**Universal Task Documentation**

Universal Automation Center support for AWS EC2

UAC-AWS EC2 Integration

Associated Activities:

Date: 17 May 2020

Author: Ravi Kumar Murugesan

Revision: 01

CONFIDENTIALITY INFORMATION

Distribution list: Stonebranch Customer

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Changes** |
| 0.1 | 10-Aug-2020 | Ravi Kumar M | Draft |
| 0.2 | 13-Aug-2020 | Ravi Kumar M | Update |

**Abstract:**

The here described Universal Tasks allow create and manage AWS EC2 Instance, So this helps provisioning EC2 resources dynamically and controlling the EC2 instance start/stop/termination appropriately.

Contents

[1 Disclaimer 3](#_Toc47973009)

[2 Scope 3](#_Toc47973010)

[3 Introduction 3](#_Toc47973011)

[4 Installation 4](#_Toc47973012)

[4.1 Software Requirements 4](#_Toc47973013)

[4.2 Installation Steps 5](#_Toc47973014)

[5 Universal Task Configuration 6](#_Toc47973015)

[6 Universal Tasks for AWS EC2 7](#_Toc47973016)

[6.1 Create New EC2 Instance 7](#_Toc47973017)

[6.2 AWS EC2 Start-Stop-Terminate-status check 9](#_Toc47973018)

[7 Document References 10](#_Toc47973019)

# 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.

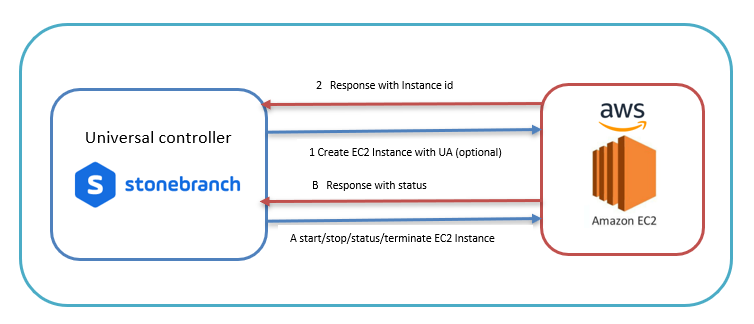
Before using this task in a production system, please perform extensive testing.

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 AWS EC2 management. If more Task will be created in the future this document will be updated accordingly.

**High Level View:**



# Introduction

Creation and manage of cloud computing resources becomes an integral part of most modern IT landscapes. With Universal Automation Center you can securely automate your AWS, Azure, or any other Cloud computing resources dynamically and integrate them into your existing scheduling flows.

As security is one of the key concerns, when moving to the cloud, the provided solution supports multi-level of security:

* All Credential for AWS EC2 (Access Key, Secret Access key and Region) are stored in an encrypted form in the database
* The current version of AWS EC2 Universal task does not support Connections towards the AWS VPC via a Proxy Server, but this can be certainly supported if there is a requirement
* Secure access to AWS Ec2 instances using *AWS EC2 policies* can be configured

The below list describes the Universal Tasks that focuses on the AMAZON AWS EC2. A similar solution as for AWS EC2 is also available for Microsoft Azure (except Azure VM creation).

Some details about the universal tasks for AWS EC2:

* The Universal Tasks are calling the python module Boto3 - the Amazon Web Services (AWS) SDK for Python. Both Boto3 API Types are used the "low-level" Client API and Resource APIs.
* The python boto3 module is called by a Universal Agent running on a Linux Server or Windows Server – Note: This document focuses on the Linux Version
* The Server Running the Universal Agent needs to have Python 2.7.x or 3.6.x installed
* All Credential for AWS EC2 (Access Key, Secret Access key and Region) are stored in an encrypted form in the database
* You can configure all connection Parameters for the Proxy and AWS via the Universal Task
* You can select different log-levels e.g. Info and debug

