


Assignment 7 - Virtualization

Machine specs on Azure:

1. For use of virtualization in Linux its better to use a VM with higher performance.
2. So we create a VM with high performance,

Size * ⓘ 
[See all sizes](#)

Checking virtualization support on machine:

1. Once the VM is created we can check the virtualization support by `cpuinfo`, `cat /proc/cpuinfo`

```
azureuser@linux-lab-virtualization:~$ cat /proc/cpuinfo
processor       : 0
vendor_id      : GenuineIntel
cpu_family     : 6
model          : 85
model name     : Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz
stepping       : 7
microcode      : 0xffffffff
cpu MHz        : 2593.906
cache size     : 36608 KB
physical id    : 0
siblings       : 2
core id        : 0
cpu cores      : 1
apicid         : 0
initial apicid : 0
fpu            : yes
fpu_exception  : yes
cpuid level    : 21
wp             : yes
flags          : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ss ht syscall nx pdpe1gb rdtscp lm
mx ssse3 fma cx16 pcid sse4_1 sse4_2 movbe popcnt aes xsave avx f16c rdrand hypervisor lahf_lm abm 3dnowprefetch tpr_shadow ept vpid ept_ad fsgsbase
seed adx smap clflushopt clwb avx512cd avx512bw avx512vl xsaveopt xsavec xgetbv1 xsaves vnni avx512_vnni arch_capabilities
vmx flags      : vnni invpid ept_x_only ept_ad tsc_offset vtpr ept vpid unrestricted_guest
bugs           : spectre_v1 spectre_v2 spec_store_bypass swapgs taa mmio_stale_data retbleed bhi
bogomips       : 5187.81
clflush size   : 64
cache_alignment : 64
address sizes   : 46 bits physical, 48 bits virtual
power management:

processor       : 1
vendor_id      : GenuineIntel
cpu_family     : 6
model          : 85
model name     : Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz
stepping       : 7
microcode      : 0xffffffff
cpu MHz        : 2593.906
cache size     : 36608 KB
physical id    : 0
siblings       : 2
core id        : 0
cpu cores      : 1
```

2. Am I using 64 bit CPU/system [x86_64/AMD64/Intel64]? IN flags check for **lm**

```
Flags          : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ss ht syscall nx pdpe1gb rdtscp lm constant_tsc rep_good nopl xtopology cpuid pni pclmulqdq v
mx ssse3 fma cx16 pcid sse4_1 sse4_2 movbe popcnt aes xsave avx f16c rdrand hypervisor lahf_lm abm 3dnowprefetch tpr_shadow ept vpid ept_ad fsgsbase bmi1 hle avx2 smep bmi2 erms invpcid rtm avx512f avx512dq rd
```

3. Do I have hardware virtualization support?

- vmx – Intel VT-x, virtualization support enabled in BIOS.
- svm – AMD SVM, virtualization enabled in BIOS.

- ```
flags : fpu_ump de pse tsc msr pae mce apic sep mtrr pge nca cmov pat pse36 clflush mmx fxsr sse sse2 ss ht syscall nx pdpebgb rdtscll local_apic constant_tsc rep_good nopl topology
cpuid : 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
vmx : vmx=1 smx=1 estx12f avx12v4 sse4e1 sse4a1 popcnt aes xsave avx f16c rdrnd hypervisor lahf_lahf_lm avx-512vbq avx-512bw avx-512vl svm avx-512ifma avx-512cd avx-512dq rdseed adx snap clflushopt clwb avx512bw avx512bw_vl avx512bw_vl_avxopt xsavec xsavecvtgbi xsaVES vnni avx512_vnni arch_capabilities
```

