From Legacy to Cloud Native: Transforming Applications with Konveyor AI

devconf.cz, Brno, 13. 6. 2025

Marek Aufart

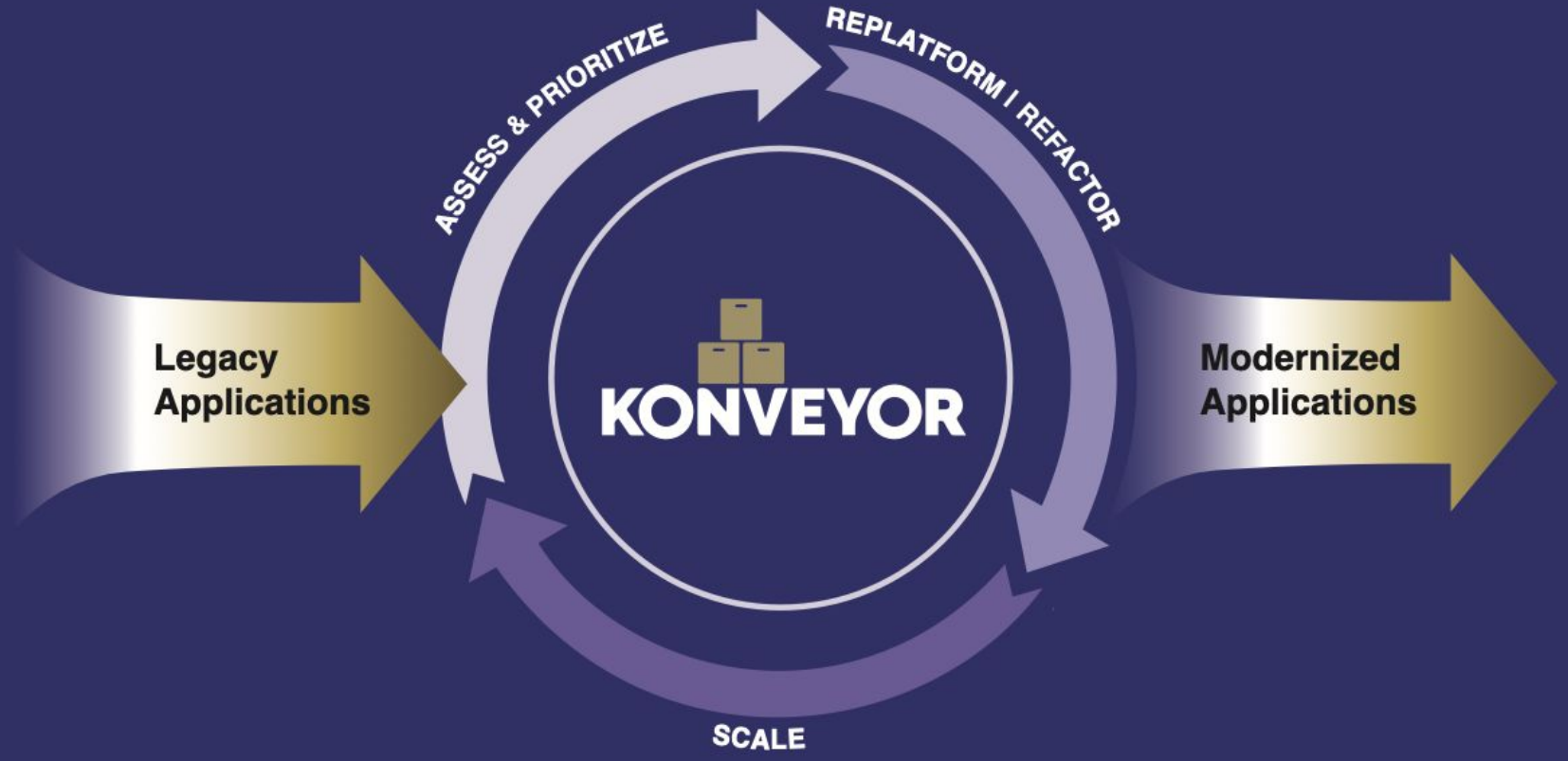
maufart@redhat.com, github.com/aufi

Intro

Konveyor

Konveyor AI / Kai

Conclusion



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 project

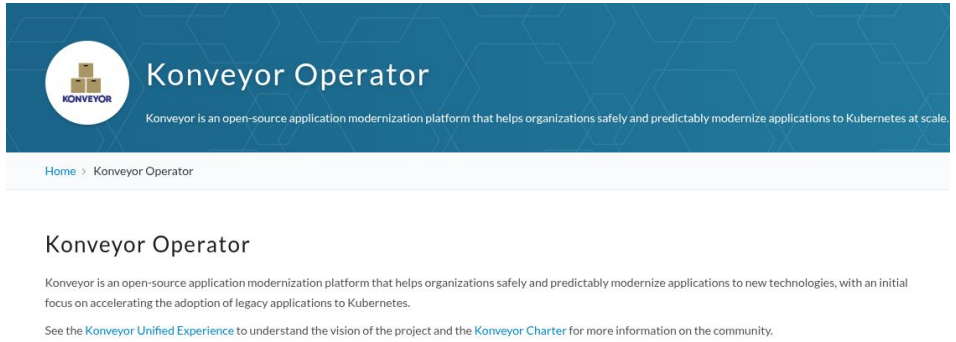
- Konveyor (CNCF sandbox project)
  - Community that helps modernize applications by providing open source tools to rehost, replatform, and refactor applications to Kubernetes and cloud-native technologies.
- Replatform & Refactor approach and methodology
- Originated in Windup tool for Java

Konveyor

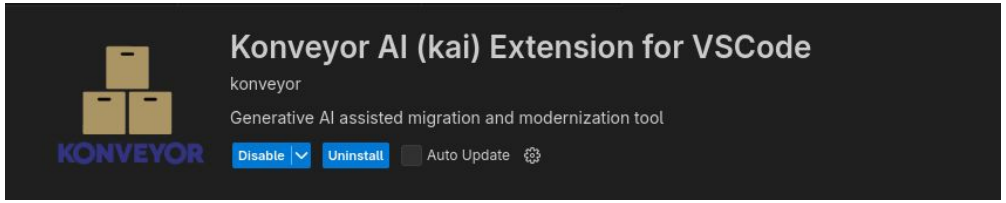
Key components and tooling

- K8s operator with web UI
  - Application portfolio, questionnaires
  - Analyzer
- CLI - “kantra” binary
  - Analyzer
  - Both container and container-less
- IDE - VS code plugin
  - IDE integration, AI tooling
  - Analyzer

- Foundation components
  - Analyzer
  - Rulesets

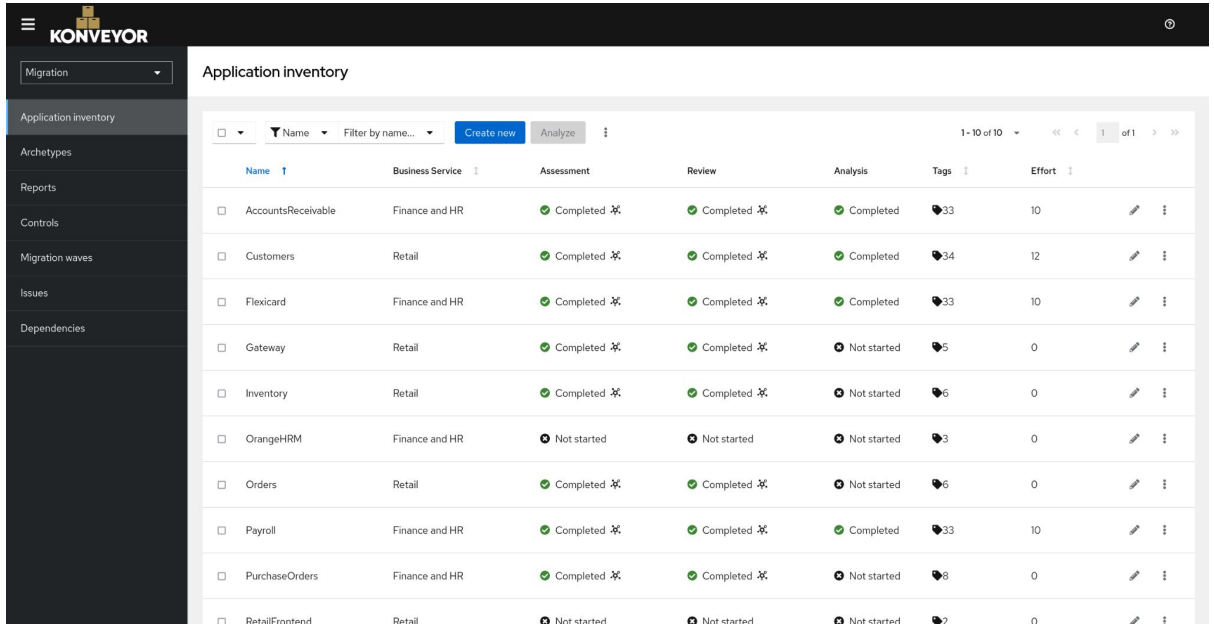



```
maufart@nb:~$ kantra
A CLI tool for analysis and transformation of applications

Usage:
  kantra [command]

Available Commands:
  analyze     Analyze application source code
  completion  Generate the autocompletion script for the specified shell
  help        Help about any command
  test        Test YAML rules
  transform   Transform application source code or windup XML rules
  version     Print the tool version

Flags:
  -h, --help              help for kantra
      --log-level uint32  log level (default 4)
      --no-cleanup        do not cleanup temporary resources

Use "kantra [command] --help" for more information about a command.
```



