You don’t think about your Streamlit app optimization until you deploy it to AWS

Data Scientists & MLOps Engineers: The Toxic Love Story

Photo by Vows on the Move on Unsplash

Data Scientists & MLOps Engineers: The Toxic Love Story

Photo by Afif Ramdhasuma on Unsplash

Data Scientists & MLOps Engineers: The Toxic Love Story

Photo by Gabby Orcutt on Unsplash

Agenda

1. What is Streamlit and why we need it
2. Disconnection 1: model loading
3. Disconnection 2: autoscaling and authentication
4. Disconnection 3: security
5. Disconnection 4: sensitive credentials storage
6. Happily ever after
7. Q&A

Darya Petrashka

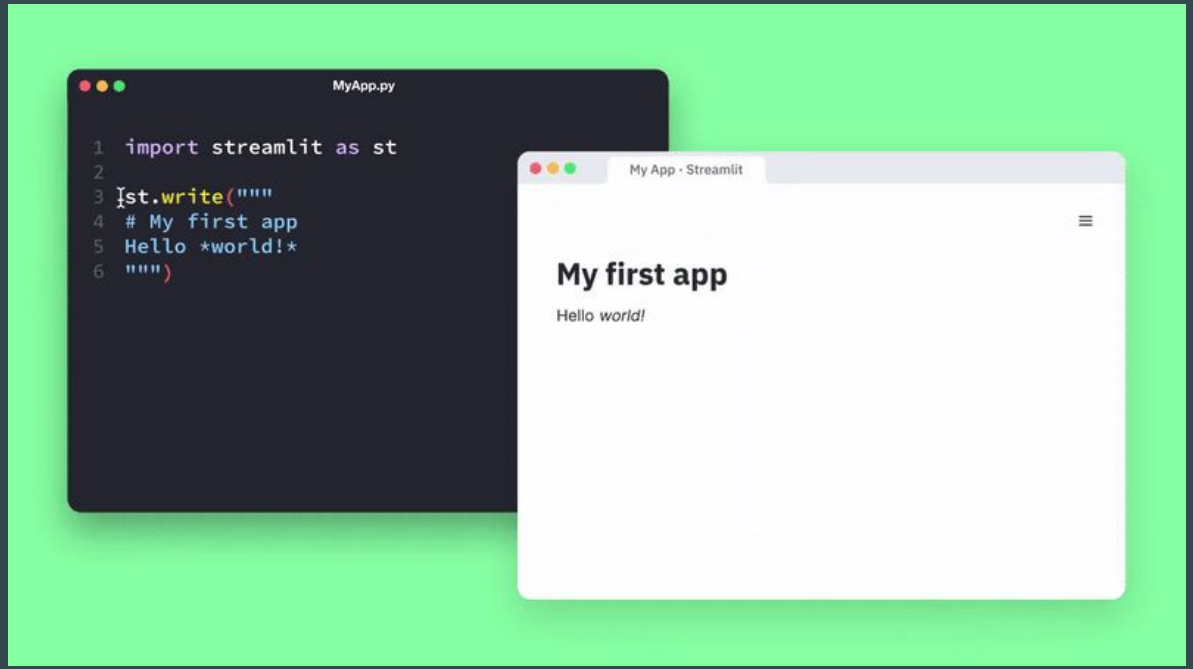
- Senior Data Scientist at SLB
- AWS Community Builder since 2022
- Into: learning, knowledge sharing, ML, NLP, GenAI, hackathons

AWS community builders

What is Streamlit and why we need it

What is Streamlit?

- open-source Python library
- create and share custom web apps for ML and data science



Source: https://docs.streamlit.io/

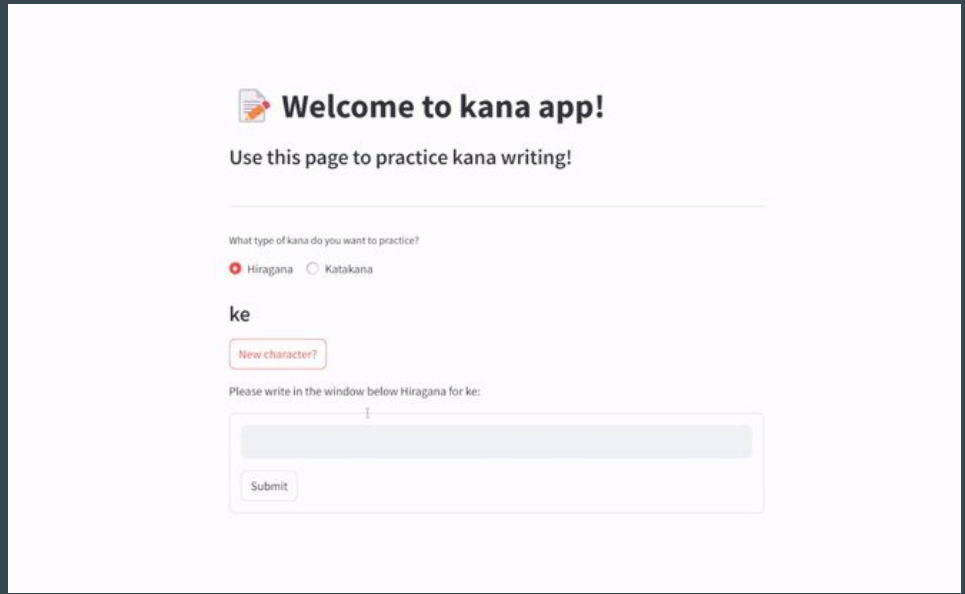
Why we need Streamlit?

