stonebranch

Universal Task Documentation

Universal Automation Center support for scheduling Azure Blob Storage file Transfers

ut-azure-blobstorage-check-for-blob-in-containers-linux

Associated Activities:

Date: 10 April 2018

Author: Nils Buer

Revision: 01

CONFIDENTIALITY INFORMATION

Distribution list: Stonebranch Marketplace

Revision	Date	Author	Changes
00	20180405	Nils Buer	Initial Document (WIP)
01	20180410	Nils Buer	Test Cases Added for blob download
02	20180416	Nils Buer	Screenshots + Testcases executed
03	20180419	Nils Buer	Delete container checks for blobs before

Abstract:

The here described Universal Tasks allow to Transfer and retrieve files from Azure Blob Storage in the Cloud. As a result, you can integrate any Azure Blob Storage file transfers into you existing or new scheduling workflows, providing a true hybrid cloud (on-premise and cloud computer) file transfer solution.

Contents

1 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

2 Scope

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

3 Introduction

Storing data in the cloud becomes an integral part of most modern IT landscapes. With Universal Automation Center you can securely automate your AWS, Azure or any other Cloud File Transfer and integrate them into your existing scheduling flows.

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

- All Credential for Azure Blob Storage are stored in an encrypted form in the database
- Connections towards the Azure Blob Storage via a Proxy Server are supported

The here described Series of Universal Tasks focuses on the Azure Blob Storage file transfer. A similar solution as for Azure is also available for Amazon S3 storage.

Some details about the universal tasks for Azure Blob Storage:

- The Universal Tasks are calling the Microsoft Azure Storage SDK for Python
- The python *azure-storage-blob* 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 Universal Task support encrypted connections via a Proxy Server
- All Credential for Azure (account_name, account_blob) are stored in an encrypted form in the database
- You can configure all connection Parameters for the Proxy and Azure via the Universal Task
- You can select different log-levels e.g. Info and debug

The following Universal Task for Azure Blob Storage have been implemented:

Command	UT Name	Description
Create_container	ut_azure_blobstorage_create_container_linux	Creates a container in Azure Blob Storage

Command	UT Name	Description
Monitor_blob_in_container	ut_ azure_blobstorage _Monitor_blob_in_container_linux	Monitors at a given interval for a blob in a container
Copy_file_to_container	ut_ azure_blobstorage _Copy_file_to_container_linux	Copies a file to a container
List_blobs_in_container	ut_ azure_blobstorage _List_blobs_in_container_linux	Lists all blobs in a container
Download_file_from_container	ut_ azure_blobstorage _Download_file_from_container_linux	Downloads an Azure Blob to a local file
Delete_blob_from_container	ut_ azure_blobstorage _Delete_blob_from_container_linux	Deletes a blob from a container
Delete_empty_container	ut_ azure_blobstorage _Delete_empty_container_linux	Deletes an empty container
List_containers	ut_azure_blobstorage _List_containers_linux	List all containers of an Azure account
Check_for_blobs_in_container	ut_azure_blobstorage_Check_for_blob_in_containers_linux	Checks for the existence of a blob in a container

4 Installation

4.1 Software Requirements

Universal Task name: *ut_azure_blobstorage_<xxx>_linux*

Related UAC XML Files for template and task: *Github repository*

Software used:

For the set-up you need:

1. Python 2.7.x (or 3.6.x) for Linux installed on a server where a Universal Agent is installed.
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.
- ***argparse***, to allow testing of the Universal TPL. script on the command line
- ***azure-storage-blob***, The Azure libraries for Python to use Azure services and manage Azure resources
- ***azure-storage-logging***, provide enhanced logging for Azure storage services

*Note: Only the module **azure-storage-blob** and **azure-storage-logging** need to be added to python 3.6.x. e.g. using pip.*

- *pip install azure-storage-blob*
- *pip install azure-storage-logging*

3. Universal Controller 6.4.5.x or higher
4. Universal Agent 6.4.2.2 or higher installed on a Linux Server
5. An Azure account to try it out

4.2 Installation Steps

The following describes the installation steps:

1. **Install Python 2.7.x or 3.6.x for Linux on the Universal Controller server or any Linux Server running a Universal Agent.**

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

   Note:

   In most cases python is already available on Linux. Check availability with: *python -V*

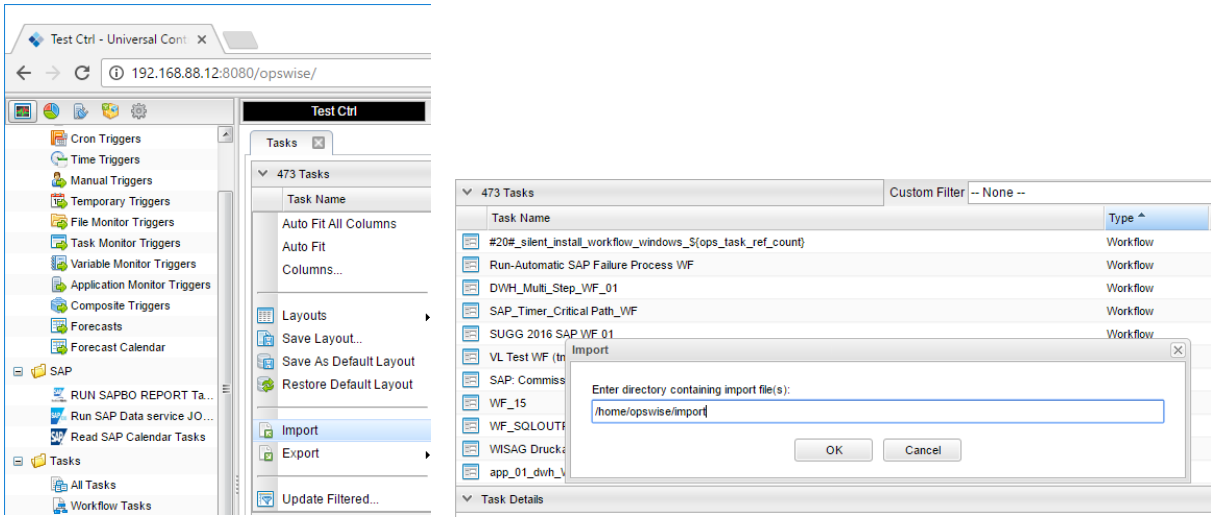
2. **Add the *azure-storage-blob* and *azure-storage-logging* modules to your python installation**

