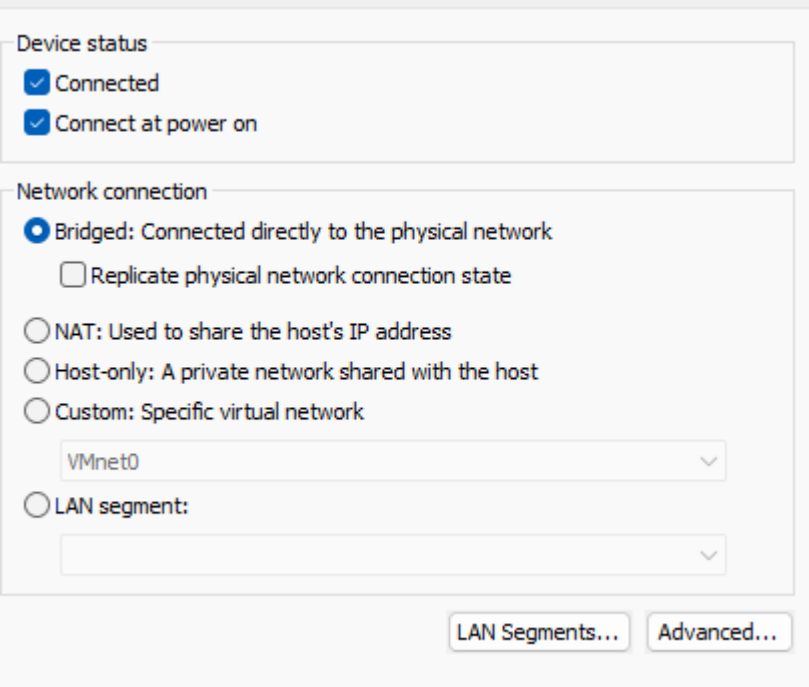
Практическая работа 5

Настройка Nginx, конфигурация Upstreams

1. Базовое REST приложение

```
from flask import Flask
import sys

app = Flask(__name__)

@app.route("/")
def hello():
    return "Hello, World!"

if __name__ == "__main__":
    port = int(sys.argv[1]) if len(sys.argv) > 1 else 5000
    app.run(host="127.0.0.1", port=port)
```

2. Использовал Vmware место Virtualbox.

