img-storage Roll Documentation

Release 6.2

Rocks Clusters

Jan 09, 2018

Contents

1 Preface **3**
- 1.1 Requirements . . . 3

2 Installing **5**
- 2.1 On a New Server . . . 5
- 2.2 On an Existing Server . . . 6

3 Using the Img-storage Roll **7**
- 3.1 Overview of the Img-Storage server . . . 7
- 3.2 Enable remote virtual disk with Img-Storage . . . 8
- 3.3 Recovering from errors . . . 11

4 Appendix **15**
- 4.1 Attributes table . . . 15
- 4.2 ROCKS Copyright . . . 15
- 4.3 Third Party Copyrights and Licenses . . . 15

5 Indices and tables **17**

Contents:

CHAPTER 1

Preface

The Img-storage Roll provides advanced virtaul machine disks management. When installed on a Rocks Cluster along with the KVM Roll, and the ZFS-Linux Roll, it provide a unified management system to store all virtual disk on a NAS appliance and serve them through iSCSI to the various VM Container. If properly configured it can also replicate disk images localy to VM container in order to provide better performances to the virtual machine and to off-load the central NAS server.

Please read the KVM Roll documentation and the ZFS-Linux Roll documentaion before proceeding.

1.1 Requirements

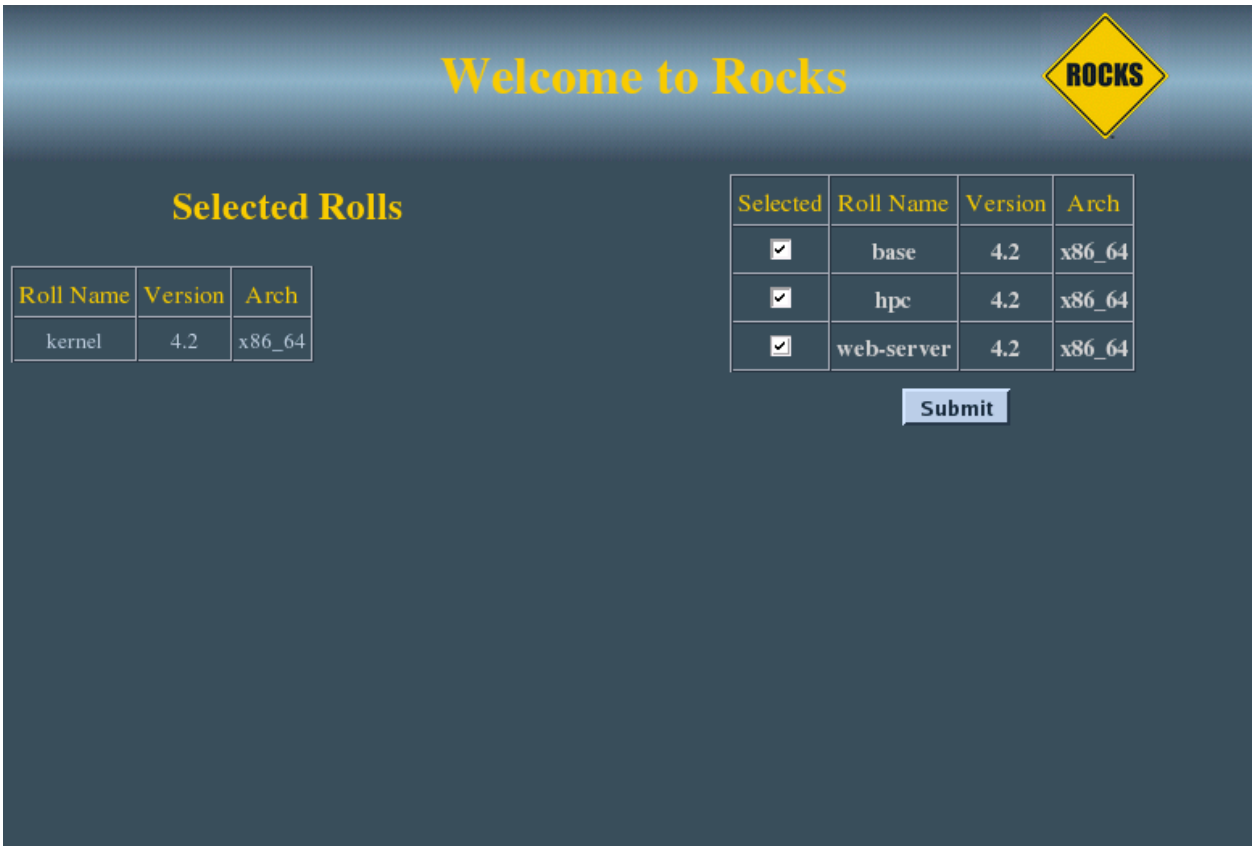
This Roll requires the KVM Roll, and the zfs-linux Roll.

