

阿里云迁移至AWS方案设计与实施文档

# 修订历史

## 修订历史记录

下表显示了本文档的修订历史。

1. 文档修订历史

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 版本 | 作者 | 详细信息 |
| 2020年3月24日 | 1.0 | 韩思捷/梁风飚/郁磊/薛召兵 | 文档结构确定 |
| 2020年3月31日 | 2.0 | 韩思捷 | 调整结构，与PPT一致 |
|  |  |  |  |
|  |  |  |  |

Table of contents

[修订历史 2](#_Toc36640227)

[修订历史记录 2](#_Toc36640228)

[阿里云迁移到AWS项目概述 5](#_Toc36640229)

[项目描述 5](#_Toc36640230)

[完成标准 5](#_Toc36640231)

[相关联系人 5](#_Toc36640232)

[文档简介 6](#_Toc36640233)

[目的 6](#_Toc36640234)

[涵盖范围 6](#_Toc36640235)

[迁移前的调研 7](#_Toc36640236)

[概述 7](#_Toc36640237)

[MRA调研 7](#_Toc36640238)

[迁移规划与设计 9](#_Toc36640239)

[概述 9](#_Toc36640240)

[着陆区(Landing Zone)概述 9](#_Toc36640241)

[着陆区 - 账号管理 10](#_Toc36640242)

[着陆区 – 应用架构规划 11](#_Toc36640243)

[着陆区 – VPC设计 11](#_Toc36640244)

[着陆区 – 安全设计 13](#_Toc36640245)

[着陆区 – 权限管理 15](#_Toc36640246)

[着陆区 – 计算和存储资源 17](#_Toc36640247)

[迁移项目的实施 19](#_Toc36640248)

[实施步骤 19](#_Toc36640249)

[建立VPN连接 19](#_Toc36640250)

[迁移wordpress应用服务器 22](#_Toc36640251)

[迁移阿里云SLB的负载均衡器的转发规则 31](#_Toc36640252)

[MySQL数据库迁移 31](#_Toc36640253)

[Redis迁移 32](#_Toc36640254)

[对象存储迁移 33](#_Toc36640255)

[系统割接及验证 34](#_Toc36640256)

[AWS技术关注点及相关预案 35](#_Toc36640257)

[概述 35](#_Toc36640258)

[性能方面的技术关注点 35](#_Toc36640259)

[弹性方面的技术关注点 35](#_Toc36640260)

[安全方面的技术关注点 36](#_Toc36640261)

[监控方面的技术关注点 37](#_Toc36640262)

[服务限制方面的技术关注点 37](#_Toc36640263)

[AWS平台网站部署技术细节Checklist 39](#_Toc36640264)

[概述 39](#_Toc36640265)

[EC2相关 39](#_Toc36640266)

[S3相关 41](#_Toc36640267)

[数据库/Cache相关 42](#_Toc36640268)

[ELB相关 42](#_Toc36640269)

[Service Limit相关 43](#_Toc36640270)

[其他 44](#_Toc36640271)

# 阿里云迁移到AWS项目概述

## 项目描述

本次迁移项目主要针对客户在阿里云上部署的应用系统，迁移到AWS平台上的迁移计划和实施所进行的技术支持，本次项目主要协助客户进行以下范围的实施和保障：

* 迁移前的调研和规划
* 对AWS账号进行设计实施
* 对AWS VPC网络进行设计，规划以及实施
* 对AWS平台的安全配置进行设计和实施
* IAM用户和角色的设计
* 应用系统从阿里云ECS何SLB迁移到EC2和ELB，以及相关的安全配置和部署
* 数据库从阿里云RDS MySQL迁移Amazon RDS
* 缓存服务从阿里云Redis迁移到Amazon Elasticache

对象存储从阿里云OSS迁移到Amazon S3，以及S3 bucket的权限管理

项目具体的实施时间为2020年3月9日到2020年3月30日。前两周密集到场支持，后续依照运行情况进行调整。

## 完成标准

前期对项目提供基于AWS的技术及架构建议，重要日程对业务系统提供现场值守保障，并及时快速提供AWS技术需求响应。最终保障系统的顺利上线以及上线后的正常运行。

## 相关联系人

本次项目的相关联系人的信息如下:

1. 相关联系人

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **公司** | **姓名** | **电话** | **角色** | **电子邮箱** | **负责人** |
| XXX | XXX | 12345678901 |  |  | 是 |
| XXX | XXX | 12345678901 |  |  | 否 |
| XXX | XXX | 12345678901 |  |  | 否 |
| AWS | XXX | 12345678901 | SA |  | 是 |
| AWS | XXX | 12345678901 | SA |  | 否 |
| AWS | XXX | 12345678901 | SA |  | 否 |

# 文档简介

## 目的

本文档的目的是用来描述客户在2020年3月9日到2020年3月30日期间所实施的迁移项目。

## 涵盖范围

本文档主要描述了以下的内容：

* 迁移前的调研
* 迁移准备和规划
* 迁移项目的具体实施
* 项目上线后的阶段性保障

# 迁移前的调研

## 概述

在迁移项目之前，AWS会进行迁移评估(Migration Readiness Assessment (MRA))，MRA会从业务、迁移项目计划、技能、着陆区、整体系统的信息完整性、迁移经验、运维模式以及安全合规这几个角度对客户进行问题调研，从而评估客户迁移项目到AWS的能力以及决心，并且明确那些需要在迁移规划阶段解决的问题。

## MRA调研

MRA以问题调研的形式，对需要迁移的系统进行整体的了解。相关的问题包括以下大类：

1. 业务
   1. 业务预算：客户是否已经明确了针对的迁移预算(其中包括客户需要投入的资源以及在AWS上产生的用量)？
   2. 明确需要迁移的业务系统：客户是否已经明确定义了需要迁移的业务系统？本次迁移的范围是否已经文档化，并且已经和该业务系统的所有的责任人进行了沟通？
2. 迁移项目计划
   1. 确定交付模式和方法：客户是否已经确定了实施迁移的合作方和方法？
   2. 项目管理能力：客户是否具有成熟的项目管理能力？使用的项目管理方法论是什么？客户是否具有非常详细的云迁移的计划？是否已经和该应用相关的团队进行过沟通？
3. 技能
   1. AWS云上技能基线：客户团队是否具有AWS认证？
   2. 培训：客户是否已经确定需要团队成员了解的AWS云上技能？是否有AWS的培训计划？
4. 着陆区
   1. AWS账号设计和配置：客户是否具有进行迁移的AWS账号？以及对账号是如何设计的？
   2. 现有的网络和基础架构：客户当前系统的网络架构是否有文档记录？
5. 整体系统的信息完整性
   1. 应用系统信息完整性：客户是否具有完整的文档对应用系统的架构以及数据流进行描述？
   2. 服务器和基础架构的信息完整性：客户是否具有完整的文档对基础架构以及服务器进行详细的描述？
6. 迁移经验
   1. 迁移系统的经验：客户是否具有迁移系统到云上的经验？
7. 运维模式
   1. 当前的运维模式：客户对该应用系统是否具有明确的SLA？
   2. 将来的需求：将来在AWS上的运维方式会和当前的方式一样吗？
   3. AMI/补丁：目前是否具有自动打补丁的功能？
   4. 备份：对迁移到AWS以后的备份操作，是否会利用AWS的一些存储的优势，比如S3？
   5. 资产管理：目前是否可以显示账号下所部署的所有资源的信息？是否对对AWS的资源定义了标签策略？
   6. 配置管理和变更管理：是否具有流程对配置和变更进行管理？
   7. 业务连续性/容灾：是否具有业务连续性和容灾的规划，从而在AWS的某些服务出现问题的时候，进行修复？
   8. 成本管理：目前是如何管理整个基础架构的成本的？
8. 安全合规
   1. 安全参考架构：目前的安全策略是怎么做的？
   2. 日志和监控：如何收集日志以及如何监控日志并报警？

# 迁移规划与设计

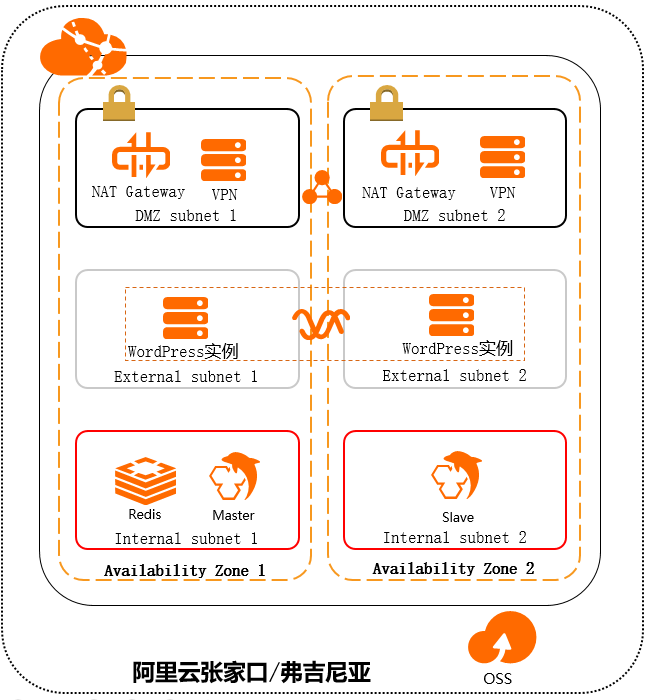
## 概述

根据迁移前的调研，在评估了客户对于迁移来说所具有的能力以后，并且会确认客户在进行迁移前的一些问题以及不足之处，这些问题和不足就需要在迁移准备和规划阶段进行解决。

目前确认需要完成的、与迁移相关的工作包括：

* 从阿里云的张家口region和弗吉尼亚region迁移到AWS Ningxia region以及AWS N. Virginia region。
* 阿里云VPC与AWS VPC通过VPN打通
* 把阿里云的Wordpress在线系统迁移到AWS平台。

下图描述了应用系统在阿里云平台上的架构图。阿里云的部署中，VPC CIDR为192.168.0.0/16。

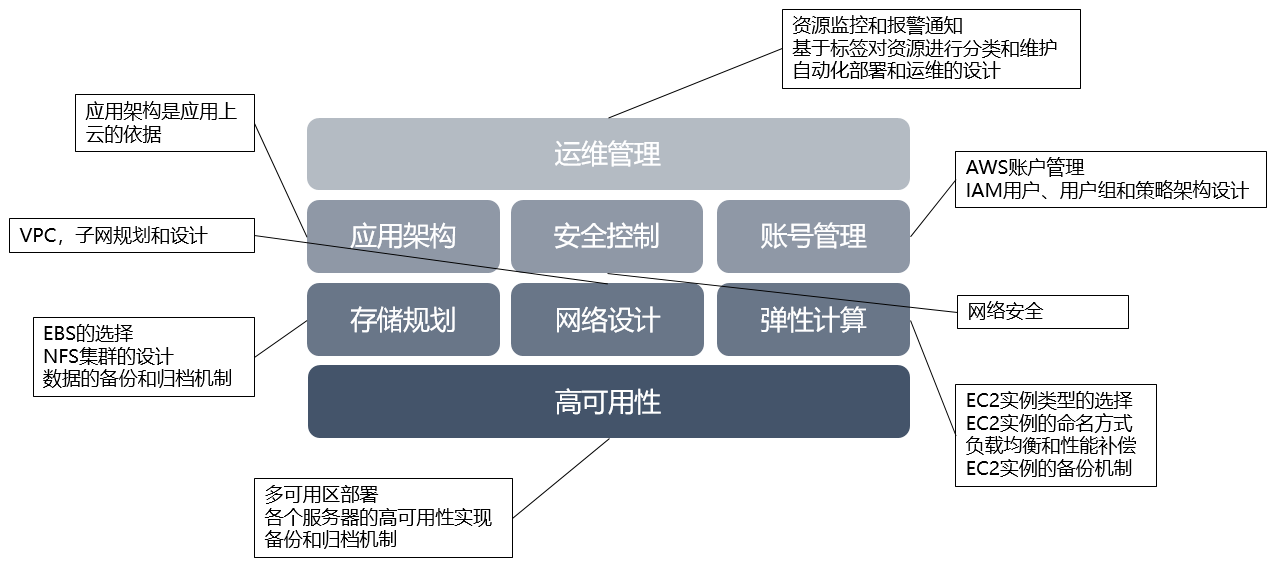


1. 阿里云上的基础设施架构图

## 着陆区(Landing Zone)概述

在准备和规划阶段，AWS会基于AWS的着陆区(Landing Zone)的思路，对迁移上AWS的系统进行架构设计。使用着陆区的目的，是为了基于AWS的最佳实践，来帮助企业构建安全的AWS多帐户环境，这个着陆区的设计，也是客户应用程序迁移的起点，基于Landing Zone构建的环境是一个允许随着时间的推移进行不断迭代和扩展的。

