

Show Submission Credentials

P4. Zeppelin for Apache Spark Zeppelin

42 days 23 hours left

✓ Introduction

✓ Installing Zeppelin on a Spark Cluster

✓ Zeppelin on Azure

✓ Zeppelin on GCP

✓ Zeppelin Hello World

✓ Developing ETL Pipelines in Zeppelin

✓ Data Visualization in Zeppelin

✓ Summary

Introduction

Introduction

You have practiced data exploratory analysis using Jupyter Notebook. In this module, we will introduce another interactive notebook, Apache Zeppelin. Zeppelin enables interactive data analytics much like Jupyter Notebook, with built-in support for Apache Spark. In addition, Zeppelin provides a lot of useful features, such as:

1. Showing the progress of a current Spark job.
2. Managing the Spark session for you
3. Letting you write and evaluate blocks of Spark code so that you can prototype and test your Extract-Transform-Load (ETL) pipeline in a notebook without the hassle of using `spark-submit`

Zeppelin also supports Spark SQL and data visualization. You can write a SQL query against Spark DataFrame in Zeppelin, and have the result set displayed in tabular format. You can also plot the distributions of your Spark DataFrame in Zeppelin using the built-in plotting APIs. Programming in Zeppelin notebooks is a great way to get started with Apache Spark. In this primer, you will learn how to deploy a Spark cluster with Apache Zeppelin installed on Azure, GCP, and AWS. You will also explore the interface and a set of built-in features of Zeppelin.

Installing Zeppelin on a Spark Cluster

Installing Zeppelin on a Spark Cluster

Zeppelin should be installed on the master node of a Spark cluster. Zeppelin manages Spark sessions, communicates with the cluster manager and executes user code on the cluster. When you launch a Spark cluster on Azure, Google Cloud Platform or AWS, you can configure the cluster to have Zeppelin installed. This section shows you how to deploy a Spark cluster with Zeppelin installed on GCP and Azure.

Danger

Deleteing Resources

Don't forget delete the cluster after the primer. Clusters can be expensive if left running for a long time.

Zeppelin on Azure

Zeppelin on Azure

Set up the project

1. Make sure you have an active subscription by checking the Azure portal (https://portal.azure.com/#blade/Microsoft_Azure_Billing/SubscriptionsBlade).
2. Registering a resource provider enables your subscription to work with a category of Azure services, for example, Microsoft.Compute for virtual machines and Microsoft.CognitiveServices for Azure Cognitive Services. By default, many resource providers are automatically registered. However, you may need to manually register some resource providers. The scope for registration is always the subscription.

To work with Azure HDInsight with a new subscription, you need to first register the following resource provider:

```
az provider register --namespace Microsoft.HDInsight
```

Confirm that you have registered the necessary provider by running:

```
az hdinsight list-usage --location eastus
# If the available cores (limit) are greater than 20, then
# you should be good to go for the project.
# Since this is asynchronous registration, please wait up to 10 minutes to see
# if the process has been completed.
```

3. In the Azure portal, select **Create a resource** > **Analytics** > **Azure HDInsight**.
4. Under **Basics**, expand **Cluster Type**, and then select **Spark** as the cluster type, and specify the Spark cluster version **Spark 2.4 (HDI 4.0)**. Set other options such as **Cluster name**, **Subscription**, **Resource group**, **Region** and **Login Password**. Click **Next** to continue to the **Storage** page.
5. Under **Storage**, **Select a Storage account**: select **Create new**, and then give a name to the new storage account. The **Default container** has a default name. You can change the name if you want. Under **Tags**, make sure to tag your cluster accordingly. Select **Review + Create** to continue to the **Summary** page.
6. Under **Configuration + pricing** choose the number of worker nodes to use. For this primer 1 worker will do.
7. Under **Tags** make sure to tag your resources with the appropriate tags.
8. On **Summary**, select **Create**. It takes about 20 minutes to create the cluster.
9. Once the cluster has been created, go to the resource in HDInsight, and click on **Zeppelin Notebook** under the Cluster Dashboard.

