# Ansible playbook

## enviroment 目录

enviroment目录大部分都是根据不同的区域对所有主机的分组与归类，然后通过对组进行定义变量，以便以后管理。下面是简单的例子

**cat enviroment/aliyun/host**

[nginx-cms-test]

nginx[1:100].cms.test.pdmi.cn

[nginx-cms-test:vars]

Server\_name = cms.test.pdmi.cn

[nginx1.cms.test.pdmi.cn]

nginx1.cms.test.pdmi.cn

[nginx1.cms.test.pdmi.cn:vars]

hostname = nginx1.cms.test.pdmi.cn

## 二、 Conf目录

### 2.1 Group目录

主要用于批量创建用户组，这个与user目录是一样的，对enviroment目录下分好的组进行批量创建用户组，下面是简单的例子

cat conf/group/aliyun/group.nginx-cms-test.yml

- host: nginx-cms-test

remote\_user: yulei

sudo: yes

vars:

group: devops

gid: 1000

tasks:

- name: add group {{ group }}

group: name={{group}} gid={{gid}}

### 2.2 user 目录

User目录主要用于创建用户，也是根据enviroment目录下不同的组来创建，下面是一个简单的例子

cat conf/user/aliyun/user.nginx-cms-test.yml

- hosts: nginx-cms-test

remote\_user: yulei

sudo: yes

vars:

user: test

uid: 1000

group: devops

password: "$6$8Mbb4uK6$s3aSbYEvT3iBzookpnTC.UEzRtSEuGqO6NcdklCucKsF8DOszL6LKGah21dPaV3B4eqTFnJYMs9rlQoFBElWH1"

#

password1: "$6$Ha/56mOd$v7/RLQcSN82dOwf6QqN8djYOqZWLANUFYir97bTb1Ui/x3.u7rFKdlpnb.eTxAx1lOzuhlsBnZ/4ITw7TFSS41"

enable\_sudo: "yes"

tasks:

- name: add user {{ user }}

user: name={{user}} uid={{uid}} password={{password}} group={{group}}

- name: add to sudoers

lineinfile: dest=/etc/sudoers state=present line='{{item}}'

with\_items:

- "{{user}} ALL=(ALL) NOPASSWD: ALL"

when: enable\_sudo == "yes"

- name: remove from sudoers

lineinfile: dest=/etc/sudoers state=absent line='{{item}}'

with\_items:

- "{{user}} ALL=(ALL) NOPASSWD: ALL"

when: enable\_sudo == "no"

### 2.3 hostname目录

Hostname目录主要用于修改主机名，因为主机名需要一个个修改，下面是简单的例子

cat conf/hostname/aliyun/hostname.nginx1.cms.test.pdmi.cn

- hosts: nginx1.cms.test.pdmi.cn       
  remote\_user: yulei         
  sudo: true

  tasks:  
    - name: change hostname for {{ ansible\_default\_ipv4.address }}   
      hostname: name={{ hostname }}

### 2.4 site 目录

site目录 主要针对不同的主机分配角色，包括安装什么软件，起什么服务等

- hosts: crt-nginx1                                                     
  remote\_user: rzuo                                                     
  sudo: yes                                                             
                                                                        
  vars:                                 
    zabbix\_server: '10.0.0.43'

zabbix\_agent\_nginx: true  
    zabbix\_agent\_redis: false                 
    zabbix\_agent\_mysql: false     
    zabbix\_agent\_memcached: false

nginx\_worker\_processes: 4                   
  roles:                                                                
    - system                                                            
    #- jdk  
    #- tomcat                                                           
    - zabbix-agent                                                     
    #- gluster-server                                                   
    #- gluster-client                                                   
    - nginx  
    - rsyncd

