PeopleSoft Administrator

TIPS

About:

Author	Ganesh Krishnamurthy	Ganesh.k1705@infosys.com
Scripts Author	Karthik R	Karthik_r01@infosys.com
Script Contributor	Anup Madhava Kumar	Anup_kumar05@infosys.com
Version	1.0	

Contents

PeopleSoft PIA

Objective:

In a System Administrator's day to day operations, there would be a need to automate some if not all of their day to day activities; Such as Bouncing of Application Servers, Web Servers, Scheduler servers and monitoring the health of these servers etc. etc., hitherto would have been done manually, which would become very mundane. The purpose/objective of this document is to provide a mechanism with which SA's can automate their daily mundane tasks. The area that is covered is related to PeopleSoft.

Audience:

PeopleSoft System Administrators

Topics Covered:

Topics covered are as follows:

1. Automate Start/Stop Application/Web/PRCS servers script – Pseudocode
2. Automate Start/Stop Elastic Search Server script – Pseudocode
3. Automate Monitoring Health of PIA Servers

Automate Start/Stop Application/Web/PRCS servers script

The following are the pre-reqs for running this script:

A. The script should be run as PeopleSoft owner... Such as *psoft* etc. Which means that this user should be the one who has installed PeopleSoft and manages PIA.
B. The script is written for Linux/Unix OS. However, this can be modified/adopted for Windows OS with very few changes.
C. This script assumes PeopleSoft PIA is setup on a Linux VM with two PeopleSoft App/Web and one PRCS domain(s).
D. The Domain names, Port numbers, ip addresses, Folder Path, Folder Structures used in the script are for sample/demonstration Purposes only. They will vary as per your installation requirements.
E. This script assumes the following:
    1. PS_HOME is set to : /usr/psft/product/pt/ps_home8.57.11/appserv
    2. PS_CFG_HOME is set to : /usr/psoft/pscfg
    3. Scripts_Home is set to : /usr/psft/scripts

StartPIA.sh Pseudocode: Please Refer to Appendix – A for details

StopPIA.sh Pseudocode: Please Refer to Appendix – B for details

Automate Start/Stop Elastic Search Server script

The following are the pre-reqs for running this script:

A. The script should be run as PeopleSoft owner... Such as *psoft* etc. Which means that this user should be the one who has installed PeopleSoft and manages PIA.
B. The script is for Linux/Unix OS. However, this can be modified/adopted for Windows OS with very few changes.
C. This script assumes PeopleSoft Elastic Search is setup on a Linux VM with one Search Instance Domain.
D. The Domain names, Port numbers, ip addresses, Folder Path, Folder Structures used in the script are for sample/demonstration Purposes only. They will vary as per your installation requirements.
E. This script assumes the following:
   4. Elastic_Home is set to : /usr/psoft/ pt/es6.1.2/pt/elasticsearch6.1.2/
   5. Service is set to : elastic
   6. Scripts_Home is set to : /usr/psoft/scripts

Elastic_Start_Stop.sh Pseudocode: Please Refer to Appendix – C for details

Automate Monitoring Health of PIA Servers

The objective of monitoring the health of PIA servers is one of the primary responsibilities of a PS Admin. There are two parts to monitoring.

a. The Monitor Script itself and
b. Scheduling the monitor script using : 'Crontab'.

Monitoring Script:

The following are the pre-reqs for this script:

A. The script should be run as PeopleSoft owner... Such as *psoft* etc. Which means that this user should be the one who has installed PeopleSoft and manages PIA.
B. The script is for Linux/Unix OS. However, this can be modified/adopted for Windows OS with very few changes.
C. This script assumes that the monitoring script is setup on a Linux VM with two App/Web domains each and one PRCS Domain.
D. If you have PeopleSoft PIA installed on multiple VMs, then, this script needs to be copied to those VMs.
E. A list of comma separated Email IDs (can be stored in a .txt file) is available for this script so that PIA health status is sent to those users who in turn can take action. If you don't want to maintain a file of email ids then please hardcode the values in the script under: 'to' variable.
F. The Domain names, Port numbers, ip addresses, Folder Path, Folder Structures used in the script are for sample/demonstration Purposes only. They will vary as per your installation requirements.
G. Please change user message verbiage to suit your needs where appropriate
H. The IP address for SMTP server in the script shows XX.XXX.X.XXX:25. Please replace 'X' with your installations SMTP server ip address.
I. This script assumes the following:
   1. monitor_Home is set to : /usr/psoft/monitor
   2. Scripts_Home is set to : /usr/psoft/scripts
   3. PS_HOME is set to : /usr/psoft/product/pt/ps_home8.57.11/appserv

