OpenStack Cinder as an SDS API
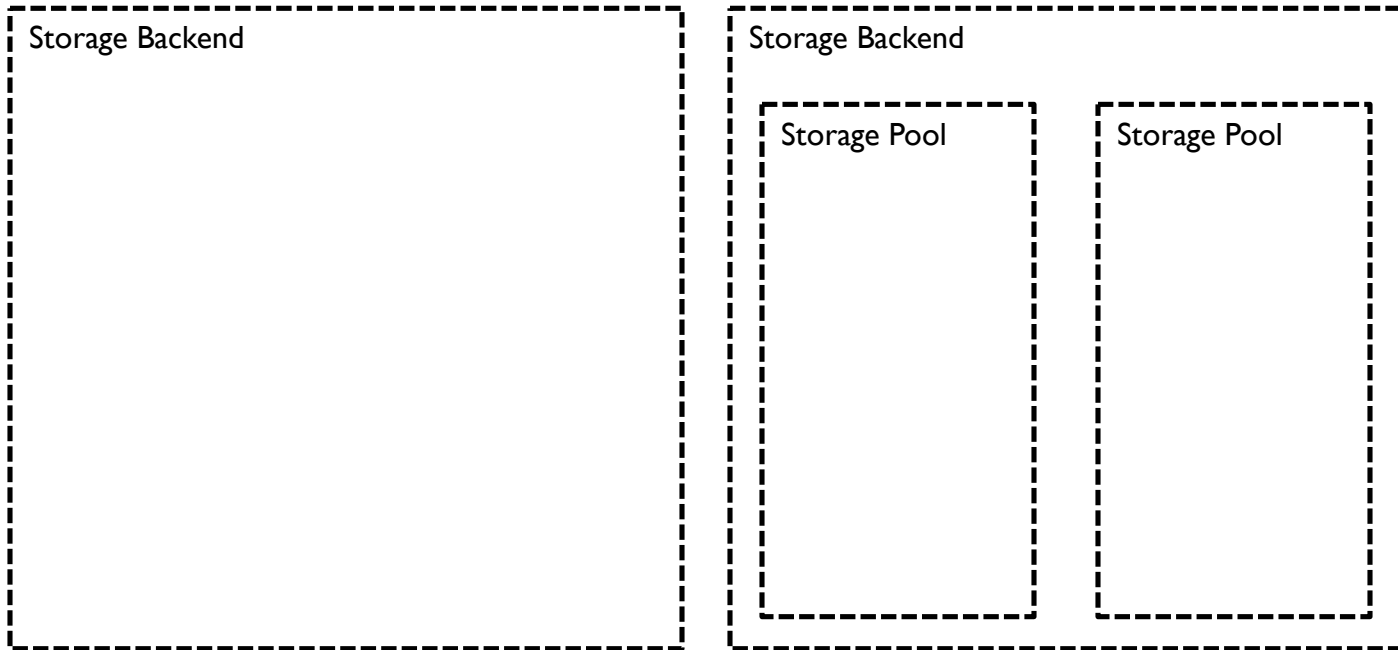
Sean McGinnis
Huawei

