The following process is same from Prod to Non prod.

**Basic details**

Apache –> Apache used in Web layer and RP Layer and new forms.

Live-www, live-online are instances of Apache.

Currently we are using Apache 2.4.58 in AWS and in Dublin we are using 2.2.

We are using openssl-1.1.1w in AWS and in Dublin 1.0.1(it’s not in support hence not receiving any update for this).

OPENSSL- OpenSSL is an open-source cryptographic toolkit that helps secure communications between network endpoints. It's used by software developers and system administrators to implement secure communication and encryption in applications such as web servers, email servers, and VPNs. OpenSSL is widely used by Internet servers, including the majority of HTTPS websites.

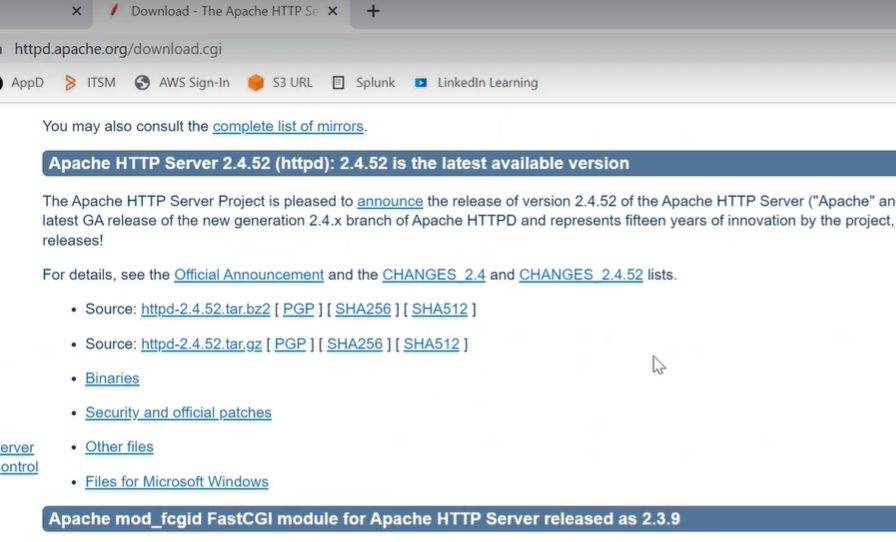
TLS and SSL protocols: OpenSSL implements Transport Layer Security (TLS) and Secure Sockets Layer (SSL) protocols.

Apache: serves as a web server software that enables the hosting of websites and web applications. Its primary purpose is to handle requests from clients (like web browsers) and deliver web content, such as HTML pages, images, and scripts.

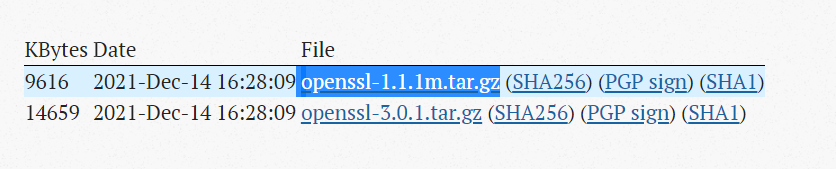
**Prerequisite for Apache and OpenSSL upgrade:**

1. Check and download latest Apache and OpenSSL .
2. For Apache visit <https://httpd.apache.org/download.cgi> as shown below

Always check change list as well for more details what changes and update we are receiving in latest version.



1. For OpenSSL we have to visit (<https://www.openssl.org/source/>) and check any new version is present or not. If any new version present, then we have to update it.



**Stakeholder Name:**

1. Architect - [senthil.ramalingam@vodafone.com](mailto:senthil.ramalingam@vodafone.com)
2. Environment Availability: [jo.steel3@vodafone.com](mailto:jo.steel3@vodafone.com)
3. Deployment in temp – eCare L2 (DL-TSSI-AO2\_UK\_Online\_eCare@vodafone.com)
4. Deployment in TCC\* **-** eCare L2 (DL-TSSI-AO2\_UK\_Online\_eCare@vodafone.com)
5. Deployment PAT **-** eCare L2 ([DL-TSSI-AO2\_UK\_Online\_eCare@vodafone.com](mailto:DL-TSSI-AO2_UK_Online_eCare@vodafone.com))
6. PAT Test Result – PAT testing team ( DL-VATS-WCP-CI-TestTeam [VATS-WCP-CI-TestTeam@vodafone.com](mailto:VATS-WCP-CI-TestTeam@vodafone.com) ) , [lovely.kumari3@vodafone.com](mailto:lovely.kumari3@vodafone.com) , [aishwarya.borate@vodafone.com](mailto:aishwarya.borate@vodafone.com) , [nikitha.mali@vodafone.com](mailto:nikitha.mali@vodafone.com)
7. Deployment in Plive2- eCare L2 ([DL-TSSI-AO2\_UK\_Online\_eCare@vodafone.com](mailto:DL-TSSI-AO2_UK_Online_eCare@vodafone.com))
8. Deployment in Live **–** eCare L2 ([DL-TSSI-AO2\_UK\_Online\_eCare@vodafone.com](mailto:DL-TSSI-AO2_UK_Online_eCare@vodafone.com))
9. DVT on Plive2 and Live - eCare L2 ([DL-TSSI-AO2\_UK\_Online\_eCare@vodafone.com](mailto:DL-TSSI-AO2_UK_Online_eCare@vodafone.com))
10. If any Issue occurs: eCare L2 ([DL-TSSI-AO2\_UK\_Online\_eCare@vodafone.com](mailto:DL-TSSI-AO2_UK_Online_eCare@vodafone.com))
11. CRQ Dates: [john.ferry@vodafone.com](mailto:john.ferry@vodafone.com) , [ukitchangemanagement@vodafone.com](mailto:ukitchangemanagement@vodafone.com)

Note: We usually try to take these changes along with AMI or Oracle Patches.

There are two ways to upgrade Apache and OpenSSL version Manual and Automated.

# Below are the Manual steps for Upgrade Apache and OpenSSL:

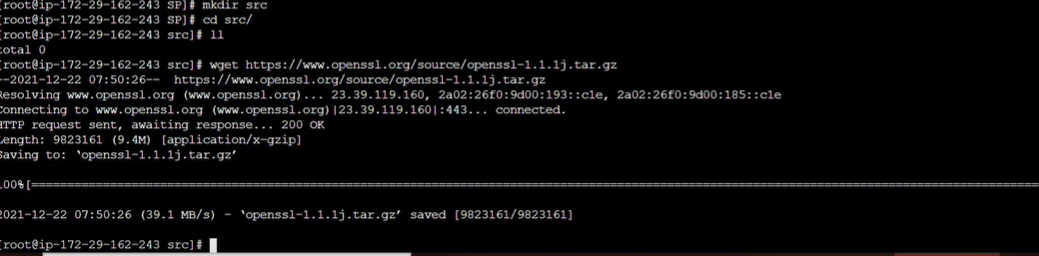
**For Creating Build, we have to follow the below sequence.**

1. Need to build openSSL first.
2. Then Apache
3. Then Apache with PHP

**Sequence for MySQL**

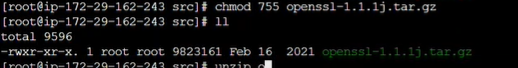
1. Open SSL
2. My SQL
3. If new version presents for Apache and OpenSSL we have to download it on local or either on server directly. In case of automation we have to upload at S3 location(path mentioned in below steps).
4. Creating directory src and download openSSL from internet using below command on buildhost.

**wget** [**https://www.openssl.org/source/openssl-1.1.1j.tar.gz**](https://www.openssl.org/source/openssl-1.1.1j.tar.gz)



1. Change the permissin

**chmod 755 openssl-1.1.1j.tar.gz**



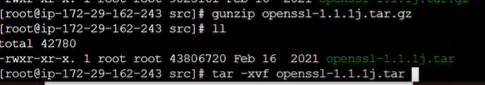
1. Unzip it

gunzip **openssl-1.1.1j.tar.gz**

**tar -xvf openssl-1.1.1j.tar**

**Run this command for perlipc cmd for the latest versuibon 3.2.p**

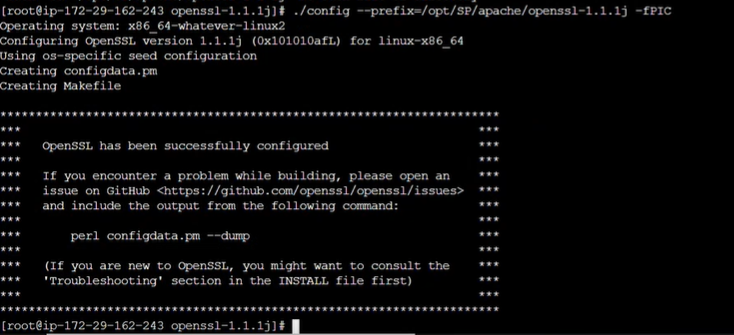
**yum install perl-IPC-Cmd**



1. Then edit config file inside **openssl-1.1.1j.**

./config --prefix=/opt/SP/apache/openssl-1.1.1j -fPIC Its folder name for our purpose but recommeded to keep version for our understanding

