Deployment Manager for 1:1 Customer Engagement

STUDENT GUIDE

PEGA Academy

This document is the property of:
Pegasystems Inc.
1 Rogers Street
Cambridge, MA 02142
Phone: (617) 374-9600
Fax: (617) 374-9620
www.pega.com

Mission: Deployment Manager for 1:1 Customer Engagement

Product: Pega Deployment Manager 5.5.6

URL: https://academy.pega.com/mission/deployment-manager-11-customer-engagement/v3

Date: 12 February 2024

Contents

Deployment Manager application pipelines

Description

Familiarize yourself with the various application pipelines used in Pega Customer Decision Hub™ implementations to leverage the full power of change management and simulations.

Learning objectives

- Define application pipelines
- Describe the different types of application pipelines

Deployment Manager application pipelines

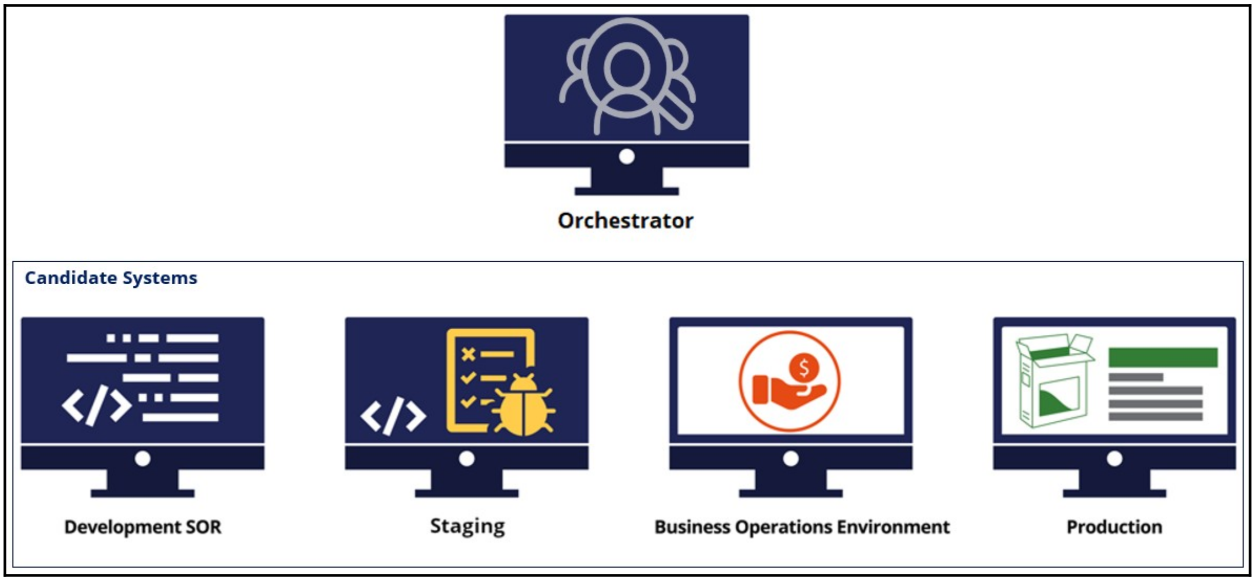
Pega Deployment Manager is the standard way to test and deploy Pega applications through application pipelines.

Deployment Manager exposes all Pega Platform™ capabilities that are necessary to automate your DevOps workflows, including branch merging, application packaging, applying quality gates, and promoting an application to different environments.

Transcript

In this video, you will learn about the Deployment Manager environments. You will also understand the significance of different application pipelines in Deployment Manager that are relevant to Customer Decision Hub implementations.

A typical decision management project has five environments involved. An orchestrator to manage and automate the deployment process of artifacts to the candidate environments. A development (System of Record) environment to develop enterprise capabilities. A staging environment to test the changes and analyze their technical impact. A Business Operations Environment (BOE) to make business changes. A production system to serve real customers with the latest and greatest artifacts.



Deployment Manager environments

The orchestrator has the Deployment Manager installed in it and it orchestrates all the tasks as defined in the different application pipeline models. Orchestrator exposes the pipeline modeling experience for users to design their deployment pipelines. It resides on a dedicated Pega server which is configured to interact with each of the candidate system.

Each Pega environment managed by Deployment Manager is known as a candidate system. Candidate systems execute most deployment tasks at the direction of the orchestrator. The candidate environments use a default operator and authentication profiles to establish communication with the orchestrator system.

Among the candidate environments, the development environment has special responsibilities, as merges are performed on this environment. Application packages are generated in this environment during deployments.

The development environment is referred to as the system of record (SoR) to highlight these responsibilities.

Deployment Manager stores application artifacts that are created throughout the lifecycle of a deployment within a repository. To use Deployment Manager, you need a supported artifact repository.

Candidate systems use repositories to move application artifacts when they are promoted between environments. The deployment process automatically versions application artifacts.

In a typical decision management project, a development team, and a business operations team work on various application artifacts.

Because various teams work on several artifacts in different environments, you need a consistent methodology and a process to ensure that the artifacts that are created are synchronized and consistent across all four candidate environments. To keep these artifacts up to date across all the environments, you use application pipelines.

You will learn about three application pipelines that leverage DevOps best practices for Customer Decision Hub implementation to deploy revisions and test how changes made to decision strategies might reflect in real-life scenarios.

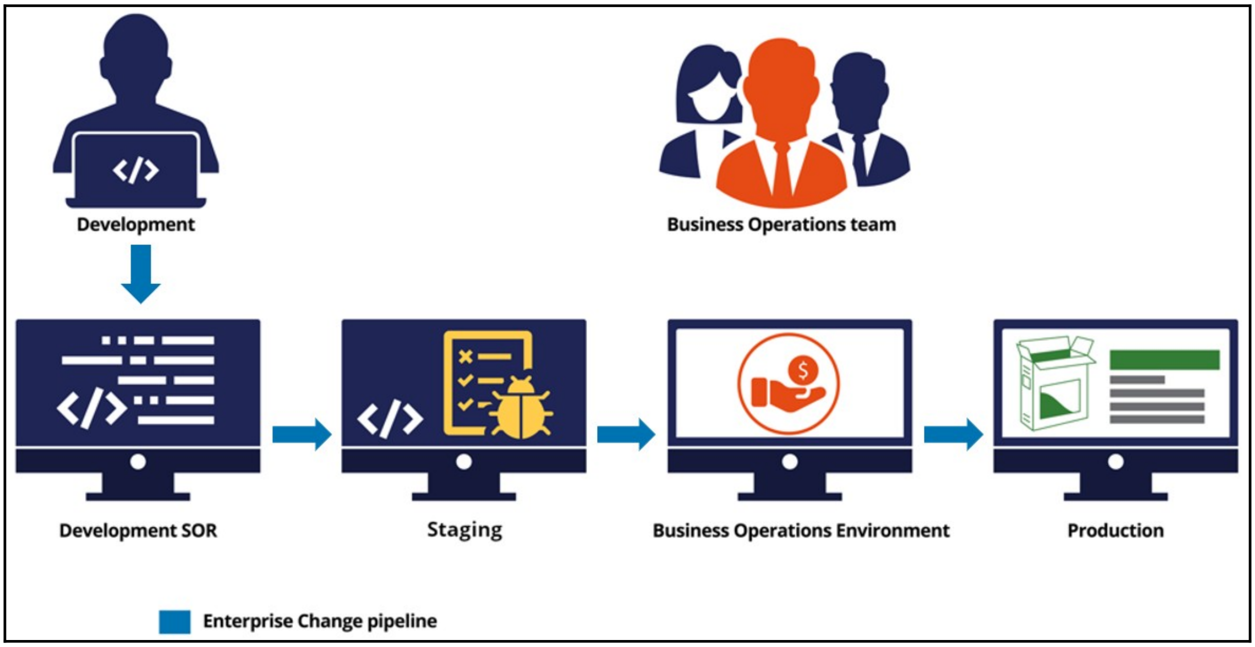
Enterprise change pipeline

The development team works on core application changes as part of enterprise release cycles to improve the capability of the application and to meet more business requirements. The changes that are made as part of the enterprise release cycles are tested and deployed through an enterprise change pipeline. Examples of a typical core application change include creating a new customer property, integrating an additional channel, and creating a new service to gather additional customer data, and so on.

The enterprise change pipeline is used to migrate core application changes released by the development team.

All these changes are packaged and pushed from the development environment to staging, BOE, and then to production system by using the enterprise change pipeline.

Updates to application functionality are handled through typical developer practices such as branching and ruleset lock-and-roll. For the Customer Decision Hub Implementation application, updates are typically maintained in the Rules ruleset, and any other client-specific rulesets. Nothing in the Artifacts ruleset should flow through this pipeline.



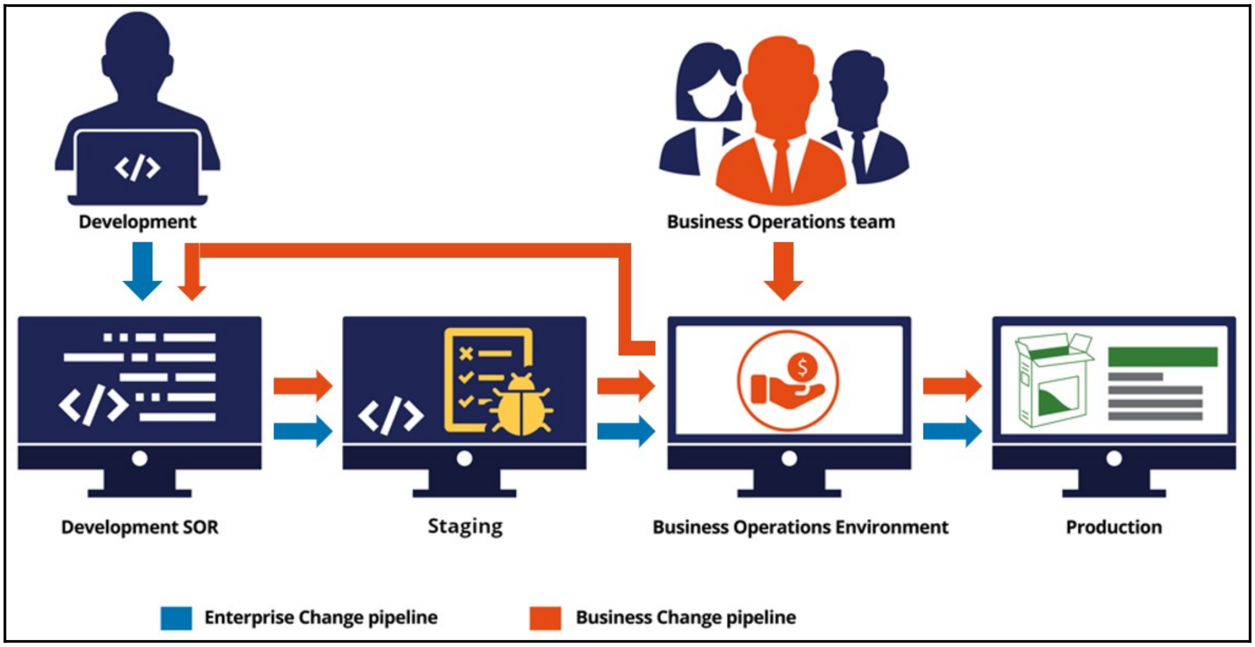
Business change pipeline

Business change pipelines support business changes to your Pega Customer Decision Hub application. This pipeline offers Customer Decision Hub business users the means to deploy changes that are implemented to their applications outside the enterprise release cycles.

The business content team and enterprise capabilities team make business changes in the BOE.

You can make two types of business changes to your Customer Decision Hub application from the BOE. The first type is a change request that is initiated in the 1:1 Operations Manager application. For example, the change might be creating an action, creating a web treatment, or updating an action-level eligibility condition. These requests are usually high-volume business changes that the business content team manages. The other type includes changes that are not managed by 1:1 Operations Manager but completed using revision management capabilities in the Customer Decision Hub portal on the BOE. For example, the change might be modifying updating a decision strategy that implements eligibility conditions, using a new decision table, or a scorecard to implement specific business requirements, and so on. These changes are low volume capability changes that the enterprise capabilities team manages. You can deploy both these changes as part of a single revision.

When a business change is completed, you can deploy the change from the BOE to all other environments. After the deployment from the BOE is initiated, the artifacts that are related to the new revision are merged into the branch in the development environment. After the branch is merged to the enterprise application, a new version of the enterprise application is created. This application is then packaged and moved to higher environments - staging, BOE, and production.



Data migration pipeline

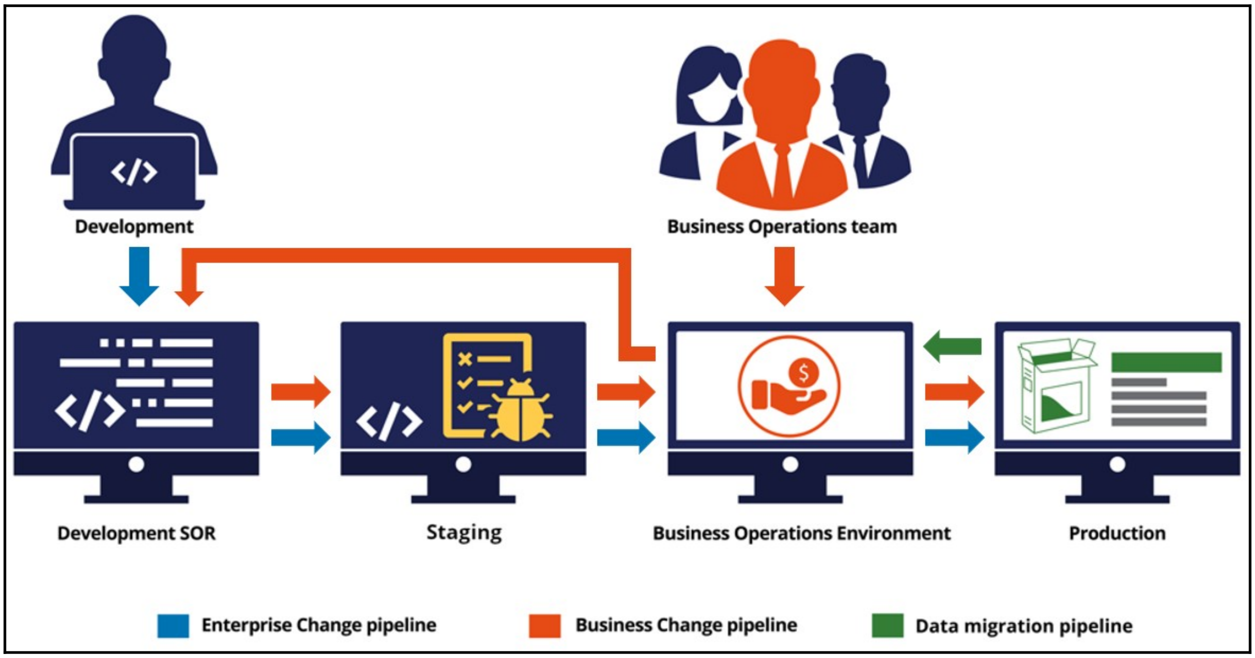
Typically, after implementing the business changes, you need to simulate the result of the changes to understand if the business outcomes are as expected. Simulation tests provide you with significant insight into how your changes to decision logic affect the results that your strategies produce. However, if tests are based on sample data, the results do not always reflect the effect that your changes have in real-life scenarios. To ensure that simulations are reliable enough to help you make important business decisions, you can migrate a sample of your production data to BOE.

The data migration pipeline provides you with the means to test how the changes that you make to decision strategies affect their results, without running these strategies in a live production environment.

By using a data migration pipeline, you can export a sampled dataset, which includes inbound and outbound interaction history and adaptive models, from the production environment, and then import that data into the BOE. As a result, the simulations are based on the migrated production data and enables you assess how your changes are reflected in real-life scenarios.

For example, you can sample a selection of 20% of real customer interactions, and then migrate the data to the BOE to test how the changes to existing strategies may affect the business outcomes.

You can also schedule a data migration pipeline to run during a specified period by creating and running a job scheduler.



To summarize, successfully setting up Deployment Manager to conduct proper change management and provide sampled production data for simulations enables business to seamlessly deploy revisions and test how decision startegy changes affect real-life scenarios.

Migrating development changes with an enterprise change pipeline

Description

Learn how an enterprise change pipeline facilitates the initial deployment and additional structural and technical ongoing enhancements that the developers made in the implementation application outside the business release cycle.

Learning objectives

- Explain how an enterprise change pipeline facilitates migration of development changes.
- Create an enterprise change pipeline for initial deployment.
- Deploy development changes with an enterprise change pipeline.

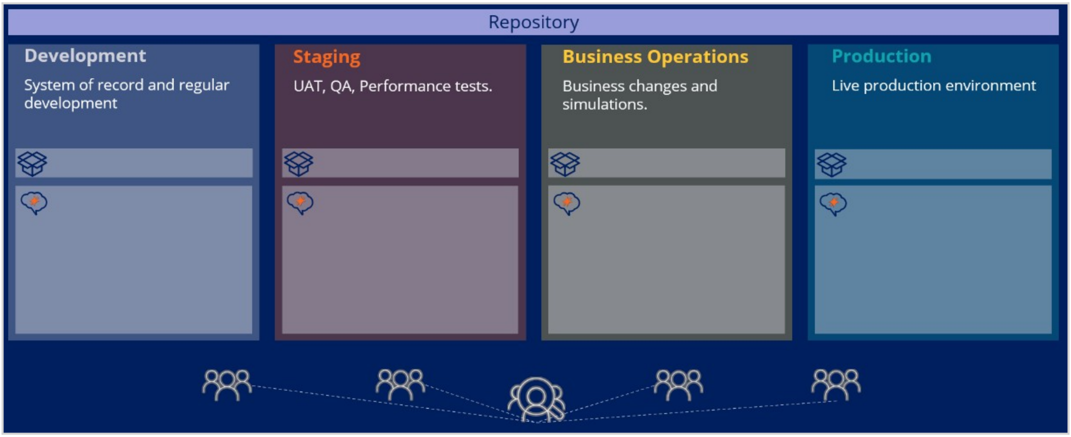
Enterprise change pipeline

Enterprise change pipelines support migrating development changes from one Pega environment to another. Developers can use these pipelines to migrate development changes through branch merging and validation, application packaging, artifact repository management, deployments, test execution, guardrail compliance, and test coverage enforcement.

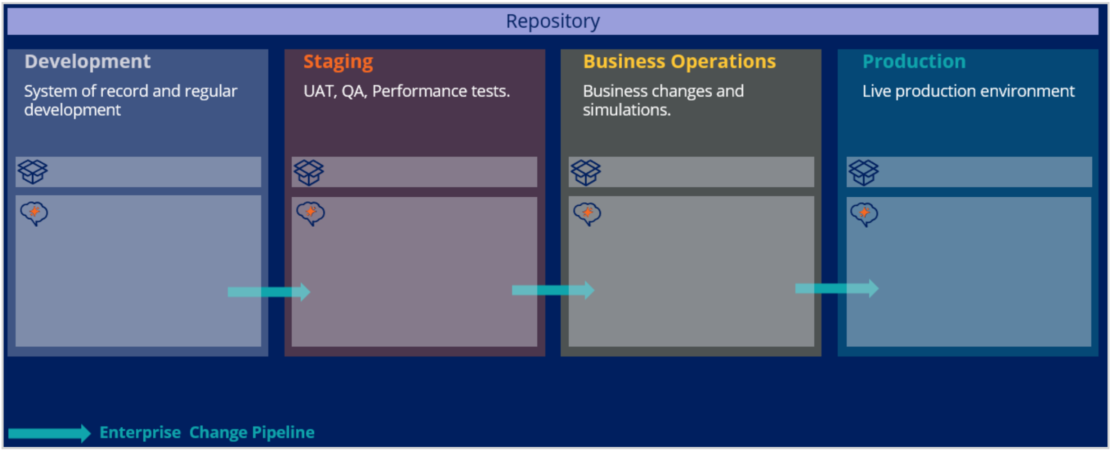
Transcript

This this video will explain how an enterprise change pipeline facilitates initial deployment and additional structural and technical changes made by the developers in the implementation application outside the business release cycle.

A typical decision management project includes five environments. A Development environment to develop enterprise features (System of record), the UAT/Staging environment to test the changes and analyze technical impact, a Business Operations Environment (BOE) to make business changes, a production system to serve customers with the most recent artifacts, and an orchestrator to manage and automate deploying artifacts from one environment to another.


Repository
Development
System of record and regular development
Staging
UAT, QA, Performance tests.
Business Operations
Business changes and simulations.
Production
Live production environment

The enterprise change pipeline is used to migrate core application changes that the development team releases.


