



Hands-on Lab

Session SL7028 / OL9613

Manage the API Lifecycle using IBM UrbanCode Deploy and API Connect

Paul Bahrs, IBM Cloud
Hollis Chui, IBM Cloud
Steven M Cotugno, IBM Cloud
Ahmed Abdelaziz, IBM Cloud



© Copyright IBM Corporation 2017

IBM, the IBM logo and ibm.com are trademarks of International Business Machines Corp., 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.

This document is current as of the initial date of publication and may be changed by IBM at any time.

The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. This information is based on current IBM product plans and strategy, which are subject to change by IBM without notice. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way.

Table of Contents

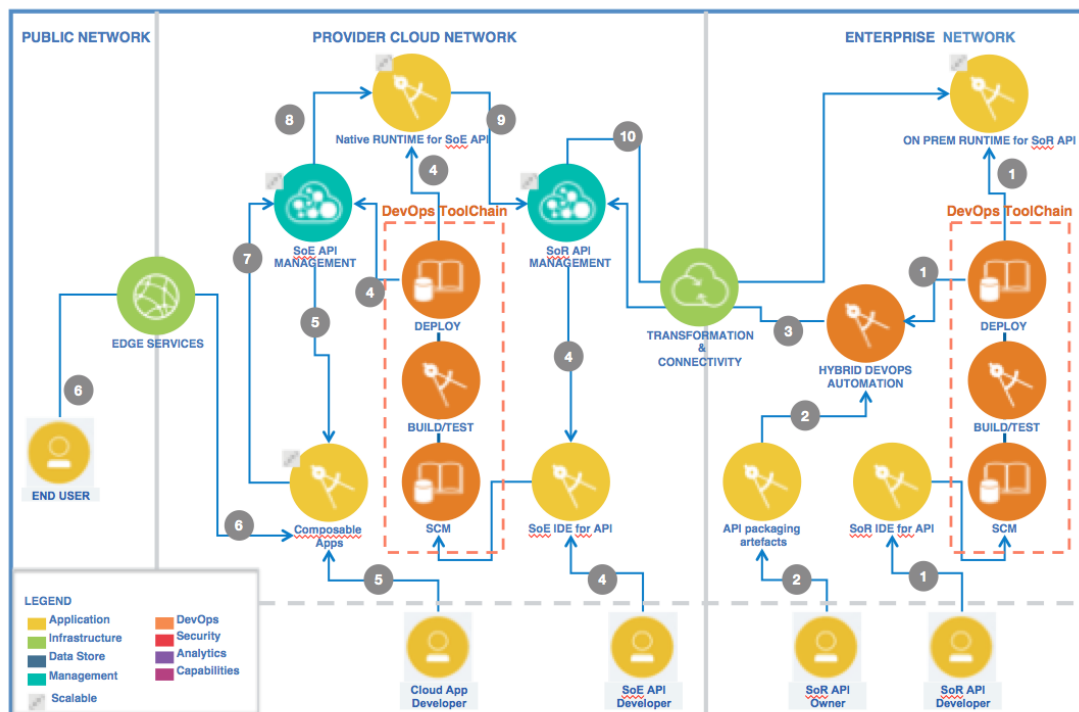
Introduction	4
Business Scenario	4
Lab Overview	6
What is Already Completed	8
Lab Exercises	12
Lab 0: Create Bluemix and GitHub Accounts	12
Lab 0.1: Create Bluemix trial account	12
Lab 0.2: Create GitHub account	12
Lab 1: Bluemix API Connect service and Catalogs	13
Lab 1.1: Create an instance of the API Connect Service	13
Lab 1.2: Create UAT and PROD Catalogs	15
Lab 2: GitHub Repository	18
Lab 2.1: Create your GitHub Repository	18
Lab 2.2: Clone acme-bank-api-yaml GitHub repository	18
Lab 2.3: Push clone content into your GitHub repository	19
Lab 2.4: Delete the clone repository directory	19
Lab 2.5: Clone your GitHub repository as a local workspace	19
Lab 3: GitHub and Jenkins integration	21
Lab 3.1: Configure the GitHub and Jenkin integration	21
Lab 4: IBM UrbanCode Deploy and Jenkins integration	24
Lab 4.1: Configure the IBM UrbanCode Deploy and Jenkins integration	24
Lab 5: Jenkins Configuration	25
Lab 5.1: Update Jenkins Job	25
Lab 6: IBM UrbanCode Deploy Configuration	27
Lab 6.1: Start the IBM UrbanCode Deploy Agent as root	27
Lab 6.2: Update component process with Bluemix password	27
Lab 6.3: Update Environment Properties	27
Lab 7: Demonstrate API Governance	29
Lab 7.1: Major change release flow	29
Lab 7.2: Minor change release flow	37
Lab 7.3: Bugfix change release flow	47
Lab 7.4: (Optional) Publish a major version release	53

Introduction

This lab will go through a governance model of automating API deployment using IBM UrbanCode Deploy (UCD) and the API Connect service on Bluemix. Through the exercises, you will have a working sample of managing the API lifecycle through automation.

Business Scenario

Businesses recognize that hybrid cloud solutions will require their development teams to have access to Systems of Record (SoRs) to expose the needed APIs for use by System of Engagement (SoEs) developers in the Cloud. By exposing these APIs through the hybrid cloud solution, both teams can perform their development work with relative independence.



Objectives

In this lab, you will go through the API governance model process outlined in the diagram below. The diagram shows how API implementations and definitions are published to API Connect based on the type of changes and the corresponding version increment of the Product / API definitions.

Dev	Non Prod Environments/Catalogs	Pre-Prod/Bug Fix Environment/Catalog	Production
Major Release	Replace last deployed	Publish	Publish
Minor Release	Replace last deployed	Replace specific version	Replace specific version
Bug Fix		Replace specific version	Replace specific version

- For major (API breaking) changes, defined as a change impacting the API endpoints or major changes to the API implementation, this will result in a major version increment (1.0.0 -> 2.0.0). The new version of the API will be published to the UAT and PROD catalogs and the subscriptions will be manually migrated. Once all the subscriptions are migrated, the older version of the API will be set to a “retired” state.
- For minor (non-breaking) changes, there will be a minor version increment (1.0.0 -> 1.1.0). The new version of the API will replace the existing version in the UAT and PROD catalogs. The subscriptions will be automatically migrated and the older version of the API will be set to the “retired” state.
- For bug fixes and/or patches to the API implementation that do not impact the API interface, no version changes to the API are expected. A bugfix will result in a version increment of the third digit (1.1.0 -> 1.1.1) and the bugfix change(s) will replace the existing version in the UAT and PROD catalogs.

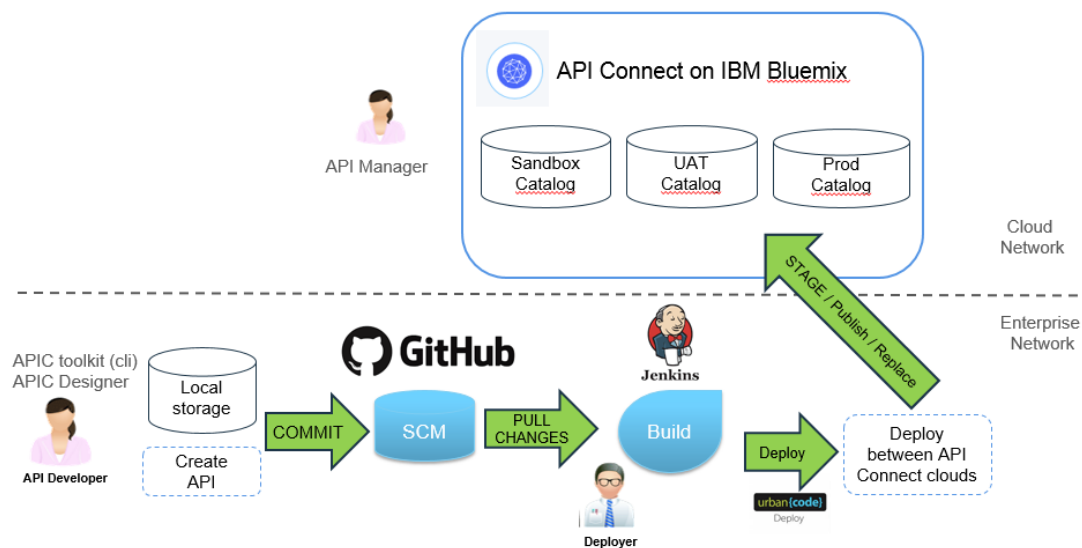
Lab Overview

The lab will go through the DevOps workflow for publishing APIs using the governance model discussed above.

As a developer creates code changes to the API implementation and yaml files for the API endpoints, these changes are tracked in source control. The changes are put through the delivery pipeline and the changes are built, deployed and tested as part of the DevOps process while the developer continues to code.

The API product owner delivers changes to the Product yaml files to reflect product level changes and the Product definitions are also stored in source control.

The changes to the API and Product yaml are promoted automatically to environments based on a defined policy or the manual activation of an automation process. This process ensures each environment / catalog will have the latest API and Product yaml files.





Before we begin the exercises, we must review the relationship between IBM UrbanCode Deploy and the API Connect service. The following table provides a mapping between API Connect and IBM UrbanCode Deploy objects in the usage model we follow in this lab exercise.

API Connect	IBM UrbanCode Deploy	Description
Catalog	Environment	A UCD environment represents an API Connect Catalog. Multiple Products may be deployed to an environment (catalog) as each product is represented by a product component resource in the environment.
Product	Component	Each API Product within API Connect will be represented by one UCD component. The component is a logical grouping of the deployable artifacts of a given API Product, each component can have multiple versions where each UCD component version represent a particular release or build of the corresponding API Product. The API Product contains the definition of one or more APIs.
API Implementation	Component(s)	Each API implementation will be represented by a UCD Component. Each API defined in the Product has a corresponding implementation. The source code that implements the API may be from a variety of technologies such as StrongLoop Node.js, java, javascript or other languages. This component stores the deployable versioned artifacts that result from the build of the API's implementation source code.

What is Already Completed

The lab will be run on an Ubuntu VM hosted on Skytap. Below are user / password for the accounts used to access the pre-installed software.

Account Information:

- Ubuntu VM: devops / devops
- Jenkins: admin / admin
- IBM UrbanCode Deploy: admin /admin

Installed Software:

- IBM UrbanCode Deploy 6.2.2 + local agent
 - API Connect plugin for UCD
- Jenkins 2.19
 - IBM UrbanCode Deploy Build Steps plugin
 - Git Tag Message plugin
 - GitHub plugin
- Node.js 4.x
- API Connect Toolkit

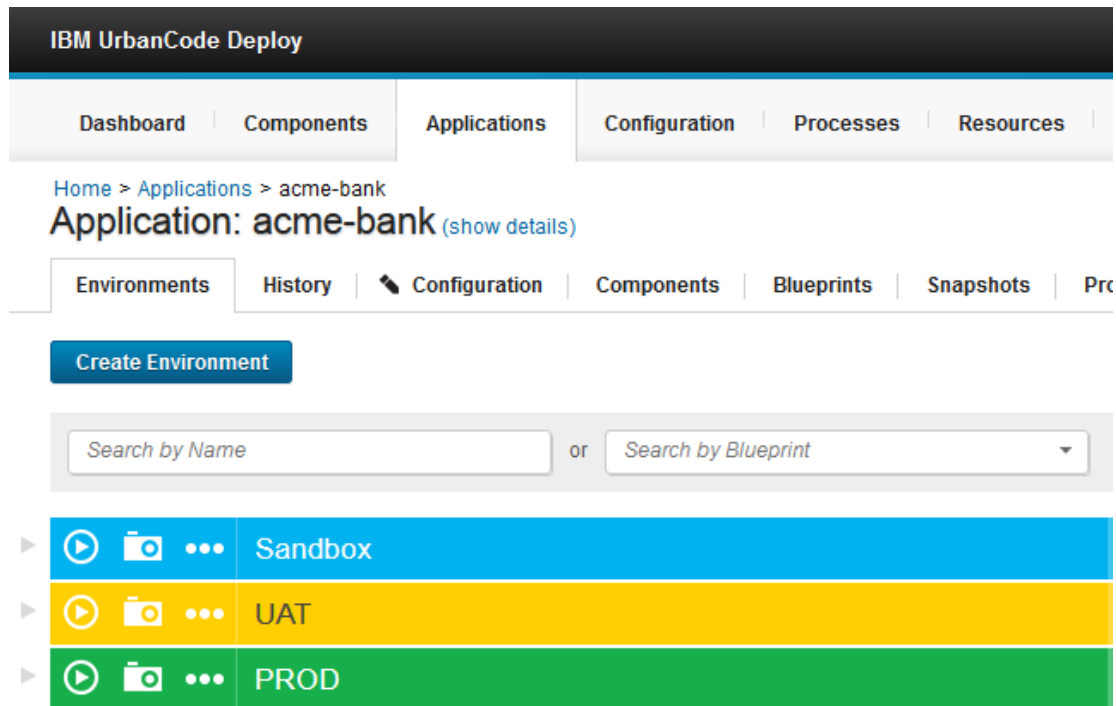
Jenkins 'acme-bank' job configuration:

The **acme-bank** job has been configured to run a build when a change is pushed to a specified GitHub repository. Once the build request is triggered, the job will perform the following activities:

- Download the GitHub repository files into the Jenkins workspace.
- Inject environment properties defined in the build.properties file.
- Create a component version in IBM UrbanCode Deploy and set component version properties.
- Add the "release" version status to the component version if it is a bugfix, so the Product / API can be deployed directly into the UAT environment / catalog.
- Publish the component version into an environment which will result in the publishing of the Product / API on API Connect.

