OrangeHRM Enterprise Application

Middleware Installation Guide

OrangeHRM 7.4 and Above
Version 1.4

Document History

Release Date	Version No.	Author(s)	Description	Recommended by	Approved by
August 31, 2021	1.0	ISO	Initial Release	TechOps	HOD
November 01, 2021	1.1	ISO	Update RabbitMq and MariaDB new versions	TechOps	HOD
November 15, 2021	1.2	ISO	Update MariaDB version to 10.3	TechOps	HOD
January 20, 2022	1.3	ISO	New middleware and SSL certificate instructions	TechOps	HOD
April 17, 2024	1.4	ISO	Update RabbitMQ Requirement as Optional	TechOps	HOD

Table of Contents

1. Introduction

OrangeHRM is a web-based HRMS application which supports Apache, MariaDB, and PHP(AMP). This document describes the steps to be taken in order to prepare the middleware layer before installing the OrangeHRM Application.

Some of the configuration details referenced in this guide may vary depending on the specific application requirements and infrastructure resource availability. OrangeHRM Support Services team will assist you with such changes.

1.1 Required OrangeHRM Instances

Two instances of OrangeHRM are required for an onsite installation.

1. **Production Instance** - This is the live system which will be used by all the end-users and admins.

2. **Pre-Production Instance** - This is the instance that mimics the Production instance and is used for testing prior to any new deployments being applied to the Production instance. Once testing successfully concludes on the Pre-Production instance, the new build/patch would then be applied to the Production instance.

It is highly recommended to have two servers/VMs for the two instances. Both servers should be exactly the same in terms of hardware and software configuration. Please refer to the Hardware Requirements section in ***OrangeHRM-HW-SW-Requirements-Guide*** for details of the required hardware specifications.

2. Software platform preparation

2.1 Register a system with the Red Hat Customer Portal

Ensure your system is registered with the Red Hat customer portal if not registered already (i.e RedHat subscription is properly configured).

This section describes how to use Red Hat Subscription Manager to register and subscribe a system to the Red Hat Customer Portal. This is required to install and update software packages.

1. Use the command below to register a system with the Red Hat Customer Portal. Use the same username and password you used to sign up for an account in the Red Hat Customer Portal.

```
orangehrm@workspace:~ sudo subscription-manager register
```

2. Attach a subscription

```
#Refresh subscriptions
orangehrm@workspace:~ sudo subscription-manager refresh

#Attach a subscription
orangehrm@workspace:~ sudo subscription-manager attach --auto
```

NOTE

- If the subscription manager is not available in your system, please contact your server vendor.

2.2 Platform Installation

The following section describes the automated installation of the following services listed below.

- Apache HTTP Server version 2.4
- MariaDB version 10.4.22
- PHP version 7.4
- RabbitMQ version 3.8.2 (Optional)
- Other required dependencies
    - **ImageMagick** - This tool is used to process images within the OrangeHRM Application.
    - **openssl-devel** - This is required to generate certificate files.
    - **poppler-utils** - This tool is used to process pdf files within the OrangeHRM Application.
    - **libreoffice modules** - This tool is used to process document files within the OrangeHRM Application.
    - **Fonts** - Font packages to support different font types within the OrangeHRM application.

IMPORTANT

- Make sure SSL certificates are prepared properly
    - Please refer to the steps outlined on how to create certificates from Appendix A - Create SSL certificates with openssl
- Prior to the installation,
    - Update packages to the latest versions and restart the server (or virtual machine).

```
orangehrm@workspace:~ sudo dnf update -y
```

    - Make sure SELinux is in permissive mode. If SELinux needs to be changed into permissive mode, please refer to Enable permissive mode in Selinux
- Commands are prepared based on RHEL 8

NOTE

Ensure RHEL codeready builder repository ("***codeready-builder-for-rhel-8-x86_64-rpms***") is enabled. Use the following command to verify the availability of codeready-builder. This command will print the repo name if the codeready-builder repository is already enabled.

```
orangehrm@workspace:~ sudo dnf repolist | grep codeready-builder
```

