Project Acronym:	**OPTIMIS**
Project Title:	**Optimized Infrastructure Services**
Project Number:	**257115**
Instrument:	**Integrated Project**
Thematic Priority:	**ICT-2009.1.2 – Internet of Services, Software and Virtualisation**

Data Management Installation Guide

Activity 5: *Federated Service Operation*

WP 5.2: *Data Management*

Due Date:		M36
Submission Date:		07/06/2013
Start Date of Project:		01/06/2010
Duration of Project:		36 months
Organization Responsible for the Deliverable:		ICCS/NTUA
Version:		1.0
Status		Final
Author(s):	George Vafiadis George Kousiouris Srijith Nair	ICCS/NTUA ICCS/NTUA BT
Reviewer(s)	Jordi Guitard Anthony Sulistio	BSC HLRS

Project co-funded by the European Commission within the Seventh Framework Programme		
Dissemination Level		
PU	Public	**X**
PP	Restricted to other programme participants (including the Commission)	
RE	Restricted to a group specified by the consortium (including the Commission)	
CO	Confidential, only for members of the consortium (including the Commission)	

Version History

Version	Date	Comments, Changes, Status	Authors, contributors, reviewers
0.1	15/2/13	Initial Version based on Y2 document	George Vafiadis (ICCS/NTUA), George Kousiouris (ICCS/NTUA) Srijth Nair (BT)
0.2	28/2/2013	Incorporated new chapters structure and initial content with regard to new advancements	George Vafiadis (ICCS/NTUA), George Kousiouris (ICCS/NTUA)
0.3	15/3/2013	Incorporated DM new features and installation instructions, FCS core scripts installation process, FCS server installation process and CA signing process	George Vafiadis, George Kousiouris (ICCS/NTUA)
0.5	22/3/2013	Document sent for internal QA	George Kousiouris (ICCS/NTUA)
0.6	25/3/2013	Received document and comments	Pavlos Kranas, A. Menychtas (ICCS/NTUA)
0.7	28/3/2013	Comments addressed	George Kousiouris (ICCS/NTUA)
0.8	31/3/2013	Document ready for submission	George Kousiouris (ICCS/NTUA)
0.9	31/5/2013	Added Amazon interoperability installation instructions, Amazon related instructions for FCS installation and ELB activation	George Vafiadis (ICCS/NTUA), George Kousiouris (ICCS/NTUA)
1	07/0672013	Final version	Malena Donato (ATOS)

Table of Contents

Index of Tables

1 Introduction

This document introduces the installation guide for the Federated Data Management Mechanisms Prototype for WP5.2 of the OPTIMIS project and includes also the contents of D6.5.1 regarding interoperability of the OPTIMIS DM with Amazon AWS.

The Data Management infrastructure is a distributed fault tolerant file system running in a number of Virtual Machine instances as a service. The file-system is based on Hadoop/HDFS. The data storage infrastructure has a Virtual machine instance called Master Node and a number of VM instances used as Data Nodes, for each different service.

The Federated Data Management Mechanisms Prototype includes the following revised components:

- Data Manager that is comprised from:
  - DFS MasterNode (single VM instance)
  - DFS DataNode (multiple VM instances)
  - RESTful Web Data Manager Gateway
  - SSH-HDFS Bridge
  - Check Legal Component
  - Data Manager client
- Federation Candidate Selector component that is comprised of:
  - FCS server implementation component
  - FCS core scripts for model creation and query based on Matlab
- DM AWS plugin
- Secure Storage Tool component

1.1 Purpose

The purpose of this document is:

- to provide information on how to get started (software dependencies, hardware requirements and platforms supported, installation instructions)

1.2 Glossary of Acronyms

Table 1 - Acronyms table

Acronym	Definition
AES	Advanced Encryption Standard
API	Application Programming Interface
AWS	Amazon Web Services
CA	Certificate Authority
D	Deliverable
DFS	Distributed File System
DPA	Data Protection Authority
ELB	Elastic Load Balancer

FCS	Federation Candidate Selector
HDFS	Hadoop Distributed File System
IP	Infrastructure Provider
REST	Representational State Transfer
SaaS	Software as a Service
SLA	Service Level Agreement
SP	Service Provider
VM	Virtual Machine

2 DFS Data Manager

2.1 Release Information

Table 2: DM Release Information

Component Name	Release Number	Release Date
DataManager Client	3	31/03/2013
Datamanager Web Services	3	31/03/2013
Optimis Distributed File System	3	31/03/2013
Optimis Cloud Provider Description Schema	0.2	31/03/2013

2.2 Minimal System Requirements

The distributed file system is running in VM instances. The hardware requirements for the master node are based on the number of supported data nodes. The higher the number of nodes the higher demand for RAM in the master node. For production use, it is recommended a dual core CPU, RAM 4-8 GB, 30GB-1TB disk.