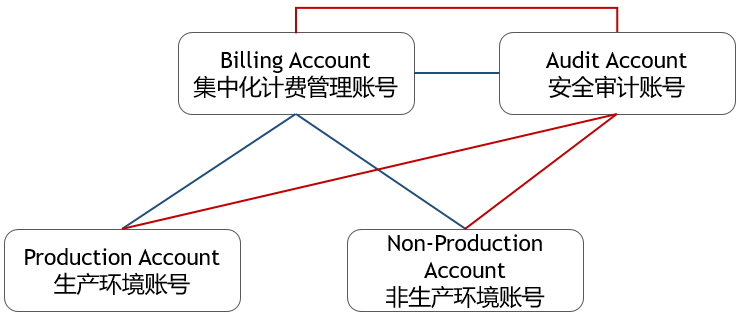
在基于着陆区的设计中，会从下图所示的这些不同的方面进行考量：



1. Landing Zone的设计理念

## 着陆区 - 账号管理

为了帮助客户部署在AWS上的系统具有更大的伸缩性，从第一步创建账号开始，就应该考虑多账号部署。如下图所示：



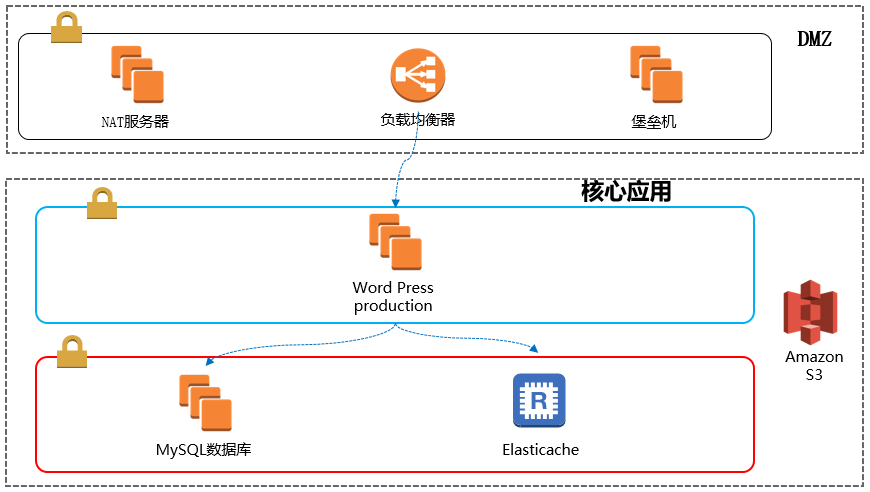
1. Landing Zone的多账号设计

在该账号设计中，创建以下4个账号：

* Billing Account：集中化计费管理账号，其他部署了系统资源的账号所产生的费用都会汇总到该账号上，并使用该账号进行付费。
* Production Account：生产环境账号，部署生产环境的账号，只有系统运维团队才有登录该账号的权限。
* Non-Production Account：非生产环境账号，用于部署开发测试环境，开发人员能够登录该账号。
* Audit Account：安全审计账号，负责接收其他账号的审计日志，比如CloudTrail log等。

## 着陆区 – 应用架构规划

从应用层面对需要迁移的系统确认其架构，如下图所示：



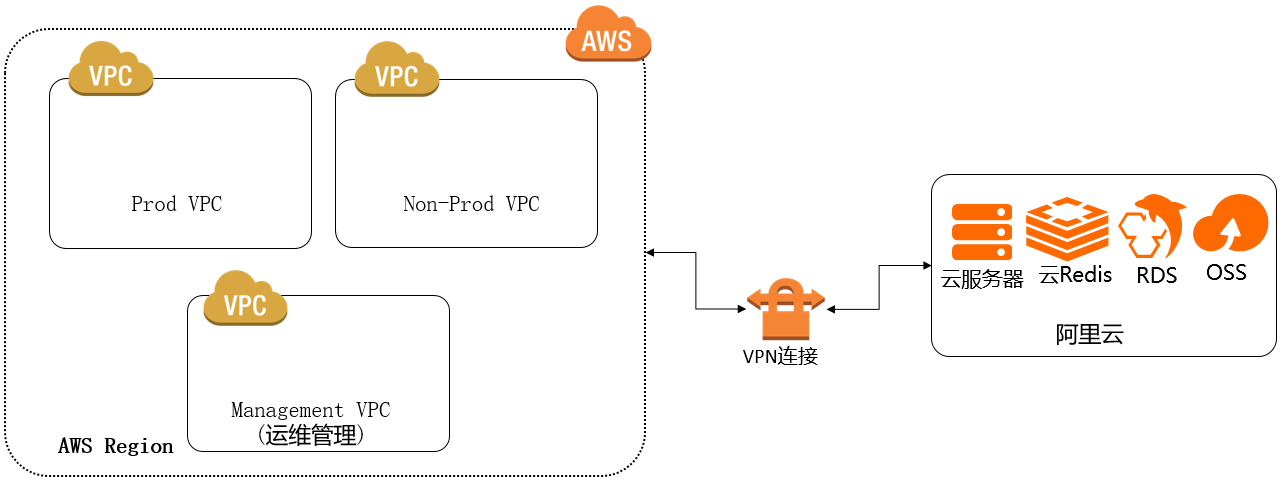
1. 应用系统架构

这是一个典型的三层Web应用的架构，包括：

* 前端用户接入层为负载均衡器，放在公网子网内。
* 中间是应用服务器集群，负责处理用户的请求，放在私有子网内。
* 后端是数据库以及缓存，放在私有子网内。
* 通过Amazon S3存放需要长期保存的数据，比如日志、图片等。

## 着陆区 – VPC设计

从应用层面对需要迁移的系统确认其架构，如下图所示：

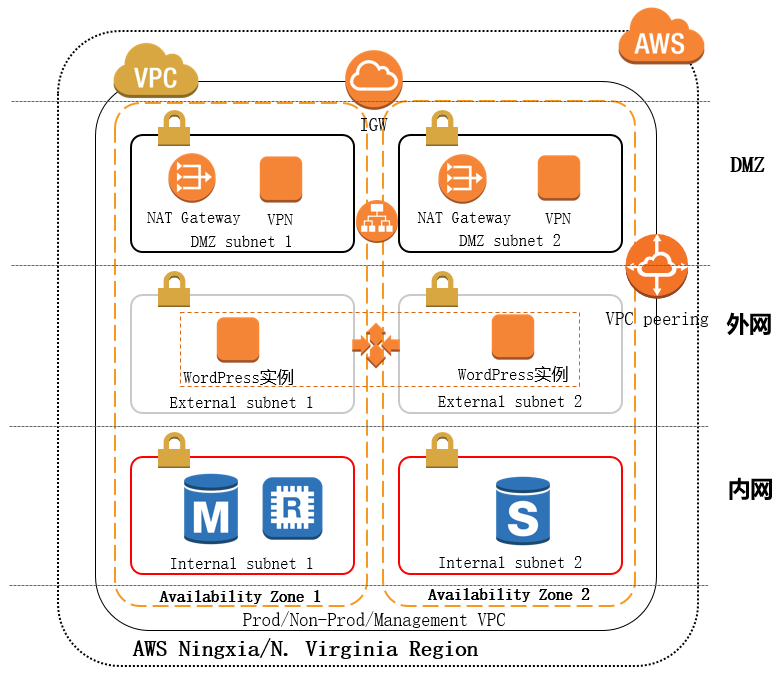


1. VPC设计

阿里云VPC到AWS VPC的连接通过使用OpenSWAN软件搭建VPN连接，从而建立通道。在VPC设计中，主要考虑下面的因素：

* Prod VPC：部署生产环境的资源
* Non-Prod VPC：如果没有部署生产账号和非生产账号的话，则需要部署Non-Prod VPC，用于容纳非生产环境，包括开发、测试以及其他比如准生产环境等。
* Management VPC：当VPC数量扩张或者共享服务比如AD服务器、堡垒机、运维管理、安全管控组件等增加时，可以单独采用共享服务VPC容纳这些服务，供多个应用系统VPC共享。共享服务VPC需与各个应用VPC采用VPC peering打通。

对Prod VPC来说，需要对子网进行设计。根据业务需求，下图描述了整个应用系统在AWS平台上的基础设施的架构图。



1. AWS上的系统架构图

在该VPC和子网的设计中，主要包含以下部分：

* VPC：创建非缺省的VPC，使用的IP地址段为：10.0.0.0/16。名称为WP-VPC-PROD。
* VPC DMZ subnet：该子网为公有子网，用于部署面向公网访问的VPN、NAT 、堡垒机和 ELB 等。子网分布在两个AZ里，名称分别为subnet-dmz-1和subnet-dmz-2，对应的IP地址段分别为：10.0.1.0/24和10.0.2.0/24。其对应的路由表为：

1. subnet-dmz所使用的路由表

|  |  |  |
| --- | --- | --- |
| **名称** | **Target** | **Target** |
| subnet-dmz-1  subnet-dmz-2 | 10.0.0.0/16 | Local |
| 0.0.0.0/0 | [igw-0cf407cb56ef3daa5](https://cn-northwest-1.console.amazonaws.cn/vpc/home?region=cn-northwest-1#igws:filter=igw-0cf407cb56ef3daa5) |

* VPC External subnet：该子网为私有子网，用于部署只允许通过 NAT 访问外网的wordpress应用服务器，以及部署AWS DMS复制实例。子网分布在两个AZ里，名称分别为subnet-ext-1和subnet-ext-2，对应的IP地址段分别为： 10.0.3.0/24和10.0.4.0/24。其对应的路由表为：

1. subnet-dmz所使用的路由表

|  |  |  |
| --- | --- | --- |
| **名称** | **Target** | **Target** |
| subnet-dmz-1  subnet-dmz-2 | 10.0.0.0/16 | Local |
| 0.0.0.0/0 | [igw-0cf407cb56ef3daa5](https://cn-northwest-1.console.amazonaws.cn/vpc/home?region=cn-northwest-1#igws:filter=igw-0cf407cb56ef3daa5) |

* VPC private subnet：部署只允许内部访问且无外网访问需求的MySQL RDS实例以及Elasticache实例。子网分布在两个AZ里，名称分别为subnet-priv-1和subnet- priv -2，对应的IP地址段分别为：对应的IP地址段为：10.0.5.0/24和10.0.6.0/24。其对应的路由表为：

1. subnet-priv所使用的路由表

|  |  |  |
| --- | --- | --- |
| **名称** | **Target** | **Target** |
| subnet-priv-1  subnet-priv-2 | 10.0.0.0/16 | Local |
| 192.168.0.0/16 | VPN EC2的网卡 |

## 着陆区 – 安全设计

本次项目在安全方面的设计，主要集中在VPC子网设计和网络层面的安全组设计。在AWS上，可以采用多层的防御安全保护基础架构：

1. 控制哪些实例暴露给公网
2. 定义NACL：类似于子网上的防火墙，控制所有子网层面上的进出流量，需要根据实际需求，指定白名单和黑名单。
3. 定义安全组：类似于实例上的防火墙，控制所有实例层面上的进出流量，只需要指定允许放心的流量即可。

基于客户的业务需求，设置的安全组规则如下：

* VPC DMZ subnet里的public ELB的安全组设置为：

1. Public ELB的安全组

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **协议** | **端口范围** | **允许的数据来源** |
| SG-WP-Public | TCP | 80, 443 | 0.0.0.0/0 |

* VPC DMZ subnet里的public ELB的安全组设置为：

1. 堡垒机的安全组

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **协议** | **端口范围** | **允许的数据来源** |
| SG-WP-Bastion | TCP | 22 | 0.0.0.0/0 |

* VPC external subnet里的EC2所使用的安全组的设置为：

1. Wordpress应用服务器EC2的安全组

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **协议** | **端口范围** | **允许的数据来源** |
| SG-WP-EC2 | TCP | 80, 443 | SG-WP-Public |
| TCP | 22 | SG-WP-Bastion |
| ALL | ALL | 192.168.0.0/16 |
| ALL | ALL | SG-WP-VPN-EC2 |
| ALL | ALL | SG-WP-EC2 |

* VPC private subnet里的RDS所使用的安全组的设置为：

1. RDS的安全组

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **协议** | **端口范围** | **允许的数据来源** |
| SG-WP-DB | TCP | 3306 | SG-WP-EC2 |
| ALL | ALL | SG-WP-DMS |

* VPC private subnet里的Elasticache所使用的安全组的设置为：

1. Elasticache的安全组

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **协议** | **端口范围** | **允许的数据来源** |
| SG-WP-Elasticache | TCP | 6379 | SG-WP-EC2 |

* VPC DMZ subnet里的OpenSWAN VPN EC2所使用的安全组的设置为：

1. OpenSWAN VPN EC2的安全组

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **协议** | **端口范围** | **允许的数据来源** |
| SG-WP-VPN-EC2 | TCP | 22 | SG-WP-Bastion |
| UDP | 4500,500 | 39.98.193.226/32 |
| ALL | ALL | 10.0.0.0/16 |

这里的39.98.193.226是阿里云上部署的Openswan的虚拟机的公网ip地址。

* VPC private subnet的DMS复制实例的安全组设置为：

1. DMS复制实例的安全组

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **协议** | **端口范围** | **允许的数据来源** |
| SG-WP-DMS | ALL | 0-65536 | SG-WP-DB |
| ALL | 0-65536 | 192.168.0.0/16 |

## 着陆区 – 权限管理

在本次项目的权限管理上，主要集中在IAM权限设置，以及对S3 bucket和MySQL RDS数据库的访问控制。

目前客户的Wordpress系统在运维时，先使用各自的用户名和密码登录到堡垒机，然后在堡垒机上对应用系统进行配置。

因此对于AWS来说，在IAM里不会为这些终端用户创建账号，因此主要在IAM里除了管理员用户以外没有其他普通用户，而通过创建各个不同的角色，并把各自不同的角色绑定到不同的应用服务EC2上，从而能够让EC2去访问S3资源。

根据业务需求，创建如下的IAM角色：

1. IAM角色

|  |  |  |
| --- | --- | --- |
| **角色名称** | **权限** | **所属的业务用途** |
| R-S3-RW | 读写S3 bucket | 用于在应用程序中对静态资源进行读写操作 |

比如R-S3-RW角色所具有的权限可以为：

{

"Version": "2012-10-17",

"Statement": [

{

"Sid": "Accessslefdirecotry",

"Effect": "Allow",

"Action": [

"s3:\*"

],

"Resource": [

"arn:aws-cn:s3:::wp-resource","arn:aws-cn:s3::: wp-resource/\*"

]

}

]

}

这些角色绑定到wordpress运行的EC2的实例上，从而能够从这些EC2实例里去访问对应的资源。

对于RDS数据库的访问控制来说，通过堡垒机或者应用服务EC2，能够对数据库进行读写操作。更细粒度的权限控制则在数据库层面完成。通过在数据库里为各个不同的终端用户创建各自的账号，并只给这些账号赋予只读的权限，从而这些终端用户在使用自己的账号连接到数据库以后，只能读取数据库，而不能对数据库进行写入操作。

对于Elasticache的访问控制来说，通过堡垒机或者应用服务EC2，能够对数据库进行读写操作。

对Amazon S3的访问控制来说，首先根据业务需求，确定需要使用的bucket以及相关的命名方式。目前需要使用如下的bucket：

1. S3 bucket

|  |  |  |
| --- | --- | --- |
| **Bucket名称** | **权限** | **所属的业务用途** |
| wp-resource | 只允许启用了角色R-S3-RW的EC2访问 | 存放外部机构的日志 |

其次，针对不同的bucket设置权限。比如wp-resource bucket上的策略可以设置为如下内容：

{

"Id": "Policy1475133430367",

"Version": "2012-10-17",

"Statement": [

{

"Sid": "Stmt1475133427887",

"Action": [

"s3:\*"

],

"Effect": "Allow",

"Resource": [

"arn:aws-cn:s3::: wp-resource",

"arn:aws-cn:s3::: wp-resource/\*"

],

"Principal": {

"AWS": [

"arn:aws-cn:iam::898111891143:role/R-S3-RW"

]

}

}

]

}

## 着陆区 – 计算和存储资源

下表展示了在本次项目中所用到的AWS各项资源的清单。

1. 使用的实例资源列表

|  |  |  |  |
| --- | --- | --- | --- |
| **服务器** | **实例类型** | **数量** | **操作系统** |
| 应用服务器 | C5.large | 2(Auto Scaling) | Amazon Linux |
| RDS数据库服务器 | M4.xlarge | 1 | MySQL 5.7 |
| Elasticache缓存服务器 | R4.large | 1 | Redis |
| VPN服务器 | C5.large | 1 | Amazon Linux |
| 堡垒机服务器 | C5.large | 1 | Amazon Linux |

1. 使用的块存储列表

|  |  |  |  |
| --- | --- | --- | --- |
| **服务器** | **EBS类型** | **容量(GB/台)** | **IOPS** |
| 应用服务器 | Magnetic | 100 | N/A |
| RDS数据库服务器 | (GP2) | 500 | 1500 |
| VPN服务器 | Magnetic | 50 | N/A |
| 堡垒机服务器 | Magnetic | 50 | N/A |

1. RDS数据库设计

|  |  |
| --- | --- |
| **条目** | **配置信息** |
| 引擎 | MySQL |
| 部署方式 | Multi-AZ |
| License Model | License-included |
| 数据库引擎版本 | 5.7.22 |
| 数据库实例类型 | db.m4.xlarge |
| 存储 | 100GB |
| 自动备份频率 | 每天 |
| 自动备份保留时间 | 7天 |

1. Elasticache数据库设计

|  |  |
| --- | --- |
| **条目** | **配置信息** |
| 引擎 | Redis |
| 部署方式 | 单节点 |
| 引擎版本 | 5.0.6 |
| 实例类型 | db.m4.xlarge |
| 自动备份频率 | 每天 |

# 迁移项目的实施

## 实施步骤

本次迁移项目计划于3月9日开始实施，包括从阿里云张家口机房迁移到AWS宁夏region，以及从阿里云弗吉尼亚机房迁移到AWS N. Virginia region。具体实施的步骤包括：

1. 实施步骤

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **步骤名称** | **完成时间** | **责任人** |
| 1 | 部署VPC和安全组 |  |  |
| 2 | 创建RDS MySQL实例 |  |  |
| 3 | 创建Elasticache实例 |  |  |
| 4 | 建立VPN连接 |  |  |
| 5 | 迁移wordpress应用服务器 |  |  |
| 6 | 迁移阿里云SLB的负载均衡器的转发规则 |  |  |
| 7 | MySQL数据库迁移 |  |  |
| 8 | Redis迁移 |  |  |
| 9 | 对象存储迁移 |  |  |
| 10 | 系统割接及验证 |  |  |

## 建立VPN连接

实施步骤中的1、2和3，即部署VPC和安全组、创建RDS MySQL实例以及创建Elasticache实例，请参考AWS官方文档。

VPN的配置信息如下表所示。

1. VPN的配置信息

|  |  |  |
| --- | --- | --- |
| **VPN组件** | **阿里云VPC** | **AWS VPC** |
| CIDR | 192.168.0.0/16 | 10.0.0.0/16 |
| VPN实例的EIP | 39.98.193.226 | 52.83.126.30 |

具体配置步骤包括：

1、AWS上的VPN实例操作系统为Amazon Linux，并禁用source/destination检查。

2、AWS的VPN实例的安全组如Table 16所示。

3、阿里云的VPN实例的操作系统为Centos 8.0，不要选安全加固。

4、为阿里云的VPN实例的安全组添加下图所示的规则：



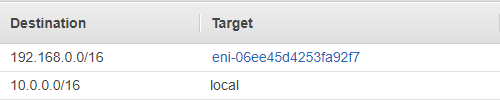
1. 阿里云的VPN实例的安全组

5、在阿里云的私有子网里添加一条路由条目，目标网段为AWS的VPC CIDR，下一跳为OpenSwan虚拟机的实例id，如下图所示：



1. 阿里云的私有子网路由表

6、在AWS的私有子网里添加一条路由条目，Destination为阿里云VPC的CIDR，Target为OpenSwan EC2，如下图所示：



1. AWS的私有子网路由表

7、在阿里云的OpenSwan虚拟机上执行以下的操作：

1) yum install -y epel-release

2) yum install -y libreswan

3) yum install -y python2

4) ln -s /usr/bin/python2 /usr/bin/python

5) vi /etc/ipsec.conf

5.1) 确保include /etc/ipsec.d/\*.conf前面没有注释符

5.2) 确保logfile=/var/log/pluto.log 前面没有注释符

6) vi /etc/ipsec.d/nettonet.conf，并添加如下的内容

conn nettonet

authby=secret

auto=start

leftid=39.98.193.226 <--阿里云VPN虚拟机的公网ip

left=%defaultroute

leftsubnet=192.168.0.0/16 <--阿里云VPC CIDR

leftnexthop=%defaultroute

rightid=ZHY

right=52.83.126.30 <--AWS VPN虚拟机的公网ip

rightsubnet=10.0.0.0/16 <--AWS VPC CIDR

keyingtries=%forever

ike=aes128-sha1;modp1024

ikelifetime=86400s

phase2alg=aes128-sha1

salifetime=3600s

pfs=no

7) vi /etc/ipsec.d/nettonet.secrets，并添加如下的内容

39.98.193.226 52.83.126.30: PSK "aws123"

这里的aws123表示密钥密码，可以是任何值。

8) vi /etc/sysctl.conf，并添加如下内容：

net.ipv4.ip\_forward = 1

net.ipv4.conf.all.accept\_redirects = 0

net.ipv4.conf.all.send\_redirects = 0

net.ipv4.conf.default.send\_redirects = 0

net.ipv4.conf.eth0.send\_redirects = 0

net.ipv4.conf.default.accept\_redirects = 0

net.ipv4.conf.eth0.accept\_redirects = 0

9)运行sysctl -p从而启用新的配置

8、在AWS的OpenSwan EC2里执行类似的操作：

1) yum install -y openswan

2) vi /etc/ipsec.conf

2.1) 确保include /etc/ipsec.d/\*.conf前面没有注释符

2.2) 确保logfile=/var/log/pluto.log 前面没有注释符

3) vi /etc/ipsec.d/nettonet.conf，并添加如下的内容

conn nettonet

authby=secret

auto=start

leftid=52.83.126.30 <--AWS VPN虚拟机的公网ip

left=%defaultroute

leftsubnet=10.0.0.0/16 <--AWS VPC CIDR

leftnexthop=%defaultroute

rightid=ALI

right=39.98.193.226 <--阿里云VPN虚拟机的公网ip

rightsubnet=192.168.0.0/16 <--阿里云VPC CIDR

keyingtries=%forever

ike=aes128-sha1;modp1024

ikelifetime=86400s

phase2alg=aes128-sha1

salifetime=3600s

pfs=no

4) vi /etc/ipsec.d/nettonet.secrets，并添加如下的内容

52.83.126.30 39.98.193.226: PSK "aws123"

