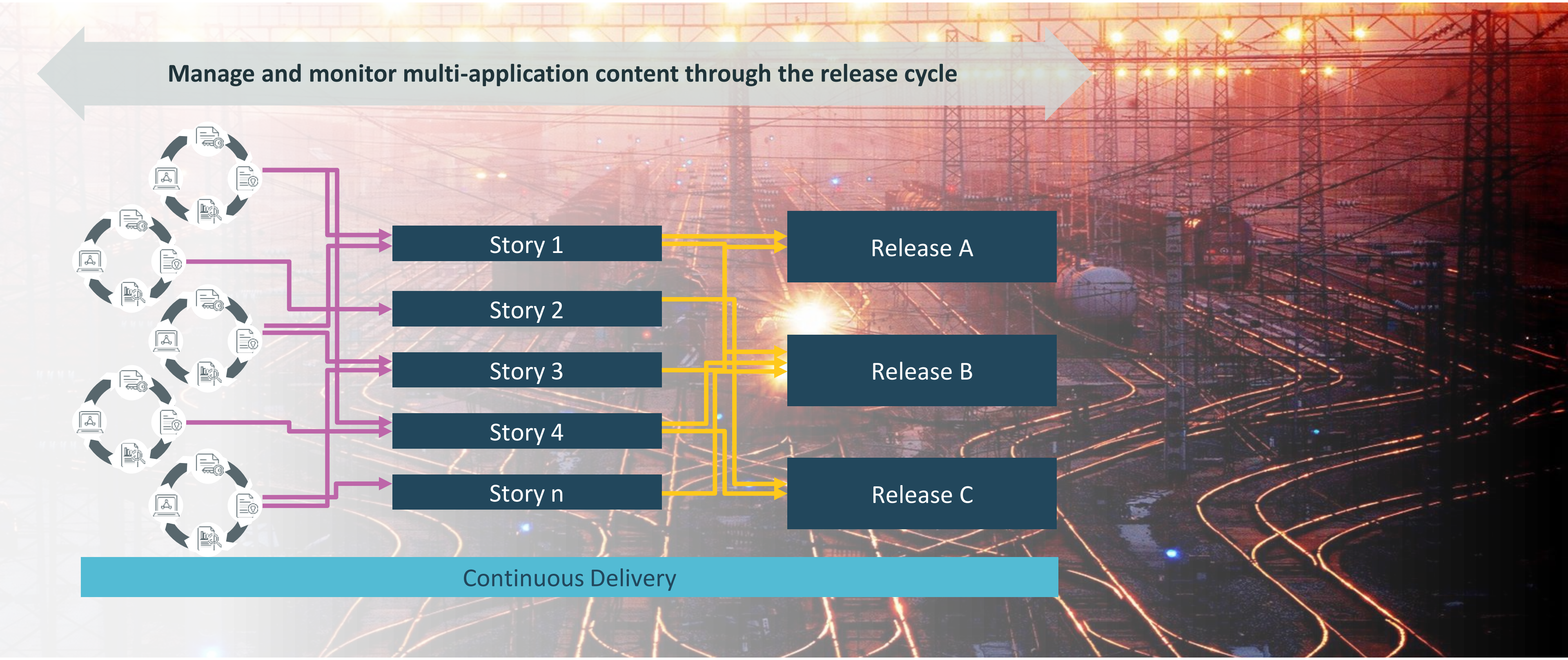
CA Release Automation Continuous Delivery Edition Integration to Agile Central

Release Auotmation Continuous Delivery Edition



Manage and monitor multi-application content through the release cycle

Story 1

Story 2

Story 3

Story 4

Story n

Release A

Release B

Release C

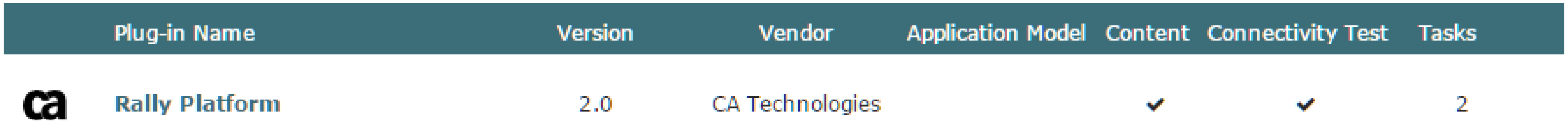
Continuous Delivery

CA Agile Central Integration

Plugin Functionality

The Plugin provides the integration between CDE and Agile Central

1. **Endpoint** – Connectivity between CDE and Agile Central
2. **Content** – Provides for the Importing of Content from Agile Central
3. **Tasks** – Allows CDE to perform Tasks within Agile Central