CHAPTER 2

Installing

2.1 On a New Server

The img-storage Roll can be installed during the initial installation of your server (or cluster). This procedure is documented in section 1.2 of the ROCKS usersguide. You should select the img-storage from the list of available ROLLs when you see a screen that is similar to the one below.



2.2 On an Existing Server

The img-storage Roll may also be added onto an existing server (or frontend). For sake of discussion, assume that you have an iso image of the roll called img-storage.iso. The following procedure will install the Roll, and after the server reboots the Roll should be fully installed and configured.

```
$ su - root
# rocks add roll img-storage.iso
# rocks enable roll img-storage
# cd /export/rocks/install
# rocks create distro
# rocks run roll img-storage | bash
# init 6
```

CHAPTER 3

Using the Img-storage Roll

To use the Img-Storage roll is necessary to have at least one NAS server which can store all the virtual disk images. Moreover it is required to install the full OS roll.

3.1 Overview of the Img-Storage server

The Img-storage roll needs several daemons to properly function. Each daemon is implemented as an init script under `/etc/init.d` and can be restared by the service command. The tree main services are:

1. *rabbitmq-server* this component is used to orchestrate all comunication between the various nodes and is installed on the frontend. It is just a standard `Rabbit MQ Server`.
2. *img-storage-vm* this is the python daemon which is in charge of managing the iSCSI mapping on the hosting machine (the machine that will run the virtual machine), creating the local ZFS volumes, and converting these to local lvms. img-storage-vm is installed by default on all VM Container appliances but it can be installed on other nodes simply by turning to true the attribute `img_storage_vm`. You will also need KVM component on the node to properly run virtual machine, so also the attribute `kvm` should be set to true. To enable the volumes synchronization, the node should have attributes `zfs` and `img_sync` set to true.
3. *img-storage-nas* this is the python daemon which manages the virtual disk repository. It uses ZFS as the underlying technology for storage. This daemon is also responsible to set up iSCSI targets for the img-storage-vm. img-storage-nas is installed by default on all NAS appliances, but it can be changed simply using the attribute `img_storage_nas` (the attribute `zfs` should also be true in order to install zfs). img-storage-nas allocates virtual disks on a zpool(s) set by frontend configuration, which should be created manually by the administrator before any virtual machine can be used.

For example, to run virtual machine on a standard compute node it is necessary to set:

```
/opt/rocks/bin/rocks add appliance attr compute img_storage_vm true
/opt/rocks/bin/rocks add appliance attr compute kvm true
```

Then reinstall all the compute nodes.

3.1.1 Using InfiniBand interface

If an infiniband network is present on the nodes, it can be used for data transfers. To use it, set the attribute `IB_net` to the name of the InfiniBand network for all the nodes.

3.1.2 Image access modes

There are two modes supported: direct iSCSI and local synchronized disk. The mode is set by `img_sync` node attribute set to either True or False.

In direct iSCSI mode the VM performs all I/O operations to the remote zvol. The zvol is mapped to iSCSI target on NAS, which is then visible as local block device on compute node.

In sync mode it starts similar to direct iSCSI, but then synchronization is performed, and in the end the VM is using a local drive which is the copy of remove zvol on NAS. The drive is synced back to NAS when VM is terminated.

The sync mode requires zfs to be set up on VM Container nodes. The attribute `vm_container_zpool` sets the name of zpool to use and can be set both globally for all the nodes or per-node if config is different from the rest of the nodes.

