Multipass CLI 2022, Ubuntu VMs on demand on any OS Cheat Sheet

by Boulard via cheatography.com/125912/cs/34234/

Create Instances

multipass launch	Creates an instance using a random name
multipass info <instance>	Get info about an instance
multipass find	find out what images are available
multipass launch <image>	launch an instance using the specified image
multipass launch <image> [--name|-n] <instance>	launch an instance and name it
multipass launch [--cpus|-c] 4 [--disk,-d] 20G [--mem,-m] 8G	Create an instance with custom CPU number, disk, and RAM
multipass launch <image> [--name,-n] primary	An instance can obtain the primary status at creation time if its name is primary. A primary instance is the default argument for several commands, such as shell , start , stop , restart , and suspend and also automatically mounts our $HOME directory in the instance.
multipass launch --network en0 --network name=bridge0,mode=manual	Create an instance specifying NICs. See https://multipass.run/docs/create-an-instance#heading--create-an-instance-with-multiple-network-interfaces

The --network option can be given multiple times, each one requesting an additional network interface (beyond the default one, which is always present, used to reach the outside world and used by multipass to communicate with the instance).

--network takes at most 3 <key>=<value>:
name — the only required value, it identifies the network,
mode that is either manual or auto (try to auto join the net),
mac to specify a mac address for the NIC

NIC: network interface controller

Create an instance with a custom DNS

multipass launch --cloud-init systemd-resolved.yaml	Creates an instance with a custom DNS
after the instance booted use the netplan.io approach	https://multipass.run/docs/create-an-instance#heading--the-netplanio-approach

In some scenarios the default of using the system-provided DNS will not be sufficient.

systemd-resolved.yaml cloud-init snippet:

```
#cloud-config
bootcmd:
- printf "[Resolve]\nDNS=8.8.8.8" > /etc/systemd/resolved.conf
- [systemctl, restart, systemd-resolved]
```

8.8.8.8 is a Google DNS

Remove or delete an instance

multipass delete <instance>	Move an instance to the ‘recycle bin’
multipass delete --all	Move all instances to the ‘recycle bin’
multipass recover <instance>	Recover an instance put to the ‘recycle bin’
multipass delete --purge <instance>	Permanently delete an instance
multipass purge	Empty the ‘recycle bin’

Running a container with the Docker workflow

multipass launch docker	Creates a virtual machine running the latest version of Ubuntu, with Docker and Portainer installed.
multipass exec docker docker	Executes Docker in the instance

Running a container with the Docker workflow (cont)

multipass alias docker:docker	Alias the docker command to our host command line. This will let us use the docker command as if Docker were running directly on our host machine.This command will return instructions showing how to add the new alias to our path. Simply copy and paste the command shown. It will likely be of this form: PATH="$PATH:/home/<user>/snap/multipass/common/bin"
docker run hello-world	Works only if the alias is created.
multipass list	Shows the IP addresses associated with the created Ddocker VM. The Portainer IP should start with a 10.
http://10.portainer-ip:9000	Shows the Portainer web interface.

Portainer config
1) Set username and password
2) Select local docker env opt. and click connect
3) Click on the newly created “Local” environment to manage the Docker instance on our local VM.
see https://multipass.run/docs/docker-tutorial

Install

brew install bash-completion	install bash completion if not already installed
brew install --cask multipass	--cask option may be omitted
brew uninstall multipass	uninstall multipass
brew uninstall --zap multipass	uninstall multipass and data
multipass version	to check the install and get the version

Modify an instance

multipass set client.primary-name=<instance name>	Grant a regular instance the primary status
multipass set [local.<instance>.(cpus|disk|memory)]	Change CPU RAM DISK

see https://multipass.run/docs/modify-an-instance

Use an Instance

multipass shell <instance>	Open a shell prompt inside an instance
multipass exec <instance> -- <cmd>	Execute a command inside an instance
multipass start|suspend|stop <instance>[<instance>]	Start, suspend or stop instances
multipass start|suspend|stop -all	Start, suspend or stop all instances

Aliases on Instances

multipass alias <instance>:<cmd> <alias>	defines an alias executed from the host, into the instance. ex: multipass alias proxy:ls lsproxy
multipass alias <instance>:<cmd> <alias> --no-map-working-directory	Do not change the working directory on the instance to the mounted directory (in case the host folder on which executing an alias is mounted on the instance)
multipass aliases	list defined aliases and their instances
multipass <alias>	execute an alias
multipass unalias <alias>[<alias>]	remove aliases
multipass unalias --all	remove all aliases

Add the multipass bin folder to the path to use <alias> instead of multipass <alias>.
Arguments are also supported

```
nano ~/.bashrc #linux
nano ~/.zshhrc #mac
PATH="$PATH:/home/user/snap/multipass/common/bin"
```

Share data with an instance using "mount"

multipass mount $HOME <instance>	/home/me will be available inside the instance.
multipass mount $HOME <instance>:</instance/path>	Specifies the path inside the instance in which to mount the local path
multipass launch --mount </local/path>:multipass launch --mount /some/local/path:/instance/path	Specifies mount on launch
multipass umount <instance>	Unmount all volumes from an instance
multipass umount <instance>:</instance/path>	Unmount a specific volume from an instance

The recommended way to share data between your host and an instance with Multipass is the mount command

Share data with an instance using "transfer"

multipass transfer <instance>:</instance/path/file1> <instance>:</instance/path/file2> </host/path>	Copy 2 files from the instance to the specified host path
multipass transfer <instance>:</instance/path> </host/path/file1> </host/path/file2><instance>:</instance/path>	Copy 2 files from the host path to the specified instance path

use transfer to just copy files around. If a file is inside an instance, prefix its path with <instance name>:.

Miscellaneous

multipass exec primary -- bash -c 'echo "PasSwOrD" | docker secret create secret-pass -'	create a docker secret in the primary node without storing it