Cinder in OpenStack
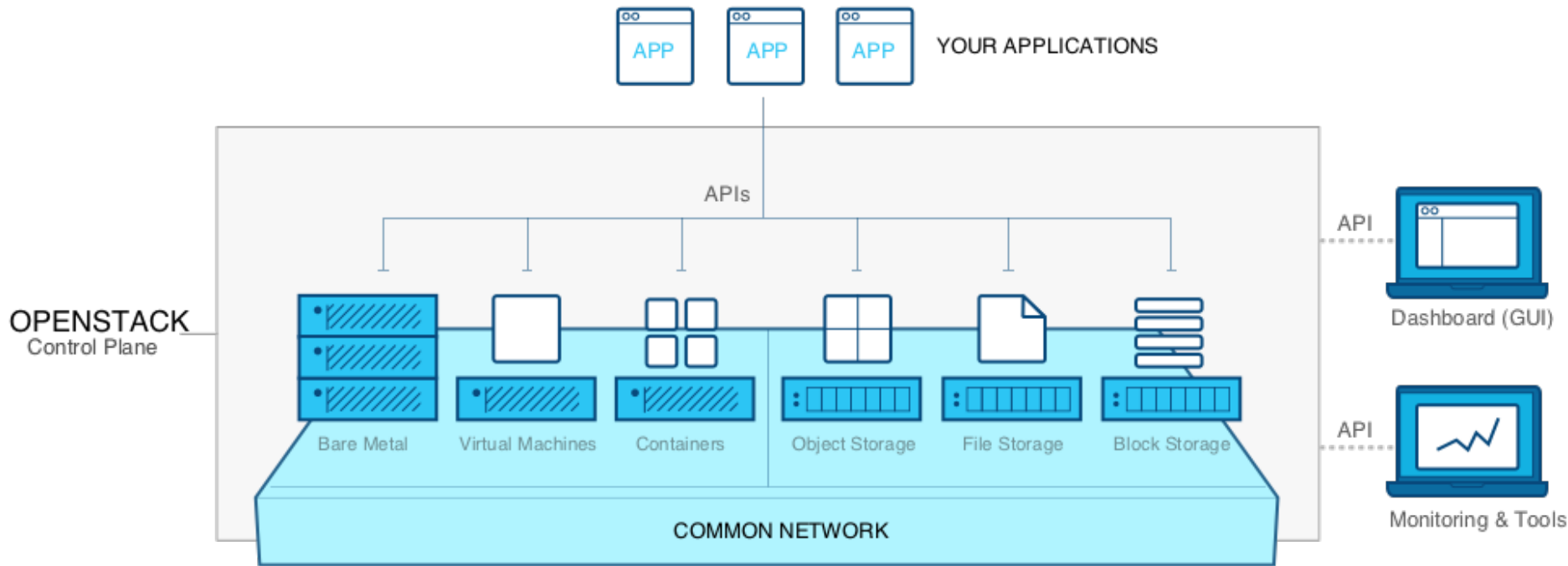
What is Cinder

☐ Block storage management API

☐ Abstraction layer of many types of storage

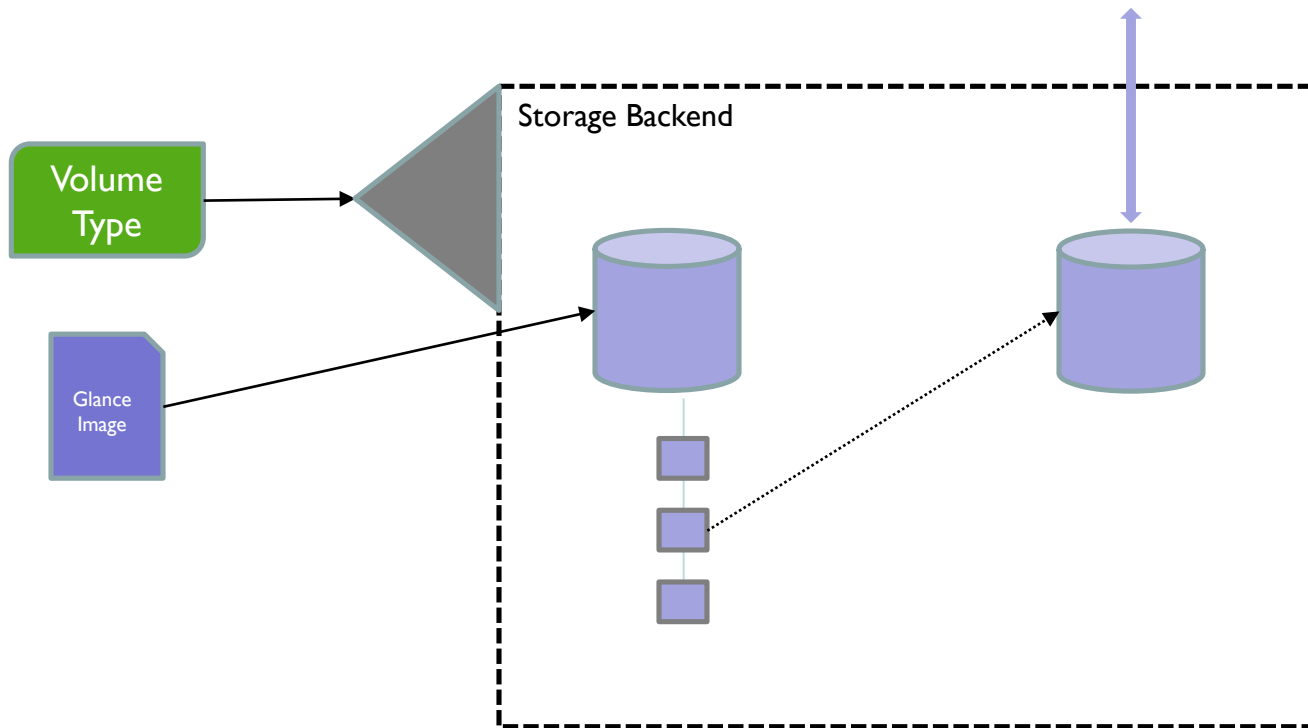# What Cinder is Not

- ❑ Not a storage provider
- ❑ Not in the IO path
- ❑ Not a general purpose storage manager

Cinder Model


```
┌ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ┐     ┌ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ┐
  Storage Backend                   Storage Backend
                                     ┌ ─ ─ ─ ─ ─ ─ ┐ ┌ ─ ─ ─ ─ ─ ─ ┐