3.1.3 VM boot workflow with image sync

The typical VM start workflow has several steps:

1. The zvol is created on NAS (if didn't exist before)
2. iSCSI target is created, allowing only the compute node to connect to it
3. Compute node creates local zvol for temporary write
4. Compute node attaches iSCSI to local block device
5. Compute node creates lvm that reads data from iSCSI and writes to temporary drive and responds success to NAS
6. NAS responds success to frontend which boots the VM
7. NAS starts asynchronous task whoch copies current ZVOL to compute node
8. When finished, sends signal to compute node which starts merging the local zvol with temporary write-only zvol
9. When done, switches the VM to merged local zvol

3.2 Enable remote virtual disk with Img-Storage

To enable a virtual machine to use a remote virtual disk, the name of the NAS and the name of the zpool holding the disk image must be set up. The command `rocks set host vm nas` can be used for this, while the command `rocks list host vm nas` can be used to show the current value of the NAS name. If the NAS name with the zpool name is not specified for a virtual host, it will use its original disks configuration which by default uses local raw files (`rocks list host vm showdisks=1`).

Once the NAS name is configured for a Virtual host, the virtual host will use a remote iSCSI disk provided by the given NAS. For example if we have a vritual compute node we can assign its NAS name with the following commands:

```
# rocks add host vm vm-container-0-14 compute
added VM compute-0-14-0 on physical node vm-container-0-14
# rocks set host vm nas compute-0-14-0 nas=nas-0-0 zpool=tank
# rocks start host vm compute-0-14-0
nas-0-0:compute-0-14-0-vol mapped to compute-0-14:/dev/mapper/compute-0-14-0-vol-snap
```

The virtual disks are saved on the NAS specified in the `rocks list host vm nas` under the zpool specified in the zpool field. Each volume name is created appending '-vol' to the virtual machine name.

```
# rocks list host vm nas
VM-HOST          NAS     ZPOOL
compute-0-0-0:   nas-0-0 tank
compute-0-14-0:  nas-0-0 tank
# ssh nas-0-0
# zfs list
NAME                        USED  AVAIL  REFER  MOUNTPOINT
tank                        231G  1.61T  8.08G  /tank
tank/compute-0-0-0-vol     37.1G  1.64T  3.60G  -
tank/compute-0-14-0-vol    37.1G  1.64T  3.84G  -
```

Instead of specifing the zpool each time, it is possible to set a host attribute called `img_zpools` which list the zpools (separated by a colon) that should be assigned to each disks. If multiple zpools are specified (e.g.: `rocks set host attr nas-0-0 img_zpools value="tank1,tank2"`), they will be used randomly each time the command `rocks set host vm nas` is invoked without the zpool paramter, so that the final distribution of virtual disk should be ballanced.

When the host is started and stopped (with `rocks start/stop host vm`) the zvol will be synced automatically between the NAS and the physical container.

```
# rocks list host storagemap nas-0-0
ZVOL                      HOST             ZPOOL   TARGET
↪             STATE    TIME
hpcdev-pub03-vol          hpcdev-pub02     tank    iqn.2001-04.com.nas-0-0-hpcdev-pub03-
↪vol          mapped   ----
vm-hpcdev-pub03-4-vol     ---------------- tank    ----------------------------------------
↪----------- unmapped ----
vm-hpcdev-pub03-3-vol     ---------------- tank    ----------------------------------------
↪----------- unmapped ----
vm-hpcdev-pub03-2-vol     ---------------- tank    ----------------------------------------
↪----------- unmapped ----
vm-hpcdev-pub03-0-vol     compute-0-1      tank    ----------------------------------------
↪----------- mapped   ----
vm-hpcdev-pub03-1-vol     compute-0-3      tank    iqn.2001-04.com.nas-0-0-vm-hpcdev-
↪pub03-1-vol    mapped   ----
vm-hpcdev-pub03-5-vol     compute-0-3      tank    ----------------------------------------
↪----------- mapped   ----
# rocks list host storagedev compute-0-3
ZVOL                      LVM                           STATUS             SIZE (GB)
↪BLOCK DEV IS STARTED SYNCED                    TIME
vm-hpcdev-pub03-1-vol     vm-hpcdev-pub03-1-vol-snap    snapshot-merge     36
↪sdc       1          9099712/73400320 17760    0:32:05
vm-hpcdev-pub03-5-vol     vm-hpcdev-pub03-5-vol-snap    linear             36         ---
↪------ ---------- ------------------------ -------
```

The vm-hpcdev-pub03-1-vol is currently merging, that's why we have the iSCSI target still established. Once it's done, the iSCSI target will be unmapped. The 9099712/73400320 17760 numbers whow the number of blocks left for merging: the task is done when first number, which constantly decreases, is equal to the third one.

