**Pokémon API Scanner & Containerization & Deployment with Helm**

**Scenario 1: Pokémon API Scanner**

**Task:** Develop a Python tool that fetches data from the PokéAPI and retrieves details about a specified Pokémon.

**Requirements:**

* Output must be in JSON format.
* The tool should accept a command-line argument for the Pokémon name.

**Step-by-Step Process:**

1. **Set up the Development Environment:**
   * Install Python 3.x.
   * Install the requests library: pip install requests
2. **Create pokemon\_scanner.py:**

Python

import requests

import json

import argparse

import sys

def fetch\_pokemon\_data(pokemon\_name):

"""Fetches data for a specified Pokémon from the PokéAPI."""

base\_url = "https://pokeapi.co/api/v2/pokemon/"

url = f"{base\_url}{pokemon\_name.lower()}"

try:

response = requests.get(url)

response.raise\_for\_status()

data = response.json()

pokemon\_details = {

"pokemon\_name": data["name"],

"base\_experience": data["base\_experience"],

"height": data["height"],

"weight": data["weight"],

"abilities": [ability["ability"]["name"] for ability in data["abilities"]]

}

return pokemon\_details

except requests.exceptions.RequestException as e:

print(f"Error fetching data: {e}")

return None

except json.JSONDecodeError:

print("Error decoding JSON response.")

return None

except Exception as e:

print(f"An unexpected error occurred: {e}")

return None

if \_\_name\_\_ == "\_\_main\_\_":

parser = argparse.ArgumentParser(description="Fetch details about a specified Pokémon from the PokéAPI.")

parser.add\_argument("pokemon", help="The name of the Pokémon to search for.")

args = parser.parse\_args()

pokemon\_name = args.pokemon

pokemon\_data = fetch\_pokemon\_data(pokemon\_name)

if pokemon\_data:

print(json.dumps(pokemon\_data, indent=4))

else:

print(f"Could not retrieve data for Pokémon: {pokemon\_name}. Please check the name and your internet connection.")

1. **Run the script:**

python pokemon\_scanner.py pikachu

**Scenario 2: Containerizing the Pokémon Scanner**

**Task:** Containerize the script from Scenario 1 using Alpine Linux as the base image and publish it on Docker Hub.

**Requirements:**

* Publish the Docker image on Docker Hub.

**Step-by-Step Process:**

1. **Create Dockerfile:**

Dockerfile

FROM python:3.9-alpine3.18

WORKDIR /app

COPY requirements.txt .

RUN pip install -r requirements.txt

COPY pokemon\_scanner.py .

CMD ["python", "pokemon\_scanner.py"]

1. **Create requirements.txt:**
2. requests
3. **Build the Docker Image:**

docker build -t pokemon-scanner .

1. **Tag the Image:**

docker tag pokemon-scanner akhilaraju247/pokemon-scanner:latest

1. **Log in to Docker Hub:**

docker login

1. **Push the Image:**

docker push akhilaraju247/pokemon-scanner:latest

**Scenario 3: Deploying with Helm on Minikube**

**Task:** Create a Helm chart to deploy the container from Scenario 2 on Minikube for manual execution.

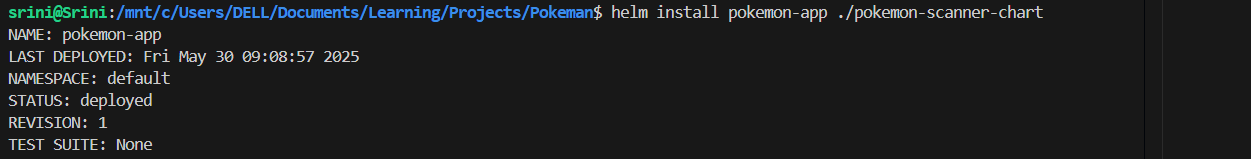
**Requirements:**

* The pod should not auto-execute but should be accessible for manual execution.

**Step-by-Step Process:**

1. **Prerequisites:** Minikube and Helm installed and running.
2. **Create Helm Chart:**

helm create pokemon-scanner-chart



1. **Modify pokemon-scanner-chart/Chart.yaml:**

YAML

1. apiVersion: v2
2. name: pokemon-scanner-chart
3. description: A Helm chart for the Pokémon API Scanner.
4. version: 0.1.0
5. appVersion: 1.0.0
6. **Modify pokemon-scanner-chart/values.yaml:**

YAML

1. image:
2. repository: akhilaraju247/pokemon-scanner:latest
3. tag: latest
4. pullPolicy: IfNotPresent
5. resources: {}
6. **Modify pokemon-scanner-chart/templates/manual-pod.yaml:**

