

第9回

Rapid Online Game Coding Festival

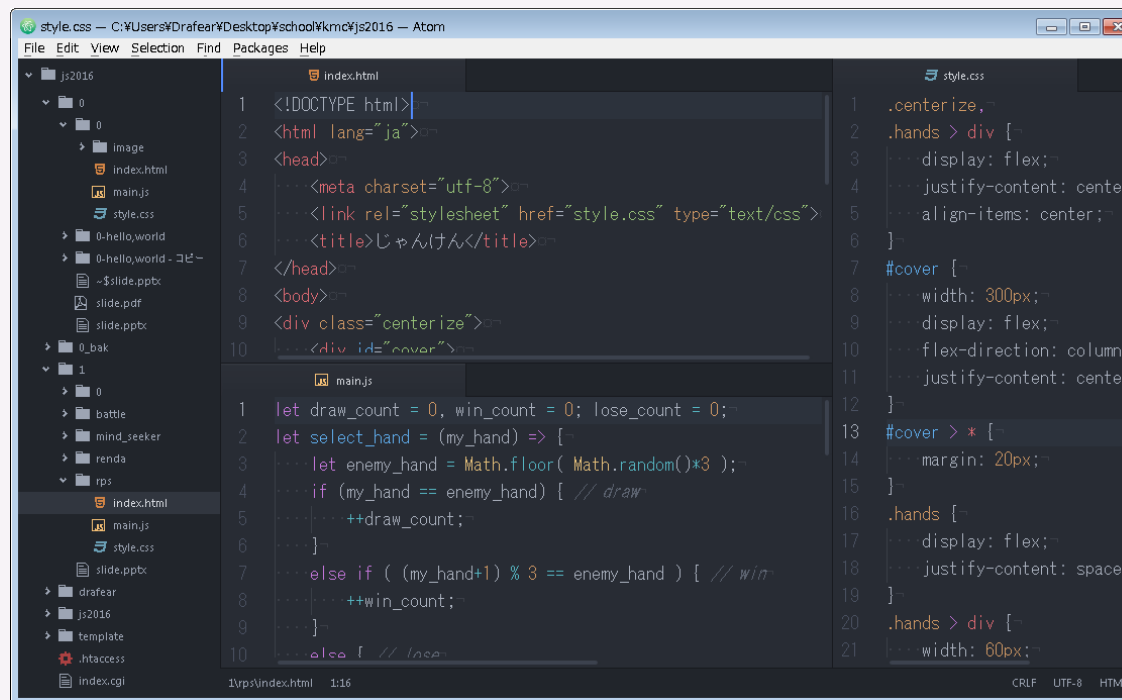
京都大学工学部情報学科
計算機科学コース3回
KMC2回 drafear



@drafear

この祭りで使用するブラウザとエディタ

- Google Chrome 
 - <https://chrome.google.com>
- Atom 
 - <https://atom.io/>



この祭りで使用する神アイテム

- Node.js (v6.2.2)
 - <https://nodejs.org/>



ほんじつ の ないよう

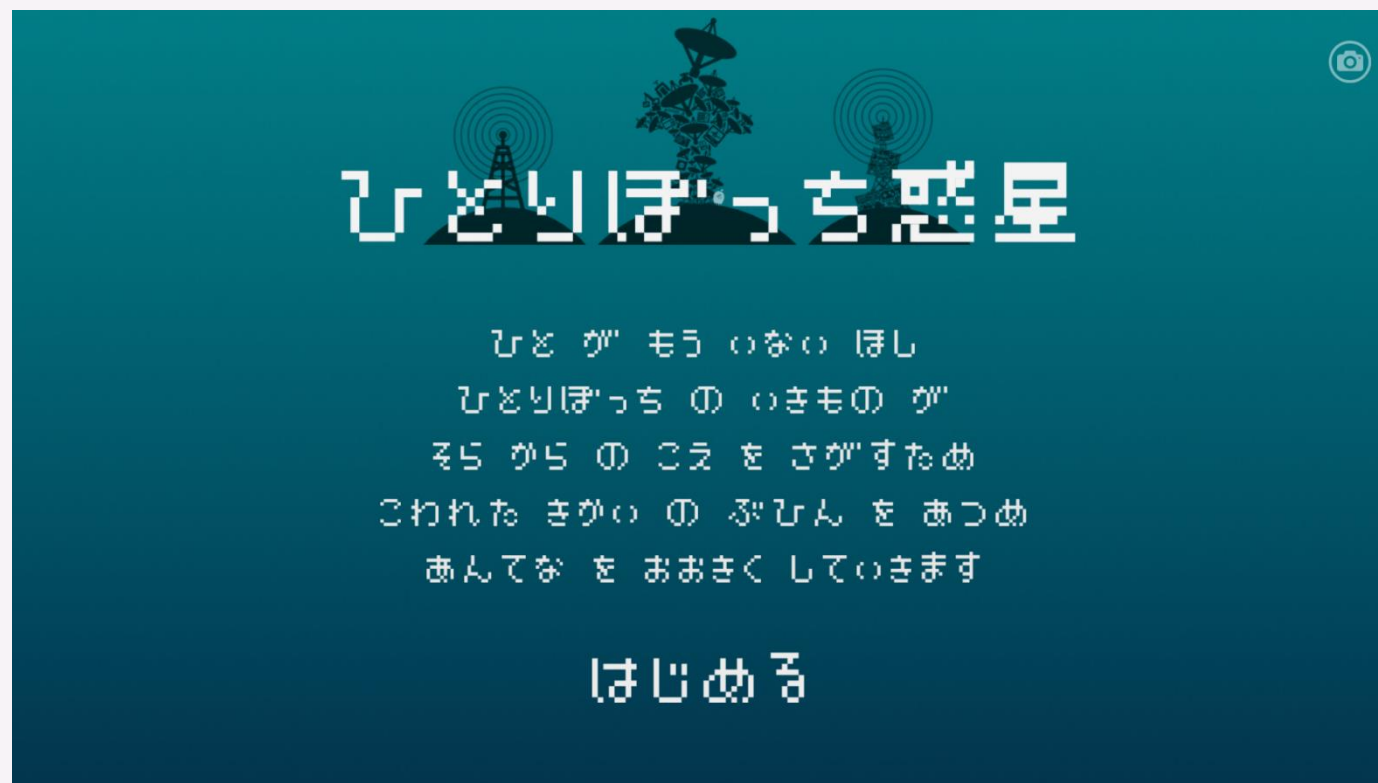
- おんらいんげーむ を つくる
- おんらいんげーむ を つくる
- おんらいんげーむ を つくる
- おんらいんげーむ を つくる
- おんらいんげーむ を つくる
- おんらいんげーむ を つくる
- おんらいんげーむ を つくる
- おんらいんげーむ を つくる
- おんらいんげーむ を つくる
- おんらいんげーむ を つくる
- おんらいんげーむ を つくる

ほんじつ の もくひょう

- のーど じゃーえす さいこう
- そけっと あいおー かみ
- おんらいんげーむ かんたんじゃね

ひらがな しか うてなく なった

ひとりぼっち惑星 のため
ひらがな しか うてません
ごりょうしょう ください





じゅんび

ひながた

- いつもどおり すらいど と てんぷれ を
だうんろーど して ください
 - えいちていーていーぴーえす ころん すら すら
ぎっとはぶ どっと こむ すら けーえむしー じえーぴー すら
じえーえす にせん じゅうろく すら
 - たんまつ の ある ひとは ぎっと くろーん で
 - ぎっと くろーん した こと がある ひとは ぎっと ぷる で
 - たんまつ の ない ひとは じっぷ で おとして
かいとう して ください