   In a command shell run as root or sudo:

   - *pip install azure-storage-blob*
   - *pip install azure-storage-logging*

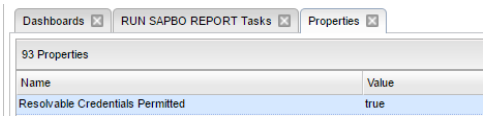
3. **Import each Azure Blob Storage Universal Task including the Universal Template to your Controller**

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



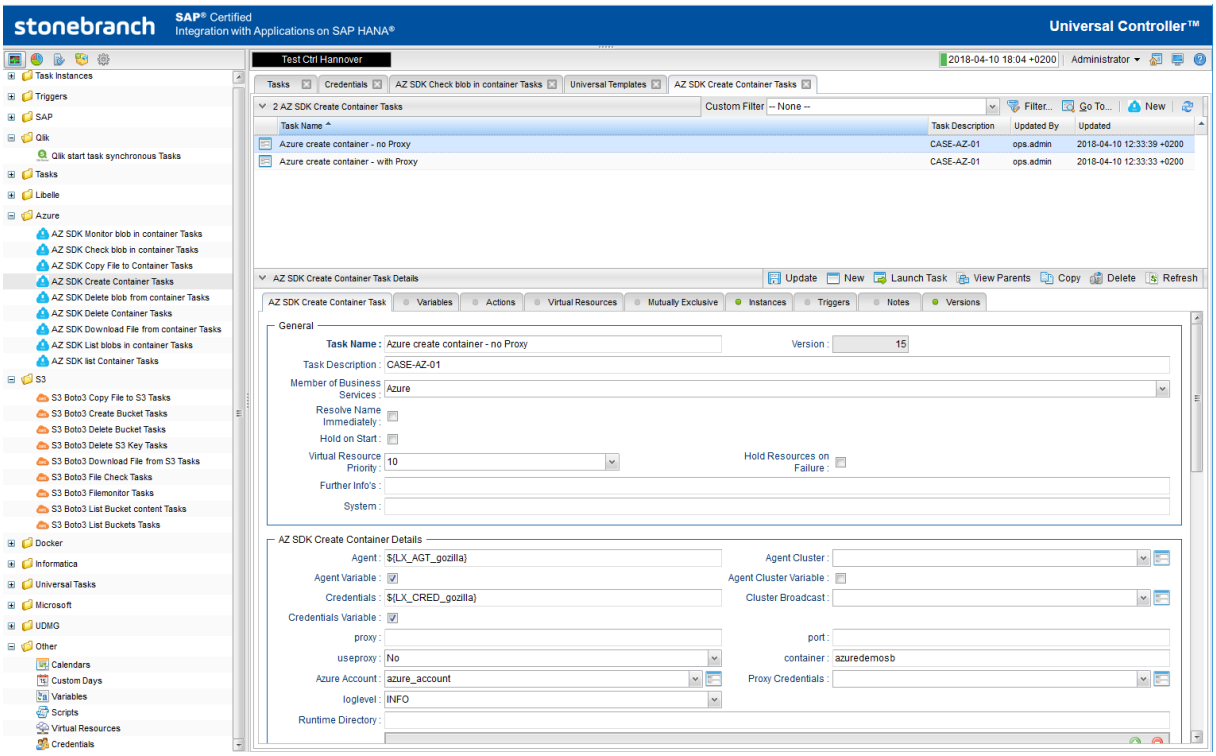
5 Universal Task Configuration

1. Activate: Resolvable Credentials in Universal Automation Center:



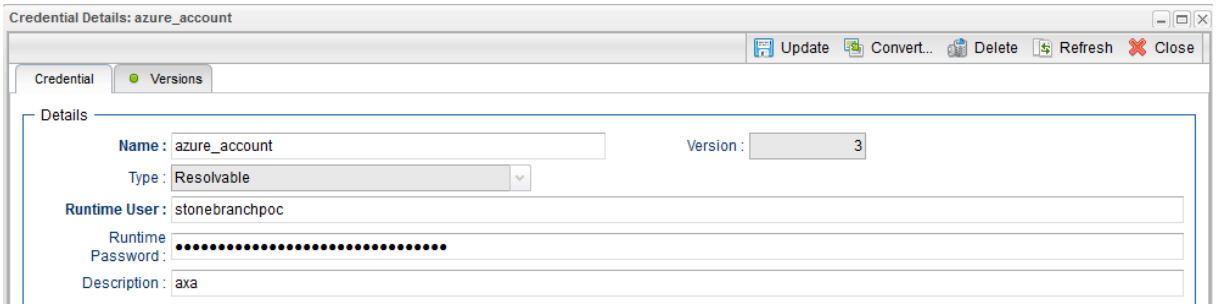
2. Fill Out the Universal Task for each Azure Blob Storage command, which you want to execute:

In the example below the Azure *Create Container Task* was selected



Fill out or select the required Credentials for Azure and optionally a Proxy Server

In the example below the azure_account credentials are shown:



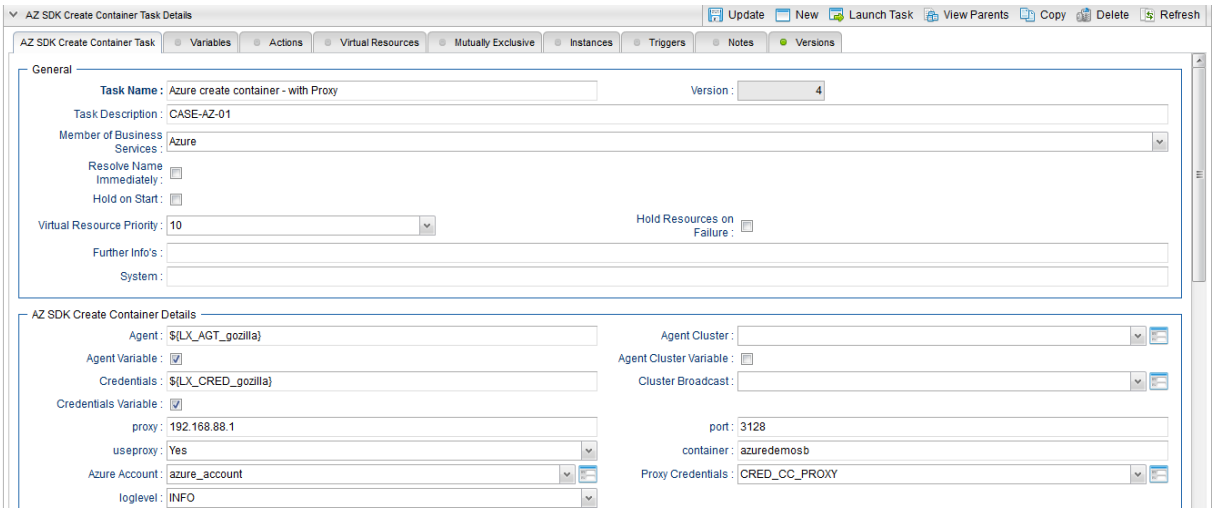
6 Universal Tasks for Azure Blob Storage

The following chapter describes the provided Azure Blob Storage Universal Tasks.

6.1 Create_Container

Command	UT Name	Description
Create_Container	ut_AZURE_BLOBSTORAGE_Create_Container_linux	Creates a container in AZURE BLOB STORAGE

Task Screenshot:



Field Description:

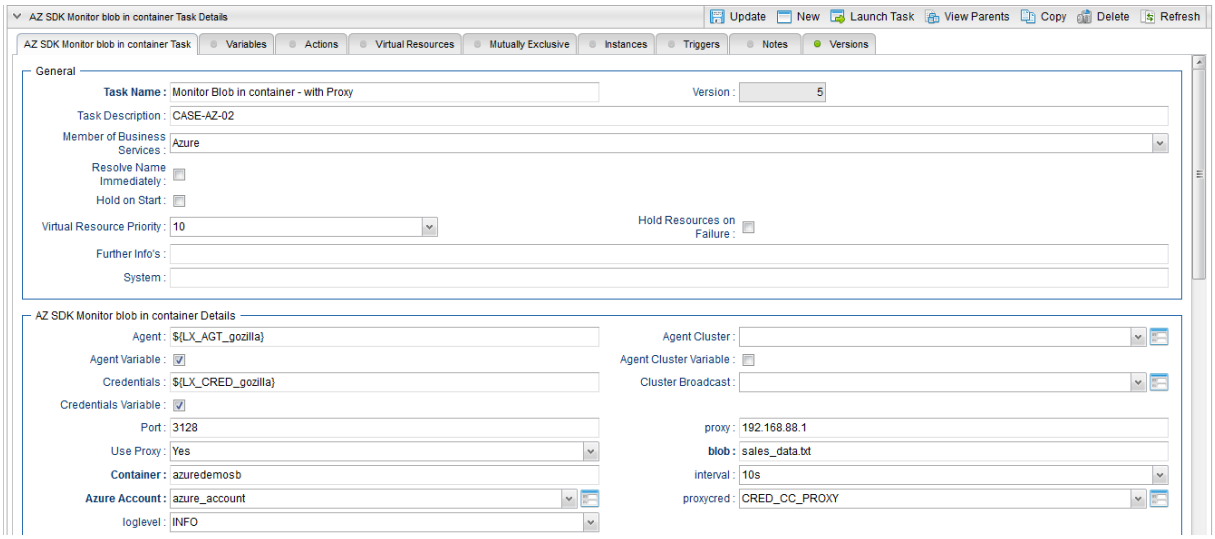
Field	Required	Description
Agent	Mandatory	The Linux Universal Agent, which runs the Python azure-storage-blob module to call the AZURE BLOB STORAGE commands
Credentials	Optional	The Credentials used on the Linux Server
Use Proxy (yes/no)	Mandatory	Decide If a Proxy Server should be used yes or no (If “no” is selected the fields Proxy,Port and proxycred are ignored)
Proxy	Optional	Proxy Server IP or hostname (only used in case Use Proxy = yes)
Proxycred	Optional	Proxy Server Credentials (only used in case Use Proxy = “yes” is selected)
Port	Optional	Proxy Server Port (only used in case Use Proxy = yes)

Field	Required	Description
Loglevel	Mandatory	logging settings DEBUG, INFO, WARNING, ERROR, CRITICAL
Container	Mandatory	Container to be created. If the container already exists, the task will fail.

