**Universal Task Documentation**

Universal Automation Center support for Jenkins

UAC-Jenkins Integration

Associated Activities:

Date: 04 Aug 2020

Author: Ravi Kumar Murugesan

Revision: 01

CONFIDENTIALITY INFORMATION

Distribution list: Stonebranch Customer

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Changes** |
| 0.1 | 07-Aug-2020 | Ravi Kumar M | Draft |
|  |  |  |  |

**Abstract:**

UAC-Ansible Integration with Jenkins allow you to execute start or trigger Jenkins Build process ,create Jenkins Build Job& other Jenkins functionality to orchestrate from UAC.

Contents

[1 Disclaimer 3](#_Toc47541393)

[2 Scope 3](#_Toc47541394)

[3 Introduction 3](#_Toc47541395)

[3.1 Ansible – UAC Integration Overview 3](#_Toc47541396)

[3.2 Implementation Details 4](#_Toc47541397)

[4 Installation 4](#_Toc47541398)

[4.1 Software Requirements for Linux Agent 4](#_Toc47541399)

[4.2 Installation Steps 4](#_Toc47541400)

[5 Universal Task Configuration 5](#_Toc47541401)

[6 Universal Tasks for Ansible-Exec-Playbook 5](#_Toc47541402)

[6.1 Ansible Universal Task Fields 5](#_Toc47541403)

[7 Document References 6](#_Toc47541404)

# Disclaimer

No support and no warranty are provided by Stonebranch GmbH for this document and the related Universal Task. The use of this document and the related Universal Task is on your own risk.

Before using this task in a production system, please perform extensive testing.

Stonebranch GmbH assumes no liability for damage caused by the performance of the Universal Tasks

# Scope

This document provides a documentation how to install and use the Universal Tasks for Jenkins. If more Task will be created in the future this document will be updated accordingly.

**High Level UAC-Jenkins Integration View:**

Create Jenkins Build Jobs

Jenkins



Universal controller



Set Next Build Number/Trigger and monitor Jenkins Build job

Rename/Copy/Enable/disable/delete/list build jobs

<

Enable/disable Nodes

Jenkins to Universal controller API call

# Introduction

## UAC – Jenkins Integration Overview

Jenkins is a continuous integration / continuous delivery open source tool based on java platform. It is used to build, test & deploy applications continuously making it easier to integrate changes to the application in form of pipelines.

Universal controller is integrated with Jenkins using python Jenkins module. This integration helps managing the operational aspects of the Jenkins functionality orchestrated from universal controller and can be very much collaborative with respect to deployment of applications and taking it over to operations in a well-controlled and automated fashion.

As show in the above sections, UAC communicates with Jenkins through the python Jenkins module and Jenkins can make REST-API calls to universal controller to trigger any task or workflow.

RightNow UAC can play a part in orchestrating the below Jenkins functionality and in future based on the customer needs & usage of this universal task ,further functionality can be adapted forming a collaborative approach between development and operations and single pane of glass providing a complete view of development and deployment progress .

* Get Jenkins Jobs build information
* Create a Jenkins Build Job
* Get Last build number of a Jenkins Job
* Set Next build number for a Jenkin job
* Trigger or start a existing build job in Jenkins
* List running build info from Jenkins
* Enable/Disable Jenkins Job
* Delete/Copy/Rename Jenkins Job
* Get Console output for a build
* Enable / disable Nodes in Jenkins
* Get all installed plugins info
* Install a plugin for Jenkins environment.

## Implementation Details

Details about the universal tasks for Jenkins as below:

* Universal task uses Jenkins module in python
* Accepts input parameters like Select a Jenkins functionality ,Jenkins Job Name, Create Jenkins job XML, Build Number, Jenkins connection timeout parameter, Nodes and Plugins
* Universal controller hosted environment should be able to post the message to Jenkins environment using the Jenkins API Calls.
* The Universal Task supports both Universal Agent for Linux/Unix and Windows.

# Installation

## Software Requirements for Linux Agent

**Universal Task name:**  Jenkins

Requirements:

* Python 3.6
* For Python the following modules are required:
  + *sys, for system-specific parameters and functions*
  + *requests, Jenkins python module to interact with Jenkins server.*

***Note: Please check requests python library is available already, if not then it needs to be added via python installer***

