

一、 认识 PowerShell

(一) 命令行 Shell

PowerShell 是新式命令 shell，其中包括其他常用 shell 的最佳功能。与大多数仅接受并返回文本的 shell 不同，PowerShell 接受并返回 .NET 对象。shell 包括以下功能：

- 可靠的命令行历史记录
- Tab 自动补全和命令预测
- 支持命令和参数别名
- 用于链接命令的管道
- 控制台内帮助系统，类似于 Unix man 页面

(二) 脚本语言

作为一种脚本语言，PowerShell 通常用于自动执行系统管理。它还用于经常在 CI/CD 环境中生成、测试和部署解决方案。PowerShell 是在 .NET 公共语言运行时 (CLR) 上构建的。所有输入和输出都是 .NET 对象。无需分析文本输出即可从输出中提取信息。PowerShell 脚本语言包含以下功能：

- 可通过函数、类、脚本和模块进行扩展
- 便于输出的可扩展格式系统
- 用于创建动态类型的可扩展类型系统
- 对常用数据格式（例如 CSV、JSON 和 XML）的内置支持

(三) 自动化平台

PowerShell 的可扩展性质使 PowerShell 模块的生态系统能够部署和管理使用的几乎所有技术。例如：

Microsoft

- Azure
- Windows
- Exchange
- SQL

第三方

- AWS
- VMWare
- Google Cloud

(四) PowerShell 和命令提示符的区别

Powershell	命令提示符
它是基于.NET 框架的基于任务的自动化命令行界面和关联的脚本语言。	它是 Microsoft Windows 操作系统的默认命令行解释器。
它可以解释批处理和 PowerShell 命令。	它只能解释批处理命令。
它用于控制和自动化 Windows 服务器上的应用程序和 Windows 操作系统。	它用于在控制台上执行给定的命令，可用于调试问题。
PowerShell 生成的输出不仅是字符流，而且是对象的集合。	命令提示符生成的输出只是字符流(文本)。
它既是 Shell 程序又是脚本编制环境，它支持创建用于管理 Windows 操作系统的大文件。	它只是一个 shell 系统，它允许用户仅执行简单和基本的脚本来执行批处理文件。

二、 PowerShell 发展历史

2002 年 Microsoft 开始开发一种管理命令行的新方法。 它还包括开发名为 Monad 的新 Shell。 2002 年发布的白皮书叫做 Monad Manifesto。 它包含此 Shell 的概念以及创建标准化平台的想法，该标准化平台通过自动化任务使用 .NET 框架。

Monad 的首次展示是在 2003 年美国专业开发人员大会上在洛杉矶进行的。 Microsoft 在 2005 年 6 月 17 日，2005 年 9 月 11 日和 2006 年 1 月 10 日发布了 Monad 的三个 beta 版本。 Microsoft 将 Monad 重命名为 Windows PowerShell 在同年 2006 年 4 月发布，并使其成为 Windows 操作系统的核心部分。

(一) PowerShell 1.0

2006 年 11 月，Microsoft 针对 Windows Vista, Windows XP SP2 和 Windows Server 2003 SP1 发布了 PowerShell 版本 1.0。 此版本是 Windows Server 2008 的可选组件。

(二) PowerShell 2.0

PowerShell 版本 2.0 与 Windows Server 2008 R2 和 Windows 7 集成在一起。它是针对 Windows Vista Service Pack 1, Windows Server 2003 Service Pack 2 和 Windows XP Service Pack 3 发行的。

此版本包括对托管 API 和脚本语言的更改。

以下是 PowerShell 2.0 的新功能：

- PowerShell 远程处理。
- 后台工作。
- 可步进管道。
- 脚本调试。
- Windows PowerShell ISE(集成脚本环境)。

- 网络文件传输。

(三) PowerShell 3.0

PowerShell 版本 3.0 与 Windows Server 2012 和 Windows 8 集成在一起。

Microsoft 还使其适用于带 Service Pack 1 的 Windows Server 2008，带 Service Pack 1 的 Windows 7 和带 Service Pack1 的 Windows Server 2008 R2。

