



# 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 Abd El Aziz, 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](http://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

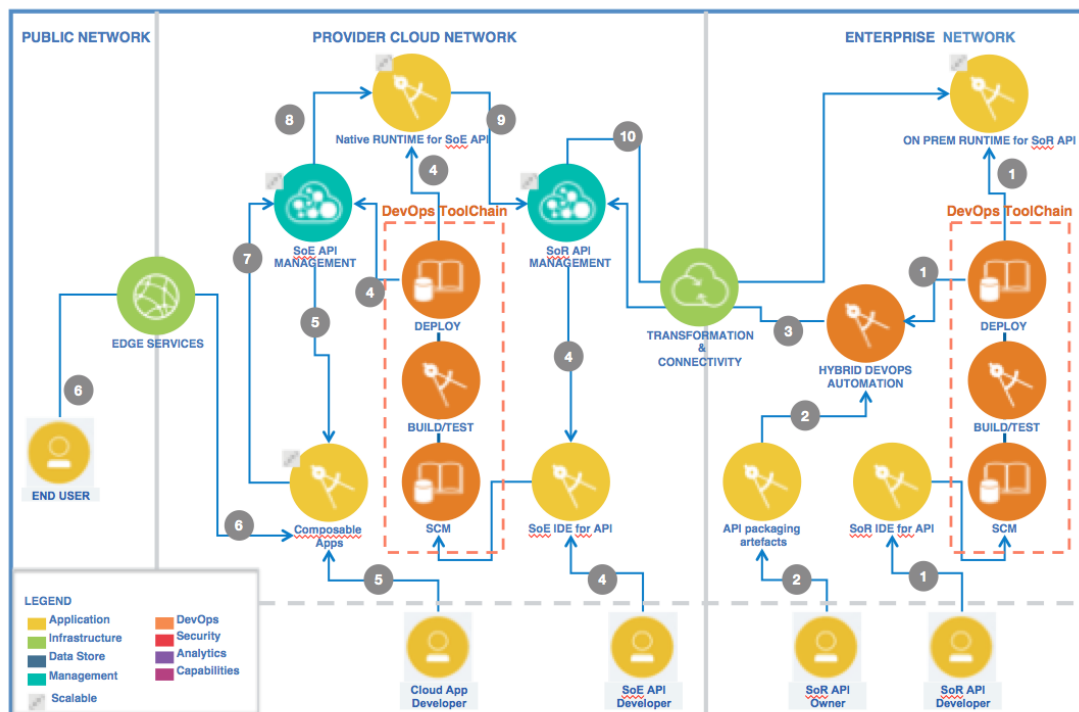
<b>Introduction</b>	4
<b>Business Scenario</b>	4
<b>Lab Overview</b>	6
<b>What is Already Completed</b>	8
<b>Lab Exercises</b>	12
Lab 0: Create Bluemix and GitHub Accounts	12
Task 1: Create Bluemix trial account	12
Task 2: Create GitHub account	12
Lab 1: Bluemix API Connect service and Catalogs	13
Lab 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 a GitHub Repository	18
Lab 2.2: Clone GitHub repository	18
Lab 2.3: Push clone content into GitHub repository	19
Lab 2.4: Delete the clone repository directory	19
Lab 2.5: Clone the 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	23
Lab 4.1: Configure the IBM UrbanCode Deploy and Jenkins integration	23
Lab 5: Jenkins Configuration	24
Lab 5.1: Update Jenkins Job	24
Lab 6: IBM UrbanCode Deploy Configuration	26
Lab 6.1: Start the IBM UrbanCode Deploy Agent as root	26
Lab 6.2: Update component process with Bluemix password	26
Lab 6.3: Update Environment Properties	26
Lab 7: Demonstrate API Governance	28
Lab 7.1: Major change release flow	28
Lab 7.2: Minor change release flow	36
Lab 7.3: Bugfix change release flow	46
Lab 7.4: (Optional) Publish a major version release	52

## 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.

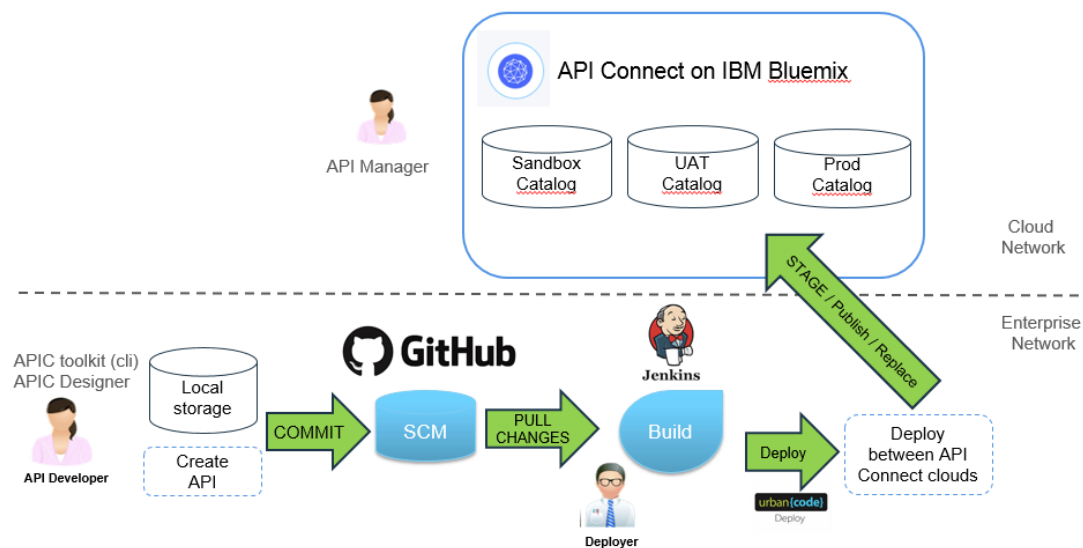
## Lab Overview

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.

<b>API Connect</b>	<b>IBM UrbanCode Deploy</b>	<b>Description</b>
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	One UCD component represents one API Product within API Connect. The component stores the version artifacts that represent the version of the API Product. The API Product contains the definition of one or more APIs.
API Implementation	Component(s)	A UCD Component represents an API Implementation. Each API defined in the Product has a corresponding implementation. The source code that provides the implementation of the API may be from a variety of technologies such as StrongLoop Node.js, java, javascript or other languages. Each API implementation is represented by a UCD component. This component stores the version artifacts that results from the build of the API's implementation source code.

## What is Already Completed

### 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:

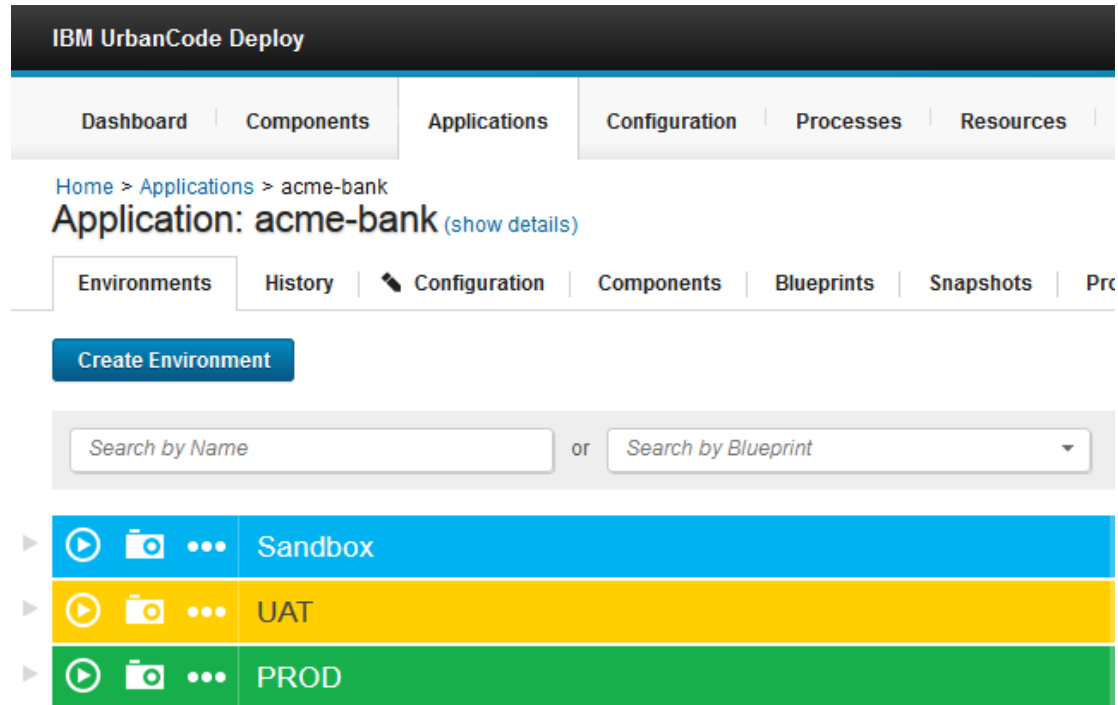
The **acme-bank** job has been configured to run a build when a change is pushed to a specified GitHub repository. Once the build request is triggered, the job will perform the following activities:

- 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.

