Rook Best Practices for Running Ceph on Kubernetes

SUSE Enterprise Storage, Ceph, Rook, Kubernetes, Container-as-a-Service Platform

Blaine Gardner, Senior Software Developer, SUSE

Alexandra Settle, Senior Information Developer, SUSE

The document at hand provides an overview of the best practices and tested patterns of using Rook v1.3 to manage your Ceph Octopus cluster running in Kubernetes.

Publication Date: May 27, 2020

Contents

1 Overview

Ceph and Kubernetes are both complex tools and harmonizing the interactions between the two can be daunting. This is especially true for users who are new to operating either system, prompting questions such as:

- How can I restrict Ceph to a portion of my nodes?
- Can I set Kubernetes CPU or RAM limits for my Ceph daemons?
- What are some ways to get better performance from my cluster?

This document covers tested patterns and best practices to answer these questions and more. Our examples will help you configure and manage your Ceph cluster running in Kubernetes to meet your needs. The following examples and advice are based on Ceph Octopus (v15) with Rook v1.3 running in a Kubernetes 1.17 cluster.

This is a moderately advanced topic, so basic experience with Rook is recommended. Before you begin, ensure you have the following requisite knowledge:

- Basics of Kubernetes
- How to create Kubernetes applications using manifests
- Kubernetes topics:
  - Pods
  - Nodes
  - Labels
  - Topology
  - Taints and tolerations
  - Affinity and Anti-affinity
  - Resource requests
  - Limits
- Ceph components and daemons, basic Ceph configuration
- Rook basics and how to install Rook-Ceph. For more information see https://rook.io/docs/rook/v1.3/ceph-storage.html

In places, we will give examples that describe an imaginary data center. This data center is hypothetical, and it will focus on the Ceph- and Rook-centric elements and ignore user applications.

Our example data center has two rooms for data storage. A properly fault tolerant Ceph cluster should have at least three monitor (MON) nodes. These should be spread across fault-tolerant rooms if possible. The example will have a separate failure domain for the third monitor node. As such, our hypothetical data center has two rooms and one failure domain, with the following configuration:

- The failure domain is small and can only to be used for the third Ceph MON; it does not have space for storage nodes.
- Eight OSD nodes provide a good amount of data safety without requiring too many nodes.
- These eight nodes should be equally separated — four to each data center room.
- The four nodes are separated in each room into two racks.
- In the event of a MON node failure, ensure that you can run MONs on each rack for failure scenarios.

The data center looks as follows:

