General Upgrade Procedures

HPE AutoPass License Server (APLS) upgrade procedure from v9.6.x to v9.14.x is briefed in this document.

Revised: April 2023

Prerequisites

You are strongly recommended to take the system backup before starting the AutoPass License Server upgrade process.

- Take the backup of the data folder specified during installation.
  - Windows Default: ***C:\Program Data\HP\HP AutoPass License Server\AutoPass***
  - UNIX Default: ***/var/opt/HP/HP AutoPass License Server***
- The location of data folder should remain the same during upgrade to ensure data retention is accomplished.
- Refer to the License Server User Guide for specific Prerequisites in Chapter 2: 'Before You Install' section.
- Identify APLS communication ports, defaults 5814, 9001 and 7900. (Appendix F)

Upgrade Notes

- When upgrading from version 9.6.x to 9.7.x or later version, it is expected that a new Lock Code will be created during installation, causing previously installed license to become invalid. In such case, request a license re-host by contacting HPE Licensing Support.
- After completing the upgrade process if the license server configuration values are lost (i.e., the data folder location changed/modified and new data folder was created) follow steps on Data Recovery after upgrade.

Single Host Upgrade

A previous version of APLS (9.6.x to 9.12.x) must be already installed on the system to be qualified for upgrade. If the target system has an earlier version of the product, this one is replaced by the new version installed.

Follow the step below to upgrade APLS versions 9.6.x to latest

1. Execute the ***setup*** script or executable and follow the instructions on the screen.
    a. APLS Service will start upon installation completion.
2. Validate APLS configuration values.

Redundancy Mode Upgrade

These steps were tested under ideal conditions and do not consider major network faults (or external factors) that may affect communication between product client and APLS servers during the upgrade process.

Prerequisites

- APLS Primary and Secondary serves should be configured for redundancy mode.
- Product should support APLS Redundancy mode configuration.
- License Server Redundancy status should be active on both Primary and Secondary Servers. (Appendix E)
- Secondary Server should be successfully handling request while Primary server is down.

Please follow the steps below to upgrade from 9.7.x to 9.14.x:

1. **Primary Server**: Stop APLS service.
2. **Primary Server**: Install Latest APLS. (Single Host Upgrade)
3. **Primary Server**: Login to APLS UI to Validate Licenses and Configurations. (Upgrade Notes)
4. **Secondary Server**: Stop APLS service.
5. **Secondary Server**: Install Latest APLS. (Single Host Upgrade)
6. **Secondary Server**: Login to APLS UI to Validate Licenses and Configurations. (Upgrade Notes)

High Availability Upgrade for 9.6.x

Upgrading from version 9.6.x to 9.7.x or later version, will cause a new Lock Code to be created during installation, causing previously installed license to become invalid. Additional steps are necessary to prevent APLS service interruption while re-hosting required licenses. These steps were tested under ideal conditions and do not consider major network faults (or external factors) that may affect communication between product client and APLS servers during the upgrade process.

Prerequisites

- APLS Primary and Secondary serves should be configured for redundancy mode.
- Product should support APLS Redundancy mode configuration.
- Elevated User Permissions are required to modify Server Internal configurations.
- License Server Redundancy status should be active on both Primary and Secondary Servers. (Appendix E)
- Secondary Server should be successfully handling request while Primary server is down.

Please follow the steps below to upgrade from 9.6.x to 9.14.x:

1. **Primary Server:** Stop APLS service.
2. **Primary Server**: Block incoming communication into Primary APLS Server**.** (Appendix A/C and F)
    a. Validate that communication is blocked from Product Client Host to Primary APLS Server. (Appendix B/D)
    b. If communication is blocked successfully:
        1. Synchronization requests between servers should be blocked while upgrading.
        2. Product Client Requests are redirected to the Secondary APLS Server.
3. **Primary Server**: Install Latest APLS. (Single Host Upgrade)
4. **Primary Server**: Login to APLS UI.
    a. Re-Host Licenses with the new Lock Code. (Upgrade Notes)
5. **Primary Server**: Install re-hosted licenses.
6. **Primary Server**: Validate Licenses and Configurations.
7. **Primary Server**: Restore incoming communication to Primary APLS Server**.** (Appendix A/C and F)
    a. Revert changes executed at step 2.
    b. Validate that communication is allowed from Product Client Host to Primary APLS Server. (Appendix B/D)
        1. Client Requests should now be handled by the Primary APLS Server.
8. **Secondary Server:** Stop APLS Service.
9. **Secondary Server**: Install Latest APLS. (Single Host Upgrade)
10. **Primary Server:** Login to APLS UI.
    a. Validate Licenses and Configurations.
    b. Validate Redundancy status for both Primary and Secondary Servers. (Appendix E)

Data Recovery after Upgrade

If the license server upgrade requires a new data folder location and data retention is required, follow the steps below:

1. Stop License Server service.
2. Replace all folders and files from the backed up data folder to the current data folder location.

   Except for the following file:

   a. <APLS Data Folder Path>\data\conf\dbconfig.xml.
3. Start the License Server service.

The database retention will preserve configuration from the previous installed version, for example:

- The password is not reset
- The old license information (borrowed, and pool) is preserved along with new graph for In Use.
- The Main Configuration and User Management Page values.
- Registered Product PD Files.

Appendix:

The purpose of the following section is to illustrate common examples for blocking and validating communication between servers, there are additional ways not covered in this document. Commands below may impact other running process and must be consulted with the appropriate infrastructure support team.

A. Blocking Communication between Linux Servers:

1. Using iptables:
    a. Validating Current Rules.
        i. `iptables -S`
    b. Blocking Incoming Connection port; for instance, for port 5814.
        i. `iptables -A INPUT -p tcp --destination-port 5814 -j DROP`
    c. Blocking Outgoing Connection port; for instance, for port 5814.
        i. `iptables -A OUTPUT -p tcp --dport 5814 -j DROP`
    d. Blocking Incoming Connection IP; for instance, for IP 1.2.3.4
        i. `iptables -A INPUT -s 1.2.3.4 -j DROP`
    e. Blocking Outgoing Connection IP; for instance, for IP 1.2.3.4
        i. `iptables -A OUTPUT -s 1.2.3.4 -j DROP`
2. Using firewall-cmd:
    a. Verify allowed/rejected port rules access, for instance all and public zones:
        i. `firewall-cmd --list-all-zones`
        ii. `firewall-cmd --zone=public --list-all`
    b. Add/Remove access to application port; for 5814 port access:
        i. `firewall-cmd --zone=public --add-port=5814/tcp --permanent`
        ii. `firewall-cmd --zone=public --remove-port=5814/tcp --permanent`
        iii. `firewall-cmd –reload`
3. If the server has direct access, for instance physical or a VM console manager, network interface can be temporary disabled.
    a. Example using ifconfig to disable Ethernet card, eth0:
        i. `ifconfig eth0 down`
4. Temporary assign a different IP Address to the machine that is being upgraded.
    a. Example using ifconfig:
        i. `ifconfig eth0 192.168.0.1 netmask 255.255.255.0`

B. Validating Communication between Linux Servers:

1. Using Nmap:

a. Validate if TCP server port is Open/Closed, for instance 1.2.3.4:5814

i. `nc -zv 1.2.3.4 5814`

a. Expected output for Closed: No route to host

b. Expected output for Open: Connected to 1.2.3.4:5814

2. Using Telnet:

a. Verify if a server TCP port is open; for instance, for 1.2.3.4:5814

i. `telnet 1.2.3.4 5814`

C. Blocking Communication between Windows Servers

a. Using Windows Firewall:

i. Open Windows Firewall Advanced Settings

ii. Right-Click the type of rule Inbound/Outbound and select New Rule…

iii. On the Rule Type page, choose Custom.

iv. On Program, choose "All programs."

v. On Protocol and Ports, provide the required values

vi. On Scope, select "These IP addresses" and provide required values

vii. On Action, choose "Block the connection."

viii. On Profile, leave the defaults of everything checked.

ix. Finally, on Name, give the rule a name and optionally a description.

b. Using the hosts file to block outbound connections

i. Locate the hosts file

1. Usually at C:\WINDOWS\system32\drivers\etc

ii. Redirect the host that is required to be blocked to local host, for example

1. `127.0.0.1 www.exampledomain.com`

D. Validating Communication between Windows Servers:

a. Using Telnet:

i. Verify if a server TCP port is open; for instance, for 1.2.3.4:5814

1. `telnet 1.2.3.4 5814`

b. Using Test-NetConnection in PowerShell

i. Example Testing connection to IP 1.2.3.4 through Port 8080

1. `Test-NetConnection 1.2.3.4 -Port 8080`

E. Inspecting License Server Redundancy Status

a. Login to APLS UI

b. Go to “License Usage” tab

i. Both Servers Up and Running in redundancy mode:

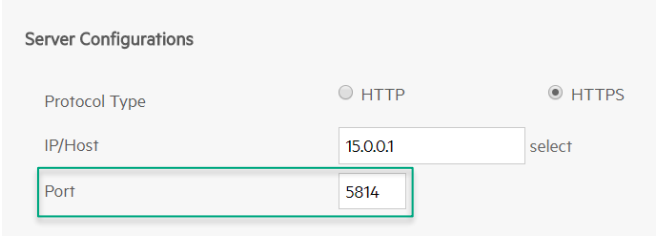
ii. Primary Server Up and Secondary Server Down

F. Identifying License Server communication port.

a. Login to APLS UI

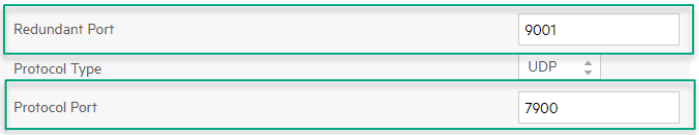
b. From the Configuration tab, go to “**Main**” to open the Main Configuration section.

c. Scroll to the “Server Configurations” section and identify the port



Note: APLS default communication port is 5814.

d. From the Configuration tab, go to “**Redundancy**” to open the Redundancy Configuration section and identify the ports



Note: APLS default redundancy ports are 9001 and 7900.