Package Development in APS for PA (Advanced)

Lab Guide

1.0.0

ISBN: N/A
Parallels
500 SW 39th Street
Suite 200
Renton, WA 98057
USA
Tel: +1 (703) 815 5670
Fax: +1 (703) 815 5675

Contents

Preface

Documentation Conventions

Before you start using this guide, please familiarize yourself with its documentation conventions.

General Conventions

Be aware of the following conventions used in this book.

- The content of this guide is divided into modules, which, in turn, are subdivided into sub modules.
- When following steps or using examples, be sure to type double-quotes ("), left single-quotes (`), and right single-quotes (') exactly as shown.
- The key referred to as RETURN is labeled ENTER on some keyboards.
- Commands in directories included in a PATH variable are used without absolute path names.

```
# $PRODUCT_ROOT_D/bin/<utility name> [parameters] [options]
```

- Steps that use commands in other, less common, directories show the absolute paths in the examples.

```
# /usr/local/pem/bin/<utility name> [parameters] [options]
```

Typographical Conventions

The following formatting conventions in the text identify special information.

Formatting convention	Type of Information	Example
Special Bold	Items you must select, such as menu options, command buttons, or items in a list.	Navigate to the QoS tab.
	Titles of modules, sections, and subsections.	Read the Basic Administration module.
Italics	Used to emphasize the importance of a point, to introduce a term or to designate a command-line placeholder, which is to be replaced with a real name or value.	These are the so-called *shared containers*. To stop the service, type `/etc/init.d/`*<service script>*
`Monospace`	The names of commands, files, and directories.	Use `less /var/log/messages` to investigate startup issues.
`Preformatted`	On-screen computer output in your command-line sessions; source code in XML, C++, or other programming languages.	`Saved parameters for Container 101`
`Preformatted Bold`	What you type, contrasted with on-screen computer output.	**`# rpm -qa | grep -i pba`**
CAPITALS	Names of keyboard keys.	SHIFT, CTRL, ALT
KEY+KEY	Key combinations - user must press and hold down one key and then press another.	CTRL+P, ALT+F4

Shell Prompts in Command Examples

Command-line examples throughout this guide presume that you are using the Bourne-again shell (bash). Whenever a command can be run as a regular user, we will display it with a dollar sign prompt. When a command is meant to be run as root, we will display it with a hash mark prompt:

Bourne-again shell prompt	`$`
Bourne-again shell root prompt	`#`

Lab Directions

To perform all the exercises, you will be working in the isolated demo-system.

Connecting to Sandbox

To connect to your Sandbox, you need to install and use a VNC client (e.g., TightVNC) on your local workstation. To access the Sandbox, use the directions below:

1 Run the VNC-client software on your local workstation

2 Use the following connection parameters:

a Jumper IP-address: provided by Instructor

NOTE: If you do not know the jumper's IP-address, ask the Trainer to provide.

b VNC-server port: 5901

Ask the instructor to provide you a password.

An example connection string would look like this one: `10.111.122.121:5901`

NOTE: You may use your browser with the java plug-in to connect to the VNC-server.
NOTE: To install the TightVNC client visit `http://www.tightvnc.com/download.php`.

Working with Sandbox

During this hands-on practice you will be working with the isolated environment (Sandbox) that includes the following virtual environments:

Hostname	Operating System	FrontNet	BackNet	Description
pba.edu.trn	CentOS 6	10.111.11.11	10.111.22.11	PBA Application Server
pbadb.edu.trn	CentOS 6		10.111.22.12	PBA Database Server
store.edu.trn	CentOS 6	10.111.11.13	10.111.22.13	PBA Online Store
poamn.edu.trn	CentOS 6		10.111.22.14	POA Management Node
poauibr.edu.trn	CloudLinux 6	10.111.11.15	10.111.22.15	POA UI server + Branding
poadns.edu.trn	CentOS 6	10.111.11.16	10.111.22.16	POA DNS Server
web01.edu.trn	CloudLinux 6	10.111.11.17	10.111.22.17	POA Linux Shared Hosting NG Server
jumper.edu.trn, ns1.edu.trn	CentOS 6	10.111.11.1	10.111.22.1	Entry point to above nodes

The *jumper.edu.trn* VE also has a special interface providing access to the Internet for all other VEs within a Sandbox. You will be using this VE to:

- Access other VEs through *ssh* connection using command line interface (CLI).
- Access the PA control panel through the browser (e.g., Chrome).

Accessing CP

To access the POA control panel through the browser:

1 On the *jumper.edu.trn* VE, open the browser.
2 To open the POA control panel through the brand access point, use the following URL: `https://cp.edu.trn`
3 To log into the control panel, use the following credentials:
   - Login: *admin*
   - Password: *setup*

Accessing VE

To access a particular server through `ssh`:

1 Open the command line environment (CLE) and type the following command: `ssh root@<Server_IP_Address>`, e.g., to connect to the POA management node, type:

```
$ ssh root@10.111.22.14
```

2 If prompted, accept the key.
3 Use the following credentials to get authorized:
   - Login: *root*
   - Password: *sw1q2w3e*

NOTE: You can use aliases, e.g.: `poamn`, `poadns`, `web01`, etc.

Lab Prerequisites

The host, on which the APS packages will be build (the *jumper* host) requires the following RPM packages to be installed:

- `aps-php-runtime`
- `apstools`
- `php`
- `php-xml`

Performing APS development and provisioning requires the *APS PHP Runtime* and *APS Tools* packages. Download and install the required packages if tasked by Instructor.

