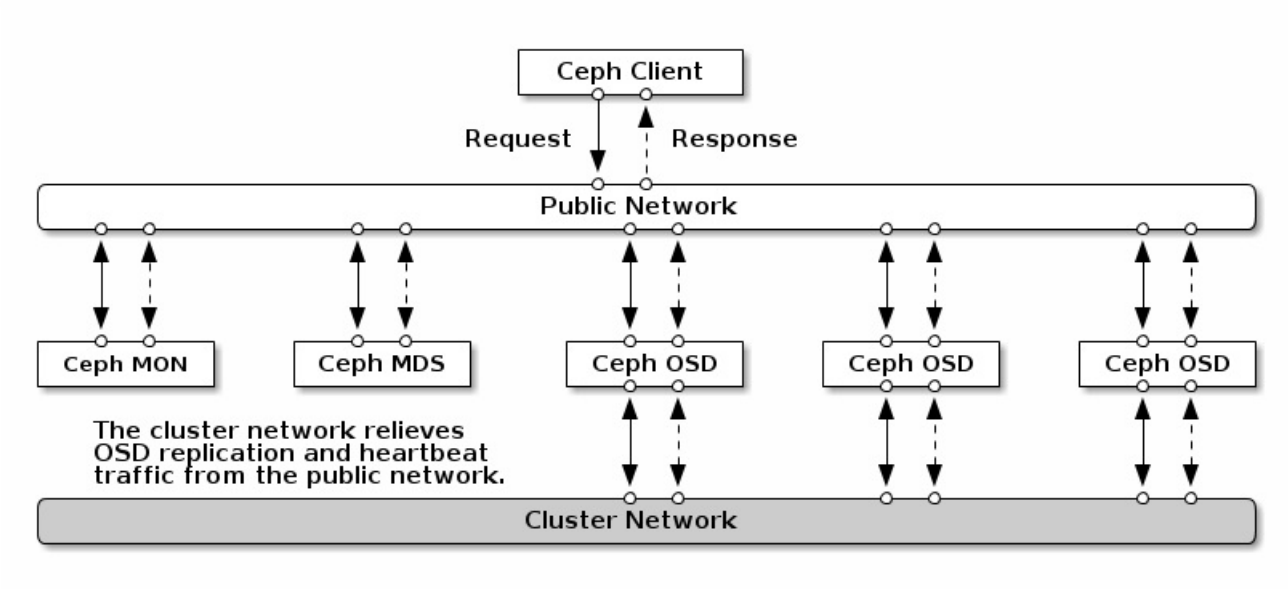
NETWORK CONFIGURATION REFERENCE

Network configuration is critical for building a high performance Ceph Storage Cluster. The Ceph Storage Cluster does not perform request routing or dispatching on behalf of the Ceph Client. Instead, Ceph Clients make requests directly to Ceph OSD Daemons. Ceph OSD Daemons perform data replication on behalf of Ceph Clients, which means replication and other factors impose additional loads on Ceph Storage Cluster networks.

Our Quick Start configurations provide a trivial Ceph configuration file that sets monitor IP addresses and daemon host names only. Unless you specify a cluster network, Ceph assumes a single “public” network. Ceph functions just fine with a public network only, but you may see significant performance improvement with a second “cluster” network in a large cluster.

We recommend running a Ceph Storage Cluster with two networks: a public (front-side) network and a cluster (back-side) network. To support two networks, each Ceph Node will need to have more than one NIC. See Hardware Recommendations - Networks for additional details.



There are several reasons to consider operating two separate networks:

1. **Performance:** Ceph OSD Daemons handle data replication for the Ceph Clients. When Ceph OSD Daemons replicate data more than once, the network load between Ceph OSD Daemons easily dwarfs the network load between Ceph Clients and the Ceph Storage Cluster. This can introduce latency and create a performance problem. Recovery and rebalancing can also introduce significant latency on the public network. See Scalability and High Availability for additional details on how Ceph replicates data. See Monitor / OSD Interaction for details on heartbeat traffic.
2. **Security**: While most people are generally civil, a very tiny segment of the population likes to engage in what’s known as a Denial of Service (DoS) attack. When traffic between Ceph OSD Daemons gets disrupted, placement groups may no longer reflect an `active + clean` state, which may prevent users from reading and writing data. A great way to defeat this type of attack is to maintain a completely separate cluster network that doesn’t connect directly to the internet. Also, consider using Message Signatures to defeat spoofing attacks.

