CAN - DEPLOYMENT GUIDE

Version 5.0

JUNE 30, 2020

AVANSEUS TECHNOLOGIES PVT. LTD.

REVISION HISTORY

Version	Date	Change description	Created by	Updated by	Reviewed by
V 1.0	April, 2017	Initial release	Chandan B.K	Sheenginee B	Chiranjib
V 2.0	March, 2018	Updated with the latest software versions	Chandan B.K	Sheenginee B	Chiranjib
V 3.0	June, 2018	Updated with few command changes	Chandan B.K	Chandan B.K	Chiranjib
V 4.0	July, 2019	Updated with formatting changes	Sandeep	Sandeep	Chiranjib
V 5.0	June, 2020	Updated with few command changes	Naveen/Umesh	Sandeep	Chiranjib

avanseus

TABLE OF CONTENTS

1. OVERVIEW

This document provides a detailed step-by-step approach to setup an application and data server for CAN application deployment and functioning. This installation guide is prepared for RHEL **7.x** OS.

There are basically 2 types of application deployment possible:

- Deployment in basic VM – This document focuses on how to deploy CAN application in basic VM machine.
- Containerized deployment – For containerized deployment please refer documentation on Docker and Kubernetes deployment which is provided separately. Deployment team can refer this document for configuration, if needed.

2. HARDWARE REQUIREMENTS

Hardware requirements are decided based on the number of physical sites for a particular customer. The Hardware Specifications document will be shared in a separate excel document.

3. SOFTWARE REQUIREMENTS

Once the hardware is procured and the decision is taken to categorize the machines as app servers and database servers, we will need the following installations on the app and DB servers.
Kindly note that for any customer POC/live trial/live installation, only one instance of Apache web server, LDAP and MongoDB is required. However, multiple tomcats can be configured based on the size of prediction.

Application Server	Database Server
Java 8	Java 8
Apache Tomcat 8	MongoDB
Apache web server	
LDAP	
Ports to be opened	
NAS mount	

4. PREREQUISITES TOOLS DOWNLOAD

Login as root and download wget using the command:

yum install wget

Install fuser
yum install psmisc

Install unzip
yum install unzip

5. JAVA 8 INSTALLATION

Login as root, download and install Java 8

- Download the latest Java SE Development Kit 8 release using the following commands to download from shell and extract the installable to /opt/ directory.

cd /opt/
wget https://s3.ap-south-1.amazonaws.com/avanseusinstallationfiles/jdk-8u131-linux-x64.tar.gz
tar -xzvf jdk-8u131-linux-x64.tar.gz

- After extracting archive file, use alternatives command to install it.

cd /opt/jdk1.8.0_131/
alternatives --install /usr/bin/java java /opt/jdk1.8.0_131/bin/java 2
alternatives --config java

Enter to keep the current selection [+], or type the selection number: <Choose the latest one>

- At this point JAVA 8 has been successfully installed on your system. We also recommend to setup javac and jar commands path using alternatives (words on RHEL):

alternatives --install /usr/bin/jar jar /opt/jdk1.8.0_131/bin/jar 2
alternatives --install /usr/bin/javac javac /opt/jdk1.8.0_131/bin/javac 2
alternatives --set jar /opt/jdk1.8.0_131/bin/jar
alternatives --set javac /opt/jdk1.8.0_131/bin/javac

For Ubuntu, use:
update-alternatives --install /usr/bin/java java /opt/jdk1.8.0_131/bin/java 2
update-alternatives --config java
update-alternatives --install /usr/bin/jar jar /opt/jdk1.8.0_131/bin/jar 2
update-alternatives --install /usr/bin/javac javac /opt/jdk1.8.0_131/bin/javac 2
update-alternatives --set jar /opt/jdk1.8.0_131/bin/jar
update-alternatives --set javac /opt/jdk1.8.0_131/bin/javac

