

## Arvados - Bug #18562

### [api] should not change the preemptible flag across the board

12/07/2021 04:45 PM - Ward Vandewege

<b>Status:</b>	Resolved	
<b>Priority:</b>	Normal	
<b>Assigned To:</b>	Tom Clegg	
<b>Category:</b>		
<b>Target version:</b>	2022-01-05 sprint	
<b>Description</b>		
The RailsAPI changes applies the default config value for the preemptible flag whenever a container record is loaded. This causes problems when the config value is changed after a container record has been created and before it is done.		
Case in point, if a-d-c tries to cancel (sends State=Cancelled) a container that was created when the config said `UsePreemptible: true` and meanwhile the config is changed to `UsePreemptible: false`, RailsAPI fails validation and returns a 422 on the cancel request.		
RailsAPI should only set/change the `Preemptible` flag in one place, maybe when the container is created or when it is committed.		
<b>Subtasks:</b>		
Task # 18567: Review 18562-preemptible		<b>Resolved</b>
<b>Related issues:</b>		
Related to Arvados - Feature #18596: Config option to enable preemptible vari...	<b>Resolved</b>	<b>03/21/2022</b>
Related to Arvados Epics - Idea #18179: Better spot instance support	<b>Resolved</b>	<b>03/01/2022</b> <b>06/30/2024</b>

#### Associated revisions

##### Revision 00f1f057 - 12/30/2021 07:00 PM - Tom Clegg

Merge branch '18562-preemptible'

fixes #18562

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

##### Revision 00f1f05789316936db75b4723b1c3d99196c252a - 12/30/2021 07:00 PM - Tom Clegg

Merge branch '18562-preemptible'

fixes #18562

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

#### History

##### #1 - 12/07/2021 04:54 PM - Ward Vandewege

- Target version changed from 2021-12-08 sprint to 2022-01-05 sprint

- Description updated

##### #2 - 12/08/2021 04:25 PM - Peter Amstutz

- Assigned To set to Tom Clegg

##### #3 - 12/16/2021 06:27 PM - Tom Clegg

I think the behavior we want is

- If there are no configured InstanceTypes with Preemptible: true, it is an error for a caller to specify preemptible=true when creating a CR, or change preemptible from false to true during an update.
- If config has UsePreemptibleInstances: true and some InstanceTypes with Preemptible: true, a newly created container request with a non-empty requesting\_container\_uuid has preemptible set to true, even if the caller specified false.
- In all other cases, preemptible is set to whatever the client specifies.
- Default config is UsePreemptibleInstances: true, i.e., if you configure some preemptible instance types, arvados will automatically use preemptible instances for child containers. If you don't, it won't.

#### #4 - 12/16/2021 06:27 PM - Tom Clegg

- Status changed from New to In Progress

#### #5 - 12/16/2021 08:34 PM - Peter Amstutz

Tom Clegg wrote:

- If config has UsePreemptibleInstances: true *and* some InstanceTypes with Preemptible: true, a newly created container request with a non-empty requesting\_container\_uuid has preemptible set to true, even if the caller specified false.
- In all other cases, preemptible is set to whatever the client specifies.

Could we tweak that to

- If config has InstanceTypes with Preemptible: true, then a newly created container request with a non-empty requesting\_container\_uuid and empty/unspecified preemptible has preemptible on the container set to the value of UsePreemptibleInstances.
- In all other cases, preemptible is set to whatever the client specifies or left unchanged in unspecified.

#### #6 - 12/16/2021 11:04 PM - Ward Vandewege

I would very much like a shortcut that makes all node types available as both preemptible or not, if no node types have an explicit preemptible flag set in the config. The current situation is very tedious, having to duplicate all these records.

I am not aware of node types that don't exist as spot.

#### #7 - 12/17/2021 03:40 PM - Tom Clegg

Peter Amstutz wrote:

Could we tweak that to

- [...] empty/unspecified preemptible has preemptible on the container set to the value of UsePreemptibleInstances.

So, change the meaning of UsePreemptibleInstances from "override user's choice" to "configure default choice"?

