MyPLC User's Guide

Mark Huang

Revision History

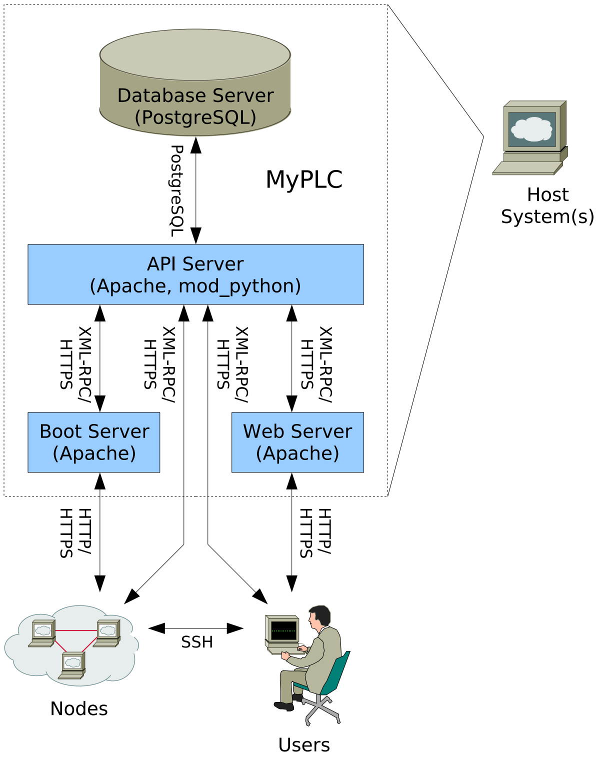
Revision 1.0   April 7, 2006   Revised by: MLH
Initial draft.

This document describes the design, installation, and administration of MyPLC, a complete PlanetLab Central (PLC) portable installation contained within a **chroot** jail. This document assumes advanced knowledge of the PlanetLab architecture and Linux system administration.

Table of Contents

Overview

MyPLC is a complete PlanetLab Central (PLC) portable installation contained within a **chroot** jail. The default installation consists of a web server, an XML-RPC API server, a boot server, and a database server: the core components of PLC. The installation is customized through an easy-to-use graphical interface. All PLC services are started up and shut down through a single script installed on the host system. The usually complex process of installing and administering the PlanetLab backend is reduced by containing PLC services within a virtual filesystem. By packaging it in such a manner, MyPLC may also be run on any modern Linux distribution, and could conceivably even run in a PlanetLab slice.



MyPLC should be viewed as a single application that provides multiple functions and can run on any host system.

Figure 1. MyPLC architecture

Installation

Though internally composed of commodity software subpackages, MyPLC should be treated as a monolithic software application. MyPLC is distributed as single RPM package that has no external dependencies, allowing it to be installed on practically any Linux 2.6 based distribution:

Example 1. Installing MyPLC.

```
# If your distribution supports RPM
rpm -U myplc-0.3-1.planetlab.i386.rpm
```

```
# If your distribution does not support RPM
cd /
rpm2cpio myplc-0.3-1.planetlab.i386.rpm | cpio -diu
```

MyPLC installs the following files and directories:

- `/plc/root.img`: The main root filesystem of the MyPLC application. This file is an uncompressed ext3 filesystem that is loopback mounted on `/plc/root` when MyPLC starts. The filesystem, even when mounted, should be treated an opaque binary that can and will be replaced in its entirety by any upgrade of MyPLC.
- `/plc/root`: The mount point for `/plc/root.img`. Once the root filesystem is mounted, all MyPLC services run in a **chroot** jail based in this directory.
- `/plc/data`: The directory where user data and generated files are stored. This directory is bind mounted into the **chroot** jail on `/data`. Files in this directory are marked with **%config(noreplace)** in the RPM. That is, during an upgrade of MyPLC, if a file has not changed since the last installation or upgrade of MyPLC, it is subject to upgrade and replacement. If the file has chanegd, the new version of the file will be created with a `.rpmnew` extension. Symlinks within the MyPLC root filesystem ensure that the following directories (relative to `/plc/root`) are stored outside the MyPLC filesystem image:
  - `/etc/planetlab`: This directory contains the configuration files, keys, and certificates that define your MyPLC installation.
  - `/var/lib/pgsql`: This directory contains PostgreSQL database files.
  - `/var/www/html/alpina-logs`: This directory contains node installation logs.
  - `/var/www/html/boot`: This directory contains the Boot Manager, customized for your MyPLC installation, and its data files.
  - `/var/www/html/download`: This directory contains Boot CD images, customized for your MyPLC installation.
  - `/var/www/html/install-rpms`: This directory is where you should install node package updates, if any. By default, nodes are installed from the tarball located at `/var/www/html/boot/PlanetLab-Bootstrap.tar.bz2`, which is pre-built from the latest PlanetLab Central sources, and installed as part of your MyPLC installation. However, nodes will attempt to install any newer RPMs located in `/var/www/html/install-rpms/planetlab`, after initial installation and periodically thereafter. You must run **yum-arch** and **createrepo** to update the **yum** caches in this directory after installing a new RPM. PlanetLab Central cannot support any changes to this directory.
  - `/var/www/html/xml`: This directory contains various XML files that the Slice Creation Service uses to determine the state of slices. These XML files are refreshed periodically by **cron** jobs running in the MyPLC root.
- `/etc/init.d/plc`: This file is a System V init script installed on your host filesystem, that allows you to start up and shut down MyPLC with a single command. On a Red Hat or Fedora host system, it is customary to use the **service** command to invoke System V init scripts:

  Example 2. Starting and stopping MyPLC.

  ```
  # Starting MyPLC
  service plc start

  # Stopping MyPLC
  service plc stop
  ```

Like all other registered System V init services, MyPLC is started and shut down automatically when your host system boots and powers off. You may disable automatic startup by invoking the **chkconfig** command on a Red Hat or Fedora host system:

Example 3. Disabling automatic startup of MyPLC.

```
# Disable automatic startup
chkconfig plc off

# Enable automatic startup
chkconfig plc on
```

- `/etc/sysconfig/plc`: This file is a shell script fragment that defines the variables PLC_ROOT and PLC_DATA. By default, the values of these variables are `/plc/root` and `/plc/data`, respectively. If you wish, you may move your MyPLC installation to another location on your host filesystem and edit the values of these variables appropriately, but you will break the RPM upgrade process. PlanetLab Central cannot support any changes to this file.
- `/etc/planetlab`: This symlink to `/plc/data/etc/planetlab` is installed on the host system for convenience.

Quickstart

Once installed, start MyPLC (see Example 2). MyPLC must be started as root. Observe the output of this command for any failures. If no failures occur, you should see output similar to the following:

Example 4. A successful MyPLC startup.

```
Mounting PLC:                                               [  OK  ]
PLC: Generating network files:                              [  OK  ]
PLC: Starting system logger:                                [  OK  ]
PLC: Starting database server:                              [  OK  ]
PLC: Generating SSL certificates:                           [  OK  ]
PLC: Generating SSH keys:                                   [  OK  ]
PLC: Starting web server:                                   [  OK  ]
PLC: Bootstrapping the database:                            [  OK  ]
PLC: Starting DNS server:                                   [  OK  ]
PLC: Starting crond:                                        [  OK  ]
PLC: Rebuilding Boot CD:                                    [  OK  ]
PLC: Rebuilding Boot Manager:                               [  OK  ]
PLC: Signing node packages:                                 [  OK  ]
```

If `/plc/root` is mounted successfully, a complete log file of the startup process may be found at `/plc/root/var/log/boot.log`. Possible reasons for failure of each step include:

- `Mounting PLC`: If this step fails, first ensure that you started MyPLC as root. Check `/etc/sysconfig/plc` to ensure that PLC_ROOT and PLC_DATA refer to the right locations. You may also have too many existing loopback mounts, or your kernel may not support loopback mounting, bind mounting, or the ext3 filesystem. Try freeing at least one loopback device, or re-compiling your kernel to support loopback mounting, bind mounting, and the ext3 filesystem. SELinux may also be enabled. If you install MyPLC on Fedora Core 4 or 5, use the `Security Level Configuration` utility to configure SELinux to be `Permissive`.
- `Starting database server`: If this step fails, check `/plc/root/var/log/pgsql` and `/plc/root/var/log/boot.log`. The most common reason for failure is that

the default PostgreSQL port, TCP port 5432, is already in use. Check that you are not running a PostgreSQL server on the host system.