- Check the installed Java version on your system using the following command:

java -version

Output:
java version "1.8.0_121"
Java(TM) SE Runtime Environment (build 1.8.0_121-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.121-b13, mixed mode, sharing)

6. VBI DEPLOYMENT (PYTHON INSTALLATION)

Prerequisites:

- Move the "VBIFiles" folder (present under can/MasterTables/) to <%naspath%> (ignore if already it has been moved).
- Import VBIAnnotations.json and Config.json (present under can/MasterTables/) to database (ignore if already it has been imported).
- Check if Python version 3.7.3 is installed.
- Use the command below command to check python version
  /usr/bin/python3 -V
- If Python version 3.7.3 is already installed, follow the below steps:
  - Check whether pip is installed in the system with the command:
    pip list

- If it is not installed, then to install, use the command:
  wget https://bootstrap.pypa.io/get-pip.py
- Run the command:
  python get-pip.py

  pip install nltk torch numpy scipy --no-cache-dir
  cd <%naspath%>/VBIFiles/VBIPythonFiles/

- To download the zip folder from S3 bucket, use the command:
  wget https://devstoragedata.s3.ap-south-1.amazonaws.com/Libraries/bert-base-uncased.tar.gz

- Unzip the folder using the command:
  tar -xzf bert-base-uncased.tar.gz
  (*** Make sure that the unzipped folder "bert-base-uncased" is under the same path where python server file "vbiServer.py" is there ***)

- Run the command:
  nohup python vbiServer.py 12001 &

o If other Version of python is installed, find the installation process for python version 3.7.3 below:
  - Login as root and run the command:

    yum install gcc openssl-devel bzip2-devel libffi-devel sqlite-devel

    mkdir src

    (*** The folder name "src" is not mandatory, you can provide any other name ***)

    cd src
  - Download the required python version:

    wget *https://www.python.org/ftp/python/3.7.3/Python-3.7.3.tgz*
  - Extract the downloaded file and follow the steps given below:

    tar xzf Python-3.7.3.tgz

    cd Python-3.7.3

    ./configure --enable-optimizations

    make altinstall

    rm /src/Python-3.7.3.tgz

    cd

    mkdir PythonEnv

    (*** The folder name "PythonEnv" is not mandatory, you can provide any name ***)

  - Check whether pip is installed in the system with the command:

    pip list
  - If it is not installed, then to install, use the command:

    wget *https://bootstrap.pypa.io/get-pip.py*
  - Run the command:

    python get-pip.py

    cd PythonEnv

    ../src/Python3.7.3/python -m venv canVBI_env

(*** The folder name “canVBI_env” is not mandatory, you can configure any name ***)

source canVBI_env/bin/activate

cd

pip install nltk torch numpy scipy --no-cache-dir

cd <%naspath%>/VBIFiles/VBIPythonFiles/

- To download the zip folder from S3 bucket Use the command:
  wget https://devstoragedata.s3.ap-south-1.amazonaws.com/Libraries/bert-base-uncased.tar.gz.
- Unzip the folder using the command:
  tar -xzf bert-base-uncased.tar.gz
  (*** Make sure that the unzipped folder “bert-base-uncased” is under the same path where python server file “vbiServer.py” is there ***)
- Run the command:
  nohup python vbiServer.py 12001 &

Further “avanseus.vbi.ip” & “avanseus.vbi.port” are to be added to the config.properties file the process is mentioned in the tomcat installation process.
Make sure that “avanseus.vbi.port” & the port used for python server (here, 12001) are same, otherwise the default port of python server file is 12003.

