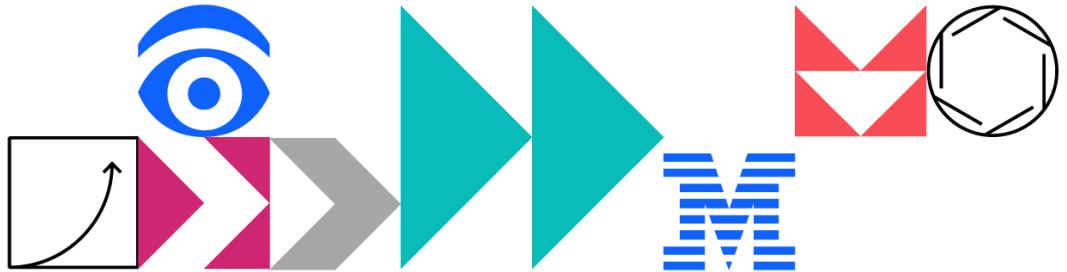




IBM TechXchange



Amplify your Integration Middleware with DevOps: The IBM DevOps Deploy Way

Session 1440

Lab Exercise Guide

Rafael de Oliveira Osorio

Learning Content Development, Automation Software

rosorio@br.ibm.com

Randall Langehennig

Solution Architect, Product Manager

rlange@ibm.com



IBM TechXchange

Notices and disclaimers

© 2024 International Business Machines Corporation. No part of this document may be reproduced or transmitted in any form without written permission from IBM.

U.S. Government Users Restricted Rights — use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.

This document is current as of the initial date of publication and may be changed by IBM at any time. Not all offerings are available in every country in which IBM operates.

Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information.

This document is distributed “as is” without any warranty, either express or implied. In no event, shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity. IBM products and services are warranted per the terms and conditions of the agreements under which they are provided. The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

IBM products are manufactured from new parts or new and used parts.

In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply.”

Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.

Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

It is the customer’s responsibility to ensure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer’s business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer follows any law.

Notices and disclaimers (Continued)

Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. **IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a purpose.**

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

IBM, the IBM logo, and ibm.com are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml.

Table of Contents

1 INTRODUCTION	5
1.1 LAB ENVIRONMENT ARCHITECTURE	5
1.2 WHAT IS APP CONNECT ENTERPRISE (ACE)	6
1.3 WHAT IS IBM DEVOPS DEPLOY	6
1.4 WHAT IS IBM DEVOPS TEST HUB	7
2 GETTING STARTED	9
2.1 HOW TO CONNECT TO THE LAB	9
3 LAB EXERCISES	11
3.1 DEVELOPING AN IBM ACE BROKER ARCHIVE USING THE IBM APP CONNECT ENTERPRISE TOOLKIT.....	11
3.1.1 OVERVIEW	11
3.1.2 WORKBOOK STEPS	11
3.2 SETUP A JENKINS CI PIPELINE TO AUTOMATE BUILD	17
3.2.1 OVERVIEW	17
3.2.2 WORKBOOK STEPS.....	18
3.3 REVIEW THE EMERALD ESTORE APPLICATION IN IBM DEVOPS DEPLOY	30
3.3.1 OVERVIEW	30
3.3.2 WORKBOOK STEPS.....	31
3.4 WORK WITH THE EMERALD-ACE-BAR COMPONENT IN DEPLOY.....	32
3.4.1 OVERVIEW	32
3.4.2 WORKBOOK STEPS	32
3.5 DEPLOY THE ACE BROKER ARCHIVE USING IBM DEVOPS DEPLOY TO DEV.....	44
3.5.1 OVERVIEW	44
3.5.2 WORKBOOK STEPS	44
3.6 PROMOTE THE EMERALD COMPONENTS TO THE QA ENVIRONMENT	51
3.6.1 OVERVIEW	51
3.6.2 WORKBOOK STEPS	51
3.7 PUBLISHING A PRODUCT TO API CONNECT USING IBM DEVOPS DEPLOY	61
3.7.1 OVERVIEW	61
3.7.2 WORKBOOK STEPS	62
4 WRAPPING THINGS UP	74
4.1 LAB RE-CAP.....	74



IBM TechXchange

1 Introduction

The purpose of the IBM DevOps Hands On workshop is to provide a high-level overview on how to achieve end-to-end automation in your DevOps pipelines to ensure good quality code is progressing from your lower environments all the way to your production environment. The focus will be to highlight App Connect Enterprise (ACE) and API Connect (APIC).

In this lab, you will be working with these two primary IBM DevOps solutions:

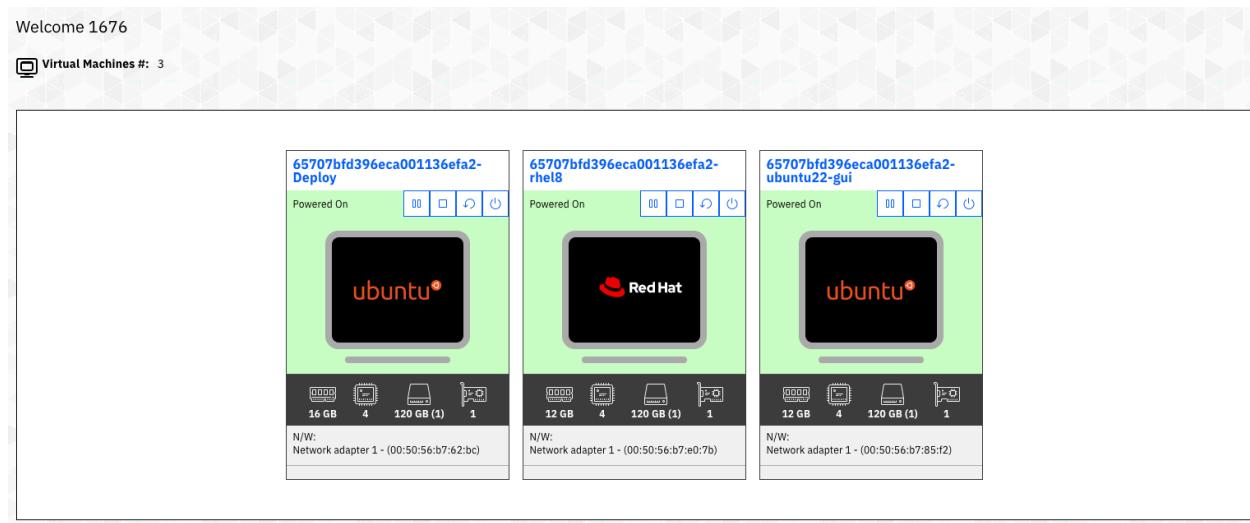
- IBM DevOps Deploy (formerly known as IBM UrbanCode Deploy)
- IBM DevOps Test Hub

Working with a container-based application called Emerald eStore, you will gain an understanding of IBM DevOps Deploy and how to configure and deploy the Emerald eStore application which includes an ACE broker archive and an API Connect Product you will publish to a catalog. You will also learn how to configure automated test execution using IBM DevOps Test Hub in your automated processes.

This lab will provide an example of how you can leverage gates in IBM DevOps Deploy for governance, call automated tests (could include integration, functional, and performance tests) following an automated deployment to track quality, and initiate releases to your target environments.

1.1 Lab Environment Architecture

The high-level architecture for the environment includes three different servers as shown below:



The main system we will utilize for our labs is the RedHat (rhel8) host found in the middle of the screen. You will connect to this host to follow the steps in the labs. One server image hosts the IBM DevOps Deploy Server and Agent. Finally, the last server image hosts the IBM DevOps Test Hub installation.



IBM TechXchange

1.2 What is App Connect Enterprise (ACE)

IBM App Connect Enterprise combines the existing industry-trusted technologies of IBM Integration Bus with cloud-native technologies to deliver a platform that supports the full breadth of integration needs across a modern digital enterprise.

You can use IBM App Connect Enterprise to connect applications together, regardless of the message formats or protocols that they support. This connectivity means that your diverse applications can interact and exchange data with other applications in a flexible, dynamic, and extensible infrastructure. IBM App Connect routes, transforms, and enriches messages from one location to any other location.

By using the IBM App Connect Enterprise Toolkit, you can develop integration solutions and deploy them to IBM Cloud Pak for Integration, the dedicated runtime of IBM App Connect Enterprise software, and to IBM App Connect Enterprise as a Service. The labs that follow will make use of this toolkit to complete our tasks.

1.3 What is IBM DevOps Deploy

IBM DevOps Deploy is an application-release solution that infuses automation into the continuous delivery and continuous deployment (CI/CD) process and provides robust visibility, traceability and auditing capabilities.

IBM DevOps Deploy provides the following capabilities:

- **Integrations that replace custom scripting**

Integrations with technologies such as Jenkins, Kubernetes, Microsoft, ServiceNow and IBM® WebSphere® make processes more robust and easier to design.

- **Scalable distributed automation**

Meet enterprise requirements with an architecture designed to scale — providing high availability, horizontal scalability and tight security.

- **Visual deployment modeling**

Model cloud environments in a simple graphical editor to deploy your applications to public, private and hybrid clouds.

- **Inventory control**

Track what resides where with visibility into your applications, environments and configurations.

- **Quality gates and approvals**



IBM TechXchange

Ensure that only application-component versions that meet quality criteria are promoted across test environments en route to production.

- **Coordination of multicontainer deployments**

Employ industry-leading release automation capabilities for the container world.

- **Rapid migration to the cloud**

UrbanCode Deploy is available as a container and certified to work with Red Hat OpenShift.

1.4 What is IBM DevOps Test Hub

The IBM DevOps Test Hub application (formerly known as IBM Rational Test Automation Server) brings together test data, test environments, and test runs and reports into a single, web-based browser for testers and non-testers.

The solution provides the following capabilities:

- **Web-based continuous testing platform**

IBM DevOps Test Hub is a web-based continuous testing platform built on modern, cloud native technologies that enables test teams to run a breadth of tests that includes API, functional, and performance tests.

- **Role-based access and security**

Security is a key concern for IBM clients and therefore, DevOps Test Hub brings a comprehensive, role-based access control scheme to the server with project owners assigning key permissions (by using roles) for specific members. For example, managing test data or working with secrets such as passwords.

- **Running of tests from the server by using Docker containers**

The DevOps Test Hub solution enables direct running of tests from the browser by using transient Docker containers.

- Connected agents for existing performance agents.

Agent owners can connect existing performance agents to the server and add them to a project for running schedules and Accelerated Functional Testing (AFT) Suites on the current infrastructure.

- **Project overview statistics**



IBM TechXchange

The Overview page for IBM DevOps Test Hub offers you a quick, simple view on the state of testing for your projects.

- **Test data authoring**

Beyond the concept of a project held in a Git repository for a simple location of tests and related assets, you can do full concurrent editing of test data sets directly from DevOps Test Hub. This true multi-user capability enables team members to collaborate more easily as well as try out data changes without impacting the rest of the team. When satisfied with the results, team members can push their changes.

- **Integration with DevOps tools**

You can integration DevOps Test Hub with various popular DevOps tools like Jenkins, DevOps Deploy, or Microsoft Azure DevOps to get more value from your DevOps pipelines. You can configure an IBM DevOps Deploy process that automatically triggers test cases and have those test insights set a status to indicate that the tests passed or failed.

DevOps Test Hub also integrates with Jira for defect tracking and GitHub for software source management and version control.



IBM TechXchange

2 Getting Started

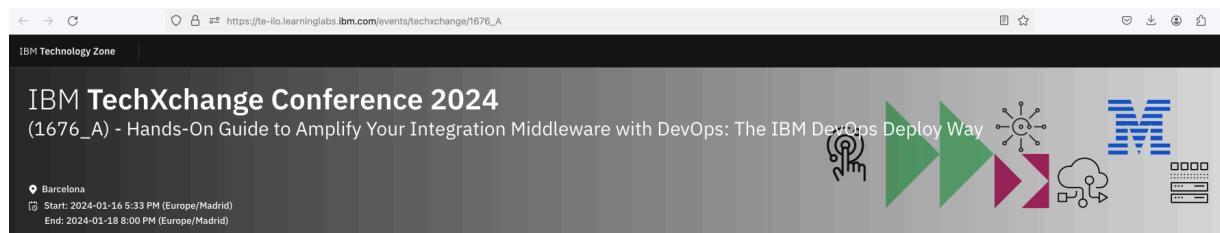
2.1 How to Connect to the Lab

Follow these instructions to connect to the lab environment.

1. On your laptop, start your Firefox browser.
2. Click on the link to your environment that has been assigned to you. For example:

https://te-ilo.learninglabs.ibm.com/events/techxchange/1676_A#

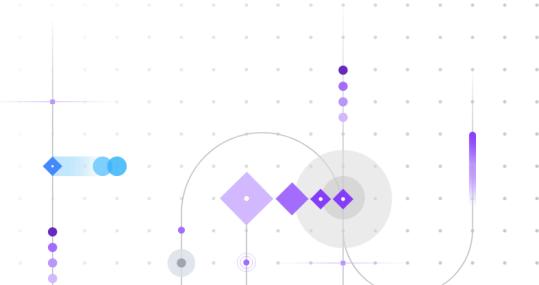
You will see a screen that looks like this:



Log in

E-mail:
be sure to spell/type correctly

Login



3. Enter your email address and click the Login button (as shown above).

You should see the screen below:

IBM TechXchange



IBM TechXchange Conference 2024

(1676_A) - Hands-On Guide to Amplify Your Integration Middleware with DevOps: The IBM DevOps Deploy Way

Barcelona
Start: 2024-01-16 5:30 PM (Europe/Madrid)
End: 2024-01-18 8:00 PM (Europe/Madrid)



Hello rrange@ibm.com

Logout

Hands-On Guide to Amplify Your Integration Middleware with DevOps: The IBM DevOps Deploy Way

1676_A

Lab Guide



Student Environments

Student environments assigned to users. (Please refresh this page to see the latest information.)

Q Search Student Assigned

Refresh

Ident

Student

Assigned

Release

S001

Student 1

Release 1



S002

Student 2

Release 2



Items per page:

25

1 - 2 of 2 items

1

- Click on the “Launch Lab” button. You will see the following screen:

The screenshot shows three student environments listed horizontally. Each environment has a small thumbnail, resource details (RAM, CPU, Storage), and a status bar at the bottom. The 'Red Hat' environment is highlighted with an orange border around its thumbnail.

Environment	RAM	CPU	Storage	Status
ubuntu22-gui	16 GB	4	120 GB (1)	1
Red Hat	12 GB	4	120 GB (1)	1
ubuntu	12 GB	4	120 GB (1)	1

N/W: Network adapter 1 - (00:50:56:b7:62:bc) N/W: Network adapter 1 - (00:50:56:b7:a0:7b) N/W: Network adapter 1 - (00:50:56:b7:85:f2)

- Click on the RedHat or ‘rhel8’ screen as shown above. This will launch the console we will be using for the labs.

The screenshot shows a Red Hat Linux desktop environment. A terminal window is open, displaying a login prompt for the 'sysadmin' user. The terminal window has a red 'Red Hat' logo at the bottom right. The desktop background is black.

- Click on the screen and you will see a login for the ‘sysadmin’ user. Provide a password of “**Passw0rd**” and press Enter.

- You are good to proceed to the first lab!



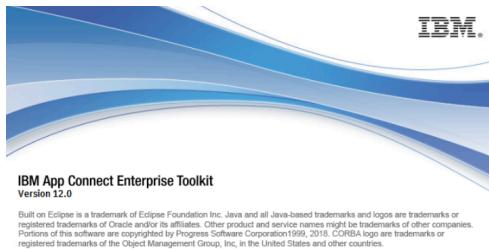
3 Lab Exercises

3.1 Developing an IBM ACE broker archive using the IBM App Connect Enterprise Toolkit

3.1.1 Overview

In this lab we will explore our IBM App Connect Enterprise (ACE) development capabilities. We will first explore the use of the IDE Toolkit to create and test a broker archive (BAR).

This toolkit is called the “IBM App Connect Enterprise Toolkit” as shown below:



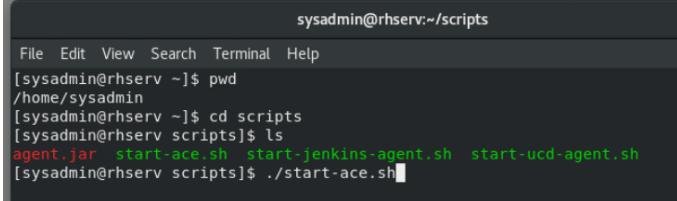
Afterwards, we will look at an example pipeline that can be used to automate the build, unit test, and ultimately the deploy of the broker archive to your target environments.

3.1.2 Workbook steps

Step	Details	Additional Information
1	Click on the Red Hat image found in the middle of the virtual machines to open this image.	
2	Login as the user sysadmin with password of Passw0rd The 0 is a zero.	

IBM TechXchange



3	<p>Open a Terminal Window by clicking the terminal shortcut on the desktop</p>		
4	<p>Before we get started, we need to start our ACE environment by running a script.</p> <p>In your terminal window, run these commands:</p> <pre>\$ cd scripts \$./start-ace.sh</pre> <p>NOTE: let the script run for a couple of minutes so you see that it starts INODEDEV, INODEQA, and INODEPROD before moving to next step.</p>		
5	<p>We will now start the IBM App Connect Enterprise (ACE) Toolkit eclipse-based IDE. It is a great tool for developers as they develop APIs.</p> <p>To start the ACE Toolkit, we need to first source the ACE mqsiprofile by running this command:</p> <pre>\$. /opt/IBM/ace-12.0.10.0/server/bin/mqsiprofile</pre>	<pre>. /opt/IBM/ace-12.0.10.0/server/bin/mqsiprofile</pre> <p>NOTE: there is a space after the . shown above.</p> 	
5	<p>We can now start the IBM App Connect Enterprise Toolkit eclipse-based IDE.</p> <p>Run the commands below to start the toolkit:</p> <pre>\$ cd /opt/IBM/ace-12.0.10.0 \$./ace toolkit</pre>	<pre># cd /opt/IBM/ace-12.0.10.0 # ./ace toolkit</pre> <p>You should see output as follows:</p>	

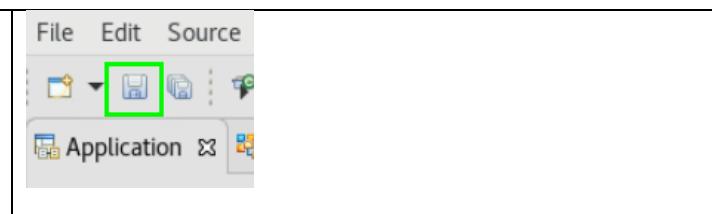
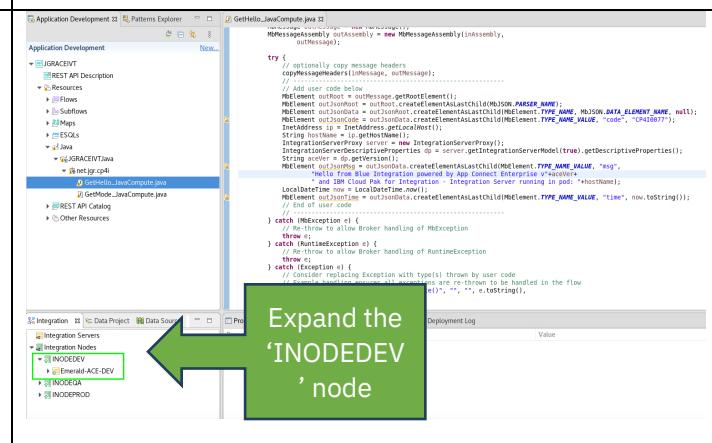
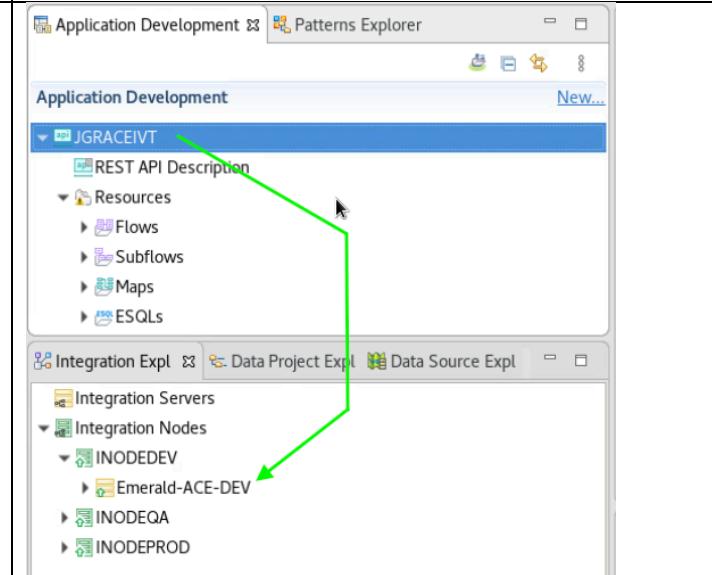
IBM TechXchange



		<pre>sysadmin@rhser:~/opt/IBM/ace-12.0.10.0 File Edit View Search Terminal Help [sysadmin@rhser ~]\$. /opt/IBM/ace-12.0.10.0/server/bin/mqsiprofile MQSI 12.0.10.0 /opt/IBM/ace-12.0.10.0/server [sysadmin@rhser ~]\$ cd /opt/IBM/ace-12.0.10.0/ [sysadmin@rhser ace-12.0.10.0]\$ ls ace common extensions ie02 license readme server swidtag tools [sysadmin@rhser ace-12.0.10.0]\$./ace toolkit Starting App Connect Enterprise Toolkit interactively [sysadmin@rhser ace-12.0.10.0]\$</pre>
6	<p>A launcher dialog will appear.</p> <p>Take the default workspace that is shown and click the “Launch” button.</p>	
7	<p>The IDE will appear as shown to the right and your workspace is already populated with an example ACE project we will use.</p>	
8	<p>As a developer, you can edit your ACE code in the IDE, compile your code, and deploy to a DEV environment.</p> <p>Let's make a small change to our program to see how this works in the IDE.</p>	
9	<p>In the GetHello_JavaCompute.java program, scroll down to the line that starts with “MbElement outJsonMsg” as highlighted to the right.</p>	
10	<p>Change the text that says “Hello from Green Integration powered by...” to “Hello from Blue Integration powered by...” as shown to the right</p>	<pre>String acever = dp.getVersion(); MbElement outJsonMsg = outJsonData.createElementAsLastChild(mbElement.T "Hello from Blue Integration powered by App Connect Enterprise " and IBM Cloud Pak for Integration - Integration Server runnin LocalDateTime.now = LocalDateTime.now(); MbElement outJsonTime = outJsonData.createElementAsLastChild(mbElement.T // End of user code // ----- } catch (MbException e) { // Re-throw to allow Broker handling of MbException</pre> <pre>String acever = dp.getVersion(); MbElement outJsonMsg = outJsonData.createElementAsLastChild(mbElement.T "Hello from Blue Integration powered by App Connect Enterprise " and IBM Cloud Pak for Integration - Integration Server runnin LocalDateTime.now = LocalDateTime.now();</pre>

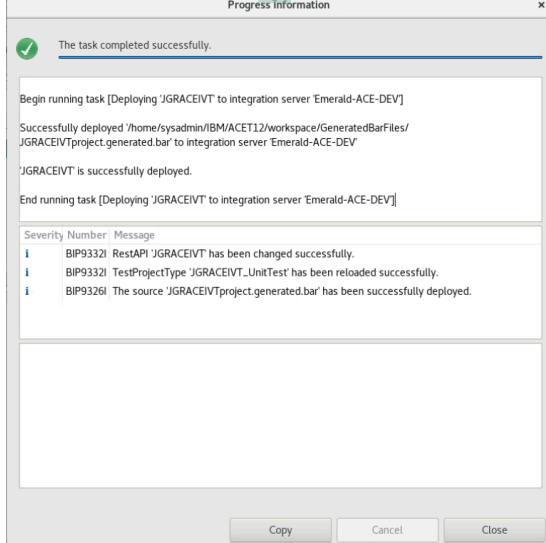
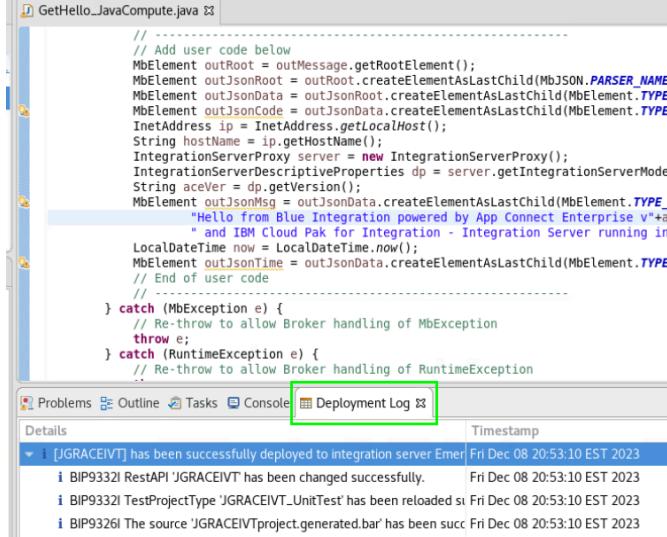
IBM TechXchange



