

QuickRun

- [Docker](#)
 - [Docker! Why](#)
 - [Install Docker](#)
 - [How to link multiple docker-compose services via network](#)
 - [Usefull commands](#)
 - [Interesting Articles](#)
 - [Docker SPA](#)
 - [Setting up Docker in Docker \(dind/socket\) with GitLab Runners](#)
 - [Docker Compose Generator](#)
- [Python](#)
 - [Nginx uWSGI Flask Python2](#)
 - [Python3 VENV](#)
- [Download Broadcom VMWare Hypervisors](#)
- [ZLib](#)
- [LINUX for Today](#)
 - [Running out of inodes](#)
 - [Extend Ubuntu Disk](#)
 - [Logical Volume Manager - LVM](#)
 - [rm -rf root permissions denied](#)
 - [Wireguard](#)
 - [Nginx Tunning](#)
 - [User RDP SSH](#)
 - [VestaCP/HestiaCP Commands](#)
 - [Register New Gitlab Runner \(Docker-in-Docker\)](#)
 - [MinIO Client Operation](#)

- [Hardware](#)
 - [Alfa Wifi AWUS036AXML](#)
 - [Xiaomi 13T](#)
- [Bloodhound Linux/Docker Quickstart](#)
- [Clone ESXi VMs](#)
- [USB Drive Read-Only Setup Guide](#)
- [Programming Notes for Professionals books](#)
- [Top 25 Penetration Testing Skills and Competencies \(Detailed\)](#)
- [LLMs Locally](#)
- [Logon Remote MSSQL via Windows Account](#)

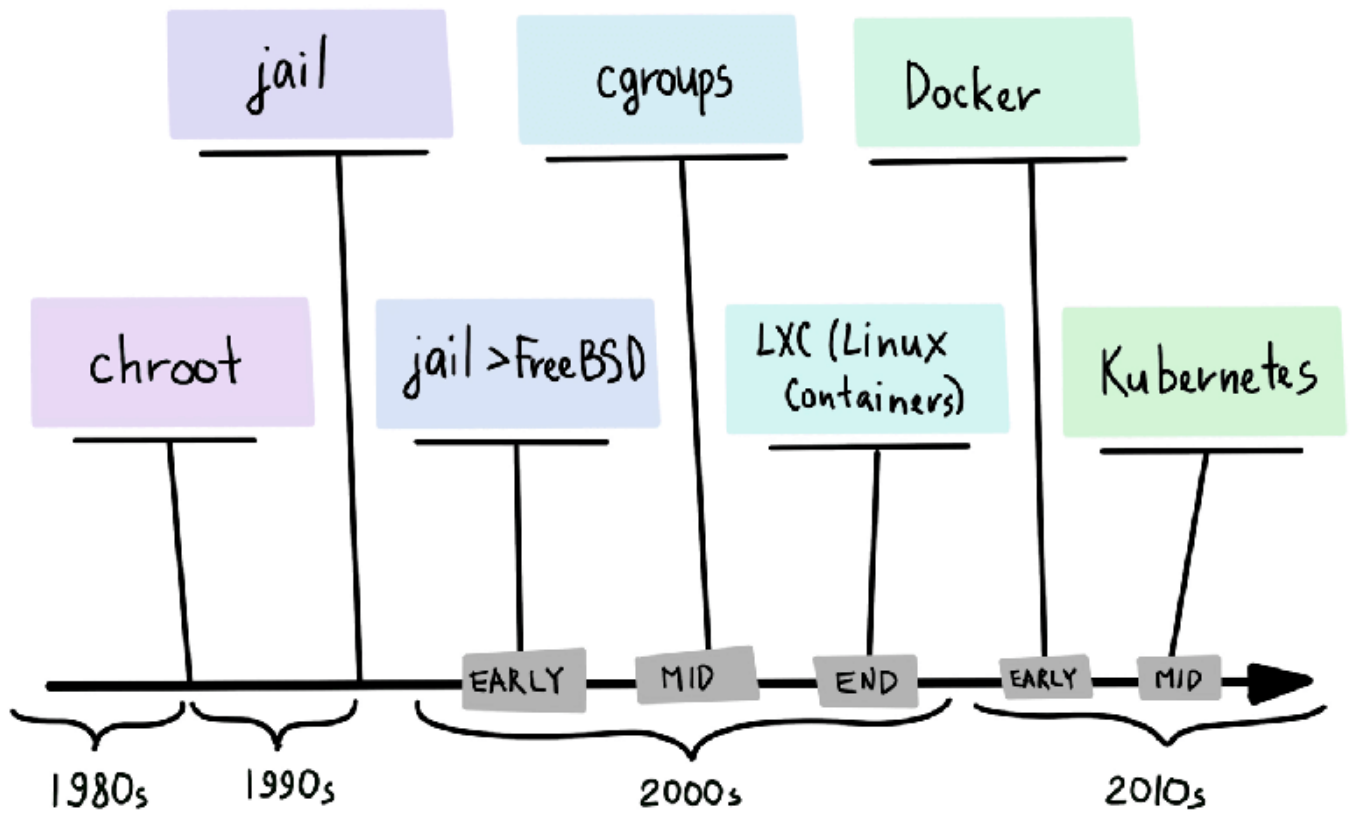
Docker

Docker

Docker! Why



Docker containers have become the de facto development standard, but it's worth mentioning that Docker wasn't the first player in the container world.



@luminousmen.com

Docker

Install Docker

Quick Install

```
curl -fsSL https://get.docker.com | bash
sudo groupadd docker
sudo usermod -aG docker $USER
```

1. https://docs.docker.com/get-started/docker_cheatsheet.pdf
2. <https://dockerlabs.collabnix.com/docker/cheatsheet/>

How to link multiple docker-compose services via network

This scenario came from a question I was asked docker-compose and network connectivity between services defined in different docker-compose.yml files.

The desired result was to be able to define a docker-compose.yml in one file, and in a second docker-compose.yml have the ability to reach the first service via service or container name for development purposes.

Default scenario: Two separate docker-compose.yml and two separate default networks

Let's take a simple docker compose file.

```
version: '3'
services:
  service1:
    image: busybox
    command: sleep infinity
```

When it starts up, a default network is created. Its name is based on the service name and the directory name of the docker-compose.yml file.

```
$ pwd
/tmp/docker-example/compose1

$ docker-compose up -d
Creating network "compose1_default" with the default driver
Creating compose1_service1_1 ... done
```

Second docker compose file

```
version: '3'
services:
  service2:
    image: busybox
    command: sleep infinity
```

Starting services in a second docker compose file, we see the same behavior. A new default network is created and used.

```
$ pwd
/tmp/docker-example/compose2

$ docker-compose up -d
Creating network "compose2_default" with the default driver
Creating compose2_service2_1 ... done
```

A side-effect of these isolated networks are that the containers are unable to ping one another by service name or container name.

Test: From Service 1 ping Service 2

```
# By service name
$ docker exec -it compose1_service1_1 ping service2
ping: bad address 'service2'

# By container name
$ docker exec -it compose1_service1_1 ping compose2_service2_1
ping: bad address 'compose2_service2_1'
```

Test: Service 2 ping Service 1

```
# By service name
$ docker exec -it compose2_service2_1 ping service1
ping: bad address 'service1'

# By container name
$ docker exec -it compose2_service2_1 ping compose1_service1_1
ping: bad address 'compose1_service1_1'
```

New scenario: Sharing a network between services

If you want define services in multiple docker-compose.yml files, and also have network connectivity between the services, you need to configure your services to use the same network.

