



Codar

软件版本:1.70

概念指南

文档发布日期:2016年7月

软件发布日期:2016年7月



Hewlett Packard
Enterprise

法律声明

担保

Hewlett Packard Enterprise 产品和服务的唯一担保已在此类产品和服务随附的明示担保声明中提出。此处的任何内容均不构成额外担保。Hewlett Packard Enterprise 不会为此处出现的技术或编辑错误或遗漏承担任何责任。

此处所含信息如有更改，恕不另行通知。

受限权利声明

机密计算机软件。必须拥有 Hewlett Packard Enterprise 授予的有效许可证，方可拥有、使用或复制本软件。按照 FAR 12.211 和 12.212，并根据供应商的标准商业许可的规定，商业计算机软件、计算机软件文档与商品技术数据授权给美国政府使用。

版权声明

© 2015 - 2016 Hewlett Packard Enterprise Development LP

商标声明

Adobe™ 是 Adobe Systems Incorporated 的商标。

Microsoft® 和 Windows® 是 Microsoft Corporation 在美国的注册商标。

OpenStack® Word Mark 和 Square O Design (合并或分开) 是 OpenStack Foundation 在美国和其他国家/地区的商标或注册商标，经 OpenStack Foundation 许可方可使用。

Oracle 和 Java 是 Oracle 和/或其附属公司的注册商标。

RED HAT READY™ 徽标和 RED HAT CERTIFIED PARTNER™ 徽标是 Red Hat, Inc. 的商标。

本产品包括 'zlib' 通用压缩库的接口，该压缩库归 Jean-loup Gailly 和 Mark Adler 版权所有 © 1995-2002。

文档更新

此文档的标题页包含以下标识信息：

- 软件版本号，用于指示软件版本。
- 文档发布日期，该日期将在每次更新文档时更改。
- 软件发布日期，用于指示该版本软件的发布日期。

要检查是否有最新的更新，或者验证是否正在使用最新版本的文档，请访问：<https://softwaresupport.hpe.com/>。

需要注册 HP Passport 才能登录此站点。要注册 HP Passport ID，请在软件支持站点上单击“注册”或在 HP Passport 登录页面上单击“创建帐户”。

此外，如果订阅了相应的产品支持服务，则还会收到更新的版本或新版本。有关详细信息，请与您的销售代表联系。

支持

访问软件支持站点：<https://softwaresupport.hpe.com>。

Hewlett Packard Enterprise 软件联机支持提供客户自助解决功能。通过该联机支持，可快速高效地访问用于管理业务的各种交互式技术支持工具。作为尊贵的支持客户，您可以通过该支持网站获得下列支持：

- 搜索感兴趣的知识文档
- 提交并跟踪支持案例和改进请求
- 下载软件修补程序
- 管理支持合同
- 查找 HPE 支持联系人
- 查看有关可用服务的信息
- 参与其他软件客户的讨论
- 研究和注册软件培训

大多数提供支持的区域都要求您注册为 HP Passport 用户再登录，很多区域还要求用户提供支持合同。要注册 HP Passport ID，请在支持站点上单击“注册”或在 HP Passport 登录页面上单击“创建帐户”。

要查找有关访问级别的详细信息，请访问：<https://softwaresupport.hpe.com/web/softwaresupport/access-levels>。

目录

Codar	5
Codar 概述	6
组件	6
服务设计	6
序列化设计	7
拓扑设计	7
基于声明性的建模	7
拓扑组合	8
微服务	9
应用程序管道管理	11
管理包	12
包操作	13
部署和重新部署	14
扩大	14
角色和用户访问	16
生命周期阶段和操作	19
包状态	20
按生命周期阶段分组服务设计	21
发布门户操作	21
管道统计信息	23
环境	24
计划发布	25
外部集成	26
Jenkins 集成	26
ALM 集成	27
基础结构即代码 (IaaS)	27
在 Docker 群集上部署容器	28
Codar 支持 SAML	29
用例:持续集成、部署和交付	31
应用程序建模	31
持续集成和部署	32

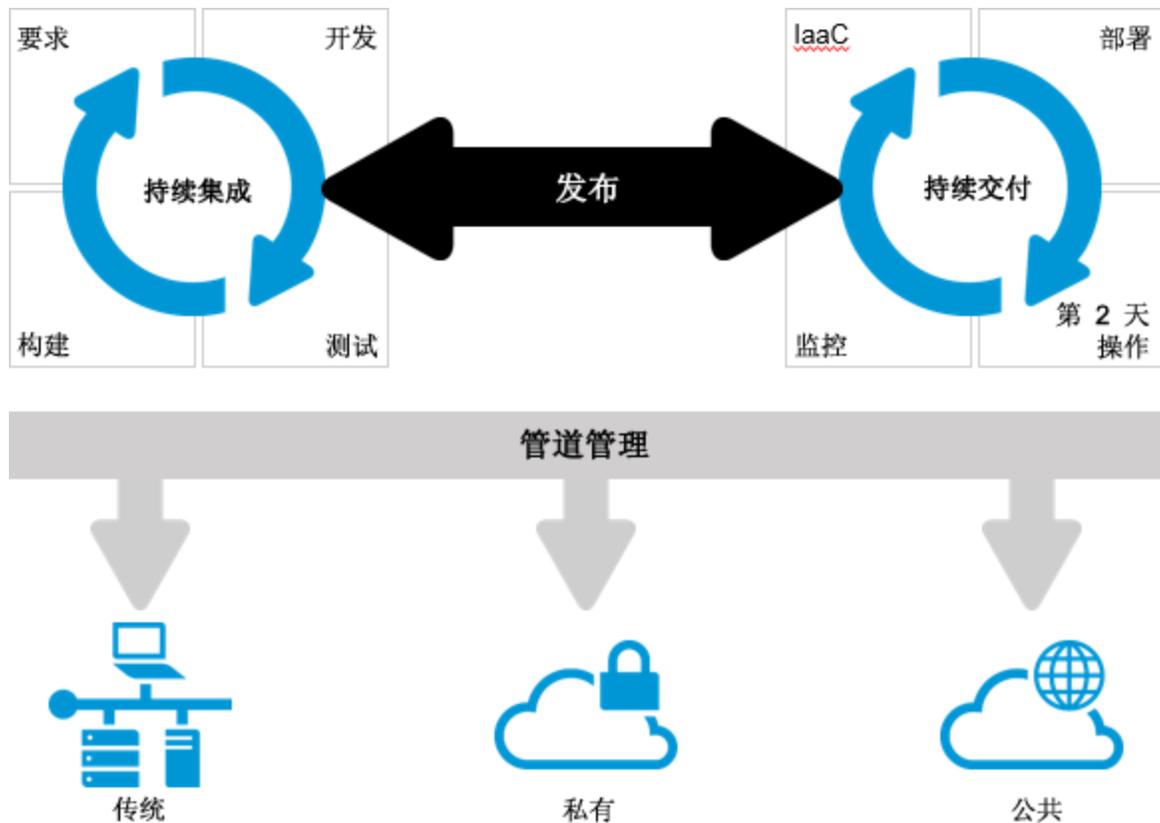
导入应用程序设计	32
在环境中部署	32
发布设计	33
用例:可自定义的发布管道	35
用例:部署和重新部署包	37
用例:部署和扩大	39
后续步骤	40
Codar 社区版	41
发送文档反馈	42

Codar

组织在将持续集成扩展为持续交付时面临新的挑战。挑战包括在考虑到从开发到生产环境之间的差异的前提下通过这些环境一致地部署应用程序。

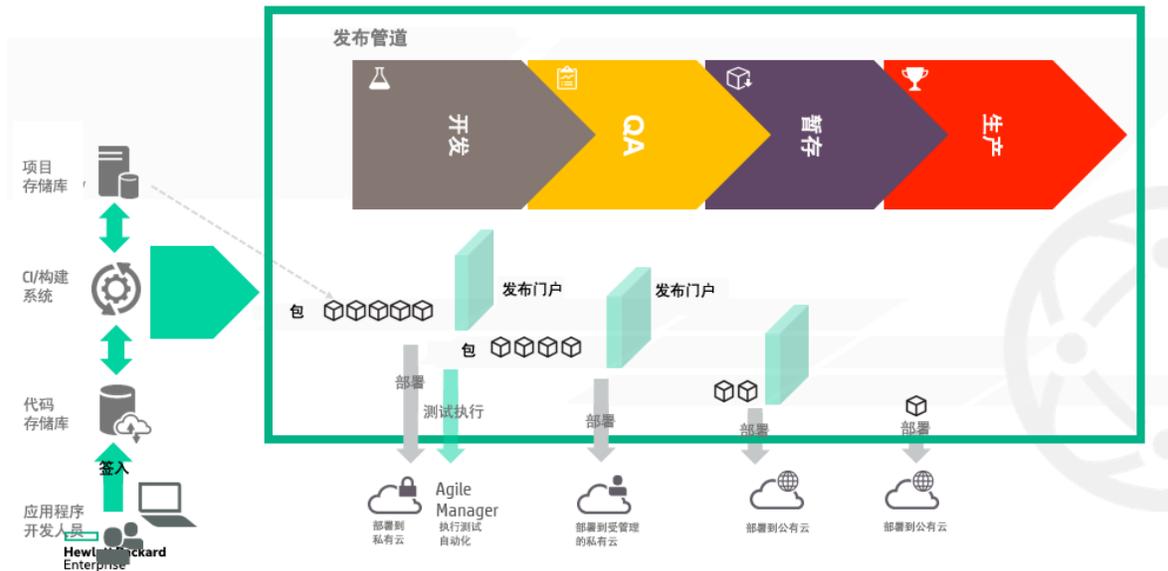
开发运营通过基于协作、自动化和控制使用一组准则、方法和实践，提供了一个框架来缩小开发 (Dev) 和运营 (Ops) 环境之间的差距。目的是将持续生成或程序集集成扩展为异类环境中的可重复且一致的应用程序部署。

下图阐释了开发运营环境中的持续集成和持续交付循环。



Codar 概述

Hewlett Packard Enterprise Codar 提供了持续交付，其中系统的每次更改都是可发布的并且所有代码更改均可在生产中部署。它将实现持续交付的自动化，其中每次代码更改都将触发已部署的生成，执行自动单元测试，并且应用程序将基于 **runbook** 自动流中定义的策略自动部署到环境。持续交付旨在频繁交付并快速获得用户反馈。



组件

组件是服务设计的元素。“组件”选项卡中仅显示拓组件。序列化组件与提供程序或提供程序类型不关联。在“组件”选项卡中，可以查看与特定提供程序实例关联的拓组件并管理这些拓组件。

服务设计

创建、配置和修改服务设计，以提供按需自动化服务交付。服务设计是自动化云的方法，包括可重用的服务组件。服务设计中的服务组件及其关系定义了用于创建服务的框架。

服务设计还提供用户在订购服务时可选择的选项结构。您可以对多个服务产品重用设计，对每个服务产品进行自定义以满足不同用户组织和组的需求。您还可以利用 **Codar** 随附的服务设计并在 **Codar** 系统之间导出和导入设计。

您可以创建序列化和拓扑设计。

序列化设计

序列化设计可指定服务组件生命周期的定向执行过程，并提供用于在部署每个组件时控制资源选择的机制。当创建序列化设计时，需要关联服务组件上的一个或多个资源产品以限制提供程序选择。此关联或链接可确保在部署服务组件时配置资源产品。还可以将资源产品与组件模板关联。

