

# آشنایی با Cloud-init

محمدرضا ملاصالحی<sup>۱</sup>

## به طور کلی Cloud-init چیست؟

مجموعه ای از اسکریپت ها به زبان پایتون برای شخصی سازی روی ماشین های مجازی است.

### قابلیت های Cloud-init

- مشخص کردن Hostname
- ساخت و تنظیم یوزر و گروه ها
- تنظیم SSH Key
- تنظیم پیکربندی شبکه
- اجرای دستور روی Terminal

### مشکلات Cloud-init<sup>۲</sup>

- ریست شدن دیتابیس بعد از هر ریست
- نبود اطلاعات کافی درباره ساخت دیتابیس
- ایجاد مشکلات در سیستم های قدیمی

---

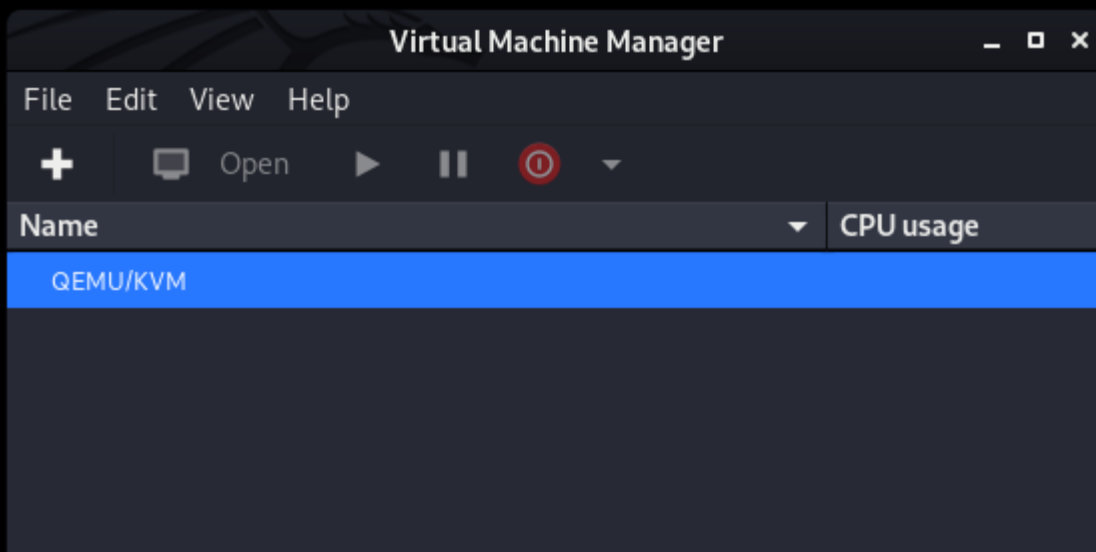
<sup>۲</sup>[https://linuxhint.com/cloud\\_init\\_vms/](https://linuxhint.com/cloud_init_vms/)

## توضیح مرحله به مرحله

اول از نصب بودن QEMU و Virtual-Machine-Manager مطمئن میشیم.

```
(mohammd@kali)-[~]  
$ sudo apt-get install qemu virt-manager  
[sudo] password for mohammd:  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
qemu is already the newest version (1:5.2+dfsg-10).  
virt-manager is already the newest version (1:3.2.0-3).  
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
```

پس از اون مطمئن میشیم که فضای مجازی سازی به QEMU وصل است.



بدیهی است که برای کار کردن ماشین مجازی در BIOS باید Hyper-V رو فعال کنید و برای PCI Passthrough باید iommu رو فعال کرد.

مطلب جدید : Virtual-Machine-Manager و Virtual-Box تفاوت دارند و برای تجهیزات قدیمی Virtual-Box بهتر هست.<sup>۳</sup>

حالا برای Custom-img نیاز به نصب ۲ اپکیشن دیگر داریم.

```
(mohammdreza@kali)-[~]
└─$ sudo apt install libvirt-daemon-system virtinst
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
libvirt-daemon-system is already the newest version (7.0.0-3).
libvirt-daemon-system set to manually installed.
virtinst is already the newest version (1:3.2.0-3).
virtinst set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
```

بعد از آن باید img اولیه رو دانلود کنیم.

```
(mohammdreza@kali)-[~/Desktop/cloud-init-2]
└─$ wget https://cloud-images.ubuntu.com/bionic/current/bionic-server-cloudimg-amd64.img
--2021-06-11 14:05:12-- https://cloud-images.ubuntu.com/bionic/current/bionic-server-cloudimg-amd64.img
Resolving cloud-images.ubuntu.com (cloud-images.ubuntu.com)... 91.189.91.124, 91.189.91.123, 2001:67c:1562::25, ...
Connecting to cloud-images.ubuntu.com (cloud-images.ubuntu.com)|91.189.91.124|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 370737152 (354M) [application/octet-stream]
Saving to: 'bionic-server-cloudimg-amd64.img.1'

bionic-server-cloudimg-amd64.  3%[>] 13.02M 1.56MB/s eta 3m 53s
```

```
(mohammdreza@kali)-[~/Desktop/cloud-init]
└─$ qemu-img info bionic-server-cloudimg-amd64.img
image: bionic-server-cloudimg-amd64.img
file format: qcow2
virtual size: 2.2 GiB (2361393152 bytes)
disk size: 354 MiB
cluster_size: 65536
Format specific information:
    compat: 0.10
    compression type: zlib
    refcount bits: 16
```

تمام فایل ها به فرمت qcow2 هست که حجم کمتر دارند ولی سرعت شان نیز نسبت به raw به کمتر است.<sup>۴</sup>

بعد از آن باید `img` خودمان را از روی `img` اصلی بسازیم. به اصطلاح `snapshot` بگیریم.

```
(mohammedreza@kali)-[~/Desktop/cloud-init]
$ qemu-img create -b bionic-server-cloudimg-amd64.img -f qcow2 -F qcow2 snapshot-bionic-server-cloudimg.qcow2 10G
Formatting 'snapshot-bionic-server-cloudimg.qcow2', fmt=qcow2 cluster_size=65536 extended_l2=off compression_type=zlib
size=10737418240 backing_file=bionic-server-cloudimg-amd64.img backing_fmt=qcow2 lazy_refcounts=off refcount_bits=16
```

برای ساخت `cloud_init.cfg` و SSH کردن به ماشین مجازی ساخته شده نیاز به `SSH public key` البته با نوشتن

رمزتو خود فایل ہم میشه ولی دو تاش بهتره 😊 پس سریع یہ SSH key ہم بسازیم کپی کنیم۔

```

mohammdreza@kali: ~/Desktop/nordvpn
mohammdreza@kali: ~/ssh
mohammdreza@kali: ~/ssh 118x54
(mohammdreza@kali)-[~/ssh]
$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/mohammdreza/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/mohammdreza/.ssh/id_rsa
Your public key has been saved in /home/mohammdreza/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:jXhloCdS8z72eGvfQHWua0hDUybl2/JXQWY8D4TvDwA mohammdreza@kali
The key's randomart image is:
+---[RSA 3072]-----+
|
|  o .  oo
|  . + .E o .*
|  . o o o+ .+.
|  . = . =.+.
|  . S .+. = o
|  o +ooo.+.
|  .+.+.o.o.
|  ooo o. o
|  .oo. ..
+---[SHA256]-----+
(mohammdreza@kali)-[~/ssh]
$ ls
id_rsa id_rsa.pub known_hosts
(mohammdreza@kali)-[~/ssh]
$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgDQDobdBcYDxTg2U5AWrP/a1RWpuEAj/LcvAuds2MYBWYtCqUNdGwEQHy6JELr0iLF+L1vkxKfW27Uxbw
bEhzKvsl/YDAqXajwC0XV7WLogWkJTP7YGAn9qjNTHSm+ZeI6cP2kxbpmEH7HLeWrr9t1/b99b04s38pgueaUFE4oWxHTKVAUiQxowfgiNrQ9K7rxo7P2f
yWEDH0pL6+q5bpgyUro2Ncl5eF5pvwbnHGOY8XpiFYSilasnGh3nBgUPW0jrsKilReYPfy3aN13q+SaF0VtB6bSLMSdNdI6Dr0ssy4icw2Z2c5fi9T3Ep
N/Fkm8/No3/qwzmuqqDgGF9P3JuirFMBaERNQc2kxM2PaIq5CXlNmZlxurKUD8BGpwEi0GAyfW6iMmtS9Qomfirhiw+IMOCmd3H+MoMsLB/3TvmAvSTZ9e
OKXW04/o+kMDqo05QP1kaH6DochEivwhvh/qjWGU379qzWSWczEBXICYe+IYpVknql94pSqMt8BlM4M= mohammdreza@kali

