

Installationsprotokoll

System aktualisieren

```
sudo apt-get update  
sudo apt-get upgrade
```

NTP installieren

```
sudo apt-get install ntp
```

Docker und Docker Registry Tools installieren

Abhängigkeiten installieren

```
sudo apt-get install ca-certificates curl gnupg  
sudo install -m 0755 -d /etc/apt/keyrings
```

Docker GPG-Schlüssel hinzufügen

```
curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor  
-o /etc/apt/keyrings/docker.gpg  
sudo chmod a+r /etc/apt/keyrings/docker.gpg
```

Docker Repository hinzufügen

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

Docker installieren

```
sudo apt-get update  
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-  
plugin
```

Docker testen

```
sudo docker run hello-world
```

GitLab Runner für CI/CD installieren

GitLab Runner Repository hinzufügen

```
curl -L
```

```
"https://packages.gitlab.com/install/repositories/runner/gitlab-runner/script.deb.sh" | sudo bash
```

GitLab Runner aktualisieren und installieren

```
sudo apt-get update  
sudo apt-get install gitlab-runner
```

Benutzer zur Docker-Gruppe hinzufügen

```
sudo usermod -aG docker gitlab-runner
```

GitLab Runner registrieren

```
sudo gitlab-runner register --url https://gitlab.rafisa.test --token glrt-  
*****
```

Docker-Images bereinigen

Cleanup Script erstellen

```
nano /home/gitlab-runner/docker_cleanup.sh
```

Cleanup Script

```
#!/bin/bash  
  
# Warnung ausgeben  
echo "WARNUNG: Dies wird ALLE Docker-Images auf diesem System entfernen."  
  
# Alle Docker Images auflisten und deren IDs  
IMAGE_IDS=$(docker images -q)  
  
# Überprüfen, ob Image-IDs vorhanden sind  
if [ -z "$IMAGE_IDS" ]; then  
    echo "Keine Docker-Images zum Entfernen gefunden."  
else  
    # Alle gefundenen Images entfernen  
    echo "Entferne alle Docker-Images..."  
    docker rmi -f $IMAGE_IDS  
    echo "Alle Docker-Images wurden entfernt."  
fi
```

Cleanup Script ausführbar machen

```
chmod +x /home/gitlab-runner/docker_cleanup.sh
```

Cleanup Script zu Crontab hinzufügen

```
crontab -e
```

Crontab Konfiguration

```
0 1 1 * * /home/gitlab-runner/docker_cleanup.sh >/dev/null 2>&1
```

Executor und Image konfigurieren

```
sudo nano /etc/gitlab-runner/config.toml
```

Konfiguration für GitLab Runner

```
concurrent = 5

[[runners]]
  name = "docker-runner"
  url = "https://gitlab.rafisa.test"
  token = "glrt-*****"
  executor = "docker"
  [runners.docker]
    tls_verify = false
    image = "alpine:latest"
    privileged = false
    disable_entrypoint_overwrite = false
    oom_kill_disable = false
    disable_cache = false
    volumes = ["/cache"]
    shm_size = 0
```

From:
<https://wiki.rafisa.net/> -

Permanent link:
<https://wiki.rafisa.net/doku.php?id=de:infrastructure:server:deployment-server:installation-protocol>

Last update: **2024/04/25 13:02**

