InterConnect
2017

March 19–23
MGM Grand &
Mandalay Bay
Las Vegas, NV

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

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.

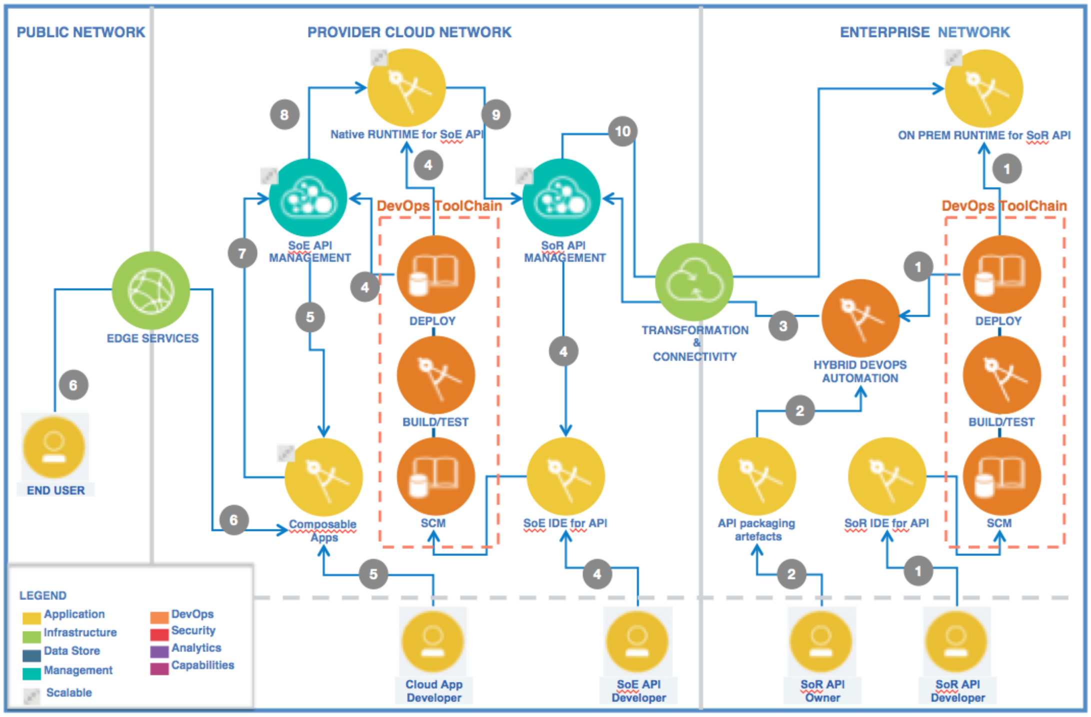
Table of Contents

Introduction

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 (ie. 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 (ie. 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.

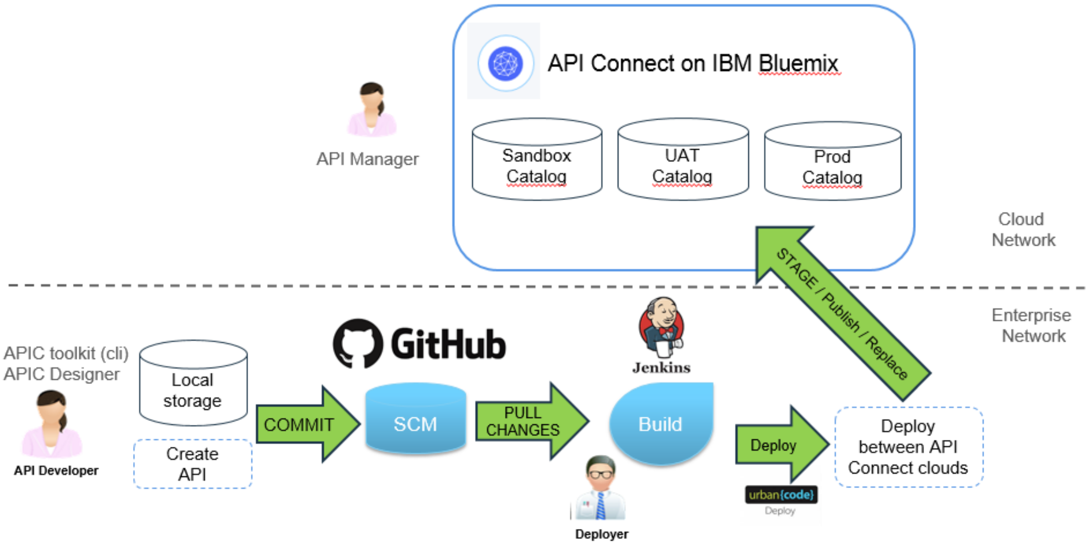
Lab Overview

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.

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	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 – devops / devops
- Jenkins – admin / admin
- IBM UrbanCode Deploy – admin /admin

Installed Software:

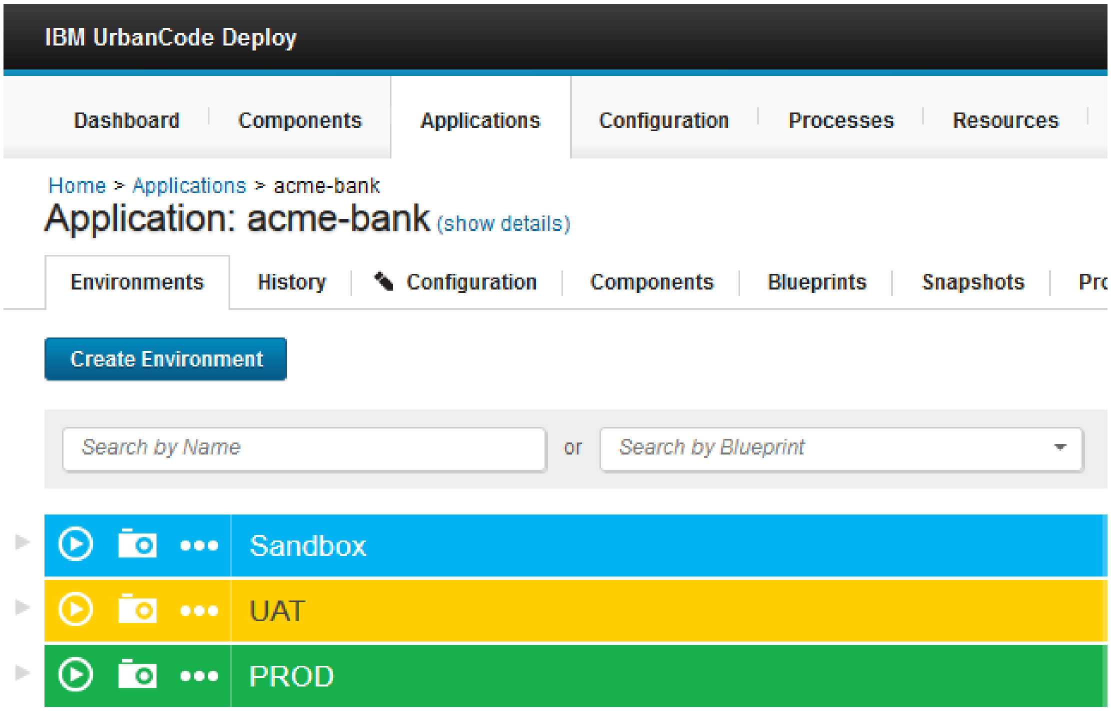
- IBM UrbanCode Deploy 6.2.2 + local agent
  - API Connect plugin for UCD
- Jenkins 2.19
  - IBM UrbanCode Deploy Build Steps plugin
  - Git Tag Message plugin
  - GitHub plugin
- Node.js 4.x
- API Connect Toolkit

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:

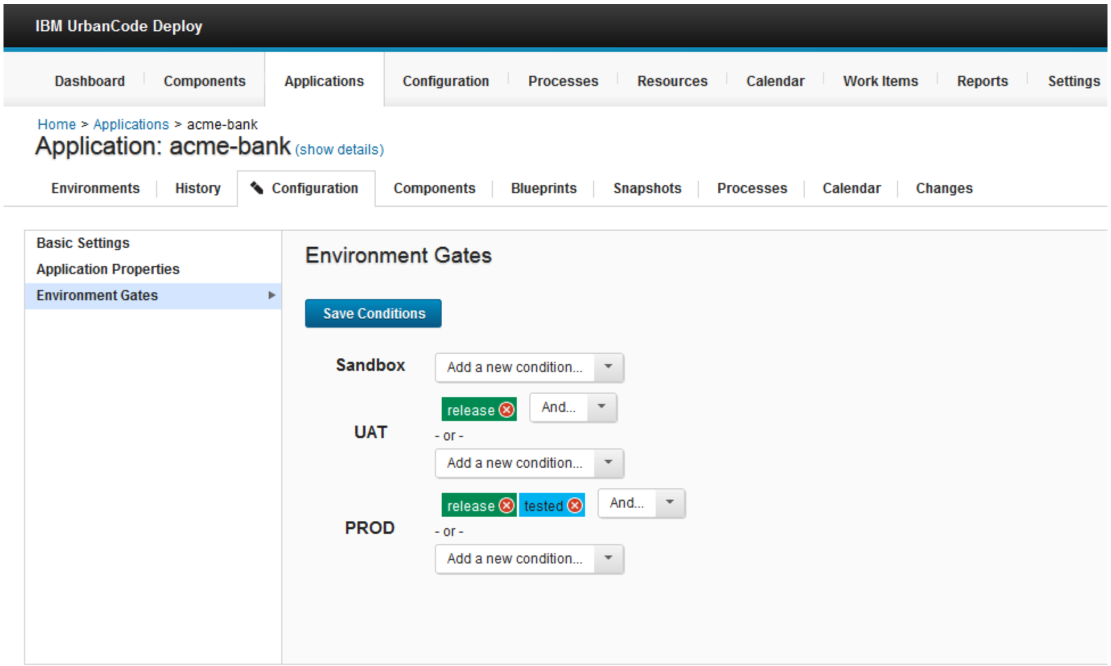
- Download the GitHub repository files into the Jenkins workspace.
- Extract the “major”, “minor” or “bugfix” keyword in the Git tag and run through logic to implement the API governance model.
- Create a component version in IBM UrbanCode Deploy and set component version properties.
- Add the “release” 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.



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.

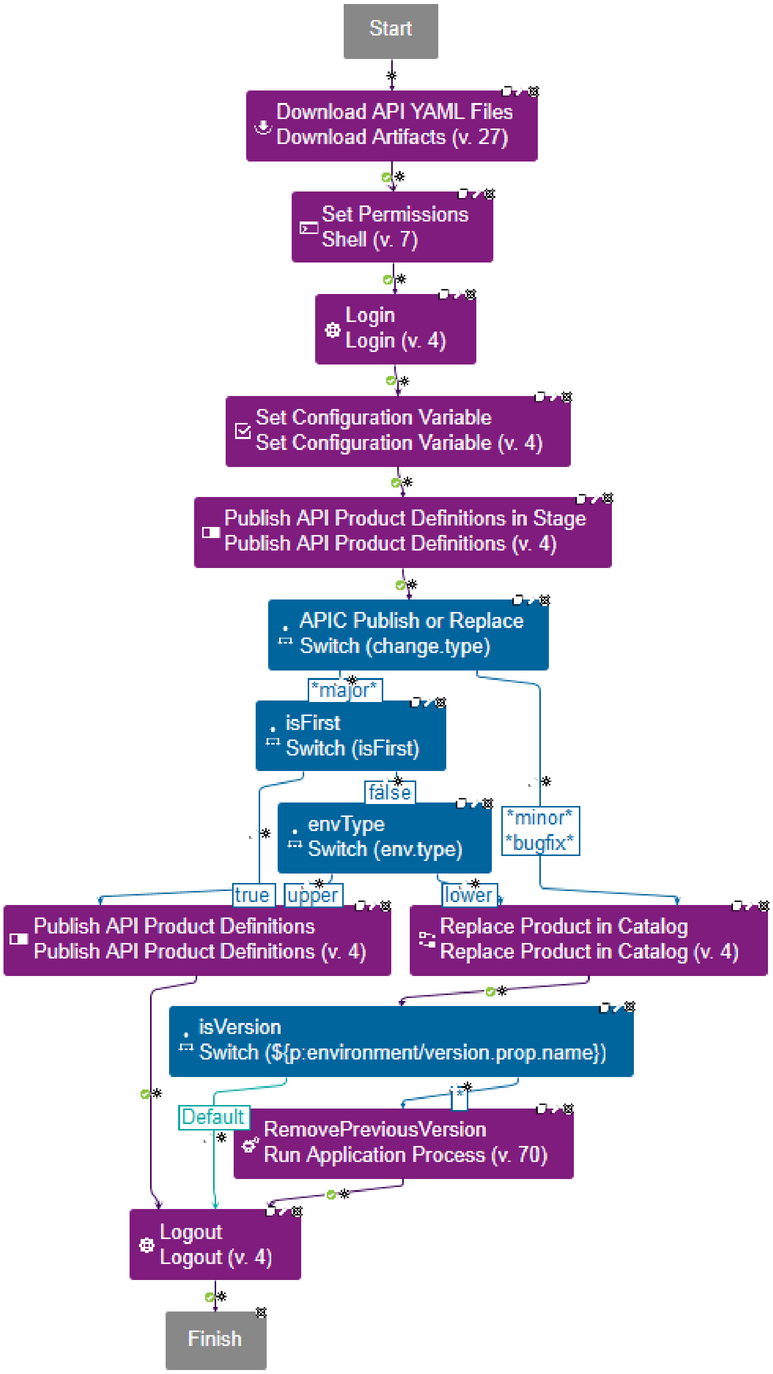


Sandbox: Add a new condition...

UAT: release - or - Add a new condition...

PROD: release tested - or - Add a new condition...

IBM UrbanCode Deploy Component Process:
The component process in combination with the Jenkins job configuration implements the API governance model discussed in the Overview section.


Start
Download API YAML Files
Download Artifacts (v. 27)
Set Permissions
Shell (v. 7)
Login
Login (v. 4)
Set Configuration Variable
Set Configuration Variable (v. 4)
Publish API Product Definitions in Stage
Publish API Product Definitions (v. 4)
APIC Publish or Replace
Switch (change.type)
major
isFirst
Switch (isFirst)
false
envType
Switch (env.type)
minor
bugfix
true
upper
lower
Publish API Product Definitions
Publish API Product Definitions (v. 4)
Replace Product in Catalog
Replace Product in Catalog (v. 4)
isVersion
Switch (${p:environment/version.prop.name})
Default
*
RemovePreviousVersion
Run Application Process (v. 70)
Logout
Logout (v. 4)
Finish

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.

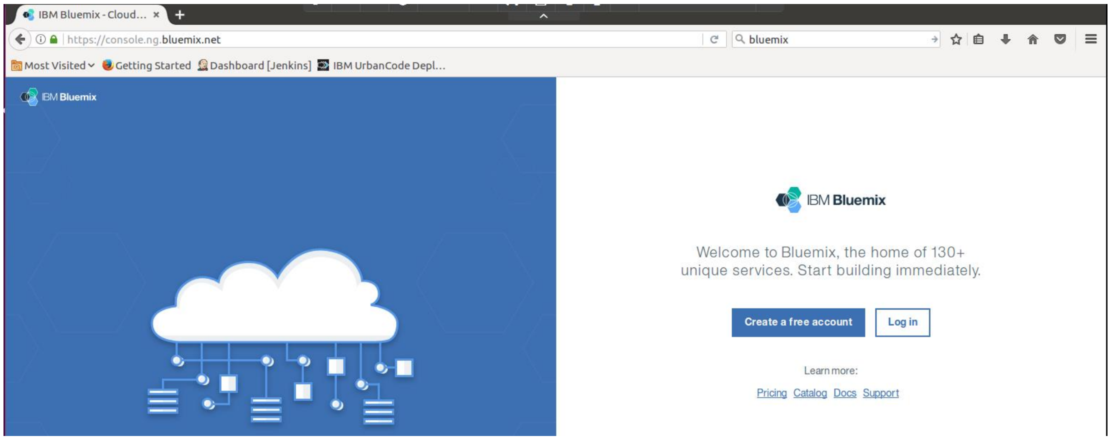
Task 2: Create GitHub account

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

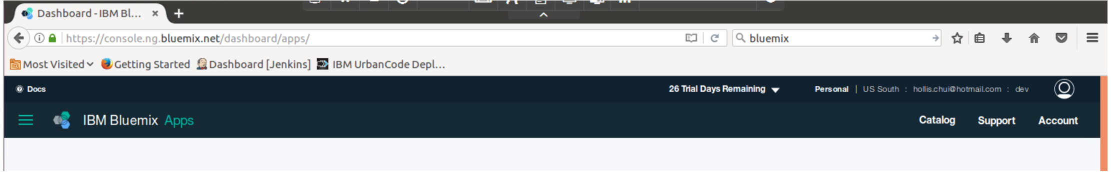
Lab 1.1 - Create an instance of the API Connect Service

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

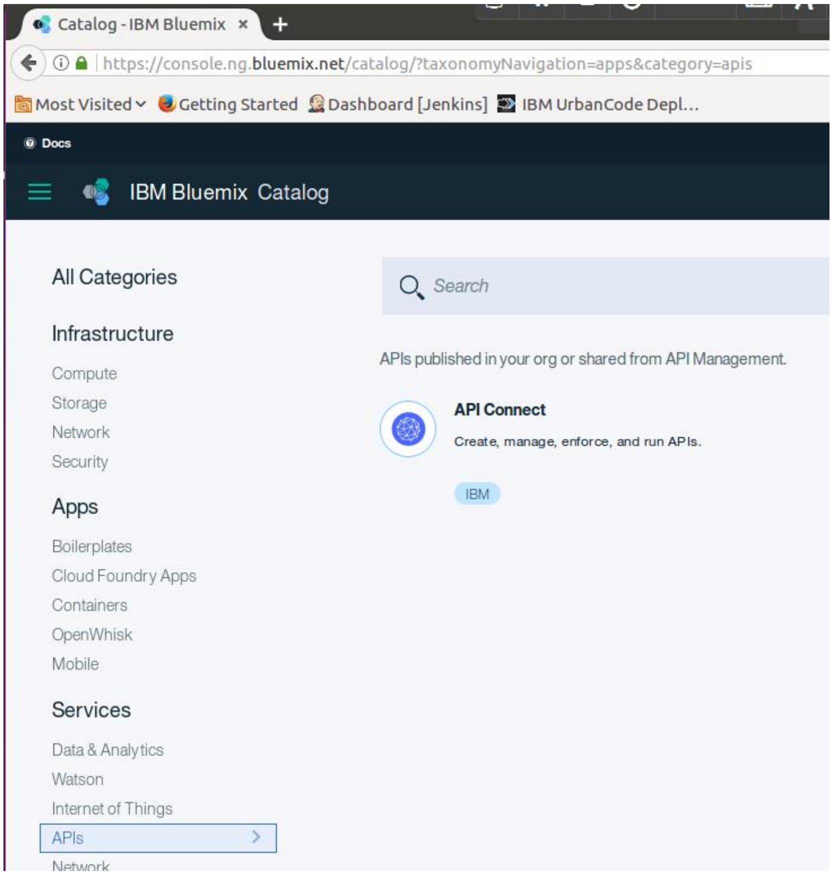


Welcome to Bluemix, the home of 130+ unique services. Start building immediately.

