# Deployment Guide

Genomic Information System for Integrated Science 2 (Genisis2) Technical Services

Build 7



December 2017

Document Version 6.0

### Department of Veterans Affairs

### Document Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Revision** | **Description** | **Author** |
| 02/10/2017 | 1.0 | Build 1 updates. | Booz Allen Hamilton |
| 05/12/2017 | 2.0 | Build 2 updates. | Booz Allen Hamilton |
| 06/16/2017 | 3.0 | Build 3 updates. | Booz Allen Hamilton |
| 09/15/2017 | 4.0 | Build 4 updates. | Booz Allen Hamilton |
| 11/03/2017 | 5.0 | Build 5 updates. | Booz Allen Hamilton |
| 11/24/2017 | 6.0 | Build 7 updates. | Booz Allen Hamilton |

**Table of Contents**

1. [Introduction 1](#_bookmark0)
   1. [Genisis Data Request Workflows Architecture 1](#_bookmark1)
   2. [Genisis Terminology Services Architecture 2](#_bookmark2)
2. [Genisis2 Web Application Server Setup 3](#_bookmark3)
   1. [Setup the Project on Remote Server (VA Environment) 3](#_bookmark4)
      1. [nisis Data Request Workflow – Apache HTTP Setup 3](#_bookmark5)
      2. [nisis Terminology Services – Apache HTTP Setup 5](#_bookmark6)
   2. [Steps to Configure Internet Explorer for PIV Authentication 8](#_bookmark7)
3. [Genisis2 Services Server Setup 10](#_bookmark8)
   1. [Services Setup for Genisis Data Request Workflows 10](#_bookmark9)
      1. [oxy Apache Server Setup 10](#_bookmark10)
      2. [Wildfly Application Server Setup 11](#_bookmark11)
   2. [Services Setup for Genisis Terminology Services 12](#_bookmark12)
      1. [cat Server Setup 12](#_bookmark13)

[Install Tomcat 12](#_bookmark14)

* + 1. [Jena/Fuseki Setup 21](#_bookmark15)
    2. [SOLR Setup 23](#_bookmark16)

1. [Database Setup 24](#_bookmark17)
   1. [Genisis Data Request Workflows Database Setup 24](#_bookmark18)
      1. [base Names 24](#_bookmark19)
      2. [base Schema 24](#_bookmark20)
      3. [st of Tables within the Database 24](#_bookmark21)
      4. [rs Recognized by the Database 25](#_bookmark22)
      5. [ase Scripts 25](#_bookmark23)

[4.2 Genisis Terminology Services Database Setup 25](#_bookmark24)

[4.2.1 Database Names 25](#_bookmark25)

[4.2.2. Database Schema 25](#_bookmark26)

* + 1. [sers Recognized by the Database 25](#_bookmark27)
    2. [base Scripts 26](#_bookmark28)

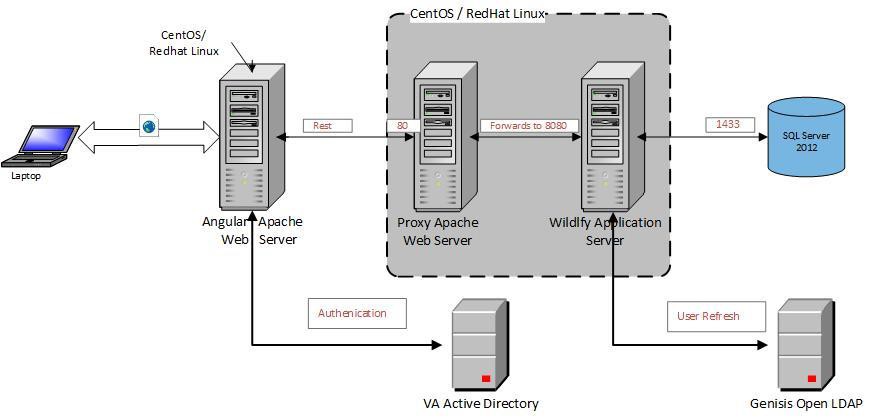
1. [Genisis Data Request Workflow - Data Operations Setup 26](#_bookmark29)
   1. [How the Table Copy Process Works 26](#_bookmark30)
   2. [Setting up Linux Environment 26](#_bookmark31)
   3. [Setting up Windows SQL Server Management Server 26](#_bookmark32)
   4. [Setting up the Source Server 27](#_bookmark33)
   5. [Setting up the Destination Server 28](#_bookmark34)
   6. [Test Reachability Between Servers 28](#_bookmark35)
2. [Genisis Data Request Workflow - Upgrading from Build 3 to Build](#_bookmark36) [4 29](#_bookmark36)
   1. [Deploy Genisis2 Angular Application 29](#_bookmark37)
   2. [Update Genisis2 Databases 29](#_bookmark38)
   3. [Deploy Genisis2 Services Application 29](#_bookmark39)

# Introduction

## Genisis Data Request Workflows Architecture

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

VA Intranet



**Figure 1: Genisis2 Application Architecture for Data Request Workflows**

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 need 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:

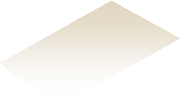
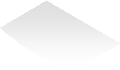
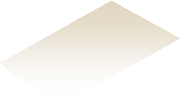
* + - Environment Properties files:
    - env.js for Angular web application
    - genisis2.properties for Services application
    - Application Help zip file
    - Angular application Release zip artifact
    - Services application Release war artifact
    - Database Scripts Release Zip artifact
 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

## Genisis Terminology Services Architecture

The Genisis2 Application Architecture for Terminology Services is shown in Figure 2.

