User Manual – Deployment Automation

Author: Veera Veluchamy

Version: 1.0

Date: 07 Jan 2016

Contents

1. Introduction

This User Manual contains a description step-by-step procedure for a user to perform Automatic deployment of EduDym and Knov8 projects.

2. Auto Deployment

Currently, the development team deploys artefacts manually in the Alpha, Sandbox and Production environments. Manual deployment requires more time, human intervention and is prone to errors.

To reduce the deployment time, human intervention and risk, the deployment process is now automated using **Ansible**.

3. Introduction to Ansible

Ansible is an open source configuration management tool. **Ansible** supports software deployment, ad hoc task execution, configuration management, database migration and utilities restart on remote servers/ nodes. It manages nodes over SSH (Secure Shell).

We can also execute individual scripts as required. Installation and configuration of Ansible is easy when compared to other configuration management tools. It is an agentless configuration management tool.

Ansible supports CLI (Command Line Interface) only. It will run the tasks and scripts based on **Playbook** configuration. Playbooks are written in YAML format which is easily readable and understandable.

For automatic deployment to work, Ansible must be installed in **/home/poc/** location of the server to be deployed on. It requires Python (2.4 or later) to be installed in host.

Ansible Playbook

Playbooks are the basis for configuration management and multi-machine deployment system. Playbooks are expressed in YAML format and have minimum syntax constraints, it intentionally tries to not be a programming language or script, but rather a model of a configuration or a process.

Tasks in a playbook can be executed in local machine and remote machines. We can categorize our tasks with required name and commands. Playbooks are located in **/home/poc/ansible** directory with '.yml' extension.

NOTE: The Playbook location "/home/poc/ansible" depends on the ansible installation directory. (We have installed 'Ansible' in /home/poc. It will be changed when we do new installation to some other directory)

Sample Playbook

```
---
- hosts: webservers
  vars:
    http_port: 80
    max_clients: 200
  remote_user: root
  tasks:
  - name: ensure apache is at the latest version
    yum: name=httpd state=latest
  - name: write the apache config file
    template: src=/srv/httpd.j2 dest=/etc/httpd.conf
    notify:
    - restart apache
  - name: ensure apache is running (and enable it at boot)
    service: name=httpd state=started enabled=yes
```

In this sample Playbook, 'name' field denotes the task that is done on the 'webservers' (hosts). Here 'webservers' refer to a host group name. The "Group of host" detail will be available in the **inventory** file.

4. Steps for Auto Deployment

I. Ansible and Flyway Installation Verification

II. Update Inventory Files

III. Update config File

IV. DB Migration

V. Copy Artefacts from Jenkins Server

VI. Execute Ansible Playbook

I. Ansible and Flyway Installation Verification

Ensure that **Ansible and Flyway** are properly installed by checking installation paths.

Generally the installation folders are /home/poc/ansible for Ansible and /home/poc/flyway-3.2.1/ for Flyway.

II. Update Inventory File

We can deploy, update configuration, database migration in multiple hosts at a time using Ansible. We could also categorize/segregate our hosts by creating host groups. A group is used to execute particular

task in a group of hosts by just specifying the group name. The group of hosts detail will be available in the **inventory** files.

Inventory file is located in **/etc/ansible/hosts**. An Ansible playbook has group of hosts which are delimited by [header] elements. You can enter hostnames or IP addresses. A hostname/IP can be member of multiple groups.

To update an inventory file, locate **/etc/ansible/hosts** and properly configure **'webservers', 'dbservers' and 'nagios-servers'.** Comment unwanted web servers entries from the host file. (To comment put # in front of the web server IP/Hostname)

a. Sample Inventory file

```
[webservers]
10.235.0.253 nagios_id=knova-alp-web02
#10.235.0.252 nagios_id=knova-alp-web01
# If you have multiple hosts following a pattern you can specify
# them like this:


# Ex 2: A collection of database servers in the 'dbservers' group

[dbservers]
10.236.0.2 nagios_id=knova-alp-web02

# Here's another example
[nagios-server]
209.235.220.254
```

