

4.3.1 安装与配置 Byzer-SQL

Byzer-SQL 安装

全自动化安装

手动安装

Byzer-SQL 控制台安装

修改配置

Byzer-SQL 由两部分构成，第一部分是引擎，第二部分是管理使用界面。这个和 MySQL 非常类似。

Byzer-SQL 安装

我们提供了两种方式进行部署安装。第一种是全自动化安装，第二种是手动安装。

全自动化安装

我们提供了一个脚本，包括控制台也会直接安装。为了保证安装顺利，有一些如下条件要求：

1. 操作系统是 Ubuntu 20.04 或者 22.04
2. 网络保持通畅
3. 如果你的机器是云上的虚拟机，在创建机器资源的时候，一般都会让你设置主机名，建议主机名就叫byzerllm(确保它有sudo权限，一般默认都会有)，会少很多事情，安装脚本也只需要运行一次；
4. 如果是国外的云厂商，有一些默认参数可以调整，现在默认参数都是为国内设置的，在国外的云厂商的主机会慢不少。比如PYPI_MIRROR可以设置为default, GIT_MIRROR可以设置为github

首先，在机器上执行如下指令

```
1  git clone https://gitee.com/allwefantasy/byzer-llm
2  cd byzer-llm/setup-machine
```

接着切换到 Root用户下执行如下脚本：

```
▼ Shell |
1  ROLE=master ./setup-machine.sh
```

此时会完成 byzerllm 账号创建登工作。

接着 切换到 byzerllm 账号下，再次执行：

```
▼ Shell |
1  ROLE=master ./setup-machine.sh
```

会完整的安装包括显卡驱动，cuda 工具集，一整套 Byzer-LLM 环境，之后就可以访问 <http://localhost:9002> 来使用 Byzer-LLM 了。

注意，如果你的机器是从云厂商创建的虚拟机，需要使用SSH隧道端口映射，让本地也能访问远程机器的9002/9003端口，可参考下面的代码，只需把local_port替换成你希望的本地端口，remote_host替换成云主机的公网ip

```
▼ Shell |
1  ssh -L local_port:localhost:9003 byzerllm@remote_host
```

用户如果想组建集群，对于从节点，可以使用如下命令，也是分别执行两次：

```
▼ Shell |
1  ROLE=worker ./setup-machine.sh
```

手动安装

下载下列引擎和控制台：

1. Byzer-lang: <https://download.byzer.org/byzer-lang/2.3.9/byzer-lang-all-in-one-linux->

amd64-3.3.0-2.3.9.tar.gz

2. Byzer-notebook: <https://download.byzer.org/byzer-notebook/1.2.6/Byzer-Notebook-1.2.6.tar.gz>

注意，byzer-lang 区分操作系统，将文件的 linux 字符替换成 darwin/win 来适配你的平台。darwin 为 macos, win 为 windows。

可以访问 <https://download.byzer.org> 查看最新版本

首先解压 byzer-lang-all-in-one-linux-amd64-3.3.0-2.3.9.tar.gz，接着进入解压后的目录 byzer-lang-all-in-one-linux-amd64-3.3.0-2.3.9 目录，目录结构如下所示：

```
byzerllm ~/Softwares/byzer-lang-all-in-one-darwin-amd64-3.3.0-2.3.9 (0.048s)
tree -L 1
.
├── LICENSE
├── README.md
├── RELEASES.md
├── bin
├── conf
├── jdk8
├── libs
├── logs
├── main
├── plugin
└── spark
```

