# Google's Cloud Dataproc

- Dataproc is a fully managed and highly scalable service for running Apache Hadoop, Apache Spark, Apache Flink, Presto, and 30+ open source tools and frameworks for batch processing, querying, streaming, and machine learning.
- Dataproc is built on open source platforms, including
  - **Apache Hadoop -** supports the distributed processing of large data sets across clusters
  - **Apache Spark -** serves as the engine for fast, large-scale data processing
  - **Apache Pig -** analyzes large data sets
  - **Apache Hive -** provides data warehousing and SQL database storage management
- Dataproc is used for data lake modernization, ETL, and secure data science, at scale, integrated with Google Cloud, at a fraction of the cost.
- Dataproc is a managed cluster service in GCP.
- Dataproc automation helps you create clusters quickly, manage them easily, and save money by turning clusters off when you don't need them.
- The most compelling feature of Dataproc as a managed service, is its billing model. Standing up a cluster ready to accept jobs takes about 90 seconds and cluster resources are billed per second with a minimum billing period of just 1 minute.
- This means you can quickly set up resources to process a dataset or scale up an existing cluster when you're expecting a larger than normal workload then scale it down again or even turn it off and only pay for what you use.
- Dataproc is completely interoperable and compatible with up-to-date versions of these open-source tools.

- Cloud Dataproc easily integrates with the following Google Cloud Platform services such as BigQuery, Bigtable, Google Cloud Storage, Stackdriver Monitoring, and Stackdriver Logging.
- Users can develop Dataproc in languages that are popular within the Spark and Hadoop ecosystem, such as Java, Scala, Python and R.
- The main benefits of dataproc are that:
    - It's a managed service, so you don't need a system administrator to set it up.
    - Dataproc supports native versions of Hadoop, Spark, Pig and Hive, allowing users to employ the latest versions of each platform, as well as the entire ecosystem of related open source tools and libraries.
    - It means that the users have control over upgrading and using the latest versions of each of these platforms.
    - It's fast. You can spin up a cluster in about 90 seconds.
    - It's cheaper than building your own cluster because you can spin up a Dataproc cluster when you need to run a job and shut it down afterward, so you only pay when jobs are running.
- Dataproc is available in three flavours:
    - **Dataproc Serverless** allows you to run PySpark jobs without needing to configure infrastructure and autoscaling. Dataproc Serverless supports PySpark batch workloads and sessions / notebooks.
    - **Dataproc on Google Compute Engine** allows you to manage a Hadoop YARN cluster for YARN-based Spark workloads in addition to open source tools such as Flink and Presto. You can tailor your cloud-based clusters with as much vertical or horizontal scaling as you'd like, including autoscaling.
    - **Dataproc on Google Kubernetes Engine** allows you to configure Dataproc virtual clusters in your GKE infrastructure for submitting Spark, PySpark, SparkR or Spark SQL jobs.

## Characteristics of Dataproc

- Dataproc is created as
  - **Regional resources**
    - If you choose regional, you can isolate all the resources used for Dataproc into a specific region such as, us-east1, or europe-west1.
    - You may need to do this for a variety of compliance or performance reasons.
    - When you specify a region, you can then choose the zone yourself or have Dataproc pick one for you.
  - **Global Resources**
    - Next one is you can choose global. This doesn't actually mean cluster is automatically a global resource.
    - It just means that it's not tied to a specific region, and we can place its resources in any available zone, worldwide.
- **Single Node Cluster(1 master, 0 workers)**
  - A single node cluster has a single VM that will run the master and work the processes.
  - Single nodes are just limited to the capacity of a single VM, and you can't auto scale a single node cluster.
- **Standard cluster (1 master, N workers)**
  - A standard cluster is most likely option.
  - This comprises a master VM, which runs the YARN Resource Manager and HDFS Name Node and 2 or more worker nodes, which each provide a YARN Node Manager and HDFS Data Node.
  - This machine is fully configurable like amount and type of disk

- With a standard cluster we can also add additional preemptible workers which is sometimes useful for large processing jobs, but it can't provide storage for HDFS.

