This guide describes how to move your Apache Hadoop jobs to Google Cloud (Google Cloud) by using Dataproc.

This is the third of three guides describing how to move from on-premises Hadoop:

- Migrating On-Premises Hadoop Infrastructure to Google Cloud (/solutions/migration/hadoop/hadoop-gcp-migration-overview) provides an overview of the migration process, with particular emphasis on moving from large, persistent clusters to an ephemeral model.
- Migrating HDFS Data from On-Premises to Google Cloud (/solutions/migration/hadoop/hadoop-gcp-migration-data) describes the process of moving your data to Cloud Storage and other Google Cloud products.
- This guide, focused on moving your Hadoop jobs to Dataproc.

You can use Dataproc to run most of your Hadoop jobs on Google Cloud. The following list summarizes the basic procedure:

1. Update your job to point to your persistent data stored in Cloud Storage.
2. Create a Dataproc cluster on which to run your job. This kind of temporary, single-use cluster is called an *ephemeral* cluster.
3. Submit your job to the ephemeral cluster.
4. Optionally, monitor your job logs using Stackdriver Logging or Cloud Storage. Logs are captured in Cloud Storage by default, using the staging bucket that you specify when you create the cluster.
5. Check your job's output on Cloud Storage.
6. When your job completes, delete the cluster.

Dataproc runs Hadoop, so many kinds of jobs are supported automatically. When you create a cluster with Dataproc, the following technologies are configured by default:

- Hadoop
- Spark
- Hive
- Pig

Dataproc provides several versions of machine images (/dataproc/docs/concepts/versioning/dataproc-versions) with different versions of open source software preinstalled. You can run many jobs with just the preconfigured software on an image. For some jobs, you might need to install other packages. Dataproc provides a mechanism called initialization actions (/dataproc/docs/concepts/configuring-clusters/init-actions), which enables you to customize the software running on the nodes of your cluster. You can use initialization actions to create scripts that run on every node when it is created.

Cloud Storage connector (/dataproc/docs/concepts/connectors/cloud-storage), which is preinstalled on Dataproc cluster nodes, enables your jobs to use Cloud Storage as a Hadoop compatible file system (HCFS). Store your data in Cloud Storage so that you can take advantage of the connector. If you do, the only necessary change to your jobs is to update the URIs, replacing `hdfs://` with `gs://`.

If you reorganize your data as part of your migration, note all source and destination paths so that you can easily update your jobs to work with the new data organization.

It's possible to store your data in HDFS in persistent clusters in the cloud, but this isn't recommended. You can learn more about moving your data in the data migration guide (/solutions/migration/hadoop/hadoop-gcp-migration-data).

In the recommended approach to running your jobs on Google Cloud, you create ephemeral clusters when you need them and delete them when your jobs are finished. This approach gives you a lot of flexibility in how you configure your clusters. You can use a different configuration for each job, or create several standard cluster configurations that serve groups of jobs.

You can find the basic steps for creating clusters (/dataproc/docs/guides/create-cluster) in the Dataproc documentation. The rest of this section describes some of the important cluster configuration considerations to help you decide how to proceed.

The first thing you need to do to define a new cluster is decide what virtual hardware to use for it. It can be difficult to calculate the perfect cluster configuration, because each job has its particular needs and idiosyncrasies. Experiment with different configurations to find the right setup for your job.

When you set up a cluster, you need to determine at a minimum:

- How many nodes to use.
- The type of virtual machine to use for your master node.
- The type of virtual machine to use for your worker nodes.

Node types are defined by the number of virtual CPUs and the amount of memory they have available. The definitions correspond to the Compute Engine machine types (/compute/docs/machine-types). You can usually find a node type that corresponds to the configuration of on-premises nodes that you are migrating from. You can use that equivalency as a starting place, setting up a cluster that's similar to your on-premises cluster. From there, the best approach is to adjust the configuration and monitor the effect on job execution. As you begin to optimize the configuration of your jobs, you'll start to get a feel for how to approach additional jobs in your system.