To create an external network, you can run `docker network create <name>`. -- where `<name>` can be a single string without spaces.

Creating the network

```
$ docker network create external-example
2af4d92c2054e9deb86edaea8bb55ecb74f84a62aec7614c9f09fee386f248a6
```

Modified first docker-compose file with network configured

```
version: '3'
services:
  servicel:
    image: busybox
    command: sleep infinity

networks:
  default:
    external:
      name: external-example
```

Restarting the services

```
$ pwd
/tmp/docker-example/compose1

$ docker-compose up -d
Creating compose1_servicel_1 ... done
```

Modified second docker-compose file with network configured

```
version: '3'
services:
  service2:
    image: busybox
    command: sleep infinity

networks:
  default:
    external:
      name: external-example
```

Restarting the services

```
$ pwd
/tmp/docker-example/compose2

$ docker-compose up -d
Creating compose2_service2_1 ... done
```

After running `docker-compose up -d` on both docker-compose.yml files, we see that no new networks were created.

```
$ docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
25e0c599d5e5       bridge             bridge             local
2af4d92c2054       external-example   bridge             local
7df4631e9cff       host               host               local
194d4156d7ab       none              null               local
```

With the containers using the `external-example` network, they are able to ping one another.

Test: Service 1 ping Service 2

```
# By service name
$ docker exec -it compose1_service1_1 ping service2
PING service2 (172.24.0.3): 56 data bytes
```

```
64 bytes from 172.24.0.3: seq=0 ttl=64 time=0.054 ms
^C
--- service2 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 0.054/0.054/0.054 ms

# By container name
$ docker exec -it compose1_service1_1 ping compose2_service2_1
PING compose2_service2_1 (172.24.0.2): 56 data bytes
64 bytes from 172.24.0.2: seq=0 ttl=64 time=0.042 ms
^C
--- compose2_service2_1 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 0.042/0.042/0.042 ms
```

Test: Service 2 ping Service 1

```
# By service name
$ docker exec -it compose2_service2_1 ping service1
PING service1 (172.24.0.2): 56 data bytes
64 bytes from 172.24.0.2: seq=0 ttl=64 time=0.041 ms
^C
--- service1 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 0.041/0.041/0.041 ms

# By container name
$ docker exec -it compose2_service2_1 ping compose1_service1_1
PING compose1_service1_1 (172.24.0.3): 56 data bytes
64 bytes from 172.24.0.3: seq=0 ttl=64 time=0.042 ms
^C
--- compose1_service1_1 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 0.042/0.042/0.042 ms
```

“ As a note, you can configure your services to use a custom container name by declaring the `container_name` key under each service (i.e., at the same level as

`image`).

[Link to Docker-compose docs - container_name](#)

Takeaway

You can connect services defined across multiple docker-compose.yml files.

In order to do this you'll need to:

1. Create an external network with `docker network create <network name>`
2. In each of your docker-compose.yml configure the default network to use your externally created network with the `networks` top-level key.
3. You can use either the service name or container name to connect between containers.

Usefull commmands

The most elegant way to **remove all non-running containers** seems to be:

```
docker rm $(docker ps -q -f status=exited)
```

- `-q` prints just the container ids (without column headers)
- `-f` allows you to filter your list of printed containers (in this case we are filtering to only show exited containers)

Interesting Articles

Docker Best Practices: Choosing Between RUN, CMD, and ENTRYPOINT

<https://www.docker.com/blog/docker-best-practices-choosing-between-run-cmd-and-entrypoint/>

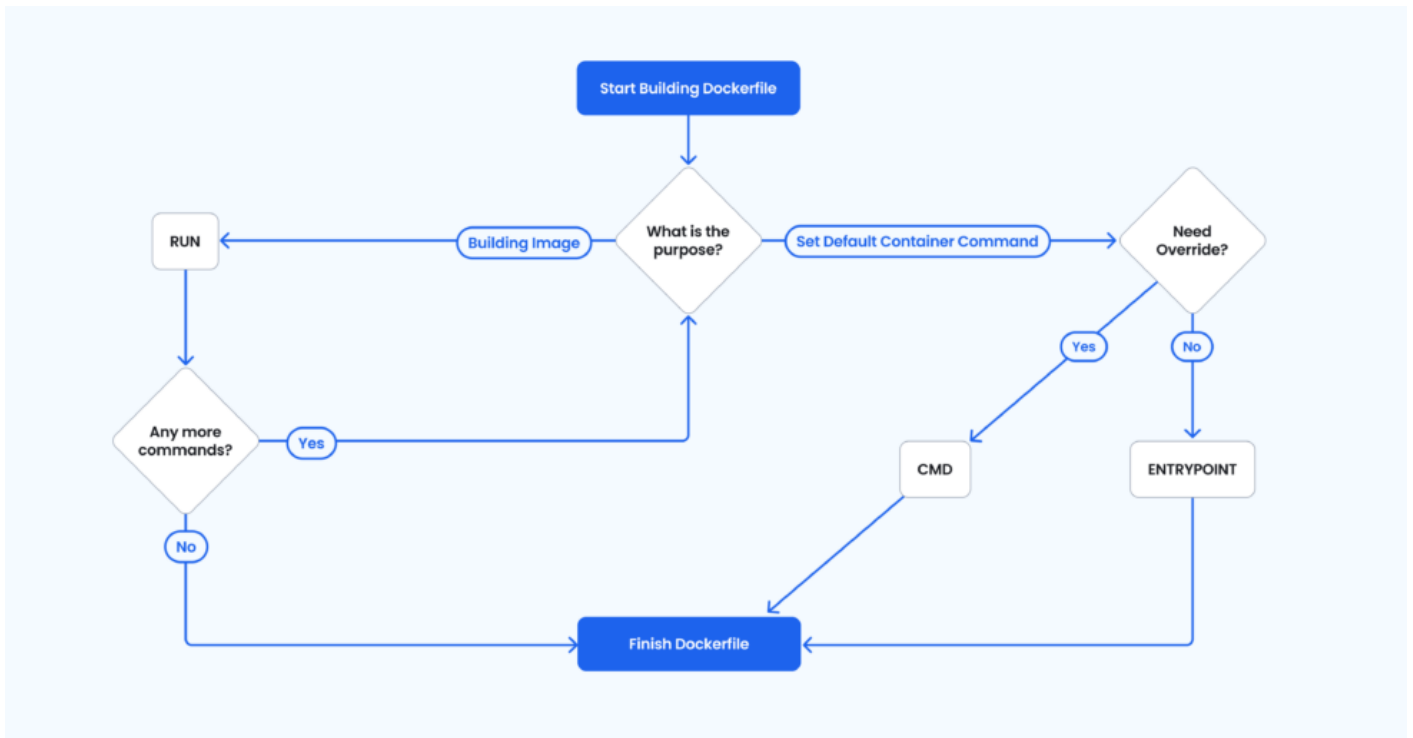


Figure 1: Decision tree — RUN, CMD, ENTRYPOINT.

