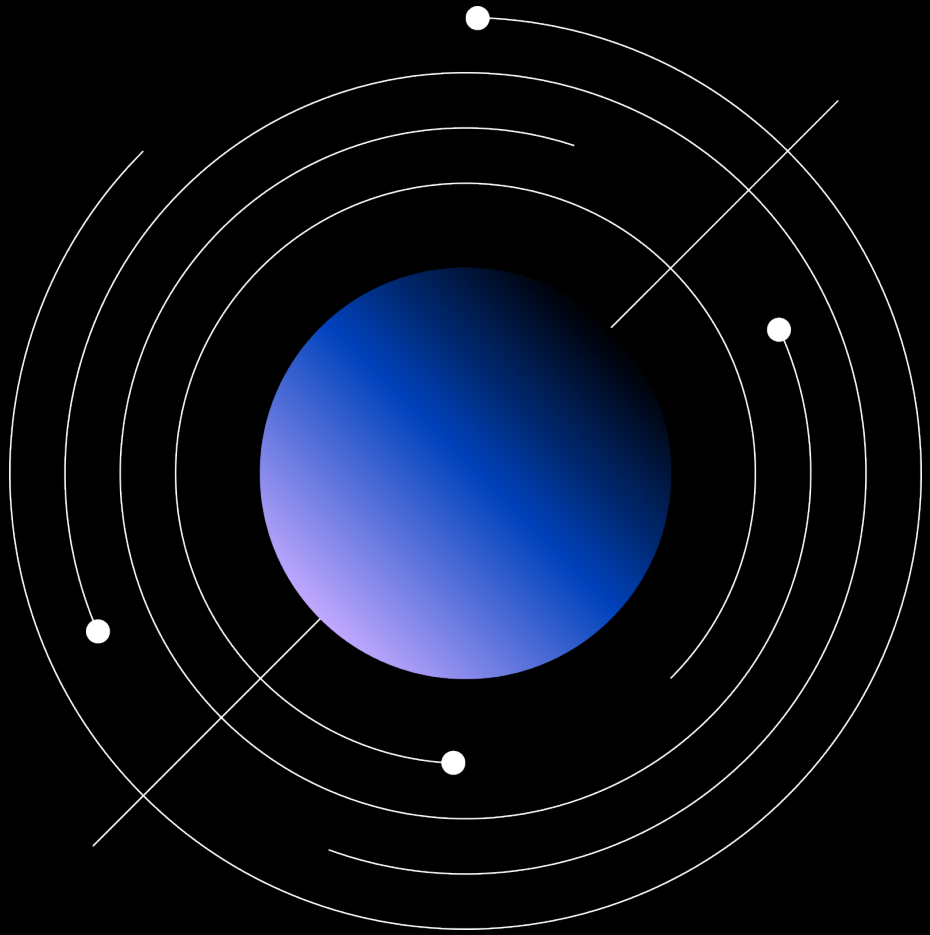
Storage Considerations

OpenShift Node Storage
- SSD/NVMe

Cluster and CloudPak Configuration Storage
- Image Registry
- Logging/Monitoring

Application Data
- File
- Block
- Object

Storage Options
- rook-ceph
- NFS
- OpenShift Container Storage
- IBM Storage Suite for CloudPaks

