Modernize Legacy Applications to Kubernetes with Konveyor

devconf.cz, Brno, 13. 6. 2024

Marek Aufart

maufart@redhat.com

KONVEYOR

Speaker and project

- Marek Aufart, Engineer at Red Hat, Brno office, github.com/aufi
- Konveyor (CNCF sandbox project)
  - Community that helps modernize applications by providing open source tools to rehost, replatform, and refactor applications to Kubernetes and cloud-native technologies.

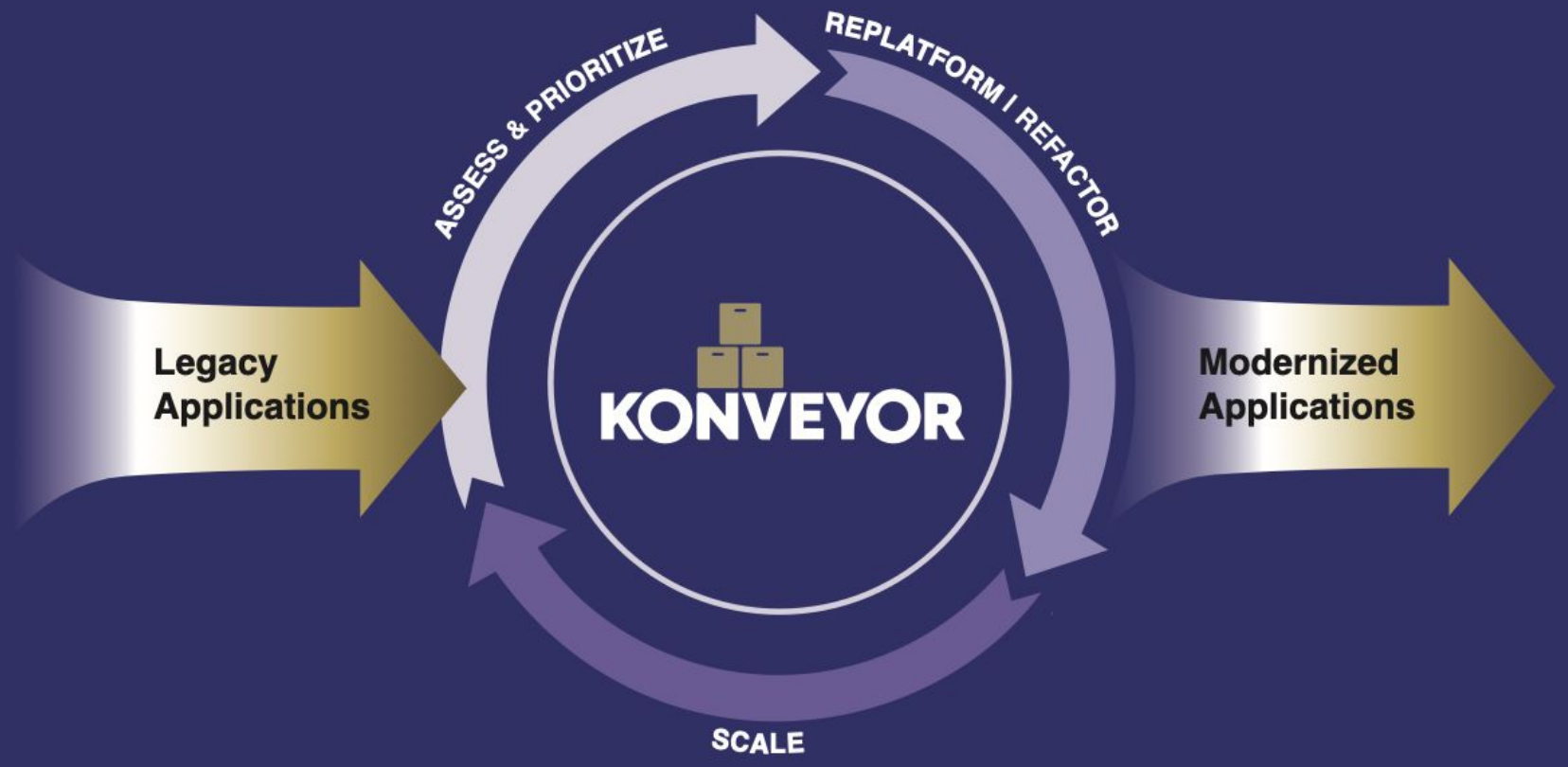
KONVEYOR

CLOUD NATIVE COMPUTING FOUNDATION

Migration and Modernization

Why it matters

- https://www.konveyor.io/modernization-report/
- Increase automation, security, reliability&scalability, CI/CD
- Main problem is complexity of existing applications
- 6R
  - retire, retain, rehost, replatform, refactor, repurchase
- Rehost
  - Forklift for VMs (kubev2v), Crane for containers
- Replatform, Refactor
  - That's Konveyor focus