6.2 *Monitor_blob_in_container*

Command	UT Name	Description
Monitor_blob_in_container	ut_AZURE_BLOBSTORAGE_Monitor_blob_in_container_linux	Monitors at a given interval for a blob in a container

Task Screenshot:



Field Description:

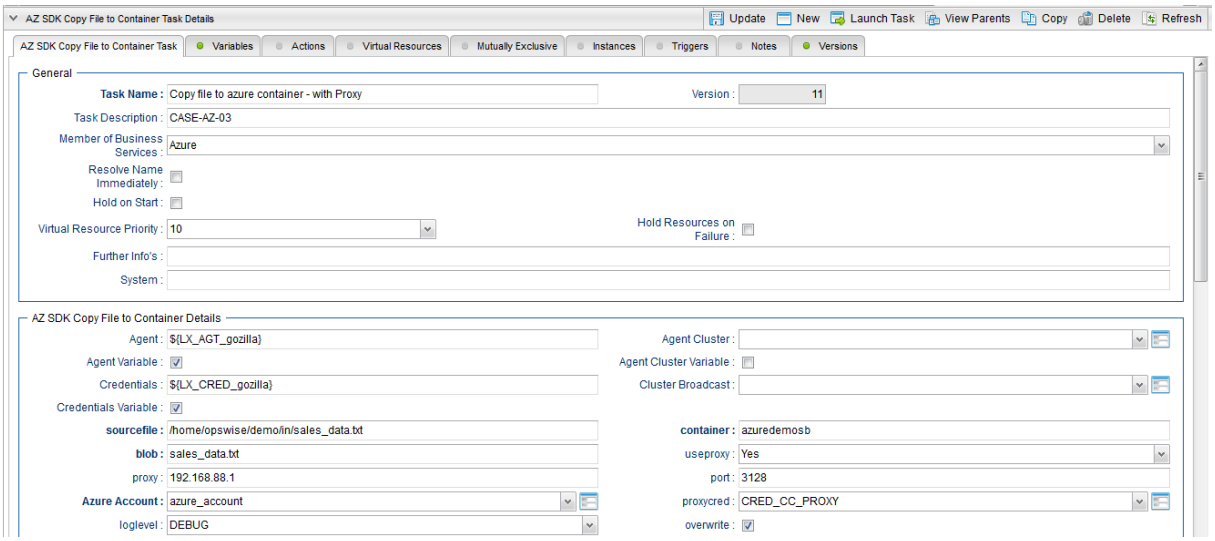
Field	Required	Description
Agent	Mandatory	The Linux Universal Agent, which runs the Python azure-storage-blob module to call the AZURE BLOB STORAGE commands
Credentials	Optional	The Credentials used on the Linux Server
Use Proxy (yes/no)	Mandatory	Decide If a Proxy Server should be used yes or no (If “no” is selected the fields Proxy,Port and proxycred are ignored)
Proxy	Optional	Proxy Server IP or hostname (only used in case Use Proxy = yes)

Field	Required	Description
Proxycred	Optional	Proxy Server Credentials (only used in case Use Proxy = “yes” is selected)
Port	Optional	Proxy Server Port (only used in case Use Proxy = yes)
Loglevel	Mandatory	logging settings DEBUG, INFO, WARNING, ERROR, CRITICAL
File to scan	Mandatory	AZURE blob (=file) to scan in the given container
Container name	Mandatory	AZURE Container to scan for the given AZURE blob (=file)
Interval	Mandatory	Scan Interval

6.3 Copy_file_to_container

Command	UT Name	Description
Copy_file_to_container	ut_AZURE_BLOBSTORAGE_Copy_file_to_container_linux	Copies a file to a container

Task Screenshot:



Field Description:

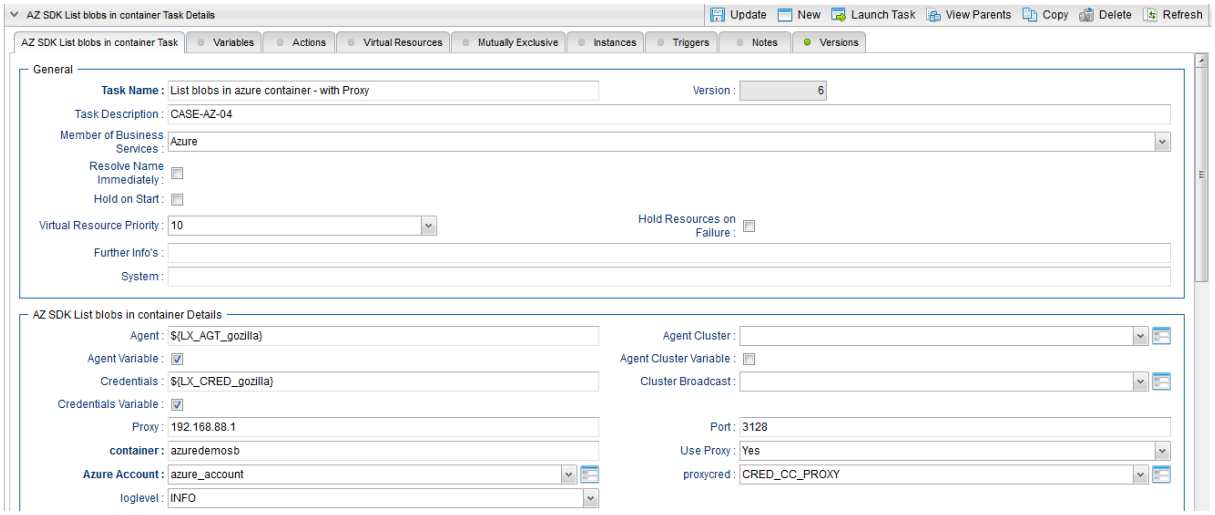
Field	Required	Description
Agent	Mandatory	The Linux Universal Agent, which runs the Python azure-storage-blob module to call the AZURE BLOB STORAGE commands
Credentials	Optional	The Credentials used on the Linux Server

