

joyent / containerpilot

A service for autodiscovery and configuration of applications running in containers

#containerpilot #consul #orchestration #containers #docker #joyent #service-discovery #triton

690 commits

7 branches

74 releases

27 contributors

MPL-2.0

Branch: masterNew pull request

Create new fileUpload filesFind fileClone or download

eliasbrange and cheapRoc Fix typo in docs for initial_status.		Latest commit bdc8fb7 on May 25
.github	contributing helpers (#395)	a year ago
client	docs: add pointers to godoc (#475)	a year ago
commands	Implement new signal based events	11 months ago
config	config: Adjust loop function to allow ranges by strings/env vars	5 months ago
control	Refactor away EventHandler into separate pub/sub interfaces (#476)	a year ago
core	Clean-up a few things around global/job shutdown	10 months ago
discovery	Fix lint errors.	6 months ago
docs	Fix typo in docs for initial_status.	4 months ago
events	logging: Ignore debug logs of timer ticks for internal telemetry heal...	5 months ago
integration_tests	Implement new signal based events	11 months ago
jobs	Rename initialStatus -> initial_status.	6 months ago
scripts	add go vet to linting, fix lints this finds	a year ago
subcommands	docs: add pointers to godoc (#475)	a year ago
sup	Implement new signal based events	11 months ago
telemetry	Refactor away EventHandler into separate pub/sub interfaces (#476)	a year ago
tests	Migrate from deprecated Consul API call PassTTL to UpdateTTL	a year ago
version	docs: add pointers to godoc (#475)	a year ago
watches	Clean-up commented out code found during CR	a year ago
.gitignore	re-remove coverage reports and add to gitignore	a year ago
.travis.yml	Update Compose in TravisCI	2 years ago
CHANGELOG.md	Update CHANGELOG.md for release 3.6.1	10 months ago
CONTRIBUTING.md	contributing helpers (#395)	a year ago
Dockerfile	Upgrade ContainerPilot to Go 1.9.x and Consul 1.0.0 (#527)	10 months ago
LICENSE	Initial commit	3 years ago
README.md	docs: add pointers to godoc (#475)	a year ago
glide.lock	Upgrade ContainerPilot to Go 1.9.x and Consul 1.0.0 (#527)	10 months ago
glide.yml	Upgrade ContainerPilot to Go 1.9.x and Consul 1.0.0 (#527)	10 months ago
main.go	docs: add pointers to godoc (#475)	a year ago
makefile	Upgrade ContainerPilot to Go 1.9.x and Consul 1.0.0 (#527)	10 months ago

README.md

ContainerPilot

An init system for cloud-native distributed applications that automates the process of service discovery, configuration, and lifecycle management inside the container, so you can focus on your apps.

What is ContainerPilot?

Orchestration is the automation of the operations of an application. Most applications require operational tasks like connecting them to related components ([WordPress needs to know where it's MySQL and Memcached servers are, for example](#)), and some applications require special attention as they start up or shut down to be sure they bootstrap correctly or persist their data. We can do all that by hand, but modern applications automate those tasks in code. That's called "orchestration."

To make this work, every application needs to do the following (at a minimum):

- Register itself in a service catalog (like Consul or Etcd) for use by other apps
- Look to the service catalog to find the apps it depends on
- Configure itself when the container starts, and reconfigure itself over time

We can write our new applications to do that, but existing apps will need some help. We can wrap each application in a shell script that registers itself with the discovery service easily enough, but watching for changes to that service and ensuring that health checks are being made is more complicated. We can put a second process in the container, but as soon as we do that we need an init system running inside the container as well.

ContainerPilot to the rescue!

ContainerPilot is an init system designed to live inside the container. It acts as a process supervisor, reaps zombies, runs health checks, registers the app in the service catalog, watches the service catalog for changes, and runs your user-specified code at events in the lifecycle of the container to make it all work right. ContainerPilot uses Consul to coordinate global state among the application containers.

Quick Start Guide

Check out our ["Hello, World" application](#) on GitHub. Assuming you have Docker and Docker Compose available, it's as easy as:

```
git clone git@github.com:autopilotpattern/hello-world.git
cd hello-world
docker-compose up -d
open http://localhost
```

This application blueprint demonstrates using ContainerPilot to update Nginx upstream configuration at runtime. Try scaling up via `docker-compose scale hello=2 world=3` to see the Nginx configuration updated.

You can also [download](#) the latest release of ContainerPilot from GitHub.

Documentation

Documentation for ContainerPilot and where it fits with the rest of the Triton ecosystem can be found at www.joyent.com/containerpilot. The index below links to the documentation in this repo for convenience.

Lifecycle

- [What is a job?](#)
- [What is an event?](#)
- [What is a watch?](#)
- [How do events trigger jobs?](#)
- [How can jobs be ordered?](#)

Design: the Why of ContainerPilot

- [Why active service discovery?](#)
- [Why isn't there a "post-start" or "started" event?](#)
- [Why Consul and not etcd or Zookeeper?](#)
- [Why are jobs not the same as services?](#)
- [Why don't watches or metrics have an exec field?](#)

- [Why use something other than ContainerPilot?](#)

Configuration

- [Installation](#)
- [Configuration file](#)
 - [Schema](#)
 - [Consul](#)
 - [Logging](#)
 - [Jobs](#)
 - [Watches](#)
 - [Control](#)
 - [Telemetry](#)
 - [Extras](#)
 - [Interfaces](#)
 - [Environment variables](#)
 - [Template rendering](#)
- [Consul](#)
 - [Client configuration](#)
 - [Consul agent configuration](#)
- [Jobs](#)
 - [Lifecycle Events](#)
 - [Configuration](#)
 - [name](#)
 - [exec](#)
 - [when](#)
 - [timeout](#)
 - [stopTimeout](#)
 - [restarts](#)
 - [health checks](#)
 - [service discovery](#)
 - [Exec arguments](#)
- [Watches](#)
- [Telemetry](#)
 - [Sensor configuration](#)
- [Control plane](#)
 - [ContainerPilot subcommands](#)
- [Logging](#)
- [Example configurations](#)

Support

- [Where to file issues](#)
- [Contributing](#)
- [Backwards compatibility](#)

You might also read [our guide building self-operating applications with ContainerPilot](#) and look at the examples below.

Examples

We've published a number of example applications demonstrating how ContainerPilot works.

- [Applications on autopilot: a guide to how to build self-operating applications with ContainerPilot](#)
- [MySQL \(Percona Server\) with auto scaling and fail-over](#)
- [Autopilot Pattern WordPress](#)
- [ELK stack](#)
- [Node.js + Nginx + Couchbase](#)

- [CloudFlare DNS and CDN with dynamic origins](#)
- [Consul, running as an HA raft](#)
- [Couchbase](#)
- [Mesos on Joyent Triton](#)
- [Nginx with dynamic upstreams](#)