**Deployment Guide**

Genomic Information System for Integrated Science 2

(Genisis2) Technical Services

Release 3



June 2017

Document Version 2.0

Department of Veterans Affairs

Document Revision History

| Date | Revision | Description | Author |
| --- | --- | --- | --- |
| 02/10/2017 | 1.0 | Release 1 | Booz Allen Hamilton |
| 05/12/2017 | 2.0 | Release 2 | Booz Allen Hamilton |
| 06/16/2017 | 3.0 | Release 3 | Booz Allen Hamilton |

Table of Contents

[1. Introduction 1](#_Toc485369078)

[2. Genisis2 Web Application Server Setup 2](#_Toc485369079)

[2.1.1. Setup the Project on Remote Server (VA Environment) 2](#_Toc485369080)

[3. Genisis2 Services Server Setup 4](#_Toc485369081)

[3.1.1. Proxy Apache Server Setup 4](#_Toc485369082)

[3.1.2. Wildfly Application Server Setup 6](#_Toc485369083)

[4. Database Setup 6](#_Toc485369084)

[4.1.1. Database Names 6](#_Toc485369085)

[4.1.2. Database Schema 6](#_Toc485369086)

[4.1.3. List of Tables within the Database 7](#_Toc485369087)

[4.1.4. Users Recognized by the Database 7](#_Toc485369088)

[4.1.5. Database Scripts 7](#_Toc485369089)

[5. Data Operations Setup 8](#_Toc485369090)

[5.1.1. How the Table Copy Process Works 8](#_Toc485369091)

[5.1.2. Setting up Linux Environment 8](#_Toc485369092)

[5.1.3. Setting up Windows SQL Server Management Server 8](#_Toc485369093)

[5.1.4. Setting up the Source Server 9](#_Toc485369094)

[5.1.5. Setting up the Destination Server 10](#_Toc485369095)

[5.1.6. Test Reachability Between Servers 10](#_Toc485369096)

# Introduction

The Genisis2 Deployment Guide describes in detail how to install Genisis2 applications in the SQA, PrePROD, and PROD environments. The Genisis2 Application Architecture is shown in Figure 1.

VA Intranet

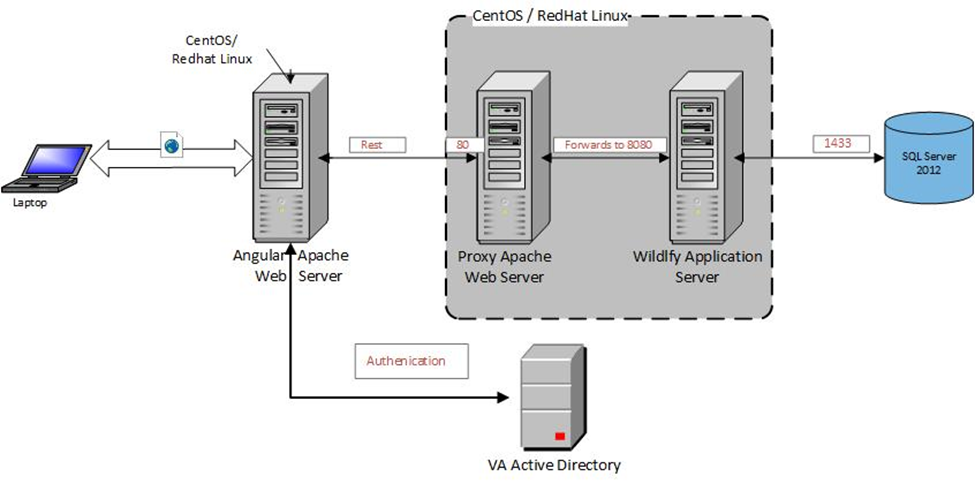


Figure : Genisis2 Application Architecture

In order to proceed with the installation, the two Red Hat Linux servers should have the following installed or at least have permissions to install:

* Apache Webserver
* Open SSL
* SSL certificates that needs to be installed on two Apache (Angular and Proxy) servers
* Wildfly 10.0.0-Final Application Server
* JDK 1.8.0\_92 or above

Other artifacts that are required to proceed with this installation are as follows:

* Environment Properties files:
* env.js for Angular web application
* genisis2.properties for Services application
* Application Help zip file
* Angular application Release zip artifact
* Services application Release war artifact
* Database Scripts Release Zip artifact
* dataTableCopy.sh shell script file

# Genisis2 Web Application Server Setup

## Setup the Project on Remote Server (VA Environment)

Assuming VA Linux boxes are Red Hat, then follow this instruction. If you were provisioned a separate server with another operating system, follow up with your System Administrator.

