



Backups (and snapshots) with QEMU

Max Reitz <mreitz@redhat.com>

KVM Forum 2016

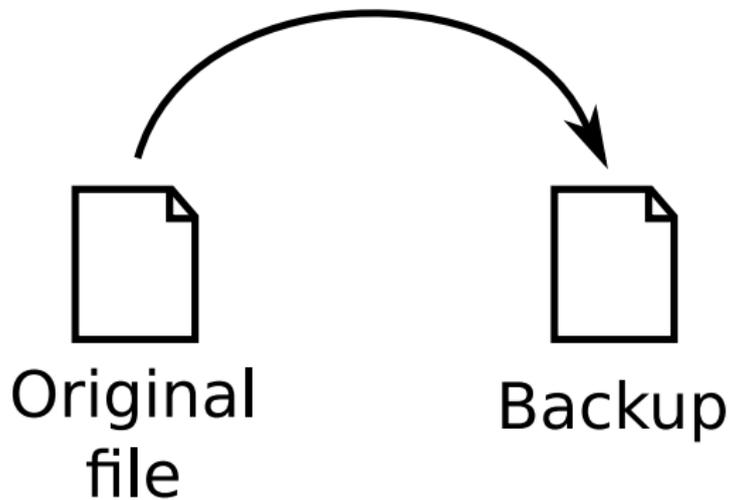


Part I

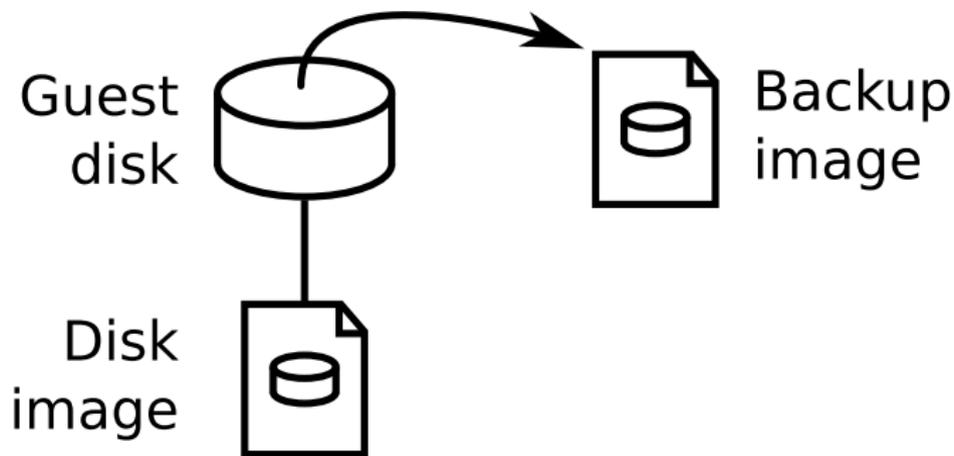
Kinds of backups

Or: Yes, snapshots are backups, too

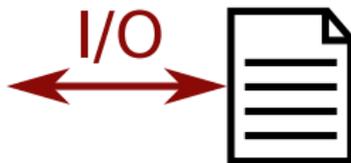
Real backups