11	<p>Save your changes by clicking on the File menu and then clicking “Save” or by clicking the Save icon as shown to the right.</p>	
12	<p>You can build and deploy your BAR file in a single step using the IDE. Expand the INODEDEV integration node that is shown towards the bottom of the page. You should see the child node, “Emerald-ACE-DEV”, which is an integration server as shown to the right.</p>	 <p>Expand the ‘INODEDEV’ node</p>
13	<p>To initiate the build and deploy, in the left-hand pane, drag “JGRACEIVT” from the top pane to the “Emerald-ACE-DEV” integration server as shown in the screenshot above.</p>	

IBM TechXchange

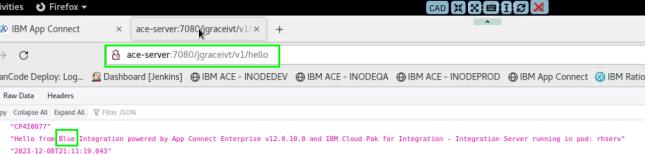


14	<p>A “Progress Information” dialog will appear to show you the progress of your build and deployment.</p> <p>The task should complete successfully as shown to the right.</p> <p>In the messages that appear, you should see that your BAR file has been successfully deployed to the integration server, Emerald-ACE-DEV.</p>	 <p>The task completed successfully.</p> <p>Begin running task [Deploying 'JGRACEIVT' to integration server 'Emerald-ACE-DEV']</p> <p>Successfully deployed '/home/sysadmin/IBM/ACET12/workspace/GeneratedBarFiles/JGRACEIVT/project.generated.bar' to integration server 'Emerald-ACE-DEV'</p> <p>JGRACEIVT is successfully deployed.</p> <p>End running task [Deploying 'JGRACEIVT' to integration server 'Emerald-ACE-DEV']</p> <p>Severity Number Message</p> <ul style="list-style-type: none"> i BIP9332I RestAPI JGRACEIVT has been changed successfully. i BIP9332I TestProjectType 'JGRACEIVT_UnitTest' has been reloaded successfully. i BIP9326I The source 'JGRACEIVTproject.generated.bar' has been successfully deployed. 										
15	Click the ‘Close’ button to close the ‘Progress Information’ window.											
16	<p>Click the ‘Deployment Log’ perspective at the bottom of the screen to see a log of your deployments.</p> <p>Expand the parent tree node to see all the messages as shown to the right.</p>	 <p>GetHello_JavaCompute.java</p> <pre> // ----- // Add user code below MBElement outRoot = outMessage.getRootElement(); MBElement outJsonRoot = outRoot.createElementAsLastChild(MbJSON.PARSER_NAME); MBElement outJsonData = outJsonRoot.createElementAsLastChild(MBElement.TYPE_N); MBElement outJsonCode = outJsonData.createElementAsLastChild(MBElement.TYPE_N); InetAddress ip = InetAddress.getLocalHost(); String hostName = ip.getHostName(); IntegrationServerProxy server = new IntegrationServerProxy(); IntegrationServerDescriptiveProperties dp = server.getIntegrationServerModel(); String acelver = dp.getVersion(); MBElement outJsonMsg = outJsonData.createElementAsLastChild(MBElement.TYPE_N); Hello from Blue Integration powered by App Connect Enterprise v"+acelver " and IBM Cloud Pak for Integration - Integration Server running in pr LocalDateTime now = LocalDateTime.now(); MBElement outJsonTime = outJsonData.createElementAsLastChild(MBElement.TYPE_N); // End of user code // ----- } catch (MbException e) { // Re-throw to allow Broker handling of MbException throw e; } catch (RuntimeException e) { // Re-throw to allow Broker handling of RuntimeException } </pre> <p>Deployment Log</p> <table border="1"> <thead> <tr> <th>Details</th> <th>Timestamp</th> </tr> </thead> <tbody> <tr> <td>[JGRACEIVT] has been successfully deployed to integration server Emer</td> <td>Fri Dec 08 20:53:10 EST 2023</td> </tr> <tr> <td>i BIP9332I RestAPI JGRACEIVT has been changed successfully.</td> <td>Fri Dec 08 20:53:10 EST 2023</td> </tr> <tr> <td>i BIP9332I TestProjectType 'JGRACEIVT_UnitTest' has been reloaded s</td> <td>Fri Dec 08 20:53:10 EST 2023</td> </tr> <tr> <td>i BIP9326I The source 'JGRACEIVTproject.generated.bar' has been succ</td> <td>Fri Dec 08 20:53:10 EST 2023</td> </tr> </tbody> </table>	Details	Timestamp	[JGRACEIVT] has been successfully deployed to integration server Emer	Fri Dec 08 20:53:10 EST 2023	i BIP9332I RestAPI JGRACEIVT has been changed successfully.	Fri Dec 08 20:53:10 EST 2023	i BIP9332I TestProjectType 'JGRACEIVT_UnitTest' has been reloaded s	Fri Dec 08 20:53:10 EST 2023	i BIP9326I The source 'JGRACEIVTproject.generated.bar' has been succ	Fri Dec 08 20:53:10 EST 2023
Details	Timestamp											
[JGRACEIVT] has been successfully deployed to integration server Emer	Fri Dec 08 20:53:10 EST 2023											
i BIP9332I RestAPI JGRACEIVT has been changed successfully.	Fri Dec 08 20:53:10 EST 2023											
i BIP9332I TestProjectType 'JGRACEIVT_UnitTest' has been reloaded s	Fri Dec 08 20:53:10 EST 2023											
i BIP9326I The source 'JGRACEIVTproject.generated.bar' has been succ	Fri Dec 08 20:53:10 EST 2023											

IBM TechXchange



17	<p>We will now validate that our changes are visible in this new version of the BAR.</p> <p>You can minimize the ACE IDE by clicking on the application workspace icon at the bottom of the screen.</p> <p>Open a Firefox dialog using the Desktop shortcut as shown to the right.</p>	
18	<p>Click on the “IBM ACE – INODEDEV” bookmark found in the bookmark bar as shown to the right.</p>	
19	<p>First, click on ‘Emerald-ACE-DEV’ and you will see the ‘JGRACEIVT’ api.</p> <p>Second, click on ‘JGRACEIVT’ and you will see the Documentation tab of the API as shown to the right.</p> <p>Notice the ‘Endpoint’ value. Copy this to your buffer.</p>	

20	<p>In your browser, open a new tab and paste the endpoint value into the browser and append /hello as shown below to the URL:</p> <p>http://ace-server:7080/jgraceit/v1/hello</p> <p>Press Enter.</p>	 <pre>JSON Raw Data Headers code: "CP418877" msg: "Hello from Blue" time: "2023-12-08T21:11:19.843"</pre>
21	<p>You should see ‘Blue’ in the message as shown above.</p> <p>You have completed this section! In the next section, we will explore a way to create an automated build process in a CI/CD pipeline as well as automatically run some unit tests against your ACE BAR.</p>	<p>NOTE: You would not want your developers performing automating deployments to QA or PROD from their local desktop IDE.</p> <p>You would want a proper DevOps pipeline to ensure you have the right controls and governance in place.</p>
End of Section – 3.1		

3.2 Setup a Jenkins CI Pipeline to Automate Build

3.2.1 Overview

In the past, many developers using their IDE would build a new BAR and copy it to a shared folder that others would pick up for deployment. The issue with this is you do not have traceability back to the build, you are not placing the artifact in a definitive asset repository that is immutable, and you don’t have any real record of the build or deployment.

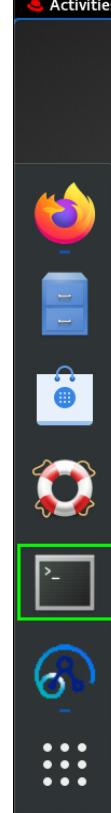
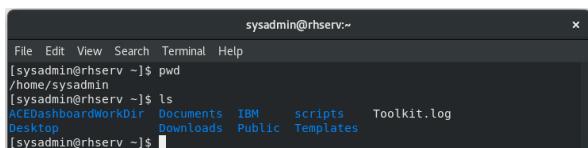
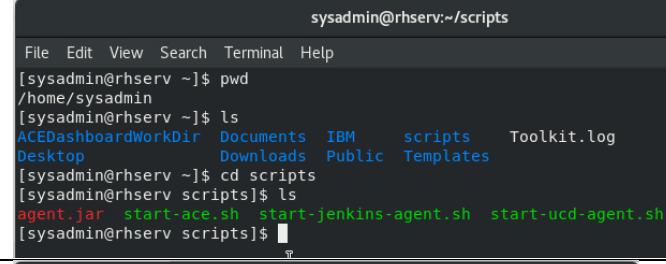
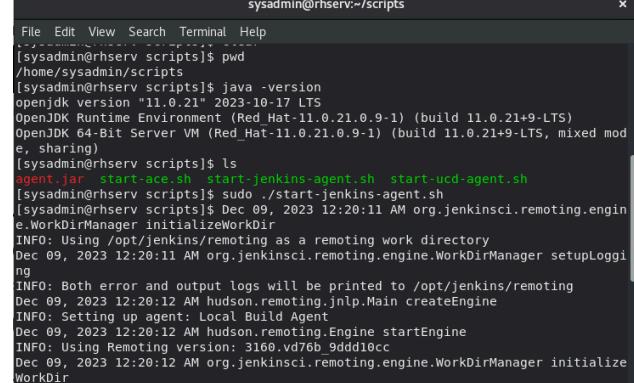
In this lab section, you will explore an example Continuous Integration (CI) Pipeline using Jenkins. This pipeline requires that developer’s check-in their code changes to a git repository. You could configure the pipeline to automatically trigger a build on code-commit if you like or on-demand.

You will see in this lab that we have incorporated a stage to run unit tests and to push the deployable artifact (a BAR file) to IBM DevOps Deploy’s version repository called ‘codestation’, which is like a JFrog Artifactory or Sonatype Nexus. Using this pipeline, we will have full traceability and trail of audit with proper governance.



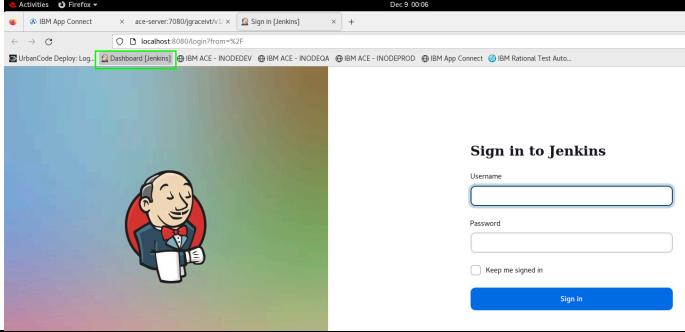
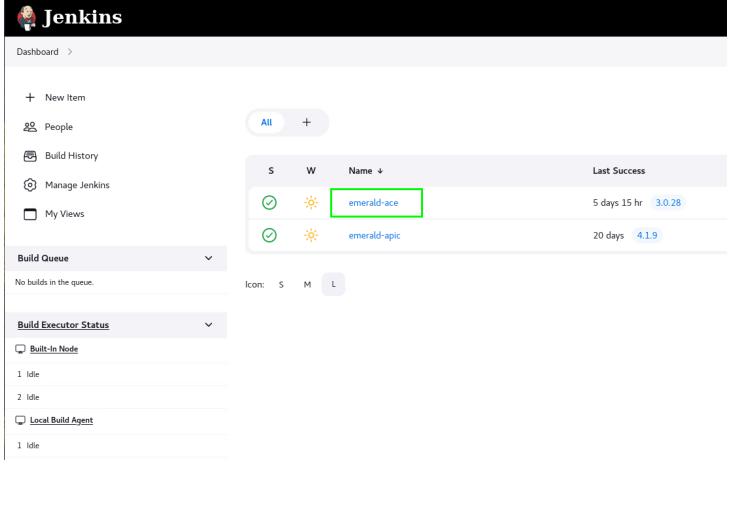
IBM TechXchange

3.2.2 Workbook Steps

Step	Details	Additional Information
1	<p>Before we open our Jenkins pipeline, we need to start our Jenkins build agent.</p> <p>Open a new terminal window by clicking on Activities at the upper left-hand corner of the screen and then click on the Terminal icon as shown to the right.</p> <p>NOTE: we are opening a new terminal to ensure JAVA_HOME is set to a Java 11 installation on the host.</p>	 
2	<p>In the terminal window, change directories to /home/sysadmin/scripts</p> <pre>\$ cd /home/sysadmin/scripts</pre>	
3	<p>Run the following command to start the Jenkins agent:</p> <pre>\$ sudo ./start-jenkins-agent.sh</pre>	

IBM TechXchange



Step	Details	Additional Information
4	<p>We also need to start the IBM DevOps Deploy Agent used for deployment.</p> <p>Run the following command to start the Deploy agent:</p> <pre>\$ sudo ./start-ucd-agent.sh</pre>	<pre>[sysadmin@rhserv bin]\$ cd /home/sysadmin/ [sysadmin@rhserv ~]\$ cd scripts [sysadmin@rhserv scripts]\$ ls agent.jar start-ace.sh start-jenkins-agent.sh start-ucd-agent.sh [sysadmin@rhserv scripts]\$ sudo ./start-ucd-agent.sh [sysadmin@rhserv scripts]\$</pre>
5	<p>We can now explore our Jenkins CI pipeline.</p> <p>In your Firefox browser window, create a new tab and click on the “Dashboard [Jenkins]” bookmark as shown to the right</p>	
6	Login to the console with a user of “ admin ” and a password of “ password ”	
7	Click on the “ emerald-ace ” pipeline as shown to the right	
INFO	This is an example pipeline that is pointed to a GitHub repository that I have created for this lab. Take time to explore the pipeline as much as you like.	

IBM TechXchange



Step	Details	Additional Information
8	<p>You will see the ‘emerald-ace’ pipeline and you can see each stage on the right-hand side which includes:</p> <ul style="list-style-type: none"> • Cloning the repository • Build • UnitTest • Push Artifact of Build to IBM DevOps Deploy 	
9	<p>Move mouse over the “GitHub” found in the left-hand pane.</p> <p>Right click on this link and select “Open Link in a New Tab” as shown to the right.</p>	
10	<p>This is the same code that you were working with in the first lab.</p> <p>The focus of this lab is to explain how you can setup a CI pipeline that includes automated unit testing and deployment as part of the process.</p>	

IBM TechXchange



Step	Details	Additional Information
11	<p>Return to the Jenkins tab in your browser.</p> <p>Notice the build history at the bottom left showing you the most recent builds.</p> <p>Notice the build tags that identify each build (3.0.28, 3.0.27, etc).</p>	
12	Click the “Configure” link in the left-hand pane to explore the build pipeline.	
13	<p>Scroll down to the Pipeline script area as shown to the right.</p> <p>NOTE: You can make the script pane larger by dragging the text area further down using your mouse.</p>	<pre> 1: node { 2: def GITHUB_REPO_URL='https://github.com/r3leng/ace-ibm/emerald-ace' 3: def REPOSITORY_APP_NAME='ace-ibm-backend-pipeline' 4: def SCM = git branch: 'main', url: "\${GITHUB_REPO_URL}" 5: def currentBuildNumber = env.BUILD_NUMBER 6: def currentBuildDisplayName = env.DISPLAY_NAME 7: def GIT_COMMIT = sh(readStdout: true, script: "git rev-parse HEAD").trim() 8: def GIT_COMMIT_ID=GIT_COMMIT 9: } 10: 11: stage ('cloning the repository'){ 12: currentBuild.displayName = "\${VERSION_NUMBER}.\${BUILD_NUMBER}" 13: echo currentBuild.displayName = "\${currentBuild.displayName}" 14: majorVersion ="\${BUILD_NUMBER}" 15: def scm = git branch: 'main', url: "\${GITHUB_REPO_URL}" 16: GIT_COMMIT = sh(readStdout: true, script: "git rev-parse HEAD").trim() 17: echo "GIT_COMMIT=\${GIT_COMMIT}" 18: } 19: 20: stage ('Build') { 21: sh 'cd \$WORKSPACE' 22: echo \$WORKSPACE 23: pwd 24: ls -al 25: source '/opt/IBM/ace-12.0.10.0/server/bin/mqsiprofile' 26: ibmint package --input-path . --output-bar jgr-cp4i-aceivt.bar' 27: } 28: 29: stage ('UnitTests') { 30: sh 'cd \$WORKSPACE' 31: echo \$WORKSPACE 32: } 33: 34: stage ('Push Artifact of Build to IBM DevOps Deploy') { 35: sh "ibmcloud devops artifact push \${currentBuildNumber} --name \${currentBuild.displayName} --path \${currentBuildNumber}/jgr-cp4i-aceivt.bar" 36: } 37: 38: stage ('Clean up') { 39: sh 'rm -rf \$WORKSPACE' 40: } 41: 42: stage ('Email') { 43: mail to: 'r3leng@ibm.com', subject: "Build \${currentBuildNumber} successful" 44: } 45: 46: stage ('Post') { 47: post { 48: success { 49: slackSend color: 'green', message: "Build \${currentBuildNumber} successful" 50: } 51: failure { 52: slackSend color: 'red', message: "Build \${currentBuildNumber} failed" 53: } 54: } 55: } 56: 57: stage ('Report') { 58: report coverage: 'coverage.html' 59: } 60: 61: stage ('Deploy') { 62: sh "ibmcloud devops artifact push \${currentBuildNumber} --name \${currentBuild.displayName} --path \${currentBuildNumber}/jgr-cp4i-aceivt.bar" 63: } 64: 65: stage ('Email') { 66: mail to: 'r3leng@ibm.com', subject: "Build \${currentBuildNumber} successful" 67: } 68: 69: stage ('Post') { 70: post { 71: success { 72: slackSend color: 'green', message: "Build \${currentBuildNumber} successful" 73: } 74: failure { 75: slackSend color: 'red', message: "Build \${currentBuildNumber} failed" 76: } 77: } 78: } 79: 80: stage ('Report') { 81: report coverage: 'coverage.html' 82: } 83: 84: stage ('Deploy') { 85: sh "ibmcloud devops artifact push \${currentBuildNumber} --name \${currentBuild.displayName} --path \${currentBuildNumber}/jgr-cp4i-aceivt.bar" 86: } 87: 88: stage ('Email') { 89: mail to: 'r3leng@ibm.com', subject: "Build \${currentBuildNumber} successful" 90: } 91: 92: stage ('Post') { 93: post { 94: success { 95: slackSend color: 'green', message: "Build \${currentBuildNumber} successful" 96: } 97: failure { 98: slackSend color: 'red', message: "Build \${currentBuildNumber} failed" 99: } 100 } 101 } 102 </pre>
14	In the pipeline view, you will see each stage of the build process. Our first stage is to ‘Clone the Repository’ which will pull down your source code from git to a workspace that is used by the Jenkins Agent to perform the build.	<pre> stage ('cloning the repository'){ currentBuild.displayName = "\${VERSION_NUMBER}.\${BUILD_NUMBER}" echo currentBuild.displayName = "\${currentBuild.displayName}" majorVersion ="\${BUILD_NUMBER}" def scm = git branch: 'main', url: "\${GITHUB_REPO_URL}" GIT_COMMIT = sh(readStdout: true, script: "git rev-parse HEAD").trim() echo "GIT_COMMIT=\${GIT_COMMIT}" } </pre>
15	Next, you see the ‘Build’ stage of the process. This is where the agent will use the ‘ibmint’ command to build your broker archive (BAR) image.	<pre> stage ('build') { sh ''#!/bin/bash echo \$WORKSPACE pwd ls -al source '/opt/IBM/ace-12.0.10.0/server/bin/mqsiprofile' ibmint package --input-path . --output-bar jgr-cp4i-aceivt.bar' } </pre>

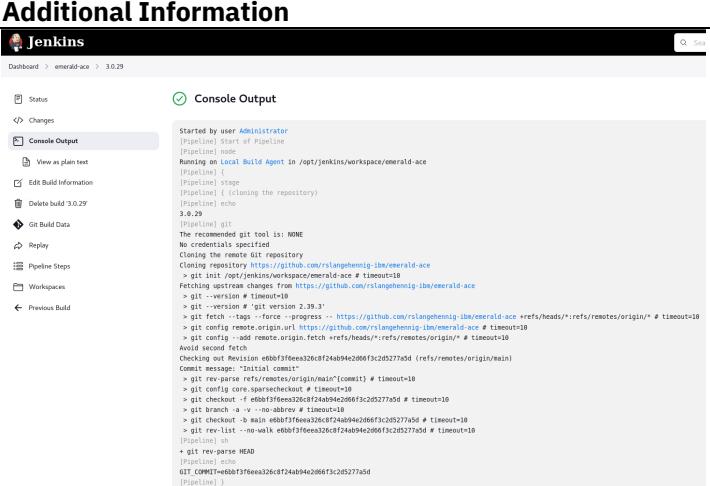
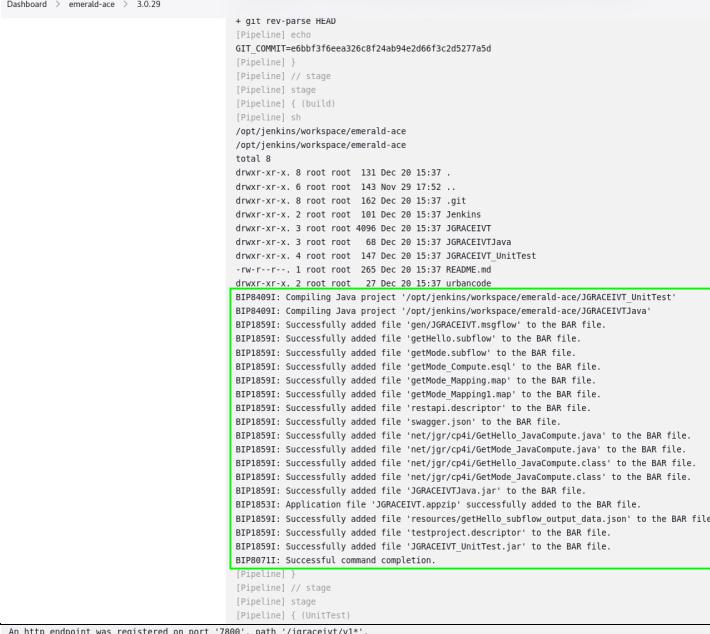
IBM TechXchange



Step	Details	Additional Information																				
16	As you scroll down, you will find a ‘UnitTest’ stage of the pipeline to run unit tests against your newly built BAR file. This helps to ensure your changes did not break any functionality.	<pre>stage('UnitTest') { sh ''"/bin/bash # Start with a clean setup - could put the work directory anywhere but /tmp is good enough for now. rm -rf /tmp/ace-server source /opt/IBM/ace-12.0.10.0/server/bin/mqsisprofile" # Create work directory and deploy the application BAR file mqsiscreateworkdir /tmp/ace-server ibmst deploy --input-bar-file jgr-cp4i-aceit.bar --output-work-directory /tmp/ace-server # Compile and deploy the unit tests for the application ibmst deploy --input-path . --output-work-directory /tmp/ace-server --project JGRACEIVT_UnitTest # Run the tests; could optimize first if needed IntegrationServer -w /tmp/ace-server --test-project JGRACEIVT_UnitTest # Could report the JUnit tests results to Jenkins if needed }</pre>																				
17	For now, scroll back to the top of the browser and click on ‘emerald-ace’ in the crumb trail as shown to the right.																					
18	Let's start a build and watch it run. Click the ‘Build Now’ button in the left-hand pane.																					
19	The build will initiate, and you will see a new build labeled ‘3.0.29’ in the bottom left-hand ‘Build History’ pane as shown to the right.	<table border="1"> <caption>Average stage times: (Average full run time: ~19s)</caption> <thead> <tr> <th>Stage</th> <th>cloning the repository</th> <th>build</th> <th>UnitTest</th> <th>Push Artifact of Build to IBM DevOps Deploy</th> </tr> </thead> <tbody> <tr> <td>Dec 20 15:37</td> <td>3s</td> <td>3s</td> <td>11s</td> <td>410ms</td> </tr> <tr> <td>Dec 15 00:17</td> <td>5s</td> <td>2s</td> <td>10s</td> <td>351ms</td> </tr> <tr> <td></td> <td>970ms</td> <td>3s</td> <td>13s</td> <td>470ms</td> </tr> </tbody> </table>	Stage	cloning the repository	build	UnitTest	Push Artifact of Build to IBM DevOps Deploy	Dec 20 15:37	3s	3s	11s	410ms	Dec 15 00:17	5s	2s	10s	351ms		970ms	3s	13s	470ms
Stage	cloning the repository	build	UnitTest	Push Artifact of Build to IBM DevOps Deploy																		
Dec 20 15:37	3s	3s	11s	410ms																		
Dec 15 00:17	5s	2s	10s	351ms																		
	970ms	3s	13s	470ms																		

