



Pure Storage OpenStack (2024.1 [Caracal]) Manila Driver Best Practices

Simon Dodsley, Director

Version 1.0, April 5, 2024

Contents

Notices	3
Executive Summary	4
Audience	4
OpenStack Architecture	5
Manila Architecture	6
Pure Storage Manila Driver for FlashBlade	7
Supported Versions	7
Getting the Pure Storage Manila driver for FlashBlade	7
Installation	7
Modify the Manila Configuration File	7
Restarting OpenStack Services	8
Advanced Functionality	9
Using Multiple Manila Backends	9
Share Eradication	9
Troubleshooting	10
About the Author	10

Notices

© 2023 Pure Storage, Inc. All rights reserved. Pure Storage, Pure1, and the P Logo are trademarks or registered trademarks of Pure Storage, Inc. in the U.S. and other countries. OpenStack and Train are trademarks or registered trademarks of the OpenStack Foundation, registered in many jurisdictions worldwide. All other trademarks are registered marks of their respective owners.

The Pure Storage products and programs described in this documentation are distributed under a license agreement restricting the use, copying, distribution, and decompilation/reverse engineering of the products. No part of this documentation may be reproduced in any form by any means without prior written authorization from Pure Storage, Inc. and its licensors if any. Pure Storage may make improvements and/or changes in the Pure Storage products and/or the programs described in this documentation at any time without notice.

THIS DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID. PURE STORAGE SHALL NOT BE LIABLE FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS DOCUMENTATION. THE INFORMATION CONTAINED IN THIS DOCUMENTATION IS SUBJECT TO CHANGE WITHOUT NOTICE.

Pure Storage, Inc.
650 Castro Street
Mountain View, CA 94041

Executive Summary

OpenStack® is an open source cloud computing platform designed as an Infrastructure as a Service (IaaS) implementation. It has the ability to control large numbers of compute, network and storage resources.

The file share portion of OpenStack is referred to as the Manila project and aims to provide shared filesystem storage to the OpenStack compute (Nova) instances. The Pure Storage Manila driver for FlashBlade is a Python®-based module integrated into the main OpenStack distributions.

This document describes the best practices for using the Pure Storage® OpenStack™ 2024.1 (Caracal) Manila driver for FlashBlade and details specific configuration recommendations to get the best from the Pure Storage Manila driver.

This document assumes the use of OpenStack 2024.1 (Caracal).

Audience

This document describes how to configure an OpenStack Manila node with a Pure Storage FlashBlade providing the shared storage. The intended audiences for this document include the following:

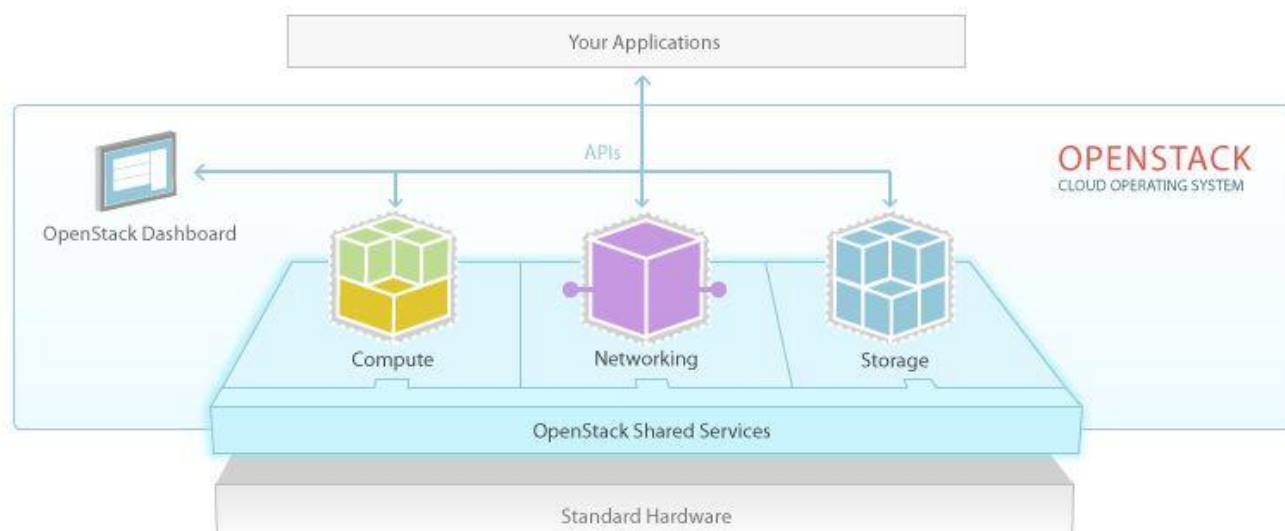
- OpenStack administrators, who configure and use OpenStack Manila
- Pure Storage system administrators, who configure and use the Purity Operating Environment
- Pure Storage partners and solutions engineers, who support the Pure Storage FlashBlade Share Driver for OpenStack Manila

This guide assumes familiarity with storage and networking concepts, OpenStack administration, Manila administration, and the Purity Operating Environment.

