

Introduction to cPanel® Troubleshooting

Presented by: Stephen Bee

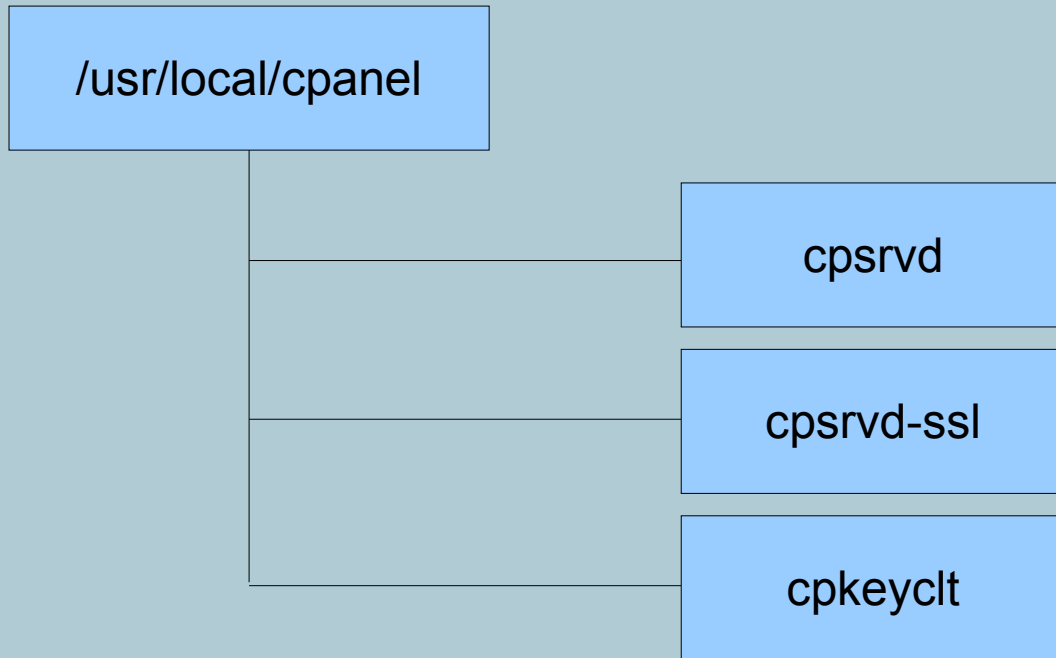
Topics Covered

- **cPanel Essentials**
 - Core directory structure
 - Configuration Files
- cPanel Services
- cPanel Maintenance
- cPanel Scripts

Core Directory Structure

- cPanel integrates itself into many locations within the host system, with the following three being the most notable:
 - /usr/local/cpanel
 - /var/cpanel
 - /scripts

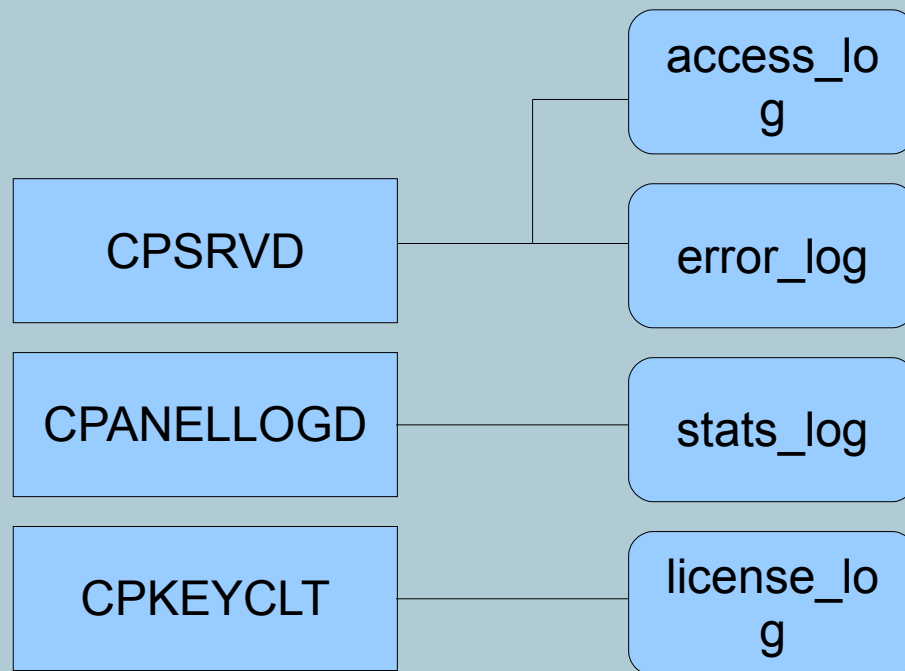
/usr/local/cpanel



/usr/local/cpanel/bin

- Houses only scripts and binaries which provide installation and configuration of many cPanel managed services
- Notable Contents:
- `eximstats`
- `checkperlmodules`

/usr/local/cpanel/logs



access_log example

Breakdown of an access_log

[source address] – [username] [[date]] [status] [url] [browser agent]

Example access_log entries:

```
127.0.0.1 - root [31/May/2006:23:11:31 -0500] "" 500 0
"https://www.cpane.net:2087/scripts/command?PFILE=main" "Mozilla/5.0 (X11; U; Linux
i686; en-US; rv:1.8.0.3) Gecko/20060512 Firefox/1.5.0.3"
127.0.0.1 - root [31/May/2006:23:11:31 -0500] "" 500 0
"https://www.cpanel.net:2087/scripts/command" "Mozilla/5.0 (X11; U; Linux i686; en-US;
rv:1.8.0.3) Gecko/20060512 Firefox/1.5.0.3"
```

error_log example

Example Errors:

```
Can't locate Net/AIM/Connection.pm in @INC (@INC contains: /scripts /usr/lib/perl5/5.8.7/i686-linux... /usr/lib/perl5/site_perl/5.8.7/i686-linux /usr/lib/perl5/site_perl/5.8.7 /usr/lib/perl5/site_perl/5.8.5 /usr/lib... /usr/lib/perl5/site_perl/5.8.3 /usr/lib/perl5/site_perl/5.8.2 /usr/lib/perl5/site_perl/5.8.1 /usr/lib/perl5/site... /usr/lib/perl5/site_perl .) at /usr/lib/perl5/site_perl/5.8.7/Net/AIM.pm line 37.  
BEGIN failed--compilation aborted at /usr/lib/perl5/site_perl/5.8.7/Net/AIM.pm line 37.  
Compilation failed in require at /scripts/cPScript/iContact.pm line 14.  
BEGIN failed--compilation aborted at /scripts/cPScript/iContact.pm line 14.  
Compilation failed in require at /scripts/wwwacct line 18.  
BEGIN failed--compilation aborted at /scripts/wwwacct line 18.
```


license_log example

Example of license_log entry:

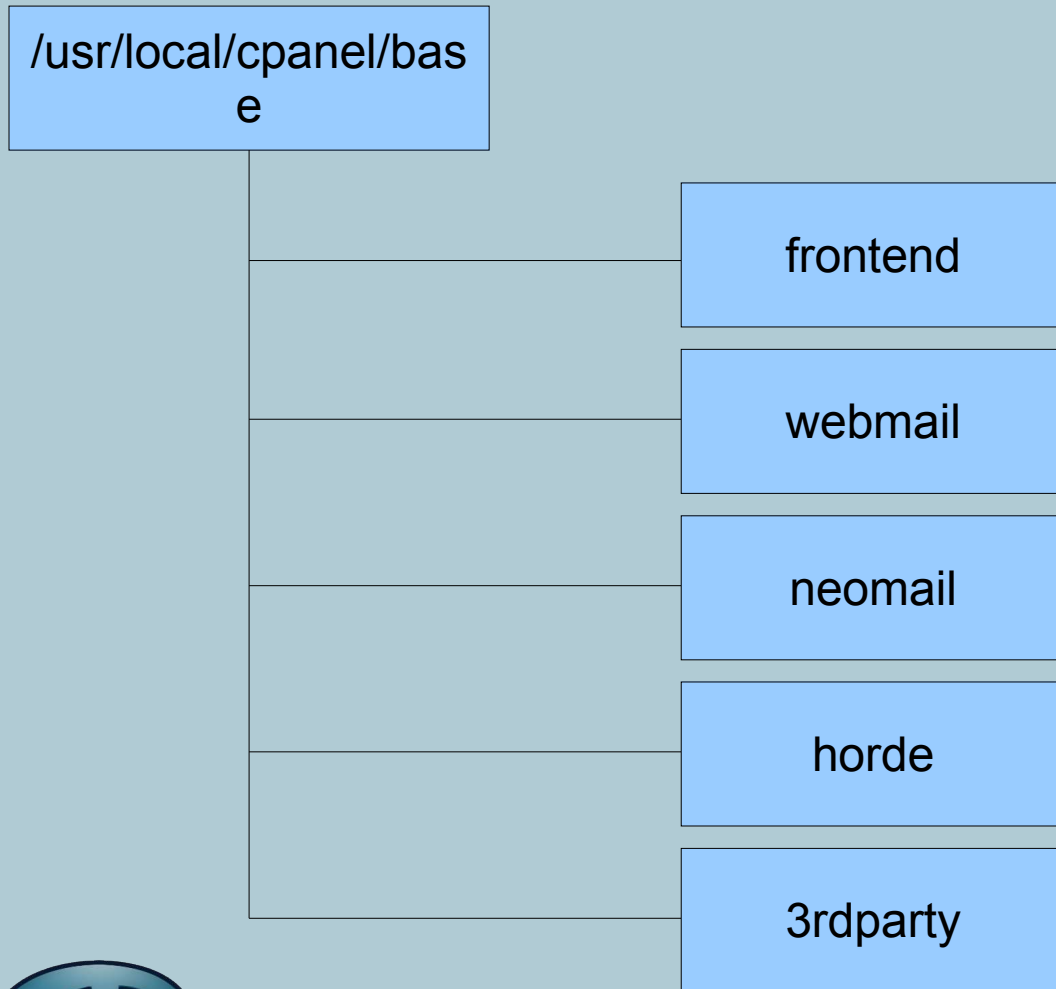
```
License Update Request at: Wed May 31  
02:16:13 2006  
License Server returned: Key Accepted  
License Server returned: Key Follows  
License Updated Successfully
```

stats_log example

Example of stats_log entry:

Update for config
"/home/techdump/tmp/awstats/awstats.cpanel.net.conf"
With data in log file "/usr/local/apache/domlogs/techdump.net"...
Phase 1 : First bypass old records, searching new record...
Direct access to last remembered record is out of file.
So searching it from beginning of log file...
Phase 2 : Now process new records (Flush history on disk after
20000 hosts)...
Jumped lines in file: 0
Parsed lines in file: 14
Found 0 dropped records,
Found 0 corrupted records,
Found 0 old records,
Found 14 new qualified records.

