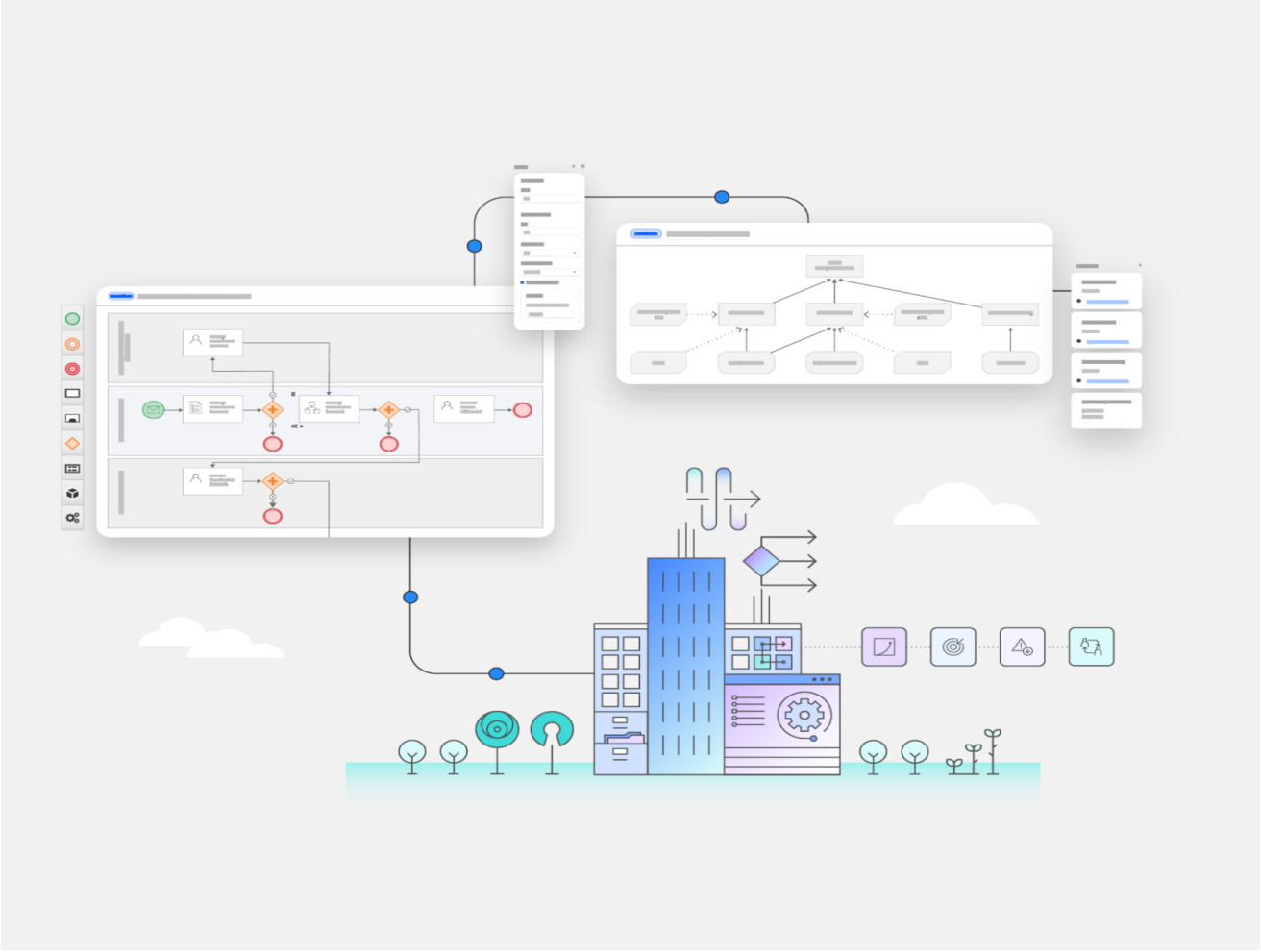
IBM Business Automation Manager Open Editions

Process Automation Lab Guide
W420G - IBM BAMOE Skills Academy



Barry Lulas
barry.lulas@ibm.com
Principal Technical Sales Leader, Business Automation SME, Americas

Jason Porter
jporter@ibm.com
Software Engineer

Document edition: March 2025

Table of Contents

Introduction

Welcome to the **IBM BAMOE Skills Academy** lab guide for **Process Automation**. This lab will cover using **IBM Process Automation Manager Open Edition**'s capabilities for process and decision automation, including guidance on how to build a complete process service as well as deployment and execution on Red Hat OpenShift.

