Fastcampus Data Science Extension SCHOOL

noSQL

noSQL

- 확장가능성, 스키마 없는 데이터 모델에 유리
- Row, Document, key-value 등 다양

RDBMS와 다른점

- Schemaless
- Join 불가능(reference 등으로 구현)
- No Transaction
- 수평확장 용이

종류

- {Key:Value} = Redis
- [Column] = Cassandra, HBase
- Document {Key:{Key:Value}} = CouchDB, MongoDB

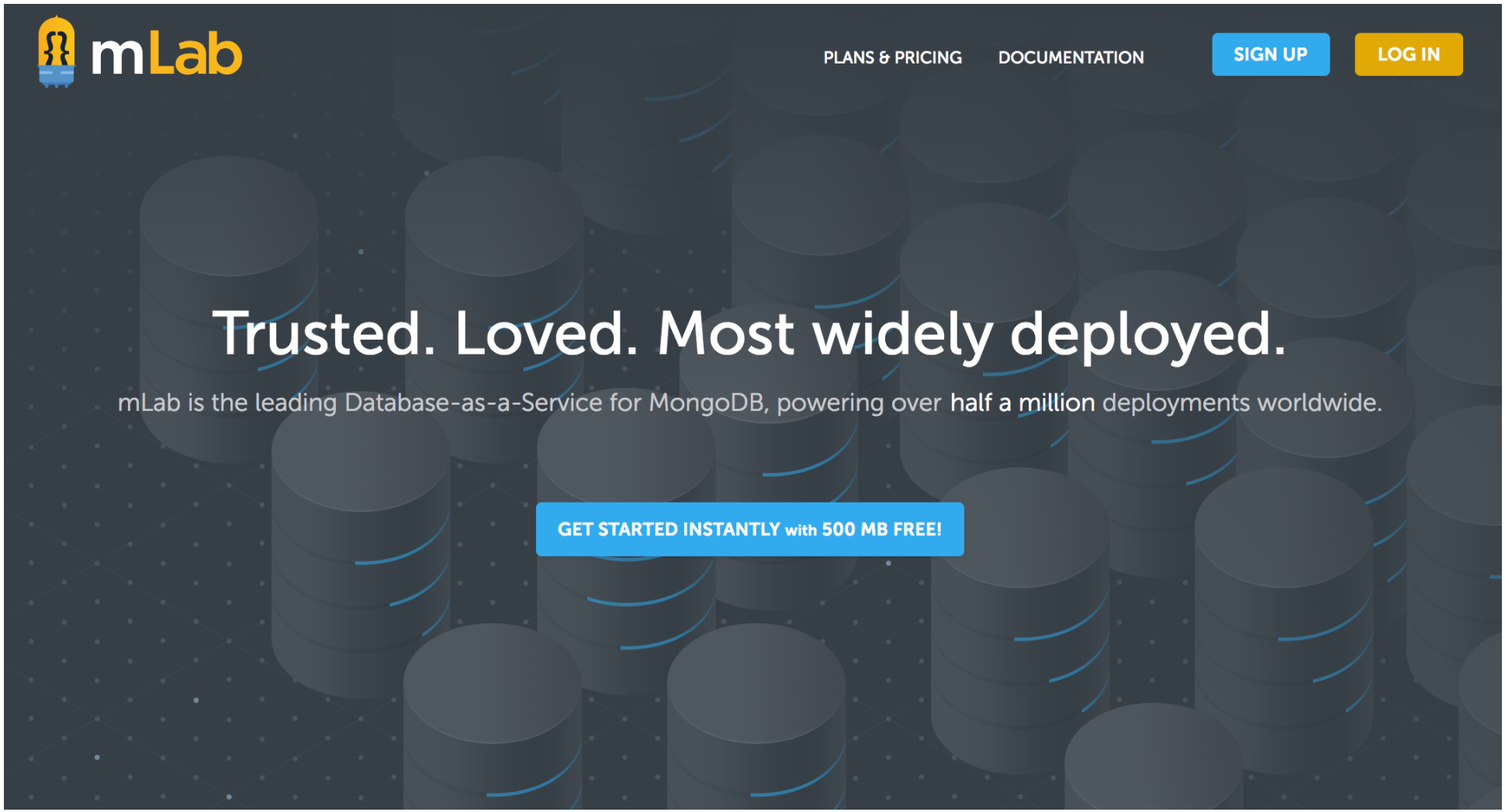
MongoDB

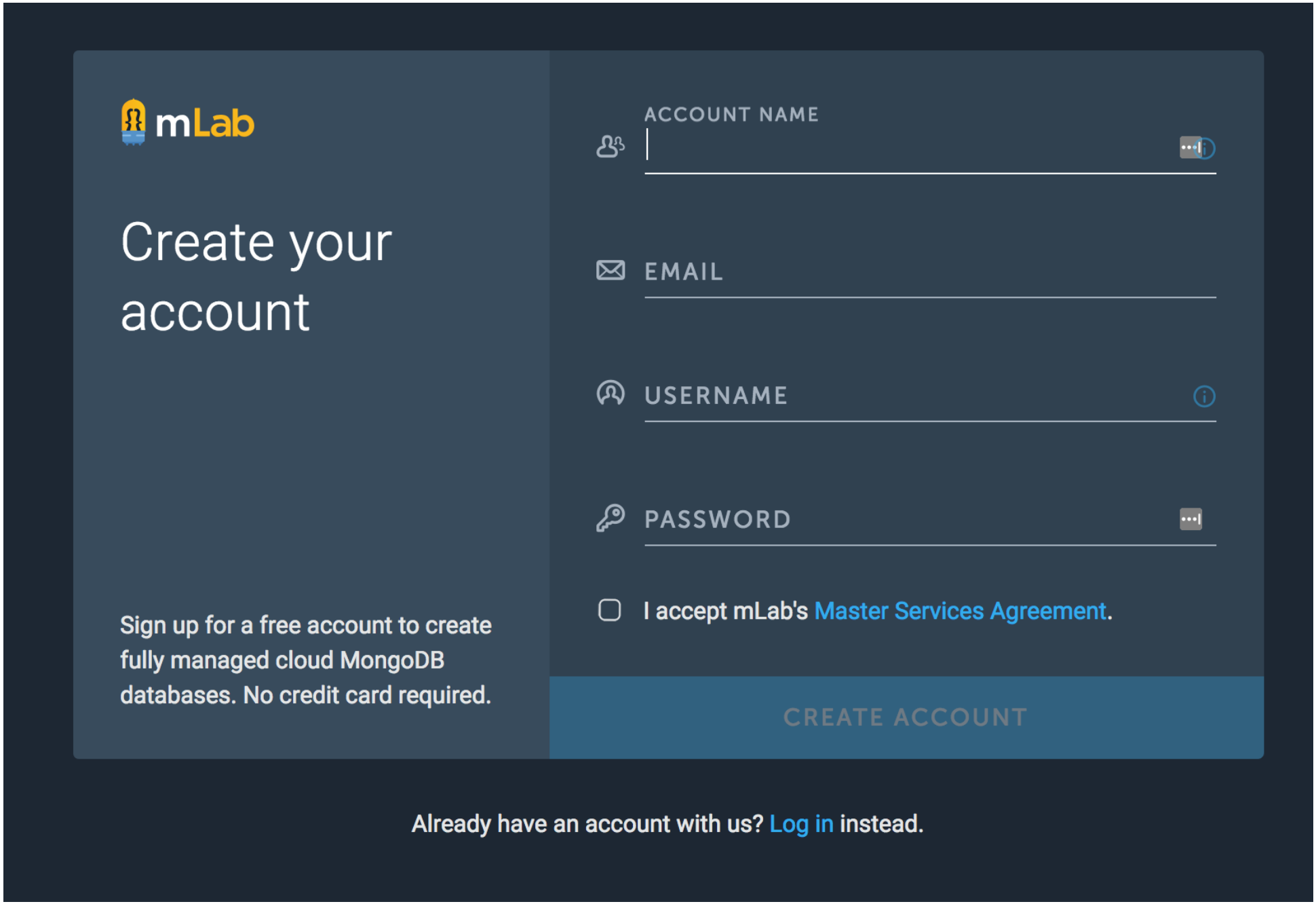
- BSON(Binary JSON) 기반 Key-Value Store
- JSON 형태 문서
- Collection -> Document -> Key:Value Data

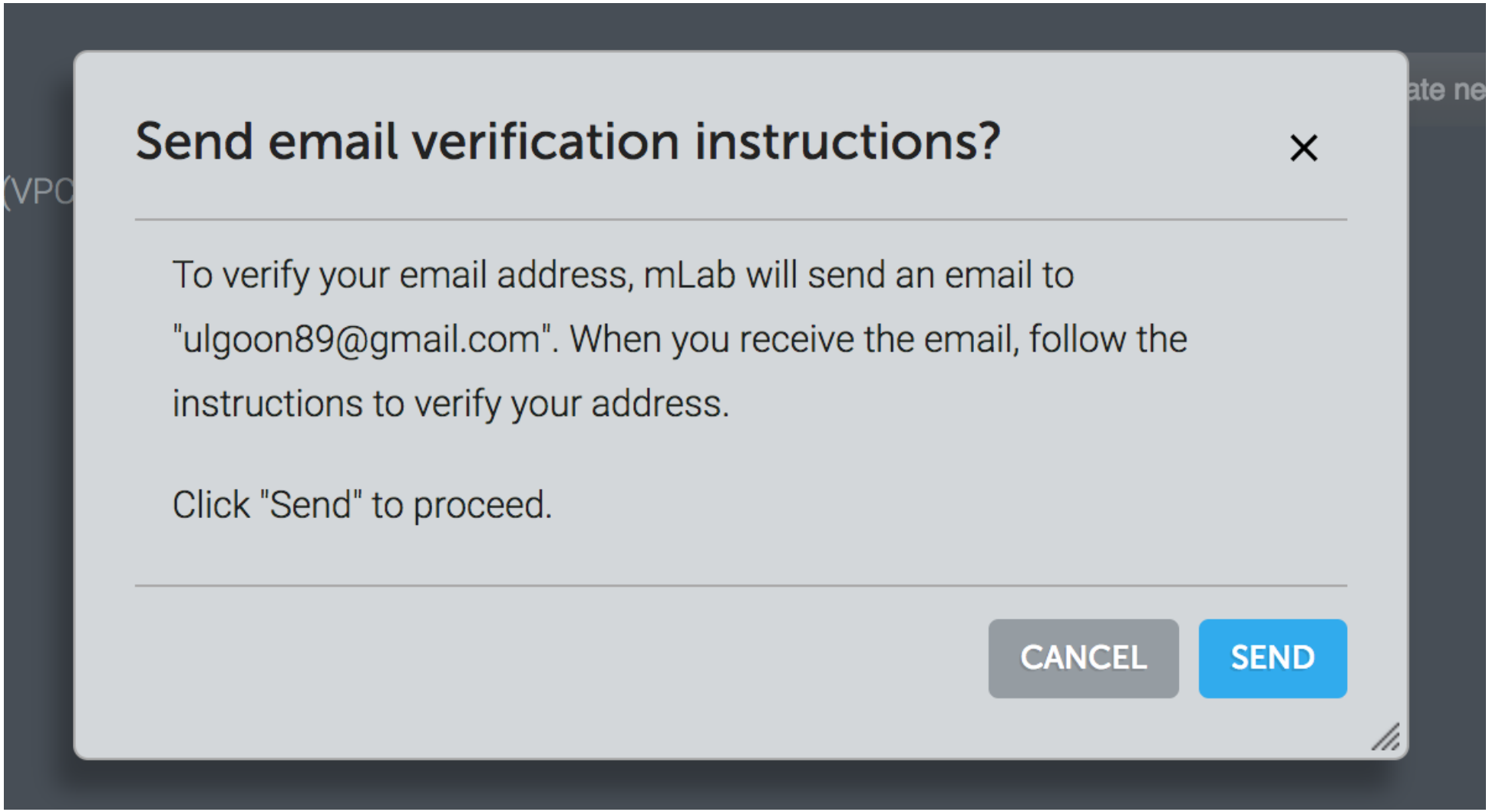
Requirements

- DB instance(mLab)
- pymongo(`$ pip install pymongo`)
- pandas(`$ pip install pandas`)
- requests(`$ pip install requests`)
- jupyter notebook(`$ pip install jupyter`)

Sign up


mLab
PLANS & PRICING
DOCUMENTATION
SIGN UP
LOG IN
Trusted. Loved. Most widely deployed.
mLab is the leading Database-as-a-Service for MongoDB, powering over half a million deployments worldwide.
GET STARTED INSTANTLY with 500 MB FREE!


mLab
Create your account
ACCOUNT NAME
EMAIL
USERNAME
PASSWORD
I accept mLab's Master Services Agreement.
Sign up for a free account to create fully managed cloud MongoDB databases. No credit card required.
CREATE ACCOUNT
Already have an account with us? Log in instead.