2.3 Platforms Supported

Most of the software is written in java with some components in C++ and ruby. The current implementation uses Linux (Ubuntu 10.0). The distributed file-system component has not been well tested on Win32, so this is not a production platform.

2.4 Software Dependencies

Table 3 DM Software dependencies

Product	Version	Licence
Java	1.6	GPL
Jersey-server	1.6	GPL
Jersey-core	1.6	GPL
Hadoop	0.20	Apache License version 2
OpenSSH	5.5p1	GPL
OpenSSL	0.9.8o	GPL
RSYNC	3.0.7	GPL
Apache tomcat	6.0	Apache License version 2

2.5 Installation Instructions

The component is distributed as a single Virtual Machine Image. All dependencies are part of the VM. In this section we describe how to create this VM image.

The first thing we need to perform is to create an appropriate virtual machine image for XEN. We need also a XEN compatible kernel.

We download from http://www.ubuntu.com/download/ubuntu/download the iso image of ubuntu 64bit server.

After following the usual install procedure we need to install some additional packages and perform some configurations.

Install ssh

```
$  install ssh openssh-server
```

Install Networking (example)

```
$ vi /etc/network/interfaces

and set a static IP (the configuration below is indicative from the internal OPTIMIS test-bed)

auto eth0


iface eth0 inet static


address    192.168.252.59


gateway    192.168.252.62


netmask    255.255.255.192


network    192.168.252.0


broadcast  192.168.252.63
```

Then restart your network:

```
$ /etc/init.d/networking restart
```

Then edit */etc/hosts*. Make it look like this:

```
192.168.252.59

 master
```

```
127.0.0.1          master
 localhost


::1     localhost ip6-localhost ip6-loopback


fe00::0 ip6-localnet


ff00::0 ip6-mcastprefix


ff02::1 ip6-allnodes


ff02::2 ip6-allrouters
```

Disable AppArmor

```
$ /etc/init.d/apparmor stop
$ update-rc.d -f apparmor remove
$ aptitude remove apparmor apparmor-utils
```

Install other dependencies

```
$ apt-get install ruby

$ sudo aptitude install build-essential libcurl4-openssl-dev libxml2-dev libfuse-dev comerr-dev libfuse2 libidn11-dev
libkadm55 libkrb5-dev libldap2-dev libselinux1-dev libsepol1-dev pkg-config fuse-utils sshfs
```

Install Sun Java 6

```
$ sudo add-apt-repository "deb http://archive.canonical.com/ maverick partner"

$ sudo apt-get update

$ sudo apt-get install sun-java6-jdk

Select Sun's Java as the default on your machine:

$ sudo update-java-alternatives -s java-6-sun
```

Install Hdfs

```
Add the repository by creating a new file/etc/apt/sources.list.d/cloudera.list with the following contents.

Release = lsb_release -c

deb http://archive.cloudera.com/debian<Release>-cdh3 contrib
deb-src http://archive.cloudera.com/debian<release>-cdh3 contrib
```

Add repository key

```
$ curl -s http://archive.cloudera.com/debian/archive.key | sudo apt-key add -
```

Install Hadoop

```
$ sudo apt-get update
$ apt-cache search hadoop
$ sudo apt-get install hadoop-0.20
$ sudo apt-get install hadoop-0.20-namenode
$ sudo apt-get install hadoop-0.20-datanode
$ sudo apt-get install hadoop-0.20-jobtracker
$ sudo apt-get install hadoop-0.20-tasktracker
```

Then patch hadoop installation with “optimis-datamanager-patch.rb” script.

