ASDN

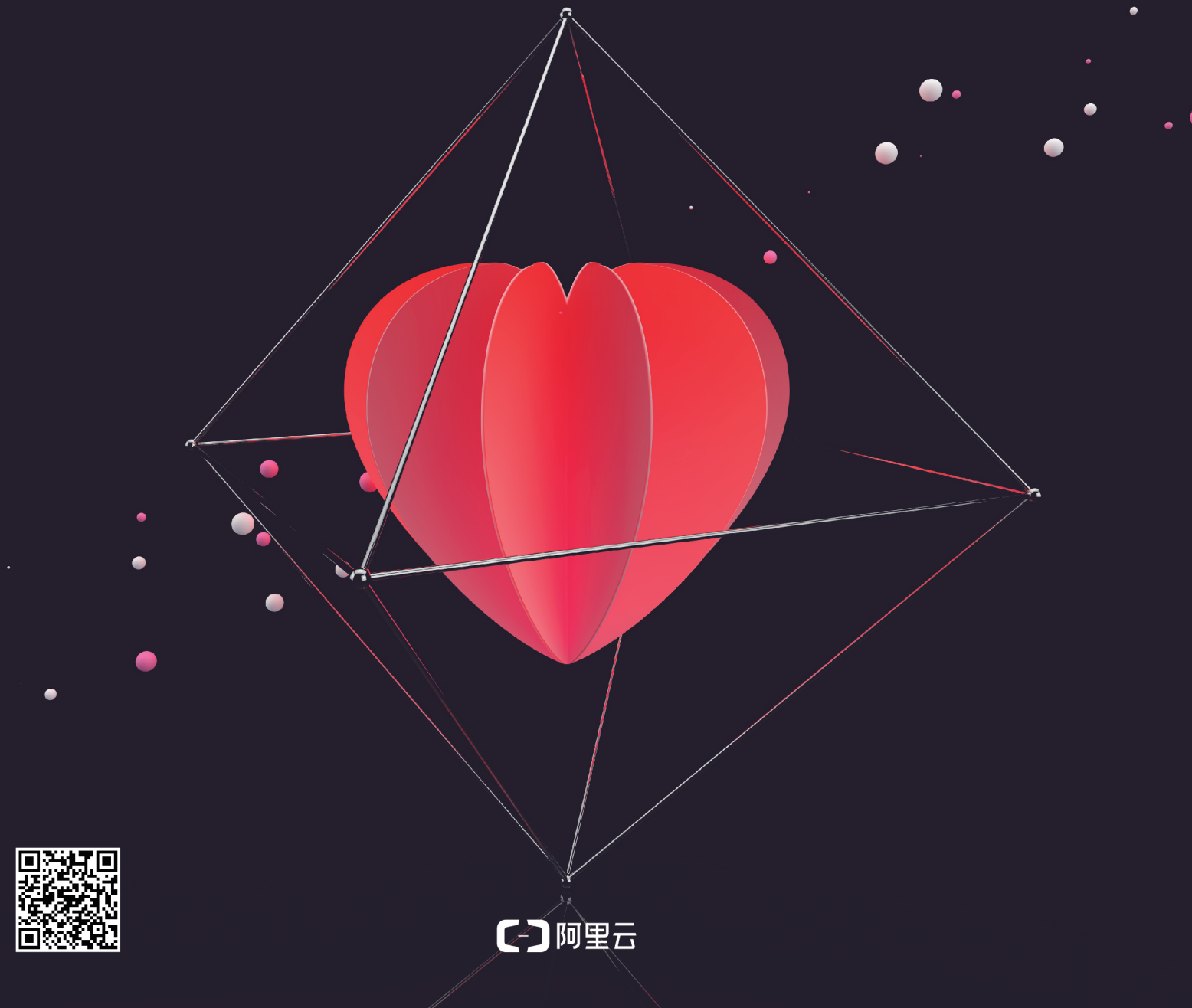
AlibabaCloud Software Developer Network

#4

阿里云ASDN月刊

总第4期

2017.09



阿里云

为 了 无 法 计 算 的 价 值

阿里云 ASDN 月刊

AlibabaCloud Software Developer Network

主编　净业

顾问　广慧　路平　风驰

编委　王虎　彭飞

设计　锦昀　野一　彭柳

主办部门　阿里云用户体验团队

合作方　阿里云飞天体验技术部　阿里云研究中心

云栖社区　云市场

阿 里 云 说

王磊

数心网络
创始人兼CEO
阿里云首批MVP

“

未来一定属于最善于使用数据的企业。但是，数据只有深度理解和结合业务，才能发挥出巨大的价值。大数据专家，既要掌握最顶尖的精通技术，也必须是最理解客户，最懂业务的人。

”

目 录

阿里云上部署开源PaaS平台 Cloud Foundry实战

思皓
资深研发工程师

一、Cloud Foundry介绍和使用场景分析

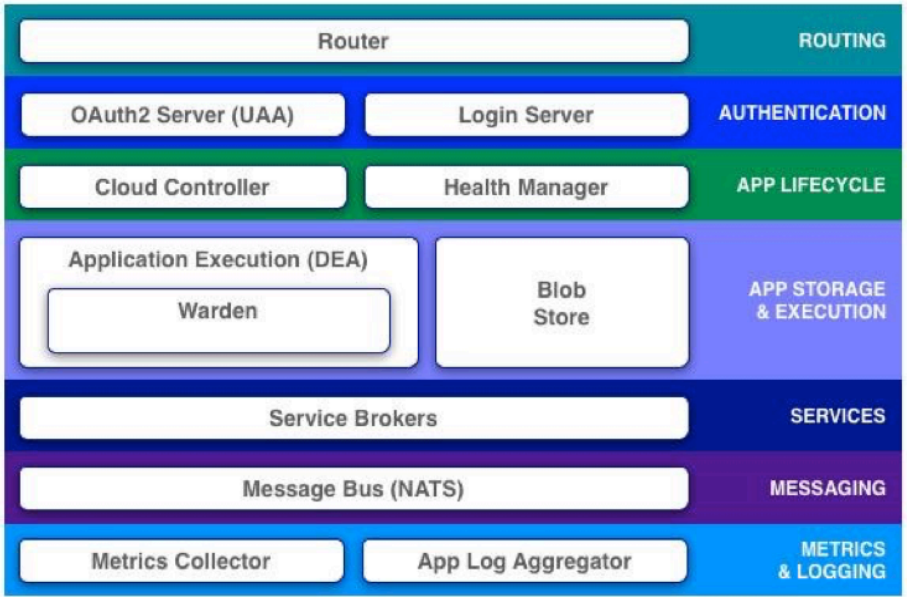
关键词：PaaS、开源

Cloud Foundry 是业界第一个开源的PaaS，号称工业界唯一的云应用平台，支持多种框架、语言、运行时环境、云平台及应用服务，兼容多种基础设施云，提供多种开发框架和应用服务。

目前开源支持部署Cloud Foundry的基础设施云包括AWS、Azure、OpenStack等等，CF在开发框架上支持Java、.NET、Ruby等等，有很强的灵活性。

Cloud Foundry组件（v2版本）

Cloud Foundry是由相对独立的多个模块构成的分布式系统，每个模块单独存在和运行，支持水平扩展，各模块之间通过消息机制进行通信。

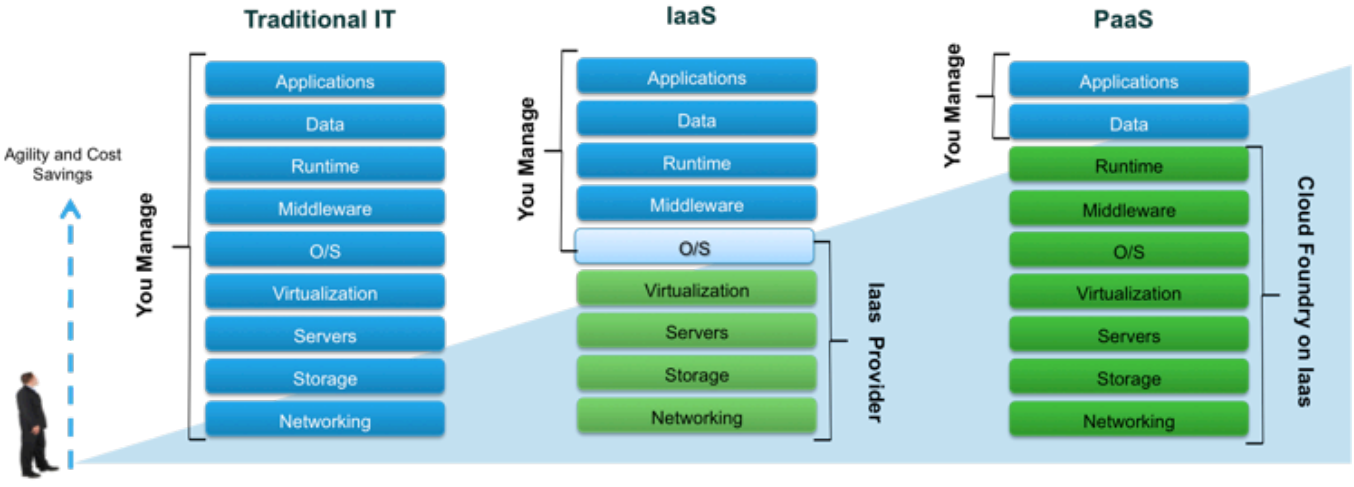


官网参考：http://docs.cloudfoundry.org/concepts/architecture/

二、Cloud Foundry使用场景分析

Cloud Foundry是一种平台即服务（PaaS）

IaaS帮助开发者和客户解决了基础设施的问题，作为PaaS，Cloud Foundry在云上使开发人员能够在几秒钟内进行应用程序的部署和扩展，无需担心任何基础架构的问题；让开发者的视角更高，只需要关注应用和数据。



1. 分布式系统

Cloud Foundry是完全模块化的分布式系统，每个模块单独存在和运行，通过消息总线进行通讯，保证了平台的可靠性。同时，分布式架构降低了系统耦合度，便于弹性动态扩展。

2. 弹性伸缩架构

部署组件上，Cloud Foundry通过自动化部署工具Bosh实现了Cloud Foundry各个组件的自动化部署，各个组件支持实时自动横向扩容。

应用上，Cloud Foundry核心组件根据应用负载情况，动态加载应用实例，应用实例支持实时水平扩展。

3. 运维自动化

开发和运维人员在Cloud Foundry上部署、管理应用，完全不用担心基础架构的问题，结合Cloud Foundry强大的交互、日志能力，简化日常运维操作。

Cloud Foundry的监控组件，发现组件实例异常时，支持自动故障恢复。

4. 简化应用部署

一键式应用快速部署，支持多种应用开发框架，包括Spring、.NET、RubyonRails，Node.js等，支持通过buildpack扩展运行不同语言应用的能力。

5. 支持多种应用服务

CloudFoundry本身支持多种数据服务，包括MySQL、mongodb、PostgreSQL等；同时支持通过service broker组件扩展多种应用服务能力，包括数据库、中间件、缓存、云存储等。

三、阿里云上部署Cloud Foundry实战

环境说明：（本文使用软件版本和环境的说明，涉及到的概念下文会详细解释）

阿里云region：华东1

ruby 版本：2.3.0

bosh-init 版本：0.0.96

Bosh CLI 版本：1.3262.4.0

部署流程：

在阿里云上部署Cloud Foundry，部署流程分为三大部分，分别为：

1. 开通阿里云环境资源：包括准备阿里云账号，开通相关云产品资源等等

2. 部署Bosh：安装bosh-init，使用bosh-init部署Bosh

3. 部署Cloud Foundry：使用Bosh部署Cloud Foundry

本文后续内容会根据三步部署流程详细讲述。

1.开通阿里云环境资源

Cloud Foundry需要部署在VPC环境下，需要开通VPC资源，并创建虚拟交换机，规划好部署Cloud Foundry的内网网段。同时，也需要创建一对Access Key ID和Access Key Secret，也可以使用现有的Access Key ID和Access Key Secret对。

创建专有网络VPC

网段没有限制，笔者选择了10.0.0./8网段。

创建Access Key ID和Access Key Secret

创建虚拟交换机

可用区和网段没有限制，方便起见，笔者选择了华东1可用区E，网段为10.0.0.0/25

2.部署Bosh

Bosh介绍

Bosh是一个统一了平台即服务软件（如Cloud Foundry）的发布、部署和生命周期管理的自动化配置部署工具。Bosh的作用之一就是部署Cloud Foundry，部署Cloud Foundry之前，首先要部署Bosh。

在讲怎么在阿里云上部署Bosh之前，首先需要了解Bosh中的几个基本概念。

Bosh官网参考：https://bosh.io/docs

- CPI

CPI全称Cloud Platform Interface，是Bosh对外开放的一组标准化接口，在IaaS上部署Bosh，需要实现这组接口，-Bosh用 CPI 对 IaaS 的资源进行管理，包括创建虚拟机，释放虚拟机，等等.....

在阿里云上部署Bosh，需要用阿里云OpenAPI实现CPI。

CPI参考：https://bosh.io/docs/cpi-api-v1.html

CPI API v1完整接口列表：

- Stemcell management
  - `create_stemcell`
  - `delete_stemcell`
- VM management
  - `create_vm`
  - `delete_vm`
  - `has_vm`
  - `reboot_vm`
  - `set_vm_metadata`
  - `configure_networks`
  - `calculate_vm_cloud_properties` (Experimental)
- Disk management
  - `create_disk`
  - `delete_disk`
  - `has_disk`
  - `attach_disk`
  - `detach_disk`
  - `get_disks`
- Disk snapshots
  - `snapshot_disk`
  - `delete_snapshot`
  - `current_vm_id`

- bosh-init

bosh-init是Bosh官网提供的一个开源工具，它的作用就是用来部署Bosh。

bosh-init参考：https://bosh.io/docs/using-bosh-init.html

• stemcell

stemcell是虚拟机镜像，预装了部署过程中需要的组件（其中最重要的组件是Bosh Agent），官方的描述是：A stemcell is a versioned Operating System image wrapped with IaaS specific packaging. Bosh用CPI创建的虚拟机，用stemcell镜像启动。

stemcell参考：https://bosh.io/docs/stemcell.html

• release

release是Bosh中一个安装部署包的概念，包含了所有安装分布式系统需要的源代码、配置文件、脚本文件等等，官网描述为：A release is a versioned collection of configuration properties, configuration templates, start up scripts, source code, binary artifacts, and anything else required to build and deploy software in a reproducible way.

例如，安装Bosh，我需要一个Bosh-release；用Bosh安装Cloud Foundry，我需要一个cf-release。

所有release都有版本迭代，都可以在Bosh官网找到。

release参考：https://bosh.io/docs/release.html

• Deployment

一个Deployment是一组VM的集合，由指定的stemcell镜像启动，用于部署对应release的系统。官方描述为：A deployment is a collection of VMs, built from a stemcell, that has been populated with specific releases and disks that keep persistent data. These resources are created based on a manifest file in the IaaS and managed by the BOSH Director, a centralized management server.

在Bosh的概念里，一个Deployment对应一个release，Deployment的具体配置，写在Deployment manifest里，在部署过程中提供。

Deployment参考：https://bosh.io/docs/deployment.html

阿里云上部署Bosh

1.创建ECS

通过阿里云控制台创建一个ECS，用于安装bosh-init。方便起见，下文以init指代这台ECS。

推荐配置：

规格：2核4G及以上

镜像：Ubuntu 14.04 64位

系统盘：40GB以上，高效云盘

网络类型：VPC实例，选择在上一步创建好的VPC和虚拟交换机，公网IP可以选择不分配。

2.给init配置公网IP

因为init是VPC实例，需要给init绑定弹性公网IP，让init可以通过公网访问。

按需购买弹性公网IP，并给init绑定弹性公网IP

3.安装bosh-init

参考文档：http://bosh.io/docs/install-bosh-init.html

(1)SSH登陆到init这台ECS

(2)下载bosh-init，下载地址见：http://bosh.io/docs/install-bosh-init.html

(3)执行权限

```
chmod +x ~/Downloads/bosh-init-*
```

(4)移动到/usr/local/bin

```
sudo mv ~/Downloads/bosh-init-* /usr/local/bin/bosh-init
```

(5)验证安装成功

```
bosh-init -v
```

(6)安装对应环境，笔者使用的是Ubuntu的环境

```
$ sudo apt-get install -y build-essential zlibc zlib1g-dev ruby ruby-dev openssl libxslt-dev libx-
```

ml2-dev libssl-dev libreadline6 libreadline6-dev lib-yaml-dev libsqlite3-dev sqlite3

(7)确保已经安装Ruby 2+环境

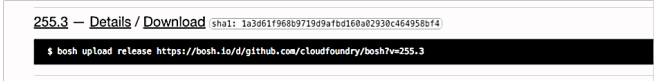
```
root@bosh-init:~# bosh-init -v
version 0.0.96-a26a88a-2016-07-19T23:13:15Z
root@bosh-init:~# ruby -v
ruby 2.3.0p0 (2015-12-25 revision 53290) [x86_64-linux]
```

4.下载bosh-release

bosh-release是开源的，可以从Bosh官网下载：

https://bosh.io/releases/github.com/cloudfoundry/bosh?all=1

笔者使用的是255.3版本的bosh-release



5.下载cpi-release

CPI近期会开源，请关注：

https://github.com/alibaba/opstools

6.配置manifest

根据以下模板，新建一个manifest文件bosh.yml，在模板中填充阿里云资源相关的内容。

完整manifest模板请参考：

https://yq.aliyun.com/articles/104841?spm=5176.100239.0.0.LIpE32

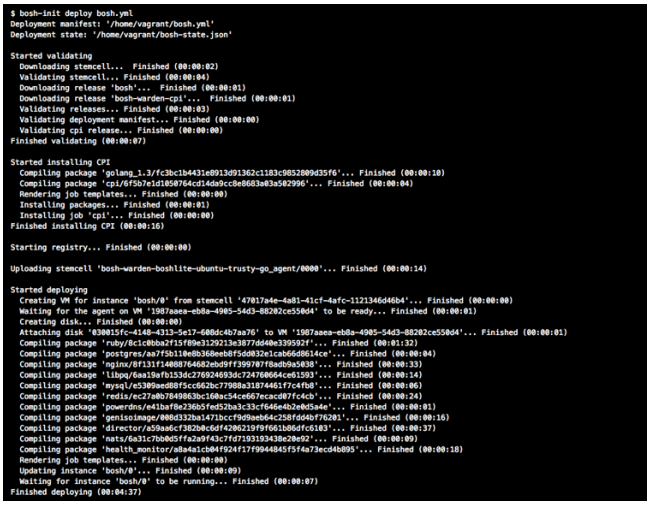
```
---
name: bosh

releases:
- name: bosh
  url: file:///root/downloads/bosh-255.3.tgz
- name: bosh-aliyun-cpi
  url: file:///root/downloads/bosh-aliyun-cpi.tgz

resource_pools:
- name: vms
  network: private
  cloud_properties:
    instance_type: ecs.n4.large  # <--- 实例规格

networks:
- name: private
  type: manual
  subnets:
  - range: 10.0.0.0/8
    gateway: 10.0.0.1
    cloud_properties: {
      SecurityGroupId: SECURITY_GROUP_ID, #<--- 安全组ID
      VSwitchId: VSWITCH-ID  # <--- 虚拟交换机ID
    }
  - name: public
    type: vip

jobs:
- name: bosh
  instances: 1
  templates:
  - {name: nats, release: bosh}
  - {name: redis, release: bosh}
  - {name: postgres, release: bosh}
  - {name: blobstore, release: bosh}
  - {name: director, release: bosh}
  - {name: health_monitor, release: bosh}
  - {name: registry, release: bosh}
  - {name: aliyun_cpi, release: bosh-aliyun-cpi}

  resource_pool: vms
  networks:
  - name: private
    static_ips: [10.0.0.2]  # <--- ECS内网IP
    default: [dns, gateway]
  - name: public
    static_ips: [STATIC_IP]  # <--- 弹性公网IP

  properties:
    aliyun: &aliyun
      access_key_id: ACCESS_KEY_ID # <--- 阿里云Access Key ID
      access_key: ACCESS_KEY # <--- 阿里云Access Key Secret
      default_key_name: bosh
      default_security_groups: [bosh]
      region_id: cn-hangzhou # <--- 阿里云 Region
```

7.部署Bosh

执行部署命令：

bosh-init deploy bosh.yml

一个部署过程的示例：



8.验证Bosh

我们用Bosh CLI验证Bosh是否部署成功。Bosh CLI是Bosh官方提供的，用于和Bosh交互的命令行工具。在部署完成Bosh之后，用Bosh CLI和Bosh交互，执行相关命令，进行Cloud Foundry的部署。

安装Bosh CLI：https://bosh.io/docs/bosh-cli.html

Bosh CLI可以安装在任意一台ECS上，或者本地主机上。

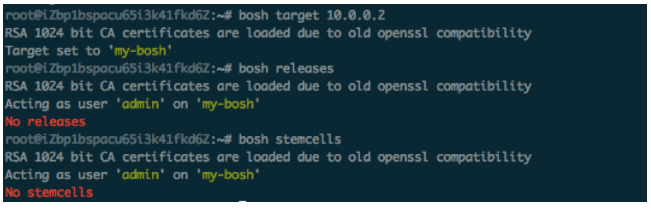
验证Bosh：

执行Bosh CLI命令：bosh target 10.0.0.2

笔者的Bosh CLI安装在同一个安全组的ECS上，因此可以和部署Bosh的ECS进行内网通讯，直接通过内网IP，target到Bosh。如果需要通过公网通讯，需要给部署Bosh的ECS绑定弹性公网IP，或者使用NAT网关产品，保证网络能通。

如图显示，成功连接到目标Bosh，验证Bosh成功。

bosh releases、bosh stemcells这两条命令，使用Bosh部署Cloud Foundry的时候会用到，我们下节细讲。



3.部署Cloud Foundry

基本概念

首先我们也需要了解用Bosh部署Cloud Foundry过程中的几个基本概念。

- Bosh CLI

上一节讲到，我们需要使用已经部署成功的Bosh来部署Cloud Foundry，通过Bosh CLI和Bosh进行交互，执行相关部署命令。因此，使用Bosh部署Cloud Foundry之前，首先需要了解Bosh CLI命令的使用。

Bosh CLI几个基本命令：

1.连接到指定Bosh

bosh target [DIRECTOR_URL]

2.列出当前release仓库中所有的release

bosh releases

3.上传release到Bosh的release仓库，只有仓库里的release可以用于部署

bosh upload release [RELEASE_FILE]

4.列出当前stemcell仓库里的所有stemcell

bosh stemcells

5.上传stemcell到Bosh的stemcell仓库，只有仓库里的stemcell可以用于部署

bosh upload stemcell STEMCELL_PATH

bosh upload stemcell STEMCELL_URL

6.列出当前所有的Deployment

bosh deployments

7.列出当前Deployment的信息

bosh deployment

8.切换到指定manifest对应的Deployment

bosh deployment [MANIFEST_PATH]

9.执行部署当前Deployment

bosh deploy

Bosh CLI官网参考：http://bosh.io/docs/sysadmin-commands.html

- CPI

在上一步部署Bosh中，已经部署好的Bosh中已经包含了CPI组件，因此在部署Cloud Foundry中不再需要CPI-release

- stemcell

和上一步部署Bosh中类似。

- cf-release

CloudFoundry官网提供的压缩包，包含Cloud Foundry所有组件的源码。

• Deployment manifest

Deployment的配置文件，主要描述了用哪个stem-cell，用哪个cf-release，需要部署哪些CF组件，需要多少VM，VM规格信息，VM的IP信息，具体哪个VM部署哪个CF组件，等等……

部署Cloud Foundry流程

1.下载cf-release

cf-release是开源的，可以从Bosh官网下载：

http://bosh.io/releases/github.com/cloudfoundry/cf-release?all=1

笔者部署用的是215版本的cf-release

2.登陆到安装有Bosh CLI的机器

我们需要用Bosh CLI和Bosh进行交互，执行部署Cloud Foundry命令

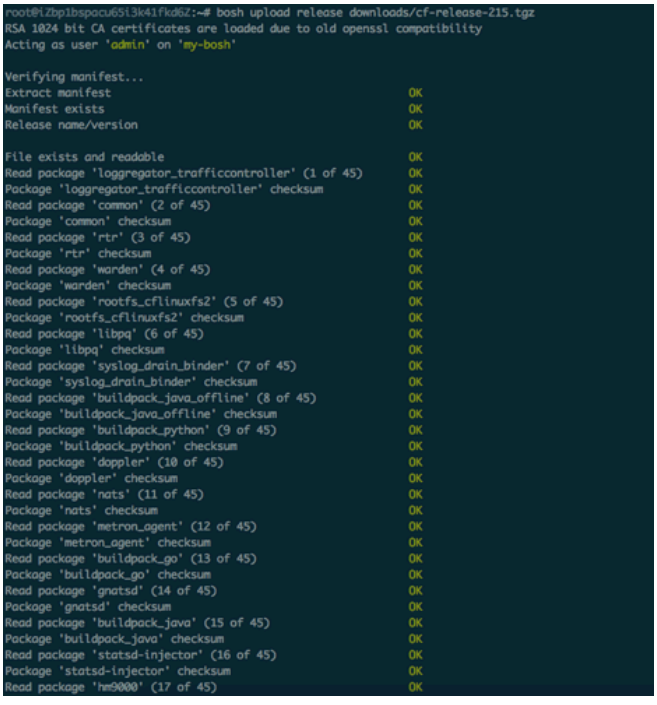
3.用Bosh CLI连接到部署好的Bosh

bosh target 10.0.0.2

4.上传cf-release

上传下载完成的cf-release，执行以下Bosh CLI命令：

bosh upload release <release_file>



5.配置manifest

根据以下模板，新建一个manifest文件cf.yml，在模板中填充阿里云资源相关的内容。

完整manifest模板请参考：

https://yq.aliyun.com/articles/104841?spm=5176.100239.0.0.LIpE32

```
---
name: ali-cf
director_uuid: BOSH_DIRECTOR_UUID  # <--- Bosh Director UUID

releases:
- {name: cf, version: 215}

networks:
- name: private
  type: manual
  subnets:
  - range: 10.0.0.0/8
    gateway: 10.0.0.1
    #dns: [10.0.0.2]
    reserved: [“10.0.0.2”]
    static: [“10.0.0.3 - 10.0.0.100”]
    cloud_properties:
      SecurityGroupId: SECURITY_GROUP_ID #<--- 安全组ID
      VSwitchId: VSWITCH_ID # <--- 虚拟交换机ID

resource_pools:
- name: small_ecs
  network: private
  cloud_properties:
    instance_type: ecs.n1.tiny # <--- 实例规格

compilation:
  workers: 1
  network: private
  reuse_compilation_vms: true
  cloud_properties:
    instance_type: ecs.n1.medium # <--- 实例规格

update:
```

```
  canaries: 1
  max_in_flight: 1
  serial: false
  canary_watch_time: 30000-600000
  update_watch_time: 5000-600000

jobs:
- name: nats
  instances: 1
  resource_pool: small_ecs
  templates:
  - {name: nats, release: cf}
  networks:
  - name: private
    static_ips: [10.0.0.20] # <--- ECS内网IP

- name: nfs
  instances: 1
  persistent_disk: 50
  resource_pool: small_ecs
  templates:
  - {name: debian_nfs_server, release: cf}
  networks:
  - name: private
    static_ips: [10.0.0.21] # <--- ECS内网IP
......
```

6.新建cf.yml对应的Deployment

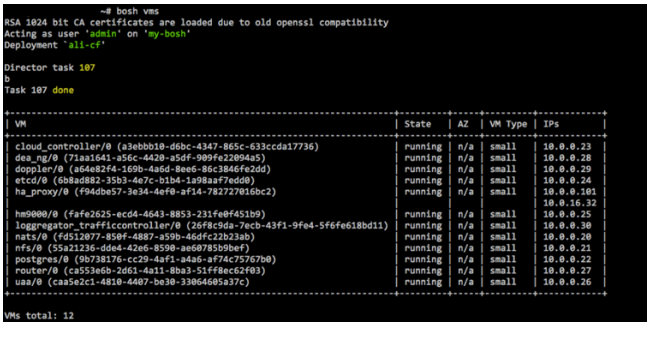
```
bosh deployment cf.yml
```

7.执行部署命令

```
bosh deploy
```

8.验证Cloud Foundry

部署完成之后，执行以下命令，查看Cloud Foundry部署详情：bosh vms



```
~# bosh vms
RSA 1024 bit CA certificates are loaded due to old openssl compatibility
Acting as user 'admin' on 'my-bosh'
Deployment `ali-cf'

Director task 107
b
Task 107 done

+-----------------------------------------------------------------------+---------+----+---------+-------------+
| VM                                                                    | State   | AZ | VM Type | IPs         |
+-----------------------------------------------------------------------+---------+----+---------+-------------+
| cloud_controller/0 (a3ebbb10-d6bc-4347-865c-633ccda17736)             | running | n/a | small  | 10.0.0.23   |
| dea_ng/0 (71aa1641-a56c-4420-a5df-909fe22094a5)                       | running | n/a | small  | 10.0.0.28   |
| doppler/0 (a64e82f4-169b-4a6d-8ee6-86c3846fe2dd)                      | running | n/a | small  | 10.0.0.29   |
| etcd/0 (6b8ad882-35b3-4e7c-b1b4-1a98aaf7edd0)                         | running | n/a | small  | 10.0.0.24   |
| ha_proxy/0 (f94dbe57-3e34-4ef0-af14-782727016bc2)                     | running | n/a | small  | 10.0.0.101  |
|                                                                       |         |     |        | 10.0.16.32  |
| hm9000/0 (fafe2625-ecd4-4643-8853-231fe0f451b9)                       | running | n/a | small  | 10.0.0.25   |
| loggregator_trafficcontroller/0 (26f8c9da-7ecb-43f1-9fe4-5f6fe618bd11) | running | n/a | small  | 10.0.0.30   |
| nats/0 (fd512077-850f-4887-a59b-46dfc22b23ab)                         | running | n/a | small  | 10.0.0.20   |
| nfs/0 (55a21236-dde4-42e6-8590-ae607859b0ef)                          | running | n/a | small  | 10.0.0.21   |
| postgres/0 (9b738176-cc29-4af1-a4a6-af74c75767b0)                     | running | n/a | small  | 10.0.0.22   |
| router/0 (ca553e6b-2d61-4a11-8ba3-51ff8ec62f03)                       | running | n/a | small  | 10.0.0.27   |
| uaa/0 (caa5e2c1-4810-4407-be30-33064605a37c)                          | running | n/a | small  | 10.0.0.26   |
+-----------------------------------------------------------------------+---------+----+---------+-------------+

VMs total: 12
```

可以看到，Cloud Foundry的核心组件都已经部署成功，Cloud Foundry能成功运行。

至此，Cloud Foundry在阿里云上部署成功。

经典网络到VPC网络的迁移方案

墨华
阿里云技术专家

一、概述

专有网络VPC（Virtual Private Cloud）正受到越来越多用户的欢迎，已经成为云上用户的首选网络类型，也是阿里云默认推荐的网络类型，其具有安全隔离、访问控制灵活、软件定义网络和丰富的网络连接等优势。然而，云上还有很多存量用户在使用经典网络，这些用户如果需要使用VPC的功能和特性，则需要考虑从经典网络迁移到VPC。本文将介绍相关的迁移方案和迁移后的优化。主要内容分两部分，第一部分主要介绍目前阿里云对经典网络迁移到VPC的产品支持，第二部分主要是结合VPC的特点，介绍客户业务在迁移中和迁移后的一些优化建议和措施。

二、产品支持

对于经典网络迁移到VPC，阿里云提供三种产品层面的支持，分别是混挂和混访、ClassicLink和单ECS迁移。这三个方案可以独立使用，也可以组合使用，以满足不同的迁移场景，下面逐个进行介绍。

2.1 混挂和混访

混挂和混访方案是一种系统平滑迁移方案，即用户通过在VPC中新建ECS等云产品实例，然后将系统平滑迁移到VPC。当所有系统都迁移到VPC后，再将经典网络内的资源释放，从而完成经典网络到VPC的迁移。

2.1.1 混挂

首先来介绍一些混挂，顾名思义即混合挂载，具体指在同一个SLB实例下，可以同时挂载经典网络ECS和VPC网络ECS，简化架构图如下

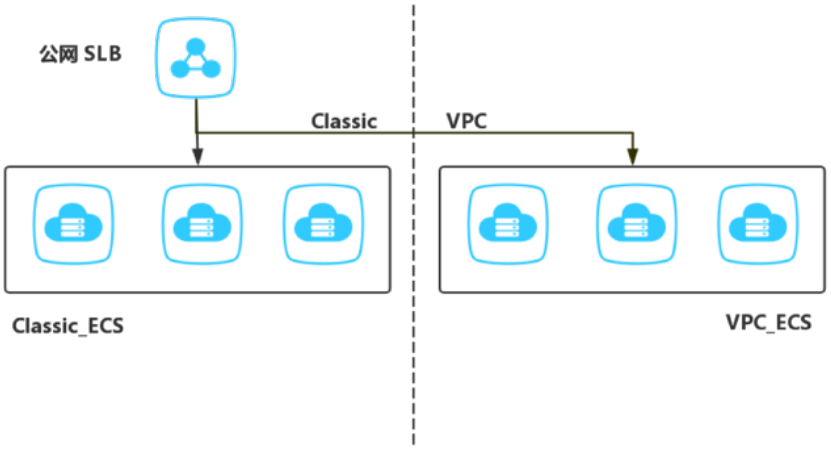


图2-1-1 混挂方案示意图

通过这种方式，可以在不影响原有业务访问的情况下，通过增加VPC环境的新部署的ECS服务器，将访问流量通过SLB负载均衡分配到新的应用服务器上，实现业务流量从经典网络到VPC的平滑切换，支持4层TCP&UDP、7层HTTP和HTTPS协议，适合于接入层流量和应用服务器迁移。

2.1.2 混访

混访，顾名思义即混合访问，具体指在同一个云产品实例，既可以在经典网络下方访问，又可以VPC网络下访问，常见的比如OSS、RDS等产品。现阶段下混访需要通过两个不同的域名来实现，分别对应Classic和VPC网络， 后续产品上会统一访问域名规则，做到无论何种环境，均可通过单一域名来访问，对业务侧做到访问切换无感知。此方案适用于数据库和存储产品迁移，由于RDS默认只支持单一网络环境访问，所以目前RDS需要主动来开启混访，开启后将提供经典网络和VPC同时访问的支持能力。

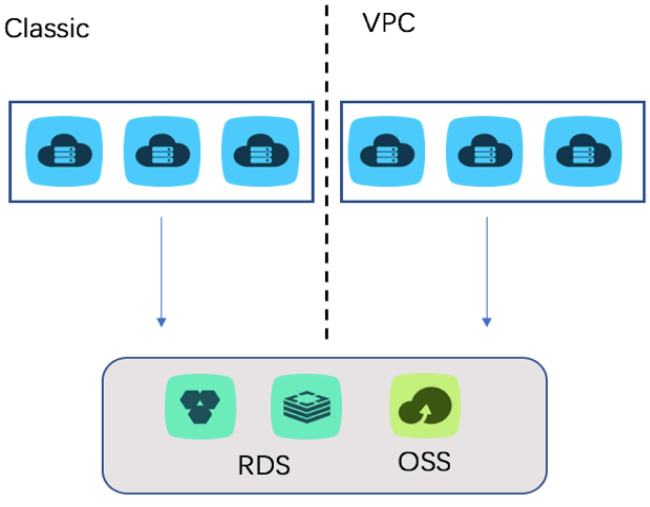


图2-1-2 混访方案示意图

2.2 ClassicLink

ClassicLink方案可以将一台经典网络ECS链接到一个VPC，从而实现经典网络ECS直接通过私网访问VPC内的云资源（如ECS，RDS，SLB等）。通过这种方式，能够在网络层解决同一个地区经典网络ECS与VPC网络的应用服务器、数据库和存储等业务的互联互通问题。由于是通过内网直接打通，因此在网络上保持了内网特性，相比于通过公网交互，这种方式在网络延时、带宽和可靠性方面均和内网保持一致，保证在迁移过程中不会因为网络变化带来业务上的性能损失。

ClassicLink适合于在业务迁移过程中，对一些关键但又无法通过分布式部署来解决迁移过程中两个环境都需要访问的情况，比如某些配置系统，认证服务等。

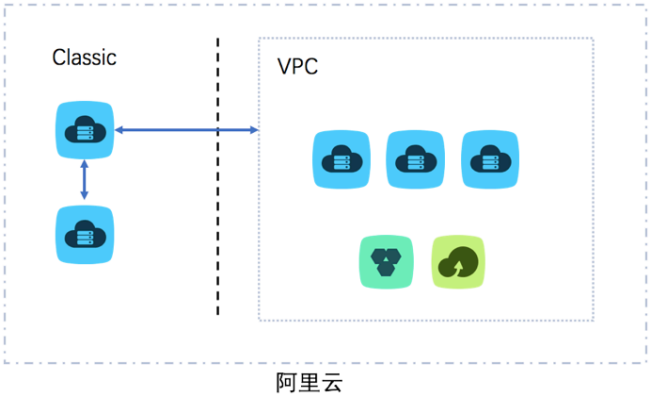


图2-2 ClassicLink方案示意图

2.3 单ECS迁移

单ECS迁移方案，是通过底层技术，将经典网络ECS通过本方案直接迁移到专有网络（需重启ECS）。相比于前两种方案，这种迁移技术的优势在于不需要重新部署应用服务，迁移过程无需创建镜像、重新购买等步骤。由于需要重启ECS，并且底层数据迁移需要一定的传输时间，因此这种方案对业务来说会造成一定的服务中断，需要合理评估后使用。

使用本迁移方案将经典网络实例迁移到VPC分两步进行：

1.预约迁移

需要在ECS控制台的待办任务中选择预约实例迁移。默认情况下，符合迁移条件的实例会显示在预约窗口列表中。

注意：当前预约迁移必须先通过客户经理申请，请将需要迁移的实例列表提供给客户经理。待阿里云处理后，您方可通过控制台进行预约迁移。

2.正式迁移

完成预约后，本迁移工具将在预约时间点开始迁移。迁移完成后，您将收到迁移成功的短信消息提醒。

三、VPC优化

经典网络和VPC网络属于两种不同的网络环境，经典网络作为最初的网络类型，其网络的划分、IP分配都是由云环境自动完成，不具备自主定义的能力，对于有内网自主规划能力和需求的客户来说是非常不方便的。专有网络VPC（Virtual Private Cloud）是基于阿里云构建的一个隔离的网络环境，专有网络之间逻辑上彻底隔离，能够在自己定义的虚拟网络中使用阿里云资源，可以完全掌控自己的虚拟网络，例如选择自己的IP地址范围、划分网段、配置路由表和网关等，从而实现安全而轻松的资源访问和应用程序访问。此外，也可以通过专线或VPN等连接方式将专有网络与传统数据中心相连，形成

一个按需定制的网络环境，实现应用的平滑迁移上云和对数据中心的扩展。

从经典网络迁移到VPC不仅仅是将业务、数据、存储和日志等完全平行迁移过来，在这个基础之上，可以对整个业务的内网进行重新规划和分配，利用VPC的优势，做到内部系统隔离，网络资源整合，安全升级，打造自主、高效、灵活的业务环境。

3.1 网络架构设计

VPC支持内网网络的自定义，用户可以根据自己的需求来划分不同的子网和授权规则。VPC支持的内网包括192.168.0.0/16，172.16.0.0/12，10.0.0.0/8以及它们的子网。同一个VPC内网段不允许重复，理论上不同VPC之间可以使用重复网段，但考虑到后续不同VPC之间的访问，原则上推荐所有的VPC都不要划分重复的网段地址，以免后续建立访问时内网地址冲突。

3.1.1网络规划

对于VPC内部的网络，每个网段都是分配到具体的vswitch上，每个云产品实例分配到某一个具体的vswitch下，并获得这个vswitch内的一个具体IP地址。

在访问控制方面，VPC环境和经典网络类似，支持ECS安全组，SLB、RDS白名单等多种措施，更具业务的需求来灵活配置访问策略。由于VPC网络的自定义特性，建议对功能相同的业务模块，分配到固定的vswitch下，一方面，后续新加入的机器会自动分配在固定的网段内，地址连续性强，便于管理。另一方面，对于ECS的安全组和RDS的白名单，可以将vswitch对应的网段整段加入，后续新加入的实例会默认拥有对应的访问权限，减少了频繁修改规则带来的工作量。建议参考以下规则：

- 业务部署相同模块部署在同一个子网中（即vswitch中）；
- 访问授权针对业务模块最小化，即只给对应的子网（vswitch）进行授权，兼顾安全和维护便利性；
- 不同业务环境分别部署在不同的VPC中，如开发测试环境、预发环境和生产环境。

例如下图所示，数据库模块只允许业务模块2访问，则RDS白名单中可将192.168.2.0/24网段加入到白名单中，后续业务模块2新增ECS实例，默认即可访问数据库，无需额外操作。对于业务模块1和业务模块2，只需在ECS安全组中分别添加对方的网段地址，后续新增实例即可互相访问。

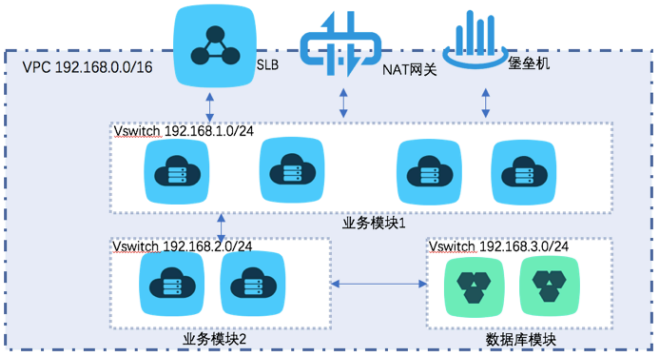


图3-1 VPC业务架构示意图

3.1.2 公网访问

前面介绍了VPC内部的访问控制，这部分我们介绍VPC在公网访问和安全方面的内容。VPC中实例默认情况下不分配公网IP，所以不具备公网访问能力，正是因为如此，VPC内部资源默认和公网保持隔离，极大地减少甚至杜绝了了来自公网的攻击。在对外服务提供以及公网访问方面，我们推荐使用SLB和NAT网关。VPC环境下SLB和经典网络环境SLB功能类似，此处不再赘述。

公网出口和特殊对外端口，建议全部使用NAT网关完成，NAT网关（NAT Gateway）是一款企业级的VPC公网网关，提供 NAT 代理（SNAT、DNAT）、10Gbps 级别的转发能力、以及跨可用区的容灾能力，借助于NAT网关的SNAT能力，可以为VPC内部实例提供可靠公网访问能力。如果业务上有特殊端口需要开发到公网，也可以使用NAT网关的DNAT功能，实现IP+端口的转发功能。

在安全管理方面，推荐使用堡垒机服务，通过集中的访问管控、最小化授权策略、角色控制等确保访问安全。另外，堡垒机也提供操作审计、安全双因子认证等保证入口安全。

对于一些个性化的需求，阿里云也提供EIP弹性公网IP，可以直接绑定到ECS实例上，提供和经典网络类似的公网访问能力，此操作会将ECS暴露在公网，尽量避免使用。

公网安全访问建议遵循以下原则：

- 实例不开启公网访问；
- 对外服务通过SLB提供，只开启业务需要的端口，如有必要，开启白名单；
- VPC出口使用NAT网关SNAT功能，特殊服务对外使用DNAT功能，无特殊需求不使用EIP；
- 运维管理使用堡垒机，最小化授权，开启双因子认证。

3.2 扩展能力

VPC网络与经典网络在扩展能力上有很大差异，借助于阿里云高速通道、VPN网关等产品，可以方便快捷的将多个地区VPC进行内网打通，提供稳定可靠的访问能力。

如下图所示，比如在华东1、华北2、华南1三个地域分别有VPC1、VPC2和VPC3三个VPC。VPC1和VPC2通过高速通道内网互通，VPC3目前没有和其他VPC通信的需求，将来可能需要和VPC2通信。另外，您在上海还有一个自建IDC，需要通过高速通道（专线功能）和华东1的VPC1私网互通。

