

Github Codespace 개발환경 유지 가이드

(GitHub Codespace – DevContainer를 활용한 Codespace 환경 설정 자동화)

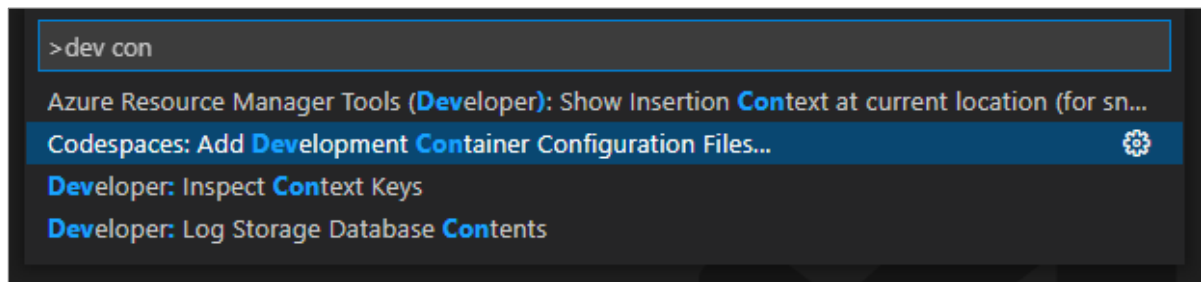
목차

GitHub Codespace – DevContainer를 활용한 Codespace 환경 설정 자동화	1
DevContainer 구성 파일 추가	2
.devcontainer 내부 파일 살펴보기 & DevContainer 파일로 환경 다시 빌드	2
DevContainer 커스터마이징 하기	5
컨테이너에 추가 패키지 설치	6
Codespaces 초기 환경 설정 수정하기 (.devcontainer.json)	6
수정된 환경 빌드 및 확인	8
DevContainer 설정 파일 공유	8

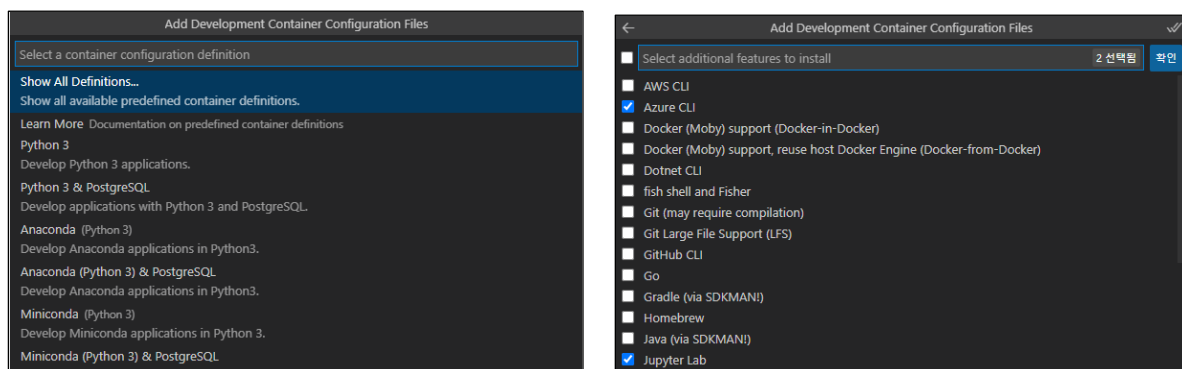
가끔 개발을 하다가 OS업데이트, 추가적인 패키지 설정 및 구성, 개발 환경 수정을 하다보면 환경이 꼬여서 기존 환경을 제거하고 처음부터 다시 구축하는 경우가 자주 있습니다. 팀에 새로운 구성원이 들어와서 개발환경 구축을 먼저 하는 경우도 자주 있는데, 이 때 문서화 해둔 개발 환경 구축 가이드로는 설명이 부족한 경우가 있기도 합니다.

