

# Ubuntuのriscv64版をqemu で動かした

Kazuhiro NISHIYAMA

*LILO&東海道らぐオンラインミーティング*

*2024-04-27*

*2024-04-27*

# self.introduction

---

- 西山 和広
- Ruby のコミッター
- github など: @znz
- 株式会社Ruby開発 [www.ruby-dev.jp](http://www.ruby-dev.jp)

# ホスト

---

- Debian GNU/Linux 12 (bookworm)
- qemu-system\* 1:7.2+dfsg-7+deb12u5
- libvirt-daemon 9.0.0-4
- u-boot-qemu 2023.01+dfsg-2
- cloud-image-utils 0.33-1

# ゲストイメージファイル

---

- <https://cloud-images.ubuntu.com> からダウンロード
  - `/${codename}/current/${codename}-server-cloudimg-${arch}.img` でデイリービルド版
- <https://cloud-images.ubuntu.com/releases/> にリリース版
  - 基本デイリービルドでたまたま問題があるビルドだったらリリース版を選ぶぐらいでもいいかも
- (実機用は <https://cdimage.ubuntu.com/releases/>)

# イメージをいい感じにする

```
img=noble-server-cloudimg-riscv64.img  
wget https://cloud-images.ubuntu.com/noble/current/$img  
qemu-img resize "$img" +5G
```

- qcow2 形式

- 速度のため raw に変換するのもあり (qemu-img convert -f qcow2 -O raw "\$orig" "\$img")

- ギリギリのディスクサイズなのでリサイズ

- ちょっと試すなら +5G ぐらい
- もっと使うなら 16G とかに増やす (qemu-img resize "\$img" 16G)

# 最低限の起動確認

```
qemu-system-riscv64 -nographic -M virt -m 1G \  
-kernel /usr/lib/u-boot/qemu-riscv64_smode/u-boot.elf \  
-drive "if=virtio,format=qcow2,file=$img" -snapshot
```

- -machine (-M) は virt で良さそう
- -m は適当に増やす
  - デフォルトの 128 だと起動しなかった
- -kernel は uboot で起動
  - -bios に opensbi の fw\_jump.elf の指定は不要
- -snapshot で書き込みは止めておいた

# ログイン準備

---

```
mkdir config
echo "instance-id: $(uuidgen || echo i-abcdefg)" > config/meta-data
vi config/user-data
cloud-localds "seed.iso" config/user-data config/meta-data
```

- 起動できるが root もパスワードがなくてログインできない
- cloud-init で設定するため ISO ファイル作成
- -drive  
"if=virtio,format=raw,file=seed.iso" を追加して起動

# config/user-data

---

- 詳細は cloud-init のドキュメントを参照
- 例: <https://cloudinit.readthedocs.io/en/latest/reference/examples.html>

```
#cloud-config

hostname: noble-riscv64

# user: ubuntu のパスワード設定
password: ubuntu
chpasswd: { expire: False }
ssh_pwauth: true

# 各種設定
timezone: Asia/Tokyo
locale: ja_JP.utf8

# 自分のssh鍵を設定
ssh_import_id:
  - gh:znz
```



# 起動

```
qemu-system-riscv64 -nographic -M virt -m 2G -smp 4 \  
-kernel /usr/lib/u-boot/qemu-riscv64_smode/u-boot.elf \  
-drive "if=virtio,format=qcow2,file=$img" \  
-drive "if=virtio,format=raw,file=seed.iso" \  
-device "virtio-net-device,netdev=net0" \  
-netdev "user,id=net0,hostfwd=tcp::2222-:22" \  
-device virtio-rng-pci \  
-snapshot
```

- メモリや CPU も増やした
- ネット接続や RNG デバイスも追加
- 「ssh -o "StrictHostKeyChecking no" -p 2222 ubuntu@localhost」でログイン可能

# その他の設定

---

- 今日はここまで
- 後日、ブログ <https://blog.n-z.jp/> に記事を書く予定
  - qemu-guest-agent 対応
  - <https://wiki.qemu.org/Documentation/9psetup> でホストとのファイル共有
  - 自動起動のため libvirt 管理下に移行

# まとめ

---

- cloud-images.ubuntu.com にある amd64, arm64, armhf, ppc64el, riscv64, s390x はどれも同じように使える(はず)
- 最低限の起動までは arch ごとに調査が必要
- cloud-init でログインできる設定が必要
- 他の設定はできるだけシェルスクリプトや ansible などの provisioner を使う方が楽かも
  - cloud-init は試行錯誤しにくい
  - ansible などは知識の流用がしやすい