                                       Storage Pool    Storage Pool
└ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ┘     └ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ┘
```

Cinder Model

Volume Type

Glance Image

Storage Backend
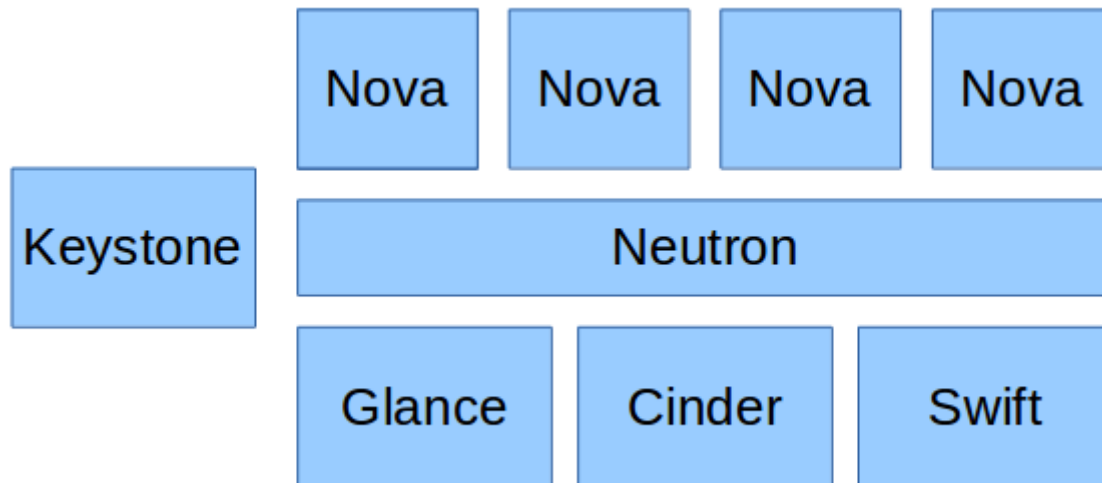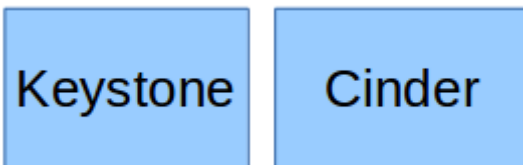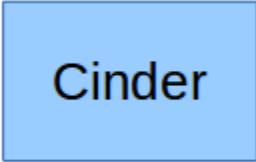
Cinder as Part of OpenStack

Cinder Standalone

Cinder Standalone - noauth

Cinder

`auth_strategy = noauth`

Flexible Deployment

- ☐ Bare metal or VM
- ☐ Containers
- ☐ Manually with system packages
- ☐ Deployment tools
  - ☐ Ansible
  - ☐ Puppet
  - ☐ Helm