可将序列化设计用于复杂服务和依赖于操作手册自动化的服务，例如与传统数据中心系统的集成。创建序列化设计，将其作为定向的组件层次结构以定义生命周期执行。序列化设计使用组件对单个实体内的多个自动化提供程序进行分组，并且它们允许明确指定生命周期操作。

拓扑设计

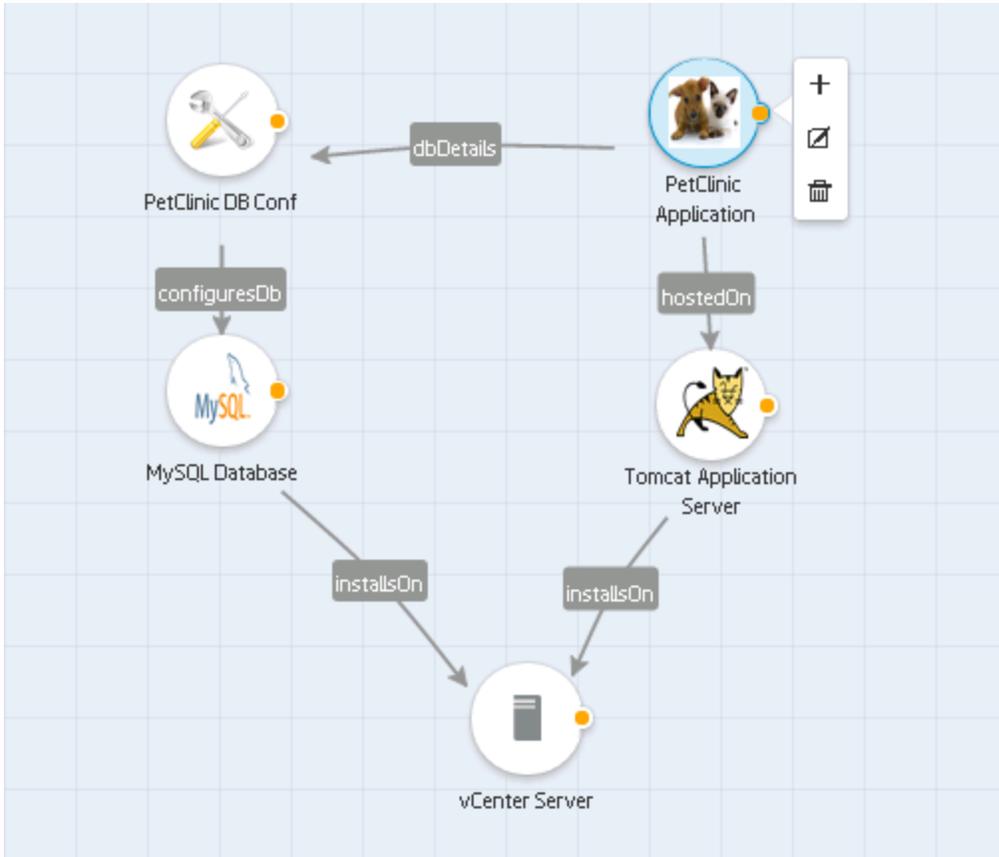
拓扑设计将指定组件、关系和属性。对比于序列化设计 (其更明确地定义将运行的操作的配置顺序和序列)，拓扑设计是声明性设计，不包括任何明确的操作或序列。配置序列由拓扑设计中各组件之间的关系推断出。

使用拓扑设计进行基础结构即服务 (IaaS)、平台即服务 (PaaS) 和软件即服务 (SaaS) 的部署，通过 **Chef**、**Puppet**、**Server Automation** 和基于 **Operations Orchestration** 流的组件启用这些部署。

基于声明性的建模

使用基于声明性的建模可实现应用程序部署的自动化，这将允许用户在后台触发达到应用程序部署的结束状态的过程时声明此状态 (应用程序组件及其之间的依赖关系)。这使用户能够将精力放在部署内容而不是实现部署的方式上，从而在更短时间内实现多层应用程序部署的自动化以及使实时管理简单得多。

Codar 支持通过用户界面对复杂设计执行创建、集成和维护的基于声明性的模型开发。模型由拓扑设计及其属性组成。Codar 使用户能够在实现过程 (类似于后期绑定) 中灵活修改这些属性。



拓扑组合

应用程序设计 (也称为拓扑设计) 指定组件及其关系以定义应用程序生命周期。应用程序设计将生命周期序列委托给云提供商。

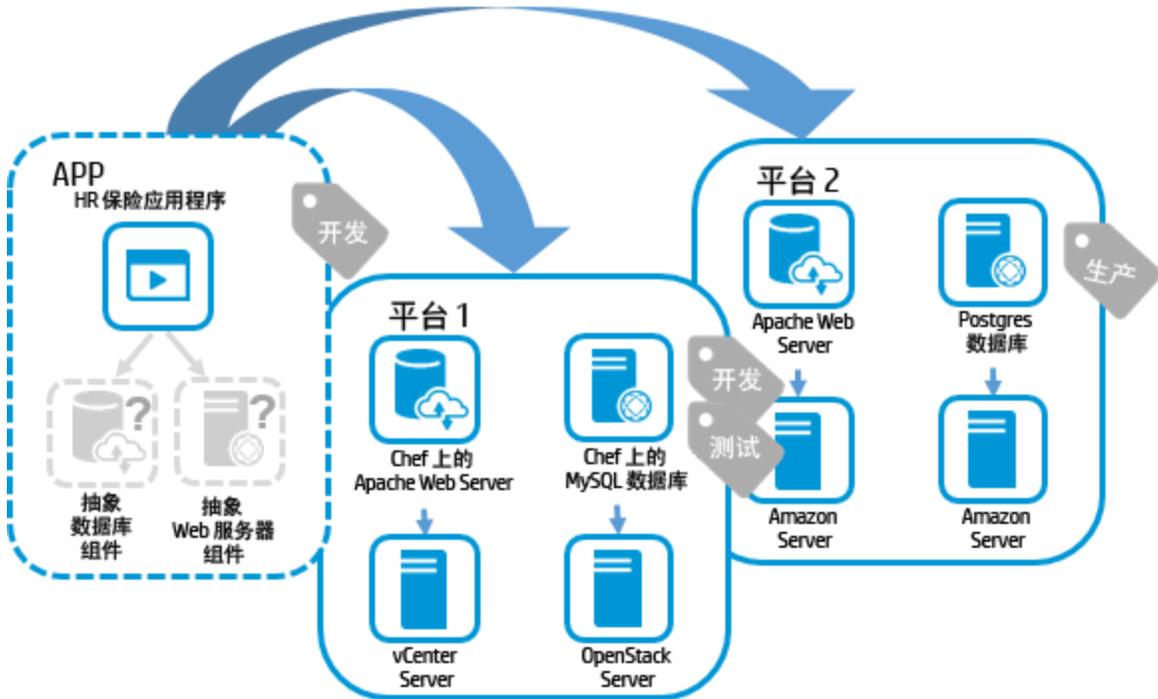
应用程序设计具有两种类型:

- 完整设计: 此设计中的所有组件均存在, 可自行完成实施
- 部分设计: 此设计需要其他设计来完成实施

拓扑组合用于在运行时将应用程序设计与基础结构设计相组合。在应用程序部署期间, 对每个部署的基础结构需求都不同。拓扑组合可帮助定义应用程序设计中的这些可变基础结构需求, 并允许在部署时与不同的基础结构设计组合。

功能和特征用于描述组件。应用程序设计将使用设计中的功能组件和特征定义要求。应用程序设计不能自行进行配置，并且要求选择兼容的服务设计。服务设计组件的功能和特征相匹配，以检查功能和匹配设计是否在部署期间被选为兼容的服务设计。

以下插图显示了 HR 保险应用程序的拓扑组合。该应用程序需要使用在应用程序设计 APP 中定义的数据库组件和 Web 服务器组件。这是通过平台 1 完成的，因为它包含具有 Web 服务器功能及其特征和 MySQL 数据库(具有数据库功能及其特征)的 Apache Web Server。同样地，平台 2 还符合 APP 要求。



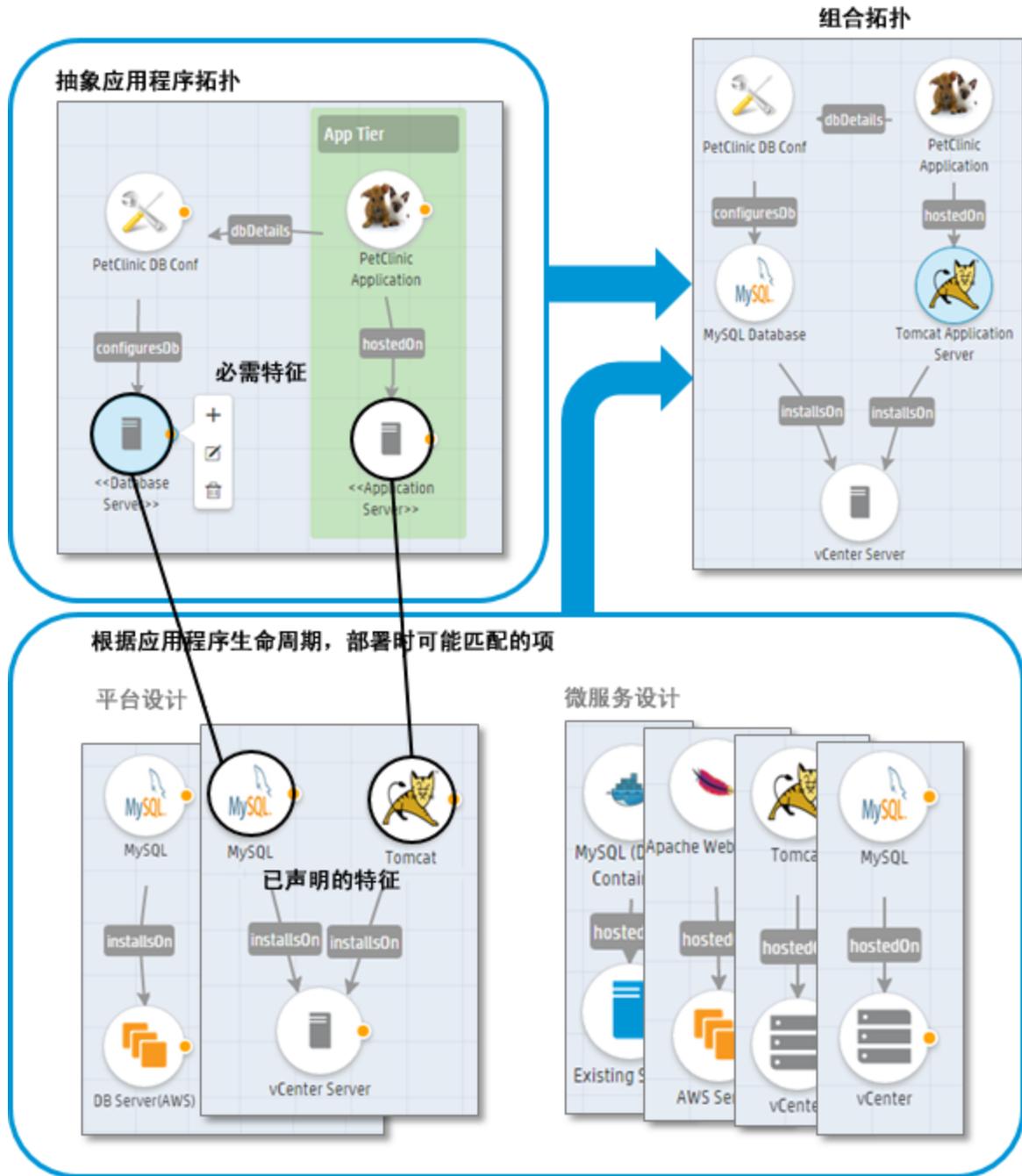
微服务

可以使用多个基础结构设计来部署部分应用程序设计，这些基础结构设计提供平台或基础结构服务而非单个服务。具有多个开放需求(功能)的部分设计可以与多个基础结构设计组合，以满足其所有开放需求并构建完整的应用程序设计。您可以选择匹配所有功能的单个设计，也可以从不同的设计中选择组件。

