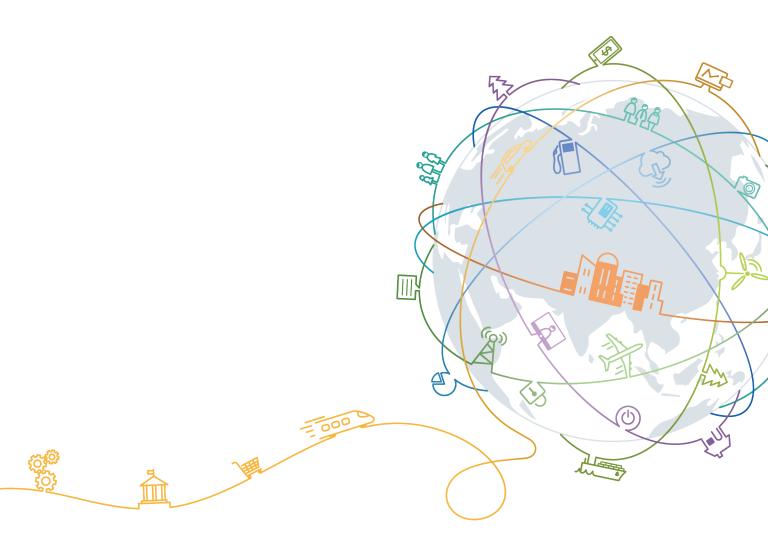
OpenStack Manila Driver

配置指南

文档版本 01

发布日期 2019-09-26





版权所有 © 华为技术有限公司 2019。 保留一切权利。

非经本公司书面许可,任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部,并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。 本文档提及的其他所有商标或注册商标,由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束,本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定,华为公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因,本文档内容会不定期进行更新。除非另有约定,本文档仅作为使用指导,本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

华为技术有限公司

地址: 深圳市龙岗区坂田华为总部办公楼 邮编: 518129

网址:http://www.huawei.com客户服务邮箱:support@huawei.com

客户服务电话: 4008302118

目录

1 简介	1
2 版本配套关系	2
3 Manila Driver 安装与部署	4
3.1 Manila Driver 获取	
3.2 Ubuntu 环境部署	5
3.3 RedHat 环境部署	5
4 Manila Driver 基本属性配置	
5 Manila Driver 高级属性配置	11
5.1 配置 Thin/Thick 属性	11
5.2 配置 SmartDedupe 属性	12
5.3 配置 SmartCompression 属性	12
5.4 配置 SmartCache 属性	13
5.5 配置 SmartPartition 属性	13
5.6 配置 SmartQoS 属性	14
5.6.1 配置控制最大 IOPS	14
5.6.2 配置控制最小 IOPS	14
5.6.3 配置控制最大带宽	
5.6.4 配置控制最小带宽	
5.6.5 配置控制时延	16
5.6.6 配置控制多策略	16
5.7 指定 Storage Pool 创建共享点	17
5.8 指定 Disk Type 创建共享点	17
5.9 配置 SectorSize	
5.10 配置 Replication	18
5.11 配置 FS 归属控制器	19
5.12 配置 NFS 客户端权限	20
5.13 配置 HyperMetro 属性	
6 最佳实践	23
61快速对接华为存储	23

 $oldsymbol{1}$ $_{lpha _{ } }$

华为Manila Driver是部署在OpenStack Manila模块上的一个插件,用于对接华为存储向OpenStack平台提供共享存储能力。

2版本配套关系

介绍华为Manila Driver版本与华为存储、OpenStack的版本配套关系。

表 2-1 Manila Driver 存储产品版本支持说明

OpenStack版本	存储产品版本
Newton	OceanStor V3 V300R003/V300R006
Ocata	OceanStor V5 V500R007
Pike	OceanStor 2600 V3 V300R005
Queens	OceanStor 18500/18800 V300R003/V300R006/V500R007
Rocky	OceanStor Dorado V3 V300R002C10
Stein	

表 2-2 Manila Driver 特性支持说明(↓:支持, x:不支持)

特性	Newto n	Ocata	Pike	Queens	Rocky	Stein
Create Share	√	√	√	√	√	√
Delete Share	√	√	√	√	√	√
Allow access	√	√	√	√	√	√
Deny access	√	√	√	√	√	√
Create Snapshot	√	√	√	√	√	√
Delete Snapshot	√	√	√	√	√	√
Revert Snapshot	X	X	X	√	√	√
Manage/Unmanage Share	√	√	√	✓	✓	√
Extend Share	√	√	√	√	√	√