2. Click on Catalog.



3. Select Services > APIs and click API Connect.



Catalog - IBM Bluemix

https://console.ng.bluemix.net/catalog/?taxonomyNavigation=apps&category=apis

Most Visited | Getting Started | Dashboard [Jenkins] | IBM UrbanCode Depl...

Docs

IBM Bluemix Catalog

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

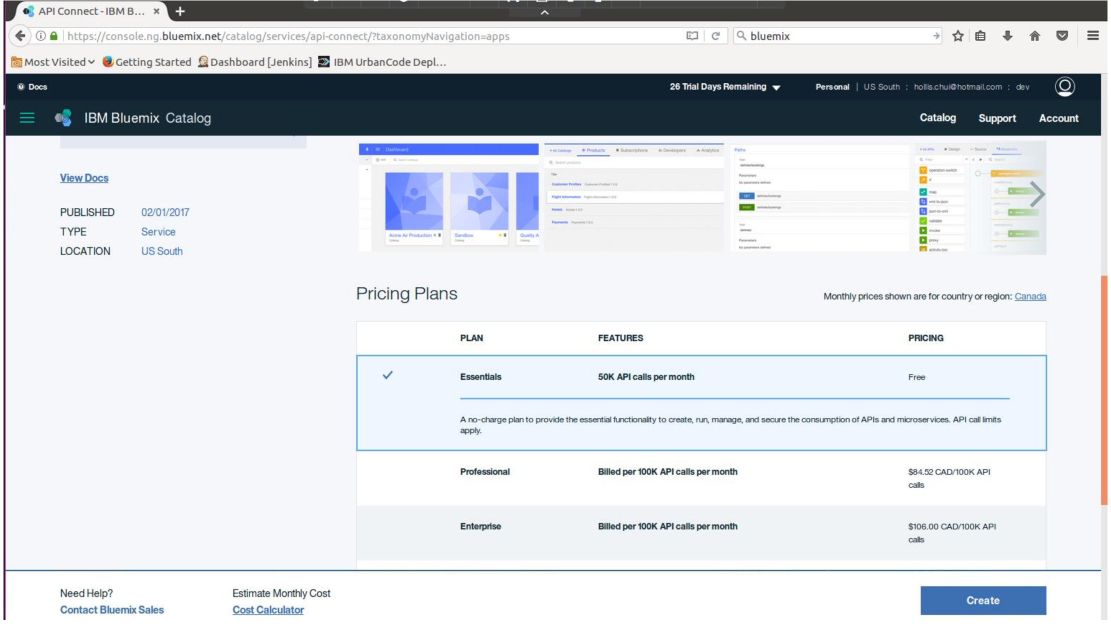
APIs published in your org or shared from API Management.

API Connect

Create, manage, enforce, and run APIs.

IBM

4. Accept the default settings and click Create.



API Connect - IBM B...

https://console.ng.bluemix.net/catalog/services/api-connect/?taxonomyNavigation=apps

bluemix

Most Visited | Getting Started | Dashboard [Jenkins] | IBM UrbanCode Depl...

Docs | 26 Trial Days Remaining | Personal | US South : hollis.chui@hotmail.com : dev

IBM Bluemix Catalog | Catalog | Support | Account

View Docs

PUBLISHED 02/01/2017
TYPE Service
LOCATION US South

Pricing Plans

Monthly prices shown are for country or region: Canada

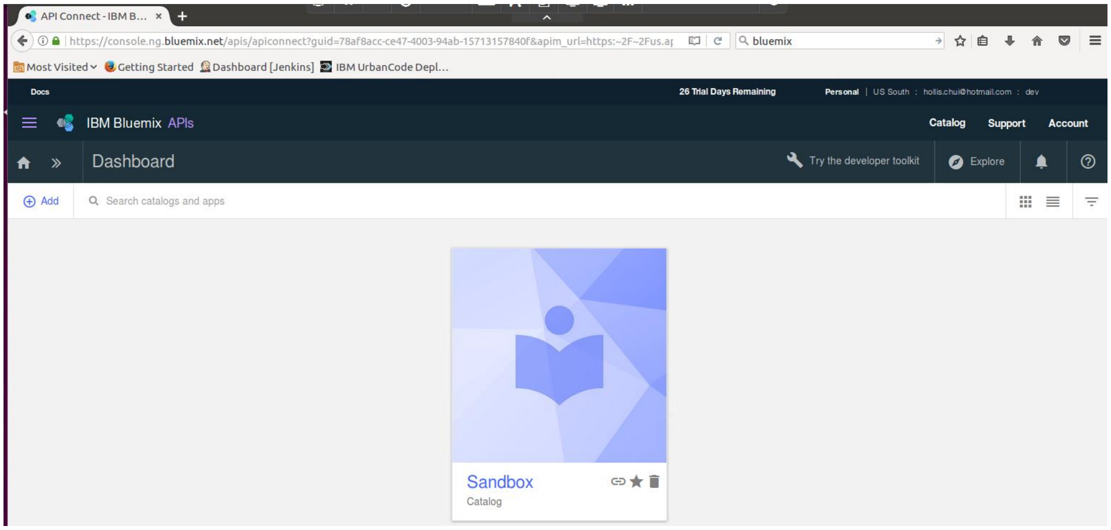
	PLAN	FEATURES	PRICING
✓	Essentials	50K API calls per month	Free
	A no-charge plan to provide the essential functionality to create, run, manage, and secure the consumption of APIs and microservices. API call limits apply.		
	Professional	Billed per 100K API calls per month	$84.52 CAD/100K API calls
	Enterprise	Billed per 100K API calls per month	$106.00 CAD/100K API calls

Need Help?
Contact Bluemix Sales

Estimate Monthly Cost
Cost Calculator

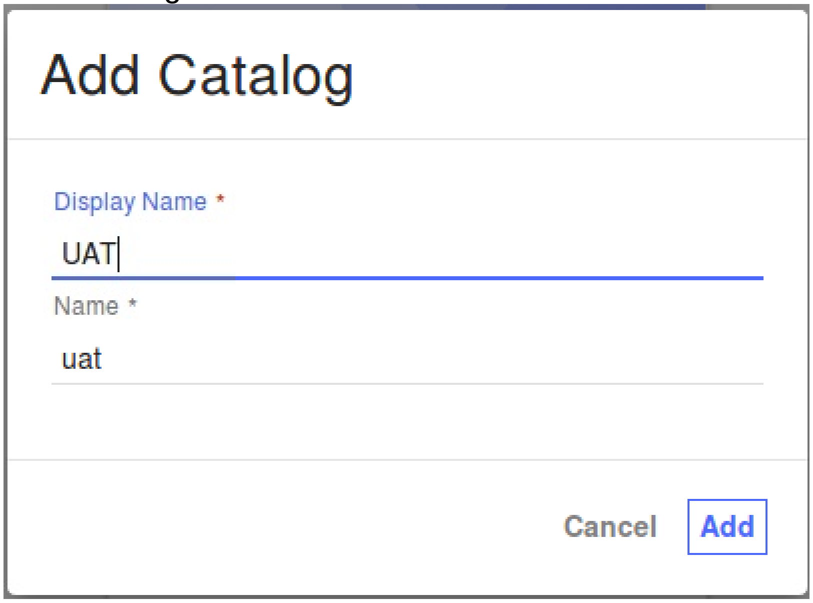
Create

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



Lab 1.2 – Create UAT and PROD Catalogs

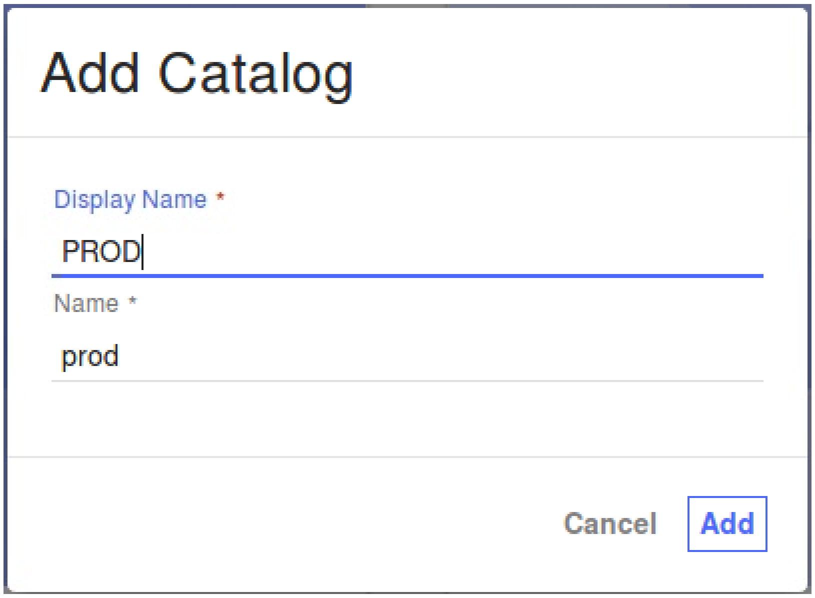
1. From the API Connect Dashboard, click Add > Catalog and create a new Catalog with the name “UAT”.


Add Catalog

Display Name *: UAT

Name *: uat

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

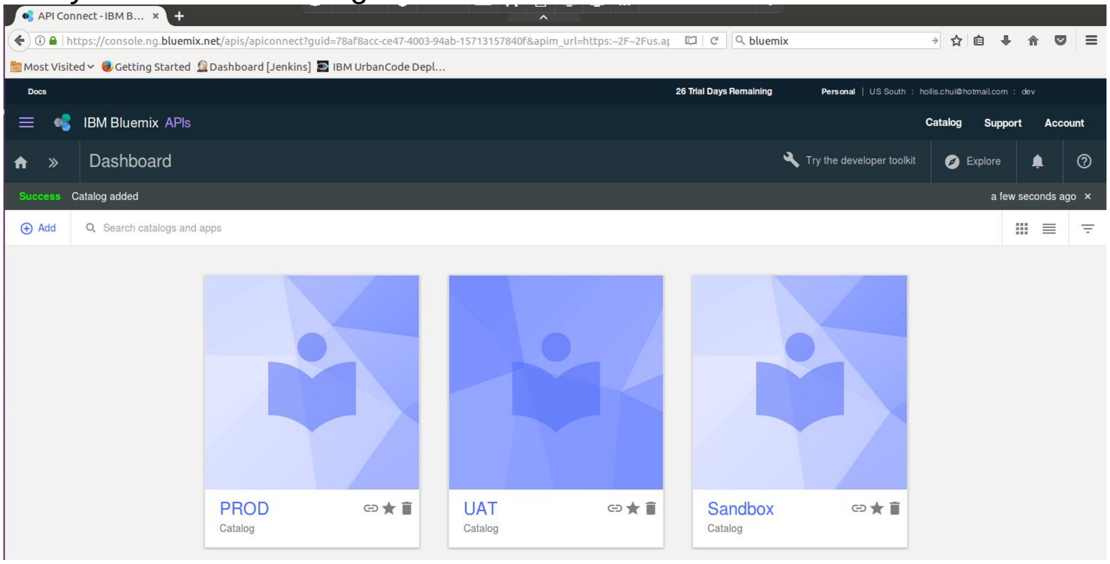


Add Catalog

Display Name * PROD

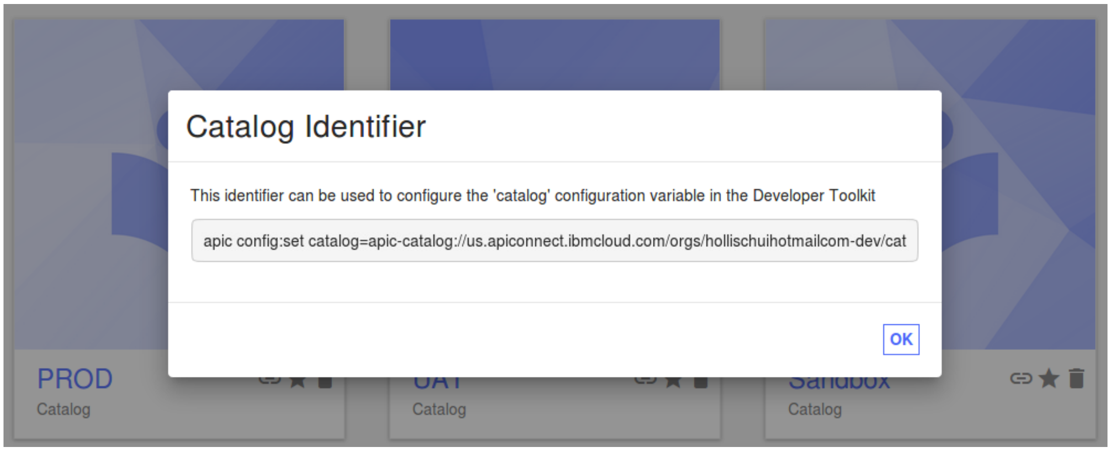
Name * prod

3. Verify there are 3 Catalogs listed.



Success Catalog added

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



Catalog Identifier

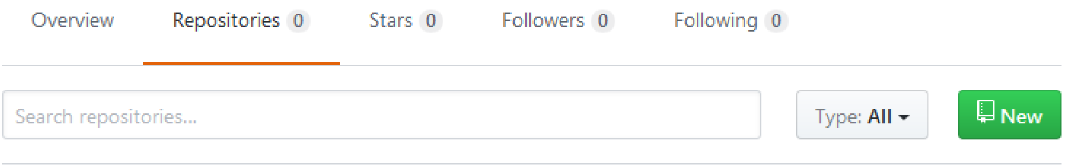
This identifier can be used to configure the 'catalog' configuration variable in the Developer Toolkit

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

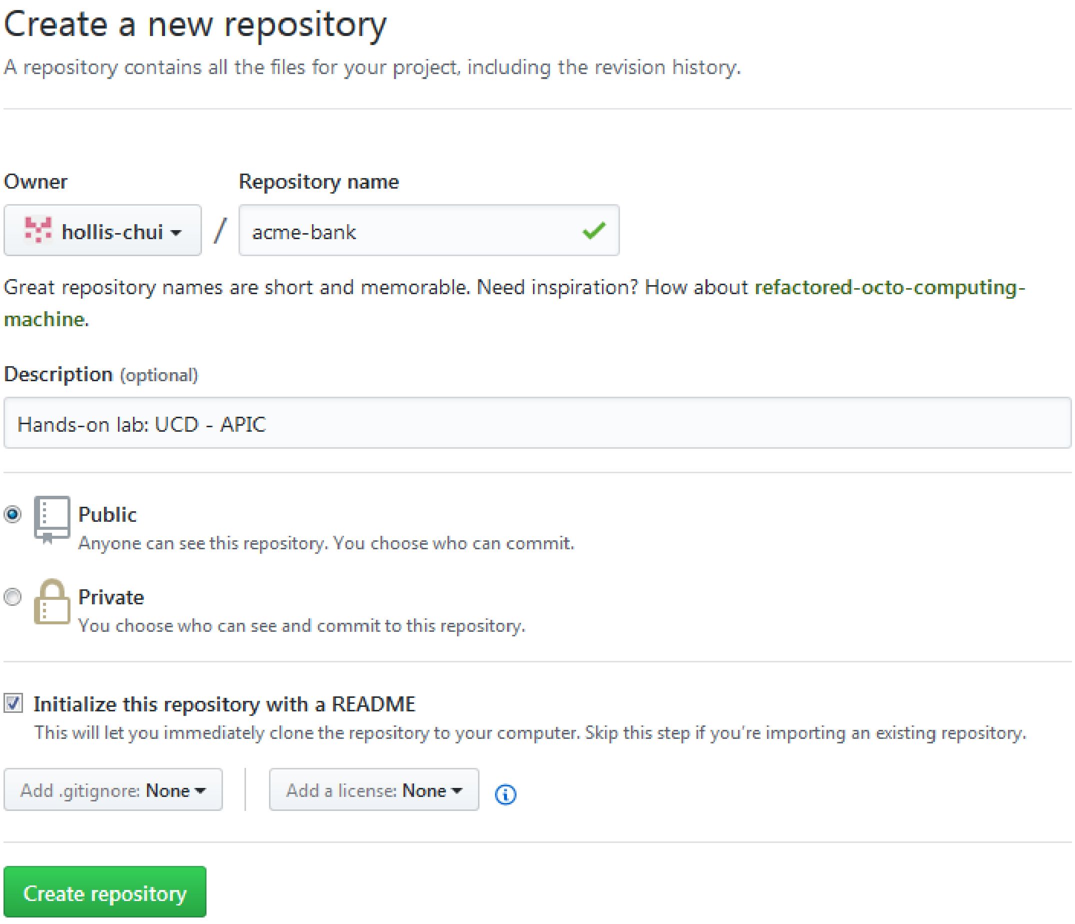
Lab 2: GitHub Repository

Lab 2.1: Create a GitHub Repository

1. Log in to https://github.com/ using the account created in Lab 0.
2. Select the Repositories tab and click the "**New**" button to create a new repository.



3. Enter the name **acme-bank**, select the *Initialize this repository with a README* checkbox and click "**Create Repository**".



Create a new repository

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

Owner: hollis-chui / Repository name: acme-bank ✓

Great repository names are short and memorable. Need inspiration? How about refactored-octo-computing-machine.

Description (optional): Hands-on lab: UCD - APIC

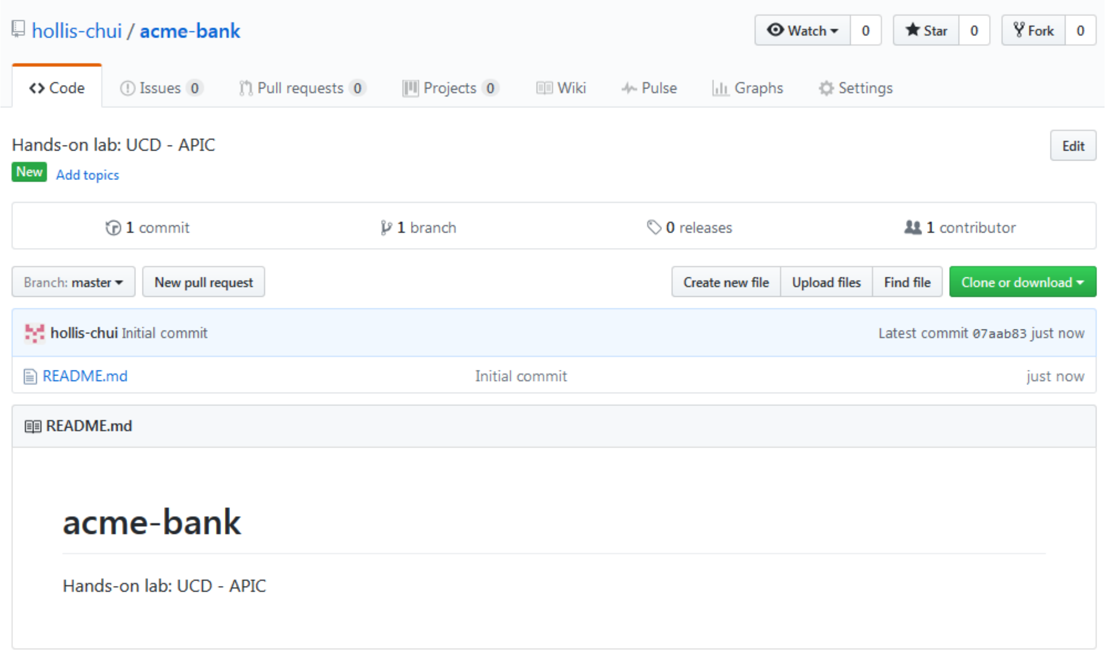
● Public
Anyone can see this repository. You choose who can commit.

○ 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

4. A new repository has been created with a README.



hollis-chui / acme-bank

Hands-on lab: UCD - APIC

1 commit · 1 branch · 0 releases · 1 contributor

hollis-chui Initial commit — Latest commit 07aab83 just now

README.md — Initial commit — just now

README.md

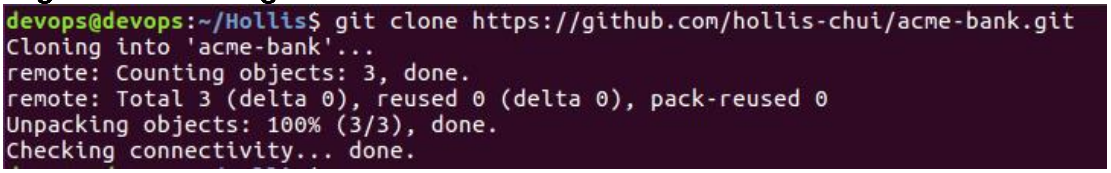
acme-bank

Hands-on lab: UCD - APIC

Lab 2.2: Set up local Git workspace

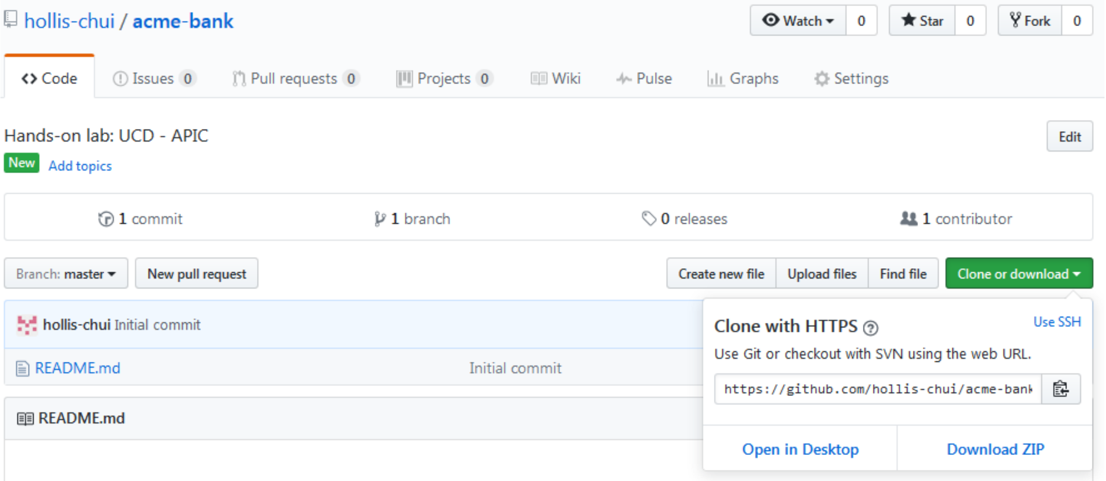
1. Set up your local workspace on the VM for the newly created repository. .

   i) Open a terminal window and go to /home/devops directory.
   ii) Create a folder called "**Lab7028"** to contain the Git workspace.
   iii) Clone the repository to create a local workspace by running the command "**git clone https://github.com/<your GitHub org>/acme-bank.git**".



```
devops@devops:~/Hollis$ git clone https://github.com/hollis-chui/acme-bank.git
Cloning into 'acme-bank'...
remote: Counting objects: 3, done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
Checking connectivity... done.
```

The https or SSH URL of the Git repository is available under the “**Clone or download**” button.



hollis-chui / acme-bank

Hands-on lab: UCD - APIC

1 commit · 1 branch · 0 releases · 1 contributor

hollis-chui Initial commit

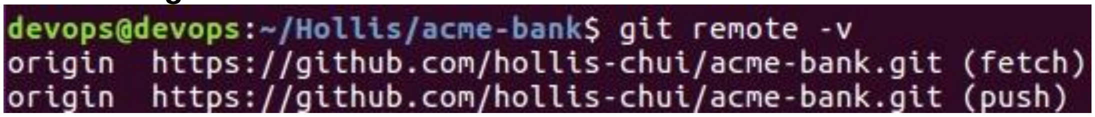
README.md — Initial commit

Clone with HTTPS

Use Git or checkout with SVN using the web URL.

https://github.com/hollis-chui/acme-bank

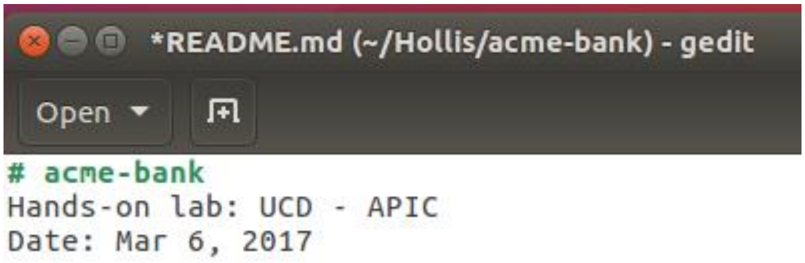
2. Go into the acme-bank directory.
   i) Check the remotes for the clone repository by using the command “**git remote -v**”.



```
devops@devops:~/Hollis/acme-bank$ git remote -v
origin  https://github.com/hollis-chui/acme-bank.git (fetch)
origin  https://github.com/hollis-chui/acme-bank.git (push)
```

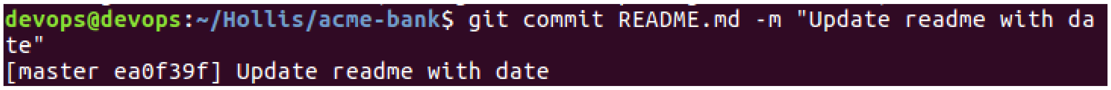
Lab 2.3: Validate the GitHub Setup

1. Validate the set up by updating the README.md and pushing a change into the GitHub repository.
   i) Open the README.md in an editor, add the date to the readme and save the file.



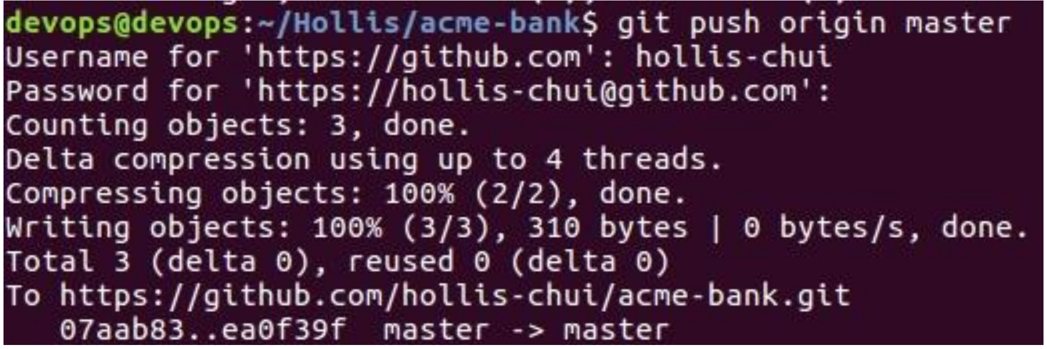
```
# acme-bank
Hands-on lab: UCD - APIC
Date: Mar 6, 2017
```

   ii) Commit the updated README.md into stage using the command “**git commit README.md -m “Update readme with date**”.



```
devops@devops:~/Hollis/acme-bank$ git commit README.md -m "Update readme with da
te"
[master ea0f39f] Update readme with date
```

iii) Push the change into the Git repository using the command “**git push origin master**”**.**



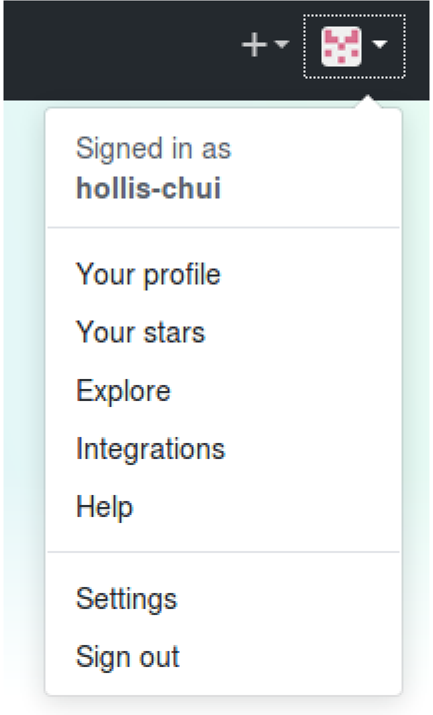
```
devops@devops:~/Hollis/acme-bank$ git push origin master
Username for 'https://github.com': hollis-chui
Password for 'https://hollis-chui@github.com':
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 310 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/hollis-chui/acme-bank.git
   07aab83..ea0f39f  master -> master
```

iv) Refresh the Github repository in your browser and validate the date appears in the readme.

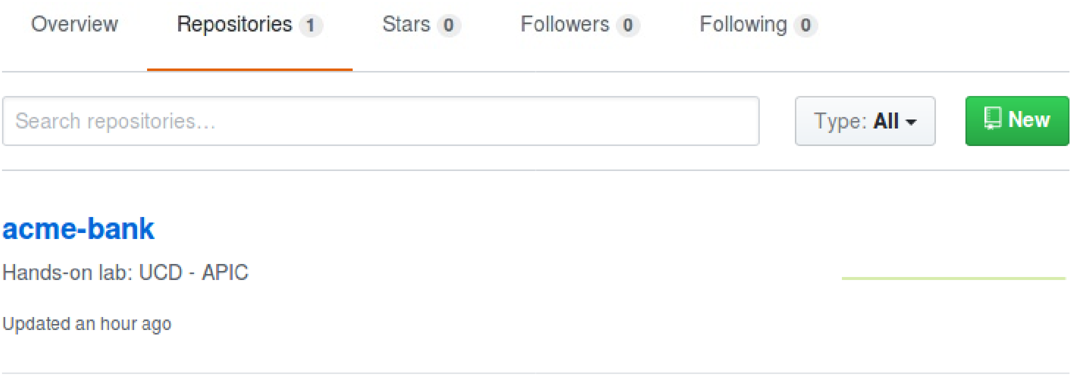
Lab 3: GitHub and Jenkins integration

Lab 3.1 – Configure the GitHub and Jenkin integration

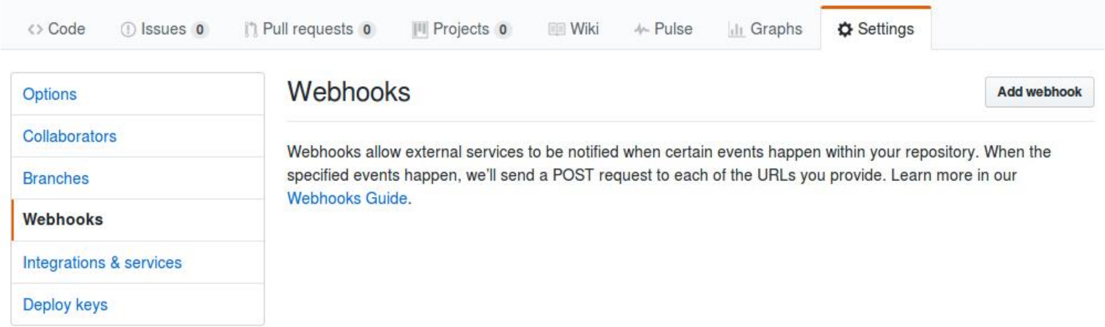
1. Open the Firefox browser, go to https://github.com and log in.
2. Click the dropdown menu on the top right corner and select Your profile.



3. Select Repositories tab and click on “**acme-bank**”.



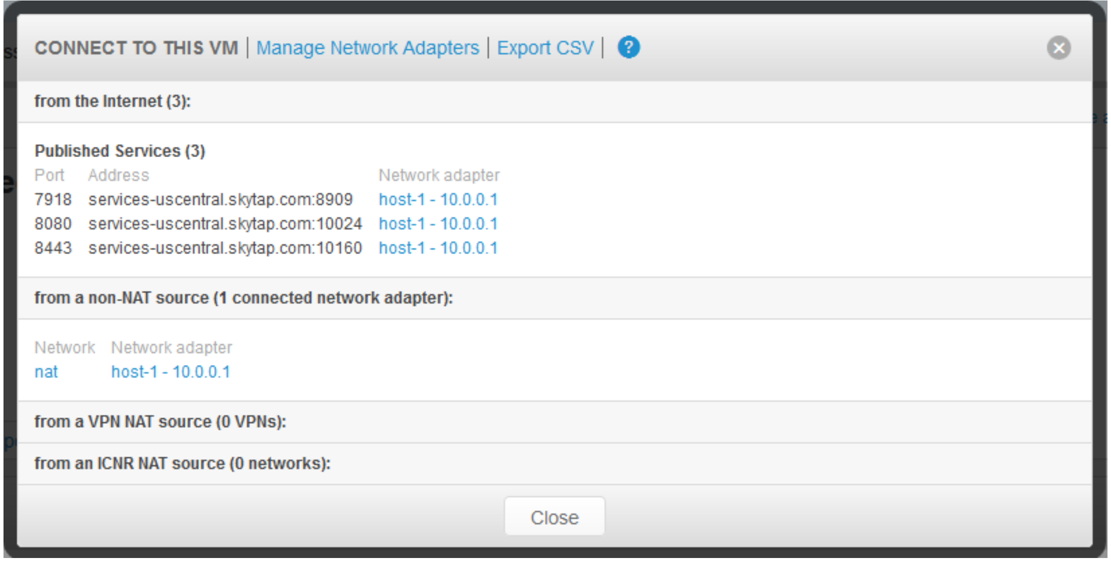
4. Select the Settings tab and click on “**Webhooks**”.



Webhooks

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.

5. Click the “**Add webhook**” button and enter the Jenkins URL endpoint URL from your Skytap VM image. The Jenkins server uses the port 8080 so the Jenkins URL endpoint for the example endpoints is *services-uscentral.skytap.com:10024*.



from the Internet (3):

Published Services (3)

Port	Address	Network adapter
7918	services-uscentral.skytap.com:8909	host-1 - 10.0.0.1
8080	services-uscentral.skytap.com:10024	host-1 - 10.0.0.1
8443	services-uscentral.skytap.com:10160	host-1 - 10.0.0.1

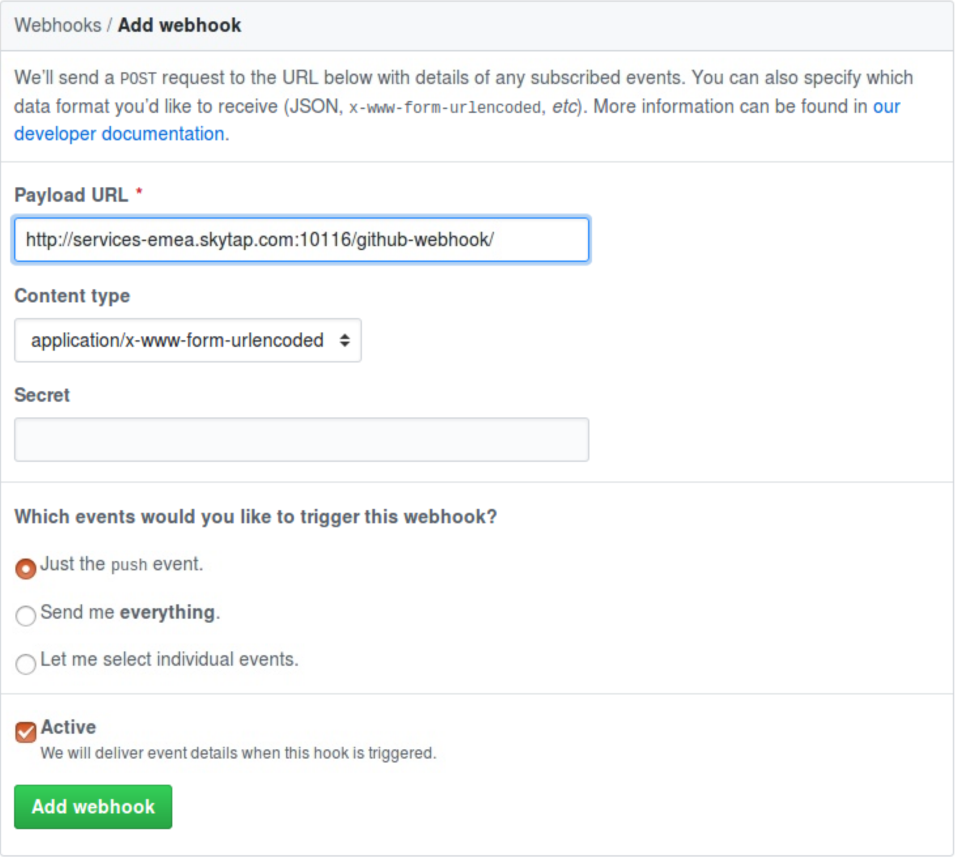
from a non-NAT source (1 connected network adapter):

Network	Network adapter
nat	host-1 - 10.0.0.1

from a VPN NAT source (0 VPNs):

from an ICNR NAT source (0 networks):

6. Input the payload URL using the URL endpoint the format http://<Jenkins URL Endpoint>/github-webhook/.



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 *

http://services-emea.skytap.com:10116/github-webhook/

Content type

application/x-www-form-urlencoded

Secret

Which events would you like to trigger this webhook?

● Just the `push` event.

○ Send me **everything**.

○ Let me select individual events.

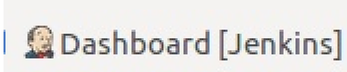
☑ **Active**
We will deliver event details when this hook is triggered.

7. Leave the remaining settings to the default values and click "**Add webhook**".

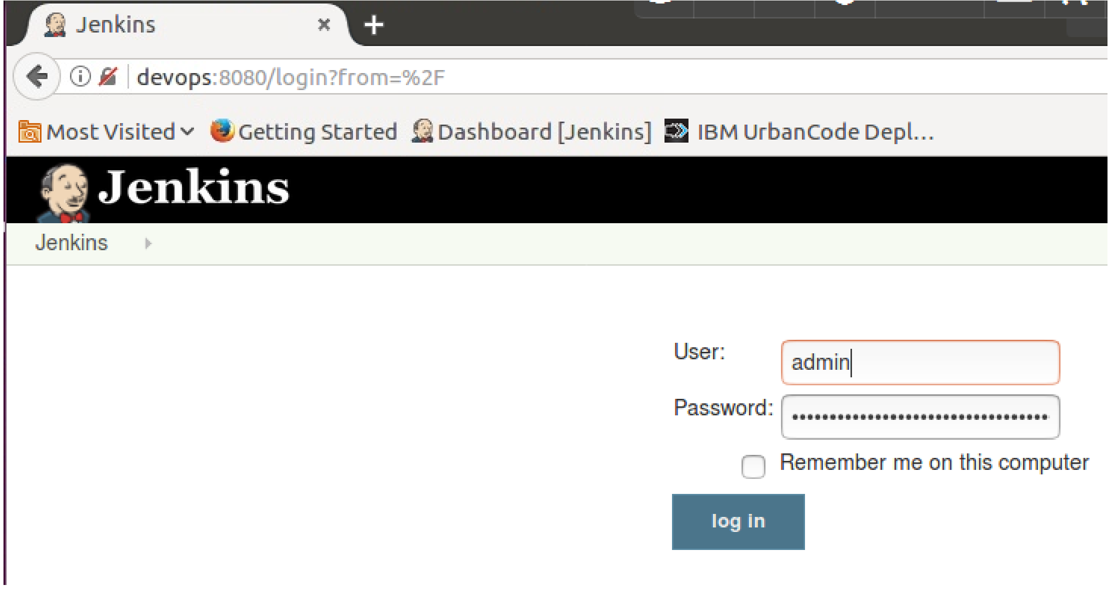
Lab 4: IBM UrbanCode Deploy and Jenkins integration

Lab 4.1 – Configure the IBM UrbanCode Deploy and Jenkins integration

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



2. Login to Jenkins using **admin** / **admin**.

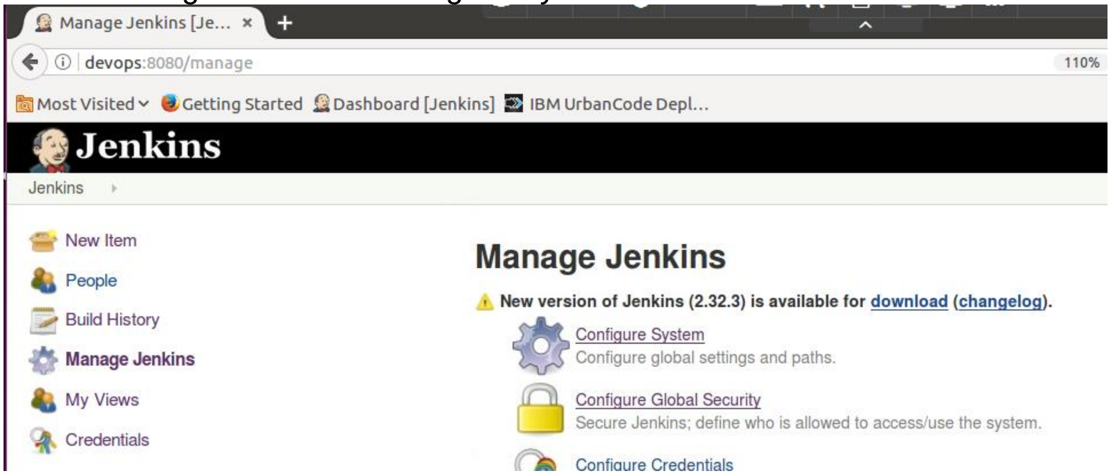


User: admin

Password: ••••••••••••••••••••••••••••••••••

□ Remember me on this computer

3. Select Manage Jenkins > Configure System.



Manage Jenkins

New version of Jenkins (2.32.3) is available for download (changelog).

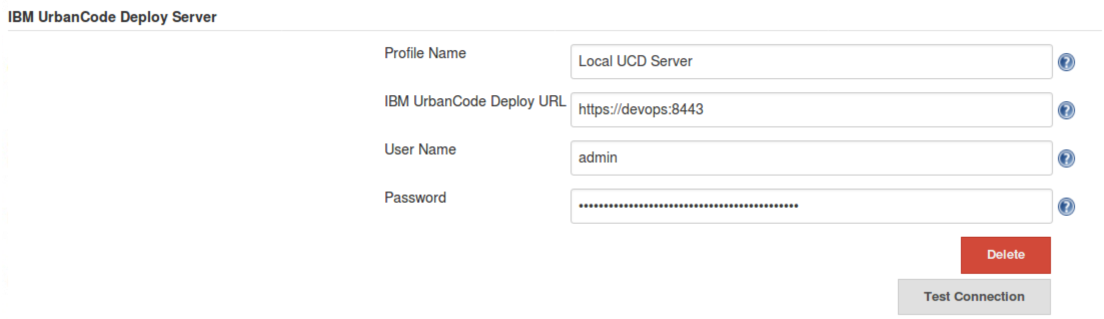
Configure System
Configure global settings and paths.

Configure Global Security
Secure Jenkins; define who is allowed to access/use the system.

Configure Credentials

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

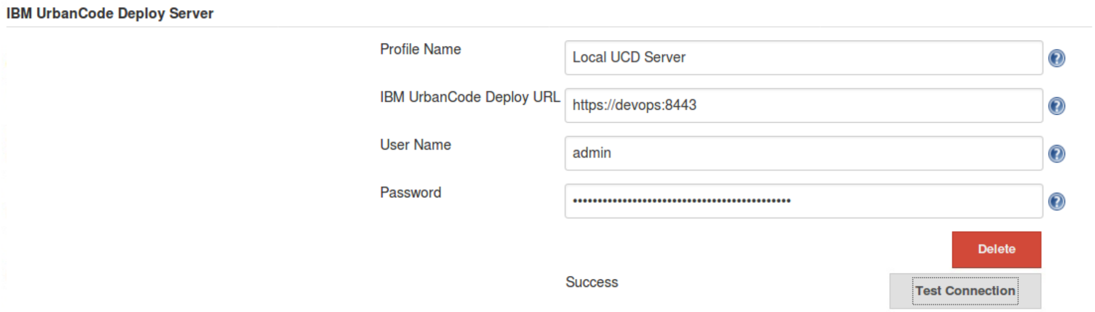
Profile Name: Local UCD Server

IBM UrbanCode Deploy URL: https://devops:8443

User Name: admin

Password: ••

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



IBM UrbanCode Deploy Server

Profile Name: Local UCD Server

IBM UrbanCode Deploy URL: https://devops:8443

User Name: admin

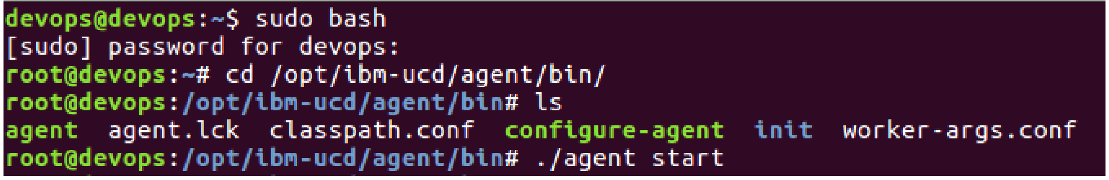
Password: ••

Success

Lab 5: IBM UrbanCode Deploy Configuration

Lab 5.1 – Start the IBM UrbanCode Deploy Agent as root

1. Open a terminal window and type in “sudo bash” to run as root. Enter devops when prompted for the password.
2. Go to the UCD Agent bin directory (/opt/ibm-ucd/agent/bin) and run ./agent start.



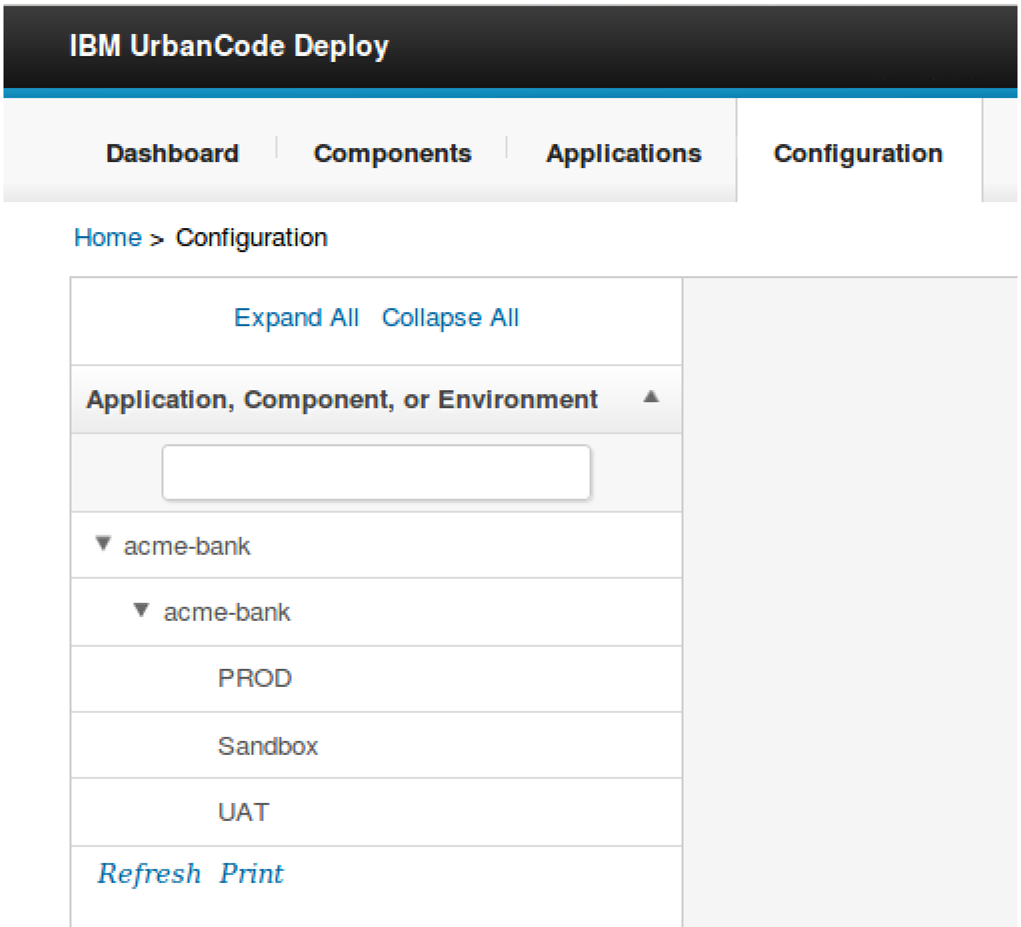
```
devops@devops:~$ sudo bash
[sudo] password for devops:
root@devops:~# cd /opt/ibm-ucd/agent/bin/
root@devops:/opt/ibm-ucd/agent/bin# ls
agent  agent.lck  classpath.conf  configure-agent  init  worker-args.conf
root@devops:/opt/ibm-ucd/agent/bin# ./agent start
```

Lab 5.2 - Update Environment Properties

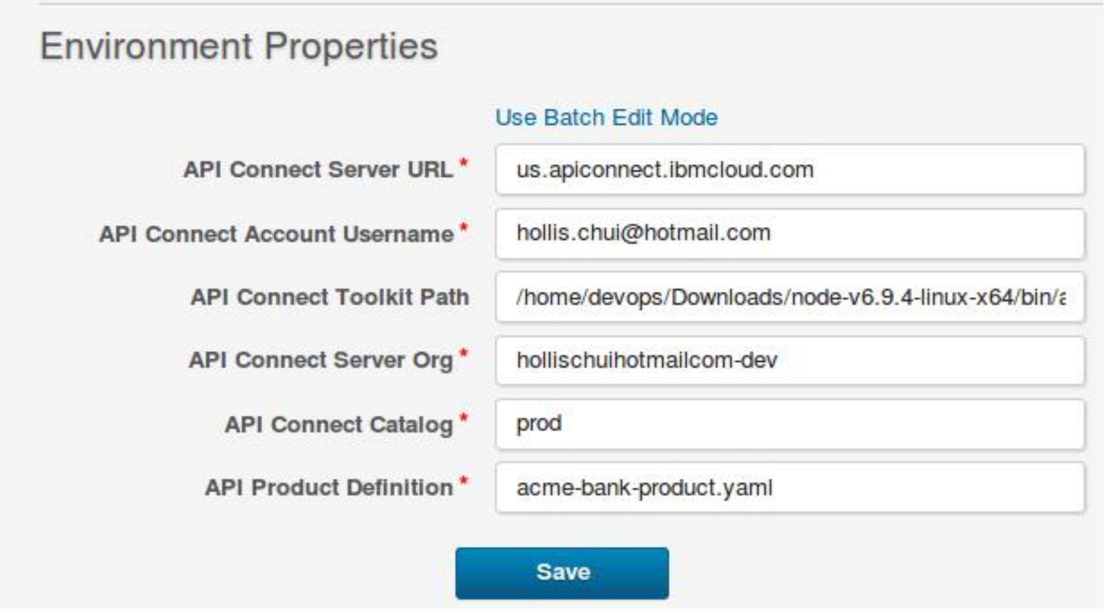
1. Open a Firefox browser and click on the IBM UrbanCode Deploy bookmark. Log in as admin / admin.

IBM UrbanCode Depl...

2. Select the Configuration tab and verify there are 3 environments listed under acme-bank.



3. For the Sandbox environment, update the following Environment Properties for each Environment based on the Catalog Identifier information from the API Connect service.
    i. API Connect Account Username = <*Your Bluemix account username*>
    ii. API Connect Server Org = <*API Connect Org*>
    iii. API Connect Catalog = **sb**



Environment Properties

API Connect Server URL * us.apiconnect.ibmcloud.com

API Connect Account Username * hollis.chui@hotmail.com

API Connect Toolkit Path /home/devops/Downloads/node-v6.9.4-linux-x64/bin/a

API Connect Server Org * hollischuihotmailcom-dev

API Connect Catalog * prod

API Product Definition * acme-bank-product.yaml

4. For the UAT environment, update the following Environment Properties for each Environment based on the Catalog Identifier information from the API Connect service.
    i. API Connect Account Username = <*Your Bluemix account username*>
    ii. API Connect Server Org = <*API Connect Org*>
    iii. API Connect Catalog = **uat**
5. For the PROD environment, update the following Environment Properties for each Environment based on the Catalog Identifier information from the API Connect service.
    i. API Connect Account Username = <*Your Bluemix account username*>
    ii. API Connect Server Org = <*API Connect Org*>
    iii. API Connect Catalog = **prod**

Lab 6: Demonstrate API Governance

Lab 6.1: Major release flow

The development team is ready to publish version 1.1.0 of the acme-bank Product. Upon a push to the GitHub repository with an annotated tag containing the keyword "**major**", the Jenkins build will be triggered.

The push will trigger a Jenkins build which will create a UCD Component Version for acme-bank:1.0.0, upload the yaml files and call the same "Deploy API Product Def" process to publish Product into the Sandbox Catalog.

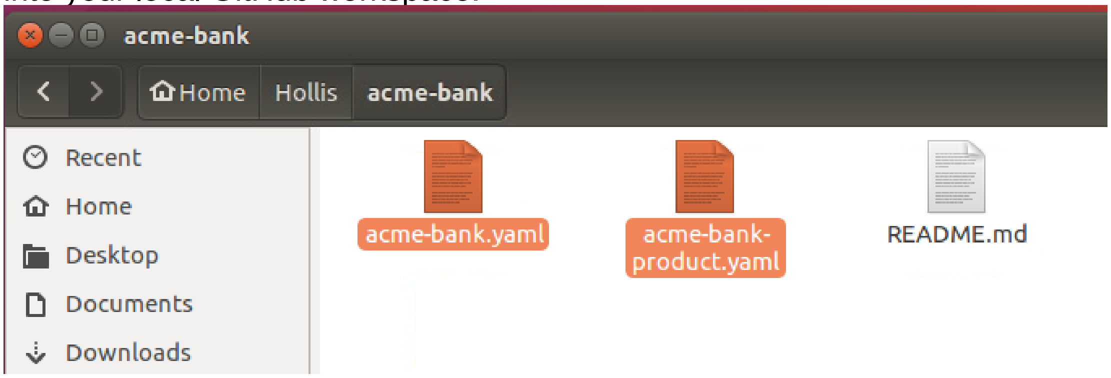
As this is a major version change, the component process will run the **apic products:publish** cli command to publish the acme-bank:1.0.0 Product definition.

On the Desktop, there is a directory called "**acme-bank**" which contains 2 yaml files (acme-bank-product.yaml and acme-bank.yaml).

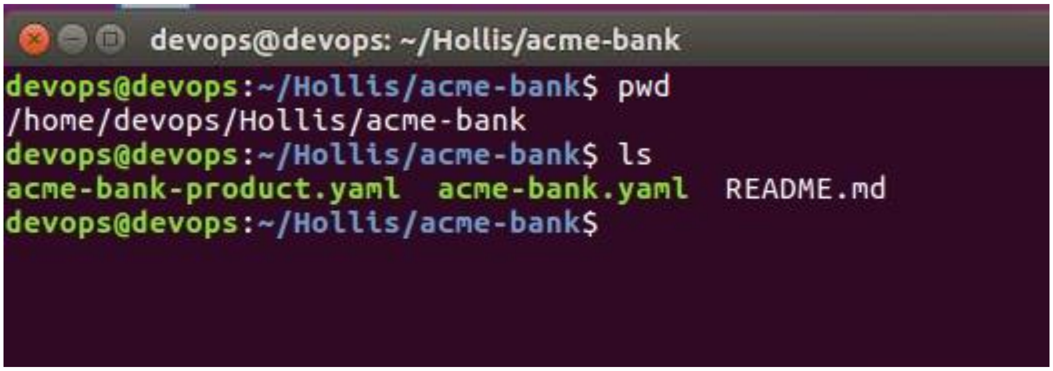
- *acme-bank-product.yaml* = Product definition
- acme-bank.yaml = API definition

To illustrate the API governance model discussed in the Overview section, we will only be dealing with the Product and API yaml and not the actual API Implementation.

1. Copy and paste the *acme-bank-product.yaml* and *acme-bank.yaml* into your local GitHub workspace.

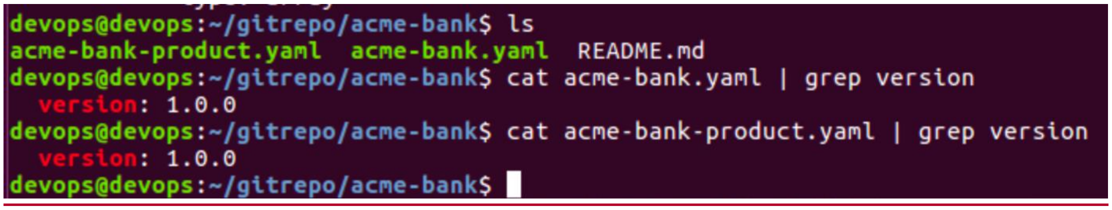


2. Open a terminal window and browse to the GitHub workspace directory.



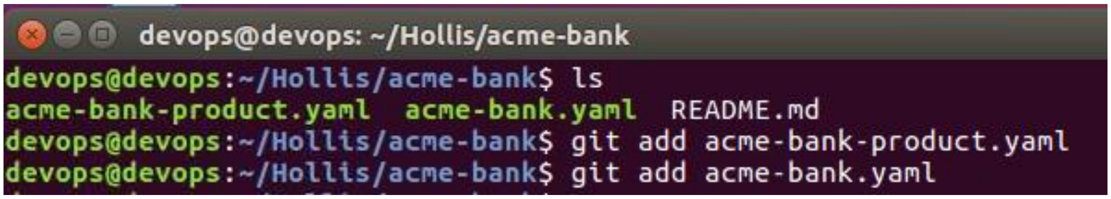
```
devops@devops: ~/Hollis/acme-bank
devops@devops:~/Hollis/acme-bank$ pwd
/home/devops/Hollis/acme-bank
devops@devops:~/Hollis/acme-bank$ ls
acme-bank-product.yaml  acme-bank.yaml  README.md
devops@devops:~/Hollis/acme-bank$
```

3. Verify the version property in both files are set to 1.0.0 by running the command “**cat <yaml file> | grep version**”**.**



```
devops@devops:~/gitrepo/acme-bank$ ls
acme-bank-product.yaml  acme-bank.yaml  README.md
devops@devops:~/gitrepo/acme-bank$ cat acme-bank.yaml | grep version
  version: 1.0.0
devops@devops:~/gitrepo/acme-bank$ cat acme-bank-product.yaml | grep version
  version: 1.0.0
devops@devops:~/gitrepo/acme-bank$ 
```

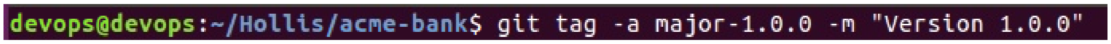
4. Add the two yaml files to the GitHub repository by running the command “**git add <yaml file>**”**.**



```
devops@devops: ~/Hollis/acme-bank
devops@devops:~/Hollis/acme-bank$ ls
acme-bank-product.yaml  acme-bank.yaml  README.md
devops@devops:~/Hollis/acme-bank$ git add acme-bank-product.yaml
devops@devops:~/Hollis/acme-bank$ git add acme-bank.yaml
```

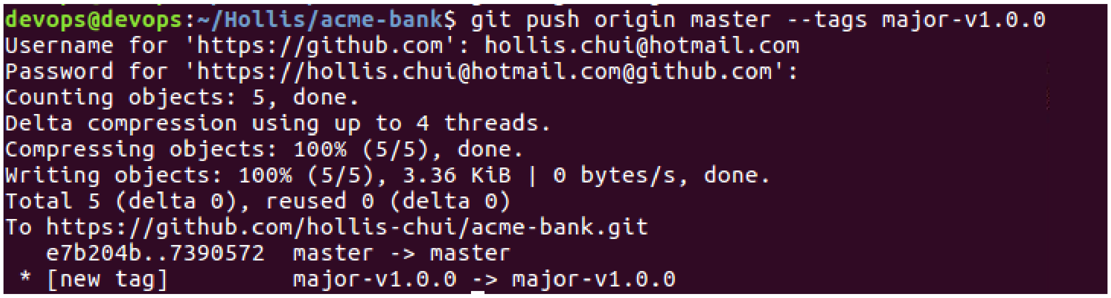
5. Commit the two yaml files to the GitHub repository by running the command “git commit -a -m <Description>”.

6. Create an annotated tag with the keyword “major” to associate with the push. The command is “**git tag -a major-1.0.0**” -m “Version 1.0.0 release”.



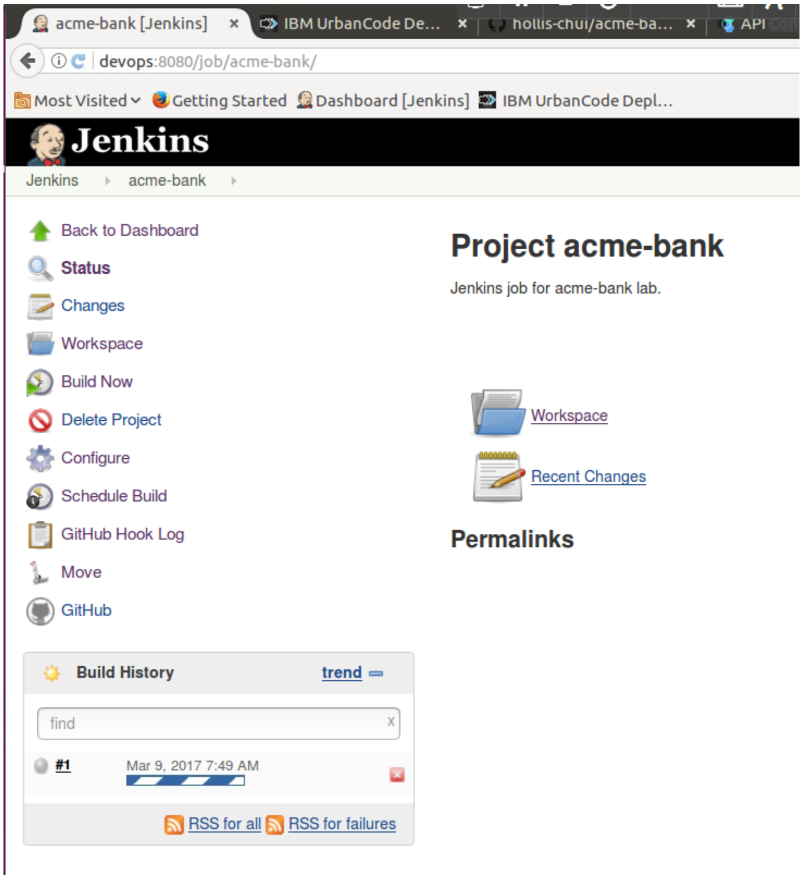
```
devops@devops:~/Hollis/acme-bank$ git tag -a major-1.0.0 -m "Version 1.0.0"
```

7. Push the changes with the tag into the Github repository by running the command “**git push origin master –tags <tag>**”.



```
devops@devops:~/Hollis/acme-bank$ git push origin master --tags major-v1.0.0
Username for 'https://github.com': hollis.chui@hotmail.com
Password for 'https://hollis.chui@hotmail.com@github.com':
Counting objects: 5, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 3.36 KiB | 0 bytes/s, done.
Total 5 (delta 0), reused 0 (delta 0)
To https://github.com/hollis-chui/acme-bank.git
   e7b204b..7390572  master -> master
 * [new tag]         major-v1.0.0 -> major-v1.0.0
```

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



Project acme-bank

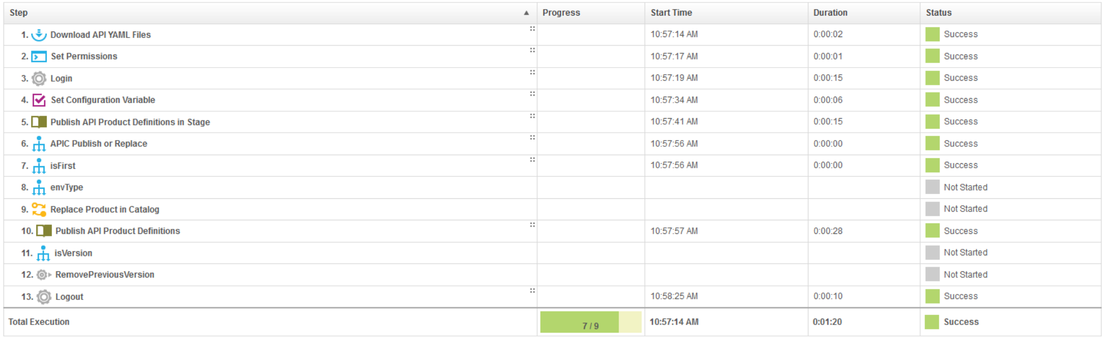
Jenkins job for acme-bank lab.

Permalinks

#1 Mar 9, 2017 7:49 AM

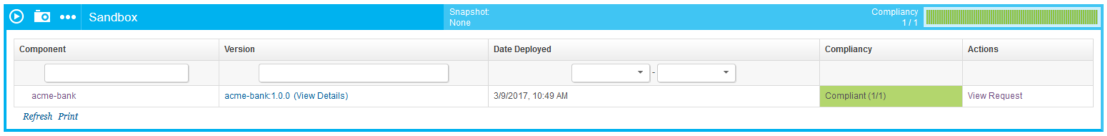
9. Verify the "**Deploy API Product Def**" component process completed successfully.
   a. Open the Firefox browser and click on the IBM UrbanCode Deploy bookmark.
   b. Log in using admin / admin.
   c. In the UCD web client, click on the Components tab and select the **acme-bank** component.
   d. 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:1.0.0
iii. Environment = Sandbox



Step	Progress	Start Time	Duration	Status
1. Download API YAML Files		10:57:14 AM	0:00:02	Success
2. Set Permissions		10:57:17 AM	0:00:01	Success
3. Login		10:57:19 AM	0:00:15	Success
4. Set Configuration Variable		10:57:34 AM	0:00:06	Success
5. Publish API Product Definitions in Stage		10:57:41 AM	0:00:15	Success
6. APIC Publish or Replace		10:57:56 AM	0:00:00	Success
7. isFirst		10:57:56 AM	0:00:00	Success
8. envType				Not Started
9. Replace Product in Catalog				Not Started
10. Publish API Product Definitions		10:57:57 AM	0:00:28	Success
11. isVersion				Not Started
12. RemovePreviousVersion				Not Started
13. Logout		10:58:25 AM	0:00:10	Success
Total Execution	7 / 9	10:57:14 AM	0:01:20	Success

e. Verify the acme-bank:1.0.0 component version has been deployed into the Sandbox Environment by selecting Applications > acme-bank and expand the Sandbox environment to view the deployed component version.

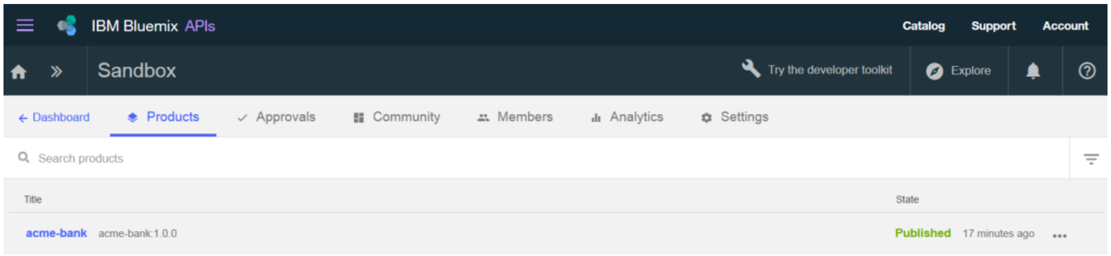


Sandbox
Snapshot: None
Compliancy 1 / 1

Component	Version	Date Deployed	Compliancy	Actions
acme-bank	acme-bank:1.0.0 (View Details)	3/9/2017, 10:49 AM	Compliant (1/1)	View Request

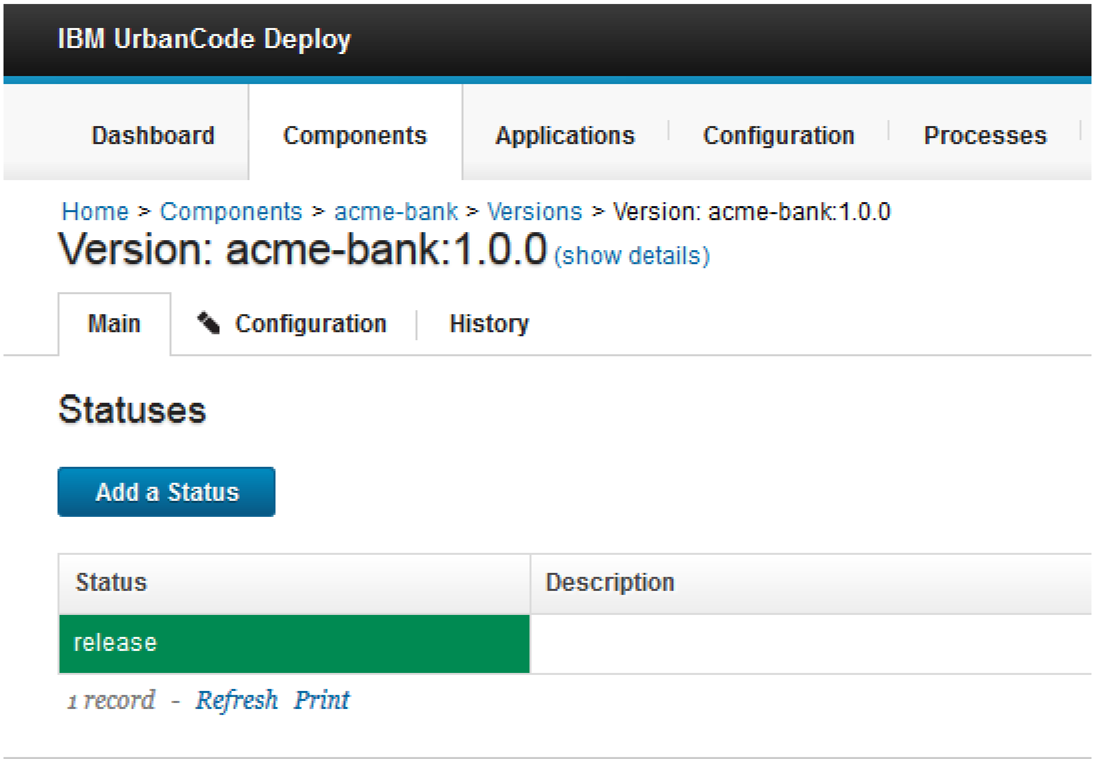
Refresh Print

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 Sandbox Catalog.



Title	State
acme-bank acme-bank:1.0.0	Published 17 minutes ago

10. Once the API has been published to API Connect Sandbox Catalog, teams can perform their testing and upon completion, the API Product owner will add the “**release**” Version Status to the **acme-bank:1.0.0** component version and manually trigger the automated process for deployment into the UAT environment / catalog.
    a. Open a Firefox browser and click the IBM UrbanCode Deploy bookmark. Log in with admin / admin.
    b. Click on the Components tab and select the **acme-bank** component.
    c. Click on the Versions tab and select the **acme-bank:1.0.0** component version.
    d. On the Main tab, click the “**Add a Status**” button, select “**release**” Status in the dropdown menu and click Save.



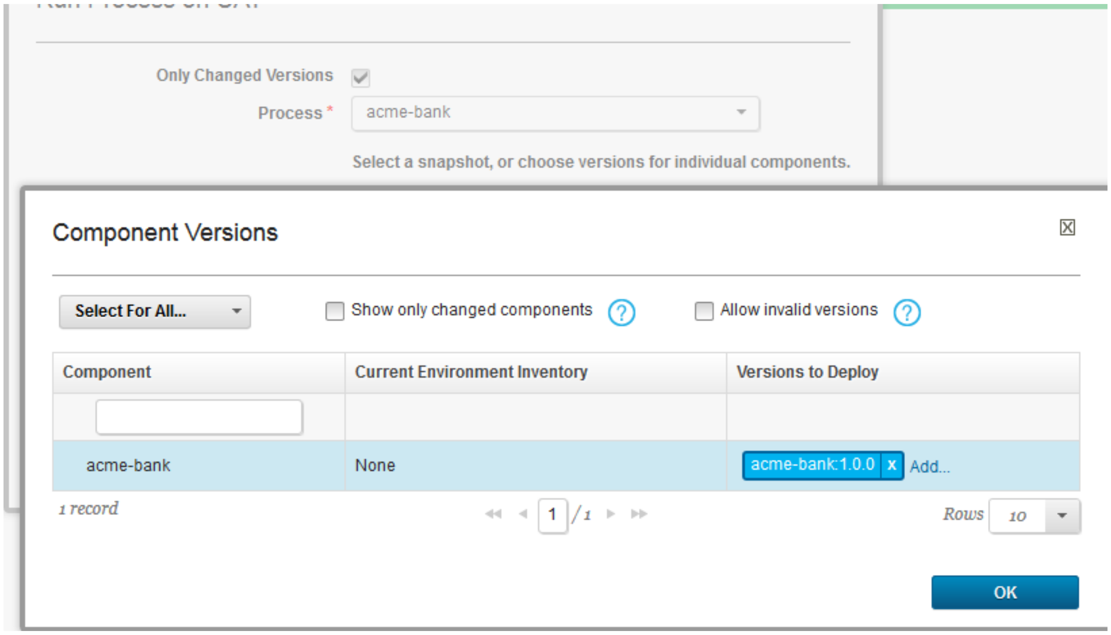
Home > Components > acme-bank > Versions > Version: acme-bank:1.0.0

Version: acme-bank:1.0.0 (show details)

Statuses

Status	Description
release	

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



Only Changed Versions ☑

Process * acme-bank

Select a snapshot, or choose versions for individual components.

Component Versions ☒

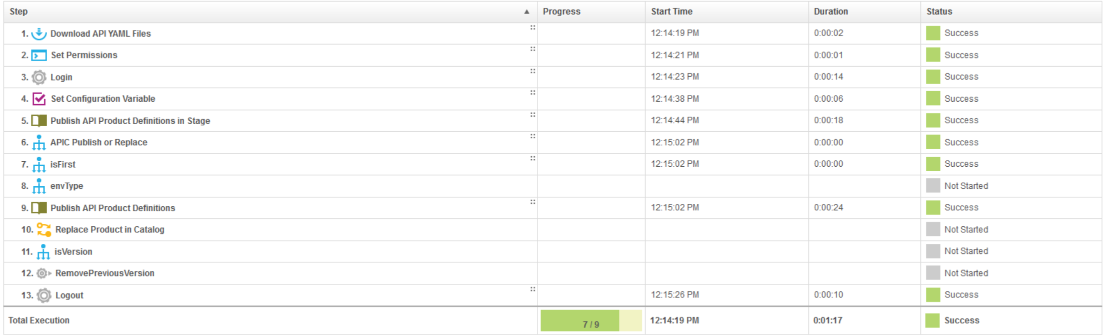
Select For All... ▾ ☐ Show only changed components ? ☐ Allow invalid versions ?

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

1 record ◂◂ ◂ 1 / *1* ▸ ▸▸ *Rows* *10* ▾

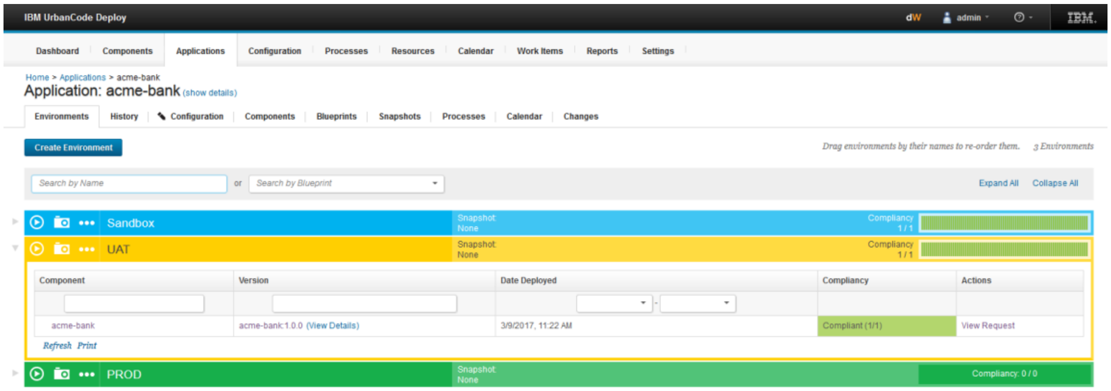
OK

12. 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:1.0.0
        iii. Environment = UAT



Step ▲	Progress	Start Time	Duration	Status
1. Download API YAML Files		12:14:19 PM	0:00:02	🟩 Success
2. Set Permissions		12:14:21 PM	0:00:01	🟩 Success
3. Login		12:14:23 PM	0:00:14	🟩 Success
4. Set Configuration Variable		12:14:38 PM	0:00:06	🟩 Success
5. Publish API Product Definitions in Stage		12:14:44 PM	0:00:18	🟩 Success
6. APIC Publish or Replace		12:15:02 PM	0:00:00	🟩 Success
7. isFirst		12:15:02 PM	0:00:00	🟩 Success
8. envType				⬜ Not Started
9. Publish API Product Definitions		12:15:02 PM	0:00:24	🟩 Success
10. Replace Product in Catalog				⬜ Not Started
11. isVersion				⬜ Not Started
12. RemovePreviousVersion				⬜ Not Started
13. Logout		12:15:26 PM	0:00:10	🟩 Success
Total Execution	7 / 9	12:14:19 PM	0:01:17	🟩 Success

    e. Verify the acme-bank:1.0.0 component version has been deployed into the UAT Environment by selecting Applications > acme-bank and expand the UAT environment to view the deployed component version.



Application: acme-bank (show details)

Drag environments by their names to re-order them. 3 Environments

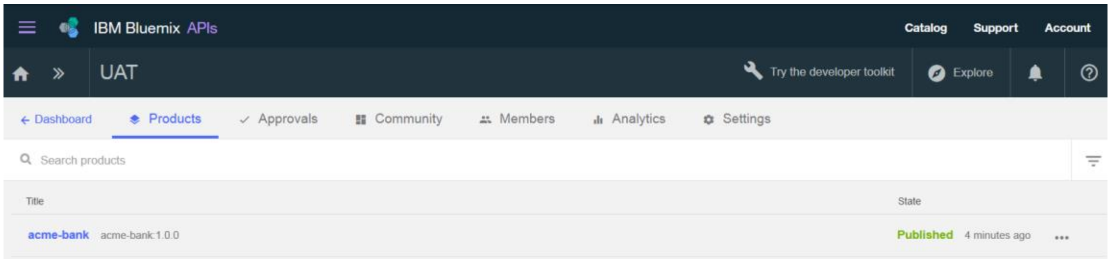
Sandbox — Snapshot: None — Compliancy 1 / 1

UAT — Snapshot: None — Compliancy 1 / 1

Component	Version	Date Deployed	Compliancy	Actions
acme-bank	acme-bank:1.0.0 (View Details)	3/9/2017, 11:22 AM	Compliant (1/1)	View Request

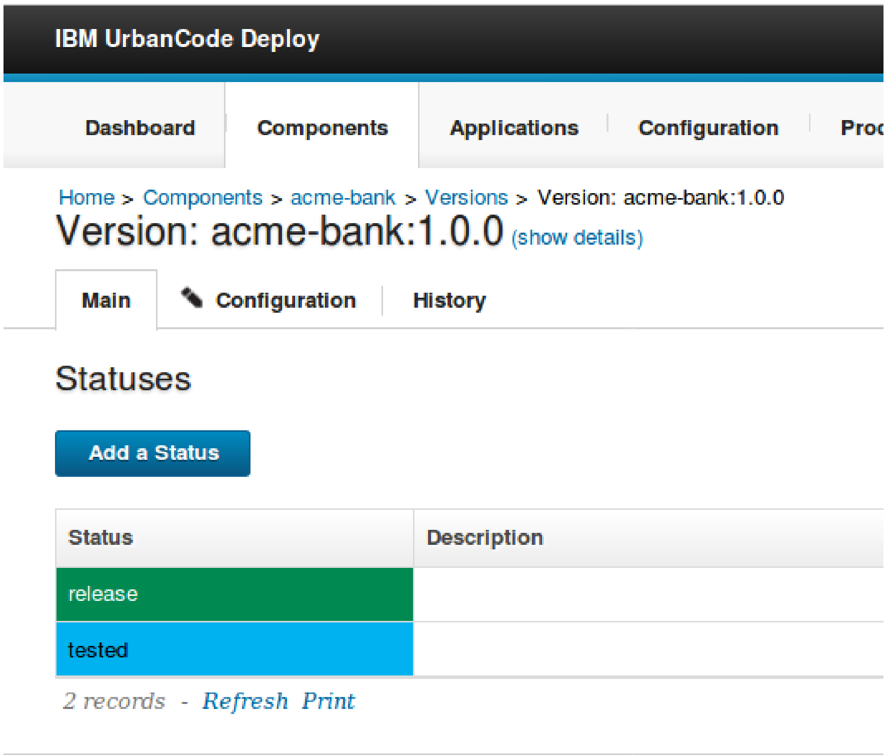
PROD — Snapshot: None — Compliancy: 0 / 0

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.



Title		State	
acme-bank	acme-bank:1.0.0	Published	4 minutes ago

13. 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 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.
    b. Click on the Components tab and select the **acme-bank** component.
    c. Click on the Versions tab and select the **acme-bank:1.0.0** component version.
    d. 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

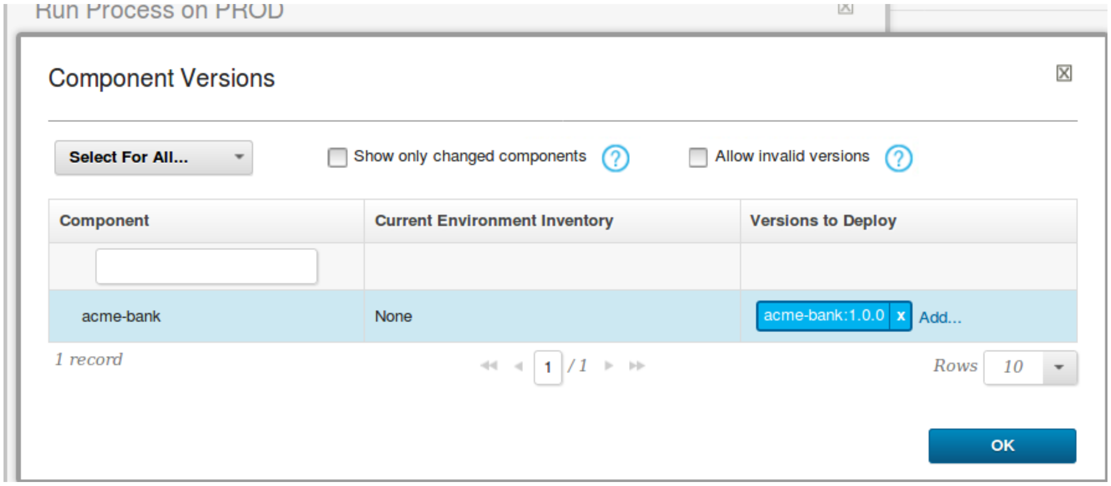
Version: acme-bank:1.0.0 (show details)

Statuses

Status	Description
release	
tested	

2 records - Refresh Print

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



Run Process on PROD

Component Versions

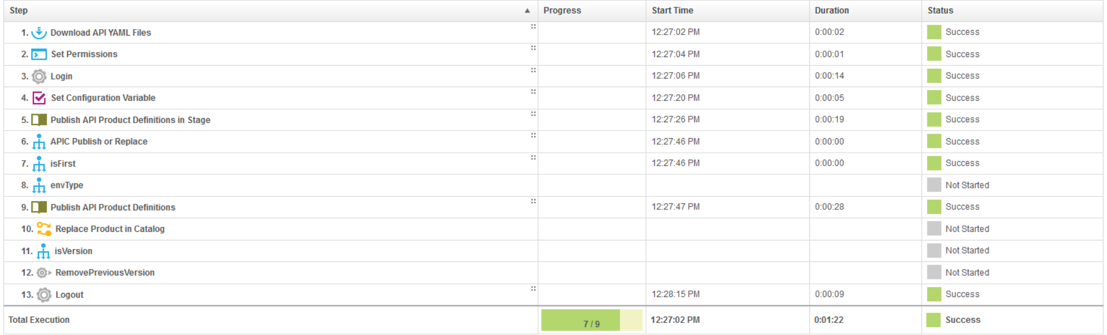
□ Show only changed components

□ Allow invalid versions

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

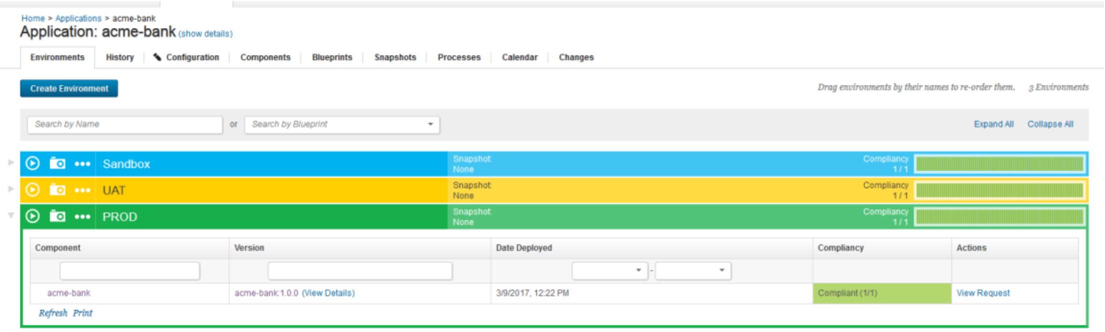
1 record

15. 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



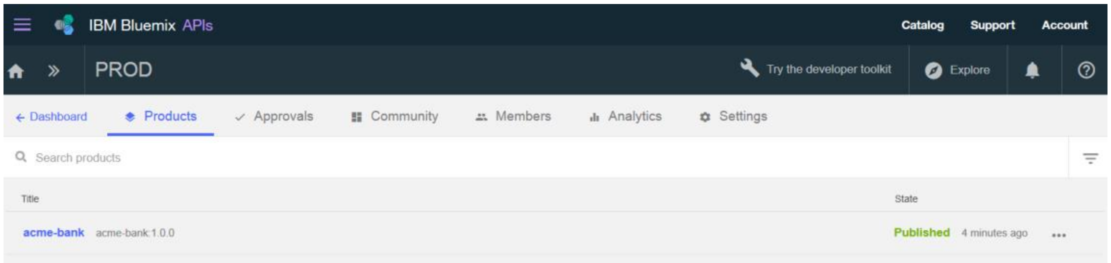
Step	Progress	Start Time	Duration	Status
1. Download API YAML Files		12:27:02 PM	0:00:02	🟩 Success
2. Set Permissions		12:27:04 PM	0:00:01	🟩 Success
3. Login		12:27:06 PM	0:00:14	🟩 Success
4. Set Configuration Variable		12:27:20 PM	0:00:05	🟩 Success
5. Publish API Product Definitions in Stage		12:27:26 PM	0:00:19	🟩 Success
6. APIC Publish or Replace		12:27:46 PM	0:00:00	🟩 Success
7. isFirst		12:27:46 PM	0:00:00	🟩 Success
8. envType				⬜ Not Started
9. Publish API Product Definitions		12:27:47 PM	0:00:28	🟩 Success
10. Replace Product in Catalog				⬜ Not Started
11. isVersion				⬜ Not Started
12. RemovePreviousVersion				⬜ Not Started
13. Logout		12:28:15 PM	0:00:09	🟩 Success
Total Execution	7 / 9	12:27:02 PM	0:01:22	🟩 Success

    e. Verify the acme-bank:1.0.0 component version has been deployed into the PROD Environment by selecting Applications > acme-bank and expand the PROD environment to view the deployed component version.



Component	Version	Date Deployed	Compliancy	Actions
acme-bank	acme-bank:1.0.0 (View Details)	3/9/2017, 12:22 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	4 minutes ago

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

Lab 4.2: Minor 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 with an annotated tag containing the keyword “**minor**”, the Jenkins build will be triggered.

The push will trigger a Jenkins build using the same job configuration as the Major release flow. The Jenkins build will create a UCD Component Version for acme-bank:1.1.0, upload the yaml files and call the same “Deploy API Product Def” process to replace the API Product definition into the Sandbox Catalog.

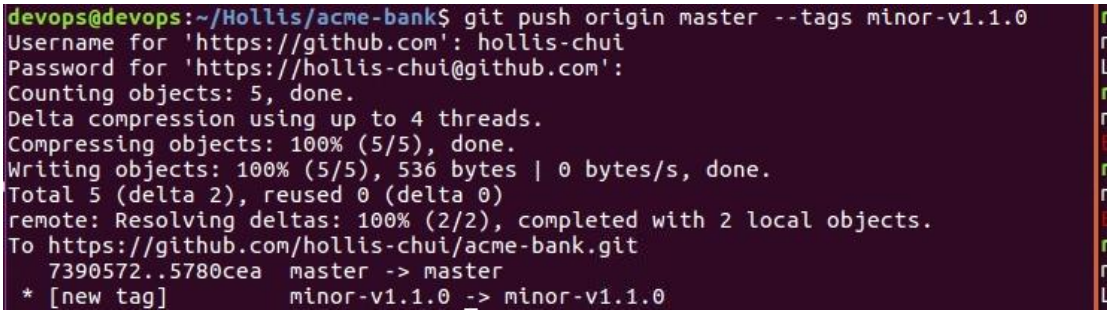
As this is a minor version change, the component process will run the **apic products:replace** cli command to replace acme-bank:1.0.0 with the acme-bank:1.1.0 API Product definition.

1. Edit the acme-bank-product.yaml and acme-bank.yaml files in a text editor and modify the **Version** from **1.0.0** to **1.1.0**. Verify the change by running the command “**cat <yaml file> | grep version**”.

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

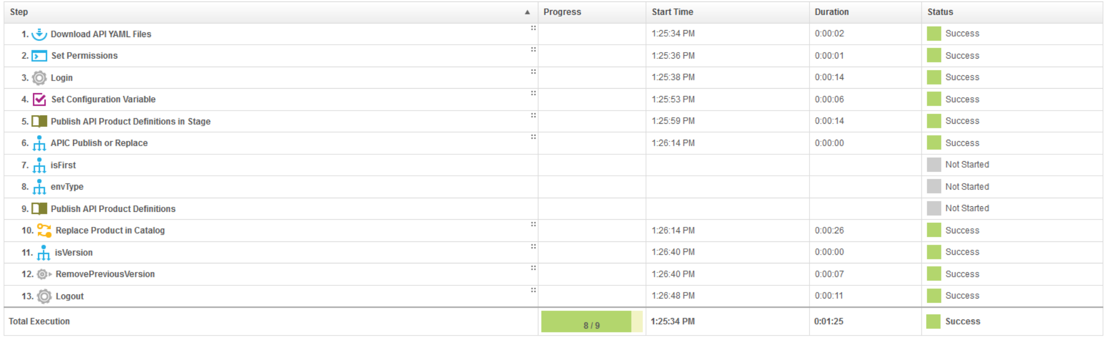
3. Create an annotated tag with the keyword “major” to associate with the push. The command is “**git tag -a minor-1.1.0**” -m “Version 1.1.0 release”.