- quick way to show your models/data to stakeholders
- simple functionality to learn and use
- easy way to test your models
- life saver for those who cannot write a backend + frontend
- it is hard to make it ugly

Disconnection 1: model loading

Example: Japanese learning app

- Given a pronunciation, write kana character
- Once the character is written, we need to recognize it
- Manga OCR model from the HuggingFace hub: optical character recognition for Japanese text



How might Data Scientist approach the task:

my-awesome-streamlit-app.py

```python
from huggingface_hub import snapshot_download


def download_model():
    model_dir = "/models/manga-ocr"
    if not os.path.exists(model_dir):
        snapshot_download(repo_id="TareHimself/manga-ocr-base",
                          local_dir=model_dir)
    return model_dir


if 'mocr' not in st.session_state:
    st.session_state.mocr = MangaOcr(pretrained_model_name_or_path=download_model())
```

Any issues?

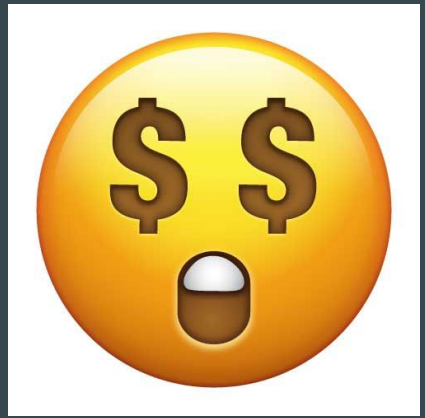
- The model is downloaded
- EVERY
- SINGLE
- TIME

my-awesome-streamlit-app.py

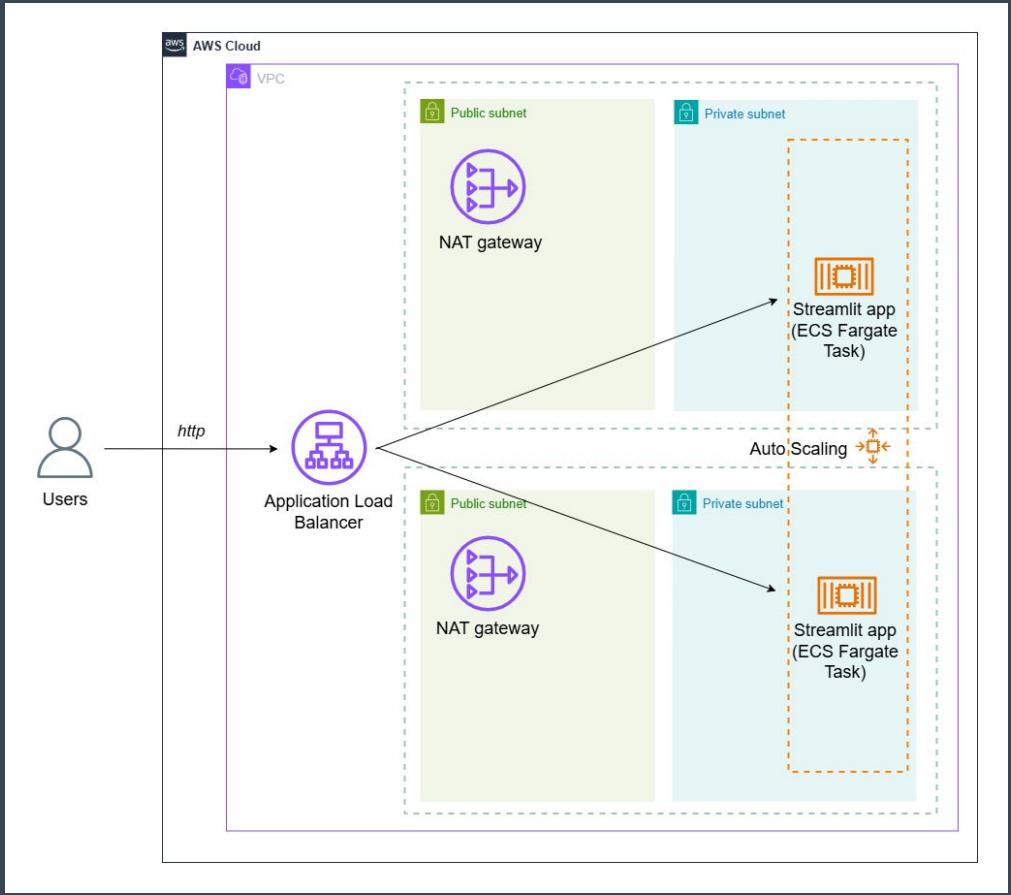
```
if 'mocr' not in st.session_state:
    st.session_state.mocr = MangaOcr(pretrained_model_name_or_path=download_model())
```

Issues are:

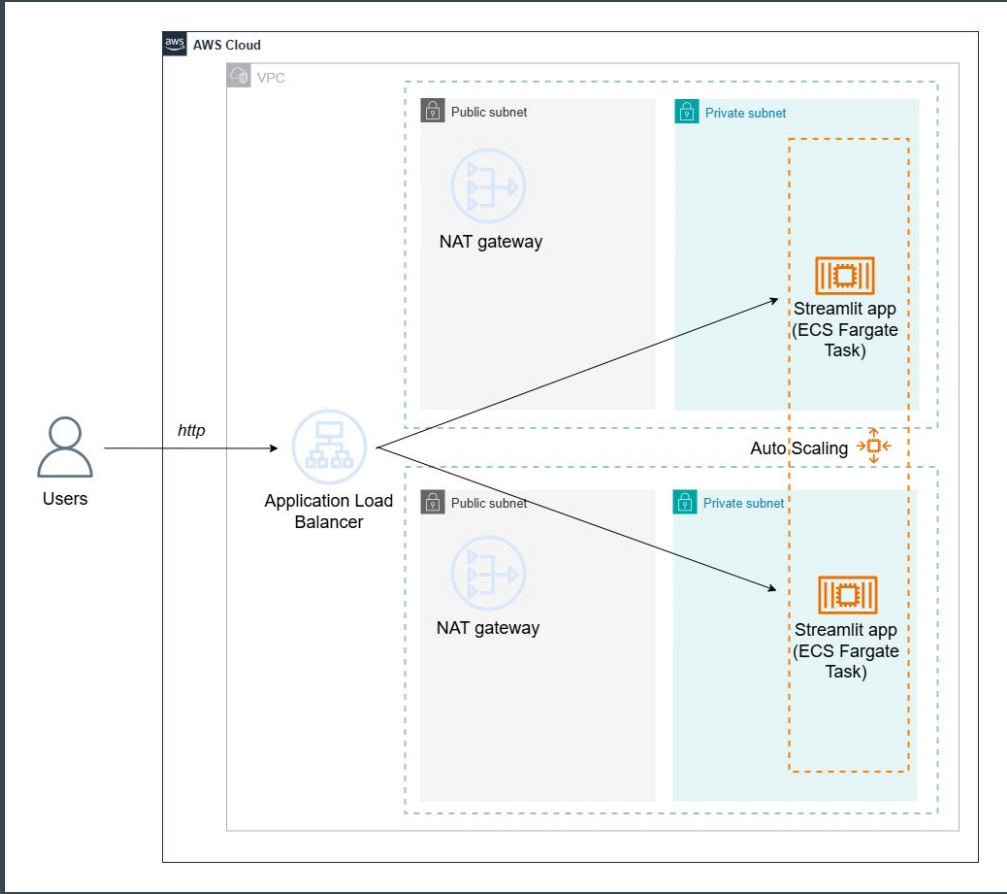
- Large model sizes slowing down app performance.
- Inefficient loading processes increasing costs and user wait times.