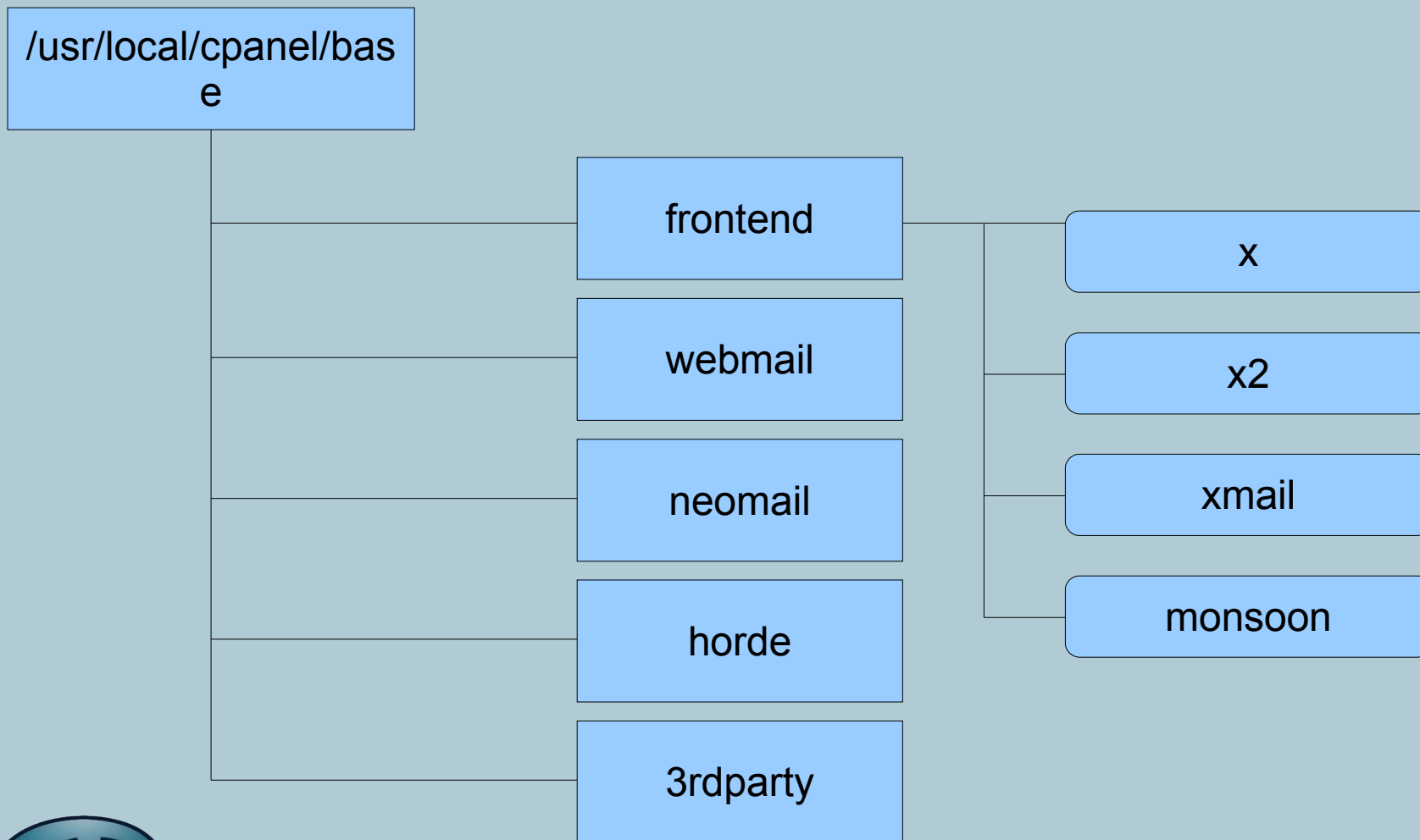
/usr/local/cpanel/base



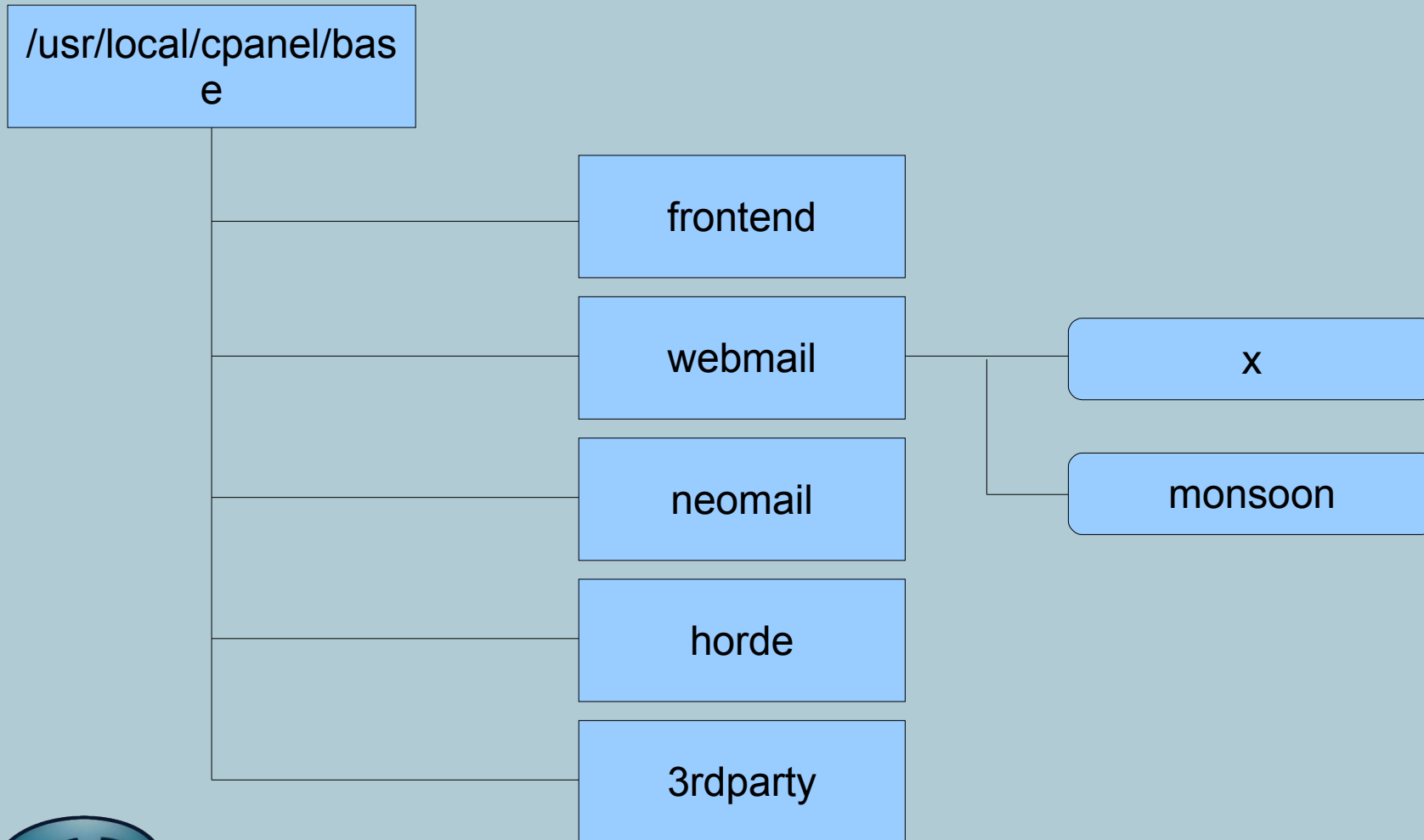
cPanel

Training Seminar 2006

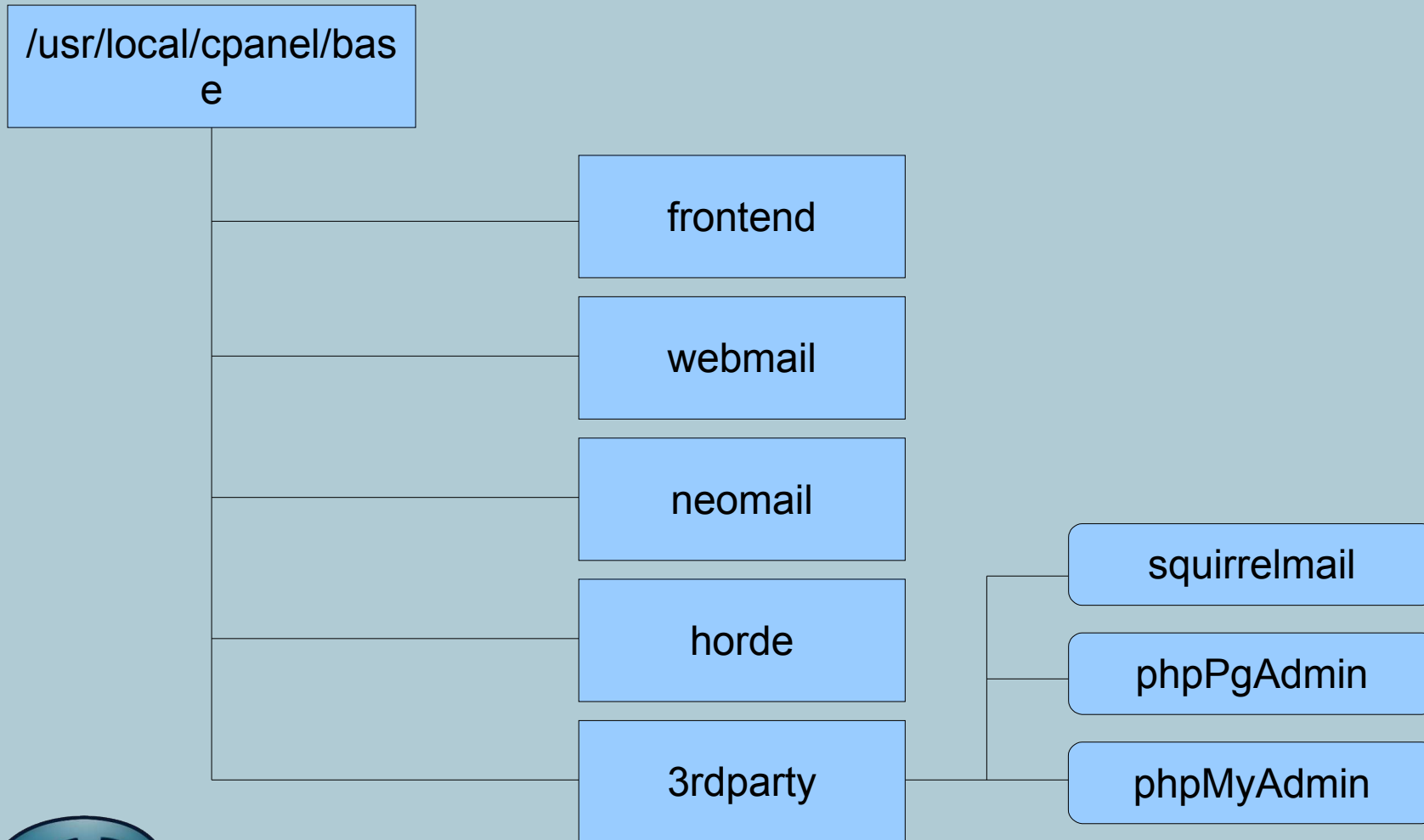
/usr/local/cpanel/base/frontend



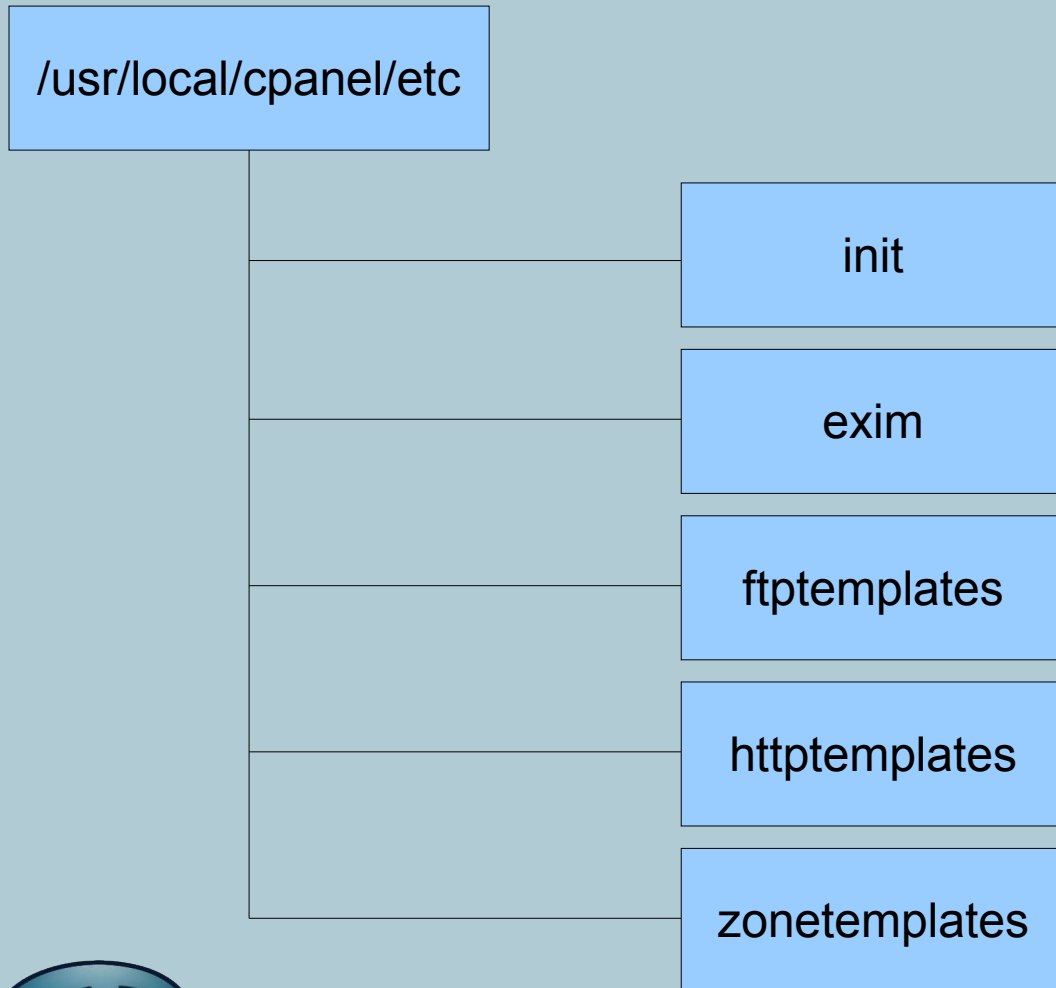
/usr/local/cpanel/base/webmail



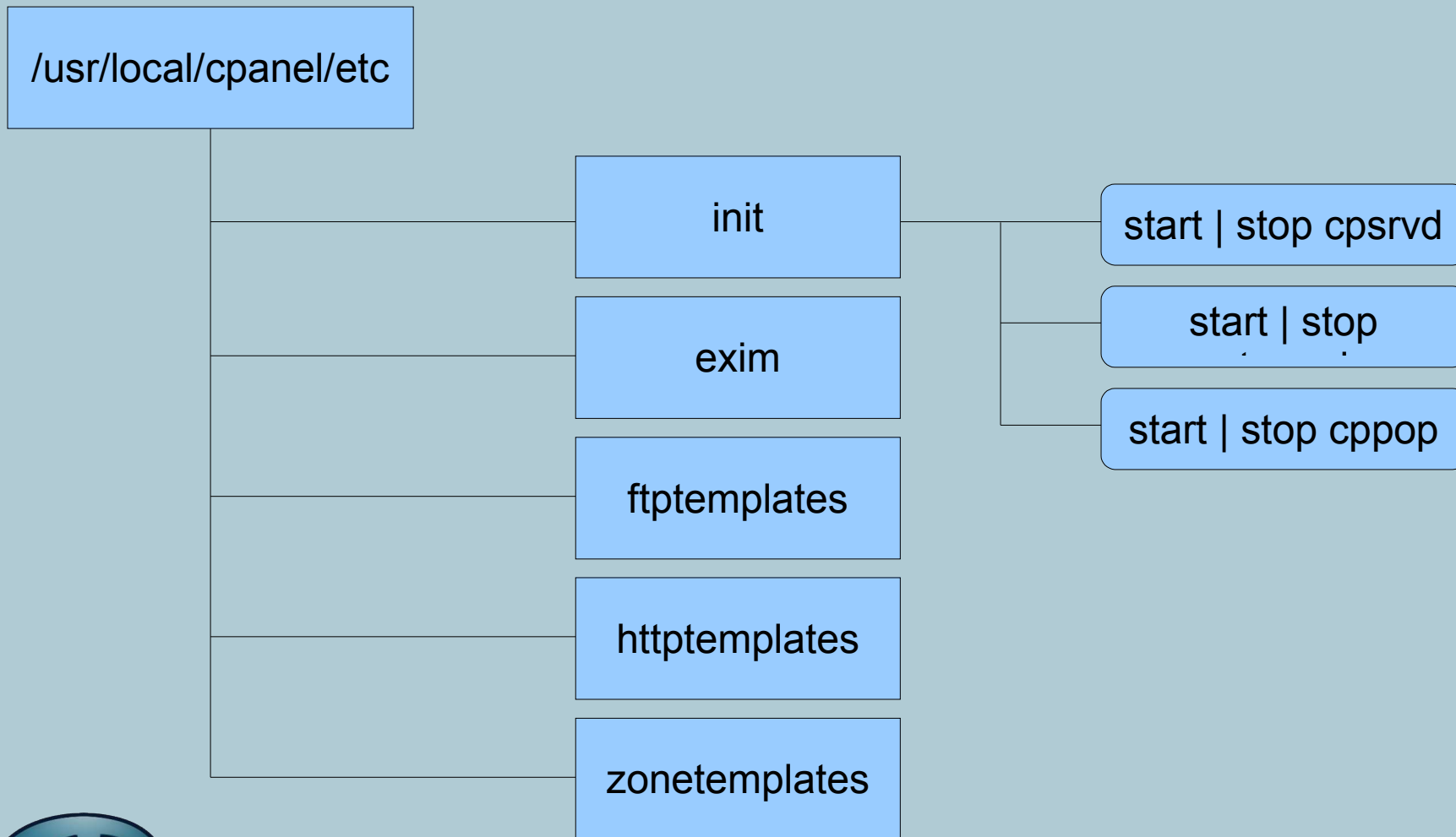
/usr/local/cpanel/base/3rdparty



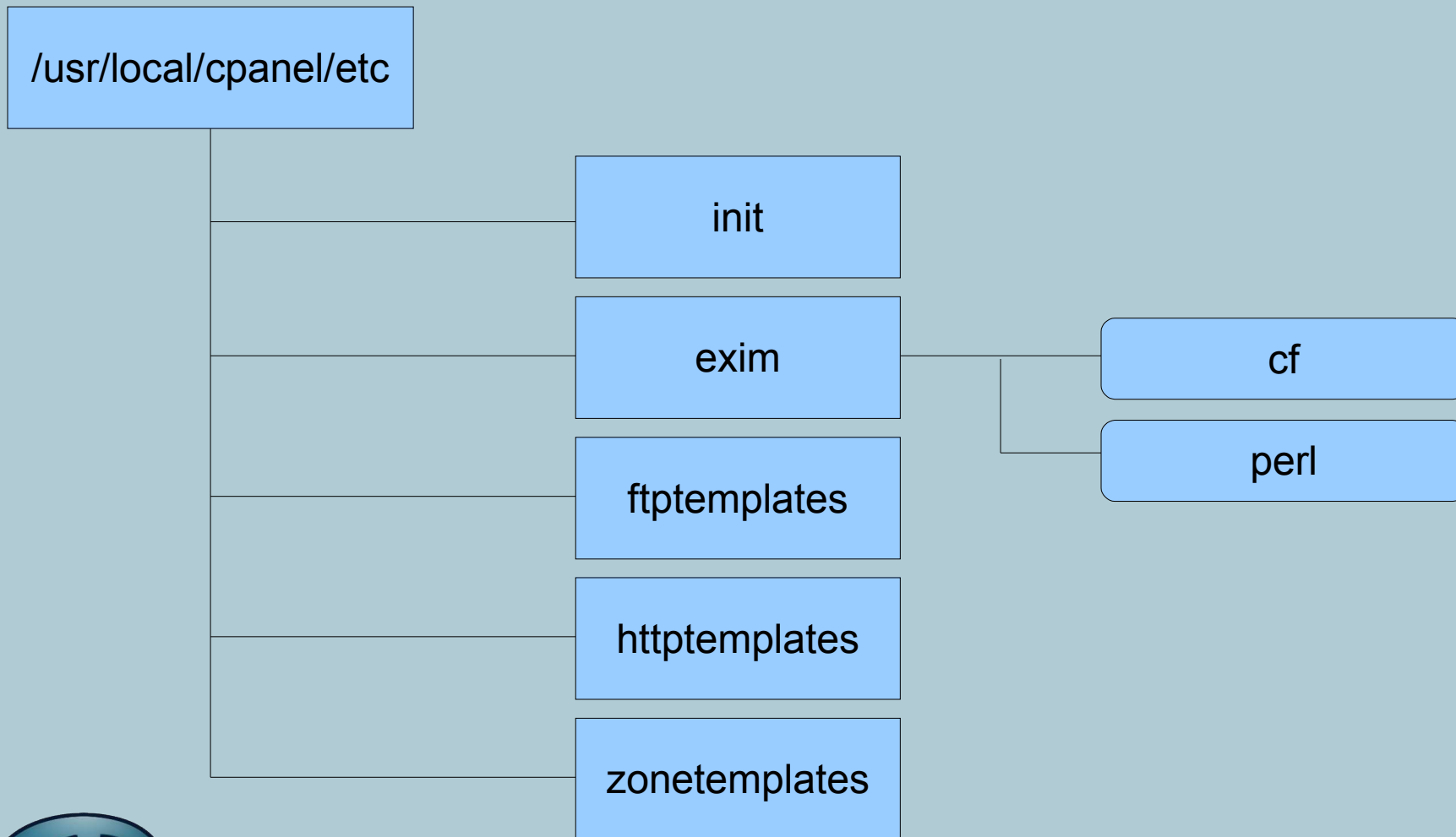
/usr/local/cpanel/etc



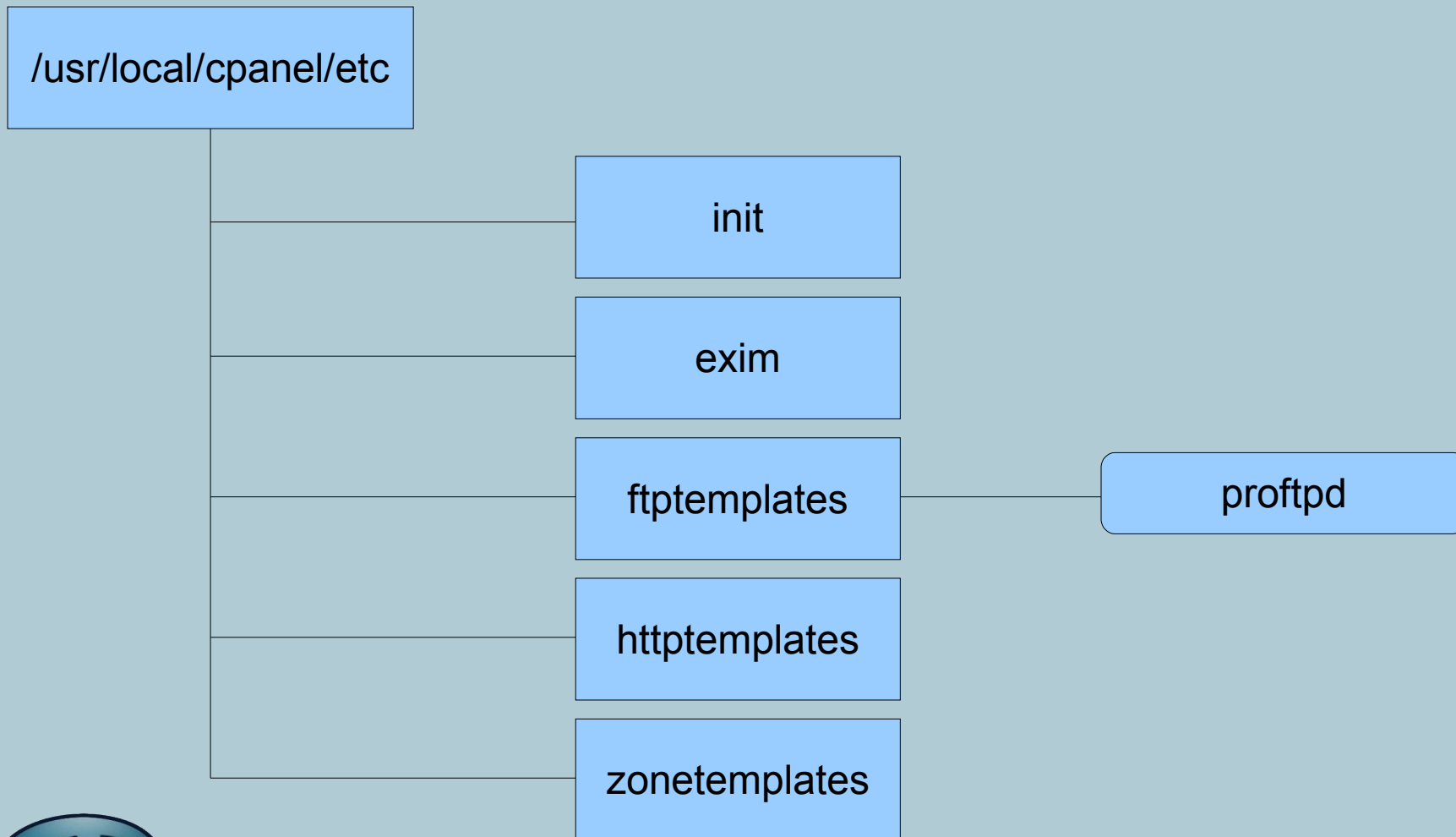
/usr/local/cpanel/etc/init



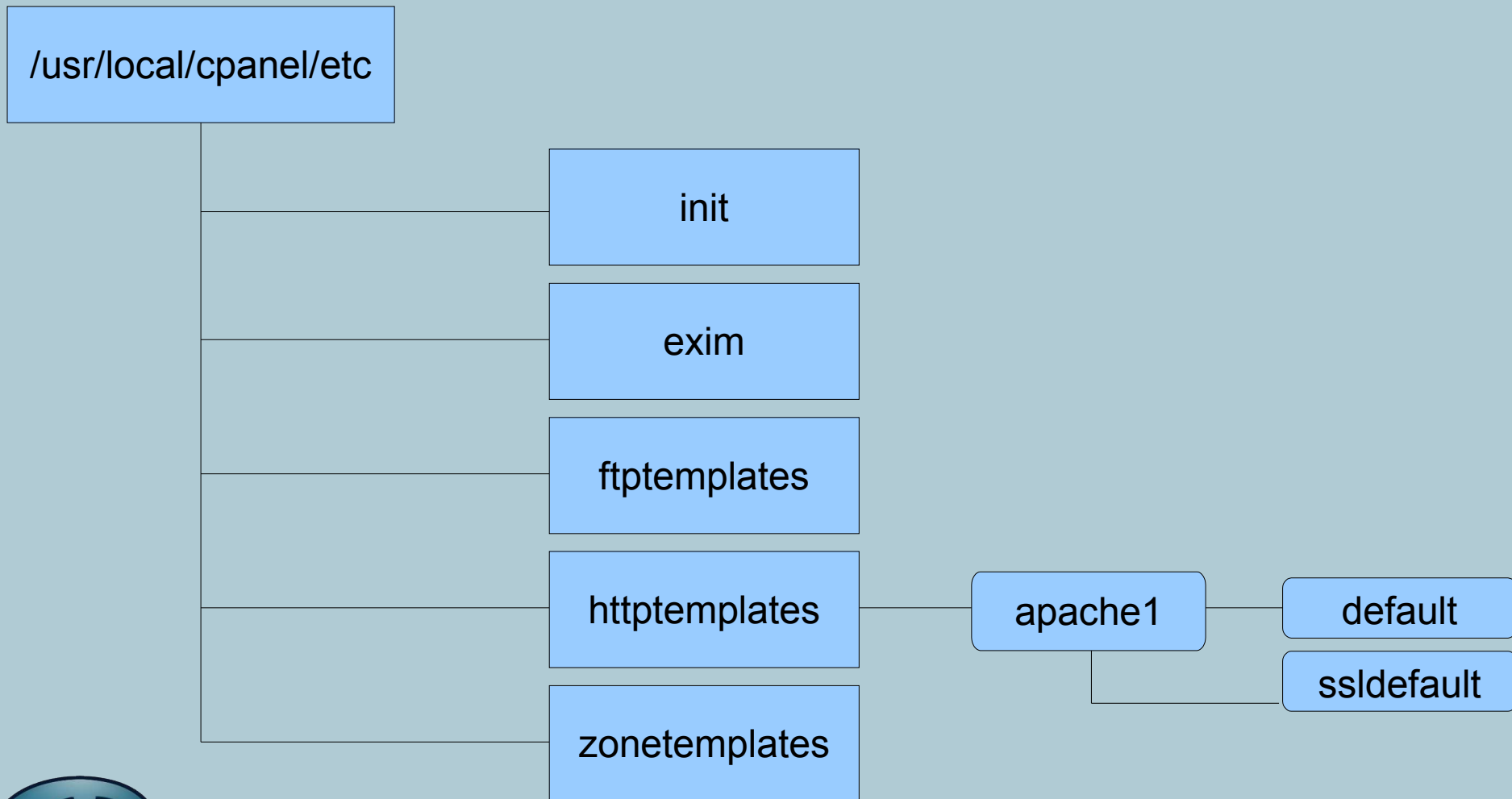