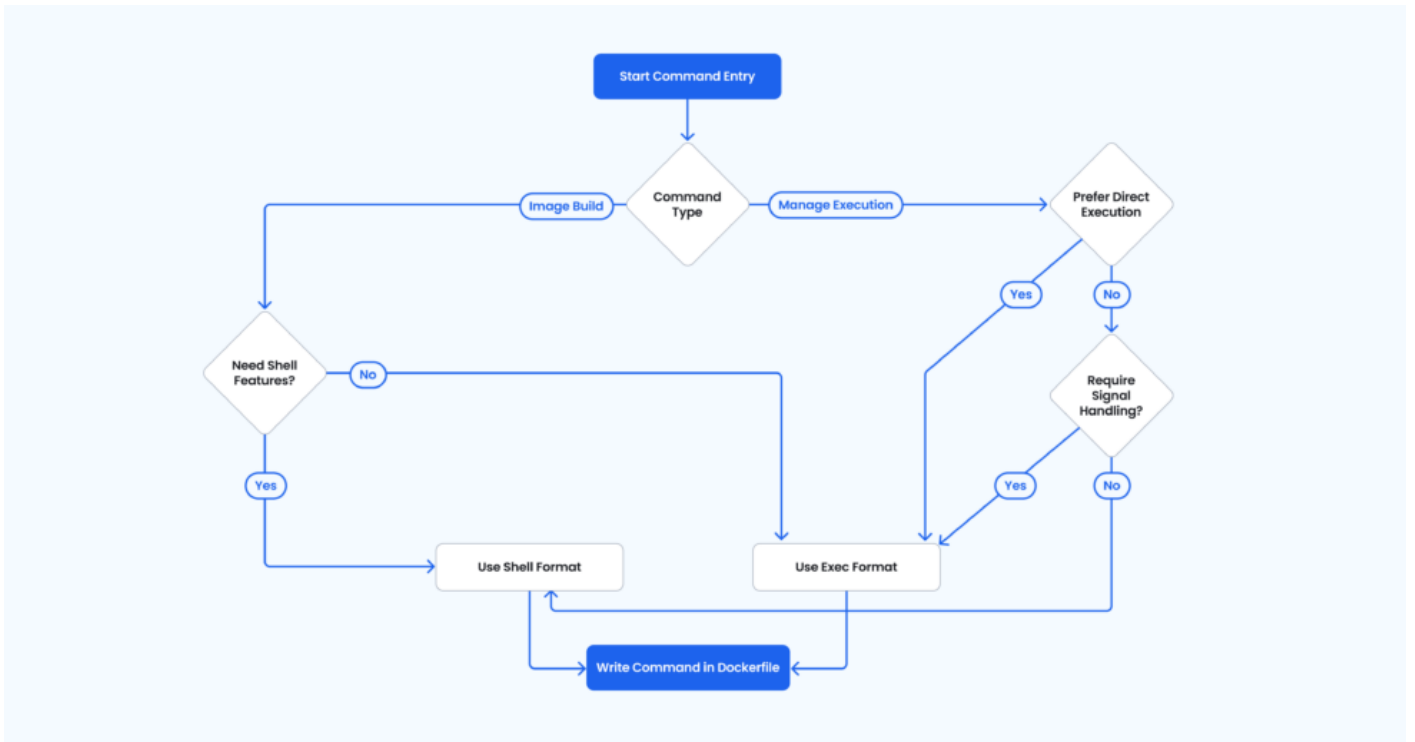


Figure 2: Decision tree — exec vs. shell form.

Docker SPA

<https://skamalakannan.dev/posts/dockerizing-your-spa/>

After building the image successfully, we can run the history command, (`docker history <<image name/id>>`) to check how the image is created. We will not see any reference to the node image we used earlier.

IMAGE	CREATED	CREATED BY	SIZE	COMMENT
f9772eb1b27e	34 hours ago	/bin/sh -c #(nop) CMD ["nginx" "-g" "daemon...	0B	
ee9a357e38f8	34 hours ago	/bin/sh -c #(nop) STOPSIGNAL SIGTERM	0B	
603954ff32ce	34 hours ago	/bin/sh -c #(nop) EXPOSE 80	0B	
86cc7e861e16	34 hours ago	/bin/sh -c #(nop) COPY dir:613e7cd220014aee4...	10.5MB	
421ffa35c898	8 weeks ago	/bin/sh -c #(nop) COPY file:445e6d60a9403657...	2.77kB	
3eec29d27b40	2 months ago	/bin/sh -c #(nop) COPY file:63d959eea7b628cc...	2.05kB	
0d8e0e4fef67	2 months ago	/bin/sh -c #(nop) COPY dir:db672636cbedd4a5...	12.9kB	
ea335eea17ab	3 months ago	/bin/sh -c #(nop) CMD ["nginx" "-g" "daemon...	0B	
<missing>	3 months ago	/bin/sh -c #(nop) STOPSIGNAL SIGQUIT	0B	
<missing>	3 months ago	/bin/sh -c #(nop) EXPOSE 80	0B	
<missing>	3 months ago	/bin/sh -c #(nop) ENTRYPOINT ["/docker-entr...	0B	
<missing>	3 months ago	/bin/sh -c #(nop) COPY file:09a214a3e07c919a...	4.61kB	
<missing>	3 months ago	/bin/sh -c #(nop) COPY file:0fd5fca330dcd6a7...	1.04kB	
<missing>	3 months ago	/bin/sh -c #(nop) COPY file:0b866ff3fc1ef5b0...	1.96kB	
<missing>	3 months ago	/bin/sh -c #(nop) COPY file:65504f71f5855ca0...	1.2kB	
<missing>	3 months ago	/bin/sh -c set -x && addgroup --system -...	61.1MB	
<missing>	3 months ago	/bin/sh -c #(nop) ENV PKG_RELEASE=1~bullseye	0B	
<missing>	3 months ago	/bin/sh -c #(nop) ENV NJS_VERSION=0.7.0	0B	
<missing>	3 months ago	/bin/sh -c #(nop) ENV NGINX_VERSION=1.21.4	0B	
<missing>	3 months ago	/bin/sh -c #(nop) LABEL maintainer=NGINX Do...	0B	
<missing>	3 months ago	/bin/sh -c #(nop) CMD ["bash"]	0B	
<missing>	3 months ago	/bin/sh -c #(nop) ADD file:a2405ebb9892d98be...	80.4MB	

Docker

Setting up Docker in Docker (dind/socket) with GitLab Runners

<https://blog.hiebl.cc/posts/gitlab-runner-docker-in-docker/#dind-vs-socket-passthrough>

Docker Compose Generator

<https://compose.ajnart.dev/>

Docker Compose Generator

Select the tools you want to include in your docker-compose.yml

[Suggest container](#) [Support me](#) [GitHub](#)

Docker Compose Maker

[Template Gallery](#)

Docker Compose Settings [Show Settings](#)

[Discover More Self-Hosted Apps](#) [Explore](#)

0 services selected [Share](#) [Unselect](#) [Copy Compose](#)