Application Inventory

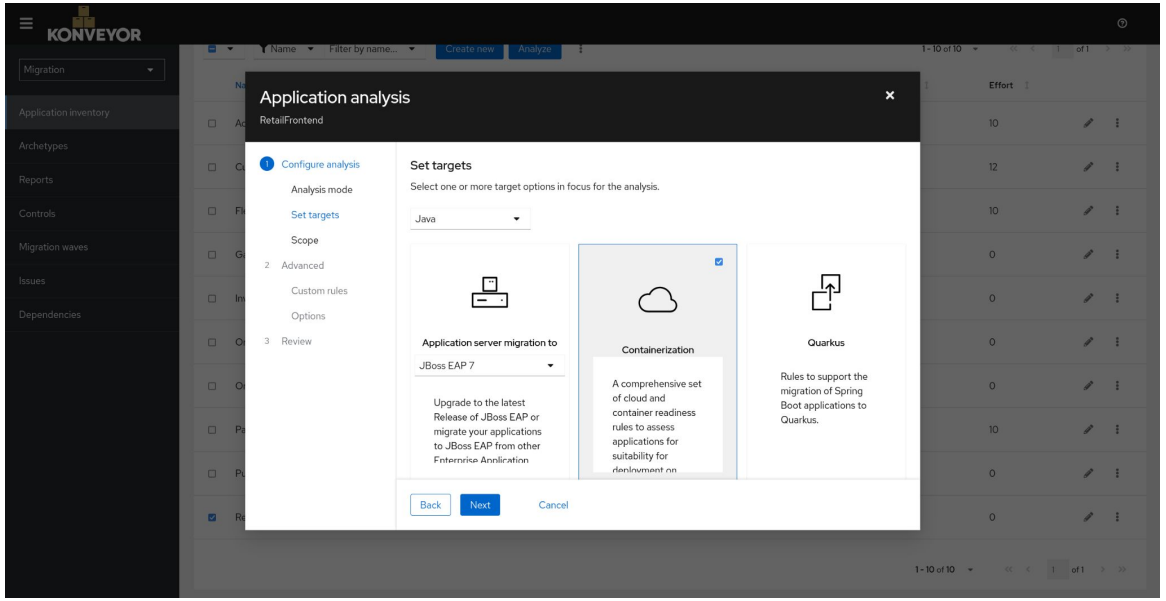
Konveyor UI



- 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 Analysis

Get precise data about your Application Portfolio and estimate migration cost



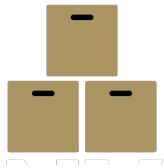
- Analyzes application source code and binaries and helps estimating the migration effort for different targets or paths.
- Analyzes applications executing an extensible set of rules to identify issues.
- Support numerous migration paths and creates a rich set of reports.
- Currently supports Java, .NET,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

Analysis Rules

Provided rulesets and custom rules

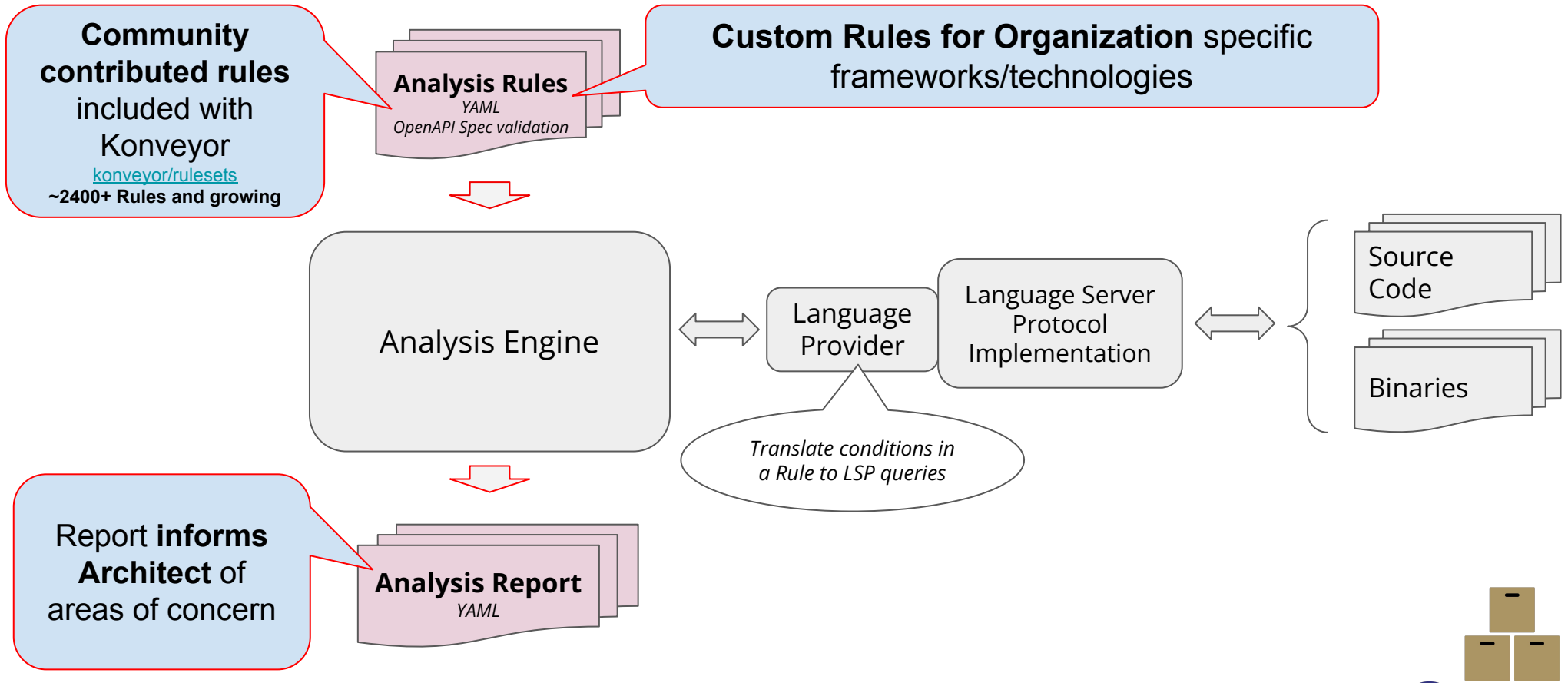
```
- category: mandatory
  customVariables: []
  description: Oracle JMS Session
  effort: 1
  labels:
  - konveyor.io/source=weblogic
  - konveyor.io/target=eap7
  - konveyor.io/target=eap
  - jms
  - weblogic
  links:
  - title: Java EE 7 - JMS Session
    url: https://docs.oracle.com/javaee/7/tutorial/jms-concepts003.htm#BNCEN
  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
```

- “If you encounter this - here is how you migrate”
- Custom rules for your own Frameworks
- 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 provided rules https://github.com/konveyor/rulesets