Install Tomcat 6.0

```
$ apt-get install tomcat6
```

The web application exposes a RESTful API. The Datamanager.war should be installed in Tomcat's application directory.

Install maven

```
$ apt-get install maven2
```

Create SSH key for accusing the data nodes:

```
$ ssh-keygen -t rsa -P ''
```

In the data nodes:

- we install ssh, hadoop-name-node, ruby
- and we run the optimis-datanode-client-install.rb <masterNodeIP> in each datanode VM.

The data node vms must be all up and running before starting masternode.

After starting the distributed file-system for the first time we have to format it

```
$ hadoop namenode –format
```

2.5.1 Data Storage SP Image Repository

The SP image repository is used to store the vm images during image creation service.

```
yum -i zip wget zip
wget http://130.239.48.114/datamanager/optimis-dm-mount-spimages-current.zip
unzip optimis-dm-mount-spimages-current.zip
chmod +x install-optimis-sp-datamanager.sh
./install-optimis-sp-datamanager.sh
```

2.5.2 Data Storage VM Image Repository

The VM repository stores all virtual machine images for each service.

```
 wget http://130.239.48.114/datamanager/optimis-dm-mount-vmimages-current.zip
 unzip optimis-dm-mount-vmimages-current.zip
 chmod +x install-optimis-datamanager.sh
 ./install-optimis-datamanager.sh <ip-provider>
<ip-provider> := atos | flex | umea


script:
#!/bin/bash
IProvider=$1
if [ "" == "$IProvider" ]; then
 echo "Error: You must specify the infrastructure provider name located in conf folder"
 exit
fi
confFile=./conf/$IProvider.conf
if [ !  -f $confFile ]; then
 echo "Error: Infrastructure provider is not valid: $confFile"
 exit
fi
cp optimis-mount  /usr/bin/
cp optimis-umount /usr/bin
cp optimis-datamanager /etc/init.d/
chmod +x /etc/init.d/optimis-datamanager
echo "Copying key for infrastructure provider $IProvider"
cp $confFile /etc/datamanager.conf
if [ -f /etc/debian_version ]; then
   OS=Debian
```

```
    apt-get install sshfs
    apt-get install ruby
    update-rc.d optimis-datamanager defaults
elif [ -f /etc/redhat-release ]; then
    OS=Centos
    wget http://packages.sw.be/rpmforge-release/rpmforge-release-0.5.2-2.el5.rf.`uname -i `.rpm
    rpm --import http://apt.sw.be/RPM-GPG-KEY.dag.txt
    rpm -K rpmforge-release-0.5.2-2.el5.rf.*.rpm
    rpm -i rpmforge-release-0.5.2-2.el5.rf.*.rpm
    yum search sshfs
    yum install fuse-sshfs.`uname -i`
    yum install ruby rubygems
    rm rpmforge-release-0.5.2-2.el5.rf.*.rpm
    chkconfig --add optimis-datamanager
    chkconfig --level 345 optimis-datamanager on
fi
```

2.5.3 DataManager main coordinator

This script installs all dependencies and all components in datamanager main coordinator VM.

```
unzip optimis-datamanager-coordinator.zip
./usr/optimis/optimis-datamanager-all.sh
```

2.5.4 DataManager FUSE Client Installation

These scripts perform mounting of the distributed storage system.

```
uname -i
wget http://packages.sw.be/rpmforge-release/rpmforge-release-0.5.2-2.el5.rf.`uname -i `.rpm
rpm --import http://apt.sw.be/RPM-GPG-KEY.dag.txt
rpm -K rpmforge-release-0.5.2-2.el5.rf.*.rpm
```

```
rpm -i rpmforge-release-0.5.2-2.el5.rf.*.rpm
yum search sshfs
yum install fuse-sshfs.`uname -i`
yum install ruby
rm /usr/bin/optimis-mount
rm /usr/bin/optimis-umount
ln -s /context/optimis-mount /usr/bin/optimis-mount
ln -s /context/optimis-umount /usr/bin/optimis-umount
chmod +x /usr/bin/optimis-mount
chmod +x /usr/bin/optimis-umount
```

2.5.5 Cloud Provider Description Schema

The Cloud Provider description schema does not need any installation other than an import to the Eclipse Environment (or any suitable tool with XML editing and validating capabilities). For doing so create a new project:

File-> New->Project-> General Project

and copy the XSD file in the root directory. Then utilize the instructions on the DM User Guide for creating an instance based on the schema.

