****

**Auto Installation Setup for Live Spread**

**Ver. 1.1**

**Document Version Control**

|  |  |
| --- | --- |
| **Document Name** | Auto installation Setup for Live Spread |
| **Document Status** | Completed |
| **Version Number** | 1.0 |
| **Date** | 24th July 2018 |
| **Author** | Jyotirmoy Sarmah |
| **Authorized by** | Molakalmuru, Tanveer Ahmed |
| **Distribution** |  |

**Change History**

| **Version** | **Issue Date** | **Author** | **Reason for Change** |
| --- | --- | --- | --- |
| 1.0 | 25th July 2018 | Deepak Pai | Added Artefact Details |
| 1.1 | 2nd Jan 2019 | Jyotirmoy Sarmah | * Added contents to accommodate new sh scripts. * Modified scripts to simplify general.properties content which took most of the effort |
|  |  |  |  |

Table of Contents

[Objective 4](#_Toc534743911)

[Scope of Automation 4](#_Toc534743912)

[Out of Scope 4](#_Toc534743913)

[Prerequisite to run ansible automation script 4](#_Toc534743914)

[Install Ansible on Jump Server 5](#_Toc534743915)

[Design Consideration 5](#_Toc534743916)

[Ansible Architecture 5](#_Toc534743917)

[Configuration simplified: 6](#_Toc534743918)

[All configurations are redesigned to facilitate Automation 6](#_Toc534743919)

[Consolidated RAGE configuration files 6](#_Toc534743920)

[Artefacts for Live Spread Automation setup: 6](#_Toc534743921)

[DB Scripts: 6](#_Toc534743922)

[Artefacts for Live Spread: 7](#_Toc534743923)

[Deployment plan of key components to be automated 7](#_Toc534743924)

[RAGE Application (LS Application): 7](#_Toc534743925)

[RAGE Process: 7](#_Toc534743926)

[Extraction Scripts: 8](#_Toc534743927)

[Clone Auto installation script 8](#_Toc534743928)

[Steps for Live Spread Automation: 8](#_Toc534743929)

[Save all artefacts on standard format and name on FTP server. 8](#_Toc534743930)

[Global Variables 12](#_Toc534743931)

[Inventory File 13](#_Toc534743932)

[Main File (site.yml) 13](#_Toc534743933)

[How to run automation script 14](#_Toc534743934)

[Ways and Options of running the Automation script 15](#_Toc534743935)

[Common Issues after running 15](#_Toc534743936)

# Objective

Objective of making automation is to automate and standardize the process of deployment with less manual intervention and save time.

This document will act as a guide for users to do auto installation on different environment.

# Scope of Automation

1. RAGE Application deployment and configuration
2. RAGE process deployment and configuration
3. Import of RAGE process xml’s
4. Deployment and configuration of extraction scripts
5. LS web application deployment and configuration

# Out of Scope

1. Provision of Network and other infrastructure components
2. Provision of Web server and application server
3. Provision and configuration of OCR server
4. Provision of Database server
5. Installation of database software
6. Database schema set up
7. Execution of database scripts
8. File server mount