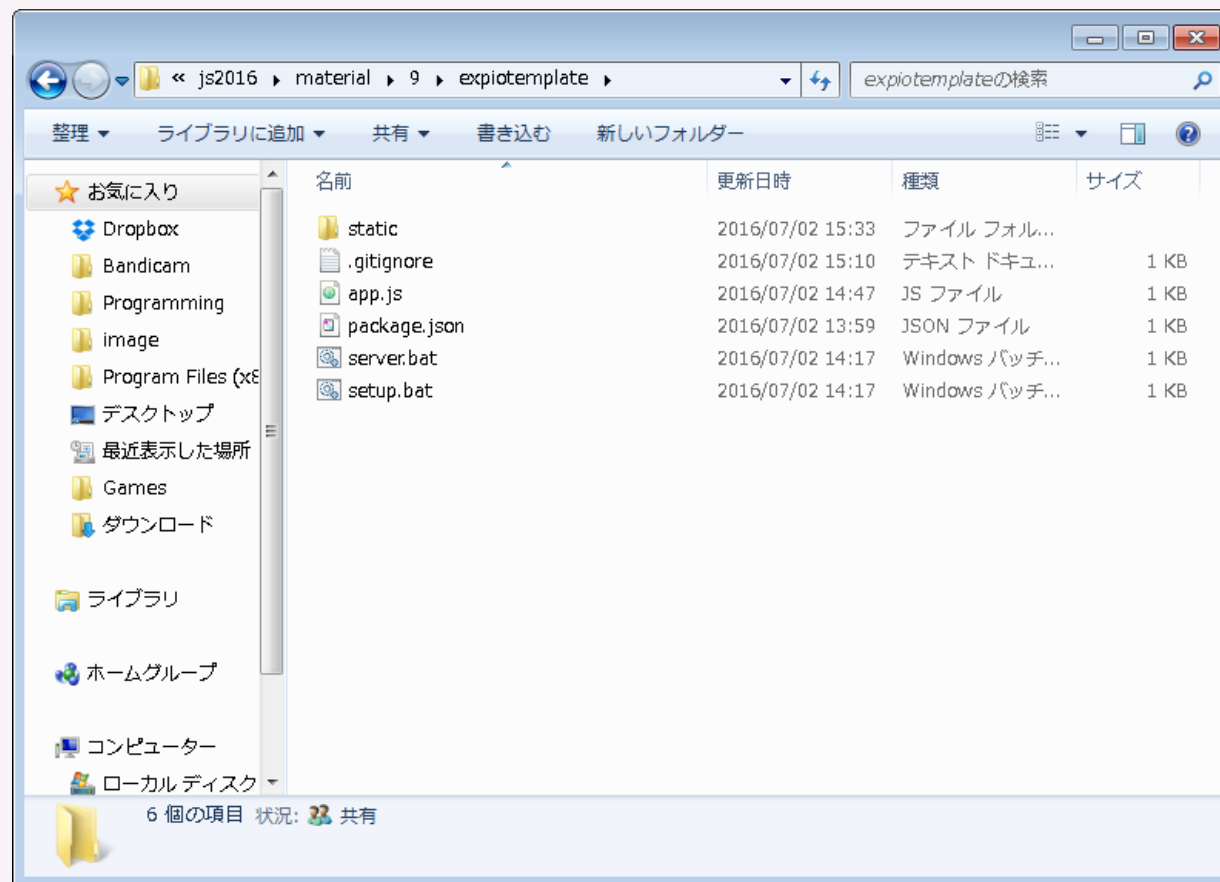
ごめんなさい

ヒガナタ

- いつもどおりスライドとテンプレをDL
 - <https://github.com/kmc-jp/js2016/>
 - 端末のある人は `git clone` で
 - `git clone` したことがある人は `git pull` で
 - 端末のない人は `zip` で