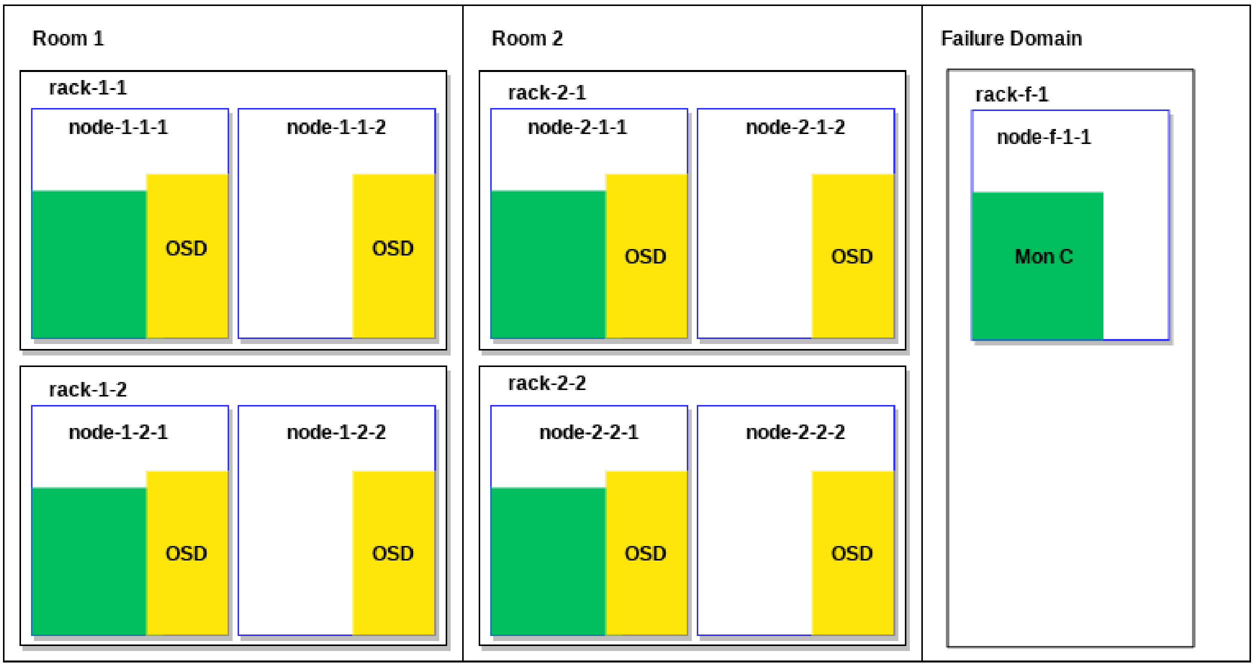

FIGURE 1: EXAMPLE DATACENTER

Now we will dig a little deeper and talk about the actual disks used for Rook and Ceph storage. To ensure we are following known Ceph best practices for this data center setup, ensure that MON storage goes on the SSDs. Because each rack should be able to run a Ceph MON, one of the nodes in each rack will have an SSD that is usable for MON storage. Additionally, all nodes in all racks (except in the failure domain) will have disks for OSD storage. This will look like the following:

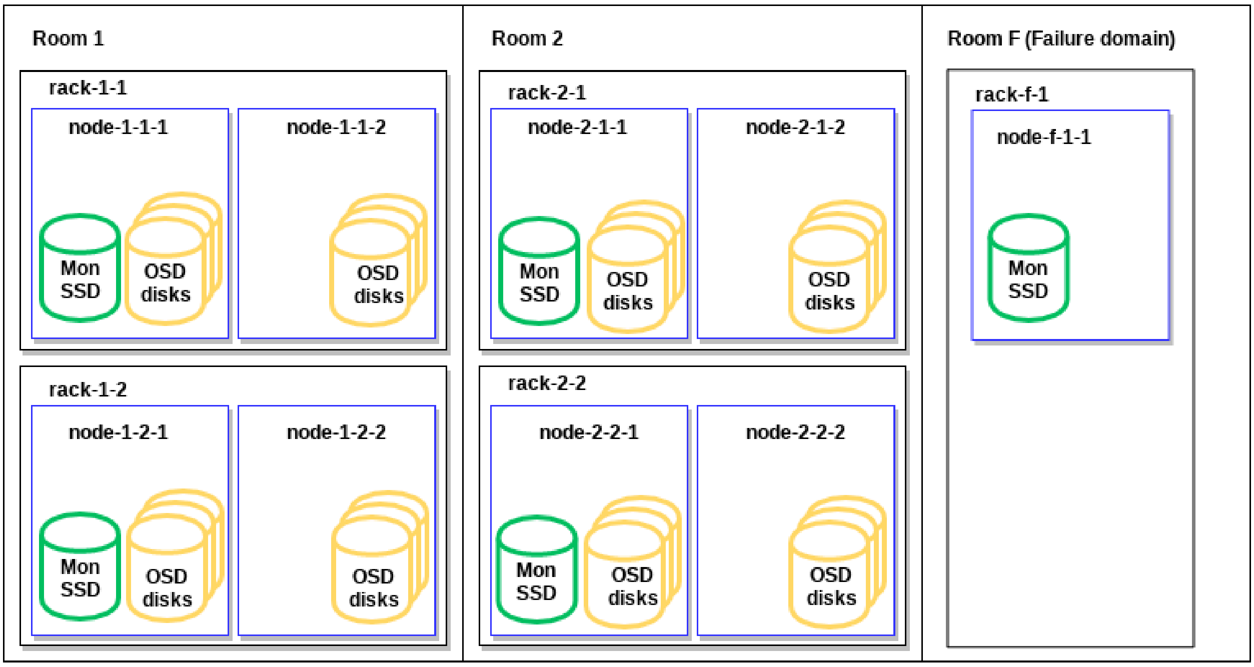

FIGURE 2: EXAMPLE DATACENTER - DISKS

Important: Diagrams

Refer to these diagrams when we discuss the example data center below.