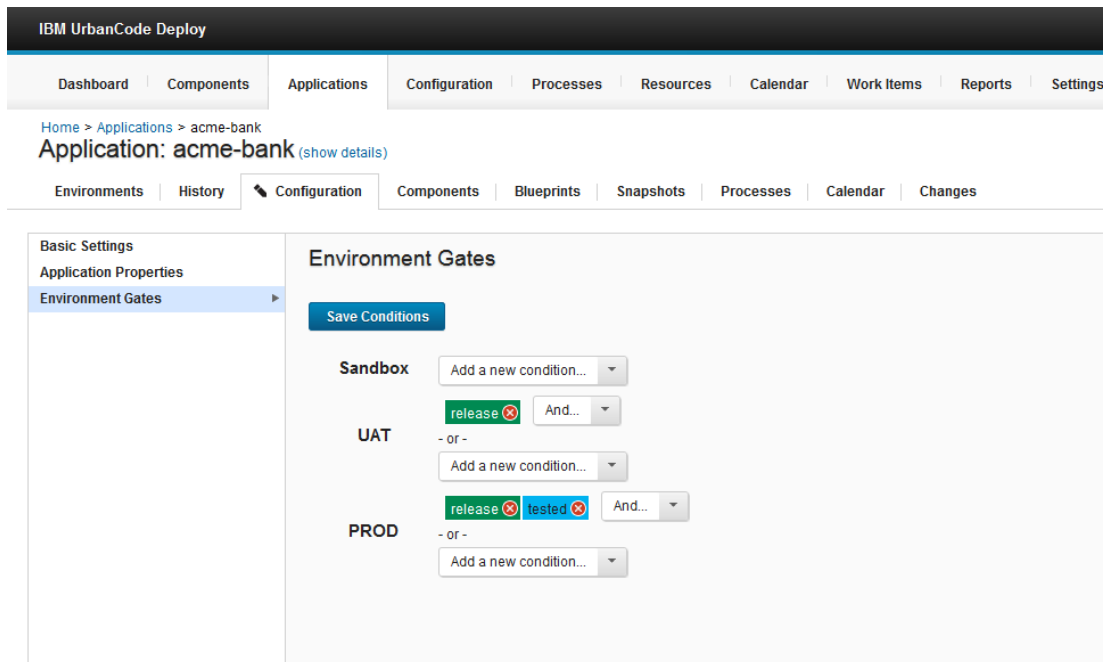




The screenshot shows the IBM UrbanCode Deploy web interface. At the top is a dark header with the text "IBM UrbanCode Deploy". Below this is a navigation bar with tabs: "Dashboard", "Components", "Applications", "Configuration", "Processes", and "Resources". The "Applications" tab is selected. Below the navigation bar, the breadcrumb "Home > Applications > acme-bank" is visible, followed by the heading "Application: acme-bank" with a "(show details)" link. A secondary navigation bar includes "Environments", "History", "Configuration" (with a gear icon), "Components", "Blueprints", "Snapshots", and "Prc". Below this is a blue "Create Environment" button. A search section contains two input fields: "Search by Name" and "Search by Blueprint". Below the search fields is a list of three environments, each with a play icon, a camera icon, a menu icon, and a name: "Sandbox" (blue background), "UAT" (yellow background), and "PROD" (green background).

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.

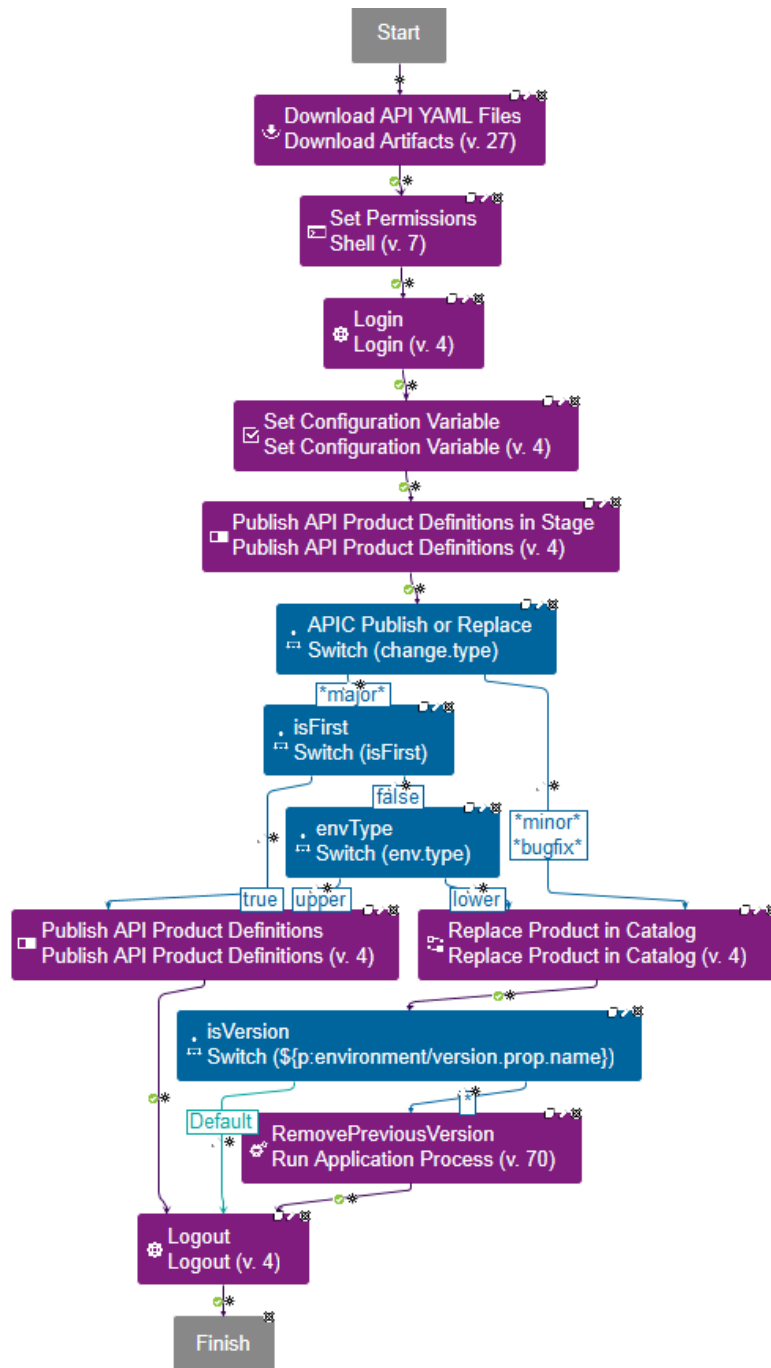
As seen in the diagram below, any deployments can be deployed to the Sandbox environment. A deployment must be tagged with a “**release**” status for it to be deployed into the UAT environment. Finally, a deployment must be tagged with both the “**release**” and “**tested**” statuses to be deployed into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



The screenshot displays the IBM UrbanCode Deploy web interface. The top navigation bar includes links for Dashboard, Components, Applications, Configuration, Processes, Resources, Calendar, Work Items, Reports, and Settings. The breadcrumb trail shows 'Home > Applications > acme-bank'. The main header indicates 'Application: acme-bank' with a '(show details)' link. Below this, a secondary navigation bar lists 'Environments', 'History', 'Configuration' (selected), 'Components', 'Blueprints', 'Snapshots', 'Processes', 'Calendar', and 'Changes'. On the left, a sidebar menu shows '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 set of conditions: 'Sandbox' has 'Add a new condition...'; 'UAT' has 'release' (with a red 'x') and 'And...'; 'PROD' has 'release' (with a red 'x'), 'tested' (with a red 'x'), and 'And...'. Each environment also has an 'Add a new condition...' button.