具有多个开放需求(功能)的部分设计可以与多个(微)服务设计组合，以满足其所有开放需求并构建完整的应用程序设计。

例如，如果应用程序需要数据库和应用程序服务，则可以选择在单个设计中具有数据库和应用程序的设计，也可以从一个服务设计中选择数据库而从另一个服务设计中选择应用程序。

组合拓扑根据选择的微服务在运行时创建。微服务可以与生命周期阶段关联。



应用程序管道管理

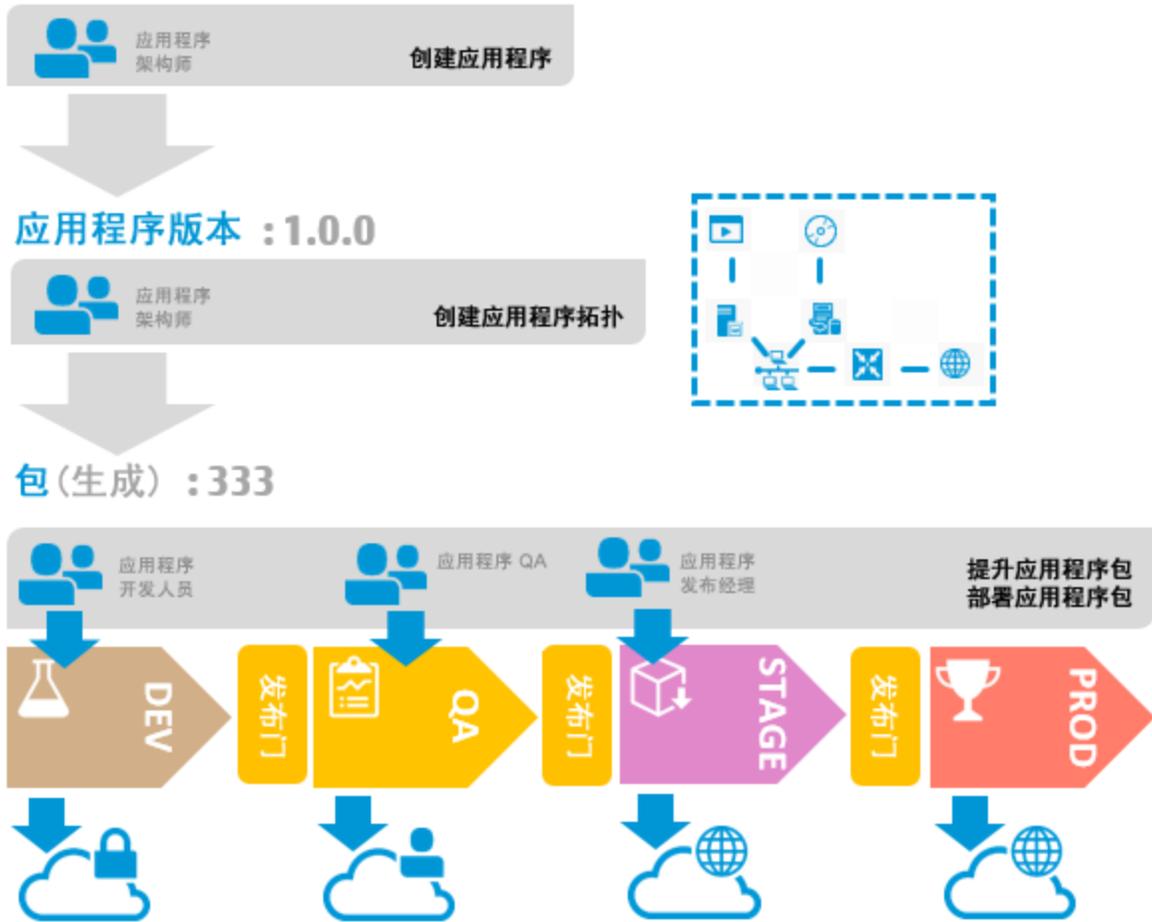
实现应用程序部署的自动化是一个复杂而漫长的过程，需要大量投资。应用程序以不同的方式部署在开发和生产环境中，因此导致出现许多错误。应用程序管道管理允许您使用同一拓扑模型在不同的环境中部署应用程序。您可以在不同的阶段选择不同的微服务；但是，应用程序设计仍保留不变。这意味着同一设计可以在不同的生命周期阶段进行部署和测试。

您还可以自定义发布管道，使得每个应用程序团队可以使用单独的生命周期阶段。这样可以实现完全自动化的连续部署。**Codar**增加了应用程序发布周期的敏捷性，同时通过消除手动步骤，提高了质量并降低了应用程序部署的成本。

Codar中的管道管理包括：

- 创建您自己的角色，从而使您能够创建自己的用户访问结构
- 除预置阶段之外，还可以创建您自己的生命周期阶段
- 选择已存在的资源环境，并将它们仅与某些生命周期阶段关联，从而创建一个由预定义生命周期阶段子集组成的生命周期阶段超集
- 查看管道统计信息并获取您的部署的可视化表示形式
- 根据包、操作和环境筛选您的视图

应用程序 : Pet Clinic



管理包

包表示应用程序设计的快照，允许在设计中参数化属性。也可以说包表示应用程序的特定生成。

包是为应用程序部署的最小单位。它表示实现项目(应部署应用程序的方式)和部署项目(应部署诸如 `war`、`ear` 等库的位置)。

包与生命周期阶段关联。包可属于“开发”、“测试”、“暂存”或“生产”阶段。

包与管道管理关联。可跨生命周期阶段管理包，如在给定阶段中提升或拒绝包。例如，具有 **QA** 角色的用户可以拒绝包。

任务

- **从特定应用程序版本创建包。** 应用程序版本可以包含多个包。
- **部署或重新部署包。** 在这种情况下，将实施应用程序设计的相应状态以及包中指定的设计的属性。
- **删除包。** 转到“发布管道”选项卡，按住 **Ctrl** 选择多个包，然后单击“删除”。

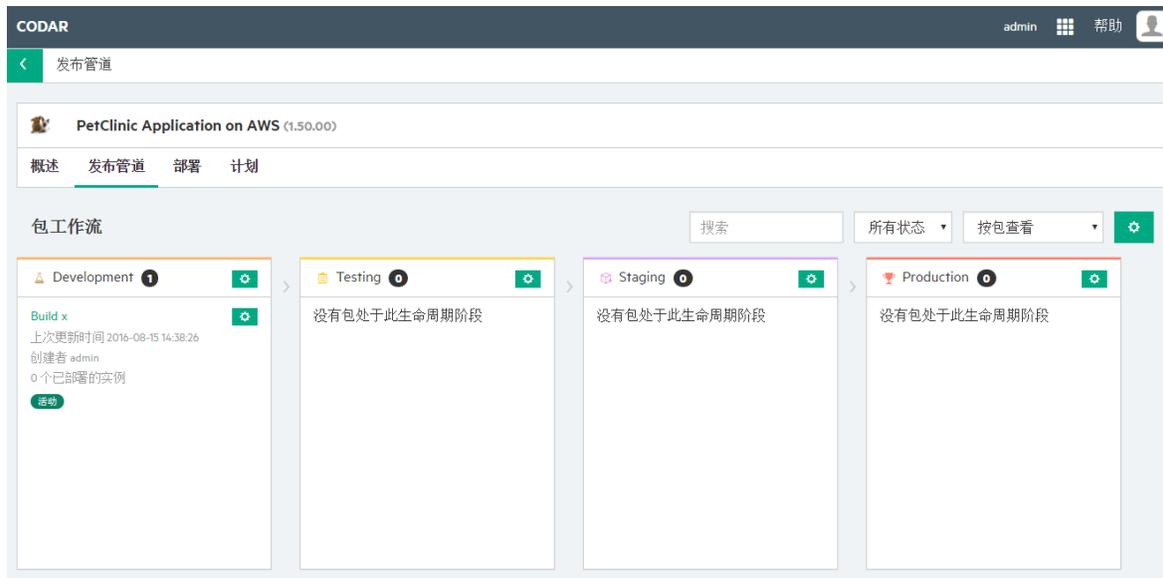
备注：无法删除与实例相关联的包。

包操作

Codar 是用于实施开发运营环境的集中式结构。不同的角色可以部署、重新部署、提升或拒绝包。包以一致且可重复的方式从一个阶段提升到另一个阶段。这样可确保将应用程序投入生产时对团队成员可见。

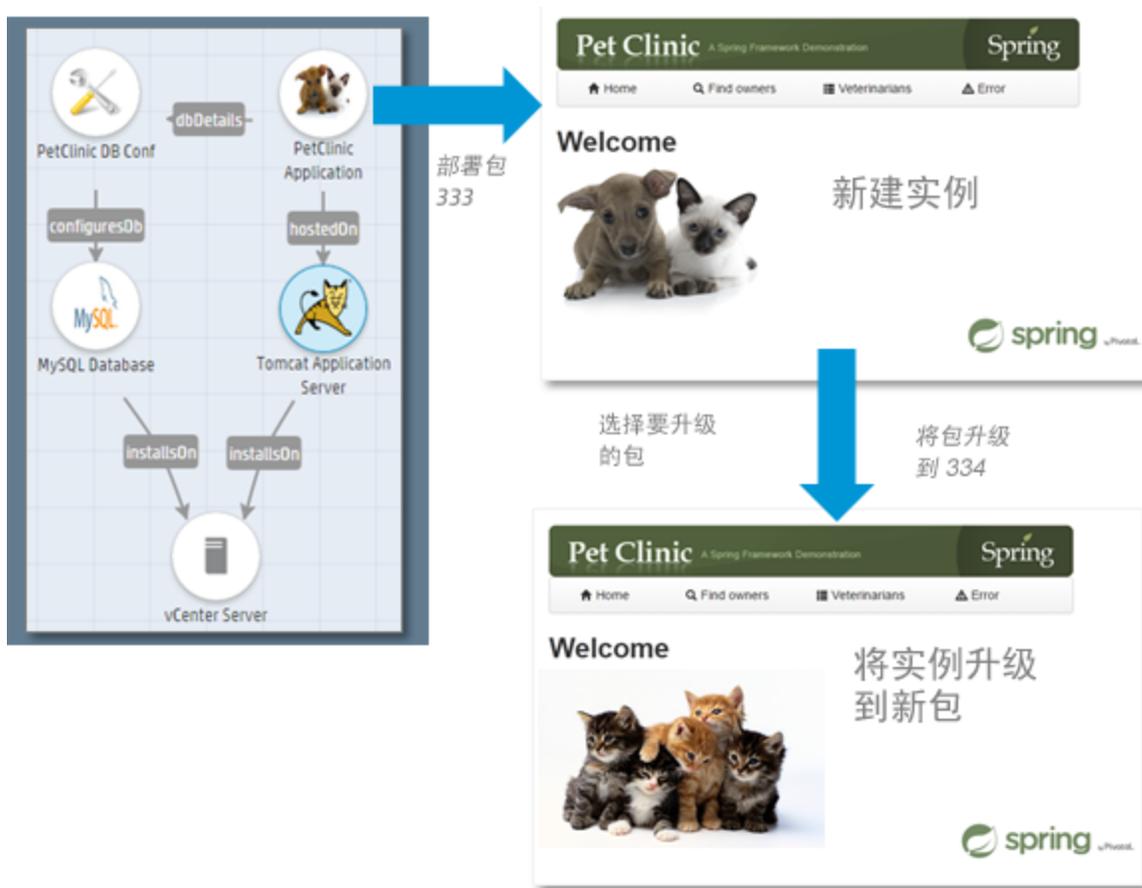
创建和部署包时，将创建新的虚拟机并部署包。您可以在已部署的实例上执行测试，也可以提升或者拒绝包。

