v3.3 Installing dependencies

1. Installing Nginx

1.1 Introduction

This document describes the process to be followed for upgrading Nginx with Openssl, Modsecurity and more Nginx modules.

1.2 Prerequisite

Below RPM packages required to be installed to compile the Nginx v1.18.0

```
        git build-essential libssl-dev libtool autoconf apache2-
prefork-dev libxml2-dev
        libcurl4-openssl-dev
        wget
        gcc make automake libtool
        pcre pcre-devel libxml2 libxml2-devel curl curl-devel httpd-
devel
        libpcre3 libpcre3-dev
        yajl
        yajl-devel
        gcc-c++ zlib-devel openssl-devel
        unzip
```

1.3 Nginx Installation Guide

Follow the steps below to install Nginx

Note: This guide assumes that `/usr/local/nginx` is the home directory of Nginx installation

1. The following dependencies must be installed on wherever the Nginx is required to deploy with the root access.

```
$ mkdir /dmapim/downloads
$ cd /dmapim/downloads
```

Copy all the relevant pre requisite rpm packages to this folder

Install the RPM dependencies

Note and observation:

Before executing the rpm commands, kindly read the below notes/observations specified and understand carefully.

1. For any of the below commands, there is a possibility of below issue(s) and please follow the solution and execute again

error: Failed dependencies:

Examples:

```
 libxml2 = 2.9.1-6.el7_2.3 is needed by (installed) libxml2-python-
2.9.1-6.el7_2.3.x86_64

 httpd = 0:2.4.6-89.el7.centos is needed by (installed) mod_ssl-1:2.4.6-
89.el7.centos.x86_64
```

Solution:

We can reutilize the below command for any dependency failure to be deleted and reinstalled

```
rpm -e –nodeps <installed rpm package name> rpm -e --nodeps libxml2-
2.9.1-6.el7_2.3.x86_64
Example: rpm -e --nodeps mod_ssl-1:2.4.6-89.el7.centos.x86_64
```

2. Remove the already installed rpm and re-execute the package which causing conflict issue(s) like below

```
file /usr/share/gcc-4.8.2/python/libstdcxx/v6/printers.pyc from install
of libstdc++-4.8.5-44.el7.x86_64 conflicts with file from package
libstdc++-4.8.5-36.el7.i686

file /usr/share/gcc-4.8.2/python/libstdcxx/v6/printers.pyo from install
of libstdc++-4.8.5-44.el7.x86_64 conflicts with file from package
libstdc++-4.8.5-36.el7.i686

file /usr/bin/lz4c from install of lz4-1.8.3-1.el7.x86_64 conflicts
with file from package lz4-1.7.5-2.el7.i686
```

Solution:

```
rpm -e --nodeps libstdc++-4.8.5-36.el7.i686
rpm -e --nodeps lz4-1.7.5-2.el7.i686
```

3. There are possible console output says few packages already installed or Newer version is installed for any rpm -Uvh commands but the command may not install all the packages specified in the command, hence recommended to modify the rpm command by removing the already installed packages from the command and re-execute the modified command.