Repository
Development
System of record and regular development
Staging
UAT, QA, Performance tests.
Business Operations
Business changes and simulations.
Production
Live production environment
Enterprise Change Pipeline

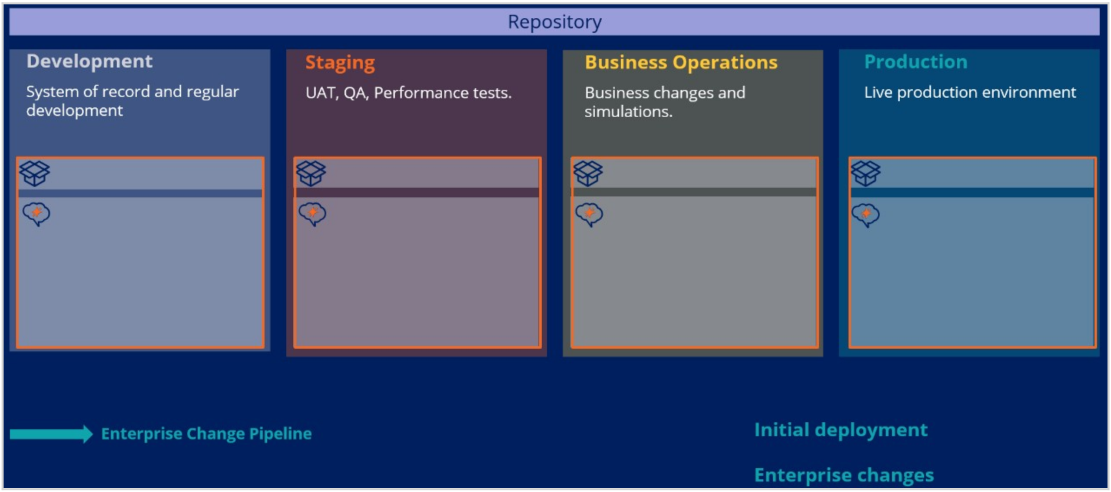
The development team and the enterprise capabilities team work on core application changes as part of enterprise release cycles to improve the application and to meet project requirements.

After they complete the initial development phase and unit-test the changes in the development environment, the system verifies whether the rules are following development standards in policy, validates test rulesets, branches, and missing rule references as per the application compliance.

Similarly to deployment processes, a predefined product rule exports and migrates the rules from one environment to another. After validation, the new application package is created and sent by the system to the repository.

As the next step, the application package is deployed from the repository to the higher environments.

The enterprise change pipeline is used to promote initial deployment of the applications and enterprise change deployment of the ongoing enhancements


Repository
Development
System of record and regular development
Staging
UAT, QA, Performance tests.
Business Operations
Business changes and simulations.
Production
Live production environment
Enterprise Change Pipeline
Initial deployment
Enterprise changes

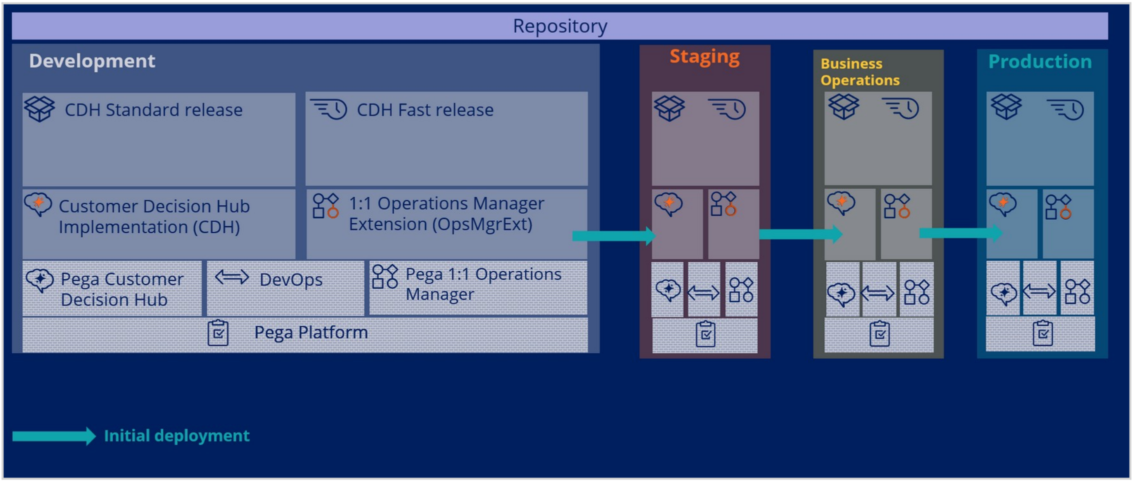
The initial deployment is triggered after the enterprise capabilities and the core structure of Pega Customer Decision Hub™ is set up.

In the development environment, Pega Customer Decision Hub is installed, and Pega Customer Decision Hub setup wizard is run. The wizard generates four applications: Customer Decision Hub implementation (CDH), 1:1 Operations Manager extension, CDH Standard release, and CDH Fast release applications. The two overlay applications are both built on top of Customer Decision Hub and 1:1 Operations Manager extension. 1:1 Operations Manager extension, with predefined configuration, sets the stage for change management. The 1:1Operations Manager extension is built on top of Pega 1:1 Operations Manager. Both Customer Decision Hub and the 1:1 Operations Manager extension applications are built on top of Pega DevOps to use the Deployment Manager features during migration.

The staging, BOE, and the production environments must have Pega Platform™, DevOps, and Customer Decision Hub installed. They also must have access to the repository.

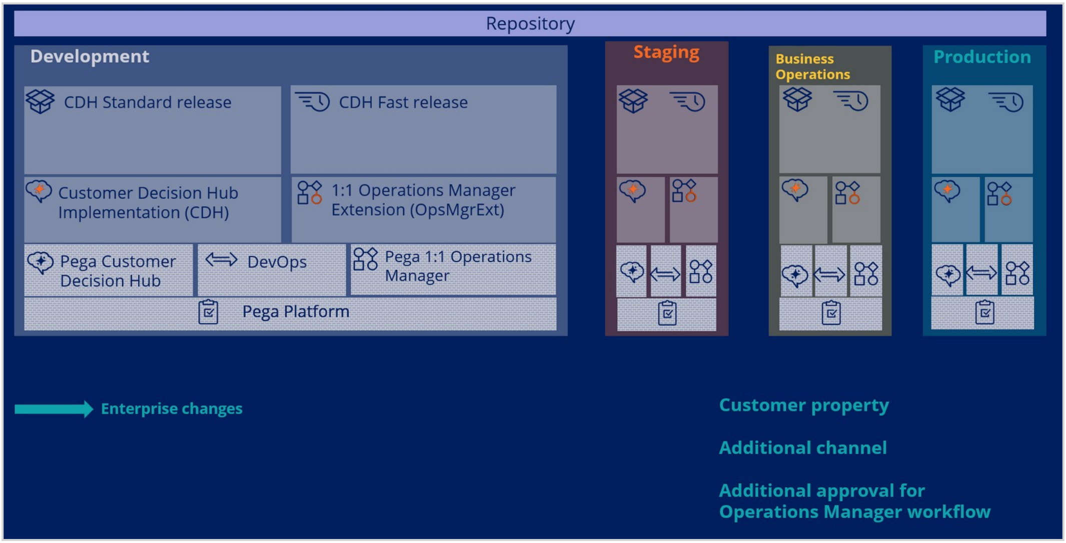
When the project reaches a mature state and is ready for change management, the initial deployment is triggered. The developed enterprise capabilities and the core structure of Customer Decision Hub is pushed from the development environment to the other environments using the enterprise change pipeline.

Post deployment, your system is under change management. The business teams can perform business changes through the change request process from the business operations environment.

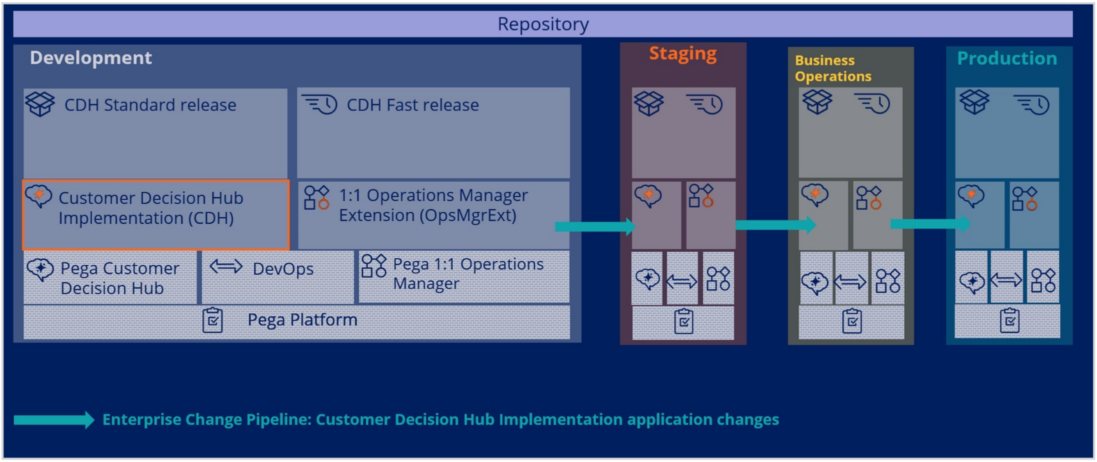

Repository
Development
CDH Standard release
CDH Fast release
Customer Decision Hub Implementation (CDH)
1:1 Operations Manager Extension (OpsMgrExt)
Pega Customer Decision Hub
DevOps
Pega 1:1 Operations Manager
Pega Platform
Staging
Business Operations
Production
Initial deployment

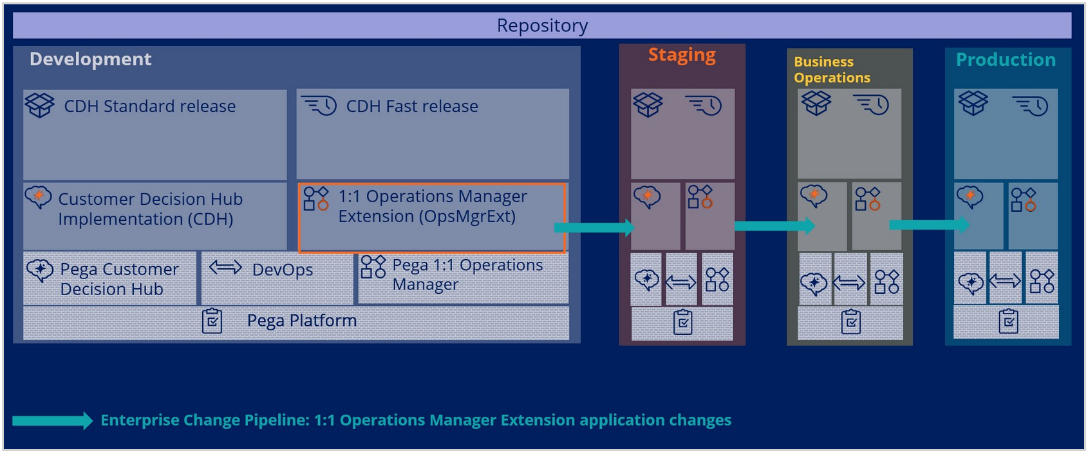
After the initial deployment, to meet more business requirements, you enhance Customer Decision Hub on an ongoing basis in the development environment. For example, create a new customer property, integrate an additional channel, add an additional approval for the Operations Manager workflow.

When the enterprise changes are deployed, only the rulesets of the implementation application and the data instances are moved.


Repository
Development
CDH Standard release
CDH Fast release
Customer Decision Hub Implementation (CDH)
1:1 Operations Manager Extension (OpsMgrExt)
Pega Customer Decision Hub
DevOps
Pega 1:1 Operations Manager
Pega Platform
Staging
Business Operations
Production
Enterprise changes
Customer property
Additional channel
Additional approval for Operations Manager workflow

A project might have multiple enterprise change pipelines to merge ongoing regular developments. For example, one that deploys developer changes made to Customer Decision Hub implementation application and another to deploy developer changes made to the 1:1 Operations Manager extension application.

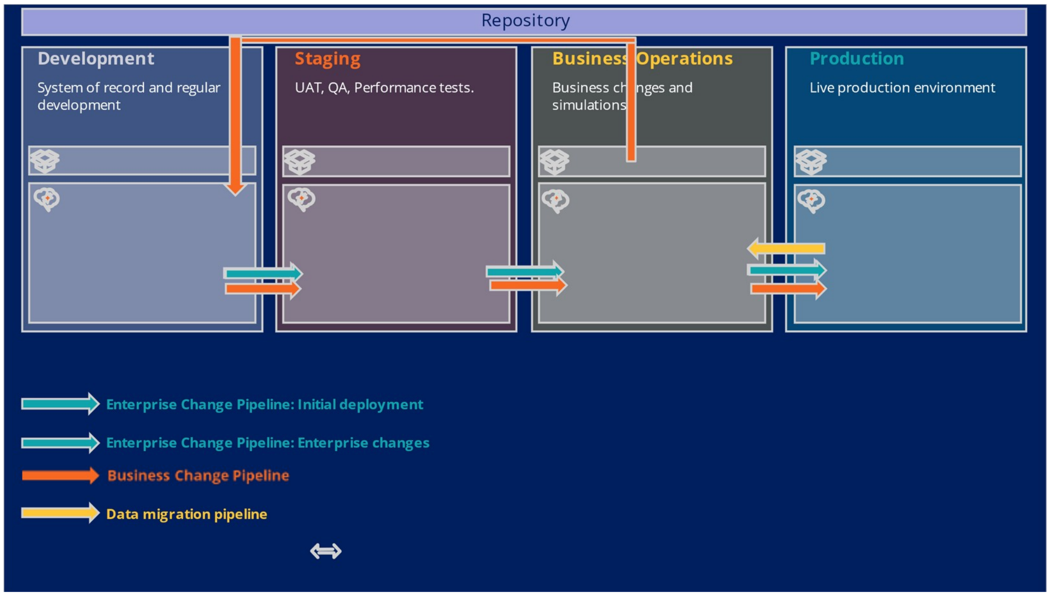

Repository
Development
CDH Standard release
CDH Fast release
Customer Decision Hub Implementation (CDH)
1:1 Operations Manager Extension (OpsMgrExt)
Pega Customer Decision Hub
DevOps
Pega 1:1 Operations Manager
Pega Platform
Staging
Business Operations
Production
Enterprise Change Pipeline: Customer Decision Hub Implementation application changes


Repository
Development
CDH Standard release
CDH Fast release
Customer Decision Hub Implementation (CDH)
1:1 Operations Manager Extension (OpsMgrExt)
Pega Customer Decision Hub
DevOps
Pega 1:1 Operations Manager
Pega Platform
Staging
Business Operations
Production
Enterprise Change Pipeline: 1:1 Operations Manager Extension application changes

As a best practice, create a new ruleset version to make enhancements to the applications. Updates to application are handled through typical developer practices such as branching and ruleset lock-and-roll.

In summary, you start with development in the development environment after running the Customer Decision Hub setup wizard. After the development process has matured and is ready for change management, use the enterprise change pipeline to deploy all development changes as part of the initial deployment. From this point, you can use the enterprise change pipeline to deploy ongoing application enhancements to higher environments to meet business objectives, such as adding a new customer property and integrating an additional channel.

Next, use the business change pipeline to deploy business changes, such as adding new actions and treatments to all other environments. Also, migrate production data from the production environment to the business operations environment through the data migration pipeline to simulate the results of the business changes to understand if the business outcomes are as expected.



Repository

Development

System of record and regular development

Staging

UAT, QA, Performance tests.

Business Operations

Business changes and simulations

Production

Live production environment

Enterprise Change Pipeline: Initial deployment

Enterprise Change Pipeline: Enterprise changes

Business Change Pipeline

Data migration pipeline

Creating an enterprise change pipeline for initial deployment

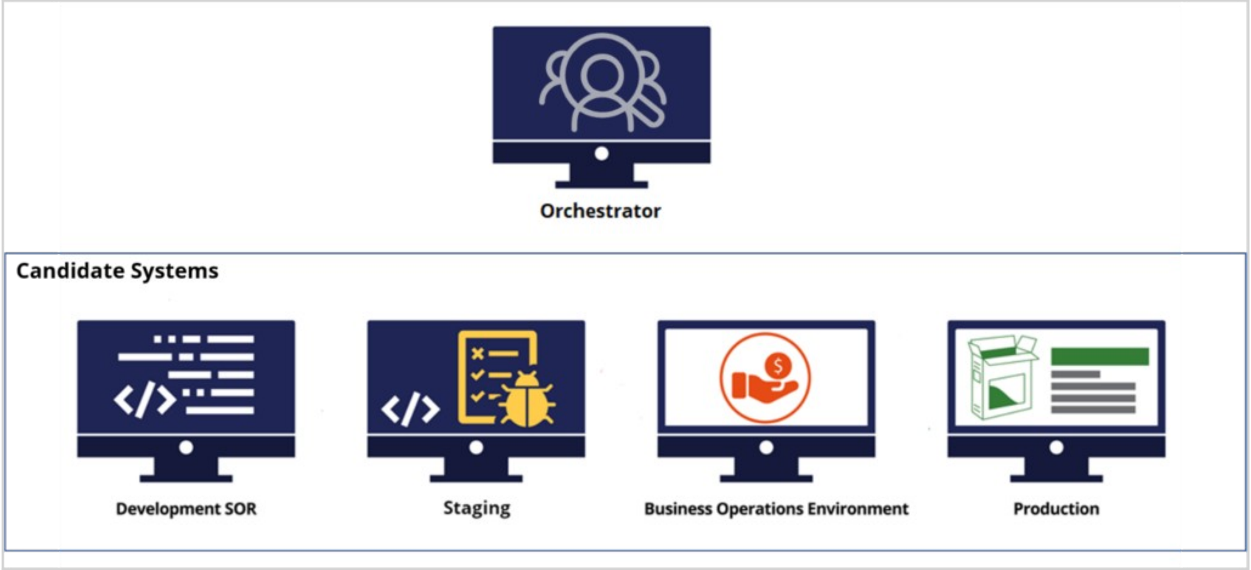
Enterprise change pipelines support initial deployment and ongoing enhancements to your Pega Customer Decision Hub™ application. This pipeline offers Pega Customer Decision Hub developers the means to deploy changes implemented to their applications as part of the enterprise release cycles.

Transcript

This demo shows you how to create an enterprise change pipeline to deploy the application, which the project team developed in the Development environment, for the first time.

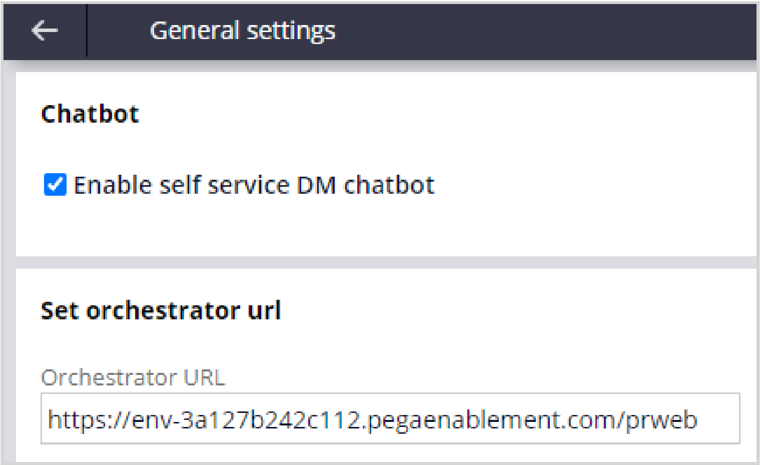
U+ Bank, a retail bank, has recently installed the Pega Deployment Manager application and wants to configure the deployment manager to create an enterprise change pipeline that facilitates initial development changes that the developers made in the implementation application.

A typical decision management project includes five environments. A **Development** environment to develop enterprise features, the system of record, a **UAT/Staging** environment to test the changes and analyze technical impact, a **Business Operations Environment** (BOE) to make business changes and run simulations, a **production** environment to serve customers with the most recent artifacts, and an **orchestrator** to manage and automate deploying artifacts from one environment to another.



In this case, you begin by setting up the orchestrator environment, and then configuring the candidate systems--Development, Staging, BOE, and Production--to communicate with the orchestrator. Finally, in the orchestrator, you define an enterprise change pipeline to facilitate the initial deployment.

This is the orchestrator. The orchestrator has the Deployment Manager and PegaDevopsFoundation application installed. To configure an environment as the orchestrator, you set the Orchestrator URL to the URL of that environment.



General settings

Chatbot

☑ Enable self service DM chatbot

Set orchestrator url

Orchestrator URL: https://env-3a127b242c112.pegaenablement.com/prweb

In a typical scenario, you update the PegaDevOpsShared dynamic system setting in the candidate systems to point to the orchestrator system URL. On the candidate systems, only the PegaDevopsFoundation application is installed along with Pega Customer Decision Hub.

Now, the candidate systems are linked to the orchestrator system.

Deploying initial development changes involves preparing the deployment and the actual deployment through an enterprise change pipeline.

As a first step, you need to have a product rule to use for packaging the artifacts in the development environment. The product rule is created for you when the CDH wizard is run.

