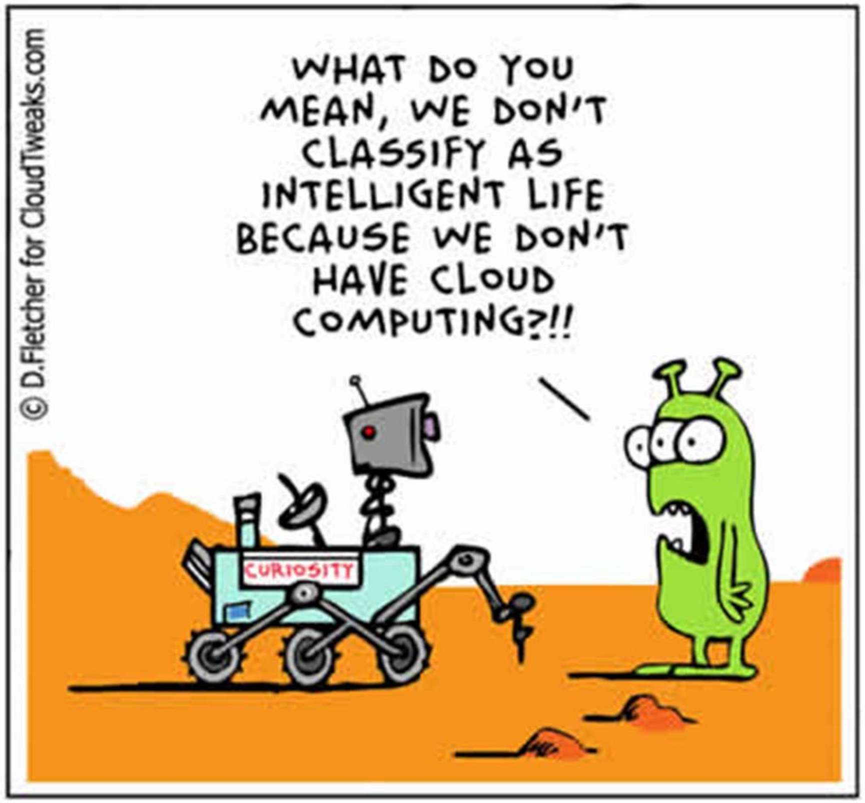
H01: IBM Cloud Services


WHAT DO YOU MEAN, WE DON'T CLASSIFY AS INTELLIGENT LIFE BECAUSE WE DON'T HAVE CLOUD COMPUTING?!!
CURIOSITY
© D.Fletcher for CloudTweaks.com

Outline

- Objective
- IBM Cloud Platform – Bluemix
  - Overview
  - Target Consumers
  - Technology
- Examples
  - PaaS: Set up a HelloWorld application using Boilerplates
  - SaaS: Data Analytics using dashDB, SQL Query and R
- Summary

Objective

- show how simple PaaS and SaaS applications can be developed using IBM Bluemix

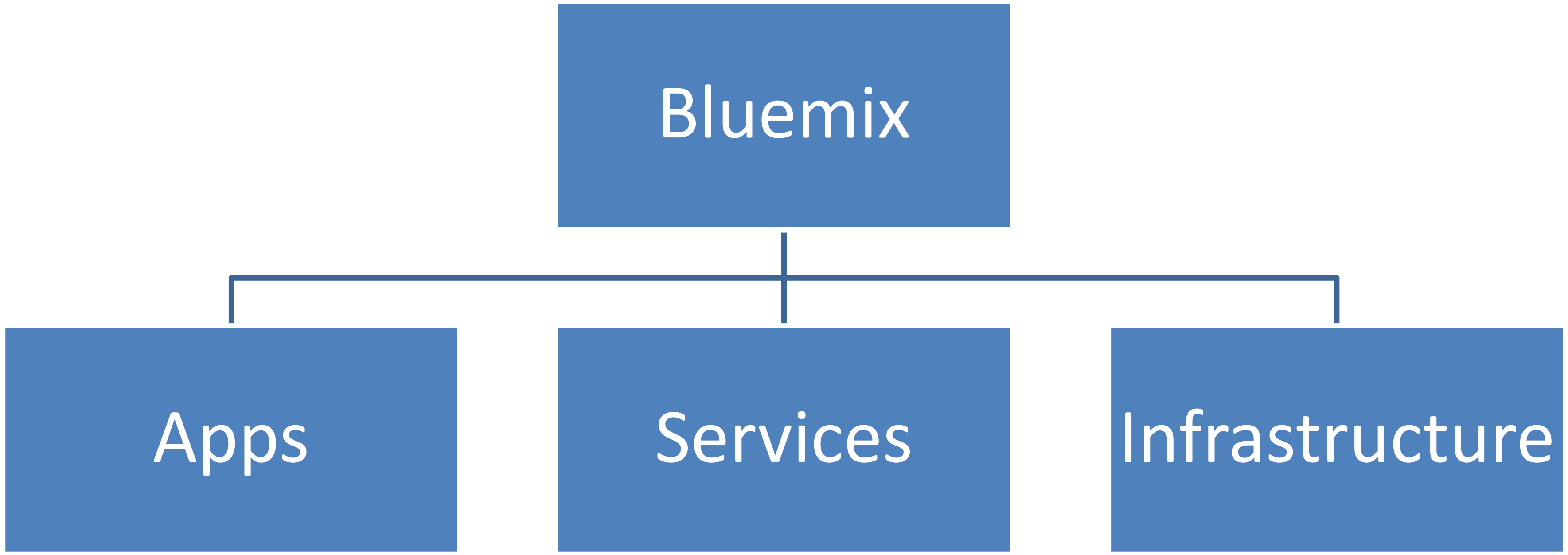
- examples:
  1. setting up a web server with a simple application (using boilerplate)
  2. a simple data analytics application using world development indicators data from the World Bank (using dashDB, SQL Query and R)

What is IBM Bluemix?

