

openQA Project - action #98727

[tools][sle][aarch64] the published hdd can't be booted up due to wrong format

2021-09-16 03:59 - rfan1

Status:	Resolved	Start date:	2021-09-16
Priority:	Low	Due date:	
Assignee:	okurz	% Done:	0%
Category:	Support	Estimated time:	0.00 hour
Target version:	Ready		
Difficulty:			

Description

Observation

From the test, we can see the hdd can't be booted.

<https://openqa.nue.suse.com/tests/7108821>

```
[2021-09-14T14:19:03.826 CEST] [warn] !!! : qemu-system-aarch64: -blockdev driver=qcow2,node-name=hd0-overlay0,file=hd0-overlay0-file,cache.no-flush=on: Could not open backing file: Image is not in qcow2 format
```

Steps to reproduce

1) Publish hdd with below case

<http://openqa.nue.suse.com/tests/7103389>

2) Run the cases with the published hdd

3) We can hit the issue mentioned above

Problem

I am wondering some performance problem with the backend worker

In this case, the test passed without any issue. but the qcow2 image seems not bootable

qemu-img info [sle-15-SP3-aarch64-187.1-textmode@aarch64_sb.qcow2](#)

image: [sle-15-SP3-aarch64-187.1-textmode@aarch64_sb.qcow2](#)

file format: raw

virtual size: 2.43 GiB (2607030272 bytes)

disk size: 2.43 GiB

Suggestion

Add some code checking for image's format

Workaround

a. re-trigger the job

b. switch to other worker with better performance

We used to publish a qcow2 hdd image during our tests, and this image can be used for later tests, however the hdd can't be booted up due to wrong format

Can someone help take a look at this issue?

History

#1 - 2021-09-16 04:01 - rfan1

rfan1 wrote:

The issue can be rarely seen on other platforms [Re-run the tests can fix the issue], but we can see it on aarch64 platform many times, not sure if any performance issue with arm worker.

We used to publish a qcow2 hdd image during our tests, and this image can be used for later tests, however the hdd can't be booted up due to wrong format

For example:

<http://openqa.nue.suse.com/tests/7103389#downloads>

In this case, the test passed without any issue. but the qcow2 image seems not bootable

qemu-img info [sle-15-SP3-aarch64-187.1-textmode@aarch64_sb.qcow2](#)

image: [sle-15-SP3-aarch64-187.1-textmode@aarch64_sb.qcow2](#)

file format: raw

virtual size: 2.43 GiB (2607030272 bytes)

disk size: 2.43 GiB

Can someone help take a look at this issue?

BTW, this issue was fixed later [we switched to another arm worker and the issue was gone finally]

#2 - 2021-09-16 07:24 - okurz

- Subject changed from *[sle][aarch64] the published hdd can't be booted up due to wrong format* to *[tools][sle][aarch64] the published hdd can't be booted up due to wrong format*

- Category set to *Infrastructure*

- Status changed from *New* to *Feedback*

- Assignee set to *okurz*

- Target version set to *Ready*

rfan1 wrote:

BTW, this issue was fixed later [we switched to another arm worker and the issue was gone finally]

sounds like a workaround.

Could you please adapt the ticket description according to <https://progress.opensuse.org/projects/openqav3/wiki/#Defects> so that we have the necessary information to proceed? I suggest to crosscheck the checksum of assets before and after to see if something goes wrong on generation, transfer or use.

#3 - 2021-09-16 08:19 - mkittler

I assume the *.qcow2 file is actually supposed to be qcow2? I'm just asking because we recently introduced a change which would require that the extension is matching the format.

