gitlab.local.dev

Author: Adrian DC

Description: Install, configure and maintain the gitlab.local.dev infrastructure with mise tasks

Date: 23/11/2026

Version: `2.0.0-43-g11c8590`

▸ 1. GitLab - Ubuntu

Installation

Ubuntu Server 24.04

- **Welcome :**
- **Language :** English (UK)
- **Action :** Enter
- **Installer update available (Optional) :**
- **Action :** Continue without updating
- **Keyboard configuration :**
- **Layout :** French
- **Variant :** French
- **Action :** Done
- **Choose type of install :**
- **Base for the installation :** Ubuntu Server (minimized)
- **Action :** Done
- **Network connections :**
- **Second interface :** Edit IPv4, IPv4 Method: Manual, Subnet, Address, Save
  - **Example for host machines :** `192.168.0.0/24`, `192.168.0.42`, `[EMPTY]`, `[EMPTY]`, `[EMPTY]`
  - **Example for GitLab Server :** `10.16.0.0/24`, `10.16.0.20`, `[EMPTY]`, `[EMPTY]`, `[EMPTY]`
  - **Example for GitLab Runner :** `10.16.0.0/24`, `10.16.0.21`, `[EMPTY]`, `[EMPTY]`, `[EMPTY]`
- **Action :** Done or Continue without network
- **Configure proxy :**
- **Action :** Done
- **Configure Ubuntu archive mirror :**
- **Action :** Done, Continue if asked
- **Guided storage configuration :**
- **Storage layout :** Use an entire disk
- **Set up this disk as an LVM group :** Enable
- **Encrypt the LVM group with LUKS :** Disable (Enable if encryption required)
  - **Passphrase :** `...` (If encryption required)
  - **Confirm passphrase :** `...` (If encryption required)
  - **Also create a recovery key :** Disable

- **Action :** Done
- **Storage configuration :**
  - **Used devices/ubuntu-lv (Optional) :**
  - **Action :** Edit
  - **Name :** `ubuntu-lv-HOSTNAME` (Adapt with the machine's name)
  - **Size :** Maximum size (shown in `(max ...)`, , ,
  - **Action :** Save
  - **Action :** Done, Continue
- **Profile setup :**
- **Your name :** `Administrator`
- **Your server's name :** `HOSTNAME`
- **Pick a username :** `administrator`
- **Choose a password :** `...`
- **Confirm your password :** `...`
- **Action :** Done
- **Upgrade to Ubuntu Pro :**
- **Skip for now :** Enable
- **Action :** Continue
- **SSH Setup :**
- **Install OpenSSH server :** Enable
- **Action :** Done
- **Featured Server Snaps :**
- **Action :** Done
- **Install complete! :**
- **Action :** Reboot Now
- **Please remove the installation medium :**
- **Action :** Remove installation medium, Enter

Table of contents

- Installation
  - Ubuntu Server 24.04
- Network access
- System preparation

- APT
  - Configurations
  - Upgrades
  - Packages
  - Automated cleanups
  - Resize filesystem
  - Configure timezone
- Services
  - Disable suspend features
  - SSH
  - NetworkManager
- Optimizations
  - Swappiness
  - ZRAM
  - RAM temporary filesystem
  - Disable Swap RAM space
- Security
  - Firewall
- Applications
  - Docker
- System
  - Reboot

Network access

- **Connect Internet access :** Bridged Interface or Host-only Adapter
- **Access machine remotely through SSH :** `ssh administrator@SPECIFIC_IP`

System preparation

- **Prepare the system with all recommended tasks below (optional ones excluded) :**

```
gitlab.local.dev system:install
```

Task sources: `system:install`

```
#!/usr/bin/env bash

# Mise
#MISE description="Prepare the Ubuntu system (all recommended system tasks, optional ones excluded)"

# Configurations
set -e -o pipefail

# Action
for task in \
  system:apt:configure \
  system:apt:upgrade \
  system:apt:packages \
  system:timezone \
  system:suspend:disable \
  system:ssh:enable \
  system:networkmanager:install \
  system:swappiness \
  system:tmpfs \
  system:firewall \
  system:docker:install; do
  mise run -q "${task}"
done
```

APT

Configurations

- **Disable the kernel restart hints :**

```
gitlab.local.dev system:apt:configure
```

