LXC Post-Install: User-Setup & SSH Hardening

Dieses Post-Install-Skript wird als `root` in einem frischen LXC-Container ausgeführt. Es automatisiert die grundlegende Einrichtung und Absicherung des Systems. Folgende Schritte werden durchgeführt:

- Erstellung eines neuen Standard-Benutzers.
- Zuweisung zur `sudo`-Gruppe und Einrichtung von passwortlosem `sudo`.
- Hinterlegung des öffentlichen SSH-Schlüssels (Public Key) für den neuen Benutzer.
- Entfernung aller hinterlegten SSH-Schlüssel für den `root`-Benutzer.
- Härtung der `/etc/ssh/sshd_config` (Deaktivierung von Root-Login und Passwort-Authentifizierung).

---

1. Vorbereitung (Host-System)

Lege das Post-Install-Skript (z.B. `post_install_ssh.sh`) und deinen Public Key (z.B. `ADMIN.pub`) in einem gemeinsamen Verzeichnis auf deinem Rechner ab.

Wechsle im Terminal in dieses Verzeichnis und starte einen temporären Python-Webserver, um die Dateien im Netzwerk bereitzustellen:

```
python -m http.server 8080
```

> **Hinweis:** Notiere dir die IP-Adresse dieses Rechners (im folgenden Beispiel `10.10.10.10`), da diese im nächsten Schritt benötigt wird.

---

2. Ausführung (LXC-Container)

Verbinde dich als `root` mit dem neuen LXC-Container.

Führe den folgenden Befehl aus. Er lädt das Skript herunter, macht es ausführbar und übergibt den Inhalt deines Public Keys direkt als Argument an das Skript.

(Passe die IP-Adresse `10.10.10.10` entsprechend an!)

```
curl -fsS
[http://10.10.10.10:8080/post_install_ssh.sh](http://10.10.10.10:8080/post_install_ssh.sh) -o /tmp/bootstrap.sh \
  && chmod +x /tmp/bootstrap.sh \
  && /tmp/bootstrap.sh "$(curl -fsS
[http://10.10.10.10:8080/ADMIN.pub](http://10.10.10.10:8080/ADMIN.pub))
"
```