IBM TechXchange



Step	Details	Additional Information
20	<p>When the build completes, click on the green checkmark next to the 3.0.29 build label in the Build History as shown above.</p> <p>This will allow you to see the Console Output for the build shown to the right.</p>	 <pre> Started by user Administrator [Pipeline] Start of Pipeline [Pipeline] Stage [Pipeline] (cloning the repository) [Pipeline] echo 3.0.29 [Pipeline] Stage [Pipeline] (cloning the repository) [Pipeline] echo The recommended git tool is: NONE No credentials specified Cloning the remote Git repository (Cloning via scp to avoid https://) > git init /opt/jenkins/workspace/emerald-ace & timeout=10 Fetching upstream changes from https://github.com/rslangheinig/ibm/emerald-ace > git -version # timeout=10 git version 2.39.3 > git config --list # timeout=10 git config --get remote.origin.url https://github.com/rslangheinig/ibm/emerald-ace & timeout=10 > git config remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10 ↳ build step test Checking out Revision e60bf3f6eea326c8f24ab94e2d66f3c2d5277a5d Commit message: 'Initial commit' > git rev-parse HEAD^ & timeout=10 > git config core.sparsecheckout & timeout=10 > git checkout -f e60bf3f6eea326c8f24ab94e2d66f3c2d5277a5d & timeout=10 > git branch -v -a & timeout=10 git branch -v -a * main e60bf3f6eea326c8f24ab94e2d66f3c2d5277a5d > git rev-list --no-walk e60bf3f6eea326c8f24ab94e2d66f3c2d5277a5d & timeout=10 [Pipeline] sh [Pipeline] stage GIT_COMMIT=e60bf3f6eea326c8f24ab94e2d66f3c2d5277a5d [Pipeline] // stage </pre>
21	Scroll down and you will see the successful build of the BAR.	 <pre> + git rev-parse HEAD [Pipeline] echo GIT_COMMIT=e60bf3f6eea326c8f24ab94e2d66f3c2d5277a5d [Pipeline] // stage [Pipeline] stage [Pipeline] (build) [Pipeline] sh /opt/jenkins/workspace/emerald-ace /opt/jenkins/workspace/emerald-ace total 8 drwxr-xr-x. 8 root root 131 Dec 20 15:37 . drwxr-xr-x. 6 root root 143 Nov 29 15:52 .. drwxr-xr-x. 8 root root 162 Dec 20 15:37 .git drwxr-xr-x. 2 root root 101 Dec 20 15:37 Jenkins drwxr-xr-x. 3 root root 4096 Dec 20 15:37 JGRACEIVT drwxr-xr-x. 3 root root 68 Dec 20 15:37 JGRACEIVTJava drwxr-xr-x. 4 root root 147 Dec 20 15:37 JGRACEIVT_UnitTest -rw-r--r--. 1 root root 265 Dec 20 15:37 README.md drwxr-xr-x. 2 root root 27 Dec 20 15:37 urbancode BIP8491: Compiling Java project '/opt/jenkins/workspace/emerald-ace/JGRACEIVT_UnitTest' BIP8491: Compiling Java project '/opt/jenkins/workspace/emerald-ace/JGRACEIVTJava' BIP18591: Successfully added file 'gen/JGRACEIVT.msgflw' to the BAR file. BIP18591: Successfully added file 'getHello.subflow' to the BAR file. BIP18591: Successfully added file 'getMode.subflow' to the BAR file. BIP18591: Successfully added file 'getMode.Compute.esql' to the BAR file. BIP18591: Successfully added file 'getMode.Mapping.map' to the BAR file. BIP18591: Successfully added file 'getMode.Mapping1.map' to the BAR file. BIP18591: Successfully added file 'restapi.descriptor' to the BAR file. BIP18591: Successfully added file 'swagger.json' to the BAR file. BIP18591: Successfully added file 'net/grpcapi/getHello.JavaCompute.java' to the BAR file. BIP18591: Successfully added file 'net/grpcapi/getHello.JavaCompute.java' to the BAR file. BIP18591: Successfully added file 'net/grpcapi/getHello.JavaCompute.class' to the BAR file. BIP18591: Successfully added file 'net/grpcapi/getMode.JavaCompute.class' to the BAR file. BIP18591: Successfully added file 'JGRACEIVTJava.jar' to the BAR file. BIP18591: Application file 'JGRACEIVT_App.zip' successfully added to the BAR file. BIP18591: Successfully added file 'resources/getHello.subflow.output.data.json' to the BAR file. BIP18591: Successfully added file 'testproject.descriptor' to the BAR file. BIP18591: Successfully added file 'JGRACEIVT_UnitTest.jar' to the BAR file. BIP8871: Successful command completion. [Pipeline] sh [Pipeline] // stage [Pipeline] stage [Pipeline] (UnitTest) </pre>
22	Scroll down a little further and you will see the Unit Tests were run and that they were successful.	<p>An http endpoint was registered on port '7800', path '/graceivt/v1'.</p> <p>2023-12-20 15:37:51.147204: BIP31321: The HTTP Listener has started listening on port '7800' for 'http' connections.</p> <p>2023-12-20 15:37:51.147205: BIP19061: Listening on HTTP URL '/jgraceivt/v1'.</p> <p>Started native listener for HTTP input node on port 7800 for URL /jgraceivt/v1*</p> <p>2023-12-20 15:37:51.147680: BIP22691: Deployed resource 'gen.JGRACEIVT' (uid:gen.JGRACEIVT,type:'MessageFlow') started successfully.</p> <p>2023-12-20 15:37:51.147697: BIP93321: RestAPI 'JGRACEIVT' has been reloaded successfully.</p> <p>2023-12-20 15:37:51.147702: BIP93321: TestProjectType 'JGRACEIVT_UnitTest' has been reloaded successfully.</p> <p>2023-12-20 15:37:51.149984: BIP28601: IBM App Connect Enterprise administration security is inactive.</p> <p>2023-12-20 15:37:51.171760: BIP31321: The HTTP Listener has started listening on port '7800' for 'RestAdmin http' connections.</p> <p>2023-12-20 15:37:51.172612: BIP99101: Running tests in test project 'JGRACEIVT_UnitTest'.</p> <p>2023-12-20 15:37:51.568 1 STARTING TEST:JGRACEIVT_getMode_subflow_TestCase_001</p> <p>2023-12-20 15:37:51.693 1 FINISHED TEST:JGRACEIVT_getMode_subflow_TestCase_001():SUCCESSFUL</p> <p>2023-12-20 15:37:51.695 1 STARTING TEST:JGRACEIVT_getMode_subflow_TestCase_002</p> <p>2023-12-20 15:37:51.700 1 FINISHED TEST:JGRACEIVT_getMode_subflow_TestCase_002():SUCCESSFUL</p> <p>2023-12-20 15:37:51.729 1 STARTING TEST:JGRACEIVT_getMode_subflow_TestCase_003</p> <p>2023-12-20 15:37:51.908 1 FINISHED TEST:JGRACEIVT_getMode_subflow_TestCase_003():SUCCESSFUL</p> <p>2023-12-20 15:37:51.901 1 STARTING TEST:JGRACEIVT_getMode_subflow_TestCase_004</p> <p>2023-12-20 15:37:51.917 1 FINISHED TEST:JGRACEIVT_getMode_subflow_TestCase_004():SUCCESSFUL</p> <p>2023-12-20 15:37:51.920 1 STARTING TEST:JGRACEIVT_getHello_subflow_TestCase_001</p> <p>2023-12-20 15:37:52.263 1 FINISHED TEST:JGRACEIVT_getHello_subflow_TestCase_001():SUCCESSFUL</p> <p>2023-12-20 15:37:52.273 1</p> <p>TEST RESULTS:</p> <p>test.JGRACEIVT_getHello_subflow_001_Test: JGRACEIVT_getHello_subflow_TestCase_001():SUCCESSFUL</p> <p>test.JGRACEIVT_getMode_subflow_001_Test: JGRACEIVT_getMode_subflow_TestCase_001():SUCCESSFUL JGRACEIVT_getMode_subflow_TestCase_002():SUCCESSFUL JGRACEIVT_getMode_subflow_TestCase_003():SUCCESSFUL JGRACEIVT_getMode_subflow_TestCase_004():SUCCESSFUL</p> <p>TOTALS: PASSED :5 FAILED :0 ABORTED :0 TIME(secs):0.76</p> <p>2023-12-20 15:37:52.274476: BIP9111: Running tests in test project 'JGRACEIVT_UnitTest' completed successfully.</p>

IBM TechXchange



Step	Details	Additional Information
23	If you scroll down, you will notice one more stage called ' Push Artifact of Build to IBM DevOps Deploy '. This stage is not configured yet. Let's modify this build project to perform this step so we can deploy our BAR file to our target environments.	We will work on this stage together to enable it to push the BAR to the DevOps Deploy Server for automated deployment.
24	In your browser window, create a new tab. Then click on the ' UrbanCode Deploy: Login ' bookmark in the toolbar as shown to the right.	
25	Login as the user ' admin ' and with a password of ' admin '	
26	Click on Components at the top of the page as shown to the right	
27	Click on 'emerald-ACE-BAR' as shown above to view this component	

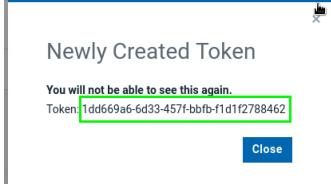
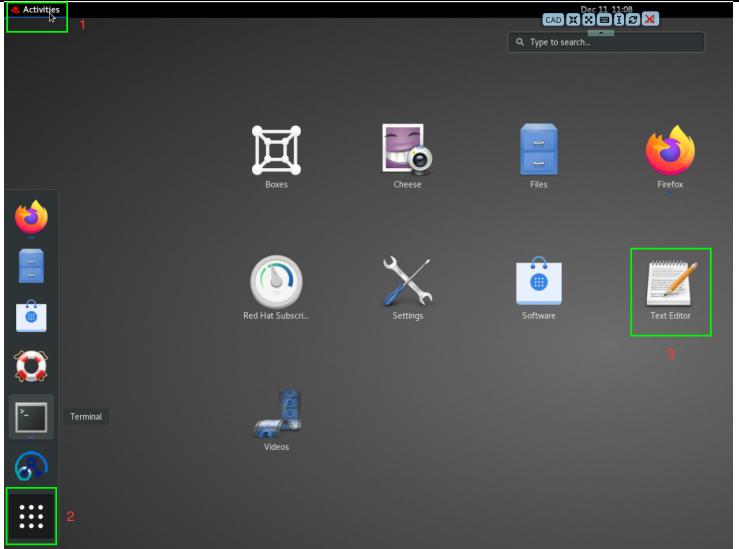
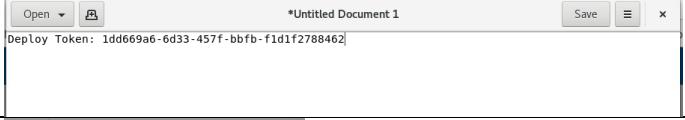
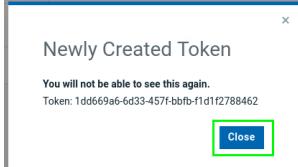
IBM TechXchange



Step	Details	Additional Information
28	<p>Notice the component versions listed.</p> <p>This is where are Jenkins build process will push the built artifact as a new component version using the build tag.</p>	
29	<p>In order for the Jenkins build to push the new component version, we need to create an Authentication Token that it can use.</p> <p>Click on the Settings tab at the top right-hand side of the page.</p>	
30	Click on Tokens found in the Security section of the Settings page.	
31	Click on the ' Create Token ' button	
32	<p>The Create Token dialog will appear. Use the following follows:</p> <p>Description: Jenkins Token User: admin Expiration Date: 12/13/2024 Expiration Time: 08:00 AM</p>	
33	Click Save	



IBM TechXchange

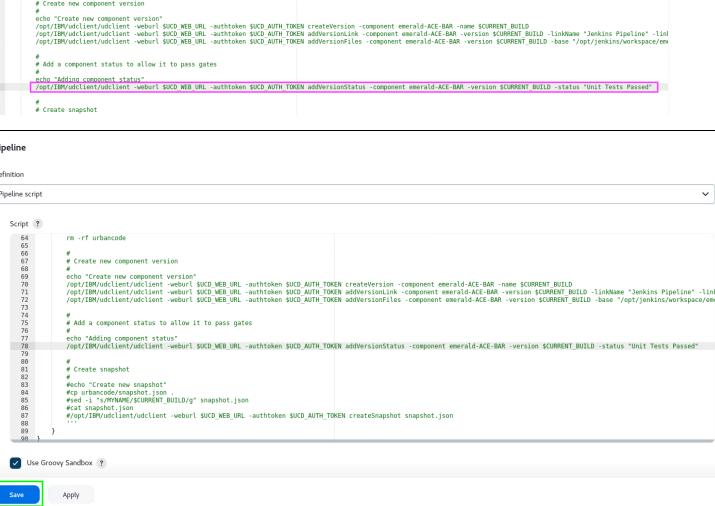
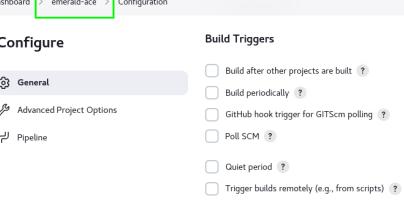
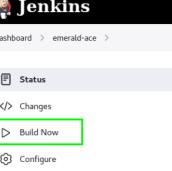
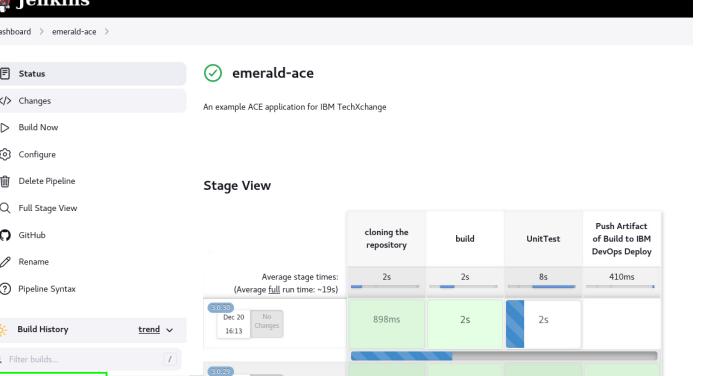
Step	Details	Additional Information
34	In the Newly Created Token dialog that appears, copy the Token value to your buffer.	
35	Open a Text Editor by clicking on Activities at the top left-hand side of the page (step 1) and then click on the dots (step 2) to see all the applications.	
36	Click on the Text Editor icon (step 3) as shown above to start the editor.	
37	In the terminal window, enter 'Deploy Token:' and then paste your auth token.	
38	Close the 'Newly Created Token' dialog in the Deploy web console by clicking the 'Close' button.	

IBM TechXchange



Step	Details	Additional Information
39	Return to your Jenkins tab in your browser. Return to the ‘emerald-ace’ project as shown to the right. You can click on ‘emerald-ace’ in the crumb trail at upper-right of screen to return to this page.	
40	Click on ‘Configure’ for the ‘emerald-ace’ pipeline as shown above	
41	Scroll down to the Pipeline script and expand the script pane to give you more space to work with.	
42	In the script area, scroll down to the ‘Push Artifact of Build to IBM DevOps Deploy’ stage.	
43	Notice line 50 of the script: export UCD_AUTH_TOKEN="replace-me"	
44	Paste your new auth token value to replace the ‘replace-me’ text as shown in the example to the right. NOTE: normally this would be configured as a secure property.	
45	Scroll down and remove the # at the beginning of lines 70-72 to uncomment them.	

IBM TechXchange

Step	Details	Additional Information
	This will allow the pipeline to create a new component version using the IBM DevOps Deploy udclient command.	
46	Scroll down to line 78 and remove the # to uncomment this line so it will add a component status of ‘Unit Tests Passed’	<pre> 66 # Create new component version 67 echo "Create new component version" 68 /opt/IBU/udclient/udclient -weburl \$UCD_WEB_URL -authtoken \$UCD_AUTH_TOKEN createVersion -component emerald-ACE-BAR -name CURRENT_BUILD 69 /opt/IBU/udclient/udclient -weburl \$UCD_WEB_URL -authtoken \$UCD_AUTH_TOKEN addVersionLink -component emerald-ACE-BAR -version CURRENT_BUILD -linkName "Jenkins Pipeline" -linkType "UD" 70 /opt/IBU/udclient/udclient -weburl \$UCD_WEB_URL -authtoken \$UCD_AUTH_TOKEN addVersionStatus -component emerald-ACE-BAR -version CURRENT_BUILD -base "/opt/jenkins/workspace/em 71 /opt/IBU/udclient/udclient -weburl \$UCD_WEB_URL -authtoken \$UCD_AUTH_TOKEN addVersionStatus -component emerald-ACE-BAR -version CURRENT_BUILD -status "Unit Tests Paused" 72 73 74 # Add a component status to allow it to pass gates 75 76 echo "Adding component status" 77 78 echo "#!/bin/sh\n# curl -XPUT \$UCD_WEB_URL/authToken:\$UCD_AUTH_TOKEN/udclient/udclient -weburl \$UCD_WEB_URL -authtoken \$UCD_AUTH_TOKEN addVersionStatus -component emerald-ACE-BAR -version CURRENT_BUILD -status \"Unit Tests Passed\"" 79 80 #!/bin/sh\n# curl -XPUT \$UCD_WEB_URL/authToken:\$UCD_AUTH_TOKEN/udclient/udclient -weburl \$UCD_WEB_URL -authtoken \$UCD_AUTH_TOKEN addVersionStatus -component emerald-ACE-BAR -version CURRENT_BUILD -status \"Unit Tests Passed\"" 81 82 83 # Create snapshot 84 85 rm -rf urbancode 86 /opt/IBU/udclient/udclient -weburl \$UCD_WEB_URL -authtoken \$UCD_AUTH_TOKEN createSnapshot 87 /opt/IBU/udclient/udclient -weburl \$UCD_WEB_URL -authtoken \$UCD_AUTH_TOKEN snapshot.json 88 /opt/IBU/udclient/udclient -weburl \$UCD_WEB_URL -authtoken \$UCD_AUTH_TOKEN createSnapshot snapshot.json 89 90 }</pre>
47	Click ‘Save’ to save your updates to the pipeline script. NOTE: you do not need to uncomment lines 83-87. Leave them as they were.	
48	In the crumb trail at the top left, click on ‘emerald-ace’	
49	Initiate a new build by clicking the ‘Build Now’ button	
50	You will see a new build initiate. The build should complete successfully.	

IBM TechXchange



Step	Details	Additional Information
51	<p>Examine the build output and you should see that it was successful in creating a new component version to push the new deployable artifact.</p> <p>NOTE: DevOps Deploy can integrate with other definitive asset repositories like JFrog Artifactory or Sonatype Nexus.</p>	<pre> Dashboard / emerald-ace 3.0.30 Create new component version { "id": "1f3c4ef7-0b00-442d-b7e2-7d71e0f733f7", "name": "3.0.30", "type": "Docker", "created": "2023-06-08T08:00:00Z", "active": true, "version": "3.0.30", "sizeInBytes": 8, "tags": "ACE,BAR", "cacheable": true, "properties": [], "parent": null, "git": "1b48138-0b00-4520-9d13-4c9308dc791", "parentGit": "1f3c4ef7-0b00-442d-b7e2-7d71e0f733f7", "parentName": "emerald-ACE-BAR", "parentVersion": "3.0.30", "parentSize": 8, "parentCacheable": false, "parentProperties": [] } parent: "components/1b48138-0b00-4520-9d13-4c9308dc791" parentVersion: "3.0.30" parentSize: 8 parentCacheable: false parentProperties: [] Operation succeeded. Adding component status [Pipeline] // stage [Pipeline] / [Pipeline] // node [Pipeline] / [Pipeline] Pipeline Finished: SUCCESS </pre>
52	In your browser, click on the 'DevOps Deploy' tab.	
53	<p>Click on Components at the top of the page.</p> <p>NOTE: you may need to re-authenticate using username of 'admin' and password of 'admin'</p>	
54	Click on the 'emerald-ACE-BAR' component.	
55	You should see a new component version, 3.0.30, in the list of Component Versions as shown to the right.	

IBM TechXchange



Step	Details	Additional Information
56	<p>Click on 3.0.30 to view the new version.</p> <p>Notice the immutable artifact in the Artifacts list, jgr-cp4i-aceivt.bar. A component version can have one or more files.</p>	
57	<p>Notice the link back to the Jenkins pipeline (see screenshot above) so you have traceability back to the build and the git commit.</p> <p>If you right click on the link, you can open this link in a new tab. It will take you to the build 'Console Output' as shown to the right.</p>	
58	<p>You have completed this lab.</p> <p>In the next lab, we will look at the deployment of the application in IBM DevOps Deploy.</p>	

End of Section – 3.2

3.3 Review the Emerald eStore Application in IBM DevOps Deploy

3.3.1 Overview

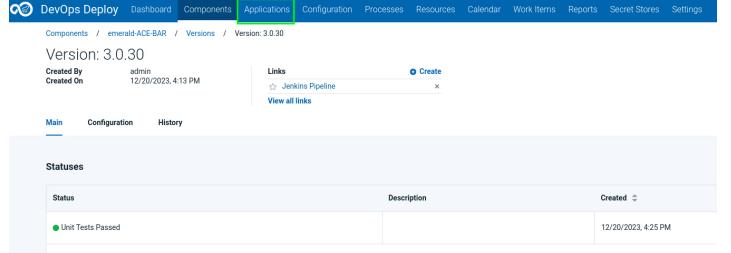
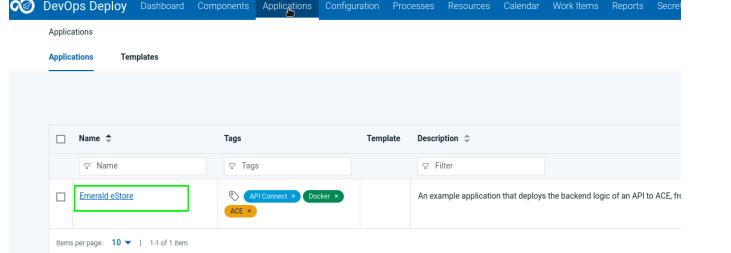
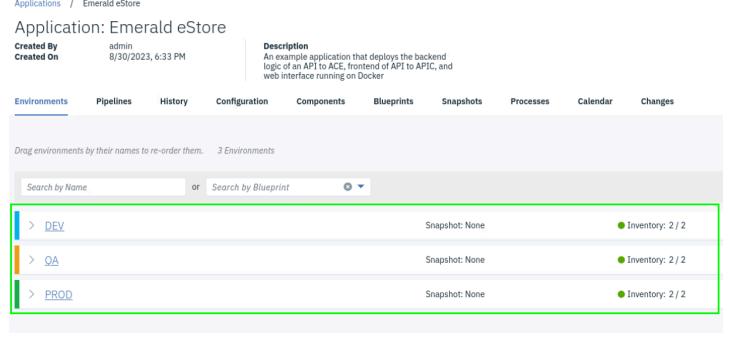
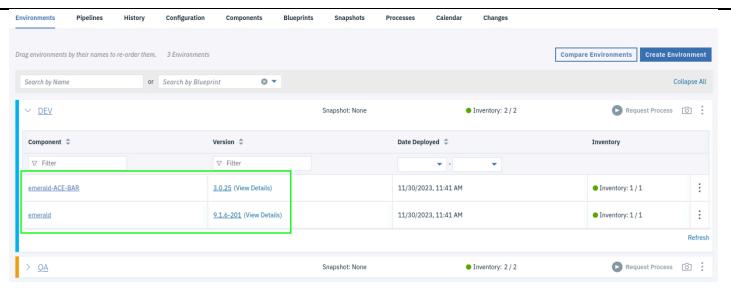
In this lab section, you will explore the ‘Emerald eStore’ application in IBM DevOps Deploy. The solution leverages an application model that is comprised of three parts:

1. Environments
2. Components that make up your application. You can have more than one.
3. Processes to perform deployments.

