Device Onboarding Customization

Status: In Progress

Sponsor User: *IBM*

Date of Submission: 05 Oct 2022

Submitted by: David Booz (booz@us.ibm.com)

Affiliation(s): IBM

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

Scope and Signoff: *(to be filled out by Chair)*

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.>

The purpose of this feature is to provide a mechanism that conditions the linux OS host on an edge device. There are several different kinds of conditioning that are needed:

- Introspection of the device to derive the device capabilities and turn those capabilities into node properties.
- Creation of resources (folders, docker volumes, networks) required by applications. These resources would persist across application restarts.
- Installation of OS packages to ensure that the host has any packages that an application depends on.
- Scan the host for security vulnerabilities.
- Prevent installation of the OH agent or OH services based on the state of the device (e.g. it has software on it with a security vulnerability).
- Apply an Accuknox security policy to the device.
- ...

From the list above, it is clear that there are 2 points in the OH device lifecycle where conditioning is required. The first is near the point where the agent is initially installed. The second is where an agreement has been made and services are about to be deployed. Further, sometimes the required conditioning is not changing anything at all, just inspecting the device in various ways and possibly conditioning the OH metadata describing the device.

Is there a similar requirement for edge clusters?

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.>

Some really rough ideas at this point:

- Completely open pre and post conditioning - A device owner provides a bash script that executes before and after agent registration. A service developer provides a bash script that runs before and after a service is started.
- A framework that invokes specifically named bash script functions at various points in the lifecycle, more closely related to the use cases described in the overview.
- A completely metadata driven approach - There is no code to write, but instead the device owner and service developer provide a metadata description of what they want the system to do and how to handle the results.

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.>

Note: the following user stories differ somewhat from the "conditioning" use cases in the overview. Applying user roles to the requirements helps focus the design by pointing to the part of the system where the requirement should be addressed.

As a device owner, I want the agent to discover custom device attributes and add them to the device's node policy.

As an application developer, I want OH to create system resources (folders, docker volumes, etc) that will persist beyond the boundaries of an agreement.

As an application deployer, I want to avoid nodes that have known security vulnerabilities (could be specific vulnerabilities or any).

As a device owner, I want to apply a security policy to the node before any applications are deployed.

As a device owner, I want OH to assess the condition of the device before allowing an agent to be installed.

As an application developer, I want OH to assess the condition of the device before allowing my application to be deployed.

As an application developer, I want to install system packages on the host OS before my application is deployed, and remove them when my application is in undeployed. **Do we really really want to do this?**

As an application developer, I want to know what versions of system packages on the host OS are installed before my application is deployed and add these to the device's node policy (eg What version of Nvidia JetPack / CUDA)

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.>

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.>