Resilient Distributed Datasets (RDDs) are a fundamental data structure in Apache Spark, a distributed computing framework. RDDs are fault-tolerant, parallel collections of data that can be processed in a distributed manner across a cluster of machines. They support both parallel processing and fault tolerance, making them suitable for large-scale data processing.

Creating RDDs:

Example 1: Creating an RDD from a local collection
```python
from pyspark import SparkContext

# Create a SparkContext
sc = SparkContext("local", "RDD Example")

# Create an RDD from a local collection (list)
data = [1, 2, 3, 4, 5]
rdd = sc.parallelize(data)

# Print the content of the RDD
print("RDD content:", rdd.collect())

# Stop the SparkContext
sc.stop()
```

Example 2: Creating an RDD from an external data source (text file)
```python
from pyspark import SparkContext

# Create a SparkContext
sc = SparkContext("local", "RDD Example")

# Create an RDD from an external text file
file_path = "path/to/textfile.txt"
rdd = sc.textFile(file_path)

# Print the content of the RDD
print("RDD content:", rdd.collect())

# Stop the SparkContext
sc.stop()
```

Transformations and Actions:

Example 3: RDD Transformation - Map
```python
from pyspark import SparkContext

# Create a SparkContext
sc = SparkContext("local", "RDD Transformation Example")

# Create an RDD from a local collection (list)
data = [1, 2, 3, 4, 5]
rdd = sc.parallelize(data)

# Use the map transformation to square each element
squared_rdd = rdd.map(lambda x: x**2)

# Print the content of the transformed RDD
print("Squared RDD content:", squared_rdd.collect())

# Stop the SparkContext
sc.stop()
```

Example 4: RDD Action - Reduce
```python
from pyspark import SparkContext

# Create a SparkContext
sc = SparkContext("local", "RDD Action Example")

# Create an RDD from a local collection (list)
data = [1, 2, 3, 4, 5]
rdd = sc.parallelize(data)

# Use the reduce action to find the sum of all elements
sum_result = rdd.reduce(lambda x, y: x + y)

# Print the result of the reduce action
print("Sum of RDD elements:", sum_result)

# Stop the SparkContext
sc.stop()
```

In these examples:

- **Creating RDDs:** Example 1 demonstrates creating an RDD from a local collection, while Example 2 shows creating an RDD from an external text file.
- **Transformations:** Example 3 illustrates the map transformation, where each element of the RDD is squared.
- **Actions:** Example 4 showcases the reduce action, computing the sum of all elements in the RDD.

Certainly! Let's continue with more examples of RDD transformations and actions:

Transformations:

Example 5: RDD Transformation - Filter
```python
from pyspark import SparkContext

# Create a SparkContext
sc = SparkContext("local", "RDD Transformation Example")

# Create an RDD from a local collection (list)
data = [1, 2, 3, 4, 5]
rdd = sc.parallelize(data)

# Use the filter transformation to keep only even numbers
even_rdd = rdd.filter(lambda x: x % 2 == 0)

# Print the content of the transformed RDD
print("Even numbers in RDD:", even_rdd.collect())

# Stop the SparkContext
sc.stop()
```

Example 6: RDD Transformation - FlatMap
```python
from pyspark import SparkContext

# Create a SparkContext
sc = SparkContext("local", "RDD Transformation Example")

# Create an RDD from a local collection (list) of words
data = ["Hello world", "Spark is amazing", "Resilient Distributed Datasets"]
rdd = sc.parallelize(data)

# Use the flatMap transformation to split lines into words
```

```
words_rdd = rdd.flatMap(lambda line: line.split(" "))

# Print the content of the transformed RDD
print("Words in RDD:", words_rdd.collect())

# Stop the SparkContext
sc.stop()
```

Actions:

Example 7: RDD Action - Count
```python
from pyspark import SparkContext

# Create a SparkContext
sc = SparkContext("local", "RDD Action Example")

# Create an RDD from a local collection (list)
data = [1, 2, 3, 4, 5]
rdd = sc.parallelize(data)

# Use the count action to get the number of elements in the RDD
element_count = rdd.count()

# Print the result of the count action
print("Number of elements in RDD:", element_count)

# Stop the SparkContext
sc.stop()
```

Example 8: RDD Action - Collect
```python
from pyspark import SparkContext

# Create a SparkContext
sc = SparkContext("local", "RDD Action Example")

# Create an RDD from a local collection (list)
data = [1, 2, 3, 4, 5]
rdd = sc.parallelize(data)

# Use the collect action to retrieve all elements from the RDD
```

```
all_elements = rdd.collect()

# Print the result of the collect action
print("All elements in RDD:", all_elements)

# Stop the SparkContext
sc.stop()
```

In these additional examples:
- **Transformations:** Example 5 demonstrates the use of the filter transformation to keep only even numbers, while Example 6 uses the flatMap transformation to split lines into words.
- **Actions:** Example 7 shows the count action to get the number of elements in the RDD, and Example 8 illustrates the collect action to retrieve all elements from the RDD.