Arvados - Feature #16625

Support creating VMs from Azure "image" resource not just bare VHD

07/22/2020 08:59 PM - Peter Amstutz

Status:	Resolved	**Start date:**	08/17/2020
Priority:	Normal	**Due date:**	
Assigned To:	Ward Vandewege	**% Done:**	100%
Category:	Crunch	**Estimated time:**	0.00 hour
Target version:	2020-08-26 Sprint		

Description

Packer warns when generating VHDs, with:

==> azure-arm: Warning: You are using Azure Packer Builder to create VHDs which is being deprecated, consider using Managed Images. Learn more http://aka.ms/packermanagedimage

Subtasks:

Task # 16694: review 16625-add-azure-managed-image-support	**Resolved**

Related issues:

Related to Arvados - Feature #16739: [a-d-c] throttle option for concurrent V...	**Resolved**	**08/31/2020**

Associated revisions

Revision 6e1e0086 - 08/03/2020 07:14 PM - medcelerate

16625: Added fix to prevent ruby from failing when @ appears in the user for the database config.

Added escaping to password and dbname.

refs #16625

Arvados-DCO-1.1-Signed-off-by: Evan Clark <evan.clark.professional@gmail.com>

Revision a1fca09b - 08/03/2020 07:27 PM - medcelerate

16625: Added fix to prevent ruby from failing when @ appears in the user for the database config.

Added escaping to password and dbname.

refs #16625

Arvados-DCO-1.1-Signed-off-by: Evan Clark <evan.clark.professional@gmail.com>

Revision 4da1edec - 08/03/2020 10:06 PM - medcelerate

16625: Added fix to prevent ruby from failing when @ appears in the user for the database config.

Added escaping to password and dbname.

refs #16625

Arvados-DCO-1.1-Signed-off-by: Evan Clark <evan.clark.professional@gmail.com>

Revision 9df537c2 - 08/20/2020 02:29 PM - Ward Vandewege

Merge branch '16625-add-azure-managed-image-support'

refs #16625

Arvados-DCO-1.1-Signed-off-by: Ward Vandewege <ward@curii.com>

Revision 66c7c739 - 08/20/2020 07:39 PM - Ward Vandewege

Merge branch '16625-add-azure-managed-image-support'

refs #16625

Arvados-DCO-1.1-Signed-off-by: Ward Vandewege <ward@curii.com>

History

#1 - 07/24/2020 04:06 PM - Evan Clark

After discussing with the azure system architects azure has migrated to a managed disk approach to handling vm images. The disks are still vhds under the hood, however they no longer provide a blob uri directly to the vhd. The workaround we have discovered is to download the vhd from the generalized managed disk using azure storage manager and then uploading it to blob as a page blob. Page blobs are what are used for random read/write which are required for vhds. [[https://docs.microsoft.com/en-us/azure/storage/blobs/storage-blob-pageblob-overview?tabs=dotnet]]

#2 - 07/27/2020 04:35 PM - Evan Clark

Something else to note, it appears phusion passenger only accepts the url encoded variant and not the one double quoted. It caused passenger to start writing a large number of error html files to disk causing the disk to fill up.

```
Error ID: 7c06e71c
Error details saved to: /tmp/passenger-error-JVqYL4.html

[ E 2020-07-27 16:33:12.6542 92625/T9 age/Cor/Con/CheckoutSession.cpp:276 ]: [Client 1-74926] Cannot checkout session because a spawning error occurred. The identifier of the error is 7c06e71c. Please see earlier logs for details about the error.
[ E 2020-07-27 16:33:12.6542 92625/Tf age/Cor/Con/CheckoutSession.cpp:276 ]: [Client 4-74925] Cannot checkout session because a spawning error occurred. The identifier of the error is 7c06e71c. Please see earlier logs for details about the error.
App 62662 output: Error: The application encountered the following error: the scheme postgresql does not accept registry part: arvados@mscic-postgresql:ia36yJxyUDq3h3QHeCt6lTZ4VUlfd3x8@mscic-postgresql.postgres.database.azure.com: (or bad hostname?) (URI::InvalidURIError)
```

#3 - 07/27/2020 05:02 PM - Evan Clark

I have created a fork of arvados with a fix to the above issue in regards to the passenger config. I would also like to note as a current workaround if you start all the other services with the double quoted method you can then edit the config file, replace the @ with %40 and restart nginx and the api server will run properly.