“an implementation of IBM’s Open Cloud Architecture, leveraging Cloud Foundry to enable developers to rapidly build, deploy, and manage their cloud applications, while tapping a growing ecosystem of available services and runtime frameworks”

https://console.ng.bluemix.net/

IBM Bluemix Services

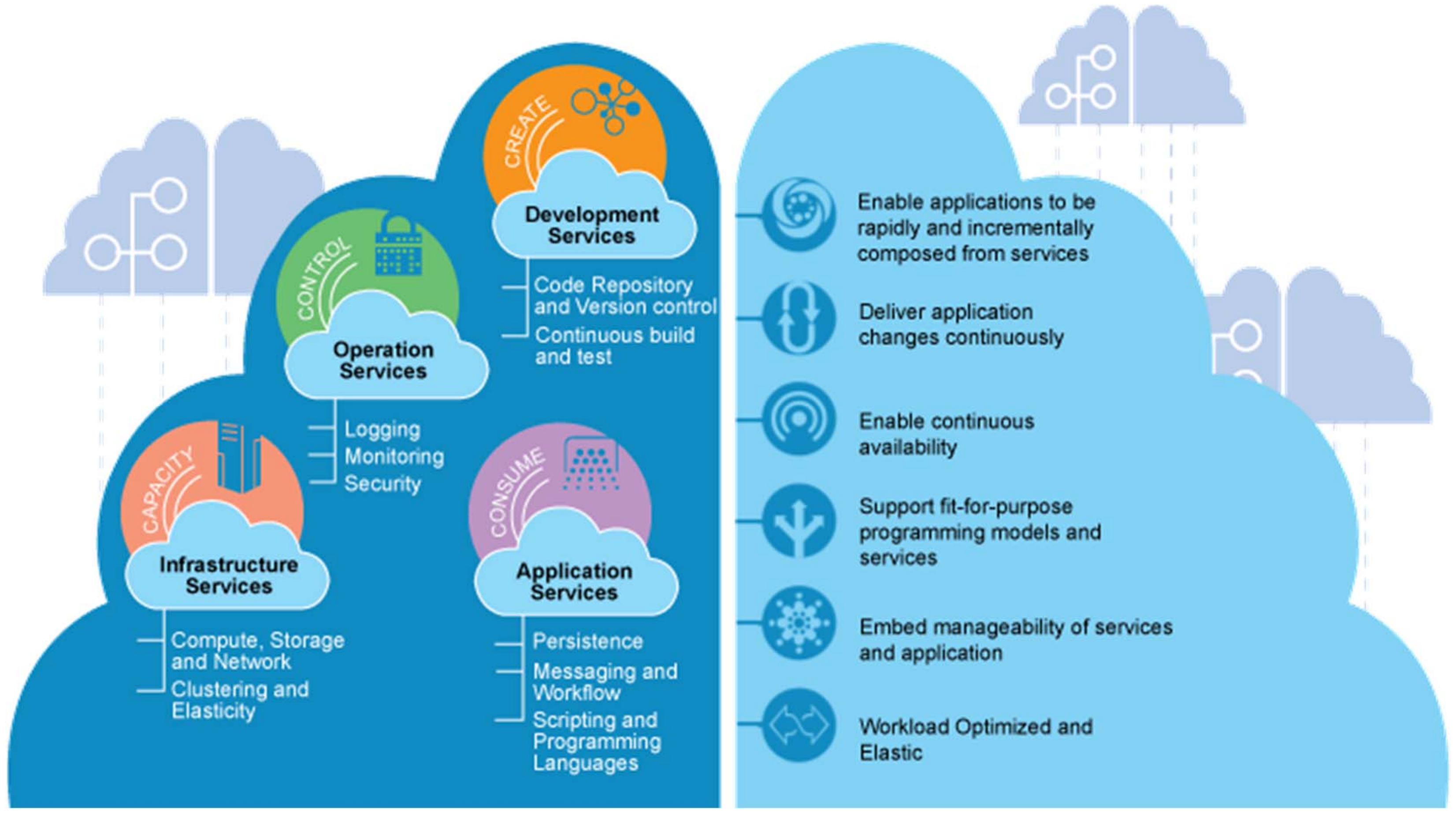


- Boilerplates
- Containers
- Mobile
- …

- Data & Analytics
- Watson
- IoT
- DevOps
- ….