注意，如果需要支持 Arm 芯片，诸如苹果 M 系列，你需要手动下载对应系统的 JDK8 然后替换安装目录里的 `jdk8` 目录即可。

在当前目录执行如下命令即可启动 Byzer 引擎：

```
1  ./bin/byzer.sh start
```

系统会自动做一些检测并且输出一些指令，在我的笔记本上是会是类似如下的信息：

```

1  ./bin/byzer.sh start
2  Starting Byzer engine...
3
4  Byzer-lang is checking installation environment, log is at /Users/allwefantasy/Softwares/byzer-lang-all-in-one-darwin-amd64-3.3.0-2.3.9/logs/check-env.out
5
6  Checking OS
7  ..... [PASS]
8  Checking Java Version
9  ..... [PASS]
10 Checking Ports Availability
11 ..... [PASS]
12
13 Checking environment finished successfully. To check again, run 'bin/check-env.sh' manually.
14
15 SPARK_HOME is: /Users/allwefantasy/Softwares/spark-3.3.1-bin-hadoop3
16 BYZER_HOME is: /Users/allwefantasy/Softwares/byzer-lang-all-in-one-darwin-amd64-3.3.0-2.3.9
17 BYZER_CONFIG_FILE is: /Users/allwefantasy/Softwares/byzer-lang-all-in-one-darwin-amd64-3.3.0-2.3.9/conf/byzer.properties
18 Starting Byzer engine in all-in-one mode...
19
20 [Java Env]
21 JAVA_HOME: /Library/Java/JavaVirtualMachines/jdk1.8.0_151.jdk/Contents/Home/
22 JAVA: /Library/Java/JavaVirtualMachines/jdk1.8.0_151.jdk/Contents/Home/bin/java
23
24 [All Config]
25 -byzer.server.runtime.driver-memory 6g -spark.kryoserializer.buffer 256k -streaming.plugin.clznames tech.mlsq.plugins.ds.MLSQExcelApp,tech.mlsq.plugins.assert.app.MLSQLAssert,tech.mlsq.plugins.shell.app.MLSQLShell,tech.mlsq.plugins.mllib.app.MLSQLMllib -streaming.spark.service true -streaming.job.cancel true -streaming.driver.port 9003 -streaming.platform spark -streaming.name Byzer-lang-desktop -streaming.thrift false -spark.kryoserializer.buffer.max 1024m -spark.sql.hive.thriftServer.singleSession true -spark.scheduler.mode FAIR -byzer.server.mode all-in-one -spark.serializer org.apache.spark.serializer.KryoSerializer -streaming.rest true -streaming.datalake.path ./data/ -byzer.server.dryrun false -streaming.enableHiveSupport false
26
27 Final command:
28 nohup /Library/Java/JavaVirtualMachines/jdk1.8.0_151.jdk/Contents/Home/bin/java -Xmx6g
29

```

```

30 -cp /Users/allwefantasy/Softwares/byzer-lang-all-in-one-darwin-amd64-3.3.0-2.3.9/main/byzer-lang-3.3.0-2.12-2.3.9.jar:/Users/allwefantasy/Softwares/byzer-lang-all-in-one-darwin-amd64-3.3.0-2.3.9/spark/*:/Users/allwefantasy/Softwares/byzer-lang-all-in-one-darwin-amd64-3.3.0-2.3.9/libs/*:/Users/allwefantasy/Softwares/byzer-lang-all-in-one-darwin-amd64-3.3.0-2.3.9/plugin/*
    tech.mlsql.example.app.LocalSparkServiceApp -byzer.server.runtime.driver-memory 6g -spark.kryoserializer.buffer 256k -streaming.plugin.clznames tech.mlsql.plugins.ds.MLSQLExcelApp,tech.mlsql.plugins.assert.app.MLSQLAssert,tech.mlsql.plugins.shell.app.MLSQLShell,tech.mlsql.plugins.mllib.app.MLSQLMllib -streaming.spark.service true -streaming.job.cancel true -streaming.driver.port 9003 -streaming.platform spark -streaming.name Byzer-lang-desktop -streaming.thrift false -spark.kryoserializer.buffer.max 1024m -spark.sql.hive.thriftServer.singleSession true -spark.scheduler.mode FAIR -byzer.server.mode all-in-one -spark.serializer org.apache.spark.serializer.KryoSerializer -streaming.rest true -streaming.datalake.path ./data/ -byzer.server.dryrun false -streaming.enableHiveSupport false >> /Users/allwefantasy/Softwares/byzer-lang-all-in-one-darwin-amd64-3.3.0-2.3.9/logs/byzer.out &
31
32
33 Byzer engine is starting. It may take a while. For status, please visit http://192.168.8.169:9003.
34
    You may also check status via: PID:20428, or Log: /Users/allwefantasy/Softwares/byzer-lang-all-in-one-darwin-amd64-3.3.0-2.3.9/logs/byzer-lang.log.

```

一般只要都是 PASS 就没问题。启动完成后就可以访问打开浏览器访问 <http://127.0.0.1:9003>，此时会出现如下页面：

←

→

🔄

📄 127.0.0.1:9003/#/

1

select 1 as a, 'jack' as b as bbc;

运行

任务列表

a	b
1	jack

表示引擎确实可以正常工作了。

如果你是一个命令行爱好者，那么可以使用 byzer-shell, 通过如下方式启动：

```
1  ./bin/byzer-shell-darwin
```

此时输出如下：

```

1  ./bin/byzer-shell-darwin
2  ✓ 11s.
3
4  _
5  | | _
6  | | _
7  | | _
8  | | _
9
10
11 version: "2.3.9"
12 buildBy: "williamzhu"
13 date: "2023-09-12T12:11Z"
14 srcChecksum: "b891b9d16d66d137603748329dc45115"
15 revision: "87f0e2dc576a4770840862d87bd7e3351a89b8a2"
16 branch: "master"
17 url: "https://github.com/byzer-org/byzer-lang.git"
18 core: "3.3.0"
19
20 Type "CTRL-C" or "CTRL-D" to exit the program.
21
22 >>

