2015

Morpheus Getting Started Guide

MORPHEUS

Copyright Notices

Morpheus Data, LLC
800 Concar Drive Suite 100
San Mateo CA 94402
Tel: 650.358.5000
Fax: 650.358.5001
www.GOMORPHEUS.com

MORPHEUS

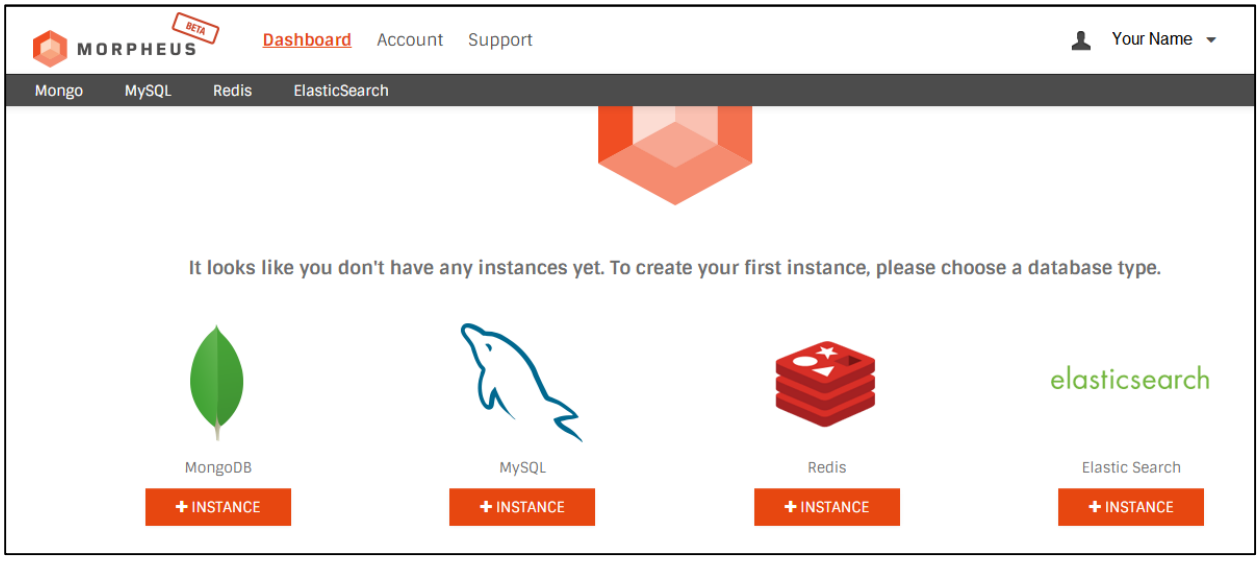
Table of Contents

Getting Started with Morpheus-created Instances

This guide shows how to create, connect and, if need be, migrate legacy database data to a new Morpheus-created cloud database instance for these database types:

- MongoDB
- MySQL
- Redis
- Elasticsearch

After you register for a new login, you are sent a Welcome to Morpheus email. Click VERIFY ACCOUNT in the email then select the type of database instance type you wish to create: MongoDB, MySQL, Redis or Elasticsearch.



Select Database Instance Type

Next, follow the *Getting Started* directions for the type of instance you selected.

Getting Started—MongoDB

Getting Started— MySQL

Getting Started— Redis

Getting Started— Elasticsearch

Getting Started—MongoDB

Before you begin, you will need the following…

- the name you wish to call your MongoDB instance
- IP address or range of addresses for machines that will have access
- a database name
- at least one user name/password

TL;DR

Follow these steps to setup a MongoDB instance:

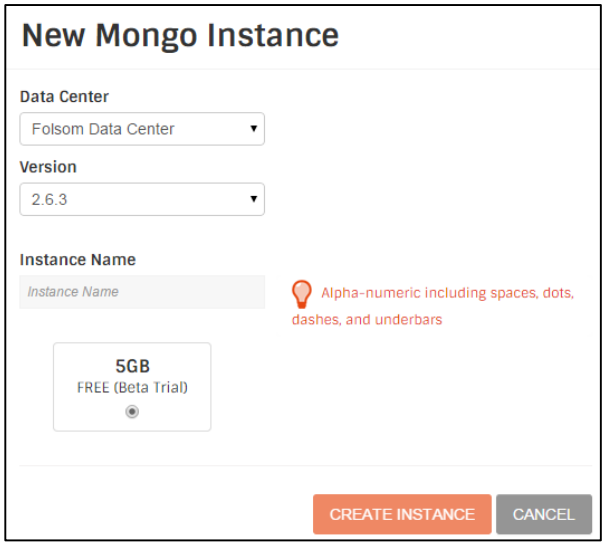
1. Create a MongoDB database instance.
2. Add IP addresses to Access Control List (ACL).
3. Add a database and username.

Then use a client tool or the MongoDB command line interface (CLI) to confirm your connection. We show you both ways. You are now ready to migrate any legacy MongoDB data to new your Morpheus-created cloud DB.

Step 1. Create a MongoDB instance.

Click 

to create your MongoDB instance. Enter the **Data Center, Version** and **Instance Name**. This alpha-numeric name can accommodate spaces, dots, dashes, and underbars.

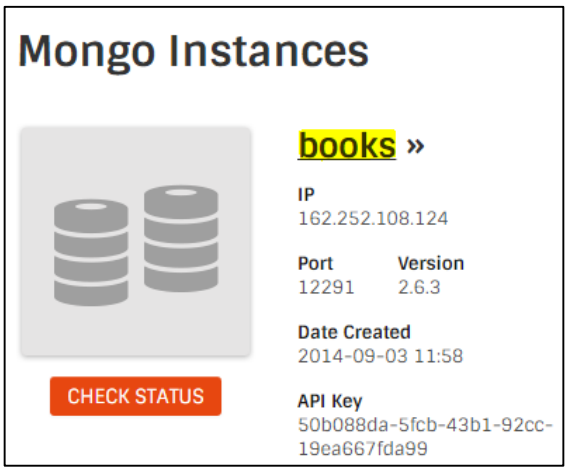


Add New MongoDB Instance Name

Click **CREATE INSTANCE**. This gives you 5GB of a MongoDB instance free. Post-beta, there will be more plan options. Check goMorpheus.com for details.

Step 2. Set up ACL (Access Control List)

An Access Control List (ACL) is a white list that specifies which computers are allowed connect to an instance. An ACL secures your database instance from unwanted intruders.
To set up your ACL, click on your new Mongo instance name. This example uses "books."



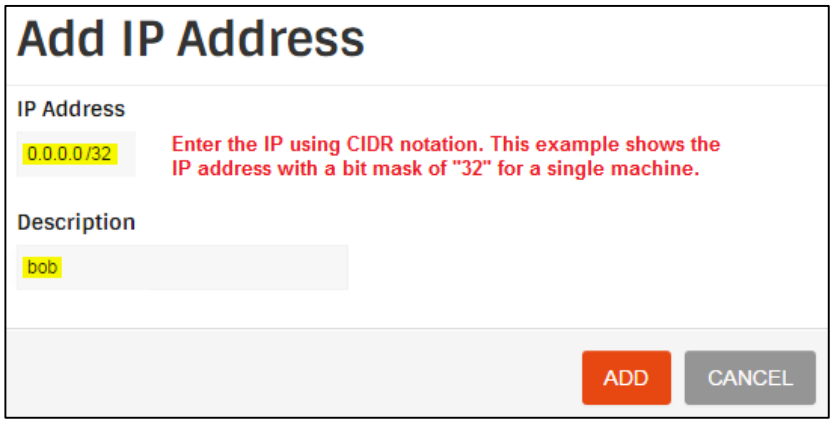
Click on Mongo Instance Name

Click ⇄ ACL / Connections and then the **+ACL** button on the far right.

