

Cloud and Web Applications

Steps for

Lab: Build a Course Material Sharer

PREFACE

Apply an Account for OpenShift

- Sign up a free account from <https://www.openshift.com/>

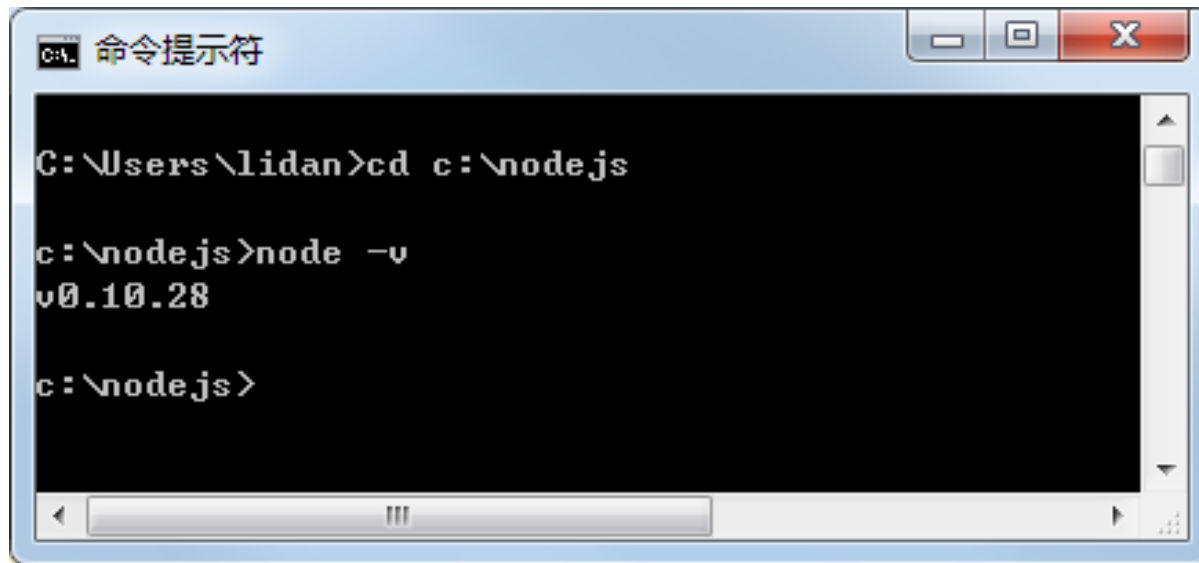
INSTALL TOOLS

Install and start MongoDB

- Create a new folder *c:\mongodb*, and install mongodb by running [*mongodb-win32-i386-2.6.3-signed.msi*](#) (from the flash mem) to that folder.
- Create a new folder *data* inside the folder *mongodb* to store data
- Copy file *startdb.bat* (from the flash mem) to the *mongodb* folder.
- Start MongoDB by running the *startdb.bat* in the *mongodb* folder.

Install and Test Node.js

- Create a new folder `c:\nodejs`
- Install Node.js by running : [node-v0.10.28-x86.msi](#) (from the flash mem) to folder `c:\nodejs`
- Open a cmd, Type `node -v` in the command line



A screenshot of a Windows Command Prompt window titled "命令提示符" (Command Prompt). The window shows the following commands and output:

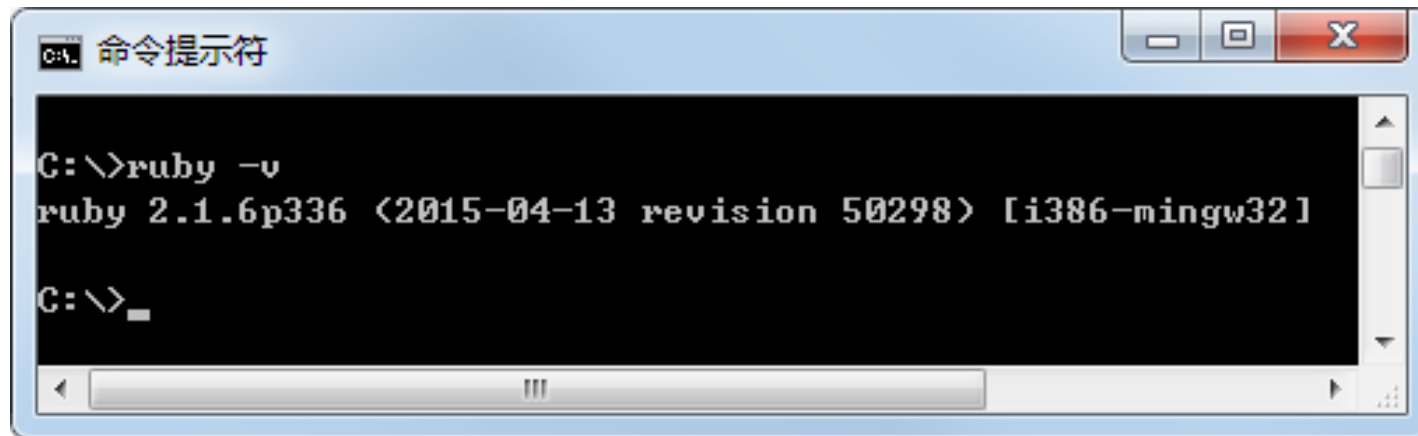
```
C:\Users\lidan>cd c:\nodejs

c:\nodejs>node -v
v0.10.28

c:\nodejs>
```

Install Ruby

- Running [rubyinstaller-2.1.6.exe](#) (from the flash mem) , select the **Add Ruby executables to your PATH** check box.
- Verify the installation:

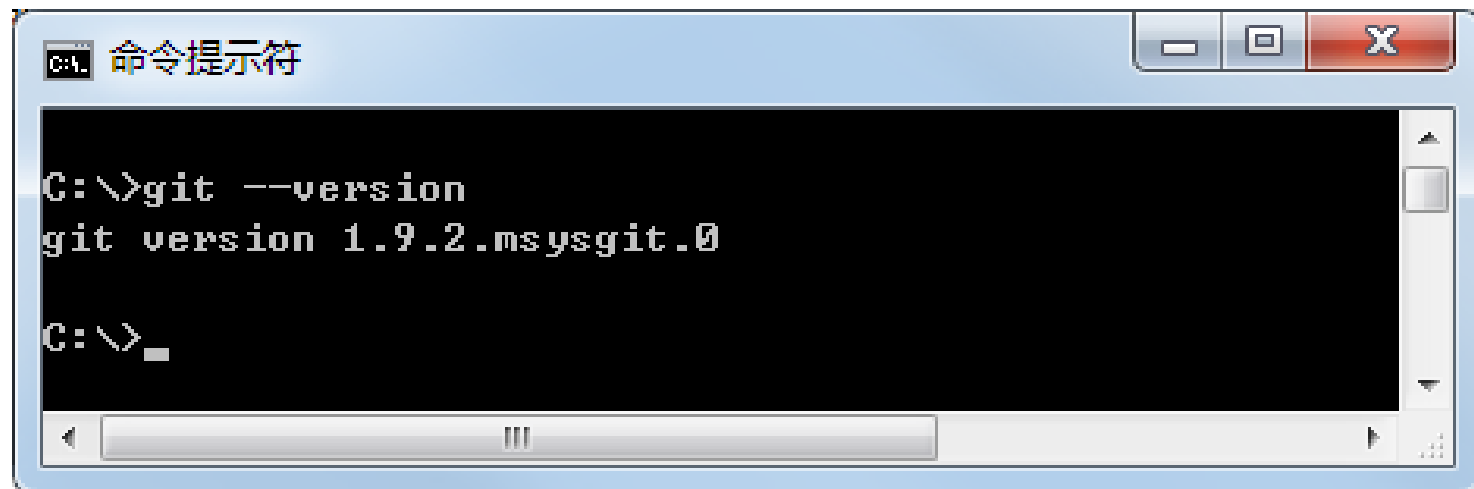


```
C:\>ruby -v
ruby 2.1.6p336 (2015-04-13 revision 50298) [i386-mingw32]

C:\>_
```

Install Git

- Running [Git-1.9.5-preview20150319.exe](#) (from the flash mem) , selecting the ***Run Git from the Windows Command Link Prompt*** checkbox, also, selecting **Checkout Windows-style, commit Unix-style line endings**.
- Verify the installation:



A screenshot of a Windows Command Prompt window titled "命令提示符" (Command Prompt). The window shows the command `C:\>git --version` being executed, followed by the output `git version 1.9.2.msysgit.0`. The prompt then shows `C:\>_` indicating the command has finished.

Install OpenShift gem

- Copy folder [rhcgems](#) from the flash mem to *c:*
- Open a cmd, cd to folder *c:\rhcgems*, and typing:

```
gem install rhc --local .\rhc-1.35.3.gem
```

* The gems package can also be downloaded from <http://pan.baidu.com/s/1qWLtjB2>

Config OpenShift Gem

- Open a cmd and run:

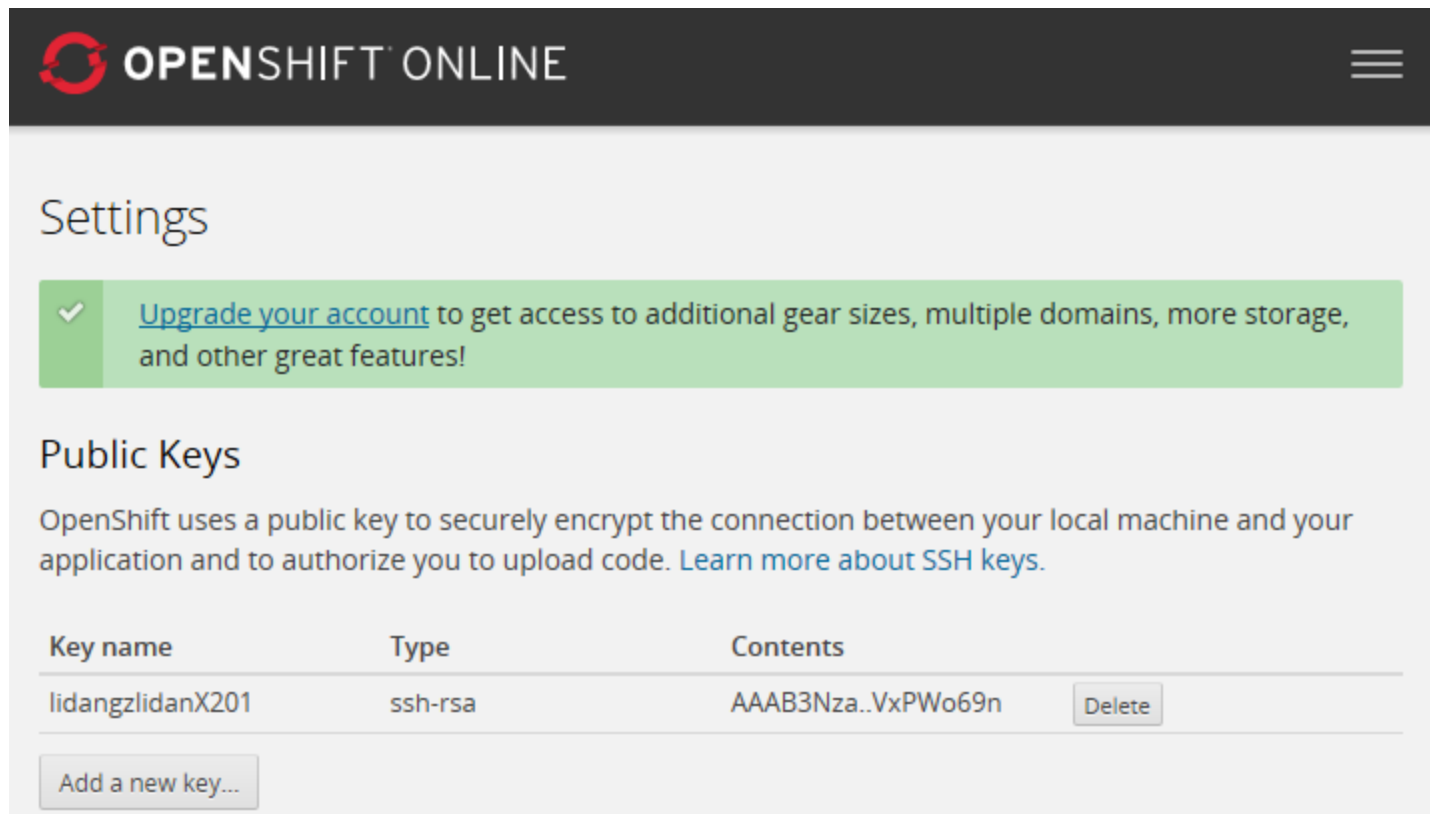
```
rhc setup
```

- Press enter to accept OpenShift server, and input OpenShift username and password;
- Answering **yes** to generate the SSH keys, and also **yes** to upload the public key.

CREATE OPENSIFT APP

Check SSH Key in OpenShift

- Login the web console (<https://www.openshift.com/>)
- Click on *Settings* to check the *Public Keys*.