特性	Newto n	Ocata	Pike	Queens	Rocky	Stein
Shrink Share	√	√	√	√	√	√
SmartCompression	√	√	√	√	√	√
SmartDedupe	√	√	√	√	√	√
SmartCache	√	√	√	√	√	√
SmartThin/Thick	√	√	√	√	√	√
SmartPartition	√	√	√	√	√	√
SmartQoS	√	√	√	√	√	√
Multi-tenancy	√	√	√	√	√	√
Ensure Share	√	√	√	√	√	√
Create Share from Snapshot	√	√	√	√	√	✓
Manage/Unmanage Snapshot	х	√	√	√	√	√
Create a share on a certain storage pool	х	√	1	√	√	√
Create a share with a certain disk type	X	√	√	√	1	√
SectorSize	X	√	√	√	√	√
Replication	X	√	√	√	√	√

3 Manila Driver 安装与部署

- 3.1 Manila Driver获取
- 3.2 Ubuntu环境部署
- 3.3 RedHat环境部署

3.1 Manila Driver 获取

您可以通过两种途径获取到OpenStack Driver:

第一种是通过OpenStack社区仓库。从Kilo版本开始,华为存储就已经将华为存储驱动完全贡献给OpenStack开源社区,用户可自由下载贡献到OpenStack社区的OpenStack Driver版本。安装完成指定的OpenStack版本后,指定的OpenStack Driver放置在.../ manila/manila/share/drivers/huawei目录下。安装完成后未发现对应的安装文件,您也可以通过OpenStack官网仓库下载指定的OpenStack Driver使用,下载地址为: https://github.com/openstack/manila。

第二种是通过华为自有OpenStack Driver仓库。仓库地址为: https://github.com/Huawei/OpenStack_Driver,访问到该仓库地址后,您可以下载到和OpenStack 社区版本相对应的OpenStack Driver。

Manila Driver获取方法具体操作如下:

□说明

OpenStack社区Kilo版本后已自带华为存储OpenStack Driver,则后续步骤可省略,直接进入 Manila Driver属性配置章节。

步骤1 打开网页浏览器,键入上述仓库地址,如华为自有仓库地址: https://github.com/huaweistorage/OpenStack_Driver。

步骤2 点击 "Download ZIP"按钮,驱动将以压缩包的方式下载到本地,对该压缩包解压。

步骤3 在解压完成后的目录下找到Manila目录,其下包含多个OpenStack版本的 OpenStack Driver插件,选择指定OpenStack插件即可。

3.2 Ubuntu 环境部署

OpenStack社区标准部署步骤如下:

步骤1 安装前需删除默认华为OpenStack Driver安装目录下所有文件,默认安装路径为/usr/lib/python2.7/dist-packages/manila/share/drivers/huawei。

⚠注意

python2.7为本机使用的Python版本,如果为其余版本,该处为相应Python版本号。

步骤2 将获取到的OpenStack Manila Driver拷贝到Manila节点驱动安装目录,默认路径参考步骤1。

步骤3 按照章节4和章节5进行配置。

步骤4 待配置完毕后,重启Manila-Share服务,启动命令为:

service manila-share restart

步骤5 使用manila service-list查看服务启动详情,查看Manila-Share服务的State状态为up状态表明服务已经启动正常。

化	* -				
root@u1404:~# manila serv	ice-list				
++	 	+	 	 	
+ T1 D:	l 11 .				
Id Binary	Host	Zone	Status	State	Updated_at
++	 	+	·	·	
+					
1 manila-scheduler	u1404	nova	enabled	up	2016-03-15T01:43:48.000000
	1404	1	111		0016 00 15701 40 50 00000
2 manila-data	u1404	nova	enabled	up	2016-03-15T01:43:50.000000
3 manila-share	u1404@v3r3	nova	enabled	up	2016-03-15T01:43:41.000000
++	+	+	+	 	

----结束

3.3 RedHat 环境部署

OpenStack社区标准部署步骤如下:

步骤1 安装前需删除默认华为OpenStack Driver安装目录下所有文件,默认安装路径为/usr/lib/python2.7/site-packages/manila/share/drivers/huawei。

⚠注意

python2.7为本机使用的Python版本,如果为其余版本,该处为相应Python版本号。

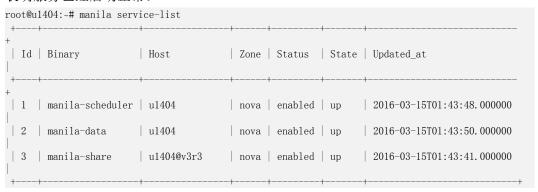
步骤2 将获取到的OpenStack Manila Driver拷贝到Manila节点驱动安装目录,默认路径参考步骤1。

步骤3 按照章节4和章节5进行配置。

步骤4 待配置完毕后,重启Manila-Share服务,启动命令为:

systemctl restart openstack-manila-volume.service

步骤5 使用manila service-list查看服务启动详情,查看Manila-Share服务的State状态为up状态表明服务已经启动正常。



4 Manila Driver 基本属性配置

介绍如何配置华为Manila Driver。

注意

- OpenStack Ocata Manila创建共享点时,如未指定share type,会使用"/etc/manila/manila.conf"文件中默认share type。
- Driver配置使用的存储池,需要事先保证在华为存储上存在,否则请手动创建,并 且存储池类型必须是"文件存储服务"。