2 Introduction

Ceph and Kubernetes both have their own well-known and established best practices. Rook bridges the gap between Ceph and Kubernetes, putting it in a unique domain with its own best practices to follow. This document specifically covers best practice for running Ceph on Kubernetes with Rook. Because Rook augments on top of Kubernetes, it has different ways of meeting Ceph and Kubernetes best practices. This is in comparison to the bare metal version of each. Out of the box, Rook is predominantly a default Ceph cluster. The Ceph cluster needs tuning to meet user workloads, and Rook does not absolve the user from planning out their production storage cluster beforehand.

For the purpose of this document, we will consider two simplified use cases to help us make informed decisions about Rook and Ceph:

- Co-located: User applications co-exist on nodes running Ceph
- Disaggregated: Ceph nodes are totally separated from user applications

3 General Best Practices

This chapter provides an outline of a series of generalized recommendations for best practices:

- Ceph monitors are more stable on fast storage (SSD-class or better) according to Ceph best practices. In Rook, this means that the `dataDirHostPath` location in the `cluster.yaml` should be backed by SSD or better on MON nodes.
- Raise the Rook log level to `DEBUG` for initial deployment and for upgrades, as it will help with debugging problems that are more likely to occur at those times.
  Ensure that the `ROOK_LOG_LEVEL` in `operator.yaml` equals `DEBUG`.
- The Kubernetes CSI driver is the preferred default but ensure that in `operator.yaml` the `ROOK_ENABLE_FLEX_DRIVER` remains set to `false`. This is because the FlexVolume driver is in sustaining mode, is not getting non-priority bug fixes, and will soon be deprecated.
- Ceph's placement group (PG) auto-scaler module makes it unnecessary to manually manage PGs. We recommend you always set this to `enabled`, unless you have some need to manage PGs manually. In `cluster.yaml`, enable the `pg_autoscaler` MGR module.
- Rook has the capability to auto-remove Deployments for OSDs which are kicked out of a Ceph cluster. This is enabled by: `removeOSDsIfOutAndSafeToRemove: true`. This means there is less user OSD maintenance and no need to delete Deployments for OSDs that have been kicked out. Rook will automatically clean up the cluster by removing OSD Pods if the OSDs are no longer holding Ceph data. However, keep in mind that this can reduce

the visibility of failures from Kubernetes Pod and Pod Controller views. You can optionally set `removeOSDsIfOutAndSafeToRemove` to `false` if need be, such as if a Kubernetes administrator wants to see disk failures via a Pod overview.

- Configure Ceph using the central configuration database when possible. This allows for more runtime configuration flexibility. Do this using the **`ceph config set`** commands from Rook's toolbox. Only use Rook's provided `ceph.conf` to override `ConfigMap` when it is required.

4 Limiting Ceph to Specific Nodes

One of the more common setups you may want for your Rook-Ceph cluster is to limit Ceph to a specific set of nodes. Even for co-located use cases, you could have valid reasons why you must not (or do not want to) use some nodes for storage. This is applicable for both co-located and disaggregated use cases. To limit Ceph to specific nodes, we can Label Kubernetes Nodes and configure Rook to have Affinity (as a hard preference).

Label the desired storage nodes with `storage-node=true`. To run Rook and ceph daemons on labeled nodes, we will configure Rook Affinities in both the Rook Operator manifest (`operator.yaml`) and the Ceph cluster manifest (`cluster.yaml`).

`operator.yaml`

```
CSI_PROVISIONER_NODE_AFFINITY:“storage-node=true”
AGENT_NODE_AFFINITY:“storage-node=true”
DISCOVER_AGENT_NODE_AFFINITY:“storage-node=true”
```

For Rook daemons and the CSI driver daemons, adjust the Operator manifest. The CSI Provisioner is best started on the same nodes as the other Ceph daemons. As above, add affinity for all storage nodes in `cluster.yaml`:

```
placement:
  all:
    nodeAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
        nodeSelectorTerms:
          - matchExpressions:
              - key:storage-node
                operator:In
                values:
                  -"true"
```

5 Segregating Ceph From User Applications

You could also have reason to totally separate Rook and Ceph nodes from application nodes. This falls under the disaggregated use-case, and it is a more traditional way to deploy storage. In this case, we still need to *Section 4, "Limiting Ceph to Specific Nodes"* as described in the section above, and we also need some additional settings.