Let's start another VM:

```
# rocks start host vm vm-hpcdev-pub03-3
nas-0-0:vm-hpcdev-pub03-3-vol mapped to compute-0-3:/dev/mapper/vm-hpcdev-pub03-3-vol-
↪snap
# rocks list host storagemap nas-0-0
ZVOL                      HOST            ZPOOL   TARGET
↪            STATE     TIME
vol1                      --------------- ------- iqn.2001-04.com.nas-0-0-vol1
↪            unmapped  -------
hpcdev-pub03-vol          hpcdev-pub02    tank    iqn.2001-04.com.nas-0-0-hpcdev-pub03-
↪vol         mapped    -------
vm-hpcdev-pub03-4-vol     --------------- tank    ----------------------------------
↪----------- unmapped  -------
vm-hpcdev-pub03-2-vol     --------------- tank    ----------------------------------
↪----------- unmapped  -------
vm-hpcdev-pub03-0-vol     compute-0-1     tank    ----------------------------------
↪----------- mapped    -------
vm-hpcdev-pub03-1-vol     compute-0-3     tank    iqn.2001-04.com.nas-0-0-vm-hpcdev-
↪pub03-1-vol    mapped    -------
vm-hpcdev-pub03-5-vol     compute-0-3     tank    ----------------------------------
↪----------- mapped    -------
vm-hpcdev-pub03-3-vol     compute-0-3     tank    iqn.2001-04.com.nas-0-0-vm-hpcdev-
↪pub03-3-vol    NAS->VM 0:00:04
# rocks list host storagedev compute-0-3
ZVOL                      LVM                           STATUS            SIZE (GB)
↪BLOCK DEV IS STARTED SYNCED                    TIME
vm-hpcdev-pub03-3-vol     vm-hpcdev-pub03-3-vol-snap    snapshot          35
↪sdd       ---------- 32/73400320 32             -------
vm-hpcdev-pub03-1-vol     vm-hpcdev-pub03-1-vol-snap    snapshot-merge    36
↪sdc       1          8950592/73400320 17472    0:36:36
vm-hpcdev-pub03-5-vol     vm-hpcdev-pub03-5-vol-snap    linear            36        ---
↪------ ---------- ------------------------ -------
```

The process of VM copy to compute node started for zvol vm-hpcdev-pub03-3-vol

There are also 'manual' commands to list, create or remove zvol synchronization, as shown below:

```
# rocks list host storagemap nas-0-0
ZVOL                      HOST            ZPOOL   TARGET
↪            STATE    TIME
hpcdev-pub03-vol          hpcdev-pub02    tank    iqn.2001-04.com.nas-0-0-hpcdev-pub03-
↪vol         mapped   ----



# rocks add host storagemap nas-0-0 tank vol1 compute-0-3 10
mapping  nas-0-0 : tank / vol1  on  compute-0-3
/dev/mapper/vol1-snap

# rocks list host storagemap nas-0-0
ZVOL                      HOST            ZPOOL   TARGET
↪            STATE     TIME
hpcdev-pub03-vol          hpcdev-pub02    tank    iqn.2001-04.com.nas-0-0-hpcdev-pub03-
↪vol         mapped    -------
vol1                      compute-0-3     tank    iqn.2001-04.com.nas-0-0-vol1
↪            NAS->VM 0:00:06

# rocks list host storagemap nas-0-0
```

```
ZVOL                        HOST            ZPOOL   TARGET
↪             STATE    TIME
hpcdev-pub03-vol            hpcdev-pub02    tank    iqn.2001-04.com.nas-0-0-hpcdev-pub03-
↪vol          mapped   ----
vol1                        compute-0-3     tank    ------------------------------------
↪----------- mapped   ----



# rocks remove host storagemap nas-0-0 vol1
unmapping   nas-0-0 : vol1
Success

# rocks list host storagemap nas-0-0
ZVOL                        HOST            ZPOOL   TARGET
↪             STATE     TIME
hpcdev-pub03-vol            hpcdev-pub02    tank    iqn.2001-04.com.nas-0-0-hpcdev-pub03-
↪vol          mapped    -------
vol1                        compute-0-3     tank    ------------------------------------
↪----------- NAS<-VM 0:00:07

# rocks list host storagemap nas-0-0
ZVOL                        HOST            ZPOOL   TARGET
↪             STATE    TIME
hpcdev-pub03-vol            hpcdev-pub02    tank    iqn.2001-04.com.nas-0-0-hpcdev-pub03-
↪vol          mapped   ----
vol1                        ---------------- tank    ------------------------------------
↪----------- unmapped ----



# rocks remove host storageimg nas-0-0 tank vol1
removing   nas-0-0 : tank / vol1
Success

# rocks list host storagemap nas-0-0
ZVOL                        HOST            ZPOOL   TARGET
↪             STATE    TIME
hpcdev-pub03-vol            hpcdev-pub02    tank    iqn.2001-04.com.nas-0-0-hpcdev-pub03-
↪vol          mapped   ----
```