The ultimate Open Source toolkit to help organizations **safely** migrate and modernize their application portfolio to leverage Kubernetes and Cloud-Native technologies, providing differential value on each stage of the adoption process

Intro

Methodology

Konveyor application

Development

Conclusion

Methodology

Konveyor Unified Experience

- Modernize applications at scale
- Unified experience
- https://github.com/konveyor/methodology

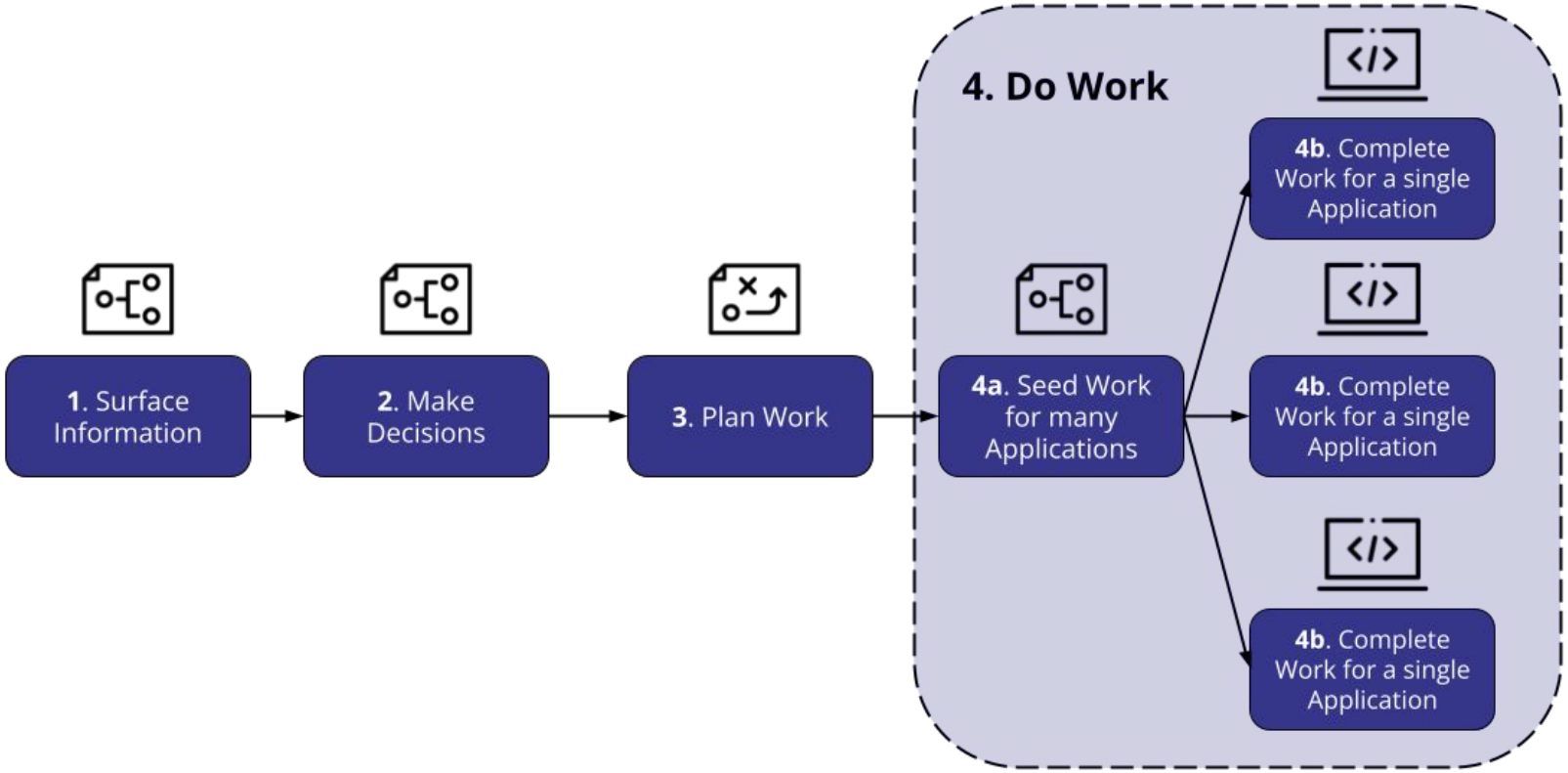
Methodology

Stages and personas in the Konveyor Workflow

Architect: Stakeholder who understands the business value of the modernization effort and makes the ultimate decision of what should be done.