Sonarr <small>Media</small> Smart PVR for usenet and bittorrent users. Monitors multiple RSS feeds for new TV show episodes, automaticall... TV PVR Automation	Radarr <small>Media</small> A fork of Sonarr to work with movies. Automatically monitors and downloads movies, handles quality upgrades, an... Movies PVR Automation	Lidarr <small>Media</small> A music collection manager for Usenet and BitTorrent users. It can monitor multiple RSS feeds for new albums... Music PVR Automation	Readarr <small>Media</small> Book collection manager for Usenet and BitTorrent users. It integrates with clients such as SABnzbd, NZBGet, ... Books PVR Automation	Prowlarr <small>Media</small> A powerful indexer manager/proxy built on the arr stack. Seamlessly integrates with Sonarr, Radarr, Lidarr, ... Indexer Proxy Integration
Whisparr <small>Media</small> An adult movie collection manager for Usenet and BitTorrent users. Adult Movies Automation	NZBGet <small>Download</small> Efficient usenet downloader, optimized for reliability, completeness, and speed. Usenet Download NZB	Bazarr <small>Media</small> A companion application to Sonarr and Radarr that manages and downloads subtitles. Subtitles Media Automation	Jackett <small>Download</small> API Support for your favorite torrent trackers, translating searches into tracker-site-specific queries. Indexer Proxy Torrent	qBittorrent <small>Download</small> A cross-platform BitTorrent client with an integrated web interface. BitTorrent Download P2P

Python

Python

Nginx uWSGI Flask Python2

Flask + Virtual Env

Python 2.7 Virtual Env

```
pip2 install virtualenv
```

```
cd /home/user/web/domain  
virtualenv --python=$(which python2) ./private/  
source private/bin/activate  
pip install -r ../public_html/requirements.txt
```

uWSGI Emperor

```
pip2 install wheel uwsgi
```

```
cat /etc/systemd/system/uwsgi.service
```

```
[Unit]  
Description=uWSGI Emperor service  
After=syslog.target  
  
[Service]  
ExecStart=/usr/local/bin/uwsgi --emperor /etc/uwsgi/sites  
Restart=always  
KillSignal=SIGQUIT  
Type=notify  
StandardError=syslog  
NotifyAccess=all  
  
[Install]  
WantedBy=multi-user.target
```

```
systemctl daemon-reload
systemctl start uwsgi.service
systemctl status uwsgi.service
systemctl enable uwsgi.service
```

Application configuration file

```
cat web.domain.site.ini
```

```
[uwsgi]
uid=appuser
gid=www-data
socket=127.0.0.1:13000
procname-prefix-spaced = web.domain.site

# Enable uWSGI stats server
# stats = :13034
# stats-http = true

# the virtualenv (full path)
virtualenv=/home/appuser/web/web.domain.site/private
# Project path
chdir=/home/appuser/web/web.domain.site/public_html
# Django's wsgi file (path starting from chdir/)
wsgi-file=run.wsgi

# Log paths

logto=/var/log/nginx/domains/web.domain.site.uwsgi.log
log-date = [%Y:%m:%d %H:%M:%S]
logformat-strftime
log-format = [%fetime]
logfile-chown = www-data:appuser
logfile-chmod = 640

# Do not log some errors caused by client disconnects
ignore-sigpipe = true
ignore-write-errors = true
disable-write-exception = true
```

```
master=true
enable-threads=true
processes=2
threads=5
single-interpreter=true
thunder-lock=true
die-on-term=true
max-requests=1000
buffer-size=32768
post-buffering=8192

# clear environment on exit
vacuum=true
```

Ngix Virtualhost

<https://uwsgi-docs.readthedocs.io/en/latest/Ngix.html>

```
location / {
    include uwsgi_params;
    uwsgi_pass uwsgi://localhost:13000;
}
```

Ref: <https://uwsgi-docs.readthedocs.io/en/latest/Systemd.html>

Python

Python3 VENV

```
apt update && apt install -y python3.12-venv  
python3 -m venv sectigo  
source sectigo/bin/activate  
  
deactivate
```

Download Broadcom VMWare Hypervisors

Login Required

- <https://support.broadcom.com/group/ecx/free-downloads>
- <https://support.broadcom.com/group/ecx/productdownloads?subfamily=VMware%20Workstation%20Pro&freeDownloads=true>
- <https://support.broadcom.com/group/ecx/productdownloads?subfamily=VMware%20Fusion&freeDownloads=true>

ESXi 8 Minimum Storage

```
Shift + 0 on boot
```

```
Append
```

```
cpuUniformityHardCheckPanic=FALSE systemMediaSize=min
```

```
Permanent apply
```

```
esxcli system settings kernel set -s cpuUniformityHardCheckPanic -v FALSE
```

```
esxcli system settings kernel set -s ignoreMsrfaults -v TRUE
```

Ref:

<https://knowledge.broadcom.com/external/article/395554/esxi-psod-with-fatal-cpu-mismatch-on-fea.html>

<https://knowledge.broadcom.com/external/article/345195/boot-option-to-configure-the-size-of-esx.html>

- min 33GB, for single disk or embedded servers
- small 69GB, for servers with at least 512GB RAM
- max all available space, for multi-terabyte servers

ZLib

<http://bookszlibb74ugqojhzhg2a63w5i2atv5bqarulgczawnbmsb6s6qead.onion/>

LINUX for Today

Running out of inodes

```
# Check Inode  
df -i  
du --inode -d 1 | sort -n | head -10
```