IBM UrbanCode Deploy Environments and Environment Gates:

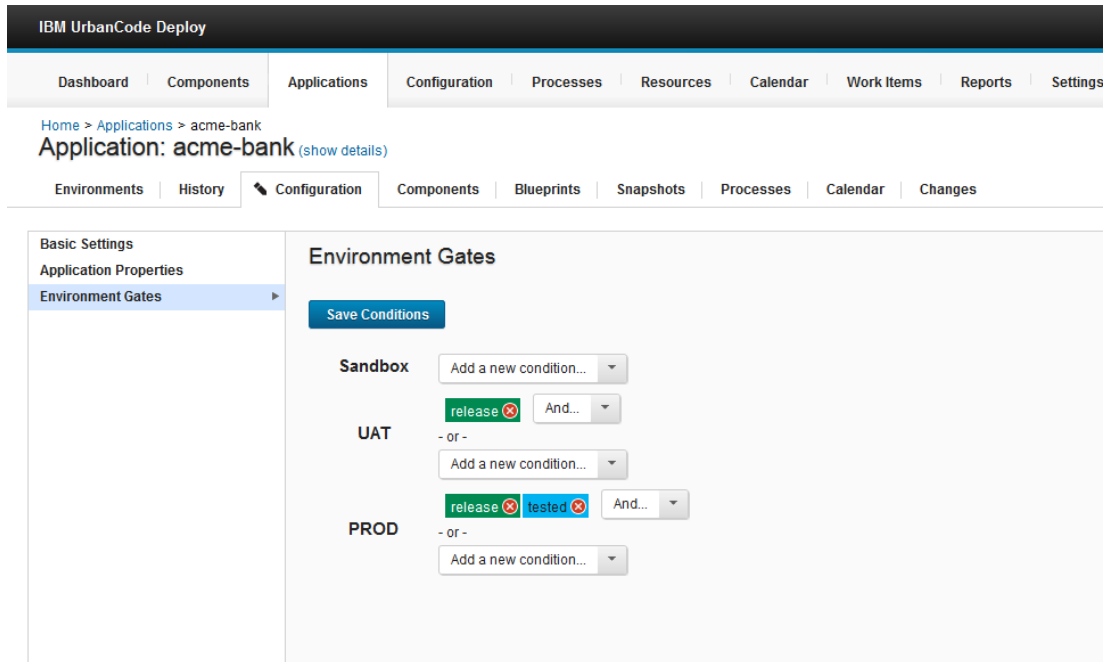
Three environments (Sandbox, UAT and PROD) have been set up in the acme-bank application to match the catalogs set up in API Connect.



Environment	Color
Sandbox	Blue
UAT	Yellow
PROD	Green

As per the API governance model, a policy is defined to control the environment(s) a change can be deployed to. The policy is configured using Environment Gates.

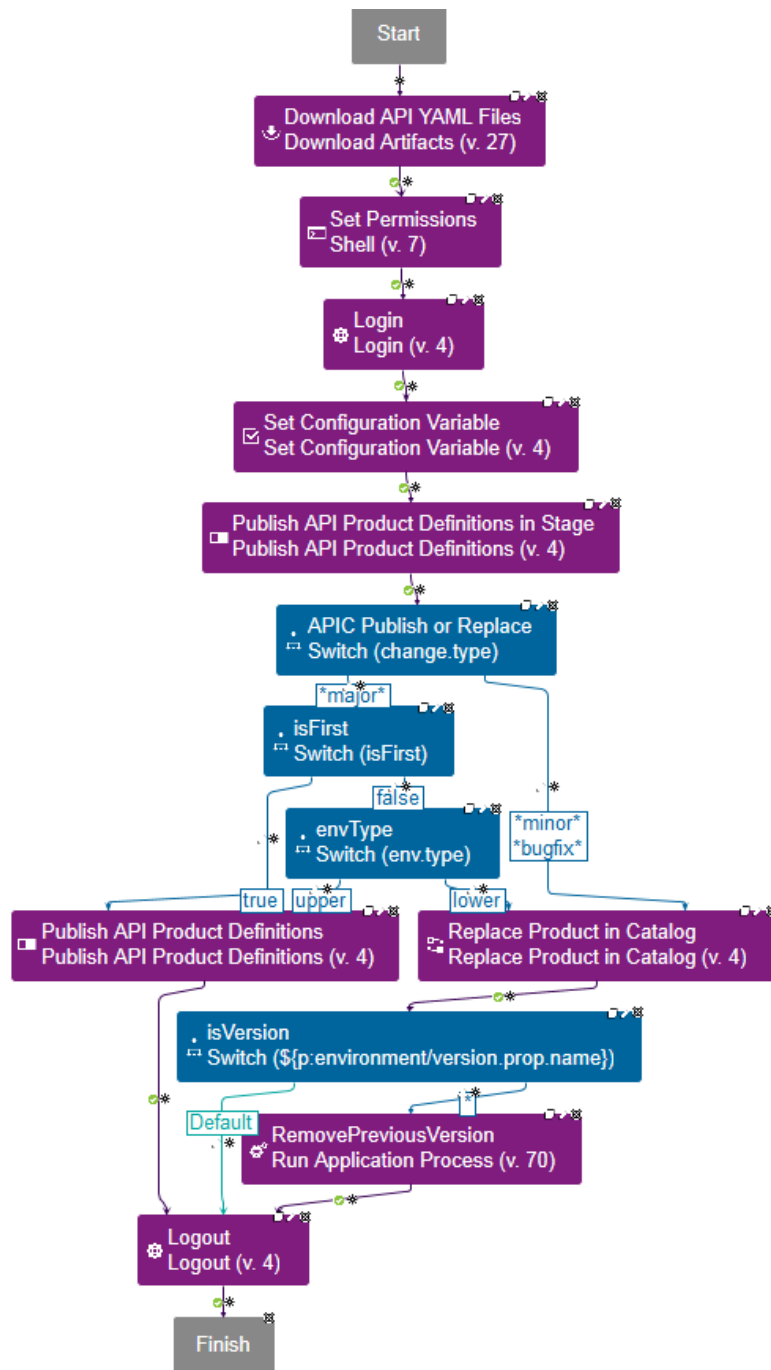
As seen in the diagram below, any deployments can be deployed to the Sandbox environment. A deployment must be tagged with a “**release**” status for it to be deployed into the UAT environment. Finally, a deployment must be tagged with both the “**release**” and “**tested**” statuses to be deployed into the PROD environment.



The screenshot displays the IBM UrbanCode Deploy web interface. At the top, a dark header bar contains the text "IBM UrbanCode Deploy". Below this is a navigation bar with tabs: Dashboard, Components, Applications, Configuration, Processes, Resources, Calendar, Work Items, Reports, and Settings. The "Applications" tab is selected, and the breadcrumb trail shows "Home > Applications > acme-bank". The main heading is "Application: acme-bank" with a "(show details)" link. Below this is another navigation bar with tabs: Environments, History, Configuration (selected), Components, Blueprints, Snapshots, Processes, Calendar, and Changes. The left sidebar shows a menu with "Basic Settings", "Application Properties", and "Environment Gates" (selected). The main content area is titled "Environment Gates" and features a "Save Conditions" button. It lists three environments: Sandbox, UAT, and PROD. Each environment has a "Add a new condition..." dropdown. The UAT environment has a "release" condition with a red 'x' icon. The PROD environment has "release" and "tested" conditions, both with red 'x' icons. Each condition has an "And..." dropdown next to it.

IBM UrbanCode Deploy Component Process:

The component process in combination with the Jenkins job configuration implements the API governance model discussed in the Objectives section.



Lab Exercises

To achieve the objectives of this lab, the following exercises must be completed:

Lab 0: Create Bluemix and GitHub Accounts

Lab 0.1: Create Bluemix trial account

1. If you already have an active Bluemix account, skip this task.
2. Open a web browser and go to <https://console.ng.bluemix.net/>.
3. Click on the **“Create a free account”** button.
4. Follow the directions to fill out the form and make a note of the password specified. Note, you will need access to the email to confirm the account creation.
5. Click **“Create Account”** and Bluemix will send a confirmation email to the account specified.
6. Login into the email account specified and open the email with the subject: **Action Required: Confirm your Bluemix account.**
7. Click on the **“Confirm Account”** button.
8. You now have an active Bluemix trial account.
9. When specifying the name of your Bluemix Org, do not use any special characters.

Lab 0.2: Create GitHub account

1. If you already have a GitHub account, skip this task.
2. Open a web browser and go to <https://github.com/>.
3. Follow the directions to fill out the form and make a note of the password specified. Note, you will need access to the email to confirm the account creation.
4. Click on the **“Sign up for GitHub”** button and GitHub will send a confirmation email to the account specified.
5. Login into the email account specified and open the email from GitHub with the subject: **Please verify your email address.**
6. Click on the **“Verify email address”** link.
7. You now have an active GitHub account.

Lab 1: Bluemix API Connect service and Catalogs

Lab 1.1: Create an instance of the API Connect Service

1. Open the Firefox browser, go to the Bluemix website (<https://console.ng.bluemix.net/>) and click the “Log in” button.



Welcome to Bluemix, the home of 130+ unique services. Start building immediately.

Create a free account

Log in

Learn more:

[Pricing](#) [Catalog](#) [Docs](#) [Support](#)

2. Log in with your Bluemix username and password.
3. Click on Catalog on the top right corner.

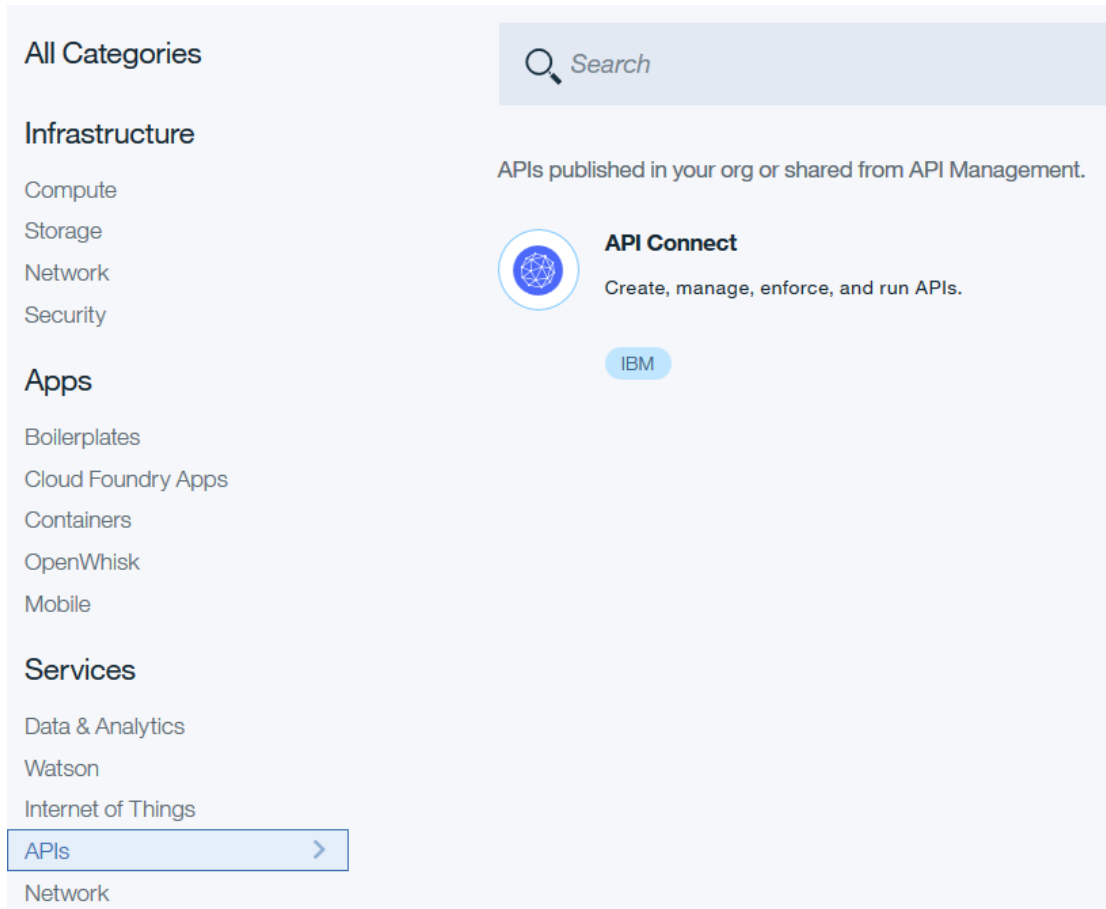
Catalog

Support

Account

Deploy and API Connect

- Under Services, select “**APIs**” link and click on the **API Connect** service.