MonitorPIA.sh Pseudocode: Please Refer to Appendix – D for details

Scheduling the monitor script using: ‘Crontab’:

The scheduling of the monitoring script in Linux/Unix OS is setup via tool called ‘Crontab’. Let us look at some examples:

1. Schedule monitorPIA.sh to run at 15-minute intervals. To do this:
    a. Login into the Linux VM where PeopleSoft as well as the scripts are setup/configured.
    b. At the prompt execute the following command:

        crontab –e → e means edit

        this will open an editor.

        Enter the following:

        ```
        */15 * * * * /usr/psoft/monitor/monitorPIA.sh > /tmp/monitorPIA.log
        ```

        Save it.

    c. To check if this entry has been setup correctly, give this command:
        i. Crontab –l → l means List
        ii. This will display the contents of crontab.

        ```
        */15 * * * * /usr/psoft/monitor/monitorPIA.sh > /tmp/monitorPIA.log
        ```

    d. The PIA servers will now be monitored every 15 minutes and if any of the servers are down, it will email those users who are either listed in email_id.txt or in the ‘To=’ of the script.

2. To bring up all PeopleSoft PIA servers whenever the VM is booted.
    a. Login into the Linux VM where PeopleSoft as well as the scripts are setup/configured.
    b. At the prompt execute the following command:

        crontab –e → e means edit

        this will open an editor.

        Enter the following:

        ```
        @reboot  /usr/psoft/scripts/StartPIA.sh > /tmp/StartPIA.log
        ```

        Save it.

    c. To check if this entry has been setup correctly, give this command:
        i. Crontab –l → l means List
        ii. This will display the contents of crontab.

        ```
        */15 * * * * /usr/psoft/monitor/monitorPIA.sh > /tmp/monitorPIA.log

        @reboot  /usr/psoft/scripts/StartPIA.sh > /tmp/StartPIA.log
        ```

    d. The PIA servers will now be started automatically whenever the VM is booted

Appendix –A

StartPIA.sh Pseudocode:

```
#!/bin/bash
host="`hostname -f`"
hostip="`hostname -i`"
PS_HOME="/usr/psoft/product/pt/ps_home8.57.11/appserv"
scripts_home="/usr/psft/scripts "
prcs_domain=PRCSDOM1
app_domain1=APPDOM1
app_domain2=APPDOM2
web_domain1=WEBDOM1
web_domain2=WEBDOM2
echo "Execute .profile"   > $scripts_home/logs/startdomains.log
. ./.profile >> $scripts_home/logs/startdomains.log
echo "PS Home: " $PS_HOME  >> $scripts_home/logs/startdomains.log
cd $PS_HOME/appserv
echo "Starting all Domains" >> $scripts_home/logs/startdomains.log
psadmin start -d *all 2>/dev/null >> $scripts_home/logs/startdomains.log
echo "Finished Starting Domains" >> $scripts_home/logs/startdomains.log
echo "Start Checking Status" >> $scripts_home/logs/startdomains.log
psadmin -c status -d $app_domain1 >>  $scripts_home/logs/startdomains.log
psadmin -c status -d $app_domain2 >>  $scripts_home/logs/startdomains.log
psadmin -p status -d $prcs_domain >>  $scripts_home/logs/startdomains.log
echo " "  >>  $scripts_home/logs/startdomains.log
echo "Web Status" >>  $scripts_home/logs/startdomains.log
psadmin -w status -d $web_domain1 >>  $scripts_home/logs/startdomains.log
psadmin -w status -d $web_domain2 >>  /$scripts_home/logs/startdomains.log
echo "Finished Checking status" >> $scripts_home/logs/startdomains.log
date >>  $scripts_home/logs/startdomains.log
```

