IBM Process Mining On-Premise Installation Guide

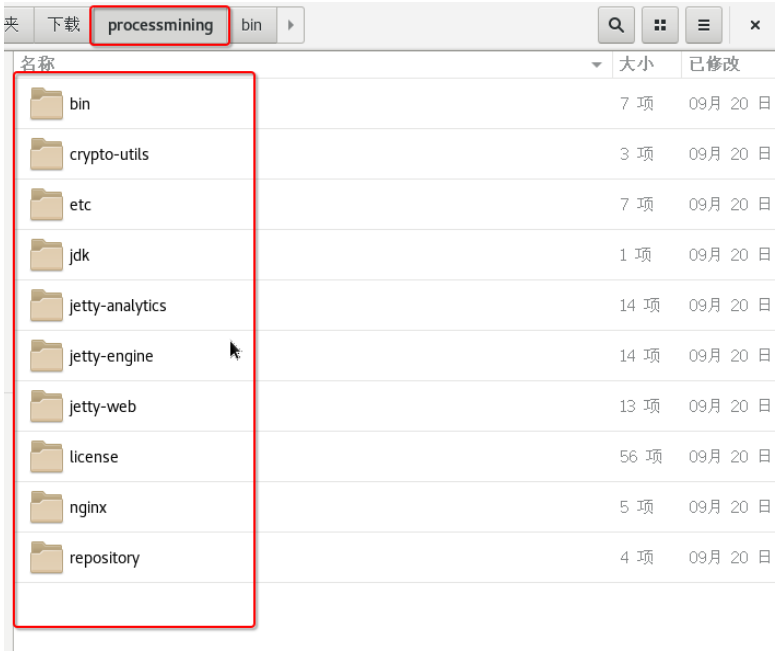
Validated with IBM Process Mining v1.9 & v1.12

Yong Qiang Zhao
Bu Feng Hou,
Zhong Tao Gao

Download & Unzip Installation Image

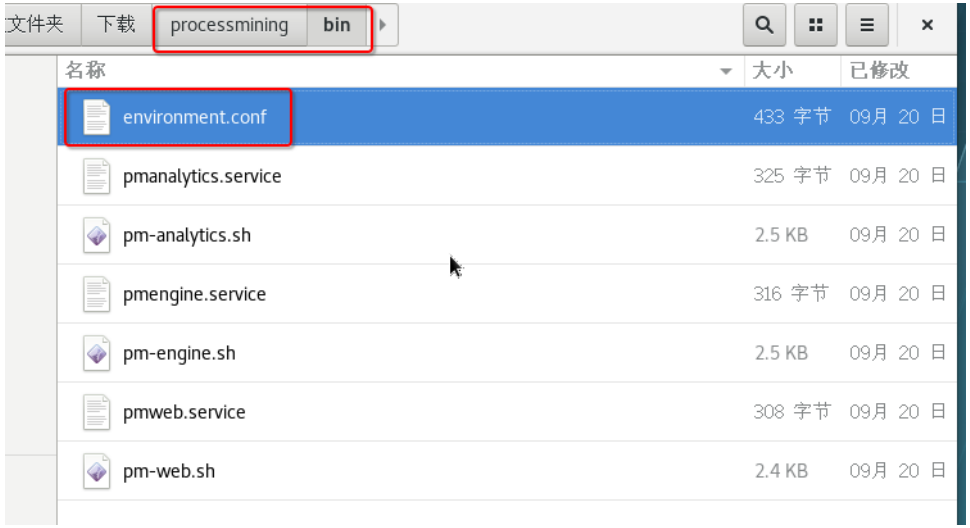
1. Download Process Mining Installation Image
   The latest version is v1.12, the guide is based on v1.9