### 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

#### Task 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.

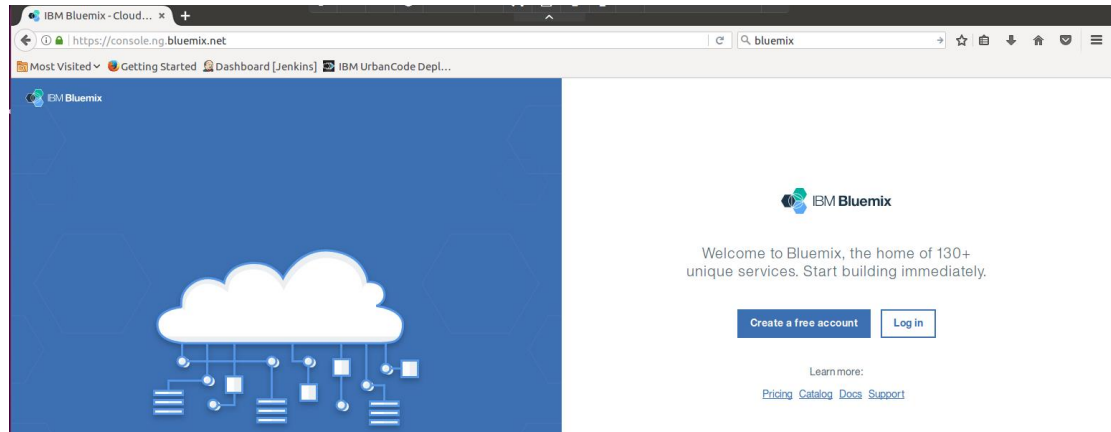
#### Task 2: Create GitHub account

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.

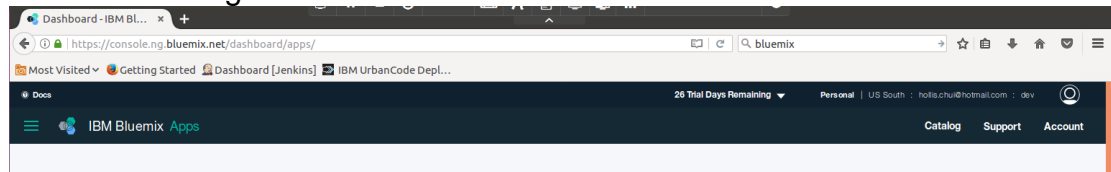
## Lab 1: Bluemix API Connect service and Catalogs

### Lab 1.1: Create an instance of the API Connect Service

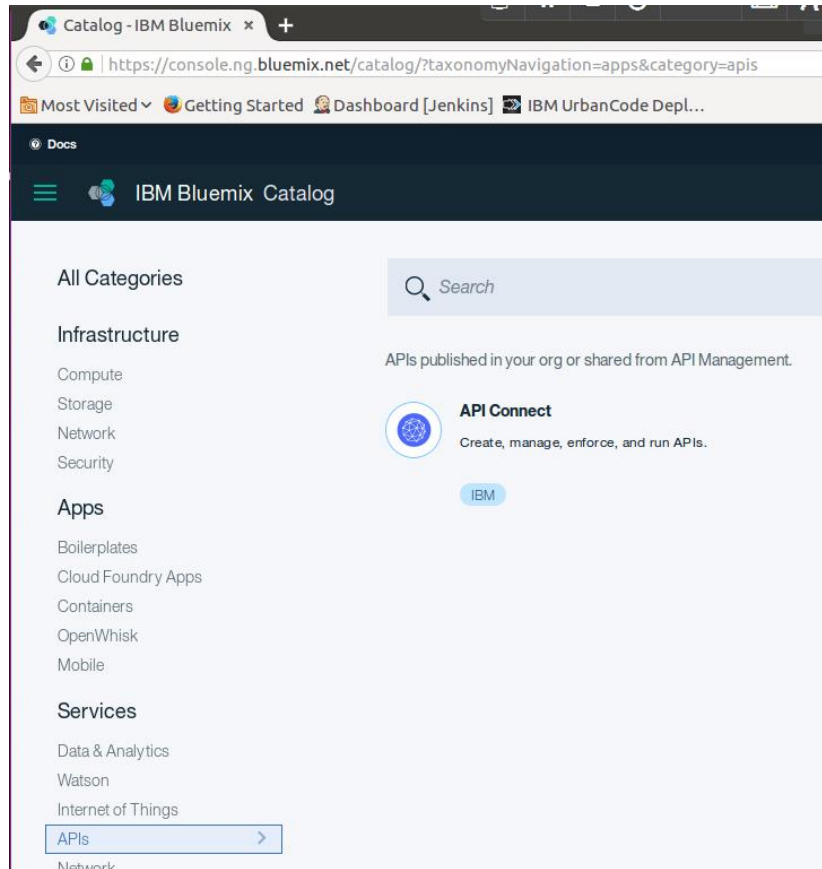
1. Open the Firefox browser, go to the Bluemix website (<https://console.ng.bluemix.net/>) and click the \*\*Log In\*\* button.



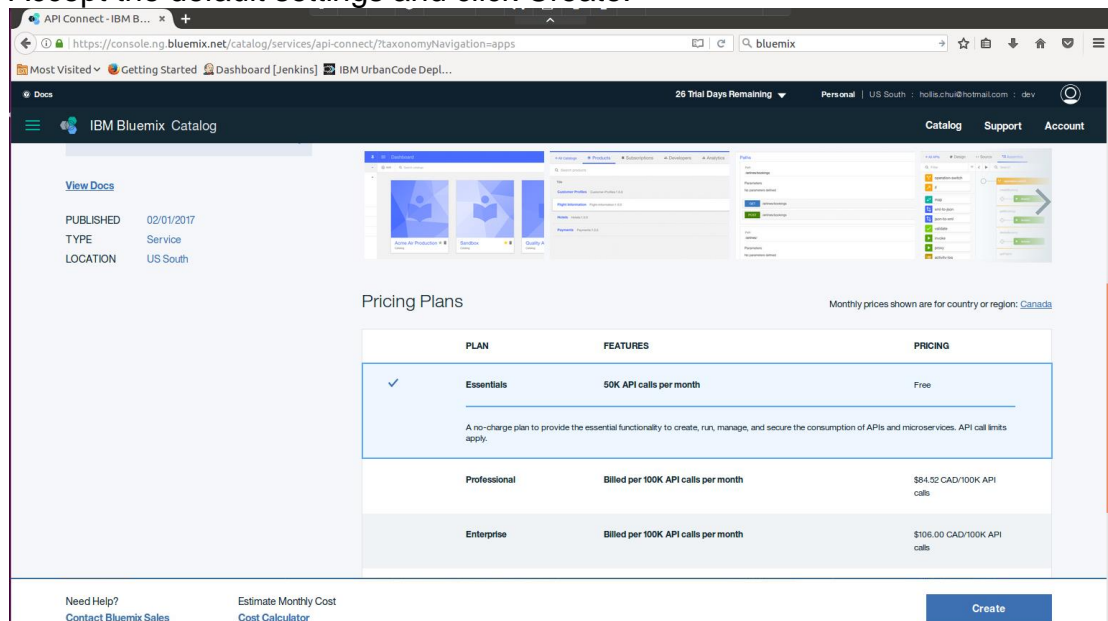
2. Click on Catalog.