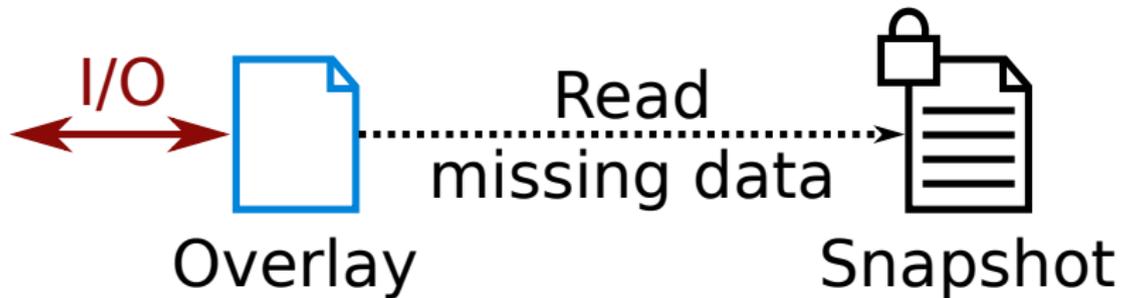
Real backups with QEMU



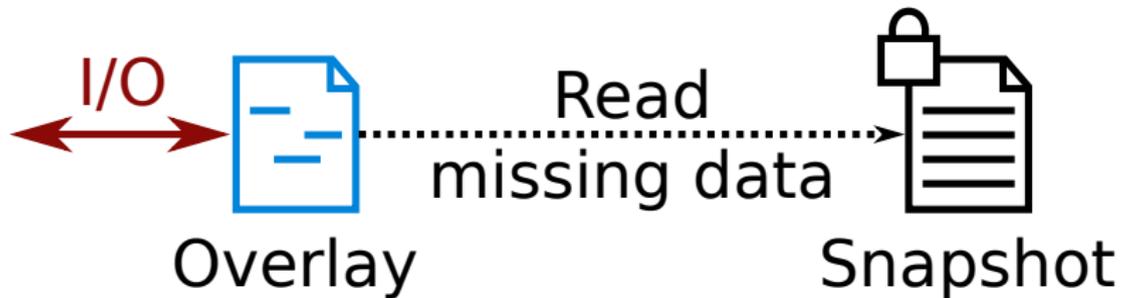
Not-so-real backups (snapshots)



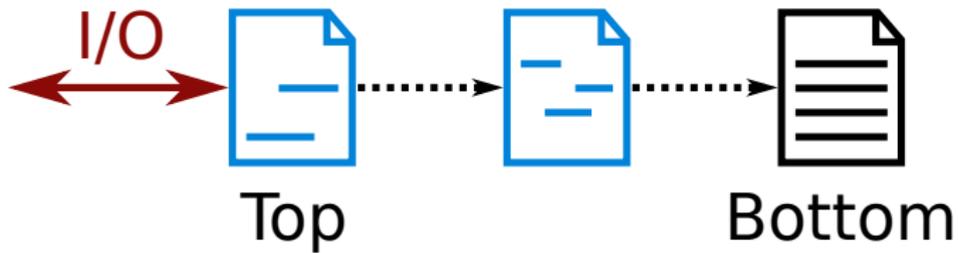
Not-so-real backups (snapshots)



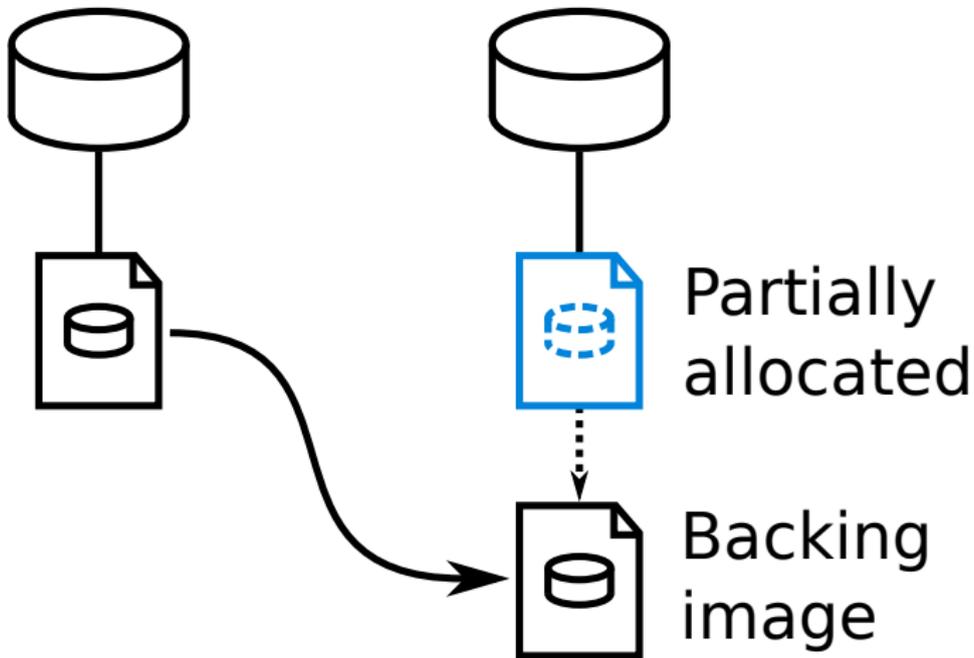
Not-so-real backups (snapshots)