Overview Get started

Dashboards



Ambari home



Ambari views



Zeppelin notebook



Jupyter notebook



Spark history server



Yarn

Recommended features

Warning

Please give up to 45 minutes to get the HDInsight quota limits assigned. You might need to request a quota CPU limit increase on Azure to deploy your cluster if your core limits are low.

Use Zeppelin notebooks on HDInsight

Please follow the official docs (<https://docs.microsoft.com/en-us/azure/hdinsight/spark/apache-spark-zeppelin-notebook>) provided Azure with examples.

Zeppelin on GCP

Zeppelin on GCP

GCP Dataproc is an IaaS to run Hadoop or Spark clusters. To provision a Spark cluster, you will perform the following steps.

Set up the project

Using the CLI

1. Install Google Cloud SDK (<https://cloud.google.com/sdk/install>).
2. Initialize Google Cloud SDK (<https://cloud.google.com/sdk/install>).
3. Create a GCP project (<https://cloud.google.com/sdk/gcloud/reference/projects/create>). For example, to create a GCP project called `zeppelin-primer`, run the following command.

```
gcloud projects create --name zeppelin-primer
```

Note down the project ID as you will use the ID very often later, please note that project ID is different from the project name which is zeppelin-primer-xxxxxx, for example, zeppelin-primer-123456.

You can see the projects you created via running the following command.

```
gcloud projects list
```

Set the default region, zone, and project:

```
gcloud config set project zeppelin-primer-xxxxxx
gcloud config set compute/region us-east1
gcloud config set compute/zone us-east1-b
```

4. Enable billing for the project (<https://cloud.google.com/billing/docs/how-to/modify-project>).
5. Enable the Cloud Dataproc APIs for the project (https://console.cloud.google.com/flows/enableapi?apiid=dataproc,compute_component&_ga=2.57372330.-1441689332.1550976377&_gac=1.246227504.1550976379.EA1aIQobChMlz_uInq3T4AlVoaGzCh2ybA5eEAAYASAAE).
6. You can create an Apache Spark cluster with Zeppelin pre-configured using the Cloud SDK via the following command.

```
gcloud dataproc clusters create zeppelin-primer --master-machine-type n1-standard-1 --master-boot-disk-size 64 --num-workers 2 --worker-machine-type n1-standard-1 --worker-boot-disk-size 64 --image-version 1.3-deb9 --region us-east1 --initialization-actions 'gs://dataproc-initialization-actions/zeppelin/zeppelin.sh'
```

Using the UI

Alternatively, you can create the cluster using the GCP console. Make sure to create a new project and to enable Billing for that project (steps 1-4 above).

1. In the GCP console, go to the Cloud Dataproc (<https://console.cloud.google.com/dataproc/clusters>) service and click on **Create Cluster**.
2. Configure the cluster (see Figures 1.1, 1.2, 1.3). A minimal cluster with a single master node and two worker nodes is recommended. For the master node, choose 1 vCPU and the default 3.75GB memory. For the worker nodes, we also recommend 1 vCPU and 3.75GB memory. We recommend a 64GB primary disk size for both the master node and the worker nodes.

←

Create a cluster

• Set up cluster
Begin by providing basic information.

• Configure nodes (optional)
Change node compute and storage capabilities.

• Customise cluster (optional)
Add cluster properties, features and actions.

• Manage security (optional)
Change access, encryption and security settings.

CREATE

CANCEL

Equivalent:

REST

command line

Name

Cluster name *
zeppelin-primer

Location

Region *
us-central1

Zone *
us-central1-c

Cluster type

☒ Standard (1 master, N workers)

☐ Single Node (1 master, 0 workers)
Provides one node that acts as both master and worker. Good for proof-of-concept or small-scale processing

☐ High availability (3 masters, N workers)
Hadoop high availability mode provides uninterrupted YARN and HDFS operations despite single-node failures or reboots

Auto-scaling

Automates cluster resource management based on an auto-scaling policy.