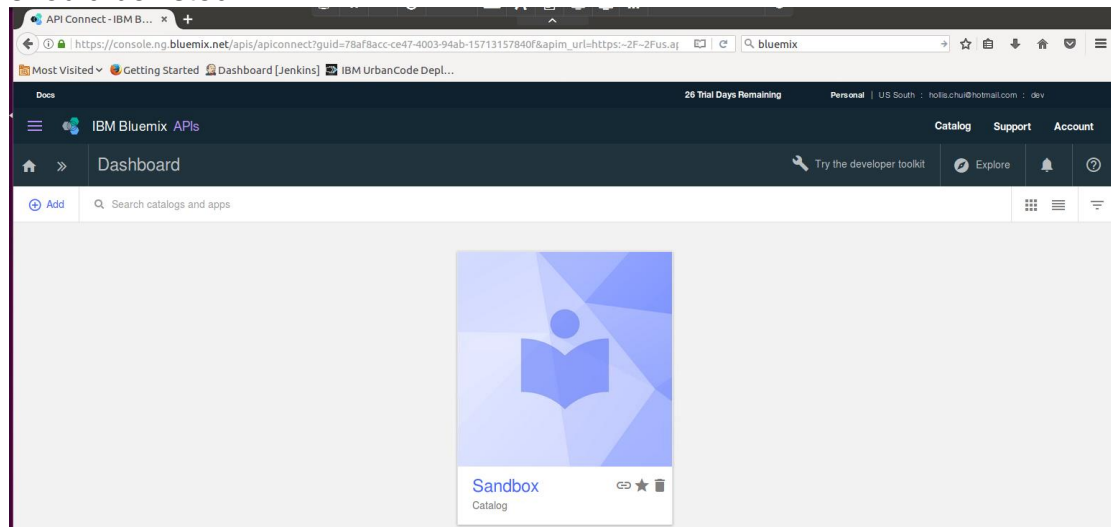
## 3. Select Services &gt; APIs and click API Connect.



## 4. Accept the default settings and click Create.



5. Go to the API Connect Dashboard and a default Sandbox Catalog should be listed.



### 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".

A screenshot of the 'Add Catalog' form. The title 'Add Catalog' is at the top. Below it, there are two input fields. The first is labeled 'Display Name \*' and contains the text 'UAT|'. The second is labeled 'Name \*' and contains the text 'uat'. At the bottom right of the form are two buttons: 'Cancel' and 'Add'.

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

## Add Catalog

Display Name \*

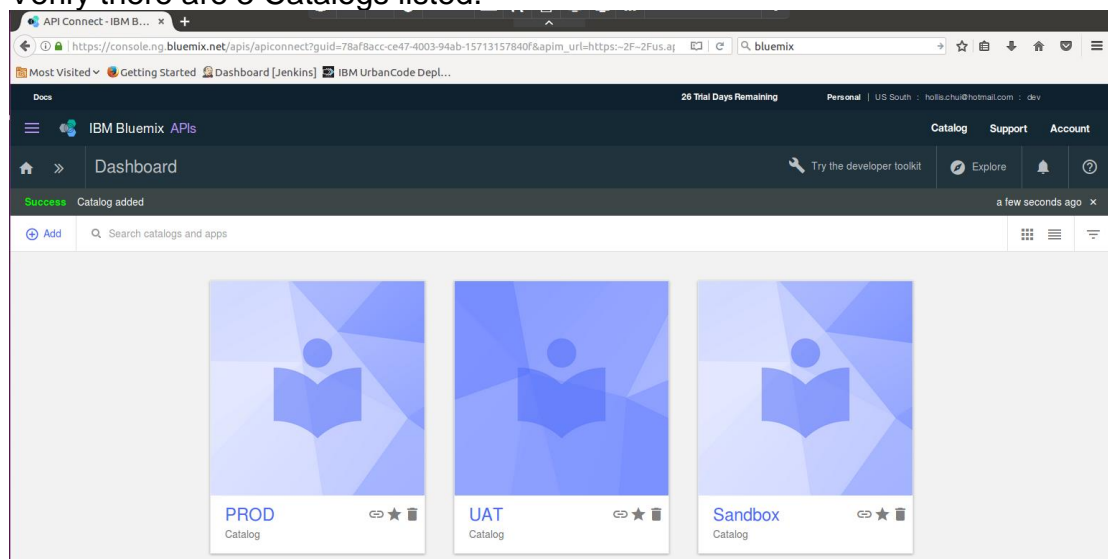
PROD

Name \*

prod

Cancel Add

### 3. Verify there are 3 Catalogs listed.

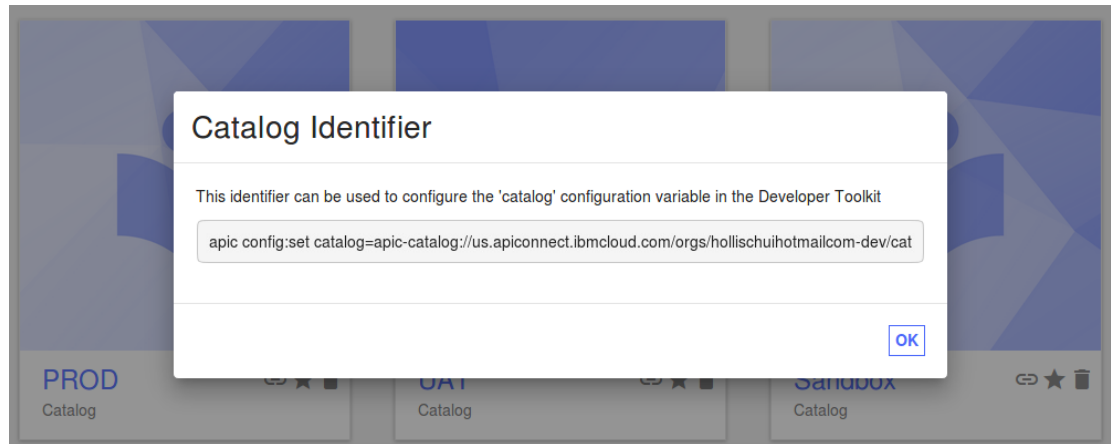


### 4. Make a note of the Catalog Identifier for each Catalog as this information will be used in later in the lab. The Catalog Identifier contains the API Connect Server URL, the API Connect Org and the Catalog information.

The value of the API Connect Org is different than the Bluemix account Org. This value will be required in Lab 6.3

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





## Lab 2: GitHub Repository

### Lab 2.1: Create a GitHub Repository

1. Launch the Firefox 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 GitHub repository

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

- Go 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 GitHub repository

- From the terminal window, go into the clone repository directory `/home/devops/<Folder name>/acme-bank-api-yaml`.
- 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>'**.
- 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

- From the terminal window, go up one level to `/home/devops/<Folder name>`.
- Remove the directory by running the command **'rm -rf acme-bank-api-yaml'**.