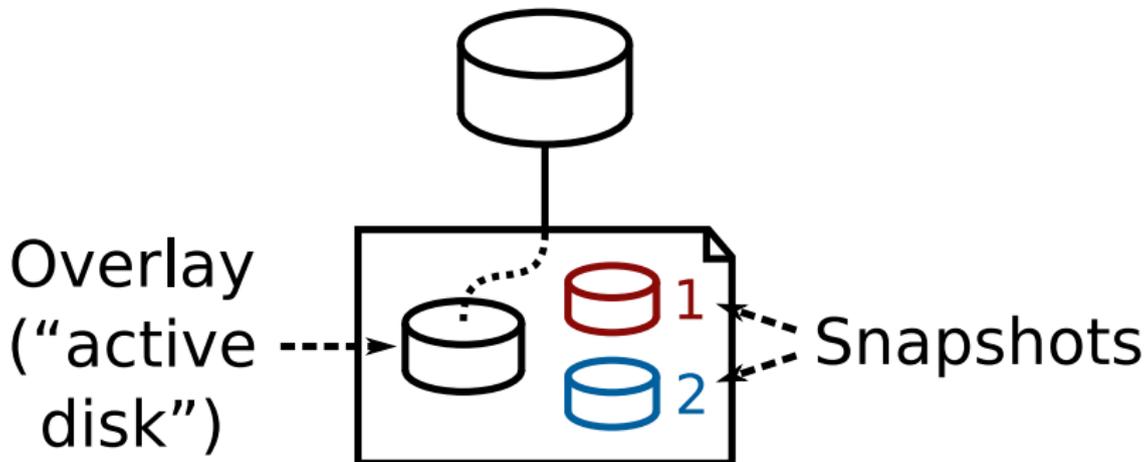
Not-so-real backups (snapshots)

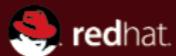


Snapshots with QEMU



Internal snapshots





Part II

How not to do backups

Or: Why this talk exists

How it doesn't work

Description of problem:

When creating an internal snapshot of virtual machine with High IO load and then remove this snapshot. The QCOW2 image is corrupted.

Status: CLOSED NOTABUG

From the command line trying to check the image also gives me:
qemu-img check foo.qcow2
qemu-img: Could not open 'foo.qcow2': Could not read snapshots: File too large

This bug appears with both the default install of qemu for ubuntu 14.04:
qemu-img version 2.0.0, Copyright (c) 2004-2008 Fabrice Bellard

And the latest version.

qemu-img version 2.1.50, Copyright (c) 2004-2008 Fabrice Bellard

I too am getting his bug.

Same error message he gets word for word.

I had the exact same issue with a VM after upgrading the host from 12.04 to 14.04.

Even going to the command prompt yields the same issue.

We had also the same issue with a vm after install the host completely new.

For example, I have a guest named "foo.img" where the backup failed. If I thereafter try to create or delete a snapshot, the following reply occurs on the command line:

```
# qemu-img snapshot -d foo /kvm/foo.img  
qemu-img: Could not open '/kvm/foo.img': Could not read snapshots: File too large
```

I never have a problem when using virsh snapshot-create or delete. Problem started with one VM when I use qemu-img snapshot. Thank you for the work-around. It's helped me too. VM working again.

Yes, we used qemu-img snapshot on the image while it was running. Did not read the man page close enough.

I stumble upon the same problem. Made the same mistake taking a snapshot using qemu-img on a running virtual machine.

Note: This bug spooked management and now I'm mandated to only use vmware. And I was sooooo close to escape too.

Zooming in...

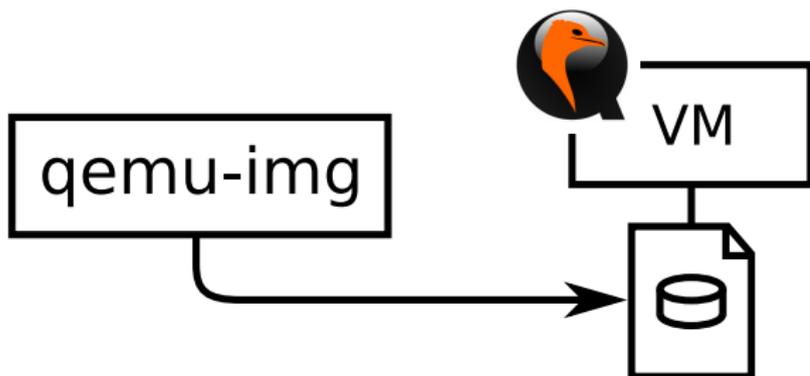
- QEMU corrupts QCOW2 images
- (Internal) snapshots taken with `qemu-img` while the VM is running



- Writing to a disk image from two processes may break it

Zooming in...