Click ACL Connections, +ACL

Enter the **IP Address** and **Description** of a machine or range of machines that you will allow to connect to your instance.

TIP: An easy way to find out your machine's IP address is by using whatismyip.com.



Enter IP Address in CIDR Notation

IP Addresses in the Morpheus ACL tool use IPv4 CIDR notation, briefly…

- `/32` for one IP address
- `/24` for a range of 256 IPs
- `/22` for a range of 1024 IPs

Two examples:

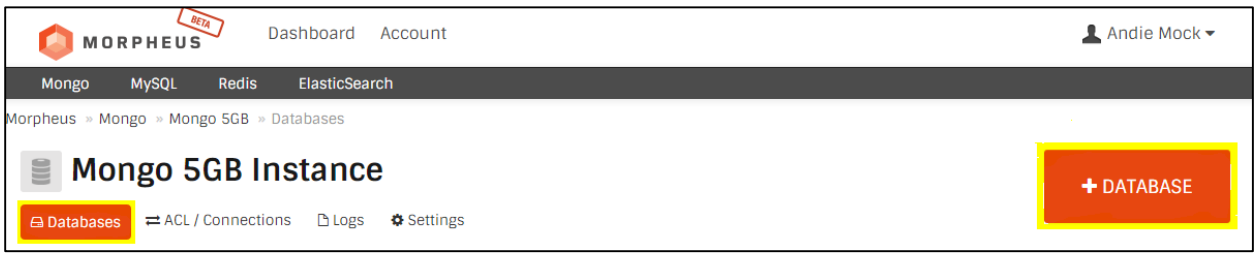
172.22.104.73/32 - a single IP Address
172.22.104.0/24 - all IP addresses between 172.22.104.0 - 172.22.104.255

⚠ Be sure to use the /32 prefix after the IP address. If you use the /0 prefix after the IP address, this opens the port to everyone. For example, specify the IP address 203.0.113.1 as 203.0.113.1/32.

NOTE: If you have a problem, make sure that you did not enter the IP that was just given to you for your instance. ACL authenticates computers connecting *to* this database instance.

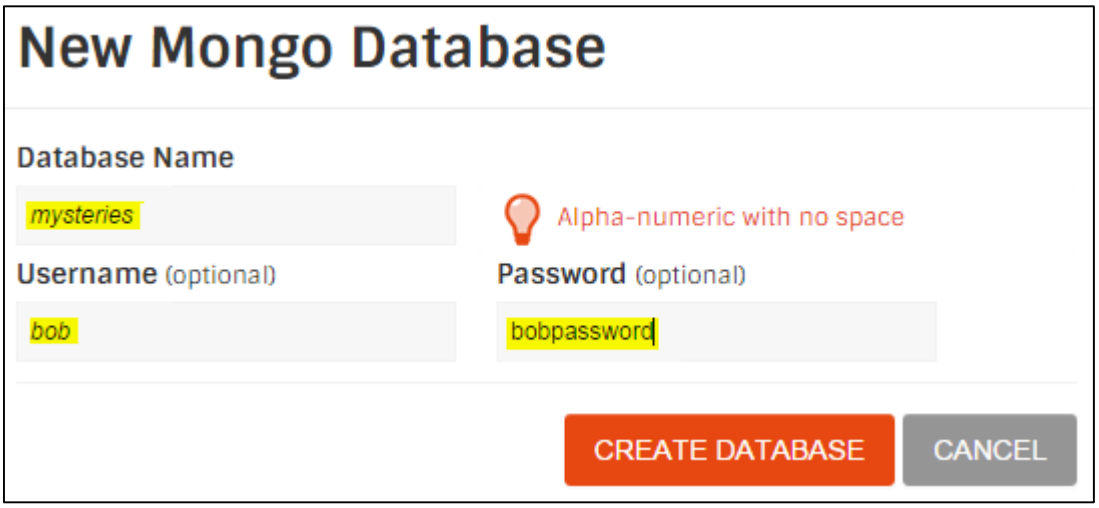
Step 3. Add Database and Username

Add a database by selecting Databases then clicking the **+DATABASE** button on the far right.



Click Databases and +Database

Enter the name of the database, in this example, “mysteries”. You cannot use spaces.



Enter Database Name, Username, Password

Click **CREATE DATABASE**.

⚠ **IMPORTANT:** Remember your username/password combination. You will need this to connect to your new database.

If you do not enter a username/password, a “default” user will be automatically created.

TIP: Mongo associates the usernames with the database, not the instance.

Congratulations. Now that you have a new Mongo instance. Let’s confirm your ability to connect to it.

Confirm MongoDB Connection

This section shows two ways to confirm your database connection

- connecting with the command line interface (CLI), or
- connecting using a third-party client tool.

Both methods use the IP address and port information found on the Mongo Instances page. You will also need the name of the database and the username/password you created in Step 3. Users are listed on your database page. If you forgot the password for this user, add another username/password combination and connect with this user.

Connecting Using the CLI

You can connect through the command line interface if you've installed the MongoBD binaries.

Here's a string to test the connection:

```
mongo [host]:[port]/[database] -u [username] -p [password]
```

Example: Using our "books" instance and "mysteries" database example, our CLI connection string would look like the following:

```
mongo 162.252.108.124:12291/mysteries -u bob -p bobpassword
```

The components of this string are:

host
The IP address found on the Mongo Instances page.

port
The port number found on the Mongo Instances page.

database
The database name you created in Step 3.

username
A username you added either in Step 3.

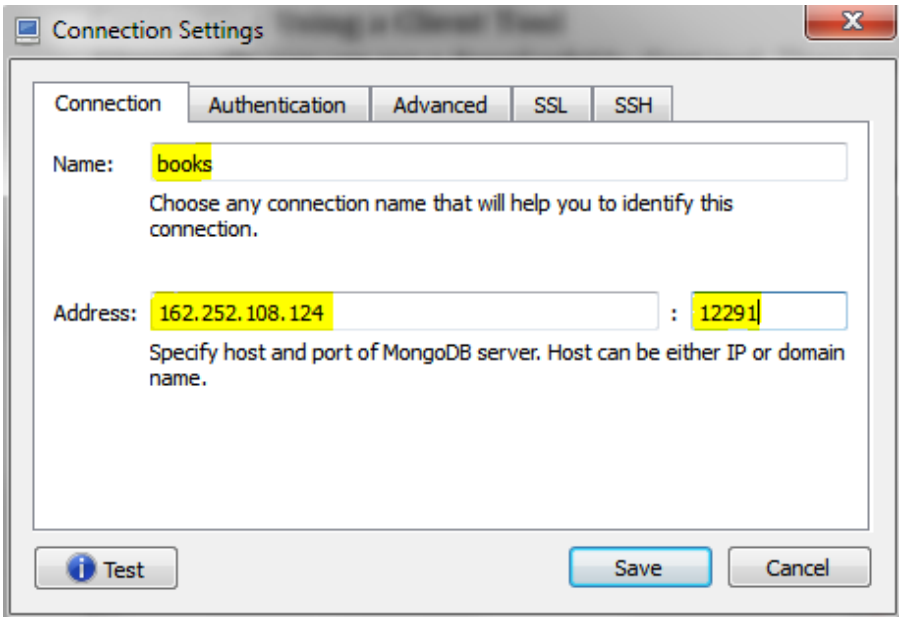
password
The password for this database username. **NOTE:** This is not the password for your Morpheus account.

Connecting Using a Client Tool

Alternatively, you can use a downloadable client tool. There are many available.
Our example uses the free, cross-platform Robomango client.

Connection Tab

In the Robomango client enter the new instance **Name**. The **Address** and **Port** are found on the Mongo Instances page. Click **Test** to confirm your connection to this instance.