- Compute
- Storage
- Network
- Security
- …

Bluemix Platform Features


CREATE
Development Services
Code Repository and Version control
Continuous build and test
CONTROL
Operation Services
Logging
Monitoring
Security
CAPACITY
Infrastructure Services
Compute, Storage and Network
Clustering and Elasticity
CONSUME
Application Services
Persistence
Messaging and Workflow
Scripting and Programming Languages
Enable applications to be rapidly and incrementally composed from services
Deliver application changes continuously
Enable continuous availability
Support fit-for-purpose programming models and services
Embed manageability of services and application
Workload Optimized and Elastic

source: http://www.ibm.com/developerworks/cloud/library/cl-bluemix-dbarnes/

Target Cloud Consumers

- Application Developers
  - Supports multiple languages
  - Auto-managed: version control, flexible capacity

- Businesses
  - Ease of deployment -> faster time-to-market
  - Lesser need for technical know-how

- Users
  - Needs are addressed fast
  - Feature updates made sooner
    - No need for special “going live mode” to update/release feature

Companies using Bluemix

Finance

IBM Bluemix accelerates digital banking innovation through the Citi Mobile Challenge.

Technology

Bitcraze builds Crazyflie drone with IBM Bluemix.

New

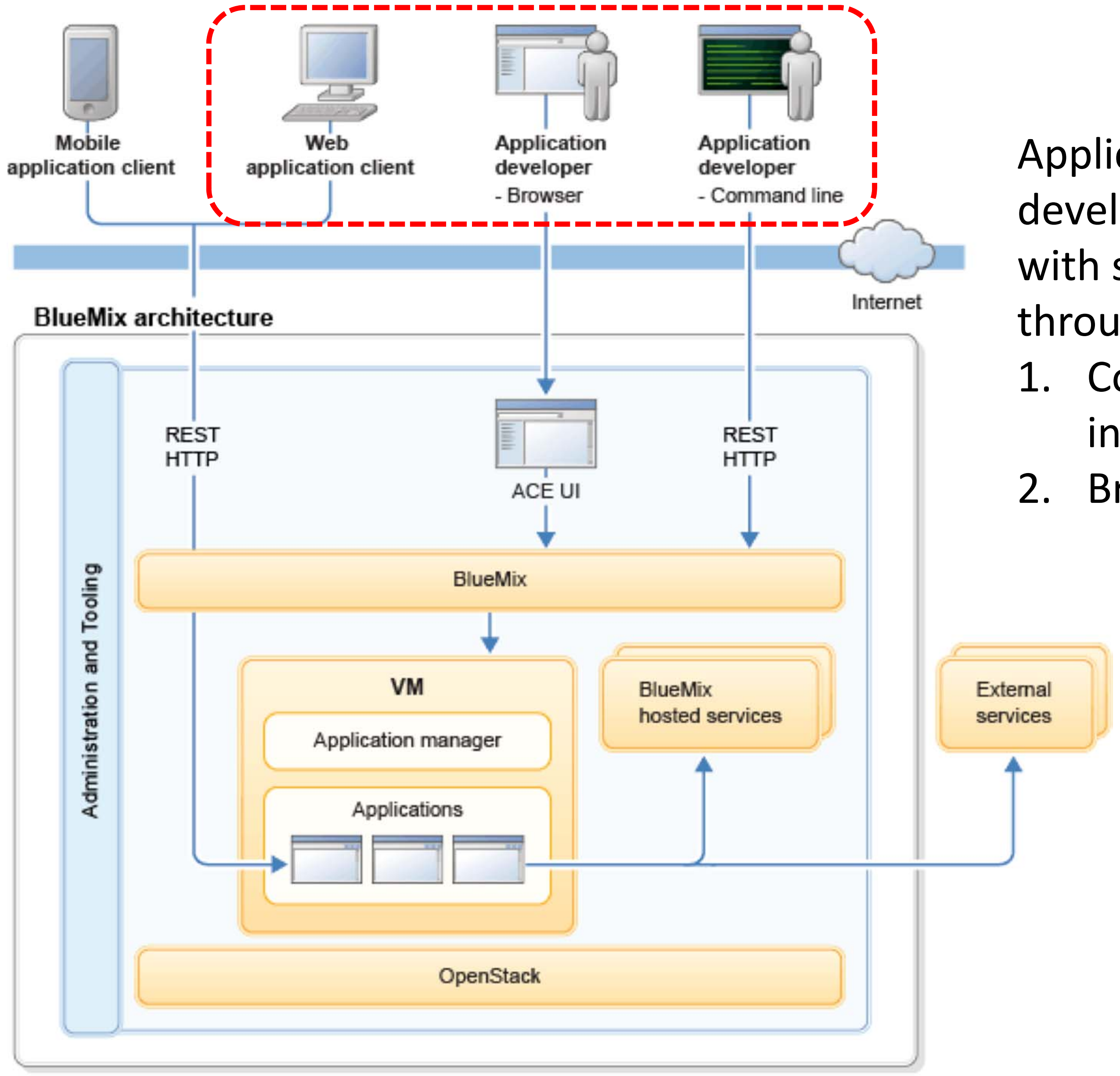
More

Creating a unique digital experience to capture the moments that matter

Retail

IBM Bluemix delivers scalable, cost-effective DevOps capabilities for BYTE restaurant feedback startup.