But otherwise it looks like the image has been somehow corrupted. We also got a warning in a related section of the autoinst log:

```
[2021-09-14T12:59:21.776 CEST] [debug] running nice ionice qemu-img convert -O qcow2 /var/lib/openqa/pool/15/r
aid/hd0-overlay1 assets_public/sle-15-SP3-aarch64-187.1-textmode@aarch64_sb.qcow2
[2021-09-14T12:59:40.816 CEST] [debug] running qemu-img info --output=json assets_public/sle-15-SP3-aarch64-18
7.1-textmode@aarch64_sb.qcow2
Use of uninitialized value in subtraction (-) at /usr/lib/os-autoinst/backend/qemu.pm line 514.
  backend::qemu::do_extract_assets(backend::qemu=HASH(0xaaaaaf6581570), HASH(0xaaaaaf502b528)) called at /usr/
lib/os-autoinst/backend/driver.pm line 97
  backend::driver::extract_assets(backend::driver=HASH(0xaaaaefe41358), HASH(0xaaaaaf502b528)) called at /usr
/lib/os-autoinst/OpenQA/Isotovideo/Utils.pm line 178
  eval {...} called at /usr/lib/os-autoinst/OpenQA/Isotovideo/Utils.pm line 178
  OpenQA::Isotovideo::Utils::handle_generated_assets(OpenQA::Isotovideo::CommandHandler=HASH(0xaaaaaf6d80420)
, 1) called at /usr/bin/isotovideo line 420
[2021-09-14T12:59:40.880 CEST] [info] ::: backend::qemu::do_extract_assets: Extracting (?^u:^pflash-vars$)
[2021-09-14T12:59:40.881 CEST] [debug] running nice ionice qemu-img convert -O qcow2 /var/lib/openqa/pool/15/r
aid/pflash-vars-overlay1 assets_public/sle-15-SP3-aarch64-187.1-textmode@aarch64-uefi-vars_sb.qcow2
[2021-09-14T12:59:41.307 CEST] [debug] running qemu-img info --output=json assets_public/sle-15-SP3-aarch64-18
7.1-textmode@aarch64-uefi-vars_sb.qcow2
[2021-09-14T12:59:41.370 CEST] [debug] stopping backend process 79668
[2021-09-14T12:59:41.371 CEST] [debug] done with backend process
79273: EXIT 0
```

By the way, has this been happening more often or is this the first occurrence?

#4 - 2021-09-16 09:06 - rfan1

mkittler wrote:

I assume the *.qcow2 file is actually supposed to be qcow2? I'm just asking because we recently introduced a change which would require that the extension is matching the format.

But otherwise it looks like the image has been somehow corrupted. We also got a warning in a related section of the autoinst log:

```
[2021-09-14T12:59:21.776 CEST] [debug] running nice ionice qemu-img convert -O qcow2 /var/lib/openqa/pool/15/raid/hd0-overlay1 assets_public/sle-15-SP3-aarch64-187.1-textmode@aarch64_sb.qcow2
[2021-09-14T12:59:40.816 CEST] [debug] running qemu-img info --output=json assets_public/sle-15-SP3-aarch64-187.1-textmode@aarch64_sb.qcow2
Use of uninitialized value in subtraction (-) at /usr/lib/os-autoinst/backend/qemu.pm line 514.
  backend::qemu::do_extract_assets(backend::qemu=HASH(0xaaaaaf6581570), HASH(0xaaaaaf502b528)) called at /usr/lib/os-autoinst/backend/driver.pm line 97
  backend::driver::extract_assets(backend::driver=HASH(0xaaaaafe41358), HASH(0xaaaaaf502b528)) called at /usr/lib/os-autoinst/OpenQA/Isotovideo/Utils.pm line 178
  eval {...} called at /usr/lib/os-autoinst/OpenQA/Isotovideo/Utils.pm line 178
  OpenQA::Isotovideo::Utils::handle_generated_assets(OpenQA::Isotovideo::CommandHandler=HASH(0xaaaaaf6d80420), 1) called at /usr/bin/isotovideo line 420
[2021-09-14T12:59:40.880 CEST] [info] ::: backend::qemu::do_extract_assets: Extracting (?^u:^pflash-vars$)
[2021-09-14T12:59:40.881 CEST] [debug] running nice ionice qemu-img convert -O qcow2 /var/lib/openqa/pool/15/raid/pflash-vars-overlay1 assets_public/sle-15-SP3-aarch64-187.1-textmode@aarch64-uefi-vars_sb.qcow2
[2021-09-14T12:59:41.307 CEST] [debug] running qemu-img info --output=json assets_public/sle-15-SP3-aarch64-187.1-textmode@aarch64-uefi-vars_sb.qcow2
[2021-09-14T12:59:41.370 CEST] [debug] stopping backend process 79668
[2021-09-14T12:59:41.371 CEST] [debug] done with backend process
79273: EXIT 0
```

By the way, has this been happening more often or is this the first occurrence?