1 On the jumper VE, download the latest versions (to date) from the following link: `http://doc.apsstandard.org/tools/`
   - `apstools-x.x-xxx.noarch.rpm`
   - `aps-php-runtime-x.x-xxx.noarch.rpm`

2 Install them:

```
$ sudo rpm -Uvh package_name.rpm
```

3 By using yum, install the `php` and `php-xml` packages:

```
$ sudo yum install php
$ sudo yum install php-xml
```

Ensure the packages are installed, for example, for the `aps-php-runtime` package:

```
$ yum list | grep aps-php-runtime
aps-php-runtime.noarch      2.0-247     installed
```

The following set of lab exercises applies to POA 5.5 deployed in the training environment.

The system infrastructure should include the following servers:

- POA management node with the name *poamn.edu.trm* (BackNet 10.111.22.14)
- POA UI/Branding NG node with the name *poauibr.edu.trn* (BackNet 10.111.22.15)
- DNS server with the name *poadns.edu.trn* (BackNet 10.111.22.16)
- Apache Webserver NG with the name *web01.edu.trn* (BackNet 10.111.22.17)

The *poadns.edu.trn* server (BackNet 10.111.22.16) will serve as the end-point host for APS provisioning, for what it requires the following RPM packages to be pre-installed:

- `aps-php-runtime`
- `php`
- `php-xml`

Be sure to enable the following options in the POA System Properties available in the provider control panel at System > Settings > System Properties:

- APS development mode
- Customers management from POA UI
- Resellers management from POA UI

Also be sure to set the Password Quality Level for Child Accounts to *None* in the System > Settings > Security > Setup screen.

In POA, make sure that the following account entities are pre-created:

- For the Provider account, create the staff member under the name *Training Admin* with the password '*password*'
- Create the customer account under the name *Training Customer* with the staff member of the same name with the password '*password*'
- Create the customer account under the name *Training Customer2* with the staff member of the same name with the password '*password*'
- Create a hosting subscription for *Training Customer2 account*

For the *Training Admin* be sure to do the following:

- Assign the *Account Administrator* role

Before proceeding to perform exercise, copy the content from the `Desktop/APS.Advanced` directory (`Lab-1`, `Lab-2`) to the `Desktop/APS2Adv` directory on the jumper VE.

NOTE: Make sure that the write permission is added to the files.

LAB 1

VPS Cloud (Event Management) Project

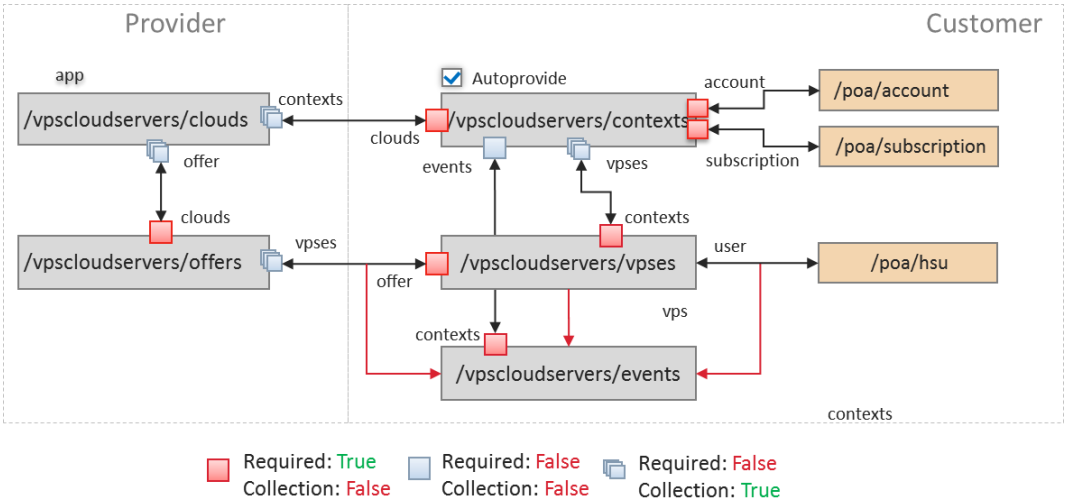
In this set of exercises, you will create a project, which is based on the project of the *Package Development in APS for PA* Lab. In addition to the existing project features, it will be enriched with the service that will monitor VPSes, hosted domains, and service users on the following event types:

- Creating VPS, hosted domain (domain zone), or service user
- Deleting VPS, hosted domain (domain zone), or service user
- Adding or removing VPS link with an offer or service user
- Changing VPS properties

In This Lab

Application Structure

In this exercise, you will build a new application with the following resource model:



The application end-point has the `/vpscloudservers/` alias as the document root. Below it, you will provide access to the services that allows creating resources.

The `APP-META.xml` file declares the following services:

- *clouds*
- *contexts*
- *offers*
- *vpses*
- *events*

For each service declared in the metadata file, a provisioning script must implement the provisioning logic of the service.

