Integrated Technical Solutions, Inc.
www.teamITS.com

Linux VPS 3.0

Technical Overview

First Edition
December 2006

Table of Contents

Introduction

This document provides a technical overview of Linux VPS 3.0. It describes how your account provides a private, protected, and fully isolated area. Although this document does not provide the detailed, technical instructions of a user's guide, it does provide descriptive technical information regarding operating system level virtualization and the features of your account. This technical overview also includes descriptions of how resources are managed on your account and how FairShare technology provides your account with a distinct advantage over other solutions. All of this information is placed in the context of the caveats required of any technical document concerning any Internet or Network technology. There are also notices and document conventions which provide you with information which will familiarize you with document sources and how instructions will be presented in customer documentation for your Linux VPS.

Acknowledgments

In addition to internally developed technologies, Linux VPS utilizes open-source technologies and applications which have grown out of development, testing, and revision conducted by open-source communities around the globe. Those communities include (but are not limited to) the following:

- Apache Software Foundation (http://www.apache.org/)
- Fedora Project, sponsored by Red Hat (http://fedora.redhat.com/)
- Free Software Foundation (http://www.fsf.org/)
- GNU Project (http://www.gnu.org/)

Purpose of this Document

The purpose of this document is to provide descriptive, introductory information regarding Linux VPS. It describes, rather than instructs, about the features of Linux VPS. Refer to customer documentation to find out how to use Linux VPS the core services it provides.

Navigating this Document

This document includes the following sections:

Overview of Linux VPS

This section provides you with a quick introduction of and context for Linux and the Red Hat Enterprise Linux (RHEL), the Linux Command Library, and how you might utilize the features.

GNU/Linux is a UNIX-like Operating System

Linux (sometimes referred to as GNU/Linux or a Linux-based GNU system) is a UNIX-like operating system. Linux is distributed under the terms of the GNU General Public License as published by the Free Software Foundation. Your Linux VPS account utilizes RHEL, a widely implemented corporate Linux standard.

Red Hat and Linux VPS

RHEL is based on open standards and is derived from the Red Hat-sponsored and the community-supported, open source Fedora project named. To locate more information about RHEL and the Fedora project, refer to the following Web sites:

- http://www.redhat.com
- http://fedora.redhat.com

The RHEL operating system provides support for GNU Compiler Collection (GCC) and the Red Hat Package Manager (RPM). As you perform configuration, administration and trouble-shooting tasks, apply your previous knowledge of open-source software applications. Your account provides services in a way that assures the account functions as a stand-alone server, independent from any other account. The account supports specific processes, applications, users, and files. Utilize root access and grant access to any ports. The account supports multiple users and provides you with access to all logs. Data backups, server security and software updates are updated through of server software updates which often do not require your intervention. Your account is a hosting environment which provides you with an approximation of your own virtual machine. Keep in mind that although your account shares remote hardware with other accounts, your account does not share software. Each account has its own complete directory structure and set of dedicated applications such as Web server and mail server. Your account can be remotely rebooted without affecting any other accounts served by the physical hardware. Your account is compliant with server monitoring software applications. Configure your account to support multiple users with super user do (*sudo*), shell, Web, File Transfer Protocol (FTP) and/or email privileges. The RHEL operating system provides a compatible base for operating- system level server virtualization, skel package, and copy-on-write optimization.

The Linux Command Library

Your account also supports your access to the Linux Command Library (or *man pages*) which provides information about the full command set supported by your account. Man pages also provide information about system calls, library calls, special files, as well as file formats and conventions.

How to Utilize Linux VPS Features

Following are examples of how to utilize the features of your account:

- Host an e-commerce Web site
- Support a corporate intranet
- Build a custom development environment
- Provide Web-based calendaring
- Provide multimedia applications
- Host an online game site
- Manage an email system
- Create a customer support tracking system
- Backup important data
- Host multiple Web sites

How Linux VPS Provides a Privacy and Protection

Linux VPS provides a private and protected area that operates as an independent server. Linux VPS behaves as if it is, in fact, a stand-alone physical server offering excellent security, performance, and flexibility.

Operating system-level server virtualization creates isolated, secure virtual environments on a single physical server. Server virtualization enables better server utilization and ensures applications do not conflict. Each account performs and executes as a stand-alone server can. Reboot your Linux VPS account independently and have and assign account root access, users, IP addresses, memory, processes, files, applications, system libraries and configuration files.