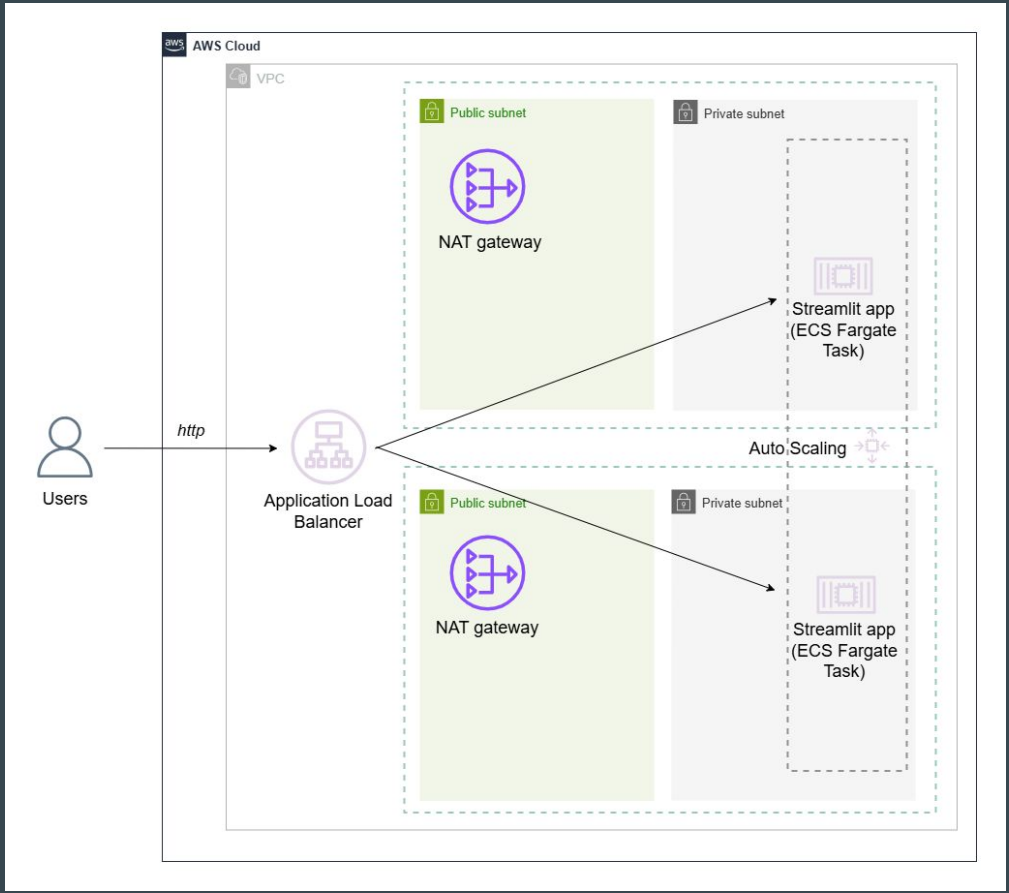
Wait, why is it expensive?



Wait, why is it expensive?



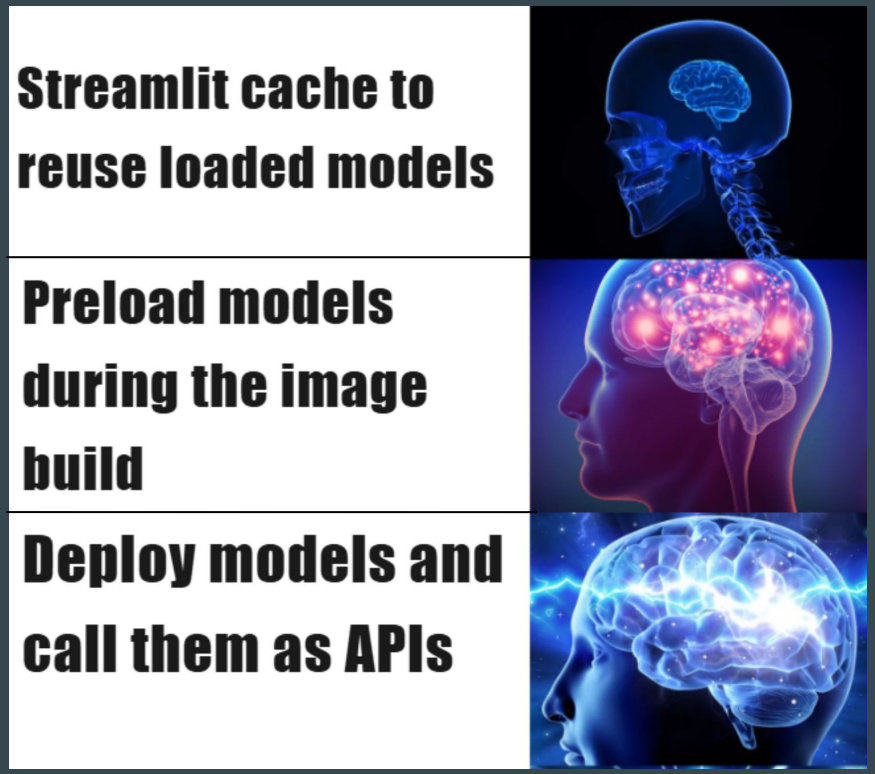
Wait, why is it expensive?



