

▼【本文档使用方式】：

- 当作【字典】使用。

即学习时遇到【不理解含义的概念 or 服务】时，使用文档中的【[查找 / 替换](#)】功能，搜索对应【**概念 or 服务 名称**】。

- 可结合【各服务简短总结(可搜索)】使用：

【[AWS 服务一览-CLF02 | 基于“試験にできる AWS サービスの範囲 \(02 版\)”](#)】

<https://sanuei.github.io/AWS-CLF02-menu/>

▼【AWS CLF-C02 学习路径】：

1. 首先，请完整学习并理解 AWS (CLF-C02) 的服务内容。

- 学习资源①——课程：

B 站搜索：[【AWS 云从业者认证速通，0 基础带你入门云计算! \(AWS CCP CLF-C02\)】](#)

- 学习资源②——官方培训：

【[AWS skillbuilder 官方培训](#)】

<https://skillbuilder.aws/learn/94T2BEN85A/aws-cloud-practitioner-essentials-/41SH4XWVQ8>

【[AWS Document 官方文档](#)】

<https://docs.aws.amazon.com/>

- * 【课程】有助于理解；【官方文档】最正式、完整，且其中配有【图表&每部分例题】。

2. 其次，请刷题练习。

- 题库①——[【AWS CLF-C02 中文题库】](#) 购买题库微信：est258258

- * 这是我刷题使用的题库。

优点：是中文题库，方便理解；

缺点：没英文原题对照——有些 AWS 服务的中文译名会变化（正式考试时也是），所以记得一定要【**对照英文原名-中文常用名**】（可以自己用 ChatGPT 查对应词）进行学习！

- 题库②——[【英文题库】AWS Certified Cloud Practitioner.zip](#)

网盘链接：<https://pan.baidu.com/s/1rFbV0Hd3-eSM-H-dotMNUw?pwd=yoyo>

- * 英文题库，从网页打开，并用插件【[沉浸式翻译](#)】可进行对照理解。

AWS CPP(CLF-C02)笔记 目录

AWS CPP(CLF-C02)笔记 目录	2
■ AWS CCP 考试 (CLF-C02) 精简总结	5
○ 【计算】 总结	5
○ 【存储】 总结	6
○ 【数据库】 总结	6
○ 【容器】 总结	7
○ 【迁移和传输】 总结	7
○ 【联网与内容分发】	8
○ 【安全性身份和合规性】	9
一、■ 云概念	10
▼ 【地域、可用区】：	10
▼ 【云种类】：	10
▼ 【云服务模式】：	10
▼ 【服务】：	10
▼ 【付费模式】:pay as you go (即用即付)	11
▼ 【责任共担模型】：(见下图)	11
▼ 【SLA 协议 (服务等级协议)】：	11
一句话概括：承诺 99.99%时间服务正常，否则按比例赔付。	11
用处：	11
▼ 【IT 财务模型】：	13
CAPEX：(一次性)	13
OPEX：(持续性)	13
●Cloud Adoption Framework (CAF)：	14
●Well-Architected Framework (WAF)：	14
二、■ 云产品资源	15
○ 【计算】	15
15▼ EC2	15
16▼ Auto Scaling	17
17▼ Lightsail (低配版 EC2)	18
19▼ Lambda (按次节省; 事件触发)	19
20▼ AWS Batch	20
21▼ Local Zones	21
22▼ AWS Outposts (混合云; 可本地构建→低延时)	22
○ 【存储】	23
24▼ EBS (= 一块 硬盘)	23
25▼ EFS (=多个实例 可共享访问 同数据)	24
26▼ FSx (高性能: 低延迟+高吞吐量)	25
27▼ S3 (关键词: 检索大规模数据、数据分析; 对象存储; 降低成本)	26
28▼ S3 Glacier (存储 长期但很少访问 的数据)	27

29▼ Storage Gateway (网关——【本地-云】互相传输【备份文件】 本地部署 SG + 云端与 S3 集成)	27
30▼ AWS Backup (【云上】存储)	28
■ Backup & Storage Gateway 区别:	28
●31【存储 总结】:	28
○【数据库】	29
32▼ Amazon RDS (关系型数据库)	29
■ MySQL & Aurora 区别:	30
33▼ Aurora (兼容 + 性能和可靠性 更高 + 价格更低 + 可自动扩缩容)	30
■ 有服务器 & 无服务器 概念:	30
34▼ DynamoDB (无服务器+NoSQL(键值对型)+完全托管+任何规模下均个位数毫秒级的性能)	30
■ 关系型 DB & 非关系型 DB 概念:	31
36▼ Neptune (高性能图形 NoSQL; 高关联度; 存储数十亿关系, 查询延迟降到毫秒级)	31
●37【数据库 总结】:	32
○【容器】	33
■ 概念: 什么是【容器】?	33
38▼ ECS (=云上托管的 Docker 容器——打包应用进容器)	34
39▼ EKS (=云上托管的 K8S——编排容器; 管理调度容器, 简化跨环境应用管理)	34
40▼ Fargate (无服务器计算引擎; 并列概念 是 EC2 弹性计算云)	34
■ 解题技巧: 微服务 or 持续集成/交互	34
■ 区分: EC2 & Elastic Beanstalk & Lambda	35
41▼ ECR (=仓库→注册表的托管)	35
●42【容器 总结】:	36
○【迁移和传输】	37
43▼ Application Discovery Service (Discovery Agent; 管理类软件)	37
● 运行 Discovery Agent 收集信息→生成信息清单→洞察并管理软件	37
44▼ AMS / MGN (【服务器】线下迁移上云: 不停机(不断同步、实时增量)迁移)	37
46▼ Migration Hub (集中化控制台, 不是产品)	38
47▼ SCT (数据库迁移工具: 换 数据库引擎)	39
48▼ Snow Family (方便部署 &高速 &量大的物理传输家族)	39
● 做题 关键词: 离线环境/网络连接不稳定[物理]、大量数据迁移[量大&高速]	39
49▼ Transfer Family (文件传输服务, 支持多种传输协议: FTP、FTPS、SFTP)	39
●50【迁移和传输 总结】:	40
○【联网与内容分发】	41
51▼ VPC (=云中 内网)	41
52▼ CloudFront (内容分发=CDN; 缓存至更近)	43
53▼ Direct Connect (混合云; 专线-传输快,搭建慢; 静态)	43
54▼ Global Accelerator (加速联网; 静+动)	44
55▼ Route 53 (DNS; 注册+管理+解析 域名; 全球负载均衡)	44
56▼ VPN	44
57▼ Transit Gateway (中央枢纽 网关)	45

58▼ API Gateway	45
○ 【安全性身份和合规性】	46
59▼ ACM (AWS Certificate Manger)	46
60▼ AWS CloudHSM (硬件安全模块)	46
61▼ Amazon Detective (安全分析工具)	46
62▼ Cognito (账号管理; 身份验证与授权: 注册管理 + 第三方接入)	47
63▼ GuardDuty (威胁检测: AWS 账户 + 工作负载)	47
64▼ IAM (权限管控: 分配账号+分配权限) (Identity and Access Management)	47
65▼ IAM Identity Center (SSO 单点登录 = AWS Single Sign-On) (用于: 内部不同应用+外部访问)	48
66▼ Amazon Inspector(安全漏洞扫描: 软件漏洞+网络暴露)	48
■ Inspector 和 GuardDuty 区别:	48
67▼ Amazon KMS(加密 服务)(Key Management Service)	48
68▼ Macie(发现和保护敏感数据——只发现, 不能进行加密, 另行用别的加密)	49
69▼ WAF(防火墙, 阻挡对应用的攻击)Web Application firewall	49
70▼ Firewall Manager(WAF 防火墙的管理工具: 跨账户集中配置+管理防火墙规则)	49
71▼ Shield(安全防护: 防 DDoS 攻击+应用保护+API 端点防护)	50
72▼ Secrets Manager(托管密钥: 存储、管理、轮转敏感信息)	50
73▼ Secrets Hub (托管的 集中管理和监控账户安全状态——即时洞察威胁和违规)	50
74▼ Artifact (提供安全合规性 文档、报告)	50
75▼ RAM(跨账户资源共享)、AM(审计管理)、DS (企业级目录服务 (用户、账号、权限))	51

作成者: 朱晓骏

作成日: 2026 年 1 月 7 日

■ AWS CCP 考试 (CLF-C02) 精简总结

○ 【计算】总结



EC2 ——弹性计算云。

4 种**付费模式**:

On-Demand Instances	按需实例(按量)	: 贵、时间短(灵活)
Spot Instances	竞价实例(按量; 会中断)	: 折扣+系统中断 (无工作负载) (可能被其它出价高的人 中断服务 , 因此很多人把这个实例【 只作计算节点、不存数据 】而【 在预留实例上存储数据 】, 组合使用实例→此称为【 无状态工作负载 】)
Reserved Instances	预留实例(长期 1年以上; 绑定)	: 便宜(72%折扣价)、时间长; 绑定 AZ、型号、地区
Savings Plans	节省计划(长期 1年以上; 不绑定)	: 同预留; 但 不绑定实例系列大小、操作系统、租赁或 AWS 区域

AS (Auto Scaling) ——

- Cloud Watch** **[实时监控]**达负载阈值
- ELB (负载均衡)** **[伸缩 资源容量]**扩展到【多个服务器】进行处理

Batch ——模型训练、排列组合分析 的 **批处理**

ElasticBeanstalk ——**自动化**帮我们**部署基础设施**, 但仍需**自己管理**。

Lightsail ——**低配版 EC2**, 但有预配置模板、上手更快更简单。

→**简易化配置服务器 (基本最高 16 核 32G)**

***Lightsail 和 EC2 都是 VPS (虚拟私有服务器)**, 即都是【**计算服务器**】。

***EC2 最高可几百 G, 而轻量化的 Lightsail 基本最高 16 核 32G。**

Lambda ——**关键词**是【**无服务化计算(更节省)**】和【**事件触发**】。

- ▼对比:
- **EC2** ——完全**自主可控**
 - **Elastic Beanstalk** ——**帮助选择好**对应的**基础设施**, 管理还是**自己管理**。
 - **Lambda** ——只负责**代码的简易部署** (适用于**事件触发**类型)

○ 【存储】总结



只有 **1 块硬盘**，只能在 1 个 EC2 上使用—— **EBS**

多个服务器 同时访问 1 块硬盘 —— **EFS** (共享文件)

高性能 同时访问 1 块硬盘 —— **FSx**

数据量大、价格低、经常使用 —— **S3**

不经常使用、深度归档 —— **S3 Glacier**

本地-云 备份 —— **Storage Gateway**

云 备份 —— **Backup**

○ 【数据库】总结



企业·用户管理 —— **RDS**

性能高、费用低 —— **Aurora (RDS)**

游戏进度、点赞评论 → **键值对 DB** —— **DynamoDB**

会话、状态信息 → **内存型 DB** —— **MemoryDB for Redis**

【**关联度**】企业关系网、个性化推荐、知识图谱 → **图 DB** —— **Neptune**

○ 【容器】 总结



- ECS ——云上 **容器** (微服务; CI/CD)
- EKS ——云上 **容器调度、编排**
- ECR ——**镜像仓库**
- Fargate ——**无服务器化容器**服务
 - ECS 的 **EC2** 托管 →云主机 [要自己管理 →有更大自主可控性]
 - ECS 的 **Fargate** 托管 →无服务化 [无需自己管理 →自动部署、扩展、负载均衡]
- EC2 ——完全**自主可控**
- Elastic Beanstalk ——帮助**选择好**对应的**基础设施**，管理还是**自己管理**。
- Lambda ——只负责**代码的简易部署** (适用于**事件触发**类型)

○ 【迁移和传输】 总结



- AMS(MGN) ——迁移【**服务器**】——**不停机(不断同步、实时增量)**
- DMS ——迁移【**数据库**】
- SCT ——换【**数据库引擎**】

Migration Hub ——迁移任务的【**集中管理和调度的控制台**】

ADS (Application Discovery Service) ——系统不熟或太庞大 → 用 ADS 来【**发现并统计**】各类应用

SnowFamily ——【**物理传输**】→**离线没网** or **数据量大**(PB、TB)

TransferFamily ——【**传输**】→协议: **FTP、SFTP、FTPS**

○ 【联网与内容分发】



VPC

(=云中 **内网**)

通过【**对等连接**】【**VPN**】进行 **VPC 之间的打通**。

通过 **Gateway (网关)** **访问互联网**。

私有子网→**NAT Gateway** →连接 互联网

公有子网→**Internet Gateway** →连接 互联网

VPC 内部 访问 S3 → **VPC endpoint**

- **安全组** : 附加在 **EC2 实例**上**控制流量进出**。
- **NACL** : 附加在**子网**中**控制流量进出**。

CloudFront (内容分发=CDN; 缓存至更近)

Direct Connect (混合云; 专线-传输快,搭建慢; 静态)

VPN (混合云-加密连接; 公网连接; 比专线便宜)

Global Accelerator (加速联网; 静+动)

Route 53 (DNS; 注册+管理+解析 域名; 全球负载均衡)

Transit Gateway (中央枢纽 网关)

API Gateway

○ 【安全性身份和合规性】



59▼ ACM (AWS Certificate Manger)

60▼ AWS CloudHSM (硬件安全模块)

61▼ Amazon Detective (安全分析工具)

62▼ Cognito (账号管理; 身份验证与授权: 注册管理 + 第三方接入)

63▼ GuardDuty (威胁检测: AWS 账户 + 工作负载)

64▼ IAM (权限管控: 分配账号+分配权限) (Identity and Access Management)

65▼ IAM Identity Center (SSO 单点登录 = AWS Single Sign-On) (用于: 内部不同应用+外部访问)

66▼ Amazon Inspector(安全漏洞扫描: 软件漏洞+网络暴露)

■ Inspector 和 GuardDuty 区别:

67▼ Amazon KMS(加密 服务)(Key Management Service)

68▼ Macie(发现和保护敏感数据——只发现, 不能进行加密, 另行用别的加密)

69▼ WAF(防火墙, 阻挡对应用的攻击)Web Application firewall

70▼ Firewall Manager(WAF 防火墙的管理工具: 跨账户集中配置+管理防火墙规则)

71▼ Shield(安全防护: 防 DDoS 攻击+应用保护+API 端点防护)

72▼ Secrets Manager(托管密钥: 存储、管理、轮转敏感信息)

73▼ Secrets Hub (托管的 集中管理和监控账户安全状态——即时洞察威胁和违规)

74▼ Artifact (提供安全合规性 文档、报告)

75▼ RAM(跨账户资源共享)、AM(审计管理)、DS (企业级目录服务 (用户、账号、权限))

一、■ 云概念

▼【地域、可用区】：

region (地域)

available zone (可用区)

datacenter

local zones (本地区域)

edge location (边缘站点)

▼【云种类】：

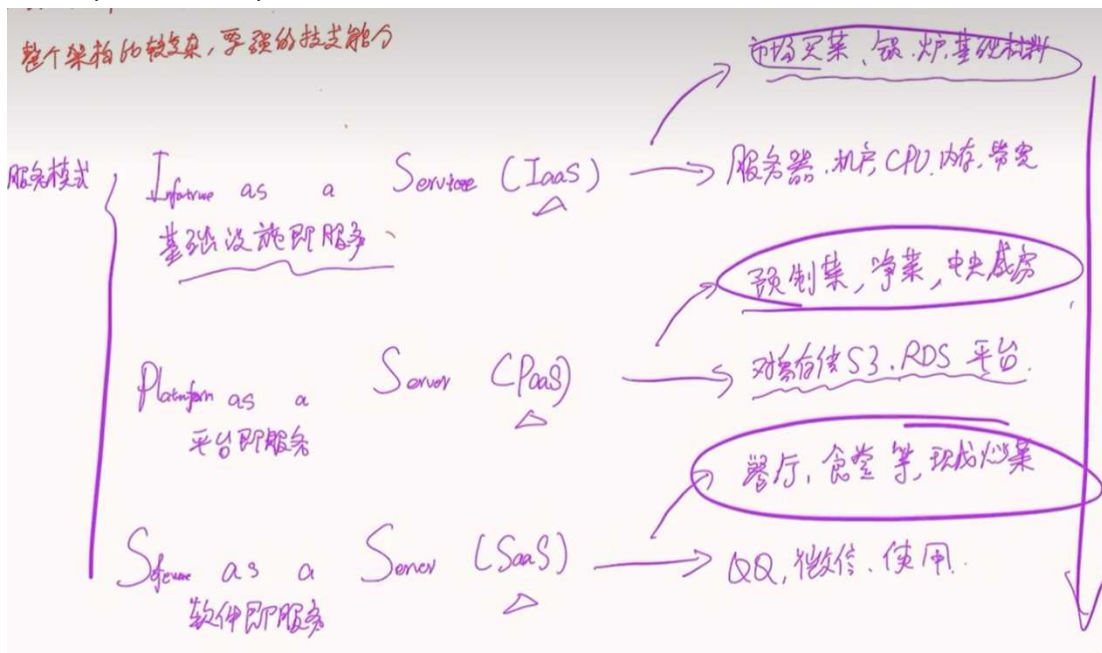
公有云、私有云、混合云

▼【云服务模式】：

IaaS (基础设施 即服务) ——给锅, 自己买菜、做菜 (infrastructure)

PaaS (平台 即服务) ——给预制菜, 自己做菜

SaaS (软件 即服务) ——直接上菜, 直接吃



▼【服务】：

(官网——产品与服务: <https://aws.amazon.com/cn/products/>)

●计算：

云服务器 EC2、Lambda (无服务器化的计算服务)、ECS、EKS

●存储：

对象存储 S3、NES (文件共享)、EBS(块存储)、EFS (完全托管型文件系统)

●网络：

VPC——route53、CloudFront 缓存、(GA)、DC、TransitGateway

●数据库:

RDS 关系型数据库——for mysql、Ms、Aurora、Neptune

非关系型数据库——redis、MongoDB、DynamoDB

●其他:

身份验证、AI 大模型、监控、审计、安全合规

▼【付费模式】:pay as you go (即用即付)

按量付费: 按小时 (1h0.5 刀、2.5h1.5 刀)、按分钟、按算力 (0.5cpu0.5memory)

包年包月 (预留) (1 年 8 折, 3 年 7 折)

▼【责任共担模型】: (见下图)

ECS的安全责任共担模型概览图如下所示。



▼【SLA 协议 (服务等级协议)】:

一句话概括: 承诺 99.99%时间服务正常, 否则按比例赔付。

用处:

如果服务器坏了, 客户向你索赔。

但按照【责任共担模型】——这是 AWS 服务商的责任, 所以你公司不该赔付、该 AWS 服务商赔付;

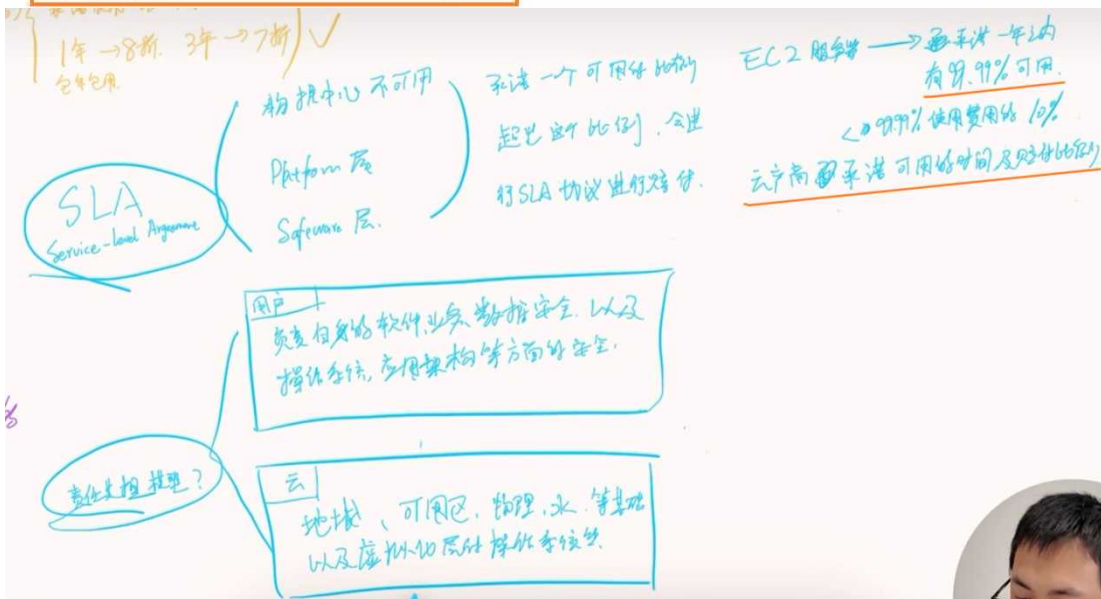
而赔付的多少, 按照【SLA 服务等级协议】的比例赔付, 如服务时间为 95~99.0%, 则赔付 30%的费用。