Appendix –B

StopPIA.sh Pseudocode:

```
#!/bin/bash

host="`hostname -f`"

hostip="`hostname -i`"

PS_HOME="/usr/psoft/product/pt/ps_home8.57.11/appserv"

scripts_home="/usr/psft/scripts "

prcs_domain=PRCSDOM1

app_domain1=APPDOM1

app_domain2=APPDOM2

web_domain1=WEBDOM1

web_domain2=WEBDOM2

echo "Execute .profile"   > $scripts_home/logs/stopdomains.log

. ./.profile >> $scripts_home/logs/stopdomains.log

echo "PS Home: " $PS_HOME  >> $scripts_home/logs/stopdomains.log

cd $PS_HOME/appserv

echo "Stopping App servers" >> $scripts_home/logs/stopdomains.log

psadmin -c shutdown -d $app_domain1  >>  /$scripts_home/logs/stopdomains.log

psadmin -c shutdown -d $app_domain2  >>  /$scripts_home/logs/stopdomains.log

echo "Stopping PRCS servers" >> $scripts_home/logs/stopdomains.log

psadmin -p stop -d $prcs_domain >>  $scripts_home/logs/stopdomains.log

echo "Stopping Web servers" >> $scripts_home/logs/stopdomains.log

psadmin -w shutdown -d $web_domain1 >>  $scripts_home/logs/stopdomains.log

psadmin -w shutdown -d $web_domain2 >>  $scripts_home/logs/stopdomains.log

echo "Finished Stopping Domains" >> $scripts_home/logs/stopdomains.log

echo "Start Checking Status" >> $scripts_home/logs/stopdomains.log

psadmin -c sstatus -d $app_domain1 >>  /$scripts_home/logs/stopdomains.log

psadmin -c sstatus -d $app_domain2 >>  $scripts_home/logs/stopdomains.log

echo " "  >>  $scripts_home/logs/stopdomains.log

echo "Web Status" >>  $scripts_home/logs/stopdomains.log

psadmin -w status -d $web_domain1 >>  $scripts_home/logs/stopdomains.log

psadmin -w status -d $web_domain2 >>  $scripts_home/logs/stopdomains.log
```

```
echo "Finished Checking status" >> $scripts_home/logs/stopdomains.log
date >>  $scripts_home/logs/stopdomains.log
```

Appendix – C

Elastic_Start_Stop.sh Pseudocode:

```
#!/bin/bash
service=elastic
scripts_home="/usr/psoft/scripts"
##echo "current dir is 0 - $(pwd)" > $scripts_home/logs/startelastic.log
echo "Execute .bash_profile"   > $scripts_home/logs/startelastic.log
. ./.bash_profile >> $scripts_home/logs/startelastic.log
if (( $(ps -ef | grep -v grep | grep $service | wc -l) > 0 ))
then
##echo "current dir is 1 - $(pwd)" >> $scripts_home/logs/startelastic.log
echo "$service is running!!!" >> $scripts_home/logs/startelastic.log
echo " value of running PID is $(ps -ef | grep -v grep | grep $service | awk '{print $2}')" >> $scripts_home/logs/startelastic.log
ps -ef | grep -v grep | grep $service | awk '{print $2}' | xargs kill -9 >> $scripts_home/logs/startelastic.log
##echo "current dir is 2 - $(pwd)" >> $scripts_home/logs/startelastic.log
cd /usr/psoft/pt/es6.1.2/pt/elasticsearch6.1.2/bin
##echo "current dir is 3 - $(pwd)" >> $scripts_home/logs/startelastic.log
nohup sh ./elasticsearch & >> $scripts_home/logs/startelastic.log
##echo "current dir is 3a - $(pwd)" >> $scripts_home/logs/startelastic.log
else
echo "$service is down!!"
##echo "current dir is 4 - $(pwd)" >> $scripts_home/logs/startelastic.log
cd / usr/psoft/pt/es6.1.2/pt/elasticsearch6.1.2/bin
##echo "current dir is 5 - $(pwd)" >> $scripts_home/logs/startelastic.log
nohup sh ./elasticsearch & >> $scripts_home/logs/startelastic.log
fi
```

