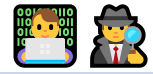


## 02 Running Spark Job on Dataproc -KirkYagami



### Cloud Dataproc - Cluster and Jobs

#### Step-01: Introduction

- ♦ Create Dataproc Single Node Cluster
- ♦ **Create Job1**: sort-words-job and verify
- ♦ **Create Job2**: distinct-list-job and verify

#### Step-02: Create Dataproc Cluster

- ♦ Go to Dataproc -> CREATE CLUSTER
- ♦ Create Dataproc cluster: Cluster on Compute Engine

#### Setup Cluster

- ♦ Cluster Name: mydataproc-cluster
- ♦ Region: us-central1
- ♦ Zone: any
- ♦ Cluster Type: Single Node (1 master, 0 workers)
- ♦ REST ALL LEAVE TO DEFAULTS

#### Configure Nodes

- ♦ Primary Disk Size: 50GB
- ♦ REST ALL LEAVE TO DEFAULTS

#### Customize Cluster

- ♦ LEAVE TO DEFAULTS

#### Manage Security

- ♦ LEAVE TO DEFAULTS
- ♦ Click on **CREATE**

#### Step-03: Review Dataproc Cluster

- ♦ Dataproc Cluster
  - ♦ MONITORING
  - ♦ JOBS
  - ♦ VM INSTANCES
  - ♦ CONFIGURATION
  - ♦ WEB INTERFACES
- ♦ Go to Compute Instances and verify VM instance created for cluster

#### Step-04: Review Python files and Upload to Cloud Storage

- ◆ These files will be used to create Jobs

## Step-04-01: distinct-list.py

---

```
#!/usr/bin/python
import pyspark

# Create Number List
numbers = [1,2,3,1,2,3,4,4,2,3,6,6,7,2,2,1,3,4,5,8,1,2]

# Python SparkContext
sc = pyspark.SparkContext()

# Create RDD with parallelize method of SparkContext
rdd = sc.parallelize(numbers)

# Return distinct elements from RDD
distinct_numbers = rdd.distinct().collect()

# Print distinct numbers which we can verify in Cloud Dataproc Logs
print('Distinct Numbers:', distinct_numbers)
```

## Step-04-02: sort-words.py

---

```
import pyspark

sc = pyspark.SparkContext()
rdd = sc.parallelize(["orange", "pear", "date", "grape", "banana", "kiwi", "cherry",
"fig", "lemon", "mango", "apple"])
words = sorted(rdd.collect())
print(words)
```

## Step-04-03: Upload files to Cloud Storage Bucket

---

- ◆ Create / Use existing Cloud Storage Bucket
- ◆ Cloud Storage Bucket: data2468
- ◆ Upload files
  - ◆ sort-words.py
  - ◆ distinct-list.py

## Step-05: Create Dataproc Job and Verify Job logs

---

### Step-05-01: Create Dataproc Job

---

- ◆ Go to Dataproc -> Jobs on Clusters -> Jobs -> **SUBMIT JOB**
- ◆ **Job ID:** sort-words-job
- ◆ **Region:** us-central1
- ◆ **Cluster:** mydataproc-cluster
- ◆ **Job Type:** PySpark

- ♦ **Main Python File:** gs://data2468/sort-words.py
- ♦ REST ALL LEAVE TO DEFAULTS
- ♦ Click on **CREATE**

## Step-05-02: Verify Output Job Logs

---

- ♦ Go to Dataproc -> Jobs on Clusters -> Jobs -> **sort-words-job**
- ♦ Verify output job logs
- ♦ **Observation:** All the words in the list will be sorted in alphabetical order

## Step-06: Create Dataproc Job and Verify Job logs

---

### Step-06-01: Create Dataproc Job

---

- ♦ Go to Dataproc -> Jobs on Clusters -> Jobs -> **SUBMIT JOB**
- ♦ **Job ID:** distinct-list-job
- ♦ **Region:** us-central1
- ♦ **Cluster:** mydataproc-cluster
- ♦ **Job Type:** PySpark
- ♦ **Main Python File:** gs://data2468/distinct-list.py
- ♦ REST ALL LEAVE TO DEFAULTS
- ♦ Click on **CREATE**

### Step-06-02: Verify Output Job Logs

---

- ♦ Go to Dataproc -> Jobs on Clusters -> Jobs -> **distinct-list-job**
- ♦ Verify output job logs
- ♦ **Observation:** All the words in the list will be sorted in alphabetical order

## Step-07: gcloud Commands: Cloud Dataproc: CleanUp

---

```
# Set Project
gcloud config set project PROJECT_ID
gcloud config set project bigdata3844

# Set Cloud Dataproc Region
gcloud config set dataproc/region VALUE
gcloud config set dataproc/region us-central1

# List Jobs
gcloud dataproc jobs list

# List Clusters
gcloud dataproc clusters list

# Delete Cluster
gcloud dataproc clusters delete mydataproc-cluster1

# List Clusters
gcloud dataproc clusters list
```



```
gcloud dataproc clusters create pyspark-cluster2 \  
  --enable-component-gateway \  
  --region=us-central1 \  
  --zone= \  
  --master-machine-type=e2-standard-2 \  
  --worker-machine-type=e2-standard-2 \  
  --num-workers=2 \  
  --master-boot-disk-size=50GB \  
  --worker-boot-disk-size=50GB \  
  --image-version=2.1-ubuntu20 \  
  --optional-components=JUPYTER
```



```
gcloud dataproc clusters create pyspark-cluster --enable-component-gateway --region us-  
central1 --master-machine-type e2-standard-2 --master-boot-disk-type pd-balanced --master-  
boot-disk-size 50 --num-workers 2 --worker-machine-type e2-standard-2 --worker-boot-disk-  
type pd-balanced --worker-boot-disk-size 50 --image-version 2.1-ubuntu20 --optional-  
components JUPYTER --project bigdata3844
```