openQA Project - action #98577

Unknown ARRAY(variables matching HDD_1 or ISO in job settings

2021-09-14 07:08 - ybonatakis

Status:	Resolved	**Start date:**	2021-09-14
Priority:	Low	**Due date:**	
Assignee:	tinita	**% Done:**	0%
Category:	Concrete Bugs	**Estimated time:**	0.00 hour
Target version:	Ready		
Difficulty:			

Description

Observation

- For instance https://openqa.suse.de/tests/7092033/file/vars.json has "ARRAY(0x55bb116ac3a0)" : "SLE-15-SP3-Full-x86_64-GM-Media1.iso" which is repeated and assigned correctly on ISO.
- Another one on publiccloud https://openqa.suse.de/tests/7096497/file/vars.json has "ARRAY(0x55b5694714b8)" : "publiccloud_15sp3_Azure_BYOS_Updates.qcow2" which is what HDD_1 represents.

i havent noticed any destruction or impact on the test so far but i havent also found where this comes from.

Acceptance criteria

- **AC1**: No variables based on stringified array types present in job settings

Suggestions

- Add unit test validating variable names (assuming this is a bug in openQA or os-autoinst)
- Mark jobs with variable names containing (as incomplete (assuming this is a bug in another tool)

History

#1 - 2021-10-21 21:08 - jlausuch

- Project changed from Containers to openQA Tests

Moving this out of containers group.

#2 - 2021-10-22 08:34 - okurz

- Project changed from openQA Tests to Containers

- Subject changed from Unknown ARRAY variables on job ssettings to Unknown ARRAY variables on job settings

jlausuch backlog triaging is also in your responsibility, see https://progress.opensuse.org/projects/openqatests/wiki#ticket-backlog-triaging . Can you please find an according squad to work on this and not just move it out of the containers backlog?

#3 - 2021-10-22 14:34 - jlausuch

okurz wrote:

> jlausuch backlog triaging is also in your responsibility,
> I know, and I do that as you know. You don't need to remind me that...
>
> Can you please find an according squad to work on this and not just move it out of the containers backlog?

This was created by ybonatakis and I'm not sure where this belongs, but for sure it's not a containers topic.
Maybe ybonatakis can give us more insights.

#4 - 2021-10-25 11:37 - ybonatakis

Sure.
First first comment clarifies that this is not only in the containers job group.
For instance, it is visible also in Yast.

After checked os-autoinst-distri-opensuse i couldnt find anything suspicious. obs_sync also doesnt show this variable. So it might be something in os-autoinst which uploads the results at the end. if so, isnt it on qa-tools??

#5 - 2021-12-28 10:24 - cdywan

- Subject changed from Unknown ARRAY variables on job settings to Unknown ARRAY(variables matching HDD_1 or ISO in job settings

- Description updated

- Status changed from New to Workable

- Target version set to Ready

https://openqa.opensuse.org/tests/2094025/file/vars.json is another example with "ARRAY(0x555ea2fdf0a8)" : "opensuse-Tumbleweed-x86_64-20211218-Tumbleweed@64bit.qcow2" matching HDD_1

#6 - 2022-01-03 10:40 - cdywan

- Project changed from Containers to openQA Project

#7 - 2022-01-03 13:32 - tinita

- Category set to Concrete Bugs

- Assignee set to tinita

#8 - 2022-01-03 14:03 - tinita

- Status changed from Workable to In Progress

#9 - 2022-01-03 16:13 - cdywan

- Description updated

#10 - 2022-01-03 17:02 - tinita

I tried to reproduce it locally, by cloning https://openqa.opensuse.org/tests/2113762 , but vars.json looks ok.
I used the latest openQA and os-autoinst code.

Regarding %bmwqemu::vars - the joys of global variables and giving everybody access to a variable from anywhere in the code :(
Just because it's possible in perl, in doesn't mean it should be done.
The problem now is backwards compatibility.

My suggestion:

1. Make that variable a tied hash, so every access goes through a method call
2. As a first step, if the key it was trying to set contains invalid characters, print out a stacktrace to STDERR. This way we will hopefully find out which code set the wrong variable by looking into the log.
3. Offer an interface through an object instead of the hash directly, e.g. $vars->set(HDD_1 => $x)
4. Long-term: a) deprecate write access to hash; b) later deprecate any direct hash access

I have done 1. and 2. already, os-autoinst tests are passing, and a openQA verification run was made.

#11 - 2022-01-03 19:39 - tinita

https://github.com/os-autoinst/os-autoinst/pull/1900 Control access to %bmwqemu::vars and warn about invalid keys

#12 - 2022-01-03 19:40 - tinita

- Status changed from In Progress to Feedback

#13 - 2022-01-03 20:37 - tinita

https://github.com/os-autoinst/os-autoinst/pull/1900 merged, waiting until it is deployed

#14 - 2022-01-04 15:39 - tinita

https://openqa.opensuse.org/tests/2121661/logfile?filename=autoinst-log.txt

```
Settings key 'ARRAY(0x5642678b5cd0)' is invalid at /usr/lib/os-autoinst/bmwqemu.pm line 88.
    bmwqemu::load_vars() called at /usr/lib/os-autoinst/bmwqemu.pm line 119
    bmwqemu::init() called at /usr/bin/isotovideo line 128
```

So it is already in the file when isotovideo starts. Must be coming from openQA.