Task sources: `system:apt:configure`

```
#!/usr/bin/env bash

# Mise
#MISE description="Configure APT to skip the kernel restart hints"

# Configurations
set -e -o pipefail

# Sources
# shellcheck source=lib/common.sh
source "${MISE_PROJECT_ROOT:?}/lib/common.sh"

# Action
step "Configure 'needrestart' kernel hints"
sudo mkdir -p "${SYSROOT}/etc/needrestart/conf.d"
echo '$nrconf{kernelhints} = 0;' | sudo tee "${SYSROOT}/etc/needrestart/conf.d/kernel_silent.conf"
```

Upgrades

- **Upgrade the system packages :**

```
gitlab.local.dev system:apt:upgrade
```

Task sources: `system:apt:upgrade`

```
#!/usr/bin/env bash

# Mise
#MISE description="Upgrade the APT packages"

# Configurations
set -e -o pipefail

# Sources
# shellcheck source=lib/common.sh
source "${MISE_PROJECT_ROOT:?}/lib/common.sh"

# Action
step 'Upgrade APT packages'
sudo apt-get update
sudo NEEDRESTART_MODE=a apt-get upgrade -y
```

Packages

- **Install the common packages :**

```
gitlab.local.dev system:apt:packages
```

Task sources: `system:apt:packages`

```
#!/usr/bin/env bash

# Mise
#MISE description="Install the common APT packages"

# Configurations
set -e -o pipefail

# Sources
# shellcheck source=lib/common.sh
source "${MISE_PROJECT_ROOT:?}/lib/common.sh"

# Action
step 'Install common APT packages'
apt_install \
  avahi-utils \
  byobu \
  dos2unix \
  entr \
  ethtool \
  htop \
  hwinfo \
  ifmetric \
  iftop \
  inotify-tools \
  iotop \
  keychain \
  make \
  nano \
  openssh-server \
  pipx \
  pv \
  python-is-python3 \
  python3-pip \
  rsync \
  screenfetch \
  tig \
  traceroute
```

Automated cleanups

- **Upgrade and cleanup the packages and journal logs :**

```
gitlab.local.dev system:apt:cleanup
```

Task sources: `system:apt:cleanup`

```
#!/usr/bin/env bash

# Mise
#MISE description="Upgrade and cleanup the APT packages and journal logs"

# Configurations
set -e -o pipefail

# Sources
# shellcheck source=lib/common.sh
source "${MISE_PROJECT_ROOT:?}/lib/common.sh"

# Action
step 'Cleanup APT packages and journal logs'
sudo apt-get update
sudo NEEDRESTART_MODE=a apt-get upgrade -y
sudo NEEDRESTART_MODE=a apt-get autoremove -y
sudo apt-get autoclean
sudo journalctl --vacuum-size=50M
```

Resize filesystem

- **Resize the LVM volume and filesystem to the full disk :**

```
gitlab.local.dev system:filesystem:resize
```

Task sources: `system:filesystem:resize`

```
#!/usr/bin/env bash

# Mise
#MISE description="Resize the LVM logical volume and filesystem to the full disk"

# Configurations
set -e -o pipefail

# Sources
# shellcheck source=lib/common.sh
source "${MISE_PROJECT_ROOT:?}/lib/common.sh"

# Action
step 'Resize LVM logical volume and filesystem'
for volume in /dev/mapper/ubuntu--vg-*; do
  sudo lvresize -v -l +100%FREE "${volume}" || true
  sudo resize2fs -p "${volume}"
done
```

Configure timezone

- **Configure the system timezone (asked, default: `Europe/Paris`) :**

```
gitlab.local.dev system:timezone
```

Task sources: `system:timezone`

```
#!/usr/bin/env bash

# Mise
#MISE description="Configure the system timezone"
#MISE arg "[timezone]" help="System timezone (asked if missing, default: Europe/Paris)"

# Configurations
set -e -o pipefail

# Sources
# shellcheck source=lib/common.sh
source "${MISE_PROJECT_ROOT:?}/lib/common.sh"

# Action
step 'Configure the system timezone'
timezone="${usage_timezone:-$(input 'Local timezone' 'Europe/Paris')}"
sudo timedatectl set-timezone "${timezone:?}"
timedatectl
```

Services

Disable suspend features

- **Disable the suspend features :**

```
gitlab.local.dev system:suspend:disable
```