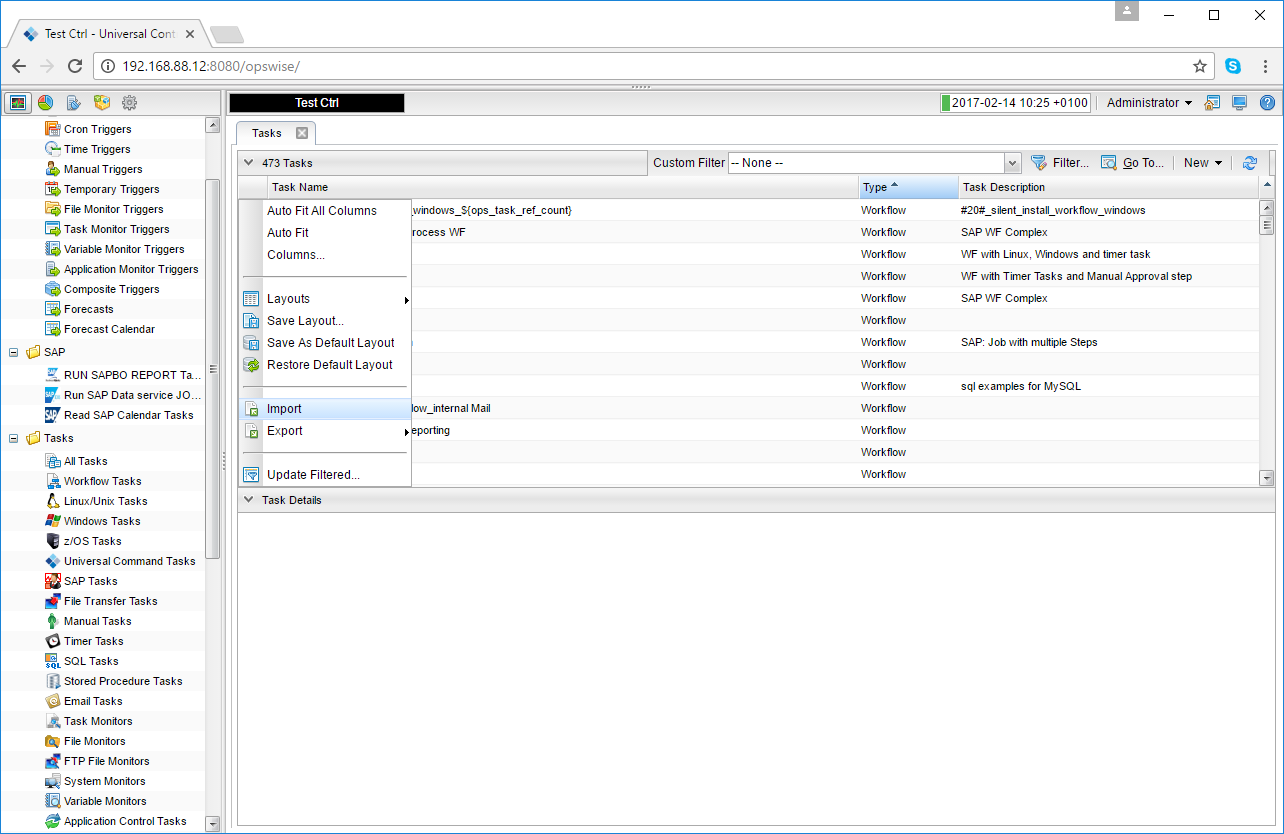
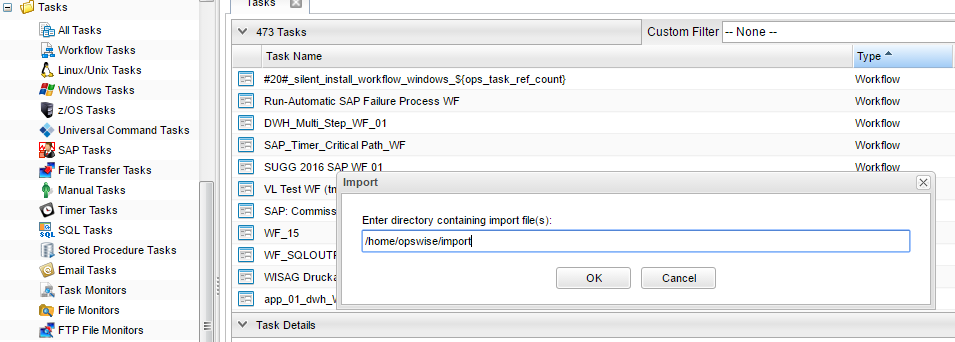
* *pip install requests*
* *pip install Jenkins*
* Universal Controller V6.4.7.0 or higher
* Universal Agent V6.5.0.0 or higher installed on a Linux/Windows Server