Your Linux VPS behaves as a stand-alone Linux server. It has standard startup scripts and software from multiple vendors can operate in the account without modification. Change any configuration file and install additional software. The file system, the processes, Interprocess Communication (IPC) mechanisms, and sysct1 variables are always fully isolated from any other account. Processes which belong to your account are scheduled for execution on all available processing power.

Your Linux VPS includes its own IP address (or its own set of IP addresses). The network traffic of your account is isolated from all other accounts. Traffic snooping is not possible. Manipulate your accounts routing table using advanced routing features.

Resource management controls the amount of resources available to your account. This enables the quality of service to meet the service level agreements associated with your account. The operating system-level server virtualization also provides performance and resource isolation which protects your account from denial of service attacks.

The isolated environment of the Linux VPS is established by creating a server sandbox. Each Linux VPS has its own complete directory structure, a virtual file system, its own set of independent applications (Web server, mail server.); a security policy that limits one Linux VPS from interfering with, or even seeing, another Linux VPS; and advanced resource management, which controls how system resources are shared among Linux VPS accounts.

- **Functional Isolation** -- Each Linux VPS (and the applications that run on it) are configured independently. Customers have full administrative control, including the ability to install, customize, and run almost any custom or off-the-shelf application. Additionally, customers have complete control of their Linux VPS configuration and settings.

- **Fault Isolation** -- Similar to separate physical servers, a fault in one application running within a Linux VPS account does not affect any applications running on other Linux VPS accounts on the same physical server.

- **Performance Isolation** -- FairShare technology manages how each Linux VPS uses system resources like Central Processing Unit (CPU), memory, and network, which protects each Linux VPS from the misbehavior of others on the same server.

Operating System Level Virtualization Features

The software architecture of operating system-level virtualization is different from traditional virtual machine architecture because it always runs the same operating system (OS) kernel as the host system. It does this while continuing to enable multiple Linux distributions in individual Linux VPS accounts.

This single-kernel implementation technology enables you to run Linux VPS accounts with near-zero overhead. Linux VPS accounts offer an order of magnitude higher efficiency and manageability than traditional virtualization technologies.

Each physical server utilized by Linux VPS is divided into small computational partitions. Each partition operates as a real, physical server. For Unix-like systems, Linux VPS technology is similar to an advanced extension of the standard `chroot` mechanism.

Operating System-level Virtualization

From the point of view of applications as well as Linux VPS users, each Linux VPS is an independent system. This independence is provided by a virtualization layer in the kernel of the Host OS which is an operating system installed on the Hardware Node (HN), a computer where the Operating System level virtualization is installed for hosting Linux VPS accounts.

Note: Only a small portion of the CPU resources is spent on virtualization (approximately one or two percent). The main features of the virtualization layer implemented in operating system-level virtualization are the following:

- Any Linux VPS looks and behaves like a regular Linux system. It has standard startup scripts; software from vendors can run inside a Linux VPS without operating system-level virtualization-specific modifications or adjustment.

- A user can change any configuration file and install additional software.

- Linux VPS accounts are completely isolated from each other (file system, processes, Inter Process Communication (IPC), `sysctl` variables).

- Processes belonging to a Linux VPS are scheduled for execution on all available CPUs. Consequently, Linux VPS accounts are not bound to only one CPU and can use all available CPU power.

Network Virtualization

The network virtualization layer isolates Linux VPS accounts from each other and from the physical network:

- By default, each Linux VPS has its own IP address. In addition, Linux VPS offers the fee-based option of supporting multiple IP addresses.

- At all times, network traffic for one Linux VPS is isolated from the other Linux VPS accounts. In other words, Linux VPS accounts are protected from each other in the way that makes traffic snooping impossible.

- Firewalls may be used inside a Linux VPS (the user can create rules limiting access to some services using the canonical `iptables` tool inside the Linux VPS). In other words, it is possible to set up firewall rules from inside a Linux VPS. In addition, IP tables can offer dynamic Denial of Service (DoS) by means of configurable chain rules

- Routing table manipulations and advanced routing features are supported for individual Linux VPS accounts. For example, setting different maximum transmission units (MTUs) for different destinations, specifying different source addresses for different destinations.

- IP tables can offer dynamic Denial of Service (DoS) by means of configurable chain rules.

