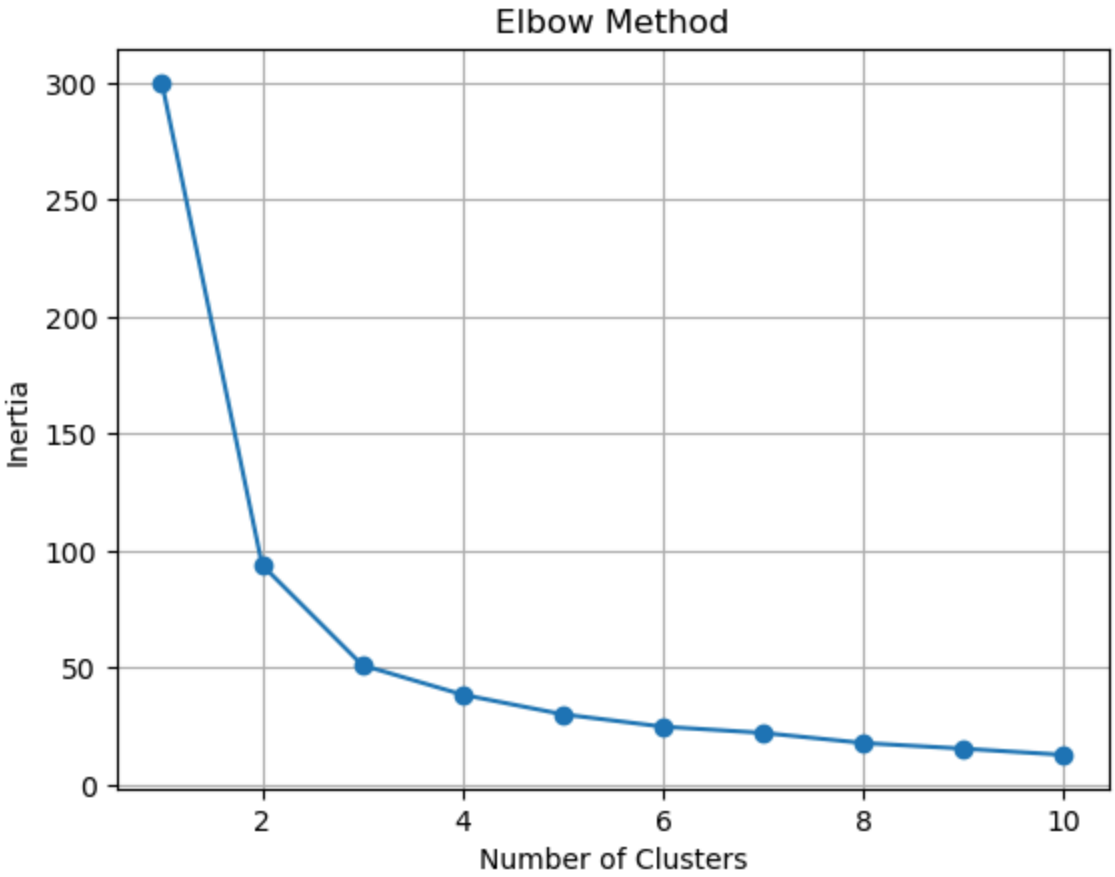
Task 3: Clustering (K-Means on Iris Dataset)