- On the provider side:
  - The `clouds.php` script implements the `/vpscloudservers/clouds` service.
  - The `offers.php` script implements the `/vpscloudservers/offers` service.
- On the customer side:
  - The `contexts.php` script implements the `/vpscloudservers/contexts` service.
  - The `vpses.php` script implements the `/vpscloudservers/vpses` service.
  - The `events.php` script implements the `/vpscloudservers/events` service.
- All the types are provisioned along with the resource references according to the scheme.
- With the *events* service, the following events are handled:
  - Creating (making available) a VPS, domain, or user with event type *Available*
  - Removing a VPS, domain, or user with event type *Removed*
  - Linking and unlinking a VPS with an offer or a user with event types *Linked* and *Unlinked* correspondingly
  - Changing any VPS properties with event type *Changed*

Exercise Directions

By completing this set of exercises, you will gain advanced practical skills in assembling of a demo APS application. You will experience the following development stages:

- Setting an application structure.
- Building a package.
- Setting an application end-point.
- Configuring and provisioning an application.

The files, which are required for these exercises are located in the `~/Desktop/APS2Adv/Lab-1` directory:

File	Description
`APP-META.xml`	The metadata file
`clouds.php`	The script implements the *clouds* service
`contexts.php`	The script implements the *contexts* service
`vpses.php`	The script implements the *vpses* service
`offers.php`	The script implements the *offers* service
`events.php`	The script implements the *events* service
`loggers.php`	The subsidiary script implements the event notifications processing
`servers.html`	The file implements a VPS management screen
`server.edit.html`	The file implements a VPS editing screen
`server.new-1.html`	The file implements a first step of VPS creation screen
`server.new-last.html`	The file implements a last step of VPS creation screen
`offers.html`	The file implements an offer management screen on the provider side
`offer.edit.html`	The file implements an offer editing screen on the provider side
`offer.new.html`	The file implements an offer configuration screen on the provider side
`counters.html`	The file implements a resource usage statistics screen
`myservers.html`	The file implements a MyCP screen
`events.html`	The file prints out the event notification log contents
`newvps.json`	The file provides initial values for a VPS
`newoffer.json`	The file provides initial values for an offer

`getUserList.js`	The file retrieve a list of users
`displayError.js`	The file contains the *displayError* function
`server-wizard.js`	The file contains description of the VPS creation steps
`endpoint.sh`	The script is used for the end-point configuration

Below is an overview of actions you are going to perform to assemble the application package.

1 Configure the package structure and create a package:
   - In the lab directory, create the application structure directory and add files to it.
   - Build the application package.
2 Set the application end-point document root by using the *endpoint.sh* script.
3 Deploy and provision the application in POA.
   - Import the application package and install the application instance.
   - Create and configure the application resources and the service template.
   - Create an application subscription and use application services.

Understanding Provisioning Logic

In this exercise, you will learn and explore the provisioning logic of event management.

To verify and examine the application and context logic:

1 On the jumper VE, navigate to the `~/Desktop/APS2Adv/Lab-1/scripts` project directory.
2 In the `contexts.php` file, observe the link to the event monitoring tool.
3 In the `events.php` file, observe the context type.

Assembling Package

In this exercise, you will assemble the APS package using the `apsbuild` utility.

To assemble the package

1 On the jumper VE, navigate to the following directory: `~/Desktop/APS2Adv`.

```
# cd ~/Desktop/APS2Adv
```

2 Run the following command:

```
# apsbuild Lab-1
```

As a result, you can find the new package at `~/Desktop/APS2Adv/VPS_Cloud_Servers-1.0-0.app.zip`

Setting Application End-point

In this exercise, you will learn and explore how to set the end-point for the demo project. Note that in this exercise, you will use the `endpoint.sh` script to configure the application end-point.

Follow the directions below:

1 Copy the application package to the end-point host.

a Run the command:

```
$ scp VPS_Cloud_Servers-1.0-0.app.zip root@10.111.22.16:/root
```

2 Configure the end-point

a Log in to the end-point host via *ssh*

```
$ ssh root@10.111.22.16
```

b Run the script

```
# sh endpoint.sh vpscloudservers /root/VPS_Cloud_Servers-1.0-0.app.zip
```

3 Ensure that the end-point is correctly deployed.

a Verify that the directory root exists and contains all project scripts.

```
# ll -aR /var/www/html/vpscloudservers
```

b Verify that the `.htaccess` contains rewrite rules for all the application services.

```
# less /var/www/html/vpscloudservers/.htaccess
```

c Verify that the end-point correctly responds to HTTP requests.

```
# curl http://poadns.edu.trn/vpscloudservers/events/
```

The output should look like:

```
{"code": 404, "type": "RuntimeException", "message": "Not Found: No appropriate
method found for url vpscloudservers/events"}
```

This concludes setting the endpoint for this project.

Importing Application Package

In this exercise, you will import the APS package through the POA control panel. To import the package:

1 Login to the POA provider control panel.
   a Enter `http://poamn.edu.trn:8080` in the browser address bar.
   b To log into the control panel, use the following credentials:
      - Login: *admin*
      - Password: *setup*
2 Navigate to Services > Applications and click Import Package.
3 Select local file and seek for the package on your jumper VE, e.g.:
   `~/Desktop/APS2Adv/VPS_Cloud_Servers-1.0-0.app.zip`
4 Click the Submit button.
5 In the Applications screen, click Refresh multiple times until package parameters appear.

As a result, the package version (e.g., 1.0-0) appears in the Latest Version column.

Setting Application Instance

In this exercise, you will create and configure an APS application instance.

