

# HP Codar

ソフトウェアバージョン: 1.50

## コンセプトガイド

ドキュメントリリース日: 2015年7月  
ソフトウェアリリース日: 2015年7月



## ご注意

## 保証

HP製品、またはサービスの保証は、当該製品、およびサービスに付随する明示的な保証文によってのみ規定されるものとします。ここでの記載は、追加保証を提供するものではありません。ここに含まれる技術的、編集上の誤り、または欠如について、HPはいかなる責任も負いません。

ここに記載する情報は、予告なしに変更されることがあります。

## 権利の制限

機密性のあるコンピューターソフトウェアです。これらを所有、使用、または複製するには、HPからの有効な使用許諾が必要です。商用コンピューターソフトウェア、コンピューターソフトウェアに関する文書類、および商用アイテムの技術データは、FAR 12.211および12.212の規定に従い、ベンダーの標準商用ライセンスに基づいて米国政府に使用許諾が付与されます。

## 著作権について

© Copyright 2014 - 2015 Hewlett-Packard Development Company, L.P.

## 商標について

Adobe® は、Adobe Systems Incorporated (アドビシステムズ社) の登録商標です。

Microsoft® およびWindows® は、米国におけるMicrosoft Corporationの登録商標です。

UNIX® は、The Open Groupの登録商標です。

## ドキュメントの更新情報

このマニュアルの表紙には、以下の識別情報が記載されています。

- ソフトウェアバージョンの番号は、ソフトウェアのバージョンを示します。
- ドキュメントリリース日は、ドキュメントが更新されるたびに変更されます。
- ソフトウェアリリース日は、このバージョンのソフトウェアのリリース期日を表します。

更新状況、およびご使用のドキュメントが最新版かどうかは、次のサイトで確認できます。<http://h20230.www2.hp.com/selfsolve/manuals>

このサイトを利用するには、HP Passportへの登録とサインインが必要です。HP Passport IDの登録は、次のWebサイトから行なうことができます。

<http://h20229.www2.hp.com/passport-registration.html>

または、HP Passportのログインページの **[New users - please register]** リンクをクリックします。

適切な製品サポートサービスをお申し込みいただいたお客様は、更新版または最新版をご入手いただけます。詳細は、HPの営業担当にお問い合わせください。

# サポート

HPソフトウェアサポートオンラインWebサイトを参照してください。 <http://www.hp.com/go/hpssoftwaresupport>

このサイトでは、HPのお客様窓口のほか、HPソフトウェアが提供する製品、サービス、およびサポートに関する詳細情報をご覧ください。

HPソフトウェアオンラインではセルフソルブ機能を提供しています。お客様のビジネスを管理するのに必要な対話型の技術サポートツールに、素早く効率的にアクセスできます。HPソフトウェアサポートのWebサイトでは、次のようなことができます。

- 関心のあるナレッジドキュメントの検索
- サポートケースの登録とエンハンスメント要求のトラッキング
- ソフトウェアパッチのダウンロード
- サポート契約の管理
- HPサポート窓口の検索
- 利用可能なサービスに関する情報の閲覧
- 他のソフトウェアカスタマーとの意見交換
- ソフトウェアトレーニングの検索と登録

一部のサポートを除き、サポートのご利用には、HP Passportユーザーとしてご登録の上、サインインしていただく必要があります。また、多くのサポートのご利用には、サポート契約が必要です。HP Passport IDを登録するには、次のWebサイトにアクセスしてください。

<http://h20229.www2.hp.com/passport-registration.html>

アクセスレベルの詳細については、次のWebサイトをご覧ください。

[http://h20230.www2.hp.com/new\\_access\\_levels.jsp](http://h20230.www2.hp.com/new_access_levels.jsp)

HP Software Solutions Now は、HPSWのソリューションと統合に関するポータルWebサイトです。このサイトでは、お客様のビジネスニーズを満たすHP製品ソリューションを検索したり、HP製品間の統合に関する詳細なリストやITILプロセスのリストを閲覧することができます。このサイトのURLは <http://h20230.www2.hp.com/sc/solutions/index.jsp> です。

# 目次

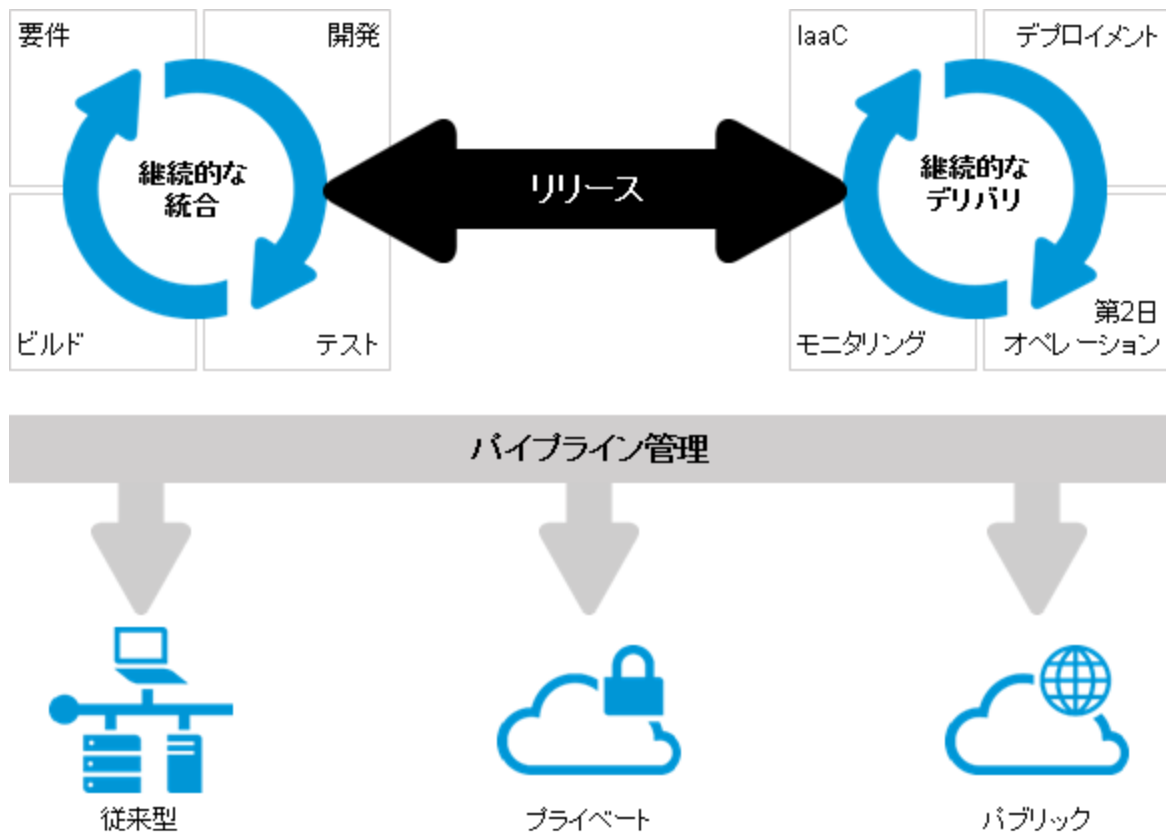
HP Codar .....	6
HP Codarの概要 .....	7
宣言ベースのモデリング .....	7
トポロジ構成 .....	8
マイクロサービス .....	9
アプリケーションのパイプラインの管理 .....	10
パッケージの操作 .....	11
デプロイと再デプロイ .....	12
スケールアウト .....	13
ユーザーアクセス .....	15
環境 .....	16
外部統合 .....	17
Jenkinsの統合 .....	17
ALMの統合 .....	18
Infrastructure as code (IaC) .....	18
ユースケース: 継続的な統合、デプロイメント、デリバリ .....	20
アプリケーションのモデル化 .....	20
継続的な統合とデプロイメント .....	21
アプリケーションデザインのインポート .....	21
パッケージの管理 .....	21
環境へのデプロイ .....	22
デザインの発行 .....	23
ユースケース: パッケージのデプロイと再デプロイ .....	24
ユースケース: デプロイメントとスケールアウト .....	25
ライフサイクルのステージおよびアクション .....	26
パッケージの状態 .....	26
ライフサイクルステージによるサービスデザインのグループ化 .....	27
次のステップ .....	28

ドキュメントのフィードバックの送信 ..... 29

## HP Codar

組織が継続的統合を継続的デリバリーに拡張する際は、新しい課題に直面します。その課題には、開発環境から運用環境に、それぞれの環境の違いを考慮しながら一貫性を持ってアプリケーションをデプロイすることがあります。

DevOpsは、コラボレーション、自動化、ガバナンスに関する一連の方針、方法、手法を使用して、開発 (Dev) 環境と運用 (Ops) 環境の溝を埋めるためのフレームワークを提供します。この目的は、ビルドまたはアセンブリの継続的な統合を、異種環境間で、再現性があり一貫性のあるアプリケーションデプロイメントに広げることです。次の図は、DevOps環境での継続的統合と継続的デリバリーのサイクルを示しています。