1. Import Libraries

In [5]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
import os
os.environ["OMP_NUM_THREADS"] = "1"
import warnings
warnings.filterwarnings('ignore')
```

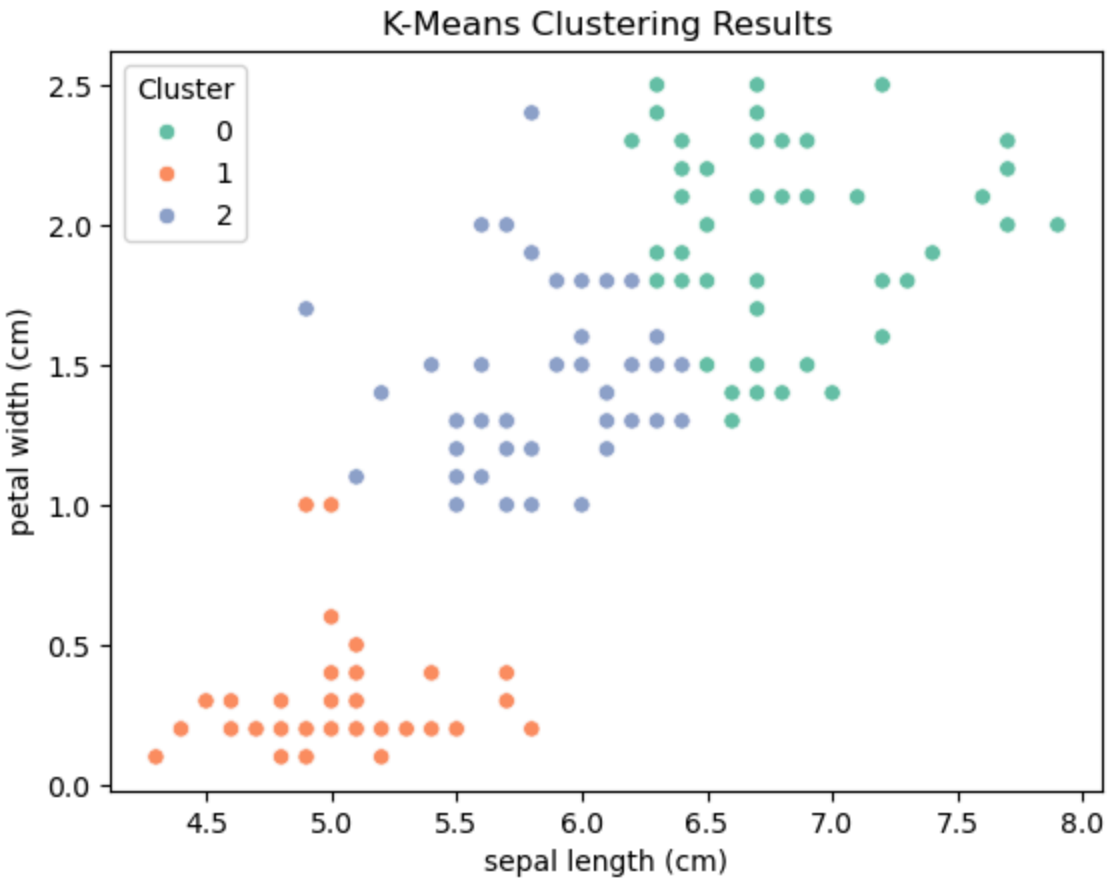
2. Elbow Method to Find Optimal Clusters¶

In [8]:
```
from sklearn.datasets import load_iris

# Load Iris dataset
iris = load_iris()
df = pd.DataFrame(iris.data, columns=iris.feature_names)

# Select two real features
X = df[['sepal length (cm)', 'petal width (cm)']]

# Standardize
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Elbow method to find optimal k
inertia = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(X_scaled)
    inertia.append(kmeans.inertia_)

