Splunk Alert Webhook to IBM uDeploy

Proof of Concept



Jeffrey Fall

Splunk support

Jeffreyfall.com

January 17 2019

# Abstract

This short paper describes the methodology used do set up and test a Splunk Alert calling an intermediate URL to actuate an actionable program state change in IBM uDeploy Application(s).

This paper does not call out or describe all the possibilities on extensive detail. Rather the reader is instructed on how to use a Splunk Alert to call an intermediate server which then formulates a suitable IBM uDeploy WSDL request. The end result is uDeploy performing a “request” on behalf of a Splunk Alert.

The “request” fulfilled by IBM uDeploy can be any Application sponsored by any Configuration relying on any Tagged Environment.

Table of Contents

[Abstract 2](#_Toc535500700)

[1 Introduction 4](#_Toc535500701)

[2 Splunk webhook POC test desired outcome 4](#_Toc535500702)

[3 POC Test Procedure 4](#_Toc535500703)

[4 Lab setup – Servers 5](#_Toc535500704)

[5 Testing Setup High Level Summary 5](#_Toc535500705)

[6 Script file linkages 7](#_Toc535500706)

[7 Verifying the test actually ran 11](#_Toc535500707)

[8 Real World Application of Splunk Webhook 13](#_Toc535500708)

[8.1 Intermediate Apache/ PHP server - Splunk Cloud on AWS currently 14](#_Toc535500709)

[8.2 Intermediate Server hosted on Windows / IIS for production 14](#_Toc535500710)

[9 Detailed Test Setup 14](#_Toc535500711)

[9.1 uDeploy Urban Code Deploy installation on linux 15](#_Toc535500712)

[9.2 Installing IBM Java Version 8 on Linux Mint or similar 15](#_Toc535500713)

[9.3 Set JAVA\_HOME for root account in Linux Mint 20](#_Toc535500714)

[9.4 Install IBM Urban Code Deploy (uDeploy) 21](#_Toc535500715)

# 1 Introduction

With the desire to move application deployments more toward automation and a CI/CD pipeline approach it was suggested to take advantage of the new Splunk Alert Webhook interface present in Splunk 7.2.3 and onwards.

uDeploy is an automation tool by IBM which is loosely akin to Jenkins but littered with IBM specific features and excellent documentation.

Splunk recently added Webhook ability as an alert mechanism in Splunk 7.2.3 and onwards.

Lately a goal has been sought to enable Splunk to control aspects of uDeploy through a Splunk Alert. This approach is fostered via a temporary POC test setup consisting of linux servers supporting instances of Splunk, uDeploy and apache2 with PHP installed. The target system under configuration of uDeploy is a Windows10 instance with an uDeploy agent installed.

# 2 Splunk webhook POC test desired outcome

* Splunk Alert with webhook is fired off
* uDeploy Application runs ONCE due to stimulation of Splunk Alert

# 3 POC Test Procedure

* A special string is created in a file in a directory which a specific Splunk Alert is monitoring. The test is run repeatedly if desired by copying the file to a new unique file name.
* uDeploy Application history on uDeploy is refreshed showing the desired application has run on behalf of the Splunk Alert

# 4 Lab setup – Servers

* udmint – a linux Mint server for uDeploy instance. Port 8443 is the standard access port for uDeploy
* splunkmint – a liinux Mint server hosting Splunk Version 7.2.3 or higher. Access Splunk as <http://splunkmint:8000>
* webhook a linux Mint server with Apache2 installed and PHP. This server acts to sink the webhook from Splunk and translate the call to a properly formatted POST call to uDeploy using curl with authentication for the uDeploy server and the proper JSON packet outlining the Application to be started.
* Windows10 – a Windows10 target server with an uDeploy agent installed on it. Typically a uDeploy Application needs at least one target to act on.