Project Manager: Stakeholder in charge of planning the work required for the modernization initiative and measuring its progress as it gets executed.

Migrator: Developer responsible for making the specific source code changes to accomplish the modernization need



Intro

Methodology

Konveyor application

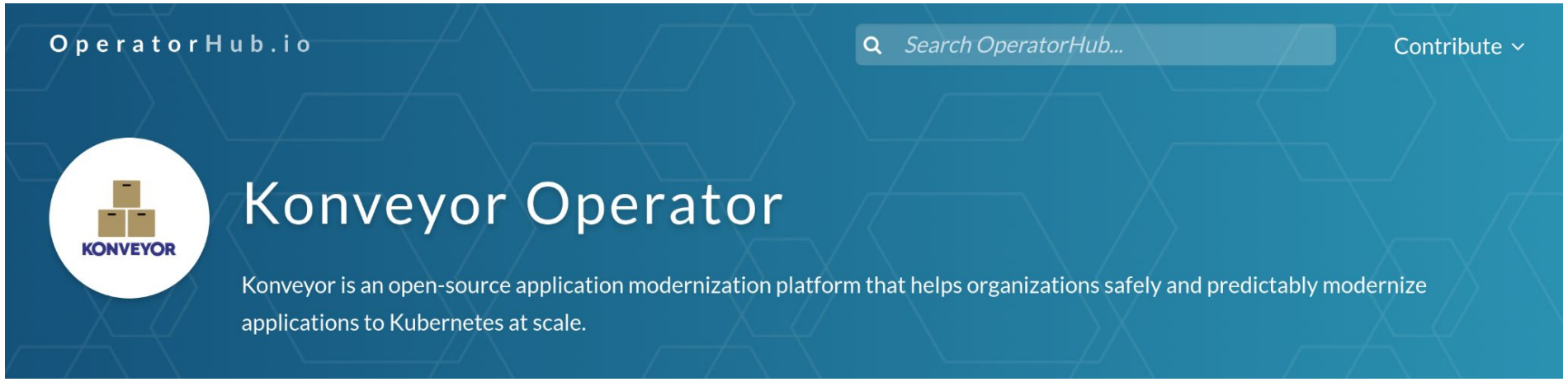
Development

