PowerBash

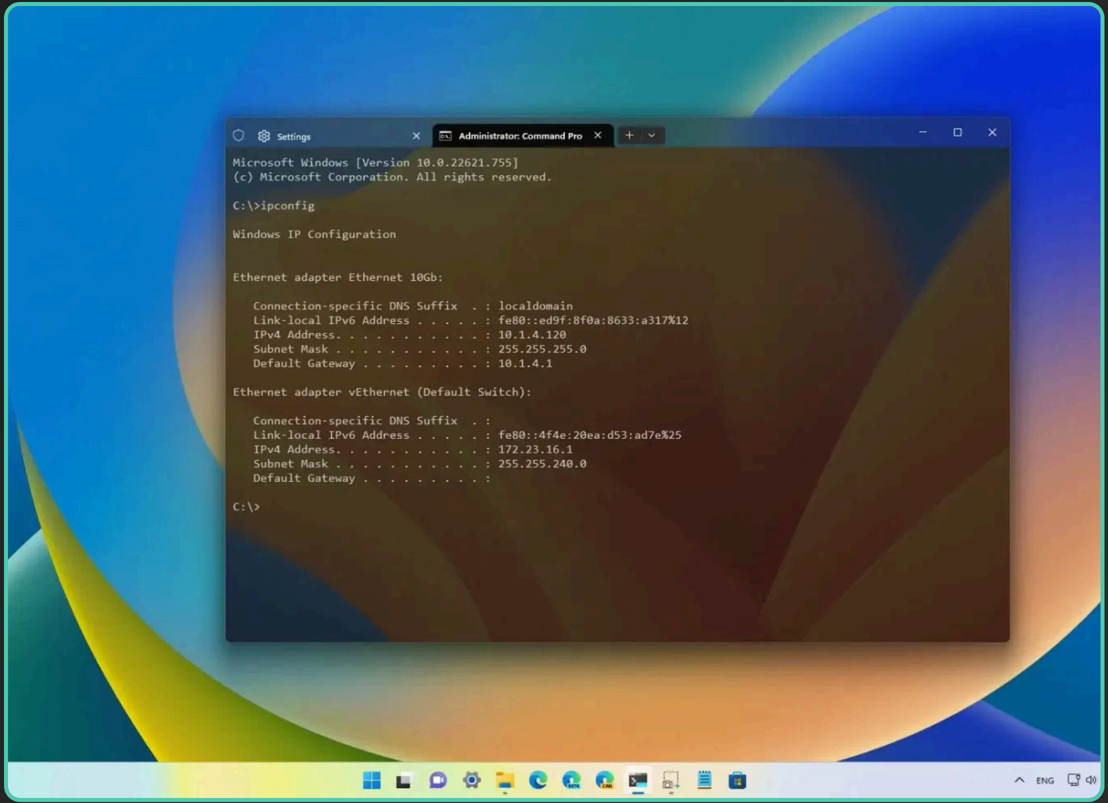
Unified System Administration
Made Simple

Vasan Dilaksan

October 2025

github.com/CyberForgeEx/PowerBash

v1.0.0



```
Microsoft Windows [Version 10.0.22621.755]
(c) Microsoft Corporation. All rights reserved.

C:\>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet 10Gb:

   Connection-specific DNS Suffix  . : localdomain
   Link-local IPv6 Address . . . . . : fe80::ed9f:8f0a:8633:a317%12
   IPv4 Address. . . . . . . . . . . : 10.1.4.120
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.1.4.1

Ethernet adapter vEthernet (Default Switch):

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::4f4e:20ea:d53:ad7e%25
   IPv4 Address. . . . . . . . . . . : 172.23.16.1
   Subnet Mask . . . . . . . . . . . : 255.255.240.0
   Default Gateway . . . . . . . . . :

C:\>
```