VA Intranet

Centos/RedHat Linux



**Port 80**

**Web Server/ Application Server Running**

* **Apache HTTP**
* **Tomcat**
* **Jena/Fuseki**
* **SOLR**

**Database Server**

* **Windows Server 2008 R2**
* **Microsoft SQL Server 2012**

**Figure 2: Genisis2 Application Architecture for Terminology Services**

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

* + - Apache Webserver
    - Open SSL
    - SSL certificates that need to be installed on two Apache (Angular and Proxy) servers
    - Tomcat 9.0-Final Application Server
    - JDK 1.8.0\_92 or above
    - Apache Jena 3.4.0
    - Apache SOLR 6.0.0

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

* + - Environment Properties files:
    - env.js for Angular web application
    - Application Help zip file
    - Angular application Release zip artifact
    - Services application Release war artifact
    - Database Scripts Release Zip artifact

# Genisis2 Web Application Server Setup

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

### Genisis Data Request Workflow – Apache HTTP Setup

Assuming VA Linux boxes are Red Hat, then follow this instruction. If 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/domain.va.gov
   2. mkdir /var/www/domain.va.gov/webroot
   3. mkdir /var/www/domain.va.gov/logs
   4. chown -Rf root:apache /var/www/domain.va.gov
   5. Set permissions chmod -Rf 775 /var/www
3. Copy AngularJS Application to /var/www/domain.va.gov/webroot
4. Make sure /var/www/domain.va.gov/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 domain.va.gov SSL Certificates
   1. Make sure private key is placed in /etc/pki/private/domain.va.gov.key
   2. Make sure certificate is placed in /etc/pki/certs/domain.va.gov.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 [admin@domain.va.gov](mailto:admin@domain.va.gov) DocumentRoot /var/www/domain.va.gov/webroot ServerName domain.va.gov

ServerAlias [www.domain.va.gov](http://www.domain.va.gov/)

ErrorLog /var/www/domain.va.gov/logs/domain.va.gov\_error\_log CustomLog /var/www/domain.va.gov/logs/domain.va.gov\_access\_log

common

CustomLog /var/www/domain.va.gov/logs/domain.va.gov\_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/domain.va.gov.crt SSLCertificateKeyFile /etc/pki/tls/private/domain.va.gov.key SSLProtocol -all +TLSv1 +TLSv1.1 +TLSv1.2 SSLCipherSuite

EKL

# 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 the following credentials:
   1. Username: AI
   2. Password: AI
4. Then go to the [following url: http://genisis20-](http://genisis20-/) nexus.boozallencsn.com/nexus/content/repositories/
5. Be sure to be off of VA VPN. The above link will not work in VA VPN.
6. SFTP Release version to Server, in any means accessible by you. SFTP will direct to

/home/<user>/ - so from there unzip the file.

1. $ unzip <Genisis2Web Zip> /var/www/domain.va.gov/webroot
2. $ sudo mkdir help
3. SFTP help.zip file to Server, in any means accessible by you. SFTP will direct to

/home/<user>/ - so from there unzip the file.

1. $ unzip <Genisis2Web Help Zip> /var/www/domain.va.gov/webroot/help
2. Copy env.js file provided under /var/www/domain.va.gov/webroot/help
3. Restart Apache (service httpd restart)

### Genisis Terminology Services – Apache HTTP Setup

Assuming VA Linux boxes are Red Hat, then follow this instruction. If 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/domain.va.gov
   2. mkdir /var/www/domain.va.gov/webroot
   3. mkdir /var/www/domain.va.gov/logs
   4. chown -Rf root:apache /var/www/domain.va.gov
   5. Set permissions chmod -Rf 775 /var/www
3. Copy AngularJS Application to /var/www/domain.va.gov/webroot
4. Make sure /var/www/domain.va.gov/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 domain.va.gov SSL Certificates
   1. Make sure private key is placed in /etc/pki/private/domain.va.gov.key
   2. Make sure certificate is placed in /etc/pki/certs/domain.va.gov.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 [admin@domain.va.gov](mailto:admin@domain.va.gov) DocumentRoot /var/www/domain.va.gov/webroot ServerName domain.va.gov

ServerAlias [www.domain.va.gov](http://www.domain.va.gov/)

ErrorLog /var/www/domain.va.gov/logs/domain.va.gov\_error\_log CustomLog /var/www/domain.va.gov/logs/domain.va.gov\_access\_log

common

CustomLog /var/www/domain.va.gov/logs/domain.va.gov\_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/domain.va.gov.crt SSLCertificateKeyFile /etc/pki/tls/private/domain.va.gov.key SSLProtocol -all +TLSv1 +TLSv1.1 +TLSv1.2 SSLCipherSuite