GitHub Codespace 는 VS Code 기반이여서, VS Code 의 DevContainer 기능도 그대로 활용이 가능합니다. 본 문서에서는 Dev Container 기능을 활용하여 각자의 프로젝트에 맞는 개발환경을 DevContainer 설정 파일과 Dockerfile 로 미리 정의하고 이를 이용하여, 개발환경이 미리 구축된 Codespace 환경을 자동으로 초기화 하는 방법에 관해 알아봅니다.

DevContainer 구성 파일 추가



Command Palette (Ctrl+Shift+P) 에서 "dev container" 를 검색하면 "Codespaces: Add Development Container Configuration Files..." 가 나오는데 이를 선택합니다.



사전에 정의된 DevContainer 구성 파일을 선택할 수 있습니다. 각 DevContainer 가 내부적으로 어떻게 정의 되어 있는지는 아래 링크에서 확인하실 수 있습니다.

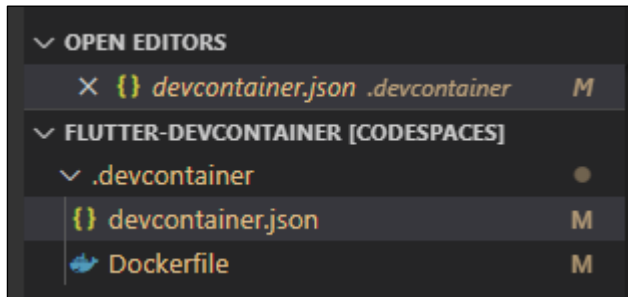
<https://github.com/Microsoft/vscode-dev-containers>

사전 정의된 DevContainer 중에는 MongoDB나 PostgreSQL 등 데이터베이스가 함께 포함된 구성도 있습니다. 필요에 따라 선택해서 사용하면 됩니다.

여기서는 Python 런타임 및 개발 도구만 있으면 된다고 가정하고, Python 3 로 선택해 초기화 하겠습니다.

.devcontainer 내부 파일 살펴보기 & DevContainer 파일로 환경 다시 빌드

이후 Python 버전 등을 선택하여 나머지 단계를 마치면 .devcontainer 디렉터리와 그 내부에 devcontainer.json, Dockerfile 이 생성되는 것을 확인할 수 있습니다.



생성된 devcontainer.json 을 열어보면 기본적으로 아래와 같은 내용으로 구성되어 있습니다. Devcontainer.json 은 Codespace 환경 초기화 시 어떤 설정값을 적용하여 초기화 할지 정의하는 파일입니다. 사용할 컨테이너 정의 파일 (Dockerfile), Codespace 설정값, 사전 설치 확장 기능 등 전반적인 환경을 구성할 수 있습니다.

```
You, 10 minutes ago | 1 author (You)
1 // For format details, see https://aka.ms/devcontainer.json. For config options, see the README at:
2 // https://github.com/microsoft/vscode-dev-containers/tree/v0.192.0/containers/python-3
3 {
4   "name": "Python 3",
5   "build": {
6     "dockerfile": "Dockerfile",
7     "context": "..",
8     "args": {
9       // Update 'VARIANT' to pick a Python version: 3, 3.6, 3.7, 3.8, 3.9
10      "VARIANT": "3",
11      // Options
12      "NODE_VERSION": "lts/*"
13    }
14  },
15
16  // Set *default* container specific settings.json values on container create.
17  "settings": { ...
31  },
32
33  // Add the IDs of extensions you want installed when the container is created.
34  "extensions": [ ...
37  ],
38
39  // Use 'forwardPorts' to make a list of ports inside the container available locally.
40  // "forwardPorts": [],
41
42  // Use 'postCreateCommand' to run commands after the container is created.
43  // "postCreateCommand": "pip3 install --user -r requirements.txt",
44
45  // Comment out connect as root instead. More info: https://aka.ms/vscode-remote/containers/non-root.
46  "remoteUser": "vscode"
47 }
48
```