Policy

None

Versioning

Use a custom image to load pre-installed packages. [Learn more](#)

Image type and version

2.0-debian10

Release date

First released on 22/1/2021.

CHANGE

Components

Component gateway

☐ Enable component gateway
Provides access to the web interfaces of default and selected optional components on the cluster. [Learn more](#)

Optional components

Select one or multiple components. [Learn more](#)

☐ Anaconda

☐ Hive WebHCat

☐ Jupyter Notebook

☒ Zeppelin Notebook

☐ Druid

☐ Presto

☐ ZooKeeper

☐ Ranger

Figure 1.1: Configure the cluster

https://projects.sailplatform.org/s22-15619/zeppelin

4/11

←

Create a cluster

• Set up cluster

Begin by providing basic information.

• Configure nodes (optional)

Change node compute and storage capabilities.

• Customise cluster (optional)

Add cluster properties, features and actions.

• Manage security (optional)

Change access, encryption and security settings.

CREATE

CANCEL

Equivalent:

REST

command line

Master node

⤴

Contains the YARN Resource Manager, HDFS NameNode and all job drivers.

Machine family

GENERAL-PURPOSE

COMPUTE-OPTIMISED

MEMORY-OPTIMISED

GPU

Machine types for common workloads, optimised for cost and flexibility

Series

N1

▼

Powered by Intel Skylake CPU platform or one of its predecessors

Machine type

Custom

▼

Cores

1

96

1

vCPU

Memory

1

6.5

3.75

GB

☐ Extend Memory

?

✓ CPU PLATFORM AND GPU

Primary disk size (min 10 G...)

64

GB

?

Primary disk type

Standard Persistent Disk

▼

?

Number of local SSDs *

0

▼

x 375GB

?

Figure 1.2: Configure the master node

Worker nodes

Each contains a YARN NodeManager and a HDFS DataNode. HDFS replication factor is 2.

Machine family

GENERAL-PURPOSE

COMPUTE-OPTIMISED

MEMORY-OPTIMISED

GPU

Machine types for common workloads, optimised for cost and flexibility

Series

N1

Powered by Intel Skylake CPU platform or one of its predecessors

Machine type

Custom

Cores

1

96

1

vCPU

Memory

1

6.5

3.75

GB

☐ Extend Memory ?

▼ CPU PLATFORM AND GPU

Number of worker nodes

2

?

Primary disk size (min 10 G...)

64

GB

?

Primary disk type

Standard Persistent Disk

?

Number of local SSDs *

0

▼ x 375GB

?

Secondary worker nodes

▼

Each contains a YARN NodeManager. HDFS does not run on secondary worker nodes. Secondary worker VMs are preemptible by default. A preemptible VM costs less, but lasts only 24 hours and can be terminated at any time due to system demands. [Learn more](#)

Sole tenancy

Enable to create this cluster on sole-tenant nodes. This grants exclusive access to a physical Compute Engine server that is dedicated to hosting only your project's VMs. If you are creating a cluster with an auto-scaling policy, it is recommended that you select a node group that also uses an auto-scaling policy. [Learn more](#)

☐ Enable

Total YARN usage

YARN cores ?

2

YARN memory ?

6 GB

Figure 1.3: Configure the worker nodes

3. Under Customize cluster, navigate to Initialisation Actions .

Initialisation actions

Use initialisation actions to customise settings, install applications or make other modifications to your cluster. Select scripts or executables that Cloud Dataproc will run when provisioning your cluster.