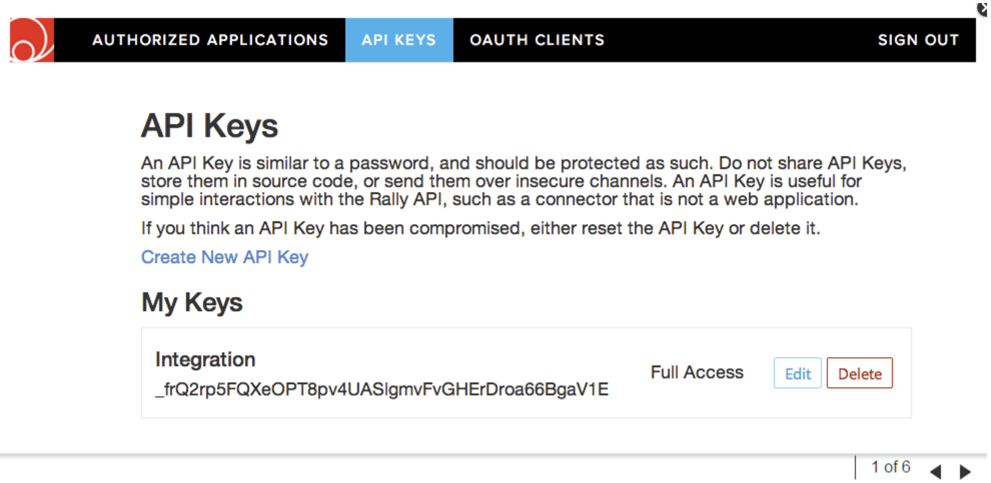
	Plug-in Name	Version	Vendor	Application Model	Content	Connectivity Test	Tasks
ca	Rally Platform	2.0	CA Technologies		✓	✓	2

CA Agile Central Integration

Endpoint

The configure the Endpoint you need three pieces of information:

- **URL** – To the CA Agile Central Server
- **Workspace Name** – CA Agile Central area you wish to access
- **API Key** – An encrypted key that grants access to CA Agile Central
  - CA Agile Central Application Manager (different URL) after login



API Keys

An API Key is similar to a password, and should be protected as such. Do not share API Keys, store them in source code, or send them over insecure channels. An API Key is useful for simple interactions with the Rally API, such as a connector that is not a web application.

If you think an API Key has been compromised, either reset the API Key or delete it.