区域级 SLA

对于同时部署在同一区域（或者如果给定区域只有一个 AZ，则至少两个区域）的两个或多个 AZ 中的所有运行中实例的 Amazon EC2，AWS 将尽商业上的合理努力，使得在任何月度账单周期内，在每种情况下，Amazon EC2 在每个 AWS 区域的每月正常运行时间百分比至少达到 99.99%（“区域级 SLA”）。如果 Amazon EC2 不符合区域级 SLA，您将有资格获得如下所述的服务抵扣额度。

每月正常运行时间百分比 服务抵扣额度百分比

小于 99.99% 但等于或大于 99.0%	10%
小于 99.0% 但等于或大于 95.0%	30%
小于 95.0%	100%



▼ 【IT 财务模型】：



CAPEX：（一次性）

资本支出——固定财务资产（一次性租服务器）、无形资产（运维、技术人员）

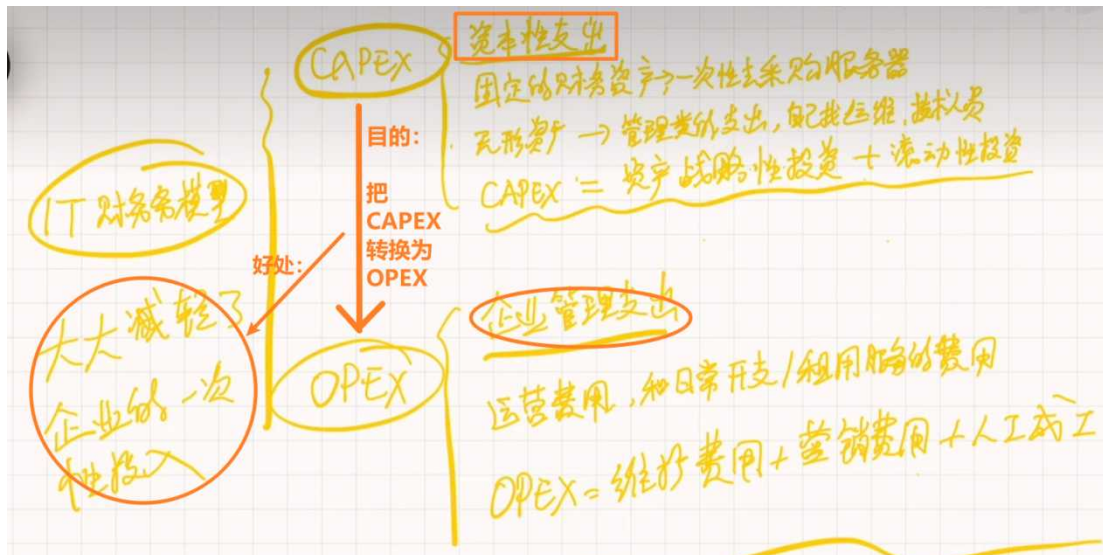
CAPEX=资产战略性投资+滚动性投资

OPEX：（持续性）

运营支出——运营费用、日常开支/租用服务的费用

OPEX=维护费用+营销费用+人工成本

目的：【把 CAPEX 转换为 OPEX】——好处：大大减轻了企业的一次性支出。



▼ 【两个框架】 CAF、WAF

● (CAF) Cloud Adoption Framework:

定义：利用最佳实践，让企业实现云转型 的实践指导（方法论）。

简述：【**理论框架**】；企业云转型(企业上云)的**方法论**。

■ 业务的【六大方面】：业务、人员、治理、平台、安全、运营

Business、People / Personnel、Governance、Platform、Security、Operations (常简称为 Ops)

平台角度：

- **数据 架构/工程**：在云中构建、管理和优化数据。
- **云基础 架构设计**
- **应用程序 架构**
- **集成和持续交付 (CI/CD)**
- **产品管理 (变更、发布)**

运营角度——**配置管理 + 补丁管理 + 事件管理**——**可观察性**

治理角度——**云采用业务成果和收益管理 + 程序和项目管理 + 风险管理** (管理风险、确保合规及优化资源使用)

业务视角——**组合管理 + 数据科学 (通过数据分析和洞察，驱动商业决策)**

■ CAF 【云转型阶段】

阶段	英文	核心含义 (一句话)	考试关键词
构想	Envision	明确为什么上云、上云要达成什么目标	展示云如何 加速业务成果 、价值、KPI
准备	Align / Prepare	建立组织、治理、安全、能力基础	组织 准备 、 治理 、技能
规划	Plan	制定迁移路线图与实施计划	迁移策略、优先级
发布	Launch	执行迁移并上线工作负载	迁移、部署、上线
扩展	Scale	大规模推广云使用，优化架构	自动化、扩展性
改进	Optimize	持续优化成本、性能、安全	成本优化、持续改进

● (WAF) Well-Architected Framework:

简述：【**技术框架**】。

■ 良好架构六大方面 (WAF 的【支柱】-最佳实践)：

可靠性、安全性、性能效率、成本优化、卓越运营、可持续性

Reliability、Security、Performance efficiency、Cost optimization、Operational excellence、Sustainability

共同点：都是指导企业上好云、管好云的白皮书、理论。

不同点：CAF 大方向理论指导；WAF 具体实践技术。

Table 1. The pillars of the AWS Well-Architected Framework

Name	Description
Operational excellence	The ability to support development and run workloads effectively, gain insight into their operations, and to continuously improve supporting processes and procedures to deliver business value.
Security	The security pillar describes how to take advantage of cloud technologies to protect data, systems, and assets in a way that can improve your security posture.
Reliability	The reliability pillar encompasses the ability of a workload to perform its intended function correctly and consistently when it's expected to. This includes the ability to operate and test the workload through its total lifecycle. This paper provides in-depth, best practice guidance for implementing reliable workloads on AWS.
Performance efficiency	The ability to use computing resources efficiently to meet system requirements, and to maintain efficiency as demand changes and technologies evolve.
Cost optimization	The ability to run systems to deliver business value at the lowest price point.
Sustainability	The ability to continually improve sustainability impacts by reducing energy consumption, increasing efficiency across all components of a workload by maximizing the benefits from provisioned resources and minimizing the total resources required.

二、■ 云产品资源

○ 【计算】

EC2、AS、Batch、ElasticBeanstalk、Lightsail、Lambda



15▼ EC2

●EC2 精简化定义:

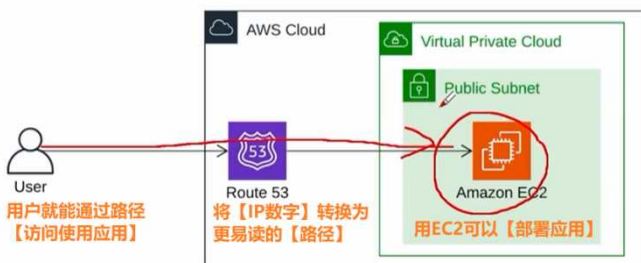
infrastructure 层服务; **弹性计算**(收缩、扩展)服务→实现**优化**; **降本**、**增加灵活性**。

我们: **租户**

产品: **实例**

●EC2 的运用 (案例) :

(结合下图案例理解) ①托管网站、②弹性计算加快处理数据、③数据备份



●付费模式:

On-Demand Instances **按需实例(按量)** : 贵、时间短(灵活)

Spot Instances **竞价实例(按量; 会中断)** : 折扣+系统中断 (**无工作负载**)

(可能被其它出价高的人**中断服务**, 因此很多人把这个实例【**只作计算节点、不存数据**】而【**在预留实例上存储数据**】, 组合使用实例→此称为【**无状态工作负载**】)

Reserved Instances **预留实例(长期 1年以上; 绑定)** : 便宜(72%折扣价)、时间长; **绑定 AZ、型号、地区**

Savings Plans **节省计划(长期 1年以上; 不绑定)** : 同预留; 但**不绑定实例系列大小、操作系统、租赁或 AWS 区域**
↑适用于**可变负载 (即业务量忽高忽低)**、**灵活地域和机器都能打折!**

Dedicated Host **专用主机** : 当用户想要利用他们现有的**每个套接字、每个核心或每个虚拟机的软件许可证**, 在 AWS 上运行 Microsoft Windows 服务器时需要。这允许**全面的自带许可证 (BYOL)** 支持。

付费模式：

On-Demand Instances 按需实例，使用按需实例，您只需要按小时或秒数支付计算容量，无需长期购买。

按需实例推荐用途：

- 希望拥有低成本和 EC2 提供的灵活性，且不想支付预付款或签订长期合同的用户
- 具有短期、难应付或无法预测且不能中断的工作负载的应用程序
- 首次在 EC2 上开发或测试的应用程序

<https://aws.amazon.com/cn/ec2/pricing/on-demand/>

Spot Instances 竞价实例，这种模式的核心特点包括折扣售卖和系统中断机制。用户可以以低于按量计费实例的价格购买这些资源，但存在一定风险，即系统可能会在后续自动回收这些折扣售卖的实例。与按需价格相比，可享受高达 90% 的折扣。

竞价实例建议用于：

- 容错或无状态工作负载
- 可以在异构硬件上运行的应用程序
- 开始时间和结束时间灵活的应用程序

<https://aws.amazon.com/cn/ec2/spot/pricing/>

Reserved Instances 预留实例，一次性签署一年或三年期合同，并获得高达 72% 的账单折扣作为回报。预留实例与指定 AZ 和实例型号绑定。

按需容量预留建议用于：

- 需要容量保证的业务关键型事件或工作负载
- 需要满足高可用性监管要求的工作负载
- 灾难恢复

https://docs.aws.amazon.com/zh_cn/AWSEC2/latest/UserGuide/ec2-capacity-reservations.html

Savings Plans 节省计划，是一种灵活的定价模式。这种定价模式为 EC2 实例的使用提供了更低的价格，而不考虑实例系列、大小、操作系统、租赁或 AWS 区域。与 EC2 预留实例一样，在一年或三年内使用特定数量计算能力的承诺，相比于按需价格，该模式可协助将费用减少多达 72%。

建议将节省计划用于：

- 承诺使用情况和稳定使用情况
- 希望在继续省钱的同时利用最新计算产品的用户

<https://aws.amazon.com/cn/blogs/china/new-saving-services/>



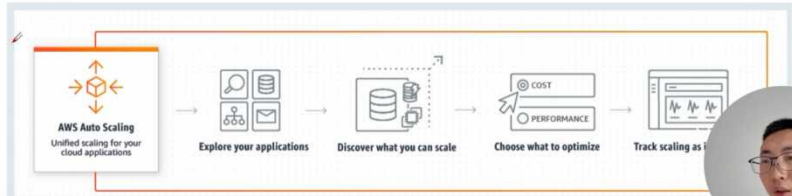
16▼ Auto Scaling

● Auto Scaling 精简定义:

把用户多时达到**特定负载阈值**，**扩展到【多个服务器】**进行处理→**按设定好的条件，实时监控、并伸缩 资源容量。**

AWS Auto Scaling 是亚马逊提供的一项自动化服务，旨在根据应用程序的需求**自动调整 AWS 资源的容量**，以应对流量的变化，并确保应用程序的性能和可用性，同时最大程度地降低成本。

AWS Auto Scaling 可以监控您的应用程序并自动调整容量，从而以尽可能低的成本来保持稳定、可预测的性能。该服务可以提供一个简单而功能强大的用户界面，让您可以为 Amazon EC2 实例和 Spot 队列、Amazon ECS 任务、Amazon DynamoDB 表和索引以及 Amazon Aurora 副本等资源制定扩展计划。如果您已经在使用 Amazon EC2 Auto Scaling 来动态扩展 Amazon EC2 实例，那么现在可以将其与 AWS Auto Scaling 结合使用，为其他 AWS 服务扩展其他资源。有了 AWS Auto Scaling，您的应用程序就始终能在合适的时间获得合适的资源。



●AS 补充:

ELB (负载均衡)

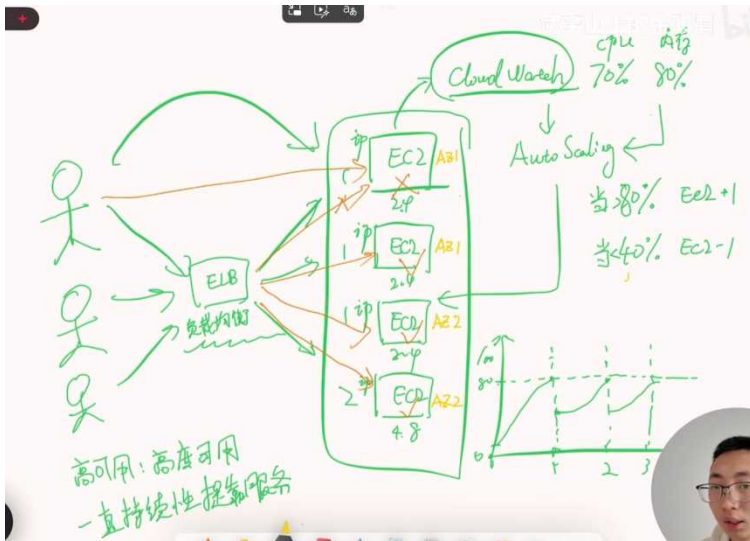
CloudWatch:

监控 CPU 和内存的负载。

例如，一台 CPU 负载到 80%时，自动加到 2 条服务器，每台 40%起的负载，三台 30%多起的负载.....

高可用:

一直持续性提供服务。——业务**至少在【2 个可用区】(AZ1、AZ2)上。**

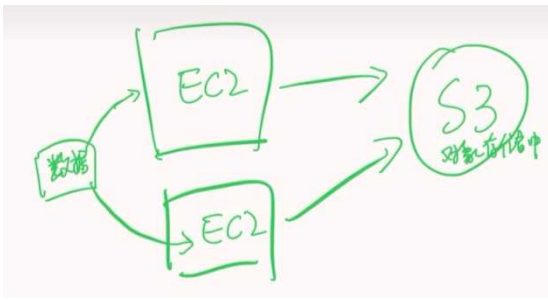


●【请求→...→存储】的流程:

用户请求 → 数据库 → ELB → EC2 → S3 (对象存储)

*注意①: 也就是说，EC2 不接收请求，而是开机后从数据库中拉取已有请求。因此可以经过 ELB 均衡处理分配负载。

*注意②: 有【用户交互】才需要设置【ELB(负载均衡)】。



17▼ Lightsail (低配版 EC2) = AWS 的“新手/小项目一站式服务器”——快速搭建

关键词:

固定价格——月费清晰、可预测

一体化——计算 + 存储 + 网络

预配置——提供 WordPress / LAMP 等模板

可升级——可迁移到 EC2

Amazon Lightsail 是亚马逊提供的一种简化的 虚拟私有服务器 (VPS) 服务, 旨在帮助开发人员快速、轻松地搭建和管理虚拟服务器。Lightsail 提供了预配置的计算资源、网络、存储和数据传输选项, 用户可以通过简单的界面选择所需的配置, 轻松部署应用程序和网站, 而无需担心复杂的基础架构管理细节。

精简定义:

低配版 EC2, 但有预配置模板、上手更快更简单。→ **简易化配置服务器 (基本最高 16 核 32G)**

*Lightsail 和 EC2 都是 **VPS (虚拟私有服务器)**, 即都是【服务器】。

*EC2 最高可几百 G, 而轻量化的 Lightsail 基本最高 16 核 32G。

18▼ Elastic Beanstalk (自动部署基础设施)

精简定义:

【**自动化帮我们部署基础设施**】, 我们只需【**上传代码**】即可。

只需:
upload 代码
安全组代码
里面的程序为水
们都部署好基础
设施。

★★★AWS Elastic Beanstalk

AWS Elastic Beanstalk 是一项用于简化应用程序部署和管理的托管服务。它允许开发人员上传他们的应用程序代码, 并自动处理底层的部署、扩展、负载均衡和监控任务, 从而使开发人员能够专注于应用程序的开发而不必担心基础设施的管理细节。

借助 Elastic Beanstalk, 您可以在 AWS 云中快速部署和管理应用程序, 而不必了解运行这些应用程序的基础设施。Elastic Beanstalk 可降低管理的复杂性, 但不会影响选择或控制。您只需上传应用程序, Elastic Beanstalk 将自动处理有关容量预配置、负载均衡、扩展和应用程序运行状况监控的部署细节。

Elastic Beanstalk 支持在 Go、Java、.NET、Node.js、PHP、Python 和 Ruby 中开发的应用程序。在部署应用程序时, Elastic Beanstalk 会构建选定的受支持的平台版本, 并预配置一个或多个 AWS 资源 (如 Amazon EC2 实例) 来运行应用程序。

案例展示

Case1: Web 应用部署
一家初创公司开发了一个新的 Web 应用, 并希望将其部署到云端。他们选择使用 AWS Elastic Beanstalk 来托管他们的应用程序。通过 Elastic Beanstalk, 他们可以轻松地上传应用程序代码, 并利用 Elastic Beanstalk 的自动扩展和负载均衡功能来管理应用程序的部署和运行。这样, 他们可以快速地应用程序部署到云端, 并在应用程序的需求增加时自动扩展计算资源, 以确保应用程序的稳定性和性能。

Case2: API 服务部署
一家软件公司开发了一组 API 服务, 用于支持其移动应用程序和网站。他们选择使用 AWS Elastic Beanstalk 来部署和管理他们的 API 服务。通过 Elastic Beanstalk, 他们可以轻松地将 API 服务部署到云端, 并利用 Elastic Beanstalk 的自动负载均衡和监控功能来确保 API 服务的高可用性和可靠性。这样, 他们可以专注于开发和维护 API 服务的功能, 而无需担心基础设施的管理问题。

Case3: 后端应用部署
一家电子商务公司需要将其后端应用程序部署到云端, 并确保应用程序的高可用性和可扩展性。他们选择使用 AWS Elastic Beanstalk 来管理他们的后端应用程序。通过 Elastic Beanstalk, 他们可以轻松地部署后端应用程序, 并利用 Elastic Beanstalk 的自动扩展功能来根据应用程序的需求动态分配计算资源。这样, 他们可以确保后端应用程序始终能够满足用户的需求, 并在流量增加时保持性能。

19▼ Lambda (无服务器化; Lambda 函数的最大执行时间是 15 分钟; 按次节省; 事件触发)

精简化定义:

按次付费, 比 EC2 (按时长付费) **更节省的服务器**; **关键词**是【**无服务器化计算**】和【**事件触发**】。

*** AWS Lambda

AWS Lambda 是亚马逊提供的一项无服务器计算服务, 它允许开发人员在无需管理服务器的情况下运行代码。Lambda 可以自动扩展以处理任何规模的请求, 并且只会收取实际执行代码的费用, 而不会收取任何预订或固定费用。



代码

购买 按使用时长付费 租房

不使用时也要付费

一整月钱

EC2

更昂贵的按量计费 宾馆

按次付费

随代码付一次钱

为每行代码付的钱

次 10s

Serverless 无服务器化计算

案例展示

Case1: 图像处理服务

一家社交媒体平台需要对用户上传的图像进行处理, 例如缩放、裁剪或添加水印。他们使用 AWS Lambda 创建了一个图像处理服务。当用户上传图像时, Lambda 函数会自动触发并对图像进行处理, 然后将处理后的图像存储回亚马逊 S3 存储桶。这样, 他们可以根据需要处理任意数量的图像, 并且只需支付实际执行代码的费用。

Case2: 日志分析

一家网络安全公司需要对大量的网络日志进行实时分析, 以检测潜在的安全威胁。他们使用 AWS Lambda 创建了一个日志分析服务。当新的日志数据到达亚马逊 CloudWatch 日志组时, Lambda 函数会自动触发并对日志数据进行分析, 识别可能的安全事件, 并发送警报通知给安全团队。通过 Lambda, 他们可以实现高效的实时日志分析, 而无需担心基础设施的管理。

Case3: 后端 API

一家在线零售商需要构建后端 API 来支持其移动应用和网站。他们使用 AWS Lambda 创建了一组后端 API。每个 API 端点都与一个 Lambda 函数关联, 负责处理特定的请求。例如, 一个 Lambda 函数负责处理用户注册请求, 另一个负责处理订单查询请求。这样, 他们可以快速搭建可扩展的后端 API, 而不必担心服务器的管理和扩展。

20 ▼ AWS Batch

关键词:

机器学习模型训练、模拟、规模分析的 **批处理**。

应用领域: **生命科学** (大量排列组合研究靶向药)、**自动驾驶系统** (多容器组合作业, 减少准备、加快开发)

☆☆☆ AWS Batch

AWS Batch 是亚马逊提供的一项批量计算服务, 旨在帮助用户高效地处理大规模的计算工作负载。AWS Batch 可以自动调度、运行和监控批处理作业, 用户无需管理底层的计算资源, 可以专注于编写和提交作业。AWS Batch 提供了灵活的配置选项, 包括不同类型的计算环境和作业队列, 以满足不同应用场景的需求。

