Installation

Your basic prerequisites for Infinity Tools installation.

1: Getting Started

This guide is designed for IT professionals who are familiar with Linux, servers, networking, and modern DevOps practices. If you're comfortable with Docker, containerization, and command-line administration, you're in the right place.

Prerequisites

Before installing Infinity Tools, ensure you have:

Required

- **Linux Server** - Any modern Linux distribution (Ubuntu, Debian, CentOS, Fedora, Arch, etc.)
- **Root/Sudo Access** - Full administrative privileges
- **Internet Connection** - Required for downloading dependencies and Docker images
- **Minimum 2GB RAM** - Recommended 4GB+ for running multiple services
- **20GB+ Storage** - For applications and data (more recommended for production)

Technical Knowledge Assumed

This guide assumes you're familiar with:

- Linux command-line operations
- Docker and Docker Compose
- Basic networking concepts (ports, DNS, reverse proxies)
- SSL/TLS certificates and Let's Encrypt
- SSH connections and key management
- System administration basics

Server Setup Options

Option 1: Self-Hosted Server

If you have physical hardware or are running a virtualization platform:

Requirements:

- Physical server or VM with a supported Linux distribution
- Direct internet access or through a router/firewall
- Ability to configure DNS records (for SSL certificates)

Considerations:

- Network configuration and firewall rules
- Static IP address recommended
- Domain name for SSL certificates (optional but recommended)

Option 2: Cloud/VPS Provider

Most IT professionals will use a cloud provider. Popular options include:

Hetzner Cloud (Recommended)

- Excellent price-to-performance ratio
- EU-based with strong data protection
- Simple interface, great for automation
- Starting at ~€4/month for basic instances

DigitalOcean

- Developer-friendly interface
- Strong documentation and community
- Good performance and reliability
- Starting at $6/month for basic droplets

Other Options:

- AWS EC2, Google Cloud Platform, Azure (more complex but powerful)
- Linode, Vultr, Scaleway (good alternatives)
- Any provider that offers a Linux server with root access

Server Configuration

Regardless of your hosting choice, configure your server with:

1. **Fresh Linux Installation**
   - Clean install of your preferred distribution
   - All security updates applied (`apt update && apt upgrade` or equivalent)
2. **Non-Root User with Sudo**
   - Create a user account with sudo privileges
   - SSH key authentication configured (password auth disabled recommended)
3. **Basic Firewall**

- At minimum, allow SSH (port 22) and HTTP/HTTPS (ports 80/443)
- UFW or firewalld configured and enabled

4. **DNS Configuration** (for SSL certificates)
   - Domain name pointing to your server's IP address
   - If using Traefik, ensure DNS A record is configured before installation

System Requirements

Minimum Specifications

- **CPU:** 1 core (2 cores recommended)
- **RAM:** 2GB (4GB+ recommended)
- **Storage:** 20GB SSD (more for production workloads)
- **Network:** Stable internet connection

Recommended for Production

- **CPU:** 2-4 cores
- **RAM:** 4-8GB
- **Storage:** 50GB+ SSD with backup storage
- **Network:** 100Mbps+ connection

Storage Considerations

Infinity Tools stores data in `/opt/speedbits/` by default:

- Application data volumes
- Database files
- Configuration files
- SSL certificates

Ensure sufficient disk space and consider:

- Using separate volumes for databases
- Regular backup strategies
- Monitoring disk usage

What You'll Need Before Installation

1. Domain Name (Optional but Recommended)

For automatic SSL certificates via Let's Encrypt:

- A domain name you control
- Ability to create DNS A records
- DNS pointing to your server's IP address

Without a domain: You can still use Infinity Tools with self-signed certificates or IP-based access, but SSL won't be automatically trusted by browsers.

2. Email Address

For SSL certificate notifications and administrative purposes:

- Valid email address (for Let's Encrypt certificate warnings)
- Consider a dedicated email for server administration

3. SSH Access

Ensure you can:

- Connect to your server via SSH
- Use sudo/root privileges
- Transfer files if needed (SCP/SFTP)

Next Steps

Once your server is ready:

1. **SSH into your server**
2. **Download or transfer Infinity Tools installer**
3. **Run the installer** (covered in the next article)

The installation process will:

- Check system readiness
- Install Docker if needed
- Set up Infinity Tools in `/opt/InfinityTools/`
- Create the `infinity-tools` command for easy access

Quick Reference

Installation Path: `/opt/InfinityTools/`
Command: `sudo infinity-tools`
Data Directory: `/opt/speedbits/`
Configuration: `/opt/speedbits/_configuration/`

Common Considerations

Firewall Ports

Default ports used by Infinity Tools:

- **80/443** - HTTP/HTTPS (Traefik)
- **22** - SSH
- **Various** - Application-specific ports (configurable)

Docker Resources

Docker will be installed if not present. Consider:

- Docker daemon configuration
- Storage driver selection
- Network configuration
- Resource limits

Backup Strategy

Infinity Tools includes Borgmatic for backups, but you should also:

- Plan for backup storage location
- Configure retention policies
- Test restore procedures

Troubleshooting

Common Issues

Port conflicts: Check for existing services using ports 80/443 or other expected ports.

Network issues: Ensure firewall allows necessary traffic and DNS is properly configured.

Ready to Install?

Once your server meets these prerequisites, proceed to the installation guide which will cover:

- Establishing SSH connection
- Getting Infinity Tools onto your server
- Running the installer
- Initial configuration

Next: Installation and Initial Setup Guide (Article 2)

2: Installing Infinity Tools

This chapter covers the complete installation process for Infinity Tools on your Linux server. We'll cover SSH connection, downloading the installer, and running the installation with technical details relevant to system administrators.

Prerequisites Recap

Ensure you have:

- □ Linux server with root/sudo access
- □ SSH client installed on your local machine
- □ Server IP address and credentials
- □ Internet connectivity on both local and server

Obtaining the Infinity Tools Installer

You'll receive the Infinity Tools installer as a file (typically named `infinity-tools-installer.run`) through your preferred delivery method.

File Transfer Methods

Method 1: SCP (Secure Copy) - Recommended

```
# From your local machine
scp infinity-tools-installer.run root@YOUR_SERVER_IP:/tmp/
```

Method 2: SFTP (Secure File Transfer Protocol)

```
# Connect via SFTP
sftp root@YOUR_SERVER_IP

# Upload file
put infinity-tools-installer.run /tmp/

# Exit SFTP
quit
```

Verify File Integrity

```
# Check file size
ls -lh infinity-tools-installer.run

# Verify file type
file infinity-tools-installer.run

# Check for corruption (if checksum available)
# sha256sum infinity-tools-installer.run
# Compare with provided checksum

# Verify file permissions
ls -la infinity-tools-installer.run
```

Installation Process

Pre-Installation Setup

Make the installer executable and verify permissions:

```
chmod +x infinity-tools-installer.run
ls -la infinity-tools-installer.run
```

Run Installation

Execute the installer with root privileges:

```
sudo ./infinity-tools-installer.run
```

Installation Process Details

The installer performs the following operations:

1. **Archive Extraction:**
   - Uses makeself to extract to temporary directory
   - Creates `/tmp/selfgz[random]/InfinityTools/`
   - Contains all scripts, configurations, and dependencies

2. **File Installation:**
   - Copies files to `/opt/InfinityTools/`
   - Sets executable permissions on all `*.sh` files
   - Preserves directory structure and file ownership
3. **Dependency Installation:**
   - Installs GUM (interactive menu tool) if not present
   - Detects package manager (apt, dnf, pacman, apk)
   - Adds repository and installs via appropriate package manager
4. **System Integration:**
   - Creates symbolic link: `/usr/local/bin/infinity-tools`
   - Creates desktop entry (if GUI environment detected)
   - Sets up proper file permissions and ownership

Installation Output

Expected installation output:

```
╔═══════════════════════════════════════════════════════════════════════════╗
║                                                                           ║
║                    ∞  Infinity Tools - Linux Server Suite                 ║
║                                                                           ║
╚═══════════════════════════════════════════════════════════════════════════╝


Installing Infinity Tools...
   Source: /tmp/selfgz12345/InfinityTools
   Target: /opt/InfinityTools

Copying files...
Setting permissions...
Checking dependencies...
   Installing 'gum' (interactive menu tool)...
Creating command shortcut...
 Creating desktop shortcut...


╔═══════════════════════════════════════════════════════════════════════════╗
║                      INSTALLATION SUCCESSFUL!                             ║
╚═══════════════════════════════════════════════════════════════════════════╝
```

First Run

Launch Infinity Tools

```
sudo bash infinity-tools
```

Initial System Check

On first run, Infinity Tools will:

- Load all helper modules from `StartHelper/`
- Check system readiness and dependencies
- Initialize the interactive menu system
- Display service status and health information

Menu Navigation

The main menu provides access to:

- □□ **APPS - SOFTWARE:** Application installation scripts
- □□ **SECURITY & NETWORKING:** Security hardening and network tools
- □□ **INFRASTRUCTURE & CORE SYSTEMS:** Docker, Portainer, system tools
- □□ **BACKUP MANAGEMENT:** Borgmatic, Rclone, backup configuration
- □□ **STATUS & HEALTH:** System monitoring and health checks

