Moonraker Project Definition

Chance Murray

Grant Perdue

Dan Kindt

Madison Robertson

Gabe Coelho

# Project Objective Statement

Create a tool to help users view the data associated with their cloud-native application bundles by April 10, 2020.

## Stakeholders

* Anyone that has cloud-native application bundles installed
* DevOps and infrastructure teams
* Site reliability engineers
* System administrators
* Anyone deploying applications using cloud-native application bundles

## Problems Moonraker Solves

* Gathers and provides information about CNAB installations
* Creates an easy-to-read dashboard
* Sets up data to automatically update packages (in the future)

## Other Contributors

* Open source developers
* Microsoft Azure Open Source Dev Team
* CNAB open-source project members

## Ideas and vision

* Visibility into installed CNAB bundles
* Graphical user interface instead of CLI tools

# Research

## Primary research

Currently, monitoring tools in the market focus more on application monitoring (Nagios, New Relic, AWS CloudWatch, AppDynamics, etc.) and cloud specific monitoring, and not cloud bundles tracking.

The “bundles” in this case are specific application installer packages that live in the cloud. There is currently no way to compile a list of application bundles and relevant data about them such as how much memory the packages require, bundle size, bundle version, and other metadata.

For more information on Cloud-Native application bundles, see appendix A.

## Secondary research

DevOps engineers and Site Reliability Engineers who use Cloud Native Application Bundles have to manually check bundle versions, size, and relevant metadata in order to track changes.

## Value proposition

Instead of manually checking for bundle specific information one by one (when there could be hundreds of bundles), users are able to see a visualization of bundles and relevant information about each.

## Known Constraints

* Having the sponsor coach about how to connect to cloud data sources
* Availability of new, critical-path plugins being developed by the sponsors themselves
* For the deployment stage, an Azure instance where we can deploy the tool.

# Requirements

For this product, it is assumed that a user (stakeholder) has configured their CNAB installers to store data in a remote (cloud) datastore.

The API and frontend will both be distributed via bundles which will include the binary for the backend and a packaged version of the front end. The users will then configure their cloud storage information and then run the service. Any device on the network will then be able to consume the exposed API and view the hosted dashboard.

# Deliverables

## Major Components and Functions

1. Dashboard (front-end user interface)
2. Fully documented API (backend and data gathering)
3. (Optional) Installation Bundle

Project Breakdown

## Task Breakdown for Next Semester

1.0 Create data schemas based off of CNAB configuration files

1.1 Decide on what data is important from claims.json files

1.2 Ask the community for data that would be useful

2.0 Create an API to collect data from cloud storage

2.1 Establish model for serving data

2.2 Read from cloud storage with sponsor’s plugin

3.0 Develop a front-end dashboard for viewing data

3.1 Use API to get data from storage

3.2 Present data in a user-friendly way

## Schema

|  |  |
| --- | --- |
| **IS** | **IS NOT** |
| JSON format per CNAB specification | Other formats |
| An accurate representation of the CNAB “claim” specification | A cloud-specific tool |

## 

## API

|  |  |
| --- | --- |
| **IS** | **IS NOT** |
| A read-only API | An application monitoring tool |
| A way to retrieve relevant information about installed CNAB bundles | A way to generate new information |
| A tool to read CNAB claim files | A tool to generate new information |
|  |  |

## Dashboard

|  |  |
| --- | --- |
| **IS** | **IS NOT** |
| Dynamically-generated web pages and data views based on code served in static web assets | Meant to replace common dashboard tools like Grafana |

*Note: We are still in the research and task breakdown process and will define our is/is not as we move forward*

## Delivery Mechanism

Create a binary with instructions on how to configure the dashboard. Our stretch goal is to incorporate that binary into its own bundle.

## Documentation

The team will create a GitHub repository with instructions in the README.MD managed by the CNAB open source community.

All teamwork and meetings are recorded in Notion.so.

## Video

The team will create a video to showcase the use cases and problems that Moonraker solves.

## Poster

The team will create a poster with relevant information about Moonraker.

Prototype

The prototype was a NodeJS version of our final product, with a rudimentary web dashboard. For demonstration purposes, we installed “HelloWorld” bundles onto a local machine under different names, enabling us to have CNAB Claim files (in JSON format) to work with. From the Claims, we gathered information about the bundles. In addition, the program tracked the last changes made on a bundle, so we could show it in our dashboard.

The prototype helped us answer a very important question. The question it answered was whether we could, in a relatively short period of time, create a simpler, localized version of our end product that accomplished the key features of reading Claims and visualizing key information. We were able to accomplish this task quite well. The only questions that remain are: 1) how hard will it be to implement our final solution in our chosen technologies of GoLang and VueJS?, 2) how will we incorporate our sponsor’s cloud-agnostic plugin into our API?, and 3) what information do our users really want in the dashboard?

References

We did not reference any articles in this document.

Appendices

1. Presentation 1 Handout content:



More About CNAB

Team Moonraker

Definition

A Cloud Native Application Bundle (CNAB) is an open source, cloud-agnostic specification for packaging and running distributed applications. It facilitates the bundling, installing and managing of [container](https://www.webopedia.com/TERM/C/container.html)-native apps and their dependent services. It describes a technology for bundling, installing, and managing distributed applications, that are by design, cloud agnostic.

Use case

Modern applications are made up of a wide range of components and services — they can be comprised of multiple cloud resources, managed services, SaaS offerings, containers, configuration formats, functions, and more. CNAB pulls these disparate components together, providing a common packaging format for multiservice applications. These bundles can be developed, managed and shared (across a registry like Docker Hub) as one immutable composite unit without forcing any specific environment/clouds. (From article linked below)

|  |  |
| --- | --- |
| **What Is CNAB?**  **https://bit.ly/2DIN9fW** | **CNAB Official Website**  **https://cnab.io** |