- **High availability cluster (3 masters, N workers)**
  - For long running Dataproc clusters, where we need to be able to guarantee reliability, final option is a high availability cluster.
  - This is similar to a standard cluster, except you now get 3 masters, with YARN and HDFS configured to run in high availability mode, providing uninterrupted operations in the event of any single node failure or reboot.
- **Submit jobs**
  - gcloud Command Line tool
  - GCP Console
  - cloud Dataproc HTTP API
  - SSH into the Master Node and submit a job locally
- Dataproc will accept several different types of job,
  - Hadoop
  - SparkR
  - Spark SQL
  - Presto
  - Spark
  - Hive
  - Pig
  - PySpark.
- Jobs are created in a pending state then move to a running state while they're being processed by the Dataproc agent and the YARN cluster.
- When a job is complete, it enters the done state. These states can be queried by the Dataproc API or simply viewed in the GCP console.

# Features of Cloud Dataproc

| | |
|---|---|
| **Serverless Spark** | • Deploy Spark applications and pipelines that autoscale without any manual infrastructure provisioning or tuning. |
| **Resizable clusters** | • Create and scale clusters quickly with various virtual machine types, disk sizes, number of nodes, and networking options. |
| **Autoscaling clusters** | • Dataproc autoscaling provides a mechanism for automating cluster resource management and enables automatic addition and subtraction of cluster workers (nodes). |
| **Versioning** | • Image versioning allows you to switch between different versions of Apache Spark, Apache Hadoop, and other tools. |
| **Cluster scheduled deletion** | • To help avoid incurring charges for an inactive cluster, you can use Dataproc's scheduled deletion, which provides options to delete a cluster after a specified cluster idle period, at a specified future time, or after a specified time period. |
| **Automatic or manual configuration** | • Dataproc automatically configures hardware and software but also gives you manual control. |

| | |
|---|---|
| **Initialization actions** | • Run initialization actions to install or customize the settings and libraries you need when your cluster is created. |
| **Optional components** | • Use optional components to install and configure additional components on the cluster.<br>• Optional components are integrated with Dataproc components and offer fully configured environments for Zeppelin, Presto, and other open source software components related to the Apache Hadoop and Apache Spark ecosystem. |
| **Custom containers and images** | • Dataproc serverless Spark can be provisioned with custom docker containers.<br>• Dataproc clusters can be provisioned with a custom image that includes your pre-installed Linux operating system packages. |
| **Component Gateway and notebook access** | • Dataproc Component Gateway enables secure, one-click access to Dataproc default and optional component web interfaces running on the cluster. |
| **Workflow templates** | • Dataproc workflow templates provide a flexible and easy-to-use mechanism for managing and executing workflows.<br>• A workflow template is a reusable workflow configuration that defines a graph of jobs with information on where to run those jobs. |

| | |
|---|---|
| **Automated policy management** | • Standardize security, cost, and infrastructure policies across a fleet of clusters. You can create policies for resource management, security, or network at a project level. |
| **Smart alerts** | • Dataproc recommended alerts allow customers to adjust the thresholds for the pre-configured alerts to get alerts on idle, runaway clusters, jobs, overutilized clusters and more. |
| **Dataproc metastore** | • Fully managed, highly available Hive Metastore (HMS) with fine-grained access control and integration with BigQuery metastore, Dataplex, and Data Catalog. |

# Why use Dataproc?

When compared to traditional, on-premises products and competing cloud services, Dataproc has a number of unique advantages for clusters of three to hundreds of nodes:

- **Low cost**
  - Dataproc is priced at only 1 cent per virtual CPU in your cluster per hour, on top of the other Cloud Platform resources you use.
  - Dataproc clusters can include preemptible instances that have lower compute prices, reducing your costs even further.
  - Instead of rounding your usage up to the nearest hour, Dataproc charges you only for what you really use with second-by-second billing and a low, one-minute-minimum billing period.
- **Super fast**
  - Without using Dataproc, it can take from five to 30 minutes to create Spark and Hadoop clusters on-premises or through IaaS providers.

- Dataproc clusters are quick to start, scale, and shutdown, with each of these operations taking 90 seconds or less, on average.