III. Update config File

Locate the configuration file at **/home/poc/ansible/config.yml. C**onfigure load balancer, release type, version number and revision (for copying artifacts from the Jenkins server) and the URLs list to download JARs from third party websites.

```
---
Load_Balancer_Config:
    10.235.0.253:
      - KNOVA-ALP-http
      - KNOVA-ALP-https

Download_Jar_Config:
    - DirectoryPath: /home/poc/test
      URL_list:
        - https://json-simple.googlecode.com/files/json-simple-1.1.1.jar
        - https://org-json-java.googlecode.com/files/org.json-20120521.jar
    - DirectoryPath: /home/poc/
      URL_list:
        - https://json-simple.googlecode.com/files/json-simple-1.1.1.jar
        - https://org-json-java.googlecode.com/files/org.json-20120521.jar
Rel_Version: 65
Rel_Type: RB
Rel_Revision: 0.1
```

The First block starting with **'Load_Balancer_Config'** are to enable/disable webserver from INETU load balancer. User need to specify the IP address of the load balancer as header and list of load balancer names to enable/disable.

The second block starting with '**Download_Jar_Config**' is to download JAR files from third party websites. User need to specify the destination directory to download next to the '**DirectoryPath'** and list of URLs to download under '**URL_list'**. User can configure multiple URLs with different destination directory path to download.

IV. Custom Scripts

We have written few custom scripts for the following tasks in deployment, they will be included and executed by Ansible Playbook. Scripts are located in **/home/poc/ansible/files/scripts** directory.

Script Name	Description	Usage
delete_file.pl	Delete unwanted files from deployment directory	perl delete_file.pl < yml configuration file with full path >
disable_enable_load_balancer.pl	Disable/Enable webserver from INETU load balancer	perl disable_enable_load_balancer.pl < yml configuration file with full path > <state>
download_jar.pl	Download third party JAR files in given destination path	perl download_jar.pl < yml configuration file with full path >

disable/enable.sh	Disable / Enable New Relic Application Policy	sh disable.sh , sh enable.sh

These scripts will accept run-time arguments, and the arguments will be read from config.yml file. User need to change the argument values based on the deployment needs.

V. DB Migration

Flyway is an open source database migration tool for running database migration remotely. **Flyway will create a metadata table automatically if it doesn't exist and it won't affect the existing database. It will run the migration on top of the database.**

It will scan the filesystem or your class path for available migrations. It will compare them to the migrations that have been applied to the database. If any difference is found, it then migrates the database to close the gap.

Migration should preferably be executed on application startup to avoid any incompatibilities between the database and the expectations of the code.

Ansible will take a backup of the previous database state before running database migrations.

Example 1: We have migrations available up to version 9, and the database is at version 5.

Migrate will apply the migrations 6, 7, 8 and 9 in order.

Example 2: We have migrations available up to version 9, and the database is at version 9.

Migrate does nothing.

a. Naming convention for database migration

- SQL files should start with prefix RB_
- File separator must be __ (**Double underscore**)
- Save the SQL files under /**home/poc/flyway-3.2.1/sql/1.0**/ directory (**This is a configurable one in flyway.conf file located in /home/poc/flyway-3.2.1/**)
- E.g.: The file name should be **'RB_6_0__demo.sql'**
  Where,
    - RB_ is the Prefix Name
    - 6_0 is the file description (You can use your own description)
    - __ is the file name separator
    - demo is the file description

VI. Copy Artifacts from Jenkins Server

To copy the WARs and JARs from Jenkins you need to provide three variables as input parameter to the common input file (**/home/poc/ansible/config.yml**). Input parameters are Rel_Version, Rel_Type and Rel_Revision.

For example, if you are going to deploy RB_65_0.1 then you need to provide Rel_Type as RB and Rel_Version as 65 and Rel_Revision as 0.1

If you are going to deploy the trunk then you need to provide Rel_Type value as Trunk and Rel_Version and Rel_Revision values as empty (blank).

VII. Execute Ansible Playbook

Navigate to the directory where Ansible is installed and run the playbook by **ansible-playbook** command.

E.g.: ansible-playbook dev_deploy.yml

- Where, dev_deploy.yml is name of the playbook