HuggingFace. Spaces 정리

> Spaces (huggingface.co) 정리

- Spaces 는 ML demo apps 를 간단하게 배포할 수 있는 방법을 제공하는 서비스다!
  - ML portfolio, showcase, conferences or to stakeholders 등의 경우에 쓸만하다고 하는데, 그냥 Model 사용 등을 쉽게 할 수 있도록 배포하는 것이 결국 그 역할이라고 생각한다!

> 전반적으로 해당 내용 확인했을 때, Overview를 제외하고는 각 Space Type 에 해당 하는 내용 및 사용 수준에 관련된 내용이 서술되어 있음. 모든 내용을 살펴보기 보다는 Overview 와 Docker 관련 Space 내용만 살펴보면 될 것으로 생각. 해당 내용들을 중점적으로 정리.

Space Overview

- ML demo apps 를 쉽게 만들고 배포.

Creating a new Space

- Spaces main page 에서, Create new space 누리기. 이후 나오는 항목들을 선택함.
- 내부적으로 Space 는 huggingspace 가 관리하는 git repository 로서 관리됨.
  - 이 덕에 git 으로 사용할 수 있는 많은 도구들을 Space 에서도 사용할 수 있음.
  - 어떤 기능을 수행할 수 있는지 궁금한 경우 해당 내용 살필 것 : Getting Started with Repositories (Commit, Push 등의 내용에 대해 전달하고 있음.)
- 각각에 대한 Space 제작이 종류에 따라 다르게 동작하므로, 해당 내용 확인할 것.
  - Creating a Gradio Space
  - Creating a Streamlit Space
  - Creating a Docker Space

Hardware resources

- Each Spaces environment is limited to 16GB RAM, 2 CPU cores and 50GB of (not persistent) disk space by default
- 늘리고 싶으면 돈 내라! (Pricing page)

Managing secrets and environment variables

- If your app requires environment variables (for instance, secret keys or tokens), do not hard-code them inside your app!
- Settings page of your Space repository 로 이동해, 거기에 Variables 와 Secret Key 를 입력해라!
- Docker Space 를 사용하는 경우 : environment management with Docker
- 비밀은 비공개이며 설정된 후에는 해당 값을 검색할 수 없음! (Docker 와 Streamlit 의 경우 좀 다르게 관리되므로, 각 문서 항목을 살필 필요가 있음.)

Duplicating a Space

- if you want to build a new demo using another demo as an initial template, 이때 Duplicate 가 유용할 수 있음.
- 스페이스 오른쪽 상단에 있는 세 개의 점을 클릭 후 복제 가능하므로 참고.

Networking

- You can make requests through the standard HTTP and HTTPS ports (80 and 443) along with port 8080. (다른 port number 의 경우 block 되므로 유의!)

Lifecycle management

- 무료면 켜두고 있는 중에만 관리 가능.
- 유료면 안 켜둬도 실행 가능함.
  - 1시간이나 이런 Sleep 시간이 지나면 자동 종료도 됨.
- 단, 1시간마다 돈이 들기 때문에 이런 건 좀 유의해 둘 필요 있음.

Helper environment variables

- Space author, repository name에 대한 metadata 가 자동으로 설정되는 걸 원할 수 있음.
- 이 또한 자동으로 변경되거나 하도록 지원!
- 어떤 환경 변수 항목이 존재하는지는 Spaces Overview (huggingface.co) 여기서 찾을 수 있도록 하기.

- OAuth 와 관련된 환경변수도 존재. (OAuth 를 까먹었으면, OpenAI. API Key 숨기기, ! API Key 숨기기, ASP.NET 제작 과정 중 익힌 점 요약 확인.)
- HuggingFace 의 Spaces 가 Resources Server (자원 요청을 받아, 실제 자원을 반환해주는 Server) 임을 가정하고, "OPENID_PROVIDER_URL" 와 같은 환경 변수를 제공받아 인증에서 사용하는 것으로 보임. (*Q. 내 이해가 맞을까? => 한번 검증해보면 좋을 듯!*)

Clone the Repository

- 쉽게 가능.

Linking Models and Datasets on the Hub

- Space's README metadata 에, model 및 dataset 을 링크할 수 있는 항목이 존재함.

> 이후 항목 약간 건너 뜀!

Docker Spaces

- Spaces accommodate (수용) custom Docker containers for apps outside the scope of Streamlit and Gradio.
    - Docker 를 수용함으로써 Spaces 는 이전 표준 SDK 로 가능했던 한계를 뛰어넘음.
        - ex - FastAPI 사용 및 구축, Go Endpoint 구축 등.

Setting up Docker Spaces

- Hugging Face – The AI community building the future. 에서 Docker 를 선택할 시, `README` 의 YAML block 에 들어있는 `sdk` property를 `docker` 로 설정함으로써 초기화를 수행한다.
    - 혹은 대신, YAML Block 이 Docker 로 설정되어있는 repository 를 복사하여 사용할 수도 있다.
    - Docker 의 기본 port number 는 7860 이다.
- Port 여러 개 설정 가능.
- If you want to expose apps served on multiple ports to the outside world, Nginx 같이 더 넓은 인터넷으로 requests 를 발송하는 역 방향 proxy server 를 활용하면 좋다.

Secrets and Variables Management

Variables

- Buildtime

- 변수는 Docker Space를 빌드할 때, `build-arg`s 로서 전달된다.
    - New 에 입력해둔 친구들이 실제 Docker 내부에 어떻게 들어가는지에 대한 내용임.
- Docker's dedicated documentation : Dockerfile 을 어떻게 사용하는지에 대한 가이드.

- Runtime

- 변수는 런타임에 컨테이너의 환경에 삽입된다.
    - (따로 사이트에서는 관리가 안 되는 듯 함. 따로 Repository 등을 직접 받아서 변경해줘야 하는 듯.)

Secret

- 보안상의 이유로 좀 다름. 비밀을 만든 후에는 Dockerfile에 다음 줄을 추가하여 비밀을 노출할 수 있음.

- Docker 를 빌드한 후, file 에 mount 한 다음.

```shell
`$(cat /run/secrets/SECRET_EXAMPLE)`.
```

Permissions

- User Id 랑 chmod (linux 명령) 기반으로 해당 permission 조작 가능한 듯.

Data Persistence

- Docker Space restarts 하면, data written on disk가 없어짐.
  - 요금제 정책, 혹은 추가 구입한 persistent storage 에 따라서, 바뀔 수 있음.
- At the moment, `/data` volume is only available at runtime, i.e. you cannot use `/data` during the build step of your Dockerfile.

Docker Spaces Tutorial

- Your First Docker Spaces (huggingface.co)
- Docker Spaces Example (huggingface.co)
- 해당 내용 확인해봤을 때, Dockerfile 에 main app 에 대한 호출이 있는 것을 보면, 해당 repository 는 자동으로 dockerfile 을 기반으로 build 를 수행하는 것으로 보임.
- 생각해보면 그렇네. repository 및 이를 돌릴 수 있는 computer 의 형태로서 spaces 가 지원된다 하면. CMD 를 기반으로 내가 명령어를 칠 수 없다면, 아무리 Docker 가 기본으로 그 원격 컴퓨터가 깔려있다 하더라도 Image 를 받아오거나 Container 를 만들어 줄 수 없잖아?
    - 이때, DockerFile 등을 기반으로 Image 를 만들어서, Container 의 배포까지를 자동으로 수행한다 하면, 합리적이지.
    - 근데 이러면, run 할 때, model 이름을 포함할 수가 없음...
    - *Q. 다른 원격 서비스(AWS, Azure)도 이렇게 시도하여야 하는가?*