- `Starting web server`: If this step fails, check `/plc/root/var/log/httpd/error_log` and `/plc/root/var/log/boot.log` for obvious errors. The most common reason for failure is that the default web ports, TCP ports 80 and 443, are already in use. Check that you are not running a web server on the host system.
- `Bootstrapping the database`: If this step fails, it is likely that the previous step (`Starting web server`) also failed. Another reason that it could fail is if PLC_API_HOST (see the Section called *Changing the configuration*) does not resolve to the host on which the API server has been enabled. By default, all services, including the API server, are enabled and run on the same host, so check that PLC_API_HOST is either `localhost` or resolves to a local IP address.
- `Starting crond`: If this step fails, it is likely that the previous steps (`Starting web server` and `Bootstrapping the database`) also failed. If not, check `/plc/root/var/log/boot.log` for obvious errors. This step starts the **cron** service and generates the initial set of XML files that the Slice Creation Service uses to determine slice state.

If no failures occur, then MyPLC should be active with a default configuration. Open a web browser on the host system and visit `http://localhost/`, which should bring you to the front page of your PLC installation. The password of the default administrator account `root@localhost.localdomain` (set by PLC_ROOT_USER) is `root` (set by PLC_ROOT_PASSWORD).

Changing the configuration

After verifying that MyPLC is working correctly, shut it down and begin changing some of the default variable values. Shut down MyPLC with **service plc stop** (see Example 2). With a text editor, open the file `/etc/planetlab/plc_config.xml`. This file is a self-documenting configuration file written in XML. Variables are divided into categories. Variable identifiers must be alphanumeric, plus underscore. A variable is referred to canonically as the uppercase concatenation of its category identifier, an underscore, and its variable identifier. Thus, a variable with an `id` of `slice_prefix` in the `plc` category is referred to canonically as PLC_SLICE_PREFIX.

The reason for this convention is that during MyPLC startup, `plc_config.xml` is translated into several different languages—shell, PHP, and Python—so that scripts written in each of these languages can refer to the same underlying configuration. Most MyPLC scripts are written in shell, so the convention for shell variables predominates.

The variables that you should change immediately are:

- PLC_NAME: Change this to the name of your PLC installation.
- PLC_ROOT_PASSWORD: Change this to a more secure password.
- PLC_MAIL_SUPPORT_ADDRESS: Change this to the e-mail address at which you would like to receive support requests.
- PLC_DB_HOST, PLC_DB_IP, PLC_API_HOST, PLC_API_IP, PLC_WWW_HOST, PLC_WWW_IP, PLC_BOOT_HOST, PLC_BOOT_IP: Change all of these to the preferred FQDN and external IP address of your host system.

After changing these variables, save the file, then restart MyPLC with **service plc start**. You should notice that the password of the default administrator account is no longer `root`, and that the default site name includes the name of your PLC installation instead of PlanetLab.

Installing nodes

Install your first node by clicking `Add Node` under the `Nodes` tab. Fill in all the appropriate details, then click `Add`. Download the node's configuration file by clicking `Download configuration file` on the *Node Details* page for the node. Save it to a floppy disk or USB key as detailed in [1].

Follow the rest of the instructions in [1] for creating a Boot CD and installing the node, except download the Boot CD image from the `/download` directory of your PLC installation, not from PlanetLab Central. The images located here are customized for your installation. If you change the hostname of your boot server (PLC_BOOT_HOST), or if the SSL certificate of your boot server expires, MyPLC will regenerate it and rebuild the Boot CD with the new certificate. If this occurs, you must replace all Boot CDs created before the certificate was regenerated.

The installation process for a node has significantly improved since PlanetLab 3.3. It should now take only a few seconds for a new node to become ready to create slices.

Administering nodes

You may administer nodes as `root` by using the SSH key stored in `/etc/planetlab/root_ssh_key.rsa`.

Example 5. Accessing nodes via SSH. Replace `node` with the hostname of the node.

```
ssh -i /etc/planetlab/root_ssh_key.rsa root@node
```

Besides the standard Linux log files located in `/var/log`, several other files can give you clues about any problems with active processes:

- `/var/log/pl_nm`: The log file for the Node Manager.
- `/vservers/pl_conf/var/log/pl_conf`: The log file for the Slice Creation Service.
- `/var/log/propd`: The log file for Proper, the service which allows certain slices to perform certain privileged operations in the root context.
- `/vservers/pl_netflow/var/log/netflow.log`: The log file for PlanetFlow, the network traffic auditing service.

Creating a slice

Create a slice by clicking `Create Slice` under the `Slices` tab. Fill in all the appropriate details, then click `Create`. Add nodes to the slice by clicking `Manage Nodes` on the *Slice Details* page for the slice.