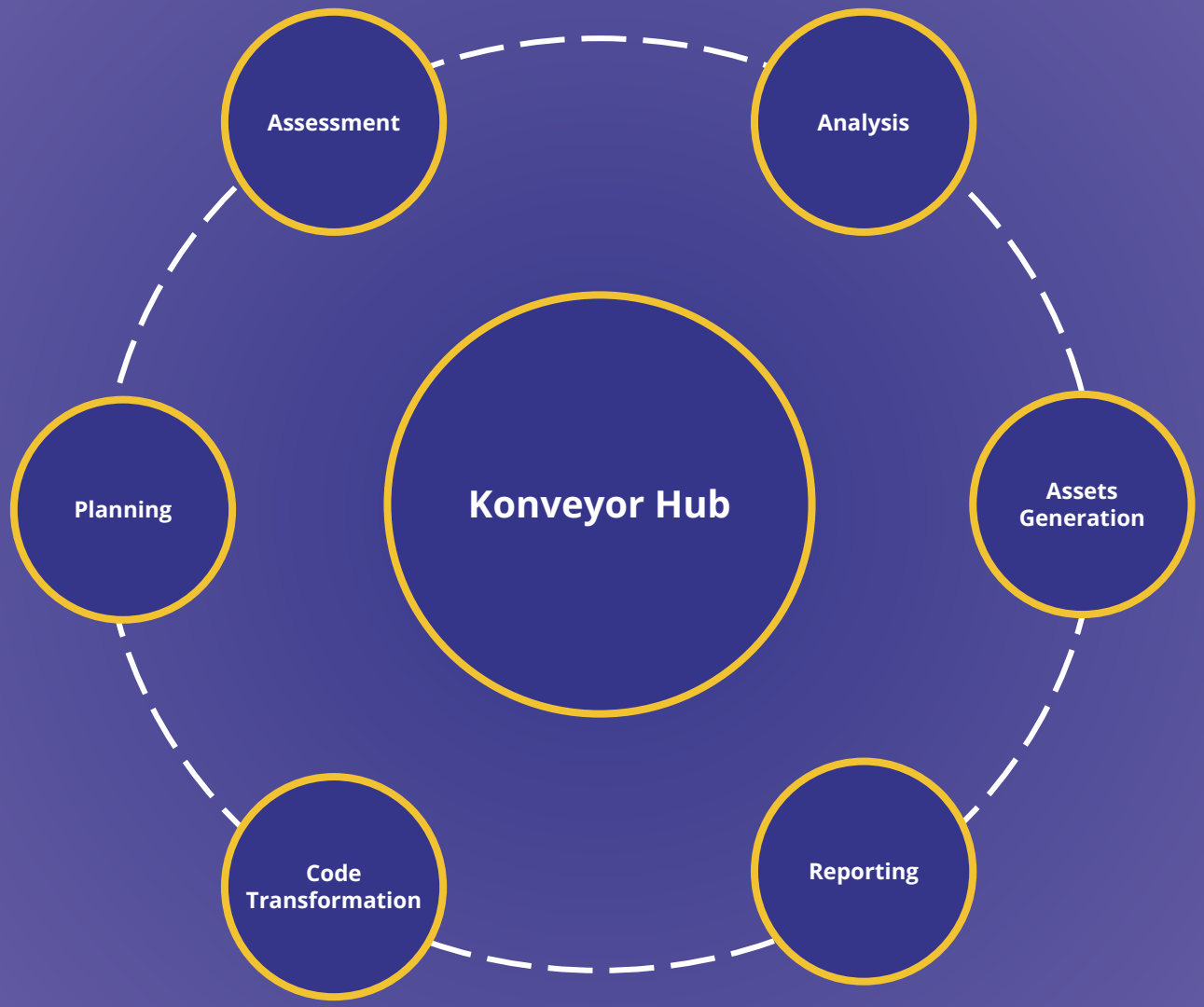
Conclusion

Konveyor application

Installation on k8s

- Originated in Java-world tools Windup and Pathfinder
- Nowadays, multi-language modernization tool
- Install as Kubernetes operator from Operator Hub

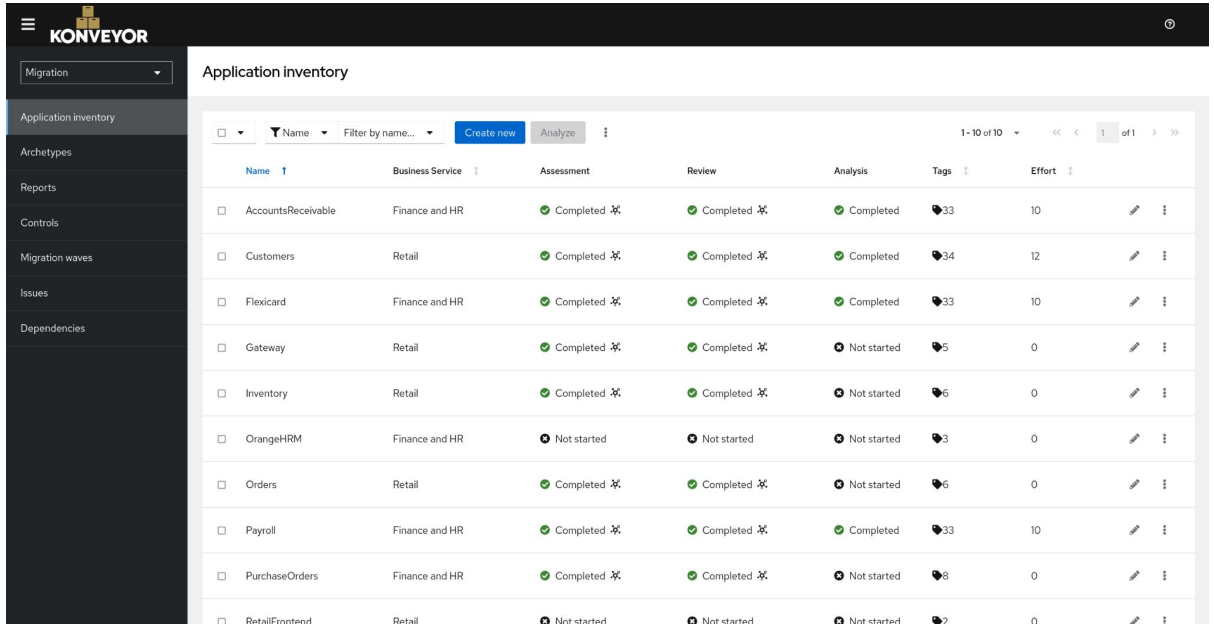



Assessment
Analysis
Planning
Konveyor Hub
Assets Generation
Code Transformation
Reporting