Resource Management

Resource management is important for Linux VPS. Utilization of the resources available on your account is considerably higher than that of typical computer system. Linux VPS resource management controls the amount of resources available to each Linux VPS accounts. The controlled resources include such parameters as CPU power, disk space, and a set of memory-related parameters. Resource management enables your account server to efficiently provide the following functions:

- Share available resources among individual accounts.
- Maintain a server's part in network Quality of Service (QoS).
- Isolate performance and resource isolation.
- Protect the server from Denial of Service (DoS) attacks.
- Collect usage information for system monitoring.

Skel Package

Technicians pre-configure each Linux VPS account with the following core services residing on the virtual private server account:

- Web -- Hypertext Transfer Protocol (HTTP) and HTTPS.
- Email -- Simple Mail Transfer Protocol (SMTP), Post Office Protocol (POP), and Internet Message Access Protocol (IMAP).
- File Transfer Protocol (FTP)
- Shell access tools -- Telnet, Secure Shell (SSH), cron

These core services are managed by support staff, but can be configured by the customer to run according to their specific needs. Our support staff will provide basic instructional support for configuring and using the core services, as well as maintaining the system functionality of these services free of charge. We reserve the right to adjust Linux VPS resources as required to preserve an optimal operating environment for all Linux VPS customers.

By default your new Linux VPS account is pre-configured as a Web and email server. Your account begins as a copy-on-write (COW) image of a technician-tested, basic RHEL installation. However, you can configure your Linux VPS account to provide additional services

Copy-on-Write

Linux VPS fully utilizes the advantages of a Copy-on-Write (COW) file system. As you update and customize the server, Linux VPS continues to use central files maintained by technicians through file system overhead with skel. The COW file system ensures that your server has relatively unfettered access to system resources such as Random Access Memory (RAM). Over time, files which are unique to your own server and configuration can grow in size to suit own your needs.

The COW file system is especially important for virtual memory operating systems; when a process creates a copy of itself, the pages in memory that might be modified by either the process or its copy are then marked copy-on-write. When one process modifies the memory, the operating system's kernel intercepts the operation and copies the memory. It does this so that changes in the memory associated with one process are not visible to any other processes.

The ability to sparsely use memory is an important advantage of the COW file system. This advantage is because the usage of physical memory only increases as data are stored in physical memory. And this enables efficient hash tables to be implemented. The hash tables use only a little more physical memory than necessary to store the objects they contain.

RPM

Your Linux VPS account supports RPM, an open packaging system for Linux and UNIX systems distributed under the terms of the GPL. The package maintains a database of installed packages and their files. This enables you to maintain and upgrade your account configurations and customizations with minimal risk of losing them as you do so.

Alternatives to Operating System-Level Virtualization

Linux VPS technology virtualizes servers on an operating system (or *kernel*) layer. The technology partitions a single physical server into multiple small computational partitions. From the point of its owner, each partition operates as though it is a physical server. For UNIX-like operating systems, this technology provides an advanced extension of the standard `chroot` mechanism.

The operating system level architecture has low overhead that helps to maximize efficient use of server resources. Due to a single-kernel approach, this type of virtualization introduces only a negligible overhead and enables hundreds of virtual private servers to run on a single physical server. In contrast, approaches such as VMware workstation virtual machine emulation, Xen machine para-virtualization, and/or User Mode Linux (UML), cannot achieve such a high level of density. This is due to the overhead required by multiple kernels. Operating system-level virtualization does not enable you to run different operating systems (or *kernels*).Different libraries and distributions of an operating system are possible.

To be fair, using operating system-level virtualization means the response time of a Web server does increase as the numbers of virtual environments are increased. However, response times are acceptable (under two seconds) on a physical server with 75 Gigabytes (GB) of Random Access Memory (RAM) when over 100 virtual environments are operating on the server. In comparison, emulation and para-virtualization implementations can typically host only approximately 10 virtual environments on a 75 GB physical server.

As there is a single kernel which maintains all the partitions, isolation and resource management become critical. Without proper isolation security can be compromised, and without proper resource management an application from one partition can abuse resources and thus cause a denial of service (DoS) for other partitions. The following are examples of finite resources which must be controlled and limited on a per-partition basis:

- CPU time
- Disk space
- I/O bandwidth
- Network access
- RAM
- Shared memory
- Locked pages
- Number of processes
- Socket buffers

Linux VPS technology provides a set of more than twenty finite resources which are controlled and limited on a per-partition basis.