Execute the below commands:

```
 $ rpm -Uvh perl-Carp-1.26-244.el7.noarch.rpm perl-Encode-2.51-7.el7.
x86_64.rpm perl-Error-0.17020-2.el7.noarch.rpm perl-Exporter-5.68-3.el7.
noarch.rpm perl-File-Path-2.09-2.el7.noarch.rpm perl-File-Temp-0.23.01-
3.el7.noarch.rpm perl-Filter-1.49-3.el7.x86_64.rpm perl-Getopt-Long-
2.40-3.el7.noarch.rpm perl-HTTP-Tiny-0.033-3.el7.noarch.rpm perl-Git-
1.8.3.1-23.el7_8.noarch.rpm perl-PathTools-3.40-5.el7.x86_64.rpm perl-
Pod-Escapes-1.04-295.el7.noarch.rpm perl-Pod-Perldoc-3.20-4.el7.noarch.
rpm perl-Pod-Simple-3.28-4.el7.noarch.rpm perl-Pod-Usage-1.63-3.el7.
noarch.rpm perl-Scalar-List-Utils-1.27-248.el7.x86_64.rpm perl-Socket-
2.010-5.el7.x86_64.rpm perl-Storable-2.45-3.el7.x86_64.rpm perl-
TermReadKey-2.30-20.el7.x86_64.rpm perl-Text-ParseWords-3.29-4.el7.
noarch.rpm perl-Time-HiRes-1.9725-3.el7.x86_64.rpm perl-Time-Local-
1.2300-2.el7.noarch.rpm perl-constant-1.27-2.el7.noarch.rpm git-1.8.3.1-
23.el7_8.x86_64.rpm perl-libs-5.16.3-295.el7.x86_64.rpm perl-macros-
5.16.3-295.el7.x86_64.rpm perl-parent-0.225-244.el7.noarch.rpm perl-
podlators-2.5.1-3.el7.noarch.rpm perl-threads-1.87-4.el7.x86_64.rpm
perl-threads-shared-1.43-6.el7.x86_64.rpm perl-5.16.3-295.el7.x86_64.
rpm git-1.8.3.1-23.el7_8.x86_64.rpm

 $ rpm -Uvh cpp-4.8.5-39.el7.x86_64.rpm glibc-2.17-307.el7.1.x86_64.rpm
gcc-4.8.5-39.el7.x86_64.rpm glibc-devel-2.17-307.el7.1.x86_64.rpm glibc-
headers-2.17-307.el7.1.x86_64.rpm libgcc-4.8.5-39.el7.x86_64.rpm
libgomp-4.8.5-39.el7.x86_64.rpm libmpc-1.0.1-3.el7.x86_64.rpm mpfr-
3.1.1-4.el7.x86_64.rpm glibc-common-2.17-307.el7.1.x86_64.rpm kernel-
headers-3.10.0-1127.19.1.el7.x86_64.rpm

 $ rpm -Uvh automake-1.13.4-3.el7.noarch.rpm autoconf-2.69-11.el7.
noarch.rpm libtool-2.4.2-22.el7_3.x86_64.rpm m4-1.4.16-10.el7.x86_64.
rpm perl-Data-Dumper-2.145-3.el7.x86_64.rpm perl-Test-Harness-3.28-3.
el7.noarch.rpm perl-Thread-Queue-3.02-2.el7.noarch.rpm

 $ rpm -Uvh wget-1.14-18.el7_6.1.x86_64.rpm

 $ rpm -Uvh make-3.82-24.el7.x86_64.rpm

 $ rpm -Uvh pcre-8.32-17.el7.x86_64.rpm pcre-devel-8.32-17.el7.x86_64.
rpm libxml2-2.9.1-6.el7.5.x86_64.rpm xz-5.2.2-1.el7.x86_64.rpm xz-libs-
5.2.2-1.el7.x86_64.rpm xz-devel-5.2.2-1.el7.x86_64.rpm zlib-1.2.7-18.
el7.x86_64.rpm zlib-devel-1.2.7-18.el7.x86_64.rpm libxml2-devel-2.9.1-6.
```

```
el7.5.x86_64.rpm

 $ rpm -Uvh libcurl-7.29.0-59.el7.x86_64.rpm curl-7.29.0-59.el7.x86_64.
rpm libssh2-1.8.0-4.el7.x86_64.rpm nss-pem-1.0.3-7.el7.x86_64.rpm nspr-
4.25.0-2.el7_9.x86_64.rpm nss-softokn-3.53.1-6.el7_9.x86_64.rpm nss-
