Chapter 06 - Deploying Your API to the Cloud

Benefits and Responsibilities of Cloud Deployment

- The cloud is an informal term to refer to the collection of computer servers and connecting infrastructure that make the public internet.
- Benefits:
  - Learn the end-to-end process of cloud development
  - You can share it.
  - You can use internet-facing tools and products to explore from the user's perspective
  - You can use generative AI services to consumer the API.
- Responsibilities:
  - They can cost you a fortune if you don't pay attention, how to control the costs?
    - Review the costs of services before using them, use cost calculation if available.
    - Use services with free tires and free trial periods.
    - Create monthly budgets and set up email notifications to notify you when you are approaching budgeted amounts.
    - Shut down or delete resources after use.
    - When you have finished working with a cloud host, clean up resources and remove your payment method.
    - Keep tight control of your login credentials.
    - Use short-lived access keys
    - Only activate the permissions for specific services you are using, and disable them after you are done.
    - Do not commit any credentials into source control repositories.
    - If any credentials do become exposed, deactivate or delete them.
  - Security, expose your API endpoints comes with responsibility of keeping API keys, user, password safe from bad actors

Choosing a Cloud Host for Your Project

- It depends on your use case.
  - Continuous integrations (CI)
  - Containerization

- Render
  - Start command: `uvicorn main:app --host 0.0.0.0 --port $PORT`
- AWS

Shipping Your Application in a Docker Container

- Docker is a very useful tool for shipping applications in containers.
  - Dockerfile: is a text document to build a Docker image.
  - Container/Docker Image: an ordered collection of root filesystem changes and the corresponding execution parameters for use within a container runtime.
  - Repository: is a set of Docker images.
  - Container runtime: is software that uses the image to create a container, which is a runtime instance of a container image.

Some Docker Commands and Their Purpose

- `docker --version`: Verify what version of the library is installed.
- `docker build -t`: Build an image from a Dockerfile.
- `docker images`: List local images in your environment.
- `docker run`: Run a container from a local image.

Dockerfile

- Dockerfile contains the instructions that Docker will use to create a container image.

Creating a `.dockerignore` File

- like `.gitignore` when you want to exclude some files in your directory from the Docker image

Building a Container Image

```
docker build -t apicontainerimage .
docker images
docker run --publish 80:80 --name apicontainerl
docker run --publish 80:80 --name apicontainer1 apicontainerimage
```

Additional Resources

- Docker Cheat Sheet
- Best practices for containerizing Python applications with Docker | Snyk
- What is a Container? | Docker
- Deploy a FastAPI App – Render Docs