Extend Ubuntu Disk

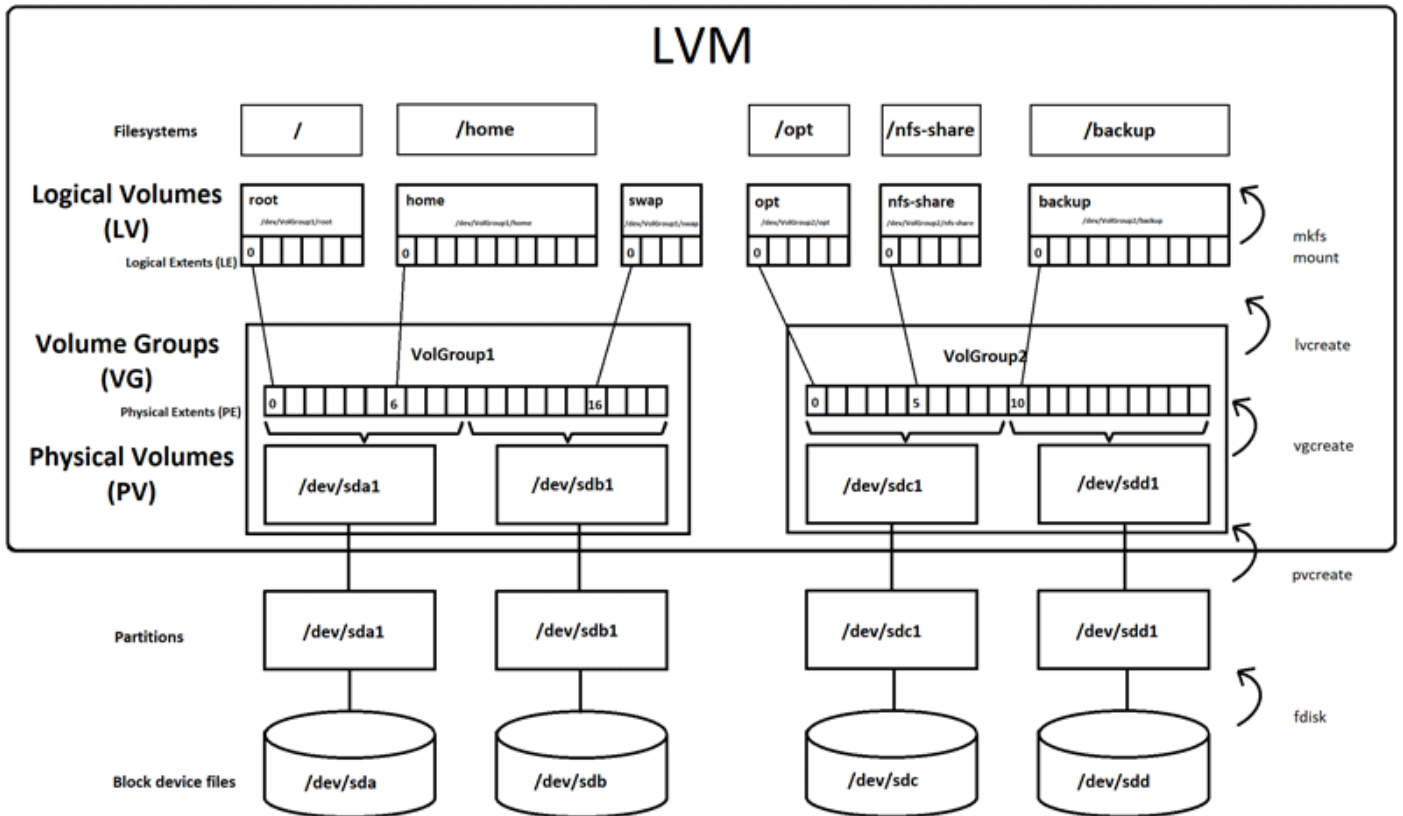
```
sudo parted -s -a opt /dev/sda "resizepart 3 100%"  
sudo lvextend -l +100%FREE /dev/ubuntu-vg/ubuntu-lv  
sudo resize2fs -p /dev/mapper/ubuntu--vg-ubuntu--lv
```

Extend single disk

```
growpart /dev/sda 2  
resize2fs /dev/sda2  
  
growpart /dev/sdb 1  
resize2fs /dev/sdb1
```

<https://netshopisp.medium.com/how-to-extend-lvm-disk-on-linux-ubuntu-20-04-35b1c2d5d5e9>

Logical Volume Manager - LVM



<https://packetpushers.net/blog/ubuntu-extend-your-default-lvm-space/>

Resize LVM, Ubuntu <https://rdr-it.com/en/ubuntu-extend-lvm-partition-and-disk/>

[servers for hackers](#)

[Ubuntu - Extend LVM partition and disk](#)

<https://rdr-it.io/en/ubuntu-extend-lvm-partition-and-disk/>

rm -rf root permissions denied

A number of things could be wrong here. From my own experience, I recommend checking the directory attributes with the `lsattr` utility. Often times I set the immutable bit on files to prevent accidental modification, and to slightly slow down any malicious person from messing with it. If the flag is indeed set, simply remove it: `chattr -R -i /var/www`

<https://unix.stackexchange.com/questions/529579/rm-rf-root-permissions-denied>

LINUX for Today

Wireguard

<https://www.ivpn.net/knowledgebase/linux/linux-autostart-wireguard-in-systemd/>

Nginx Tunning

Nginx GeoIP

<https://mailfud.org/geoip-legacy/>

Install Nginx Module

```
apt install nginx-module-geoip
```

The GeoIP dynamic modules for nginx have been installed.

To enable these modules, add the following to `/etc/nginx/nginx.conf` and reload nginx:

```
load_module modules/ngx_http_geoip_module.so;
load_module modules/ngx_stream_geoip_module.so;
```

```
http {
    geoip_country /usr/share/GeoIP/GeoIP.dat;
    geoip_city /usr/share/GeoIP/GeoIPCity.dat;
}
```

```
VHOST{
    add_header X-GeoIP "$geoip_country_code";
}
```

NGINX Tuning For Best Performance

<https://gist.github.com/denji/8359866>

User RDP SSH

```
useradd -m -s /bin/bash user
usermod -aG sudo user
passwd user
su user
echo "xfce4-session" | tee .xsession
sudo systemctl restart xrdp
```

Allow specific user login ssh without password/key

```
$ useradd ukztuxyysvotlekjk -m -s /bin/bash

add /etc/ssh/sshd_config

Match User ukztuxyysvotlekjk
    PermitEmptyPasswords yes
    PasswordAuthentication yes

Emty password
$ sudo passwd -d ukztuxyysvotlekjk
```

VestaCP/HestiaCP Commands

<https://forum.vestacp.com/viewtopic.php?t=10893#p41235>

Obtain the list of system domains

```
v-list-users | tail -n +3 | awk '{print "v-list-web-domains \"$1\" | tail -n +3"}' | bash  
v-list-users | tail -n +3 | awk '{print "v-list-web-domains \"$1\" | tail -n +3"}' | bash | awk  
'{ print $1}'
```

Register New Gitlab Runner (Docker-in-Docker)

Register New Runner

```
docker run --rm -it -v /opt/GitlabRunner/config:/etc/gitlab-runner gitlab/gitlab-runner
register --url https://gitlab.com --token TOKEN
```

compose.yml

```
services:
  gitlab-runner-container:
    image: gitlab/gitlab-runner:latest
    container_name: gitlab-runner-container
    restart: always
    volumes:
      - ./config:/etc/gitlab-runner/
      - /var/run/docker.sock:/var/run/docker.sock
```

MinIO Client Operation

```
mc admin user add alias user TOKEN
```

```
mc admin policy create alias key-upload-only key-upload-only.json
```

```
mc admin policy attach alias key-upload-only --user user
```

```
mc alias set key-user https://s3.alias.space user TOKEN
```

```
mc cp FILE key-user/key/
```

Hardware

Hardware

Alfa Wifi AWUS036AXML

USB Adapter AWUS036AXML

<https://zsecurity.org/product/alfa-awus036ach-2-4-5-ghz-usb-wireless-adapter/>

Key Features

- **Dual-Band Support (2.4 & 5 GHz):** Offers flexibility for both longer range (2.4 GHz) and faster speeds (5GHz).
- **Kali Linux Support:** Fully compatible with Kali Linux whether it is installed as a main OS or as a virtual machine on Windows, Linux or Apple Mac OS.
- **Monitor Mode, AP Mode & Packet Injection Support:** Making it perfect for ethical hacking.
- **High Performance:** Supports up to 867 Mbps on 802.11ac networks, ideal for high-bandwidth tasks like streaming and gaming.
- **External 5dBi Antennas:** Provides better signal strength and range.

Applications

- **Network Auditing & Security Testing:** Used for testing the security of Wi-Fi networks.
- **Packet Sniffing & Capturing:** Capture network traffic to analyse data and identify weak points or network performance issues.
- **Deauthentication & Injection Attacks:** Perform deauthentication attacks and packet injection to simulate attacks and evaluate network security defenses.
- **Capture WPA/WPA2 Handshakes:** Essential for cracking WPA/WPA2 passwords by capturing the four-way handshake during a Wi-Fi connection.
- **Fake Access Point:** Start a fake AP or a honeypot to steal passwords and exfiltrate data.

