

Multipass est un utilitaire qui permet de lancer des machines virtuelles Ubuntu très facilement.

En fonction de l'OS, Multipass peut utiliser différents hyperviseurs:

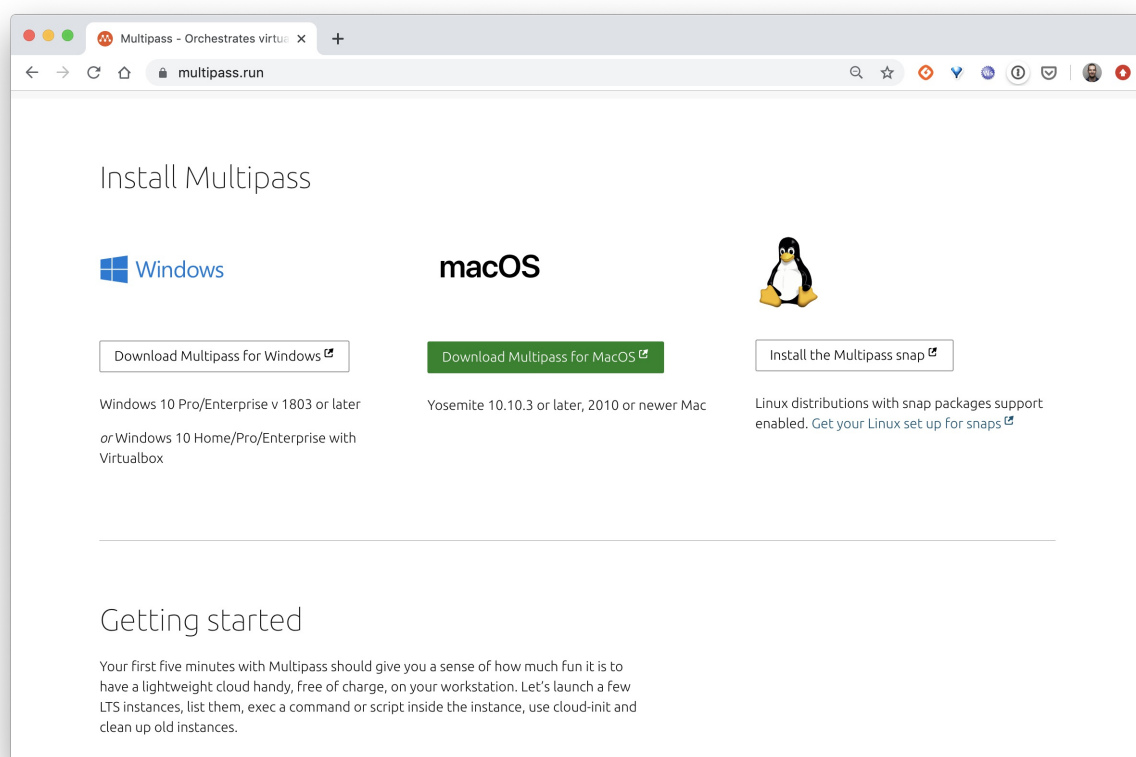
- Hyper-V
- HyperKit
- KVM
- VirtualBox

En s'intégrant de manière native à ces hyperviseurs, il permet de démarrer des machines virtuelles très rapidement.

## Installation

---

Vous trouverez sur le site <https://multipass.run> la procédure d'installation de Multipass en fonction de votre OS.



## Commandes disponibles

---

La liste des commandes disponibles pour la gestion du cycle de vie des VMs peut être

obtenue avec la commande suivante:

```
$ multipass
Usage: multipass [options] <command>
Create, control and connect to Ubuntu instances.

This is a command line utility for multipass, a
service that manages Ubuntu instances.

Options:
  -h, --help      Display this help
  -v, --verbose   Increase logging verbosity, repeat up to three times for more
                  detail

Available commands:
  delete      Delete instances
  exec        Run a command on an instance
  find        Display available images to create instances from
  get         Get a configuration setting
  help        Display help about a command
  info        Display information about instances
  launch      Create and start an Ubuntu instance
  list        List all available instances
  mount       Mount a local directory in the instance
  purge       Purge all deleted instances permanently
  recover     Recover deleted instances
  restart     Restart instances
  set         Set a configuration setting
  shell       Open a shell on a running instance
  start       Start instances
  stop        Stop running instances
  suspend     Suspend running instances
  transfer    Transfer files between the host and instances
  umount      Unmount a directory from an instance
  version     Show version details
```

Nous allons voir quelques unes de ces commandes sur des exemples.