# Prerequisite to run ansible automation script

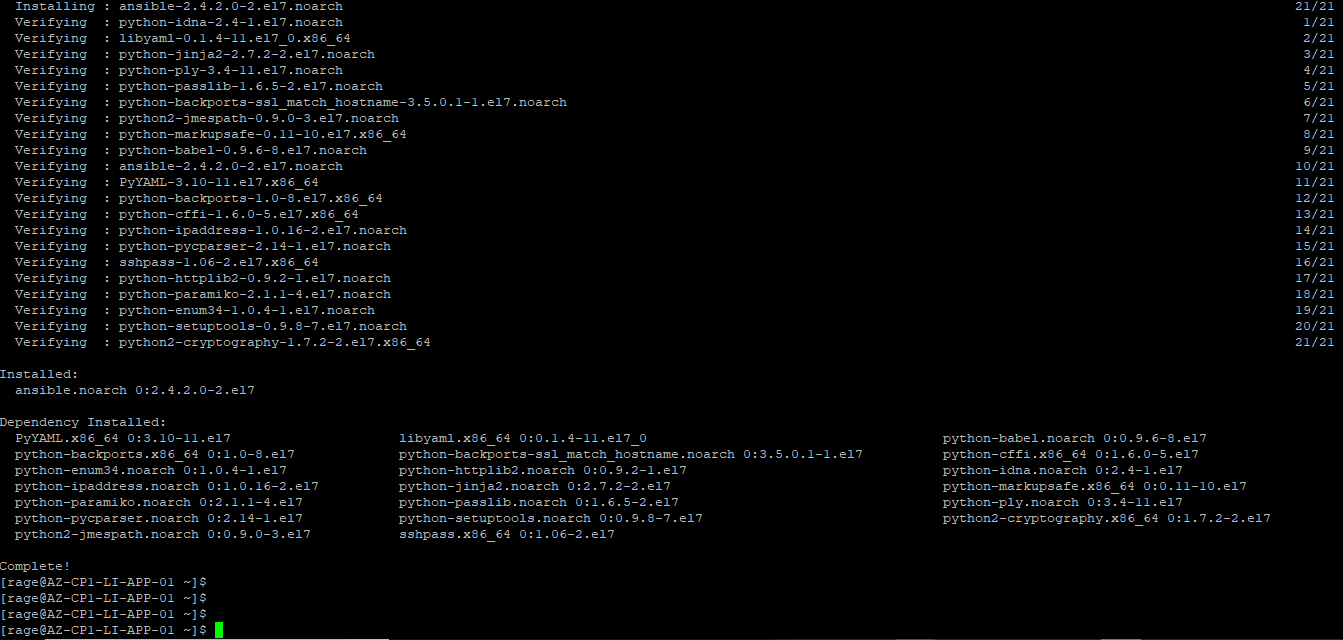
* Linux Jump Server and minimum one processing server and one application server.
* Configure root path as /opt/rage or /home/rage with rage as user and with sudo access.
* All required artefacts are prepared and shared on an ftp server which needs to be accessible from jump server through scp command.
* Internet access needs to be available temporarily or ansible needs to be pre-installed on Jump Server.
* ssh connection need to work within each Linux machine.
* DB connection needs to be there with both VM’s
* Firewall issues should not be there to access VM’s.
* Relevant ports needs to be opened.
* VM’s needs to have access to ftp server
* Folder structure and standards of LS artefacts maintained in FTP server needs to be same apart from the version folder which will keep on changing.
* Access to ftp server from cloud vm’s

# Install Ansible on Jump Server

Use below command to install ansible on your Linux Jump Server

**sudo yum install ansible**

Screen shot after installing:

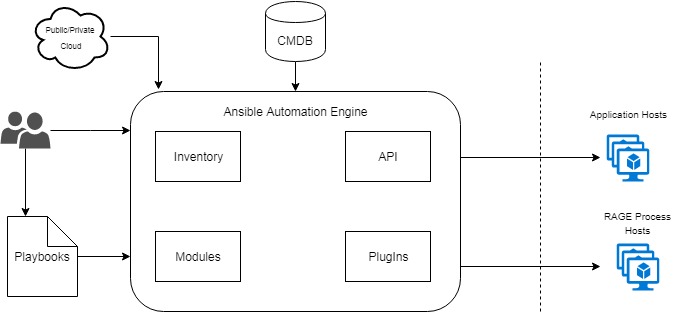


# Design Consideration

Modularity, plug and play, reusability and easy configuration. Ansible role are used to achieve modularity and reuse, Handlers are used to trigger events like restart of services like tomcat etc.

Templates, var’s, var files are used achieve easy configuration external dynamic inventory like Cobbler or CMDB can be integrated, playbooks are used for orchestration. Ansible playbooks are source controlled

# Ansible Architecture



# Configuration simplified:

# All configurations are redesigned to facilitate Automation

Moved configurations out of Live Spread war file. Consolidated application configuration files into single file i.e. input.properties.

Removed dependency on RD client for import/export of RAGE processes. Import and Export is now supported via RAGE jar execution.

# Consolidated RAGE configuration files

Consolidated extraction scripts configuration files

# Artefacts for Live Spread Automation setup:

To start with automation as a prerequisite we need to have a list of artefacts ready on our ftp server on a specified location. Below were the list of standard artefacts that need to be on our ftp server for Live spread automation to start with.