AWS Batch

适用于机器学习模型训练、模拟和任何规模分析的批处理

AWS Batch 入门

创建 AWS 账户



案例展示

Case1: 科学计算

一个科研团队需要对大量的科学数据进行分析 and 计算。他们利用 AWS Batch 创建了一个计算环境, 并将计算任务提交到 AWS Batch 的作业队列中。AWS Batch 自动调度并运行这些任务, 并根据需求动态分配计算资源。通过 AWS Batch, 他们可以高效地完成科学计算任务, 加速研究进程。

Case2: 生命科学

生物制药和基因组学公司依赖高性能计算来推动产品上市。AWS Batch 简化了不同应用领域的操作, 例如计算化学、临床建模、分子动力学, 以及基因组测序测试和分析等。在药物筛选期间, AWS Batch 使研究科学家能够高效地搜索小分子库, 从而确定最有可能绑定到药物靶标 (通常是蛋白受体或酶) 的结构。这一过程有助于药物设计, 有潜力促进更有效药物和疗法的开发。对于 DNA 测序, 在生物信息学家完成对基因组序列的初步分析并生成原始文件后, 他们可以利用 AWS Batch 自动执行二次分析, 包括将原始 DNA 读取内容汇编成完整的基因组序列, 从而减少错误。

Case3: 自动驾驶数据

汽车公司在开发和测试自动驾驶汽车 (AV) 以及高级驾驶员辅助系统 (ADAS) 时依赖模拟。工程师使用容器, 将模拟中的每项要素 (车辆传感器、交通和 3D 环境) 建模成更小的模块化组件。由于能够使用 AWS Batch 运行多容器作业, 您可以享受到 AWS Batch 的高级扩展、调度和成本优化功能, 而无需将您的系统重构为一个复杂的整体式容器。相反, 您可以使用多个更小的模块化容器代表不同的系统组件。此功能通过减少作业准备步骤来加快开发速度, 消除了构建额外内部工具的需求, 并简化了开发 (Dev)、IT 运营 (Ops) 和调试。



21 ▼ Local Zones

关键词:

AWS Local Zones 是亚马逊提供的一种区域性基础设施扩展服务，距离主要 AWS 区域更近的地理位置，以降低延迟并提高性能。Loc

☆☆☆ AWS Local Zones

AWS Local Zones 是亚马逊提供的一种区域性基础设施扩展服务，它提供了与 AWS 区域相似的功能和服务，但位于距离主要 AWS 区域更近的地理位置，以降低延迟并提高性能。Local Zones 允许客户在更接近其用户和数据的地方部署应用程序，并提供与主要 AWS 区域相同的可靠性和安全性。



22 ▼ AWS Outposts (混合云【云→本地】；低延时)

关键词:

混合云; 【云服务→部署到→本地数据中心】(将 AWS 基础设施和服务部署到本地数据中心)

将 AWS 计算和存储 扩展到 本地数据中心→低延迟;

使能在 本地环境中构建和运行 AWS 云端一致的应用。

应用场景:

本地处理敏感数据, AWS 云服务器交换和集成数据。

案例展示

Case1: 敏感数据处理

一家银行机构拥有大量的敏感数据, 需要将部分数据在本地数据中心进行处理和存储, 同时又需要与 AWS 云端的服务进行集成。他们选择使用 AWS Outposts, 在本地数据中心部署 Outposts, 以便在本地处理敏感数据, 并利用 Outposts 与 AWS 云端服务进行数据交换和集成。这样, 他们可以在满足合规性要求的前提下, 利用 AWS 的灵活性和弹性来构建和运行业务应用。

Case2: 边缘计算应用

一家物联网公司需要在边缘设备附近处理和分析大量的传感器数据, 并实时做出决策。他们选择使用 AWS Outposts, 在边缘设备所在地部署 Outposts, 并在本地进行数据处理和分析。通过 Outposts, 他们可以利用与 AWS 云端相同的服务和工具来构建和管理边缘计算应用, 同时又能够在本地环境中实现低延迟的数据处理和响应。

Case3: 应用程序现场部署

一家制造公司需要在现场部署应用程序, 以支持其生产线和工厂操作。由于生产环境对延迟和可用性要求较高, 他们选择使用 AWS Outposts 在现场部署应用程序。通过 Outposts, 他们可以在生产现场部署与 AWS 云端相同的应用程序和服务, 确保应用程序能够与云端服务实时交互, 并提供高可靠性和可用性。

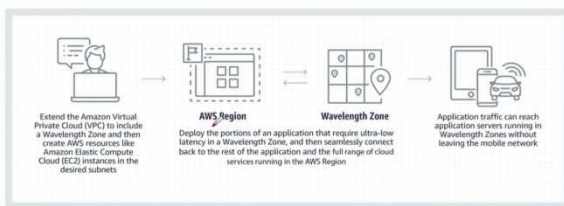
23 ▼ AWS Wavelength (5G 边缘计算→低延迟(数据传输))

关键词: 5G 边缘计算基础设施

☆☆☆ AWS Wavelength

5G 边缘计算基础设施

AWS Wavelength 是亚马逊提供的一种边缘计算服务, 旨在将 AWS 的计算和存储服务扩展到运营商的移动网络边缘, 以实现低延迟的应用程序服务。AWS Wavelength 将 AWS 计算和存储服务嵌入到 5G 网络中, 为开发、部署和扩展超低延迟应用程序提供移动边缘计算基础设施。Wavelength 允许开发人员在运营商的移动网络基础设施中部署应用程序, 并将应用程序与 AWS 云端服务进行集成, 从而在移动网络边缘进行低延迟的数据处理和分析。



案例展示

Case1: 在线游戏服务

一家在线游戏公司需要在移动网络边缘部署游戏服务器, 以提供低延迟的游戏体验。他们选择使用 AWS Wavelength, 在运营商的移动网络基础设施中部署游戏服务器, 并与 AWS 云端服务进行集成。通过 Wavelength, 他们可以实现游戏服务器与游戏客户端之间的低延迟通信, 从而提高游戏玩家的体验和满意度。

Case2: 智能城市监控

一座智能城市项目需要在移动网络边缘部署监控摄像头和传感器, 以实时监测城市的交通、环境和安全情况。他们选择使用 AWS Wavelength, 在运营商的移动网络基础设施中部署监控设备, 并与 AWS 云端服务进行集成。通过 Wavelength, 他们可以实现监控设备与云端服务之间的低延迟数据传输, 以及实时响应城市各种事件和情况。

Case3: 5G 边缘应用

一家物联网公司需要在移动网络边缘部署 5G 边缘应用, 以支持其物联网设备和传感器的数据处理和分析。他们选择使用 AWS Wavelength, 在运营商的移动网络基础设施中部署 5G 边缘应用, 并与 AWS 云端服务进行集成。通过 Wavelength, 他们可以实现 5G 边缘应用与云端服务之间的低延迟通信, 从而实现物联网设备和传感器的实时数据处理和分析。

○ 【存储】

PART



- ★★★★ Amazon Elastic Block Store (Amazon EBS)
- ★★★★ Amazon Elastic File System (Amazon EFS)
- ★★☆ Amazon FSx
- ★★★★ Amazon S3
- ★★★★ Amazon S3 Glacier
- ★★☆ AWS Storage Gateway
- ★★☆ AWS Backup

24 ▼ EBS (= 一块 硬盘) → 文件是操作系统自己在 EBS 上建的

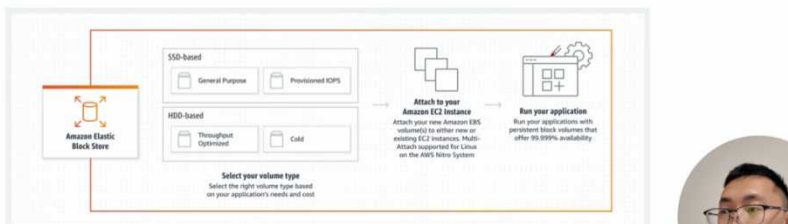
持久性; 高可靠性;

可根据需求 弹性扩容;

EBS 只可挂载到 1 个服务器上。

★★★★ Amazon Elastic Block Store (Amazon EBS)

Amazon Elastic Block Store (Amazon EBS) 是亚马逊提供的一种持久性块存储服务, 可供 EC2 实例使用。它提供了可通过网络连接到的 EC2 实例的持久性块存储, 可以在实例之间移动, 从而为用户提供了在云端运行的持久性块存储解决方案。



案例展示

Case1: 数据库存储

一家电子商务公司正在部署其在线商店的数据库。他们选择使用 Amazon EBS 来存储数据库文件, 以确保数据持久性和高可靠性。通过将数据库文件存储在 Amazon EBS 卷上, 他们可以保证数据在 EC2 实例重启或故障时不会丢失, 并且可以根据需要扩展存储容量。

Case2: 文件存储

一家媒体公司需要在多个 EC2 实例之间共享大量的文件数据。他们使用 Amazon EBS 创建了一个共享文件系统, 并将其挂载到每个 EC2 实例上。这样, 他们的团队可以在不同的 EC2 实例之间共享和访问相同的文件数据, 从而实现团队协作和文件共享。

选EFS

Case3: 应用程序日志存储

一家软件开发公司正在部署其应用程序到 AWS 上。他们使用 Amazon EBS 来存储应用程序生成的日志文件, 并将其挂载到 EC2 实例上。通过将日志文件存储在 Amazon EBS 卷上, 他们可以方便地对日志进行存储和检索, 并且可以根据需要扩展存储容量以应对日志数据的增长。

25 ▼ EFS (=多个实例 可共享访问 同数据)

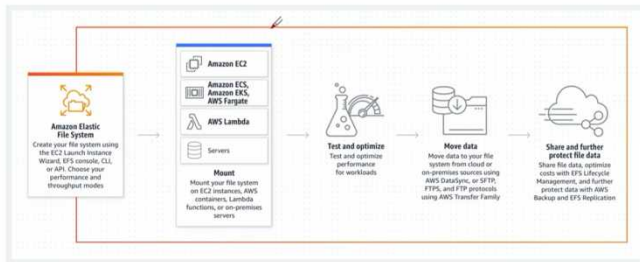
精简: **多实例访问同数据。**

应用: **文件共享** (方便多实例工作协作)、**数据文件备份** (云端安全可靠、且可按需扩容)

★★★ Amazon Elastic File System (Amazon EFS)

Amazon Elastic File System (Amazon EFS) 是亚马逊提供的一种托管文件存储服务, 可在多个 EC2 实例之间提供可扩展的、高性能的共享文件存储。

Amazon EFS 是一种通用的、高可用性的文件存储服务, 适用于各种不同类型的工作负载, 如应用程序部署、文件共享、数据备份等。



案例展示

Case1: 应用程序部署

一家软件开发公司正在部署其新的微服务应用程序。他们选择使用 Amazon EFS 来存储应用程序的共享配置文件和静态资源, 以便多个 EC2 实例可以共享相同的文件系统。这样, 他们的开发团队可以方便地访问和更新共享的配置文件, 同时确保所有实例都使用相同的配置。

Case2: 媒体文件存储

一家媒体公司需要存储大量的媒体文件, 如音乐、视频和图像。他们使用 Amazon EFS 创建了一个共享文件系统, 将其挂载到多个 EC2 实例上。这样, 他们的团队可以方便地共享和访问媒体文件, 并且可以在多个实例之间实现协作和文件共享, 从而提高工作效率。

Case3: 数据备份存储

一家企业需要一个可靠的文件备份解决方案, 以备份其重要数据。他们使用 Amazon EFS 创建了一个备份文件系统, 并将其挂载到他们的备份服务器上。这样, 他们可以将备份数据存储在 Amazon EFS 上, 以确保数据的安全性和可靠性, 并且可以根据需要扩展存储容量以应对备份数据的增长。



26▼ FSx (高性能: 低延迟+高吞吐量; 文件储存、共享系统)

关键词: 低延迟; 高吞吐量 (大规模、高性能计算); 可扩展

本质: 高性能文件共享系统。

★★☆ Amazon FSx

Amazon FSx 是亚马逊提供的一种托管文件存储服务, 它为用户提供了可扩展的、高性能的文件存储解决方案, 可用于在云中运行各种工作负载。

Amazon FSx 通常提供更高性能的文件存储, 适用于需要低延迟和高吞吐量的工作负载, 如大规模数据分析、高性能计算 (HPC) 等。



案例展示

Case1: Windows 文件共享

一家企业需要在 AWS 上部署 Windows 文件共享服务, 以便员工可以方便地共享和访问文件。他们选择使用 Amazon FSx for Windows File Server 来部署文件共享服务。Amazon FSx 提供了一个托管的 Windows 文件系统, 可以轻松地与现有的 Windows 环境集成, 并为用户提供了与本地文件共享服务类似的体验。

Case2: HPC 文件存储

一家科学研究机构需要一个高性能的文件系统来支持其高性能计算 (HPC) 工作负载。他们选择使用 Amazon FSx for Lustre 来部署 Lustre 文件系统。Amazon FSx for Lustre 提供了一个可扩展的、高性能的文件系统, 可用于存储和处理大规模的科学数据, 以支持复杂的 HPC 应用程序和工作流程。

Case3: SAP 文件共享

一家制造公司正在迁移其 SAP 系统到 AWS 上, 并需要一个可靠的文件共享解决方案来存储 SAP 数据和文档。他们选择使用 Amazon FSx for Windows File Server 来部署 SAP 文件共享服务。Amazon FSx for Windows File Server 提供了与 SAP 系统兼容的高可靠性、可扩展的 Windows 文件系统, 可以满足其对文件共享的需求, 并确保数据的安全性和可靠性。

27 ▼ S3 (关键词: 检索大规模数据、数据分析; 对象存储; 降低成本; 无服务器) = 云端网盘 / 文件仓库

* **持久性、高可用性(高度耐用)、可扩展性** 的 **对象存储** 服务。

* **检索大规模** 的数据对象 (如文件、图像、视频等) 。

* S3 (简单存储服务) 是**无服务器**的, 用户**无需管理**存储服务器

应用场景: 静态网页托管; 数据存储; 大规模(PB)数据分析

★★★ Amazon S3

Amazon S3 (Simple Storage Service) 是亚马逊提供的一种持久性、高可用性、可扩展的对象存储服务, 用于存储和检索大规模的数据对象, 如文件、图像、视频等。

- 扩展存储资源, 通过 99.999999999% (11 个 9) 的数据持久性满足不断变化的需求。
- 以 Amazon S3 存储类存储数据, **降低成本**, 无需前期投资或硬件更新周期。
- 通过无与伦比的安全性、合规性和审核功能保护您的数据。
- 通过强大的访问控制、灵活的复制工具和组织范围的可见性轻松管理任何规模的数据。



案例展示

Case 1: 静态网站托管

一家小型公司需要托管其静态网站, 并确保网站内容的高可用性和可靠性。他们选择使用 Amazon S3 来存储网站文件, 并将 S3 存储桶配置为静态网站托管。这样, 他们的网站文件将被存储在高度可靠的亚马逊基础设施上, 并且可以通过 S3 提供的 CDN (内容分发网络) 功能在全球范围内快速分发网站内容。

Case 2: 数据备份和存档

一家医疗保健机构需要一个可靠的数据备份和存档解决方案, 以确保其关键数据的安全性和可恢复性。他们选择使用 Amazon S3 来存储备份和存档数据。通过使用 Amazon S3 的高度持久性和可靠性, 他们可以确保备份和存档数据不会丢失, 并且可以根据需要轻松地检索和恢复数据。

Case 3: 大规模数据分析

一家电商公司需要对其大规模的数据进行分析, 以了解其客户行为和购买模式。他们选择将所有交易和用户活动数据存储在 Amazon S3 中, 并使用 AWS 的数据分析服务 (如 Amazon Athena 和 Amazon Redshift) 来分析这些数据。通过将数据存储在 Amazon S3 中, 他们可以轻松地扩展存储容量, 并利用 S3 提供的高可用性和低延迟来快速访问和处理数据。

存储类

访问频率高

S3 存储类包括:

S3 Standard, 适用于频繁访问的数据;

S3 Intelligent-Tiering, 可自动为具有未知或不断变化的访问模式的数据节省成本;

S3 Express One Zone, 适用于访问频率较高的数据;

S3 Standard-Infrequent Access, S3 Standard-IA 和 S3 One Zone-Infrequent Access (S3 One Zone-IA), 适用于访问频率较低的数据;

S3 Glacier Instant Retrieval, 适用于需要即时访问的归档数据;

S3 Glacier Flexible Retrieval (前称为 S3 Glacier), 适用于很少访问且不需要即时访问的长期数据;

Amazon S3 Glacier Deep Archive (S3 Glacier Deep Archive), 适用于以最低的云存储成本进行长期归档和数字保存。

访问频率低

■ 对象存储 (S3)、块存储 (EBS)、文件存储 (EFS) 具体是存什么呢?

类型	AWS 服务	本质在“存什么”
对象存储	Amazon S3	文件本身 (一个个独立文件)
块存储	Amazon EBS	磁盘本身 (给服务器用的硬盘)
文件存储	Amazon EFS	共享文件夹 (多人/多机共用)

28▼ S3 Glacier (存储 长期但很少访问 的数据)

★★★ Amazon S3 Glacier

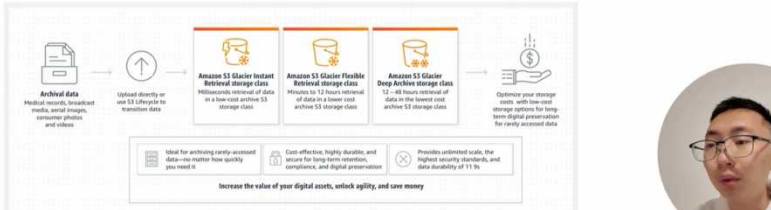
Amazon S3 Glacier 是 Amazon S3 的一种存储类别，专门用于长期存储和归档数据，具有低成本、高可靠性和安全性的特点。Glacier 主要用于存储需要长期保存但很少需要访问的数据。

目前S3 Glacier有3种存储类：

Amazon S3 Glacier Instant Retrieval 即时检索存储类，即时检索为每季度访问一次且需要毫秒级检索的长期数据提供成本最低的存储。成本最多降低 68%。

Amazon S3 Glacier Flexible Retrieval 灵活检索存储类，为每年访问 1-2 次且异步检索的归档数据提供低成本存储，成本最多降低 10%。

Amazon S3 Glacier Deep Archive 深度存档存储类，提供最低成本的存储，比灵活检索低 75%，适用于每年访问少于一次且异步检索的长期归档数据。



案例展示

Case 1: 数据归档

一家法律公司需要对其大量的案件文件进行长期归档，并确保这些数据安全可靠。他们选择使用 Amazon S3 Glacier 来存储归档文件。通过将文件存储在 Glacier 中，他们可以用极低的成本存储大量数据，并且可以根据需要随时检索和访问这些归档文件。

Case 2: 科学研究数据存储

一个科学研究机构需要存储其历史上产生的大量科学数据，并确保这些数据长期保存和可靠。他们选择使用 Amazon S3 Glacier 来存储科学研究数据。通过将数据存储在 Glacier 中，他们可以用低廉的价格存储大规模的数据，并且可以利用 Glacier 提供的安全性和持久性来确保数据的安全和完整性。

Case 3: 数字存档

一个图书馆或博物馆需要对其珍贵的数字文物进行长期存储和保存。他们选择使用 Amazon S3 Glacier 来存储这些数字存档。通过将数字文物存储在 Glacier 中，他们可以用极低的成本保存大量的数字文物，并且可以通过 Glacier 提供的高度可靠性和安全性来确保这些文物的长期保存。

29▼ Storage Gateway (混合云网关——【本地-云】互相传输【备份文件】 | 本地部署 SG + 云端与 S3 集成)

- 混合云：本地 备份到 云端、或 云端 备份到 本地 —— (类似 百度云盘备份文件)
- 关键词：混合云 → 本地部署 SG + 云端与 S3 集成；提供几乎无限的云存储访问权限。

* 可为本地应用程序提供对存储在 AWS 云中的数据的高延迟访问。它使本地应用程序能够访问和处理云中的数据，确保数据具有最小的延迟可用。

★★★ AWS Storage Gateway

AWS Storage Gateway 是一项服务，它允许您在本地环境和 AWS 云之间无缝地集成存储解决方案。它提供了一种简单、安全的方式，让您可以将本地应用程序连接到云存储服务，如 Amazon S3、Amazon Glacier、Amazon EBS 等。

案例展示

Case 1: 混合云备份

一家中型企业需要一种备份解决方案，以确保关键数据在本地和云端之间具有高可用性和冗余性。他们使用 AWS Storage Gateway 部署了一个混合云备份解决方案。他们将 Storage Gateway 部署在本地数据中心，并配置了备份策略，将数据备份到 Amazon S3。这样，他们可以通过 AWS 云服务来管理备份数据，并在需要时轻松地恢复数据到本地或云端。