#15 - 2022-01-04 15:58 - tinita

- Status changed from Feedback to In Progress

#16 - 2022-01-04 16:37 - tinita

Problem likely located:
https://github.com/os-autoinst/openQA/blob/master/lib/OpenQA/Worker/Engines/isotovideo.pm#L99

```
sub cache_assets ($cache_client, $job, $vars, $assets_to_cache, $assetkeys, $webui_host, $pooldir, $callback)
{
                                              ^ arrayref
    return $callback->(undef) unless my $this_asset = shift @$assets_to_cache;

...

            $error
              = _handle_asset_processed($cache_client, $assets_to_cache, $asset_uri, $status, $vars, $webui_ho
st,
                                                       ^ arrayref
                $pooldir)
              unless $error;
...
}
sub _handle_asset_processed ($cache_client, $this_asset, $asset_uri, $status, $vars, $webui_host, $pooldir) {
                                            ^ arrayref

...
    $vars->{$this_asset} = ...;
            ^ arrayref
```

Easy to fix.
Trying out if I can test that.
Currently _handle_asset_processed is called directly from the test, with correct parameters.

Things like this staying unnoticed for several months could be prevented if we were using a vars object that is tied to the new bmwqemu::tiedvars class instead of a plain hash.

#17 - 2022-01-05 18:17 - tinita

PR: https://github.com/os-autoinst/openQA/pull/4431 Pass the correct asset variable to _handle_asset_processed

#18 - 2022-01-07 17:51 - tinita

- Status changed from In Progress to Feedback

Asked Santiago to comment on the UEFI_PFLASH_VARS thing, and also waiting for Marius to comment when he's back, so I put the PR on hold.

#19 - 2022-01-10 15:10 - tinita

PR https://github.com/os-autoinst/openQA/pull/4431 was merged.

From the discussion:

> @foursixnine:
> I'd add an extra test case to check that defined pflash with undefined uefi, also works, for today and for the future's sake, as there are no tests verifying that path at all.
> tinita:
> I think I don't understand how such a test would look like.
> Can you elaborate?

#20 - 2022-01-10 15:30 - szarate

tinita wrote:

> PR https://github.com/os-autoinst/openQA/pull/4431 was merged.
>
> From the discussion:
>
> > @foursixnine:
> > I'd add an extra test case to check that defined pflash with undefined uefi, also works, for today and for the future's sake, as there are no tests verifying that path at all.
> > tinita:
> > I think I don't understand how such a test would look like.
> > Can you elaborate?

Right now we're only testing the cases for either correct settings (this issue) or that asset caching actually works as expected... But there's no test for the case where the job does have UEFI=1 with or without PFLASH defined.

As it is a very seldom use case, I fear that some other weird regression might show up years in the future (Like this one for instance), might not be harmful, but annoying to find, I might be mistaken of course :) but is not really a big need atm... as 3 and 4 from you comment above would be

https://github.com/os-autoinst/openQA/pull/4431/files#diff-831bb8b106b0ca2b8cfaa3dd2dfd6627b1aa8e86349cd67aa279afd922091d0eR219

#21 - 2022-01-14 11:16 - tinita

4 days ago we still had the array reference: https://openqa.opensuse.org/tests/2130637/file/vars.json
3 days ago when the fix was deployed it is gone: https://openqa.opensuse.org/tests/2132212/file/vars.json

Regarding the test szarate suggests - I wouldn't know how to create a test with or without PFLASH defined. I searched the database for jobs with a PFLASH setting but didn't find any.

#22 - 2022-01-14 11:21 - tinita

tinita wrote:

> My suggestion:
>
> 1. Make that variable a tied hash, so every access goes through a method call
> 2. As a first step, if the key it was trying to set contains invalid characters, print out a stacktrace to STDERR. This way we will hopefully find out which code set the wrong variable by looking into the log.
> 3. Offer an interface through an object instead of the hash directly, e.g. $vars->set(HDD_1 => $x)
> 4. Long-term: a) deprecate write access to hash; b) later deprecate any direct hash access
>
> I have done 1. and 2. already, os-autoinst tests are passing, and a openQA verification run was made.

Since the actual bug was not in bwmqemu::vars directly, but in openQA, and with the tied hash we have an easy mechanism to add checks or forbid certain keys, 3 and 4 would be a nice-to-have only.
But it would also make sense to have the same tied hash in openQA when passing around the %vars hash and filling it.

#23 - 2022-01-19 12:35 - tinita

- Status changed from Feedback to Resolved

As discussed in today's meeting, the actual issue is resolved, and we currently think we don't need a test as suggested above (or it needs to be specified more clearly how the test should look like).