Send email verification instructions?
To verify your email address, mLab will send an email to "ulgoon89@gmail.com". When you receive the email, follow the instructions to verify your address.
Click "Send" to proceed.
CANCEL
SEND

mLab to me 1:42

Hello from mLab,

The "ulgoon" user (ulgoon89@gmail.com) associated with the "ulgoon" mLab account has initiated a request to verify this email address, ulgoon89@gmail.com.

To complete the email verification process, click the following link and then log in to your mLab account:

https://mlab.com/verifyemail/AFgwlIF1otIe6yTxzeO3y9OIzN7YDHOPgglCJoQrPHiocXylGkZDJEU-aeV-qoLncrKHll5ePBiui-AqzaeAsQ

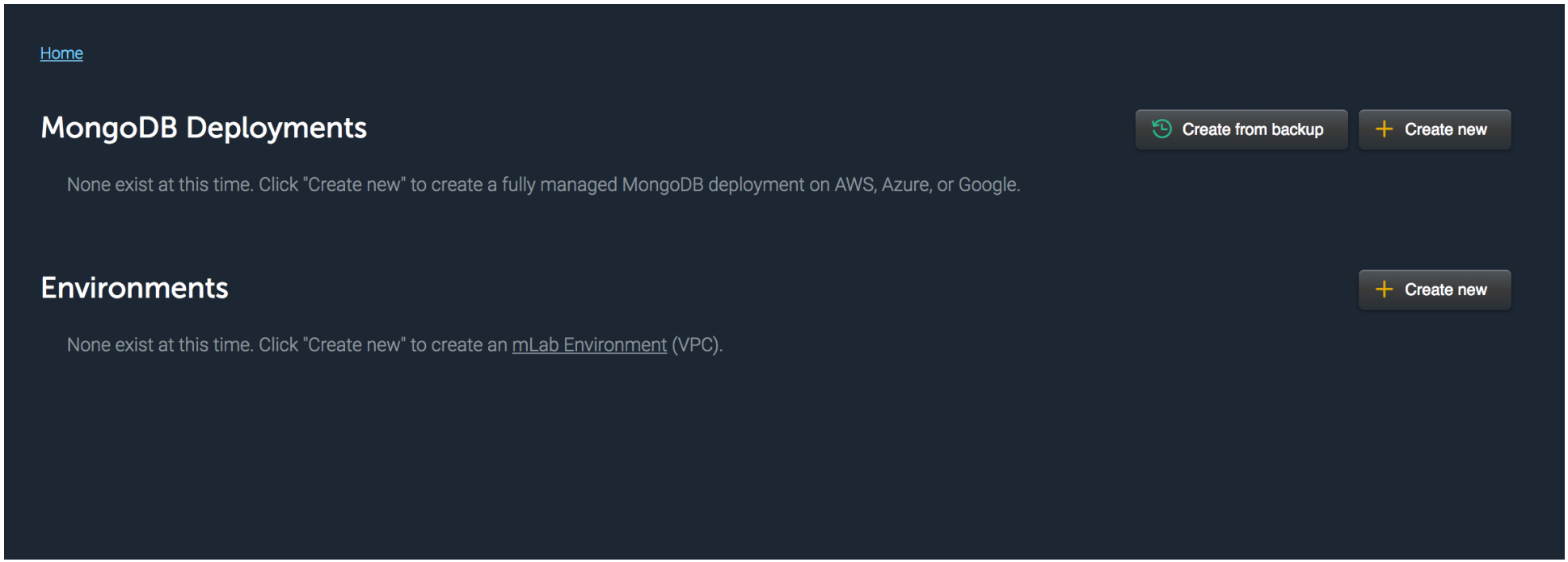
For your account's protection, the above link is good for single use and expires in one week.

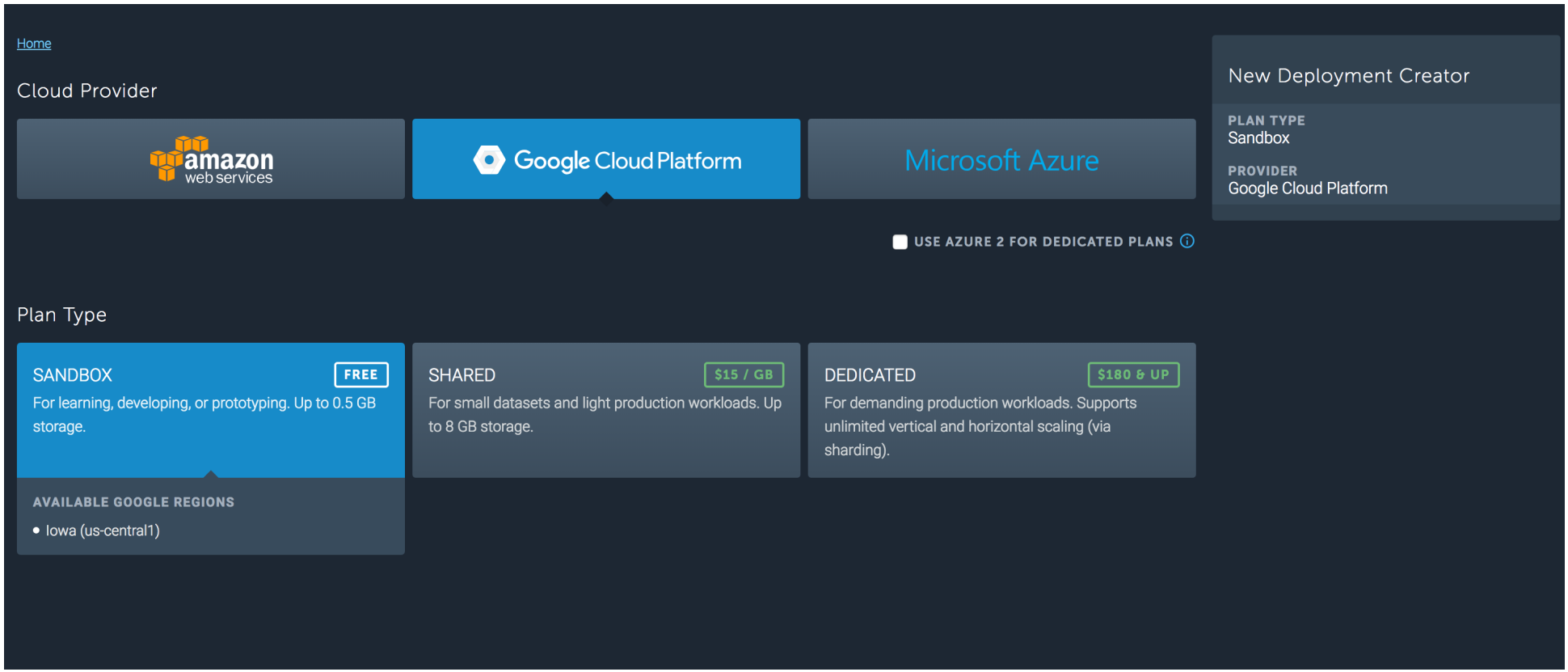
If you have never logged into this mLab account before, forward this email to the account's Admin User (ulgoon89@gmail.com) who will either provide you with your mLab username/password OR click the verification link on your behalf because your email is not associated with a login profile (e.g., Billing, Technical, or Emergency Contact).