Commit:
[[https://github.com/medcelerate/arvados/commit/dd281e202b977a417000f410306ca4e2a431d87b]]

#4 - 08/03/2020 04:57 PM - Peter Amstutz

- Release set to 25

#5 - 08/12/2020 03:54 PM - Ward Vandewege

- Assigned To set to Ward Vandewege

#6 - 08/13/2020 07:59 PM - Ward Vandewege

- Description updated

#7 - 08/13/2020 07:59 PM - Ward Vandewege

- Status changed from New to In Progress

#8 - 08/15/2020 08:32 PM - Ward Vandewege

First version ready for review at 24f3823f6e96c60025cbde15c3cb94557f3d0bec on branch 16625-add-azure-managed-image-support. I haven't added a test yet for the new feature.

CI tests are at https://ci.arvados.org/view/Developer/job/developer-run-tests/2005/. 2004 too, but that one failed with - apparently - a flaky test for a-d-c.

#9 - 08/17/2020 07:08 PM - Tom Clegg

In manageDisks():

The paging loop looks suspect: it calls response.Next() but then throws away the results and starts a new query on the next loop iteration. Should probably look more like the loop in manageNics()?

This should use regexp.QuoteMeta(az.namePrefix):

```
re := regexp.MustCompile(`^` + az.namePrefix + `.*-os$`)
```

Should use const compute.DiskStateUnattached instead of literal string "Unattached".

Worth checking for disk.Name==nil before dereferencing. The Azure SDK has surprised us with nils before, like #15007.

There seem to be several ways of checking whether to use the old/new code path: az.azconfig.StorageAccount!="", az.blobcont!=nil (which assumes az.azconfig.BlobContainer!=""), and imageID=~urlRe. Suggest

- setup() should check StorageAccount!="" && BlobContainer!="" (perhaps warn if only one of them is set, since that would be useless?)
- Create() should set blobname only in the unmanaged case, and check blobname!="" when deciding whether to clean up (it looks like the current code will try to delete the vhd blob if a storage container is configured, even if the failed VM didn't invoke the vhd blob code, which is probably harmless but a bit surprising)
- Create() should error out if blobcont==nil on the vhd path ("can't configure image URL unless you have configured BlobContainer and StorageAccount" seems to be the rule here)

On that note, is an analogous cleanup step needed after a failed create call in the non-vhd case, or does Azure automatically clean it up, or do we just wait for manageDisks() to clean it up? The answer could go in a comment there.

Func comment should be spelled manageDisks, not ManageDisks (manageNics and manageBlobs comments have the same mismatch, probably missed when changing funcs from public to private)

....and should say "unattached", not "available".

In setup():

This could be written more simply as "for disk := range az.deleteDisk {" (same goes for deleteNIC and deleteBlob nearby)

```
                        for {
                                disk, ok := <-az.deleteDisk
                                if !ok {
                                        return
                                }
```

Suggest mentioning in config.default.yml comments that the "managed" way is preferred/normal, and unmanaged is deprecated ("use this only if ...?")

Suggest updating the minimal example in doc/install/install-dispatch-cloud to use a name instead of URL

It seems like everyone can/should convert to the new way, IOW the only reason to support the old way is to keep things running while updating config/automation, is that right? If so, perhaps "using deprecated VHD image" should be at loglevel=warn instead of info? And either way I think it would be good to give more of a hint about how to migrate to the new way (link to doc page?).