2.5.6 CA Authority signing of the CPDS instances

Following the creation of the instance of the CPDS, the OPTIMIS process dictates that this should be signed by a CA (in essence the Data Protection Authority of a country). To offer this functionality we provide the dm-legal.zip file. This does not include any specific installation process since it is distributed as a normal zip file for linux environments. The only configuration aspects that need attention are the following:

- The scripts are run from the <unzip location>
- The directory structure under the <unzip location> is the following
  - ./ca
    - Certificates, public and private keys
  - ./cpds
    - Unsigned CPDS instances: a new instance should be placed in this folder
  - ./signed_cpds
    - Folder in which signed CPDS instances are kept

2.6 Getting started

2.6.1 Using the Software

See DM User Guide document

3 FCS Core Scripts for model creation and query

3.1 Release information

Table 4: FCS core scripts release information

Component Name	Release Number	Release Date
FCS core scripts	1.0	31/3/2013

3.2 Minimal System Requirements

The scripts run satisfactorily on a Core 2 Duo Intel Processor with 2 GB of RAM. OS is Windows but can be changed to Linux if the paths inside the scripts are altered to the Unix directory notation. The time to finish is arbitrary and depends on a variety of parameters, such as the dataset size, the cores available to Matlab, the GA iterations etc. It should be expected that this duration is in the range of hours (e.g. with a 2800-value data set and 50 GA generations the wall clock time was about three hours of execution for the model creation and optimization).

3.3 Platforms Supported

The software has been tested in the Windows XP platform. However, the scripts are Matlab-based, which means that they can be executed in whatever platform is supported by Matlab. The only dependency is the path names and annotations that may change from platform to platform and are highlighted in the installation section.

3.4 Software Pre-requisites and Dependencies

Table 5 Software dependencies

Product	Version	Licence
Matlab	R2007b	Proprietary
Matlab GA toolkit	R2007b	Proprietary
Matlab NN toolkit	R2007b	Proprietary

3.5 Installation Instructions

The zip file is comprised of the following files

- gatime: script for launching the model creation, handles the Genetic Algorithm launching and ANN optimization

- nncreator: internal script for creating, training and validating each ANN
- BestMean.mat, indicator.mat, generation.mat : helper files for keeping temporary information regarding best performance so far in the GA execution, generation count. Auto-initialized each time the GA launches
- Inputdata.mat: file containing the initial time series dataset in a column matlab format
- preprocess.m: script for transforming the inputdata series to a normalized series in the interval [-1,1], for usage in the ANN
- crossvalidateMulti.m : script for validating the model output for multiple steps ahead.

The following configuration actions may be applied to the setup of the files"

- <Path2>= base path to folder where models produced during gatime are stored
- <Path>= base path+ folder for a specific GA setup run

Actions to install and execute:

- Download the zip file and extract all the contents on the same folder
- Install Matlab and point the working folder to your zip extraction folder
- Create the folder that is specified in the <path2> variable in your system
- Scripts preprocess, gatime and crossvalidateMulti can now be run from inside the Matlab environment
- Run preprocess from inside the Matlab environment
- Run gatime from inside the Matlab environment
- Run crossvalidateMulti from inside the Matlab environment
- Go to folder <Path2> to check the results (models and stats created)

IMPORTANT NOTICE: If you execute the gatime for the same GA configuration, the results will overwrite the previous identical run, since the path depends on the base path plus the GA properties (generations, elite count etc.).

3.6 Getting started

3.6.1 Using the Software

For this information please consult the usage guide.

4 FCS Server Installation

4.1 Release information

Table 6: FCS server release information

Component Name	Release Number	Release Date
FCS_server	1.0	31-3-2013

4.2 Minimal System Requirements

The FCS server is a war file that can be deployed in any Linux environment and relevant container. It has been tested on an Intel Core 2 Duo CPU (2.93Ghz) with 4 GB of RAM, running Ubuntu 10.04 LTS, in an Apache Tomcat 6 container.