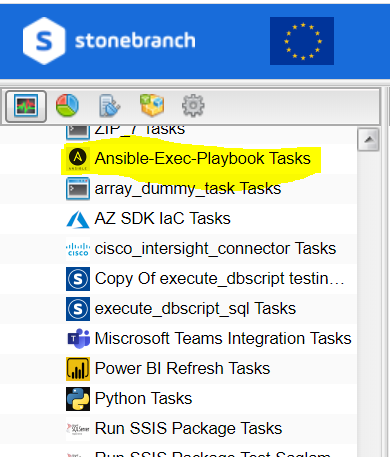
## Installation Steps

The following describes the installation steps:

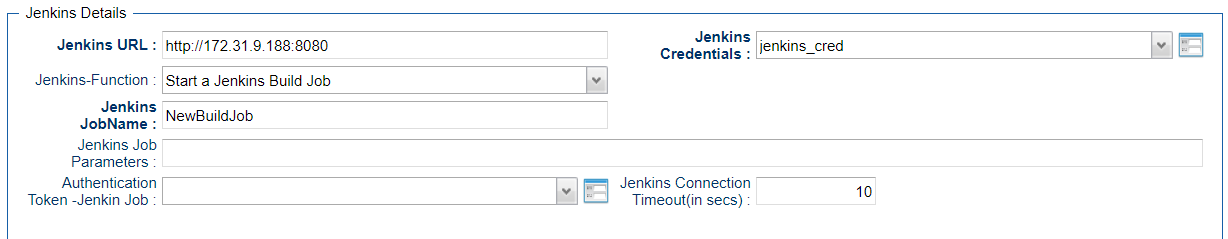
1. **Import Jenkins Universal Task including the Universal Template to your Controller**

Go to “All Tasks” and load via the Import functionality the Universal Tasks configuration into the Controller.