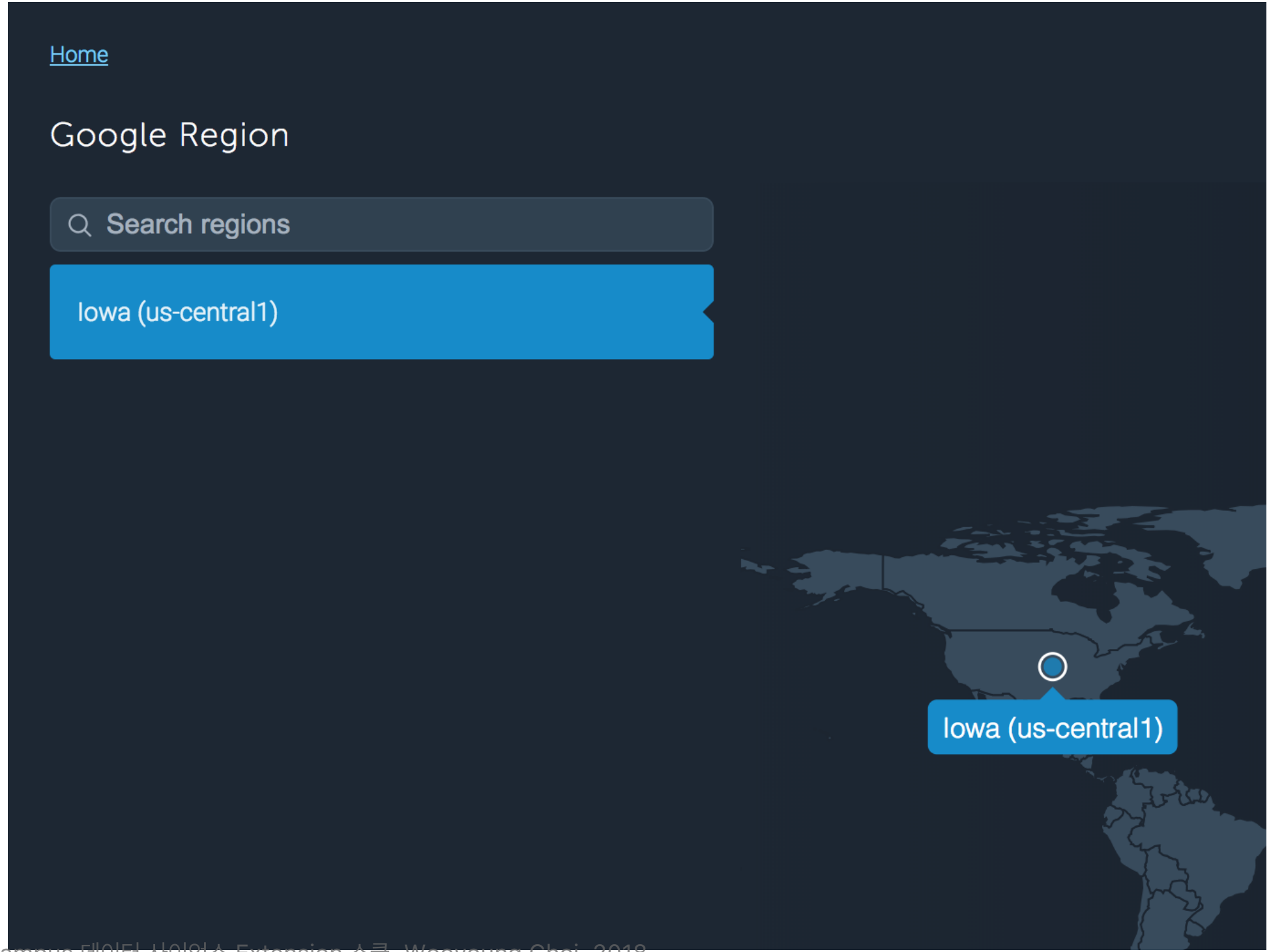
Best regards,
mLab

https://mlab.com

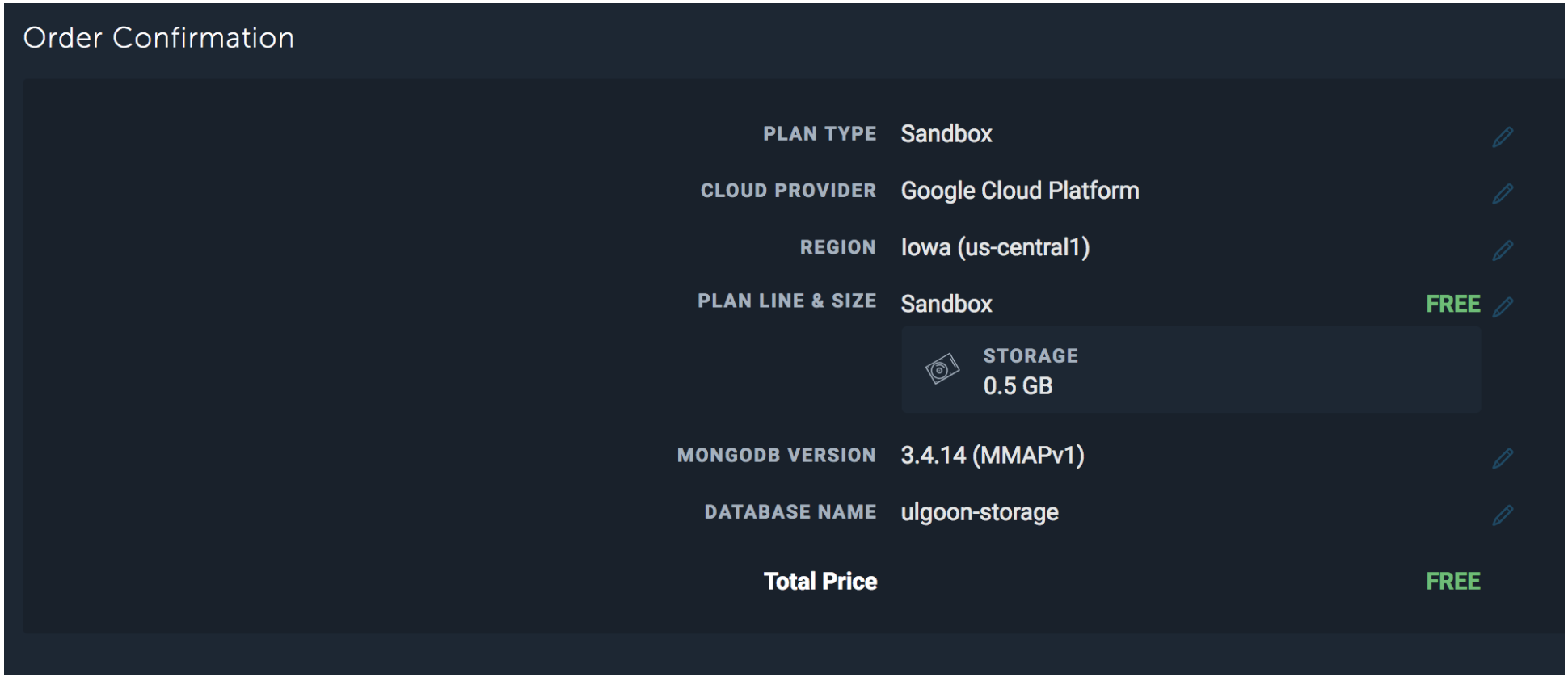
Create New DB

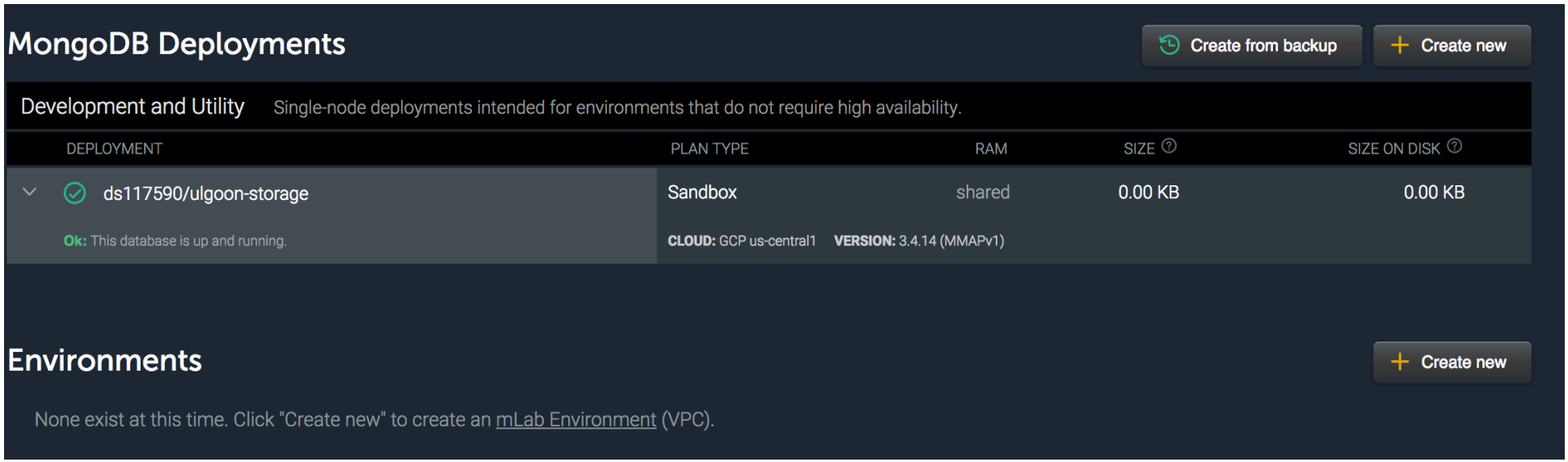

Home
MongoDB Deployments
Create from backup
Create new
None exist at this time. Click "Create new" to create a fully managed MongoDB deployment on AWS, Azure, or Google.
Environments
Create new
None exist at this time. Click "Create new" to create an mLab Environment (VPC).


Home
Cloud Provider
amazon web services
Google Cloud Platform
Microsoft Azure
USE AZURE 2 FOR DEDICATED PLANS
New Deployment Creator
PLAN TYPE
Sandbox
PROVIDER
Google Cloud Platform
Plan Type
SANDBOX
FREE
For learning, developing, or prototyping. Up to 0.5 GB storage.
SHARED
$15 / GB
For small datasets and light production workloads. Up to 8 GB storage.
DEDICATED
$180 & UP
For demanding production workloads. Supports unlimited vertical and horizontal scaling (via sharding).
AVAILABLE GOOGLE REGIONS
Iowa (us-central1)

