

NTP система в docker

Данный репозиторий демонстрирует демонстрационный проект, подготовленный в рамках выполнения курсовой работы по предмету "Системное Программное Обеспечение" студентом МФ МГТУ им. Н. Э. Баумана группы КЗ-56Б Зозулей Артемом

Ниже будет приведена инструкция по работе с демонстрационным проектом

Прerequisites

Необходимо иметь установленный и запущенный движок docker на хосте, а также сопутствующие пакеты, такие как docker-compose-v2, docker-buildx-plugin

Весь набор пакетов, связанных с docker, можно установить по инструкции с [официального сайта Docker](#)

Развертывание

Склонируйте данный репозиторий локально на хост, где будет проходить работа со стендом

Первым шагом необходимо запустить тестовый стенд с помощью `docker compose`, для этого подготовлен скрипт `run.sh`

```
> ./run.sh
Building Docker image...
[+] Building 1.6s (9/9) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 597B
=> [internal] load metadata for docker.io/library/alpine:edge
=> [auth] library/alpine:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/3] FROM docker.io/library/alpine:edge@sha256:b93f4f6834d5c6849d859a4c07cc88f5a7d8ce5fb8
=> => resolve docker.io/library/alpine:edge@sha256:b93f4f6834d5c6849d859a4c07cc88f5a7d8ce5fb8
=> [internal] load build context
=> => transferring context: 32B
=> CACHED [2/3] RUN apk add --no-cache chrony tzdata &&      rm /etc/chrony/chrony.conf &&
=> CACHED [3/3] COPY ./startup.sh /bin/startup
=> exporting to image
=> => exporting layers
=> => exporting manifest sha256:adeeb9faedbedcb8d2f32f0f0629665ad71c123ad42cfe4b9c0cbb021bcfd
=> => exporting config sha256:0f3b3552558146eae383e4d52d0a5c23750e314695648cca3f193c9e68c8412
=> => exporting attestation manifest sha256:529d4211824823f62a7a60660f0ab03a7e097ff73f1cdeabb
=> => exporting manifest list sha256:8c28a406d34230e352a1fe5976bc4f0aa7e7e4df6ad86a71833989f6
```

```
=> => naming to docker.io/custom/chrony:local
=> => unpacking to docker.io/custom/chrony:local
Running docker compose up -d...
WARN[0000] /home/user404/ntp-docker/docker-compose.yaml: the attribute `version` is obsolete,
[+] Running 5/5
✓ Network ntp-docker_ntp_network Created
✓ Container chrony-local1 Started
✓ Container chrony-local3 Started
✓ Container chrony-local2 Started
✓ Container chrony-main Started
```

Скрипт выполняет следующие шаги:

- Проверяет наличие нужных пакетов в системе
- Собирает локальный docker-образ с ntp-сервером chrony
- запускает сервисы (набор ntp-серверов на основе собранного ранее docker-образа) из конфигурации `docker-compose.yaml`

После успешного выполнения скрипта можно посмотреть на состояние запущенных контейнеров с помощью команды:

```
> docker ps
CONTAINER ID   IMAGE                COMMAND                  CREATED        STATUS
42667d543569   custom/chrony:local  "/bin/startup"          7 minutes ago Up 7 minutes (healthy)
90866ca3510b   custom/chrony:local  "/bin/startup"          7 minutes ago Up 7 minutes (healthy)
06af3035df87   custom/chrony:local  "/bin/startup"          7 minutes ago Up 7 minutes (healthy)
d8fd3be21bd3   custom/chrony:local  "/bin/startup"          7 minutes ago Up 7 minutes (healthy)
```

Эксплуатация

После успешного развертывания стенда предлагается изучить параметры его компонентов.