aes – Applications performing encryption and decryption using the Advanced Encryption Standard on Intel and AMD cpus.

```
azureuser@linux-lab-virtualization:~$ grep --color aes /proc/cpuinfo
```

```
flags : fpu vme de pse tsc mrs pae csm x86 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ss ht syscall nm pdpeibg rdtscl pm constant_tsc rep_good nopl xtology
cpuid pcmlntdqw vms vss5f fmx avx16 pcid sse11 sse42 movbe popcnt aes xsave avx f16c rdrnd hypervisor lahf_lm abm 3dnowprefetch vrt_shadow ept vpid ept_ad fsbase bnil hle avx2 smep
bm2l erms invpcid rtm avx512f avx512dq rdseed adx smap clflushopt clwb avx512cd avx512bw avx512vl xsaveopt xsavec xgetbv1 xsaues vmni avx512_vnni arch_capabilities
```

```
flags : fpu vme de pse tsc mrs pae csm x86 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ss ht syscall nm pdpeibg rdtscl pm constant_tsc rep_good nopl xtology
cpuid pcmlntdqw vms vss5f fmx avx16 pcid sse11 sse42 movbe popcnt aes xsave avx f16c rdrnd hypervisor lahf_lm abm 3dnowprefetch vrt_shadow ept vpid ept_ad fsbase bnil hle avx2 smep
bm2l erms invpcid rtm avx512f avx512dq rdseed adx smap clflushopt clwb avx512cd avx512bw avx512vl xsaveopt xsavec xgetbv1 xsaues vmni avx512_vnni arch_capabilities
```

```
azureuser@linux-lab-virtualization:~$
```

1. `sudo apt install cpu-checker`

```
azureuser@linux-lab-virtualization:~$ sudo apt install cpu-checker
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
cpu-checker is already the newest version (0.7-1.3build2).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
azureuser@linux-lab-virtualization:~$
```

```
azureuser@linux-lab-virtualization:~$ sudo kvm-ok
INFO: /dev/kvm exists
KVM acceleration can be used
azureuser@linux-lab-virtualization:~$
```

1. update your system by using apt: `apt update && apt upgrade -y`

```

azureuser@linux-lab-virtualization:~$ sudo apt update && apt upgrade -y
Hit:1 http://azure.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://azure.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://azure.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://azure.archive.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 https://download.docker.com/linux/ubuntu noble InRelease [48.8 kB]
Get:6 http://azure.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [865 kB]
Get:7 http://azure.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [151 kB]
Get:8 http://azure.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1014 kB]
Get:9 http://azure.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [363 kB]
Get:10 http://azure.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]
Get:11 http://azure.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 B]
Get:12 http://azure.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [208 B]
Get:13 http://azure.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [17.7 kB]
Get:14 http://azure.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:15 http://azure.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Get:16 http://azure.archive.ubuntu.com/ubuntu noble-security/main amd64 Packages [617 kB]
Get:17 http://azure.archive.ubuntu.com/ubuntu noble-security/main amd64 Components [8956 B]
Get:18 http://azure.archive.ubuntu.com/ubuntu noble-security/universe amd64 Packages [803 kB]
Get:19 http://azure.archive.ubuntu.com/ubuntu noble-security/universe amd64 Components [51.9 kB]
Get:20 http://azure.archive.ubuntu.com/ubuntu noble-security/restricted amd64 Components [212 B]
Get:21 http://azure.archive.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [212 B]
Fetched 4321 kB in 1s (4167 kB/s)
Reading package lists... Done
Building dependency tree... Done

```

```

azureuser@linux-lab-virtualization:~$ apt upgrade -y
E: Could not open lock file /var/lib/dpkg/lock-frontent - open (13: Permission denied)
E: Unable to acquire the dpkg frontend lock (/var/lib/dpkg/lock-frontent), are you root?
azureuser@linux-lab-virtualization:~$ sudo apt upgrade -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages will be upgraded:
 krb5-locales libgssapi-krb5-2 libk5crypto3 libkrb5-3 libkrb5support openssl-client openssl-server
 openssl-sftp-server
8 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
8 standard LTS security updates
Need to get 2072 kB of archives.
After this operation, 4096 B of additional disk space will be used.
Get:1 http://azure.archive.ubuntu.com/ubuntu noble-updates/main amd64 libgssapi-krb5-2 amd64 1.20.1-6ubuntu2.4
[143 kB]
Get:2 http://azure.archive.ubuntu.com/ubuntu noble-updates/main amd64 libkrb5-3 amd64 1.20.1-6ubuntu2.4 [347 k
B]
Get:3 http://azure.archive.ubuntu.com/ubuntu noble-updates/main amd64 libkrb5support0 amd64 1.20.1-6ubuntu2.4
[33.9 kB]
Get:4 http://azure.archive.ubuntu.com/ubuntu noble-updates/main amd64 libk5crypto3 amd64 1.20.1-6ubuntu2.4 [81
.9 kB]
Get:5 http://azure.archive.ubuntu.com/ubuntu noble-updates/main amd64 openssl-sftp-server amd64 1:9.6p1-3ubuntu
u13.8 [37.3 kB]
Get:6 http://azure.archive.ubuntu.com/ubuntu noble-updates/main amd64 openssl-server amd64 1:9.6p1-3ubuntu13.8

```

```

Running kernel seems to be up-to-date.

Restarting services...
systemctl restart fwupd.service packagekit.service

No containers need to be restarted.

User sessions running outdated binaries:
azureuser @ session #1: sshd[1741]

No VM guests are running outdated hypervisor (qemu) binaries on this host.