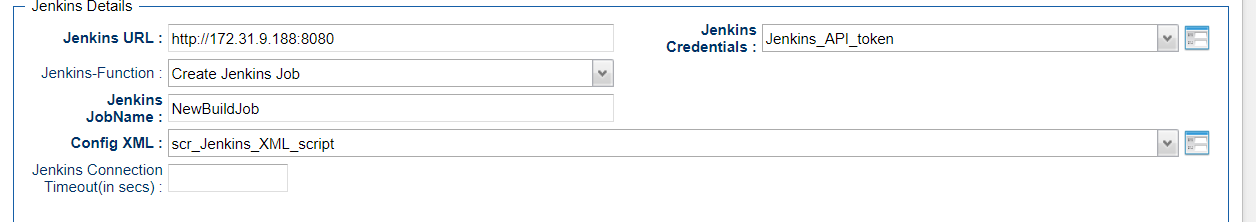
 

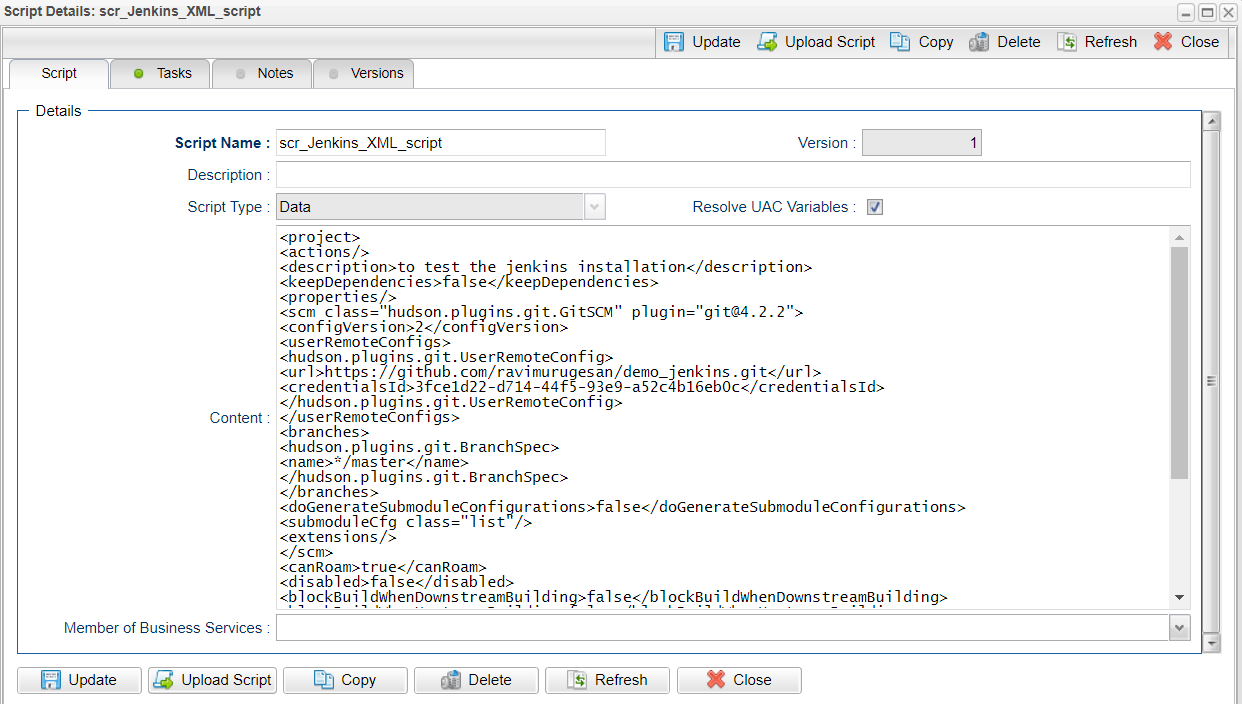


# Universal Task Configuration Samples

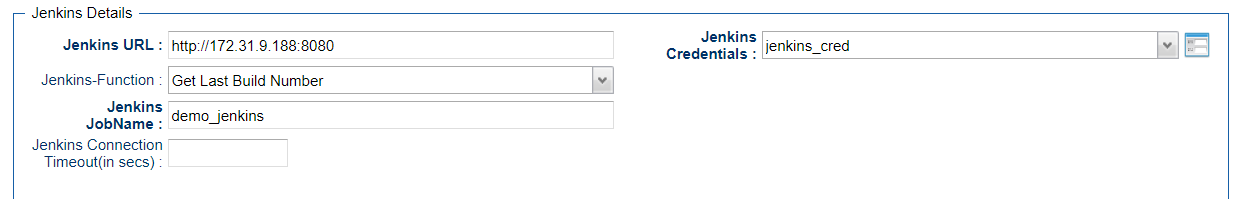
**1.1 Start a Jenkins Build**

**1.2 Creating a Jenkins Job**

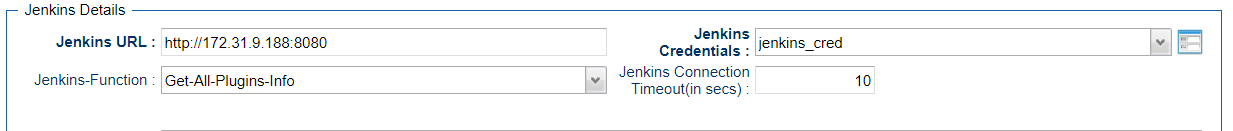




**1.3 Get Last Build Number of a Jenkins Job**



**1.4 List Installed Plugins**



# Universal Tasks for Jenkins

The following contents describes the Ansible notifications through the universal task.

| UT Name | Description |
| --- | --- |
| Jenkins | Manage and orchestrate functionality in Jenkins and taking over the application operations seamlessly |

## Jenkins Universal Task Fields

**Field Description:**

|  |  |
| --- | --- |
| Field | Description |
| Agent | Universal agent where ansible is installed |
| Jenkins URL | URL of the Jenkins server, to make api calls |
| Jenkins Credentials | Jenkins User credentials or auth token to authenticate API calls |
| Jenkins Function | Select the desired function you would need to perform in Jenkins |
| Jenkins Job Name | Name of the Jenkins Job Name |
| Jenkins Job Parameters | The parameters that would need to passed along for starting a Jenkins Build |
| Job Token | If the Jenkins can be triggered remotely by using a job token then pass on the Job token parameter |
| Config XML | Provide the XML script for the creation of new job in Jenkins |
| Rename Job | specify the job name that needs to be renamed |
| From jobname | give the existing from where its needs to be copied eg: demo\_job or folder/demo\_job |
| To Job Name | copy/rename to a new Job Name eg: demo\_job or folder/demo\_job |
| Enable Node | specify the node name that needs to be enabled |
| Disable Node | specify the node name that needs to be disabled |
| Delete Job | Specify the job that needs to be deleted in Jenkins |
| Next Build Number(+ Last Build Number) | should be greater than the last builder otherwise jenkins will ignore the request |
| Enable Job | Specify the job name that needs to be enabled |
| Disable Job | specify the job that needs to be disabled |
| Jenkins Build Number | Jenkins build number for the job(int) |
| Jenkins Connection Timeout(in secs) | Specify the connection time out interval in seconds |
| Jenkins Plugin Name | Provide the Jenkin Plugin short name to be installed |

# Document References

This document references the following documents:

|  |  |
| --- | --- |
| Ref# | Description |
| 1. Jenkins | <https://pypi.org/project/python-jenkins/> |