Field	Required	Description
Use Proxy (yes/no)	Mandatory	Decide If a Proxy Server should be used yes or no (If “no” is selected the fields Proxy,Port and proxycred are ignored)
Proxy	Optional	Proxy Server IP or hostname (only used in case Use Proxy = yes)
Proxycred	Optional	Proxy Server Credentials (only used in case Use Proxy = “yes” is selected)
Port	Optional	Proxy Server Port (only used in case Use Proxy = yes)
Loglevel	Mandatory	logging settings DEBUG, INFO, WARNING, ERROR, CRITICAL
Container Name	Optional	Name of the target container
Overwrite	Mandatory	If checked, allow to overwrite an existing blob in the given container
Blob	Mandatory	blob to copy to the given container

6.4 List_blobs_in_container

Command	UT Name	Description
List_blobs_in_container	ut_AZURE_BLOBSTORAGE_List_blobs_in_container_linux	Lists all blobs in a container

Task Screenshot:



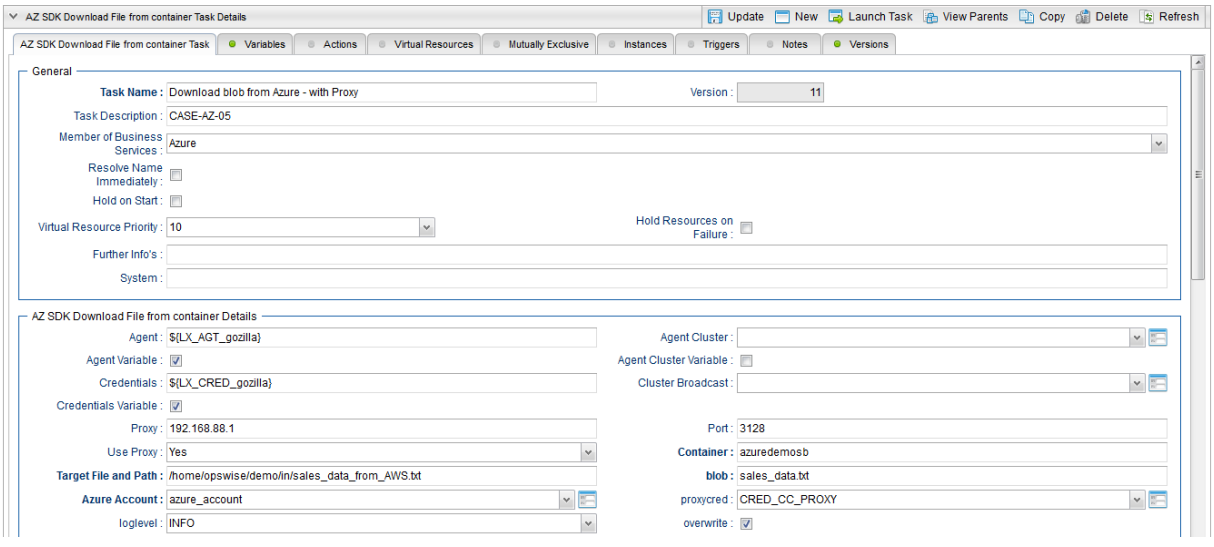
Field Description:

Field	Required	Description
Agent	Mandatory	The Linux Universal Agent, which runs the Python azure-storage-blob module to call the AZURE BLOB STORAGE commands
Credentials	Optional	The Credentials used on the Linux Server
Use Proxy (yes/no)	Mandatory	Decide If a Proxy Server should be used yes or no (If “no” is selected the fields Proxy, Port and proxycred are ignored)
Proxy	Optional	Proxy Server IP or hostname (only used in case Use Proxy = yes)
Proxycred	Optional	Proxy Server Credentials (only used in case Use Proxy = “yes” is selected)
Port	Optional	Proxy Server Port (only used in case Use Proxy = yes)
Loglevel	Mandatory	logging settings DEBUG, INFO, WARNING, ERROR, CRITICAL
Container	Mandatory	List all blobs in the container

6.5 Download_file_from_container

Command	UT Name	Description
Download_file_from_container	ut_AZURE_BLOBSTORAGE_Download_file_from_container_linux	Downloads an AZURE BLOB STORAGE blob to a local file

Task Screenshot:



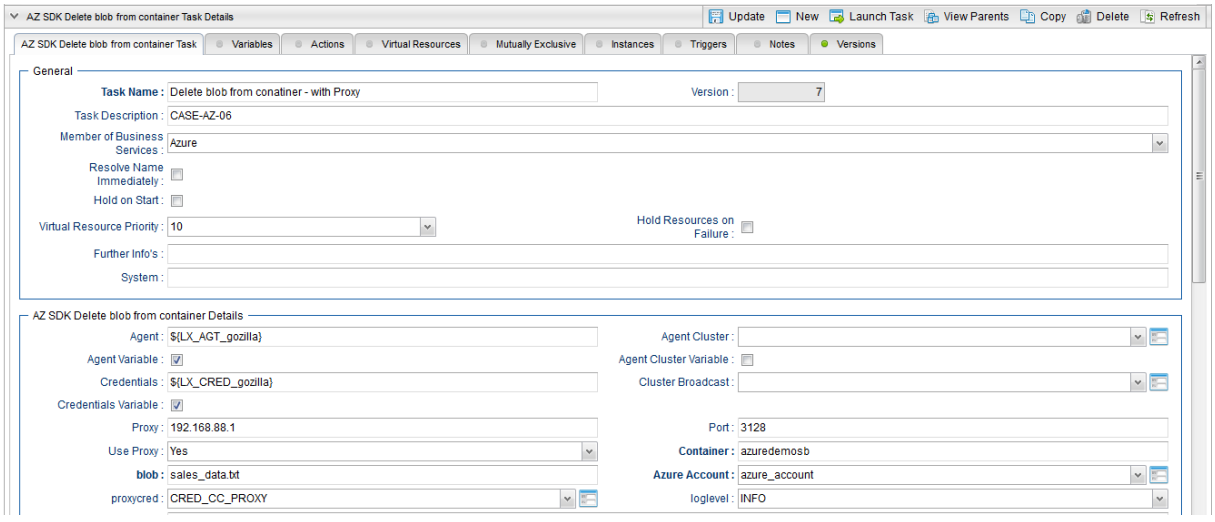
Field Description:

Field	Required	Description
Agent	Mandatory	The Linux Universal Agent, which runs the Python azure-storage-blob module to call the AZURE BLOB STORAGE commands
Credentials	Optional	The Credentials used on the Linux Server
Use Proxy (yes/no)	Mandatory	Decide If a Proxy Server should be used yes or no (If “no” is selected the fields Proxy,Port and proxycred are ignored)
Proxy	Optional	Proxy Server IP or hostname (only used in case Use Proxy = yes)
Proxycred	Optional	Proxy Server Credentials (only used in case Use Proxy = “yes” is selected)
Port	Optional	Proxy Server Port (only used in case Use Proxy = yes)
Loglevel	Mandatory	logging settings DEBUG, INFO, WARNING, ERROR, CRITICAL
Target File	Mandatory	Target file name and directory e.g. */home/opswise/demo/in/sales_data_from_Azure.txt*
Blob	Mandatory	Source Blob file to download
Container	Mandatory	Source container

6.6 Delete_blob_from_container

Command	UT Name	Description
Delete_blob_from_container	ut_AZURE_BLOBSTORAGE_Delete_blob_from_container_linux	Deletes a blob from a container

Task Screenshot:



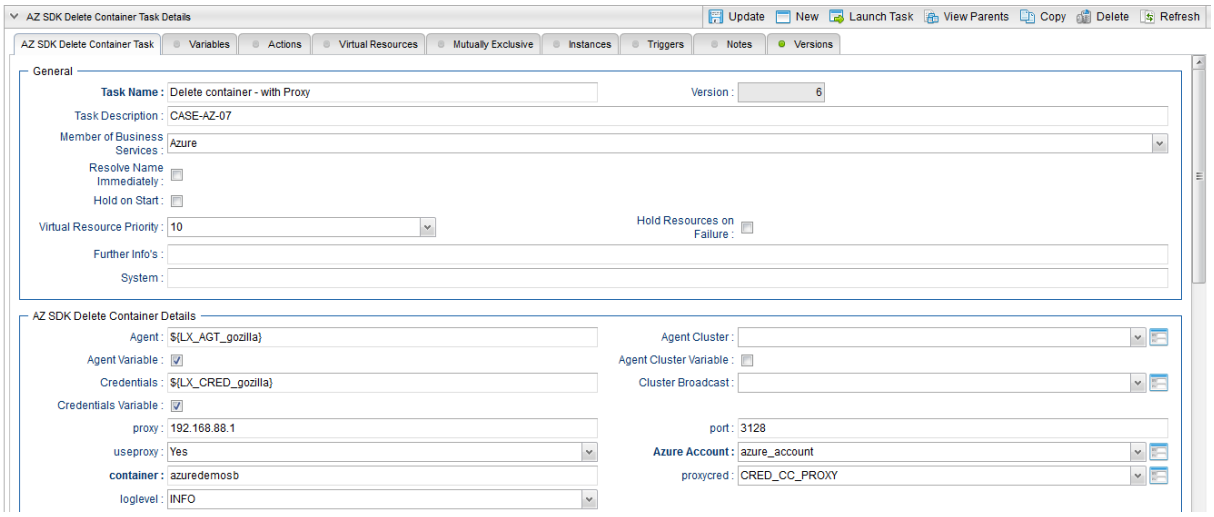
Field Description:

Field	Required	Description
Agent	Mandatory	The Linux Universal Agent, which runs the Python azure-storage-blob module to call the AZURE BLOB STORAGE commands
Credentials	Optional	The Credentials used on the Linux Server
Use Proxy (yes/no)	Mandatory	Decide If a Proxy Server should be used yes or no (If “no” is selected the fields Proxy,Port and proxycred are ignored)
Proxy	Optional	Proxy Server IP/ hostname (used if Use Proxy = “yes”)
Proxycred	Optional	Proxy Server Credentials (used if Use Proxy = “yes”)
Port	Optional	Proxy Server Port (only used in case Use Proxy = yes)
Loglevel	Mandatory	logging settings DEBUG, INFO, WARNING, ERROR, CRITICAL
Blob	Mandatory	Blob to delete from the given container
Container	Mandatory	Container, which contains the blob to delete

6.7 Delete_empty_container

Command	UT Name	Description
Delete_empty_container	ut_AZURE_BLOBSTORAGE_Delete_empty_container_linux	Deletes an empty container

Task Screenshot:



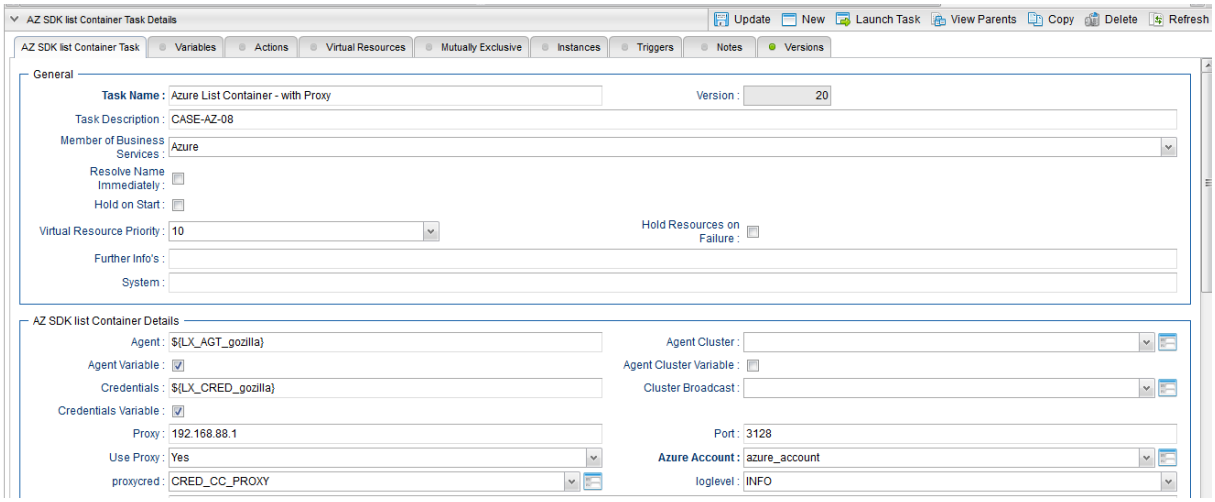
Field Description:

Field	Required	Description
Agent	Mandatory	The Linux Universal Agent, which runs the Python azure-storage-blob module to call the AZURE BLOB STORAGE commands
Credentials	Optional	The Credentials used on the Linux Server
Use Proxy (yes/no)	Mandatory	Decide If a Proxy Server should be used yes or no (If “no” is selected the fields Proxy,Port and proxycred are ignored)
Proxy	Optional	Proxy Server IP/ hostname (used if Use Proxy = “yes”)
Proxycred	Optional	Proxy Server Credentials (used if Use Proxy = “yes”)
Port	Optional	Proxy Server Port (only used in case Use Proxy = yes)
Loglevel	Mandatory	logging settings DEBUG, INFO, WARNING, ERROR, CRITICAL
Container	Mandatory	Container to delete

6.8 List_containers

Command	UT Name	Description
List_containers	ut_AZURE_BLOBSTORAGE_List_containers_linux	List all containers of an Azure account

Task Screenshot:



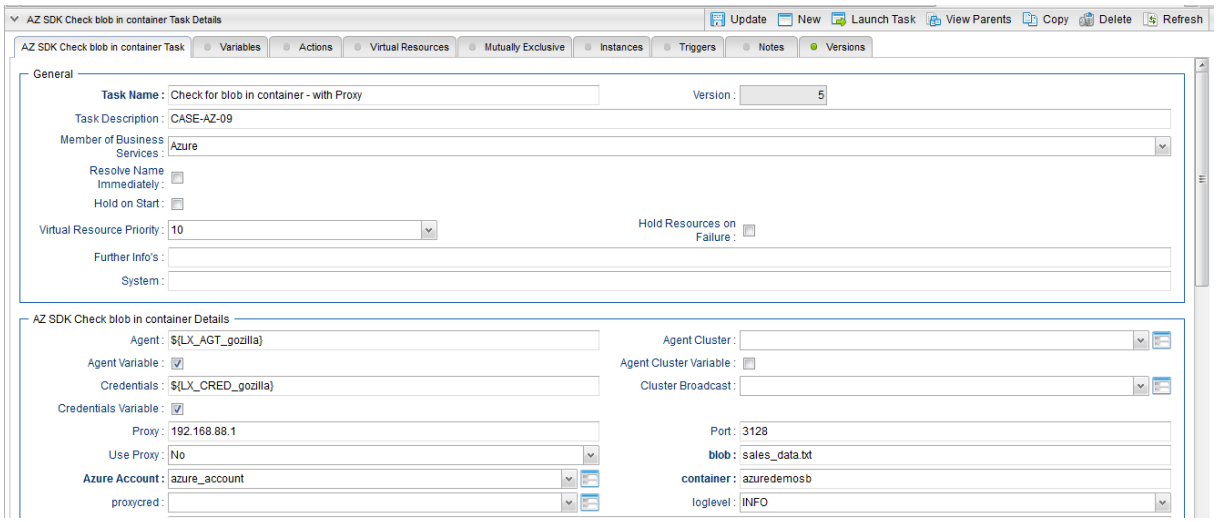
Field Description:

Field	Required	Description
Agent	Mandatory	The Linux Universal Agent, which runs the Python azure-storage-blob module to call the AZURE BLOB STORAGE commands
Credentials	Optional	The Credentials used on the Linux Server
Proxy	Optional	Proxy Server IP/ hostname (used if Use Proxy = “yes”)
Proxycred	Optional	Proxy Server Credentials (used if Use Proxy = “yes”)
Port	Optional	Proxy Server Port (only used in case Use Proxy = yes)
Port	Optional	Proxy Server Port (only used in case Use Proxy = yes)
Loglevel	Mandatory	logging settings DEBUG, INFO, WARNING, ERROR, CRITICAL

