In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
```

In [4]:
```
# Load the Iris dataset
df = pd.read_csv("Iris.csv")
```

In [6]:
```
# Select features (attributes) for clustering (e.g., sepal_length, sepal_width, pet
X = df.iloc[:, 1:-1]  # Exclude the first column (id) and the last column (species)
```

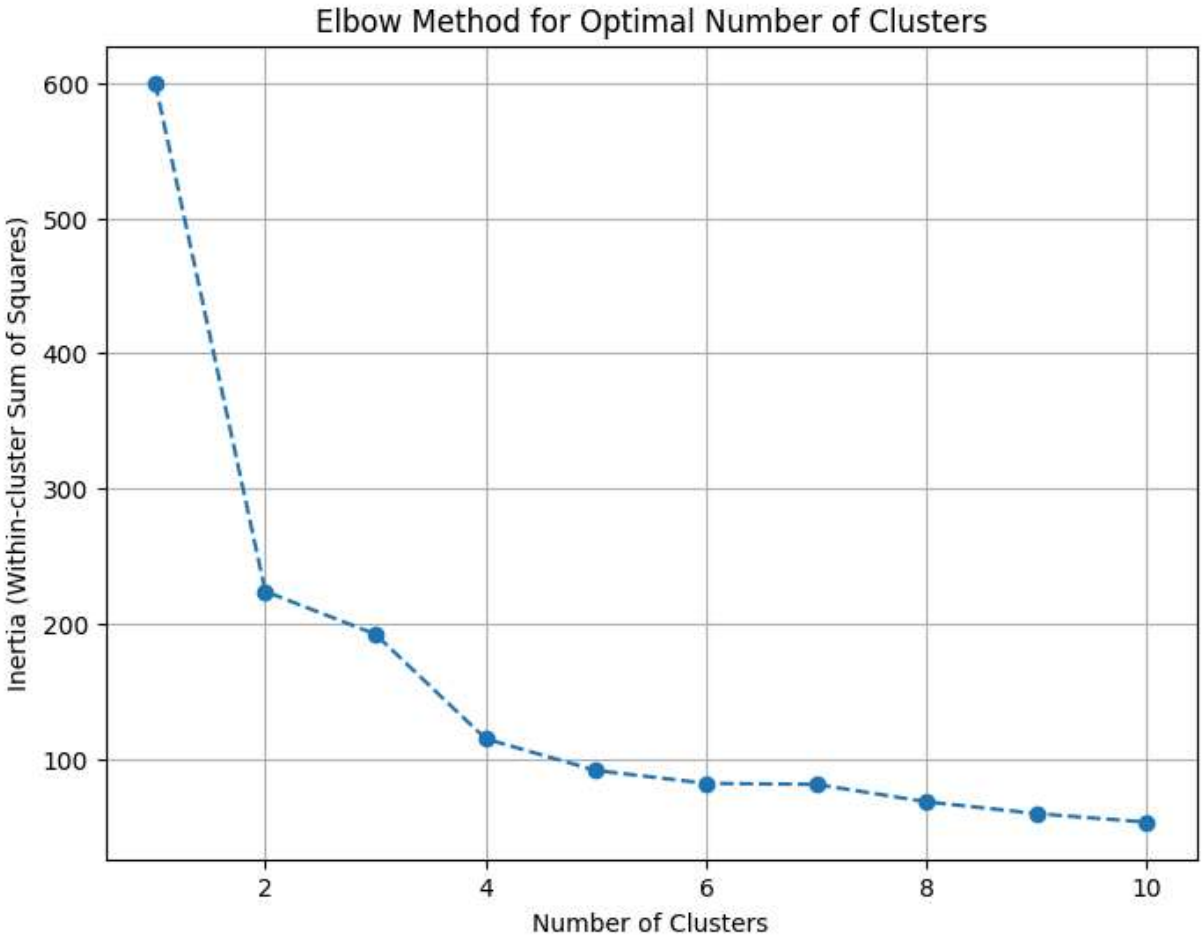
In [8]:
```
# Standardize the feature matrix (important for K-Means)
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

In [10]:
```
# Determine the optimal number of clusters using the elbow method
inertia = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(X_scaled)
    inertia.append(kmeans.inertia_)
```

In [11]:
```
print(inertia)
```

```
[600.0, 223.73200573676345, 192.03717409190028, 114.68221609937967, 91.295444740669
8, 81.76026132860625, 80.98238131032988, 68.08623905064636, 59.38528882045367, 52.98
999721015859]
```

In [14]:
```
# Plot the elbow curve
plt.figure(figsize=(8, 6))
plt.plot(range(1, 11), inertia, marker='o', linestyle='--')
plt.xlabel('Number of Clusters')
plt.ylabel('Inertia (Within-cluster Sum of Squares)')
plt.title('Elbow Method for Optimal Number of Clusters')
plt.grid()
plt.show()
```


Elbow Method for Optimal Number of Clusters
Inertia (Within-cluster Sum of Squares)
600
500
400
300
200
100
2
4
6
8
10
Number of Clusters

In [ ]:

In [ ]: