
poseidon Documentation

liuwenjun

Jun 12, 2020

Contents:

1	vscode	1
1.1	SSH	1
1.2	ReadTheDoc	2
1.3		2
2	reStructuredText	3
2.1		3
2.2	vscode reStructuredText	3
3	vboxarc linux	5
3.1	enp0s8ip	5
3.2	zsh.bash_profile	5
4	Node	7
4.1	stream	7
4.2	node-addon-api	7
4.3	npm	7
5	Node	9
5.1	PM2	9
6	TypeOrm SQL	11
6.1	join on	11
6.2	ubuntupgsql	11
6.3		12
7	python	13
7.1	conda	13
8	Poseidon3	15
8.1		15
8.2	flow	15
9	TTS	17
9.1	windows sapi	17
10	docker	19
10.1	docker	19

10.2	docker	19
11	esp32	21
11.1	esp32	21
11.2	vscode	21
11.3		22
12		23
12.1		23
13		25

1.1 SSH

1.1.1

- opensshgdb
- git,vscodeRemote-Development

1.1.2 ssh

git bash

```
1 ssh-keygen //
2 ssh-copy-id -i ~/.ssh/id_rsa.pub  @192.168.x.xxx
3 ssh @192.168.x.xxx //
```

1.1.3

.vscode/setting.json

```
1 {
2   "version": "0.2.0",
3   "configurations": [
4     {
5       "name": "gdb Remote Launch",
6       "type": "cppdbg",
7       "request": "launch",
8       "program": "~/git/hello/a.out",
9       "args": [
10        "myarg1",
```

(continues on next page)

(continued from previous page)

```

11         "myarg2",
12         "myarg3"
13     ],
14     "stopAtEntry": true,
15     "environment": [],
16     "externalConsole": false,
17     "MIMode": "gdb",
18     "miDebuggerPath": "gdb",
19     "miDebuggerArgs": "gdb",
20     "linux": {
21         "MIMode": "gdb",
22         "miDebuggerPath": "/usr/bin/gdb",
23         "miDebuggerServerAddress": "192.168.56.101:2333",
24     },
25     "logging": {
26         "moduleLoad": false,
27         "engineLogging": false,
28         "trace": false
29     },
30     "setupCommands": [
31         {
32             "description": "Enable pretty-printing for gdb",
33             "text": "-enable-pretty-printing",
34             "ignoreFailures": true
35         }
36     ],
37     "cwd": "${workspaceFolder}",
38 }
39 ]
40 }
```

1) VSCode GDB

1.2 ReadTheDoc

miniconda,conda CMD

```

1 conda create -n py37 python=3.7
2 conda activate py37
3 pip config set global.index-url https://pypi.tuna.tsinghua.edu.cn/simple
4 pip install flake8 sphinx sphinx-autobuild sphinx_rtd_theme
```

pythonctrl+shift+p,python:select interpreterpython

- 1)
- 2) pypi
- 3) conda

1.3

CHAPTER 2

reStructuredText

2.1

- 1) rst
- 2) rst
- 3) rst <<https://github.com/buke/openerp-doc/wiki/reStructuredText%E7%AE%80%E6%98%8E%E6%95%99%E7%A8%8B>>‘_#)
‘rst
- 4) rst

2.2 vscodereStructuredText

1. vscode
2. miniconda
3. conda install sphinx sphinx-autobuild sphinx_rtd_theme
4. reStructuredText

3.1 enp0s8ip

hostonlyenp0s8IPIPacmanNetManagerNetworkManager

```
1 pacman -S NetworkManager
2 systemctl enable NetworkManager
3 systemctl start NetworkManager
4 systemctl status NetworkManager
```

3.2 zsh.bash_profile

zsh.bash_profilezsh ~/.zshrc.zshrc

4.1 stream

<https://tech.meituan.com/2016/07/15/stream-internals.html>

4.2 node-addon-api

poseidon3@passoa/cxbnode-addon-apiC++

1. <https://github.com/nodejs/node-addon-api>
2. <https://github.com/pass0a/cxb>
1. <https://github.com/pass0a/cvip>
2. <https://github.com/pass0a/pcan>

4.3 npm

`npm publish --access public`

5.1 PM2

```
1 pm2 start )
2 pm2 save ()
3 pm2 startup ()
4
5 #
6 # [PM2] [v] Command successfully executed.
7 # +-----+
8 # [PM2] Freeze a process list on reboot via:
9 # $ pm2 save
10 # [PM2] Remove init script via:
11 # $ pm2 unstartup systemd
```

```
1 # pm2 startup
2 # [PM2] Init System found: upstart
3 # [PM2] To setup the Startup Script, copy/paste the following command:
4 # sudo env PATH=$PATH:/opt/bitnami/nodejs/bin /opt/bitnami/nodejs/lib/node_modules/
  ↪pm2/bin/pm2 startup upstart -u bitnami --hp /home/bitnami
```

sudo env....

sudo reboot pm2 list

6.1 join on

/// @todo

```
1 select * from version_plan a
2   left join (
3     select max("id") as id, "versionId" from version_history group by "versionId") b
4   on a.id=b."versionId"
5   left join version_history c
6   on c.id=b.id order by c.stage,a.version
```

6.2 ubuntupgsql

1. /etc/postgresql/9.x/mainpostgresql.conflisten_addresses = '*'
2. /etc/postgresql/9.x/mainpg_hba.confhost all all 0.0.0.0/0 md5
- 3.

```
1 sudo -u postgres psql
2 \password
3 \q
4 # sudo /etc/init.d/postgresql start
5 # sudo /etc/init.d/postgresql stop
6 sudo /etc/init.d/postgresql restart
```