```

2. check if you just updated your kernel or other systems needing a complete system reboot, and if so, reboot: `sudo reboot`

```

azureuser@linux-lab-virtualization:~$ sudo reboot

Broadcast message from root@linux-lab-virtualization on pts/1 (Wed 2025-02-19 11:29:50 EET):

The system will reboot now!

azureuser@linux-lab-virtualization:~$ client_loop: send disconnect: Connection reset
PS C:\Users\rashm> |

```

3. Install Snap: `sudo apt install snap -y`

```

azureuser@linux-lab-virtualization:~$ sudo apt install snap -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
snap is already the newest version (2013-11-29-11).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
azureuser@linux-lab-virtualization:~$ |

```

4. install lxd using snap: `sudo snap install lxd`

```

azureuser@linux-lab-virtualization:~$ sudo snap install lxd
lxd (5.21/stable) 5.21.3-75def3c from Canonical✓ installed
azureuser@linux-lab-virtualization:~$

```

5. check lxd version and installation: `lxd --version`

```

azureuser@linux-lab-virtualization:~$ lxd --version
5.21.3 LTS
azureuser@linux-lab-virtualization:~$ |

```

6. check that your user belongs to LXD group: `id`, and look for LXD. If you do not find lxd group, add user to it: `sudo usermod -aG lxd $USER`

```

azureuser@linux-lab-virtualization:~$ sudo usermod -aG lxd $USER
azureuser@linux-lab-virtualization:~$ id
uid=1000(azureuser) gid=1000(azureuser) groups=1000(azureuser),4(adm),24(cdrom),27(sudo),30(dip),105(lxd),988(docker)
azureuser@linux-lab-virtualization:~$

```

7. check lxc system for listing of machines and containers: `lxc list`

```

azureuser@linux-lab-virtualization:~$ lxc list
If this is your first time running LXD on this machine, you should also run: lxd init
To start your first container, try: lxc launch ubuntu:24.04
Or for a virtual machine: lxc launch ubuntu:24.04 --vm

+-----+-----+-----+-----+-----+-----+
| NAME | STATE | IPV4 | IPV6 | TYPE | SNAPSHOTS |
+-----+-----+-----+-----+-----+-----+
azureuser@linux-lab-virtualization:~$

```

8. initialize lxd, to configure system to your environment: lxd init. Make sure you run this as basic user, without root / sudo privileges it will ask series of questions, so based on your requirement answer them.
  - When it asks about clustering, choose 'no' (unless you're setting up a cluster)
  - For storage, I recommend saying 'yes' to a new storage pool
  - The 'dir' backend is fine for beginners
  - Say 'yes' to a network bridge
  - Choose 'no' to make LXD sever available over the network.

```

azureuser@linux-lab-virtualization:~$ lxd init
Would you like to use LXD clustering? (yes/no) [default=no]:
Do you want to configure a new storage pool? (yes/no) [default=yes]:
Name of the new storage pool [default=default]:
Name of the storage backend to use (dir, lvm, powerflex, zfs, btrfs, ceph) [default=zfs]: dir
Would you like to connect to a MAAS server? (yes/no) [default=no]:
Would you like to create a new local network bridge? (yes/no) [default=yes]:
What should the new bridge be called? [default=lxdbr0]:
What IPv4 address should be used? (CIDR subnet notation, "auto" or "none") [default=auto]:
What IPv6 address should be used? (CIDR subnet notation, "auto" or "none") [default=auto]:
Would you like the LXD server to be available over the network? (yes/no) [default=no]:
Would you like stale cached images to be updated automatically? (yes/no) [default=yes]:
Would you like a YAML "lxd init" preseed to be printed? (yes/no) [default=no]:
azureuser@linux-lab-virtualization:~$