This command compile



1. Now Install openSSL. Using command **make** and **make install**

First execute ***make clean – it remove all existing files***

Then execute ***make - checking all the dependency and all***

Finally ***make install***

1. After installing we can see openssl installed or upgraded at below location

**/opt/SP/apache/openssl-1.1.1j**

1. Next task is to download Apache and Apache APR (if any new version for APR otherwise we can ignore APR )

***cd /opt/SP/source***

***wget*** [***https://mirrors.estointernet.in/apache//httpd/httpd-2.4.58.tar.gz***](https://mirrors.estointernet.in/apache//httpd/httpd-2.4.46.tar.gz)

***wget*** [***https://mirrors.estointernet.in/apache//apr/apr-1.7.0.tar.gz***](https://mirrors.estointernet.in/apache//apr/apr-1.7.0.tar.gz)

***tar -xvf httpd-2.4.58.tar.gz***

***tar -xvf apr-1.7.0.tar.gz***

1. Copy APR to specific location (if new APR not present then still do we need to copy APR to specific loaction ,Path of APR needed)
2. Then execute below command for ***configure***

***LDFLAGS="-L/usr/lib64"; export LDFLAGS;***

***./configure \***

***--with-ssl=/opt/SP/apache/openssl-1.1.1j \***

***--prefix=/opt/SP/apache/apache2.4.58.ssl-111j \***

***--enable-mods-shared=all \***

***--enable-mods-static=all \***

***--enable-mpms-shared="worker event prefork" \***

***--enable-alias \***

***--enable-allowmethods \***

***--enable-buffer \***

***--enable-cache-disk \***

***--enable-cache-socache \***

***--enable-file-cache \***

***--enable-cache \***

***--enable-charset-lite \***

***--enable-data \***

***--enable-deflate \***

***--enable-dir \***

***--enable-env \***

***--enable-expires \***

***--enable-filter \***

***--enable-headers \***

***--enable-heartbeat \***

***--enable-heartmonitor \***

***--enable-http \***

***--enable-http2 \***

***--enable-include \***

***--enable-log-config \***

***--enable-log-debug \***

***--enable-logio \***

***--enable-macro \***

***--enable-mime-magic \***

***--enable-mime \***

***--enable-negotiation \***

***--enable-proxy=static \***

***--enable-proxy-html=static \***

***--enable-proxy-connect \***

***--enable-proxy-http \***

***--enable-proxy-balancer \***

***--enable-proxy-express \***

***--enable-proxy-http2 \***

***--enable-ratelimit \***

***--enable-reqtimeout \***

***--enable-request \***

***--enable-reflector \***

***--enable-rewrite \***

***--enable-sed \***

***--enable-setenvif \***

***--enable-session \***

***--enable-session-cookie \***

***--enable-slotmem-shm \***

***--enable-slotmem-plain \***

***--enable-so \***

***--enable-socache-memcache \***

***--enable-ssl \***

***--enable-status \***

***--enable-substitute \***

***--enable-usertrack \***

***--enable-unique-id \***

***--enable-unixd \***

***--enable-vhost-alias \***

***--enable-watchdog \***

***--with-pcre \***

***--with-included-apr \***

***--with-nghttp2 \***

***--disable-actions \***

***--disable-autoindex \***

***--disable-auth-basic \***

***--disable-asis \***

***--disable-cgi \***

***--disable-cgid \***

***--disable-dialup \***

***--disable-dav \***

***--disable-dav\_fs \***

***--disable-dav\_lock \***

***--disable-dbd \***

***--disable-dumpio \***

***--disable-echo \***

***--disable-ext-filter \***

***--disable-imagemap \***

***--disable-log\_forensic \***

***--disable-proxy-scgi \***

***--disable-proxy-ajp \***

***--disable-proxy\_fcgi \***

***--disable-proxy\_scgi \***

***--disable-proxy\_fdpass \***

***--disable-proxy\_wstunnel \***

***--disable-proxy\_ajp \***

***--disable-session\_dbd \***

***--disable-speling \***

***--disable-userdir \***

***"LDFLAGS=-L/usr/lib64" \***

***"$@"***

***Ex: --with-ssl=/opt/SP/apache/openssl-1.1.1j \ 🡪***

***--prefix=/opt/SP/apache/apache2.4.58.ssl-111j \ 🡪***

***--enable-http \***

***--enable-http2 \***

***Enable -> this module is enable***

***Disable – this module is disabled***

***LDFLAG – libraray path***

1. Now Install the apache

***make clean – it remove all existing files***

***make - checking all the dependency and all***

***make install***

1. Apache build done at location

***/opt/SP/apache/apache2.4.58.ssl-111j***

***Note : If only apche version is upgraded then in below comand we have to provide the exsting openssl version***

***LDFLAGS="-L/usr/lib64"; export LDFLAGS;***

***./configure \***

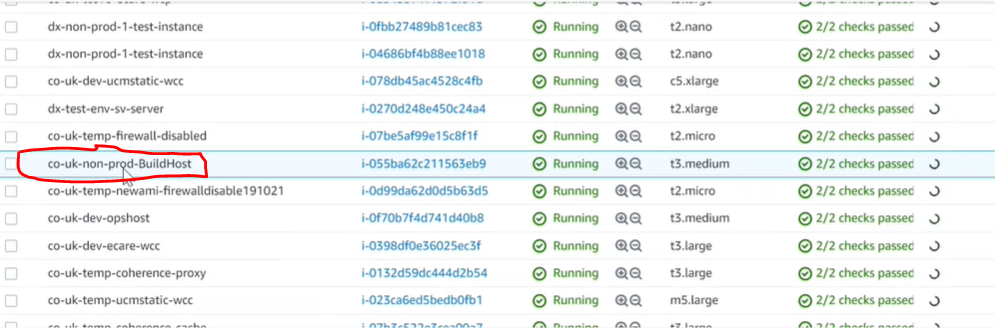
***--with-ssl=/opt/SP/apache/openssl-1.1.1j \***

***--prefix=/opt/SP/apache/apache2.4.58.ssl-111j \ -- it can be any name but for our understanding we are gaive name same as apache version***