7. APACHE TOMCAT INSTALLATION

- Download the Apache tomcat 8

  wget https://s3.ap-south-1.amazonaws.com/avanseusinstallationfiles/apache-tomcat-8.5.16.tar.gz
- Extract the tar.gz file

  tar -xzxvf apache-tomcat-8.5.16.tar.gz

- Configure the [server port, HTTP connector port and AJP connector port] in the tomcat's server.xml file. Kindly note that [server port, HTTP connector port and AJP connector port] must be unique for each tomcat instance on a particular machine.
- Configure 3 such tomcats as follows:

For CAN deployment:

```
<Server port="8005" shutdown="SHUTDOWN">
<Connector port="8080" protocol="HTTP/1.1" connectionTimeout="20000" redirectPort="8443" />
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
```

For CAS deployment:

```
<Server port="8006" shutdown="SHUTDOWN">
<Connector port="8081" protocol="HTTP/1.1" connectionTimeout="20000" redirectPort="8443" />
<Connector port="8010" protocol="AJP/1.3" redirectPort="8443" />
```

- Load the config.properties files onto each tomcat home directory. Sample config.properties file will be sent separately in a word file.

Explanation of config.properties attributes:

- avanseus.app.<module>.domain - Domain address/IP address to access the module. This is the public IP address. No need to configure the Port as web server will take care of the configuration.
- avanseus.protocol - Domain protocol or access protocol.
- avanseus.app.cas.host - Host IP address and port. This is the private IP address and port. In case where private IP address is not assigned, this IP is same as the public IP address.
- avanseus.mongodb.ip - IP of the host where MongoDB is installed.
- avanseus.mongodb.port - MongoDB access port
- avanseus.mongodb.username - MongoDB authentication username
- avanseus.mongodb.password - MongoDB authentication password
- avanseus.mongodb.dbName - MongoDB access schema
- avanseus.mongodb.admin.username - MongoDB admin authentication username
- avanseus.mongodb.admin.password – MongoDB admin authentication password
- avanseus.mongodb.admin.dbName - MongoDB admin access schema
- avanseus.log.path - Log file path for application to write the logs
- avanseus.log.thread.poolsize - Logger thread pool size
- avanseus.ldap.organisation - LDAP organization config.
- avanseus.ldap.domain - LDAP domain config.
- avanseus.ldap.url1 - LDAP URL
- avanseus.ldap.url2 - LDAP URL
- avanseus.ldap.userDn - LDAP directory manager config.
- avanseus.ldap.password - LDAP password
- avanseus.app.nas.path - CAN NAS mount location
- avanseus.email.fromId - Email username
- avanseus.email.pwd - Email password
- avanseus.email.host - Email hostname
- avanseus.email.fromName - Email from name
- avanseus.email.port - Email port
- avanseus.app.googleKey - Google maps API key for loading maps
- avanseus.predictionNode.name- CAN node name
- avanseus.mongodb.admin.dbName= MongoDB access schema admin
- avanseus.mongodb.admin.username=MongoDB authentication admin
- avanseus.mongodb.admin.password=MongoDB authentication admin password (encrypted)
- avanseus.vbi.ip=IP of the host where Python is installed (Internal ip or 127.0.0.1)
- avanseus.vbi.port=VBI access port (port no. 12001)

There are two types of CAN installation:

1. Single instance
2. Cluster setup.

In a single instance, avanseus.predictionNode.name can be NodeA. In case of cluster setup, avanseus.predictionNode.name can be NodeA, NodeB, NodeC etc based on the number of CAN instances.

- Copy the CAN/CAS/GSC war file to the webapps folder inside tomcat.
- Start the tomcat from within bin directory of the tomcat:

  ./catalina.sh start

8. TOMCAT SECURITY CONFIGURATION

1. Password Encryption/Decryption changes

We store the encryption key which is used for the encryption/decryption of plain text passwords stored in DB and config.properties. This encryption key has to be stored in tomcat's catalina.properties file.

Go to the tomcat folder where it is installed.

- Go to the conf folder under the tomcat directory.
- Open the catalina.properties file using vi editor and add the below lines

  PASSWORD_KEY=<encryption_key>

Config.properties changes:
Following property values in config.properties file have to be encrypted and put.
avanseus.mongodb.password
avanseus.mongodb.admin.password
avanseus.ldap.password

Note: For Encryption/Decryption of password a document has been created. please follow the document for the encryption/decryption of passwords.