[DEFAULT]

root@ubuntu-001:~# manila type-key default_share_type set create share from snapshot support=True

操作步骤

步骤1 修改Manila配置文件"manila.conf"。增加"share_driver","share_backend_name" 和"manila_huawei_conf_file",修改"driver_handles_share_servers"。

以Driver配置文件名称"manila huawei conf.xml"为例。

● 配置一个存储系统,举例如下:

```
[DEFAULT]
enabled_share_backends = huawei_manila_backend
[huawei_manila_backend]
share_driver = manila.share.drivers.huawei.huawei_nas.HuaweiNasDriver
manila_huawei_conf_file = /etc/manila/manila_huawei_conf.xml
share_backend_name = huawei_manila_backend
driver_handles_share_servers = False
```

● 配置多个存储系统,举例如下:

```
[DEFAULT]
enabled_share_backends = huawei_manila_1, huawei_manila_2
[huawei_manila_1]
share_driver = manila.share.drivers.huawei.huawei_nas.HuaweiNasDriver
manila_huawei_conf_file = /etc/manila/manila_huawei_conf_1.xml
share_backend_name = huawei_manila_1
driver_handles_share_servers = False
[huawei_manila_2]
share_driver = manila.share.drivers.huawei.huawei_nas.HuaweiNasDriver
manila_huawei_conf_file = /etc/manila/manila_huawei_conf_2.xml
share_backend_name = huawei_manila_2
driver_handles_share_servers = False
```

□说明

"driver_handles_share_servers"的参数值可为"True"或者"False"。"True"表示支持多租户; "False"表示不支持多租户。

步骤2 在 "/etc/manila" 路径下创建Driver配置文件(例如manila_huawei_conf.xml)。Driver 配置文件的具体信息如下所示:

```
<?xml version='1.0' encoding='UTF-8'?>
 <Config>
    <Storage>
     <Product>V3</Product>
     <LogicalPortIP>x.x.x.x</LogicalPortIP>
     <Port>abc;CTEO. A. H1</Port>
     <UserName>xxx</UserName>
     <UserPassword>xxx</UserPassword>
     <SnapshotSupport>True</SnapshotSupport>
     <ReplicationSupport>False/ReplicationSupport>
    </Storage>
    <Filesystem>
     \label{eq:constraint} $$ \StoragePool>xxx</StoragePool> $$
     <SectorSize>64</SectorSize>
     <WaitInterval>3</WaitInterval>
     <Timeout>60</Timeout>
     <SnapshotReserve>20</SnapshotReserve>
     <NFSClient>
       \langle IP \rangle x. x. x. x \langle /IP \rangle
     </NFSClient>
     <CIFSClient>
       <UserName>xxx</UserName>
       <UserPassword>xxx</UserPassword>
     </CIFSClient>
     </Filesystem>
```

在配置文件中,各属性描述如表4-1所示。

表 4-1 配置文件属性说明

属性名称	默认值	说明	类型
Product	V3	存储产品类型。	必选
LogicalPortIP	-	逻辑端口IP。支持配置多个IP,以分号;分隔。	非多租户模式下必 选。
Port	-	可用的业务端口(绑定端口或者物理端口)。	多租户模式下可选,如不配置,则使用所有online的端口。
RestURL	-	Rest接口访问地址。	必选。
UserName	-	阵列管理员用户名。	必选。
UserPassword	-	阵列管理员密码。	必选。
StoragePool	-	需要使用的存储池名称。	必选。
SectorSize	64	文件系统磁盘块大小,可选值 "4"、"8"、"16"、"32"、 "64",单位KB。	可选。
SnapshotSuppor t	True	是否提供快照功能。	可选
ReplicationSupp ort	False	是否提供远程复制功能。	可选
WaitInterval	3	查询文件系统状态间隔时间,单位为 秒(s)。	可选。
Timeout	60	等待存储设备执行命令的超时时间, 单位为秒(s)。	可选。
SnapshotReserv e	20	快照空间比例(%)。(范围为 0~50)	可选
NFSClient\IP	-	从快照创建共享点时,管理节点上用 于挂载NFS共享点所用IP。	从快照创建共享点 必选。
CIFSClient \UserName	-	从快照创建共享点时,管理节点上用 于挂载CIFS共享点所用用户名。	从快照创建共享点 必选。
CIFSClient \UserPassword	-	从快照创建共享点时,管理节点上用 于挂载CIFS共享点所用用户密码。	从快照创建共享点 必选。

∭ 说明

- 在配置文件中可配置多个RestURL、StoragePool和Port,以分号隔开
- 共享的详细配置信息请查询CLI指令参考文档中的show share命令
- 快照功能和复制功能不能同时开启
- 所有参数值中不能含有XML特殊字符<>&'"