***--enable-mods-shared=all \***

# Automated steps for upgrading Apache & OpenSSL version

1. BuildHost is taking care of builds

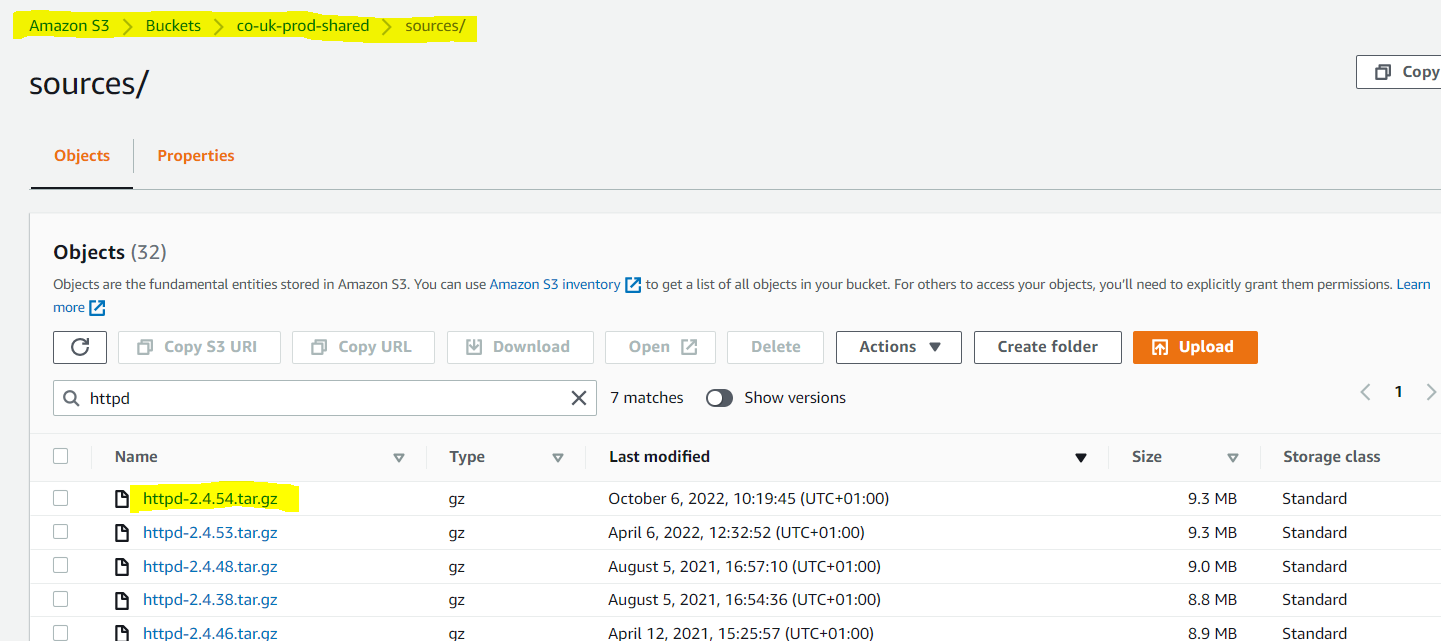


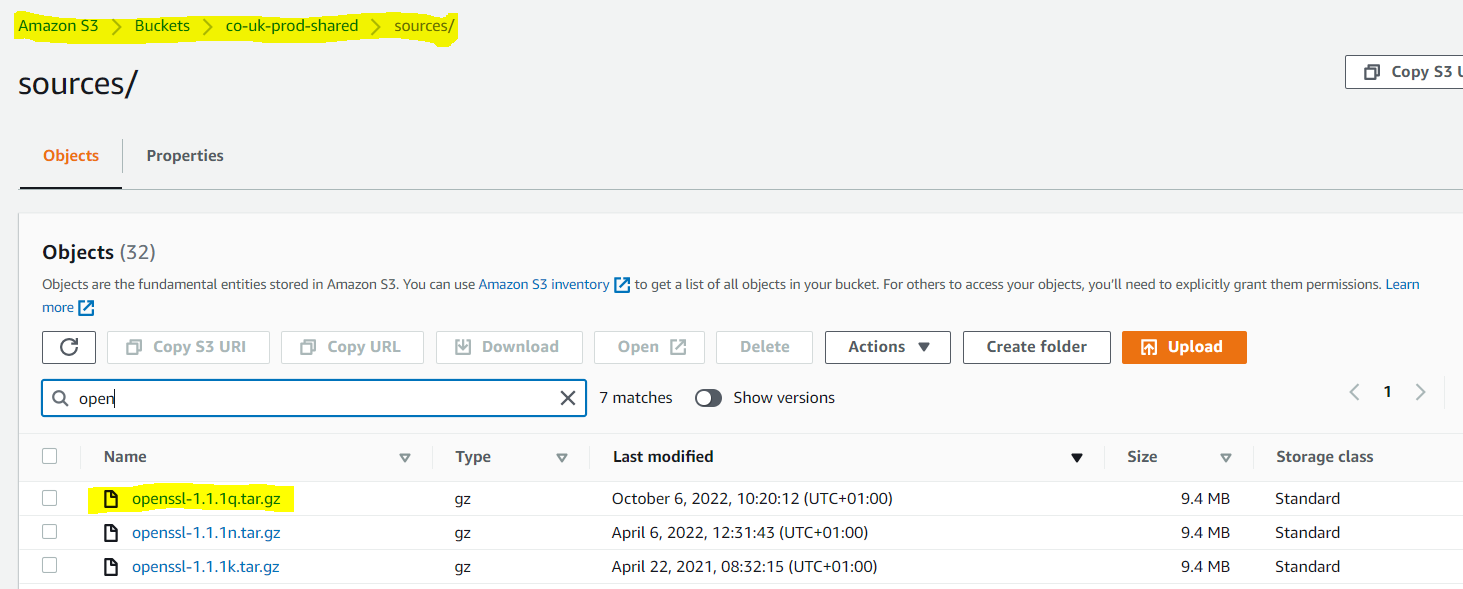
1. Then need to place package tar.gz at below S3 location, Then corresponding to Live loaction we have to place file. For Apache file name start with httpd.