KONVEYOR

Application Inventory

Konveyor UI

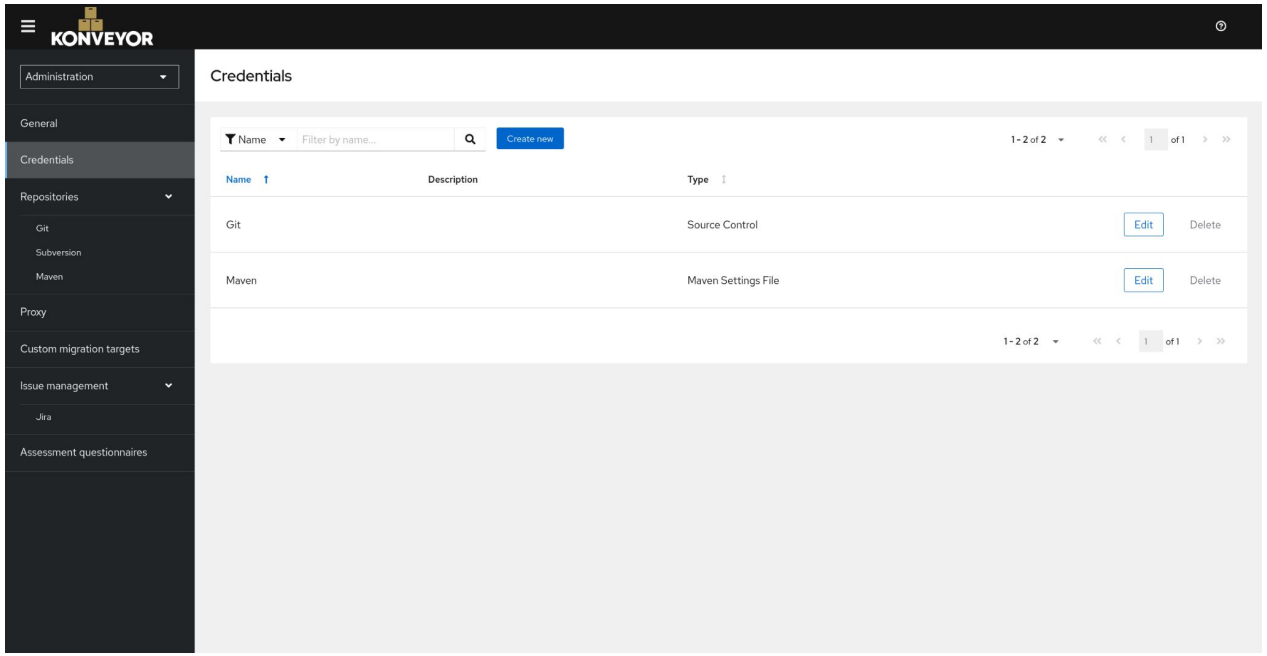


- Used to maintain a portfolio of applications
- It is the hub, and natural integration point for all Konveyor projects in the future
- Applications can be linked to the business services that they support
- Application interdependencies can be defined and managed
- Through the use of tags extensible metadata can be added to describe and categorize the applications in multiple dimensions

Application Inventory

Konveyor UI

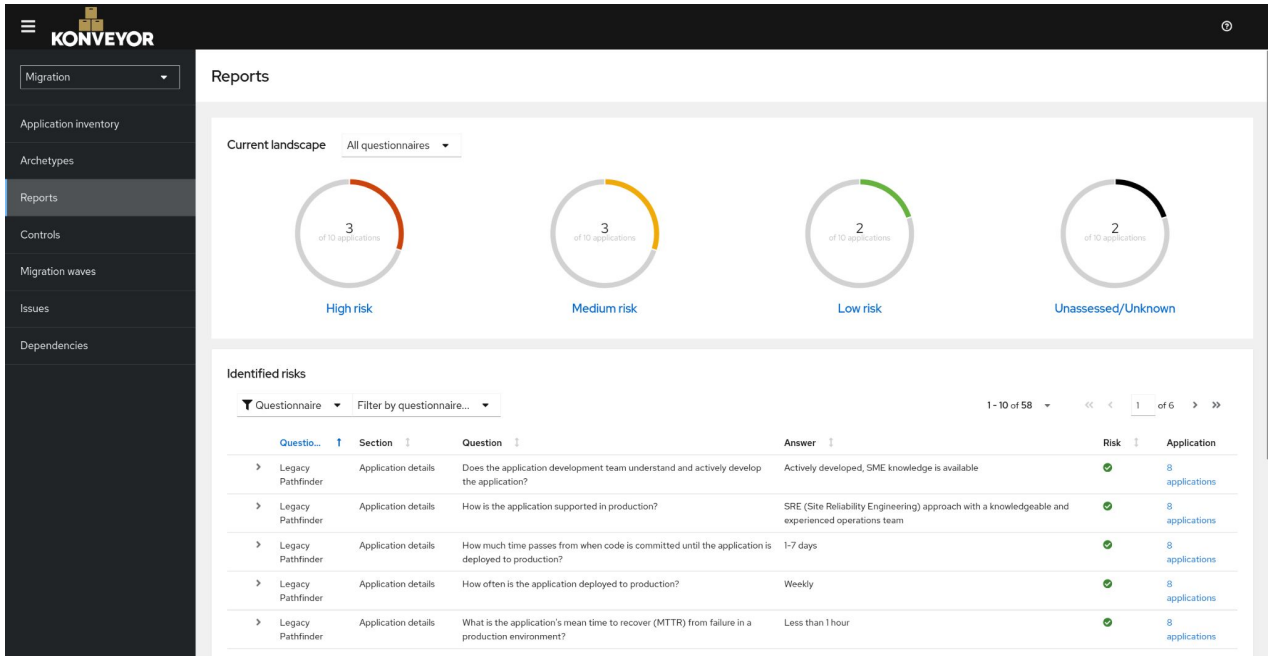
- Integration with source code and binaries repositories:
  - Git
  - Subversion
  - Maven Artifact repositories
- Secure store for multiple credential types:
  - Source control
  - Maven settings files
  - Proxy
- Credentials are managed by administrators and assigned by architects to applications.