Case 2: 本地文件共享

一家小型工程公司需要一个简单而有效的文件共享解决方案，以便员工在不同的办公地点之间共享和访问工程图纸和文件。他们使用 AWS Storage Gateway 部署了一个本地文件共享解决方案。他们将 Storage Gateway 部署在本地办公室，并将其配置为与 Amazon S3 集成。这样，员工可以通过本地网络访问文件共享，并且所有数据都存储在 AWS 云中，具有高可用性和可靠性。

Case 3: 灾难恢复

一家大型企业需要一个可靠的灾难恢复解决方案，以确保其业务连续性和数据安全性。他们使用 AWS Storage Gateway 部署了一个灾难恢复解决方案。他们在本地数据中心和 AWS 云中各部署一个 Storage Gateway，并配置了数据复制策略，将关键数据备份到云端。这样，他们可以在发生灾难时快速恢复业务，并确保数据的安全性和可靠性。



30 ▼ AWS Backup (【云上】存储)

● 案例:

数据库备份 ;

存储卷备份 (搬家公司一点点搬) ;

服务器镜像备份 (直接屋子整个平移)



案例展示

Case1: 数据库备份

一家电子商务公司需要定期备份其关键的数据数据库, 以确保数据的安全性和可恢复性。他们使用 AWS Backup 来创建数据库备份计划, 并将其应用于他们的 Amazon RDS 数据库实例。通过 AWS Backup, 他们可以轻松地定期备份数据库, 并在需要时快速恢复数据, 从而保护其业务数据不受损失。

Case2: 存储卷备份

一个科技公司需要备份其 Amazon EC2 实例上的存储卷数据, 以确保数据的持久性和安全性。他们使用 AWS Backup 来创建存储卷备份策略, 并将其应用于他们的 Amazon EBS 卷。通过 AWS Backup, 他们可以定期备份存储卷数据, 并使用备份数据进行灾难恢复或数据迁移操作。

Case3: 服务器镜像备份

一家云服务提供商需要备份其托管的 EC2 实例镜像, 以便在需要时快速恢复整个服务器环境。他们使用 AWS Backup 来创建 EC2 实例镜像备份计划, 并将其应用于他们的 Amazon EC2 实例。通过 AWS Backup, 他们可以轻松地备份 EC2 实例镜像, 并在需要时快速恢复整个服务器环境, 从而确保业务连续性和数据安全。

■ Backup & Storage Gateway 区别:

Backup — 【云上】存储。

Storage Gateway — 网关, 【本地-云】互相传输存储。

● 31 【存储 总结】 :

只有 **1 块硬盘**, 只能在 1 个 EC2 上使用—— **EBS**

多个服务器 同时访问 1 块硬盘 —— **EFS** (共享文件)

高性能 同时访问 1 块硬盘 —— **FSx**

海量数据、低成本、经常使用 —— **S3**

不经常使用、深度归档 —— **S3 Glacier**

混合云 备份 —— **Storage Gateway**

云上 备份 —— **Backup**

○ 【数据库】

PART



- ★★★★ Amazon RDS
- ★★★★ Amazon Aurora
- ★★★★ Amazon DynamoDB
- ★★★★ Amazon MemoryDB for Redis
- ☆☆☆ Amazon Neptune

32 ▼ Amazon RDS (关系型数据库)

- 把 RDS 运维托管给 AWS。

★★★★ Amazon RDS

Amazon Relational Database Service (RDS) 是亚马逊提供的一种托管关系型数据库服务，可用于在云中轻松设置、运行和扩展关系型数据库。

目前支持：

RDS for MySQL

RDS for PostgreSQL

RDS for MariaDB

Amazon Aurora

RDS for SQL Server

RDS for Oracle

RDS for Db2



案例展示

Case1: 电子商务网站

一家电子商务公司需要一个可靠的数据库解决方案来存储其产品信息、订单数据和用户信息。他们选择使用 Amazon RDS 来托管他们的数据库。他们创建了一个 MySQL 或 PostgreSQL 数据库实例，并将其用于存储产品目录、订单记录和用户信息。通过 Amazon RDS，他们可以确保其数据库具有高可用性、可伸缩性和安全性，从而保证其网站的顺利运行。

Case2: 移动应用后端

一家创业公司正在开发一款新的移动应用程序，并需要一个可靠的后端数据库来存储用户数据、内容信息和应用状态。他们选择使用 Amazon RDS 来托管他们的后端数据库。他们创建了一个 Amazon Aurora 或 Microsoft SQL Server 数据库实例，并将其用于存储用户配置、应用内容和交互数据。通过 Amazon RDS，他们可以轻松地管理和扩展其后端数据库，以满足不断增长的用户需求。

Case3: 企业内部系统

一家大型企业需要一个可靠的数据库解决方案来支持其内部业务系统，如人力资源管理、财务管理和供应链管理。他们选择使用 Amazon RDS 来托管他们的企业数据库。他们创建了一个 Oracle 或 Microsoft SQL Server 数据库实例，并将其用于存储企业数据和业务应用数据。通过 Amazon RDS，他们可以确保其数据库具有高可用性、可伸缩性和安全性，从而支持其内部业务系统的顺利运行。

MySQL & Aurora 区别:

MySQL —— **开源**

Aurora —— **Amazon 自研** → **非开源** + **性能优于 MySQL** + **费用低于 MySQL** + **可自动扩缩容**

33 ▼ Aurora (兼容 + 性能和可靠性 更高 + 价格更低 + 可自动扩缩容)

- 兼容 MySQL 和 PostgreSQL 的功能;
且 **性能 (吞吐量) 和可靠性 更高**;
且 **价格 是 商业数据库的十分之一**。

★★★ Amazon Aurora

Amazon Aurora 是亚马逊自研提供的一种高性能、高可用性的关系型数据库引擎，兼容 MySQL 和 PostgreSQL，提供了与这两种数据库引擎兼容的功能，同时具有更高的性能和可用性。

Amazon Aurora 在全球范围内提供无与伦比的高性能和可用性，完全兼容 MySQL 和 PostgreSQL，而成本仅为商业数据库的十分之一。Aurora 的吞吐量是 MySQL 的 5 倍，是 PostgreSQL 的 3 倍。Aurora 拥有广泛的合规性标准和一流的安全功能。Aurora 通过使数据在 3 个可用区内持久耐用（客户只需支付 1 个副本的费用）来提供存储弹性。Aurora 的可用性高达 99.99%，跨 AWS 区域部署时，客户可以使用全球数据库访问本地读取性能。使用无服务器功能，Aurora 可在不到一秒的时间内扩展到能够处理数十万个事务的能力。

案例展示

Case 1: 电子商务网站
一家电子商务公司需要一个可靠的数据存储解决方案来存储和处理大量数据。他们选择使用 Amazon Aurora 来托管他们的数据库。他们部署了一个 Amazon Aurora MySQL 数据库实例，并配置了存储副本。通过 Amazon Aurora，他们可以获得与 MySQL 兼容的功能，并实现更高的性能和可用性，从而保证网站的稳定性和性能。

Case 2: 移动应用程序
一家移动应用程序开发人员需要一个可靠的数据库来存储和处理大量数据。他们选择使用 Amazon Aurora 来托管他们的数据库。他们部署了一个 Amazon Aurora PostgreSQL 数据库实例，并配置了存储副本。通过 Amazon Aurora，他们可以获得与 PostgreSQL 兼容的功能，并实现更高的性能和可用性，从而满足其不断增长的用户需求。

Case 3: 分析型数据库
一家数据驱动的公司需要一个可靠的数据库解决方案来存储和分析大量的业务数据。他们选择使用 Amazon Aurora 来托管他们的分析型数据库。他们部署了一个 Amazon Aurora MySQL 数据库实例，并配置了存储副本。通过 Amazon Aurora，他们可以获得与 MySQL 兼容的功能，并实现更高的性能和可用性，从而提高数据分析和决策的效率和准确性。

有服务器 & 无服务器 概念:

【有服务器】 本义 = 指需要自己管理服务器 ≈ 手动预设

——手动 Provision (预置) 或管理 EC2 实例。

【有服务器化】 ≈ 预定付费

——时间段 A 内定量，不管时间段 A 内有没有使用，都按时间付钱。

【无服务器】 本义 = 指不需要自己管理服务器 ≈ 自动计算

——AWS 自动处理计算资源的伸缩和维护，无需 Provision (预置) 或管理 EC2 实例。

【无服务器化】 ≈ 按需付费

——用多少，付多少钱。

34 ▼ DynamoDB (无服务器+NoSQL(键值对型)+完全托管+任何规模下均个位数毫秒级的性能)

- 无服务器+NoSQL+完全托管+任何规模下均个位数毫秒级的性能
- 冷启动——即 瞬间启动
- DynamoDB 是 **键值对型数据库 (key-value)**

冷启动和热启动好像是在胡扯，感觉类似于热插拔那种，不关机插拔，表示还有状态，个人理解这边冷启动就是表示没有状态的，热启动就是表示有状态的环境下

★★★ Amazon DynamoDB

无服务器, NoSQL, 完全托管的数据库, 在任何规模下均具有个位数毫秒级的性能

Amazon DynamoDB 是一种无服务器的 NoSQL 数据库服务，您可以通过它来开发任何规模的现代应用程序。作为无服务器数据库，您只需按使用量为其付费，DynamoDB 可以扩展到零，没有冷启动，没有版本升级，没有维护窗口，没有修补，也没有停机维护。DynamoDB 提供一系列广泛的安全控制措施和合规性标准。对于全球分布式应用程序，DynamoDB 全局表是一个多区域、多活动数据库，具有 99.999% 的可用性 SLA 和更高的弹性。托管备份、时间点恢复等功能有助于确保 DynamoDB 的可靠性。借助 DynamoDB 流，您可以构建无服务器的事件驱动型应用程序。

案例展示

Case 1: 实时数据存储
一家社交媒体公司需要一个实时数据存储解决方案，用于存储用户的实时活动和社交数据。他们选择使用 Amazon DynamoDB 来存储这些实时数据。他们创建了一个 DynamoDB 表，并将用户的活动数据写入到表中。通过 DynamoDB，他们可以轻松地处理高并发和延迟的数据写入和读取，从而支持实时数据分析和处理。

Case 2: 游戏数据存储
一家游戏开发公司需要一个可靠的数据库解决方案，用于存储游戏玩家的数据。他们选择使用 Amazon DynamoDB 来存储游戏数据。他们创建了一个 DynamoDB 表，并将游戏玩家的数据写入到表中。通过 DynamoDB，他们可以轻松地处理高并发的游戏数据写入和读取，从而确保游戏玩家的数据安全和可靠。

Case 3: 物联网设备数据存储
一家智能家居公司需要一个可靠的数据库解决方案，用于存储物联网设备生成的大量数据。他们选择使用 Amazon DynamoDB 来存储物联网设备数据。他们创建了一个 DynamoDB 表，并将设备生成的数据写入到表中。通过 DynamoDB，他们可以轻松地处理大规模的设备数据写入和读取，并且可以根据需要扩展存储容量，以应对数据的增长。

■ 关系型 DB & 非关系型 DB 概念:

关系型 DB

——表格, 行列 11 对应。

——适用关系型 DB: **用户订单数据**

非关系型 DB

——见下图多种都是

——适用非关系型 DB: **点赞评论、游戏进度成就、设备状态配置**

* **【内存型数据库】是 NoSQL 的一种。** ▼



35▼ MemoryDB for Redis (内存型 NoSQL; 兼容 Redis, 可实现超快性能)

- 案例: 实时数据分析; 会话存储(临时状态存储); 高速缓存
- 做题技巧: 见到需要【**低延迟、缓存**】类型的 DB, 就选**内存型数据库**。

★★★ Amazon MemoryDB for Redis

与 Redis 兼容且持久的内存数据库服务, 可实现超快性能

Amazon MemoryDB for Redis 是亚马逊提供的一种完全托管的内存数据库服务, 兼容 Redis 协议, 专门用于高性能、低延迟的实时数据处理和缓存应用场景。MemoryDB 是一个持久的数据库, 具有每秒级读取、低至个位数的毫秒级写入, 可扩展性和企业级安全性。MemoryDB 可提供 99.99% 的可用性和近乎即时的恢复, 不会丢失任何数据。

- 扩展到每秒数百万个请求和每个集群超过 100TB 的存储。
- 使用多可用区事务日志持久存储数据, 可实现 99.99% 的可用性和近乎即时的恢复, 而不会丢失数据。
- 使用 Redis 数据结构、丰富的开源 API 快速构建应用程序, 并轻松与其他 AWS 服务集成。

案例展示

Case 1: 实时数据分析

一家电商公司需要进行实时数据分析, 以了解其用户行为、交易模式和产品趋势。他们选择使用 Amazon MemoryDB for Redis 来存储实时数据并进行分析。他们构建了一个 MemoryDB 实例, 并将用户活动和交易数据写入到 Redis 数据结构中。通过 MemoryDB for Redis, 他们可以实时获取和处理数据, 从而支持实时数据分析和业务决策。

Case 2: 会话存储

一个在线游戏平台需要一个可靠且可扩展的会话存储解决方案, 以存储玩家的会话信息和游戏数据。他们选择使用 Amazon MemoryDB for Redis 来存储会话数据。他们构建了一个 MemoryDB 实例, 并将玩家的会话信息和游戏数据写入到 Redis 中。通过 MemoryDB for Redis, 他们可以实现高吞吐量的会话存储和快速的数据检索, 从而提升游戏体验。

Case 3: 高速缓存

一家在线零售服务提供商需要一个高性能的缓存解决方案, 以存储商品目录和商品信息。他们选择使用 Amazon MemoryDB for Redis 来存储商品目录。他们构建了一个 MemoryDB 实例, 并将商品目录数据写入到 Redis 中。通过 MemoryDB for Redis, 他们可以实现低延迟的数据访问和快速的数据更新, 从而提升网站的性能和用户体验。

36▼ Neptune (高性能图形 NoSQL; 高关联度; 存储数十亿关系, 查询延迟降到毫秒级)

- 高性能图形 NoSQL; 高关联度; 存储数十亿关系, 查询延迟降到毫秒级
- 案例: 社交网络分析 (分析关联度 → 个性推荐); 知识图谱

☆☆☆ Amazon Neptune

Amazon Neptune 是一项快速、可靠且完全托管的图形数据库服务, 可帮助您轻松构建和运行使用图数据库的应用程序。Amazon Neptune 的核心是专门构建的高性能图形数据库引擎, 它进行了优化以存储数十亿个关系并将图形查询延迟降低到毫秒级。

案例展示

Case 1: 社交网络分析

一家社交媒体公司需要进行社交网络分析, 以了解用户之间的关系和互动模式。他们选择使用 Amazon Neptune 来存储和分析社交网络数据。他们构建了一个 Neptune 实例, 并将用户关系和互动数据写入到 Neptune 中。通过 Neptune, 他们可以实现高效的社交网络分析, 从而提升用户体验和运营效率。

Case 2: 推荐系统

一家电子商务公司需要实现个性化推荐功能, 以帮助用户发现和购买相关产品。他们选择使用 Amazon Neptune 来构建推荐系统。他们构建了一个 Neptune 实例, 并将用户浏览历史和购买记录写入到 Neptune 中。通过 Neptune 提供的图形查询功能, 他们可以分析用户之间的相似性和产品之间的关联性, 从而提升用户的购物体验。

Case 3: 知识图谱

一家研究机构需要构建一个知识图谱, 以存储和管理跨学科领域的知识关联信息。他们选择使用 Amazon Neptune 来构建知识图谱。他们构建了一个 Neptune 实例, 并将跨学科领域的知识关联信息写入到 Neptune 中。通过 Neptune 提供的图形查询功能, 他们可以分析知识关联信息, 从而提升科学研究的质量和效率。

●37 【数据库 总结】：

PART



企业·用户管理 —— **RDS**

性能高、费用低 —— **Aurora**

游戏进度、点赞评论 → **键值对 DB** —— **DynamoDB**

会话、状态信息 → **内存型 DB** —— **MemoryDB for Redis**

【**关联度**】企业关系网、个性化推荐、知识图谱 → **图 DB** —— **Neptune**

○ 【容器】

PART



★★★Amazon Elastic Container Service (Amazon ECS)

★★★Amazon Elastic Kubernetes Service (Amazon EKS)

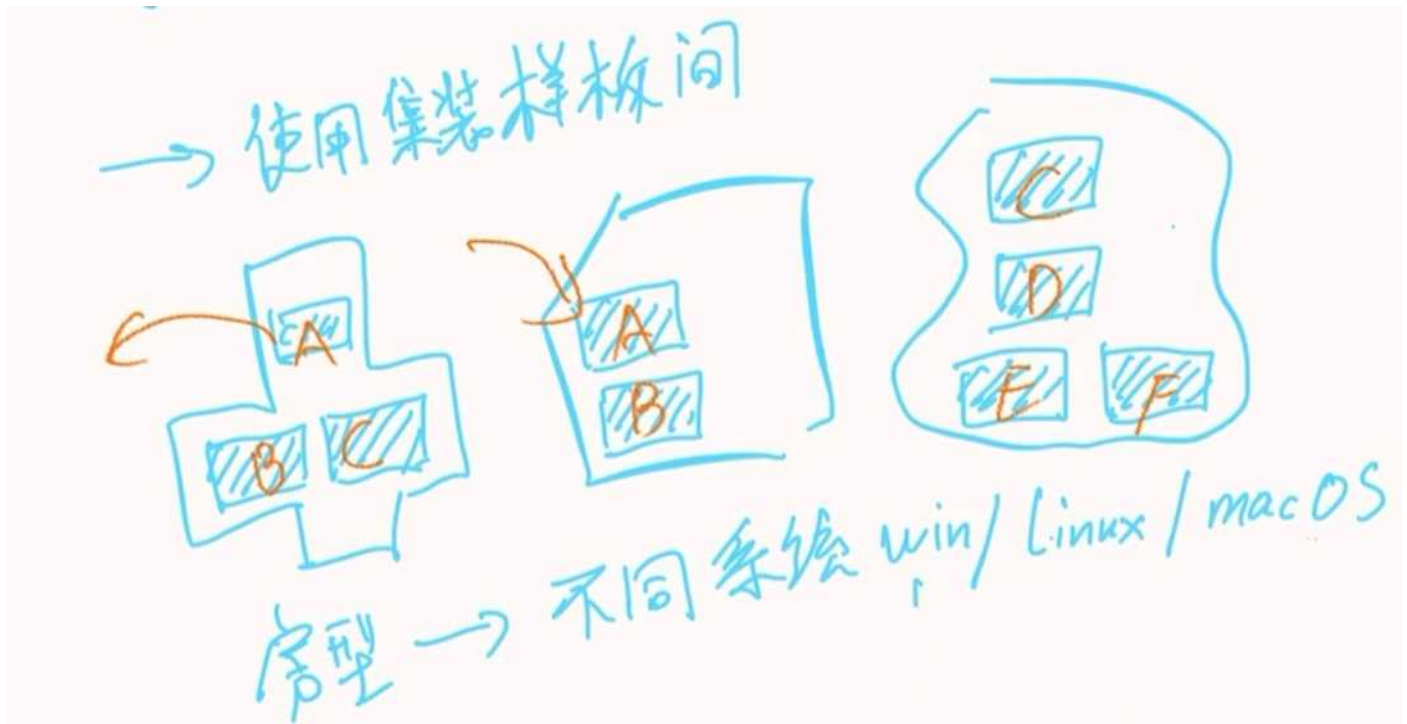
★★★Amazon Elastic Container Registry (Amazon ECR)

★★★AWS Fargate

■ 概念：什么是【容器】？

答：容器 = 把“应用 + 运行环境”一起打包，在哪都能用同样方式运行

标准定义：



38 ▼ ECS (=云上托管的 Docker 容器——打包应用进容器)

- Elastic Container Service
- 案例：把微服务打包为 Docker 容器来统一编排和管理、持续集成和交付(CI/CD)、大规模网站托管

案例展示

Case1: 微服务架构
一家互联网公司正在采用微服务架构构建其应用程序，他们需要一种可靠的容器编排服务来管理各种微服务。他们选择使用 Amazon ECS 来托管他们的微服务，他们将每个微服务打包为 Docker 容器，并在 Amazon ECS 上创建任务定义和服务定义来运行和扩展这些微服务。通过 Amazon ECS，他们可以实现高可用性和弹性的微服务架构，并能够快速部署和管理其应用程序。

Case2: 持续集成和持续交付 [CI/CD]
一家软件开发团队需要一个持续集成和持续交付 [CI/CD] 平台来自动化构建、测试和部署他们的应用程序。他们选择使用 Amazon ECS 来构建他们的 CI/CD 环境。他们将应用程序的代码存储在代码仓库中，并使用 CI/CD 工具 [如 AWS CodePipeline 和 AWS CodeBuild] 来触发构建和部署流水线。通过 Amazon ECS，他们可以将每个版本的应用程序打包为 Docker 容器，并在 Amazon ECS 上进行部署和运行，从而实现持续集成和持续交付。