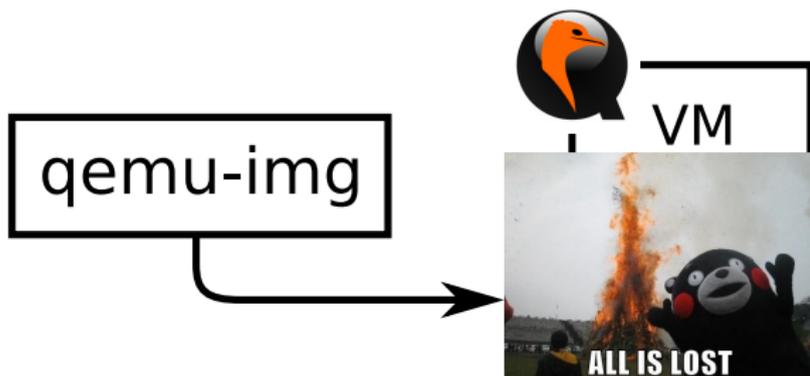
- QEMU corrupts QCOW2 images
- (Internal) snapshots taken with `qemu-img` while the VM is running



- Writing to a disk image from two processes may break it

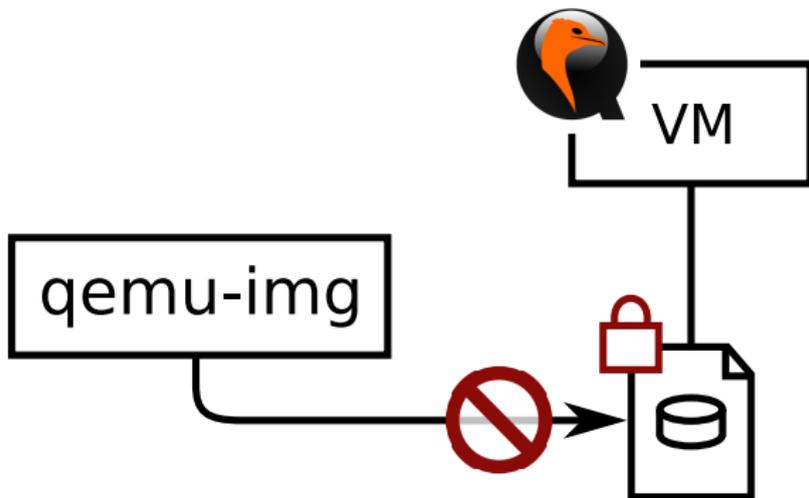
Zooming in...

- QEMU corrupts QCOW2 images
- (Internal) snapshots taken with `qemu-img` while the VM is running

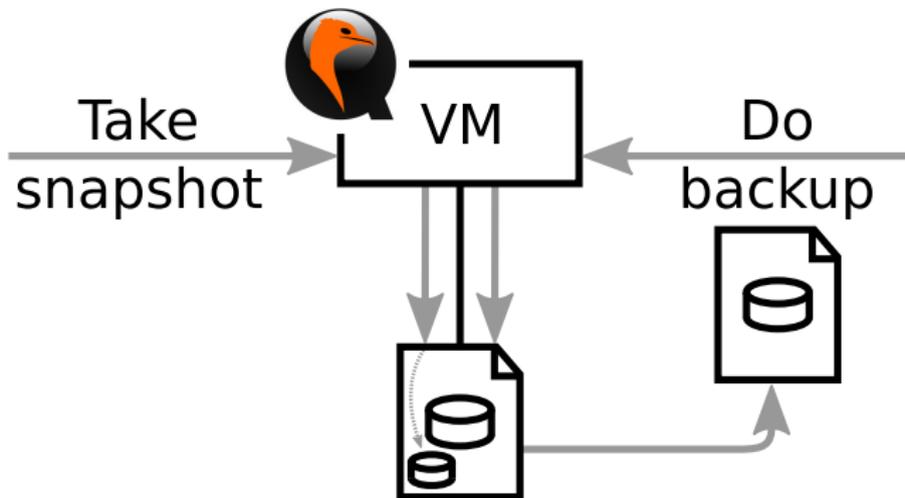


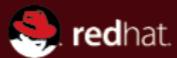
- Writing to a disk image from two processes may break it

What to do about this: Lock images



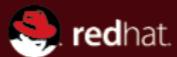
What to do about this: Inform people





Part III

Creating backups

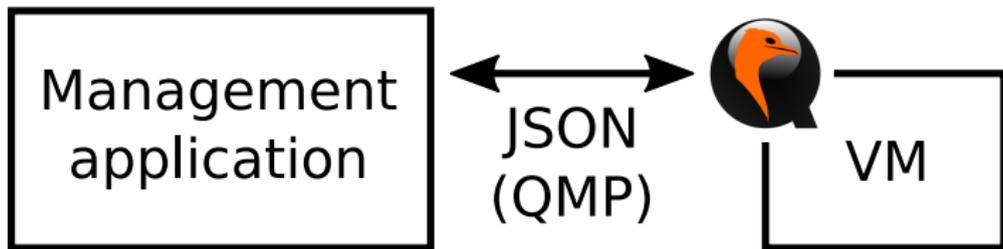


Section 1

Introduction to QMP



QMP



Run-time management:

- Pause, resume, ...
- Add/remove devices
- ...and much more, see `qmp-commands.txt`

Try it: `-qmp stdio`

QMP on the line

```
<- { "execute": "some-command",  
      "arguments": { "arg1": "value1" } }
```

```
-> { "return": {} }
```

```
-> { "error": { "class": "GenericError",  
               "desc": "..." } }
```

```
-> { "event": "SOME_EVENT",  
      "timestamp": { ... } }
```

Example command execution

```
<- { "execute": "stop" }  
-> { "timestamp": {  
    "seconds": 1467422834,  
    "microseconds": 157734 },  
    "event": "STOP" }  
-> { "return": {} }
```

Short command notation

```
{ "execute": "some-command",  
  "arguments": { "arg1": "value1" } }
```

=

```
some-command(arg1="value1")
```



Section 2

Creating snapshots

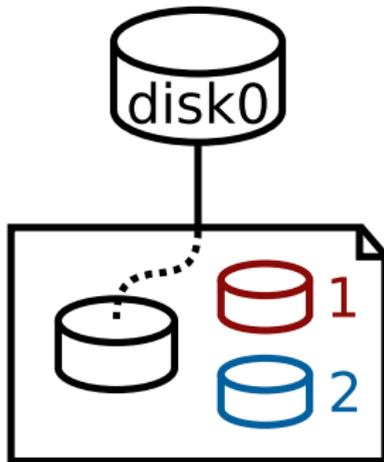
Internal snapshots

```
blockdev-snapshot-internal-sync
```

- `device`: Name of the block device
- `name`: Name of the new snapshot

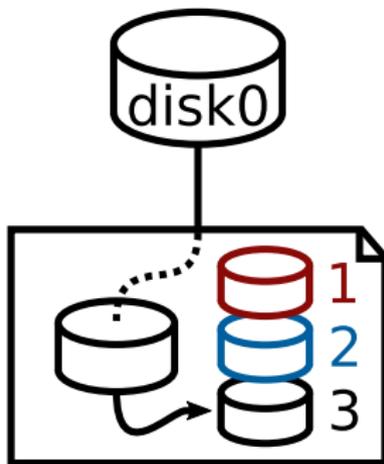
Internal snapshots

```
blockdev-snapshot-internal-sync(  
    device="disk0", name="3"  
)
```



Internal snapshots

```
blockdev-snapshot-internal-sync(  
    device="disk0", name="3"  
)
```



External snapshots

`blockdev-snapshot-sync`

- `device`: Name of the block device
- `snapshot-file`: File name of the new overlay image
- `format`: Format of the new overlay image
- (and more)

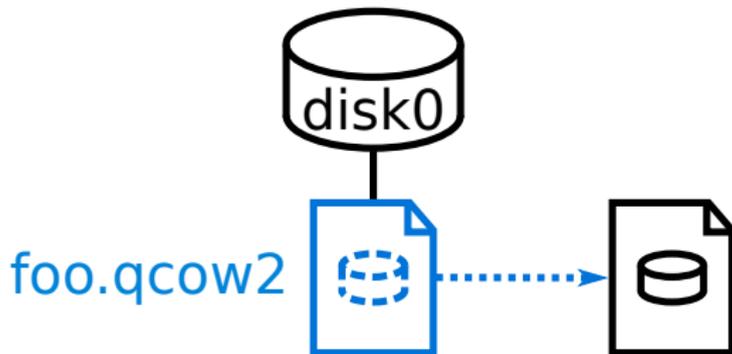
External snapshots

```
blockdev-snapshot-sync(device="disk0",  
    snapshot-file="foo.qcow2",  
    format="qcow2")
```



External snapshots

```
blockdev-snapshot-sync(device="disk0",  
    snapshot-file="foo.qcow2",  
    format="qcow2")
```





Section 3

Introduction to block jobs

What are block jobs?