This product rule is used to migrate various rules and data:

1. The applications, components, and rulesets
2. Rule instances such as:
    a. relevant records
    b. indexes for the containers and events
    c. Application specific settings

d. Access groups

e. Dynamic System Settings

f. data flow schedules and import jobs created in Customer Profile Designer

g. Class instances: the rules and classes that are auto generated through batch and simulation runs are automatically filtered

3. Other individual class instances that are required for Pega Customer Decision Hub to operate.

> If you add new rules as part of your implementation, you must add them manually to the product rule.

Next, you must create an enterprise change pipeline in the Deployment Manager portal in the orchestrator. With the enterprise change pipeline, you deploy the core application changes, establish communication between environments, and then define the sequence of tasks to perform in each environment.

Deployment Manager comes with pipeline templates that help you structure pipelines based on their purpose.

For this requirement, use the Deployment Pipeline template. Start by defining the **Application packaging environment**, which is typically your development environment that acts as the system of record (SOR) and contains the product rule that defines how the application is packaged. In this case, enter the development environment URL.

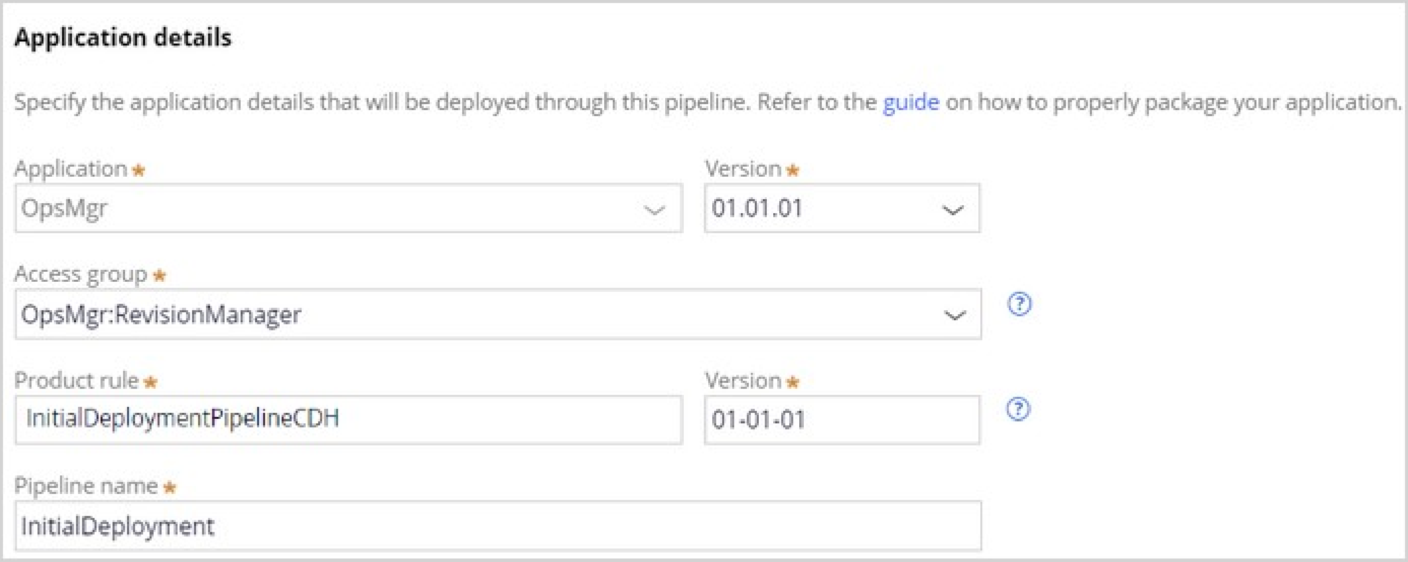
Then, select an **Authentication profile** to communicate with the candidate environments, and connect to the application packaging environment (development) to build the deployment package. An authentication profile is created when you install Deployment Manager. The orchestrator uses this authentication profile to communicate with candidate systems to run tasks in the pipeline. You must have this authentication profile in all the candidate systems with the same operator ID.

As your next step, select the name and version of the overlay application, which is the application where the business changes will reside.

Then, select the access group for which you want to run the pipeline tasks. Ensure that this access group is in all environments.

Now, enter the name and version of the product rule to use for packaging the artifacts to deploy to the other candidate systems.

Enter a unique name for the pipeline, in this case, Initial Deployment.



Application details

Specify the application details that will be deployed through this pipeline. Refer to the guide on how to properly package your application.

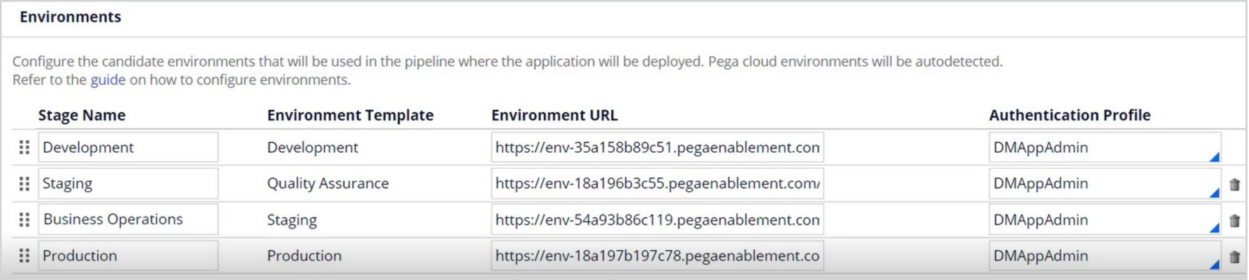
Application *	OpsMgr
Version *	01.01.01
Access group *	OpsMgr:RevisionManager
Product rule *	InitialDeploymentPipelineCDH
Version *	01-01-01
Pipeline name *	InitialDeployment

Next, configure the candidate environment details.

Notice that the development environment system URL is already populated based on the information that you provided earlier.

A generic enterprise change pipeline has Development, Quality Assurance, Staging, and Production as the candidate environments. As a typical decision management project includes Development, Staging, Business Operations, and Production, rename the candidate systems to reflect the required environments.

Enter the URLs of the other candidate systems, and then select the authentication profile for each system.

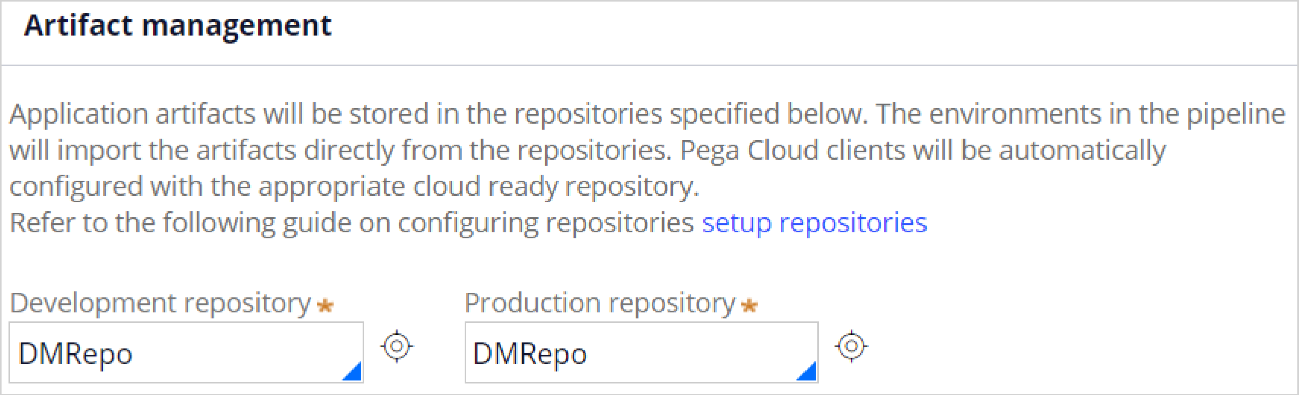


Environments

Configure the candidate environments that will be used in the pipeline where the application will be deployed. Pega cloud environments will be autodetected.
Refer to the guide on how to configure environments.

Stage Name	Environment Template	Environment URL	Authentication Profile
Development	Development	https://env-35a158b89c51.pegaenablement.con	DMAppAdmin
Staging	Quality Assurance	https://env-18a196b3c55.pegaenablement.com/	DMAppAdmin
Business Operations	Staging	https://env-54a93b86c119.pegaenablement.con	DMAppAdmin
Production	Production	https://env-18a197b197c78.pegaenablement.co	DMAppAdmin

Now, specify the development and production repositories where you want to store the packaged rules. The pipeline uses the development repository to generate artifacts and version for an application. The pipeline uses the production repository to store the production-ready artifacts.



Artifact management

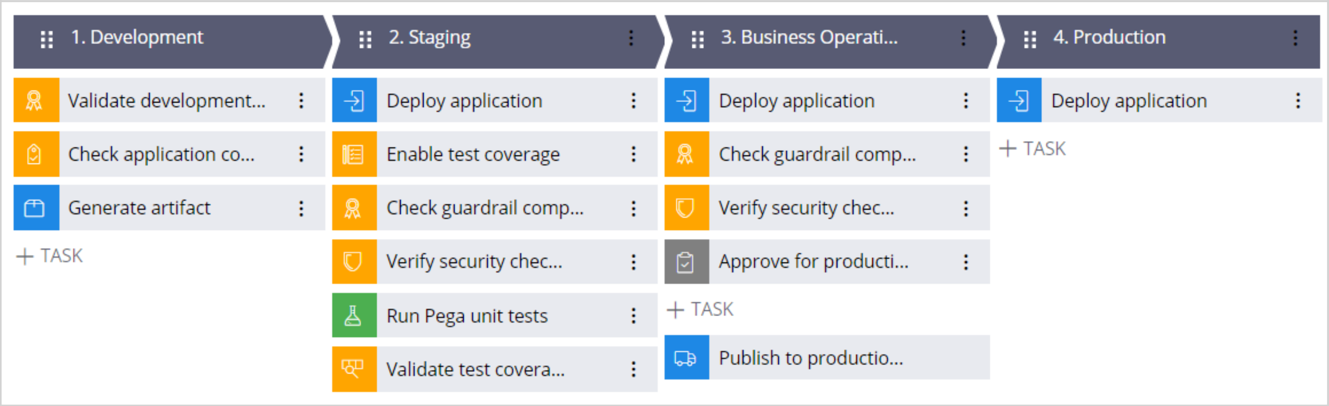
Application artifacts will be stored in the repositories specified below. The environments in the pipeline will import the artifacts directly from the repositories. Pega Cloud clients will be automatically configured with the appropriate cloud ready repository.
Refer to the following guide on configuring repositories setup repositories

Development repository * DMRepo

Production repository * DMRepo

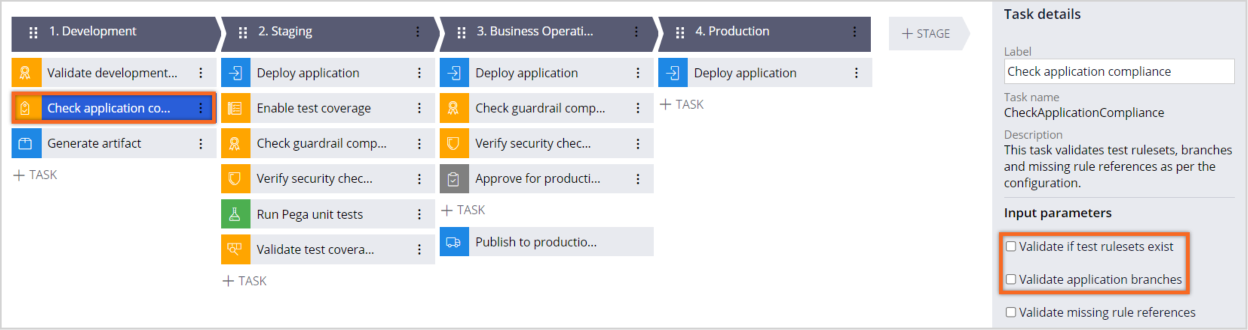
The environments in the pipeline import the application package directly from these repositories. In this case, use the same repository.

A pipeline model is created. The pipeline model is a case type that defines various predefined tasks that are necessary to accomplish a successful initial deployment.



From here, you can adjust your stages and tasks to replicate the quality standard of the release process that your organization defines.

In the **Check application compliance** step of the Development environment, clear the options in input parameters to include test rulesets and branches as some out-of-the-box test rulesets are in the application, and the overlay application branch should also be migrated.



Task details

Label

Check application compliance

Task name

CheckApplicationCompliance

Description

This task validates test rulesets, branches and missing rule references as per the configuration.

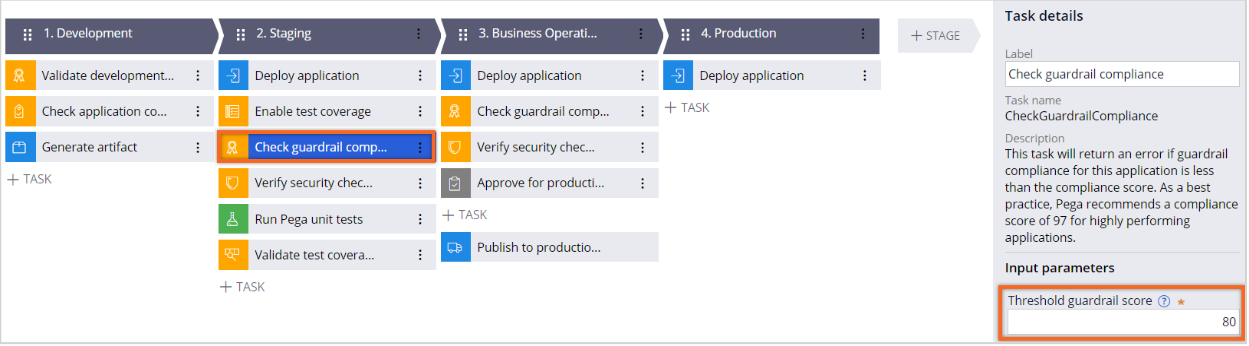
Input parameters

□ Validate if test rulesets exist

□ Validate application branches

□ Validate missing rule references

In the **Check guardrail compliance** step of the Staging environment, enter a guardrail threshold score that is less than your application score. For high performance, Pega recommends a high guardrail score for applications. In this case, as the application guardrail score is 84, a lower score of 80 is set as the threshold.



Task details

Label

Check guardrail compliance

Task name

CheckGuardrailCompliance

Description

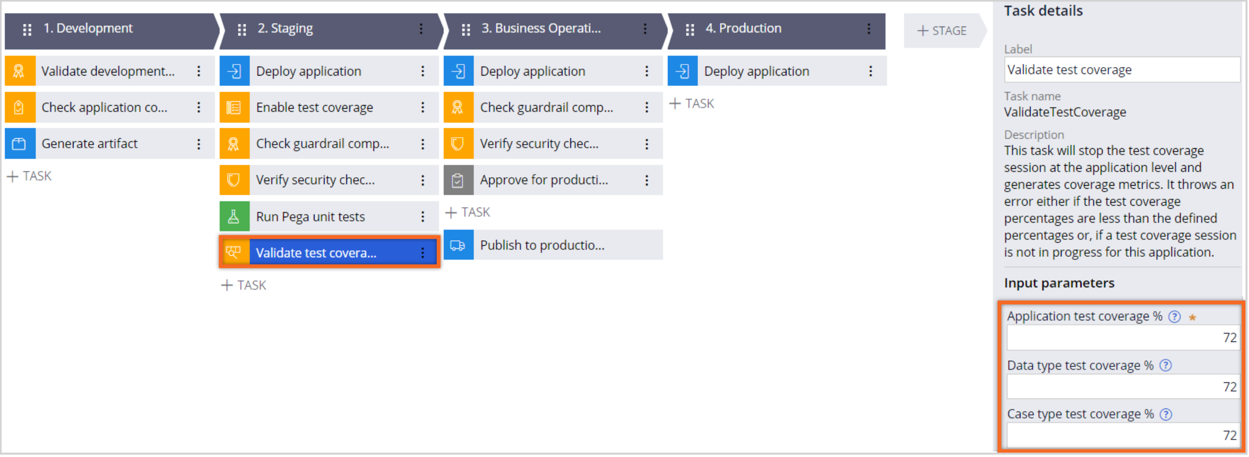
This task will return an error if guardrail compliance for this application is less than the compliance score. As a best practice, Pega recommends a compliance score of 97 for highly performing applications.

Input parameters

Threshold guardrail score ⓘ *

80

In the **Validate test coverage** step, enter the percentage of test coverage for the application, data types, and case types. This case has no test coverage. Consequently, the value is set to zero.



Task details

Label: Validate test coverage

Task name: ValidateTestCoverage

Description: This task will stop the test coverage session at the application level and generates coverage metrics. It throws an error either if the test coverage percentages are less than the defined percentages or, if a test coverage session is not in progress for this application.

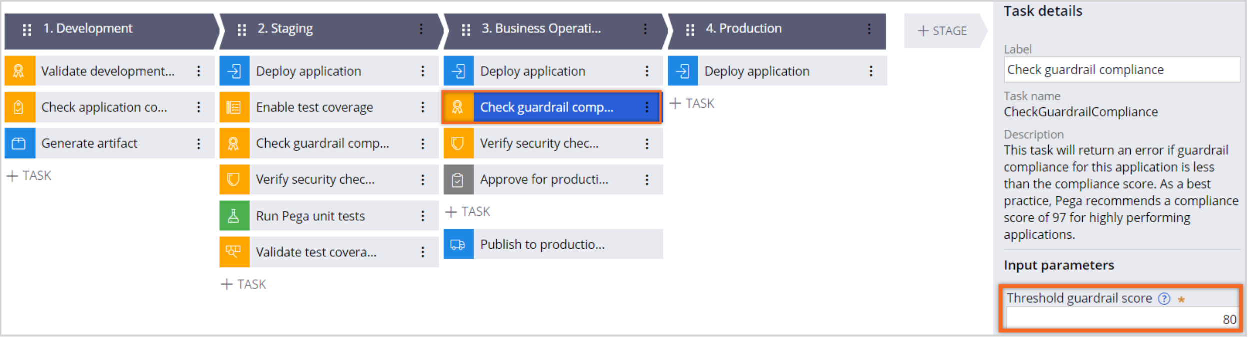
Input parameters

Application test coverage % ⓘ ★: 72

Data type test coverage % ⓘ: 72

Case type test coverage % ⓘ: 72

In the **Check guardrail compliance** step of the Business Operations environment, set the guardrail score as you did in the Staging stage.



Task details

Label: Check guardrail compliance

Task name: CheckGuardrailCompliance

Description: This task will return an error if guardrail compliance for this application is less than the compliance score. As a best practice, Pega recommends a compliance score of 97 for highly performing applications.

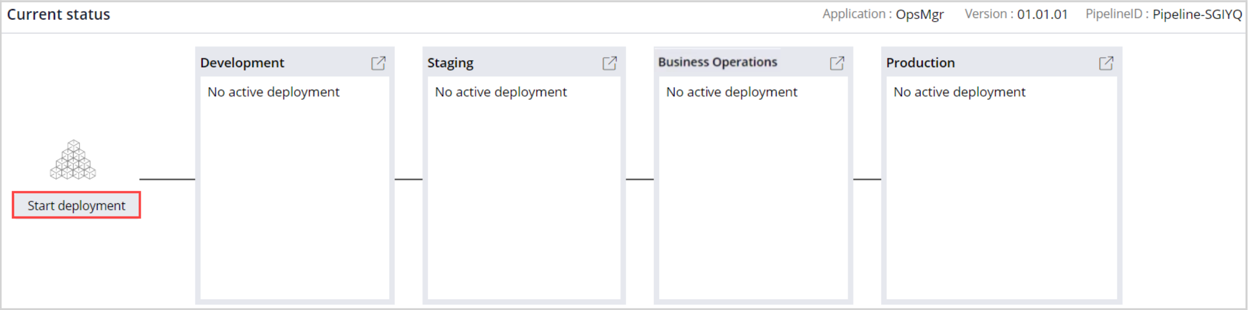
Input parameters

Threshold guardrail score ⓘ ★: 80

These adjustments to the stages and steps ensure that the default settings do not affect the application package creation and deployment.

Run diagnostics to check whether the orchestrator can communicate with all candidate systems and whether the pipeline details are defined correctly. If the check is successful, you can use the pipeline for the initial deployment.

The Release manager initiates the deployment process by using the enterprise change pipeline that is created in the Deployment Manager in the orchestrator environment.


Current status

Application : OpsMgr   Version : 01.01.01   PipelineID : Pipeline-SGIYQ

Development: No active deployment

Staging: No active deployment

Business Operations: No active deployment

Production: No active deployment

Deployment Manager orchestrates the process, and then the candidate environments run the tasks in the pipeline model. The application is packaged and sent to the repository to make them available to other environments that are defined in the pipeline.

You can see the progress of the deployment and each completed task. Monitoring the process in Deployment Manager helps in debugging.

Notice that the process indicates an action that the user needs to complete.

When the deployment is successful, you can view the report in the **Deployment history** section to check the process details and logs.

You have reached the end of this demo. What did it show you?