步骤3 重新启动Manila服务。

5 Manila Driver 高级属性配置

介绍配置存储高级属性的操作步骤。

华为存储支持一系列的smartx高级特性,这些高级特性能通过与特定的share类型关联的方式,在OpenStack环境中使用。

- 5.1 配置Thin/Thick属性
- 5.2 配置SmartDedupe属性
- 5.3 配置SmartCompression属性
- 5.4 配置SmartCache属性
- 5.5 配置SmartPartition属性
- 5.6 配置SmartQoS属性
- 5.7 指定Storage Pool创建共享点
- 5.8 指定Disk Type创建共享点
- 5.9 配置SectorSize
- 5.10 配置Replication
- 5.11 配置FS归属控制器
- 5.12 配置NFS 客户端权限
- 5.13 配置HyperMetro属性

5.1 配置 Thin/Thick 属性

介绍配置Thin/Thick属性操作步骤。

操作步骤

步骤1 执行manila type-create thin_type False命令,创建share类型。"thin_type"代表share类型名称,由用户指定;"False"表示不支持多租户,若需配置多租户,该参数配置为"True"。

步骤2 配置Thin属性: 执行以下命令, 配置thin provisioning属性为 "true"的键值对。

root@ubuntu:~# manila type-key thin_type set capabilities:thin_provisioning='<is> true'

配置Thick属性: 执行以下命令, 配置thin_provisioning属性为 "false"的键值对。

root@ubuntu:~# manila type-key thin_type set capabilities:thin_provisioning='<is> false'

步骤3 执行manila create --name test001 NFS 2 --share-type thin_type,创建支持以上属性的 share。

----结束

□ 说明

- share-type中配置 "thin_provisioning" 属性为"true", 共享点类型为"Thin"; "thin_provisioning"属性为"false"时, 共享点类型为"Thick"。
- share-type中未配置 "thin provisioning", 共享点默认类型为"Thin"。

5.2 配置 SmartDedupe 属性

介绍配置SmartDedupe属性操作步骤。

操作步骤

步骤1 执行**manila type-create dedupe_type False**命令,创建share类型。"dedupe_type"代表 share类型名称,由用户指定;"False"表示不支持多租户,若需配置多租户,该参数 配置为"True"。

步骤2 执行以下命令,配置thin_provisioning属性为"true"的键值对。

root@ubuntu:~# manila type-key dedupe_type set capabilities:thin_provisioning='<is> true'

步骤3 执行以下命令,配置SmartDedupe属性为"true"的键值对。

root@ubuntu:~# manila type-key dedupe_type set capabilities:dedupe='<is> true'

步骤4 执行manila create --name test001 NFS 2 --share-type dedupe_type, 创建支持以上属性的share。

----结束

□ 说明

仅"Thin"共享点支持配置SmartDedupe。

5.3 配置 SmartCompression 属性

介绍配置SmartCompression属性操作步骤。

操作步骤

步骤1 执行manila type-create compression_type False命令,创建share类型。

"compression_type"代表share类型名称,由用户指定; "False"表示不支持多租户,若需配置多租户,该参数配置为"True"。

步骤2 执行以下命令,配置thin provisioning属性为"true"的键值对。

root@ubuntu:~# manila type-key compression_type set capabilities:thin_provisioning='<is> true'

步骤3 执行以下命令,配置SmartCompression属性为"true"的键值对。

root@ubuntu:~# manila type-key compression_type set capabilities:compression='<is> true'

步骤4 执行manila create --name test001 NFS 2 --share-type compression_type,创建支持以上属性的share。

----结束

∭说明

仅 "Thin" 共享点支持配置SmartCompression。

5.4 配置 SmartCache 属性

介绍配置SmartCache属性操作步骤。

操作步骤

步骤1 执行manila type-create cache_type False命令,创建share类型。 "cache_type" 代表 share类型名称,由用户指定; "False"表示不支持多租户,若需配置多租户,该参数 配置为"True"。

步骤2 执行以下命令,配置SmartCache属性为"true"的键值对。

root@ubuntu:~# manila type-key cache type set capabilities:huawei smartcache='<is> true'

配置存储设备上已经存在的SmartCache名称,将cache关联到share类型中。

root@ubuntu:~# manila type-key cache_type set huawei_smartcache:cachename='test_name'

步骤3 执行manila create --name test001 NFS 2 --share-type cache_type, 创建支持以上属性的 share。

----结束

5.5 配置 SmartPartition 属性

介绍配置SmartPartition属性操作步骤。

操作步骤

步骤1 执行manila type-create partition_type False命令,创建share类型。"partition_type"代表share类型名称,由用户指定;"False"表示不支持多租户,若需配置多租户,该参数配置为"True"。

步骤2 执行以下命令,配置SmartPartition属性为"true"的键值对。

root@ubuntu:~# manila type-key partition_type set capabilities:huawei_smartpartition='<is> true'