Codeready builder repository name differs based on the service provider (eg: in Google Cloud the repo is referred to as ***rhui-codeready-builder-for-rhel-8-x86_64-rhui-rpms***).
The following command can be used to enable the codeready-builder repository using subscription-manager.

```
orangehrm@workspace:~ sudo subscription-manager config
--rhsm.manage_repos=1

orangehrm@workspace:~ sudo subscription-manager repos --enable
codeready-builder-for-rhel-8-x86_64-rpms
```

Install the following utility tools.

```
orangehrm@workspace:~ sudo dnf install -y langpacks-en glibc-all-langpacks
unzip vim wget make curl
```

Please follow the steps given below to complete the installation. This will download the OrangeHRM automated middleware installer.

```
orangehrm@workspace:~ wget
https://alge.orangehrm.com/downloads/ohrm-middleware-installer-rhel
orangehrm@workspace:~ unzip ohrm-middleware-installer-rhel
orangehrm@workspace:~ cd ohrm-middleware-installer
orangehrm@workspace:~ sudo bash ohrm-deploy.sh
```

IMPORTANT

OrangeHRM support will promptly notify you if RabbitMQ installation is needed. If required, please refer to the RabbitMQ Installation Guide.

Note:
Before the script is initiated, you will be asked the following questions (Please answer "yes" to all the questions listed below):

- Do you wish to install Apache 2.4? (yes/no)
- Do you wish to install MariaDB 10.4? (yes/no)
- Do you wish to install PHP 7.4? (yes/no)
- Do you wish to run the installation verification script upon completion? (yes/no)

Upon execution of the MariaDB installation, you will be asked a set of general questions. You can answer them based on your preferences.(We have included suggested inputs below).

- Enter current password for root (enter for none): **press enter**
- Switch to unix socket authentication? [Y/n] : **n**
- Change the root password? [Y/n] : **Y**
    - We recommend creating a strong password
- Remove anonymous users? [Y/n] : **Y**
- Disallow root login remotely? [Y/n] : **Y**
- Remove the test database and access to it? [Y/n] : **Y**
- Reload privileged tables now? [Y/n] : **Y**

After completing the installation (End of script execution),

- Go to ***/etc/php.ini*** and change **date.timezone** value as per the region
    - Eg: If server resides in Asia, you can set **asia/colombo**
    - Default value of the **date.timezone** is set as **GMT**
    - Refer to the following php supported time zone list
      https://www.php.net/manual/en/timezones.php

- Please refer to the section: Apache virtual host configuration and complete the Apache virtual host configurations.

2.3 Installation script general troubleshooting

This section describes the general use case scenarios which may lead to script failure and how to recover from those failures

1. Installation will fail due to the absence of internet

   All the middleware dependencies included in the installation script will connect to their official repositories and download the necessary packages. Therefore, the particular server should have access to the internet in order to facilitate this.
2. Installation may fail if the recommended server specification is not available (especially disk capacity)

   Ensure disk capacity is sufficient to accommodate the installation of the required middleware services (refer to the Hardware Requirements section in ***OrangeHRM-HW-SW-Requirements-Guide***)
3. Installation may fail due to the server's outbound traffic restrictions imposed by firewalls

   Installation requires both ports 443 and 80 to be open to download all the necessary packages.
4. Installation will fail if the necessary permission is not given to the user executing the script

   The person executing the script should have root level execute permission to run the script. This is required because all the dependencies will be installed via dnf repository manager and the script attempts to update MySQL/PHP/Apache configuration files.

5. Installation script may fail to complete if any of the following services are already present in the environment:
   - Apache
   - MariaDB
   - PHP

   It is recommended to run this installation script on a vanilla server environment rather than running it on a server where the above services may be partially installed.

3. Apache virtual host configuration

This section explains how apache virtual host configurations should be done before installing the OrangeHRM Application.

1. Create a directory in web root to host the OrangeHRM application

```
orangehrm@workspace:~ sudo mkdir /var/www/html/orangehrm
```

2. Modifying Apache configuration of the Web server allows support for .htaccess files.

    a. Create orangehrm.conf

```
orangehrm@workspace:~ sudo vim /etc/httpd/conf.d/orangehrm.conf
```

    b. Add the following content into the orangehrm.conf file, save and exit

```
<Directory /var/www/html/orangehrm/symfony/web>
    Options FollowSymLinks
    AllowOverride All
    Order allow,deny
    allow from all
</Directory>
```

Above is an example. You may need to alter as per your environment details.