A **cron** job runs every five minutes and updates the file `/plc/data/var/www/html/xml/slices-0.5.xml` with information about current slice state. The Slice Creation Service running on every node polls this file every ten minutes to determine if it needs to create or delete any slices. You may accelerate this process manually if desired.

Example 6. Forcing slice creation on a node.

```
# Update slices.xml immediately
service plc start crond

# Kick the Slice Creation Service on a particular node.
ssh -i /etc/planetlab/root_ssh_key.rsa root@node \
vserver pl_conf exec service pl_conf restart
```

A. Configuration variables

Listed below is the set of standard configuration variables and their default values, defined in the template `/etc/planetlab/default_config.xml`. Additional variables and their defaults may be defined in site-specific XML templates that should be placed in `/etc/planetlab/configs/`.

PLC_NAME

Type: string

Default: PlanetLab Test

The name of this PLC installation. It is used in the name of the default system site (e.g., PlanetLab Central) and in the names of various administrative entities (e.g., PlanetLab Support).

PLC_SLICE_PREFIX

Type: string

Default: pl

The abbreviated name of this PLC installation. It is used as the prefix for system slices (e.g., pl_conf). Warning: Currently, this variable should not be changed.

PLC_ROOT_USER

Type: email

Default: root@localhost.localdomain

The name of the initial administrative account. We recommend that this account be used only to create additional accounts associated with real administrators, then disabled.

PLC_ROOT_PASSWORD

Type: password

Default: root

The password of the initial administrative account. Also the password of the root account on the Boot CD.

PLC_ROOT_SSH_KEY_PUB

Type: file

Default: /etc/planetlab/root_ssh_key.pub

The SSH public key used to access the root account on your nodes.

PLC_ROOT_SSH_KEY

Type: file

Default: /etc/planetlab/root_ssh_key.rsa

The SSH private key used to access the root account on your nodes.

PLC_DEBUG_SSH_KEY_PUB

Type: file

Default: /etc/planetlab/debug_ssh_key.pub

The SSH public key used to access the root account on your nodes when they are in Debug mode.

PLC_DEBUG_SSH_KEY

Type: file

Default: /etc/planetlab/debug_ssh_key.rsa

The SSH private key used to access the root account on your nodes when they are in Debug mode.

PLC_ROOT_GPG_KEY_PUB

Type: file

Default: /etc/planetlab/pubring.gpg

The GPG public keyring used to sign the Boot Manager and all node packages.

PLC_ROOT_GPG_KEY

Type: file

Default: /etc/planetlab/secring.gpg

The SSH private key used to access the root account on your nodes.

PLC_NET_DNS1

Type: ip

Default: 127.0.0.1

Primary DNS server address.

PLC_NET_DNS2

Type: ip

Default:

Secondary DNS server address.

PLC_DNS_ENABLED

Type: boolean

Default: true

Enable the internal DNS server. The server does not provide reverse resolution and is not a production quality or scalable DNS solution. Use the internal DNS server only for small deployments or for testing.

PLC_MAIL_ENABLED

Type: boolean

Default: false

Set to false to suppress all e-mail notifications and warnings.

PLC_MAIL_SUPPORT_ADDRESS

Type: email

Default: root+support@localhost.localdomain

This address is used for support requests. Support requests may include traffic complaints, security incident reporting, web site malfunctions, and general requests for information. We recommend that the address be aliased to a ticketing system such as Request Tracker.

PLC_MAIL_BOOT_ADDRESS

Type: email

Default: root+install-msgs@localhost.localdomain

The API will notify this address when a problem occurs during node installation or boot.

PLC_MAIL_SLICE_ADDRESS

Type: email

Default: root+SLICE@localhost.localdomain

This address template is used for sending e-mail notifications to slices. SLICE will be replaced with the name of the slice.

PLC_DB_ENABLED

Type: boolean

Default: true

Enable the database server on this machine.

PLC_DB_TYPE

Type: string

Default: postgresql

The type of database server. Currently, only postgresql is supported.

PLC_DB_HOST

Type: hostname

Default: localhost.localdomain

The fully qualified hostname of the database server.

PLC_DB_IP

Type: ip

Default: 127.0.0.1

The IP address of the database server, if not resolvable by the configured DNS servers.

PLC_DB_PORT

Type: int

Default: 5432

The TCP port number through which the database server should be accessed.

PLC_DB_NAME

Type: string

Default: planetlab3

The name of the database to access.

PLC_DB_USER

Type: string

Default: pgsqluser

The username to use when accessing the database.