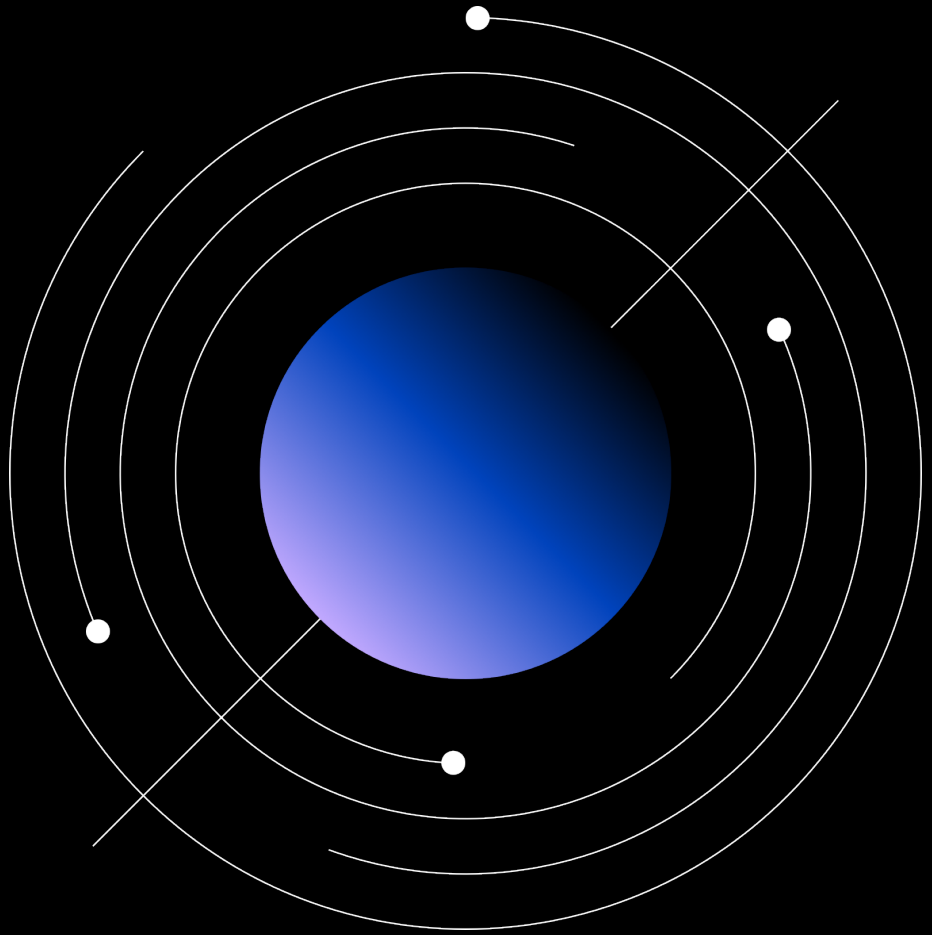


Security Considerations

Disconnected Environments
- Proxy
- Internal Registry
  - Platform
  - Marketplace
  - CloudPaks

Multi Tenancy
- NetworkPolicies
- Machine Config Operator

Self Provisioning
- DEV – OK
- QA/STAGE/PROD – Pipelines



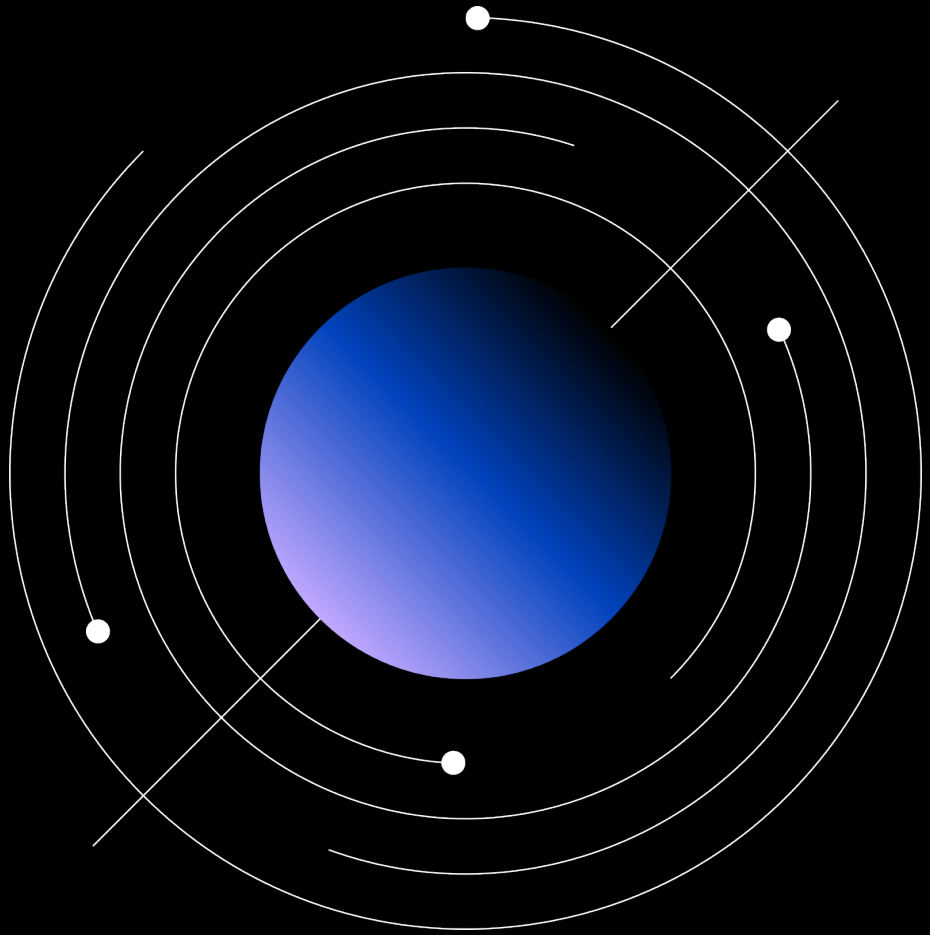
Assets

OpenShift Deployment
Use IPI deployment whenever possible
http://cloud.redhat.com/openshift/install