/usr/local/cpanel/etc/exim



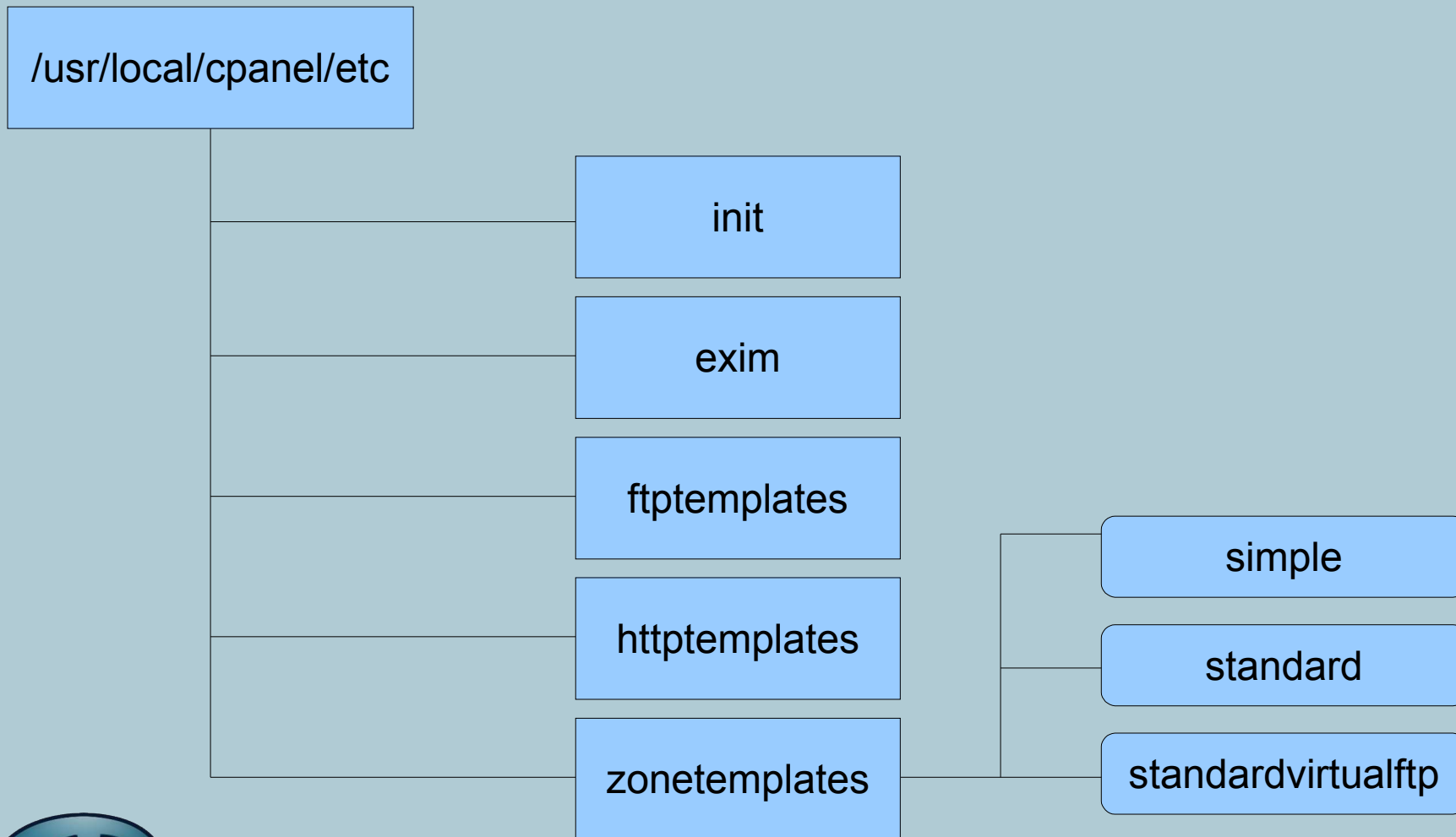
/usr/local/cpanel/etc/ftptemplates



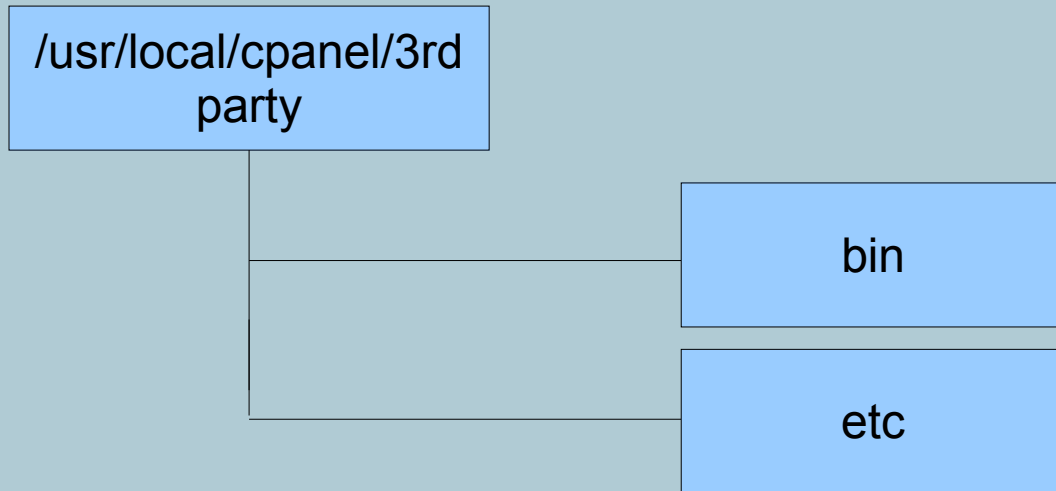
/usr/local/cpanel/etc/httptemplates



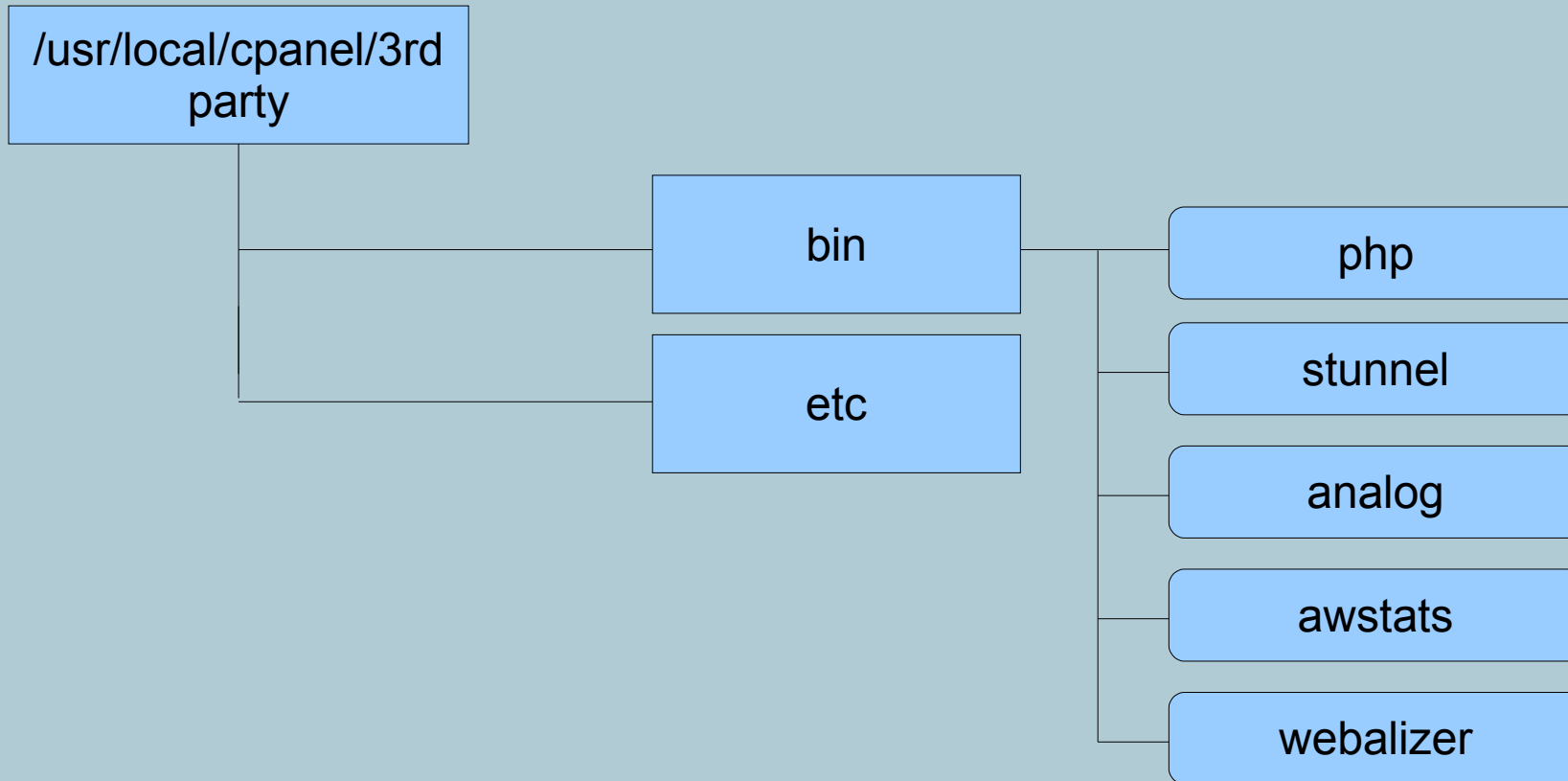
/usr/local/cpanel/etc/zonetemplates



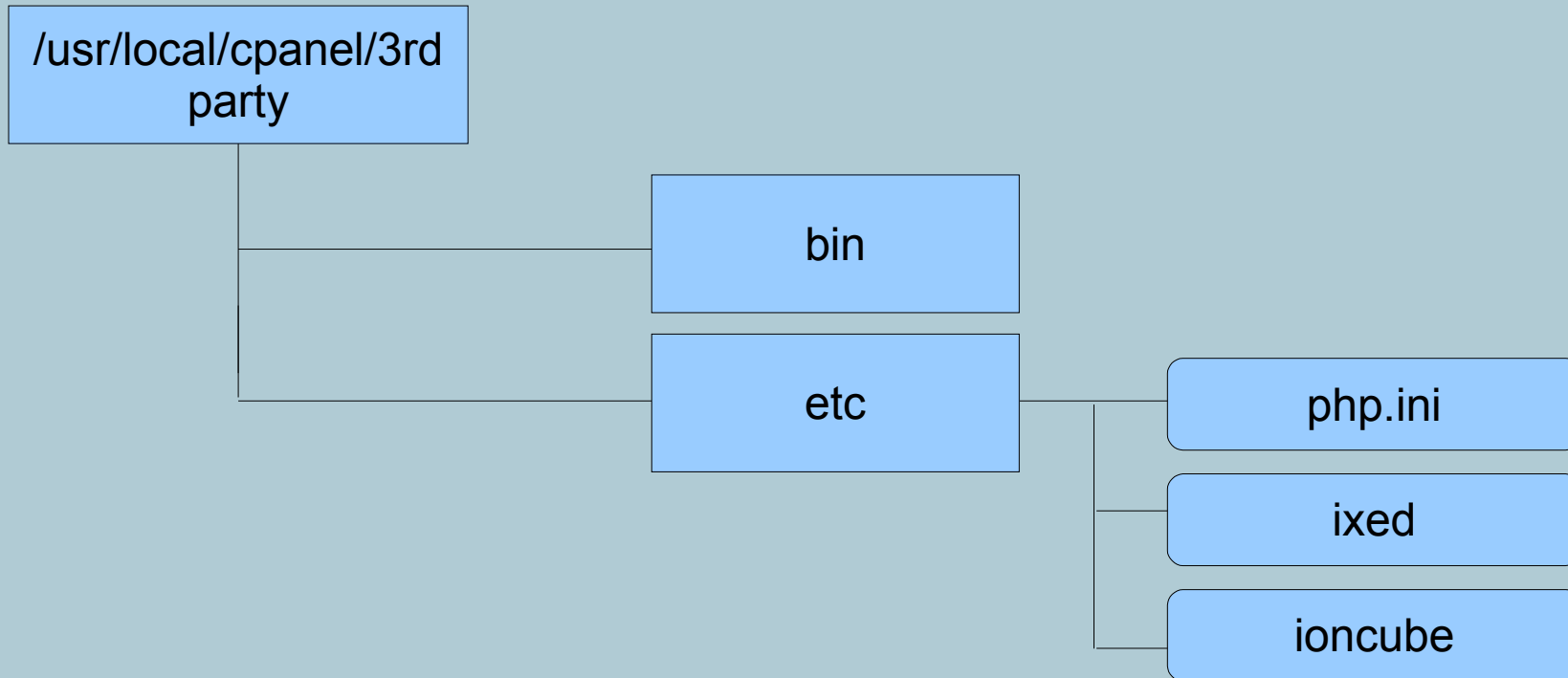
/usr/local/cpanel/3rdparty



/usr/local/cpanel/3rdparty



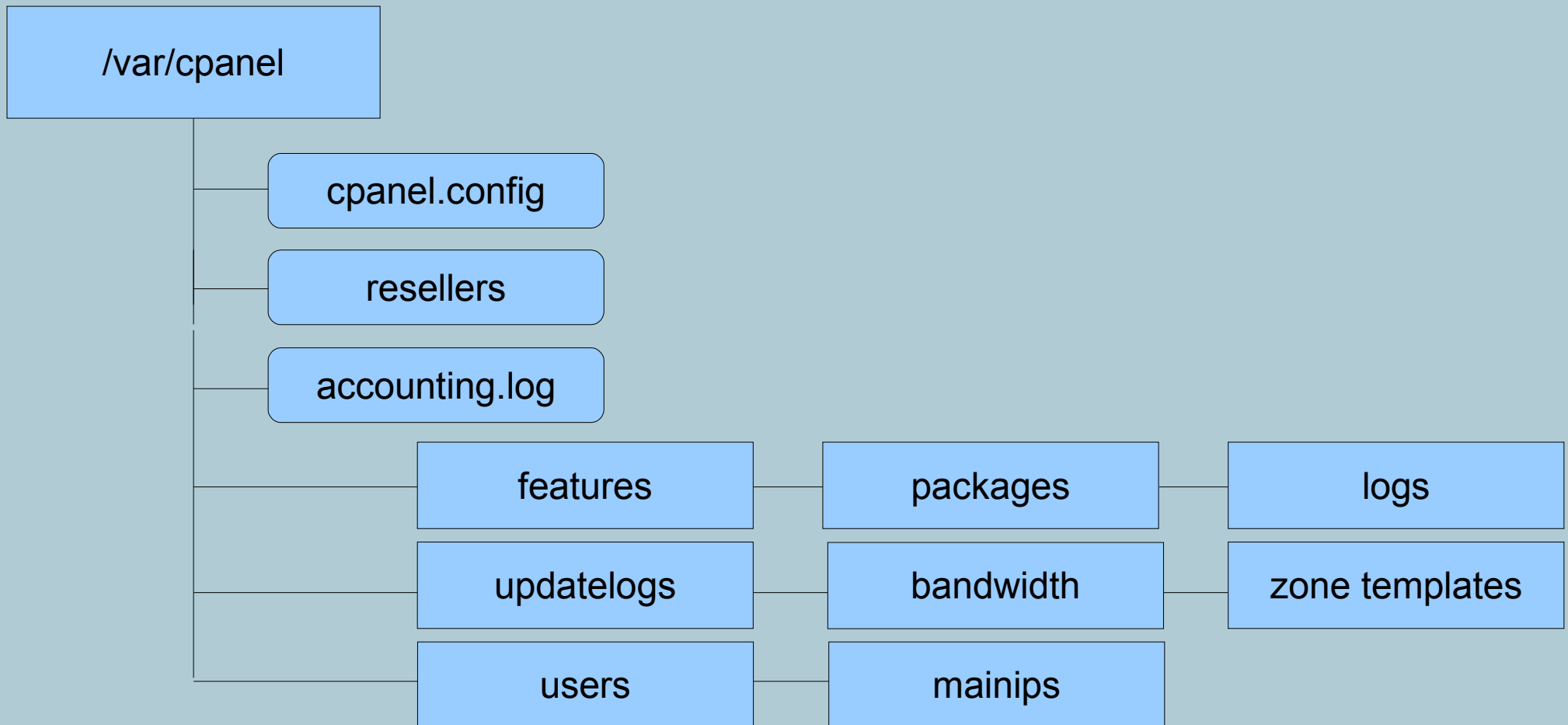
/usr/local/cpanel/3rdparty



/var/cpanel

- Houses proprietary configuration data for cPanel, including:
 - Primary cPanel configuration
 - User configurations
 - Reseller configurations
 - Accounting, conversion, and update logs
 - Bandwidth data
 - Customized service templates

/var/cpanel



/var/cpanel/cpanel.config

- The primary cPanel configuration file
- Each variable within influences the way cPanel behaves
- Variables are line delimited, with variables separated by an equal sign
- If file does not exist, cpanel falls back to defaults

Example:

```
dumplogs=1      # Dump logs after processing
keeplogs=0      # Keep a copy of access logs at the end of each month
skipwebalizer=1 # skip parsing webalizer statistics
extracpus=5     # The load limit of which backup and log processing services can
                # operate
maildir=1       # Server is using a maildir configuration
maxmem=128      # The maximum amount of memory a cPanel process may use
                # before it's killed
mysql-version=5.0 # The MySQL version to be installed on the sever
phploader=ioncube # The decoder to be used by the internal PHP interpreter
```

/var/cpanel/resellers

Lists each reseller with a comma-delimited list of WHM resources that reseller has access to.

