

## Arvados - Bug #16392

### [keep] trailing slash in InternalURLs entry for keepstore causes problems with keepproxy

05/01/2020 01:08 AM - Ward Vandewege

<b>Status:</b>	Resolved	<b>Start date:</b>	05/13/2020
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assigned To:</b>	Lucas Di Pentima	<b>% Done:</b>	100%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	2020-05-20 Sprint		
<b>Description</b>			
Our dev cluster, `ce8i5`, had this in its `config.yml` file:			
<pre>Services:   Keepstore:     InternalURLs:       "http://10.47.0.6:25107/": {}</pre>			
Since feature <a href="#">#16328</a> , keepproxy now uses those entries instead of trying to discover keepstores via the api server.			
It turns out that the trailing slash in the InternalURLs entry causes problems. This is what happens on the keepstore in question:			
<pre>May 01 01:02:54 keep0.ce8i5.arvadosapi.com keepstore[42411]: {"PID":42411,"RequestID":"req-ck4ozsd n8womgzwr5fva","level":"info","msg":"request","remoteAddr":"10.47.0.5:39818","reqBytes":24,"reqFor wardedFor":"","reqHost":"10.47.0.6:25107","reqMethod":"PUT","reqPath":"/2ad1390f7b534cf6b257656067 96626a","reqQuery":"","time":"2020-05-01T01:02:54.680169064Z"} May 01 01:02:54 keep0.ce8i5.arvadosapi.com keepstore[42411]: {"PID":42411,"RequestID":"req-ck4ozsd n8womgzwr5fva","level":"info","msg":"response","remoteAddr":"10.47.0.5:39818","reqBytes":24,"reqFo rwardedFor":"","reqHost":"10.47.0.6:25107","reqMethod":"PUT","reqPath":"/2ad1390f7b534cf6b25765606 796626a","reqQuery":"","respBytes":0,"respStatus":"Moved Permanently","respStatusCode":301,"time": "2020-05-01T01:02:54.680293465Z","timeToStatus":0.000129,"timeTotal":0.000132,"timeWriteBody":0.00 0002} May 01 01:02:54 keep0.ce8i5.arvadosapi.com keepstore[42411]: {"PID":42411,"RequestID":"req-ck4ozsd n8womgzwr5fva","level":"info","msg":"request","remoteAddr":"10.47.0.5:39818","reqBytes":0,"reqForw ardedFor":"","reqHost":"10.47.0.6:25107","reqMethod":"GET","reqPath":"/2ad1390f7b534cf6b25765606796 626a","reqQuery":"","time":"2020-05-01T01:02:54.687455786Z"} May 01 01:02:54 keep0.ce8i5.arvadosapi.com keepstore[42411]: {"PID":42411,"RequestID":"req-ck4ozsd n8womgzwr5fva","level":"info","msg":"response","remoteAddr":"10.47.0.5:39818","reqBytes":0,"reqFo rwardedFor":"","reqHost":"10.47.0.6:25107","reqMethod":"GET","reqPath":"/2ad1390f7b534cf6b2576560679 6626a","reqQuery":"","respBody":"Forbidden\n","respBytes":10,"respStatus":"Forbidden","respStatusC ode":403,"time":"2020-05-01T01:02:54.687560487Z","timeToStatus":0.000094,"timeTotal":0.000100,"tim eWriteBody":0.000006}</pre>			
And of course keepproxy then relays the familiar, but not so informative error to the client:			
Traceback (most recent call last): File "/usr/bin/arv-put", line 7, in <module> main() File "/usr/share/python2.7/dist/python-arvados-python-client/lib/python2.7/site-packages/arvados /commands/put.py", line 1304, in main writer.start(save_collection=not (args.stream or args.raw)) File "/usr/share/python2.7/dist/python-arvados-python-client/lib/python2.7/site-packages/arvados /commands/put.py", line 628, in start self._local_collection.manifest_text() File "/usr/share/python2.7/dist/python-arvados-python-client/lib/python2.7/site-packages/arvados /arvfile.py", line 270, in synchronized_wrapper return orig_func(self, *args, **kwargs) File "/usr/share/python2.7/dist/python-arvados-python-client/lib/python2.7/site-packages/arvados /collection.py", line 1014, in manifest_text self._my_block_manager().commit_all() File "/usr/share/python2.7/dist/python-arvados-python-client/lib/python2.7/site-packages/arvados /arvfile.py", line 816, in commit_all raise KeepWriteError("Error writing some blocks", err, label="block")			

```
arvados.errors.KeepWriteError: Error writing some blocks: block 2ad1390f7b534cf6b25765606796626a+2
4 raised KeepWriteError (failed to write 2ad1390f7b534cf6b25765606796626a after 2 attempts (wanted
2 copies but wrote 0): service https://keep.ce8i5.arvadosapi.com:443/ responded with 413 HTTP/1.1
100 Continue
HTTP/1.1 413 Request Entity Too Large)
```

Removing the trailing slash from the InternalURLs entry resolved the problem.

It would probably be most user friendly if keepproxy (or our config parsing library?) can handle entries both with and without trailing slash. We've named the field `InternalURLs`, so it is to be expected that some people will put a trailing slash there, and some will not.

#### Subtasks:

Task # 16410: Review 16392-url-trailing-slash

Resolved

Task # 16432: Remove any trailing slashes from exported config URLs on WB2

Resolved

#### Related issues:

Related to Arvados - Bug #16328: [keep-proxy] should use config.yaml to ident...

