SHA256 checksums for checking uploads

Bug Reference

CLOUDSTACK-10035 - Jira project doesn't exist or you don't have permission to view it.

https://issues.apache.org/jira/browse/CLOUDSTACK-10035

Introduction

Purpose

the obsoletion of the md5 checksum algorithm requires that CloudStack checks downloaded templates by means of an alternative algorithm to the current one. The algorithm type should be configurable and include both the obfuscated MD5 and, as an example SHA256.

Requirement

A user must be able to enter the algorithm to use. The available algorithms must be configured through automatic discovery by the spring framework. A certain check algorithm will only work if the algorithm code is available in the management server and in the SSVM. In the present specs of java all implementations must provide at least {MD5, SHA-1, SHA-256}.

Use

The parameter containing the checksum will be considered a MD5sum if it is not prefixed and just a plain ascii/utf8 representation of a hexadecimal string. If it is required to use another algorithm the hexadecimal string is to be prefixed with a string of the form, "{<algorithm>}", not including the double quotes. In this <algorithm> is the exact string representing the java supported algorithm, i.e. MD5 or SHA-256. Note that java does not contain an algorithm called SHA256 or one called sha-256, only SHA-256.

UI

At the moment only the Volume upload Form has a field for entering a checksum value. This field will be abused by letting the user enter a checksum-type prefix to the checksum value.

The register template and - ISO dialogs do not have such a field, in spite of the corresponding API calls having the same parameter. Consideration must be put into whether we want to burdon the user with the complexity of retrieving a checksum and deciding on the algorithm for these types. Altering the UI for those upload types is out of scope for this enhancement.

API

the API calls based on AbstractGetUploadParamsCmd, must be able to accept the algorithm to use for checking and/or be able to parse the checksum parameter and extract the checksum algorithm from it. The default algorithm if none specified will remain the present md5.