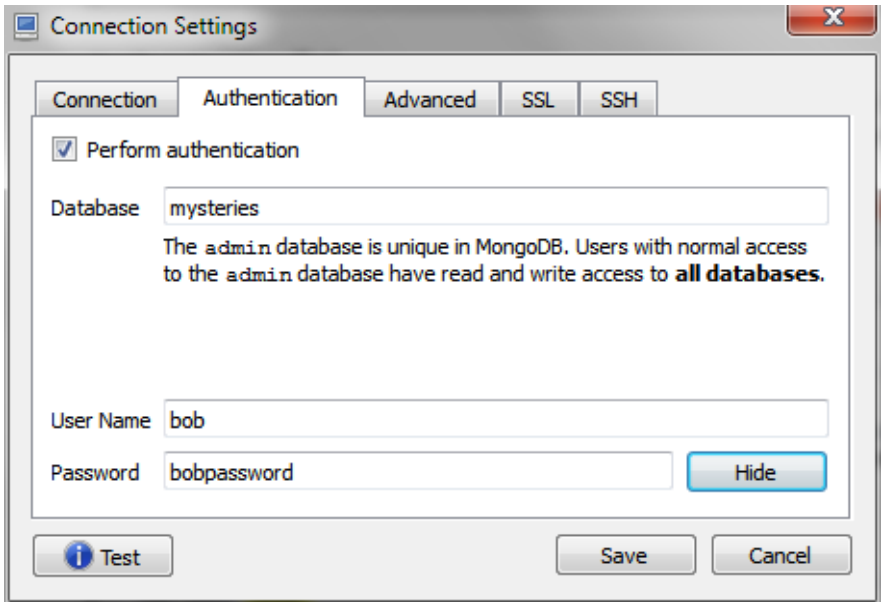


Connection Tab Settings

If you are moving from a local install to Morpheus cloud database and you have a connection tool already setup, simply swap the local IP/port for the new Morpheus IP/port.

Authentication Tab

Enter the **Database, User Name,** and **Password** created in Step 3.
Click **Test** to confirm that your name, password and ACL are setup up properly.



Authentication Tab Settings

Troubleshooting Connection

The most common connection problem is mismatch between the entries for User Name and IP address in the ACL, and the connection tool or CLI. If there is a mismatch, change either the ACL information or the CLI/ client tool connection information so that they match.

See Also

For details about programming with MongoDB, see the documentation found online at: http://docs.mongodb.org/manual/

Migrating MongoDB Data

Use *mongodump* and *mongorestore* to add data from your old MongoDB instance to the new Morpheus-created MongoDB instance in two steps:

1. Perform a binary dump of the old data using the *mongodump* command.
2. Connect to your new Morpheus MongoDB instance using the new MongoDB instance's IP address and port number and run the *mongorestore* command.

Example *mongodump*

The example below uses the mongo shell to runs *mongodump* to create a database dump located at `/opt/backup/morpheus_dump` from a database running on port `37017` on the host `oldmongo.example.net` using the username `bob` and password `bobspassword`.

```
//Example dump to file from old database

mongodump –host oldmongo.example.net –port 37017 –username bob –
password bobspassword /opt/backup/morpheus_dump
```

Example *mongorestore*

From the mongo shell, run `mongorestore` to add data from a binary database dump located at `/opt/backup/morpheus_dump` to a new MongoDB instance with IP address `162.252.108.123` running on port `12291` with username `jack` and password `jackspassword`.

```
//Example restore dump file to new database

mongorestore –host 162.252.108.123 –port 12291 –username jack –
password jackspassword /opt/backup/morpheus_dump
```

See Also

For complete details about the *mongodump* command see:
http://docs.mongodb.org/manual/reference/program/mongodump

For complete details about the *mongorestore* command see:
http://docs.mongodb.org/manual/reference/program/mongorestore

Getting Started— MySQL

Before you begin, you will need the following…

- the name you wish to call your MySQL instance,
- IP address or range of addresses for machines that will have access,
- a database name, and
- at least one user name/password.

TL;DR

Follow these steps to setup a MySQL instance:

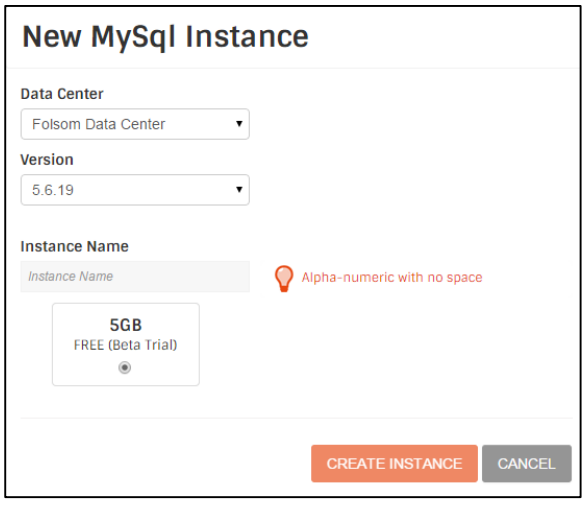
1. Create a MySQL database instance.
2. Add IP addresses to Access Control List (ACL).
3. Add username and (optional) database.

Step 1. Create a MySQL instance.

Click 

to create your MySQL instance.

Enter the database instance name. This alpha-numeric name can use spaces, dashes, and/or dots.

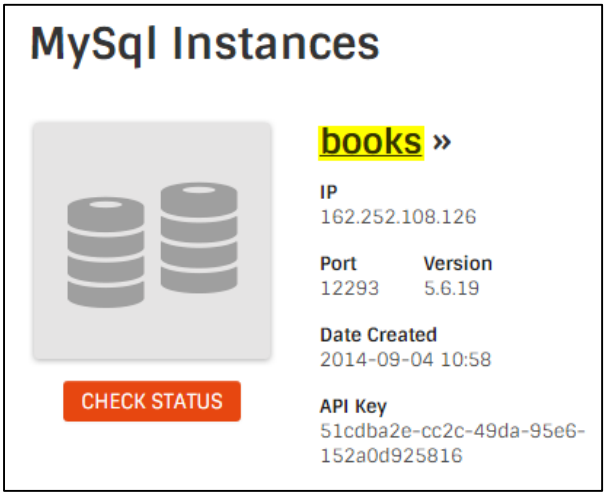


Enter the New Instance Name

Click **CREATE INSTANCE**. This gives you 5GB free. Post-beta, there will be more plan options. Check goMorpheus.com for details.

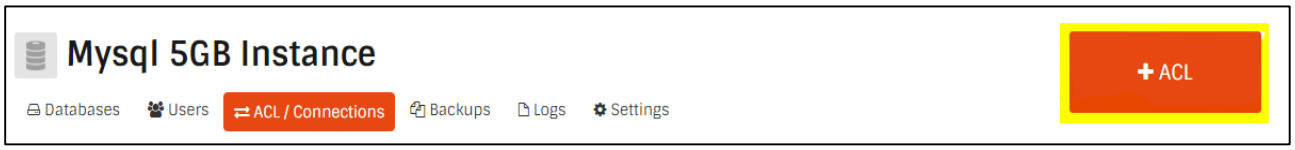
Step 2. Set up ACL (Access Control List)

An Access Control List (ACL) is a white list that specifies which computers are allowed connect to an instance. An ACL secures your database instance from unwanted intruders.
To set up your ACL, click on your MySQL database instance name, in this example "books."



Click on the Instance Name

Click ⇄ ACL / Connections and then the **+ACL** button on the far right.



Click ACL Connections, +ACL

Enter the **IP Address** and **Description** of a machine or range of machines that you will allow to connect to your instance.

TIP: An easy way to find out your machine's IP address is by using whatismyip.com.