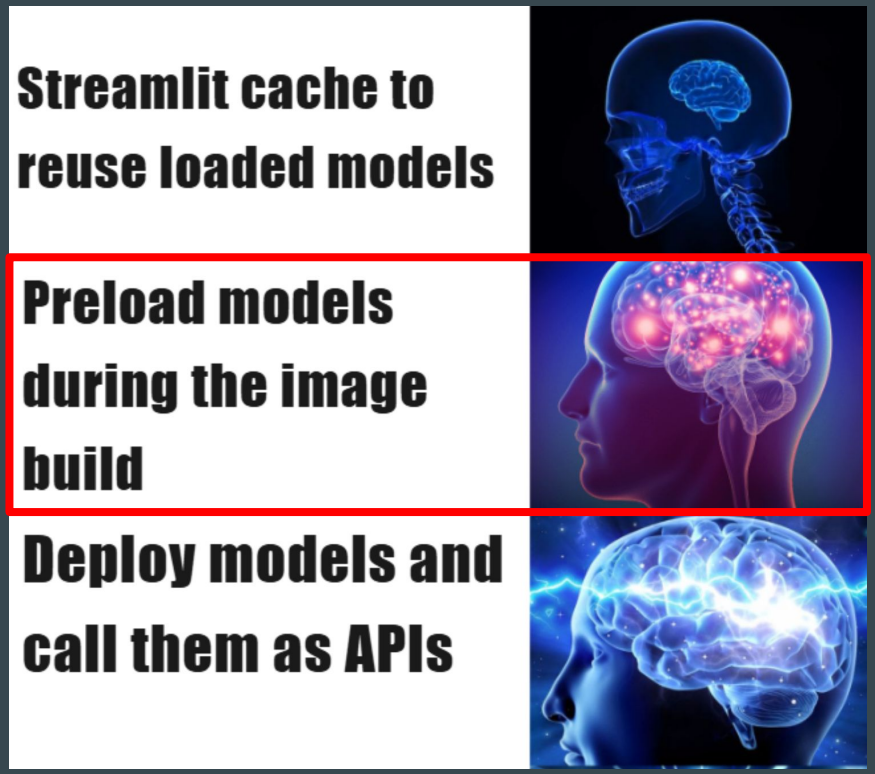
Wait, why is it expensive?

Amazon Elastic Compute Cloud NatGateway		USD 22.91
$0.045 per GB Data Processed by NAT Gateways	57.957 GB	USD 2.61
$0.045 per NAT Gateway Hour	451 Hrs	USD 20.30

How to overcome this disconnection:



How to overcome this disconnection:



How might Data Scientist approach the task:

my-awesome-streamlit-app.py

```python
from huggingface_hub import snapshot_download


def download_model():
    model_dir = "/models/manga-ocr"
    if not os.path.exists(model_dir):
        snapshot_download(repo_id="TareHimself/manga-ocr-base",
                          local_dir=model_dir)
    return model_dir


if 'mocr' not in st.session_state:
    st.session_state.mocr = MangaOcr(pretrained_model_name_or_path=download_model())
```

Better option to approach the task:

preload-model.py

```python
import os
from huggingface_hub import snapshot_download


def download_model():
    """
    Downloads the Manga OCR model from the Hugging Face Hub if it doesn't already exist
locally.
    """
    model_dir = "/models/manga-ocr"

    if not os.path.exists(model_dir):
        snapshot_download(repo_id="TareHimself/manga-ocr-base", local_dir=model_dir)


if __name__ == "__main__":
    download_model()
```

Better option to approach the task:

Dockerfile

```
RUN pip install --no-cache-dir -r requirements.txt huggingface-hub

# Preload the Hugging Face model
RUN python preload_model.py
```

Better option to approach the task:

preload-model.py

```python
import os
from huggingface_hub import snapshot_download


def download_model():
    """
    Downloads the Manga OCR model from the Hugging Face Hub if it doesn't already exist
locally.
    """
    model_dir = "/models/manga-ocr"
```

my-awesome-streamlit-app.py

```python
if 'mocr' not in st.session_state:
    st.session_state.mocr = MangaOcr(pretrained_model_name_or_path="/models/manga-ocr")
```