IBM TechXchange



3.3.2 Workbook steps

Step	Details	Additional Information
1	In your Firefox browser, return to the “Deploy” tab.	
2	Click on the “Applications” tab at the top of the page as highlighted above.	NOTE: you may be asked to re-authenticate to the Deploy Console using your admin/admin credentials
3	You will see the “ Emerald eStore ” application in the Applications list. Click on “Emerald eStore” to view the application	
4	The first part of the application model which we can see are the target environments which are DEV, QA, and PROD which are already configured for you.	
5	Expand the DEV environment by clicking the arrow next to the DEV environment. You will see the second part of the model which are the components. In this case, we have a component named “emerald” and our ACE component named “emerald-ACE-BAR”. Notice that IBM DevOps Deploy tracks inventory of what is currently deployed in the DEV environment (e.g. 9.1.6-201 for “emerald” which is usually a build tag for traceability)	
INFO	To see all the components, you can also click on the “Components” sub-tab. Not all components will necessarily track inventory. For instance, the “emerald-RTAS” component will run operational processes to initiate test execution using Rational Test Automation Server.	

IBM TechXchange



Step	Details	Additional Information												
6	<p>The last part of the application model are processes. For now, click on the “Processes” sub-tab for our application to see our application process which will perform the deployment of the Emerald eStore application and more.</p> <p>We will be updating a process as part of the next lab section.</p>	<p>Application: Emerald eStore</p> <p>Created By admin Created On 8/30/2023, 6:33 PM</p> <p>Description An example application that deploys the backend logic of an API to ACE, frontend of API to APIC, and web interface running on Docker</p> <p>Environments Pipelines History Configuration Components Blueprints Snapshots Processes Calendar Changes</p> <table border="1"> <thead> <tr> <th>Process</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Deploy Emerald</td> <td>This deploys the ACE API and Emerald Web App</td> </tr> <tr> <td>Deploy Emerald ACE</td> <td>This only deploys the ACE API (backend logic)</td> </tr> <tr> <td>Deploy Emerald APIC</td> <td>This will publish an APIC Product to a Sandbox Catalog</td> </tr> <tr> <td>Deploy Emerald WEB</td> <td>The only deploys the Emerald Web Interface</td> </tr> <tr> <td>Run ACE RTAS Test</td> <td></td> </tr> </tbody> </table>	Process	Description	Deploy Emerald	This deploys the ACE API and Emerald Web App	Deploy Emerald ACE	This only deploys the ACE API (backend logic)	Deploy Emerald APIC	This will publish an APIC Product to a Sandbox Catalog	Deploy Emerald WEB	The only deploys the Emerald Web Interface	Run ACE RTAS Test	
Process	Description													
Deploy Emerald	This deploys the ACE API and Emerald Web App													
Deploy Emerald ACE	This only deploys the ACE API (backend logic)													
Deploy Emerald APIC	This will publish an APIC Product to a Sandbox Catalog													
Deploy Emerald WEB	The only deploys the Emerald Web Interface													
Run ACE RTAS Test														

End of Section – 3.3

3.4 Work with the emerald-ACE-BAR Component in Deploy

3.4.1 Overview

In this lab section, you will review the emerald-BAR component which will be used to deploy your broker archive that was built earlier in the lab. You will update the component process and then test the deployment to a DEV environment.

3.4.2 Workbook steps

Step	Details	Additional Information																																								
1	Continuing to work in the IBM DevOps Deploy application, click on the Components tab at the top of the page	<p>DevOps Deploy / Components</p> <p>Components Templates</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Tags</th> <th>Latest Import</th> <th>Latest Version</th> <th>Template</th> <th>Description</th> <th>Created</th> <th>By</th> </tr> </thead> <tbody> <tr> <td>emeral</td> <td>Tags</td> <td>Successful</td> <td>Latest</td> <td>Docker Template</td> <td>An App Connect Enterprise (ACE) broker archive</td> <td>8/30/2023, 6:16 PM</td> <td>admin</td> </tr> <tr> <td>emerald-ACE-BAR</td> <td>Tags</td> <td>3.0.0</td> <td>3.0.0</td> <td></td> <td>An App Connect Enterprise (ACE) broker archive</td> <td>11/18/2023, 10:57 PM</td> <td>admin</td> </tr> <tr> <td>emerald-APC-Product</td> <td>Tags</td> <td>4.1.8</td> <td>4.1.8</td> <td></td> <td>An example API Connect API definition that uses a sample OpenAPI spec. This points to an ACE endpoint</td> <td>11/20/2023, 4:41 PM</td> <td>admin</td> </tr> <tr> <td>emerald-RTAS</td> <td>Tags</td> <td>NO VERSION</td> <td>NO VERSION</td> <td></td> <td>This component will run tests for Emerald eStore</td> <td>11/20/2023, 5:58 PM</td> <td>admin</td> </tr> </tbody> </table>	Name	Tags	Latest Import	Latest Version	Template	Description	Created	By	emeral	Tags	Successful	Latest	Docker Template	An App Connect Enterprise (ACE) broker archive	8/30/2023, 6:16 PM	admin	emerald-ACE-BAR	Tags	3.0.0	3.0.0		An App Connect Enterprise (ACE) broker archive	11/18/2023, 10:57 PM	admin	emerald-APC-Product	Tags	4.1.8	4.1.8		An example API Connect API definition that uses a sample OpenAPI spec. This points to an ACE endpoint	11/20/2023, 4:41 PM	admin	emerald-RTAS	Tags	NO VERSION	NO VERSION		This component will run tests for Emerald eStore	11/20/2023, 5:58 PM	admin
Name	Tags	Latest Import	Latest Version	Template	Description	Created	By																																			
emeral	Tags	Successful	Latest	Docker Template	An App Connect Enterprise (ACE) broker archive	8/30/2023, 6:16 PM	admin																																			
emerald-ACE-BAR	Tags	3.0.0	3.0.0		An App Connect Enterprise (ACE) broker archive	11/18/2023, 10:57 PM	admin																																			
emerald-APC-Product	Tags	4.1.8	4.1.8		An example API Connect API definition that uses a sample OpenAPI spec. This points to an ACE endpoint	11/20/2023, 4:41 PM	admin																																			
emerald-RTAS	Tags	NO VERSION	NO VERSION		This component will run tests for Emerald eStore	11/20/2023, 5:58 PM	admin																																			
2	Click on the “emerald-ACE-BAR” component as highlighted above																																									
3	Click on the “History” sub-tab as highlighted to the right.	<p>Components / emerald-ACE-BAR</p> <p>Component: emerald-ACE-BAR</p> <p>Created By admin Created On 11/18/2023, 10:57 PM</p> <p>Description An App Connect Enterprise (ACE) broker archive</p> <p>Used By Emerald eStore</p> <p>Versions History Usage Configuration Calendar Processes Changes</p> <p>Component Versions</p> <table border="1"> <thead> <tr> <th>Version</th> <th>Statuses</th> <th>Type</th> <th>Created By</th> </tr> </thead> <tbody> <tr> <td>3.0.0</td> <td>Unit Tests Passed</td> <td>Full</td> <td>admin</td> </tr> <tr> <td>3.0.25</td> <td>Unit Tests Passed Functional Tests Passed</td> <td>Full</td> <td>admin</td> </tr> </tbody> </table>	Version	Statuses	Type	Created By	3.0.0	Unit Tests Passed	Full	admin	3.0.25	Unit Tests Passed Functional Tests Passed	Full	admin																												
Version	Statuses	Type	Created By																																							
3.0.0	Unit Tests Passed	Full	admin																																							
3.0.25	Unit Tests Passed Functional Tests Passed	Full	admin																																							

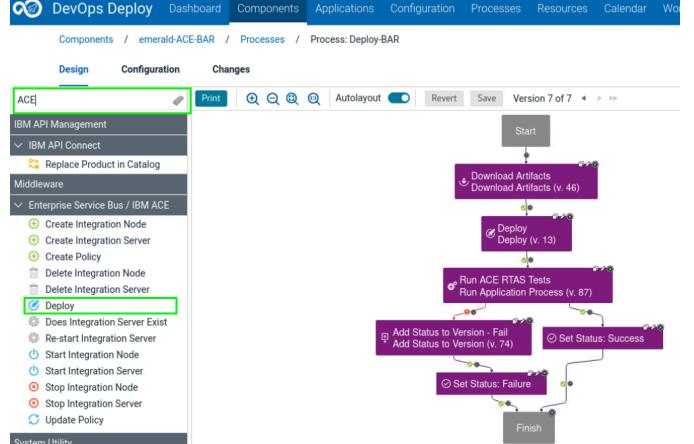
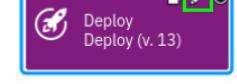
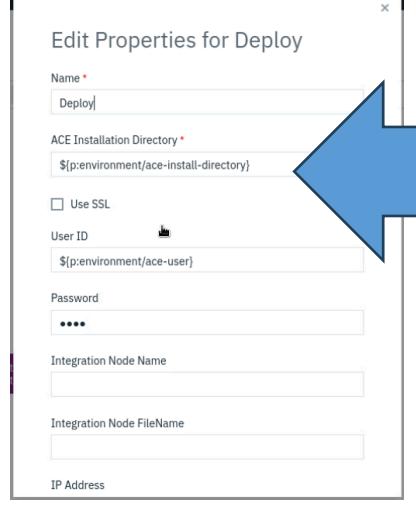
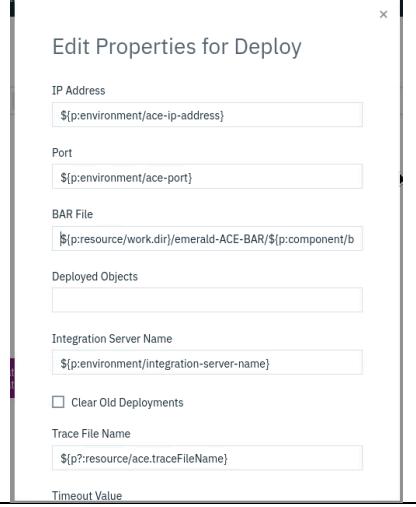
IBM TechXchange



Step	Details	Additional Information
4	This view will show you which versions of a component are deployed in each target environment.	
5	Click on the “Usage” sub-tab just to the right of the History sub-tab.	
6	The Usage tab will allow you to see which component version is currently deployed to a target environment, what used to be deployed in the environment prior, and at what days over time was it deployed there.	
7	Click on the “Processes” sub-tab (see below)	
8	Click on the “Deploy-BAR” process as shown to the right	
9	<p>The visual process designer will appear and display the component process.</p> <p>This left-hand pane of the designer is where you will find all the steps that you can drag and drop onto the process design. You can add to these steps by installing automation plugins.</p> <p>Notice the flow of the process. The first step downloads the artifact for deployment from IBM DevOps Deploy's codestation repository.</p>	

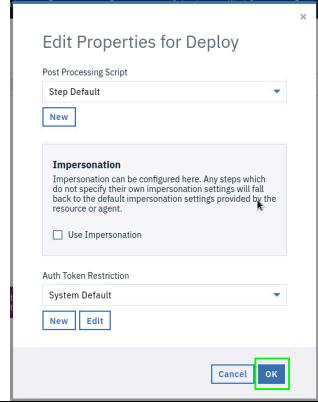
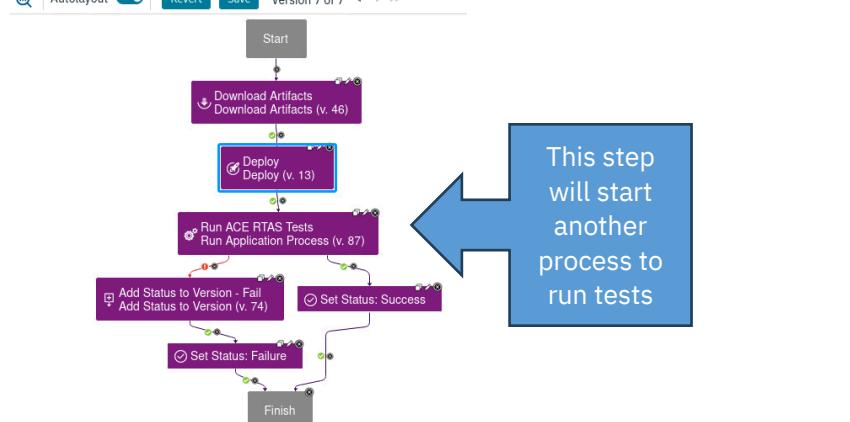
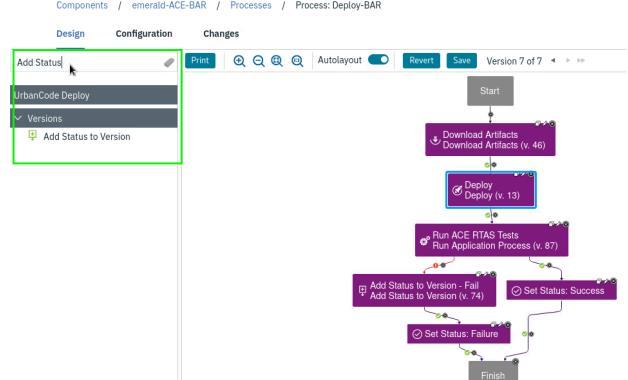
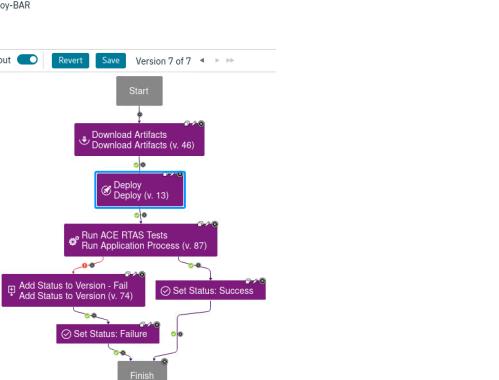
IBM TechXchange



Step	Details	Additional Information
10	<p>In the ‘Type to filter’ text area found at the upper left, type ‘ACE’ as shown.</p> <p>The left-hand step-palette will display all of the IBM ACE plugin steps including the ‘Deploy’ step. This is the step we will use to deploy the broker archive (BAR) to an integration server running on an integration node.</p>	
11	<p>Click the pencil icon for the existing ‘Deploy’ step in the component process to examine the step details.</p>	
12	<p>Examine the inputs for the step.</p> <p>With IBM DevOps Deploy, you can pass in a hard-code string value for some of the inputs, however, it is most common to leverage properties.</p> <p>A property in IBM DevOps Deploy looks like `\${p:my-property}`. If a value could vary from one environment to the next, you can set the property at an environment scope. You can then reference the property with `\${p:environment/my-property}` in your step.</p>	 <div data-bbox="1204 893 1416 998"> <p>This property points to the ace home directory.</p> </div> <div data-bbox="1204 1036 1416 1110"> <p>This resolves to: /opt/IBM/ace-12.0.10.0/server</p> </div>
13	<p>Scroll down and observe some additional properties for the ACE Deploy step including ip-address, ace-port, and integration-server-name.</p>	

IBM TechXchange



Step	Details	Additional Information
14	<p>Check the box to ‘Clear Old Deployments’</p> <p>Notice the help pop-up dialog for each step you can hover your mouse over.</p> <p>The help tells us that when ‘Clear Old Deployments’ is enabled (or checked), this will run a complete deployment (as opposed to incremental).</p>	<p>Edit Properties for Deploy</p> <p>Deployed Objects</p> <p>Integration Server Name: \${p:environment/integration-server-name}</p> <p><input checked="" type="checkbox"/> Clear Old Deployments</p> <p>Trace File Name: \${p?:resource/ace.traceFileName}</p> <p>Timeout Value: \${p?:resource/ace.timeoutValue}</p>
15	Scroll down and click OK	
17	Notice that the next step in the process will ‘Run ACE RTAS Tests’. This will execute some automated tests in IBM DevOps Test Hub to verify the deployed BAR is working as expected.	 <pre> graph TD Start((Start)) --> DownloadArtifacts[Download Artifacts Download Artifacts (v. 46)] DownloadArtifacts --> Deploy[Deploy Deploy (v. 13)] Deploy --> RunTests[Run ACE RTAS Tests Run Application Process (v. 87)] RunTests --> AddStatusFail[Add Status to Version - Fail Add Status to Version (v. 74)] RunTests --> SetStatusSuccess[Set Status: Success] AddStatusFail --> SetStatusFailure[Set Status: Failure] SetStatusFailure --> Finish((Finish)) SetStatusSuccess --> Finish </pre>
18	<p>IBM DevOps Deploy has a governance feature that you can leverage called ‘gates’. The gates help to ensure that a component version meets entry criteria for a target environment. For example, “Unit Tests Passed” or Functional Tests Passed”.</p> <p>Here is an example of gates configured for an application:</p>	  <pre> graph TD Start((Start)) --> DownloadArtifacts[Download Artifacts Download Artifacts (v. 46)] DownloadArtifacts --> Deploy[Deploy Deploy (v. 13)] Deploy --> RunTests[Run ACE RTAS Tests Run Application Process (v. 87)] RunTests --> AddStatusFail[Add Status to Version - Fail Add Status to Version (v. 74)] RunTests --> SetStatusSuccess[Set Status: Success] AddStatusFail --> SetStatusFailure[Set Status: Failure] SetStatusFailure --> Finish((Finish)) SetStatusSuccess --> Finish </pre>

IBM TechXchange



Step	Details	Additional Information
	<p>For our process, we need to update the process so that if the functional test pass, we automatically set a status called 'Functional Tests Passed'. This is needed so we can promote this to the QA environment.</p> <p>In the search 'Type to filter' text field, type 'Add Status' as shown to the right</p>	
19	<p>Drag the 'Add Status to Version' step from the step palette and drop it on the process design.</p> <p>The best way to do this is to drag the step and hover over the line that connects the 'Run ACE RTAS Tests' step to 'Set Status: Success' step. The line will turn a blue color. (See the screen shot to the right). While the line is blue, release your mouse to drop the step onto the design.</p>	

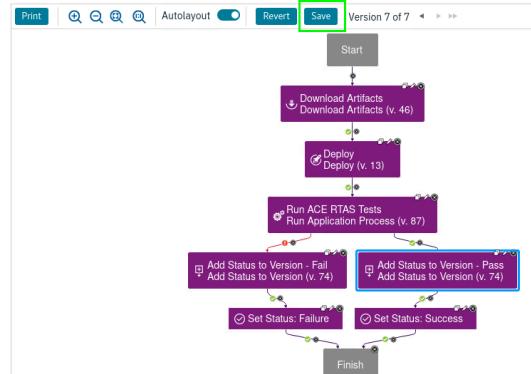
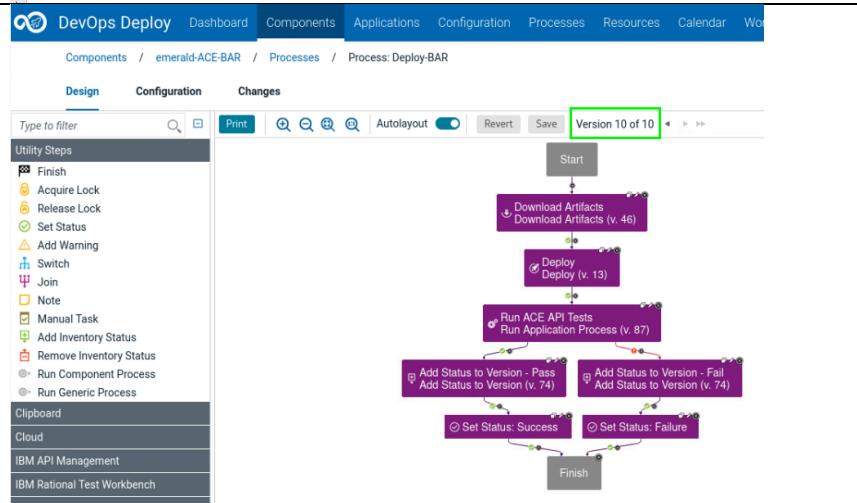
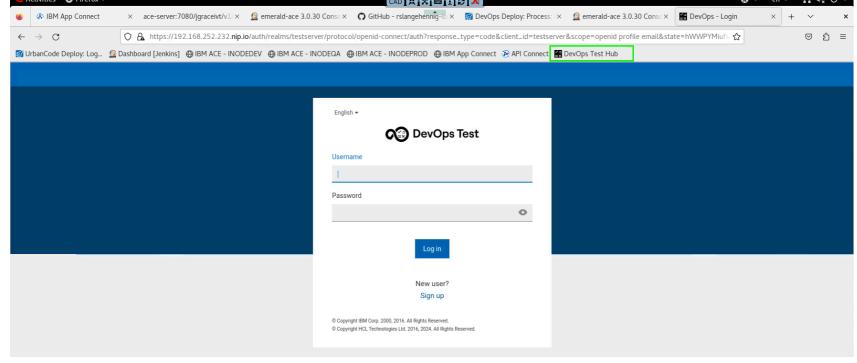
IBM TechXchange



Step	Details	Additional Information								
20	<p>Your process design should now look like this.</p> <p>Click the pencil icon for the new step you added to the process design</p>	<pre> graph TD Start([Start]) --> DA1[Download Artifacts Download Artifacts (v. 46)] DA1 --> Deploy[Deploy Deploy (v. 13)] Deploy --> RARTAS[Run ACE RTAS Tests Run Application Process (v. 87)] RARTAS --> ASV1[Add Status to Version - Fail] ASV1 --> SSF1[Set Status: Failure] SSF1 --> SSU1[Set Status: Success] SSU1 --> Finish([Finish]) </pre>								
21	<p>In the Edit Properties dialog that appears, provide the following initial values:</p> <p>Name: Add Status to Version - Pass Status: Functional Tests Passed</p> <p>NOTE: the status is case sensitive and it must match an existing Status that exists. Be sure the value you type is 'Functional Tests Passed' as shown to the right.</p>	<p>Edit Properties for Add Status to Version</p> <table border="1"> <tr> <td>Name *</td> <td>Add Status to Version - Pass</td> </tr> <tr> <td>Status</td> <td>Functional Tests Passed</td> </tr> <tr> <td>Working Directory</td> <td>[empty]</td> </tr> <tr> <td>Precondition</td> <td>[empty]</td> </tr> </table>	Name *	Add Status to Version - Pass	Status	Functional Tests Passed	Working Directory	[empty]	Precondition	[empty]
Name *	Add Status to Version - Pass									
Status	Functional Tests Passed									
Working Directory	[empty]									
Precondition	[empty]									
22	Scroll down and click OK	<p>Edit Properties for Add Status to Version</p> <table border="1"> <tr> <td>New</td> </tr> <tr> <td>Impersonation Impersonation can be configured here. Any steps which do not specify their own impersonation settings will fall back to the default impersonation settings provided by the resource or agent.</td> </tr> <tr> <td><input type="checkbox"/> Use Impersonation</td> </tr> <tr> <td>Auth Token Restriction System Default</td> </tr> <tr> <td>New Edit</td> </tr> <tr> <td><input type="checkbox"/> Show Hidden Properties</td> </tr> <tr> <td>Cancel OK</td> </tr> </table>	New	Impersonation Impersonation can be configured here. Any steps which do not specify their own impersonation settings will fall back to the default impersonation settings provided by the resource or agent.	<input type="checkbox"/> Use Impersonation	Auth Token Restriction System Default	New Edit	<input type="checkbox"/> Show Hidden Properties	Cancel OK	
New										
Impersonation Impersonation can be configured here. Any steps which do not specify their own impersonation settings will fall back to the default impersonation settings provided by the resource or agent.										
<input type="checkbox"/> Use Impersonation										
Auth Token Restriction System Default										
New Edit										
<input type="checkbox"/> Show Hidden Properties										
Cancel OK										