Fragmented Workflows Hinder Efficiency

Platform Disparity

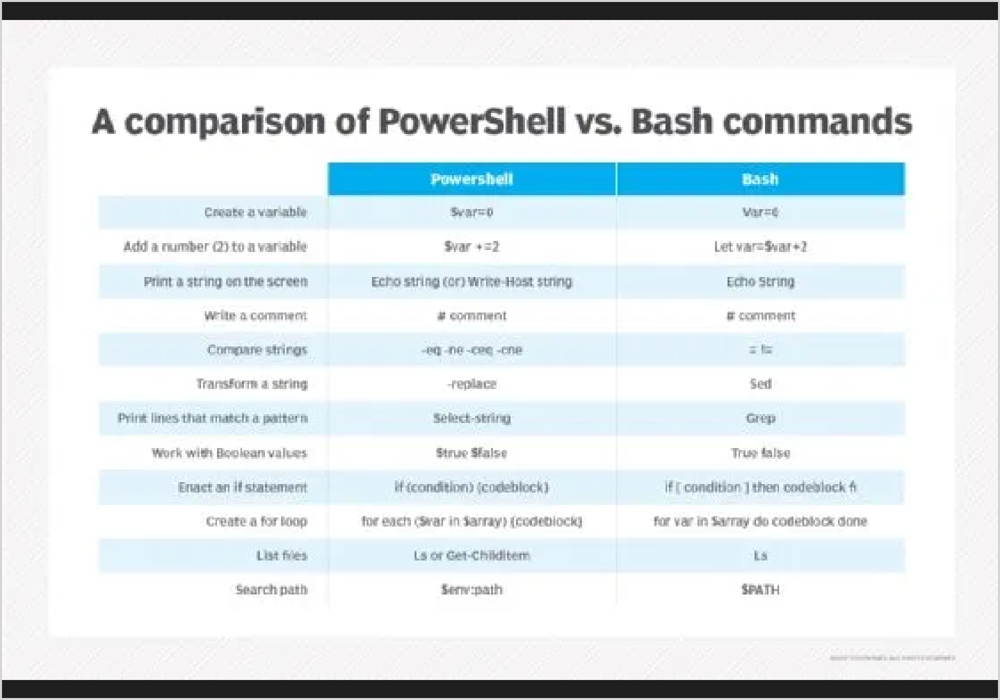
Windows (`PowerShell`) and Linux (`Bash`) require different commands and syntax for identical tasks.

The Time Sink

Repetitive manual operations consume valuable hours that could be spent on high-value projects.

Audit Inconsistency

Manual logging is often error-prone or entirely absent, leading to poor audit trails and governance.



A comparison of PowerShell vs. Bash commands

	Powershell	Bash
Create a variable	$var=0	Var=0
Add a number (2) to a variable	$var +=2	Let var=$var+2
Print a string on the screen	Echo string (or) Write-Host string	Echo String
Write a comment	# comment	# comment
Compare strings	-eq -ne -ceq -cne	= !=
Transform a string	-replace	Sed
Print lines that match a pattern	Select-string	Grep
Work with Boolean values	$true $false	True false
Enact an if statement	if (condition) {codeblock}	if [condition] then codeblock fi
Create a for loop	for each ($var in $array) {codeblock}	for var in $array do codeblock done
List files	Ls or Get-Childitem	Ls
Search path	$env:path	$PATH

Unified Solution for Cross-Platform Management

- ✓ **Seamless Compatibility**

  Works natively on `PowerShell 5.1+` and `Bash 4.0+` without additional runtimes.