To create an application instance:

1 In the POA provider control panel, navigate to Services > Applications.
2 Open a profile of the *VPS Cloud Servers* package.
3 On the Instances tab, click Install.
4 Enter the following end-point URI:
`http://10.111.22.16/vpscloudservers`
and click Next.
5 Set the global parameters declared in the `clouds` type as follows:
   a Set the apphost: 10.111.22.16
   b Set the application administrator's credentials as you like and click Next.
6 Click Finish to confirm the settings and complete the installation. If needed, refresh the screen.

   NOTE: The application must be in the Ready status.

7 Create one or more offers.
   a On the Instances tab, open the application instance.
   b On the Offer tab, click New to create new offers.
   c Fill out the offer's parameters, e.g.:
   Offer Name: *VPS Offer Premium*
   CPU cores: 16;
   Diskspace: 64;
   RAM: 8192
   d Click Save Offer.

As a result, we have the application instance configured.

Setting Resource Types

In this exercise, you will create the resource types for the application services.

To create the resource types for the application services.

1 In the POA provider control panel, navigate to Services > Applications.
2 Open a profile of the *VPS Cloud Servers* package and switch to the Resource Types tab.
3 Create the application resource on the basis of the *Application Service Reference* class.
   a Click Create and select *Application Service Reference*.
   b In the Name field, enter *VPS Clouds - Application* and click Next.
   c Select *VPS cloud globals* as the APS type.
   d In the Resource column, click on the instance ID.
   e Click Finish.
4 Create the resource type that will be used for the provisioning of the management contexts for customers.
   a Click Create and select *Application Service*.
   b In the Name field, enter *VPS Clouds - VPS Management* and click Next.
   c Select *VPS Management* as the application service.
   d Leave the Priority field as is.
   e Leave the Automatically provision service check-box unmarked and click Next.
   Why do we leave this checkbox unmarked?

   _____

   _____

   • Click Finish.
5 For provisioning VPS, create the additional resource type.
   a Click Create and select *Application Service*.
   b In the Name field, enter *VPS Clouds - Virtual Server* and click Next.
   c Select *Virtual Private Server* as the application service.
   d Leave the Priority field as is.
   e Leave the Automatically provision service check-box unmarked and click Next.
   f Click Finish.
6 For VPS offers, create the additional resource types.
   a Click Create and select *Application Service Reference*.
   b In the Name field, enter *VPS Clouds - Offer Premium*
   c Select the *VPS Parameters* as the APS type.
   d Click on the instance ID for *VPS Offer Premium*
   e Click Finish.
7 For the CPU resource counter, create the additional resource type.
   a Click Create and select *Application Counter (unit)*.
   b In the Name field, enter *VPS Clouds - CPU Usage*
   c Select the *cpuusage* resource as the APS type.
   d Click Finish.

8 For the memory resource counter, create the additional resource type.
   a Click Create and select *Application Counter (KB)*.
   b In the Name field, enter *VPS Clouds - Memory Usage*
   c Select the *memoryusage* resource as the APS type.
   d Click Finish.
9 For the disk space resource counter, create the additional resource type.
   a Click Create and select *Application Counter (KB)*.
   b For the resource type, use the *VPS Clouds - Disk Usage* name.
   c Select the *diskusage* resource as the APS type.
   d Click Finish.
10 Create the resource for the event notifications monitor of the *Application Service* class.
   a Click Create and select *Application Service*.
   b In the Name field, enter *VPS Clouds - Event Monitor* and click Next.
   c Select *Event Processing* as the application service.
   d Leave the Priority field as is.
   e Select the Automatically provision service check-box and click Next.

   NOTE: On the resource provisioning, this option set will require provisioning of the related VPS Management resource.

   f Click Finish.

As a result, we have all the resources ready to be provisioned.

Setting Service Template

In this exercise, you will create a service template for the VPS Cloud (event processing) APS application.

To create a POA service template for the application:

1 In the POA provider control panel, navigate to Products > Service Templates.
2 Start creating the new service template
   a Click Add New Service Template.
3 Configure the general parameters.
   a In the Name field, enter *VPS Clouds Services*
   b Select the Autoprovisioning check-box.
   c Set the Type option as Custom.
   d Click Next.
4 In the list of resource types, select all the resource types created earlier and click Next.
5 Click Next, then Finish. (Leave the limits as they are).
6 Activate the service template.
   a Open the profile of the newly created service template.
   b In the General section, click Activate.

As the result, you can now use the service template for resource provisioning.

Provisioning Application

In this exercise, you will provision the application.

To create a subscription for a customer, follow the directions below:

1 In the POA provider control panel, navigate to Products > Service Templates.
2 Open a profile of the newly created *VPS Clouds Services* service template.
3 Switch to the Subscriptions tab and click Create New Subscription.
4 Select the *Training Customer*.
5 Do not change the limits and click Next.
6 Click Finish and wait until the screen refreshes.
7 Verify the successful subscription provisioning:
   a Navigate to Operations > Subscriptions.
   b Locate the subscription in the list. Its status should be *Enabled*.

As a result, POA creates the new subscription and provisions the service.

Exploring Application Features

In this exercise, you will learn and explore the new features of the of the *VPS Cloud Server Services* subscription.

