AWS 白皮书

标记 AWS 资源的最佳实践

标记 AWS 资源的最佳实践: AWS 白皮书

Table of Contents

标记 AWS 资源的最佳实践

发布日期：2023 年 3 月 30 日（文档修订）

Amazon Web Services (AWS) 允许您以标签的形式为许多 AWS 资源分配元数据。每个标签都是由一个密钥和一个可选值组成的简单标签，用于存储有关资源的信息或保留在该资源上的数据。本白皮书重点介绍标记用例、策略、技术和工具，有助于您按照目的、团队、环境或与业务相关的其他标准对资源进行分类。实施一致的标记策略可以更轻松地筛选和搜索资源，监控成本和使用情况，以及管理 AWS 环境。

本白皮书以使用多个账户组织您的 AWS 环境白皮书中提供的实践和指导为基础。建议您在阅读本白皮书之前先阅读该白皮书。AWS 建议您以全面的方式建立云基础。有关更多信息，请参阅 AWS 上的建立您的云基础。

您的架构是否良好？

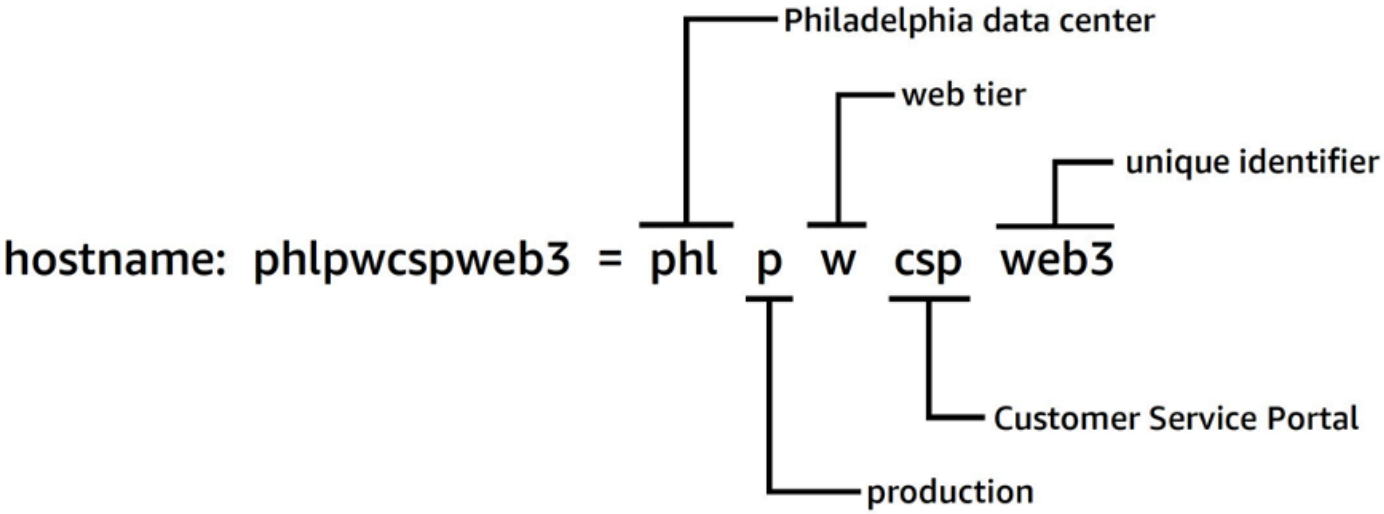
当您在云端构建系统时，AWS Well-Architected Framework 可助您了解所作决策的利弊。利用此框架的六个支柱，您可以了解到设计和运行可靠、安全、高效、经济有效且可持续的系统的架构最佳实践。您可以使用 AWS Management Console 免费提供的 AWS Well-Architected Tool，回答与每个支柱相关的一组问题，即可根据这些最佳实践检查自己的工作负载。

有关云架构的更多专家指导和最佳实践（参考架构部署、图表和白皮书），请参阅 AWS 架构中心。

简介

通过创建 Amazon EC2 实例、Amazon EBS 卷、安全组和 AWS Lambda 功能等资源，AWS 可让您在 AWS 中轻松部署工作负载。您还可以扩展和增加 AWS 资源的实例集，以托管您的应用程序、存储您的数据，并随着时间的推移扩展您的 AWS 基础设施。随着 AWS 的使用量增加到跨越多个应用程序的多种资源类型，您将需要一种机制来跟踪哪些资源分配给了哪些应用程序。利用这一机制支持您的运营活动，例如成本监控、事件管理、补丁、备份和访问控制。

在本地环境中，这些知识通常保存在知识管理系统、文档管理系统和内部维基页面中。使用配置管理数据库 (CMDB)，您可以使用标准变更控制流程存储和管理相关的详细元数据。这种方法可提供治理，但需要额外的开发和维护工作。您可以采用结构化方法来命名资源，但资源名称只能包含有限的信息。



结构化的资源命名方法

例如，EC2 实例有一个名为“名称”的预定义标签，可提供类似的功能，并允许您在将工作负载移至 AWS 时为其命名。

2010 年，AWS 推出了资源标签，为资源附加元数据提供了一种灵活、可扩展的机制。本白皮书将指导您在 AWS 环境中制定和实施稳健的标记策略。本指南将帮助您确保标记的一致性和覆盖范围，从而支持您的决策和运营活动

标签是什么？

标签是应用于资源的密钥值对，用于保存有关该资源的元数据。每个标签都是由一个密钥和一个可选值组成的。目前并非所有服务和资源类型都支持标签（请参阅支持资源组标记 API 的服务）。其他服务可通过自己的 API 支持标签。需要注意的是，标签未加密，不应用于存储敏感数据，如个人身份信息 (PII)。

用户使用 AWS CLI、API 或 AWS Management Console 创建并应用于 AWS 资源的标签称为用户定义的标签。一些 AWS 服务（例如 AWS CloudFormation、Elastic Beanstalk 和 Auto Scaling）会自动为其创建和管理的资源分配标签。这些密钥被称为 AWS 生成的标签，通常带有 aws 前缀。该前缀不能用于用户定义的标签密钥。

在 AWS 资源中添加用户自定义标签的数量有使用要求和数量限制。有关更多信息，请参阅《AWS 一般参考指南》中的标签命名限制和要求。AWS 生成的标签不计入这些用户定义标签的限制。

表 1 - 用户定义的标签密钥和值示例

实例 ID	标签密钥	标签值
i-01234567abcdef89a	CostCenter	98765
	Stack	Test
i-12345678abcdef90b	CostCenter	98765
	Stack	Production

表 2 - AWS 生成的标签示例

AWS 生成的标签密钥	理由
aws:ec2spot:fleet-request-id	标识启动实例的 Amazon EC2 竞价型实例请求。
aws:cloudformation:stack-name	标识创建资源的 AWS CloudFormation 堆栈
lambda-console:blueprint	标识用作 AWS Lambda 功能模板的蓝图

AWS 生成的标签密钥	理由
elasticbeanstalk:environment-name	标识创建资源的应用程序。
aws:servicecatalog:provisionedProductArn	供应产品的的 Amazon 资源名称 (ARN)
aws:servicecatalog:productArn	从中启动预配置产品的产品的 ARN。

AWS 生成的标签构成命名空间。例如，在 AWS CloudFormation 模板中，您可以定义一组要在 stack 中一起部署的资源，其中 stack-name 是您指定的用于标识它的描述性名称。如果您检查诸如 aws:cloudformation:stack-name 的密钥，则可以发现用于限定参数范围的命名空间使用了三个元素：aws 表示组织，cloudformation 表示服务，而 stack-name 表示参数。

用户定义的标签也可以使用命名空间，建议使用组织标识符作为前缀。这可帮助您快速识别标签是您托管模式中的内容，还是您在环境中使用的服务或工具所定义的内容。

在[在 AWS 上建立云基础]白皮书中，我们推荐了一组应实施的标签。不同的企业很可能有不同的允许模式和特定标签的不同列表。查看表 3 中的示例：

表 3 - 相同的标签密钥，不同的值验证规则

组织	标签密钥	标签值验证	标签值示例
A 公司	CostCenter	5432, 5422, 5499	5432
B 公司	CostCenter	ABC*	ABC123

如果这两个架构位于不同的组织中，就不会出现标签冲突问题。但是，如果这两个环境合并，命名空间就会发生冲突，验证就会变得更加复杂。这种情况似乎不太可能发生，但企业会被收购或合并，还有其他一些情况，例如客户与托管服务提供商、游戏出版商或风险投资企业合作，来自不同组织的账户是共享 AWS 组织的一部分。如表 4 所示，使用企业名称作为前缀来定义唯一的命名空间，就可以避免这些挑战：

表 4 - 标签密钥中命名空间的使用

组织	标签密钥	标签值验证	标签值示例
A 公司	company-a :CostCenter	5432, 5422, 5499	5432
B 公司	company-b :CostCenter	ABC*	ABC123