Installation Architecture

Directory Structure

```
/opt/InfinityTools/
├── start-tools.sh              # Main launcher script
├── install.sh                  # Installation script
├── Solutions/                  # Application installers
│   ├── setup-traefik.sh
│   ├── setup-wordpress.sh
│   └── ...
├── Infrastructure/             # Core system tools
│   ├── install-docker.sh
│   ├── increase-security.sh
│   └── ...
```

```
├── StartHelper/                # Helper modules
│   ├── menu-builder.sh
│   ├── service-metadata.sh
│   └── ...
├── Uninstallations/            # Removal scripts
├── Fallback/                   # Recovery tools
└── Repair/                     # Maintenance utilities
```

Data Directory

Application data is stored in:

```
/opt/speedbits/
├── _configuration/             # Global configuration
├── traefik/                    # Traefik data
├── wordpress/                  # WordPress instances
├── vaultwarden/                # Vaultwarden data
└── ...
```

Configuration Management

Global Configuration

Infinity Tools uses a centralized configuration system:

```
# Main configuration file
/opt/speedbits/_configuration/config

# Port configuration
Infrastructure/ports-config.yml

# Service metadata
StartHelper/service-metadata.sh
```

Environment Variables

Key environment variables for customization:

```
# Debug mode
export INFINITY_DEBUG=true

# Custom installation path (not recommended)
export INFINITY_INSTALL_DIR=/custom/path

# Disable tracking
export INFINITY_NO_TRACKING=true
```

Security Considerations

File Permissions

```
# Check script permissions
find /opt/InfinityTools -name "*.sh" -ls

# Verify ownership
ls -la /opt/InfinityTools/

# Check for world-writable files
find /opt/InfinityTools -perm -002 -type f
```

Network Security

Infinity Tools implements:

- Network isolation between services
- SSL/TLS termination at Traefik
- Firewall configuration via UFW
- SSH hardening options

Troubleshooting

Common Issues

Installation fails with permission errors:

```
# Ensure running as root
sudo whoami

# Check file permissions
ls -la infinity-tools-installer.run

# Verify disk space
df -h /opt
```

GUM installation fails:

```
# Manual GUM installation
curl -sL
https://github.com/charmbracelet/gum/releases/download/v0.13.0/gum_0.13.0_Linux_x86_64.tar.gz
| tar -xz
sudo mv gum /usr/local/bin/
```

Command not found after installation:

```
# Check symbolic link
ls -la /usr/local/bin/infinity-tools

# Recreate if missing
sudo ln -sf /opt/InfinityTools/start-tools.sh /usr/local/bin/infinity-tools

# Update PATH if necessary
echo 'export PATH="/usr/local/bin:$PATH"' >> ~/.bashrc
source ~/.bashrc
```

Log Analysis

```
# Check installation logs
journalctl -u infinity-tools 2>/dev/null || echo "No systemd service"

# Check Docker logs
docker logs infinity-tools 2>/dev/null || echo "No Docker container"

# Check system logs
tail -f /var/log/syslog | grep -i infinity
```

Uninstallation

Complete Removal

```
# Remove Infinity Tools
sudo rm -rf /opt/InfinityTools/

# Remove symbolic link
sudo rm -f /usr/local/bin/infinity-tools

# Remove desktop entry
sudo rm -f /usr/share/applications/infinity-tools.desktop

# Remove data (WARNING: This deletes all application data)
sudo rm -rf /opt/speedbits/
```

Selective Cleanup

To remove only Infinity Tools while preserving data:

```
# Remove only the application
sudo rm -rf /opt/InfinityTools/

# Keep data directory
# /opt/speedbits/ remains intact
```

Next Steps

After successful installation:

1. **Install Docker:** Navigate to Infrastructure → Install Docker
2. **Configure Traefik:** Set up reverse proxy and SSL
3. **Deploy Applications:** Install your required services
4. **Configure Backups:** Set up Borgmatic for data protection
5. **Security Hardening:** Apply security configurations

Quick Reference

```
# Start Infinity Tools
sudo infinity-tools

# Manual execution
cd /opt/InfinityTools && sudo bash start-tools.sh

# Check status
sudo infinity-tools --status 2>/dev/null || echo "Status check not available"

# View logs
sudo journalctl -f -u infinity-tools 2>/dev/null || echo "No systemd service"

# Update (if available)
cd /opt/InfinityTools && git pull 2>/dev/null || echo "Not a git repository"
```

Installation Complete: Infinity Tools is now installed and ready for configuration. Proceed to the next chapter to begin deploying your infrastructure stack.

Next: Infrastructure Setup and Application Deployment (Chapter 3)