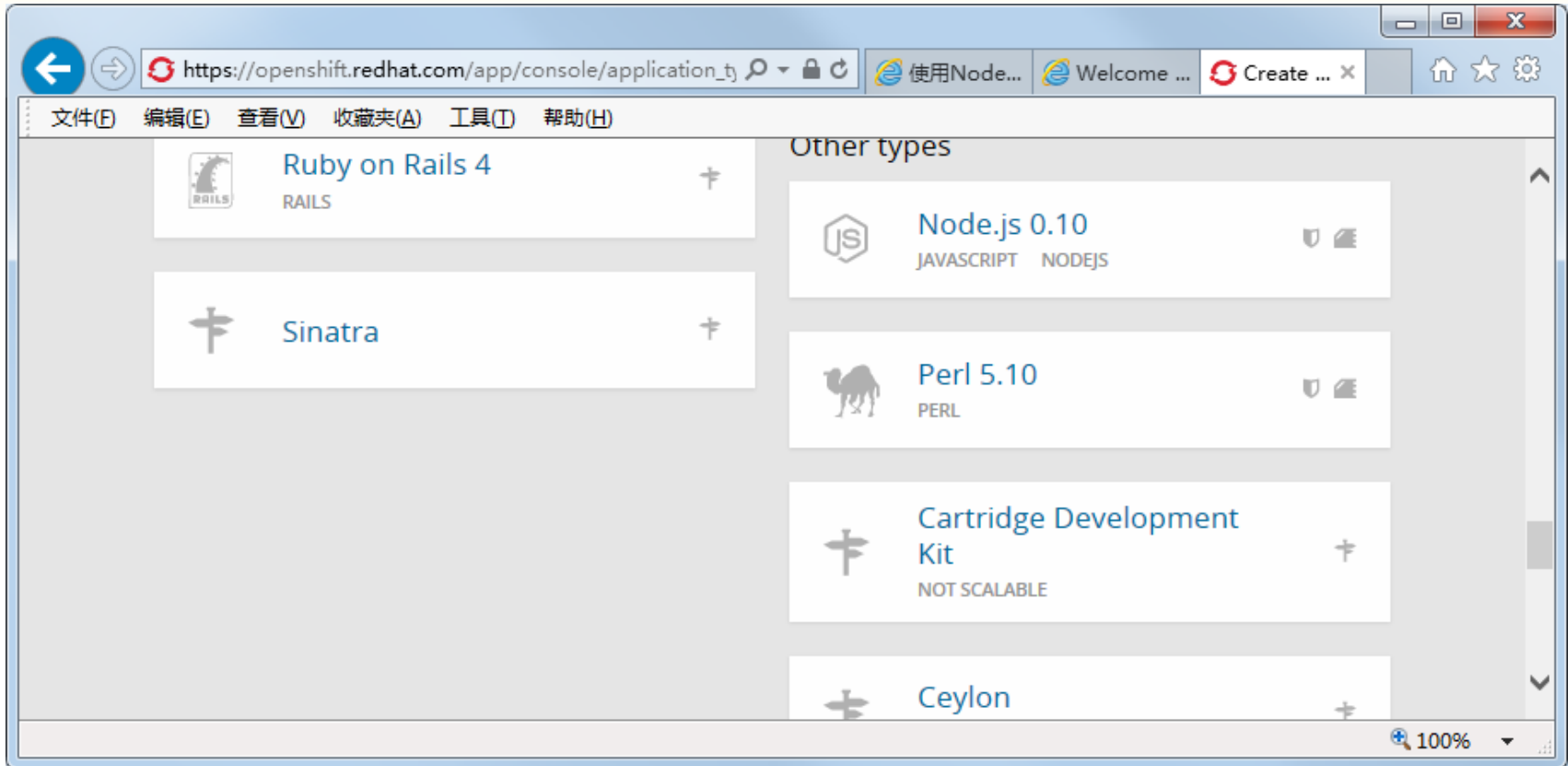


The screenshot shows the OpenShift Online web console interface. At the top, there's a dark header with the OpenShift logo and the text "OPENSHIFT ONLINE". Below the header, the "Settings" page is displayed. A green notification banner at the top of the settings page says "Upgrade your account to get access to additional gear sizes, multiple domains, more storage, and other great features!". Below this, the "Public Keys" section is visible. It contains a paragraph explaining that OpenShift uses a public key to securely encrypt the connection between the local machine and the application, and to authorize code uploads. Below the text is a table with columns for "Key name", "Type", and "Contents". There is one key listed with the name "lidangzlidanX201", type "ssh-rsa", and content "AAAB3Nza..VxPW69n". A "Delete" button is next to the key. At the bottom of the table, there is a button labeled "Add a new key...".