IBM Business Automation Manager Open Editions (IBM BAMOE) is a cloud-native business automation technology for building cloud-ready business applications. It is built from various open-source projects including Drools, jBPM, and most notably Kogito. The letter *"K"* in Kogito refers to Kubernetes, the base for Red Hat OpenShift, as the target cloud platform for IBM BAMOE and the Knowledge is Everything (KIE) open-source business automation project for which IBM BAMOE originates.

IBM BAMOE is optimized for a hybrid cloud environment and adapts to your domain and tooling needs. The core objective of IBM BAMOE is to help you mold a set of business processes and decisions into your own domain-specific cloud-native set of services.

When you use IBM BAMOE, you are building a cloud-native application as a set of independent domain-specific services to achieve some business value. The processes and decisions that you use to describe the target behavior are executed as part of the services that you create. The resulting services are highly distributed and scalable with no centralized orchestration service, and the runtime that your service uses is optimized for what your service needs.

What you will need to access

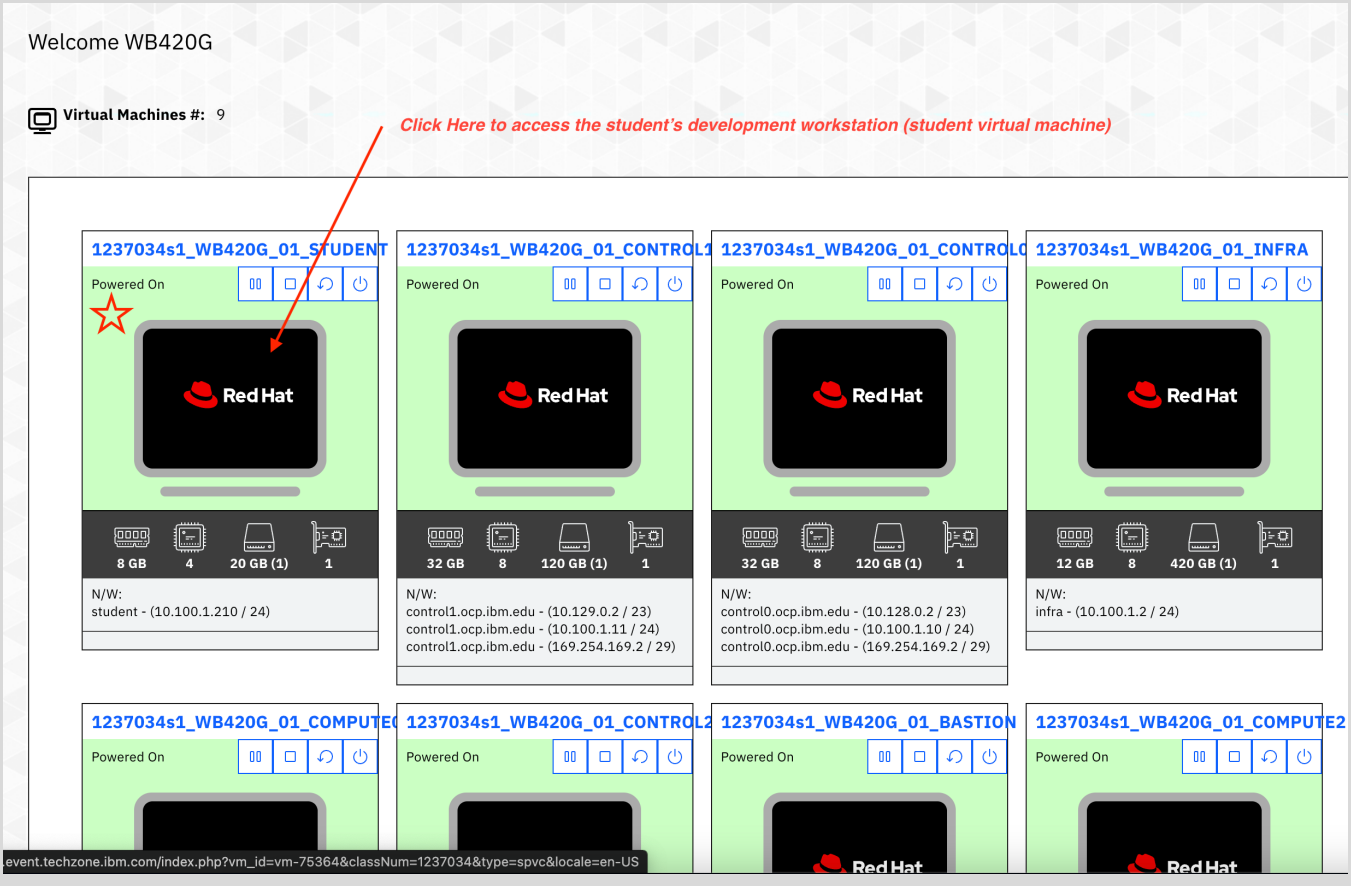
This lab utilizes two environments:

- **IBM BAMOE Developer Workstation** which provides the student with all the necessary tools and technologies for building, testing, executing, and deploying IBM BAMOE applications. This environment is based on Red Hat Linux v9 and represents a common developer workstation for IBM BAMOE. Students will have direct access to the OpenShift cluster from the Student VM, as well as have administrative console access via a browser and command-line interface.

- **Red Hat OpenShift Cluster** which provides infrastructure services in support of IBM BAMOE applications. This environment is based or Red Hat OpenShift and represents a common target execution environment for IBM BAMOE. The student will not necessarily need to login to this environment, as they have access via the OpenShift command-line interface or the web-based OpenShift administrative console.

*We will use the following environments for the Student's Developer Workstation (**please see the environment assignment sheet provided by the instructor for the environment that you are assigned. The assignment sheet is also located in the bamoe-skills-academy-documents folder under the environments folder**). Be sure to select the **WB420_STUDENT** virtual machine from the list of available virtual machines of this environment.*

Important: We will use the following account information to login and use the student's virtual machine. Please see the environment assignment sheet, provided by the instructor. Be sure to select the *WB420G_XX_STUDENT virtual machine, from the list of other virtual machines when you visit the assigned environment URL, as seen in the following example:



Virtual Machine Accounts

Once you have successfully accessed your assigned student virtual machine, the following table depicts the user accounts and passwords used in the lab environments:

Table 1: User IDs and passwords for your lab environment

VM name	Account	Password	Comment
Student Virtual Machine (Developer Workstation)	ibmuser	Passw0rd	BAMOE Developer Workstation (RHEL9)
	Root	1l0veibmrh	Use sudo
Lab OpenShift Cluster Administration	ocadmin	ibmrhocp	default-route-openshift-image-registry.apps.ocp.ibm.edu

Important: From time to time, you may be requested to enter the student's account password in various tools, such as VS Code or Google Chrome. If that happens, simply use the student virtual machine account password, noted above.

High Level Lab Architecture

The following diagram depicts the high-level architecture as it relates to typical BAMOE application life-cycle management. Students, playing the role of **developer** or **modeler**, will utilize the **Student Virtual Machine**, which provides a typical example of how a developer machine would be configured to develop BAMOE applications. Applications built during the labs will be automatically deployed to the **Lab OpenShift Cluster**, which provides a typical example of how an OpenShift or Kubernetes cluster would be configured to deploy, test, and execute BAMOE applications. Positioned in the middle of the developer workstation and the OpenShift cluster, is the enterprise Git repository management system, for which all BAMOE applications are published and utilized.