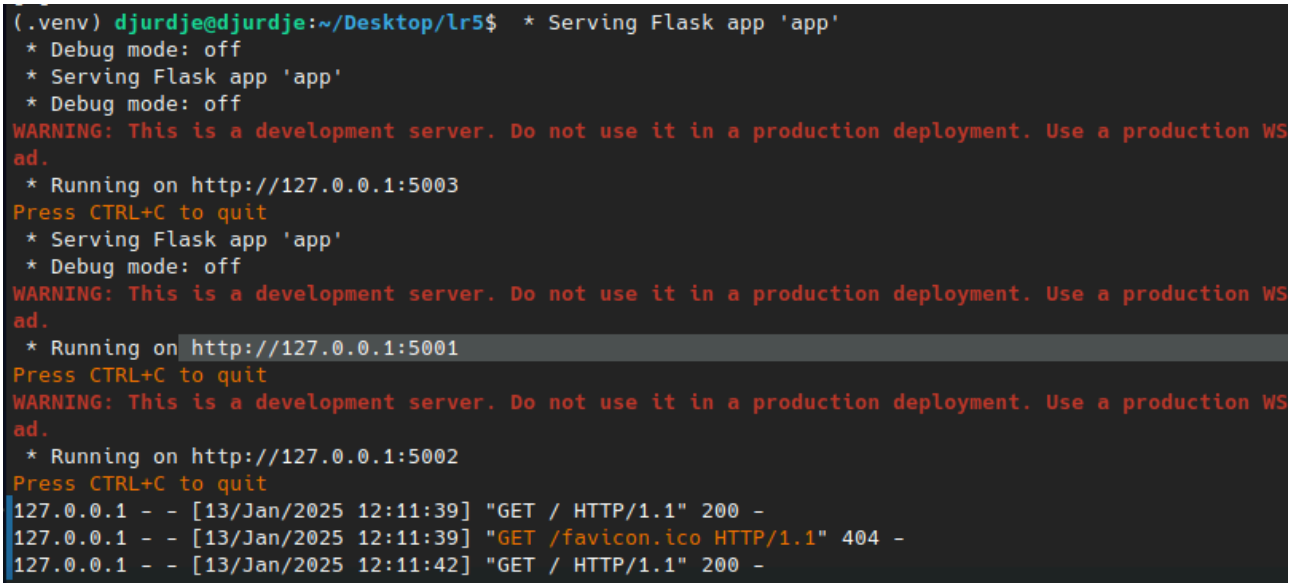


Vmware -> 192.168.0.11

Пинг с хостовой ОС Windows:

```
C:\Users\djurd>ping 192.168.0.11

Pinging 192.168.0.11 with 32 bytes of data:
Reply from 192.168.0.11: bytes=32 time<1ms TTL=64
Reply from 192.168.0.11: bytes=32 time<1ms TTL=64
Reply from 192.168.0.11: bytes=32 time<1ms TTL=64
Reply from 192.168.0.11: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.0.11:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

3. 3 реплики приложения на интерфейсе **127.0.0.1** (localhost), чтобы реплики были доступны только внутри виртуальной машины. Это гарантирует, что они не будут напрямую доступны с хостовой машины.



4. nginx

Сначала нужно установить nginx

```
sudo apt install nginx
```

nginx конфигурация

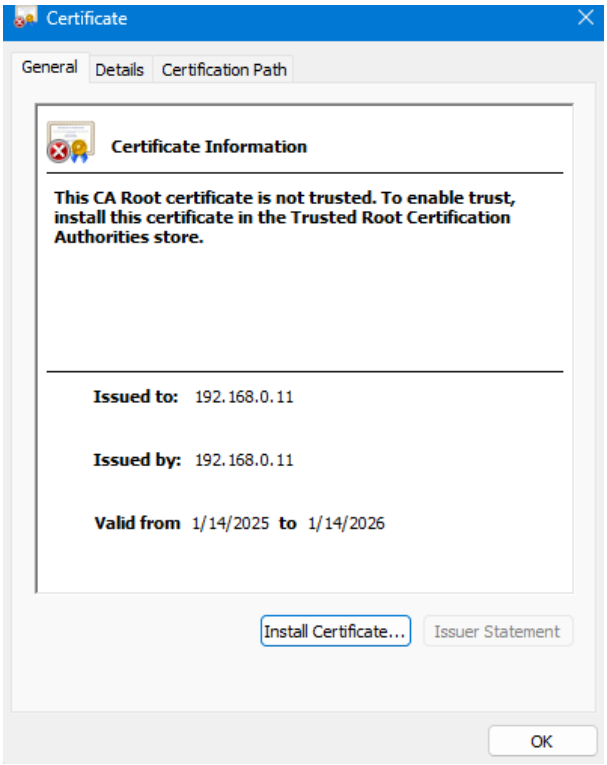
```
upstream flask_backends {
    server 127.0.0.1:5001;
    server 127.0.0.1:5002;
    server 127.0.0.1:5003;
}

server {
    listen 443 ssl;
    server_name 192.168.0.11;

    ssl_certificate /etc/nginx/ssl/server.crt;
    ssl_certificate_key /etc/nginx/ssl/server.key;

    location / {
        proxy_pass http://flask_backends;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }
}

