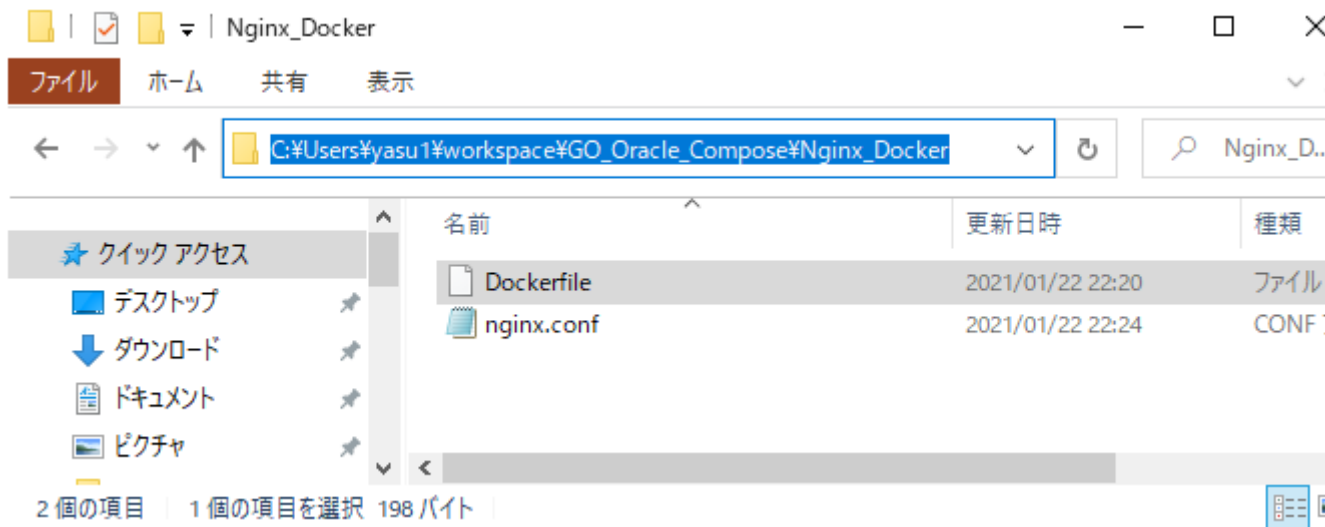


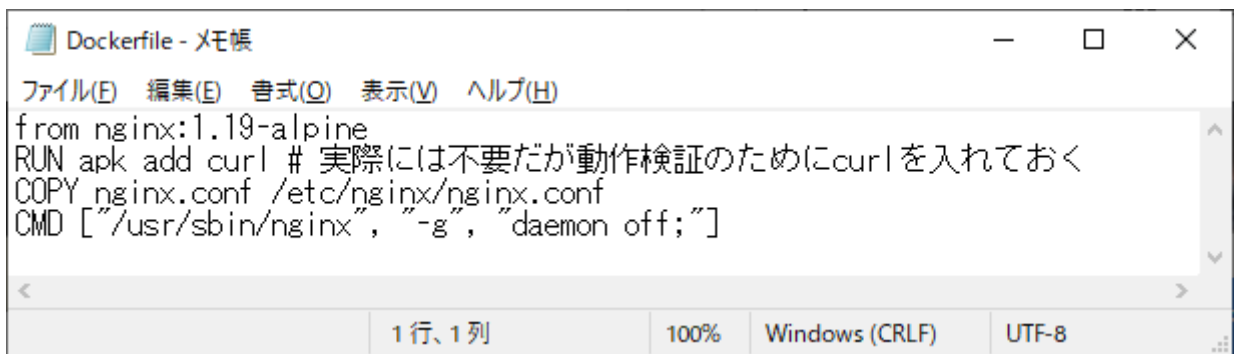
Nginxコンテナ構築

1. Dockerイメージ作成

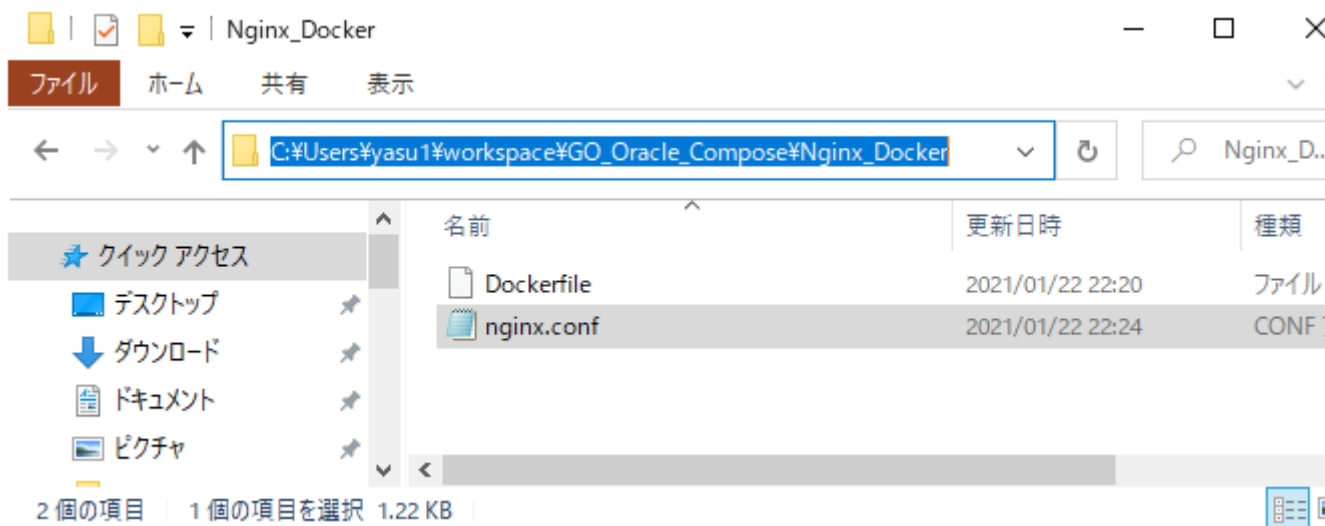
- a. 「C:\Users\yasu1\workspace\GO_Oracle_Compose\Nginx_Docker」フォルダに「Dockerfile」を作成



- b. 「Dockerfile」の内容は以下の通り



- c. 「C:\Users\yasu1\workspace\GO_Oracle_Compose\Nginx_Docker」フォルダに「nginx.conf」を作成



- d. 「nginx.conf」の内容は以下の通り

```
nginx.conf - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
user nginx;
worker_processes 1;

pid /var/run/nginx.pid;

events {
    worker_connections 1024;
}

http {
    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    proxy_cache_path /var/cache/nginx keys_zone=cache-zone:1m levels=2:2 max_size=1g inactive=365d;
    proxy_temp_path /var/cache/nginx/cache;