To test event notification functionality perform the following actions:

1 Login to the customer control panel.
   a Navigate to Operations > Customers.
   b Open a profile of *Training Customer*.
   c On the General tab, click the Staff Members sub-tab.
   d Click the Login as Customer link.
   e Select the *VPS Cloud Server Services* subscription.
2 Create a sub-domain for the *training2.trn* domain.
   a On the Home tab, click Hosted Domains and then click Add New Subdomain.
   b Specify the sub-domain name, e.g., "sub", and click Next then Finish.
3 Make sure the event notifications monitor is functioning.
   a Navigate to VPS Management > Event Notifications sub-tab.
   b Click Refresh.
   c Verify if the event notification log is printed out.
4 Create a user for *Training Customer*.
   a On the Users tab, click Add New Service User.
   b Fill out the appeared form and click Next and then click Finish.
   c Delete the created service user.
5 Make sure the event notifications monitor is functioning.
   a Navigate to VPS Management > Event Notifications sub-tab and click Refresh.
   b Verify if the event notification log is printed out. As a result, you will notice two notifications.
6 Create a VPS for a user.
   a On the Users tab, add a new service user.
   b Navigate to VPS Management > Servers sub-tab.
   c Click New and specify the VPS parameters.

   NOTE: Here, you will find the limits set during offers' parameters configuration.

   d Click Next then Finish.
7 Change VPS's state.
   a Select the checkbox near the created VPS.
   b Click Start.
8 Make sure the event notifications monitor is functioning.
   a Navigate to VPS Management > Event Notifications sub-tab.
   b Click Refresh.
   c Verify if the event notification log is printed out.

Changing UI Elements

In this exercise, you will learn and explore:

- How to modify a JS code of a screen to change the UI elements view on the fly to test their look and feel.

Follow the directions below:

1 Make sure that the APS development mode in POA is enabled.

a In the POA control panel, navigate to System > Settings > System Properties.

b Ensure that the APS development mode system property is *enabled*.

2 Locate your package in the POA management node.

a In your CLE, login to POA management node through *ssh*, e.g.:

```
$ ssh root@poamn.edu.trn
```

b Navigate to the directory where POA keeps packages, e.g.:

```
# cd /usr/local/pem/APS/packages/
```

c Sort the file list by time to identify the directory that contains the source of your package.

```
# ls -lat
```

NOTE: The destined directory will be the first in the list.

1 Open the screen's HTML file for editing.

a Navigate to the UI subdirectory of the package source directory:

```
# cd 718da66-1c73-4151-9218-b4656bb366c3/ui/
```

b Open the `servers.html` for editing, e.g.:

```
# vim servers.html
```

2 Modify the HTML file of the screen.

a Locate the code block that handles modification of a VPS state:

```
       registry.byId("start").on("click", function() {
                  var self = this;
              var grid = registry.byId("grid");
              var sel = grid.get("selectionArray");
              var page = registry.byId("page");
              var messages = page.get("messageList");
              messages.removeAll();
              for (var i=0; i<sel.length; ++i) {
                    vpsId = sel[i];
                    console.log("I'm trying to start VPS with id = [" + vpsId +
"]");
                    var vps = {
                         aps: { id: vpsId },
                           state: "Running"
                     };
```

b Change the state description from *"Running"* to *"Up and running"*.

c Locate the code block that describes the buttons names.

```
              <div dojoType="aps/Toolbar">
                    <div dojoType="aps/Button" id="new"
iconClass="iconStart">New</div>
                    <div id="start" dojoType="aps/Button" iconClass="iconStart"
>Start</div>
```

```
                    <div id="stop" dojoType="aps/Button"
iconClass="iconStart">Stop</div>
                    <div id="delete" dojoType="aps/Button"
iconClass="iconStart">Delete</div>
```

d Change the name of the button that initiates creation of a new VPS from *"New"* to *"New VPS"*.

3 Save the changes.

4 Start VPS for a user.

a In customer control panel, navigate to VPS Management > Servers sub-tab.

b Refresh the screen.

c Select the checkbox near a VPS and click the Start button.

d Notice the changes you have done.

NOTE: You may need to clean the browser cache by selecting Tools > Clear Browser Data > Empty the cache in your browser in order to the changes be visible.

LAB 2

Asynchronous Provisioning

In this set of exercises, you will modify the *VPS Cloud (Event Management)* application so that VPS can be created asynchronously.

In This Lab

Implementing Business Logic

In this exercise, you will learn and explore:

- How to approach modification of an APS application to implement asynchronous provisioning of the application resources.

Hence, you are going to modify the code of the *VPS Cloud (Event Management)* application modeling the asynchronous provisioning so that a VPS will not be created immediately.

Follow the directions below:

1 On the jumper VE, navigate to the project directory:

```
# cd ~/Desktop/APS2Adv/Lab-1/scripts
```

2 Implement the asynchronous provisioning logic in the `vpses.php` script.

a Open the `vpses.php` script for editing.

```
# vim vpses.php
```

b In the `vpses.php` script declare and implement the *provision()* function, e.g.:

```
public function provision()
{
    $this->state = "Creating";
    throw new \Rest\Accepted($this, "Creating VPS", 60);
}
```

NOTE: You can set task description and timeout as you like.

c In the `vpses.php` script declare and implement the *provisionAsync()* function.

```
public function provisionAsync()
{
    sleep(30); //provisioning code
    $this->state = "Ready";
}
```

3 In the end-point host, replace the `vpses.php` script with its new version.

a Open CLE and copy the `vpses.php` script to the application directory on the end-point host:

```
# cd ~/Desktop/APS2Adv/Lab-1/scripts
# scp vpses.php root@poadns.edu.trn:/var/www/html/vpscloudservers/vpses.php
```

b Confirm you want to rewrite the script (if needed).