- How to create an enterprise change pipeline and establish communication between environments using the pipeline.
- How to facilitate an initial deployment using an enterprise change pipeline.
- How the core application changes are deployed from the development environment to the higher environments.

Business change pipeline

Description

Learn how a business change pipeline facilitates change management in Pega Customer Decision Hub™ applications by promoting business changes from the business operations environment (BOE) to other environments without the constraints of enterprise release cycles.

Learning objectives

- Define a business change pipeline
- Configure a business change pipeline

Business change pipeline

Business change pipelines support business changes to your Pega Customer Decision Hub™ application. This pipeline offers business users of the Customer Decision Hub the means to test and deploy changes that they implement to their applications outside the enterprise release cycles.

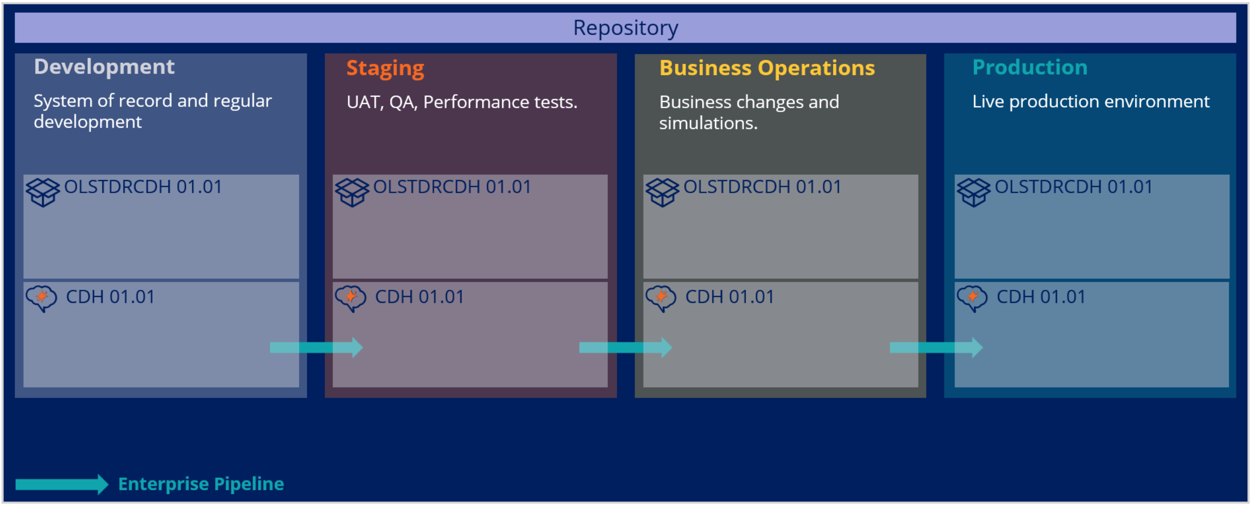
Transcript

In this video, you will understand how a business change pipeline facilitates change management in Customer Decision Hub applications by promoting business changes from the business operations environment (BOE) to other environments in the project.

A typical decision management project includes five environments. An orchestrator manages and automates the deployment process of artifacts to the candidate environments. A development (System of Record) environment develops enterprise capabilities. A staging environment tests the changes and analyzes their technical impact. A business operations environment (BOE) makes business changes. A production system provides real customers with the latest and greatest artifacts.

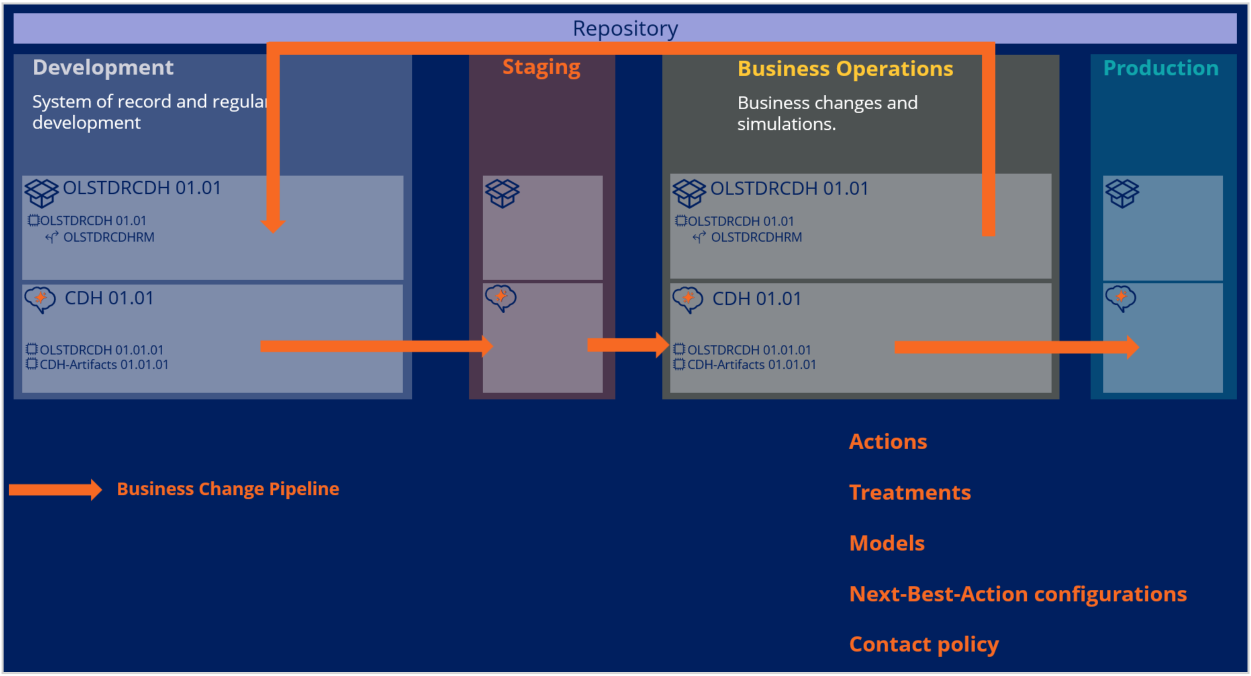
All application pipelines use a repository to move generated artifacts from one environment to another.

In a new project, you set up enterprise capabilities and core structure of the Customer Decision Hub application in the development environment. When the core development comes to a mature state and it is tested, you then deploy the application to all other environments.



When the initial deployment is complete, all environments are in sync and have the same application ruleset stack and versions. After this point, business users can start making changes to the next-best-action artifacts in the BOE and deploy these changes to other environments through the business change pipeline.

Ensure that you make ongoing changes to next-best-action artifacts, such as changes to actions, treatments, models, next-best-action designer configurations, contact policy, and similar artifacts in the business domain, in the BOE. The changes are managed through revisions and are packaged automatically for you when you use 1:1 Operations Manager and Revision Management.



Here is an example of how a business change is initiated from the BOE and then promoted to the other environments.

Consider UBank, a retail bank, that has five environments, including one orchestrator and four candidate systems, to deploy changes through the business change pipeline.

In the development environment, note that **OLSTDRCDH** is the overlay application. The application is built on **CDH 01.01** (the enterprise application) with the OLSTDRCDH ruleset and the CDH-Artifacts ruleset. The current version is **01-01-01**. **OLSTDRCDHRM** is a branch of the **OLSTDRCDH** ruleset. This is branch to which you merge business changes. At this point, the staging, BOE, and production environments are all in sync.

After the initial deployment is complete, the change management starts, and business users have completed their initial sprint. All the development is stored in the **OLSTDRCDHRM** branch in the BOE. When the Revision Manager deploys the change through the business change pipeline, the rules in the branch are packaged and moved to the repository to which all the environments have access.

For example, when the business content team on the UBank creates a new action in the 1:1 Operations Manager application in the BOE, the rules that are associated with the action are added to the branch. When the Revision Manager deploys the change, the rules from the branch are packaged and moved to the repository.

Then, the artifacts that are related to the new revision are imported and merged into the development system. Then, a new version of the enterprise application is created with the merged rules. A new application package is created for the enterprise application and sent back to the repository. This package is created from a predefined product rule, which is configured as part of the business change pipeline.

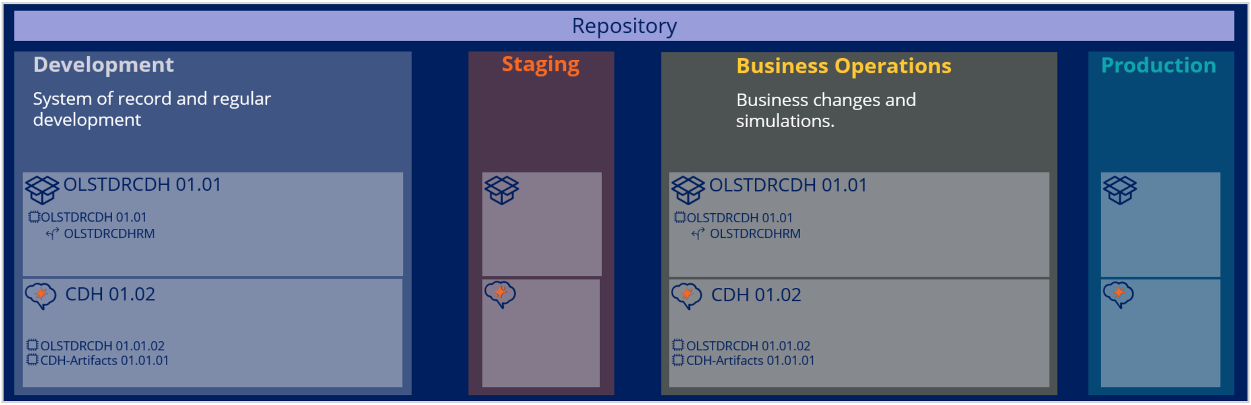
As the next step, the application package is deployed from the repository to the staging environment, and the enterprise and the overlay application versions are updated.

Next, the application package is deployed from the repository to the BOE, and the enterprise and the overlay application versions are updated.

Finally, the application package is deployed from the repository to the production environment, and the enterprise and the overlay ruleset versions are updated.

In the production environment, the change is activated in two stages. In the first stage, the production environment application ruleset stack version is updated, but a user action is required before the change can be deployed. The revision manager logs in to the BOE to enable a set of users who can view business changes. If the test users confirm that the changes are fine, you can make the actual change available to all other users. You achieve this two-stage deployment by using a second set of access group that is prefixed with rollback. During consecutive deployments, the original access group and the rollback access group are rotated.

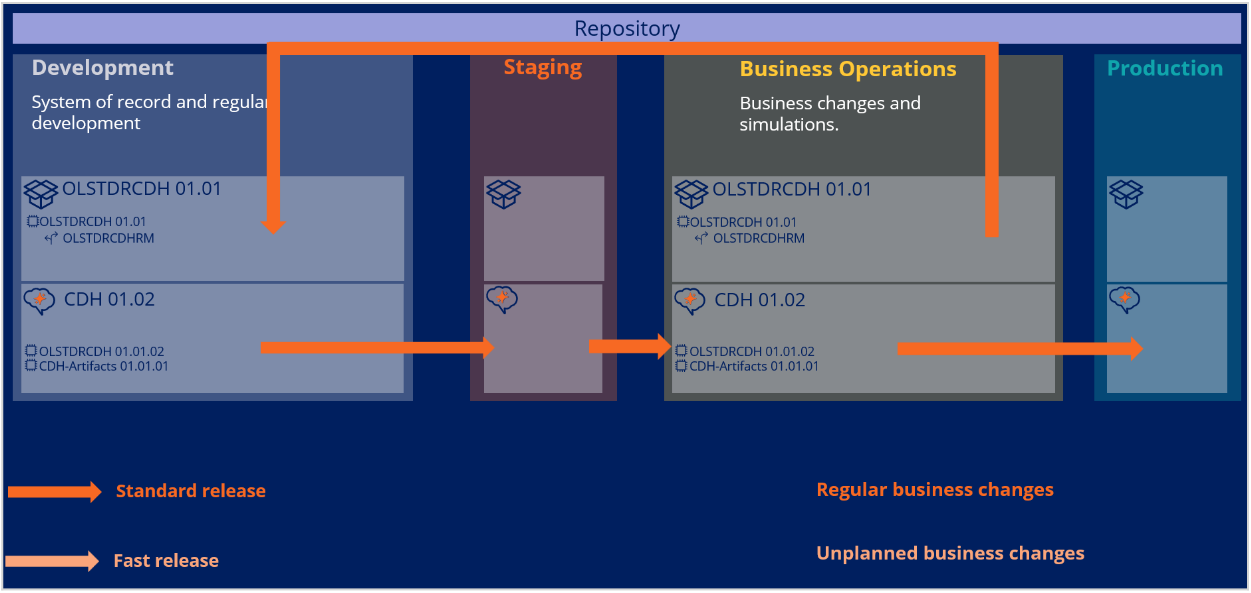
Once the deployment through the business change pipeline is successful, the new version of the enterprise application with the new version of the overlay application ruleset is available in all the environments.



Typically, business changes can be one of two types: standard release and fast release revisions.

Regular business changes consist of one or more change requests. After individually implementing these change requests, when they are ready for deployment, the Revision Manager deploys all the change requests together as a release. This is called a standard release.

However, you might need to implement some unplanned business changes that are of high priority. These deployments are called fast releases.



Examples of fast release change requests include:

1. Fixing a typo in the action that was deployed to production (updating an existing action).
2. Deactivating an action from production (updating an existing action).
3. Communicating with customers in an emergency (creating a new action).

With fast release revisions, the business content team can implement and deploy critical business changes without having to wait for other change requests to be implemented and ready for deployment.

As a best practice, create two business change pipelines: one for the standard release and another for the fast release.

Configuring a business change pipeline

Business change pipelines support business changes to your Pega Customer Decision Hub™ application. This pipeline type offers Pega Customer Decision Hub business users the means to deploy changes implemented to their applications outside of the enterprise release cycles.

Additionally, users can use the pipeline to respond to changing business requirements by modifying and deploying application rules in a controlled manner. Responses can include any business changes in the overlay applications.

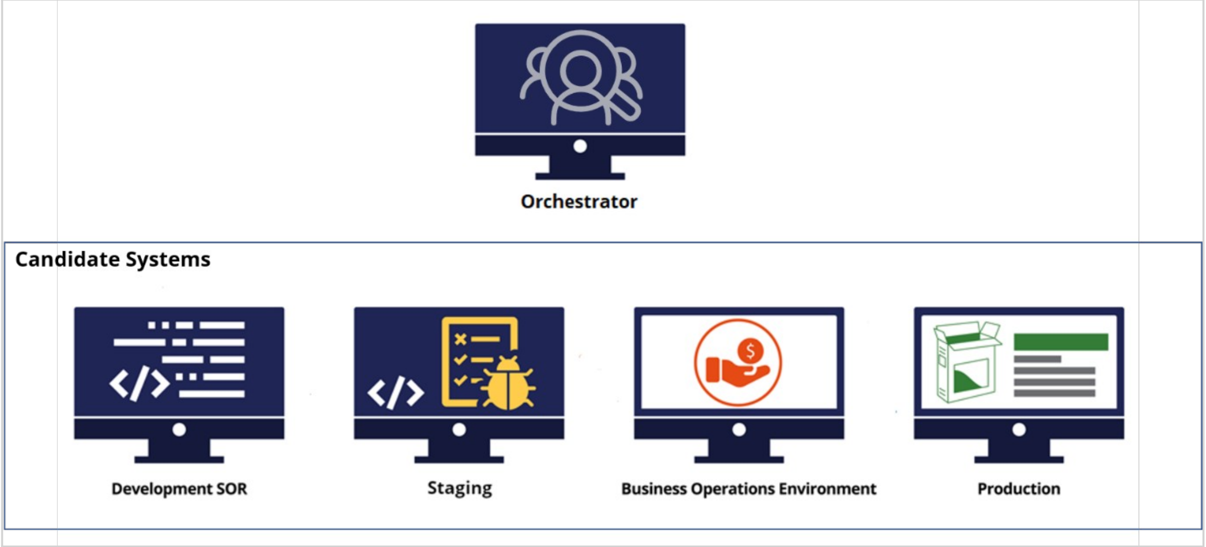
Standard release business change pipeline

Transcript

This demo shows you how to configure a business change pipeline to deploy changes that are implemented by the business content team and enterprise capabilities team in the Business Operations environment.

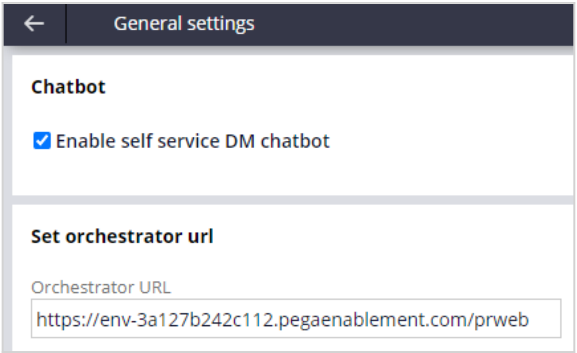
U+ Bank, a retail bank, has recently installed the Pega Deployment Manager application and wants to configure Deployment Manager to set up a business change pipeline so that the business team can efficiently implement business changes outside of the enterprise release cycle for faster market response.

A typical decision management project has five environments involved. An orchestrator to manage and automate the deployment process of artifacts to the candidate environments. A development (System of Record) environment to develop enterprise capabilities. A staging environment to test the changes and analyze their technical impact. A Business Operations Environment (BOE) to make business changes. A production system to serve real customers with the latest and greatest artifacts.



In this case, you first set up the orchestrator environment, then configure the candidate systems (Development, Staging, BOE, and Production) to communicate with the orchestrator. Finally, in the orchestrator, you define a business change pipeline to define the deployment process of a business change.

This is the orchestrator. The orchestrator has the Deployment manager and PegaDevopsFoundation application installed. To configure an environment as the orchestrator, you set the *OrchestratorURL* to the URL of that environment.



Chatbot

☑ Enable self service DM chatbot

Set orchestrator url

Orchestrator URL

https://env-3a127b242c112.pegaenablement.com/prweb

In a typical scenario, you update the PegaDevOpsShared dynamic system setting in all the candidate systems to point to the orchestrator system URL. On the candidate systems, only the PegaDevopsFoundation application is installed along Pega Customer Decision Hub and overlay applications.

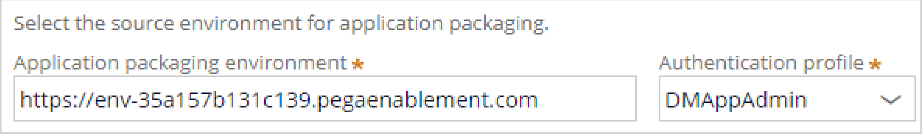
Now, all the candidate systems are linked to the orchestrator system.

Next, you need to configure a business change pipeline in the Deployment Manager portal in the orchestrator. With the business change pipeline, you can deploy the implemented changes by establishing communication between environments and defining the sequence of tasks to perform in each environment.

Deployment Manager comes with pipeline templates that help you structure pipelines based on their purpose.

For this requirement, use the Business Change Pipeline template. Start by defining the **Application packaging environment**. This is typically your development environment that acts as the system of record (SOR) and contains the product rule that defines how the application is packaged. In this case, enter the development environment URL.

Then, select an **Authentication profile** to communicate with the candidate environments and connect to the application packaging environment (development) to build the deployment package. An authentication profile is automatically created when you install Deployment Manager. The orchestrator uses this authentication profile to communicate with candidate systems to run tasks in the pipeline. You must have this authentication profile in all the candidate systems with the same operator ID.



Select the source environment for application packaging.

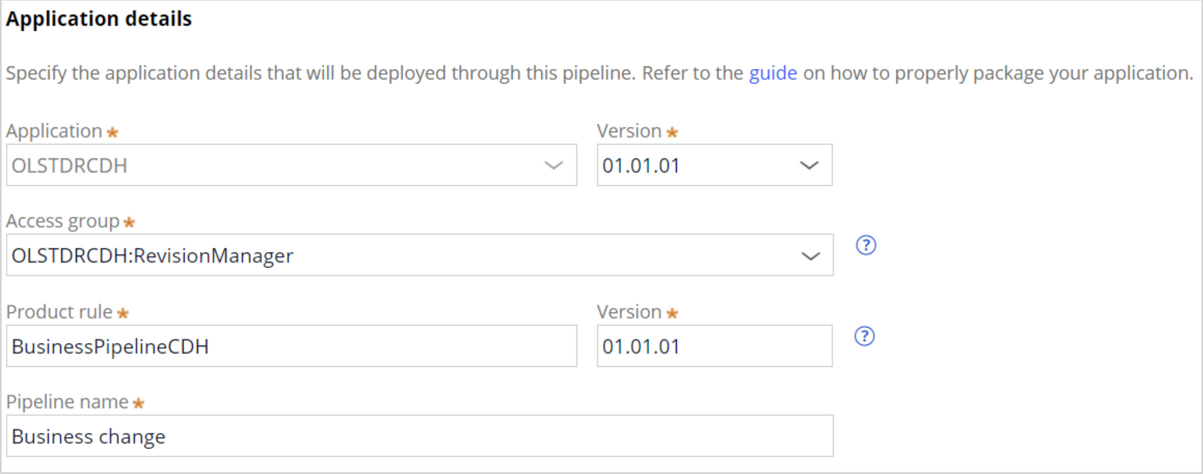
Application packaging environment *	Authentication profile *
https://env-35a157b131c139.pegaenablement.com	DMAppAdmin

As your next step, select the name and version of the overlay standard release application, which is the application where business changes reside.