Task sources: `system:suspend:disable`

```
#!/usr/bin/env bash

# Mise
#MISE description="Disable the system suspend features"

# Configurations
set -e -o pipefail

# Sources
# shellcheck source=lib/common.sh
source "${MISE_PROJECT_ROOT:?}/lib/common.sh"

# Action
step 'Disable suspend features'
sudo systemctl mask sleep.target suspend.target hibernate.target hybrid-sleep.target
```

SSH

- **Enable the SSH server :**

```
gitlab.local.dev system:ssh:enable
```

Task sources: `system:ssh:enable`

```
#!/usr/bin/env bash

# Mise
#MISE description="Enable the SSH server"

# Configurations
set -e -o pipefail

# Sources
# shellcheck source=lib/common.sh
source "${MISE_PROJECT_ROOT:?}/lib/common.sh"

# Action
step 'Enable SSH server'
sudo systemctl enable --now ssh
```

- **Disable the SSH password authentication :**

Optional

This section is optional, use if needed.

```
gitlab.local.dev system:ssh:passwords:disable
```

Task sources: `system:ssh:passwords:disable`

```
#!/usr/bin/env bash

# Mise
#MISE description="Disable the SSH password authentication (optional)"

# Configurations
set -e -o pipefail

# Sources
# shellcheck source=lib/common.sh
source "${MISE_PROJECT_ROOT:?}/lib/common.sh"

# Action
step 'Disable SSH password authentication'
sudo mkdir -p "${SYSROOT}/etc/ssh/sshd_config.d"
printf '%s\n' 'PubkeyAuthentication yes' 'PasswordAuthentication no' \
  | sudo tee "${SYSROOT}/etc/ssh/sshd_config.d/50-password-authentication.conf"
sudo systemctl restart ssh
```

NetworkManager

- **Install NetworkManager and manage all network interfaces :**

```
gitlab.local.dev system:networkmanager:install
```

Task sources: `system:networkmanager:install`

```
#!/usr/bin/env bash

# Mise
#MISE description="Install NetworkManager and manage all network interfaces"

# Configurations
set -e -o pipefail

# Sources
# shellcheck source=lib/common.sh
source "${MISE_PROJECT_ROOT:?}/lib/common.sh"

# Install NetworkManager
step 'Install NetworkManager'
sudo apt-get update
apt_install network-manager
sudo systemctl enable --now NetworkManager

# Manage network interfaces
step 'Manage network interfaces with NetworkManager'
sudo mkdir -p "${SYSROOT}/etc/NetworkManager/conf.d"
sudo rm -f "${SYSROOT}/etc/NetworkManager/conf.d/10-globally-managed-devices.conf"
sudo touch "${SYSROOT}/etc/NetworkManager/conf.d/10-globally-managed-devices.conf"
sudo systemctl restart NetworkManager
```

Optimizations

Swappiness

- **Decrease the swap usage :**

```
gitlab.local.dev system:swappiness
```

Task sources: `system:swappiness`

```
#!/usr/bin/env bash

# Mise
#MISE description="Decrease the swap usage (swappiness 10)"

# Configurations
set -e -o pipefail

# Sources
# shellcheck source=lib/common.sh
source "${MISE_PROJECT_ROOT:?}/lib/common.sh"

# Action
step 'Configure swappiness'
printf '\n# Decrease swap usage to a more reasonable level\nvm.swappiness = 10\n' \
  | sudo tee "${SYSROOT}/etc/sysctl.d/50-swappiness.conf"
sudo sysctl -p "${SYSROOT}/etc/sysctl.d/50-swappiness.conf"
```

ZRAM

- **Install ZRAM :**

Optional

This section is optional, use if needed.

```
gitlab.local.dev system:zram
```

Task sources: `system:zram`

```
#!/usr/bin/env bash

# Mise
#MISE description="Install the ZRAM compressed swap (optional)"

# Configurations
set -e -o pipefail

# Sources
# shellcheck source=lib/common.sh
source "${MISE_PROJECT_ROOT:?}/lib/common.sh"

# Action
step 'Install ZRAM'
apt_install zram-config
```

RAM temporary filesystem

- **Mount /tmp as a RAM temporary filesystem :**

```
gitlab.local.dev system:tmpfs
```