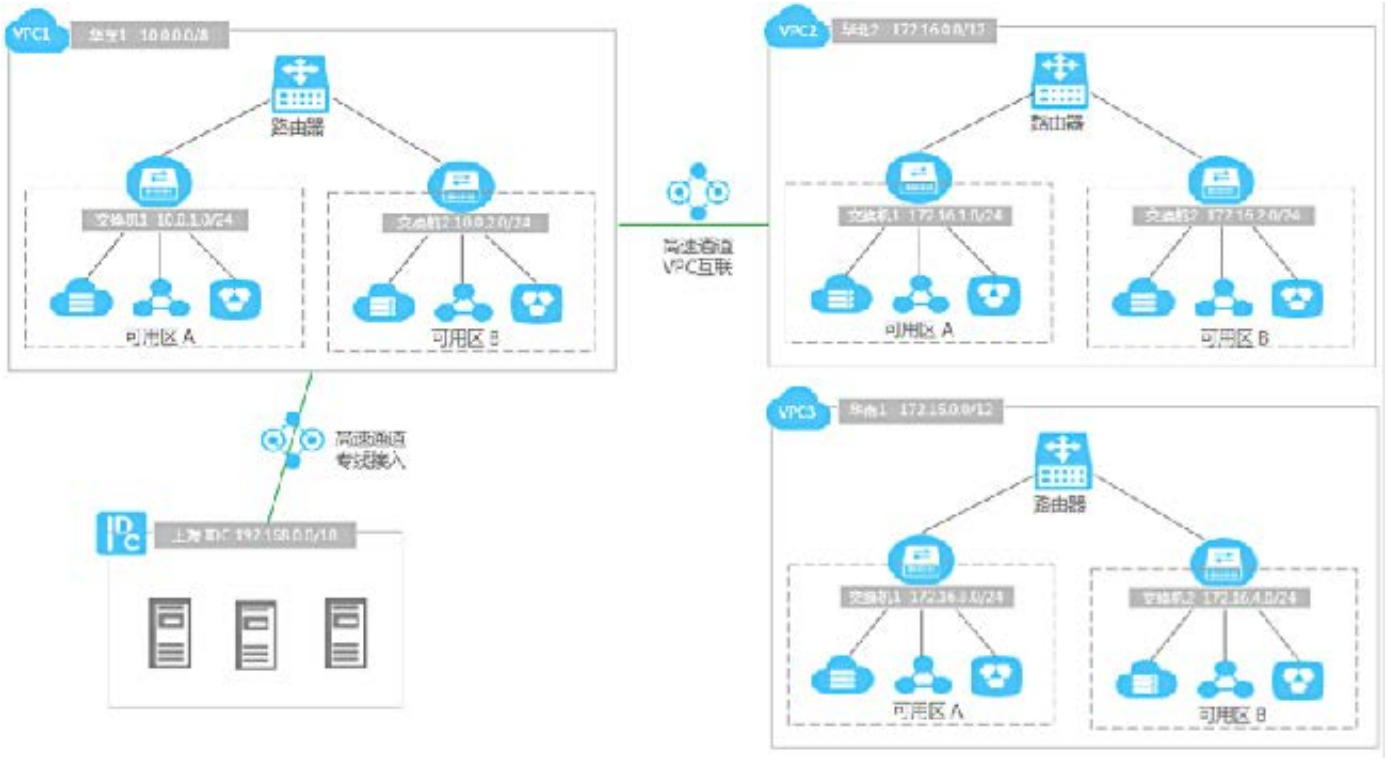


图3-2 多地域VPC互通架构示意图

VPC互通要求VPC的网段不能冲突或者相同。因此，在这个例子中VPC1和VPC2使用了不同的网段，而VPC3暂时没有和其他VPC互通的需求，所以VPC3的网段和VPC2的网段相同。但考虑到将来VPC2和VPC3之间有私网互通的需求，所以两个VPC中的交换机的网段都不相同。VPC互通要求地址不能冲突，确切的说是交换机的网段不能一样，但VPC的网段可以一样。

在需要多VPC的情况下，建议遵循如下网段规划原则：

- 尽可能做到不同VPC的网段不同，不同VPC可以使用标准网段的子网来增加VPC可用的网段数；
- 如果不能做到不同VPC的网段不同，则尽量保证不同VPC的交换机网段不同；
- 如果也不能做到交换机网段不同，则保证要通信的交换机网段不同。

基于阿里云产品的全链路评估

饮冰
阿里云技术专家

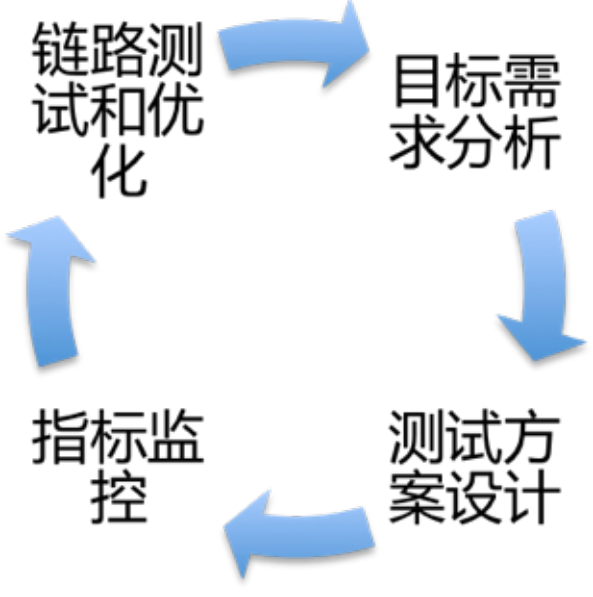
一、什么是全链路评估

在突发流量或者是新业务上线的场景，在未来某个特定的日期或者时间点有一轮大流量用户请求的时候，由于没有真实生产环境运行时候的历史信息做参考，就特别需要通过链路评估来对系统承载能力进行评估，从而保证系统的可靠稳定运行。

这里面说的链路评估，指的是从访问入口开始，对整条链路的所有潜在的瓶颈点，进行全方位的测量，通过改造链路结构和容量配比，达到提升整体链路性能和可靠性的目的。

这里面的链路评估是相对单系统调优来说的，单系统调优重点关注的是在“机器操作”的微观级别上做具体的软件调整用以挖掘硬件的潜能，单系统调优关注的是单系统的瓶颈，解决的是一个点的问题。而我们说的链路评估，希望解决的是一条线上的问题。

整体链路评估的工作通过对活动目标进行需求分析、设计测试方案、设计指标监控方案，来完成对整个系统的链路测试和优化工作。这四个环节是一个不断循环不断迭代的过程，每当活动目标发生变化，业务场景发生变化，指标的测量结果发生变化，都需要给整个环节带来再平衡。



现在“发红包”已经成为除夕之夜上一道必备菜品，每年的春节除夕，支付宝、qq、微博等都会联合众多行业，每个行业一个品牌，每个品牌一段固定的时间，进行红包派发的活动。社交平台借着派发红包提高了平台人气，各大知名品牌，通过社交平台完成向自

己的网站的引流，达到了推广自己产品的目的。在整个活动中，社交平台的网民们拼的是手速，而各大知名品牌为了可靠的推广自己的产品，则拼的是自家网站后台的技术架构。接下来，我们以一个真实案例，来分析一下，红包大促这种的高并发场景的促销活动如何进行链路评估。

二、红包业务背景

首先介绍一下活动背景，金融行业一个互联网公司，准备在春节除夕，利用与一个大型社交平台合作的机会，进行红包派发活动，并向自己的app引流，吸引更多的用户使用自家的app，在自家的app上完成购买产品和消费的操作。

活动期间，社交平台将会在固定的半个小时时段内，为该公司的app开启专属的红包派送通道，用户可在平台的首页通过刷一刷参与活动。到时候，将有上亿用户，在这半个小时的时间内，通过在平台的首页上不停的刷一刷，进入到我们客户自己的红包页面，参与到3000多万个现金或者是代金券的秒杀活动中。

按照客户的活动预期，现有250万的用户规模，希望借着这次红包活动希望能够吸引到500万以上的新注册用户，也即是说，在抢完红包之后，预计会有250多万的既有用户和500多万的新注册用户，涌入客户的手机app，进行红包的查询和消费操作。红包的业务规模如下图所示。

业务量	用户规模	峰值场景
• 活动时间：30分钟 • 红包总数：3000万+ • 抢红包用户：某社交平台上亿用户	• 当前app用户：250万+ • 新注册app用户：500万+	• 抢红包 • 登陆、注册 • 消费红包

在如此的业务规模之下，我们面临两大挑战：

1.第一大挑战，3000多万红包，半个小时，一亿以上的用户来抢，而且是整点准时抢，到时候瞬间峰值会非常有考验；

2.第二大挑战是，该红包是线上既可以消费，所以抢完红包之后，会紧接着带来线上用户注册和线上消费的峰值，抢红包、登陆注册、线上交易三种高并发写入的场景的峰值重叠在了一起，这更是增加了大促活动线上流量的复杂度；

三、目标需求分析

了解完活动背景，我们首先进行目标需求分析，目标需求分析包括三部分工作：首先对业务架构进行梳理，确认整套业务架构都由哪些场景组成，各场景的操作流程是什么样的，哪些是入口场景，哪些是后续的场景，场景之间的交易路径和关联关系是什么？其次，我们需要对活动目标进行分析，从而预估出每个场景具体的业务峰值。第三，输出整个系统业务模型，包括所有的场景，及对应的目标峰值。

四、业务架构梳理

通过和客户的技术人员的调研，我们了解到整体业务架构，包括入口层、服务层、资源层和外部接口层。

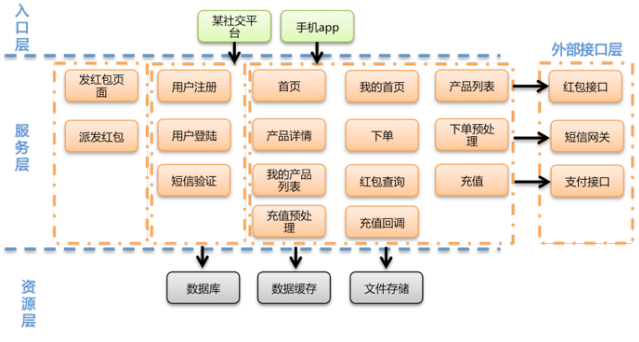
入口层指的是发起终端或者调用渠道，该系统的用户的来源是社交平台进行引流，通过手机浏览器在红包系统上抢完红包之后，再下载手机app进行注册登陆并进行一系列消费活动。

服务层提供了具体的应用模块的实现，登陆app之后查询红包列表，然后会选择查询产品列表，查询产品信息，充值，消费红包等等。

资源层则用到了阿里云的rds数据库、redis数据库、oss存储等技术组件。

外部接口包括支付接口、短信网关接口以及红包接口。

将入口层、服务层、资源层、接口层串起来，得出了完整的业务处理流程，从而描绘出了整个业务架构。业务架构梳理图如下。



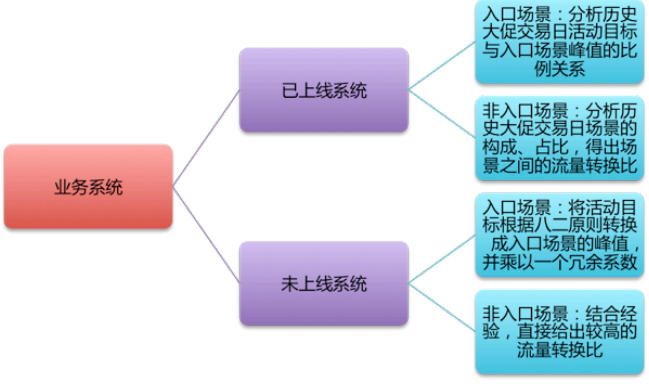
五、业务峰值预估方法

梳理完业务系统，我们需要对整个业务系统的每个场景的峰值进行评估。

对于已经上线的系统，我们需要通过该系统的历史大促峰值流量信息对系统的业务规模进行定量的分析，得出大促活动都有哪些场景构成，各场景的占比关系如何，活动目标与入口业务峰值的比例如何。对于入口业务，也就是说在业务流程上，调用完业务，才可以继续后端的业务流程。需要根据历史日志信息计算出活动目标与业务峰值的比例关系，比如说将具体的活动目标，一般包括在线用户数，新注册用户数，多长时间，达成多大的业务目标，转换成具体的入口业务的峰值目标。而对于非入口场景，在业务峰值的估算方式上，具体的计

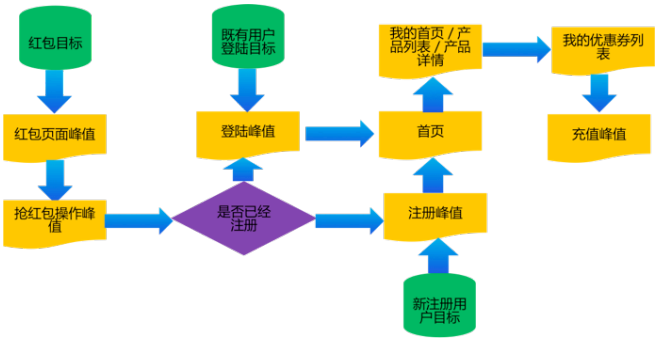
算方式可以是考察在历史日志信息中，各个场景占总交量的百分比，进而换算统计这些场景之间的比例关系是怎样的。依据各场景之间的比例转换关系，计算本次大促活动的各场景的峰值。

对于没有具体的历史业务信息作为参考的未上线系统，我们同样需要面对两种情况，入口场景和非入口场景。对于入口场景，我们通用的方式是采用八二原则将活动目标转换成业务峰值。八二原则的定义就是：20%的投入支撑80%的产出。在我们的峰值评估场景里面就是20%的时间支撑了80%的业务量。对于非入口业务来说，从业务入口进来的流量，在业务不频繁变更的情况下，一定会以一定的比例落到后端的各业务逻辑上。以业务入口峰值作为基准，按照经验值，给出业务入口与后续各业务的比例关系，推算出各业务行为的峰值流量。



六、业务峰值预估

结合本次具体抢红包业务场景，我们来看下各场景的峰值应该怎么样来预估。在半个小时的活动时间内，我们有三个活动目标，3000万的红包个数、250万的老用户登陆、500万的新增注册用户。对于本次需要评估的系统的入口业务，抢红包峰值、注册登陆峰值活动都没有历史信息作为依据，所以采用八二原则，根据活动目标预估峰值tps。对于后端链路的非入口业务，参照历史交易信息的比例进行流量转换。考虑到红包场景的特殊性，将某些业务之间的转换率，根据经验，提升到了1:1。



七、建立业务模型

完成了业务架构的分析，得出了活动涉及到的所有场景；完成了对与活动目标和历史交易信息的分析，我们得出了各场景的业务峰值。于是我们就可以完成业务模型的建立。如表所示，我们看到一共有3个入口业务和10个非入口业务，根据八二原则对3个入口业务的峰值进行了预估，根据历史交易的比例信息，对非入口业务进行了流量的转换。最极端情况下，各种业务峰值重叠在一起，最高峰值tps会达到34.36万。

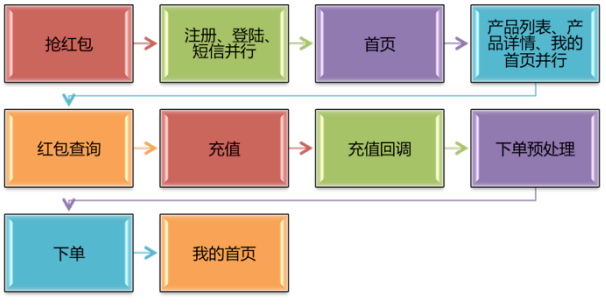
序号	场景类型	场景名称	活动目标或者对应的前端场景	峰值 tps(万)
1	入口场景	抢红包	30 分钟、3000 万+个红包	10.7
2	入口场景	老用户登录	30 分钟、老会员 250 万+	0.8
3	入口场景	新用户注册	30 分钟、新注册会员 500 万+	1.7
4	非入口场景	短信	场景 3	1.7
5	非入口场景	首页	场景 1*5/6	8.9
6	非入口场景	我的首页	场景 2+场景 3	2.5
7	非入口场景	产品列表	场景 2+场景 3	2.5
8	非入口场景	产品详情	场景 7*2/3	1.7
9	非入口场景	红包查询	场景 2+场景 3	2.5
10	非入口场景	充值	场景 8*1/5	0.34
11	非入口场景	充值回调	场景 10	0.34
12	非入口场景	下单预处理	场景 10	0.34
13	非入口场景	下单	场景 10	0.34

八、测试方案设计

完成了业务模型的建立，接下来我们就需要根据业务模型来完成测试方案的设计这部分我们要做两件事情，第一，选取需要测试的业务场景，第二，选择测试模型。

九、选取业务场景

考虑到尽量模仿真实的用户逻辑，尽量覆盖到所有的业务峰值彼此叠加的场景，尽量能够模拟用户抢红包的整条业务链路，我们选取了服务层所有业务场景。这张图就是所有业务场景混合在一起的整条业务链路。我们就会对这张图的所有场景进行脚本模拟，但是我们怎么样对这些场景进行测试呢？



十、选择测试模型

考虑到实际的业务场景目标需求，我们选取了以下四种测试模型。单场景基准测试模型、单场景负载测试模型、混合场景的负载测试模型、系统超时流控测试模型。

•单场景基准测试模型：单支交易在系统无压力情况下重复执行多次，检查该交易是否存在性能缺陷，并获取每个交易的基准性能，同时为容量测试提供参考数据。

•单场景负载测试模型：单交易负载测试模型通过逐步增加并发量进行负载测试，获取单交易业务处理性能峰值，验证单场景是否存在并发性问题。

•混合场景负载测试模型：是按照业务模型的约定在逐渐增加并发情况下进行测试，通过测试获取模拟实际生产环境中被测系统性能表现数据。

•系统超时流控测试模型：根据对系统中设定流控策略的场景的超时流控功能点有效性、可靠性、稳定性等验证在模拟实际生产系统情况下，验证多个超时流控机制并存情况下的正确性。

单场景基准测试模型	• 验证无压力的响应时间，建立性能基线
单场景负载测试模型	• 验证系统能够支撑的单场景最高峰值
混合场景负载测试模型	• 按照业务模型，获取多场景下的最高业务峰值
超时流控测试模型	• 对系统中各个超时流控功能点的有效性、可靠性进行验证

十一、指标监控

制定好了测试方案，接下来要完成指标监控工作，这样才能在进行压力测试实施的时候，能够对整条链路有一个数据评估。

十二、数据链路梳理

各个业务场景的流量，最终都要量化到数据链路的容量上。每条数据链路通常包括访问渠道、阿里云的网络产品、应用集群、阿里云的数据库产品、第三方接口等，通常，依赖于链路跟踪功能的系统能够最方便的获取到关联链路的上下文依赖信息，这里由于在这里我们是通过调用地址信息、业务日志的片段信息、连接配置信息来梳理整条数据调用链路。在这里我们选择一个链路较长的业务，抢红包的业务为例：通过手机浏览器发起操作，数据流量首先经过阿里云的ddos防护产品，然后经过负载均衡设备，到达ecs上的应用集群，对redis和rds的数据写入操作，通过统一的nat网关，完成对接口的调用。整条数据链路是一条同步链路，在对抢红包业务进行链路的能力测量的时候，其中的任何一个环节出现性能问题，都会对整个请求的响应时间造成影响。

十三、确认监控指标和监控方法

对于同一个阿里云产品组件，可能承担了几个场景的访问流量，对于数据链路上不同类型的技术组件来说，我们需要重点关注哪些相应的度量指标呢?

1.前端指标：这也是最能够反映出用户真实体验的指标，我们使用阿里云的pts产品，在压测的过程中能够最直观的关注到tps，响应时间，成功率等指标。如果前端指标符合预期，而后端的数据逻辑也确实跑通的话，我们认为我们的测试符合预期。

2.对于网络接入，网络调度类的技术组件，由于不涉及深入计算及io操作，主要是考量pps、cps、qps、带宽一类的网络技术指标。比如：高防ip、waf防火墙、slb、nat网关等网络产品都需要重点关注这类指标，在测试评估过程中，在前端指标出现衰减的情况下，首先需要考虑到网络指标是否存在瓶颈。由于业务访问量和这些网络技术指标是可以成正比关系的，如果一旦网络成为第一瓶颈点，我们可以通过网络资源的扩展，有效提高业务支撑能力。

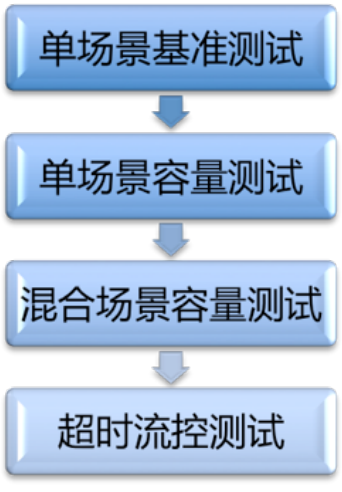
3.对于部署了应用中间件的ecs集群，我们关注三方面指标，操作系统指标、应用中间件指标和业务指标。应用集群的业务支撑能力取决于三个变量，如果存在后端依赖链路的性能问题，单纯的对应用集群进行规格扩容，对应用进行线程、内存或者计算优化，都无法提升应用集群的整体并发能力。在这里我们通过云监控来关注操作系统的指标，通过开源工具监（Probe、Jconsole）控应用中间件的指标，通过在日志里埋点，通过日志服务收集日志，并通过云监控订阅日志的方式，来完成业务指标的监控。同时，在应用集群上，接受请求的tps、处理时间、成功率，对后端调用的tps、响应时间，调用成功率，是反映数据链路健康程度的重要指标，只有这些指标是健康的，能够达到目标的，才能说明前端正常处理的tps请求是可信的。

4.数据库类技术组件，涉及深入复杂的计算及io操作，不同的请求会有不同的cpu消耗或者io消耗，所以cpu使用

率,qps、内存利用率等都可以通过云监控来关注。另外通过dms数据管理工具，可以对sql会话进行实时的监控。

十四、链路测试和优化

完成了测试方案的制订，监控方案的制定，那么最后一步，就可以通过模拟业务压力，测量整条链路的支撑能力，得出链路的短板，通过容量配比和链路架构优化的手段来达支撑活动目的的最终目的。根据测试方案的设计，我们的链路测试工作包括如下四轮。单场景基准测试、单场景的容量测试、混合场景的容量测试、最后一轮，制定超时流控的方案，并进行验证。



1.单场景基准测试

单场景基准测试需要达成两个目的。

(1)保证我们选取的13个场景，在系统没有压力的情况下，能够把数据链路跑通，跑通的定义是什么？就是模拟的脚本跑完之后，在压测端能看到成功响应，在应用集群上能看到业务日志，在数据库上能看到写入的信息。

(2)我们要获取到每个基准测试的响应时间，根据基准的性能，我们需要让业务人员评估出一个在用户看来能够接受的响应体验。根据“并发用户数=峰值tps*响应时间”这个公式，我们能够了解到，在这个保守的响应体验的基础上，要想达到场景峰值，需要多少并发用户数。为接下来的单场景的容量测试，提供并发用户数的依据。

序号	业务类型	场景名称	活动目标或者对应的前端场景	峰值tps(万)	响应时间（s）	并发用户数(万)
1	入口场景	抢红包	30 分钟、3000 万+个红包	10.7	0.25	2.675
2	入口场景	老用户登录	30 分钟、老会员 250 万+	0.8	0.25	0.2
3	入口场景	新用户注册	30 分钟、新注册会员 500 万+	1.7	0.25	0.425
4	非入口场景	短信	场景 3	1.7	0.25	0.425
5	非入口场景	首页	场景 1*5/6	8.9	0.2	1.78
6	非入口场景	我的首页	场景 2+场景 3	2.5	0.2	0.5
7	非入口场景	产品列表	场景 2+场景 3	2.5	0.2	0.5
8	非入口场景	产品详情	场景 7*2/3	1.7	0.2	0.34
9	非入口场景	红包查询	场景 2+场景 3	2.5	0.2	0.5
10	非入口场景	充值	场景 8*1/5	0.34	0.4	0.136
11	非入口场景	充值回调	场景 10	0.34	0.4	0.136
12	非入口场景	下单预处理	场景 10	0.34	0.4	0.136
13	非入口场景	下单	场景 10	0.34	0.4	0.136

2.单场景容量测试

在链路测试的第二个阶段，我们需要对每个场景进行并发数递增的施压过程，直到tps达到也场景峰值的需要。

由于这一步是容量测试，也就是说链路上并发能力的问题开始暴露出来。

3.数据缓存热点问题

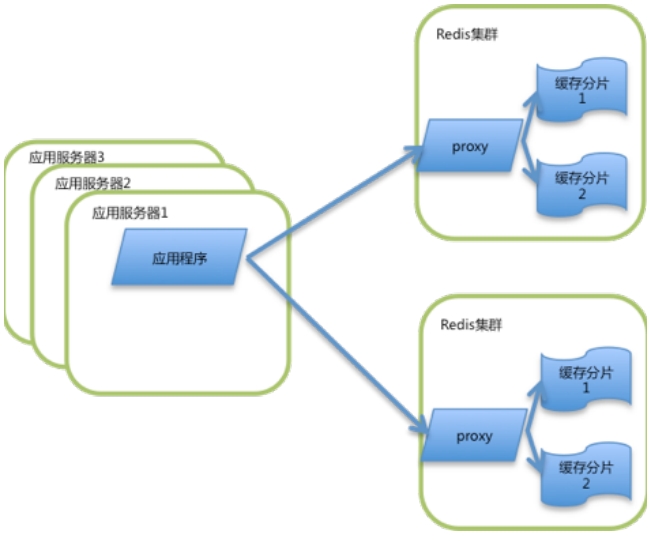
我们发现在抢红包的场景和新用户注册的场景出现了大量的调用超时。通过云监控查看Ecs所组成的应用集群的资源占用率不高，redis的调用qps也没有达到redis集群的极限，而从业务日志监控上来看对于redis的调用却出现了大量超时。

我们分析了一下具体的业务场景：redis在这套架构中主要是充当缓存和计数器使用。我们采用集群模式的redis提供服务，集群模式的redis的后端采用多分片的架构，扩展能力极强，而且可以破除单机redis单线程机制的性能瓶颈。对于redis多分片的集群，key比较分散的场景可以对资源进行充分的利用。而在发红包和注册的计数器应用上，两个计数器应用，分别是两个key，却承担了高并发场景所带来的高并发操作，而一个key的所有qps操作却分别只能hash到一个分片上，于是这个计数器的qps和其他场景将这个分片上的能力迅速撑爆。我们对客户提出了将热点key进行隔离的建议：

(1)注册峰值和红包峰值的热点key使用独立的redis实例；

(2)在redis集群上对热点key进行单独的集群分片路由；

在优化之后的在此容量测试中，得到了预期的场景峰值。



4.高并发写入延迟问题

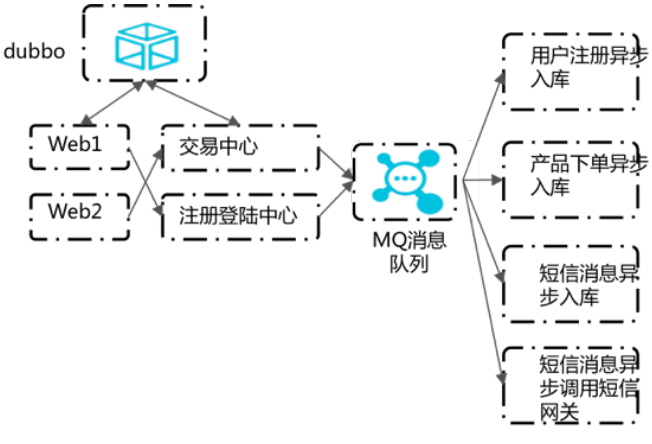
应用集群采用了基于dubbo的服务化设计，交易中心和注册登陆中心，分别负责产品下单信息的入库和注册信息的入库。

我们在进行单场景容量测试的时候，发现抢注册场景、产品下单信息入库的两个场景的响应时间都特别长。而这个场景对应的集群的业务日志监控发现，数据库的写入时间非常长。而且短信网关的也已经开始调用超时。对该场景进行分析：

(1)消费信息峰值、用户注册峰值、短信入库峰值对于数据库的写入造成了较大的压力；

(2)短信验证码的峰值超过了短信通道的峰值能力；

我们通过消息队列来控制对后端数据库和网关的调用频率，尽量降低对后端的高并发压力，同时也快速响应了前端用户。



5.混合场景容量测试

对各个场景分别进行了容量测试，我们从各个单场景的角度进行了链路的优化，那么这些单场景叠加在一起，能否分别达到各自场景的容量峰值呢?

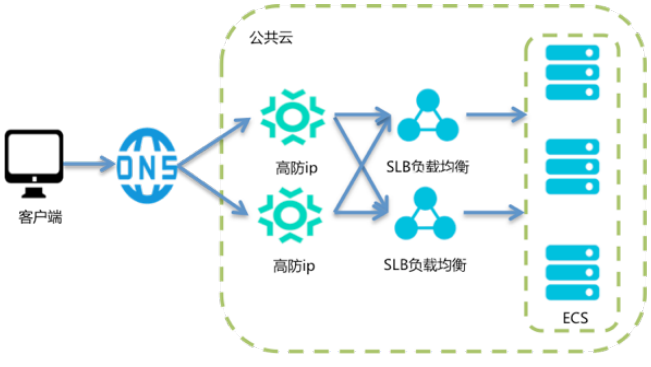
6.网络容量问题

我们按照实际的业务链路的先后顺序，依次对各场景进行并发压力增加的操作。当压力增长到一定值之后，我们发现，前端指标中，响应时间和tps指标衰减变得非常严重，而数据链路中，应用集群，数据库上面的指标几乎没有增长。

我们对场景进行分析，网络接入场景非常简单如图所示，采用通过高防ip来进行ddos和cc攻击的防护，通过slb来实现七层负载均衡的调度。

按照从前往后的顺序排查瓶颈点，首先，我们发现高防ip的回源带宽指标达到了瓶颈，首先对高防ip进行了多实例的扩展；接下来，我们发现所购买的slb实例已经达到了规格上限，带宽指标和pps指标已经达到了极限，然后我们又对slb实例进行了多实例的扩展；由于红包大促平台要求，各商家要统一使用https接入，对原来的http请求全部进行了https改造，我们也继续通过slb实例的扩展来抵消掉改造成https所带来的性能的衰减。

网络接入是所有业务场景的入口，在确保了网络接入流量不成为瓶颈之后，后续的容量测试过程中，请求流量顺利的从这里进入应用系统、缓存、消息队列、数据库中。



7.超时流控测试

由于第三方接口的容量不能随着我们调用方的峰值要求，进行扩展，甚至还会在高并发场景下进行限流，所以在链路测试的最后一个阶段，我们要设计好应对流控超时场景下的方案，并进行验证。

nat网关可以充当一个高可用的SNAT设备，同时能够使用一个固定的ip池，能够满足第三方接口对于白名单的需求，所以全站对外部接口的调用的需求，统一使用NAT网关产品。针对于每个虚拟机交换机下的ECS资源池，可以购买不同

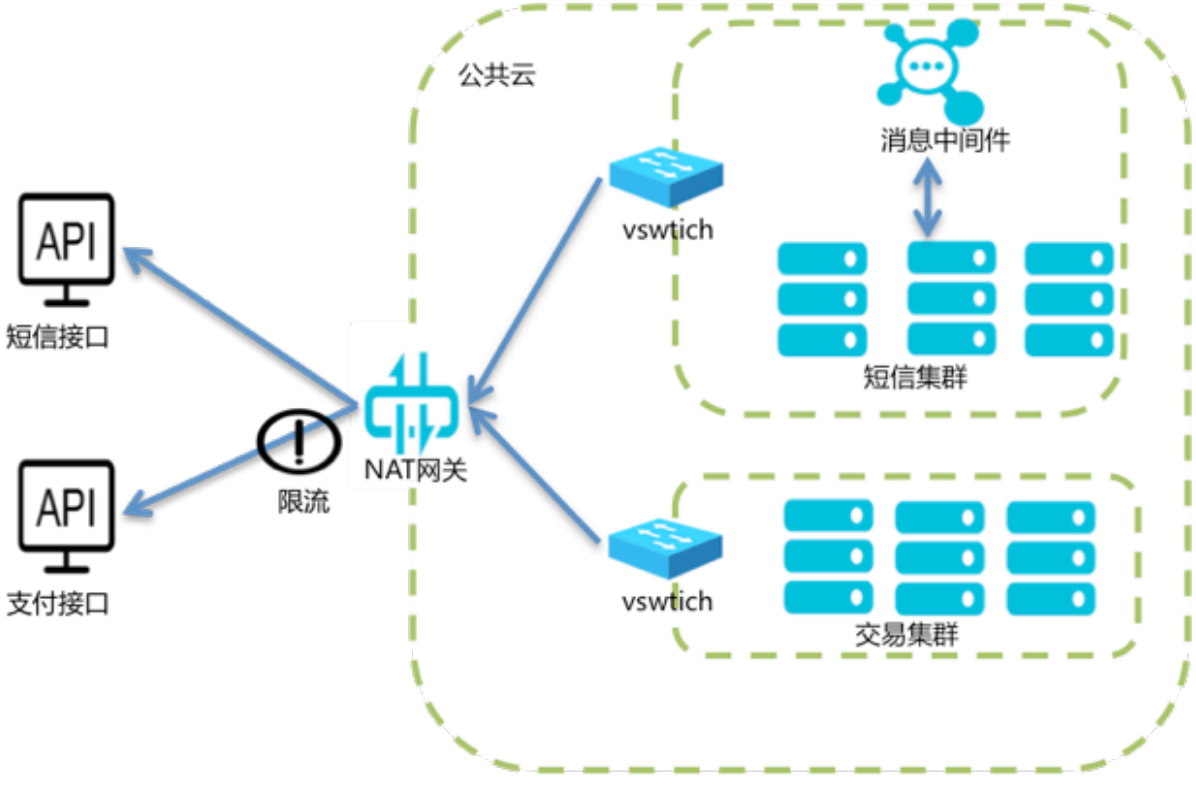
规格的的NAT网关的带宽来满足不同虚拟交换机下的应用集群的对外调用网络容量需求。

降级流控策略的设计和验证：

(1)在对于短信网关的调用上，虽然我们采用了异步化的方式，大大降低了对于短信网关的并发调用量，但是短信网关调用对应的业务场景是注册验证码，但注册验证码是有时间限制，对于时间存在一定的敏感度。于是我们建议客户对于异步化对外接口的调用，通过消息队列的积压度来进行降级流控策略，当队列积压度达到一定的数量，返回运营页面，减缓一下用户的注册频率。

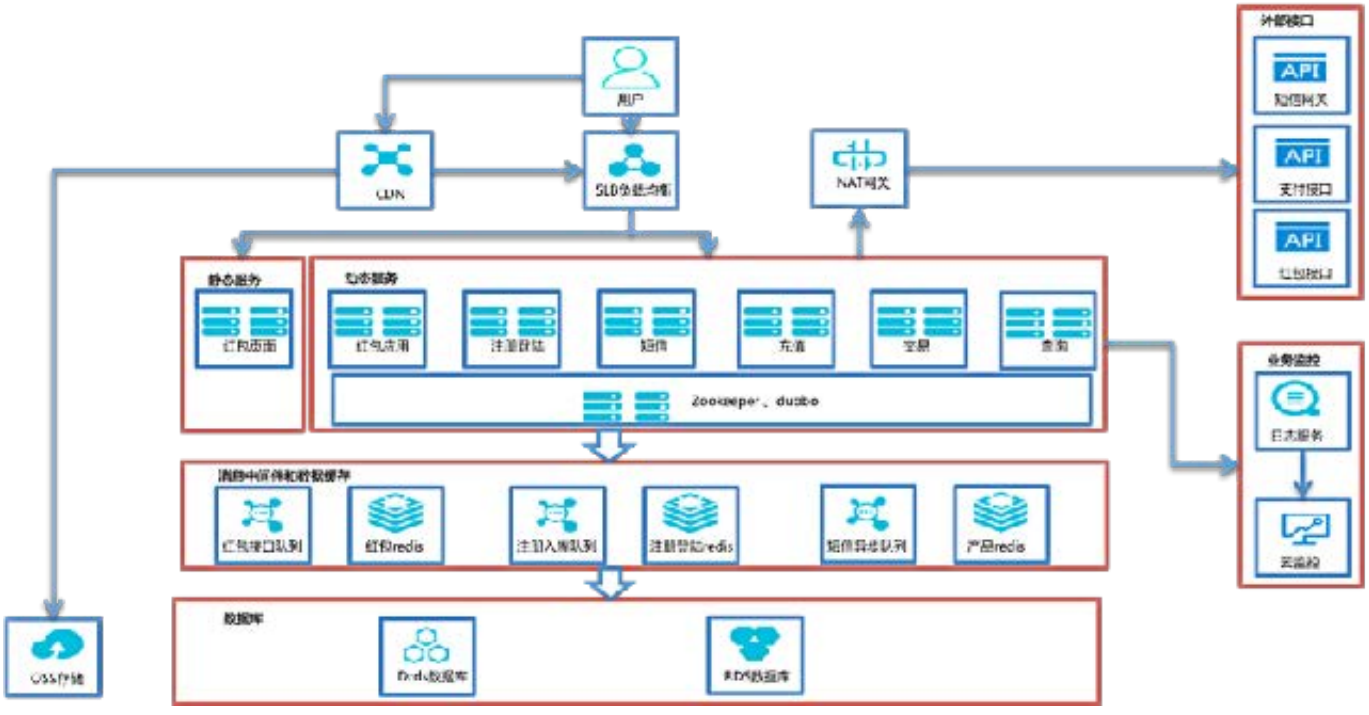
(2)对于支付接口的调用的调用上，为了避免线程堆积对应用集群带来的雪崩，我们采用的降级流控策略是降低超时时间和重试次数，为了在外部接口不可用的时候，快速通过友好的界面将失败返回。

在实际测试的时候，我们将所有场景保持在了一定的水位上，对支付和注册的场景进行了压力递增的测试。通过限制支付场景对外调用的NAT网关的流量，我们成功验证了在高压力下支付接口调用超时的流控策略；通过暂停掉注册场景的队列消费进程，我们也成功验证了在队列积压度达到一定比率的时候，注册场景的降级策略。



十五、最终架构设计

四轮测试下来，我们对每条链路上的架构进行了重新的评估。得出了最终的系统架构。从应用分层的横向的纬度，划分了四层结构。第一层是网络层，管理网络流量的进出，第二层是应用层，第三层通过redis实现了数据缓存，通过阿里云消息中间件实现了高并发调用或者写入的异步化处理，第四层是数据库和文件存储层。从系统纵向切分的角度，在网络层上，CDN提供静态服务，实现了对接入流量的负载均衡，NAT网关实现了对外部接口的统一调用。从应用垂直拆分的角度，通过使用dubbo框架，将不同的业务（或者数据库）切分到不同的服务器（或者数据库）之上，将动态静态业务也进行了拆分，拆分到了不同的应用集群上，进行垂直扩展，将不同的功能和服务分割开来，便于不同模块的分布式部署。在缓存层面，不同的业务，也可以使用独立的缓存实例，对热点业务进行隔离，在数据存储层面，数据库随着业务的拆分，同样进行纵向拆分。在分布式扩展上面：将分层和分割的应用和服务模块或者是阿里云的技术组件进行分布式扩展，以便获取更大的支撑能力。



总结

通过链路评估的工作，我们支持客户做了两个活动预案：一个是扩容预案，另外一个是降级预案。

1.在整点活动前2个小时完成了对活动系统的扩容准备工作，可以说把钱用到了刀刃上，这么短的成本周期，只有在公共云上才能实现。

2.在活动当晚，所使用的第三方支付平台恰好与别的商家活动发生了重叠，导致支付接口出现了一些波动，我们提前准备的快速失败的降级策略，避免了线程的堆积，而且返回给了用户一个比访问超时较好的体验；

活动期间参与红包活动的用户高达2亿+，成功派发数亿现金及卡券红包。经过评估的系统支撑了业内第一的日用户注册及充值交易转化率。客户非常满意。活动峰值持续了5分钟，活动高峰整整持续了半个小时，在此期间，整个的系统的业务指标表现的非常稳定，外部接口的不可靠也通过降级预案进行了完美的控制。链路评估的工作成果也体现在了结果上。

传统大型企业平滑上云典型架构实践

马柯
阿里云技术专家

一、传统大型企业平滑上云典型架构实践

随着云计算技术的不断发展和普及,使得对网络建设、业务运营、系统运维等多个角度对传统IT系统建设产生了深远的影响。越来越多的企业选择将应用系统迁移部署到云平台上，利用云计算平台产品特性构建低成本、弹性、高性能、高可靠性、高安全、按需获取计算能力的IT业务系统。

传统架构注重于硬件上的高可用，而云平台通过分布式架构已经确保自身服务的高可用，并且集成了备份，监控，HA，审计等一系列基础运维服务，云平台采用直接就可用的服务方式提供，使用方随时购买随时就可用，无需考虑一系列繁琐的底层运维，云使用者可以更加专注于业务上的研发。那传统企业需要平滑上云的时候，主要关注那些方面呢?