Keep in mind that you can scale your cluster as needed, so you don't need to have the perfect specification defined from the start.

You can specify the size of the primary disk used by your worker nodes. The right options for a cluster depend on the types of jobs you're going to run on it. Use the default value and evaluate the results unless you know that your jobs have unusual demands on primary disk usage.

If your job is disk-intensive and is executing slowly on individual nodes, you can add more primary disk space. For particularly disk-intensive jobs, especially those with many individual read and write operations, you might be able to improve operation by adding local SSDs. Add enough SSDs to contain all of the space you need for local execution. Your local execution directories are spread across however many SSDs you add.

You can gain low-cost processing power for your jobs by adding preemptible worker nodes to your cluster. These nodes use preemptible virtual machines (/compute/docs/instances/preemptible) .

Consider the inherent unreliability of preemptible nodes before choosing to use them. Dataproc attempts to smoothly handle preemption, but jobs might fail if they lose too many nodes. Only use preemptible nodes for jobs that are fault-tolerant or that are low enough priority that occasional job failure won't disrupt your business.

If you decide to use preemptible worker nodes, consider the ratio of regular nodes to preemptible nodes. There is no universal formula to get the best results, but in general, the more preemptible nodes you use relative to standard nodes, the higher the chances are that the job won't have enough nodes to complete the task. You can determine the best ratio of preemptible to regular nodes for a job by experimenting with different ratios and analyzing the results.

Note that SSDs are not available on preemptible worker nodes. If you use SSDs on your dedicated nodes, any preemptible worker nodes that you use will match every other aspect of the dedicated nodes, but will have no SSDs available.

Dataproc provides multiple interfaces you can use to launch your jobs, all of which are described in the product documentation (/dataproc/docs/guides/submit-job). This section describes options and operations to consider when running your Hadoop jobs on Google Cloud.

Jobs you run on Dataproc usually have several types of output. Your job might write many kinds of output directly—for example, to files in a Cloud Storage bucket or to another cloud product, like BigQuery. Dataproc also collects logs and console output and puts them in the Cloud Storage staging bucket associated with the cluster you run the job on.

When you submit a job, you can configure it to automatically restart (/dataproc/docs/concepts/jobs/restartable-jobs) if it encounters issues. This option is useful for jobs that rely on resources or circumstances that are highly variable. For example, jobs that stream data across potentially unreliable channels (such as the public internet) are especially prone to random failure due to timeout errors and similar networking issues. Run jobs as restartable if you can imagine situations where the job would fail but would successfully run a short time later.

Dataproc makes it easy to add or remove nodes for your cluster at any time, including while your job is running. The Dataproc documentation includes detailed instructions for scaling your cluster (/dataproc/docs/concepts/configuring-clusters/scaling-clusters). Scaling includes the option for gracefully decommissioning nodes. With this option, nodes that are going to be deleted are given time to complete in-progress processing.

Dealing with individual jobs isn't usually complex, but a Hadoop system can include dozens or hundreds of jobs. Over time, the number of logs, output files, and other information associated with each job proliferates, which can make it difficult to find any individual piece of information. Here are some things that you can do to make it easier to manage your jobs for the future:

- Use custom labels (/dataproc/docs/concepts/labels) to identify jobs, clusters, and other resources. Using labels makes it easy to use a filter to find resources later. Dataproc

supports custom labels using the standard Google Cloud label system, so when you label a resource it can help you manage that resource in other GCP services.

- Organize your Cloud Storage buckets to keep different types of jobs separate. Grouping your data into buckets that correspond to your business structure or functional areas can also make it easier to manage permissions.
- Define clusters for individual jobs or for closely related groups of jobs. It is much easier to update the setup for your ephemeral clusters if you use each configuration only for well-scoped jobs.

- Check out the other parts of the Hadoop migration guide:
  - Overview (/solutions/migration/hadoop/hadoop-gcp-migration-overview)
  - Data migration guide (/solutions/migration/hadoop/hadoop-gcp-migration-data)
- Try out other Google Cloud features for yourself. Have a look at our tutorials (/docs/tutorials).