BlueMix High-level Architecture



Application developer interacts with systems through:

1. Command line interface
2. Browser interface

Bluemix Cloud Foundry

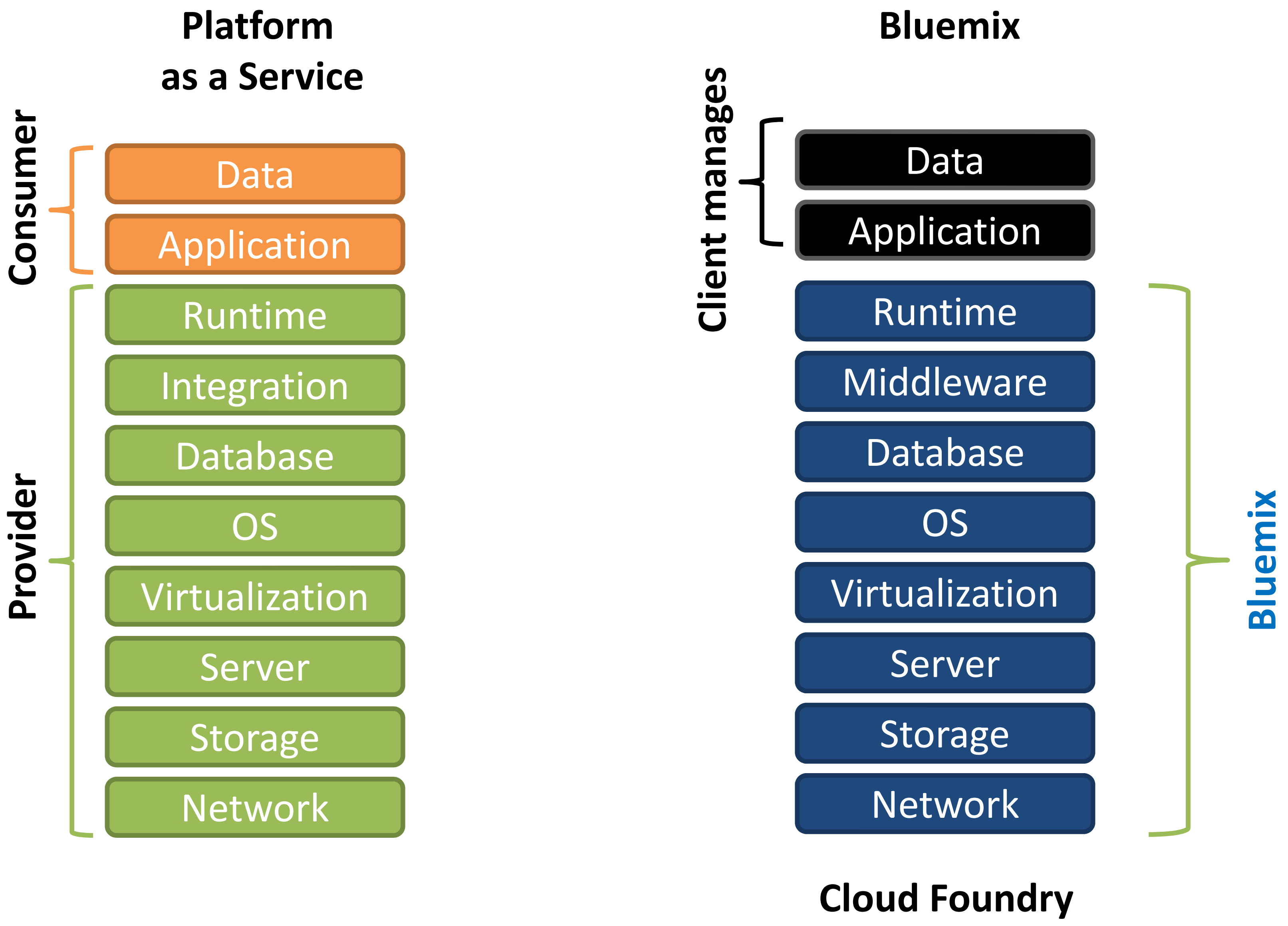
- Users focus on application code, i.e., don’t have to worry about the OS and infrastructure layers

- Cloud Foundry is an open source PaaS
  - Corporate users: SAS, Cisco, Rakuten, Baidu, SAP Verizon, ..
  - Leverages on the broad community
  - Push, extend and manage applications using
    - a command line tool (cf)
    - eclipse plugin
    - DevOps …
  - Addresses PaaS lack of cross-compatibility among different cloud providers