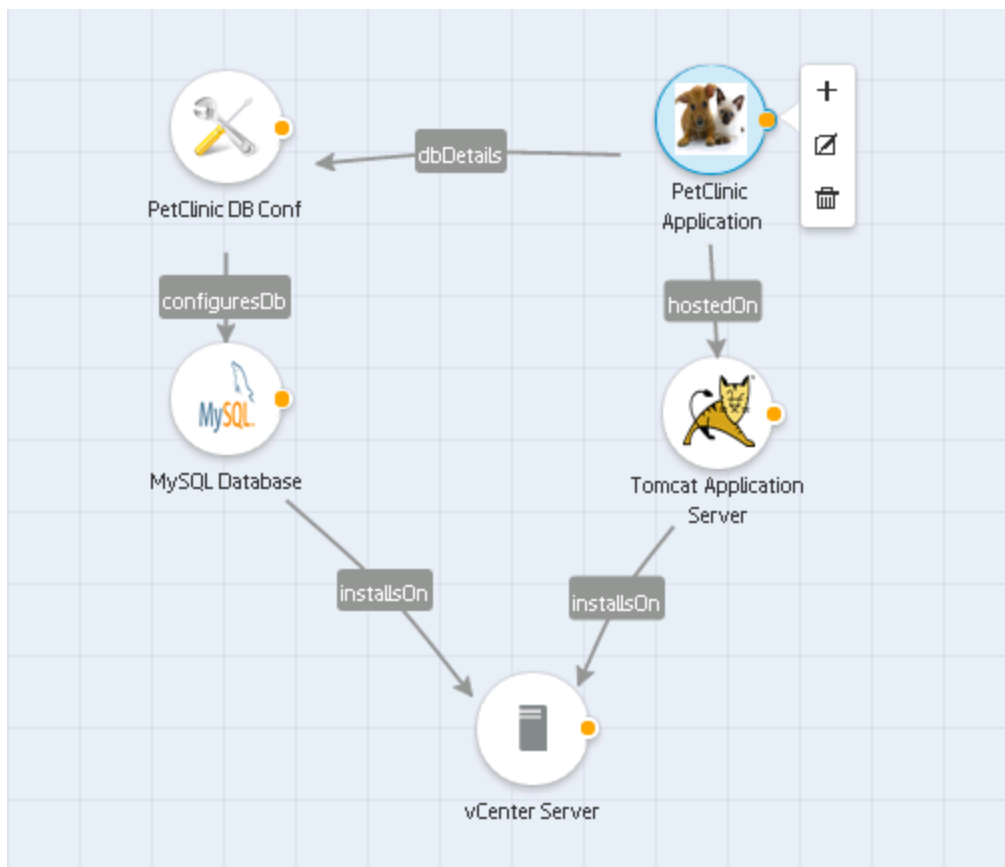
## HP Codarの概要

HP Codarは、システムへのあらゆる変更がリリース可能で、すべてのコード変更を運用環境にデプロイ可能な継続的デリバリーを促進します。HP Codarは、継続的デリバリーの自動化を実現します。その結果、コード変更があると、ビルドが開始されデプロイされます。自動化されたユニットテストも実行されます。アプリケーションは、ランブック自動化フローで定義されたポリシーに基づいて、環境に自動的にデプロイされます。継続的デリバリーの目的は、デリバリーを頻繁に行い、ユーザーからのフィードバックを迅速に得ることです。

## 宣言ベースのモデリング

宣言ベースのモデリングを使用してアプリケーション開発を自動化する場合、ユーザーはアプリケーション開発の終了状態(アプリケーションのコンポーネントと、コンポーネント間の依存性)を宣言できます(その状態に至るプロセスは、バックグラウンドで開始されます)。こうして、ユーザーは、どのようにデプロイするかではなく、何をデプロイするかに集中できます。その結果、マルチティアアプリケーションのデプロイメントを短時間で自動化できます。長期にわたる管理も大幅に簡素化されます。

HP Codarは、ユーザーインターフェイスを使って複雑なデザインの作成、統合、管理を行う宣言ベースのモデル開発をサポートしています。モデルは、トポロジデザインとそのプロパティで構成されます。HP Codarでは、プロパティを具現化しているときに柔軟に変更できます(レイトバインディングと同様です)。



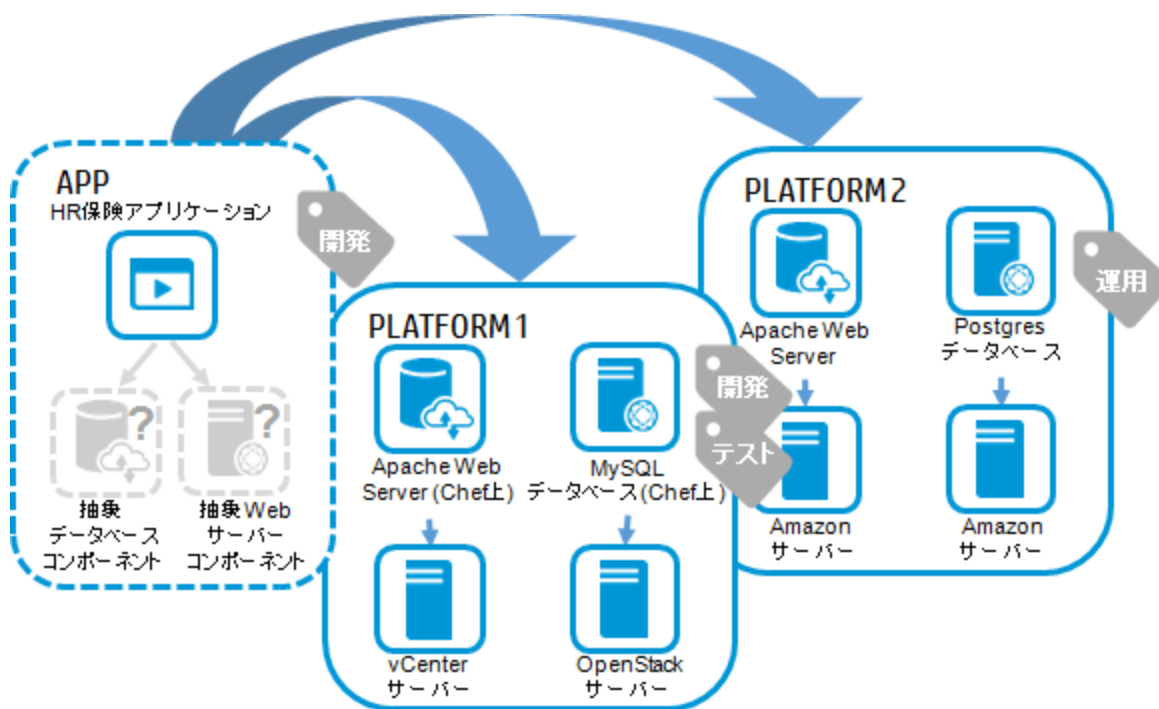
## トポロジ構成

トポロジ構成は、実行時に、インフラストラクチャーデザインを使ってアプリケーションデザインを構成する際に使用します。アプリケーションデプロイメントでは、デプロイメントごとにインフラストラクチャーのニーズは異なります。このように、アプリケーションデザインで必要とされるさまざまなインフラストラクチャーを定義する作業ではトポロジ構成が役立ち、デプロイメント時にさまざまなインフラストラクチャーデザインを使った構成が可能になります。

コンポーネントの記述には、機能と特性が使用されます。アプリケーションデザインは、デザインで機能コンポーネントと特性を使用して、要件を定義します。アプリケーションデザインはそれだけではプロビジョニングできず、互換性のあるインフラストラクチャーデザインを選択する必要があります。インフラストラクチャーデザインコンポーネントでは、機能および特性に基づいて互換性チェックが実行され、適合するデザインが互換インフラストラクチャーデザインとしてデプロイメント時に選択されます。

次の図は、HRインスタンスアプリケーションのトポロジ構成を示しています。このアプリケーションはデータベースコンポーネントとWebサーバーコンポーネントを必要とします。これは、アプリケーションデザインAPPで定義されています。実行にはPLATFORM1が使用されます。ここでは、Webサーバーの機能および特性を持つApache Web Serverと、データベースの機能および特性を持つMySQLデータベースが稼働しています。同様に、PLATFORM2もAPP要件を満たしています。