... But if you need UPI
https://github.com/ibm-cloud-architecture/terraform-openshift4-vmware
https://github.com/ibm-cloud-architecture/terraform-openshift4-aws
https://github.com/ibm-cloud-architecture/terraform-openshift4-azure
https://github.com/ibm-cloud-architecture/terraform-openshift4-gcp

Cloud-Native Toolkit
https://cloudnativetoolkit.dev

CSM Playbook for OpenShift TBD



References

OpenShift Multus

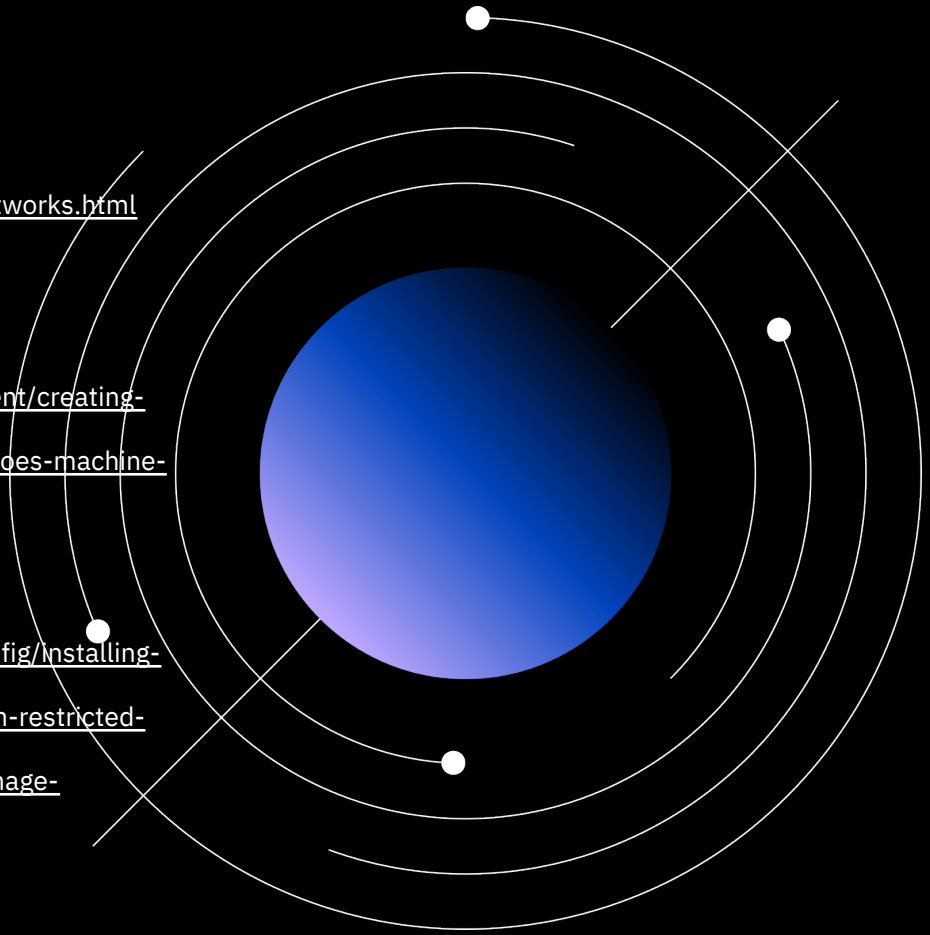
- https://docs.openshift.com/container-platform/4.6/networking/multiple_networks/understanding-multiple-networks.html
- https://www.openshift.com/blog/demystifying-multus

OpenShift Machine Config Operator

- https://docs.openshift.com/container-platform/4.6/machine_management/creating-infrastructure-machinesets.html
- https://www.redhat.com/en/blog/openshift-container-platform-4-how-does-machine-config-pool-work

OpenShift Restricted Networks

- https://docs.openshift.com/container-platform/4.6/installing/install_config/installing-restricted-networks-preparations.html
- https://docs.openshift.com/container-platform/4.6/operators/admin/olm-restricted-networks.html
- https://docs.openshift.com/container-platform/4.6/openshift_images/image-configuration.html



Thank you

Noel Colón, ncolon@us.ibm.com
1-787-688-8466
GTM Assets and Architecture

John Abbott, jabbott@us.ibm.com
1-612-220-1550
IBM Technology Garage