- name: DevContainer 환경 이름
- build: 컨테이너 환경 설정, (환경 빌드에 사용할 Dockerfile, 빌드시 입력할 인자 등 지정)
- settings: Codespace 에 사전 적용할 설정 값(확장기능 활성화 여부, 테마 등)

```
// Set *default* container specific settings.json values on container create.
"settings": {
  "python.pythonPath": "/usr/local/bin/python",
  "python.languageServer": "Pylance",
  "python.linting.enabled": true,
  "python.linting.pylintEnabled": true,
  "python.formatting.autopep8Path": "/usr/local/py-utils/bin/autopep8",
  "python.formatting.blackPath": "/usr/local/py-utils/bin/black",
  "python.formatting.yapfPath": "/usr/local/py-utils/bin/yapf",
  "python.linting.banditPath": "/usr/local/py-utils/bin/bandit",
  "python.linting.flake8Path": "/usr/local/py-utils/bin/flake8",
  "python.linting.mypyPath": "/usr/local/py-utils/bin/mypy",
  "python.linting.pycodestylePath": "/usr/local/py-utils/bin/pycodestyle",
  "python.linting.pydocstylePath": "/usr/local/py-utils/bin/pydocstyle",
  "python.linting.pylintPath": "/usr/local/py-utils/bin/pylint"
},
```

- extensions: Codespace 에 사전 설치할 확장기능. 확장ID 로 지정

```
// Add the IDs of extensions you want installed when the container is created.
"extensions": [
  "ms-python.python",
  "ms-python.vscode-pylance"
],
```

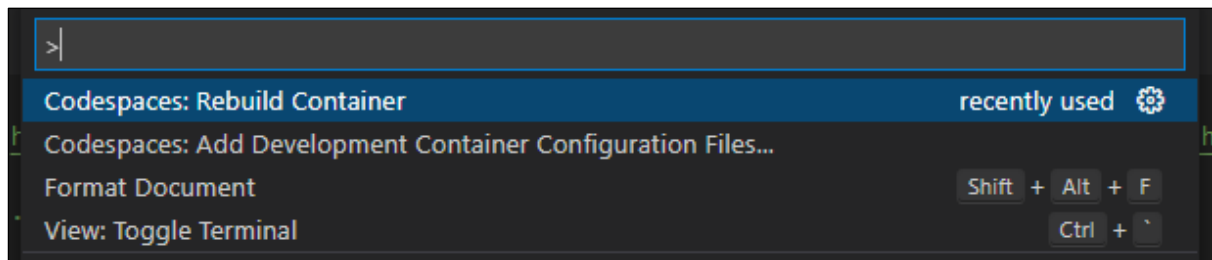
- remoteUser: 터미널에서 사용하는 root 가 아닌 기본 사용자 명을 지정합니다.

.devcontainer 내부의 Dockerfile 은 Codespace 초기화 시 컨테이너 환경 빌드할 때 필요한 프로그램 설치나 OS 환경 변수 설정 작업을 정의할 때 사용합니다. 사전 생성된 Dockerfile 에는 GitHub 에서 사전에 빌드한 Codespace 용 컨테이너 이미지를 기반으로 작성되어 있음을 확인 할 수 있습니다.