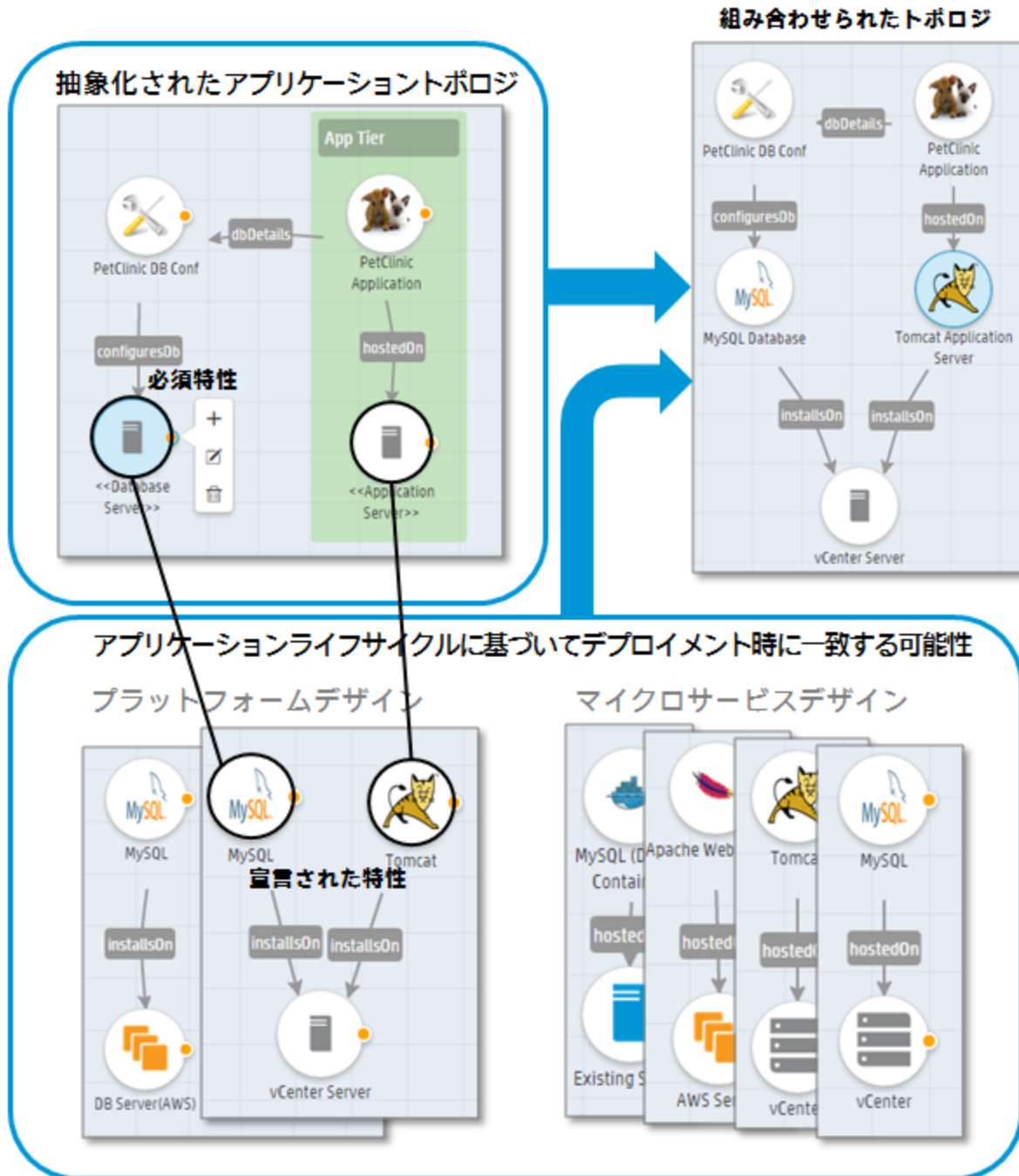


## マイクロサービス

単一のサービスではなくプラットフォームまたはインフラストラクチャーサービスを提供する複数のインフラストラクチャーデザインを使用して部分的なアプリケーションをデプロイできます。部分的なアプリケーションは複数のインフラストラクチャーからのコンポーネントを使用できます。すべての機能と一致する単一のサービスを選択することも、異なる複数のデザインからのコンポーネントを選択することもできます。

たとえば、アプリケーションにデータベースサービスとアプリケーションサービスが必要な場合、単一デザイン内にデータベースとアプリケーションがあるデザインを選択することも、1つのサービスデザインからデータベースを選択し、別のサービスデザインからアプリケーションを選択することもできます。

マイクロサービスの選択に基づいて実行時に組み合わせられたトポロジが作成されます。マイクロサービスはライフサイクルのステージに関連付けることができます。

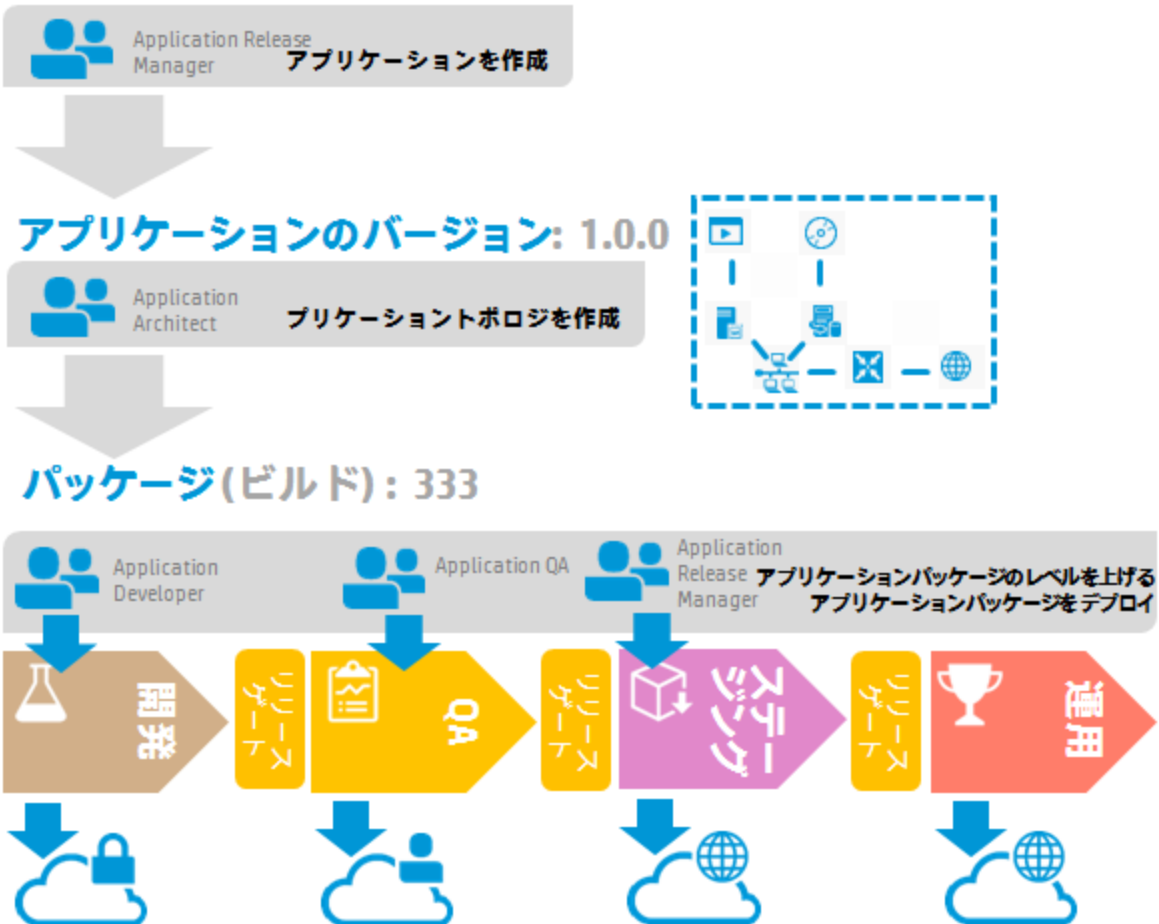


## アプリケーションのパイプラインの管理

アプリケーションのデプロイメントの自動化は、複雑で時間を要するプロセスであり、多くの調査を必要とします。開発と運用でアプリケーションのデプロイ方法が異なると、多くのエラーの原因にな

ります。アプリケーションパイプライン管理を使用すると、前に説明したように、同じトポロジモデルを使用して異なる複数の環境にアプリケーションをデプロイできます。各ステージで異なるマイクロサービスを選択できますが、アプリケーションデザインは変わりません。つまり、異なるライフサイクルステージを通じて同じデザインがデプロイおよびテストされます。

## アプリケーション: Pet Clinic



## パッケージの操作

HP Codarは、DevOps環境を実装するための一元化された構造です。さまざまなロールで、パッケージのデプロイ、再デプロイ、レベルを上げる、拒否を行うことができます。パッケージを、一貫した再現性のある方法で、1つのステージから次のステージへレベルを上げます。このようにして、アプリケーションが運用状態に進むときの可視性が開発者に保証されます。

パッケージを作成してデプロイすると、新しい仮想マシンが作成され、パッケージがデプロイされます。デプロイ済みのインスタンスについてテストを実行し、パッケージのレベルを上げるか拒否することができます。