Executable file

bucket/folder/file

BROWSE

+ ADD INITIALISATION ACTION

Figure 1.4: Add initialization action

4. Click on Browse and create a new bucket (see figure 1.5)

Select object

< Buckets ▼

Figure 1.5: Create a new bucket

5. After creating a new bucket copy the Zeppelin installation script provided by GCP, `gs://dataproc-initialization-actions/zeppelin/zeppelin.sh` to this bucket by running the following command.

```
gsutil cp gs://dataproc-initialization-actions/zeppelin/zeppelin.sh gs://<your-bucket-name>
```

Connect to Zeppelin through Web Browser

1. Create an SSH tunnel with dynamic port forwarding to the master node where Zeppelin is hosted.

```
MASTER_HOSTNAME='zeppelin-primer-m'
gcloud compute ssh ${MASTER_HOSTNAME} -- -L 8080:localhost:8080
```

2. Go to `localhost:8080` to access Zeppelin's web interface.

Deleting Resources

You can delete the cluster in the Cloud Dataproc console (<https://console.cloud.google.com/dataproc/clusters>).

Zeppelin Hello World

Below are several examples, each is provided to you as a Zeppelin notebook for you to run and experiment with.

Zeppelin Hello World

In this “hello-world” example, you will be running a worked example that computes friends of friends in a Zeppelin notebook. The code we provide is divided into a few code blocks in a Zeppelin notebook. You can run each code block and monitor its progress. Can you explain why some code blocks take longer to run than other ones? (hint, think about the difference between RDD transformations and actions).

A brief background on the social graph. Friendship can be modeled as an undirected graph. Suppose, A and B are friends, then (A, B) is in the edge set. Now suppose, A is also a friend with C, i.e., (A, C) is in the edge set. But (B, C) is not in the edge set. Then we say that B is a friend of a friend of C since they are not friends yet but they share the common friend A. Friends of friends is a strategy for friend recommendations. For example, a friend recommendation system can recommend C to B since C is not a friend of B but both B and C know A.

Here is the solution provided within a Zeppelin notebook. You can download the notebook via the command:

```
wget "https://clouddeveloper.blob.core.windows.net/assets/iterative-processing/primer/zeppelin-primer/notebooks/Hello World.js
on"
```

You can upload the notebook to Zeppelin via the `Import note` button on Zeppelin's web interface at `localhost:8080` (if you followed the previous steps to set up Zeppelin on GCP).

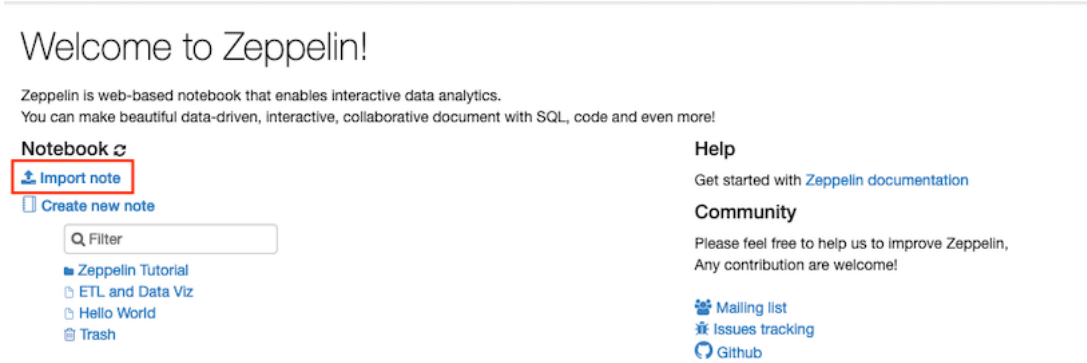


Figure 2.1: Import button

And you will be using a Facebook dataset (<https://snap.stanford.edu/data/egonets-Facebook.html>), which contains tuples (A, B) representing the edges in the social graph. You can run this notebook on the cloud provider of your choice.