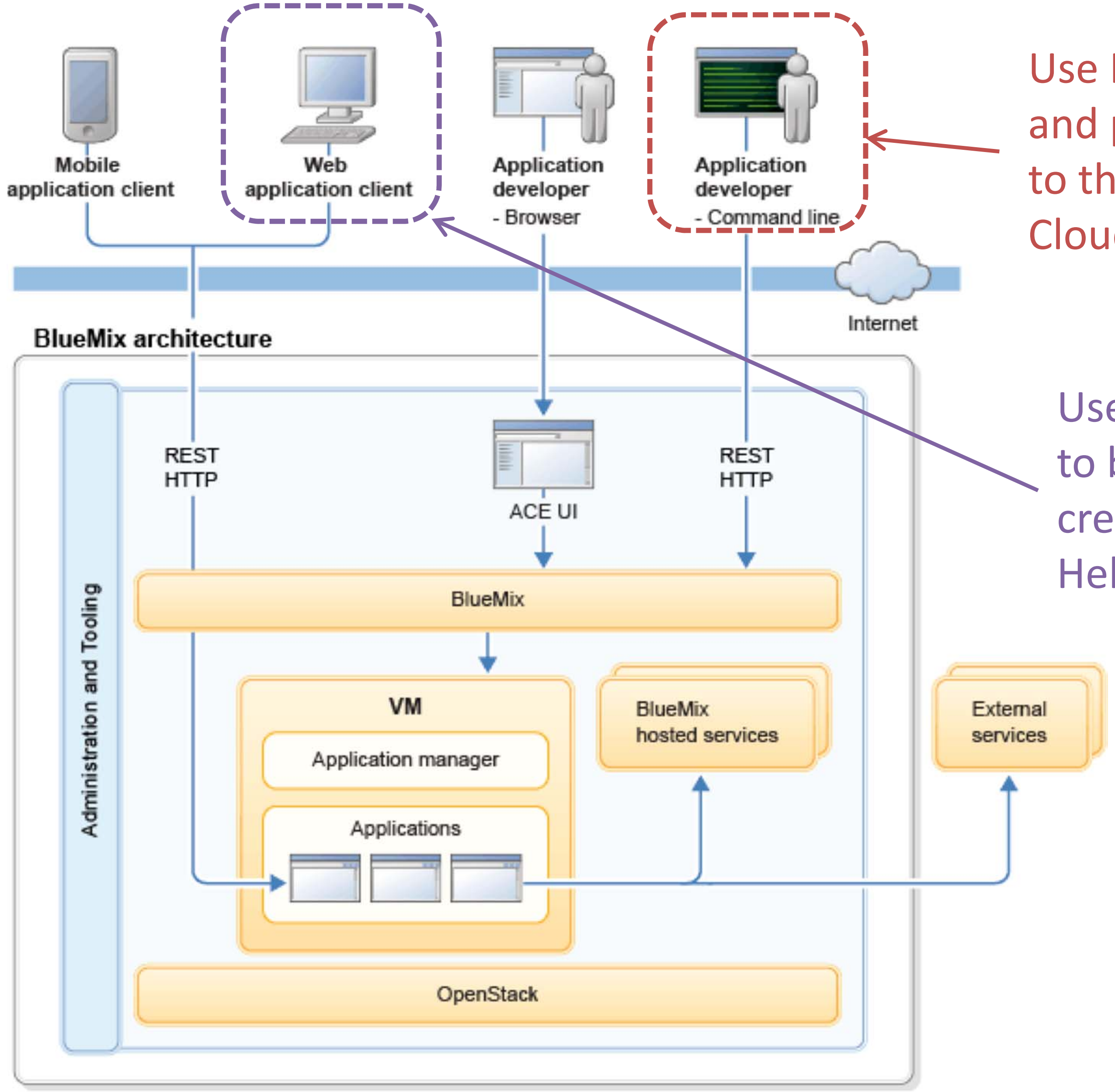
PaaS: HelloWorld Application

- Objective: set up a web server application using Cloud Foundry boilerplate (Node.js platform)
- Example: HelloWorld application running on a cloud-based web server using node.js boilerplate
- Services can be added on top of the boilerplate runtime (next dashDB example)

What is PaaS?


Platform
as a Service
Consumer
Provider
Data
Application
Runtime
Integration
Database
OS
Virtualization
Server
Storage
Network
Bluemix
Client manages
Data
Application
Runtime
Middleware
Database
OS
Virtualization
Server
Storage
Network
Bluemix
Cloud Foundry

PaaS: HelloWorld Application



Use HelloWorld.js and push the app to the Cloud using CloudFoundry CLI

Use web-browser to browse the created HelloWorld server

Main Steps: Web Application using Node.js

1. Sign up for IBM Bluemix

2. Select Cloud Foundry Node.js app from the catalog – this is needed for every new application

3. Provide a hostname that serves as the web application URL – hostname.mybluemix.net

4. Install the Cloud Foundry (CF) CLI (command line interface) client on your local machine - this is required if you have not done so previously

5. The following are performed using CF CLI to interact with the Cloud Foundry. (alternatively developer can use a web browser interface) To start a new web application, download the starter code (public/index.html, manifest.yml, app.js, package.json,..) onto your local directory