The following Universal Task for AWS EC2 have been implemented as described below:

| Command | UT Name | Description |
| --- | --- | --- |
| Create New EC2 Instance | AWS-EC2-Create-Instance | Creates a EC2 Instance based on the parameters that are provided in the form |
| Launch EC2 from template | AWS-EC2-Create-Instance | Create a EC2 Instance based on a template in AWS |
| Start EC2 Instance | AWS-EC2-Start-Stop-Terminate-Instance | Start one or Multiple EC2 instance |
| Stop EC2 Instance | AWS-EC2-Start-Stop-Terminate-Instance | Stops one or Multiple EC2 instance |
| Terminate EC2 Instance | AWS-EC2-Start-Stop-Terminate-Instance | Terminate one or Multiple EC2 instance |
| EC2 status | AWS-EC2-Start-Stop-Terminate-Instance | Provides one or Multiple EC2 instance status |

**Create EC2 Instance:**

This Universal task allows you to create an EC2 instance with parameters either supplied in the task form or simply creating an EC2 instance from the existing AWS Launch Template and this tasks optionally provides the feature to install Linux/UNIX universal agent in the newly provisioned EC2 Instance.

**Start-Stop-Terminate EC2 Instance:**

This universal task lets you provide one or more Instance ids as input for the start, stop &terminate EC2 functions and then it used python boto3 to interact with AWS platform using the credentials supplied in the task.

# Installation

## Software Requirements

**Universal Task name:** AWS-EC2-Create-Instance & AWS-EC2-Start-Stop-Terminate-Instance

**Pre-requisite Setup:**

For the set-up you need:

1. Python 3.6 – can be installed as part of the Universal Agent
2. For Python the following modules are required:
   * ***Re****,* to support regular expression matching operations
   * ***glob****, to* find Unix pathnames matching a specified pattern
   * ***os****,* to support operating system dependent commands
   * ***sys***, for output re-direct processing
   * ***datetime, date*** and time stamps for messages
   * ***logging***, to provide logging capabilities for debug, info etc.
   * ***boto3,*** provide the Amazon Web Services (AWS) SDK for Python
   * ***botocore,*** Botocore is a low-level interface to a growing number of Amazon Web Services. Botocore serves as the foundation for the AWS-CLI command line utilities.
   * ***argparse***, to allow testing of the Universal TPL. script on the command line

*Note: Only the module* ***boto3*** *and* ***botocore*** *need to be added to python 3.6.x. e.g. using pip.*

* *pip install boto3*
* *pip install botocore*

1. Universal Controller 6.4.5.x or higher
2. Universal Agent 6.4.2.2 or higher installed on a Linux Server
3. An Amazon AWS account with EC2 admin access to try it out

## Installation Steps

The following describes the installation steps:

1. **Check the current Python Version**

*python -V (Note: Captial “V”)*

If your Version is Python 3.6 or later all is fine. If a no python or a lower Version has been installed upgrade your python Version or install the Universal Agent with the Python binding option (--python yes). This option will install python 3.6. along with your universal agent.

e.g.

sudo sh ./unvinst --network-provider oms --oms-servers 7878@192.168.88.12 --oms-port 7878 --oms-autostart no --ac-netname OPSAUTOCONF --opscli yes --python yes

**NOTE: the above install string does not work in case you did a user mode install**

Official Download link: <https://www.python.org/downloads/>

Note: To check the current Python Version type in a shell: python -V (capital V)