Example of a reseller entry:

```
[resellername]:resources,reseller,has,access,to
```

```
example_reseller:add-pkg-ip,add-pkg-shell,all,allow-addoncreate,allow-  
parkedcreate,allow-unlimited-disk-pkgs,allow-unlimited-pkgs,create-acct,create-  
dns,edit-account,edit-dns,edit-mx,frontpage,kill-acct,kill-dns,limit-bandwidth,list-  
accts,mod-subdomains,park-dns,passwd,quota,rearrange-accts,res-cart,show-  
bandwidth,ssl,ssl-gencrt,stats,status,suspend-acct,upgrade-account,
```

/var/cpanel/accounting.log

Contains a list of accounting functions performed through WHM, including account removal and creation.

Example of an accounting.log entry:

```
Sat Nov 19 15:10:22 2005:CREATE:root:root:cpanel.net:198.66.78.235:cpnet
Sat Nov 19 15:17:04
2005:CREATE:root:root:testdomain.com:198.66.78.235:testdom
1133120763:REMOVE::root:cpanel.net:cpnet
1133381280:REMOVE:root:root:testdomain.com:testdom
```

/var/cpanel/bandwidth

- Files contain a list of the bandwidth history for each account. Each named after their respective user.
- History files are stored in human-readable format, while actual bandwidth data are stored in round robin databases.

Example of a bandwidth history file:

```
1.1.2006-all=455  
1.1.2006-http=455  
1.11.2006-all=1852  
1.11.2006-smtp=1852  
1.12.2006-all=3935  
1.12.2006-http=309  
1.12.2006-smtp=3626  
1.14.2006-all=27182
```

/var/cpanel/features

- File name is inherited from the feature list name
- Contains a line delimited list of feature variables and a zero or one value
- Variables control what cPanel resources are available to users

Example of a feature list file:

```
diskusageviewer=1  
indexmanager=1  
sslmanager=1  
emaildomainfwd=1  
cpanelpro_support=1  
cpanelpro_leechprote  
ct=0
```

/var/cpanel/packages

- Contains a list of packages, named after the packages they represent
- If package belongs to reseller, file name is prefixed with reseller name
- Each of these values determines the values created in cPanel user file

Example of a package file:

```
FEATURELIST=default # Feature list to be assigned to account  
It # Quota (in megabytes) to be assigned to account  
QUOTA=100 # CGI should be enabled for this account  
CGI=y # Frontpage extensions should be enabled for this account  
FRONTPAGE=y # The cPanel theme to use for this account  
CPMOD=x # Maximum subdomains the account can create  
MAXSUB=unlimited # Account bandwidth limit  
BWLIMIT=unlimited # Account has shell access  
HASSHELL=y
```

/var/cpanel/users

- Contains a list of cPanel user configuration files, named after the user they pertain to.
- Variables define account resources, themes, domains, etc.

Example of a package file:

```
DNS=exampledomain.com    # The primary DNS entry for this site, the account domain name.  
FEATURELIST=default      # The name of the feature list this account has been assigned  
IP=198.66.78.235         # The IP address assigned to this site  
LANG=english             # The language the assigned cPanel theme should use  
MAXADDON=1000            # Allotted one thousand add-on domains  
OWNER=reseller_example   # Owned by the 'reseller_example' user  
PLAN=unlimited_resources  # Account is assigned the 'unlimited_resources' package  
RS=x                     # Account uses the cPanel x theme
```


Other notable /var/cpanel directories

- **LOGS**
 - This directory contains logs from account copies/transfers.
- **UPDATELOGS**
 - Contains the output of each cPanel update executed on the server.
- **MAINIPS**
 - Named after the respective reseller users they represent, each contains only the IP address which should be used as that resellers main shared ip
- **ZONETEMPLATES**
 - Contains customized DNS zone templates created in WHM

/scripts

This directory houses a large number of scripts which serve as building blocks for many cPanel/WHM features.

The scripts can be used to:

- Update cPanel, and many of the services of which it manages
- Customize account creation routines
- Perform backups of cPanel accounts
- Install and update cPanel managed services

cPanel Services

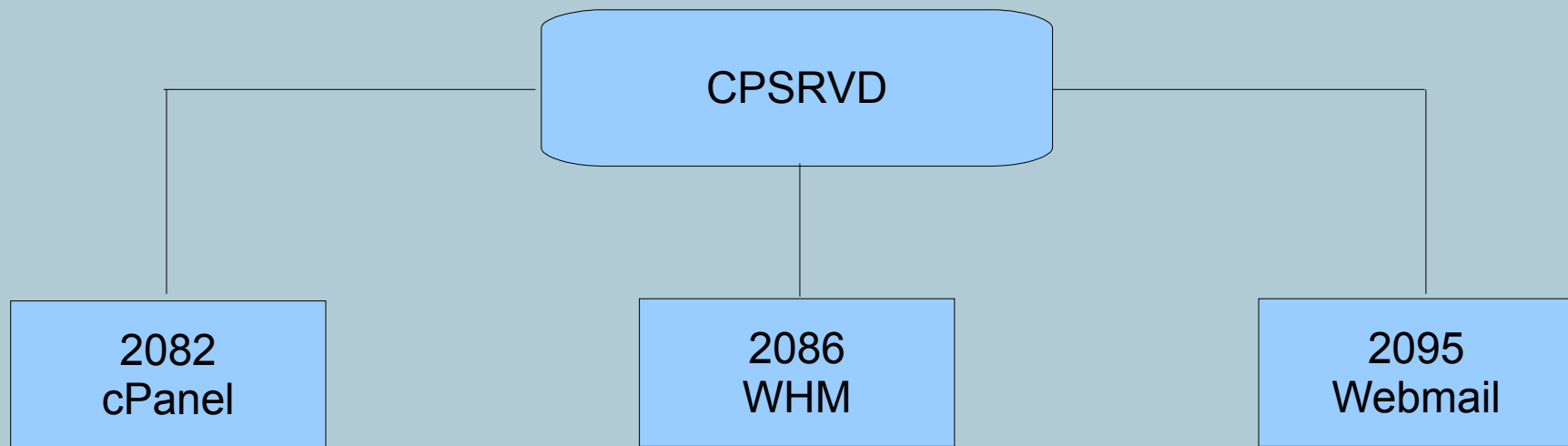
Services

- **CPSRVD**
- CHKSERVD
- CPANELLOGD
- CPBACKUP
- EXIMSTATS

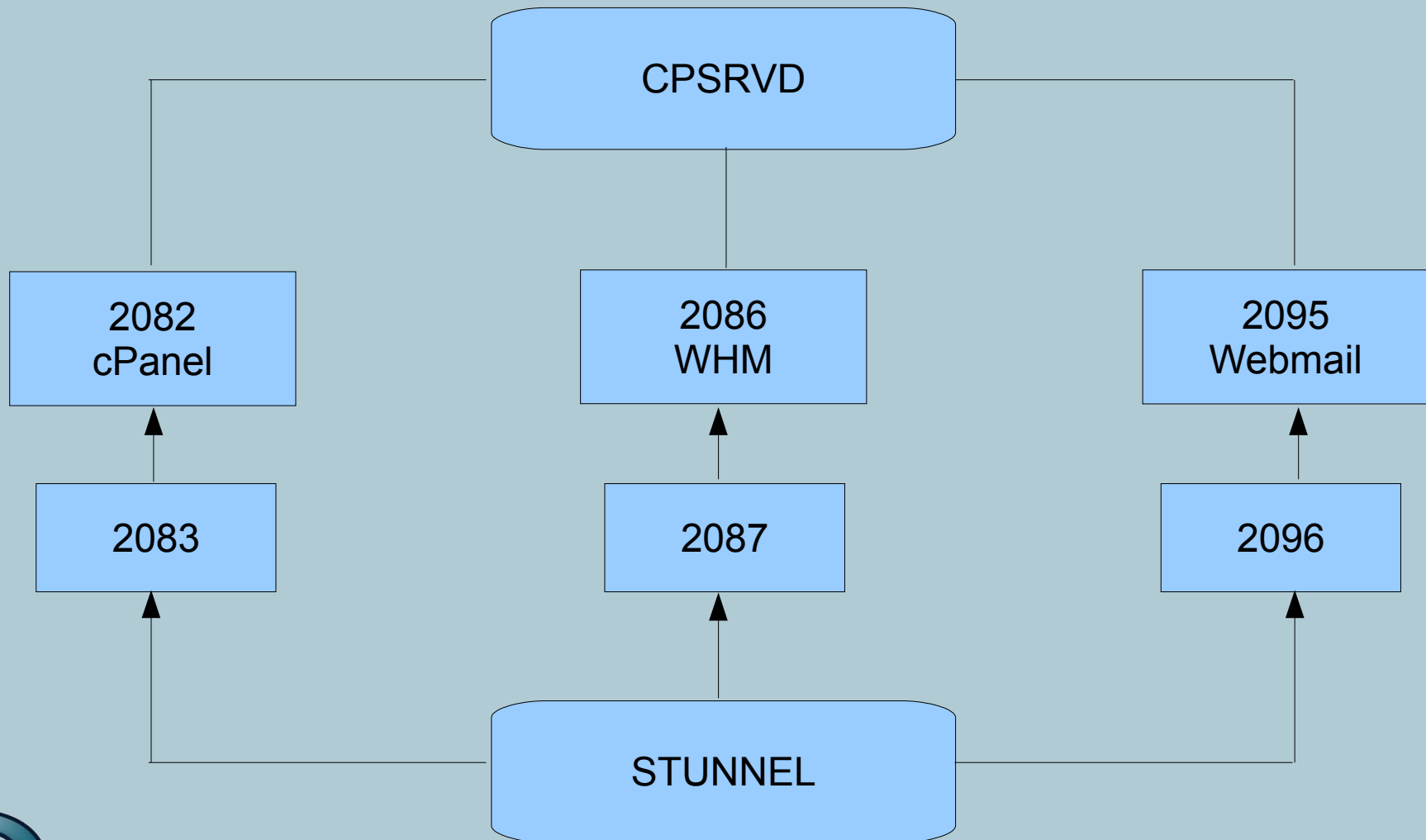
cpsrvd

- cpsrvd is the 'master' process for cPanel.
- Handles and dispatches all requests made through the cPanel, WHM, and Webmail interfaces.
- Logs to access_log and error_log

cpsrvd



cpsrvd and stunnel relationship



SSL Certificates

- Default certificate and key are stored in /usr/local/cpanel/etc/cpanel.pem
- User installed cert and cabundle are stored in:
 - /usr/local/cpanel/etc/mycpanel.pem
 - /usr/local/cpanel/etc/mycpanel.cabundle

cPanel Startup

- The following services are controlled by the cPanel init script
 - cpsrvd, both plain and secure
 - cPanel POP Services
 - cPanel Log Services
 - Eximstats
 - Chat Services
 - Mailman
 - Interchange

Troubleshooting Startup Issues

- Verify if ports are in use
 - `netstat -lnp | egrep '20(8|9)'`

```
# netstat -lnp | egrep '20(8|9)'
```

tcp	0	0	0.0.0.0:2082	0.0.0.0:*	LISTEN	10185/cpsrvd - wait
tcp	0	0	0.0.0.0:2083	0.0.0.0:*	LISTEN	325/stunnel-4.15loc
tcp	0	0	0.0.0.0:2086	0.0.0.0:*	LISTEN	10185/cpsrvd - wait
tcp	0	0	0.0.0.0:2087	0.0.0.0:*	LISTEN	325/stunnel-4.15loc
tcp	0	0	0.0.0.0:2095	0.0.0.0:*	LISTEN	10185/cpsrvd - wait
tcp	0	0	0.0.0.0:2096	0.0.0.0:*	LISTEN	325/stunnel-4.15loc
#						

Troubleshooting Startup Issues (SSL)

- If SSL services are not available
 - execute `/usr/local/cpanel/startstunnel`
 - check `/usr/local/cpanel/3rdparty/bin/stunnel.log`
- If `cpsrvd` is not available
 - execute it directly ``/usr/local/cpanel/cpsrvd``
 - check `/usr/local/cpanel/logs/error_log`

Licensing

- License requests are handled by `/usr/local/cpanel/cpkeyclt`
- Requests are transmitted to `auth.cpanel.net` over port 2089
- License requests are logged to `license_log`
- License key is stored at `/usr/local/cpanel/cpanel.lisc`

A valid license request:

```
root@server [~]# /  
usr/local/cpanel/cpkeyclt  
Updating Internal cPanel  
Information.....Done  
root@server [~]#
```

Troubleshooting License Issues

CHECKLIST:

- Verify if license is active for main server IP at <http://verify.cpanel.net>
- Check if server can establish connection to auth.cpanel.net over port 2089
- If the previous steps fail, check `license_log` for notable errors.
- If license is active, but refused with no notable errors, lodge support request.

```
root@server [~]# telnet auth.cpanel.net 2089
Trying 198.66.78.9...
Connected to auth.cpanel.net (198.66.78.9).
Escape character is '^]'.
200 cPanel License Service Version 12.0
root@server [~]#
```

cPanel Requests

cPanel Requests

- Logins are authenticated against the system passwd and shadow files.
- Documents root is /usr/local/cpanel/base
- Theme is defined by RS variable in user's cPanel configuration file.
- Resources are limited by the feature list of assigned to the given user.

