kadende-cluster-management Documentation

Release 0.0.0

Francis Mwangi

Jul 30, 2018

Contents:

A platform for creating/maintaining and scaling a cluster of machines in different providers.

CHAPTER 1

Design goals

Kadende cluster management is being inspired by amazon autoscaling group, infrakit(https://github.com/docker/infrakit), coorp (http://coopr.io/), and docker cloud (which was terminated)

Its not meant to do the work of other orchestrator like swarm and kubernetes its infact meant to work with them.

Then main goal that cluster management is trying to solve is provisioning a group on nodes in any cloud provider and maintain the size of the cluster. And do health checks on the cluster.

What to run on the cluster is upto you i.e if you decide to run servers there or even swarm or kubernetes its all up to the user

Design Goals.

1. Making it easy to provision a group of nodes of the same kind and maintain the cluster size

    Keep on doing health checks on the cluster if its down, terminate and create new one to maintain cluster size

    With docker-cloud it was easy to create a cluster in the providers that they supported, this worked well but you had to maintain the cluster size yourself, in-case the node was not reachable docker-cloud notified the user but they didn't maintain the cluster by themselves

    With Terraform you could archive this, provisioning a group of nodes in with any cloud provider even with you own data-center. But to doing health check and maintain the cluster is not the easiest solution. You might have to use terraform Enterprise to archive this.

    With infrakit you can achieve all of this but everything is based on cli and not apis. Was a bit difficult to host it in the cloud. Then their plugin architecture makes it possible to write any provider in any language but it does not scale well to multiple clusters and multiple nodes. i.e the plugins needs to be run high availability and still run infrakit in high availaibiliy.

    With coorp (http://coopr.io/) you can archive this but I didn't find it to be user friendly

2. User to have total control of how to configure his nodes and how to do health checks.

    With docker-cloud health checks are done via agent installed in the nodes reporting to cluster management. And you don't have any option of adding your custom healthcheck. You don't have control of customizing what will be installed when the cluster is being provisioned

The rest coopr, terraform and infrakit support this.

3. Be able to run it in high available i.e capability of supporting multiple clusters and nodes

    You could archive this with docker-cloud (since it wasn't self managed)

4. Should work with any provider
5. Support for user management

    coor and docker-cloud support this the rest don't.

CHAPTER 2

Architecture Overview

Kadende cluster was architectured with the Design Goals in mind and heavily borrows from infrakit.

Kadende heavly relies on plugin architecture.

See also:

More info on plugins kadendePluginOverview

We have divided the architecture to three main components:

2.1 ClusterController

This is a stateless app that its called with a cluster id. With the cluster id it gets the cluster configuration and cluster resources from the store.

cluster configuration is the spec that was given by the user, and cluster resources is all the resources that cluster controller has created.

Its main job is to maintain the size and health of a cluster.

A cluster configuration contains a group of clusters. Each cluster is a group of resources of the same resource type.

Sample cluster configuration

```
version: "0.0.1"
name: basic_cluster
groups:
  group1:
    size: 2
    resourcePlugin:
      name: aws
      version: latest
    resourceProperties:
      key: mwas
      instanceType: t2.small
```

(continues on next page)

(continued from previous page)

```
    image: debian
group2:
  sub-group1:
    size: 2
    resourcePlugin:
      name: gcd
      version: 0.0.1
    resourceProperties:
      key: mwas
      instanceType: t2.micro
      image: amazonlinux
  sub-group2:
    size: 3
    resourcePlugin:
      name: digitalOcean
      version: latest
    resourceProperties:
      instanceType: standard
      memory: 1
      sshKey: mwas
customDatacenter:
  size: 2
  resourcePlugin:
    name: my-custom-plugin
    version: 0.0.1
    url: https://github.com/mwaaas/kadende-provider-file/releases/download/0.0.
↪1/plugin.so
  resourceProperties:
    instanceSize: abcd
```

Fields that cluster configuration has:

1. **id (Optional field)** This would be the pk of the cluster configuration in the store.

   If not defined a generic one would be generated.

2. **version (Mandatory field)** This contains the version of the cluster

3. **name (Mandatory field) - should be unique** This contains the name of the cluster.

4. **groups (Mandatory field)** this is a list of one or more group.

   Fields in a group:

   - **resourcePlugin** This defines the plugin that's going to create the resource.

     It requires the following fields:

     – name

     – version (optional defaults to latest)

     – url (optional if you want to define your own plugin)

   - **resourceProperties** This defines properties that are going to be used by the resourcePlugin to create resource.

     The fields are determined and validated by the plugin you selected.

2.1.1 Cluster controller workflow

Flow:

1. **Given cluster id** We get cluster configuration and cluster resources from the store.
2. **Loop.** We then loop each group.

   In each loop we do the flowing.

   - Get all the resources via tags and return those whose status is active. check the instance that we have got are the ones in our cluster resource

     Will archive this by calling resourcePlugin describe method.
   - Delete those instance that are not in our cluster resources.
   - check if the number of the resources matches the number of the cluster group. create/delete to maintain the cluster size.
   - Store all events in the store to be used by the ui.

2.2 NodeController

CHAPTER 3

Indices and tables

- genindex
- modindex
- search