PFB screenhot of latest apache 2.4.58 and SSl 1.1.1w

PLIVE : S3/co-uk-non-prod-shared/sources

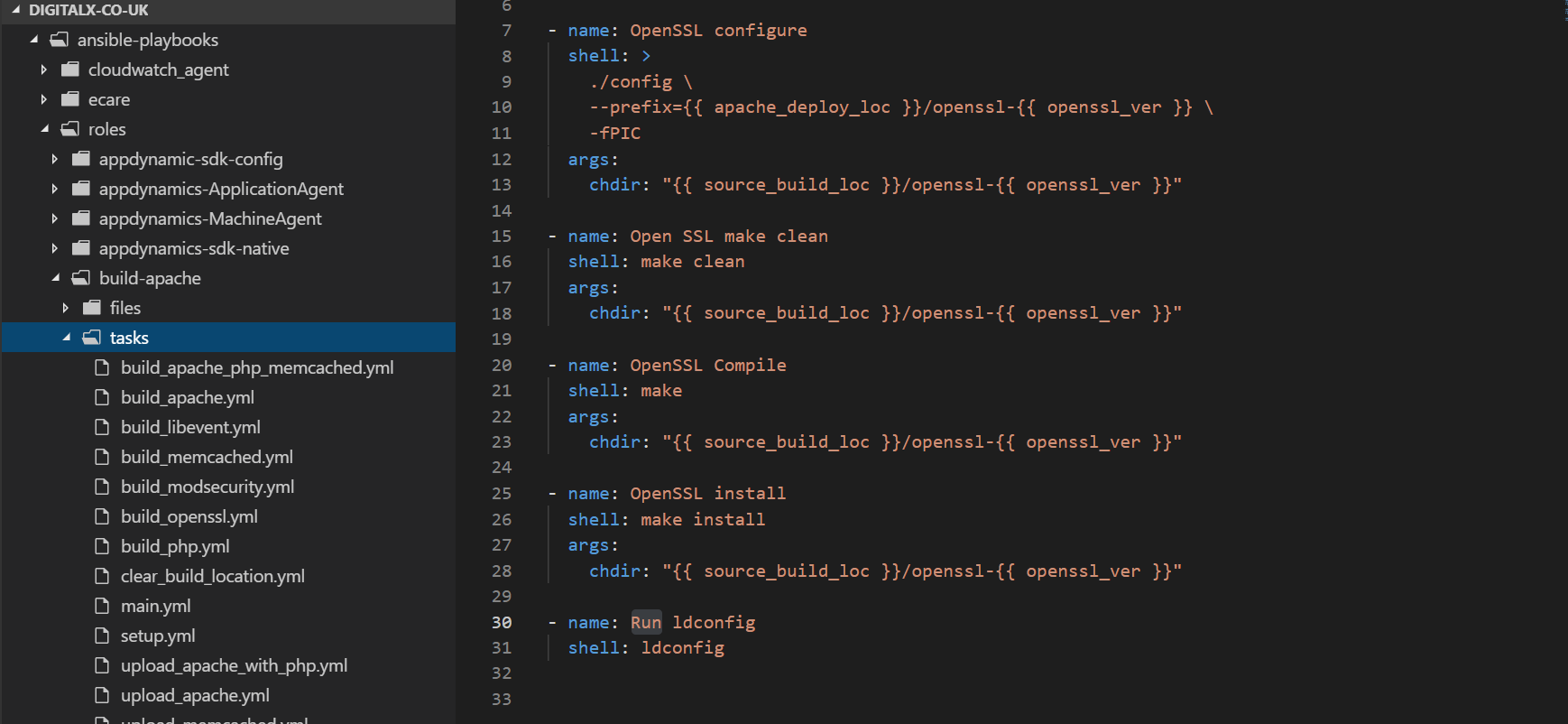
LIVE : S3/co-uk-prod-shared/sources





1. Below is the path of files for changing and upgrading OpenSSL, Apache APR

Playbooks -> roles->build-apache->task



1. Setup.yml : Having all the dependency***.***

# Package Installs

- name: Install Packges for build

package:

name: [ "gcc", "pcre-devel", "zlib-devel", "libxml2-devel", "nghttp2", "libnghttp2-devel", "expat-devel", "expat", "python3", "libevent", "libev-libevent-devel", "perl" ,"python2-botocore", "python2-boto", "python-boto", "python2-pip", "oniguruma", "oniguruma-devel", "sqlite-devel" ]

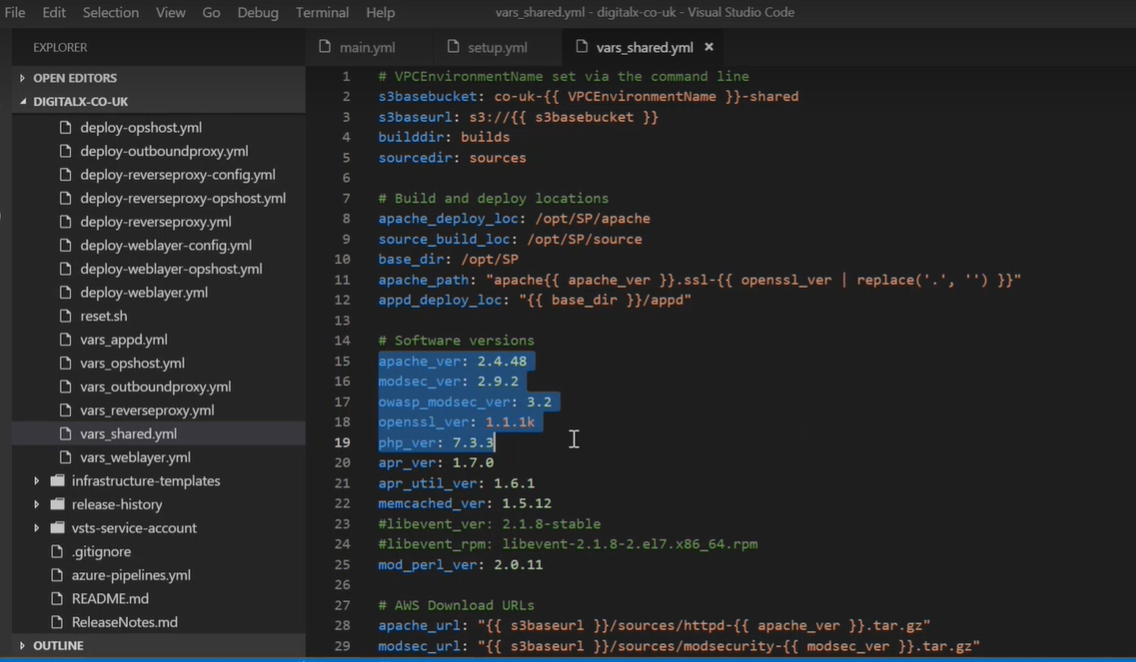
state: present

upload\_openssl.yml : Package Installs , Create Initial Directories , Download sources from S3 loaction , Extract archives to relevant location ,Copy APR into Apache srclib