All Categories

Infrastructure

- Compute
- Storage
- Network
- Security

Apps

- Boilerplates
- Cloud Foundry Apps
- Containers
- OpenWhisk
- Mobile

Services

- Data & Analytics
- Watson
- Internet of Things
- APIs**
- Network

Search

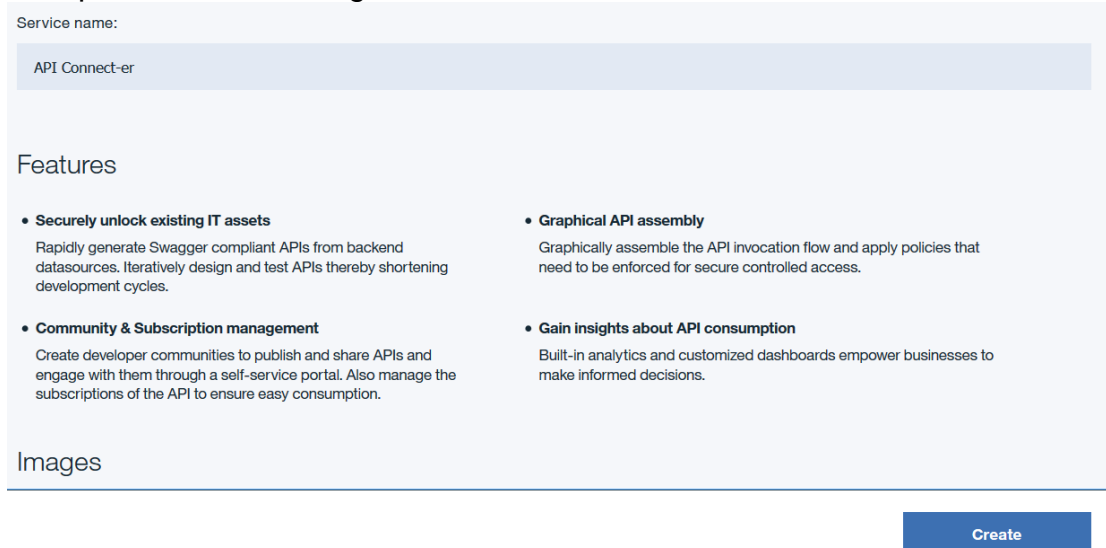
APIs published in your org or shared from API Management.

API Connect

Create, manage, enforce, and run APIs.

IBM

- Accept the default settings and click the “**Create**” button.



Service name:

API Connect-er


Features

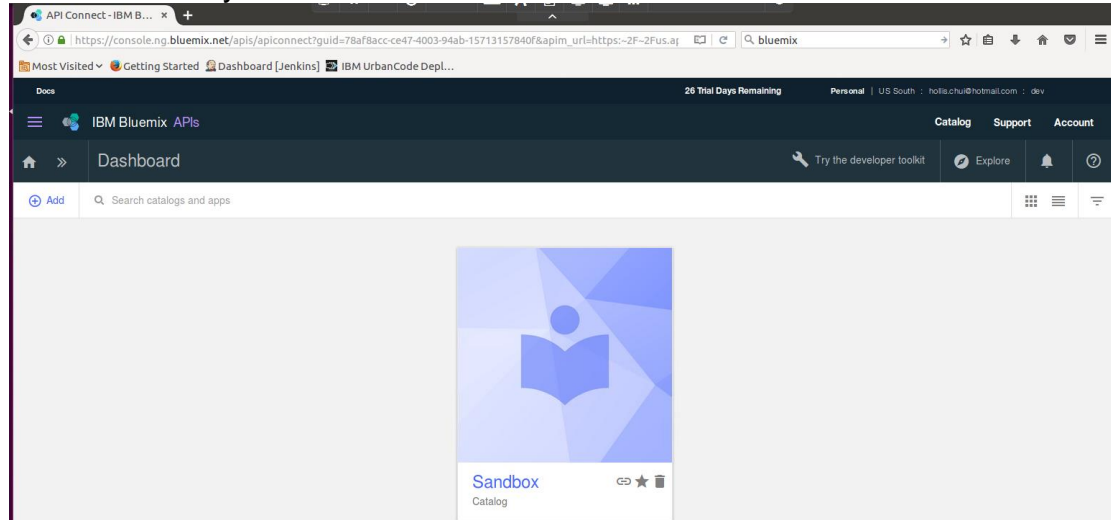
- Securely unlock existing IT assets**
 Rapidly generate Swagger compliant APIs from backend datasources. Iteratively design and test APIs thereby shortening development cycles.
- Graphical API assembly**
 Graphically assemble the API invocation flow and apply policies that need to be enforced for secure controlled access.
- Community & Subscription management**
 Create developer communities to publish and share APIs and engage with them through a self-service portal. Also manage the subscriptions of the API to ensure easy consumption.
- Gain insights about API consumption**
 Built-in analytics and customized dashboards empower businesses to make informed decisions.

Images

Create



6. On the Drafts page, click  and select **“Dashboard”**.
7. The API Connect Dashboard lists the Sandbox Catalog, which has been created by default.



Lab 1.2: Create UAT and PROD Catalogs

1. From the API Connect Dashboard, click Add > Catalog and create a new Catalog with the name “UAT”.

Add Catalog

Display Name *

UAT

Name *

uat

Cancel Add

2. Click Add > Catalog and create a new Catalog with the name “PROD”.

Add Catalog

Display Name *

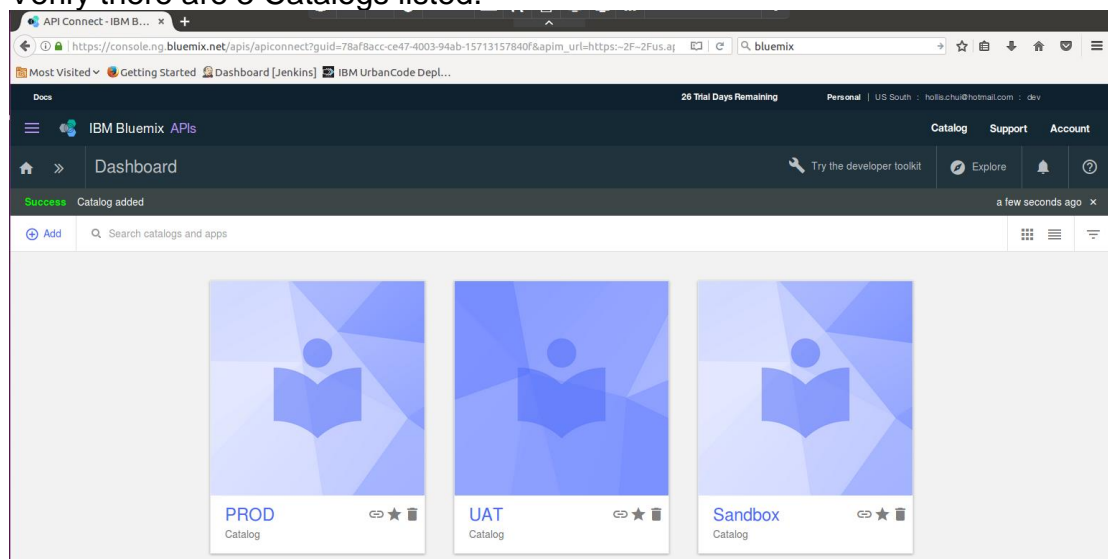
PROD

Name *


prod

Cancel Add

3. Verify there are 3 Catalogs listed.



4. Complete the table below based on the identifier for each catalog

as this information will be used Lab 6.3. Click on the  icon to view the catalog Identifier.

Catalog Identifier

This identifier can be used to configure the 'catalog' configuration variable in the Developer Toolkit

```
apic config:set catalog=apic-catalog://us.apiconnect.ibmcloud.com/orgs/hollischuihotmailcom-dev/catalog
```




The catalog identifier for all catalogs have the following format. The API Connect Org is NOT the same as the Org of your Bluemix account.

apic config:set catalog=apic-catalog://<API Connect Server>/orgs/<API Connect Org>/catalogs/<Catalog>

Environment	Sandbox	UAT	PROD
APIC Connect Server	us.apiconnect.ibmcloud.com	us.apiconnect.ibmcloud.com	us.apiconnect.ibmcloud.com
API Connect Org			
Catalog	sb	uat	prod


Lab 2: GitHub Repository

Lab 2.1: Create your GitHub Repository

1. Launch a web browser and go to <https://github.com>.
2. Click the “**Sign in**” link and log in using the account created in Lab 0.
3. Click the “**Start a project**” button.
4. On the Create a new repository page, enter **acme-bank-api-yaml** and click the “**Create repository**” button.


Create a new repository


A repository contains all the files for your project, including the revision history.

Owner	Repository name
 hollis-chui ▾	/ acme-bank-api-yaml ✓

Great repository names are short and memorable. Need inspiration? How about **cuddly-carnival**.

Description (optional)

☒  **Public**
Anyone can see this repository. You choose who can commit.

☐  **Private**
You choose who can see and commit to this repository.

☐ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▾ | Add a license: **None** ▾ ⓘ

Create repository

Lab 2.2: Clone acme-bank-api-yaml GitHub repository

1. Open a terminal window and change to the /home/devops directory.
2. Create a directory to store your work using the command “**mkdir <Folder name>**”.

Deploy and API Connect

3. Change into the directory and clone the lab repository using the command “**git clone** <https://github.com/hollisc/acme-bank-api-yaml.git>”.

```
devops@devops:~/Hollis$ git clone https://github.com/hollisc/acme-bank-api-yaml.git
git
Cloning into 'acme-bank-api-yaml'...
remote: Counting objects: 471, done.
remote: Compressing objects: 100% (29/29), done.
remote: Total 471 (delta 11), reused 0 (delta 0), pack-reused 441
Receiving objects: 100% (471/471), 49.41 KiB | 0 bytes/s, done.
Resolving deltas: 100% (254/254), done.
Checking connectivity... done.
```

Lab 2.3: Push clone content into your GitHub repository

1. From the terminal window, change into the clone repository directory /home/devops/<Folder name>/acme-bank-api-yaml.
2. Push the contents into the GitHub repository created in Lab 2.1 by running the command: “**git push --mirror** <https://github.com/<GitHub Username>/acme-bank-api-yaml.git>”.
3. Enter your GitHub username and password.

```
devops@devops:~/Hollis/acme-bank-api-yaml$ git push --mirror https://github.com/hollis-chui/acme-bank-api-yaml.git
Username for 'https://github.com': hollis-chui
Password for 'https://hollis-chui@github.com':
Counting objects: 471, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (215/215), done.
Writing objects: 100% (471/471), 49.41 KiB | 0 bytes/s, done.
Total 471 (delta 254), reused 471 (delta 254)
remote: Resolving deltas: 100% (254/254), done.
To https://github.com/hollis-chui/acme-bank-api-yaml.git
* [new branch]      master -> master
* [new branch]      origin/HEAD -> origin/HEAD
* [new branch]      origin/master -> origin/master
```

Lab 2.4: Delete the clone repository directory

1. From the terminal window, change to the /home/devops/<Folder name> directory.
2. Remove the directory by running the command ‘**rm -rf acme-bank-api-yaml**’.