次のスクリーンショットに示すように、HP Codarはアプリケーションパイプライン管理機能を促進します。

概要    パッケージ

すべてのステージ    すべてのの状態    検索

---


**DEVELOPMENT**

 **build 20**  
説明:  
デプロイ済みインスタンス: 0  
最終更新: 2015/06/18 PM 2:49:46

デプロイ    レベルを上げる

---


**TESTING**

 **build 19**  
説明:  
デプロイ済みインスタンス: 0  
最終更新: 2015/06/18 PM 2:49:32

デプロイ    レベルを上げる    拒否

---


**STAGING**

 **build 18**  
説明:  
デプロイ済みインスタンス: 1  
最終更新: 2015/06/18 PM 2:50:00

デプロイ    レベルを上げる    拒否

---

**PRODUCTION**

 **build 17**  
説明:  
デプロイ済みインスタンス: 1  
最終更新: 2015/06/18 PM 2:50:19

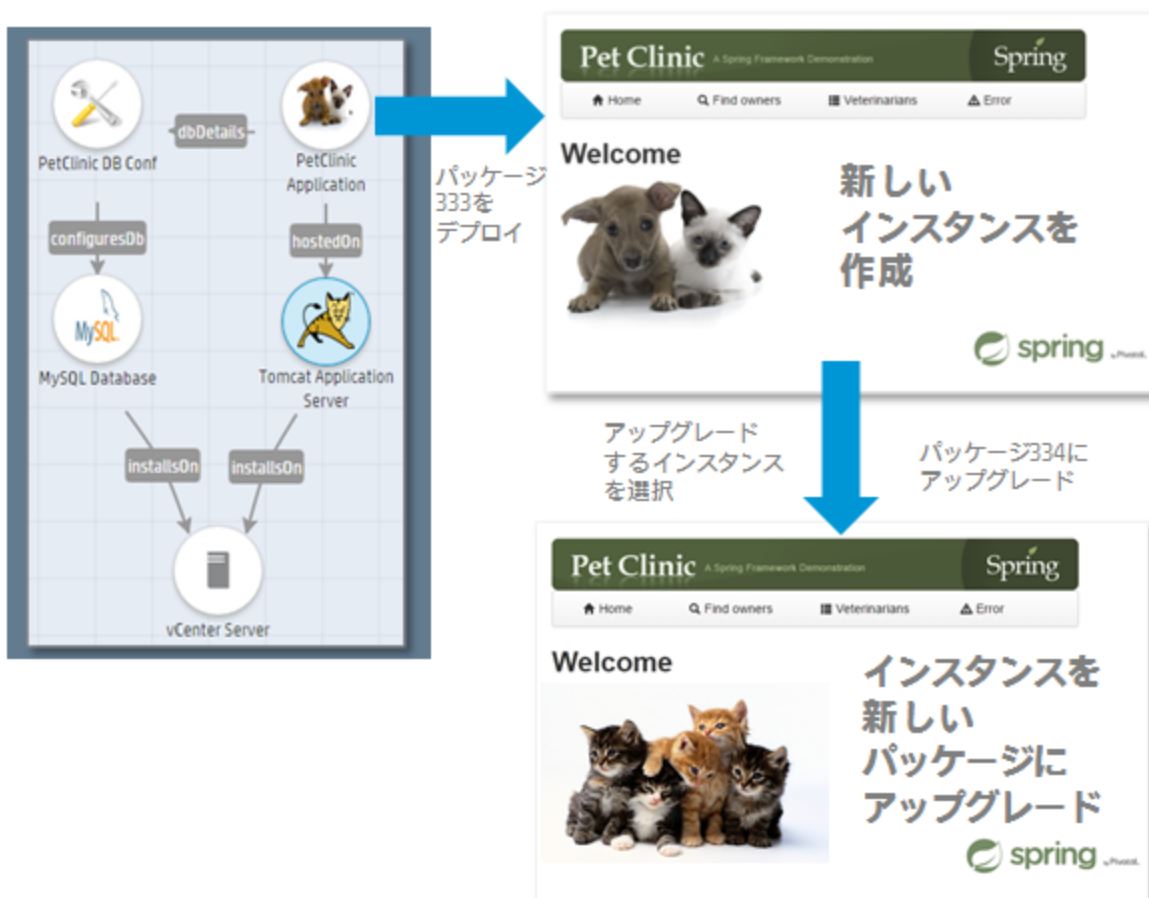
デプロイ    拒否

---

作成    パッケージの削除

## デプロイと再デプロイ

古いパッケージがデプロイされているインスタンス上にパッケージを再デプロイすることができます。インスタンスの詳細を表示し、既存の変わらない要求を選択できます。再デプロイを使用して、コンポーネントをアップグレードしたりパッチを適用したりすることもできます。再デプロイにより、すべてのコンポーネントの変更アクションが開始されるので、デザイン内のすべてのコンポーネントを新しいバージョンにアップグレードすることができます。

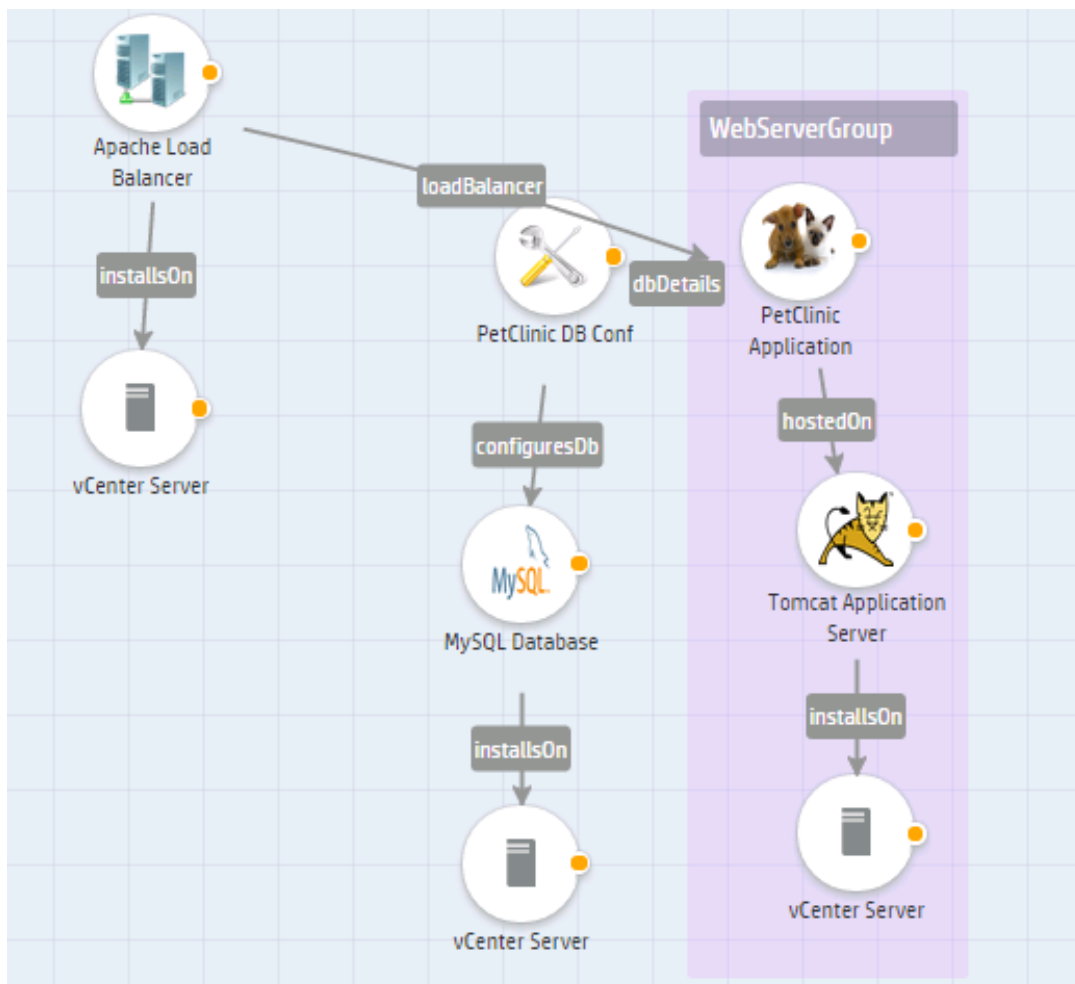


## スケールアウト

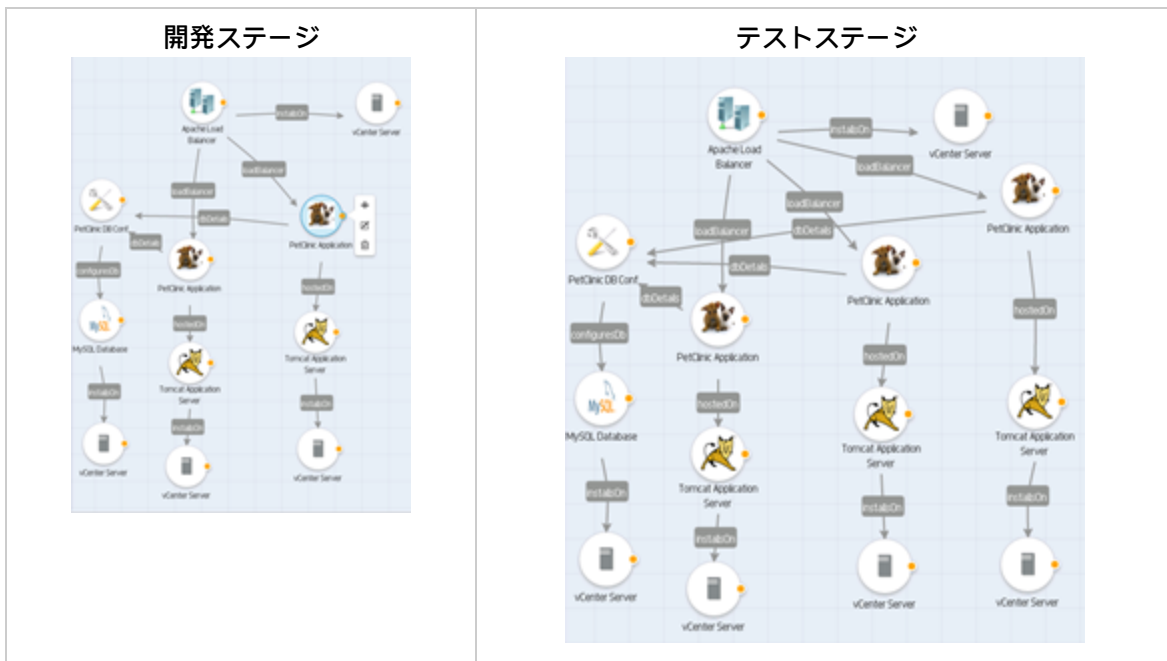
トポロジデザインの作成中に、スケーリンググループを作成できます。スケーリンググループは、スケーラブルスタックを表します。1つのアプリケーションデザイン内で複数のスケーラブルグループを使用できます。