2. Unzip Installation Image



Modify environment.conf

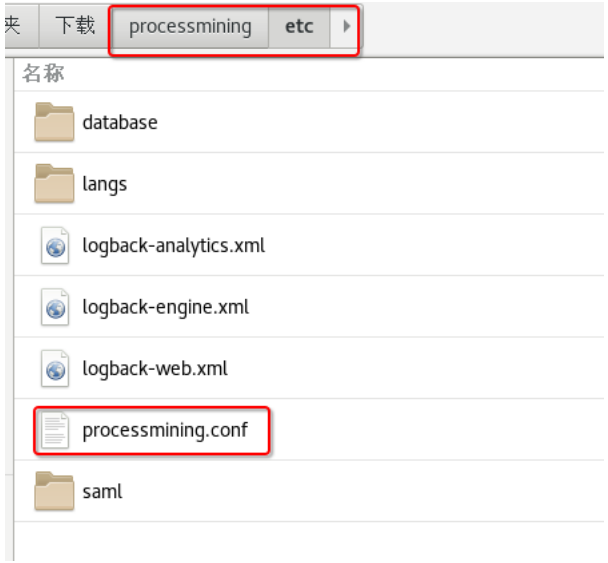
1. Edit the server configuration file, <PM_HOME>/bin/environment.conf, to match the installation folders:
    1. Change the owner of <PM_HOME> to the RUNAS user, for example, myuser: sudo chown -R myuser:myuser <PM_HOME>/
    2. PM_HOME=/opt/processmining
    3. TMPDIR: TMPDIR=/opt/processmining/repository/temp




```
RUNAS=root
JAVA_HOME=../jdk/linux/jdk8u282-b08
PM_HOME=/opt/processmining
TMPDIR=/opt/processmining/repository/temp

BIND_HOST=0.0.0.0
HTTP_PORT=8080
HTTPS_PORT=9443

BIND_HOST_ENGINE=127.0.0.1
HTTP_PORT_ENGINE=8070
HTTPS_PORT_ENGINE=7443

HTTP_PORT_ENGINE_ANALYTICS=9070
HTTPS_PORT_ENGINE_ANALYTICS=9071

JVM_MAX_HEAP=2g
JVM_MAX_HEAP_ENGINE=8g

JVM_MAX_DIRECT_MEMORY=512m
JVM_MAX_DIRECT_MEMORY_ENGINE=1g

SEC_DEVICE=legacy
```

Modify processmining.conf

1. Change to installer etc folder
2. Modify filesystem.home to point to data folder




```
##########################################
# system config section
##########################################

filesystem.home: "/opt/processmining/repository/data/",

##########################################
# database
##########################################

persistence: {
  mongodb: {
    database: "processmining",
    host: "127.0.0.1",
    port: 27017,
    user: "processmining",
    password: "",

    ssl: {
      enabled: false,
      trustStore: "",
      trustStorePassword: "",
      keyStore: "",
      keyStorePassword: ""
    }
  }
},

##########################################
# email SMTP
##########################################

email: {
```

Install Mongodb & shell client

1. MongoDB version 3.6 is the required database
2. A MongoDB instance can be installed in the following ways
   - ✓ **Install the Community Edition on the same server as the application or a separated DB server**
   - ✓ Install the Enterprise Edition on the same server as the application or a separated DB Server
   - ✓ Subscribe to the MongoDB Atlas service on AWS Cloud (the same region as the application)
3. Download v3.6 and install it, certainly, you can also use other approaches to install the different version of mongodb as long as it is v3.6 and above.

```
[root@odm2 mongodb]# ls
mongodb-org-server-3.6.23-1.el7.x86_64.rpm  mongodb-org-shell-3.6.23-1.el7.x86_64.rpm
[root@odm2 mongodb]# rpm -ivh mongodb-org-s*
warning: mongodb-org-server-3.6.23-1.el7.x86_64.rpm: Header V3 RSA/SHA1 Signature, key ID 91fa4ad5: NOKEY
Preparing...                          ################################# [100%]
Updating / installing...
   1:mongodb-org-shell-3.6.23-1.el7   ################################# [ 50%]
   2:mongodb-org-server-3.6.23-1.el7  ################################# [100%]
Created symlink from /etc/systemd/system/multi-user.target.wants/mongod.service to /usr/lib/systemd/system/mongod.service.
[root@odm2 mongodb]#
```