### Lab 2.5: Clone the GitHub repository as a local workspace

- From the terminal window, go 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](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.
```

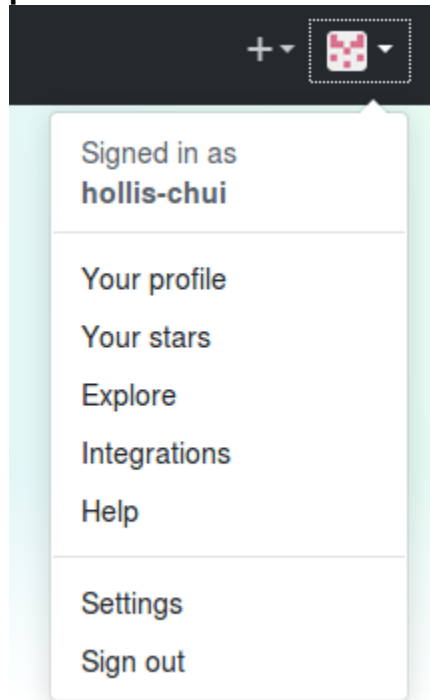
3. Go into the '**acme-bank-api-yaml**' directory and verify the following files exist.

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

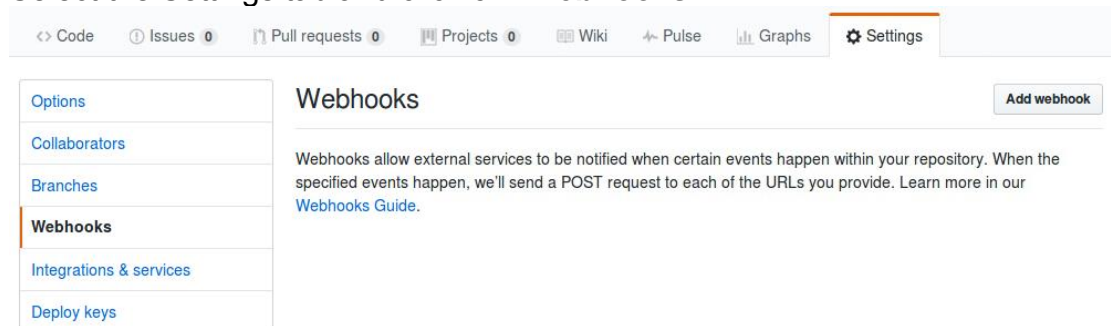
## Lab 3: GitHub and Jenkins integration

### Lab 3.1: Configure the GitHub and Jenkin integration

1. Launch the Firefox 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”.

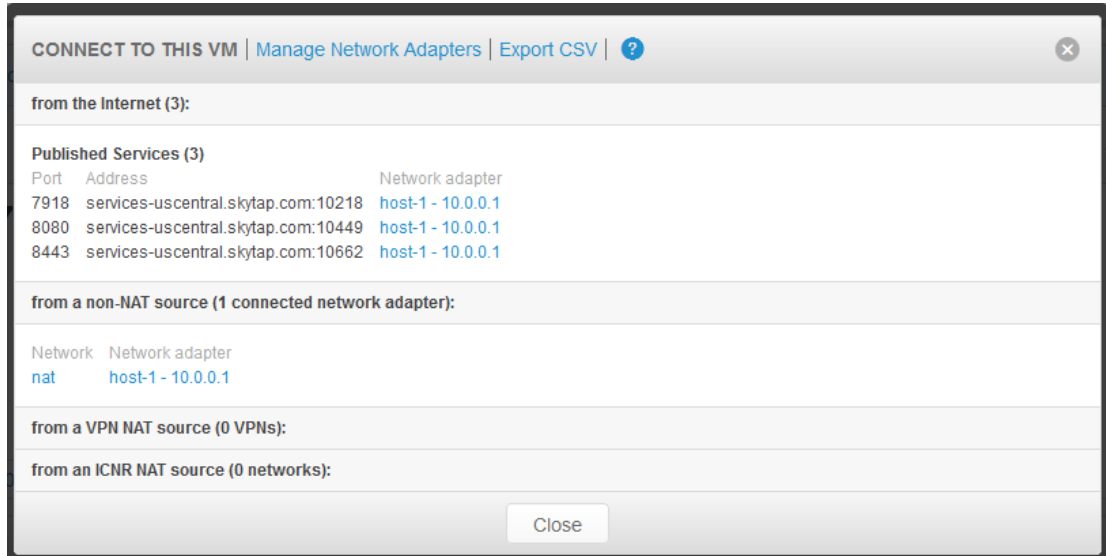


4. Select the Repositories tab and click on the “acme-bank-api-yaml” repository.  
Select the Settings tab and click on “Webhooks”.



5. For the Skytap VM, find the Address corresponding to Port 8080. This is the external URL to access the Jenkins UI. This URL address will be used in the webhook. In the diagram below, <http://services-uscentral.skytap.com:10449> is the URL for the

Jenkins UI.



6. Click the “**Add webhook**” button.
7. In the Payload URL, enter the URL [‘http://services-uscentral.skytap.com:10449/github-webhook/’](http://services-uscentral.skytap.com:10449/github-webhook/) and click the “**Add webhook**” button.

Webhooks / **Add webhook**

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in [our developer documentation](#).

**Payload URL \***

**Content type**

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

## Webhooks

[Add webhook](#)

Webhooks allow external services to be notified when certain events happen within your repository. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our [Webhooks Guide](#).

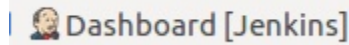
✓ <http://services-uscentral.skytap.com:10449/github-webhook/> (push)

[Edit](#) [Delete](#)

## Lab 4: IBM UrbanCode Deploy and Jenkins integration





### Lab 4.1: Configure the IBM UrbanCode Deploy and Jenkins integration

1. Launch the Firefox browser and click on the Jenkins bookmark.



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="*****"/>	

Delete

Test Connection

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="*****"/>	

Delete

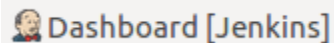
Success

Test Connection

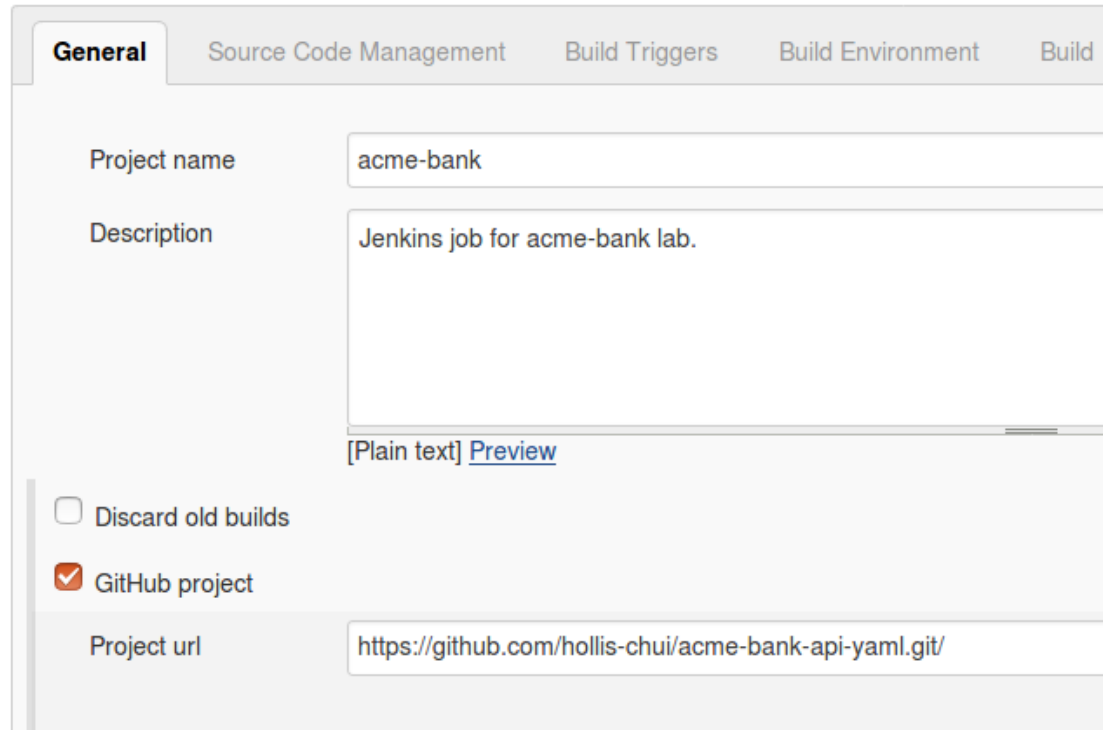
## Lab 5: Jenkins Configuration

### Lab 5.1: Update Jenkins Job

1. Launch a Firefox browser and click on the Jenkins bookmark.



1. Log in to Jenkins using **admin / admin**.
2. Click on the **“acme-bank”** link.
3. Click on the **“Configure”** link on the left menu.
4. On the **“General”** tab, update the **“GitHub project > Project url”** property with your GitHub repository URL in HTTPS format.



The screenshot shows the Jenkins job configuration page for a job named 'acme-bank'. The 'General' tab is selected, showing fields for 'Project name' (acme-bank), 'Description' (Jenkins job for acme-bank lab.), and 'Project url' (https://github.com/hollis-chui/acme-bank-api-yaml.git/). The 'GitHub project' checkbox is checked, and the 'Discard old builds' checkbox is unchecked. A 'Preview' link is visible below the description field.

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

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



General

Source Code Management

Build Triggers

Build Environment

Build

Post-I

### Source Code Management

☐ None

☒ Git

Repositories

Repository URL

git@github.com:hollis-chui/acme-bank-api-yaml.git

Credentials

- none -

Add

## 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**’.

### Lab 6.2: Update component process with Bluemix password

2. Launch a Firefox browser and click on the IBM UrbanCode Deploy bookmark.

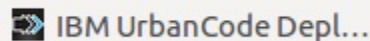


3. Log in to UCD using **admin / admin**.
4. Select the “**Component**” tab and click on the “**acme-bank**” link.
5. Select the “**Processes**” tab and click on the “**Deploy API Product Def**” link.
6. Edit the “**Login**” step on the Component Process diagram.
7. Update the “**Password**” property with your Bluemix account password.

Note: This value can be parameterized as a Environment property.

### Lab 6.3: Update Environment Properties

1. Launch a Firefox browser and click on the IBM UrbanCode Deploy bookmark.



2. Login to UCD using **admin / admin**.
3. Select the “**Configuration**” tab and verify there are 3 environments listed under acme-bank.

▼ acme-bank
▼ acme-bank
PROD
Sandbox
UAT
<a href="#">Refresh</a> <a href="#">Print</a>

4. For the Sandbox environment, update the following properties.
  - i. API Connect Account Username = *<Your Bluemix account username>*
  - ii. API Connect Server Org = *<API Connect Org based on the Catalog Identifier>*

### 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](#)

5. Repeat Step 4 for the UAT and PROD environments. The **API Connect Server Org** property can be found in the Catalog Identifier.

## 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:
```