3.3 Recovering from errors

> **Warning:** The scripts will not recover the data from VM container, it will be destroyed. You should manually sync back the snapshots to NAS if needed.

There is administrator script being installed with the package on NAS and VM Container nodes called imgstorageadmin. It allows cleaning the state of VM when something went wrong and return it to usable condition.

The script asks questions in order to fully recover the VM in sync mode. User can reply y(default) to run the action or type n to skip.

Example:

On VM container:

```
# imgstorageadmin
Unmap iSCSI target? [y]|n: y
From which NAS? (Don't forget .ibnet if used) nas-0-0.ibnet
0 10.2.20.250:3260,1 iqn.2001-04.com.nas-0-0-vm-hpcdev-pub03-2-vol
1 10.2.20.250:3260,1 iqn.2001-04.com.nas-0-0-vm-hpcdev-pub03-4-vol
2 10.2.20.250:3260,1 iqn.2001-04.com.nas-0-0-vol1
Which target would you like to delete? (number)2
====================================
Destroy lvm? [y]|n: y
0 vol1-snap: 0 18874368 snapshot 32/18874368 32
1 vm-hpcdev-pub03-4-vol-snap: 0 73400320 snapshot 13457872/73400320 26256
2 vm-hpcdev-pub03-2-vol-snap: 0 75497472 snapshot-merge 1086176/73400320 2144
Which lvm would you like to destroy? (number)0
====================================
Remove zvol? [y]|n: y
0 tank
1 tank/vm-hpcdev-pub03-2-vol
2 tank/vm-hpcdev-pub03-2-vol-temp-write
3 tank/vm-hpcdev-pub03-4-vol
4 tank/vm-hpcdev-pub03-4-vol-temp-write
5 tank/vol1
6 tank/vol1-temp-write
Which zvol would you like to delete? (number)5
====================================
```

Then delete second zvol manually ('zfs destroy tank/vol1-temp-write -r') or rerun the script

On NAS:

```
# imgstorageadmin
Unmap iSCSI target? [y]|n: y
Target 1: iqn.2001-04.com.nas-0-0-hpcdev-pub03-vol
Target 2: iqn.2001-04.com.nas-0-0-vol1
Target 3: iqn.2001-04.com.nas-0-0-vm-hpcdev-pub03-2-vol
Target 4: iqn.2001-04.com.nas-0-0-vm-hpcdev-pub03-4-vol
Which target number would you like to delete? (number) 2
Remove zvol mapping to VM in DB? [y]|n: y
0 hpcdev-pub03-vol tank iqn.2001-04.com.nas-0-0-hpcdev-pub03-vol hpcdev-pub02
1 vm-hpcdev-pub03-0-vol
2 vm-hpcdev-pub03-2-vol tank iqn.2001-04.com.nas-0-0-vm-hpcdev-pub03-2-vol compute-0-1
3 vm-hpcdev-pub03-5-vol tank
4 vm-hpcdev-pub03-4-vol tank iqn.2001-04.com.nas-0-0-vm-hpcdev-pub03-4-vol compute-0-1
5 vm-hpcdev-pub03-1-vol tank
6 vm-hpcdev-pub03-3-vol tank
7 vol1 tank iqn.2001-04.com.nas-0-0-vol1 compute-0-1
Which zvol? (number) 7
Done
Unbusy the zvol? [y]|n: y
0 vm-hpcdev-pub03-4-vol amq.gen-Esp2W6XQojClmQ7APoHAvQ 1409862185.2
1 vol1 amq.gen-bT045S_sjCeNcni0V-pkkQ 1409872201.94
Which zvol? (number) 1
```