OpenStack Architecture

OpenStack provides services to allow storage, compute and network resources to be connected together to form a cloud using independent programs.

The high-level architecture is shown here.



At the time of writing these are the OpenStack services and associated project names:

Services	OpenStack Project
Object Storage	Swift
Compute	Nova
Image	Glance
Dashboard	Horizon
Identity	Keystone
Network	Neutron
Block Storage	Cinder
Telemetry	Ceilometer
Orchestration	Heat
File Systems	Manila
Bare Metal	Ironic

Manila Architecture

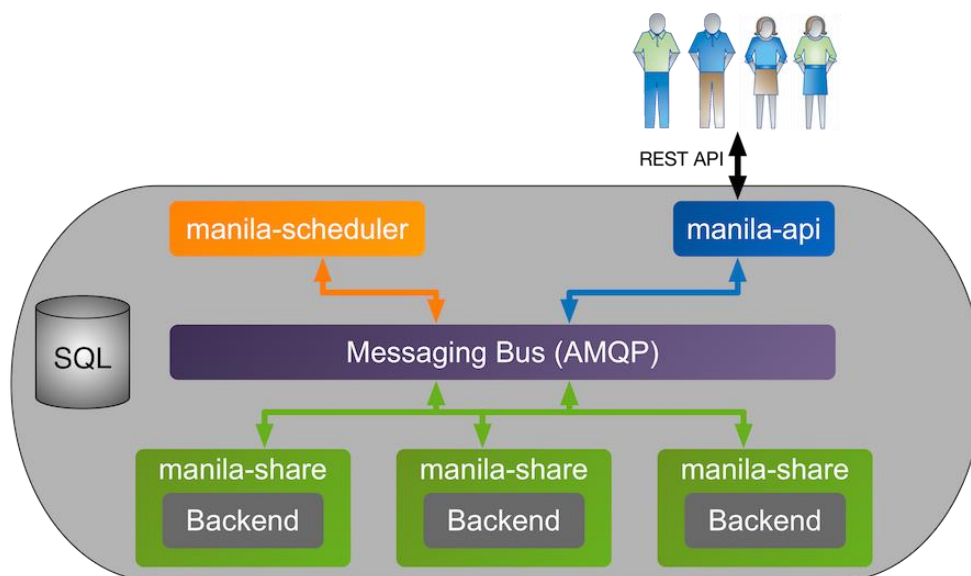
Manila is the File Systems provider to OpenStack and relies on 3 main components: Manila-API, Manila-Scheduler, and Manila-Share. Communication between these components is achieved using an advanced message queue (AMQP).

The Manila-API service provides the RESTful API for the major share operations, such as create, delete, access, expand, etc.

The API service sends requests using AMQP to the Scheduler service. This then ensures that sufficient resources are available to service before dispatching the requests to the Share service. There will be a Share service instance for each Manila backend configured in the environment.

In the case of a backend system from Pure Storage, the Manila-share instance will load the Pure Storage Manila driver to control and communicate with the backend system.

A high-level view of this can be seen below:



Pure Storage Manila Driver for FlashBlade

Supported Versions

The Pure Storage Manila driver for FlashBlade is supported for all Pure Storage FlashBlade models running Purity//FB 2.3.0 or higher, however, specific versions of Purity are required for some of the newer functionality within Manila. More details can be obtained from your Pure Storage account team.

Getting the Pure Storage Manila driver for FlashBlade

Pure Storage has contributed a Manila driver since the official Xena release of the OpenStack project. Any OpenStack release from Xena onwards will incorporate the Pure Storage Manila driver for FlashBlade.

Installation

The Pure Storage Manila driver for FlashBlade is available as a standard driver under the Manila Share service. Please refer to the installation guide of your specific OpenStack deployment, or the main OpenStack website (<http://www.openstack.org>), to ensure you have the Manila project correctly installed.

After ensuring that your OpenStack deployment has the Manila share service correctly installed, it is necessary to additionally install the Purity FB Python SDK kit. This is publicly available and is installed using the following command:

```
pip3 install purity_fb
```

Ensure that you have the *python-pip3* package installed prior to running this command, using the appropriate package installation procedures for your Linux distribution. The Purity FB Python SDK has a prerequisite of the *requests* Python package, therefore ensure that this is available.