3.53.1-3.el7_9.x86_64.rpm nss-softokn-freebl-3.53.1-6.el7_9.x86_64.rpm
nss-sysinit-3.53.1-3.el7_9.x86_64.rpm nss-util-3.53.1-1.el7_9.x86_64.
rpm nss-tools-3.53.1-3.el7_9.x86_64.rpm libcurl-devel-7.29.0-59.el7.
x86_64.rpm apr-1.4.8-7.el7.x86_64.rpm apr-devel-1.4.8-7.el7.x86_64.rpm
apr-util-1.5.2-6.el7.x86_64.rpm cyrus-sasl-2.1.26-23.el7.x86_64.rpm apr-
util-devel-1.5.2-6.el7.x86_64.rpm cyrus-sasl-lib-2.1.26-23.el7.x86_64.
rpm cyrus-sasl-devel-2.1.26-23.el7.x86_64.rpm expat-2.1.0-12.el7.x86_64.
rpm expat-devel-2.1.0-12.el7.x86_64.rpm httpd-devel-2.4.6-95.el7.centos.
x86_64.rpm httpd-tools-2.4.6-95.el7.centos.x86_64.rpm libdb-5.3.21-25.
el7.x86_64.rpm libdb-devel-5.3.21-25.el7.x86_64.rpm libdb-utils-5.3.21-
25.el7.x86_64.rpm mailcap-2.1.41-2.el7.noarch.rpm openldap-2.4.44-22.
el7.x86_64.rpm httpd-2.4.6-95.el7.centos.x86_64.rpm openldap-devel-
2.4.44-22.el7.x86_64.rpm

 $ rpm -Uvh yajl-2.0.4-4.el7.x86_64.rpm

 $ rpm -Uvh yajl-devel-2.0.4-4.el7.x86_64.rpm

 $ rpm -Uvh cpp-4.8.5-44.el7.x86_64.rpm gcc-4.8.5-44.el7.x86_64.rpm
libgcc-4.8.5-44.el7.x86_64.rpm libgomp-4.8.5-44.el7.x86_64.rpm

 $ rpm -Uvh libstdc++-4.8.5-44.el7.x86_64.rpm libstdc++-devel-4.8.5-44.
el7.x86_64.rpm gcc-c++-4.8.5-44.el7.x86_64.rpm

 $ rpm -Uvh cryptsetup-libs-2.0.3-6.el7.x86_64.rpm e2fsprogs-libs-
1.42.9-19.el7.x86_64.rpm keyutils-libs-devel-1.5.8-3.el7.x86_64.rpm
e2fsprogs-1.42.9-19.el7.x86_64.rpm krb5-devel-1.15.1-50.el7.x86_64.rpm
libcom_err-1.42.9-19.el7.x86_64.rpm libcom_err-devel-1.42.9-19.el7.
x86_64.rpm libkadm5-1.15.1-50.el7.x86_64.rpm libgudev1-219-78.el7_9.2.
x86_64.rpm krb5-libs-1.15.1-50.el7.x86_64.rpm libselinux-2.5-15.el7.
x86_64.rpm libselinux-devel-2.5-15.el7.x86_64.rpm libselinux-python-2.5-
15.el7.x86_64.rpm libselinux-utils-2.5-15.el7.x86_64.rpm libsemanage-
2.5-14.el7.x86_64.rpm libsemanage-python-2.5-14.el7.x86_64.rpm libsepol-
devel-2.5-10.el7.x86_64.rpm libss-1.42.9-19.el7.x86_64.rpm libsepol-2.5-
10.el7.x86_64.rpm libverto-devel-0.2.5-4.el7.x86_64.rpm lz4-1.8.3-1.el7.
x86_64.rpm openssl-1.0.2k-19.el7.x86_64.rpm openssl-devel-1.0.2k-19.el7.
x86_64.rpm openssl-libs-1.0.2k-19.el7.x86_64.rpm policycoreutils-python-
2.5-34.el7.x86_64.rpm policycoreutils-2.5-34.el7.x86_64.rpm selinux-
policy-3.13.1-268.el7_9.2.noarch.rpm setools-libs-3.3.8-4.el7.x86_64.
rpm systemd-219-78.el7_9.2.x86_64.rpm systemd-libs-219-78.el7_9.2.
x86_64.rpm systemd-sysv-219-78.el7_9.2.x86_64.rpm selinux-policy-
targeted-3.13.1-268.el7_9.2.noarch.rpm

 $ rpm -Uvh unzip-6.0-21.el7.x86_64.rpm
```