Task sources: `system:tmpfs`

```
#!/usr/bin/env bash

# Mise
#MISE description="Mount /tmp as a RAM temporary filesystem"

# Configurations
set -e -o pipefail

# Sources
# shellcheck source=lib/common.sh
source "${MISE_PROJECT_ROOT:?}/lib/common.sh"

# Action
step 'Enable RAM temporary filesystem'
sudo cp -v "${SYSROOT}/usr/share/systemd/tmp.mount" "${SYSROOT}/etc/systemd/system/"
sudo systemctl enable tmp.mount
```

Disable Swap RAM space

- **Disable the swap file :**

Optional

This section is optional, use if needed.

```
gitlab.local.dev system:swap:disable
```

Task sources: `system:swap:disable`

```
#!/usr/bin/env bash

# Mise
#MISE description="Disable the swap file (optional)"

# Configurations
set -e -o pipefail

# Sources
# shellcheck source=lib/common.sh
source "${MISE_PROJECT_ROOT:?}/lib/common.sh"

# Action
step 'Disable swap file'
if [ -e "${SYSROOT}/swap.img" ]; then
  sudo swapoff "${SYSROOT}/swap.img"
  sudo rm -fv "${SYSROOT}/swap.img"
fi
sudo sed -i 's|^\(/swap\.img\)|#\1|g' "${SYSROOT}/etc/fstab"
free -h
```

Security

Firewall

- **Install the persistent firewall, allowing ICMP and SSH :**

```
gitlab.local.dev system:firewall
```

Task sources: `system:firewall`

```
#!/usr/bin/env bash

# Mise
#MISE description="Install the persistent firewall and allow ICMP and SSH"

# Configurations
set -e -o pipefail

# Sources
# shellcheck source=lib/common.sh
source "${MISE_PROJECT_ROOT:?}/lib/common.sh"

# Install
step 'Install persistent firewall'
echo 'iptables-persistent iptables-persistent/autosave_v4 boolean false' | sudo debconf-set-selections
echo 'iptables-persistent iptables-persistent/autosave_v6 boolean false' | sudo debconf-set-selections
apt_install iptables-persistent

# Configure (idempotent rules)
step 'Configure firewall rules'
for rule in '-p icmp -j ACCEPT' '-p tcp --dport 22 -s 0/0 -j ACCEPT'; do
  # shellcheck disable=SC2086
  sudo iptables -C INPUT ${rule} 2>/dev/null || sudo iptables -A INPUT ${rule}
done
sudo iptables -L
sudo netfilter-persistent save
sudo netfilter-persistent reload
```

Applications

Docker

- **Install Docker :**

```
gitlab.local.dev system:docker:install
```

Task sources: `system:docker:install`

```
#!/usr/bin/env bash

# Mise
#MISE description="Install Docker from the official Docker repository"

# Configurations
set -e -o pipefail

# Sources
# shellcheck source=lib/common.sh
source "${MISE_PROJECT_ROOT:?}/lib/common.sh"

# Install dependencies
step 'Install Docker'
apt_install \
  apt-transport-https \
  ca-certificates \
  curl \
  gnupg-agent \
  software-properties-common

# Install repository
sudo rm -f "${SYSROOT}/usr/share/keyrings/docker-archive-keyring.gpg"
curl -fsSL https://download.docker.com/linux/ubuntu/gpg \
  | sudo gpg --dearmor -o "${SYSROOT}/usr/share/keyrings/docker-archive-keyring.gpg"
echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg]
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" \
  | sudo tee "${SYSROOT}/etc/apt/sources.list.d/docker.list"

# Install Docker
sudo apt-get update
apt_install \
  docker-ce \
  docker-compose-plugin

# Configure Docker rights (https://docs.docker.com/engine/install/linux-postinstall/)
sudo usermod -aG docker "${USER}"

# Test Docker
sudo docker run --rm --name test alpine:latest cat /etc/os-release
```

System

Reboot

- **Reboot the system :**

```
gitlab.local.dev system:reboot
```

Task sources: `system:reboot`

```
#!/usr/bin/env bash

# Mise
#MISE description="Reboot the system"

# Configurations
set -e -o pipefail

# Sources
# shellcheck source=lib/common.sh
source "${MISE_PROJECT_ROOT:?}/lib/common.sh"

# Action
confirm 'Reboot the system now ?'
sudo reboot
```

gitlab.local.dev