IP Addresses in the Morpheus ACL tool use IPv4 CIDR notation, briefly…

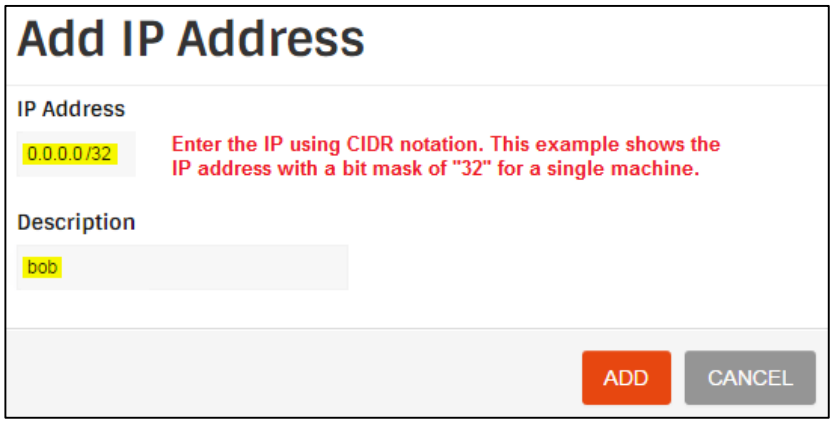
- `/32` for one IP address
- `/24` for a range of 256 IPs
- `/22` for a range of 1024 IPs

Two examples:

172.22.104.73/32 - a single IP Address
172.22.104.0/24 - all IP addresses between 172.22.104.0 - 172.22.104.255

⚠ Be sure to use the /32 prefix after the IP address. If you use the /0 prefix after the IP address, this opens the port to everyone. For example, specify the IP address 203.0.113.1 as 203.0.113.1/32.

NOTE: If you have a problem, make sure that you did not enter the IP that was just given to you for your instance. ACL authenticates computers connecting *to* this database instance.

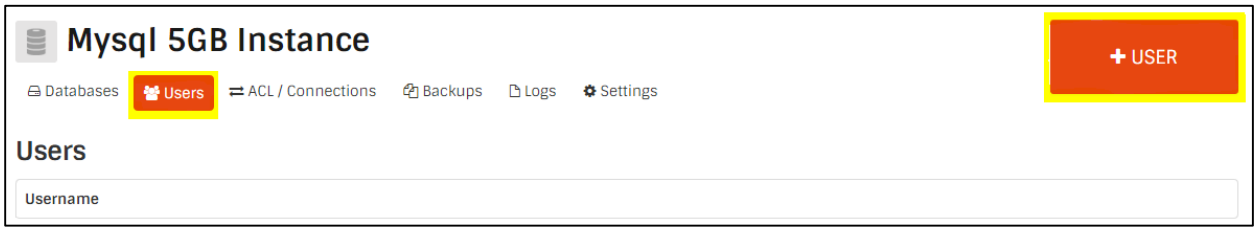


Enter IP Address Using CIDR Notation

Click **ADD**.

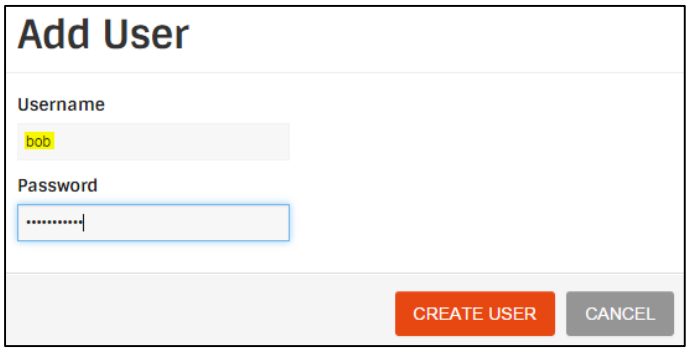
Step 3. Setup User and Database

Setup usernames/passwords by clicking Users then click the **+Users** button on the far right.



Click Users, +User

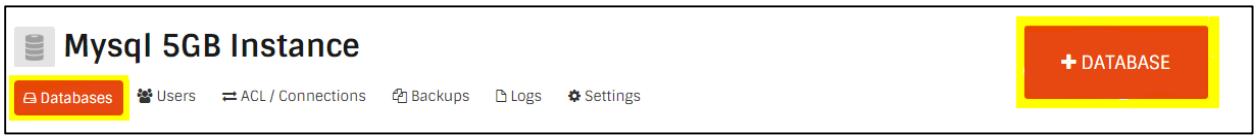
Enter the name and password for at least one user. **NOTE:** MySQL associates users with the instance, not the database.



Enter Username and Password

Click **CREATE USER**.

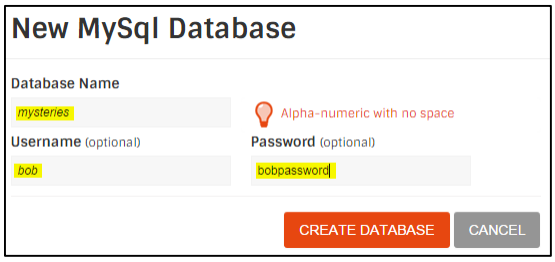
Add a database by selecting Databases and clicking the **+DATABASE** button on the far right.



Click Databases, + Database

Click **DATABASES** then + **DATABASE.**

Enter the alpha-numeric **Database Name** with no spaces, a **Username** and **Password**.
⚠ **IMPORTANT:** Make sure that this user's computer's IP address is added to the ACL.



Enter Database Name

Click **CREATE DATABASE**.

NOTE: when you create a MySQL instance, two databases, **performance_schema** and **test** are automatically created.

Congratulations! Your new MySQL database is set up. You can now test your connection to it using either a CLI connection or a MySQL connection tool. Both methods use the instance's IP

address and port found on the MySQL instance page and the username/password you created in Step 3. Users are listed on your database page.

 **TIP:** If you forget the password for this user, add another username/password combination and connect with this user.

Connecting Using the CLI

You can connect through the command line interface, referred to as the CLI, if you installed the MySQL binaries. See http://dev.mysql.com/downloads/mysql/ to download a MySQL instance.

Here's the syntax for a string that tests the connection to the instance:

```
Server=xxx.xxx.xxx.xxx;Port=xxxx;Database=myDataBase;
Uid=myUsername;Pwd=myPassword;
```

The components of this string are:

Server
The instance's IP address. Find this information on your Morpheus MySQL instance page.

Port
The instance's port number. Find this information on your Morpheus MySQL instance page.

Database
The name of the database you added in Step 3.

Uid
The MySQL username you added in Step 3.

Pwd
The password for the username you added in Step 3. Note, this is not the password for your Morpheus account.

Example:

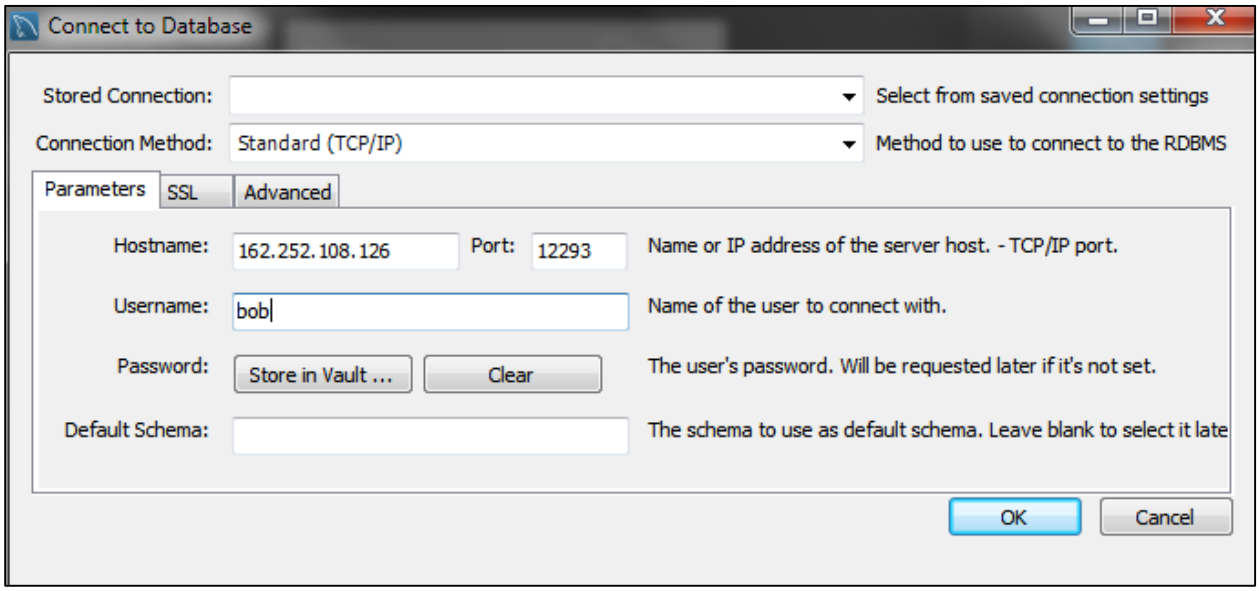
For the user, "bob" with a password "bobpassword", our "books" instance's IP address, and database named "mysteries" our CLI connection string looks like:

```
Server=191.215.159.123;Port=12293;Database=mysteries;Uid=bob;Pwd
=bobpassword;
```

Connecting with a MySQL Tool

Alternatively, you can use one of the many downloadable client tools. Our example shows the cross-platform, MySQL Workbench client.

In MySQL Workbench, select **Database** then **Connect to Database**.



Connection Parameters for MySQL

Hostname
The instance IP Address found on the Morpheus MySQL Instances page.

Port
The port number, also found on the Morpheus MySQL Instances page.

Username
The user you added in Step 3.

Troubleshooting Connections

The most common connection problem is mismatch between the entries for User Name and IP address in the ACL and the connection tool or CLI. If there is a mismatch, change either the ACL information or the CLI/tool connection information so that they match.

See Also

If you are moving from a local install to Morpheus cloud database, simply swap the IP/port. For details about programming with MySQL, see the excellent documentation found online at: http://dev.mysql.com/doc/index-topic.html

Migrating MySQL Data

Use *myqldump* and *mysqlimport* to transfer data from your old MySQL instance to your new, Morpheus-created MySQL instance.

Before you begin you will need…

- Host name or IP address and port number of old and new instances
- Username and password of old and new instances

TL;DR

1. Run *mysqldump* on your old MySQL instance.
2. Connect to your new Morpheus MySQL instance using the IP address and port number provided then run the *mysqlimport* command.

mysqldump Example

The example below uses the mysql shell to run *mysqldump* to create a database dump located at `/opt/backup/morpheus_dump` from a database running on port `37017` on the host `oldmysql.example.net` authenticated with the username, `bob`, and password, `bobspassword`.

```
//Example dump to file from old database

mysqldump –host oldmysql.example.net –p 37017 –username bob –
password bobspassword /opt/backup/morpheusdump
```

mysqlimport Example

The example below uses the mysql shell to run `mysqlimport` to add data from a binary database dump located at `/opt/backup/morpheus_dump` to a new Morpheus-created MySQL instance with IP address `162.252.108.123` running on port `12291` with username `jack` and password `jackspassword`.

```
//Example import dump file to new database

mysqlimport –host 162.252.108.123 –p 12291 –username jack –
password jackspassword /opt/backup/morpheus_dump
```

See Also

For complete details about the *mysqldump* command see:
http://dev.mysql.com/doc/refman/5.1/en/mysqldump.html

For complete details about the *myqlimport* command see:
http://dev.mysql.com/doc/refman/5.0/en/mysqlimport.html

Getting Started— Redis

Before you begin, you will need the following…

- the name you wish to call your new Redis instance,
- IP address or range of addresses for machines that will have access, and
- at least one user name/password.

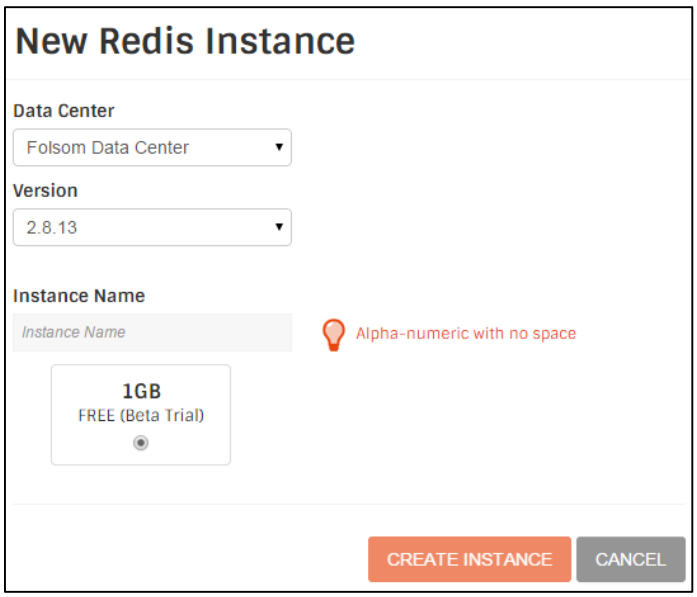
TL;DR

1. Create a Redis database instance.
2. Add IP addresses to Access Control List (ACL).

Step 1. Create a Redis instance.

Click 

to create your Redis instance. Select the **Data Center** and **Version** from the pull down options. Enter the database **Instance Name**. This alpha-numeric name can use spaces, dashes, and/or dots.

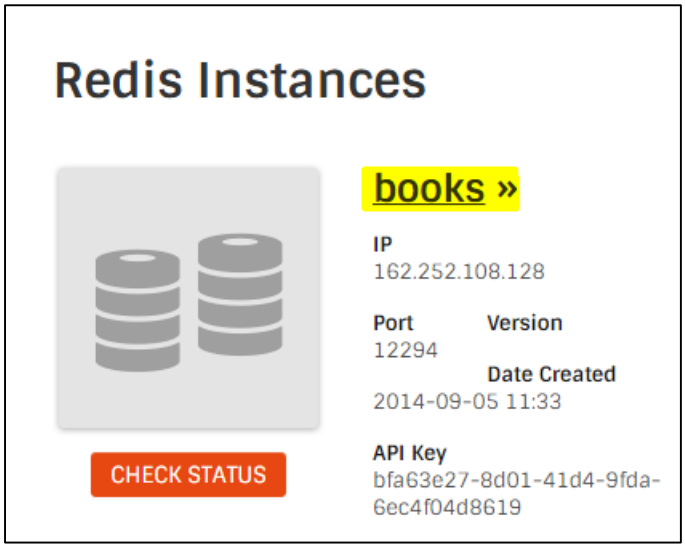


Enter New Redis Instance Name

Click **CREATE INSTANCE**. This gives you 1GB free. Post-beta, there will be more plan options. Check goMorpheus.com for details.

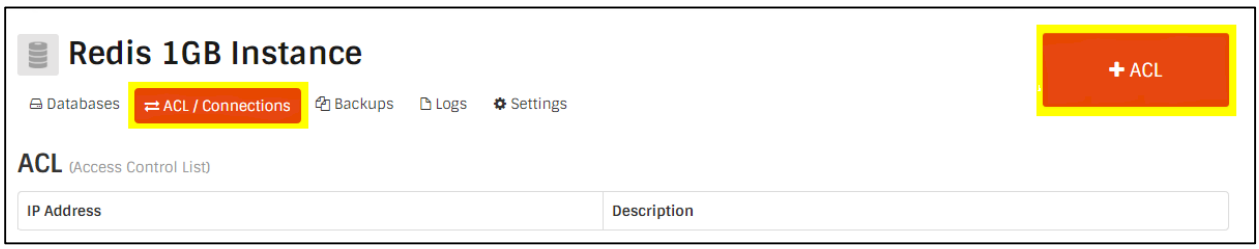
Step 2. Set up ACL (Access Control List)

An Access Control List (ACL) is a white list that specifies which computers are allowed connect to an instance. An ACL secures your database instance from unwanted intruders.
To set up your ACL, click on your Redis database instance name, in this example "books."