这里的aws123表示密钥密码，必须和阿里云VPN虚拟机的配置完全一样。

5) vi /etc/sysctl.conf，并添加如下内容：

net.ipv4.ip\_forward = 1

net.ipv4.conf.all.accept\_redirects = 0

net.ipv4.conf.all.send\_redirects = 0

net.ipv4.conf.default.send\_redirects = 0

net.ipv4.conf.eth0.send\_redirects = 0

net.ipv4.conf.default.accept\_redirects = 0

net.ipv4.conf.eth0.accept\_redirects = 0

6)运行sysctl -p从而启用新的配置

9、在阿里云和AWS的两个VPN实例上启动OpenSwan并检查配置：

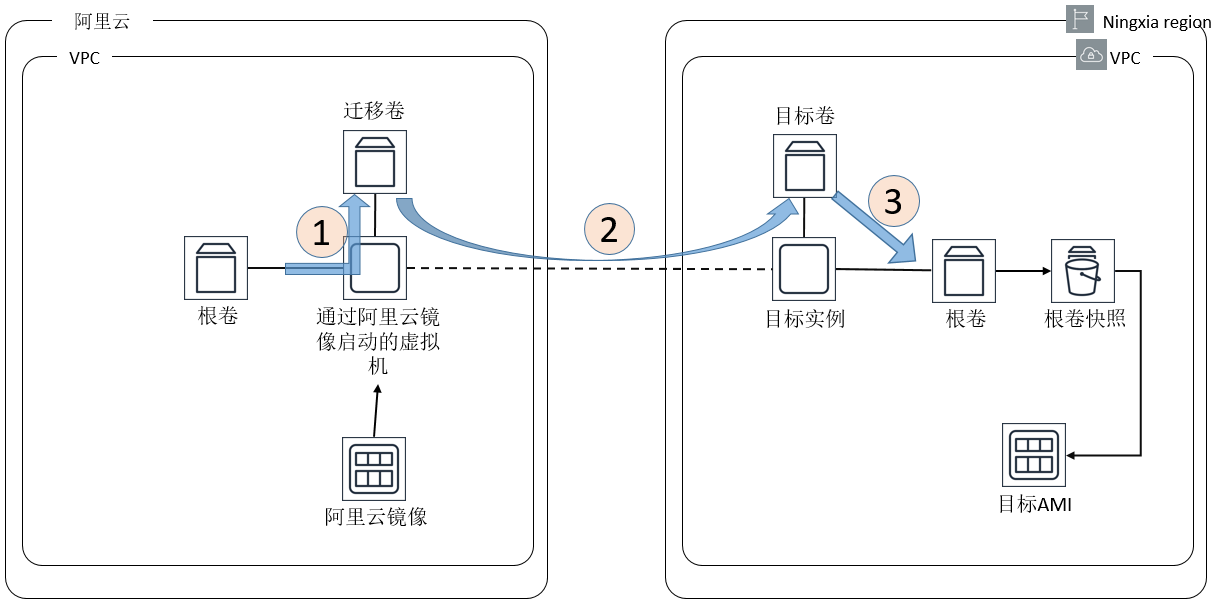
systemctl start ipsec

10、运行ipsec verify命令确认OpenSwan运行正常。

## 迁移wordpress应用服务器

对于迁移wordpress应用服务器来说，可以有两种方式来完成：

1. 对于从阿里云张家口机房迁移到AWS Ningxia region来说，截至到2020年3月，CloudEndure还没有在Ningxia region上线，因此可以通过使用dd硬盘的方式进行迁移。下图显示了使用dd硬盘的方式对wordpress应用系统所在的虚拟机进行迁移的过程。



1. 使用dd方式迁移虚拟机流程

在该过程中：

1. 根据当前应用系统所使用的最新的阿里云镜像（假设该镜像使用的是Centos 7.5），部署一台虚拟机。
2. 修改阿里云虚拟的网络配置。

vi /etc/sysconfig/network-scripts/ifcfg-eth0，添加如下内容

DEVICE=eth0

BOOTPROTO=dhcp

ONBOOT=yes

TYPE=Ethernet

USERCTL=yes

PEERDNS=yes

DHCPV6C=yes

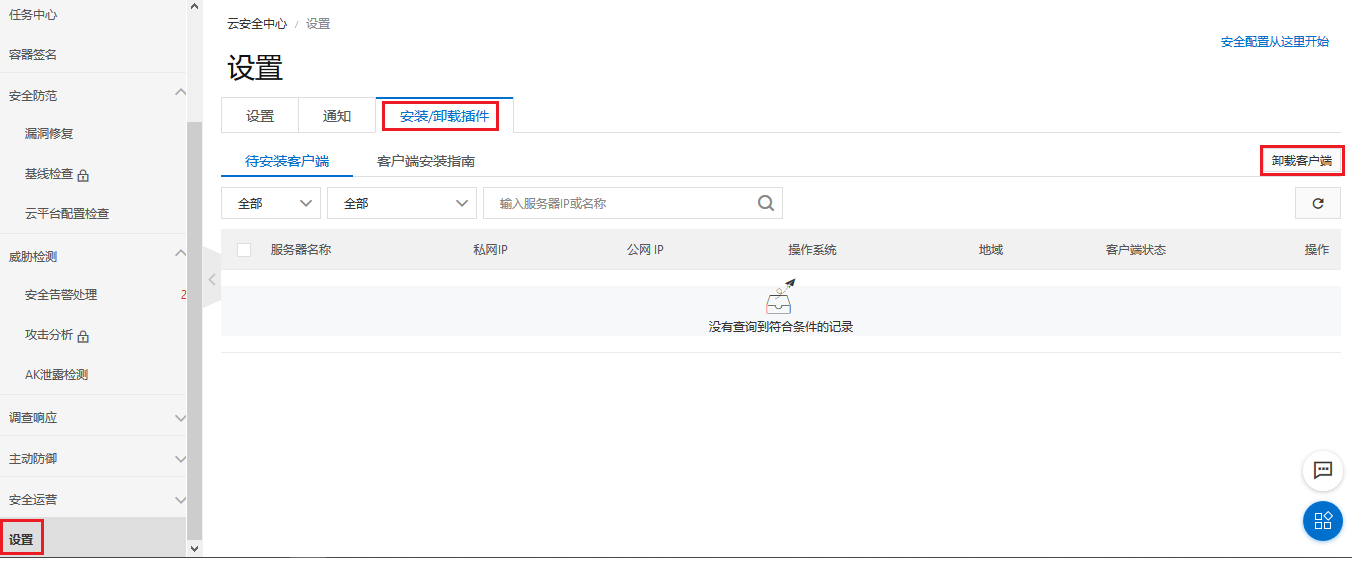
DHCPV6C\_OPTIONS=-nw

PERSISTENT\_DHCLIENT=yes

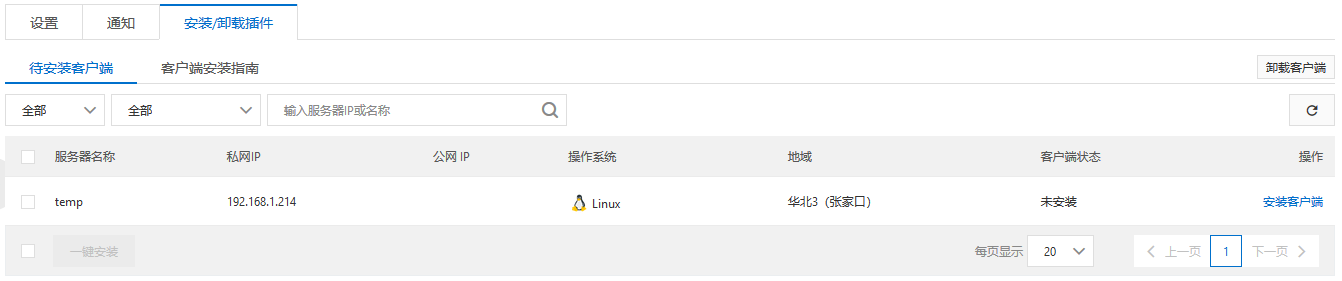
RES\_OPTIONS="timeout:2 attempts:5"

DHCP\_ARP\_CHECK=no

1. 卸载阿里云安骑士：登录云盾服务器安全（安骑士）管理控制台，单击左侧导航栏设置 > 安装/卸载 。如下图所示。



1. 阿里云虚拟机上卸载云盾
2. 选择卸载客户端，直至未安装安骑士的虚拟机出现在列表中。如下图所示。



1. 阿里云虚拟机上卸载云盾
2. 检查fstab文件，确保无其他挂载卷。
3. 更新软件包及YUM源设置：

yum update -y

mv /etc/yum.repos.d/CentOS-Base.repo /etc/yum.repos.d/CentOS-Base.repo\_bak

yum remove epel-release

wget -O /etc/yum.repos.d/CentOS-Base.repo http://mirrors.163.com/.help/CentOS7-Base-163.repo

yum makecache

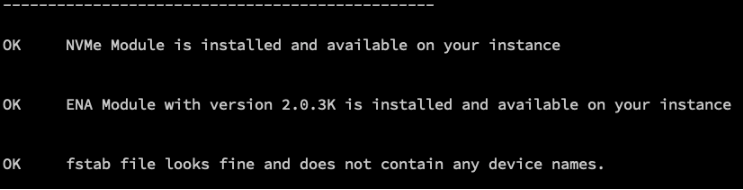
1. 下载并运行AWS 5系检查脚本：

wget https://raw.githubusercontent.com/awslabs/aws-support-tools/master/EC2/NitroInstanceChecks/nitro\_check\_script.sh

chmod +x nitro\_check\_script.sh

./nitro\_check\_script.sh

如下图所示则说明检查通过。如遇到无ENA驱动，参考此文章进行安装：https://aws.amazon.com/cn/premiumsupport/knowledge-center/install-ena-driver-rhel-ec2/



1. 确认ENA驱动成功安装
2. 将阿里云虚拟机启动卷写入文件

8.1) 在阿里云创建新卷并挂载至虚拟机

8.2) 为新卷创建文件系统：

mkfs -t ext4 /dev/vdb

8.3) 挂载：

mount /dev/vdb /mnt

8.4) 用dd命令把根卷（假设wordpress应用安装在根卷）写入文件：

dd if=/dev/vda of=/mnt/centos.bin bs=100M

1. 把centos.bin传输到AWS的S3上，或者也可以选择先压缩再传。
2. 将centos.bin文件写入目标EBS根卷

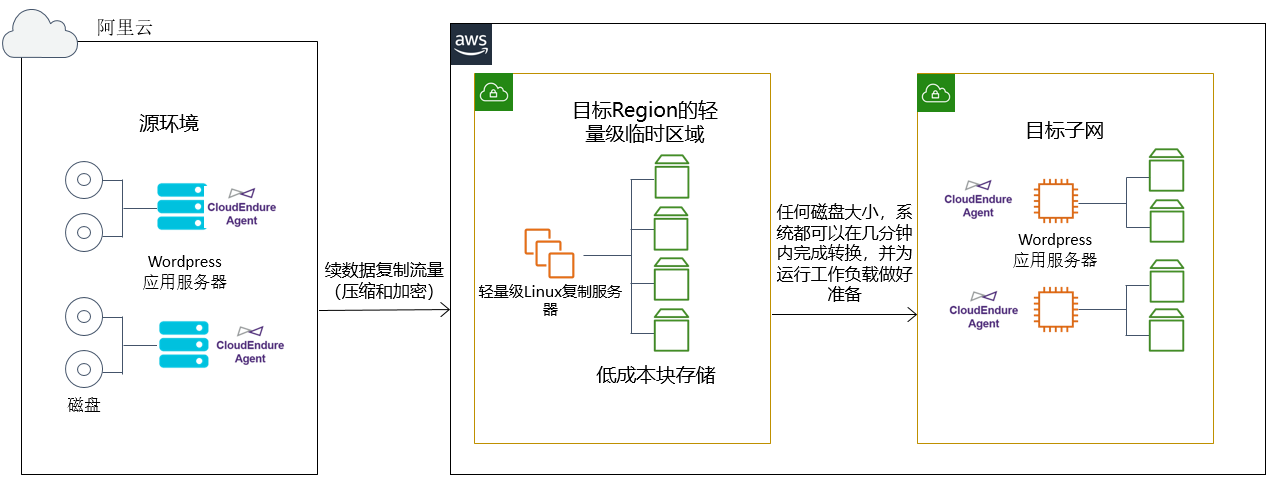
10.1) 在目标实例上创建并挂载EBS根卷，容量不小于阿里云虚拟机的启动卷。

10.2) 执行dd命令，把centos.bin文件写入EBS根卷（假设是/dev/xvdg）

dd if=centos.bin of=/dev/xvdg bs=1M oflag=direct

umount /dev/xvdg

1. 对EBS根卷创建快照，并使用该快照来创建AMI镜像
2. 使用创建的AMI镜像启动虚拟机，部署在私有子网里。可使用原用户名以及key文件进行登录。
3. 对于从阿里云弗吉尼亚迁移到AWS N. Virginia region来说，通过使用CloudEndure进行迁移。其过程如下图所示。