Cinder References

https://docs.openstack.org/cinder/latest/install/

https://www.youtube.com/watch?v=YmeGEBVuSNc

https://thenewstack.io/deploying-cinder-stand-alone-storage-service/

https://github.com/openstack/cinder/blob/master/contrib/block-box/README.md

https://gorka.eguileor.com/standalone-cinder/

Interface – Horizon

Interface – Command Line

smcginnis@tower:~$

Interface – REST API

https://developer.openstack.org/api-ref/block-storage/

Typical Usage

- Point and click management via Horizon

- Interactive CLI – OpenStackClient or CinderClient

- Programmatic using REST API or SDK

    - OpenStackSDK

    - Gophercloud

    - Etc…

- Integration with config management tools

Scenario #1

- Production data remains untouched
- Need to run local tests against real data
- Create volume from snapshot of data

Scenario #1 - Setup

- ❑ Services used
  - ❑ Cinder, Keystone (optional)
- ❑ Platform
  - ❑ Windows, SQL Server, PowerShell
- ❑ Example
  - ❑ Direct REST API usage

Scenario #1 - Script

Full script:

https://github.com/stmcginnis/presentations/blob/master/Vancouver2018/CinderSDS/assets/powershell.ps1

- ❑ Useful cmdlets
  - ❑ Invoke-WebRequest
  - ❑ Invoke-RestMethod
  - ❑ New-IscsiTargetPortal
  - ❑ Connect-IscsiTarget

Scenario #1 - Authentication

```powershell
$result = Invoke-WebRequest -Headers $headers -Method Post -Body @"
{
 "auth": { "identity": {
    "methods": ["password"],
        "password":
            { "user": { "name": "$($creds.UserName)",
                        "domain": {"id": "default"},
                        "password": "$($creds.GetNetworkCredential().password)"}}},
    "scope": { "project": { "domain": {"name": "Default"}, "name": "admin"}}}}
"@ -Uri http://192.168.1.230:5000/v3/auth/tokens

$os_token = $result.Headers['X-Subject-Token']
$headers.Add('X-Auth-Token', $os_token)
```

Scenario #1 – Find Volume

```
$result = Invoke-RestMethod -Headers $headers -Uri `
    "http://192.168.1.230:8776/v3/$project_id/volumes"

foreach($vol in $result.volumes) {
 if ($vol.name -eq "Demo-Vol") {
     $src_vol = $vol.id
     break
 }
}
```

Scenario #1 – Create Snapshot

```
$result = Invoke-RestMethod -Headers $headers -Method Post -Body @"
{
 "snapshot": {
    "name": "demo-snap",
    "description": "Demo snapshot for testing",
    "volume_id": "$src_vol",
    "force": "True"
 }
}"@ -Uri "http://192.168.1.230:8776/v3/$project_id/snapshots"

$snapshot = $result.snapshot
```

Scenario #1 – Create Volume

```powershell
$result = Invoke-RestMethod -Headers $headers -Method Post -Body @"
{
 "volume": {
    "snapshot_id": "$($snapshot.id)"
 }
}"@ -Uri "http://192.168.1.230:8776/v3/$project_id/volume"

$volume = $result.volumes
```

Senario #1 – Use Volume

```
foreach ($target in $connection_info.target_iqns) {
 $connections += Get-IscsiTarget -NodeAddress $target |
   Connect-IscsiTarget -AuthenticationType ONEWAYCHAP `
     -ChapSecret $connection_info.auth_password `
     -ChapUsername $connection_info.auth_username
}

foreach ($connection in $connections) {
 $disk = Get-Disk –iSCSISession $connection
 Set-Disk -Number $disk.Number -IsOffline $False
 Set-Disk -Number $disk.Number -IsReadonly $False
}

$volume = Get-Volume -FileSystemLabel "SQLData"
$mdf = "$($volume.DriveLetter)\Data\MyDB.mdf"
$ldf = "$($volume.DriveLetter)\Data\MyDB.ldf"
Invoke-Sqlcmd @"USE [master]CREATE DATABASE [MyDB] ON (FILENAME = '$mdf'),
(FILENAME = '$ldf') FOR ATTACH"@ -QueryTimeout 3600 -ServerInstance 'Local-
Instance'
```

Scenario #1 - References

https://docs.microsoft.com/powershell/module/microsoft.powershell.utility/invoke-restmethod

https://developer.openstack.org/api-ref/block-storage/

Scenario #2

- ☐ Managing system configurations
- ☐ Need to include standard storage configuration
- ☐ Ansible playbook configures volumes through Cinder