1. Move into a directory where you can download and store the installation files

```
cd /usr/src/
```

2. Download the below files and extract it to the installation path.

```
a. Nginx
        wget http://nginx.org/download/nginx-1.20.1.tar.gz
b. ModSecurity
        git clone --depth 1 -b v3/master --single-branch
SpiderLabs/ModSecurity
c. OpenSSL
        wget https://ftp.openssl.org/source/openssl-1.1.1k.tar.gz
d. Nginx-ModSecurity Connector
        git clone --depth 1 <https://github.com/SpiderLabs
/ModSecuritynginx.git> - Connect to preview
e. Header-more-nginx module
        git clone openresty/headers-more-nginx-module
```

3. Move to the downloaded mod-security directory to build.

```
cd /usr/src/ModSecurity
```

4. Run the below commands which you find under mod-security directory.

```
git submodule init
git submodule update
./build.sh
```

5. Configure mod-security with below command.

```
./configure
```

6. Run the make command to complete the installation.

```
make
make install
```

7. Now, we are done with mod-security installation.
8. Move to /usr/src/ directory and extract OpenSSL and Nginx tar files

```
cd /usr/src/
tar -xvzf openssl-1.1.1k.tar.gz
tar -xvzf nginx-1.20.1.tar.gz
```

9. Move to the extracted directory and start building Nginx with mod-security

```
cd nginx-1.20.1
```

10. Run the below command to configure nginx.

```
./configure --with-stream --with-compat --add-dynamic-module=..
/ModSecurity-nginx --with-debug –-with-http_ssl_module --with-
openssl=../openssl-1.1.1k/ --add-dynamic-module=../headers-more-
nginx-module/
```

11. Run make command to complete the installation.

```
make modules
make install
```

12. After successful installation, you will find the Nginx directory in the following path

```
/usr/local/nginx/
```

13. Check the version of the new nginx using the following command and check if the version is 1.18.0

```
/usr/local/nginx/sbin/nginx -V
```

14. Compare the old Nginx and new Nginx config file shared for v3.0 under dmapim-supportcomponents folder and modify location rules accordingly for gateway.
15. Copy the rules and dynamic components from the backup using the following command.

```
cp <<Nginx Backup Location>/conf/dynamic-modules.conf /usr/local
/nginx/conf cp <<Nginx Backup Location>/conf/rules.conf /usr/local
/nginx/conf
```

16. Copy the nginx.conf from the release package and configure the placeholder depends on the need.

```
cp -R <Release Package>/dmapim-support-components/nginx.conf /usr
/local/nginx/config
```

17. Once the sanity is performed, clean up the dependencies.

```
rm -rf /dmapim/downloads
```

2. Installing Apache Tomcat

1. Copy all the files from product deliverables to the home directory of target machine.
2. Download the Apache tomcat 9.0.52 package from below provided link

```
 https://archive.apache.org/dist/tomcat/tomcat-9/v9.0.52/bin
/apache-tomcat-9.0.52.tar.gz
```

3. Copy the downloaded file to home directory of target machine.
4. Copy the Apache Tomcat tar file under the path '/dmapim/' and extract it using the command below.

```
 tar xvf /user's-home-directory/apache-tomcat-9.0.52.tar.gz -C
/dmapim/
```

5. Create a user and group for Apache Tomcat.

```
 useradd tomcat
 groupadd tomcat
```

6. Set the below given permissions for apache-tomcat-9.0.52 directory.

```
 chown -R tomcat:tomcat /dmapim/apache-tomcat-9.0.52
```

7. The default error404.html and error500.html error pages for tomcat can be copied at

```
 '/dmapim/apache-tomcat-9.0.52/webapps/ROOT'  and can be
customized according to client’s requirement.
```

8. Start the tomcat using the below command.

```
/dmapim/apache-tomcat-9.0.52/bin/startup.sh
```

9. Recommand to follow the guide from the deliverable package dmapim-support-documents/DMAPIM-TomcatUpgradeAndSecurityGuide.pdf

3. Installing Java

1. Copy all the files from product deliverables to the home directory of target machine.
2. Copy the oracle java tar file jdk-8u301-linux-x64.tar.gz or AdoptOpenJDK java tar files OpenJDK8U-jdk_x64_linux_openj9_8u212b04_openj9-0.14.2.tar.gz and OpenJDK8U-jdk_x64_linux_8u232b09.tar.gz to the path '/dmapim/' and extract using the command below.

```
tar xvf /user's-home-directory/jdk-8u301-linux-x64.tar.gz -C
/dmapim/
```