此版本是(WMF3)Windows Management Framework 程序包的内部部分，该程序包包含支持远程处理的 WinRM 服务。

以下是 PowerShell 3.0 的新功能：

- 预定作业。
- 会话连接。
- 代表团支持。
- 自动模块检测。
- 改进的代码编写。
- 帮助更新。
- 新命令。

(四) PowerShell 4.0

PowerShell 版本 4.0 与 Windows Server 2012 R2 和 Windows 8.1 集成在一起。

Microsoft 还使其可用于 Windows Server 2008 R2 SP1，Windows 7 SP1 和 Windows Server 2012。

以下是 PowerShell 4.0 的新功能：

- 所需状态配置(DSC)。
- 新的默认执行策略。
- 保存帮助。
- 增强调试。

- 网络诊断。

(五) PowerShell 5.0

此版本已作为 Windows 管理框架 (WMF 5.0) 的内部部分于 2016 年 2 月 24 日发布到 Web 上。

默认情况下，此版本安装在 Windows 10 和 Windows Server 2012 上。

以下是 PowerShell 5.0 的新功能：

- PowerShell .NET 枚举。
- DSC 部分配置。
- 调试 PowerShell 后台作业。
- 在远程进程中调试 PowerShell Runspace。
- PowerShell 类定义 (属性，方法)。

(六) PowerShell 5.1

PowerShell 5.1 与 Windows 10 Anniversary 更新一起于 2016 年 8 月 2 日发布。该版本最终于 2017 年 1 月 19 日发布。Microsoft 已使其可用于 Windows 7, Windows Server 2008, Windows Server 2008 R2, Windows 7, Windows Server 2012 和 Windows Server 2012 R2。

三、 Powershell 功能特点

- dlet，这些 cmdlet 允许设置和管理断点以及查看调用堆栈。
- 选项卡扩展：选项卡扩展是自动完成的一种实现，通过按一次 Tab 键可完成 cmdlet，属性和参数名。
- 可步进管道：此功能允许将脚本块拆分为可步进管道。然后，它提供了调用脚本块的 `begin()`，`process()` 和 `end()` 方法的选项，以控制执行顺序。

- 受约束的运行空间： 它允许创建具有一组约束的 PowerShell 运行空间，其中包括访问和执行脚本，cmdlet 和语言元素的能力。
- Windows PowerShell Web 访问： PowerShell 控制台在 Windows Server 2012 中引入了基于 Web 的版本。可以从任何 Web 浏览器运行 PowerShell cmdlet，而台式机以及任何平板电脑或移动设备均不可用。
- 网络文件传输： 此功能提供本机支持，以使用 BITS(后台智能传输服务)在计算机之间优先进行文件的异步传输。
- Windows PowerShell 集成脚本环境(ISE)： 它是 Windows PowerShell 的基于主机 GUI 的应用程序。此功能提供了制表符补全，多行编辑，语法着色，上下文相关帮助，选择性执行以及对从右到左语言的支持。
- 事务： Windows PowerShell 的此功能使我们能够启动事务，指示哪个命令是事务的一部分以及回滚或提交事务。

四、 Powershell 与 Bash Shell 的区别

PowerShell 和 Bash Shell 都使用它们自己的配置管理工具来提供一个智能命令行界面(CLI)。

PowerShell 与 Bash 相似。通常，bash 中使用的命令可以在 PowerShell 中使用，例如：rm，ls，“cp”。

这两个 Shell 程序都包含用于管理文件，导航目录和启动其他程序的命令。

PowerShell 与 Bash Shell 比较

Powershell	Bash Shell
PowerShell 是大多数 Windows 操作系统的命令 Shell 和相关的脚本语言。	Bash 是大多数 Linux 操作系统的命令 Shell 程序和脚本语言。
PowerShell 于 2006 年推出。	Bash Shell 于 1989 年推出。
它将输入和输出视为对象。	它始终接受输入和输出作为文本结

	构。
PowerShell 的用户界面是图形命令行界面 CLI。	Bash Shell 用户界面是基于文本的命令行界面。
它可以在从 Window 97 以上版本的 Windows 上执行。	主要针对 Linux 和 Unix 操作系统进行准备。

Shell 与 Powershell 命令比较

Shell	Powershell	描述
ls	ls, dir, get-childitem	列出文件和文件夹
cd	cd, set-childitem	变更目录
pwd	pwd, get-location	显示当前工作目录
clear	cls, clear	清屏
cp	cp, copy-item	复制文件
rm	remove-item, ri, rmdir, rd, del, rm	删除或删除文件
echo "string"	write-host "string" 或者 echo "string"	打印字符串
touch <filename.txt>	new-item <filename.txt>	创建一个新的文本文件

五、 在 Windows 上安装 PowerShell

有多种方法可以在 Windows 中安装 PowerShell。 每种安装方法旨在支持不同的方案和工作流。 选择最符合需求的方法。

- Winget - 在 Windows 客户端上安装 PowerShell 的建议方法。

- MSI 包 - Windows Server 和企业部署方案的最佳选择。
- ZIP 包 - “旁加载”或安装多个版本的最简单方法。
- 将此方法用于 Windows Nano Server、Windows IoT 和基于 Arm 的系统。
- .NET 全局工具 - 非常适用于安装和使用其他全局工具的 .NET 开发人员。
- Microsoft Store 包 - 适用于 PowerShell 临时用户进行安装的一种简单方法，但有一些限制。