YAML

1. apiVersion: v1
2. kind: Pod
3. metadata:
4. name: pokemon-scanner-manual-pod
5. labels:
6. app: pokemon-scanner
7. spec:
8. dnsPolicy: ClusterFirstWithHostNet
9. # dnsConfig:
10. #   nameservers:
11. #     - 8.8.8.8
12. containers:
13. - name: pokemon-scanner
14. image: "local-pokemon-scanner:latest"
15. imagePullPolicy: Never
16. command: ["/bin/sh", "-c", "echo 'Pod is ready for manual execution. Use kubectl exec -it pokemon-scanner-manual-pod -- python pokemon\_scanner.py <pokemon-name>' && sleep 3600"]
17. **Remove templates/tests and simplify templates:**

rm -rf pokemon-scanner-chart/templates/tests

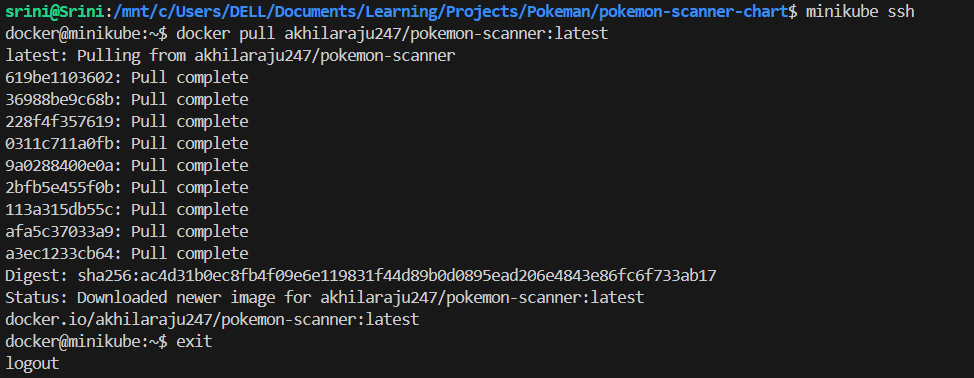
1. **Pull Image to Minikube:**

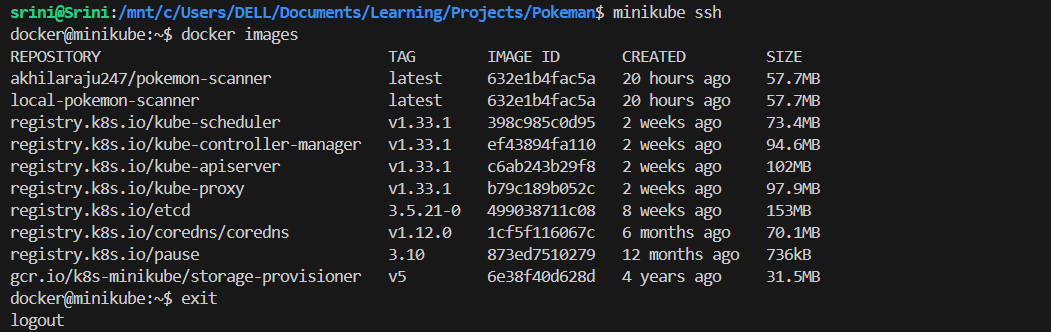
minikube ssh

docker tag pokemon-scanner akhilaraju247/pokemon-scanner:latest

docker push akhilaraju247/pokemon-scanner:latest

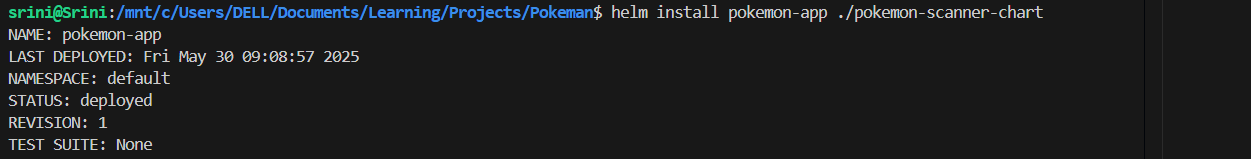
exit

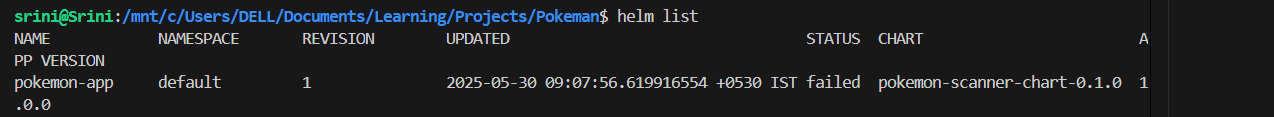




1. **Install Helm Chart:**

helm install pokemon-app ./pokemon-scanner-chart





1. **Manually Execute Scanner:**

kubectl exec -it pokemon-scanner-manual-pod -- python pokemon\_scanner.py pikachu