To segregate Ceph from user applications, we will also label all non-storage nodes with `storage-node=false`. The CSI plugin pods must run where user applications run and not where Rook or Ceph pods are run. Add a CSI plugin Affinity for all non-storage nodes in the Rook operator configuration.

```
CSI_PLUGIN_NODE_AFFINITY:"storage-node=false"
```

In addition to that, we will set Kubernetes Node Taints and configure Rook Tolerations. For example, Taint the storage nodes with `storage-node=true:NoSchedule` and then add the Tolerations below to the Rook operator in `operator.yaml`:

```
AGENT_TOLERATIONS:|
  - key:storage-node
    operator:Exists
```

```
DISCOVER_TOLERATIONS:|
  - key:storage-node
    operator:Exists
```

```
CSI_PROVISIONER_TOLERATIONS:|
  - key:storage-node
    operator:Exists
```

Also add a Toleration for all Ceph daemon Pods in `cluster.yaml`:

```
placement:
  all:
    tolerations:
      - key:storage-node
        operator:Exists
```

6 Setting Ceph CRUSH Map via Kubernetes Node Labels

A feature that was implemented early in Rook's development is to set Ceph's CRUSH map via Kubernetes Node labels. For our example data center, we recommend labelling Nodes with `room`, `rack`, and `chassis`.

As a note, Rook will only set a CRUSH map on initial creation for each OSD associated with the node. It will not alter the CRUSH map if labels are modified later. Therefore, modifying the CRUSH location of an OSD after Rook has created it must be done manually.

For example, in our hypothetical data center, labeling nodes will look like the following:

```
# -- room-1 --

kubectl label node node-1-1-1 topology.rook.io/room=room-1
kubectl label node node-1-1-1 topology.rook.io/rack=rack-1-1
kubectl label node node-1-1-1 topology.rook.io/chassis=node-1-1-1

kubectl label node node-1-1-2 topology.rook.io/room=room-1
kubectl label node node-1-1-2 topology.rook.io/rack=rack-1-1
kubectl label node node-1-1-2 topology.rook.io/chassis=node-1-1-2

kubectl label node node-1-2-1 topology.rook.io/room=room-1
kubectl label node node-1-2-1 topology.rook.io/rack=rack-1-2
kubectl label node node-1-2-1 topology.rook.io/chassis=node-1-2-1

kubectl label node node-1-2-2 topology.rook.io/room=room-1
kubectl label node node-1-2-2 topology.rook.io/rack=rack-1-2
kubectl label node node-1-2-2 topology.rook.io/chassis=node-1-2-2

# -- room-2 --

kubectl label node node-2-1-1 topology.rook.io/room=room-2
kubectl label node node-2-1-1 topology.rook.io/rack=rack-2-1
kubectl label node node-2-1-1 topology.rook.io/chassis=node-2-1-1

kubectl label node node-2-1-2 topology.rook.io/room=room-2
kubectl label node node-2-1-2 topology.rook.io/rack=rack-2-1
kubectl label node node-2-1-2 topology.rook.io/chassis=node-2-1-2

kubectl label node node-2-2-1 topology.rook.io/room=room-2
kubectl label node node-2-2-1 topology.rook.io/rack=rack-2-2
kubectl label node node-2-2-1 topology.rook.io/chassis=node-2-2-1
```

```
kubectl label node node-2-2-2 topology.rook.io/room=room-2
kubectl label node node-2-2-2 topology.rook.io/rack=rack-2-2
kubectl label node node-2-2-2 topology.rook.io/chassis=node-2-2-2

# -- room-f (failure domain) --

kubectl label node node-f-1-1 topology.rook.io/room=room-f
kubectl label node node-f-1-1 topology.rook.io/rack=rack-f-1
kubectl label node node-f-1-1 topology.rook.io/chassis=node-f-1-1
```

7 Planning the Nodes Where Ceph Daemons Will Run

7.1 Ceph MONS

Using the hypothetical data center described in the *Section 1, "Overview"*, this section will look at planning the nodes where Ceph daemons are going to run.