Case3: 大规模网站托管
一家在线零售商需要一个可靠的托管解决方案来支持其高流量的电子商务网站。他们选择使用 Amazon ECS 来托管他们的网站。他们将网站的前端和后端应用程序打包为 Docker 容器，并在 Amazon ECS 上创建任务定义和服务定义来运行和扩展这些应用程序。通过 Amazon ECS，他们可以实现自动化的容器扩展和负载均衡，以应对不断增长的流量，并能够快速部署新的功能和更新。

统一编排和管理

★★★ Amazon Elastic Container Service (Amazon ECS)

Amazon ECS (Elastic Container Service) 是亚马逊提供的一种完全托管的容器编排服务，用于在云中运行、扩展和管理 Docker 容器化的应用程序。

Docker → ECS

39 ▼ EKS (=云上托管的 K8S——编排容器；管理调度容器，简化跨环境应用管理)

- Elastic Kubernetes(K8s) Service
- 案例：容器编排服务来管理微服务；持续集成和交付(CI/CD)自动化构建、测试、部署应用；混合云部署(托管工作负载、管理 K8S 集群，统一容器管理和部署，简化跨环境应用管理)

案例展示

Case1: 微服务架构
一家软件公司正在采用微服务架构构建其应用程序，他们需要一种可靠的容器编排服务来管理各种微服务。他们选择使用 Amazon EKS 来托管他们的微服务，他们将每个微服务打包为容器，并在 Amazon EKS 上创建 Kubernetes Pod 和 Service 来运行和扩展这些微服务。通过 Amazon EKS，他们可以实现高可用性和弹性的微服务架构，并能够快速部署和管理其应用程序。

Case2: 持续集成和持续交付 [CI/CD]
一家软件开发团队需要一个持续集成和持续交付 [CI/CD] 平台来自动化构建、测试和部署他们的应用程序。他们选择使用 Amazon EKS 来构建他们的 CI/CD 环境。他们将应用程序的代码存储在代码仓库中，并使用 CI/CD 工具 [如 Jenkins 或 GitLab CI] 来触发构建和部署流水线。通过 Amazon EKS，他们可以将每个版本的应用程序打包为容器，并在 Kubernetes 集群上进行部署和运行，从而实现持续集成和持续交付。

Case3: 混合云部署
一家大型企业需要一个可靠的容器管理平台来在混合云环境中管理其应用程序。他们选择使用 Amazon EKS 来托管他们的容器工作负载。他们在 Amazon EKS 上创建了一个 Kubernetes 集群，并将其与他们在本地数据中心或其他云提供商的 Kubernetes 集群集成。通过 Amazon EKS，他们可以实施统一的容器管理和部署，从而简化跨多个环境的应用程序管理。

Docker → ECS
K8S → EKS

★★★ Amazon Elastic Kubernetes Service (Amazon EKS)

借助 Amazon Elastic Kubernetes Service (Amazon EKS)，您可以在亚马逊云上轻松创建、管理和扩展容器化应用程序。

编排
Docker → ECS
Kubernetes → EKS
容器编排服务
容器编排服务

40 ▼ Fargate (无服务器容器计算引擎；并列概念是 EC2 弹性计算云)

- Fargate 含义：无服务器容器托管 / 无服务器计算引擎
- ECS 的 EC2 托管 → 云主机 [要自己管理 → 有更大自主可控性]
- ECS 的 Fargate 托管 → 无服务化 [无需自己管理 → 自动部署、扩展、负载均衡]

★★★ AWS Fargate

适用于容器的无服务器计算

AWS Fargate 是亚马逊提供的一种容器管理服务，它允许开发人员在 AWS 云中轻松运行容器化应用程序，而无需管理底层的服务器基础架构。Fargate 可以自动管理容器的部署、扩展和负载均衡，并提供了与 ECS (Elastic Container Service) 和 EKS (Elastic Kubernetes Service) 等 AWS 容器服务集成的能力。

ECS → EC2 → 云主机 → 自主管理
ECS → Fargate → 无服务化 → 无需管理

- 解题技巧：微服务 or 持续集成/交互
- 见【微服务架构部署】和【持续集成/持续交互(CI/CD)】
- 就想到【Docker → ECS】【K8S → EKS】

■ 区分: EC2 & Elastic Beanstalk & Lambda

- EC2 ——完全 **自主可控**
- Elastic Beanstalk ——帮助选择好对应的基础设施, 管理还是**自己管理**。
- Lambda ——只负责**代码的简易部署** (适用于**事件触发**类型)
- ECS ——云上 **容器**
- EKS ——云上 **容器编排**

案例展示

Case1: 微服务架构部署

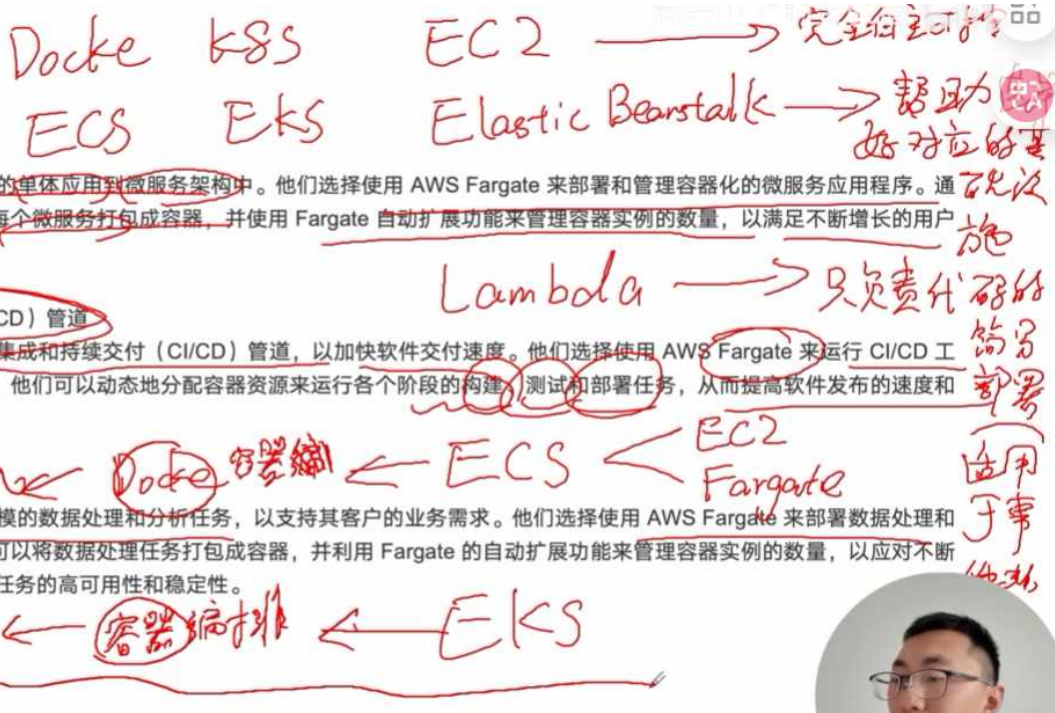
一家互联网企业正在迁移其传统的单体应用到微服务架构中。他们选择使用 AWS Fargate 来部署和管理容器化的微服务应用程序。通过 Fargate, 他们可以轻松地~~将每个微服务打包成容器~~, 并使用 Fargate 自动扩展功能来管理容器实例的数量, 以满足不断增长的用户需求。

Case2: 持续集成/持续交付 (CI/CD) 管道

一家软件开发团队正在建立持续集成和持续交付 (CI/CD) 管道, 以加快软件交付速度。他们选择使用 AWS Fargate 来运行 CI/CD 工具和流水线任务。通过 Fargate, 他们可以动态地分配容器资源来运行各个阶段的构建、测试和部署任务, 从而提高软件发布的速度和效率。

Case3: 数据分析和处理

一家数据科技公司需要进行大规模的数据处理和分析任务, 以支持其客户的业务需求。他们选择使用 AWS Fargate 来部署数据处理和分析作业。通过 Fargate, 他们可以将数据处理任务打包成容器, 并利用 Fargate 的自动扩展功能来管理容器实例的数量, 以应对不断增长的数据处理需求, 同时确保任务的高可用性和稳定性。



41 ▼ ECR (=仓库 →注册表的托管)

- Elastic Container **Registry** (注册表、登记处)
- 案例:
 - ▲ 容器镜像存储和部署、
 - ▲ 持续集成/交付 (commit → 推送到 ECR → 镜像构建到云端 Docker)、
 - ▲ 多环境部署 (不同环境 存储到 不同镜像仓库 → 选择不同的、更稳定的版本镜像, 进行版本控制管理)

* Docker 镜像 = 环境

案例展示

Case1: 容器化应用部署

一家软件开发团队正在将他们的应用程序容器化, 并需要一个可靠的容器镜像注册表来存储和管理他们的 Docker 镜像。他们选择使用 Amazon ECR 来托管他们的容器镜像。他们将应用程序的 Docker 镜像推送到 Amazon ECR 中, 并使用 ECR 提供的权限控制和版本管理功能来管理镜像的访问和更新。通过 Amazon ECR, 他们可以实现安全、高可用性的容器镜像存储和部署。

Case2: 持续集成/持续交付

一家软件开发团队正在构建一个持续集成和持续交付 (CI/CD) 流水线, 以实现自动化的应用程序构建、测试和部署。他们选择使用 Amazon ECR 来存储他们的 Docker 镜像, 并将其集成到他们的 CI/CD 流水线中。在每次代码提交时, CI/CD 流水线会触发 Docker 镜像的构建, 并将构建后的镜像推送到 Amazon ECR 中。通过 Amazon ECR, 他们可以实现高效的容器镜像管理和部署, 从而加快应用程序的交付速度。

Case3: 多环境部署

一家企业需要在多个环境 (如开发、测试和生产环境) 中部署他们的容器化应用程序, 并需要一个集中式的容器镜像存储解决方案来管理不同环境的镜像。他们选择使用 Amazon ECR 来存储他们的 Docker 镜像, 并为每个环境创建不同的镜像仓库。他们在开发环境中推送开发版本的镜像, 而在测试和生产环境中推送经过验证的稳定版本的镜像。通过 Amazon ECR, 他们可以实现灵活的制和环境管理, 从而简化多环境部署的流程。



●42 【容器 总结】：



- ECS ——云上 **容器** (微服务; CI/CD)
- EKS ——云上 **容器调度、编排**
- ECR ——**镜像仓库**
- Fargate ——**无服务器化容器**服务
 - ECS 的 **EC2** 托管 →云主机 [要自己管理 →有更大自主可控性]
 - ECS 的 **Fargate** 托管 →无服务化 [无需自己管理 →自动部署、扩展、负载均衡]
- **EC2** ——完全**自主可控**
- **Elastic Beanstalk** ——帮助**选择好**对应的**基础设施**，管理还是**自己管理**。
- **Lambda** ——只负责**代码的简易部署** (适用于**事件触发**类型)

○ 【迁移和传输】

数据库和主机 → 迁移上 → 云

或

其它云 → 迁移到 → AWS 云

PART



- ★ ★ ★ AWS Application Migration Service
- ★ ★ ★ AWS Database Migration Service (AWS DMS)
- ★ ☆ ☆ AWS Migration Hub
- ☆☆ ☆ AWS Schema Conversion Tool (AWS SCT)
- ★ ☆ ☆ AWS Snow Family
- ☆☆ ☆ AWS Transfer Family
- ☆☆ ☆ AWS Application Migration Service



43 ▼ Application Discovery Service (Discovery Agent; 管理类软件)

- 运行 Discovery Agent 收集信息 → 生成信息清单 → 洞察并管理软件

☆☆☆ AWS Application Discovery Service

AWS Application Discovery Service 可以收集有关您的本地数据中心的信息，发现本地服务器清单和行为，以帮助客户规划云迁移项目。

案例展示

Case 1: 混合部署
一家企业正在将部分本地服务器迁移到云端，以便更好地管理和扩展。他们选择使用 AWS Application Discovery Service 来收集有关本地服务器的信息。通过安装本地代理 (Discovery Agent)，他们可以轻松地发现和评估本地服务器。然后，他们可以使用 AWS Migration Hub 来规划迁移。

Case 2: 无代理发现
一家企业正在将本地服务器迁移到云端，以便更好地管理和扩展。他们选择使用 AWS Application Discovery Service 来收集有关本地服务器的信息。通过安装本地代理 (Discovery Agent)，他们可以轻松地发现和评估本地服务器。然后，他们可以使用 AWS Migration Hub 来规划迁移。

Case 3: 混合部署
一家企业正在将本地服务器迁移到云端，以便更好地管理和扩展。他们选择使用 AWS Application Discovery Service 来收集有关本地服务器的信息。通过安装本地代理 (Discovery Agent)，他们可以轻松地发现和评估本地服务器。然后，他们可以使用 AWS Migration Hub 来规划迁移。

管理

44 ▼ AMS / MGN (【服务器】线下迁移上云: 不停机(不断同步、实时增量)迁移)

- 全称: Application Migration Service (区分 [AMS]: AWS 托管服务——Managed Services (AMS) ——托管 AWS 环境的日常运维)
- 功能: 将线下服务器迁移上云端——AMS
- 迁移方式: 进行整机的(服务器)不停机(不断同步、实时增量)迁移

☆☆☆ AWS Application Migration Service

AWS Application Migration Service (AWS Application Migration Service 的服务名称缩写为 AWS MGN。"MGN"是单词"migration" (迁移) 的缩写。) 是一项全面的迁移服务，旨在帮助企业将其本地服务器和虚拟机迁移到云端，包括 AWS 和 VMware Cloud on AWS。

案例展示

Case 1: 本地服务器迁移至 AWS
一家企业拥有大量的本地服务器，希望将其迁移到云端以获得更灵活的扩展和管理。他们选择使用 AWS Application Migration Service 来进行服务器迁移。通过 AWS MGN，他们可以轻松地发现和评估本地服务器。然后，他们可以使用 AWS MGN 的迁移计划和工具来自动化迁移过程，并确保迁移后的应用程序在云端顺利运行。

Case 2: VMware 环境迁移至 AWS
一家企业正在使用 VMware 虚拟化平台来运行其应用程序，希望将其迁移到云端以降低成本和提高灵活性。他们选择使用 AWS Application Migration Service 来进行 VMware 环境迁移。通过 AWS MGN，他们可以将现有的 VMware 虚拟机迁移到 AWS 中的 EC2 实例，并利用 VMware Cloud on AWS 提供的功能来实现混合云环境的管理和运维。

Case 3: 迁移至 AWS 的容器化应用
一家企业正在将其应用程序容器化，并希望将其迁移到云端以实现更高的弹性和可扩展性。他们选择使用 AWS Application Migration Service 来进行容器化应用迁移。通过 AWS MGN，他们可以将容器化的应用程序部署到 AWS 中的 ECS 或 EKS 集群，并利用 AWS MGN 提供的迁移工具来简化迁移过程。他们还可以利用 AWS MGN 的监控和调优功能来优化迁移后的应用程序性能和成本。



■ 补充: Discovery Agent 的【服务器迁移】过程

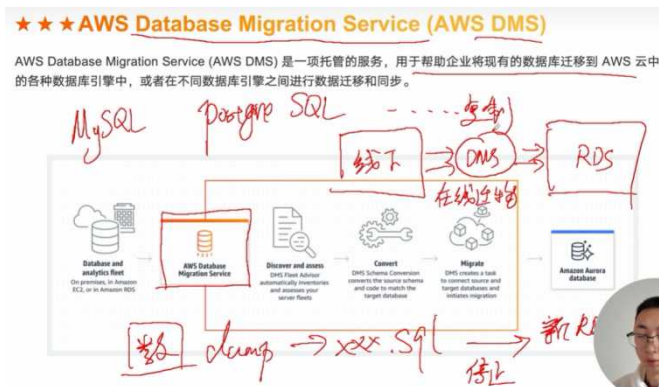
【Agent】——不断地收集新增数据到云中 EC2

- 发现 云中 EC2 和 线下数据 一致时
- 切断 Agent 和 EC2 的连接
- 把用户流量引导到 EC2 中去
- 完成【服务器迁移】

45 ▼ DMS (【数据库】在线迁移)

- Database Migration Service
- 【数据库】在线迁移—— 在线 = 不停机
- 案例: 跨数据库引擎迁移 (源端→目标端); 数据库版本升级; 跨云迁移 ↓

迁移 (用 DMS)	迁移原因	源端	目标端
跨 数据库引擎	Aurora 高可用、可扩展	MySQL	AWS 的 Aurora
数据库 版本升级	数据量大→手动成本和风险高	旧版本 Oracle	
跨 云	服务质量、费用	某云 (运行着 SQL Server)	AWS 的 RDS SQL Server



案例展示

数据库迁移工具

Case1: 跨数据库引擎迁移
一家企业决定将其现有的 MySQL 数据库迁移到 AWS 的 Aurora PostgreSQL 数据库中, 以利用 Aurora 的高性能和可扩展性。他们选择使用 AWS Database Migration Service (AWS DMS) 来进行跨数据库引擎的迁移。通过 AWS DMS, 他们可以配置 MySQL 数据库作为源端, Aurora PostgreSQL 数据库作为目标端, 并设置迁移任务以将数据从源端复制到目标端。AWS DMS 将会自动处理模式转换和数据迁移, 并确保数据的一致性和完整性。

Case2: 数据库版本升级
一家公司使用较旧版本的 Oracle 数据库。他们决定升级到最新版本以获得更好的性能和安全性。然而, 由于数据量庞大, 手动升级成本和风险较高。因此, 他们选择使用 AWS Database Migration Service (AWS DMS) 来进行数据库版本升级。通过 AWS DMS, 他们可以将旧版本的 Oracle 数据库作为源端, 新版本的 Oracle 数据库作为目标端, 并使用迁移任务将数据从源端复制到目标端。AWS DMS 可以确保在迁移过程中数据的一致性和完整性, 使得升级过程更加安全和可靠。

Case3: 跨云迁移
一家公司在另一个云提供商上运行其关键业务的 SQL Server 数据库, 但由于服务质量问题和高昂的费用, 他们决定将数据库迁移到 AWS 云上的 RDS SQL Server 中。为了实现跨云迁移, 他们选择使用 AWS Database Migration Service (AWS DMS)。他们将源端数据库作为源端, RDS SQL Server 数据库作为目标端, 并创建迁移任务以将数据从源端复制到目标端。通过 AWS DMS, 他们能够以简单、快速和可靠的方式完成跨云迁移, 并确保数据的一致性和完整性。

46 ▼ Migration Hub (集中化控制台, 不是产品)

- 【Hub】: 中转的仓库。
- 案例:
 - 多种迁移任务管理 (程序、数据库、服务器 的迁移)、
 - 跨多个 AWS 服务迁移 (EC2、RDS、S3 的迁移)、
 - 合规性报告 (迁移任务状态、进度、成本等的详细报告)



案例展示

2-3 → 几个 上百个

Case1: 多种迁移任务管理
一家企业计划在亚马逊云中进行多个迁移项目, 包括应用程序迁移、数据库迁移和服务器迁移等。他们选择使用 AWS Migration Hub 来管理这些迁移任务。通过 Migration Hub, 他们可以在一个集中的控制台中查看所有迁移任务的状态和进度, 轻松地跟踪每个迁移项目的情况, 并监控整个迁移过程的整体进展。

Case2: 跨多个 AWS 服务迁移
一家公司正在考虑将其现有的基础设施从本地数据中心迁移到亚马逊云中的多个 AWS 服务中, 如 EC2、RDS、S3 等。他们选择使用 AWS Migration Hub 来统一管理这些迁移活动。通过 Migration Hub, 他们可以轻松地创建和监控不同 AWS 服务之间的迁移任务, 并确保各个迁移项目按计划进行, 从而最大程度地降低迁移风险和成本。

Case3: 合规性和报告
一家企业需要符合行业标准和法规要求, 并需要生成详细的迁移报告以便审核和审计。他们选择使用 AWS Migration Hub 来帮助实现合规性和报告需求。通过 Migration Hub, 他们可以生成包括迁移任务状态、进度和成本等在内的详细报告, 以及跟踪整个迁移过程中的事件和变更。这些报告可以帮助他们证明其迁移活动符合相关的合规性要求, 并提供给审计人员进行审查。

47 ▼ SCT (数据库迁移工具: 换数据库引擎)

• Schema Conversion(转换) Tool

Schema(模式、架构)

• 案例: Oracle→RDS PostgreSQL; SQL Server→Aurora; IBM Db2→Redshift(数据仓库)

• 数据库 (小) ——小组小型 (冰箱、书柜)

数据仓库 (大) ——企业大型 (图书馆)

☆☆☆ AWS Schema Conversion Tool (AWS SCT)