デプロイメントが完了した後でスケールアウトすることができます。スケールアウトすると、スタック全体が複製されます。

たとえば、下の図は、webServerGroupという名前の論理的なグループとしてのWeb層を示しています。



このグループは、開発ステージで1つのグループにスケールインされ、テストステージで下の図のように2つのグループにスケールされます。



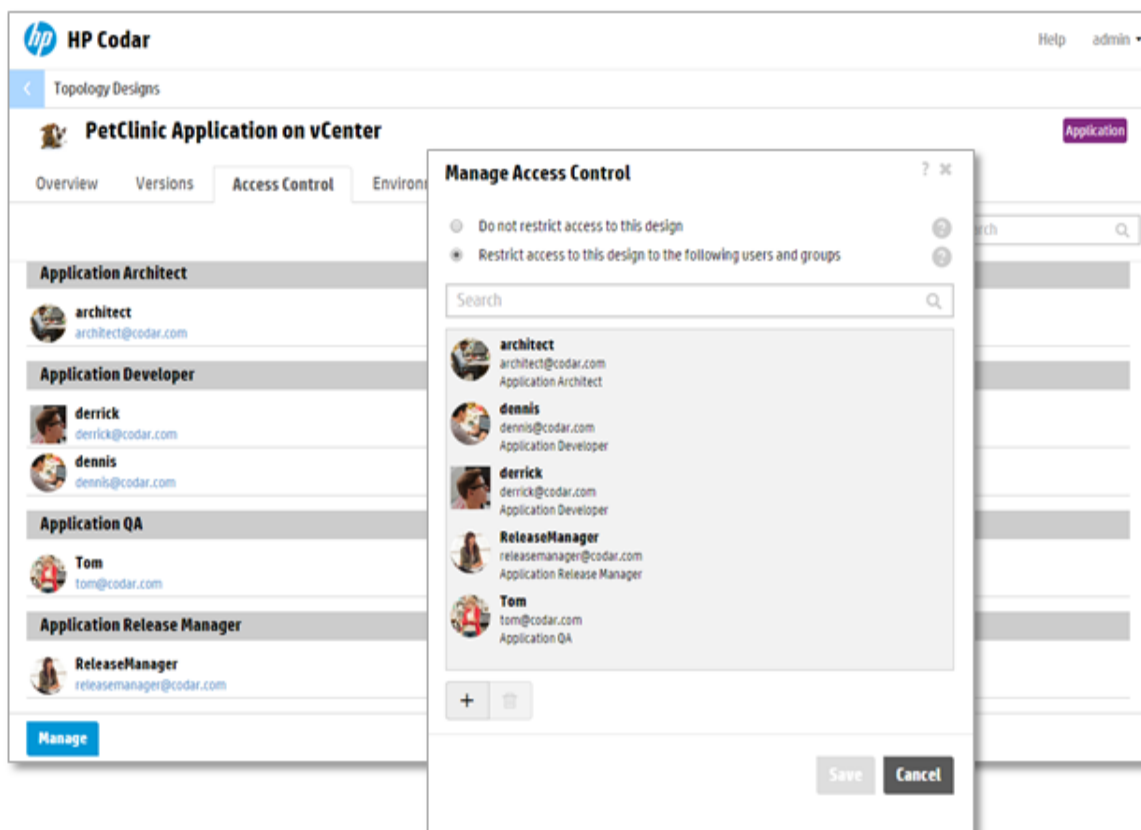
## ユーザーアクセス

ユーザーアクセスは、トポロジデザインで構成できます。ユーザーとLDAPグループの両方をデザインに追加できます。アプリケーションアーキテクトは、デザインを作成し、デザインをパブリックにするか、特定のユーザーに制限することができます。

アプリケーションアーキテクトは、次のようにユーザーとグループを構成できます。

- 各ロールのユーザーは、細かく制御できるようにアプリケーションレベルで定義されます。
- グループは、アプリケーションに対するロールを自動的に割り当てるためのアプリケーションチームを表す必要があります。

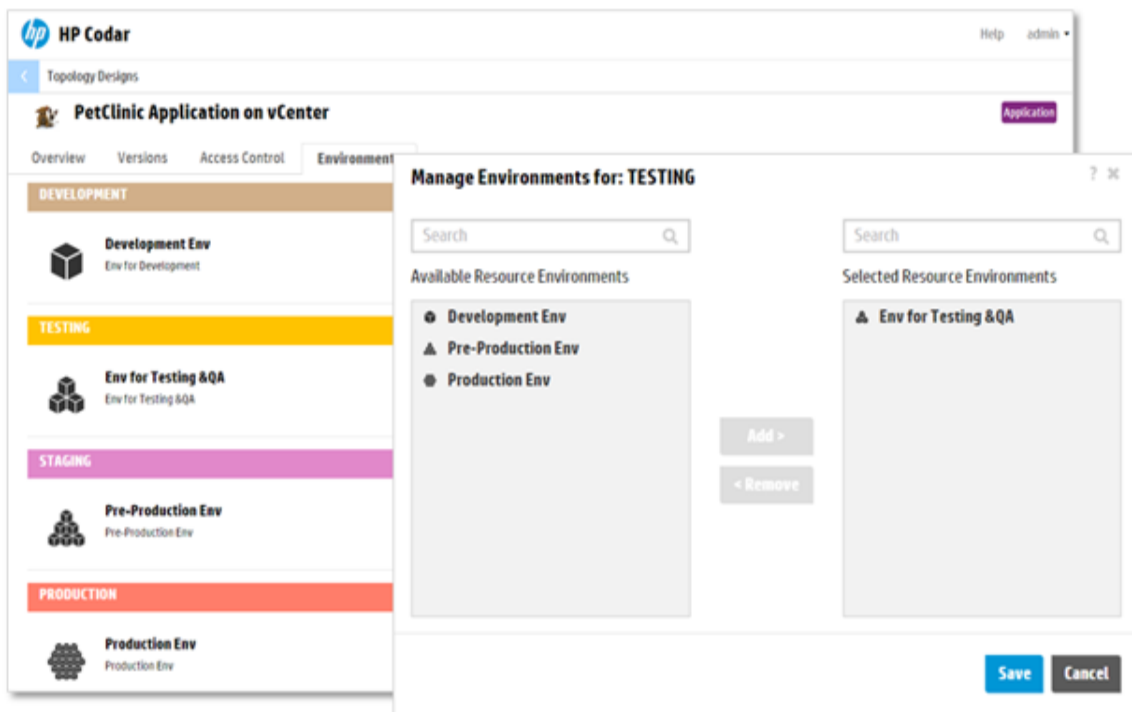
次の図は、アプリケーションアーキテクト、開発者、QA、リリースマネージャーを含むさまざまなユーザー用に構成されたデザインを示しています。



## 環境

各ライフサイクルステージに合わせてアプリケーションレベルで異なる環境を選択できます。たとえば、デプロイメント用にvCenterを構成し、ステージング用にはパブリッククラウド環境を選択することができます。