```

<sup>4</sup><https://serverfault.com/questions/677639/which-is-better-image-format-raw-or-qcow2-to-use-as-a-baseimage-for-other-vms>

حالا باید فایل cloud\_init.cfg را ایجاد کنیم.

```
1 #cloud-config
2 hostname: bioniccloud1
3 users:
4   - default
5     - name: test
6       sudo: ALL=(ALL) NOPASSWD:ALL
7       groups: users, admin
8       home: /home/test
9       shell: /bin/bash
10      lock_passwd: false
11      ssh-authorized-keys:
12        - ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgQDQDobdBcYDxTg2U5AWrP/a1RWpuEAj/LcvAUds2MYBWY
13 chpasswd:
14   list: |
15     root:password
16   expire: False
17 packages:
18   - curl
19
20 # run once for setup
21 runcmd:
22   - lscpu >>/tmp/info.json
23   - curl -X POST -d @/tmp/info.json http://192.168.1.10:4444
```

دستور lscpu جزئیات سیستم رو چاپ میکند با سیو کردن خروجی دستور در Temp می توان به فایل Json از دیتا داشته باشیم.

برای ساخت Webhook ساده ما با استفاده از دستور curl و با متد پست دیتا رو به جایی که می خواهیم می فرستیم.

پس این دستورات را در runcmd قرار میدهیم که در آخر init اجرا شود.

در اون طرف هم Listener ساده روی پورت ۴۴۴۴ قرار میدهیم.

```
(mohammdreza@kali)-[~]  
$ nc -lnvp 4444  
Listening on 0.0.0.0 4444  
█
```

قبل از ایجاد ماشین مجازی باید از صحیح بودن cloud\_init.cfg مطمئن شویم.

```
(mohammdreza@kali)-[~/Desktop/cloud-init-2]  
$ cloud-init devel schema --config-file cloud_init.cfg  
Valid cloud-config: cloud_init.cfg
```

پس از آن باید ماشین مجازی رو ایجاد کنیم.

```
(mohammdreza@kali)-[~/Desktop/cloud-init]  
$ sudo virt-install --name test1 \  
--virt-type kvm --memory 2048 --vcpus 2 \  
--boot hd,menu=on \  
--disk path=insert.img,device=cdrom \  
--disk path=snapshot-bionic-server-cloudimg.qcow2,device=disk \  
--graphics vnc \  
--os-type Linux --os-variant ubuntu18.04 \  
--network network:default \  
--console pty,target_type=serial
```

Starting install...

ERROR Requested operation is not valid: network 'default' is not active

Domain installation does not appear to have been successful.

If it was, you can restart your domain by running:

virsh --connect qemu:///system start test1

otherwise, please restart your installation.

یه ارور که با یه سرچ ساده حل میشه (باید Bridge مجازی رو فعال کنیم)

```
(mohammdreza@kali)-[~/Desktop/cloud-init]
$ sudo virsh net-list --all
```

Name	State	Autostart	Persistent
default	inactive	no	yes

```
(mohammdreza@kali)-[~/Desktop/cloud-init]
$ sudo virsh net-start default
Network default started
```

```
(mohammdreza@kali)-[~/Desktop/cloud-init]
$ sudo virsh net-list --all
```