(一) 使用 Winget 安装 PowerShell（推荐）

Windows 包管理器 Winget 是一种命令行工具，用户可以使用它在 Windows 客户端计算机上查找、安装、升级、删除和配置应用程序。此工具是 Windows 程序包管理器服务的客户端接口。默认情况下，winget 命令行工具作为应用安装程序与 Windows 11 和现代版本的 Windows 10 捆绑在一起。

搜索最新版本的 PowerShell

```
winget search Microsoft.PowerShell
```

输出

Name	Id	Version	Source
PowerShell	Microsoft.PowerShell	7.3.2.0	winget
PowerShell Preview	Microsoft.PowerShell.Preview	7.4.1.0	winget

使用 id 参数安装 PowerShell 或 PowerShell 预览版

```
winget install --id Microsoft.Powershell --source winget
```

```
winget install --id Microsoft.Powershell.Preview --source winget
```


NOTE

在使用 X86 或 X64 处理器的 Windows 系统上，winget 安装 MSI 包。在使用 Arm64 处理器的系统上，winget 安装 Microsoft Store (MSIX) 包。

(二) 安装 MSI 包

若要在 Windows 上安装 PowerShell，请使用以下链接从 GitHub 下载安装包。

<https://github.com/PowerShell/PowerShell/releases/download/v7.3.2/PowerShell-7.3.2-win-x64.msi>

<https://github.com/PowerShell/PowerShell/releases/download/v7.3.2/PowerShell-7.3.2-win-x86.msi>

下载后，双击安装程序文件并按照提示进行操作。

安装程序在 Windows “开始” 菜单中创建一个快捷方式。

- 默认情况，包安装位置为`$env:ProgramFiles\PowerShell\<version>`
- 可以通过“开始”菜单启动 PowerShell
- 也可以 `$env:ProgramFiles\PowerShell\<version>\pwsh.exe` 启动 PowerShell

六、 在 Linux 系统上安装 PowerShell

可在不同的 Linux 发行版上安装 PowerShell。大多数 Linux 平台和发行版每年都会发布一个主要版本，并提供一个包管理器用于安装 PowerShell。本文介绍当前支持的版本以及使用的包管理器。

为了实现最佳兼容性，请选择长期支持版本 (LTS)。

(一) 通过直接下载安装

PowerShell 7.2 引入了一个简化安装过程的通用包。从版本页中将通用包下载到 Debian 10 计算机。指向当前版本的链接是：

PowerShell 7.3.2 -

https://github.com/PowerShell/PowerShell/releases/download/v7.3.2/powershell_7.3.2-1.deb_amd64.deb

PowerShell 7.2.9 -

https://github.com/PowerShell/PowerShell/releases/download/v7.2.9/powershell-lts_7.2.9-1.deb_amd64.deb

(二) 通过包存储库在 Debian11 上安装

为简化安装和更新，已将适用于 Linux 的 PowerShell 发布到包存储库。首选方法如下所示：

```
# 安装系统组件
sudo apt update && sudo apt install -y curl gnupg apt-transport-https

# 导入公共仓库 GPG 密钥
curl https://packages.microsoft.com/keys/microsoft.asc | sudo apt-key add -

# 注册微软产品提供
sudo sh -c 'echo "deb [arch=amd64]
https://packages.microsoft.com/repos/microsoft-debian-bullseye-prod bullseye
main" > /etc/apt/sources.list.d/microsoft.list'

# 安装 PowerShell
sudo apt update && sudo apt install -y powershell
```

```
# 开启 PowerShell  
pwsh
```

PowerShell 路径

- \$PSHOME 是 /opt/microsoft/powershell/7/
- 用户配置文件是从 ~/.config/powershell/profile.ps1 中读取的
- 默认配置文件是从 \$PSHOME/profile.ps1 中读取的
- 用户模块是从 ~/.local/share/powershell/Modules 中读取的
- 共享模块是从 /usr/local/share/powershell/Modules 中读取的
- 默认模块是从 \$PSHOME/Modules 中读取的
- PSReadline 历史记录被记录到 ~/.local/share/powershell/

PSReadLine/ConsoleHost_history.txt 中

配置文件采用 PowerShell 的按主机配置，所以默认主机特定配置文件位于相同位置下的 Microsoft.PowerShell_profile.ps1 中。