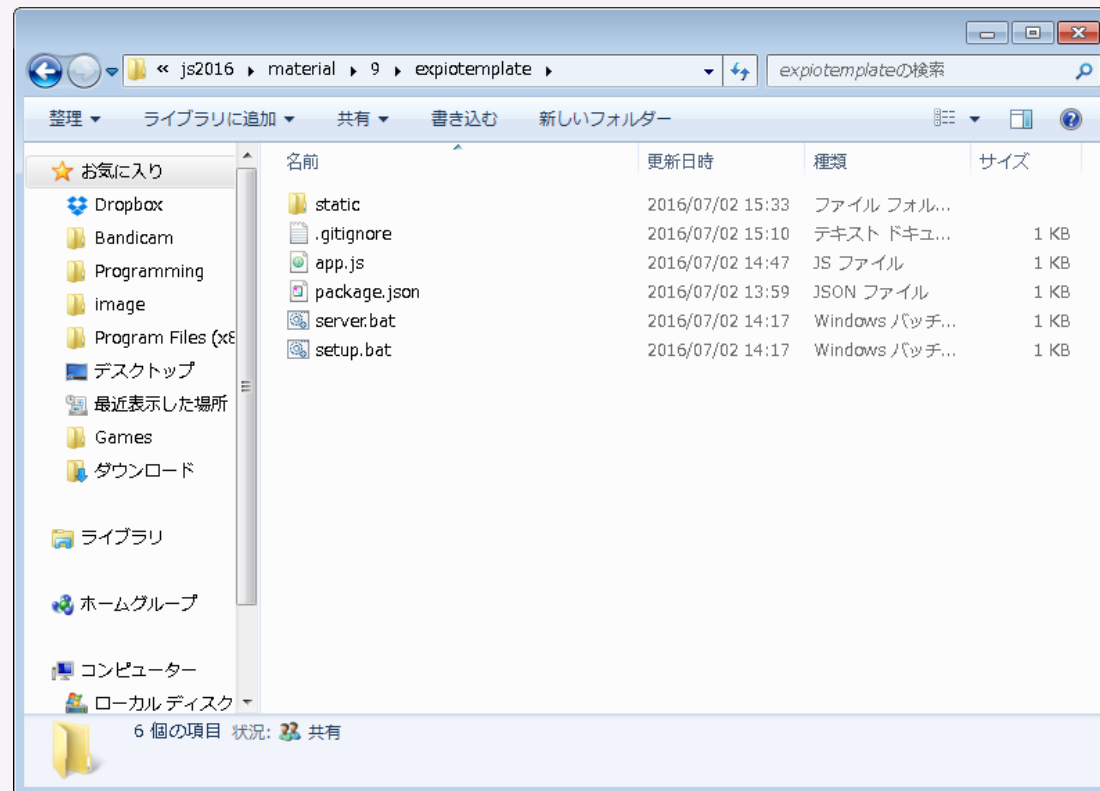
ヒガナタ ヒナガタ

- material/9/expiotemplate がテンプレートになります
- 適当なディレクトリにコピーして下さい



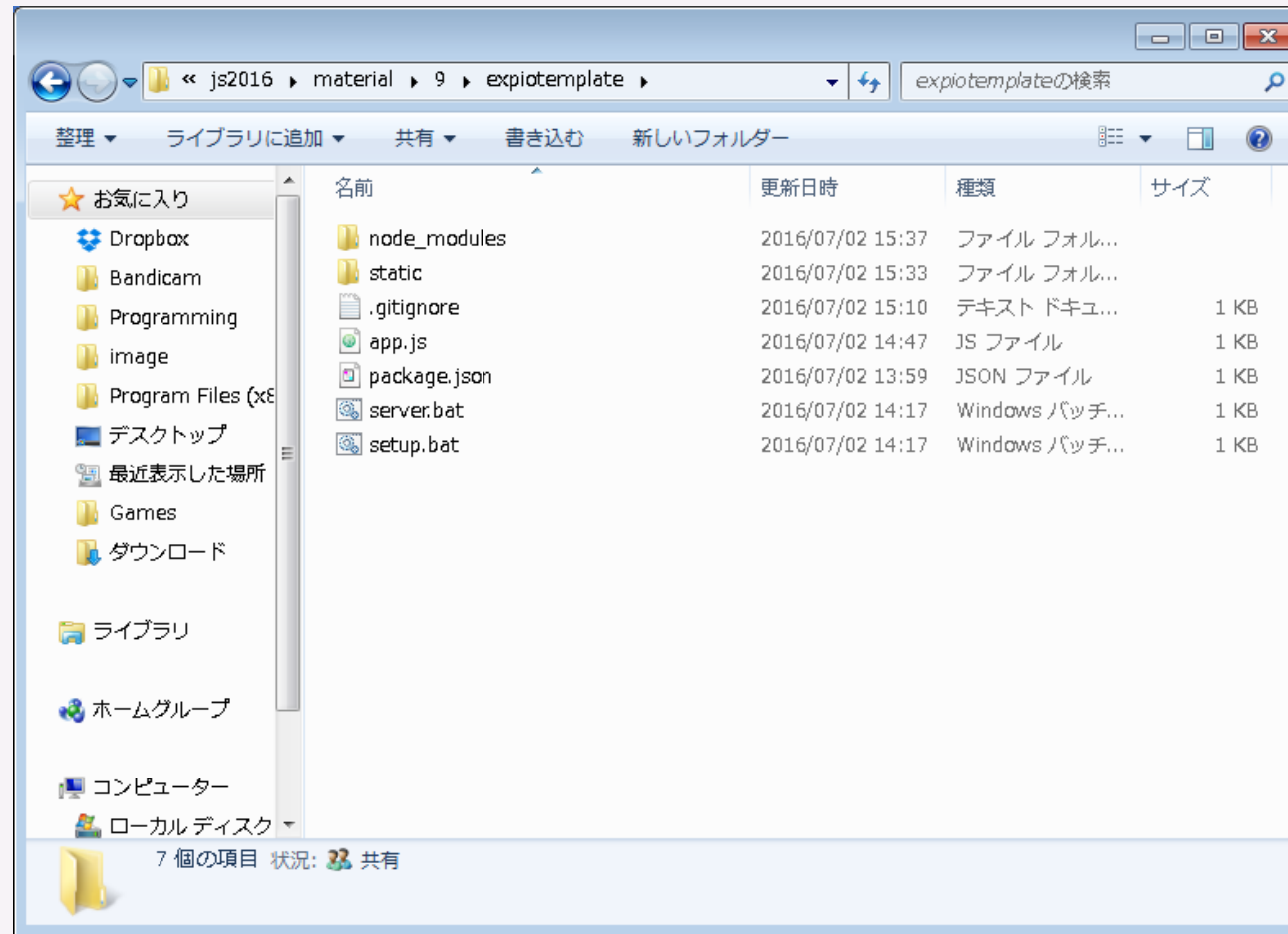
ら い ぶ ら り (socket.io, express) い れ る

- setup.bat を起動してライブラリをインストールしましょう
- 端末のある人は cd でこのディレクトリに移動して npm install でも可



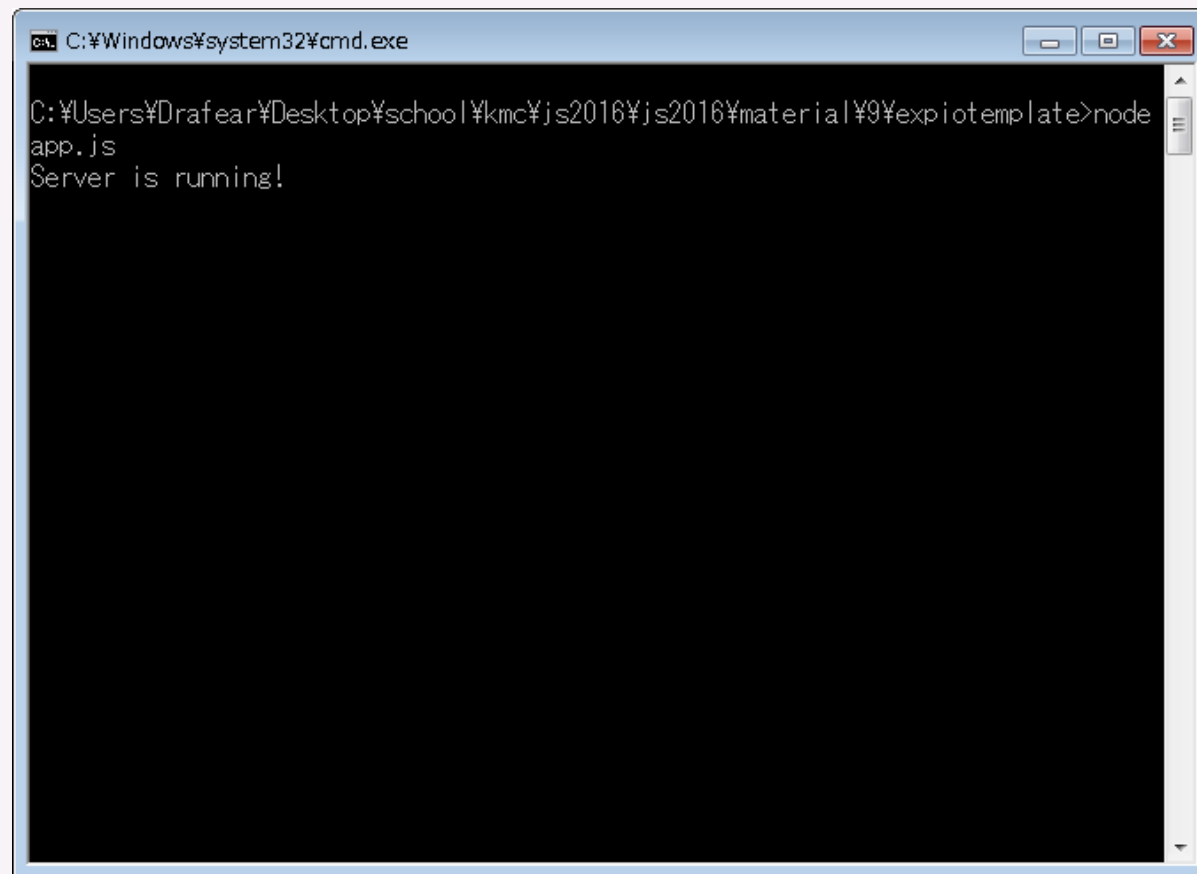
ら い ぶ ら り (socket.io, express) い れ る

- node_modules ってディレクトリができるハズ



サーバを起動してみる

- server.bat を起動すればサーバが起動
- 端末のある人は node app.js で

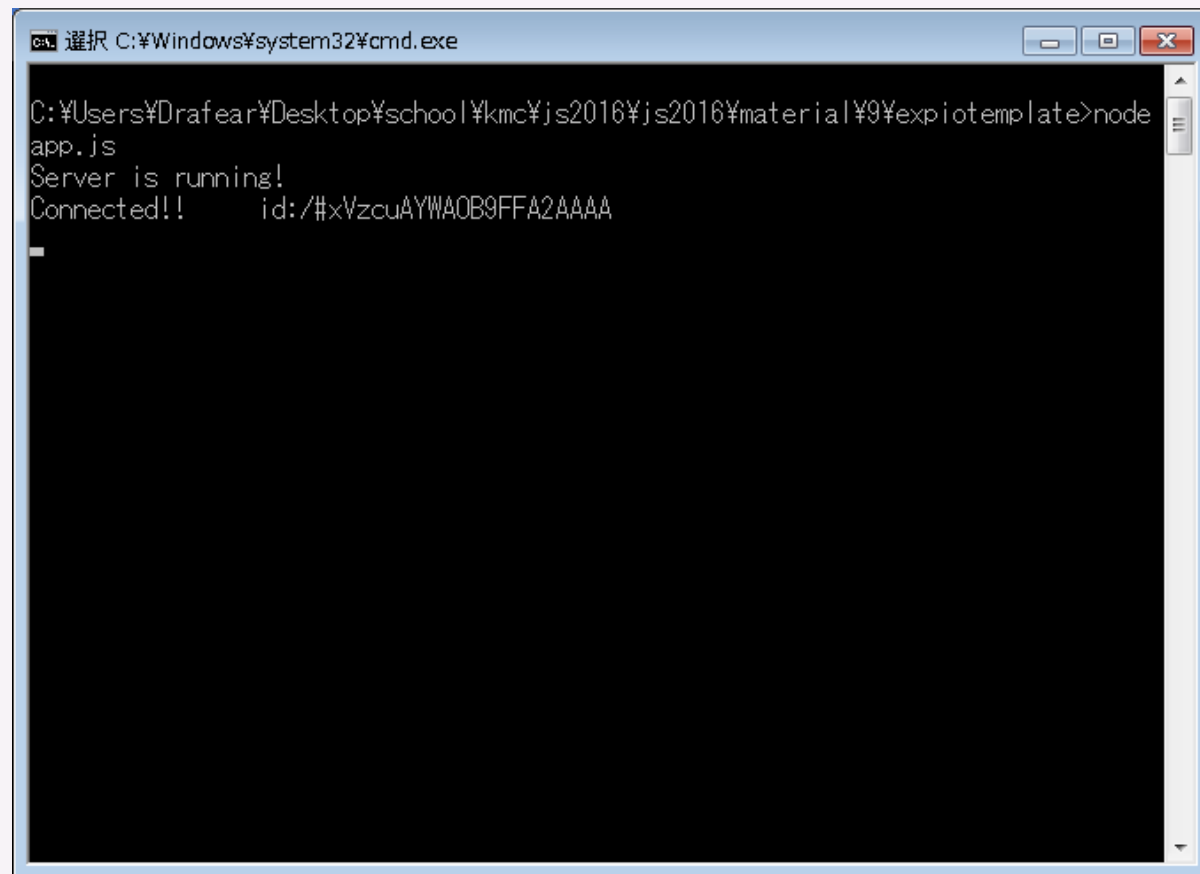


```
C:\Windows\system32\cmd.exe

C:\Users\Drafean\Desktop\school\kmc\js2016\js2016\material\9\expiotemplate>node
app.js
Server is running!
```

さーば を きどう してみる

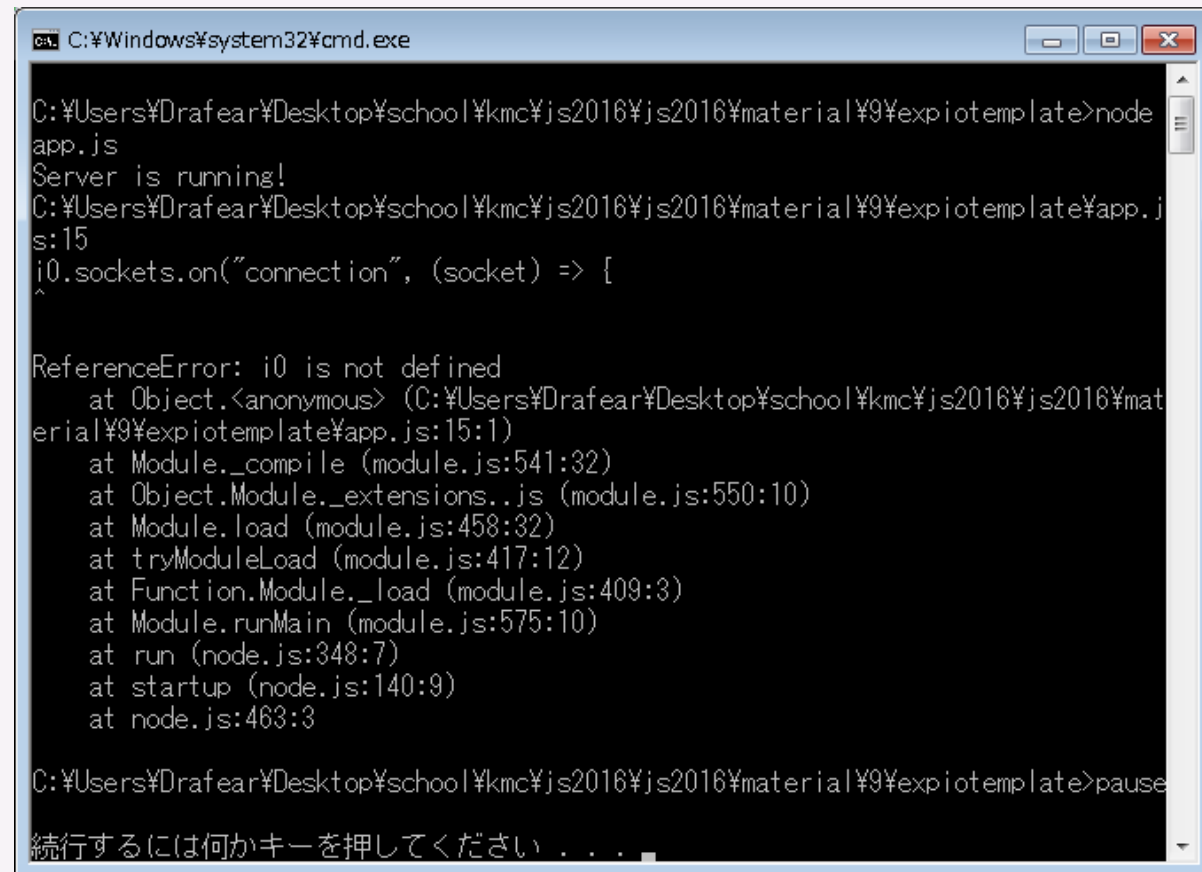
- 接続してみよう！
 - <http://localhost:8000/>



```
cmd 選択 C:\Windows\system32\cmd.exe
C:\Users\Drafeear\Desktop\school\kmc\js2016\js2016\material\9\expiotemplate>node
app.js
Server is running!
Connected!!      id:/#xVzcuAYWAQB9FFA2AAAA
```