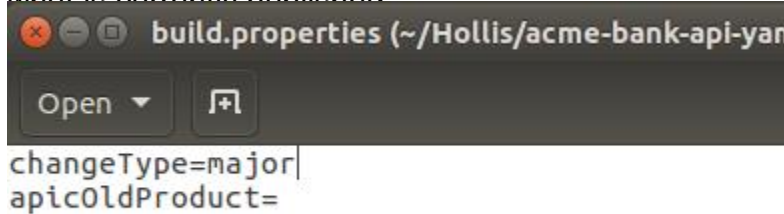
3. Open “**acme-bank.yaml**” in an editor and update **version** property from **x.x.x** to **1.0.0**. Save the file and exit the editor.

```
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

## Deploy and API Connect

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.

**Jenkins**

Jenkins > acme-bank >

[Back to Dashboard](#)

[Status](#)

[Changes](#)

[Workspace](#)

[Build Now](#)

[Delete Project](#)

[Configure](#)

[Schedule Build](#)

[GitHub Hook Log](#)

[Move](#)

[GitHub](#)

## Project acme-bank

Jenkins job for acme-bank lab.

[Workspace](#)

[Recent Changes](#)

### Permalinks

**Build History** [trend](#)

find X

#1 Mar 12, 2017 6:02 PM X

[RSS for all](#) [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.
 

IBM UrbanCode Depl...
  - 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 = 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. First	100%	9:03:59 PM	0:00:00	Success
8. envType	0%			Not Started
9. Replace Product in Catalog	0%			Not Started
10. Publish API Product Definitions	100%	9:03:59 PM	0:00:33	Success
11. Version	0%			Not Started
12. RemovePreviousVersion	0%			Not Started
13. Logout	100%	9:04:33 PM	0:00:10	Success
Total Execution	7 / 9	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 (1/1)	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

Q 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.  
 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.

Dashboard
Components
Applications
Configuration
Processes

Home > Components > acme-bank > Versions > Version: acme-bank:1.0.0

Version: acme-bank:1.0.0 (show details)

Main
Configuration
History

### Statuses

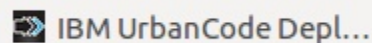
Add a Status

Status	Description
release	

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
acme-bank	None	acme-bank:1.0.0 x Add...

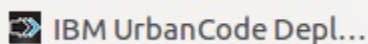
1 record

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	Not Started			Not Started
9. Replace Product in Catalog	Not Started			Not Started
10. Publish API Product Definitions	100%	9:23:34 PM	0:00:29	Success
11. isVersion	Not Started			Not Started
12. RemovePreviousVersion	Not Started			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.

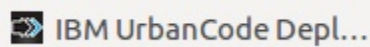
Component	Version	Date Deployed	Compliance	Actions
acme-bank	acme-bank:1.0.0 (View Details)	3/12/2017, 9:22 PM	Compliant (100%)	View Request

- 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 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.

Dashboard	Components	Applications	Configuration	Processes
Home > Components > acme-bank > Versions > Version: acme-bank:1.0.0				
Version: acme-bank:1.0.0 (show details)				
Main	Configuration	History		

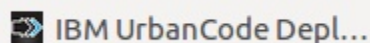
## Statuses

Add a Status

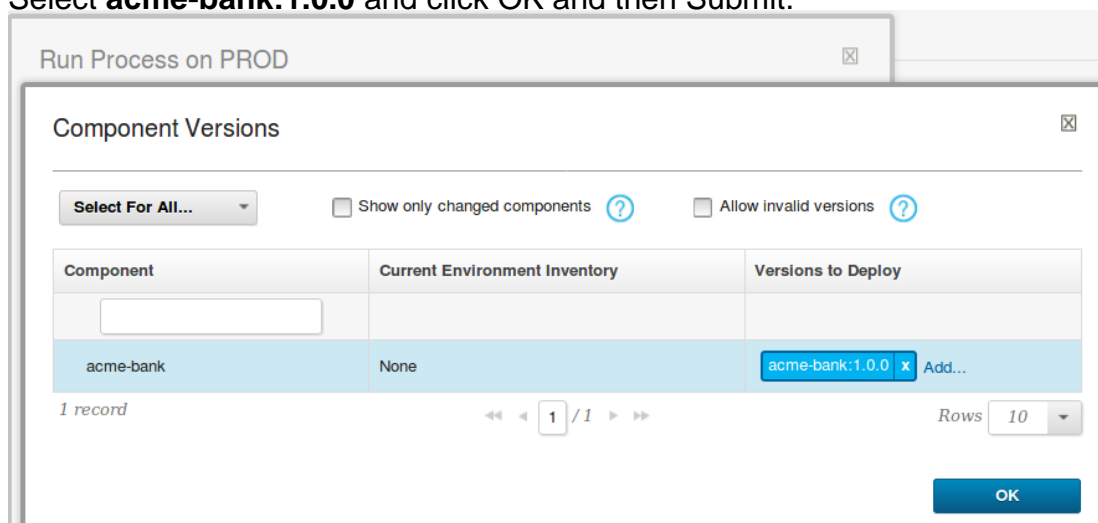
Status	Description
release	
tested	

2 records - Refresh Print

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.



Run Process on PROD

Component Versions

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

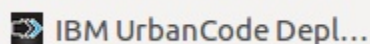
Component	Current Environment Inventory	Versions to Deploy
acme-bank	None	acme-bank:1.0.0 x Add...

1 record

Rows 10

OK

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. IstFirst	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](#)

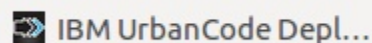
find X

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.

- 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 = 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

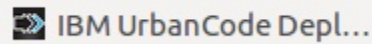
9. Once the API has been published to API Connect Sandbox Catalog, teams can perform their testing. Once this is completed, the API Product owner will add the “**release**” Version Status to the **acme-bank:1.1.0** component version and then manually trigger the



## Deploy and API Connect

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.

**IBM UrbanCode Deploy**

Dashboard | Components | Applications | Configuration | Processes