IP TABLES

By default, daemons bind to ports within the `6800:7300` range. You may configure this range at your discretion. Before configuring your IP tables, check the default `iptables` configuration.

```
sudo iptables -L
```

Some Linux distributions include rules that reject all inbound requests except SSH from all network interfaces. For example:

```
REJECT all -- anywhere anywhere reject-with icmp-host-prohibited
```

You will need to delete these rules on both your public and cluster networks initially, and replace them with appropriate rules when you are ready to harden the ports on your Ceph Nodes.

MONITOR IP TABLES

Ceph Monitors listen on port 6789 by default. Additionally, Ceph Monitors always operate on the public network. When you add the rule using the example below, make sure you replace `{iface}` with the public network interface (e.g., `eth0`, `eth1`, etc.), `{ip-address}` with the IP address of the public network and `{netmask}` with the netmask for the public network.

```
sudo iptables -A INPUT -i {iface} -p tcp -s {ip-address}/{netmask} --dport 6789 -j ACCEPT
```

MDS AND MANAGER IP TABLES

A Ceph Metadata Server or Ceph Manager listens on the first available port on the public network beginning at port 6800. Note that this behavior is not deterministic, so if you are running more than one OSD or MDS on the same host, or if you restart the daemons within a short window of time, the daemons will bind to higher ports. You should open the entire 6800-7300 range by default. When you add the rule using the example below, make sure you replace `{iface}` with the public network interface (e.g., `eth0`, `eth1`, etc.), `{ip-address}` with the IP address of the public network and `{netmask}` with the netmask of the public network.

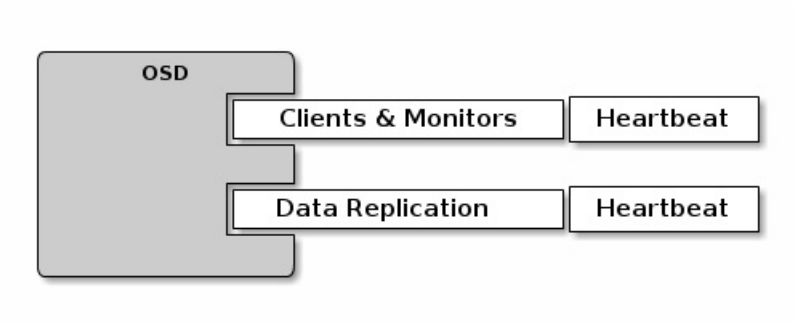
For example:

```
sudo iptables -A INPUT -i {iface} -m multiport -p tcp -s {ip-address}/{netmask} --dports 6800
```

OSD IP TABLES

By default, Ceph OSD Daemons bind to the first available ports on a Ceph Node beginning at port 6800. Note that this behavior is not deterministic, so if you are running more than one OSD or MDS on the same host, or if you restart the daemons within a short window of time, the daemons will bind to higher ports. Each Ceph OSD Daemon on a Ceph Node may use up to four ports:

1. One for talking to clients and monitors.
2. One for sending data to other OSDs.
3. Two for heartbeating on each interface.



When a daemon fails and restarts without letting go of the port, the restarted daemon will bind to a new port. You should open the entire 6800-7300 port range to handle this possibility.

If you set up separate public and cluster networks, you must add rules for both the public network and the cluster network, because clients will connect using the public network and other Ceph OSD Daemons will connect using the cluster network. When you add the rule using the example below, make sure you replace `{iface}` with the network interface (e.g., `eth0`, `eth1`, etc.), `{ip-address}` with the IP address and `{netmask}` with the netmask of the public or cluster network. For example:

```
sudo iptables -A INPUT -i {iface}  -m multiport -p tcp -s {ip-address}/{netmask} --dports 680
```

Tip: If you run Ceph Metadata Servers on the same Ceph Node as the Ceph OSD Daemons, you can consolidate the public network configuration step.