- **Integrated**
  - Dataproc has built-in integration with other Google Cloud Platform services, such as BigQuery, Cloud Storage, Cloud Bigtable, Cloud Logging, and Cloud Monitoring, so you have more than just a Spark or Hadoop cluster—you have a complete data platform.
- **Managed**
  - Use Spark and Hadoop clusters without the assistance of an administrator or special software.
  - You can easily interact with clusters and Spark or Hadoop jobs through the Google Cloud console, the Cloud SDK, or the Dataproc REST API.
  - When done with a cluster, you can simply turn it off, so you don't spend money on an idle cluster.
- **Simple and familiar**
  - You don't need to learn new tools or APIs to use Dataproc, making it easy to move existing projects into Dataproc without redevelopment.
  - Spark, Hadoop, Pig, and Hive are frequently updated, so you can be productive faster.

# Workflow Templates in Dataproc

There are different workflow templates embedded within Dataproc for users to execute different jobs in a feasible manner. The different kinds are:

- **Managed Cluster**
  - The managed cluster workflow template allows you to create a short-duration cluster for running the desired or set jobs.
  - Then you can easily delete the cluster once the workflow is over.
- **Cluster Selector**
  - This workflow template specifies any of the existing clusters upon which the workflow jobs can run after specifying the user labels.
  - The workflow then intends to run over clusters that match with all of the other specified labels.

- In case there are multiple clusters that match the labels within this workflow execution, then Dataproc will be selecting the one that has the most available YARN memory for running the workflow jobs.
- And at the end of workflow job completion, the cluster is not deleted.

- **Inline**
  - This workflow template type intends to instantiate the workflows with the use of gcloud command.
  - You can make use of YAML files or call the Instantiate Inline API of Dataproc for the same.
  - Inline workflows do not have the ability to create or modify the workflow template resources

- **Parameterized**
  - This workflow template allows you to execute different values over it multiple times.
  - And in the process, you can avoid editing the template again and again for multiple executions by setting up the parameters within that template.
  - And using that parameter, you can intend to pass different values to the template for every run.

# Cloud Dataproc pricing

- Dataproc pricing is based on the number of vCPU and the duration of time that they run.
- There is a specific pricing formula for evaluating the billing amount for the use of Dataproc. The formula is as follows:
  - **$0.016 * # of vCPUs * hourly duration**
- The pricing formula calculates the amount in the hourly rate, but Dataproc can also be billed as per seconds, and the increments are always billed in 1 second cock time. Hence, the minimum billing time is 1-minute.

# Best Practices

- **Workflow Scheduling**
  - The workflow templates, as discussed, offer a flexible and easy mechanism for managing or executing the workflow jobs.
  - These are like reusable configurations for executing workflows! And they usually have graphs of all of the jobs that are about to be executed.
- **Using the Custom Images at the Right Instance**
  - When you are making use of image versions for bundling the Big Data components and operating systems, then custom images come into play. They are used for provisioning the Dataproc clusters.
  - The image versions can be used for merging OS, Google Cloud connectors, and the Big Data components to form the unity package. This complete package is then deployed onto your cluster, as a whole, without breaking it apart.
  - Therefore, in case you have certain dependencies, such as Python libraries, that you intend to transfer onto the cluster, then you should make use of custom images.
- **Gaining Control over Initialization Actions**
  - One of the best practices of Google Cloud Dataproc is to gain control over the initialization actions.
  - These actions intend to allow customization of Cloud Dataproc with specific implementations.
- **Stay Updated on Dataproc Release Notes**
  - Cloud Dataproc publishes weekly release notes that correspond to each change made to Cloud Dataproc.
- **Be Specific About Dataproc Cluster Image Versions**
  - Dataproc image versions are an important part about how the service works. Cloud Dataproc uses images to merge Google Cloud Platform connectors, Apache Spark, and Apache Hadoop components into a single package that can be deployed in a Dataproc cluster.

- Specifying an image version when creating clusters is a key best practice because it associates cluster creation steps with a specific Cloud Dataproc version in the production environment.

## Limitations of DataProc

- No choice of selecting a specific version of Hadoop/hive/spark stack
- You cannot pause/stop Data Proc Cluster
- No UI for managing cluster-specific configuration like Ambari/Cloudera Manager