PLC_DB_PASSWORD

Type: password

Default:

The password to use when accessing the database. If left blank, one will be generated.

PLC_API_ENABLED

Type: boolean

Default: true

Enable the API server on this machine.

PLC_API_DEBUG

Type: boolean

Default: false

Enable verbose API debugging. Do not enable on a production system!

PLC_API_HOST

Type: hostname

Default: localhost.localdomain

The fully qualified hostname of the API server.

PLC_API_IP

Type: ip

Default: 127.0.0.1

The IP address of the API server, if not resolvable by the configured DNS servers.

PLC_API_PORT

Type: int

Default: 80

The TCP port number through which the API should be accessed. Warning: SSL (port 443) access is not fully supported by the website code yet. We recommend that port 80 be used for now and that the API server either run on the same machine as the web server, or that they both be on a secure wired network.

PLC_API_PATH

Type: string

Default: /PLCAPI/

The base path of the API URL.

PLC_API_MAINTENANCE_USER

Type: string

Default: maint@localhost.localdomain

The username of the maintenance account. This account is used by local scripts that perform automated tasks, and cannot be used for normal logins.

PLC_API_MAINTENANCE_PASSWORD

Type: password

Default:

The password of the maintenance account. If left blank, one will be generated. We recommend that the password be changed periodically.

PLC_API_MAINTENANCE_SOURCES

Type: hostname

Default:

A space-separated list of IP addresses allowed to access the API through the maintenance account. The value of this variable is set automatically to allow only the API, web, and boot servers, and should not be changed.

PLC_API_SSL_CRT

Type: file

Default: /etc/planetlab/api_ssl.crt

The signed SSL certificate to use for HTTPS access. If not specified or non-existent, a self-signed certificate will be generated.

PLC_API_SSL_KEY

Type: file

Default: /etc/planetlab/api_ssl.key

The corresponding SSL private key used for signing the certificate, and for signing slice tickets. If not specified or non-existent, one will be generated.

PLC_API_SSL_KEY_PUB

Type: file

Default: /etc/planetlab/api_ssl.pub

The corresponding SSL public key. If not specified or non-existent, one will be generated.

PLC_WWW_ENABLED

Type: boolean

Default: true

Enable the web server on this machine.

PLC_WWW_DEBUG

Type: boolean

Default: false

Enable debugging output on web pages. Do not enable on a production system!

PLC_WWW_HOST

Type: hostname

Default: localhost.localdomain

The fully qualified hostname of the web server.

PLC_WWW_IP

Type: ip

Default: 127.0.0.1

The IP address of the web server, if not resolvable by the configured DNS servers.

PLC_WWW_PORT

Type: int

Default: 80

The TCP port number through which the unprotected portions of the web site should be accessed.

PLC_WWW_SSL_PORT

Type: int

Default: 443

The TCP port number through which the protected portions of the web site should be accessed.

PLC_WWW_SSL_CRT

Type: file

Default: /etc/planetlab/www_ssl.crt

The signed SSL certificate to use for HTTPS access. If not specified or non-existent, a self-signed certificate will be generated.

PLC_WWW_SSL_KEY

Type: file

Default: /etc/planetlab/www_ssl.key

The corresponding SSL private key. If not specified or non-existent, one will be generated.

PLC_BOOT_ENABLED

Type: boolean

Default: true

Enable the boot server on this machine.

PLC_BOOT_HOST

Type: hostname

Default: localhost.localdomain

The fully qualified hostname of the boot server.

PLC_BOOT_IP

Type: ip

Default: 127.0.0.1

The IP address of the boot server, if not resolvable by the configured DNS servers.

PLC_BOOT_PORT

Type: int

Default: 80

The TCP port number through which the unprotected portions of the boot server should be accessed.

PLC_BOOT_SSL_PORT

Type: int

Default: 443

The TCP port number through which the protected portions of the boot server should be accessed.

PLC_BOOT_SSL_CRT

Type: binary

Default: /etc/planetlab/boot_ssl.crt

The signed SSL certificate to use for HTTPS access. If not specified, or nonexistent a self-signed certificate will be generated.

PLC_BOOT_SSL_KEY

Type: binary

Default: /etc/planetlab/boot_ssl.key

The corresponding SSL private key. If not specified or non-existent, one will be generated.

Bibliography

[1] Mark Huang, *PlanetLab Technical Contact's Guide*[1].

Notes

1. http://www.planet-lab.org/doc/TechsGuide.php