GIAC | BEIJING Dec.12.16-17

QINGCLOUD青云

企业应用云化架构设计

周小四 | Ray Zhou
青云QingCloud AppCenter及大数据平台负责人

技术架构未来

QINGCLOUD青云

Agenda

- 概述
- 架构演变
- 标准化平台设计

概述

QINGCLOUD青云

云计算常态化

云应用常态化

极低门槛

标准化平台

架构演变

- 完全孤立、每个应用独立开发
- 底层调用系统框架化，每个应用独立开发
- 端到端标准化应用云化开发平台
  - 高度抽象、高度标准化
  - 批量生产照样适合云产品开发
  - 云技术不应该是核心，应用本身才是

标准化平台设计

- 难点
- 现有解决方案
- 青云解决方案

难点

- 应用种类繁多
  - 无角色集群、单主、一主一从，一主多从，多主多从，分片多主多从
  - 各类应用集群管理方式多样化
- 应用配置变更
  - 不同角色节点配置变更
- 服务依赖感知
  - 依赖服务如ZK节点个数变化，IP变化、endpoint变更...
- 集群弹性伸缩

现有方案

- Rancher, Mesosphere DC/OS, Docker Swarm, K8S
- 离生产环境还有距离

QingCloud AppCenter 2.0 目标

QINGCLOUD青云

- 人人能开发云产品
- 开发周期：几个月→几天
- 学习成本低
- 合作伙伴拥有完整的云平台：运营、运维、开发、销售

集群管理引擎

- 输入信息 cluster.json

QINGCLOUD青云

示例 1 - ZooKeeper (无角色)

```
{
  "app_id": "app-zkv33646",
  "app_version": "1.0",
  "name": "ZK",
  "description": "my ZooKeeper App",
  "vxnet": "vxnet-p050mao",
  "node": {
     "container": {
        "type": "kvm",
        "image": "img-zookeeper",
        "zone": "pek3a"
     },
     "instance_class": 0,
     "count": 3,
     "cpu": 1,
     "memory": 512,
     "volume": {
         "size": 3,
         "mount_point": "/zk_data",
         "filesystem": "xfs",
         "class": 0
     },
     "server_id_upper_bound":255,
     "service": {
        "start": {
           "cmd": "/opt/zookeeper/bin/zkServer.sh start"
        },
        "stop": {
           "cmd": "/opt/zookeeper/bin/zkServer.sh stop"
        }
     }
  },
  "advanced_action": ["change_vxnet", "scale_horizontal"]
}
```

示例 2 – Hbase(多角色/应用依赖/应用配置)

```
{
    "app_id": "app-hbase160",
    "app_version": "1.0",
    "name": "MyHBase",
    "description": "my hbase App",
    "vxnet": "vxnet-t8szyjn",
    "links": {
        "zk_service": "cl-w8qh5hf6"
    },
    "node": [{
        "role": "hbase-master",
        "container": {
            "type": "kvm",
            "image": "img-bzf0t38c",
            "zone": "pek3a"
        },
        "count": 1,
        "cpu": 1,
        "memory": 1024,
        "volume": {
            "size": 10,
            "mount_point": "/bigdata1",
            "filesystem": "ext4",
            "class": 0
        },
        "instance_class": 0,
        "passphraseless": "ssh-dsa",
        "advanced_action": ["change_vxnet", "scale_horizontal"],
        "service": {
            "start": {
                "order": 2,
                "cmd": "USER=root /opt/hbase/bin/start.sh"
            },
            "stop": {
                "order": 2,
                "cmd": "USER=root /opt/hbase/bin/stop.sh"
```

```
            },
            "scale_out": {
                "order": 2,
                "nodes_to_execute_on": 1,
                "cmd": "USER=root /opt/hbase/bin/start-hbase.sh"
            }
        },
```

```
    {
        "role": "hbase-slave",
        "container": {
            "type": "kvm",
            "image": "img-bznsfhwn",
            "zone": "pek3a"
        },
        "count": 3,
        "cpu": 1,
        "memory": 2048,
```