Create New API Key

My Keys

Integration
_frQ2rp5FQXeOPT8pv4UASlgmvFvGHErDroa66BgaV1E

Full Access

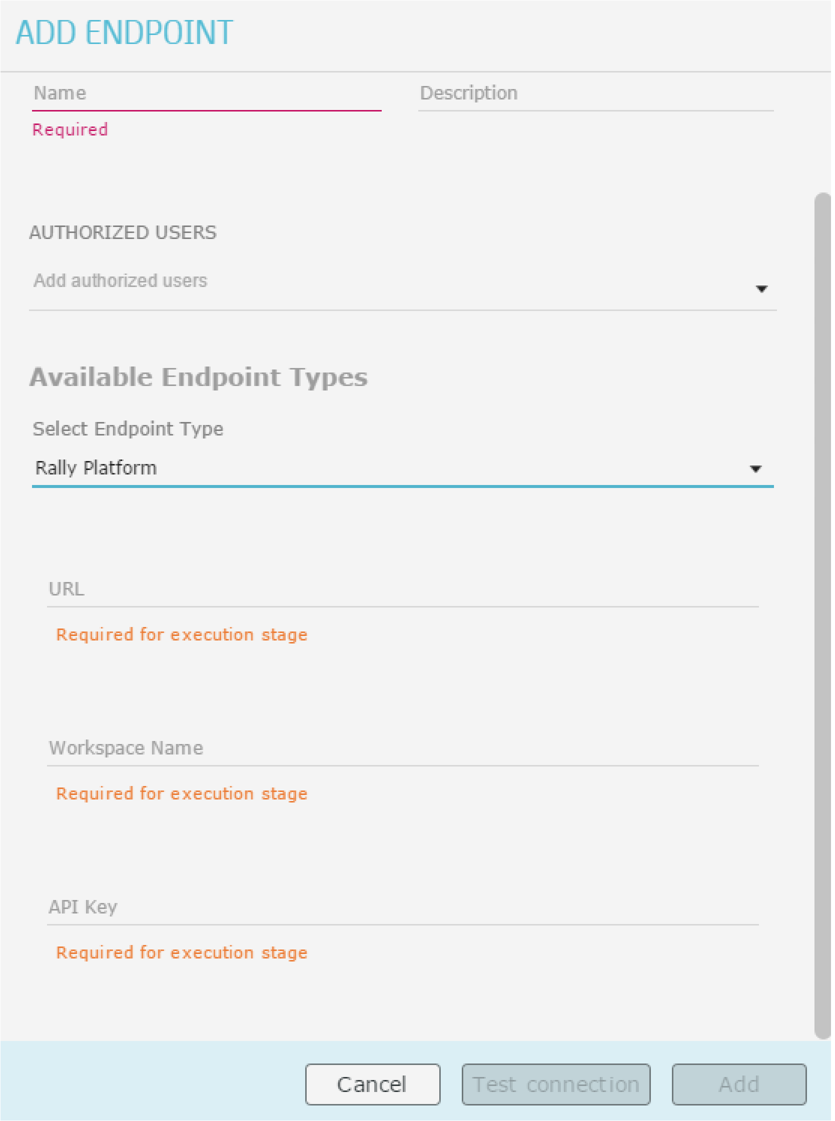
CA Agile Central Integration

Endpoint Configuration

- **Name** – Unique for connection
- **Description** – Optional
- **Endpoints Type** – Rally Platform
- **URL**
- **Workspace Name**
- **API Key**

Test Connection

Agile Central Server - Production ENDPOINT



ADD ENDPOINT

Name

Required

Description

AUTHORIZED USERS

Add authorized users

Available Endpoint Types

Select Endpoint Type: Rally Platform

URL

Required for execution stage

Workspace Name

Required for execution stage

API Key

Required for execution stage

CA Agile Central Integration

Import Content

- **Features** – an improvement that is to be developed (group of User Stories)
- **User Stories** – a grouping of Tasks and Defects that are part of a Feature
- **Tasks** – a unit of work within a User Story
- **Defects** – a bug or flaw in the program, that may be added to a User Story

Required Fields

Name of Content Source	
Type = (Rally Platform) Add Content	
Endpoint = Agile Central Server	
Project Name	@
Type (from list above)	list

Filter Fields

Release Name	@
Iteration Name	@
Item Status	@

CA Agile Central Integration

Perform Tasks

- **Rally Update** – changes the Item Status of a specific Task or Defect
- **Check Test Case Results** – for test performed on a User Story or Defect

Rally Update

Project Name	@
Type (Defect or Task)	list
Rally Item ID (DE or TA)	
New Item Status	@