server {
    listen 80;
    return 301 https://$host$request_uri;
}
```

Потом сгенерировать openssl сертификат:

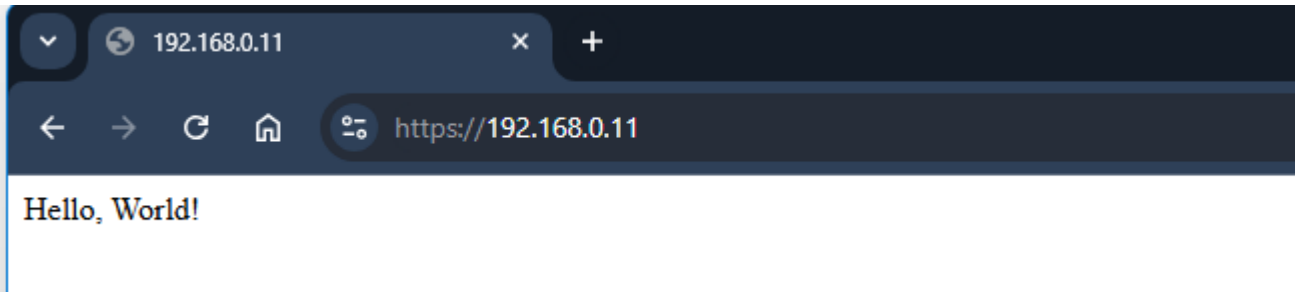
```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
    -keyout /etc/nginx/ssl/server.key \
    -out /etc/nginx/ssl/server.crt \
    -subj "/C=RS/ST=Belgrade/L=Belgrade/O=ITMO/OU=test/CN=192.168.0.11" \
    -addext "subjectAltName=IP:192.168.0.11"
```

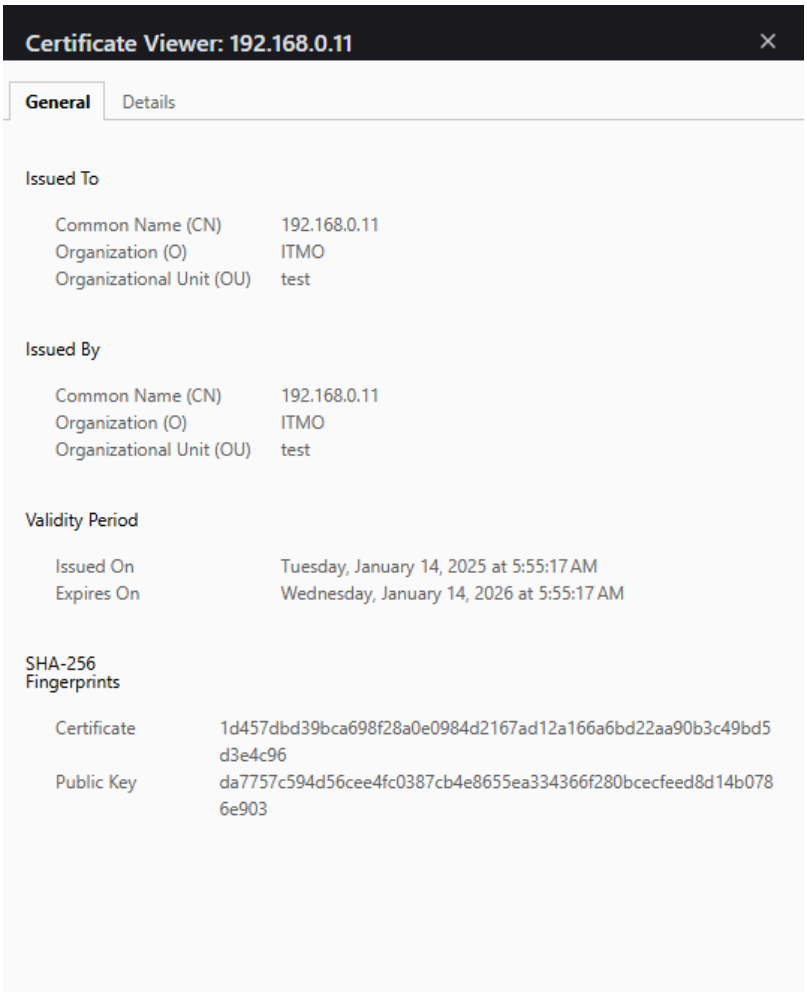
Одно важное примечение - это добавка subjectAltName. Браузеры это проверяют дополнительно, с сравнению с проверкой через терминал, где достачно указать только CN.

На конец нужно установить сгенерированый сертификат на хостовой машине. (в моем случае Windows)



Сейчас можем проверить что можно подключатся с хостовой машине и если сертификат правилный.


192.168.0.11
https://192.168.0.11
Hello, World!


Certificate Viewer: 192.168.0.11
General
Details
Issued To
Common Name (CN) 192.168.0.11
Organization (O) ITMO
Organizational Unit (OU) test
Issued By
Common Name (CN) 192.168.0.11
Organization (O) ITMO
Organizational Unit (OU) test
Validity Period
Issued On Tuesday, January 14, 2025 at 5:55:17 AM
Expires On Wednesday, January 14, 2026 at 5:55:17 AM
SHA-256 Fingerprints
Certificate 1d457dbd39bca698f28a0e0984d2167ad12a166a6bd22aa90b3c49bd5d3e4c96
Public Key da7757c594d56cee4fc0387cb4e8655ea334366f280bcecfeed8d14b0786e903

При зарпросе на порт 80 идет редирект:

```
C:\Users\djurd>curl -v http://192.168.0.11
*   Trying 192.168.0.11:80...
* Connected to 192.168.0.11 (192.168.0.11) port 80
> GET / HTTP/1.1
> Host: 192.168.0.11
> User-Agent: curl/8.9.1
> Accept: */*
>
< HTTP/1.1 301 Moved Permanently
< Server: nginx/1.24.0 (Ubuntu)
< Date: Tue, 14 Jan 2025 05:15:06 GMT
< Content-Type: text/html
< Content-Length: 178
< Connection: keep-alive
< Location: https://192.168.0.11/
<
```