WHM Requests

WHM Requests

- Root password will authenticate any reseller user
- Document root is /usr/local/cpanel/whostmgr/docroot/
- Reseller resources are limited by Access Control List
 - Defined in WHM > Resellers > Reseller Center > Edit Privileges/Nameservers
 - Privileges are stored in /var/cpanel/resellers

cPanel Services

Services

- CPRVD
- **CHKSERVD**
- CPANELLOGD
- CPBACKUP
- EXIMSTATS

Service Monitoring

- Located at `/usr/local/cpanel/libexec/chkservd`
- `chkservd` is a scalable connection and process based service monitoring tool
- Provides monitoring of CPU, Memory, and Disk usage
- `chkservd` scans services once every eight minutes
 - Logs to `/var/log/chkservd.log`
- Alerts are dispatched to server contact defined in Basic cPanel/WHM Setup

chkserverd Configuration

- Monitored services are determined by values stored in /etc/chkserv.d/chkservd.conf.
 - Syntax: servicename:0 for no monitoring, servicename:1 for monitoring
- Actions, expected responses, and failure events are defined in service configuration files stored in /etc/chkserv.d/{servicename}
- Status files are stored in /var/run/chkservd/{servicename}
 - Plus (+) sign for active, Minus (-) sign for failed

Anatomy of a chkservd Service File

`service[{servicename}]=port,send,response,restart,command,owner`

- The restart field can be any number of shell commands, separated by semi-colons
 - Tip: use to determine state of service/system prior to service restart

Connection Based Monitoring

Required Arguments:

`service[{servicename}]=port,send,response,restart`

`service[pop]=110,QUIT,.OK,/scripts/restartsrv_cppop`

CONNECTION BASED OPTIONS

- If no command field is defined, service scan is assumed as connection based
- Expected response can be any valid perl regular expression
- Will connect over loopback interface (127.0.0.1). If service is not accessible via loopback, chkservd will report the service as failed!

Command Based Monitoring

Required Definitions

`service[{servicename}]=,,,restart,command,owner`

`service[mysql]=x,x,x,/scripts/restartsrv_mysql,mysql,mysql`

COMMAND BASED OPTIONS

- Multiple users can be specified with a pipe delimited list (user1|user2)
- Command name can be any valid regular expression
note: This expression will be matched against all processes owned by that user. Be specific!

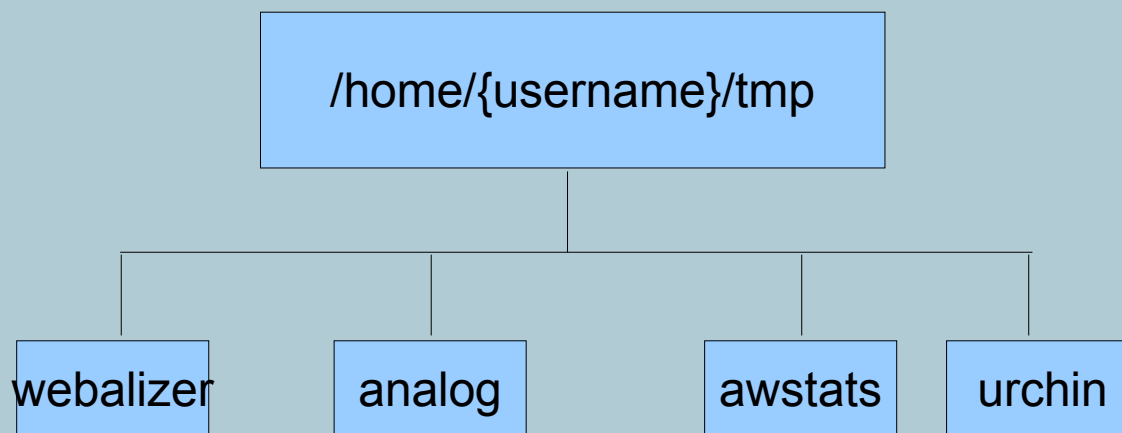
cPanel Services

Services

- CPSRVD
- CHKSERVD
- **CPANELLOGD**
- CPBACKUP
- EXIMSTATS

cpanellogd

- cpanellogd is responsible for parsing and updating bandwidth logs, and dispatching statistics generators on each account, per their individual configurations.
- Configured through Statistics Software Configuration and Tweak Settings in WHM
- Statistics are compiled and stored for each account in `/home/{username}/tmp`, with each respective statistics application being assigned it's own individual subdirectory.



cpanellogd (Cont.)

- Optional server-wide statistics configurations are stored in /etc/stats.conf, while user-specific configurations may reside in /home/{username}/tmp.
- **Notable Variables in /etc/stats.conf:**
 - BLACKHOURS: Comma separated list of numeric values, which specify hours that logs may not be parsed.
 - VALIDUSERS: Users which are allowed to supply their own combination of statistics generators. By default users are restricted to the generators defined by the administrator.

Calling cpanellogd

- cpanellogd is started with the cPanel service, but can be executed directly with:
 - No Argument: Daemonize, and wait for a suitable time to scan logs
 - One Argument (username): Execute an immediate statistics run for the specified user, and exit once completed.
- Two scripts are available to provide these functions as well:
 - /scripts/runlogsnow - Execute a full log run immediately
 - /scripts/runweblogs {username} - Execute a log run for a single user

Bandwidth Statistics

- Bandwidth statistics are accumulated from a combination of the following cPanel managed services:
 - HTTP
 - EXIM
 - IMAP / POP
 - FTP
- Bandwidth data is logged to `/usr/local/apache/domlogs/*bytes_log`
- Parsed bandwidth data is stored in `/var/cpanel/bandwidth`

Bandwidth Statistics (Cont.)

COMMON ISSUES

- Bandwidth parsing is taking an exceedingly long time to complete
 - First check the size of the logs being parsed. Excessively large log files can and typically will take a long time to complete.
 - Additionally, if RRDtool is not installed, bandwidth parsing performance will drop significantly.
 - RRDtool can be installed by executing ``/scripts/rrdtoolinstall``

Log Processing

- Statistics are parsed for each child domain of the given account.
- Will be influenced by variables in `/var/cpanel/cpanel.config`
 - Skip statistics generator
 - `skip{generator_name}`
 - Logs will be retained or deleted based on
 - `keeplogs` – keep logs at the end of the month.
 - `dumplogs` – dump logs after parsing

Common cpanellogd Issues

- **Statistics are stalling, or are taking unreasonable amounts of time.**
 - Usually indicates that the server load average is consistently exceeding the defined load limit.
 - Limit is defined as 'extracpus' in /var/cpanel/cpanel.config
 - Restrictive BLACKHOUR definitions in WHM > Statistics Software Configuration.
 - All other issues should be present in /usr/local/cpanel/logs/stats_log

cPanel Services

Services

- CPSRVD
- CHKSERVD
- CPANELLOGD
- **CPBACKUP**
- EXIMSTATS

cPanel Backups

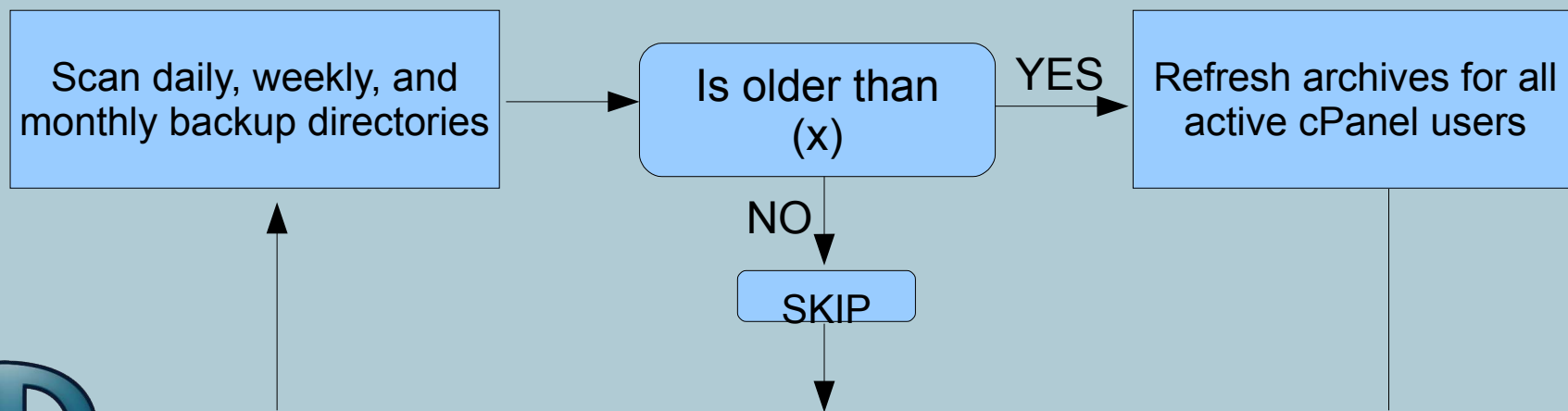
GENERAL INFORMATION

- Backup configuration is performed in WHM > Backup > Configure Backup
- cPanel backups are performed by /scripts/cpbackup, which is configured by default to execute at 1:00 AM in the root crontab.
- Backup archives are created using the /scripts/pkgacct utility, and may be restored using /scripts/restorepkg respectfully.
- Uses CPU resource limits based upon extracpus definition in cpanel.config

Backup Configuration

BACKUP INTERVALS

- Backup script can be configured to operate in daily, weekly, and monthly intervals.
- Each interval is given it's own respective directory within the backup root.
- Backup intervals are executed when the current time minus the last modification time of the interval directory is less than or equal to zero.



Backup Configuration (Cont.)

BACKUP METHODS

Three backup methods are available:

- Standard: This method entails archiving the accounts, and storing them at the specified path/mount point. This is the default method used by the backup script.
- Incremental: This method uses rsync to incrementally backup user data. This option will only operate locally, storing the data at the specified path/mount point.
- Remote: This method transmits account archives to a specified ftp server. Remote backups are typically more time consuming, and more error prone when transmitting large accounts.

Common Backup Issues

- **Backup intervals are not executed when expected.**
 - Modification times are incorrect or not functional
 - System time is incorrect.
 - Backups have not been defined to run on that day.
- **Backups stall, or take an exceedingly long time to complete.**
 - Verify that the transmission rate to remote server is suitable
 - Verify that server load average has not exceeded defined resource limit.

Common Backup Issues (Cont.)