## DB Scripts:

1. LIVESPREAD\_SEQUENCES.txt

2. LIVESPREAD\_SP\_FUNCTION.txt

3. LIVESPREAD\_TABLESTRUCTREONLY.txt

4. LIVESPREAD\_TABLESTRUCTREWITHDATA.txt

5. APPLICATION\_DB\_SEQUENCES.txt

6. PLATFORM\_SP\_VIEW\_FUNCTION.txt

7. PLATFORM\_TABLESTRUCTREONLY.txt

8. PLATFORM\_TABLESTRUCTREWITHDATA.txt

9. UIC.txt

10. UIC\_SP.txt

### Artefacts for Live Spread:

1. certificate.tar.gz
2. deployments.tar.gz
3. rage.tar.gz
4. codec.tar.gz
5. eoplugin.tar.gz
6. standalone-full.xml
7. ApplicationPackage
   1. LiveSpread.war
   2. input.properties
8. LIVESPREAD-FS.zip
9. LS\_File\_Conversion
10. Parser\_Scripts
11. AutomationJarAndProp
12. FileConversionService
13. setenv.sh
14. RDClientV15.zip
15. Rage\_Process\_XMLS

# Deployment plan of key components to be automated

## RAGE Application (LS Application):

1. Copy Extraction Script (Parser Script)
2. Install JDK
3. Set up OS user
4. Install Tomcat
5. Set up tomcat to run as service
6. Copy war file
7. Apply input.properties template
8. Restart tomcat

## RAGE Process:

1. Install JDK
2. Set up OS user
3. Install wildfly
4. Copy ear file
5. Apply standalone-full.xml template
6. Restart wildfly
7. Import RAGE processes xml

## Extraction Scripts:

1. Copy scripts
2. Apply templates

# Clone Auto installation script

Below is the GIT repository path from where you can clone our auto installation package.

URL: https://genpactdigitalengineering.visualstudio.com/CORA\_AI\_LIVESPREAD/\_git/LIVESPREAD\_AUTOMATION\_SCRIPTS

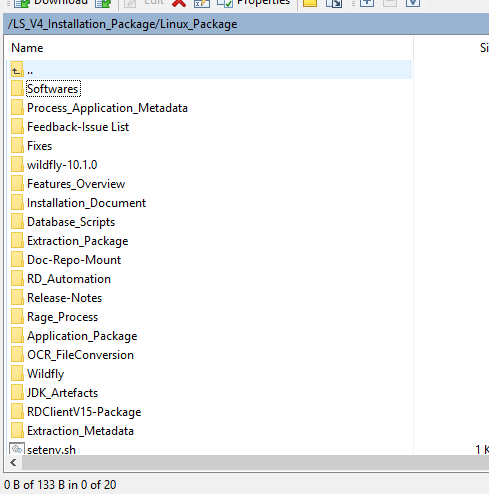
Or

Copy ansible automation script “live spread-automation” to location **/home/rage/CORA-LS-Automation/livespread-automation-ansible-ver1.1**

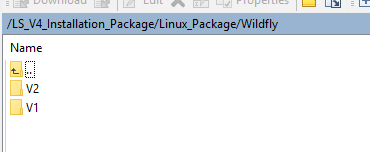
# Steps for Live Spread Automation:

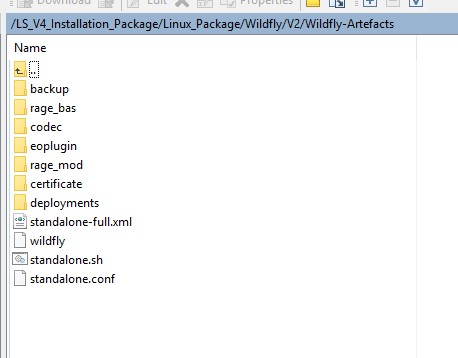
## Save all artefacts on standard format and name on FTP server.