1. 使用CloudEndure复制虚拟机

在该过程中：

1. 按照前面描述的方式，在阿里云的应用服务器上卸载阿里云安骑士。
2. 在Marketplace订阅Cloudendure，打开https://aws.amazon.com/marketplace/pp/B01MQCH96B 进行订阅，并按照提示完成Cloudendure 注册。
3. 创建项目：

使用注册的帐户登录到Cloudendure Console: <https://console.cloudendure.com>

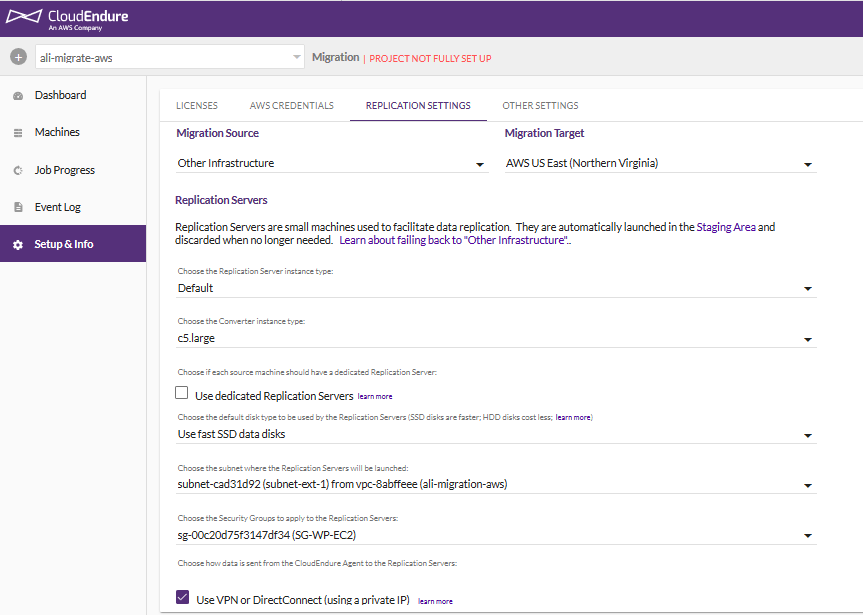
点击创建项目，填填写项目名称、类型选择Migration，并创建。

在下一步的Setup & Info中的AWS CREDENTIALS tab页中填入AWS的AK/SK访问凭据。

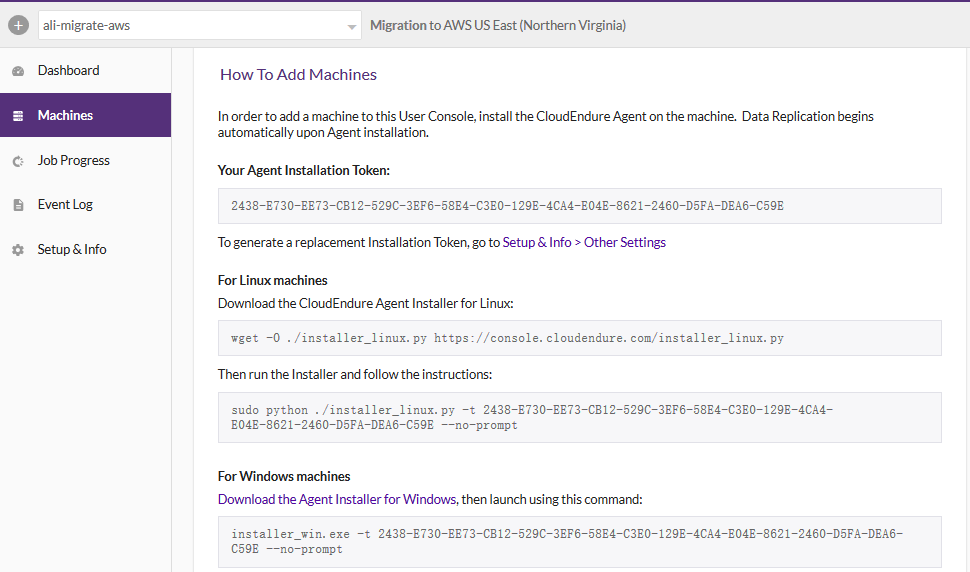
接下来在REPLICATION SETTINGS中：

* 设置Migration Source为Other Infrastructure，Migration Target为AWS Northern Virginiar。
* 在Replication Servers里选择机型类型，如果需要复制的数量量较大，可适当选择稍大一些的机型。
* 选择subnet-ext作为Replication Servers部署的子网
* 配置CloudEndure agent通过VPN连接到Replication Servers。

如下图所示，选择保存。



1. 配置REPLICATION SETTINGS
2. 安装Agent，点击左侧Machine按钮，来获取Agent安装信息。



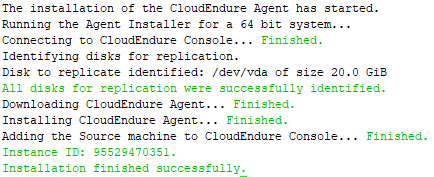
1. 安装Agent

在阿里云的wordpress虚拟机上运行命令：

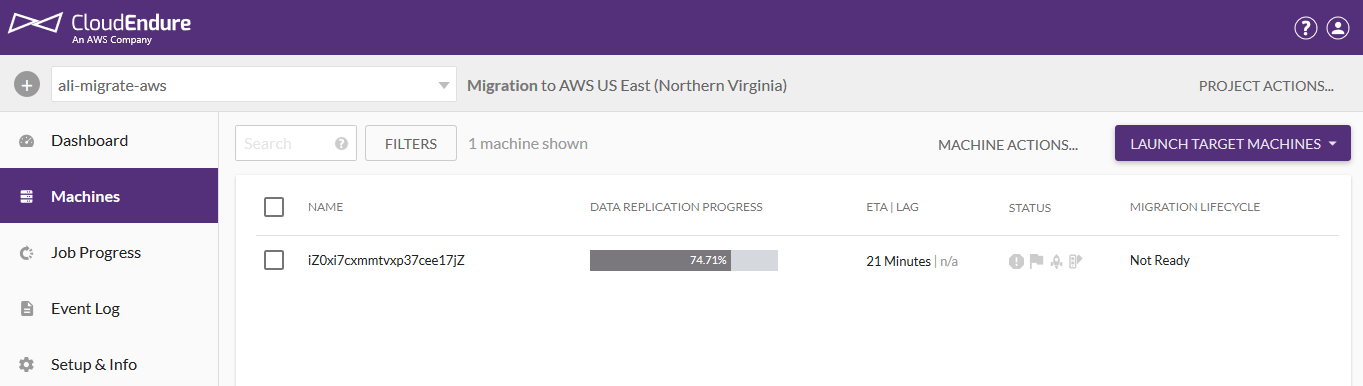
wget -O ./installer\_linux.py <https://console.cloudendure.com/installer_linux.py>

sudo python ./installer\_linux.py -t 2438-E730-EE73-CB12-529C-3EF6-58E4-C3E0-129E-4CA4-E04E-8621-2460-D5FA-DEA6-C59E --no-prompt

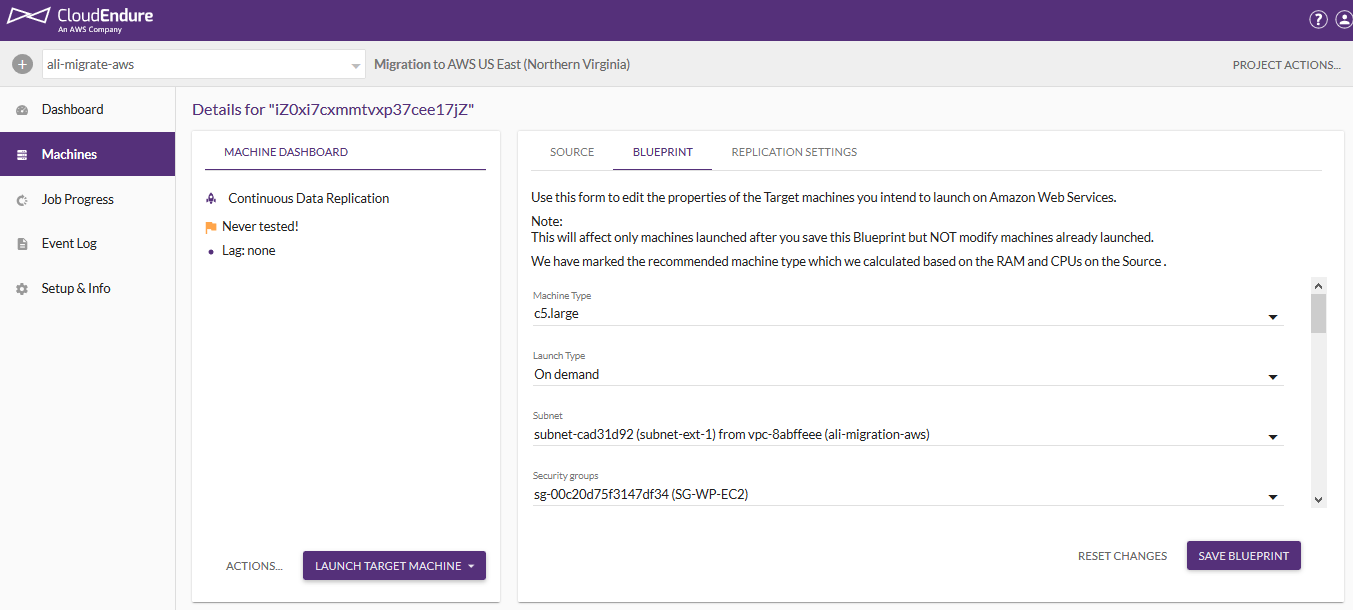
如果看到下面的输出，则说明安装成功。



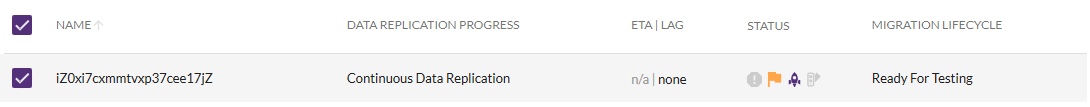
1. 成功安装Agent的输出
2. Agent安装完成后，要复制的内容会出现在控制台上。需要等待任务初始化，并创建Replication Server EC2实例后，复制将会自动开始。如下图所示。预计完成复制的时间由当前在阿里云上的OpenSwan虚拟机到internet的网络带宽（该带宽如果使用了弹性IP，则带宽最高可以增加到200Mb/s，并且不需要重启实例）、AWS平台上的复制实例的类型等决定。