Thanks all for the kindly help on this case! the we have met this issue many times before, especially when we run a openqa job with our own branch (very strange result, any restriction with our own branches?)

But, we used to re-run the job and then the issue is fix on x86_platform, but on arm platforms, we hit 5+ times even we tried to re-run the tests again and again, finally, we tried to switch to another worker and issue was gone.

#5 - 2021-09-17 02:17 - rfan1

- Description updated

rfan1 wrote:

The issue can be rarely seen on other platforms [Re-run the tests can fix the issue], but we can see it on aarch64 platform many times, not sure if any performance issue with arm worker.

We used to publish a qcow2 hdd image during our tests, and this image can be used for later tests, however the hdd can't be booted up due to wrong format

For example:

<http://openqa.nue.suse.com/tests/7103389#downloads>

In this case, the test passed without any issue. but the qcow2 image seems not bootable

qemu-img info [sle-15-SP3-aarch64-187.1-textmode@aarch64_sb.qcow2](#)

image: [sle-15-SP3-aarch64-187.1-textmode@aarch64_sb.qcow2](#)

file format: raw

virtual size: 2.43 GiB (2607030272 bytes)

disk size: 2.43 GiB

Can someone help take a look at this issue?

#6 - 2021-09-23 17:14 - okurz

hm, I suspect that our workers openqaworker-arm-[123] can be even trusted less than we think.

note to team: I suggest to crosscheck the checksum of assets before and after to see if something goes wrong on generation, transfer or use.

@rfan I think you could help by updating the ticket with a regex matching the error condition to use

<https://github.com/os-autoinst/scripts/blob/master/README.md#auto-review---automatically-detect-known-issues-in-openqa-jobs-label-openqa-jobs-with-ticket-references-and-optional-retrigger> . Also if you like you can try to generate an image and use accordingly by triggering openQA jobs with WORKER_CLASS=openqaworker-arm-4 to force pinning to one of our newer ARM machines to see if the problem appears there as well.

#7 - 2021-09-24 01:50 - rfan1

okurz wrote:

hm, I suspect that our workers openqaworker-arm-[123] can be even trusted less than we think.

note to team: I suggest to crosscheck the checksum of assets before and after to see if something goes wrong on generation, transfer or use.

@rfan I think you could help by updating the ticket with a regex matching the error condition to use <https://github.com/os-autoinst/scripts/blob/master/README.md#auto-review---automatically-detect-known-issues-in-openqa-jobs-label-openqa-jobs-with-ticket-references-and-optionally-retrigger> . Also if you like you can try to generate an image and use accordingly by triggering openQA jobs with WORKER_CLASS=openqaworker-arm-4 to force pinning to one of our newer ARM machines to see if the problem appears there as well.

Thanks Oliver! will do!

<https://openqa.suse.de/tests/7214160>

qemu-img info [sle-15-SP3-aarch64-187.1-textmode@aarch64_sb.qcow2](#)

image: [sle-15-SP3-aarch64-187.1-textmode@aarch64_sb.qcow2](#)

file format: qcow2

virtual size: 20 GiB (21474836480 bytes)

disk size: 3.17 GiB

cluster_size: 65536

Format specific information:

compat: 1.1

compression type: zlib

lazy refcounts: false

refcount bits: 16

corrupt: false

extended l2: false

#8 - 2021-10-13 09:07 - okurz

- Priority changed from Normal to Low

as so far no further impact was reported by others I regard this as low prio. [rfan1](#) any update from your side?

#9 - 2021-10-13 09:47 - rfan1

Thanks Oliver!

Agree with you since the issue is not seen any more with higher performance worker.

#10 - 2021-10-15 03:40 - rfan1

- Copied to action #101015: [tools][sle][x86_64][aarch64][QEMU TPM] can openqa create swtprm device automatically? size:M added

#11 - 2021-10-15 03:42 - rfan1

- Copied to deleted (action #101015: [tools][sle][x86_64][aarch64][QEMU TPM] can openqa create swtprm device automatically? size:M)

#12 - 2021-11-16 10:42 - okurz

- Project changed from openQA Tests to openQA Project

- Category changed from Infrastructure to Support

- Status changed from Feedback to Resolved

Alright. I still consider it worthwhile to ensure that assets are correctly generated/uploaded/downloaded with checksums. So this can be a potential future improvement. Added in [#65271](#)