6.9 Check_for_blob_in_containers

Command	UT Name	Description
Check_for_blob_in_containers	ut_AZURE_BLOBSTORAGE_Check_for_blob_in_containers_linux	Checks for the existence of a blob in a container

Task Screenshot:



Field Description:

Field	Required	Description
Agent	Mandatory	The Linux Universal Agent, which runs the Python azure-storage-blob module to call the AZURE BLOB STORAGE commands
Credentials	Optional	The Credentials used on the Linux Server
Use Proxy (yes/no)	Mandatory	Decide If a Proxy Server should be used yes or no (If “no” Proxy, Port and proxycred are ignored)
Proxy	Optional	Proxy Server IP/ hostname (used if Use Proxy = “yes”)
Proxycred	Optional	Proxy Server Credentials (used if Use Proxy = “yes”)
Port	Optional	Proxy Server Port (only used in case Use Proxy = yes)
Loglevel	Mandatory	logging settings DEBUG, INFO, WARNING, ERROR, CRITICAL
Blob	Mandatory	Blob to check for in the given container
containername	Mandatory	Container to check for the given blob

7 Test Cases

The following basic test cases has been performed:

Case#	Assumed behavior	Result
Create a container in AZURE BLOB STORAGE	Log message: INFO - Container azuredemosb created	Correct
Creates a container in AZURE BLOB STORAGE (container already exists)	Log message: ERROR - Container may already exists, new Container azuredemosb not created	Correct
Monitors at a given interval (10s) for a blob in a container	Log message: INFO - Blob: sales_data.txt found in container: azuredemosb	Correct
Copies a file to a container, if the blob exists (flag overwrite is set)	Log message: INFO - Overwrite flag set, start Uploading file: /home/opswise/demo/in/sales_data.txt ,to container: azuredemosb as blob: sales_data.txt INFO - Finished Uploading file: /home/opswise/demo/in/sales_data.txt ,to container: azuredemosb as blob: sales_data.txt	Correct
Copies a file to a container, if the blob exists (flag overwrite is not set)	Log message: ERROR - Blob: sales_data.txt exists and will not be overwritten, set overwrite flag	Correct
Copies a file to a container (file does not exist in container)	Log message INFO - Start Uploading file: /home/opswise/demo/in/sales_data.txt ,to container: azuredemosb as blob: sales_data.txt INFO - Starting new HTTPS connection (1): stonebranchpoc.blob.core.windows.net INFO - Finished Uploading file: /home/opswise/demo/in/sales_data.txt ,to container: azuredemosb as blob: sales_data.txt	Correct
Copies a file to a container (container does not exist)	azure.common.AzureMissingResourceHttpError: The specified container does not exist.	Correct
Lists all blobs in a container	Blobs are listed in stdout: List of all blobs in the container is displayed sales_data.txt	Correct

Downloads an AZURE BLOB STORAGE blob to a local file if file does not yet exist	Log message: INFO - Start downloading file: sales_data.txt ,from container: azuredemosb to : /home/opswise/demo/in/sales_data_from_AWS.txt INFO - Starting new HTTPS connection (1): stonebranchpoc.blob.core.windows.net INFO - Finished downloading file: sales_data.txt ,from container: azuredemosb to : /home/opswise/demo/in/sales_data_from_AWS.txt	Correct
Downloads an AZURE BLOB STORAGE blob to a local file if already exist (flag overwrite is set)	Log message: INFO - File: /home/opswise/demo/in/sales_data_from_AWS.txt exists and will be overwritten due to overwrite flag set INFO - Start downloading file: sales_data.txt ,from container: azuredemosb to : /home/opswise/demo/in/sales_data_from_AWS.txt INFO - Finished downloading file: sales_data.txt ,from container: azuredemosb to : /home/opswise/demo/in/sales_data_from_AWS.txt	Correct
Downloads an AZURE BLOB STORAGE blob to a local file if already exist (flag overwrite is not set)	Log message: ERROR - File: /home/opswise/demo/in/sales_data_from_AWS.txt exists and will not be overwritten, set overwrite flag	Correct
Downloads an AZURE BLOB STORAGE blob to a local file if file does not yet exist and overwrite flag is set.	Log message: INFO - Start downloading file: sales_data.txt ,from container: azuredemosb to : /home/opswise/demo/in/sales_data_from_AWS.txt INFO - Finished downloading file: sales_data.txt ,from container: azuredemosb to : /home/opswise/demo/in/sales_data_from_AWS.txt	Correct
Deletes a blob from a container	Log message: INFO - blob sales_data.txt deleted sucessfuly	Correct
Deletes a blob from a container (blob does not exist)	Log message: ERROR - blob sales_data.txt unable to delete, blob may not exist	
Deletes an empty container Force flag set	Log message: INFO - container azuredemosb deleted sucessfuly	Correct
Deletes an empty container Force flag not set	Log message: INFO - container azuredemosb deleted sucessfuly	Correct

Deletes a not empty container, force flag not set	Log message: ERROR - Blob exists in container azuredemosb, container will not be deleted, set force flag	Correct
Deletes a not empty container, force flag set	Log message: INFO - Force flag set, container: azuredemosb ,with all blob will be deleted INFO - Container: azuredemosb ,with all blob has been deleted - It may take some time until reuse of container is possible	Correct
Deletes a container, which does not exist	Log message: The specified container does not exist.	Correct
List all containers of an AZURE account	Log message: List of containers is displayed azuredemosb	Correct
Checks for the existence of a blob in a container	Log message: INFO - Blob: sales_data.txt found in container: azuredemosb	Correct
Checks for the existence of a blob in a container, if blob does not exist	Log message: ERROR - Blob: sales_data.txt not found in container: azuredemosb	Correct

8 Document References

There are no document references.