VERSION 1.0

SEPTEMBER 29, 2024

BROKERAPP

OPERATION DEPLOYMENT DOCUMENT

CONTENTS

1. OVERVIEW

Project Name:	Brokerage Firm Backend API
Application/Service:	Backend API for managing stock orders, deposits, and withdrawals.
Version:	1.0.0
Deployment Date:	29.09.2024
Prepared by:	CAN GUNDUZ
Reviewed by:	CAN GUNDUZ

Project Source Code: https://github.com/cangdev/exercise/tree/master/brokerapp

2. Deployment Objective

This document outlines the steps required to deploy the Brokerage Firm Backend API that allows employees to manage stock orders and customer transactions such as deposits and withdrawals. The application is built using Spring Boot and is compatible with JDK 17.

3. Deployment Components

Application JAR	: brokerapp-0.0.1-SNAPSHOT.jar
Startup Script	: start-brokerapp.sh
Shutdown Script	: stop-brokerapp.sh
Database	: H2 db
Configuration Files	: application.properties located in the “/brokerapp/resources” folder
Logs Directory	: “/brokerapp/logs”

4. Prerequisites

4.1. Hardware Requirements

Minimum CPU: 2 cores

RAM: 4 GB

Disk Space: 20 GB (for application and logs)

4.2. Software Requirements

Operating System: Linux (preferred)

JDK Version: JDK 17 (Ensure that JDK 17 is installed and set as default)

4.3. Access Requirements

SSH access to the production server.

Permissions to deploy artifacts and start/stop the application.

Database credentials aren’t required.

5. Deployment Process

5.1. Pre-Deployment Steps

1. Ensure Backup:
   - Confirm that the database and relevant files are backed up.
     `mv brokerapp-0.0.1-SNAPSHOT.jar brokerapp-0.0.1-SNAPSHOT@240929`
   - Ensure there's enough free disk space (minimum 5 GB) for the deployment.

2. Check JDK Version:
   - Verify JDK 17 is installed by running:

     `java -version`
   - If not installed, follow the appropriate steps to install JDK 17.

3. Stop Running Application:
   - Execute the stop script to stop the current application:
     `./stop-brokerapp.sh`

5.2. Deployment Steps

5.2.1. Application Deployment Steps

1. Copy the Application JAR:
   - Upload the latest version of the JAR file (brokerapp.jar) to the target directory:
     /appdata/brokerapp/

2. Verify Configuration Files:
   - Ensure that `application.properties` is correctly set up in the `/appdata/brokerapp/resources/` folder.
   - Update environment-specific configurations such as database URL, credentials, and log file paths.

3. Start the Application:
   - Start the application using the provided startup script
     `./start-brokerapp.sh`

4. Verify Logs:
   - Ensure the application has started successfully by checking the logs:
     `tail -100f brokerapp.log`

5. Network Check
    - Verify that the application is accessible (*please replace localhost with ip*): `http://localhost:9090/actuator/health`

5.2.2. DB Deployment Steps

There is no db deployment step.

5.3. Post-Deployment Steps

1. Test Application:

    - Ensure that key functionalities are working, such as:
        - Sending, listing, and deleting asset orders.
        - Deposit and withdrawal transactions.

2. Monitor Application:

    - Verify the database and application logs for any unusual activity or errors.

6. Rollback Plan Process

If the deployment fails or critical issues arise, follow the rollback plan below:

1. Stop the New Application:
   - Use the stop script to terminate the running instance of the new version:
     `stop-brokerapp.sh`

2. Restore Previous Version:
   - Restore the previous version of the JAR file from the backup or artifact repository.
   - Move the old JAR back to the deployment folder:
     `mv brokerapp-0.0.1-SNAPSHOT@240929 brokerapp-0.0.1-SNAPSHOT.jar`

4. Start the Previous Version:
   - Run the startup script for the previous version:
     `start-brokerapp.sh`

6. Verify Rollback:
   - Ensure the application is functioning as expected by checking logs and testing endpoints.

7. Network Check:
   - Verify that the application is accessible (*please replace localhost with ip*):
     `http://localhost:9090/actuator/health`