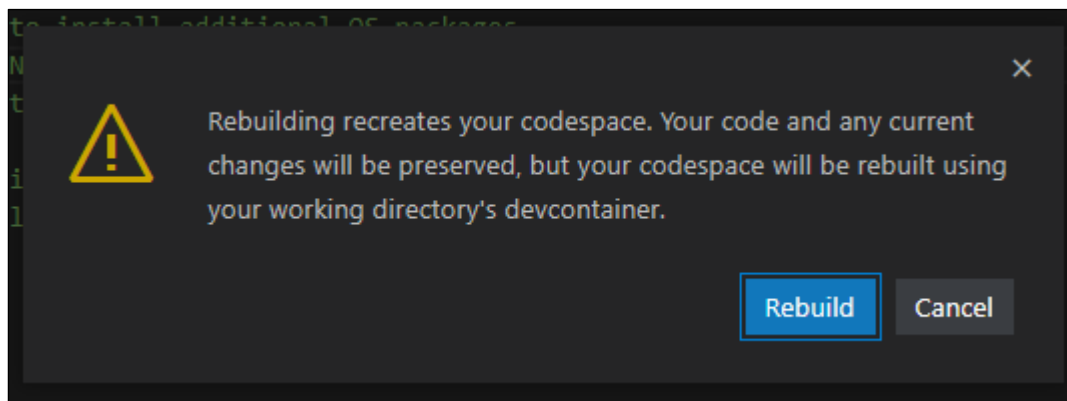
```
1 # See here for image contents: https://github.com/microsoft/vscode-dev-containers/tree/v0.192.0/containers/python-3/.devcontainer/base.Dockerfile
2
3 # [Choice] Python version: 3, 3.9, 3.8, 3.7, 3.6
4 ARG VARIANT="3.9"
5 FROM mcr.microsoft.com/vscode/devcontainers/python:0-${VARIANT}
6
7 # [Choice] Node.js version: none, lts/*, 16, 14, 12, 10
8 ARG NODE_VERSION="none"
9 RUN if [ "${NODE_VERSION}" != "none" ]; then su vscode -c "umask 0002 && . /usr/local/share/nvm/nvm.sh && nvm install ${NODE_VERSION} 2>&1"; fi
10
11 # [Optional] If your pip requirements rarely change, uncomment this section to add them to the image.
12 # COPY requirements.txt /tmp/pip-tmp/
13 # RUN pip3 --disable-pip-version-check --no-cache-dir install -r /tmp/pip-tmp/requirements.txt \
14 #     && rm -rf /tmp/pip-tmp
15
16 # [Optional] Uncomment this section to install additional OS packages.
17 # RUN apt-get update && export DEBIAN_FRONTEND=noninteractive \
18 #     && apt-get -y install --no-install-recommends <your-package-list-here>
19
20 # [Optional] Uncomment this line to install global node packages.
21 # RUN su vscode -c "source /usr/local/share/nvm/nvm.sh && npm install -g <your-package-here>" 2>&1
```

이제 새로 생성한 DevContainer 파일로 환경을 다시 초기화 해보겠습니다. Command Palette 에

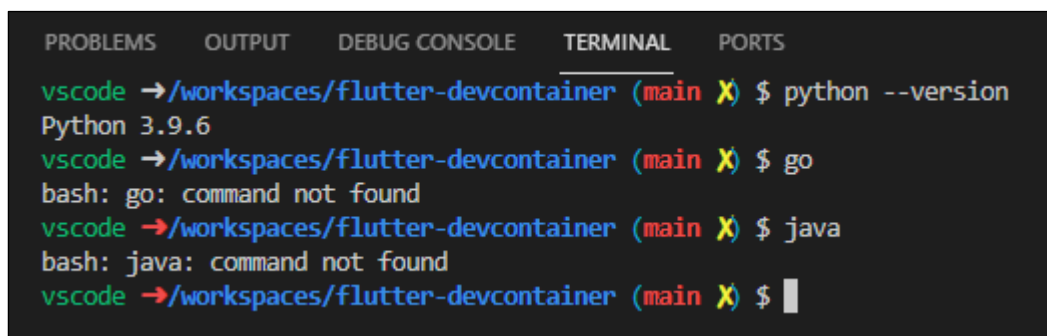
서 "Codespaces: Rebuild Container" 를 선택합니다.



Rebuild 를 선택하여 다시 빌드합니다.



새로 다시 초기화되어 열린 Codespace 에는 기존 파일은 그대로 있지만, 터미널에서 명령을 실행 해 보면 python 이외 다른 개발 도구는 이제 사용이 불가능 한 것을 확인할 수 있습니다.