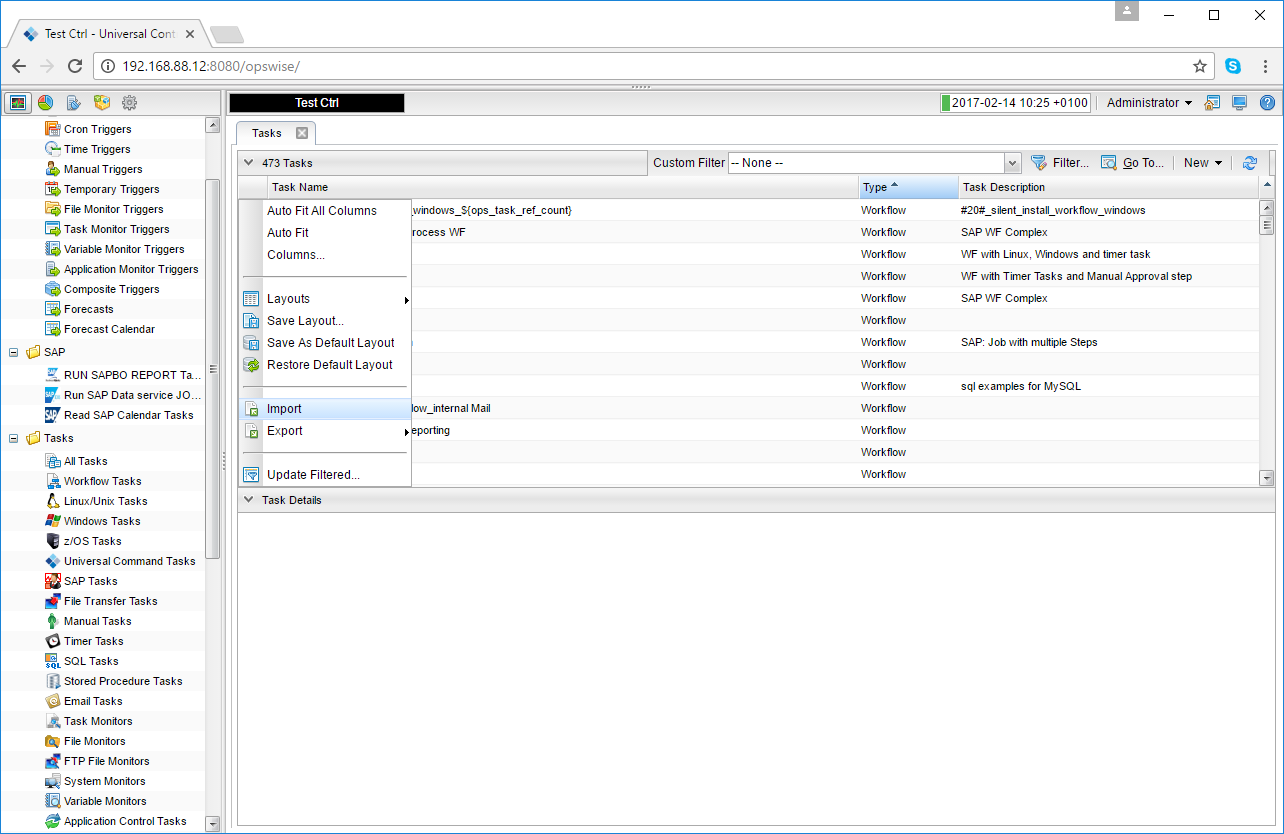
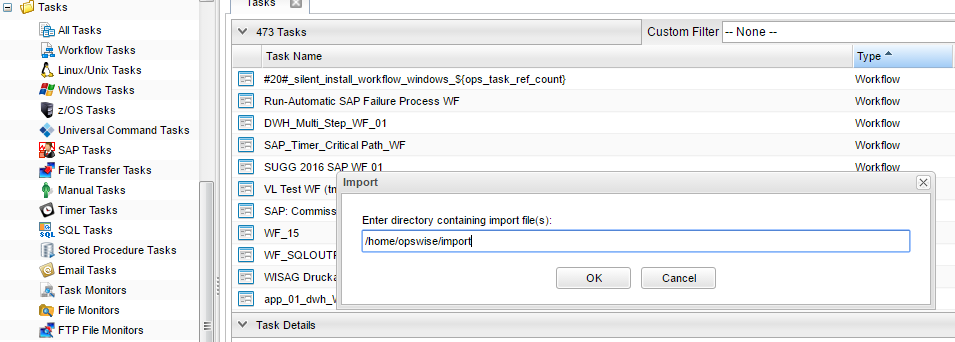
1. 17TAdd the boto3 and botocore modules to your python installation

17TIn a command shell run as root or sudo:

* *pip install boto3*
* *pip install botocore*

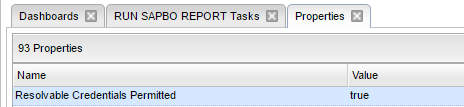
1. **Import each AWS EC2 Universal Task including the Universal Template to your Controller**