Scenario #2 – Setup

- Services used
  - Cinder, Keystone
- Platform
  - Any Ansible supported environment
- Example
  - Configuration management using Ansible

Scenario #2 – Cinder task

```
- name: create data volume
  hosts: localhost
  tasks:
  - name: Create 100g volume
    os_volume:
      state: present
      cloud: test-lab
      size: 100
      display_name: {{ ansible_hostname }} Data Vol
```

Scenario #2 – clouds.yaml

```yaml
clouds:
 test-lab:
   region_name: test_lab
   auth:
     username: 'autovonbot'
     password: password1
     project_name: 'IT'
     auth_url: 'http://192.168.1.230/identity/'
```

Scenario #2 - Sidebar

```python
connection = openstack.connect(cloud='test-lab')

volume = connection.create_volume(
 10, name='New Boot Vol', image=image_id, wait=True)
```

Scenario #2 – Cinder task

```yaml
- name: create data volume
  hosts: localhost
  tasks:
  - name: Create 100g volume
    os_volume:
      state: present
      cloud: test-lab
      size: 100
      display_name: {{ ansible_hostname }} Data Vol
```

Scenario #2 – Storage Configuration

```
- parted:
    device: /dev/sdb
    number: 1
    state: present
- filesystem:
    fstype: ext4
    dev: /dev/sdb1
- mount:
    path: /data
    src: UUID=c3f48145-fa3b-4bbe-a700-a2a179ec9077
    fstype: ext4
    state: present
```

Scenario #2 - References

http://docs.ansible.com/ansible/latest/modules/os_volume_module.html

https://www.ansible.com/

https://docs.openstack.org/openstacksdk/

Scenario #3

□ Local script scheduled to run periodically

□ Check for volume space consumption

□ If low, automatically extend volume

Scenario #3 - Setup

❑ Services used
  ❑ Cinder, Keystone
❑ Platform
  ❑ Baremetal or VM
❑ Example
  ❑ Scripted CLI usage

Scenario #3 - Script

```bash
stats=`df -H /SanVol | tail -1`

size=`echo $stats | awk '{print $2}' | sed -e "s/G//"`
used=`echo $stats | awk '{print $5}' | sed -e "s/\%//"`

if [ $used -gt 90 ]; then
    # Need to extend the volume
    openstack volume extend MyDataVol $(($size + 10))

    # Resize the local volume
    diskutil cs resizeStack \
        d3eaf95e-e2e0-4410-9c96-6f093f91407a $(($size +10))
fi
```

Scenario #3 - Script

```
stats=`df -H /SanVol | tail -1`

size=`echo $stats | awk '{print $2}' | sed -e "s/G//"`
used=`echo $stats | awk '{print $5}' | sed -e "s/\%//"`

if [ $used -gt 90 ]; then
    # Need to extend the volume
    openstack volume extend MyDataVol $(($size + 10))

    # Resize the local volume
    diskutil cs resizeStack \
        d3eaf95e-e2e0-4410-9c96-6f093f91407a $(($size +10))
fi
```

Scenario #3 - Script

```
stats=`df -H /SanVol | tail -1`

size=`echo $stats | awk '{print $2}' | sed -e "s/G//"`
used=`echo $stats | awk '{print $5}' | sed -e "s/\%//"`

if [ $used -gt 90 ]; then
    # Need to extend the volume
    openstack volume extend MyDataVol $(($size + 10))

    # Resize the local volume
    diskutil cs resizeStack \
        d3eaf95e-e2e0-4410-9c96-6f093f91407a $(($size +10))
fi
```

SDC 18

Scenario #3 - Script

```
stats=`df -H /SanVol | tail -1`

size=`echo $stats | awk '{print $2}' | sed -e "s/G//"`
used=`echo $stats | awk '{print $5}' | sed -e "s/\%//"`

if [ $used -gt 90 ]; then
    # Need to extend the volume
    openstack volume extend MyDataVol $(($size + 10))

    # Resize the local volume
    diskutil cs resizeStack \
        d3eaf95e-e2e0-4410-9c96-6f093f91407a $(($size +10))
fi
```

Scenario #3 – References

https://docs.openstack.org/python-openstackclient/

Thank You

Please share your experiences and use cases!
@SeanTMcGinnis