DevContainer 커스터마이징 하기

생성된 DevContainer 를 수정하여 원하는 형태의 개발 환경이 Codespace 초기화 시 자동으로 구축되도록 할 수 있습니다. 여기서는 앞서 생성한 Python 환경에 Django CLI와 Django 확장 기능을 설치하여 초기화 하도록 해 보겠습니다.

컨테이너에 추가 패키지 설치

.devcontainer/Dockerfile 을 수정하여 추가 패키지를 설치하도록 지정합니다. 먼저 해당 Dockerfile 이 사용하는 Base 이미지를 먼저 확인하여 어떤 OS 환경에서 추가 패키지 설치 작업을 하는지 확인합니다.

본 문서에서 예시로 사용한 Python 환경은 아래 이미지를 Base 로 사용합니다.

```
ARG VARIANT="3.9"
FROM mcr.microsoft.com/vscode/devcontainers/python:0-${VARIANT}
```

아래 링크에서 해당 이미지에 대한 정보를 확인하면, 이미지가 Debian 을 컨테이너 OS 로 사용하고 있음을 확인할 수 있습니다.

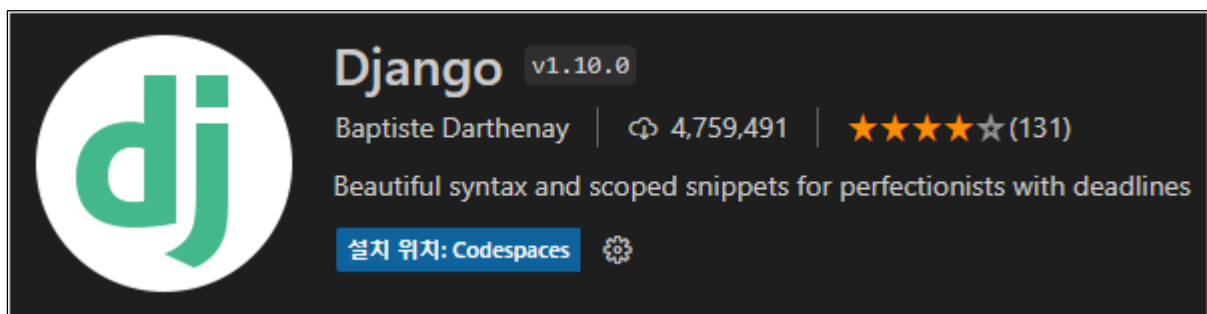
<https://github.com/microsoft/vscode-dev-containers/tree/v0.192.0/containers/python-3>

RUN 지시자로 pip 명령을 실행하여 Django 모듈을 설치하는 명령을 Dockerfile 에 추가합니다.

```
RUN pip install django
```

Codespaces 초기 환경 설정 수정하기 (.devcontainer.json)

여기서는 아래 Django 확장 기능(확장 ID: batisteo.vscode-django) 를 설치하고 Codespaces 환경과 연동되어 작동하도록 설정해 보겠습니다.



먼저 위 확장기능이 사전 설치 되도록, .devcontainer.json 의 extensions 에 확장ID 를 추가합니다.

(기존)

```
"extensions": [  
    "ms-python.python",  
    "ms-python.vscode-pylance"  
],
```

(수정)

```
"extensions": [  
    "ms-python.python",  
    "ms-python.vscode-pylance",  
    "batisteo.vscode-django"  
],
```

해당 확장 기능이 요구하는 파일 확장자 연동 설정도 자동으로 초기화 되도록 해 줍시다. 이를 위해 "settings" 속성을 활용합니다.

```
"settings": {  
    ... (생략)  
},
```

여기에 아래와 같은 설정을 추가합니다.

```
"files.associations": {  
    "**/*.html": "html",  
    "**/templates/**/*.html": "django-html",  
    "**/templates/**/*": "django-txt",  
    "**/requirements{/**/*.*,*}.txt,in)": "pip-requirements"  
},
```

수정된 코드는 아래와 같습니다.

```
"settings": {  
    ... (생략),  
    "files.associations": {  
        "**/*.html": "html",  
        "**/templates/**/*.html": "django-html",  
        "**/templates/**/*": "django-txt",  
        "**/requirements{/**/*.*,*}.txt,in)": "pip-requirements"  
    },  
},
```

수정된 환경 빌드 및 확인

앞에서 언급한 방법으로 DevContainer 를 다시 빌드하여 확인해 봅시다.