## Quelques exemples

---

La manipulation de VMs se fait très facilement:

- création d'une nouvelle VM nommée *node1* (en quelques dizaines de secondes seulement)

```
$ multipass launch -n node1
```

Launched: node1

Par défaut cette VM est configurée avec 1G de RAM, 1 cpu et 5 Go de disque mais différentes options peuvent être utilisées pour modifier ces valeurs. La commande suivante permet par exemple de créer une VM nommée *node2* avec 2 cpu, 3 Go de RAM et 10 Go de disque:

```
$ multipass launch -n node2 -c 2 -m 3G -d 10G
```

- information sur une VM

La commande suivante retourne les différents paramètres de configuration de la VM

```
$ multipass info node1
Name:          node1
State:         Running
IPv4:          192.168.64.11
Release:       Ubuntu 18.04.3 LTS
Image hash:    6afb97af96b6 (Ubuntu 18.04 LTS)
Load:          0.07 0.06 0.02
Disk usage:    999.4M out of 4.7G
Memory usage:  75.0M out of 986.0M
```

Il est également possible d'obtenir ces informations dans les formats json, csv ou yaml.

Exemple en json:

```
$ multipass info node1 --format json
{
  "errors": [
  ],
  "info": {
    "node1": {
      "disks": {
        "sda1": {
          "total": "5019643904",
          "used": "1675780096"
        }
      },
      "image_hash":
"6afb97af96b671572389935d6579557357ad7bbf2c2dd2cb52879c957c85dbec",
      "image_release": "18.04 LTS",
      "ipv4": [
        "192.168.64.11"
      ],
    }
  },
}
```

```

        "load": [
            0.1,
            0.08,
            0.06
        ],
        "memory": {
            "total": 1033945088,
            "used": 139534336
        },
        "mounts": {
        },
        "release": "Ubuntu 18.04.3 LTS",
        "state": "Running"
    }
}
}

```

- liste des VM créés

```

$ multipass list

```

Name	State	IPv4	Image
node1	Running	192.168.64.11	Ubuntu 18.04 LTS

- Lancement d'un shell dans la VM *node1*

```

$ multipass shell node1
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-72-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Tue Dec 24 14:09:23 CET 2019

System load:  0.01          Processes:            107
Usage of /:   33.4% of 4.67GB Users logged in:      0
Memory usage: 19%          IP address for enp0s2: 192.168.64.11
Swap usage:   0%           IP address for docker0: 172.17.0.1

3 packages can be updated.
0 updates are security updates.

Last login: Tue Dec 24 14:06:16 2019 from 192.168.64.1
ubuntu@node1:~$

```

On obtient alors un shell avec l'utilisateur *ubuntu* qui est notamment dans le groupe *sudo*.

- lancement d'une commande dans une VM

La commande suivante permet d'installer Docker dans la VM *node1*

```
$ multipass exec node1 -- /bin/bash -c "curl -sSL https://get.docker.com | sh"
```

On peut alors vérifier que l'installation s'est déroulée correctement:

```
$ multipass exec node1 -- sudo docker version
Client: Docker Engine - Community
Version:      19.03.5
API version:  1.40
Go version:   go1.12.12
Git commit:   633a0ea838
Built:        Wed Nov 13 07:29:52 2019
OS/Arch:      linux/amd64
Experimental: false

Server: Docker Engine - Community
Engine:
Version:      19.03.5
API version:  1.40 (minimum version 1.12)
Go version:   go1.12.12
Git commit:   633a0ea838
Built:        Wed Nov 13 07:28:22 2019
OS/Arch:      linux/amd64
Experimental: false
containerd:
Version:      1.2.10
GitCommit:    b34a5c8af56e510852c35414db4c1f4fa6172339
runc:
Version:      1.0.0-rc8+dev
GitCommit:    3e425f80a8c931f88e6d94a8c831b9d5aa481657
docker-init:
Version:      0.18.0
GitCommit:    fec3683
```

- montage d'un répertoire local dans une VM

```
# Création d'un fichier en local
$ mkdir /tmp/test && touch /tmp/test/hello

# Montage du répertoire dans le filesystem de la VM node1
$ multipass mount /tmp/test node1:/usr/share/test
```

```
# Vérification
$ multipass exec node1 -- ls /usr/share/test
hello
```

La commande *umount* permet de faire l'opération inverse et de supprimer ce point de montage:

```
$ multipass umount node1:/usr/share/test
```

- copie de fichiers entre la machine local et les VMs

Il est possible de transférer des fichiers locaux vers une VM et inversement, sans avoir à monter un répertoire (cf exemple précédent)

```
# Copie d'un fichier depuis la machine locale
$ multipass transfer /tmp/test/hello node1:/tmp/hello

# Vérification
$ multipass exec node1 -- ls /tmp/hello
/tmp/hello
```

- les commandes start / stop / restart / delete permettent de gérer le cycle de vie des VMs

```
$ multipass delete node1
```

## En résumé

---

Comme nous venons de le voir dans les exemples ci-dessus, Multipass est un utilitaire très pratique et extrêmement simple d'utilisation. Je vous conseille de l'installer car nous l'utiliserons dans la suite pour instancier plusieurs VMs et mettre en place un cluster Kubernetes en local.