Specifications

Brand: Alfa.

Model: AWUS036ACH.

Chipset: Realtek RTL8812AU.

Standards: IEEE 802.11 a/b/g/n/ac.

Data Rate:

802.11b: UP to 11Mbps.

802.11g: UP to 54Mbps.

802.11a: UP to 54Mbps.

802.11n: UP to 150Mbps.

802.11ac: UP to 867Mbps.

OS Supported: Kali Linux (whether it is installed as a virtual machine or a main OS).

Antenna: 2 x 5dBi 2.4GHz Antenna.

Frequency Range: 2.4 & 5 GHz.

Security: WEP 64/128, 802.1X support, Wi-Fi Protected Access (WPA/WPA2), WPA-PSK, WPA II-PSK.

Driver:

<https://files.alfa.com.tw/?dir=%5B1%5D%20WiFi%20USB%20adapter/AWUS036AXML>





Hardware

Xiaomi 13T

Firm: <https://mifirm.net/model/aristotle.ttt#>

Bloodhound Linux/Docker Quickstart

<https://bloodhound.specterops.io/get-started/quickstart/community-edition-quickstart>

```
wget https://github.com/Specter0ps/bloodhound-cli/releases/latest/download/bloodhound-cli-  
linux-amd64.tar.gz  
tar -xvzf bloodhound-cli-linux-amd64.tar.gz  
./bloodhound-cli install
```

Clone ESXi VMs

```
[root@esxi1:/opt/scripts] ./clonevm.sh master node3 datastore2
Checking source VM power state...
Creating destination directory: /vmfs/volumes/datastore2/node3
Cloning and converting the virtual disk...
Destination disk format: VMFS zeroedthick
Cloning disk '/vmfs/volumes/datastore2/master/master.vmdk'...
Clone: 100% done.
Copying and updating the VMX configuration...
Registering the new VM in the ESXi inventory...
[] Clone completed! The new VM is registered with ID: 261
Please power on the new VM and select 'I Copied it' if prompted.
```

USB Drive Read-Only Setup Guide

Run file [USB Write Protector.bat](#) as Administrator

```
=====
                        USB READ-ONLY MANAGER (DISKPART)
=====

Microsoft DiskPart version 10.0.26100.1150

Copyright (C) Microsoft Corporation.
On computer: ADS-PENTAHO

DISKPART>

  Disk ###  Status              Size       Free      Dyn  Gpt
  -----  -
  Disk 0    Online                476 GB    1024 KB           *
  Disk 1    Online                119 GB         0 B

DISKPART>
[1] Lock USB (Set Read-Only)
[2] Unlock USB (Make Writable)
[3] Check Current Status (Default)
[4] Exit

Choose an option (1-4) [Default: 3]: 1

Enter the DISK NUMBER: 1

Executing command set readonly for Disk 1...
```

Microsoft DiskPart version 10.0.26100.1150

Copyright (C) Microsoft Corporation.

On computer: ADS-PENTAHO

Disk 1 is now the selected disk.

Disk attributes set successfully.

SanDisk Extreme Pro USB Device

Disk ID: 00000000

Type : USB

Status : Online

Path : 0

Target : 0

LUN ID : 0

Location Path : UNAVAILABLE

Current Read-only State : Yes

Read-only : Yes

Boot Disk : No

Pagefile Disk : No

Hibernation File Disk : No

Crashdump Disk : No

Clustered Disk : No

Volume ###	Ltr	Label	Fs	Type	Size	Status	Info
Volume 3	D	SupMentos	NTFS	Removable	119 GB	Healthy	

=====
Command completed.
=====

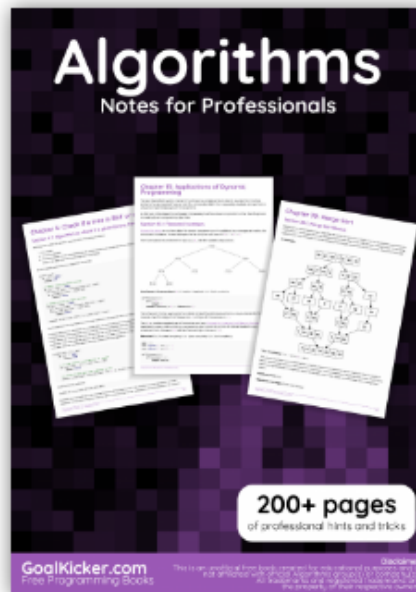
Press any key to continue . . .

Programming Notes for Professionals books

<https://books.goalkicker.com>



.NET Framework Notes for Professionals book



Algorithms Notes for Professionals book



Android® Notes for Professionals book



Angular 2 Notes for Professionals book



AngularJS Notes for Professionals book



Bash Notes for Professionals book

Top 25 Penetration Testing Skills and Competencies (Detailed)

<https://www.infosecmatter.com/top-25-penetration-testing-skills-and-competencies-detailed/>

<https://archive.sectigo.space/archive/1771996935.843272/index.html>

LLMs Locally

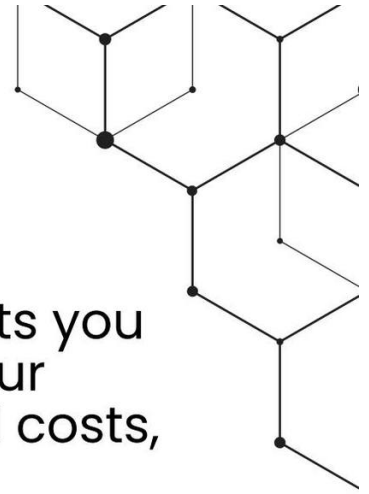
Run LLMs Locally

— with Ollama —



What is Ollama?

Your Local AI Powerhouse Ollama is a lightweight, extensible framework that lets you run large language models locally on your machine. No cloud dependencies, no API costs, complete privacy!

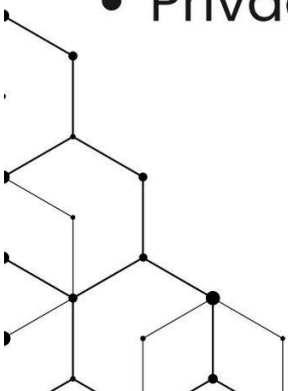


Key Features

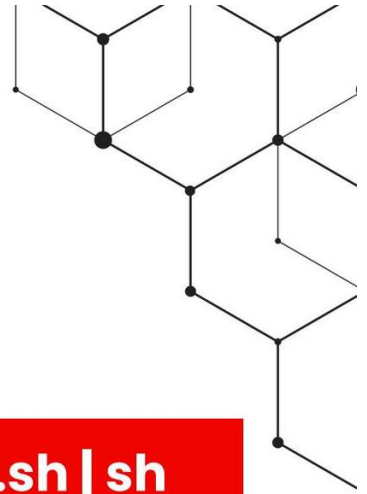
- Run models like Mistral, Llama, CodeLlama locally
- Easy installation and setup
- Multiple framework integrations
- Complete data privacy
- Offline capabilities