    server {
        server_name localhost;
        listen 80;

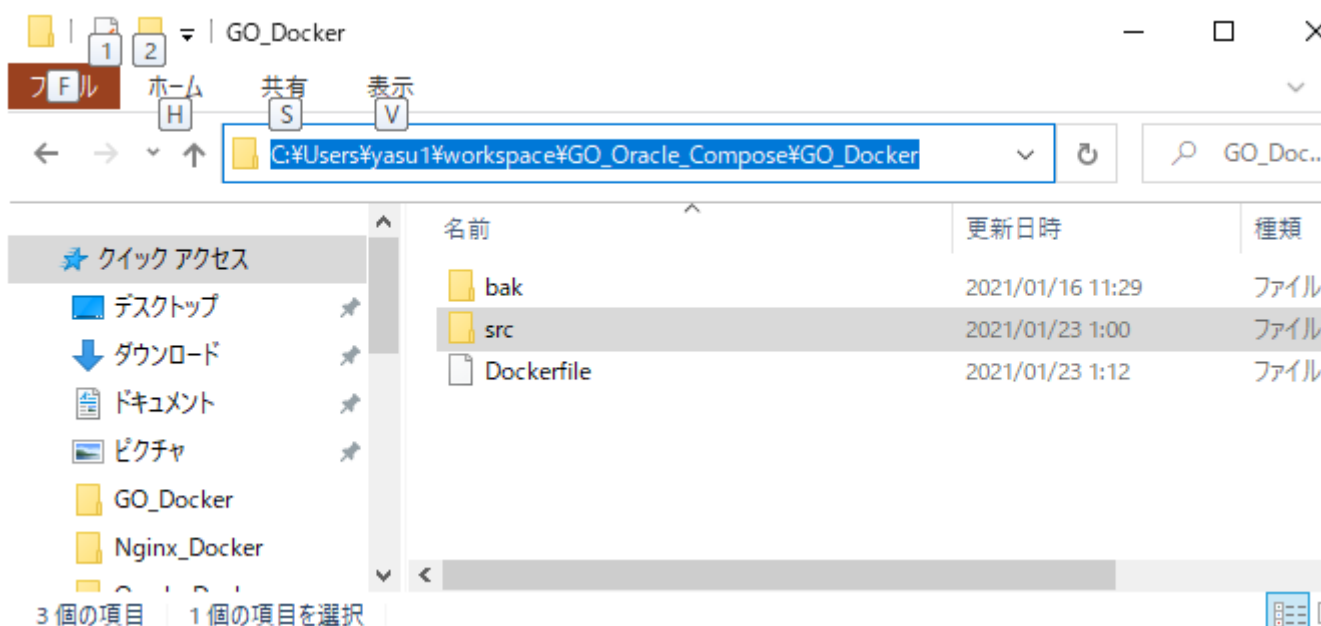
        location ^~ / {
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_set_header X-Forwarded-Proto $scheme;
            proxy_set_header X-Real-IP $remote_addr;

            # キャッシュを行うURLを指定
            proxy_pass http://goContainerSys01/;
            proxy_cache cache-zone;
            proxy_ignore_headers X-Accel-Redirect X-Accel-Expires Cache-Control Expires Vary Set-Cookie;
            proxy_cache_key $host$uri$is_args$args;

            # キャッシュを行うステータスコードとその際の有効期限を指定
            proxy_cache_valid 200 201 300 301 302 365d;

            # キャッシュが有効かどうかをレスポンスヘッダーに付与する
            add_header X-Cache-Status $upstream_cache_status;
        }
    }
}
```