在定期收购和剥离业务的大型复杂组织中，这种情况会更频繁地发生。随着新收购公司的流程和做法在更大范围内得到统一，问题也就迎刃而解了。使用不同的命名空间很有帮助，因为可以报告旧标签的使用情况，并联系相关团队采用新架构。命名空间还可用于表示一个范围，或代表一个用例或一个与组织所有者相一致的责任领域。

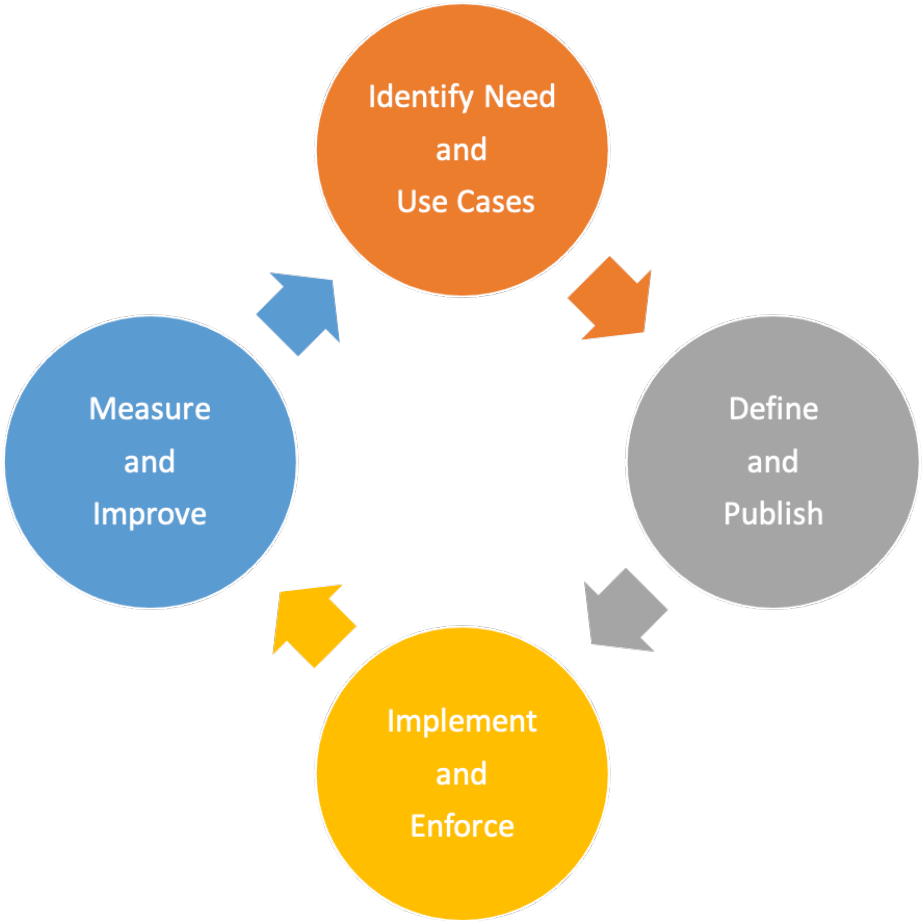
表 5 - 标签密钥中的作用域或用例范围示例

用例	标签密钥	理由	允许的值
数据分类	example-inc:info-sec:data-classification	信息安全定义的数据分类集	sensitive ，company-confidential ，customer-identifiable
操作	example-inc:dev-ops:environment	实施测试和开发环境的调度	development ，staging, quality-assurance ，production
灾难恢复	example-inc:disaster-recovery:rpo	定义资源的恢复点目标 (RPO)	6h, 24h
成本分配	example-inc:cost-allocation:business-unit	财务团队需要关于各团队使用和支出情况的成本报告	corporate ，recruitment ，support, engineering

标签简单易用。标签的密钥和值都是长度可变的字符串，可以支持多种字符集。有关长度和字符集的更多信息，请参阅《AWS 一般参考》中的标记 AWS 资源。标签区分大小写，这意味着 costCenter 和 costcenter 是不同的标签密钥。在不同的国家/地区，单词的拼写可能会有所不同，这可能会影响您的密钥。例如，在美国，人们可能会将密钥定义为 costcenter，但在英国，人们可能更倾向于使用 costcentre。从资源标记的角度来看，这些是不同的密钥。定义拼写、大小写和标点符号作为标记策略的一部分。任何人在创建或管理资源时都可以参考这些定义。下一节 制定您的标记策略 将详细讨论这一主题。

制定您的标记策略

与许多运营实践一样，实施标记策略也是一个不断迭代和改进的过程。先从当务之急小处着手，然后根据需要发展标记方案。



标记策略迭代和改进周期

在整个过程中，所有权是问责制和取得进展的关键。由于标签可以用于不同用途，因此可以将总体标记策略划分为组织内部的责任领域。通过标记，可以对依赖于资源特征的活动采取程序化方法。可以从标记中受益的利益相关者的范围取决于组织的规模和运营方式。明确界定参与构建和实施标记战略的团队的职责，可使规模较大的组织从中受益。一些利益相关者可以负责确定标记需求（定义用例）；另一些利益相关者可以负责维护、实施和改进标记策略。

通过分配所有权，您可以很好地实施策略的各个方面。在适当情况下，可以将这种所有权正式确定为策略，并记录在责任矩阵（例如，RACI：负责、问责、咨询和知情）或责任共担模式中。在规模较小的组织中，团队可能在标记策略中扮演多重角色，从需求定义到实施和执行。

定义需求和用例

开始构建您的策略时，首先应与具有使用元数据基本需求的利益相关者进行互动。这些团队定义了资源需要标记的元数据，以支持其活动，例如报告、自动化和数据分类。他们概述了需要如何组织资源以及这些资源需要与哪些策略相对应。这些利益相关者在组织中的角色和职能示例包括：

- 财务部门和业务部门需要了解投资的价值，将其与成本挂钩，以便在解决低效率问题时优先考虑需要采取的行动。了解成本与所产生价值的对比有助于识别不成功的业务部门或产品供应。这样可以就继续提供支持、采用替代方案（例如，使用 SaaS 产品或托管服务）或淘汰没有利润的业务产品作出明智的决策。
- 治理和合规部门需要了解数据的分类（例如，公开、敏感或机密）、特定工作负载是否在特定标准或法规的审计范围内，以及服务的关键性（服务或应用程序是否对业务至关重要），以便应用适当的控制和监督，例如权限、策略和监控。
- 运营和开发部门需要了解工作负载生命周期、所支持产品的实施阶段、发布阶段的管理（例如，开发、测试、生产拆分）及其相关的支持优先级和利益相关者管理要求。还需要定义和理解备份、补丁、可观察性和报废等职责。
- 信息安全 (InfoSec) 和安全运营 (SecOps) 概述了必须应用哪些控制措施以及推荐使用哪些控制措施。InfoSec 通常定义控制措施的实施，而 SecOps 通常负责管理这些控制措施。

根据您的用例、优先级、组织规模和运营实践，您可能需要组织内不同团队的代表，例如财务（包括采购）、信息安全、云支持和云运营。您还需要应用程序和流程所有者的代表，以实现补丁、备份和恢复、监控、任务调度和灾难恢复等功能。这些代表有助于推动标记策略的定义、实施和效果评估。他们应从利益相关者及其用例出发逆向工作，并且开展跨职能研讨会。在研讨会上，他们有机会分享自己的观点和需求，并帮助制定整体战略。本白皮书稍后将举例说明各种用例中的参与者及其参与情况。

利益相关者还可以定义和验证强制性标签的密钥，并就可选标签的范围提出建议。例如，财务团队可能需要将资源与内部成本中心、业务部门或两者相关联。因此，他们可能会要求将某些标签密钥（例如 `CostCenter` 和 `BusinessUnit`）设为强制性。个别开发团队可能会决定使用其他标签来实现自动化，例如 `EnvironmentName`、`OptIn` 或 `OptOut`。

主要利益相关者需要就标记策略方法达成共识，并记录合规性和治理相关问题的答案，例如：

- 需要解决哪些用例？
- 谁负责标记资源（实施）？
- 如何执行标签，将使用哪些方法和自动化（主动或被动）？
- 如何衡量标记的有效性和目标？

- 应多久审查一次标记策略？
- 谁来推动改进？ 如何做到这一点？

然后，云赋能、云业务办公室和云平台工程等业务职能部门可以在制定标记策略的过程中发挥促进者的作用，通过衡量进度、消除障碍和减少重复工作等方面帮助推动标记策略的采用并确保应用的一致性。

定义和发布标记方案

在标记 AWS 资源时，无论是强制性标签还是可选标签，都要采用一致的方法。全面的标记方案有助于实现这种一致性。以下示例可以帮助您开始：

- 商定强制性标签密钥
- 定义可接受的值和标签命名规则（大写或小写、破折号或下划线、层次结构等）
- 确认值不构成个人身份信息 (PII)
- 决定谁可以定义和创建新标签密钥
- 商定如何添加新的强制性标签值和如何管理可选标签