Home > Components > acme-bank > Versions > Version: acme-bank:1.1.0

**Version: acme-bank:1.1.0** (show details)

Main | Configuration | History

**Statuses**

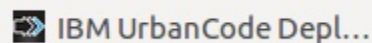
Add a Status

Status	Description
release	

1 record - Refresh Print

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.

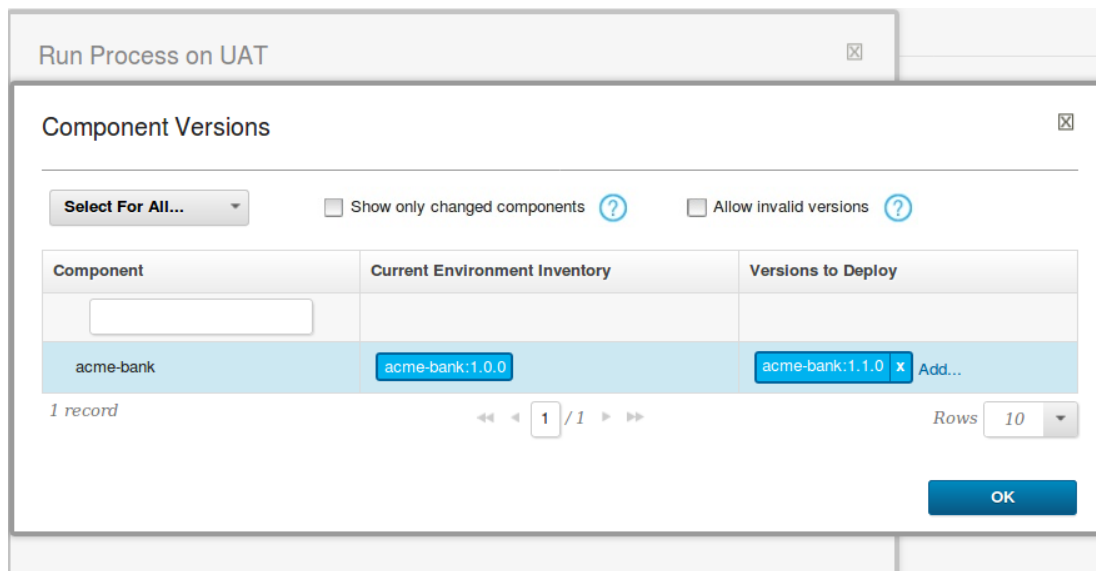
- a. Open a Firefox browser and click the IBM UrbanCode Deploy bookmark.



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

## Deploy and API Connect

- 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:1.1.0** and click OK and then Submit.



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

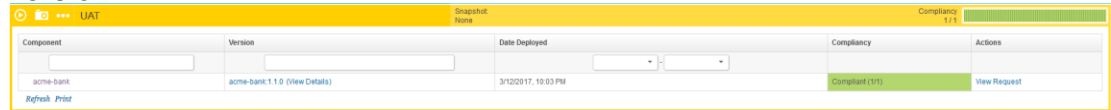
- a. Open the Firefox browser and click on the IBM UrbanCode Deploy bookmark.



- 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

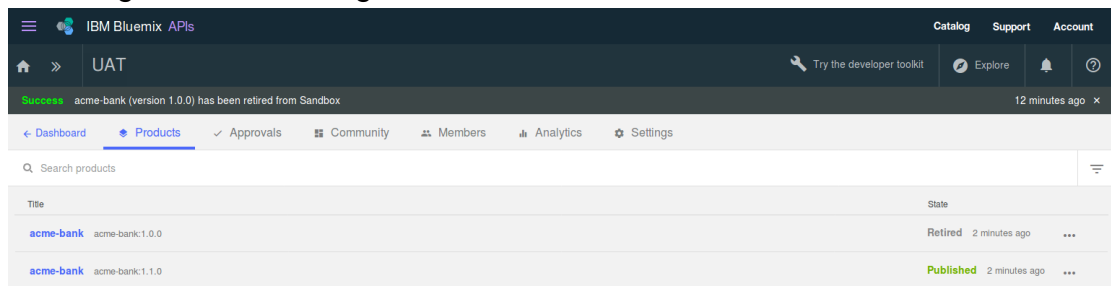
Step	Progress	Start Time	Duration	Status
1. Download API YAML Files	100%	10:03:52 PM	0:00:02	Success
2. Set Permissions	100%	10:03:54 PM	0:00:01	Success
3. Login	100%	10:03:56 PM	0:00:15	Success
4. Set Configuration Variable	100%	10:04:11 PM	0:00:05	Success
5. Publish API Product Definitions in Stage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



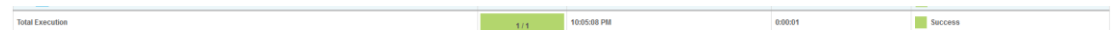
Component	Version	Date Deployed	Compliance	Actions
acme-bank	acme-bank:1.1.0 (View Details)	3/12/2017, 10:03 PM	Compliant (1/1)	View Request

- 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.



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



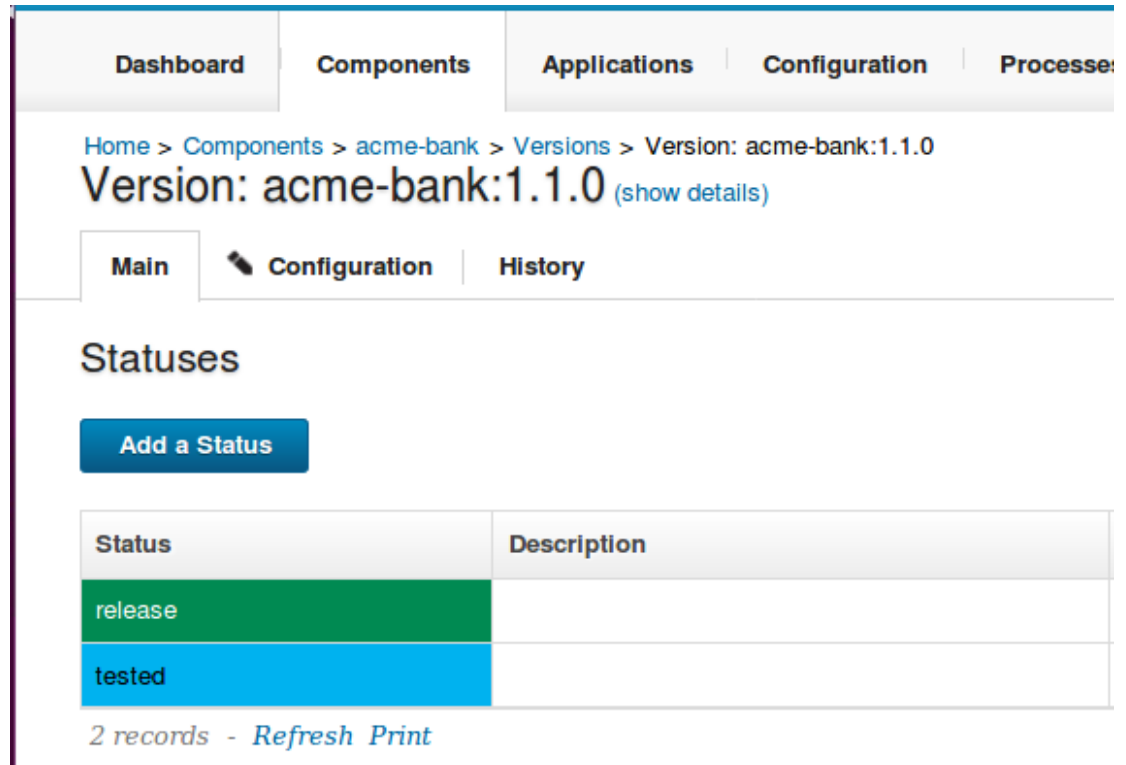
Total Execution
Success

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.

- 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 **“tested”** Status in the dropdown menu and click Save.



Dashboard | Components | Applications | Configuration | Processes

Home > Components > acme-bank > Versions > Version: acme-bank:1.1.0

Version: acme-bank:1.1.0 (show details)

Main | Configuration | History

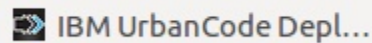
### Statuses

Add a Status

Status	Description
release	
tested	

2 records - Refresh Print

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



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

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

### 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...

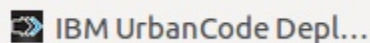
1 record

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

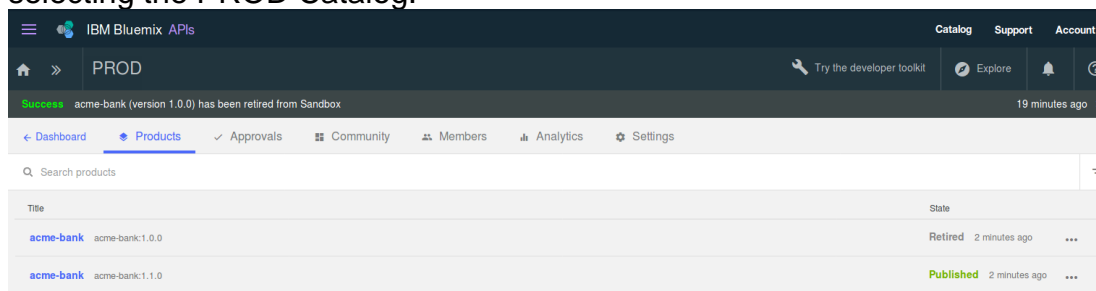
Step	Progress	Start Time	Duration	Status
1. Download API YAML Files		10:10:57 PM	0:00:02	Success
2. Set Permissions		10:10:59 PM	0:00:01	Success
3. Login		10:11:01 PM	0:00:13	Success
4. Set Configuration Variable		10:11:14 PM	0:00:05	Success
5. Publish API Product Definitions in Stage		10:11:20 PM	0:00:20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

Refresh Print

- 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.



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

process	environment	version	requester	environment	submitted on	by	status	request
RemoveFromInventory	/acme-bank/PROD/devops_local_agent/acme-bank	acme-bank:1.0.0	acme-bank	PROD	3/12/2017, 10:12 PM	admin	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. .

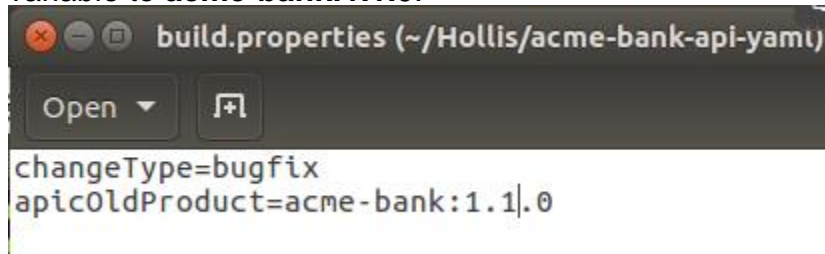
1. Open a terminal window and go to the local GitHub workspace “**/home/devops/<Folder name>/**”.
2. Open “**acme-bank-product.yaml**” in an editor and update **version** property from **1.1.0** to **1.1.1**. Save the file and exit the editor.

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

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.

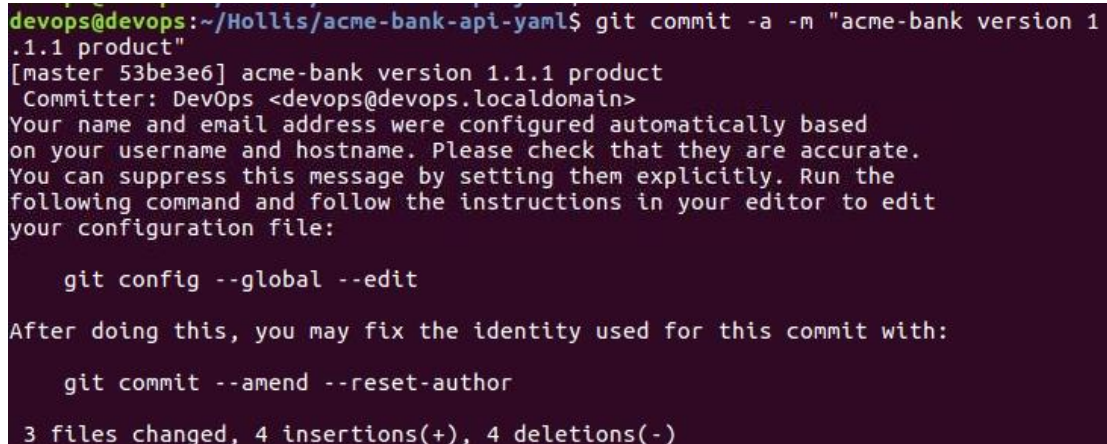
```
basePath: /api
swagger: '2.0'
info:
  x-ibm-name: acme-bank
  version: 1.1.1
  title: acme-bank
schemes:
  - https
host: $(catalog.host)
consumes:
```

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



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

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 1.1.1 product"
[master 53be3e6] acme-bank version 1.1.1 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(-)
```