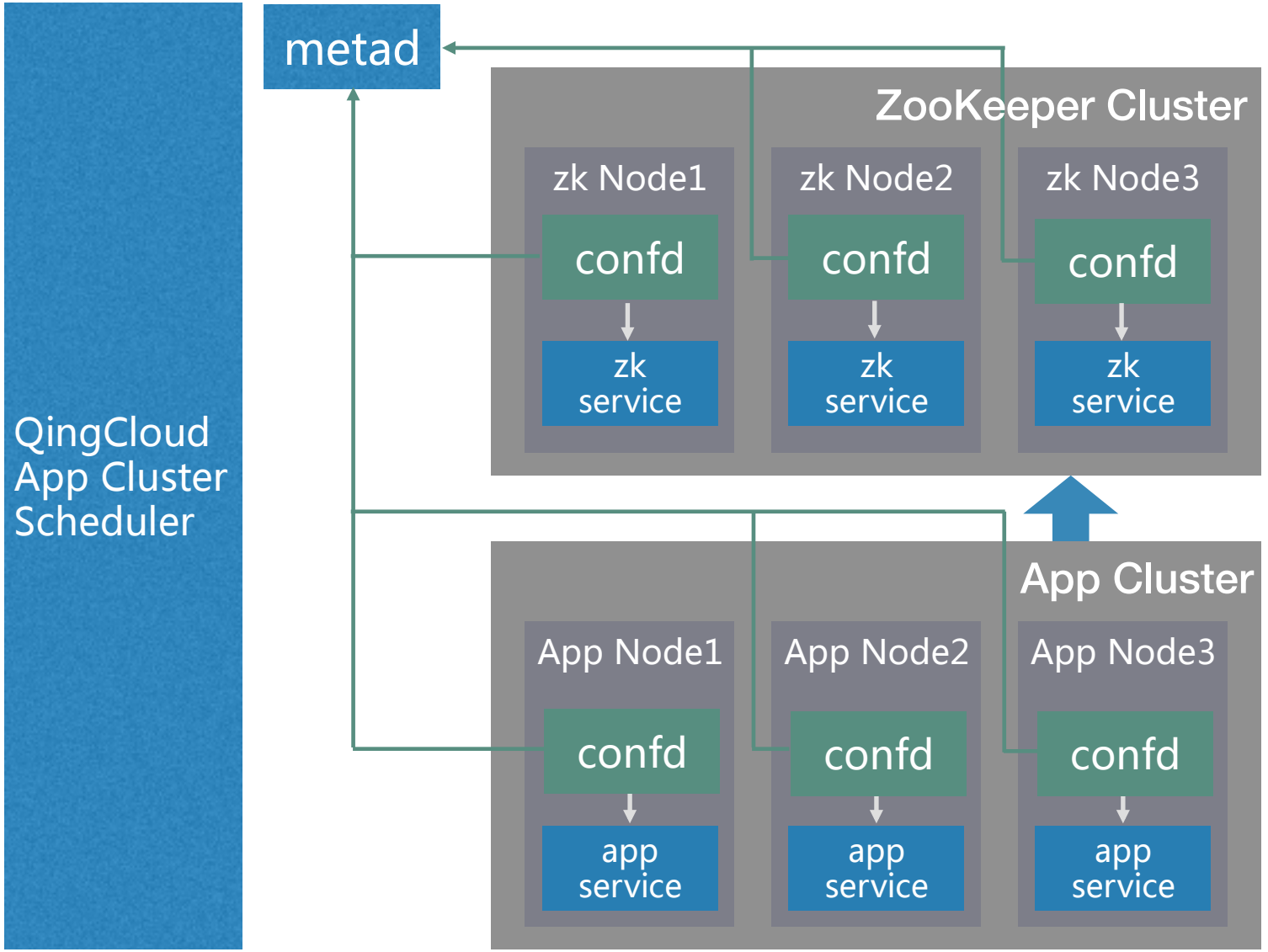
```
219     "env": {
220         "fs.trash.interval": 1440,
221         "dfs.replication": 2,
222         "dfs.namenode.handler.count": 10,
223         "dfs.datanode.handler.count": 10,

246          "qingcloud.phoenix.on.hbase.enable": "false",
247          "phoenix.functions.allowUserDefinedFunctions": "false",
248          "phoenix.transactions.enabled": "false"
249     },

250     "endpoint": {
251         "rest_port": {
252             "port": 8000
253         },
254         "thrift_port": {
255             "port": 9090
256         }
257     }
258 }
```

架构图

- 调度系统统一管理
- 元数据管理 – metad
- 配置自动变更 - confd



元数据结构(1)

- **/self**
  - **/hosts**/*[role name]*/[instance_id]*
    - **/ip** [IP address]
    - **/mac** [MAC address]
    - **/sid** [server ID]
    - **/node_id** [node ID]
    - **/cpu** [cpu]
    - **/memory** [memory in MB]
    - ***/pub_key*** [pub key string]
    - ***/physical_machine*** [ID of the physical machine that hosts the instance]
  - **/host**
    - **/ip** [IP address]
    - **/mac** [MAC address]
    - **/sid** [server ID]
    - **/node_id** [node ID]
    - **/cpu** [cpu]

元数据结构(2)