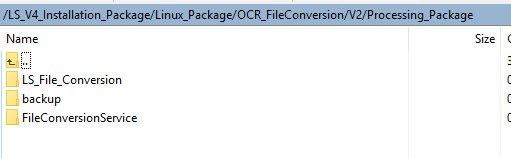
Save all artefacts delivered by individual teams on a central ftp server as below for reference:

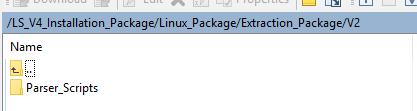


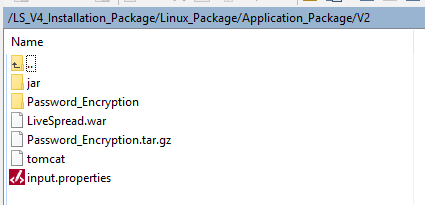
Versions were maintained as for example:



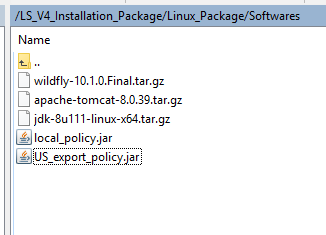








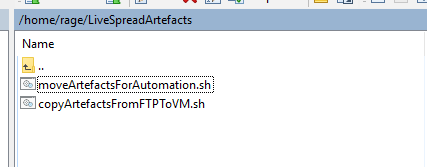
Software’s are kept on ftp server as:



1. Create a folder “**LiveSpreadArtefacts”** on /<root>/rage/LiveSpreadArtefacts
2. Copy Scripts **copyArtefactsFromFTPToVM.sh** and **moveArtefactsForAutomation.sh and deleteRageArtefactsFromAutoScripts.sh** from location **/home/rage/CORA-LS-Automation/livespread-automation-ansible-ver1.1/Scripts**

To location/home/rage/LiveSpreadArtefacts

It should look like below:

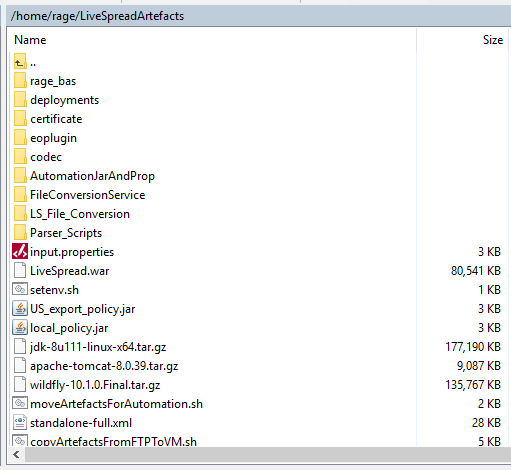


1. On terminal go to /home/rage/LiveSpreadArtefacts and execute below command, before executing the scripts check with the variable values used there.

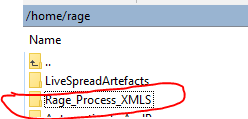


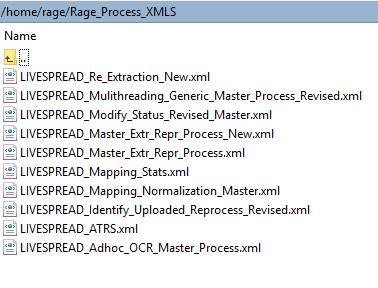
After script **copyArtefactsFromFTPToVM.sh** is executed**,** we will have below artefacts ready on specified format on VM:

**NOTE**: Please check the values of the variables on the script before executing.



A new folder is also created on /<root>/rage path with all rage process xml’s that needs to be imported.



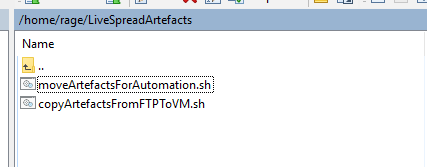


1. Now we need to move this artefacts to specific location of script, run below script



After running above script your above artefact folder should like below:

**NOTE**: Please check the values of the variables on the script before executing.

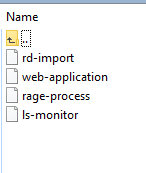


Verify on script if all your artefacts and software’s were copied on required location.

# Global Variables

We have a common set of global variable files for each modules (rage, UI, rage, rage automation). Before running the automation script we need to update all the required parameters with environment specific values.

