ZEBRA installation for Mainframe platform

Mainframe Side

1) RMF customization

   a. Check if STC GPMSERVE is started (frequently resides in SYS1.PROCLIB) of DDS (Distributed Data Server) in each LPAR, if not, activate it with the following possible commands

      i. F RMF,DDS or
      ii. START GPMSERVE, MEMBER=01 (where the number and configuration of MEMBER will be explained in the next point)

   b. GPMSRVnn member configuration

      i. This member resides in SYS1.PARMLIB (or the PARMLIB customized in the installation) and the last two characters are the specifies in the STC start
      ii. The minimum basic configuration to start STC GPMSERVE is the adjuncted below, where you only have to indicate the IP of the ZEBRA server that will be scraping
      iii. By default it listens in the port 8803, as far as possible, it should be maintained, since otherwise the port defined to reach the DDS must also be adjusted in ZEBRA
      iv. Can be configured for more secure HTTP requirements, but that configuration has not been tested yet

   c. LPARs IPs identification

      i. LPR1: nnn.nnn.nnn.nn
      ii. LPR2: nnn.nnn.nnn.nn
      iii. LPR3: nnn.nnn.nnn.nn

Midrange Side

2) Have a Linux virtual server (RedHat desirable) with the following characteristics

    a. 2 processors
    b. 8 GB of memory
    c. Operating System + 20 GB of storage for apps

3) ZEBRA Installation and Configuration

    a. ZEBRA Installation

        i. For installation it is recommended to follow the steps on the site https://github.com/zowe/zebra/tree/main/Documentation/User%20Documentation

        ii. ZEBRA has two configuration files

            1. ***zconfig.json***: The LPARs to monitor and solution options are defined (with or without Prometheus, with or without MongoDB, etc.)
            2. ***metrics.json***: the metrics to monitor are defined

        iii. The most outstanding aspects of its configuration are:

            1. Definition of the LPARs to access
            2. Define the use of prometheus and mongodb (in both cases the port on which they listen and user in the case of the DB), with true or false you define its use or not
            3. In the first stage it is recommended to work with prometheus but without mongodb
            4. The PCI variable in the configuration is the MIPS of the physical equipment

    b. ZEBRA start

    c. Prometheus installation (last version)

        i. Download and install prometheus (https://prometheus.io/download/)

        ii. Configure Prometheus to do scrapping of ZEBRA

            - Review the monitoring frequency of the RMF monitor III (frequently 100sec.) since it is the same that must be configured in the Prometheus collection job to avoid queries with unnecessary frequency
            - In the prometheus configuration file (prometheus.yml) define the capture job for ZEBRA

- Configure a high time out to avoid loss of information due to delay (~60s, depending on the previous one)

d. Grafana installation (last version)

1. Download and install grafana (https://grafana.com/grafana/download)

e. Verify access from the midrange side to the mainframe

Execute the curl command against the ip and ports of the monitored LPARs

curl http://ip_lpr1:8803

curl http://ip_lpr2:8803

curl http://ip_lprn:8803

The satisfactory answer must be:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
     "http://www.w3.org/TR/html4/frameset.dtd">
<html lang="en">
<head>
 <meta http-equiv="Content-Type" content="text/html;
     charset=iso-8859-1"/>
 <title>RMF Data Portal</title>
 <link rel="stylesheet" type="text/css" href="/gpm/include/ddsml.css"/>
</head>
<frameset rows="60px,*" border="0" frameborder="0" framespacing="0"
 role="document" aria-label="RMF Data Portal Home Page">
 <frame src="/gpm/include/frame-title.html" name="title"
      title="RMF Data Portal Menu"
      scrolling="no" frameborder="0" marginwidth="0"
      marginheight="0">
 <frame src="/gpm/include/frame-home.html" name="content"
      title="Home" frameborder="0" marginwidth="10" marginheight="0">
 <noframes role="region" aria-label="No frames message">
  <p class="redbold">
   GPM9997I Sorry, your WebBrowser does not support frames.
  </p>
 </noframes>
</frameset>
</html>
```

Networking side/Security

Request access from the server at point 2) [Midrange Side] access to the IPs at point 1).c [Mainframe Side] port 8803.

More information about ZEBRA

a) Zebra github site: https://github.com/zowe/zebra: on the site itself there is very complete information for installing the solution

   a. https://github.com/zowe/zebra/tree/main/Documentation/User%20Documentation
   b. https://github.com/zowe/zebra/tree/main/Documentation/Installation%20guides

b) GPMSRVnn configuration member example