Application Assessment

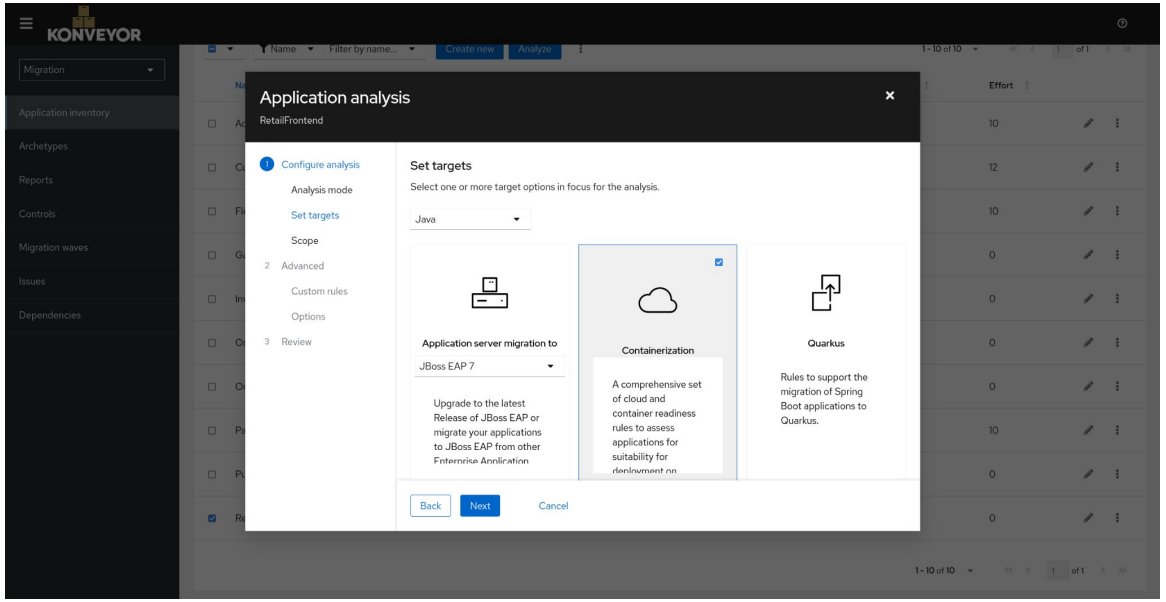
Assess your Application Portfolio for containerization suitability

- A questionnaire based tool that assesses the suitability of applications for deployment in containers within an enterprise Kubernetes platform
- The reports provide information about the suitability of the applications for containerization, highlighting risks and producing an adoption plan informed by effort, priority and dependencies



Application Analysis

Get precise data about your Application Portfolio and estimate migration cost



- Analyzes application source code and binaries and helps estimating the migration effort for different targets or paths.
- Analyzes applications executing an extensible set of rules to identify issues.
- Support numerous migration paths and creates a rich set of reports.
- Currently supports Java and Go.
  - Subsequent minor releases will include support for .NET, Typescript and Python. More languages to come.

Application Analysis

Issue type analysis and support for effort estimation

KONVEYOR

Migration

- Application inventory
- Archetypes
- Reports
- Controls
- Migration waves
- Issues
- Dependencies

Issues

This report provides a concise summary of all issues identified.

All issues | Single application

Application name | Filter by application name...