4.3 Software Pre-requisites and Dependencies

Table 7 FCS Software dependencies

Product	Version	Licence
Matlab executable scripts and model	R2007b/R2011	Distributed by OPTIMIS DM as Apache License version 2
Java	1.6	
Matlab Compiler Runtime		Free
Apache tomcat	6.0	Apache License version 2
OPTIMIS DM Logging subsystem and client	Y3 Release	Apache License version 2

The licensing issue with Matlab has two parts. Initially Matlab is available under a proprietary license. However if an entity (e.g. Model Provider) acquires a Matlab license, then it can compile and distribute (as an executable) the prediction model (specific requirements for toolkits depend on model technology implementation) to potential clients. In order for the executable to be actually executed on the client's machine, the client can install the Matlab Compiler Runtime (MCR) without a license and run the executable sent by the Model Provider.

4.4 Installation Instructions

The FCS models are created offline by the scripts provided in Chapter 4. For the online deployment and usage (query) of them through the FCS server the following steps need to be applied:

1. Install Matlab Compiler Runtime for MatlabR2007b (instructions in [10])
2. Install Tomcat and deploy the FCS server war (instructions in [11])
    a. One point that needs attention is the fact that when tomcat is deployed through the apt-get method, it is launched by a tomcat6 user, that does not have the necessary privileges in the MCR folders etc. Thus it is highly recommended that the following process is followed:

Download tomcat 6 directly from available links, e.g.:

wget http://apache.tsl.gr/tomcat/tomcat-6/v6.0.36/bin/apache-tomcat-6.0.36.zip

Then create a folder in the home directory of the user under which it should be run and unzip the file. Move the FCS war file to the webapps subdir and start tomcat through the script provided in the bin directory of the unzipped folder.

For automatic start, change the paths to the ones in which you have installed tomcat (up to the /bin directory), following instructions in [12].

3. Configuration details for the necessary paths
    a. The path to the model XML descriptors must be created as /usr/share/eclipse/data
        i. This is the location where the server will find the past values that are needed for the model prediction
        ii. The XML descriptors are simple XML structures that are created manually for each model by the model developers
        iii. This path can be altered in the source code (in both methods GiveModelInfo and GetPrediction of the FCS_Server class)
    b. The path to the models and the Matlab executables must be created as /usr/share/eclipse/models
        i. This is the location where the server will search for the models (in the naming format of net<MODELID>.mat)
        ii. This is the location where the server will search for the Matlab executable file (run_givePrediction.sh) and libraries that are called for acquiring the estimation
        iii. This path can be changed in the EstimThread class provided with the FCS server
    c. The path to the Matlab Compiler Runtime installation folder must be set as /usr/local/MATLAB/R2011b/ or alternatively it can be changed to whatever path in the EstimThread class.

4.5 Installation Instructions for Amazon VM creation

Specific details that should be followed for Amazon VM creation:

A dynamic IP is assigned to the VMs when they are restarted by the Amazon AWS environment, if an elastic IP has not been given to the specific instance. Thus it is necessary to alter this IP in the client code for getting the prediction. Alternatively, if an elastic IP is assigned to this instance, the change is not necessary unless the elastic IP is released at some point back to Amazon.

Furthermore, the base templates of Amazon may not contain all the necessary libraries for the MCR installation. From our experience with Ubuntu 12.04 32 bit template we used for the migration to Amazon, the following two libraries must be installed:

sudo apt-get install lesstif2-dev

sudo apt-get install libxmu6

4.5.1 Load Balancing features

In order to enable the basic elasticity features, an architecture with an Elastic Load Balancer must be followed. This means that an ELB instance is created in the front end, and identical back end FCS server VMs are started in order to distribute the load. This implies that all the models that have been created are copied to all the back end VMs. The connections to the FCS server are stateless, so there is no need for sticky connections or other kind of provision.

4.5.1.1 Instance creation from original FCS Server

After the base FCS server image has been created, we can create a template from this, in order to be used for all the instances. In order to do so, follow the process identified in [13]:

1. Select the checkbox of the FCS server.

2. Click 'Instance actions'

3. Click 'Create Image (EBS AMI)', following the instructions. This will create a base image from which more instances may be launched

4. Click 'Instance actions' again while still selected on the FCS server.

5. Click 'Launch more like this' in the menu.

6. Choose from the ‘My AMIs’ group the one that has just been created.

This will create an identical image instance of the original FCS server.

4.5.1.2 ELB activation

For creating an AWS Elastic Load Balancer, the following steps must be followed (modified from [14])

1. Launch the AWS Management Console
2. Select Load Balancers from the left column of the management console.
3. In the upper right pane, select “Create Load Balancer”
4. Create a unique name for the Load Balancer, e.g. FCSLoadBalancer.

5. Select the protocol(s) to manage via the Load Balancer (HTTP, HTTPS, TCP, SSL, or Custom). For the FCS you will need to add to the default HTTP and port 80 the Tomcat port (HTTP at port 8080).
6. Configure health checks for the Load Balancer. You should change the default 80/index.htm to a path that can be found in the FCS server (e.g. 8080/RELEASE-NOTES.txt). The Load Balancer will ping this location to determine whether the server is up and running.
7. Select the instances that you wish to place behind the Load Balancer. These should be the original FCS server plus the added VMs that were created in the previous paragraph.
8. Change the URL of the client to the DNS name of the ELB.

4.6 Getting started

4.6.1 Using the Software

For this information please consult the User Guide.

4.7 FAQ

1. Do I have to install a full version of Matlab?

No, only the MCR.. Prediction Models (e.g. ANNs) however need to be created and stored as Matlab .mat files in the according environment.

2. Can I change the paths to the models and XML model descriptors?

Yes, but in this case you will have to rebuild the war file in an environment like Eclipse of Netbeans.

5 DM AWS Plugin

The DataManager AWS Plugin provides two main functionalities. The first one is virtual machine upload to S3 Object Storage System. This component is part of the DataManagerClient. The DataManagerClient provides a wrapper of the amazon services for uploading VM images to S3 using the OPTIMIS DM API. This component is included inside the DataManagerClient, the client is responsible for abstracting the usage and loading of the plugin. The second component is a server side component, which is responsible for interacting with Amazon EC2 web services. This component is configured and installed only in Amazon DataManager in order to support Federation of Data Node VMs.

For the DM Openebula interoperability scripts, no specific installation process is necessary, they are included in the normal DM installation described in Chapter 2.

5.1 Release information

Table 8 AWS Plugin release information

Component Name	Release Number	Release Date
DM AWS Plugin	1.0	31-5-2013

5.2 Minimal System Requirements

The client side of the DataManager AWS Plugin is implemented as a java library and it is loaded with the DataManagerClient. All dependencies are included inside the building instructions of the component. The server side component responsible for interacting with Amazon EC2 web services has been implemented as a python service using the boto python framework[15]. The requirements for running this python service is a Linux OS and an installation of Python 2.6.6 and Python 2.7.1. The service is started using God - A Process Monitoring Framework[16].

5.3 Platforms Supported

The system is working in Linux Distributions, specifically tested on Ubuntu 10.10.

5.4 Software Pre-requisites and Dependencies

Table 9 AWS Plug-in Software dependencies

Product	Version	License
Python	2.6.6, 2.7.1	GPL-compatible
pip	v1.4	MIT
boto	2.9.5	GPL-compatible

5.5 Installation Instructions

unzip optimis-amazon-datamanager-ec2.zip

cd optimis-amazon-datamanager-ec2

./install.sh

5.6 Getting started

5.6.1 Using the Software

The optimis-amazon-datamanager-ec2 is working as a standalone service exposing an RPC interface. This RPC interface is called by the datamanager services. The RPC is using msgpack RPC[17] protocol. The main function is CreateCluster(serviceID, number_nodes).

5.7 FAQ

Where I have to set the credentials for Amazon Web service API?