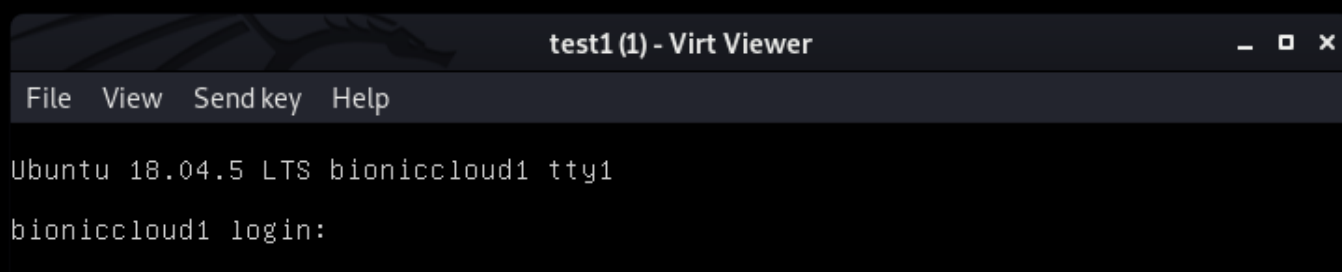
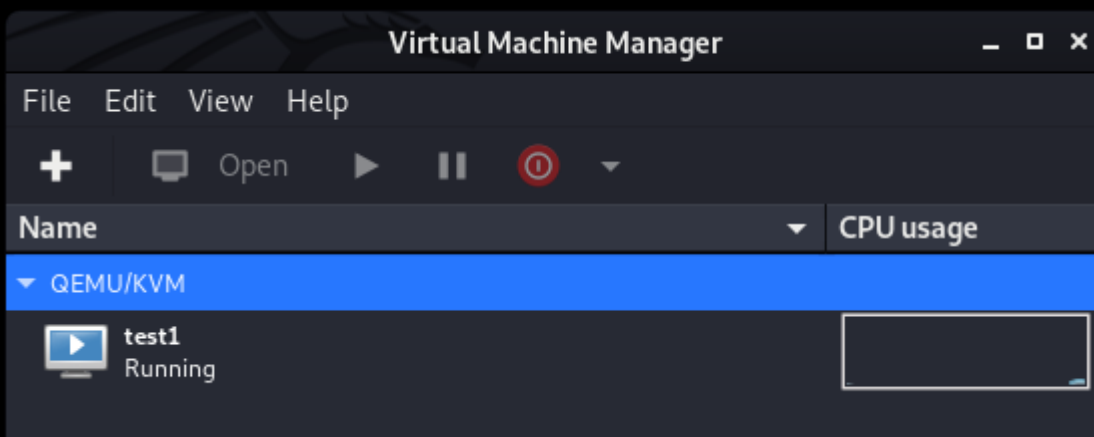
Name	State	Autostart	Persistent
default	active	no	yes

امتحان دوباره

```
(mohammdreza@kali)-[~/Desktop/cloud-init]
$ sudo virt-install --name test1 \
--virt-type kvm --memory 2048 --vcpus 2 \
--boot hd,menu=on \
--disk path=insert.img,device=cdrom \
--disk path=snapshot-bionic-server-cloudimg.qcow2,device=disk \
--graphics vnc \
--os-type Linux --os-variant ubuntu18.04 \
--network network:default \
--console pty,target_type=serial

Starting install...
Running graphical console command: virt-viewer --connect qemu:///system --wait test1
```





و پس از چند دقیقه دیتا کامل منتقل می شود

```
mohammdreza@kali: ~/Desktop/cloud-init
mohammdreza@kali: ~/Desktop/nordvpn
mohammdreza@kali: ~/Desktop/cloud-init 118x23

(mohammdreza@kali)-[~/Desktop/cloud-init]
$ sudo virt-install --name test1 \
--virt-type kvm --memory 2048 --vcpus 2 \
--boot hd,menu=on \
--disk path=insert.img,device=cdrom \
--disk path=snapshot-bionic-server-cloudimg.qcow2,device=disk \
--graphics vnc \
--os-type Linux --os-variant ubuntu18.04 \
--network network:default \
--console pty,target_type=serial

Starting install...
Running graphical console command: virt-viewer --connect qemu:///system --wait test1

mohammdreza@kali: ~/Desktop/cloud-init 118x30

(mohammdreza@kali)-[~/Desktop/cloud-init]
$ nc -lnvp 4444
Listening on 0.0.0.0 4444
Connection received on 192.168.122.84 40246
POST / HTTP/1.1
Host: 192.168.1.10:4444
User-Agent: curl/7.58.0
Accept: */*
Content-Length: 1087
Content-Type: application/x-www-form-urlencoded
Expect: 100-continue

Architecture:      x86_64CPU op-mode(s):      32-bit, 64-bitByte Order:      Little EndianCPU(s):      2
On-line CPU(s) list: 0,1Thread(s) per core: 1Core(s) per socket: 1Socket(s):      2NUMA node(s):      1Vendor
ID:      AuthenticAMDfamily:      21Model:      2Model name:      AMD Opteron 63xx class CP
UStepping:      0CPU MHz:      4063.608BogoMIPS:      8127.21Hypervisor vendor:      KVMVirtualization
type: fullL1d cache:      64KLi1 cache:      64KL2 cache:      512KL3 cache:      16384KNUMA no
de0 CPU(s):      0,1Flags:      fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush
mmx fxsr sse sse2 syscall nx mmxext fxsr_opt pdpe1gb rdtscp lm rep_good nopl cpuid extd_apicid tsc_known_freq pni pclm
ulqddq ssse3 fma cx16 sse4_1 sse4_2 x2apic popcnt tsc_deadline_timer aes xsave avx f16c hypervisor lahf_lm cmp_legacy c
r8_legacy abm sse4a misalignsse 3dnowprefetch osvw xop fma4 tbnm perfctr_core ssbd ibpb vmcall tsc_adjust bmi1 virt_ss
bd arat arch_capabilities
```

