openQA Infrastructure - action #87970

tumbleweed container images in gitlab CI fail on salt calls, e.g. "KeyError: 'cmd.run_all'"

2021-01-19 10:27 - okurz

Status:	Resolved	**Start date:**	2021-02-02
Priority:	Low	**Due date:**	2021-02-16
Assignee:	okurz	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:	Ready		

Description

Observation

openqa-salt-states gitlab CI pipelines fail, likely since some change in upstream Tumbleweed, e.g. in https://gitlab.suse.de/openqa/salt-states-openqa/-/jobs/322945#L5895 with:

```
    Function: pkg.installed
      Result: False
     Comment: An exception occurred in this state: Traceback (most recent call last):
                File "/usr/lib/python3.8/site-packages/salt/state.py", line 1987, in call
                  # duration in milliseconds.microseconds
                File "/usr/lib/python3.8/site-packages/salt/loader.py", line 2031, in wrapper
                  # improperly loaded
                File "/usr/lib/python3.8/site-packages/salt/states/pkg.py", line 1865, in installe
d
                  pkg_ret = __salt__["pkg.install"](
                File "/usr/lib/python3.8/site-packages/salt/modules/zypperpkg.py", line 1662, in i
nstall
                  __zypper__(
                File "/usr/lib/python3.8/site-packages/salt/modules/zypperpkg.py", line 336, in __
call
                  self.__call_result = __salt__["cmd.run_all"](cmd, **kwargs)
                File "/usr/lib/python3.8/site-packages/salt/loader.py", line 1269, in __getitem__
                File "/usr/lib/python3.8/site-packages/salt/utils/lazy.py", line 111, in __getitem
__
              KeyError: 'cmd.run_all'
```

I assume this is because of recent changes in upstream Tumbleweed.

Suggestions

We could try to run a leap based image as temporary workaround.

History

#1 - 2021-01-19 12:26 - okurz

- Due date set to 2021-02-03

- Status changed from In Progress to Feedback

- Priority changed from High to Low

- Start date changed from 2021-01-19 to 2021-02-02

I reproduced the problem with a run of a pipeline against master branch. Within https://gitlab.suse.de/openqa/salt-states-openqa/-/merge_requests/433 we could see the problem fixed. Merged. We should try again with the Tumbleweed based image in some time, e.g. after 20201-02-02.

#2 - 2021-02-05 14:26 - cdywan

okurz wrote:

> I reproduced the problem with a run of a pipeline against master branch. Within https://gitlab.suse.de/openqa/salt-states-openqa/-/merge_requests/433 we could see the problem fixed. Merged. We should try again with the Tumbleweed based image in some time, e.g. after 20201-02-02.

Should we try and switch back then? Or, arguably we can keep it as-is since there's nothing particularly interesting in this container anyway.

#3 - 2021-02-07 21:43 - okurz

Hm, we could just keep the Leap image until the next time that makes problems or one creates a merge request trying out if the Tumbleweed image works ok again. What do you think?

#4 - 2021-02-10 18:53 - okurz

- Due date deleted (2021-02-03)

With a Tumbleweed container image since some weeks we see errors when applying a salt state, like here https://gitlab.suse.de/okurz/salt-states-openqa/-/jobs/336438 , no problems in a Leap 15.2 image , https://gitlab.suse.de/openqa/salt-states-openqa/-/jobs/336166, an error shown for example is :

```
[ERROR    ] Rendering exception occurred
Traceback (most recent call last):
  File "/usr/lib/python3.8/site-packages/salt/utils/templates.py", line 498, in render_jinja_tmpl
    output = template.render(**decoded_context)
  File "/usr/lib/python3.8/site-packages/jinja2/environment.py", line 1090, in render
    self.environment.handle_exception()
  File "/usr/lib/python3.8/site-packages/jinja2/environment.py", line 832, in handle_exception
    reraise(*rewrite_traceback_stack(source=source))
  File "/usr/lib/python3.8/site-packages/jinja2/_compat.py", line 28, in reraise
    raise value.with_traceback(tb)
  File "<template>", line 8, in top-level template code
  File "/usr/lib/python3.8/site-packages/jinja2/filters.py", line 1016, in do_list
    return list(value)
  File "/usr/lib/python3.8/site-packages/jinja2/filters.py", line 1323, in select_or_reject
    for item in seq:
  File "/usr/lib/python3.8/site-packages/jinja2/filters.py", line 1116, in do_map
    if seq:
jinja2.exceptions.UndefinedError: 'dict object' has no attribute 'SSDs'
During handling of the above exception, another exception occurred:
Traceback (most recent call last):
  File "/usr/lib/python3.8/site-packages/salt/utils/templates.py", line 260, in render_tmpl
    output = render_str(tmplstr, context, tmplpath)
  File "/usr/lib/python3.8/site-packages/salt/utils/templates.py", line 505, in render_jinja_tmpl
    raise SaltRenderError("Jinja variable {}{}".format(exc, out), buf=tmplstr)
salt.exceptions.SaltRenderError: Jinja variable 'dict object' has no attribute 'SSDs'
```

I am not sure where the attribute "SSDs" should come from.

Asked in https://chat.suse.de/channel/salt?msg=63HKsbAAuBcQHhysw, waiting for feedback

#5 - 2021-02-11 12:32 - okurz

- Due date set to 2021-02-16

Ok, I learned with the awesome help from agraul in https://chat.suse.de/channel/salt that "SSDs" is replaced to "ssds". So testing based on a Tumbleweed image helps us to be future-proof :)

https://gitlab.suse.de/openqa/salt-states-openqa/-/merge_requests/450 is good to go now.

#6 - 2021-02-15 16:00 - okurz

- Status changed from Feedback to Resolved

merged.

https://gitlab.suse.de/okurz/salt-states-openqa/-/pipelines/100433