3. Run the below given command to install Java.

```
cd /dmapim/<jdk extracted folder name>/
update-alternatives --install /usr/bin/java java /dmapim/<jdk
extracted folder name>/bin/java 1
update-alternatives --install /usr/bin/javac javac /dmapim/<jdk
extracted folder name>/bin/javac 1
```

4. Execute the below given command to check the current version of Java being used. Press enter to keep the current selection[+], or type selection number.

```
update-alternatives --config java
update-alternatives --config javac
```

5. Set the Java environment variables in '/etc/bashrc' file.

```
vi /etc/bashrc
#Setup JAVA_HOME Variable
export JAVA_HOME=/dmapim/<jdk extracted folder name>
#Setup JRE_HOME Variable
export JRE_HOME=/dmapim/<jdk extracted folder name>/jre
#Setup PATH Variable
export PATH=$PATH:/dmapim/<jdk extracted folder name>/bin:/dmapim
/<jdk extracted folder name>/jre/bin
```

6. Execute the below given command for the environment variables to take effect.

```
source /etc/bashrc
```

7. Check the installed Java version using the command below.

```
java -version
```

4. Installing Redis

1. Use the command below to install Redis v5.0.8 as a root user.

```
sudo su
yum install make
yum groupinstall 'Development Tools'
yum install tcl
mkdir -p /dmapim/Redis/

cd /dmapim/Redis/
wget http://download.redis.io/releases/redis-5.0.8.tar.gz
tar -xvf redis-5.0.8.tar.gz
cd redis-5.0.8
make
```

2. Create a directory for Redis logs and set correct permissions.

```
mkdir -p /dmapim/var/log/redis
chmod -R 755 /dmapim/var/
chown root:root /dmapim/var/log/redis
```

3. Update the log file path in redis.conf file.

```
vi /dmapim/Redis/redis-5.0.8/redis.conf
logfile "/dmapim/var/log/redis/redis.log"
```

4. Run the command below to start Redis server.

```
cd /dmapim/Redis/redis-5.0.8/src
./redis-server &
```

5. How to stop redis server

```
ps -ef | grep redis
kill <process id>
```

5. Installing MariaDB

1. Execute below command and Add the below details in file /etc/yum.repos.d/MariaDB.repo.

```
[mariadb]
name = MariaDB
baseurl=https://archive.mariadb.org/mariadb-10.4.12/yum/centos7-amd64
gpgkey=https://yum.mariadb.org/RPM-GPG-KEY-MariaDB
gpgcheck=1
```

2. Install mariaDB using below command

```
sudo yum install MariaDB-server MariaDB-client
```

3. Run the command below to start the mariadb service.

```
sudo systemctl start mariadb
sudo systemctl enable mariadb
```

4. MariaDB includes a security script to change some of the less secure default options, Use this command to run the security script to set up the root user password, remove anonymous user accounts etc,.

```
sudo mysql_secure_installation
```

5. Create a directory for MariaDB logs and set the below permissions.

```
mkdir -p /dmapim/var/log/mariadb
chmod -R 755 /dmapim/var/
chown mysql:mysql /dmapim/var/log/mariadb
```

6. Set MariaDB parameter tuning in /etc/my.cnf file.
vi /etc/my.cnf

```
 [mysqld]
 sql_mode='STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,
ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER,
NO_ENGINE_SUBSTITUTION'
 event_scheduler=ON
 collation-server = utf8mb4_general_ci
 init-connect='SET NAMES utf8mb4'
 character-set-server = utf8mb4
 log-error=/dmapim/var/log/mariadb/mariadb.log
 lower_case_table_names = 1
 key_buffer_size=8M
 max_allowed_packet=500M
 max_connections=5000
 # bind-address=0.0.0.0
 [mysql]
 default-character-set=utf8mb4
 [client]
 default-character-set=utf8mb4
```