さーば を きどう してみる

- エラーがあればココに出ます
- (何かキーを押すとサーバが再起動)



```
C:\Windows\system32\cmd.exe

C:\Users\Drafeard\Desktop\school\kmc\js2016\js2016\material\9\expiotemplate>node
app.js
Server is running!
C:\Users\Drafeard\Desktop\school\kmc\js2016\js2016\material\9\expiotemplate>app.j
s:15
i0.sockets.on("connection", (socket) => {
^
ReferenceError: i0 is not defined
    at Object.<anonymous> (C:\Users\Drafeard\Desktop\school\kmc\js2016\js2016\mat
erial\9\expiotemplate\app.js:15:1)
    at Module._compile (module.js:541:32)
    at Object.Module._extensions..js (module.js:550:10)
    at Module.load (module.js:458:32)
    at tryModuleLoad (module.js:417:12)
    at Function.Module._load (module.js:409:3)
    at Module.runMain (module.js:575:10)
    at run (node.js:348:7)
    at startup (node.js:140:9)
    at node.js:463:3

C:\Users\Drafeard\Desktop\school\kmc\js2016\js2016\material\9\expiotemplate>pause
続行するには何かキーを押してください . . .
```


サーバをきどうしてみる

- サーバを強制終了したいときは x ボタンで
- 端末を使ってる人は CTRL+C で

ディレクトリ構造

- 赤いファイルのみ編集していきます

```
.  
├── app.js (サーバ)  
├── package.json  
├── server.bat  
├── setup.bat  
└── static/ (クライアント)  
    ├── index.html  
    ├── lib/  
    │   └── socket.io.js  
    ├── main.js  
    └── style.css
```

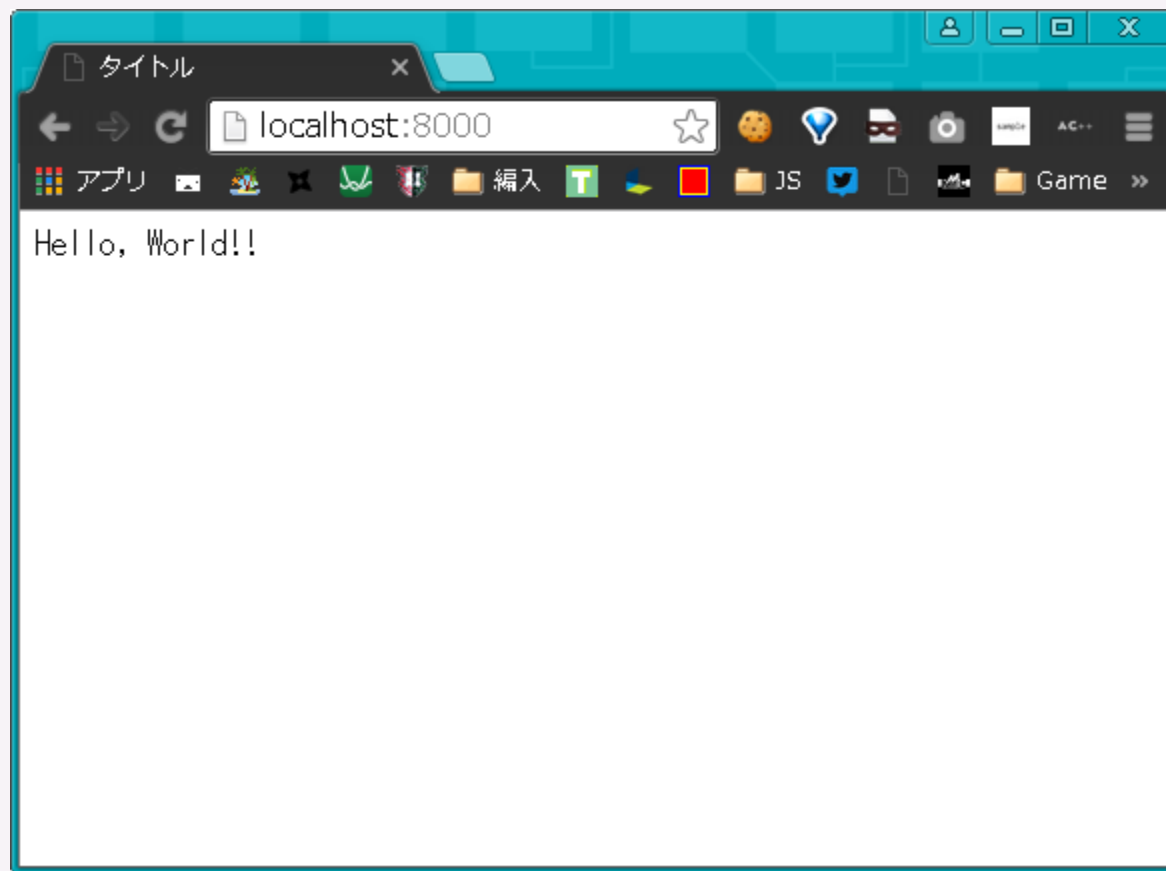
static #とは

- index.html を編集してもう一度アクセスしてみよう
 - <http://localhost:8000/>

index.html

Hello, World!!

static #とは



static #とは

- static以下に置いたファイルにアクセスできる
 - <http://localhost:8000/>
 - <http://localhost:8000/index.html>
 - <http://localhost:8000/main.js>
 - <http://localhost:8000/style.css>
- index.htmlは省略できる



オンラインゲームを作ろう

前回のまとめ

- クライアント側

- 送信: サーバに `data` データを乗せて `"message"` メッセージを送信する

```
socketio.emit("message", data);
```

- 受信: サーバから `"message"` メッセージを受信すると `(data) => { ... }` が実行される

```
socketio.on("message", (data) => { ... });
```

前回のまとめ

- サーバ側

- 送信1: 特定のクライアントにdataデータを乗せて
"message"メッセージを送信する

```
socket.emit("message", data);
```

- 送信2: 全てのクライアントにdataデータを乗せて
"message"メッセージを送信する

```
io.sockets.emit("message", data);
```


前回のまとめ

- サーバ側

- 送信3: "roomid" ルーム(後述)に居る全てのクライアントに data データを乗せて "message" メッセージを送信する

```
io.sockets.in("roomid").emit("message", data);
```

- 受信: 特定のクライアントから "message" メッセージを送信すると (data) => { ... } が実行される

```
socket.on("message", (data) => { ... });
```

前回のまとめ

- サーバ側
 - 部屋1: 特定のクライアントを"roomid"ルームに入れる

```
socket.join("roomid");
```

- 部屋2: 特定のクライアントを"roomid"ルームから出す

```
socket.leave("roomid");
```

オンラインゲームを作ろう

- 何かを作ろう
 - 三目並べ(マルバツ)
 - 大乱闘なんちゃら
 - しえるたん
 - ハイパーロボット
 - 連打ゲーム
 - Chrome拡張 + Node.js ! ?
 - マウスポインター鬼ごっこ
 - みんなでRPG