1 - 5 of 5 | 1 of 1

Issue	Category	Source	Target(s)	Effort	Affected applications
File system - Java IO	mandatory	None	cloud-readiness	1	4
Hardcoded IP Address	mandatory	None	cloud-readiness	1	4
JBoss API reference	potential	java, 2 more	eap, 1 more	0	3

Total affected applications
4 - View affected applications
Target technologies
cloud-readiness
Source technologies
None
Rule set
cloud-readiness
Rule
local-storage-00001
Labels
konveyor.io/source, storage

File system - Java IO
- OpenShift Container Platform: Input secrets and ConfigMaps
- OpenShift Container Platform: Understanding cluster logging
- OpenShift Container Platform: Understanding persistent storage
- Twelve-Factor App: Backing services
- Twelve-Factor App: Config
- Twelve-Factor App: Logs

Application Analysis

Issue identification and guidance for developers

KONVEYOR

Migration

Application inventory

Archetypes

Reports

Controls

Migration waves

Issues

Dependencies

Affected applications

/addon/source/tackle-testapp/src/main/java/io/konveyor/demo/ordermanagement/config/...

Incident #1: Line 43 | Incident #2: Line 66

```
33          em.setDataSource(dataSource());
34          em.setPackagesToScan("io.konveyor.demo.orderm
35          em.setJpaVendorAdapter(new HibernateJpaVendor
36          em.setJpaProperties(additionalProperties());
37
38          return em;
39      }
40
41      @Bean
42      public DataSource dataSource() {
43          ApplicationConfiguration config = new Applica
44          final DriverManagerDataSource dataSource = ne
45          dataSource.setDriverClassName(config.getProp
46          dataSource.setUrl(config.getProperty("jdbc.u
47          dataSource.setUsername(config.getProperty("j
48          dataSource.setPassword(config.getProperty("j
49
50          return dataSource;
51      }
52
53      @Bean
```

Legacy configuration with io.konveyor.demo.config.App...

Line 43

The legacy ApplicationConfiguration class is being used in this application. This is discouraged by the migration guidelines, and should be replaced by a more standard approach using Spring's @PropertySource annotation and Environment class:

```
@PropertySource("classpath:persistence.properties")
public class PersistenceConfig
}}
```

This allows externalizing the configuration in Kubernetes by injecting it as a ConfigMap or a Secret in the lib directory from the container running the Tomcat instance.

Baeldung - Properties with Spring and Spring Boot

Mkyong - Spring @PropertySource example

Spring Documentation - PropertySource javadoc

Close

Application Analysis

Technology identification

KONVEYOR

Migration

- Application inventory
- Archetypes
- Reports
- Controls
- Migration waves
- Issues
- Dependencies

Application inventory

Name | Filter by name... | Create new | Analyze

Name	Business Service	Assessment	Review	Analysis
AccountsReceivable	Finance and HR	Completed	Completed	Completed
Customers	Retail	Completed	Completed	Completed
Flexicard	Finance and HR	Completed	Completed	Completed
Gateway	Retail	Completed	Completed	Not started
Inventory	Retail	Completed	Completed	Not started
OrangeHRM	Finance and HR	Not started	Not started	Not started
Orders	Retail	Completed	Completed	Not started
Payroll	Finance and HR	Completed	Completed	Completed
PurchaseOrders	Finance and HR	Completed	Completed	Not started
RetailFrontend	Retail	Not started	Not started	Not started

Name
Customers

Details | Tags | Reports | Reviews

Filter by: Source 1 | Tag category

Source Analysis × | Clear all filters

Analysis

Bean
EJB XML

HTTP
Servlet

Integration
Micrometer

Inversion of Control
Spring DI

Other
JNI | Properties

Persistence
Spring Data JPA

Web
Spring Web

Connect
JNI | Servlet | EJB XML

Application Analysis

Dependencies identification

KONVEYOR

Migration

- Application inventory
- Archetypes
- Reports
- Controls
- Migration waves
- Issues
- Dependencies

Dependencies

Application name | Filter by application name...

Dependency name	Language
antlr.antlr	java
ch.qos.logback.logback-classic	java
ch.qos.logback.logback-core	java
com.fasterxml.classmate	java
com.fasterxml.jackson.core.jackson-annotations	java
com.fasterxml.jackson.core.jackson-core	java
com.fasterxml.jackson.core.jackson-databind	java
com.fasterxml.jackson.datatype.jackson-datatype-jsr310	java
com.oracle.database.jdbc.ojdbc11	java

Dependency / Language

ch.qos.logback.logback-classic / java

Applications

Application Name | Filter by name...

