Arvados - Feature #14200

[API] Reduce privilege exposure via API tokens in multi-cluster workflows

09/12/2018 04:43 PM - Peter Amstutz

Status:	New	**Start date:**	
Priority:	Normal	**Due date:**	
Assigned To:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:	To Be Groomed		

Description

A container running on cluster A might have inputs located on cluster B. Therefore, it must have a runtime token capable of authorizing API calls to cluster B. However, the container does not need all of the privileges on cluster B that it needs on cluster A: for example, it does not need to create a log collection on cluster B.

Proposal:

- Additional "cluster_scope" column restricting which clusters should accept it? If cluster B tries do use with cluster C, cluster A will tell cluster C not to use it.
- "cluster_ scope" could also instruct remote clusters to limit their scope (so token used on cluster C still only has access to read-only collections).
  - Proposed format: {cluster1: [scope1, scope2], cluster2: [scope3, scope4]}

Related issues:

Related to Arvados - Feature #14262: [Controller] Specify runtime_token when ...	**Resolved**	**10/24/2018**

History

#1 - 09/12/2018 07:00 PM - Peter Amstutz

- Description updated

#2 - 09/12/2018 07:50 PM - Tom Clegg

It would be good to have a way to expire the token when the container ends.

Couldn't the existing "salt" mechanism be used when cluster B wants to give cluster C a cluster-C-only token?

As with the single-cluster case, using scopes to limit access to input collections sounds good, but how do we express "permitted to create and update log/output/temp collections"?

#3 - 09/14/2018 01:38 PM - Peter Amstutz

Tom Clegg wrote:

> It would be good to have a way to expire the token when the container ends.

Thoughts:

- Cluster A can set a default token expiration date
- Cluster A can poll cluster B to see when the container request is finalized (not sure what component would do this)
- Cluster B can push cluster A to expire the token when the container request is finalized

The 2nd and 3rd bullet points assumes non-malicious cooperation on the part of cluster B (but that's probably fine, since we're already trusting it to do compute for us).

> Couldn't the existing "salt" mechanism be used when cluster B wants to give cluster C a cluster-C-only token?

Yes, this would prevent cluster C from using the token to access other clusters, which seems fine and desirable. However, to prevent cluster B from using the token it on clusters other than those intended, I think we still want per-cluster scope.

> As with the single-cluster case, using scopes to limit access to input collections sounds good, but how do we express "permitted to create and update log/output/temp collections"?

The assumption here is that the token scope limits what can be done on cluster A and C, but not cluster B.

#4 - 09/26/2018 08:09 PM - Tom Clegg

- Subject changed from [API] can delegate permissions to remote container requests to [API] Reduce privilege exposure via API tokens in multi-cluster workflows

- Description updated

#5 - 10/01/2018 02:15 PM - Peter Amstutz

- Related to Feature #14262: [Controller] Specify runtime_token when creating container requests on a remote cluster added