Currently, a-c-r never specifies anything, so we could update controller to propagate null/missing, and interpret null in rails the way it currently interprets false, and the end result would be the same behavior we have now. In future, when a-c-r lets the user choose, it would still be able to choose "I don't know, whatever cluster config says".

The configurable default could be useful, but IMO it would be better to implement by having a-c-r look up the default in the exported config. For one thing, it can tell the user what the default is.

(I also don't love making tri-state fields with bool pointers etc.)

#### #8 - 12/17/2021 04:08 PM - Peter Amstutz

Tom Clegg wrote:

Peter Amstutz wrote:

Could we tweak that to

- [...] empty/unspecified preemptible has preemptible on the container set to the value of UsePreemptibleInstances.

So, change the meaning of UsePreemptibleInstances from "override user's choice" to "configure default choice"?

Currently, a-c-r never specifies anything, so we could update controller to propagate null/missing, and interpret null in rails the way it currently interprets false, and the end result would be the same behavior we have now. In future, when a-c-r lets the user choose, it would still be able to choose "I don't know, whatever cluster config says".

The configurable default could be useful, but IMO it would be better to implement by having a-c-r look up the default in the exported config. For one thing, it can tell the user what the default is.

- If config has InstanceTypes with Preemptible: true, then a newly created container request with a non-empty requesting\_container\_uuid and value of "false" for preemptible, has preemptible on the container set to the value of UsePreemptibleInstances.
- In all other cases, preemptible is set to whatever the client specifies or left unchanged if unspecified.

(I also don't love making tri-state fields with bool pointers etc.)

That's definitely an impedance mismatch between Go and Python/Ruby.

How do you do "left unchanged if unspecified"? If "false" is treated as default/unspecified then it's impossible to set a preemptible flag that's been set to true back to false?

#### #9 - 12/17/2021 09:59 PM - Tom Clegg

How do you do "left unchanged if unspecified"? If "false" is treated as default/unspecified then it's impossible to set a preemptible flag that's been set to true back to false?

Right, I just don't think we need a value that means "unspecified". It can just be true or false, and you get what you ask for, unless the "force preemptible child containers even if client specifies false" config is enabled.

#### #10 - 12/20/2021 07:42 PM - Tom Clegg

- Related to Feature #18596: Config option to enable preemptible variants of all instance types added

#### #11 - 12/23/2021 03:06 PM - Tom Clegg

18562-preemptible @ [534b7df510b99923a7dc273a2f8cacfd0c599800](#) -- [developer-run-tests: #2853](#) [icon?job=developer-run-tests&build=2853](#)

- Do not automatically set preemptible=true if there are no preemptible instance types available.
- Do not automatically set preemptible=true on a container request that has already reached "Committed" state with preemptible=false.
- If a container request reaches "Committed" state with preemptible=true, and then all preemptible instance types are removed from config, do not reject subsequent updates to the container request. (Was: refuse all updates, even cancel, if config UsePreemptibleInstances changes to false after a CR is created with preemptible=true.)
- UsePreemptibleInstances=true means enable preemptible=true on child CRs *if any preemptible instance types are configured*. (Was: set preemptible flag even if that means the container will never run because there are no preemptible instance types configured.)
- UsePreemptibleInstances=false means leave preemptible flag alone, do whatever the client asks. (Was: impossible to set preemptible=true)

The [doc page](#) says "To use preemptible instances, set UsePreemptibleInstances: true and add entries to InstanceTypes with Preemptible: true to config.yml." Now that the default for UsePreemptibleInstances is true, should we remove "set UsePreemptibleInstances: true and" or change it to "make sure UsePreemptibleInstances isn't set to false"? Or just leave it the way it is?