CEPH NETWORKS

To configure Ceph networks, you must add a network configuration to the `[global]` section of the configuration file. Our 5-minute Quick Start provides a trivial Ceph configuration file that assumes one public network with client and server on the same network and subnet. Ceph functions just fine with a public network only. However, Ceph allows you to establish much

more specific criteria, including multiple IP network and subnet masks for your public network. You can also establish a separate cluster network to handle OSD heartbeat, object replication and recovery traffic. Don't confuse the IP addresses you set in your configuration with the public-facing IP addresses network clients may use to access your service. Typical internal IP networks are often `192.168.0.0` or `10.0.0.0`.

Tip: If you specify more than one IP address and subnet mask for either the public or the cluster network, the subnets within the network must be capable of routing to each other. Additionally, make sure you include each IP address/subnet in your IP tables and open ports for them as necessary.

Note: Ceph uses CIDR notation for subnets (e.g., `10.0.0.0/24`).

When you have configured your networks, you may restart your cluster or restart each daemon. Ceph daemons bind dynamically, so you do not have to restart the entire cluster at once if you change your network configuration.

PUBLIC NETWORK

To configure a public network, add the following option to the `[global]` section of your Ceph configuration file.

```
[global]
        # ... elided configuration
        public network = {public-network/netmask}
```

CLUSTER NETWORK

If you declare a cluster network, OSDs will route heartbeat, object replication and recovery traffic over the cluster network. This may improve performance compared to using a single network. To configure a cluster network, add the following option to the `[global]` section of your Ceph configuration file.

```
[global]
        # ... elided configuration
        cluster network = {cluster-network/netmask}
```

We prefer that the cluster network is **NOT** reachable from the public network or the Internet for added security.

CEPH DAEMONS

Ceph has one network configuration requirement that applies to all daemons: the Ceph configuration file **MUST** specify the `host` for each daemon. Ceph also requires that a Ceph configuration file specify the monitor IP address and its port.

Important: Some deployment tools (e.g., `ceph-deploy`, Chef) may create a configuration file for you. **DO NOT** set these values if the deployment tool does it for you.

Tip: The `host` setting is the short name of the host (i.e., not an fqdn). It is **NOT** an IP address either. Enter `hostname -s` on the command line to retrieve the name of the host.

```
[mon.a]

        host = {hostname}
        mon addr = {ip-address}:6789

[osd.0]
        host = {hostname}
```

You do not have to set the host IP address for a daemon. If you have a static IP configuration and both public and cluster networks running, the Ceph configuration file may specify the IP address of the host for each daemon. To set a static IP address for a daemon, the following option(s) should appear in the daemon instance sections of your `ceph.conf` file.

```
[osd.0]
        public addr = {host-public-ip-address}
```

```
        cluster addr = {host-cluster-ip-address}
```

One NIC OSD in a Two Network Cluster

Generally, we do not recommend deploying an OSD host with a single NIC in a cluster with two networks. However, you may accomplish this by forcing the OSD host to operate on the public network by adding a `public addr` entry to the `[osd.n]` section of the Ceph configuration file, where n refers to the number of the OSD with one NIC. Additionally, the public network and cluster network must be able to route traffic to each other, which we don't recommend for security reasons.

NETWORK CONFIG SETTINGS

Network configuration settings are not required. Ceph assumes a public network with all hosts operating on it unless you specifically configure a cluster network.

PUBLIC NETWORK

The public network configuration allows you specifically define IP addresses and subnets for the public network. You may specifically assign static IP addresses or override `public network` settings using the `public addr` setting for a specific daemon.