6. Make changes to the starter code as required

7. Deploy/upload application on Bluemix. Using the CF CLI, connect the CF client to the selected Bluemix region (server)

8. Using the CF CLI, login to Bluemix

9. Using the CF CLI, upload your application to Bluemix!

10. To access the web application, enter your host URL (in step 3)

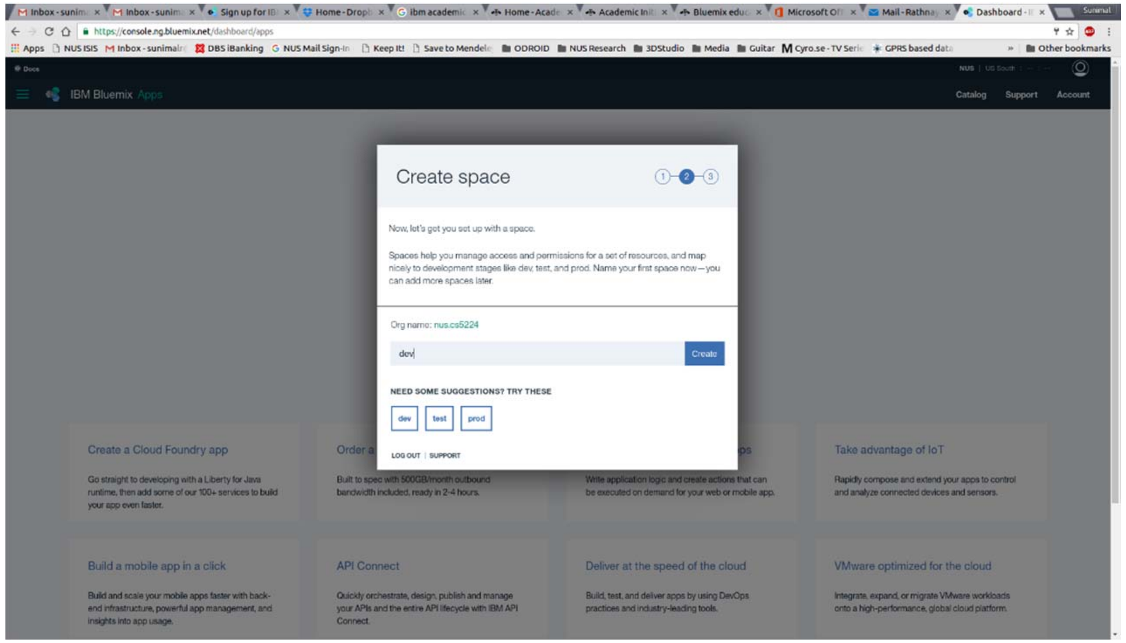
Step 1: Signing up

- IBM ID registration:
  https://www.ibm.com/account/profile/us?page=reg

- Bluemix signin with created IBM ID:
  https://console.ng.bluemix.net/

- Organization:
  Region: *US South*
  Org name: *nus.cs5224*

- Create space:
  Space: *dev*



Create space

Now, let's get you set up with a space.

Spaces help you manage access and permissions for a set of resources, and map nicely to development stages like dev, test, and prod. Name your first space now—you can add more spaces later.

Org name: nus.cs5224

dev

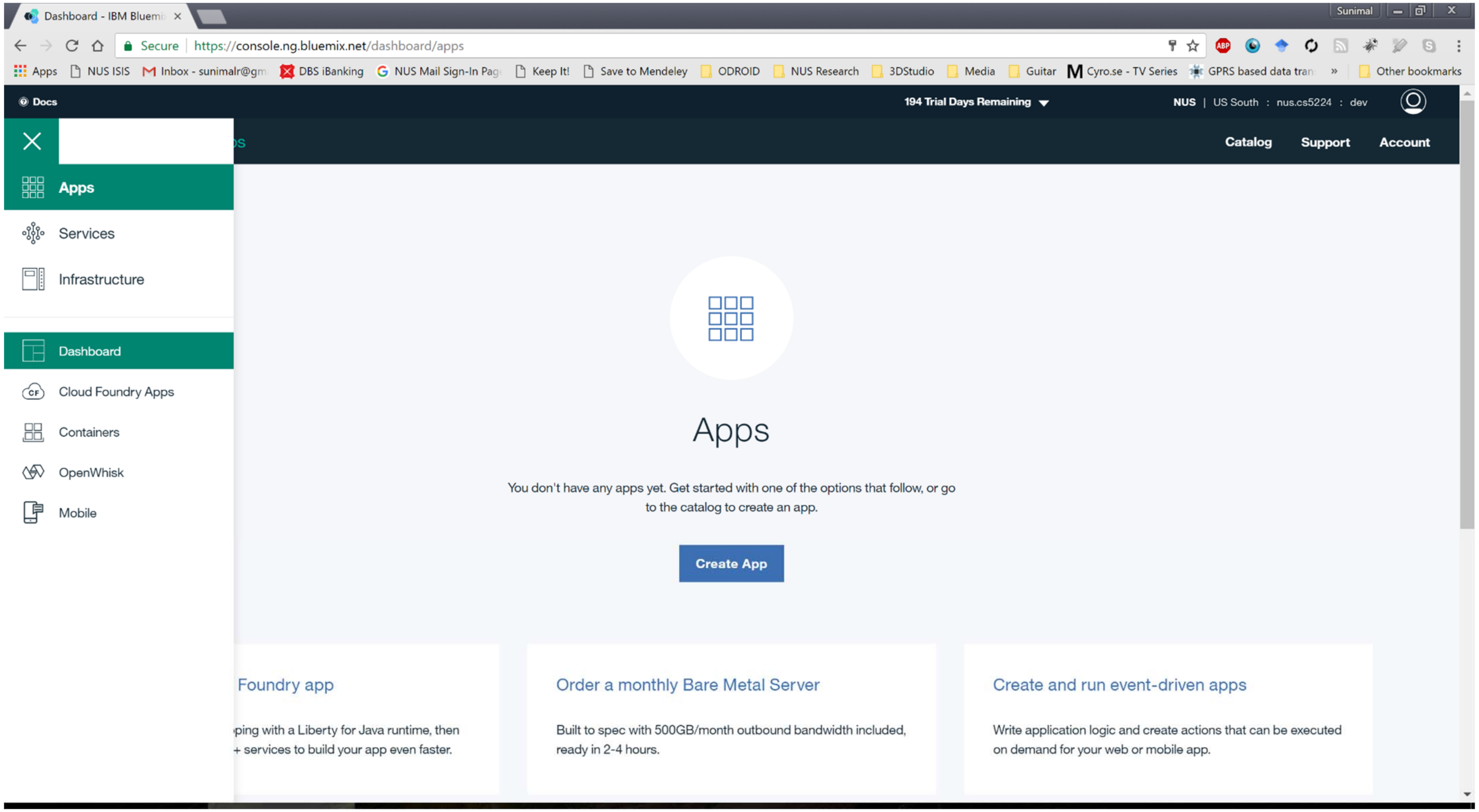
Create

NEED SOME SUGGESTIONS? TRY THESE

dev test prod

LOG OUT | SUPPORT

IBM Cloud Platform – Bluemix



Apps

You don't have any apps yet. Get started with one of the options that follow, or go to the catalog to create an app.