Should the "using managed image" log message be level=debug (or removed)?

Nit: I think the timestamp comparisons in manageDisks() would be a bit simpler if we did something like this

```
-timestamp := time.Now()
+threshold := time.Now().Add(-az.azconfig.DeleteDanglingResourcesAfter.Duration())
...
if ... && d.DiskProperties.TimeCreated.ToTime().Before(threshold) {
```

...also, logging the actual creation time is probably more useful than logging the age in seconds at the time the manageDisks() func started.

Looks like Stop() should add close(az.deleteDisk)

https://docs.microsoft.com/en-us/azure/virtual-machines/windows/create-vm-generalized-managed mentions "one managed image supports up to 20 simultaneous deployments." Does this mean starting 100 VMs at once will go badly with this approach? I wonder if we should implement a configurable throttle to avoid this. (They suggest using Shared Image Galleries for better scaling, but automating that seems unappealing at this point.)

#10 - 08/18/2020 03:24 PM - Ward Vandewege

Tom Clegg wrote:

> In manageDisks():
>
> The paging loop looks suspect: it calls response.Next() but then throws away the results and starts a new query on the next loop iteration. Should probably look more like the loop in manageNics()?

OK, fixed, it's a little more subtle because the interface for nics is a bit different.

> This should use regexp.QuoteMeta(az.namePrefix):
>
> [...]

Fixed.

> Should use const compute.DiskStateUnattached instead of literal string "Unattached".

Fixed, it's compute.Unattached apparently.

> Worth checking for disk.Name==nil before dereferencing. The Azure SDK has surprised us with nils before, like #15007.

OK, added.

> There seem to be several ways of checking whether to use the old/new code path: az.azconfig.StorageAccount!="", az.blobcont!=nil (which assumes az.azconfig.BlobContainer!=""), and imageID=~urlRe. Suggest
>
> - setup() should check StorageAccount!="" && BlobContainer!="" (perhaps warn if only one of them is set, since that would be useless?)

Sure, done.

> - Create() should set blobname only in the unmanaged case

Ok, I've made that change.

> , and check blobname!="" when deciding whether to clean up (it looks like the current code will try to delete the vhd blob if a storage container is configured, even if the failed VM didn't invoke the vhd blob code, which is probably harmless but a bit surprising)

The cleanup code was already gated on az.blobcont != nil, and it must remain that way, because otherwise

`_, delerr := az.blobcont.GetBlobReference(blobname).DeleteIfExists(nil)`

can be a nil pointer dereference. blobname is set when the imageID matches the URI pattern, but that doesn't guarantee that az.blobcont is set.

> - Create() should error out if blobcont==nil on the vhd path ("can't configure image URL unless you have configured BlobContainer and StorageAccount" seems to be the rule here)

Yes, added. I would prefer to check this earlier, during a-d-c startup (like we discussed, it would be very nice to have a hook into each driver for config testing).

> On that note, is an analogous cleanup step needed after a failed create call in the non-vhd case, or does Azure automatically clean it up, or do we just wait for manageDisks() to clean it up? The answer could go in a comment there.

I was leaving it to manageDisks() and have added a comment. That whole section seems of dubious utility, since we already garbage collect.

> Func comment should be spelled manageDisks, not ManageDisks (manageNics and manageBlobs comments have the same mismatch, probably missed when changing funcs from public to private)

Yep, fixed them all.

> ....and should say "unattached", not "available".

Right, also fixed.

> In setup():
>
> This could be written more simply as "for disk := range az.deleteDisk {" (same goes for deleteNIC and deleteBlob nearby)

All fixed.

> Suggest mentioning in config.default.yml comments that the "managed" way is preferred/normal, and unmanaged is deprecated ("use this only if ...?")

I've added '(deprecated)' behind each mention of 'unmanaged disks' in config.default.yml. I'll leave it to Azure themselves to provide more information.