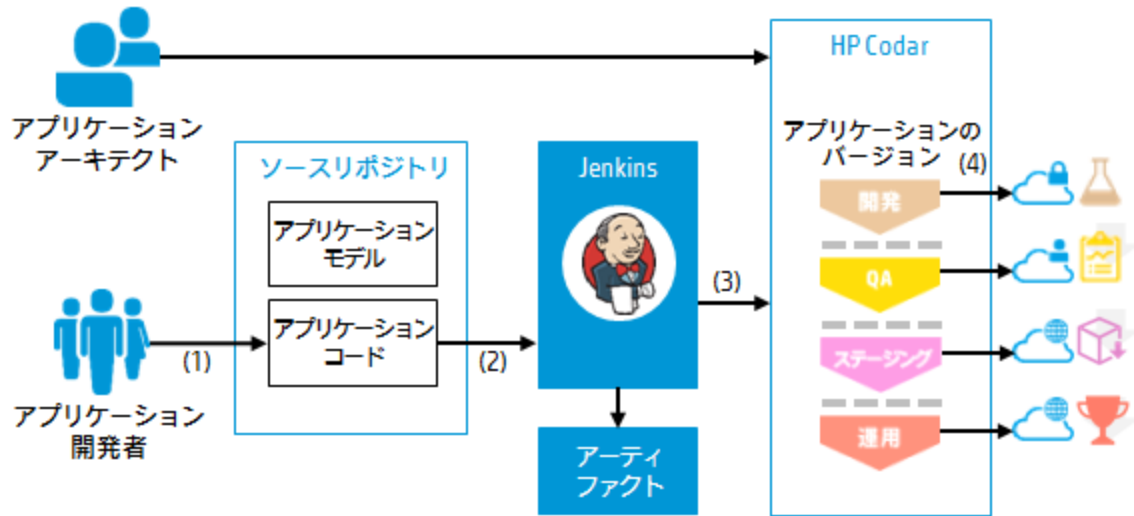


## 外部統合

HP Codarは、オープンで拡張可能です。これは、Jenkins、Hudsonなどのさまざまなビルドシステムと統合できます。REST APIの包括的なセットを他の外部ツールと併用すると、継続的な統合、デプロイメント、デリバリを実現できます。HP Codarのアーキテクチャーには、DevTestおよびDevOpsのカスタマイズ済みフローにフックするオプションも用意されています。

## Jenkinsの統合

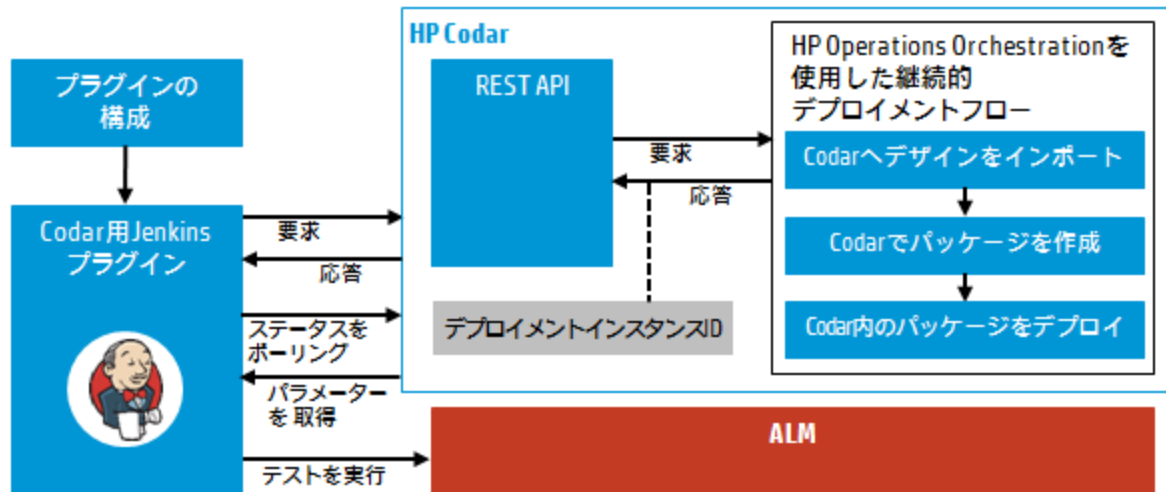
HP Codarには、継続的デプロイメントのためのJenkinsプラグインが含まれています。次の図は、その仕組みを示しています。



1. 開発者が変更をチェックインします。
2. 継続的統合によってビルドが開始されます。
3. Jenkinsプラグインがパッケージを作成してデプロイします。
4. ライフサイクルステージでアプリケーションが異なる環境にデプロイされます。

## ALMの統合

デプロイメントが成功した後に、Application Lifecycle Management (ALM) とHP Codarを統合してテストを実行できます。次の図は、Jenkinsが調整機能としてどのような処理を行うかを示しています。

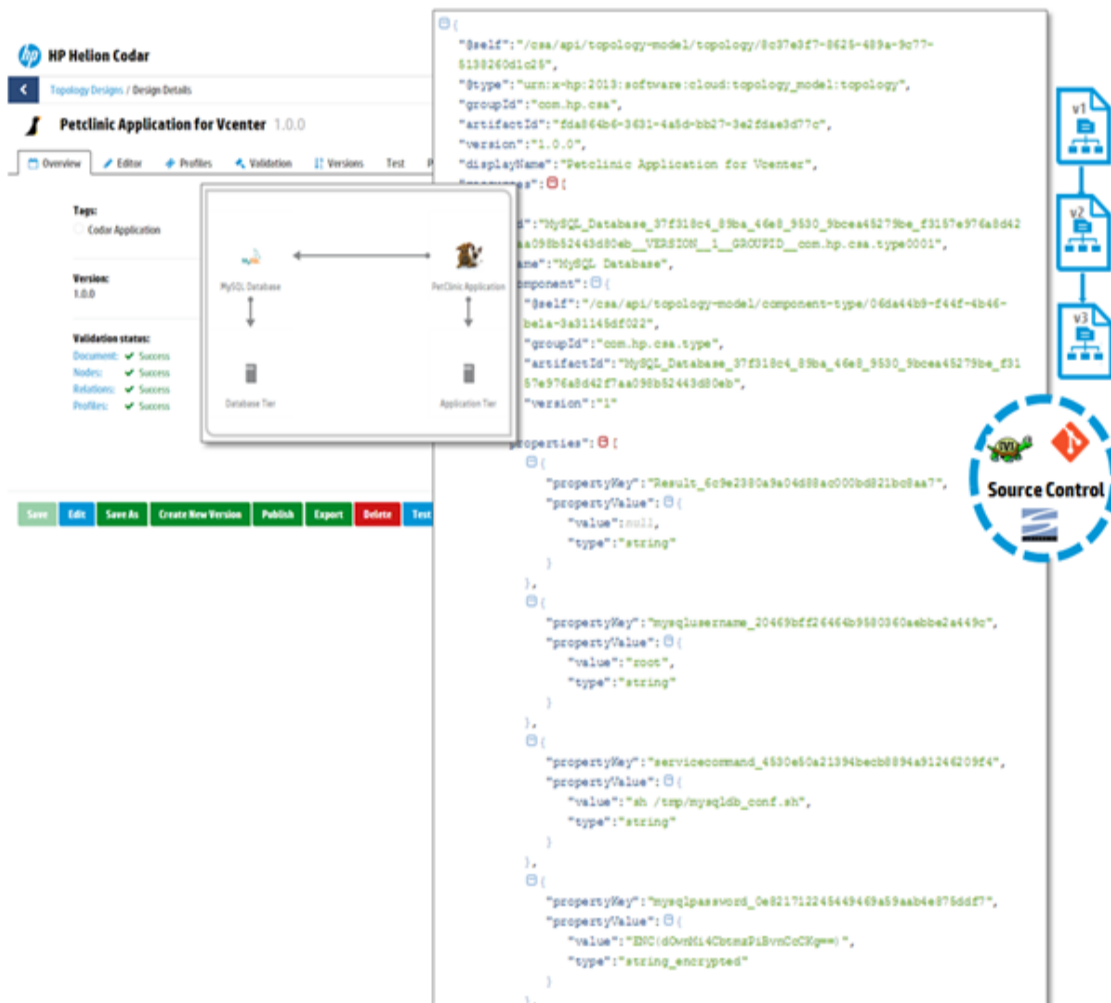


## Infrastructure as code (IaC)

IaC (Infrastructure as Code) を管理すると、ITチームは、インフラストラクチャーとアプリケーションのプロビジョニング方法のコードレビューやユニットテストのコードを開発する際にベストプラク

ティスを利用できます。

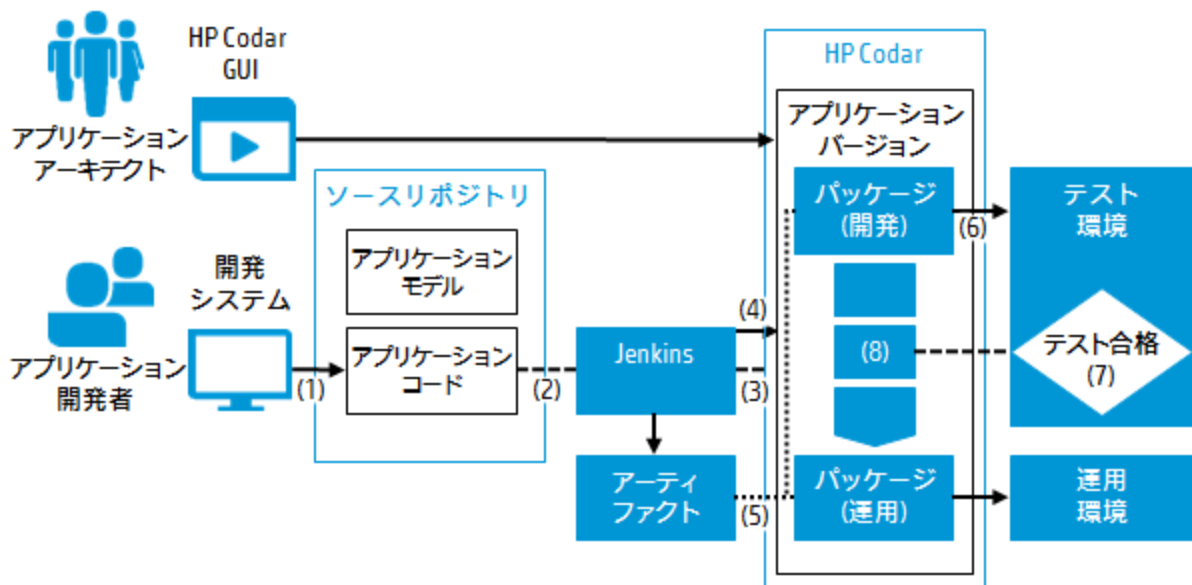
HP Codarは、インフラストラクチャーをコードとして管理できます。サーバー構成、ネットワーク、ボリューム、関係、アプリケーション固有の詳細(アプリケーションバージョン、パッケージ情報など)を含むことができるトポロジデザインは、JSON形式でエクスポートし、ソース管理システムのアプリケーションで管理できます。開発者は、テキストエディタを使用してモデルに変更を加え、自動化のために使用できます。変更したモデルをインポートして、HP Codarに戻すこともできます。