**Location:** /home/rage/CORA-LS-Automation/livespread-automation-ansible-ver1.1/group\_vars



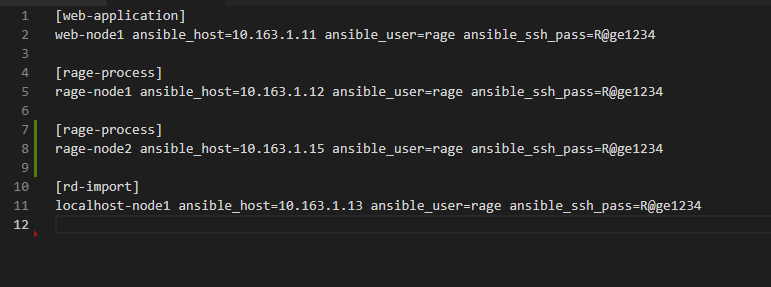
Each file is for a specific module below, Update below file with environment specific configuration:

1. rd-import – it’s for rage automation script module
2. web-application – it’s for Live Spread UI module
3. rage-process – it’s for rage module

**NOTE:** If rage have new processes or any new modules, please verify the general.properties template. We will need to append or delete accordingly

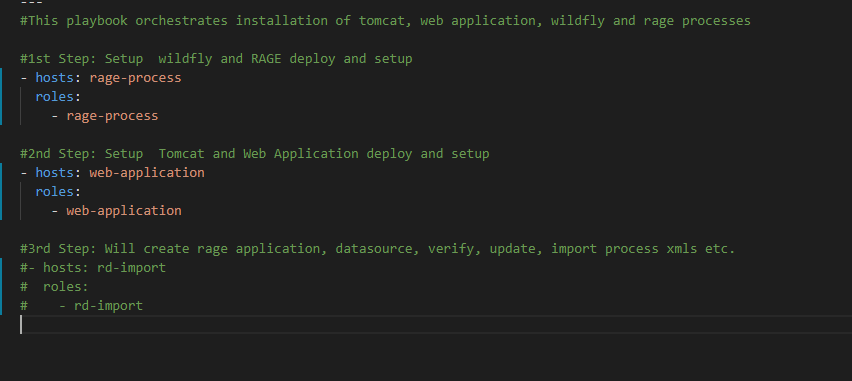
# Inventory File

We have a hosts file on below location, where we need to mention the IP of each corresponding host that we need to install, configure and deploy our specific modules.



# Main File (site.yml)

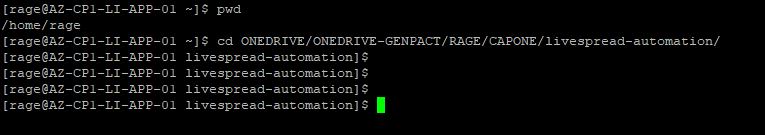
We have site.yml as our main file, where we can setup as our requirement which modules needs to be auto installed.



# How to run automation script

Below are the steps that is required to run your ansible script:

1. Open your Jump server Linux terminal.
2. Login to your terminal
3. Change to directory : /home/rage/CORA-LS-Automation/livespread-automation-ansible-ver1.1



1. Execute below command run ansible:

**ansible-playbook --ask-become-pass -i hosts site.yml**



We need to give our password to execute

After providing the password it executes…

# Ways and Options of running the Automation script

Automation scripts can be run with two options:

1. **Option1**: Run from scratch which includes rage setup as well as import/deploy of processes at one go,

UI Live spread setup etc. (Execution time: min 8 to 9 hrs.)

1. **Option2**: As rage process import, deploy and verify process takes lot of time hence we can skip import and deploy rage process through script automation and this will be handled by using db dumps, and files dump from rage server. (Execution time: 2 to 2.5 hrs.)

Depending on the approach above we can comment and uncomment some of the properties on general.properties file.

**NOTE:** If rage have new processes or any new modules, please verify the general.properties template. We will need to append or delete accordingly

# Common Issues after running

1. Sometimes some folders were shown as root access for a specific host after running the script. So in that case it’s better to run below command:

sudo chown –R <user>.<group> /home/rage/