アプリ公開

アプリ公開

- 無料レンタルサーバ OpenShift を使います
 - <https://www.openshift.com/>



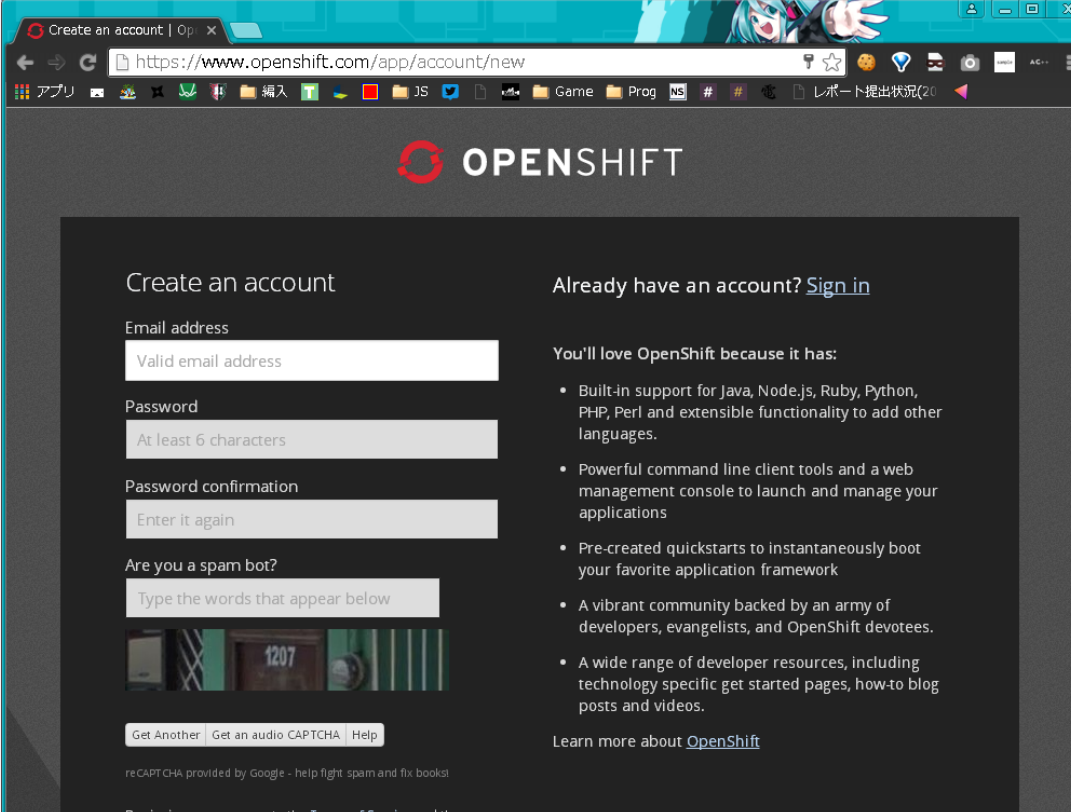
アプリ公開

1. 登録
2. Web Consoleへログイン
3. アプリ作成
4. 認証用の鍵作成
5. 認証用の鍵登録
6. `git clone`
7. ファイルの追加・削除

アプリ公開

1. 登録

- 右上の "SIGN UP FOR FREE" から登録（または以下URL）
- <https://www.openshift.com/app/account/new>



The screenshot shows the OpenShift 'Create an account' page. The browser address bar displays 'https://www.openshift.com/app/account/new'. The page features the OpenShift logo at the top. On the left, under 'Create an account', there are input fields for 'Email address' (with placeholder 'Valid email address'), 'Password' (with placeholder 'At least 6 characters'), and 'Password confirmation' (with placeholder 'Enter it again'). Below these is a CAPTCHA section titled 'Are you a spam bot?' with a text input 'Type the words that appear below' and a CAPTCHA image showing the number '1207'. At the bottom of the form are links for 'Get Another', 'Get an audio CAPTCHA', and 'Help'. On the right, under 'Already have an account? Sign in', there is a list of features: 'Built-in support for Java, Node.js, Ruby, Python, PHP, Perl and extensible functionality to add other languages.', 'Powerful command line client tools and a web management console to launch and manage your applications', 'Pre-created quickstarts to instantaneously boot your favorite application framework', 'A vibrant community backed by an army of developers, evangelists, and OpenShift devotees.', and 'A wide range of developer resources, including technology specific get started pages, how-to blog posts and videos.' At the bottom right, there is a link 'Learn more about OpenShift'.

アプリ公開

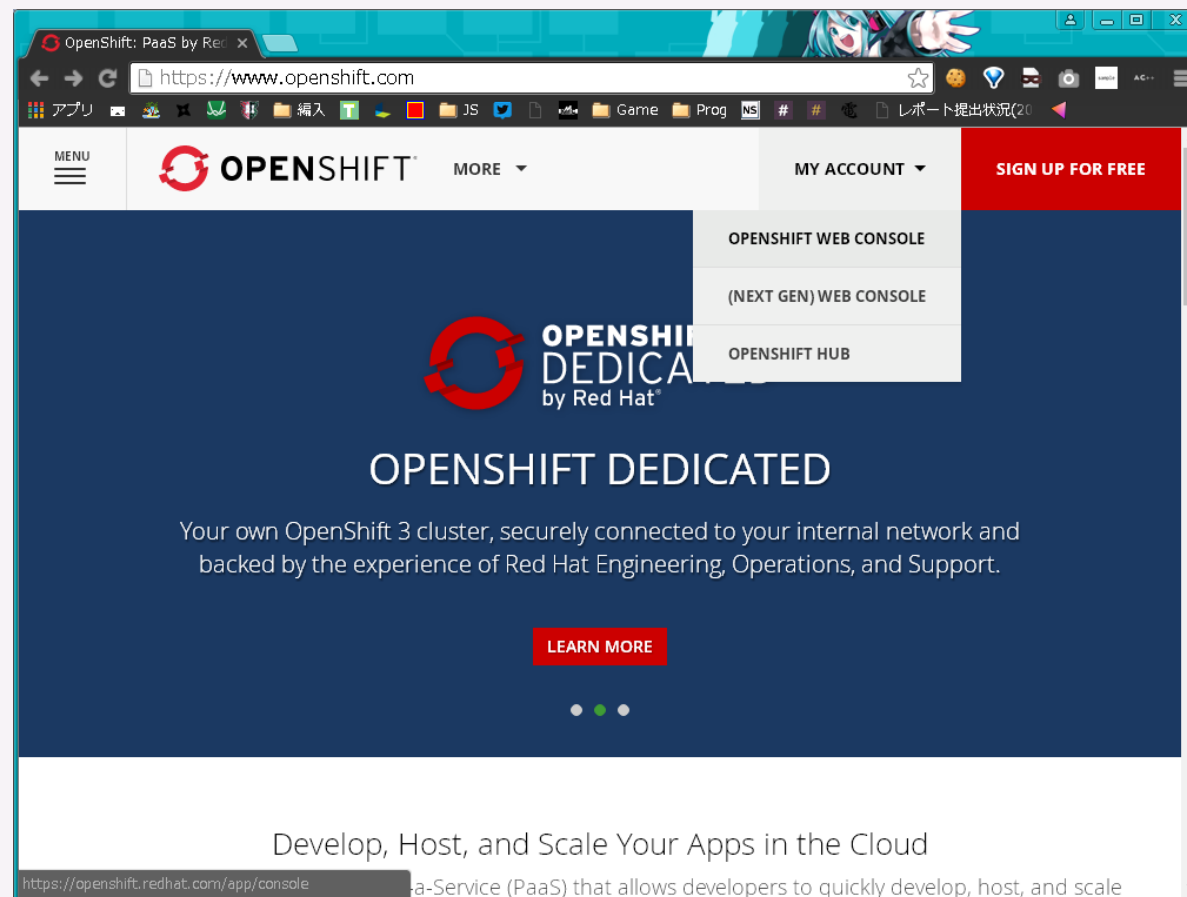
1. 登録

- Sign Up したらメールが届くはずなので Confirm する

アプリ公開

2. Web Console へログイン

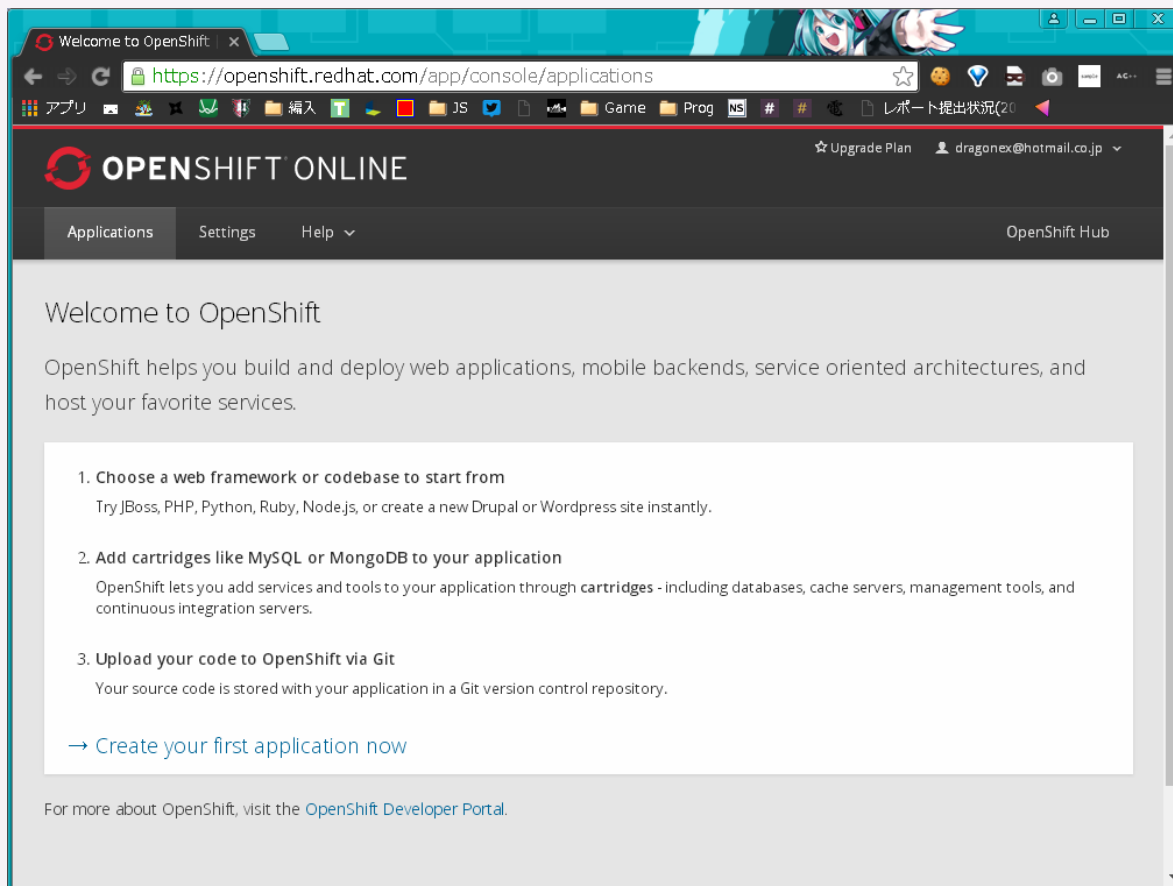
- トップページ右上 "MY ACCOUNT" → "OPENSHIFT WEB CONSOLE"