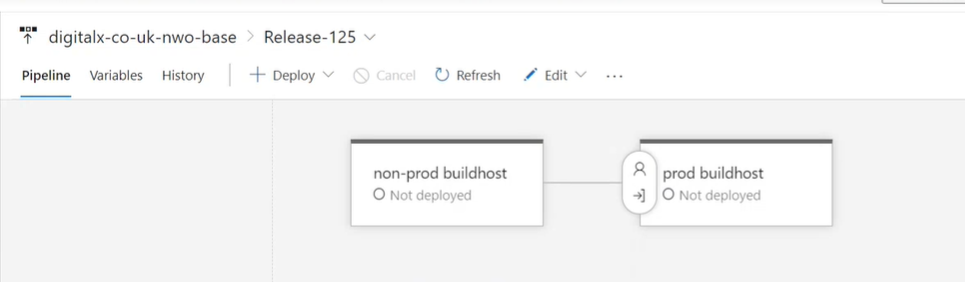
Files staterd with build (build\_apache.yml , build\_openssl.yml) are used for creating build. All build command are witten here.

Files stated with Upload (upload\_apache.yml , upload\_openssl.yml ) will upload upload build to S3

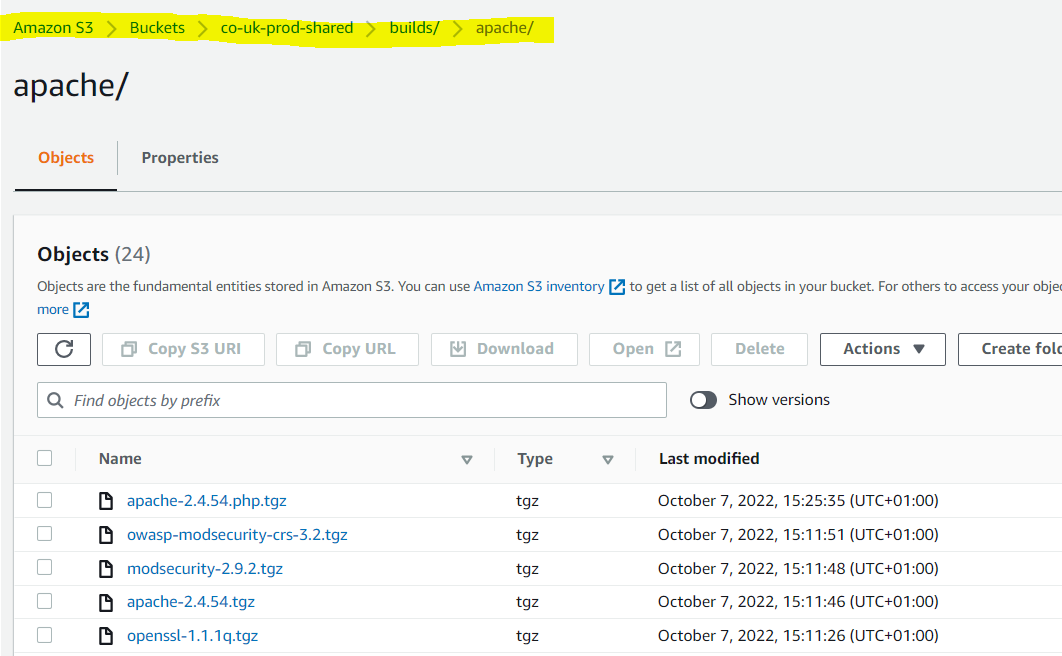
1. Need to change some variables as below. We have to change appropriate version at below highlighted files (**vars\_shared.yml**) as per requirement.



1. Then create build from latest code digitalx-co-uk
2. After that have to run buildhost stage with new release in digitalx-co-uk-now-base in pipeline then it wil automatically create build and all.