Appendix – D

MonitorPIA.sh Pseudocode:

```
#!/bin/bash
host="`hostname -f`"
hostip="`hostname -i`"
monitor_home="/usr/psoft/monitor"
from="PeopleSoft_Monitor@JohnDoe.com"
to="`cat $monitor_home/mail_ids.txt`"
#to="Joh.Doe@Johndoe.com"
prcs_domain=PRCSDOM1
app_domain1=APPDOM1
app_domain2=APPDOM2
web_domain1=WEBDOM1
web1_http_port=8000
web1_https_port=8100
web_domain2=WEBDOM2
web2_http_port=8200
web2_https_port=8300
echo "Execute .profile"   > $monitor_home/tmp/psdomainstatus.log
. ./.profile >> $monitor_home/tmp/psdomainstatus.log
echo "PS Home: " $PS_HOME  >> $monitor_home/tmp/psdomainstatus.log
cd $PS_HOME/appserv
psadmin -p sstatus -d $prcs_domain >  $monitor_home/tmp/prcsdomainstatus.log
psadmin -c sstatus -d $app_domain1 >  $monitor_home/tmp/app1domainstatus.log
psadmin -c sstatus -d $app_domain2 >  $monitor_home/tmp/app2domainstatus.log

psadmin -w status -d $web_domain1 >  $monitor_home/tmp/web1domainstatus.log
psadmin -w status -d $web_domain2 >  $monitor_home/tmp/web2domainstatus.log

echo "Datetime = `date`" >>  $monitor_home/tmp/prcsdomainstatus.log
echo "Datetime = `date`" >>  $monitor_home/tmp/app1domainstatus.log
echo "Datetime = `date`" >>  $monitor_home/tmp/app2domainstatus.log


echo "Datetime = `date`" >>  $monitor_home/tmp/web1domainstatus.log

echo "Datetime = `date`" >>  $monitor_home/tmp/web2domainstatus.log


#Check PRCS server

if [ -n "$prcs_domain" ]; then

    echo "Checking PRCS domain" >> $monitor_home/tmp/psdomainstatus.log

    PRCS=$(cat $monitor_home/tmp/prcsdomainstatus.log | grep -c "PSPRCSRV")

    MONITSRV=$(cat $monitor_home/tmp/prcsdomainstatus.log |grep -c "MONITORSRV")

    if [ $PRCS -eq 0 ] || [ $MONITSRV -eq 0 ]; then

        echo -e "Hi,\n\nPlease check below Process Scheduler Server Status on $host -
$hostip.\n\n==========================================\n`cat
$monitor_home/tmp/prcsdomainstatus.log`\n\nThanks,\nInfra Team."| mailx -v -r "$from" -s "Process
Scheduler Server on $host - $hostip : DOWN" -S smtp="XX.XXX.X.XXX:25" "$to" >>
$monitor_home/tmp/psdomainstatus.log

        echo "`date` - Process Scheduler Server : DOWN " >> $monitor_home/prcs_status.log

    else

        echo "`date` - Process Scheduler Server : UP and Running " >> $monitor_home/prcs_status.log

        echo " Process Scheduler Server: UP and RUNNING " >> $monitor_home/tmp/psdomainstatus.log

    fi

fi

#Check App server1

if [ -n "$app_domain1" ]; then

    jsl_process=$(cat $monitor_home/tmp/app1domainstatus.log | grep -c "JSL")

    psappsrv_process=$(cat $monitor_home/tmp/app1domainstatus.log | grep -c "PSAPPSRV")

    if [ $jsl_process -eq 0 ] || [ $psappsrv_process -eq 0 ];then

        echo -e "Hi,\n\nPlease check below App Server Domain ($app_domain1) Status on $host -
$hostip.\n\n==========================================\n`cat
$monitor_home/tmp/app1domainstatus.log`\n\nThanks,\nInfra Team."| mailx -v -r "$from" -s "App Server
$app_domain1 on $host - $hostip : DOWN" -S smtp="XX.XXX.X.XXX:25" "$to" >>
$monitor_home/tmp/psdomainstatus.log

        echo "`date` - App Server $app_domain1: DOWN " >> $monitor_home/app_status.log

    else

        echo "`date` - App Server $app_domain1: UP and Running " >> $monitor_home/app_status.log

        echo " App Server $app_domain1: UP and RUNNING " >> $monitor_home/tmp/psdomainstatus.log

    fi
```

```
fi

#Check App Server2

if [ -n "$app_domain2" ]; then

        echo "Checking App2 domain" >> $monitor_home/tmp/psdomainstatus.log

        jsl_process2=$(cat $monitor_home/tmp/app2domainstatus.log | grep -c "JSL")

        psappsrv_process2=$(cat $monitor_home/tmp/app2domainstatus.log | grep -c "PSAPPSRV")