1. 成功安装Agent的输出
2. 双击该虚拟机，进行Blueprint的设置，也就是为该实例在AWS上的EC2实例配置信息，如实例类型、网络配置等等。对我们这里的项目来说，该EC2实例部署在subnet-ext网络中，并指定实例类型为c5.large。确定以后，选择保存。如下图所示：

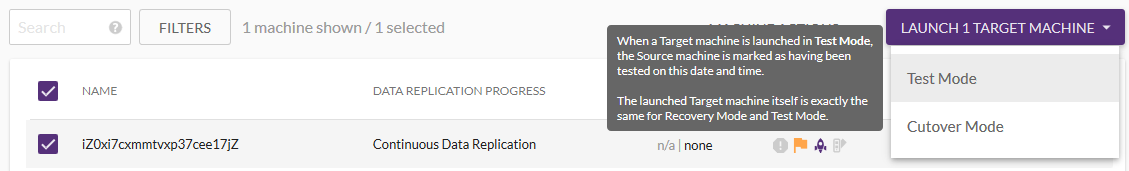


1. 设置BLUEPRINT
2. 当被复制的虚拟机的状态变为Continuous Data Replication，则完成存量数据的复制，进入变更复制阶段。此时可以启动EC2进行测试。在此阶段，需要确认迁移过后的EC2所必需的调整和配置，如：部分软件的升级、yum source、应用的重新配置等内容，并对最终的切换流程进行确认。如下图所示：



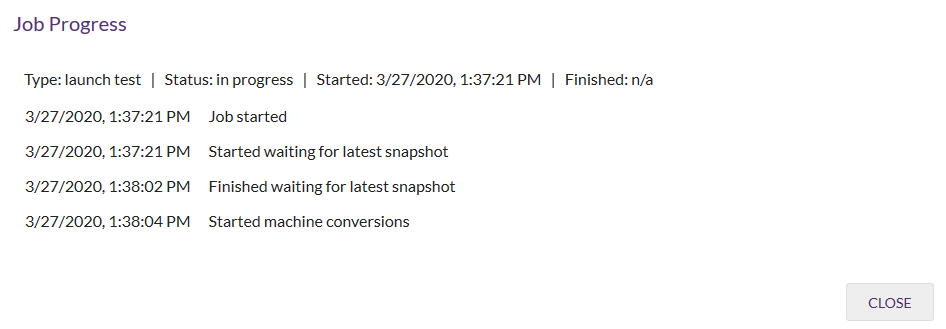
1. 虚拟机复制状态

在菜单选择Test Mode，即可执行对虚拟机的转换并创建EC2实例。



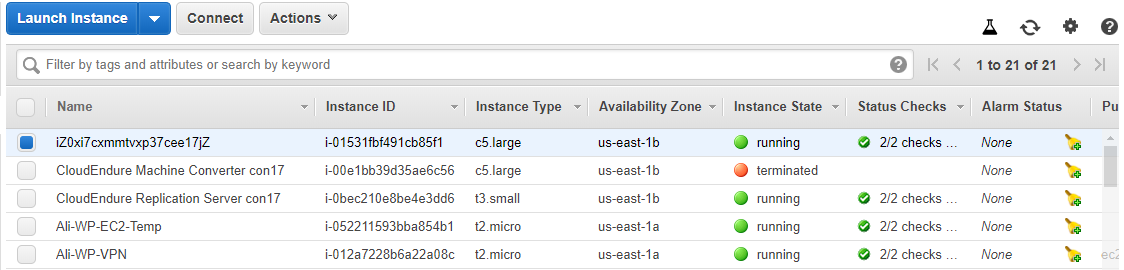
1. 运行Test Mode

可以点击右侧的Job Progress观察任务的进度。



1. Test Mode任务运行状态

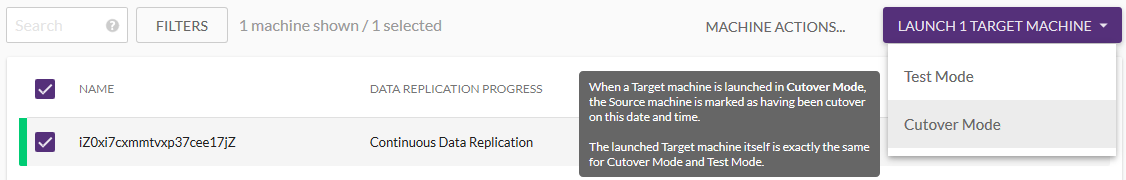
在EC2 Console中可以看到创建好的EC2实例。并且可以用阿里云上的用户名和key进行登录。



1. 在EC2 Console上确认创建好的EC2实例
2. 当测试完毕，就可以进行系统切换，按照在测试阶段制订好的切换计划和流程进行。比如：

1) 停止阿里云上的生产环境

2) 在CloudEndure Console上点击Cutover



1. 运行Cutover Mode

3) 对启动后的EC2实例进行必要的设置和调整，如：数据库RDS域名的变更等。具体参考迁移步骤中的最后一步：系统割接及验证。

## 迁移阿里云SLB的负载均衡器的转发规则

在大型生产环境中，负载均衡器上往往配置了较多的转发规则，当用户进行跨平台迁移时，且对目标平台不太熟悉的情况下，对规则的重新创建工作是一项比较繁琐的工作。我们可以通过自动化的方法，将用户以前在阿里云上创建的规则复制至AWS ELB之上，详情请见：<https://github.com/liangfb/SLBRulestoALB>。

该工具目前支持的负载均衡器类型包括：

* 阿里：SLB（七层）
* AWS：ALB

在使用该工具之前，需要满足以下条件：

* 创建阿里云AK/SK
* 创建AWS AK/SK
* 在本地电脑上安装nodejs
* 在AWS上创建好ALB和相关的TargetGroup

执行如下操作，完成负载均衡器到ALB的迁移

1. 从Github下载规则复制工具

git clone https://github.com/liangfb/SLBRulestoALB.git

cd SLBRulestoALB

npm install

2. 依次进行如下配置

2.1) 编辑config.json文件  
2.2) 修改Alicloud和AWS的AccessKey和SecretKey  
2.3) 修改Alicloud和AWS的Endpoint和区域  
2.4) 编辑映射关系：

1. SLBId: 阿里云上的SLB负载均衡Id
2. TargetGroupArn: 在AWS上对应的Target Group ARN
3. Targets: 阿里云SLB下的侦听端口和对应AWS ALB的Listener ARN
4. 注：映射关系和Targets均可配置多条

3. 运行工具：node slb.js。该命令会输出已创建的规则信息：



1. 确认SLB迁移到ALB

## MySQL数据库迁移

对数据库的迁移，可以通过AWS DMS（Database Migration Service）服务。DMS可以快速并安全地将数据库迁移至 AWS。源数据库在迁移过程中可继续正常运行，从而最大程度地减少依赖该数据库的应用程序的停机时间。AWS Database Migration Service 可以在广泛使用的开源或者商业数据库之间迁移数据。

按照标准做法即可完成数据库的迁移工作，具体如下：

1. 在subnet-ext子网里创建复制实例，同时关闭public access，以及选择SG-WP-DMS安全组。
2. 创建source endpoint和target endpoint，并测试连接是否正常。
3. 创建DMS任务，复制的时候，迁移类型选择“迁移现有数据并复制持续更改”。启用CloudWatch Logs。在选择规则部分，指定复制的数据库为wordpress，如下图所示。其他保留缺省配置，并创建任务。



1. DMS选择规则
2. 创建任务以后，启动任务，并确保任务正常运行，说明数据正常从阿里云的RDS迁移到了AWS的RDS。

## Redis迁移

由于阿里云的Redis托管服务把sync/psync命令给禁用了，因此无法通过RMT（https://github.com/vipshop/redis-migrate-tool）工具进行在线迁移。因此，现在的做法是在进行应用割接时，停止应用系统，然后使用rump工具（https://github.com/stickermule/rump）进行存量数据的迁移。

使用过程如下：

1. 安装rump：

curl -SL https://github.com/stickermule/rump/releases/download/1.0.0/rump-1.0.0-linux-amd64 -o rump && chmod +x rump;

2. 停止应用系统

3. 进行数据拷贝:

./rump -from redis://<阿里云Redis托管服务的域名>:6379/1 -to redis://<AWS Elasticache的域名>:6379/1

## 对象存储迁移

对于从阿里云对象存储OSS迁移到Amazon S3来说，可以使用AWS的开源工具进行迁移：<https://github.com/aws-samples/amazon-s3-resumable-upload>

该工具在迁移时具有如下优势：

1. 支持断点续传

2. 已经迁移的文件不会再次迁移，也就是只会迁移变化的数据

迁移过程如下:

1. 在工作机上根据文档（https://github.com/aws-samples/amazon-s3-resumable-upload）的说明，部署工作环境。其操作过程为：

yum install epel-release

yum install https://centos7.iuscommunity.org/ius-release.rpm

yum install python36u

ln -s /bin/python3.6 /bin/python3

yum install python36u-pip

ln -s /bin/pip3.6 /bin/pip3

1. 安装工具，以及安装boto3和oss2：

git clone https://github.com/aws-samples/amazon-s3-resumable-upload.git

cd amazon-s3-resumable-upload

pip install -r requirements.txt --user

1. 编辑s3\_upload\_config.py文件，主要关注的参数包括：

JobType = "ALIOSS\_TO\_S3"

SrcFileIndex = "\*" #表示所有文件

S3Prefix = "" #表示指定的OSS的bucket下所有的文件，不要被这里的参数名：S3Prefix所误导，这个参数不仅仅只是指S3，也指OSS的bucket

DesProfileName = "default"

DesBucket = "ali-migrate-demo"

ali\_SrcBucket = "your bucket name" # 阿里云OSS 源Bucket

ali\_access\_key\_id = "xxxxxxxxxxx" # 阿里云 RAM 用户访问密钥

ali\_access\_key\_secret = "xxxxxxxxxxxx"

ali\_endpoint = "oss-cn-zhangjiakou.aliyuncs.com (http://oss-cn-zhangjiakou.aliyuncs.com/)" # OSS 区域 endpoint，在OSS控制台界面可以找到

1. 执行数据同步命令：python3 s3\_upload.py

## 系统割接及验证

正式从阿里云割接到AWS的时候，通常都需要对应用程序进行一定的变更和配置。对于我们这次迁移的wordpress应用来说，则需要在AWS上进行如下的配置：

1. 登录到迁移wordpress EC2上，把RDS MySQL（包括登录用户名和密码）和Elasticache的连接字符串更新到/usr/share/nginx/html/wordpress/wp-config.php文件里。根据该EC2创建Golden AMI以后，部署Auto Scaling Group，并挂载到ALB上。
2. 因为在wordpress应用中，阿里云的源站记录的域名是写在数据库的表里的，因此同步到AWS平台以后，仍然是阿里云的源站的SLB的域名，需要改成AWS的ALB的域名。所以需要登录到AWS的RDS里，进入wordpress数据库以后，执行下面的语句更新应用系统的表：

UPDATE wp\_options SET option\_value = REPLACE(option\_value, '旧SLB域名', '新的ALB域名') WHERE option\_name = 'home' OR option\_name = 'siteurl';

UPDATE wp\_posts SET post\_content = REPLACE(post\_content, '旧SLB域名' , '新ALB域名');

UPDATE wp\_posts SET guid = REPLACE(guid, '旧SLB域名' ,'新ALB域名');

1. 由于wordpress在Elasticache的缓存里也记录了旧的SLB域名，因此也需要清空：flushall
2. 访问网站，确认应用系统正常运行。

# AWS技术关注点及相关预案

## 概述

本次项目需要注意一些技术检查点，主要分布在性能，弹性，安全，系统限制以及监控方面，对这些技术风险点需要有一定的预案。

## 性能方面的技术关注点

在性能方面，需要注意的技术关注点包括：

* 基础设施部分在设计中是否有压力点

比如在实际实施过程中，可能会发生实例的带宽不足的情况。因此AWS建议：部分AWS资源存在一些限制，例如特定的EC2实例的CPU/网络性能可能满足一般要求，但如果在运行过程中有突发较高压力出现，建议更换性能更好的实例以满足事件期间的性能要求。所以对业务核型的EC2实例，要有升级的演练。