Disconnection solved

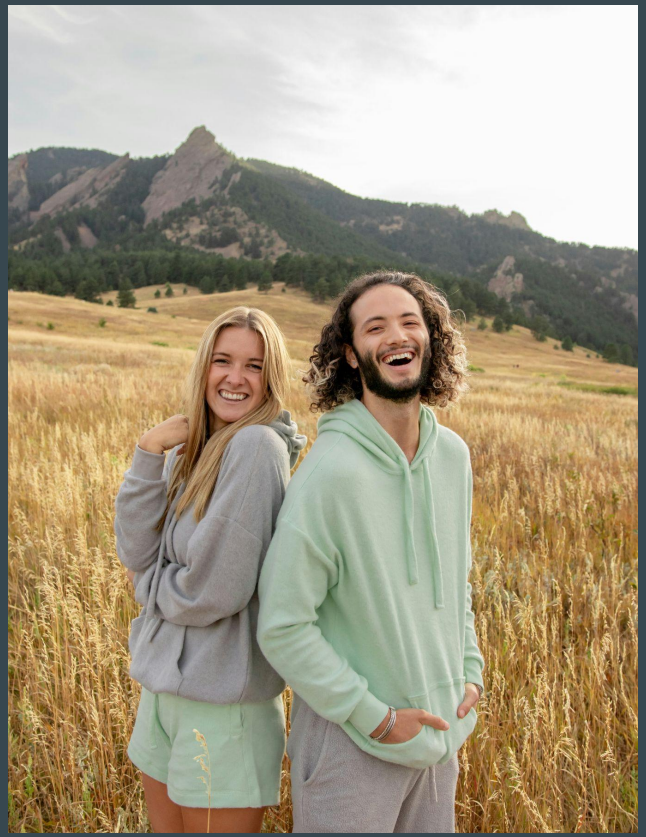


Photo by Woody Kelly on Unsplash

Disconnection 2: autoscaling and authentication

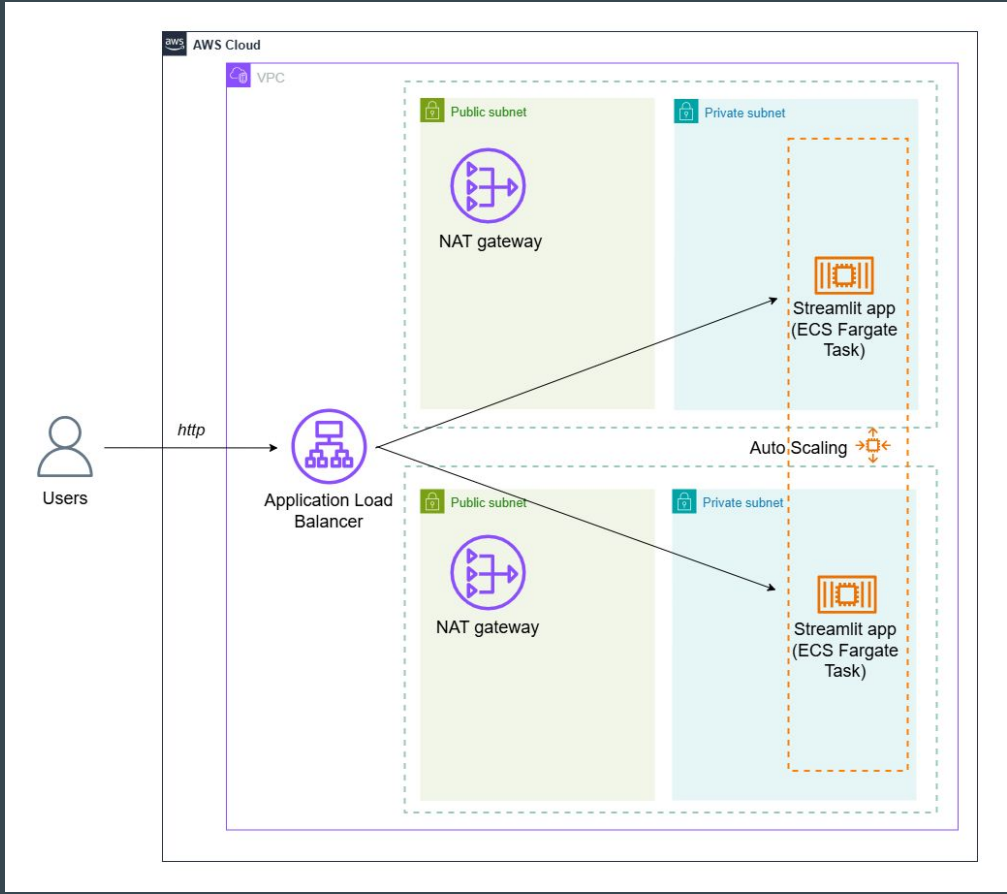
When the app is deployed...

Deployment option	Autoscaling
within Amazon SageMaker Studio using JupyterLab 3	
on EC2 (Elastic Compute Cloud)	
on AWS Fargate + ECS – Managed Containers	

Autoscaling is...

Deployment option	Autoscaling
within Amazon SageMaker Studio using JupyterLab 3	Autoscaling???
on EC2 (Elastic Compute Cloud)	Challenging
on AWS Fargate + ECS – Managed Containers	Easy

Autoscaling with Fargate + ECS



Who will take care of the authentication?

Image source: https://studyfinds.org/kitchen-confrontations-217-arguments-over-cleaning-dishes/

Who will take care of the authentication?

Option	Data Scientist	MLOps engineer
Streamlit-based Streamlit-Authenticator	everything: login widget, auth logic, user privileges, etc.	no tasks
Amazon Cognito	initialise CognitoAuthenticator, authenticate user, check if the user is authenticated (and their permissions)	create an AWS Cognito user pool, configure app client, set up domain name, configure callback URLs

Authentication with Cognito

auth.py

```python
from streamlit_cognito_auth import CognitoAuthenticator
import boto3
import json

# Function to fetch Cognito credentials from AWS Secrets Manager
def get_authenticator(secret_id, region):
    client = boto3.client("secretsmanager", region_name=region)
    secret = json.loads(client.get_secret_value(SecretId=secret_id)['SecretString'])

    return CognitoAuthenticator(
        pool_id=secret['pool_id'],
        app_client_id=secret['app_client_id'],
        app_client_secret=secret['app_client_secret'],
    )
```

Authentication with Cognito

streamlit-app-with-auth.py

```python
import streamlit as st
from auth import get_authenticator


secret_id = "your-secret-id"    # Replace with actual secret ID
region = "your-region"          # Replace with AWS region

authenticator = get_authenticator(secret_id, region)
user = authenticator.login()

if user:
    st.success(f"Welcome, {user['username']}!")
    if st.button("Logout"):
        authenticator.logout()
else:
    st.warning("Please log in.")
```