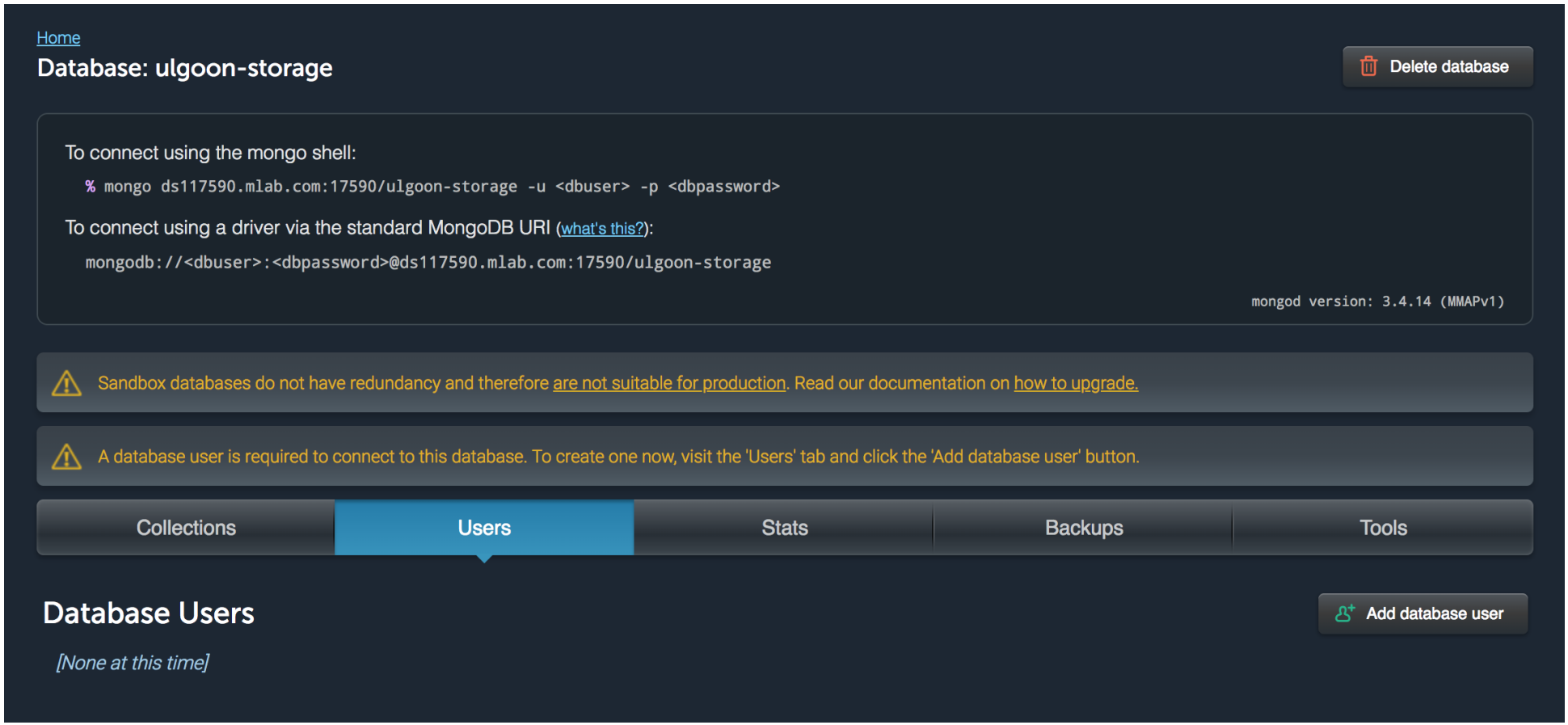

Home
Google Region
Search regions
Iowa (us-central1)
Iowa (us-central1)

