

Hi, this is Doug Rehnstrom. Welcome to my demo, bulk loading Cloud Datastore data from Google Cloud Storage. In this demo, we will show you how you can use Apache Beam to perform transformations from CSP data in Google Cloud storage to entities in Cloud Datastore.

We will, first, create a Compute Engine instance and install the software needed to run Apache Beam. Then we'll review a simple Apache Beam pipeline that reads data from storage and saves data to Cloud Datastore.

Finally, we will verify our program worked by acquiring the Datastore entities.

Notice, I'm in a project, app-dev-demos. An App Engine application was already created for this project. Now, we'll have to create the Compute Engine virtual machine which will configure to run our Apache Beam application. Let's go into Compute Engine. We're going to create a pretty standard machine. We'll call it apache-beam-demo. And we'll put it in us-central1-a. We'll accept all the rest of the defaults except we'll allow full access to all the Google Cloud APIs and we'll hit Create. The machine is ready now. Let's SSH into it and we'll install some software. We're going to use apt-get to install some software, but first, we'll update packages.

$sudo apt-get update

|  |
| --- |
| $sudo apt-get install \  Openjdk-8-jdk \  maven \  python-setuptools \  python-pip \  git |
| $python –version  $pip –version |
| $pip install apache-beam  $pip install apache-beam[gcp] |
| $echo hell world hell > test.txt  $python –m apache\_beam.examples.wordcount –input test.txt counts  $cat counts-00000-of-00001 |

While that's running, let's take a look at the first command we're going to need to run. So we're going to run this, sudo apt-get install, and we're going to install the Java 8-jdk, Maven, we'll install Python and Pip, and then finally, we'll install Git. Let's copy that to the clipboard and go back to our machine and run it. When that's done, we're going to check our Python versions. We need Python 2.7 for our Apache Beam code to work. We'll render these commands to verify that is correct. Okay, looks good. We have a python version 2.7 installed. We also have Pip installed. Now, apt-get, that's a package manager for Linux, Pip is a package manager for Python. We're going to use Pip in the following two commands to install Apache Beam for our GCP application.

Okay.

Now our application has all the software we need to run our Apache Beam project. Let's do that next. Before we go any further, let's just test the Apache Beam and make sure it was installed correctly. First, I'll enter this command. All it will do is write the words "hello world" and "hello" to a file called test.txt. Just to verify that it worked, let's just take a look at test.txt. So that worked. This command runs a sample Apache Beam word count project. We're going to point it at test.txt and it should count the words in that file. So when it's finished, we should have the word "hello" twice and "world" once. If we type ls, we'll see if file was created, counts zero of one. Let's take a look at that file. And apparently, our program worked. We see the word "world" was in our file once and the word "hello" was in our file twice.

Now, let's run a pipeline to move some data from Cloud Storage to Google Cloud Datastore. The code has already been written for us. I'll clone this repository to go get it.

$git clone <https://github.com/GoogleCloudPlatform/traing-data-analyst>

$ cd ~/training-data-analyst/courses/developingapps/demos/gs2ds/

$cat usa\_presidents.csv

$cat gs2ds.py

$gsutil mb gs://apache-beam-demo-dar/

$gsutil cp usa\_presidents.csv gs://apache-beam-demo-dar/

$nano gs2ds.py

Here, we'll cd into the folder that contains our demo code. From there, let's take a look at the CSV file that contains the data we're going to upload into Datastore. Notice, the file contains the name of all the presidents of the United States who have completed their terms at this point. Let's take a look at the program. This is our Apache Beam pipeline code. Notice, we're going to start off by reading from a Cloud Storage bucket which will create and upload the CSV file into it. For each entity that's in the CSV file, we're going to have to parse it, create a Google Cloud Datastore entity from each line of data and then, eventually, we'll write that data with the last line here into Google Cloud Datastore. Let's make a bucket. Now, let's copy that CSV file we looked at a moment ago into the Google Cloud Storage bucket. We have to make a couple of edits to our Python program. Let's do that using Nano. The first thing we'll need to change is the name of our project. That value is located right here. Our project name is app-dev-demos. We also have to change the name of our Cloud Storage bucket. Our bucket name is apache-beam-demo-dar. Okay, we should be good. Now, let's try running our program.

To see if our program worked, we'll go on to Google Cloud Datastore and see if we have our 44 presidents. Back in the management console, we'll go to Products & Services, select Datastore. We have presidents. And there, we see we have the data that we just imported. Excellent. We executed our pipeline locally and it ran really fast.

We could have also used Google Cloud Dataflow to execute the pipeline. Running in Dataflow would take several minutes because a number of virtual machines would have to be started before the job would run. However, with Cloud Dataflow, a very large job that processes millions or billions of records can be executed in parallel on these machines. In this demo, we showed how easy it is to install Apache Beam and execute pipeline code that moves data into Google Cloud Datastore. Thanks for watching. This is Doug Rehnstrom.