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

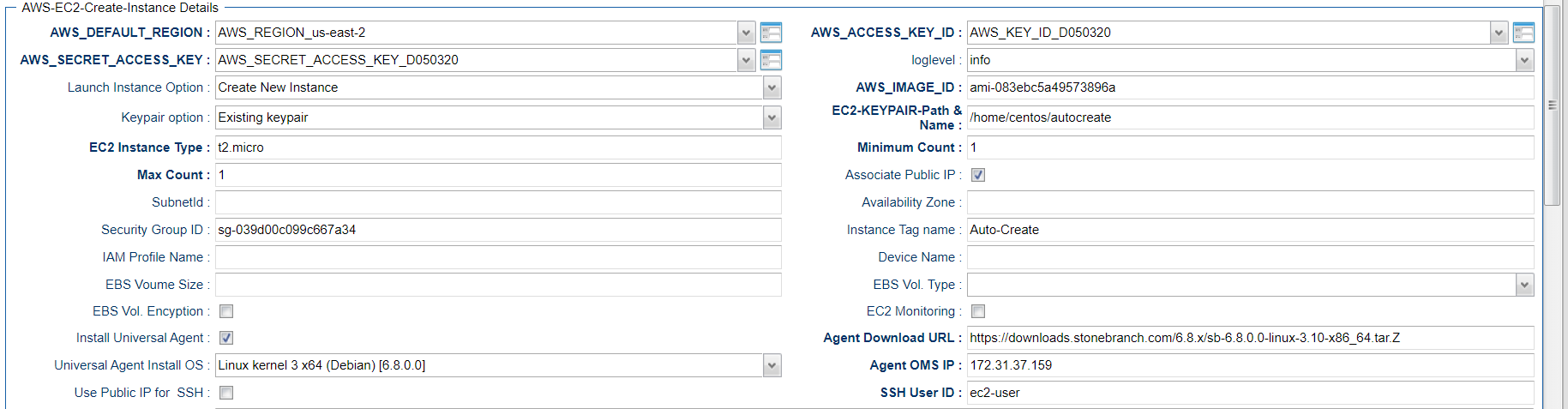
# Universal Task Configuration

1. **Activate: Resolvable Credentials in Universal Automation Center:**



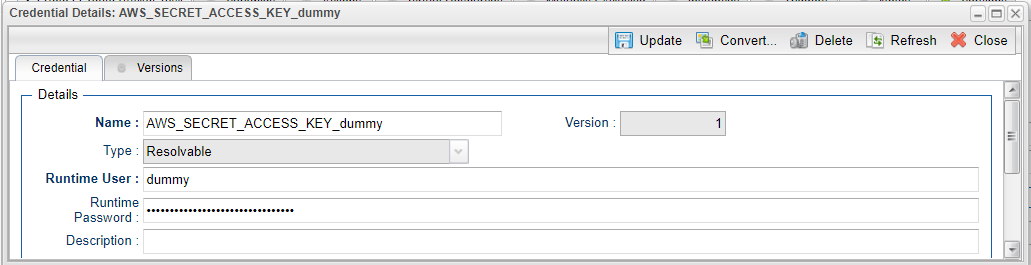
1. **Fill Out the Universal Task for each AWS command, which you want to execute:**

In the example below the *EC2 Boto3 Create New Instance* was selected



**Fill out or select the required Credentials for AWS**

In the example below the *AWS-SECRET-ACCESS-KEY* credentials are shown:



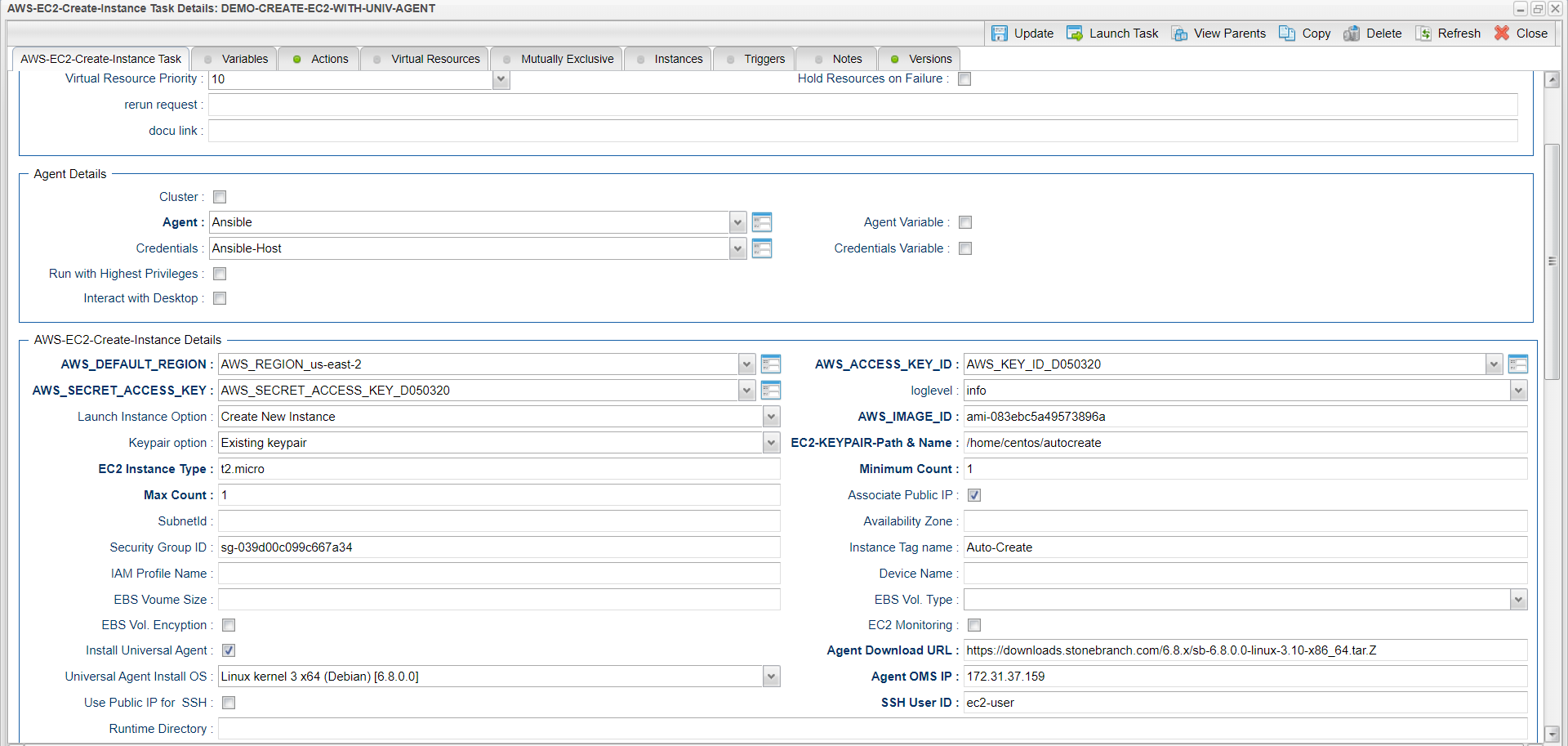
# Universal Tasks for AWS EC2

The following chapter describes the provided AWS Ec2 Universal Tasks

## Create New EC2 Instance

| Command | UT Name | Description |
| --- | --- | --- |
| Create New EC2 Instance | AWS-EC2-Create-Instance | Creates a EC2 Instance based on the parameters that are provided in the form |
| Launch EC2 from template | AWS-EC2-Create-Instance | Create a EC2 Instance based on a template in AWS |