Overview of Alternatives

Following are alternatives to the Linux VPS technology:

- **Chroot** – A UNIX command-line program to be used for process management. The program enables you to run and tightly control different instances of the RHEL environment on a single system. These instances operate simultaneously and without an intervention such as rebooting.
  Although the program offers file system isolation, it does not offer any of the other features of other alternatives.
  For more, refer to the manual pages included with the RHEL (or other Gnu-Unix) operating system.

- **Linux Vserver** – An open-source software distributed by the Linux-VServer Project (http://linux-vserver.org/Welcome_to_Linux-VServer.org) under the terms of the Gnu Public License (GPL). The software provides kernel-level virtualization for GNU/Linux systems by enabling multiple virtual units running on the same kernel.

- **FreeBSD Jail** – A utility provided by the FreeBSD operating system (http://www.freebsd.org/). The utility manipulates operating system `chroot` capabilities to provide discrete, secure partitions. For more, refer to the operating system man pages or, for details regarding the utility refer to Chapter Four of the *FreeBSD Architecture Handbook* (http://www.freebsd.org/doc/en_US.ISO8859-1/books/arch-handbook/jail.html).

- **Containers** (or *Zones*) –The Solaris Enterprise System (http://www.sun.com/software/solaris/) provides support for isolated, virtual environments within one physical server. These environments are referred to as *zones* or *containers*. For more, refer to the *Zones and Containers FAQ* located on the Open Solaris Web site (http://www.opensolaris.org/os/community/zones/faq/).

- **Virtuozzo** – A solution which provides isolated partitions at the operating system level. For more, refer to the SWSoft Web site (http://www.swsoft.com/en/products/virtuozzo).

Features offered by Linux VPS

The alternatives offer some, but not all, of the features of Linux VPS. Only the Linux VPS implementation of operating system-level virtualization offers all of the following features:

- File System Isolation
- Disk Quotas
- I/O Rate Limiting
- Memory Limit
- CPU Quotas
- Network
- Isolation
- Partition check pointing
- Live migration

Resource Management and FairShare Technology

The phrase *Fair Share* refers to an innovative technology which promotes fair use of shared system resources such as CPU, memory, and network resources. The technology enables customer Web sites and applications to perform consistently. It does this by preventing any single Linux VPS from abusing shared resources. Each Linux VPS is provisioned with an allocation of its own disk space and processes. Linux VPS controls usage of a number of resources including CPU, memory, and network.

Resource allocations and FairShare technology are used to accomplish two specific goals:

- Prevent abuse by any single Linux VPS
- Differentiate the capacity of Linux VPS plans

A core principle of FairShare technology is to enable each Linux VPS to access appropriate levels of resources while preventing any single Linux VPS from abusing those resources by using more than its fair share. Usage thresholds are put into place to prevent abuse. Available resources and the demand on those resources then regulate the performance of each Linux VPS account.

There are three Linux VPS plans, as follows:

- Linux VPS Basic
- Linux VPS Pro
- Linux VPS Pro Plus

During a low-activity period on the server, a Linux VPS could use as much CPU as it demands. However, at the point there is competition for that CPU resource, usage thresholds are applied and the Linux VPS plan type dictates access to available resources. Each Linux VPS is allocated its fair share, but the fair share of a Linux VPS Basic plan is less than that of a Linux VPS Pro. And the fair share of a Linux VPS Pro is less than that of a Linux VPS Pro Plus.

Resource allocation and the implementation of FairShare technology provide an important competitive advantage of the Linux VPS platform. The information significant to customers is that which will guide appropriate usage of each Linux VPS plan and possibly indicate when an upgrade to a larger Linux VPS plan is necessary. The guidelines below may assist customers in making such decisions.

The main goal of resource management in Operating System-level Virtualization is to provide Service Level Management or Quality of Service (QoS) for Linux VPS accounts. Resource management settings prevent serious impacts resulting from the resource over-usage (accidental or malicious) of any other Linux VPS accounts. Using resource management parameters for Quality of Service management also enables data center administrators to enforce fairness of resource usage among Linux VPS accounts and service quality for Linux VPS accounts.

Default Applications for Linux VPS

The following table describes the applications which are installed by default in the configuration of your Linux VPS account. In addition, the table provides an overview of the location and usage of the application.