# ユースケース: 継続的な統合、デプロイメント、デリバリ

目的は、アプリケーションを継続的統合 (CI) と継続的デプロイメントに対応させることです。アプリケーション開発者はアプリケーションをコード化し、アプリケーションアーキテクトはアプリケーションをHP Codarインターフェースでモデル化し、そのアプリケーションモデルをコード (IaC) としてエクスポートします。アプリケーション開発者がコードをチェックインすると、Jenkinsビルドがトリガーされ、アプリケーションモデルを使用してアプリケーションが特定の環境にデプロイされます。アプリケーションのデプロイが終了すると、継続的開発プロセスが継続的デリバリに拡張されますが、デプロされたインスタンスでアプリケーション固有のテストを自動的に実行できます。このとき、テストの結果によっては、アプリケーションが別の環境にデプロイされる可能性があります。

次の項では、このシナリオがHP Codarでどのように実現されるかを説明します。



## アプリケーションのモデル化

アプリケーションアーキテクトは、アプリケーションをグラフィカルにモデル化します。これは、デザイナーインターフェースで、デザインの必要コンポーネントを追加し、それを関係を使って接続して行われます。HP Codarには、標準的なコンポーネントのパレットがあります。また、HP Operations Orchestration、Chefなどの各種開発エンジンからコンポーネントをインポートする (取り込む) こともできます。このようなデザインはアプリケーションモデルと呼ばれ、アプリケーションのデプロイ方法を表現します。アプリケーションモデルは、JSON形式でエクスポートして、外部のソースリポジトリで管理できるため、IaC (Infrastructure as Code) を実現できます。

## 継続的な統合とデプロイメント

継続的統合では、サンプルアプリケーションのコードと、アプリケーションをデプロイするためのモデル (JSON形式) は、ソースリポジトリにあります。

アプリケーションの開発者がアプリケーションのコード変更を行い、そのコードをソースリポジトリにチェックインすると (1)、Jenkinsによりビルドが開始されます (2)。

HP Codarによって、HP CodarのIPアドレス、ユーザー名、パスワードなどの詳細情報を持つJenkinsプラグインが提供されます。接続がポストビルドステップの一環として確立され、APIが起動されます (3)。次に、そのAPIによって、各種アクションを実行するワークフローが起動され、継続的開発と継続的デリバリが実現されます。

## アプリケーションデザインのインポート

アプリケーションモデルがHP Codarにまだインポートされていない場合や、アプリケーションモデルに変更があった場合、継続的デプロイメントワークフローは、それをアプリケーションデザインの新しいバージョンとして、JSON形式 (IaC) でHP Codar (4) にインポートします。これにより、アプリケーションの開発者とアーキテクトが行った変更を、デプロイメント時に考慮することができます。

アプリケーションモデルがすでにインポート済みか、またはアプリケーションデザインに変更がない場合、このインポートオペレーションは実行されず、HP Codar内のアプリケーションバージョンは同じままに保たれます。この点は、注意してください。アプリケーションモデルは、デザイナーの [トポロジ] タイルに表示できます。

## パッケージの管理

パッケージはアプリケーションデザインのスナップショットを表し、デザイン内でのプロパティのパラメーター化を可能にします。パッケージはアプリケーションの特定のビルドを表すということもできます。

パッケージはアプリケーションに関するデプロイ可能な最小単位です。これは、実装アーティファクト (アプリケーションがデプロイされる方法) と、デプロイメントアーティファクト (war、earなど、デプロイされるライブラリの場所) の両方を表します。

パッケージは、ライフサイクルステージに関連付けられます。1つのパッケージは、開発、テスト、ステージング、運用のいずれかのステージに属することができます。

パッケージはパイプライン管理に関連付けられます。パッケージは、特定のステージでの昇格や拒否など、ライフサイクルステージをまたいで管理できます。たとえば、QAルールを持つユーザーは、パッケージを拒否できます。

## タスク

- **特定のアプリケーションバージョンからパッケージを作成** - アプリケーションバージョンは、複数のパッケージから構成される場合があります。
- **パッケージのデプロイまたは再デプロイ** - この場合、アプリケーションデザインに対応する状態と、パッケージに指定されたデザインのプロパティのフルフィルメントが行われます。
- **パッケージの削除** - [リリースパイプライン] タブを開き、Ctrlキーを押しながらパッケージを複数選択し、[削除] をクリックします。  
インスタンスが関連付けられているパッケージは削除できません。

## 環境へのデプロイ

パッケージが作成されると、継続的デプロイメントワークフローが、環境に基づいてアプリケーションデザインのフルフィルメントを実行します (6)。パッケージのデプロイメントは、[デプロイメント] タブに表示できます。

The screenshot shows the HP Helion Codar interface. At the top, there is a navigation bar with the HP logo, the text "HP Helion Codar", and links for "Help" and "admin". Below this is a breadcrumb trail: "パッケージ". The main content area is titled "Build 36" and has three tabs: "概要", "構成", and "デプロイメント", with the last one being active. Below the tabs is a table listing deployment instances.

Build ID	Status	User	Timestamp
Build 36:Application	オンライン	admin	2014/11/11 PM 3:04:33
Build 36:my dev environment based on build #36	オンライン	admin	2014/11/11 PM 3:04:11

ランブック自動化エンジンは、インフラストラクチャーレイヤー、プラットフォームフェイルオーバーレイヤー、アプリケーションレイヤーのフルフィルメントを実行するデザインに基づいて、実行計画を作成します。ユーザーは、特定のパッケージのデプロイメントのステータスを監視し、デプロイされたアプリケーションのグラフィック表現を表示できます。それには、コンポーネントレベルのプロパティとアクションも含まれています。

## デザインの発行

デザインを発行すると、サービスコンシューマーに対する提供として使用可能になります。デザインを発行するには、HP CSAライセンスがインストールされている必要があります。

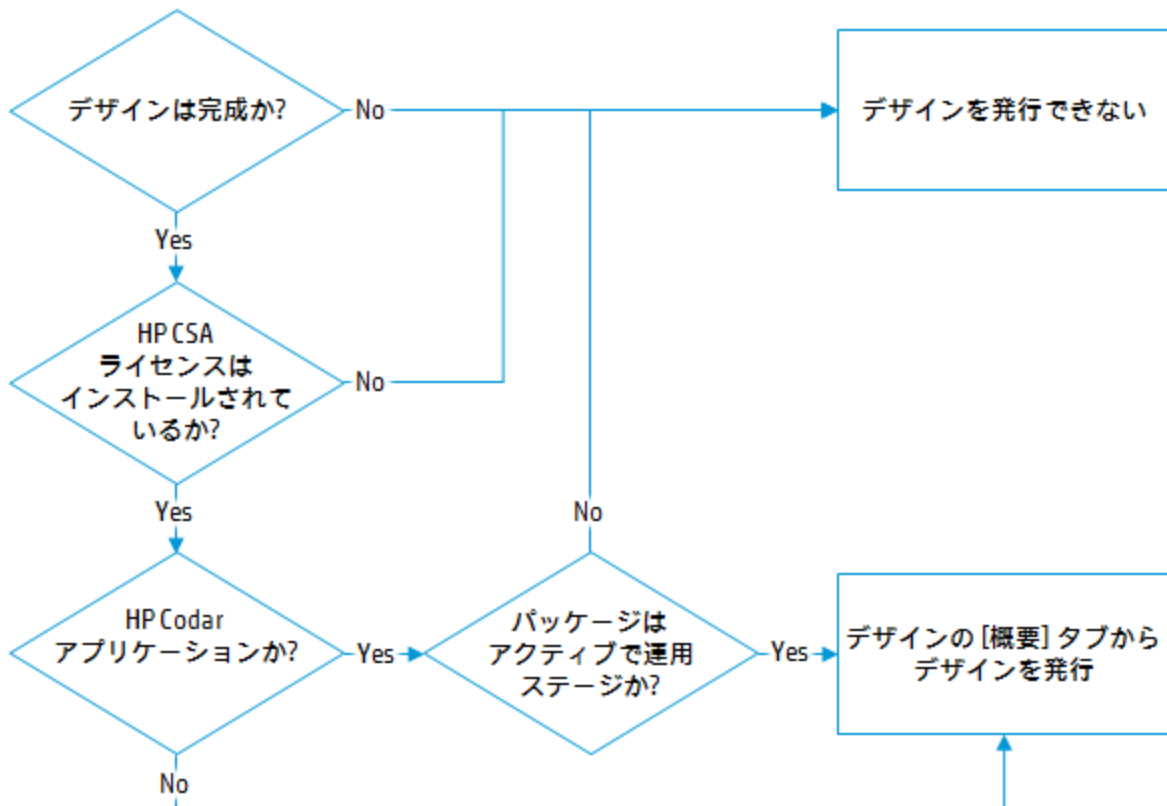
運用ステージのアクティブパッケージを持つ完全デザインは、パッケージ固有のプロパティをデザインの一部として持ち、発行可能です。