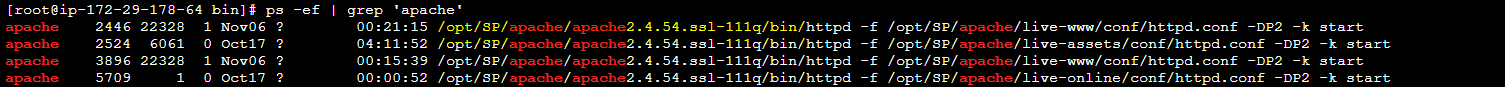
1. It will upload build to S3

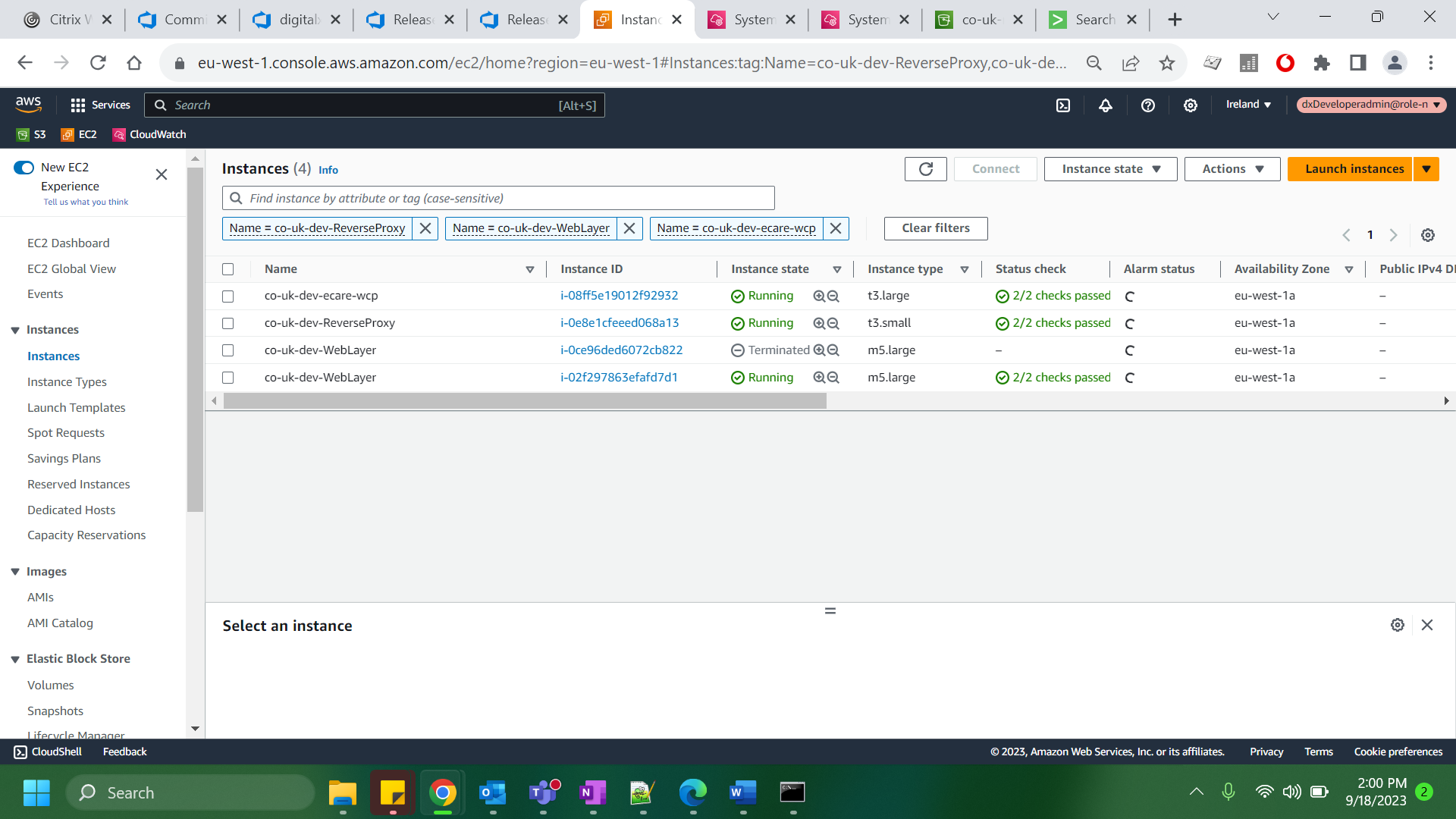


***If the new artificats were not generated and instance was not reinitiated then terminate the instance and wait until the build got succeded and check the new artifiacts***

1. Then we have to deploy on plive2 and Live RP and weblayer stages through **digitalx-co-uk-nwo-middleware** pipeline reverseproxy and weblayer
2. Build is successful logs we have to check in cfn-init.log logs at /var/log location
3. Then check on then server new version reflected or not.

ps -ef | grep ‘apache’





***In the same way once we have added the latest version in repo and placed the jars in S3 we can create build and create a new release under “AMI-Automation-Pipeline” and deploy under stages.***