## Deploy and API Connect

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

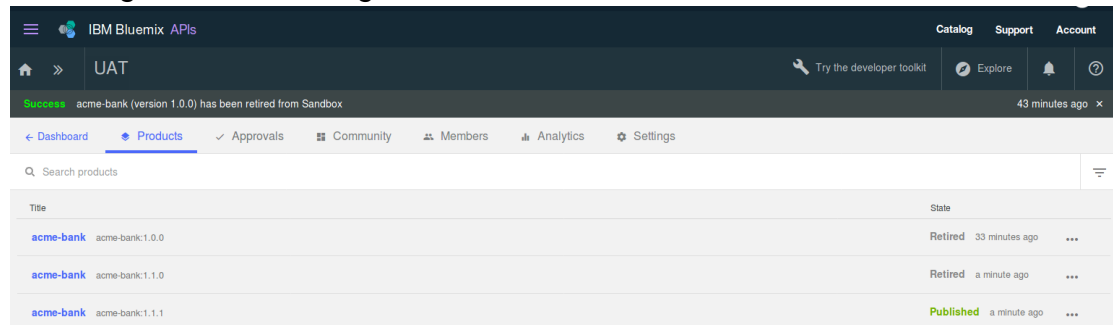
Step	Progress	Start Time	Duration	Status
1. Download API YAML Files	100%	10:35:39 PM	0:00:02	Success
2. Set Permissions	100%	10:35:42 PM	0:00:01	Success
3. Login	100%	10:35:43 PM	0:00:14	Success
4. Set Configuration Variable	100%	10:35:57 PM	0:00:05	Success
5. Publish API Product Definitions in Stage	100%	10:36:03 PM	0:00:19	Success
6. API: Publish or Replace	100%	10:36:23 PM	0:00:00	Success
7. nfFirst	Not Started			Not Started
8. envType	Not Started			Not Started
9. Replace Product in Catalog	100%	10:36:24 PM	0:00:36	Success
10. Publish API Product Definitions	Not Started			Not Started
11. nVersion	100%	10:37:00 PM	0:00:00	Success
12. RemovePreviousVersion	100%	10:37:00 PM	0:00:05	Success
13. Logout	100%	10:37:06 PM	0:00:10	Success
Total Execution	8 / 9	10:35:39 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.



- 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	0	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.



- 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:1.1.1** component version.
- On the Main tab, click the “**Add a Status**” button, select “**tested**” Status in the dropdown menu and click Save.

Dashboard
Components
Applications
Configuration
Processes

Home > Components > acme-bank > Versions > Version: acme-bank:1.1.1

Version: acme-bank:1.1.1 (show details)

Main
Configuration
History


### Statuses

Add a Status

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.

 IBM UrbanCode Depl...
  - 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

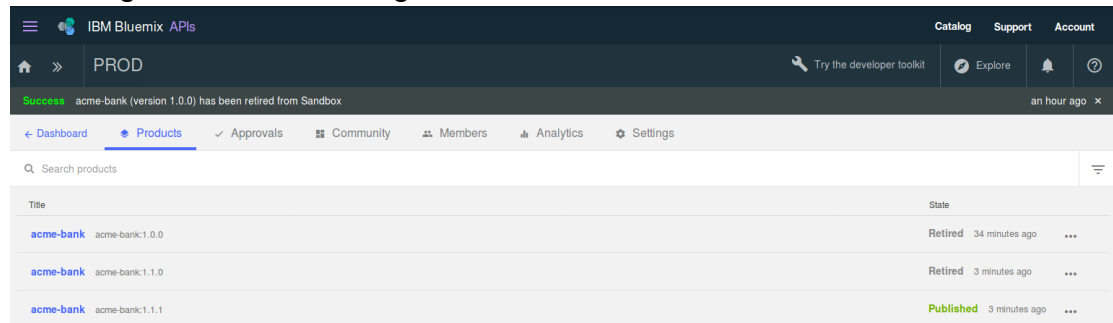
Step	Progress	Start Time	Duration	Status
1. Download API YAML Files	<div></div>	10:41:50 PM	0:00:02	Success
2. Set Permissions	<div></div>	10:41:52 PM	0:00:01	Success
3. Login	<div></div>	10:41:53 PM	0:00:15	Success
4. Set Configuration Variable	<div></div>	10:42:09 PM	0:00:06	Success
5. Publish API 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. InFirst	<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. WVersion	<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.

Component	Version	Date Deployed	Compliance	Actions
acme-bank	acme-bank:1.1.1 (New Details)	3/12/2017, 10:41 PM	Compliant (100%)	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.



- 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	1	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.