6.3

```
1 SELECT * FROM pg_stat_activity WHERE datname='flow' and waiting=true;
```

7.1 conda

<https://mirrors.tuna.tsinghua.edu.cn/help/anaconda/>

7.1.1 conda

```
1 conda update --strict-channel-priority --all
```

7.1.2

vscode+pythonvscodewave.openIDE

```
1 import wave
2 fo = wave.open(u"test.wav", "wb") # type: wave.Wave_write
3 fo.setnchannels(1)
4 fo.setsampwidth(16)
```


8.1

1. ()
2. `git tag v1.0.x`
3. `git push origin v1.0.x`
1. <https://passoa.coding.net/p/repertory/ci/job?id=246128>
- 2.

8.2 flow

```
1 pm2 list #
2 pm2 start dist/main.js # flow
3 # flowrestart/start/stopflow
4 # 0pm2 listid
5 pm2 restart/start/stop 0
```


9.1 windows sapi

windowswin7win10 [https://docs.microsoft.com/en-us/previous-versions/windows/desktop/ee125082\(v=vs.85\)](https://docs.microsoft.com/en-us/previous-versions/windows/desktop/ee125082(v=vs.85))

9.1.1

CHAPTER 10

docker

10.1 docker

```
1 #
2 sudo docker images
3 #
4 sudo docker ps -a
5 #
6 sudo docker rm $(sudo docker ps -aq)
7 #
8 sudo docker stop $(sudo docker ps -q) & sudo docker rm $(sudo docker ps -aq)
9 # ()nsisnsis
10 sudo docker run --name nsis \
11     -it \
12     -v /home/passoa/tmp:/container \
13     flow123d/nsis-3.05-1 \
14     /container/install.nsi
15 # docker
16 sudo docker attach $container_id #container_id docker ps id
```

10.2 docker

systemd /etc/docker/daemon.json

```
1 {
2     "registry-mirrors": ["https://mirror.ccs.tencentyun.com"]
3 }
```

```
1 sudo systemctl daemon-reload
2 sudo systemctl restart docker
```


11.1 esp32

esp32docker *docker* .

```
1 # esp32
2 sudo docker pull espressif/idf
3 #
4 sudo docker run -v $PWD:/home/passoa/git/esp -w /home/passoa/git/esp -it espressif/idf
5 # apt
6 apt-get update
7 # openssh-server
8 apt-get install openssh-server
9 # /etc/ssh/sshd_config
10 # PermitRootLogin yes
11 # UsePAM no
12 # sshd
13 mkdir -p /var/run/sshd
14 /usr/sbin/sshd -D &
15 #
16 sudo docker run --rm -p "2020:22" -v $PWD:/home/passoa/git/esp -w ~/esp -it esp32
17 #
18 sudo docker commit $container_id new_name #container_id docker ps id
```

11.2 vscode

vscodeesp32~/ssh/config

```
1 Host linux_esp
2 HostName 192.168.56.101
3 User passoa
```

(continues on next page)

(continued from previous page)

```

4 Port 22
5 IdentityFile C:/Users/hsae/.ssh/id_rsa
6
7 Host docker_esp
8 HostName 192.168.56.101
9 User root
10 Port 2020
11 IdentityFile C:/Users/hsae/.ssh/id_rsa

```

11.3

11.3.1 1GPG errorNO PUBKEYkey

```
1 apt-key adv --recv-keys --keyserver keyserver.ubuntu.com $PUBKEY
```

11.3.2 2 apt

souces.list/etc/apt/sources.list

```
1 cp ./sources.list /etc/apt/sources.list
```

sources.list

```

1 # apt update
2 deb http://mirrors.tuna.tsinghua.edu.cn/debian/ stretch main contrib non-free
3 # deb-src http://mirrors.tuna.tsinghua.edu.cn/debian/ stretch main contrib non-free
4 deb http://mirrors.tuna.tsinghua.edu.cn/debian/ stretch-updates main contrib non-free
5 # deb-src http://mirrors.tuna.tsinghua.edu.cn/debian/ stretch-updates main contrib
  ↳non-free
6 deb http://mirrors.tuna.tsinghua.edu.cn/debian/ stretch-backports main contrib non-
  ↳free
7 # deb-src http://mirrors.tuna.tsinghua.edu.cn/debian/ stretch-backports main contrib
  ↳non-free
8 deb http://mirrors.tuna.tsinghua.edu.cn/debian-security stretch/updates main contrib
  ↳non-free
9 # deb-src http://mirrors.tuna.tsinghua.edu.cn/debian-security stretch/updates main
  ↳contrib non-free

```

11.3.3 3 openssh-server

openssh-server

```
1 openssh-server : Depends: openssh-client (= 1:7.4p1-10+deb9u7) but 1:7.6p1-4ubuntu0.3
  ↳is to be installed
```

client

```
1 apt-get install openssh-client=1:7.4p1-10+deb9u7
```

12.1

12.1.1

$$m, y = f(x) \quad y = \phi(x)$$

12.1.2

$$\rho(x_i, y_i) \quad i = 1, 2, \dots, m \quad y = \phi(x) \quad \phi(x) = f(x) \quad \rho(x_i, y_i) \quad \delta_i = \phi(x) - y, i = 1, 2, \dots, m$$

CHAPTER 13

- `genindex`
- `modindex`
- `search`