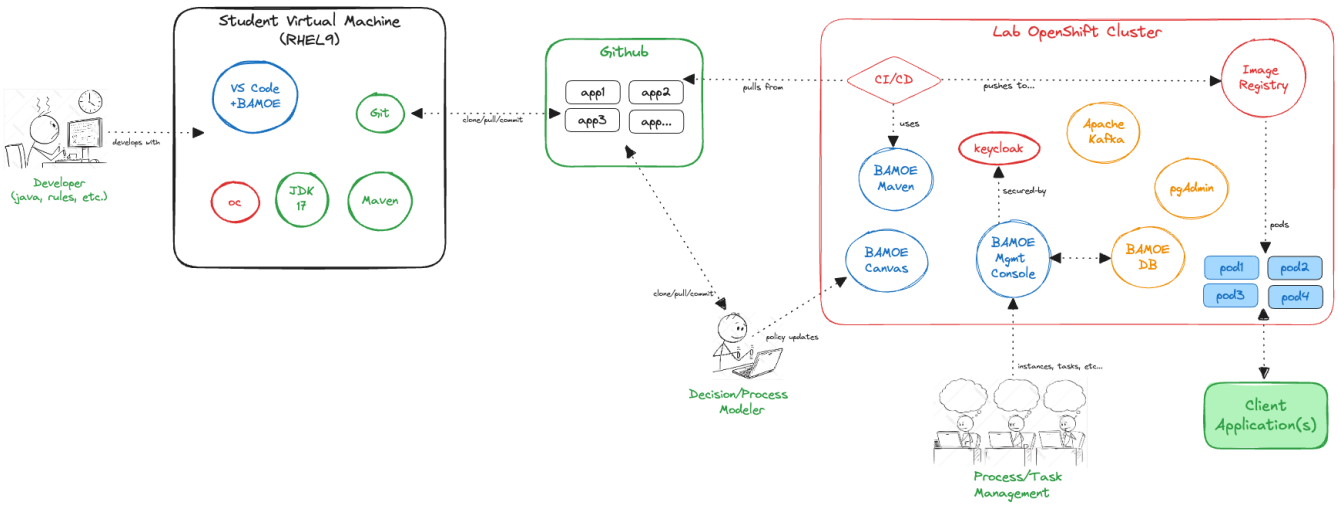
The developer workstation is installed with typical Java-based development tools, such as the VS Code IDE, with the BAMOE Developer Tools extensions already installed. In addition, supporting tools and technology, such as the Java Development Kit v17, Git command-line interface, Maven command-line interface, and the OpenShift command-line interface all provide the developer with the necessary tools to build, test and deploy BAMOE applications.

The OpenShift cluster is configured with all necessary infrastructure services that BAMOE needs. In the case of Process Automation projects, those services include the BAMOE Maven Repository, BAMOE Canvas, and the BAMOE Management Console, as well as supporting infrastructure services such as a PostgreSQL database (*for process persistence*), pgAdmin (*for database administration*), Apache Kafka (*for event-based applications*), and Keycloak (*for securing REST endpoints*). All BAMOE applications are deployed to the OpenShift cluster as container images and automatically create all OpenShift objects, such as deployments, pods, services, and routes. Images are pushed to the OpenShift container image registry, from either the developer workstation or a CI/CD pipeline, using standard Maven commands.

All BAMOE projects are standard Maven projects and are stored in an accessible Git repository. Tools such as the BAMOE Developer Tools or BAMOE Canvas access these Maven projects directly from the tool, simply by using standard Git commands.

The following diagram represents the typical architecture for process automation applications, using IBM BAMOE:

Figure 1: Lab architecture for Process Automation applications



The process automation labs described in this document focus on how to create BAMOE projects, add business automation resources, including technical rules written in Drools Rule Language (DRL), decision models written in Decision Modeling & Notation (DMN), and orchestrated with stateless or stateful workflow written in Business Process Modeling & Notation (BPMN). You will learn how to configure the features of each process service through adding various Maven dependencies and updating the service's property setting for each target deployment profile. You will also learn how to deploy and execute process services as standalone applications or as container images on Docker and Kubernetes. Finally, you will learn how to properly test your process services.