Perfect for

- Building RAG applications
- Creating AI agents
- Prototyping without API costs
- Privacy-sensitive projects



Installation Process



Quick Installation Steps

Linux:

```
curl -fsSL https://ollama.com/install.sh | sh
```

Download installer from ollama.com

Download Ollama



macOS

Linux

Windows

Download for macOS

Requires macOS 12 Monterey or later

Download Ollama



macOS

Linux

Windows

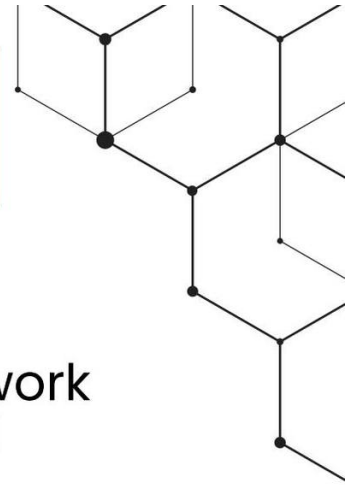
Download for Windows

Requires Windows 10 or later

Commands

```
ollama --version      # Check installed Ollama version
ollama serve          # Start Ollama service
ollama pull mistral   # Download Mistral model
ollama pull llama2    # Download LLaMA2 model
ollama pull codellama # Download CodeLLaMA model
ollama run mistral    # Test installation by running Mistral model
```

Framework Integration



Multiple Integration Options

Supported Frameworks:

- LangChain – Full-featured LLM framework
- LlamaIndex – Document indexing and retrieval
- Haystack – Enterprise-grade NLP pipelines
- Direct API – Pure HTTP requests

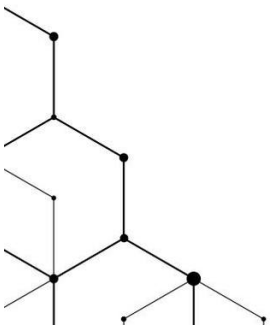
Installation Commands:

```
# LangChain
pip install langchain-community

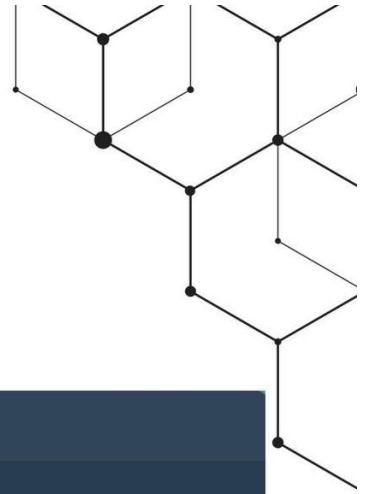
# LlamaIndex
pip install llama-index-llms-ollama

# Haystack
pip install ollama-haystack

# Direct API (requests)
pip install requests
```



LangChain Integration



LangChain + Ollama

Code Example:

```
from langchain_community.llms import Ollama

# Initialize Ollama LLM
llm = Ollama(model="mistral")

# Run inference
response = llm("Explain the difference between transformers and RNNs.")

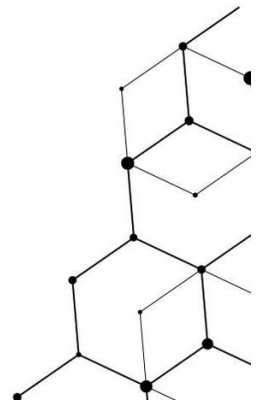
print(response)
```

Benefits

- Seamless integration with LangChain ecosystem
- Access to chains, agents, and tools
- Perfect for RAG applications
- Memory management built-in

Use Cases

- Chatbots with memory
- Document Q&A systems
- Multi-step reasoning tasks



LlamaIndex Integration

LlamaIndex + Ollama

Code Example:

```
from llama_index.llms.ollama import Ollama

# Initialize with custom settings
llm = Ollama(model="mistral", request_timeout=120.0, context_window=8000)

# Simple inference
response = llm.complete("Summarise India's independence movement in one paragraph.")

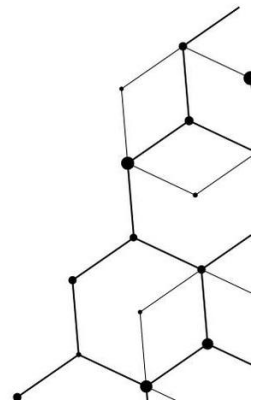
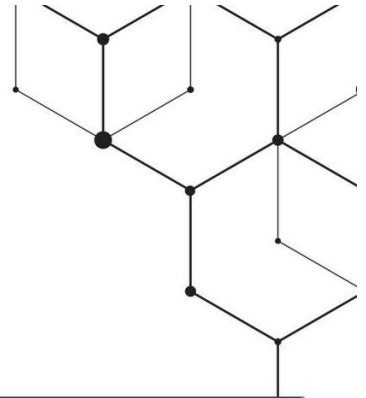
print(response)
```

Benefits

- Optimized for document indexing
- Built-in RAG capabilities
- Efficient context management
- Easy data ingestion

Use Cases

- Knowledge base systems
- Document search and retrieval
- Research assistants



Haystack Integration

Haystack + Ollama

Code Example:

```
from haystack_integrations.components.generators.ollama import OllamaChatGenerator
from haystack.dataclasses import ChatMessage

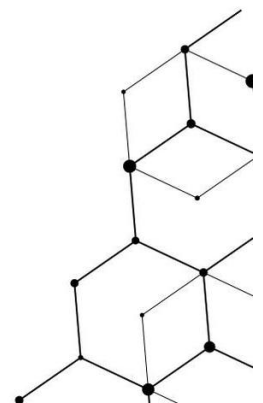
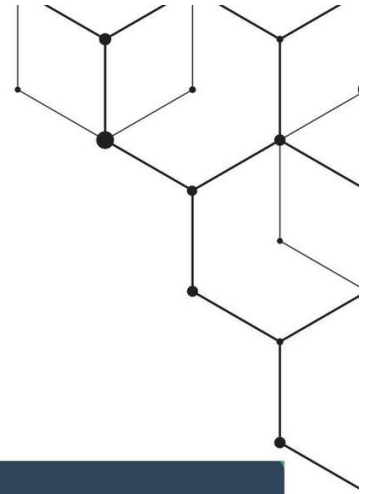
generator = OllamaChatGenerator( model="mistral", url="http://localhost:11434",
                                generation_kwargs={ "num_predict": 100, "temperature": 0.9, })

messages = [ ChatMessage.from_system("You are a helpful assistant"),
              ChatMessage.from_user("What's Natural Language Processing?")]

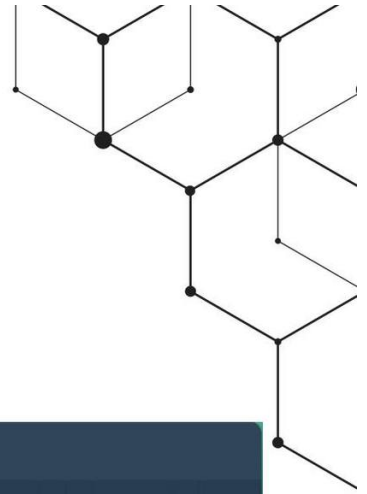
print(generator.run(messages=messages))
```

Benefits

- Enterprise-grade pipelines
- Advanced preprocessing
- Scalable architecture
- Production-ready



Direct API Integration



Direct API Access

Code Example:

```
import requests

url = "http://localhost:11434/api/generate"

prompt = "What is the capital of France?"

response = requests.post(url, json={ "model": "mistral", "prompt": prompt, "stream": False})

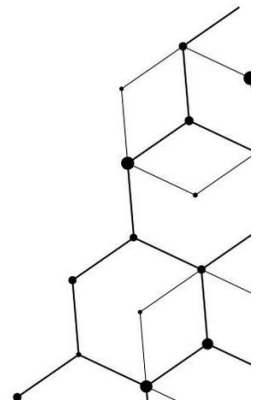
print(response.json()['response'])
```

Benefits

- Maximum control and flexibility
- Lightweight integration
- Custom request handling
- No additional dependencies

Use Cases

- Custom applications
- Microservices
- Simple integrations



Benefits & Advantages

✓ Why Choose Ollama?