> Suggest updating the minimal example in doc/install/install-dispatch-cloud to use a name instead of URL

Ah, yes, I had forgotten to do that, I've documented both ways there now.

> It seems like everyone can/should convert to the new way, IOW the only reason to support the old way is to keep things running while updating config/automation, is that right? If so, perhaps "using deprecated VHD image" should be at loglevel=warn instead of info?

Done.

> And either way I think it would be good to give more of a hint about how to migrate to the new way (link to doc page?).

Done, though I only linked to https://doc.arvados.org to avoid future link breakage (they can search for 'managed' or 'unmanaged').

> Should the "using managed image" log message be level=debug (or removed)?

Yeah I removed it.

> Nit: I think the timestamp comparisons in manageDisks() would be a bit simpler if we did something like this

Ah, nice, changed.

> ...also, logging the actual creation time is probably more useful than logging the age in seconds at the time the manageDisks() func started.

Done.

> Looks like Stop() should add close(az.deleteDisk)

Oh, yes, fixed.

> https://docs.microsoft.com/en-us/azure/virtual-machines/windows/create-vm-generalized-managed mentions "one managed image supports up to 20 simultaneous deployments." Does this mean starting 100 VMs at once will go badly with this approach? I wonder if we should implement a configurable throttle to avoid this. (They suggest using Shared Image Galleries for better scaling, but automating that seems unappealing at this point.)

I saw that. I agree we're going to need to do something here. Even just a few seconds delay between VM requests may be sufficient, though having a throttle that defaults to 20 simultaneous starts on Azure would be perfect. Should that be a separate ticket?

Ready for another look at c76e5bc93b066d93a668c4e00b52aa550028e1f4 on branch 16625-add-azure-managed-image-support. Tests are at https://ci.arvados.org/view/Developer/job/developer-run-tests/2012/

#11 - 08/19/2020 09:15 PM - Tom Clegg

> > check blobname!="" when deciding whether to clean up
>
> cleanup code was already gated on az.blobcont != nil, and it must remain that way, because ... nil pointer

Except that if az.blobcont == nil then we would have errored out earlier, instead of setting blobname to a non-empty value.

This looks like it will error out if the imageID is a "new" kind, and StorageAccount/BlobContainer are also [still] configured. That kind of config isn't strictly necessary, but it seems like there's no reason it shouldn't work, and the resulting error message would definitely be misleading:

```
	if re.MatchString(string(imageID)) && az.blobcont != nil {
		// (old way)
	} else if az.blobcont == nil {
		// (new way)
	} else {
		return nil, wrapAzureError(errors.New("Invalid configuration: can't configure unmanaged image U
RL without StorageAccount and BlobContainer"))
	}
```

I think you meant

```
	if re.Match(...) {
		if az.blobcont == nil {
			// error out
		}
		// (old way)
	} else {
		// (new way)
	}
```

> I was leaving it to manageDisks() and have added a comment. That whole section seems of dubious utility, since we already garbage collect.

Looks like this was added in #14844 because without it "an unbounded number of new unused nics and blobs pile up during times when VMs can't be created and the dispatcher keeps retrying."

This seems less likely to be a problem with the new managed disk method, since Azure understands that the VM is the only reason for the disk to exist.

> having a throttle that defaults to 20 simultaneous starts on Azure would be perfect. Should that be a separate ticket?

Not sure. Given the way Azure docs discuss it, I'm inclined to not call this "done" until we have some way to acknowledge/address it. But you could merge this, and do a follow-up branch on the same ticket.

The rest LGTM, thanks.

#12 - 08/19/2020 09:47 PM - Ward Vandewege

Tom Clegg wrote:

> > > check blobname!="" when deciding whether to clean up
>
> > cleanup code was already gated on az.blobcont != nil, and it must remain that way, because ... nil pointer
>
> Except that if az.blobcont == nil then we would have errored out earlier, instead of setting blobname to a non-empty value.