Start MongoDB and check its status

```
[root@odm2 mongodb]# systemctl start mongod
[root@odm2 mongodb]#
```

```
[root@odm2 mongodb]# systemctl status mongod
● mongod.service - MongoDB Database Server
   Loaded: loaded (/usr/lib/systemd/system/mongod.service; enabled; vendor preset: disabled)
   Active: active (running) since Thu 2021-07-15 15:59:42 CST; 1min 6s ago
     Docs: https://docs.mongodb.org/manual
  Process: 18790 ExecStart=/usr/bin/mongod $OPTIONS (code=exited, status=0/SUCCESS)
  Process: 18786 ExecStartPre=/usr/bin/chmod 0755 /var/run/mongodb (code=exited, status=0/SUCCESS)
  Process: 18782 ExecStartPre=/usr/bin/chown mongod:mongod /var/run/mongodb (code=exited, status=0/SUCCESS)
  Process: 18778 ExecStartPre=/usr/bin/mkdir -p /var/run/mongodb (code=exited, status=0/SUCCESS)
 Main PID: 18793 (mongod)
    Tasks: 24
   CGroup: /system.slice/mongod.service
           └─18793 /usr/bin/mongod -f /etc/mongod.conf

Jul 15 15:59:41 odm2 systemd[1]: Starting MongoDB Database Server...
Jul 15 15:59:41 odm2 mongod[18790]: about to fork child process, waiting until server is ready for connections.
Jul 15 15:59:41 odm2 mongod[18790]: forked process: 18793
Jul 15 15:59:42 odm2 mongod[18790]: child process started successfully, parent exiting
Jul 15 15:59:42 odm2 systemd[1]: Started MongoDB Database Server.
[root@odm2 mongodb]#
```

Enable mongoDB security authorization

1. Edit /etc/mongo.conf and enable security. Certainly, you can skip this step if you don't want to enable mongoDB security

```
# how the process runs
processManagement:
  fork: true  # fork and run in background
  pidFilePath: /var/run/mongodb/mongod.pid  # location of pidfile
  timeZoneInfo: /usr/share/zoneinfo

# network interfaces
net:
  port: 27017
  bindIp: 127.0.0.1  # Enter 0.0.0.0,:: to bind to all IPv4 and IPv6 addresses or, al

security:
  authorization: enabled
#operationProfiling:

#replication:

#sharding:

## Enterprise-Only Options

#auditLog:

#snmp:
```

Create MongoDB and user for Process Mining

Run mongoDB shell console to create admin user and database for process mining

- use admin
- db.createUser({user:"admin", pwd:"passw0rd", roles:[{role:"userAdminAnyDatabase", db:"admin"}]})
- use processmining
- db.createUser({user:"admin", pwd:"passw0rd", roles:[{role:"dbOwner", db:"processmining"}]})

```
[root@odm2 etc]# mongo
MongoDB shell version v3.6.23
connecting to: mongodb://127.0.0.1:27017/?gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("1c9f2ad8-7092-4290-b88c-7fb1ab7db7ac") }
MongoDB server version: 3.6.23
Server has startup warnings:
2021-07-15T15:59:42.698+0800 I CONTROL  [initandlisten]
2021-07-15T15:59:42.698+0800 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2021-07-15T15:59:42.698+0800 I CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
2021-07-15T15:59:42.698+0800 I CONTROL  [initandlisten]
2021-07-15T15:59:42.699+0800 I CONTROL  [initandlisten]
2021-07-15T15:59:42.699+0800 I CONTROL  [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/enabled is 'always'.
2021-07-15T15:59:42.699+0800 I CONTROL  [initandlisten] **        We suggest setting it to 'never'
2021-07-15T15:59:42.699+0800 I CONTROL  [initandlisten]
2021-07-15T15:59:42.699+0800 I CONTROL  [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/defrag is 'always'.
2021-07-15T15:59:42.699+0800 I CONTROL  [initandlisten] **        We suggest setting it to 'never'
2021-07-15T15:59:42.699+0800 I CONTROL  [initandlisten]
> use admin
switched to db admin
> db.createUser({user:"admin", pwd:"passw0rd", roles:[{role:"userAdminAnyDatabase", db:"admin"}]})
Successfully added user: {
        "user" : "admin",
        "roles" : [
                {
                        "role" : "userAdminAnyDatabase",
                        "db" : "admin"
                }
        ]
}
>
```

1. Check /etc/selinux/config and make sure selinux is disabled, you need to restart the system to effect the change

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of three values:
#     targeted - Targeted processes are protected,
#     minimum - Modification of targeted policy. Only selected processes are protected.
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

2. Restart mongodb by running command below
   - systemctl restart mongod

3. Retrieve encrypted mongodb password string, you can skip this step if mongoDB security is not enabled.

   /opt/processmining/crypto-utils/crypt-utils.sh passw0rd

```
[root@absinth1 processmining]# ls
bin           etc  jetty-analytics  jetty-web  nginx
crypto-utils  jdk  jetty-engine     license    repository
[root@absinth1 processmining]# cd crypto-utils/
[root@absinth1 crypto-utils]# ls
crypt-utils.bat  crypt-utils.sh  lib
[root@absinth1 crypto-utils]# ./crypt-utils.sh passw0rd
String To Encrypt: passw0rd
Encrypted String:KSx+W1ICw9VoMsGoW6pFZw==
[root@absinth1 crypto-utils]#
```

