Cloud-init

Cloud-Init für Ubuntu-Server:

neu mit ranger, atuin, lazydocker und btop++

Public SSH-Key selbst eintragen!

```
#cloud-config

# Hostname und Zeitzone
hostname: hz-01
timezone: Europe/Berlin

# System-Updates und Paketquellen
package_update: true
package_upgrade: true
package_reboot_if_required: true

# Zu installierende Pakete
packages:
  - apt-transport-https
  - ca-certificates
  - curl
  - gnupg
  - lsb-release
  - fail2ban
  - unattended-upgrades
  - vim
  - htop
  - ranger
  - net-tools
  - git
  - wget
  - build-essential

# Root-User SSH-Konfiguration
ssh_authorized_keys:
```

```
  - DEIN_SSH_PUBLIC_KEY_HIER

# SSH-Konfiguration - Root-Login per Key erlauben, Password auth deaktivieren
ssh_pwauth: false
disable_root: false

# SSH-Config anpassen für Root-Login nur mit Key
write_files:
  - path: /etc/ssh/sshd_config.d/99-custom.conf
    content: |
      PermitRootLogin prohibit-password
      PasswordAuthentication no
      PubkeyAuthentication yes
      ChallengeResponseAuthentication no

  - path: /etc/fail2ban/jail.local
    content: |
      [DEFAULT]
      bantime = 3600
      findtime = 600
      maxretry = 5

      [sshd]
      enabled = true
      port = 22
      logpath = /var/log/auth.log

  - path: /etc/motd
    content: |
      ╔═══════════════════════════════════════════════╗
      ║   Docker Server hz-01 Firma XYZ               ║
      ║   Portainer: https://<IP>:9443                ║
      ║                                               ║
      ║                                               ║
      ║    btop++, lazydocker (lzd), atuin, ranger    ║
      ╚═══════════════════════════════════════════════╝

# Docker und Portainer installieren
runcmd:
  # Docker GPG-Key und Repository hinzufügen
  - mkdir -p /etc/apt/keyrings
```

```
  - curl -fsSL https://download.docker.com/linux/ubuntu/gpg | gpg --dearmor -o
/etc/apt/keyrings/docker.gpg
  - echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg]
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | tee
/etc/apt/sources.list.d/docker.list > /dev/null

  # Docker installieren
  - apt-get update
  - apt-get install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-
compose-plugin

  # Docker-Daemon starten und aktivieren
  - systemctl enable docker
  - systemctl start docker

  # Portainer Volume und Container erstellen
  - docker volume create portainer_data
  - docker run -d -p 9000:9000 -p 9443:9443 --name portainer --restart=always -v
/var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data portainer/portainer-
ce:latest

  # Fail2ban konfigurieren
  - systemctl enable fail2ban
  - systemctl start fail2ban

  # Unattended-Upgrades konfigurieren
  - echo 'Unattended-Upgrade::Automatic-Reboot "false";' >> /etc/apt/apt.conf.d/50unattended-
upgrades
  - echo 'Unattended-Upgrade::Remove-Unused-Kernel-Packages "true";' >>
/etc/apt/apt.conf.d/50unattended-upgrades
  - echo 'Unattended-Upgrade::Remove-Unused-Dependencies "true";' >>
/etc/apt/apt.conf.d/50unattended-upgrades
  - systemctl enable unattended-upgrades
  - systemctl start unattended-upgrades

  # Docker-Logs rotieren (verhindert Speicherprobleme)
  - |
    cat > /etc/docker/daemon.json <<EOF
    {
      "log-driver": "json-file",
```

```
      "log-opts": {
        "max-size": "10m",
        "max-file": "3"
      }
    }
    EOF
  - systemctl restart docker

  # Kernel-Hardening (Sysctl)
  - |
    cat >> /etc/sysctl.conf <<EOF
    # IP-Forwarding für Docker
    net.ipv4.ip_forward = 1
    # Schutz gegen SYN-Flood-Attacken
    net.ipv4.tcp_syncookies = 1
    # Ignoriere ICMP-Redirects
    net.ipv4.conf.all.accept_redirects = 0
    net.ipv6.conf.all.accept_redirects = 0
    # Source-Route-Pakete ablehnen
    net.ipv4.conf.all.accept_source_route = 0
    net.ipv6.conf.all.accept_source_route = 0
    EOF
  - sysctl -p

  # btop++ installieren
  - |
    BTOP_VERSION=$(curl -s https://api.github.com/repos/aristocratos/btop/releases/latest |
grep -oP '"tag_name": "v\K[^"]+')
    wget -q https://github.com/aristocratos/btop/releases/download/v${BTOP_VERSION}/btop-
x86_64-linux-musl.tbz -O /tmp/btop.tbz
    tar -xjf /tmp/btop.tbz -C /tmp
    cd /tmp/btop && make install
    rm -rf /tmp/btop /tmp/btop.tbz

  # lazydocker installieren
  - |
    LAZYDOCKER_VERSION=$(curl -s
https://api.github.com/repos/jesseduffield/lazydocker/releases/latest | grep -oP '"tag_name":
"v\K[^"]+')
    wget -q
```

```
https://github.com/jesseduffield/lazydocker/releases/download/v${LAZYDOCKER_VERSION}/lazydocke
r_${LAZYDOCKER_VERSION}_Linux_x86_64.tar.gz -O /tmp/lazydocker.tar.gz
    tar -xzf /tmp/lazydocker.tar.gz -C /tmp
    install /tmp/lazydocker /usr/local/bin/
    rm /tmp/lazydocker*

  # atuin für alle Benutzer installieren
  - curl --proto '=https' --tlsv1.2 -LsSf https://setup.atuin.sh | sh

  # atuin für admin-User konfigurieren
  - sudo -u admin bash -c 'curl --proto "=https" --tlsv1.2 -LsSf https://setup.atuin.sh | sh'
  - sudo -u admin bash -c 'echo "eval \"\$(atuin init bash)\"" >> ~/.bashrc'

  # Aliases für admin-User hinzufügen
  - |
    sudo -u admin bash -c 'cat >> ~/.bashrc <<EOF

    # Custom Aliases
    alias lzd="lazydocker"
    alias dc="docker compose"
    alias dps="docker ps"
    alias dlog="docker logs"
    EOF'

# Finaler Reboot nach Abschluss (optional)
power_state:
  mode: reboot
  timeout: 300
  condition: true
```