Lab 2.5: Clone your GitHub repository as a local workspace

1. From the terminal window, change to the /home/devops/<Folder name> directory.

Deploy and API Connect

2. Clone your GitHub repository using the command “**git clone** https://github.com/<GitHub_Username>/acme-bank-api-yaml.git”.

```
devops@devops:~/Hollis$ git clone https://github.com/hollis-chui/acme-bank-api-yaml.git
Cloning into 'acme-bank-api-yaml'...
remote: Counting objects: 471, done.
remote: Compressing objects: 100% (215/215), done.
remote: Total 471 (delta 254), reused 471 (delta 254), pack-reused 0
Receiving objects: 100% (471/471), 49.41 KiB | 0 bytes/s, done.
Resolving deltas: 100% (254/254), done.
Checking connectivity... done.
```

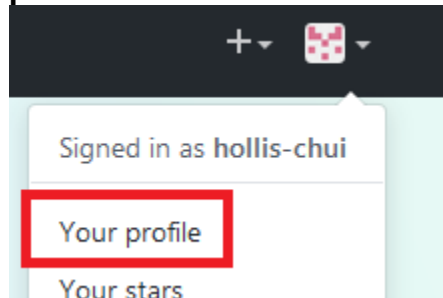
3. Change to the /home/devops/<Folder name>/acme-bank-api-yaml directory and verify the following files exist.

```
devops@devops:~/Hollis/acme-bank-api-yaml$ ls
acme-bank-product.yaml  acme-bank.yaml  build.properties  README.md
```

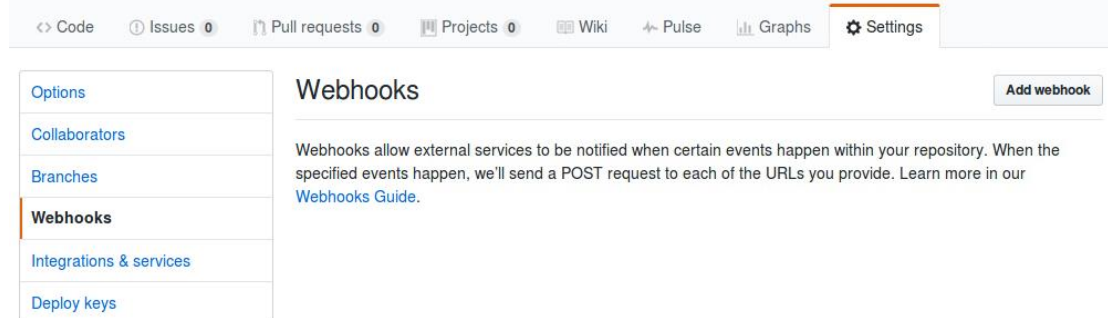
Lab 3: GitHub and Jenkins integration

Lab 3.1: Configure the GitHub and Jenkin integration

1. Launch a web browser and go to <https://github.com>.
2. Click the “**Sign in**” link and log in using the account created in Lab 0.
3. Click the dropdown menu on the top right corner and select “**Your profile**”.



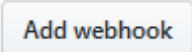
4. Select the Repositories tab and click on the “**acme-bank-api-yaml**” repository. Select the Settings tab and click on “**Webhooks**”.



5. The URL for Jenkins is published as a service from the Skytap VM. This URL is unique for each VM instance. Follow the steps below to find the published service URL for Jenkins on your VM instance. Note, if you are using an on-premise install of Jenkins, simply use the URL to access the Jenkins UI.
 - a) Go to the IBM Cloud Conference Scheduler with the link to launch your VM.
 - b) There is a table listing the internal ports and corresponding external IP and ports. The Jenkins application is using port 8080 on the VM. The corresponding external IP and Port will be used for the Webhook URL.

— Published Services

Internal Port	External IP:External Port
7918	services-uscentral.skytap.com:10768
8080	<u>services-uscentral.skytap.com:10770</u>
8443	<u>services-uscentral.skytap.com:10771</u>

- Go back to your web browser and click the  button.
- In the Payload URL, enter the URL '<http://<External IP:External Port>/github-webhook/>' and click the “**Add webhook**” button. Note, the ending “/” is required in the URL.

Webhooks / Add webhook

We'll send a POST request to the URL below with details of any subscribed events in the format you'd like to receive (JSON, x-www-form-urlencoded, etc). More info [documentation](#).

Payload URL *

Content type

8. Verify that a green checkmark appears beside the added webhook.

Webhooks


Webhooks allow external services to be notified when certain events happen. When events happen, we'll send a POST request to each of the URLs you provide.

✓ <http://services-uscentral.skytap.com:10770/github-webhook/> (push)

Lab 4: IBM UrbanCode Deploy and Jenkins integration






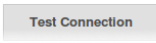
Lab 4.1: Configure the IBM UrbanCode Deploy and Jenkins integration

1. From within the VM, launch the Firefox browser and click on the

Jenkins bookmark  Dashboard [Jenkins].







2. Login to Jenkins using **admin / admin**.
3. Select the “**Manage Jenkins**” link and click on the “**Configure System**” link.
4. Search for “**IBM UrbanCode Deploy Server**” and verify the properties match the following:
 - i) Profile Name = Local UCD Server
 - ii) IBM UrbanCode Deploy URL = <https://devops:8443>
 - iii) User Name = admin
 - iv) Password = admin

IBM UrbanCode Deploy Server

Profile Name	<input type="text" value="Local UCD Server"/>	
IBM UrbanCode Deploy URL	<input type="text" value="https://devops:8443"/>	
User Name	<input type="text" value="admin"/>	
Password	<input type="password" value="....."/>	
		
		

5. Click the “**Test Connection**” button and verify there is a successful connection.

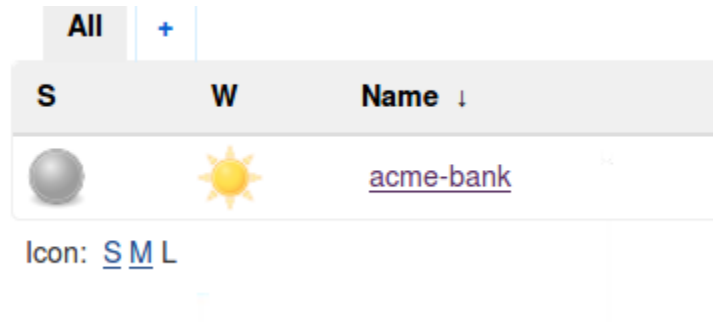
IBM UrbanCode Deploy Server

Profile Name	<input type="text" value="Local UCD Server"/>	
IBM UrbanCode Deploy URL	<input type="text" value="https://devops:8443"/>	
User Name	<input type="text" value="admin"/>	
Password	<input type="password" value="....."/>	
		
Success		

Lab 5: Jenkins Configuration

Lab 5.1: Update Jenkins Job

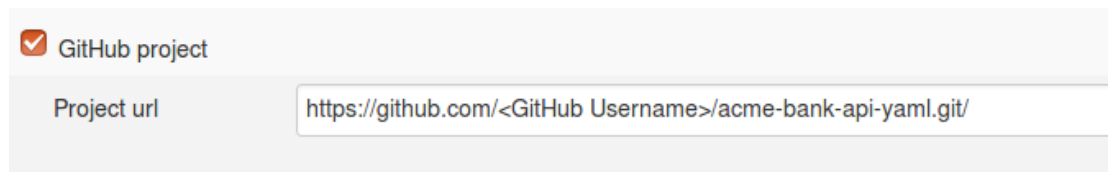
1. From the Firefox browser, bring up the Jenkins Dashboard and click on the “**acme-bank**” link.



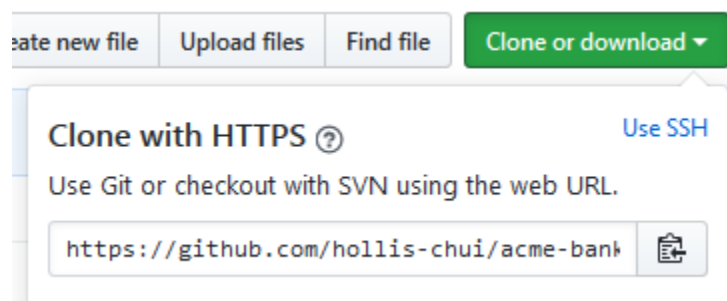
2. Click on the “**Configure**” link on the left menu.



3. On the “**General**” tab, update the “**GitHub project > Project url**” property with your GitHub repository URL in HTTPS format.



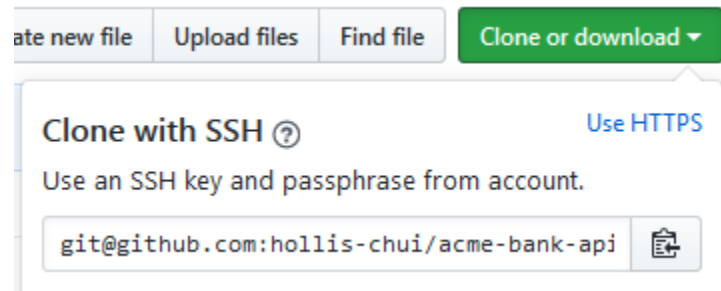
Note: The GitHub repository URL in HTTPS format can be found on the GitHub repository page and clicking on the “**Clone or download**” button. Ensure HTTPS is selected.



4. Click on the “**Source Code Management**” tab and update the Git Repository URL property with your GitHub repository URL in SSH format.

Repository URL `git@github.com:<GitHub Username>/acme-bank-api-yaml.git`

Note: The GitHub repository URL can be found by going to the GitHub repository page and clicking on the “**Clone or download**” button. Ensure SSH is selected.



5. Click the “**Save**” button.

Lab 6: IBM UrbanCode Deploy Configuration**Lab 6.1: Start the IBM UrbanCode Deploy Agent as root**

1. Open a terminal window and type in “**sudo bash**” to run as root. When prompted for the password for the devops user, enter devops.

```
devops@devops:~/Hollis/acme-bank-api-yaml$ sudo bash
[sudo] password for devops:
```

2. Go to the UCD Agent bin directory (**/opt/ibm-ucd/agent/bin**) and run ‘**./agent start**’ to start the agent.

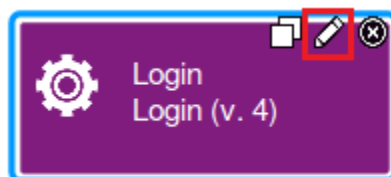
```
root@devops:/opt/ibm-ucd/agent/bin# pwd
/opt/ibm-ucd/agent/bin
root@devops:/opt/ibm-ucd/agent/bin# ./agent start
```

Lab 6.2: Update component process with Bluemix password

1. From within the VM, open a Firefox browser and click on the IBM

UrbanCode Deploy bookmark  IBM UrbanCode Depl...

2. Log in to UCD using **admin / admin**.
3. Select the “**Component**” tab and click on the “**acme-bank**” link.
4. Select the “**Processes**” tab and click on the “**Deploy API Product Def**” link.
5. Edit the “**Login**” step on the Component Process diagram.



6. Update the “**Password**” property with your Bluemix account password.
Note: This value can be parameterized as a Environment property.

Password *

Lab 6.3: Update Environment Properties

1. From the Firefox browser, select the “**Configuration**” tab and verify there are 3 environments listed under acme-bank.

▼ acme-bank
▼ acme-bank
PROD
Sandbox
UAT

2. For the Sandbox environment, update the following properties:
 - i. **API Connect Account Username:** Email address associated with your Bluemix account.
 - ii. **API Connect Server Org:** This value is NOT the Bluemix Org and can be found in the Catalog Identifier. Refer to the table from Step 4 in Lab 1.2. It is in the format <email address without any characters>-<Blueix space>.