The vol1 is now in clean unmapped state and is ready for mapping:

```
[root@hpcdev-pub02 ~]# rocks list host storagemap nas-0-0
ZVOL                    HOST            ZPOOL   TARGET
↪            STATE    TIME
```

```
hpcdev-pub03-vol          hpcdev-pub02    tank    iqn.2001-04.com.nas-0-0-hpcdev-pub03-
↪vol          mapped   ----
vm-hpcdev-pub03-0-vol     --------------- ------- -------------------------------------
↪----------- unmapped ----
vm-hpcdev-pub03-2-vol     compute-0-1     tank    iqn.2001-04.com.nas-0-0-vm-hpcdev-
↪pub03-2-vol    mapped   ----
vm-hpcdev-pub03-5-vol     --------------- tank    -------------------------------------
↪----------- unmapped ----
vm-hpcdev-pub03-4-vol     compute-0-1     tank    iqn.2001-04.com.nas-0-0-vm-hpcdev-
↪pub03-4-vol    mapped   ----
vm-hpcdev-pub03-1-vol     --------------- tank    -------------------------------------
↪----------- unmapped ----
vm-hpcdev-pub03-3-vol     --------------- tank    -------------------------------------
↪----------- unmapped ----
vol1                      --------------- ------- -------------------------------------
↪----------- unmapped ----
```

CHAPTER 4

Appendix

4.1 Attributes table

Attribute Name	Description of its function
`img_storage_vm`	It enables installation of the client side disk management system (by default all vm-container appliance)
`img_storage_nas`	It enables installation of the server side disk management system (by default all NAS appliance)
`IB_net`	It can be used to specify the network interface used to send the IO data
`img_zpools`	It can be used to specify a default zpool to allocate VM disk images on the NAS. It will be used by the `rocks set host vm nas` to set the zpool parameter
`img_sync`	If equal to true it will enable local synchronized disk on the given vm container (default unset)
`vm_container_zpool`	If img_sync is enable this attribute specifies the zpool name that will be used to store temporarly VM disk images on the vm-container
`img_part_zfs_mirror`	If equal to true it enables standard partitioning on nodes where img_storage_vm is enabled.
`img_download_speed` `img_upload_speed`	Optional parameters for VM container nodes for throttling the download/upload speeds (f.e. 10m, 1g) default: unlimited
`img_sync_workers`	Optional parameter setting the number of image sync workers working in parallel on NAS. Default: 5

4.2 ROCKS Copyright

4.3 Third Party Copyrights and Licenses

This section enumerates the licenses from all the third party software components of this Roll. A “best effort” attempt has been made to insure the complete and current licenses are listed. In the case of errors or ommisions please contact

the maintainer of this Roll. For more information on the licenses of any components please consult with the original author(s) or see the Rocks® GIT repository.

4.3.1 RabbitMQ

```
The contents of this file are subject to the Mozilla Public License
Version 1.1 (the "License"); you may not use this file except in
compliance with the License. You may obtain a copy of the License at
http://www.mozilla.org/MPL/

Software distributed under the License is distributed on an "AS IS"
basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the
License for the specific language governing rights and limitations
under the License.

The Original Code is RabbitMQ.

The Initial Developer of the Original Code is GoPivotal, Ltd.
Copyright (c) 2007-2013 GoPivotal, Inc.  All Rights Reserved.
```

4.3.2 Pika

```
The contents of this file are subject to the Mozilla Public License
Version 1.1 (the "License"); you may not use this file except in
compliance with the License. You may obtain a copy of the License at
http://www.mozilla.org/MPL/

Software distributed under the License is distributed on an "AS IS"
basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the
License for the specific language governing rights and limitations
under the License.

The Original Code is Pika.

The Initial Developers of the Original Code are VMWare, Inc. and
Tony Garnock-Jones.

Portions created by VMware, Inc. or by Tony Garnock-Jones are
Copyright (C) 2009-2011 VMware, Inc. and Tony Garnock-Jones.
```

You can download this guide as a PDF download.

CHAPTER 5

Indices and tables

- search