混合云构建：

混合云构建是将企业本地数据中心资源与云资源的集成。对于大多数企业而言，为降低IT的成本和实现业务快速创新而采用云计算，在混合架构中是必然的选择。迁移老的应用和系统上云是有一定的时间和成本消耗，因此，选择一家能够帮助企业实施全面混合战略的云计算厂商，这对简化企业IT运营以及更轻松地实现业务目标至关重要。

安全性：

企业对云的安全性要求优先级也是最高的，基于阿里云可以通过不同的云架构满足不同级别的安全要求，利用云计算的优势只需要给使用的服务付费，在保护云上资产安全的同时降低为安全消耗的成本。

业务连续性：

阿里云数据中心遍布全球14个区域，通过架构的设计可以使用跨可用区、跨区域的方式部署实现同城异地应用和数据级的灾备方案，提供企业系统的业务连续性。

合规性：

目前企业也需要云能够提供企业级的管理、监控、审计、安全等云服务功能来构建满足一些特定行业、国家的合规性法规和审计标准，目前阿里云共云已经通过等保3级，金融云通过等保4级。根据监管部门明确的结论复用原则，阿里云上的租户系统通过等级保护测评时，物理安全、部分网络安全和安全管理的结论可以复用，阿里云可提供说明。

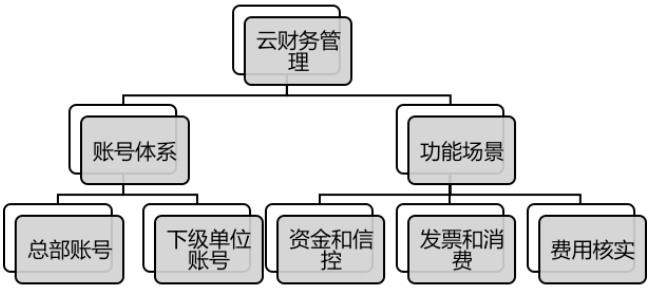
快速创新：

云其实是可以驱动业务创新的，除了用于新的业务模式以外，云还可以用于流程的创新，以提升效率，降低成本。顶级云计算战略家Joe Weinman曾经提到：云的未

来，不是一个成本中心，而是一个利润中心，它能创新很多新的业务模式，成为企业新的利润增长点。

1. 基于阿里云的企业级云架构实践

企业云财务管理



在阿里云的企业云财务管理中，提供总部及下级单位账号，为满足一些业务/合规/安全等需求而产生的多云账号的财务管理：

总部帐号：阿里云帐号，可以发起邀请下级单位帐号和对下级单位帐号做现金、信用额度，代金券的划拨，以及查看下级单位帐号的订单、消费记录，可以代下级单位开发票，可以限制下级单位提现和开发票的能力；

下级单位帐号：阿里云帐号，当确认接受总部帐号的邀请后，便成为该总部帐号的下级单位，自动继承总部帐号的财务优惠政策，可以获得总部帐号提供的现金和信用额度的划拨，但是会受到总部的消费记录查询、财务操作权限控制以及账户的控制，包括自身账户上的现金能够被总部划拨走。

企业云财务使用功能场景：

限制下级单位财务能力–支持提现和发票管理两种能力限制，限制发票管理能力后，下级单位将无法索取发票、设置发票抬头和设置发票寄送地址。

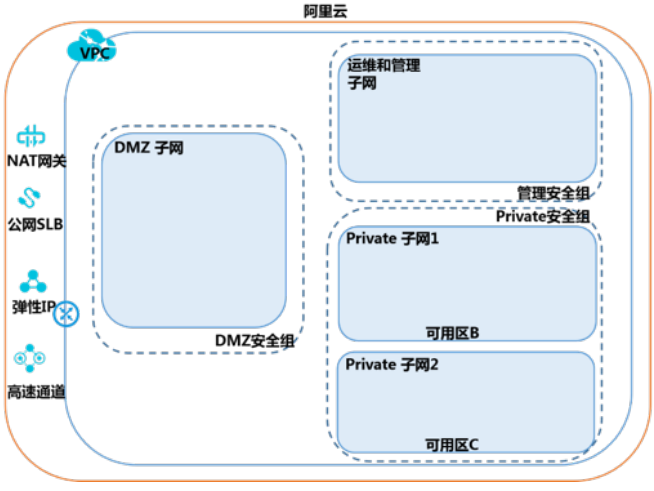
资金划拨–支持现金划拨，信用额度划拨，代金券分配管理。允许总部帐号把下级单位的资金划拨到总部帐号，也允许总部帐号把本账户的资金划拨到下级单位帐号。代金券仅支持分配，不支持回收。

消费管理–支持总部对下级单位的消费查询。

发票管理–支持总部代下级单位索取发票、设置发票寄送地址、设置发票抬头。

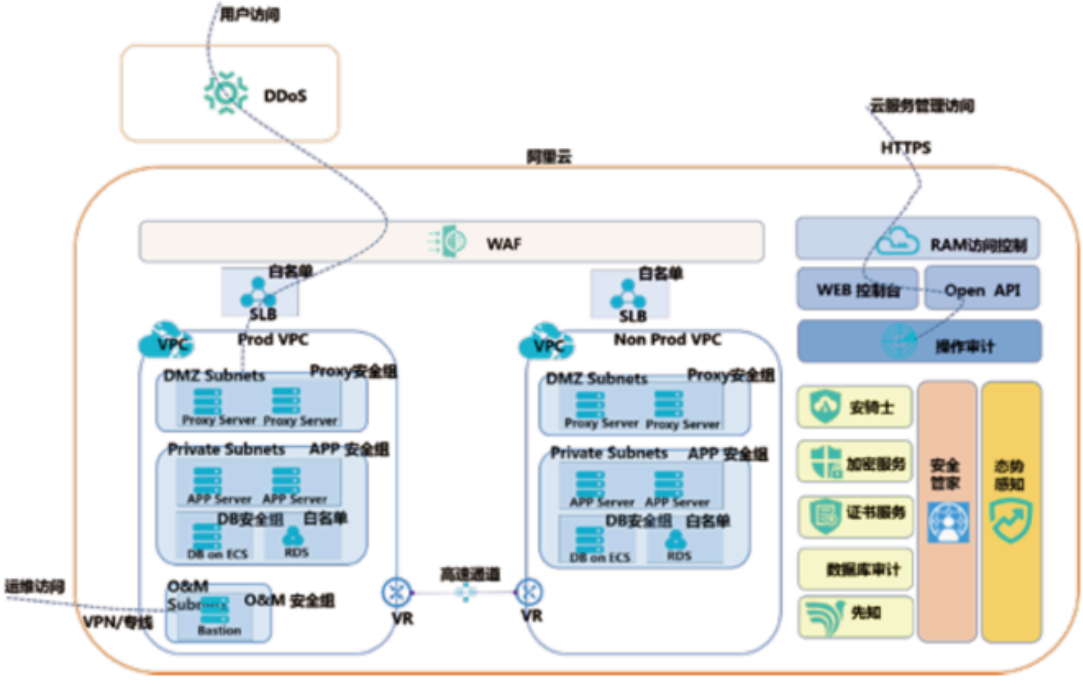
费用核算管理–费用核算管理功能支持将资源按照核算单元进行分组，并按照核算单元进行财务核算。

二、云上网络基础架构



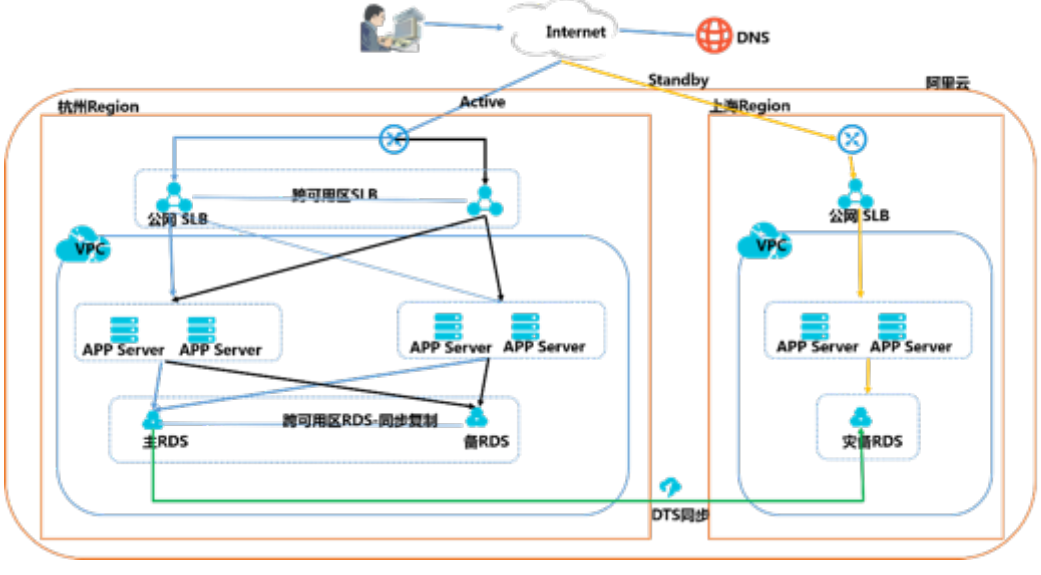
基于阿里云可以通过不同网络产品组合搭建网络基础架构，比如VPC虚拟专有网络可以实现自定义网络环境，通过虚机交换机将专有网络私有IP地址划分为一个或多个子网，根据需求将不同功能服务部署到不同的子网中，比如直接公网访问的服务部署DMZ，管理模块的服务部署运维和管理子网，应用服务和数据库服务部署到私有子网，另外可以根据业务需求配置虚拟路由器的路由规则，设置VPC网络流量转发策略和路径。VPC主要基于OverLay技术实现网络虚拟化，保障VPC之间网络逻辑隔离，并且通过安全组可以实现不同安全域的网络隔离，比如给比如每个子网从逻辑上放到一个安全组，可以实现子网之间的网络访问控制。公网SLB提供互联网接入和负载均衡能力，弹性IP/NAT网关也提供VPC对公网出入访问能力。通过物理专线+高速通道或VPN网关可以实现云上VPC与云下数据中心互联，轻松构建混合云架构。

三、云上安全基础架构



目前阿里云提供整套安全解决方案，包括了网络/主机/应用/数据等各个层面的安全产品，比如通过 DDoS 高防保障网络防护安全，通过安骑士提供服务器主机层的安全防御，CA证书服务和 WAF 提供应用层的的安全防护，数据库审计和加密服务提供数据层的安全保障。另外还有数据风控和绿网可以保障业务层的安全。另外除了各个层面的防御产品外，阿里云也有态势感知、安全管家、先知等云安全管理和测评服务，帮忙客户主动发现和管理安全威胁、加强安全防御体系。各云基础云产品也提供安全组和白名单的方式来进行访问控制、以及通过 RAM 访问控制管理和授权云服务访问权限，将云上资源访问控制统一管理。我们可以看到通常的3个访问路径，首先是互联网业务用户访问，先经过 DDoS 和 WAF 安全过滤后才能到达业务系统。运维访问通过物理专线从云下 IDC 或者通过VPN连接到云上堡垒机，通过堡垒机进行运维访问管理云上服务器和数据库，并记录运维操作和高风险操作拦截。云服务管理访问者需要通过访问控制进行认证后才能进行相关云资源操作，并且资源类变更操作会在操作审计中进行记录；

四、云上同城异地容灾架构



在云上帮助应用系统在设计和部署时，可以基于阿里云强大的基础设施，搭建同城异地灾备架构，实现同城高可用和异地容灾，保证应用和业务的恢复时间远远小于数据中心或物理设备的恢复时间，降低容灾系统建设成本和管理维护难度。阿里云 SLB 和 RDS 原生支持跨可用区部署，当发生故障后自动切换，不需要人为干预，而且没有额外的成本。图中杭州 Region 的里面的架构，SLB和RDS都是跨可用区实例，只需要将中间层 ECS 实例分别部署到不同的可用区，就可以很简答的实现应用系统跨机房的同城容灾；另外我们在上海 region 通过 DTS 数据传输服务搭建数据库灾备实例和应用服务 ECS 实例完成部署异地容灾环境，实现应用和数据级别的异地容灾，当杭州 Region 出现故障的时候，我们通过修改 DNS 云解析，将流量切换到上海灾备环境，提高业务连续性。

2. 客户案例分享

某传统企业“平滑上云”需求

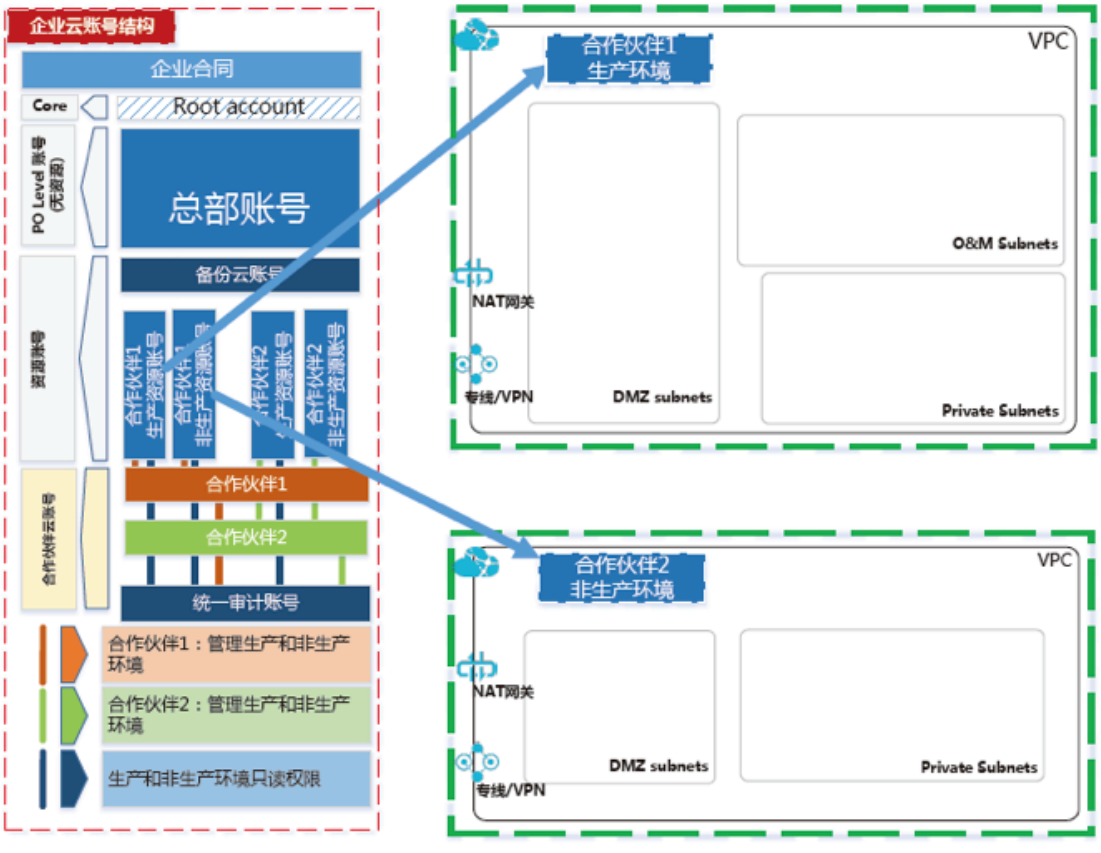
在这里我们先介绍一下这家企业上云的背景，这个客户是某世界500强外企，中国区刚好有个机房准备下线搬迁，希望能够快速迁移到云环境，利用云计算满足客户对计算和存储的需求，实现 IT 创新和成本降低。但在前期目标中，是希望快速将下线机房的应用迁移上云，并满足企业安全和合规要求。所以提出以下“平滑上云”需求：

(1)应用不改，在应用不改造的情况下从云下 IDC 环境平滑迁移到云上环境

(2)组织不变，从维护物理环境到维护云平台环境，组织结构基本不变

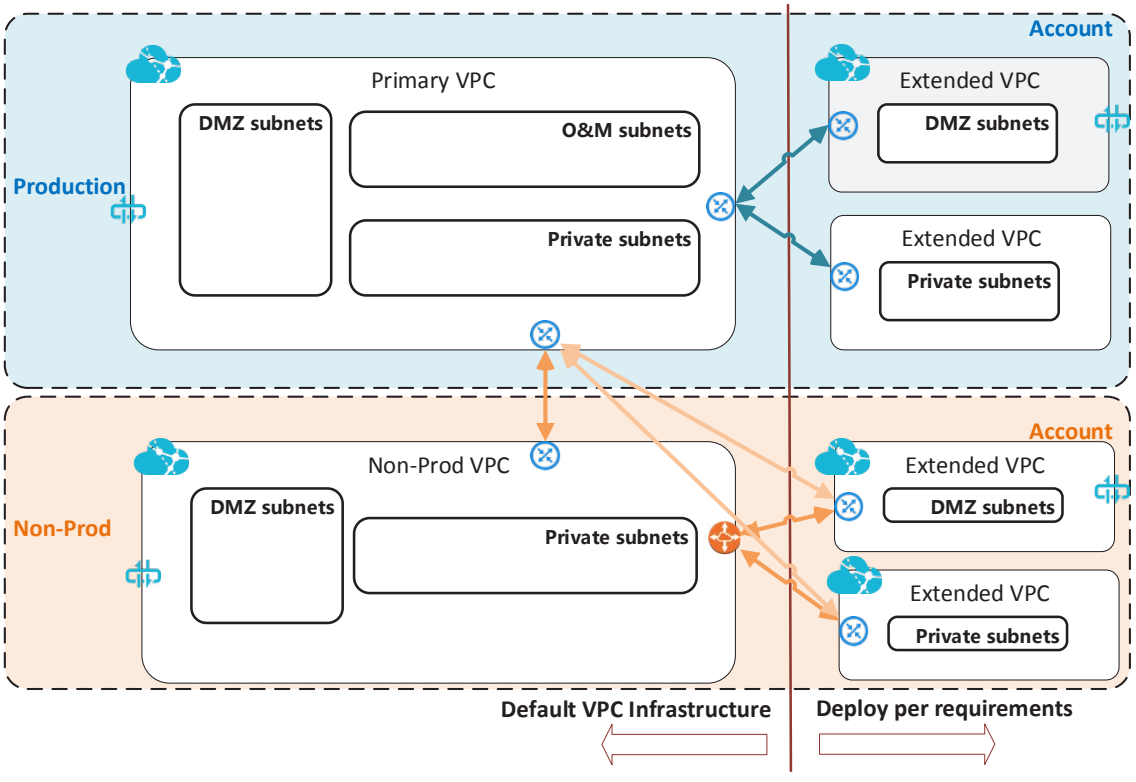
(3)业务体验一致，关键应用云化后，在业务体验、安全性、连续性需要保持或提高

五、云账号体系设计



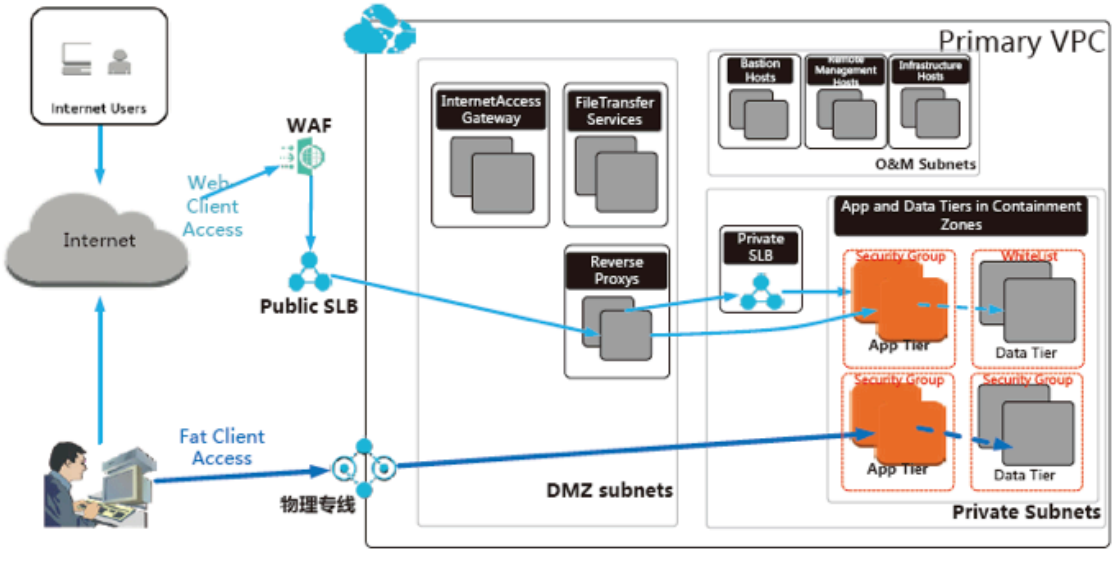
我们先看一下企业的云账号架构，通过结合企业管理需求和之前介绍企业云财务管理的功能特性，总部账号为部门级Root账号来实现整个部门级的云账号统一管理，下级单位账号分为资源账号、备份账号、统一审计账号。资源账号主要用于管理云资源，给IT管理服务供应商合作伙伴使用；备份账号主要用于备份数据存储使用账号；审计账号用于审计活动访问使用。资源账号通过阿里云 RAM 访问控制的角色扮演功能赋予备份账号备份数据功能，赋予审计账号只读权限来进行审计访问；

六、云上VPC设计架构

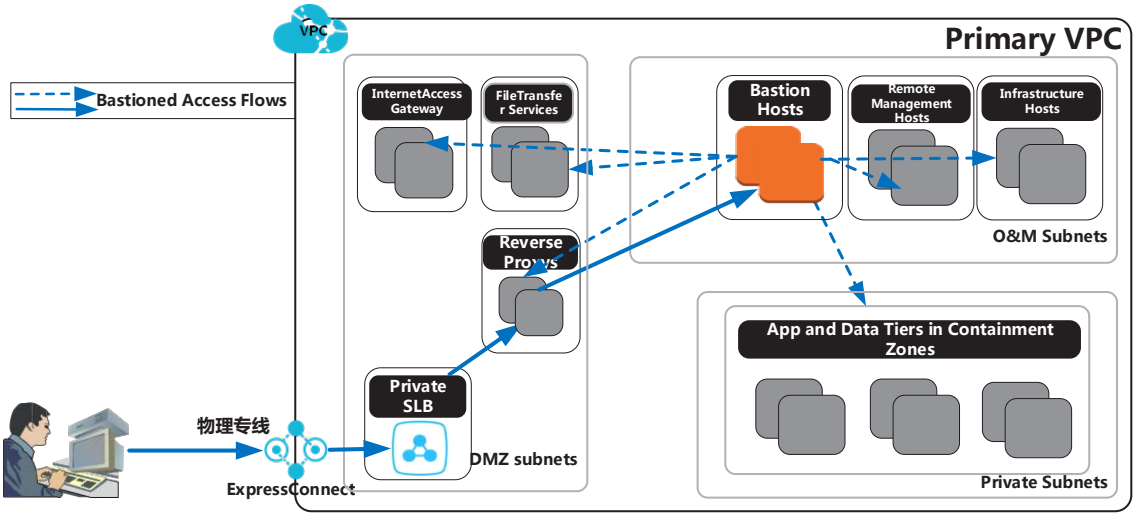


上图为这家企业云上VPC的架构设计，生产和非生产环境使用不同VPC，实现生产和测试环境网络隔离，生产环境部署生产业务系统资源，非生产系统部署测试、开发、验收环境等资源。生产和非生产VPC部署到不同的云账号中，另外规划后续有额外的需求可以不是额外的VPC，比如IP地址限制需求、额外的安全合规带来的隔离需求等。结合虚拟交换机和安全组、白名单实现子网划分和网络安全隔离，DMZ子网区域用于部署互联网暴露的资源，运维和管理子网部署运维和管理的资源，私有子网部署应用程序和数据服务资源。不同的VPC可以通过高速通道打通实现VPC互联，比如生产环境管理子网的资源需要访问管理非生产环境VPC中的资源。

七、用户业务访问、管理访问的“路径”

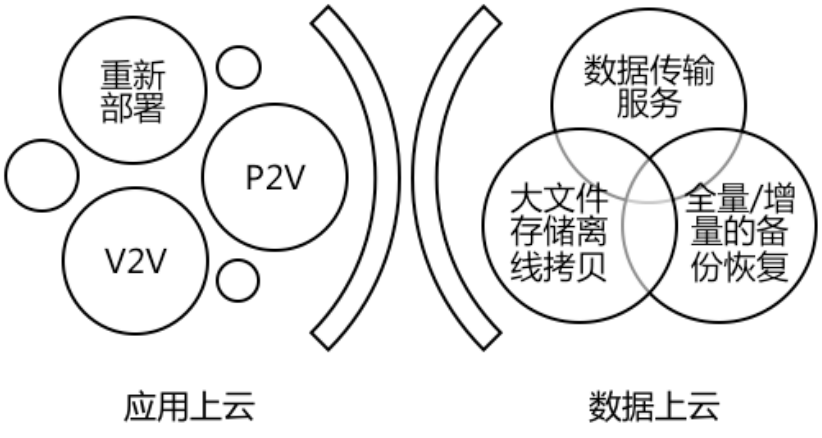


用户将互联网访问网关、文件交换传输服务、方向代理服务部署到DMZ子网，堡垒机、远程管理主机、基础设施管理主机都部署运维管理子网，应用程序和数据服务部署到私有子网，并且在私有子网中各应用和数据库服务之间通过安全组和白名单进行访问隔离，对于WEB类型业务应用全部通过互联网走WAF进行安全过滤访问，胖客户端（遗留的老应用，无WEB客户端）访问通过VPN到企业内网后通过走物理专线访问云上应用。



企业运维管理访问主要通过在企业内网中经过物理专线线路访问云上堡垒机后，通过堡垒机进行各项ECS或RDS进行运维操作，而对云资源的生命周期管理、部分运维操作直接通过阿里云WEB控制台或Open API进行访问管理。

八、应用和数据平滑上云



企业在基础架构设计和搭建完成后，需要将应用和数据迁移到云上环境中，线下IDC环境中大部分服务器部署在VMWare虚拟化平台，少量部署在物理服务器中；

应用层迁移主要考虑采用重新部署和P2V或V2V镜像迁移的方式，因为镜像迁移应用方式因限制原因（镜像文件大、不支持数据盘等）较多，对于应用部署安装复杂的应用才做镜像迁移，而部署简单的应用才采用手工重新部署方式进行迁移；

另外企业在线下IDC大部分应用的数据为结构化数据，数据库管理引擎主要为MySQL和SQLServer：对于MySQL类型结构化数据阿里云提供数据传输服务工具DTS可以进行全量和增量数据的迁移到RDS for MySQL，数据量较小的SQLServer数据库也可以采用DTS迁移；当前数据量过大,超过500G的SQLServer采用全量备份恢复和增量备份恢复的迁移到云上RDS for SQLServer；

对于大量文件的非结构化数据因为企业互联网带宽和存储量大的原因采用离线拷贝的方式进行迁移，节约带宽成本和迁移时间。

阿里云IPv6实施方案

周翰
阿里云技术专家

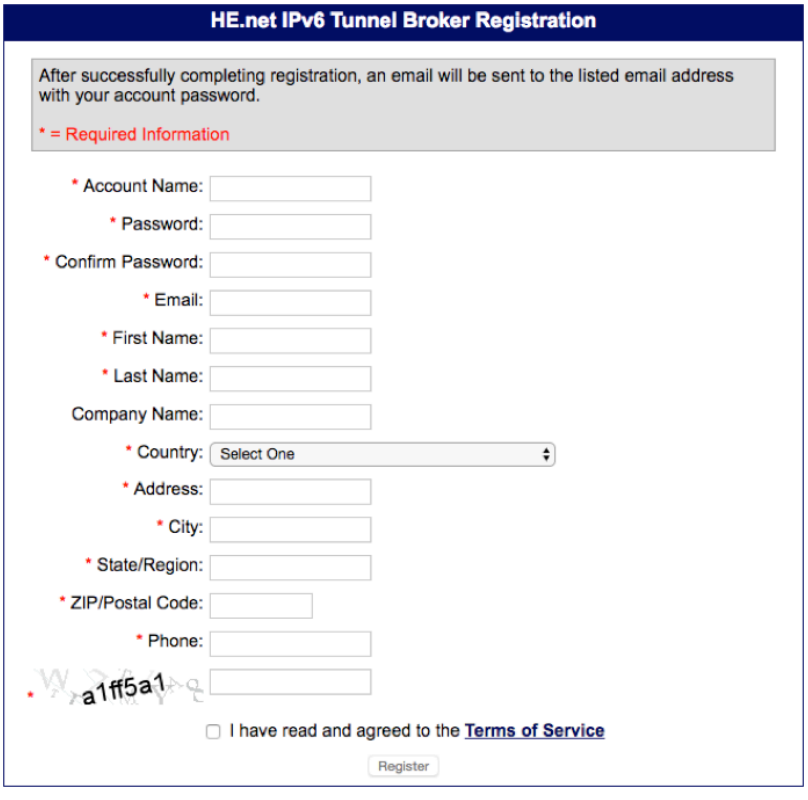
一、IPv6背景

2011年2月份，互联网地址分配机构(IANA)已将IPv4地址空间段的最后2个“/8”地址组分配出去。这标志着地区性注册机构RIR已没有空闲的可用IPv4地址空间。同时由于IPv6比IPv4有更大的地址空间、更高的路由性能、以及更好的安全性，所以从IPv4向IPv6过滤是大势所趋。基于此，自2016年6月1日起，苹果公司要求所有提交App Stored的应用必须支持IPv6-only环境，否则无法被审核通过。

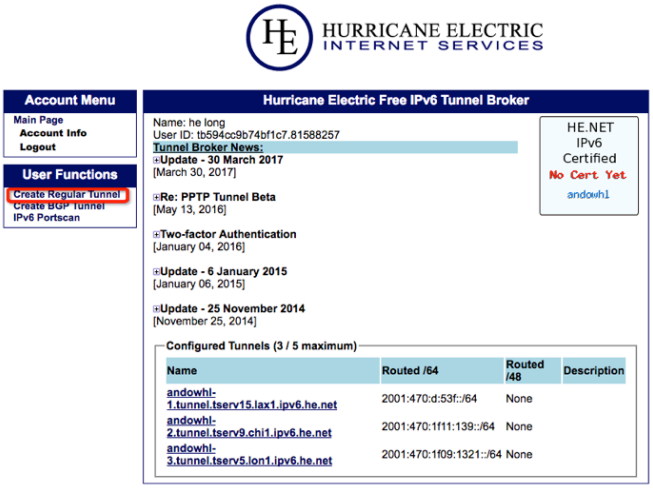
本文主要是为帮助苹果APP的国内厂商，基于阿里云ECS如何构建IPv6环境，顺利通过苹果官方基于IPv6-only环境对APP的审核。由于目前阿里云不支持分配IPv6地址，阿里云ECS服务器为能支持IPv6-only环境对APP的访问，需要配合在he.net申请和配置IPv6 tunnel来实现，同时通过阿里云提供的高速通道跨境实施方案来保证跨境网络访问质量的稳定性。

二、申请账号和IPv6 Tunnel

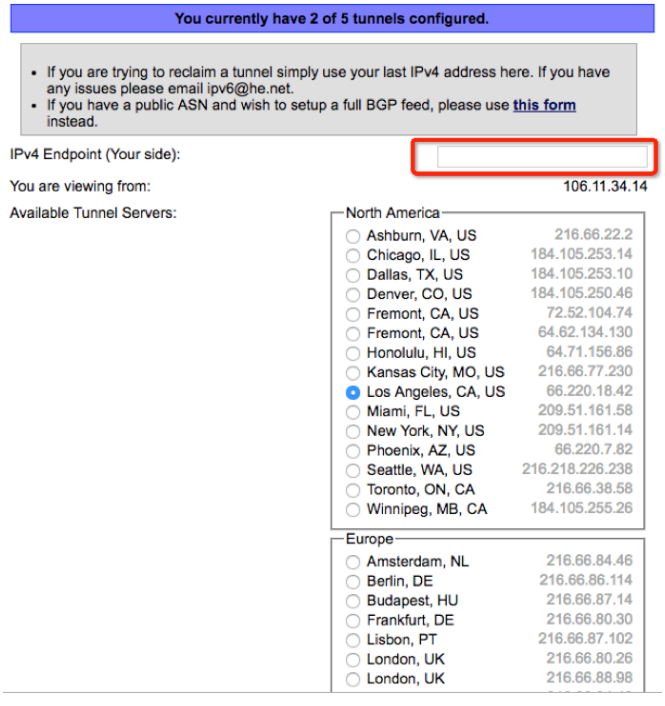
1.he.net网站：tunnelbroker.net，申请he.net账号



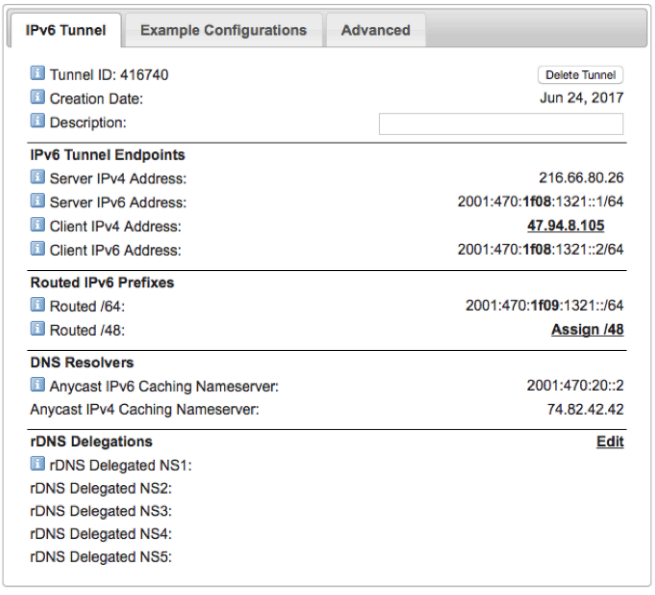
2.登录tunnelbroker.net后，左下角创建tunnel，Create Regular Tunnel



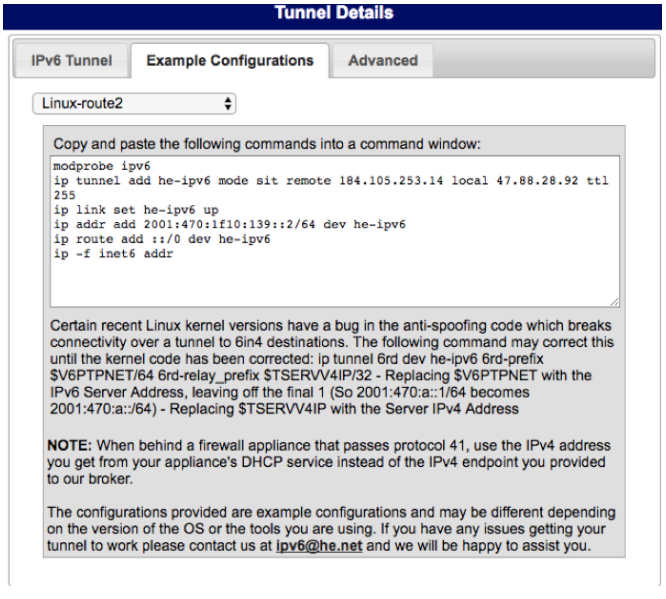
3.输入IPv4地址，IPv4 Endpoint(Your side)，并选择对应地域的IPv6地址，点击最下方CreateTunnel按钮完成IPv6地址分配和创建tunnel，并返回结果。



返回申请结果



4.点击Tunnel Details，选择Linux-route2



三、阿里云ECS服务器配置

1.ECS服务器配置IPv6地址和通道生成，复制1-4)内容执行在ECS服务器执行

modprobe ipv6

ip tunnel add he-ipv6 mode sit remote 184.105.253.14 local 47.88.28.92 ttl 255

ip link set he-ipv6 up

ip addr add 2001:470:1f10:139::2/64 dev he-ipv6

ip route add ::/0 dev he-ipv6

ip -f inet6 addr

2.经典网络ECS服务器CentOS7.0，开启IPv6

编辑 /etc/sysctl.conf 文件，将其中三条禁用IPv6的设置更改为：

net.ipv6.conf.all.disable_ipv6 = 0

net.ipv6.conf.default.disable_ipv6 = 0

net.ipv6.conf.lo.disable_ipv6 = 0

执行sysctl –p 生效IPv6配置

3.CentOS7.0设置IPv6域名解析

编辑/etc/resolv.conf文件，增加如下2行内容：

nameserver 2001:4860:4860::8888

nameserver 2001:4860:4860::8844

4.CentOS7.0测试是否正常访问IPv6服务，可以返回正常记录，表示IP地址和Tunnel已经生效。

ping6 –c 5 ipv6.google.com

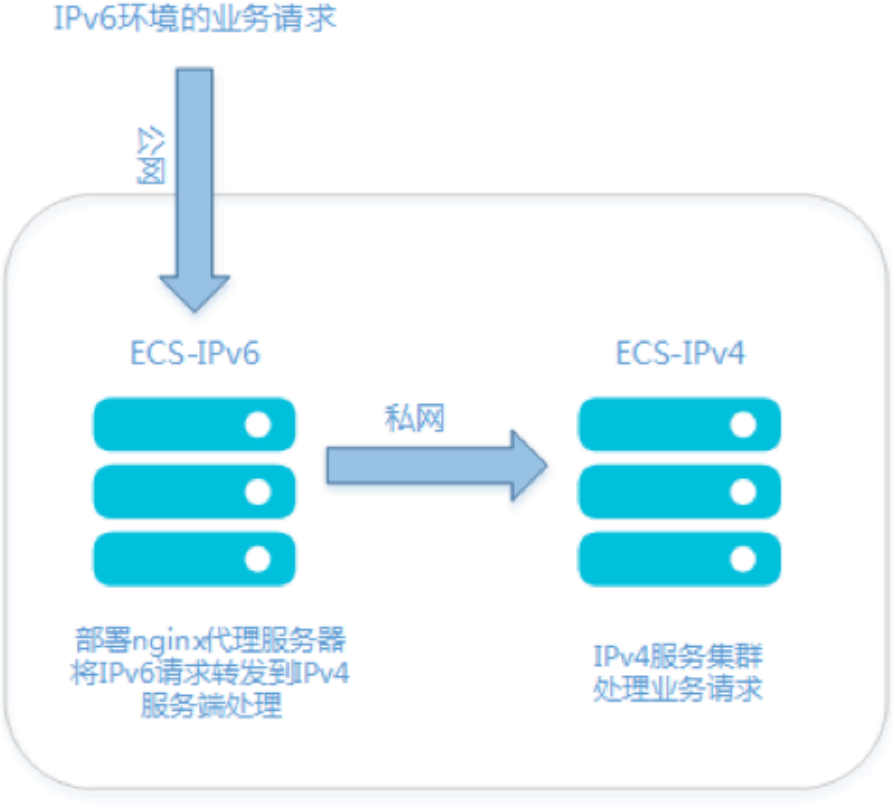
curl ipv6.google.com

四、Apple IPv6审核相关配置

以上操作完成后ECS即可以提供IPv6-only方式访问。对业务来说为满足APPLE对APP的审核需求，有两种部署方式来实现客户端运行在IPv6-only环境对业务服务方的访问。一是IPv6的ECS服务器申请在国内，国内到海外的访问直接通过公网访问，此方案网络的稳定性无法保障；二是IPv6的ECS服务器申请在美西或美东，美西或美东的ECS服务器通过阿里云跨境高速通道和国内服务端进行互联，此方案优势是跨境访问的网络稳定性可以保障，但会产生一定的跨境高速通道的网络带宽费用。

4.1 ECS公网跨境直访方案

4.1.1ECS公网跨境直访部署图



4.1.2跨境ECS公网直访实施步骤

(1)ECS-IPv6服务配置IPv6地址和he.net Tunnel

参考第1和2章节。

(2)ECS-IPv6安装nginx

```
yum install nginx
nginx -V #确认支持ipv6，包含--with-ipv6。
```

(3)ECS-IPv6配置nginx，编辑配置文件：/etc/nginx/nginx.conf，server部分。

```
server {
        listen [::]:80 ipv6only=on;
        server_name  c.miaopai.com _;
        location / {
                proxy_pass http://c.miaopai.com:80; #此部分为实际服务地址
            proxy_redirect     off;
                proxy_set_header   X-Real-IP  $remote_addr;
                proxy_set_header                X-Forwarded-For $proxy_add_x_forwarded_for;
                proxy_set_header Host c.miaopai.com;
                chunked_transfer_encoding off;
                client_max_body_size 50m;
  }
}
```

(4)ECS-IPv6启动nginx服务

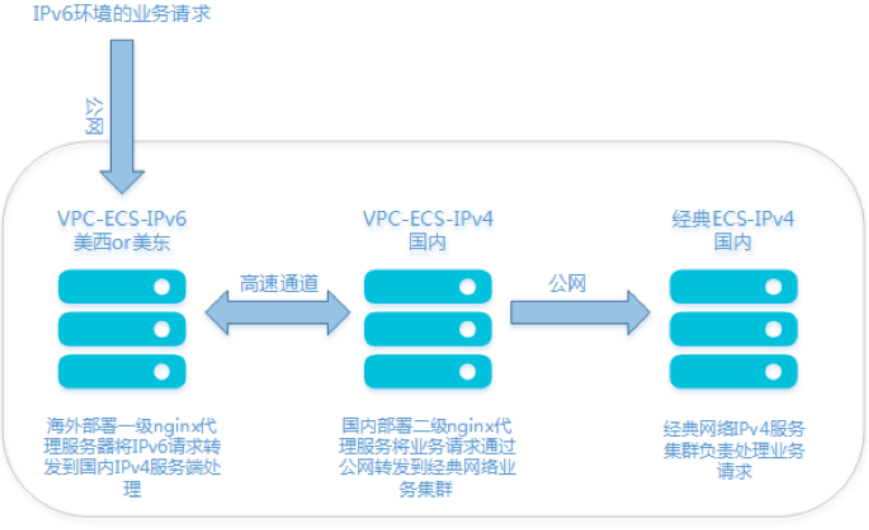
```
nginx -c /etc/nginx/nginx.conf
```

(5)测试代理服务是否正常

```
#测试ipv6 80端口是否启动
telnet [2001:470:1f10:139::2] 80
#测试业务调用是否正常
curl -6 -g “http://[2001:470:1f10:139::2]:80”
```

4.2 高速通道跨境访问方案

4.2.1高速通道跨境访问部署图



4.2.2高速通道跨境访问方案实施步骤

(1)VPC-ECS-IPv6服务配置IPv6地址和he.net Tunnel

参考第1和2章节。

(2)VPC-ECS-IPv6和VPC-ECS-IPv4安装nginx反向代理服务

参考3.1.2 nginx安装方法。

(3)VPC-ECS-IPv6和VPC-ECS-IPv4 修改nginx配置文件server部分。

VPC-ECS-IPv6：

```
server {
        listen [::]:80 ipv6only=on;
        server_name  用户自定义 _;
        location / {
            proxy_pass VPC-ECS-IPv4私网IP:80;
          proxy_redirect     off;
            proxy_set_header    X-Real-IP  $remote_addr;
            proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_set_header Host c.miaopai.com;
            chunked_transfer_encoding off;
            client_max_body_size 50m;
     }
    }
```

VPC-ECS-IPv4：

```
    server {
        listen [::]:80 ipv6only=on;
        server_name  自定义 _;
        location / {
            proxy_pass经典网络业务集群的外网域名:端口;
          proxy_redirect     off;
            proxy_set_header  X-Real-IP  $remote_addr;
            proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_set_header Host c.miaopai.com;
            chunked_transfer_encoding off;
            client_max_body_size 50m;
     }
    }
```

(4)访问测试，参考3.1.2-5)方法

五、阿里云IPv6 DNS域名配置

添加域名和IPv64A记录，请参考阿里云云解析官方操作文档：

https://help.aliyun.com/product/29697.html?spm=5176.doc29716.3.1.OXmsPl

应用迁云之镜像迁移

马柯
阿里云技术专家

一、应用迁云之镜像迁移概述

镜像迁移是指通过把源主机上的操作系统和应用程序及数据“镜像”到一个虚拟磁盘文件并上传到阿里云镜像中心，成为上传用户的自定义镜像后通过此镜像启动一个和源主机一模一样的ECS主机实例，来达到应用上云迁移的目的。

镜像迁移与手工重新部署迁移的技术对比

迁移技术类型		实现手段	优点	缺点
手工重新部署迁移		和物理主机部署方式一致	通用性强	效率低，操作复杂，需要较多人工干预
镜像迁移	冷迁移	通过工具直接镜像被迁移服务器主机，无法保障数据一致性	简单、效率高、成功率高	适用范围有限
	热迁移（阿里云不支持）	通过镜像迁移工具部署在被迁移服务主机或远程连接的方式迁移，迁移过程可以保持数据实时同步	简单、效率高、业务不中断	适用范围有限

目前到阿里云的镜像迁移主要需求场景来源有以下几种：

- 线下IDC机房的物理主机迁移到阿里云ECS主机实例
- 传统虚拟化平台的虚拟主机迁移到阿里云ECS主机实例
- 其他公有云的虚拟主机实例迁移到阿里云ECS主机实例
- 阿里云ECS主机实例在各Region、各VPC中间进行迁移

镜像迁移到阿里云根据迁移类型又可以分为：

- P2V迁移

P2V 指迁移物理服务器上的操作系统及其上的应用软件和数据到 阿里云平台管理的ECS服务器中。这种迁移方式，主要是使用各种工具软件，把物理服务器上的系统状态和数据“镜像”到一个虚拟磁盘文件中，阿里云启动的时候在虚拟磁盘文件中“注入”存储硬件与网卡驱动程序，使之能够启动并运行。

- V2V迁移

V2V是指从其他云平台或传统虚拟化平台的虚拟主机迁移到阿里云的ECS虚拟主机，比如Vmware迁移到阿里云，AWS迁移到阿里云等

二、镜像迁移可行性评估

目前无论是P2V还是V2V的方式迁移到阿里云还存在一些限制，我们在选择镜像

迁移的时候需要对被迁移的服务器主机和镜像迁移的工具进行评估：

• 被迁移服务器主机操作系统类型、文件系统类型、服务器已使用空间大小

• 镜像迁移工具支持导出的虚拟磁盘镜像文件格式

• 兼容性要求及限制

1.被迁移服务器主机操作系统支持类型

(1)Windows（32 和 64 位）

√ Microsoft Windows Server 2012 R2（标准版）

√ Microsoft Windows Server 2012（标准版、数据中心版）

√ Microsoft Windows Server 2008 R2（标准版、数据中心版、企业版）

√ Microsoft Windows Server 2008（标准版、数据中心版、企业版）

√ Microsoft Windows Server 2003 R2（标准版、数据中心版、企业版）

√ Win7 专业版，企业版

√ 不支持win xp ，windows 8，windows 10

(2)Linux（64 位)

√ Red Hat Enterprise Linux (RHEL) 5,6,7

√ CentOS 5,6,7

√ Ubuntu 10,12,13,14

√ Debian 6,7

√ OpenSUSE 13.1

√ SUSE Linux 10,11,12

√ CoreOS 681.2.0+

2.被迁移服务器主机的文件系统类

目前windows操作系统的文件系统类型支持NTFS，Linux操作系统的文件系统类型支持ext3，ext4

3.被迁移服务器磁盘及空间使用情况

如果被迁移的服务器来自传统IDC、传统虚拟化平台以及其他云平台，只支持系统盘迁移，不支持数据盘的迁移；并且系统盘大小不能超过500GB。

被迁移的服务器本身在阿里云上，只是需要迁移到不同的region或者不同VPC中，是可以支持系统盘和数据盘进行同时迁移，同样系统盘大小不能超过500GB。

4.兼容性要求及限制

(1)Windows限制

√ 导入的 Windows 镜像是提供 Windows 激活服务

√关闭防火墙。不关闭防火墙无法远程登录，需要放开3389端口

√ 关闭 UAC

(2)Linux 限制

√ 不支持开启 SELinux

√ 关闭防火墙 默认打开22端口

√ 关闭或删除Network Manager

√ 导入的 Red Hat Enterprise Linux (RHEL) 镜像必须使用 BYOL 许可。需要自己向厂商购买产品序列号和服务。

√ 不支持跟分区使用LVM

(3)其他限制

√ 不支持多个网络接口

√ 不支持 IPv6 地址

5.镜像迁移工具支持导出的虚拟磁盘镜像文件格式

阿里云支持上传的镜像文件格式为RAW和VHD；其他格式的镜像文件都不支持，需要通过镜像文件格式转换工具进行转换。

三、镜像迁移工具

目前在镜像迁移过程中主要使用镜像制作工具及镜像文件格式转换工具，镜像制作工具主要是把被迁移服务器主机的操作系统及应用程序和数据制作成镜像文件。因为不同的虚拟化平台的镜像文件或虚拟磁盘文件使用的格式不同，所以需要镜像格式转换工具对镜像文件格式进行转换来适配不同虚拟化平台。