Disconnection solved

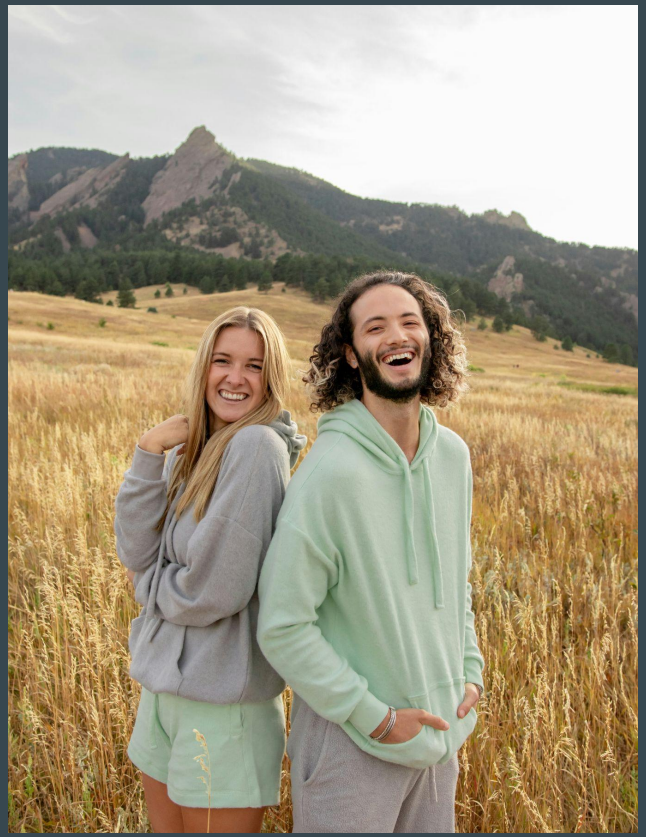
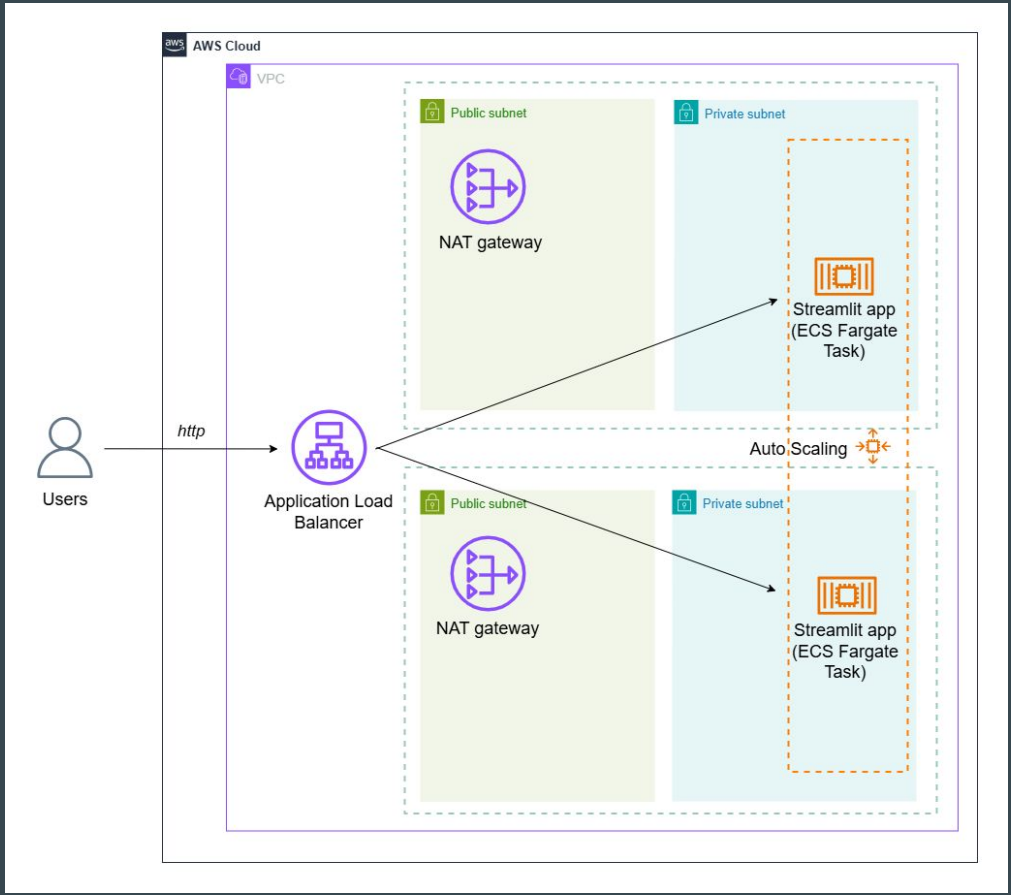