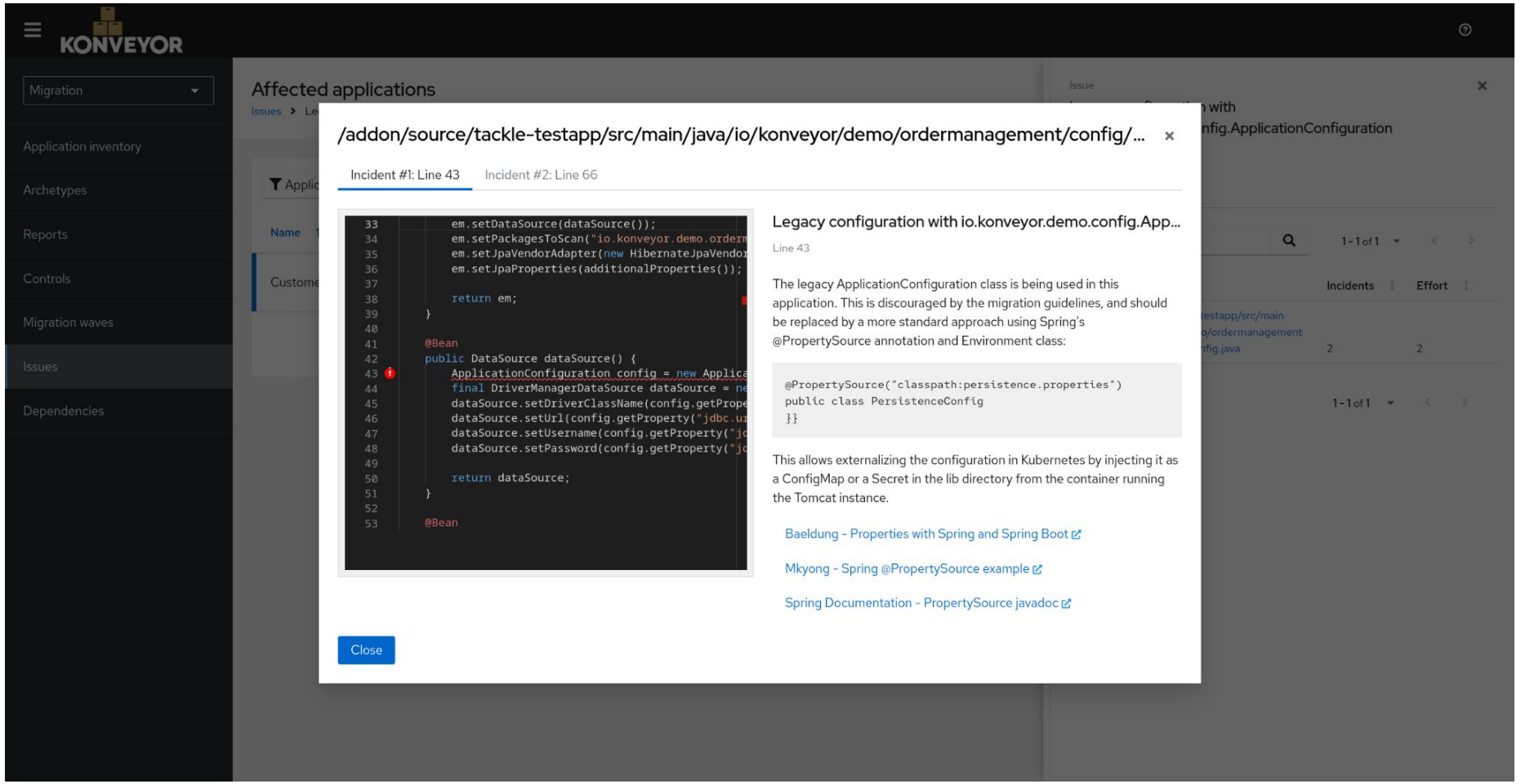
Analysis flow

Community Knowledge + Custom Rules



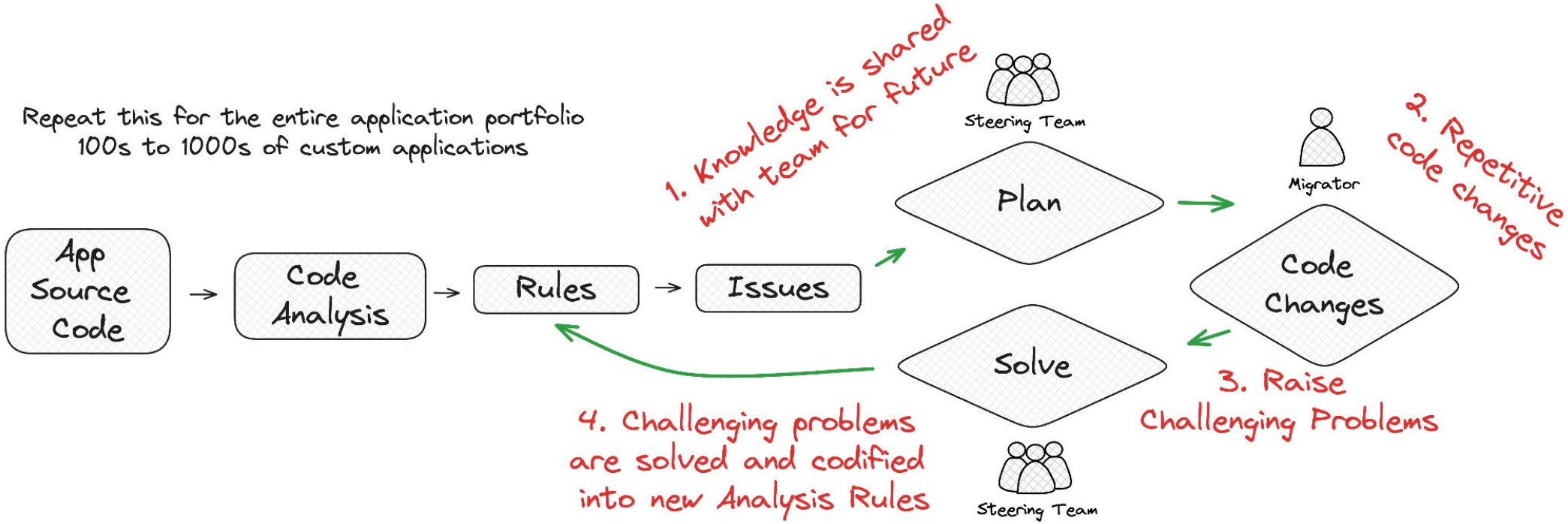
Application Analysis

Issue identification and guidance for developers



Modernization running at Scale

General Pattern: “Migration Factory”



Intro

Konveyor

Konveyor AI / Kai

Conclusion

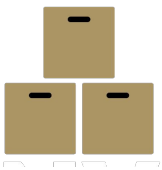
Konveyor AI (Kai)

Goal and AI use generally

Goal: Improve the economics of re-platforming and refactoring applications to Kubernetes and cloud-native technologies by leveraging Generative AI

Approach:

- Expand Konveyor capabilities beyond surfacing information and beyond OpenRewrite capabilities
- Generate code suggestions for discovered migration issues
  - Work with LLMs using structured migration data Konveyor has collected
- Avoid fine-tuning each model by using Retrieval Augmented Generation (RAG)
  - Shape LLM results with examples of how Organization has solved similar problems in the past
- Model agnostic
  - Integration with commercial models
  - Ability to bring your own model



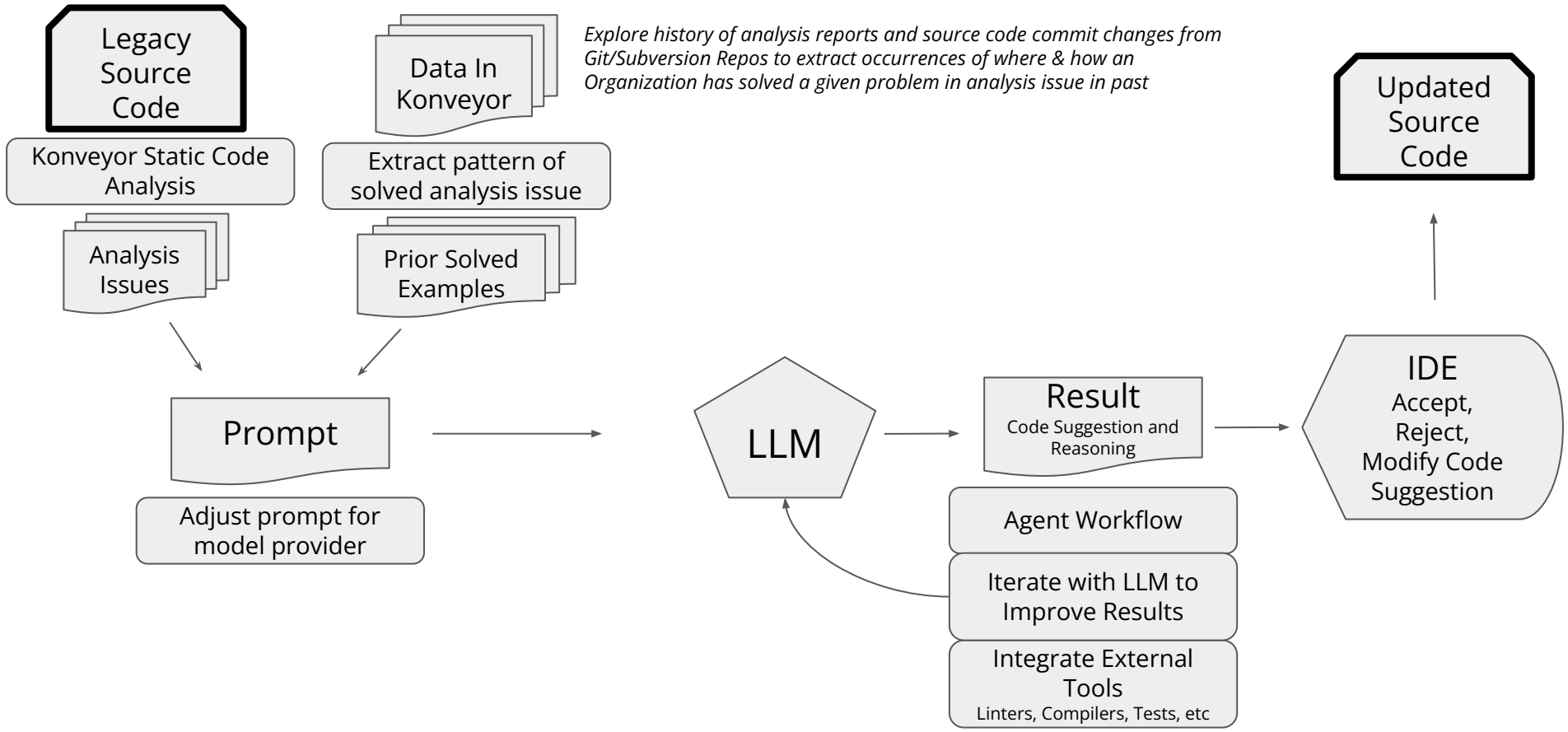
Konveyor AI (Kai)

Generative AI to automate source code changes between technologies

- **Uses data in Konveyor** to generate code suggestions, primary Java
  - Source code analysis with Rules
    - Pinpoint issues to adopting a new technology
  - Changelog history from source repositories
    - Before/After of previously solved issues
- **Crafts a tailored LLM prompt** based on:
  - Knowledge of specific issues that need to be addressed
  - Knowledge of prior successful code changes
- **Agentic workflows** to leverage external tools
- Provides suggested code changes via **IDE plugin**

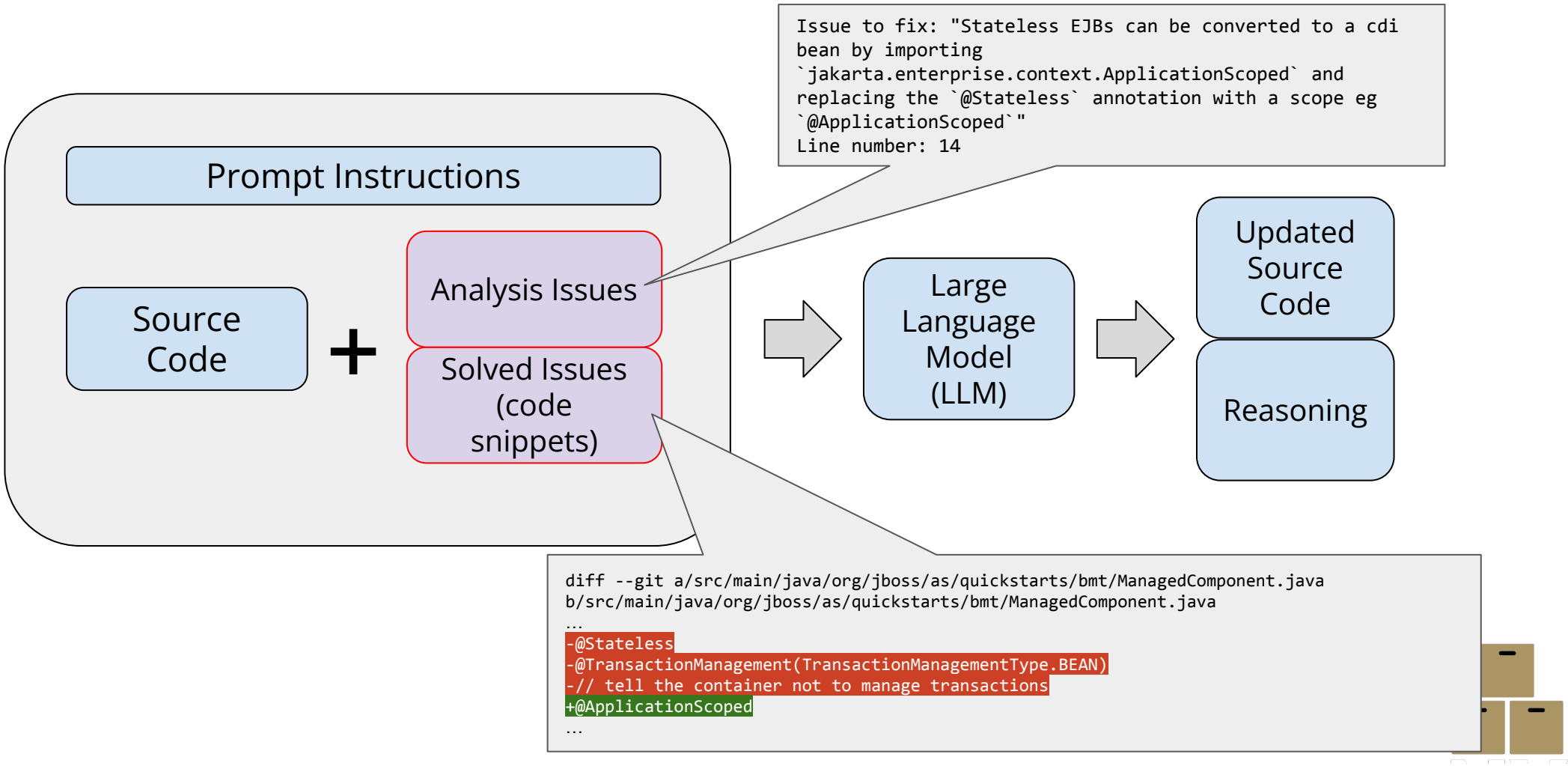
Konveyor AI (Kai)