4. Configure mongodb connection in ./processmining/etc/processmining.conf, make sure the persistence mongodb parameters are the same as you created in previous steps

```
########################################
# system config section
########################################

filesystem.home: "/opt/processmining/repository/data/",

########################################
# database
########################################

persistence: {
  mongodb: {
    database: "processmining",
    host: "127.0.0.1",
    port: 27017,
    user: "admin",
    password: "KSx+W1ICw9VoMsGoW6pFZw==",

    ssl: {
      enabled: false,
      trustStore: "",
      trustStorePassword: "",
      keyStore: "",
```

5. Start process mining by running follow commands
   - `./processmining/bin/pm-web.sh start`
   - `./processmining/bin/pm-engine.sh start`
   - `./processmining/bin/pm-analytics.sh start`

Note: if you saw message something like "….FAILED …" when executing above script, this might be caused by timeout issue, you can increase timeout setting in those .sh fils following instruction below,

1. Edit (for example using vi) the file pm-web.sh
2. Localize the row with export JETTY_START_TIMEOUT=120
3. Increase the value of timeout, for example 300
4. Save the file & Retry

Check and Install below packages before proceed

1. yum -y install gcc
2. yum -y install gcc-c++
3. yum install -y zlib-devel

Install nginx

1. Create nginx.repo

   sudo vi /etc/yum.repos.d/nginx.repo

2. Paste the following lines

   [nginx]
   name=nginx repo baseurl=http://nginx.org/packages/mainline/rhel/7/$basearch/ gpgcheck=0
   enabled=1

3. Install and start nginx

   sudo yum update
   sudo yum install nginx
   systemctl enable nginx
   systemctl start nginx

4. Copy the following VirtualHost file

   sudo mv /etc/nginx/conf.d/default.conf /etc/nginx/conf.d/default_origin.conf
   sudo cp <PM_HOME>/nginx/processmining.conf /etc/nginx/conf.d/default.conf

Create self-signed certificate

If you have office certificate provided by a certificate authority, you can skip this step, or follow below steps to create a self-signed certificate

1. Create a folder to store certificate
   Sudo mkdir /etc/nginx/ssl

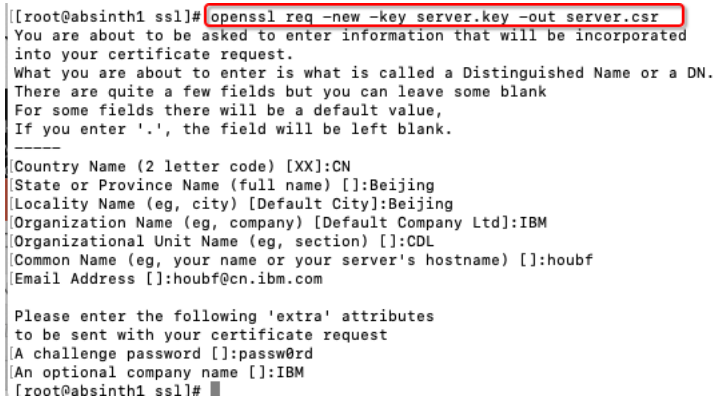
2. Create nginx SSL certificate at folder /usr/local/webserver/nginx
   - openssl genrsa -out server.key 2048

```
[root@absinth1 ssl]# openssl genrsa -out server.key 2048
Generating RSA private key, 2048 bit long modulus
.....+++
.........................................+++
e is 65537 (0x10001)
[root@absinth1 ssl]#
```

Create self-signed certificate(cont...)

If you have office certificate provided by a certificate authority, you can skip this step, or follow below steps to create a self-signed certificate

3. Generate SSL certificate file
   - openssl req -new -key server.key -out server.csr



Create self-signed certificate(cont...)