Key name	Type	Contents
lidangzlidanX201	ssh-rsa	AAAB3Nza..VxPW69n

Create an App

- Click on **Applications** and **Add Application**
- Select cartridge *Node.js 0.10*



- In the next page, input, for example “**mycourse**” to **Public URL**, and press **Create Application**

Add NongoDB to the App

- *Continue to the application overview page*
- Click [Add MongoDB 2.4](#), and [Add Cartridge](#) in next page

The screenshot shows the OpenShift Online interface. At the top, the header includes the OpenShift logo, the text "OPENSIFT ONLINE", and links for "Upgrade Plan" and a user profile "lidan_gz@163.com". Below the header is a navigation bar with "Applications", "Settings", and "Help". The main content area displays the application name "mycourse-lidangz.rhcloud.com" with a "change" link, the status "Started", and a gear icon. Below this, the "Cartridges" section shows a table with one cartridge: "Node.js 0.10" with status "Started", "1 small" gears, and "1 GB" storage. To the right of the cartridges is the "Source Code" section with an SSH URL and instructions to clone the repository. At the bottom left, the "Databases" section lists "Add MongoDB 2.4", "Add MySQL 5.5", and "Add PostgreSQL 9.2". To its right, the "Continuous Integration" section has a link to "Enable Jenkins". At the bottom right, the "Remote Access" section asks "Want to log in to your application?" and includes a "Delete this application..." button.

OPENSIFT ONLINE

Applications Settings Help OpenShift Hub

mycourse-lidangz.rhcloud.com change Started 1

Created 9 minutes ago in domain lidangz and the aws-us-east-1 region

Cartridges

	Status	Gears	Storage
Node.js 0.10	Started	1 small	1 GB

Source Code

ssh://5563d521e0b8cd9e7d0000a

Pass this URL to 'git clone' to copy the repository locally.

Remote Access

Want to log in to your application?

Delete this application...

Databases

- Add MongoDB 2.4
- Add MySQL 5.5
- Add PostgreSQL 9.2

Continuous Integration

- Enable Jenkins

App's url

Overview of your App.

ssh url

The screenshot shows the Heroku application overview page. At the top, the application name is 'mycourse-lidangz.rhcloud.com' with a 'change' link. Below it, it says 'Created 35 minutes ago in domain lidangz and the aws-us-east-1 region'. To the right, it says 'Started 1' with a gear icon and a refresh icon. Below this, there are two main sections: 'Cartridges' and 'Source Code'. The 'Cartridges' section lists two cartridges: 'Node.js 0.10' and 'MongoDB 2.4'. The 'MongoDB 2.4' cartridge has a 'Database: mycourse' field, a 'User: admin' field, and a 'Password: show' field. The 'Source Code' section shows an 'ssh://5563d521e0b8cd9e7d0000a' URL and a note 'Pass this URL to 'git clone' to copy the repository locally.' Below the 'Source Code' section, there is a 'Remote Access' section with a link 'Want to log in to your application?' and a 'Delete this application...' button. At the bottom, there are two sections: 'Continuous Integration' with a link 'Enable Jenkins' and 'Tools and Support' with links 'Add 10gen Mongo Monitoring Service Agent' and 'Add RockMongo 1.1'.

mycourse-lidangz.rhcloud.com [change](#)
Created 35 minutes ago in domain lidangz and the aws-us-east-1 region

Started 1

Cartridges

Icon	Name	Status	Gears	Storage
	Node.js 0.10	Started	1 small	1 GB
	MongoDB 2.4	Database: mycourse	User: admin	Password: show

Source Code

ssh://5563d521e0b8cd9e7d0000a

Pass this URL to 'git clone' to copy the repository locally.

Remote Access

[Want to log in to your application?](#)

[Delete this application...](#)

Continuous Integration

[Enable Jenkins](#)

Tools and Support

- [Add 10gen Mongo Monitoring Service Agent](#)
- [Add RockMongo 1.1](#)