```

9. Once lxd is initialized successfully, we can verify the information using following set of commands:

\$lxc profile list \$lxc network list \$lxc storage list

```

azureuser@linux-lab-virtualization:~$ lxc profile list
+-----+-----+-----+
| NAME | DESCRIPTION | USED BY |
+-----+-----+-----+
| default | Default LXD profile | 0 |
+-----+-----+-----+

azureuser@linux-lab-virtualization:~$ lxc network list
+-----+-----+-----+-----+-----+-----+-----+-----+
| NAME | TYPE | MANAGED | IPV4 | IPV6 | DESCRIPTION | USED BY | STATE |
+-----+-----+-----+-----+-----+-----+-----+-----+
| br-70b4198ba5ec | bridge | NO | | | | 0 | |
+-----+-----+-----+-----+-----+-----+-----+-----+
| docker0 | bridge | NO | | | | 0 | |
+-----+-----+-----+-----+-----+-----+-----+-----+
| enP12319s1 | physical | NO | | | | 0 | |
+-----+-----+-----+-----+-----+-----+-----+-----+
| eth0 | physical | NO | | | | 0 | |
+-----+-----+-----+-----+-----+-----+-----+-----+
| lxdbr0 | bridge | YES | 10.181.48.1/24 | fd42:e703:354c:3aae::1/64 | | 1 | CREATED |
+-----+-----+-----+-----+-----+-----+-----+-----+

azureuser@linux-lab-virtualization:~$ lxc storage list
+-----+-----+-----+-----+-----+-----+
| NAME | DRIVER | SOURCE | DESCRIPTION | USED BY | STATE |
+-----+-----+-----+-----+-----+-----+
| default | dir | /var/snap/lxd/common/lxd/storage-pools/default | | 1 | CREATED |
+-----+-----+-----+-----+-----+-----+

azureuser@linux-lab-virtualization:~$

```

10. In order to list all available images, run:

\$lxc image list images:

```

azureuser@linux-lab-virtualization:~$ lxc image list images
+-----+-----+-----+-----+-----+-----+-----+-----+
| ALIAS | FINGERPRINT | PUBLIC | DESCRIPTION | ARCHITECTURE | TYPE | SIZE | UPLOAD DATE |
+-----+-----+-----+-----+-----+-----+-----+-----+
azureuser@linux-lab-virtualization:~$

```

11. Create your first container:

```
$lxc launch ubuntu:24.04 demo-container
```

```

azureuser@linux-lab-virtualization:~$ lxc launch ubuntu:24.04 demo-container
Launching demo-container
azureuser@linux-lab-virtualization:~$ |

```

12. Access the console of container. Run:

```
$ lxc exec demo-container -- bash
```

```

azureuser@linux-lab-virtualization:~$ lxc exec demo-container -- bash
root@demo-container:~# |

```

## Installing and managing Docker engine based system

### Basic steps:

Follow good instructions from Docker, at <https://docs.docker.com/>

```

azureuser@linux-lab-virtualization:~$ lxc exec demo-container -- bash
root@demo-container:~# apt update & apt install nginx -y
[1] 652
Reading package lists... Done
Building dependency tree... Done.com (2620:2d:4002:1::103)] [Connecting to security.ubuntu.com (2620:2d:4000:1::101)]
Reading state information... Done
The following additional packages will be installed:
 nginx-common
Suggested packages:
 fcgiwrap nginx-doc ssl-cert
The following NEW packages will be installed:
 nginx nginx-common
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 552 kB of archives.
After this operation, 1596 kB of additional disk space will be used.
Ign:1 http://security.ubuntu.com/ubuntu noble-security InRelease
Ign:2 http://archive.ubuntu.com/ubuntu noble InRelease
Ign:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Ign:4 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Ign:1 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 nginx-common all 1.24.0-2ubuntu7.1
Ign:2 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 nginx amd64 1.24.0-2ubuntu7.1
Ign:1 http://security.ubuntu.com/ubuntu noble-security InRelease
Ign:2 http://archive.ubuntu.com/ubuntu noble InRelease
Ign:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Ign:4 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Ign:1 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 nginx-common all 1.24.0-2ubuntu7.1
Ign:2 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 nginx amd64 1.24.0-2ubuntu7.1
Ign:1 http://security.ubuntu.com/ubuntu noble-security InRelease