Codar 提供了应用程序管道管理功能，如下图所示。



部署和重新部署

可以在具有旧包的实例上重新部署包。您可以查看实例详细信息并选择一个现有实例。重新部署还可以用于升级或修补组件。由于重新部署会调用所有组件的修改操作，因此设计中的所有组件可以升级到新版本。

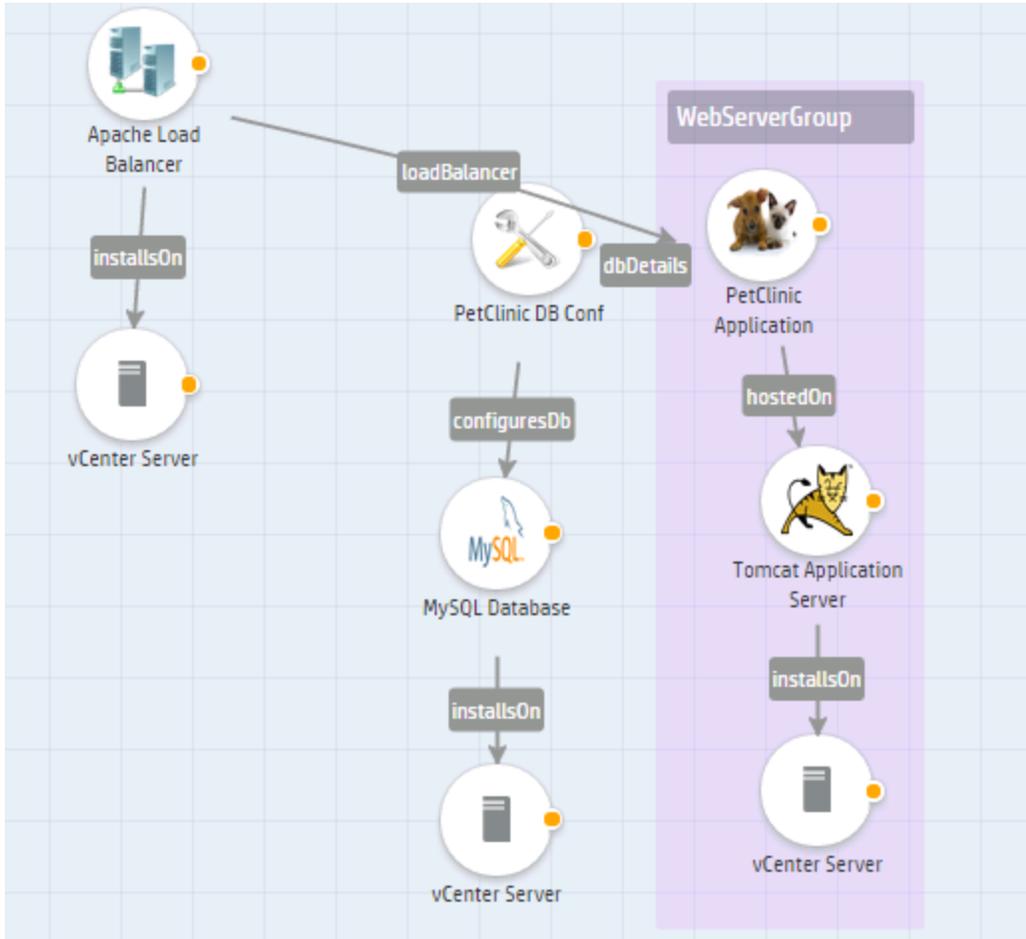


扩大

在拓扑设计创建期间，您可以创建扩展组。扩展组表示可扩展的堆栈。一个应用程序设计中可以有多个可扩展组。

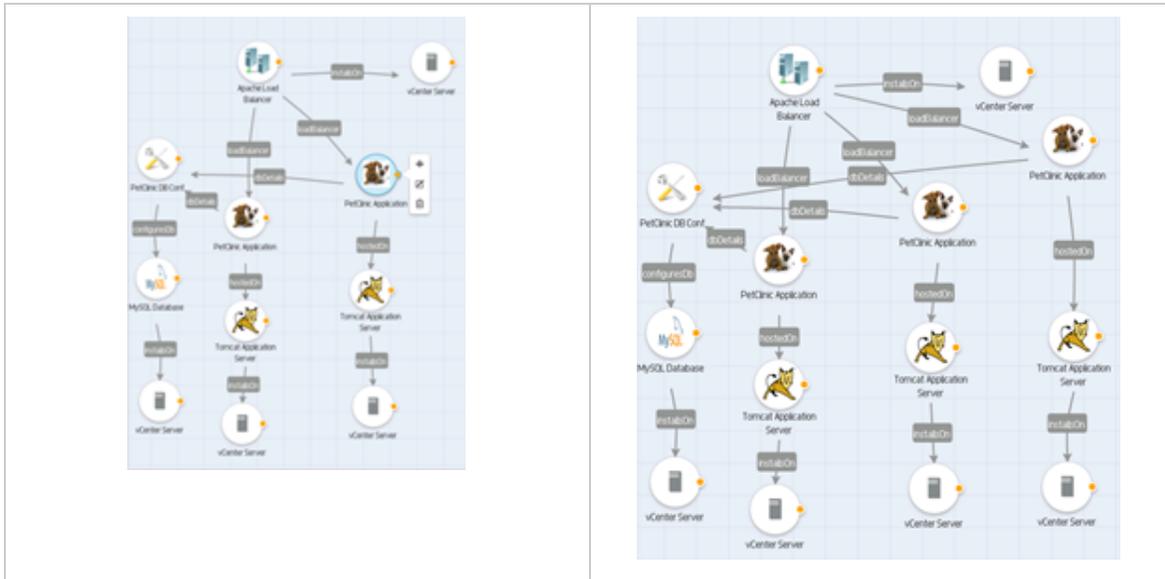
在部署完成之后，您可以扩大部署。扩大时，将复制整个堆栈。

例如，下图显示了作为 `webServerGroup` 逻辑组的 Web 层。



在下图中，此组在开发阶段缩小为一个组，而在测试阶段扩展为两个组。

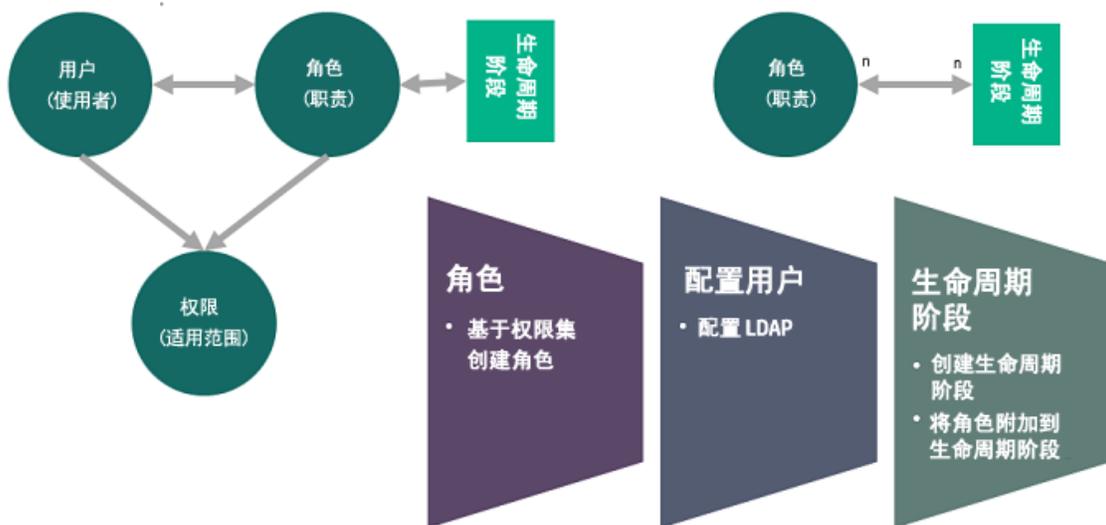
开发阶段	测试阶段
------	------



角色和用户访问

可以为拓扑设计配置用户访问。用户和 **LDAP** 组都可以添加到设计中。应用程序架构师可以创建设计，使设计公开或将其限制为供特定用户使用。

在 **Codar** 中，用户访问包括角色和权限。为 **Codar** 中的每个用户都分配了一个或多个角色。每个角色被分配一个或多个权限。因此，属于特定角色的用户具有为该角色定义的所有权限。**Codar** 包含一些预置角色；但是，用户也可以创建自己的角色，然后将权限分配给他们创建的角色。有关如何创建自定义角色的信息，请参阅“**Codar** 联机帮助”。



管理员和应用程序架构师可以配置用户和组：

- 每个角色的用户在应用程序级别进行定义，以实现粒度级别的控制。
- 组表示可自动为应用程序分配角色的应用程序团队。

下图显示为各种用户 (包括应用程序架构师、开发人员、QA 和发布经理) 配置的设计。



生命周期阶段和操作

生命周期操作构成服务的初始部署，通过流程引擎 (如 HP Operations Orchestration) 与服务提供商进行通信。生命周期操作还提供其他重要功能，如用于根据请求修改服务所需的操作或用于从部署中删除服务所需的操作。

每个生命周期包含一个阶段，每个阶段都具有与其关联的角色。这意味着特定生命周期阶段的角色中包含的所有用户均可在该阶段执行所属角色中定义的操作。例如，如果开发生命周期阶段具有与其关联的应用程序架构师角色，则属于应用程序架构师角色的用户可以在开发生命周期阶段中执行与该角色关联的任务。

以下是 Codar 提供的预置生命周期阶段：

- **开发**:这通常是第一个阶段，将在此阶段开发代码并创建应用程序项目。
- **测试**:将在此阶段根据在开发阶段开发的代码执行测试案例。
- **模拟**:复制生产环境的预生产阶段；用于测试代码和项目。
- **生产**:这通常是最后一个阶段，将在此阶段在实际生产环境中部署应用程序。