Ceph MON scheduling is one of the more detailed, and more important, things to understand about maintaining a healthy Ceph cluster. The goals we will target in this section can be summarized as: "Avoid risky co-location scenarios, but allow them if there are no other options, to still have as much redundancy as possible."

This can lead us to the following specific goals:

- Allow MONs to be in the same room if a room is unavailable.
- Allow MONs to be in the same rack if no other racks in the room are available.
- Allow MONs to be on the same host only if no other hosts are available.
  We must allow this specifically in the cluster configuration `cluster.yaml` by setting `allowMultiplePerNode: true`.

> **Important**
>
> This cannot be set to `true` for clusters using host networking.

> **Tip: Topology Labels**
>
> We recommend using the same topology labels used for informing the CRUSH map here for convenience.

Because of our MON SSD availability, in our hypothetical data center, we only want monitors to be able to run where shown below in green. We need to plan for monitors to fail over, and so we will make two nodes explicitly available for this scenario. In our example, we want any node with a MON SSD to be a MON failover location in emergencies, for maximum cluster health. This is highlighted in orange below. This will give us the most redundancy under failure conditions.

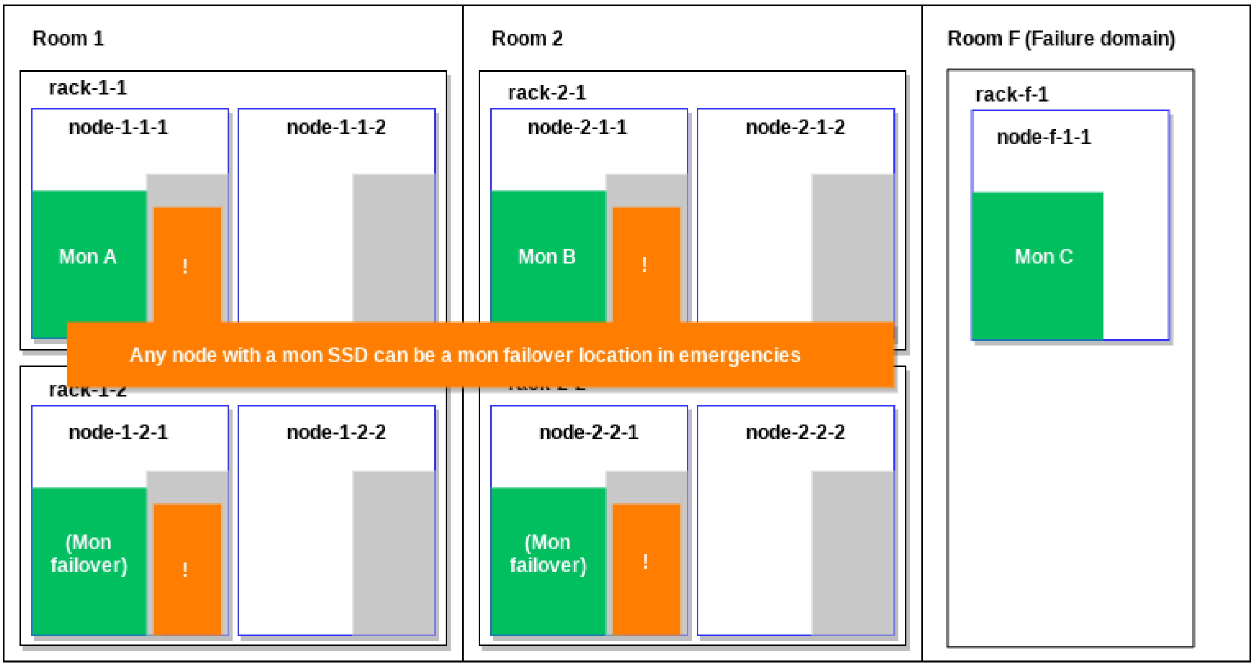

FIGURE 3: EXAMPLE DATACENTER - MON FAILOVER

To implement this in Rook, ensure that Rook will only schedule MONs on nodes with MON SSDs. There is a required Affinity for those nodes, which can be accomplished by applying a `ceph-mon-ssd=true` label to nodes with SSDs for Ceph MONs. Note that the MON section's `nodeAffinity` takes precedence over the `all` section's `nodeAffinity`. Make sure that you re-specify the rules from the all section to ensure Ceph MONs maintain affinity only for storage nodes.

```
nodeAffinity:
  requiredDuringSchedulingIgnoredDuringExecution:
    nodeSelectorTerms:
      - matchExpressions:
          - key:role
            operator:In
            values:
              - storage-node
      - matchExpressions:
          - key:ceph-mon-ssd
            operator:In
```

```
        values:
          -"true"
```