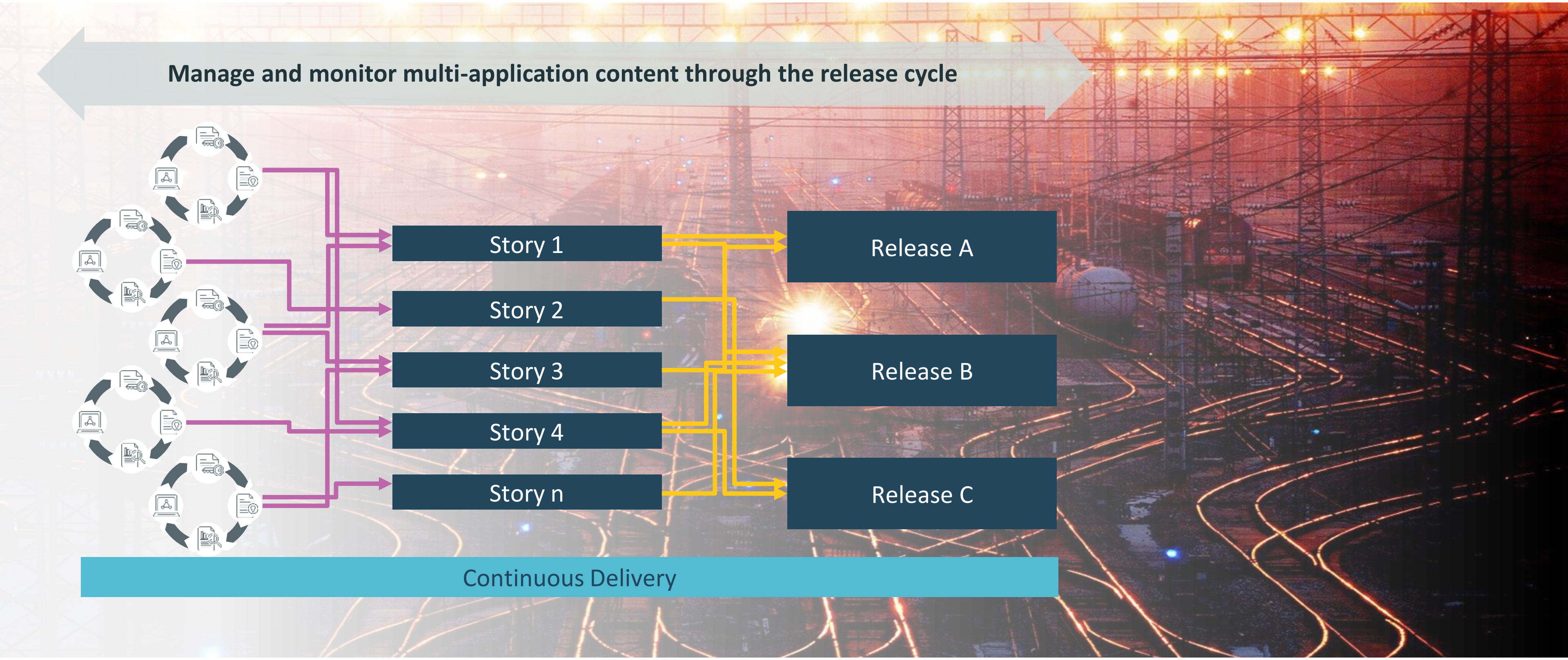
Check Test Case Results

Project Name	@
User Story and/or Defect ID	
Test Case Type	list
Build	

Release Automation Continuous Delivery Edition Integration to Agile Central Demonstration

CA Release Automation Continuous Delivery Edition Integration to JIRA

Release Automation Continuous Delivery Edition



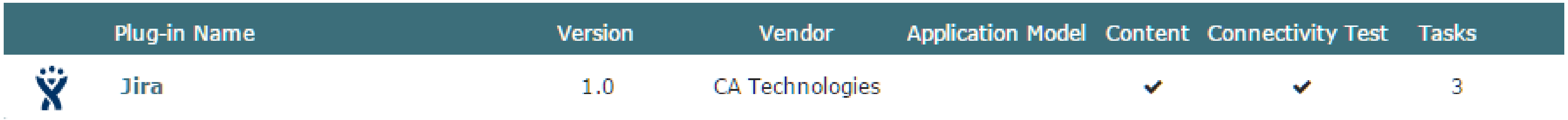
technologies

JIRA Integration

Plugin Functionality

The Plugin provides the integration between CDE and JIRA

1. **Endpoint** – Connectivity between CDE and JIRA
2. **Content** – Provides for the Importing of Content from JIRA
3. **Tasks** – Allows CDE to perform Tasks within JIRA



	Plug-in Name	Version	Vendor	Application Model	Content	Connectivity Test	Tasks
	Jira	1.0	CA Technologies		✓	✓	3

JIRA Integration

Endpoint Configuration

The configure the Endpoint you need three pieces of information:

- **URL** – To the JIRA Server
- **User Name** – With rights to access the JIRA Server
- **Password** – To the User

Additional Proxy Related Fields

- Proxy Host
- Proxy Port
- Proxy Username
- Proxy Password
- HTTPS Enable
- Time Out

