```
[ceph: root@node /]# ceph orch apply osd --all-available-devices --unmanaged=true
Scheduled osd.all-available-devices update...
```

> **Note**
> You can also update the `unmanaged` flag with a service specification file.

Specific Target Provisioning

You can create OSD daemons by using a specific device and host. To create a single OSD daemon with a specific host and storage device, use the `ceph orch daemon add` command.

```
[ceph: root@node /]# ceph orch daemon add osd node:/dev/vdb
Created osd(s) 12 on host 'node'
```

To stop an OSD daemon, use the `ceph orch daemon stop` command with the OSD ID.

```
[ceph: root@node /]# ceph orch daemon stop osd.12
```

To remove an OSD daemon, use the `ceph orch daemon rm` command with the OSD ID.

```
[ceph: root@node /]# ceph orch daemon rm osd.12
Removed osd.12 from host 'node'
```

To release an OSD ID, use the `ceph osd rm` command.

```
[ceph: root@node /]# ceph osd rm 12
removed osd.12
```

Service Specification Provisioning

Use service specification files to describe the cluster layout for OSD services. You can customize the service provisioning with filters. With filters, you can configure the OSD service without knowing the specific hardware architecture. This method is useful when automating cluster bootstrap and maintenance windows.

The following is an example service specification YAML file that defines two OSD services, each using different filters for placement and BlueStore device location.

```
service_type: osd
service_id: osd_size_and_model
placement:
  host_pattern: '*'
data_devices:
  size: '100G:'
db_devices:
  model: My-Disk
wal_devices:
  size: '10G:20G'
unmanaged: true
```