We want to schedule MONs so they are spread across failure domains whenever possible. We will accomplish this by applying Anti-affinity between MON pods. Rook labels all MON pods `app=rook-ceph-mon`, and that is what will be used to spread the monitors apart. There is one rule for rooms, and one for racks if a room is down. We want to ensure a higher weight is given to riskier co-location scenarios:

We do not recommend running MONs on the same node unless absolutely necessary. Rook automatically applies an Anti-affinity with medium-level weight. However, this might not be appropriate for all scenarios. For our scenario, we only want node-level co-location in the worst of failure scenarios, so we want to apply the highest weight Anti-affinity for nodes.

```
cluster.yaml:
placement:
  mon:
    # ... nodeAffinity from above ...
    podAntiAffinity:
      preferredDuringSchedulingIgnoredDuringExecution:
        - weight:80
          podAffinityTerm:
            labelSelector:
              matchLabels:
                app:rook-ceph-mon
            topologyKey:topology.rook.io/room
        - weight:90
          podAffinityTerm:
            labelSelector:
              matchLabels:
                app:rook-ceph-mon
            topologyKey:topology.rook.io/rack
        - weight: 100
          podAffinityTerm:
            labelSelector:
              matchLabels:
                app: rook-ceph-mon
            topologyKey: kubernetes.io/hostname
```

Note

If `hostNetworking` is enabled, you cannot co-locate MONs, because the ports will collide on nodes. To enforce this, if host networking is enabled, Rook will automatically set a `requiredDuringSchedulingIgnoredDuringExecution` Pod Anti-affinity rule.

7.2 Ceph OSDS

There is a lot of planning that goes into the placement of monitors, and this is also true for OSDs. Fortunately, because the planning is already done with the monitors and because we have discussed the methods, it is quite a bit easier to plan for the OSDs.