3. Configure the Apache Virtual Host for OrangeHRM with SSL connections. (*Below is an example. You may need to alter as per your environment details*)

    a. Edit orangehrm.conf

```
orangehrm@workspace:~ sudo vim /etc/httpd/conf.d/orangehrm.conf
```

b. Add the following content:

```
<VirtualHost *:443>
  ServerAdmin webmaster@orangehrm.com
  DocumentRoot /var/www/html/orangehrm/symfony/web
  ServerName <web site domain>
  SSLCertificateFile <public key path>
  SSLCertificateKeyFile <private key path>
  SSLCACertificateFile <CA certificate path>
  SSLEngine on
  SSLProtocol -all +TLSv1.2
  RequestHeader unset Client-IP
  RequestHeader append Client-IP "%{REMOTE_ADDR}s"
  Header edit Set-Cookie ^(.*)$ $1;SameSite=none
</VirtualHost>
```

c. Reload Apache service

```
orangehrm@workspace:~ sudo httpd -t
orangehrm@workspace:~ sudo service httpd reload
```

IMPORTANT

Installing the OrangeHRM application is the responsibility of the OrangeHRM Support Services team. Please contact your OrangeHRM Sales representative or Account Manager for assistance after completing the Middleware installation and Apache setup, or if you have any issues during the automated platform preparation process.

4. Software Platform Preparation - Security Enhancements

The following rules must be adhered to in order to reduce the risk of an external attack. These rules prevent outsiders from gaining access to the system or overloading servers making them vulnerable to malicious and harmful attacks. However, you can use any other commercial tools (if preferred) and conduct the assessments listed below.

4.1 General Recommendations

4.1.1 Disable Directory Listing and Special File Access

The “symfony” framework used in OrangeHRM itself supports the protection of the business logic from unauthorized access. It is done by making all the directories inaccessible by URL if the document root of the application is set to symfony/web directory. But in order to ensure external parties cannot access files outside the web directory, we have to make sure the following configurations are added to the server.

Add the following tag to the `httpd.conf` and then reload the httpd server

```
<Directory "give the path to OrangeHRM root directory">
    Options -Indexes
</Directory>
```

If you need to reproduce the above mentioned security risk, remove the minus(-) from the parameter -Indexes in apache2.conf. This will enable directory listing and you can navigate through the file structure with a browser.

Append the following best practice recommendations to the httpd.conf file

```
# Allowing .htaccess
<Directory _ORANGEHRM_ROOT_DIR_>
  AllowOverride All
</Directory>
# Limiting .svn listing
<DirectoryMatch .*\.svn/.*>
   Order allow,deny
   Deny From All
```

```
</DirectoryMatch>
# Restricting .YML files
<Files ~ "\.yml$">
 Order allow,deny
 Deny from all
</Files>

DirectoryIndex index.html index.php
TraceEnable Off
```

Add the following configurations under the log_config_module

```
LogFormat "%h %l %u %t \"%m %U\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\""
combined
LogFormat "%h %l %u %t \"%m %U\" %>s %b" common

<IfModule logio_module>
# You need to enable mod_logio.c to use %I and %O
   LogFormat "%h %l %u %t \"%m %U\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" %I
%O" combinedio
</IfModule>
```

After defining the logs please make sure to add log rotation for your apache access and error logs.

4.1.2 Block Unnecessary Ports

Use iptables/firewalld to enforce the firewall security on the server. OrangeHRM recommends that, as a default list, only the following ports should be kept open. However, based on clients' requirements it can be determined what other legitimate ports need to be opened.

In-Bound	Outbound
HTTP/HTTPS - 80/443	**HTTP/HTTPS** - 80/443
SSH - 22 or any other custom port if exists	**SMTP** - 587 or 465 (Decide based on the application mail configuration)

ICMP - (If required 3rd party hosting providers and/or monitoring tools. Allow only to specific IPs)	**ICMP** - (If required 3rd party hosting providers and/or monitoring tools. Allow only to specific IPs)
	SFTP - 22 or any other custom port if exists
	DNS - Default port 53 (UDP)
	LDAP/LDAPS - 636 or 389 (Decide based on the LDAP protocol)
	NTP - Default port 123 (UDP)