EKL

# 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"

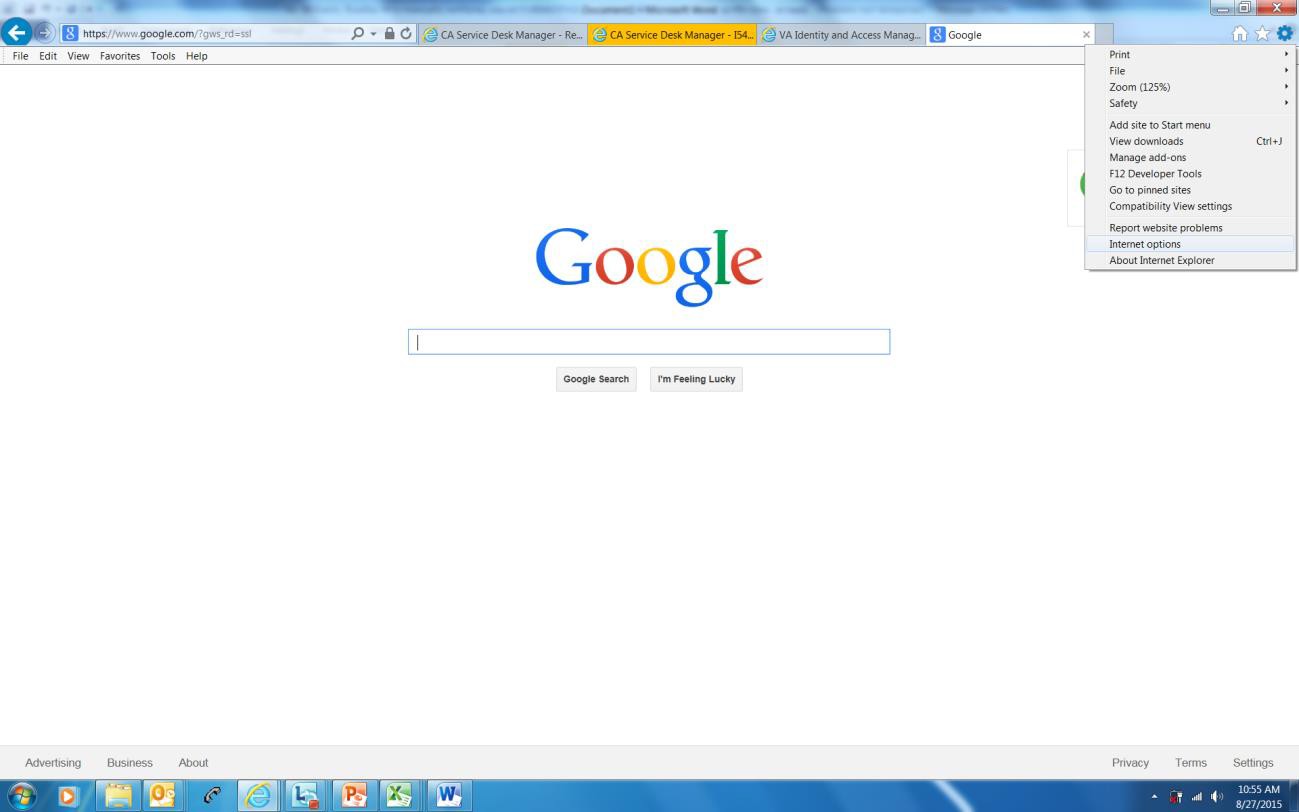
Header set SSL\_SERVER\_S\_DN\_OU "%{SSL\_SERVER\_S\_DN\_OU}s" Header set SSL\_CLIENT\_VERIFY "%{SSL\_CLIENT\_VERIFY}s"

</Location>

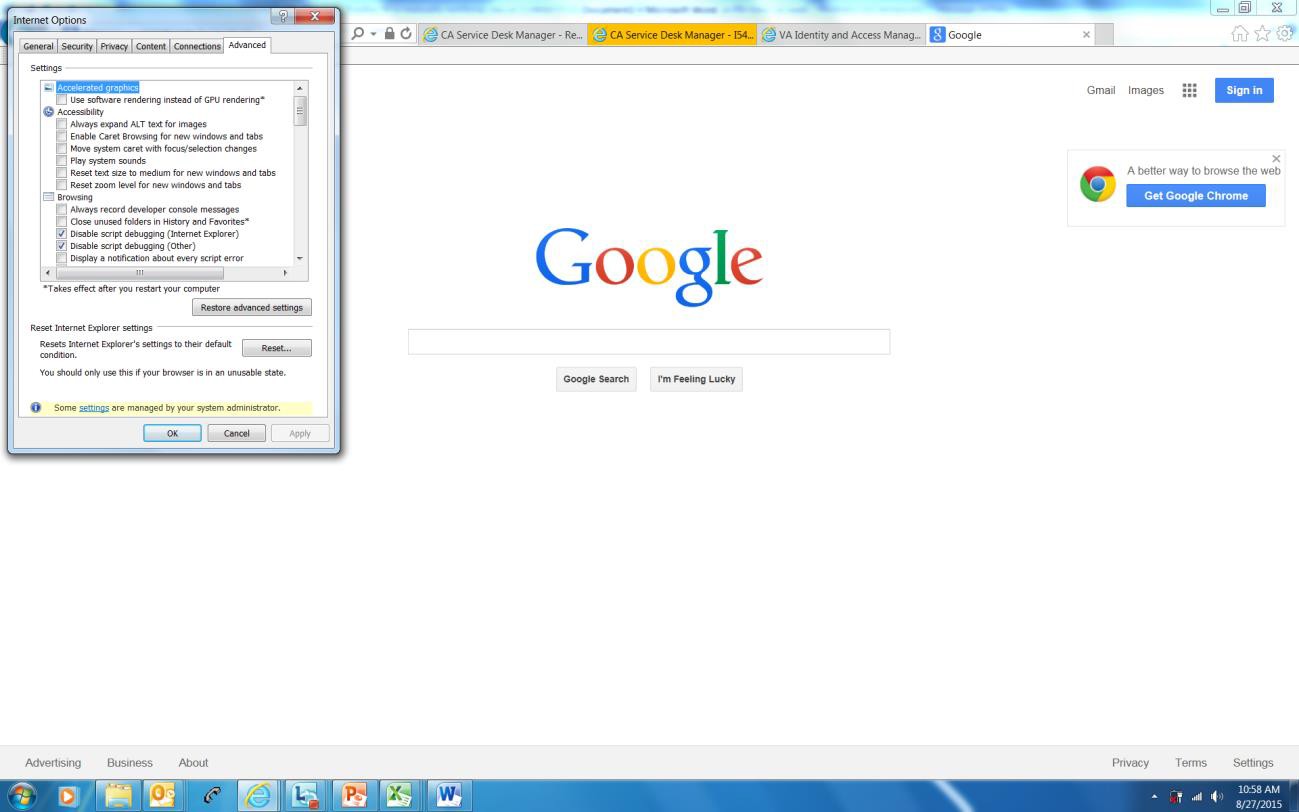
</VirtualHost>

## Steps to Configure Internet Explorer for PIV Authentication

1. Open your Web Browser (e.g. Internet Explorer)
2. Go to the **Tools** Menu (gear shaped icon in upper right corner)



1. Select the **Internet Options** menu in the Tools Menu
2. Select the **Advanced** tab



Make sure the **SSL 2.0** and **SSL 3.0** boxes are unchecked

1. Make sure the **TLS 1.0**, **TLS 1.1**, and **TLS 1.2** are checked
2. Make sure the “**check for publisher’s certificate revocation**” is unchecked
3. Close out and re-open Internet Explorer and retry authentication to the target application

# Genisis2 Services Server Setup

## Services Setup for Genisis Data Request Workflows

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

### oxy Apache Server Setup

* + - 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 Proxy web application
         1. mkdir /var/www/domain.va.gov
         2. mkdir /var/www/domain.va.gov/webroot
         3. Mkdir /var/www/domain.va.gov/logs
         4. Chown -Rf root:apache /var/www/domain.va.gov
         5. Set permissions chmod -Rf 775 /var/www
      3. Copy Proxy Application to /var/www/domain.va.gov/webroot
      4. Request and Install domain.va.gov SSL Certificates.
         1. Make sure private key is placed in /etc/pki/private/domain.va.gov.key
         2. Make sure certificate is placed in /etc/pki/certs/domain.va.gov.crt
      5. Setup Apache for domain Proxy
         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>

<VirtualHost \*:443>

ServerAdmin [admin@domain.va.gov](mailto:admin@domain.va.gov) DocumentRoot /var/www/domain.va.gov/webroot ServerName domain.va.gov

ServerAlias [www.domain.va.gov](http://www.domain.va.gov/) ErrorLog

/var/www/domain.va.gov/logs/domain.va.gov\_error\_log CustomLog

/var/www/domain.va.gov/logs/domain.va.gov\_access\_log common CustomLog

/var/www/domain.va.gov/logs/domain.va.gov\_ssl\_request\_log "%t %h

%{SSL\_PROTOCOL}x %{SSL\_CIPHER}x$

# activate HTTPS on the reverse proxy

SSLEngine On

SSLCertificateFile /etc/pki/tls/certs/domain.va.gov.crt SSLCertificateKeyFile /etc/pki/tls/private/domain.va.gov.key SSLProtocol -all +TLSv1 +TLSv1.1 +TLSv1.2 SSLCipherSuite

EKL

ProxyPass "/" "http://localhost:PORT/" ProxyPass "/Genisis2Services"

"http://localhost:PORT/Genisis2Services/" ProxyPassReverse "/Genisis2Services"

"http://localhost:PORT/Genisis2Services/"

#Wildfly admin console access

ProxyPass "/console" "http://localhost:PORT/console" ProxyPassReverse "/console" "http://localhost:PORT/console"

ProxyPass "/management" "http://localhost:PORT/management" ProxyPassReverse "/management"

"http://127.0.0.1:PORT/management"

</VirtualHost>

a) 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>/

* + - 1. Copy the war file to /opt/wildfly/standalone/deployments folder
         1. sudo cp /home/<user>/ Genisis2Services.war /opt/wildfly/standalone/deployments
      2. Before starting the server, make sure that database setup is complete and the property
      3. Start the wildfly server
         1. nohup /opt/wildfly/bin/standalone.sh &

## Services Setup for Genisis Terminology Services

For Terminology Services, the Webserver and the Application Server are one and the same. The following services need to be set up here for Terminology Services

### Tomcat Server Setup Install Tomcat

1. Install Apache Tomcat using the following installer to the drive and folder where needed. <https://tomcat.apache.org/download-80.cgi>
2. Add the following jar to <tomcat-home>/lib – mssql-jdbc-6.2.1.jre8.jar
3. Create <tomcat-home>/conf/termservice.properties and add the following contents

#Mapping Service URIs ts.ms.create.concept.mapping.uri=http://localhost:PORT/MappingService/mappings

ts.ms.get.concept.mapping.concept.uri=http://localhost:PORT/MappingService/mappings? conceptUri={conceptUri}

#Bookmarking Service URIs ts.ms.create.book.marks.uri=http://localhost:PORT/BookmarkingService/bookmarks

ts.ms.create.book.marks.with.label.uri=http://localhost:PORT/BookmarkingService/bookm arks/{id}/labels

ts.ms.get.book.marks.by.username.uri=http://localhost:PORT/BookmarkingService/bookm arks/{username}

ts.ms.get.book.marks.by.label.uri=http://localhost:PORT/BookmarkingService/bookmarks/ labels/{label}

ts.ms.get.book.marks.label.by.username.uri=http://localhost:PORT/BookmarkingService/l abels/{username}

ts.ms.delete.book.marks.by.id.uri=http://localhost:PORT/BookmarkingService/bookmarks/

{id}

ts.ms.get.book.mark.by.id=http://localhost:PORT/BookmarkingService/bookmarks/{id}

ts.ms.delete.book.marks.by.label.uri=http://localhost:PORT/BookmarkingService/bookmar ks/{id}/labels/{labelName}