Create a simple text file /etc/amazon.txt and set the values below:

Amazon_account_name=XXXXXXXXX

Amazon_account_password=XXXXXXXXXXXXX

6 Secure Storage

6.1 Release Information

Table 10: Secure Storage Release Information

Component Name	Release Number	Release Date
Secure Storage	1.2	29/05/2012

6.2 Minimal System Requirements

CPU - One virtual-core processor
Memory - 650MB
Hard disk space - 30MB to install Secure Storage Runtime Agent
Guest OS:
. CentOS 5.5 32-bit version
. CentOS 5.5 64-bit version
. Red Hat Enterprise Linux 6 32-bit version
. Red Hat Enterprise Linux 6 64-bit version
. Ubuntu 10.10 32-bit version
. Ubuntu 10.10 64-bit version
. SUSE Linux Enterprise Linux 11.4
. Windows 7
. Windows 7 Ultimate
. Windows Server 2003 R2 Datacenter SP2 32-bit version
. Windows Server 2003 R2 Datacenter SP2 64-bit version
. Windows Server 2008 R2 Datacenter SP2 32-bit version
. Windows Server 2008 R2 Datacenter SP2 64-bit version
. Windows Server 2008 Datacenter SP2 32-bit version
. Windows Server 2008 Datacenter SP2 64-bit version

6.3 Platforms Supported

X86 CPU architecture, CentOS Linux.

6.4 Software Dependencies

The software dependencies are:

Table 11 Secure Storage Software dependencies

Product	Version	Licence
python-xml	Latest	GPL
lua	Latest	GPL
cryptsetup	Latest	GPL

6.5 Installation Instructions in a Linux Environment

1. Install the package management system.

CentOS/RHEL/SUSE:

Use Rpm to install the RPM packagE by executing the following command:

rpm -ivh <scagnet.rpm>

where <scagent.rpm> is the name of the installation.

Ubuntu:

Install Gdebi to then install the DEB package by executing the following command:

gdebi <scagent.deb>

where <scagent.deb> is the name of the installation package download in step 1.

2. Launch the Secure Storage Configuration Tool by executing the following command:

/var/lib/securecloud/scconfig.sh

3. Accept the license agreement.

4. When prompted, select the appropriate option for your cloud service provider.

5. Enter your Cloud Service Provider credentials as prompted.

These credentials will differ depending on the IP.

Note: Not all Cloud Service Provider environments require credentials.

6. Enter your Secure Storage Account ID when prompted.

The Secure Storage Account ID can be found at "SecureCloud Web console | Administration > User Management".

7. Enter your provision passphrase when prompted.

8. Leave the Web Service URI field blank and press Enter to use the default WebService API URL if you are in SaaS mode. Otherwise, enter your Web Service API URL when promoted.

The Web Service URI can be found at "SecureCloud Web Console | Administration > User Management".

9. Follow the prompts as they appear.

The provisioning passphrase can be found and set at "SecureCloud Web Console | Administration > User Management".

6.5.1 Using the Software

Since this is a service that will be started on the boot of the VM, there is no specific need for the user to interact with the software.

7 References

1. Architecture Design document, OPTIMIS project Deliverable D1.2.1.1, UMEA and other partners, September 2010.
2. Requirements Analysis, OPTIMIS project Deliverable D1.1.1.1, ULEEDS and other partners, September 2010.
3. Cloudera distribution:
   http://www.cloudera.com/
4. Apache Hadoop Distributed File System:
   http://hadoop.apache.org/hdfs/
5. Apache Hadoop Distributed File System design concepts
   http://hadoop.apache.org/core/docs/current/hdfs_design.html
6. OPTIMIS deliverable D1.2.2.1 (MD1) OPTIMIS Detailed Design document, UMEA and other partners, November 2010
7. FUSE client page:
   http://fuse.sourceforge.net/
8. SNIA traces:
   http://iotta.snia.org/traces/list/BlockIO
9. OPTIMIS SLA framework and term languages, OPTIMIS project Deliverable D2.2.2.1, Fraunhofer and other partners, May 2010.
10. Matlab MCR component:
    http://www.mathworks.com/products/compiler/mcr/
11. Tomcat 6 setup process:
    http://tomcat.apache.org/tomcat-6.0-doc/setup.html
12. Tomcat 6 installation process in Ubuntu distribution:
    http://www.howtogeek.com/howto/linux/installing-tomcat-6-on-ubuntu/
13. Instructions for EC2 Image cloning:
    http://webdevrefinery.com/forums/topic/8012-cloning-an-ec2/page__p__80988#entry80988
14. AWS ELB:
    http://aws.amazon.com/elasticloadbalancing/
15. Boto framework:
    http://code.google.com/p/boto/
16. God Process monitoring framework:
    http://godrb.com/
17. Msgpack-RPC project:
    https://github.com/msgpack/msgpack-rpc