当前镜像迁移到阿里云使用较多的工具有很多，比如Alip2v、Disk2VHD，DD等镜像文件制作工具以及Xen-Convert、StarWindConverter、qemu-img等镜像格式转换工具。它们都可以互相搭配使用，下面我来一一介绍。

1.Disk2VHD

Disk2VHD可用于将逻辑磁盘转换为 vhd 格式虚拟磁盘的实用工具。利用该工具我们可以轻松地将当前Windows系统中的C盘生成为一个 vhd 文件，然后上传到阿里云。

Disk2VHD能够运行在Windows XP SP2，Windows Server 2003 SP1或更高版本的Windows系统之上，并且支持 64位系统。

Disk2VHD下载地址：http://publicread081.oss-cn-hangzhou.aliyuncs.com/Disk2vhd.zip

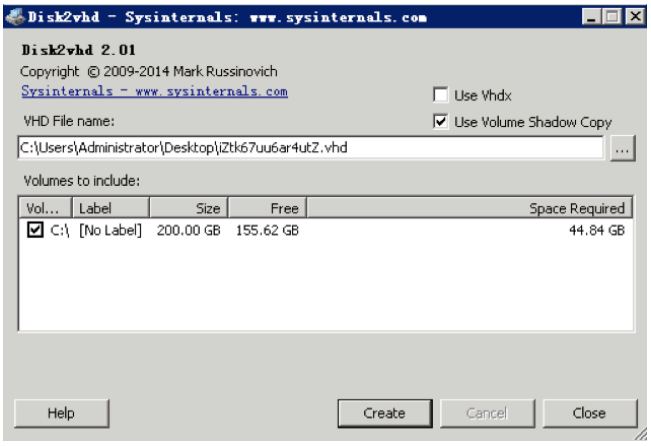


图4-19 disk2vhd软件使用界面

2.Linux DD命令工具

DD命令是Linux数据复制命令，通过DD可以将Linux跟分区所在系统磁盘镜像到一个RAW格式的文件。Linux DD的这个特性，我可以使用DD制作镜像文件。

3.XenConvert镜像格式转换工具

XenConvert是用于实现物理到虚拟（P2V）转换的工具，另外此工具提供了镜像格式转换的功能，其中包括VMDK格式转换为VHD格式。

下载地址：http://publicread081.oss-cn-hangzhou.aliyuncs.com/XenConvert_Install_x64.exe

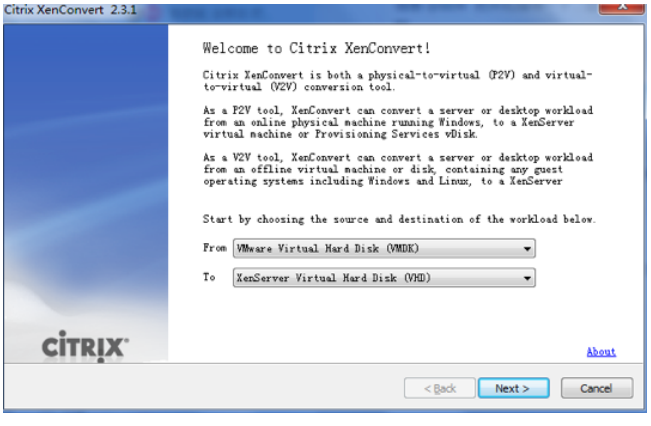


图4-20 XenConvert软件使用界面

4.StarWind Converter

StarWind Converter是一个格式转换软件，可以实现VMDK转换为VHD、或将VHD转换为VMDK，或转为StarWind的原生IMG格式。

下载地址：http://publicread081.oss-cn-hangzhou.aliyuncs.com/starwindconverter.exe

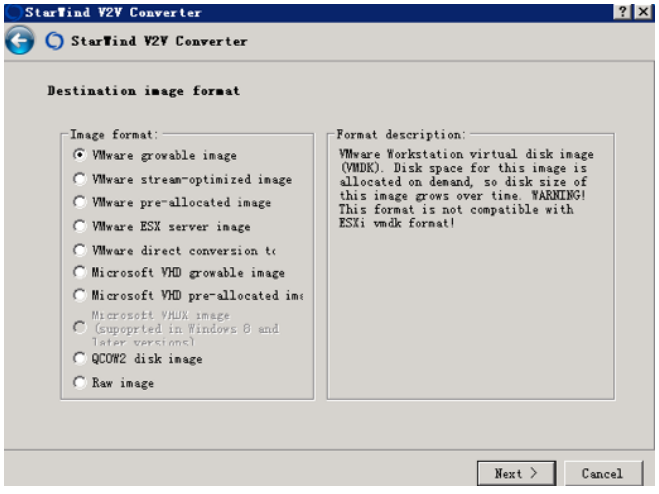


图4-21 StarWind Converter使用界面

5.qemu-img

qemu-img是QEMU的磁盘管理工具，也是QEMU/KVM使用过程中一个比较重要的工具。qemu-img命令工具的convert选项支持多种镜像文件的格式互相转换，主要包括Qcow、Qcow2、VHD、RAW、VMDK等。

比如VMDK转VHD命令样例：

qemu-img convert -f vmdk -O vpc vmware_img.vmdk aliyun_img.vhd

四、镜像迁移到阿里云实施流程和实践方法

我们参看如下实施流程图，镜像迁移分5个主要步骤进行。

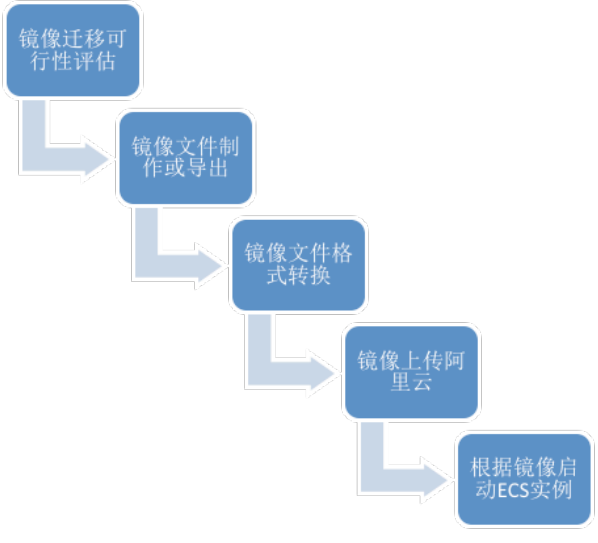


图4-22 镜像迁移实施流程图

1.镜像迁移可行性评估

当我们选择镜像迁移前需要对被迁移的服务器主机详细信

息进行调研，按照镜像迁移可行性评估小节中描述的要求及限制进行评估，评估是否可行、及是否需要采用镜像迁移的方式来进行迁移。

如果被迁移服务器主机数量规模大、并且大多都带系统盘、网络条件不好的情况建议不要使用镜像迁移方式；因为往往镜像文件都比较大，在此条件下进行镜像迁移反而会加大迁移的时间及人力成本。

如果被迁移服务器主机中应用配置比较复杂、无人维护、网络条件好我们建议使用镜像迁移的方式，虽然数据盘不支持镜像迁移，但是可以先把系统盘镜像迁移到阿里云后，数据盘数据可才采用文件同步的方式同步到阿里云的数据盘。

通常镜像迁移前需要一些准备工作：

√ 镜像文件存放公共目录准备

Windows类：

通过Alip2v或者DISK2VHD工具对Windows操作系统的系统盘进行镜像文件制作，我们可以把镜像文件存放地址输入公共目录地址，比如某台有台大容量空间的windows系统共享目录。

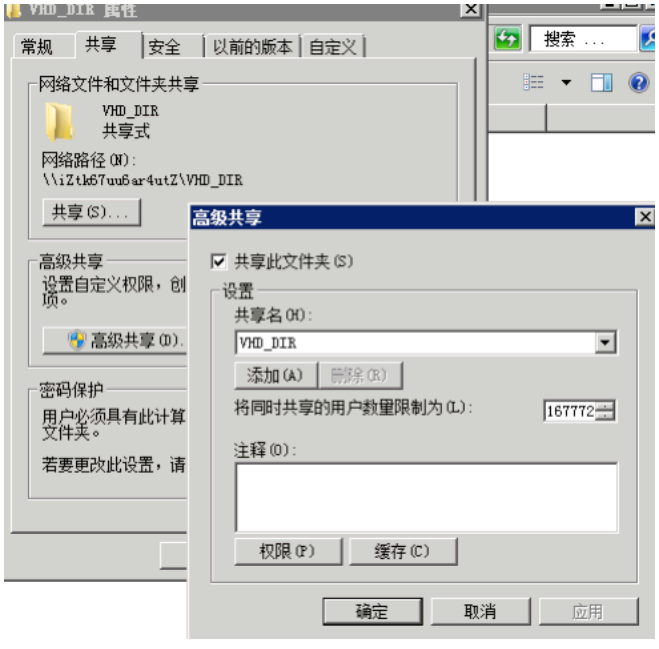


图4-23 设置windows共享目录

然后在Alip2v或者DISK2VHD的镜像文件保存地址中输入网络路径，比如\\iZtk67uu6ar4utZ\VHD_DIR，可以将镜像文件写入共享目录中进行统一管理。

Linux类：

通过Alip2v或者DD工具对Linux操作系统的系统盘进行镜像文件制作的时候可以把输出路径设置为一些挂载NFS的共享的目录，把镜像文件输出到统一的共享目录中（共享目录通常部署到镜像文件格式转换工具平台上）。

NFS环境搭建方法示例：

(1)环境示例

共享目录服务器端：CentOS6.5 192.168.0.10

被迁移服务器端：CentOS6.5 192.168.0.11

(2)共享目录服务器端安装配置

a.先用rpm-qa命令查看所需安装包（nfs-utils、rpc-bind）是否已经安装：

```
[root@local /]# rpm -qa | grep “rpcbind”
rpcbind-0.2.0-11.el6.x86_64
[root@local /]# rpm -qa | grep “nfs”
nfs-utils-1.2.3-39.el6.x86_64
nfs4-acl-tools-0.3.3-6.el6.x86_64
nfs-utils-lib-1.1.5-6.el6.x86_64
```

b.如查询结果如上，说明服务器自身已经安装了NFS，如果没有安装，则用yum命令来安装：

```
[root@local /]# yum -y install nfs-utils rpcbind
```

c.创建共享目录：

```
[root@local /]# mkdir /sharestore
```

d.NFS共享文件路径配置：

编辑/etc/exports添加下面一行，添加后保存退出。

```
[root@local /]# vi /etc/exports
/sharestore  *(rw,sync,no_root_squash)
```

e.启动NFS服务（先启动rpcbind，再启动nfs；如果服务器自身已经安装过NFS，那就用restart重启两个服务）：

```
[root@local /]# service rpcbind start
Starting rpcbind:                          [  OK  ]
[root@local /]# service nfs start
Starting NFS services:                     [  OK  ]
Starting NFS quotas:                       [  OK  ]
Starting NFS mountd:                       [  OK  ]
Stopping RPC idmapd:                       [  OK  ]
Starting RPC idmapd:                       [  OK  ]
Starting NFS daemon:                       [  OK  ]
[root@local /]#
```

f.设置NFS服务开机自启动：

```
[root@local /]# chkconfig rpcbind on
[root@local /]# chkconfig nfs on
```

(3)被迁移服务器端挂载配置

a.创建一个挂载点：

```
[root@localhost ~]# mkdir /mnt/store
```

b.挂载：

[root@localhost ~]# mount -t nfs 192.168.0.10:/sharestore /mnt/store

√ 镜像文件格式转换工具平台准备

镜像文件格式转换平台搭建主要是安装镜像格式转换工具，并且需要保证平台磁盘空间有较大容量来保存镜像文件，对镜像文件进行统一存储和管理。具体容量空间大小需根据迁移镜像规模而定。在格式转换平台上需要安装OSS工具，在镜像文件完成格式转换完成后上传到用户具体账号下阿里云OSS对象存储中。

Windows类操作系统可以安装XenConvert或StarWindConverter工具来作为镜像文件格式转换平台的基础工具，安装非常简单，在此就不在叙述；

Linux类操作系统需安装qemu-img工具来作为镜像文件格式转换平台的基础工具，安装方法如下：

已CentOS为例：

yum install qemu-img

√镜像导出前操作系统检查准备工作

Windows系统关闭防火墙，UAC、启用远程桌面

(1)关闭防火墙操作方法：开始-控制面板-windows防火墙-打开和关闭防火墙，选择关闭防火墙

(2)关闭UAC（用户帐户控制）：开始-运行-输入MSCONFIG打开系统配置-工具Tab-更改UAC设置-设置最低-重启系统生效

(3)启用远程桌面：开始-计算机-属性-远程设置-启用远程桌面

Linux 系统关闭防火墙、Selinux、Network Manager

(1)关闭Linux系统防火墙：执行命令chkconfig iptables off，重启生效

(2)关闭Selinux：修改/etc/selinux/config文件中的“SELINUX=” 为 disabled ，重启生效

(3)关闭或删除Network Manager

(4)在/etc/fstab文件中去掉mount配置

2. 镜像文件制作或导出

对于传统IDC的物理服务器主机或者其他云平台服务器主机的Windows类型，我们使用Alip2v或者DISK2VHD工具进行Windows系统C盘的镜像文件制作，这两个工具都非常简单，使用风格及步骤基本都差不多。

Alip2v windows版工具使用为示例：

工具安装注意事项：

• Alip2v工具的安装需要Microsoft Windows Installer，即微软用来运行MSI安装程序所使用的Windows程序模块，若计算机上未安装相同或更高版本的WindowsInstaller，需要自行进行安装（一般系统自带，无需安装）

• Alip2v工具的运行依赖于 NET Framework 4.0，若计算机上未安装相同或更高版本的 .NET Framework，安装包下有DotNetFX40Client 文件夹，可点击进行安装

• 运行安装包里的 setup.exe 文件，按照提示安装，即可完成 Alip2v 工具的安装

工具使用

(1)点击运行Ali-P2V，系统提示被迁移系统信息，点击下一步

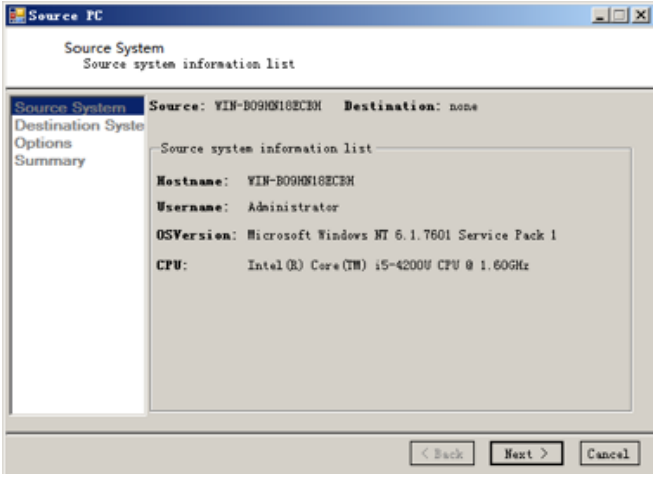


图4-24 ali-p2V启动界面

(2)选择镜像目的运行平台KVM或者Xen（在阿里云IO实例优化为KVM实例，非IO实例优化为Xen实例）、选择镜像文件格式VHD、VMDK、RAW（一般选择VHD格式），输入系统盘大小，勾选系统盘C盘（不支持数据盘迁移）

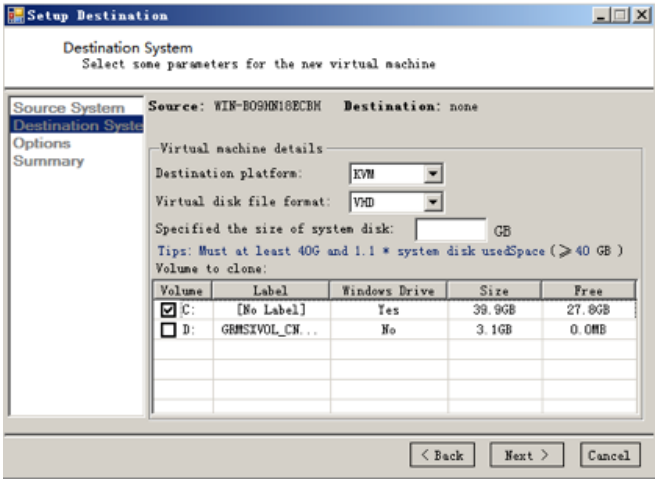


图4-25 目标平台设置界面

(3)设置无需进行克隆的文件或目录以减少转换的时间和空

间，如相关日志等信息，也可为空，不指定、设置镜像文件存储路径（注意空间是否足够）

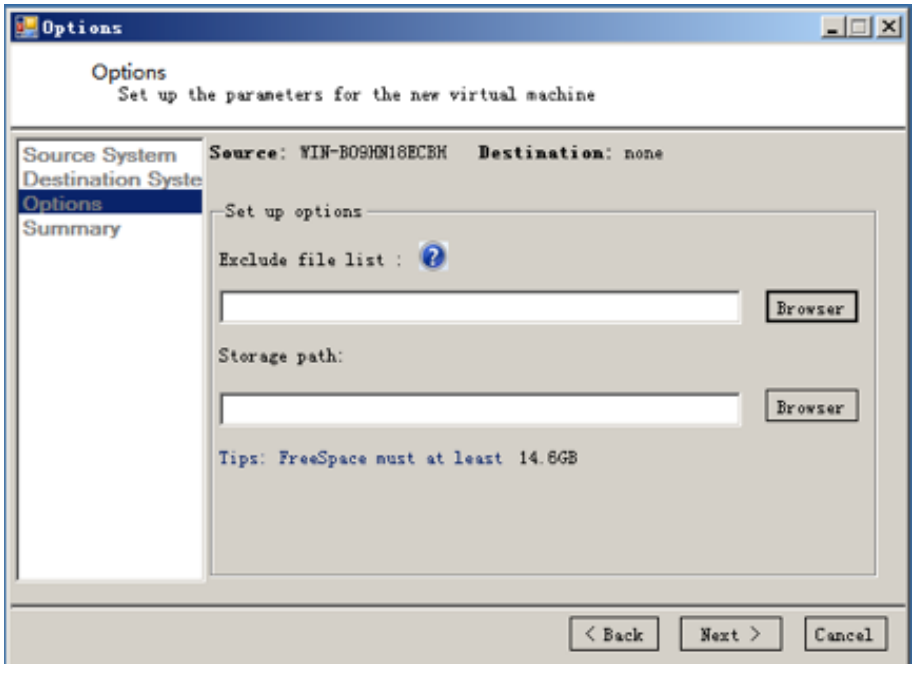


图4-26 参数选项界面

(4)确认信息后点击运行，运行完毕后镜像文件保存在目标路径下。

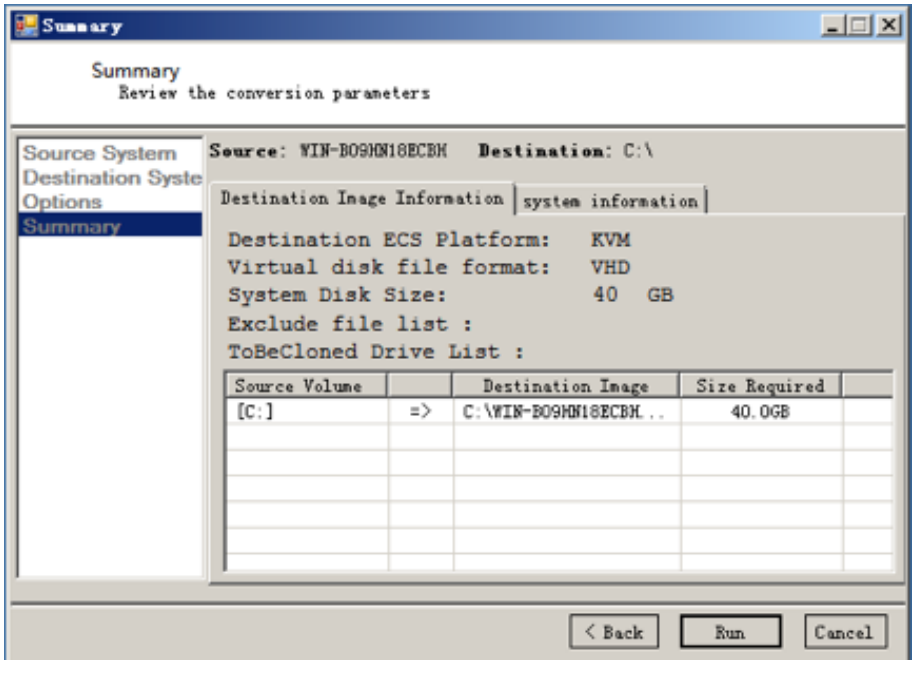


图4-27 确认运行界面

对于传统IDC的物理服务器主机或者其他云平台服务器主机的Linux类型，我们使用Alip2v或者DD工具进行Linux系统盘的导出，这两个工具导出的都是RAW格式镜像文件，RAW文件一般都比较大（和系统盘size一样大）。RAW虽然可以直接上传的阿里云使用，但是不建议这样做，可以使用qemu-img转换为VHD后上传，可以节约网络传输时间。

Alip2v linux版工具使用为示例：

限制说明：

• 外设：目前不支持外接物理设备如SAN/NAS/HBA卡/FC卡/加密设备等

- 在线迁移前确保应用已经停止,尽量减少数据不一致的情况
- 排除不需要的目录,比如数据盘挂载点,不支持网络文件系统
- 默认创建的目标镜像是稀疏的raw文件,需要自己转换成其他格式
- 默认创建的目标文件系统为ext3或ext4,不支持其他文件系统

工具使用指导

【迁移工具选项说明】

```
[root@iZbp10c5bxe2ymqddszhuuZ ~]# ./p2v
Usage: p2v [OPTION...]

P2V ("Physical-to-Virtual") involves the process of decoupling and migrating a physical server's operating system (OS), applications, and data from that physical server to
a virtual-machine guest hosted on a virtualized platform.

Examples:
  p2v -c                                          # Check OS environment and exit.
  p2v -C -n disk.raw --exclude=/media --exclude=/disk1  # Create a raw format disk image exclude /media and /disk1 folder

Options:
  -c, --check                check os environment, only support CentOS 5/6, RedHat 5/6, SUSE 11
  -C, --create               create a p2v disk image, image file format is raw
  -n, --name                 given as a image file name
      --exclude=PATTERN      exclude files, given as a PATTERN
  -v, --version              print program version
  -h, --help                 give this help list
```

图4-28 迁移工具帮助选项

说明:

- 首先进行环境检查，确保可以顺利迁移
- 创建新的磁盘镜像，默认是raw格式
- 默认拷贝本地根分区（除/dev,/proc,/sys），其他需要排除的目录请手动指定
- 设定镜像参数，确保目标系统可以正常启动

【环境检查】

```
[root@localhost ~]# ./p2v -c
Check OS version:                                          [  OK  ]
Check processor type:                                      [  OK  ]
Check OS release:                                          [  OK  ]
Check disk usage:                                          [  OK  ]
Check pv driver:                                           [  OK  ]
Check selinux status:                                      [FAILED]
```

图4-29 迁移工具环境检查

说明:

- 操作系统版本检查，目前只支持Linux
- 处理器类型检查，目前只支持Intel x86架构
- 发行版检查，目前只支持CentOS5/6，RedHat5/6，SUSE 11
- 本地文件系统空间检查，是否有足够的空间存放镜像
- 检查驱动，需要能够支持xen，若检查失败请先安装驱动
- 检查selinux状态，不支持开启selinux

【镜像导出执行】

```
[root@localhost ~]# ./p2v -C -n test.raw --exclude /disk1
Check OS version:                                          [  OK  ]
Check processor type:                                      [  OK  ]
Check OS release:                                          [  OK  ]
Check disk usage:                                          [  OK  ]
Check pv driver:                                           [  OK  ]
Check selinux status:                                      [  OK  ]
Create disk image:                                         [  OK  ]
Create default partition:                                  [  OK  ]
Create ext3 filesystem:                                    [  OK  ]
Clone root filesystem:                                     [  OK  ]
Change OS config:                                          [  OK  ]
Setup bootloader:                                          [  OK  ]
Umount disk image:                                         [  OK  ]

Output p2v disk image: /migration/test.raw
```

图4-30 镜像导出操作

说明：

• 在线迁移前确保应用已经停止，尽量减少数据不一致的情况

• 排除不需要的目录，比如数据盘挂载点，不支持网络文件系统

• 不支持selinux，请先关闭重启生效，然后重新进行迁移

• 默认创建的目标镜像是稀疏的 raw 文件，需要自己转换成其他格式

• 默认创建的目标文件系统为ext3或ext4，不支持其他文件系统

DD工具使用为示例：

(1)通过df和fdisk查看跟分区位置，在/dev/vda

[root@iZbp1be1ftlybmieiuqpqeZ ~]# df -k

Filesystem 1K-blocks Used Available Use% Mounted on

/dev/vda1 41151808 1649216 37405544 5% /

tmpfs 1962256 0 1962256 0% /dev/shm

//10.28.44.86/c$ 209713148 46532092 163181056 23% /mnt/samba

10.27.88.123:/share_dir

206291712 150970880 44835328 78% /mnt/nfs

/dev/mapper/p2v-lvm 30832636 2794168 26465604 10% /home

[root@iZbp1be1ftlybmieiuqpqeZ ~]# fdisk -l

Disk /dev/vda: 42.9 GB, 42949672960 bytes

255 heads, 63 sectors/track, 5221 cylinders

Units = cylinders of 16065 * 512 = 8225280 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk identifier: 0x00078f9c

Device Boot Start End Blocks Id System

/dev/vda1 * 1 5222 41940992 83 Linux

(2)通过dd命令制作镜像文件

[root@iZbp1be1ftlybmieiuqpqeZ ~]# dd if=/dev/vdc of=/mnt/nfs/centos65.raw

3. 镜像格式转换

对于有的云平台可以导出镜像文件而且基本是VHD的格式，这种情况我们可以省去镜像制作和格式转换的步骤。

在传统虚拟化平台VMware类型的虚拟主机迁移我们不用镜像制作，目前VMware虚拟主机底层虚拟磁盘文件为VMDK格式，我们自己到ESX Server中把VMDK文件拷贝到镜像格式转换平台后直接转换：

VMDK转VHD：

qemu-img convert -f vmdk vmdkfile.vmdk -O vpc vhdfile.vhd

RAW 转 VHD：

qemu-img convert -f raw centos65.raw -O vpc centos65.vhd

qemu-img convert说明：

qemu-img convert [-c] [-e] [-f format] filename [-O output_format] output_filename

当然也可以在windows系统中部署Xenconvert或者StarWindConverter工具来进行格式转换，基本是傻瓜操作，这里我就不在详细叙述。

镜像格式转换阶段主要是正对VMDK转VHD，RAW转VHD；

注意：

VMware的虚拟磁盘vmdk文件在创建的时候可以选择分割的方式，这样会导致一个虚拟机有N个虚拟磁盘文件，使用XenConvert转成VHD格式只能输入一个，需要使用vmware-vdiskmanager.exe合并多个虚拟磁盘vmdk文件为一个vmdk文件。

4. 镜像文件上传并设置为自定义镜像

在云下导出或制作好镜像后需要上传的阿里云的镜像中心，上传过程中需要使用OSS服务。所以如果使用的阿里云账号还没有开通OSS服务，请先开通OSS服务；使用OSS的第三方工具客户端，OSS API或者OSS SDK把制作好的文件上传到和导入ECS用户自定义镜像相同地域的bucket里面，如果对怎么上传文件到OSS不熟悉，可以参考https://help.aliyun.com/document_detail/32185.html?spm=5176.doc32184.6.951.c6Ckyf

图4-31 镜像文件上传步骤

镜像上传到OSS后，可以在阿里云控制台发起工单申请ECS导入镜像的权限，并且主动把OSS的访问权限授权给ECS官方的服务账号。

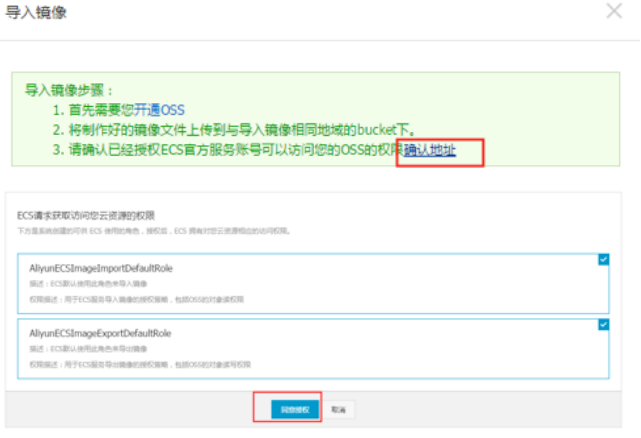


图4-32 导入镜像页面

授权完成后，进入阿里云ECS控制台导入镜像，导入前需要填写导入镜像信息表单，填写过程中需要注意镜像信息一定要正确。

图4-33 导入镜像表单信息

如果对导入表单信息有不确认的，可以参考表4-3进行确认并填写。

表4-3 镜像表单信息说明表

表单属性	属性解释
地域	请选择您即将要部署应用的地域
镜像文件 OSS 地址	直接复制从 OSS 的控制台的 Object 对象的获取地址的内容
镜像名称	长度为 2-128 个字符，以大小写字母或中文开头，可包含数字，"."，"_"或"-"
系统盘大小	Windows 系统盘大小取值：40-500GB,Linux 系统盘大小 20-500G
系统架构	64 位操作系统选择 x86_64,32 位操作系统选择 i386
操作系统类型	windows 或者 linux
系统发行版	暂时支持的操作系统发行版：windows 支持 Windows Server 2003,2008,2012 和 windows 7 ，linux 支持 CentOS，redhat，SUSE，Ubuntu，Debian，gentoo，FreeBSD，CoreOS，Other linux（请提交工单确认是否支持）.如果您的镜像的操作系统是根据 linux 内核定制开发的，请发工单联系阿里云
镜像格式	支持 RAW 和 VHD 两种格式，建议客户使用 RAW 格式，成功率会高很多，不支持使用 qemu-image 创建 vhd 格式的镜像
镜像描述	填写镜像描述信息

在镜像导入过程中，通过任务管理，找到该导入的镜像，可以对这个导入镜像进行取消任务操作。导入镜像需要耐心等待，一般需要数小时才能完成，完成的时间取决于镜像文件的大小和当前导入任务繁忙程度，可以在导入地域的镜像列表中看到这个镜像进度。

5.根据镜像启动ECS实例

镜像导入到阿里云后，可以进入阿里云ECS控制台通过上传的镜像进行实例创建；在镜像选择的时候镜像来源需要选择自定义镜像，可以在自定义镜像列表看到导入的镜像。

图4-34 根据镜像创建ECS

启动完成后可以根据以下4-4和4-5检查项列表来进入ECS实例分别进行Windows或Linux操作系统的相关检查。

表4-4Windows镜像实例检查列表

检查内容	说明
ip（内网 ip/外网 ip） 掩码 网关	1、内网 ip 校验：能通过另外一台 vm ping 通 2、外网 ip：外网 ping 通
路由	正常访问外网
密码	administrator 密码登陆
hostname	计算机-属性-高级系统设置-计算机名 修改后重启计算机
DNS	ping DNS 服务是否能 ping 通/是否能正常访问外网
默认网关	正常访问外网
host 文件	位于:C:\Windows\System32\drivers\etc 测试域名绑定
挂载数据磁盘	挂载磁盘是否成功？格式化磁盘是否成功？ 是否能正确写入文件？（check 是否存在写保护）
ntp	校验机器时间
KMS	1、运行输入框中输入“Slmgr.vbs -dlv”命令并回车 2、查看批量激活过期时间
注入启动 AliyunService 进程以及 XEN 或 KVM 模块	任务管理器查看是否存在以下进程 shutdownmon（老版本叫 shutdownmon）/AliyunService

表4-5 Linux镜像实例检查列表

检查内容	说明
ip 掩码 网关（公私网卡）	1、内网 ip 校验：能通过另外一台 vm ping 通 2、外网 ip：外网 ping 通
路由	正常访问外网
密码	root 密码
hostname	修改 hostname
dns	ping DNS 服务是否能 ping 通/是否能正常访问外网
默认网关	正常访问外网
hos 文件	/etc/sysconfig/network 修改 hostname，需要重启 reboot
ssh key	/etc/ssh/ssh_host_key(一般不会修改)
挂载数据磁盘	mount 磁盘是否成功？格式化磁盘是否成功？ 是否能正确写入文件？（check 是否存在写保护）
ntp	查看服务器时间
yum/apt 源	自动安装 yum 或 apt 软件
注入启动 gshell 进程以及 XEN 或 KVM 模块	'ps -ef | grep gshell | grep -v grep | wc -l'

五、阿里云上跨VPC和区域、账号镜像迁移实践

目前在阿里云云上的镜像迁移主要需求场景：

- 跨VPC迁移ECS实例，比如从VPC A迁移到VPC B环境中；
- 跨区域迁移ECS实例，比如从上海区域迁移到杭州区域
- 跨账号迁移ECS实例，比如从账号A迁移到账户B

阿里云提供ECS实例快照和自定义镜像（支持系统盘和数据盘）的功能，并且自定义镜像可以跨区域复制和共享给其他账号使用，基于这些功能特性我们就可以来实现跨VPC、跨区域、跨账号的镜像迁移。

跨VPC镜像迁移流程如下:

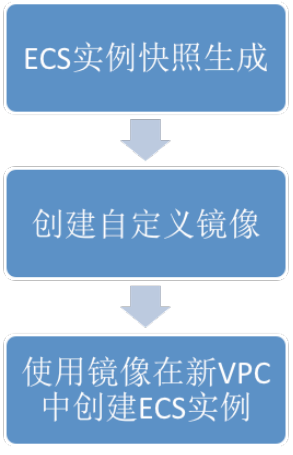


图4-35 跨VPC镜像迁移流程

跨区域镜像迁移流程如下:

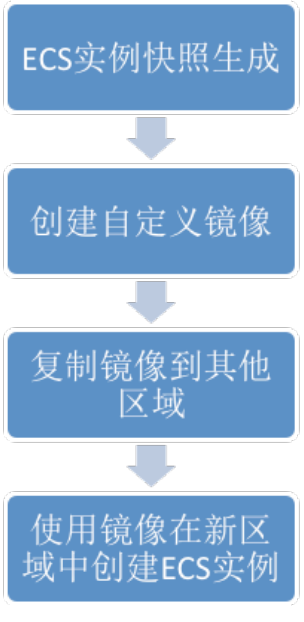


图4-36 跨区域镜像迁移流程

跨账号镜像迁移流程如下:

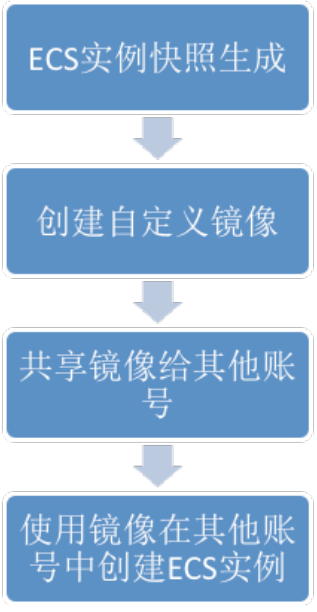


图4-37 跨帐号镜像迁移流程

1. ECS快照生成

所谓快照，就是某一个时间点上某一个磁盘的数据备份，需要注意的是，如果需要保持数据的一致性，需要停机或停服务的方式进行快照；

ECS快照操作流程：

(1)登录云服务器管理控制台

(2)单击实例所在的地域，然后单击左侧导航的实例。单击实例的名称，或在实例右侧，单击管理

图4-38 ECS管理控制台

(3)单击左侧的本实例磁盘，对系统盘和数据盘进行创建快照

图4-39 ECS快照页面

2.创建自定义镜像

镜像是云服务器 ECS 实例运行环境的模板，一般包括操作系统和预装的软件。

通常自定义镜像来源一下渠道：

• 根据现有的云服务器 ECS 实例的快照创建自定义镜像

• 把线下环境的镜像文件导入到ECS的集群中生成一个自定义镜像。

进入ECS实例管理控制，点击管理，点击在左侧本实例快照，确定快照的磁盘属性是系统盘，数据盘不能单独用于创建镜像。然后单击，创建自定义镜像。

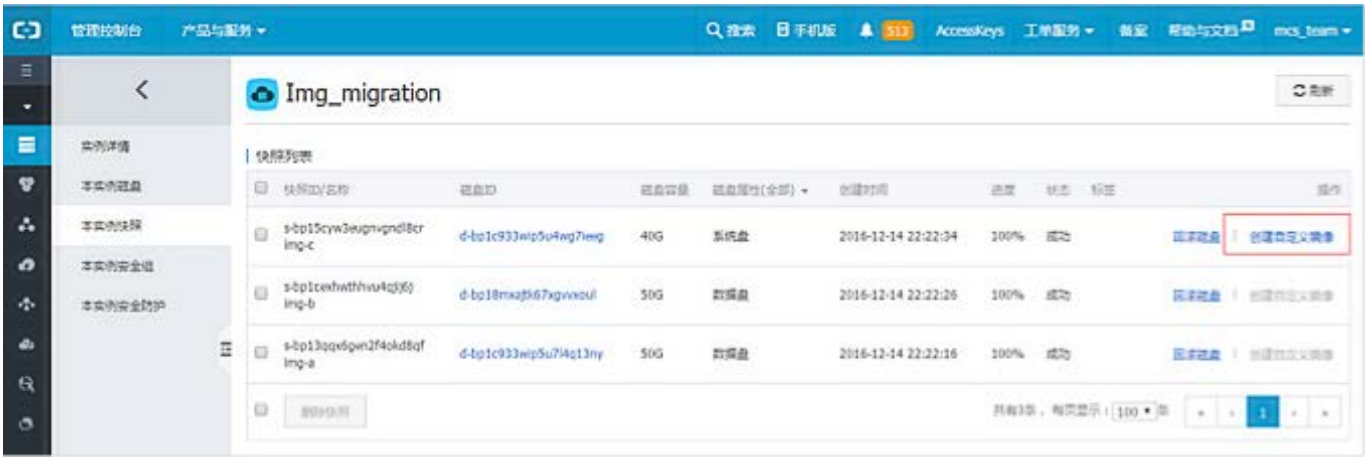


图4-40 创建自定义镜像

3.镜像跨区域复制

当前跨地域复制镜像处于公测状态，如需使用可以提交工单申请白名单，工单中注明需复制镜像的总大小信息。自定义镜像是不能跨地域使用的，但是如果需要跨地域使用自定义镜像，可以通过复制镜像的方式，可以把当前地域的自定义镜像复制到其他地域进行镜像迁移复制。

复制镜像需要通过网络把源地域的镜像文件传输到目标地域，复制的时间取决于网络传输速度和任务队列的排队数量。

复制自动义镜像的步骤如下：

(1)登录云服务器管理控制台；

(2)单击左侧导航中的镜像，可以看到镜像列表；

(3)选择页面顶部的地域；

(4)选中需要复制的镜像，镜像类型必须是自定义镜像，单击复制镜像。在弹出的对话框中，可以看到选中镜像的 ID；

(5)选择需要复制镜像的目标地域；

(6)输入目标镜像的名称和描述；

(7)单击确定，镜像复制任务就创建成功了；

4.镜像共享

在阿里云可以把自己的自定义镜像共享给其他用户，该用户可以通过管理控制台或 ECS API 查询到其他账号共享到本账号的共享镜像列表；被共享用户可以使用其他账号共享的镜像创建 ECS 实例和更换系统盘。

分享镜像的步骤如下：

(1)登录云服务器管理控制台；

(2)单击左侧导航中的镜像，可以看到镜像列表；

(3)选择页面顶部的地域；

(4)选中需要复制的镜像，镜像类型必须是自定义镜像，单击共享镜像；

(5)在弹出的对话框中，选择账号类型和输入阿里云账号。有两种账号类型：

• Aliyun账号，输入要共享给其他用户的阿里云账号（登陆账号）。

• AliyunID，输入要共享给其他的阿里云账号ID。 AliyunID 可以从阿里云官网的用户中心获取：账号管理 > 安全设置 > 账号ID。可通过下面链接直接登录访问：https://account.console.aliyun.com/#/secure

(6)单击共享镜像，完成自定义镜像的共享；

阿里云客户服务部

阿里云通过自助、智能、售后工程师团队、生态合作伙伴多种模式，为全球200多个国家和地区的客户提供7x24技术支持，保障客户云上系统的稳定运行。为客户提供可信赖可托付的服务支持，提升客户云上体验是阿里云服务团队的第一宗旨。

服务产品
Service Products

基础服务
- 丰富的产品介绍文档、视频和最佳实践
- 7×24工单&电话支持
- 支持200+国家和地区的多语言服务

智能服务
- 云博士智能机器人
- 智能服务对话分析
- 一站式智能诊断平台

企业服务
- 企业专线电话
- 7x24小时钉钉群服务
- 专属TAM（技术服务经理）
- “云台”提供工具化支撑

专家服务
- 上云迁移
- 云上系统专家保障
- 技术管理服务
- 专业技术服务

区域服务
- 阿里云认证的区域服务提供商
- 本地化服务优势
- 上云前、中、后不同阶段提供相应的服务支持

服务渠道
Service Channels

查阅文档　咨询智能机器人　寻求工程师支持　建议与反馈




帮助文档	提供阿里云全线产品文档和常见问题的高效搜索，让用户更方便地获取云产品使用的信息
智能服务	云博士：提供PC端，APP端，钉钉群等多端智能问答服务 / 智能对话分析：全量实时质检，保障服务质量 / 云台：一站式智能诊断平台
社区问答	面向开发者的开放型技术问答平台，服务于云计算技术全生态，帮助技术人快速成长与发展
云享	凝聚阿里云多年服务经验，携手合作伙伴与业界专家，匠心打造云服务技术共享
工单支持	阿里云售后支持配备了经验丰富的技术专家，提供7×24工单支持服务，确保及时快速解决问题
95187热线	不管白天黑夜，阿里云的客服人员都会通过95187为用户提供温暖专业的电话支持服务
聆听	客户监督和驱动阿里云不断完善的平台，通过该平台提交的建议，会专人评审、落实到产品优化

阿里云支持频道
https://www.aliyun.com/support

阿里云客户满意中心
http://weibo.com/p/1006062935804003

SAP在阿里云上的HA配置

回雁
阿里云技术专家

一、SAP ERP上云背景

SAP是世界上最大ERP软件提供商，SAP在全世界拥有几十万家企业客户，几乎所有行业的业务流程都可以被SAP ERP管理起来。在传统的IT环境里，比较典型的SAP ERP应用是搭建在小型机UNIX系统上的。当云计算时代来临后，各大云厂商纷纷与SAP合作，将SAP ERP迁移到云上，以便给企业客户更多的基础设施选择，从而降低成本并享受到比传统部署方式更大的灵活性。

2016年8月，阿里云与SAP正式牵手合作，两家技术人员走到一起研究将SAP ERP和SAPHANA等系统迁移到云上。目前，阿里云已经完成了数款ECS机型对SAP HANA的认证测试，同时对SAP ECC的支持，阿里云也在客户的上云实践中进行探索。

二、部署架构

小型机都有厂家提供的HA软件，比如IBM的PowerHA，这样的方案也是被SAP认证通过的。Linux上的环境，SUSE LINUX也是被 SAP 认证通过的，但是在阿里云上，尤其是HA的方案，还没有完全成型，本文档重点研究的就是如何在阿里云ECS Linux环境搭建一套可行的HA方案提供给SAP系统使用。

我们以SAP AFS（Apprarel and Footwear Solution），即SAP在服装鞋帽行业的解决方案为例，来搭建在阿里云ECS上的HA。

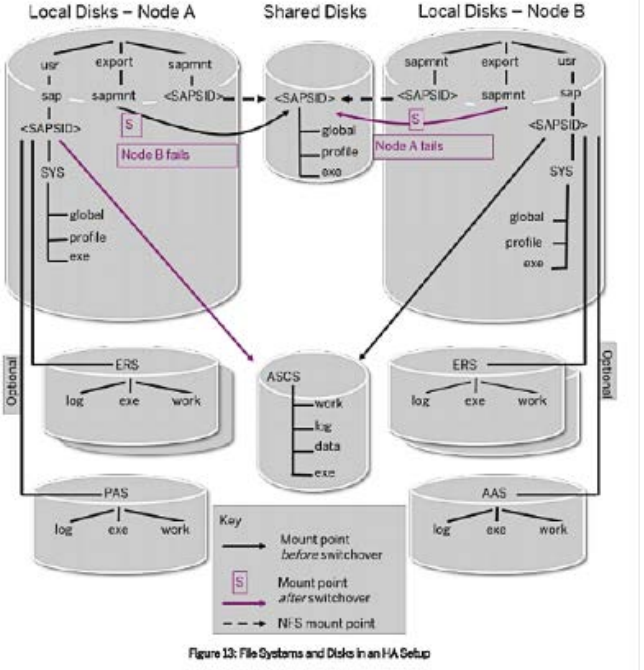


Figure 13: File Systems and Disks in an HA Setup

上图中是SAP实例的官方部署建议图，基础软件环境的安装和部署情况：阿里云SUSE Linux 11 SP3 + SAP ECC 6.0 EHP6 + SAP NetWeaver 7.3.1 + Sybase 15.7。