#SOLR

solr.search.service.endpoint=http://IP:PORT/solr/fuseki

#solr.search.service.endpoint=http://IP:PORT/solr/fuseki/ts?wt=json&fl=id,Label

,Definition&q=Diabetes&start=%s

#SPARQL Queries ts.sparql.endpoint=http://IP:PORT/ds/sparql ts.sparql.endpoint.update=http://IP:PORT/ds/update concept.name.sparql.query=select (str(?name) as ?strname) \

where { ?subject ?p ?name; FILTER(?subject = <%s> && ( \

?p = <[http://www.w3.org/2000/01/rdf-schema#label>](http://www.w3.org/2000/01/rdf-schema#label) || \

?p = <[http://rdf.cdisc.org/ct/schema#cdiscSubmissionValue>](http://rdf.cdisc.org/ct/schema#cdiscSubmissionValue) || \

?p = <<http://www.w3.org/2004/02/skos/core#prefLabel>> \

)) }

concept.name.query.headvar.name=strname

concept.subtype.sparql.query=select ?child\_uri ?child\_name \

where { ?child\_uri <[http://www.w3.org/2000/01/rdf-schema#subClassOf>](http://www.w3.org/2000/01/rdf-schema#subClassOf)

<%s> . \

?child\_uri ?p ?child\_name; FILTER( \

?p = <[http://www.w3.org/2000/01/rdf-schema#label>](http://www.w3.org/2000/01/rdf-schema#label) || \

?p = <[http://rdf.cdisc.org/ct/schema#cdiscSubmissionValue>](http://rdf.cdisc.org/ct/schema#cdiscSubmissionValue) || \

?p = <<http://www.w3.org/2004/02/skos/core#prefLabel>> \

) \

}

subtype.query.head.var.child.uri=child\_uri subtype.query.head.var.child.name=child\_name

concept.predicate.sparql.query=select ?p ?o ?n \ where { \

{ ?s ?p ?o; FILTER(?s = <%s> && !isURI(?o)) } \ UNION \

{ ?s ?p ?o; FILTER(?s = <%s> && isURI(?o)) . \

?o ?pn ?n; FILTER( \

?pn = <[http://www.w3.org/2000/01/rdf-schema#label>](http://www.w3.org/2000/01/rdf-schema#label) || \

?pn = <[http://rdf.cdisc.org/ct/schema#cdiscSubmissionValue>](http://rdf.cdisc.org/ct/schema#cdiscSubmissionValue) || \

?pn = <[http://www.w3.org/2004/02/skos/core#prefLabel>](http://www.w3.org/2004/02/skos/core#prefLabel) \

) } \

}

concept.predicate.query.head.var.p=p concept.predicate.query.head.var.o=o concept.predicate.query.head.var.n=n

conceptmapping.data.elements.sparql.query=ASK { SELECT ?o WHERE { <%s>

<[http://genisis.va.gov/mvp-schema#MappingDataElement>](http://genisis.va.gov/mvp-schema#MappingDataElement) ?o } }

#SPARL Query to know URI exists in Fuseki

generated.uri.exists.sparql.query=ASK { { SELECT \* { <%s> ?p ?o } } UNION { SELECT \* {

?s <%s> ?o } } UNION { SELECT \* { ?s ?p <%s> } } }

ts.mvp.prefix=<http://genisis.va.gov/mvp-schema> ts.mvp.uri.delimiter=\_? ts.mvp.new.concept.regex=^.\*\\\_\\?\\d{1,3}$ Adjust your files to the following

#### <tomcat-home>/conf/tomcat-users.xml

<?xml version="1.0" encoding="UTF-8"?>

<!--

Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership.

The ASF licenses this file to You under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with

the License. You may obtain a copy of the License at<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

--><tomcat-users version="1.0" xmlns="<http://tomcat.apache.org/xml>" xmlns:xsi="<http://www.w3.org/2001/XMLSchema-instance>" xsi:schemaLocation="<http://tomcat.apache.org/xml> tomcat-users.xsd">

<!--

NOTE: By default, no user is included in the "manager-gui" role required to operate the "/manager/html" web application. If you wish to use this app, you must define such a user - the username and password are arbitrary. It is

strongly recommended that you do NOT use one of the users in the commented out section below since they are intended for use with the examples web

application.

-->

<!--

NOTE: The sample user and role entries below are intended for use with the examples web application. They are wrapped in a comment and thus are ignored when reading this file. If you wish to configure these users for use with the examples web application, do not forget to remove the <!.. ..> that surrounds them. You will also need to set the passwords to something appropriate.

-->

<!--

<role rolename="tomcat"/>

<role rolename="role1"/>

<user username="tomcat" password="<must-be-changed>" roles="tomcat"/>

<user username="both" password="<must-be-changed>" roles="tomcat,role1"/>

<user username="role1" password="<must-be-changed>" roles="role1"/>

-->

<role rolename="admin"/>

<role rolename="admin-gui"/>

<role rolename="admin-script"/>

<role rolename="manager"/>

<role rolename="manager-gui"/>

<role rolename="manager-script"/>

<role rolename="manager-jmx"/>

<role rolename="manager-status"/>

<user name="admin" password="adminadmin" roles="admin,manager,admin-gui,admin- script,manager-gui,manager-script,manager-jmx,manager-status" />

</tomcat-users>

#### <tomcat-home>/conf/server.xml

<?xml version="1.0" encoding="UTF-8"?>

<!--

Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership.

