autopilotpattern / **etcd**

A service for autodiscovery and configuration of applications running in containers

| 20 commits | 1 branch | 0 releases | 3 contributors | MPL-2.0 |
|---|---|---|---|---|

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

cheapRoc Merge pull request #7 from cheapRoc/b-restart ... Latest commit fea9979 on Jun 22, 2017

| | | |
|---|---|---|
| bin | Remove public ports from provisioning on Triton | a year ago |
| .gitignore | Upgrade to etcd v3.2 | a year ago |
| Dockerfile | Remove ContainerPilot artifacts | a year ago |
| LICENSE | Initial commit | 2 years ago |
| README.md | Touch up README.md | a year ago |
| docker-compose.yml | Forgot restart always within docker-compose files | a year ago |
| local-compose.yml | Forgot restart always within docker-compose files | a year ago |
| start.sh | Only start on Triton if Docker certs are from triton CLI | a year ago |

README.md

# Autopilot pattern etcd

*Autopilot Pattern implementation of etcd*


This repo is a demonstration of etcd designed for self-operation according to the Autopilot pattern.

An etcd cluster needs an external source of data for all the nodes to find each other initially. This can be a bootstrap service (another etcd cluster) or an SRV record. Triton CNS does not yet support SRV records, so until it does we're standing up a temporary single-node cluster to bootstrap the cluster. After the cluster is scaled-up we can remove the bootstrap node.

## Getting started

1. Get a Joyent account and add your SSH key.
2. Install the Docker Toolbox (including `docker` and `docker-compose`) on your laptop or other environment.
3. Install the Triton Docker CLI when utilizing this pattern on Triton as well. This provides both `triton-docker` and `triton-compose`.
4. Install the the Joyent Triton CLI (`triton` replaces our old `sdc-*` CLI tools) and set up your Triton profile.

At this point you're ready to start the cluster. A script `./start.sh` has been provided. It detects your deployment options, either Triton or local Docker, and uses Docker Compose to create each node. An initial bootstrap node is created along with a discovery token (see the etcd docs on cluster discovery for details) before the cluster can be scaled up. You can pass an environment variable `SCALE` to the `./start.sh` script to set the cluster size to something other than the default 3 nodes.

### Starting

```
$ ./start.sh
Using discovery node for bootstrapping local 3-node cluster.
Starting e_bootstrap_1
{"action":"set","node":{"key":"/discovery/2AF6A60D-5196-4358-AA7B-A706DC74D3BD/_config/size","value":"3","modifiedInde
e_bootstrap_1 is up-to-date
Creating e_etcd_3
Creating e_etcd_2
Creating e_etcd_1
Desired container number already achieved
Stopping bootstrap node, no longer required
e_bootstrap_1
```

```
Displaying cluster health
member 4ed7a60797acad53 is healthy: got healthy result from http://172.18.0.4:2379
member aef40bf222ea5e2f is healthy: got healthy result from http://172.18.0.3:2379
member c329c4aa339bb6f7 is healthy: got healthy result from http://172.18.0.5:2379
```

```
$ docker ps
CONTAINER ID        IMAGE                   COMMAND                  CREATED             STATUS              PORTS
8e65eb4e319c        autopilotpattern/etcd   "/usr/local/bin/et..."   12 minutes ago      Up 12 minutes       0.0.0.0:3
ef351973dd19        autopilotpattern/etcd   "/usr/local/bin/et..."   12 minutes ago      Up 12 minutes       0.0.0.0:3
930804bf69fd        autopilotpattern/etcd   "/usr/local/bin/et..."   12 minutes ago      Up 12 minutes       0.0.0.0:3
```

## Stopping

```
$ COMPOSE_PROJECT_NAME=e docker-compose -f local-compose.yml stop
Stopping e_etcd_1 ... done
Stopping e_etcd_2 ... done
Stopping e_etcd_3 ... done
```

You can also run the previous example with `COMPOSE_PROJECT_NAME=e triton-compose stop`.