Lab Environment Setup and Basic Usage

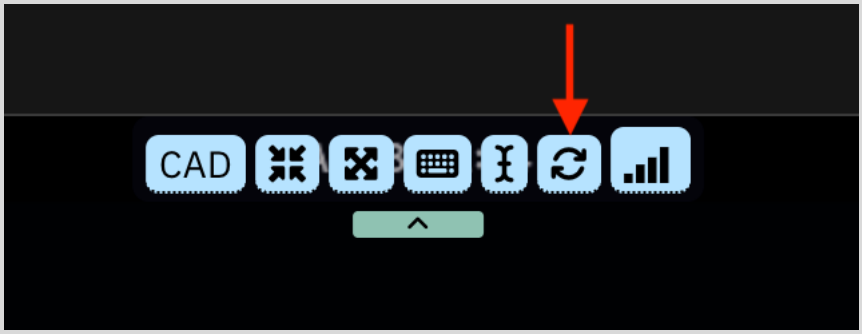
The lab environment representing the Developer Workstation (*student virtual machine*) has been pre-configured with all student lab guides and lab project repositories. However, there may be situations where you need to refresh these artifacts, as instructors may be required to make minor updates during the delivery of the course, and the instructor will show you how to do that in the first class session.

Access the Developer Workstation (Student Virtual Machine)

The Developer Workstation is a virtual machine configured as a Red Hat Linux v9 workstation and made available to students through their own laptop's browser. Once you login, you will see the following:

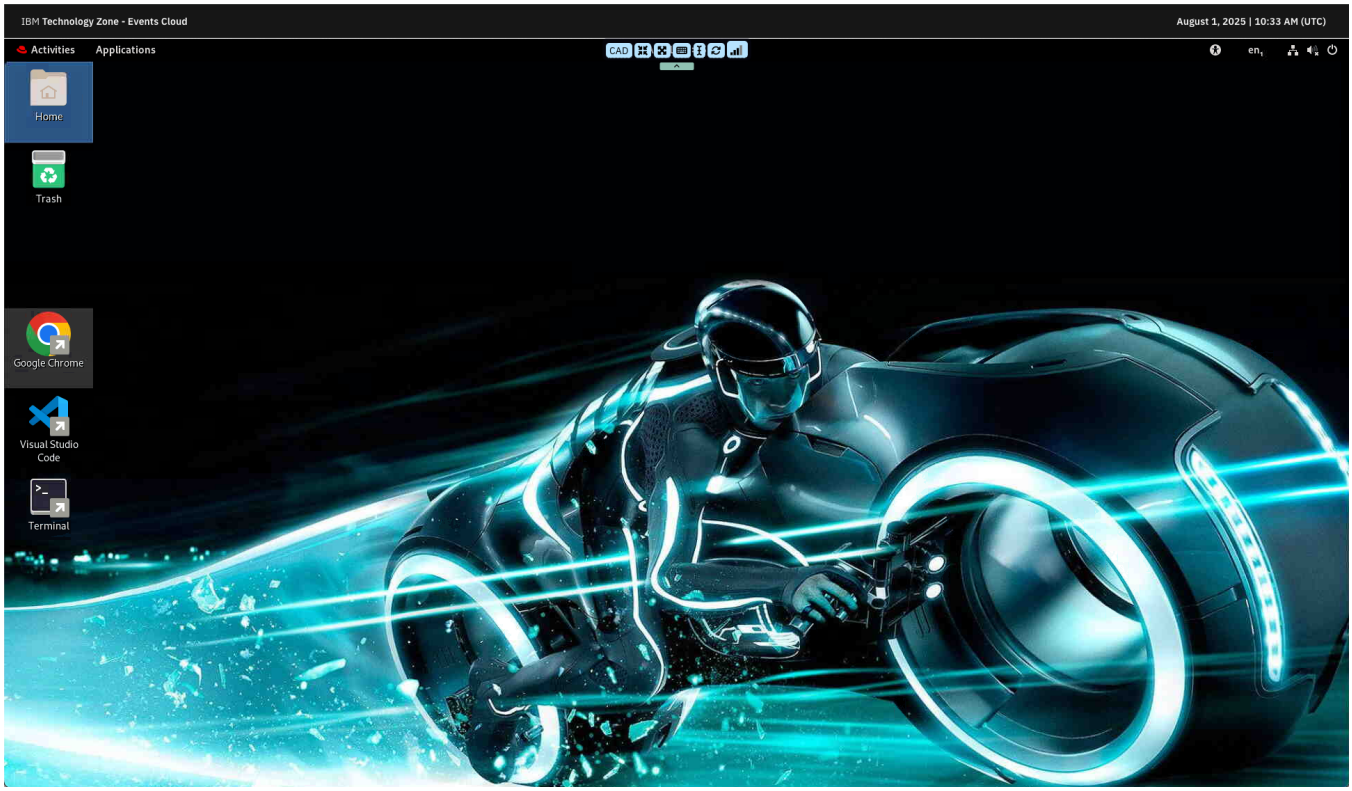
Important: While any browser will work for accessing the student virtual machine from the student's personal laptop, is it recommended to use Google Chrome. The student virtual machine comes pre-configured with Google Chrome, which is one of the recommended browsers for IBM BAMOE Canvas and IBM BAMOE Management Console.