Чтобы посмотреть информацию вышестоящего стратума для каждого из трех контейнеров `chrony-local<n>` можно выполнить следующие команды:

```
> docker exec chrony-local1 chronyc sources
MS Name/IP address          Stratum Poll Reach LastRx Last sample
=====
^* 162.159.200.123           3    6   377    9   -32ms[ -76ms] +/- 11ms
> docker exec chrony-local1 chronyc sourcestats
Name/IP Address            NP  NR  Span Frequency  Freq Skew  Offset  Std Dev
=====
162.159.200.123            9   4   517   -3.271    573.415   -33us    59ms
>
> docker exec chrony-local1 chronyc tracking
Reference ID      : A29FC87B (162.159.200.123)
```

```

Stratum          : 4
Ref time (UTC)   : Thu Oct 31 22:52:09 2024
System time      : 0.118983239 seconds slow of NTP time
Last offset      : +0.061203212 seconds
RMS offset       : 0.137246609 seconds
Frequency        : 537.010 ppm fast
Residual freq    : +114.795 ppm
Skew             : 694.085 ppm
Root delay       : 0.025834698 seconds
Root dispersion  : 0.041465286 seconds
Update interval  : 65.0 seconds
Leap status      : Norma
>
> docker exec chrony-local2 chronyc sources
MS Name/IP address      Stratum Poll Reach LastRx Last sample
=====
^* 162.159.200.123      3    6   377   49    -86ms[ -108ms] +/-   24ms
> docker exec chrony-local2 chronyc sourcestats
Name/IP Address          NP  NR  Span  Frequency  Freq Skew  Offset  Std Dev
=====
162.159.200.123          9   4   518    -15.460    604.201  -809us   45ms
>
> docker exec chrony-local2 chronyc tracking
Reference ID      : A29FC87B (162.159.200.123)
Stratum          : 4
Ref time (UTC)   : Thu Oct 31 22:52:09 2024
System time      : 0.233108655 seconds slow of NTP time
Last offset      : -0.063350037 seconds
RMS offset       : 0.137813374 seconds
Frequency        : 15.096 ppm fast
Residual freq    : +59.821 ppm
Skew             : 903.832 ppm
Root delay       : 0.022655277 seconds
Root dispersion  : 0.034543112 seconds
Update interval  : 64.3 seconds
Leap status      : Norma

```

Как мы видим, все 3 chrony-сервера действительно используют указанный в конфигурации сервер `time.cloudflare.com` и активны

```

> dig -x 162.159.200.123

; <<>> DiG 9.18.1-1ubuntu1.3-Ubuntu <<>> -x 162.159.200.123
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 10355
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:

```

```
;123.200.159.162.in-addr.arpa. IN PTR
;; ANSWER SECTION:
123.200.159.162.in-addr.arpa. 472 IN PTR time.cloudflare.com.

;; Query time: 9 msec
;; SERVER: 10.255.255.254#53(10.255.255.254) (UDP)
```

Обратимся к подобной информации у основного контейнера, который выступает gateway-ем для использования стэнда:

```
> docker exec chrony-main chronyc sources
MS Name/IP address          Stratum Poll Reach LastRx Last sample
=====
^+ chrony-local1.ntp-docker> 4 6 377 26 -49ms[ -339ms] +/- 55ms
^* chrony-local2.ntp-docker> 4 6 377 25 -75ms[ -365ms] +/- 38ms
^- chrony-local3.ntp-docker> 4 6 377 22 -137ms[ -137ms] +/- 26ms
> docker exec chrony-main chronyc sourcestats
Name/IP Address            NP NR Span Frequency Freq Skew Offset Std Dev
=====
chrony-local1.ntp-docker> 6 3 321 -2984.880 2917.655 -55ms 93ms
chrony-local2.ntp-docker> 6 3 323 -3247.837 2166.339 -125ms 74ms
chrony-local3.ntp-docker> 6 3 323 -3073.639 2771.274 -154ms 95ms
>
> docker exec chrony-main chronyc tracking
Reference ID      : AC120002 (chrony-local2.ntp-docker_ntp_network)
Stratum          : 5
Ref time (UTC)   : Thu Oct 31 22:54:24 2024
System time      : 0.000034495 seconds fast of NTP time
Last offset      : -0.290164471 seconds
RMS offset       : 0.198053524 seconds
Frequency        : 7242.182 ppm fast
Residual freq    : -3154.388 ppm
Skew             : 1105.048 ppm
Root delay       : 0.049896758 seconds
Root dispersion  : 0.219717875 seconds
Update interval  : 64.6 seconds
Leap status      : Normal
```

По результатам выполнения команд выше можно убедиться в том, что основной контейнер `chrony-main` обращается к трем локальным и обладает stratum-ом большим, чем у них.