#### #12 - 12/23/2021 07:05 PM - Ward Vandewege

Tom Clegg wrote:

18562-preemptible @ [534b7df510b99923a7dc273a2f8cacfd0c599800](#) -- [developer-run-tests: #2853](#) [icon?job=developer-run-tests&build=2853](#)

- Do not automatically set preemptible=true if there are no preemptible instance types available.
- Do not automatically set preemptible=true on a container request that has already reached "Committed" state with preemptible=false.
- If a container request reaches "Committed" state with preemptible=true, and then all preemptible instance types are removed from config, do not reject subsequent updates to the container request. (Was: refuse all updates, even cancel, if config UsePreemptibleInstances changes to false after a CR is created with preemptible=true.)
- UsePreemptibleInstances=true means enable preemptible=true on child CRs *if any preemptible instance types are configured*. (Was: set preemptible flag even if that means the container will never run because there are no preemptible instance types configured.)
- UsePreemptibleInstances=false means leave preemptible flag alone, do whatever the client asks. (Was: impossible to set preemptible=true)

This is a bit weird. As it is now, the name of the flag is misleading. Maybe we should rename the flag to 'ForcePreemptibleInstances' or 'AlwaysUsePreemptibleInstances' ? Also, the meaning of 'false' should be documented in config.default.yml.

The [doc page](#) says "To use preemptible instances, set UsePreemptibleInstances: true and add entries to InstanceTypes with Preemptible: true to config.yml." Now that the default for UsePreemptibleInstances is true, should we remove "set UsePreemptibleInstances: true and" or change it to "make sure UsePreemptibleInstances isn't set to false"? Or just leave it the way it is?

It definitely needs to be updated to reflect the default and the meaning of both 'true' and 'false'.

Otherwise, LGTM thanks.

#### #13 - 12/23/2021 10:00 PM - Tom Clegg

18562-preemptible @ [0a08f54c405dff0dbda5d6dbc14c1f1c6eeecd39](#) -- [developer-run-tests: #2855](#) [icon?job=developer-run-tests&build=2855](#)

- update doc/config comments
- rename config to AlwaysUsePreemptibleInstances

#### #14 - 12/27/2021 01:02 PM - Ward Vandewege

Tom Clegg wrote:

18562-preemptible @ [0a08f54c405dff0dbda5d6dbc14c1f1c6eeecd39](#) -- [developer-run-tests: #2855](#)  
[icon?job=developer-run-tests&build=2855](#)

- update doc/config comments
- rename config to AlwaysUsePreemptibleInstances

LGTM thanks. Will you also add a note to the upgrade guide about the renaming of the preemptible global config flag?

#### #15 - 12/27/2021 04:16 PM - Tom Clegg

18562-preemptible @ [87f5edeeb1ec8c03a71c2cfa1656176735bfedca](#)

Added upgrade note:

### **Preemptible instance types are used automatically, if any are configured**

The default behavior for selecting "preemptible instances":[{{site.baseurl}}/admin/spot-instances.html](#) has changed. If your configuration lists any instance types with Preemptible: true, all child (non-top-level) containers will automatically be scheduled on preemptible instances. To avoid using preemptible instances except when explicitly requested by clients, add AlwaysUsePreemptibleInstances: false in the Containers config section. (Previously, preemptible instance types were never used unless the configuration specified UsePreemptibleInstances: true. That flag has been removed.)

#### #16 - 12/27/2021 10:06 PM - Ward Vandewege

Tom Clegg wrote:

18562-preemptible @ [87f5edeeb1ec8c03a71c2cfa1656176735bfedca](#)

Added upgrade note:

### **Preemptible instance types are used automatically, if any are configured**

The default behavior for selecting "preemptible instances":[{{site.baseurl}}/admin/spot-instances.html](#) has changed. If your configuration lists any instance types with Preemptible: true, all child (non-top-level) containers will automatically be scheduled on preemptible instances. To avoid using preemptible instances except when explicitly requested by clients, add AlwaysUsePreemptibleInstances: false in the Containers config section. (Previously, preemptible instance types were never used unless the configuration specified UsePreemptibleInstances: true. That flag has been removed.)

Yeah that's good, thanks!

#### #17 - 12/31/2021 07:52 AM - Tom Clegg

- Status changed from In Progress to Resolved

Applied in changeset [arvados|00f1f05789316936db75b4723b1c3d99196c252a](#).

#### #18 - 01/12/2022 07:06 PM - Ward Vandewege

- Related to Idea #18179: Better spot instance support added

#### #19 - 03/24/2022 07:28 PM - Peter Amstutz

- Release set to 46