

Arvados - Story #13579

Document storage classes for sysadmin

06/06/2018 03:06 PM - Tom Morris

Status: Resolved	Start date: 06/13/2018
Priority: Normal	Due date:
Assigned To: Peter Amstutz	% Done: 100%
Category:	Estimated time: 0.00 hour
Target version: 2018-06-20 Sprint	
Description	
Subtasks:	
Task # 13585: Review 13579-keepstore-config-doc	Resolved
Related issues:	
Related to Arvados - Bug #13623: [Documentation] Update install docs to ackno...	New

Associated revisions

Revision af3bfbb6 - 06/18/2018 07:27 PM - Peter Amstutz

Merge branch '13579-keepstore-config-doc' refs #13579 refs #13580

Arvados-DCO-1.1-Signed-off-by: Peter Amstutz <pamstutz@veritasgenetics.com>

History

#1 - 06/06/2018 03:20 PM - Tom Morris

- Parent task deleted (#11184)

#2 - 06/06/2018 03:20 PM - Tom Morris

- Tracker changed from Task to Story

#3 - 06/06/2018 03:22 PM - Peter Amstutz

- Assigned To set to Peter Amstutz

#4 - 06/11/2018 02:01 PM - Peter Amstutz

So, our keepstore install documentation doesn't document using keepstore config files (it still describes the old way which is using command line flags). However StorageClasses can only be specified in the config file. So in order to document how the admin should configure StorageClasses without terribly confusing the reader is to first update the keepstore install documentation for config files.

#5 - 06/11/2018 02:20 PM - Tom Clegg

[8de0cca20](#) is an interesting experiment in getting the docs closer to the code, but I'm not sure whether it's really better in the end.

Thoughts

- Is there some way we can use traditional godoc- and editor-friendly struct field comments to generate docs?
- What's the most important place to offer the detailed information -- "keepstore -help", or doc.arvados.org, or...?

If we go with struct tags, instead of sort-of-parsing the struct tags with `strings.Index(string(f.Tag), "doc:")` etc., we can probably use `f.Tag.Get("doc")` ... <https://golang.org/pkg/reflect/#StructTag.Get> ... but (I'm guessing) only if we use proper struct tag syntax, ``doc:"..."``, which presumably we should do anyway.

#6 - 06/11/2018 02:30 PM - Peter Amstutz

Tom Clegg wrote:

[8de0cca20](#) is an interesting experiment in getting the docs closer to the code, but I'm not sure whether it's really better in the end.

Thoughts

- Is there some way we can use traditional godoc- and editor-friendly struct field comments to generate docs?

So the problem is, unlike struct Tags, comments aren't available at runtime, so we would need a separate tool to parse the file and generate the documentation, which seems a lot more complicated. I suppose it's possible there's already a tool out there that does something like this, I can do some googling.

- What's the most important place to offer the detailed information -- "keepstore -help", or doc.arvados.org, or...?

So the goal here actually is to offer detailed documentation on doc.arvados.org, but in a way that minimizes the friction of keeping it up to date (ideally it would even be part of the documentation generation process).

If we go with struct tags, instead of sort-of-parsing the struct tags with strings.Index(string(f.Tag), "doc:") etc., we can probably use f.Tag.Get("doc") ... <https://golang.org/pkg/reflect/#StructTag.Get> ... but (I'm guessing) only if we use proper struct tag syntax, `doc:"..."`, which presumably we should do anyway.

I tried that, it didn't like newlines in the tag value, which is why I did it that way.

#7 - 06/11/2018 07:31 PM - Nico César

a handy copy and paste oneliner is:

```
awk '/MemTotal/ { printf "%d\n", ($2 / (88 * 1024)) }' /proc/meminfo
```

this could be added in the MaxBuffers description

#8 - 06/11/2018 07:52 PM - Tom Clegg

Peter Amstutz wrote:

So the problem is, unlike struct Tags, comments aren't available at runtime, so we would need a separate tool to parse the file and generate the documentation, which seems a lot more complicated. I suppose it's possible there's already a tool out there that does something like this, I can do some googling.

Maybe something like <https://github.com/robertkrimen/godocdown> ?

So the goal here actually is to offer detailed documentation on doc.arvados.org, but in a way that minimizes the friction of keeping it up to date (ideally it would even be part of the documentation generation process).

If the goal is doc.arvados.org then the runtime introspection stuff seems like an unnecessary layer of challenges. Comments→docs is the way godoc etc. work, and seems more direct.

```
f.Tag.Get()
```

I tried that, it didn't like newlines in the tag value, which is why I did it that way.

Conflicting with standard tooling is a red flag. Sounds to me like struct tags are not a good fit here.

#9 - 06/12/2018 07:11 PM - Peter Amstutz

13579-keepstore-config-doc @ [296ac256c898c145f7e4566d86d5f09f4e966b43](https://github.com/Arvados/keepstore/pull/13579)

Rebased and removed changes to keepstore.

#10 - 06/13/2018 09:18 PM - Tom Morris

- Status changed from New to In Progress

- Story points set to 1.0

doc/install/install-keepstore.html.textile.liquid

- need a discussion of how to decide on number and size of Keepstore servers - redundancy, network & disk I/O bandwidth, disk capacity, RAM, etc
- I don't see any explanation of "pull operations" ref. PullWorkers

Notes on storage management

- The keep-balance service ~~service~~
- A block which is in the trash ~~has been moved out of the way and~~ is no longer accessible by read requests, but has not yet been permanently deleted.

Configure storage volumes

- supported volume types would scan better as a list
- local inline vs Azure & S3 on their own pages feels asymmetric
- I don't think this is true:

```
# When using spinning disks where storage partitions map 1:1 to  
# physical disks that are dedicated to Keepstore, enabling this may  
# reduce contention and improve throughput by minimizing seeks.
```

modern disk controllers know how to reorder requests to minimize seek times, but they can only do that when they have visibility into the entire set of pending requests.

doc/install/configure-s3-object-storage

- As an alternative to local and network-attached POSIX filesystems,- Keepstore can store data in object storage compatible with the S3 AP
- What are possible values for Endpoint ?
- Split GCP example YAML file into separate code block and give it a heading so that it's more visible

doc/install/configure-azure-blob-storage

- ~~As an alternative to local and network-attached POSIX filesystems,~~ Keepstore can store data in one or more Azure Storage containers.
- 10 minutes seems like a **very** long default timeout for operations

#11 - 06/13/2018 10:16 PM - Peter Amstutz

Tom Morris wrote:

doc/install/install-keepstore.html.textile.liquid

- need a discussion of how to decide on number and size of Keepstore servers - redundancy, network & disk I/O bandwidth, disk capacity, RAM, etc

Agree, but Ward or Nico would be better qualified to write this.

Configure storage volumes

- supported volume types would scan better as a list
- local inline vs Azure & S3 on their own pages feels asymmetric

Sure, should move the "local" config to its own page then?

modern disk controllers know how to reorder requests to minimize seek times, but they can only do that when they have visibility into the entire set of pending requests.

That was partly copied from some earlier text. I'll just make it more vague, like, may be useful "if you are experiencing contention with simultaneous requests to the same volume".

doc/install/configure-s3-object-storage

- As an alternative to local and network-attached POSIX filesystems,- Keepstore can store data in object storage compatible with the S3 AP
- What are possible values for Endpoint ?

It's relevant for GCP / RADOS but not for S3, will try to tweak the documentation for that.

- 10 minutes seems like a **very** long default timeout for operations

I think that's what keepstore's default is but I agree that is excessive.

#12 - 06/14/2018 01:49 PM - Tom Clegg

This seems out of scope, but just in case you want to deal with it anyway... The keepstore package enables keepstore as a systemd service, which the docs should acknowledge (like <https://doc.arvados.org/install/install-ws.html?>). On debian and ubuntu I suspect this version (still) leaves you with a systemd service and a runit service competing for port 25107.

Rather than maintaining its own differently-formatted version of the install docs (which already seems to be out of sync) perhaps the keepstore -help usage message should offer a link to the docs.

Nit: the word "actual" seems superfluous in "The actual directory that will be used".

#13 - 06/14/2018 02:32 PM - Tom Clegg

- Related to Bug #13623: [Documentation] Update install docs to acknowledge systemd added

#14 - 06/14/2018 07:32 PM - Peter Amstutz

13579-keepstore-config-doc now @ [679d5507d6c390fd03e5c71300001f2696b1f5b1](#)

Incorporated comments from above. Added documentation page about storage classes to "Admin" section (which is what this ticket was originally about).

#15 - 06/14/2018 10:42 PM - Nico César

added [d9c1f53cd](#) 13579-keepstore-config-doc for Peter to improve.

#16 - 06/18/2018 03:09 PM - Peter Amstutz

13579-keepstore-config-doc @ [18e820bdfc2598ec1901fd52fd80c8dae4f55c32](#)

Covers [#13580](#) because they are closely related. Added section about setting storage classes to user guide. Admin page links to it.

#17 - 06/18/2018 03:47 PM - Tom Morris

LGTM

#18 - 06/20/2018 03:29 PM - Peter Amstutz

- Status changed from In Progress to Resolved

#19 - 07/23/2018 06:41 PM - Tom Morris

- Release set to 13