除了预置的生命周期阶段，您还可以创建自定义阶段。有关创建、编辑和删除自定义阶段的信息，请参阅 **Codar** 联机帮助。

下表列出了每个生命周期阶段中发生的与包相关的操作：

阶段	提升	部署、重新部署	编辑	删除	拒绝
第一个阶段 (通常是开发阶段)	是	是	是	是	是
中间阶段	是	是	是	是	是
最后一个阶段 (通常是生产阶段)	否	是	是	是	是

您可以使用“发布自动化”>“管道配置”磁贴访问生命周期阶段。

使用以下操作可跨阶段部署或移动包。在未定义发布门户操作时，这些操作描述流。

- **提升**：将包移至下一生命周期阶段。包状态仍为“活动”。
- **部署、重新部署**:部署包。
- **编辑**:更改包的属性。
- **拒绝**:停止将包提升至另一阶段。包将仍处于其当前阶段，其状态将设置为“已拒绝”，

并且操作按钮将不再可用。

- **删除**:删除包。将从系统中永久删除包。

备注: 只有取消并删除所有关联的已部署实例后才能删除包。

- **刷新**:检索当前包状态。

包状态

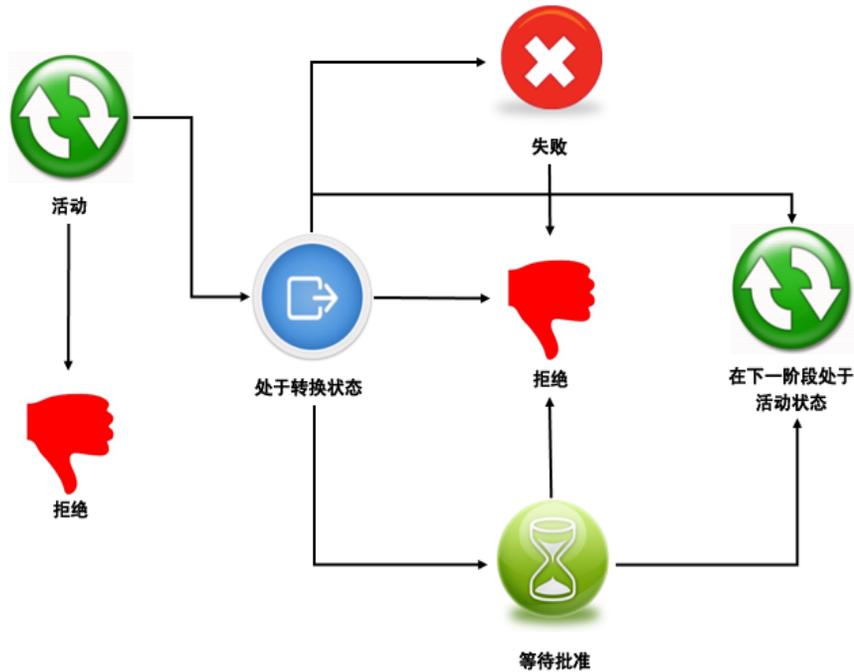
包具有下列状态：

- **活动**:包在当前生命周期阶段中处于活动状态
- **已拒绝**:包已被拒绝，不会移到下一个生命周期阶段
- **转换**:包正在转换到下一个生命周期阶段
- **失败**:包提升失败

如果您拒绝包，则它将仍处于其当前阶段，其状态将设置为“已拒绝”，并且不可应用任何进一步的操作；但是可以将其从系统中删除。

提升包时，包将移至下一阶段并处于“活动”状态。始终在第一个生命周期阶段创建包。如果已配置 **Codar Jenkins** 插件，则在成功生成后，**Jenkins** 插件将连接到 **Codar** 并创建包。

包状态



Hewlett Packard
Enterprise

按生命周期阶段分组服务设计

具有活动包的部分设计需要选择服务设计，以在部署包向导中进行配置。这些服务设计可以针对不同的生命周期阶段进行分组。此分组允许某个生命周期阶段中的包部署仅列出该生命周期阶段中已分组的服务设计。

要对某个生命周期阶段的服务设计进行分组，请在每个拓扑设计中创建以生命周期阶段为名称的标记。当创建生命周期阶段时，会自动创建一个具有该生命周期阶段的标记。因此，只有设计需要与正确的生命周期阶段标记关联。例如，您可以创建一个“开发”标记，并将其与处于开发生命周期状态的所有所需设计关联。

注：“测试”选项卡中的测试运行向导将列出所有设计，但不按标记分组。

发布门户操作

发布门户操作是用户定义的操作，在两个生命周期阶段之间充当提升请求检查。仅当某个包通过了每个启用的操作，且生命周期阶段中所有操作的状态均为成功时，该包才提升到下一个阶段。

发布门户操作分为四种类型：

- 部署操作

此操作根据预定义 **Operations Orchestration (OO)** 内容包来部署部分或完整的应用程序设计。可以配置部署操作，以便向启动提升请求的用户发送电子邮件，通知他们提升是成功还是失败。如果部署操作无法执行且包尚未提升，则用户甚至可以选择拒绝包并清理部署。

有关创建、编辑和删除部署操作的详细信息，请参阅“**Codar 联机帮助**”。

- 自定义操作

此操作在 **Operations Orchestration** 中执行指定流。通常，此操作可以用于执行包含或不包含部署实例的特定测试。

可以配置自定义操作，以便向启动提升请求的用户发送电子邮件，通知他们提升是成功还是失败。如果自定义操作无法执行，则用户可以选择拒绝包或继续进行包提升。

有关创建、编辑和删除自定义操作的详细信息，请参阅“**Codar 联机帮助**”。

- 批准操作

仅当指定的批准人手动批准或拒绝包提升时，此操作才提升包。也可以将批准操作配置为自动批准或拒绝包提升。

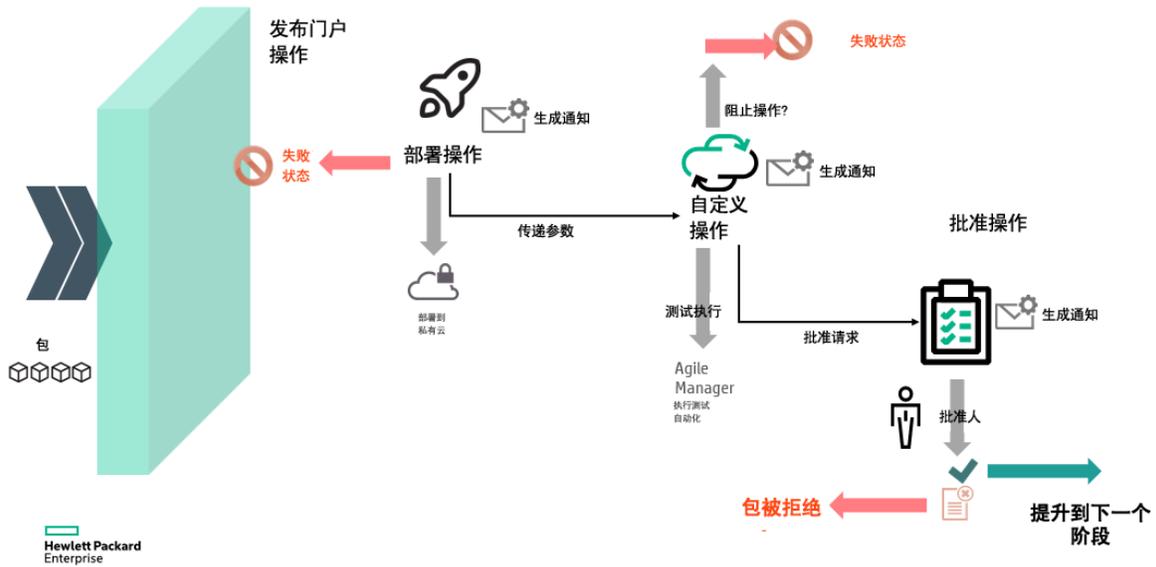
有关创建、编辑和删除批准操作的详细信息，请参阅“**Codar 联机帮助**”。

- 测试集操作

通常，此操作可以用于在 **Application Lifecycle Management** 中执行包含或不包含部署实例的特定测试。执行测试集操作时也可以添加 **UAT** 环境。