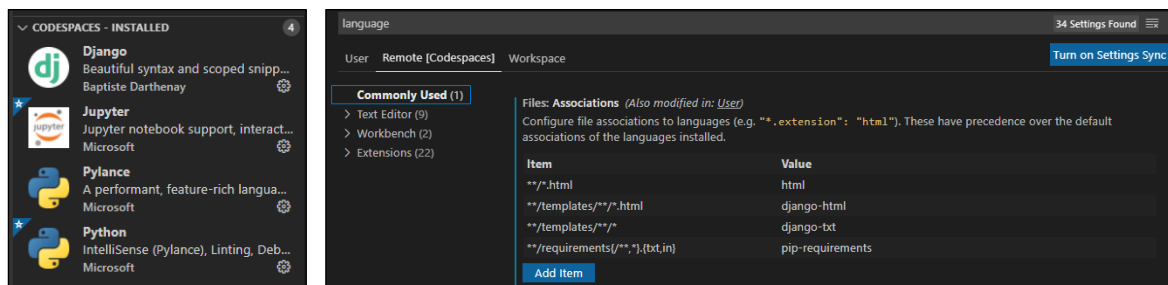
Django-admin 명령 실행시 아래와 같이 정상적으로 실행 됨을 확인이 가능합니다.

```
vscode ->/workspaces/flutter-devcontainer (main X) $ django-admin
Type 'django-admin help <subcommand>' for help on a specific subcommand.

Available subcommands:
[django]
  check
  compilemessages
  createcachetable
  dbshell
  diffsettings
  dumpdata
  flush
  inspectdb
  loaddata
  makemessages
  makemigrations
  migrate
  runserver
  sendtestemail
  shell
  showmigrations
  sqlflush
  sqlmigrate
  sqlsequencereset
  squashmigrations
  startapp
  startproject
  test
  testserver

Note that only Django core commands are listed as settings are not properly configured.
SETTINGS_MODULE or call settings.configure() before accessing settings.).
vscode ->/workspaces/flutter-devcontainer (main X) $
```

확장 기능과, Codespace 설정 또한 아래 사진처럼 미리 적용 되어 있는 것을 확인할 수 있습니다.



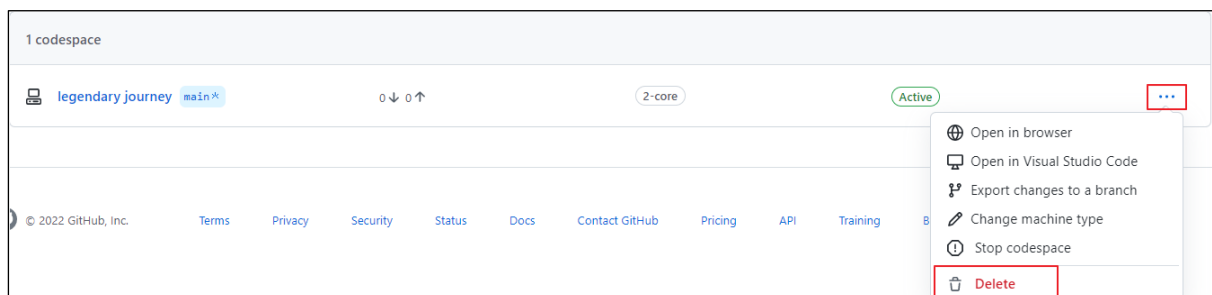
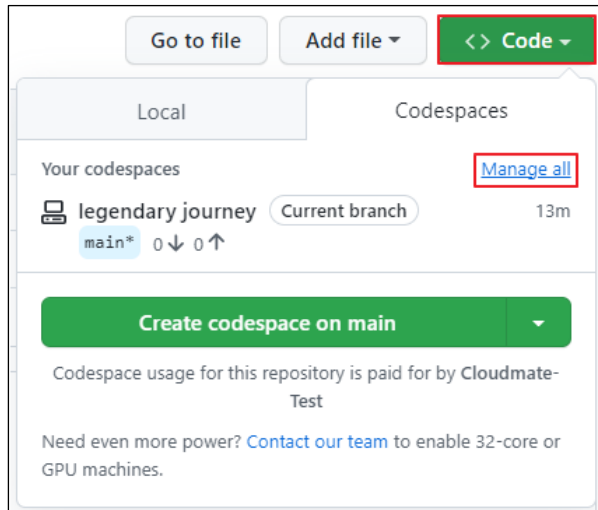
DevContainer 설정 파일 공유