Now you can check how it affects the application behavior.

Verifying VPS Provisioning

In this exercise, you will learn and explore

- The new behavior of the *VPS Cloud (Event Management)* application that provisions VPS asynchronously.
- The periodic task scheduled by the APS controller to support asynchronous provisioning.

Follow the directions below:

1 Login to the customer control panel.
   a Navigate to Operations > Customers.
   b Open a profile of *Training Customer*.
   c On the General tab, click the Staff Members sub-tab.
   d Click the Login as Customer link.
2 In the customer control panel, create a new VPS.
   a In the subscription selector, select the *VPS Cloud (Event Management)* subscription.
   b On the VPS Management tab, click New and follow the wizard.
   c Set the VPS's hardware parameters as you want and click Next.
   d Click Finish and notice the VPS's state.

   NOTE: The VPS state must be "Creating" as you have set in the `provision()` method.

   e Refresh the screen to ensure this state does not change.
3 Examine the related POA task.
   a In the provider control panel, navigate to Operations > Tasks.
   b On the Background tab, find the task named as:
   Provisioning "vpses" for APS application

   NOTE: In case of no such a task, open the *Task Log* to locate it there.

   c Open the task and review its description and state.

   NOTE: The description must contain the "Creating VPS" entry.
4 Finally, verify that VPS is provisioned.
   a In the customer control panel, navigate to the VPS Management tab.
   b Refresh the screen and verify the state of the VPS. It must have changed to "Ready".

This concludes the exercise.

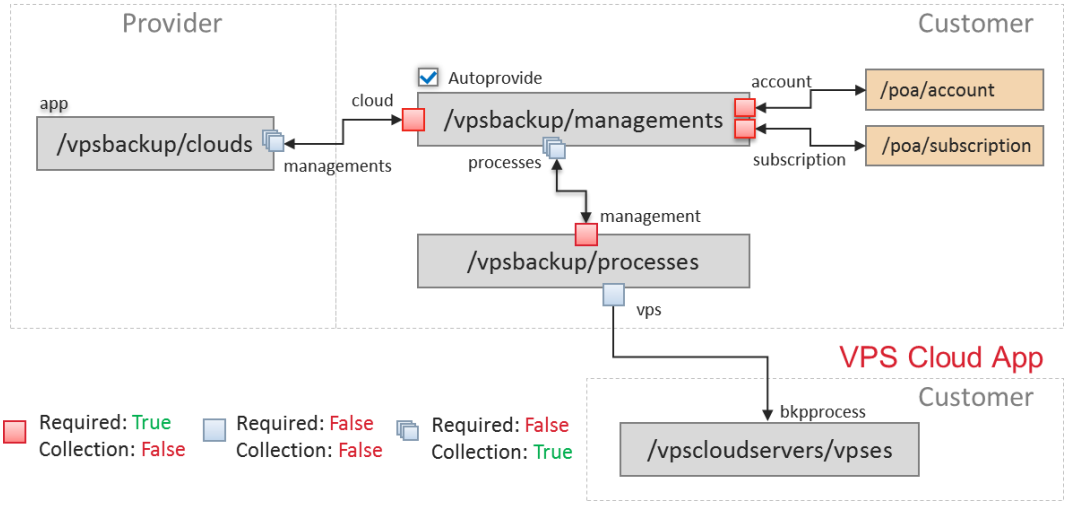
LAB 3

VPS Backup Project

In This Lab

Application Structure

In this exercise, you will build a backup application with the following resource model:



The application end-point has the `/vpsbackup/` alias as the document root. Below it, you will provide access to the services that allows creating resources.

- In the `APP-META.xml` file, declare three services - *clouds*, *managements* and *processes*.
- For each service declared in the metadata file, a provisioning script must implement the provisioning logic of the service.
  - On the provider side, the `clouds.php` script implements the `/vpsbackup/clouds` service.
  - On the customer side, the `managements.php` script implements the `/vpsbackup/managements` service, and the `processes.php` script implements the `/vpsbackup/processes` service.
- The *clouds* type must be provisioned along with the reference to the *managements* (`vpsbackup/managements`) type.
- The *managements* type must be provisioned along with the required references:
  - Reference to the *clouds* type (`vpsbackup/clouds`).
  - Reference to the built-in POA account (`/poa/account`) defines the POA owner of the context.
  - Reference to the built-in POA subscription (`/poa/subscription`) defines the subscription from which the context is created.
  - Reference to the *processes* type (`vpsbackup/processes`) is required.
- The *processes* type must be provisioned along with the references:
  - Reference to the *managements* (`vpsbackup/managements`).
  - Reference to the *vpses* type (`vpscloudservers/vpses`).

Exercise Directions

In this set of exercises, you will gain new practical skills in configuring and deploying the APS application intended for VPS backup processing. The *VPS Backups* APS application provides a backup processing ability for customers' VPSes. Customer accounts subscribed to both applications will be able to create and manage VPSes created in the *VPS Cloud (Event Management)* application and back up their servers using the *VPS Backups* application.