Environment Properties

[Use Batch Edit Mode](#)

API Connect Server URL *	us.apiconnect.ibmcloud.com
API Connect Account Username *	<Bluemix account username>
API Connect Toolkit Path	/home/devops/Downloads/node-v6.9.4-linux-x64/bin/
API Connect Server Org *	<APIC Org>
API Connect Catalog *	sb
API Product Definition *	acme-bank-product.yaml

[Save](#)

3. Repeat Step 2 for the UAT and PROD environments. Refer to the table in Step 4 Lab 1.2 for the **API Connect Server Org** values.

Lab 7: Demonstrate API Governance

Lab 7.1: Major change release flow

The development team is ready to publish version 1.0.0 of the acme-bank Product. Upon a push to the GitHub repository, this will trigger the “**acme-bank**” Jenkins job.

The job will create a UCD Component Version for acme-bank:1.0.0 containing the acme-bank Product and API yaml files then call the “**Deploy API Product Def**” process to publish the Product into the Sandbox Catalog.

As this is a major version change, the component process will run the **apic products:publish** cli command. To illustrate the API governance model discussed in the Overview section, we will only be dealing with a single Product and a single API yaml. This lab does not include the API Implementation which can be incorporated into the devops process.

1. Open a terminal window and go to the local GitHub workspace “**/home/devops/<Folder name>**”. The workspace should contain the following files.

```
devops@devops:~/Hollis/acme-bank-api-yaml$ ls
acme-bank-product.yaml acme-bank.yaml build.properties README.md
```

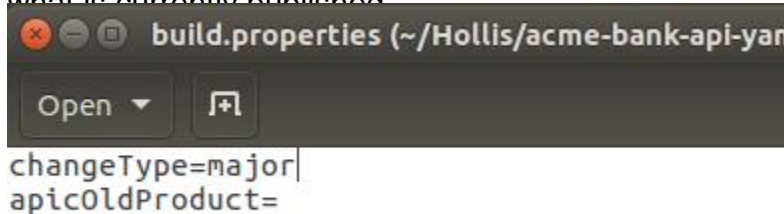
2. Open “**acme-bank-product.yaml**” in an editor and update **version** property from **x.x.x** to **1.0.0**. Save the file and exit the editor.

```
product: 1.0.0
info:
  name: acme-bank
  title: acme-bank
  version: 1.0.0
apis:
  acme-bank:
    $ref: acme-bank.yaml
visibility:
```

3. Open “**acme-bank.yaml**” in an editor and update **version** property from **x.x.x** to **1.0.0**. Save the file and exit the editor.

```
basePath: /api
swagger: '2.0'
info:
  x-ibm-name: acme-bank
  version: 1.0.0
  title: acme-bank
schemes:
  - https
host: $(catalog.host)
consumes:
```

4. Open “**build.properties**” in an editor and for the **changeType** variable, update the value to “**major**” as this is a major version change. The **apicOldProduct** variable will remain blank as this variable is used by the developer to specify the current published Product to replace for bugfix and minor changes going into the UAT and PROD environment and catalog. For the Sandbox environment and catalog, major and minor changes will replace what is currently published.



5. Commit the two yaml files and build.properties to the GitHub repository by running the command “**git commit -a -m <Description>**”.

```
devops@devops:~/Hollis/acme-bank-api-yaml$ git commit -a -m "acme-bank version 1
.0.0 product"
[master 416e3d8] acme-bank version 1.0.0 product
Committer: DevOps <devops@devops.localdomain>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author


3 files changed, 4 insertions(+), 4 deletions(-)
```

6. Push the changes with the tag into the GitHub repository by running the command “**git push origin master**” and provide your GitHub username and password when prompted.

```
devops@devops:~/Hollis/acme-bank-api-yaml$ git push origin master
Username for 'https://github.com': hollis-chui
Password for 'https://hollis-chui@github.com':
Counting objects: 5, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 508 bytes | 0 bytes/s, done.
Total 5 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/hollis-chui/acme-bank-api-yaml.git
79cf618..416e3d8 master -> master
```

7. Once the changes are pushed into the GitHub repository this will trigger a Jenkins build.

The screenshot shows the Jenkins web interface for the 'acme-bank' project. The top navigation bar includes 'Jenkins' and 'acme-bank'. The left sidebar contains various action links: 'Back to Dashboard', 'Status', 'Changes', 'Workspace', 'Build Now', 'Delete Project', 'Configure', 'Schedule Build', 'GitHub Hook Log', 'Move', and 'GitHub'. The main content area is titled 'Project acme-bank' and describes it as a 'Jenkins job for acme-bank lab.'. It includes links to 'Workspace' and 'Recent Changes'. Below this is a 'Permalinks' section. At the bottom, a 'Build History' widget displays a single build from March 12, 2017, at 6:02 PM, with links for 'RSS for all' and 'RSS for failures'.

8. The Jenkins build will create and publish the **acme-bank:1.0.0** component version. Verify that the “**Deploy API Product Def**” component process completed successfully.
 - a. Open the Firefox browser and click on the IBM UrbanCode Deploy bookmark.
 
 - b. Log in using admin / admin.
 - c. In the UCD web client, click on the Components tab and select the **acme-bank** component.

Deploy and API Connect

- d. Under the Component Request History section, find the following entry and click on the **View Request** link.
 - i. Process = Deploy API Product Def
 - ii. Version = acme-bank:1.0.0
 - iii. Environment = Sandbox

Step	Progress	Start Time	Duration	Status
1. Download API YAML Files	100%	9:03:04 PM	0:00:03	Success
2. Set Permissions	100%	9:03:08 PM	0:00:01	Success
3. Login	100%	9:03:10 PM	0:00:25	Success
4. Set Configuration Variable	100%	9:03:36 PM	0:00:06	Success
5. Publish API Product Definitions in Stage	100%	9:03:42 PM	0:00:16	Success
6. API Publish or Replace	100%	9:03:58 PM	0:00:00	Success
7. InFirst	100%	9:03:59 PM	0:00:00	Success
8. envType	100%			Not Started
9. Replace Product in Catalog	100%			Not Started
10. Publish API Product Definitions	100%	9:03:59 PM	0:00:33	Success
11. InVersion	100%			Not Started
12. RemovePreviousVersion	100%			Not Started
13. Logout	100%	9:04:33 PM	0:00:10	Success
Total Execution	1 / 13	9:03:04 PM	0:01:38	Success

- e. Verify the **acme-bank:1.0.0** component version has been deployed into the Sandbox environment by selecting **“Applications”** tab and clicking on the **“acme-bank”** link. Expand the Sandbox environment to view the deployed component version.

Component	Version	Date Deployed	Compliance	Actions
acme-bank	acme-bank:1.0.0 (View Details)	3/12/2017, 9:03 PM	Compliant (100%)	View Request

- f. Finally, validate the **acme-bank:1.0.0** Product is published in the API Connect by logging into the API Connect Dashboard and selecting the Sandbox catalog.


IBM Bluemix APIs	Catalog	Support	Account
Sandbox	Try the developer toolkit	Explore	
Dashboard	Products	Approvals	Community
Members	Analytics	Settings	
Search products			
Title	State		
acme-bank acme-bank:1.0.0	Published		14 minutes ago

9. Once the API has been published to API Connect Sandbox Catalog, teams can perform their testing. Once this is completed, the API Product owner will add the **“release”** Version Status to the **acme-bank:1.0.0** component version and then manually trigger the automated process for deployment into the UAT environment / catalog.
 - a. Open a Firefox browser and click the IBM UrbanCode Deploy bookmark.
 - b. Log in with admin / admin.
 - c. Click on the **“Components”** tab and select the **acme-bank** component.

Deploy and API Connect

- d. Click on the “**Versions**” tab and select the **acme-bank:1.0.0** component version.
- e. On the “**Main**” tab, click the “**Add a Status**” button, select “**release**” Status in the dropdown menu and click Save.

The screenshot shows the IBM UrbanCode Deploy web interface. At the top, there's a navigation bar with tabs: Dashboard, Components, Applications, Configuration, and Processes. Below this is a breadcrumb trail: Home > Components > acme-bank > Versions > Version: acme-bank:1.0.0. The main heading is 'Version: acme-bank:1.0.0' with a '(show details)' link. Below the heading are three tabs: Main, Configuration, and History. The 'Main' tab is active. Under the 'Main' tab, there's a section titled 'Statuses'. It contains a blue button labeled 'Add a Status'. Below the button is a table with two columns: 'Status' and 'Description'. The table has one row with the status 'release' in a green cell. At the bottom of the table, it says '1 record - Refresh Print'.

10. The API Product owner will now proceed to manually trigger the automation process to publish the **acme-bank:1.0.0** Product into the UAT environment / catalog.
 - a. Open a Firefox browser and click the IBM UrbanCode Deploy bookmark.
 
 - b. Log in with admin / admin.
 - c. Click on the “**Applications**” tab and select the **acme-bank** application.
 - d. Click the play button for the UAT environment and the “Run Process on UAT” window should appear.
 - e. For the Process field, select **acme-bank**.
 - f. For the Versions, click on the **Choose Versions** link and a Component Versions window will pop up.
 - g. Under the Versions to Deploy column, select the **Add...** link.
 - h. Select **acme-bank:1.0.0** and click OK and then Submit.

Run Process on UAT

Only Changed Versions ☒

Process * acme-bank

Select a snapshot, or choose versions for individual components.

Component Versions

Select For All... ☐ Show only changed components ☐ Allow invalid versions ☐

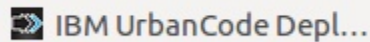
Component	Current Environment Inventory	Versions to Deploy
<input type="text"/>		
acme-bank	None	acme-bank:1.0.0 x Add...

1 record

Rows 10

OK

11. Verify the “**Deploy API Product Def**” component process completed successfully on the UAT environment.
- Open the Firefox browser and click on the IBM UrbanCode Deploy bookmark.



- Log in using admin / admin.
- In the UCD web client, click on the Components tab and select the **acme-bank** component.
- Under the Component Request History section, find the following entry and click on the **View Request** link.
 - Process = Deploy API Product Def
 - Version = acme-bank:1.0.0
 - Environment = UAT

Step	Progress	Start Time	Duration	Status
1. Download API YAML Files	100%	9:22:49 PM	0:00:02	Success
2. Set Permissions	100%	9:22:51 PM	0:00:01	Success
3. Login	100%	9:22:53 PM	0:00:14	Success
4. Set Configuration Variable	100%	9:23:07 PM	0:00:06	Success
5. Publish API Product Definitions in Stage	100%	9:23:13 PM	0:00:21	Success
6. APIC: Publish or Replace	100%	9:23:34 PM	0:00:00	Success
7. isFirst	100%	9:23:34 PM	0:00:00	Success
8. envType	0%			Not Started
9. Replace Product in Catalog	0%			Not Started
10. Publish API Product Definitions	100%	9:23:34 PM	0:00:29	Success
11. isVersion	0%			Not Started
12. RemovePreviousVersion	0%			Not Started
13. Logout	100%	9:24:03 PM	0:00:10	Success
Total Execution	7 / 13	9:22:49 PM	0:01:24	Success

- Verify the **acme-bank:1.0.0** component version has been deployed into the UAT environment by selecting the “**Applications**” tab and clicking on the “**acme-bank**” link. Expand the UAT environment to view the deployed component version.

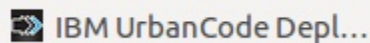
UAT				Snapshot Name		Compliance 1/1	
Component	Version	Date Deployed		Compliance		Actions	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>				
acme-bank	acme-bank-1.0.0 (View Details)	3/12/2017, 9:22 PM		Compliant (1/1)		View Request	
Refresh Print							

- f. Finally, validate the **acme-bank:1.0.0** Product is published on API Connect by logging into the API Connect Dashboard and selecting the UAT catalog.

IBM Bluemix APIs					Catalog	Support	Account
UAT	Try the developer toolkit				Explore		
Dashboard	Products	Approvals	Community	Members	Analytics	Settings	
Search products							
Title	State						
acme-bank acme-bank:1.0.0	Published 3 minutes ago ...						

12. Once testing is complete in the UAT catalog, the API Product Owner will add the “**tested**” Version Status to the **acme-bank:1.0.0** component version and then manually trigger the automated process for deployment into the PROD environment / catalog.

- a. Open a Firefox browser and click the IBM UrbanCode Deploy bookmark.

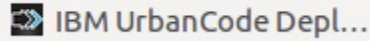


- b. Log in with admin / admin.
 c. Click on the “**Components**” tab and select the **acme-bank** component.
 d. Click on the “**Versions**” tab and select the **acme-bank:1.0.0** component version.
 e. On the Main tab, click the “**Add a Status**” button, select “**tested**” Status in the dropdown menu and click Save.