7. Restart MariaDB service.

```
sudo systemctl restart mariadb
```

6. Installing Kafka

1. Create a new user called kafka using the below command from root user

```
adduser kafka
```

2. Create a folder using the below command

```
mkdir -p /dmapim/kafka/
```

3. Assign ownership of the folder to kafka user using the below command

```
chown -R kafka:kafka /dmapim/kafka/
```

4. Login to kafka user using below command

```
sudo su kafka
```

5. Change the directory

```
cd /dmapim/kafka/
```

6. Download Kafak 2.5.0 using the below command

```
wget https://archive.apache.org/dist/kafka/2.5.0/kafka_2.12-2.5.0.
tgz
```

7. Unzip the kafka_2.12-2.5.0.tgz using the below command

```
tar -xzf kafka_2.12-2.5.0.tgz
```

8. Navigate to kafka folder using below command and copy sh files also replace the placeholders

```
cd /dmapim/kafka/
cp <dmapim-support-components from product deliverable>/zookeeper.
sh .
cp < dmapim-support-components from product deliverable>/kafka.sh
.
chmod 764 zookeeper.sh kafka.sh
dos2unix zookeeper.sh kafka.sh
```

9. Configure some basic zookeeper and kafka server properties

```
mkdir /dmapim/kafka/kafka_2.12-2.5.0/data
vi /dmapim/kafka/kafka_2.12-2.5.0/config/zookeeper.properties
dataDir=data/zookeeper
vi /dmapim/kafka/kafka_2.12-2.5.0/config/server.properties
advertised.listeners=PLAINTEXT://<HOST_NAME or IP>:<KafkaPort>
log.dirs=data/kafka-logs
```

10. Start zookeeper using the below command

```
./zookeeper.sh -start
```

11. Start kafka using the below command

```
./kafka.sh -start
```

12. To stop the Kafka and Zookeeper

```
./zookeeper.sh -stop
./kafka.sh -stop
```

7 Installing Elasticsearch

1. Login as root user
2. Extract the Elasticsearch binary (Required internet to download the Elasticsearch binary otherwise download in prior and copy to the specific location with root permission)

```
 cd <product installation folder>
 curl -L https://artifacts.elastic.co/downloads/elasticsearch
/elasticsearch-oss-7.6.0-linux-x86_64.tar.gz -o elasticsearch-oss-
7.6.0-linux-x86_64.tar.gz
 tar -xzvf elasticsearch-oss-7.6.0-linux-x86_64.tar.gz
```

3. Create elasticsearch user for Logstash and Elasticsearch

```
 useradd elasticsearch
```

4. Change ownership using the below command

```
       chown -R elasticsearch:elasticsearch <product installation
folder>/elasticsearch-7.6.0/
```

5. Login as elasticsearch user

```
 cd elasticsearch-7.6.0
 su elasticsearch
```

6. Set JAVA_HOME by adding the below lines in ~/.bashrc file

```
export JAVA_HOME=/<product installation folder>/jdk8u232-b09
export PATH=$PATH:$JAVA_HOME/bin
```

7. Logout and relogin as elasticsearch user
8. Open config/elasticsearch.yml file and edit the configurations as given below,

```
cluster.name: elasticsearch
#Increase the below value if more than 500 indexes needs to be created in Elasticsearch
cluster.max_shards_per_node: 1000
node.name: es
network.host: 0.0.0.0
#Elasticsearch port
http.port: 9200
cluster.initial_master_nodes: ["es"]
```

Note : If the above configurations are commented please uncomment it.

1. (Optional) To update Elasticsearch heap size to 512MB execute the below steps,

   8.1. Open config/jvm.options file and edit the configurations as given below,

```
Replace the line "-Xms1g" with "-Xms512m"
Replace the line "-Xmx1g" with "-Xmx512m"
```

2. Add the below lines in ~/.bashrc file

```
export ELASTICSEARCH_HOME=/<product installation folder>/elasticsearch-7.6.0
export PATH=$PATH:$ELASTICSEARCH_HOME/bin
```

3. Logout and relogin to execute ~/.bashrc
4. Login as elasticsearch user and start elasticsearch

```
elasticsearch &
```

5. To stop elasticsearch

```
ps -ef | grep elasticsearch
kill -9 <processid>
```

Note : Elasticsearch logs can be found in the following path
“$ELASTICSEARCH_HOME/logs”