4.1.3 Stop Unnecessary Applications

Stop already running services which are not required and turn them off from `systemctl` to remove them from the startup services list
You can use the following command to view the list of services available on the server.

```
orangehrm@workspace:~ sudo systemctl list-units
```

4.1.4 Security - Patch Updates

This section will explain how to automate security updates in installed packages. [Only the packages installed through repositories will be affected. Any package/ binary or any module which was installed manually will have to be patched manually]

1. Install dnf-automatic

```
orangehrm@workspace:~ sudo dnf install -y dnf-automatic
```

2. Configure dnf-automatic tool to update security patches (Recommended)

    a. Open the dnf-automatic config file and change upgrade_type value to security and change the apply_updates to yes

```
orangehrm@workspace:~ vim /etc/dnf/automatic.conf
```

```
#Change update_cmd = default to update_cmd= security
upgrade_type = security

#Change apply_updates = no to apply_updates= yes
apply_updates = yes
```

b. Enable the service

```
orangehrm@workspace:~ sudo systemctl enable --now dnf-automatic.timer
```

4.2 Security - Access Control Related Tasks

4.2.1 Restrict SSH 'root' Access

DO NOT grant ssh access to root/regular users unless it is requested.

4.2.2 Configure SSH ACL

Restrict SSH access to production servers to only support engineers/system administrators/DBAs using SSH ACL.

4.2.3 Configure User Groups

Group 1 : Application Maintenance team (OrangeHRM) should have

1. Privileges to create, read, delete and update to OrangeHRM code directory.
2. Stop, Start, Reload, Config test Apache and MariaDB services
3. Edit MariaDB, Apache and php configuration
4. Execute mysqldump command with routines option

4.2.4 Restrict FTP

FTP users should be created as non-privileged users. Disable the transfer of files from a production server using FTP. Instead, use SFTP services.

4.2.5 Enable security headers for Apache

a. Content Security policy is another Apache header which needs to be handled at the server level (add the policy as client requires. This is not a mandatory requirement for OrangeHRM app installation)

b. Following header (HPKP) is not required for the OHRM application to function properly. However, it is recommended to enhance infrastructure security.
    i. Click on this link to configure.

4.2.6 Enabling LDAPS

If you are planning to set up an LDAPS connection you need to have the CA certificate of the LDAP Instance and perform the following steps. This section considers that you have an Active Directory-Based LDAP implementation.

1. If the certificate is available please proceed to the 'Linux based client' section below
2. Alternatively, to obtain this for the Windows AD server. Please follow the steps below
    - Click Start -> Click on administrative tools -> issued certificates
    - Right-click on the root certificate chain - > click properties
    - General Tab -> Select the required CA cert -> view certificate
    - Select details tab
    - Click on "*Copy to file*" -> Click Next -> Select "*Base-64 encoded X.509*" -> click on Next
    - Provide a filename and save

Linux Based Client

- Then upload the CA cert [Which you have acquired from the above step] into the respective server.
    - Location of the certificate : /etc/openldap/certs
    - Create a symlink for the certificate. Make sure symlink name adheres to the following format
        - <CA Hash value>.0
        - CA Hash value can be generated using the following command
            - openssl x509 -noout -hash -in <ca cert file
- Edit /etc/openldap/ldap.conf and add following snippet
    - TLS_CACERTDIR /etc/openldap/cacerts
    - If you encounter the following line in the ldap.conf comment it out
        - #TLS_REQCERT never

4.3 Security - Data Access Related Tasks

4.3.1 Define sudo Aliases Permissions

Define sudo aliases and permissions to execute for the user groups, according to organizational policies.

4.3.2 Restrict Execution Permissions for Selected Linux Commands

Remove execution permissions from regular users for the following Linux commands:

- cp (except from and to within home directory)
- rm
- mv (except from and to within home directory)
- mysqldump
- mtools
- ftp
- sftp
- scp

4.3.3 Disable Telnet (If Enabled)

1. SSH into the server and login as root.
2. In the shell command prompt, type the following command and press Enter.

```
orangehrm@workspace:~ sudo service telnet.socket stop
orangehrm@workspace:~ sudo systemctl disable telnet.socket
```

4.4 Security - Audit Related Tasks

4.4.1 Setup a Preferred Audit Tool (optional)

I.e MariaDB audit plugin
https://mariadb.com/kb/en/library/mariadb-audit-plugin/

Also, we would recommend configuring an Intrusion Detection System such as PHPIDS which will add an additional layer of security.