- ✓ **Consistent Experience**

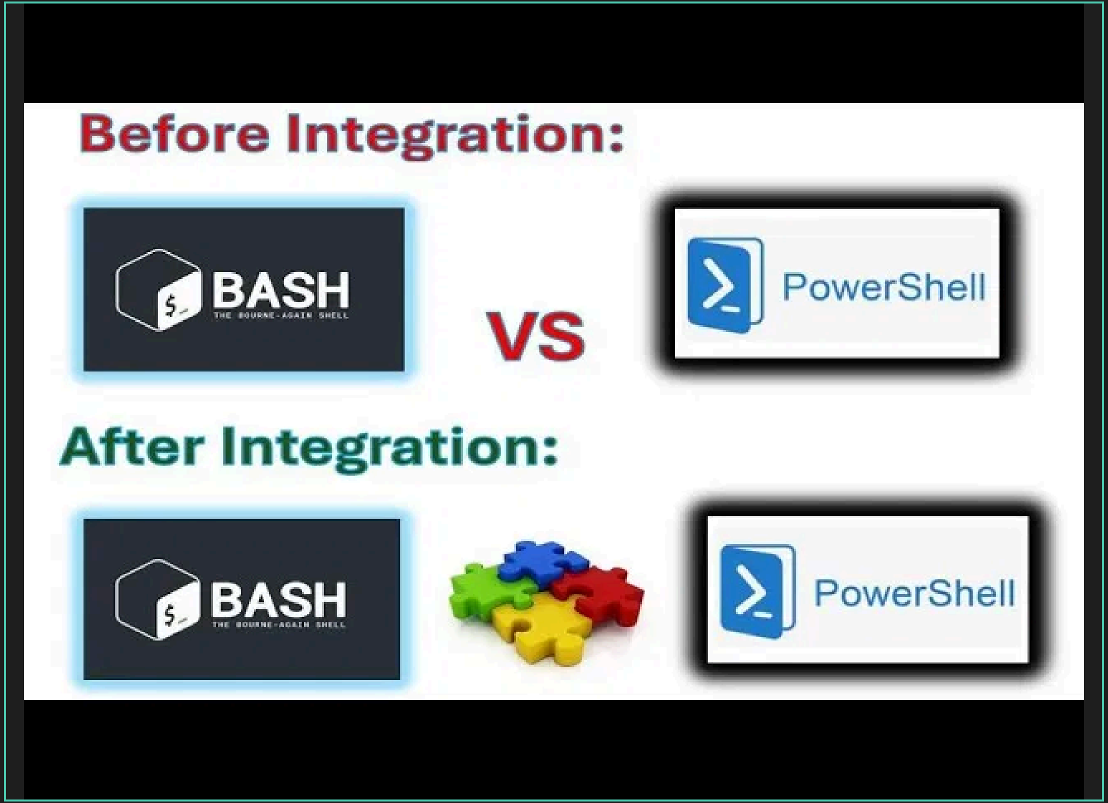
  A single, easy-to-use menu-driven interface regardless of the underlying operating system.

- ✓ **Automated Governance**

  Built-in timestamped logging and error recovery ensure reliable and auditable operations.

- ✓ **Visual Feedback**

  Color-coded output provides instant, readable status updates for administrators.



Eight Essential Tools in One Package

// Core functionality for unified administration

Process & Monitoring

- Kill Process by name/PID
- Real-time CPU tracking
- Memory usage monitoring

Storage Governance

- Disk space availability
- Automated < 30GB warnings
- Multi-drive support

Integrity & Security

- SHA256 hash calculation
- File integrity verification
- PDF AES-256 protection

Network Intelligence

- IP Geolocation (ISP/GPS)
- Interface traffic stats
- Active connection status

Advanced Capabilities and Parity

// Extending the core administrative toolkit

Automated Backups

Cross-platform compatible `ZIP compression` with automated timestamp generation for reliable data preservation.

Network Statistics

Track `bytes sent/received` and monitor active connection status in real-time across all interfaces.

Secure PDF Protection WINDOWS ONLY

Military-grade `AES-256 encryption` for PDF files, leveraging the robust qpdf utility for secure document handling.

Extensible Design

Modular function architecture allows for rapid integration of `custom administrative tools` tailored to specific needs.

Technical Architecture and Reliability

Layered Design

- User Interface Layer
- Core Functions Layer
- System API Layer
- Logging Layer
- Error Handling Layer

Technology Stack

`Windows: PowerShell 5.1+`

Native integration with Windows Management Instrumentation.

`Linux: Bash 4.0+`

Standard shell scripting for high portability across distros.

`External: ip-api.com`

RESTful integration for geolocation services.