2. Java Security Manager changes
   Step 1: Get security.policy file from git repository.
   Step 2: Place security.policy file inside <tomcat.home>/conf/ folder.
   Step 3: If the setenv.sh file already exists in <tomcat.home>/bin/ folder add following lines to setenv.sh, else create setenv.sh file and add following lines.
   export JAVA_OPTS=" $JAVA_OPTS -Dapplication.nasmount=<nasmount.path>";
   export JAVA_OPTS=" $JAVA_OPTS -Dmongo.host=<ip address> -Dmongo.port=<port>";
   export JAVA_OPTS=" $JAVA_OPTS -Djava.security.manager -
   Djava.security.policy=<tomcat.home>/conf/security.policy";
   export TOMCAT_OPTS=" $TOMCAT_OPTS -Djava.security.debug=access,failure";

   Note: Do necessary changes by replacing <nasmount.path> with CAN nasmount path, <ip address> ip address of the server where mongodb installed, <port> the port at which mongodb is running and <tomcat.home> the path where tomcat is installed.

3. Cookie Configuration changes:
   Go to the tomcat folder where it is installed.
   - Go to the conf folder under the tomcat directory.
   - Open the web.xml file using vi editor and add the below lines under <session-config>

   <tracking-mode>COOKIE</tracking-mode>
   <cookie-config>
   <name>id</name>
   <path>/<application_name>/</path>
   </cookie-config>

   Note: For Path attribute <application_name> write respective application name eg: for CAN it will be <path>/CAN/</path>

9. PORTS TO BE OPENED

The following inbound traffic ports needs to be open on the firewall. Based on the configurations of the tomcats, web server and LDAP, the ports may change.

Port	Use
27017	MongoDB default port
1389	For LDAP use

4444	For LDAP use
80	HTTP default port
22	SSH default port
8009	AJP connector port for CAN
8010	AJP connector port for CAS
8011	AJP connector port for GSC
8065-8095	For tomcat clustering and load balancing

10. NAS MOUNT

NAS mount can be created on the application server and the path needs to be provided as a configuration in the config.properties file.

11. MONGODB INSTALLATION

On the database server, login as deployment user and install the MongoDB. Follow the steps below:

- Download the installation file as follows:

  wget https://s3.ap-south-1.amazonaws.com/avanseusinstallationfiles/mongodb-linux-x86_64-3.4.6.tgz

- Untar the package as follows:

  tar –xzvf mongodb-linux-x86_64-3.4.6.tgz

- Create a data path and log path to store database data and database logs. Make a note of the paths.
- Download the config file using and perform the following edits:

  wget https://s3.ap-south-1.amazonaws.com/avanseusinstallationfiles/mongod.conf

- Edit systemLog path to the log path created above.
- Edit dbPath to the data path created above.
- Edit pidFilePath to the data path created above
- Start the mongo server using:

Set Path in bashrc file.

vi ~/.bashrc

export PATH= $PATH:/home/dbuser/mongodb-linux-x86_64-3.4.6/bin

Please edit the file “mongod.conf” and uncomment or add below lines

 # where to write logging data.

systemLog:

destination: file

logAppend: true

path: <mongo_installation_path>/mongologs/mongod.log

```
# Where and how to store data.
    Storage:
      dbPath: <mongo_installation_path>/mongoDB/mongodata/
      journal:
        enabled: true

# how the process runs
    processManagement:
      fork: true  # fork and run in background
      pidFilePath: <mongo_installation_path>/mongoDB/mongodata/mongod.pid  # location of pidfile

# network interfaces
    net:
      port: 27017
      bindIp: 0.0.0.0

#security:
    security:
      authorization: enabled
     ./mongod –config=<path to mongod.conf>
```