```

在这里你就可以直接输入一条 Byzer-SQL 语句验证下：

```
>> select 1 as num as table1;
```

✓ 2s.

num
1

>> 

你也可以加载一个 csv 或者 excel 文件:

```
>> load csv.`/Users/allwefantasy/projects/anything/*.csv` where header="true" as table2;
```

✓ 1s.

content	product-name
hello world	byzer

>> █

如果想退出，按 `ctrl + c` 即可。

接着，让我们继续安装 Web 版本的控制台。

Byzer-SQL 控制台安装

Byzer-SQL 控制台的安装略微复杂些，因为需要一个 MySQL 数据库。目前我们测试过的版本是 MySQL 5.7（比较老的版本了），所以也推荐用户安装该版本的 MySQL。

安装完 MySQL，剩下的事情就比较简单了。首先把前面下载好的 Byzer-Notebook-1.2.6.tar.gz 进行解压，这个时候你会得到如下一个目录：

```
byzerllm ~/Softwares/byzer-notebook (0.043s)
tree -L 1
.
├── CHANGELOG.md
├── VERSION
├── bin
├── commit_SHA1
├── commit_SHA1.frontend
├── conf
├── lib
├── logs
└── sample

5 directories, 4 files
```

注意，解压后的目录名字就叫 byzer-notebook。进入该目录后，还需要做两个动作才能启动控制台。

修改配置

进入 conf 目录下的，打开 notebook.properties 文件，首先要修改的是前面我们安装的 MySQL 的配置：


```
1 notebook.database.type=mysql
2 notebook.database.ip=127.0.0.1
3 notebook.database.port=3306
4 notebook.database.name=notebook
5 notebook.database.username=root
6 notebook.database.password=root
```

接着是配置引擎信息：

```
1 notebook.user.home=/mlsql
2 notebook.url=http://localhost:9002
3 notebook.mlsql.engine-url=http://localhost:9003
4 notebook.mlsql.engine-backup-url=http://localhost:9004
```

`notebook.user.home` 是用来存储引擎临时数据文件的，配置一个你实际存在的路径，并且确保前面的Byzer-SQL 引擎可以访问。`notebook.mlsql.engine-url` 这个配置的引擎地址，也就是前面我们访问的 `http://127.0.0.1:9003` 地址。`notebook.mlsql.engine-backup-url` 是备份引擎地址，保持和前面的 `notebook.mlsql.engine-url` 一致即可。

修改后的样子大概是这样的：

```
1 notebook.user.home=/tmp/byzersql
2 notebook.url=http://localhost:9002
3 notebook.mlsql.engine-url=http://localhost:9003
4 notebook.mlsql.engine-backup-url=http://localhost:9003
```

有了这几项修改后，就可以启动控制台了，重新进入项目安装目录，然后执行如下指令：

```
1 ./bin/notebook.sh start
```

启动结果应该类似这样：

```

1  ./bin/notebook.sh start
2  Starting Byzer Notebook...
3
4  Byzer Notebook is checking installation environment, log is at /Users/allw
  efantasy/Softwares/byzer-notebook/logs/check-env.out
5
6  Checking Java Version
7  ..... [PASS]
8  Checking MySQL Availability & Version
9  ..... [PASS]
10 Checking Byzer engine
11 ..... [PASS]
12 Checking Ports Availability
13 ..... [PASS]
14
15 Checking environment finished successfully. To check again, run 'bin/check
  -env.sh' manually.
16
17
18 NOTEBOOK_HOME is:/Users/allwefantasy/Softwares/byzer-notebook
19 NOTEBOOK_CONFIG_FILE is:/Users/allwefantasy/Softwares/byzer-notebook/conf/
  notebook.properties
20 NOTEBOOK_LOG_FOLDER is: /Users/allwefantasy/Softwares/byzer-notebook/logs
  .
21
22 2024-07-30 18:40:52 Start Byzer Notebook...
23
24 Byzer Notebook is starting. It may take a while. For status, please visit
  http://192.168.8.169:9002.
25
26 You may also check status via: PID:77933, or Log: /Users/allwefantasy/Soft
  wares/byzer-notebook/logs/notebook.log.