Click on New Instance Name

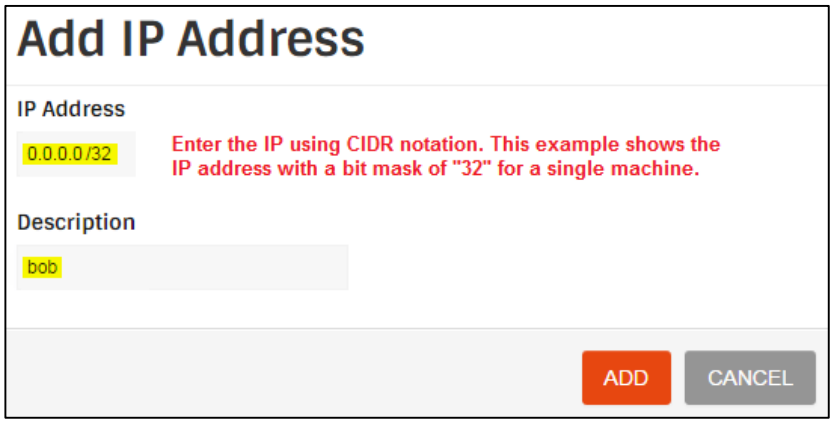
Click ⇄ ACL / Connections and then the **+ACL** button on the far right.



Click ACL Connections, +ACL

Enter the **IP Address** and **Description** of a machine or range of machines that you will allow to connect to your instance.

TIP: An easy way to find out your machine's IP address is by using whatismyip.com.



Add IP Address in CIDR Notation

IP Addresses in the Morpheus ACL tool use IPv4 CIDR notation, briefly…

- `/32` for one IP address
- `/24` for a range of 256 IPs
- `/22` for a range of 1024 IPs

Two examples:

172.22.104.73/32 - a single IP Address
172.22.104.0/24 - all IP addresses between 172.22.104.0 - 172.22.104.255

⚠ Be sure to use the /32 prefix after the IP address. If you use the /0 prefix after the IP address, this opens the port to everyone. For example, specify the IP address 203.0.113.1 as 203.0.113.1/32.

NOTE: If you have a problem, make sure that you did not enter the IP that was just given to you for your instance. ACL authenticates computers connecting *to* this database instance.

Your Redis database is set up. You can now test your connection by …

- connecting with the Redis command line interface, **redis-cli**, or
- connecting using the cross-platform tool, Redis Desktop Manager.

Both methods will use the instance's IP address and port found on the Redis instance page and the name of the Redis database instance you created in Step 1.

Connecting to the Redis CLI

If you have already installed the Redis binaries, you can test your connection using **redis-cli**, the Redis command line interface (CLI). **NOTE:** See http://redis.io/download for Redis binaries.

```
redis-cli -h "YOUR REDIS IP ADDRESS" -p "YOUR PORT NUMBER"
```

Where

"YOUR REDIS IP ADDRESS" is the IP address found on your Redis instance page.

"YOUR REDIS PORT" is the port number found on your Redis instance page.

Example

```
redis-cli -h 162.252.108.126 -p 12293
```

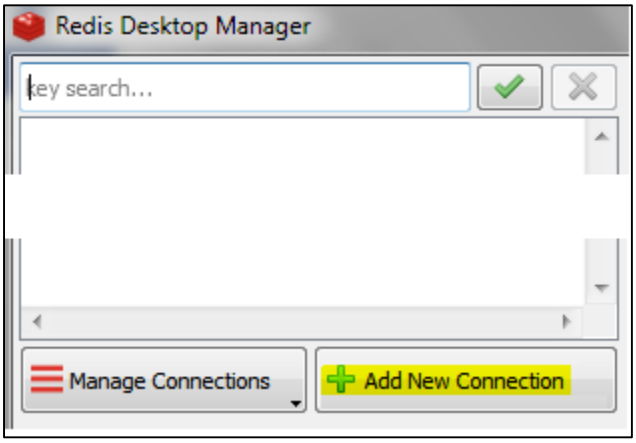
Troubleshooting

If you have problems, make sure your client machine's IP address is listed in your ACL list.

Connecting Using the Redis Desktop Manager

Alternatively, you can use one of several, downloadable client tools. Our example shows the Redis Desktop Manager found here http://redisdesktop.com/.

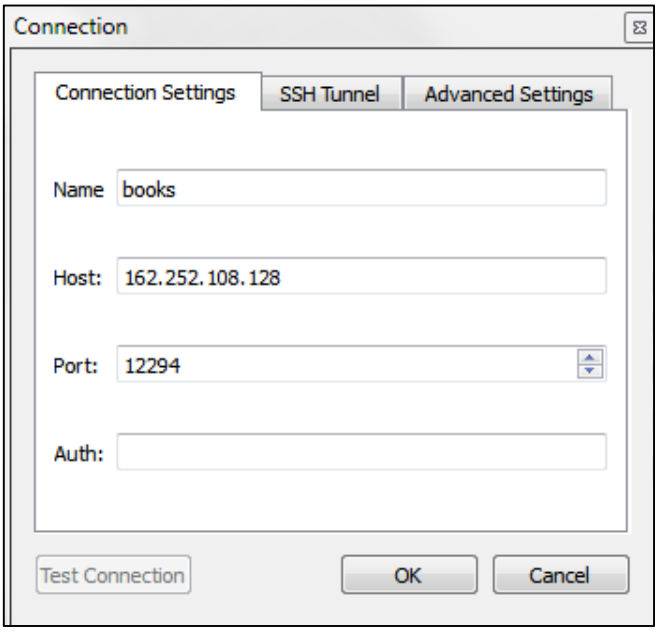
In the Redis Desktop Manager, click **Add New Connection** (at the bottom of the page.)



Add New Redis Connection

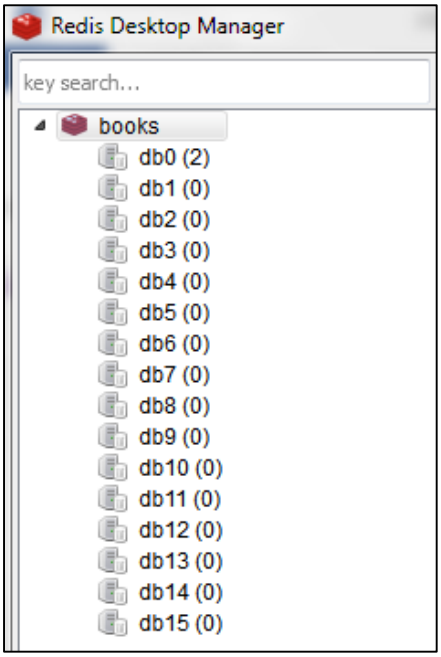
Name is the instance name. **Hostname** is the IP address, **Port** the port number, both are found on the Morpheus Redis Instances page.

Our example below uses “books” as the instance name.



Enter Instance Name and Port Number

Click on “books”. A properly setup connection lists `db0` through `db15`.



Proper Setup for “books”

If you are moving from a local install to Morpheus cloud database, simply swap the IP/port. For details about programming with Redis, see the excellent documentation found online at: http://redis.io/documentation

Migrating Redis Data

Use master-slave replication to move your old Redis instance to your Morpheus Redis instance.