The files, which are required for these exercises are located in the `~/Desktop/APS2Adv/Lab-2` directory:

File	Description
`APP-META.xml`	The metadata file
`clouds.php`	The script implements the *clouds* service for managing application instances
`db_utils.php`	The script implements database management operations
`managements.php`	The script implements the *managements* service for managing the environment on the customer side
`processes.php`	The script implements the *processes* service for managing backup processes
`scheduler.php`	The script implements the backup processes scheduling
`utils.php`	The script implements date-time management functions
`pre-configure.sh`	The script is used to simplify deployment of the package and simulated application on endpoint host
`process.edit.html`	The file implements a backup process editing screen
`process.new.html`	The file implements a backup process creation screen
`processes.html`	The file implements a list of backup processes
`restore.html`	The file implements a restore configuration screen
`newproc.json`	The file provides initial values for backup processing configuration
`displayError.js`	The file contains the *displayError* function
`getVPSlist.js`	The file contains the *getVPSlist* function
`endpoint.sh`	The script is used for the end-point configuration

Below is an overview of actions you are going to perform to assemble the application package.

1 Configure the package structure and create a package:
- In the lab directory, create the application structure directory and add files to it.
- Build the application package.

2 Set the application end-point document root by using the *endpoint.sh* script.
3 Deploy and provision the application in POA.
   - Import the application package and install the application instance.
   - Create and configure the application resources and the service template.
   - Create an application subscription and use application services.

Understanding Provisioning Logic

In this exercise, you will learn and explore the provisioning logic of backup process management.

To verify and examine the application and context logic:

1 On the jumper VE, navigate to the `~/Desktop/APS2Adv/Lab-2/scripts` project directory.
2 Open the `processes.php` file for editing.
   a Add the relation to the *vpses* type.

```
 /**
  * @link("http://edu.trn/vpscloudservers/vpses/1.0")
  */
 public $vps;
```

```
// Optional singular link with a VPS of an external application

/**
 * @link("http://edu.trn/vpscloudservers/vpses/1.0")
 */
public $vps;
```

3 Save changes and close the files.
4 In the `clouds.php` file, locate the *provision()* function.

   What does it serve for? _____

5 In the `process.new.html` file, locate the *getVPSList()* function.

   What is it intended for? _____

Assembling Package

In this exercise, you will assemble the APS package using the `apsbuild` utility.

To assemble the package

1 On the jumper VE, navigate to the following directory: `~/Desktop/APS2Adv`.

```
# cd ~/Desktop/APS2Adv
```

2 Run the following command:

```
# apsbuild Lab-2
```

As a result, you can find the new package at `~/Desktop/APS2Adv/VPS_Backups-1.0-0.app.zip`

Setting Application End-Point

In this exercise, you will learn and explore how to set the end-point for the demo project.

Follow the directions below:

1 Copy the application package to the end-point host.

a On the jumper VE, run the command:

```
$ cd ~/Desktop/APS2Adv/
$ scp VPS_Backups-1.0-0.app.zip root@10.111.22.16:/root
```

2 Configure the end-point

a Log in to the end-point host via *ssh*

```
$ ssh root@10.111.22.16
```

b Run the script

```
# sh ./endpoint.sh vpsbackup /root/VPS_Backups-1.0-0.app.zip
```

Wait for the installation of all components including the *MySQL* database will be completed.

3 Ensure that the end-point is correctly deployed.

a Verify that the directory root exists and contains all project scripts.

```
# ll -aR /var/www/html/vpsbackup
```

b Verify that MySQL and PHP services needed for the simulated application are installed.

```
# rpm -qa | egrep 'mysql|php'
```

c Verify that the `.htaccess` contains rewrite rules for all the application services.

```
# less /var/www/html/vpsbackup/.htaccess
```

d Verify that the end-point correctly responds to HTTP requests.

```
# curl http://poadns.edu.trn/vpsbackup/clouds/
```

The output should look like:

```
{"code": 404, "type": "RuntimeException", "message": "Not Found: No appropriate
method found for url vpsbackup/clouds"}
```

This concludes setting the endpoint for this project.

Importing Application Package

In this exercise, you will import the APS package through the POA control panel. To import the package:

1 Login to the POA provider control panel.

2 Navigate to Services > Applications and click Import Package.

3 Select local file and seek for the package on your jumper VE, e.g.:
`~/Desktop/APS2Adv/VPS_Backups-1.0-0.app.zip`

4 Click the Submit button.

5 In the Applications screen, click Refresh multiple times until package parameters appear.

As a result, the package version (e.g., 1.0-0) appears in the Latest Version column.

Setting Application Instance

In this exercise, you will create and configure an APS application instance.

To create an application instance:

1 In the POA provider control panel, navigate to Services > Applications.
2 Open a profile of the *VPS Backups* package.
3 On the Instances tab, click Install.
4 Enter the following end-point URI:
`http://10.111.22.16/vpsbackup`
and click Next.
5 Set the global parameters declared in the `clouds` type as follows:
a Set the Host domain or IP: 10.111.22.16
b Set the application administrator's credentials as you like and click Next.
6 Click Finish to confirm the settings and complete the installation. If needed, refresh the screen.

NOTE: The application must be in the Ready status.

As a result, we have configured application instance.

Setting Resource Types

In this exercise, you will create the resource types for the application services.

To create the resource types for the application services.