有关创建、编辑和删除测试集操作的详细信息，请参阅“**Codar 联机帮助**”。

下图演示了发布门户操作的工作方式：

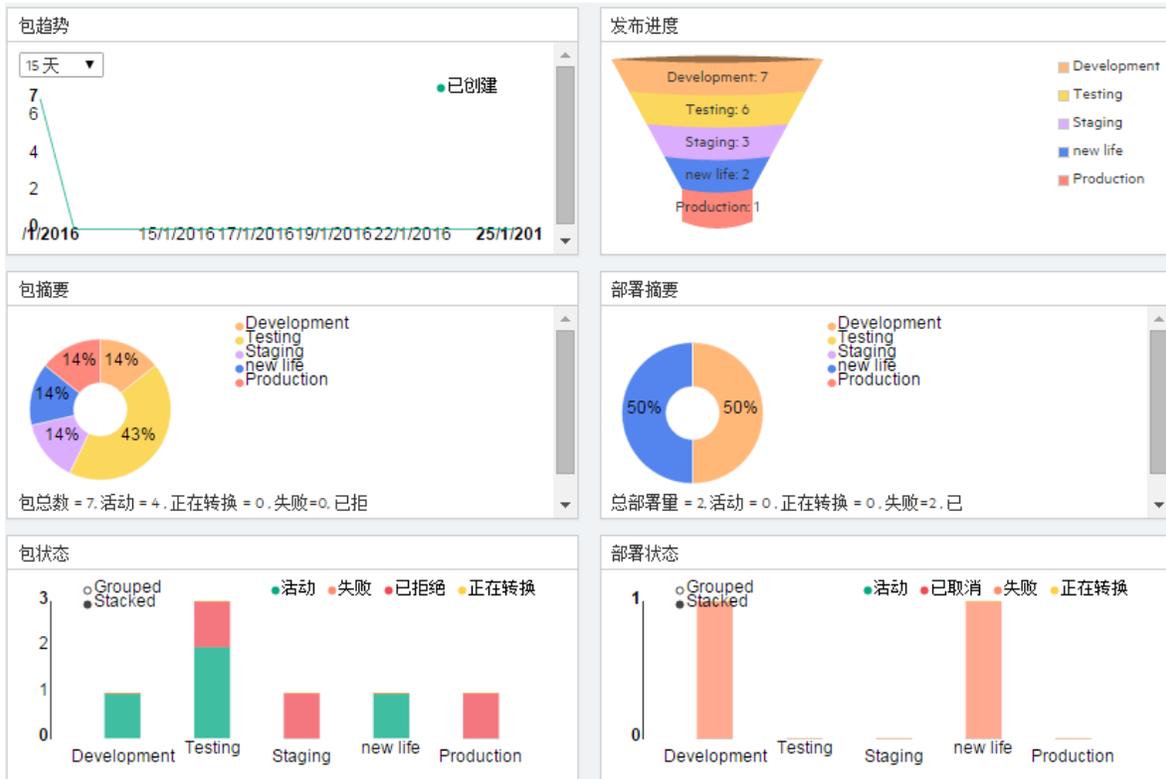


管道统计信息

“管道统计信息”选项卡显示包的详细信息，并包含包摘要、趋势、状态、部署状态等的图形表示。它提供所有包和部署的整体视图，以便您就包部署作出明智的决策。

它显示有关在任何日期创建的包数、成功转换数、部署数等方面的信息。

有关管道统计信息的详细信息，请参阅“Codar 联机帮助”。



环境

您可以在应用程序级别为每个生命周期阶段选择不同的环境。例如，您可以为部署配置 vCenter，但是为暂存配置公有云环境。



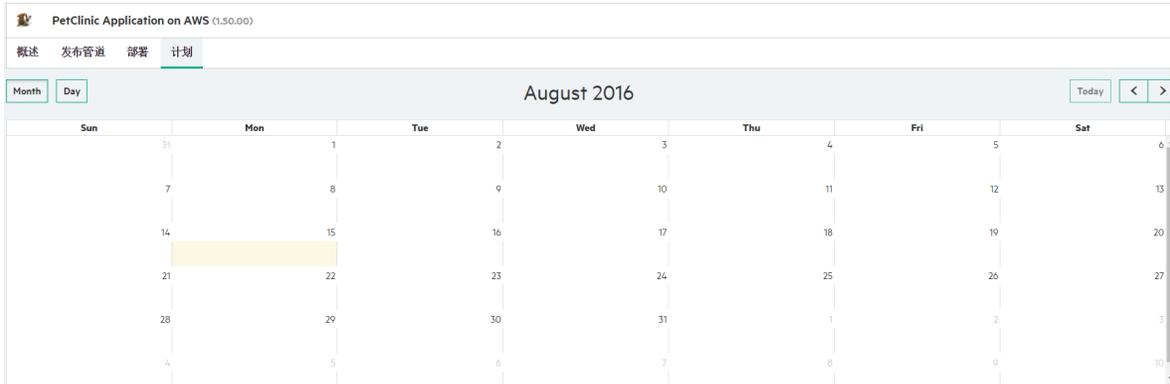
计划发布

您可以跨发布管道计划和管理包的转换。您可以设置以下计划：

- 自动将包从一个生命周期阶段提升到另一个阶段
- 定期自动清除包和部署

可以在日历视图中查看计划，以便跨生命周期阶段查看计划的包提升。

包06 ⚙️					
概述 配置 部署 <u>计划</u> 发布门户					
计划 管理计划					
升级阶段	计划名称	计划日期	计划方	描述	状态
Development	Add Schedule01	08/16/2016 02:22 下午	admin		待审

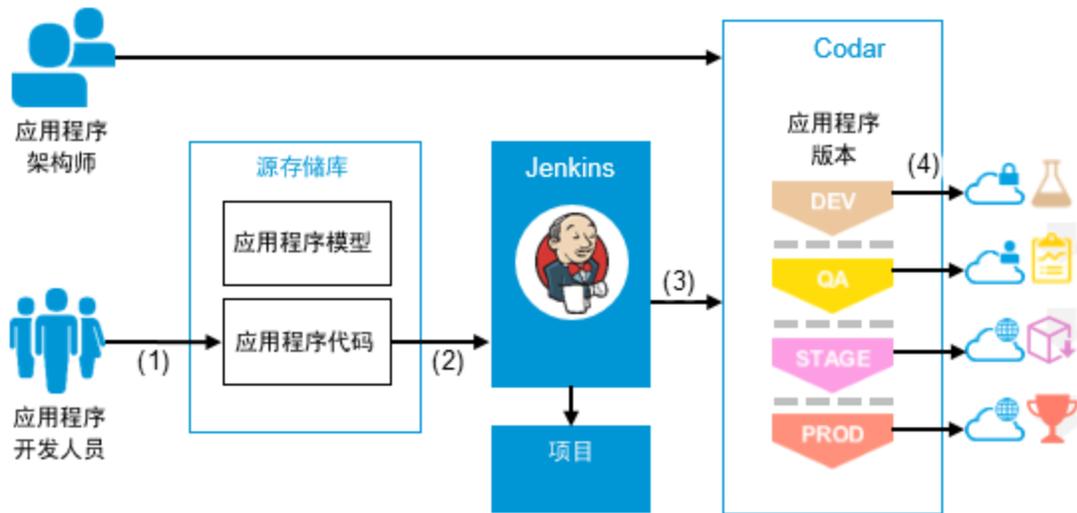


外部集成

Codar 是开放且可扩展的，并且可与 Jenkins、Hudson 等其他构建系统集成。一组全面的 REST API 可与其他外部工具一起用于完成持续的集成、部署和交付。Codar 架构还为您提供了挂钩到开发测试和开发运营的自定义流的选项。

Jenkins 集成

Codar 包括用于连续部署的 Jenkins 插件。以下插图显示了它的工作方式。

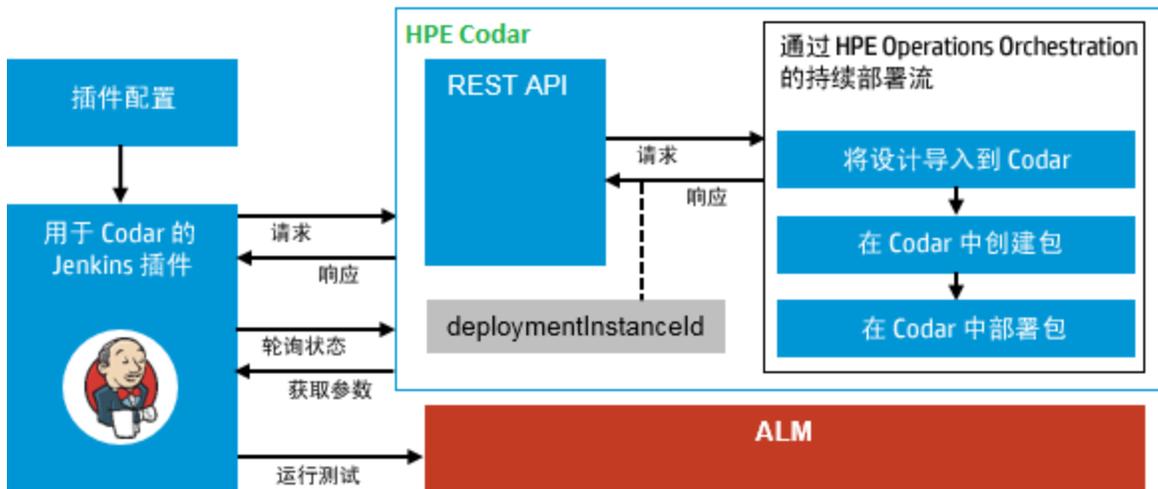


1. 开发人员签入更改。
2. 持续集成触发构建。
3. Jenkins 插件创建并部署包。
4. 根据生命周期阶段，应用程序将部署到不同的环境中。
5. 在连续提升时，如果成功执行所有发布门户操作，则将包移到最后一个生命周期阶段。

备注： Bamboo 和 TFS 也可提供类似的 OOTB 集成。

ALM 集成

您可以将 Application Lifecycle Management (ALM) 与 Codar 集成，以在成功部署之后运行测试。以下插图显示了 Jenkins 如何充当协调者。

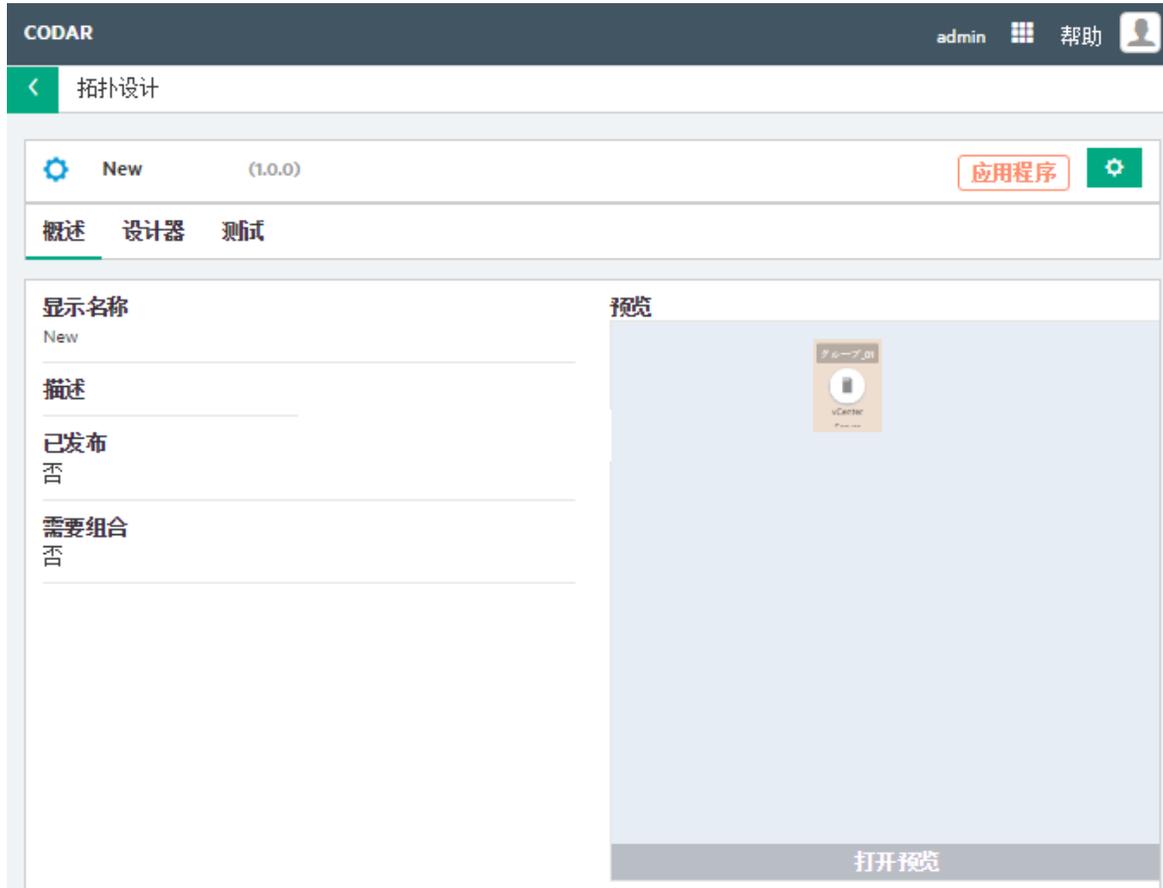


基础结构即代码 (IaaS)

管理基础结构即代码 (IaaS) 允许 IT 团队利用最佳实践开发代码，例如针对如何配置基础结构和应用程序的代码审核和单元测试。

Codar 可以管理基础结构即代码。拓扑设计可包含服务器配置、网络、卷、关系和诸如应用程序版本和包信息的应用程序特定信息。拓扑设计可按 JSON 格式导出并在源代码管理