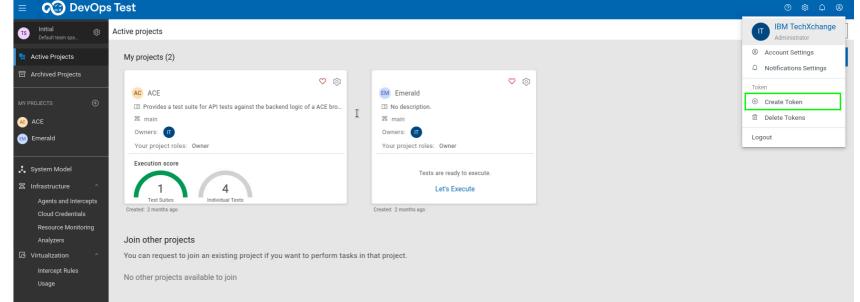
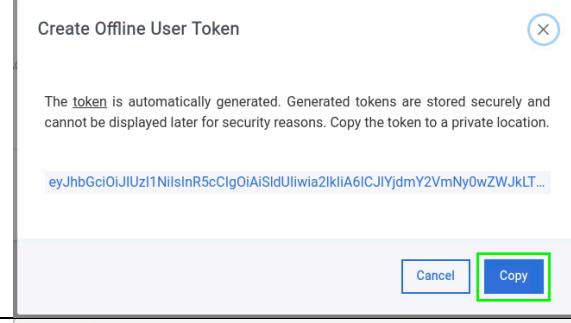
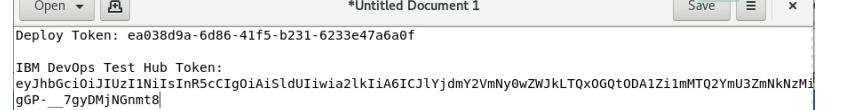
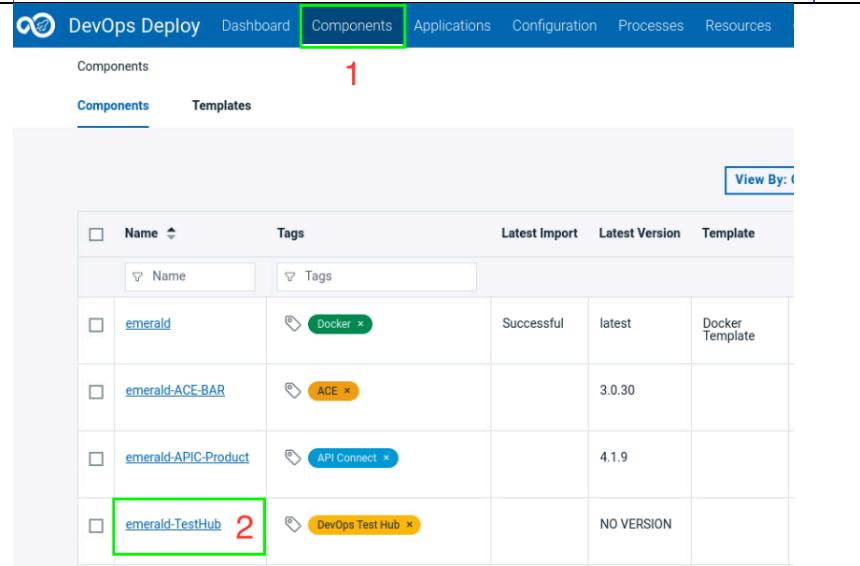
IBM TechXchange



Step	Details	Additional Information
23	Click the Save button to save the component process	 <pre> graph TD Start((Start)) --> DownloadArtifacts1[Download Artifacts
Download Artifacts (v. 46)] DownloadArtifacts1 --> Deploy1[Deploy
Deploy (v. 13)] Deploy1 --> RunACERTASTests[Run ACE RTAS Tests
Run Application Process (v. 67)] RunACERTASTests --> AddStatus1[Add Status to Version - Fail
Add Status to Version (v. 74)] RunACERTASTests --> AddStatus2[Add Status to Version - Pass
Add Status to Version (v. 74)] AddStatus1 --> SetStatusFailure1[Set Status: Failure] AddStatus2 --> SetStatusSuccess1[Set Status: Success] SetStatusFailure1 --> Finish((Finish)) SetStatusSuccess1 --> Finish </pre>
24	<p>You should notice that the version of the process is now 'Version 10 of 10' as highlighted to the right.</p> <p>DevOps Deploy has built-in versioning capabilities in the tool you can leverage. This process can be exported in a JSON format and checked into a git repository.</p>	
25	<p>Before we complete this section of the lab, we need to generate a new API Token in IBM DevOps Test Hub that will be used by our processes to test what we deploy to the target environments. The API Token is needed to authenticate with the IBM DevOps Test Hub application.</p> <p>In your browser, create a new tab and then click the 'DevOps Test Hub' bookmark found the toolbar.</p>	
26	Login to IBM DevOps Test Hub as the user techxchange with password of password	

IBM TechXchange



Step	Details	Additional Information
27	Right click on the user profile at the top-right of the screen as shown to the right (step 1).	
28	Click the ' Create Token ' menu item as highlighted above (step 2).	
29	<p>A new API Token will appear.</p> <p>Click the 'Copy' button to copy the token to your buffer.</p>	
30	Open your 'Text Editor' window and paste the API Token for future reference as shown in the example to the right.	
31	<p>Return to your DevOps Deploy tab in your browser.</p> <p>Click on Components at the top of the screen (step 1).</p>	
32	Click on the 'emerald-TestHub' component (step 2) as shown above.	

IBM TechXchange



Step	Details	Additional Information
33	Click on the Processes sub-tab	
34	Click on the 'Run Automated Test for ACE API' process.	
35	<p>The visual process design will appear for the process.</p> <p>Click the pencil icon for the 'Run IBM Rational Test Automation Server test' step as shown to the right.</p>	
36	Paste the new API Key from your Text Editor window into the ' Offline Token ' field as shown to the right.	

IBM TechXchange



Step	Details	Additional Information
37	Scroll down and click OK.	<p>Edit Properties for Run IBM Rational Test Automation Server test</p>
38	<p>In the Visual Process Designer, click 'Save' to save the changes to the process.</p>	<p>Components / emerald-RTAS / Processes / Process: Run Automated Test for ACE API</p>
39	<p>We need to repeat these steps for one more process.</p> <p>Click on 'Processes' next to 'emerald-TestHub' in the crumb trail as shown to the right</p>	<p>Components / emerald-TestHub / Processes / Process: Run Automated Test for ACE API</p>

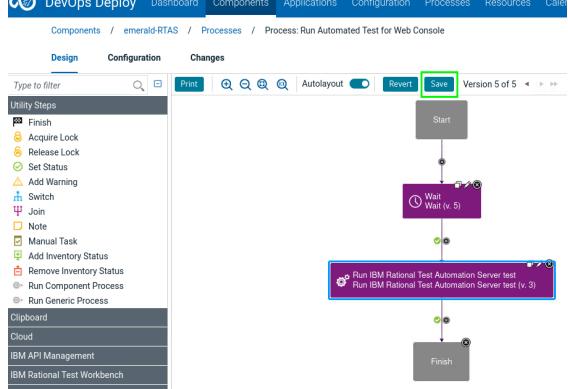
IBM TechXchange



Step	Details	Additional Information
40	Click on the ' Run Automated Test for Web Console ' process	<p>Components / emerald-TestHub</p> <p>Component: emerald-TestHub</p> <p>Created By admin Created On 11/28/2023, 5:08 PM</p> <p>Description This component will run tests for Emerald eStore</p> <p>Used By Emerald eStore View All</p> <p>Versions History Usage Configuration Calendar Processes Changes</p> <p>Process</p> <p>Run Automated Test for ACE API</p> <p>Run Automated Test for Web Console</p> <p>Items per page: 10 1-2 of 2 items</p>
41	<p>The Visual Process Designer will appear for the process.</p> <p>Click the pencil icon on the step 'Run IBM Rational Test Automation Server test' as shown to the right.</p>	<p>Components / emerald-RTAS / Processes / Process: Run Automated Test for Web Console</p> <p>Design Configuration Changes</p> <p>Type to filter</p> <p>Utility Steps</p> <ul style="list-style-type: none"> <input checked="" type="checkbox"/> Finish <input checked="" type="checkbox"/> Acquire Lock <input checked="" type="checkbox"/> Release Lock <input checked="" type="checkbox"/> Set Status <input checked="" type="checkbox"/> Add Warning <input checked="" type="checkbox"/> Switch <input checked="" type="checkbox"/> Join <input checked="" type="checkbox"/> Note <input checked="" type="checkbox"/> Manual Task <input checked="" type="checkbox"/> Add Inventory Status <input checked="" type="checkbox"/> Remove Inventory Status <input checked="" type="checkbox"/> Run Component Process <input checked="" type="checkbox"/> Run Generic Process <p>Clipboard Cloud IBM API Management IBM Rational Test Workbench</p> <pre> graph TD Start((Start)) --> Wait((Wait)) Wait --> Run[Run IBM Rational Test Automation Server test] Run --> Finish((Finish)) </pre>
42	Paste the API Key from your Text Editor into the ' Offline Token ' field as shown to the right.	<p>Edit Properties for Run IBM Rational Test Automation Server test</p> <p>Name * Run IBM Rational Test Automation Server test</p> <p>IBM Rational Test Automation Server URL * https://192.168.252.232.nip.io/</p> <p>Offline Token * <input type="text" value="****"/></p> <p>Team Space Name * Initial Team Space</p> <p>Project Name * Emerald</p> <p>Branch Name main</p> <p>Repository Link *</p>
43	Scroll down and click the OK button	<p>Edit Properties for Run IBM Rational Test Automation Server test</p> <p>New</p> <p>Implementation</p> <p>Impersonation can be configured here. Any steps which do not specify their own impersonation settings will fall back to the default impersonation settings provided by the resource or agent.</p> <p><input type="checkbox"/> Use Impersonation</p> <p>Auth Token Restriction</p> <p>System Default</p> <p>New Edit</p> <p>Show Hidden Properties</p> <p>Cancel OK</p>



IBM TechXchange

Step	Details	Additional Information
44	In the Visual Process Designer view, click the ' Save ' button to save your changes to the process.	
45	<p>You have completed this lab section!</p> <p>In the next section, we will deploy our new BAR file to the DEV environment and an automated test will be executed in IBM DevOps Test Hub.</p>	

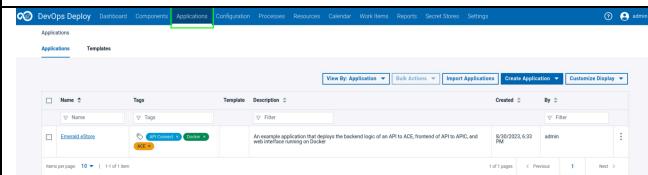
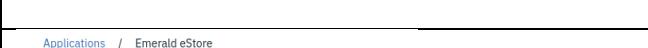
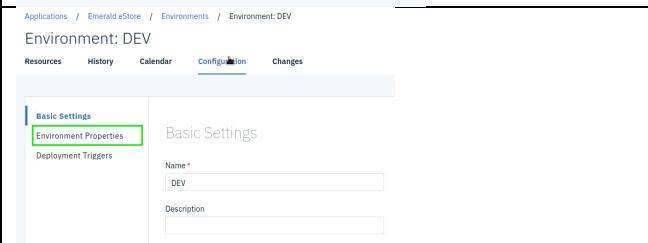
End of Section - 3.4

3.5 Deploy the ACE Broker Archive using IBM DevOps Deploy to DEV

3.5.1 Overview

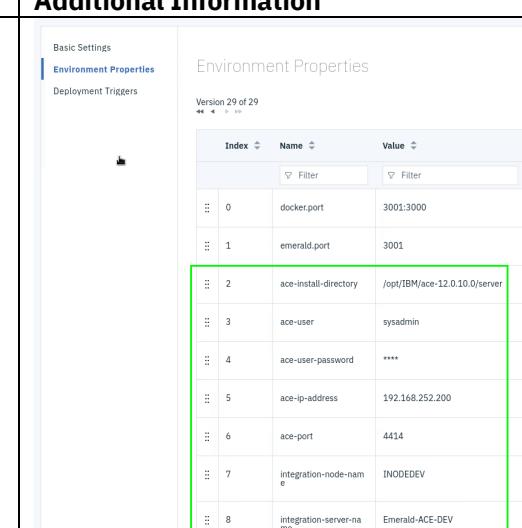
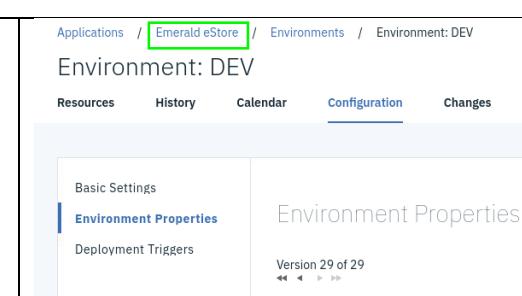
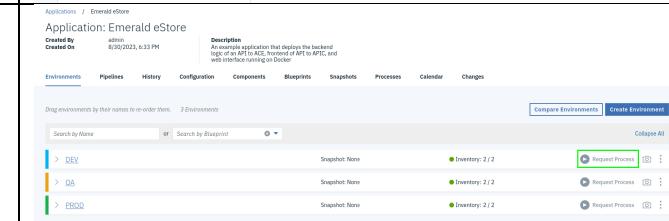
In this lab section, we will deploy the built ACE broker archive file to the DEV environment and invoke an automated API test using Rational Test Automation Server (RTAS).

3.5.2 Workbook steps

Step	Details	Additional Information
1	Return to your IBM DevOps Deploy web console in your browser.	
2	Click on Applications at the top of the page	
3	Click on the ‘Emerald eStore’ application in the list	
4	Before we initiate the deployment, we will review the environment properties that are set for the DEV environment. Click on the “DEV” text for the DEV environment as shown to the right.	
5	In the ‘Environment: DEV’ page, click on the ‘Configuration’ sub-tab	
6	Click on ‘Environment Properties’ in the left-hand pane	

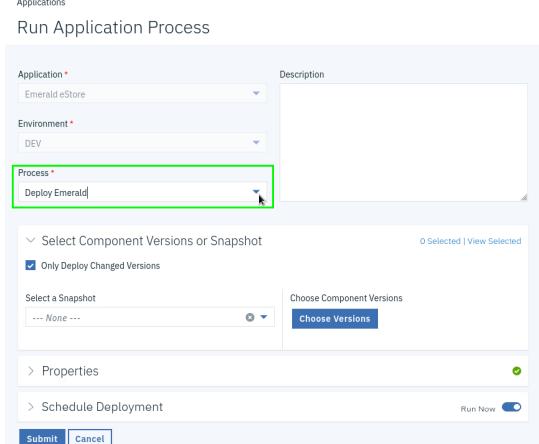
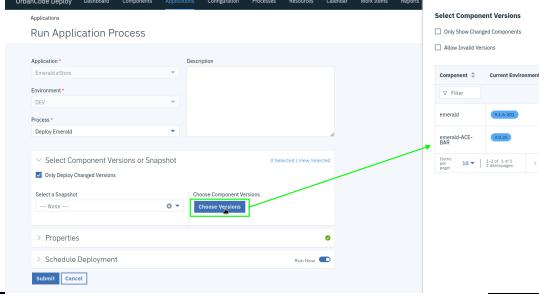
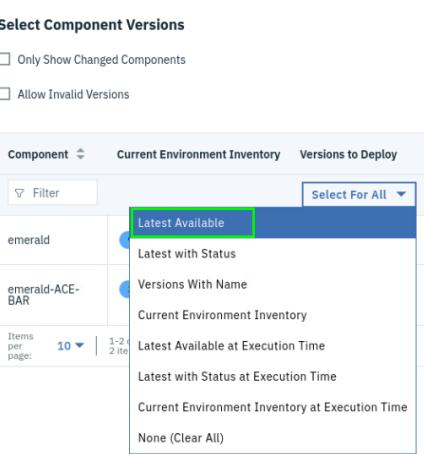
IBM TechXchange



Step	Details	Additional Information																														
7	<p>Here you will see the property values that are used in our automated deployment processes.</p> <p>Notice the ACE specific properties that you saw referenced in the component process we reviewed in the previous lab.</p>	 <table border="1"> <thead> <tr> <th>Index</th> <th>Name</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>docker.port</td> <td>3001:3000</td> </tr> <tr> <td>1</td> <td>emerald.port</td> <td>3001</td> </tr> <tr> <td>2</td> <td>ace-install-directory</td> <td>/opt/IBM/ace-12.0.10.0/server</td> </tr> <tr> <td>3</td> <td>ace-user</td> <td>sysadmin</td> </tr> <tr> <td>4</td> <td>ace-user-password</td> <td>****</td> </tr> <tr> <td>5</td> <td>ace-ip-address</td> <td>192.168.252.200</td> </tr> <tr> <td>6</td> <td>ace-port</td> <td>4414</td> </tr> <tr> <td>7</td> <td>integration-node-name</td> <td>INODEDEV</td> </tr> <tr> <td>8</td> <td>integration-server-name</td> <td>Emerald-ACE-DEV</td> </tr> </tbody> </table>	Index	Name	Value	0	docker.port	3001:3000	1	emerald.port	3001	2	ace-install-directory	/opt/IBM/ace-12.0.10.0/server	3	ace-user	sysadmin	4	ace-user-password	****	5	ace-ip-address	192.168.252.200	6	ace-port	4414	7	integration-node-name	INODEDEV	8	integration-server-name	Emerald-ACE-DEV
Index	Name	Value																														
0	docker.port	3001:3000																														
1	emerald.port	3001																														
2	ace-install-directory	/opt/IBM/ace-12.0.10.0/server																														
3	ace-user	sysadmin																														
4	ace-user-password	****																														
5	ace-ip-address	192.168.252.200																														
6	ace-port	4414																														
7	integration-node-name	INODEDEV																														
8	integration-server-name	Emerald-ACE-DEV																														
INFO	<p>If the way you deploy one BAR file is the same way you will deploy the other 10 BAR files you typically deploy, you can setup a reusable component template to handle these deploys. This provides two benefits:</p> <ul style="list-style-type: none"> • You can quickly on-board other ACE components based on the template • You have one place to make updates to the ACE deployment process (helping with maintenance) <p>The use of properties, rather than hard-coded string values, is vital to the creation of reusable templates. Properties can be set at the environment, component, application, and resource scope in IBM DevOps Deploy.</p>																															
8	<p>Let's initiate our deployment.</p> <p>Click on “Emerald eStore” in the crumb trail at the top of the page.</p>																															
9	<p>Click the “Request Process” button for your DEV environment as shown to the right</p>																															

IBM TechXchange



Step	Details	Additional Information
10	For the ‘Run Application Process’ page that appears, select the Process labeled ‘Deploy Emerald’ from the pulldown menu.	
11	<p>Click the ‘Choose Versions’ button to choose component versions.</p> <p>A ‘Select Component Versions’ dialog will appear to the right.</p>	
12	In the Select Component Versions dialog that appears to the right, click the ‘Select For All’ pulldown and choose ‘Latest Available’ as shown to the right	

IBM TechXchange



Step	Details	Additional Information
13	You should see that the values in the 'Versions to Deploy' column look different from what is in the 'Current Environment Inventory' column.	<p>Select Component Versions</p> <p><input type="checkbox"/> Only Show Changed Components</p> <p><input type="checkbox"/> Allow Invalid Versions</p>
14	Back in the 'Run Application Process' page, notice that it shows that there are '2 Selected' component versions for deployment.	<p>Applications</p> <p>Run Application Process</p>
15	Click the ' Submit ' button to initiate the deployment (as shown above)	NOTE: you can schedule the deployment for a specific day and time as well if you like.
INFO	NOTE: We are running the deployment through the web console, however, keep in mind that this deployment could be initiated automatically following a build process. You can make REST or CLI calls to start deployments automatically from your Jenkins or GitLab pipeline.	
16	<p>The deployment will start, and you will have a nice trail of audit for each deployment as you see to the right.</p> <p>You can click 'Expand All' or manually expand the deployment for the BAR file.</p>	
17	<p>Click the layer cake (dots) pulldown menu to the right of the 'Deploy' step as shown to the right.</p> <p>Then click the 'View Output Log' menu item for the pulldown menu</p>	

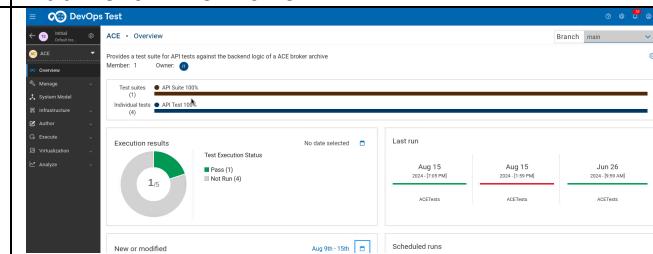
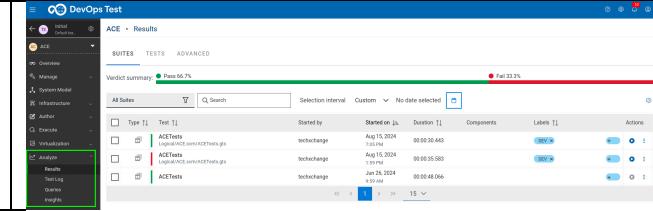
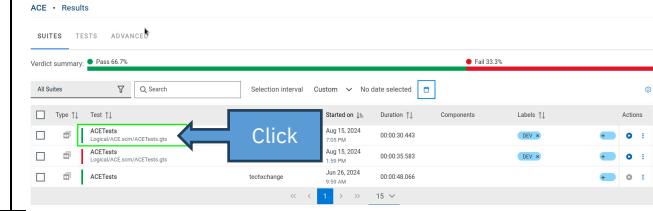
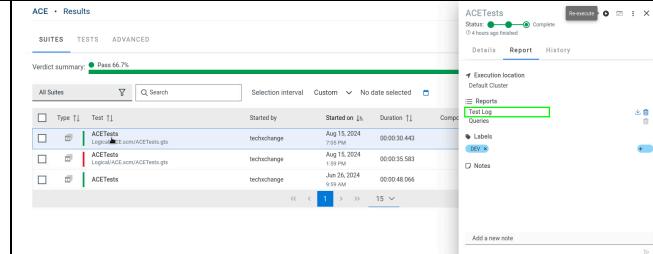
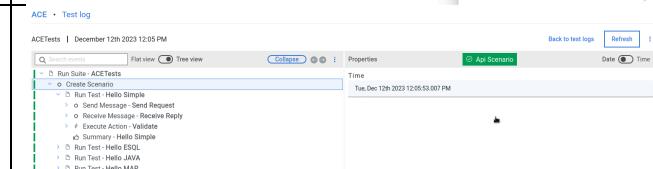
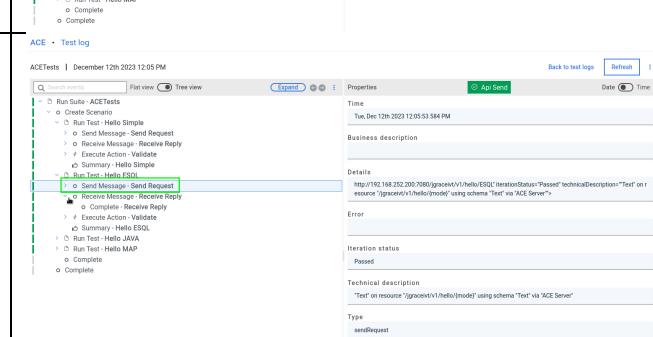
IBM TechXchange



Step	Details	Additional Information
18	<p>You can view the Output Log for any of the steps that were executed.</p> <p>Here we see that the deployment of the BAR file was successful.</p>	
19	Click the 'Close' button to close the Output Log dialog	
20	Notice that following the deployment that we are incorporating automated test execution to validate the ACE APIs using IBM DevOps Test Hub to run an automated API test suite.	
INFO	IBM DevOps Test Hub can run many different types of tests including: <ul style="list-style-type: none"> • API Tests • Performance Tests • Functional Tests 	
21	<p>In your browser, return to the IBM DevOps Test Hub tab.</p> <p>If you need to create a new tab, click the 'DevOps Test Hub' bookmark found the toolbar as shown to the right.</p>	
22	Login to IBM DevOps Test Hub as the user techxchange with password of password	
23	You will see your Active Projects in the view. Click on the ACE project as highlighted to the right.	