**Task Screenshot:**



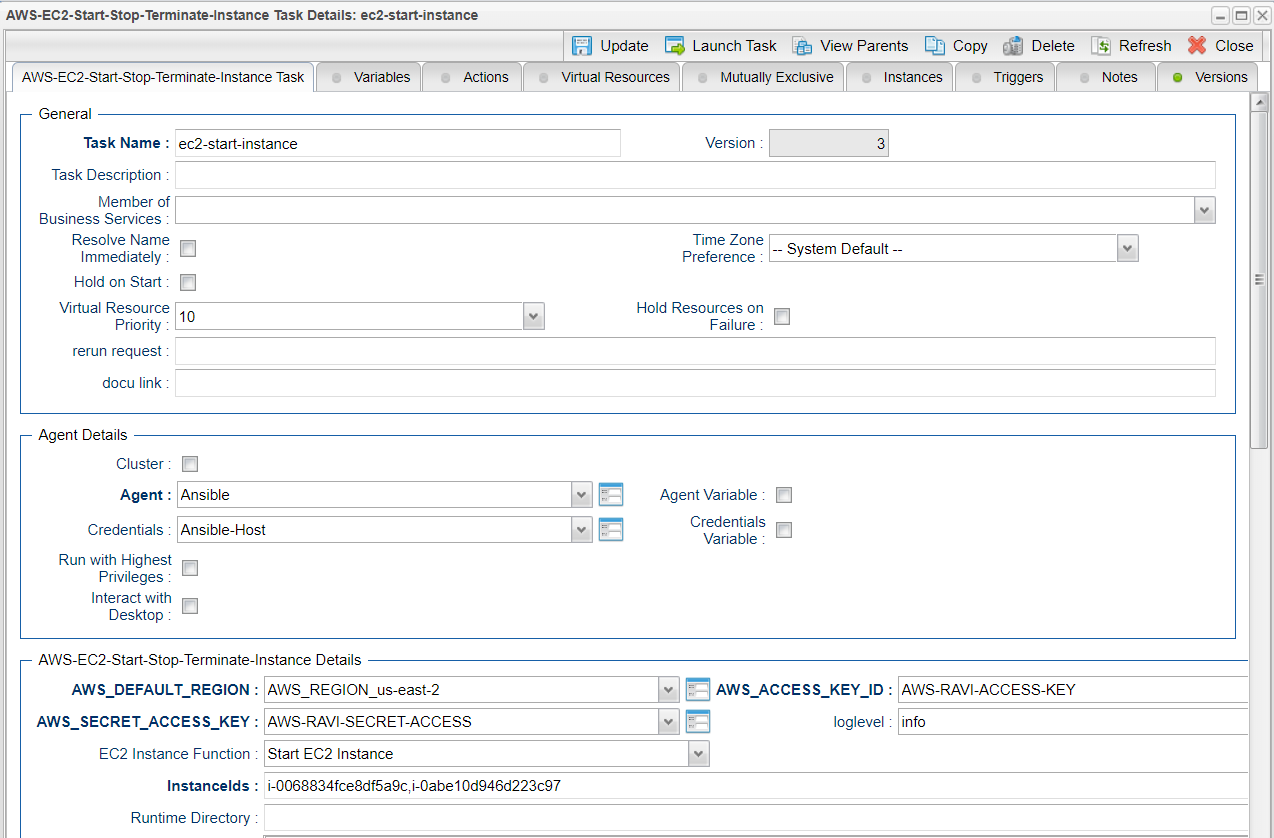
**Field Description:**

|  |  |  |
| --- | --- | --- |
| Field | Required | Description |
| Agent | Mandatory | The Linux Universal Agent, which runs the Python boto3 module to call the AWS EC2 commands |
| Credentials | Optional | The Credentials used on the Linux Server |
| loglevel | Mandatory | logging settings DEBUG, INFO, WARNING, ERROR, CRITICAL |
| AWS-DEFAULT-REGION | Mandatory | AWS Region kept as credential |
| AWS-SECRET-ACCESS-KEY | Mandatory | AWS Secret Key |
| AWS-ACCESS-KEY-ID | Mandatory | AWS Access Key |
| Launch Instance Option | Mandatory | Select either launch from template or create a brand new ec2 instance with the parameter supplied in the form |
| LaunchTemplateName | Mandatory | Mandatory if launch\_instance\_option=” Launch from template” |
| AWS\_IMAGE\_ID | Mandatory | Provide the AWS machine ID ,Mandatory if launch\_instance\_option=” new\_instance” |
| Keypair option | Mandatory | PEM file creation choice , Select either existing Key pair or New Key pai |
| EC2-KEYPAIR-Path & Name | Mandatory | Provide Keypair file name and the path (Donot give the extension) for new and for existing keypair just the name |
| EC2 Instance Type | Mandatory | provide ec2-instance type like t2. Micro , if Launch from template = “Create New Instance” |
| Minimum Count | Mandatory | Minimum Count of instance that need to be created, if Launch from template = “Create New Instance” |
| Max Count | Mandatory | Max count of instance that needs to be created, if Launch from template = “Create New Instance” |
| associate\_public\_ip | Optional | If a public IP needs to be created when a instance is created |
| SubnetId | Optional | Provide subnetID where the instance to be associated within AWS |
| Availability Zone | Optional | Provide Availability Zone where the instance to be associated within AWS |
| Security Group ID | Optional | provide security group ids, if multiple ID's then separate by comma |
| Instance Tag name | Optional | EC2 Instance Tag Name |
| iam\_instance\_profile\_name | Optional | If applicable provide the IAM Instance Profile Name |
| device\_name | optional | Provide the device name e.g. /dev/sda1 |
| ebs\_volume\_size | Optional | Provide EBS Volume size |
| EBS Vol. Type | Optional | Select either standard or io1 or gp2 or sc1 or st1 |
| EBS Vol. Encyption | Optional | Check if encryption needs needed |