---

3. Das Post-Install-Skript (`post_install_ssh.sh`)

> **Wichtig:** Passe den Wert `USER_NAME="ADMIN"` an deinen gewünschten Benutzernamen an, bevor du das Skript über den Webserver bereitstellst.

```
#!/usr/bin/env bash
set -euo pipefail

USER_NAME="ADMIN"
PUBKEY="${1:-}"
SSHD_CONFIG="/etc/ssh/sshd_config"

echo "==== SSH Bootstrap Starting ===="

########################################
# Ensure user exists
########################################
if id "${USER_NAME}" &>/dev/null; then
    echo "User ${USER_NAME} already exists."
else
    echo "Creating user ${USER_NAME}..."
    useradd -m -s /bin/bash "${USER_NAME}"
fi
```

```
#########################################
# Ensure sudo privileges
#########################################
if ! id -nG "${USER_NAME}" | grep -qw sudo; then
    echo "Adding ${USER_NAME} to sudo group..."
    usermod -aG sudo "${USER_NAME}"
fi

#########################################
# Configure passwordless sudo
#########################################
echo "Installing sudo if it is not installed"
apt install sudo -y -q

SUDO_FILE="/etc/sudoers.d/${USER_NAME}"

echo "Configuring passwordless sudo for ${USER_NAME}..."
echo "${USER_NAME} ALL=(ALL) NOPASSWD: ALL" > "${SUDO_FILE}"
chmod 440 "${SUDO_FILE}"

# Validate sudoers syntax before proceeding
visudo -cf "${SUDO_FILE}"

#########################################
# Configure SSH key
#########################################
SSH_DIR="/home/${USER_NAME}/.ssh"
AUTHORIZED_KEYS="${SSH_DIR}/authorized_keys"

mkdir -p "${SSH_DIR}"
chmod 700 "${SSH_DIR}"
touch "${AUTHORIZED_KEYS}"
chmod 600 "${AUTHORIZED_KEYS}"
chown -R "${USER_NAME}:${USER_NAME}" "${SSH_DIR}"

if [[ -n "${PUBKEY}" ]]; then
    if ! grep -qxF "${PUBKEY}" "${AUTHORIZED_KEYS}"; then
        echo "Installing public key..."
        echo "${PUBKEY}" >> "${AUTHORIZED_KEYS}"
    fi
else
    echo "WARNING: No public key supplied."
fi

#########################################
# Remove all root SSH authorized keys
#########################################
```

```
ROOT_SSH_DIR="/root/.ssh"
ROOT_AUTH_KEYS="${ROOT_SSH_DIR}/authorized_keys"
ROOT_AUTH_KEYS2="${ROOT_SSH_DIR}/authorized_keys2"

echo "Removing root SSH authorized keys..."

if [[ -d "${ROOT_SSH_DIR}" ]]; then
    rm -f "${ROOT_AUTH_KEYS}" "${ROOT_AUTH_KEYS2}"
    chmod 700 "${ROOT_SSH_DIR}" 2>/dev/null || true
    chown root:root "${ROOT_SSH_DIR}" 2>/dev/null || true
    echo "Root authorized_keys removed."
else
    echo "No root .ssh directory present."
fi

########################################
# Backup sshd_config
########################################
BACKUP="${SSHD_CONFIG}.$(date +%F-%H%M%S).bak"
cp "${SSHD_CONFIG}" "${BACKUP}"
echo "Backup created at ${BACKUP}"

########################################
# Idempotent config helper
########################################
set_config() {
    local key="$1"
    local value="$2"

    if grep -qiE "^\s*#?\s*${key}\b" "${SSHD_CONFIG}"; then
        sed -ri "s|^\s*#?\s*${key}\b.*|${key} ${value}|I"
"${SSHD_CONFIG}"
    else
        echo "${key} ${value}" >> "${SSHD_CONFIG}"
    fi
}

########################################
# Apply SSH hardening
########################################
echo "Applying SSH hardening..."

set_config "PermitRootLogin" "no"
set_config "PasswordAuthentication" "no"
set_config "KbdInteractiveAuthentication" "no"
set_config "ChallengeResponseAuthentication" "no"
set_config "PubkeyAuthentication" "yes"
```

```
set_config "AllowUsers" "${USER_NAME}"

########################################
# Validate before restart
########################################
echo "Validating sshd config..."
sshd -t

########################################
# Restart SSH
########################################
echo "Restarting SSH service..."
if command -v systemctl &>/dev/null; then
    systemctl restart sshd 2>/dev/null || systemctl restart ssh
else
    service ssh restart
fi

echo "==== SSH Hardening Complete ===="
echo "? Root SSH disabled"
echo "? Password auth disabled"
echo "? Only ${USER_NAME} allowed"
```

---

4. Beispiel-Ausgabe

Bei einer erfolgreichen Ausführung sollte die Ausgabe im Terminal deines LXC-Containers in etwa so aussehen:

```
==== SSH Bootstrap Starting ====
Creating user ADMIN...
Adding ADMIN to sudo group...
Installing sudo if it is not installed
Reading package lists...
Building dependency tree...
Reading state information...
sudo is already the newest version (1.9.16p2-3+deb13u1).
Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 0
Configuring passwordless sudo for ADMIN...
/etc/sudoers.d/ADMIN: parsed OK
Installing public key...
Removing root SSH authorized keys...
Root authorized_keys removed.
Backup created at /etc/ssh/sshd_config.2026-03-18-191349.bak
```

```
Applying SSH hardening...
Validating sshd config...
Restarting SSH service...
==== SSH Hardening Complete ====
? Root SSH disabled
? Password auth disabled
? Only ADMIN allowed
```

---

5. Login nach der Einrichtung

Sobald das Skript erfolgreich durchgelaufen ist, wurde der `root`-Login über SSH deaktiviert. Du kannst dich nun von deinem Client-Rechner aus (auf dem der passende Private Key liegt) sicher mit dem neu erstellten Benutzer anmelden:

```
ssh ADMIN@<IP-ADRESSE-DES-LXC>
```

Da das Skript passwortloses `sudo` eingerichtet hat, kannst du administrative Befehle nun einfach mit vorangestelltem `sudo` ausführen, ohne ein Passwort eingeben zu müssen (z.B. `sudo apt update`).

Revision #2

Created 2026-03-18 19:15:31 UTC by Carsten

Updated 2026-03-18 19:29:49 UTC by Carsten