Home > Components > acme-bank > Versions > Version: acme-bank:1.0.0	
Version: acme-bank:1.0.0 (show details)	
Main	Configuration History
Statuses	
Add a Status	
Status	Description
release	
tested	

Deploy and API Connect

13. The API Product owner will now proceed to manually trigger the automation process to publish the **acme-bank:1.0.0** Product into the PROD environment / catalog.
 - a. Open a Firefox browser and click the IBM UrbanCode Deploy bookmark.



- b. Log in with admin / admin.
- c. Click on the “**Applications**” tab and select the **acme-bank** application.
- d. Click the play button for the PROD environment and the “**Run Process on PROD**” window should appear.
- e. For the Process field, select **acme-bank**.
- f. For the Versions, click on the “**Choose Versions**” link and a Component Versions window will pop up.
- g. Under the Versions to Deploy column, select the “**Add...**” link.
- h. Select **acme-bank:1.0.0** and click OK and then Submit.

14. Verify the “**Deploy API Product Def**” component process completed successfully on the PROD environment.
 - a. Open the Firefox browser and click on the IBM UrbanCode Deploy bookmark.



- b. Log in using admin / admin.
- c. In the UCD web client, click on the “**Components**” tab and select the **acme-bank** component.
- d. Under the Component Request History section, find the following entry and click on the **View Request** link.
 - i. Process = Deploy API Product Def
 - ii. Version = acme-bank:1.0.0

iii. Environment = PROD

Step	Progress	Start Time	Duration	Status
1. Download API YAML Files	100%	9:33:23 PM	0:00:02	Success
2. Set Permissions	100%	9:33:26 PM	0:00:01	Success
3. Login	100%	9:33:27 PM	0:00:14	Success
4. Set Configuration Variable	100%	9:33:42 PM	0:00:05	Success
5. Publish API Product Definitions in Stage	100%	9:33:48 PM	0:00:21	Success
6. APIC Publish or Replace	100%	9:34:09 PM	0:00:00	Success
7. If First	100%	9:34:10 PM	0:00:00	Success
8. envType	100%			Not Started
9. Replace Product in Catalog	100%			Not Started
10. Publish API Product Definitions	100%	9:34:10 PM	0:00:29	Success
11. In Version	100%			Not Started
12. RemovePreviousVersion	100%			Not Started
13. Logout	100%	9:34:40 PM	0:00:10	Success
Total Execution	7 / 9	9:33:23 PM	0:01:28	Success

- e. Verify the **acme-bank:1.0.0** component version has been deployed into the PROD environment by selecting the “**Applications**” tab and clicking on the “**acme-bank**” link. Expand the PROD environment to view the deployed component version.

Component	Version	Date Deployed	Compliance	Actions
acme-bank	acme-bank:1.0.0 (View Details)	3/13/2017, 9:33 PM	Compliant (1/1)	View Request

- f. Finally, validate the **acme-bank:1.0.0** Product is published on API Connect by logging into the API Connect Dashboard and selecting the PROD Catalog.

Title		State
acme-bank	acme-bank:1.0.0	Published 3 minutes ago ...

15. The **acme-bank:1.0.0** Product has now been published to the Sandbox, UAT and PROD Catalogs as well as being deployed to the corresponding UCD Environments.

Lab 7.2: Minor change release flow

A minor version update has been made and version 1.1.0 of the acme-bank API Product definition needs to be released. Upon a push to the GitHub repository, this will trigger the “**acme-bank**” Jenkins job.

The job will create a UCD Component Version for acme-bank:1.1.0 containing the acme-bank Product and API yaml files then call the “**Deploy API Product Def**” process to publish the Product into the Sandbox Catalog.

As this is a minor version change, the component process will run the **apic products:replace** cli command to replace **acme-bank:1.0.0** with the **acme-bank:1.1.0** Product.

Deploy and API Connect

1. Open a terminal window and go to the local GitHub workspace “/home/devops/<Folder name>”.
2. Open “acme-bank-product.yaml” in an editor and update **version** property from **1.0.0** to **1.1.0**. Save the file and exit the editor.

```
product: 1.0.0
info:
  name: acme-bank
  title: acme-bank
  version: 1.1.0
apis:
  acme-bank:
    $ref: acme-bank.yaml
visibility:
```

3. Open “acme-bank.yaml” in an editor and update **version** property from **1.0.0** to **1.1.0**. Save the file and exit the editor.

```
basePath: /api
swagger: '2.0'
info:
  x-ibm-name: acme-bank
  version: 1.1.0
  title: acme-bank
schemes:
  - https
host: ${catalog.host}
consumes:
```

4. Open “build.properties” in an editor and for the **changeType** variable, update the value from **major** to **minor** as this is a minor version change. Update the **apicOldProduct** variable to **acme-bank:1.0.0**.



```
changeType=minor
apicOldProduct=acme-bank:1.0.0
```

5. Commit the two yaml files to the GitHub repository by running the command “git commit -a -m <Description>”.

```
devops@devops:~/Hollis/acme-bank-api-yaml$ git commit -a -m "acme-bank version 1.1.0 product"
[master 82ef001] acme-bank version 1.1.0 product
Committer: DevOps <devops@devops.localdomain>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

2 files changed, 2 insertions(+), 2 deletions(-)
```

6. Push the changes with the tag into the GitHub repository by running the command **"git push origin master"** and provide your GitHub username and password when prompted.

```
devops@devops:~/Hollis/acme-bank-api-yaml$ git push origin master
Username for 'https://github.com': hollis-chui
Password for 'https://hollis-chui@github.com':
Counting objects: 4, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 452 bytes | 0 bytes/s, done.
Total 4 (delta 2), reused 1 (delta 1)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/hollis-chui/acme-bank-api-yaml.git
3f6151b..82ef001 master -> master
```

7. Once the changes are pushed into the GitHub repository this will trigger a Jenkins build.

Jenkins

Jenkins > acme-bank >

[Back to Dashboard](#)

[Status](#)

[Changes](#)

[Workspace](#)

[Build Now](#)

[Delete Project](#)

[Configure](#)

[Schedule Build](#)

[GitHub Hook Log](#)

[Move](#)

[GitHub](#)

[Purge Build History](#)

Project acme-bank

Jenkins job for acme-bank lab.

[Workspace](#)

[Recent Changes](#)

Permalinks

- [Last build \(#2\), 18 sec ago](#)
- [Last stable build \(#1\), 45 min ago](#)
- [Last successful build \(#1\), 45 min ago](#)
- [Last completed build \(#1\), 45 min ago](#)

Build History

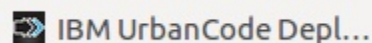
[trend](#)

find

Build	Time	Status
#2	Mar 12, 2017 6:48 PM	Success
#1	Mar 12, 2017 6:02 PM	Success

[RSS for all](#) [RSS for failures](#)

8. The Jenkins build will create and publish the **acme-bank:1.1.0** component version. Verify that the “**Deploy API Product Def**” component process completed successfully.
 - a. Open the Firefox browser and click on the IBM UrbanCode Deploy bookmark.



- b. Log in using admin / admin.
- c. In the UCD web client, click on the Components tab and select the **acme-bank** component.
- d. Under the Component Request History section, find the following entry and click on the **View Request** link.
 - i. Process = Deploy API Product Def
 - ii. Version = acme-bank:1.1.0
 - iii. Environment = Sandbox

Step	Progress	Start Time	Duration	Status
1. Download API YAML Files		1:25:34 PM	0:00:02	Success
2. Set Permissions		1:25:36 PM	0:00:01	Success
3. Login		1:25:38 PM	0:00:14	Success
4. Set Configuration Variable		1:25:53 PM	0:00:06	Success
5. Publish API Product Definitions in Stage		1:25:59 PM	0:00:14	Success
6. APIC Publish or Replace		1:26:14 PM	0:00:00	Success
7. IsFirst				Not Started
8. envType				Not Started
9. Publish API Product Definitions				Not Started
10. Replace Product in Catalog		1:26:14 PM	0:00:26	Success
11. IsVersion		1:26:40 PM	0:00:00	Success
12. RemovePreviousVersion		1:26:40 PM	0:00:07	Success
13. Logout		1:26:48 PM	0:00:11	Success
Total Execution	8 / 9	1:25:34 PM	0:01:25	Success

- e. Verify the **acme-bank:1.1.0** component version has been deployed into the Sandbox environment by selecting “**Applications**” tab and clicking on the “**acme-bank**” link. Expand the Sandbox environment to view the deployed component version.

Component	Version	Date Deployed	Compliance	Actions
acme-bank	acme-bank:1.1.0 (- 1 more)	3/12/2017, 9:48 PM	Compliant (20)	View Request

- f. Finally, validate the **acme-bank:1.0.0** Product is published in the API Connect by logging into the API Connect Dashboard and selecting the Sandbox catalog.

Title		State
acme-bank	acme-bank:1.0.0	Retired a few seconds ago ...
acme-bank	acme-bank:1.1.0	Published 2 minutes ago ...

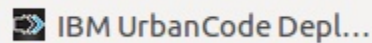
- g. If you notice carefully, there is an additional step that is run called “**RemovePreviousVersion**”. This step will remove the **acme-bank:1.0.0** component version from the Sandbox environment to keep it in sync with Sandbox catalog. Verify this step completed successfully by looking under the Component Request History section and finding the following entry and click on the **View Request** link.
- Process = RemoveFromInventory
 - Version = acme-bank:1.0.0
 - Environment = Sandbox

Step	Progress	Start Time	Duration	Status
1. Remove From Inventory		1:26:45 PM	0:00:01	Success
Total Execution	1 / 1	1:26:45 PM	0:00:01	Success

9. Once the API has been published to API Connect Sandbox Catalog, teams can perform their testing. Once this is completed,

the API Product owner will add the “**release**” Version Status to the **acme-bank:1.1.0** component version and then manually trigger the automated process for deployment into the UAT environment / catalog.

- a. Open a Firefox browser and click the IBM UrbanCode Deploy bookmark.



- b. Log in with admin / admin.
- c. Click on the “**Components**” tab and select the **acme-bank** component.
- d. Click on the “**Versions**” tab and select the **acme-bank:1.1.0** component version.
- e. On the “**Main**” tab, click the “**Add a Status**” button, select “**release**” Status in the dropdown menu and click Save.



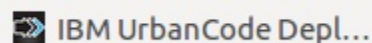
Statuses



Status	Description
release	

10. The API Product owner will now proceed to manually trigger the automation process to publish the **acme-bank:1.1.0** Product into the UAT environment / catalog.

- a. Open a Firefox browser and click the IBM UrbanCode Deploy bookmark.



- b. Log in with admin / admin.
- c. Click on the “**Applications**” tab and select the **acme-bank** application.
- d. Click the play button for the UAT environment and the “Run Process on UAT” window should appear.
- e. For the Process field, select **acme-bank**.
- f. For the Versions, click on the **Choose Versions** link and a Component Versions window will pop up.

Deploy and API Connect

- g. Under the Versions to Deploy column, select the **Add...** link.
- h. Select **acme-bank:1.1.0** and click OK and then Submit.

Run Process on UAT

Component Versions

Select For All... ☐ Show only changed components ☐ Allow invalid versions

Component	Current Environment Inventory	Versions to Deploy
acme-bank	acme-bank:1.0.0	acme-bank:1.1.0 x Add...

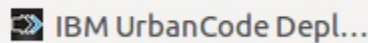
1 record

Rows 10

OK

11. Verify the “Deploy API Product Def” component process completed successfully on the UAT environment.

- a. Open the Firefox browser and click on the IBM UrbanCode Deploy bookmark.