AWS Schema Conversion Tool (AWS SCT) 是一款用于数据库迁移的工具, 旨在帮助用户将现有的数据库模式 (包括表、视图、存储过程等) 从一个数据库引擎转换到另一个数据库引擎。

AWS 提供两种模式转换解决方案, 使异构数据库迁移可预测、快速、安全且简单。客户可以选择: 1) 登录 AWS Database Migration Service (AWS DMS) 控制台以启动 AWS DMS Schema Conversion (DMS SC) 工作流程, 从而获得完全托管的体验; 或 2) 将 AWS Schema Conversion Tool (AWS SCT) 软件下载到他们的本地驱动器。

案例展示

Case1: Oracle 到 Amazon RDS PostgreSQL 的迁移
一家企业决定将其原有的 Oracle 数据库迁移到 Amazon RDS 的 PostgreSQL 数据库中, 以节省成本并提高性能。他们选择使用 AWS Schema Conversion Tool (AWS SCT) 来帮助他们的迁移。通过 AWS SCT, 他们可以自动将 Oracle 数据库模式转换为 PostgreSQL 兼容的模式。首先, 他们使用 SCT 工具检查源数据库的状态。然后, 他们使用 AWS SCT 提供的工具将源数据库迁移到 Amazon RDS 的 PostgreSQL 实例中, 完成整个迁移过程。

Case2: Microsoft SQL Server 到 Amazon Aurora MySQL 的迁移
一家公司正在考虑将其基于 Microsoft SQL Server 的数据库迁移到 Amazon Aurora 的 MySQL 数据库中, 以实现更好的性能和可扩展性。他们选择使用 AWS Schema Conversion Tool (AWS SCT) 来进行迁移。通过 AWS SCT, 他们可以自动将 SQL Server 数据库模式转换为 MySQL 兼容的模式。首先, 他们使用 SCT 工具检查源数据库的状态。然后, 他们使用 AWS SCT 提供的工具将源数据库迁移到 Amazon Aurora 的 MySQL 实例中, 完成整个迁移过程。

Case3: IBM Db2 到 Amazon Redshift 的迁移
一家企业正在考虑将其原有的 IBM Db2 数据库迁移到 Amazon Redshift 数据仓库中, 以支持其大数据分析需求。他们选择使用 AWS Schema Conversion Tool (AWS SCT) 来进行迁移。通过 AWS SCT, 他们可以自动将 Db2 数据库模式转换为 Redshift 兼容的模式。首先, 他们使用 SCT 工具检查源数据库的状态。然后, 他们使用 AWS SCT 提供的工具将源数据库迁移到 Amazon Redshift 实例中, 完成整个迁移过程。

48 ▼ Snow Family (方便部署 & 高速 & 量大的物理传输家族)

• 做题 关键词: 离线环境/网络连接不稳定 [物理]、大量数据迁移 [量大 & 高速]

• Snow Family (家族)

——Snowcone、Snowball、

Snowball Edge (单台 80TB → 适合迁移数百 TB)、

Snowmobile (50PB ~ 最高 100PB)

• 简记:

Snowball = 大号移动硬盘 —— 巨大量数据 (PB 级)、高速传输到云端的 “移动硬盘”

Snowcone —— 任何地方部署、超便捷的数据传输和边缘计算

• 下单方式:

AWS 下单填写容量 (30T/50T/80T) → 物理寄到公司 → 拷贝数据到 Snowball

→ 打电话运走 Snowball → AWS 负责把数据拷到云端

• 基础补充: MB → GB → TB → PB (每级 1000 倍关系)

☆☆☆ AWS Snow Family 家族

AWS Snow Family 是一系列物理设备, 用于在边缘环境或断开的网络环境中进行 PB 级数据收集、存储和处理, 然后将数据安全地传输到亚马逊云中进行进一步的处理和分析。

目前提供:

- Snowcone 在任何地方部署超便捷数据传输和边缘计算
- Snowball 将离线数据或远程存储快速移动到云端

案例展示

Case1: 边缘计算
一家公司在偏远地区运营着一个大型的工厂, 但由于该地区的网络连接不稳定, 无法直接将数据上传到云端进行分析。为了解决这个问题, 他们选择使用 AWS Snow Family 中的 AWS Snowcone 设备。他们将 Snowcone 设备部署在工厂现场, 用于收集工厂中的传感器数据和设备数据, 并在本地对数据进行初步处理。然后, 他们使用 Snowcone 设备上的加密功能将数据安全地存储, 并在网络连接稳定时将数据上传到亚马逊云中进行进一步的分析。

Case2: 数据迁移
一家大型公司需要将大量的数据从本地数据中心迁移到亚马逊云中, 用于长期存储和分析。然而, 由于数据量巨大, 直接通过网络传输将花费大量时间和成本。因此, 他们选择使用 AWS Snowball 设备进行数据迁移。他们将 Snowball 设备运送到本地数据中心, 将数据加载到设备中, 并使用设备的加密功能保护数据安全。然后, 他们将 Snowball 设备运送到 AWS 数据中心, 将数据上传到亚马逊云中进行存储和分析。

Case3: 临时数据处理
一家研究机构需要在偏远地区进行一次临时性的数据收集和处理项目, 但由于该地区的网络连接有限, 无法直接与亚马逊云进行通信。为了解决这个问题, 他们选择使用 AWS Snowmobile 设备。他们将 Snowmobile 设备运送到现场, 用于收集和存储数据, 并进行临时性的数据处理和分析。然后, 当项目完成后, 他们将 Snowmobile 设备运回 AWS 数据中心, 将数据上传到亚马逊云中进行长期存储和分析。

49 ▼ Transfer Family (文件传输服务, 支持多种传输协议: FTP、FTPS、SFTP)

• 复习: 大容量、便宜的存储 —— 选 S3

• 做题关键词: 看到 FTP、FTPS、SFTP —— 选 Transfer Family

案例展示

Case: 文件传输服务
一家制造公司需要定期向供应商发送大量的设计文件和技术规格，以便进行生产。他们选择使用 AWS Transfer Family 来搭建一个安全的文件传输服务。他们设置了一个 Transfer Family 传输节点，并与他们的 Amazon S3 存储桶进行集成。供应商可以通过 SFTP 或 FTPS 访问该传输节点，以便快速、安全地下载所需的文件。

Case: 备份文件上传
一家医疗保健提供商需要将患者的数据备份到 Amazon S3 中，以确保数据的安全性和可移植性。他们利用 AWS Transfer Family 搭建了一个文件传输服务，并与他们的 Amazon S3 存储桶进行集成。他们设置了一个 SFTP 传输节点，医生和医院工作人员可以通过 SFTP 将患者数据文件上传到 AWS Transfer Family，并自动将其存储到 Amazon S3 中。

Case: 跨部门文件共享
一家大型企业需要其不同部门之间共享大量的业务文件和报告。为了确保文件传输的安全性和可移植性，他们选择使用 AWS Transfer Family 来搭建一个文件共享服务。他们设置了多个 SFTP 传输节点，与他们的 Amazon S3 存储桶进行集成。因为每个部门都需要了单独的用户权限，这样，不同部门的员工可以通过 SFTP 客户端轻松上传和下载文件，同时保持数据的安全性控制策略。

☆☆☆ AWS Transfer Family

AWS Transfer Family 是一项 AWS 服务，提供了一种简单、安全且可扩展的方式来传输文件。它支持多种传输协议，包括 FTP、FTPS、和 SFTP，并可以与现有的身份认证和权限管理系统集成，为用户提供强大的权限控制功能。

●50 【迁移和传输 总结】：

PART



- ☆☆☆ AWS Application Migration Service
- ☆☆☆ AWS Database Migration Service (AWS DMS)
- ☆☆☆ AWS Migration Hub
- ☆☆☆ AWS Schema Conversion Tool (AWS SCT)
- ☆☆☆ AWS Snow Family
- ☆☆☆ AWS Transfer Family
- ☆☆☆ AWS Application Migration Service



AMS(MGN) ——迁移【服务器】——不停机(不断同步、实时增量)

DMS ——迁移【数据库】

SCT ——换【数据库引擎】

Migration Hub ——迁移任务的【集中管理和调度的控制台】

ADS (Application Discovery Service) ——系统不熟或太庞大 → 用 ADS 来【发现并统计】各类应用

SnowFamily ——【物理传输】→离线没网 or 数据量大(PB、TB)

TransferFamily ——【传输】 →协议：FTP、SFTP、FTPS

○ 【联网与内容分发】

PART

联网
和
内容
分发

- ★★★Amazon VPC
- ★★★Amazon API Gateway
- ★★★Amazon CloudFront
- ★★★AWS Direct Connect
- ★★★AWS Global Accelerator
- ★★★Amazon Route 53
- ★★★AWS VPN
- ★★★Transit Gateway

51 ▼ VPC (=云中内网)

*IDC = 内网

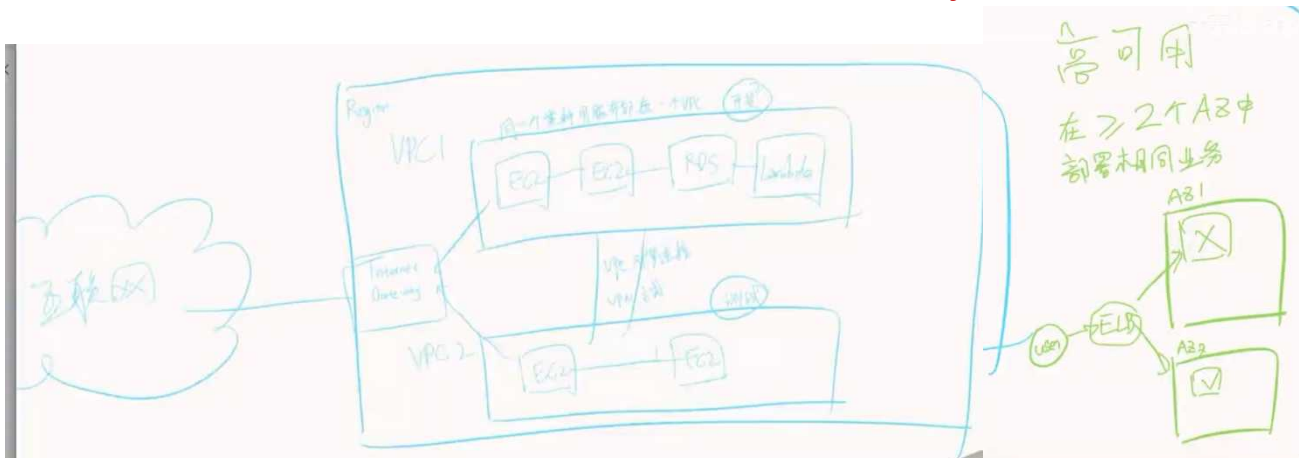
★★★Amazon VPC

Amazon Virtual Private Cloud (VPC) 是一项用于在 AWS 云中创建一个逻辑隔离的虚拟网络的服务，使用户能够在云中启动 AWS 资源（例如 EC2 实例）并将其放置在自己定义的虚拟网络中。

Amazon Virtual Private Cloud (Amazon VPC) 让您能够全面地控制自己的虚拟网络环境，包括资源放置、连接性和安全性。首先在 AWS 服务控制台中设置 VPC。然后，向其中添加资源，例如 Amazon Elastic Compute Cloud (EC2) 和 Amazon Relational Database Service (RDS) 实例。最后，您可以定义 VPC 相互之间以及跨账户、可用区或 AWS 区域通信的方式。在以下示例中，每个区域内的两个 VPC 之间共享网络流量。



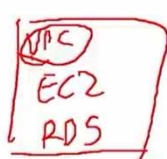
↓ 【同一种类服务】(开发、测试...)部署在同一个 VPC 下。 // 通过 Gateway (网关) 访问互联网。



总结:

① VPC 是云中的内网 云中服务在 VPC 中运行 (EC2, RDS)
 ② 同一个 VPC 内的通信是互通的, 不同 VPC 默认不互通。
 ③ 通过“对等连接”, “VPN”进行 VPC 的打通
 ④ 将业务都部署在同一 VPC 下的不同可用区, 可以提高可用
 ⑤ 安全组: 附加在 EC2 实例上控制流量进出
 NACL: 附加在子网中控制流量进出

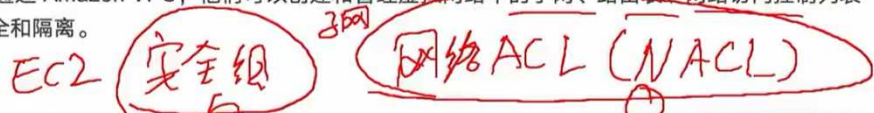
案例展示



云上网络基础

Case1: 企业内部网络扩展

一家企业正在考虑将其内部网络扩展到 AWS 云中, 并希望能够在云中运行一些应用程序或服务。他们选择使用 Amazon VPC 来创建一个虚拟网络, 并将其扩展到 AWS 云中。通过 Amazon VPC, 他们可以创建和管理虚拟网络中的子网、路由表、网络访问控制列表 (ACL) 等, 以实现对应用程序和数据的安全和隔离。



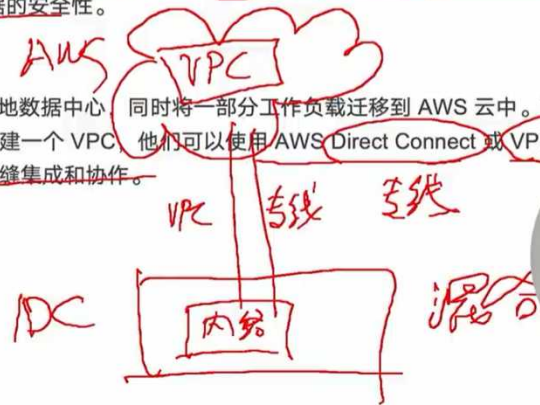
Case2: 多层应用程序部署

一家公司正在构建一个多层次的应用程序, 并希望能够在 AWS 云中部署这个应用程序。他们选择使用 Amazon VPC 来创建一个多层次的架构, 包括公共子网、私有子网和数据库子网。通过 Amazon VPC, 他们可以将不同层次的组件部署在不同的子网中, 并使用安全组和网络 ACL 来控制流量的访问, 以保护应用程序和数据的安全性。



Case3: 混合云环境连接

一家公司正在实施混合云架构, 部署了一些应用程序和服务在本地数据中心, 同时将一部分工作负载迁移到 AWS 云中。为了实现安全可靠的连接, 他们选择使用 Amazon VPC。通过在 AWS 云中创建一个 VPC, 他们可以使用 AWS Direct Connect 或 VPN 连接将本地数据中心连接到 AWS 云中的 VPC, 从而实现混合云环境的无缝集成和协作。



52 ▼ CloudFront (内容分发=CDN; 缓存至更近)

关键词:

内容分发(CDN)。

缓存到用户更近位置——加速传输、减少延迟; 提升网站性能、减轻后端服务器压力。

★★★ Amazon CloudFront

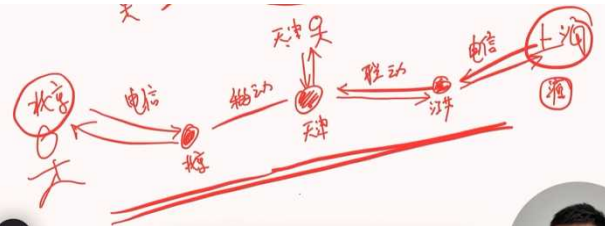
Amazon CloudFront 是一项全球性的内容分发网络 (CDN) 服务, 旨在提高 Web 内容的传输速度、安全性和可靠性, 通过将内容缓存到离用户更近的位置, 加速内容传输, 并减少延迟。

案例展示 *CDN 静态内容*

Case1: 静态网站加速
一家公司有一个静态网站, 但用户在全球各地访问速度不稳定。为了提高网站的访问速度和性能, 他们选择使用 Amazon CloudFront。他们将网站的静态内容 (如 HTML、CSS、JavaScript 文件) 缓存到 CloudFront 的边缘节点上, 并通过 CloudFront 分发内容给用户。这样, 用户可以从离他们更近的位置快速加载网站, 提升了用户体验。

Case2: 动态内容缓存
一家电子商务公司有一个动态网站, 其中包含了大量的产品信息和用户生成内容。为了提高网站的性能, 并减轻后端服务器的负载压力, 他们选择使用 Amazon CloudFront。他们将网站的动态内容 (如产品图片、用户评论等) 通过 CloudFront 缓存到边缘节点上, 然后将请求路由到最近的边缘节点, 以提供最佳的用户体验。这样一来, 用户可以快速加载动态内容, 而且后端服务器的负载得到了分担。

Case3: 视频流加速
一家视频服务提供商希望提供高质量的视频流服务, 但用户在某些地区经常遇到缓冲和加载问题。为了解决这个问题, 他们选择使用 Amazon CloudFront。他们将视频内容通过 CloudFront 缓存到全球各地的边缘节点上, 并利用 CloudFront 的流媒体分发功能来加速视频流的传输。这样, 用户可以更流畅地观看视频, 而且视频内容加载速度更快, 提升了用户满意度。



53 ▼ Direct Connect (混合云; 专线-传输快,搭建慢; 静态)

关键词:

- 【静态 使用】
- (IDC 私网-云)混合云;
- 专线连接——

传输安全、高速低延迟;
搭建专线慢且贵, 因要物理挖土拉线

★★★ AWS Direct Connect *专线 打通 IDC 和云*

AWS Direct Connect 是一项 AWS 服务, 允许用户建立私有连接从其本地数据中心、办公室或合作伙伴网络连接到 AWS 云中, 实现高性能、安全且低延迟的专用网络连接。

AWS Direct Connect 云服务是通往 AWS 资源的最短路径。传输时, 您的网络流量保持在 AWS 全球网络上, 不会接触公共互联网。这样可减少遇到瓶颈或延迟意外增加的可能。

- 构建混合网络
- 扩大现有网络
- 管理大数据集

公网不稳定, 公网不安全

A diagram showing a direct connection between an IDC (IDC) and AWS Cloud via a dedicated line (专线). The IDC is represented by a server rack icon, and the AWS Cloud is represented by a cloud icon. A red arrow labeled '专线 (物理线路)' points from the IDC to the AWS Cloud. A note says '公网不稳定, 公网不安全' (Public network is unstable and insecure). Another note says '专线' (Dedicated line).

