

Quick Reference Guide

Docker on Ubuntu 2026

Complete installation and best practices

Mars 2026

Table of Contents

1. Docker Engine installation (official method)
2. Post-installation configuration
3. Essential Docker commands
4. docker-compose.yml - Structure
5. Environment variables (.env)
6. Docker Volumes
7. Docker Networks
8. Container updates

1. Docker Engine Installation

```
# Remove old versions
sudo apt remove -y docker docker-engine docker.io 2>/dev/null || true

# Prerequisites
sudo apt update
sudo apt install -y ca-certificates curl gnupg

# Docker GPG key
sudo install -m 0755 -d /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg \
  | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg

# Official repository
echo "deb [arch=$(dpkg --print-architecture) \
  signed-by=/etc/apt/keyrings/docker.gpg] \
  https://download.docker.com/linux/ubuntu \
  $(. /etc/os-release && echo $VERSION_CODENAME) stable" \
  | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

# Install
sudo apt update
sudo apt install -y docker-ce docker-ce-cli containerd.io \
  docker-buildx-plugin docker-compose-plugin
docker --version && docker compose version
```

2. Post-Installation

```
# Add to docker group (avoid sudo)
sudo usermod -aG docker $USER
newgrp docker

# Enable at startup
sudo systemctl enable docker
sudo systemctl start docker
docker ps # Test without sudo
```

3. Essential Commands

```
docker ps                # Active containers
docker ps -a            # All containers
docker run -d -p 8080:80 nginx # Start nginx
docker stop / start / restart NAME # Management
docker rm NAME          # Remove
docker exec -it NAME bash # Interactive shell
docker logs -f NAME     # Follow logs
docker stats            # Real-time CPU/RAM
docker images           # Local images
docker pull IMAGE:TAG   # Download image
docker image prune -f   # Cleanup
```

4. docker-compose.yml

```
version: '3.8'

services:
  webapp:
    image: nginx:alpine
    container_name: my-site
    restart: unless-stopped
    ports:
      - "8080:80"
    volumes:
      - ./html:/usr/share/nginx/html:ro
    networks:
      - app-network

  database:
    image: postgres:16-alpine
    environment:
      POSTGRES_PASSWORD: ${DB_PASSWORD}
    volumes:
      - db-data:/var/lib/postgresql/data

volumes:
  db-data:

networks:
  app-network:
```

5. .env Variables

```
# .env file (NEVER commit to git)
DB_PASSWORD=MySecurePassword
APP_KEY=random-secret-key

# Add .env to .gitignore
echo ".env" >> .gitignore
```

6. Volumes

```
# Bind mount (host folder)
- ./data:/var/data

# Named volume (Docker-managed)
volumes:
  my-volume:

# Backup a volume
docker run --rm \
  -v my-volume:/data \
  -v $(pwd):/backup \
  alpine tar czf /backup/backup-$(date +%Y%m%d).tar.gz /data
```

7. Networks

```
# Same compose services see each other
# by service name (e.g., http://database:5432)

docker network ls
docker network create my-network
docker network inspect my-network
```

8. Container Updates

```
cd /path/to/app
docker compose pull      # New images
docker compose up -d    # Redeploy
docker compose ps       # Verify
docker image prune -f   # Cleanup
```

Full article: botum.ca/install-docker-ubuntu/

Website: www.botum.ca -

- Canada