```

会做一些列的检测，包括 Java 版本，MySQL 连通性，Byzer引擎的连通性，端口是否被占用。

如果你系统默认配置了 JAVA_HOME 变量，并且版本不是 JDK8,那么你可以使用 Byzer Engine里的 JDK 启动 控制台。

执行方式如下：

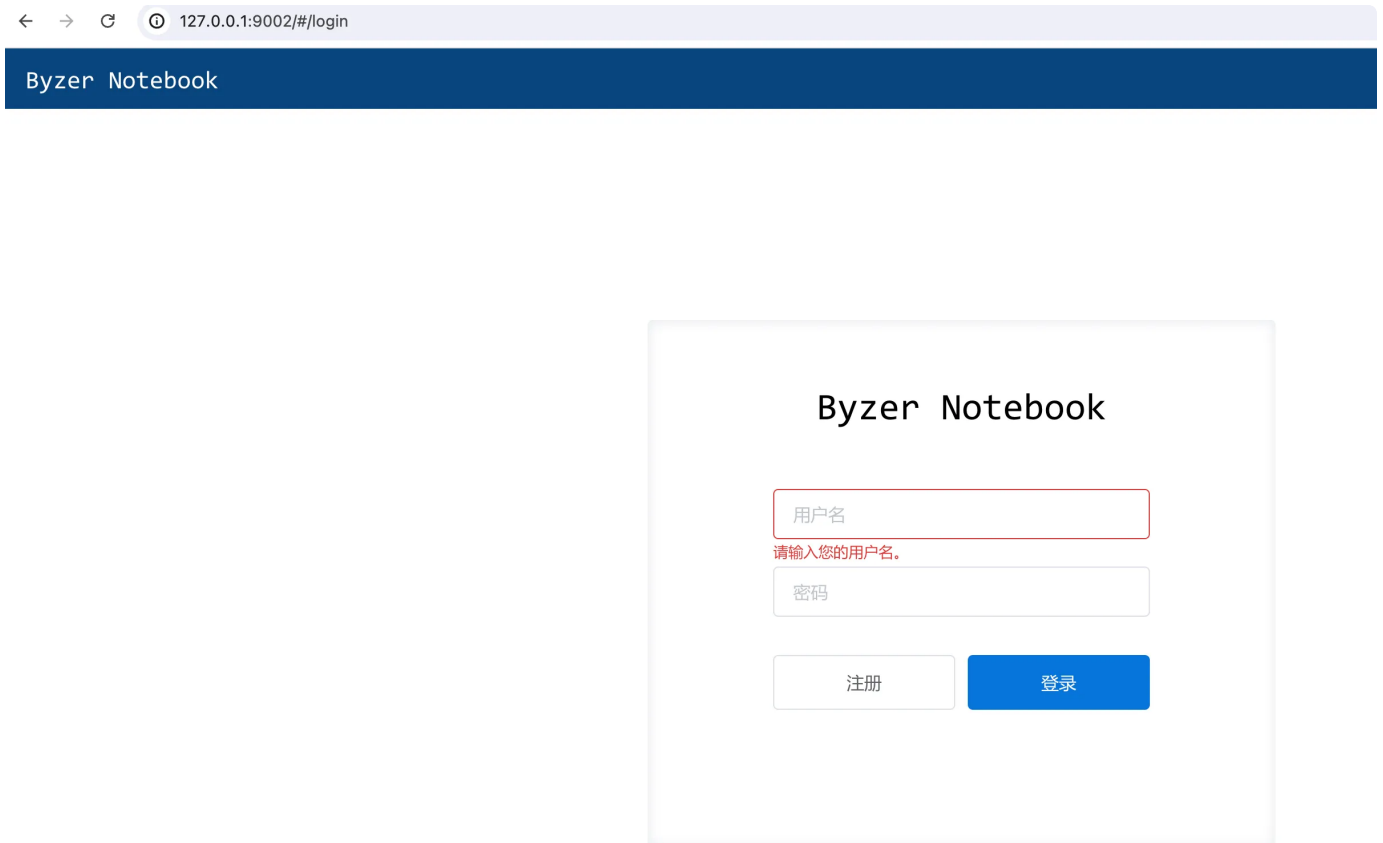
```

1  export JAVA_HOME=/Users/allwefantasy/Softwares/byzer-lang-all-in-one-darwin
  -amd64-3.3.0-2.3.9/jdk8
2  ./bin/notebook.sh start

```

注意，修改 JAVA_HOME 为你的 byzer引擎的实际安装地址。

现在，可以访问 <http://127.0.0.1:9002> 进入控制台：



← → ↻ ⓘ 127.0.0.1:9002/#/login

Byzer Notebook

Byzer Notebook

用户名

请输入您的用户名。

密码

注册 登录

随便输入一个账号和密码，然后点击左侧注册按钮，即可完成账号注册，然后自动进入主界面：

快速开始

开始使用示例笔记本

探索

导入和连接数据

快速导入和连接数据

上传

创建新笔记本

创建笔记本以开始分析、训练和可视化您的数据。

创建