- e. 「C:\Users\yasu1\workspace\GO_Oracle_Compose\GO_Docker」フォルダに「Dockerfile」を作成



- f. 「Dockerfile」の内容は以下の通り

```
Dockerfile - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
# vim:set ft=dockerfile:
FROM golang:1.15-alpine3.12
RUN apk add --no-cache git
RUN go get github.com/lib/pq
RUN go get github.com/labstack/echo
RUN go get github.com/dgrijalva/jwt-go
RUN go get golang.org/x/time/rate
WORKDIR /src
COPY ./src/main.go /src
COPY ./src/config.json /src
RUN go build -o /bin/start_appserver main.go
CMD ["/bin/start_appserver"]

1 行, 1 列 100% Windows (CRLF) UTF-8
```

g. Dockerイメージを作成

```
01_Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

新しいクロスプラットフォームの PowerShell をお試しください https://aka.ms/pscore6

PS C:\Users\yasu1> cd .\workspace\GO_Oracle_Compose\Nginx_Docker\
PS C:\Users\yasu1\workspace\GO_Oracle_Compose\Nginx_Docker> █
```

```
01_Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

新しいクロスプラットフォームの PowerShell をお試しください https://aka.ms/pscore6

PS C:\Users\yasu1\workspace\GO_Oracle_Compose\Nginx_Docker> docker image build ./ -t nginx
[+] Building 13.2s (4/7)
=> [internal] load build definition from Dockerfile
    0.1s
=> => transferring dockerfile: 237B
    0.0s
=> [internal] load .dockerignore
    0.1s
=> => transferring context: 2B
    0.0s
=> [internal] load metadata for docker.io/library/nginx:1.19-alpine
    4.3s
=> [1/3] FROM docker.io/library/nginx:1.19-alpine@sha256:c2ce58e024275728b00a554ac25628af25c54782865b3487b11c 8.7s
=> => resolve docker.io/library/nginx:1.19-alpine@sha256:c2ce58e024275728b00a554ac25628af25c54782865b3487b11c 0.0s
```

```
01_Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

新しいクロスプラットフォームの PowerShell をお試しください https://aka.ms/pscore6

PS C:\Users\yasu1> cd .\workspace\GO_Oracle_Compose\GO_Docker\
PS C:\Users\yasu1\workspace\GO_Oracle_Compose\GO_Docker> docker image build ./ -t go
[+] Building 5.8s (15/15) FINISHED
=> [internal] load build definition from Dockerfile
    0.0s
=> => transferring dockerfile: 406B
```

```

0.0s
=> => naming to docker.io/library/go
0.0s
PS C:\Users\yasu1\workspace\GO_Oracle_Compose\GO_Docker> docker image ls
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
go                   latest          e79a63ee3f42   54 seconds ago  460MB
nginx                latest          b67ba2152bb4   17 minutes ago  24.2MB
oracle/database     19.3.0-se2     f8c10909c98c   7 days ago     6.53GB
postgres            13             a6cd88e1dfce   6 weeks ago    314MB
postgres            latest          a6cd88e1dfce   6 weeks ago    314MB
PS C:\Users\yasu1\workspace\GO_Oracle_Compose\GO_Docker>

```

h. ネットワークを作成し、Dockerコンテナを起動

```

01_Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

新しいクロスプラットフォームの PowerShell をお試しください https://aka.ms/pscore6

PS C:\Users\yasu1> docker network create bridge01
72e369cdfadceaf0a8dff8e013524452b9d44a62cc70d523abed4fd614972b28
PS C:\Users\yasu1>

```

```

01_Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

新しいクロスプラットフォームの PowerShell をお試しください https://aka.ms/pscore6

PS C:\Users\yasu1> docker run --rm -e POSTGRES_PASSWORD=mysecretpassword -d -v C:\Users\yasu1\workspace\GO_postgresql_Compose\postgresql_Docker\data:/var/lib/postgresql/data --network bridge01 --name postgresqlContainerSys01 -p 5432:5432 postgres 1f2bfd9a86c59e284979e278d24bd2120e788f44f3d10a82568b11fd0e8fd858
PS C:\Users\yasu1> docker run --rm -d --name goContainerSys01 --network bridge01 go 39be6c9d7c9857e0a63189216cac5d750207a9bfe7a5b2f58de2dd4ccceef30c
PS C:\Users\yasu1> docker container run --rm -it -d -p 80:80 --network bridge01 --name nginxContainerSys01 nginx 4ff56a48b338db0f3bf3afff5efb54e4a35723b4c3c3de65fb2b50e018c245bd
PS C:\Users\yasu1> docker container ls -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS
4ff56a48b338   nginx         "/docker-entrypoint.…"   8 seconds ago Up 7 seconds  0.0.0.0:80->80/tcp      nginxContainerSys01
39be6c9d7c98   go           "/bin/start_appserver"   22 seconds ago Up 21 seconds                goContainerSys01
1f2bfd9a86c5   postgres     "docker-entrypoint.s…"   37 seconds ago Up 36 seconds  0.0.0.0:5432->5432/tcp   postgresqlContainerSys01
5d369879b7f7   oracle/database:19.3.0-se2 "/bin/sh -c 'exec $O…"   6 hours ago   Up 6 hours (healthy)  0.0.0.0:1521->1521/tcp, 0.0.0.0:5500->5500/tcp   oracle
PS C:\Users\yasu1>

```

2. postmanで確認

a. localhostの80番ポート(nginx)にGET。169msでレスポンス。

Postman

File Edit View Help

+ New Import Runner

My Workspace Invite

Filter

History Collections APIs

+ New Collection Trash

Test 5 requests

Launch... GET TestG... POST Tes... DEL Test... PUT Test... GET TestAl... + ... No Environment

TestGET

GET http://localhost/products Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (6) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   {
3     "id": "1-Cheese",
4     "href": "/products/1-Cheese",
5     "productNo": 1,
6     "name": "Cheese",
7     "price": 9.99
8   },
9   {
10    "id": "2-Bread",
11    "href": "/products/2-Bread",
12    "productNo": 2,
13    "name": "Bread",
14    "price": 1.99
15  }
16 }
```

Find and Replace Console Bootcamp

- b. もう一度localhostの80番ポート(nginx)にGET。19msでレスポンス。キャッシュを利用して返却。

Postman

File Edit View Help

+ New Import Runner

My Workspace Invite

Filter

History Collections APIs

+ New Collection Trash

Test 5 requests

Launch... GET TestG... POST Tes... DEL Test... PUT Test... GET TestAl... + ... No Environment

TestGET

GET http://localhost/products Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (6) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   {
3     "id": "1-Cheese",
4     "href": "/products/1-Cheese",
5     "productNo": 1,
6     "name": "Cheese",
7     "price": 9.99
8   },
9   {
10    "id": "2-Bread",
11    "href": "/products/2-Bread",
12    "productNo": 2,
13    "name": "Bread",
14    "price": 1.99
15  }
16 }
```

Find and Replace Console Bootcamp