| EC2 Monitoring | Optional | Check this box if detailed monitoring required |
| Install Universal Agent | Optional | Check this box if you would need to install universal agent with this new EC2 instance created |
| Agent Download URL | Mandatory | Provide the path to download the agent URL, if install universal agent option is selected |
| Universal Agent Install OS | Optional | select the OS where universal agent needs to be installed |
| Agent OMS IP | Mandatory | Provide the OMS server IP for the universal agent to be connected after installation , if install universal agent option is selected |
| Use Public IP for SSH | Mandatory | Select if you would need to use the public or provide IP for SSH |
| os\_user\_id | Mandatory | Provide the OS user ID that will be used to make SSH connection |

## AWS EC2 Start-Stop-Terminate-status check

| Command | UT Name | Description |
| --- | --- | --- |
| Start EC2 Instance | AWS-EC2-Start-Stop-Terminate-Instance | Start one or Multiple EC2 instance |
| Stop EC2 Instance | AWS-EC2-Start-Stop-Terminate-Instance | Stops one or Multiple EC2 instance |
| Terminate EC2 Instance | AWS-EC2-Start-Stop-Terminate-Instance | Terminate one or Multiple EC2 instance |
| EC2 status | AWS-EC2-Start-Stop-Terminate-Instance | Provides one or Multiple EC2 instance status |

**Task Screenshot:**



**Field Description:**

|  |  |  |
| --- | --- | --- |
| Field | Required | Description |
| Agent | Mandatory | The Linux Universal Agent, which runs the Python boto3 module to call the AWS EC2 commands |
| Credentials | Optional | The Credentials used on the Linux Server |
| log level | Mandatory | logging settings DEBUG, INFO, WARNING, ERROR, CRITICAL |
| AWS-DEFAULT-REGION | Mandatory | AWS Region kept as credential |
| AWS-SECRET-ACCESS-KEY | Mandatory | AWS Secret Key |
| AWS-ACCESS-KEY-ID | Mandatory | AWS Access Key |
| EC2 Instance Function | Mandatory | Select either start or stop or terminate or check ec2 instance |
| Instance ID’s | Mandatory | Provide the Instance ID which you would need to start or stop or term |

# Document References

This document references the following documents:

|  |  |
| --- | --- |
| Ref# | Description |
| 1. Slack | <https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/ec2.html> |
| 1. requests | <https://pypi.org/project/requests/#description> |
| 1. Universal controller API | <https://docs.stonebranch.com/confluence/display/UC67/RESTful+Web+Services+API> |