IBM TechXchange



Step	Details	Additional Information
24	In the ACE Overview page that appears, you can see the latest execution results.	
25	In the left-hand menu, click on Analyze to expand the menu. Click on “Results” in the pulldown menu. This should show you all the Results from your tests as shown to the right.	
26	You can drill into the test logs for a given test run. Click on our latest test (click on the “ACETests” text) that was run against the DEV environment.	
27	A dialog will appear to the right. Click on the ‘Test Log’ link	
28	The Test Log will appear. Expand the ‘Create Scenario’ node in the left pane to see each test that was executed.	
29	Expand the ‘Run Test – Hello ESQL’ test and examine the Send Message, Receive Message, and more to see the results of the test.	

IBM TechXchange



Step	Details	Additional Information
30	<p>Return to the IBM DevOps Deploy tab in your browser.</p> <p>The deployment of both the ACE BAR and the Emerald web application should succeed as shown to the right.</p>	
31	Click on ‘Emerald eStore’ in the crumb trail as shown to the right	
32	Expand the DEV environment to see that the inventory has been updated to reflect the component versions that you deployed.	
33	Click on the ‘emerald-ACE-BAR’ component	
34	<p>Notice that version 3.0.30 that we just deployed and ran automated tests against now has a ‘Functional Tests Passed’ status added to it.</p> <p>In IBM DevOps Deploy, you can use these status values to ensure Gate criteria are met before a deployment occurs to the next environment.</p> <p>This helps to ensure good quality code is progressing through your pipeline using IBM DevOps Deploy.</p>	
35	<p>Let’s go check what was deployed.</p> <p>In your browser, create a new tab. Click the “IBM ACE – INODEDEV” bookmark to open this page.</p>	

IBM TechXchange



Step	Details	Additional Information
36	Click on “Emerald-ACE-DEV” or select “Open” from the layer cake pulldown menu item to see that the JGRACEIVT API has been deployed to this Integration Server.	
37	If you open the JGRACEIVT API, you can see the REST APIs that are available for it as shown to the right. NOTE: Our DevOps Test Hub automated test exercised these APIs to ensure they were working properly.	
38	In your browser window, view the Emerald web application running in the DEV environment by opening a new tab and navigating to: http://192.168.252.201:3001 You should see the application as shown to the right.	
39	You have successfully deployed the ACE BAR to the DEV environment along with the Emerald web application! We will promote our updates to the QA environment in the next lab.	

End of Section – 3.5

3.6 Promote the Emerald Components to the QA Environment

3.6.1 Overview

In this lab section, we will deploy the built ACE broker archive file to the QA environment and invoke an automated API test using Rational Test Automation Server (RTAS). We will review the results.

3.6.2 Workbook steps

Step	Details	Additional Information
1	In your browser, return to the IBM DevOps Deploy tab.	

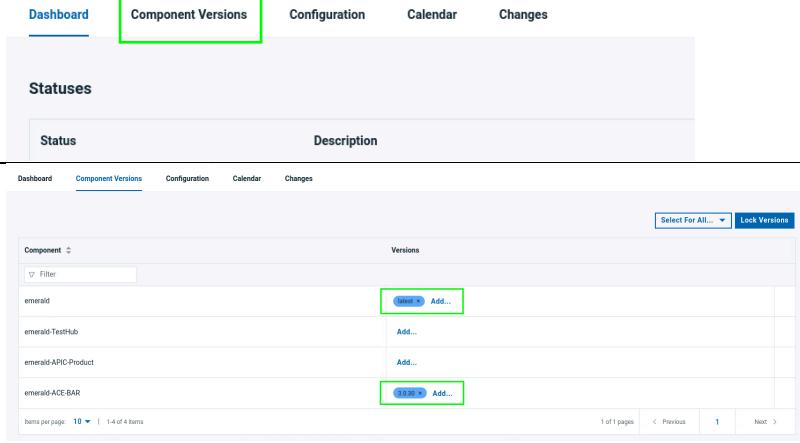
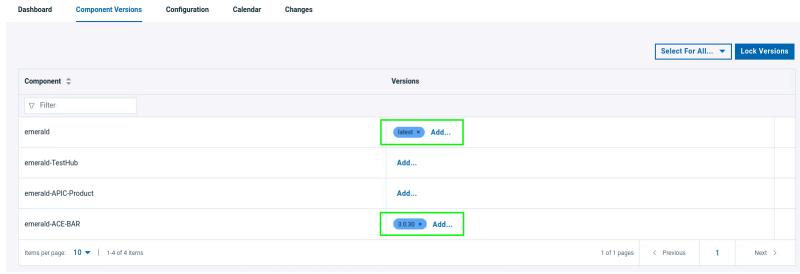
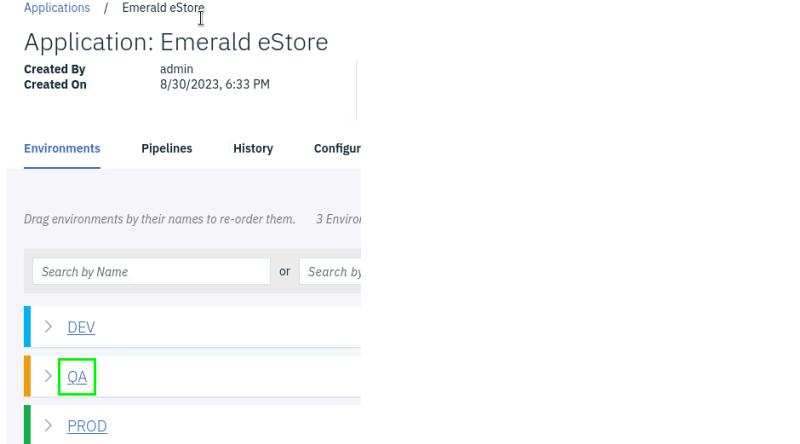
IBM TechXchange



Step	Details	Additional Information
2	Click on the ‘Applications’ tab at the top of the page and click on the ‘Emerald eStore’ application as shown to the right.	<p>The screenshot shows the 'Applications' tab selected in the navigation bar. Below it, a table lists applications, with 'Emerald eStore' highlighted by a green box and the number '2' next to it. A tooltip indicates it's an example application for API to ACE, API to APC, and web interface.</p>
3	Expand the DEV environment to show what is currently deployed to the DEV environment.	<p>The screenshot shows the 'Environments' tab selected. Under the 'DEV' environment, two components are listed: 'emeraldACE-BAR' and 'emerald'. The 'emeraldACE-BAR' row has a '3.0.0' link highlighted with a green box and the number '1' next to it.</p>
4	<p>One of the nice features of IBM DevOps Deploy is that you can create a Snapshot of an environment.</p> <p>This allows you to easily promote what you have tested together (the component versions for the ACE and WEB components) to the next environment.</p> <p>Click on the ‘camera’ icon for the DEV environment as shown to the right to create a Snapshot.</p>	<p>The screenshot shows the 'Environments' tab selected. Under the 'DEV' environment, two components are listed: 'emeraldACE-BAR' and 'emerald'. The 'emeraldACE-BAR' row has a '3.0.0' link highlighted with a green box and the number '1' next to it. A blue arrow points to the camera icon in the top right corner of the environment section.</p>
5	<p>Enter the following values in the ‘Create Snapshot’ dialog that appears:</p> <p>Name: 2024-Nov-01</p> <p>You may optionally add a description similar to what is shown in the screen capture to the right.</p>	<p>The screenshot shows the 'Create Snapshot' dialog box. It has fields for 'Name' (2024-Nov-01) and 'Description' (Snapshot for promotion of artifacts with Nov 2024 release). There are 'Cancel' and 'Save' buttons at the bottom.</p>
6	Click the ‘Save’ button	
	The snapshot page will appear.	

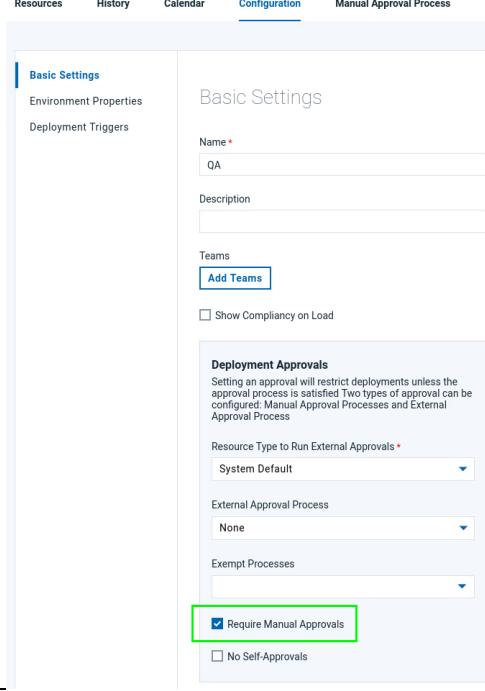
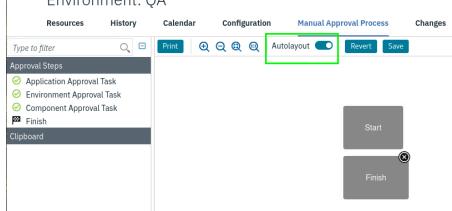
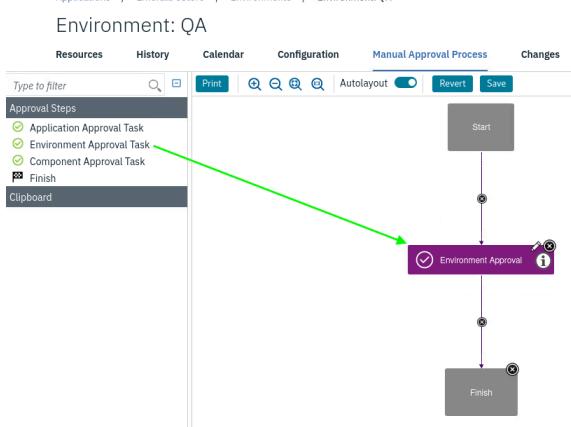
IBM TechXchange



Step	Details	Additional Information
7	Click on the 'Component Versions' sub-tab as shown to the right.	<p>Snapshot: 2024-Nov-01</p> <p>Versions Locked: No Configuration Locked: No</p> <p>Description: Snapshot for promotion of artifacts with Nov 2024 release</p> 
8	<p>Notice that the current inventory of the DEV environment was added as Component Versions for the snapshot.</p> <p>NOTE: you can press the 'Lock Versions' button to ensure this snapshot cannot be modified.</p>	
9	In the crumb trail at the top, click on 'Emerald eStore'	<p>Snapshot: 2024-Nov-01</p> <p>Versions Locked: Yes Configuration Locked: No</p> <p>Description: Snapshot for promotion of artifacts with Nov 2024 release</p> 
10	<p>Let's examine our QA environment to see that we have setup an approval process before a deployment can take place.</p> <p>Click on the text 'QA' for the QA environment.</p>	
11	Click on the 'Configuration' sub-tab as shown to the right.	

IBM TechXchange



Step	Details	Additional Information
12	On the Basic Settings page that appears, scroll down to the 'Deployment Approvals' section and notice that the 'Require Manual Approvals' is enabled (or checked) as shown to the right.	<p>Applications / Emerald eStore / Environments / Environment: QA</p> <p>Environment: QA</p> 
13	Click on the ' Manual Approval Process ' sub-tab that appears	<p>Applications / Emerald eStore / Environments / Environment: QA</p> <p>Environment: QA</p> 
14	In the process design that appears, slide the 'Autolayout' slider to the on position.	<p>Applications / Emerald eStore / Environments / Environment: QA</p> <p>Environment: QA</p> 
15	<p>Drag and drop the 'Environment Approval Task' step in the left-hand palette to the process design.</p> <p>NOTE: Using the Autolayout feature, the process design will automatically connect the Start and Finish steps for you.</p>	<p>Applications / Emerald eStore / Environments / Environment: QA</p> <p>Environment: QA</p> 

IBM TechXchange



Step	Details	Additional Information
16	Click the pencil icon for the ‘Environment Approval’ step to edit it	
17	The properties of the step will be displayed. Take the defaults as shown and scroll down and click OK.	
18	Click ‘Save’ to save your approval process.	
19	Click on ‘Emerald eStore’ in the crumb-trail at the top to return to the application.	
20	In IBM DevOps Deploy, you can compare inventory of one environment to another. Click the layer-cake icon next to the DEV environment and then click ‘Compare’ from the pull-down menu as shown to the right.	
22	In the ‘Compare Environments’ dialog that appears, choose the QA environment from the pull-down menu.	
23	Click ‘Compare’	

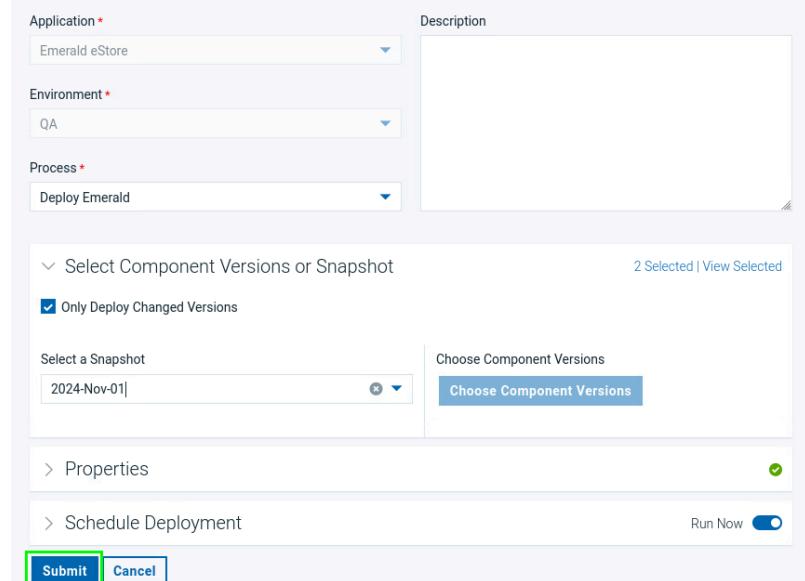
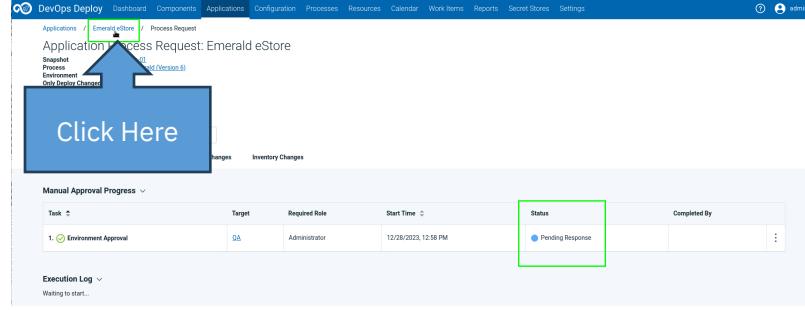
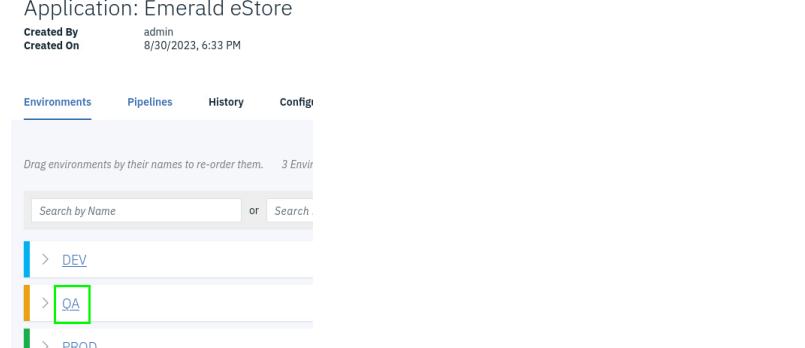
IBM TechXchange



Step	Details	Additional Information
24	An Environment Comparison page will be displayed so you can see the inventory differences between the DEV and QA environments.	<p>The screenshot shows the 'Inventory Comparison' section of the Environment Comparison page. It lists components and their versions across two environments. The components include 'emerald', 'emeraldaACE-BAR', and 'emeraldaACE-BAR'. The 'emerald' component has four versions: 'latest', '3.0.24', '3.0.30', and '9.1.6.201'. The 'emeraldaACE-BAR' component has three versions: '3.0.24', '3.0.30', and '9.1.6.201'. The status column indicates that all components are active. The DEV environment has green checkmarks in the status column, while the QA environment has yellow checkmarks.</p>
25	<p>Let's deploy the snapshot to the QA environment.</p> <p>Click on 'Emerald eStore' in the crumb-trail as you have before to return the main page for the application.</p>	<p>Applications / Emerald eStore / Environments</p> <p>Environment Comparison</p> <p>Environment 1 DEV Environment 2 QA</p>
26	Click the 'Request Process' button next to the QA environment	<p>The screenshot shows the 'Application: Emerald eStore' overview page. It includes details like 'Created By: admin' and 'Created On: 8/30/2023, 6:33 PM'. Below this is a navigation bar with tabs: Environments, Pipelines, History, Configuration, Components, Blueprints, Snapshots, Processes, Calendar, and Changes. Under the Environments tab, there are three environments listed: DEV, QA, and PROD. Each environment has a 'Request Process' button highlighted with a green box.</p>
27	In the 'Run Application Process' page that appears, select 'Deploy Emerald' from the Process pulldown menu.	<p>Run Application Process</p> <p>The screenshot shows the 'Run Application Process' page. It has fields for 'Application' (set to 'Emerald eStore'), 'Environment' (set to 'QA'), and 'Process' (dropdown menu showing 'Deploy Emerald' highlighted with a green box). Below these are sections for selecting component versions and snapshots, with a checkbox for 'Only Deploy Changed Versions'.</p>
28	In the 'Select a Snapshot' pulldown, select '2024-Nov-01' as shown to the right.	<p>Run Application Process</p> <p>The screenshot shows the 'Run Application Process' page again. The 'Process' field is now set to 'Deploy Emerald'. The 'Select a Snapshot' dropdown at the bottom is highlighted with a green box and shows '2024-Nov-01' selected. A 'Choose Component Versions' button is also visible.</p>

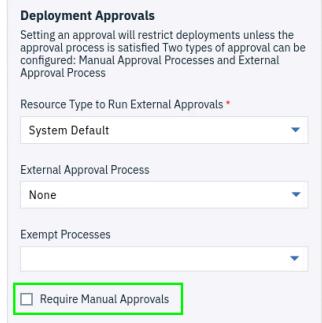
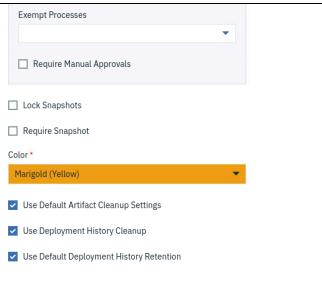
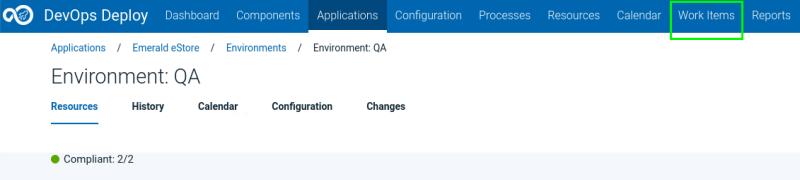
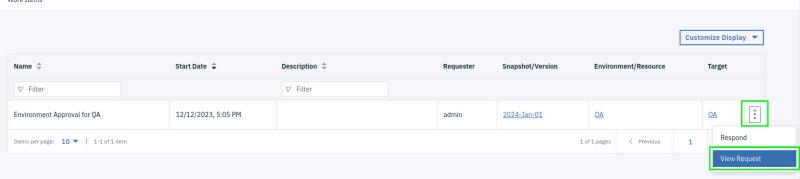
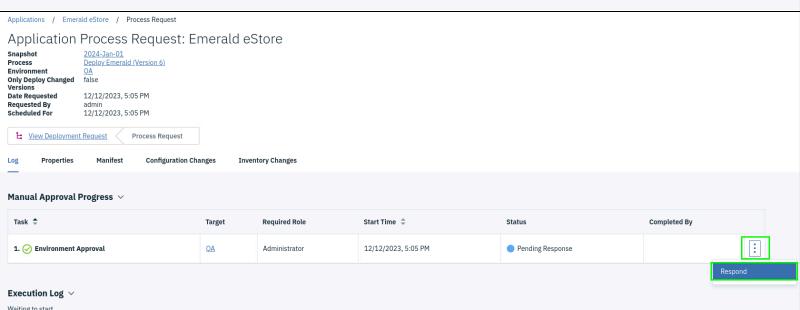
IBM TechXchange



Step	Details	Additional Information
29	Scroll down and click the ' Submit ' button to initiate the deployment.	
30	<p>Notice that the deployment is waiting for an approval before it can begin.</p> <p>Before we respond to this approval request, we are going to disable all future approval requirements just for this lab.</p> <p>Click on 'Emerald eStore' in the crumb-trail as you have before</p>	
31	Click on the QA environment text	

IBM TechXchange



Step	Details	Additional Information
32	Click on the ‘Configuration’ sub-tab and scroll down to uncheck the ‘Require Manual Approvals’ checkbox as shown to the right.	
33	Scroll down and click ‘Save’	
34	Let’s return to our deployment. Click on the ‘Work Items’ tab at the top of the page.	
35	Click the layer cake icon next to the QA approval request and select ‘View Request’ as shown to the right. This will take us back to our deployment request.	
36	After reviewing the deployment, click the layer-cake icon to the right and select ‘Respond’ from the pulldown menu.	

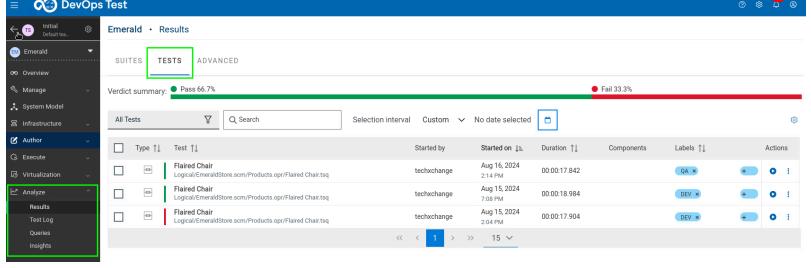
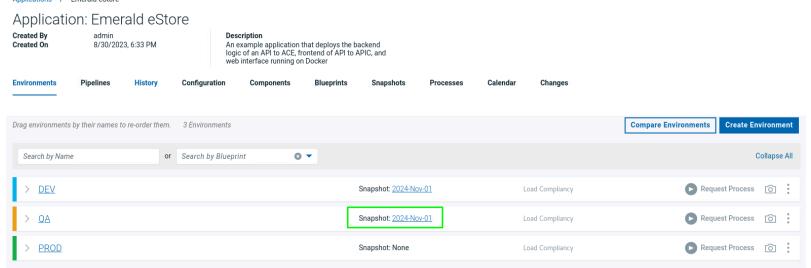
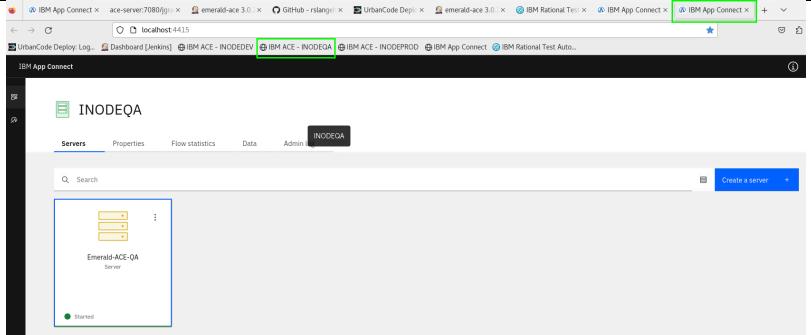
IBM TechXchange



Step	Details	Additional Information
37	In the Respond dialog that appears, select 'Approve' from the response choice menu item and add an optional comment. Click Submit	
38	The deployment process will begin to the QA environment and you can view the progress of the deployment.	
39	The deployment of the ACE BAR should succeed to QA just as it did earlier for the DEV environment. The automated IBM DevOps Test Hub functional test will run against the API in the QA environment.	
40	Likewise, the deployment of the emerald WEB component should succeed as well. NOTE: we are running functional tests against the web component as well using IBM DevOps Test Hub.	
41	Optionally, review the test results in IBM DevOps Test Hub by switching to that tab in your browser. For the ACE project, click on 'Analyze' in the left-hand pane and then click on 'Results'.	