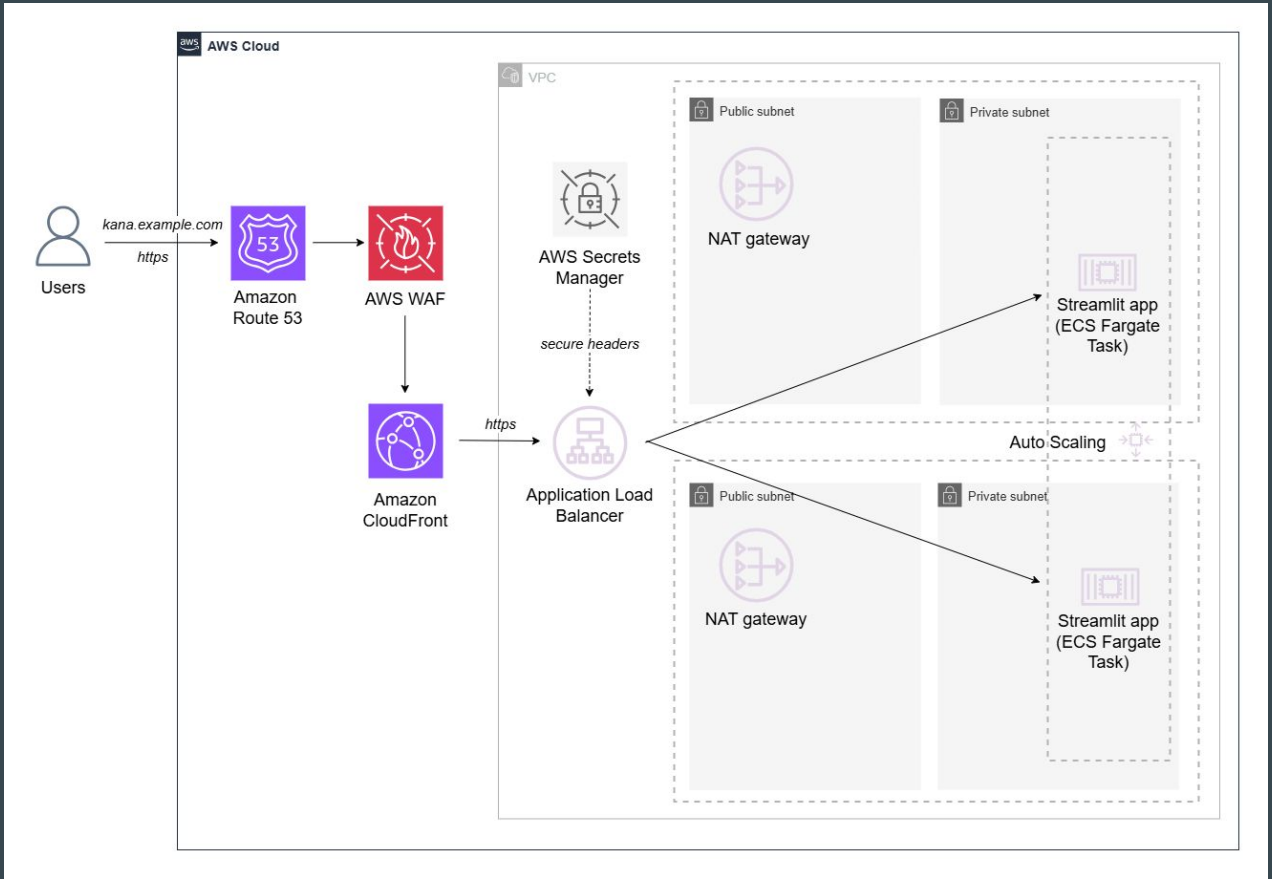
Photo by Woody Kelly on Unsplash

Disconnection 3: security

The app is done..?



Enhancing security



Enhancing security

1. Setting up HTTPS with Route 53 and TLS certificates for secure connections.
2. Configuring CloudFront to protect against DDoS attacks and improve performance.
3. Using AWS Web Application Firewall (WAF) to block malicious traffic.

Disconnection solved

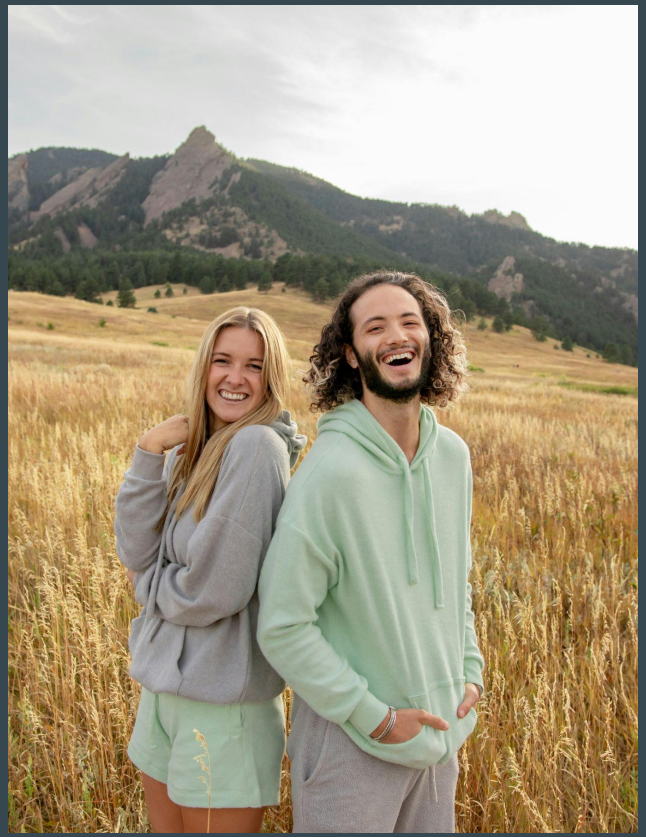
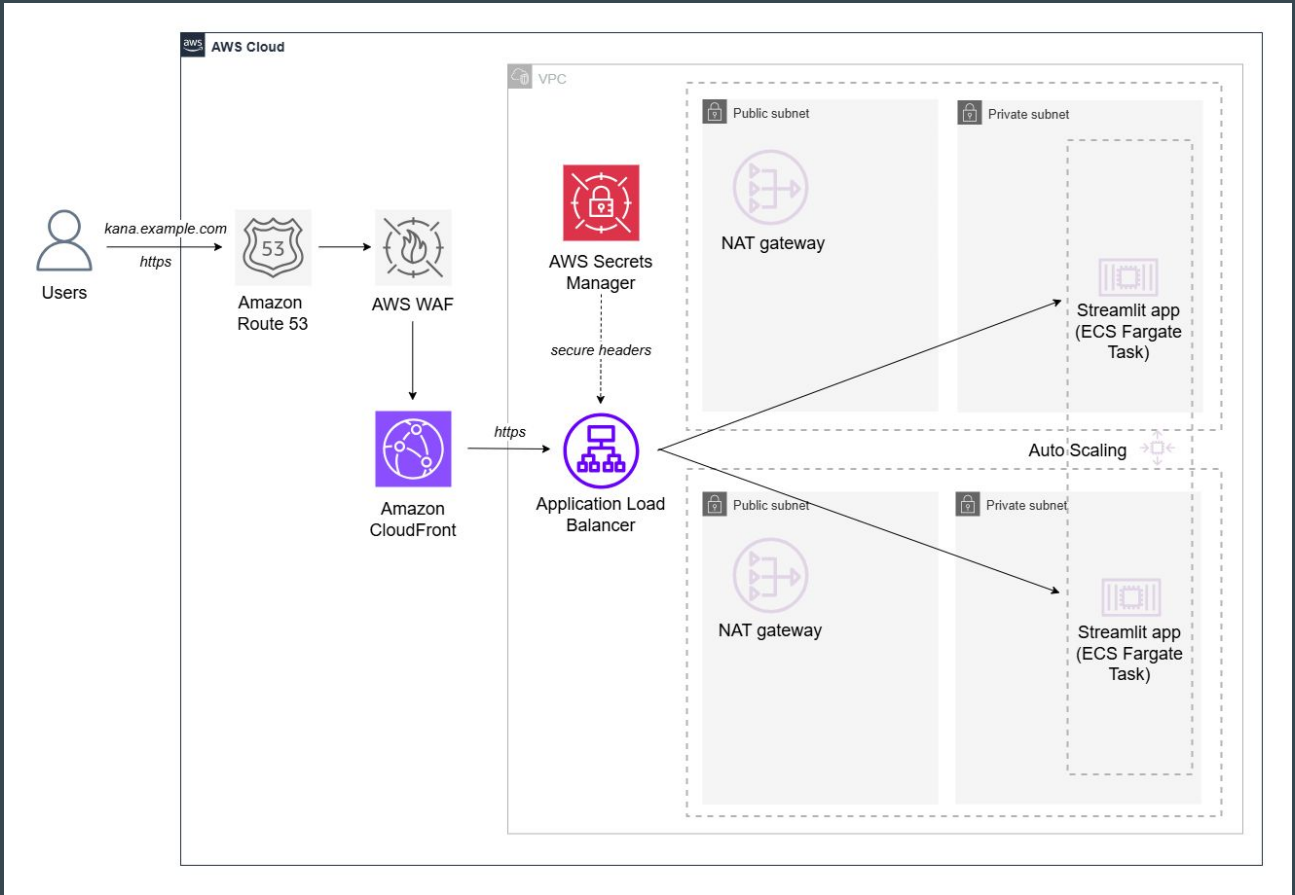