Privacy & Security:

- Data never leaves your machine
- No cloud dependencies
- Complete control over your data
- GDPR/compliance friendly

Cost Efficiency:

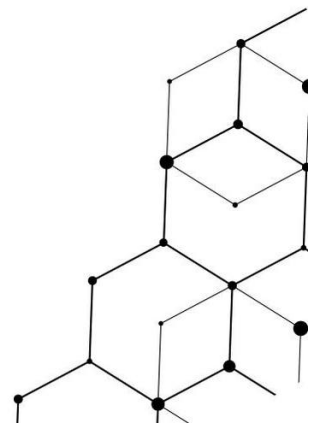
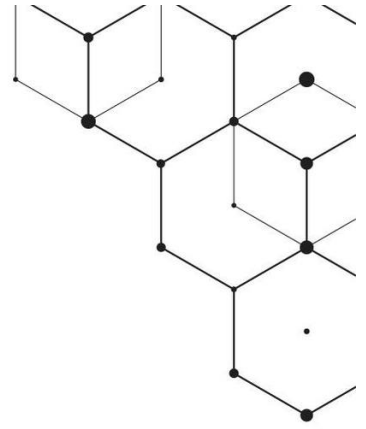
- No API fees or usage limits
- One-time setup cost
- Unlimited inference
- Perfect for experimentation

Performance:

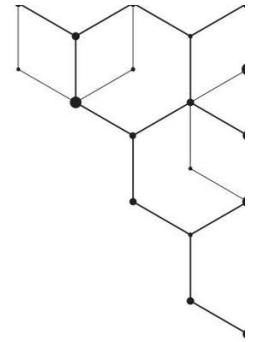
- Low latency responses
- Offline capabilities
- Consistent performance
- Hardware optimization

Flexibility:

- Multiple model choices
- Custom model fine-tuning
- Various framework integrations
- Open-source ecosystem



Popular Models



Choose the Right Model

deepseek-r1

DeepSeek-R1 is a family of open reasoning models with performance approaching that of leading models, such as O3 and Gemini 2.5 Pro.

[tools](#) [thinking](#) [1.5b](#) [7b](#) [8b](#) [14b](#) [32b](#) [70b](#) [671b](#)

↓ 52.1M Pulls ↻ 35 Tags ⌚ Updated 3 days ago

gemma3n

Gemma 3n models are designed for efficient execution on everyday devices such as laptops, tablets or phones.

[e2b](#) [e4b](#)

↓ 74.4K Pulls ↻ 9 Tags ⌚ Updated 1 week ago

gemma3

The current, most capable model that runs on a single GPU.

[vision](#) [1b](#) [4b](#) [12b](#) [27b](#)

↓ 7.7M Pulls ↻ 21 Tags ⌚ Updated 2 months ago

qwen3

Qwen3 is the latest generation of large language models in Qwen series, offering a comprehensive suite of dense and mixture-of-experts (MoE) models.

[tools](#) [thinking](#) [0.6b](#) [1.7b](#) [4b](#) [8b](#) [14b](#) [30b](#) [32b](#) [235b](#)

↓ 3.3M Pulls ↻ 35 Tags ⌚ Updated 1 month ago

qwen2.5vl

Flagship vision-language model of Qwen and also a significant leap from the previous Qwen2-VL.

[vision](#) [3b](#) [7b](#) [32b](#) [72b](#)

↓ 332.3K Pulls ↻ 17 Tags ⌚ Updated 1 month ago

llama3.1

Llama 3.1 is a new state-of-the-art model from Meta available in 8B, 70B and 405B parameter sizes.

[tools](#) [8b](#) [70b](#) [405b](#)

↓ 97M Pulls ↻ 93 Tags ⌚ Updated 7 months ago

llama3.2

Meta's Llama 3.2 goes small with 1B and 3B models.

[tools](#) [1b](#) [3b](#)

↓ 23.4M Pulls ↻ 63 Tags ⌚ Updated 9 months ago

mistral

The 7B model released by Mistral AI, updated to version 0.3.

[tools](#) [7b](#)

↓ 16.2M Pulls ↻ 84 Tags ⌚ Updated 2 weeks ago

qwen2.5

Qwen2.5 models are pretrained on Alibaba's latest large-scale dataset encompassing up to 18 trillion tokens. The model supports up to 128K tokens and has multilingual support.

[tools](#) [0.5b](#) [1.5b](#) [3b](#) [7b](#) [14b](#) [32b](#) [72b](#)

↓ 10.8M Pulls ↻ 133 Tags ⌚ Updated 9 months ago

llama3

Meta Llama 3: The most capable openly available LLM to date

[8b](#) [70b](#)

↓ 9M Pulls ↻ 68 Tags ⌚ Updated 1 year ago

llava

🔥 LLaVA is a novel end-to-end trained large multimodal model that combines a vision encoder and Vicuna for general-purpose visual and language understanding. Updated to version 1.6.

[vision](#) [7b](#) [13b](#) [34b](#)

↓ 7.4M Pulls ↻ 98 Tags ⌚ Updated 1 year ago

gemma2

Google Gemma 2 is a high-performing and efficient model available in three sizes: 2B, 9B, and 27B.

[2b](#) [9b](#) [27b](#)

↓ 5.7M Pulls ↻ 94 Tags ⌚ Updated 11 months ago

qwen2.5-coder

The latest series of Code-Specific Qwen models, with significant improvements in code generation, code reasoning, and code fixing.

[tools](#) [0.5b](#) [1.5b](#) [3b](#) [7b](#) [14b](#) [32b](#)

↓ 5.7M Pulls ↻ 199 Tags ⌚ Updated 1 month ago

Logon Remote MSSQL via Windows Account

```
PS C:\WINDOWS\system32> runas /netonly /user:DOMAIN\inon "C:\Program Files\Microsoft SQL Server Management Studio 22\Release\Common7\IDE\Ssms.exe"
```

Enter the password for DOMAIN\inon:

```
Attempting to start C:\Program Files\Microsoft SQL Server Management Studio 22\Release\Common7\IDE\Ssms.exe as user "DOMAIN\inon" ...
```

Server Name: Remote IP Server (keep default ALL)