1. Sync the old Redis instance with the new Morpheus instance using the `SLAVEOF` command, for example `SLAVEOF oldip:oldport`
2. If you have not already done so, configure new master to use old master's password.
3. On the new instance run the command, `SLAVEOF NO ONE`

Replication Example

Using the redis-cli, login to the new Morpheus master and type:

```
//Example SLAVEOF command issued on Redis Morpheus instance

redis-cli> SLAVEOF 127.0.0.1 12291
```

At this point the new Morpheus master will also be a slave of the old master. If your old master has a password, the new master/slave needs to be able to use that password. To do it on a running instance, use the redis-cli on the new Morpheus instance and type:

```
//Set authorization on new slave

redis-cli> config set masterauth <password>
```

Once the sync is complete, the data has been replicated to the new master that is still a slave of the old instance so next, issue the following on the Morpheus instance.

```
//Remove Morpheus instance as a SLAVE

redis-cli> SLAVEOF NO ONE
```

Be sure to point your application to the new Redis instance.

See Also

More information about replicating redis servers can be found here:

http://redis.io/topics/replication

Getting Started— Elasticsearch

Before you begin, you will need the following…

- the name you wish to call your new Elasticsearch instance, and
- an IP address or range of addresses for machines that will have access

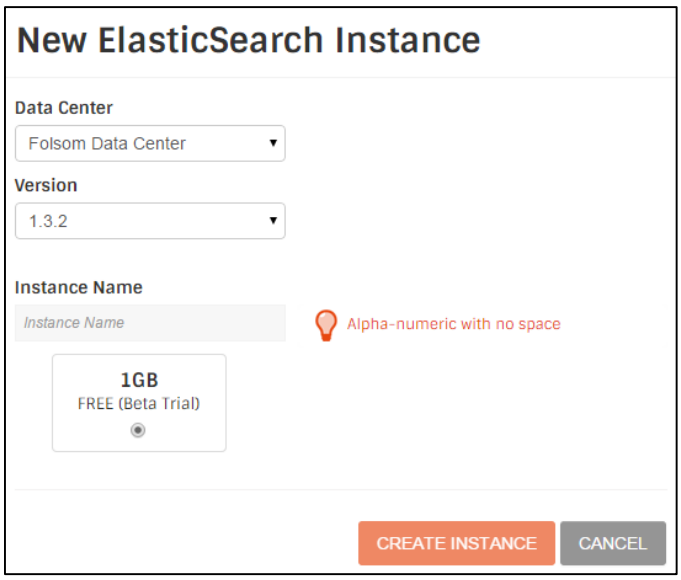
TL;DR

1. Create your first SSD-hosted, Elasticsearch database instance.
2. Add IP addresses to Access Control List (ACL) Connections.
3. Confirm your connection using the cURL command line tool.

Step 1. Create an Elasticsearch instance.

Click 

to create your Elasticsearch instance. Enter the database instance name. This alpha-numeric name can use spaces, dashes, and/or dots.

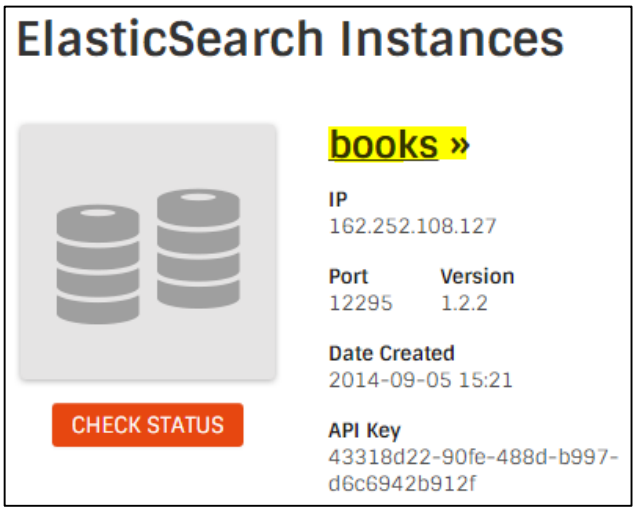


Enter New Instance Name

This gives you 1GB free. Post-beta, there will be more plan options. Check goMorpheus.com for details.

Step 2. Set up ACL (Access Control List) Connections

An Access Control List (ACL) is a white list that specifies which computers are allowed connect to an instance. An ACL secures your database instance from unwanted intruders.
To set up your ACL Connections, click on your Elasticsearch database instance name, in this example "books."



Click on New Instance Name

Click ⇄ ACL / Connections and then the **+CONNECTIONS** button on the far right.

Click ACL/Connections, +Connections

Enter the **IP Address** and **Description** of a machine or range of machines that you will allow to connect to your instance.

TIP: An easy way to find out your machine's IP address is by using whatismyip.com.

IP Addresses in the Morpheus ACL tool use IPv4 CIDR notation, briefly…

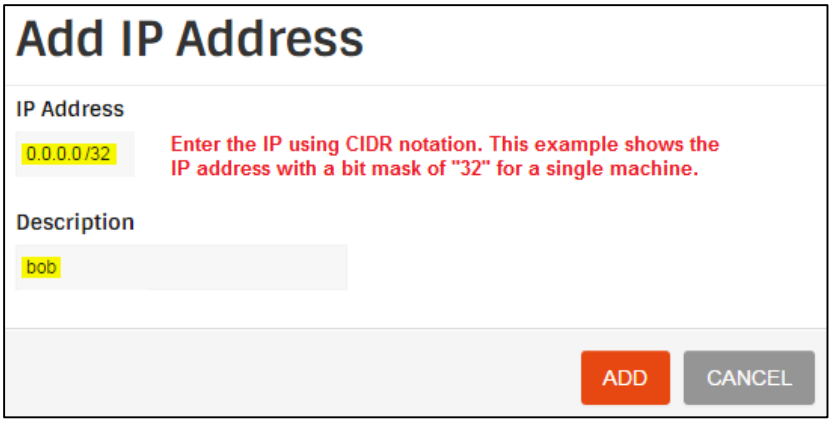
- `/32` for one IP address
- `/24` for a range of 256 IPs
- `/22` for a range of 1024 IPs

Two examples:

172.22.104.73/32 - a single IP Address
172.22.104.0/24 - all IP addresses between 172.22.104.0 - 172.22.104.255

⚠ Be sure to use the /32 prefix after the IP address. If you use the /0 prefix after the IP address, this opens the port to everyone. For example, specify the IP address 203.0.113.1 as 203.0.113.1/32.

NOTE: If you have a problem, make sure that you did not enter the IP that was just given to you for your instance. ACL authenticates computers connecting *to* this database instance.



Enter IP Address and Description

Click **ADD.**

Congratulations. You are now ready to test your connection.

Step 3. Confirm Connection

Because Elasticsearch is HTTP-based, it can be tested with a simple tool such as cURL.
The examples below retrieves your instances cluster health and cluster stats as JSON documents:

```
curl -XGET 'http://<IP Address>:<port>/_cluster/health?pretty=true'
curl -XGET 'http://<IP Address>:<port>/_cluster/stats?pretty=true'
```

Where…

IP Address
The IP Address of the Elasticsearch instance you just created. Find this information on your Morpheus Elasticsearch instance page.

Port
The port number for this instance. Find this information on your Morpheus Elasticsearch instance page.

Troubleshooting
Make sure that the machine running cURL has its IP address added to the ACL.

If you are moving from a local install to Morpheus cloud database, simply swap the IP/port.

See Also
For details about programming with Elasticsearch, see the documentation found online at:
http://www.elasticsearch.org/guide/

Migrating Elasticsearch Data

For migration, we recommend using the small script, Elasticsearch-Exporter (available on github) to export data from one Elasticsearch instance into another.