alt:

```
#cloud-config
users:
  - name: pl-admin
    groups: users, admin
    sudo: ALL=(ALL) NOPASSWD:ALL
    shell: /bin/bash
```

```
    ssh_authorized_keys:
      - <public_ssh_key>
package_update: true
package_upgrade: true
packages:
  - fail2ban
  #- ufw
  #Docker:
  - apt-transport-https
  - ca-certificates
  - curl
  - gnupg-agent
  - software-properties-common
runcmd:
  - printf "[sshd]\nenabled = true\nbanaction = iptables-multiport" > /etc/fail2ban/jail.local
  - systemctl enable fail2ban
  #- ufw allow ssh
  #- ufw enable
  - sed -i -e '/^\(#\|\)PermitRootLogin/s/^.*$/PermitRootLogin no/' /etc/ssh/sshd_config
  - sed -i -e '/^\(#\|\)PasswordAuthentication/s/^.*$/PasswordAuthentication no/'
/etc/ssh/sshd_config
  - sed -i -e '/^\(#\|\)KbdInteractiveAuthentication/s/^.*$/KbdInteractiveAuthentication no/'
/etc/ssh/sshd_config
  - sed -i -e '/^\(#\|\)ChallengeResponseAuthentication/s/^.*$/ChallengeResponseAuthentication
no/' /etc/ssh/sshd_config
  - sed -i -e '/^\(#\|\)MaxAuthTries/s/^.*$/MaxAuthTries 2/' /etc/ssh/sshd_config
  - sed -i -e '/^\(#\|\)AllowTcpForwarding/s/^.*$/AllowTcpForwarding no/' /etc/ssh/sshd_config
  - sed -i -e '/^\(#\|\)X11Forwarding/s/^.*$/X11Forwarding no/' /etc/ssh/sshd_config
  - sed -i -e '/^\(#\|\)AllowAgentForwarding/s/^.*$/AllowAgentForwarding no/'
/etc/ssh/sshd_config
  - sed -i -e '/^\(#\|\)AuthorizedKeysFile/s/^.*$/AuthorizedKeysFile .ssh\/authorized_keys/'
/etc/ssh/sshd_config
  - sed -i '$a AllowUsers pl-admin' /etc/ssh/sshd_config
  # Docker:
  - install -m 0755 -d /etc/apt/keyrings
  - curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
  - chmod a+r /etc/apt/keyrings/docker.asc
  - echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/ubuntu \
```

```
  $(. /etc/os-release && echo "${UBUNTU_CODENAME:-$VERSION_CODENAME}") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
  - apt-get update
  - apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-
plugin
  - docker volume create portainer_data
  - docker run -d -p 8000:8000 -p 9443:9443 --name portainer --restart=always -v
/var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data portainer/portainer-ce:lts
  - reboot
```

Revision #4
Created 2025-03-11 14:48:08 UTC
Updated 2026-02-08 15:58:25 UTC