请查看下面的标记类别表，它可以作为您的标记方案中可能包含的内容的基准。您仍然需要确定标签密钥使用的约定，以及每个标签密钥允许的值。标记方案是您在其中为环境定义标记架构的文档。

表 6 - 明确的标记方案示例（第 1 部分）

用例	标签密钥	理由	允许的值（列出的值或值前缀/后缀）	用于成本分配	资源类型	范围	必需
成本分配	example-inc:cost-allocation: ApplicationId	跟踪各业务部门产生的成本与价值对比情况	DataLakeX，RetailSiteX	是	全部	全部	强制性
成本分配	example-inc:cost-allocation: BusinessUnitId	按业务部门监控成本	Architecture，DevOps, Finance	是	全部	全部	强制性
成本分配	example-inc:cost-allocation:CostCenter	按成本中心监控成本	123-*	是	全部	全部	强制性
成本分配	example-inc:cost-allocation:Owner	哪位预算负责人负责这项工作负载	Marketing，RetailSupport	是	全部	全部	强制性
访问控制	example-inc:access-control:LayerId	根据角色确定子组件/层以授予对资源访问权限	DB_Layer, Web_Layer，App_Layer	否	全部	全部	可选
自动化	example-i	实施测试和开发	Prod, Dev,	否	EC2、RDS	全部	强制性

表 6 - 明确的标记方案示例（第 2 部分）

用例	标签密钥	理由	允许的值（列出的值或值前缀/后缀）	用于成本分配	资源类型	范围	必需
DevOps	example-inc:operations: Owner	哪个团队/小组负责创建和维护资源	Squad01	否	全部	全部	强制性
灾难恢复	example-inc:disaster-recovery:rpo	定义资源的恢复点目标 (RPO)	6h, 24h	否	S3、EBS	Prod	强制性
数据分类	example-inc:data:classification	对数据进行分类以实现合规性和治理	Public, Private, Confidential , Restricted	否	S3、EBS	全部	强制性
合规性	example-inc:compliance:framework	确定工作负载所遵循的合规性框架	PCI-DSS, HIPAA	否	全部	Prod	强制性

定义标记方案后，应在版本控制的存储库管理方案，所有相关利益相关者都可以访问该存储库，以便于参考和跟踪更新。这种方法提高了效率，实现了灵活性。

AWS Organizations–标签策略

使用 AWS Organizations 中的策略，您能够将其他类型的管理应用于组织中的 AWS 账户。标签策略是以 JSON 格式表达标记方案的方式，这样平台就能在您的 AWS 环境中报告并有选择地执行该方案。标签策略定义了特定资源类型的标签密钥可接受的值。该策略可以采用值列表的形式，也可以采用前缀后加通配符 (*) 的形式。简单前缀法不如离散值列表严格，但维护要求较低。

以下示例说明了如何定义标记策略来验证给定密钥可接受的值。根据架构的人性化表格定义，您可以将这些信息转录到一个或多个标签策略中。可以使用单独的策略来支持授权所有权，或者某些策略可能仅适用于特定场景。

ExampleInc-CostAllocation.json

以下是报告成本分配标签的标签策略示例：

```
{
  "tags": {
    "example-inc:cost-allocation:ApplicationId": {
      "tag_key": {
        "@@assign": "example-inc:cost-allocation:ApplicationId"
      },
      "tag_value": {
        "@@assign": [
          "DataLakeX",
          "RetailSiteX"
        ]
      }
    },
    "example-inc:cost-allocation:BusinessUnitId": {
      "tag_key": {
        "@@assign": "example-inc:cost-allocation:BusinessUnitId"
      },
      "tag_value": {
        "@@assign": [
          "Architecture",
          "DevOps",
          "FinanceDataLakeX"
        ]
```

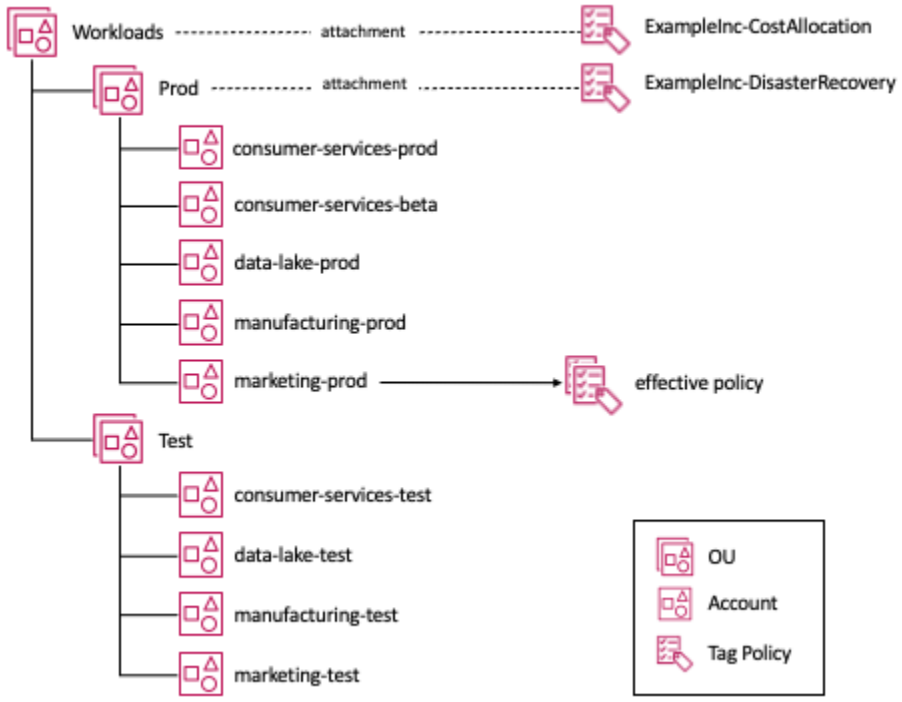
```
      }
    },
    "example-inc:cost-allocation:CostCenter": {
      "tag_key": {
        "@@assign": "example-inc:cost-allocation:CostCenter"
      },
      "tag_value": {
        "@@assign": [
          "123-*"
        ]
      }
    }
  }
}
```

ExampleInc-DisasterRecovery.json

以下是报告灾难恢复标签的标签策略示例：

```
{
    "tags": {
        "example-inc:disaster-recovery:rpo": {
            "tag_key": {
                "@@assign": "example-inc:disaster-recovery:rpo"
            },
            "tag_value": {
                "@@assign": [
                    "6h",
                    "24h"
                ]
            }
        }
    }
}
```

在该示例中，ExampleInc-CostAllocation 标签策略附加到 Workloads OU，因此适用于 Prod 和 Test 子 OU 中的所有账户。同样，ExampleInc-DisasterRecovery 标签策略附加到 Prod OU，因此仅适用于该 OU 以下的账户。使用多个账户组织环境白皮书更详细地探讨了推荐的 OU 结构。



将标签策略附加到 OU 结构

查看图中的 `marketing-prod` 账户，两个标签策略都适用于该账户，因此我们就有了有效策略的概念，即适用于账户的特定类型的策略的卷积。如果您主要是手动管理资源，则可以通过访问控制台中的资源组和标签编辑器：标签策略来查看有效策略。如果您使用基础设施即代码 (IaC) 或脚本来管理资源，则可以使用 `AWS::Organizations::DescribeEffectivePolicy` API 调用。

实施和执行标记

在该部分中，我们将向您介绍可用于以下资源管理策略的工具：手动、基础设施即代码 (IaC) 和持续集成/持续交付 (CI/CD)。这些方法的关键在于部署的频率越来越高。

手动托管的资源

这些通常是属于采用基础或迁移阶段的工作负载。通常情况下，这些都是使用传统书面程序构建的基本静态的简单工作负载，或者是使用 CloudEndure 等工具从本地环境按原样迁移的工作负载。迁移工具（例如 Migration Hub 和 CloudEndure）可以在迁移过程中应用标签。但是，如果在最初的迁移过程中没有应用标签，或者标签方案在迁移后发生了变化，则可以使用标签编辑器（AWS Management Console 的特征）使用各种搜索条件搜索资源，并批量添加、修改或删除标签。搜索条件可以包括带有或不带有特定标签或值的资源。AWS 资源标记 API 允许您以编程方式执行这些功能。

随着这些工作负载的现代化，引入了诸如自动扩缩组等资源类型。这些资源类型具有更大的弹性和更强的故障恢复能力。自动扩缩组代表您管理 Amazon EC2 实例，但您可能仍希望 EC2 实例与手动创建的资源保持一致。Amazon EC2 启动模板提供了指定自动扩缩应应用于其创建的实例的标签的方法。

当手动管理工作负载的资源时，自动标记资源会很有帮助。目前有多种解决方案。一种方法是使用 AWS Config 规则，它可以检查 `required_tags`，然后启动一个 Lambda 函数来应用它们。本白皮书稍后将详细介绍 AWS Config 规则。