配置存储设备上已经存在的SmartPartition名称,将partition关联到share类型中。

root@ubuntu:~# manila type-key partition_type set huawei_smartpartition:partitionname='test_name'

步骤3 执行manila create --name test001 NFS 2 --share-type partition_type,创建支持以上属性的share。

5.6 配置 SmartQoS 属性

关于本章

华为支持如下QoS属性,同一类属性可单个或多个同时配置到一个QoS属性中。

保护策略: latency、minIOPS、minBandWidth

限制策略: maxIOPS、maxBandWidth

⚠注意

保护策略和限制策略为一对互斥的特性,如果同时配置会导致创建share失败。 QoS属性中,IOType为必选项,如果QoS策略中未配置IOType会导致创建share失败。

5.6.1 配置控制最大 IOPS

介绍配置控制最大IOPS操作步骤。

操作步骤

步骤1 执行**manila type-create maxiops_type False**命令,创建share类型。"maxiops_type"代表share类型名称,由用户指定;"False"表示不支持多租户,若需配置多租户,该参数配置为"True"。

步骤2 执行以下命令,配置SmartQoS属性为"true"的键值对。

root@ubuntu:~# manila type-key maxiops_type set capabilities:qos='<is> true'

执行以下命令,配置QoS控制属性参数。

root@ubuntu:~# manila type-key maxiops_type set qos:IOType=0 qos:maxIOPS=50

- maxIOPS: 最大IOPS限制,数值为大于0的整数。
- IOType(必选):控制读写类型。"0"表示控制读IO;"1"表示控制写IO; "2"表示控制读写IO。

步骤3 执行manila create --name test001 NFS 2 --share-type maxiops_type,创建支持以上属性的share。

----结束

5.6.2 配置控制最小 IOPS

介绍配置控制最小IOPS操作步骤。

操作步骤

步骤1 执行manila type-create miniops_type False命令,创建share类型。"miniops_type"代表share类型名称,由用户指定;"False"表示不支持多租户,若需配置多租户,该参数配置为"True"。

步骤2 执行以下命令,配置SmartQoS属性为"true"的键值对。

root@ubuntu:~# manila type-key miniops_type set capabilities:qos='<is> true'

执行以下命令,配置OoS控制属性参数。

root@ubuntu:~# manila type-key miniops_type set qos:IOType=0 qos:minIOPS=50

- minIOPS: 最小IOPS限制,数值为大于0的整数。
- IOType(必选):控制读写类型。"0"表示控制读IO;"1"表示控制写IO; "2"表示控制读写IO。

步骤3 执行manila create --name test001 NFS 2 --share-type miniops_type,创建支持以上属性的share。

----结束

5.6.3 配置控制最大带宽

介绍配置控制最大带宽的操作步骤。

操作步骤

步骤1 执行manila type-create maxbandwidth_type False命令,创建share类型。

"maxbandwidth_type"代表share类型名称,由用户指定;"False"表示不支持多租户,若需配置多租户,该参数配置为"True"。

步骤2 执行以下命令,配置SmartQoS属性为"true"的键值对。

root@ubuntu:~# manila type-key maxbandwidth_type set capabilities:qos='<is> true'

执行以下命令,配置QoS控制属性参数。

root@ubuntu:~# manila type-key maxbandwidth_type set qos:IOType=0 qos:maxBandWidth=50

- maxBandWidth: 最大带宽限制,数值为大于0的整数,单位为MB/s。
- IOType(必选):控制读写类型。"0"表示控制读IO;"1"表示控制写IO; "2"表示控制读写IO。

步骤3 执行manila create --name test001 NFS 2 --share-type maxbandwidth_type,创建支持以上属性的share。

----结束

5.6.4 配置控制最小带宽

介绍配置控制最小带宽的操作步骤。

操作步骤

步骤1 执行manila type-create minbandwidth type False命令,创建share类型。

"minbandwidth_type"代表share类型名称,由用户指定;"False"表示不支持多租户,若需配置多租户,该参数配置为"True"。

步骤2 执行以下命令,配置SmartQoS属性为"true"的键值对。

root@ubuntu:~# manila type-key minbandwidth_type set capabilities:qos='<is> true'

执行以下命令, 配置OoS控制属性参数。

root@ubuntu:~# manila type-key minbandwidth type set qos:IOType=0 qos:minBandWidth=50

- minBandWidth: 最小带宽限制,数值为大于0的整数,单位为MB/s。
- IOType(必选):控制读写类型。"0"表示控制读IO;"1"表示控制写IO; "2"表示控制读写IO。

步骤3 执行manila create --name test001 NFS 2 --share-type minbandwidth_type ,创建支持以上属性的share。

----结束

5.6.5 配置控制时延

介绍配置控制时延的操作步骤。

操作步骤

步骤1 执行manila type-create latency_type False命令,创建share类型。"latency_type"代表 share类型名称,由用户指定;"False"表示不支持多租户,若需配置多租户,该参数 配置为"True"。

