ContainerPilot 3.0 Released with Multiprocess Container Support

 | by **Hrishikesh Barua** on Jul 27, 2017. Estimated reading time: 2 minutes |

Joyent released version 3.0 of ContainerPilot which functions as an init system for running multiple processes inside containers. It automates the processes of service registration, service discovery, health checks, and lifecycle management for the processes. It has a new HTTP-based API, a simpler configuration language and Consul-only support.

An init system in a container handles the PID-1 problem by handling the child processes that come after it. ContainerPilot was initially built with the concept of a single "main" application. In version 3 it functions as a controller for multiple processes inside a container. This was driven mostly by user feedback, ranging from confusion about multiple configuration options to interprocess dependencies. In the new version, each process has its own health checks, dependencies, frequency of running, and lifecycle hooks for startup and shutdown.

Multiple processes in a container usually result when there are supporting processes like service discovery agents running alongside the main app. Configuration of these processes may be different from that of the main application. They can also depend on other services which might run in the same container or in external containers. These factors complicate the task of orchestration.

A pattern called "Autopilot", promoted by Joyent, aims to move all the orchestration responsibilities to the application itself, so that no external orchestrator is required. Orchestration here usually involves actions like registration of the service endpoint in a registry so that other services can find it, definition of health checks, defining the dependencies and managing its lifecycle. In ContainerPilot's case, the latest release focuses only on Hashicorp's Consul as the service discovery mechanism. It drops support for the previously supported etcd.

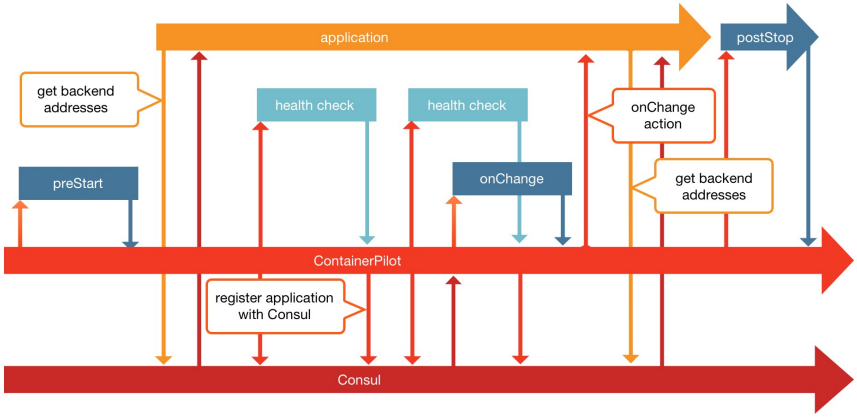

Image Courtesy: https://www.joyent.com/blog/containerpilot-hello-world

The new HTTP-based API in ContainerPilot allows users to send signals to control applications and their environment. The HTTP requests can be sent to a socket inside the container to update environment variables, toggle maintenance mode, record metrics for a Prometheus endpoint, or reload the ContainerPilot configuration. This replaces the previous signal-based mechanism which could only toggle state without any resulting feedback.

The configuration language in v3 has consolidated the way that users can define dependencies between service lifecycle events. For example, nginx can be defined to start only after the resident Consul agent has started. These dependencies can be defined in terms of starting or stopping another service, or any of their intermediate states. The release also consolidates all previously defined "behaviour" hooks into an abstraction called a Job. Various jobs can be tied together to create a chain of dependent applications.

ContainerPilot grew out of a previous Joyent product called ContainerBuddy. It is also available as an open source project.

Tell us what you think

Community comments