

Setting Up Your Machine

Setting Up Your Machine

Anonymous · 10/10/2026

Setting Up Your Machine

Review what you need to do, install, or know

Configure ZSH on local macOS machine

If you're on macOS and use ZSH as your shell, you can configure ZSH to accept comments in commands with:

```
echo "setopt interactivecomments" >> ~/.zshrc
source ~/.zshrc
```

SSH Key on local machine

It's necessary to have an SSH key on your local machine. This key will be used to configure your Docker container.

With the following command, you'll get an existing key or create a new one in `$HOME/.ssh/id_rsa` and `$HOME/.ssh/id_rsa.pub`. If a new one is created, you'll need to enter your email to identify the key owner. If you prefer, you can enter your username and machine name (in case you have multiple SSH keys in different environments).

```
SSH_KEY="$HOME/.ssh/id_rsa"
if [ -f "$SSH_KEY.pub" ]; then
    cat "$SSH_KEY.pub"
else
    echo -n "Enter your email: "; read COMMENT
    ssh-keygen -t rsa -b 4096 -N "" -C "$COMMENT" -f "$SSH_KEY"
    cat "$SSH_KEY.pub"
fi
```

With this command, the public key will be displayed on screen. If you need to see it again in the future, run:

```
cat $HOME/.ssh/id_rsa.pub
```

Important: The key in `$HOME/.ssh/id_rsa` is **private** and should never be shared.

Add SSH key to GitHub

1. Go to [GitHub](#).
2. In Title, enter the same email or comment you chose when creating the key.
3. In Key, paste the public key extracted with `cat $HOME/.ssh/id_rsa.pub`.
4. Click on Add SSH key.

Docker container with PHP and Caddy

Prepare [Docker](#) in `$DOCKER_DIR` on your local machine:

```
DEV_DIR=$HOME/dev
DOCKER_DIR=$DEV_DIR/docker-sites-php
mkdir -p $DEV_DIR
git clone https://github.com/derafu/docker-php-caddy-server.git $DOCKER_DIR
cat $HOME/.ssh/id_rsa.pub > $DOCKER_DIR/config/ssh/authorized_keys
cd $DOCKER_DIR
```

Copy and edit the `.env` file and configure the environment variables as needed.

```
# It's recommended to at least configure the DEPLOYER_HOST variable.
cp .env-dist .env
```

Warning

Don't proceed until you've reviewed and configured the `.env` file.

Build the Docker container:

```
docker-compose up -d
```

With this configuration, the folder where sites to be developed will be installed will be in `$DOCKER_DIR/sites`. This folder will be shared between your local machine and the Docker container.

Connect to Docker container

Configure the `dev` SSH alias on your local machine:

```
echo "
Host dev
  HostName localhost
```

```
User admin
Port 2222
IdentityFile $HOME/.ssh/id_rsa
StrictHostKeyChecking no
UserKnownHostsFile /dev/null
" >> $HOME/.ssh/config
```

Then you can enter the Docker container with:

```
ssh dev
```

Note: The alias name can be whatever you want; in this case, dev was used.

SSH key in Docker container

To work with private repositories and deploy to the production server, it's necessary that the SSH key from your local machine be added to the Docker container.

On your local machine, run:

```
scp $HOME/.ssh/id_rsa* dev:~/.ssh/
```

Basic Git configuration

Perform the following configurations inside the Docker container.

First, configure your GitHub email and name:

Warning

Before pasting this command in the Docker container, edit it to use your GitHub email and name.

```
git config --global user.email "you@example.com" # Your github email.
git config --global user.name "Your Name"       # Your name.
```

Configure case sensitivity and avoid mixing changes:

```
git config --global core.ignorecase false      # Case sensitive.
git config --global pull.rebase false          # Rebase instead of merge.
git config --global merge.ff false             # Fast forward.
```

Configure text editor:

```
git config --global core.editor nano           # Default editor nano or any other.
```

Configure commit signing in Git

First, create the SSH key on your local machine:

```
SSH_KEY="$HOME/.ssh/id_ed25519"
if [ -f "$SSH_KEY.pub" ]; then
    cat "$SSH_KEY.pub"
else
    echo -n "Enter your email: "; read COMMENT
    ssh-keygen -t ed25519 -N "" -C "$COMMENT" -f "$SSH_KEY"
    cat "$SSH_KEY.pub"
fi
```

Then add the SSH key to the Docker container:

```
scp $HOME/.ssh/id_ed25519* dev:./ssh/
```

Finally, inside the Docker container, configure Git to sign commits:

```
git config --global commit.gpgSign true           # Sign commits.
git config --global user.signingkey ~/.ssh/id_ed25519 # Your ssh key.
git config --global gpg.format ssh                 # Use ssh key.
git config --global tag.gpgSign true               # Sign tags.
```

Last updated on 10/10/2026