Foundry app

ping with a Liberty for Java runtime, then + services to build your app even faster.

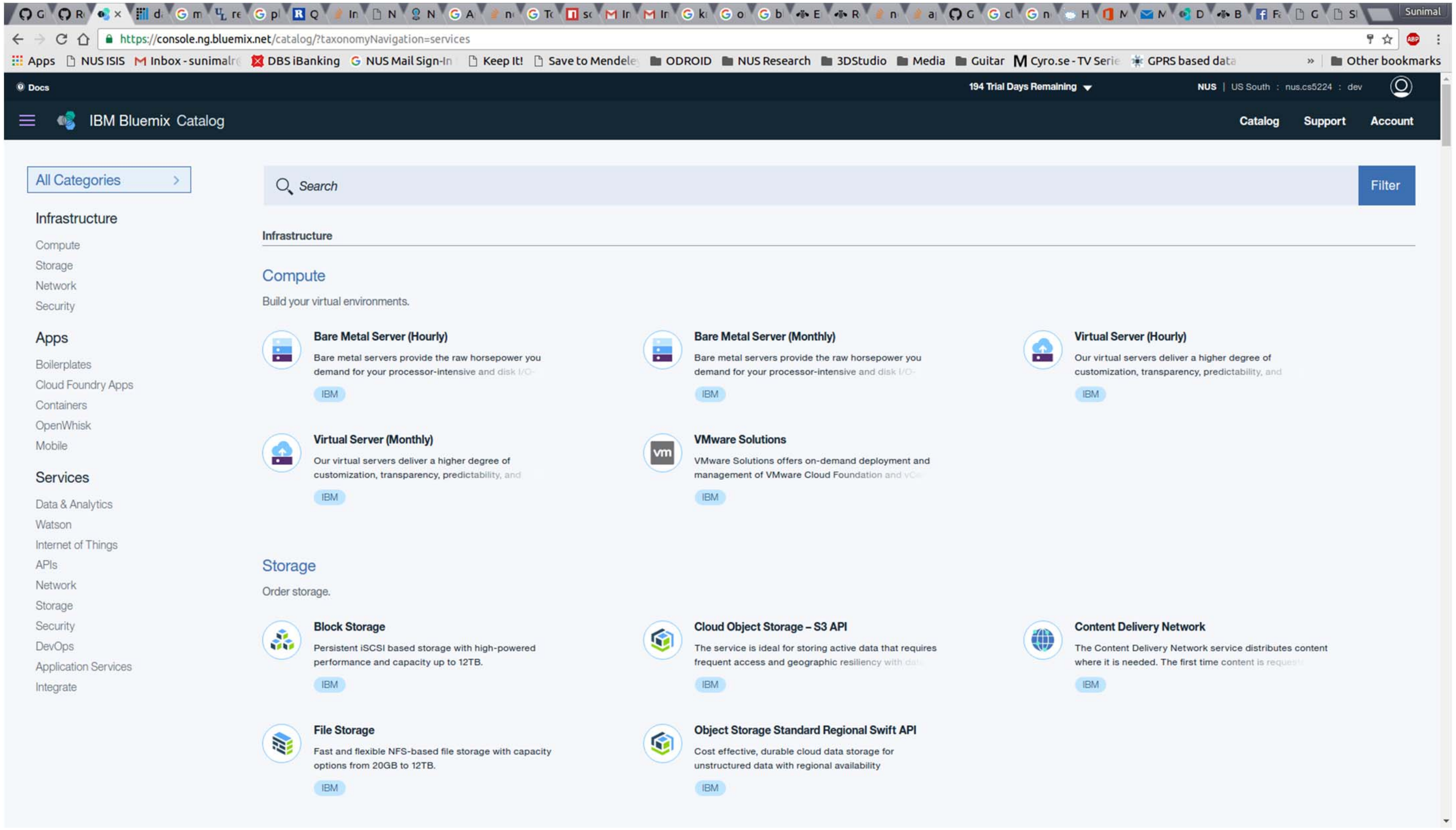
Order a monthly Bare Metal Server

Built to spec with 500GB/month outbound bandwidth included, ready in 2-4 hours.

Create and run event-driven apps

Write application logic and create actions that can be executed on demand for your web or mobile app.

Bluemix Catalog


https://console.ng.bluemix.net/catalog/?taxonomyNavigation=services
Docs
194 Trial Days Remaining
NUS | US South : nus.cs5224 : dev
IBM Bluemix Catalog
Catalog
Support
Account
All Categories
Search
Filter

Infrastructure
- Compute
- Storage
- Network
- Security

Apps
- Boilerplates
- Cloud Foundry Apps
- Containers
- OpenWhisk
- Mobile

Services
- Data & Analytics
- Watson
- Internet of Things
- APIs
- Network
- Storage
- Security
- DevOps
- Application Services
- Integrate

Infrastructure

Compute

Build your virtual environments.