步骤2 执行以下命令,配置SmartOoS属性为"true"的键值对。

root@ubuntu:~# manila type-key latency_type set capabilities:qos='<is> true'

执行以下命令,配置QoS控制属性参数。

root@ubuntu:~# manila type-key latency type set gos:IOType=0 gos:latency=50

- latency: 时延限制,数值为大于0的整数,单位为ms。
- IOType(必选):控制读写类型。"0"表示控制读IO;"1"表示控制写IO; "2"表示控制读写IO。

步骤3 执行manila create --name test001 NFS 2 --share-type latency_type ,创建支持以上属性的share。

----结束

5.6.6 配置控制多策略

介绍配置控制多策略的操作步骤。

操作步骤

步骤1 执行manila type-create multiple strategy type False命令,创建share类型。

"multiple_strategy_type"代表share类型名称,由用户指定;"False"表示不支持多租户,若需配置多租户,该参数配置为"True"。

步骤2 执行以下命令,配置SmartQoS属性为"true"的键值对。

root@ubuntu:~# manila type-key multiple_strategy_type set capabilities:qos='<is> true'

执行以下命令,配置QoS控制属性参数。

root@ubuntu:~# manila type-key multiple_strategy_type set qos:IOType=0 qos:latency=50
qos:minIOPS=50 qos:minBandWidth=50

- latency: 时延限制,数值为大于0的整数,单位为ms。
- minIOPS: 最小IOPS限制,数值为大于0的整数。
- minBandWidth:最小带宽限制,数值为大于0的整数。

● IOType (必选): 控制读写类型。 "0"表示控制读IO; "1"表示控制写IO; "2"表示控制读写IO。

步骤3 执行manila create --name test001 NFS 2 --share-type multiple_strategy_type,创建支持以上属性的share。

----结束

5.7 指定 Storage Pool 创建共享点

介绍配置指定存储池创建共享点操作步骤。

操作步骤

步骤1 执行manila type-create target_pool_type False命令,创建share类型。 "target_pool_type"代表share类型名称,由用户指定;"False"表示不支持多租户,若需配置多租户,该参数配置为"True"。

步骤2 执行以下命令,配置目标存储池。

- 配置单个目标存储池 root@ubuntu:~# manila type-key target_pool_type set pool_name=StoragePool001
- 配置多个目标存储池 root@ubuntu:-# manila type-key target_pool_type set pool_name="<or>
 StoragePool002"

步骤3 执行manila create --name test001 NFS 2 --share-type target_pool_type,创建支持以上属性的share。

----结束

5.8 指定 Disk Type 创建共享点

介绍配置指定存储池磁盘类型创建共享点操作步骤。

操作步骤

步骤1 执行manila type-create disk_type False命令,创建share类型。"disk_type"代表share 类型名称,由用户指定;"False"表示不支持多租户,若需配置多租户,该参数配置为"True"。

步骤2 执行以下命令,配置存储池类型。

- 配置一种磁盘类型 root@ubuntu:~# manila type-key disk_type set huawei_disk_type=sas
- 配置多种磁盘类型 root@ubuntu:~# manila type-key disk_type set huawei_disk_type="<or>
 sas <or>
 ssd"

disk_type可选值为"ssd"、"sas"、"nl_sas"和"mix",其中"mix"是由"ssd"、"sas"、"nl sas"中的2种及以上混合。

步骤3 执行manila create --name test001 NFS 2 --share-type disk_type, 创建支持以上属性的 share。

5.9 配置 SectorSize

介绍配置共享点源文件系统磁盘块大小操作步骤。

操作步骤

步骤1 执行**manila type-create sectorsize_type False**命令,创建share类型。"sectorsize_type" 代表share类型名称,由用户指定;"False"表示不支持多租户,若需配置多租户,该参数配置为"True"。

步骤2 执行以下命令,配置SectorSize属性为"true"的键值对。

root@ubuntu:~# manila type-key sectorsize_type set capabilities:huawei_sectorsize='<is> true'

配置SectorSize取值。

root@ubuntu:~# manila type-key sectorsize_type set huawei_sectorsize:sectorsize=4

🛄 说明

针对不同的应用场景, 华为存储系统提供了5个可选的SectorSize值(4/8/16/32/64), 单位KB。

步骤3 执行manila create --name test001 NFS 2 --share-type sectorsize_type,创建支持以上属性的share。

----结束

∭说明

- share-type和xml文件中同时配置了SectorSize, 优先使用share-type中的SectorSize值;
- share-type中无可用值且xml文件中同时配置了SectorSize, 取xml文件中的SectorSize值。
- share-type和xml文件均未配置SectorSize,使用阵列提供的默认值64KB。