案例展示 *安全*

Case1: 混合云连接
一家公司正在实施混合云架构, 部署了一些关键业务在本地数据中心, 同时将一部分工作负载迁移到 AWS 云中。为了实现安全可靠的混合云连接, 他们选择了 AWS Direct Connect。他们通过 AWS Direct Connect 连接了本地数据中心和 AWS 云, 使得本地应用程序可以直接访问 AWS 云中的资源, 实现了混合云环境的无缝集成和协作。

Case2: 高性能数据传输
一家科研机构需要大量的数据传输能力, 以支持其对遥感数据的处理和分析。为了实现高性能的数据传输, 并避免因公共互联网造成的延迟和不稳定性, 他们选择使用 AWS Direct Connect。通过 AWS Direct Connect, 他们可以建立高速、低延迟的专用连接, 将遥感数据直接传输到 AWS 云中进行处理和存储, 提高了数据处理的效率和质量。

Case3: 大规模应用迁移
一家大型企业计划将其大规模应用迁移到 AWS 云中以实现更好的可扩展性和灵活性。为了确保迁移过程的顺利进行, 并最大程度地减少迁移期间的业务中断, 他们选择使用 AWS Direct Connect。通过 AWS Direct Connect, 他们可以建立高速、安全的连接, 将大量的应用数据快速、可靠地迁移到 AWS 云中, 保证了迁移过程的顺利进行, 并最大程度地减少了业务中断的影响。

54 ▼ Global Accelerator (加速联网; 静+动)

关键词:

- 【静+动 使用】; AWS 自家服务
- 加速服务——路由到最近的 AWS 边缘位置

*** AWS Global Accelerator

案例展示

Case1: 全球网络应用
一家国际性企业运营着一个全球性的在线服务平台。他们的用户遍布世界各地。为了提供一致的用户体验和性能，他们选择使用 AWS Global Accelerator。通过 AWS Global Accelerator，用户可以连接到由最近边缘的 AWS 边缘位置，减少了网络延迟，并提高了访问速度。这使得用户无论位于何处，都能够快速访问到他们的服务，提升了用户满意度。

Case2: 跨地区应用负载均衡
一家跨国公司在多个地区运营着相同的应用程序，并希望将用户流量分发到全球各地的不同区域以提高性能和可用性。为了实现这一目标，他们选择使用 AWS Global Accelerator。通过 AWS Global Accelerator，他们可以轻松地配置和管理全球负载均衡，将用户流量智能地分发到最近的 AWS 区域，实现了全球应用程序的高性能和可用性。

Case3: 全球内容交付
一家媒体公司拥有大量的媒体内容，并希望其快速、可靠地分发到全球各地的用户。为了实现全球内容交付，他们选择使用 AWS Global Accelerator。通过 AWS Global Accelerator，他们可以以全球范围内的配置和管理内容分发网络 (CDN)，将内容存储在最近的 AWS 边缘位置，以提高内容的传输速度和可用性。这使得用户无论位于何处，都能够快速访问到他们需要的媒体内容。

55 ▼ Route 53 (DNS; 注册+管理+解析 域名; 全球负载均衡)

关键词: 域名系统(DNS 解析)——

- 注册+管理 域名
- 解析域名——域名→IPv4 地址
- 全球负载均衡——每周导向每周的服务器、中国导向中国服务器→减少访问时长

*** Amazon Route 53

案例展示

Case1: 网站托管
一家公司正在构建一个新的网站，并希望以低成本和高可用性的方式进行托管。他们选择使用 Amazon Route 53 来注册域名，并将用户请求路由到他们的网站。通过 Route 53 的负载均衡功能，他们可以轻松地配置多个 Amazon S3 存储桶或 EC2 实例，以确保网站的高可用性和性能。

Case2: 全球负载均衡
一家跨国企业在多个地区运营着相同的应用程序，并希望能够以全球负载均衡的方式将用户流量分发到不同地区的服务器上。他们选择使用 Amazon Route 53 的全球负载均衡功能。通过配置多个记录集并使用地理位置路由策略，他们可以智能地将用户请求路由到离用户最近的服务器，提高了用户的访问速度和体验。

Case3: 域名注册和管理
一家新创企业准备推出一个新的产品，并需要注册一个域名来进行推广。他们选择使用 Amazon Route 53 来注册域名，并管理域名相关的 DNS 设置。通过 Route 53 的简单直观的自助界面，他们可以轻松地注册新的域名、设置 DNS 记录、配置转发规则等，以实现对其域名的灵活管理和控制。

56 ▼ VPN ([IDC-云] 加密连接——混合云; 公网连接; 比专线 便宜)

- IDC-云 加密连接——混合云
- 公网连接
- 比专线 便宜

*** AWS VPN

案例展示

Case1: 远程办公接入
一家公司有许多远程员工需要访问公司的内部网络资源。为了满足这些员工的安全需求，公司决定使用 AWS VPN 来建立远程办公接入。他们在 AWS 云中创建了一个 VPN 网关，并将其与公司内部网络连接起来。这样员工可以通过 VPN 连接安全访问公司内部资源，从而实现了远程办公。

Case2: 跨区域网络互联
一家公司在多个地区部署了应用程序和服务，并希望能够将这些地区的网络连接起来，实现跨区域的网络互联。为了实现这个目标，他们选择使用 AWS VPN。通过在 AWS 云中创建 VPN 网关，并将其与本地数据中心连接起来，他们可以轻松地实现跨区域的网络互联，从而实现数据的共享和协作。

Case3: 混合云环境连接
一家公司正在实施混合云架构，即将一部分应用程序和服务迁移到本地数据中心，而将另一部分工作负载迁移到 AWS 云中。为了实现安全可靠的连接，他们需要连接本地数据中心和 AWS 云。通过在 AWS 云中创建一个 VPN 网关，并将其与本地数据中心连接起来，他们可以安全地实现混合云环境的无缝集成和协作，从而实现对应用程序和数据的安全访问和管理。

■ VPN(加密通道) & VPC(云上私网) 对比

项目	VPN	VPC
全称	Virtual Private Network	Virtual Private Cloud
本质	连接方式 / 加密通道	网络环境 / 内网 云上的“私有局域网”
解决什么	把两端网络连起来	在云上建私有网络
典型场景	远程访问	AWS 云服务器网络

57▼ Transit Gateway (中央枢纽 网关)

- **中央枢纽 网关** 连接 VPC——高度可扩展、每个新连接只要建立一次
- **案例①**: 不分云上云下——跨 VPC、VPN、IDC (即跨环境) 都可接入传输
- **案例②**: 跨 AWS 账户 VPC 连接 (即跨公司) 也可

★★★ Transit Gateway

$N = \frac{n(n-1)}{2}$

将 Amazon VPC、AWS 账户和本地网络连接到一个网关中。AWS Transit Gateway 通过中央枢纽连接 Amazon 虚拟私有云 (VPC) 和本地网络。此连接简化了您的网络，并且消除了复杂的对等关系。Transit Gateway 充当高度可扩展的云路由，每个新的连接只建立一次。

案例展示

Case 1: 跨多个 VPC 进行网络连接
一家企业在 AWS 中拥有多个 VPC，每个 VPC 用于不同的部门或应用程序。他们使用 Transit Gateway 将每个 VPC 连接到一个中心位置，以实现 VPC 之间的高速、可靠的网络通信。这种架构简化了网络配置和管理，同时提供了更好的网络性能和可扩展性。

Case 2: 连接本地数据中心
一家公司在 AWS 上托管其应用程序，并需要与本地数据中心进行混合云连接。他们使用 Transit Gateway 将 AWS VPC 和本地数据中心之间的连接集中到一个网关上。这样，他们可以轻松地管理跨云和本地环境的网络流量，并实现更好的网络可用性和性能。

Case 3: 跨账户 VPC 连接
一个组织拥有多个 AWS 账户，并且希望在这些账户之间共享网络资源。他们使用 Transit Gateway 在不同 AWS 账户的 VPC 之间建立网络连接，实现账户之间的资源共享和通信。这种跨账户的网络连接架构提高了资源的访问安全性和安全性，同时简化了管理和配置的复杂性。

58▼ API Gateway

- **API 是 (前后端交互的) 接口; API Gateway 是 接口连接。**
- **案例①**: 统一标准, 实现不同微服务之间的通信

★★★ AWS API Gateway

Amazon API Gateway 是一种完全托管的服务，可以帮助开发人员轻松创建、发布、维护、监控和保护任意规模的 API。API 是应用程序的前门，可从您的后端服务访问数据、业务逻辑或功能。使用 API Gateway，您可以创建 RESTful API 和 WebSocket API，以便实现实时双向通信应用程序。API Gateway 支持容器化和无服务器工作负载，以及 Web 应用程序。

用户 → ApiGW → Lambda → S3

联网与内容分发 | 计算 | 存储

○ 【安全性身份和合规性】

PART

安全性身份和合规性

- ☆☆☆AWS Artifact
- ☆☆☆AWS Audit Manager
- ☆☆☆Amazon Inspector
- ☆☆☆AWS CloudHSM
- ☆☆☆Amazon Detective
- ☆☆☆Amazon Cognito
- ☆☆☆AWS Directory Service
- ☆☆☆AWS Firewall Manager
- ☆☆☆Amazon GuardDuty
- ☆☆☆AWS Shield
- ☆☆☆AWS IAM Identity Center (AWS Single Sign-On)
- ☆☆☆AWS IAM Identity Center (AWS Single Sign-On)
- ☆☆☆AWS Key Management Service (AWS KMS)
- ☆☆☆AWS Certificate Manager (ACM)
- ☆☆☆Amazon Macie
- ☆☆☆AWS Resource Access Manager (AWS RAM)
- ☆☆☆AWS Secrets Manager
- ☆☆☆AWS Security Hub
- ☆☆☆AWS WAF(Web Application firewall)
- ☆☆☆AWS Identity and Management (IAM)

59▼ ACM (AWS Certificate Manger)

- **购买创建 和 托管管理 HTTPS 证书 (SSL/TLS 证书) 的保管类服务。**

☆☆☆AWS Certificate Manager (ACM)

使用 AWS Certificate Manager (ACM) 通过 AWS 服务和内部连接的资源预置、管理和部署公有和私有 SSL/TLS 证书以供使用。使用 ACM, 您无需再为购买、上传和更新 SSL/TLS 证书而经历耗时的手动流程。

案例展示

Case1: 网站加密
一家电子商务公司需要为其网站启用 HTTPS 加密, 以保护用户的数据安全和隐私。他们使用 AWS Certificate Manager (ACM) 创建了一个 SSL/TLS 数字证书, 并将其绑定到他们的网站上。这样, 他们的网站就能够使用 HTTPS 加密协议进行通信, 确保用户与网站之间的数据传输安全。

Case2: 应用程序负载均衡
一家软件公司正在部署一个基于 AWS 的应用程序, 并使用 AWS 负载均衡器来管理流量。为了保证数据在应用程序和用户之间的安全传输, 他们使用 AWS Certificate Manager (ACM) 创建了一个 SSL/TLS 数字证书, 并将其绑定到他们的负载均衡器上。这样, 他们的应用程序就能够通过 HTTPS 安全地处理用户请求。

Case3: API 网关
一家云服务提供商正在构建一个 API 平台, 用于向客户提供访问其服务的接口。为了确保 API 的安全性, 他们使用 AWS Certificate Manager (ACM) 创建了一个 SSL/TLS 数字证书, 并将其配置到他们的 API 网关上。这样, 客户就可以通过 HTTPS 安全地访问他们的 API, 并确保数据传输的安全性和完整性。

Handwritten notes: HTTP → HTTPS, ACM 证书购买, 证书托管管理, AWS, ELD, EC2, NGINX

* 补充: HTTP 非加密; HTTPS 加密。

60▼ AWS CloudHSM (硬件安全模块)

- HSM——Hardware Security Module——**硬件安全模块**
以高度安全的方式轻松**生成和管理加密密钥**。它旨在**通过确保加密密钥安全存储和管理来保护数据**。
- 案例: 数据加密; 数字签名; 合规性要求 (安全性、隐私性)

☆☆☆AWS CloudHSM

AWS CloudHSM 是一项 AWS 托管的硬件安全模块 (HSM) 服务, 用于在云中提供安全的密钥存储和加密操作, 帮助客户保护敏感数据并满足合规性要求。

借助 AWS CloudHSM, 您可以在经过 FIPS 验证的硬件上管理和访问密钥, 这些硬件由在您自己的虚拟私有云 (VPC) 中运行的客户拥有的单租户 HSM 实例提供保护。

Handwritten note: Hardware Service Manage

61▼ Amazon Detective (安全分析工具)

- 分析、呈现安全数据, 调查潜在安全问题。

☆☆☆Amazon Detective

分析和直观呈现安全数据, 以调查潜在的安全问题。Amazon Detective 使您可以更轻松的分析、调查和快速确定潜在的安全问题可疑活动的根本原因。Amazon Detective 会自动从您的 AWS 资源中收集日志数据并使用机器学习、统计分析和图论来构建一组关联的数据, 使您能够轻松地进行更快、更有效的安全调查。

- 分类安全检测结果: 通过调查 AWS Identity and Access Management (IAM) 角色、用户、IP 地址和 AWS 账户来验证或证伪可疑的检测结果。
- 调查事件: 通过分析相关历史活动的行为模式, 确定恶意活动的程度、其产生的影响和根本原因。
- 深入跟踪威胁: 密切关注特定资源, 例如 Amazon Elastic Compute Cloud (EC2) 实例, 审查相关活动的详细可视化内容。

案例展示

Case1: 异常行为分析
一家企业发现其 AWS 账户的某些资源出现了异常行为, 例如登录失败次数异常增加或者访问了不寻常的 IP 地址。为了及时发现并应对潜在的安全威胁, 他们使用 Amazon Detective 对客户中的日志数据进行分析。通过 Amazon Detective 提供的可视化分析工具, 他们可以迅速识别出异常行为模式, 并采取措施增强其安全性。

Case2: 检测异常事件
一家云服务商提供异常检测和及时发现并调查其客户的 AWS 账户中的异常活动。为了发现这一目标, 他们选择使用 Amazon Detective 来检测安全事件。通过 Amazon Detective 提供的安全分析和功能与其独特的威胁情报, 他们可以快速识别出潜在的异常威胁, 并针对异常行为保护客户的云环境安全。

Case3: 安全审计与合规性
一家金融公司需要定期进行安全审计, 并确保其 AWS 账户的合规性。为了简化安全审计过程, 他们使用 Amazon Detective 对其账户中的安全活动进行审计和分析。通过 Amazon Detective 提供的可视化报告和工具, 他们可以快速识别出潜在的安全风险和合规性问题, 并采取必要的措施加强安全防护和满足合规性要求。

62 ▼ Cognito (账号管理; 身份验证与授权: 注册管理 + 第三方接入)

- 注册管理 + 第三方接入 (如案例③的设备认证)
- 案例: 移动应用身份验证; Web 应用身份验证; IoT 设备认证

案例展示

- Case 1: 移动应用身份验证
一家新兴的移动应用开发公司需要为其应用程序实现用户身份验证和授权功能。他们选择使用 Amazon Cognito 来管理用户身份验证, 并轻松地集成到他们的移动应用中。通过 Amazon Cognito, 用户可以使用各种身份提供者进行登录, 例如用户名/密码、社交媒体登录等。开发人员可以使用 Amazon Cognito 提供的 SDK 快速集成身份验证功能, 并保护他们的移动应用的安全性和用户隐私。
- Case 2: Web 应用程序身份验证
一家电子商务公司需要为其 Web 应用程序实现用户身份验证和授权功能。他们选择使用 Amazon Cognito 来管理用户身份, 并在他们的 Web 应用程序中集成。通过 Amazon Cognito, 用户可以通过用户名/密码登录, 或者使用他们的社交媒体账号进行登录。开发人员可以使用 Amazon Cognito 提供的 JavaScript SDK 快速集成身份验证功能, 并保护他们的 Web 应用程序的安全性和用户隐私。
- Case 3: IoT 设备认证
一家智能家居设备制造商需要为其 IoT 设备实现安全的认证和授权功能。他们选择使用 Amazon Cognito 来管理设备认证, 并为每个设备分配唯一的身份标识。通过 Amazon Cognito, 设备可以使用 AWS 凭证进行认证, 并通过 AWS IAM 角色进行授权访问 AWS 资源。制造商可以使用 Amazon Cognito 提供的 API 快速集成设备认证功能, 并保护他们的 IoT 设备和云端服务的安全性。

* 补充: 【IoT】= 物联网 (如: 智能家居)

63 ▼ GuardDuty (威胁检测: AWS 账户 + 工作负载)

- 威胁检测服务——监控 AWS 账户 + 工作负载 的恶意活动
- 案例: 异常行为检测; 恶意行为检测; 安全合规性

Amazon GuardDuty

Amazon GuardDuty 是一项威胁检测服务, 它持续监控您的 AWS 账户和工作负载的恶意活动, 并提供详细的安全检测结果以实现可见性和修复。

GuardDuty 是一种智能威胁检测服务, 可持续监测您的 AWS 账户、Amazon Elastic Compute Cloud (Amazon EC2) 实例、AWS Lambda 函数、Amazon Elastic Kubernetes Service (Amazon EKS) 集群、Amazon Aurora 登录活动以及存储在 Amazon Simple Storage Service (Amazon S3) 中的数据是否存在恶意活动。如果检测到潜在恶意活动, 例如异常行为、凭证泄露或命令和控制基础设施 (C2) 通信, GuardDuty 将生成详细的安全检测结果, 可用于获得安全可见性并协助进行修复。此外, 使用 Amazon GuardDuty 恶意软件防护功能还有助于检测挂载到 Amazon EC2 实例和容器工作负载的 Amazon Elastic Block Store (Amazon EBS) 卷上的恶意文件。

案例展示

- Case 1: 异常行为检测
一家电子商务公司使用 Amazon GuardDuty 来监控其 AWS 账户中的异常行为, 并及时发现潜在的威胁。GuardDuty 可以自动分析日志数据、网络流量和 AWS CloudTrail 事件, 以识别可能的恶意活动, 如异常的登录尝试、未经授权访问以及恶意 IP 地址的活动。通过及时发现和报告异常行为, 该公司能够加强其安全防护并保护其业务免受威胁。
- Case 2: 恶意行为检测
一家云服务提供商使用 Amazon GuardDuty 来监控其云环境中的恶意行为, 并保护其客户的数据和工作负载。GuardDuty 可以检测到潜在的恶意行为, 如端口扫描、恶意软件活动以及 C&C 通信。通过实时检测和警报, 该服务提供商能够及时采取行动, 阻止潜在的攻击, 并保护其客户的云环境安全。
- Case 3: 安全合规性
一家金融机构需要满足 PCI DSS 等行业的安全合规性要求, 并保护其在 AWS 上的数据 and 应用程序免受安全威胁。他们使用 Amazon GuardDuty 来监控其 AWS 账户中的安全事件, 并生成合规性报告以满足监管要求。GuardDuty 提供的安全事件日志和报告功能帮助他们实时监控追踪安全事件, 并提供必要的审计证据以证明其符合合规性标准。

64 ▼ IAM (权限管控: 分配账号+分配权限) (Identity and Access Management)

- 云上架构设计需要遵循【最小权限原则】——防止账号泄露时, 大权限被盗。
- 案例: 用户身份管理; 服务角色管理; 跨账户访问控制

* 补充: 【服务角色】——云服务之间 调用。

如, 若 lambda 需要访问 DynamDB, 就要给 lambda 分配一个【角色】, 才能对 DynamDB 进行一些操作调用。

案例展示

- Case 1: 用户身份管理
一家软件公司需要在 AWS 上创建多个开发大账户, 并对它们进行不同的权限控制。他们使用 AWS IAM 来管理这些用户的身份和访问权限。通过 AWS IAM, 他们可以创建和管理用户账户, 分配不同的权限策略, 以确保每个开发人员只能访问他们需要的 AWS 资源和服务。
- Case 2: 服务角色管理
一家在电子公司需要将他们的 Web 应用程序连接到 AWS 服务, 以实现自动化任务和资源管理。他们使用 AWS IAM 创建服务角色并为其分配必要的权限, 以允许这些 AWS 服务访问特定的 AWS 资源。通过 AWS IAM, 他们可以为服务角色定义细粒度的权限, 以确保安全地连接他们的应用程序到 AWS 服务。
- Case 3: 跨账户访问控制
一家跨国企业拥有多个 AWS 账户, 并希望实现跨账户的资源共享和访问控制。他们使用 AWS IAM 中的跨账户访问控制功能, 为不同账户中的用户分配访问权限。通过使用跨账户角色和资源共享策略, 他们可以实现跨账户的安全访问控制, 确保资源的安全性和隔离性。

65 ▼ IAM Identity Center (SSO 单点登录 = AWS Single Sign-On) (用于: 内部不同应用+外部访问)

- SSO 单点登录——即，登录一次 就可以 访问所有受密码保护的资源，而无需重复登录。
- 用于——内部不同应用+外部访问

☆☆☆ AWS IAM Identity Center (AWS Single Sign-On)

AWS IAM Identity Center, 又称为 AWS Single Sign-On (SSO), 是一项 AWS 托管的身份验证服务, 用于集中管理多个 AWS 账户和 SaaS 应用程序的用户身份验证和授权。

单点登录 (SSO) 是一种身份验证解决方案, 可以让用户通过一次性用户身份验证登录多个应用程序和网站。基于当今的用户体验需求, 其高访问应用程序, 因此组织正在优先考虑改善安全性和用户体验的访问管理策略。SSO 兼具各方面的优点, 因为一旦验证身份, 用户就可以访问所有受密码保护的资源, 而无需重复登录。

淘家
二级登录

案例展示

Case1: 企业内部身份集中管理
一家大型企业拥有多个 AWS 账户和各种 SaaS 应用程序, 需要统一管理用户的身份验证和授权。他们使用 AWS IAM Identity Center 来集中管理所有用户的身份和访问权限。通过 AWS SSO, 他们可以将所有用户的身份信息集中在单个位置, 并使用统一的登录凭证访问所有的 AWS 账户和 SaaS 应用程序。

Case2: 跨组织合作
一家跨国公司与多个合作伙伴和供应商合作, 需要为合作伙伴和供应商提供安全的访问权限。他们使用 AWS IAM Identity Center 创建外部身份提供商, 并向合作伙伴和供应商提供统一的登录凭证。通过 AWS SSO, 他们可以安全地与外部组织合作, 确保资源的安全性和高可用性。

Case3: 安全审计与合规性
一家金融服务公司需要满足 PCI DSS 等行业的合规性要求, 并确保其 AWS 账户和 SaaS 应用程序的安全性。他们使用 AWS IAM Identity Center 进行安全审计和合规性管理。通过 AWS SSO 提供的审计日志和报告功能, 他们可以跟踪用户的身份和访问活动, 并生成合规性报告以满足监管要求。

66 ▼ Amazon Inspector (安全漏洞扫描: 软件漏洞+网络暴露)

案例展示

Case1: 安全漏洞扫描
一家金融机构需要对其在 AWS 上托管的应用程序进行安全漏洞扫描, 以确保其符合行业标准和最佳实践。他们使用 Amazon Inspector 来扫描其应用程序的漏洞和弱点, 并识别潜在的安全风险。通过 Amazon Inspector 提供的漏洞扫描功能, 他们可以及时发现并解决潜在的安全漏洞, 提高其应用程序的安全性和可靠性。