# Plot elbow curve
plt.plot(range(1, 11), inertia, marker='o')
plt.title('Elbow Method')
plt.xlabel('Number of Clusters')
plt.ylabel('Inertia')
plt.grid(True)
plt.show()
```



3.K-Means Clustering

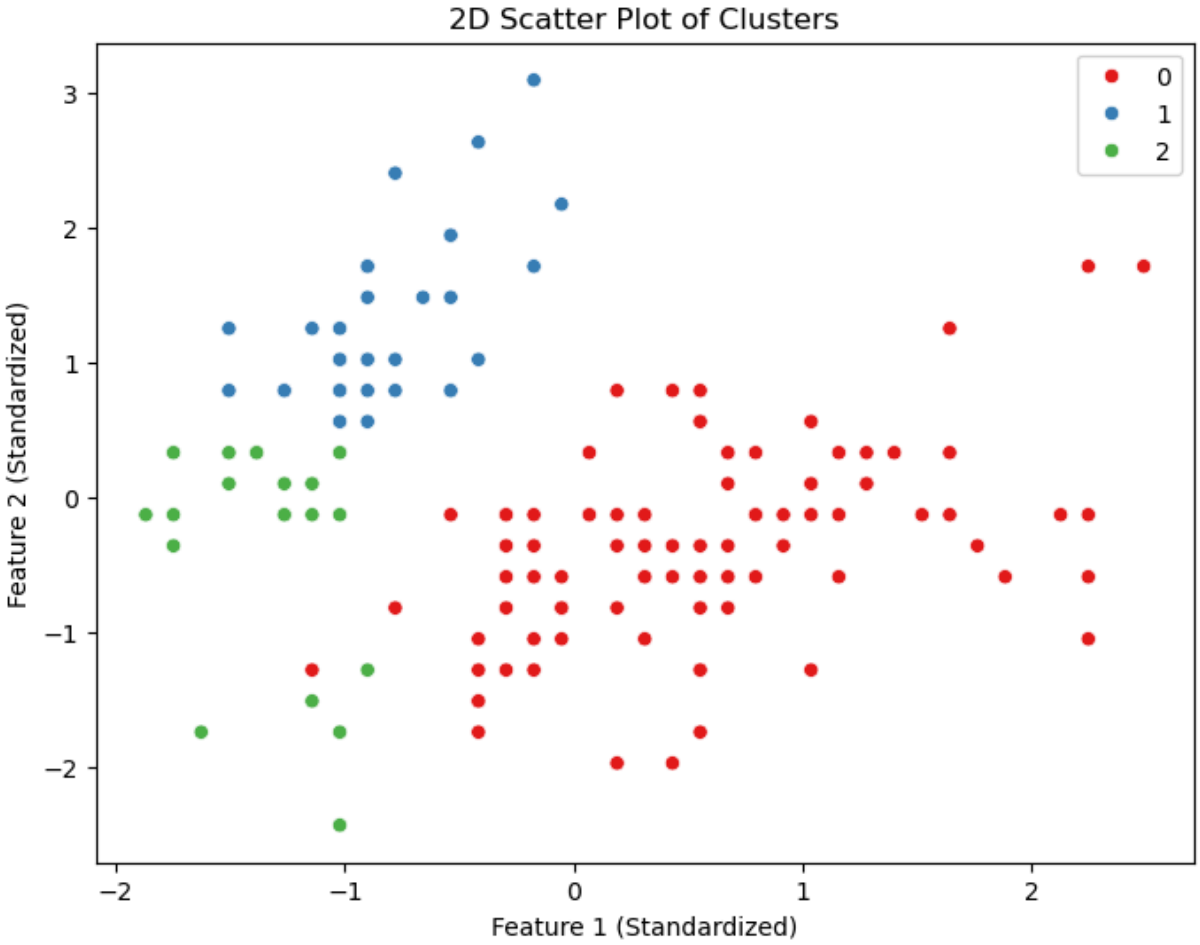
In [7]:
```
# Fit KMeans with optimal k (e.g., 3)
kmeans = KMeans(n_clusters=3, random_state=42)
df['Cluster'] = kmeans.fit_predict(X_scaled)

# Visualize clusters
sns.scatterplot(x='sepal length (cm)', y='petal width (cm)', hue='Cluster', data=df
plt.title('K-Means Clustering Results')
plt.show()
```



3. Visualize Clusters (2D Scatter Plot)

In [13]:

```
from sklearn.datasets import load_iris
from sklearn.preprocessing import StandardScaler


# Load and scale data
iris = load_iris()
X = iris.data
scaler = StandardScaler()
scaled_data = scaler.fit_transform(X)

# Apply KMeans clustering
kmeans = KMeans(n_clusters=3, random_state=42)
clusters = kmeans.fit_predict(scaled_data)

# 2D Scatter Plot
plt.figure(figsize=(8, 6))
sns.scatterplot(x=scaled_data[:, 0], y=scaled_data[:, 1], hue=clusters, palette='Se
plt.xlabel('Feature 1 (Standardized)')
plt.ylabel('Feature 2 (Standardized)')
plt.title('2D Scatter Plot of Clusters')
plt.show()
```


2D Scatter Plot of Clusters
Feature 2 (Standardized)
Feature 1 (Standardized)
0
1
2
3
2
1
0
−1
−2
−2
−1
0
1
2