# 5 Testing Setup High Level Summary

1. Create a Windows10 target system as a VM.
2. Download the uDeploy 7.x.x. trial Urban Code Deploy tarball from IBM download
3. Download special IBM Java Version 8 bin file from IBM download site.
4. Install IBM java on uDeploy linux Mint server.
5. Add the line: declare –x JAVA\_HOME = /opt/ibm/java-x86\_64-80 To the end of the .bashrc file of the root account. Log out and log back in as root.
6. Install uDeploy on a Linux Mint VM from root account. This process is involved. Take most of the defaults. Use default derby database. Set uDeploy user name / password as admin and password admin
7. Log onto the uDeploy server. Example: <https://udmint:8443>
8. Install a udeploy agent on the Windows10 VM machine. Log into uDeploy from the Windows10 VM as <https://udmint:8443>. Use the life preserver on the upper right and then tools and you will see the UrbanCode Deploy Agent. Download the agent. Then download IBM Java Version 8 for windows. Install Java and then install the uDeploy agent in the Windows10 Box. Put the DNS name address of the udeploy box into the agent’s configuration. Verify the agent shows up as a windows service.
9. Create an Application in uDeploy with a full configuration and actionable target system. with agent installed. Follow the Hello World application lesson from IBM. Lesson 1: <https://www.ibm.com/support/knowledgecenter/SS4GSP_6.2.7/com.ibm.udeploy.tutorial.doc/topics/quickstart_abstract.html?view=kc> Lesson 2: <https://www.ibm.com/support/knowledgecenter/SS4GSP_6.2.7/com.ibm.udeploy.tutorial.doc/topics/webapp_abstract.html?view=kc> This might take hours or all day.
10. Create an alert in Splunk which monitors a directory on the splunk server for keywords which are inserted into a file
11. Create a linux Mint or similar server and name it webhook. Note – the name really does not matter but for examples – webhook is used. On this linux server install apache2 and PHP.
12. On the webhook apache / PHP server as root go to /var/www/html. Create a directory under /var/www/html to be your application directory WSDL. This directory will contain a PHP file of the form applicationName.php Inside this php file call a shell file .sh file which will send out a curl command to the uDeploy instance.
13. In a directory on the webhook server create a .sh (shell) file to run a curl command which will send the request directly to the uDeploy server. Make sure the execute permission on the file is set with chmod +x. The top line of the file should be #!/bin/sh
14. Enable the webhook in the alert to point to a URL on an intermediate linux webhook server with apache and PHP. The URL will correspond to the application on uDeploy to be run a Splunk alert
15. Activate the splunk alert by placing key words which trigger the alert into a file. To run test again just copy the file to a new unique name. Example: cp file1 file2
16. Log into udmint server – uDeploy server and click on the application. Find your application under test which in this case is “Hello World” and look under history. On the web browser keep refreshing. Look for the word RUNNING with an amber background and a green COMPLETED. Look at the completion times.

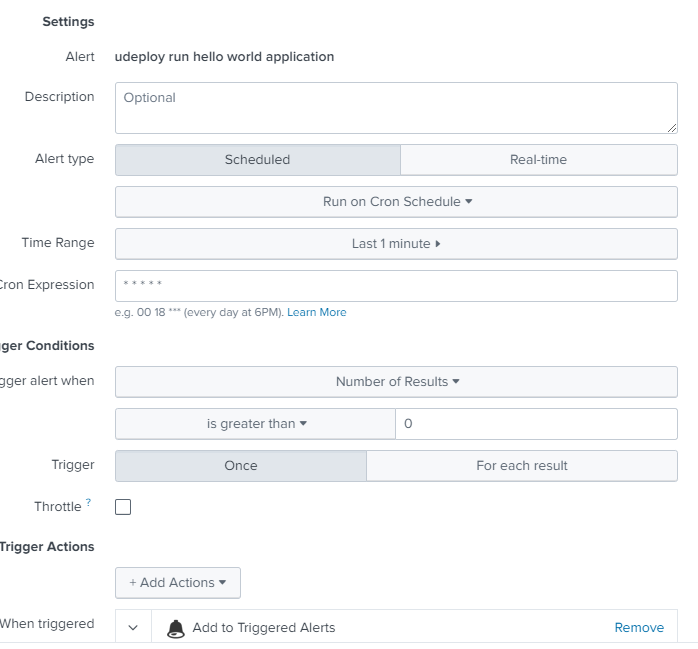
# 6 Script file linkages

Splunk webhook will call a URL on a server called webhook which calls a PHP URL. The PHP program behind the URL calls a shell program which then sends a curl request to the uDeploy server..

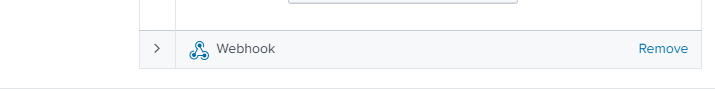
Splunk itself is not evolved enough yet to send a fully formatted request with a custom JSON payload to uDeploy. Therefore an intermediate server is needed.

Splunk will call a webhook on an intermediate server – in this lab case it is called webhook.

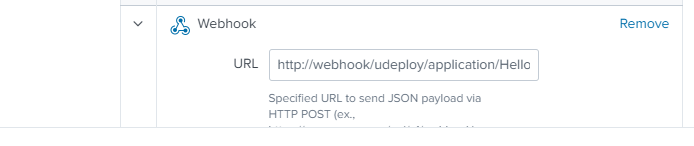
The alert:



The webhook in the alert:



The Webhook Detail:



The URL to the webhook server:

<http://webhook/udeploy/application/Hello_Application/start.php>

The start.php script:

Path to start.php:

/var/www/html/udeploy/application/Hello\_Application

**start.php**

-rw-r--r-- 1 root root 144 Jan 16 16:05 start.php

**start.php contents:**

<?php

$command = escapeshellcmd('/home/jfall/udeploy/Application/Hello\_World/callcurl.sh');

$output = shell\_exec($command);

echo $output;

?>

Start.php calls callcurl.sh

**File callcurl.sh**

PATH:

/home/jfall/udeploy/Application/Hello\_World/callcurl.sh

**Callcurl.sh contents:**

#!/bin/sh

echo "Calling curl for Hello World Application test<br><br>"

/usr/bin/curl -k -u admin:admin https://udmint:8443/cli/applicationProcessRequest/request -X PUT -d @/home/jfall/udeploy/Application/Hello\_World/applicationDeploy.json

echo "Call to curl completed<br><br>"

The lines in RED should be on one line in the file.

**File: applicationDeploy.json contents:**

{

"application": "Hello Application",

"applicationProcess": "hello App Process",

"environment": "helloDeploy",

"versions": [{

"version": "1.0",

"component": "helloWorld"

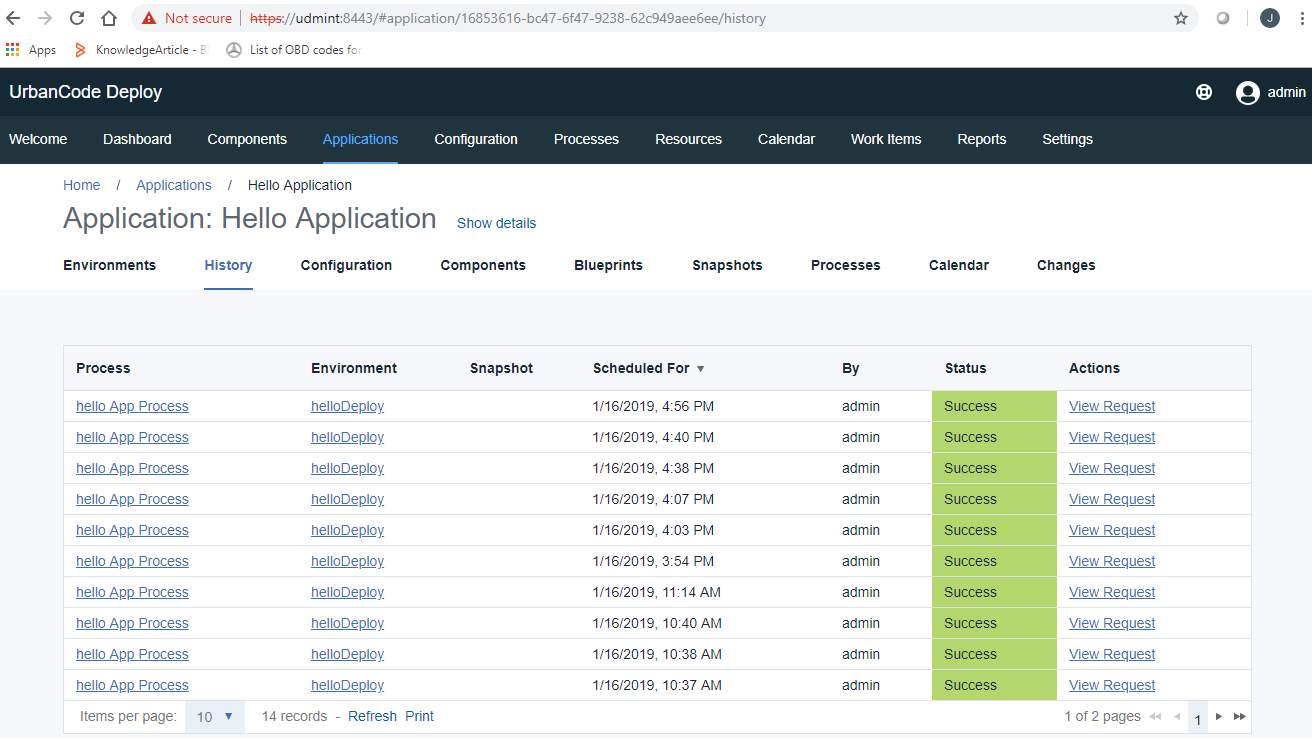
}]

}

# 7 Verifying the test actually ran

* Look at Apache log in webhook intermediate server.
* Look at the Application Log in uDeploy

Udeploy Application HISTORY:



The test is run by copying a file in a monitored splunk directory to a different name:

Here is the splunk monitored directory:

jfall@splunkmint:~/webhooktest$ ls

testfile testfile2 testfile3

Here is the testfile:

jfall@splunkmint:~/webhooktest$ more testfile

udeploy run hello world application

The alert will trigger when Splunk indexes the string: “udeploy run hello world application”

When you copy the file to a new file name Splunk finds a new file and indexes the contents of the file which is the same string which fires the alert.

The alert runs once a minute testing for the string “udeploy run hello world application”

This provides an almost instant run of the uDeploy Application.

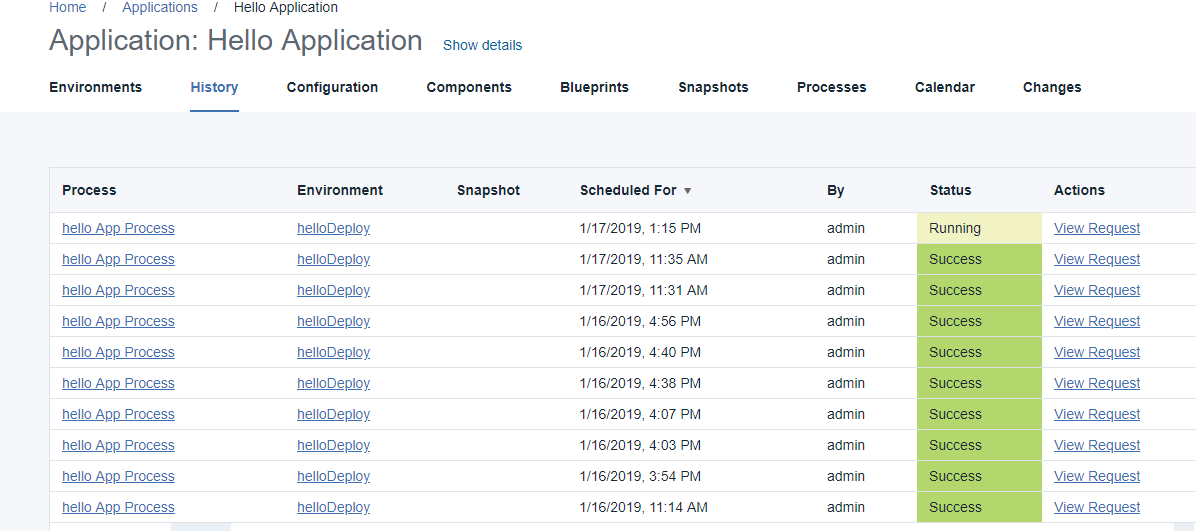
Look for the state change to RUNNING.

Begin the test by copying the file in the Splunk monitoring directory:

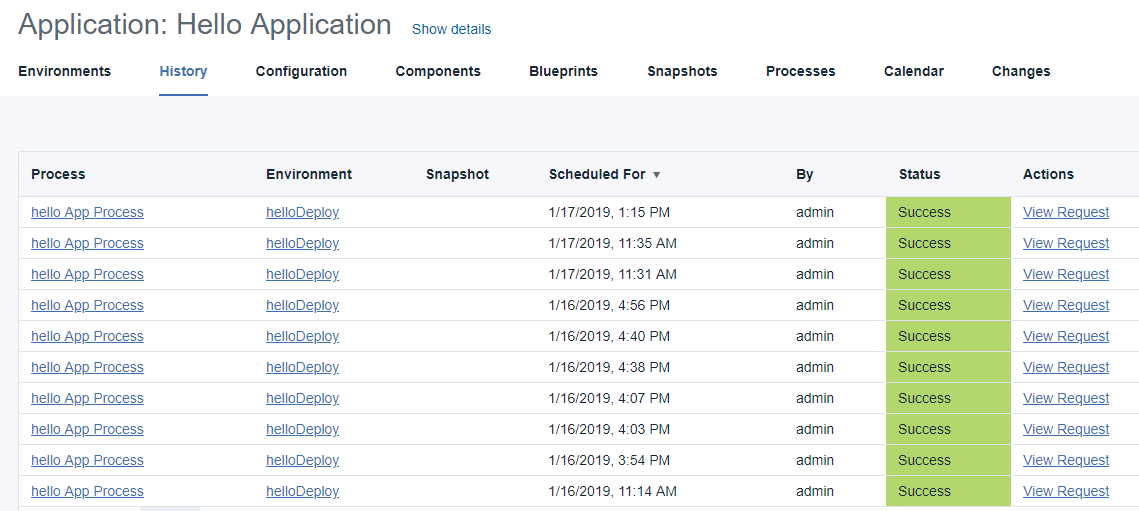


Now keep refreshing the Application History on the uDeploy server by holding down SHIFT and the refresh on the browser.

Soon RUNNING should be visible. This is a new Run of the Hello World application.



Finally the run is complete and the time of the run is updated on the top most ROW:



This proves the splunk alert fired the application on uDeploy.

# 8 Real World Application of Splunk Webhook

Currently the Splunk Webhook has its faults. To date the Alert Splunk Webhook can not send username / password verification. And the JSON package sent from Splunk is boilerplate.

So the connection in this POC and likely in the real world is unsecure. The only way to run it securely will be within the context of a VPN tunnel. And this still leaves the connection insecure within the organization. One might be able to send an SSL public key in as part of the SPLUNK results and have SSL decryption on the intermediate (webhook) server and then pass the request to uDeploy if the SSL keys agree.

Of course a firewall could block the port on the webhook intermediate server to only one IP address from a trusted server. This is trivial setup on AWS.

## 8.1 Intermediate Apache/ PHP server - Splunk Cloud on AWS currently

Splunk has admitted Splunkcloud.com is hosted on AWS. With this in mind an intermediate server to translate the Splunk Webhook to a properly formatted uDeploy POST / JSON request could also be hosted on AWS and could only respond to the proper IP. Of course the POC consisted of quick set up with Apache and PHP on Linux.

## 8.2 Intermediate Server hosted on Windows / IIS for production

Many companies has settled on Windows infrastructure and the intermediate webhook server could be hosted on Windows Server with IIS. PHP in this case might not be the best way forward with IIS

Encompassing Powershell. A named URL should be possible with a power shell component being callable from the Web and kicking off a Powershell send of the correct POST and JSON packet to the uDeploy application desired to run from the Splunk Alert. Of course AWS can host linux or Windows servers.

# 9 Detailed Test Setup

The most obtuse setup to the uninitiated would absolutely be the installation of uDeploy on a linux server. This process is very dependent on the correct Java instance being used for the setup and run of uDeploy. One MUST use the IBM sanctioned Java. Or uDeploy will fail on obscure Java stack dumps during installation.

Once uDeploy is setup one needs to log in and make an Application which one can run and test against.

Without vast knowledge of uDeploy it is sufficient to start with IBM’s lesson plan with the “Hello World” App and deploy it following the lesson plan. One will end up with the testing results as detailed in this paper if the instructions are followed carefully. The instructions seem to be written for a uDeploy version less then Version 7.0.0 so interpolation of the UI is necessary at times.

For uDeploy Administrators this paper will be easy to follow.

For the uDeploy uninitiated again here is the IBM link you can follow exactly to setup the Application to run with this Splunk webhook POC test: <https://www.ibm.com/support/knowledgecenter/SS4GSP_6.2.7/com.ibm.udeploy.tutorial.doc/topics/quickstart_abstract.html?view=kc>

## 9.1 uDeploy Urban Code Deploy installation on linux

Again the most critical part of this test is a uDeploy instance to test from. The first priority is installing a sanctioned IBM version of Java 8 onto Linux. Secondly is installing Urban Code Deploy or uDeploy on the linux server. Pretty much all of the defaults will work fine. Take note on HOW to start the uDeploy server. In Debian / Ubuntu / Mint linux links are added to /etc/init.d to start a link.

The link is a softlink to the start up script supplied by IBM in the installation. This allows uDeploy to start across linux reboots.

## 9.2 Installing IBM Java Version 8 on Linux Mint or similar

Obtain the IBM linux distribution file: ibm-java-x86\_64-sdk-8.0-5.26.bin

One can run the file in a GUI environment and install via GUI. Or run in command shell environment and install with command shell.

As root run the IBM java installer: ./ ibm-java-x86\_64-sdk-8.0-5.26.bin

Preparing to install...

Extracting the JRE from the installer archive...

Unpacking the JRE...

Extracting the installation resources from the installer archive...

Configuring the installer for this system's environment...

Launching installer...

Graphical installers are not supported by the VM. The console mode will be used instead...

===============================================================================

Choose Locale...

----------------

1- Bahasa Indonesia

2- Català

3- Deutsch

->4- English

5- Español

6- Français

7- Italiano

8- Português

CHOOSE LOCALE BY NUMBER: 4

===============================================================================

IBM® 64-bit SDK for Linux®, v8.0 (created with InstallAnywhere)

-------------------------------------------------------------------------------

Preparing CONSOLE Mode Installation...

===============================================================================

International License Agreement for Non-Warranted Programs

Part 1 - General Terms

BY DOWNLOADING, INSTALLING, COPYING, ACCESSING, CLICKING ON AN

"ACCEPT" BUTTON, OR OTHERWISE USING THE PROGRAM, LICENSEE AGREES TO

THE TERMS OF THIS AGREEMENT. IF YOU ARE ACCEPTING THESE TERMS ON

BEHALF OF LICENSEE, YOU REPRESENT AND WARRANT THAT YOU HAVE FULL

AUTHORITY TO BIND LICENSEE TO THESE TERMS. IF YOU DO NOT AG

Press Enter to continue viewing the license agreement, or enter "1" to

accept the agreement, "2" to decline it, "3" to print it, or "99" to go back

to the previous screen.: 1

===============================================================================

Introduction

------------

InstallAnywhere will guide you through the installation of IBM® 64-bit SDK for

Linux®, v8.0.

It is strongly recommended that you quit all programs before continuing with

this installation.

Respond to each prompt to proceed to the next step in the installation. If you

want to change something on a previous step, type 'back'.

You may cancel this installation at any time by typing 'quit'.

PRESS <ENTER> TO CONTINUE:

===============================================================================

Choose Install Folder

---------------------

Where would you like to install?

Default Install Folder: /opt/ibm/java-x86\_64-80

ENTER AN ABSOLUTE PATH, OR PRESS <ENTER> TO ACCEPT THE DEFAULT

:

===============================================================================

Pre-Installation Summary

------------------------

Please Review the Following Before Continuing:

Product Name:

IBM® 64-bit SDK for Linux®, v8.0

Install Folder:

/opt/ibm/java-x86\_64-80

Disk Space Information (for Installation Target):

Required: 252,728,257 Bytes

Available: 7,114,473,472 Bytes

PRESS <ENTER> TO CONTINUE:

===============================================================================

Installing...

-------------

[==================|==================|==================|==================]

[----

----------------|------------------|------------------|------------------]

===============================================================================

Installation Complete

---------------------

Congratulations. IBM® 64-bit SDK for Linux®, v8.0 has been successfully

installed to:

/opt/ibm/java-x86\_64-80

PRESS <ENTER> TO EXIT THE INSTALLER:

Congratulations. You have installed IBM Java !!!

## 9.3 Set JAVA\_HOME for root account in Linux Mint

* cd /root
* vi .bashrc
* Add this to the end of the .bashrc file: declare –x JAVA\_HOME=” /opt/ibm/java-x86\_64-80”
* Log out of root and log back in.

## 9.4 Install IBM Urban Code Deploy (uDeploy)

* Download this trial file UBCD\_7.0.0\_TRIAL.zip
* Make any directory – make one called UCDEPLOY
* mv UBCD\_7.0.0\_TRIAL.zip UCDEPLOY
* cd UCDEPLOY
* unzip UBCD\_7.0.0\_TRIAL.zip
* Make sure JAVA\_HOME IS set. Try echo $JAVA\_HOME should resolve to: /opt/ibm/java-x86\_64-80
* If JAVA\_HOME is not set: declare –x JAVA\_HOME=” /opt/ibm/java-x86\_64-80”
* cd ibm-ucd-install
* Locate the file named: install-server.sh
* Start the installer: ./install-server.sh

Enter the directory of the server to upgrade(leave blank for installing to a clean directory).