Then, select the access group for which you want to run the pipeline tasks. Ensure that this access group is present in all environments.

Now, enter the name and version of the product rule to use for packaging the artifacts to deploy to the other candidate systems.

Enter a unique name for the pipeline, in this case, *Business Change*.

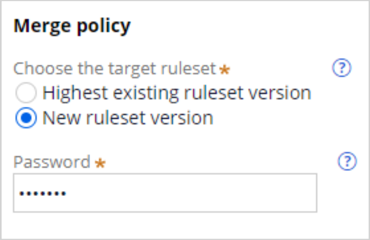


Application details

Specify the application details that will be deployed through this pipeline. Refer to the guide on how to properly package your application.

Application *	Version *
OLSTDRCDH	01.01.01
Access group *	
OLSTDRCDH:RevisionManager	
Product rule *	Version *
BusinessPipelineCDH	01.01.01
Pipeline name *	
Business change	

Then, select the **New ruleset version** option and provide a password. This action creates and locks a new ruleset version that contains business changes after a successful merge.



Merge policy

Choose the target ruleset *

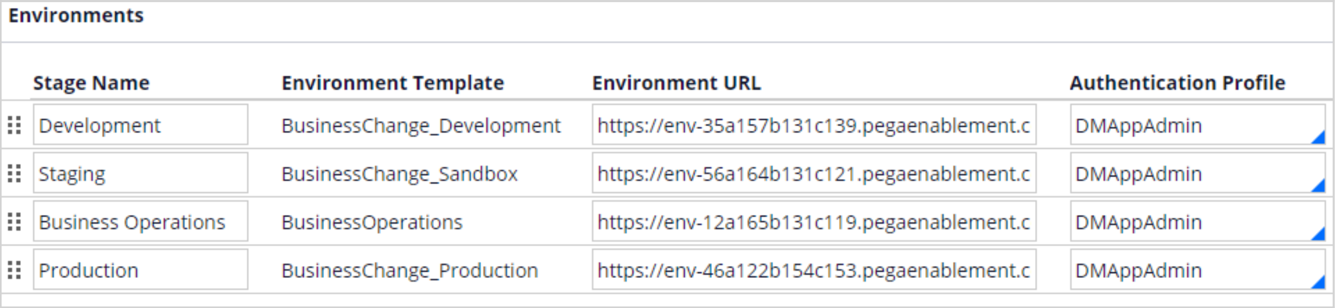
○ Highest existing ruleset version
● New ruleset version

Password *

•••••••

Next, configure the candidate environment details.

Notice that the development environment system URL is already populated based on the information that you provided earlier. Enter the URLs of all the other candidate systems, and then select the authentication profile for each system.

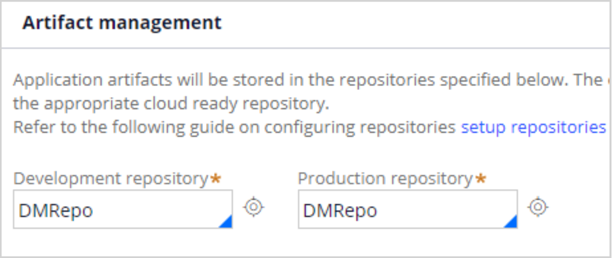


Environments

Stage Name	Environment Template	Environment URL	Authentication Profile
Development	BusinessChange_Development	https://env-35a157b131c139.pegaenablement.c	DMAppAdmin
Staging	BusinessChange_Sandbox	https://env-56a164b131c121.pegaenablement.c	DMAppAdmin
Business Operations	BusinessOperations	https://env-12a165b131c119.pegaenablement.c	DMAppAdmin
Production	BusinessChange_Production	https://env-46a122b154c153.pegaenablement.c	DMAppAdmin

Now, specify the development and production repositories where in which you want to store the packaged rules. Define at least one repository that each candidate system can access.
The pipeline uses the development repository to generate artifacts and version for an application. The pipeline uses the production repository to store the production-ready artifacts. Having a separate production repository is optional.

The environments in the pipeline directly import the application package from these repositories. In this case, use the same repository.



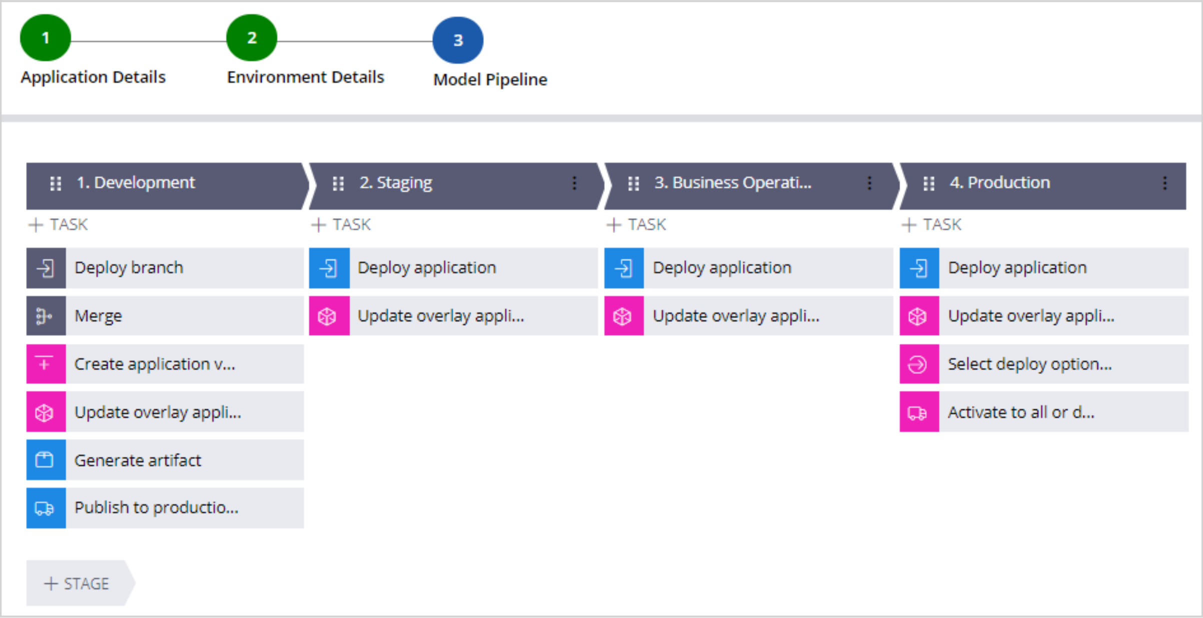
Artifact management

Application artifacts will be stored in the repositories specified below. The
the appropriate cloud ready repository.
Refer to the following guide on configuring repositories setup repositories

Development repository *: DMRepo

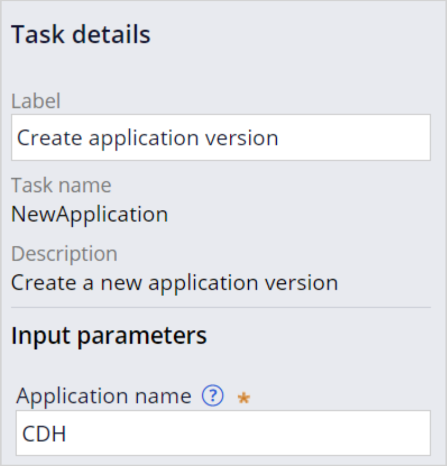
Production repository *: DMRepo

A pipeline model is created. The pipeline model is a case type that defines various predefined tasks that are necessary to accomplish a successful business change deployment.



From here, you can adjust your stages and tasks to replicate the quality standard of the release process that is defined by your organization.

In the **Create application version** task, enter the enterprise application name for the input parameter. After a business change branch is merged, a new version of the enterprise application is created which is then deployed to other environments. Create the pipeline.



Task details

Label
Create application version

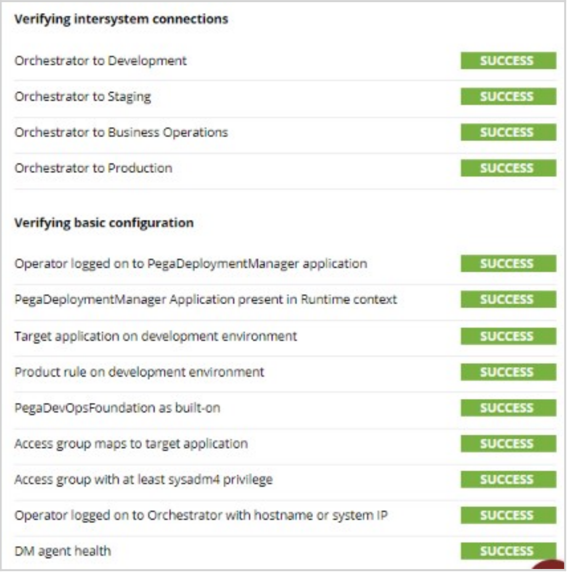
Task name
NewApplication

Description
Create a new application version

Input parameters

Application name ? *
CDH

Run diagnostics to check whether the orchestrator can communicate with all the candidate systems and whether the pipeline details are defined correctly. If the check is successful, you can use the pipeline for deployment of overlay application changes from the BOE.

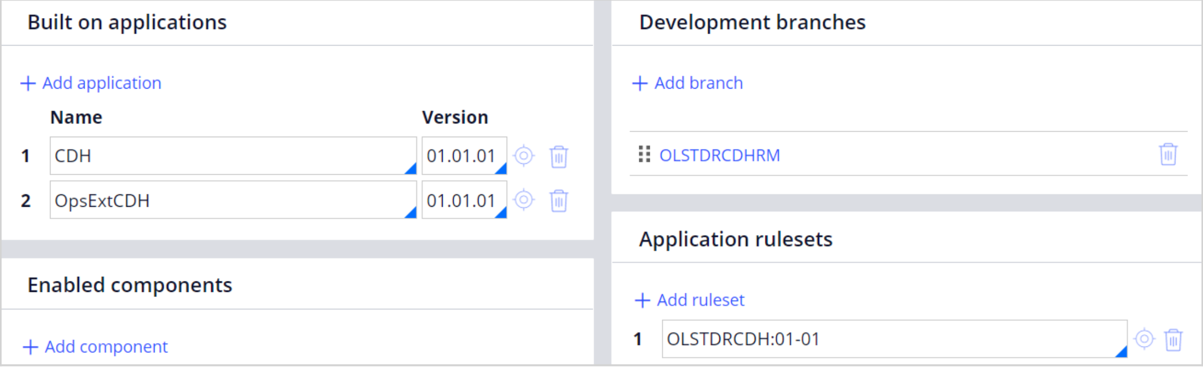


Verifying intersystem connections	
Orchestrator to Development	SUCCESS
Orchestrator to Staging	SUCCESS
Orchestrator to Business Operations	SUCCESS
Orchestrator to Production	SUCCESS
Verifying basic configuration	
Operator logged on to PegaDeploymentManager application	SUCCESS
PegaDeploymentManager Application present in Runtime context	SUCCESS
Target application on development environment	SUCCESS
Product rule on development environment	SUCCESS
PegaDevOpsFoundation as built-on	SUCCESS
Access group maps to target application	SUCCESS
Access group with at least sysadm4 privilege	SUCCESS
Operator logged on to Orchestrator with hostname or system IP	SUCCESS
DM agent health	SUCCESS

In the development environment, navigate to the overlay application to view the current state of the application stack and versions.

OLSTDRCH is the overlay application, which is built on CDH 01.01.01 (the enterprise application) and Ops Manager Extension. Note that the patch version should not be provided in the application ruleset version and should only include the minor version 01-01 .

The **OLSTDRCDHRM** is a branch of the **OLSTDRCDH** ruleset. The current version is **01-01-01**. This branch is where you create business changes.



Built on applications

+ Add application

	Name	Version
1	CDH	01.01.01
2	OpsExtCDH	01.01.01

Enabled components

+ Add component

Development branches

+ Add branch

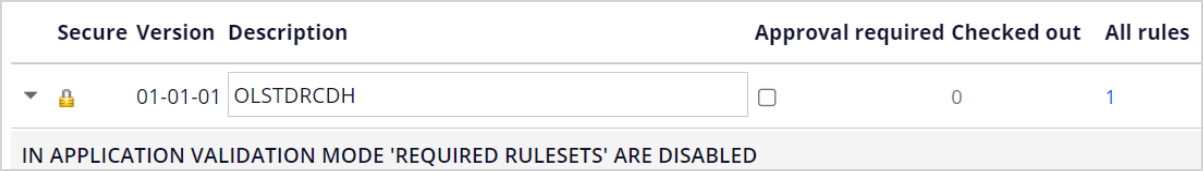
OLSTDRCDHRM

Application rulesets

+ Add ruleset

1 OLSTDRCDH:01-01

When you deploy a revision, a new version of the enterprise application is created with an incremented application version. This new enterprise application version points to the merged overlay application version and a new ruleset version is available in the overlay application with the new set of rules added. Currently, there is 1 rule in the **OLSTDRCDH** ruleset.



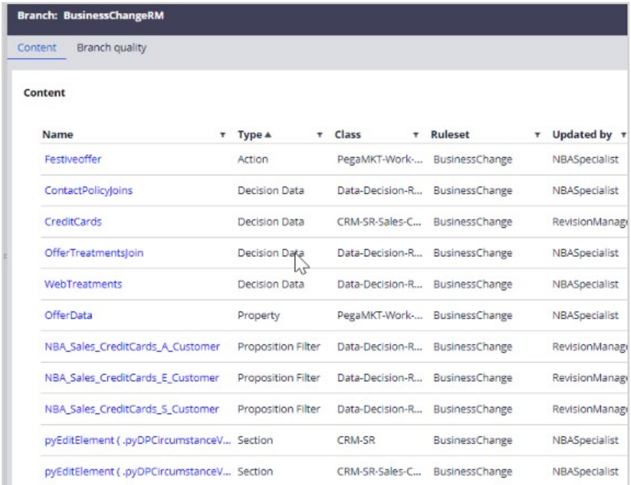
	Secure	Version	Description	Approval required	Checked out	All rules
▾	🔒	01-01-01	OLSTDRCDH	□	0	1

IN APPLICATION VALIDATION MODE 'REQUIRED RULESETS' ARE DISABLED

Now, in the BOE, open the overlay application ruleset to view the rules within the branch. Currently, no rules need to be deployed because a new revision is not created yet.

Let us consider a regular business change use case where business requested the addition of a new action: Cashback 15 card. This is a typical business change, and the business content team starts working on this request in the BOE system. A business user first creates a change request in Pega 1:1 Operations Manager to initiate adding a new action. Then, a team leader prioritizes the change request and assigns the request to an NBA Specialist for build. Next, the NBA Specialist completes the action build tasks. The Team leader then confirms the changes and sets the change request ready for deployment. The business change is finally complete.

Review the current state of the overlay application in the BOE system before proceeding with the deployment. Open the overlay application ruleset to view the rules within the branch. All changes that were made to the rules are stored in the branch ruleset. Note that a new action with the name **Cashback15card** was added.

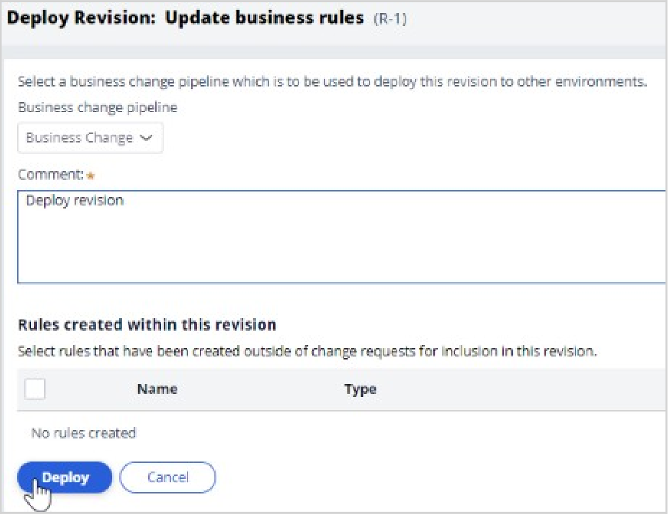


Branch: BusinessChangeRM

Content

Name	Type ▲	Class	Ruleset	Updated by
Festiveoffer	Action	PegaMKT-Work-...	BusinessChange	NBASpecialist
ContactPolicyJoins	Decision Data	Data-Decision-R...	BusinessChange	NBASpecialist
CreditCards	Decision Data	CRM-SR-Sales-C...	BusinessChange	RevisionManag
OfferTreatmentsJoin	Decision Data	Data-Decision-R...	BusinessChange	NBASpecialist
WebTreatments	Decision Data	Data-Decision-R...	BusinessChange	NBASpecialist
OfferData	Property	PegaMKT-Work-...	BusinessChange	NBASpecialist
NBA_Sales_CreditCards_A_Customer	Proposition Filter	Data-Decision-R...	BusinessChange	RevisionManag
NBA_Sales_CreditCards_E_Customer	Proposition Filter	Data-Decision-R...	BusinessChange	RevisionManag
NBA_Sales_CreditCards_S_Customer	Proposition Filter	Data-Decision-R...	BusinessChange	RevisionManag
pyEditElement (.pyDPCircumstanceV...	Section	CRM-SR	BusinessChange	NBASpecialist
pyEditElement (.pyDPCircumstanceV...	Section	CRM-SR-Sales-C...	BusinessChange	NBASpecialist

The Revision manager now initiates the deployment process by using the business change pipeline that is configured in the Deployment Manager in the orchestrator environment.



Deploy Revision: Update business rules (R-1)

Select a business change pipeline which is to be used to deploy this revision to other environments.

Business change pipeline: Business Change

Comment: * Deploy revision

Rules created within this revision

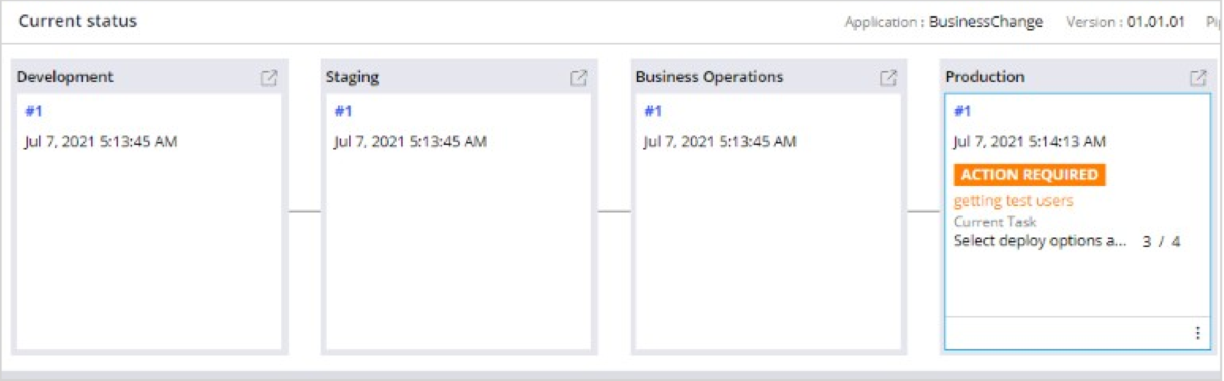
Select rules that have been created outside of change requests for inclusion in this revision.

□	Name	Type
No rules created		

The rules in the branch are packaged and sent to the repository to make them available to other environments that are defined in the pipeline.

Deployment Manager starts orchestrating the process and the candidate environments run the tasks that are defined in the pipeline model. You can see the progress of the deployment and each completed task. The default way to monitor the process is in the Revision Management portal. However, monitoring the process in Deployment Manager also helps in debugging.

Notice that the process indicates an action the user needs to complete.



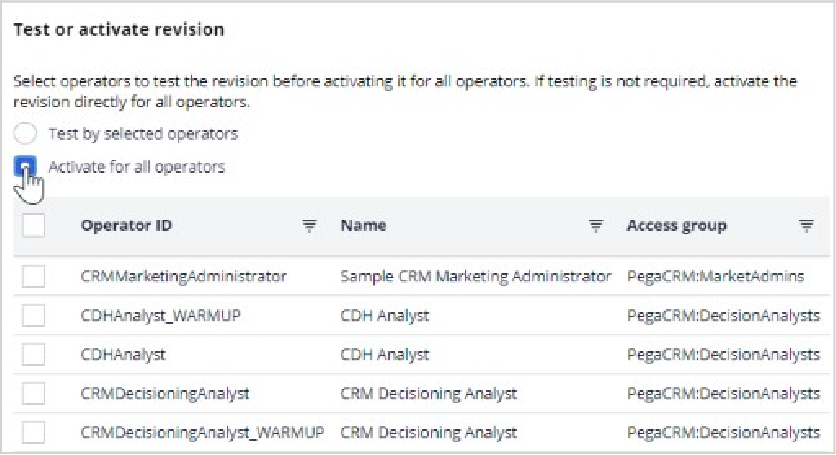
Current status

Application : BusinessChange   Version : 01.01.01   Pi

Development	Staging	Business Operations	Production
#1 Jul 7, 2021 5:13:45 AM	#1 Jul 7, 2021 5:13:45 AM	#1 Jul 7, 2021 5:13:45 AM	#1 Jul 7, 2021 5:14:13 AM ACTION REQUIRED getting test users Current Task Select deploy options a... 3 / 4

After the new application version is migrated to all environments, within the BOE environment, select a subset of users to test the modifications in the production environment and roll out the changes to all users when the test results are acceptable.