- **Can't call method "login" on an undefined value**
This indicates the host or passive setting is not properly defined for remote backups.
- **Unable to login to remote FTP server.**
This indicates that either the username and password were not specified, or are incorrect in the backup configuration.
- **Can't call method "prepare" on an undefined value**
The password stored for the root mysql user in /root/.my.cnf is incorrect. Reset or correct this password, and re-execute the backup script.

cPanel Services

Services

- CPSRVD
- CHKSERVD
- CPANELLOGD
- CPBACKUP
- **EXIMSTATS**

eximstats

- The eximstats daemon is responsible for harvesting bandwidth information from exim transactions.
- Continually monitors the exim_mainlog, and stores information in the eximstats database, including host and sender information, message size, and transaction times.
- Is started with the cPanel service, but can be called directly at /usr/local/cpanel/bin/eximstats

eximstats (Cont.)

- Heavily mysql dependent
 - data is stored in the 'eximstats' database.
- 'eximstats' mysql user password is stored in /var/cpanel/eximstatspass.
 - password is generated by /usr/local/cpanel/bin/eximstatspass
- Database can be installed by running /usr/local/cpanel/bin/updateeximstats

cPanel Maintenance

- Update configuration
- Update scripts
- Applying updates

cPanel Maintenance

- By default, cPanel applies nightly updates at 2:13AM in the root crontab.
- /scripts/upcp dispatches these updates, using the following key components:
 - /scripts/updatesnow - synchronize /scripts directory
 - /scripts/sysup - updates cPanel managed rpms
 - /scripts/rpmup - all other system updates
- Updates are logged to timestamped files in /var/cpanel/updateslogs
- Update configuration is stored in /etc/cpupdate.conf.

/etc/cpupdate.conf

- The following variables are available in cpupdate.conf:
 - **CPANEL = [manual-] stable | release | current | edge**
This variable controls which update branch is used for cPanel updates, and controls whether the updates are applied manually or automatically (Default value: release)
 - **SYSUP = never** (all other values are assumed true)
 - **RPMUP = never** (all other values are assumed true)

```
CPANEL=current  
RPMUP=daily  
SYSUP=daily
```


Running cPanel Updates

- cPanel updates can be called outside of the regularly scheduled cron time simply by executing `/scripts/upcp`.
- If cPanel components are missing or corrupted that were not replaced with the regular cPanel update, they can be replaced by executing `/scripts/upcp -force`

Components of upcp

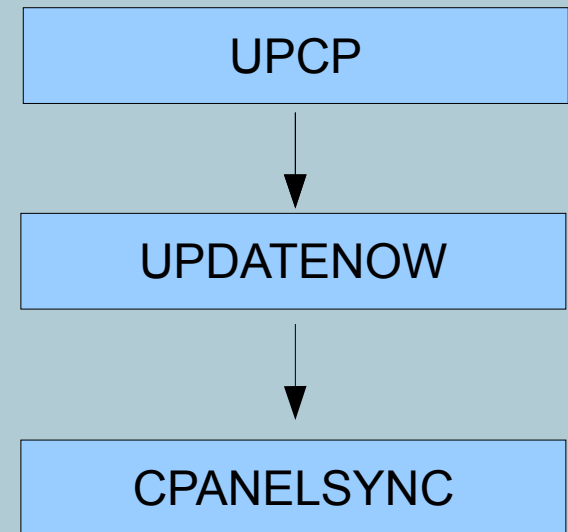
- /scripts/cpanelsync
- /scripts/updatenow
- /scripts/sysup
- /scripts/rpmup

cpanelsync

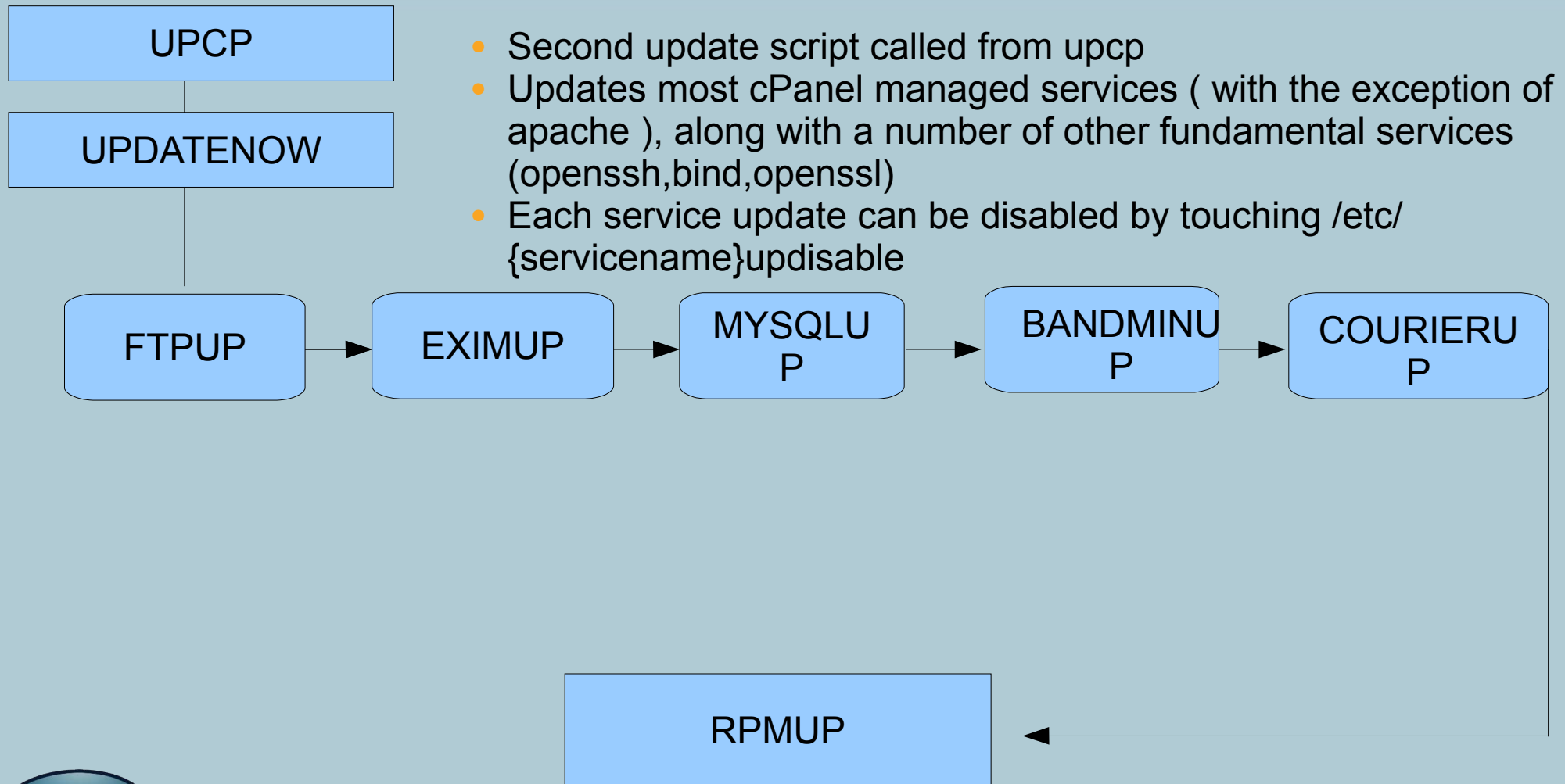
- /scripts/cpanelsync is called upon by /scripts/uptdatenow and /scripts/upcp
- Provides md5sum based synchronization with update servers
- md5sum table is stored in /destination_directory/.cpanelsync
- Accepts three arguments host, remote path, local path :
 /scripts/cpanelsync 'httpupdate.cpanel.net'
 '/cpanelsync/RELEASE/scripts' '/scripts'

updatenow

- Calls cpanelsync to update contents of scripts directory, which then stores it's md5sum table at /scripts/.cpanelsync
- Should only be run from upcp, but can be executed from command line when '--fromupcp' is passed.
- Is the first update script called upon from /scripts/upcp



cPanel Update Scripts (sysup)



rpmup

- Calls the underlying package manager to apply system package updates
- The package manager which is used is determined by the presence of:
 - /var/cpanel/useup2date (Redhat)
 - /var/cpanel/useyum (CentOS,Fedora)
 - /var/cpanel/useapt (Debian)
 - /var/cpanel/useswup (Trustix)
 - /var/cpanel/userug (SuSE)

cPanel Updates

- After updatenow, sysup, and rpmup complete, cpanelsync is used to complete the cPanel updates based on md5sum table stored at /usr/local/cpanel/.cpanelsync
- If any special configurations are required on server after updates, they can be applied in /scripts/postupcp, which is executed if such a file exists and is executable.
- Once updates complete, all cPanel services are restarted for changes to take effect

cPanel Scripts

- Account Management
- Package Management
- Service Update and Configuration
 - MySQL
 - Exim
 - Named
 - Apache
- cPanel and System

Account Management Scripts

- **/scripts/wwwacct (account creation)**

Accounts can be created via the command line using the following syntax: `/scripts/wwwacct exampledomain.com username password 0`
x n

- **/scripts/killacct (account termination)**

Takes a single argument of the user to terminate.

- **/scripts/suspendacct (account suspension)**

Will suspend an account from accessing all cPanel managed services.

- **/scripts/unsuspendacct**

Will reinstate any account suspended via `suspendacct`

Account Management (Cont.)

Useful Scripts – Account Management (Cont.)

- **/scripts/addpop (Create pop account)**
Handles creation of virtual mail accounts. Accepts either no arguments, or two arguments consisting of the e-mail address and password.
- **/scripts/updateuserdomains**
Updates the user:owner and user:domain tables stored in:
 - /etc/userdomains
 - /etc/trueuserdomains
 - /etc/trueuserowners
 - These tables are used to enumerate and keep track of accounts and their owners.

Package Management

- **/scripts/ensurerpm**
Takes argument list of rpms, which are then passed to the underlying package manager
- **/scripts/ensurepkg**
The equivalent of ensurerpm for FreeBSD. Updates specified packages from ports.
- **/scripts/realperlinstaller**
Takes argument list of perl modules to install via CPAN
- Each of the aforementioned scripts can accept an argument of '--force' to force package installations.



MySQL Update and Configuration

- **/scripts/mysqlup**
Can be called to apply MySQL updates independent of upcp
- **/scripts/cleanupmysqlprivs**
Will clean up the default MySQL privilege tables, by installing a more restrictive privilege schema.
- **/scripts/mysqlconnectioncheck**
Will verify that mysql is accessible with password stored in /root/.my.cnf, and force a reset with a random 16 character string if inaccessible.
- **/scripts/restartsrv_mysql**

Exim Update and Configuration

- **/scripts/eximup**
Can be called to apply exim updates independent of upcp
- **/scripts/buildeximconf**
Will rebuild exim.conf, and merge local, distribution, and cPanel configurations
- **/scripts/restartsrv_exim**

BIND Configuration

- **/scripts/rebuildnamedconf**
Rebuild named.conf based on existing zone files
- **/scripts/restartsrv_bind**

Apache Update and Configuration

- **/scripts/easyapache**
Download, extract, and execute apache build script
- **/scripts/rebuildhttpdconf**
Rebuilds httpd.conf based on DNS entries found in each cPanel user configuration
- **/scripts/restartsrv_httpd**

cPanel Scripts

Useful Scripts – cPanel and System

- **/scripts/restartsrv_{servicename}**
The majority of cPanel managed service can be scripts named appropriately.
- **/scripts/makecpphp**
Will rebuild the PHP interpreter used internally by cpsrzd
- **/usr/local/cpanel/bin/checkperlmodules**
Will scan for and install any Perl modules required by cPanel.
- **/scripts/fullhordereset**
Updates horde and resets the horde mysql user password
- **/scripts/fixquotas**
Will attempt to rebuild quota database per information stored in /etc/quota.conf