基础设施即代码 (IaC) 托管资源

AWS CloudFormation 提供了一种通用语言，用于在您的 AWS 环境中配置所有基础设施资源。CloudFormation 模板是以自动方式创建 AWS 资源的 JSON 或 YAML 文件。使用 CloudFormation 模板创建 AWS 资源时，您可以使用 CloudFormation 资源标签属性在创建时将标签应用到支持的资源类型。使用 IaC 管理标签和资源有助于确保一致性。

当 AWS CloudFormation 创建资源时，该服务会自动将一组 AWS 定义的标签应用于 AWS CloudFormation 模板创建的资源。这些是：

```
aws:cloudformation:stack-name
aws:cloudformation:stack-id
aws:cloudformation:logical-id
```

您可以根据 AWS CloudFormation 堆栈轻松定义资源组。该堆栈创建的资源将继承这些 AWS 定义的标签。但是，对于自动扩缩组中的 Amazon EC2 实例，需要在 AWS CloudFormation 模板中的自动扩缩组定义中设置 `AWS::AutoScaling::AutoScalingGroup TagProperty`。另外，如果您使用的是带有自动扩缩组的 EC2 启动模板，则可以在其定义中定义标签。建议将带有自动扩缩组或 AWS 容器服务的 EC2 启动模板一起使用。这些服务有助于确保对 Amazon EC2 实例进行一致的标记，还支持跨多种实例类型和购买选项的自动扩缩，从而提高故障恢复能力并优化计算成本。

AWS CloudFormation 挂钩为开发人员提供了一种使应用程序的关键方面与其组织标准保持一致的方法。挂钩可配置为提供警告或阻止部署。该特征最适合检查模板中的关键配置元素，例如是否将自动扩缩组组配置为将要启动的所有 Amazon EC2 实例应用客户定义的标签，或者确保使用所需的加密设置创建所有 Amazon S3 存储桶。在这两种情况下，对合规性的评估都是在部署前通过 AWS CloudFormation 挂钩推送到部署流程的早期阶段。

AWS CloudFormation 提供了检测功能，当从模板配置的资源（请参阅支持偏差检测的资源）被修改，且资源不再符合其预期的模板配置时，可进行检测。这就是所谓的偏差。如果您使用自动化技术将标签应用到通过 IaC 托管的资源上，则您就是在修改这些标签，从而引入偏差。在使用 IaC 时，目前建议

将任何标记要求作为 IaC 模板的一部分进行托管，实施 AWS CloudFormation 挂钩，并发布可由自动化使用的 AWS CloudFormation Guard 规则集。

CI/CD 管道托管资源

随着工作负载成熟度的提高，很可能会采用持续集成和持续部署 (CI/CD) 等技术。这些技术通过提高测试的自动化程度，可以更轻松地更频繁地部署小更改，从而有助于降低部署风险。可观察性策略如果能检测到部署带来的意外行为，就能在对用户影响最小的情况下自动回滚部署。当您每天部署数十次时，追溯性地应用标签就不再实用了。一切都需要以代码或配置的形式表达出来，进行版本控制，并尽可能在部署到生产环境之前进行测试和评估。在开发和运营 (DevOps) 组合模式中，许多实践都将操作注意事项当作代码来处理，并在部署生命周期的早期对其进行验证。

理想情况下，您希望尽可能早地在流程中推送这些检查（如 AWS CloudFormation 挂钩所示），这样您就可以在 AWS CloudFormation 模板离开开发人员的计算机之前确信它们符合您的策略。

AWS CloudFormation Guard 2.0 提供了为您可以使用 CloudFormation 定义的任何内容编写预防性合规性规则的方法。该模板已根据开发环境中的规则进行验证。显然，这一特征有多种应用，但在本白皮书中，我们将仅举例说明如何确保始终使用 AWS::AutoScaling::AutoScalingGroup TagProperty。

以下是 CloudFormation Guard 规则的示例：

```
let all_asgs = Resources.*[ Type == 'AWS::AutoScaling::AutoScalingGroup' ]

rule tags_asg_automation_EnvironmentId when %all_asgs !empty {
    let required_tags = %all_asgs.Properties.Tags.*[
        Key == 'example-inc:automation:EnvironmentId' ]
    %required_tags[*] {
        PropagateAtLaunch == 'true'
        Value IN ['Prod', 'Dev', 'Test', 'Sandbox']
        <<Tag must have a permitted value
          Tag must have PropagateAtLaunch set to 'true'>>
    }
}

rule tags_asg_costAllocation_CostCenter when %all_asgs !empty {
    let required_tags = %all_asgs.Properties.Tags.*[
        Key == 'example-inc:cost-allocation:CostCenter' ]
    %required_tags[*] {
        PropagateAtLaunch == 'true'
        Value == /^123-/
        <<Tag must have a permitted value
```

```
            Tag must have PropagateAtLaunch set to 'true'>>
    }
}
```

在代码示例中，我们筛选了模板中所有 AutoScalingGroup 类型的资源，然后制定了两条规则：

- **tags_asg_automation_EnvironmentId** - 检查是否存在具有此密钥的标签，其值是否在允许的值列表内，以及 PropagateAtLaunch 是否设置为 true
- **tags_asg_costAllocation_CostCenter** - 检查是否存在具有此密钥的标签，其值是否以定义的前缀值开头，以及 PropagateAtLaunch 是否设置为 true

执行

如前所述，资源组和标签编辑器提供了一种方法来确定哪些资源不符合应用于组织 OU 的标签策略中定义的标记要求。从 Organization 成员账户访问资源组和标签编辑器控制台工具，会显示适用于该账户的策略，以及该账户中不符合标签策略要求的资源。如果从管理账户访问（且标签策略已在 AWS Organizations 下的服务中启用访问），则可以查看组织内所有关联账户的标签策略合规性。

在标签策略本身中，您可以启用对特定资源类型的执行功能。在以下策略示例中，我们添加了执行功能，要求所有 ec2:instance 和 ec2:volume 类型的资源都必须符合该策略。有一些已知的限制，例如资源上必须有标签才能由标签策略对其进行评估。有关列表，请参阅支持使用标签策略执行的资源。

ExampleInc-Cost-Allocation.json

以下是报告和/或执行成本分配标签的标签策略示例：

```
{
  "tags": {
    "example-inc:cost-allocation:ApplicationId": {
      "tag_key": {
        "@@assign": "example-inc:cost-allocation:ApplicationId"
      },
      "tag_value": {
        "@@assign": [
          "DataLakeX",
          "RetailSiteX"
        ]
      },
      "enforced_for": {
        "@@assign": [
```

```
          "ec2:instance",
          "ec2:volume"
        ]
      }
    },
    "example-inc:cost-allocation:BusinessUnitId": {
      "tag_key": {
        "@@assign": "example-inc:cost-allocation:BusinessUnitId"
      },
      "tag_value": {
        "@@assign": [
          "Architecture",
          "DevOps",
          "FinanceDataLakeX"
        ]
      },
      "enforced_for": {
        "@@assign": [
          "ec2:instance",
          "ec2:volume"
        ]
      }
    },
    "example-inc:cost-allocation:CostCenter": {
      "tag_key": {
        "@@assign": "example-inc:cost-allocation:CostCenter"
      },
      "tag_value": {
        "@@assign": [
          "123-*"
        ]
      },
      "enforced_for": {
        "@@assign": [
          "ec2:instance",
          "ec2:volume"
        ]
      }
    }
  }
}
```

AWS Config (**required_tag**)

AWS Config 是一项允许您评估、审计和评价资源配置的服务（请参阅 AWS Config 支持的资源类型）。在标记方面，我们可以使用 required_tags 规则来识别哪些资源缺少具有特定密钥的标签（请参阅 required_tags 支持的资源类型）。根据前面的示例，我们可以测试所有 Amazon EC2 实例是否存在该密钥。如果密钥不存在，则该实例将被注册为不合规。该 AWS CloudFormation 模板描述了一个 AWS Config 规则，用于在 Amazon S3 存储桶、Amazon EC2 实例和 Amazon EBS 卷上测试表中描述的强制性标签密钥是否存在。

```
Resources:
 MandatoryTags:
   Type: AWS::Config::ConfigRule
   Properties:
     ConfigRuleName: ExampleIncMandatoryTags
     Description: These tags should be in place
     InputParameters:
       tag1Key: example-inc:cost-allocation:ApplicationId
       tag2Key: example-inc:cost-allocation:BusinessUnitId
       tag3Key: example-inc:cost-allocation:CostCenter
       tag4Key: example-inc:automation:EnvironmentId
     Scope:
       ComplianceResourceTypes:
       - "AWS::S3::Bucket"
       - "AWS::EC2::Instance"
       - "AWS::EC2::Volume"
     Source:
       Owner: AWS
       SourceIdentifier: REQUIRED_TAGS
```