- Use the admin db and create a admin user with admin access. (Share the admin credentials with the project manager only).

```
use admin
db.createUser({user: "admin", pwd: “Avanseus$0", roles: ["userAdminAnyDatabase", "readWriteAnyDatabase", "dbAdminAnyDatabase", "clusterAdmin"]});
exit
```

- Connect using admin user:

```
mongo localhost:27017/admin -u admin -p '<db password>
```

- Create a schema for deployment and a user to access the schema (please ensure to note the username and password to share with development team).

```
use <customer name>
db.createUser({user: "<customer name as user name>", pwd: “<any complex password here>", roles: ["userAdmin", "dbAdmin", "readWrite"]});
```

- You can connect to admin DB and deployment DB as follows:

```
mongo localhost:27017/admin -u admin -p '<admin password>’
mongo localhost:27017/ deploymentdb -u deploymentdb -p '<deployment user password>'
```

12. HTTPD INSTALLATION

Prerequisites:
Three Certificate files are required

1. Server Certificate
2. Server Private Key
3. Certificate Authority (CA)

Domain name required

- Domain name

Login as root to do all operations.
Install the below packages using command:

yum install httpd httpd-devel gcc
#modssl Module for https
yum -y install mod_ssl

13. MOD_JK INSTALLATION

- This is a tomcat connector package which helps in load balancing or masking the tomcat servers from outside world.
- This package needs to be installed by compiling source code.
- Download the connector package from

  wget https://s3.ap-south-1.amazonaws.com/avanseusinstallationfiles/tomcat-connectors-1.2.42-src.tar.gz
- Extract the package with command: *tar -xzvf tomcat-connectors-1.2.42-src.tar.gz*
- cd tomcat-connectors-1.2.42-src/native/
- Run the script file with parameters: ./configure --with-apxs=/usr/bin/apxs
- Once the script runs successfully, use the command “make” in shell prompt to build the module.
- Finally, install the module using command “make install” in shell prompt. This will install the MOD-JK binary/library file mod-jk.so in HTTP Web server.

14. HTTP SERVER WORKER CONFIGURATION

- Create a file: vi /etc/httpd/conf/workers.properties
- Copy the below content in the file and save.

worker.list=canlb, caslb.

worker.canlb.port=8050
worker.canlb.host=<Internal IP>
worker.canlb.type=ajp13

worker.caslb.port=8053
worker.caslb.host=<Internal IP>
worker.caslb.type=ajp13

worker.gsclb.port=8054
worker.gsclb.host=<Internal IP>
worker.gsclb.type=ajp13

In above content, <Internal IP> should be replaced with the Internal IP of the machine where that tomcat is installed.

15. HTTP SERVER URL CONTEXT CONFIGURATION

Open file: vi /etc/httpd/conf/httpd.conf
Copy the below content anywhere in the file and save:

#Mod_jk
LoadModule jk_module modules/mod_jk.so
JkWorkersFile /etc/httpd/conf/workers.properties

<IfModule mod_autoindex.c>
AddIcon /etc/httpd/images/parent.png ..
*AddIcon /etc/httpd/images/zip.png *.zip*
*AddIcon /etc/httpd/images/xls.png *.xlsx *.xls*
*AddIcon /etc/httpd/images/png.png *.png*
*AddIcon /etc/httpd/images/jpg.png *.jpg*
</IfModule>

16. HTTP SERVER HEADER AND SSL CONFIGURATION

Open the ssl.conf file and add the below mentioned Configuration

Replace SSLCertificateFile, SSLCertificateKeyFile and SSLCertificateChainFile with respective file paths

SSLCertificateFile <file_path>

SSLCertificateKeyFile <file_path>

SSLCertificateChainFile <file_path>

Note: <file_path> is the path to the respective files where they are stored.