The ASF licenses this file to You under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with

the License. You may obtain a copy of the License at<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

--><!-- Note: A "Server" is not itself a "Container", so you may not

define subcomponents such as "Valves" at this level. Documentation at /docs/config/server.html

--><Server port="8005" shutdown="SHUTDOWN">

<Listener className="org.apache.catalina.startup.VersionLoggerListener"/>

<!-- Security listener. Documentation at /docs/config/listeners.html

<Listener className="org.apache.catalina.security.SecurityListener" />

-->

<!--APR library loader. Documentation at /docs/apr.html -->

<Listener SSLEngine="on" className="org.apache.catalina.core.AprLifecycleListener"/>

<!-- Prevent memory leaks due to use of particular java/javax APIs-->

<Listener className="org.apache.catalina.core.JreMemoryLeakPreventionListener"/>

<Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener"/>

<Listener className="org.apache.catalina.core.ThreadLocalLeakPreventionListener"/>

<!-- Global JNDI resources

Documentation at /docs/jndi-resources-howto.html

-->

<GlobalNamingResources>

<!-- Editable user database that can also be used by UserDatabaseRealm to authenticate users

-->

<Resource name="jdbc/TS\_DB" auth="Container" type="javax.sql.DataSource"

maxTotal="20" maxIdle="5" maxWaitMillis="-1" username="sa"

password="admin2$123" driverClassName="com.microsoft.sqlserver.jdbc.SQLServerDriver" url="jdbc:sqlserver://127.0.0.1:PORT;databaseName=TS\_DB"/>

<Resource auth="Container" description="User database that can be updated and saved" factory="org.apache.catalina.users.MemoryUserDatabaseFactory" name="UserDatabase" pathname="conf/tomcat-users.xml" type="org.apache.catalina.UserDatabase"/>

</GlobalNamingResources>

<!-- A "Service" is a collection of one or more "Connectors" that share a single "Container" Note: A "Service" is not itself a "Container", so you may not define subcomponents such as "Valves" at this level. Documentation at /docs/config/service.html

-->

<Service name="Catalina">

<!--The connectors can use a shared executor, you can define one or more named thread pools-->

<!--

<Executor name="tomcatThreadPool" namePrefix="catalina-exec-" maxThreads="150" minSpareThreads="4"/>

-->

<!-- A "Connector" represents an endpoint by which requests are received and responses are returned. Documentation at :

Java HTTP Connector: /docs/config/http.html Java AJP Connector: /docs/config/ajp.html APR (HTTP/AJP) Connector: /docs/apr.html

Define a non-SSL/TLS HTTP/1.1 Connector on port PORT

-->

<Connector connectionTimeout="20000" port="PORT" protocol="HTTP/1.1" redirectPort="PORT"/>

<!-- A "Connector" using the shared thread pool-->

<!--

<Connector executor="tomcatThreadPool" port="PORT" protocol="HTTP/1.1" connectionTimeout="20000" redirectPort="PORT" />

-->

<!-- Define a SSL/TLS HTTP/1.1 Connector on port PORT This connector uses the NIO implementation. The default

SSLImplementation will depend on the presence of the APR/native library and the useOpenSSL attribute of the

AprLifecycleListener.

Either JSSE or OpenSSL style configuration may be used regardless of the SSLImplementation selected. JSSE style configuration is used below.

-->

<!--

<Connector port="PORT" protocol="org.apache.coyote.http11.Http11NioProtocol" maxThreads="150" SSLEnabled="true">

<SSLHostConfig>

<Certificate certificateKeystoreFile="conf/localhost-rsa.jks" type="RSA" />

</SSLHostConfig>

</Connector>

-->

<!-- Define a SSL/TLS HTTP/1.1 Connector on port PORT with HTTP/2 This connector uses the APR/native implementation which always uses OpenSSL for TLS.

Either JSSE or OpenSSL style configuration may be used. OpenSSL style configuration is used below.

-->

<!--

<Connector port="PORT" protocol="org.apache.coyote.http11.Http11AprProtocol" maxThreads="150" SSLEnabled="true" >

<UpgradeProtocol className="org.apache.coyote.http2.Http2Protocol" />

<SSLHostConfig>

<Certificate certificateKeyFile="conf/localhost-rsa-key.pem" certificateFile="conf/localhost-rsa-cert.pem" certificateChainFile="conf/localhost-rsa-chain.pem" type="RSA" />

</SSLHostConfig>

</Connector>

-->

<!-- Define an AJP 1.3 Connector on port PORT -->

<Connector port="PORT" protocol="AJP/1.3" redirectPort="PORT"/>

<!-- An Engine represents the entry point (within Catalina) that processes every request. The Engine implementation for Tomcat stand alone analyzes the HTTP headers included with the request, and passes them on to the appropriate Host (virtual host).

Documentation at /docs/config/engine.html -->

<!-- You should set jvmRoute to support load-balancing via AJP ie :

<Engine name="Catalina" defaultHost="localhost" jvmRoute="jvm1">

-->

<Engine defaultHost="localhost" name="Catalina">

<!--For clustering, please take a look at documentation at:

/docs/cluster-howto.html (simple how to)

/docs/config/cluster.html (reference documentation) -->

<!--

<Cluster className="org.apache.catalina.ha.tcp.SimpleTcpCluster"/>

-->

<!-- Use the LockOutRealm to prevent attempts to guess user passwords via a brute-force attack -->

<Realm className="org.apache.catalina.realm.LockOutRealm">

<!-- This Realm uses the UserDatabase configured in the global JNDI resources under the key "UserDatabase". Any edits

that are performed against this UserDatabase are immediately available for use by the Realm. -->

<Realm className="org.apache.catalina.realm.UserDatabaseRealm" resourceName="UserDatabase"/>

</Realm>

<Host appBase="webapps" autoDeploy="true" name="localhost" unpackWARs="true">

<!-- SingleSignOn valve, share authentication between web applications Documentation at: /docs/config/valve.html -->

<!--

<Valve className="org.apache.catalina.authenticator.SingleSignOn" />

-->

<!-- Access log processes all example.