对于手动托管资源的环境，可以增强 AWS Config 规则，通过 AWS Lambda 函数自动修复，自动将缺少的标签密钥添加到资源中。虽然这适用于静态工作负载，但当您开始通过 IaC 和部署管道托管资源时，其效率会逐渐降低。

AWS Organizations - 服务控制策略 (SCP) 是一种组织策略，可用于管理组织中的权限。SCP 为您组织或组织单位 (OU) 中的所有账户提供对最大可用权限的集中控制。SCP 只影响用户和角色，此类用户和角色由属于组织的账户进行管理。虽然它们不会直接影响资源，但会限制用户和角色的权限，包括标记操作的权限。在标记方面，除了标签策略所能提供的功能外，SCP 还能为标签执行提供额外的精度。

在以下示例中，该策略将拒绝不存在 example-inc:cost-allocation:CostCenter 标签的 ec2:RunInstances 请求。

以下是拒绝 SCP：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyRunInstanceWithNoCostCenterTag",
      "Effect": "Deny",
      "Action": "ec2:RunInstances",
      "Resource": [
        "arn:aws:ec2:*:*:instance/"
      ],
      "Condition": {
        "Null": {
          "aws:RequestTag/example-inc:cost-allocation:CostCenter": "true"
        }
      }
    }
  ]
}
```

在设计上不可能检索到适用于相关账户的有效服务控制策略。在使用 SCP 执行标记时，需要向开发人员提供文档，以便他们确保其资源符合应用于其账户的策略。在开发人员的账户中提供对 CloudTrail 事件的只读访问权限，可以帮助他们在资源不合规时进行调试。

衡量标记有效性并推动改进

实施标记策略后，必须根据目标用例来衡量其有效性。有效性的衡量标准将因用例而异。例如：

- 成本归因 - 您可以使用 AWS Cost Explorer 或 AWS 成本和使用情况报告等工具，根据支出来衡量资源的标记覆盖率。例如，您可以跟踪产生费用的已标记或未标记资源的百分比，特别是监控特定的标签密钥。
- 自动化 - 您可能需要审计是否达到了预期的效果。例如，非生产 Amazon EC2 实例是否在工作时间以外暂停，审计实例的启动和停止时间。

管理账户中的资源组和标签编辑器提供了额外的功能，可分析组织中所有相关账户的标签策略合规性。

根据对标记有效性的测量结果，确定是否需要改进或更改任何步骤，例如用例定义、标记方案的实施或执行。进行必要的更改并重复该周期，直到达到预期效果。在成本归因示例中，您可以查看改进的百分比。

由于需要对资源进行实际标记的是开发人员和操作人员，因此让他们掌握主动权至关重要。这并不是团队在采用 AWS 的过程中通常要代入的唯一额外责任。对开发和运行其应用程序的安全性和成本承担更多责任也很重要。Organizations 经常使用目标和指标作为激励采用新实践的手段，因此这也适用于此处。

标记用例

主题

成本分配和财务管理的标签

组织通常首先要解决的一个标记用例就是成本和使用情况的可见性和管理。这通常有几个原因：

- 这通常是一个很好理解的场景，需求也是众所周知的。例如，财务团队希望查看跨多个服务、特征、账户或团队的工作负载和基础设施的总成本。实现成本可见性的方法之一是对资源进行一致的标记。
- 明确定义标签及其值。通常，组织的财务系统中已经存在成本分配机制，例如，按成本中心、业务部门、团队或组织职能进行跟踪。
- 快速、可观的投资回报。当资源标记一致时，就可以跟踪一段时间内的成本优化趋势，例如，对资源进行调整、自动扩展或列入计划。

了解您在 AWS 中如何产生成本，可以让您做出明智的财务决策。了解您在资源、工作负载、团队或组织层面上产生的成本，就能更好地理解与所取得的业务成果相比，在适用层面上所实现的价值。

工程团队可能没有资源财务管理的经验。聘请一名具有 AWS 财务管理专业技能的人员，可以对工程和开发团队进行 AWS 财务管理基础知识的培训，并在财务和工程之间建立联系，培养FinOps 文化，这将有助于为业务取得可衡量的成果，并鼓励团队在建设过程中考虑成本。“架构完善框架的成本优化支柱”详细介绍了建立良好的财务实践，但我们将在本白皮书中介绍一些基本原则。

成本分配标签

成本分配是指按照既定流程，将发生的成本分配或分发给这些成本的使用者或受益人。在本白皮书的背景下，我们将成本分配分为两种类型：对账和扣款。

对账工具和机制有助于提高成本意识。扣款有助于收回成本并提高成本意识。对账是关于特定实体（例如业务部门、应用程序、用户或成本中心）产生的成本的列报、计算和报告。例如：“基础设施工程团队负责上个月的 AWS 支出中的 X 美元”。扣款是指通过组织的内部会计流程（例如财务系统或会计凭证）将发生的成本实际记入这些实体的账上。例如：“从基础设施工程团队的 AWS 预算中扣除了 X 美

元”。在这两种情况下，对资源进行适当标记都有助于将成本分配给实体，唯一的区别在于是否有人要付款。

贵组织的财务管理部门可能需要对应用程序、业务部门、成本中心和团队层面产生的成本进行透明核算。在成本分配标签的支持下进行成本归属，可为您提供必要的数据，以准确归属实体从适当标记的资源中产生的成本。

- 问责制 - 确保将成本分配给负责资源使用的人员。可由单一服务点或小组负责支出审查和报告。
- 财务透明度 - 通过为领导层创建有效的控制面板和有意义的成本分析，清晰地了解用于 IT 的现金分配情况。
- 明智的 IT 投资 - 根据项目、应用程序或业务范围跟踪投资回报率，使团队能够做出更好的业务决策，例如，为创收应用程序分配更多资金。

总之，成本分配标签有助于您了解：

- 谁拥有支出并负责对其进行优化？
- 哪些工作负载、应用程序或产品产生了支出？ 在哪个环境或阶段？
- 哪些支出领域增长最快？
- 根据过去的趋势，可以从 AWS 预算中扣除多少支出？
- 特定工作负载、应用程序或产品中的成本优化工作会产生什么影响？

激活用于成本分配的资源标签有助于定义组织内部的衡量方法，从而提供 AWS 使用情况的可见性，提高支出问责制的透明度。它还侧重于在成本和使用情况可见性方面创建适当的粒度水平，并通过成本分配报告和 KPI 跟踪来影响云使用行为。

制定成本分配策略

定义和实施成本分配模型

为 AWS 中部署的资源创建账户和成本结构。确定 AWS 支出成本、这些成本是如何产生的，以及这些成本的产生者或原因之间的关系。常见的成本结构基于 AWS Organizations、AWS 账户、环境和组织内的实体，例如业务范围或工作负载。成本结构可以基于多种属性，以便以不同方式或不同粒度水平审查成本，例如将单个工作负载的成本汇总到其所服务的业务范围。

在选择与预期成果相一致的成本结构时，应根据实施的难易程度和预期的准确性来评估成本分配机制。这可能包括问责制、工具可用性和文化变革方面的考虑因素。AWS 客户通常从以下三种流行的成本分配模式入手：

- 基于账户 - 这种模式所需的工作量最少，对账和扣款的准确性高，适用于具有明确账户结构的组织（与使用多账户组织 AWS 环境白皮书的建议一致）。这样，每个账户的成本都一目了然。在成本可见性和分配方面，您可以使用 AWS Cost Explorer、成本和使用情况报告以及 AWS 预算进行成本监控和跟踪。这些工具提供按AWS 账户筛选和分组的选项。从成本分配的角度来看，这种模式不必依赖于对单个资源的准确标记。
- 业务部门或基于团队 - 成本可分配给企业内部团队、业务部门或组织。这种模式所需的工作量适中，对账和扣款的准确性较高，适用于具有明确账户结构（通常使用 AWS Organizations）、不同团队、应用程序和工作负载类型之间分开的组织。这为各团队和应用程序提供了清晰的成本可视性，并降低了在单个 AWS 账户 内达到 AWS 服务限额的风险。例如，每个团队可能有五个帐户（`prod`、`staging`、`test`、`dev`、`sandbox`），并且任何两个团队和应用程序都不会共用同一个账户。拥有这样的结构，AWS Cost Categories 就可以提供将账户或其他标签（“元标记”）归类的功能，并可通过上例中提到的工具进行跟踪。值得注意的是，AWS Organizations 允许对账户和组织单位 (OU) 进行标记，但是这些标签不适用于成本分配和账单报告（也就是说，您不能在 AWS Cost Explorer 中按组织单位对成本进行分组或筛选）。AWS为此，应使用 Cost Categories。