        if [ $jsl_process2 -eq 0 ] || [ $psappsrv_process2 -eq 0 ];then

                echo -e "Hi,\n\nPlease check below App Server Domain ($app_domain2) Status on $host - $hostip.\n\n==========================================\n`cat $monitor_home/tmp/app2domainstatus.log`\n\nThanks,\nInfra Team."| mailx -v -r "$from" -s "App Server $app_domain2 on $host - $hostip : DOWN" -S smtp=" XX.XXX.X.XXX:25" "$to" >> $monitor_home/tmp/psdomainstatus.log

                echo "`date` - App Server $app_domain2: DOWN " >> $monitor_home/app_status.log

        else

                echo "`date` - App Server $app_domain2: UP and Running " >> $monitor_home/app_status.log

                echo " App Server $app_domain2: UP and RUNNING " >> $monitor_home/tmp/psdomainstatus.log

        fi

fi



#Check Web server1 Process

if [ -n "$web_domain1" ]; then

        /sbin/lsof -i:$web1_http_port > $monitor_home/tmp/webserver1_http.log

        count_web1_http=`cat $monitor_home/tmp/webserver1_http.log |grep java |wc -l`

        /sbin/lsof -i:$web1_https_port > $monitor_home/tmp/webserver1_https.log

        count_web1_https=`cat $monitor_home/tmp/webserver1_https.log |grep java |wc -l`

        if [ $count_web1_http -eq 0 ] || [ $count_web1_https -eq 0 ];then

                echo -e "Hi,\n\nPlease check Web Server Domain status of -- ($web_domain1) -- on $host - $hostip.\n\n========WebDomainStatus==================================\n`cat $monitor_home/tmp/web1domainstatus.log` \n\n========http_PortStatus==================================\n`cat $monitor_home/tmp/webserver1_http.log` \n\n========httpsPortStatus==================================\n`cat $monitor_home/tmp/webserver1_https.log`\n\nThanks,\nInfra Team."| mailx -v -r "$from" -s "Web Server $web_domain1 on $host - $hostip : DOWN" -S smtp=" XX.XXX.X.XXX:25" "$to" >> $monitor_home/tmp/psdomainstatus.log

                echo "`date` - Web Server $web_domain1: DOWN " >> $monitor_home/web_status.log

        else

                echo "`date` - Web Server $web_domain1: UP and Running " >> $monitor_home/web_status.log

                echo " Web Server $web_domain1: UP and RUNNING " >> $monitor_home/tmp/psdomainstatus.log
```

```
    fi
fi
#Check Web server2 Process
if [ -n "$web_domain2" ]; then
    echo "Checking Web2 domain" >> $monitor_home/tmp/psdomainstatus.log
    /sbin/lsof -i:$web2_http_port > $monitor_home/tmp/webserver2_http.log
    count_web2_http=`cat $monitor_home/tmp/webserver2_http.log |grep java |wc -l`
    /sbin/lsof -i:$web2_https_port > $monitor_home/tmp/webserver2_https.log
    count_web2_https=`cat $monitor_home/tmp/webserver2_https.log |grep java |wc -l`
    if [ $count_web2_http -eq 0 ] || [ $count_web2_https -eq 0 ];then
        echo -e "Hi,\n\nPlease check Web Server Domain status of -- ($web_domain2) -- on $host -
$hostip.\n\n========WebDomainStatus==================================\n`cat
$monitor_home/tmp/web2domainstatus.log`
\n\n========http_PortStatus===================================\n`cat
$monitor_home/tmp/webserver2_http.log`
\n\n========httpsPortStatus==================================\n`cat
$monitor_home/tmp/webserver2_https.log`\n\nThanks,\nInfra Team."| mailx -v -r "$from" -s "Web Server
$web_domain2 on $host - $hostip : DOWN" -S smtp=" XX.XXX.X.XXX:25" "$to" >>
$monitor_home/tmp/psdomainstatus.log
        echo "`date` - Web Server $web_domain2: DOWN " >> $monitor_home/web_status.log
    else
        echo "`date` - Web Server $web_domain2: UP and Running " >> $monitor_home/web_status.log
        echo " Web Server $web_domain2: UP and RUNNING " >>
$monitor_home/tmp/psdomainstatus.log
    fi
fi
exit 0
```