上面的架构图中没有体现DB，我们必须把DB也考虑进来。同时，根据SAP应用的特点，我们在部署上也做了一些改变，最重要的就是将SAPSID部署在了一个能多节点同时挂载和读写的存储上，避免了在两个节点做切换的过程，提升系统的可用性。

节点A部署的有：阿里云SUSE Linux + ASCS + SAP primary Application Server；

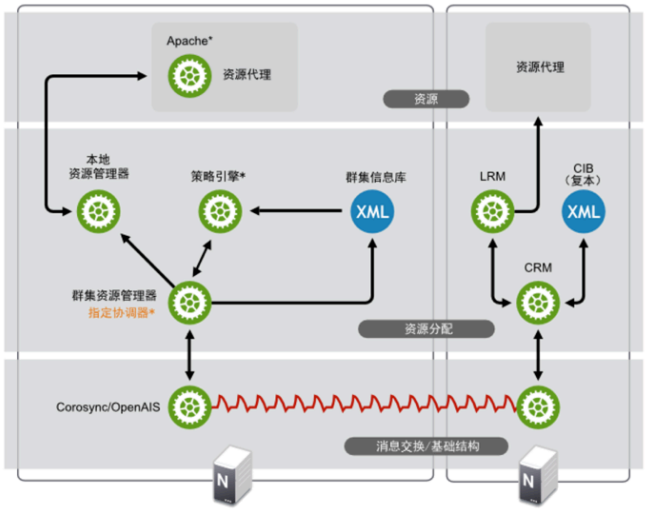
节点B部署的有：阿里云SUSE Linux + DB + SAP Secondary Application Server。

在这个架构中，pplication Server在ASCS后端，ASCS提供客户端的统一访问，然后分发到后端的application Server。所以，我们发现整个架构存在2个单点，一个是DB，一个是ASCS。HA的目标就是解决这两个单点，在发生故障时能够进行自动切换，保证系统整体的高可用性。

三、方案评估

我们在两个方案上做了评估和选择，一个方案是与SUSE Linux配合紧密的HAE（High Availability Extension），另一个是开源的keepalived。

3.1 HAE介绍



HAE的层次结构主要有：信息交换/基础结构、资源分配和资源层。第一层和第二层基本是HAE自己控制和管理的，主要用来协调基础信息，集群配置，控制上层资源等。对用户来说，可以配置最多的一层在最上层，即资源代理（resource agent）。资源代理是已写入的用来启动、停止和监视某种服务（资源）的程序（通常是shell脚本）。资源代理仅由 LRM 调用。第三方可将他们自己的代理放在文件系统中定义的位置，这样就为各自的软件提供了现成群集集成。目前，HAE已经集成了适用于市面上大多数主流应用和软件的资源代理，包括mysql、lvm、oracle等，包括SAPDatabase和SAPInstance也是已经包装好的资源代理。

经测试，SAPDatabase可以正常管理本测试中用到的sybase15.7，但是SAPInstance代理由于我们的部署架构将一个实例在两个节点进行了拆分，所以无法被管理。在尝试完全重写对ASCS的资源代理时，由于对ASCS的原理理解不够深入，在规定的时间内没能调整好脚本到预期的状态，所以最终暂时放弃了HAE方案。

理论上说，HAE是比较成熟的商业软件，提供图形化的配置运维管理界面，降低了对客户的技术要求，而且还有软件厂家的售后服务。如果可能，还是会建议客户首选HAE。

3.2 Keepalived介绍

Keepalived是一个基于VRRP协议来实现的LVS服务高可用方案，可以利用其来避免单点故障。一个LVS服务会有多台服务器运行Keepalived，一台为主服务器（MASTER），其他为备份服务器（BACKUP），这些服务器是通过优先级来选举出来最高优先级的那个节点为主服务器的。这些节点对外表现为一个虚拟IP，主服务器会发送特定的消息给备份服务器，当备份服务器收不到这个消息的时候，即主服务器宕机的时候，备份服务器就会接管虚拟IP，继续提供服务，从而保证了高可用性。

在实际测试过程中，发现keepalived的配置，尤其是对脚本的控制比较简单直接，容易理解，也好上手，最终也及时地完成了测试和客户需求。所以我们以keepalived为例来说明在阿里云上如何配置对SAP的HA。

四、实施过程

所有的HA都要满足以下几点，在阿里云上实现也不能例外：

- VIP：为需要做高可用保证的应用服务提供统一的对外IP，能在集群中节点进行漂移和切换；
- 共享存储：上面部署共享的数据，提供在主节点down掉后，备节点可以挂载的可能性；
- 状态监控、故障切换功能：异常/故障状态下的处理机制实现。

我们在下面的实施步骤中重点说明和解决这几个问题，我们会创建两个VRRP_INSTANCE，分别管理DB和ASCS。

4.1 VIP的创建

阿里云提供一款叫HaVIP的产品，可以帮助客户在指定的

VPC网络内创建最多5个能够在该CIDR网段内没有被占用的VIP。客户可以通过提工单的形式申请HAVIP，申请成功后，在VPC的配置界面将出来一个新的菜单：“高可用虚拟IP”。

根据提示选择交换机，创建2个IP，一个IP是172.16.32.30，这个是DB的VIP；一个IP是172.16.32.40，这是ASCS的VIP。这两个IP创建出来先预留，不要使用。

如果您很不幸没有申请到havip，也不要沮丧，还有一款开源软件：n2n VPN可以解决云上的虚拟网卡和多IP的问题。可以移步这个地址来参考实现：http://blog.csdn.net/yetyongjin/article/details/7419894。

4.2 共享存储

阿里云已经在国内所有region推出公测版本的共享块存储，即SSD云盘。在正式商业化前，我们在阿里云上暂时只能使用开源的iscsi来实现共享。在安装配置共享存储前，我们需要熟悉一下iscsi的相关概念，比如iscsi-target和iscsi-initiator，在规划中，我们使用了4台target，每台target上挂载一个800GB的SSD云盘。配置HA的2个节点自然就是iscsi-initiator了。

目前主流的linux发行版都已经自带了iscsi，阿里云上的suse linux、centOS也不例外，安装一下就好了。

在SUSE Linux 11SP3的版本中，配置iscsi-target的步骤：

(1)打开yast，选择Network-Service-iSCSI Target，系统会提示没有安装，选择安装，稍等一会儿就安装好了；

(2)安装完后，就可以退出yast了，如果继续用yast来做配置，会发现后面共享出去的磁盘是没有办法被识别的；

(3)此时，需要编辑一下/etc/ietd.conf文件，这个文件的格式大约是：

Target iqn.2017-03.com.example:05261bd3-258a-4baa-935f-1700be499e44

Lun 0 Path=/dev/vdb,Type=blockio

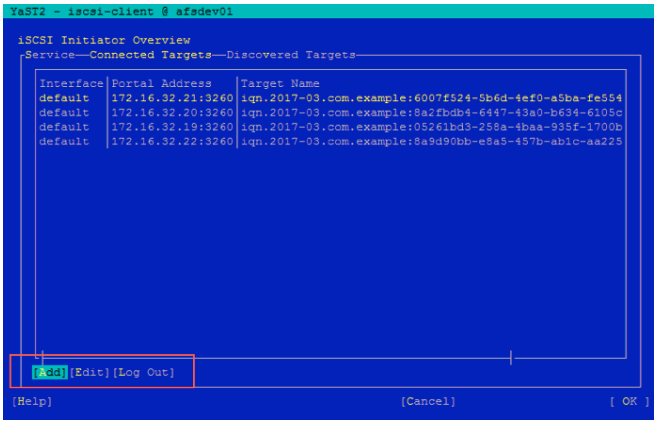
这里的关键是第二行的Path和Type，Path定义了我们要共享的本地磁盘路径，Type定义了共享磁盘的类型。

(4)然后就可以执行#service iscsitarget start启动iscsi-target了；

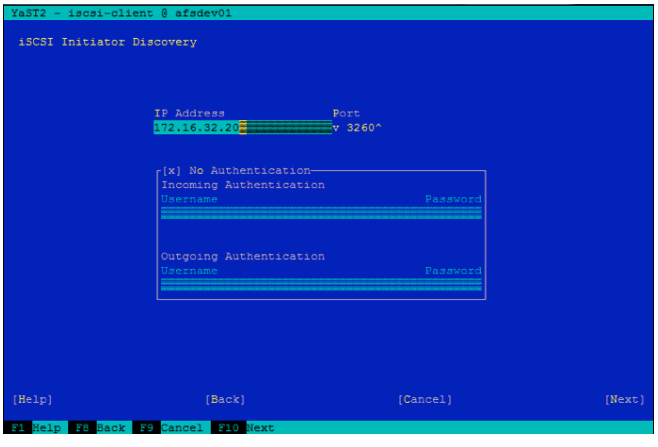
(5)每个iscsi-target上都做一番这样的配置；

(6)在两个HA节点上，打开yast，选择Network-Service-iSCSI Initiator，按照提示安装；

(7)安装完毕，在Conneted Targets界面选择Add添加



(8)填上iscsi-target的IP地址去发现共享出来的磁盘，参照下图



(9)然后NEXT，然后再参照步骤（6）-（8）添加，直到把所有的target都添加进来。此时fdisk-l和lsscsi都能看到新挂载的磁盘，保证2个节点都能看到；

(10)在任意一个节点，使用lvm可以创建vg/lv/fs，将这些资源分配和挂载给需要的节点使用。这里我们创建了两个vg:sapvg和datavg。sapvg存放ASCS数据，挂载了3个目录：/export/sapmnt、/export/usr/sap/trans和/usr/sap/TST/ASCS10，默认在节点1；datavg存放数据库数据，挂载了1个目录：/sybase，默认在节点2。

4.3 NAS

NAS还要求能给两个节点提供一个可以同时挂载并发读写的2个文件系统，最初使用的NFS，将一个节点的本地目录共享给另一个节点，在实际使用中发现当服务节点异常时，使用NFS共享目录的节点会受到很大的影响。比如NFS Server已经挂掉了，那么client节点因为无法正常umount

NFS目录会导致涉及到NFS目录的所有操作都长时间hang住，这样的架构将会给整个系统带来非常大的稳定性隐患。

于是，我们更换了一个方案，将阿里云的NAS引入了方案中。实践发现，NAS作为性能要求不那么严苛的共享文件系统是十分合适的，也非常稳定。

所以，两个节点都挂载了2个NAS文件系统：

112f4483d9-byb70.cn-beijing.nas.aliyuncs.com://sapmnt

1f8dc4b72a-yaq0.cn-beijing.nas.aliyuncs.com://usr/sap/trans

4.4 部署SAP

这部分不是本文的重点，按照第一章的设计将SAP AFS部署好即可，其部署方式与云下SUSE LINUX系统没有差异，前提是先把HAVIP对应的存储资源、NAS共享文件系统都准备好。

4.5 部署HA

(1)下载keepalived

我们希望尽可能下载使用最新版本，但是在实际中发现SUSE 11 SP3和keepalived的兼容性还是存在比较严重的问题的。经过反复测试，我们选用了1.2.20版本。

下载地址：http://www.keepalived.org/download.html

(2)安装keepalived前的准备

zypper -n install gcc gcc-c++ kernel-source make glib*

可能还需要升级openssl，依然是去网上找openssl 1.1.0e的包，自己make和make install安装。这里可能还有其他的包需要安装，在安装keepalived时会进行检查，如果不通过，就回到这一步来补齐。

(3)安装keepalived

#tar zxvf keepalived-1.2.20.tar.gz

#cd keepalived-1.2.20

#./configure --prefix=/usr/local/keepalived

#make

#make install

(4)配置keepalived

请注意，keepalived安装完后，会自动生成一个配置文件：/usr/local/keepalived/etc/keepalived/keepalived.conf。修改这个文件是没有意义的，因为keepalived认可的配置文件是：/etc/keepalived/keepalived.conf！！！

在正式进行配置之前，请务必先保证手工执行脚本能完美实现对vrrp实例资源的完整管理和切换，如果手工执行都无法正常运行，那么加上HA自动化后将会有更多不可预知的情况发生！！！

当你已经确定手工执行脚本已经没有任何问题，就可以配置keepalived了。2个节点（afsdev01和afsdev02）都得配置，修改完成后，我们的节点1配置文件长这样子：

```
! Configuration File for keepalived

global_defs {
   notification_email {
    huiyan.hq@alibaba-inc.com
   }
    notification_email_from  huiyan.hq@alibaba-inc.com
   smtp_server 127.0.0.1
   smtp_connect_timeout 30
   router_id afsdev
}

vrrp_instance ascs {
    state MASTER
    interface eth0
    virtual_router_id 20
    priority 100
    advert_int 1

authentication {
       auth_type PASS
       auth_pass 1111
       }

   virtual_ipaddress {
       172.16.32.40/24 dev eth0 label eth0:1
   }

#   track_script {
#      check_service_down
#   }

unicast_src_ip 172.16.32.17
   unicast_peer {
```

```
        172.16.32.18
    }

notify_master /root/init_master.sh
notify_backup /root/init_backup.sh
}

vrrp_instance db {
    state BACKUP
    interface eth0
    virtual_router_id 30
    priority 40
    advert_int 1

authentication {
        auth_type PASS
        auth_pass 1111
        }

    virtual_ipaddress {
        172.16.32.30/24 dev eth0 label eth0:0
    }

#    track_script {
#        check_service_down
#    }

unicast_src_ip 172.16.32.17
    unicast_peer {
        172.16.32.18
    }

notify_master /root/init_db_master.sh
notify_backup /root/init_db_backup.sh
}
```

节点2配置文件长这样子：

```
! Configuration File for keepalived

global_defs {
   notification_email {
     huiyan.hq@alibaba-inc.com
        }
        notification_email_from huiyan.hq@alibaba-inc.
com
        smtp_server 127.0.0.1
        smtp_connect_timeout 30
        router_id afsdev
    }

    vrrp_instance ascs {
        state BACKUP
        interface eth0
        virtual_router_id 20
        priority 40
        advert_int 1

    authentication {
            auth_type PASS
            auth_pass 1111
            }

        virtual_ipaddress {
            172.16.32.40/24 dev eth0 label eth0:1
        }

    #    track_script {
    #        check_service_down
    #    }

    unicast_src_ip 172.16.32.18
        unicast_peer {
            172.16.32.17
        }

    notify_master /root/init_master.sh
    notify_backup /root/init_backup.sh
    }

    vrrp_instance db {
        state MASTER
        interface eth0
        virtual_router_id 30
        priority 100
```

```
    advert_int 1

  authentication {
      auth_type PASS
      auth_pass 1111
      }

    virtual_ipaddress {
      172.16.32.30/24 dev eth0 label eth0:0
    }

  #   track_script {
  #      check_service_down
  #   }

  unicast_src_ip 172.16.32.18
    unicast_peer {
      172.16.32.17
    }

  notify_master /root/init_db_master.sh
  notify_backup /root/init_db_backup.sh
  }
```

这两个配置文件定义了两个vrrp实例，实例中的节点优先级，VIP 资源，实例在成为 master 时执行的脚本，在成为backup时执行的脚本。

特别提醒一点，正常VRRP协议是要通过组播来实现的，阿里云对ECS网络禁用了组播，所以我们需要使用点对点的模式，即“unicast”方式。

这里我们定义希望master节点选举出来后，就远程到对端把服务下掉，然后在自己的节点启动服务。相应的脚本：

```
  init_master.sh
  echo “Now is “ `date` >> /tmp/master.log
  echo “I’m afsdev02, I am the master node now” >> /tmp/master.log
  timeout 10 ssh afsdev01 /root/sap_stop.sh >> /tmp/master.log
  timeout 5 ssh afsdev01 umount -f /usr/sap/TST/ASCS10 >> /tmp/master.log
  timeout 3 ssh afsdev01 umount -f /export/sapmnt >> /tmp/master.log
  timeout 3 ssh afsdev01 umount -f /export/usr/sap/trans >> /tmp/master.log
  timeout 3 ssh afsdev01 vgchange -a n sapvg >> /tmp/master.log
  vgchange -a y sapvg >> /tmp/master.log
  mount/dev/mapper/sapvg-lvsapmnt/export/sapmnt
  mount/dev/mapper/sapvg-lvtrans /export/usr/sap/trans
  mount/dev/mapper/sapvg-lvascs/usr/sap/TST/ASCS10
  /root/sap_start.sh >> /tmp/master.log
  echo “`date` Master is ready!!!” >> /tmp/master.log

  init_db_master.sh
  echo “`date`: Receiving command to being master, stop resources on the other node…”
  timeout 3 ssh afsdev01 ifconfig eth0:0 172.16.32.30/24 up
  timeout 3 ssh afsdev01 /root/db_stop.sh >> /tmp/master_db.log
  timeout 3 ssh afsdev01 ifconfig eth0:0 down
  timeout 3 ssh afsdev01 umount -f /sybase
  timeout 3 ssh afsdev01 vgchange -a n datavg >> /tmp/master_db.log
  echo “`date`: I’m master DB node, starting re-sources…” >> /tmp/master_db.log
  vgchange -a y datavg >> /tmp/master_db.log
  mount /dev/mapper/datavg-lv_sybase /sybase
  /root/db_start.sh >> /tmp/master_db.log
  echo “`date`: Master DB is online now” >> /tmp/master_db.log

  sap_start.sh
  su - tstadm -c “startsap R3 ASCS10 afsdev”

  sap_stop.sh
  su - tstadm -c “stopsap R3 ASCS10 afsdev”

  kill_process(){
      pid=`ps axu | grep -w START_ASCS10_afsdev | grep -v grep | awk ‘{print $2}’`
      for i in $pid
      do
          kill $i
```

```
        done
    }

    kill_process

    db_start.sh
    su - sybtst << EOF
    cd /sybase/TST/ASE-15_0/install
    nohup ./startserver -f /sybase/TST/ASE-15_0/install/RUN_TST &
    nohup ./startserver -f /sybase/TST/ASE-15_0/install/RUN_TST_BS &
    EOF

    db_stop.sh
    su - tstadm -c "stopdb"
```

如果希望HA的管理更精细，我们还可以加入track_script，定义一段状态判断的脚本，比如定义每个几秒判断某个进程是否存在，如果存在就返回正常，不存在就返回异常，异常时将该节点的优先级调低多少点，从而导致主节点的优先级低于其他备节点，重新选举出一个新的主节点来，最终实现了instance的切换。

4.6 其他的配合工作

为了能让系统在重启后也能支持SAP实例的正常工作，我们还需要增加一些启动项，比如自动挂载NAS目录，修改一些目录和文件的权限，以免在vg的切换过程中发生异常。

```
#vi /etc/init.d/after.local，内容：
mount -t nfs -o vers=3,nolock,proto=tcp 112f4483d9-byb70.cn-beijing.nas.aliyuncs.com:/ /sapmnt
mount -t nfs -o vers=3,nolock,proto=tcp 1f8dc-4b72a-yaq0.cn-beijing.nas.aliyuncs.com:/ /usr/sap/trans
chown -R tstadm:sapsys /sapmnt/TST
chown -R tstadm:sapsys /usr/sap/trans/EPS
chown -R tstadm:sapsys /usr/sap/trans/bin
chown -R tstadm:sapsys /usr/sap/trans/buffer
chown -R tstadm:sapsys /usr/sap/trans/cofiles
chown -R tstadm:sapsys /usr/sap/trans/data
chown -R tstadm:sapsys /usr/sap/trans/etc
chown -R tstadm:sapsys /usr/sap/trans/log
chown -R tstadm:sapsys /usr/sap/trans/sapnames
chown -R tstadm:sapsys /usr/sap/trans/storage
chown -R tstadm:sapsys /usr/sap/trans/tmp
```

4.7 启动keepalived

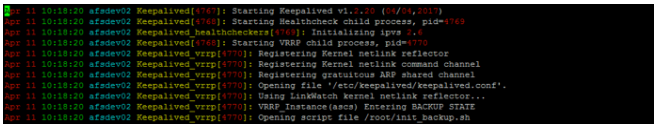
当确认配置和脚本都没有问题，就可以在两个节点分别启动keepalived了。

执行：# /usr/local/keepalived/sbin/keepalived

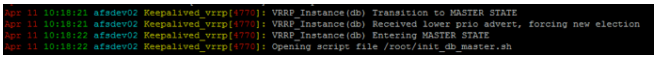
会看到有几个进程启动：

```
#ps -fe|grep keepalived
root     4768     1  0 10:18 ?        00:00:00 /usr/local/keepalived/sbin/keepalived
root     4769  4768  0 10:18 ?        00:00:00 /usr/local/keepalived/sbin/keepalived
root     4770  4768  0 10:18 ?        00:00:01 /usr/local/keepalived/sbin/keepalived
root     11411  9870  0 21:00 pts/1    00:00:00 grep keepalived
```

然后观察keepalived的日志文件，默认为：/var/log/messages，能看到这样的记录，代表这keepalived启动。



紧接着能看到的提示，这意味着对于VRRP_Instance(db)来说，本节点被选举为主节点，并执行主节点定义的脚本：



然后，需要去脚本定义的日志文件查看，我们所执行的动作是否执行成功。

4.8 测试keepalived切换

在我们配置的keepalived中，支持两类失败的资源切换：keepalived高可用程序失败和ECS节点失败。

测试过程：

(1)确认该节点的资源和keepalived运行正常；

(2)强制 shutdown 该节点或杀掉keepalived进程；

(3)观察另外一个节点该原来运行在对端节点的一系列资源接管过来，恢复在短暂中断后自动恢复，切换时间在15秒到 120 秒不等。这个切换速度与小型机的环境相当；

(4)重新启动被关闭的节点，重新运行keepalived；

(5)观察重新启动的节点在数秒内接管回来自己的资源。

如果我们定义了对异常状态的检测，我们可以再加入异常状态的场景测试内容。

同时，由于阿里云ECS底层物理网卡做了bond，加上物理机NC的自动化运维和故障热迁移等都由阿里云负责，所

以在云下要考虑的一种场景，即网络或网卡故障的情形是可以不用考虑的。当然，有一种例外，即有人执行了#fconfig eth0 down，这会造成脑裂（brain-split），要解决这种异常，也需要在keepalived的配置中加入另一种track_script，即每隔一段时间ping一下网关地址，如果连续多次没有正常返回，那么就认为自己已经异常，执行一个脚本把自己的keepalived服务和系统的 SAP 服务一起杀掉，避免脑裂的发生。这些配置都与云下使用keepalived是相同的，不再单独做实现和测试了。

阿里云上Oracle 11gR2 RAC部署方案

回雁
阿里云技术专家

一、应用背景

传统IT架构以数据库为中心，而商业数据库最流行且市场份额雄踞多年第一的就是OracleRDBMS，Oracle数据库是Oracle公司最重要且最成功的一款产品，所以Oracle几乎成了传统关系型数据库的代名词。Oracle提供了多种平台的支持，阿里云ECS作为底层以KVM虚拟化为基础的云服务天生提供了对Oracle的支持，今天有许多的Oracle数据库运行在阿里云ECS上，而传统企业用户应用场景里最为常见的Oracle RAC则成为了大家关注的焦点。作为同时满足数据库HA和负载均衡的产品级解决方案，在阿里云上是否能够成功部署和使用呢?

细心的用户可能发现，阿里云早在2016年深圳云栖大会就官方发布了对Oracle RAC的支持，但是相关产品却一直没能同步推出，事实就是我们早在2015年年底就内部测试实现了对Oracle RAC的部署支持，但由于种种原因而延缓了向市场的推广。

2017年上半年，阿里云紧密推出两款新产品：共享块存储和ECS多网卡。这两款产品目前还在公测和邀测阶段，即将正式商业化。共享块存储和ECS多网卡将打通众多关键云下应用上云的最后一公里，为用户提供更多的便利。在我们能正式体验到新产品之前，阿里云服务技术中台团队也将云上的Oracle RAC安装配置手册放出，希望能给大家提供更多不同的体验和选择。本期的部署方案我们先介绍开源版，后续我们再介绍阿里云产品版。

二、安装说明

我们再来分析一下RAC对部署环境的需求，最主要解决问题：一是共享存储，二是多网卡多IP配置，三是ECS多物理机部署。我们目前有两大思路来建设解决方案：

方案一、完全依赖阿里云产品。共享存储产品2015年已经完成内测，且成功实现RAC部署；ECS多网卡可以给ECS提供默认网卡外的多个新建网卡，结合HaVIP提供多ECS共享多IP能力，多网卡多IP配置也能满足。除HaVIP需要开白名单申请外，另两个产品还在测试阶段，该方案在两个产品正式商业化后刊文介绍。

方案二、借助开源产品。经过测试，iSCSI可以完美解决ECS间存储共享问题，而n2n VPN则可以在多个能同时访问某一个公共服务器的ECS之间轻松建立一个自主的多IP网络环境。

本手册以VPC环境中如何使用开源产品搭建Oracle RAC为例进行编写，在经典网络环境或者专有云里，该手册一样适用。对于第三个需求，现阶段在部署时与阿里云工程师确认一下，ECS是否在同一个物理机即可，未来我们也有对应的产品来保证用户申请的 ECS 不在同一个物理机。

值得说明的一点：本文讨论的是技术和方案本身，在阿里云上安装和使用Oracle仅用于测试和教学用途，如果作为商业用途，请用户联系Oracle公司获得正式许可。否则，因此带来的法律风险，请使用者自行承担。

三、GRID安装前的系统配置工作

以下各安装步骤，若无特殊说明，RAC的每个节点均需执行。

1. RAC对系统的要求

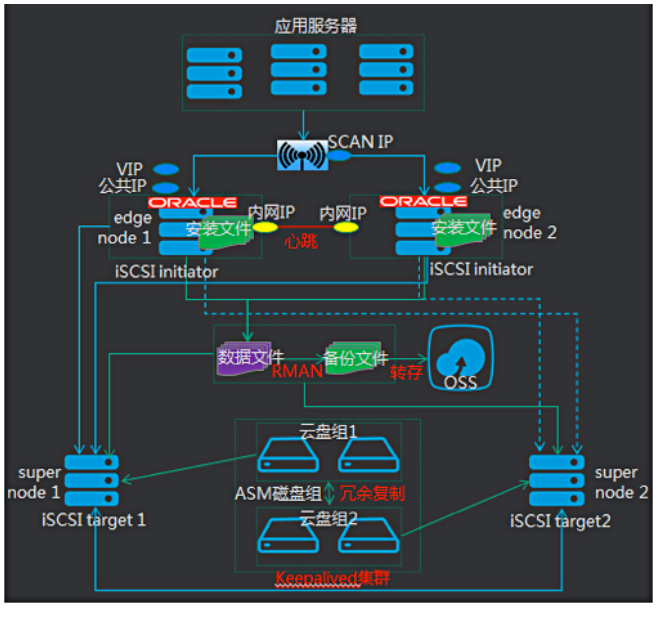
数据库节点：2个ECS，选用阿里云提供的CentOS 6.5或以上版本的公共镜像。数据库节点同时还是共享存储isc-si-initiator节点和多网卡n2n VPN的边缘节点。

公共服务节点：作为数据库集群的公共服务节点，提供共享存储和多网卡能力。同时作为共享存储iscsi-target和n2n VPN的超级节点存在。

磁盘：SSD云盘，至少申请2个，1个大小为10G左右，存储CRS数据。1个存放Oracle数据，大小根据业务而定。两个云盘挂载给公共服务节点。

部署架构参见下图：为了避免共享服务器的单点存在，我们可以再增加1台备份服务节点。备份公共服务节点也可以挂载规格相同的SSD云盘，通过iSCSI共享给数据库集群，数据库集群ASM将2套磁盘做成内部冗余，可以防止磁盘的单点故障。而通过keepalived HA软件可以把n2n VPN服务进程做成高可用服务，以避免多网卡n2n VPN服务单点失败。

为了表述方便，我们实际中只测试了一个公共服务节点的场景。



IP规划：

	Private IP	Public IP	Virtual IP	SCAN IP
节点 1	1 个，ECS 创建时已具备。本例中，采用 172.18.0.6	1 个，与 Private IP 不同网段，需要用 n2n VPN 手工添加。本例中采用 192.168.0.11	1 个，与 Public IP 同网段，在配置文件中配置即可。本例中采用 192.168.0.21	1-3 个，与 Public IP 同网段，在配置文件中配置即可。本例中采用 192.168.0.31
节点 2	1 个，ECS 创建时已具备。本例中，采用 172.18.0.7	1 个，与 Private IP 不同网段，需要用 n2n VPN 手工添加。本例中采用 192.168.0.12	1 个，与 Public IP 同网段，在配置文件中配置即可。本例中采用 192.168.0.22	

OSS：1个bucket，配置好ossfs，挂载给ECS。OSS相对云盘块存储容量更大，价格更低，适合存放备份文件。

2. 安装配置iSCSI

(1)选定target和initiator

iSCSI架构包含1个target(服务端)和多个initiator（客户端），最好申请单独的ECS做iSCSI的target，而RAC的节点则自然成为initiator。

target地址：172.18.0.5

initiator地址：172.18.0.6和172.18.0.7

(2)挂载云盘

在选定的target使用fdisk格式化成ext4格式，挂载给target ECS使用，文件系统名自定义。

(3)获取安装包

在开源网站上可以下载到iscsitarget源码包

wgethttp://sourceforge.net/projects/iscsitarget/files/latest/download?source=directory

(4)安装target

```
tar zxvf iscsitarget-1.4.20.1.tar.gz
cd iscsitarget-1.4.20.1
make
make install
```

安装后在/etc下会生成iet目录，iscsitarget的主要配置文件就在该目录下。

(5)配置

修改/etc/iet/ietd.conf，涉及到改动的内容有：

```
Target iqn.2001-04.com.sharestorage:racdb.crs1
Lun 0 Path=/dev/vdb,Type=blockio

Target iqn.2001-04.com.sharestorage:racdb.data1
Lun 0 Path=/dev/vdc,Type=blockio
```

这里斜体部分请自定义，而粗体部分必须严格匹配系统中云盘的实际地址。

修改/etc/iet/targets.allow，定义允许访问target的地址范围。

```
ALL 172.18.0.0/20
```

(6)启动iscsi-target

/etc/init.d/iscsi-target start

(7)安装iscsi-initiator

两个RAC节点都需要安装iscsi-initiator。iscsi-initiator在centOS自带光盘iso镜像中，挂载好iso，配置好yum源，直接安装：

yum -y install iscsi-initiator-utils-*

(8)启动iscsid服务，并配置开机自动重启

service iscsid start

chkconfig iscsid on

chkconfig iscsi on

(9)发现target

在两个initiator节点执行：iscsiadm-mdiscovery-t sendtargets-p 172.18.0.5

(10)登录到iscsi-target

在两个initiator节点执行：

iscsiadm-mnode-Tiqn.2001-04.com.sharestor-age:racdb.data1-p 172.18.0.5 -l

iscsiadm-mnode-Tiqn.2001-04.com.sharestor-age:racdb.crs1-p 172.18.0.5 -l

(11)配置initiator节点重启后自动登录target

在两个initiator节点执行：

iscsiadm-mnode-Tiqn.2001-04.com.sharestor-age:racdb.crs1 -p 172.18.0.5 --op update -n node.startup -v automatic

iscsiadm-mnode-Tiqn.2001-04.com.sharestor-age:racdb.data1 -p 172.18.0.5 --op update -n node.start-up -v automatic

(12)检查云盘

在initiator节点执行fdisk-l，此时能看到新增了2个磁盘，这是由target共享过来的。

3. 安装配置n2n VPN

(1)所有节点都安装倚赖包

yum install subversion gcc-c++ openssl-devel

(2)所有节点都需要安装n2n

在线安装可以使用如下命令，也可以离线将二进制文件下载上传到本地进行安装。

svn co https://svn.ntop.org/svn/ntop/trunk/n2n

cd n2n/n2n_v2

make

make install

(3)超级节点启动服务

supernode -l 5000

(4)边缘节点配置IP

节点1：edge -d edge0 -a 192.168.0.11 -c mynetwork -k password -l 172.18.0.5:5000 -m AA:54:64:FC:46:25 -E -r

节点2：edge -d edge0 -a 192.168.0.12 -c mynetwork -k password -l 172.18.0.5:5000 -m 96:95:2C:96:48:01 -E -r

这样，节点1和节点2在192.168.0.0网段的public IP就创建成功了，使用ifconfig能看到一个名为edge0的新端口以及对应的IP。

4. 安装缺失的RPM包

Oracle RAC安装时会进行必要系统检查，其中包括了对一系列必需的RPM的检查，为了防止安装报错，这些RPM包必须要提前装好，而安装RPM最方便的方法是通过yum工具。

以下是配置yum工具的步骤：

(1)默认专有云/公有云环境下都会预先配置好yum，如果没有配置好，请自行配置好；

(2)install package

yum install -y binutils-2.*

yum install -y compat-libstdc++-33*

yum install -y sysstat-9.*

yum install -y glibc-2.*

yum install -y libgcc-4.*

yum install -y libstdc++-4.*

yum install -y elfutils-libelf-0* elfutils-li-belf-devel-0*

yum install -y libtool-ltdl*

yum install -y ncurses*

yum install -y readline*

yum install -y unixODBC*

yum install -y compat-libcap1*

yum install -y compat-libstdc++*

注：一些包可能会因为系统已经有了更高的版本而无法安装，忽略即可，一些包在阿里云提供的yum源中不存在，也可先忽略。

5. 修改系统核心参数

(1)将如下内容加入到”/etc/sysctl.conf” 文件中。

fs.aio-max-nr = 1048576

fs.file-max = 6815744

kernel.shmmni = 4096

kernel.sem = 250 32000 100 128

net.ipv4.ip_local_port_range = 9000 65500

net.core.rmem_default=262144

net.core.rmem_max=4194304

net.core.wmem_default=262144

net.core.wmem_max=1048586

运行如下命令使修改即刻生效

#/sbin/sysctl -p

(2)修改”/etc/security/limits.d/90-nproc.conf”

将如下行：

*　　soft　nproc　1024

改为：

* - nproc 16384

6. 创建用户和组

(1)运行如下命令创建用户和组：

创建oinstall组

#groupadd oinstall

创建dba组

#groupadd dba

创建grid用户

#groupadd asmdba

#groupadd asmadmin

#groupadd asmoper

#useradd -g oinstall -G dba grid

设置grid用户密码

创建oracle用户

#useradd -g oinstall -G dba oracle

设置grid和oracle用户密码

#passwd grid

#passwd oracle

#usermod -G dba,asmdba,asmadmin,asmoper grid

#usermod -G dba,asmdba oracle

(2)调整oracle用户的shell限制

-Add the following lines to the “/etc/security/limits.conf” file.

oracle　　soft　nproc　2047

oracle　　hard　nproc　16384

oracle　　soft　nofile　4096

oracle　　hard　nofile　65536

oracle　　soft　stack　10240

grid　　soft　nproc　2047

grid　　hard　nproc　16384

grid　　soft　nofile　4096

grid　　hard　nofile　65536

--Add the following lines to the “/etc/pam.d/login” file, if it does not already exist.

session　required　pam_limits.so

将Oracle grid的安装介质上传至服务器上，并解压缩，安装其中的cvuqdisk包。

--Install the following package from the Oracle grid media after you’ve defined groups.

#cd $your_path_to_grid/rpm

#rpm -Uvh cvuqdisk*

7. 网络设置

(1)编辑/etc/hosts文件，将各节点的public IP, Private IP, Virtual IP, Scan IP填入其中：

#Public

192.168.0.11 testOracle1Z.com　testOracle1Z

192.168.0.12 testOracle3Z.com　testOracle3Z

#Private

172.18.0.6　testOracle1Z-PRI.com　testOracle1Z-PRI

172.18.0.7　testOracle3Z-PRI.com　testOracle3Z-PRI

#Virtual

192.168.0.21 testOracle1Z-VIP.com testOracle1Z-VIP

192.168.0.22 testOracle3Z-VIP.com testOracle3Z-VIP

#SCAN

192.168.0.31 ORASCAN.com　ORASCAN

(2)修改 /etc/resolv.conf,使nameserver指向本机

nameserver localhost

(3)重启dnsmasq服务，并使其随系统启动

可能要安装一下dnsmasq服务先

#/etc/init.d/dnsmasq restart

#chkconfig dnsmasq on

8. 关闭SELINUX以及iptables

编辑 “/etc/selinux/config”， 将其改为：

SELINUX=disabled

关闭iptables

service iptables stop

service ip6tables stop

chkconfig iptables off

chkconfig ip6tables off

9. 关闭NTP服务

暂时不关看看情况

Either configure NTP, or make sure it is not configured so the Oracle Cluster Time Synchronization Service (ctssd) can synchronize the times of the RAC nodes. If you want to deconfigure NTP do the following.

service ntpd stop

Shutting down ntpd: [OK]

chkconfig ntpd off

mv /etc/ntp.conf /etc/ntp.conf.orig

rm /var/run/ntpd.pid

10. 创建Oracle软件的安装目录

Create the directories in which the Oracle software will be installed.

#mkdir -p /u01/grid

#mkdir -p /u01/oracle/11.2.0

#chown -R grid:oinstall /u01/grid

#chmod -R 775 /u01/grid

#chown -R oracle:oinstall /u01/oracle

#chmod -R 775 /u01/oracle

#mkdir -p /u01/crs/11.2.0

#chown -R grid:oinstall /u01/crs

#chmod -R 775 /u01/crs

11. 配置用户的环境变量

修改grid的配置文件：/home/grid/.bash_profile

ORACLE_HOSTNAME=testOracle1Z.com; export ORACLE_HOSTNAME

ORACLE_BASE=/u01/grid; export ORACLE_BASE

ORACLE_HOME=/u01/crs/11.2.0; export ORACLE_HOME

ORACLE_SID=+ASM1; export ORACLE_SID

PATH=$ORACLE_HOME/bin:$PATH; export PATH

LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib; export LD_LIBRARY_PATH

CLASSPATH=$ORACLE_HOME/JRE:$ORACLE_HOME/jlib:$ORACLE_HOME/rdbms/jlib;

export CLASSPATH

修改oracle的配置文件：/home/oracle/.bash_profile

ORACLE_HOSTNAME=testOracle1Z.com; export ORACLE_HOSTNAME

ORACLE_BASE=/u01/oracle; export ORACLE_BASE

ORACLE_HOME=/u01/oracle/11.2.0; export ORACLE_HOME

ORACLE_SID=orarac1; export ORACLE_SID

PATH=$ORACLE_HOME/bin:$PATH; export PATH

LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib; export LD_LIBRARY_PATH

CLASSPATH=$ORACLE_HOME/JRE:$ORACLE_HOME/jlib:$ORACLE_HOME/rdbms/jlib;

export CLASSPATH

以上是节点1的配置，若在节点2上面配置，请将对应的ORACLE_SID和HOSTNAME改为对应的值。

12. 启用心跳网卡eth0的ARP协议

2个节点的内网网卡eth0作为RAC内部通讯的网卡，其ARP协议必须开通，以下是网卡配置文件示例，注意其中的“ARP=yes”和OPTIONS=”layer2=1”两项设置。

more /etc/sysconfig/network-scripts/ifcfg-eth0

DEVICE=eth0

ONBOOT=yes

BOOTPROTO=static

IPADDR=172.18.0.6

NETMASK=255.255.224.0

ARP=yes

OPTIONS=”layer2=1”

13. 配置共享磁盘

所有共享磁盘都在此配置别名、属主、属组以及权限。

vi/etc/udev/rules.d/12-asm-permissions.rules文件，加入如下内容：

KERNEL==”vdc”,NAME=”crs01”,OWNER:=”grid”,GROUP:=”oinstall”,MODE:=”660”

KERNEL==”vdd”, NAME=”crsdata01”,OWNER:=”grid”,GROUP:=”oinstall”,MODE:=”660”

两个节点都改完之后重启系统

#shutdown -r now

14. 建立节点间的信任关系

(1)两节点分别执行（直接回车确认）

su – grid

mkdir ~/.ssh

ssh-keygen -t rsa

ssh-keygen -t dsa

su – oracle

mkdir ~/.ssh

ssh-keygen -t rsa

ssh-keygen -t dsa

(2)主节点执行

su – grid

cat ~/.ssh/id_rsa.pub>>./.ssh/authorized_keys --公钥存在authorized_keys文件中,写到本机

cat ~/.ssh/id_dsa.pub>>./.ssh/authorized_keys

ssh testOracle3Z-PRI cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys --第二个节点的公钥写到本机

ssh testOracle3Z-PRI cat ~/.ssh/id_dsa.pub >> ~/.ssh/authorized_keys

scp ~/.ssh/authorized_keys testOracle3Z-PRI:~/.ssh/authorized_keys

su – oracle

cat ~/.ssh/id_rsa.pub>>./.ssh/authorized_keys --公钥存在authorized_keys文件中,写到本机

cat ~/.ssh/id_dsa.pub>>./.ssh/authorized_keys

ssh testOracle3Z-PRI cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys --第二个节点的公钥写到本机

ssh testOracle3Z-PRI cat ~/.ssh/id_dsa.pub >> ~/.ssh/authorized_keys

scp ~/.ssh/authorized_keys testOracle3Z-PRI:~/.ssh/authorized_keys

(3)两个节点分别验证

ssh testOracle1Z date (public网卡)

ssh testOracle3Z date

ssh testOracle1Z-PRI date (private网卡)

ssh testOracle3Z-PRI date

四、安装Oracle GRID软件

1. 检查grid CRS组件的安装先决条件是否满足

在节点1上执行如下命令检查系统是否符合grid的安装先决条件，检查结果应该都是pass。

#su – grid

$cd grid的软件目录

$./runcluvfy.sh stage -pre crsinst -n testOracle1Z,-testOracle3Z -verbose

报这个错：

/bin/rm:cannot remove`/tmp/bootstrap’:Operation not permitted

./runcluvfy.sh: line 99:/tmp/bootstrap/ouibootstrap.log: Permission denied

将/tmp/bootstrap的权限改成777，属主属组改成grid:oinstall

会报NTP的错误，可以忽略，其他的报错，请按照提示进行修复。

2. 配置

编辑crs_install_test.rsp，文件目录：/u01/soft/oracle/grid/response/。两边同时编辑，以下只收录要修改或重视的内容，以节点1为例：

ORACLE_HOSTNAME=testOracle1Z.com

INVENTORY_LOCATION=/u01/grid/oraInventory

oracle.install.option=CRS_CONFIG

ORACLE_BASE=/u01/grid

ORACLE_HOME=/u01/crs/11.2.0

oracle.install.asm.OSDBA=asmdba

oracle.install.asm.OSOPER=asmoper

oracle.install.asm.OSASM=asmadmin

oracle.install.crs.config.gpnp.scanPort=1521

oracle.install.crs.config.clusterName=rac-test

oracle.install.crs.config.clusterNodes=testOracle1Z:-testOracle1Z-VIP,testOracle3Z:testOracle3Z-VIP

oracle.install.crs.config.privateInterconnects=eth0:192.168.0.0:2,eth1:172.18.0.0:1 //1代表public，2代表private，3代表在群集中不使用该网卡

oracle.install.crs.config.storageOption=ASM_STORAGE

oracle.install.asm.SYSASMPassword=system

oracle.install.asm.diskGroup.name=OCR

oracle.install.asm.diskGroup.redundancy=EXTERNAL

oracle.install.asm.diskGroup.disks=/dev/crs01

oracle.install.asm.diskGroup.name=OCR

oracle.install.asm.diskGroup.redundancy=EXTERNAL

oracle.install.asm.diskGroup.diskDiscoveryString=/

dev/crs*

oracle.install.asm.monitorPassword=system

3. 运行runInstaller

在节点1的CRS软件目录执行：

$./runcluvfy.sh stage -pre crsinst -n testOracle1Z,testOracle3Z -verbos

./runInstaller -silent -responseFile /u01/soft/oracle/grid/response/crs_install_test.rsp -ignoreSysPrereqs -ignorePrereq