```

More specifically from Ubuntu engine install instructions for Ubuntu system, and using convenience script:

<https://docs.docker.com/engine/install/ubuntu/#install-using-the-convenience-script>

```
curl -fsSL https://get.docker.com -o get-docker.sh
```

```
sudo sh get-docker.sh
```

Executing docker install script, commit: 7cae5f8b0decc17d6571f9f52eb840fbc13b2737

<...>

```
azureuser@linux-lab-virtualization:~$ curl -fsSL https://get.docker.com -o get-docker.sh
azureuser@linux-lab-virtualization:~$ sudo sh ./get-docker.sh --dry-run
Executing docker install script, commit: 4c94a56999e10efcf48c5b8e3f6afea464f9108e
Warning: the "docker" command appears to already exist on this system.

If you already have Docker installed, this script can cause trouble, which is
why we're displaying this warning and provide the opportunity to cancel the
installation.

If you installed the current Docker package using this script and are using it
again to update Docker, you can ignore this message, but be aware that the
script resets any custom changes in the deb and rpm repo configuration
files to match the parameters passed to the script.

You may press Ctrl+C now to abort this script.
+ sleep 20

^C
azureuser@linux-lab-virtualization:~$ |
```

post installation guide tells us to make user part of Docker group on Linux machine:

sudo groupadd docker

sudo usermod -aG docker \$USER

newgrp docker

```
azureuser@linux-lab-virtualization:~$ sudo groupadd docker
groupadd: group 'docker' already exists
azureuser@linux-lab-virtualization:~$ sudo usermod -aG docker $USER
azureuser@linux-lab-virtualization:~$ newgrp docker
azureuser@linux-lab-virtualization:~$ id
uid=1000(azureuser) gid=988(docker) groups=988(docker),4(adm),24(cdrom),27(sudo),30(dip),105(lxd),1000(azureuser)
azureuser@linux-lab-virtualization:~$
```

check docker version

```
azureuser@linux-lab-virtualization:~$ docker version
Client: Docker Engine - Community
Version: 27.5.1
API version: 1.47
Go version: go1.22.11
Git commit: 9f9e405
Built: Wed Jan 22 13:41:48 2025
OS/Arch: linux/amd64
Context: default

Server: Docker Engine - Community
Engine:
Version: 27.5.1
API version: 1.47 (minimum version 1.24)
Go version: go1.22.11
Git commit: 4c9b3b0
Built: Wed Jan 22 13:41:48 2025
OS/Arch: linux/amd64
Experimental: false
containerd:
Version: 1.7.25
GitCommit: bcc810d6b9066471b0b6fa75f557a15a1cbf31bb
runc:
Version: 1.2.4
GitCommit: v1.2.4-0-g6c52b3f
docker-init:
Version: 0.19.0
GitCommit: de40ad0
azureuser@linux-lab-virtualization:~$
```

## Run Nginx on docker:

1. Get the latest Nginx

```
azureuser@linux-lab-virtualization:~$ docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
Digest: sha256:91734281c0ebfc6f1aea979cffeed5079cfe786228a71cc6f1f46a228cde6e34
Status: Image is up to date for nginx:latest
docker.io/library/nginx:latest
azureuser@linux-lab-virtualization:~$
```

2. Start docker nginx image
  - docker run -p 80:80 nginx
  - docker run -d -p 80:80 nginx (To run on background use -d tag)



```
azureuser@linux-lab-virtualization:~$ docker run -p 80:80 nginx
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2025/02/19 10:26:22 [notice] 1#1: using the "epoll" event method
2025/02/19 10:26:22 [notice] 1#1: nginx/1.27.4
2025/02/19 10:26:22 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)
2025/02/19 10:26:22 [notice] 1#1: OS: Linux 6.8.0-1021-azure
2025/02/19 10:26:22 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2025/02/19 10:26:22 [notice] 1#1: start worker processes
2025/02/19 10:26:22 [notice] 1#1: start worker process 29
2025/02/19 10:26:22 [notice] 1#1: start worker process 30
```

3. If we open the IP of our virtual machine we can see Nginx is running,

← → ↻ ⚠ Not secure 104.45.95.208

## Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](https://nginx.org).  
Commercial support is available at [nginx.com](https://nginx.com).

*Thank you for using nginx.*