アプリ公開

3. アプリ作成

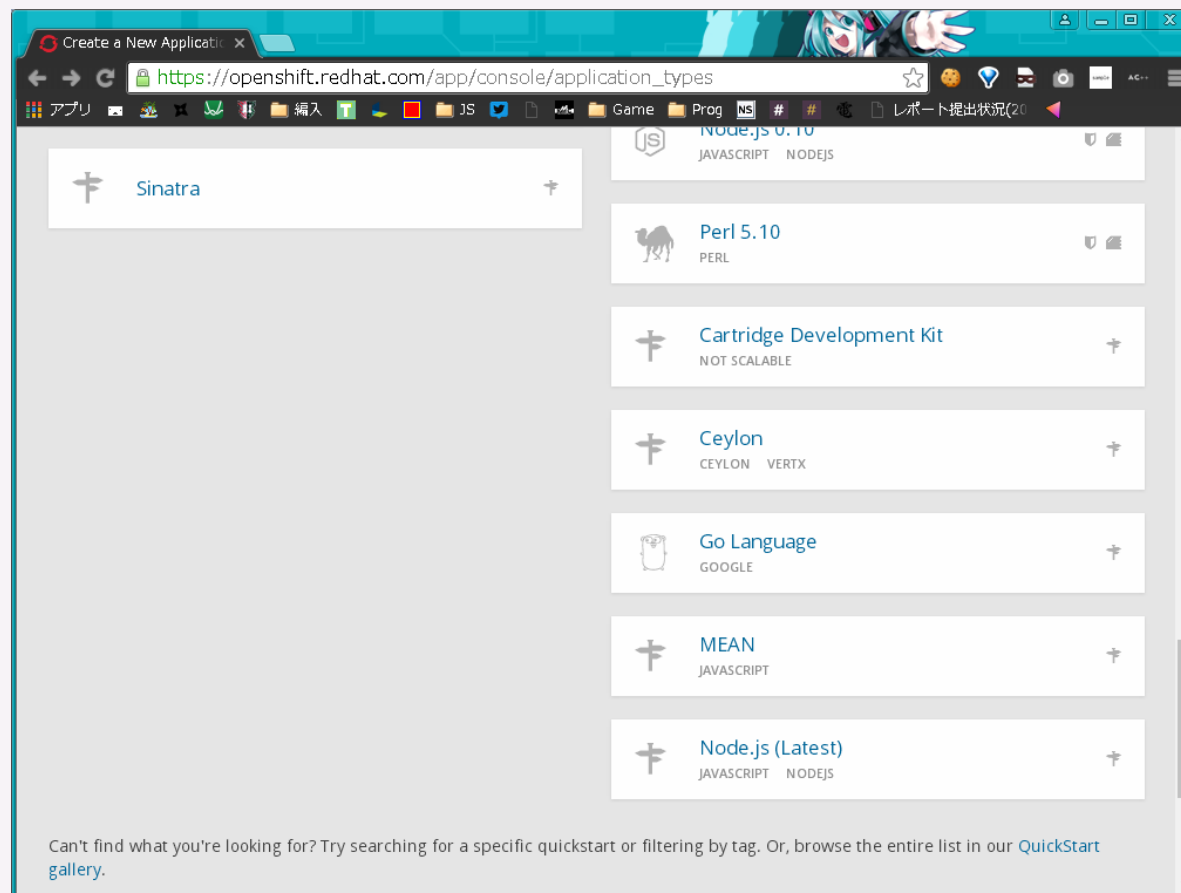
- "Create your first application now" よりアプリ作成画面へ



アプリ公開

3. アプリ作成

- "Node.js (Latest)" を選択



アプリ公開

3. アプリ作成

- **Public URL** のみ設定し, Create Application (作成には時間がかかるので作成中に次の手順を)

Create a New Application

https://openshift.redhat.com/app/console/application_type/quickstart!243

OPENSIFT ONLINE

Applications Settings Help

1 Choose a type of application 2 Configure the application 3 Next steps

Based On

Node.js (Latest) Quickstart

This custom Node.js cartridge takes care of auto-updating the Node.js and NPM versions on each build.

Node.js is a platform built on Chrome's JavaScript runtime for easily building fast, scalable network applications.

[Learn more](#)

Community created
Does not receive automatic security updates

Public URL

http:// sample - drafear .rhcloud.com

Because this is your first application, you need to provide a domain under which your applications will be grouped

OpenShift will automatically register this domain name for your application. You can add your own domain name later.

Source Code

Optional URL to a Git repository Branch/tag

Your application will start with an exact copy of the code and configuration provided in this

アプリ公開

4. 認証用の鍵作成

- 端末(ターミナル)を開いて以下のコマンドを実行して下さい

```
$ cd ~/.ssh
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/Drafear/.ssh/id_rsa): [何も入力せずEnter]
Enter passphrase (empty for no passphrase): [パスワードを入力. 画面上には表示されない]
Enter same passphrase again: [同じパスワードを入力(確認用)]
$ ls
id_rsa  id_rsa.pub
```

アプリ公開

5. 認証用の鍵登録

- Settingsより登録できる
- <https://openshift.redhat.com/app/console/settings>
- 以下のコマンドを実行すると鍵がクリップボードにコピーされるのでページの Public Keys 欄に貼り付けましょう

```
$ cat ~/.ssh/id_rsa.pub | clip
```

1
orize
on
can
ss

Paste the contents of your public key file (.pub)

The contents of your public key file

Save

アプリ公開

6. git clone

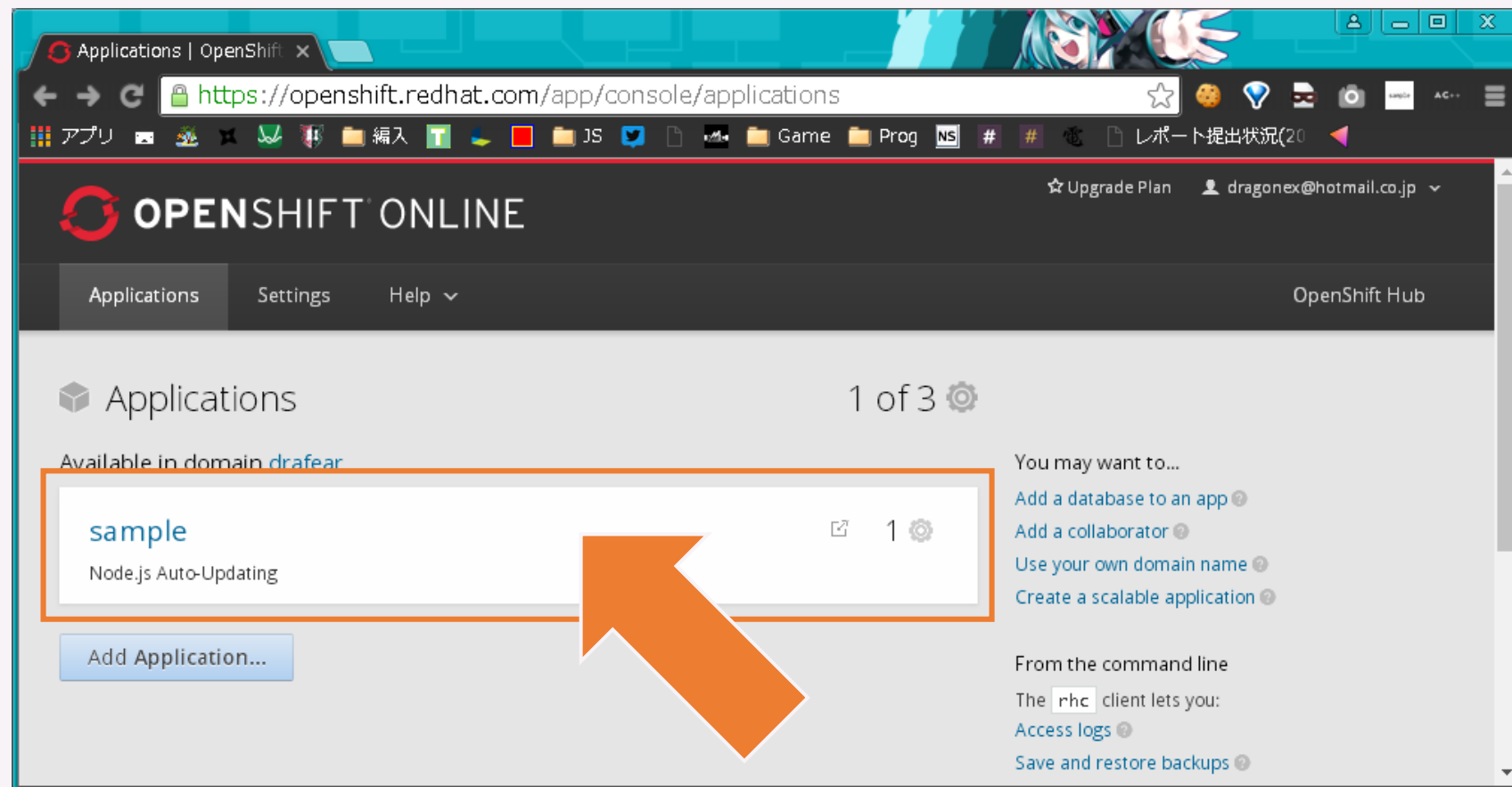
- 雛形をダウンロードします
- ダウンロードしたい場所へ cd コマンドで移動して下さい

```
$ cd "ダウンロードしたい場所のパス"
```