Case2: 合规性审计
一家医疗保健公司需要定期进行合规性审计, 并确保其 AWS 资源符合 HIPAA 和其他行业法规的要求。他们使用 Amazon Inspector 对其 AWS 资源进行合规性审计, 并生成合规性报告以满足监管要求。通过 Amazon Inspector 提供的合规性审计功能, 他们可以及时发现并解决违规行为, 并确保其符合监管要求。

Case3: 容器安全性
一家互联网技术公司使用容器来托管其微服务应用程序, 并确保容器的安全性。他们使用 Amazon Inspector 对其容器进行安全漏洞扫描, 并识别潜在的安全风险。通过 Amazon Inspector 提供的容器安全性扫描功能, 他们可以及时发现并修复容器中的漏洞和弱点, 提高其容器环境的安全性和稳定性。

☆☆☆ Amazon Inspector

Amazon Inspector 是一项 AWS 托管的安全漏洞扫描服务, 是一项自动化漏洞管理服务, 可持续扫描 AWS 工作负载的软件漏洞和网络暴露, 用于自动检查应用程序和资源的安全性, 发现潜在的安全风险和漏洞, 并提供相关的建议和修复。

Inspector 和 GuardDuty 区别:

- Inspector ——安全漏洞扫描: 软件漏洞+网络暴露
- GuardDuty ——威胁监测: AWS 账户 + 工作负载

KMS 和 HMS 区别:

- KMS ——密钥 加密
- HMS ——硬件 管理

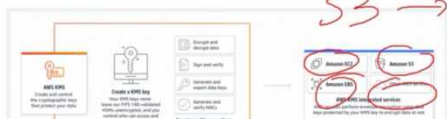
67 ▼ Amazon KMS (加密 服务) (Key Management Service)

- 创建和管理 加密密钥 的托管服务
- 案例: 数据·数据库·日志数据 加密

☆☆☆ AWS Key Management Service (AWS KMS)

创建和控制用于对数据进行加密或数字签名的密钥。

AWS Key Management Service (AWS KMS) 是一项用于创建和管理加密密钥的托管服务, 可帮助用户保护其数据的安全性和机密性。



案例展示

1Case: 数据加密
一家医疗保健公司需要在 AWS 上存储敏感的病人健康数据, 需要对数据进行加密以确保数据机密性。他们使用 AWS KMS 创建加密密钥, 并将其用于加密存储在 Amazon S3 中的病人健康数据。通过 AWS KMS 提供的加密功能, 他们可以确保数据在存储和传输过程中始终受到保护。

2Case: 数据库加密
一家金融服务公司需要在 AWS RDS 中存储客户的金融数据, 并确保对数据进行加密以确保数据的安全性。他们使用 AWS KMS 创建数据库加密密钥, 并将其用于加密存储在 RDS 中的金融数据。通过 AWS KMS 提供的数据库加密功能, 他们可以确保敏感数据并保护客户的金融数据免受未经授权的访问。

3Case: 日志数据加密
一家电子商务公司需要在 AWS CloudWatch 中存储应用程序的日志数据, 并希望为日志数据进行加密以保护客户的隐私信息。他们使用 AWS KMS 创建加密密钥, 并将其用于加密存储在 CloudWatch 中的日志数据。通过 AWS KMS 提供的日志数据加密功能, 他们可以确保日志数据在存储和分析过程中的安全性和机密性。

68 ▼ Macie (发现和保护敏感数据——只发现，不能进行加密，另行用别的加密)

- 保护 数据安全和隐私 服务
- 案例：敏感数据发现；异常行为检测；数据泄露防护 (发现敏感数据，防止明文保存，确保数据不泄露)

案例展示

Case1: 敏感数据发现
一家金融服务公司需要确保其存储在 Amazon S3 中的数据不包含敏感信息，如信用卡号、社会安全号等。他们使用 Amazon Macie 来扫描其 S3 存储桶，并自动发现其中的敏感数据。通过 Macie 提供的敏感数据发现功能，他们可以及时识别和分类敏感数据，并采取必要的措施加强其安全性。

Case2: 异常行为检测
一家电子商务公司需要监控其 S3 存储桶，以及检测异常的数据访问行为，如未经授权访问、异常下载行为等。他们使用 Amazon Macie 来监控其 S3 存储桶，并自动检测异常的数据访问行为。通过 Macie 提供的异常行为检测功能，他们可以及时发现并阻止潜在的严重威胁，保护其数据免受未经授权访问。

Case3: 数据泄露防护
一家医疗保健公司需要保护其存储在 S3 中的医疗记录数据免受数据泄露和不当访问。他们使用 Amazon Macie 来监控其 S3 存储桶，并设置警报规则以检测潜在的数据泄露事件。通过 Macie 提供的数据泄露防护功能，他们可以及时发现并阻止数据泄露事件，并保护医疗记录数据的隐私和安全。

☆☆☆ Amazon Macie

Amazon Macie 是一项数据安全和数据隐私服务，它利用机器学习 (ML) 和模式匹配来发现和保护敏感数据。可帮助客户发现、分类和保护其敏感数据，以及监控其数据仓库的安全性。

69 ▼ WAF (防火墙，阻挡对应用的攻击) Web Application Firewall

* 此 WAF 并非【良好架构框架】!

- 阻挡对应用的攻击：如，SQL 注入、跨站脚本攻击(XSS)、跨站请求伪造(CSRF)
- 常见关键词：防止数据泄露、系统入侵、账户被盗、资金流失→用 WAF 防御

案例展示

Case1: SQL 注入防护
一家电子商务公司的在线商城经常受到 SQL 注入攻击，导致用户数据泄露和系统被入侵。他们使用 AWS WAF 来防御 SQL 注入攻击，通过设置适当的规则和过滤器，他们可以检测和阻止潜在的 SQL 注入攻击，并保护其数据库免受恶意访问。

Case2: 跨站脚本 (XSS) 防护
一家社交媒体平台的用户经常受到跨站脚本 (XSS) 攻击，导致恶意脚本在用户浏览器中执行，窃取用户凭据或执行其他恶意操作。他们使用 AWS WAF 来防御跨站脚本 (XSS) 攻击，通过实施适当的规则和过滤器，他们可以检测和阻止恶意脚本的注入，保护用户免受 XSS 攻击的影响。

Case3: 跨站请求伪造 (CSRF) 防护
一家金融服务公司的在线银行应用经常受到跨站请求伪造 (CSRF) 攻击，导致用户账户被盗和资金流失。他们使用 AWS WAF 来防御跨站请求伪造 (CSRF) 攻击，通过配置适当的规则和过滤器，他们可以检测和阻止恶意的 CSRF 请求，保护用户账户的安全性和资金的完整性。

☆☆☆ AWS WAF (Web Application Firewall)

AWS WAF (Web Application Firewall) 是一项 AWS 托管的网络安全服务，用于保护 Web 应用程序免受常见的 Web 攻击，如 SQL 注入、跨站脚本 (XSS)、跨站请求伪造 (CSRF) 等。

70 ▼ Firewall Manager (WAF 防火墙的管理工具：跨账户集中配置+管理防火墙规则)

- 跨账户集中配置 和 管理防火墙规则
- 案例：VPC 中的安全组集中管理；Web 应用防火墙的管理；合规性审计和报告

案例展示

Case1: 安全组集中管理
一家企业在 AWS 上运行多个 VPC，并且每个 VPC 都有各自的安全组规则。为了确保一致的安全策略和规范，他们使用 AWS Firewall Manager 来集中管理所有 VPC 中的安全组规则。通过 AWS Firewall Manager，他们可以创建、审查和更新安全组规则，并确保所有 VPC 遵循企业的安全标准和政策。

Case2: Web 应用程序防火墙
一家在线零售商在 AWS 上托管其 Web 应用程序，需要保护其应用程序免受常见的 Web 攻击，如 SQL 注入和跨站脚本攻击。他们使用 AWS Firewall Manager 配置 Web 应用程序防火墙，以监控和阻止传入的 HTTP 请求，并阻止恶意流量。通过 AWS Firewall Manager 提供的 Web 应用程序防火墙功能，他们可以及时识别和应对潜在的安全威胁，保护其 Web 应用程序的安全性和可用性。

Case3: 合规性审计和报告
一家金融服务公司需要定期进行安全合规性审计，并生成相应的合规性报告以满足监管要求。他们使用 AWS Firewall Manager 对其 AWS 资源上的防火墙规则进行审计和监控，并生成合规性报告。通过 AWS Firewall Manager 提供的合规性审计和报告功能，他们可以及时发现并解决潜在的安全风险，并满足金融行业的合规性要求。

☆☆☆ AWS Firewall Manager

跨账户集中配置和管理防火墙规则

AWS Firewall Manager 是一项安全管理服务，可让您在 AWS Organizations 中跨账户和应用程序集中配置和管理防火墙规则。在创建新应用程序时，您可以借助 Firewall Manager 实施一套通用的安全规则，更轻松地将新应用程序和资源从一开始就达到合规要求。

71 ▼ Shield (安全防护: 防 DDoS 攻击 + 应用保护 + API 端点防护)

- DDoS(分布式拒绝服务)保护服务, 保护应用免受网络攻击。
 - * **DDoS 攻击**——给容量 100 人的服务器塞 1000 个请求, 拥堵→服务器瘫痪
- Shield 防御**——判断并丢弃异常请求。

● 案例:

*** AWS Shield → 隐微 → 安全防护产品

AWS Shield 是一项 AWS 托管的 DDoS (分布式拒绝服务) 保护服务, 旨在帮助客户保护其应用程序免受网络攻击的影响。

案例展示

- Case1: DDoS 攻击防护
一家电子商务公司的网站经常成为 DDoS 攻击的目标, 导致网站服务不可用和业务中断。他们使用 AWS Shield 来保护其网站免受 DDoS 攻击的影响, 通过 AWS 提供的自动化防护机制, 他们可以快速检测并缓解潜在的 DDoS 攻击, 确保其网站持续可用。
- Case2: 应用程序保护
一家在线游戏公司的游戏服务器经常遭受 DDoS 攻击, 导致游戏体验受到影响, 玩家流失严重。他们使用 AWS Shield 来保护其游戏服务器免受 DDoS 攻击的影响, 通过 AWS 提供的实时监控和自动缓解功能, 他们可以快速应对各种规模和类型的 DDoS 攻击, 保护游戏服务器的稳定运行。
- Case3: API 端点防护
一家金融科技公司的 API 端点经常遭受 DDoS 攻击, 导致客户无法访问其核心业务服务。他们使用 AWS Shield 来保护其 API 端点免受 DDoS 攻击的影响, 通过 AWS 提供的高级防护功能, 他们可以定制防护策略, 并保障核心业务服务的稳定性和可用性。

72 ▼ Secrets Manager (托管密钥: 存储、管理、轮转敏感信息)

- 存储、管理、轮转(更改)敏感信息——如, 数据库密码、API 密钥、OAuth 令牌
- 案例: 数据库凭证管理; API 密钥管理; 加密密钥管理

*** AWS Secrets Manager

AWS Secrets Manager 是一项 AWS 托管的服务, 用于安全地存储、管理和轮转敏感信息, 如数据库密码、API 密钥、OAuth 令牌等。AWS Secrets Manager 助您在整个生命周期内轻松管理、检索和轮转数据库凭证、API 密钥和其他密钥。

案例展示

- Case1: 数据库凭证管理
一家企业需要在其应用程序中安全地管理数据库凭证, 并定期轮转这些凭证以提高安全性。他们使用 AWS Secrets Manager 来存储数据库凭证, 并设置轮转策略。通过 Secrets Manager 提供的轮转功能, 他们可以自动更新数据库凭证, 确保安全性, 并防止未经授权的访问。
- Case2: API 密钥管理
一家软件开发公司需要安全地管理其应用程序所需的 API 密钥, 并限制对这些密钥的访问。他们使用 AWS Secrets Manager 来存储和管理 API 密钥, 并通过 IAM 策略控制对密钥的访问权限。通过 Secrets Manager 提供的访问控制功能, 他们可以确保只有授权的用户才能访问 API 密钥, 保护其应用程序免受未经授权访问。
- Case3: 加密密钥管理
一家金融服务公司需要安全地管理其应用程序所需的加密密钥, 并确保这些密钥的安全性和机密性。他们使用 AWS Secrets Manager 来存储和管理加密密钥, 并将其用于应用程序中的数据加密和解密操作。通过 Secrets Manager 提供的密钥管理功能, 他们可以轻松地管理加密密钥, 并确保其安全性和机密性。

73 ▼ Secrets Hub (托管的集中管理和监控账户安全状态——即时洞察威胁和违规)

■ Hub——集中式管理工具(枢纽)

- 集中管理和监控 AWS 账户安全状态**——提供威胁和违规行为的**即时洞察**

*** AWS Security Hub

AWS Security Hub 是一项 AWS 托管的服务, 用于集中管理和监控 AWS 账户的安全和合规性状态, 并提供有关安全威胁和违规行为的即时洞察。

案例展示

- Case1: 安全合规性扫描
一家企业需要监控其 AWS 账户中的安全合规性状态, 并确保其符合行业标准和最佳实践。他们使用 AWS Security Hub 来收集和分享安全合规性扫描结果, 并评估其合规性。通过 Security Hub 提供的合规性扫描功能, 他们可以及时发现并解决潜在的安全风险和违规行为, 提高其安全合规性水平。
- Case2: 威胁检测与响应
一家云原生技术公司需要及时发现并应对其 AWS 账户中的安全威胁。他们使用 AWS Security Hub 来集中监控其云环境中的安全事件, 并设置警报以检测异常活动。通过 Security Hub 提供的威胁检测与响应功能, 他们可以快速识别并响应潜在的安全威胁, 保护其云资源免受攻击。
- Case3: 安全洞察和建议
一家金融服务公司希望提高其 AWS 账户安全状态的洞察和建议。他们使用 AWS Security Hub 来查看其账户中的安全洞察和最佳实践, 并对采取的措施改进其安全性。通过 Security Hub 提供的安全洞察和建议功能, 他们可以了解其安全状况并采取主动的防御措施来加强其安全防护。

74 ▼ Artifact (提供安全合规性文档、报告)

*** AWS Artifact

AWS Artifact 是您很重要的与合规性相关的信息的最佳中央来源。AWS Artifact 是一项服务, 提供了一系列用于安全合规的文档、报告和资源, 以帮助用户满足其合规性和监管要求。它允许按需提供来自 AWS 和在 AWS Marketplace 上销售产品的 ISV 的安全性和合规性报告。

案例展示

- Case1: 合规性审计
一家金融机构需要进行合规性审计, 以确保其 AWS 云环境符合监管标准和法规要求。他们使用 AWS Artifact 来获取各种合规性文档和报告, 包括 SOC 报告、PCI DSS 报告等。这些报告帮助他们验证他们的 AWS 云环境的合规性, 并提供给审计人员进行审计。
- Case2: 安全审查
一家公司计划对其 AWS 资源进行安全审查, 以确保其 AWS 账户的安全性和完整性。他们使用 AWS Artifact 获取安全控制评估 (SCA) 工作表和其他安全相关文档。这些文档帮助他们评估他们的 AWS 资源的安全性, 识别潜在的安全风险, 并采取相应的措施加强安全防护。
- Case3: 合作伙伴认证
一家软件公司计划将其产品部署到 AWS Marketplace 上, 并需要通过 AWS 合作伙伴认证。为了满足认证要求, 他们需要提供一些合规性文档和报告。他们使用 AWS Artifact 获取所需的文档和报告, 并将其提交给 AWS 进行合作伙伴认证。这些文档和报告帮助他们证明他们的产品符合 AWS 的安全和合规性标准, 从而顺利完成合作伙伴认证流程。

75 ▼ RAM (跨账户资源共享)、AM (审计管理)、DS (企业级目录服务 (用户、账号、权限))

- IAM 管 “云资源权限”； Directory Service 管 “人和账号”
- Active Directory = 企业里 “统一管理账号、电脑和权限” 的系统
 - 打比方来理解：

AD = 公司的人事 + 门禁 + 钥匙管理系统

人事表：谁是员工

门禁：能进哪些门

钥匙：能用哪些系统

☆☆☆ AWS Resource Access Manager (AWS RAM)

AWS RAM 可帮助您的组织或组织单元 (OU) 内的 AWS 账户之间安全地共享资源，还可针对支持的资源类型与 IAM 角色和 IAM 用户共享，并实现跨账户资源的集中管理和控制。

案例展示

Case1: 跨账户资源共享

一家跨国企业拥有多个 AWS 账户，希望在这些账户之间共享特定的资源，如 Amazon S3 存储桶、Amazon RDS 数据库等。他们使用 AWS RAM 创建资源共享组，并将所需的资源添加到共享组中。通过 AWS RAM 提供的资源共享功能，他们可以轻松地不同账户之间共享资源，实现资源的集中管理和协作。

Case2: 组织内部资源管理

一家大型企业拥有多个部门和团队，希望在组织内部共享特定的 AWS 资源，以实现资源的最大化利用和优化。他们使用 AWS RAM 创建资源共享组，并将所需的资源分配给不同的部门和团队。通过 AWS RAM 提供的资源共享功能，他们可以在组织内部方便地共享和管理资源，提高资源利用率和效率。

Case3: 合作伙伴资源访问

一家软件开发公司与多个合作伙伴合作开发应用程序，并希望开发过程中共享特定的 AWS 资源，如 Amazon EC2 实例、Amazon S3 存储桶等。他们使用 AWS RAM 创建资源共享组，并向合作伙伴授予对所需资源的访问权限。通过 AWS RAM 提供的资源共享功能，他们可以安全地与合作伙伴共享资源，实现合作开发项目的顺利进行。

☆☆☆ AWS Directory Service

AWS Directory Service for Microsoft Active Directory 又称为 AWS Managed Microsoft AD，可以激活目录感知型工作负载和 AWS 资源，在 AWS 云中创建和管理 Microsoft Active Directory (AD) 或其他目录服务，帮助客户简化目录集成和管理工作，从而让您可以在 AWS 上使用托管的 AD。

案例展示

Case1: 企业用户身份管理

一家企业需要在 AWS 云中实现与其本地 Active Directory (AD) 的集成, 以便统一管理用户身份和访问控制。他们选择使用 AWS Directory Service 来创建一个托管的 Microsoft AD, 然后将其与其本地 AD 进行连接。这样, 企业可以通过 AWS Directory Service 管理和同步用户身份, 实现跨云和本地环境的统一身份管理。

Case2: Windows 应用程序部署

一家软件开发公司需要在 AWS 云中部署基于 Windows 的应用程序, 并需要一个中心化的用户身份验证解决方案。他们选择使用 AWS Directory Service 创建一个托管的 Microsoft AD, 并将其作为他们应用程序的身份提供者。通过 AWS Directory Service, 用户可以使用他们在 Microsoft AD 中的凭证登录和访问 Windows 应用程序, 从而实现方便的身份验证和访问控制。

Case3: AWS 资源访问控制

一家大型企业需要对其在 AWS 云中的资源实施严格的访问控制和权限管理。他们选择使用 AWS Directory Service 创建一个托管的 Microsoft AD, 并将其与 AWS Identity and Access Management (IAM) 进行集成。通过 AWS Directory Service, 企业可以将其 Microsoft AD 中的用户和组织结构映射到 AWS IAM 中, 从而实现对 AWS 资源的精细化访问控制和权限管理。

☆☆☆ AWS Audit Manager

持续审计您的 AWS 使用情况, 以简化风险与合规性的评估

AWS Audit Manager 是一项 AWS 服务, 可帮助企业自动化合规性审计过程, 管理合规性工作流程, 并生成合规性报告。

案例展示

Case1: 合规性审计

一家金融服务公司需要进行定期的合规性审计, 以确保其 AWS 云环境符合监管标准和内部策略。他们使用 AWS Audit Manager 创建了合规性框架, 并配置了一系列合规性规则和要求。AWS Audit Manager 自动化执行审计程序, 收集审计结果, 并生成合规性报告, 帮助他们快速识别和解决潜在的合规性问题。

Case2: 安全性审计

一家电子商务公司需要对其 AWS 资源进行安全性审计, 以确保其云环境的安全性和完整性。他们使用 AWS Audit Manager 创建了安全性审计框架, 并配置了一系列安全性规则和要求。AWS Audit Manager 自动化执行审计程序, 收集安全性评估数据, 并生成安全性审计报告, 帮助他们评估其云环境的安全性, 并采取必要的措施加强安全防护。

Case3: 合作伙伴合规性

一家软件公司计划将其产品部署到 AWS Marketplace 上, 并需要通过 AWS 合作伙伴认证。为了满足认证要求, 他们需要进行合规性审计, 并生成合规性报告。他们使用 AWS Audit Manager 创建了合规性框架, 并根据 AWS 的合规性标准配置了一系列合规性规则。AWS Audit Manager 自动化执行审计程序, 收集合规性数据, 并生成合规性审计报告, 帮助他们证明其产品符合 AWS 的安全和合规性要求, 顺利完成合作伙伴认证流程。