4.4.2 Post Security Scanning

This section references the set of tools and services which help to identify improper security configurations and vulnerabilities which may exist in the server

- **Lynys - Host based Scanning tool**

  This tool will detect improper middleware and OS level configurations in the host.

  - Tool is available in RHEL repositories. If this is not available in the host, please download it as follows:

```
orangehrm@workspace:~ sudo dnf install lynis
```

  - Then execute the following command to run the system audit

```
orangehrm@workspace:~ sudo lynis audit system
```

- **Qualys web security tool**

  This tool will evaluate the SSL configuration of the server and provide recommendations upon results.

  - This is an online tool. Refer to the URL below
    - **https://www.ssllabs.com/ssltest/**
  - Provide a URL of your instance and let the tool scan the configuration
  - Try to achieve the A+ Grade by applying the amendments recommended by the tool

NOTE

None of the above settings are mandatory for OrangeHRM applications to be deployed. However, these configurations are needed in order to enhance the security of the deployment environment.

4.5 Service Monitoring

4.5.1 Setup a Preferred Monitoring Tool

We recommend adding the OrangeHRM deployed server to an infrastructure monitoring tool such as Datadog, Zenoss, New Relic etc.

- New Relic - https://newrelic.com/
- Zenoss - https://www.zenoss.com/

If the servers are running on VMWare or Hyper-V or any other similar environment, the monitoring of servers (host machine) should also be enabled.

We would recommend monitoring the following parameters and setting up alerts that would trigger if the parameters are exceeded based on predefined threshold values .

- CPU
- Memory
- Disk Usage
- Load Average
- MySQL Connections
- Network interface errors/Utilization

4.6 Configure NGINX Reverse Proxy [optional]

Go through the following steps to set up Nginx:

1. Stop existing Apache service (if the service is running)

```
orangehrm@workspace:~ sudo service httpd stop
```

2. Install nginx

```
orangehrm@workspace:~ sudo dnf install nginx
```

3. Create nginx proxy

```
orangehrm@workspace:~ sudo touch /etc/nginx/conf.d/orangehrm.conf
```

4. Copy the following example configuration file and replace its placeholders with required values as appropriate

```
server {
    listen 443 ssl;
    server_name "<domain name>";

    ssl_certificate           <SSL certification location>;
    ssl_certificate_key       <SSL key location>;

    location / {

        limit_except GET POST PUT HEAD DELETE PATCH OPTIONS {
             deny   all;
        }

        proxy_pass https://<localhost or local IP>:<httpd port>;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_set_header Client-IP $remote_addr;
    }
}
```

5. Edit nginx.conf file and update the server block which listens to http protocol (refer to the example given below). This modification will direct all the HTTP web traffic that comes through port 80 into HTTPS.

```
server {
        listen 80 default_server;
        listen [::]:80 default_server;
        server_name _;
        return 301 https://$host$request_uri;
}
```

6. Add the following configurations into the /etc/nginx/nginx.conf file

```
keepalive_timeout 60
gzip on
```

7. Start nginx.

```
orangehrm@workspace:~ sudo service nginx start
```

8. Update Apache listening port (ensure original configuration file is backed up before editing)

```
orangehrm@workspace:~ sudo vim /etc/httpd/conf.d/ssl.conf
```

   a. Change the current listening port to 8080 (if 8080 is unavailable change the port number accordingly)

```
Listen 8080 https
```

   b. Go to the OrangeHRM applications vhost file and change the listen port to 8080

```
<VirtualHost *:8080>
  ServerAdmin webmaster@orangehrm.com
  DocumentRoot _ORANGEHRM_ROOT_DIR_/symfony/web
  ServerName <web site domain>
  SSLCertificateFile <public key certificate path>
  SSLCertificateKeyFile <key certificate path>
  SSLCACertificateFile <CA certificate path>
  SSLEngine on
  RequestHeader unset Client-IP
```

```
  RequestHeader append Client-IP "%{REMOTE_ADDR}s"
</VirtualHost>
```

c. Start the Apache service

```
orangehrm@workspace:~ sudo service httpd start
```

9. Go to nginx proxy file (/etc/nginx/sites-available/orangehrm.conf) and change proxy port to 8080

```
proxy_pass https://<localhost or local IP>:8080;
```

NOTE

If nginx is used as a reverse proxy, make sure that you remove the ***Client-IP HTTP header*** from the Apache level, if it is configured at the apache level.