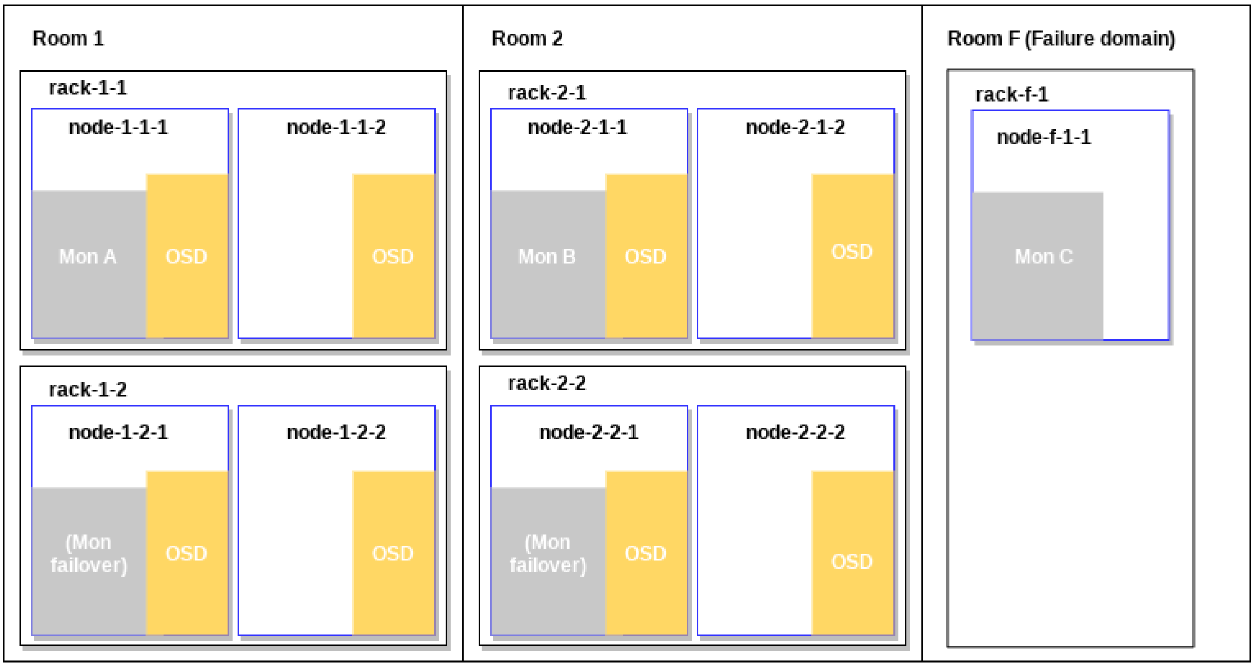

FIGURE 4: EXAMPLE DATACENTER - OSD PLACEMENT

There are two ways to select nodes to use for OSDs:

- Apply Kubernetes Node labels and tell Rook to look for those labels. Specify in the `cluster.yaml` `storage:useAllNodes true` and specify `osd` `nodeAffinity` using `ceph-osd=true` label using the same Affinity methods we used for MONs.
- Specify node names in the `CephCluster` definition (`cluster.yaml`) individually in `storage:nodes`.

Choosing which option to use depends on your desired management strategy. There is no single strategy we would recommend over any other.

7.3 Other Ceph Daemons

Placing the other Ceph daemons follows the same logic and methods as MONs and OSDs: MGR, MDS, RGW, NFS-Ganesha, and RBD mirroring daemons can all be placed as desired. For more information, see https://rook.io/docs/rook/v1.3/ceph-cluster-crd.html#placement-configuration-settings

8 Hardware Resource Requirements and Requests

Kubernetes can watch the system resources available on nodes and can help schedule applications—such as the Ceph daemons—automatically. Kubernetes uses Resource Requests to do this. For Rook, we are notably concerned about Kubernetes' scheduling of Ceph daemons.

Kubernetes has two Resource Request types: *Requests* and *Limits*. *Requests* govern scheduling, and *Limits* instruct Kubernetes to kill and restart application Pods when they are over-consuming given *Limits*.

When there are Ceph hardware requirements, treat those requirements as *Requests*, not *Limits*. This is because all Ceph daemons are critical for storage, and it is best to never set Resource *Limits* for Ceph Pods. If Ceph Daemons are over-consuming *Requests*, there is likely a failure scenario happening. In a failure scenario, killing a daemon beyond a *Limit* is likely to make an already bad situation worse. This could create a “thundering herds” situation where failures synchronize and magnify.

Generally, storage is given minimum resource guarantees, and other applications should be limited so as not to interfere. This guideline already applies to bare-metal storage deployments, not only for Kubernetes.

As you read on, it is important to note that all recommendations can be affected by how Ceph daemons are configured. For example, any configuration regarding caching. Keep in mind that individual configurations are out of scope for this document.

8.1 Resource Requests - MON/MGR

Resource Requests for MONs and MGRs are straightforward. MONs try to keep memory usage to around 1 GB — however, that can expand under failure scenarios. We recommend 4 GB RAM and 4 CPU cores.

Recommendations for MGR nodes are harder to make, since enabling more modules means higher usage. We recommend starting with 2 GB RAM and 2 CPU cores for MGRs. It is a good idea to look at the actual usage for deployments and do not forget to consider usage during failure scenarios.

MONs:

- *Request* 4 CPU cores
- *Request* 4GB RAM (2.5GB minimum)

MGR:

- Memory will grow the more MGR modules are enabled
- *Request* 2 GB RAM and 2 CPU cores

8.2 Resource Requests - OSD CPU

Recommendations and calculations for OSD CPU are straightforward.

Hardware recommendations:

- 1 x 2GHz CPU Thread per spinner
- 2 x GHz CPU Thread per SSD
- 4 x GHz CPU Thread per NVMe

Examples:

- 8 HDDS journaled to SSD – 10 cores / 8 OSDs = 1.25 cores per OSD
- 6 SSDs without journals – 12 cores / 6 OSDs = 2 cores per OSD
- 8 SSDs journaled to NVMe – 20 cores / 8 OSDs = 2.5 cores per OSD

Note that resources are applied cluster-wide to all OSDs. If a cluster contains multiple OSD types, you must use the highest Requests for the whole cluster. For the examples below, a mixture of HDDs journaled to SSD and SSDs without journals would necessitate a *Request* for 2 cores.