Application	Location	Usage
Apache modules	• Main: /www/modules	Apache modules are code segments that are written to comply with the Apache API specification and can be loaded into the Apache Web Server. Apache modules can be loaded in one the following ways. The modules are statically loaded in the compiled `httpd` daemon. Or they are dynamically loaded in the Web server configuration file.
Auto responder support	• N/A	Through CPX: Control Panel, your account supports auto responder (or *auto reply*) as a means of distribution for information about your organization. It provides an email alias which executes a program that automatically replies to any email sent to the specified address.
FTP server	• Configuration: /etc/proftpd.conf • Main: /usr/sbin/proftpd /usr/sbin/in.proftpd	proFTPD provides anonymous configuration and support for the Transport Layer Security (TLS) protocols such as anonymous File Transfer Protocol (FTP) and FTP server processes. Your account supports unlimited FTP users.
GCC	• Main: /usr/bin/gcc • Main Library: /usr/lib/gcc • Documents: /usr/shre/doc/dovevot-0.*xx.yy*	GNU Compiler Collection (GCC) provides program language compilers.

Application	Location	Usage
IMAP	Documents: /usr/share/doc/dovecot-0.*xx.yy* Main: /usr/sbin/dovecot Other Important Directories: /var/run/dovecot /var/run/dovecot-login Configuration: /etc/dovecot.conf /etc/pam.d/dovecot	Your account uses Dovecot as secure Internet Message Access Protocol (IMAP) server software. Your account does not limit the number of IMAP email users.
OpenSSH	Documentation: /usr/share/doc/ openssh-3.*x* Configuration: /etc/ssh Main Files: /usr/bin/ssh /usr/bin/scp /usr/bin/ssh-agent Other Common SSH Utilities: /usr/bin	OpenSSH provides connectivity tools that encrypt all traffic (including passwords) to eliminate eavesdropping, connection hijacking, and other attacks. OpenSSH also supports secure tunneling capabilities and several authentication methods, and supports commonly used versions of the Secure Shell (SSH) protocol.
Perl and mod_perl	• Mod_perl: /www/modules/ mod_perl.so • Main perl executable: /usr/local/bin/perl • Main perl directory: /usr/lib/perl5	Perl provides a cross-platform programming language.
POP Server	• N/A	Your account supports Post Office Protocol (POP) services for the purpose of handling email. Your account does not limit POP email users.
Python	• Main Executable: /usr/bin/python • Library: /usr/lib/python2.*x* • Docs: /usr/share/doc/python-2.*x.y*/	Python provides a dynamic programming language comparable to Tcl, Perl, Ruby, Scheme, and Java.

Application	Location	Usage
Red Hat	Main Version: /etc/redhat-release RPM: /bin/rpm	Linux VPS utilizes the Red Hat Enterprise Linux (RHEL) operating system version 3.*x.y*. Red Hat Package Manager (RPM) version 3.*x.y* provides a command line-driven package management system. The package enables you to install, uninstall, verify, query, and update software features on your account.
Ruby	• Main Executable: /usr/bin/ruby • Main Libraries: /usr/lib/ruby /usr/lib/site_ruby • Main Documents: /usr/share/doc/ruby-1.*x.y* • Development Documents: /usr/share/doc/ruby-devel-1.*x.y*	Ruby provides interpreted scripting language for object-oriented programming.
Sendmail SMTP Server	• Main Executable: /usr/sbin/sendmail • Main Configuration directory: /etc/mail • Documents: /usr/share/doc/sendmail	SMTP – Sendmail with unlimited aliases.
SSL	• N/A	Privacy and encryption provided by support for the Secure Sockets Layer (SSL) protocol 2.0. Your account defaults to utilize a shared SSL certificate.
Unauthorized relay protection	• N/A (a pre-configured component of Sendmail)	Your account supports Secure Shell (SSH) service under inetd and enables the ability to limit the rate of incoming SSH connections to eliminate this additional load. If you modify your inetd configuration, verify your configuration continues to work with the new configuration.

Application	Location	Usage
Web Server	- Main Directory: /www - Main Executable: /usr/sbin/httpd - Main Configuration File: /www/conf/httpd.conf - Restart Apache Script: /usr/local/sbin/restart_apache - Document Root: /www/htdocs - Documents: /usr/share/doc/httpd-2.*x*	Apache HTTP (or *Web*) Server and Web Server Modules provided by the Apache Hypertext Transfer Protocol (HTTP) Server version 2.*x*.