- b. Log in using admin / admin.
- c. In the UCD web client, click on the Components tab and select the **acme-bank** component.
- d. Under the Component Request History section, find the following entry and click on the **View Request** link.
 - i. Process = Deploy API Product Def
 - ii. Version = acme-bank:1.1.0
 - iii. Environment = UAT

Step	Progress	Start Time	Duration	Status
1. Download API YAML Files	100%	10:03:52 PM	0:00:02	Success
2. Set Permissions	100%	10:03:54 PM	0:00:01	Success
3. Login	100%	10:03:56 PM	0:00:15	Success
4. Set Configuration Variable	100%	10:04:11 PM	0:00:05	Success
5. Publish API Product Definitions in Stage	100%	10:04:17 PM	0:00:17	Success
6. APIC Publish or Replace	100%	10:04:34 PM	0:00:00	Success
7. Set First	0%			Not Started
8. Set Type	0%			Not Started
9. Replace Product in Catalog	100%	10:04:35 PM	0:00:29	Success
10. Publish API Product Definitions	0%			Not Started
11. WVersion	100%	10:05:05 PM	0:00:00	Success
12. RemovePreviousVersion	100%	10:05:05 PM	0:00:06	Success
13. Logout	100%	10:05:11 PM	0:00:10	Success
Total Execution	100%	10:05:52 PM	0:01:29	Success

- e. Verify the **acme-bank:1.1.0** component version has been deployed into the UAT environment by selecting the “Applications” tab and clicking on the “acme-bank” link. Expand the UAT environment to view the deployed component

version.

Component	Version	Date Deployed	Compliance	Actions
acme-bank	acme-bank:1.1.0 (View Details)	3/12/2017, 10:03 PM	Compliant (1/1)	View Request

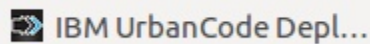
- f. Finally, validate the **acme-bank:1.1.0** Product is published on API Connect by logging into the API Connect Dashboard and selecting the UAT catalog.

IBM Bluemix APIs	Catalog	Support	Account
UAT	Try the developer toolkit	Explore	12 minutes ago
Success acme-bank (version 1.0.0) has been retired from Sandbox			
Dashboard	Products	Approvals	Community
Members	Analytics	Settings	
Search products			
Title	State		
acme-bank acme-bank:1.0.0	Retired	2 minutes ago	...
acme-bank acme-bank:1.1.0	Published	2 minutes ago	...

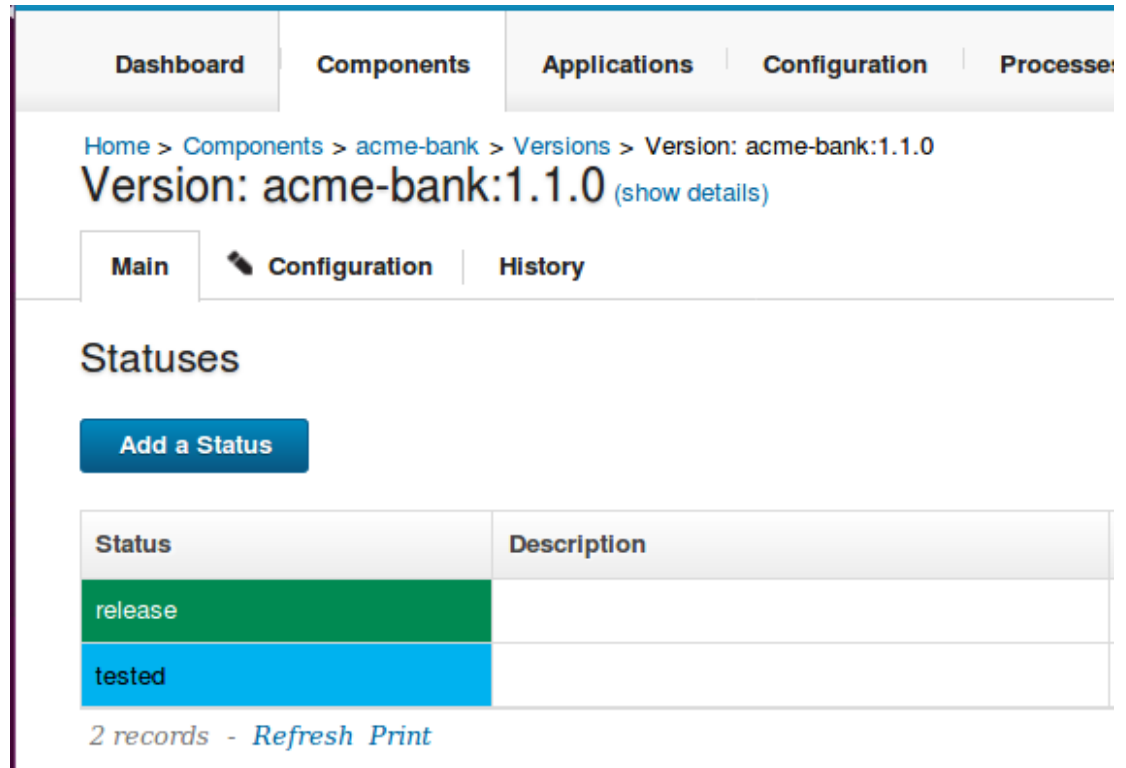
- g. Verify the “**RemovePreviousVersion**” component process completed successfully on the UAT environment and removed the acme-bank:1.0.0 component version. Under the Component Request History section, find the following entry and click on the **View Request** link.
- Process = RemoveFromInventory
 - Version = acme-bank:1.0.0
 - Environment = UAT

Step	Progress	Start Time	Duration	Status
1. Remove From Inventory		10:05:00 PM	0:00:01	Success
Total Execution	1 / 1	10:05:00 PM	0:00:01	Success

12. Once testing is complete in the UAT catalog, the API Product Owner will add the “**tested**” Version Status to the **acme-bank:1.1.0** component version and then manually trigger the automated process for deployment into the PROD environment / catalog.
- Open a Firefox browser and click the IBM UrbanCode Deploy bookmark.



- Log in with admin / admin.
- Click on the “**Components**” tab and select the **acme-bank** component.
- Click on the “**Versions**” tab and select the **acme-bank:1.1.0** component version.
- On the Main tab, click the “**Add a Status**” button, select “**tested**” Status in the dropdown menu and click Save.

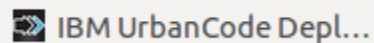


The screenshot shows the IBM UrbanCode Deploy web interface. The top navigation bar includes 'Dashboard', 'Components', 'Applications', 'Configuration', and 'Processes'. The breadcrumb trail is 'Home > Components > acme-bank > Versions > Version: acme-bank:1.1.0'. The main heading is 'Version: acme-bank:1.1.0' with a '(show details)' link. Below this are tabs for 'Main', 'Configuration', and 'History'. The 'Main' tab is active, showing a 'Statuses' section with an 'Add a Status' button. A table lists two statuses: 'release' (green background) and 'tested' (blue background). At the bottom, it says '2 records - Refresh Print'.

Status	Description
release	
tested	

13. The API Product owner will now proceed to manually trigger the automation process to publish the **acme-bank:1.1.0** Product into the PROD environment / catalog.

- a. Open a Firefox browser and click the IBM UrbanCode Deploy bookmark.



- b. Log in with admin / admin.
- c. Click on the “**Applications**” tab and select the **acme-bank** application.
- d. Click the play button for the PROD environment and the “**Run Process on PROD**” window should appear.
- e. For the Process field, select **acme-bank**.
- f. For the Versions, click on the “**Choose Versions**” link and a Component Versions window will pop up.
- g. Under the Versions to Deploy column, select the “**Add...**” link.
- h. Select **acme-bank:1.1.0** and click OK and then Submit.

Run Process on PROD ☒

Component Versions ☒

Select For All... ☐ Show only changed components ? ☐ Allow invalid versions ?

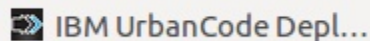
Component	Current Environment Inventory	Versions to Deploy
acme-bank	acme-bank:1.0.0	acme-bank:1.1.0 ✕ Add...

1 record ◀◀ 1 / 1 ▶▶ Rows 10 ▼

OK

14. Verify the “**Deploy API Product Def**” component process completed successfully on the PROD environment.

- Open the Firefox browser and click on the IBM UrbanCode Deploy bookmark.



- Log in using admin / admin.
- In the UCD web client, click on the “**Components**” tab and select the **acme-bank** component.
- Under the Component Request History section, find the following entry and click on the **View Request** link.
 - Process = Deploy API Product Def
 - Version = acme-bank:1.1.0
 - Environment = PROD

Step	Progress	Start Time	Duration	Status
1. Download API YAML Files	100%	10:10:57 PM	0:00:02	Success
2. Set Permissions	100%	10:10:59 PM	0:00:01	Success
3. Login	100%	10:11:01 PM	0:00:13	Success
4. Set Configuration Variable	100%	10:11:14 PM	0:00:05	Success
5. Publish API Product Definitions in Stage	100%	10:11:20 PM	0:00:20	Success
6. API: Publish or Replace	100%	10:11:40 PM	0:00:00	Success
7. If First	0%			Not Started
8. envType	0%			Not Started
9. Replace Product in Catalog	100%	10:11:41 PM	0:00:33	Success
10. Publish API Product Definitions	0%			Not Started
11. WVersion	100%	10:12:14 PM	0:00:00	Success
12. RemovePreviousVersion	100%	10:12:14 PM	0:00:05	Success
13. Logout	100%	10:12:20 PM	0:00:10	Success
Total Execution	100%	10:10:57 PM	0:01:33	Success

- Verify the **acme-bank:1.1.0** component version has been deployed into the PROD environment by selecting the “**Applications**” tab and clicking on the “**acme-bank**” link. Expand the PROD environment to view the deployed

component version.

Component	Version	Date Deployed	Compliance	Actions
acme-bank	acme-bank:1.1.0 (View Details)	3/12/2017, 10:10 PM	Compliant (1/1)	View Request

- f. Finally, validate the **acme-bank:1.1.0** Product is published on API Connect by logging into the API Connect Dashboard and selecting the PROD Catalog.

Title	State	Actions
acme-bank acme-bank:1.0.0	Retired 2 minutes ago	...
acme-bank acme-bank:1.1.0	Published 2 minutes ago	View Request

- g. Verify the “**RemovePreviousVersion**” component process completed successfully on the PROD environment and removed the acme-bank:1.0.0 component version. Under the Component Request History section, find the following entry and click on the **View Request** link.

- i. Process = RemoveFromInventory
- ii. Version = acme-bank:1.0.0
- iii. Environment = PROD

PROCESS	COMPONENT	VERSION	ENVIRONMENT	COMPLETION TIME	STATUS	ACTIONS
RemoveFromInventory	acme-bank / PROD / devops_local_agent / acme-bank	acme-bank:1.0.0	acme-bank	PROD	3/12/2017, 10:12 PM	Success View Request

- h. The **acme-bank:1.1.0** Product has now been published to the Sandbox, UAT and PROD Catalogs as well as being deployed to the corresponding UCD Environments.

Lab 7.3: Bugfix change release flow

A bugfix has been made and version 1.1.1 of the acme-bank API Product definition will need to be released. Upon a push to the GitHub repository, this will trigger the “**acme-bank**” Jenkins job.

The job will create a UCD Component Version for acme-bank:1.1.1 containing the acme-bank Product and API yaml files, add the “**release**” version status to the component version and then call the “**Deploy API Product Def**” process to publish the Product into the Sandbox Catalog.

As this is a bugfix change, the component process will run the **apic products:replace** cli command to replace **acme-bank:1.1.0** with the **acme-bank:1.1.1** Product in the UAT environment. Bugfix changes are NOT deployed to the Sandbox environment as per the API Governance model. .