Overall diagram



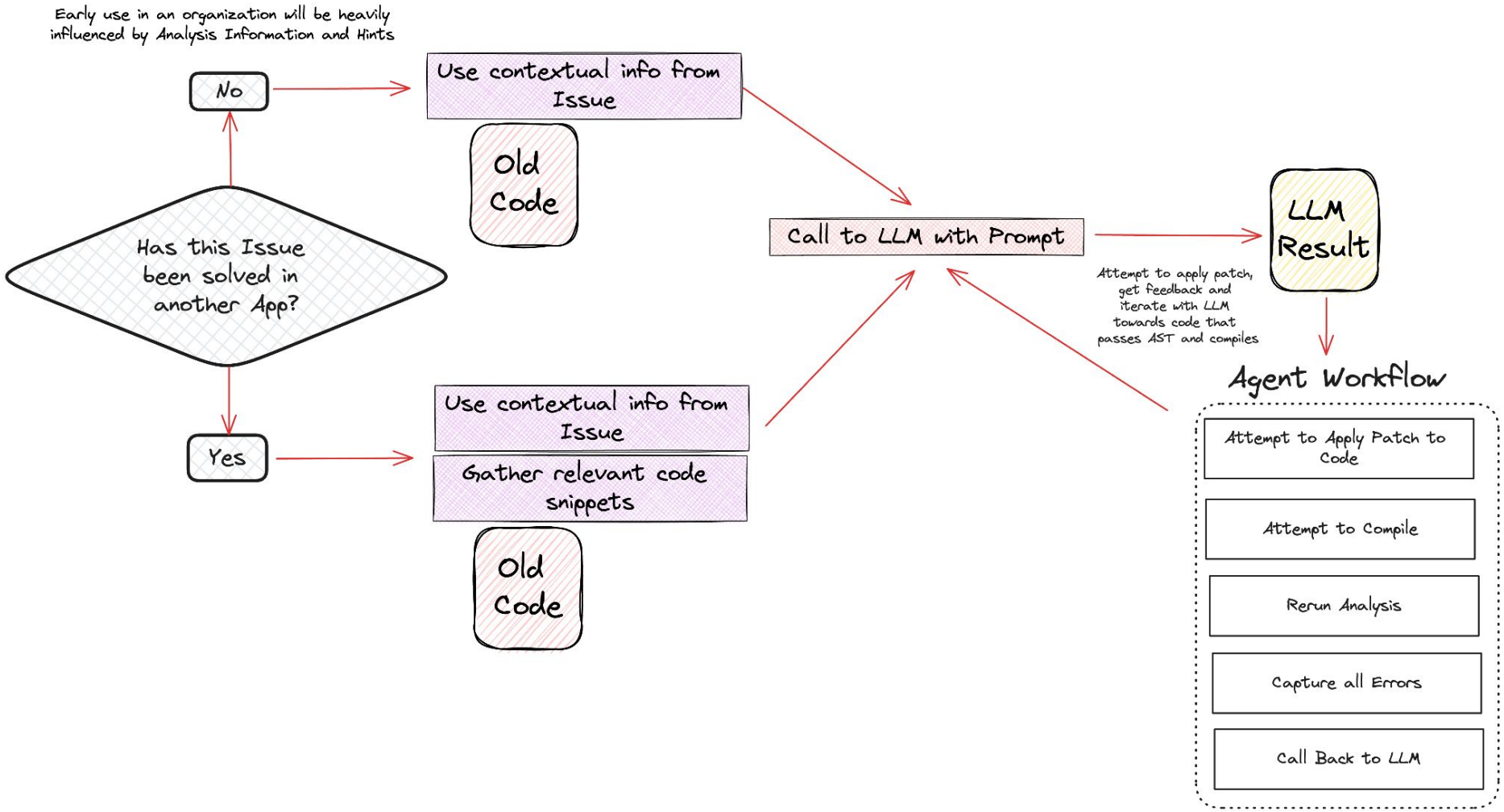
Prompt approach

Retrieval Augmented Generation (RAG)



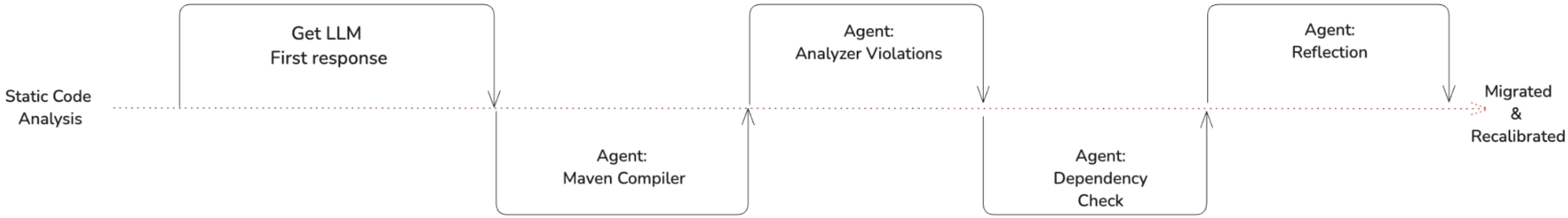
Agentic Workflows

Solved issues and result refinement through post processing



Agentic Workflows

Result refinement steps



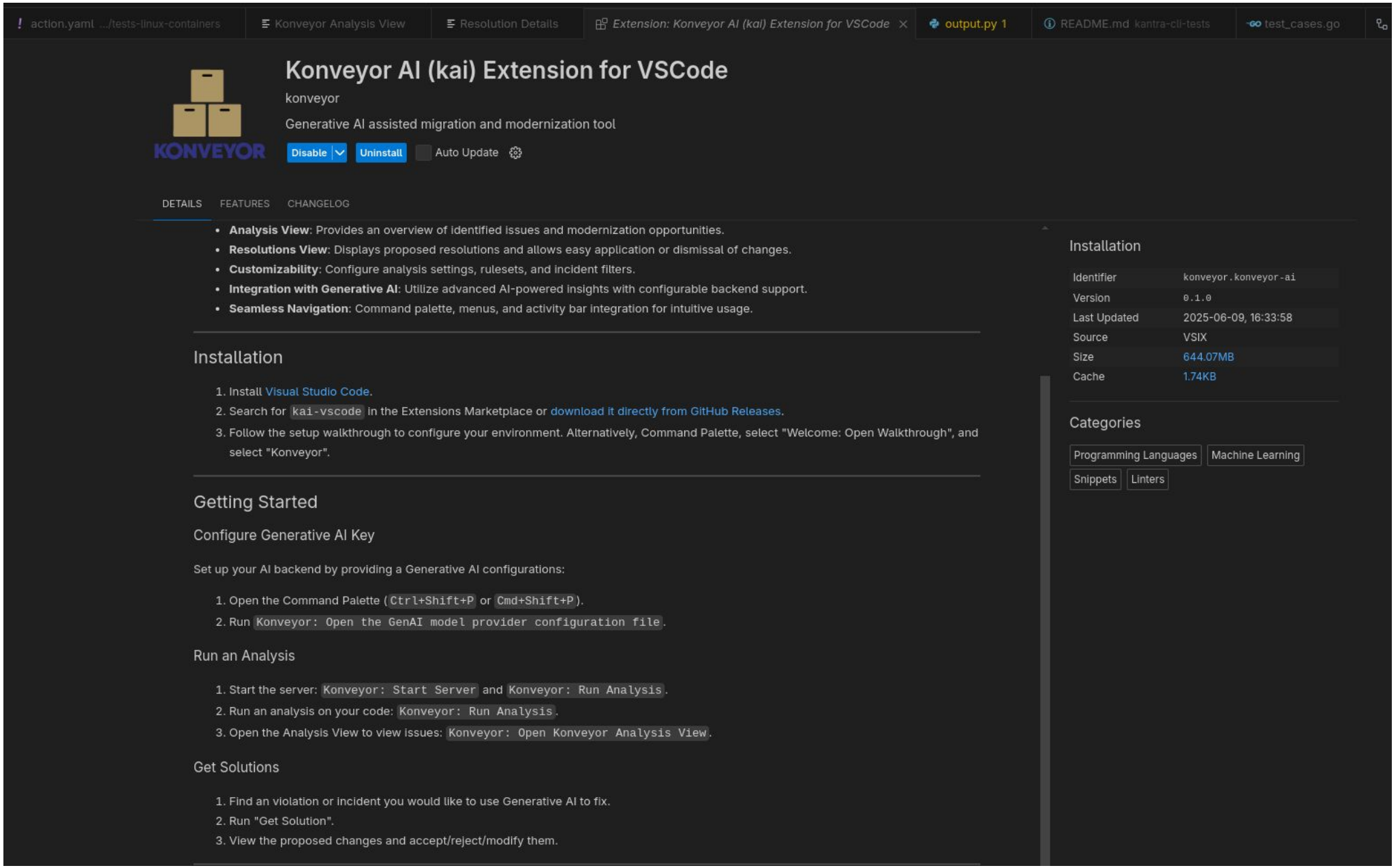
- “Its not just about a single response from the LLM”
- Sanitization of responses
- Compilation
- Understanding that it works as per the context
- Ensuring the code is usable again.