It is also important to note that the performance of the browser-based access to the virtual machines can at sometimes become a little slow. You may need to refresh the desktop occasionally, especially if you get into situations where mouse clicks and typing is slow. At the top of the virtual machine window is a toolbar, show below. **Click the icon directly below the red arrow**, show below, to refresh the virtual machine:



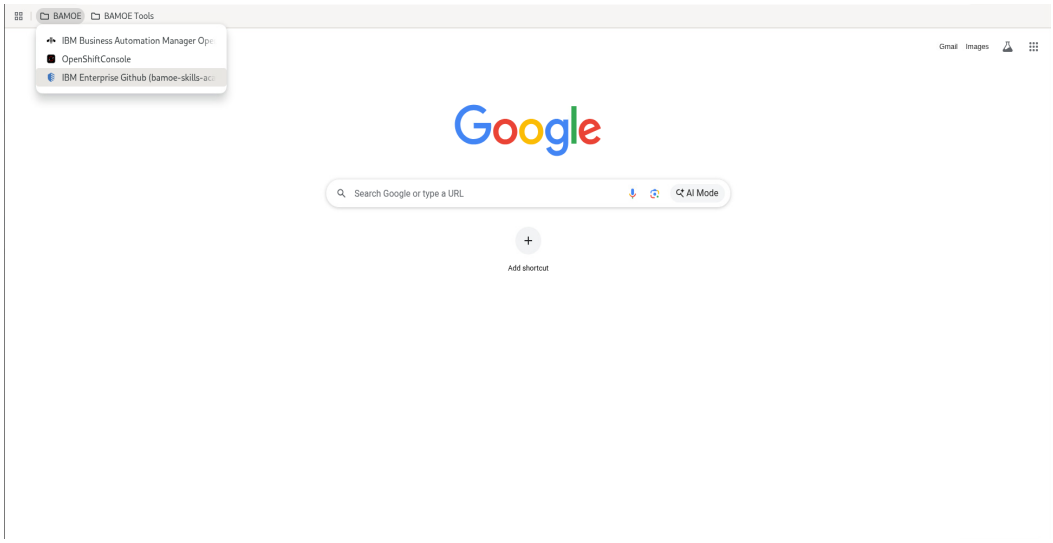
From here, your virtual machine desktop should look like the following:

Figure 3: Developer Workstation (student virtual machine) Desktop

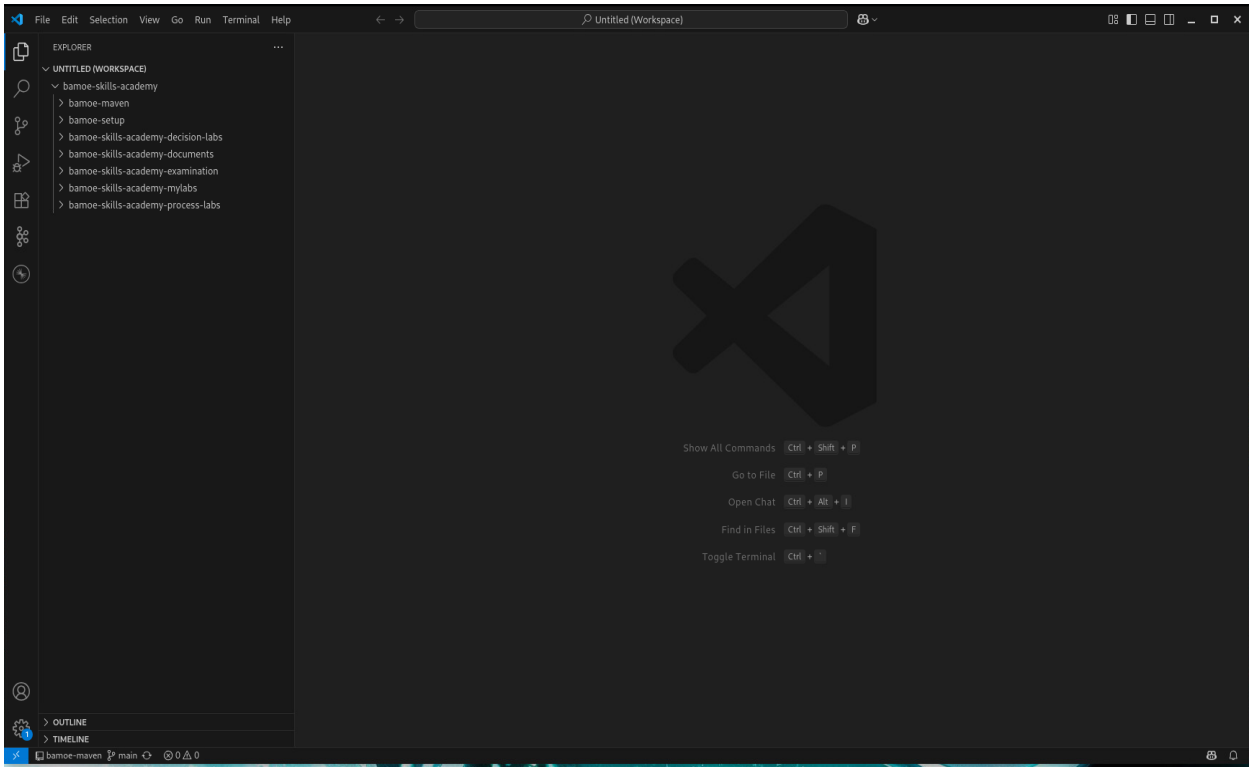


There are three main desktop shortcuts available to the student:

- **Google Chrome**, which is used to access web-based tools such as IBM BAMOE Canvas, IBM BAMOE Management console, pgAdmin (database) console, and the OpenShift administration console. Please see the provided Google Chrome bookmarks for access to these applications.



- **Microsoft Visual Studio Code (VS Code)**, which is the IDE used for IBM BAMOE project development. The IBM BAMOE extensions for BPMN, DMN, and Test Scenarios are already installed into VS Code, as well as extensions for Kafka and testing REST API endpoints. In addition, all lab repositories are already added to the VS Code workspace.



- **Terminal Window**, which is used for issuing Maven and Git commands on IBM BAMOE projects. You can also use the Terminal Window capabilities of VS Code vs using the desktop terminal window.

GUIDED DEMO: Creating and Configuring Process Service Projects

In this lab you will:

- Familiarize yourself with where items are in the BAMOE VS Code IDE extension
- Create a simple process that logs to the console
- Execute that process through the BAMOE Dev UI

Create a new project folder and file

Create a new folder with in the “bamoe-skills-academy-mylabs” folder called “hello-task.” Open this folder within VS Code.

Create a new file named “hello-process.bpmn”. This will load up the BPMN editor within VS Code.

Create the process

- Drag onto the canvas a starting node
- Link that starting node to a new task labeled “Hello”
- Convert that task to a Script task
- Add the following code to the Script section within Properties:

```
System.out.println("Hello Task”);
```

- Link the “Hello” task to an end node

Apply the accelerator

Select the “hello-task” folder in the explorer within VS Code. Use the hotkey `CTRL + SHIFT + P` to bring up the command pallet. Select the “BAMOE Tools: Apply Accelerator” action. Then select “Quarkus Full”. This will create the necessary files and folder structure for a Business Process project.