## 三、 roles 目录

Roles 主要是决定对服务器角色是怎么定义的，包括nginx,redis,memcached等一系列服务和上线，备份等角色。

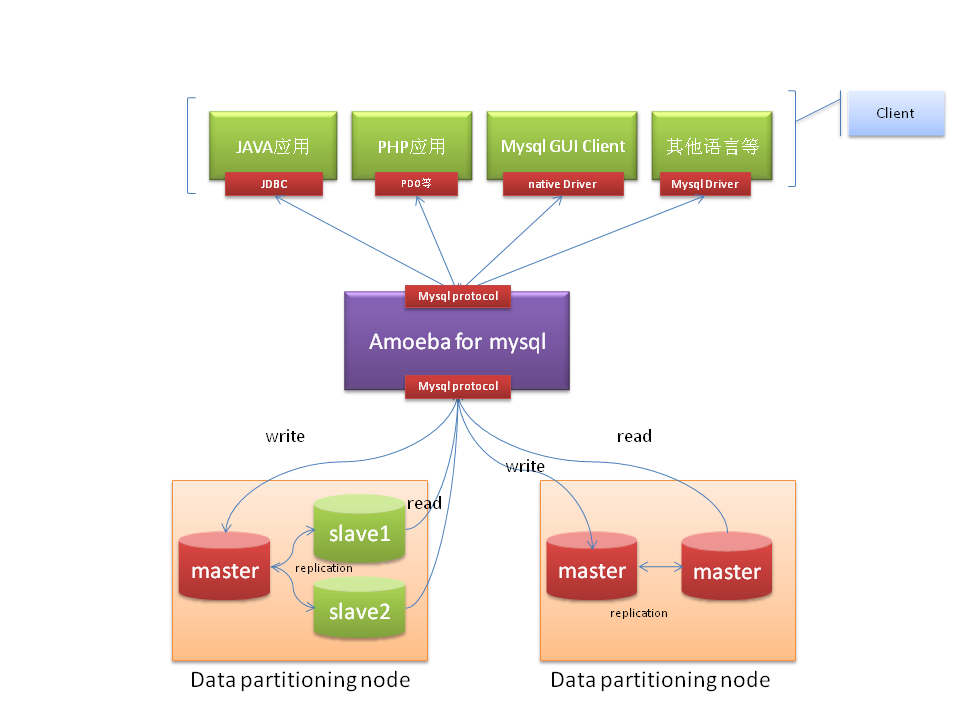
### 3.1 Amoeba

Amoeba致力于分布式数据库前端物理层，它主要在应用层访问mysql时充当mysql路由功能，专注于分布式数据库代理层。具有切分数据库，负载均衡，高可用性，SQL过滤，读写分离，可路由关系的到目标数据库。

切分（Sharding）并不是特定数据库产品所附属的功能，而是在具体技术细节之上的抽象处理。是水平扩展（Scale Out）的解决方案，主要目的是解决单节点数据库服务器的能力限制，以及整个应用其架构的可扩展性（Scalability）。

切分主要有两种方式：水平切分（Horizental Sharding）和垂直切分（Vertical Sharding）。水平切分所指的是通过一系列的切分规则将数据水平分布到不同的DB或table中，在通过相应的DB路由 或者table路由规则找到需要查询的具体的DB或者table以进行Query操作，比如根据用户ID将用户表切分到多台数据库上。垂直切分指的是按业务、产品切分，将不同类型的数据且分到不同的服务器上，通过数据库代理疏通程序与多个数据库的通讯、降低应用的复杂度。

读写分离简单的说是把对数据库读和写的操作分开对应不同的数据库服务器，这样能有效地减轻数据库压力，也能减轻io压力。主数据库提供写操作，从数据库提供读操作，这样既避免了主数据库服务器（Master）的过载，也有效地利用了从数据库服务器（Slave）的资源



但是amoeba不支持事务，不支持存储过程，暂不支持分库分表，amoeba目前只做到分数据库实例。每个被切分的节点需要保持数据库表结构一致

#### 3.1.1 Amoeba的安装

Amoeba需要java的支持，所以提前配置好java环境。

Ameoba安装非常简单，有现有的压缩包解压后便可以直接使用，现在是具体命令

tar zxf ~/amoeba-mysql-3.0.5-RC.tar.gz –C /usr/local

chown –R root.root /usr/local/amoeba-mysql-3.0.5-RC

ln –s /usr/local/amoeba-mysql-3.0.5-RC /usr/local/amoebad

#### 3.1.2 Amoeba配置

1. 想象Amoeba作为数据库代理层，它一定会和很多数据库保持通信，因此它必须知道由它代理的数据库如何连接，比如最基础的：主机IP、端口、Amoeba使用的用户名和密码等等。这些信息存储在$AMOEBA\_HOME/conf/dbServers.xml中。
2. Amoeba为了完成数据切分提供了完善的切分规则配置，为了了解如何分片数据、如何将数据库返回的数据整合，它必须知道切分规则。与切分规则相关的信息存储在$AMOEBA\_HOME/conf/rule.xml中。
3. 当我们书写SQL来操作数据库的时候，常常会用到很多不同的数据库函数，比如：UNIX\_TIMESTAMP()、SYSDATE()等等。这些函数如何被Amoeba解析呢？$AMOEBA\_HOME/conf/functionMap.xml描述了函数名和函数处理的关系。
4. 对$AMOEBA\_HOME/conf/rule.xml进行配置时，会用到一些我们自己定义的函数，比如我们需要对用户ID求HASH值来切分数据，这些函数在$AMOEBA\_HOME/conf/ruleFunctionMap.xml中定义。
5. Amoeba可以制定一些可访问以及拒绝访问的主机IP地址，这部分配置在$AMOEBA\_HOME/conf/access\_list.conf中
6. Amoeba允许用户配置输出日志级别以及方式，配置方法使用log4j的文件格式，文件是$AMOEBA\_HOME/conf/log4j.xml。