運用ステージのアクティブパッケージを持つ部分デザインは、パッケージ固有のプロパティをデザインの一部として持ちますが、運用パッケージをデプロイすることによって最終的な構成されたデザインを作成するまでは発行できません。

部分デザインの発行方法は、インストールされているライセンスによって異なります。

- 運用ステージまで進んだHP Codarアプリケーションデザインは、運用インフラストラクチャーにデプロイされ、構成された運用デザインがその後の運用デプロイメントで見えるようになります。その後、デザインをサービスコンシューマーに発行できます。
- HP Codarアプリケーションデザインでないデザインは、[テスト]タブから構成されたデザインとして保存する必要があります。その後、デザインをサービスコンシューマーに発行できます。

次の図は、使用するライセンスでデザインを発行できる条件を示しています。



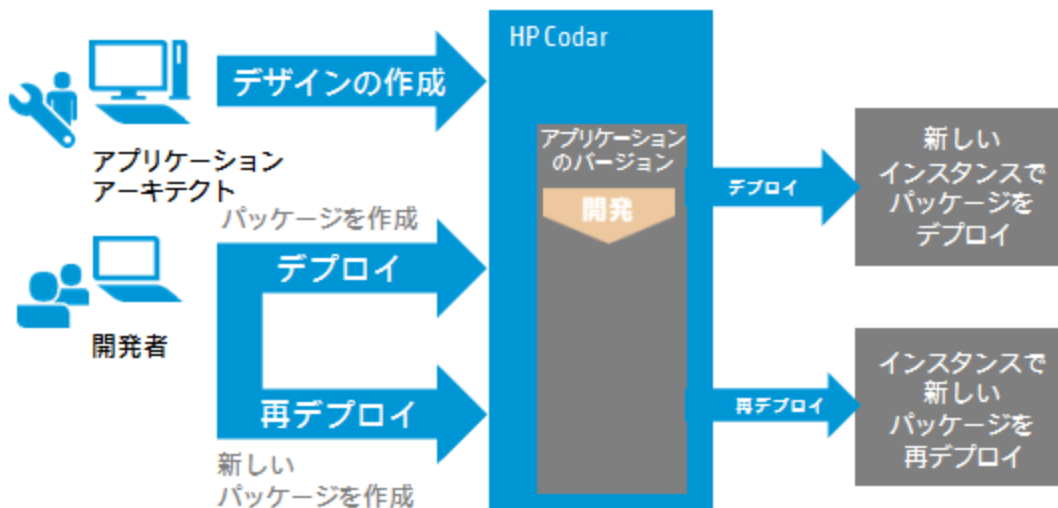
# ユースケース: パッケージのデプロイと再デプロイ

この目的は、パッケージをデプロイし、同じインスタンスを使用して、古いバージョンがデプロイされているインスタンス上で新しいバージョンのパッケージを再デプロイすることです。

アプリケーションアーキテクトが、アプリケーションをモデル化し、パイプライン管理用にマークを付けます。次に、開発者がパッケージを作成してデプロイします。パッケージがデプロイされると、新しいインスタンスが作成されます。デプロイメントはトポロジデザインを基にしています。

デプロイメントの後に、同じインスタンスを使用して、パッケージの新しいバージョンを再デプロイすることができます。インスタンスを新しいパッケージまたはビルドにアップグレードしたりパッチを適用したりすることができます。

次の図は、このプロセスを示しています。





# ユースケース: デプロイメントとスケールアウト

目的は、スケーラブルスタックを作成し、デプロイメント後に必要に応じてスタックをスケールアウトすることです。アプリケーションアーキテクトが、アプリケーションをモデル化し、パイプライン管理用にアプリケーションにマークを付けます。アーキテクトが、さまざまなライフサイクルステージでスケールアウトする必要があるコンポーネントを識別します。スケーラブルスタックには、1つのコンポーネントまたはコンポーネントのグループを含めることができます。開発中に、スタックを1つスケールインしたり、テスト中にスタックを2つにスケールアウトしたりすることができます。

# ライフサイクルのステージおよびアクション

パッケージには、次のライフサイクルステージとそれに対応するアクションがあります。

ステージ	レベルを上げる	デプロイ、再デプロイ	編集	削除	拒否
開発	○	○	○	○	×
テスト	○	○	×	○	○
ステージング	○	○	×	○	○
運用	×	○	×	○	○

注: 各自のロールにより、このすべてまたは一部の機能にアクセスできない場合があります。

[パッケージ] タブには、ライフサイクルステージと現在のステージが表示されます。またライフサイクルステージは、パッケージの [デプロイメント] タブで、デプロイメントごとに表示されます。

次のアクションを使用して、パッケージをあらゆるステージでデプロイまたは移動します。

- **レベルを上げる:** パッケージを次のライフサイクルステージに移動します。パッケージの状態は、アクティブなままに保たれます。
- **デプロイ、再デプロイ:** パッケージをデプロイします。
- **編集:** パッケージのプロパティを変更します。
- **拒否:** パッケージを他のステージに進めないようにします。パッケージは、現在のステージに留まり、その状態は拒否済みに設定されます。アクションボタンは、使用できなくなります。
- **削除:** パッケージを削除します。パッケージはシステムから恒久的に削除されます。パッケージを削除できるのは、デプロイ済みの関連するすべてのインスタンスがキャンセルまたは削除されている場合のみです。
- **更新:** パッケージの現在のステータスを取得します。

## パッケージの状態

パッケージには、次の状態があります。

- アクティブ
- 拒否済み

パッケージを拒否すると、そのパッケージは現在のステージに保たれ、その状態は拒否済みに設定され、それ以降はアクションを適用できません。ただし、パッケージを削除することは可能です。パッケージはシステムから削除されます。

パッケージのレベルを上げると、パッケージは次のステージに移動し、アクティブ状態に留まります。パッケージは常に開発ステージで作成されます。HP Codar Jenkinsプラグインが構成されている場合、ビルドが成功すると、JenkinsプラグインがHP Codarと通信してパッケージを作成します。

## ライフサイクルステージによるサービスデザインのグループ化

アクティブパッケージの部分デザインでは、パッケージのデプロイウィザードでプロビジョニングを行う際に、サービスデザインを選択する必要があります。サービスデザインはライフサイクルステージごとにグループ化することが可能です。これにより、ライフサイクルステージでパッケージをデプロイする際、そのライフステージで指定されたサービスデザイングループのみが表示されます。

特定のライフステージ用にサービスデザインをグループ化するには、各トポロジデザインで、ライフサイクルステージの名前でタグを作成します。たとえば、Developmentタグを作成し、Developmentライフサイクルステージに必要なデザインに関連付けることが可能です。

注:[テスト]タブでテスト実行ウィザードを実行する場合、タグでグループ化されたデザインではなく、すべてのデザインが表示されます。

## 次のステップ

『HP Codar Installation Guide』および『HP Codar Configuration Guide』に、このソフトウェアをダウンロードし、インストールして構成する方法の説明が記載されています。『HP CodarAPI and CLI Reference』では、REST APIの概要が説明され、各APIの詳細を参照する方法が説明されています。さらに、コマンドラインインターフェイスについても説明されています。また、アプリケーションからオンラインヘルプにアクセスし、具体的なタスクのヘルプ情報を取得することができます。

# ドキュメントのフィードバックの送信

本ドキュメントについてのご意見、ご感想については、電子メールで[ドキュメント制作チームまでご連絡](#)ください。このシステムで電子メールクライアントが設定されていれば、このリンクをクリックすることで、以下の情報が件名に記入された電子メールウィンドウが開きます。

## **Feedback on コンセプトガイド (Codar 1.50)**

本文にご意見、ご感想を記入の上、[送信] をクリックしてください。

電子メールクライアントが利用できない場合は、上記の情報をコピーしてWebメールクライアントの新規メッセージに貼り付け、[csadocs@hp.com](mailto:csadocs@hp.com)宛にお送りください。

お客様からのご意見、ご感想をお待ちしています。