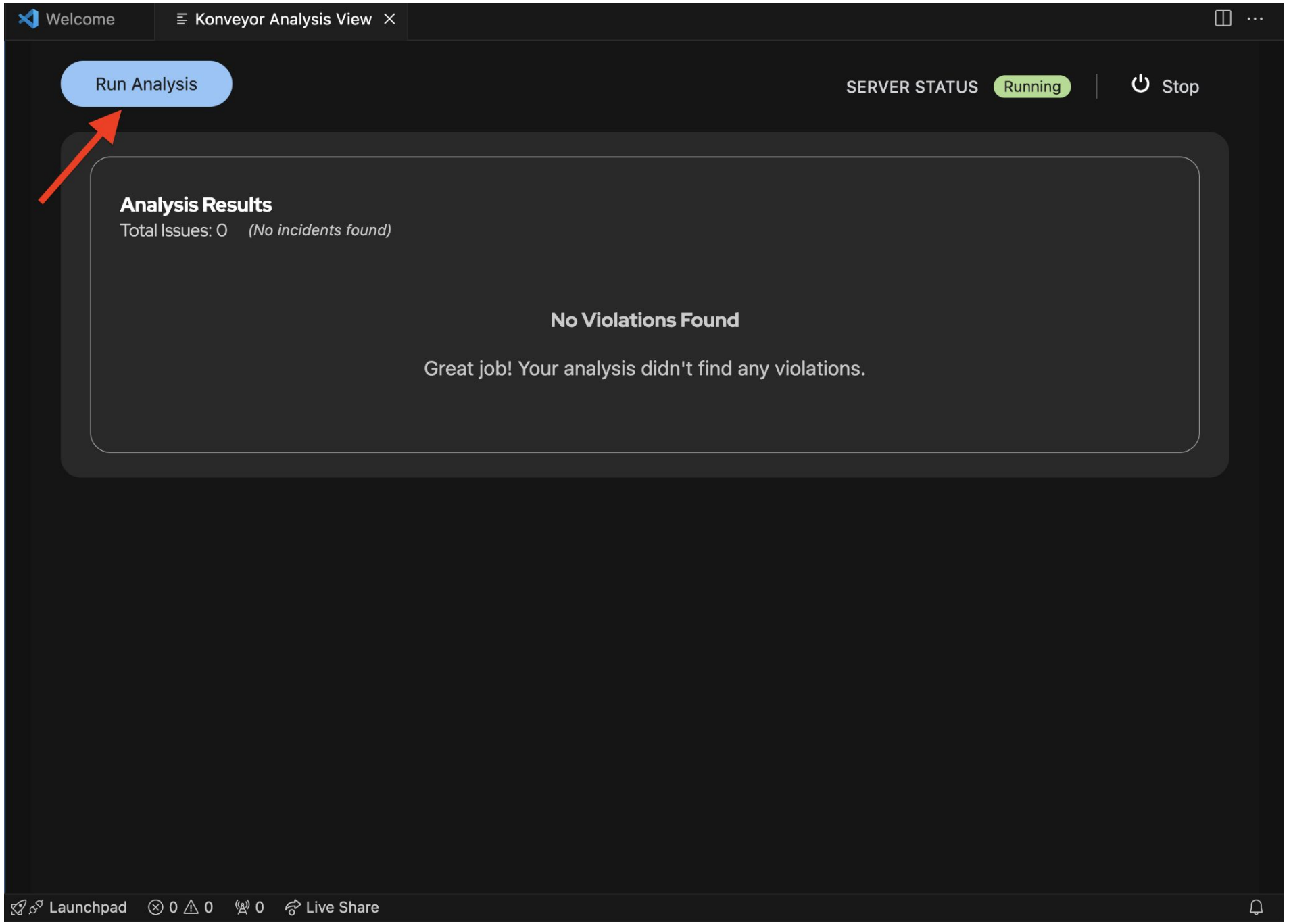
Konveyor AI (Kai)