Vinstall Utilities Library

Linux VPS accounts include a library of custom installation scripts (or *vinstall utilities*). At the time of initial release, all Linux VPS plans include a library of vinstall utilities which are limited to the following:

Note: Java applications are recommended for Linux VPS Pro Plus accounts only. For more, see "Regarding Java Applications" on page 16.

- ClamAV
- CPX: Control Panel
- FormMail
- Java Development Kit and Java Run-time Environment.
- Mailman
- MySQL
- PHP: Hypertext Processor (php4 and php5)
- phpMyAdmin
- Postgres
- Python
- ShopSite (Fee-based, optional feature)
- SpamAssassin
- SquirrelMail
- Swish-e
- Tomcat
- Urchin
- Webalizer
- WordPress
- Zend Optimizer

Linux VPS Resource Allocation

There are three Linux VPS plan levels: Basic, Pro, and Pro Plus. The plans utilize FairShare technology as well as operating system-level virtualization to allocate server resources. There are six aspects of the resource allocation, as follows:

- Disk Space – Determines the maximum number of gigabytes (GB) of data you can store on your virtual private server account.

- Total Processes – Determines the maximum number of tasks (both active and inactive) on your virtual private server account.

- Total Open Files – Determines a maximum number of files which the private server can open at a time.

- CPU Priority – Determines a maximum amount of time, relative to other plan types, on the server hardware Central Processing Unit (CPU).

- Memory – Determines a maximum number of megabytes (MB) of memory your private server utilizes.

- Burstable Memory – Determines a maximum number of megabytes (MB) or gigabytes (GB) of memory available to your virtual private server account during short, spikes or peak usage periods.

The following table describes the resource allocations for each Linux VPS plan level:

Resource Allocations	Linux VPS Basic	Linux VPS Pro	Linux VPS Pro Plus
Disk Space	10 GB	20 GB	40 GB
Total Processes	75	150	300
Total Open Files	800	1600	3200
CPU Priority	1	2 (or twice the priority of Basic)	4 (or quadruple the priority of Basic)
Memory	128 MB	256 MB	512 MB
Burstable Memory	512 MB	1 GB	1.5 GB

The following table describes usage guidelines for each Linux VPS. There are three usage guidelines, as follows:

- Bandwidth Usage – Guides amount of data passed through a communication channel during a given period of time.

- Average Number of Sub-hosts – Guides the number of sub-hosts supported by your virtual private server account.

- Average Number of email users -- Guides the number of sub-hosts supported by your virtual private server account.

Usage Guidelines	Linux VPS Basic	Linux VPS Pro	Linux VPS Pro Plus
Bandwidth Usage	Less than 100 GB per month.	Less than 200 GB per month.	Less than 400 GB per month.
Average Number of Sub-hosts	15	30	60
Average Number of Email users	50	100	200

Suggested Uses and Applications by Plan

Following are suggested uses for a Linux VPS Basic plan:

- Web sites
- Email

Following are suggested uses for a Linux VPS Pro plan:

- Web-based Commerce (or *E-Commerce*)
- Databases
- Dynamic Applications

Following are suggested uses for a Linux VPS Pro Plus plan:

- Multimedia development suites
- Enterprise development suites
- Java Development Kit (JDK)

Regarding Java Applications

Note: Many Java applications consume significant CPU and memory resources and may not be appropriate for use on a Linux VPS. Java applications on a Linux VPS should be restricted for use only on Web sites with a low expected workload. In addition, some larger Java applications may not be suitable for use on a Linux VPS even with low workloads. You must conduct sufficient performance testing of your Java application on a Linux VPS before relying on it for critical business needs. You must build contingency plans in case your Java application does not perform as expected; alternative solutions may include:

- Extensive optimization of the Java application
- Moving the Java application to a dedicated server such as the Managed Private Server (MPS).
- Implementing an alternative solution to using Java. (For example, moving away from Java to an optimized C program.)

There are a wide range of Java-based applications. Some are designed to handle multiple services and others focus on a smaller range of services. As an example, Zimbra (http://www.zimbra.com/) is open-source server and client software used to process email, group calendaring, and contacts. In addition, the software supports Web document management and authoring. The software offers a wide range of services, some of which duplicate applications included with the default configuration of Linux VPS Pro Plus. The software offers these services by drawing on Java and its own, duplicate installation of

Tomcat. These factors make the software better suited to a more powerful, dedicated server offering such as MPS.