4. Push the changes with the tag into the Github repository by running the command “**git push origin master –tags <tag>**”.



```
devops@devops:~/Hollis/acme-bank$ git push origin master --tags minor-v1.1.0
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% (5/5), done.
Writing objects: 100% (5/5), 536 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.git
   7390572..5780cea  master -> master
 * [new tag]         minor-v1.1.0 -> minor-v1.1.0
```

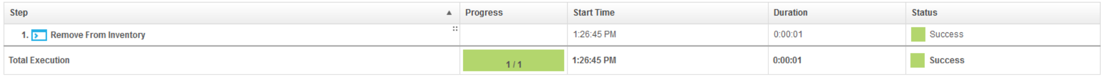
5. Once the changes are pushed into the GitHub repository, this will trigger the Jenkins build to deploy the minor update to the Sandbox environment / catalog. Verify the “**Deploy API Product Def**” component process completed successfully.
   a. Open the Firefox browser and click on the IBM UrbanCode 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

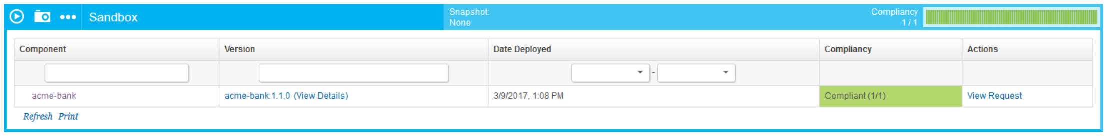
   e. If you notice carefully, there is an additional step that is run called “**RemovePreviousVersion**”. This step will remove the acme-bank:1.0.0 component version from the Sandbox Environment to keep the Environment in sync with what is available in the Sandbox Catalog. Verify this step completed successfully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

i. Process = RemoveFromInventory
ii. Version = acme-bank:1.1.0
iii. 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

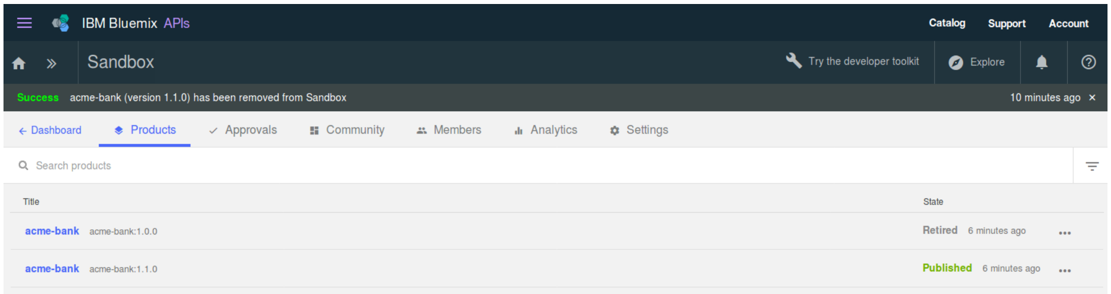
f. Verify the acme-bank:1.1.0 component version has been deployed into the Sandbox Environment by selecting Applications > acme-bank and expand the Sandbox environment to view the deployed component version.



Component	Version	Date Deployed	Compliancy	Actions
acme-bank	acme-bank:1.1.0 (View Details)	3/9/2017, 1:08 PM	Compliant (1/1)	View Request

Refresh Print

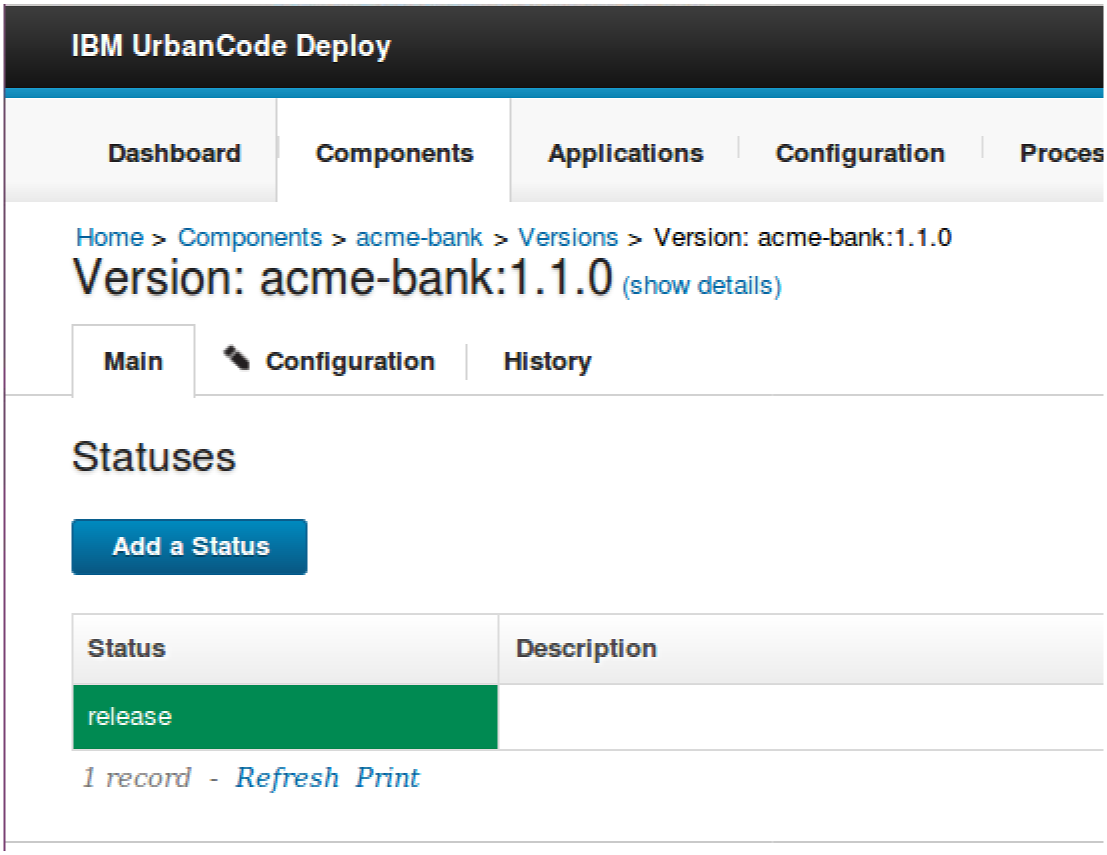
g. Finally, validate the acme-bank:1.1.0 Product is published on API Connect by logging into the API Connect Dashboard and selecting the Sandbox Catalog.



Success acme-bank (version 1.1.0) has been removed from Sandbox 10 minutes ago ×

Title	State
acme-bank acme-bank:1.0.0	Retired 6 minutes ago
acme-bank acme-bank:1.1.0	Published 6 minutes ago

6. Once the API has been published to API Connect Sandbox Catalog, teams can perform their testing and upon completion, the API Product owner will add the "**release**" Version Status to the **acme-bank:1.1.0** component version and manually trigger the automated process for deployment into the UAT environment / catalog.
   a. Open a Firefox browser and click the IBM UrbanCode Deploy bookmark. Log in with admin / admin.
   b. Click on the Components tab and select the **acme-bank** component.
   c. Click on the Versions tab and select the **acme-bank:1.1.0** component version.
   d. On the Main tab, click the "**Add a Status**" button, select "**release**" Status in the dropdown menu and click Save.



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

Version: acme-bank:1.1.0 (show details)

Statuses

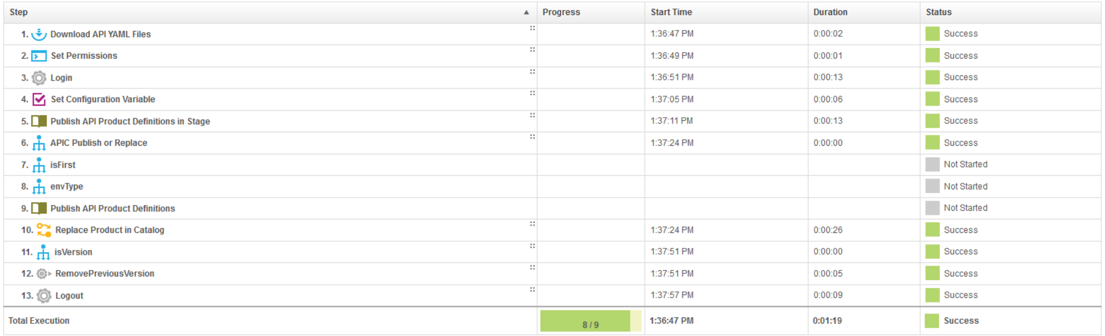
Status	Description
release	

1 record - Refresh Print

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

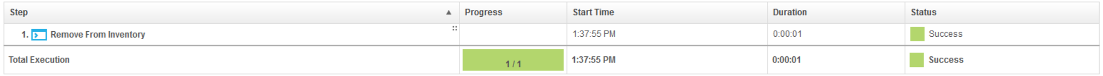
8. Verify the “**Deploy API Product Def**” component process completed successfully on the UAT environment.
    a. Open the Firefox browser and click on the IBM UrbanCode Deploy bookmark.
    b. Log in using admin / admin.
    c. In the UCD web client, click on the Components tab and select the **acme-bank** 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



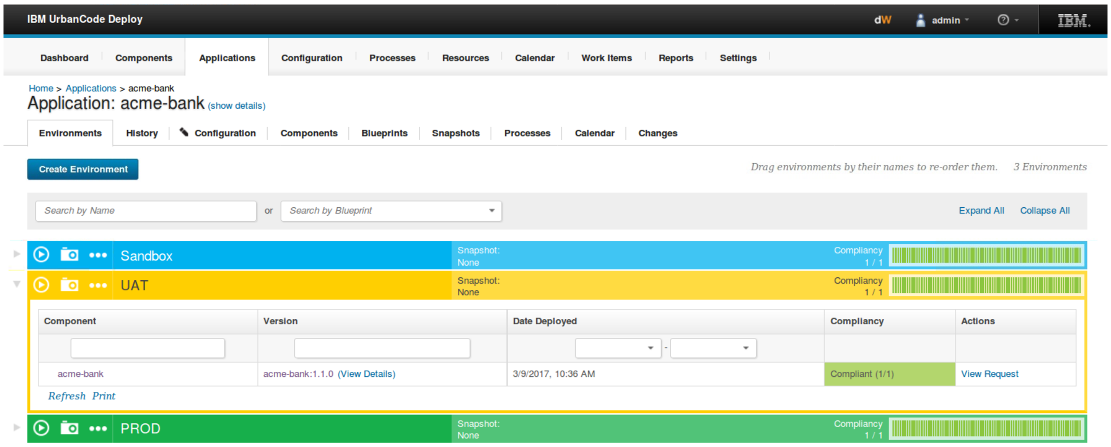
Step	Progress	Start Time	Duration	Status
1. Download API YAML Files		1:36:47 PM	0:00:02	🟩 Success
2. Set Permissions		1:36:49 PM	0:00:01	🟩 Success
3. Login		1:36:51 PM	0:00:13	🟩 Success
4. Set Configuration Variable		1:37:05 PM	0:00:06	🟩 Success
5. Publish API Product Definitions in Stage		1:37:11 PM	0:00:13	🟩 Success
6. APIC Publish or Replace		1:37:24 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		1:37:24 PM	0:00:26	🟩 Success
11. isVersion		1:37:51 PM	0:00:00	🟩 Success
12. RemovePreviousVersion		1:37:51 PM	0:00:05	🟩 Success
13. Logout		1:37:57 PM	0:00:09	🟩 Success
Total Execution	8 / 9	1:36:47 PM	0:01:19	🟩 Success

e. 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:1.1.0
   iii. Environment = UAT



Step	Progress	Start Time	Duration	Status
1. Remove From Inventory		1:37:55 PM	0:00:01	🟩 Success
Total Execution	1 / 1	1:37:55 PM	0:00:01	🟩 Success

f. Verify the acme-bank:1.1.0 component version has been deployed into the UAT Environment by selecting Applications > acme-bank and expand the UAT environment to view the deployed component version.



Application: acme-bank (show details)

Drag environments by their names to re-order them. 3 Environments

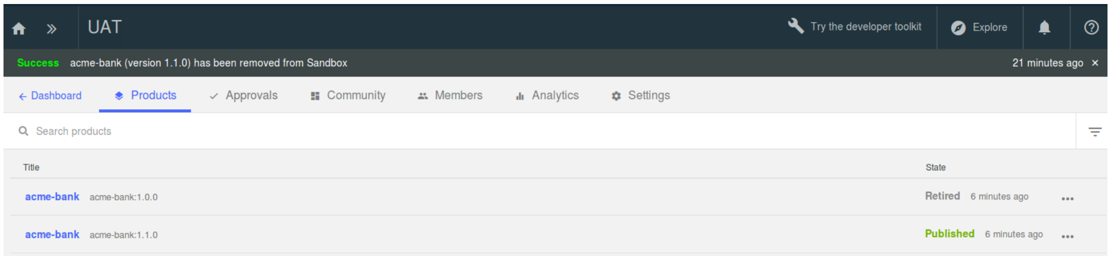
Sandbox — Snapshot: None — Compliancy 1 / 1

UAT — Snapshot: None — Compliancy 1 / 1

Component	Version	Date Deployed	Compliancy	Actions
acme-bank	acme-bank:1.1.0 (View Details)	3/9/2017, 10:36 AM	Compliant (1/1)	View Request

PROD — Snapshot: None — Compliancy 1 / 1

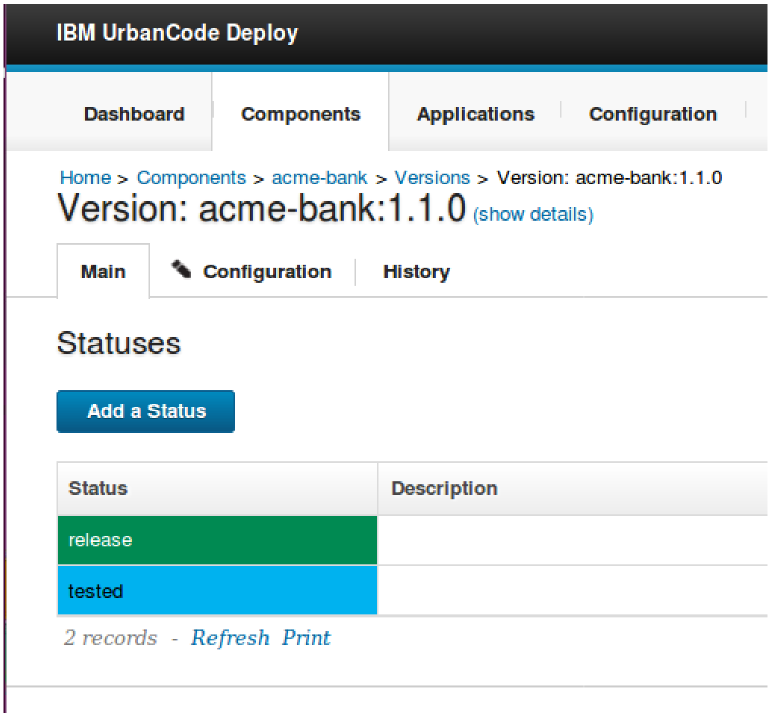
g. Finally, validate the acme-bank:1.1.0 Product is published on API Connect by logging into the API Connect Dashboard and selecting the UAT Catalog.



Success acme-bank (version 1.1.0) has been removed from Sandbox 21 minutes ago ×

Title		State		
acme-bank	acme-bank:1.0.0	Retired	6 minutes ago	...
acme-bank	acme-bank:1.1.0	Published	6 minutes ago	...

9. 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 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.
    b. Click on the Components tab and select the **acme-bank** component.
    c. Click on the Versions tab and select the **acme-bank:1.1.0** component version.
    d. On the Main tab, click the "**Add a Status**" button, select "**tested**" Status in the dropdown menu and click Save.



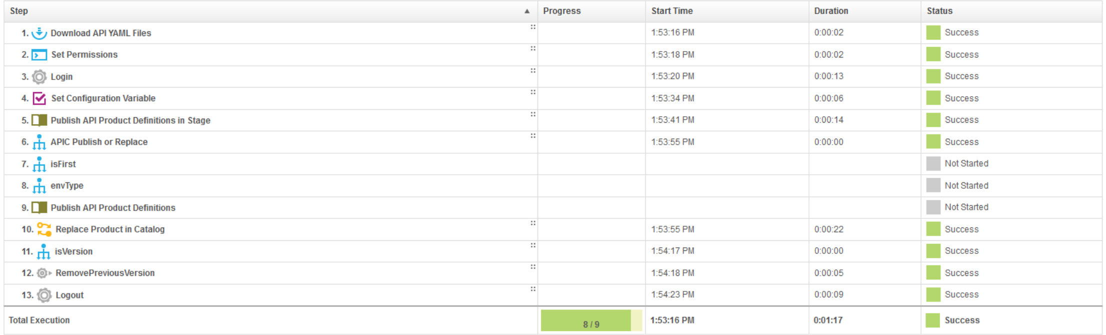
Status	Description
release	
tested	

2 records - 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 PROD environment / catalog.
    a. Open a Firefox browser and click the IBM UrbanCode Deploy bookmark. Log in with admin / admin.

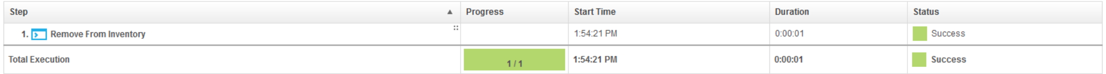
b. Click on the Applications tab and select the **acme-bank** application.
c. Click the play button for the PROD environment and the “Run Process on PROD” window should appear.
d. For the Process field, select **acme-bank**.
e. For the Versions, click on the *Choose Versions* link and a Component Versions window will pop up.
f. Under the Versions to Deploy column, select the *Add…* link.
g. 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 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:1.1.0
        iii. Environment = PROD



Step	Progress	Start Time	Duration	Status
1. Download API YAML Files		1:53:16 PM	0:00:02	🟩 Success
2. Set Permissions		1:53:18 PM	0:00:02	🟩 Success
3. Login		1:53:20 PM	0:00:13	🟩 Success
4. Set Configuration Variable		1:53:34 PM	0:00:06	🟩 Success
5. Publish API Product Definitions in Stage		1:53:41 PM	0:00:14	🟩 Success
6. APIC Publish or Replace		1:53:55 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		1:53:55 PM	0:00:22	🟩 Success
11. isVersion		1:54:17 PM	0:00:00	🟩 Success
12. RemovePreviousVersion		1:54:18 PM	0:00:05	🟩 Success
13. Logout		1:54:23 PM	0:00:09	🟩 Success
Total Execution	8 / 9	1:53:16 PM	0:01:17	🟩 Success

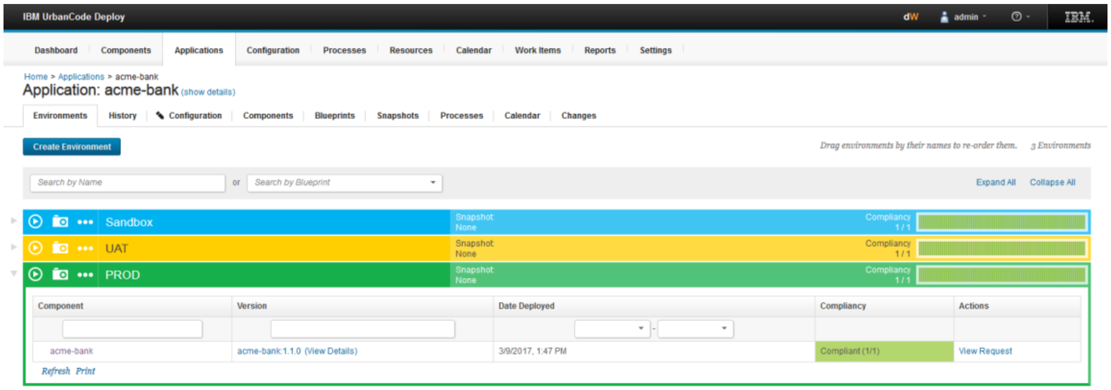
    e. Verify the “**RemovePreviousVersion**” component process completed successfully on the PROD environment and removed the acme-bank:1.0.0 component version. Under the Component Request History section, find the following entry and click on the **View Request** link.
        i. Process = RemoveFromInventory
        ii. Version = acme-bank:1.1.0
        iii. Environment = PROD



Step	Progress	Start Time	Duration	Status
1. Remove From Inventory		1:54:21 PM	0:00:01	🟩 Success
Total Execution	1 / 1	1:54:21 PM	0:00:01	🟩 Success

    f. Verify the acme-bank:1.1.0 component version has been deployed into the PROD Environment by selecting Applications

> acme-bank and expand the PROD environment to view the deployed component version.