8 Installing Analytics Server

8.1 Install Python3.6 and pip

1. Login as root
2. Install dependencies

```
yum install autoconf
yum install automake
yum install binutils
yum install bison
yum install flex
yum install gcc
yum install gcc-c++
yum install gettext
yum install libtool
yum install make
yum install patch
yum install pkgconfig
yum install redhat-rpm-config
yum install rpm-build
yum install rpm-sign
yum install zlib-devel
yum install openssl-devel.x86_64 openssl.x86_64
```

3. Copy Python-3.6.4.tar.xz to /tmp
4. Extract the Python package

```
cd /tmp
tar -xJf Python-3.6.4.tar.xz
```

5. Install Python 3.6

```
cd Python-3.6.4
./configure
make
make install
update-alternatives --install /usr/bin/python python /usr/local
/bin/python3.6 1
update-alternatives --install /usr/bin/pip pip /usr/local/bin
/pip3.6 1
```

6. Execute the below given command to check the current version of Python being used.
   Press enter to keep the current selection[+], or type selection number.

```
update-alternatives --config python
update-alternatives --config pip
```

7. Edit Python dependency for Yum

```
a. Open the file /usr/bin/yum
b. Change the first line #!/usr/bin/python to #!/usr/bin/python2
c. Save and Continue
```

8.2 Install Gunicorn and Wheel

Gunicorn will be used as the Web Server. Wheel will to be used for binary. To install Gunicorn and wheel, please execute the below command,

```
pip install gunicorn
pip install wheel
pip install --upgrade pip wheel setuptools
pip install tqdm
pip install --user --upgrade twine
ln -fs /usr/local/bin/gunicorn /usr/bin/gunicorn
```

9 Installing Logstash

1. Login as root user
2. Extract the Logstash binary (Required internet to download the Logstash binary otherwise download in prior and copy to the specific location with root permission)

```
cd <product installation folder>
curl -L https://artifacts.elastic.co/downloads/logstash/logstash-oss-7.6.0.tar.gz -o logstash-oss-7.6.0.tar.gz
tar -xzvf logstash-oss-7.6.0.tar.gz
cd logstash-7.6.0/
```

3. Change ownership using the below command

```
chown -R elasticsearch:elasticsearch <product installation folder>/logstash-7.6.0/
```

4. Login as elasticsearch user

```
su elasticsearch
```

5. Make sure JAVA_HOME environment variable is set
6. Update the below lines in config/pipeline.yml

```
 - pipeline.id: dmapim
  path.config: "<product installation folder>/logstash-7.6.0/config
/dmapim_logstash.conf"
  queue.type: persisted
  pipeline.workers: 1
```

7. If Kafka is installed and enabled then please add the below lines in config/pipeline.yml

```
- pipeline.id: gateway_events
  path.config: "<product installation folder>/logstash-7.6.0/config
/json_multiline.conf"
  queue.type: persisted
  pipeline.workers: 1
- pipeline.id: kafka-es
  path.config: "<product installation folder>/logstash-7.6.0/config
/kafka_elasticsearch.conf"
  queue.type: persisted
  pipeline.workers: 4
```

8. (Optional) To update Logstash heap size to 512MB execute the below steps,
   7.1. Open config/jvm.options file and edit the configurations as given below,

```
 Replace the line "-Xms1g" with "-Xms512m"
 Replace the line "-Xmx1g" with "-Xmx512m"
```

9. Add the below lines in ~/.bashrc

```
 export LOGSTASH_HOME=/<product installation folder>/logstash-7.6.0
 export PATH=$PATH:$LOGSTASH_HOME/bin
```

10. Logout and relogin to execute ~/.bashrc
11. Login as elasticsearch user and start logstash by executing the below command

```
logstash &
```

12. To stop logstash

```
ps -ef | grep logstash
kill -9 <processid>
```

10 Installing Elasticsearch Curator

1. Login as root user
2. Install Elasticsearch Curator using the below command (Required internet to download the Elasticsearch curator binary otherwise download in prior and copy to the specific location with root permission)

```
 cd /tmp
 curl -L https://packages.elastic.co/curator/5/centos/7/Packages
/elasticsearch-curator-5.8.1-1.x86_64.rpm -o elasticsearch-curator-
5.8.1-1.x86_64.rpm
 rpm -ivh elasticsearch-curator-5.8.1-1.x86_64.rpm
```