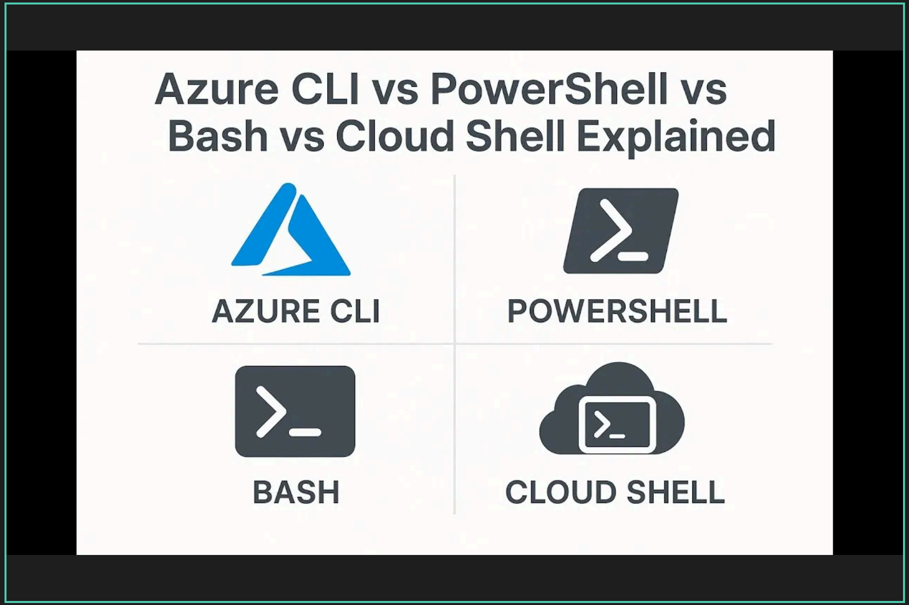
`Dependencies: curl, zip, qpdf`

Lightweight utilities for network and file operations.

Feature Parity Across Environments

Feature	Windows	Linux
Kill Process / Health Check	✓	✓
Disk Space / File Hashing	✓	✓
IP Geolocation / Network Stats	✓	✓
Automated Backup System	✓	✓
PDF Protection (AES-256)	✓	✗

** PDF Protection requires qpdf utility on Windows systems.*



Unified Experience: Despite underlying OS differences, PowerBash provides the exact same menu structure and standardized output for cross-team consistency.

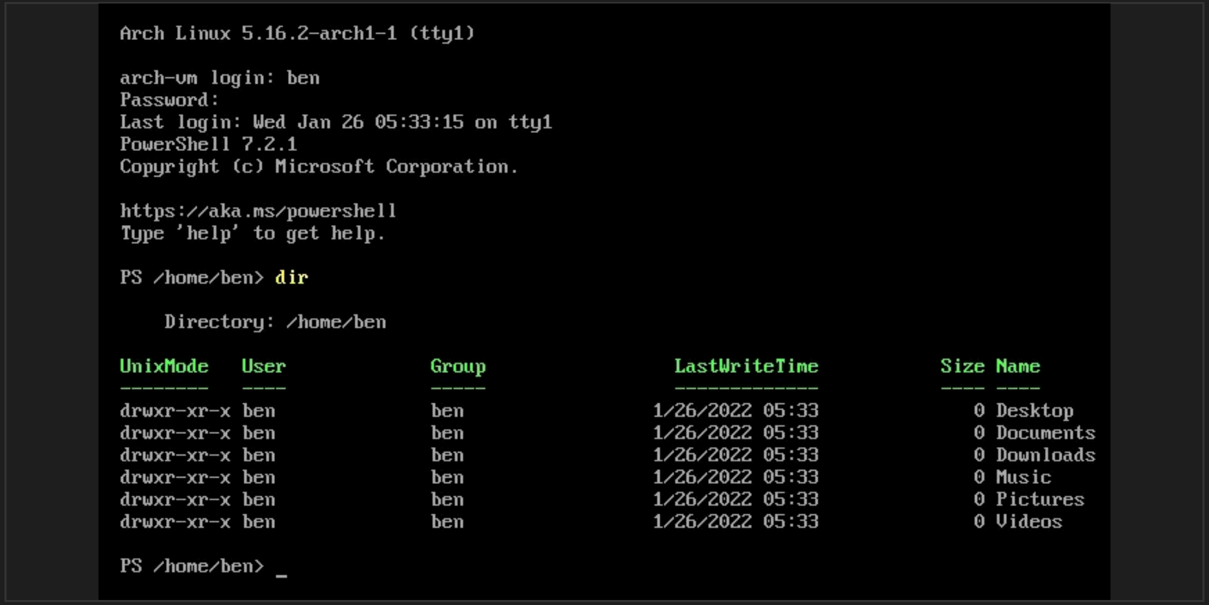
Rapid Deployment and Simple Installation

Windows Setup

```
# Clone repository
git clone https://github.com/...
# Set execution policy
Set-ExecutionPolicy RemoteSigned
# Run as Administrator
.\script.ps1
```

