PUBLISHED: FEBRUARY 05 2020

Connect to remote MongoDB on AWS EC2 simply and securely via SSH tunnel

To connect to a remote MongoDB server running on AWS EC2 that only allows local connections you can use an *SSH tunnel*, also known as *SSH port forwarding*.

An SSH tunnel creates a connection between a port on your local machine and an address + port on the server, and *tunnels* the data to the server over SSH on port 22 so it doesn't get blocked by any firewalls or security groups (assuming SSH connections are allowed to the server). After the data reaches the SSH server it gets forwarded to the address + port you specified when you created the SSH tunnel. An important detail here is that the destination address + port are accessed from the context of the server, so `localhost` or `127.0.0.1` refer to the server machine on the destination side of the tunnel, not your local machine.

Create an SSH Tunnel to a remote MongoDB server

Here's an example command that I used to create an SSH tunnel between port `8000` on my local machine and the MongoDB port `27017` on an Ubuntu server running on AWS EC2.

```
ssh -i ~/my-aws-key.pem -N -f -L 8000:localhost:27017 ubuntu@ec2-54-252-168-106.ap-south
```

- `-i ~/my-aws-key.pem` specifies the path to the ssh private key file to use for authentication.
- `-N` instructs ssh not execute a remote command, so it won't open a remote shell on the server.
- `-f` instructs ssh to run in the background.
- `-L 8000:localhost:27017` tells ssh to connect your local (`L`) port `8000` to the address + port `localhost:27017` *on the server*, in other words to MongoDB running locally on the server.
- `ubuntu@ec2-54-252-168-106.ap-southeast-2.compute.amazonaws.com` is the username and address to the AWS EC2 server that I used for testing.

Connect Mongo Shell to remote MongoDB server

After creating an SSH tunnel with the above command you can connect to the remote mongodb server as if it were running on your local machine on port 8000 like this.

```
mongo --port 8000
```

Connect MongoDB Compass to remote server

To connect to the remote mongodb server from MongoDB Compass use the following connection string to connect through the SSH tunnel on port 8000.

```
mongodb://localhost:8000
```

Close SSH Tunnel to MongoDB

To close an SSH tunnel that is open in the background you can kill the background process with the command `kill -9 <process id>`

Run this command to find out the id of the process for the SSH tunnel (and all other 'ssh' processes').

```
ps aux | grep ssh
```

Then run this command to kill the process and close the SSH tunnel, replacing <process id> with the id of the actual process from the above list.

```
kill -9 <process id>
```

Setup MongoDB Server on AWS EC2

For instructions on how to quickly setup a production ready MongoDB + Node.js server on AWS EC2 see Setup Node.js + MongoDB Production Server on Ubuntu. You can also see this server setup on YouTube as part of another video tutorial which covers deploying a MEAN Stack application to AWS.

Subscribe or Follow Me For Updates

Subscribe to my YouTube channel or follow me on Twitter or GitHub to be notified when I post new content.

- Subscribe on YouTube at https://www.youtube.com/JasonWatmore
- Follow me on Twitter at https://twitter.com/jason_watmore
- Follow me on GitHub at https://github.com/cornflourblue
- Feed formats available: RSS, Atom, JSON

My Motorcycle Adventure Vlog

Other than coding I'm currently attempting to travel around Australia by motorcycle with my wife Tina on a pair of Royal Enfield Himalayans. You can follow our adventure on YouTube at https://www.youtube.com/TinaAndJason.
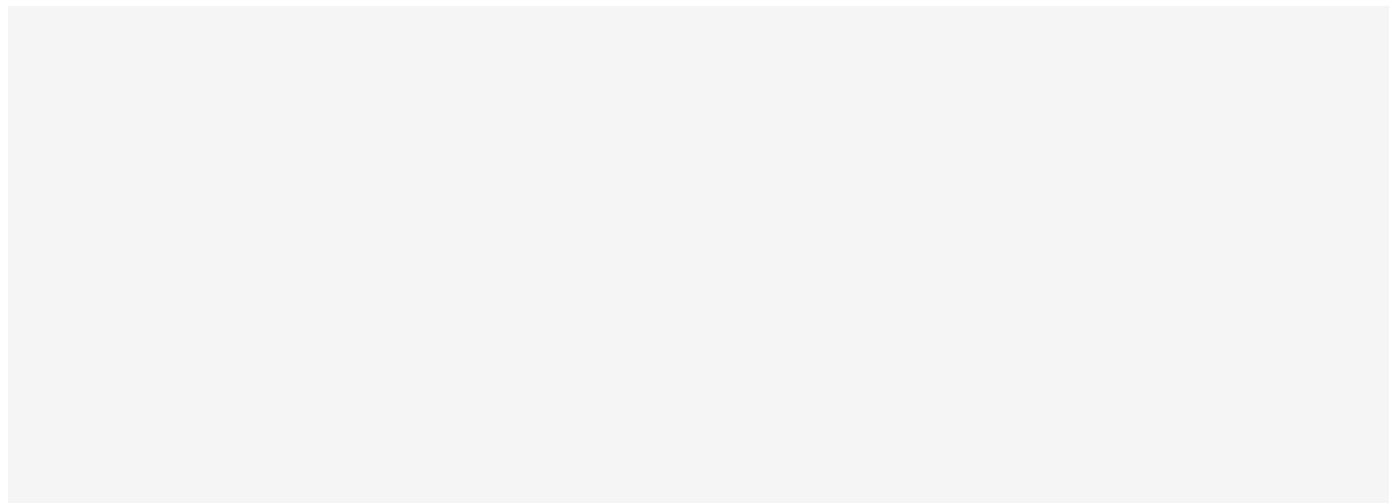
Tags: MongoDB, SSH, AWS