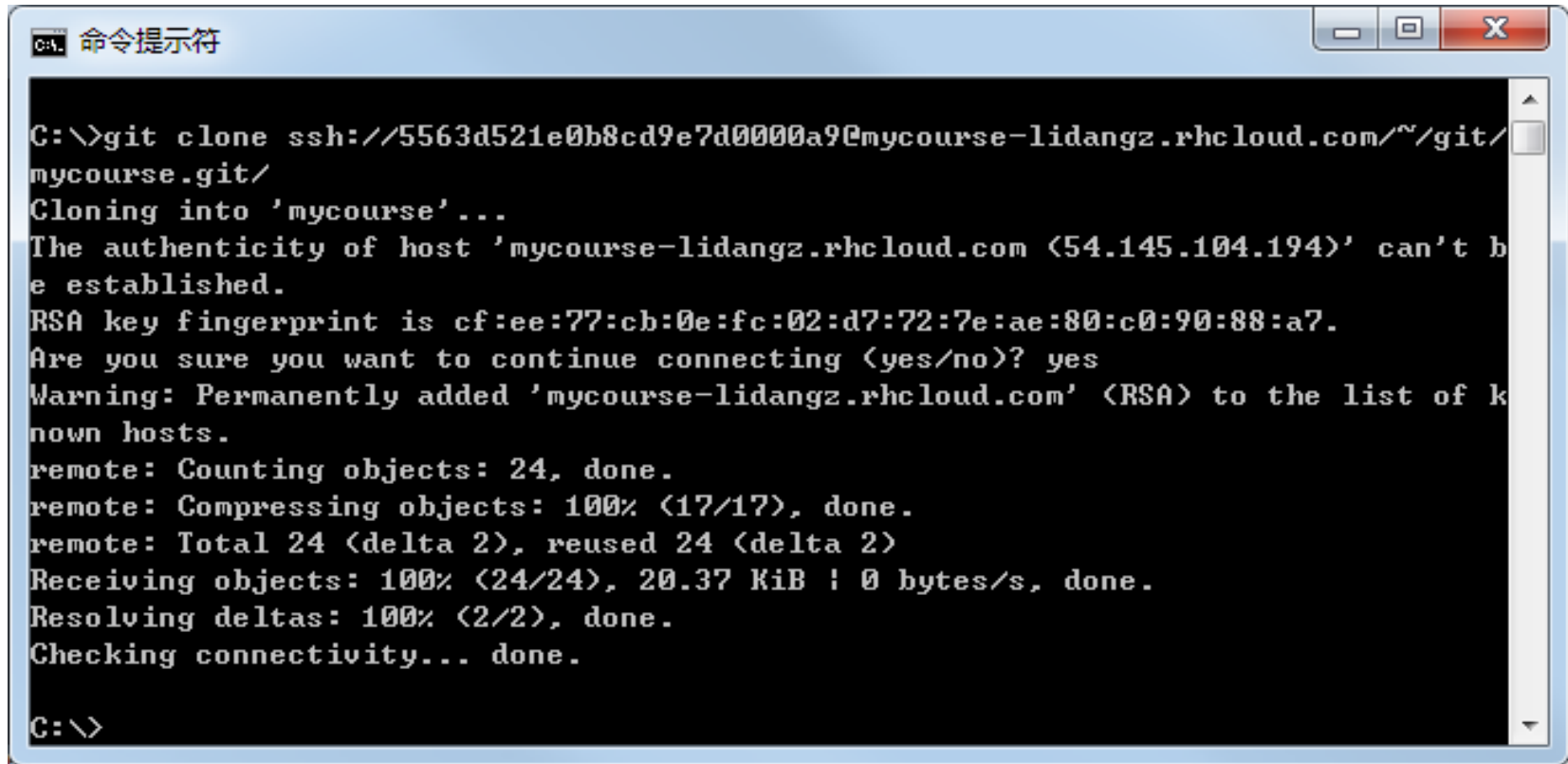
Name of your database

- Copy your `sshurl` under the **Source Code**

Download Skeleton Code

- Open a cmd, goto `c:\`, and run

```
git clone <sshUrl>
```



The screenshot shows a Windows Command Prompt window titled "命令提示符" (Command Prompt). The command entered is `git clone ssh://5563d521e0b8cd9e7d0000a9@mycourse-rhcloud.com/~git/mycourse.git/`. The output shows the cloning process, including a warning about the host's authenticity and the successful completion of the clone. The prompt ends at `C:\>`.

```
C:\>git clone ssh://5563d521e0b8cd9e7d0000a9@mycourse-rhcloud.com/~git/mycourse.git/
Cloning into 'mycourse'...
The authenticity of host 'mycourse-rhcloud.com (54.145.104.194)' can't be established.
RSA key fingerprint is cf:ee:77:cb:0e:fc:02:d7:72:7e:ae:80:c0:90:88:a7.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'mycourse-rhcloud.com' (RSA) to the list of known hosts.
remote: Counting objects: 24, done.
remote: Compressing objects: 100% (17/17), done.
remote: Total 24 (delta 2), reused 24 (delta 2)
Receiving objects: 100% (24/24), 20.37 KiB | 0 bytes/s, done.
Resolving deltas: 100% (2/2), done.
Checking connectivity... done.

C:\>
```

- Now, the `c:\mycourse` is your project folder.