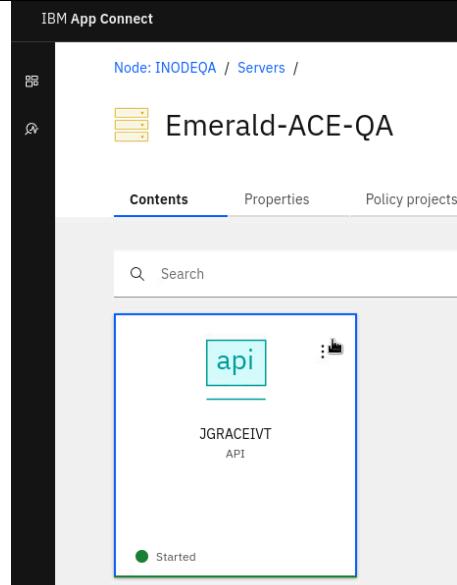
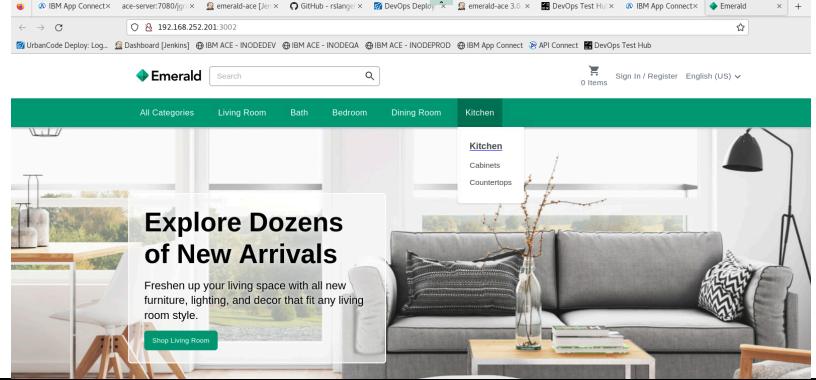
IBM TechXchange



Step	Details	Additional Information
42	<p>Optionally, review the test results in IBM DevOps Test Hub for the ‘Emerald’ project.</p> <p>For the Emerald project, click on ‘Analyze’ in the left-hand pane and then click on ‘Results’.</p> <p>In the right hand Results view, click on the ‘Tests’ sub-tab as shown to the right.</p>	
43	In your IBM DevOps Deploy tab, return to the ‘Emerald eStore’ application view.	
44	Notice that inventory is updated for the QA environment to show that the snapshot has been deployed	
45	In your browser, create a new tab and click the ‘IBM ACE – INODEQA’ bookmark	

IBM TechXchange



Step	Details	Additional Information
46	Open the Emerald-ACE-QA integration server to see that the JGRACEIVT API is deployed as expected.	
47	In your browser window, view the Emerald web application by opening a new tab and navigating to: http://192.168.252.201:3002/ You should see the application as shown to the right.	
48	Congratulations! You have successfully promoted the BAR file as well as the web component.	
End of Section – 3.6		

3.7 Publishing a Product to API Connect using IBM DevOps Deploy

3.7.1 Overview

IBM API Connect is a full lifecycle API management solution that uses you create, manage, secure, socialize, and monetize APIs. In this final lab, you will publish a product, which includes APIs you want to socialize to your end users in a secure manner. The product will be published to a ‘sandbox’ catalog that is already setup and ready for you to use in a DEV environment.

NOTE: this Lab is leveraging an API Connect instance that is running in the IBM Cloud. Each student in the lab will be provided a unique login and password to authenticate with this API Connect instance.



3.7.2 Workbook steps

Step	Details	Additional Information
1	<p>For this lab, each student will be assigned a student number which will provide you with a unique login credential as well as an organization to use for API Connect.</p> <p>Open or return to a terminal window and run this command:</p> <pre># cd /home/sysadmin/scripts # ls -al</pre> <p>You should see a 'student-passwords.txt' file as shown to the right.</p>	<pre>sysadmin@rhserve:~/scripts File Edit View Search Terminal Help [sysadmin@rhserve scripts]\$ cd /home/sysadmin/scripts [sysadmin@rhserve scripts]\$ ls -al total 1372 drwxrwxr-x. 3 sysadmin sysadmin 4096 Dec 29 12:57 . drwxr-----. 20 sysadmin sysadmin 4096 Dec 29 16:21 .. -rw-r--r--. 1 root root 1371733 Dec 29 16:26 agent.jar drwxrwxr-x. 2 sysadmin sysadmin 74 Dec 29 12:59 resources -rwxrwxr-x. 1 sysadmin sysadmin 213 Nov 28 22:45 start_ace.sh -rwxr-xr-x. 1 root root 251 Nov 29 17:34 start-jenkins-agent.sh -rwxrwxr-x. 1 sysadmin sysadmin 40 Nov 28 22:45 start-ucd-agent.sh -rw-r--r--. 1 root root 561 Dec 15 01:03 student-passwords.txt -rwxr-xr-x. 1 sysadmin sysadmin 858 Dec 29 12:57 test-apic-deploy.sh -rwxr-xr-x. 1 sysadmin sysadmin 265 Dec 29 12:52 test-apic-login.sh [sysadmin@rhserve scripts]\$</pre>
2	<p>Run this command:</p> <pre># cat student-passwords.txt</pre> <p>NOTE: your login will be 'studentxx' based on your number. For this example, I will be using 'student20'.</p>	<pre>sysadmin@rhserve:~/scripts File Edit View Search Terminal Help [sysadmin@rhserve scripts]\$ cat student-passwords.txt 01 - XjQgrFmzqXoWMe7Rj4KB 02 - BMEK9PsnXcEBC4sJ7Uj 03 - eKTf2b6CYrUwtGNHx73ua 04 - n9PAQn7jeab03EGwT2Z5 05 - Tr549k6g8bRkVjVgV73xH 06 - ZNz3Q9rD2wV3JyTeSuam 07 - qM9tL9R9vD9s9uB9D9 08 - Jxp3tTyntD29s9uScvWa 09 - dtuQVq9X4anzeqaP6q0 10 - YaQKNQ27bKUjLtn6hqTu 11 - Yw5BFc2D9Wrcc4XkqJ8E 12 - BMEK9PsnXcEBC4sJ7Uj 13 - eKTf2b6CYrUwtGNHx73ua 14 - K9bYOs77xwMe3Nz26VrG4 15 - rj10WfTBg2A8s4HLb9Zf3w 16 - UnA320xw95CxJ6etyhBp4 17 - qM9tL9R9vD9s9uB9D9 18 - VbC9tG1rLufBEMhkoQ 19 - Dl2XrTmhvJpRULs0xaTH 20 - pkwBRt5wJB97UKfdTsZrb [sysadmin@rhserve scripts]\$</pre>
3	<p>For your student id, copy the line which contains your password.</p> <p>For this example, I am using 'student20' so I copy the last line to my buffer.</p>	<pre>sysadmin@rhserve:~/scripts\$ cat student-passwords.txt 01 - XjQgrFmzqXoWMe7Rj4KB 02 - BMEK9PsnXcEBC4sJ7Uj 03 - eKTf2b6CYrUwtGNHx73ua 04 - n9PAQn7jeab03EGwT2Z5 05 - Tr549k6g8bRkVjVgV73xH 06 - ZNz3Q9rD2wV3JyTeSuam 07 - qM9tL9R9vD9s9uB9D9 08 - Jxp3tTyntD29s9uScvWa 09 - dtuQVq9X4anzeqaP6q0 10 - YaQKNQ27bKUjLtn6hqTu 11 - Yw5BFc2D9Wrcc4XkqJ8E 12 - BMEK9PsnXcEBC4sJ7Uj 13 - eKTf2b6CYrUwtGNHx73ua 14 - K9bYOs77xwMe3Nz26VrG4 15 - rj10WfTBg2A8s4HLb9Zf3w 16 - UnA320xw95CxJ6etyhBp4 17 - qM9tL9R9vD9s9uB9D9 18 - VbC9tG1rLufBEMhkoQ 19 - Dl2XrTmhvJpRULs0xaTH 20 - pkwBRt5wJB97UKfdTsZrb [sysadmin@rhserve scripts]\$</pre>
4	<p>Return to your 'Text Editor' window that you had opened earlier. Paste the student password into this file for future reference.</p>	<pre>Open Save *Untitled Document 1 Deploy Token: f01bae02-ed0c-4131-ab1c-fc443abb895 API Connect User Password 20 20 - pkwBRt5wJB97UKfdTsZrb</pre>
5	<p>Return to your Firefox browser and your IBM DevOps Deploy tab.</p>	

IBM TechXchange



Step	Details	Additional Information
6	Click on the ‘Components’ tab at the top of the page as shown to the right.	
7	You will see a ‘emerald-APIC-Product’ component in the list. Click on the ‘ emerald-APIC-Product ’ component text	
8	Notice we have a component version already for this component. Let’s examine this by clicking on the ‘4.1.9’ component version.	
9	You will see that the component version consists of two yaml files: <ul style="list-style-type: none">• jgraceivt-product_1.0.1.yaml• jgraceivt-api_1.0.0.yaml The ‘product’ yaml describes your product that you want to publish to a catalog to expose the ‘api’ that it references which is in the ‘api’ yaml file above.	
10	Click on ‘emerald-APIC-Product’ in the crumb trail as shown to the right to return to the component.	

IBM TechXchange



Step	Details	Additional Information																																																
11	Click on the ‘Configuration’ sub-tab	<p>Components / emerald-APIC-Product</p> <p>Component: emerald-APIC-Product</p> <p>Created By admin Created On 11/29/2023, 4:41 PM</p> <p>Description An example API Connect API definition that uses a sample OpenAPI spec that points to an ACE endpoint</p> <p>Used By Emerald eStore View All</p> <p>Versions History Usage Configuration Calendar Processes Changes</p> <p>Component Versions</p> <table border="1"> <thead> <tr> <th>Version</th> <th>Statuses</th> <th>Type</th> <th>Created By</th> </tr> </thead> <tbody> <tr> <td>4.1.9</td> <td>Unit Tests Passed</td> <td>Full</td> <td>admin</td> </tr> </tbody> </table> <p>Items per page: 10 1-1 of 1 item</p>	Version	Statuses	Type	Created By	4.1.9	Unit Tests Passed	Full	admin																																								
Version	Statuses	Type	Created By																																															
4.1.9	Unit Tests Passed	Full	admin																																															
12	Click on ‘Component Properties’ in the left-hand pane	<p>Components / emerald-APIC-Product</p> <p>Component: emerald-APIC-Product</p> <p>Created By admin Created On 11/29/2023, 4:41 PM</p> <p>Description An example API Connect API definition that uses a sample OpenAPI spec that points to an ACE endpoint</p> <p>Versions History Usage Configuration Calendar Processes Changes</p> <p>Basic Settings</p> <p>Component Properties</p> <p>Environment Property Definitions Resource Property Definitions Version Property Definitions Configuration File Templates Version Import History</p> <p>Basic Settings</p> <p>Component Template: None</p> <p>Name: emerald-APIC-Product</p> <p>Description: An example API Connect API definition that uses a sample OpenAPI spec that points to an ACE endpoint</p> <p>Teams: Add Teams</p>																																																
13	You will see the component properties that will be used by our deployment process listed in the right-hand pane.	<p>Component: emerald-APIC-Product</p> <p>Created By admin Created On 11/29/2023, 4:41 PM</p> <p>Description An example API Connect API definition that uses a sample OpenAPI spec that points to an ACE endpoint</p> <p>Used By Emerald eStore View All</p> <p>Versions History Usage Configuration Calendar Processes Changes</p> <p>Component Properties</p> <p>Version 20 of 20</p> <table border="1"> <thead> <tr> <th>Index</th> <th>Name</th> <th>Value</th> <th>Is Inherited</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>env-api-target-url</td> <td>http://12.11.10.230:7080</td> <td>false</td> </tr> <tr> <td>1</td> <td>env-api-gateway-server-v1</td> <td>https://api-in-demo-ingress-api-manager-cp4-qm-fortang-wcf06-c3c16-bbd6d42b695ed4f9911bd1f20e822-0000.us-east.containers.appdomain.cloud/techexchange-gw</td> <td>false</td> </tr> <tr> <td>2</td> <td>apic.server</td> <td>https://api-in-demo-ingress-api-manager-cp4-qm-fortang-wcf06-c3c16-bbd6d42b695ed4f9911bd1f20e822-0000.us-east.containers.appdomain.cloud/techexchange-gw</td> <td>false</td> </tr> <tr> <td>3</td> <td>apic.username</td> <td>studentxx</td> <td>false</td> </tr> <tr> <td>4</td> <td>apic.password</td> <td>changeme</td> <td>false</td> </tr> <tr> <td>5</td> <td>apic.path</td> <td>/opt/IBM/apic-sim/apic</td> <td>false</td> </tr> <tr> <td>6</td> <td>apic.realm</td> <td>provider/default-lb-2</td> <td>false</td> </tr> <tr> <td>7</td> <td>apic.catalog</td> <td>sandbox</td> <td>false</td> </tr> <tr> <td>8</td> <td>apic.username</td> <td>studentxx</td> <td>false</td> </tr> <tr> <td>9</td> <td>apic.password</td> <td>changeme</td> <td>false</td> </tr> <tr> <td>10</td> <td>apic.path</td> <td>/opt/IBM/apic-sim/apic</td> <td>false</td> </tr> </tbody> </table>	Index	Name	Value	Is Inherited	0	env-api-target-url	http://12.11.10.230:7080	false	1	env-api-gateway-server-v1	https://api-in-demo-ingress-api-manager-cp4-qm-fortang-wcf06-c3c16-bbd6d42b695ed4f9911bd1f20e822-0000.us-east.containers.appdomain.cloud/techexchange-gw	false	2	apic.server	https://api-in-demo-ingress-api-manager-cp4-qm-fortang-wcf06-c3c16-bbd6d42b695ed4f9911bd1f20e822-0000.us-east.containers.appdomain.cloud/techexchange-gw	false	3	apic.username	studentxx	false	4	apic.password	changeme	false	5	apic.path	/opt/IBM/apic-sim/apic	false	6	apic.realm	provider/default-lb-2	false	7	apic.catalog	sandbox	false	8	apic.username	studentxx	false	9	apic.password	changeme	false	10	apic.path	/opt/IBM/apic-sim/apic	false
Index	Name	Value	Is Inherited																																															
0	env-api-target-url	http://12.11.10.230:7080	false																																															
1	env-api-gateway-server-v1	https://api-in-demo-ingress-api-manager-cp4-qm-fortang-wcf06-c3c16-bbd6d42b695ed4f9911bd1f20e822-0000.us-east.containers.appdomain.cloud/techexchange-gw	false																																															
2	apic.server	https://api-in-demo-ingress-api-manager-cp4-qm-fortang-wcf06-c3c16-bbd6d42b695ed4f9911bd1f20e822-0000.us-east.containers.appdomain.cloud/techexchange-gw	false																																															
3	apic.username	studentxx	false																																															
4	apic.password	changeme	false																																															
5	apic.path	/opt/IBM/apic-sim/apic	false																																															
6	apic.realm	provider/default-lb-2	false																																															
7	apic.catalog	sandbox	false																																															
8	apic.username	studentxx	false																																															
9	apic.password	changeme	false																																															
10	apic.path	/opt/IBM/apic-sim/apic	false																																															
14	For the apic.username and apic.password properties, use the three dots (layer cake) menu item and select ‘Edit’ to change these values to your studentxx number you were assigned and the password found in your Text Editor.	<p>For the apic.username and apic.password properties, use the three dots (layer cake) menu item and select ‘Edit’ to change these values to your studentxx number you were assigned and the password found in your Text Editor.</p>																																																

IBM TechXchange



Step	Details	Additional Information																								
15	<p>The values for the ‘apic.username’ and ‘apic.password’ should reflect what you were assigned as shown to the right.</p> <p>NOTE: you can ‘secure’ passwords in IBM DevOps Deploy where a user cannot see them.</p>	<p>Component Properties</p> <p>Version 22 of 22 « « » »»</p> <table border="1"> <thead> <tr> <th>Index</th> <th>Name</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>env-api-target-url</td> <td>http://52.116.10.230:7080</td> </tr> <tr> <td>1</td> <td>env-api-gateway-server-url</td> <td>https://apim-demo-mgmt-api-manager-cp4i.ajr-xx/sandbox/jgraceiv/v1</td> </tr> <tr> <td>2</td> <td>apic.server</td> <td>https://apim-demo-mgmt-api-manager-cp4i.ajr</td> </tr> <tr> <td>3</td> <td>apic.username</td> <td>student20</td> </tr> <tr> <td>4</td> <td>apic.password</td> <td>pkW8Rt5wjBA97UKfDTsZrb</td> </tr> <tr> <td>5</td> <td>apic.path</td> <td>/opt/IBM/apic-slim/apic</td> </tr> <tr> <td>6</td> <td>apic.realm</td> <td>provider/default-idp-2</td> </tr> </tbody> </table>	Index	Name	Value	0	env-api-target-url	http://52.116.10.230:7080	1	env-api-gateway-server-url	https://apim-demo-mgmt-api-manager-cp4i.ajr-xx/sandbox/jgraceiv/v1	2	apic.server	https://apim-demo-mgmt-api-manager-cp4i.ajr	3	apic.username	student20	4	apic.password	pkW8Rt5wjBA97UKfDTsZrb	5	apic.path	/opt/IBM/apic-slim/apic	6	apic.realm	provider/default-idp-2
Index	Name	Value																								
0	env-api-target-url	http://52.116.10.230:7080																								
1	env-api-gateway-server-url	https://apim-demo-mgmt-api-manager-cp4i.ajr-xx/sandbox/jgraceiv/v1																								
2	apic.server	https://apim-demo-mgmt-api-manager-cp4i.ajr																								
3	apic.username	student20																								
4	apic.password	pkW8Rt5wjBA97UKfDTsZrb																								
5	apic.path	/opt/IBM/apic-slim/apic																								
6	apic.realm	provider/default-idp-2																								
16	In the component properties list, scroll down to the ‘apic.organization’ property.	<table border="1"> <tbody> <tr> <td>5</td> <td>apic.path</td> <td>/opt/IBM/apic-slim/apic</td> </tr> <tr> <td>6</td> <td>apic.realm</td> <td>provider/default-idp-2</td> </tr> <tr> <td>7</td> <td>apic.catalog</td> <td>sandbox</td> </tr> <tr> <td>8</td> <td>apic.organization</td> <td>techxchange-org-xx</td> </tr> </tbody> </table>	5	apic.path	/opt/IBM/apic-slim/apic	6	apic.realm	provider/default-idp-2	7	apic.catalog	sandbox	8	apic.organization	techxchange-org-xx												
5	apic.path	/opt/IBM/apic-slim/apic																								
6	apic.realm	provider/default-idp-2																								
7	apic.catalog	sandbox																								
8	apic.organization	techxchange-org-xx																								
17	<p>Edit this value to be ‘techxchange-org-xx’ where xx is your student number.</p> <p>In my case, it is ‘techxchange-org-20’ as shown to the right.</p>	<table border="1"> <tbody> <tr> <td>8</td> <td>apic.organization</td> <td>techxchange-org-20</td> </tr> </tbody> </table>	8	apic.organization	techxchange-org-20																					
8	apic.organization	techxchange-org-20																								
18	<p>We will now quickly review the component process for deployment.</p> <p>Click on the ‘Processes’ sub-tab as shown to the right.</p>	<p>DevOps Deploy / emerald-APIC-Product</p> <p>Component: emerald-APIC-Product</p> <p>Created By: admin Created On: 11/29/2023, 4:41 PM</p> <p>Description: An example API Connect API definition that uses a sample OpenAPI spec that points to an ACE endpoint.</p> <p>Used By: Emerald eStore View All</p> <p>Versions History Usage Configuration Processes Resources Calendar Work Items</p> <p>Configuration</p> <p>Component Properties</p> <p>Version 24 of 24</p> <table border="1"> <thead> <tr> <th>Index</th> <th>Name</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>env-api-target-url</td> <td>http://52.116.10.230:7080</td> </tr> <tr> <td>1</td> <td>env-api-gateway-server-url</td> <td>https://apim-demo-mgmt-api-manager-cp4i.ajr-xx/sandbox/jgraceiv/v1</td> </tr> <tr> <td>2</td> <td>apic.server</td> <td>https://apim-demo-mgmt-api-manager-cp4i.ajr-forlang-wdc06-c3</td> </tr> <tr> <td>3</td> <td>apic.username</td> <td>student20</td> </tr> </tbody> </table>	Index	Name	Value	0	env-api-target-url	http://52.116.10.230:7080	1	env-api-gateway-server-url	https://apim-demo-mgmt-api-manager-cp4i.ajr-xx/sandbox/jgraceiv/v1	2	apic.server	https://apim-demo-mgmt-api-manager-cp4i.ajr-forlang-wdc06-c3	3	apic.username	student20									
Index	Name	Value																								
0	env-api-target-url	http://52.116.10.230:7080																								
1	env-api-gateway-server-url	https://apim-demo-mgmt-api-manager-cp4i.ajr-xx/sandbox/jgraceiv/v1																								
2	apic.server	https://apim-demo-mgmt-api-manager-cp4i.ajr-forlang-wdc06-c3																								
3	apic.username	student20																								

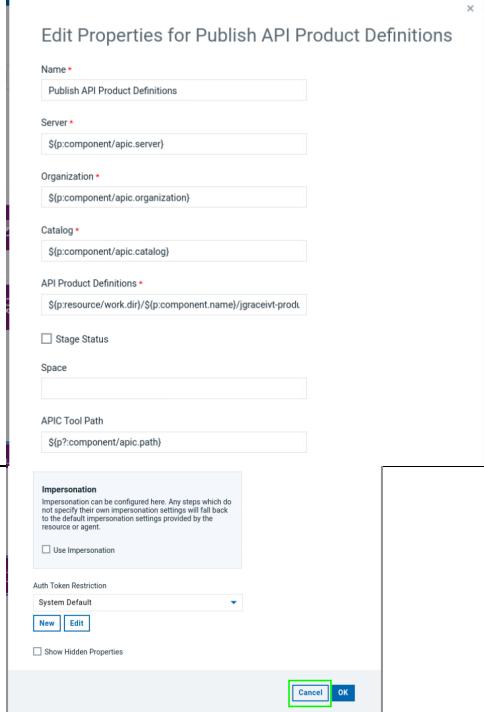
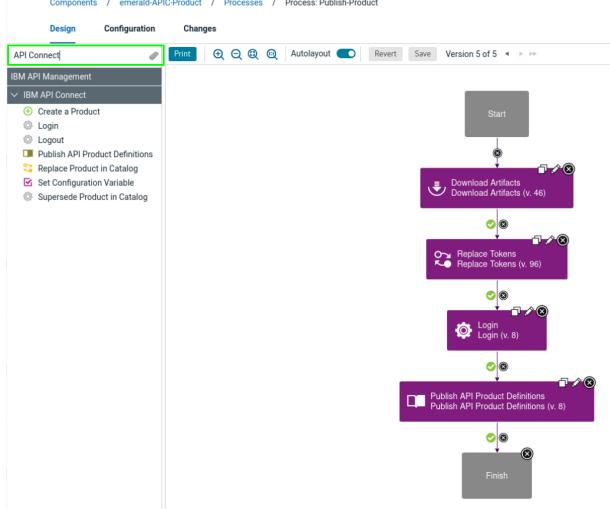
IBM TechXchange



Step	Details	Additional Information
19	Click on the 'Publish-Product' process	<p>The screenshot shows the 'Components / emerald-APIC-Product' page. Under the 'Processes' tab, the 'Publish-Product' process is selected. The description states: 'An example API Connect API definition that uses a sample OpenAPI spec that points to an ACCE endpoint'. It is used by 'Emerald eStore'. The process has two steps: 'DraftProduct' and 'PublishProduct'. The 'PublishProduct' step is highlighted with a green border.</p>
20	<p>The visual process designer will appear.</p> <p>The process starts with a 'Download Artifacts' step that will download the yaml files to a host that has the API Connect Toolkit installed.</p> <p>The second step will replace tokens in the 'jgraceivt-api_1.0.0.yaml' file with environment properties set in DevOps Deploy. At time of deployment, these values will be substituted (these values can vary from one environment to the next). Here is our example:</p> <pre data-bbox="241 1262 540 1431"> 74 properties: 75 target-url: 76 value: TARGET URL 77 description: The URL of the target service 78 enabled: false 79 activity-log: 80 enabled: true 81 success-content: activity 82 error-content: payload 83 servers: 84 - url: API GATEWAY URL 85 description: '' 86 type: 87 - production 88 - development </pre> <p>The third and fourth steps of the process use the APIC plugin steps to Login and 'Publish the API Product Definition' to your API Connect target environment catalog.</p>	<p>The screenshot shows the 'DevOps Deploy' interface with the 'Processes' tab selected. A process named 'Process: Publish-Product' is displayed. The workflow consists of the following steps:</p> <pre> graph TD Start((Start)) --> DA1[Download Artifacts Download Artifacts v. 46] DA1 --> RT[Replace Tokens Replace Tokens v. 96] RT --> L[Login Login v. 8] L --> PAPD[Publish API Product Definitions Publish API Product Definitions v. 8] PAPD --> Finish((Finish)) </pre> <p>The 'Download Artifacts' step is highlighted with a purple background. The 'Replace Tokens' and 'Login' steps are also highlighted with purple backgrounds. The 'Publish API Product Definitions' step is highlighted with a pink background.</p>
21	Click the pencil icon for the 'Publish API Product Definitions' step.	