Также можно посмотреть логи этого сервера командой:

```
> docker logs -t chrony-main
2024-10-31T22:31:34.134814336Z 2024-10-31T22:31:34Z chronyd version 4.6.1 starting (+CMDMON +N
2024-10-31T22:31:34.135115635Z 2024-10-31T22:31:34Z Could not read valid frequency and skew fr
2024-10-31T22:31:34.135125488Z 2024-10-31T22:31:34Z Initial frequency 13747.737 ppm
2024-10-31T22:31:44.867687789Z 2024-10-31T22:31:44Z Selected source 172.18.0.4 (chrony.local1)
```

```

2024-10-31T22:36:06.149943015Z 2024-10-31T22:36:06Z Selected source 172.18.0.3 (chrony.local3)
2024-10-31T22:43:37.807462431Z 2024-10-31T22:43:37Z Selected source 172.18.0.2 (chrony.local2)
2024-10-31T22:43:40.238584139Z 2024-10-31T22:43:40Z Detected falseticker 172.18.0.3 (chrony.lo
2024-10-31T22:52:15.350455882Z 2024-10-31T22:52:15Z Selected source 172.18.0.4 (chrony.local1)
2024-10-31T22:52:17.688273436Z 2024-10-31T22:52:17Z Detected falseticker 172.18.0.4 (chrony.lo
2024-10-31T22:52:17.688292851Z 2024-10-31T22:52:17Z Selected source 172.18.0.2 (chrony.local2)
2024-10-31T22:56:33.646077487Z 2024-10-31T22:56:33Z Selected source 172.18.0.4 (chrony.local1)

```

Использование стенда как ntp-сервера на хосте

Для начала убедимся, что chrony-сервер внутри контейнера chrony-main доступен с хост-машины. Для этого нам понадобятся следующие пакеты - `ntpd`, `ntp`

Нужно определить IP-адрес контейнера chrony-main в виртуальной docker-bridge-сети. Получаем ее id, а затем определяем IP-адрес нужного контейнера:

```

> docker network ls | grep ntp
7986b2a49483    ntp-docker_ntp_network    bridge    local
> export CID="chrony-main" && docker inspect $CID | grep IPAddress | grep -v null | cut -d '"'
172.19.0.5
# IP-адрес будет отличаться

```

Теперь выполним команду, которая выполнит запрос к данному ntp-серверу по IP:

```

> ntpdate -q 172.19.0.5
server 172.19.0.5, stratum 5, offset +0.020142, delay 0.02571
1 Nov 02:31:01 ntpdate[55736]: adjust time server 172.19.0.5 offset +0.020142 sec

```

Убедившись, что сервер в контейнере отвечает с локальной машины, мы можем установить его для запущенного на хост-машине ntp-сервера. Чтобы узнать, запущен ли он, можно выполнить следующую команду:

```

> systemctl list-units --type=service | grep ntp && systemctl status ntp
ntp.service                                loaded active running Network Time Service
• ntp.service - Network Time Service
   Loaded: loaded (/lib/systemd/system/ntp.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2024-10-31 23:25:38 MSK; 3h 10min ago
     Docs: man:ntpd(8)
  Process: 27714 ExecStart=/usr/lib/ntp/ntp-systemd-wrapper (code=exited, status=0/SUCCESS)
 Main PID: 27720 (ntpd)
    Tasks: 2 (limit: 4570)
   Memory: 1.3M
      CPU: 611ms
   CGroup: /system.slice/ntp.service
           └─27720 /usr/sbin/ntpd -p /var/run/ntpd.pid -g -u 115:120

```

```
Oct 31 23:25:38 ntp-test2 ntpd[27720]: Listen normally on 7 br-2b664f08e709 [fe80::42:a2ff:fe4
Oct 31 23:25:38 ntp-test2 ntpd[27720]: Listen normally on 8 veth1b0a614 [fe80::88d4:2dff:fe94:
Oct 31 23:25:38 ntp-test2 ntpd[27720]: Listen normally on 9 veth75c4721 [fe80::4482:9bff:fe2b:
Oct 31 23:25:38 ntp-test2 ntpd[27720]: Listen normally on 10 veth3275597 [fe80::3cd9:4eff:fe1f
Oct 31 23:25:38 ntp-test2 ntpd[27720]: Listen normally on 11 veth0f6632c [fe80::4c0:2aff:fe4d:
Oct 31 23:25:38 ntp-test2 ntpd[27720]: Listening on routing socket on fd #28 for interface upd
Oct 31 23:25:38 ntp-test2 ntpd[27720]: kernel reports TIME_ERROR: 0x41: Clock Unsynchronized
Oct 31 23:25:38 ntp-test2 ntpd[27720]: kernel reports TIME_ERROR: 0x41: Clock Unsynchronized
Oct 31 23:25:38 ntp-test2 systemd[1]: Started Network Time Service.
```