Share:

More MongoDB Posts

- Node.js + MongoDB API - JWT Authentication with Refresh Tokens
- Node + Mongo - Boilerplate API with Email Sign Up, Verification, Authentication & Forgot Password
- MongoDB - How to backup and restore a database
- Deploy to Heroku - Node + Mongo API for Authentication, Registration and User Management
- Setup Node.js + MongoDB Production Server on Ubuntu 18.04 - Ubuntu 19.04
- NodeJS + MongoDB - Simple API for Authentication, Registration and User Management
- MEAN with Angular 2/5 - User Registration and Login Example & Tutorial
- MEANie - Lightweight MEAN Stack CMS & Blogging Platform
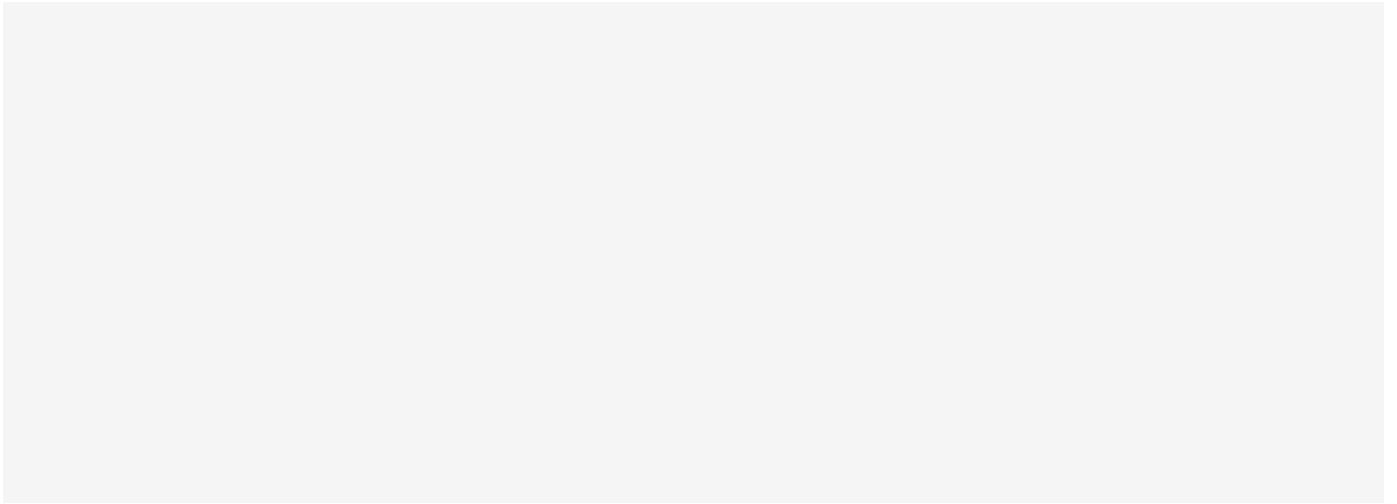
What do you think?