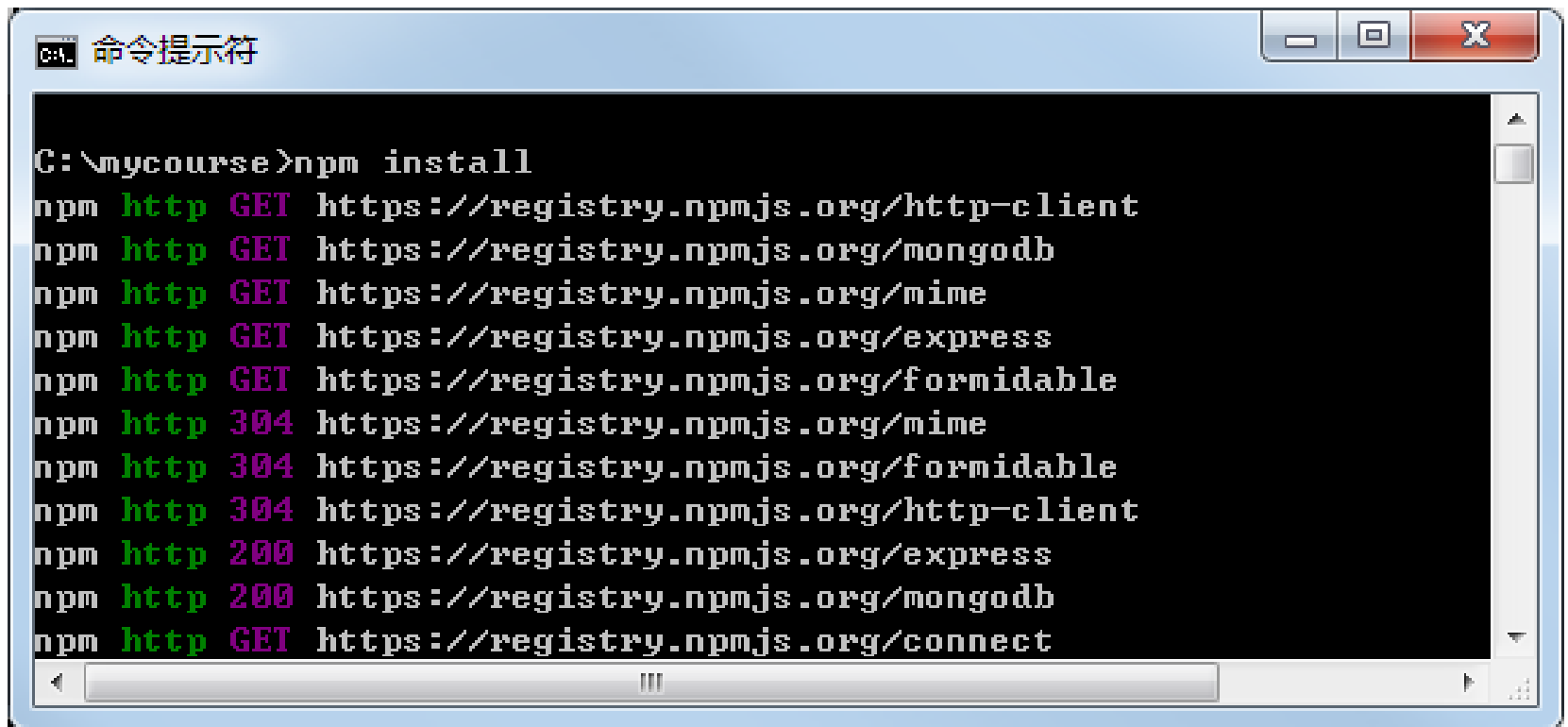
Copy App Source Code

- Copy all files from [Code_Part4_Lab.zip](#) (in the flash mem) to the project folder [c:\mycourse](#)
- Now the project folder has the following files.
 - Server-side
 - [server.js](#) -- app main entry, route the requests
 - [handlers.js](#) -- actually deal with the requests
 - [dbutils.js](#) -- database connection
 - [package.json](#) -- configuration file
 - Client-side
 - [Index.html](#) -- front-end html file

Install Dependencies

- In the cmd, goto the project folder `c:\mycourse`, run:

```
npm install
```



A screenshot of a Windows Command Prompt window titled "命令提示符". The window shows the execution of the command `npm install` in the directory `C:\mycourse`. The output displays a series of HTTP requests to the npm registry for various packages, including `http-client`, `mongodb`, `mime`, `express`, `formidable`, and `connect`. The status codes for these requests are shown in green, indicating successful installations or checks.

```
C:\mycourse>npm install
npm http GET https://registry.npmjs.org/http-client
npm http GET https://registry.npmjs.org/mongodb
npm http GET https://registry.npmjs.org/mime
npm http GET https://registry.npmjs.org/express
npm http GET https://registry.npmjs.org/formidable
npm http 304 https://registry.npmjs.org/mime
npm http 304 https://registry.npmjs.org/formidable
npm http 304 https://registry.npmjs.org/http-client
npm http 200 https://registry.npmjs.org/express
npm http 200 https://registry.npmjs.org/mongodb
npm http GET https://registry.npmjs.org/connect
```