等待一段时间后，会提示在两个节点分别用root用户执行：

/u01/grid/oraInventory/orainstRoot.sh

/u01/crs/11.2.0/root.sh

请注意以下内容，这是Oracle的安装bug导致需要进行的特殊workaroud：

在执行root.sh脚本时出现Adding daemon to inittab的时候，使用root立即执行命令：

/bin/ddif=/var/tmp/.oracle/npohasdof=/dev/nullbs=1024 count=1

vi /etc/init/oracle-ohasd.conf

Oracle OHASD startup

start on runlevel [35]

stop on runlevel [!35]

respawn

exec /etc/init.d/init.ohasd run >/dev/null 2>&1 </dev/null

4. 增加磁盘组

asmca -silent -createDiskGroup -sysAsmPassword system-diskString '/dev/crs*' -diskGroupName DATA-diskList '/dev/crsdata01' -redundancy EXTERNAL -compatible.asm 11.2 -compatible.rdbms 11.2

asmca -silent -addDisk -sysAsmPassword system -diskGroupName DATA -diskList '/dev/crsdata01'

vi cfgrsp.properties

```
[grid@testOracle1Z cfgtoollogs]$ more cfgrsp.properties
oracle.assistants.asm|S_ASMPASSWORD=system
oracle.assistants.asm|S_ASMMONITORPASSWORD=system
```

chmod 660 cfgrsp.properties

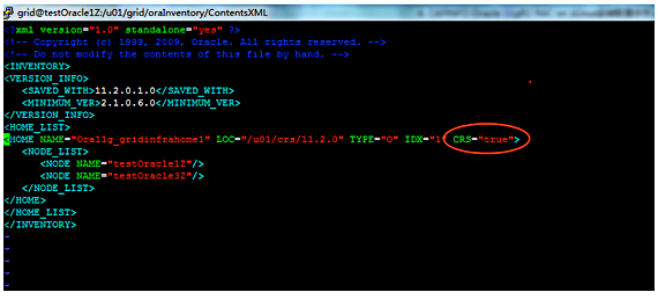
./configToolAllCommands RESPONSE_FILE=./cfgrsp.properties

五、安装Oracle DB软件

在两个数据库节点上都执行：

vi/u01/grid/oraInventory/ContentsXML/inventory.xml

加入



准备好responsefile，位置：/u01/soft/oracle/database/response/db_install_test.rsp。以节点1为例，该文件需要修改和注意的内容有：

oracle.install.option=INSTALL_DB_SWONLY

ORACLE_HOSTNAME=testOracle1Z.com

UNIX_GROUP_NAME=oinstall

INVENTORY_LOCATION=/u01/oracle/oraInventory

SELECTED_LANGUAGES=en

ORACLE_HOME=/u01/oracle/11.2.0

ORACLE_BASE=/u01/oracle

oracle.install.db.InstallEdition=EE

oracle.install.db.isCustomInstall=false

oracle.install.db.customComponents=oracle.server:11.2.0.1.0,oracle.sysman.ccr:10.2.7.0.0,oracle.xdk:11.2.0.1.0,oracle.rdbms.oci:11.2.0.1.0,oracle.network:11.2.0.1.0,oracle.network.listener:11.2.0.1.0,oracle.rdbms:11.2.0.1.0,oracle.options:11.2.0.1.0,oracle.rdbms.partitioning:11.2.0.1.0,oracle.oraolap:11.2.0.1.0

oracle.install.db.DBA_GROUP=dba

oracle.install.db.OPER_GROUP=oinstall

oracle.install.db.CLUSTER_NODES=testoracle1z,testoracle3z

DECLINE_SECURITY_UPDATES=true

依次执行：

%/u01/crs/11.2.0/bin/cluvfy stage-pre dbinst-n testOracle1Z,testOracle3Z -verbose //grid用户

%./runInstaller -silent -ignoreSysPrereqs -ignorePrereq -responseFile/u01/soft/oracle/database/re-

sponse/db_install_test.rsp //grid用户

安装完毕，按照提示新开终端，切换成root用户，依次执行：

#/u01/grid/oraInventory/orainstRoot.sh//root用户

#/u01/oracle/11.2.0/root.sh//root用户

六、创建数据库

准备好response文件，文件位置：/u01/soft/oracle/database/response/dbca_test.rsp。

以节点1为例，该文件需要修改和注意的内容有：

GDBNAME = “orarac”

SID = “orarac”

NODELIST=testoracle1z,testoracle3z

SYSPASSWORD = “system”

SYSTEMPASSWORD = “system”

STORAGETYPE=ASM

DISKGROUPNAME=DATA

RECOVERYGROUPNAME=DATA

CHARACTERSET = “ZHS16GBK”

在节点1执行：

dbca-silent-responseFile /u01/soft/oracle/database/response/dbca_test.rsp

```
[oracle@testOracle1Z response]$ dbca -silent -responseFile /u01/soft/oracle/database/response/dbca_test.rsp
Copying database files
1% complete
3% complete
9% complete
15% complete
21% complete
27% complete
30% complete
Creating and starting Oracle instance
32% complete
36% complete
40% complete
44% complete
45% complete
48% complete
50% complete
Creating cluster database views
52% complete
70% complete
Completing Database Creation
73% complete
76% complete
85% complete
94% complete
100% complete
Look at the log file "/u01/oracle/cfgtoollogs/dbca/orarac/orarac.log" for further details.
[oracle@testOracle1Z response]$
```

七、IO校准测试

以下三个截图是三次IO校准测试的结果，IOPS（随机IO,每次1024KB）最大值大约在3390，吞吐量大约在900MBps。

```
NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
memory_target                        big integer 3152M
SQL> set timing on serveroutput on
SQL> declare
v_max_iops BINARY_INTEGER;
  2    3  v_max_mbps BINARY_INTEGER;
  4  v_act_lat BINARY_INTEGER;
begin
  5    6  dbms_resource_manager.CALIBRATE_IO(1,20,v_max_iops,v_max_mbps,v_act_lat);
  7  dbms_output.put_line('max iops : ' || v_max_iops );
  8  dbms_output.put_line('max mbps : ' || v_max_mbps );
  9  dbms_output.put_line('actual latency : ' || v_act_lat );
 10  end;
 11  /


max iops : 3390
max mbps : 857
actual latency : 15

PL/SQL procedure successfully completed.
```

```
Elapsed: 00:12:33.38
SQL> SQL> SQL> declare
v_max_iops BINARY_INTEGER;
  2    3  v_max_mbps BINARY_INTEGER;
  4  v_act_lat BINARY_INTEGER;
  5  begin
  6  dbms_resource_manager.CALIBRATE_IO(5,100,v_max_iops,v_max_mbps,v_act_lat);
  7  dbms_output.put_line('max iops : ' || v_max_iops );
  8  dbms_output.put_line('max mbps : ' || v_max_mbps );
  9  dbms_output.put_line('actual latency : ' || v_act_lat );
 10  end;
/ 11
max iops : 3386
max mbps : 944
actual latency : 1

PL/SQL procedure successfully completed.

Elapsed: 00:12:32.18
```

```
SQL> declare
v_max_iops BINARY_INTEGER;
  2    3  v_max_mbps BINARY_INTEGER;
  4  v_act_lat BINARY_INTEGER;
  5  begin
  6  dbms_resource_manager.CALIBRATE_IO(5,10,v_max_iops,v_max_mbps,v_act_lat)
  7  dbms_output.put_line('max iops : ' || v_max_iops );
  8  dbms_output.put_line('max mbps : ' || v_max_mbps );
  9  dbms_output.put_line('actual latency : ' || v_act_lat );
end;
 10   11  /
max iops : 3395
max mbps : 941
actual latency : 11

PL/SQL procedure successfully completed.

Elapsed: 00:15:39.75
```

Azure SQL数据库迁移阿里云RDS SQLserver实践

正博
阿里云技术专家

一、适用场景

本方案适用于采用SQL server官方工具SSMS从AZURE RDS迁移SQL SERVER到阿里云RDS。

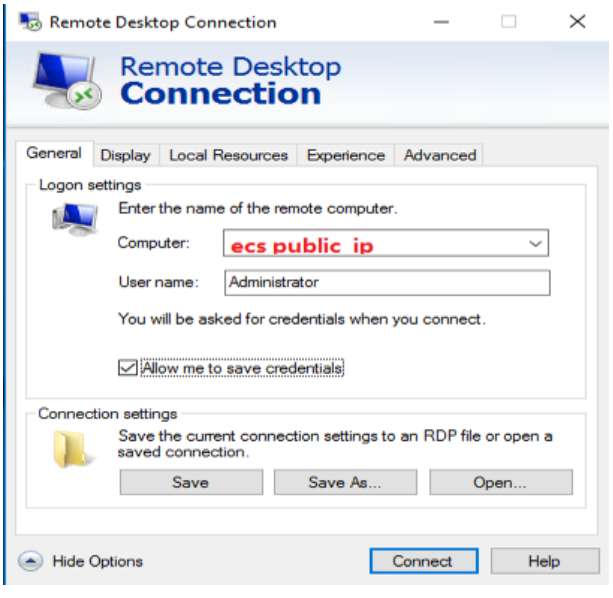
二、前提条件

1.在阿里云平台已创建目标实例和数据库，已配置迁移账号。
2.已购买云服务器ECS,操作系统为windows server 2016 (英文版),有公网IP。
3.在购买的ECS上安装SQL server的客户端工具（SSMS）。
(1)SSMS微软下载地址：https://msdn.microsoft.com/library/mt238290.aspx
(2)全程默认安装即可，如有不确定请参考微软官网。

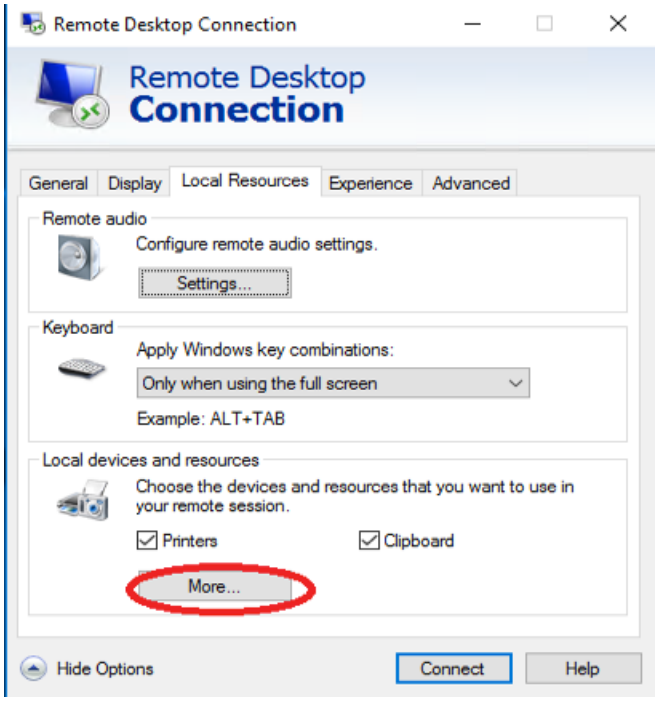
三、迁移操作步骤

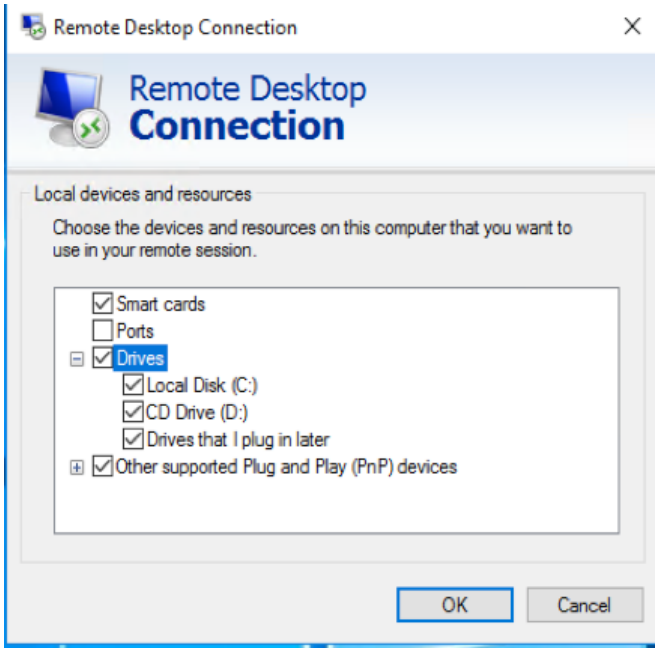
1.远程登陆ECS

(1)打开 开始菜单 > 远程桌面连接，或在 开始菜单 > 搜索 中输入 mstsc。也可以使用快捷键 Win+R 来启动运行窗口，输入mstsc后回车启动远程桌面连接。
(2)在远程桌面连接 对话框中，输入实例的公网 IP 地址。单击 显示选项。
(3)输入用户名，默认为 Administror。然后单击 allow me to save credentials，这样以后登录就不需要手动输入密码。



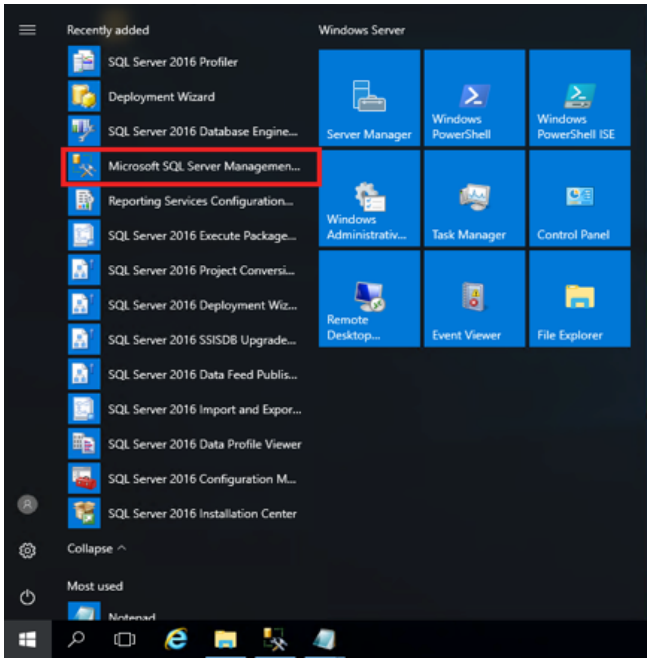
(4)为了方便将本地文件拷贝到实例中，您还可以设置通过远程桌面共享本地电脑资源。单击local resources选项卡中进行设置，一般选择clipboard。但剪贴板只能从本地直接复制文字信息到实例，不能复制文件。如果需要复制文件，需点击more，选择驱动器，然后选择文件存放的盘符。



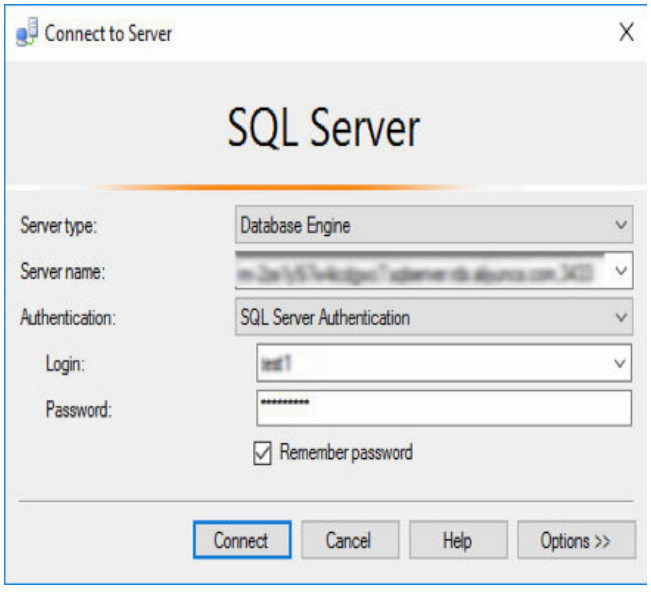


(5)单击ok，然后单击connect，您现在成功连接到实例，可以进行其它操作。

2.启动SSMS

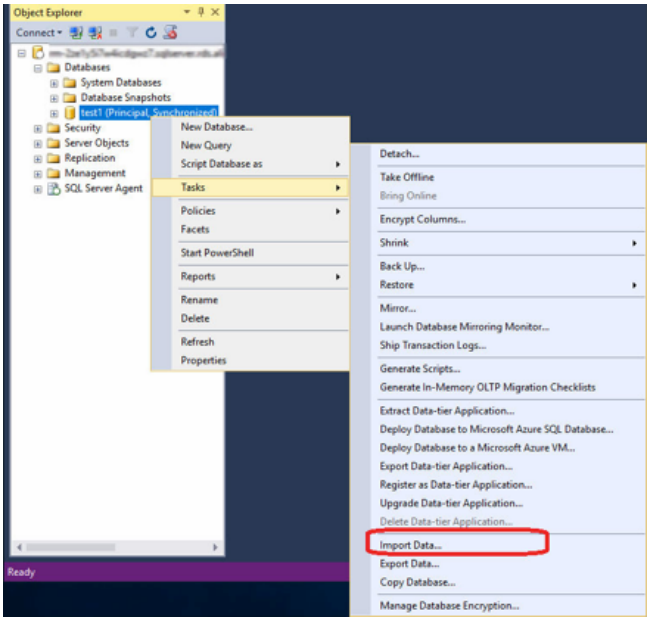


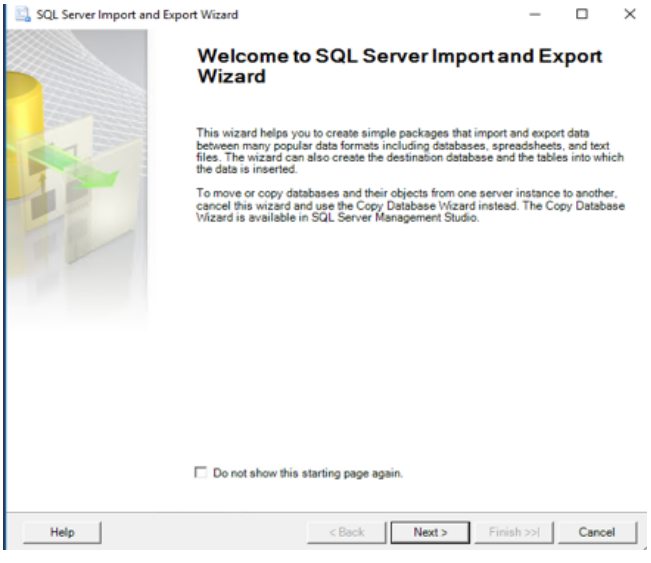
(1)连接到新建目标数据库



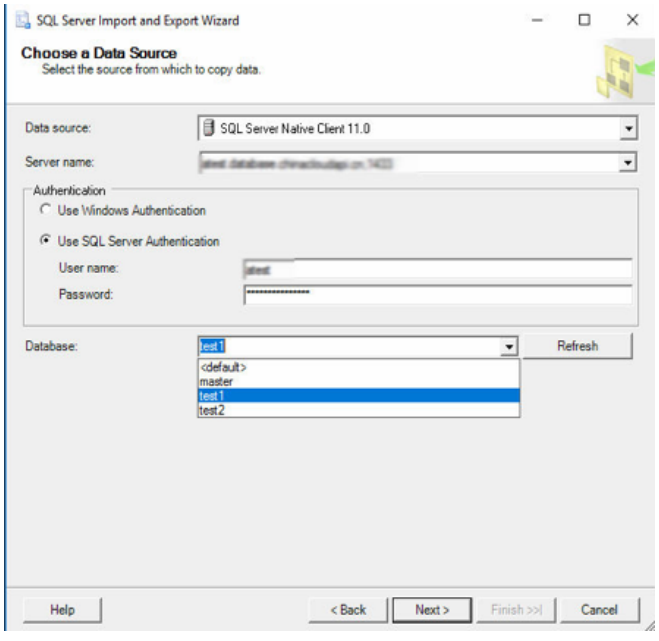
(2)开始配置导入任务

a.选择test1数据库->tasks->import data(导入数据)

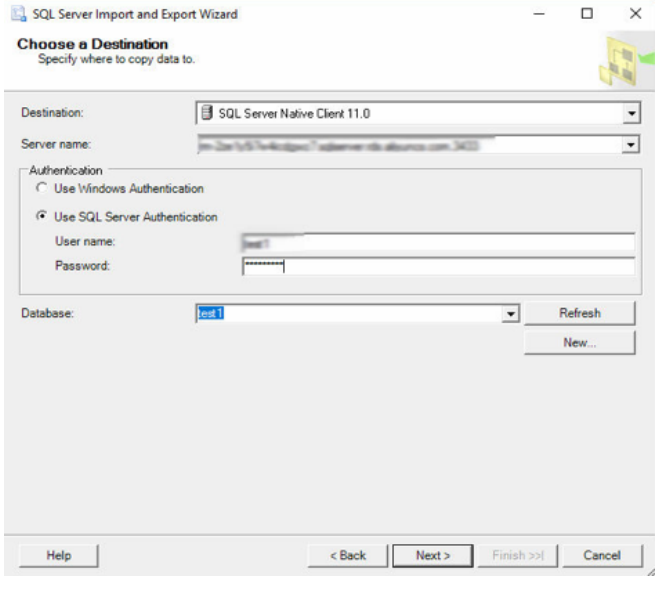




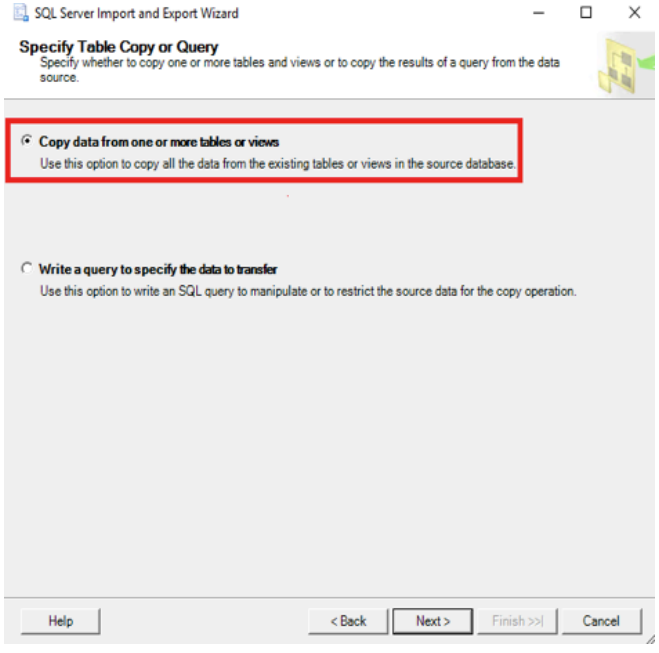
b.填写数据源连接信息



c.填写目标数据库连接信息

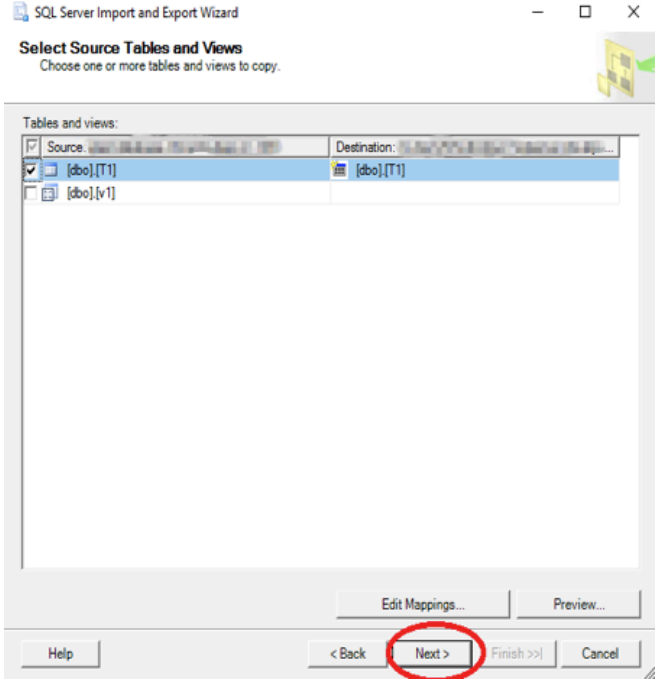


d.选择要复制的数据



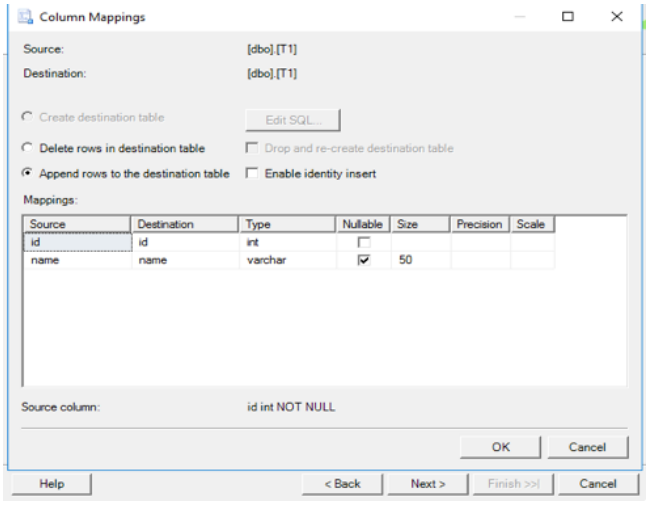
e.编辑表的对应关系

选择表的对应关系，也可以编辑字段的对应关系，可以预览迁移后的数据，一般都是同名对应，如果目标创建的表结构，自动同名对应也可以自定义对应关系，如果目标没有创建表结构，这里可以自动生成与原表结构一致。

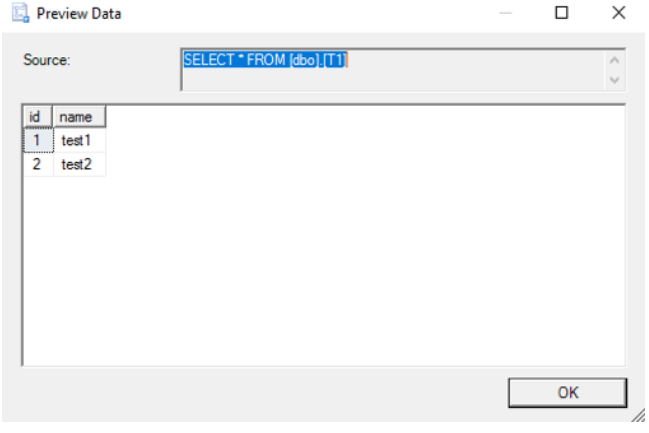


f.编辑列的对应关系(点击edit mappings)

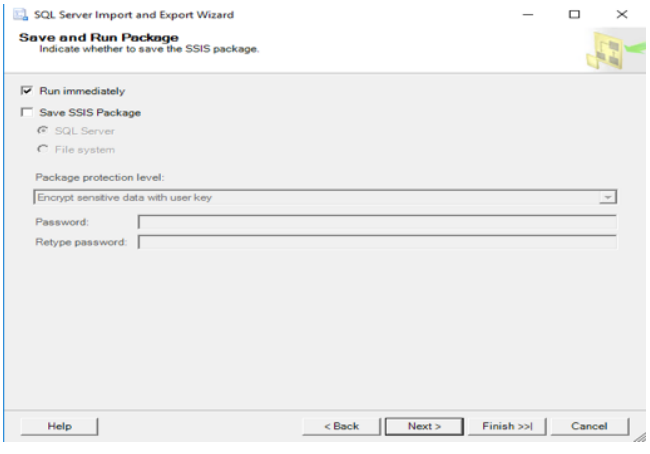
列之间的映射关系可以自定义，默认同名对应。同时可以选择追加，还是删除目标表数据，重新导入。其中插入标示如果选中，代表源表中的标示是自动增长的，源表是什么值就插入目标表是什么值，如果不选中，默认重新生成标示。

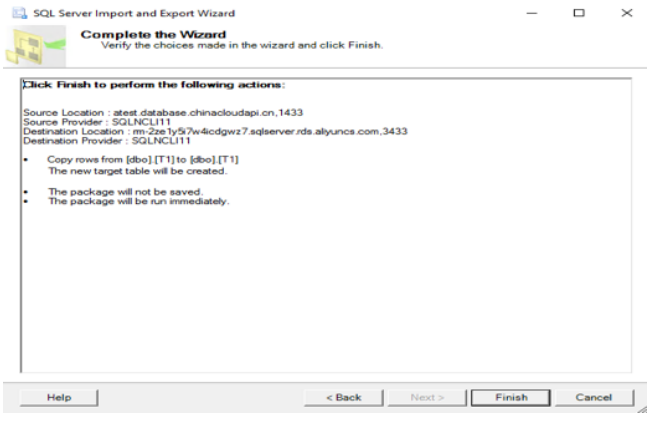


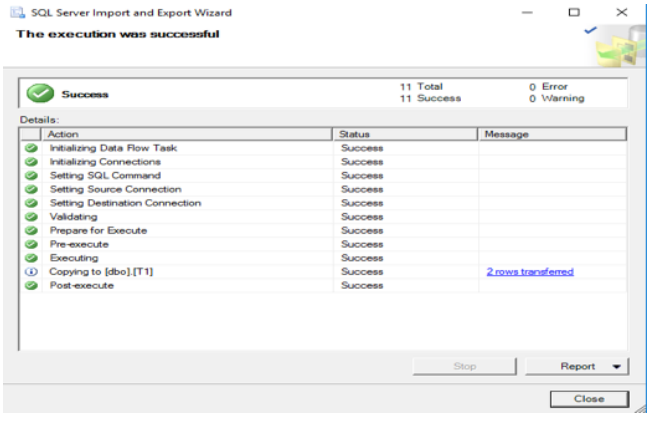
preview预览迁移的数据如图，如果没有问题开始正式迁移



g.执行选项





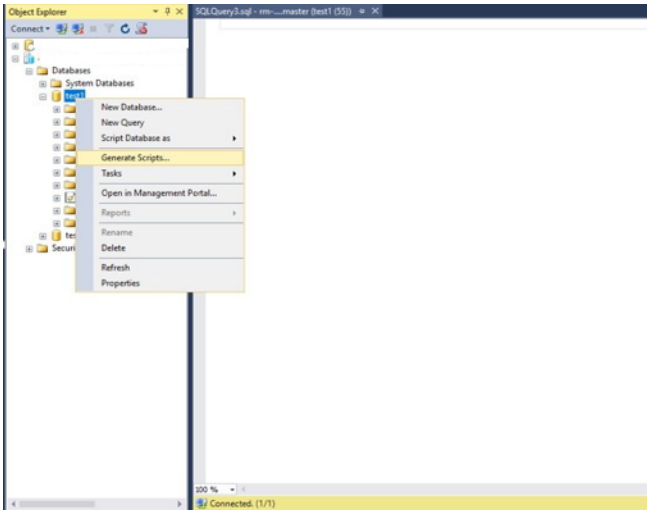


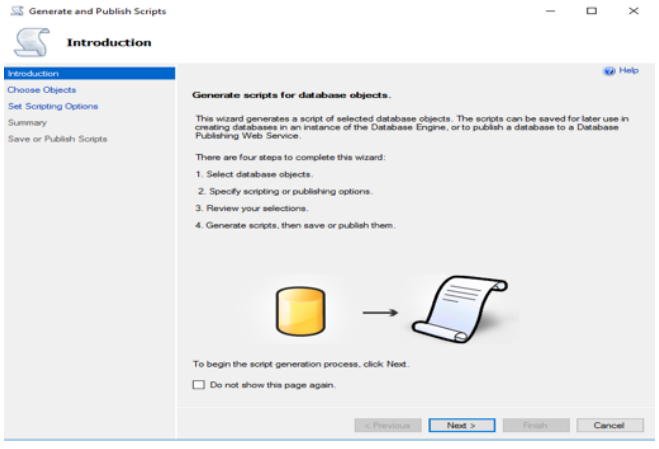
如果正常到这里就完成了数据的迁移，如果有问题，在消息列会提供报错信息，根据报错信息进行调整即可。

3.生成迁移对象脚本

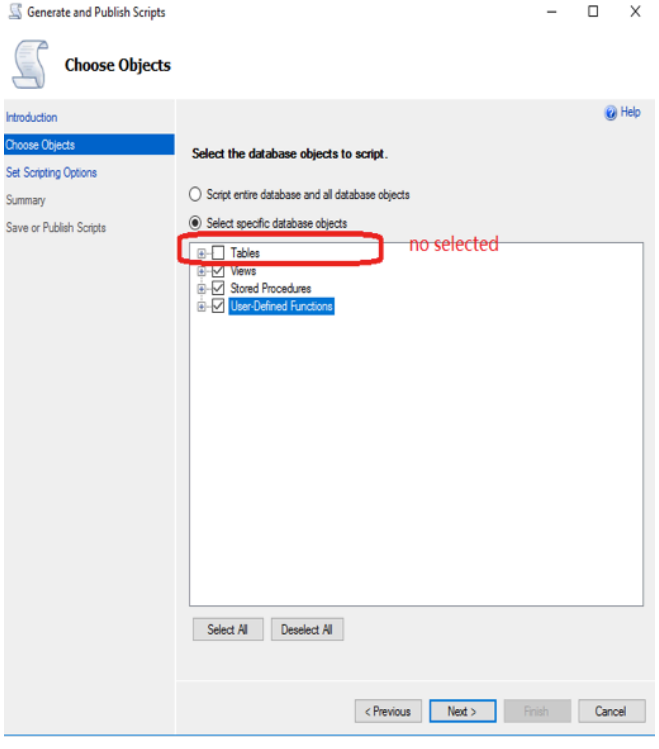
数据迁移后，还有一些脚本性代码没有迁移，比如存储过程、试图、自定义函数。迁移方法是连接到源库，生成存储过程、试图、自定义函数的脚本，拿到目标库执行，创建这些对象。另外执行计划，需要登录到目标库重新创建即可。

(1)生成迁移对象脚本 在数据源库操作

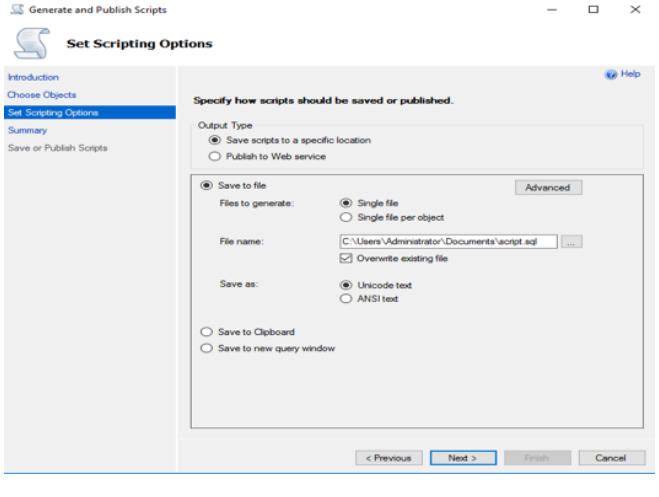


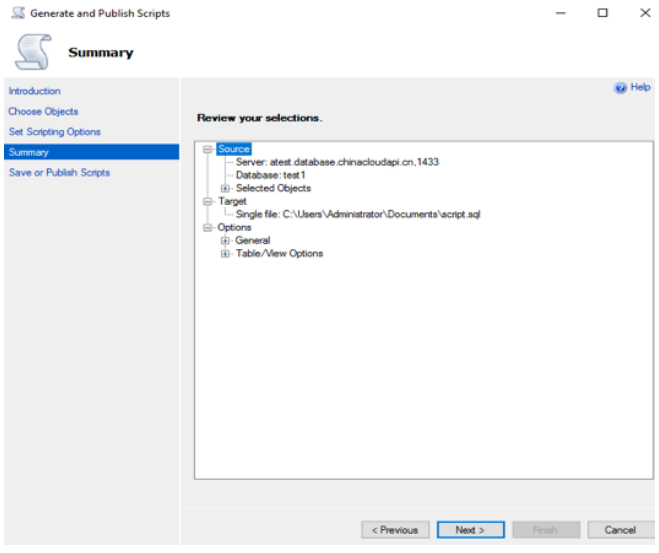


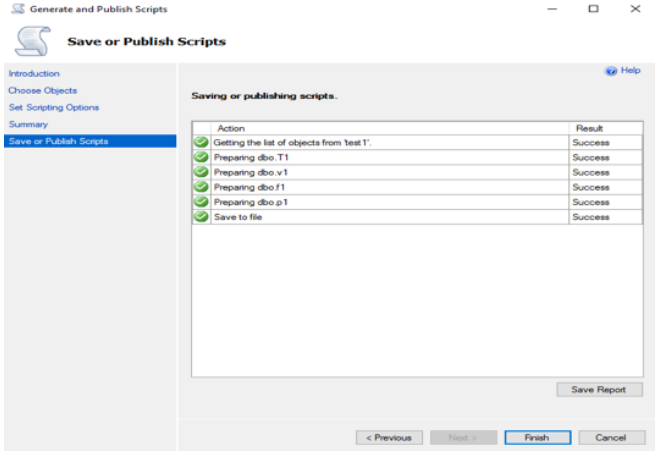
Next→选择生成的对象，tables 不用选！



Next→选择存放位置和格式





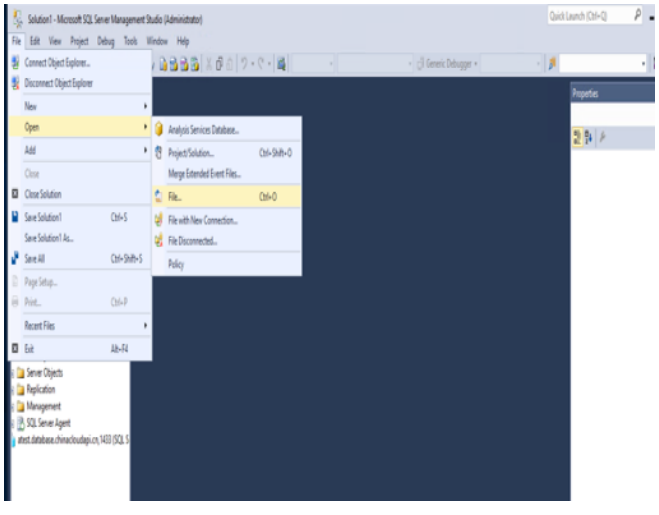


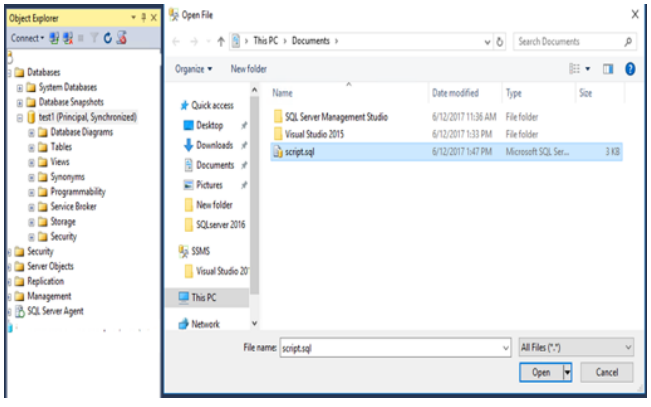
点击finish,完成脚本创建。

把生成的脚本，拿到目标库执行，注意脚本不能有操作系统表的行为，否则会超出RDS给提供的用户权限。如果存在建议应用实现这样的功能。

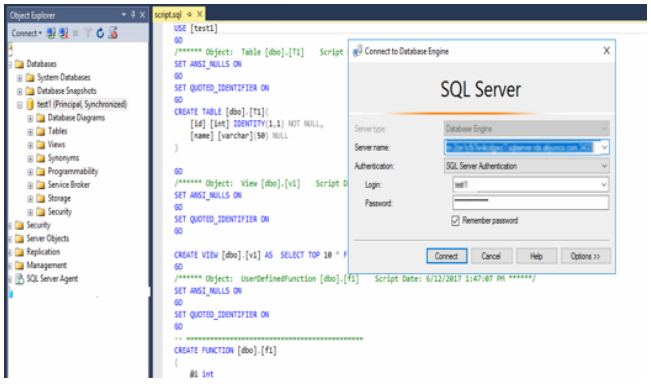
(2)脚本在目标库执行

a.选择脚本存放路径 file->open->file

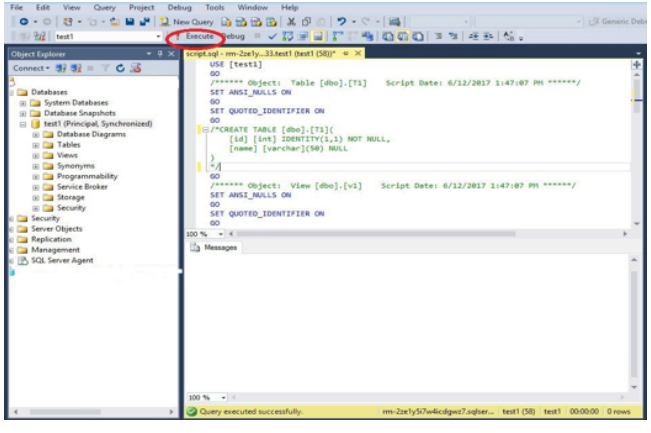




b.连接目标数据库 点击connect,连接到数据库test1。

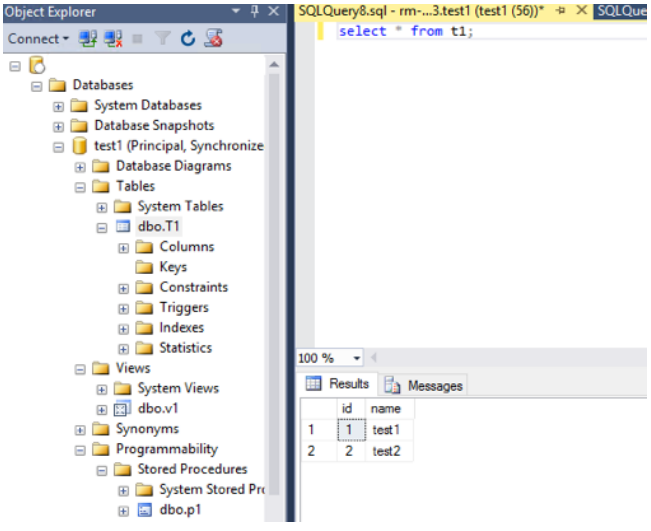


c.执行脚本提示query executed successfully,完成脚本执行。

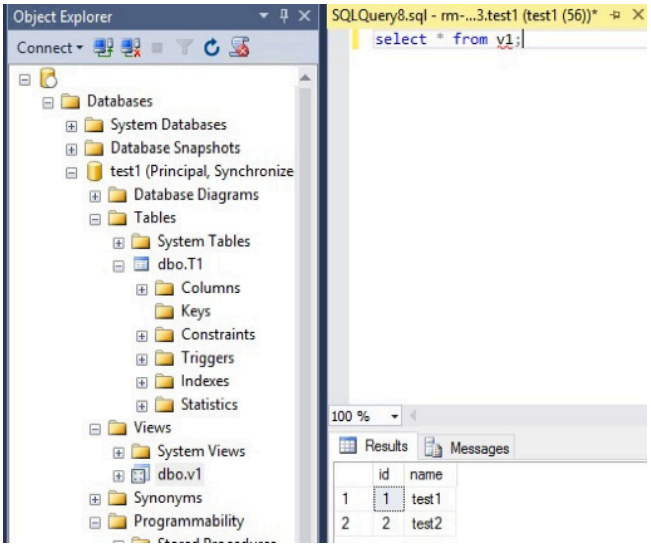


4、数据校验

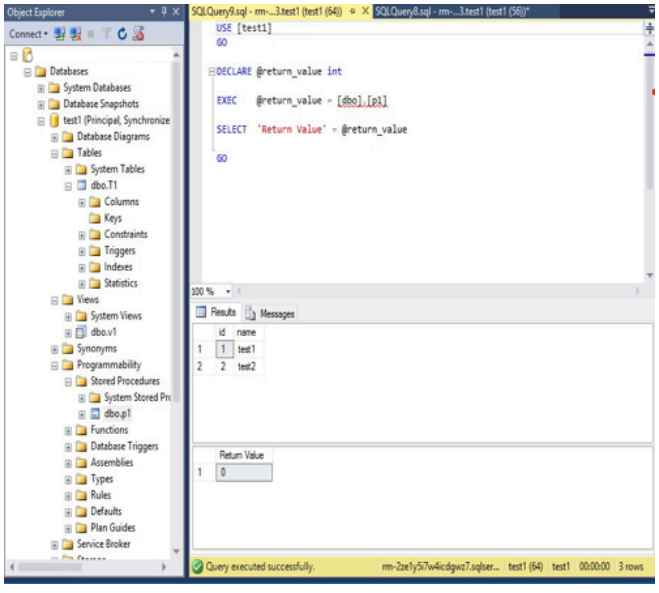
(1)Test1库数据校验



(2)视图



(3)存储过程



四、注意事项

1.本方案只适用于全量迁移，不适用于增量迁移。

2.迁移过程中先完成数据迁移，再进行其他对象迁移如视图、存储过程、函数。

3.停机时间取决于数据量的大小，如果停机时间较长，业务不可接受，可以考虑采用这种方式分批迁移数据，需要业务的配合。

4.为了确保数据一致性，请停止数据源端数据库的写入操作。

5.如果数据库里有数据，请做好备份，目标数据库中的数据将被覆盖。

物理专线流量切换方案

墨华
阿里云技术专家

一、物理专线流量切换方案

在从传统IDC向云上迁移过程中，物理专线作为连接云上和云下的桥梁，在混合云架构中占有绝对重要的地位。作为基础设施，在伴随业务不断发展的过程当中，也会进行相应的更换升级，在承担了生产环境流量的线路上进行操作需要十分小心，本文将结合阿里云VPC及路由相关知识，介绍如何平滑的进行线路流量切换。