- Asynchronous operations in QEMU's block layer
- Controlled using QMP



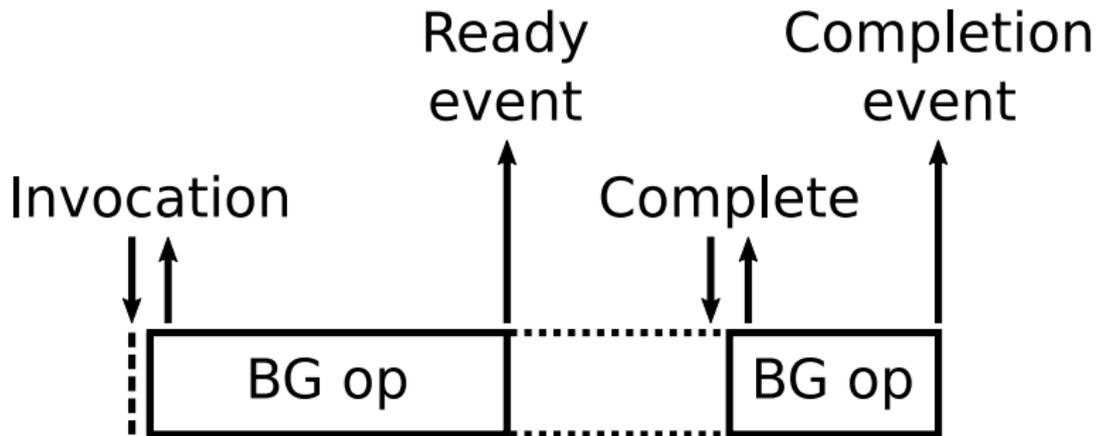
How to use block jobs: QMP

```
<- $BLOCK_JOB_NAME(job-id="foo", ...)  
-> { "return": {} }
```

[Some time passes...]

```
-> { "event": "BLOCK_JOB_COMPLETED",  
      "data": { "device": "foo", ... } }
```

Block jobs with a “ready” phase



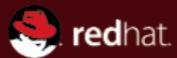
Block jobs with a “ready” phase: QMP

```
-> { "event": "BLOCK_JOB_READY",  
      "data": { "device": "foo", ... } }
```

```
<- block-job-complete(device="foo")
```

```
-> { "return": {} }
```

```
-> { "event": "BLOCK_JOB_COMPLETED",  
      "data": { "device": "foo", ... } }
```



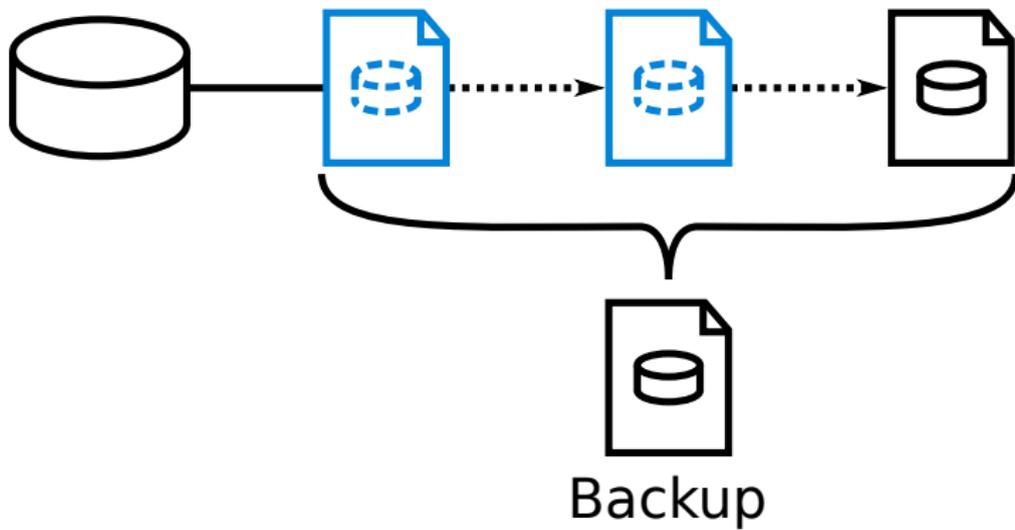
Section 4

Copying off

drive-backup **block job**

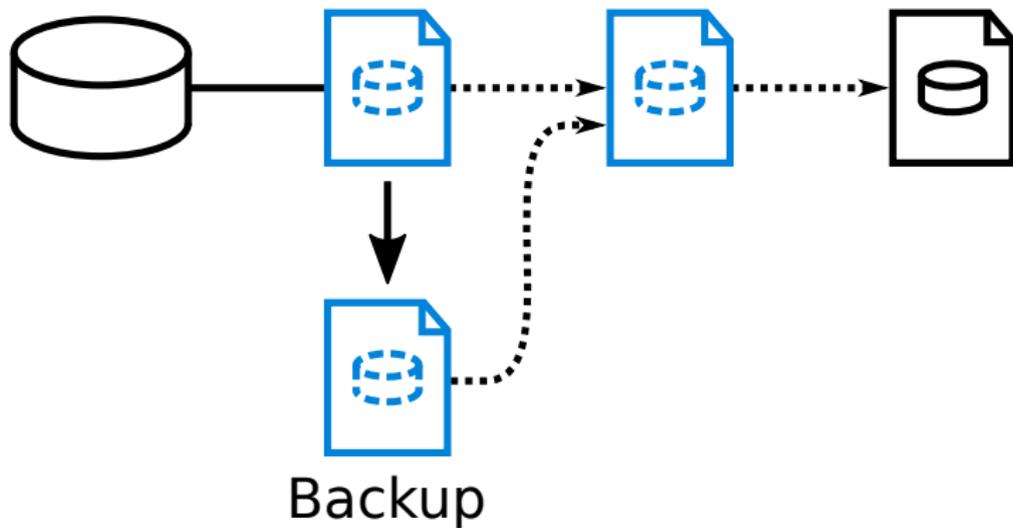
- Copies disk's data (at the start of the job) to another file
- Stalls guest writes to areas not yet saved
→ May slow the guest down
- Arguments (excerpt):
 - device: Source virtual drive
 - target: Target file name
 - sync: What to copy
 - mode: Should the target file be created?