Linux Setup

```
# Clone repository
git clone https://github.com/...
# Make executable
chmod +x script.sh
# Run with sudo
sudo ./script.sh
```



```
Arch Linux 5.16.2-arch1-1 (tty1)

arch-vm login: ben
Password:
Last login: Wed Jan 26 05:33:15 on tty1
PowerShell 7.2.1
Copyright (c) Microsoft Corporation.

https://aka.ms/powershell
Type 'help' to get help.

PS /home/ben> dir

    Directory: /home/ben

UnixMode   User             Group                 LastWriteTime           Size Name
--------   ----             -----                 -------------           ---- ----
drwxr-xr-x ben              ben                 1/26/2022 05:33              0 Desktop
drwxr-xr-x ben              ben                 1/26/2022 05:33              0 Documents
drwxr-xr-x ben              ben                 1/26/2022 05:33              0 Downloads
drwxr-xr-x ben              ben                 1/26/2022 05:33              0 Music
drwxr-xr-x ben              ben                 1/26/2022 05:33              0 Pictures
drwxr-xr-x ben              ben                 1/26/2022 05:33              0 Videos

PS /home/ben> _
```

Comprehensive Logging and Auditing

// Automated governance for system operations

Automatic Tracking

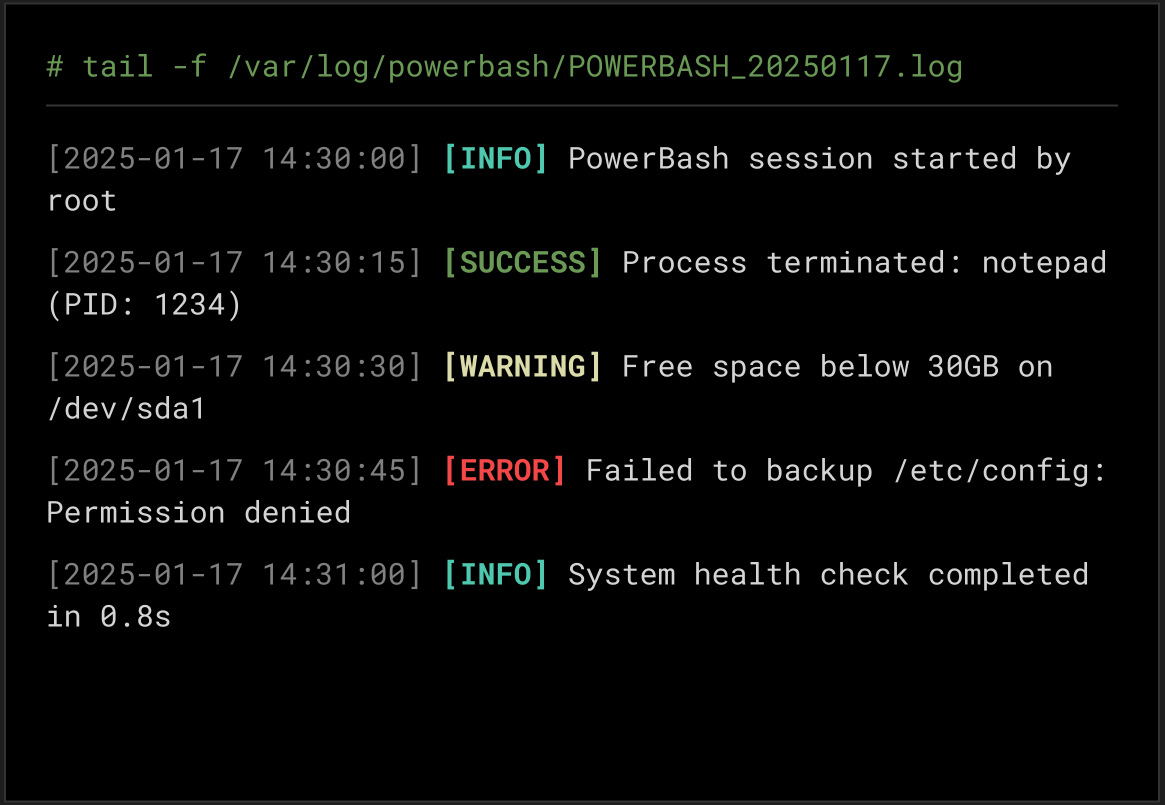
Every operation is logged automatically without user intervention, ensuring a complete history of system changes.