In this case, select all the users.



Test or activate revision

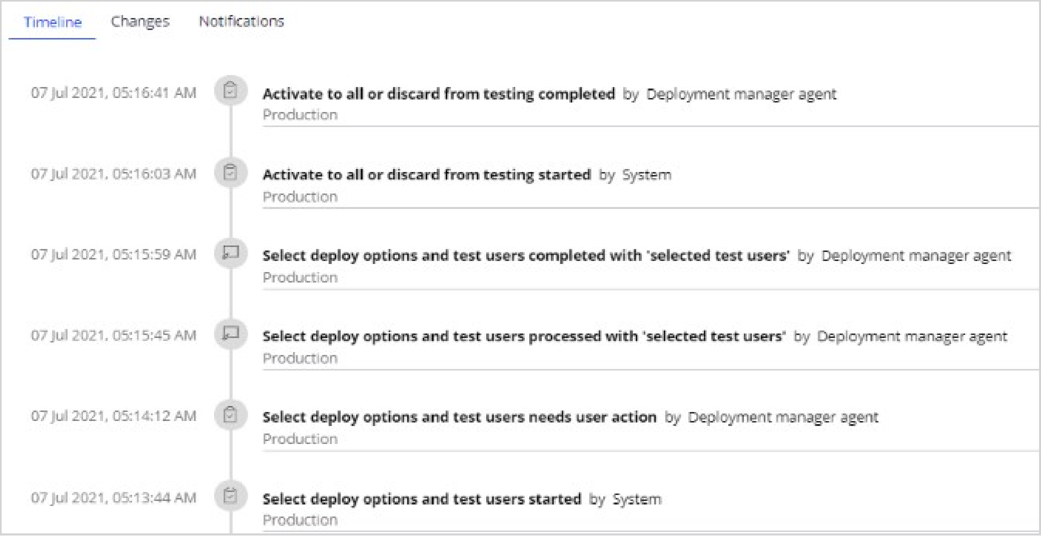
Select operators to test the revision before activating it for all operators. If testing is not required, activate the revision directly for all operators.

○ Test by selected operators

☑ Activate for all operators

□	Operator ID	Name	Access group
□	CRMMarketingAdministrator	Sample CRM Marketing Administrator	PegaCRM:MarketAdmins
□	CDHAnalyst_WARMUP	CDH Analyst	PegaCRM:DecisionAnalysts
□	CDHAnalyst	CDH Analyst	PegaCRM:DecisionAnalysts
□	CRMDecisioningAnalyst	CRM Decisioning Analyst	PegaCRM:DecisionAnalysts
□	CRMDecisioningAnalyst_WARMUP	CRM Decisioning Analyst	PegaCRM:DecisionAnalysts

When the deployment is successful, you can view the report in the **Deployment history** section to check the process details and logs.



07 Jul 2021, 05:16:41 AM — **Activate to all or discard from testing completed** by Deployment manager agent
Production

07 Jul 2021, 05:16:03 AM — **Activate to all or discard from testing started** by System
Production

07 Jul 2021, 05:15:59 AM — **Select deploy options and test users completed with 'selected test users'** by Deployment manager agent
Production

07 Jul 2021, 05:15:45 AM — **Select deploy options and test users processed with 'selected test users'** by Deployment manager agent
Production

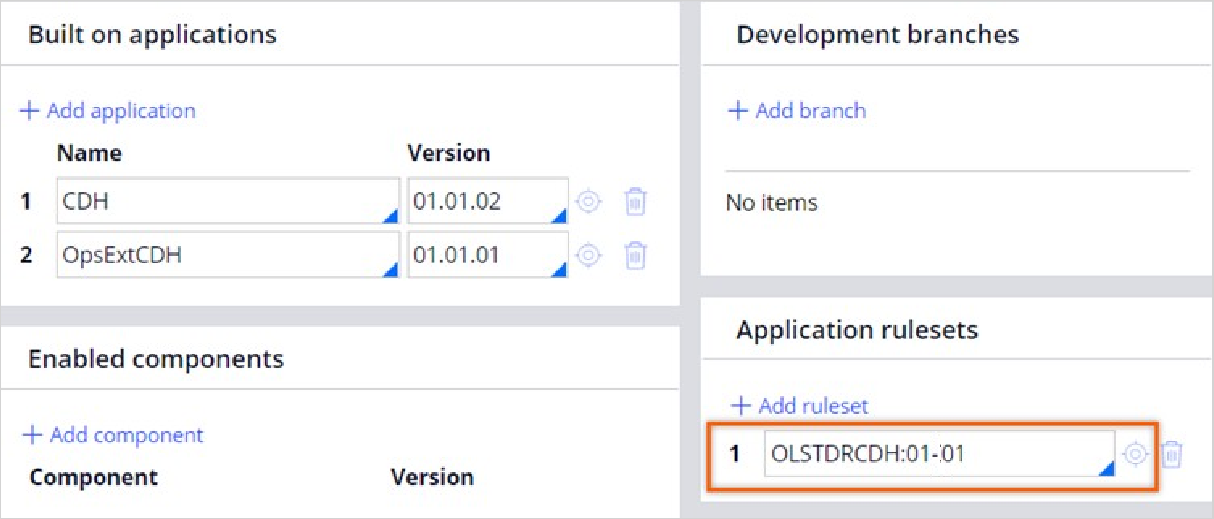
07 Jul 2021, 05:14:12 AM — **Select deploy options and test users needs user action** by Deployment manager agent
Production

07 Jul 2021, 05:13:44 AM — **Select deploy options and test users started** by System
Production

After activating the change for all users and completing the revision, the change is available to all users.

At this point, in the development environment, navigate to the overlay application to view the current state of the application stack and versions. Note that a new enterprise application was created with version 01.01.02 and the OLSTDRCDH application now inherits from this application. This new enterprise application has the merged ruleset version

(OLSTDRCDH:01-01-02).



Built on applications

	Name	Version
1	CDH	01.01.02
2	OpsExtCDH	01.01.01

Development branches

No items

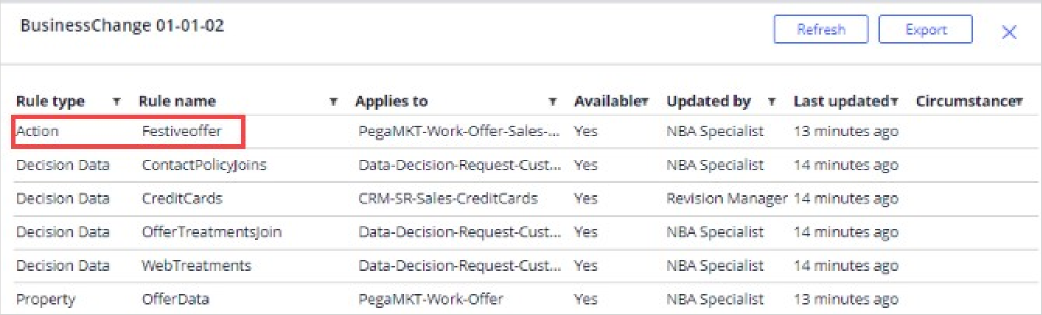
Enabled components

Component	Version

Application rulesets

1 OLSTDRCDH:01-01

Open the ruleset to view the new ruleset version and the rules available. According to the initial pipeline configuration, the branch was merged as a new ruleset and is locked. Open the latest ruleset to view the new action that was added. Notice that this ruleset also contains all the rules that were added in the BOE system branching ruleset and is now part of the development system and the other candidate systems after deployment.



BusinessChange 01-01-02

Rule type	Rule name	Applies to	Available	Updated by	Last updated	Circumstance
Action	Festiveoffer	PegaMKT-Work-Offer-Sales-...	Yes	NBA Specialist	13 minutes ago	
Decision Data	ContactPolicyJoins	Data-Decision-Request-Cust...	Yes	NBA Specialist	14 minutes ago	
Decision Data	CreditCards	CRM-SR-Sales-CreditCards	Yes	Revision Manager	14 minutes ago	
Decision Data	OfferTreatmentsJoin	Data-Decision-Request-Cust...	Yes	NBA Specialist	14 minutes ago	
Decision Data	WebTreatments	Data-Decision-Request-Cust...	Yes	NBA Specialist	14 minutes ago	
Property	OfferData	PegaMKT-Work-Offer	Yes	NBA Specialist	13 minutes ago	

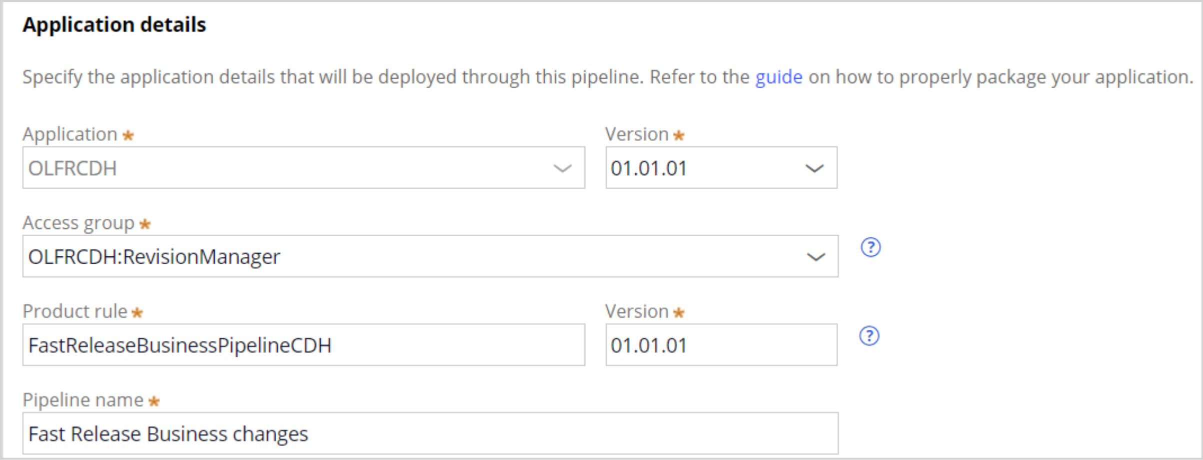
You have reached the end of this demo. What did it show you?

- How to configure a business change pipeline and establish communication between environments using the pipeline.
- How to create and deploy a revision through a business change pipeline.
- How a new version of an enterprise application and an application overlay ruleset is created for every revision.

Fast release business change pipeline

The process of creating a fast release business change pipeline is the same as the standard release business change pipeline. However, when configuring the fast release pipeline, use:

1. The name and version of the overlay fast release application, which is the application where emergency/unplanned business changes reside.
2. The access group for which you want to run the fast release pipeline tasks. Ensure that this access group is present in all environments.
3. The name and version of the product rule to use for packaging the fast release artifacts to deploy to the other candidate systems.



Application details

Specify the application details that will be deployed through this pipeline. Refer to the guide on how to properly package your application.

Application * OLFRCDH

Version * 01.01.01

Access group * OLFRCDH:RevisionManager

Product rule * FastReleaseBusinessPipelineCDH

Version * 01.01.01

Pipeline name * Fast Release Business changes

Both standard release and fast release changes are deployed using a single overlay ruleset.

Fast release business changes can be deployed without disrupting the standard release cycle.

If you need to address a business-related issue or make an emergency change in production, it is recommended to use a fast release. This will help you avoid affecting your regular, planned, standard release. Some important considerations when considering a fast release are:

1. When addressing production issues through a fast release, it's crucial that all instances are synchronized and have the latest deployed overlay rule set version.
2. It is not recommended to use both standard and fast releases during each release cycle. The fast release should only be used for emergency changes and not as a regular development tool.
3. As a best practice, avoid running the fast release and the standard release pipelines simultaneously.

4. When moving change requests from one release to another, consider all the dependencies on the Next Best Action Designer's decision data record. For example, if you added an eligibility in the standard release and then decide to move it to a fast release, ensure that you also move the associated artifacts.
5. Upon initiating the deployment of a release, the application undergoes a conflict check process that examines the impact of changes made between releases. This assessment is based exclusively on the changes that have been deployed, rather than any alterations that may have been made within each individual release.

Data migration pipeline

Description

Learn how a data migration pipeline enhances the process of automatically moving sampled production data into the business operations environment (BOE) and provides the means to simulate what-if scenarios on actual customer interactions and data.

Learning objectives

- Define a data migration pipeline
- Prepare environments for data migration
- Migrate sample production data to the BOE

Data migration pipeline

The data migration pipeline enhances the process of automatically moving sample production data into the Business Operations Environment (BOE). The pipeline provides you with the means to simulate what-if scenarios on sample production data without affecting the production environment.

Simulation tests provide you with significant insights into how your business changes affect the results that your strategies produce. However, if tests are based on test data, the results do not always reflect the effect that your changes have in real-life scenarios. To ensure that simulations are reliable enough to help you make important business decisions, you can deploy a sample of your production data to a dedicated BOE for testing.

Through sampling, you can focus on a representative subset of a large population; therefore, you do not have to analyze large groups of subjects. You can also use sampling to reduce the amount of time that is needed to run a simulation that delivers accurate projections by extrapolating across the entire population.

Pega Customer Decision Hub™ provides separate inbound and outbound samples of your production customer data that you can use as the audience input for a simulation.

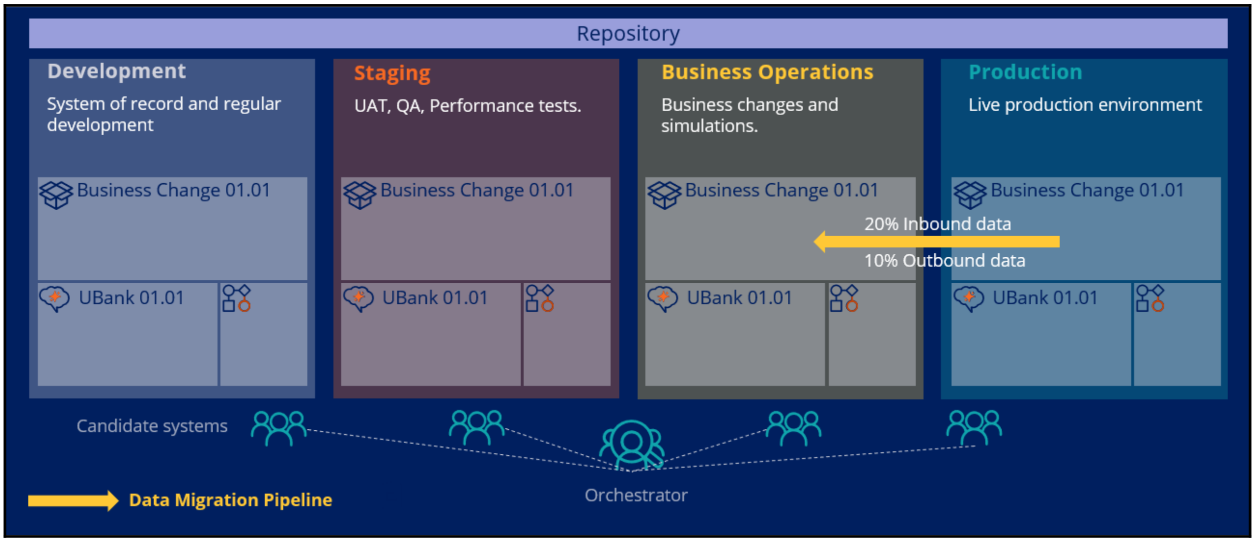
By using a data migration pipeline, you can export an inbound and outbound sample dataset, which includes **associated** interaction history and adaptive models, from the production environment, and then import that data into the BOE.

You can schedule refreshes of the sample datasets in the BOE at specific intervals.

As a result, the simulations are based on the migrated production data, that is, customer details, adaptive model data, and interaction history, which helps you assess how your changes are reflected in real-life scenarios.

The Pega Customer Decision Hub, inbound and outbound sampling algorithm randomly samples up to 20% of your associated decision requests. By default, the inbound sample size is 5% and the outbound sample size is 10%. You can consider limiting the sample size as required depending on the amount of available interactions.

For example, you can sample a selection of 20% inbound and 10% outbound real customer interactions, and then migrate the data to the BOE to test how the changes to existing strategies might affect the business outcomes.



To migrate the sample production data, you need to complete a one-time configuration in the development environment to generate the data migration artifacts and move them to production environment and BOE. You can set up that activity by migrating the artifacts with an enterprise pipeline or manually importing rules.

Data migration artifacts contain sampling data sets, data flows, and file repository data sets that are required to sample and migrate the production data and the dynamic system settings.

After the data migration artifacts are generated in the development environment and moved to the Production and BOE systems, and when the data migration pipeline is configured, you can migrate the simulation data.

You can also schedule a data migration pipeline to run during a specified period by creating and running a job scheduler.

Preparing the artifacts for data migration

To prepare your environment for data migration, complete a one-time configuration to generate all the required artifacts.

Transcript

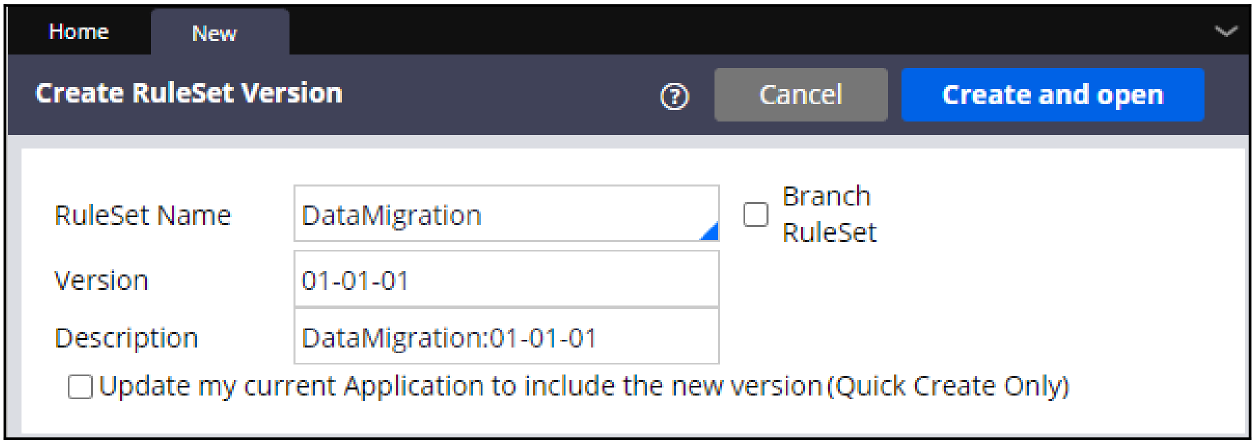
This demo shows you how to generate the artifacts and configure the settings that are required to assemble the data migration pipeline.

U+ Bank, a retail bank, would like to prepare their systems to be able to generate the artifacts and configure the settings that are required to migrate sampled data from the production environment to the Business Operations Environment (BOE).

Typically, a system architect completes the initial configuration in the development environment, and then you promote the generated artifacts and settings to higher environments.

In the development environment, as a best practice, start by creating a new ruleset to host all system-generated artifacts such as data sets and data flows. Storing these artifacts in a dedicated ruleset is not mandatory; however, this action is useful later when you package the artifacts in cases where an enterprise pipeline is not available. If an enterprise pipeline exists, you can use the pipeline for migrating the rules to the production and BOE environments.

Create a new rule set, and then add that ruleset to your application. This ruleset holds all the system-generated artifacts.