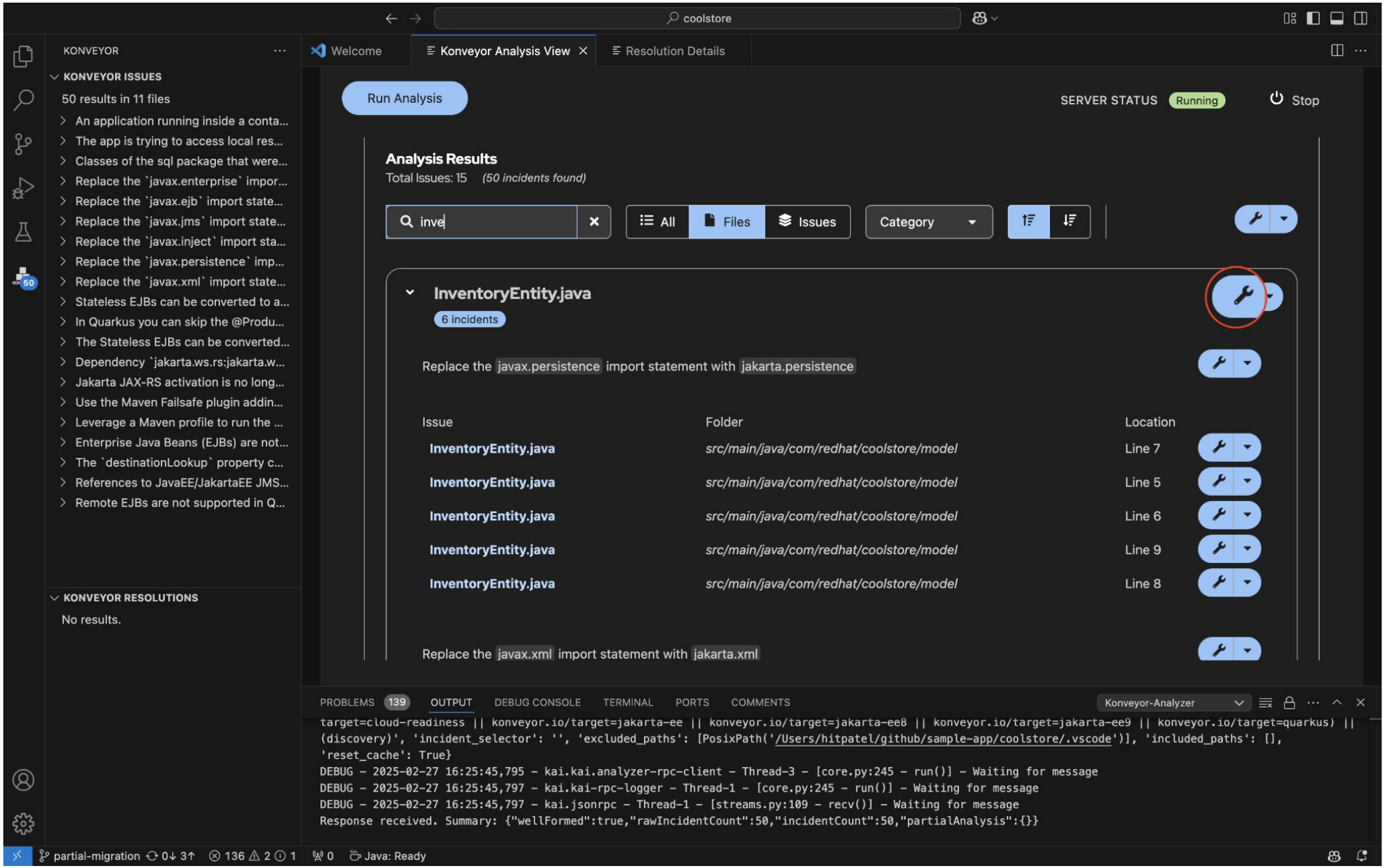
Try it

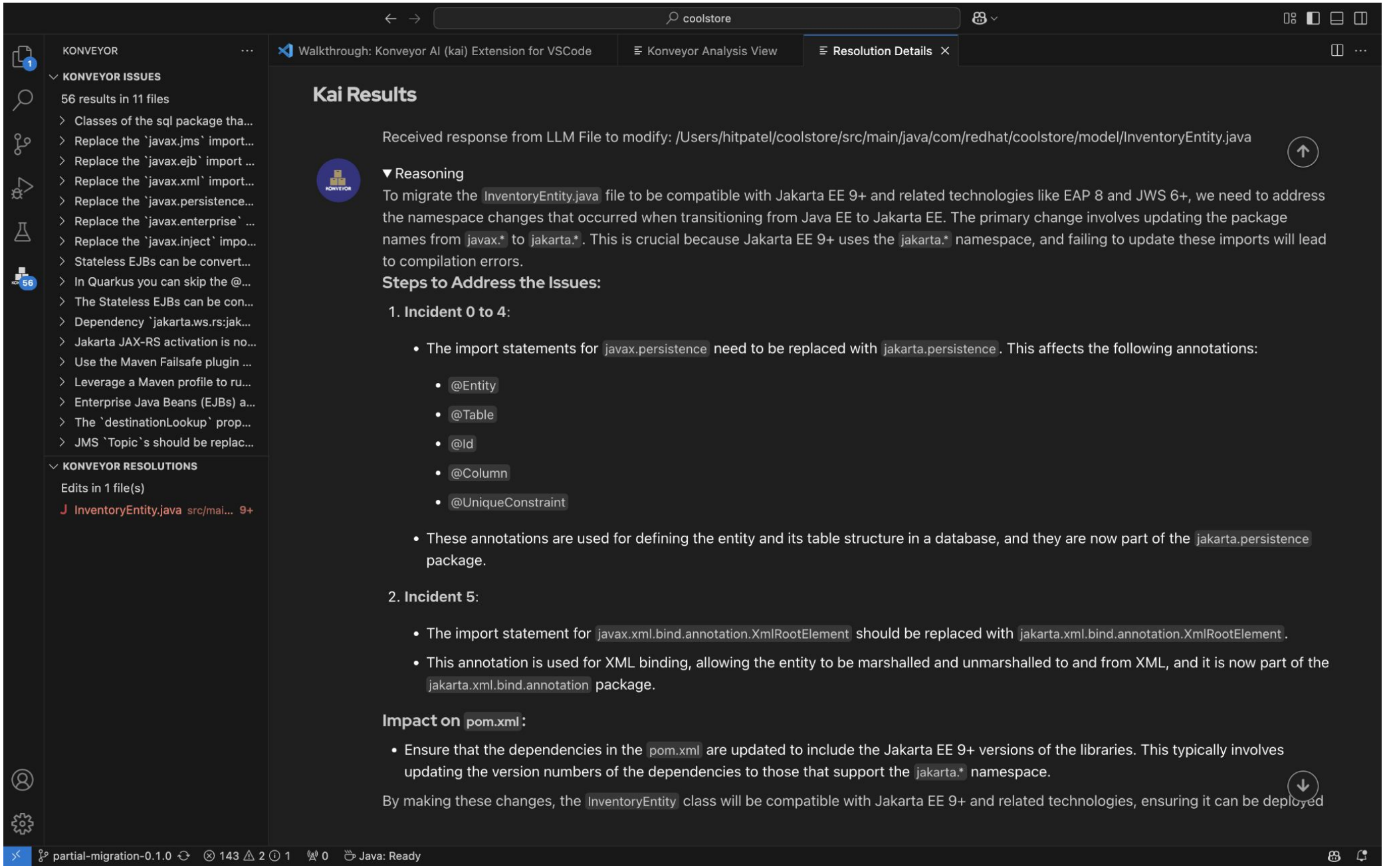
- Installation via Editor Extension (vsix)
  https://github.com/konveyor/editor-extensions/releases
- v0.1.0 release from March
  - https://konveyor.io/blog/2025/kai-release-announcement/
  - And we're progressing

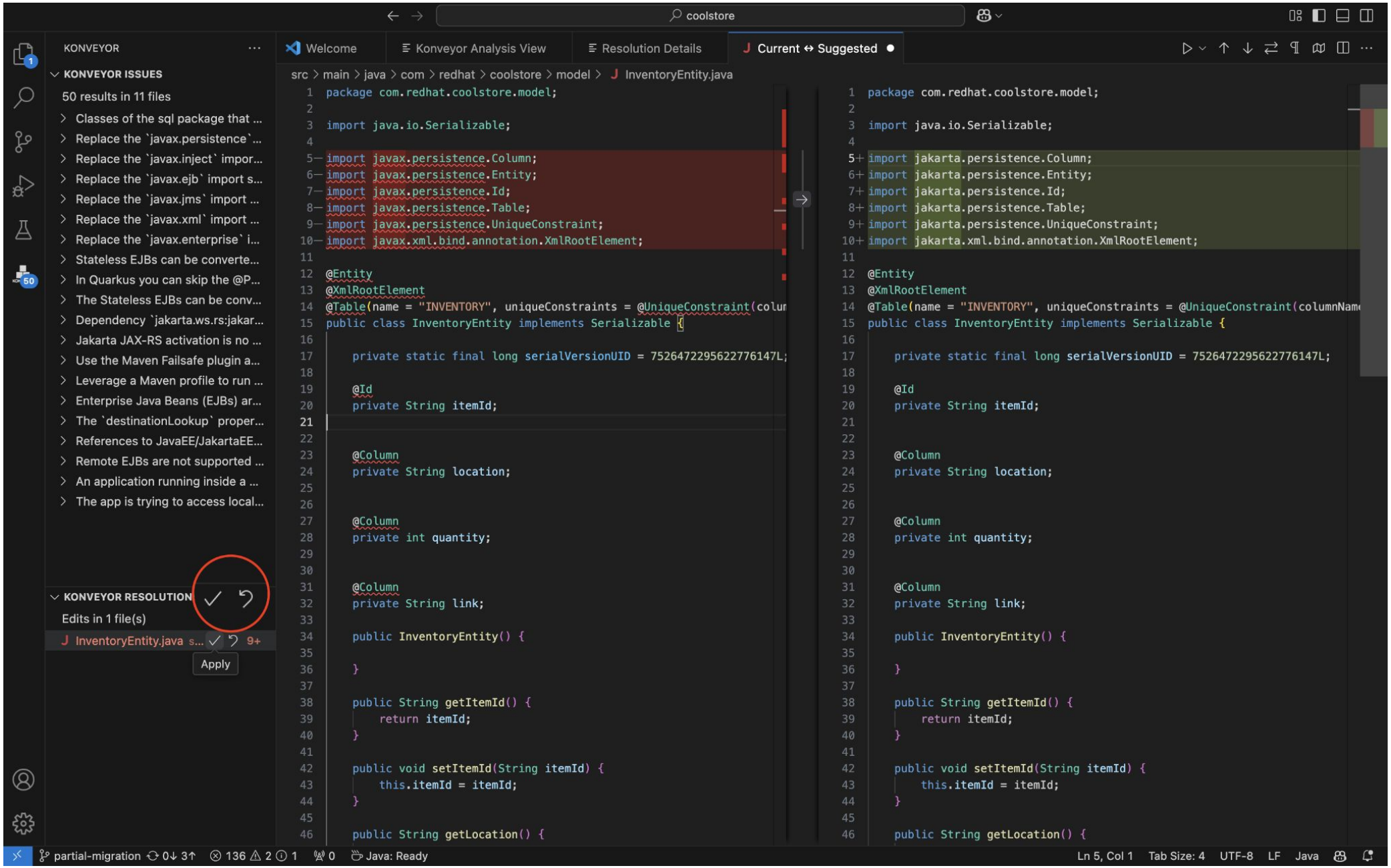
- Getting started
  - https://github.com/konveyor/kai/blob/main/docs/getting_started.md
  - Follow "guided scenarios", primary Java