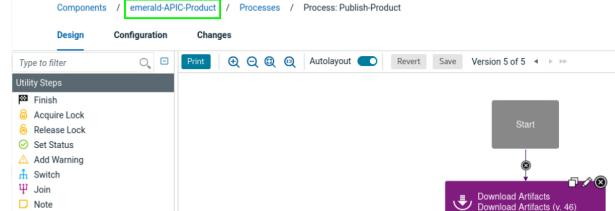
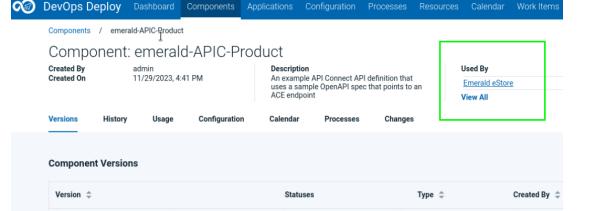
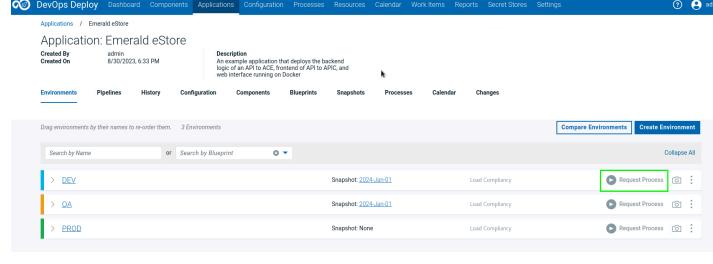
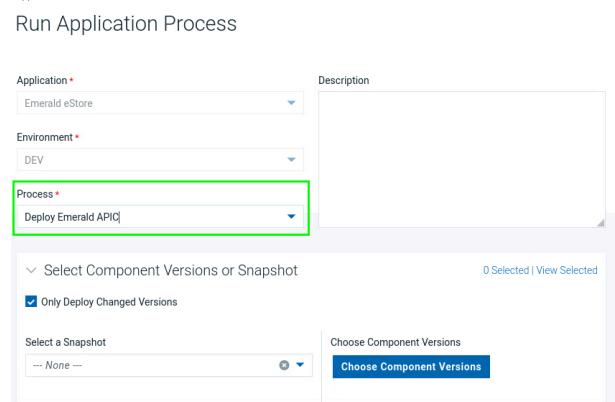
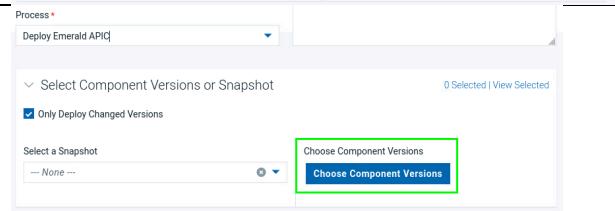
IBM TechXchange



Step	Details	Additional Information
22	Notice the use of the component properties that we reviewed and updated.	
23	Scroll down and click Cancel to close the 'Edit Properties' dialog without making any changes.	
24	<p>In your visual process design view, type 'API Connect' in the 'Type to filter' search text field as shown to the right.</p> <p>This will show you the plugin steps that come with the 'IBM API Connect' automation plugin that you can leverage in IBM DevOps Deploy.</p> <p>NOTE: IBM DevOps Deploy has plugins for MQ, DataPower and other technologies that many of our clients manage to run operational processes (e.g. 'Create Queue Manager') or deployment processes.</p>	

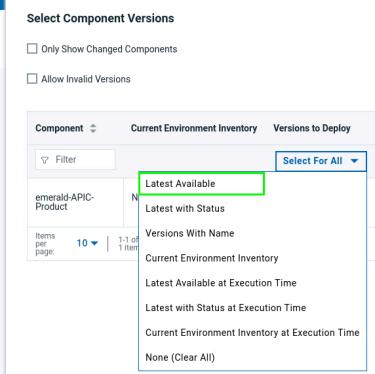
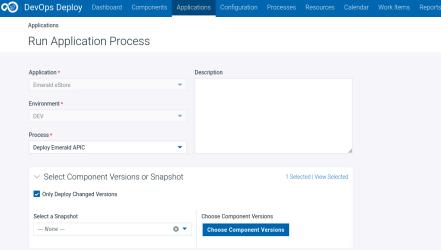
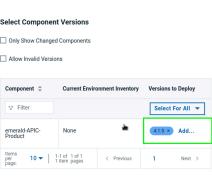
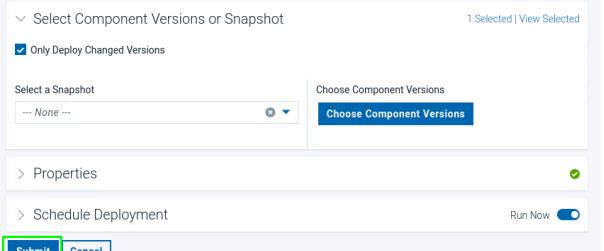
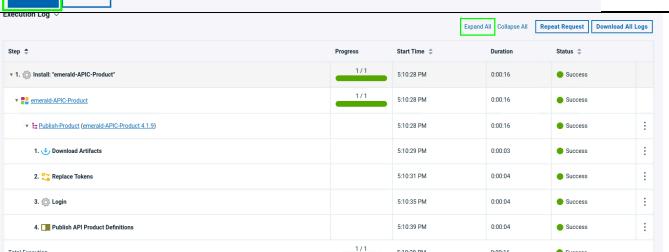
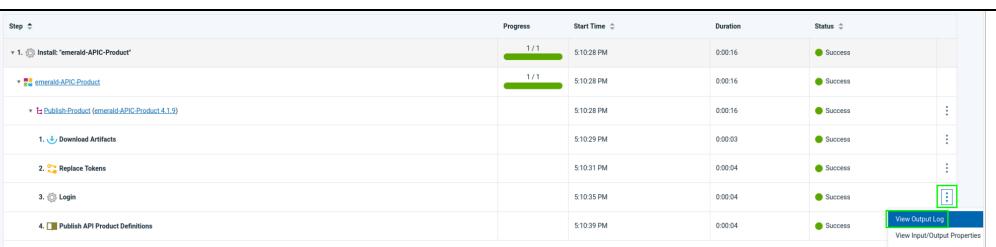
IBM TechXchange



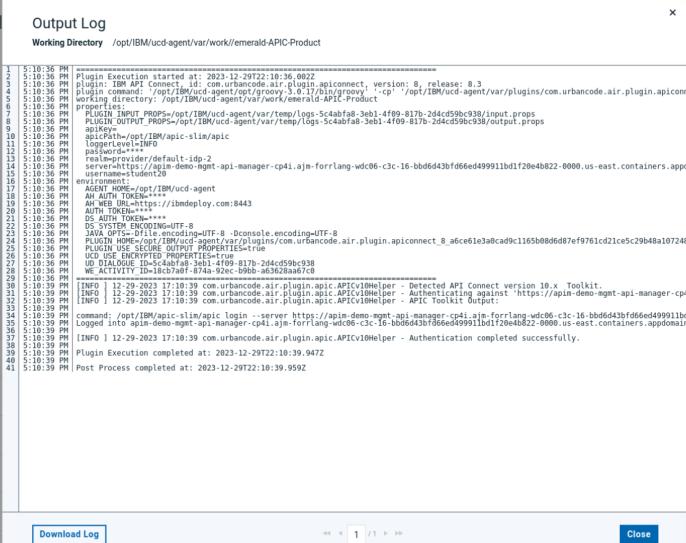
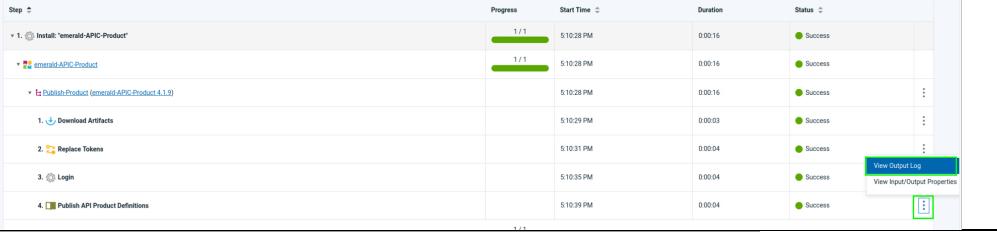
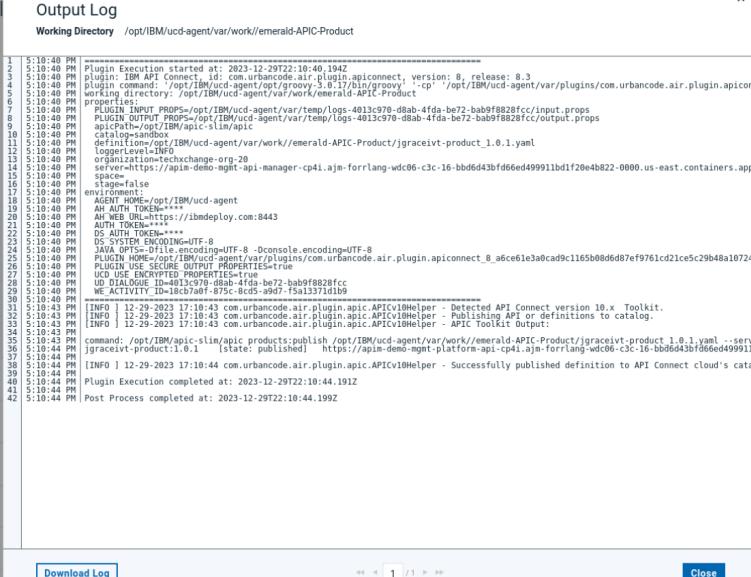
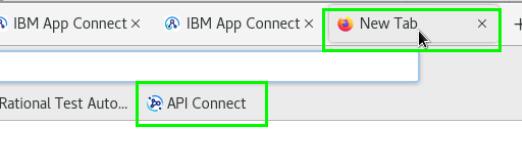
Step	Details	Additional Information
25	In the crumb trail at the top, click on ' emerald-APIC-Product ' to return to the component view.	 <p>The screenshot shows the 'Components' section of the IBM API Connect interface. The crumb trail at the top shows 'Components / emerald-APIC-Product / Processes / Process: Publish-Product'. A green box highlights the component name 'emerald-APIC-Product'. On the right, there is a process diagram with a 'Start' node and a 'Download Artifacts' node.</p>
26	<p>At the top of the Components view, you will see a 'Used By' field as shown to the right.</p> <p>Click on 'Emerald eStore' to return to the application.</p>	 <p>The screenshot shows the 'Components' section of the IBM API Connect interface. The component 'emerald-APIC-Product' is selected. A green box highlights the 'Used By' field, which shows 'Emerald eStore'. Below it, there is a table for 'Component Versions'.</p>
27	<p>From the Application view, we will now deploy our API Connect product to the DEV environment.</p> <p>Click on the 'Request Process' button for the DEV environment as shown to the right.</p>	 <p>The screenshot shows the 'Applications' section of the IBM DevOps Deploy interface. The application 'Emerald eStore' is selected. A green box highlights the 'Request Process' button for the 'DEV' environment.</p>
28	In the Run Application Process page that appears, for the Process pull-down menu, select ' Deploy Emerald APIC '	 <p>The screenshot shows the 'Run Application Process' dialog box. The 'Process' dropdown menu is open, and a green box highlights the option 'Deploy Emerald APIC'.</p>
29	Click the 'Choose Component Versions' button	 <p>The screenshot shows the 'Run Application Process' dialog box. A green box highlights the 'Choose Component Versions' button.</p>

IBM TechXchange



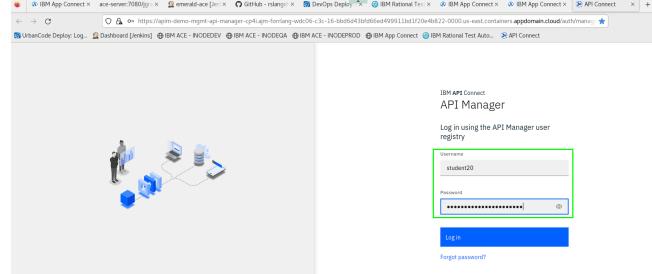
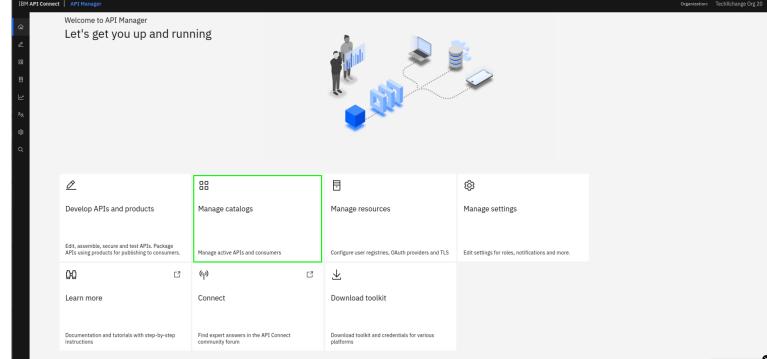
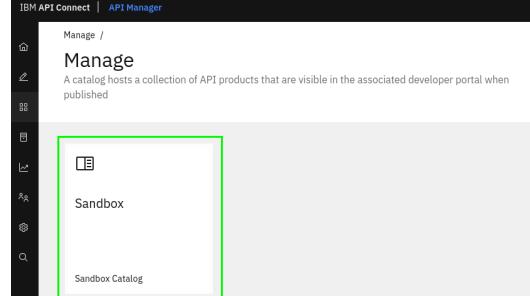
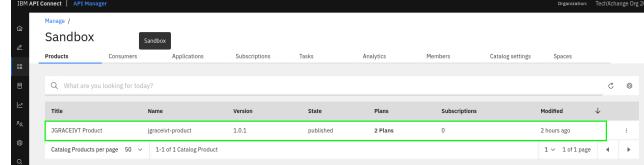
Step	Details	Additional Information
30	From the 'Select for All' pulldown that appears, select the ' Latest Available '	
31	You should see version 4.1.9 selected to be deployed	 
32	Scroll down and click the 'Submit' button	
33	The deployment will begin. You can click the 'Expand All' link to show each step that is executed with the deployment.	
34	The deployment should succeed as shown above.	
35	View the output for the 'Login' step by clicking on the layer cake (three dot) icon to the right of the step and then select 'View Output Log'	

IBM TechXchange

Step	Details	Additional Information
36	You should see ‘Authentication completed successfully’ as shown to the right. Click ‘Close’ to close the Output Log dialog window.	
37	Repeat the same steps to view the output for the ‘Publish API Product Definitions’ step.	
38	You should see the ‘[state: published]’ message as shown to the right along with a ‘Successfully published definition’ message in the output log. Click Close to close the Output Log dialog window.	
39	Open a new tab in your browser. Click on the ‘API Connect’ bookmark as shown to the right.	

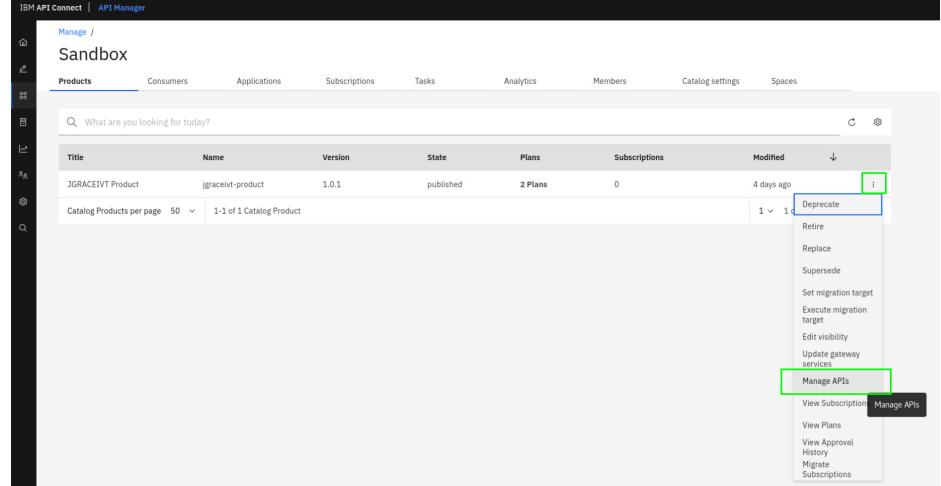
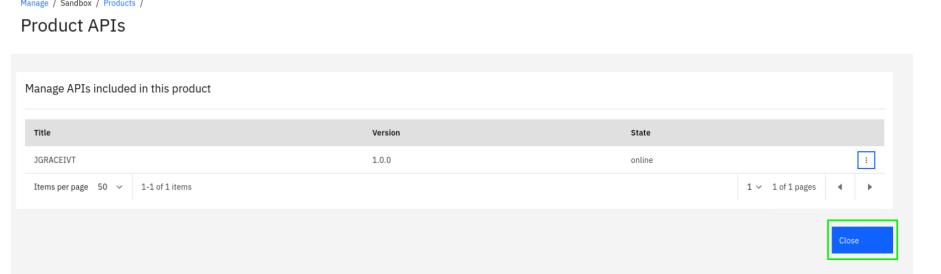
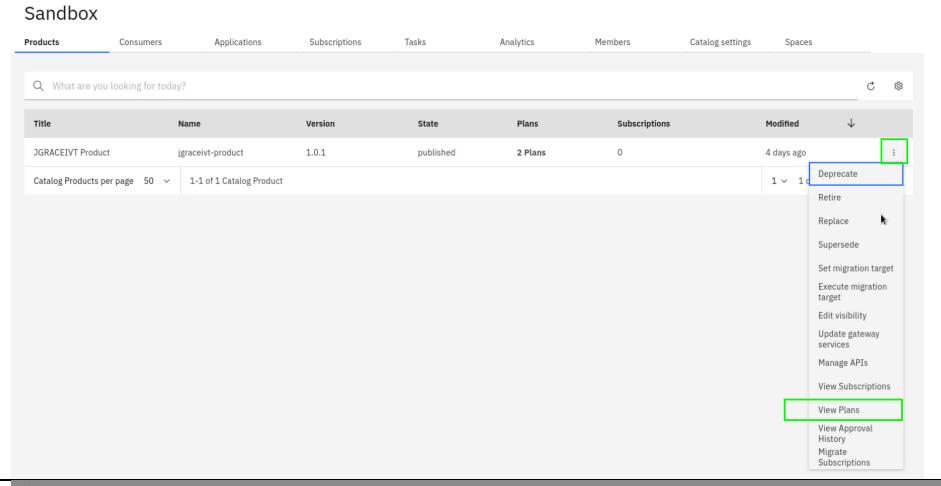
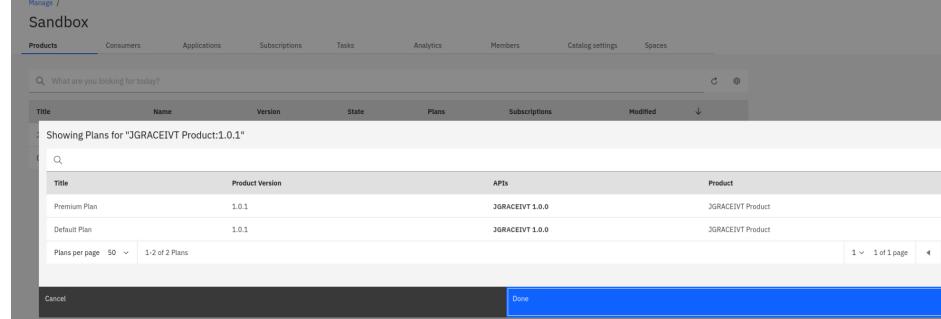
IBM TechXchange



Step	Details	Additional Information														
40	<p>The API Connect Login Page will appear.</p> <p>For the Username field, enter your studentxx number.</p> <p>Copy the studentxx password from your Text Editor window. Paste this value into the Password field.</p>															
41	Click the 'Log in' button.															
42	Click on the ' Manage catalogs ' tile as shown to the right.															
43	Click on the 'Sandbox' catalog															
44	You should see the 'JGRACEIVT Product' in the list of products following your deployment as shown to the right!	 <table border="1"> <thead> <tr> <th>Title</th> <th>Name</th> <th>Version</th> <th>State</th> <th>Plans</th> <th>Subscriptions</th> <th>Modified</th> </tr> </thead> <tbody> <tr> <td>JGRACEIVT Product</td> <td>jgraceivt-product</td> <td>1.0.1</td> <td>published</td> <td>2 Plans</td> <td>0</td> <td>2 hours ago</td> </tr> </tbody> </table>	Title	Name	Version	State	Plans	Subscriptions	Modified	JGRACEIVT Product	jgraceivt-product	1.0.1	published	2 Plans	0	2 hours ago
Title	Name	Version	State	Plans	Subscriptions	Modified										
JGRACEIVT Product	jgraceivt-product	1.0.1	published	2 Plans	0	2 hours ago										

IBM TechXchange



Step	Details	Additional Information
45	By clicking on the layer cake (three dot) menu to the right, select 'Manage APIs' to view the APIs that are securely exposed to end users.	
46	<p>Here you see the APIs included with the product.</p> <p>Click the 'Close' button</p>	
47	Repeat the same process to view the plans by clicking the 'View Plans' menu item.	
48	You will see there is a 'Default Plan' as well as a 'Premium Plan' provided for this API that end users can subscribe to.	



IBM TechXchange

Step	Details	Additional Information
49	You have completed the lab!	
End of Section – 3.7		



IBM TechXchange

4 Wrapping Things Up

4.1 Lab Re-Cap

Congratulations!

In this workbook you have gone through the process of integrating IBM DevOps capabilities with IBM App Connect Enterprise (ACE) and API Connect to provide an end-to-end DevOps experience. By executing each lab, you were able to:

- Explore IBM App Connect Enterprise (ACE) and how one can create, build, and deploy APIs from both the ACE Toolkit IDE and through a pipeline like Jenkins which integrates with DevOps Deploy to handle the deployment.
- Explore IBM DevOps Deploy and its application model which includes target environments, components, and processes. We explored how to setup processes to handle the deployment of both IBM App Connect Enterprise (ACE) and API Connect assets in the labs.
- Explore Rational Test Automation Server (RTAS) and how it can run automated tests and be triggered automatically as part of a deployment process.
- Explore IBM API Connect and how you can automate the publish of a product to a target catalog using DevOps Deploy. IBM API Connect is a full lifecycle API management solution that is used to create, manage, secure, socialize, and monetize APIs.

For more information on each solution, please refer to the documentation links below:

IBM DevOps Test Hub Overview - <https://www.ibm.com/products/devops-test/hub>

IBM DevOps Deploy Documentation - <https://www.ibm.com/docs/en/urancode-deploy/8.0.0?topic=notes-whats-new>

IBM App Connect Enterprise Documentation - <https://www.ibm.com/docs/en/app-connect/12.0?topic=overview-whats-new-in-version-120>

IBM API Connect Overview - <https://www.ibm.com/products/api-connect>