
Table of Contents

Introduction	1.1
docker	1.2
~/vimrc	1.2.1
~/bash_aliases	1.2.2

Gitbook

How to Use This Git

Git clone the source

```
git clone git@github.com:newstatusralph/gitbook.git
./gitbook/docker/run.sh
```

Run docker container and install packages and run gitbook service

```
cd /work/book/gitbook/docker
./apt.sh
cd /work/work
./serve.sh
```

Docker

version: 0.2

date: 2021/09/13 11:31am

Docker

docker run container

```
sudo docker run --network host --privileged --name gitbook --hostname gitbook -v $(realpath ../):/work/book -it ubuntu:xenial
```

docker container

```
sudo docker start gitbook
# interaction mode
sudo docker start -i gitbook
```

container bash

```
sudo docker exec -it gitbook bash
```

gitbook_cli

nodejs

```
apt-get update
apt-get install -y curl
curl -sL https://deb.nodesource.com/setup_14.x | bash -
# apt-get update
apt-get install -y nodejs
```

gitbook_cli

```
npm install -g gitbook-cli
```

```
mkdir -p book  
cd book  
gitbook init
```

.vimrc

```
if has("autocmd")
  au BufReadPost * if line("'\"") >0 && line("'\"") <= line("$") | exe "normal! g`\"" |
  endif
endif
set t_te=
set modeline
set modelines=5
au BufRead,BufNewFile *.ino set filetype=cpp
set tags=./tags;/
" set tags+=/work/contrend/builder.001/build/trace/user/tags
set tags+=/work/contrend/tags
"set ts=4 sw=4 et
set ts=4 sw=4
```

.bash_aliases

```
export LANG=C.UTF-8
gitb(){
    git branch -a
}
gitl(){
    git log --graph --oneline --decorate $@
}
gitla(){
    git log --graph --oneline --decorate --all $@
}
gitlb(){
#   git log --graph --oneline --decorate --branches="$1"
    git log --decorate --pretty="%C(yellow)%h %C(green)%d %n %C(white) %ad %an: %s " --date=short ${1:---branches="*${1}*"}
}
gits(){
    git status -s $@
}
giturl(){
    echo $(cd ${2:-.}/; git remote get-url ${1:-origin})
}
```