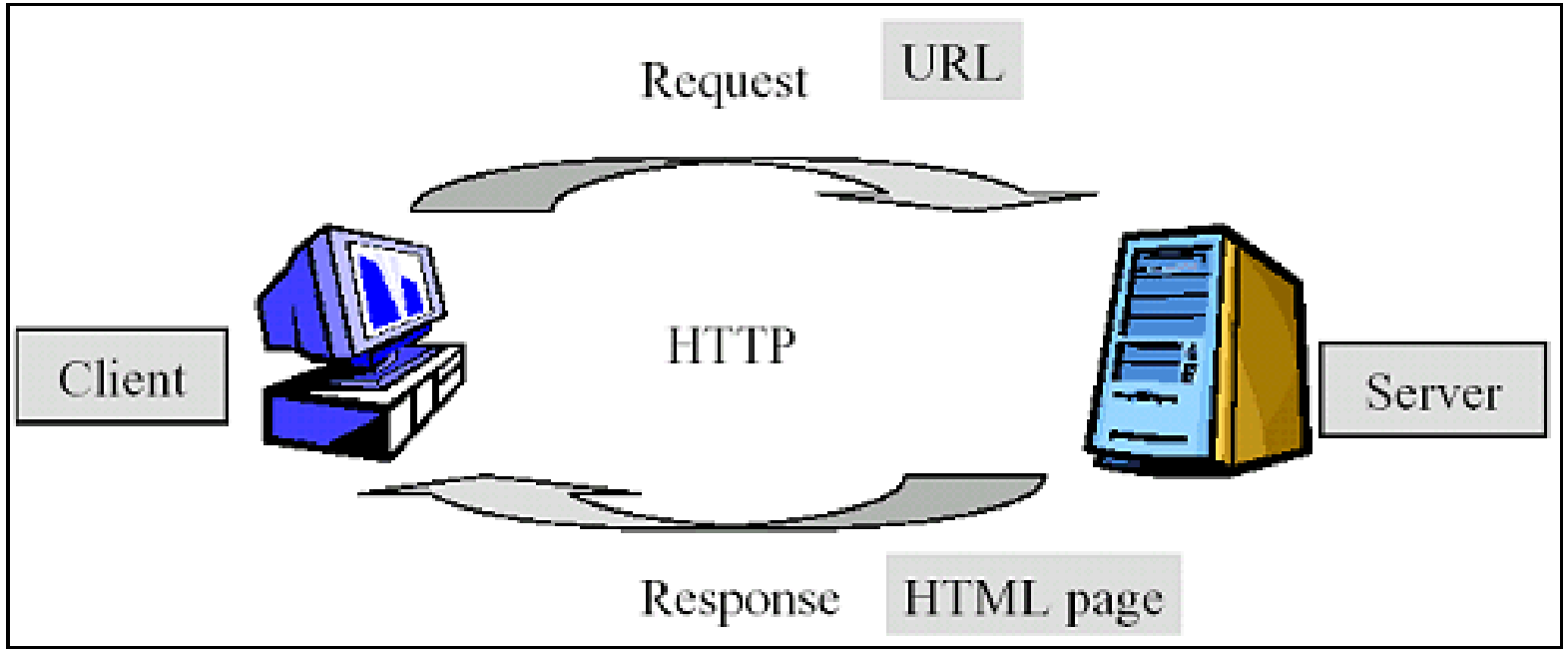
Client Server Architectures

Prepared By:

Viswanath M S

Client Server Models



Server Side

- We are going to make database queries on the server side and print the result of these queries in the client side.
- Server Side Language: JavaScript
- Framework Used: Node JS

Installing Node JS

- You can find three different installations of Node JS(Mac, Windows,Linux) here.
- Download and run to install.
- In your command prompt type npm and check the output.

Getting started with the application

- After you install the Node JS clone this application.
- The application has a basic hello world application.
- Start the application using the following commands. Node JS application generally run on port 3000.

1. npm install
2. npm start

- Open your browser and type localhost:3000. You should see Hello World printed.

Configuring the path

- Open app.js in the folder that you just downloaded
- You will be able to see a connection string construction code. Under that you will find a line

  - app.get(‘/', routes.listSysTables(ibmdb,connString));

  - The first parameter is a path and the second parameter is the method to be called if this is the path that is entered in the browser.
  - ‘/’ represents the default path

Configuring the path – Contd.

- You can add more paths by just copy pasting this call. Eg. app.get(‘/csci4140', routes.getCourses(ibmdb,connString));
- This means that if user enters something like http://localhost:3000/csci4140 the getCourses() method is called.
- You should be writing your queries in this method.

Writing Database Queries

- Inside routes folder you will be able to find a file name index.js
- All queries should be written inside conn.query method. There is no need to change anything else.
- Just replicate this method and rename it to support more queries.

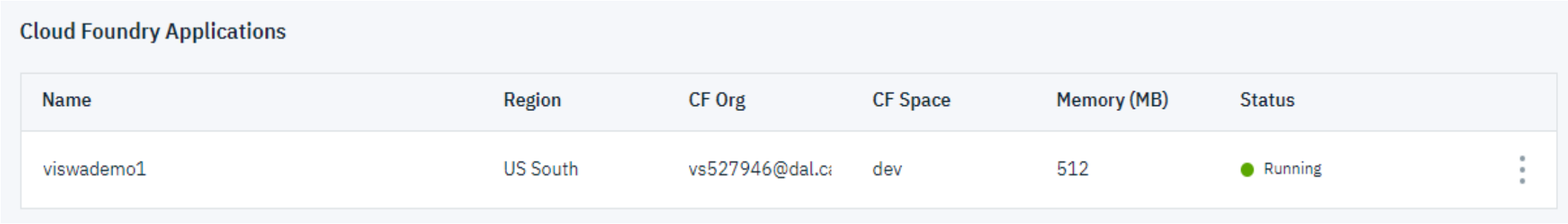
```
exports.listSysTables = function(ibmdb,connString,querystring) {
    return function(req, res) {


       ibmdb.open(connString, function(err, conn) {
            if (err ) {
             res.send("error occurred " + err.message);
            }
            else {
                var url_parts = url.parse(req.url,true);
                res.render("THisss"+url_parts.query);
                conn.query("SELECT FIRST_NAME, LAST_NAME, EMAIL, WORK_PHONE from GOSALESHR.employee FETCH FIRST 10 ROWS ONLY", function(err, tables, moreResultSets) {

                res.render("QUERY String"+ queryString);
                if ( !err ) {
                    res.render('tablelist', {
                        "tablelist" : tables,
                        "tableName" : "10 rows from the GOSALESHR.EMPLOYEE table",
                        "message": "Congratulations. Your connection to dashDB is successful."

                     });


                } else {
                   res.send("error occurred " + err.message);
                }

                /*
                    Close the connection to the database
                    param 1: The callback function to execute on completion of close function.
                */
                conn.close(function(){
                    console.log("Connection Closed");
                    });
                });
            }
         } );

    }
    }
```

Uploading your application to Bluenose

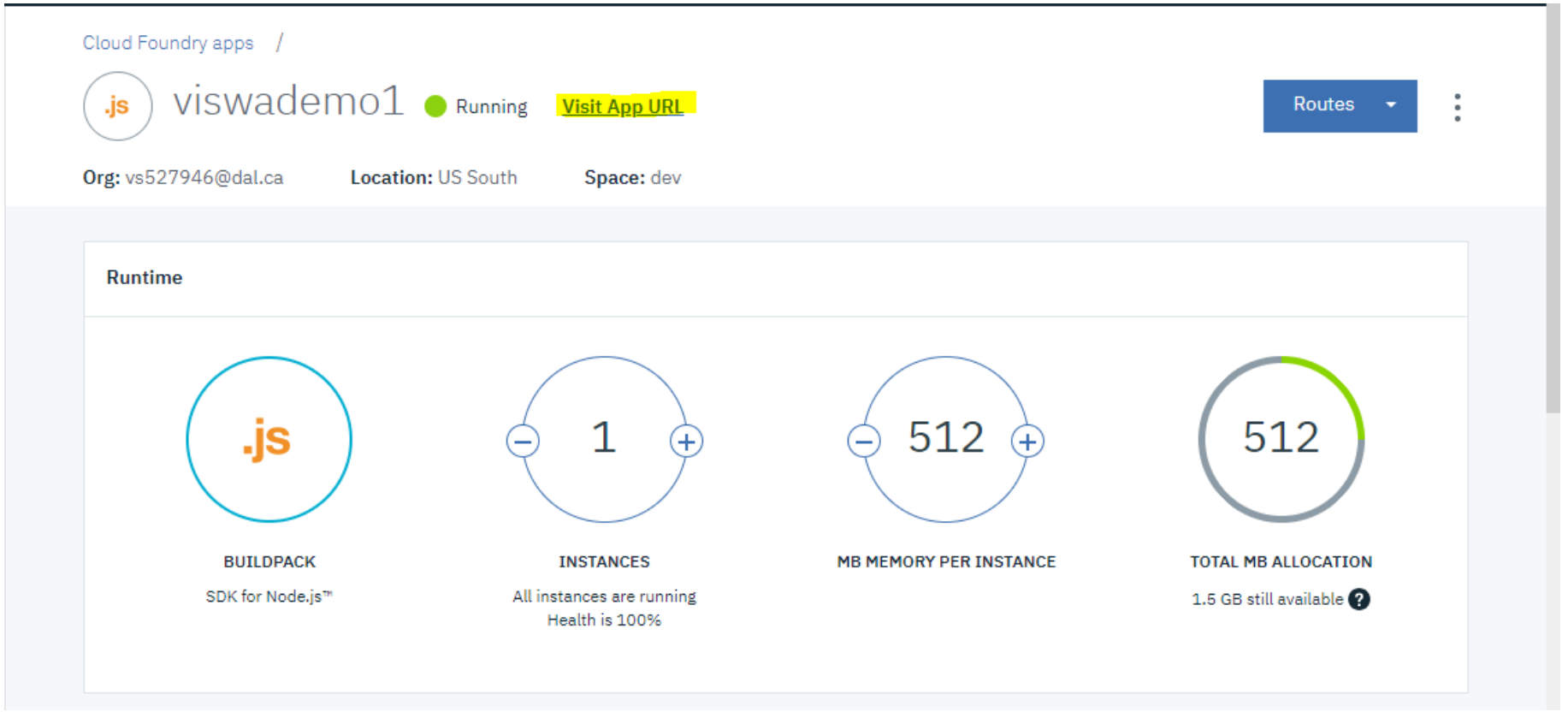
- Obtain Cloud foundry tools from here
- Configure your server using the following comman
    - cf api https://api.ng.bluemix.net
- Login to the server using
    - cf login and enter user bluemix user name and password
- Push your code to the server using
    - cf push

See your application in action

- Log in to IBM Cloud
- Your will see a section called cloud foundry applications
- Click on your application



- Click on visit app url



Thank You!