4.7 Data Backups

We recommend you keep regular server backups. The preferred retention period is 2 weeks.
It is preferable to have separate backups for file-level backups and MariaDB/MySQL backups as it will be more convenient for data recovery purposes.

4.8 SELinux Settings

4.8.1 Enable permissive mode in Selinux

1. Setting SELinux to Permissive mode permanently

   a. Open /etc/selinux/config file

   b. Change the SELINUX value to “SELINUX=permissive”

   c. Edit the kernel boot line and append “enforcing=0” to the kernel boot options

   (Assuming SELinux is not set to disabled as in the section above).

2. Restart the device

3. To check the status of SELinux execute sestatus command

Appendix

Appendix A

Create SSL certificates with openssl

This section describes the steps on how to create a private key and generate a certificate signing request using openssl.

A.1 Create a certificate signing request (CSR)

SAN (subject alternative name) must be referenced in the CSR. Create a file called **san.cnf** and add the following content. Replace the content according to the organization's specific requirements.

```
[ req ]
default_bits       = 2048
distinguished_name = req_distinguished_name
req_extensions     = req_ext
prompt = no
[ req_distinguished_name ]
countryName                = <COUNTRY_NAME>
stateOrProvinceName        = <STATE_OR_PROVINCE_NAME>
localityName               = <LOCALITY_NAME>
organizationName           = <ORGANIZATION_NAME>
commonName                 = <COMMON_NAME>
[ req_ext ]
subjectAltName = @alt_names
[alt_names]
DNS.1   = <DNS_ONE>
DNS.2   = <DNS_TWO>
DNS.3   = <DNS_THREE>
```

- <COUNTRY_NAME> = Country name.(two letter code - US, UK, etc)
- <STATE_OR_PROVINCE_NAME> = State or Province Name (full name)
- <LOCALITY_NAME> = Locality Name (eg, city)
- <ORGANIZATION_NAME> = Organization Name (eg, company)
- <COMMON_NAME> = Common Name (e.g. server FQDN or YOUR name)

- <DNS.1> = <COMMON_NAME> (e.g. hrm.example.com)

Complete the **alt_name** section with your additional DNS names (subdomains). If you only have one DNS name, then remove the DNS.2 (e.g. hrm2.example.com) and DNS.3 (e.g. hrm3.example.com) records.

- Run the command below to generate the private key and CSR (Certificate Signing Request).

```
orangehrm@workspace:~ openssl req -out <EXAMPLE.COM>.csr -newkey rsa:2048
-nodes -keyout <EXAMPLE.COM>.key -config san.cnf
```

- Use the following command to generate the CSR if you already have a private key.

```
orangehrm@workspace:~ openssl req -new -out <EXAMPLE.COM>.csr -key
<EXAMPLE.COM>.key -config san.cnf
```

- You can verify your CSR by checking that the CSR contains the SAN, which you specified in the san.cnf file.

```
orangehrm@workspace:~ openssl req -noout -text -in <EXAMPLE.COM>.csr | grep
DNS
```

A.2 Submit the CSR to a Certificate Authority

- Certificates should be signed by a trusted certificate authority.
- You should submit the generated CSR to your certificate authority and get the signed certificate.
- Make sure to export the signed certificate in base-64 format. This will be your public key.

A.3 Verify the certificates

- Private key and public key should be compatible with each other. Use the following command to verify the compatibility. If outputs are identical from each command then the keys are compatible.

```
orangehrm@workspace:~ openssl x509 -noout -modulus -in <public key> |
openssl md5
orangehrm@workspace:~ openssl x509 -noout -modulus -in <private key> |
```

```
openssl md5
```

- Verify the subject alternative name and common name are compatible with the application's domain name. Use the following command to verify the compatibility.
    - For the Subject alternative name
      If the output is received with DNS records which match the application's domain name then SSL certificate's subject alternative name should be compatible.