5.10 配置 Replication

以举例的方式,介绍配置Replication的操作步骤。

前提条件

步骤1 在 "/etc/manila/manila.conf" 文件中配置有远程复制关系的两个后端。

```
[DEFAULT]
...
enabled_share_backends = huawei_manila_1, huawei_manila_2
...

[huawei_manila_1]
share_driver = manila. share. drivers. huawei. huawei_nas. HuaweiNasDriver
manila_huawei_conf_file = /etc/manila/manila_huawei_conf_1. xml
share_backend_name = huawei_manila_1
driver_handles_share_servers = False
replication_domain = huawei_domain
local_replication = False

[huawei_manila_2]
share_driver = manila. share. drivers. huawei. huawei_nas. HuaweiNasDriver
manila_huawei_conf_file = /etc/manila/manila_huawei_conf_2. xml
share_backend_name = huawei_manila_2
driver_handles_share_servers = False
```

replication_domain = huawei_domain
local_replication = False

□ 说明

- "replication_domain"在需要使用远程复制功能时设置,参数值为自定义字符串。有远程复制关系的后端之间需要设置相同的参数值。
- "manila_huawei_conf_1.xml"、"manila_huawei_conf_2.xml"为driver自定义的配置文件,请参考章节4 Manila Driver基本属性配置。
- 要使用阵列内远程复制功能,如上添加"local_replication"参数并设置为True, "manila_huawei_conf_1.xml"、"manila_huawei_conf_2.xml"配置对接同一阵列。如不设置 该参数或参数值设置为False,表示阵列间远程复制。

步骤2 重新启动Manila服务。

----结束

操作步骤

步骤1 执行manila type-create replication_type False命令,创建share类型。 "replication_type" 代表share类型名称,由用户指定; "False"表示不支持多租户,若需配置多租户,该参数配置为"True"。

步骤2 执行以下命令,配置远程复制类型。

root@ubuntu:~# manila type-key replication_type set replication_type=dr

步骤3 执行manila create --name test001 NFS 2 --share-type replication_type, 创建支持以上属性的share。

步骤4 执行manila share-replica-create test001,针对share创建复制。

∭说明

目前Manila Driver仅支持"dr"类型的远程复制。关于"dr"详见http://docs.openstack.org/developer/manila/devref/share replication.html.

----结束

5.11 配置 FS 归属控制器

介绍配置FS归属控制器操作步骤。

操作步骤

步骤1 执行manila type-create controller_type False命令,创建share类型。 "controller_type" 代表share控制器类型名称,由用户指定; "False"表示不支持多租户,若需配置多租户,该参数配置为"True"。

root@ubuntu:~# manila type-create controller_type_A False

步骤2 执行以下命令,配置huawei_controller属性为"true"的键值对。

root@ubuntu:~# manila type-key controller_type_A set capabilities:huawei_controller='<is> true'

配置存储设备上已经存在的controllername名称,将controller_type_A 关联到share类型中。

root@ubuntu:~# manila type-key controller_type_A set huawei_controller:controllername='CTEO.A'

步骤3 执行manila create --name test001 NFS 2 --share-type controller_type_A,创建支持以上属性的share。

----结束

5.12 配置 NFS 客户端权限

介绍配置NFS客户端权限操作步骤。

操作步骤

步骤1 执行manila type-create share privilege type False命令,创建share类型。

"share_privilege_type"代表share控制器类型名称,由用户指定; "False"表示不支持多租户,若需配置多租户,该参数配置为"True"

root@ubuntu:~# manila type-create share_privilege_type False

步骤2 执行以下命令,配置huawei_share_privilege属性为"true"的键值对。

root@ubuntu:~# manila type-key share_privilege_type set capabilities:huawei_share_privilege='<is>true'

配置存储设备上已经存在的属性类型名称,将share_privilege_type关联到share类型中。

表 5-1 参数取值说明

参数名称	参数说明	参数取值
sync	写入模式	0: 同步 1: 异步
allsquash	权限限制	0: all_squash 1: no_all_squash
rootsquash	Root权限限制	0: root_squash 1: no_root_squash
secure	源端口校验限制	0: secure 1: insecure

```
root@ubuntu:~# manila type-key share_privilege_type set huawei_share_privilege:sync=0
root@ubuntu:~# manila type-key share_privilege_type set huawei_share_privilege:allsquash=0
root@ubuntu:~# manila type-key share_privilege_type set huawei_share_privilege:rootsquash=0
root@ubuntu:~# manila type-key share_privilege_type set huawei_share_privilege:secure=0
```

步骤3 执行manila create --name test001 NFS 2 --share-type share_privilege_type,创建支持以上属性的share。