**System Level Requirements:**

1. $ sudo yum update
2. $ sudo yum search httpd
3. Select the correct httpd distro provided by VA RPM repository
4. $ sudo yum install httpd.<version>
5. If it already states installed, ignore and move on.

**Configure HTTPD:**

1. Make sure /var/www is owned by root and assigned correct privileges
   1. chown -Rf root:apache /var/www
2. Create directory for the AngularJS web application
   1. mkdir /var/www/DNS
   2. mkdir /var/www/DNS/webroot
   3. mkdir /var/www/DNS/logs
   4. chown -Rf root:apache /var/DNS
   5. Set permissions chmod -Rf 775 /var/www
3. Copy AngularJS Application to /var/www/DNS/webroot
4. Make sure /var/www/DNS/webroot/help/env.js is present
   1. Make sure the json key value for keyname ‘apiURL’ is populated with the url for the wildfly application server.
5. Request and Install DNS SSL Certificates
   1. Make sure private key is placed in /etc/pki/private/DNSkey
   2. Make sure certificate is placed in /etc/pki/certs/DNS.crt
6. Retrieve root certificate bundle and install
   1. /etc/pki/tls/certs/ca-bundle.crt
7. Setup Apache for domain
   1. Edit /etc/httpd/conf.d/vhosts.conf as Follows:

<VirtualHost \*:80>

RewriteEngine on

RewriteCond %{SERVER\_PORT} !^443$

RewriteRule ^/(.\*) https://%{HTTP\_HOST}/$1 [NC,R=301,L]

</VirtualHost>

NameVirtualHost \*:443

<VirtualHost \*:443>

ServerAdmin PII.

DocumentRoot /var/www/DNS/webroot

ServerName DNS

ServerAlias DNS

ErrorLog /var/www/DNS/logs/DNS\_error\_log

CustomLog /var/www/DNS/logs/DNS\_access\_log common

CustomLog /var/www/DNS/logs/DNS\_ssl\_request\_log "%t %h %{SSL\_PROTOCOL}x %{SSL\_CIPHER}x %{SSL\_CLIENT\_S\_DN}x \"%r\" %b"

# activate HTTPS on the reverse proxy

SSLEngine On

SSLCertificateFile /etc/pki/tls/certs/DNS.crt

SSLCertificateKeyFile /etc/pki/tls/private/DNS.key

SSLProtocol -all +TLSv1 +TLSv1.1 +TLSv1.2

SSLCipherSuite EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:ECDHE- RSA-AES128-SHA:DHE-RSA-AES128-GCM-SHA256:AES256+EDH:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GC$

# activate the client certificate authentication

SSLCACertificateFile /etc/pki/tls/certs/ca-bundle.crt

SSLVerifyClient require

SSLVerifyDepth 3

# initialize the special headers to a blank value to avoid http header forgeries

Header set SSL\_CLIENT\_S\_DN ""

Header set SSL\_CLIENT\_I\_DN ""

Header set SSL\_SERVER\_S\_DN\_OU ""

Header set SSL\_CLIENT\_VERIFY ""

<Location />

# add all the SSL\_\* you need in the internal web application

Header set SSL\_CLIENT\_S\_DN "%{SSL\_CLIENT\_S\_DN}s"

Header set SSL\_CLIENT\_I\_DN "%{SSL\_CLIENT\_I\_DN}s"

Header set SSL\_SERVER\_S\_DN\_OU "%{SSL\_SERVER\_S\_DN\_OU}s"

Header set SSL\_CLIENT\_VERIFY "%{SSL\_CLIENT\_VERIFY}s"

</Location>

</VirtualHost>

**Installing Genisis2WEB:**

1. Get the appropriate release versions from the dev team for this release.
2. Go to the url: http://genisis20-nexus.boozallencsn.com/nexus/
3. Click on login on top right side and enter following credentials
4. Username: Jenkins
5. Password: root2017
6. Then go to the following url: http://genisis20-nexus.boozallencsn.com/nexus/content/repositories/
7. Be sure to be off of VA VPN. The above link will not work in VA VPN.
8. SFTP Release version to Server, in any means accessible by you. SFTP will direct to /home/<user>/ - so from there unzip the file.
9. $ unzip <Genisis2Web Zip> /var/www/DNS/webroot
10. $ sudo mkdir help
11. SFTP help.zip file to Server, in any means accessible by you. . SFTP will direct to /home/<user>/ - so from there unzip the file.
12. $ unzip <Genisis2Web Help Zip> /var/www/DNS/webroot/help
13. Copy env.js file provided under /var/www/DNS/webroot/help
14. Restart Apache (service httpd restart)

# Genisis2 Services Server Setup

There are two severs that need to be setup. One is Apache, which redirects all the requests to Wildfly Application Server.