3 Responses

| 👍 Upvote | 😝 Funny | 😍 Love | 😮 Surprised | 🤧 Angry | 😢 Sad |

Join the discussion…
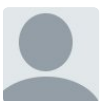
Talin • 3 months ago

HI @J**@Jason Watmore**
The MERN App configured in single EC2 instance is working fine,
But when
I have Installed them in separate instances REACT with nginx in public ,nodejs in private and mongodb in private subnets through NAT gate way,....When i try to register it shows the following error " Operation `users.findOne()` buffering timed out after 10000ms ",

⌃ | ⌄ • Reply • Share ›

Garg Vaibhav • a year ago

Hi @Jason Watmore,
My MERN app doesn't have a seperate webpackconfig.js
The server is as ./bin/www
And it uses two ports 4000 for http and 4443 for https.
The server is configured with Cors , corswithOptions(one with white list).
How should I proceed -
1.to intergrate these server ports with client on NGINX
2.configure a new secutiy group on amazon aws ec2 instance so https requests can be used with http

⌃ | ⌄ • Reply • Share ›

Sahil Bajaj • a year ago

Hi **@Jason Watmore** ,

Do we have to install mongo db on local for this to connect from local?

Thanks.

⌃ | ⌄ • Reply • Share ›

Jason Watmore `Mod` ↱ Sahil Bajaj • a year ago

Hi Sahil,

No you only need the mongo shell which is included when you install MongoDB but it can be installed separately as well - https://docs.mongodb.com/manual/mongo/#download-the-mongo-shell.

Cheers,
Jason

1 ∧ | ∨ · Reply · Share ›

Sahil Bajaj ➜ Jason Watmore · a year ago

Awesome. Thanks Jason for your quick response.

∧ | ∨ · Reply · Share ›

Dileep · a year ago

Hey Jason,
Can you tell me full step, One by one. to connect local system compass to aws server mongodb.

after running bellow code i m not understand what next to do.

ssh -i ~/my-aws-key.pem -N -f -L 8000:localhost:27017 ubuntu@ec2-54-252-168-106.ap-southeas... *with my key and my EC2

∧ | ∨ · Reply · Share ›

Sukh · a year ago

what is the path to ~/my-aws-key.pem , I'm not able to figure this out

∧ | ∨ · Reply · Share ›

Jason Watmore Mod ➜ Sukh · a year ago

Hi Sukh, on a mac the ~/ path points to the current user's directory e.g. /Users/jason

Cheers,
Jason

∧ | ∨ · Reply · Share ›

Pete Walker → Jason Watmore • a year ago

when I call

ssh -i ~/my-aws-key.pem -N -f -L 8000:localhost:27017 ubuntu@ec2-54-252-168-106.ap-southeas... *with my key and my EC2

I'm prompted with

bind [127.0.0.1]:8000: Address already in use
channel_setup_fwd_listener_tcpip: cannot listen to port: 8000
Could not request local forwarding.

see more

∧ | ∨ • Reply • Share ›

Jason Watmore Mod → Pete Walker • a year ago
Hi Pete, it looks like something is already running on port 8000 on your local machine, you can change it to any other port number (e.g. 8001, 8002, 9000 etc) and use that instead. Just remember to use the new port number when connecting the mongo shell as well.

Cheers,
Jason

see more

∧ | ∨ • Reply • Share ›

ABOUT

I'm a web developer in Sydney Australia and the technical lead at Point Blank Development, I've been building websites and web applications in Sydney since 1998.

Find me on:

Subscribe to Feed: RSS, Atom, JSON

Code Snippets

MONTHS

TAGS

Alerts, Angular, Angular 10, Angular 11, Angular 2, Angular 4, Angular 5, Angular 6, Angular 7, Angular 8, Angular 9, Angular Directive, Angular UI Router, AngularJS, Animation, ASP.NET, ASP.NET Core, ASP.NET Web API, Authentication and Authorization, AWS, Axios, Azure, Basic Authentication, Blazor, Bootstrap, C#, Chai, CKEditor, CSS3, DDD, Deployment, Design Patterns, Dynamic LINQ, EF Core, ELMAH, ES6, Exchange, Facebook, Fetch,

Fluent NHibernate, Formik, Google Analytics, Google API, Google Maps API, Google Plus, Heroku, HTML5, HTTP, IIS, Insecure Content, Instagram API, Ionic Framework, iOS, iPhone, JavaScript, jQuery, JWT, LinkedIn, LINQ, Login, MEAN Stack, MERN Stack, MEVN Stack, Mocha, Modal, MongoDB, Moq, MVC, MVC5, MySQL, .NET, NextJS, NGINX, ngMock, NHibernate, Ninject, NodeJS, npm, Pagination, Pinterest, Razor Pages, React, React Hooks, Redmine, Redux, Registration, Repository, RxJS, Security, Sequelize, Shell Scripting, Sinon, SinonJS, SSH, TDD, Terraform, Twitter, TypeScript, Ubuntu, Umbraco, Unit of Work, Unit Testing, URL Rewrite, Validation, VS Code, Vue, Vue 3, Vuex, Webpack, Windows Server 2008

Powered by MEANie