----结束

5.13 配置 HyperMetro 属性

介绍配置HyperMetro属性的操作步骤。

前提条件

需要用户提前在本端和远端阵列上完成双活域、租户、租户pair以及租户可见的逻辑端口的配置。

操作步骤

步骤1 在 "/etc/manila/manila.conf" 文件中配置有NAS双活关系的两个后端。

```
[DEFAULT]
enabled_share_backends = huawei_manila_1, huawei_manila_2
[huawei_manila_1]
share_driver = manila.share.drivers.huawei.huawei_nas.HuaweiNasDriver
manila_huawei_conf_file = /etc/manila/manila_huawei_conf_1.xml
share_backend_name = huawei_manila_1
driver_handles_share_servers = False
metro_info =
    metro_domain: huawei_domain,
    local_vStore_name: local_vstore,
    remote_vStore_name: remote_vstore,
    remote_backend: remote_backend,
    metro_logic_ip: metro_logic_ip
[huawei_manila_2]
share_driver = manila.share.drivers.huawei.huawei_nas.HuaweiNasDriver
manila_huawei_conf_file = /etc/manila/manila_huawei_conf_2.xml
share_backend_name = huawei_manila_2
driver_handles_share_servers = False
metro_info =
    metro_domain: huawei_domain,
    local_vStore_name: local_vstore,
    remote_vStore_name: remote_vstore,
    remote_backend: remote_backend,
    metro_logic_ip: metro_logic_ip
```

□说明

- "metro_info"在需要使用NAS双活功能时设置,参数值为自定义列表,列表包含多个元素,每个元素包含"metro_domain"、"local_vStore_name"、
 "remote_vStore_name", "remote_backend"和 "metro_logic_ip"字段,用","隔开。具体说明参考表5-2;
- "manila_huawei_conf_1.xml", "manila_huawei_conf_2.xml"为driver自定义的配置文件,请参考章节4 Manila Driver基本属性配置;
- 暂不支持从双活share的快照创建share的操作。

表 5-2 参数含义说明

属性名称	说明
metro_domain	双活域名称。有双活关系的后端之间,需要设置相同的参 数值
local_vStore_name	双活的本端租户名称。
remote_vStore_name	双活的远端租户名称。
remote_backend	具有NAS双活关系的远端Manila后端的名称。构成NAS双活的两个后端互为"remote_backend"

属性名称	说明
metro_logic_ip	租户下的逻辑IP,构成NAS双活的两个后端配置 同一个 IP。

步骤2 重新启动Manila服务。

systemctl restart openstack-manila-volume.service

步骤3 执行manila type-create HyperMetro False命令,创建share类型。"HyperMetro"代表 share类型名称,由用户指定;"False"表示不支持多租户,若需配置多租户,该参数 配置为"True"。

步骤4 执行以下命令,配置HyperMetro类型键值对。

root@ubuntu:~# manila type-key HyperMetro set capabilities:hypermetro='<is> true'

步骤5 执行manila create --name metro001 NFS 2 --share-type HyperMetro,创建双活share。

6 最佳实践

6.1 快速对接华为存储

6.1 快速对接华为存储

配置流程

配置流程介绍了在OpenStack上通过配置Huawei Manila Driver对接华为存储系统配置步骤。

步骤1 参考3.1章节获取Manila Driver。

步骤2 在华为存储上创建或者查看待使用的文件存储池。

步骤3 配置manila.conf文件和华为自定义Driver配置文件(详见章节4)。

- 在 "/etc/manila"路径下创建华为自定义Driver配置文件,文件格式为XML。可根据实际情况修改Driver配置文件名称。例如,Driver配置文件名为 "manila huawei conf.xml"。
- 配置华为自定义Driver配置文件参数。

● 检查XML文件拥有者及用户组。

确保"/etc/manila/manila_huawei_conf.xml"文件的拥有者及用户组与"/etc/manila/manila.conf"文件的拥有者及用户组一致:

```
-rw-r--r- 1 manila manila 2662 Jul 29 02:13 manila.conf
-rw-r--r- 1 manila manila 778 Jul 30 02:56 manila_huawei_conf.xml
```

● 配置manila.conf文件。

在"/etc/manila/manila.conf"文件的最后添加如下配置项,其中volume_driver表示加载的Driver文件,share_backend_name表示的是manila share 后端的名称,manila_huawei_conf_file表示指定的华为自定义配置文件:

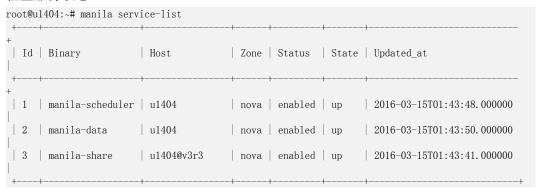
```
[huawei]
share_driver = manila.share.drivers.huawei.huawei_nas.HuaweiNasDriver
manila_huawei_conf_file = /etc/manila/manila_huawei_conf.xml
share_backend_name = huawei
driver_handles_share_servers = False
```

在[DEFAULT] 区块中修改以下内容,配置huawei后端:

```
[DEFAULT]
...
enabled_share_backends=huawei
```

步骤4 重新启动Manila服务。

步骤5 检查服务状态。



服务状态为up,代表服务已正常启动。