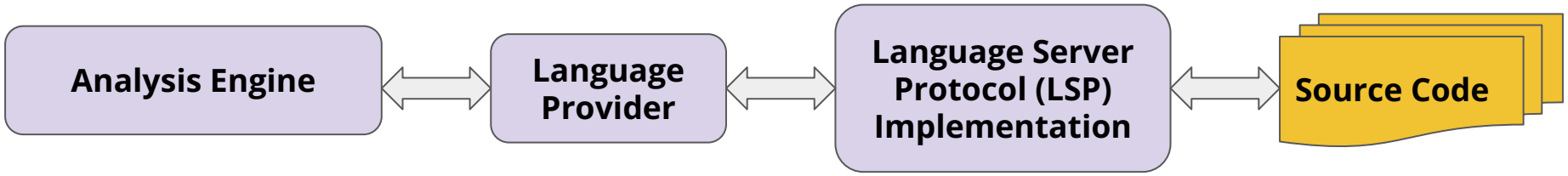
1 - 4 of 4

Application	Version	Management	Relationship
AccountsReceivable	1.1.7	Managed	Direct
Customers	1.1.7	Managed	Direct
Flexicard	1.1.7	Managed	Direct
Payroll	1.1.7	Managed	Direct

1 - 4 of 4

Application Analysis

Multilanguage overview



- Support multiple languages for analysis by abstracting away the analysis engine and delegating the complexity of each language to a Language Server Protocol implementation.
- Language Providers will integrate with the implementation of a given language and translate rules conditions into LSP queries.
- Support for languages Java, Python, .net and generic stuff like YAML, XML, …
- Rules are YAML formatted definitions of anti-patterns checks that are executed on analyzed code, provided out-of-the box and allow user to create and use their custom rules.

Application Analysis

Provided rulesets and custom rules

```yaml
- category: mandatory
  customVariables: []
  description: Oracle JMS Session
  effort: 1
  labels:
  - konveyor.io/source=weblogic
  - konveyor.io/target=eap7
  - konveyor.io/target=eap
  - jms
  - weblogic
  links:
  - title: Java EE 7 - JMS Session
    url: https://docs.oracle.com/javaee/7/tutorial/jms-concepts003.htm#BNCEN
  message: "\n Oracle JMS sessions are used for producing and consuming messaging
    API objects such as message producers, message\n consumers, messages, queue browsers,
    and temporary queues and topics.\n\n This reference should be replaced with the
    Java EE\n standard API: `javax.jms.Session`.\n "
  ruleID: weblogic-jms-eap7-01000
  when:
    java.referenced:
      pattern: oracle.jms.AQjmsSession
```

- Konveyor provided rules from https://github.com/konveyor/rulesets
- Custom rules for your own Frameworks
  - “If you encounter this - here is how you migrate”.
- Also great for large engagements, once you have built your “cookbook”.
- Provide your internal guidance and link directly to your documentation.
- Fully documented YAML syntax aimed at simplifying rules authoring.

Konveyor from CLI

Kantra CLI

Features

- Runs application analysis locally.
  - Generates static HTML reports for analysis results.
  - Helps testing custom rules easily.
- Leverages Podman to avoid complex installations.
- Provides flexibility for automation and enables simple integration with CI/CD pipelines.
- Transforms legacy XML rules into the new YAML syntax.
- Leverages OpenRewrite to automate source code changes

```
kantra analyze --input /home/user/myapp --output /home/user/reports --target eap7
 --target cloud-readiness --source weblogic
```

Intro

Methodology

Konveyor application

<u>Development</u>

Conclusion

Platform Awareness

Integration with platforms to gather application insights

- Enable Konveyor to retrieve information about applications directly from the platform in which they are running:
  - Deployment configuration
  - Runtime configuration
- Flexible enough to obtain information from multiple platform types:
  - Container platforms
  - Application servers
  - Hypervisors and VMs
  - ...

Assets Generation

Generate custom tailored deployment and configuration assets

- Flexible enough to generate all assets required to deploy an application on k8s (and potentially other platforms in the future)
- Provide opinionated best practices out of the box.
- Allow organizations to create their own corporate assets easily:
  - Use templating as much as possible.
  - Build on industry standards
  - Avoid having to learn new programming languages or proprietary APIs.

Kai

Leverage GenAI for application modernization and migration

- Short for Konveyor AI
- Goal: Automate source code changes as much as possible, even when applications use custom corporate technologies and frameworks. Integrated in IDE.
- Leverage the structured migration data stored in Konveyor to enhance commercial LLMs via prompt engineering.

- https://www.konveyor.io/blog/kai-deep-dive-2024/

Intro

Methodology

Konveyor application

Development

Conclusion

That’s it!

Your questions?

- Contact and more information: https://konveyor.io
- Our source code: github.com/konveyor
- Open community, experience sharing, join us!

“The ultimate Open Source toolkit to help organizations safely migrate and modernize their application portfolio to leverage Kubernetes and Cloud-Native technologies, providing differential value on each stage of the adoption process”

KONVEYOR

Thank you!

KONVEYOR