## Proxy Apache Server Setup

1. Make sure /var/www is owned by root and assigned correct privileges
2. Chown -Rf root:apache /var/www
3. Create directory for the Proxy web application
4. mkdir /var/www/DNS
5. mkdir /var/www/DNS/webroot
6. Mkdir /var/www/DNS/logs
7. Chown -Rf root:apache /var/www/DNS
8. Set permissions chmod -Rf 775 /var/www
9. Copy Proxy Application to /var/www/DNS/webroot
10. Request and Install DNS SSL Certificates.
11. Make sure private key is placed in /etc/pki/private/DNS.key
12. Make sure certificate is placed in /etc/pki/certs/DNS.crt
13. Setup Apache for domain Proxy
14. Edit /etc/httpd/conf.d/vhosts.conf as Follows:

<VirtualHost \*:80>

RewriteEngine on

RewriteCond %{SERVER\_PORT} !^443$

RewriteRule ^/(.\*) https://%{HTTP\_HOST}/$1 [NC,R=301,L] </VirtualHost>

<VirtualHost \*:443>

ServerAdmin PII

DocumentRoot /var/www/DNS/webroot

ServerName DNS

ServerAlias DNS

ErrorLog /var/www/DNS/logs/DNS\_error\_log

CustomLog /var/www/DNS/logs/DNS\_access\_log common

CustomLog /var/www/DNS/logs/DNS\_ssl\_request\_log "%t %h %{SSL\_PROTOCOL}x %{SSL\_CIPHER}x$

# activate HTTPS on the reverse proxy

SSLEngine On

SSLCertificateFile /etc/pki/tls/certs/DNScrt

SSLCertificateKeyFile /etc/pki/tls/private/DNS.key

SSLProtocol -all +TLSv1 +TLSv1.1 +TLSv1.2

SSLCipherSuite EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:ECDHE-RSA-AES128-SHA:DHE-RSA-AES128-GCM-SHA256:AES256+EDH:ECDHE-RSA-AES256-GCM-SHA384:E$

ProxyPass "/" "http://localhost:8080/"

ProxyPass "/Genisis2Services" "http://localhost:8080/Genisis2Services/"

ProxyPassReverse "/Genisis2Services" "http://localhost:8080/Genisis2Services/"

#Wildfly admin console access

ProxyPass "/console" "http://localhost:9990/console"

ProxyPassReverse "/console" "http://localhost:9990/console"

ProxyPass "/management" "http://localhost:9990/management"

ProxyPassReverse "/management" "http://127.0.0.1:9990/management"

</VirtualHost>

* 1. Restart Apache (service httpd restart)

## Wildfly Application Server Setup

1. Download the Wildfly application 10.0.0-Final zip file from [wildfly.org/downloads](http://www.wildfly.org/downloads/)
2. Download and install JDK 1.8.0\_92 and install it under /opt/JDK\_1.8.0\_92
3. Set JAVA\_HOME variable to point to bin directory in java installation folder
4. Under /opt directory create wildfly directory
   1. mkdir wildfly
   2. Set permissions chmod 775 -R /opt/wildfly
5. Unzip the downloaded Wildfly zip file to /opt/wildfly directory
6. You should now see a “*standalone”* directory under your wildlfy home
7. Copy the gensis2.properties under /opt/wildfly/standalone/configuration directory
8. Open genisis2.properties file using vim and enter correct values pertaining to the environment on which the application is being installed
9. Reach out to the development team for the correct values to be filled for each property listed in the genisis2.properties file
10. SFTP Release version to Server by any means accessible by you. SFTP will direct to /home/<user>/
11. Copy the war file to /opt/wildfly/standalone/deployments folder
    1. sudo cp /home/<user>/ Genisis2Services.war /opt/wildfly/standalone/deployments
12. Before starting the server, make sure that database setup is complete and the property
13. Start the wildfly server
    1. nohup /opt/wildfly/bin/standalone.sh &

# Database Setup

## Database Names

Genisis\_DB is our application database.

Activiti\_DB is the activiti workflow database.

## Database Schema

The schema used for the application database is ‘dbo’.

## List of Tables within the Database

1. Request History
2. RequestType
3. Request
4. CommentHistory
5. StudyApproval
6. WorkflowStatus
7. Source
8. Users
9. User\_Role\_Type
10. User\_Approver
11. User\_Type
12. Role\_Type
13. Management Table (We are creating the Management table only for the table copy function. This behaves as a log for the Table copy process.)
14. Other systems generated tables by Activiti

## Users Recognized by the Database

“genisis” is the application user in the database with the rights of data reader and data writer within the database.

## Database Scripts

Unzip the dbscripts and run the scripts in the following order:

1. GENISIS\_DB
   1. CreateDatabase.sql
   2. User.Sql
   3. Tables.Sql
   4. looklookuptables.sql
2. ACTIVITI\_DB
   1. CreateDatabase.sql
   2. User.Sql

# Data Operations Setup

## How the Table Copy Process Works

There are three servers involved in table copy process as follows:

1. Management Server – This is the same as the Genisis2 Microsoft Windows 2012/ SQL Server 2012 Database server.
2. Source Server – Microsoft Windows /SQL Server database from which tables are copied (Example: VINCI Landing Zone server).
3. Destination Server – Microsoft Windows/SQL Server to which tables are copied (Example: Genisis Landing Zone server).

## Setting up Linux Environment

1. Make sure all the properties are updated in /opt/wildfly/standalone/configuration/genisis2.properties
2. SFTP Release version to Server, in any means accessible by you. SFTP will direct to /home/<user>/
3. Copy the jar file to /opt/genisisDataOps folder.
   1. sudo cp /home/<user>/ Genisis2DataOps.jar /opt/genisisDataOps/Genisis2DataOps.jar
4. SFTP the shell script and copy to the /opt/genisisDataOps folder
   1. sudo cp /home/<user>/ databaseTableCopy.sh /opt/genisisDataOps/ databaseTableCopy.sh

The table copy process will be initiated from the Genisis2 user interface. It executes the above .sh file (shell script). In turn, the.jar file that performs the table copy process is invoked.

## Setting up Windows SQL Server Management Server

1. Users and permissions needed
2. Make sure that Microsoft Windows 2012 R2 and SQL server 2012 are installed in the Destination and Management databases
3. Always use Fully qualified Domain Names (FQDNs and not IP addresses)
4. You need a username and password with the following permissions. This is for routine Genisis2 initiated Table Copy Operations.
5. ddlread
6. ddlwrite
7. We need to create a linked server in the Source Server. For this, you may need access to a username and password with Administrator rights for setting the linked server up. This is a one-time task.
8. Creating a Linked Server from the Management Server (Example: DNS- VA SQA Source Server to the (use FQDN not ip) to the Source Server (Example: DNS VA SQA Source Server) to Linked Server:
9. Login to the Management server as an Administrator
10. Create linked server in the Management for Source Server
11. Once you see the Linked Source Server, open it and make sure you can see the Database you are given access to and the tables within them. Run an auto generated SQL script to see say the first 1000 rows of a table just to make sure that the management server can read that database and tables within that database.
12. Create a Linked Server from the Destination Server to the Management Server (For Ex: DNS - VA SQA Destination Server):
13. Table Copying is done in two steps
14. Management Server PULLS the table from the Source Server in to a TEMP database (GENISISTEMPDATA)
15. Destination Server PULLS the table from the Management Server (GENISISTEMPATA) to the its database (GENISISDATAOPSDATA – This is also the Genisis Landing Zone Database)
16. You need a username and password with ADMINISTRATOR rights in the Destination Server just for creating the Linked Server)
17. Create a Linked Server in the Destination Server for the Management Server
18. Once you see the Linked Management Server, open it and make sure you can see the GENISISTEMPDATA Database
19. Management Table part of GenisisDB (Script) (This part is included in the Genisis2 build and this is for reference – No tasks to be done here)

## Setting up the Source Server

The Source Server can be a remote server like the VINCI landing Zone. Some other group may be responsible for administering that database. This step involves coordination with that group and making sure that the following steps are completed:

1. Getting an account on the source server: Coordinate with the source server administrative team to obtain a user name and password created with following permissions and the name of the Database from which you will be copying tables. The permissions needed for this database are:
   1. ddlRead
   2. ddlWrite
   3. ddladmin
2. We are asking for the above permissions; since in the future, this Source Server can serve as a Destination Server also. You should be able to create a table there and write to it.

## Setting up the Destination Server

1. Get an account on the Destination Server with the following permissions:
2. ddlRead
3. ddlWrite
4. ddlAdmin

## Test Reachability Between Servers

1. Management Server to Source Server:
2. From a Windows server (within the same subnet with permissions to access the Destination Server) login to the Management Server using SQL Server Management Studio. Determine if you can click on linked servers, open them, and see Genisis\_DB and the Management\_Table. You need to be able to do this to verify if tables have been copied successfully, and the Number of Rows and Checksums, BEFORE, and AFTER a Table Copy. This table contains the log for Table Copy Operations.
3. Confirm that you can reach the Source Server and look at the database given to you. Determine if the tables are there, and that you are able to run queries against this database.
4. Destination Server to Management Server:
5. From any Windows server, login to the Destination Server using SQL Server Management Studio. Confirm that you can click on the linked servers and open the Management Server. Determine if you can see the GENISISTEMPDATA database. Since it is a temporary database where tables are stored temporarily, you may not see any tables.