Modify the Manila Configuration File

The `/etc/manila/manila.conf` file must include settings to allow the Manila share service to communicate with and control the Pure Storage FlashBlade. To enable this to happen, create a new stanza for the Pure Storage FlashBlade, such as `[PURE]`, and add the following parameters:

```
share_driver=manila.share.drivers.purestorage.flashblade.FlashBladeShareDriver
flashblade_mgmt_vip=<MANAGEMENT-INTERFACE>
flashblade_data_vip=<DATA-INTERFACE>
flashblade_api=<API-TOKEN>
```

where

MANAGEMENT-INTERFACE: The IP address for the FlashBlade's management interface virtual IP address or the fully qualified domain name assigned to the FlashBlade.

DATA-INTERFACE: The IP address for a FlashBlade's data interface virtual IP address or the fully qualified domain name. This should be accessible by all Nova compute nodes requiring instances to use the Manila service.

API-TOKEN: The Purity API token that the share driver uses to perform share management on this FlashBlade. More details on how to obtain the API Token are given in the Pure Storage FlashBlade documentation which can be accessed through the FlashBlade GUI.

Restarting OpenStack Services

After modifying any OpenStack configuration file, it is required to restart the associated service to ensure your modifications are accepted and implemented.

Advanced Functionality

Using Multiple Manila Backends

The previous sections have assumed that there is only one Manila backend. In reality, there may be a number of backends available to the OpenStack administrator, either of the same type or from different vendors.

If the multiple backends are exclusively using the same Pure Storage Manila for FlashBlade driver, then only the following modifications would need to be made to the above recommendations:

Each backend requires its own stanza with the `/etc/manila/manila.conf` file. The following example shows two backend systems, each with its own stanza.

```
[Pure-1]
share_backend_name=pure-1
share_driver=manila.share.drivers.purestorage.flashblade.FlashBladeShareDriver
flashblade_mgmt_vip=10.1.2.3
flashblade_data_vip=10.1.2.4
flashblade_api=<API-TOKEN>

[Pure-2]
share_backend_name=pure-2
share_driver=manila.share.drivers.purestorage.flashblade.FlashBladeShareDriver
flashblade_mgmt_vip=10.2.3.10
flashblade_data_vip=10.2.3.11
flashblade_api=<API-TOKEN>
```

Notice that each section has a new parameter, `share_backend_name`. These are unique to each backend and are used to tell Manila which backends are enabled (you may have backends listed that are not enabled for use if required).

To inform Manila which backends are actually enabled, modify the `/etc/manila/manila.conf` file and set the parameter `enabled_share_backends` with comma-separated backend names. For example:

```
enabled_share_backends=pure-1, pure-2
```

If you have backends using different drivers, i.e. different vendor systems, you need to check whether certain configuration options are supported or are going to be implemented across one or all backends.

Share Eradication

By default, when a Pure Storage share is deleted by OpenStack, the share is immediately destroyed and eradicated from the Pure Storage system.

You may set the Pure Storage Manila driver to not eradicate shares upon deletion. In this case the deleted shares are placed in the 'Destroyed Filesystems' group within the Pure Storage FlashBlade where it will remain for 24 hours before it is finally eradicated. The setting to disable immediate share eradication is:

```
flashblade_eradicate = False
```

This setting is backend specific and should be added to individual backend stanzas for the storage this feature is required upon.

Troubleshooting

If you are experiencing any issues in OpenStack, whether it be with a Manila driver or another module with the project, the first and best method of root cause analysis is by investigating the OpenStack log files.

Enhanced logging is available in all modules of OpenStack by setting the `verbose` and `debug` parameters to `true` in the module configuration files.

Examples of configuration files that can be relevant to issues with Manila drivers and their interaction with other modules are `/etc/manila/manila.conf` and `/etc/nova/nova.conf`. You must restart these services if you modify the configuration files.

All logs will be created in a module-specific subdirectory from `/var/log`. For example, the Manila log files will be found in `/var/log/manila`.

Additional troubleshooting documentation can be found on the Pure Storage Support website.

About the Author



As Director of Open Source Integrations, Simon is helping direct and implement Open Source technologies in cloud, automation and orchestration technologies within Pure Storage. Core items include best practices, reference architectures and configuration guides.

With over 25 years of storage experience across all aspects of the discipline, from administration to architectural design, Simon has worked with all major storage vendors' technologies and organizations, large and small, across Europe and the USA as both customer and service provider. He also specializes in Data Migration methodologies assisting customers in their Pure Storage transition.

Blog: <http://blog.purestorage.com/author/simon>



Pure Storage, Inc.
Twitter: [@purestorage](https://twitter.com/purestorage)
www.purestorage.com

650 Castro Street, Suite #260
Mountain View, CA 94041

T: 650-290-6088
F: 650-625-9667

Sales: sales@purestorage.com
Support: support@purestorage.com
Media: pr@purestorage.com
General: info@purestorage.com