앞에서 로컬 저장소 루트 디렉터리에 생성한 .devcontainer 디렉터리 커밋하여 DevContainer 설정 공유는 물론, 새 팀원이 저장소에서 Codespace 를 초기화 할 때 커밋한 DevContainer 설정 파일로 초기화 되도록 할 수 있습니다.

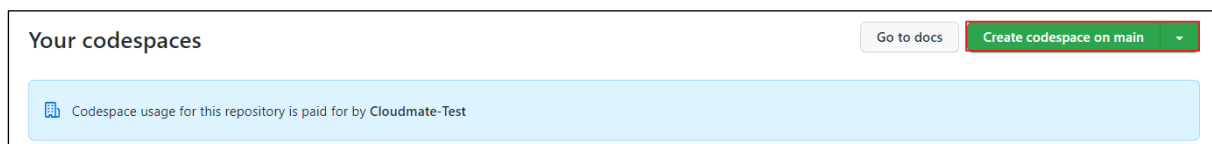
아래 명령으로 DevContainer 파일을 커밋하고 푸시합니다.

```
git add .devcontainer/  
git commit -m "Add DevContainer Config"  
git push
```

https://github.com/codespaces 에서 기존 Codespace 를 삭제합니다.



DevContainer 설정 파일 커밋을 올린 저장소 페이지에서 우측 상단의 "Code" 버튼을 누르고 Codespace 를 새로 생성합니다.



아래 사진처럼 환경 초기화 중 DevContainer 에 정의한 대로 Django 모듈이 설치 되는 것을 확인할 수 있으며, 초기화 후 DevContainer 에 정의한 대로 환경이 준비 되는 것을 보실 수 있습니다.

Setting up your codespace

```
✓ Image found.
🔧 Building container...
  Hide logs

Step 4/5 : RUN if [ "${NODE_VERSION}" != "none" ]; then su vscode -c "umask 0002 &&
. /usr/local/share/nvm/nvm.sh && nvm install ${NODE_VERSION} 2>&1"; fi
---> Running in 5d7c7af336e4
Downloading and installing node v14.17.6...
Downloading https://nodejs.org/dist/v14.17.6/node-v14.17.6-linux-x64.tar.xz...
##### 100.0%
Computing checksum with sha256sum
Checksums matched!
Now using node v14.17.6 (npm v6.14.15)
Creating default alias: default -> lts/* (-> v14.17.6 *)
Removing intermediate container 5d7c7af336e4
---> a9910214ae0f
Step 5/5 : RUN pip install django ←
---> Running in 4cd36218744f
```

추가로 참고 할만한 문서

- devcontainer.json 구성 (GitHub 문서)
 - <https://docs.github.com/en/codespaces/customizing-your-codespace/configuring-codespaces-for-your-project>
- Create a development container (VS Code 문서)
 - <https://code.visualstudio.com/docs/remote/create-dev-container>
- Advanced Container Configuration (VS Code 문서)
 - <https://code.visualstudio.com/docs/remote/containers-advanced>
- devcontainer.json reference (VS Code 문서)
 - <https://code.visualstudio.com/docs/remote/devcontainerjson-reference>