RuleSet Name: DataMigration

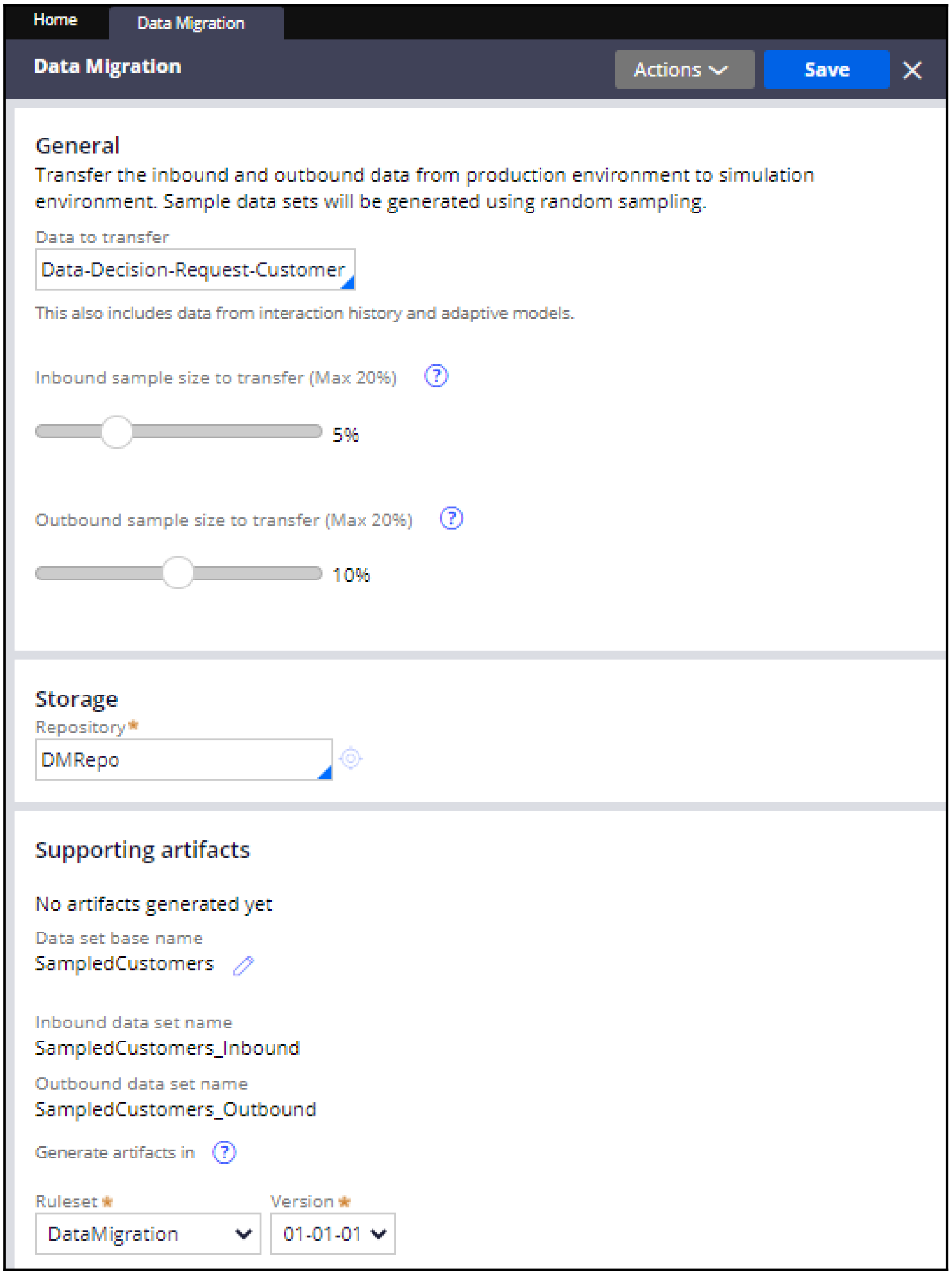
□ Branch RuleSet

Version: 01-01-01

Description: DataMigration:01-01-01

□ Update my current Application to include the new version (Quick Create Only)

As your next step, navigate to the **Data Migration** landing page. By default, the **Data-Decision-Request-Customer** class is selected in the **Data to transfer** field. Leave this default setting, and then select the inbound and outbound sample size. The sample size must be less than 20% for inbound and outbound sampling. Select a repository that both the BOE and the Production environment can access. If necessary, you can rename the data sets that ultimately hold the sample customer data. Now, choose the ruleset in which you want to store the new artifacts. In this case, select your new ruleset.



Data Migration

General

Transfer the inbound and outbound data from production environment to simulation environment. Sample data sets will be generated using random sampling.

Data to transfer: Data-Decision-Request-Customer

This also includes data from interaction history and adaptive models.

Inbound sample size to transfer (Max 20%): 5%

Outbound sample size to transfer (Max 20%): 10%

Storage

Repository*: DMRepo

Supporting artifacts

No artifacts generated yet

Data set base name: SampledCustomers

Inbound data set name: SampledCustomers_Inbound

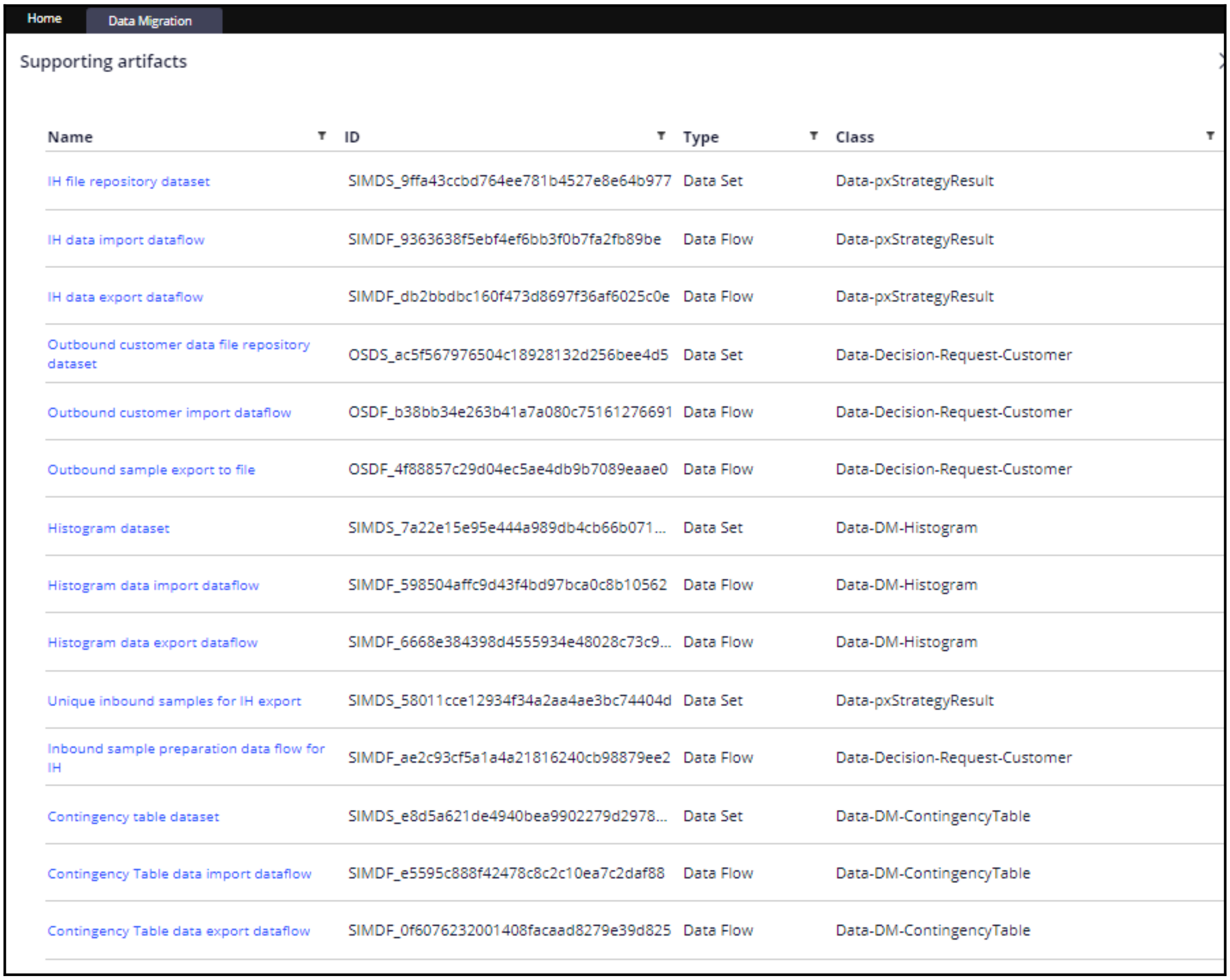
Outbound data set name: SampledCustomers_Outbound

Generate artifacts in

Ruleset*: DataMigration

Version*: 01-01-01

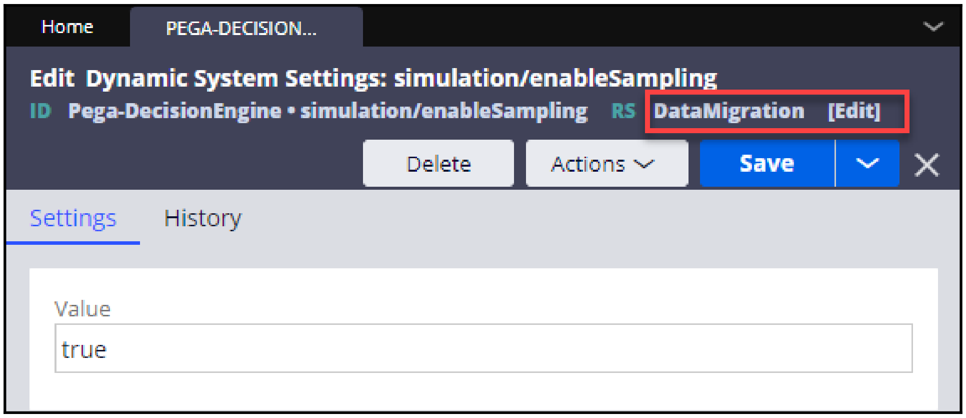
After saving the configuration, you can browse the generated artifacts by clicking the **Show artifacts** link. The data migration artifacts contain sampling data sets, data flows, and file repository data sets.



Supporting artifacts

Name	ID	Type	Class
IH file repository dataset	SIMDS_9ffa43ccbd764ee781b4527e8e64b977	Data Set	Data-pxStrategyResult
IH data import dataflow	SIMDF_9363638f5ebf4ef6bb3f0b7fa2fb89be	Data Flow	Data-pxStrategyResult
IH data export dataflow	SIMDF_db2bbdbc160f473d8697f36af6025c0e	Data Flow	Data-pxStrategyResult
Outbound customer data file repository dataset	OSDS_ac5f567976504c18928132d256bee4d5	Data Set	Data-Decision-Request-Customer
Outbound customer import dataflow	OSDF_b38bb34e263b41a7a080c75161276691	Data Flow	Data-Decision-Request-Customer
Outbound sample export to file	OSDF_4f88857c29d04ec5ae4db9b7089eaae0	Data Flow	Data-Decision-Request-Customer
Histogram dataset	SIMDS_7a22e15e95e444a989db4cb66b071...	Data Set	Data-DM-Histogram
Histogram data import dataflow	SIMDF_598504affc9d43f4bd97bca0c8b10562	Data Flow	Data-DM-Histogram
Histogram data export dataflow	SIMDF_6668e384398d4555934e48028c73c9...	Data Flow	Data-DM-Histogram
Unique inbound samples for IH export	SIMDS_58011cce12934f34a2aa4ae3bc74404d	Data Set	Data-pxStrategyResult
Inbound sample preparation data flow for IH	SIMDF_ae2c93cf5a1a4a21816240cb98879ee2	Data Flow	Data-Decision-Request-Customer
Contingency table dataset	SIMDS_e8d5a621de4940bea9902279d2978...	Data Set	Data-DM-ContingencyTable
Contingency Table data import dataflow	SIMDF_e5595c888f42478c8c2c10ea7c2daf88	Data Flow	Data-DM-ContingencyTable
Contingency Table data export dataflow	SIMDF_0f6076232001408facaad8279e39d825	Data Flow	Data-DM-ContingencyTable

Depending on the requirement, you can choose to only sample inbound interactions (inbound calls to Pega Customer Decision Hub™), create a sample from the entire customer base for running outbound simulations, or both. You control this functionality through dynamic system settings. To enable inbound sampling, save the **simulation/enableSampling** dynamic system setting to the same ruleset that you created earlier, and then set it to *true*.



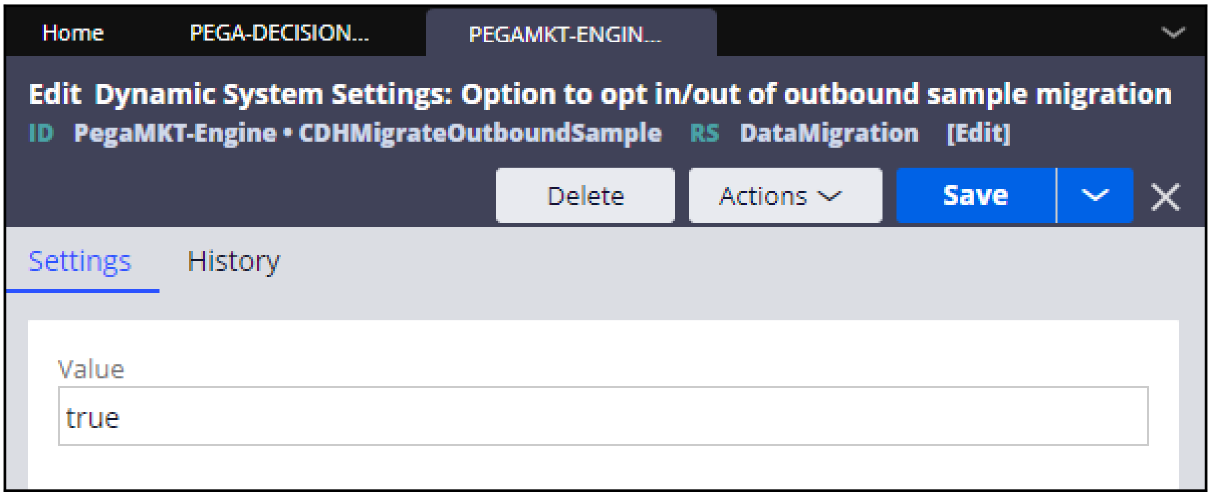
Edit Dynamic System Settings: simulation/enableSampling

ID Pega-DecisionEngine • simulation/enableSampling RS DataMigration [Edit]

Value

true

To enable outbound sampling, save the **CDHMigrateOutboundSample** dynamic setting to the same ruleset that you created earlier, and then set it to *true*.



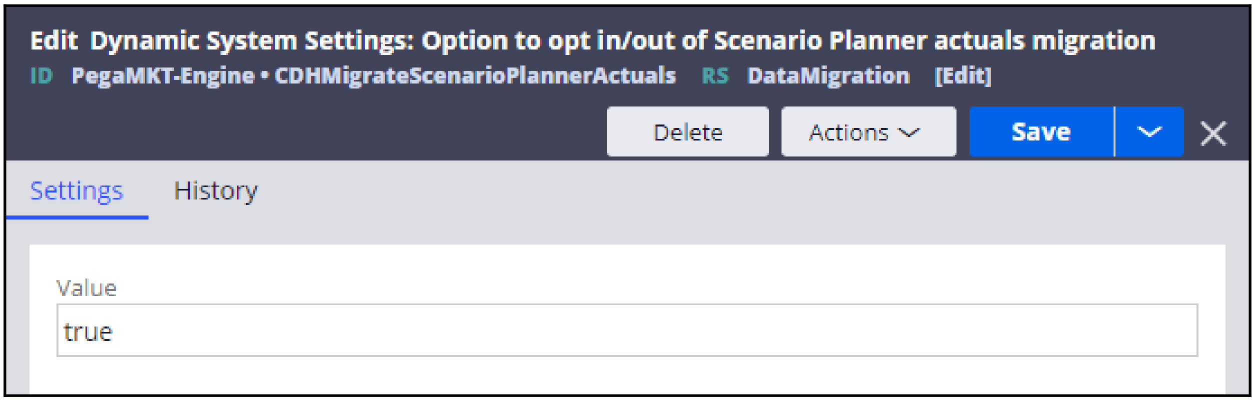
Edit Dynamic System Settings: Option to opt in/out of outbound sample migration

ID PegaMKT-Engine • CDHMigrateOutboundSample RS DataMigration [Edit]

Value

true

Besides migrating sampled customer data, the data migration pipeline can be used to migrate actual production data from production environment to the BOE. The actual data is used in scenario planner to determine how close the results are to meeting expected targets. To migrate the actuals data set, save the **CDHMigrateScenarioPlannerActuals** dynamic setting to the same ruleset that you created earlier, and then set it to *true*.



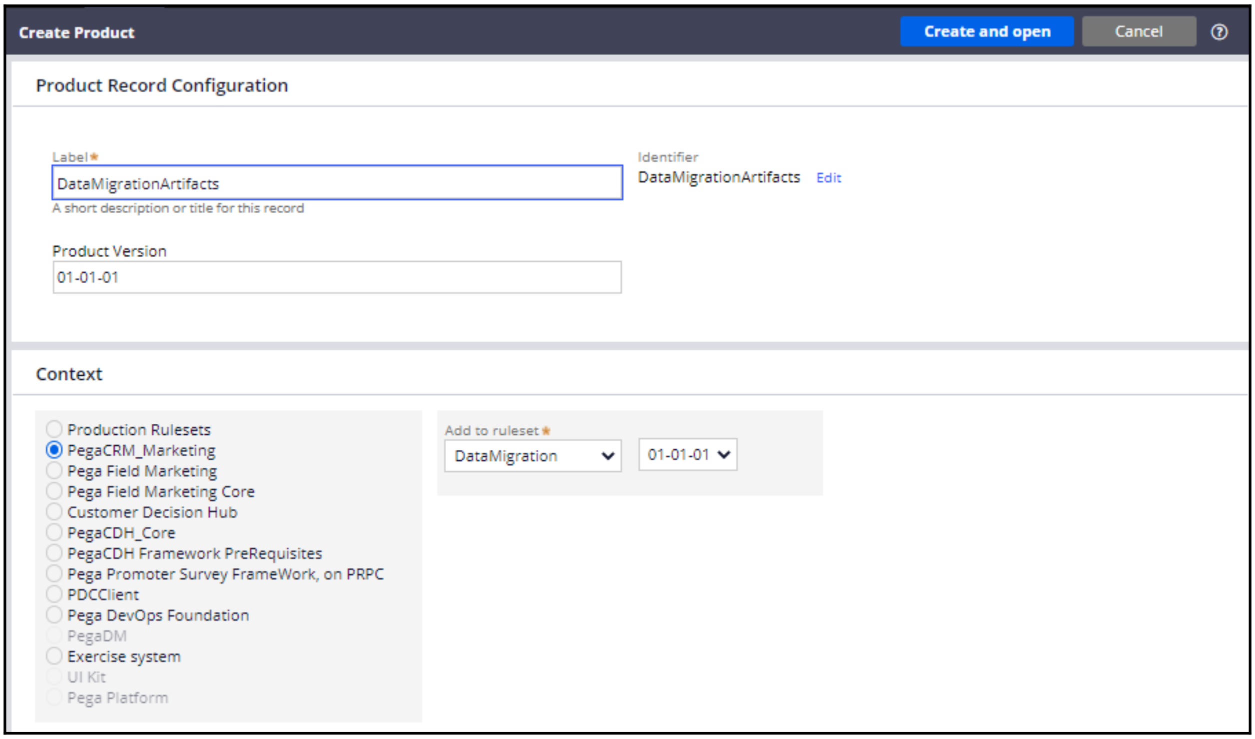
Edit Dynamic System Settings: Option to opt in/out of Scenario Planner actuals migration

ID PegaMKT-Engine • CDHMigrateScenarioPlannerActuals RS DataMigration [Edit]

Value

true

Now, create a product rule to package the generated artifacts. Give the rule a name and a product version.



Product Record Configuration

Label*: DataMigrationArtifacts
A short description or title for this record

Identifier: DataMigrationArtifacts Edit

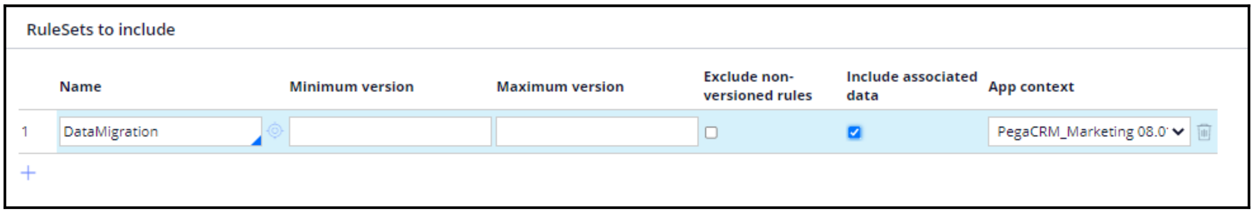
Product Version: 01-01-01

Context

○ Production Rulesets
● PegaCRM_Marketing
○ Pega Field Marketing
○ Pega Field Marketing Core
○ Customer Decision Hub
○ PegaCDH_Core
○ PegaCDH Framework PreRequisites
○ Pega Promoter Survey FrameWork, on PRPC
○ PDCClient
○ Pega DevOps Foundation
○ PegaDM
○ Exercise system
○ UI Kit
○ Pega Platform

Add to ruleset*: DataMigration 01-01-01

In the **RuleSets to Include** section, choose the ruleset in which the data migration artifacts are stored, and then check the **Include associated data** check box. This setting ensures that the **Data Migration** landing page where the settings were configured is also migrated.