Add the below lines just before the <VirtualHost _default_:443>

#http virtual host configurations
*<VirtualHost *:80>*
Redirect Permanent /CAN https://<domain_name>/CAN
Redirect Permanent /CAS https://<domain_name>/CAS
RewriteEngine on
RewriteCond
%{REQUEST_METHOD}^(TRACE|TRACK|MOVE|PROFIND|OPTIONS|HEAD|DELETE|PUT)
RewriteRule .$ - [F,L]*
</VirtualHost>

Note: <domain_name> should be the domain name of the server.

SSL Protocol setting: Set the SSLProtocol field as shown below
SSLProtocol -all +TLSv1.2

SSL Cipher and Cipher Order Setting: Set the SSLCipherSuite and SSLHonorCipherOrder field as shown below
SSLCipherSuite
ALL:!RSA:!CAMELLIA:!aNULL:!eNULL:!LOW:!3DES:!MD5:!EXP:!PSK:!SRP:!DSS:!RC4:!SHA1:!SHA256:!SHA384
SSLHonorCipherOrder on

Add the below lines in the <VirtualHost _default_:443>.....</VirtualHost> section in ssl.conf folder
#Mod_jk
JkMount /CAN/ canlb*
JkMount /CAS/ caslb*

#Cache headers
<IfModule mod_mime.c>

```
    AddType text/css .css
    AddType application/x-javascript .js
    AddType text/x-component .htc
    AddType text/html .html .htm
    AddType text/richtext .rtf .rtx
    AddType image/svg+xml .svg .svgz
    AddType text/plain .txt
    AddType text/xsd .xsd
    AddType text/xsl .xsl
    AddType text/xml .xml
    AddType video/asf .asf .asx .wax .wmv .wmx
    AddType video/avi .avi
    AddType image/bmp .bmp
    AddType application/java .class
    AddType video/divx .divx
    AddType application/msword .doc .docx
    AddType application/vnd.ms-fontobject .eot
    AddType application/x-msdownload .exe
    AddType image/gif .gif
    AddType application/x-gzip .gz .gzip
    AddType image/x-icon .ico
    AddType image/jpeg .jpg .jpeg .jpe
    AddType application/vnd.ms-access .mdb
    AddType audio/midi .mid .midi
    AddType video/quicktime .mov .qt
    AddType audio/mpeg .mp3 .m4a
    AddType video/mp4 .mp4 .m4v
    AddType video/mpeg .mpeg .mpg .mpe
    AddType application/vnd.ms-project .mpp
    AddType application/x-font-otf .otf
    AddType application/vnd.oasis.opendocument.database .odb
    AddType application/vnd.oasis.opendocument.chart .odc
    AddType application/vnd.oasis.opendocument.formula .odf
    AddType application/vnd.oasis.opendocument.graphics .odg
    AddType application/vnd.oasis.opendocument.presentation .odp
    AddType application/vnd.oasis.opendocument.spreadsheet .ods
    AddType application/vnd.oasis.opendocument.text .odt
    AddType audio/ogg .ogg
    AddType application/pdf .pdf
    AddType image/png .png
    AddType application/vnd.ms-powerpoint .pot .pps .ppt .pptx
    AddType audio/x-realaudio .ra .ram
    AddType application/x-shockwave-flash .swf
    AddType application/x-tar .tar
    AddType image/tiff .tif .tiff
    AddType application/x-font-ttf .ttf .ttc
    AddType audio/wav .wav
    AddType audio/wma .wma
    AddType application/vnd.ms-write .wri
    AddType application/vnd.ms-excel .xla .xls .xlsx .xlt .xlw
    AddType application/zip .zip
</IfModule>

<IfModule mod_expires.c>
 ExpiresActive On
 ExpiresDefault "access plus 1 seconds"
 ExpiresByType image/gif "access plus 120 minutes"
 ExpiresByType image/jpeg "access plus 120 minutes"
 ExpiresByType image/png "access plus 120 minutes"
 ExpiresByType text/css "access plus 60 minutes"
 ExpiresByType text/javascript "access plus 60 minutes"
```



```
ExpiresByType application/javascript "access plus 60 minutes"
ExpiresByType application/x-javascript "access plus 60 minutes"
ExpiresByType text/xml "access plus 60 minutes"
</IfModule>

# No caching for dynamic files
<LocationMatch "\.(htm|html||jsp)$ ">
    ExpiresDefault A0
    Header set Cache-Control "no-store, no-cache, must-revalidate, max-age=0"
    Header set Pragma "no-cache"
</LocationMatch>

#Cookie settings
Header edit Set-Cookie ^(.*)$  $ 1;Secure;SameSite=Strict;
#HSTS settings
Header always set Strict-Transport-Security "max-age=10886400; includeSubDomains"
#Clickjacking attack settings
Header always append X-Frame-Options SAMEORIGIN

# restricting http method configurations
RewriteEngine on
RewriteCond %{REQUEST_METHOD}
^(TRACE|TRACK|MOVE|PROFIND|OPTIONS|HEAD|DELETE|PUT)
RewriteRule .*$  - [F,L]

#http request size configurations
<Location "/">
LimitRequestBody 10000000
</Location>
```

Once all the above configurations are done, the installation procedure is complete.

Disable SELINUX and reboot the system

```
vi /etc/selinux/config
```

Set SELINUX=disabled

```
reboot
```

17. START OR STOP HTTP WEB SERVER

To start webserver use the command: apachectl start
To stop webserver use the command: apachectl stop
To restart webserver use the command: apachectl restart

18. LDAP SERVER INSTALLATION

For Ubuntu:

1. Pre-installation mandate:
    - Create a ldap user: useradd ldap
    - Su -l ldap
    - Download Java version 1.6 and install it.
    - Download jdk 1.6 using the command: *wget* https://s3.ap-south-1.amazonaws.com/avanseusinstallationfiles/jdk-6u45-linux-x64.bin
    - Assign permissions using the command: *chmod a+x jdk-6u45-linux-x64.bin*
    - Inflate using the command: *./jdk-6u45-linux-x64.bin*
2. During installation:

- Download OpenDS zip file using the command: *wget* https://s3.ap-south-1.amazonaws.com/avanseusinstallationfiles/OpenDS-2.2.1.zip
- To unzip the zip file, run the command: *unzip OpenDS-2.2.1.zip*
- Cd OpenDS-2.2.1
- Under OpenDS folder, run the command:*./setup --cli*
- The initial root user DN for the DirectoryServer? [cn=Directory Manager] will be kept as it is.
- Set the password for the LDAP Directory Server. For example: abc123$
- Set the Listener port as required. For example: 1389
- Set the port for Administration Connector as required. For example: 4444
- Set the Directory Base DN as *dc=avanseus, dc=com*.
- Enter Choice 1 for creating Base Entry.
- Enable SSL.
- Set the port for Directory Server to accept connections from LDAPS clients. For example: 1636.
- Enable Start TLS
- Enter Choice 1 to generate self-signed certificate.
- Select 'Yes' to start the server after completion of setup.
- Select Choice 1 to setup the server with the parameters above.
- Under OpenDS folder, open 'config' folder and select to open java.properties file. Command: *vi java.properties*
- Set the variable *default.java-home=JRE*_path. For example: */home/ldap/jdk1.6.0_45/jre*
- Save the file. Command: *Esc:wq*
- Under OpenDS folder, open the 'bin' folder under which, run *./dsjavaproperties* command.

3. Post installation:

- *wget* https://s3.ap-south-1.amazonaws.com/avanseusinstallationfiles/setup.ldif
- *./import*-ldif -b dc=avanseus,dc=com -n userRoot -l setup.ldif
- To start OpenDS, go to 'bin' folder under OpenDS folder and run the command *./start-ds*
- To stop go to 'bin' folder, run the command *./stop-ds*.

-END OF DOCUMENT-