* 应用程序部分在设计中是否有压力点

比如，在应用系统中是否会有频繁读写的数据库中的某个特定的行的情况。AWS建议：应该尽量避免类似的串行化点的设计，如果确信应用系统具有类似的设计缺陷，应认真评估在系统运行过程中流量峰值时该问题对整个业务系统的影响。

* 项目上线前是否进行过压力测试

AWS建议：应该对应用架构的压力承载能力有一个合理的压测数据统计，这样可以事件中，当发生预期以外的压力时，可以正确评估资源调整计划以及业务扩展计划。

## 弹性方面的技术关注点

在弹性方面，需要注意的技术关注点包括：

* 当前应用程序在项目实施中是否有已知的高可用性问题

AWS建议：在AWS云端设计应用架构时应采用Design For Failure的软件设计哲学，在应用架构设计中，默认应该考虑任何一个模块（包括AWS提供的基础设施模块）都可能会因为各种各样的问题而导致不可用，或者性能无法达到预期的要求。如果此类问题发生，应用系统应该有预定的应急预案，以避免或减轻某个模块的故障而对整个应用系统产生较大的影响。如果有已知的可靠性隐患，应该在项目实施前及时修复。通常在项目实施中，如果应用系统的压力超常于日常的压力范围，可能会放大可靠性隐患发生的概率，从而对整个应用平台运行产生不利的影响。

比如在本次迁移项目中，OpenSWAN VPN的设置就存在高可用性问题，一旦阿里云VPC里的OpenSWAN VPN主机崩溃，或者AWS VPC里的OpenSWAN VPN EC2实例崩溃，则阿里云VPC与AWS VPC的数据通道将被关闭。需要具有一定的预案，来快速恢复OpenSWAN VPN的配置。比如配置一台备用的OpenSWAN VPN实例，一旦主实例崩溃，则立刻使用该备用实例，并把EIP关联到该新的备用实例上。

另外，堡垒机也存在高可用性问题，一旦堡垒机崩溃则会导致业务人员无法使用系统。所以建议预先准备好堡垒机的模板，一旦堡垒机崩溃则可以快速的通过堡垒机模板来启动一个新的堡垒机。

* 是否有单点故障

比如单可用区部署等。AWS建议：采用多可用去部署应用系统，来避免由于AWS单可用去可能产生的问题而导致的业务在系统运行过程中的不可用。

* 数据保存的位置是否具有弹性

AWS建议：重要的数据应该保存在具有高持久性属性的存储上，比如S3存储桶。EBS卷的持久性低于S3存储桶，仅适合存放较频繁使用的热数据，如果在EBS卷上保存较重要的热数据，应该定时使用EBS快照功能对EBS卷本身进行备份。

* 任何内容，数据或者系统损坏的影响是什么

AWS建议：重要数据应该保持在具有高持久性属性的存储上，比如S3存储桶。如果数据非常关键，应做到多地区多副本备份方式。比如数据库定期备份，从而在发生误删除的时候，能够通过备份进行恢复。

* 一旦出现异常情况,是否有服务恢复计划

AWS建议：应该制定快速、自动化的服务恢复机制，以保证在异常时可以快速恢复业务。

* Autocaling/ELB是否使用了跨可用区的部署方式

AWS建议：一些关键服务支持跨可用区方式部署，建议开启这些默认功能。这些机制可以有效避免因AWS可用区所可能产生的问题，导致对整个应用系统产生不良影响。在开启AWS服务的多可用区部署机制的同时，应用系统也应该采用跨可用区的部署，以达到业务平台整体无单可用区现象的发生。

* 如果发现部分模块故障，或者有应用性能下降的情况出现，主环境切换到备用环境的过程是自动完成还是需要人工干预

AWS建议：一些AWS服务已经具有了自动恢复的功能，比如RDS，S3， ELB等。用户只需要对自己开发，以及自己部署的应用系统实现自动恢复的功能，就可以大大减低因部分模块故障或性能下降产生的宕机时间。

## 安全方面的技术关注点

在安全方面，需要注意的技术关注点包括：

* 最近一次的AWS KEY的更换时间

AWS建议：如果正在使用AWS KEY来管理AWS资源，则建议定期对AWS KEY进行更换，来避免KEY泄露而对整个在AWS平台上部署的应用系统产生安全隐患。如果应用系统需要访问AWS资源，比如写入S3存储桶，则AWS建议采用IAM role的方式对运行在指定EC2上的应用程序进行授权，而绝不应采用在代码中嵌入KEY的方式。

* 在系统运行过程中，如何访问系统环境

AWS建议：严格控制对整个AWS系统环境的访问权限设计。在安全组和network ACL的设计中采用最小权限赋予原则，应避免授予过大权限给到相关操作人员。应用系统所在的EC2实例之间的通讯授权也应该采取最小权限赋予原则，只对相关应用的EC2开放需要的相关权限。应避免授予0.0.0.0/0访问权限给EC2实例。如果应用服务需要为internet提供服务，应该把这个应用服务置于ELB后面，应避免直接使用EIP或Public IP开启互联网访问应用系统的方式。当管理人员需要通过互联网访问AWS VPC从而执行管理操作时，应该使用堡垒机机制，即内部资源只允许堡垒机登录，而堡垒机只允许指定的IP地址（比如办公室公网地址，或者VPN网段地址）登录。堡垒机应避免开启0.0.0.0/0的访问权限。

* 当系统故障或部分模块不可用时，如何访问环境

AWS建议: 应该建立多重访问系统的机制。例如默认允许VPN访问堡垒机，当VPN出现异常时，堡垒机也同时允许来自办公室固定公网IP的访问连接。

## 监控方面的技术关注点

在监控方面，需要注意的技术关注点包括：

* 可用于定位潜在反复发生的性能问题的关键资源

比如，数据库以及应用服务器的监控是否被开启，并且监控日志是否保存在了S3或者其他地方。AWS建议：对关键资源保持必要的监控（例如CPU使用情况，网络流量情况以及IO延时），一旦发生某些监控指标异常，可以第一时间对该异常做出反应，并且通过丰富的监控数据，也可以协助AWS的技术支持人员更快的定位问题来源，以达到尽快解决问题的目的。

* 是否设置了多种多样的性能监控警告

比如，在CloudWatch或者第三方监控工具上设置告警功能。AWS建议： 在定制相关资源监控的同时，也需要对监控的内容设置阀值报警。AWS 的CloudWatch支持对指定监控参数设置阀值（比如EC2实例的CPU使用率等），并且通过邮件方式通知到值班工程师。

* 是否有使用了内部的监控工具

如果使用了内部的监控工具，则是否可以提供权限给AWS现场工程师。

* 需要额外关注的，关键的监控信息

AWS建议：列出相关监控信息并且提前AWS现场工程师。

## 服务限制方面的技术关注点

在服务限制方面，需要注意的技术关注点包括：

* 依次核实下列资源当前限制的情况

账号，Regions，实例，EBS存储，ELBs。

* 哪些AWS服务可能会增加限制

AWS建议：在项目实施或者系统运行过程中，需要大幅增加AWS资源，应该先确认相关资源是否有软硬限制。如果有软限制（比如能够启动的EC2实例的个数），可以考虑提前申请提高该软限制。如果有硬限制，则应该在设计时避免发生触发该硬限制。

* 是否有Autoscaling groups带有会启动更多实例的自动扩展规则

AWS建议：如果使用Autoscaling groups自动启动更多实例，应该对相关实例的软限制做检查，避免因为软限制导致无法开启足够的实例以应付较大的访问流量。

* 是否有可能产生某些EBS卷吞吐量超过最大允许值的场景发生

AWS建议：正确选择EBS卷的吞吐量。如果评估后，发现有可能会产生较大的I/O活动，则应该提早更换使用更强I/O能力的EBS卷，比如PIOPS的EBS卷。

* EC2实例是否会发送电子邮件，以及发送频率

AWS建议：如果应用中有使用电子邮件的场景，建议提前告知AWS在系统实施以及运行中，最高峰时，邮件的发送频率。未报备的过高频率的邮件传输，可能会被AWS误判为垃圾邮件发送端，从而导致被AWS误禁止掉该服务。

* 系统实施已经运行过程中，是否要创建额外的S3存储桶，以及是否会碰到每个用户100个存储桶的硬限制

AWS建议：AWS对S3存储桶有硬性限制，此限制无法突破。应提前做好S3存储桶的整体规划工作。

# AWS平台网站部署技术细节Checklist

## 概述

除了以上的概述性的技术关注点以外，还需要在AWS平台上，在部署网站的时候，对常见的技术细节进行检查。

## EC2相关

1. EC2 是否开启AutoRecovery？
   1. 在EC2控制台界面，勾选实例，在”状态检查”选项中，点击”创建状态检查警报”，勾选”请执行以下操作: 恢复此实例”。开启AutoRecovery之后，当实例所在的底层硬件故障，AWS后台服务会自动将该实例stop/start后迁移到健康的底层硬件上。
   2. 有关AutoRecovery的介绍，请参考: <https://docs.aws.amazon.com/zh_cn/AWSEC2/latest/UserGuide/ec2-instance-recover.html>
2. 是否使用官方的AMI？
   1. 任何人都可以在Community中发布自己的ami，因此使用未知来源的community ami 有一定的风险，不建议您在生产环境中使用community ami.
   2. 不同的ami的登录用户名不一样，您可以参考下面的文档获取ssh登录的默认用户名:<https://docs.aws.amazon.com/zh_cn/AWSEC2/latest/UserGuide/AccessingInstancesLinux.html#AccessingInstancesLinuxSSHClient>
3. 公网访问的EC2 是否绑定EIP？
   1. 在不使用弹性IP(EIP)的情况下，如果将实例关机/开机，那么实例的公网IP地址就会发生变化。如果您想固定实例的公网IP地址，请分配EIP，并将EIP绑定到实例: <https://docs.aws.amazon.com/zh_cn/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html>
4. 是否只能通过密钥对登录Linux EC2？
   1. 默认情况下，AWS提供的Linux镜像只开启了通过密钥对方式登录，尽管您可以通过修改配置文件的方式启用密码登录，但这样会增大实例被入侵的风险，我们强烈不建议您启用密码登录。
   2. 如果您想为linux设置多个用户，您也可以在登录linux之后自行创建用户，然后手动导入key pair: <https://docs.aws.amazon.com/zh_cn/AWSEC2/latest/UserGuide/ec2-key-pairs.html>
5. VPC是否设置了公有子网和私有子网？公有子网是否添加了IGW，私有子网是否添加NAT Gateway？
   1. 如果一个子网的流量被路由到 Internet 网关，这个子网便是公有子网。如果一个子网没有通向 Internet 网关的路由，这个子网便是私有子网。
   2. 一般建议将需要在互联网上访问的资源放在公有子网(如面向Internet的Web Server)，将内部资源的服务器放在私有子网(如数据库,Cache等)。私有子网的实例如果想要访问互联网下载软件或更新应用，VPC路由中需要关联NAT网关或NAT实例。公有子网需要关联IGW.
6. 内网服务如数据库或应用程序服务器是否放在私有子网中？
   1. 由于数据库或应用程序服务器都是不直接面向最终用户的，作为最佳实践，建议放在私有子网中以加强安全性。
7. EC2 安全组是否只开放特定端口？
   1. 请检查EC2安全组，建议只开放自己部署的服务端口，尽量不要将所有流量都开放给0.0.0.0/0
   2. 对于一些特殊端口，如用于SSH的TCP 22端口，如果可能的话，建议只对特定网段开放。
8. 是否有对重要的实例EBS卷定期做快照？
   1. 您可以参考下面的文档对EBS卷定期做快照，以防止EBS卷误删导致数据丢失: [https://amazonaws-china.com/cn/blogs/china/construct-ebs-life-circle-management](https://amazonaws-china.com/cn/blogs/china/construct-ebs-life-circle-management/)
9. 在高并发场景下是否有调优Linux内核参数？
   1. 在高并发的场景下，您有可能需要对每个用户、系统进程所消耗的资源进行调优，如:
      1. ulimit -n 调节单个进程所能打开的最大文件描述符数量
      2. ulimit -u 调节用户最大可用进程数
   2. 请根据自己的业务情况进行调节，如: ulimit -n 10240将单个进程打开最大的fd数量修改为10240，调节之后再使用 ulimit -a 查看是否生效。如果要永久生效，需要修改 /etc/security/limits.conf。
   3. 在高并发的场景下，您还需要关注调节tcp backlog，backlog是指已连接但还未进行accept处理的socket队列大小，在nginx配置中，backlog为listen后的第二个参数，如: listen 80 backlog=65535.
   4. 没有任何一个具体的调优数值适合所有的场景，您需要结合自己的业务进行相应的调优。有关调优的更多信息，请参考: <https://access.redhat.com/documentation/zh-cn/red_hat_enterprise_linux/6/html/performance_tuning_guide/>
10. EBS 卷是如何选择的？EBS和实例之间的带宽是否满足需求？
    1. EBS 卷提供了gp2, io1, st1, sc1 四种类型，不同的卷类型提供的IOPS以及吞吐量均不同，一般对于数据库来说，选择gp2或io1更适合。请检查自己选择的卷的最高IOPS是否能满足业务峰值的需求。卷类型介绍: <https://docs.aws.amazon.com/zh_cn/AWSEC2/latest/UserGuide/EBSVolumeTypes.html>
    2. 实例和EBS之间的最大带宽和最大吞吐量是有限制的，对于EBS优化的机型，您可以参考下面的文档检查是否能满足您的需求: <https://docs.aws.amazon.com/zh_cn/AWSEC2/latest/UserGuide/EBSOptimized.html> 对于非EBS优化机型，实例和EBS之间的通信占用实例本身带宽，受实例最大带宽限制。
11. 是否开启了terminate保护？
    1. 对于重要的实例，建议您开启EC2终止保护，以防止误操作导致实例被意外删除:  <https://docs.aws.amazon.com/zh_cn/AWSEC2/latest/UserGuide/terminating-instances.html#Using_ChangingDisableAPITermination>