### 3.2 backup-ql

针对不同项目进行日志备份，其中包括两个变量projectName, projectPath

### 3.3 deploy

Deploy目录主要针对上线代码，里面包含的变量有

* deploy\_project
* deploy\_date
* deploy\_serialno
* deploy\_package\_name
* deploy\_target\_path

主要上线过程是，

1. 首先创建war包存放目录及备份目录，
2. 把war包cp到这两目录里，
3. 停tomcat，删除deploy\_target\_path (估计是webapp下所有东西
4. 根据不同的项目创建deploy\_target\_path及下面对应目录
5. 把war包解压到deploy\_target\_path
6. 启动tomcat

注： 这样对tomcat停的时间太长了，所以介意通过软链接的方式来做上线，大概方法可以这样

1. 通过日期创建war包存放目录/src/project/deploy/20170411/1/
2. 把war包scp到这个目录里
3. 解压war包到 /src/project/deploy/20170411/1/webapps/
4. 在webapps下创建对应的项目目录
5. 停止tomcat
6. 删除/usr/local/tomcat/webapps 到/src/project/deploy/20170401/1/webapps 的软链接，创建/usr/local/tomcat/webapps 到/src/project/deploy/20170411/1/webapps/ 的软链接
7. 启动tomcat

3.4 glusterfs

Gluster clien 与server 仅是通过复制rpm包然后安装glusterfs。

yum install glusterfs

yum install glusterfs-client

### 3.4 h5mk-app

主要执行任务顺序为把site.tar 复制到远程服务器上，然后把包解压到 **install\_path** 目录下面然后把site/mugeda\_config 这个目录复制到**install\_path**目录下，再增加nginx的配置，再copy site-installation.sh 这个脚本到远程服务器执行该脚本来安装mogeda。

这里需要安装rsync,docker,glusterfs,

这个roles用于上线h5mk，其中变量有

Mongodb的配置

mongo\_mastet\_ip, mongo\_slave\_ip, db\_name, db\_user,db\_passwd,

oss配置

mugeda\_aliyun\_oss\_host\_endpoint, mugeda\_aliyun\_oss\_key, mugeda\_aliyun\_oss\_bucket, mugeda\_aliyun\_oss\_domain

其它配置

mugeda\_weika\_host, mugeda\_stats\_endpoint, mugeda\_weixin\_ticker\_url, es\_server\_ip, es\_port

mugeda\_partner, mugeda\_partner\_host, enable\_upload\_to\_fs, mount\_point, mugeda\_fs\_domain,

gfs配置

gluster\_server\_ip1,gluster\_server\_ip2,volume\_name, mount\_point,

### 3.5 jdk

Java环境的安装是使用复制二进制包到远程机子上，然后增加环境变量，使用的是1.8版本

### 3.6 keepalived

Keepalived 的task有问题，里面包含的都是有关keepalived配置的相关文件

### 3.7 logrotate

主要是设备定时任务，每天的23点59分的时候 执行logrotate.sh来切分nginx,tomcat 日志

### 3.8 memcached

Memcached 编译安装需要依赖于libevent，安装也比较简单

tar zxf /usr/local/src/memcached-1.4.22.tar.gz

cd /usr/local/src/memcached-1.4.22

./configure –prefix=/usr/local/memcached

make && make install

### 3.9 mongodb

Mongodb也是使用的二进制包，直接解压到/usr/local/目录下便可使用，然后设置下环境变量就完成。这里有一个变量 **mongodb\_version** 用于控制mongodb的版本，

在 mongodb\_arbitor\_install中增加了key，与mongodb的配置，并且启动了mongodb，里面的变量有

1. mongodb\_start\_dir 用于存放安装mongodb安装脚本的位置，
2. mongodb\_data\_dir
3. mongodb\_base\_dir
4. db\_name
5. mongo\_install\_path 这个是写死的是 /usr/local/ 脚本里下一条命令软链接没法做

在 mongodb\_init\_install 中把mongodb\_init.sh复制到远程服务器并执行脚本来对mongodb初始化。 附加变量有

1. mongo\_master\_ip
2. mongo\_slave\_ip
3. mongo\_arbitor\_ip
4. admin\_username
5. admin\_passwd
6. db\_monitor\_username
7. monitor\_passwd
8. db\_name
9. db\_passwd

在mongodb\_master\_install, mongodb\_slave\_install目录内容一致。与mongodb一样

### 3.10 mysql

在mysql-client中仅安装mysql-client 和mysql-shared-compat这两个包，然后把my.cnf复制到/usr/目录下面

在mysql-server中安装mysql-server mysql-client 两个包，并且配置/usr/my.cnf。 在启动mysql后修改密码，如果远程服务器的角色是salve，那么还会执行mysql-config-slave.sh脚本

其中变量有

1. mysql\_new\_password
2. mysql\_server\_hostname
3. mysql\_replication\_user
4. mysql\_slave\_ip
5. mysql\_replication\_password
6. mysql\_server\_ip

mysql\_init 目录主要用于对mysql做双主,并把weika库的数据导到数据库中。变量有

1. mysql\_db\_user
2. mysql\_db\_password
3. mysql\_port
4. mysql\_sock
5. mysql\_db\_name
6. mysql\_master\_ip
7. mysql\_basedir
8. mysql\_sync\_user 用于用主从同步的用户
9. mysql\_sync\_passwd 用于主从同步的用户密码
10. mysql\_database\_user
11. mysql\_passwd
12. mysql\_slave\_ip

### 3.11 nginx

编译安装的nginx有两个版本 1.7.8， 1.10.2 这两个版本于变量 **nginx\_version**控制，其中变量有

1. nginx\_port
2. nginx\_proxy\_pass
3. nginx\_upstream\_servers
4. nginx\_server\_name
5. proxy\_pass 这个和 nginx\_proxy\_pass 相等？

### 3.12 ntp

同步时间根据不同的centos版本号安装ntpdate,然后每20分钟同步一次时间到本地，有变量 **ntpdate\_server\_ip**

### 3.13 python

安装的python版本号有2.7.8,2.7.10，由**python\_version**变量控制

### 3.14 redis

Redis 部署主要通过配置redis.conf 搭建redis主从，redis集群的部署。

变量有 **redis\_master\_ip**

3.15 rsyncd

增加rsync配置，增加启动脚本，并加到自启动，根据不同的项目进行备份

变量有

1. projectPath
2. rsyncd\_port

rsync-init目录可以合并到rsyncd里面

### 3.15 system

主要是关闭selinux, iptables,关闭 transparent hugepages.修改soft limit 和hard limit

### 3.16 tomcat

使用二进制包安装tomcat，增加启动脚本，配置环境变量，

变量有

1. tomcat\_xms
2. tomcat\_xmx
3. tomcat\_jmx\_server\_ip
4. tomcat\_msm\_memcached\_nodes
5. tomcat\_manager\_allow\_ip

### 3.17 xtrabackup

用于安装xtrabackup，通过yum install percona-xtrabackup 安装

### 3.18 yum\_repo

增加私有仓库的repo里面有两个变量

1. yum\_server\_ip
2. yum\_server\_ip2

考虑到私有仓库可能可能处于两个局域网之间，所以需要两个ip,如果有dns解析的话这里就可以写成域名并写死，然后通过dns来切换ip

### 3.19 zabbix

Zabbix-agent

安装zabbix-agent,增加自定义的key和脚本。主要的变量有

1. zabbix\_server 这里可以是server的ip也可以是proxy的ip
2. zabbix\_server\_active 同上
3. zabbix\_pdmi\_application

zabbix-proxy

用于安装zabbix-proxy 代理server收集agent的数据，并提交给server. 其中需要安装mysql，主要变量有

1. zabbix\_proxy\_server
2. zabbix\_proxy\_dbname
3. zabbix\_proxy\_dbuser
4. zabbix\_proxy\_dbpass

zabbix-server

用于安装并配置zabbix-server, 其中包含了mysql， 主要变量有

1. zabbix\_server\_dbhost
2. zabbix\_server\_dbname
3. zabbix\_server\_dbpassword
4. zabbix\_server\_dbuser
5. zabbix\_server\_javagateway (ip)
6. zabbix\_server\_javagetewayPort
7. zabbix\_server\_weixin\_corpid 如果需要weixin报警，这两个地方需要改。
8. zabbix\_server\_weixin\_secret