If you have office certificate provided by a certificate authority, you can skip this step, or follow below steps to create a self-signed certificate

4. Create Self-signed certification based on SSL certificate file
   - openssl x509 -req -in server.csr -out server.crt -signkey server.key -days 3650

```
[root@absinth1 ssl]# openssl x509 -req -in server.csr -out server.crt -signkey server.key -days 3650
Signature ok
subject=/C=CN/ST=Beijing/L=Beijing/O=IBM/OU=CDL/CN=houbf/emailAddress=houbf@cn.ibm.com
Getting Private key
[root@absinth1 ssl]#
```

5. Remove key password
   - openssl rsa -in server.key -out server.key

```
[root@absinth1 ssl]# openssl rsa -in server.key -out server.key
writing RSA key
[root@absinth1 ssl]#
```

6. Check the certificate files as below

```
[root@absinth1 ssl]# ls -l
总用量 12
-rw-r--r-- 1 root root 1265 12月  13 19:35 server.crt
-rw-r--r-- 1 root root 1094 12月  13 19:30 server.csr
-rw-r--r-- 1 root root 1671 12月  13 19:36 server.key
[root@absinth1 ssl]#
```

Apply Certificates

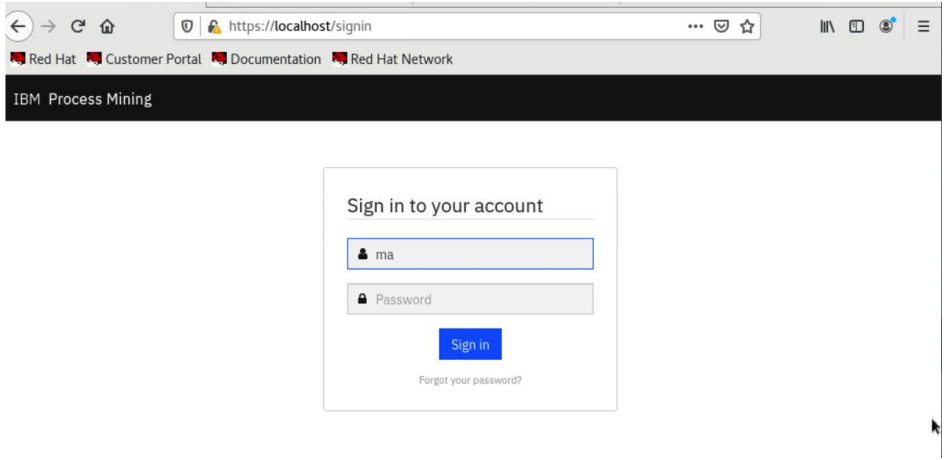
1. Edit the VirtualHost - /etc/nginx/conf.d/default.conf
   vi /etc/nginx/conf.d/default.conf

2. Set the correct certificate files by changing the properties you created above. Or you can use your official ceretificate
   ssl_certificate
   ssl_certificate_key

```
# dDOS slow-body mitigate attack
#limit_req_zone $binary_remote_addr zone=one:10m rate=30r/m;
#limit_conn_zone $binary_remote_addr zone=addr:10m;

server {
        listen   443 ssl;
        server_name _; #for production environment replace _ with the name of your host
        keepalive_timeout   70;
        server_tokens off;

        ssl on;
        ssl_certificate /etc/nginx/ssl/server.crt;
        ssl_certificate_key /etc/nginx/ssl/server.key;

                ssl_session_timeout 1d;
                ssl_session_cache shared:SSL:50m;
                ssl_session_tickets off;


                # modern configuration. tweak to your needs.
                ssl_protocols TLSv1.2;
                ssl_ciphers 'ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20
-POLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384:ECDHE-ECDSA-AES128-SHA256';
                ssl_prefer_server_ciphers on;
```

Restart nginx

1. Stop nginx
   - systemctl stop nginx
2. Start nginx
   - systemctl start nginx
3. Verify if nginx and process mining works by accessing https://ProcessMining Server



User Name：maintenance.admin
Initial password：pmAdmin$1

First Login Setup

- Follow instructions to setup user and tenants: https://www.ibm.com/docs/en/cloud-paks/1.0?topic=guide-application-administration