- 基于标签 - 与前两种模式相比，这种模式需要更多的工作量，但可以根据要求和最终目标提供高高准确性的对账和扣款。虽然我们强烈建议您采用使用多账户组织 AWS 环境白皮书中概述的做法，但实际上，客户往往会发现自己的账户结构混合而复杂，需要花时间才能摆脱这种结构。在这种情况下，实施严格有效的标记策略是关键，其次是在账单与成本管理控制台中激活用于成本分配的相关标签（在 AWS Organizations 中，只能从管理付款人账户激活用于成本分配的标签）。在激活用于成本分配的标签后，就可以使用前面方法中提到的成本可见性和分配工具进行对账和扣款。请注意，成本分配标签不具有追溯性，只有在激活用于成本分配的账单报告和成本跟踪工具后，它们才会出现在账单报告和成本跟踪工具中。

总之，如果您需要按业务部门跟踪成本，则可以使用AWS Cost Categories 对 AWS 组织内的关联账户进行相应的分组，并在账单报告中查看分组情况。在为生产环境和非生产环境分别创建账户后，您还可以在 AWS Cost Explorer 等工具中筛选与环境相关的成本，或使用 AWS 预算跟踪这些成本。最后，如果您的用例需要更精细的成本跟踪，例如按单个工作负载或应用程序进行成本跟踪，则可以对这些账户中的资源进行相应标记，在管理账户上激活这些用于成本分配的标签密钥，然后在账单报告工具中按标签密钥筛选该成本。

建立成本报告和监测程序

首先确定对内部利益相关者重要的成本类型（例如，每日支出、按账户计算的成本、按 X 计算的成本、摊销成本）。与等待最终确定的 AWS 发票相比，这样做可以更快地降低与意外或异常支出相关的预算风险。标签提供了实现这些报告方案的属性。从报告中获得的洞察力可以为您的行动提供依据，以

减轻异常和意外支出对财务预算的影响。当成本出现意外激增时，一定要评估交付的价值是否出现意外激增，这样您就可以确定是否需要采取行动以及需要采取哪些行动。

在制定支持成本分配的标记策略时，应牢记以下要素：

- AWS Organizations - 多个账户内的成本分配可按账户、账户组或为这些账户上的资源创建的标签组进行。为 AWS Organizations 中单个账户中的资源创建的标签只能用于管理账户的成本分配。
- AWS 账户 - 一个 AWS 账户 内的成本分配可按服务或区域等其他维度进行。还可以进一步标记账户内的资源，并使用这些资源标签组。
- 成本分配标签 - 必要时，用户创建的标签和 AWS 生成的标签都可激活，用于成本分配。在账单控制台（AWS Organizations 的管理账户）中启用成本分配标签，有助于显示对账和扣款。
- 成本类别 - AWS Cost Categories 允许在一个 AWS 组织内对账户和标签进行分组（“元标记”），这进一步提供了通过 AWS Cost Explorer、AWS Budgets 和 AWS 成本与使用情况报告等工具分析与这些类别相关的成本的功能。

为企业内的业务部门、团队或组织进行对账和扣款

在成本结构和成本分配标签的支持下，使用成本分配流程进行成本分配。标签可用于向不直接负责支付成本但对产生这些成本负责的团队提供反馈。这种方法使人们了解他们对支出的贡献以及这些费用是如何产生的。向直接负责成本的团队执行扣款，以收回他们所消耗的资源支出，并让他们了解这些成本及其产生的方式。

衡量和流通效率或价值 KPI

商定一组单位成本或 KPI 指标，以衡量您的云财务管理投资的影响。这项工作为技术和业务利益相关者创造了一种共同语言，并讲述了一个基于效率的故事，而不是一个只关注绝对总支出的故事。如需了解更多信息，请查看此博客，其中介绍了单位指标如何有助于在业务职能之间建立一致性。

分配不可分配的支出

根据组织的会计惯例，不同的收费类型可能需要不同的处理方法。确定无法标记的资源或成本类别。根据所使用的服务和计划使用的服务，商定如何处理和衡量此类不可分配支出的机制。例如，请查看《AWS Resource Groups 和标签用户指南》中 AWS Resource Groups 和标签编辑器支持的资源列表。

无法标记的成本类别的一个常见例子是一些基于承诺的折扣的一些成本，如预留实例 (RI) 和节省计划 (SP)。虽然订阅费用和未使用的 SP 和 RI 成本无法在账单报告工具中提前标记，但您可以在事后跟踪 RI 和 SP 折扣如何应用于 AWS Organizations 中的账户、资源及其标签。例如，在 AWS Cost Explorer 中可以查看摊销成本，根据相关标签密钥对支出进行分组，并应用与您的用例相关的筛选条

件。在 AWS 成本和使用情况报告 (CUR) 中，您可以筛选出与 RI 和 SP 折扣所涵盖的使用情况相对应的行（请在 CUR 文档的用例部分中阅读更多内容），然后选择仅与您相关的列。为成本分配而激活的每个标签密钥都将显示在 CUR 报告末尾的单独一栏中，这与其他传统账单报告（例如月度成本分配报告）中的显示方式类似。如需更多参考信息，请查看 AWS Well-Architected Labs ，了解从 CUR 数据中获取成本和使用情况洞察力的示例。

报告

除了可用的 AWS 工具来帮助处理对账和扣款外，还有一系列其他 AWS 已创建的和第三方的解决方案可以帮助监控标记资源的成本，并衡量标记策略的有效性。根据组织的要求和最终目标，可以投入时间和资源来构建定制的解决方案，也可以购买 AWS Cloud 管理工具能力合作伙伴提供的工具。如果您决定创建自己的单一真实来源成本分配工具，其中包含与业务相关的受控参数，则 AWS 成本和使用情况报告 (CUR) 可提供最详细的成本和使用情况数据，并可创建自定义优化控制面板，允许按账户、服务、成本类别、成本分配标签和其他多个维度进行筛选和分组。在 AWS 开发的基于 CUR 的解决方案中，有一种可用作上述工具，请查看 AWS Well-Architected Labs 网站上的云智能控制面板。

运营和支持的标签

一个 AWS 环境将有多个账户、资源和工作负载，其运营要求各不相同。标签可用于提供上下文和指导，以支持运营团队加强对服务的管理。标签还可用于为托管资源提供运营治理的透明度。

推动一致定义运营标签的一些主要因素是：

- 在自动基础设施活动期间筛选资源。例如，在部署、更新或删除资源时。另一个是扩大资源规模，以优化成本并减少非工作时间的使用量。有关工作示例，请参阅 AWS 实例调度程序解决方案。
- 识别孤立或已弃用的资源。应适当标记已超过规定使用期限或已被内部机制标记为需要隔离的资源，以协助支持人员进行调查。在隔离、存档和删除之前，应标记已弃用的资源。
- 一组资源的支持要求。资源通常有不同的支持要求，例如，这些要求可以在团队之间协商确定，也可以作为应用程序关键性的一部分来设定。有关运营模式的更多指导，请参阅卓越运营支柱。
- 加强事件管理流程。通过对资源进行标记，提高事件管理流程的透明度，支持团队和工程师以及重大事件管理 (MIM) 团队可以更有效地管理事件。
- 备份。标签还可用于确定需要备份资源的频率、备份副本的去向或恢复备份的位置。有关 AWS 的备份和恢复方法的规范性指导。
- 修补。修补AWS中运行的可变实例对于您的总体修补策略和修补未修补漏洞都至关重要。有关更广泛的修补策略的更深入的指导，请参阅规范性指南。本博客中讨论了修补未修补漏洞的问题。
- 运营可观察性。将运营关键绩效指标策略转换为资源标签，有助于运营团队更好地跟踪目标是否得到满足，从而提高业务需求。制定关键绩效指标策略是一个单独的话题，但往往侧重于处于稳定状态的

业务运营，或者在哪里衡量变革的影响和成果。KPI 控制面板 (AWS Well-Architected Labs) 和运营 KPI 研讨会（AWS 企业支持主动服务）都解决了衡量稳定状态下的绩效问题。AWS 企业策略博客文章《衡量变革的成功》https://aws.amazon.com/blogs/enterprise-strategy/measuring-the-success-of-your-transformation/探讨了如何衡量变革计划的关键绩效指标，例如 IT 现代化或从本地迁移到 AWS。

自动基础设施活动

在管理基础设施时，标签可用于各种自动化活动。例如，通过使用 AWS Systems Manager，您可以在您创建的已定义密钥值对指定的资源上管理自动化和运行手册。对于托管式节点，您可以定义一组标签，按操作系统和环境跟踪或锁定节点。然后，您可以为组中的所有节点运行更新脚本，或查看这些节点的状态。还可以 Systems Manager Resources 进行标记，以进一步完善和跟踪您的自动化活动。

自动化环境资源的启动和停止生命周期可以显著降低任何组织的成本。AWS上的实例调度程序是一个解决方案示例，它可以在不需要时启动和停止 Amazon EC2 和 Amazon RDS 实例。例如，使用 Amazon EC2 或 Amazon RDS 实例的开发人员环境，如果不需要在周末运行，就无法利用关闭这些实例所带来的成本节约潜力。通过分析团队及其环境的需求，并对这些资源进行正确标记以实现自动化管理，就能有效利用预算。

实例调度程序在 Amazon EC2 实例上使用的调度标签示例：