Ah, you're right, I had tested that before adding the error path, earlier. OK, I've changed it to checking for an empty blobname now.

> This looks like it will error out if the imageID is a "new" kind, and StorageAccount/BlobContainer are also [still] configured. That kind of config isn't strictly necessary, but it seems like there's no reason it shouldn't work, and the resulting error message would definitely be misleading: [...]

Yeah, changed, thanks.

> > I was leaving it to manageDisks() and have added a comment. That whole section seems of dubious utility, since we already garbage collect.
>
> Looks like this was added in #14844 because without it "an unbounded number of new unused nics and blobs pile up during times when VMs can't be created and the dispatcher keeps retrying."
>
> This seems less likely to be a problem with the new managed disk method, since Azure understands that the VM is the only reason for the disk to exist.

Ah. That makes sense. I've added a comment along those lines for future reference. Like you say it's probably less of an issue with disks (and there's no obvious way to get a reference to the disk object that may have been created as part of a failed VM creation), so I'll leave it like that.

> > having a throttle that defaults to 20 simultaneous starts on Azure would be perfect. Should that be a separate ticket?
>
> Not sure. Given the way Azure docs discuss it, I'm inclined to not call this "done" until we have some way to acknowledge/address it. But you could merge this, and do a follow-up branch on the same ticket.

Ok, sounds good, I'll look at that throttle.

Changes at 91ce664427d1ba915230dde3bc14163b57342af6 on branch 16625-add-azure-managed-image-support

#13 - 08/20/2020 02:24 PM - Tom Clegg

LGTM @ 91ce66442, thanks

#14 - 08/20/2020 03:57 PM - Ward Vandewege

> Ok, sounds good, I'll look at that throttle.

I tried the shared image gallery thing, that seems to work.

After discussion, plan of action:

1) add a few extra config fields for shared image galleries to the Azure driver (name of shared image gallery + version of the image)

This change is ready for review in df279041b8f688270b8c6791d035ffbcd40688c3 on branch 16625-add-azure-managed-image-support. Tests at https://ci.arvados.org/view/Developer/job/developer-run-tests/2017/

2) add an a-d-c throttle field (not at the driver level) for concurrent create requests, because it may be useful for other providers, too. Reuse the quota limit mechanism in a-d-c. Default the throttle to 20 for Azure in our config example.

#15 - 08/20/2020 07:29 PM - Tom Clegg

> This change is ready for review in df279041b8f688270b8c6791d035ffbcd40688c3 on branch 16625-add-azure-managed-image-support. Tests at https://ci.arvados.org/view/Developer/job/developer-run-tests/2017/

LGTM except tests are panicking -- suspect adding a make(chan compute.Disk) in GetInstanceSet() in azure_test.go will fix this.

#16 - 08/20/2020 07:40 PM - Ward Vandewege

Tom Clegg wrote:

> > This change is ready for review in df279041b8f688270b8c6791d035ffbcd40688c3 on branch 16625-add-azure-managed-image-support. Tests at https://ci.arvados.org/view/Developer/job/developer-run-tests/2017/
>
> LGTM except tests are panicking -- suspect adding a make(chan compute.Disk) in GetInstanceSet() in azure_test.go will fix this.

Thanks, I fixed the tests that way and merged.

#17 - 08/21/2020 03:39 PM - Ward Vandewege

- Related to Feature #16739: [a-d-c] throttle option for concurrent VM create requests added

#18 - 08/21/2020 03:40 PM - Ward Vandewege

- Status changed from In Progress to Resolved

Ward Vandewege wrote:

> 2) add an a-d-c throttle field (not at the driver level) for concurrent create requests, because it may be useful for other providers, too. Reuse the quota limit mechanism in a-d-c. Default the throttle to 20 for Azure in our config example.

I created a new ticket #16739 for this work.