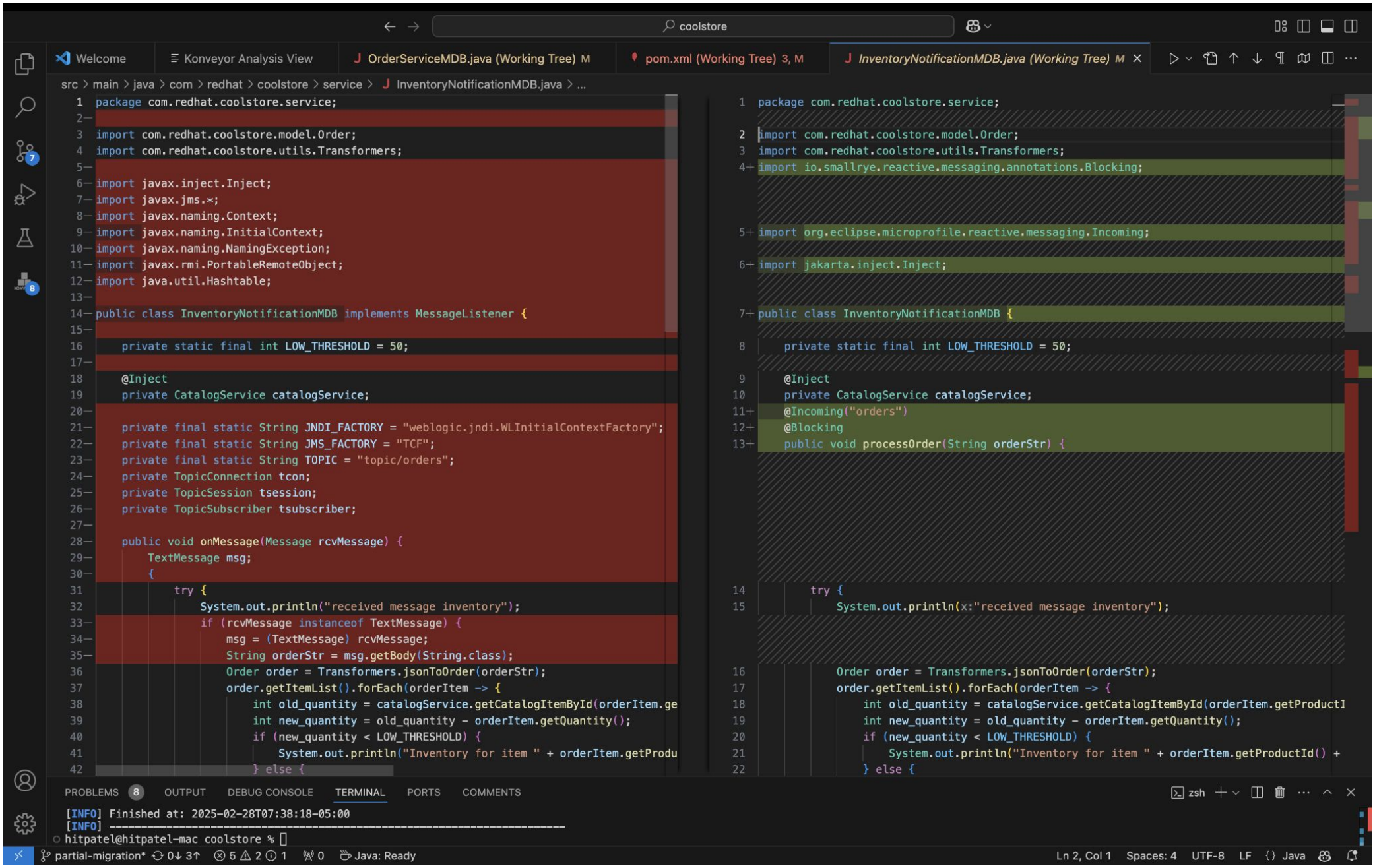

action.yaml .../tests-linux-containers
Konveyor Analysis View
Resolution Details
Extension: Konveyor AI (kai) Extension for VSCode
output.py 1
README.md kantra-cli-tests
test_cases.go
Konveyor AI (kai) Extension for VSCode
konveyor
Generative AI assisted migration and modernization tool
Disable
Uninstall
Auto Update
DETAILS
FEATURES
CHANGELOG
Analysis View: Provides an overview of identified issues and modernization opportunities.
Resolutions View: Displays proposed resolutions and allows easy application or dismissal of changes.
Customizability: Configure analysis settings, rulesets, and incident filters.
Integration with Generative AI: Utilize advanced AI-powered insights with configurable backend support.
Seamless Navigation: Command palette, menus, and activity bar integration for intuitive usage.
Installation
1. Install Visual Studio Code.
2. Search for kai-vscode in the Extensions Marketplace or download it directly from GitHub Releases.
3. Follow the setup walkthrough to configure your environment. Alternatively, Command Palette, select "Welcome: Open Walkthrough", and select "Konveyor".
Getting Started
Configure Generative AI Key
Set up your AI backend by providing a Generative AI configurations:
1. Open the Command Palette (Ctrl+Shift+P or Cmd+Shift+P).
2. Run Konveyor: Open the GenAI model provider configuration file.
Run an Analysis
1. Start the server: Konveyor: Start Server and Konveyor: Run Analysis.
2. Run an analysis on your code: Konveyor: Run Analysis.
3. Open the Analysis View to view issues: Konveyor: Open Konveyor Analysis View.
Get Solutions
1. Find an violation or incident you would like to use Generative AI to fix.
2. Run "Get Solution".
3. View the proposed changes and accept/reject/modify them.
Installation
Identifier konveyor.konveyor-ai
Version 0.1.0
Last Updated 2025-06-09, 16:33:58
Source VSIX
Size 644.07MB
Cache 1.74KB
Categories
Programming Languages
Machine Learning
Snippets
Linters

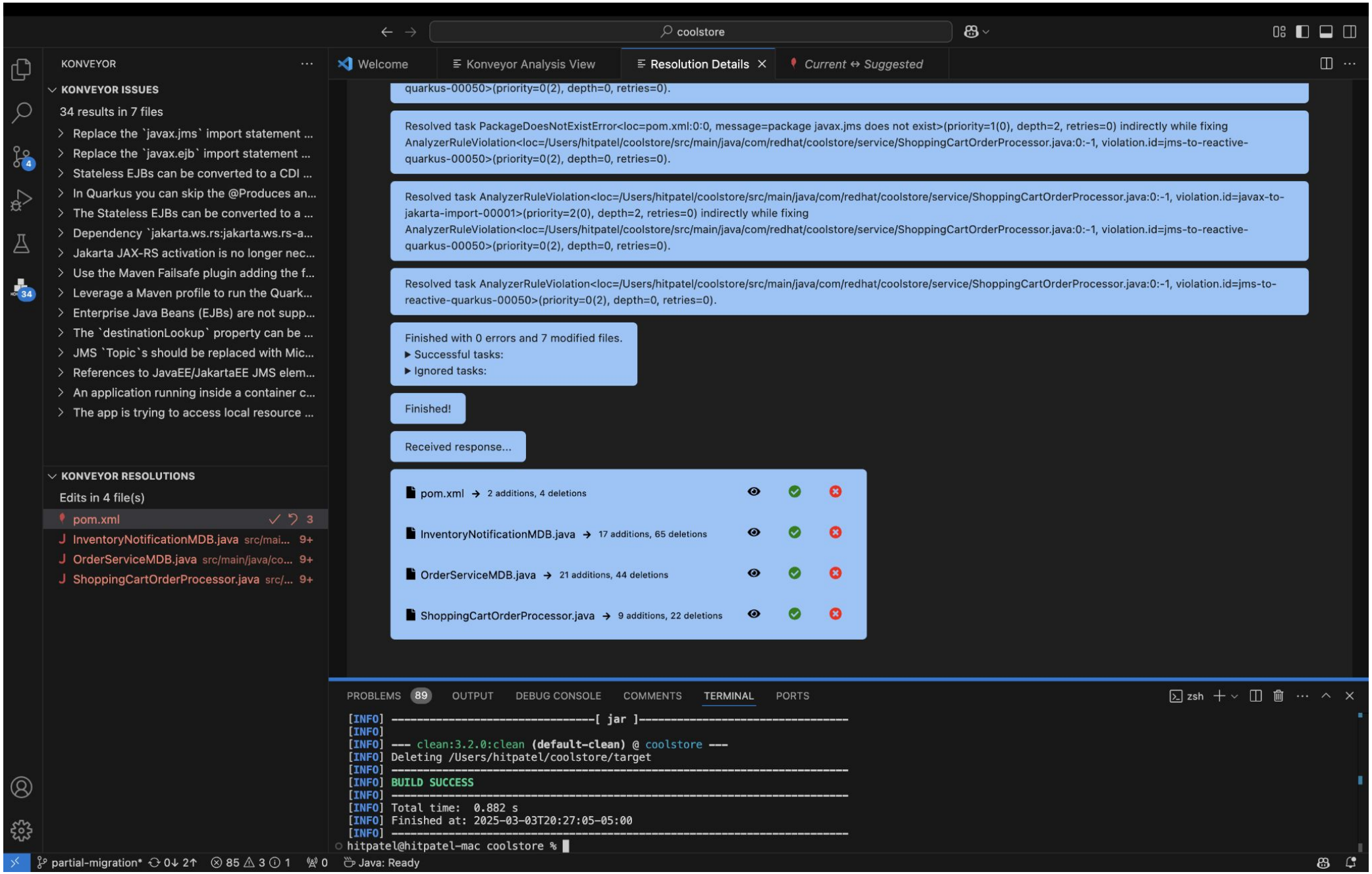

Welcome
Konveyor Analysis View
Run Analysis
SERVER STATUS
Running
Stop
Analysis Results
Total Issues: 0
(No incidents found)
No Violations Found
Great job! Your analysis didn't find any violations.
Launchpad
0
0
0
Live Share


coolstore
KONVEYOR
KONVEYOR ISSUES
50 results in 11 files
An application running inside a conta...
The app is trying to access local res...
Classes of the sql package that were...
Replace the `javax.enterprise` impor...
Replace the `javax.ejb` import state...
Replace the `javax.jms` import state...
Replace the `javax.inject` import sta...
Replace the `javax.persistence` imp...
Replace the `javax.xml` import state...
Stateless EJBs can be converted to a...
In Quarkus you can skip the @Produ...
The Stateless EJBs can be converted...
Dependency `jakarta.ws.rs:jakarta.w...
Jakarta JAX-RS activation is no long...
Use the Maven Failsafe plugin addin...
Leverage a Maven profile to run the ...
Enterprise Java Beans (EJBs) are not...
The `destinationLookup` property c...
References to JavaEE/JakartaEE JMS...
Remote EJBs are not supported in Q...
KONVEYOR RESOLUTIONS
No results.
Welcome
Konveyor Analysis View
Resolution Details
Run Analysis
SERVER STATUS
Running
Stop
Analysis Results
Total Issues: 15
(50 incidents found)
inve
All
Files
Issues
Category
InventoryEntity.java
6 incidents
Replace the javax.persistence import statement with jakarta.persistence
Issue
Folder
Location
InventoryEntity.java
src/main/java/com/redhat/coolstore/model
Line 7
InventoryEntity.java
src/main/java/com/redhat/coolstore/model
Line 5
InventoryEntity.java
src/main/java/com/redhat/coolstore/model
Line 6
InventoryEntity.java
src/main/java/com/redhat/coolstore/model
Line 9
InventoryEntity.java
src/main/java/com/redhat/coolstore/model
Line 8
Replace the javax.xml import statement with jakarta.xml
PROBLEMS 139
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
COMMENTS
Konveyor-Analyzer
target=cloud-readiness || konveyor.io/target=jakarta-ee || konveyor.io/target=jakarta-ee8 || konveyor.io/target=jakarta-ee9 || konveyor.io/target=quarkus) ||
(discovery)', 'incident_selector': '', 'excluded_paths': [PosixPath('/Users/hitpatel/github/sample-app/coolstore/.vscode')], 'included_paths': [],
'reset_cache': True}
DEBUG - 2025-02-27 16:25:45,795 - kai.kai.analyzer-rpc-client - Thread-3 - [core.py:245 - run()] - Waiting for message
DEBUG - 2025-02-27 16:25:45,797 - kai.kai-rpc-logger - Thread-1 - [core.py:245 - run()] - Waiting for message
DEBUG - 2025-02-27 16:25:45,797 - kai.jsonrpc - Thread-1 - [streams.py:109 - recv()] - Waiting for message
Response received. Summary: {"wellFormed":true,"rawIncidentCount":50,"incidentCount":50,"partialAnalysis":{}}
partial-migration
0↓ 3↑
136 2 1
0
Java: Ready


coolstore
KONVEYOR
KONVEYOR ISSUES
56 results in 11 files
Classes of the sql package tha...
Replace the `javax.jms` import...
Replace the `javax.ejb` import ...
Replace the `javax.xml` import...
Replace the `javax.persistence...
Replace the `javax.enterprise` ...
Replace the `javax.inject` impo...
Stateless EJBs can be convert...
In Quarkus you can skip the @...
The Stateless EJBs can be con...
Dependency `jakarta.ws.rs:jak...
Jakarta JAX-RS activation is no...
Use the Maven Failsafe plugin ...
Leverage a Maven profile to ru...
Enterprise Java Beans (EJBs) a...
The `destinationLookup` prop...
JMS `Topic`s should be replac...
KONVEYOR RESOLUTIONS
Edits in 1 file(s)
InventoryEntity.java src/mai... 9+
Walkthrough: Konveyor AI (kai) Extension for VSCode
Konveyor Analysis View
Resolution Details
Kai Results
Received response from LLM File to modify: /Users/hitpatel/coolstore/src/main/java/com/redhat/coolstore/model/InventoryEntity.java
▼Reasoning
To migrate the InventoryEntity.java file to be compatible with Jakarta EE 9+ and related technologies like EAP 8 and JWS 6+, we need to address the namespace changes that occurred when transitioning from Java EE to Jakarta EE. The primary change involves updating the package names from javax.* to jakarta.*. This is crucial because Jakarta EE 9+ uses the jakarta.* namespace, and failing to update these imports will lead to compilation errors.
Steps to Address the Issues:
1. Incident 0 to 4:
The import statements for javax.persistence need to be replaced with jakarta.persistence. This affects the following annotations:
@Entity
@Table
@Id
@Column
@UniqueConstraint
These annotations are used for defining the entity and its table structure in a database, and they are now part of the jakarta.persistence package.
2. Incident 5:
The import statement for javax.xml.bind.annotation.XmlRootElement should be replaced with jakarta.xml.bind.annotation.XmlRootElement.
This annotation is used for XML binding, allowing the entity to be marshalled and unmarshalled to and from XML, and it is now part of the jakarta.xml.bind.annotation package.
Impact on pom.xml:
Ensure that the dependencies in the pom.xml are updated to include the Jakarta EE 9+ versions of the libraries. This typically involves updating the version numbers of the dependencies to those that support the jakarta.* namespace.
By making these changes, the InventoryEntity class will be compatible with Jakarta EE 9+ and related technologies, ensuring it can be deployed
partial-migration-0.1.0
143 2 1
0
Java: Ready