## S3相关

1. S3 LifeCycle Rule是否配置？
2. S3对象有多种存储类别，不同存储类别的定价不一样。为解决您的成本，您可以考虑将一些历史归档数据移动到STANDARD\_IA或Glacier中，以节约成本。
3. 您可以通过S3生命周期规则将数天前上传的文件归档到某一种类别。如对象创建30天自动归档到STANDARD\_IA，100天后自动归档到Glacier中。示例请参考: <https://docs.aws.amazon.com/zh_cn/AmazonS3/latest/user-guide/create-lifecycle.html>
4. S3 version control是否开启？
5. 为防止S3对象误删除，建议您开启版本控制功能。注意 ：版本控制一旦开启，则无法关闭(您可以暂停版本控制功能)。 <https://docs.aws.amazon.com/zh_cn/AmazonS3/latest/dev/Versioning.html>
6. S3 bucket policy 是否设置权限细分？
7. 在使用S3存储桶策略的时候，建议您通过一些限制条件来将访问限制在一个较小的范围内，以防止您的重要数据泄漏.
8. 例如: 使用S3托管静态网站不建议直接完全公开化存储桶，最好加上referral header限制，如下述策略，只允许来源为https://www.example.com网站访问BucketName下的文件:

{  
    "Version": "2012-10-17",  
    "Id": "http referer policy example",  
    "Statement": [  
        {  
            "Sid": "Allow get requests originating from www.example.com.",  
            "Effect": "Allow",  
            "Principal": "\*",  
            "Action": "s3:GetObject",  
            "Resource": "arn:aws-cn:s3:::BucketName/\*",  
            "Condition": {  
                "StringLike": {  
                    "aws:Referer": "https://www.example.com/\*"  
                }  
            }  
        }  
    ]  
}

## 数据库/Cache相关

1. RDS auto backup是否启用？备份保留期是否能满足业务需求？
2. 使用RDS API或AWS CLI创建的数据库实例，默认备份保留期为1天，通过控制台创建的数据库实例，默认备份保留期为7天。请不要禁用自动备份，因为该操作会禁用时间点恢复(PITR): <https://docs.aws.amazon.com/zh_cn/AmazonRDS/latest/UserGuide/USER_WorkingWithAutomatedBackups.html#USER_WorkingWithAutomatedBackups.BackupRetention>
3. 您也可以创建手动快照来永久保留某一个特定时间点的数据库备份。
4. RDS 数据库是否使用了自定义参数组？
5. 在创建RDS的时候需要选择一个参数组，默认参数组的参数里面的参数无法修改，可能会对后期维护产生不便。建议您创建一个新的参数组，将RDS应用到新的参数组上。
6. RDS 子网组是否都为private subnet？
7. 除非您希望将RDS数据库公开访问，否则为了您的数据安全，建议您将数据库实例启动到私有子网中，在创建RDS子网组时，请选择私有子网的subnet。
8. RDS 是否启用了Multi-AZ？
9. Amazon RDS 多可用区域部署为数据库 (DB) 实例提供了增强的可用性和持久性，当您配置多可用区域数据库实例时，Amazon RDS 会自动创建主数据库实例并将数据同步复制到其他可用区域 (AZ) 中的备用实例。每个可用区域在其独立的、物理上显著不同的基础设施中运行，并已设计为具备高可靠性。万一发生基础设施故障，Amazon RDS 可自动故障转移至备用实例中 (如果是 Amazon Aurora，则会故障转移至只读副本中)，以便您能够在故障转移结束后立即恢复数据库操作。由于故障转移后数据库实例的终端节点维持不变，因此应用程序可在无需手动管理干预的情况下恢复数据库操作
10. 在生产环境中，建议您将RDS开启多可用区部署，以提高其可用性: <https://docs.aws.amazon.com/zh_cn/AmazonRDS/latest/UserGuide/Concepts.MultiAZ.html>
11. ElastiCache for Redis 是否有充足的预留内存？
12. 预留内存是为非数据使用情况留出的内存。执行备份或故障转移时，Redis 使用可用的内存来记录将集群数据写入 .rdb 文件时对集群执行的写入操作。如果您没有足够的内存可供所有写入使用，则进程失败。
13. 建议：对于 2.8.22 版之前的 Redis，预留节点类型 maxmemory 值一半的内存，对于 Redis 2.8.22 版和更高版本则为四分之一。详情请参考: <https://docs.aws.amazon.com/zh_cn/AmazonElastiCache/latest/red-ug/redis-memory-management.html>

## ELB相关

1. AutoScaling 是否使用ELB健康检查，还是EC2健康检查？
2. 建议将AutoScaling的健康检查设置为ELB健康检查。使用EC2健康检查的情况下，只有当EC2实例不健康，AutoScaling Group才会将实例终止并启动新实例替换。
3. 如果使用ELB健康检查，那么当应用中断之后，ELB就会把实例标记为不健康，AutoScaling Group会将实例终止，然后从AutoScaling 启动配置/启动模板 中新起一个实例替换。
4. ELB 是否启用访问日志？
5. 建议开启ELB的访问日志，可以帮助您分析业务的流量，以及出问题时方便troubleshooting.
6. 应用程序负载均衡器开启日志: <https://docs.aws.amazon.com/zh_cn/elasticloadbalancing/latest/application/load-balancer-access-logs.html>
7. 传统负载均衡器日志开启:<https://docs.aws.amazon.com/zh_cn/elasticloadbalancing/latest/classic/access-log-collection.html>
8. ELB是否开启了跨可用区路由？
9. 默认情况下，ELB节点只会把流量发送给自己同一个可用区的后端实例。如果您的后端实例在可用区内分布不均匀，有可能会出现负载不均衡的情况，建议您将ELB开启跨区域负载均衡，从而减小不均衡对客户端的影响。
10. 参考<https://docs.aws.amazon.com/zh_cn/elasticloadbalancing/latest/classic/enable-disable-crosszone-lb.html#enable-cross-zone>
11. ELB是否需要预热？
12. ELB会随着您的流量的增加而弹性自动进行扩容，在一些特殊场景下，如电商秒杀等，会出现瞬间流量突增的情况，ELB扩容的速度可能来不及处理瞬间突增的流量。在这种场景下，您可以提交AWS Support技术支持案例，后台工程师会帮助您提前把ELB进行扩容，以使您能够平稳度过瞬间流量突增的活动。

## Service Limit相关

1. EC2 相关机型limit 是否满足预测峰值机器数量需求？
2. 默认情况下，EC2能起20个实例，但不同实例类型的限制也有所差异，您可以在EC2 Console界面，点击左侧的"限制"查看具体机型的限制额度。如果发现实例机型数量不能满足您的业务需求，请提前提交AWS Support案例来提升限制额度。
3. EBS 如果使用 io1类型 piops account 级别的上限是否足够，EBS 的容量上限是否足够？
4. 默认情况下，AWS账号级别的预配置IOPS(IO1类型卷)的限制为30万，预配置IOPS(IO1卷)类型的卷存储容量限制为300TB，如果您发现即将达到存储限制，请提交AWS Support案例来提升。
5. 安全组条目，路由规则条目等？
6. 默认情况下每个安全组能够添加60条规则，每个路由表能支持50条，以及每个VPC的子网数量，每个子网ACL规则数等都有相关限制，您可以在EC2 Console 界面，点击左侧的“限制”查看具体限制额度，如果需要提升额度，请联系AWS Support。
7. 每个region内第一次起EC2需要等2个小时，是否已经移除这个限制？
8. 每个区域内，首次启动EC2实例需要等待2个小时，请在大规模部署前，务必先启动一个实例来移除限制。
9. EIP默认只有5个，是否提高了limit？
10. 默认情况下，EIP限制为5个，如果需要申请更多EIP，请联系AWS Support。
11. SES sandbox 环境移出以及发送速率的提升
12. 新Global AWS账号的SES服务处于沙盒中，每秒发送邮件速率以及每天发送邮件数量都有限制。如果您会用到SES服务，请在支持中心开启案例来移除SES沙盒。

## 其他

1. 是否不同用户有不同的账号，特权账户启用了MFA？
2. 建议您对不同的业务部门同事创建不同的IAM用户，分配不同的策略来做权限细分管理。同时为了加强安全性，请开启MFA: <https://docs.aws.amazon.com/zh_cn/IAM/latest/UserGuide/id_credentials_mfa.html>
3. CloudTrail 是否开启？
4. CloudTrail 可帮助对您的 AWS 账户进行监管、合规性检查、操作审核和风险审核，您在AWS上的API操作都可以用CloudTrail进行追踪。建议您开启CloudTrail追踪，并将追踪日志保存在S3: <https://docs.aws.amazon.com/zh_cn/awscloudtrail/latest/userguide/creating-an-organizational-trail-in-the-console.html>
5. IAM Access Key/Secret Key是否定期更新？是否在应用中将key写死？
6. IAM用户的Access Key/Secret Key请务必保持定期更新，以免key的泄漏导致风险事件发生。在您开发的应用程序中，请务必不要写死IAM用户的key。您可以通过Role的方式或sts方式获取临时的credential，也可以通过Cognito服务将用户认证与第三方idp进行关联。
7. 国内AWS账号是否开通80/443/8080端口，是否进行ICP备案？
8. 国内AWS账号默认80/443/8080端口是关闭的，以及S3公开下载是被禁止的，您需要通过ICP备案的方式移除该限制。有关ICP备案的信息，请联系AWS Support。
9. 国内API Gateway是否申请白名单？
10. 国内API Gateway默认无法公开使用，需要提交白名单申请开通。具体情况请联系AWS Support。