(leave this blank) hit enter

Enter the home directory for the JRE/JDK that the new server or already installed server uses. Default [/opt/ibm/java-x86\_64-80]:

(the installer picked up your $JAVA\_HOME) hit enter

Buildfile: install.with.groovy.xml

[mkdir] Created dir: /home/jfall/UCDEPLOY/ibm-ucd-install/compiled

version-check:

compile:

[groovyc] Compiling 4 source files to /home/jfall/UCDEPLOY/ibm-ucd-install/compiled

[copy] Copying 20 files to /home/jfall/UCDEPLOY/ibm-ucd-install/compiled

install:

[unzip] Expanding: /home/jfall/UCDEPLOY/ibm-ucd-install/conf.zip into /tmp/install-2354954912834746300.tmp

[echo] Found sub-installer UCDeployInstaller.groovy

LICENSE INFORMATION

The Programs listed below are licensed under the following License Information

terms and conditions in addition to the Program license terms previously

agreed to by Client and IBM. If Client does not have previously agreed to

license terms in effect for the Program, the International License Agreement

for Evaluation of Programs (Z125-5543-05) applies.

Keep hitting enter to get thru licensing stuff

[echo] Do you accept the license? [y,n] y

[echo] Installing IBM UrbanCode Deploy version 7.0.0.0.984935

[echo] Enter the directory where the IBM UrbanCode Deploy should be installed. [Default: /opt/ibm-ucd/server]

Take default directory

Enter

[echo] The specified directory does not exist. Do you want to create it? Y,n [Default: Y] y

[echo] Installing IBM UrbanCode Deploy to: /opt/ibm-ucd/server

[echo] JVM Version detected: 1.8.0\_191

[echo] JAVA\_HOME: /opt/ibm/java-x86\_64-80

[echo] Will this server be used as a node in a high availability cluster? y,N [Default: N]

Enter for default

[echo] Where should the server store application data such as logs, plugins, and keystores? [Default: /opt/ibm-ucd/server/appdata]

Enter for default

[echo] The specified directory for application data (/opt/ibm-ucd/server/appdata) doesn't exist. Do you want to create it? Y,n [Default Y]

Enter for default

[echo] What host name will users access the Web UI at? [Default: webhook]

Take the default. This server hostname in this case is webhook and it is webhook in DNS.

[echo] Do you want the Web UI to always use secure connections using SSL? Y,n [Default: Y]

Yes – default Enter

[echo] Enter the port on which the Web UI should listen for secure HTTPS requests. [Default: 8443]

Take default 8443. Remember this port number.

[echo] Enter the port on which the Web UI should redirect unsecured HTTP requests from. [Default: 8080]

Enter – default again

[echo] Enter the initial password for the admin user

Use admin for a password

[echo] Passwords do not match. Enter the initial password for the admin user.

[echo] Please type password again.

[echo] Enter the port to use for JMS agent communication. [Default: 7918]

Enter for default

[echo] Do you want the Server and Agent communication to require mutual authentication? This requires a manual key exchange between the server and each agent. See the documentation for more details. y,N [Default: N]

Enter for default

[echo] UCD's web-based replacement for JMS agent communication requires a URL that is used to set a listening port and to allow servers to exchange agent messages.

[echo] The URL has the following requirements:

[echo] 1) The URL must have the form "wss://<host-or-ip>:<port>". Example: wss://ucd1.example.org:7919

[echo] 2) The URL port must not conflict with ports for the UI or JMS.

[echo] 3) Each server must have a distinct URL.

[echo] 4) Every server in an HA cluster must be able to connect directly to this server with the URL.

[echo] 5) Agents and relays must be able to connect to one or more servers at their respective URL port.

[echo] Agents and relays may be configured to use a common URL shared by all servers (for example, one provided by a load balancer), but servers must be configured with direct URLs.

[echo] Enter the web agent communication URL for this server. [Default: wss://webhook:7919]

Enter for default

[echo] Enter the port and hostname of a Rational License Key Server containing product licenses for UrbanCode Deploy, in the form of port@hostname. (e.g. 27000@licenses.example.com) Alternatively, you may leave this blank to begin a 60-day evaluation period. [Default: none]

Default none

Use enter for default

[echo] When installing a server as a part of an existing cluster or when using a pre-populated database, it is not necessary to create the database schema. This step must be performed when installing a standalone server to a fresh database or when installing the first server in a cluster.

[echo] Create database schema? (For high availability servers, this should only be done for the first server in the cluster) Y,n [Default: Y]

Enter N

Not part of a cluster.

[echo] Please enter the database connection information for this server to use when connecting to the existing database:

[echo] The following database types are supported: derby, mysql, oracle, sqlserver, db2, db2zos.