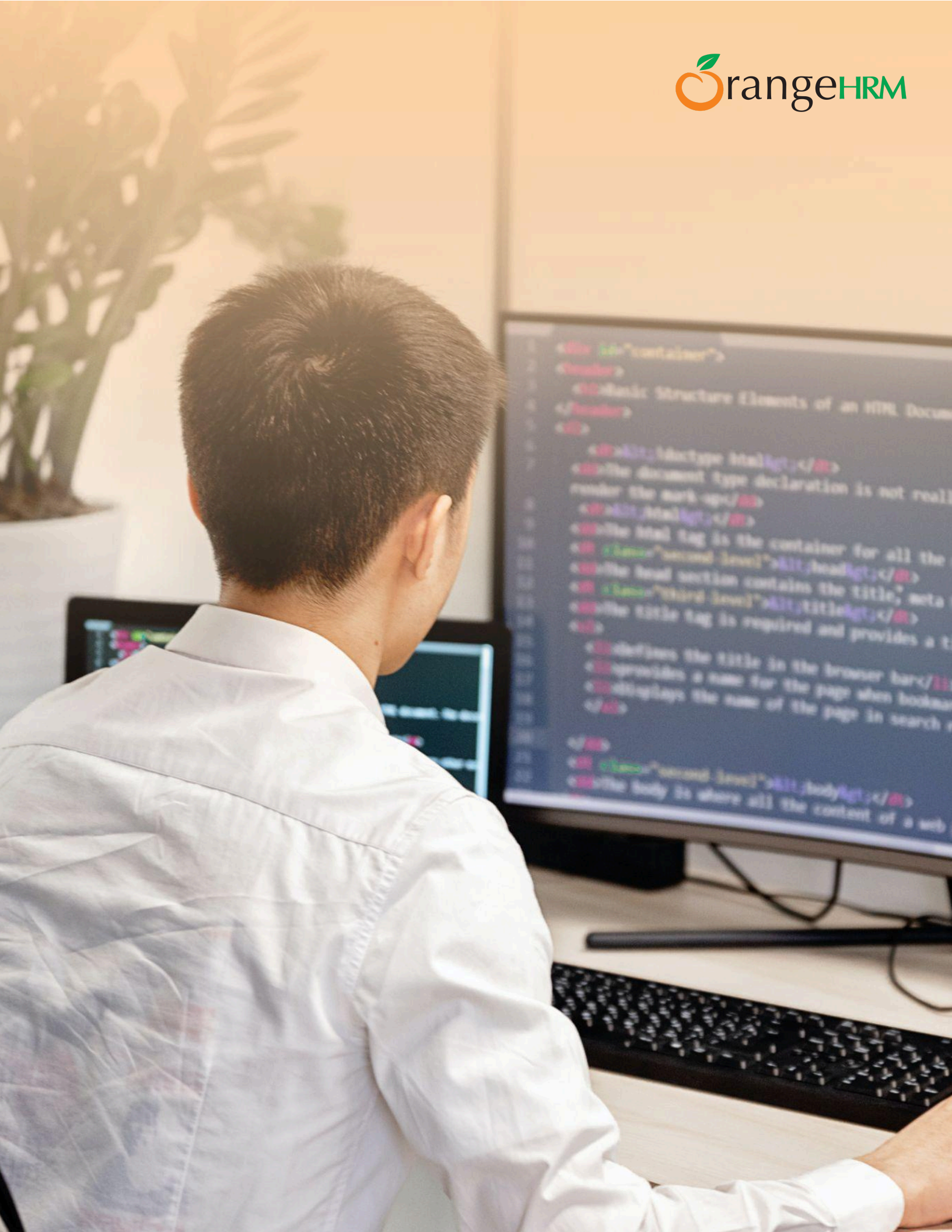
```
orangehrm@workspace:~ openssl x509 -text -noout -in <public key> | grep DNS
```

    - For the common name
      If the CN value of the output is compatible with the application's domain name then SSL certificate's common name should be compatible.

```
orangehrm@workspace:~ openssl x509 -noout -subject -in <public key>
```

NOTE

- If your certificate authority (CA) is an intermediate certificate authority, then make sure to create a single CA certificate including the public certificates of each CA.
- Intermediate CA certificates should be arranged in the correct order. Root CA certificate should be added to the bottom of the single CA certificate created.


OrangeHRM