Если мы получили аналогичный вывод, сообщающий, что данный сервис существует и активен, то можно переходить к изменению его конфигурации - отредактируем файл `/etc/ntp.conf`, указав в соответствующей части IP-адрес нашего chrony-сервера в Docker вместо всех других:

```
...
# Specify one or more NTP servers.
pool 172.19.0.5
...
```

После чего перезапускаем сервис и проверяем изменившиеся настройки:

```
> systemctl restart ntp
> ntpq -p
```

| remote | refid | st | t | when | poll | reach | delay | offset | jitter |
|------------|------------|----|---|------|------|-------|-------|--------|--------|
| 172.19.0.5 | .POOL. | 16 | p | - | 64 | 0 | 0.000 | +0.000 | 0.000 |
| 172.19.0.5 | 172.19.0.2 | 5 | u | 3 | 64 | 1 | 0.040 | +7.857 | 0.000 |

```
> date
Fri Nov 1 02:41:15 AM MSK 202
```

Использование стенда как ntp-сервера для приложений

Для демонстрации функционала использования стенда как ntp-сервера для приложений было разработано примитивное приложение на Python, которое мы запустим в контейнере внутри одной docker-bridge-сети со стендом, для этого выполним следующую команду находясь в директории клонированного репозитория:

```
> cd ntp-user/ && docker build -t custom/ntp-user:local . && docker run --network ntp-docker_n
-container custom/ntp-user:local chrony.main --interval 5
[+] Building 0.9s (10/10) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 206B
=> [internal] load metadata for docker.io/library/python:3.12.2-slim
=> [internal] load .dockerignore
```

```

=> => transferring context: 2B
=> [1/5] FROM docker.io/library/python:3.12.2-slim@sha256:5dc6f84b5e97bfb0c90abfb7c55f3cacc2c
=> => resolve docker.io/library/python:3.12.2-slim@sha256:5dc6f84b5e97bfb0c90abfb7c55f3cacc2c
=> [internal] load build context
=> => transferring context: 148.95kB
=> CACHED [2/5] WORKDIR /app
=> CACHED [3/5] COPY requirements.txt .
=> CACHED [4/5] RUN pip install --no-cache-dir -r requirements.txt
=> CACHED [5/5] COPY . .
=> exporting to image
=> => exporting layers
=> => exporting manifest sha256:ab64f6e03f52ea0444bca6829197f60d10fd07e388e40f87dcd45e2fd4391
=> => exporting config sha256:729434fd93830c23518347c354de475b19e966e3fa218bcb30ab0e430df78b6
=> => exporting attestation manifest sha256:6290375922052bf250a56a45289f25a3f6dbdc1dd19dc04ed
=> => exporting manifest list sha256:371f372c4fd5112c98e6b4803785445366cff88d4431853a5c6776dd
=> => naming to docker.io/custom/ntp-user:local
=> => unpacking to docker.io/custom/ntp-user:local
2024-10-31 23:49:31,799 - INFO - Fetching time from chrony.main every 5 seconds
2024-10-31 23:49:36,993 - INFO - -----
2024-10-31 23:49:36,993 - INFO - NTP Time: 2024-10-31 23:49:37.070570
2024-10-31 23:49:36,993 - INFO - Time difference: 5.183922 seconds
2024-10-31 23:49:36,994 - INFO - Average time: 2024-10-31 23:49:34.478609
2024-10-31 23:49:36,994 - INFO - Accuracy: 5.183922 seconds
2024-10-31 23:49:41,995 - INFO - -----
2024-10-31 23:49:41,995 - INFO - NTP Time: 2024-10-31 23:49:42.064291
2024-10-31 23:49:41,995 - INFO - Time difference: 4.993721 seconds
2024-10-31 23:49:41,995 - INFO - Average time: 2024-10-31 23:49:39.567430
2024-10-31 23:49:41,995 - INFO - Accuracy: 4.993721 seconds
...

```

Приложение начнет выводить время и ряд статистических данных, полученные с chrony-сервера chrony.main с заданным интервалом (5 секунд). Тем самым демонстрируется взаимодействие приложений с ntp-серверов внутри Docker-контейнеров.