系统中使用应用程序进行管理。开发人员可使用文本编辑器更改模型并使用模型实现自动化。修改后的模型还可导入回 Codar。



在 Docker 群集上部署容器

Docker Universal Control Plane 提供适用于 Docker 应用程序的本机或虚拟私有云 (VPC) 容器管理解决方案。

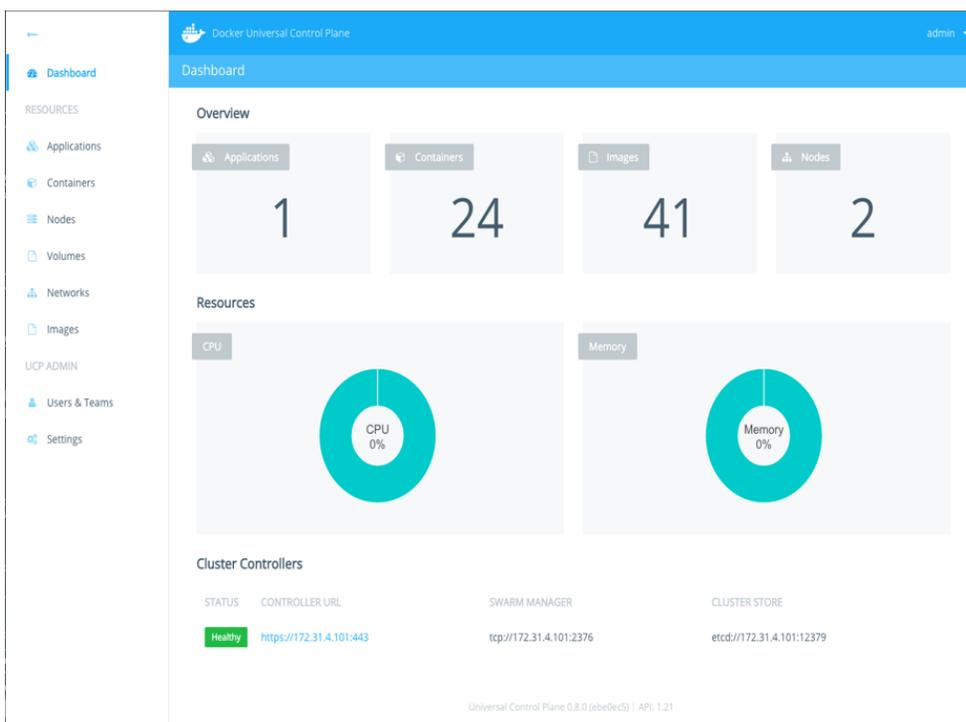
您可以使用 Docker UCP 执行以下操作。

- 将 Docker UCP 配置为 Codar 中的资源提供程序。
- 创建基于 Docker 内容的设计。
- 在 Docker UCP 上部署容器并执行管道管理。

Docker Universal Control Plane

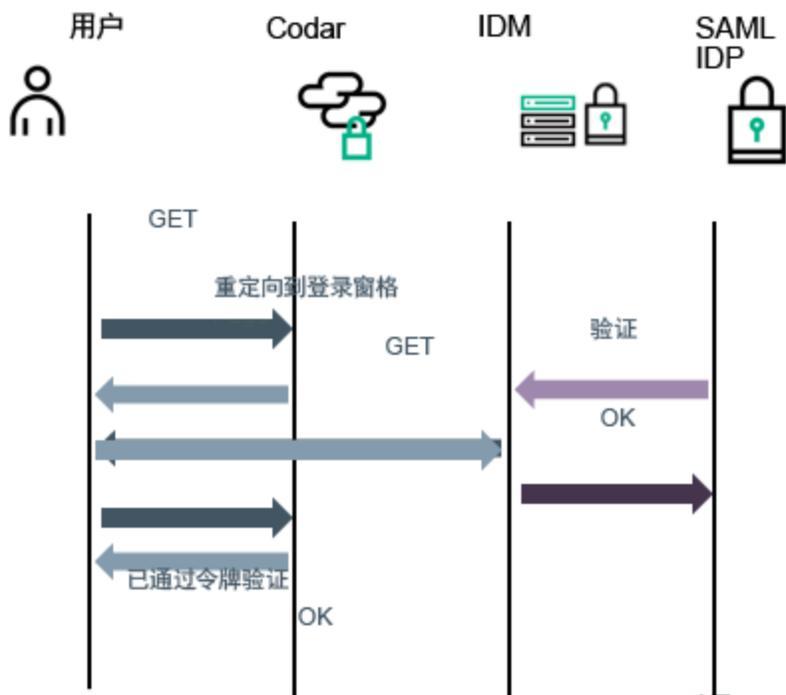


Docker UCP 面板



Codar 支持 SAML

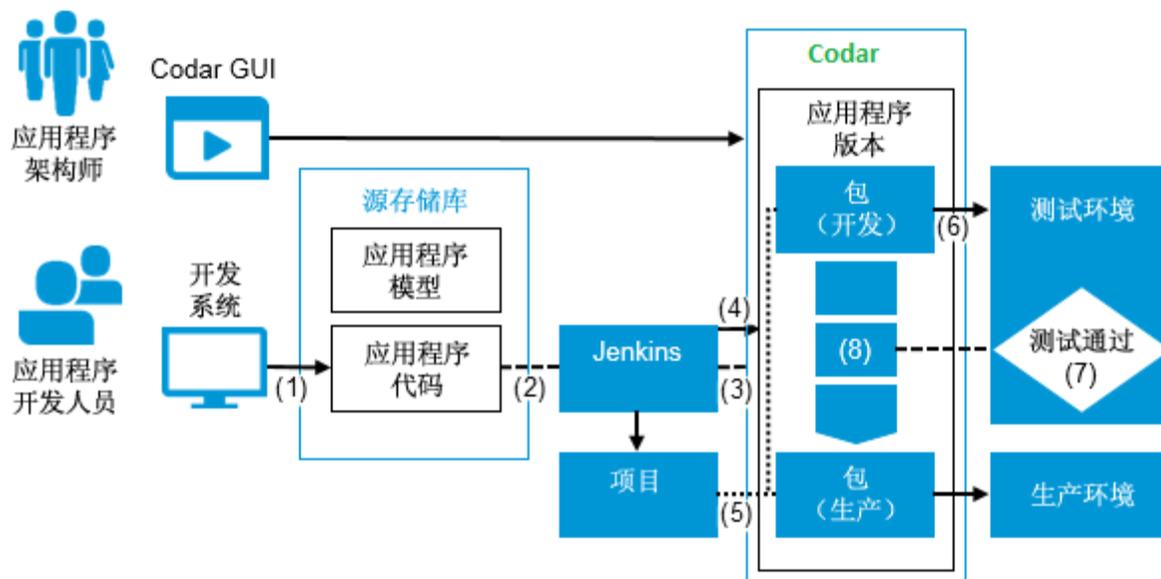
Codar 提供基于 SAML 的 SSO。只需进行一次身份验证 (使用支持 SAML 的标识提供程序), 后续使用 Cloud Service Automation 时无需再次键入密码。



用例:持续集成、部署和交付

目的是让应用程序实现持续集成 (CI) 和连续部署。应用程序开发人员为应用程序编码，而应用程序架构师在 **Codar** 界面中对应用程序建模，然后导出应用程序模型即代码 (IaC)。当应用程序开发人员签入代码后，将触发 **Jenkins** 构建并且将在特定环境中使用应用程序模型部署应用程序。在部署应用程序之后，连续部署过程将扩展为持续交付，而特定于应用程序的测试可在已部署的实例上自动运行，根据测试结果，应用程序可能已部署到其他环境。

以下部分说明 **Codar** 如何实现此情景。



应用程序建模

应用程序架构师以图形方式对应用程序建模，方法是在设计器界面中包括必需的设计组件并通过关系连接这些组件。**Codar** 包含标准组件面板，这些组件可通过各种部署引擎 (例如 **HPE Operations Orchestration** 和 **Chef**) 导入 (使用)。这种设计 (称作应用程序模型) 是应用程序部署方法的表示形式。应用程序模型可采用 **JSON** 格式导出并在外部源存储库中进行管理，从而实现基础结构即代码 (IaC)。

持续集成和部署

在持续集成中，源存储库中提供了示例应用程序的代码和该应用程序的部署模型(JSON 格式)。

如果应用程序开发人员对此应用程序进行了代码更改并将更改签入源存储库 (1)，则 **Jenkins** 将触发构建 (2)。

Codar 提供了一个 **Jenkins** 插件，该插件具有 **Codar** 的 IP 地址、用户名和密码等详细信息。该插件将在构建后步骤 (3) 中建立连接并调用 API。然后，该 API 调用执行各种操作的工作流，以实现连续部署和持续交付。

导入应用程序设计

如果应用程序模型尚未导入 **Codar** 或者此模型已更改，则连续部署工作流会使用 JSON 格式 (IaaS) 将其作为新版本的应用程序设计导入 **Codar** (4) 中。这允许在部署时考虑应用程序开发人员和架构师进行的更改。

如果应用程序模型已导入或者应用程序设计中无更改，则不会发生此导入操作并且 **Codar** 中的应用程序版本仍保留不变。您可在设计器的“拓扑/序列化”磁贴中查看应用程序模型。

在环境中部署

在创建包之后，连续部署工作流将根据环境 (6) 完成应用程序设计。可以在“部署”选项卡上查看包的部署：



runbook 自动引擎将创建基于完成基础结构层、平台层和应用程序层的设计的执行计划。您可监控特定包的部署状态并查看已部署应用程序的图形表示形式，其中包含组件级别的属性和操作。

发布设计

发布设计可将设计作为面向服务用户的产品。必须先安装 CSA 许可证，然后才能发布设计。

“生产”阶段中具有活动包的完整设计包含特定于包的属性作为设计的一部分，并且可对其进行发布。

“生产”阶段中具有活动包的部分设计包含特定于包的属性作为设计的一部分，并且在通过部署“生产”阶段包创建最终组合设计后才能对其进行发布。

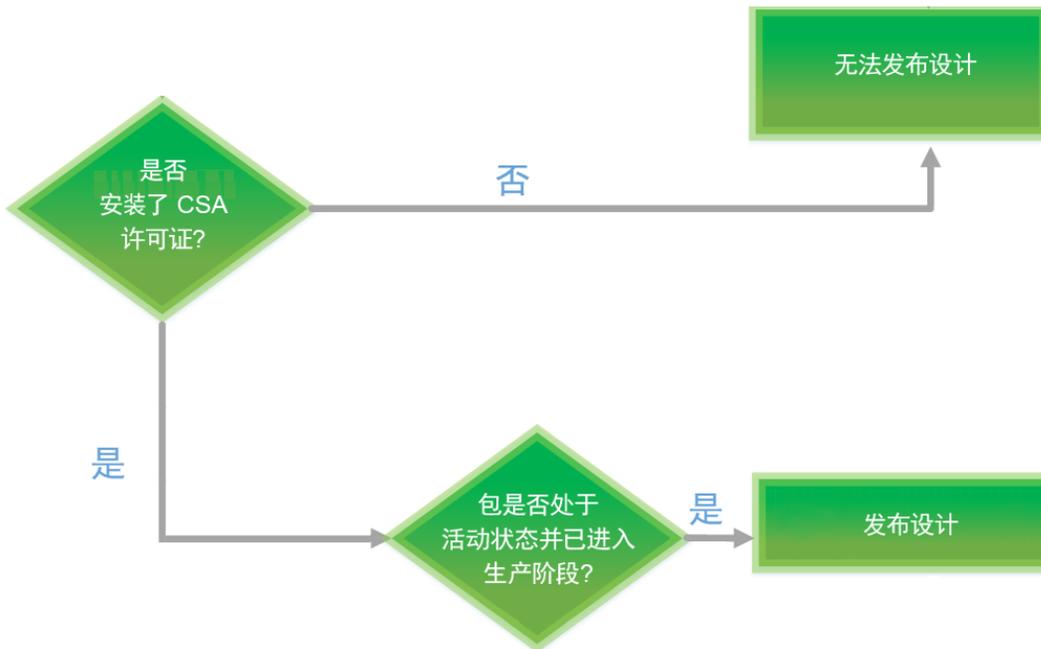
根据您安装的许可证，发布部分设计会有所不同：

- 在生产基础结构上部署已提升至“生产”阶段的 Codar 应用程序设计，然后在成功的生产部署中将组合生产设计设为可见。然后可以向服务用户发布该设计。
- 不属于 Codar 应用程序设计的设计必须在“测试”选项卡中保存为组合设计。然后可以向服务用户发布该设计。