Photo by Woody Kelly on Unsplash

Disconnection 4: sensitive credentials storage

AWS Secrets Manager for secure headers



Handling of sensitive data

my-awesome-streamlit-app.py

```
# 🚨 BAD PRACTICE: Hardcoding sensitive credentials in the script
API_KEY = "sk-1234567890abcdef"  # 🔥 Hardcoded secret (DO NOT DO THIS)
```

Secure handling of sensitive data

my-awesome-streamlit-app.py

```python
import boto3
import json
import os

# Get API Key securely from AWS Secrets Manager
def get_secret(secret_name, region_name="us-east-1"):
    client = boto3.client("secretsmanager", region_name=region_name)
    response = client.get_secret_value(SecretId=secret_name)
    secret = json.loads(response["SecretString"])
    return secret["API_KEY"]

# Securely fetch API Key
secret_name = os.getenv("AWS_SECRET_NAME", "my_secret_api_key")
secrets = get_secret(secret_name)
```

Disconnection solved

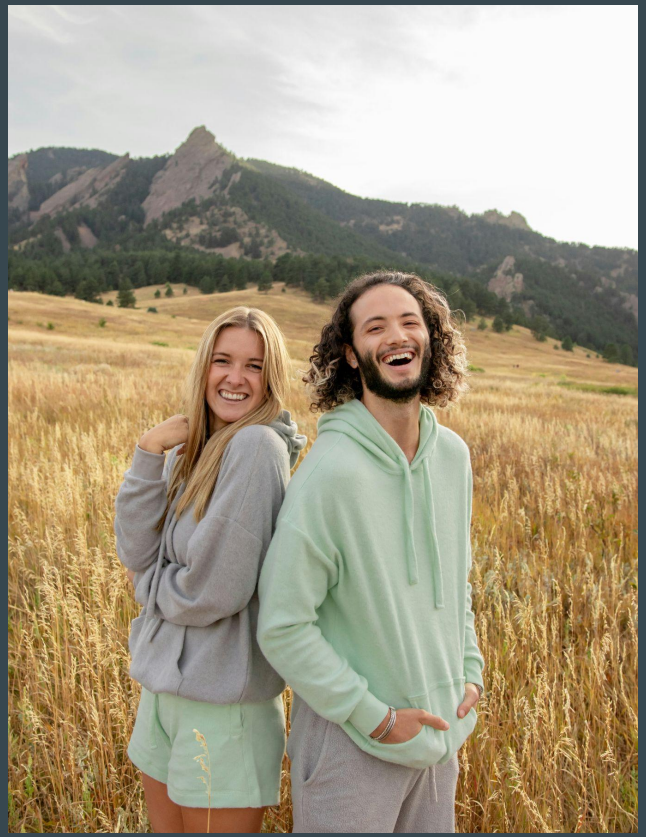


Photo by Woody Kelly on Unsplash

Happily ever after

Happily ever after: takeaways

1. Communication is a key
2. For Data Scientist: consider performance, scalability, authentication, and security during app development
3. For MLOps engineer: collaborate to simplify development while ensuring secure and reliable deployment

If you don’t want to start from scratch

https://github.com/dashapetr/kana--streamlit-app

Q&A