1 In the POA provider control panel, navigate to Services > Applications.
2 Open a profile of the *VPS Backups* package and switch to the Resource Types tab.
3 Create the backup application resource on the basis of the *Application Service Reference* class.
   a Click Create and select *Application Service Reference*.
   b In the Name field, enter *VPS Backups - Backup Cloud* and click Next.
   c Select *Backup Cloud* as the APS type.
   d In the Resource column, click on the instance ID.
   e Click Finish.
4 Create the resource type that will be used for the provisioning of the backup management environment.
   a Click Create and select *Application Service*.
   b In the Name field, enter *VPS Backups - Backup Management* and click Next.
   c Select *Backup Management* as the application service.
   d Leave the Priority field as is.
   e Select the Automatically provision service check-box and click Next.
   f Click Finish.
5 Create the resource type that will be used for the backup processing.
   a Click Create and select *Application Service*.
   b In the Name field, enter *VPS Backups - Backup Process* and click Next.
   c Select *Backup Process* as the application service.
   d Leave the Priority field as is.
   e Leave the Automatically provision service check-box unmarked and click Next.
   f Click Finish.

As a result, we have all the resources ready to be provisioned.

Setting Service Template

In this exercise, you will create a service template for the *VPS Backup* APS application.

To create a POA service template for the application:

1 In the POA provider control panel, navigate to Products > Service Templates.
2 Start creating the new service template
   a Click Add New Service Template.
3 Configure the general parameters.
   a In the Name field, enter *VPS Backups*
   b Select the Autoprovisioning check-box.
   c Set the Type option as Custom.
   d Click Next.
4 In the list of resource types, select all the resource types created earlier and click Next.
5 Click Next, then Finish. (Leave the limits as they are).
6 Activate the service template.
   a Open the profile of the newly created service template.
   b In the General section, click Activate.

As the result, you can now use the service template for resource provisioning.

Provisioning Application

In this exercise, you will provision the application.

To create a subscription for a customer, follow the directions below:

1 In the POA provider control panel, navigate to Products > Service Templates.
2 Open a profile of the newly created *VPS Backups* service template.
3 Switch to the Subscriptions tab and click Create New Subscription.
4 Select the *Training Customer*.
5 Do not change the limits and click Next.
6 Click Finish and wait until the screen refreshes.
7 Verify the successful subscription provisioning:
   a Navigate to Operations > Subscriptions.
   b Locate the subscription in the list. Subscription status should be *Enabled*.

As a result, POA creates the new subscription and provisions the service.

Exploring Application Features

To ensure the backup processing of the APS application and test its functionality:

1 Login to the customer control panel on behalf of *Training Customer*.

2 Examine VPSes backup processing task.

   a Ensure the *VPS Clouds Services* subscription is selected.

   b On the VPS Management tab, create several VPSes.

   c Switch to the *VPS Backups* subscription.

   d On the VPS Backup tab, click New.

   e In the Process Name field, enter a name.

   f In the VPS Name drop-down box, select a VPS and click Submit.

   g Make sure that the backup process is scheduled.

3 Now let us examine a backup processing task without VPSes.

   a Switch to the *VPS Clouds Services* subscription.

   b On the VPS Management tab, delete all VPSes.

   c Switch to the *VPS Backups* subscription.

   d On the VPS Backup tab, click New.

   e Notice that there are some other fields, which allows application to work in a standalone mode.

As a result, we experienced the integration of two APS applications.

LAB 4

Enhanced Offer

In this exercise, you will combine resources of the two integrated applications: *VPS Backup* and *VPS Cloud*.

In This Lab

Exercise Directions

In the following part of the lab you will make use the advantage of both APS applications by integrating their resources into a single subscription.

The key difference between *the single subscription* with *two applications* and *two separate subscriptions* with *one application* is:

- No need to switch between two subscriptions.
- Customer control panel includes two tabs with separate application management functionality.

Below is an overview of actions you are going to perform to assemble the application package.

1 Configure the service template in the provider control panel.
   a Create the service template, which includes the resources of both applications.
2 Deploy and provision the application in POA.
   a Create an application subscription and check the subscription.

Setting Service Template and Provisioning Application

In this exercise, you will create a service template, which includes *VPS Cloud Services* and *Backup* applications.

To create a POA service template:

1 In the POA provider control panel, navigate to Products > Service Templates.
2 Start creating the new service template
   a Click Add New Service Template.
3 Configure the general parameters.
   a In the Name field, enter *VPS Clouds Services + Backup*
   b Select the Autoprovisioning check-box.
   c Set the Type option as Custom.
   d Click Next.
4 In the list of resource types, select all the resource types created earlier for both applications and click Next.
5 Click Next, then Finish.
6 Activate the service template.
   a Open the profile of the newly created service template.
   b In the General section, click Activate.

After the service template is set, create a subscription for the *Training Customer 2* account as you did earlier.

Exploring Application Features

To ensure the backup processing of the APS application and test its functionality:

1 Login to the customer control panel on behalf of *Training Customer 2*.
2 Ensure the *VPS Clouds Services + Backups* subscription is selected.
3 Examine a backup processing task.
   a On the Users tab, create a user account.
   b On the VPS Management tab, create several VPSes.
   c On the VPS Backup tab, click New.
   d In the Process Name field, enter a name.
   e In the VPS Name drop-down box, select a VPS and click Submit.
   f Make sure that the backup process is scheduled.

As a result, we experienced a single subscription with resources of two APS applications.