Alpine Linux

LXC Setup

Version 0.2 alpha by: Trevor Smale

Nov 4th 2024

Overview

Intro

LXC Containers are minimal, low resource environments much like Docker or Podman containers that can be run in ProxMox. This is a guide for proxmox users to create a container for practicing for the RHCSA & RHCE with low stakes machines that can be created and destroyed from templates.

🪶 LXC

LXC containers are lighter and faster than virtual machines (VMs) because they share the host's kernel and resources, avoiding the overhead of running a separate operating system for each instance. This allows containers to use less memory and CPU compared to VMs, which need to emulate full hardware environments through hypervisors.

🏔️ Alpine

Alpine Linux is a lightweight, security-focused Linux distribution designed for efficiency and simplicity, commonly used in containerized environments due to its minimal resource footprint.

🧐 In this Guide

We will go through the following procedures:

1. Adding & Configuring a new user with sudo privledges
2. Installing, Configuring and Activating SSH services for remote access
3. Installing useful packages like VIM, Python, GO, SSH, Ansible, Podman ++
4. Converting completed LXC container into a template for replication

🔍 Pre-Requisites

- Must have a ProxMox VE setup.
- Must have prior understanding of how to find and download LXC templates within ProxMox.
- Must have prior linux command line experience.

⚙️ Recommended Container Specs

- Alpine 3.20 LXC Template
- Processors = 1
- Memory = 512 MiB
- Swap = 512 MiB
- Bootdisk = 4GiB (1GiB Req'd for packages)

⚙️ Configuration Procedure

👨‍🚀 Add user and make them Super🦄

1. Create a new user: **`apk add sudo'`**
2. Create a password for that new user **`adduser 'username' wheel`**
3. Update the sudoers file **`visudo`**

Changes to sudoers file by adding a line above or below the root user privileges. This can be found near bottom add user below root and give same privs

`'username' ALL=(ALL) ALL\`

5. **Write** changes and **Quit vi (esc :wq)**
6. Become this new user. **`su 'username'`**
7. test **`sudo`** powers **`sudo apk update`** 👍

☎️ Install and Configure SSH 🔧

Adding SSH involves several steps that must be done correctly in order for things to work properly

1. **`apk update`**
2. **`apk add openssh`**
3. **`rc-update add sshd`**
4. **`vi /etc/ssh/sshd_config`**

These are the settings we must modify using vi. We do this by first un-commenting the line by removing the **#** mark, then altering the text. You will find these lines sequentially from top to bottom:

Fields in the **sshd_config** file that need to be altered:

- AuthorizedKeysFile
- PasswordAuthentication yes
- Hostkey /etc/ssh/ssh_host_ed25519_key
- PermitRootLogin no
- PubkeyAuthentication yes
- AuthorizedKeysFile .ssh/authorized_keys

7. **Write** changes and **Quit vi (esc :wq)**
8. Restart sshd service:

`service sshd start`

9. Test your SSH connection 👍

Download and install packages

1. Update repository reference info:

```
apk update
```

2. Install most packages:

```
apk add go git shadow gcc musl-dev ansible tree fping arp-scan pwgen bash-completion logrotate
iperf3 atop psmisc tmux net-tools wget nmap bind-tools sysstat iotop tcpdump strace lsof ncdu iftop
htop curl vim ca-certificates rsync tar unzip sudo iptables bash libc6-compat podman
```

© Template Conversion

- Ensure that you are satisfied with the containers operation.
- **Test SSH** Access to both the Root and new sudo privileged user.
- Once satisfied, convert the container into a template.
  - do this by right clicking on the container in the left hand column of the pve list and choosing **convert to template**.
- Be patient and allow for the process to proceed uninterrupted.
- Once the template is created, it is trivial to make copies either by **GUI**, **TUI** or **CLI**

📦 Full APK Package List

Package Name	Description
ansible	Automation tool for configuration management tasks
arp-scan	Scans IP addresses via ARP protocol
atop	Performance monitoring tool for system processes
bash	Essential command-line shell for Linux
bash-completion	Autocompletion functionality for bash commands
bind-tools	DNS query tools (e.g., dig)
ca-certificates	SSL certificate authority files for security
curl	Transfers data from URLs using commands
fping	Ping multiple hosts over IP network
gcc	GNU C Compiler for building software
git	Distributed version control system for code
htop	Interactive process viewer for system monitoring
iftop	Real-time network bandwidth usage monitor
iperf3	Measures network performance and bandwidth usage
iptables	Command-line firewall utility for packet filtering
iotop	Monitors disk I/O usage in real-time
libc6-compat	Compatibility libraries for glibc on musl
logrotate	Automatically rotates and compresses system logs
lsof	Lists open files and associated processes
musl-dev	Musl development libraries for C programming
ncdu	Disk usage analyzer with text interface
net-tools	Legacy networking tools (e.g., ifconfig)
nmap	Network scanner and mapper for security
podman	Manages Linux containers without needing Docker
psmisc	Utilities like killall, fuser, and more
pwgen	Generates random passwords from command-line tool
rsync	File copying and synchronization tool for networks
shadow	Manages encrypted user account passwords securely
strace	Traces system calls and signals for debugging
sudo	Executes commands as root or superuser
sysstat	Performance monitoring tools for CPU, disk, memory
tar	Archiving utility for file compression and packaging
tcpdump	Analyzes and captures network traffic packets
tmux	Terminal multiplexer for managing multiple sessions
tree	Displays directories and files in tree format
unzip	Extracts files from ZIP format archives
vim	Powerful, customizable text editor for developers
wget	Command-line downloader for HTTP/FTP protocols