二、业务场景

伴随着业务发展，对于基础资源的需求不断增大，比如专线所能够提供的带宽；另一方面，对于专线的高可用需求也会逐渐凸显，比如多线容灾，多点接入都是需要考虑的方案，那么在这些方案实施的过程中，就涉及到对原有线路的升级改造，如更换线路，线路升级等，下面就可能场景进行介绍。

在业务发展初期，对基础资源需求较低，成本方面也会比较节约，这种情况下，基于阿里云VPC的混合云架构，在专线带宽选择上一般都千兆以内，物理线路上也是以电缆为主，能够支持百兆到千兆带宽，前期在业务上使用时能够满足需求的。在中期，一方面是业务有更大的需求，另一方面，生产业务之外的一些需求也逐渐增加，像云上环境和IDC的数据同步和备份，日志的传输分析等都是非常消耗带宽，这个时候我们就需要对原有的专线进行升级，一般情况物理线路是推荐升级到光纤，以便更好的支持千兆到万兆的网络。那在这个升级的过程中，线路流量的倒换就要做到快速，低风险，减少对业务的影响。

另外一个比较典型的业务场景是对现有的线路进行可用性提升，从原来的单线连接升级到双线、三线甚至四线负载均衡，目前阿里云VPC对多线环境是支持的，通过ECMP协议，能够将流量HASH到不同线路，某条线路故障时，通过自动检测和自动切换机制，剔除掉故障线路保障业务稳定。那在这个需求背景下，对于VPC侧的路由改造难以避免，多线的聚合网络接口是一次生成，如果需要增加线路，就需要创建新的网络接口，更换网络接口则涉及到流量的切换。

二、方案原理

在进行具体的操作之前，先对方案的原理进行一些介绍。路由器对流量进行转发是依赖路由规则的匹配，路由表中采用最长前缀匹配作为流量的路由选路规则。最长前缀匹配是指IP网络中当路由表中有多条条目可以匹配目的IP时，采用掩码最长(最精确)的一条路由作为匹配项并确定下一跳。

例如某专有网络(下面称VPC)中路由表中路由条目如下表。

目标地址段	下一跳地址	路由接口
10.23.8.0/16	10.23.1.1	ri-1
192.168.0.0/16	192.168.0.1	ri-2
192.168.1.0/24	192.168.1.1	ri-3

其中某台ECS要访问192.1681.1.3，那么VPC的路由的匹配过程有下面几个步骤：

1. 第一条路由网络号不匹配，跳过；
2. 第二条路由网络号匹配，但掩码不是最长的；
3. 第三条路由网络号匹配，掩码长长度最长，命中。

那么这个数据包的下一跳地址将会是192.168.1.1。有了这个规律之后，那么我们可以在路由上做一些特殊的设置，来达到流量调整的目的，比如下面这种路由条目。

目标地址段	下一跳地址	路由接口
10.23.8.0/16	10.23.1.1	ri-1
192.168.0.0/16	192.168.1.1	ri-2
192.168.1.0/24	192.168.1.1	ri-3

假如访问的目的地址依旧是192.1681.1.3，那么最终还是匹配到第三条规格，如果此时我们将第三条规则删除掉，那么会匹配到第二条规则，但是这两条规则的下一跳地址相同，所以删除第三条路由规则并不会影响网络访问，但是网络端口从ri-3切换到了ri-2，本方案正是利用这个特性来完成。

基本步骤

1.新专线完成接入，包括线路施工，互联地址、VBR路由配置；

2.健康检查配置，如果有涉及双线容灾，需要配置好健康检；

3. 梳理专线涉及的VPC以及路由条目；

4. 流量切换操作实施；

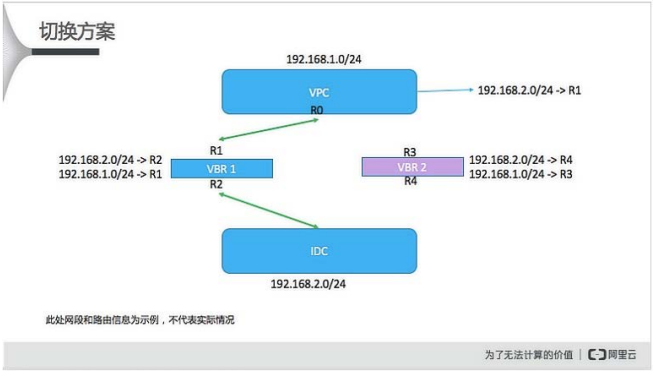
5. 业务验证及网络质量观察。

三、实施方案

下面我们通过文字描述和图示来对整个切换方案进行说明。

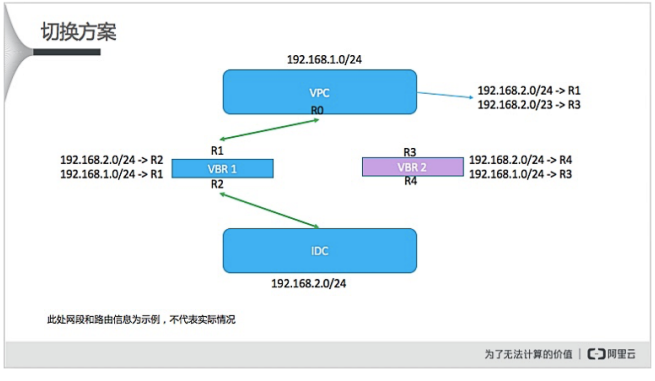
1.业务场景

下面的架构图展示了基础的专线网络架构，VBR1为老专线对应的VBR，VBR2为新专线对应的VBR，在VPC和VBR上均有对应网段的路由条目，VPC以192.168.1.0/24、IDC以192.168.2.0/24为例，实际生产环境会有多个子网，但原理保持一致。

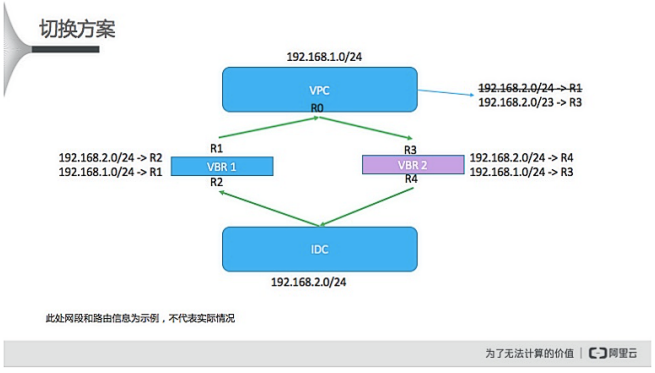


2.切换步骤

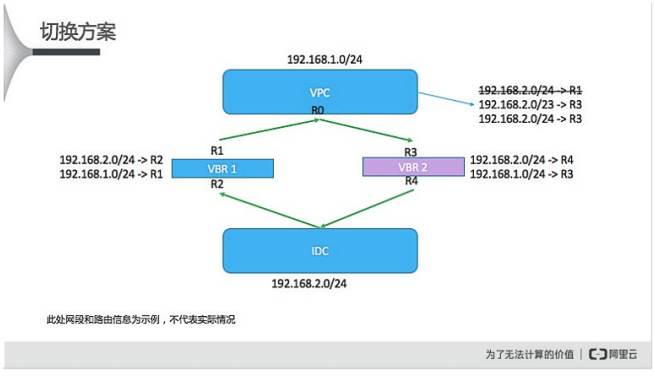
A.VPC增加路由规则192.168.2.0/23->R3，新添加网段必须略大于需要切换的网段，路由接口指向新VBR接口R3，此时流量路径不变，这步操作的主要功能是相当于为实际使用的192.168.2.0/24网络设置一个“默认”路由，因为它是192.168.2.0/23的子集，如果它自己的路由条目删除后，那么它将会按照这条指定的路由来转发流量到R3接口，即新专线的网络接口，而不是走0.0.0.0/0，从而导致流量异常。



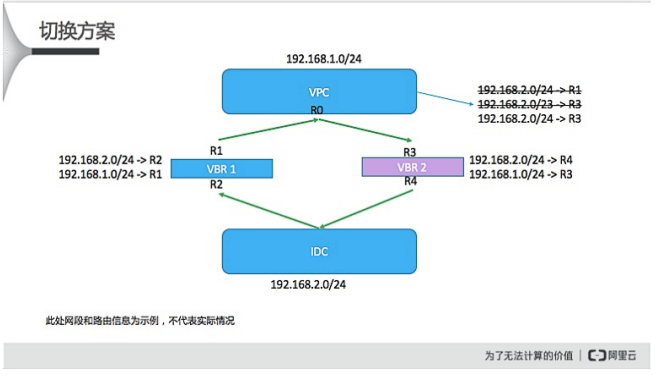
B.删除VPC中192.168.2.0/24->R1路由规则，此时，VPC到IDC的流量将会经过R3接口，原理前面已经说明，经过VBR2到达IDC，而IDC到VPC的回程路由仍然保持原路不变，VPC的发出和进入流量路径呈环状。



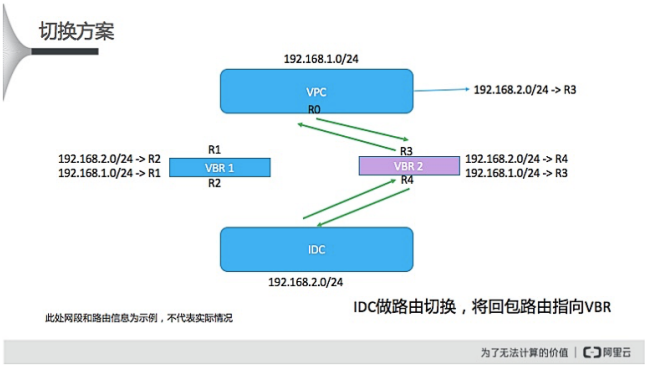
C.在VPC中添加 192.168.2.0/24 -> R3 路由规则，VPC到IDC的流量将会经过R3接口到达IDC，但实际匹配到的规则变成192.168.2.0/24 -> R3，192.168.2.0/23 -> R3已经不再生效，IDC到VPC的流量仍旧按照原来的规则从老的VBR1进入VPC。



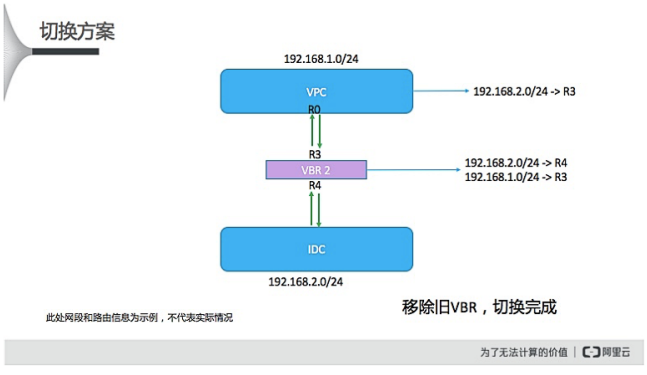
D.删除192.168.2.0/23 -> R3路由条目，这条路由在C步骤之后已经不再生效，因此此时删除这条路由对网络通信没有任何影响(前提是这条路由没有把其它子网包含进去)。



E.IDC侧做路由切换，经过前面的几部操作，VPC到IDC的流量已经是从VBR2(即新专线)通过，回程的流量还在老专线上，因此需要在IDC侧网络设备上进行路由更新，变更内容为192.168.1.0/24->R2更新为192.168.1.0/24->R4，指向VBR2。至此，双向流量已经全部切换完成。



F.同时监控带宽流量以及业务访问情况，如有异常，按照操作步骤逆序回滚。



如果有多条路由，重复A-E步骤，直到VBR1上的所有路由切换完成。

小结

不同于在IDC中的网络直接切换，VPC对网络层面的操作进行了抽象的封装，目前情况下，VPC添加路由条目之后不支持修改，即无法直接修改路由的下一跳接口,所以在线路发生变化的时候必须是通过新接口来接入，这个也是有别于IDC侧网络设置差异最明显的地方。因此本方案着重介绍了VPC侧路由的切换方式，IDC方面则是更改路由指向的方式直接进行调整。

VPC与专线的搭配提供了非常灵活的网络建设方式，借助于阿里云多地区服务能力，能够快速的搭建基于阿里云的跨地区甚至是覆盖全球的网络基础环境，更好的帮助客户实现快速发展。

基于阿里云视频服务开展直播业务的最佳实践

睿得
阿里云技术专家

一、引言

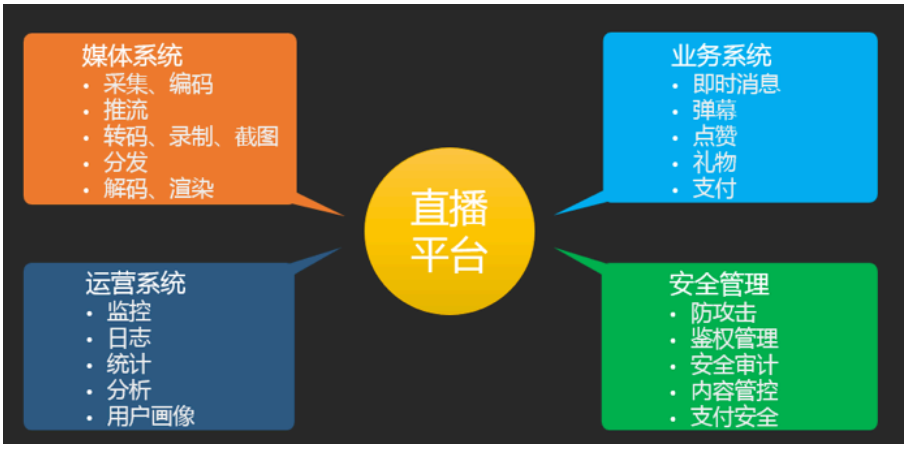
2016年是直播大爆发的一年，公认的直播年。据统计，在2016年，网络直播用户的规模已经达到了3.25亿人，直播市场规模达到150亿元，大大小小的直播平台数量已经超过200家。

去年8月，奥运会“行走的表情包”——傅园慧在某直播平台进行了直播首秀，直播的内容是和粉丝聊天。共有1000多万人在线观看，一个小时的直播活动，带来30多万元人民币的收入。目前顶级网红主播的年收入可以超过千万人民币，同时直播吸引了大量的眼球，给直播平台带来了巨额的收益。

直播很火，很多公司和创业者都想抓住这个机遇。那么问题来了——如何搭建一个视频直播系统呢?

要回答这个问题，首先需要来了解一下直播平台的基本组成。

二、直播平台基本组成



首先，直播系统最核心、最特殊的部分就是直播系统，或者叫媒体系统。直播系统包含了直播内容的采集、编码、推流、转码、录制、截图、分发、解码和渲染等功能模块。

其次，由于直播业务的特点，有一些特殊的业务功能，比如即时消息——粉丝需要通过即时消息与主播进行互动；在观看直播的时候通过发送弹幕进行互动；可以给的主播点赞，送礼物。礼物是要花钱购买的，所以必须要有支付的功能。

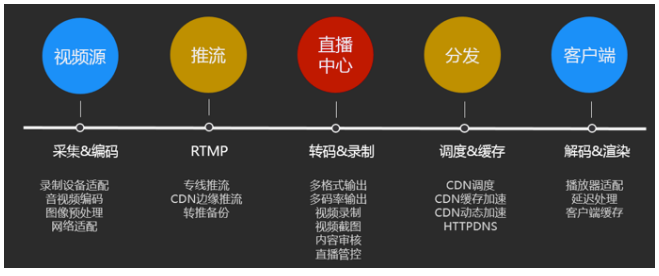
接下来是管理运营系统，要对系统进行实时监控、对业务日志进行统计分析、对用户画像，实现精准的推送、营销。

最后还有安全系统，包括防攻击、防盗链、鉴权管理、内容管控、支付安全

等等。

上面这些都是直播平台比较有特点的功能。其他还有一些传统或者其他业务平台需要的技术功能点，比如操作系统、数据库、业务系统的水平扩展和高可用，等等。所以总体来说，一个直播平台涉及到的技术点纷繁复杂，技术门槛是比较高的。本文重点介绍一下最核心的——直播系统部分的技术和实现。

三、直播系统基本架构



一个直播系统主要由视频源、直播推流、直播中心、内容分发和客户端这5大部分组成。

视频源负责直播内容的采集和编码，比如摄像机、PC摄像头、手机摄像头、监控摄像头。

编码后的直播内容，通过推流的方式，推送到直播中心。推流一般采用RTMP的协议。

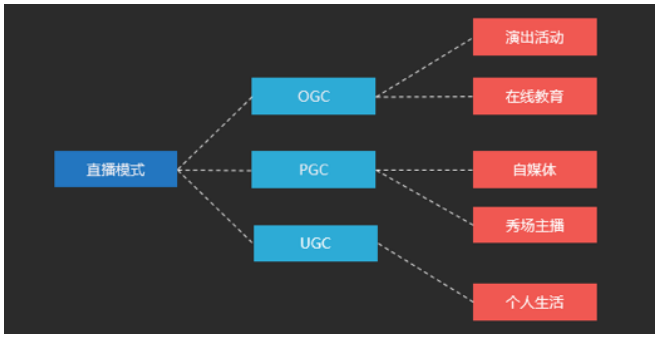
直播中心对直播内容进行处理，例如转封装、转码、加水印等等，处理后的直播内容再进行分发。

分发一般会使用CDN，利用CDN的节点覆盖，保证不同地区、不同运营商网络下客户的网络覆盖。

最后是客户端，在接收到直播内容后，客户端进行解码和渲染，最终显示在终端的屏幕上。

四、直播业务划分

在具体介绍直播系统的最佳实践之前，先介绍一下直播业务的划分。



业内比较通用的分类方式是将直播业务按照内容来源分为三类。

第一类是OGC（Occupationally-generatedContent），职业产生的内容，比如演唱会、赛事直播、在线教育。

第二类是PGC（Professionally-generatedContent），专业的人事产生的内容，比如秀场，展示自己的才艺。自媒体，网红，大V等等。

第三类是UGC（User-generated Content），用户自主产生的内容。比如直播一些个人的日常生活，聊聊天。

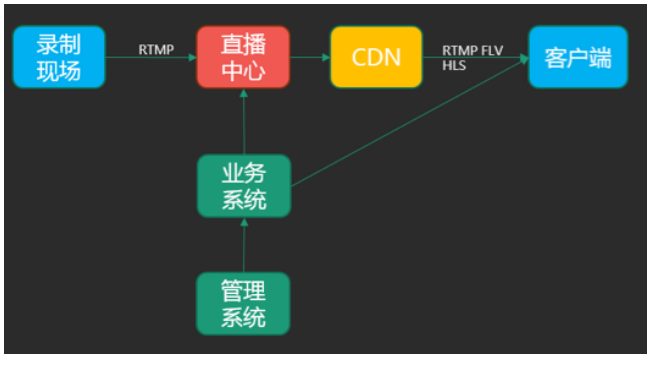
OGC直播业务场景的特点是直播源较为集中，直播源所在地域、使用的网络环境相对固定，业务上对直播的质量要求较高。

PGC与UGC类似，可以归为同一个大类，特点是直播源地域分布广泛，使用的网络环境各不相同各不相同，以移动终端为主。

五、搭建直播系统的最佳实践

针对上述两类业务场景情况不同的特点，具体介绍一下搭建直播系统的基本架构和需要注意的问题。

1.OCG业务场景



(1)OCG业务场景的特点

- 录制现场使用的是专业的视频采集设备。
- 直播活动一般只有一路或少数几路流（不同摄像机位）
- 直播活动时间表可以事前确定。
- 直播视频源相对单一和固定（比如演播厅、体育馆等）

(2)需要关注的问题

a.协议

目前主流的协议有RTMP、FLV和HLS三种。

前文提到的RTMP是Adobe公司推出的一种专门用于直播的协议，它的优点是时延迟较小，一般在1-3秒左右。缺点是使用了TCP协议，默认为1935端口，有可能会被防火墙等网络设备拦截掉。另外，由于它是专门的协议，通用性不如HTTP协议，比如在Android和IOS系统的移动终端上，原生

都是不支持RTMP协议的直播播放的。

FLV是基于HTTP协议的一种直播方式，利用FLV支持边下边播的特点，模拟一个无限大的FLV视频文件，实现直播，延迟与RTMP相当。由于采用了HTTP协议，所以通用性好一些，但是在Android和IOS设备上，原生也是不支持的。

HLS是苹果牵头推出的直播协议，把直播的视频流切成一个个小片段，每个小片段叫做一个分片。然后利用一个索引文件，类似于播放器里面的播放列表，终端根据索引文件，逐一的请求和播放每一个小分片。HLS的通用性最好，Android和IOS都可以原生支持，比如一个HLS的直播地址，直接使用手机中自带的浏览器访问，就可以观看直播。播放直播内容时使用的就是手机系统自带的播放器程序。HLS最大的缺点就是延时比较长，一般在几十秒，甚至分钟级。

所以要根据具体的业务场景来选择协议，如果是PC端观看，推荐使用FLV，如果是手机端观看，推荐使用HLS。

如果对延时的要求高，比如有互动的需求，就需要使用RTMP或者FLV，如果是手机端，就需要引入SDK或者相应的库文件进行定制开发，来支持RTMP或者FLV协议的播放。

b.码率

一般来说，码率越高，画质越好。码率低就会出现画面模糊的情况，特别是演唱会这样灯光复杂，变化频繁的场景，码率过低非常影响观看体验。

但是码率过高也会有问题，首先是带宽成本增加，CDN的费用会增加。其次客户端网络带宽不足或网络质量不佳就会出现卡顿。而且码率过高，客户端解码的压力也会增加，导致手机发热、电量消耗快等等。所以码率也是需要根据业务实际情况进行平衡选择的。

c.关键帧

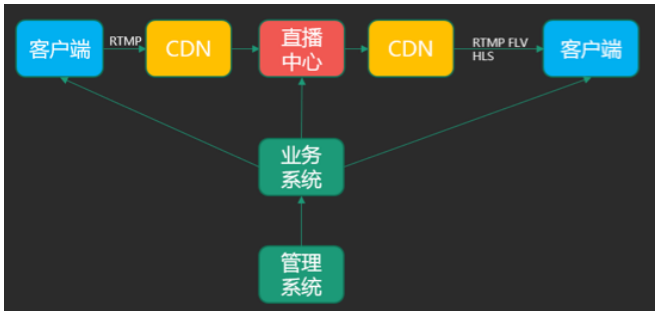
我们都希望用户在点击打开直播间的时候，第一时间就可以看到直播画面。但是视频是由多个画面组GOP（Group of Pictures）组成的，解码时必须要从关键帧开始才可以播放，假如GOP是10s，而客户在直播到第2秒的位置点击播放，那么就要等解码到8s后的位置才可以看到画面，可以设当的减小GOP长度来缓解这个问题。但是如果GOP长度设置过小，又会导致码率增加，播放带宽的需求增加。所以要设置合理的GOP间隔，一般来说1-2s一个关键帧比较合适。

d.缓存

服务端——直播中心对关键帧进行缓存，可以提升客户的首屏显示时间，即在直播中心中缓存最近一个完整的GOP的内容。这样客户端每次连上服务器的时候，都可以第一时间拿到关键帧。

同样，客户端的缓存设置，可以减少网络抖动引起的卡段，但是缓存过大会导致延迟增加，所以客户端的缓存也需要根据业务实际情况设置合适的大小。

2.UCG/PGC业务场景



(1)UCG/PCG业务场景的特点

- 录制采用的一般是非专业的PC端或移动端设备。
- 直播活动会有几百甚至上千路流同时进行直播。
- 直播活动时间不固定，无法实现确定。
- 直播视频源分散在各地，使用的接入网络也各不相同。

(2)需要关注的问题

a.终端适配

由于推流设备不同，首先需要做好推流端的适配，IOS相对好一些，终端型号数量不多。Android手机则是五花八门，不同型号的摄像头、CPU，系统版本等等，对终端的适配提出了很大的挑战。所以这类直播业务场景下，客户端的适配是一个很重要的关注点。

b.预处理

直播内容需要在客户端做一些预处理，比如加一些滤镜，最常见的就是美颜。大家都知道，现在在朋友圈里发个照片，不事先美颜一下是肯定不敢往外发的。同样直播的时候，主播们也是一样，没有美颜过得视频也是不敢直播的。另外，还有各种滤镜的功能，比如通过人脸识别，然后加一个兔子耳朵，小猫的胡子之类的滤镜的功能，这些都是要在客户端进行处理的。

c.视频编码

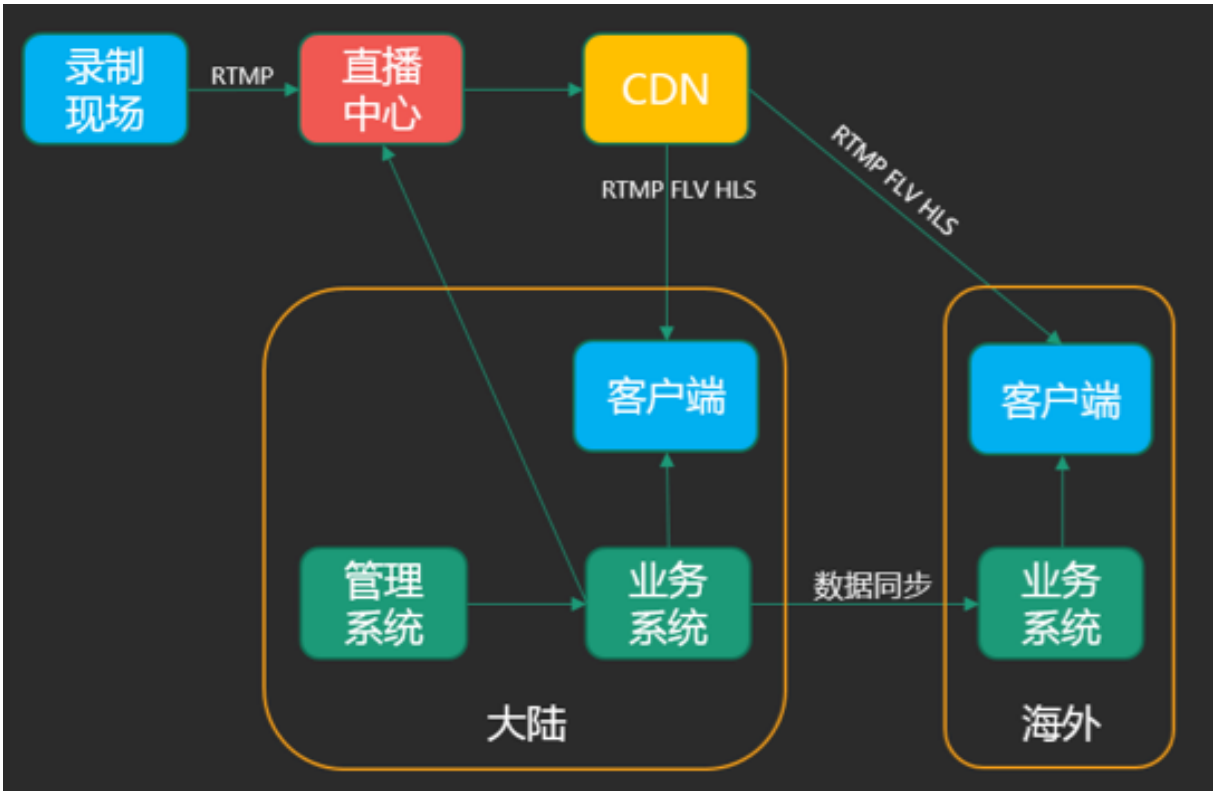
目前主流的视频编码格式是H264，兼容性好。H265是更新一代的视频编码技术，可以在同样的码率下传输更清晰地画面。但是目前还不够普及，比如在IOS设备上，原生的播放器，播放H265编码的视频没有画面只有声音。另外，H265编码的算法更加复杂，对播放设备的计算压力更大，更容易导致手机发热和电量损耗。所以目前还是推荐使用H264的编码格式。音频编码方面，推荐使用AAC格式，兼容性最好。考虑到手机的性能、功耗等因素，一般手机端推流，帧率在10-15

帧就可以。

d.边缘推流

由于在这种业务场景下，推流的客户端地域分布广，使用的运营商、网络各不相同。所以，建议使用CDN边缘推流，将直播内容先推送到就近的CDN节点上，然后利用CDN的网络传输至直播中心，解决推流侧网络的覆盖问题。

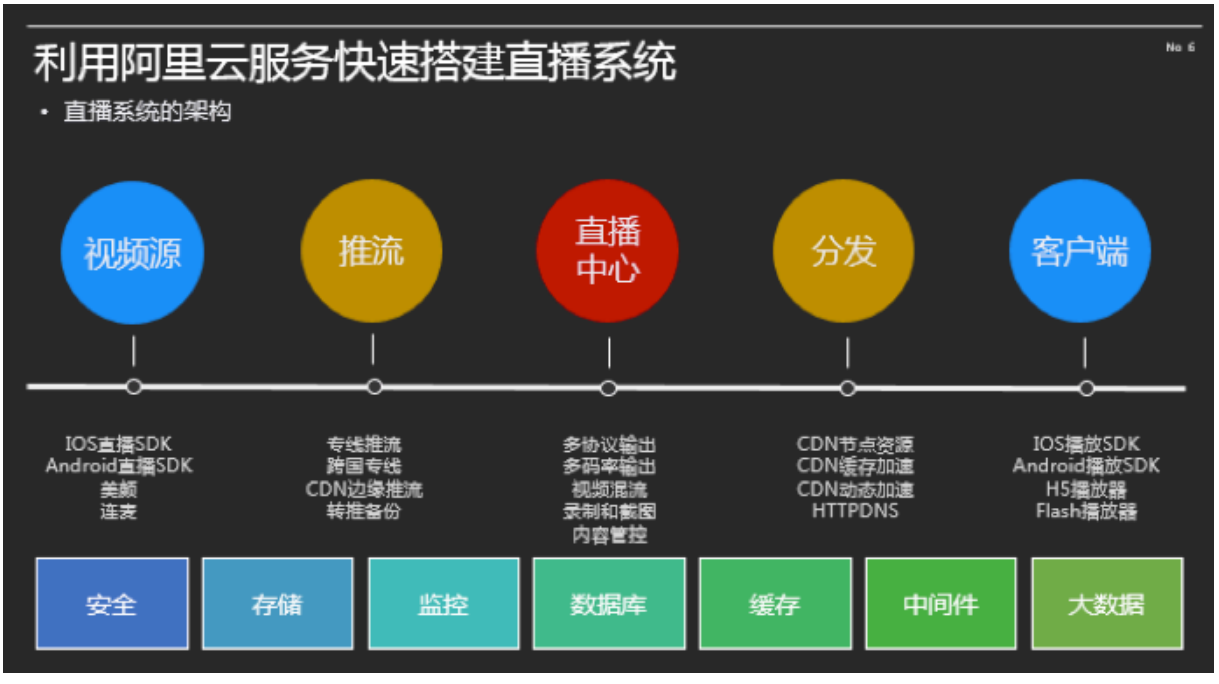
3.全球直播



还有一个比较特殊的场景是全球直播业务。一般是OGC的业务中会使用到，这种场景要求全球一致的观看体验。由于国际链路不稳定，建议国际专线推流到服务端，利用CDN做视频内容的分发。业务系统分布式部署，进行数据同步。如果有更高的要求，CDN也可以分开，不同地域使用不同的CDN服务商进行覆盖，做智能解析。

六、总结

最后回顾一下直播系统的5大组成部分：



在视频源，阿里云提供了Android、IOS的直播SDK，包括视频的采集、编码和推流功能，同时支持美颜、连麦等功能。

在推流的环节，阿里云的高速通道产品，提供了专线，包括国际专线的接入，可以部署快速，且计费方式灵活，非常适用于赛事或演出场景的需要。阿里云的CDN支持边缘推流，可以解决UGC、PGC类业务场景下推流端的网络接入问题。

阿里云视频直播业务中的直播中心，支持实时转码，多格式多码率的输出；支持视频混流，比如多主播互动、广告垫片等功能。同时还提供直播视频的录制，截图，以及图像的识别、鉴黄等内容管控的功能。

在分发的环节，阿里云CDN提供了超过1000个的节点，支持静态的比如TS切片的缓存，以及动态的网络加速功能。同时提供HTTPDNS服务，防止客户端被运营商劫持。

最后在客户端，阿里云提供了IOS、Android的播放器SDK、网页方式的H5播放器以及Flash播放器，覆盖了手机端和PC端。

除此之外，阿里云拥有的完美的云产品生态圈，提供了安全、存储、数据库、中间件、大数据等产品协助用户实现直播业务其他相关业务系统的搭建。可以大大的降低开展直播业务的门槛，让用户技术型团队的聚焦在自己的业务上，使创业团队有限的预算产生最大的效果。

负载均衡（SLB）使用最佳实践

五贤
阿里云技术专家

负载均衡 (Server Load Balancer，下文简称SLB) 的引入，可以降低单台云服务器ECS（下文简称ECS）出现异常时对业务的冲击，提升业务的可用性。同时，结合弹性伸缩服务，通过动态调整后端服务器，可以快速对业务进行弹性调整(扩容或缩容)，以快速应对业务的发展。

本文先对SLB的使用限制和常见误区进行说明，然后介绍SLB的使用最佳实践。

一、SLB 基础原理

在开始使用SLB之前，建议您提前阅读阿里云官方如下产品文档，了解SLB的相关基础原理：

- 负载均衡技术原理浅析(文档ID-39444)
- 负载均衡健康检查原理浅析(文档ID -39455)
- 负载均衡网络流量路径说明(文档ID-39440)
- 负载均衡高可用概要说明(文档ID -39449)

二、SLB 使用限制

在使用SLB进行业务部署之前，请您务必了解SLB的相关使用限制，以免对后续使用造成困扰或对业务造成影响。

1.产品与业务规格限制

SLB的相关资源或功能存在限制，部分可以通过提交工单申请调整，部分则无法调整。详细说明，可以参阅官方文档：SLB 使用限制 (文档 ID - 32459)。

另外，目前官方已经推出性能保障型实例，对实例性能提供保障。您可配置和查询不同规格实例的具体性能指标。

相关说明可以参阅官方文档：性能保障型实例 (文档 ID -27657)。

2.技术限制

SLB 在技术层面还在逐步增强和完善，截止本文发稿，还存在如下技术限制：

- 在 4 层 (TCP协议) 服务中，不支持添加进后端云服务器池的 ECS 既作为Real Server，又作为客户端向所在的 SLB 实例发送请求。因为，返回的数据包只在云服务器内部转发，不经过 SLB。所以，在 SLB 后端 ECS 去访问其 VIP 是不通的。
- 仅支持 TCP/UDP (4 层) 和 HTTP/HTTPS (7 层) 这 4 种协议。
- 后端服务器仅支持 ECS，不支持第三方云服务器。
- 仅支持轮询(RR)、加权轮询(WRR)和最小加权连接数 (WLC) 这 3 种调度算法。

• 不支持 7 层 SSL Session 超时时间的调整。

• 不支持 7 层 HTTP Keep-Alive 超时时间的调整。

说明:如果客户端访问 SLB HTTP 监听时使用长连接，那么这条连接最长的空闲时间为 15s 。即如果超过 15s 没有发送任何 HTTP 请求，这条连接将会被SLB主动断开。如果您的业务可能会出现超过15s的空闲,需要从业务层面检测连接的断开并重新发起连接。

• 不支持转发超时时间的调整:

当前配置: TCP 900s，UDP 300s，HTTP 60s，HTTPS 60s。

说明:上述配置是指 SLB 服务端从后端接收数据并进行转发的超时时间，并非健康检查超时时间间隔。如果超时，通常会向客户端返回 504 错误码。

• 金融云环境SLB基于安全性考虑，仅允许开放如下特定端口:80，443，2800-3300，6000-10000，13000-14000。

三、SLB 常见误区

从阿里云处理的历史案例看，用户在SLB的规划和使用过程中有很多常见误区。接下来逐一说明。

【误区1】:SLB后端只需添加一台ECS，确保链路能跑通即可

用户在 SLB 后端只添加一台服务器时，虽然链路能跑通，客户端也能正常访问。但却失去了 SLB 消除 ECS 单点的基本能力。如果这台仅有的ECS出现异常，那边整个业务访问也会出现异常。

建议:至少在 SLB 后端加入两台以上 ECS。以便单一服务器出现异常时，业务还能持续正常访问。

【误区2】:后端ECS能正常访问，但SLB无法访问，则说明SLB出现了异常

用户通过SLB访问业务出现异常。但 hosts 绑定后端 ECS 的公网IP能正常访问。用户据此推断后端业务是正常的，是SLB服务端出现异常导致业务访问异常。

其实，由于SLB的数据转发和健康检查都是通过内网进行的。所以，从后端ECS的公网IP进行对比访问测试，并没有可比性，并不能反应真实访问情况。

建议:出现异常时，在后端 ECS 之间，通过内网 IP 做对比访问测试。

【误区3】: SLB 的 VIP 能 ping 通就说明配置是正常的

用户通过 ping SLB 的 VIP 地址来判断 SLB 服务的有效性。

其实这种测试非常不可靠。因为ping响应是由SLB服务端直接完成的，与后端ECS无关。所以，正常情况下:

• 只要配置了任意监听，即便相应监听处于异常状态，SLB VIP ping也是正常的。

• 相反，如果 SLB 没有配置任何监听，其 VIP 是 ping 不通的。

建议:对于 4 层服务，通过 telnet 监听端口进行业务可用性测试;对于 7 层服务，通过实际的业务访问进行可用性测试。

【误区4】:已经调整了健康检查间隔，结果还会出现访问超时

用户反馈已经调大了健康检查的最大间隔时间，但客户端还是由于访问超时收到 504 错误。

其实，虽然健康检查及业务转发都是由 SLB 服务端相同的服务器承载，但却是完全不同维度的处理逻辑。来自客户端的请求，经由 SLB 转发给后端 ECS 后，SLB 服务端会有接收返回数据的超时窗口;而另一方面，SLB服务端持续对后端ECS根据检查间隔配置进行健康检查。这两者之间没有直接关联关系，唯一的影响是在后端 ECS 健康检查失败后，SLB 不会再对其进行数据转发。

建议:客户端访问超时时，结合业务与 SLB 默认超时时间进行比对分析;健康检查超时时，结合健康检查与业务超时时间进行比对分析。

【误区5】:从后端日志看，健康检查间隔与监听配置的间隔时间不一致

用户反馈通过 SLB 后端 ECS 的业务日志进行统计分析，发现健康检查的间隔非常短，与之前在创建监听时配置的健康检查间隔时间不一致。

这个问题在文档负载均衡健康检查原理浅析(文档ID-39455) 中有相关说明:LVS 集群内所有节点，都会独立、并行的遵循健康检查配置去对后端 ECS 进行定期检查。由于各LVS节点的检查时间并不同步，所以，如果从后端某一ECS上进行单独统计，会发现健康检查请求在时间上是连续的，并不会与配置的间隔时间一致。

建议:如果健康检查频率过高对业务造成影响，可以参阅文档健康检查导致大量日志的处理 (文档 ID -39458) 进行处理。

【误区6】:大量健康检查形成 DDoS 攻击，导致服务器性能下降

用户认为 SLB 服务端使用上百台机器进行健康检查，大量健康检查请求会形成 DDoS 攻击，造成后端 ECS 性能降低。

实际上，无论何种模式的健康检查，其规模均不足以达到类似于 DDoS 攻击的量级: SLB 集群会利用多台机器(假定为 M 个，个位数级别)，对后端 ECS 的每个服务监听端口 (假定

为 N 个），按照配置的健康检查间隔(假定为 O 秒，一般建议最少 2s) 进行健康检查。那么，以 TCP 协议健康检 查为例，每秒由健康检查产生的 TCP 连接建立数为：M*N/O 。

从该公式可以看出，M 和 N 都是固定的，而且值很小。所以，最终健康检查带来的每秒 TCP 并发请求数，主要取决于创建的监听端口数量。所以，除非有巨量的监听端口，否则由健康检查产生的连接请求，根本无法达到DDos攻击级别，实际对后端 ECS 的网络压力也极低。

建议:如果健康检查频率过高对业务造成影响，可以参阅文档健康检查导致大量日志的处理 (文档 ID -39458) 进行处理。

【误区7】:为了降低健康检查的影响，将健康检查间隔设置得很长

用户为了降低健康检查对业务的影响，将检查间隔时间设置得很长。

其实，这样配置会导致当后端 ECS 出现异常时，负载均衡需要经过较长时间才能侦测到异常。尤其当后端 ECS 间歇性不可用时，由于需要连续多次检测失败时才会判定不健康进而停止业务转发。所以，如果检查间隔过长，会导致负载均衡集群可能根本无法发现后端 ECS 不可用。

建议:如果健康检查频率过高对业务造成影响，可以参阅文档健康检查导致大量日志的处理 (文档ID-39458) 进行处理。

【误区8】:移除服务器与权重置零的效果是一样的

用户在进行业务调整时，认为直接将服务器从 SLB 后端移除，或将其权重置零即可，两者效果是一样的。

其实，两者有很大区别，相关操作对业务的影响也不一致：

• 移除服务器：已经建立的连接会一并中断，新建连接也不会再转发到该 ECS。

• 权重置零：已经建立的连接不会中断，直至超时或主动断开。新连接不会转到该ECS。

建议:在业务调整或服务器维护时，提前将相应服务器的权重置零，直至连接持续衰减至零。业务维护完成后，再恢复权重配置，以降低对业务的影响。

【误区9】:单个连接就能使用监听的峰值带宽

SLB在创建监听时可以指定带宽峰值。但用户通过单一客户端进行测试时，发现始终无法达到该峰值。

其实，由于SLB是基于集群方式部署和提供服务的。所以，所有前端请求会被均分到集群内不同的服务器上进行转发。相应的，在监听上设定的带宽峰值也会被均分后设置到各服务器节点。因此，单个连接下载的流量上限公式为：

单个连接下载峰值=设置的负载均衡总带宽/(N-1)
*注：N 为 SLB 服务端流量转发分组个数，当前值一般为 4.