`public network`

Description: The IP address and netmask of the public (front-side) network (e.g., `192.168.0.0/24`). Set in `[global]`. You may specify comma-delimited subnets.
Type: `{ip-address}/{netmask} [, {ip-address}/{netmask}]`
Required: No
Default: N/A

`public addr`

Description: The IP address for the public (front-side) network. Set for each daemon.
Type: IP Address
Required: No
Default: N/A

CLUSTER NETWORK

The cluster network configuration allows you to declare a cluster network, and specifically define IP addresses and subnets for the cluster network. You may specifically assign static IP addresses or override `cluster network` settings using the `cluster addr` setting for specific OSD daemons.

`cluster network`

Description: The IP address and netmask of the cluster (back-side) network (e.g., `10.0.0.0/24`). Set in `[global]`. You may specify comma-delimited subnets.
Type: `{ip-address}/{netmask} [, {ip-address}/{netmask}]`
Required: No
Default: N/A

`cluster addr`

Description: The IP address for the cluster (back-side) network. Set for each daemon.
Type: Address
Required: No

Default: N/A

BIND

Bind settings set the default port ranges Ceph OSD and MDS daemons use. The default range is `6800:7300`. Ensure that your IP Tables configuration allows you to use the configured port range.

You may also enable Ceph daemons to bind to IPv6 addresses instead of IPv4 addresses.

`ms bind port min`

Description: The minimum port number to which an OSD or MDS daemon will bind.
Type: 32-bit Integer
Default: `6800`
Required: No

`ms bind port max`

Description: The maximum port number to which an OSD or MDS daemon will bind.
Type: 32-bit Integer
Default: `7300`
Required: No.

`ms bind ipv6`

Description: Enables Ceph daemons to bind to IPv6 addresses. Currently the messenger *either* uses IPv4 or IPv6, but it cannot do both.
Type: Boolean
Default: `false`
Required: No

`public bind addr`

Description: In some dynamic deployments the Ceph MON daemon might bind to an IP address locally that is different from the `public addr` advertised to other peers in the network. The environment must ensure that routing rules are set correclty. If `public bind addr` is set the Ceph MON daemon will bind to it locally and use `public addr` in the monmaps to advertise its address to peers. This behavior is limited to the MON daemon.
Type: IP Address
Required: No
Default: N/A

HOSTS

Ceph expects at least one monitor declared in the Ceph configuration file, with a `mon addr` setting under each declared monitor. Ceph expects a `host` setting under each declared monitor, metadata server and OSD in the Ceph configuration file. Optionally, a monitor can be assigned with a priority, and the clients will always connect to the monitor with lower value of priority if specified.

`mon addr`

Description: A list of `{hostname}:{port}` entries that clients can use to connect to a Ceph monitor. If not set, Ceph searches `[mon.*]` sections.
Type: String
Required: No
Default: N/A

`mon priority`

Description: The priority of the declared monitor, the lower value the more prefered when a client selects a monitor when trying to connect to the cluster.
Type: Unsigned 16-bit Integer
Required: No
Default: 0

`host`

Description: The hostname. Use this setting for specific daemon instances (e.g., `[osd.0]`).
Type: String
Required: Yes, for daemon instances.
Default: `localhost`

Tip: Do not use `localhost`. To get your host name, execute `hostname -s` on your command line and use the name of your host (to the first period, not the fully-qualified domain name).

Important: You should not specify any value for `host` when using a third party deployment system that retrieves the host name for you.

TCP

Ceph disables TCP buffering by default.

`ms tcp nodelay`

- **Description:** Ceph enables `ms tcp nodelay` so that each request is sent immediately (no buffering). Disabling Nagle's algorithm increases network traffic, which can introduce latency. If you experience large numbers of small packets, you may try disabling `ms tcp nodelay`.
- **Type:** Boolean
- **Required:** No
- **Default:** `true`

`ms tcp rcvbuf`

- **Description:** The size of the socket buffer on the receiving end of a network connection. Disable by default.
- **Type:** 32-bit Integer
- **Required:** No
- **Default:** `0`

`ms tcp read timeout`

- **Description:** If a client or daemon makes a request to another Ceph daemon and does not drop an unused connection, the `ms tcp read timeout` defines the connection as idle after the specified number of seconds.
- **Type:** Unsigned 64-bit Integer
- **Required:** No
- **Default:** `900` 15 minutes.