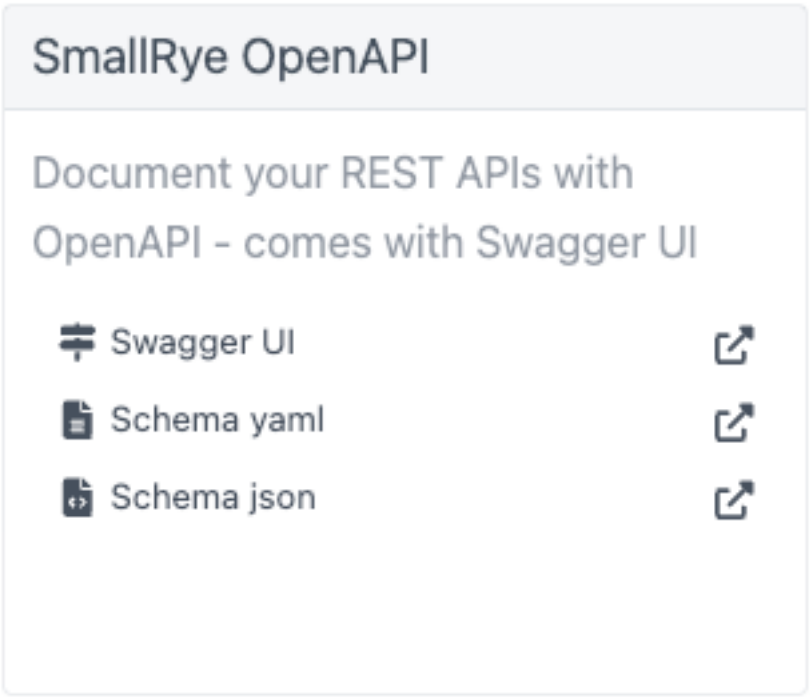
Run the project using the terminal

Open a new terminal within VS Code and execute the command:

```
mvn clean quarkus:dev
```

This will build the project and start the Quarkus development environment.

- Access the Dev UI by opening Chrome and navigating to `localhost:8080/q/dev`
- Scroll down to “Smallrye OpenAPI” and click “Swagger UI”



- Click on the POST /hello_process
- Click “Try it out”
- Click “Execute”

Note: We do not need anything more than an empty JSON object in the POST request

- Notice in the terminal window within VS Code the process started, nodes were triggered and the output from the script is also printed to the terminal.

Summary

In this lab, you have

- Created a simple process
- Added the accelerator to the project
- Run the project in development
- Executed the process using the Quarkus Dev UI

LAB: Model Your Morning

In this lab you will model out your morning routine. The specifics of the process are up to you, however, there is a list of minimum requirements:

1. The process must include a start node
2. The process must include an end node
3. The process must have and at least three different tasks two of which should be different types of tasks.

It is recommended this lab is done in a new folder with the accelerator applied, however, obtaining the proper structure for the project is left up to you.

Below is an image of a simple workflow which satisfies the requirements.

Summary

In this lab you learned how to

- Create a Business Process Service
- Build the Business Service
- Start a workflow through REST
- Interact with various task types
- Utilize Quarkus Dev UI to explore the various endpoints of a service

LAB: Software Release

In this lab you will model a typical software development release cycle.

Your solution should include steps for requirements gathering, coding, testing, documentation, building and releasing.

Your solution must:

- Include Process Variables
- Gateways
    - At least one parallel gateway
    - At least one exclusive gateway
- At least one sub-process
- At least one user process were a user adds data to the system
- A test to validate the process works correctly

Summary

In this lab you have learned:

- How to include variables and other process data into your process
- Make decisions using gateways
- Execute tasks in parallel
- Create User Tasks and forms where a human can interact with your process
- Create and execute sub-processes
- Build tests to verify your process

LAB: Final

This final lab builds upon the last lab by

- Adding security to the user task endpoints
- Utilizing events and Kafka
- Creating and scheduling timers
- Handling errors within your process

The following users are available within Keycloak without changing any configuration:

- admin
    - Roles: admin, managers, user, IT, HR
- alice
    - Roles: user, HR
- jdoe
    - Roles: managers, user, IT

You may add more users and roles if desired, but that is beyond the scope of this course.

The Keycloak instance is accessible within the OpenShift cluster at:

- keycloak.bamoe

The Kafka cluster is available within the OpenShift cluster at:

- bamoe-kafka-cluster-kafka-bootstrap.openshift-operators:9092

Summary

In this lab, you have learned how to:

- Secure endpoints using Keycloak
- Integrate Keycloak users and realm using OpenID Connect (OIDC)
- Leverage Kafka as an event source within your processes
- Create and use timers within a process
- Handle errors that may occur during process execution