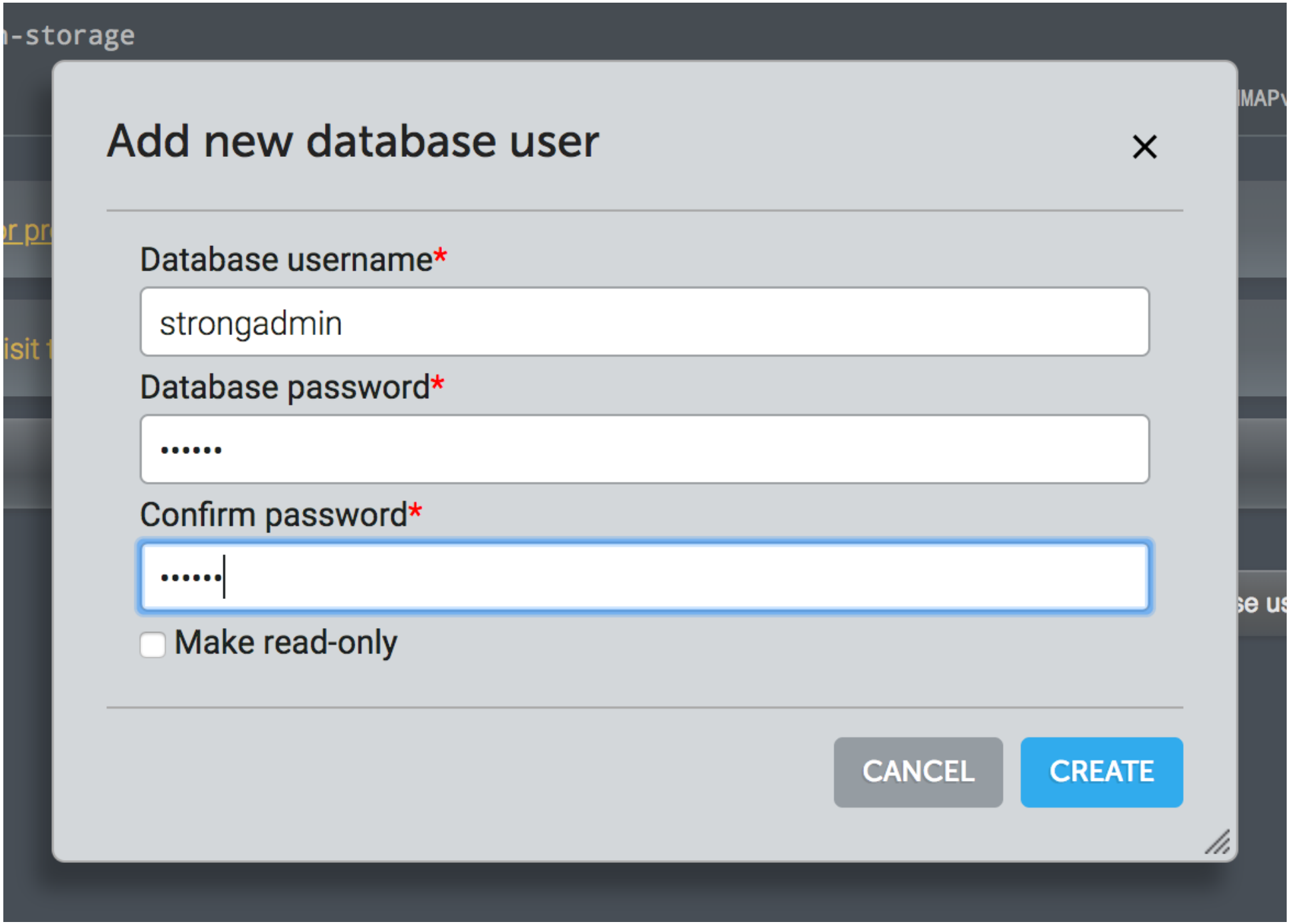

MONGODB VERSION
3.4.14 (MMAPv1)
DATABASE NAME
ulgoon-storage
New Deployment Creator
PLAN TYPE
Sandbox
PROVIDER
Google Cloud Platform
REGION
Iowa (us-central1)
PLAN LINE & SIZE
- 0.5 GB
MONGODB VERSION
3.4.14 (MMAPv1)
DATABASE NAME
ulgoon-storage

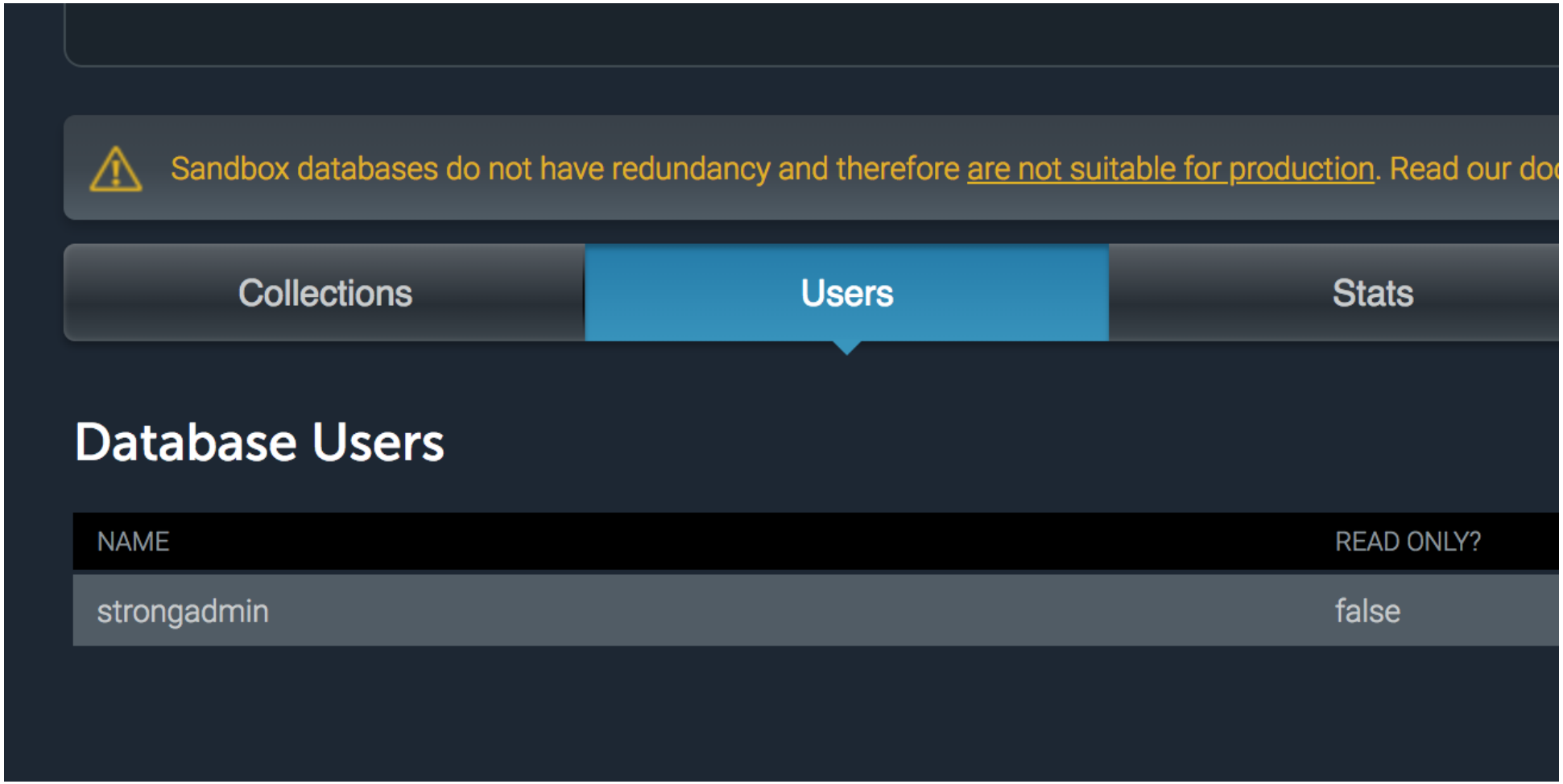

Order Confirmation
PLAN TYPE
Sandbox
CLOUD PROVIDER
Google Cloud Platform
REGION
Iowa (us-central1)
PLAN LINE & SIZE
Sandbox
FREE
STORAGE
0.5 GB
MONGODB VERSION
3.4.14 (MMAPv1)
DATABASE NAME
ulgoon-storage
Total Price
FREE


MongoDB Deployments
Create from backup
Create new
Development and Utility
Single-node deployments intended for environments that do not require high availability.
DEPLOYMENT
PLAN TYPE
RAM
SIZE
SIZE ON DISK
ds117590/ulgoon-storage
Sandbox
shared
0.00 KB
0.00 KB
Ok: This database is up and running.
CLOUD: GCP us-central1
VERSION: 3.4.14 (MMAPv1)
Environments
Create new
None exist at this time. Click "Create new" to create an mLab Environment (VPC).