Audit-Ready Files

Daily log files with standardized naming conventions: `POWERBASH_YYYYMMDD.log`.

Visual Severity

Standardized levels (INFO, SUCCESS, WARNING, ERROR) for rapid analysis and automated alerting.



```
# tail -f /var/log/powerbash/POWERBASH_20250117.log

[2025-01-17 14:30:00] [INFO] PowerBash session started by
root
[2025-01-17 14:30:15] [SUCCESS] Process terminated: notepad
(PID: 1234)
[2025-01-17 14:30:30] [WARNING] Free space below 30GB on
/dev/sda1
[2025-01-17 14:30:45] [ERROR] Failed to backup /etc/config:
Permission denied
[2025-01-17 14:31:00] [INFO] System health check completed
in 0.8s
```

Security-First Administration Approach

// Hardened utilities for sensitive environments

Privilege Control

Strict requirement for `Administrative/Root` access to prevent unauthorized system modifications and ensure operational integrity.

Encryption Standards

Implementation of industry-standard `AES-256 encryption` for PDF protection, ensuring data confidentiality at rest.

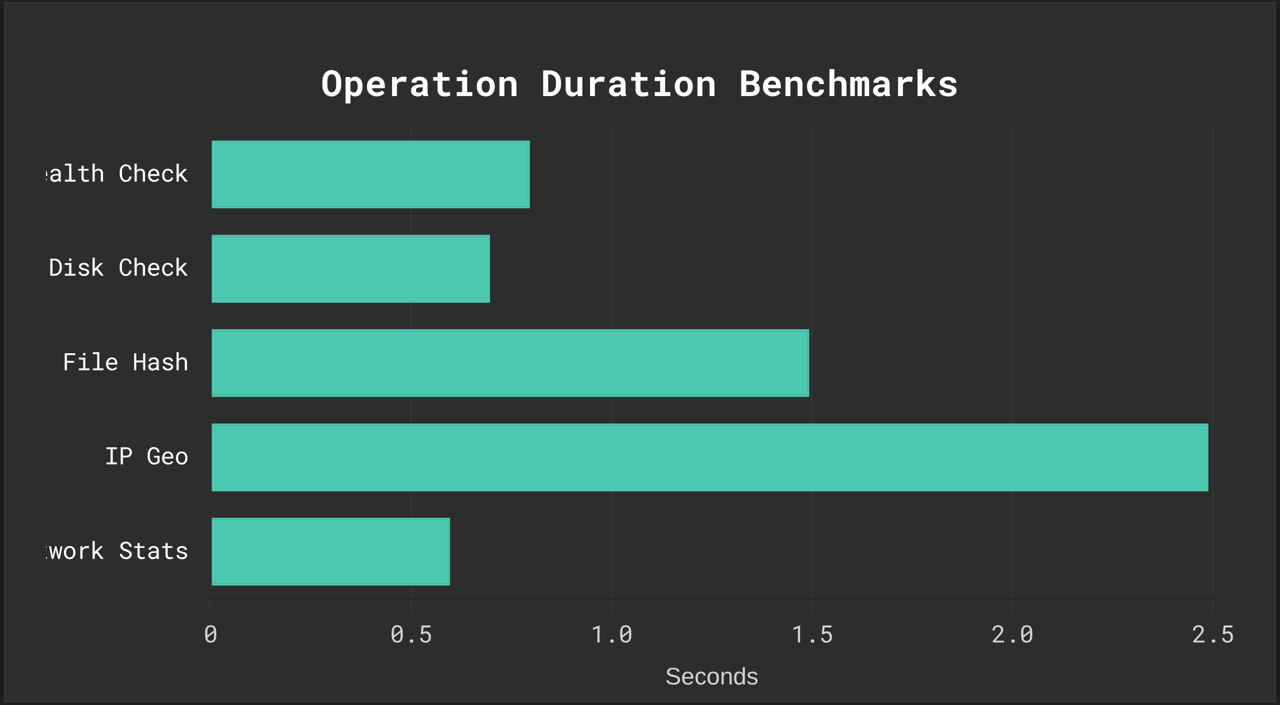
Input Sanitization

All user inputs are `validated and sanitized` to prevent command injection vulnerabilities and unexpected script behavior.