Alternatively, JSP Wiki (http://www.jspwiki.org/) uses standard Java 2 Platform, Enterprise Edition (J2EE) components such as Java, servlets, and JSP. The software enables collaborative authoring for Web sites. The software enables visitors to comment upon, add to, or remove from Web page (or *wiki*) content. The open-source software focuses on providing a singular, unique service. It does this by utilizing the Java components which are part of the default configuration of Linux VPS Pro Plus. And the software does not duplicate services which are also supported by Linux VPS, such as email.

JSP Wiki is an example of a Java-based application which is well-suited to Linux VPS Pro Plus only when put under low to moderate workloads. Results will vary by application, but JSPWiki running on a Linux VPS Pro Plus will potentially handle up to tens of thousands of wiki page views per day. Should hundreds of thousands of wiki page views per day be required, upgrading to a dedicated server such as an MPS is recommended.

For details of Linux VPS plan resource allocations and recommended usage, please refer to the following sections of this document:

- "Linux VPS Resource Allocation" on page 15.
- "Suggested Uses and Applications by Plan" on page 16.

Caveats

Each Linux VPS operates in a sandboxed environment that protects, but does not always prevent, performance from being impacted by another Linux VPS. All customers share some common resources including CPU, memory, network, and disk space. Resource availability is not guaranteed.

The information in the tables above is provided for informational purposes only and must be recognized as a guideline and not a performance guarantee. Operating system-level virtualization technology combined with sound server administration practices will enable resources to adjust as required in order to preserve an optimal operating environment for all Linux VPS customers.

Actual resource availability may fluctuate both above and below these values. Ultimately, the demand on available resources will determine the performance of each Linux VPS. While this information may be helpful in gauging appropriate usage, it should not be the sole determinate in capacity planning.

We reserve the right to adjust Linux VPS resources as required. These adjustments preserve an optimal operating environment for all Linux VPS accounts.

Document Conventions

The conventions used in this document are designed to be completely predictable and are used for the following specific purposes.

Conventions List

Typeface	Usage
Italic	Used to indicate the following: The first mention of new terms in any information unit. For example: 　　The *rudaplex* and the *strataguide* have been the modified for this model. References to titles of books, chapters, headings, CDs, diskettes, or software programs. For example: 　　Refer to *The Technical Manual* for technical term descriptions. Variables that the user types. For example: 　　Type the *User ID* in the User ID text box.
Bold	Used to indicate the following: Exact text strings typed. For example: Type ABCDEFG. Keyboard keys pressed. For example: Press Ctrl+a, then press Enter.
[Blue Underline](#)	Used to indicate linked email, IP, Network, or Web addresses. For example: Go to http://www.microsoft.com for more information about Microsoft products.
Cross-Reference	Used to indicate a reference to another part of the same document. The grey portion of the cross-reference is hot linked to the appropriate section of the document, followed by a page number, also hot-linked to the same portion of the document. For example: For more information about the Document Conventions, see the "Document Conventions" on page 18.
`Operating System Text`	Used to indicate text that appears in a shell session for an operating system. The displayed text pertains to operating system text only, not application elements. For example: Type `LIST MAIN FOLDER`. The screen displays the Main folder.

`Program Code`	Used to indicate code listings. For example:
	<pre>{
	# do something;
	}
	# check to see if $user has the attrib 'atrib'
	if (hasKey($user_obj, 'atrib', $dbh) != 1)
	{
	print "User not Authorized to update!";25
	}</pre>

Window Element	Window elements consist of anything that is displayed on window (exclusive of the operating system). This includes toolbar menu items, drop-down lists and items in a drop-down list, buttons, or anything else a user sees on screen. For example: From the Printer drop-down list, choose Local Printer. The Are You Sure? dialog box appears. Click OK. • The following message appears: User Not Authorized

Special Elements

These elements provide a variety of information ranging from warnings that the reader should not neglect to supplementary information and tips that will simply enhance their reading experience.

Used to point out helpful ideas, some not-so-obvious features, quick or alternate ways to get a particular job done, and techniques you might not discover by yourself. The Tip List special element is used when multiple tips are used.

Note: Used to highlight certain information for the reader. Generally, the Note element provides additional information on the current topic. The **Notes:** special element is used when multiple notes are required.

Important:
Used for information that is considered more pertinent to the reader than information presented in Note elements.

Caution:
Used as a hazard light in customer documentation. Information included in a Caution element could save the reader from hours of lost work.