Developing ETL Pipelines in Zeppelin

Developing ETL Pipelines in Zeppelin

Real-world datasets are often large-scale, complex and dirty. The schema of a dataset can be convoluted and/or poorly documented so that you have to inspect the dataset and find out its structure by yourself. When you are developing a pipeline for real-world datasets, it will save you a lot of time to run the transformations on the dataset (or a subset of it) step by step and inspect the intermediate results of each stage. Zeppelin

is a great tool for exploring a new dataset and prototyping ETL pipelines. In this section, you will be running ETL of a Yelp restaurant review dataset. We are providing you with a Zeppelin notebook. You can follow the instruction in the notebook and explore it. You can download the notebook with

```
wget "https://clouddeveloper.blob.core.windows.net/assets/iterative-processing/primer/zeppelin-primer/notebooks/ETL and Data Viz.json"
```

You will need to setup up a MySQL database in order to run the Load (L) step in the ETL pipeline for this task. The MySQL database serves as the storage for the output of the ETL process.

Follow the steps below to provision a MySQL database using GCP Cloud SQL and configure the firewall rules.

- 1. Create a MySQL Database using GCP Cloud SQL from the web console.
 - 1. Go to the Cloud SQL (<https://console.cloud.google.com/sql/instances>) service and click on `Create Instance` (see figure 3.1).

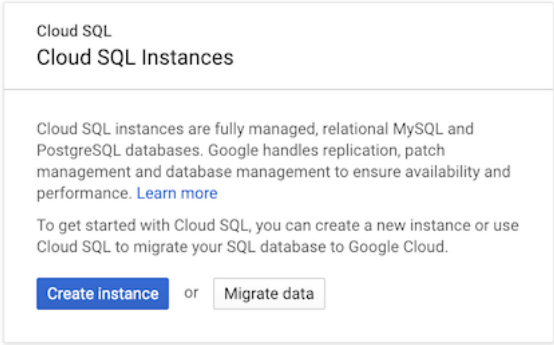


Figure 3.1: Create Cloud SQL instance.

- 2. Select `MySQL` (see figure 3.2).



Figure 3.2: Choose DB Engine

- 3. Set `Instance ID`, `Root password`, and `Location` as follows and click on `Create` (see figure 3.3).

Create a MySQL instance

Instance info

Instance ID *

zeppelin-primer-mysql

Use lowercase letters, numbers and hyphens. Start with a letter.

Password *

.....

GENERATE

☐ No password

Database version *

MySQL 5.7

Choose region and zonal availability

For better performance, keep your data close to the services that need it. Region is permanent, while zone can be changed any time.

Region

us-east1 (South Carolina)

☒ Single zone

In case of outage, no failover. Not recommended for production.

☐ Multiple zones (highly available)

Automatic failover to another zone within your selected region. Recommended for production instances. Increases cost. Enables binary logs (required for replication) and automatic backups. Make sure that your storage can support at least seven days of logs.

Primary zone

us-east1-b

^ HIDE ZONES

Customise your instance

You can also customise instance configurations later

^ SHOW CONFIGURATION OPTIONS

CREATE INSTANCE

CANCEL

Figure 3.3: Set root password

4. Expand Show configuration options , under the Connections dropdown, add a network policy that allows any IP address to connect to the instance (see Figure 3.4).