3. Copy elasticsearch_curator directory from deliverables to Product installation folder (/dmapim for DMAPIM, /apic for APIConnect)
4. Change ownership using the below command

```
 chown -R elasticsearch:elasticsearch <product installation folder>
/elasticsearch_curator/
```

5. Switch to elasticsearch user

```
 su elasticsearch
 cd <product installation folder>/elasticsearch_curator/
```

6. Update the placeholder values in curator-config.yml and curator-action.yml
7. Schedule the curator in cron using the below command,

```
 crontab -e
 0 0 * * * cd <product installation folder>/elasticsearch_curator
&& curator curator-action.yml --config curator-config.yml >>
indices_retention.log
```

Note :

• Update the placeholder value in the above command
• Logs can be found in the following path “<product installation folder>/elasticsearch_curator/indices_retention.log”

Installing FileBeat in Gateway server

1. Login into gateway server where logs has to be fetched
2. Extract the Filebeat binary (Required internet to download the Filebeat binary otherwise download in prior and copy to the specific location with gateway user permission)

```
 cd <product installation folder>
 curl -L https://artifacts.elastic.co/downloads/beats/filebeat
/filebeat-oss-7.6.0-linux-x86_64.tar.gz -o filebeat-oss-7.6.0-
linux-x86_64.tar.gz
 tar -xzvf filebeat-oss-7.6.0-linux-x86_64.tar.gz
```

3. Change ownership as gateway user using the below command

```
 chown -R <gateway user>:<gateway user> <product installation
folder>/filebeat-7.6.0-linux-x86_64/
 cd <product installation folder>/filebeat-7.6.0-linux-x86_64/
```

4. Login as gateway user
5. Open filebeat.yml file and add the below configurations under filebeat.inputs: as given below,

```
- type: log
 # Change to true to enable this input configuration.
 enabled: true
 # Paths that should be crawled and fetched. Glob based paths.
 paths:
 - <path to gateway application logs>
 fields:
 type: api_gateway
 multiline.pattern: '^[[:space:]]+(\<)|^\<'
 multiline.negate: false
 multiline.match: after
- type: log
 # Change to true to enable this input configuration.
 enabled: true
 # Paths that should be crawled and fetched. Glob based paths.
 paths:
 - <api gateway error logs>
 fields:
 type: api_error
 multiline.pattern: '^[[:space:]]|^]|^$'
 multiline.negate: false
 multiline.match: after
```

6. If Kafka output is disabled in gateway, open filebeat.yml file and add the below configurations under filebeat.inputs: as given below,

```
- type: log
  # Change to true to enable this input configuration.
  enabled: true
  # Paths that should be crawled and fetched. Glob based paths.
  paths:
    - <path to gateway transaction logs>
  fields:
    type: gateway_transactions
```

7. Comment the below lines as given below

```
#output.elasticsearch:
# Array of hosts to connect to.
#hosts: ["localhost:9200"]
```

8. Edit the below lines as given below,

```
output.logstash:
 # The Logstash hosts
 hosts: ["<logstash ip>:5044"]
```

Please Note : If the port number is changed in “dmapim_logstash.conf” file, then in the above line configure the same port instead of default value 5044.

8. Start filebeat by using the below command,

```
 ./filebeat -e -c filebeat.yml -d "publish" >>
filebeat_output_logs_`date +'%Y-%m-%d'`.log &
```

Please Note : Filebeat logs can be found in the following path “<product installation folder>/filebeat-7.6.0-linux-x86_64/” with prefix “filebeat_output_logs”

9. Schedule the cron for logs retention using the below command,

```
 crontab -e
 0 0 * * * find <product installation folder>/filebeat-7.6.0-linux-
x86_64/ -name
 'filebeat_output_logs_*' -mtime +<number_of_days_to_retainlogs> -type
f | xargs rm
```

Note :
• Update the placeholder value in the above command