```
{
    "Tags": [
        {
            "Key": "Schedule",
            "ResourceId": "i-1234567890abcdef8",
            "ResourceType": "instance",
            "Value": "mon-9am-fri-5pm"
        }
    ]
}
```

工作负载生命周期

审查支持性运营数据的准确性。确保定期审查与工作负载生命周期相关的标签，并确保相应的利益相关者参与这些审查。

表 7 - 查看作为工作负载生命周期一部分的运营标签

用例	标签密钥	理由	示例值
账户所有者	example-inc:account-owner:owner	账户的所有者及其包含的资源。	ops-center ，dev-ops, app-team
账户所有者审查	example-inc:account-owner:review	审查账户所有权详细信息是否最新且正确。	<以标签库中定义的正确格式表示的审查日期>
数据所有者	example-inc:data-owner:owner	存放数据的账户的数据所有者。	bi-team, logistics ，security
数据所有者审查	example-inc:data-owner:review	审查数据所有权详细信息是否最新且正确。	<以标签库中定义的正确格式表示的审查日期>

在迁移到暂停的 OU 之前为暂停的账户分配标签

在按照《使用多个账户组织 AWS 环境》https://docs.aws.amazon.com/whitepapers/latest/organizing-your-aws-environment/organizing-your-aws-environment.html白皮书中的详细说明暂停账户并将其移入暂停的 OU 之前，应为账户添加标签，以帮助您对账户的生命周期进行内部跟踪和审计。例如，组织的 ITSM 票务系统上的相对网址或票务参考，可显示被暂停的应用程序的审计跟踪记录。

表 8 - 在工作负载生命周期进入新阶段时添加操作标签

用例	标签密钥	理由	示例值
账户所有者	example-inc:account-owner:owner	账户的所有者及其包含的资源。	ops-center ，dev-ops, app-team
数据所有者	example-inc:data-owner:owner	存放数据的账户的数据所有者。	bi-team, logistics ，security

用例	标签密钥	理由	示例值
暂停状态	example-inc:suspension:date	账户被暂停的日期	<以标记库中定义的正确格式表示的暂停日期>
暂停批准	example-inc:suspension:approval	账户暂停批准链接	workload/deprecation

事件管理

从事件记录、优先级排序、调查、沟通、解决到关闭，标签在事件管理的各个阶段都能发挥重要作用。

标签可详细说明应在何处记录事件、应将事件通知哪个团队或哪些团队，以及定义的上报优先级。请务必记住，标签未加密，因此请考虑您在标签中存储了哪些信息。此外，在组织、团队和报告关系中，职责会发生变化，因此请考虑存储安全门户网站的链接，以便更有效地管理这些信息。这些标签不一定是排他性的。例如，应用程序 ID 可用于在 IT 服务管理门户网站中查找上报路径。请确保在操作定义中明确说明该标签有多种用途。

还可以详细列出操作要求标签，以帮助事件管理人员和操作人员进一步完善应对事件或活动的目标。

运行手册和操作手册的相关链接（指向知识系统库网址）可作为标签纳入，以帮助响应团队识别相应的流程、程序和文档。

表 9 - 使用操作标签为事件管理提供信息

用例	标签密钥	理由	示例值
事件管理	example-inc:incident-management:escalationlog	支持团队使用的事件记录系统	jira, servicenow ，zendesk
事件管理	example-inc:incident-management:escalationpath	上报路径	ops-center ，dev-ops, app-team

用例	标签密钥	理由	示例值
成本分配和事件管理	example-inc:cost-allocation:CostCenter	按成本中心监控成本。这是一个双重用途标签的示例，其中成本中心被用作事件记录的应用程序代码	123-*
备份计划	example-inc:backup:schedule	资源备份计划	Daily
操作手册/事件管理	example-inc:incident-management:playbook	有记录的操作手册	webapp/incident/playbook

修补

通过使用 AWS Systems Manager Patch Manager 和 AWS Lambda，组织可以自动执行针对可变计算环境的修补策略，并使可变实例与该应用环境中定义的补丁基准保持一致。通过将上述实例分配到补丁组和维护窗口，可以管理这些环境中可变实例的标记策略。有关开发 → 测试 → 生产拆分，请参阅以下示例。可变实例的补丁管理有AWS规范性指导。

表 10 — 操作标签可能因环境而异

开发	生产前调试	生产
{ "Tags": [{ "Key": "Maintenance Window", "ResourceId": "i-012345678ab9ab111", "ResourceType": "instance",	{ "Tags": [{ "Key": "Maintenance Window", "ResourceId": "i-012345678ab9ab444", "ResourceType": "instance",	{ "Tags": [{ "Key": "Maintenance Window", "ResourceId": "i-012345678ab9ab777", "ResourceType": "instance",

开发	生产前调试	生产

"Value": "cron(30 23 ? * TUE#1 *)"
},
{
"Key": "Name",
"ResourceId": "i-012345678ab9ab222",
"ResourceType": "instance",
"Value": "WEBAPP"
},
{
"Key": "Patch Group",
"ResourceId": "i-012345678ab9ab333",
"ResourceType": "instance",
"Value": "WEBAPP-DEV-AL2"
}
]
}
	"Value": "cron(30 23 ? * TUE#2 *)"
	},
	{
	"Key": "Name",
	"ResourceId": "i-012345678ab9ab555",
	"ResourceType": "instance",
	"Value": "WEBAPP"
	},
	{
	"Key": "Patch Group",
	"ResourceId": "i-012345678ab9ab666",
	"ResourceType": "instance",
	"Value": "WEBAPP-TEST-AL2"
	}
	]
	}
		"Value": "cron(30 23 ? * TUE#3 *)"
		},
		{
		"Key": "Name",
		"ResourceId": "i-012345678ab9ab888",
		"ResourceType": "instance",
		"Value": "WEBAPP"
		},
		{
		"Key": "Patch Group",
		"ResourceId": "i-012345678ab9ab999",
		"ResourceType": "instance",
		"Value": "WEBAPP-PROD-AL2"
		}
		]
		}

也可以通过定义标签来管理未修补漏洞，以补充修补策略。有关详细指南，请参阅使用 AWSSystems Manager 进行当天安全修补，避免未修补漏洞。

运营可观察性

需要具备可观察性，才能深入了解环境的性能，并帮助您发现和调查问题。它还有一个次要用途，即允许您定义和衡量关键绩效指标 (KPI) 和服务级别目标 (SLO)，例如正常运行时间。对于大多数组织来说，重要的运营 KPI 是检测到事件的平均时间 (MTTD) 和从事件中恢复的平均时间 (MTTR)。

在整个可观察性过程中，上下文非常重要，因为数据收集后，相关的标签也会被收集。无论您关注的是哪种服务、应用程序或应用程序层，您都可以针对特定数据集进行筛选和分析。标签可用于自动加入 CloudWatch 警报，以便在突破某些指标阈值时可以向正确的团队发出警报。例如，标签密钥 `example-inc:ops:alarm-tag` 及其上的值可能表示已创建 CloudWatch 警报。使用标签为 Amazon EC2 实例创建和维护 Amazon CloudWatch 警报一文中描述了一个演示此功能的解决方案。

配置过多的警报很容易造成警报风暴 - 大量警报或通知迅速压垮操作员，降低他们的整体效率，同时操作员还要手动分类各个警报并确定其优先级。警报的其他上下文可以标签的形式提供，这意味着可以在 Amazon EventBridge 中定义规则，以帮助确保关注上游问题而非下游依赖关系。

运维与 DevOps 的作用经常被忽视，但对许多组织来说，中央运维团队仍在正常工作时间之外提供关键的第一响应。（有关该模式的更多详细信息，请参阅卓越运营白皮书。）与负责工作负载的 DevOps 团队不同，他们的知识深度通常不一样，因此标签在控制面板和警报中提供的上下文可以引导他们针对问题找到正确的运行手册，或者启动自动运行手册（请参阅博客文章使用 AWS Systems Manager 自动处理 Amazon CloudWatch 警报）。

数据安全、风险管理和访问控制的标签

在妥善处理数据存储和处理方面，Organizations 有不同的需求和义务需要履行。数据分类是访问控制、数据留存、数据分析和合规性等多个用例的重要先决条件。

数据安全和风险管理