```
/*******************************************************************/
/*                                                                 */
/* NAME:      GPMSRV00                                             */
/*                                                                 */
/* DESCRIPTION: PARMLIB MEMBER FOR THE RMF DISTRIBUTED DATA SERVER */
/*          HOST ADDRESS SPACE (GPMSERVE)                          */
/*                                                                 */
/* COPYRIGHT:   Licensed Materials - Property of IBM               */
/*          5650-ZOS                                               */
/*          Copyright IBM Corp. 1996, 2019                         */
/*          Status=HRM77C0 (z/OS V2R4 RMF)                         */
/*                                                                 */
/* CHANGE ACTIVITY:                                                */
/*   $E0=GPMIPM,HRM6603,,GBO:      Initial version                 */
/*   $G1=TIVDM,HRM6607,,GBO:       Support for Tivoli DM           */
/*   $H1=HTTP,HRM7703,,GBO:        HTTP                            */
/*   $J1=R705,HRM7705,,GBO:        Default is HTTP=OFF             */
/*   $J2=SEC,HRM7705,01OCT2001,GBO:  HTTP authentification         */
/*   $L1=CIM,HRM7720,,GUB:         Support for RMF CIM provider    */
/*   $M1=CIM,HRM7730,,GBO:         HTTP via UNIX domain sockets    */
/*   $21=MISC,HRM7750,,GUB:        RMF LDAP backend withdrawn      */
/*   $81=MISC,HRM77B0,,PMU:        OPTION cleanup          @81A    */
/*   $91=ATTLS,HRM77C0,,KGE:       CLIENT_CERT                     */
/*   $92=ATTLS,OA57325,,MAI:       HTTPS                           */
/*                                                                 */
/*******************************************************************/
/* General parameters:                                             */
/* ===================                                             */
/* - CACHESLOTS specifies the number of CACHE entries (one for each */
/*   MINTIME).                                                     */
/*   Default: CACHESLOTS(4)                                        */
/*                                                                 */
/* - DEBUG_LEVEL specifies the amount of messages that is sent to  */
/*   the SYSPRINT dataset. DEBUG_LEVEL(0) suppresses all           */
/*   informational messages                                        */
/*   Default: DEBUG_LEVEL(3)                                       */
/*                                                                 */
/* - SERVERHOST allows to specify an IP address (or hostname) that */
/*   the server binds to when he opens any listener sockets. This  */
/*   option should only be used, if a host has serveral IP addresses */
/*   (different network adapters) and you want the DDS server to   */
/*   bind its services to one specific IP address.                 */
/*   Make sure, that the value you specify is a valid IP-address   */
```

```
/*   (or hostname) of the host where the DDS server runs.                  */
/*   On z/OS you may use the TSO HOMETEST command to find out the          */
/*   valid IP-addresses.                                                   */
/*   Example: SERVERHOST(9.164.123.244)                                    */
/*   Default: SERVERHOST(*) - ANY IP ADDRESS                               */
/*                                                                         */
/*************************************************************************/
CACHESLOTS(4)              /* Number of timestamps in CACHE                */

DEBUG_LEVEL(0)              /* No  informational messages                  */
SERVERHOST(*)              /* Don't bind to specific IP-Address            */
/*************************************************************************/
/* HTTP section:                                                         */
/* =============                                                         */
/* - MAXSESSIONS_HTTP specifies the number of concurrent HTTP server     */
/*   threads, that are allowed.                                          */
/*   Default: MAXSESSIONS_HTTP(20)                                       */
/*                                                                       */
/* - HTTP_PORT specifies the TCPIP port number for HTTP requests.        */
/*   Default: HTTP_PORT(8803)                                            */
/*                                                                       */
/* - HTTP_ALLOW specifies the host names/IP addresses that can use       */
/*   the HTTP interface. Wildcards * and ? are allowed. More than        */
/*   one HTTP_ALLOW statement may be present.                            */
/*   Examples: HTTP_ALLOW(*.ibm.com)                                     */
/*          HTTP_ALLOW(9.164.*.*)                                        */
/*          HTTP_ALLOW(bosch.boeblingen.de.ibm.com)                      */
/*   Default:  HTTP_ALLOW(*)                                             */
/*                                                                       */
/* - HTTP_NOAUTH specifies the host names/IP addresses that can use      */
/*   the HTTP interface without authentification (userid/password).      */
/*   Wildcards * and ? are allowed. More than one HTTP_NOAUTH            */
/*   may be present.                                                     */
/*   Note: The host running the RMF CIM provider must be specified       */
/*       here.                                                           */
/*   Example: HTTP_NOAUTH(sysa.boeblingen.ibm.com)                       */
/*   Default: HTTP_NOAUTH()                                              */
/*                                                                       */
/* - HTTPS specifies whether secure AT-TLS setup is required for         */
/*   incoming HTTP connections.                                          */
/*   For HTTPS(ATTLS), the DDS verifies that incoming connections        */
/*   are secured by an AT-TLS setup. If a connection is not secured      */
/*   by AT-TLS, then the connection is rejected.                         */
/*   Examples: HTTPS(ATTLS)                                              */
/*          HTTPS(NO)                                                    */
/*   Default:  HTTPS(ATTLS)                                              */
/*                                                                       */
/* - CLIENT_CERT specifies whether the DDS accepts an user ID taken      */
/*   from an AT-TLS provided client certificate on incoming HTTP         */
/*   connections. This user ID is also used to check the authori-        */
/*   zation against the ERBSDS.MON3DATA facility class profile           */
/*   instead of an user ID and password authentication.                  */
/*   For CLIENT_CERT(NONE), the DDS does not use a client                */
```

```
/*   certificate for authentication of a requester.                  */
/*   Example: CLIENT_CERT(ACCEPT)                                    */
/*   Default: CLIENT_CERT(NONE)                                      */
/*******************************************************************/
MAXSESSIONS_HTTP(20)                /* MaxNo of concurrent HTTP requests */
HTTP_PORT(8803)                     /* Port number for HTTP requests     */
HTTP_ALLOW(nnn.nnn.nnn.nnn)         /* Mask for hosts that are allowed   */
HTTP_NOAUTH()                       /* No server can access without auth.*/
HTTPS(ATTLS)                        /* AT-TLS setup required             */
CLIENT_CERT(NONE)                   /* No client certificate is used     */
```

HTTP_ALLOW(nnn.nnn.nnn.nnn) ← IP del servidor Linux al que se le dió acceso