`sync=full`
Copy everything



`sync=top`

Copy only the topmost overlay in the snapshot chain





sync=incremental

Copy data that has changed from the last backup

See John's and Vladimir's talk from last year:

The slide has a dark red background with diagonal stripes. At the top left is the Red Hat logo. To its right is the text "redhat. Odin". Below this is the main title "Incremental Backups" in a large, white, sans-serif font. Underneath the title is a subtitle in a smaller, italicized white font: "(Good things come in small packages!)". At the bottom, there are two columns of text in white. The left column identifies John Snow as a Software Engineer at Red Hat, with the date 2015-08-20. The right column identifies Vladimir Sementsov-Ogievskiy as a Software Developer at Odin.

 redhat. Odin

Incremental Backups

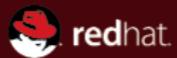
(Good things come in small packages!)

John Snow *(yes, I know)*
Software Engineer, Red Hat
2015-08-20

Vladimir Sementsov-Ogievskiy
Software Developer, Odin

drive-mirror **block job**

- Copies disk's data (at the **end** of the job) to another file
- Essential for storage migration
- Does not stall guest, but may copy the same area repeatedly
- Has a ready phase
 - `block-job-complete`: Attaches the target image to the guest disk
 - `block-job-cancel`: Similar to `drive-backup`
- No incremental sync mode



Part IV

Rolling back

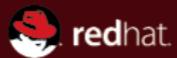
Rolling back

Backups and external snapshots:

- 1 Switch off VM
- 2 Replace active image by the backup

Internal snapshots:

- 1 Switch off VM
- 2 Use `qemu-img snapshot -a`



Part V

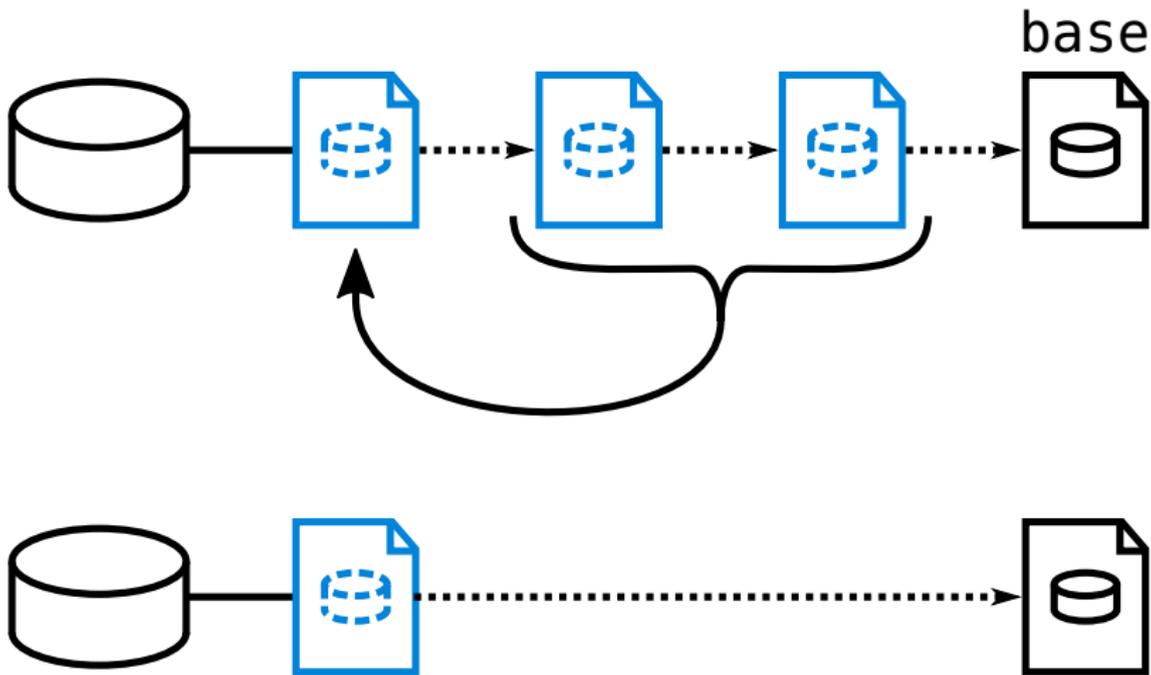
Discarding backups

Internal snapshots

`blockdev-snapshot-delete-internal-sync`

- `device`: Name of the block device
- `name`: Name of the snapshot to be deleted
- (and more)

