

Anax Container Image Clean-up

<Please fill out the Overview, Design and User Experience sections for an initial review of the proposed feature.>

Overview

<Briefly describe the problem being solved, not how the problem is solved, just focus on the problem. Think about why the feature is needed, and what is the relevant context to understand the problem.>

Each time an agent starts running a newer service version or a new service, another container image is downloaded to the edge device. Currently anax agents never remove the images that are no longer in use. This will eventually take up an unacceptable amount of storage on the edge device.

Design

<Describe how the problem is fixed. Include all affected components. Include diagrams for clarity. This should be the longest section in the document. Use the sections below to call out specifics related to each aspect of the overall system, and refer back to this section for context. Provide links to any relevant external information.>

Since agreements can be cancelled and remade for the same service by certain changes (like userinput), the image clean-up should not be a part of the agreement lifecycle. By making image-removal behavior configurable through node management policy, users can decide how aggressive they want the cleaning to be or disable entirely if they don't want images ever removed.

There will not be any explicit resolution of conflicting removal policies, whichever is most aggressive will delete the image first in practice.

Image policy - NMPs can be matched by policy or pattern. The agent will store matching NMPs and a new subworker will periodically check if any removals currently need to be executed as well as maintaining a list of images in the database that are affected by the current matching NMPs.

Images downloaded by the agent will be added to the table when it is successfully downloaded. Images not downloaded by the agent that are affected by an image policy will be added to the table by the subworker. The time since last used will be set by the subworker with '0' to indicate it is currently in use. When the subworker runs it will check each image and if it is currently in use by an agent-created container then set the time last used to '0'. If the time last used is currently '0' and there is no agent-created container using that image then set the Time Last Used to the current time.

Image removal is only relevant to edge devices, not clusters as k8s manages service images for clusters outside of the agent's control.

For a policy to apply to all images use *. Partial wildcards will also be allowed. For example, image1:* to apply to all versions of image1. Image names must be complete, the cli will throw an error if a user tries to publish a partial image name (missing version tag)

The following is an example of the field to add to the existing NMP structure. Note the name "image_policy" is left general with a subfield for removal to leave open the possibility for other kinds of image manipulation that we might want to manage through node management policy.

```
"image_policy": {  
  "removal": [{  
    "image_name": "image1:0.0.1",  
    "delete_after_minutes": 30,  
    "agent_downloaded_only": true  
  },  
  {  
    "image_name": "*",  
    "delete_after_minutes": 60,  
    "agent_downloaded_only": false  
  }  
]  
}
```

User Experience

<Describe which user roles are related to the problem AND the solution, e.g. admin, deployer, node owner, etc. If you need to define a new role in your design, make that very clear. Remember this is about what a user is thinking when interacting with the system before and after this design change. This section is not about a UI, it's more abstract than that. This section should explain all the aspects of the proposed feature that will surface to users.>

To use this feature, users will create a node management policy with an "image_policy" section that specifies the images and time to delete them. After publishing it to the exchange, agents will check if it applies to them and if so save it to the local db and a subworker will begin checking if it is time to delete any images.

Command Line Interface

<Describe any changes to the hzn CLI, including before and after command examples for clarity. Include which users will use the changed CLI. This section should flow very naturally from the User Experience section.>

Add the "image_policy" to the nmp template.

Validate the new section and enforce mutual exclusivity of the "upgrade" and "image_policy" subfields.

External Components

<Describe any new or changed interactions with components that are not the agent or the management hub.>

Affected Components

<List all of the internal components (agent, MMS, Exchange, etc) which need to be updated to support the proposed feature. Include a link to the github epic for this feature (and the epic should contain the github issues for each component).>

Security

<Describe any related security aspects of the solution. Think about security of components interacting with each other, users interacting with the system, components interacting with external systems, permissions of users or components>

APIs

<Describe and new/changed/deprecated APIs, including before and after snippets for clarity. Include which components or users will use the APIs.>

Build, Install, Packaging

<Describe any changes to the way any component of the system is built (e.g. agent packages, containers, etc), installed (operators, manual install, batch install, SDO), configured, and deployed (consider the hub and edge nodes).>

Documentation Notes

<Describe the aspects of documentation that will be new/changed/updated. Be sure to indicate if this is new or changed doc, the impacted artifacts (e.g. technical doc, website, etc) and links to the related doc issue(s) in github.>

Test

<Summarize new automated tests that need to be added in support of this feature, and describe any special test requirements that you can foresee.>