8.3 Resource Requests - OSD RAM

There are node hardware recommendations for OSD RAM usage, and this needs to be translated to RAM requests on a per-OSD basis. The node-level recommendation below describes `osd_memory_target`. This is a Ceph configuration that is described in detail further on.

```
Total RAM required = [number of OSDs] x (1 GB + osd_memory_target) + 16 GB
```

Ceph OSDs will attempt to keep heap memory usage under a designated target size set via the `osd_memory_target` configuration option. Ceph's default `osd_memory_target` is 4GB, and we do not recommend decreasing the `osd_memory_target` below 4GB. You may wish to increase this value to improve overall Ceph read performance by allowing the OSDs to use more RAM. While the total amount of heap memory mapped by the process should stay close to this target, there is no guarantee that the kernel will actually reclaim memory that has been unmapped.

For example, a node hosting 8 OSDs, memory *Requests* would be calculated as such:

```
8 OSDs x (1GB + 4GB) + 16GB = 56GB per node
```

Allowing resource usage for each OSD:

```
56GB / 8 OSDs = 7GB
```

Ceph has a feature that allows it to set `osd_memory_target` automatically when a Rook OSD Resource Request is set. However, Ceph sets this value `1:1` and does not leave overhead for waiting for the kernel to free memory. Therefore, we recommend setting `osd_memory_target` in Ceph explicitly, even if you wish to use the default value. Set Rook's OSD resource requests accordingly and to a higher value than `osd_memory_target` by at least an additional 1GB. This is so Kubernetes does not schedule more applications or Ceph daemons onto a node than the node is likely to have RAM available for.

OSD RAM *Resource Requests* come with the same cluster-wide *Resource Requests* note as for OSD CPU. Use the highest *Requests* for a cluster consisting of multiple different configurations of OSDs.

8.4 Resource Requests - Gateways

For gateways, the best recommendation is to always tune your workload and daemon configurations. However, we do recommend the following initial configurations:

RGWs:

- 6-8 CPU cores
- 64 GB RAM (32 GB minimum – may only apply to older "civetweb" protocol)

> **Note**
>
> The numbers below for RGW assume a lot of clients connecting. Thus they might not be the best for your scenario. The RAM usage should be lower for the newer “beast” protocol as opposed to the older “civetweb” protocol.

MDS:

- 2.5 GHz CPU with a least 2 cores
- 3GB RAM

NFS-Ganesha:

- 6-8 CPU cores (untested, high estimate)
- 4GB RAM for default settings (settings hardcoded in Rook presently)

9 Basic Performance Enhancements

The following are some basic performance enhancements. These are a few easy, low-hanging-fruit recommendations.

> **Note**
>
> Not all of these will be right for all clusters or workloads. Always performance test and use your best judgment.

- You can gain performance by using a CNI plugin with an accelerated network stack. For example, Cilium uses eBPF to improve performance over some other CNI plugins.
- Enable host networking to improve network performance. Notably, this provides lower, more stable latency. This does, however, step outside of Kubernetes' network security domain. In `cluster.yaml` set `network:provider:host`.
- Use jumbo frames for your networking. This can be applied to both host networking and the CNI plugin.
- For performance-sensitive deployments, ensure Ceph OSDs always get the performance they need by not allowing other Ceph daemons or user applications to run on OSD nodes. Co-locating MONs and MGRs with OSDs can still be done fairly safely as long as there are enough hardware resources to also include monitors and managers.

10 Legal notice

Documents published as part of the **SUSE Best Practices** series have been contributed voluntarily by SUSE employees and third parties. They are meant to serve as examples of how particular actions can be performed. They have been compiled with utmost attention to detail. However, this does not guarantee complete accuracy. SUSE cannot verify that actions described in these documents do what is claimed or whether actions described have unintended consequences. SUSE LLC, its affiliates, the authors, and the translators may not be held liable for possible errors or the consequences thereof.

Below we draw your attention to the license under which the articles are published.

GNU Free Documentation License

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or non-commercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/ ↗.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

```
Copyright (c) YEAR YOUR NAME.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.2
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
A copy of the license is included in the section entitled "GNU
 Free Documentation License".
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts". line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.