External snapshots: block-stream block job



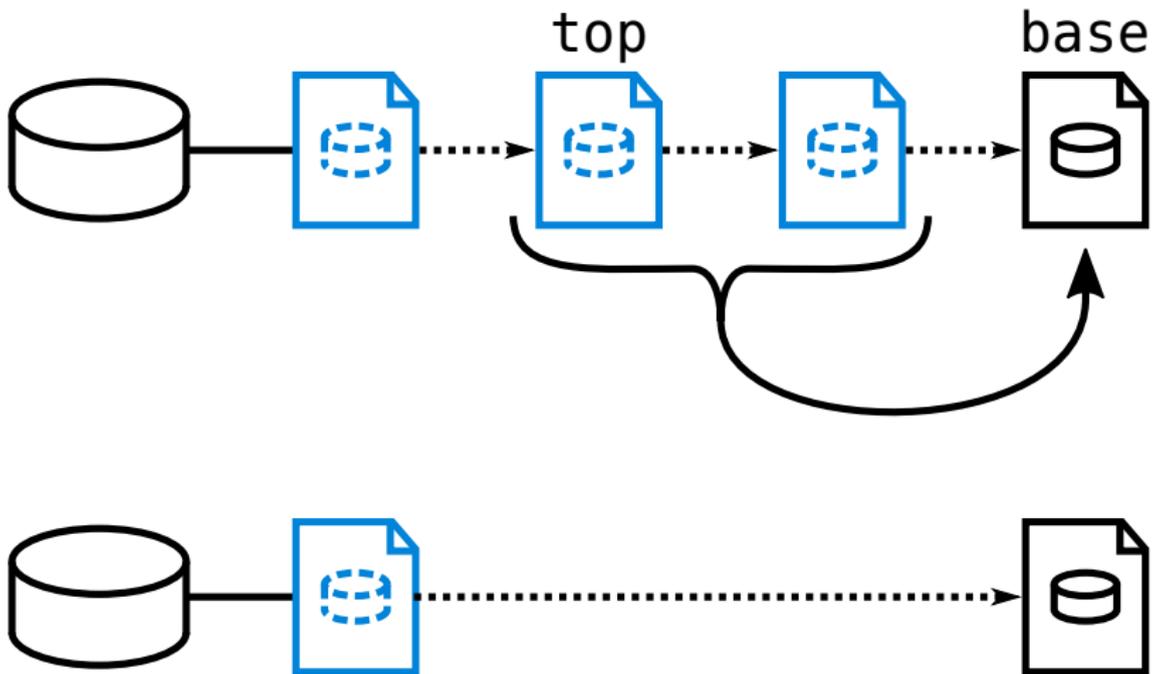
External snapshots: `block-stream` **block job**

Copies data from the snapshot to the overlay, then drops the snapshot from the backing chain

Arguments:

- `device`: Name of the block device
- `base`: File name of the topmost snapshot to keep (none if not specified)
- (and more)

External snapshots: block-commit **block job**



External snapshots: `block-commit` **block job**

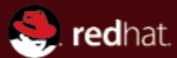
Copies data from the overlay to the snapshot, then drops the overlay from the backing chain

Arguments:

- `device`: Name of the block device
- `base`: File name of the snapshot to write into
- `top`: File name of the topmost overlay to drop (very topmost overlay if not specified)
- (and more)

Backups

- Stand-alone backups: Just delete the file
- Incremental backups: `qemu-img commit`



Part VI

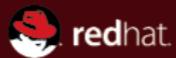
Which to use

Comparison

	Int. Sn.	Ext. Sn.	Backup
Take	Rather quick ⁽¹⁾	Quick	Slow
Roll back	Rather quick ⁽¹⁾	Quick	Slow
Discard	Rather quick ⁽¹⁾	Slow	Depends ⁽²⁾
Format	-specific	-specific	-agnostic
Off-site	No	Not really	Yes

(1) Requires quite a few metadata updates

(2) Quick for stand-alone; slow for incremental



Thank you for listening!

Questions?