DataProc Overview:

* Managed Service for Running Hadoop & Spark jobs.
* Why Managed? Takes away the behind the scene toils.
* Cloud Dataproc from provide Managed clusters using parameters you define it creates a Master Node (running the YARN resource Manager and HDFS nodes) and Worker Nodes running (YARN node manager & HDFS data nodes)
* Each node come pre-installed with the recent stable version of Hadoop & Spark as well as Zoo Keeper, Apache Hive, Jupyter , GCS Connector.

Dataproc Benefits:

* Cluster actions complete in approx. 90 secs
* Pay per second – Minimum 1 min.
* Scale up/down – or turn off at will.

Using Dataproc:

* Submit Hadoop/ Spark Jobs – dataproc is interoperable & compatible with these open-source tools.
* Enable autoscaling – if necessary, to cope with the load of the job.
* Output to GCP services – e.g. bq, GCS, bigtable.
* Monitor with Stackdriver- fully integrated logging & monitoring for job performance & output.

Dataproc Resources:

* Cluster Location – Regional, Global (no tied to any specific region)
* Cluster Type – single node cluster, standard clusters (Master VM + 2 or more worker noders) – You can also add preemptible VMS
* High Availability Cluster – 3 Masters.

Submitting Jobs:

* Gcloud CLI
* GCP Consoles
* Dataproc API
* SSH to Master Node

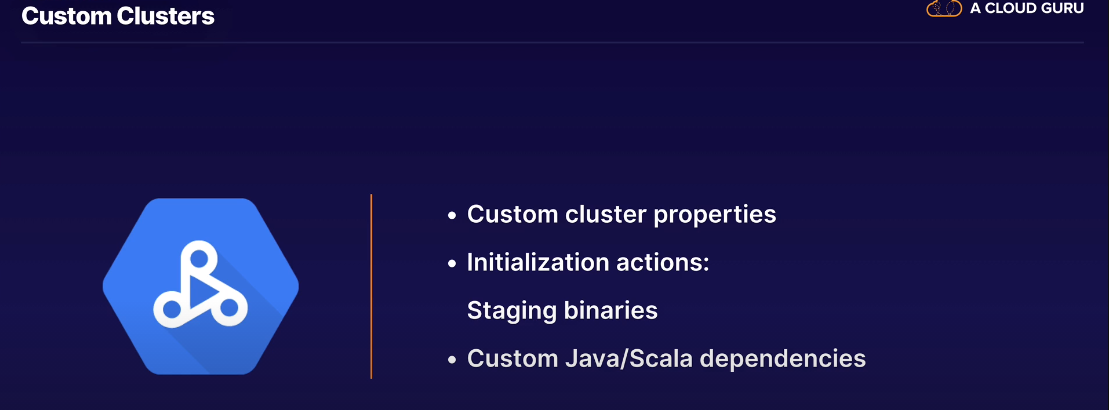
Jobs are created in a pending state, then moved to a running state, then Done state.

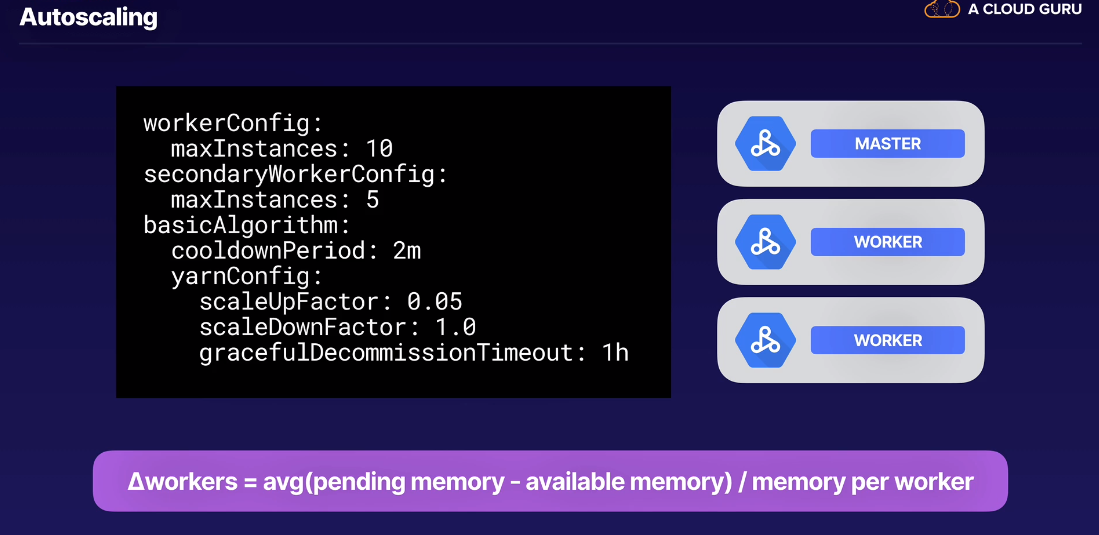
Monitoring & Logging: Use stackdriver monitoring to monitor cluster:

* cluster/yarn/allocated\_memory\_percentage
* cluster/hdfs/storage\_utilization
* cluster/hdfs/unhealthy\_blocks

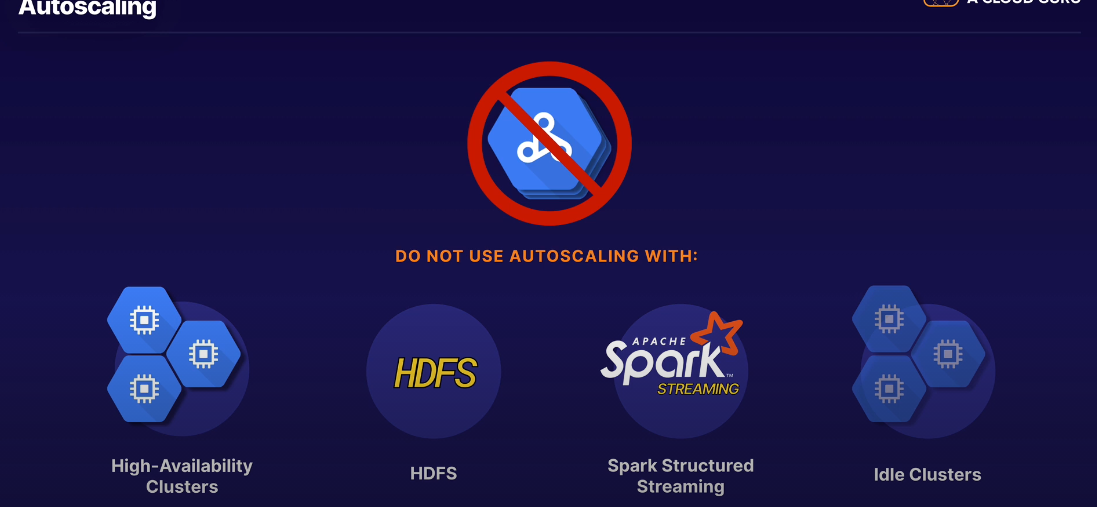
Advanced Dataproc:

* Dataproc node is preconfigured VMs built from Google Image that contains Hadoop, Spark etc.
* If you need completely different packages, you can create your own custom image that will be used in your cluster instead of the default image.
* Google provides simple script that will create a compute engine VM based on the default dataproc image then apply a customization script that you’ve written & finally store the image of this new VM in GCS ready to be used in your Dataproc cluster.
* There are a few other ways you can customize a dataproc cluster without having to completely rebuild the image it’s running on
* Open-Source components of Dataproc such as Hadoop & Spark ship with numerous config files.
* With Custome Cluster properties you can apply changes to any of the values in dozens of config files when setting up your cluster.
* You can also leave everything at default & add custom initialization scripts. - commonly used for staging binaries or any binary that maybe required for a job.
* Another feature, specifically for Spark jobs -You can specify Custom Java/Scala dependencies – saves you from precompiling these dependencies. For your entire cluster when they’re only going to be needed for specific Spark jobs.





Autoscaling policies are written in YAML



* GCS connector is installed by default. We just have to use it when submitting our jobs.

Exam Tips:

* Know when to choose Dataproc – dataproc is a great choice for quickly migrating Hadoop & Spark workloads into GCP.
* Understand the benefits of Dataproc over self-managed Hadoop or Spark cluster – ease of scaling ,being able to use Cloud Storage instead of HDFS, and the connectors to other GCP services including bq & bigtable.
* Familiarize yourself with the cluster options – understand pros & cons of the different cluster options; standard vs high availability, autoscaling & ephemeral, Look for clues in the questions.
* Get to know the open-source ecosystem – Hadoop, Spark, Zookeepr, Hive, Tez and Jupyter.
* Go with the flow – Cloud Dataflow is sometimes the preferred product for ingesting big data, in particular for streaming workloads, so you can’t always assume Cloud Dataproc is the right choice. Cloud Dataflow implements the Apache Beam SDK.