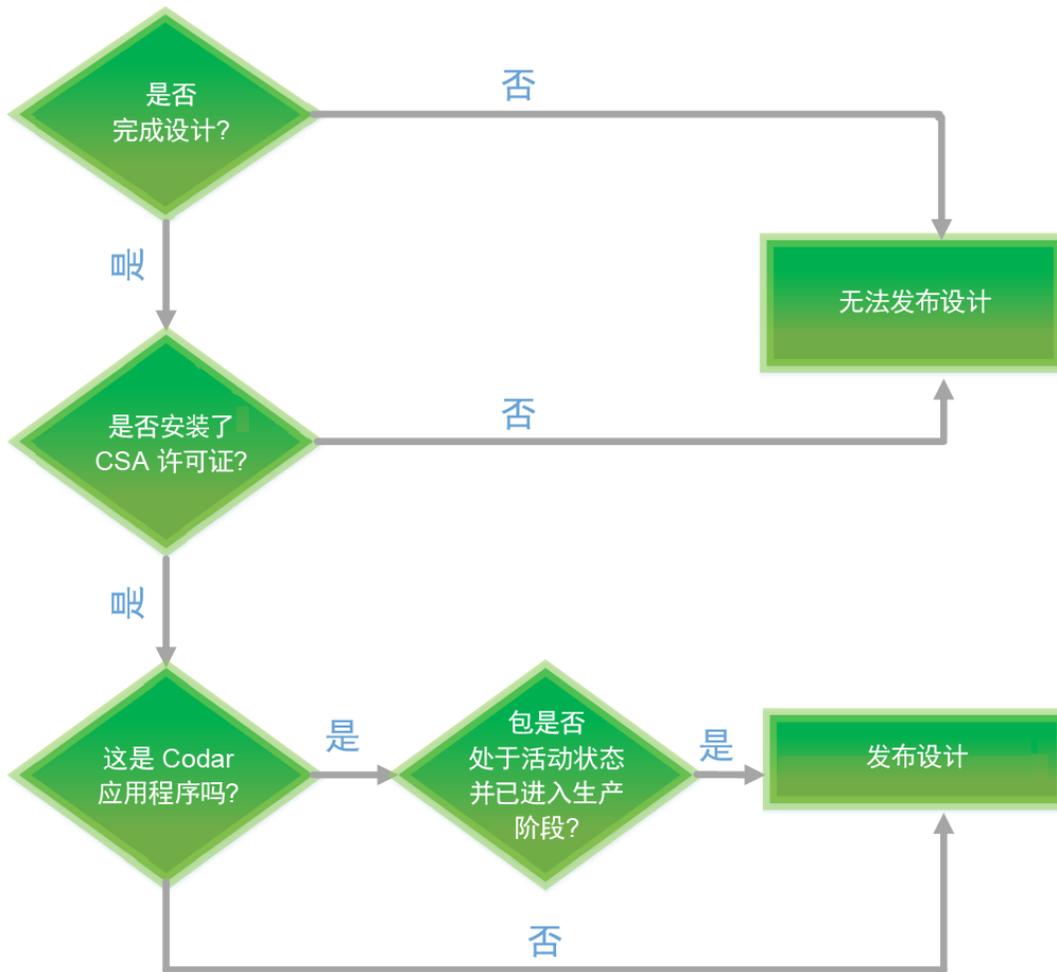
备注：仅当序列化设计没有包或者包处于管道管理的最后一个阶段并具有活动的部署实例时，才能发布序列化设计。

如果已为拓扑设计添加应用程序标记，除非删除该标记，否则无法发布设计。

下图阐释了基于所使用的许可证可在何时发布序列化设计：



下图阐释了基于所使用的许可证可在何时发布拓扑设计：



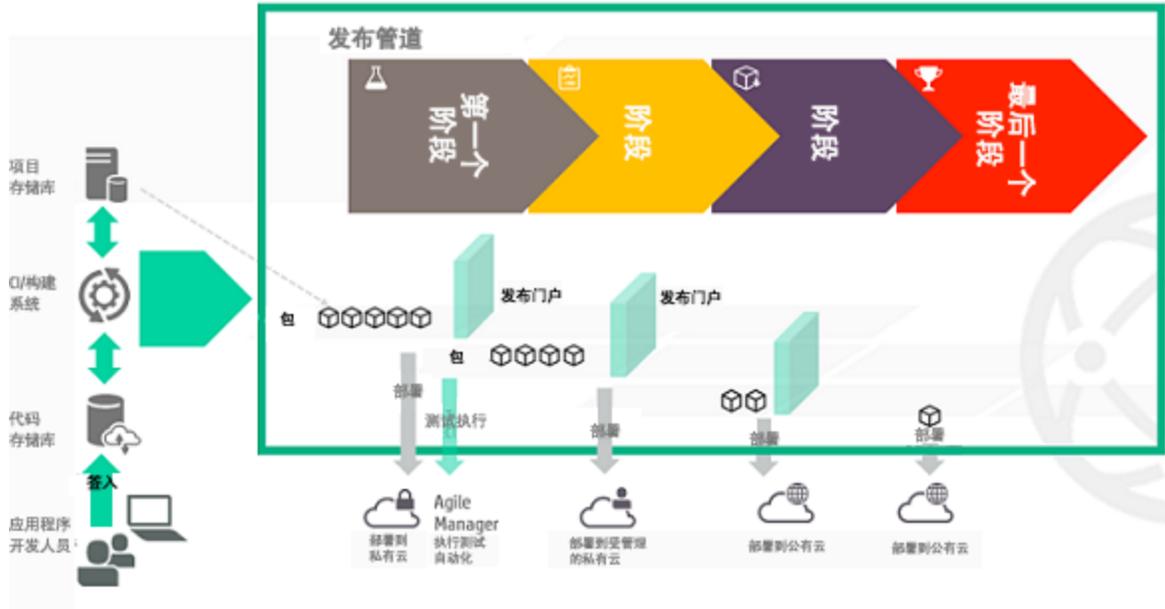
用例:可自定义的发布管道

目的是允许用户构建自己的自定义发布管道，并且让每个应用程序团队定义和使用单独的生命周期阶段。以下是用户定义自己的发布管道的高级步骤。有关如何执行这些步骤的信息，请参阅“Codar 联机帮助”。

1. 创建角色并添加权限
2. 创建生命周期阶段，并将角色与每个生命周期阶段关联
3. 将生命周期阶段添加到应用程序设计中
4. 将发布门户操作添加到每个生命周期阶段中
5. 使用连续提升 API 创建包

如果所有发布门户执行成功，则将包移到最后一个生命周期阶段。

以下插图描述了可自定义的发布管道。



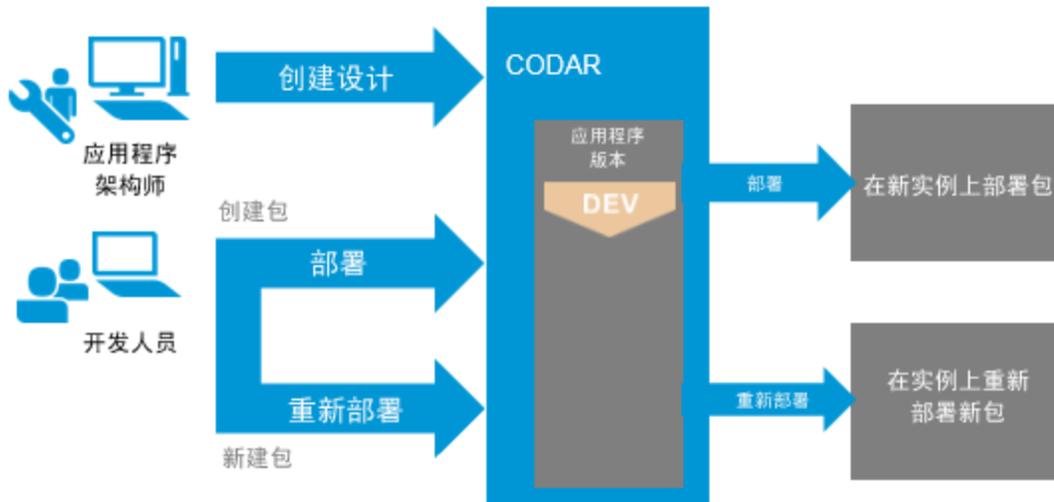
用例:部署和重新部署包

目的是部署包，然后使用同一实例在具有旧版本的实例上重新部署包的较新版本。

应用程序架构师将对应用程序建模，并标记它以进行管道管理。然后，开发人员会创建一个包并进行部署。在部署包时，会创建一个新的实例。部署基于拓扑设计。

在部署之后，可以使用同一实例重新部署包的较新版本。可以将实例升级或修补到较新的包或内部版本。

下图显示了该过程：



用例:部署和扩大

目的是创建可扩展的堆栈，并在部署之后根据需要扩大该堆栈。应用程序架构师将对应用程序建模，并标记该应用程序以进行管道管理。架构师确定应在不同生命周期阶段中扩大的组件。可扩展的堆栈可能包含单个组件或一组组件。在开发期间，可以将堆栈缩小为一个堆栈，然后在测试期间将该堆栈扩展为两个堆栈，等等。

后续步骤

《Codar Installation Guide》和《Codar Configuration Guide》介绍了如何下载、安装和配置软件，《Codar API and CLI Reference》简要介绍了 REST API，并介绍了如何获得每个 API 的详细信息。它还介绍了命令行界面。您可以从应用程序中访问联机帮助，获取与任务相关的帮助。

Codar 社区版

具备完整功能的 **Codar 30 天试用版** 以 **Docker** 镜像的形式提供。

社区版试用

要使用 **Codar 社区版**，需要注册并激活产品。

完成注册后，用户将会收到 **30 天试用版许可证** 的产品激活代码。可以通过电子邮件中的链接或者使用激活代码在线激活产品。

发送文档反馈

如果对本文档有任何意见，可以通过电子邮件[与文档组联系](#)。如果在此系统上配置了电子邮件客户端，则在单击以上链接时将会打开一个电子邮件窗口，其主题行中显示以下信息：

有关《概念指南》的反馈 (Codar 1.70)

只需在电子邮件中添加反馈并单击“发送”即可。

如果没有可用的电子邮件客户端，请将以上信息复制到 **Web** 邮件客户端的新邮件中，然后将反馈发送到 clouddocs@hpe.com。

感谢您提供反馈！