アプリ公開

6. git clone

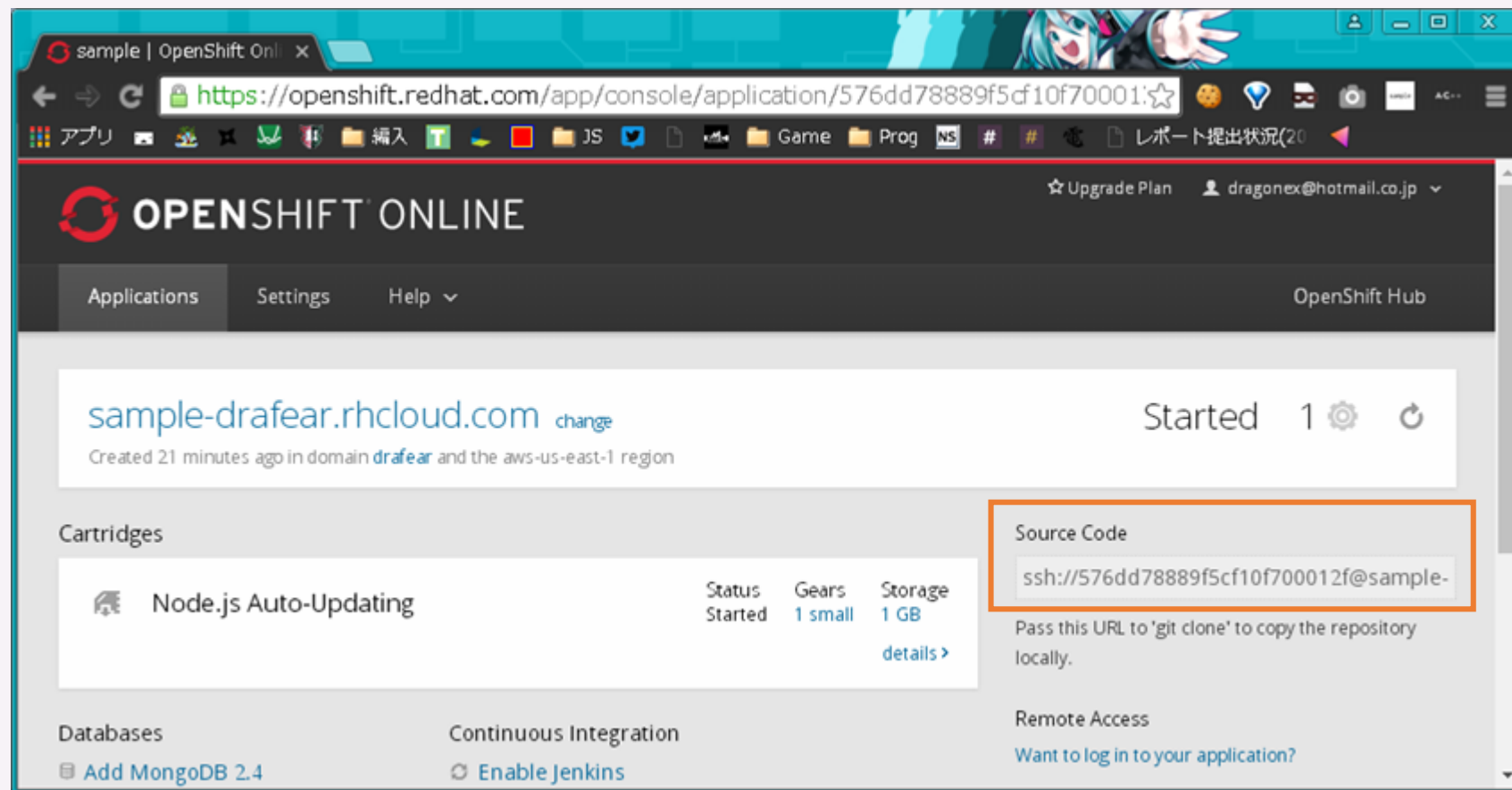
- アプリのページを開いて下さい



アプリ公開

6. git clone

- Source Code 部分をコピー



アプリ公開

6. git clone

- 端末で `git clone` [今コピーしたもの] を実行します

```
$ git clone [今コピーしたもの]
```

アプリ公開

vaScript

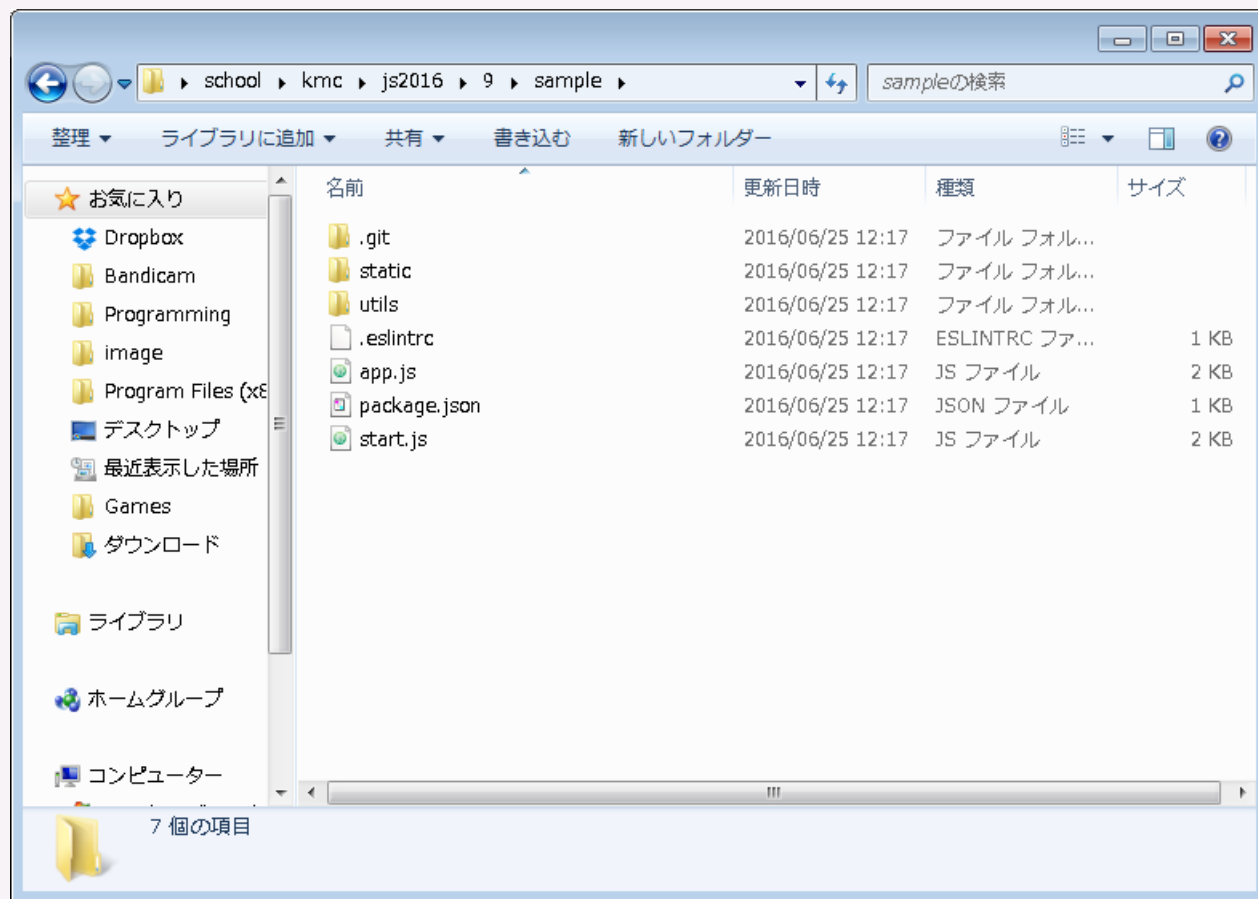
6. git clone

```
$ git clone ssh://576dd78889f5cf10f700012f@sample-drafear.rhcloud.com/~/.git/sample.git/  
Cloning into 'sample'...  
The authenticity of host 'sample-drafear.rhcloud.com (54.210.13.53)' can't be established.  
RSA key fingerprint is SHA256:FeJcKIUL7D41FMGNtM5NYM4XoboxZGYEqH0WcAmOX+I.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added 'sample-drafear.rhcloud.com,54.210.13.53' (RSA) to the list of known hosts.  
Enter passphrase for key '/home/Drafear/.ssh/id_rsa': [鍵のパスワード(画面上には表示されない)]  
remote: Counting objects: 17, done.  
remote: Compressing objects: 100% (17/17), done.  
remote: Total 17 (delta 0), reused 17 (delta 0)  
Receiving objects: 100% (17/17), 43.33 KiB | 0 bytes/s, done.  
Checking connectivity... done.
```