Audit Trails

Comprehensive `timestamped logging` of every operation, providing a verifiable history for security audits and troubleshooting.

Performance Benchmarks and Efficiency



< 1s

Average execution for core monitoring tasks

AES-256

Encryption standard for secure operations

Minimal

CPU and Memory footprint during idle

100%

Native execution without heavy runtimes

Future Roadmap: Version 2.0 and Beyond

// Scaling the unified administration ecosystem

Enhanced Monitoring

Real-time dashboards with historical data tracking and performance graphs.

Cloud Integration

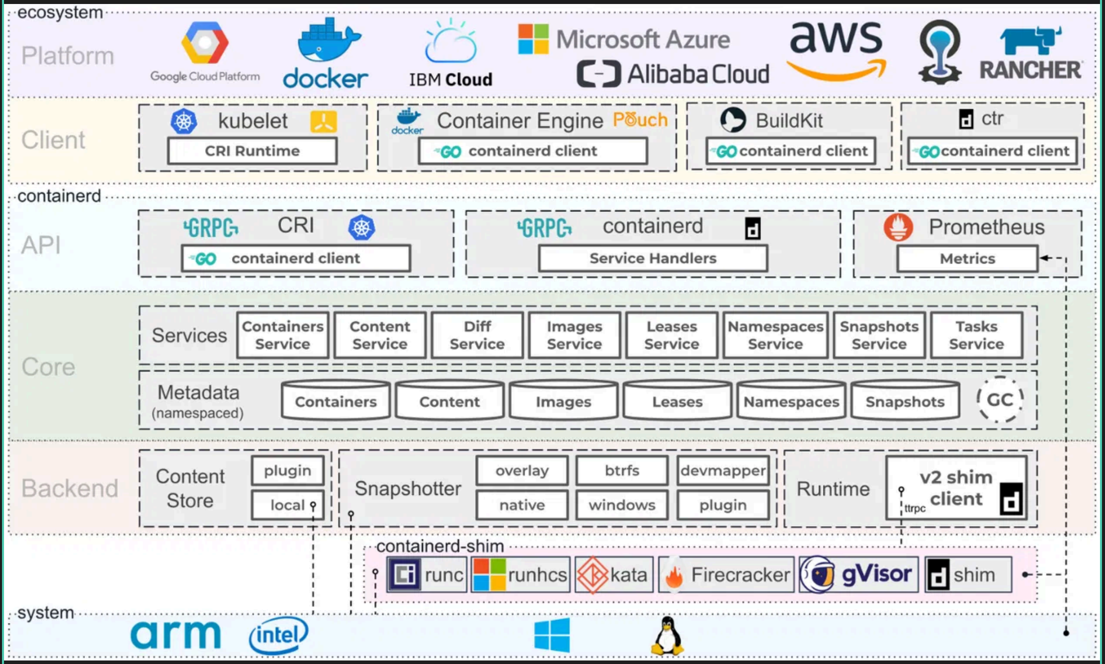
Incremental backups with direct integration to AWS, Azure, and GCP storage.

Advanced Security

Multi-file encryption and password manager integration for secure workflows.

Smart Notifications

Integrated email, SMS, and webhook alerts for automated system warnings.


ecosystem
Platform
Google Cloud Platform
docker
IBM Cloud
Microsoft Azure
Alibaba Cloud
aws
RANCHER
Client
kubelet
CRI Runtime
Container Engine
Pouch
containerd client
BuildKit
containerd client
ctr
containerd client
containerd
API
GRPC
CRI
containerd client
GRPC
containerd
Service Handlers
Prometheus
Metrics
Core
Services
Containers Service
Content Service
Diff Service
Images Service
Leases Service
Namespaces Service
Snapshots Service
Tasks Service
Metadata (namespaced)
Containers
Content
Images
Leases
Namespaces
Snapshots
GC
Backend
Content Store
plugin
local
Snapshotter
overlay
btrfs
devmapper
native
windows
plugin
Runtime
v2 shim client
ttrpc
containerd-shim
runc
runhcs
kata
Firecracker
gVisor
shim
system
arm
intel