Check the *dbutils.js*

```
var mongostr = {                                     // local machine
    "hostname": "localhost",
    "port": 27017,
    "username": "tom",
    "password": "1234",
    "name": "",
    "db": "course"
}

if(process.env.OPENSIFT_NODEJS_PORT){               // OpenShift
    mongostr = {
        "hostname": process.env.OPENSIFT_MONGODB_DB_HOST,
        "port": process.env.OPENSIFT_MONGODB_DB_PORT,
        "username": process.env.OPENSIFT_MONGODB_DB_USERNAME,
        "password": process.env.OPENSIFT_MONGODB_DB_PASSWORD,
        "name": "",
        "db": "mycourse"
    }
}

exports.getMongoUrl = function() {
    return "mongodb://" + mongostr.username + ":" +
        mongostr.password + "@" + mongostr.hostname + ":" +
        mongostr.port + "/" + mongostr.db;
}
```

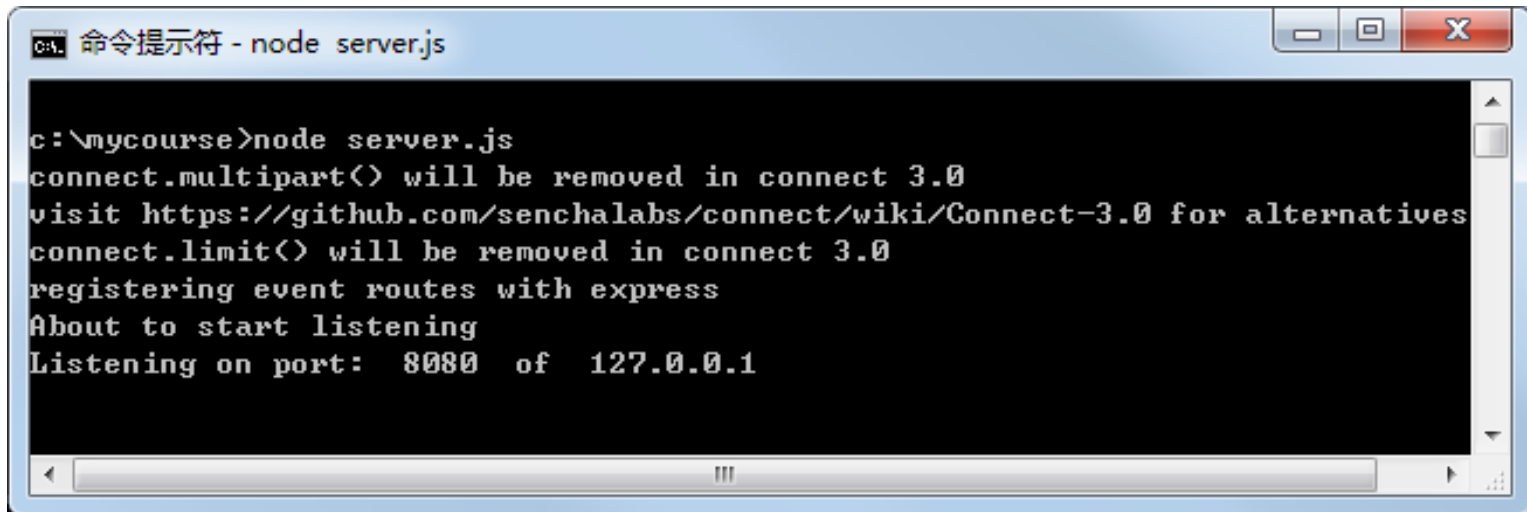
Change the name and password

Change to name of your database

TEST AND DEPLOY THE APP

Test the App in Local

- Be sure the mogondb is started
- Open a cmd, cd to the project folder, and start the Node.js server



```
c:\mycourse>node server.js
connect.multipart() will be removed in connect 3.0
visit https://github.com/senchalabs/connect/wiki/Connect-3.0 for alternatives
connect.limit() will be removed in connect 3.0
registering event routes with express
About to start listening
Listening on port:  8080  of  127.0.0.1
```

- Access <http://localhost:8080/> from an IE
- Check the log message showed by Nodejs

Upload code to OpenShift

- Open a cmd, cd to the project folder, runing

1. Add the two new js files to git control

```
git add handlers.js dbutils.js
```

2. Commit the changes to local git repository

```
git commit -am "My first change"
```

3. Push the code to OPENSHIFT server*

```
git push
```

* Try several times if errors

Test the App in OpenShift

- Open an IE, access you app's url using https