1. Open a terminal window and go to the local GitHub workspace “**/home/devops/<Folder name>/**”.

Deploy and API Connect

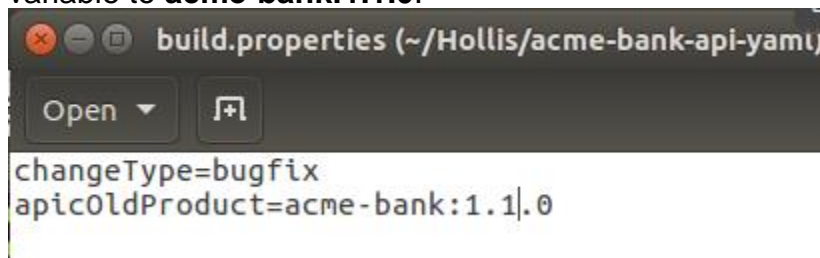
- Open “**acme-bank-product.yaml**” in an editor and update **version** property from **1.1.0** to **1.1.1**. Save the file and exit the editor.

```
product: 1.0.0
info:
  name: acme-bank
  title: acme-bank
  version: 1.1.1|
apis:
  acme-bank:
    $ref: acme-bank.yaml
visibility:
```

- Open “**acme-bank.yaml**” in an editor and update **version** property from **1.0.0** to **1.1.0**. Save the file and exit the editor.

```
basePath: /api
swagger: '2.0'
info:
  x-ibm-name: acme-bank
  version: 1.1.1|
  title: acme-bank
schemes:
  - https
host: $(catalog.host)
consumes:
  - application/json
```

- Open “**build.properties**” in an editor and for the **changeType** variable, update the value from **minor** to **bugfix**. Update the **apicOldProduct** variable to **acme-bank:1.1.0**.



```
changeType=bugfix
apicOldProduct=acme-bank:1.1.0
```

- Commit the two yaml files to the GitHub repository by running the command “**git commit -a -m <Description>**”.


```
git config --global --edit

After doing this, you may fix the identity used for this commit with:

git commit --amend --reset-author

3 files changed, 4 insertions(+), 4 deletions(-)
```

6. Push the changes with the tag into the GitHub repository by running the command “**git push origin master**” and provide your GitHub username and password when prompted.

```
devops@devops:~/Hollis/acme-bank-api-yaml$ git push origin master
Username for 'https://github.com': hollis-chui
Password for 'https://hollis-chui@github.com':
Counting objects: 5, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 510 bytes | 0 bytes/s, done.
Total 5 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/hollis-chui/acme-bank-api-yaml.git
82ef001..53be3e6 master -> master
```

7. Once the changes are pushed into the GitHub repository this will trigger a Jenkins build. The Jenkins build will create and publish the **acme-bank:1.1.1** component version into the UAT environment / catalog. As per the API governance model, bugfixes are not released into the Sandbox environment / catalog. Verify that the “**Deploy API Product Def**” component process completed successfully.

- a. Open the Firefox browser and click on the IBM UrbanCode Deploy bookmark.



- b. Log in using admin / admin.
- c. In the UCD web client, click on the Components tab and select the **acme-bank** component.
- d. Under the Component Request History section, find the following entry and click on the **View Request** link.
 - i. Process = Deploy API Product Def
 - ii. Version = acme-bank:1.1.1
 - iii. Environment = UAT

12. RemovePreviousVersion	10:37:00 PM	0:00:05	Success
13. Logout	10:37:06 PM	0:00:10	Success
Total Execution	10:38:39 PM	0:01:37	Success

- e. Verify the **acme-bank:1.1.1** component version has been deployed into the UAT environment by selecting the “**Applications**” tab and clicking on the “**acme-bank**” link. Expand the UAT environment to view the deployed component version.

Component	Version	Date Deployed	Compliance	Actions
acme-bank	acme-bank:1.1.1 (View Details)	3/12/2017, 10:35 PM	Compliant (1/1)	View Request

- f. Finally, validate the **acme-bank:1.1.1** Product is published on API Connect by logging into the API Connect Dashboard and selecting the UAT catalog.

Title		State	
acme-bank	acme-bank:1.0.0	Retired	33 minutes ago
acme-bank	acme-bank:1.1.0	Retired	a minute ago
acme-bank	acme-bank:1.1.1	Published	a minute ago

- g. Verify the “**RemovePreviousVersion**” component process completed successfully on the UAT environment and removed the acme-bank:1.1.0 component version. Under the Component Request History section, find the following entry and click on the **View Request** link.
- Process = RemoveFromInventory
 - Version = acme-bank:1.1.0
 - Environment = UAT

Step	Progress	Start Time	Duration	Status
1. Remove From Inventory	1	10:37:03 PM	0:00:01	Success
Total Execution	1 / 1	10:37:03 PM	0:00:01	Success

8. Once testing is complete in the UAT catalog, the API Product Owner will add the “**tested**” Version Status to the **acme-bank:1.1.1** component version and then manually trigger the automated process for deployment into the PROD environment / catalog.

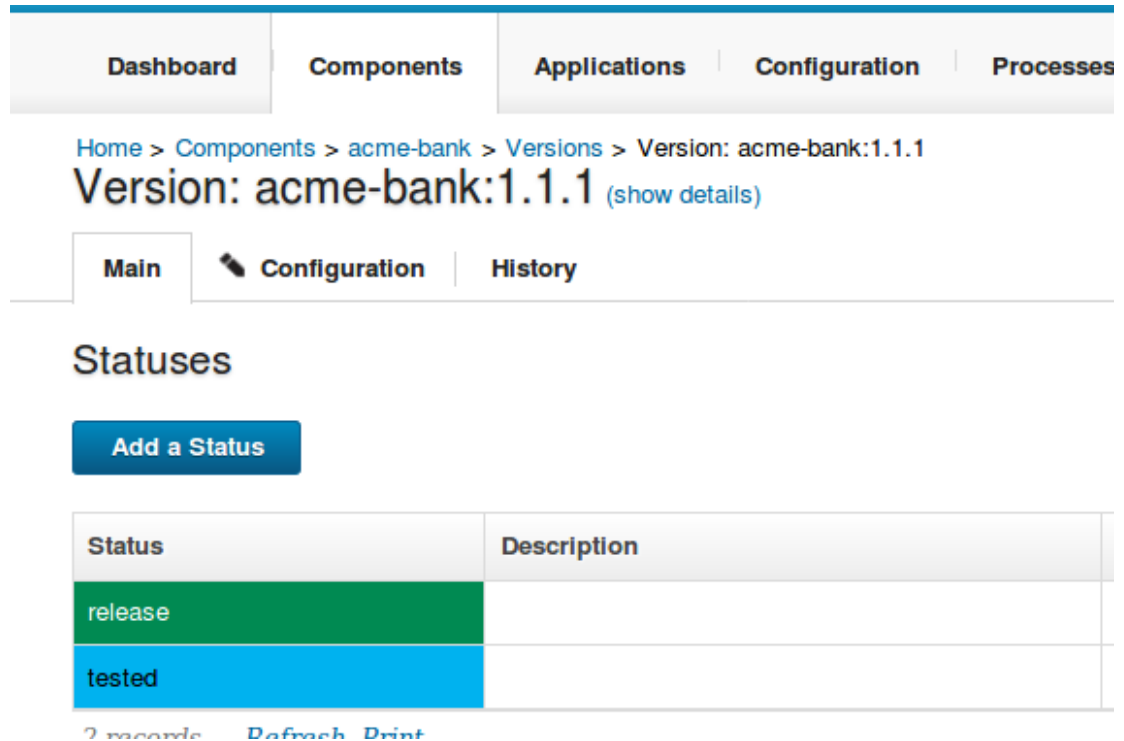
- a. Open a Firefox browser and click the IBM UrbanCode Deploy bookmark.



- b. Log in with admin / admin.
c. Click on the “**Components**” tab and select the **acme-bank** component.

Deploy and API Connect


- d. Click on the “**Versions**” tab and select the **acme-bank:1.1.1** component version.
- e. On the Main tab, click the “**Add a Status**” button, select “**tested**” Status in the dropdown menu and click Save.



The screenshot shows the IBM UrbanCode Deploy web interface. At the top, there are navigation tabs: Dashboard, Components, Applications, Configuration, and Processes. Below these, a breadcrumb trail reads: Home > Components > acme-bank > Versions > Version: acme-bank:1.1.1. The main heading is 'Version: acme-bank:1.1.1' with a '(show details)' link. Below this is a sub-navigation bar with 'Main', 'Configuration', and 'History' tabs. The 'Main' tab is active, showing a 'Statuses' section with an 'Add a Status' button. Below the button is a table with two columns: 'Status' and 'Description'. The table contains two rows: 'release' (green background) and 'tested' (blue background). At the bottom of the table, it says '2 records' followed by 'Refresh' and 'Print' links.

Status	Description
release	
tested	

2 records Refresh Print

9. The API Product owner will now proceed to manually trigger the automation process to publish the **acme-bank:1.1.1** Product into the PROD environment / catalog.
 - a. Open a Firefox browser and click the IBM UrbanCode Deploy bookmark.
 
 - b. Log in with admin / admin.
 - c. Click on the “**Applications**” tab and select the **acme-bank** application.
 - d. Click the play button for the PROD environment and the “**Run Process on PROD**” window should appear.
 - e. For the Process field, select **acme-bank**.
 - f. For the Versions, click on the “**Choose Versions**” link and a Component Versions window will pop up.
 - g. Under the Versions to Deploy column, select the “**Add...**” link.
 - h. Select **acme-bank:1.1.1** and click OK and then Submit.

Run Process on PROD ☒ ents by their names to re-

Component Versions ☒

Select For All... ☐ Show only changed components ☐ Allow invalid versions

Component	Current Environment Inventory	Versions to Deploy
acme-bank	acme-bank:1.1.0	acme-bank:1.1.1 <input checked="" type="checkbox"/> Add...

1 record 1 / 1 Rows 10

OK

10. Verify the “**Deploy API Product Def**” component process completed successfully on the PROD environment.

- Open the Firefox browser and click on the IBM UrbanCode Deploy bookmark.



- Log in using admin / admin.
- In the UCD web client, click on the “**Components**” tab and select the **acme-bank** component.
- Under the Component Request History section, find the following entry and click on the **View Request** link.
 - Process = Deploy API Product Def
 - Version = acme-bank:1.1.1
 - Environment = PROD

Step	Progress	Start Time	Duration	Status
1. Download API YAML Files	<div></div>	10:41:50 PM	0:00:02	Success
2. Set Permissions	<div></div>	10:41:52 PM	0:00:01	Success
3. Login	<div></div>	10:41:53 PM	0:00:15	Success
4. Set Configuration Variable	<div></div>	10:42:09 PM	0:00:06	Success
5. Publish API Product Definitions in Stage	<div></div>	10:42:15 PM	0:00:19	Success
6. APIC Publish or Replace	<div></div>	10:42:34 PM	0:00:00	Success
7. Init first	<div></div>			Not Started
8. envType	<div></div>			Not Started
9. Replace Product in Catalog	<div></div>	10:42:35 PM	0:00:31	Success
10. Publish API Product Definitions	<div></div>			Not Started
11. InVersion	<div></div>	10:43:06 PM	0:00:00	Success
12. RemovePreviousVersion	<div></div>	10:43:06 PM	0:00:05	Success
13. Logout	<div></div>	10:43:12 PM	0:00:09	Success
Total Execution	<div></div>	10:41:50 PM	0:01:31	Success

- Verify the **acme-bank:1.1.1** component version has been deployed into the PROD environment by selecting the “**Applications**” tab and clicking on the “**acme-bank**” link. Expand the PROD environment to view the deployed component version.

acme-bank	acme-bank:1.1.1 (View Details)	3/12/2017, 10:41 PM	Compliant (1/1)	View Request
Refresh	Print			

- f. Finally, validate the **acme-bank:1.1.1** Product is published on API Connect by logging into the API Connect Dashboard and selecting the PROD Catalog.

IBM Bluemix APIs		Catalog	Support	Account
Home	»	PROD	Try the developer toolkit	Explore
Success acme-bank (version 1.0.0) has been retired from Sandbox an hour ago				
Dashboard	Products	Approvals	Community	Members
Search products				
Title				State
acme-bank acme-bank:1.0.0				Retired 34 minutes ago
acme-bank acme-bank:1.1.0				Retired 3 minutes ago
acme-bank acme-bank:1.1.1				Published 3 minutes ago

- g. Verify the “**RemovePreviousVersion**” component process completed successfully on the PROD environment and removed the acme-bank:1.1.0 component version. Under the Component Request History section, find the following entry and click on the **View Request** link.
- Process = RemoveFromInventory
 - Version = acme-bank:1.1.0
 - Environment = PROD

Step	Progress	Start Time	Duration	Status
1. Remove From Inventory		10:43:10 PM	0:00:01	Success
Total Execution	1 / 1	10:43:10 PM	0:00:01	Success

- h. The **acme-bank:1.1.1** Product has now been published to the UAT and PROD Catalogs as well as being deployed to the corresponding UCD Environments.

Lab 7.4: (Optional) Publish a major version release

- As an optional exercise, publish a major version change (**acme-bank:2.0.0**) and deploy it to the Sandbox, UAT and PROD environments / catalogs. Notice that the automation process will replace **acme-bank:1.1.0** in Sandbox environment / catalog as Sandbox will only have one release at any time. For the UAT and PROD environment / catalogs, **acme-bank:2.0.0** will be published alongside acme-bank:1.1.1 and both Products will be published. In the build.properties, set **changeType = major** and **apicOldProduct = acme-bank:1.1.1**.

You have completed all the lab exercises.