Resolved

04/21/2020

### Associated revisions

#### Revision 18286a31 - 05/13/2020 08:16 PM - Tom Clegg

Merge branch '16392-url-trailing-slash'

refs #16392

Arvados-DCO-1.1-Signed-off-by: Tom Clegg <[tom@tomclegg.ca](mailto:tom@tomclegg.ca)>

#### Revision 366e40d9 - 05/14/2020 09:37 PM - Lucas Di Pentima

15881: Simplifies code that decides if it need to show a login form.

Also, remove any trailing slashes on service's ExternalURLs coming from the exported cluster config.

Refs #16392

Arvados-DCO-1.1-Signed-off-by: Lucas Di Pentima <[lucas@di-pentima.com.ar](mailto:lucas@di-pentima.com.ar)>

### History

#### #1 - 05/01/2020 01:16 AM - Ward Vandewege

- Description updated

#### #2 - 05/01/2020 01:17 AM - Ward Vandewege

- Description updated

#### #3 - 05/01/2020 01:21 AM - Ward Vandewege

- Related to Bug #16328: [keep-proxy] should use config.yaml to identify upstream keepstores added

#### #4 - 05/06/2020 03:44 PM - Tom Clegg

- Assigned To set to Tom Clegg

#### #5 - 05/12/2020 01:53 PM - Tom Clegg

- Status changed from New to In Progress

16392-url-trailing-slash @ [8605bce9000a8a4dfa1ae0ea5cbe1cfac0b8cc1b](https://ci.arvados.org/view/Developer/job/developer-run-tests/1848/) -- <https://ci.arvados.org/view/Developer/job/developer-run-tests/1848/>

#### #6 - 05/13/2020 06:25 PM - Lucas Di Pentima

Reviewing [8605bce9000a8a4dfa1ae0ea5cbe1cfac0b8cc1b](https://ci.arvados.org/view/Developer/job/developer-run-tests/1848/)

- Is the config.UnmarshalText() change enough to make keepproxy et al work with & without a trailing slash?
- Do you think a specific test addressing the particular issue on this ticket would be useful, or it isn't worth it as it fixes a problem happening in between releases?
- Arvbox config URLs are being generated mixed with and without trailing slashes, it may be generating the issue described below.
- Workbench2 on arvbox auto logs out when logging in. Ran the integration tests against this branch (that use PAM instead of SSO, not sure if it has anything to do with this) and they pass OK.

## #7 - 05/13/2020 08:05 PM - Tom Clegg

- Is the `config.UnmarshalText()` change enough to make `keepproxy` et al work with & without a trailing slash?
- Do you think a specific test addressing the particular issue on this ticket would be useful, or it isn't worth it as it fixes a problem happening in between releases?

The `UnmarshalText` change exposes the bug in the `ServerRequiredConfigYmlSuite.TestGetIndex` test, and then this fixes it:

```
    for k := range cluster.Services.Keepstore.InternalURLs {  
-       arv.KeepServiceURIs = append(arv.KeepServiceURIs, k.String())  
+       arv.KeepServiceURIs = append(arv.KeepServiceURIs, strings.TrimRight(k.String(), "/"))  
    }
```

(If you remove the `TrimRight` you should see the test fail.)

- `Arvbox` config URLs are being generated mixed with and without trailing slashes, it may be generating the issue described below.
- `Workbench2` on `arvbox` auto logs out when logging in. Ran the integration tests against this branch (that use PAM instead of SSO, not sure if it has anything to do with this) and they pass OK.

I don't quite follow -- are you saying this branch breaks `arvbox+wb2` login? The goal here is that you can add/remove trailing slashes in your config files with no effect at all, so I don't think that should be an issue. However, it's possible this causes a double-slash problem. We've had a situation before where `POST https://example//foo` gets transparently redirected (by Go stdlib) to `GET https://example/foo`, causing surprising behavior. Perhaps it's something like that?

```
src/common/config.ts:88:    config.rootUrl = clusterConfigJSON.Services.Controller.ExternalURL;  
src/store/auth/auth-action-session.ts:266:        window.location.href = `${rootUrl.toString()}/login?return_to=` + encodeURI(`${window.location.protocol}//${window.location.host}/add-session?baseURL=` + encodeURI(rootUrl.toString()));
```

Perhaps [arvados-workbench-2:source:src/common/config.ts](#) should systematically strip trailing "/" from all of the config-exported URLs so stuff like ``${rootUrl.toString()}/login`` doesn't end up with a double slash regardless of API version?

## #8 - 05/13/2020 08:12 PM - Lucas Di Pentima

Thanks, yes! `WB2` will need some fixes, should I add a subtask about it assigned to me?

LGTM.

## #9 - 05/13/2020 08:39 PM - Tom Clegg

- Assigned To changed from Tom Clegg to Lucas Di Pentima

## #10 - 05/20/2020 03:44 PM - Tom Clegg

- Status changed from In Progress to Resolved

## #11 - 10/07/2020 02:08 AM - Peter Amstutz

- Release set to 25

## #12 - 10/07/2020 02:10 AM - Peter Amstutz

- Release changed from 25 to 36

## #13 - 10/07/2020 02:14 AM - Peter Amstutz

- Release changed from 36 to 25