Home > Applications > acme-bank

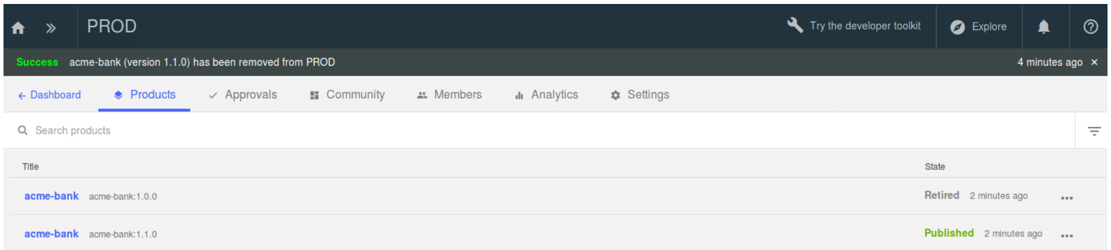
Application: acme-bank (show details)

Drag environments by their names to re-order them. *3 Environments*

Sandbox	Snapshot None	Compliancy 1 / 1
UAT	Snapshot None	Compliancy 1 / 1
PROD	Snapshot None	Compliancy 1 / 1

Component	Version	Date Deployed	Compliancy	Actions
acme-bank	acme-bank:1.1.0 (View Details)	3/9/2017, 1:47 PM	Compliant (1/1)	View Request

g. 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.



Success acme-bank (version 1.1.0) has been removed from PROD 4 minutes ago ×

Title		State	
acme-bank	acme-bank:1.0.0	Retired	2 minutes ago
acme-bank	acme-bank:1.1.0	Published	2 minutes ago

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

Lab 4.3: Bugfix release flow

A bugfix has been made and version 1.1.1 of the acme-bank API Product definition will need to be published. Upon a push to the GitHub repository with an annotated tag containing the keyword “**bugfix**”, the Jenkins build will be triggered.

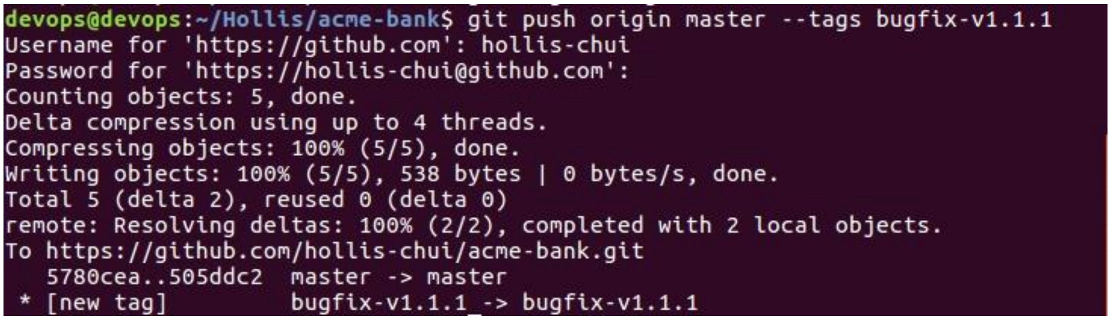
The push will trigger a Jenkins build using the same job configuration. The Jenkins build will create a UCD Component Version for acme-bank:1.1.1, upload the yaml files, add the “**release**” Version Status to the component version and call the same “Deploy API Product Def” process to replace the API Product definition into the UAT Catalog. As this is a bugfix, it will only need to be deployed to the upper environments such as UAT and PROD. Similar to minor version changes, 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 API Product definition.

1. Edit the acme-bank-product.yaml and acme-bank.yaml files in a text editor and modify the **Version** from **1.1.0** to **1.1.1**. Verify the change by running the command "**cat <yaml file> | grep version**".

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

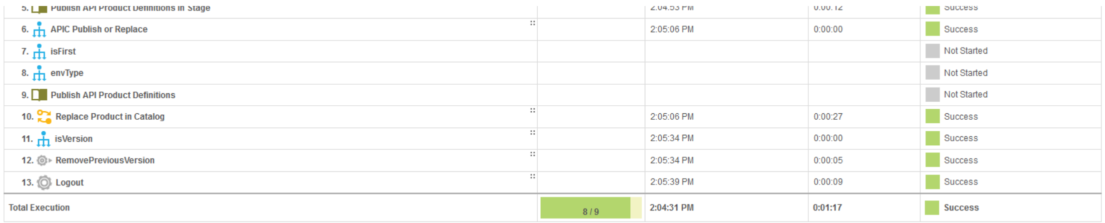
3. Create an annotated tag with the keyword "major" to associate with the push. The command is "**git tag -a bugfix-1.1.1**" -m "Version 1.1.1 release".

4. Push the changes with the tag into the Github repository by running the command "**git push origin master –tags <tag>**".



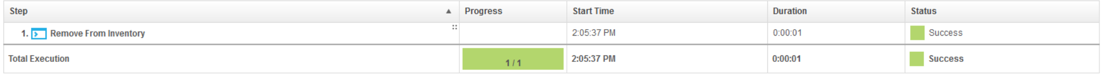
```
devops@devops:~/Hollis/acme-bank$ git push origin master --tags bugfix-v1.1.1
Username for 'https://github.com': hollis-chui
Password for 'https://hollis-chui@github.com':
Counting objects: 5, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 538 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.git
   5780cea..505ddc2  master -> master
 * [new tag]         bugfix-v1.1.1 -> bugfix-v1.1.1
```

5. Once the changes are pushed into the GitHub repository, this will trigger the Jenkins build to deploy the minor update to the Sandbox environment / catalog. Verify 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



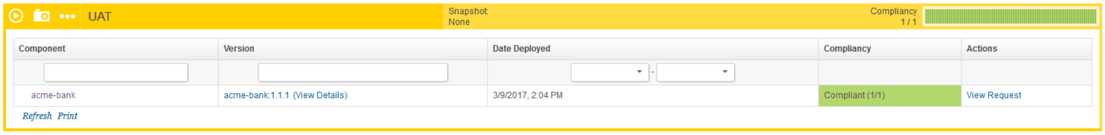
5. Publish API Product Definitions in Stage		2:04:53 PM	0:00:12	Success
6. APIC Publish or Replace		2:05:06 PM	0:00:00	Success
7. isFirst				Not Started
8. envType				Not Started
9. Publish API Product Definitions				Not Started
10. Replace Product in Catalog		2:05:06 PM	0:00:27	Success
11. isVersion		2:05:34 PM	0:00:00	Success
12. RemovePreviousVersion		2:05:34 PM	0:00:05	Success
13. Logout		2:05:39 PM	0:00:09	Success
Total Execution	8 / 9	2:04:31 PM	0:01:17	Success

e. 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.
   i. Process = RemoveFromInventory
   ii. Version = acme-bank:1.1.1
   iii. Environment = UAT



Step	Progress	Start Time	Duration	Status
1. Remove From Inventory		2:05:37 PM	0:00:01	Success
Total Execution	1 / 1	2:05:37 PM	0:00:01	Success

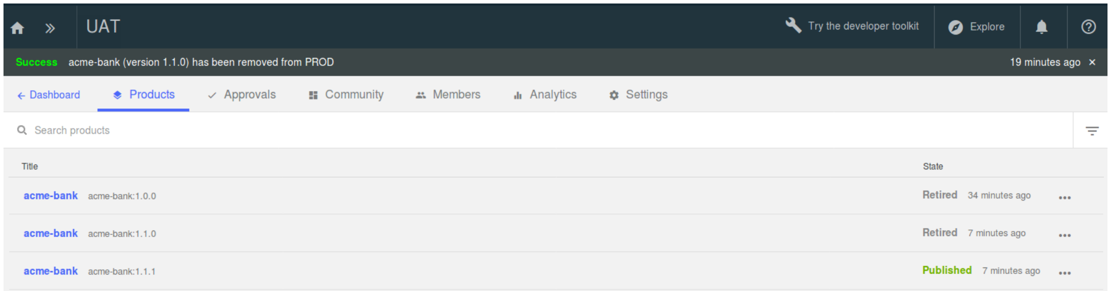
f. Verify the acme-bank:1.1.1 component version has been deployed into the UAT Environment by selecting Applications > acme-bank and expand the UAT environment to view the deployed component version.



UAT — Snapshot: None — Compliancy 1 / 1

Component	Version	Date Deployed	Compliancy	Actions
acme-bank	acme-bank:1.1.1 (View Details)	3/9/2017, 2:04 PM	Compliant (1/1)	View Request

g. 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.

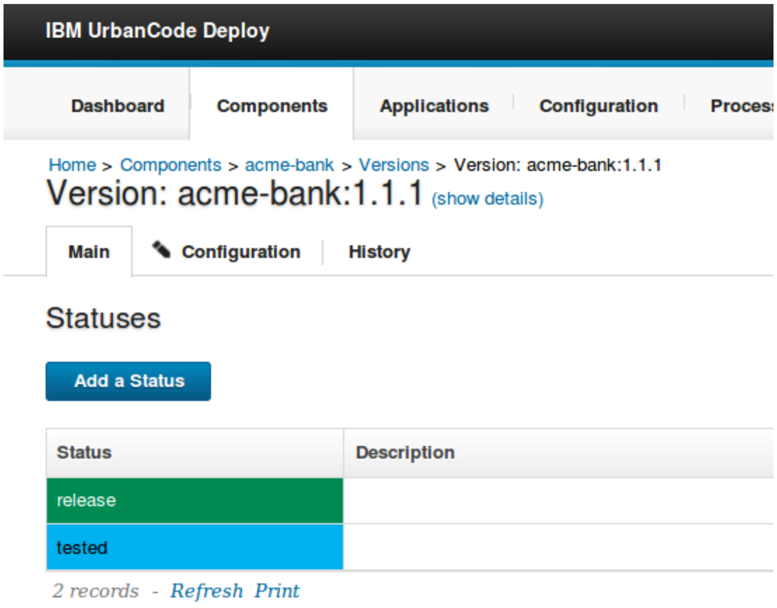


Success acme-bank (version 1.1.0) has been removed from PROD 19 minutes ago ×

Title		State	
acme-bank	acme-bank:1.0.0	Retired	34 minutes ago
acme-bank	acme-bank:1.1.0	Retired	7 minutes ago
acme-bank	acme-bank:1.1.1	Published	7 minutes ago

6. 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 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.
   b. Click on the Components tab and select the **acme-bank** component.

c. Click on the Versions tab and select the **acme-bank:1.1.1** component version.
d. 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:1.1.1

Version: acme-bank:1.1.1 (show details)

Statuses

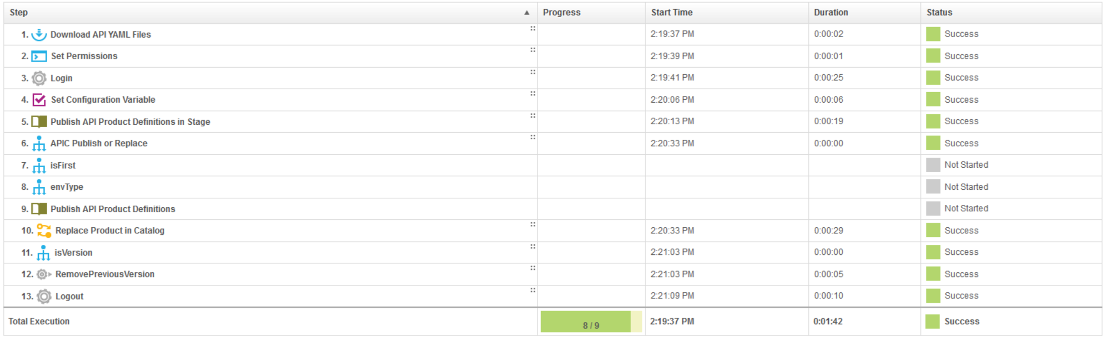
Status	Description
release	
tested	

2 records - Refresh Print

7. 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. Log in with admin / admin.
    b. Click on the Applications tab and select the **acme-bank** application.
    c. Click the play button for the PROD environment and the “Run Process on PROD” window should appear.
    d. For the Process field, select **acme-bank**.
    e. For the Versions, click on the *Choose Versions* link and a Component Versions window will pop up.
    f. Under the Versions to Deploy column, select the *Add…* link.
    g. Select acme-bank:1.1.1 and click OK and then Submit.

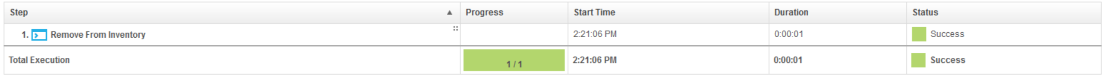
8. Verify the “**Deploy API Product Def**” component process completed successfully on the PROD environment.
    a. Open the Firefox browser and click on the IBM UrbanCode Deploy bookmark.
    b. Log in using admin / admin.
    c. In the UCD web client, click on the Components tab and select the **acme-bank** 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:1.1.1
   iii. Environment = PROD



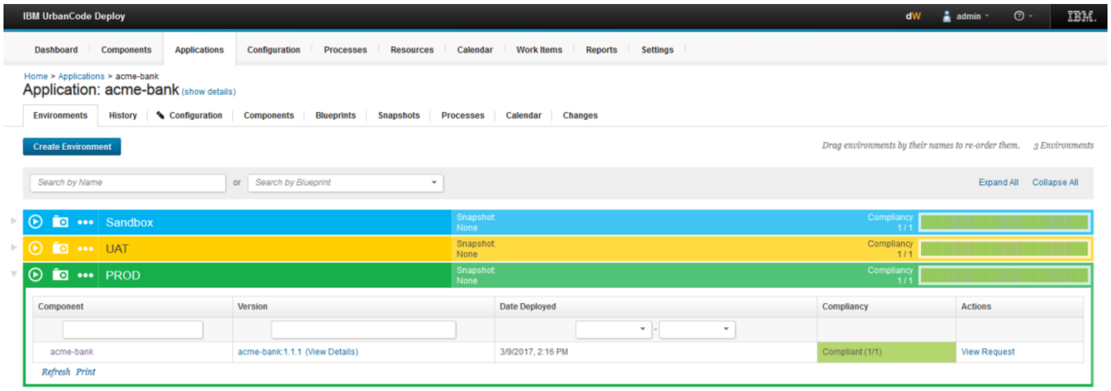
Step	Progress	Start Time	Duration	Status
1. Download API YAML Files		2:19:37 PM	0:00:02	🟩 Success
2. Set Permissions		2:19:39 PM	0:00:01	🟩 Success
3. Login		2:19:41 PM	0:00:25	🟩 Success
4. Set Configuration Variable		2:20:06 PM	0:00:06	🟩 Success
5. Publish API Product Definitions in Stage		2:20:13 PM	0:00:19	🟩 Success
6. APIC Publish or Replace		2:20:33 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		2:20:33 PM	0:00:29	🟩 Success
11. isVersion		2:21:03 PM	0:00:00	🟩 Success
12. RemovePreviousVersion		2:21:03 PM	0:00:05	🟩 Success
13. Logout		2:21:09 PM	0:00:10	🟩 Success
Total Execution	8 / 9	2:19:37 PM	0:01:42	🟩 Success

e. 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.
   i. Process = RemoveFromInventory
   ii. Version = acme-bank:1.1.1
   iii. Environment = PROD



Step	Progress	Start Time	Duration	Status
1. Remove From Inventory		2:21:06 PM	0:00:01	🟩 Success
Total Execution	1 / 1	2:21:06 PM	0:00:01	🟩 Success

f. Verify the acme-bank:1.1.1 component version has been deployed into the PROD Environment by selecting Applications > acme-bank and expand the PROD environment to view the deployed component version.

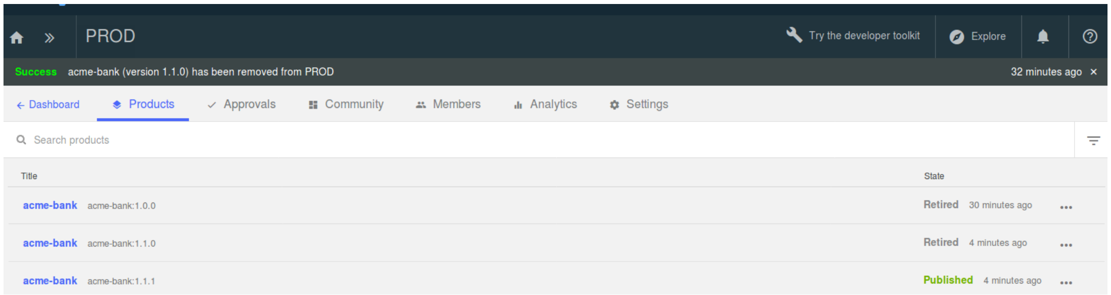


Application: acme-bank (show details)

Environment	Snapshot	Compliancy
Sandbox	None	1 / 1
UAT	None	1 / 1
PROD	None	1 / 1

Component	Version	Date Deployed	Compliancy	Actions
acme-bank	acme-bank:1.1.1 (View Details)	3/9/2017, 2:16 PM	Compliant (1/1)	View Request

g. 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.



Success acme-bank (version 1.1.0) has been removed from PROD 32 minutes ago

Title		State	
acme-bank	acme-bank:1.0.0	Retired	30 minutes ago
acme-bank	acme-bank:1.1.0	Retired	4 minutes ago
acme-bank	acme-bank:1.1.1	Published	4 minutes ago

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

You have completed all the lab exercise.