- /**cluster**
  - /**app_id** [application ID]
  - /**cluster_id** [cluster ID]
  - /**vxnet** [VxNet ID]
- /**env**/[parameter key]* [parameter value]
- /**adding-hosts**/[instance_id]*
  - /**ip** [IP address]
  - /**mac** [MAC address]
  - /**sid** [server ID]
  - /**node_id** [node ID]
  - /**cpu** [cpu]
  - /**memory** [memory in MB]
  - /***pub_key*** [pub key string]
  - /***physical_machine*** [ID of the physical machine that hosts the instance]

元数据结构(3)

- **/links**/[service name]* [cluster_id]
- /**cmd**
  - /**id** [cmd ID]
  - /**cmd** [cmd string]
  - /**timeout** [timeout(second)]

- ***/endpoint***/[service name]*
  - /**port** [port or a reference to env parameter]
  - /**protocol** [protocol]

配置管理(1)

- 创建/etc/confd/conf.d/zoo.cfg.toml

```
[template]
src = "zoo.cfg.tmpl"
dest = "/opt/zookeeper/conf/zoo.cfg"
keys = [
    "/",
]
reload_cmd = "/opt/zookeeper/bin/restart-server.sh"
```

- 创建/etc/confd/conf.d/myid.toml

```
[template]
src = "myid.tmpl"
dest = "/zk_data/zookeeper/myid"
keys = [
    "/",
]
```

配置管理(2)

QINGCLOUD青云

- 创建/etc/confd/templates/zoo.cfg.tmpl

```
tickTime=2000
initLimit=10
syncLimit=5
dataDir=/zk_data/zookeeper
clientPort=2181
maxClientCnxns=1000
{{range $dir := lsdir "/hosts"}}{{$sid := printf "/hosts/%s/sid" $dir}}
{{$ip := printf "/hosts/%s/ip" $dir}}server.{{getv $sid}}={{getv $ip}}:2888:3888
```

- 创建/etc/confd/templates/myid.tmpl

```
{{getv "/host/sid"}}
```

应用管理

- 创建应用所需配置包
  - config.json 应用配置信息
  - cluster.json.mustache 创建集群的模板文件
- config.json + cluster.json.mustache = cluster.json 既应用实例

应用管理

- config.json前端根据此文件展现给用户来配置应用

```
{
    "key": "vxnet",
    "label": "VxNet",
    "description": "Choose a vxnet to join",
    "type": "string",
    "default": "",
    "required": "yes"
},
{
    "key": "node",
    "label": "node",
    "description": "role-based node properties",
    "type": "array",
    "properties": [
        {
            "key": "cpu",
            "label": "CPU",
            "description": "CPUs of each node",
            "type": "integer",
            "default": 1,
            "range": [
                1,
                2,
                4,
                8
            ],
            "required": "yes"
        },
```

应用管理

- cluster.json.mustache创建应用实例模板

* 注: {{}} 内为变量，来自用户根据config.json在前端填写的内容

```
{
  "name": {{cluster.name}},
  "description": {{cluster.description}},
  "vxnet": {{cluster.vxnet}},
  "node": [
      {
        "container": {
            "type": "kvm",
            "zone": "allinone",
            "image": "img-zkv33646"
        },
        "instance_class": {{cluster.node.instance_class}},
        "count": {{cluster.node.count}},
        "cpu": {{cluster.node.cpu}},
        "memory": {{cluster.node.memory}},
        "volume": {
            "size": {{cluster.node.memory}} * 4 / 1024 * 10,
            "mount_point": "/zk_data",
            "filesystem": "xfs"
        },
```

应用编排

- 单应用配置包可以创建复杂的应用
  - 多角色：ZK, Kafka，Storm, Hadoop，组成一个日志系统应用
- 应用嵌套 - 多个应用组成一个大应用
  - 多应用： ZK, Kafka, Storm, Hadoop，组成一个日志系统应用

日志分析系统 1.0

100%

取消 保存并退出

系统应用

子应用 主机 负载均衡器

创建的应用 | 安装的应用

亦维工具集 诸葛IO 诸葛IO

云监控 安全宝 XMeter

性能极客 云观测 ZooKeeper

云监控 Hodgep +

添加网络 防火墙 SSH 秘钥

私有网络 1

子应用：主站备库

Master
主机
秘钥 kp-ttryr3v5

Slave
主机
秘钥 kp-58id0xi6

防火墙
sg-vb6c3te1

Hak2d
角色名称

ZooKeeper
角色名称

诸葛 IO
角色名称

性能极客
角色名称

应用配置 | 实例属性设置

集群名称 市场业务

版本版本 2.0.1

CPU (核) 8

内存 (MB) 2048

节点数 (个) 13

主机类型 超高性能型

硬盘类型 容量型

关注我们

QingCloud-IaaS

青云QingCloud

www.qingcloud.com