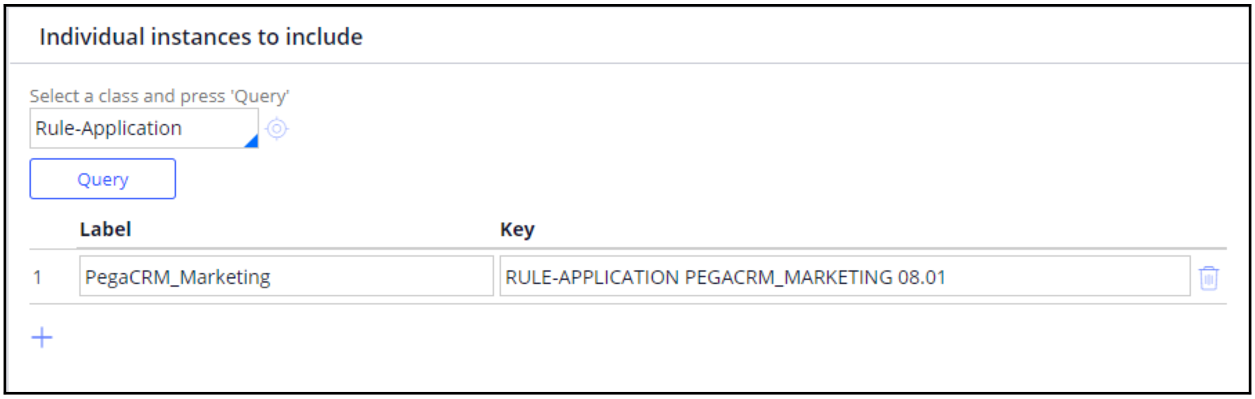


RuleSets to include

	Name	Minimum version	Maximum version	Exclude non-versioned rules	Include associated data	App context
1	DataMigration			□	☑	PegaCRM_Marketing 08.0

As a best practice, to keep the application stack in sync, add the application instance to the Product rule by querying the **Rule-Application** class, and then selecting the application.

In this case, select **PEGACRM_MARKETING 08.01**.



Individual instances to include

Select a class and press 'Query'

Rule-Application

Query

	Label	Key
1	PegaCRM_Marketing	RULE-APPLICATION PEGACRM_MARKETING 08.01

Save the product rule and preview the rules that are included. Then, lock the ruleset and generate the product rule.

As the next step, ensure that the setup of **MigrateSimulationData**, **DataSyncPipeline**, **api**, and **cicd** service packages is correct, and the **Require TLS/SSL for rest service in this package** check box is unchecked in the BOE and production environments. Complete these actions manually because the security requirements in environments might vary. The pipeline diagnostics tool ensures that you select the right option in each environment.

The final step in preparing the environments for data migration is to import the product rule into all environments. Use the enterprise pipeline to migrate artifacts when possible. This action ensures that each environment is in sync and that both the BOE and Production environments have the required artifacts to migrate data.

You have reached the end of this demo. What did it show you?

- How to configure the data migration settings and generate artifacts.
- How to enable inbound and outbound sampling.
- How to enable actuals data migration for scenario planner.
- How to create a product rule with the required data migration artifacts and import that rule into the BOE and Production systems.

Migrating sampled data with a data migration pipeline

The data migration pipeline facilitates moving sampled production data into the business operations environment (BOE) automatically. It provides you with the means to simulate what-if scenarios on the sample production data without affecting the production environment.

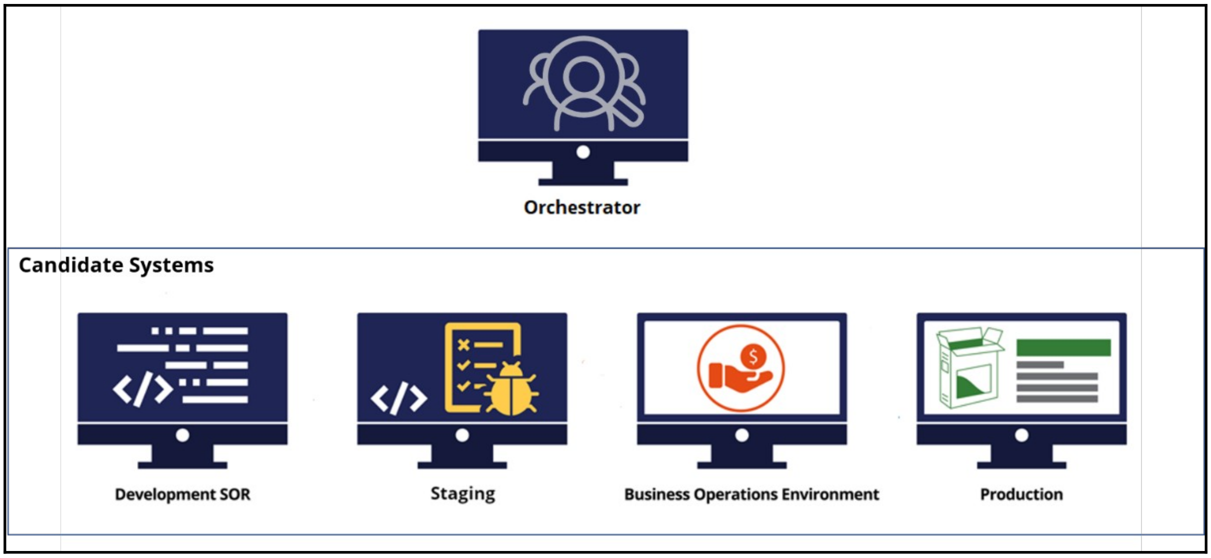
Transcript

This video shows you how to configure a data migration pipeline to migrate sampled production data from the production environment to the business operations environment.

U+ Bank, a retail bank, plans to configure Pega Deployment Manager to set up a data migration pipeline and migrate 20% of its customer base from the production environment to the BOE for the business users to simulate what-if scenarios.

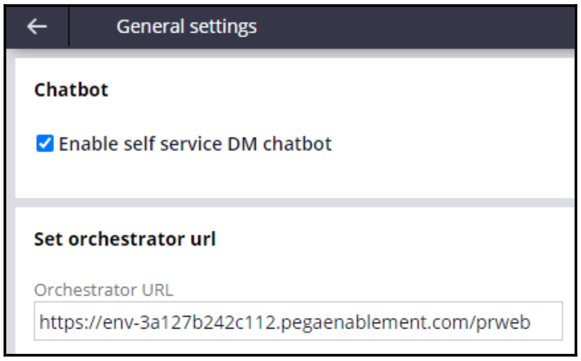
A typical decision management project involves five environments:

- An orchestrator to manage and automate the deployment process of artifacts to the candidate environments
- A development (system of record) environment to develop enterprise capabilities
- A staging environment to test the changes and analyze their technical impact
- A business operations environment (BOE) to make business changes
- A production system to serve customers with the latest artifacts



To implement this requirement, first ensure that the BOE and the production environment communicates with the orchestrator. Next, in the orchestrator, configure a data migration pipeline to define the migration process of the sampled production data. Once the Release Manager configures the pipeline, customer interactions in the production environment are sampled and migrated to BOE through a data migration pipeline.

In the BOE and the production environment, update the **PegaDevOpsShared** dynamic system setting (DSS) to point to the orchestrator system URL. Updating the DSS links the BOE and production systems to the orchestrator system.



Chatbot

☑ Enable self service DM chatbot

Set orchestrator url

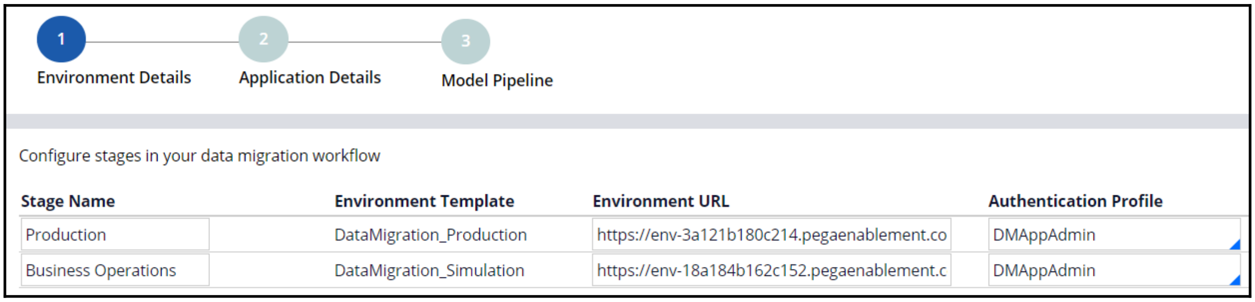
Orchestrator URL: https://env-3a127b242c112.pegaenablement.com/prweb

Then, configure a data migration pipeline in the Deployment Manager portal in the orchestrator. The data migration pipeline allows you to move data from one environment to another so that you can run simulations on a preselected sample set of customers. You can make millions of decisions simultaneously and simulate the outcome of your decision management framework.

Deployment Manager comes with out-of-the-box pipeline templates that help you structure pipelines based on their purpose. For this requirement, use the **Data Migration Pipeline** template. Begin by providing the URL of the production environment.

Then, select the **Authentication profile** that communicates with the candidate environments. An authentication profile is automatically created during the installation of Deployment Manager. The orchestrator uses this authentication profile to communicate with candidate systems to run tasks in the pipeline. Use this authentication profile in all the candidate systems with the same operator ID.

Now, provide the URL of the BOE, and then select the authentication profile.

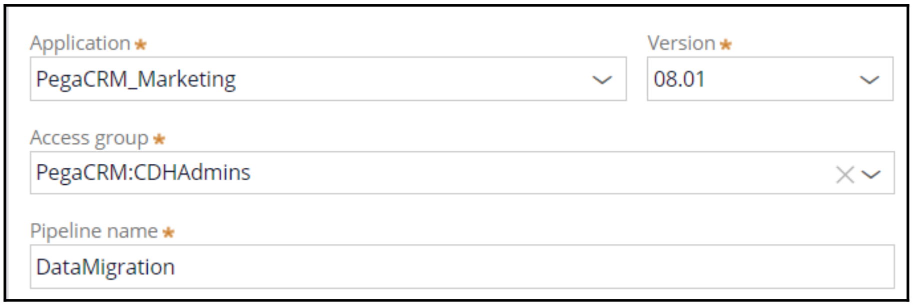


Configure stages in your data migration workflow

Stage Name	Environment Template	Environment URL	Authentication Profile
Production	DataMigration_Production	https://env-3a121b180c214.pegaenablement.co	DMAppAdmin
Business Operations	DataMigration_Simulation	https://env-18a184b162c152.pegaenablement.c	DMAppAdmin

Select the name and version of the enterprise application. Then, select the access group for which you want to run the pipeline tasks. Ensure that this access group is present in all environments.

Now, enter a unique name for the pipeline, *Data Migration* to create a model pipeline.



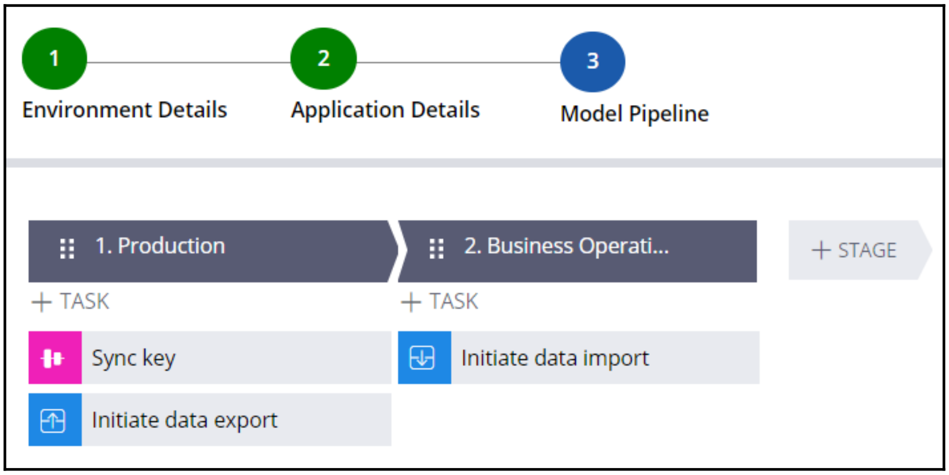
Application * PegaCRM_Marketing

Version * 08.01

Access group * PegaCRM:CDHAdmins

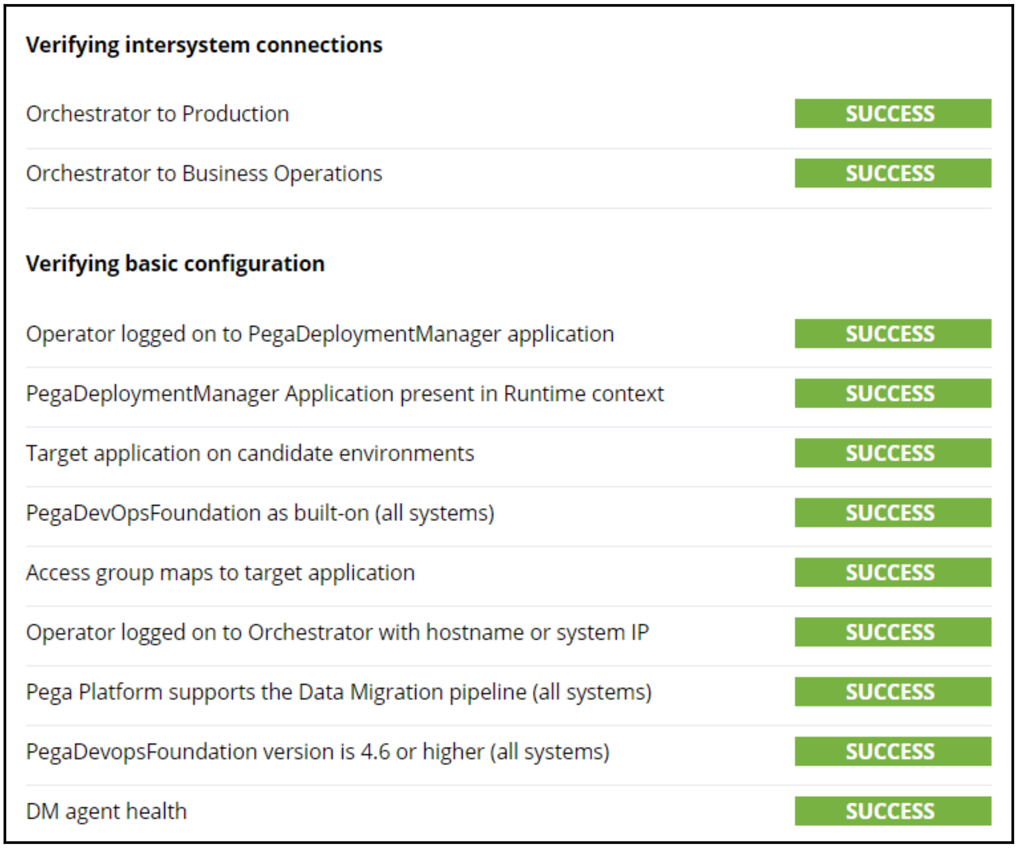
Pipeline name * DataMigration

The pipeline model is a case type that defines various predefined tasks that are necessary to accomplish a successful data migration.



Run diagnostics to check if the orchestrator can communicate with the BOE and production systems and application details on the pipeline are defined correctly. On successful check, the pipeline is ready to migrate the sampled data from the production environment to the

BOE.



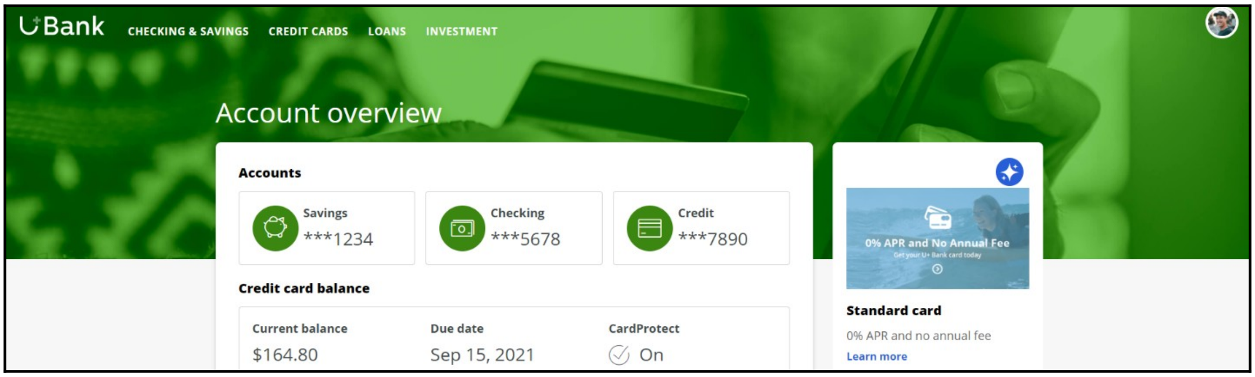
Verifying intersystem connections

Orchestrator to Production	SUCCESS
Orchestrator to Business Operations	SUCCESS

Verifying basic configuration

Operator logged on to PegaDeploymentManager application	SUCCESS
PegaDeploymentManager Application present in Runtime context	SUCCESS
Target application on candidate environments	SUCCESS
PegaDevOpsFoundation as built-on (all systems)	SUCCESS
Access group maps to target application	SUCCESS
Operator logged on to Orchestrator with hostname or system IP	SUCCESS
Pega Platform supports the Data Migration pipeline (all systems)	SUCCESS
PegaDevopsFoundation version is 4.6 or higher (all systems)	SUCCESS
DM agent health	SUCCESS

Access the U+ Bank website from the production environment. Log in as different customers to record varied interactions. When you prepare the system for data migration, you configure the data migrations settings that sample the customer interactions.



Account overview

Accounts

Savings ***1234

Checking ***5678

Credit ***7890

Credit card balance

Current balance	Due date	CardProtect
$164.80	Sep 15, 2021	✓ On

Standard card

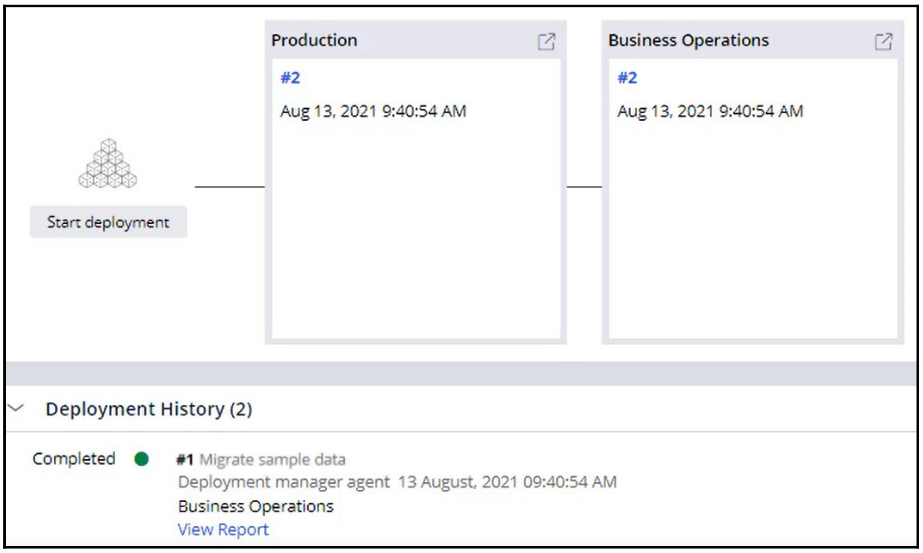
0% APR and no annual fee

You can initiate the data migration now that you have successfully created the pipeline and recorded some interactions.

While the migration is in the first stage, access the production environment to view the case that the system creates to export production data. Monitor the case to see the various tasks that are performed. The case shows the number of processed inbound samples, adaptive model data, and reports.

Notice that all adaptive models from the production system are migrated to the BOE. This migration ensures that the simulations that run in the BOE are accurate in gauging expected outcomes. On completion of the case, navigate to the orchestrator to see that the deployment advances to the second stage.

Navigate to the BOE to view the data import case. Monitor the case to see the various tasks that the performed. On completion of the case, navigate to the orchestrator to see that the deployment is complete.



Production

#2

Aug 13, 2021 9:40:54 AM

Business Operations

#2

Aug 13, 2021 9:40:54 AM

Deployment History (2)

Completed ● **#1** Migrate sample data
Deployment manager agent 13 August, 2021 09:40:54 AM
Business Operations
View Report

In the BOE, search for and open one of the sampled data sets to view the structure and content of the sampled data. You can also view the **Recent Interactions** in the **Interaction history**.

When the system creates new interaction records, the records accumulate in the production environment. After reinitiating the data migration pipeline, the system samples the latest production data, which overwrites the sampled data set in the BOE. In this way, the BOE contains a sample of the latest production data.

You can also schedule a data migration pipeline to run during a specified period by creating and running a job scheduler.

You have reached the end of this video. What did it show you?

- How to configure a data migration pipeline and establish communication between environments.
- How to migrate sampled production data with a data migration pipeline.