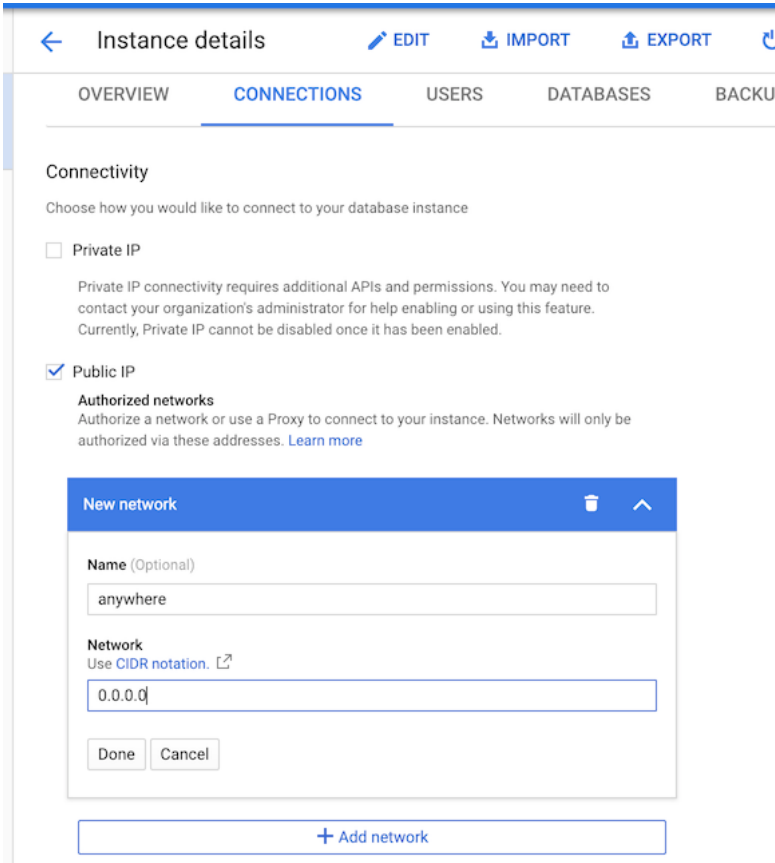


Figure 3.4: Cloud SQL network policy

Once the instance is ready:

1. Go to the instance dashboard, `Users` tab, create a new user called `etl_user` and note down the password.
2. Under the `Databases` tab, create a new database, called `etl_db`.

Now follow the steps in the notebook. Make sure to update the notebook with your database's IP address and password.

Data Visualization in Zeppelin

Data Visualization in Zeppelin

Zeppelin has built-in tools for data visualizations. For example, you can write queries against a Spark Dataframe using SQL, and display the result set in a tabular format. You can also plot histograms and a scatter plot of the result set using Zeppelin’s web interface (see figure 4.1). In this task, you will do some simple data visualization on the dataset obtained in the previous ETL process to examine which features influence the restaurant ratings. We have provided you with worked examples in the Zeppelin notebook to get you started. The notebook is the same as the one in the previous section.

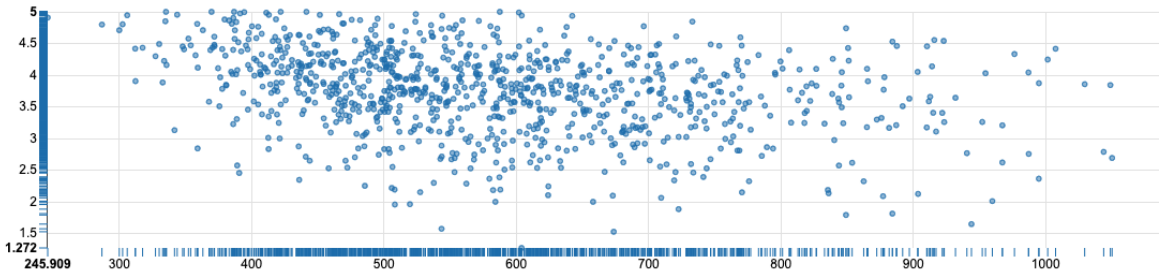


Figure 4.1: Data visualization in Zeppelin

Summary

Summary

In this primer, you have learned about Apache Zeppelin, which is an interactive notebook built for Apache Spark.

You have practiced implementing a sample ETL pipeline on Zeppelin to:

- extract the raw data from cloud storage
- inspect the data schema
- implement data transformation solutions to join datasets
- produce a new dataset for further data analytics, and load the transformed data to a MySQL database
- explore data visualization tools in Zeppelin

You also wrote SQL queries against DataFrames and plotted the distributions of fields of interest.

Zeppelin is a great tool for data exploration and analysis of large datasets, prototyping and debugging Spark ETL pipelines and large-scale machine learning algorithms. You are ready to use Zeppelin notebooks with Apache Spark in future projects.