Documentation at: /docs/config/valve.html

Note: The pattern used is equivalent to using pattern="common" -->

<Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs" pattern="%h %l %u %t &quot;%r&quot; %s %b" prefix="localhost\_access\_log" suffix=".txt"/>

<!-- <Context docBase="/home/michaeledoror/Desktop/apache-tomcat- 9.0.0.M26/wtpwebapps/ts-bookmarking-service" path="/ts-bookmarking-service" reloadable="true" source="org.eclipse.jst.j2ee.server:ts-bookmarking-service"/>--></Host>

</Engine>

</Service>

</Server>

### Jena/Fuseki Setup

This process details installing apache jena and fuseki to the development machines. For simplicity sake, you will download and configure both programs locally and then copy it over to the /usr/share/ folders.

This process will need to differ on the VA machines as apt-get installs will need to be used to manage software.

* + - 1. Install Java. Fuseki runs on java java 1.8. Install 1.8 and make sure the default java version is 1.8

sudo yum install java-1.8.0

sudo yum remove java-1.7.0-openjdk

* + - 1. Download Apache-Jena + Fuseki to home directory

wget <http://www-us.apache.org/dist/jena/binaries/apache-jena-fuseki-3.4.0.tar.gz> tar -xvzf apache-jena-fuseki-3.4.0.tar.gz

* + - 1. Download Apache-Jena to home directory

wget <http://www-us.apache.org/dist/jena/binaries/apache-jena-3.4.0.tar.gz> tar -xvzf apache-jena-3.4.0.tar.gz

* + - 1. Start and stop fuseki to generate run folder and structure cd apache-jena-fuseki-3.4.0/

./fuseki start

* + - 1. Unzip ontologies

tar -xvzf Ontologies\_08182017.tar.gz

* + - 1. Load ALL ontologies into one large default graph (note: database directory path will change later)

~/apache-jena-3.4.0/bin/tdbloader --loc ~/apache-jena-fuseki-3.4.0/run/databases/ MVP\_Master\_Internal\_Ontology\_final\_published.owl NCIT/NCIT\_1706d.rdf HP/HP\_06302017.rdf DOID/DOID\_2017\_0613.rdf CDISC/adam-terminology.rdf CDISC/glossary-terminology.rdf CDISC/sdtm-terminology.rdf CDISC/cdash- terminology.rdf CDISC/qs-terminology.rdf CDISC/send-terminology.rdf ICD10/ICD10CM.ttl ICD9/HOM-ICD9\_04262011.rdf

* + - 1. Configure fuseki to run standalone

vi ~/apache-jena-fuseki-3.4.0/run/shiro.ini

change line 15 from “admin=pw” to “admin=12plaintextpass34” comment out line 25

uncomment line 31

* + - 1. Run fuseki as a standalone service

apache-jena-fuseki-3.4.0/fuseki-server --update --loc=/home/ec2-user/apache-jena-fuseki- 3.4.0/run/databases /ds

Not sure if the next step is necessary, but previous installs were placed in the /usr/share/ directory

* + - 1. Move fuseki and jena to /usr/share/fuseki/ and /usr/share/jena

sudo mkdir /usr/share/jena/

sudo cp -r ~/apache-jena-3.4.0/\* /usr/share/jena/

sudo mkdir /usr/share/fuseki/

sudo cp -r ~/apache-jena-fuseki-3.4.0/\* /usr/share/fuseki/

* + - 1. Add jena and fuseki libraries to PATH sudo touch /etc/profile.d/jena-fuseki.sh

sudo echo “export PATH=$PATH:/usr/share/fuseki/bin:/usr/share/jena/bin” >> jena-

fuseki.sh

* + - 1. Create /etc/default/fuseki file and add fuseki args

sudo touch /etc/default/fuseki

sudo echo “FUSEKI\_ARGS='--update --loc=/usr/share/fuseki/run/databases /ds'” >> fuseki

Notes

To start/stop fuseki:

sudo ./usr/share/fuseki/fuseki start sudo ./usr/share/fuseki/fuseki stop

### SOLR Setup

##### Installation:

* + - 1. Check locations of config files and JSON directory containing files to index
      2. If JAVA\_HOME has been exported, make sure $JAVA\_HOME/bin/java exists
      3. Prep: sudo yum install wget, ruby
      4. Download:wget <http://archive.apache.org/dist/lucene/solr/6.0.0/solr-6.0.0.tgz>
      5. Unpack: tar xzf sor-6.0.0.tgz

##### Create a “fuseki” collection:

1. Set location: cd to where you want solr installed
2. Startup Solr: solr-6.0.0/bin/solr start
3. Create “fuseki” collection: solr-6.0.0/bin/solr create –c fuseki
4. Stop Solr: solr-6.0.0/bin/solr stop

##### Configure “fuseki” collection

1. Copy config files:

cp /path/to/{managed-schema,solrconfig.xml} solr-6.0.0/server/solr/fuseki/conf

1. Restart Solr: solr-6.0.0/bin/solr start

##### Index ontologies:

1. Add JSON files: solr-6.0.0/bin/post –c fuseki /path/to/JSON/\* >& solr-index.output

&

1. Wait about 20 minutes

# Database Setup

## Genisis Data Request Workflows Database Setup

### Database Names

Genisis\_DB is the 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

## 4.2 Genisis Terminology Services Database Setup

### 4.2.1 Database Names

TS\_DB is the application database.