بدیهی است که برای بدست آوردن جزئیات دستورات میتوان از دستور **man** یا اون دستور با سوئیچ **-h** یا **help** استفاده کرد.

```

mohammdreza@kali: ~/Desktop/cloud-init
mohammdreza@kali: ~/Desktop/nordvpn
mohammdreza@kali: ~/Desktop/cloud-init 118x26
QEMU-IMG(1) QEMU QEMU-IMG(1)

NAME
    qemu-img - QEMU disk image utility

SYNOPSIS
    qemu-img [standard options] command [command options]

DESCRIPTION
    qemu-img allows you to create, convert and modify images offline. It can handle all image formats supported by QEMU.

    Warning: Never use qemu-img to modify images in use by a running virtual machine or any other process; this may destroy the image. Also, be aware that querying an image that is being modified by another process may encounter inconsistent state.

OPTIONS
    Standard options:

    -h, --help
        Display this help and exit

    -V, --version
        Display version information and exit

Manual page qemu-img(1) line 1 (press h for help or q to quit)

(mohammdreza@kali)-[~/Desktop/cloud-init]
$ qemu-img -h
qemu-img version 5.2.0 (Debian 1:5.2+dfsg-10)
Copyright (c) 2003-2020 Fabrice Bellard and the QEMU Project developers
usage: qemu-img [standard options] command [command options]
QEMU disk image utility

    '-h', '--help'      display this help and exit
    '-V', '--version'   output version information and exit
    '-T', '--trace'     [[enable=<pattern>][,events=<file>][,file=<file>]]
                        specify tracing options

Command syntax:
    amend [--object objectdef] [--image-opts] [-p] [-q] [-f fmt] [-t cache] [--force] -o options filename
    bench [-c count] [-d depth] [-f fmt] [--flush-interval=flush_interval] [-i aio] [-n] [--no-drain] [-o offset] [--pat
tern=pattern] [-q] [-s buffer_size] [-S step_size] [-t cache] [-w] [-U] filename
    bitmap (--merge SOURCE | --add | --remove | --clear | --enable | --disable)... [-b source_file [-F source_fmt]] [-g
granularity] [--object objectdef] [--image-opts] [-f fmt] filename bitmap
    check [--object objectdef] [--image-opts] [-q] [-f fmt] [--output=ofmt] [-r [leaks | all]] [-T src_cache] [-U] filen
ame
    commit [--object objectdef] [--image-opts] [-q] [-f fmt] [-t cache] [-b base] [-r rate_limit] [-d] [-p] filename
    compare [--object objectdef] [--image-opts] [-f fmt] [-F fmt] [-T src_cache] [-p] [-q] [-s] [-U] filename1 filename2
    convert [--object objectdef] [--image-opts] [--target-image-opts] [--target-is-zero] [--bitmaps] [-U] [-C] [-c] [-p]
[-q] [-n] [-f fmt] [-t cache] [-T src_cache] [-O output_fmt] [-B backing_file] [-o options] [-l snapshot_param] [-S s
parse_size] [-r rate_limit] [-m num_coroutines] [-W] [--salvage] filename [filename2 [...]] output_filename
    create [--object objectdef] [-q] [-f fmt] [-b backing_file] [-F backing_fmt] [-u] [-o options] filename [size]
    dd [--image-opts] [-U] [-f fmt] [-O output_fmt] [bs=block_size] [count=blocks] [skip=blocks] if=inout of=output

```