Export Example

From the node.js CLI, use `exporter.js` to move Elasticsearch data from `production.example.net` instance authenticated with user and password `myuser:mypass` to a new Morpheus instance with IP address `162.252.108.123` running on port `12291`.

```
// Move Elasticsearch data from old production to new Morpheus
instance

node exporter.js -a production.example.net -A myuser:mypass -b
162.252.108.123:12291
```

See Also
The Elasticsearch-Exporter script and documentation is found at
https://github.com/mallocator/Elasticsearch-Exporter

For official Elasticsearch documentation, see http://www.elasticsearch.org/guide/

Appendix A—Terms and Acronyms

Below is a list of terms and acronyms you may encounter working with the Morpheus Appliance or Cloud Database site.

Term or Acronym	Meaning
ACL	A Morpheus Access Control List (ACL) refers to a white list of IP addresses for machines and descriptions permitted to connect to a Morpheus database instance in the cloud.
Appliance	Generally speaking, a virtual appliance is a virtual machine image file consisting of a pre-configured operating system environment and a single application such as a Java Virtual Machine (JVM). In the case of Morpheus, this is a database provisioning and management application that simplifies delivery and operation of an application. To this end, only necessary operating system components are included. The Morpheus appliance provides single-click provisioning of SQL, NoSQL, and/or in-memory databases on your own private or public clouds such as Amazon Web Services or Rackspace. (Is this true? Can the spin up a Database on their server?) Am I getting too down in the weeds here?
API Key	The API key appears on the Morpheus instance page and acts as both a unique identifier and a secret token for authentication which can be submitted alongside web service (or similar) requests in order to identify the origin of the request. NOTE: Per Jordan this is a feature coming in the future.
CIDR	CIDR stands for Classless Inter-Domain Routing, and refers to the standard of dividing the entire IP address space into smaller networks of variable size.
CIDR Notation	CIDR notation is a syntax for specifying IP addresses and their associated routing prefix. Use CIDR notation when adding an IP address to your Morpheus ACL. Enter the IP address then append a slash character and the decimal number of leading bits of the routing prefix. This is an IPv4 example: 192.168.2.0/24 NOTE: Morpheus only uses IPv4 at this time.  TIP: Use http://www.whatismyip.com/ to look up you machine's

	IP address. Be sure to add the slash followed the bit mask for the network. The example above shows /24 for a single machine.
CLI	The Command Line Interface is a basic way to interact with operating systems or databases commonly used by users, administrators, and programmers. Also referred to as a command line shell or utility.
Cloud computing	At the foundation of cloud computing is the broader concept of converged infrastructure and shared services to maximize the effectiveness of the shared resources.
Cluster	A database cluster is a collection of databases managed by a single instance of a running database server.
Collection	A grouping of MongoDB documents. A collection is the equivalent of an RDBMS table. A collection exists within a single database. Collections do not enforce a schema. Documents within a collection can have different fields. Typically, all documents in a collection have a similar or related purpose.
Database	A collection of files that either store application data or are used by the instance to store information that the database instance needs to operate. A database is a collection of information organized to provide efficient retrieval. The collected information could be in any number of formats (electronic, printed, graphic, audio, statistical, combinations). There are physical (paper/print) and electronic databases. A database could be as simple as an alphabetical arrangement of names in an address book or as complex as a database that provides information in a combination of formats.
Database instance	A database instance is an isolated database environment in the cloud. A DB instance can contain multiple user-created databases and be accessed using the same tools and applications you currently use to connect to a stand-alone database. An application program must connect to and interact with the database instance in order to insert, retrieve, modify, and/or delete data stored in the database.
Database name	A database is identified by the database name you give it.
DBaaS	Database as a Service (DBaaS) is a cloud-based approach to the storage and management of structured data. DBaaS delivers database functionality similar to a locally database but is hosted in the cloud.
Elasticsearch	The Elasticsearch database provides a distributed, multitenant-capable full-text search engine with a RESTful web interface and schema-free JSON documents. Elasticsearch is developed in Java and is released as open source under the terms of the Apache License used to dramatically improve search functionality across vast fields of data. Elasticsearch is used by Klout, StumbleUpon, Xing, and GitHub.

IPv4	Internet Protocol version 4 is the fourth version in the development of the Internet Protocol (IP) Internet, and routes most traffic on the Internet. A successor protocol, IPv6, has been defined and intermittently deployed. Morpheus only uses IPv4 as of this writing.
MongoDB	MongoDB (from "humongous") is a free, open-source, cross-platform document-oriented database. Classified as a NoSQL database, MongoDB eschews the traditional table-based relational database structure in favor of JSON-like documents with dynamic schemas (MongoDB calls the format BSON), making the integration of data in certain types of applications easier and faster. MongoDB has been adopted as backend software by a number of major websites and services, including Craigslist, eBay, Viacom, and the *New York Times*, among others. MongoDB is the most popular NoSQL database system.[2] MongoDB scales horizontally using sharding. The user chooses a shard key, which determines how the data in a collection will be distributed. The data is split into ranges (based on the shard key) and distributed across multiple shards. (A shard is a master with one or more slaves.) In MongoDB, a collection may store a number of documents and is analogous to a table of a RDBMS. A collection may store documents of varying structures because MongoDB is a *schema-free* database. MongoDB does not require such a set of formula defining structure of data. So, it is quite possible to store documents of varying structures in a collection. Practically, you don't need to define a column and it's datatype unlike in RDBMS, while working with MongoDB.
MySQL	MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack. LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python."
NoSQL database	NoSQL is a type of datastore; it's a way of storing and retrieving data quickly, much like a relational database except it isn't based on a relationship between tables as a traditional relational database does. NoSQL datastores can be much faster than relational databases. The growth of ultra large websites like Facebook and Google has led to the development of NoSQL as a way of breaking through speed constraints. Most work on storing and fetching Key-Values, though some, for example Redis, store large quantities of data entities, called documents, in XML or JSON formats.
Persistence layer	Any software layer that makes it easier for a program to persist its state is generically called a persistence layer. Most persistence layers will not achieve persistence directly but will use an underlying database management system that provides, not only persistence, but

	also other services such as queries, auditing, and access control.
Redis	Redis is an open-source, networked, in-memory, key-value data store with optional durability. It is written in ANSI C. Every Morpheus Redis instance comes with: • Cache Behavior Configuration • Key Support for Strings, Hashes, Lists, Sets, & Sorted Sets • Disk-Dump Config Control For In-Memory Dataset • Performance monitoring • Single click provisioning
Schema	In a relational database like MySQL, a schema defines the organization / structure of data in database. A database schema is a collection of meta-data and described as the "layout" of a database. In other words, the schema is the "blueprint" that outlines the way data is organized into tables.
Site	What is a site for Morpheus purposes?
Sharding	Sharding, or horizontal scaling, divides the data set and distributes the data over multiple servers, or shards. Each shard is an independent database, and collectively, the shards make up a single logical database.
Subnet	In Internet Protocol (IP) networking, devices on a subnet share contiguous ranges of IP address numbers.
Subnet mask	A subnet mask (or network mask) defines the boundaries of an IP subnet. The correspondence between subnet masks and IP address ranges follows defined mathematical formulas. You can use a subnet calculator like http://mxtoolbox.com/subnetcalculator.aspx to calculate your subnet.
Virtual appliance	Generally, a virtual appliance is a virtual machine image file consisting of a pre-configured operating system environment and a single application. In the case of Morpheus, this is a database provisioning and management app. The purpose of a virtual appliance is to simplify delivery and operation of an application. To this end, only necessary operating system components are included.
White list	A white list is a list (like Morpheus's ACL) that allows only these people or machines to connect to this database instance. It's opposite is a black list, a list of people or machines not allowed to connect.