### 4.2.2. Database Schema

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

### List of Tables within the Database

* + - 1. Bookmarks
      2. Concept\_mapping
      3. Concept\_mapping\_data\_elements
      4. Data\_element
      5. Data\_element\_components
      6. Data\_element\_source
      7. Data\_element\_type
      8. Data\_type
      9. Labels
      10. Simple\_data\_element

### Users Recognized by the Database

“genesis\_ts” 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:

TS\_DB

* + - 1. Create\_Database.sql
      2. Create\_User.Sql
      3. Create\_Tables.Sql

# Genisis Data Request Workflow - 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

1. SFTP Release version to Server, in any means accessible by you. SFTP will direct to

/home/<user>/

1. Copy the jar file to /opt/genisisDataOps folder.
   1. sudo cp /home/<user>/ Genisis2DataOps.jar

/opt/genisisDataOps/Genisis2DataOps.jar

1. 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
   1. Make sure that Microsoft Windows 2012 R2 and SQL server 2012 are installed in the Destination and Management databases
   2. Always use Fully qualified Domain Names (FQDNs and not IP addresses)
   3. You need a username and password with the following permissions. This is for routine Genisis2 initiated Table Copy Operations.
   4. ddlread
   5. ddlwrite
   6. 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.
2. Creating a Linked Server from the Management Server (Example: DNS Source Server to the (use FQDN not ip) to the Source Server (Example: DNS SQA Source Server) to Linked Server:
   1. Login to the Management server as an Administrator
   2. Create linked server in the Management for Source Server
   3. 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.
3. Create a Linked Server from the Destination Server to the Management Server (For Ex: DNS Destination Server):
   1. Table Copying is done in two steps
   2. Management Server PULLS the table from the Source Server in to a TEMP database (GENISISTEMPDATA)
   3. Destination Server PULLS the table from the Management Server (GENISISTEMPATA) to the its database (GENISISDATAOPSDATA – This is also the Genisis Landing Zone Database)
   4. You need a username and password with ADMINISTRATOR rights in the Destination Server just for creating the Linked Server)
   5. Create a Linked Server in the Destination Server for the Management Server
   6. Once you see the Linked Management Server, open it and make sure you can see the GENISISTEMPDATA Database
4. 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. Acquire an account on the source server: Coordinate with the source server administrative team to obtain a user name and password created with the 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. The aforementioned permissions are requested; since in the future, this Source Server can serve as a Destination Server also. A table should be able to be created there and the ability to write to it.

## Setting up the Destination Server

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

## Test Reachability Between Servers

1. Management Server to Source Server:
   1. 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.
   2. 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.
2. Destination Server to Management Server:
   1. 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.

# Genisis Data Request Workflow - Upgrading from Build 3 to Build 4

If Build 3 has been installed on the VA servers, follow the steps in Section 6.1 for each server to deploy artifacts.

## Deploy Genisis2 Angular Application

1. Get the appropriate release versions from the dev team for this release.
2. Go to the ur[l: http://genisis20-nexus.boozallencsn.com/nexus/](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. Go to the following url: [http://genisis20-](http://genisis20-/) nexus.boozallencsn.com/nexus/content/repositories/releases/gov/va/genisis2/GenisisAng ularWebApp/<release vesion>
7. Be sure to be off of the VA VPN; the link in the preceding step 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

1. $ unzip <Genisis2Web Zip> /var/www/domain.va.gov/webroot (overwrite the existing files if prompted)
2. Delete contents under help directory $ rm -rf /var/www/domain.va.gov/webroot/help
3. SFTP help.zip file to Server, in any means accessible by you. . SFTP will direct to

/home/<user>/ - so from there unzip the file (overwrite the existing files if prompted)

1. $ unzip <Genisis2Web Help Zip> /var/www/domain.va.gov/webroot/help
2. Restart Apache (service httpd restart)

## Update Genisis2 Databases

Run the following sql script to update GENISIS\_DB.

1) Release4.sql

## Deploy Genisis2 Services Application

1. Before updating the war file, update the database and upgrade the Angular application
2. Open genisis2.properties file using vim and enter correct new properties and its values pertaining to the environment on which the application is being installed.
3. Reach out to the development team for the correct values to be filled for each property listed in the genisis2.properties file. Below are the new properties introduced in Build 4:

# Path of pom.properties of deployed genisis service genisis2.pom.props.path=/META- INF/maven/gov.va.genisis2/Genisis2Services/pom.properties

# LDAP User Refresh cron expression

# every day at 1 AM

#ldap.refresh.cron.expression=0 0 1 \* \* \* ldap.refresh.cron.expression=0 \*/30 \* \* \* \*

# LDAP Context Source ldap.url=ldap://IP:PORT ldap.base=AI ldap.username=AI ldap.password=AI

1. Go to the ur[l: http://genisis20-nexus.boozallencsn.com/nexus/](http://genisis20-nexus.boozallencsn.com/nexus/)
2. Click on login on top right side and enter the following credentials
   1. Username: AI
   2. Password: AI
3. Go to the following url: [http://genisis20-](http://genisis20-/) nexus.boozallencsn.com/nexus/content/repositories/releases/gov/va/genisis2/ Genisis2Services/<release vesion>
4. Be sure to be off of the VA VPN; the link in the preceding step will not work in VA VPN
5. SFTP Release version to Server by any means accessible. SFTP will direct to

/home/<user>/

1. Stop the server by killing the process or if configured as a service, stop the wildfly service
2. Delete the existing war file under /opt/wildfly/standalone/deployments folder
3. Copy the relevant war file under /home/<user>/ to /opt/wildfly/standalone/deployments folder
4. sudo cp /home/<user>/ Genisis2Services.war /opt/wildfly/standalone/deployments
5. Before starting the server, make sure that database scripts are run and property file update is complete
6. Start the wildfly server
7. nohup /opt/wildfly/bin/standalone.sh &