How to check if Process Mining has been started

Check if ProcessMining has been started successfully using **curl**

1. Curl -v http://localhost:8080

```
[root>curl -v http://localhost:8080
* About to connect() to localhost port 8080 (#0)
*   Trying 127.0.0.1...
* Connected to localhost (127.0.0.1) port 8080 (#0)
> GET / HTTP/1.1
> User-Agent: curl/7.29.0
> Host: localhost:8080
> Accept: */*
>
< HTTP/1.1 302 Found
< Access-Control-Allow-Origin: domain
< Access-Control-Allow-Methods: POST, GET, OPTIONS, DELETE
< Access-Control-Max-Age: 3600
< Access-Control-Allow-Headers: *
< Set-Cookie: XSRF-TOKEN=30e23182-b488-43e6-b20b-e1a987440548;Path=/
< Expires: Thu, 01 Jan 1970 00:00:00 GMT
< Content-Language: en
< X-XSS-Protection: 1; mode=block
< X-Content-Type-Options: nosniff
< Content-Security-Policy: script-src 'self' 'unsafe-inline' 'unsafe-eval' www.google-analytics.com js.hs-scripts.com js.hs-analytics.net js.hsleadflows.net js.h
scollectedforms.net js.usemessages.com
< X-Robots-Tag: noindex
< Referrer-Policy: no-referrer
< Set-Cookie: INVENIOSID=NzkwZmEmMjMtZTQyMS00NDc4LThjMWYtODdlMzU2ZmE1Zjc5; Path=/; HttpOnly; SameSite=Lax
< Location: http://localhost:8080/signin
< Content-Length: 0
<
* Connection #0 to host localhost left intact
```

How to check if nginx will forward request to ProcessMining

Check if ProcessMining has been started successfully using **curl**

1. Curl -v –k https://localhost

```
[root>curl -v -k https://localhost
* About to connect() to localhost port 443 (#0)
*   Trying 127.0.0.1...
* Connected to localhost (127.0.0.1) port 443 (#0)
* Initializing NSS with certpath: sql:/etc/pki/nssdb
* skipping SSL peer certificate verification
* SSL connection using TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
* Server certificate:
*       subject: E=houbf@cn.ibm.com,CN=houbf,OU=CDL,O=IBM,L=Beijing,ST=Beijing,C=CN
*       start date: 12月 14 03:35:34 2021 GMT
*       expire date: 12月 12 03:35:34 2031 GMT
*       common name: houbf
*       issuer: E=houbf@cn.ibm.com,CN=houbf,OU=CDL,O=IBM,L=Beijing,ST=Beijing,C=CN
> GET / HTTP/1.1
> User-Agent: curl/7.29.0
> Host: localhost
> Accept: */*
>
< HTTP/1.1 302 Found
< Server: nginx

< Date: Fri, 17 Dec 2021 06:30:33 GMT
< Transfer-Encoding: chunked
< Connection: keep-alive
< Access-Control-Allow-Origin: domain
< Access-Control-Allow-Methods: POST, GET, OPTIONS, DELETE
< Access-Control-Max-Age: 3600
< Access-Control-Allow-Headers: *
< Set-Cookie: XSRF-TOKEN=9d749163-2fc5-4646-a234-7842f4793e67;Path=/
< Expires: Thu, 01 Jan 1970 00:00:00 GMT
< Content-Language: en
< X-XSS-Protection: 1; mode=block
< X-Content-Type-Options: nosniff
< Content-Security-Policy: script-src 'self' 'unsafe-inline' 'unsafe-eval' www.google-analytics.com js.hs-scripts.com js.hs-analytics
scollectedforms.net js.usemessages.com
< X-Robots-Tag: noindex
< Referrer-Policy: no-referrer
< Set-Cookie: INVENTOSID=YmUwMzNlZjgtZjE5Zi00NmRkLWI4MzEtNDZkYTAwOGJkODIy; Path=/; HttpOnly; SameSite=Lax
< Location: https://localhost/signin
< Strict-Transport-Security: max-age=31536000; includeSubDomains; preload
<
```

Trouble shooting

In case you can’t get Process Mining login page, please follow steps below to check the system

1. Check ./processmining/repository/logs/pm_web.log and see if any issue
2. Check mongoDB log from /var/log/mongo and see if there is database authorization issue, in case there is any process mining DB authorization issue, please follow steps at Page 8 to grant the user access
3. Check nginx log from /var/log/nginx and see if there is any access permission issue, if there is any issue like below, it is caused by SELinux most likely , you can use command “setenforce 0” to disable it

127.0.0.1:8080 failed (13: Permission denied) while connecting to upstream, client: 127.0.0.1, server: _, request: "GET / HTTP/1.1", upst ream: "http://127.0.0.1:8080/", host: "localhost"