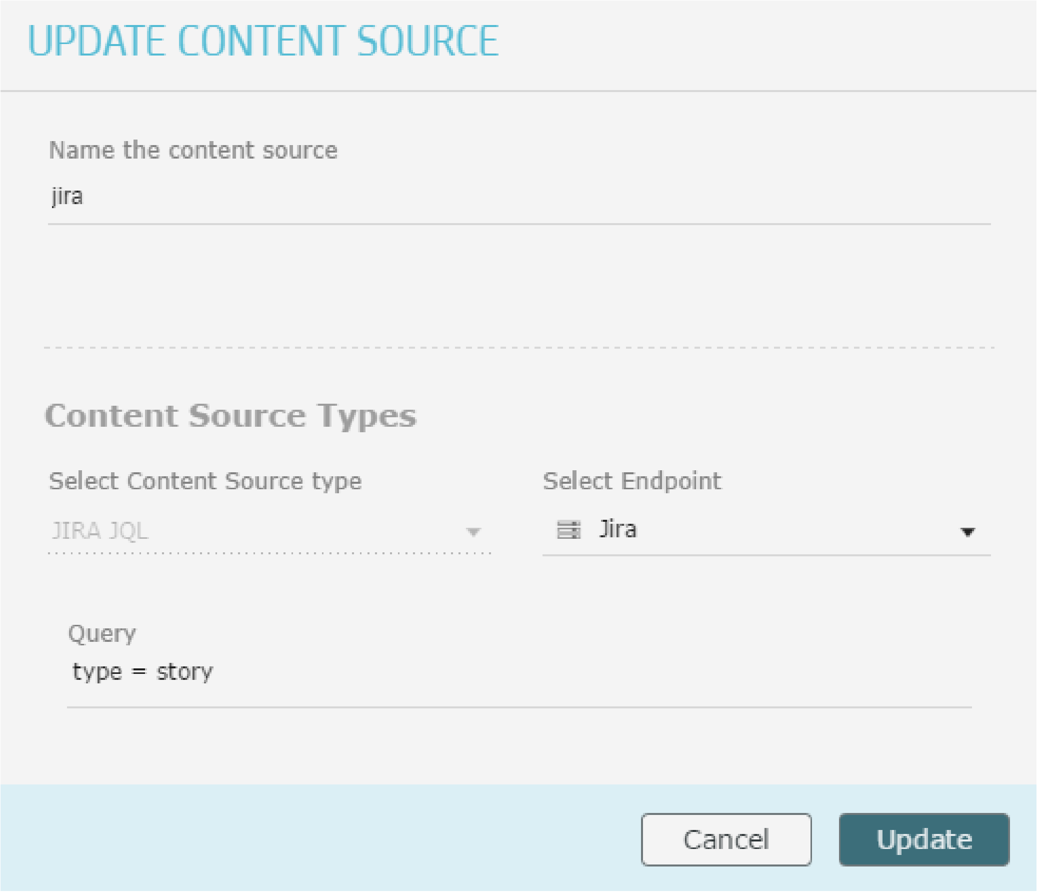
Test Connection

🔌

JIRA Integration

Import Content

- Name of Content Source
- Type = JIRA JQL
- Endpoint = JIRA Server
- Query
  - type = story



UPDATE CONTENT SOURCE

Name the content source
jira

Content Source Types

Select Content Source type
JIRA JQL

Select Endpoint
Jira

Query
type = story

JIRA Integration

Import Content – JQL Query

- Summary
  - Example: summary = issue ID_123456
- Type
  - Example: type = bug and status = resolved
- External ID
  - Example: External issue ID ~YourID_123456
- Display Type
  - Example: type = status = open
- Status
  - Example: status = open and priority = urgent and assignee = jsmith

JIRA Integration

Perform Tasks

- **Add JIRA Issue Comment** – adds comments to existing JIRA Issue
- **Update JIRA Issue Status** – changes the status of an existing JIRA Issue
- **Create JIRA Issue** – creates a new JIRA Issue

Add JIRA Issue Comment

- Issue Key
- Comment
- Role @

Update JIRA Issue Status

- Issue Key
- Transition (New State) @
- Resolution @
- Comment

JIRA Integration

Perform Tasks

Create JIRA Issue

- JIRA Project Key (CI) @
- Issue Type (task or story or bug) @
- Summary (Title)
- Epic Name
- Priority
- Due Date
- Affects Versions
- Fix Versions
- Assignee
- Reporter
- Environment
- Descriptions
- Labels

Release Automation Continuous Delivery Edition Integration to JIRA Demonstration