Add New User


Home
Database: ulgoon-storage
Delete database
To connect using the mongo shell:
% mongo ds117590.mlab.com:17590/ulgoon-storage -u <dbuser> -p <dbpassword>
To connect using a driver via the standard MongoDB URI (what's this?):
mongodb://<dbuser>:<dbpassword>@ds117590.mlab.com:17590/ulgoon-storage
mongod version: 3.4.14 (MMAPv1)
Sandbox databases do not have redundancy and therefore are not suitable for production. Read our documentation on how to upgrade.
A database user is required to connect to this database. To create one now, visit the 'Users' tab and click the 'Add database user' button.
Collections
Users
Stats
Backups
Tools
Database Users
Add database user
[None at this time]


-storage
Add new database user
Database username*
strongadmin
Database password*
Confirm password*
Make read-only
CANCEL
CREATE


Sandbox databases do not have redundancy and therefore are not suitable for production. Read our doc
Collections
Users
Stats
Database Users
NAME
READ ONLY?
strongadmin
false

How to connect to MongoDB

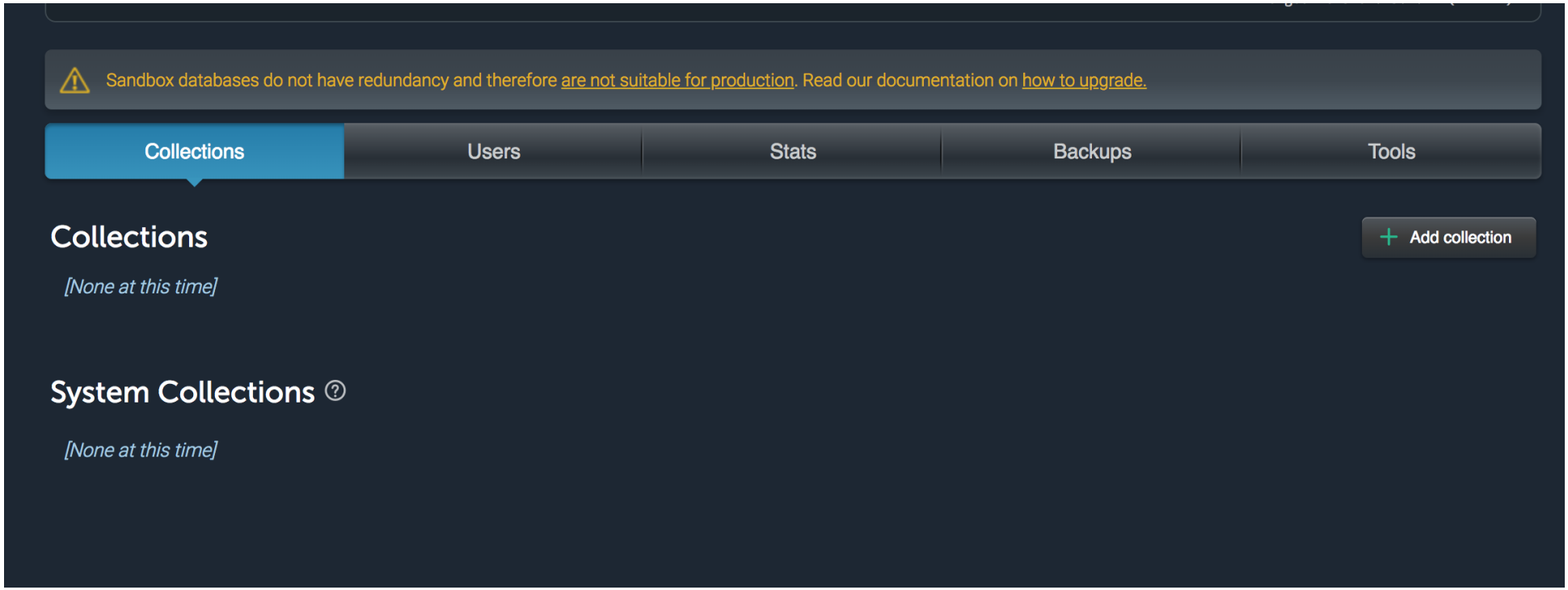
To connect using the mongo shell:

```
% mongo ds117590.mlab.com:17590/ulgoon-storage -u <dbuser> -p <dbpassword>
```

To connect using a driver via the standard MongoDB URI (what's this?):

```
mongodb://<dbuser>:<dbpassword>@ds117590.mlab.com:17590/ulgoon-storage
```

Create New Collection


Sandbox databases do not have redundancy and therefore are not suitable for production. Read our documentation on how to upgrade.
Collections
Users
Stats
Backups
Tools
Collections
Add collection
[None at this time]
System Collections
[None at this time]

MongoDB with jupyter

connect

```
from pymongo import MongoClient
client = MongoClient("mongodb://..")
client.{DBName}.collection_names()
```

Assign DB, Collection

```
db = client.{DBName}
new_collection = db.{CollectionName}

or

db = client[{DBName}]
new_collection = db[{CollectionName}]
```

INSERT data

```
some_user = {
        "name":"Fastcampus",
    "email":"help@fastcampus.co.kr",
}
new_collection.insert_one(some_user)
```

SELECT

```
new_collection.find_one()

or

query = {}
new_collection.find_one(query)
```

SELECT all data

```
query = {}
cursor = new_collection.find(query)
[item for item in cursor]
```

SELECT * WHERE name = "fastcampus"

```
query = {"name":"fastcampus"}
new_collection.find_one(query)
```

INSERT lots of data in one time

```
data_list = [
        {
        "name":"jyp",
        "email":"jyp@fastcampus.co.kr",
    },
    {
        "name":"gd",
        "email":"gd@fastcampus.co.kr",
    },
]

new_collection.insert_many(data_list)
```

WHERE in ("jyp", "gd")

```
query = {
        "name":{
        "$in": ["jyp", "gd"]
        }
}
new_collection.find(query)
```

AND, OR

```
query = {
        "name":"jyp",
    "email":"jyp@fastcampus.co.kr",
}
new_collection.find(query)
```

```
query = {
        $or:[
        {"name":"jyp"},
        {"email":"gd@fastcampus.co.kr"},
    ]
}
new_collection.find(query)
```

Operator

`{field:{<operator>:<value>}}`

Operator	NoSQL
=	$eq
!=	$ne
>	$gt
>=	$gte
<	$lt
<=	$lte
IN	$in
NOT IN	$nin

COUNT(*)

```
new_collection.count()
```

GROUP BY

```
# like temporary table
cursor = collection.aggregate([
        {
        "$group":{condition}
    }
])
```

pymongo with requests

import requests

```
import requests

url = ""
headers = {}

response = requests.get(url, headers=headers)
```

json decode

```
item_list = response.json()["items"]
```

insert lots of data

```
item_list.insert_many(item_list)
```

MongoDB with Educational information

MongoDB with Real Estate