示例：假设在控制台上设置的带宽峰值为10MB/s，那么单个客户端可下载的最大流量为：10/(4-1)≈3.33MB/s

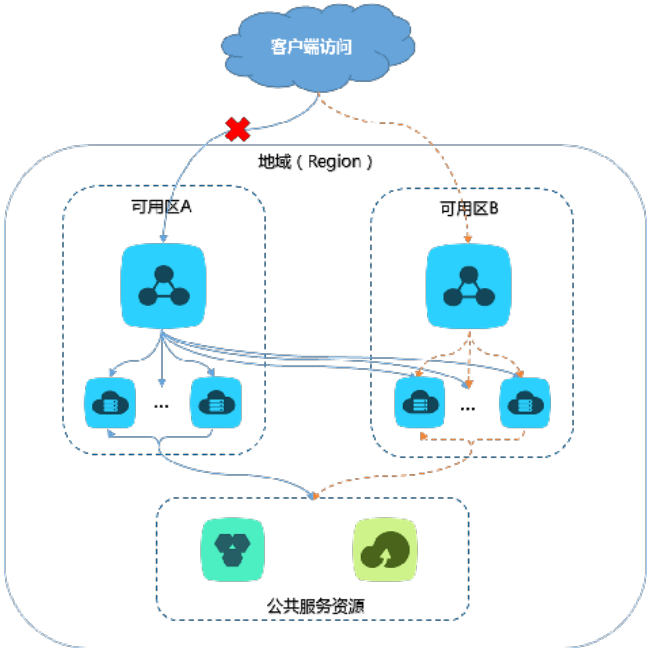
建议:建议在配置单个监听的带宽峰值时，根据实际的业务情况并结合上述实现方式设定一个较为合理的值，从而确保业务的正常对外服务不会受影响和限制。

四、SLB 使用最佳实践

接下来，结合SLB的产品特性与历史案例，从多个维度阐述使用 SLB 的最佳实践。

1.业务架构

注：内网环境下，不支持多可用区，只看图例的单边即可。



SLB在公网环境下的典型业务架构如上图所示。基于多可用区特性，当主可用区出现异常时，SLB会自动将流量切换到备可用区。但在实际的业务架构设计中，建议您同步关注如下要点：

• 实例类型与计费方式

如果只用于内部服务分发，则创建私网实例即可。

按流量计费，适用于波峰波谷效应明显的业务。

按带宽计费，适用于带宽较为平稳的业务。

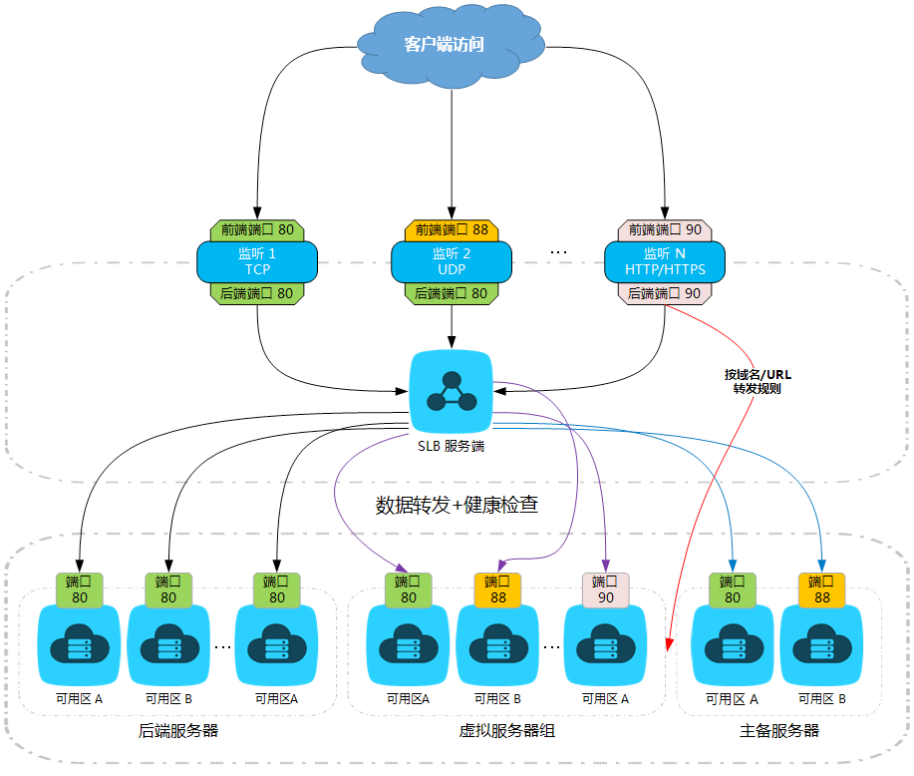
• 区域与多可用区

为了降低延迟，建议选择距离您的客户最近的地域进行SLB部署。

并非所有区域都支持多可用区特性(具体支持情况以您在控制台所见为准)，请您结合业务情况选择合适的可用区。

在配合使用多可用区特性时，后端 ECS 也必须同时在相应的主备可用区部署，才能保障出现异常时，相应可用区有 ECS 能进行业务承载。

2.配置监听



如上图所示，SLB 支持创建多种协议监听，然后按转发策略，将前端业务请求转发到后端多种逻辑服务器集上进行处理。在 SLB 服务配置的各主要步骤中，请关注如下要点。

3.选择监听协议

目前 SLB 支持创建 TCP/UCP/HTTP/HTTPS 这 4 种协议的监听。您可参考以下表格，根据业务情况选择适合的协议创建监听：

协议类型	建议的应用场景	特性
TCP	• 注重可靠性，对数据准确性要求高，速度可以相对较慢的场景； • 示例：文件传输、发送或接收邮件、远程登录、无特殊要求的 Web 应用；	• 面向连接的协议； • 在正式收发数据前，必须和对方建立可靠的连接； • 基于源地址会话保持，在网络层可直接看到来源地址； • 监听支持 TCP 和 HTTP 两种方式进行健康检查； • 转发路径短，所以性能比 7 层更好，数据传输速度更快；
HTTP	需要对数据内容进行识别的应用，如 Web 应用、小的手机游戏等。	• 应用层协议，主要解决如何包装数据； • 基于 Cookie 会话保持，可以使用 X-Forward-For 获取源地址； • 监听只支持 HTTP 方式健康检查；
HTTPS	有更高的安全性需求，需要加密传输的应用。	• 加密传输数据，可以阻止未经授权的访问； • 统一的证书管理服务，用户可以将证书上传到负载均衡，解密操作直接在负载均衡上完成。
UDP	• 关注实时性而相对不注重可靠性的场景； • 示例：视频聊天、金融实时行情推送；	• 面向非连接的协议； • 在数据发送前不与对方进行三次握手，直接进行数据包发送，不提供差错恢复和数据重传； • 可靠性相对低但数据传输速度快；

补充说明：

• 并非只要是 Web 网站就必须使用 HTTP 协议。大部分没有特殊 HTTP 要求的网站，使用 TCP 监听 80 端口就可以满足业务需求。

• SLB 集群采用 LVS 和 Tengine 实现。其中 4 层监听 (TCP/UDP) 经过 LVS 后直接到达后端服务器，而 7 层监听 (HTTP/HTTPS) 经过 LVS 后，还需要再经过 Tengine 转发，最后达到后端服务器。由于比 4 层多了一个处理环节。因此，7 层监听的性能相对要差一些。

4.选择后端服务器集模式

当前 SLB 后端支持按 3 种逻辑创建服务器集。对比说明如下：

集合模式	配置权重	指定服务端口	服务器数量限制	其它特性
后端服务器	支持	不支持	无限制	创建监听时的默认映射服务器集
虚拟服务器组	支持	支持	无限制	有更大的灵活性
主备服务器组	支持	支持	2 台	只能在 TCP/UDP 监听上使用

建议配置：

• 按业务逻辑创建不同的虚拟服务器组，然后创建相应的监听与之对应。

• 无论创建何种服务器集合，均结合SLB多可用区特性，同时加入位于不同可用区的服务器，以实现跨可用区容灾。

• 设置过多转发规则会增加业务维护的复杂度，建议尽量精简策略条目。

5.选择转发策略

当前 SLB 支持 3 种转发策略，其使用场景及要点如下：

注：权重代表相应服务器所承载的业务的相对占比，而非绝对值。

转发策略	算法说明	使用要点
加权轮询 (WRR)	按权重比例轮流分配新增连接。	• 根据后端 ECS 规格的不同，配置相应的权重。 • 如果是长连接业务，可能会导致老服务器的连接数持续增加， 而新加入服务器的连接数相对非常低，造成负载不均的假象。
加权最小连接数 (WLC)	• 在 SLB 服务端，实时统计与后端 ECS 已建立的 ESTABLISHED 状态连接数，来评估相应服务器的负载情况。 • 按权重比例，将新增连接分配给活动连接较少的服务器，最终尽可能使后端服务器上已建立连接数与其权重成正例。	当前暂未实现新增服务器的过载保护或缓冲机制。所以，如果业务并发非常高，可能会导致新增服务器连接数陡增，对业务造成影响。建议新增服务器时，逐步调高权重。
轮询 (RR)	按顺序逐一分发新增连接。	必须手工确保后端 ECS 的业务承载能力一致。

示例:假设有100个新增连接，则在不同的调度算法下，不同服务器分配的连接数示意如下:

服务器	权重	占比	加权轮询	加权最小连接数	轮询
A	50	50/(100+50+50)=25%	将 100*25%=25 个连接分发给服务器 A	实时统计连接数，逐一将新增连接分配给活动连接最少的服务器。最终使其新增连接数占比大致为 25%	不考虑权重，按顺序分发新增连接到服务器 A/B/C
B	100	100/(100+50+50)=50%	将 100*50%=50 个连接分发给服务器 B	↑ 同上，最终使其新增连接数占比大致为 50%	↑ 同上
C	50	50/(100+50+50)=25%	将 100*25%=25 个连接分发给服务器 C	↑ 同上，最终使其新增连接数占比大致为 25%	↑ 同上
D	0	0/(100+50+50)=0%	服务器下线，不分配任何连接	← 同左	← 同左

补充说明:

如果配置的是 7 层监听，还可以配置按域名/URL 的转发策略，更多相关描述，可以参阅产品文档:负载均衡按域名和 URL 转发常见问题 (文档 ID- 27651)。

6.配置健康检查

健康检查用于实时探测后端 ECS 上的业务可用性，以避免新增连接被分发到异常服务器从而对业务访问造成影响。由于是 SLB 的核心服务，所以 4 层协议 (TCP/UCP) 监听的健康检查无法关闭。

在配置健康检查时，注意如下要点:

• 根据业务情况合理选择 TCP 或 HTTP 模式健康检查

TCP模式健康检查只能检查端口的可用性，对 7 层 HTTP 状态无法感知。所以，Web 类 7 层业务建议配置 HTTP 模式健康检查。

TCP 模式健康检查由于在连接建立后即主动发送 RST 断开连接，所以可能会导致上层业务认为是非正常中断，导致产生大量关联错误日志。该问题可以参阅文档健康检查导致大量日志的处理 (文档 ID -39458) 进行处理。

如果使用 HTTP 模式健康检查，则建议合理调整后端日志配置，避免过多健康检查 HEAD 请求业务性能造成影响。

• 合理配置健康检查间隔

根据实际业务需求配置健康检查间隔，避免因为健康检查间隔过长导致无法及时发现后端 ECS 出现异常。具体建议可以参考文档:健康检查的相关配置，是否有相对合理的推荐值? (文档 ID -39453)

• 合理配置检查端口与检查 URL

对于 4 层 TCP 协议，由于虚拟服务器组可以由后端 ECS 上不同的端口来进行业务承载。所以，在配置健康检查时不要设置检查端口，而应默认使用后端服务器的端口进行健康检查。否则可能导致健康检查失败。

对于 7 层 HTTP 协议，使用虚拟服务器组转发时，确保健康检查配置策略中的 URL 在后端每台机器上都可成功访问。

使用 HTTP 模式健康检查时，不要对根目录进行检查，建议配置为对某个静态页面或文件进行检查，以降低对主页的冲击。但是，另一方面，相应的，这可能会带来误判:即健康检查 URL 虽然访问是正常的，但主页实际上已经出现异常。

• UDP 健康检查

当前 UDP 协议健康检查可能存在服务真实状态与健康检查结果不一致的情况:

如果后端 ECS 是 Linux，在大并发场景下，由于 Linux 的 ICMP 攻击防护机制，可能会限制服务器发送 ICMP 的速度。此时，即便服务已经出现异常，但由于无法向前端返回 “port XX unreachable” 报错信息，就会导致 SLB 由于没收到 ICMP 应答进而判定健康检查成功，最终导 致出现真实服务状态与健康检查结果不一致的情况。

如果相应服务器非正常关机，由于也不会返回“port XX unreachable”报错信息。所以也会导致健康检查处于“正常”状态的误报。

7.选择与配置后端 ECS

在后端 ECS 的选择与配置时，注意如下要点：

• 在同一个服务器集中，同时加入不同可用区的服务器，以实现同城容灾。

• 根据服务器规格的差异，配置与之成配比的权重。

• SLB后端最少配置两台以上 ECS，以避免单台 ECS 出现异常导致整个业务不可用。

• 后端 ECS 建议无状态部署：即只用于计算，而数据调用与存储后连至后端的OSS、RDS等公共服务。这样可以无需受 ECS 之间的数据同步问题困扰。

• 建议基于相同的镜像或自定义镜像创建后端ECS服务器，以保障配置的一致性。

• 后端 ECS 归属安全组及系统内部防火墙必须对 SLB 服务端地址段进行访问放行。

8.调整后端 ECS 的操作建议

(1)新增后端服务器

在往 SLB 内新增服务器时，注意如下要点：

• 建议新增服务器时，将权重置零，在确保业务正常运行后，再调整权重使其上线。

• 如果业务是长连接，在新增服务器时，建议先将相应监听的调度算法修改为轮询模式，然后逐步调高新增服务器的权重，以降低对该新增服务器的过高冲击。

(2)调整、维护业务和后端服务器

在后端ECS 服务器的日常调整、维护过程中，注意如下要点：

• 选择合适的业务窗口，定期对后端ECS进行重启操作(操作前先创建快照进行备份)，以便应用系统补丁，并主动触发、感知系统内部错误等可能引发服务器无法正常启动等问题。

• 业务更新或系统维护时建议的操作步骤：

a. 将相应 ECS 的权重置零。此时，已建立的连接不受影响，但新增连接不会再分发到该服务器。

b. 周期性的在系统内通过 netstat/ss 等指令，确保已建立连接逐步自动断开，直至衰减置零(如果客户端有自动重连机制，则根据业务情况也可以忽略本步骤)。

c. 对服务器进行快照备份。

d. 进行业务调整、服务器配置、服务器重启等操作。

e. 操作完毕后，确认业务重新正常上线。然后逐步调高服务器权重，使其重新上线。

f. 逐一对服务器集内所有服务器按上述步骤进行维护。

(3)移除服务器

如果确认服务器不再使用，不建议直接移除服务器。而是参阅前述步骤，将相应服务器权重置零。待已建立连接全部正常断开后，再移除该服务器。

9.常见问题的排查思路

SLB在使用过程中的常见问题与排查思路可以参阅下列文档，本文不再详述：

• 健康检查异常的排查思路 (文档 ID -27702)

• 负载均衡返回 HTTP500/502/504 错误的处理 (文档 ID-39481)

• 后端 ECS 负载不均衡问题的处理 (文档 ID -39491)

10.典型案例

后端 ECS 负载分发不均

(1)问题场景

客户反馈后端 ECS 的规格一致，但其中几台 PPS 一直非常高。而且新加入的服务器，相对负载非常低。同时，业务高峰期，相关服务器健康检查异常，导致异常服务器下线，并进一步导致其它服务器负载升高。最终引发整个 SLB 后端出现雪崩式异常，业务受到严重影响。

(2)排查思路

• 业务监听使用 WRR (加权轮询)调度算法，在 SLB 服务端统计，在单位时间内分发到各服务器的连接数是均衡的。

• 异常服务器上的连接数持续居高不下，尝试调高 OS 层面的 TCP 连接数限制等配置，无明显效果。

• 和客户交流，结合业务分析，相关连接来源正常，都是正常的业务连接。同时，业务场景是物联网的监控数据上报，使用的是长连接。所以除非主动断开相关连接，否则相关连接会持续堆积。

(3)应对措施：

• 将负载较低的服务器的权重值调低。

• 将监听的调度算法修改为WLC(加权最小连接数)。

• 将异常服务器的权重值置零。

• 将负责较低的服务器的权重值逐步调高。

说明：以上步骤用于将负载逐步转移到新加入的服务器，并避免对业务造成重大冲击。

• 待异常服务器的连接数下降到合理水平后，将其重新上线(重新配置权重)。

• 由于后端服务器负载总体偏高，导致业务高峰期多台服务器无法满足业务需求，出现雪崩式异常。所以，同步建议客户对 SLB 后端服务器进行横向扩容。

——阿里云智能对话分析服务深度解析

从服务对话中挖掘价值

仓鹰

阿里云资深开发工程师

一、智能对话分析服务的由来

阿里云作为多个季度营收同比增长100%以上的云计算公司，业务突飞猛进的同时，也带来了服务量的激增，阿里云的工程师每天都要处理大量的工单以及电话，为了保证每个用户的问题都能得到解决，并有良好的服务体验，对服务数据进行多维度的分析成为迫切的需求。如何从海量的服务数据中筛选出有服务风险的场景？智能对话分析服务应运而生。

对服务质量的把控和服务数据的挖掘不仅是阿里云的需求，大多数企业用户也普遍存在这样的需求，所以在满足内部需求以后，阿里云把这种技术能力开放给了用户，帮助企业提升服务质量、监控服务风险、优化服务策略。

二、智能对话分析服务的介绍

智能对话分析服务能够对所有服务过程产生的数据进行全量的自动化智能分析，并全面挖掘服务数据的价值，可做到电话一挂断，立刻触发语音转文本并做分析，实时生成分析结果，可供监测人员及时处理；通过灵活的规则配置，覆盖多种复杂的业务场景，不仅做到低成本的数据分析，还能保证分析结果充分满足企业业务所需。

智能对话分析最核心的技术是规则系统，通过有逻辑关系的基本单元——算子，组成拥有一定智能的规则，来对数据进行检索，极大的提升了数据质检的准确率，把服务质检这件事从传统的体力劳动中解放出来。

系统的质检规则：

1.基本单元“算子”

算子是规则中不可分割的最基本的最小单位，例如检测关键词，检测对话时间间隔，检测是否疑问句等功能。

2.算子组成“条件”

例如，创建2个算子，分别为检测客户是否有抱怨情绪，检测回答是否超时，此时将这三个算子 1 && 2组合成一个条件，这样就生成了一个“因为回复超时而造成客户情绪不好”的条件。

3.条件组成“规则”

步骤2创建a，b，c三个条件，通过例如 a && b || c 这样的逻辑组合成一个规则。

智能对话服务分析服务通过一个个这样的规则，对数据进行分析。

注： && 代表逻辑“与”，|| 代表逻辑“或”，! 代表逻辑“非”

三、智能对话分析场景示例

图 1

对话内容如上图所示，阿里云工程师和客户沟通中向客户索取密码做诊断后，必须要提醒客户修改密码，这是个相对比较复杂的服务规则场景。智能对话分析如何对这一规则进行进行自动检测呢?

规则=!条件2

条件1：向客户索要密码：

(1)检查范围：全文客服说过的话；

(2)算子：匹配“向客户索要密码”的语义；

条件2：提醒客户修改密码：

(1)检查范围：条件1命中的句子之后，客服说的话

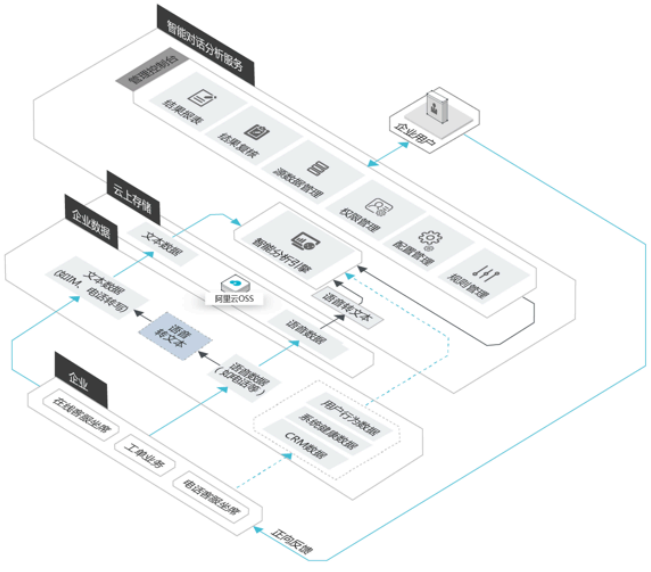
(2)算子：匹配“提醒客户修改密码”的语义

系统对这段对话应用该规则的过程如图所示：

图 2

以上是智能对话分析服务中最简单的例子，智能对话分析的系统支持7种算子，“与”，“或”，“非”逻辑关系，基于这几种基本单元，用户可以创建多种算子组合而成的应用场景，让复杂的对话分析成为可能。

四、智能对话分析平台核心能力



1.工单、电话数据化能力

企业售前、售中、售后都存在海量的服务数据，挖掘这类数据的第一步便是将这些数据结构化，包含两个重要步骤：

(1)语音数据文本化

基于阿里云自主研发的ASR系统首先可以将语音通话内容转换为文本，通过语音中的声纹特点，自动鉴定A/B角色，从而根据角色进行对话分析。

(2)文本数据结构化

智能对话分析服务能够对通话内容结构化存储（其它类型数据客户可自行封装），从而方便对数据进行检索和分析，将杂乱的数据有序化。

2.对话内容智能分析

智能对话分析服务的调度系统能够灵活地手动启动分析任务，也可以设置定时任务，自动对新数据进行入库和分析，轻松实现数据的 T+0 或 T+1 分析。

基于智能对话分析服务的数据处理能力，已经有金融、制造、电商等多个领域顶级厂商对系统进行试用，未来阿里云还将提供呼叫中心+对话分析一揽子解决方案。

阿里云智能机器人—云博士

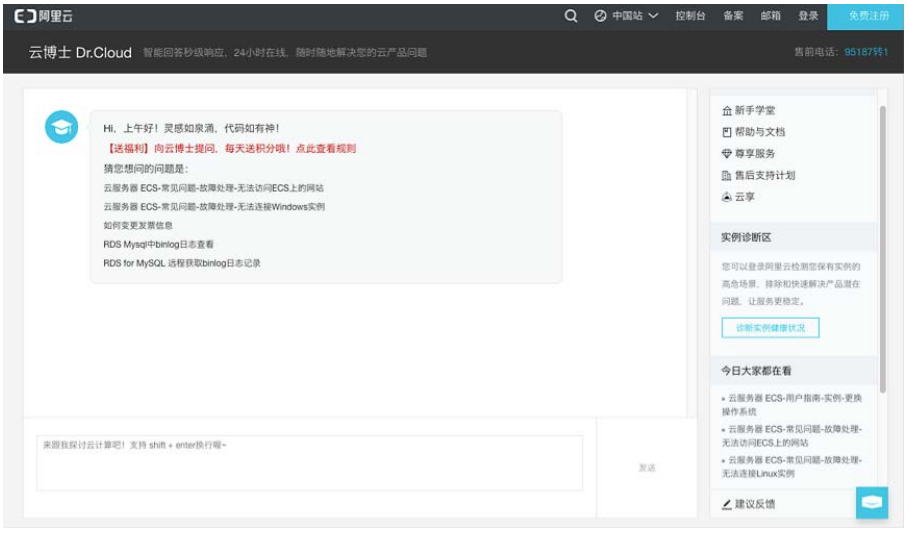
舜刀
阿里云研发专家

一、简介

近几年来，阿里云作为国内最大的云计算服务提供商，随着用户量的急剧上升，产品规模的迅速扩大，用户的服务需求也随之攀升，如何利用阿里云人工智能技术高效的解决用户的问题？

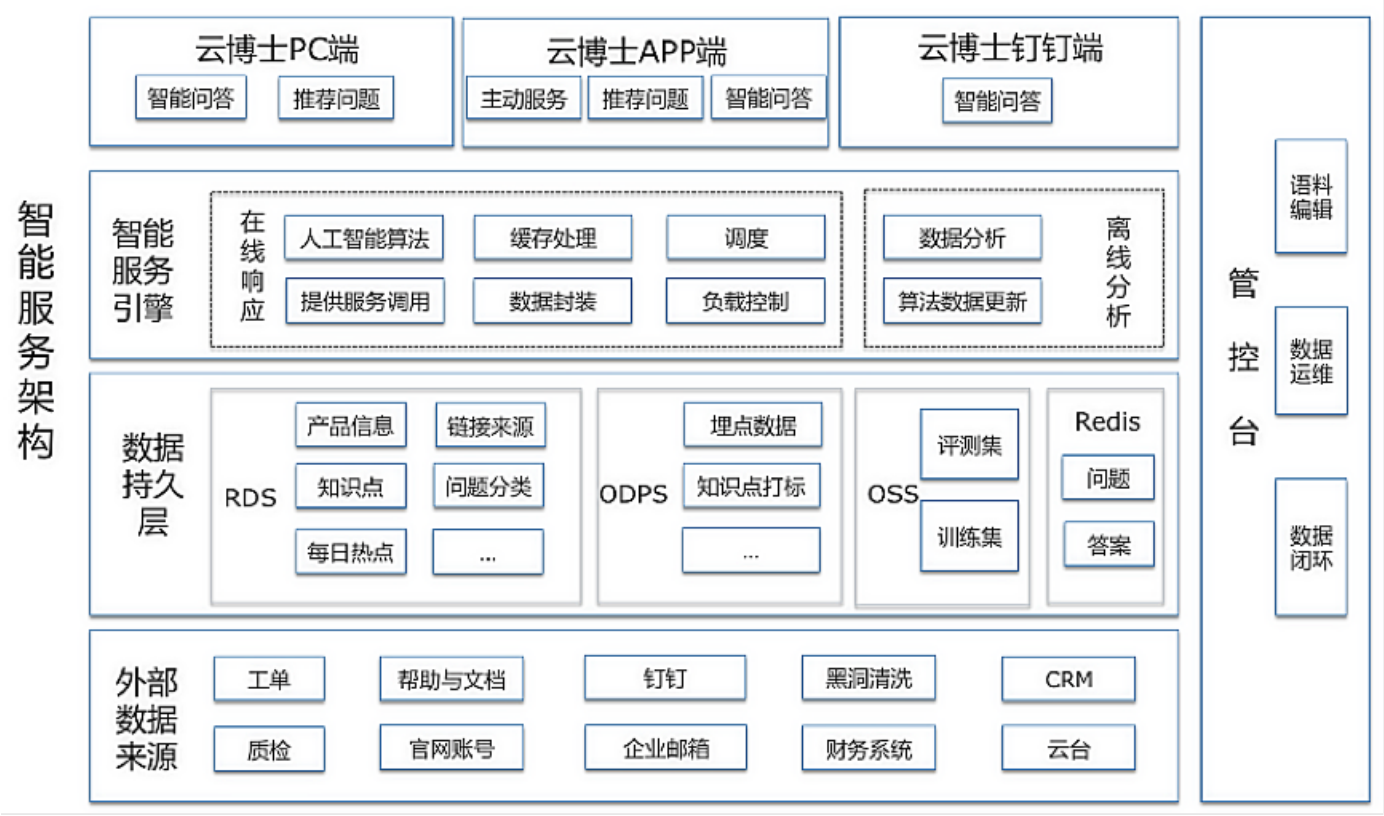
在2016年年初，阿里云服务团队和ET人工智能团队合作，启动了云博士机器人的项目，旨在用人工智能技术提升用户的服务体验。去年11月，阿里云的智能服务机器人正式对外提供服务。

云博士（Dr. Cloud）的官网地址为https://drcloud.aliyun.com，主要功能是通过大数据分析来猜测用户可能遇到的问题并给出解决方案，通过人工智能的技术提供秒级的在线智能问答服务。



二、技术架构

云博士技术架构主要分为四层：portal层、智能服务引擎层、数据持久层及外部数据来源层。同时有管控台支持云博士的语料编辑、数据运维与数据闭环。



portal层：portal层是用户与云博士进行交互的界面，包含PC端、阿里云APP端与钉钉端。一方面用户可以通过PC和阿里云APP两个途径和云博士进行便捷交互，另一方面可以将把云博士作为机器人加入到了钉钉群，群内所有用户只要在群里@云博士就可以开启提问模式，也可以在钉钉上直接和云博士进行一对一的会话。

智能服务引擎层：智能服务引擎层是云博士处理所有重要逻辑的核心层，同时提供接口供portal层调用，使得portal层只需要关注input和output。引擎层主要分为在线响应和离线分析两部分，包涵几个重要的模块：

•人工智能算法：用户输入问题后，引擎层调用阿里云ET的问答引擎获取相应答案，根据业务需要对备选答案进行再次处理和答案可信度计算，最后依据设置的可信度阈值来判断答案是否能解决用户的问题。

•缓存处理：为了保证高并发情况下的秒级应答体验，云博士使用缓存机制提升性能。在流量比较大的时候，云博士把问答输入和输出存入缓存，之后来自用户的问题如果命中缓存中的问答输入，云博士会把缓存中对应的答案直接返回给用户。

•数据封装：把从数据库和调用外部服务获取的各种数据进行封装，输出给portal层。

•调度：portal层通过调用HSF（类似Dubbo服务）接口，把请求分发到引擎层不同的服务器中实现调度均衡。其中job调度利用DTS（阿里自研的任务调度服务，类似Quartz）和ODPS（大数据计算服务）来运行相应的job。

•负载控制：当流量陡然增大时，为了防止服务超负荷运转出现问题甚至宕机，引擎层还做了负载控制，在超过系统承受范围时做限流，保证服务能够正常运转。

•提供服务调用：系统通过HTTP和HSF接口为portal层提供服务调用。

•数据分析：把云博士所有问答记录取出进行分析，以帮助算法和数据的优化。

•算法数据更新：云博士需要通过不断地进行算法优化和新数据的训练，才能变得更加智能，引擎层会每天不断地推送新的数据去训练相应的模型。

数据持久层：是云博士进行数据持久化的架构层，数据主要以四个方式持久化：

•RDS：即MySQL数据库，主要存放的是需要经常使用，对于读写实时性要求高的数据。

•ODPS：一般存放的是离线任务执行后生成的数据，对于读写的实时性要求并不高。离线任务生成的部分数据会同步到RDS，

供引擎层使用。

•OSS：主要存放训练算法模型的训练集数据。

•Redis：主要存放缓存数据，比如为了提升性能的问答输入和答案输出。

外部数据来源层：调用外部系统获取相应的数据使得云博士更好的进行状态判定和问题排查，比如调用账单系统查看用户有没有因为欠费而导致服务处于异常状态，调用黑洞清洗查看用户实例有没有处于黑洞中等等。

除了上面的四层之外，云博士的管控台主要有如下几个功能：

•语料编辑：供人工智能训练师编辑语料。

•数据运维：实时监控云博士的各种指标，如准确率、PV、UV、响应时间和一些业务指标。

•数据闭环：提升云博士问答准确率，如处理用户提交的对于云博士答案的反馈，进行数据挖掘等。

三、问答流程

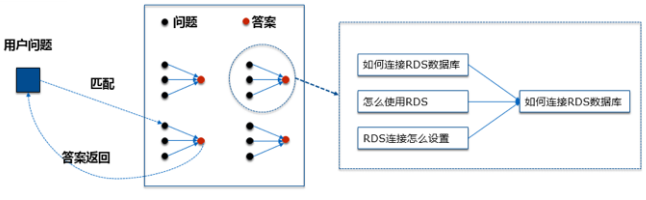
在云博士底层有一套完整的知识管理系统，所以云博士引入了知识点的概念。在云博士的系统中，知识点主要包括以下部分：

•知识点标题：如 “如何连接RDS数据库？”

•扩展问句集： 因为不同的用户会有不同的问法，有些问法之间算法相似度并不高，需要算法把同一个知识点不同问法都放入扩展问句集中。

•答案：知识点中的问题解决方案。

•产品域：知识点所属产品。

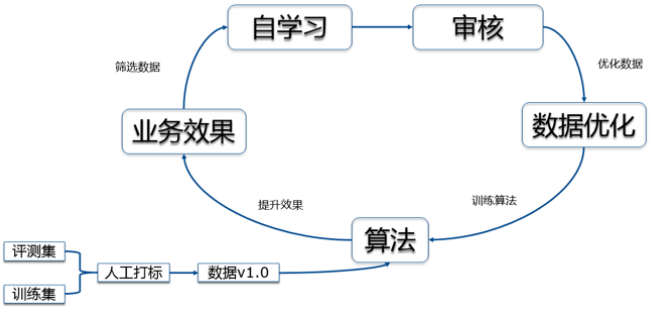


简单来说，用户问题会和训练集中的知识点标题和扩展问句进行相似度计算，把相似度高的知识点作为答案返回。

四、数据闭环

准确率是云博士的一个极其重要的核心指标，直接影响云博士的业务效果，提升准确率也是云博士一项贯穿始终的重要工作。

云博士第一版上线之后，通过阿里云售后工程师打标数据获取到V1.0版的数据去训练算法，然后根据云博士线上的业务效果，自动筛选效果差的数据，同时把不能返回答案的问题进行聚类，通过人工审核、修改扩展问句和新增知识点等方式，对数据进行了优化，形成闭环，从而提升业务效果。



目前，云博士以智能问答机器人的形式秒级响应用户的问题，还在用户的工单排队期间提前提供建议的解决方案，提升用户的体验。更为重要的是，通过客户的反馈，云博士用强化正确结果和调整负面评价结果的方式持续提升对云计算的理解，实现自学习和成长。最终，阿里云希望通过应用人工智能技术，真正为用户提供极致的服务体验。

“主板零食第一股”来伊份的逆生长

遣雨
阿里云服务运营专家

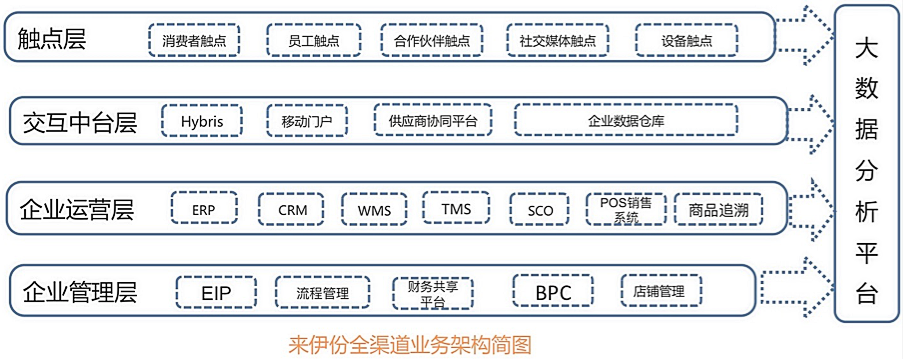
来伊份，这家创立于1999年的企业，而今的定位是集“线下门店＋线上电商＋移动支付app”一体化O2O全渠道休闲食品品牌和销售平台运营商。在门店销售继续领跑的同时，线上电商和自有app强势增长。在今年“618大促”中，来伊份邀请了40位人气主播直播泳装狂欢趴，当晚订单成交量力压同类品牌，被评论“花式营销成618黑马”。其背后，部署在云上的电商系统支撑着黑马的稳健冲杀。

要知道，曾几何时，来伊份信奉口口相传的口碑营销，在2008年之前没有做过广告。曾几何时，因为一次躺枪式的事件而影响了首次IPO，线上营销遭遇零食电商品牌的挑战。曾几何时，由于IT硬件复杂度较高，来伊份的上云之路也曾经挑战重重。

零食帝国的信息化之路

作为传统的零食帝国，来伊份的信息化大厦随着其发展而逐步建立。从2003年开始，来伊份在企业内部初步建立了信息化系统，随后，SAP、门店管理系统、会员卡系统、绩效分析系统、门户等系统逐步建立。随着业务的高速发展，预测补货系统、自动派车系统、门店关系系统、WMS等开始逐步引入。而随着互联网的风潮涌动，来伊份又启动了全渠道融合的过程，BI、CRM等系统具有了云化的可能性和特征。

互联网化的业务，意味着全渠道营销、大型促销活动、快速多变的业务诉求，也意味着对系统的稳定性、扩展性的更高要求，对基础架构的扩展性、灵活性的巨大挑战。来伊份也自然而然地开始考虑云计算服务。



来伊份全渠道业务架构简图

寻找云计算的最优解

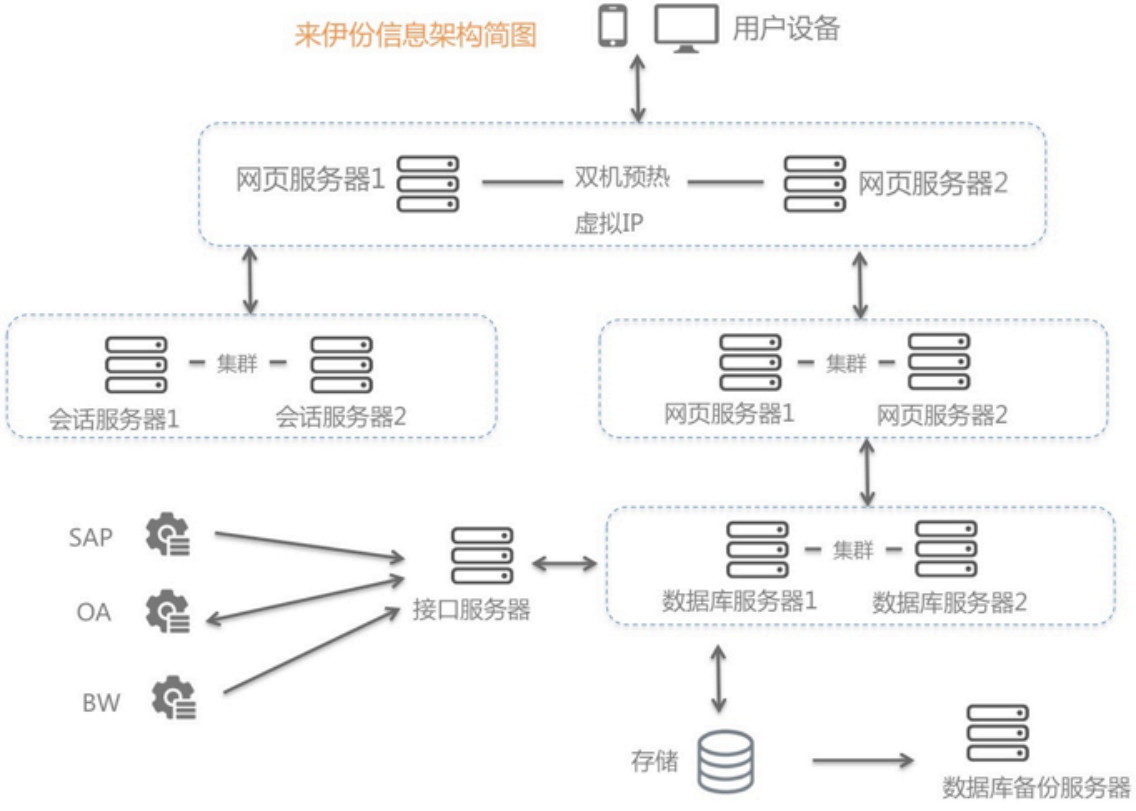
来伊份的线下部署的多个系统在此时看上去有些“重“，且IT硬件复杂度较高，拥有各类小型机、机架服务器、刀片服务器、存储、带宽等硬件资源。究竟应该如何部署和使用云计算，优先选择哪些系统、采用何种路径上云，才是对来伊份来说的最

优选择呢?

来伊份选择了阿里云，同时选择了阿里云的区域服务合作伙伴驻云科技来提供上云服务。从上云的“亲和度“、”难度“入手，对自己的多个信息化系统进行了一一评估，选择出高云亲和力、低难度和风险的系统。

采购成本分析系统的上云是来伊份上云进行尝试的第一步。驻云科技为来伊份提供了整套网络基础设计（IPSec VPN）和系统部署架构设计。该系统基于阿里云VPC部署，通过点对点隧道VPN与线下的PAC等系统进行对接，解决了云上云下系统互通的问题。

采购成本分析系统之后，电商系统上云成为了重要一步。来伊份在线下已经有一套网上商城系统，基于对双11等业务场景的巨大压力的深刻体会，希望将其部署在云计算平台上，来支撑双11的业务压力。驻云为这个系统提供了完整的云上部署方案。



接着，帮助来伊份在短时间高效完成域名备案，完成企业数据中心至阿里云的专线搭建和调试、联合阿里云提供云上安全体系建议和安全方案，驻云科技用其丰富、专业的云计算经验，帮助来伊份这个零食航母顺利驶入了云计算的海洋。

开启云&大数据的未来时

云上体验，确立了来伊份使用云计算的信心。来伊份希望通过“动态混合云”的架构，让业务负载有最适合的基础架构进行支撑，在不增加资产投资的情况下满足周期性容量需求，最大化现有IT的投资回报，快速扩展计算能力，在速度与风险间间实现平衡。

来伊份相信，通过不断探索，智能物流、自动化仓库、大数据分析、精准营销、无人值守门店…….云+大数据，会带给来伊份更大的想象空间。通过与阿里云、驻云的共同努力，来伊份的系统架构正在持续优化，为互联网背景下的业务转型打下坚实的基础。

从传统企业到互联网转型的观念转变，从传统IT架构到动态混合云的发展，从传统品牌营销策略到互联网化的品牌营销策略，来伊份这样的传统企业正在“逆生长”——迎接互联网的挑战，抓住云计算、大数据带来的机会。

与其他具有变局思维的传统企业一起，来伊份，拥抱的是这个时代。

飞利浦中国数据中心整体迁移上云

马柯
阿里云技术专家

上个月底，飞利浦（中国）投资有限公司（下文称飞利浦）在苏州的区域数据中心整体搬迁至阿里云。这家驰骋在医疗保健、照明和优质生活领域的“百年老店”，结束了数据中心时代，将以更为灵活的资源调配能力、更为快捷的新应用部署能力、更为深刻的数据洞察能力，推进其创新化的战略。

飞利浦大中华区IT运营总监王坚强表示：“把数据中心整体迁移上云，是一个里程碑式的成功。未来飞利浦智能化产品和解决方案的数据存储和处理需求将通过云计算实现。”

积极引入移动互联、云计算与大数据等技术，帮助中国社会应对城市化发展、人口老龄化等带来的挑战，是飞利浦在中国的重要战略方向之一。基于这样的战略方向，飞利浦自身也在不断转型，在医疗健康领域，飞利浦希望运用创新的数字化技术参与慢性病防治和分级诊疗建设，提供从产品、软件到以专病为基础的整体解决方案。比如，通过建立区域影像中心和专病临床数据库，就可以实现患者信息共享、远程会诊，为当地医生提供手术指导、诊断决策支持，从而提高整个医疗服务体系的效率。类似这样的整体解决方案的开发需要IT基础设施的支撑，而基于云平台来做方案实施除具备灵活、便捷等优势外，还可以将云上提供的大数据、人工智能平台作为基础设施快速融入业务系统。在通过可穿戴设备和互联移动设备获取海量数据的时代，充分发挥人工智能和大数据的的整合、分析、预测、决策支持作用，实现健康解决方案的全面创新，云计算的优势不容忽视。

经过近年来的飞速发展，云计算在存储、计算和网络服务上的上的高性能、高稳定性、高安全性、和低成本，在金融、科技等各行各业不断缔造优秀的案例。更为重要的是，想要行驶在科技创新的快车道，云计算是更为清洁高效的能源——云计算不仅能够以轻资产的形式间接降低成本，更能够让飞利浦具备更快部署新应用的能力，获得对海量数据更深刻的洞察能力。

谈到上云选择，王坚强表示：“飞利浦希望在中国找到一家不光有很强大的基础设施、在中国国内市场有很深的渗透率，也要有清晰的海外发展战略和甚至海外节点；不光有高性能的网络，也要有强大的安全治理和隐私保护的合作伙伴。除了考虑数据中心规模、产品体系的竞争力、发展速度，飞利浦非常看重云服务商是不是也同样具备创新的基因——引入能够共同创新的的伙伴，这是飞利浦的传统。”实际上，早在2014年飞利浦和阿里巴巴就签订了一项IT基础设施服务框架协议，阿里云将为飞利浦提供云计算服务。王坚强谈到：“在此前文件云项目的合作中，阿里云基于飞利浦的需求，不断丰富平台应用，带来了多元化的应用体验，这种共同成长的伙伴式的合作模式，正是我们想要的。”

在包括混合云架构构建、安全架构、业务连续性、企业级云管理服务的能力与资质等一系列技术评审后，双方的系统规划师将系统规划要求做了对标，找到可操作的技术方案并形成项目规划。2017年2月，飞利浦苏州数据中心正式启动向阿里云的迁移。阿里云驻场飞利浦，共同技术攻坚，高效推进项目，在四个月的时间内，完成了一整套数据中心的顺利搬迁上云。

借助阿里云在云计算和大数据方面的业界经验，飞利浦希望与阿里云一起来共同推动数据化价值的提升，让云计算支持大数据，支持人工智能的达成，使飞利浦的创新更好更快地惠及消费者。阿里云云市场提供丰富的即插即用的应用，可以帮助飞利浦的云上系统快捷部署新应用，阿里云的大数据产品，也能帮助飞利浦深入地整合分析海量数据。目前，飞利浦和阿里云大在IT运营监控大屏、无线投屏技术方面的合作已经开始。未来，飞利浦还将与阿里云在云客户、云桌面、直播点播视频服务等SAAS服务，以及研发云、分布式中间件等云服务方面进行新的联合创新。

售前电话

95187-1

阿里云合作伙伴 & 渠道接洽

010-65985888-16506 / 16787

channel@list.alibaba-inc.com

订阅ASDN杂志

阿里云手机APP

手机阿里云APP，满足您随时随地触达阿 里云的需求。您可以购买，监控产品数据，接收报警，瞻仰大牛技术分享，联系客服等。

云栖社区

云栖社区是由阿里云负责运营、阿里巴巴技术协会和阿里巴巴集团各技术团队提供内容支持的开放式技术社区。

云市场

阿里云云市场是阿里云生态落地的最后一公里，通过全方位赋能，帮助合作伙伴拓展商业，打造软件交付和交易第一平台。

阿里云

订阅ASDN杂志

网站：www.aliyun.com　　联系人：pengfei.pf@alibaba-inc.com　　地址：北京朝阳区望京街道阿里中心望京A座