coolstore
KONVEYOR
KONVEYOR ISSUES
50 results in 11 files
Classes of the sql package that ...
Replace the `javax.persistence`...
Replace the `javax.inject` impor...
Replace the `javax.ejb` import s...
Replace the `javax.jms` import ...
Replace the `javax.xml` import ...
Replace the `javax.enterprise` i...
Stateless EJBs can be converte...
In Quarkus you can skip the @P...
The Stateless EJBs can be conv...
Dependency `jakarta.ws.rs:jakar...
Jakarta JAX-RS activation is no ...
Use the Maven Failsafe plugin a...
Leverage a Maven profile to run ...
Enterprise Java Beans (EJBs) ar...
The `destinationLookup` proper...
References to JavaEE/JakartaEE...
Remote EJBs are not supported ...
An application running inside a ...
The app is trying to access local...
KONVEYOR RESOLUTION
Edits in 1 file(s)
InventoryEntity.java s...
9+
Apply
Welcome
Konveyor Analysis View
Resolution Details
Current ↔ Suggested
src > main > java > com > redhat > coolstore > model > InventoryEntity.java
package com.redhat.coolstore.model;
import java.io.Serializable;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;
import javax.persistence.UniqueConstraint;
import javax.xml.bind.annotation.XmlRootElement;
@Entity
@XmlRootElement
@Table(name = "INVENTORY", uniqueConstraints = @UniqueConstraint(colum
public class InventoryEntity implements Serializable {
private static final long serialVersionUID = 7526472295622776147L;
@Id
private String itemId;
@Column
private String location;
@Column
private int quantity;
@Column
private String link;
public InventoryEntity() {
}
public String getItemId() {
return itemId;
}
public void setItemId(String itemId) {
this.itemId = itemId;
}
public String getLocation() {
import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.Id;
import jakarta.persistence.Table;
import jakarta.persistence.UniqueConstraint;
import jakarta.xml.bind.annotation.XmlRootElement;
@Table(name = "INVENTORY", uniqueConstraints = @UniqueConstraint(columnNam
partial-migration 0↓ 3↑ 136 2 1 0 Java: Ready
Ln 5, Col 1 Tab Size: 4 UTF-8 LF Java

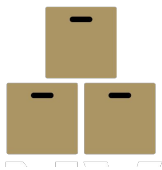

coolstore
Welcome
Konveyor Analysis View
OrderServiceMDB.java (Working Tree) M
pom.xml (Working Tree) 3, M
InventoryNotificationMDB.java (Working Tree) M
src > main > java > com > redhat > coolstore > service > InventoryNotificationMDB.java > ...
package com.redhat.coolstore.service;
import com.redhat.coolstore.model.Order;
import com.redhat.coolstore.utils.Transformers;
import javax.inject.Inject;
import javax.jms.*;
import javax.naming.Context;
import javax.naming.InitialContext;
import javax.naming.NamingException;
import javax.rmi.PortableRemoteObject;
import java.util.Hashtable;
public class InventoryNotificationMDB implements MessageListener {
private static final int LOW_THRESHOLD = 50;
@Inject
private CatalogService catalogService;
private final static String JNDI_FACTORY = "weblogic.jndi.WLInitialContextFactory";
private final static String JMS_FACTORY = "TCF";
private final static String TOPIC = "topic/orders";
private TopicConnection tcon;
private TopicSession tsession;
private TopicSubscriber tsubscriber;
public void onMessage(Message rcvMessage) {
TextMessage msg;
try {
System.out.println("received message inventory");
if (rcvMessage instanceof TextMessage) {
msg = (TextMessage) rcvMessage;
String orderStr = msg.getBody(String.class);
Order order = Transformers.jsonToOrder(orderStr);
order.getItemList().forEach(orderItem -> {
int old_quantity = catalogService.getCatalogItemById(orderItem.ge
int new_quantity = old_quantity - orderItem.getQuantity();
if (new_quantity < LOW_THRESHOLD) {
System.out.println("Inventory for item " + orderItem.getProdu
} else {
import com.redhat.coolstore.model.Order;
import com.redhat.coolstore.utils.Transformers;
import io.smallrye.reactive.messaging.annotations.Blocking;
import org.eclipse.microprofile.reactive.messaging.Incoming;
import jakarta.inject.Inject;
public class InventoryNotificationMDB {
private static final int LOW_THRESHOLD = 50;
@Inject
private CatalogService catalogService;
@Incoming("orders")
@Blocking
public void processOrder(String orderStr) {
try {
System.out.println(x:"received message inventory");
Order order = Transformers.jsonToOrder(orderStr);
order.getItemList().forEach(orderItem -> {
int old_quantity = catalogService.getCatalogItemById(orderItem.getProductI
int new_quantity = old_quantity - orderItem.getQuantity();
if (new_quantity < LOW_THRESHOLD) {
System.out.println("Inventory for item " + orderItem.getProductId() +
} else {
PROBLEMS 8
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
COMMENTS
zsh
[INFO] Finished at: 2025-02-28T07:38:18-05:00
[INFO] --
hitpatel@hitpatel-mac coolstore %
partial-migration*
0↓ 3↑
5 2 1
0
Java: Ready
Ln 2, Col 1
Spaces: 4
UTF-8
LF
Java


coolstore
KONVEYOR
KONVEYOR ISSUES
34 results in 7 files
Replace the `javax.jms` import statement ...
Replace the `javax.ejb` import statement ...
Stateless EJBs can be converted to a CDI ...
In Quarkus you can skip the @Produces an...
The Stateless EJBs can be converted to a ...
Dependency `jakarta.ws.rs:jakarta.ws.rs-a...
Jakarta JAX-RS activation is no longer nec...
Use the Maven Failsafe plugin adding the f...
Leverage a Maven profile to run the Quark...
Enterprise Java Beans (EJBs) are not supp...
The `destinationLookup` property can be ...
JMS `Topic`s should be replaced with Mic...
References to JavaEE/JakartaEE JMS elem...
An application running inside a container c...
The app is trying to access local resource ...
KONVEYOR RESOLUTIONS
Edits in 4 file(s)
pom.xml 3
InventoryNotificationMDB.java src/mai... 9+
OrderServiceMDB.java src/main/java/co... 9+
ShoppingCartOrderProcessor.java src/... 9+
Welcome
Konveyor Analysis View
Resolution Details
Current ↔ Suggested
quarkus-00050>(priority=0(2), depth=0, retries=0).
Resolved task PackageDoesNotExistError<loc=pom.xml:0:0, message=package javax.jms does not exist>(priority=1(0), depth=2, retries=0) indirectly while fixing AnalyzerRuleViolation<loc=/Users/hitpatel/coolstore/src/main/java/com/redhat/coolstore/service/ShoppingCartOrderProcessor.java:0:-1, violation.id=jms-to-reactive-quarkus-00050>(priority=0(2), depth=0, retries=0).
Resolved task AnalyzerRuleViolation<loc=/Users/hitpatel/coolstore/src/main/java/com/redhat/coolstore/service/ShoppingCartOrderProcessor.java:0:-1, violation.id=javax-to-jakarta-import-00001>(priority=2(0), depth=2, retries=0) indirectly while fixing AnalyzerRuleViolation<loc=/Users/hitpatel/coolstore/src/main/java/com/redhat/coolstore/service/ShoppingCartOrderProcessor.java:0:-1, violation.id=jms-to-reactive-quarkus-00050>(priority=0(2), depth=0, retries=0).
Resolved task AnalyzerRuleViolation<loc=/Users/hitpatel/coolstore/src/main/java/com/redhat/coolstore/service/ShoppingCartOrderProcessor.java:0:-1, violation.id=jms-to-reactive-quarkus-00050>(priority=0(2), depth=0, retries=0).
Finished with 0 errors and 7 modified files.
▸ Successful tasks:
▸ Ignored tasks:
Finished!
Received response...
pom.xml → 2 additions, 4 deletions
InventoryNotificationMDB.java → 17 additions, 65 deletions
OrderServiceMDB.java → 21 additions, 44 deletions
ShoppingCartOrderProcessor.java → 9 additions, 22 deletions
PROBLEMS 89
OUTPUT
DEBUG CONSOLE
COMMENTS
TERMINAL
PORTS
zsh
[INFO] ----------[jar]----------
[INFO]
[INFO] --- clean:3.2.0:clean (default-clean) @ coolstore ---
[INFO] Deleting /Users/hitpatel/coolstore/target
[INFO] --
[INFO] BUILD SUCCESS
[INFO] --
[INFO] Total time: 0.882 s
[INFO] Finished at: 2025-03-03T20:27:05-05:00
[INFO] --
hitpatel@hitpatel-mac coolstore %
partial-migration* 0↓ 2↑ 85 3 1 0 Java: Ready

Guided scenario

JavaEE to Quarkus with Konveyor AI

https://github.com/konveyor/kai/blob/main/docs/scenarios/javaEE_to_quarkus/README.md



Intro

Konveyor

Konveyor AI / Kai

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!
- #konveyor Slack channel at kubernetes.slack.com

KAI

KONVEYOR

Thank you!