在 AWS 环境中，您的账户可能会有不同的合规性和安全要求。例如，您可能有一个开发人员沙盒和一个托管生产环境的账户，用于处理付款等高度受监管的工作负载。通过将它们隔离到不同的账户中，您可以应用不同的安全控制措施，限制对敏感数据的访问，并缩小受监管工作负载的审计范围。

对所有工作负载采用单一标准可能会带来挑战。虽然许多控制措施同样适用于整个环境，但对于不需要满足特定监管框架的账户和不会出现个人身份数据的账户（例如，开发人员沙盒或工作负载开发账户）来说，有些控制措施过多或无关紧要。这通常会导致假阳性安全发现，必须在不采取任何行动的情况下对这些调查发现进行分类和关闭，这就占用了本应调查的调查发现的精力。

表 11 - 数据安全和风险管理标签示例

用例	标签密钥	理由	示例值
事件管理	`example-inc:incident-management:escalationlog`	支持团队使用的事件记录系统	`jira`, `servicenow`，`zendesk`
事件管理	`example-inc:incident-management:escalationpath`	上报路径	`ops-center`，`dev-ops`, `app-team`

用例	标签密钥	理由	示例值
数据分类	`example-inc:data:classification`	对数据进行分类以实现合规性和治理	`Public`, `Private`, `Confidential` ，`Restricted`
合规性	`example-inc:compliance:framework`	确定工作负载所遵循的合规性框架	`PCI-DSS`, `HIPAA`

在整个 AWS 环境中手动管理不同的控制措施既费时又容易出错。下一步是自动部署适当的安全控制措施，并根据该账户的分类配置对资源的检查。通过对账户及其中的资源应用标签，可以自动部署控制措施，并根据工作负载进行适当配置。

示例：

工作负载包括一个 Amazon S3 存储桶，其标签为 `example-inc:data:classification`，值为 `Private`。安全工具自动化会部署 AWS Config 规则 `s3-bucket-public-read-prohibited`，该规则会检查 Amazon S3 存储桶的阻止公开访问设置、存储桶策略和存储桶访问控制列表 (ACL)，确认存储桶的配置适合其数据分类。为了确保存储桶的内容与分类一致，可以将 Amazon Macie 配置为检查个人身份信息 (PII)。在使用 Amazon Macie 验证 S3 存储桶数据分类的博客中更深入地探讨了这种模式。

某些监管环境，例如保险和医疗保健，可能需要遵守强制性的数据留存政策。使用标签进行数据留存，再加上 Amazon S3 生命周期策略，可以有效而简单地将对象转移到不同的存储层。Amazon S3 生命周期规则也可用于在强制保留期到期后使对象过期以进行数据删除。有关此过程的深入指南，请参阅使用 Amazon S3 生命周期对象标签简化数据生命周期。

此外，在对安全调查发现进行分类或处理时，标签可以为调查人员提供重要的上下文信息，有助于确定风险的性质，并有助于让适当的团队参与调查或减轻调查发现的影响。

身份管理和访问控制的标签

使用 AWS IAM Identity Center 管理 AWS 环境中的访问控制时，标签可启用多种扩展模式。有几种授权模式可以应用，其中一些基于标记。我们将逐一讨论，并提供进一步阅读的链接。

单独资源的 ABAC

IAM Identity Center 用户和 IAM 角色支持基于属性的访问权限控制 (ABAC)，允许您根据标签定义对操作和资源的访问。ABAC 有助于减少更新权限策略的需求，并帮助您根据公司目录中的员工属性进行访问。如果您已经在使用多账户策略，则除了基于角色的访问权限控制 (RBAC) 外，还可以使用 ABAC 为在同一账户中操作的多个团队提供对不同资源的精细访问权限。例如，IAM Identity Center 用户或 IAM 角色可以包含限制访问特定 Amazon EC2 实例的条件，否则这些实例必须在每个策略中明确列出才能访问。

由于 ABAC 授权模式依赖于标签来访问操作和资源，因此一定要提供防护栏以防止意外访问。SCP 仅允许在特定条件下修改标签，从而可用于保护整个组织的标签。在 AWS Organizations和 IAM 实体的权限边界中使用服务控制策略确保用于授权的资源标签安全的博客中，提供了有关如何实施的信息。

如果使用使用期长的 Amazon EC2 实例来支持更传统的操作实践，则可以使用这种方法，在为 Amazon EC2 实例和系统管理器会话管理器配置 IAM Identity Center ABAC 的博客中更详细地讨论了这种基于属性的访问控制形式。如前所述，并非所有资源类型都支持标记，而在支持标记的资源类型中，也并非所有资源类型都支持使用标记策略来执行，因此在 AWS 账户 上开始实施这一策略之前，最好先对此进行评估。

要了解支持 ABAC 的服务，请参阅与 IAM 一起使用的 AWS 服务。

结论

对 AWS 资源进行标记可以出于各种目的，包括实施成本分配战略、支持自动化或授权访问 AWS 资源。由于涉及的利益相关方群体数量众多，以及诸如数据来源和标签治理之类的注意事项，实施标签策略对某些组织来说可能具有挑战性。

在本白皮书中，我们根据操作实践、已定义的用例、流程中涉及的利益相关者以及 AWS 提供的工具和服务，概述了有关在组织中设计和实施标记策略的建议。就标记策略而言，这是一个迭代和改进的过程，在这个过程中，您要从当务之急小处着手，确定整个组织的相关用例，然后根据需要实施和发展标记方案，同时不断衡量和提高有效性。我们已经指出，在组织内部建立一套定义明确的标签，可以将 AWS 的使用和消耗与负责资源和业务目的的团队联系起来，从而与组织战略和价值保持一致。

贡献者

本文档的贡献者包括：

- Amazon Web Services 高级专业技术客户经理 Chris Pates
- Amazon Web Services 企业支持主管 Vijay Shekhar Rao
- Amazon Web Services 高级专业技术客户经理 Nataliya Godunok
- Amazon Internet Services Private Limited 高级解决方案架构师 Yogish Kutkunje Pai
- Amazon Web Services 高级专业技术客户经理 Jamie Ibbs

延伸阅读

有关更多信息，请参阅

- AWS re:Invent 2020：逆向工作：Amazon 实现创新的方法
- AWS规范性指导：使用 AWS Systems Manager 为混合云中的可变实例自动打补丁
- AWS 架构中心

AWS Well-Architected

- AWS Well-Architected Framework
- 卓越运营支柱 - AWS Well-Architected Framework
- 灾难恢复 (DR) 计划 - AWS Well-Architected 可靠性支柱
- 成本优化支柱 - AWS Well-Architected Framework
- AWS Well-Architected Labs：启用 AWS 生成的成本分配标签
- AWS Well-Architected Labs：标签策略
- AWS Well-Architected Labs：AWS CUR 查询库

AWS 博客

- AWS Health Aware – 为组织和个人 AWS 账户定制 AWS Health 提醒
- 如何根据 API 事件自动标记 Amazon EC2 资源
- AWS 生成的与用户定义的成本分配标签
- 使用 AWS Organizations 进行成本标记和报告
- 使用 AWS Systems Manager 补丁管理器为 Windows EC2 实例打补丁
- 使用 AWS Systems Manager 同日安全补丁避免零日漏洞

AWS 文档

- 使用成本分配标签 - AWS Billing and Cost Management
- 什么是 AWS 成本和使用情况报告
- AWS Resource Groups API 参考
- 如何使用 IAM 策略标签来限制 EC2 实例或 EBS 卷的创建方式？

- 可变与不可变的更新模型

其他

- Bryar, C. 和 Carr, B. (2021)。Working Backwards: Insights, Stories, and Secrets from Inside Amazon. London Macmillan
- AWS CloudFormation Guard (GitHub)

文档修订

如需获取有关该白皮书更新的通知，请订阅 RSS 信息源。

变更	说明	日期
次要更新	身份管理更新	2023 年 3 月 30 日
次要修订	更新了 ABAC 中有关个别资源的参考。	2023 年 2 月 24 日
次要修订	更新了指南，使其符合 IAM 最佳实践。有关更多信息，请参阅 IAM 安全最佳实践。	2023 年 2 月 6 日
主要修订	为 AWS Config 规则 `required_tags` 支持的资源类型添加了更具体的参考。	2023 年 1 月 18 日
主要修订	更新内容包括最新的实践和服务能力，特别是在身份识别领域。	2022 年 9 月 29 日
次要更新	修正了 PDF 版本中的表格格式。	2022 年 4 月 25 日
主要修订	更新了文档结构，扩展了“标记策略”和“用例”部分。根据最新的工具、技术和可用资源，添加了更多规范性指导。	2022 年 4 月 22 日
初次发布	白皮书首次发布。	2018 年 12 月 1 日

> **Note**
>
> 要订阅 RSS 更新，您必须为当前使用的浏览器启用 RSS 插件。

注意事项

客户有责任对本文档中的信息进行单独评测。本文档：（a）仅供参考，（b）代表当前的 AWS 产品和实践，如有更改，恕不另行通知，以及（c）不构成 AWS 及其附属公司、供应商或许可方的任何承诺或保证。AWS 产品或服务“按原样”提供，不附带任何明示或暗示的保证、陈述或条件。AWS 对其客户承担的责任和义务受 AWS 协议制约，本文档不是 AWS 与客户直接协议的一部分，也不构成对该协议的修改。

AWS 术语表

有关最新的 AWS 术语，请参阅《AWS 词汇表参考》中的 AWS 词汇表。