[echo] Enter the database type to use. [Default: derby]

Enter for default

[echo] Enter the database username. [Default: ucd]

Enter for default

[echo] Enter the database password. [Default: password]

Enter for default

[copy] Copying 2 files to /home/jfall/UCDEPLOY/ibm-ucd-install/bin/server

[mkdir] Created dir: /opt/ibm-ucd/server/bin

[mkdir] Created dir: /opt/ibm-ucd/server/conf/server

[mkdir] Created dir: /opt/ibm-ucd/server/endorsed

[mkdir] Created dir: /opt/ibm-ucd/server/lib

[mkdir] Created dir: /opt/ibm-ucd/server/licenses

[mkdir] Created dir: /opt/ibm-ucd/server/native

[mkdir] Created dir: /opt/ibm-ucd/server/extensions

[mkdir] Created dir: /opt/ibm-ucd/server/var/log

[mkdir] Created dir: /opt/ibm-ucd/server/var/temp

[mkdir] Created dir: /opt/ibm-ucd/server/appdata/patches

[mkdir] Created dir: /opt/ibm-ucd/server/appdata/var

[mkdir] Created dir: /opt/ibm-ucd/server/appdata/conf/server

[copy] Copying 1 file to /opt/ibm-ucd/server/conf

[sync] Copying 209 files to /opt/ibm-ucd/server/lib

[sync] Copying 9165 files to /opt/ibm-ucd/server/opt/tomcat

[copy] Copying 1 file to /opt/ibm-ucd/server/bin

[copy] Copying 1 file to /opt/ibm-ucd/server/opt/tomcat/conf

[mkdir] Created dir: /opt/ibm-ucd/server/bin/init

[copy] Copying 2 files to /opt/ibm-ucd/server/bin

[copy] Copying 1 file to /opt/ibm-ucd/server/bin

[copy] Copying 1 file to /opt/ibm-ucd/server/bin

[unzip] Expanding: /home/jfall/UCDEPLOY/ibm-ucd-install/udconf.zip into /tmp/install-964821376385008321.tmp

[copy] Copying 9 files to /opt/ibm-ucd/server/conf

[copy] Copied 1 empty directory to 1 empty directory under /opt/ibm-ucd/server/conf

[copy] Copying 12 files to /opt/ibm-ucd/server/appdata/conf

[copy] Copying 272 files to /opt/ibm-ucd/server/database

[copy] Copying 1 file to /opt/ibm-ucd/server/database

[copy] Copying 13 files to /opt/ibm-ucd/server

[copy] Copying 28 files to /opt/ibm-ucd/server/appdata

[copy] Copying 4 files to /opt/ibm-ucd/server/bin

[propertyfile] Updating property file: /opt/ibm-ucd/server/conf/server/installed.properties

[copy] Copying 1 file to /opt/ibm-ucd/server/appdata/conf/server

[echo] OS: linux

[echo] Architecture: x64

[delete] Deleting directory /tmp/install-964821376385008321.tmp

[echo] Creating new AES encryption key.

[propertyfile] Updating property file: /opt/ibm-ucd/server/conf/server/installed.properties

[echo] Starting embedded database ...

[echo] waiting for db to start - 60 seconds remaining

[echo] Database Started

[echo] Stopping embedded database ...

[java] Thu Jan 17 14:59:49 EST 2019 : Apache Derby Network Server - 10.8.3.1 - (1476465) shutdown

[propertyfile] Updating property file: /opt/ibm-ucd/server/conf/server/installed.properties

[propertyfile] Updating property file: /opt/ibm-ucd/server/conf/server/secured-installed.properties

[propertyfile] Updating property file: /opt/ibm-ucd/server/conf/server/installed.properties

[propertyfile] Updating property file: /opt/ibm-ucd/server/conf/server/secured-installed.properties

[copy] Copying 1 file to /opt/ibm-ucd/server

[propertyfile] Updating property file: /opt/ibm-ucd/server/conf/server/installed.properties

[propertyfile] Creating new property file: /opt/ibm-ucd/server/conf/installed.version

[propertyfile] Creating new property file: /opt/ibm-ucd/server/appdata/conf/installed.version

[echo] After starting the server, you may access the web UI by pointing your web-browser at

[echo] https://webhook:8443 to complete the Installation.

[echo] Installer Complete. (press return to exit installer)

[delete] Deleting directory /tmp/install-7585716124094101340.tmp

[copy] Copying 1 file to /opt/ibm-ucd/server/var/install-log

BUILD SUCCESSFUL

Total time: 3 minutes 7 seconds

root@udmint:/home/jfall/UCDEPLOY/ibm-ucd-install#

now your installation of Urban Code Deploy TRIAL is complete.

You can start the server and log in with

<https://udmint:8443>

admin is the username

admin is the password