アプリ公開

7. ファイル追加・削除

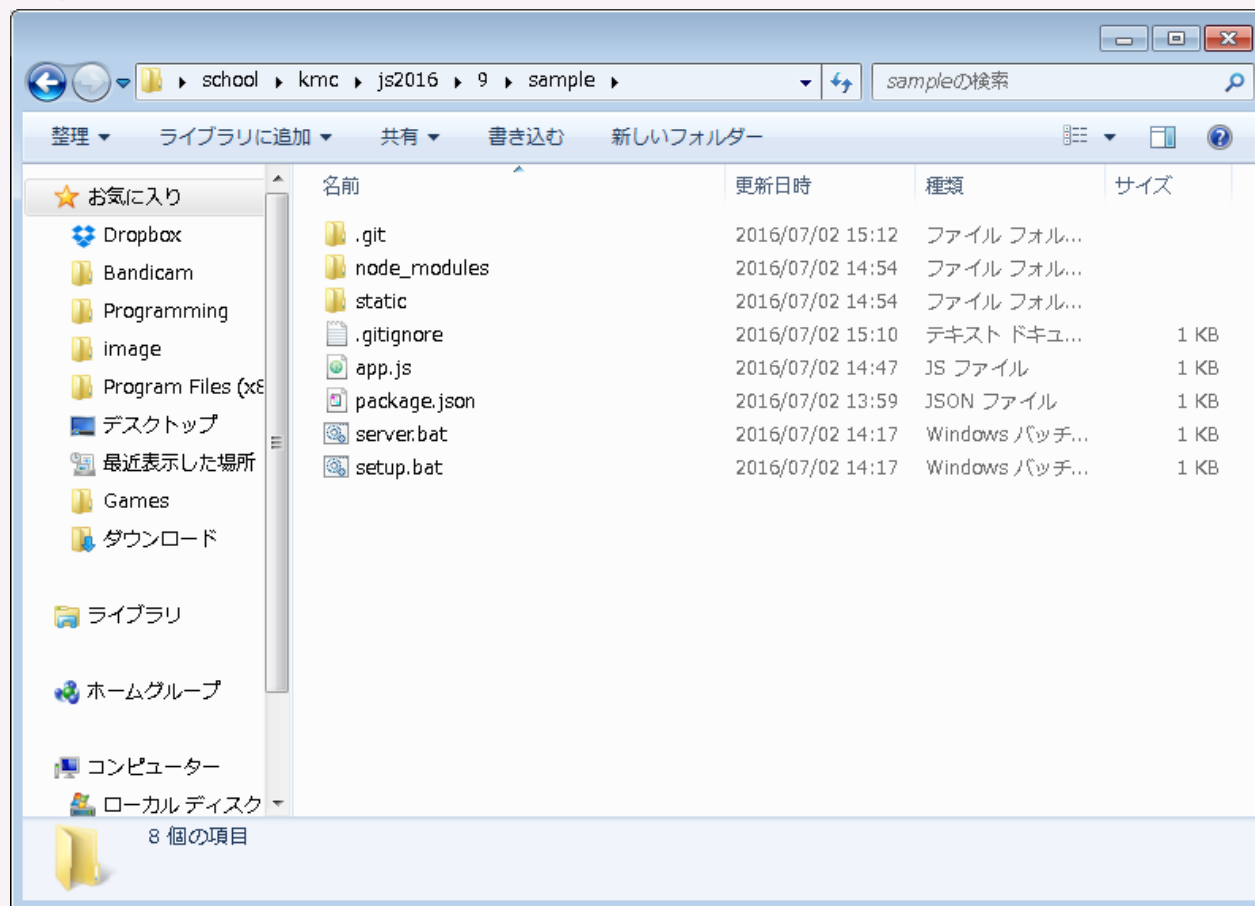
- git clone した場所を開きます



アプリ公開

7. ファイル追加・削除

- 全て削除し，自作アプリのファイル一式を入れます



アプリ公開

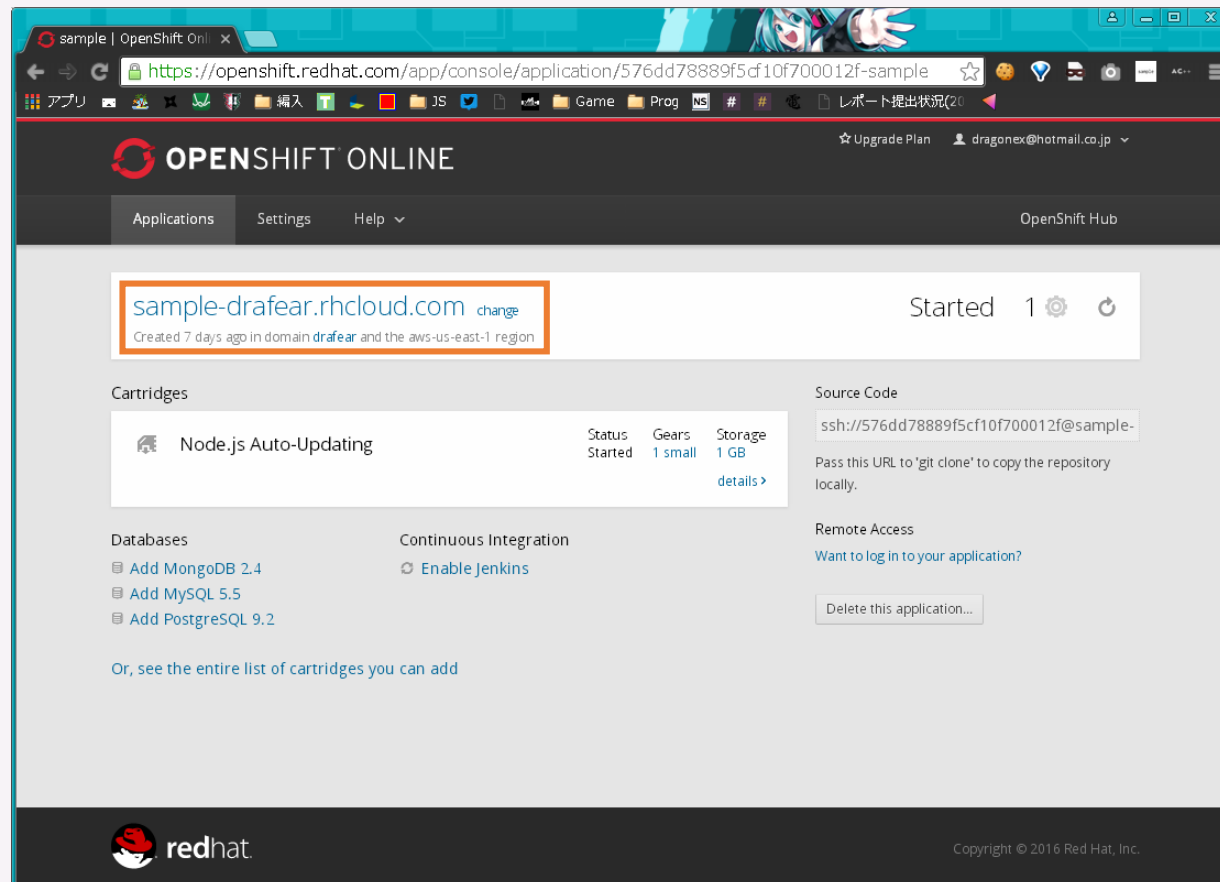
7. ファイル追加・削除

- 次のコマンドを実行します

```
$ cd [git cloneでできたディレクトリ]  
$ git add -A  
$ git commit -m "initial commit"  
$ git push
```

アプリ公開

- 公開完了！お疲れ様でした！！
- アクセスはここから！！



アプリ公開

- 次回以降はこれだけで更新を反映！

```
$ cd [git cloneでできたディレクトリ]  
$ git add -A  
$ git commit -m "どう変更したかを記述"  
$ git push
```