Bare Metal Server (Hourly)
Bare metal servers provide the raw horsepower you demand for your processor-intensive and disk I/O-
IBM

Bare Metal Server (Monthly)
Bare metal servers provide the raw horsepower you demand for your processor-intensive and disk I/O-
IBM

Virtual Server (Hourly)
Our virtual servers deliver a higher degree of customization, transparency, predictability, and
IBM

Virtual Server (Monthly)
Our virtual servers deliver a higher degree of customization, transparency, predictability, and
IBM

VMware Solutions
VMware Solutions offers on-demand deployment and management of VMware Cloud Foundation and vC
IBM

Storage

Order storage.

Block Storage
Persistent iSCSI based storage with high-powered performance and capacity up to 12TB.
IBM

Cloud Object Storage – S3 API
The service is ideal for storing active data that requires frequent access and geographic resiliency with da
IBM

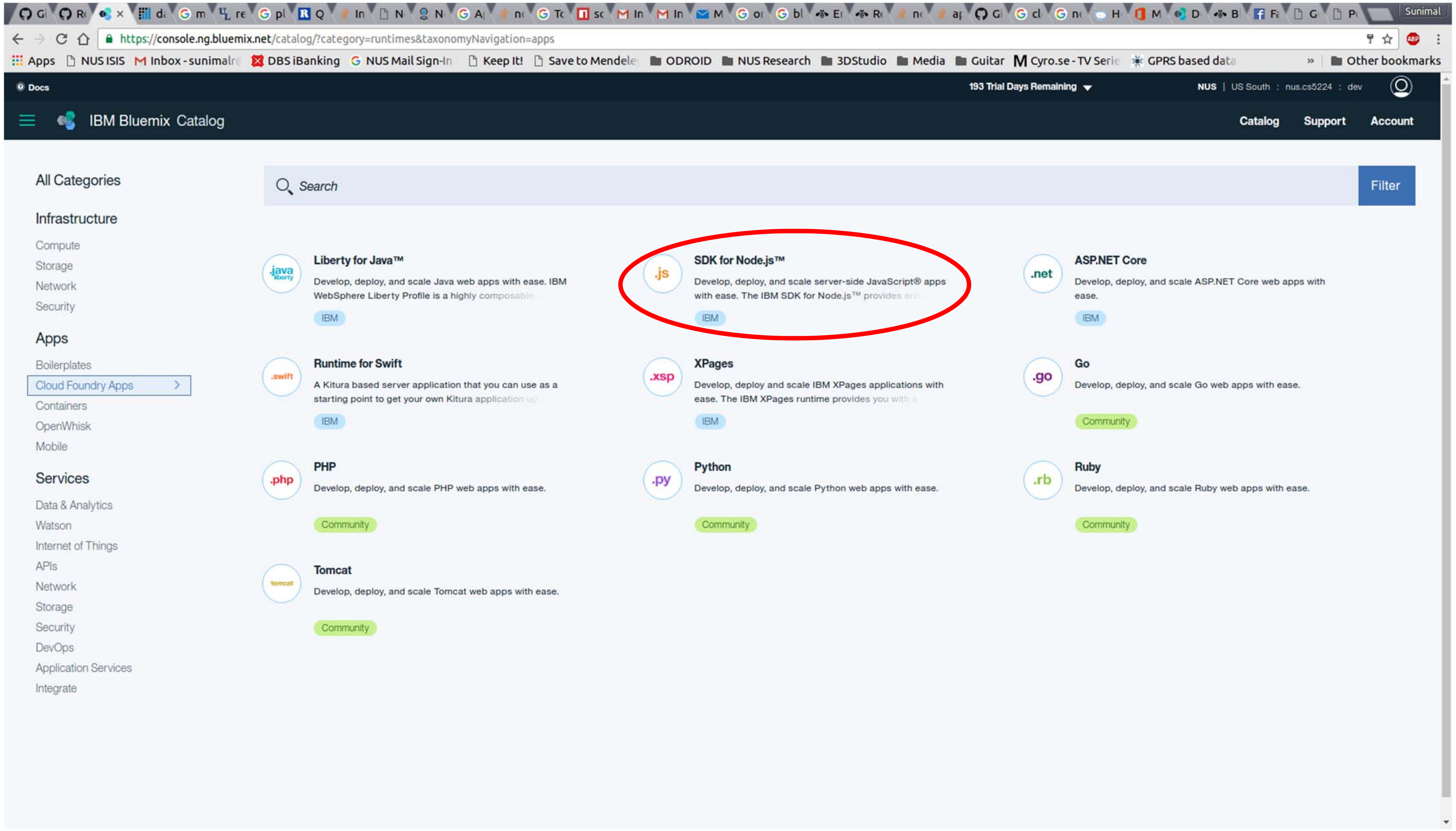
Content Delivery Network
The Content Delivery Network service distributes content where it is needed. The first time content is reques
IBM

File Storage
Fast and flexible NFS-based file storage with capacity options from 20GB to 12TB.
IBM

Object Storage Standard Regional Swift API
Cost effective, durable cloud data storage for unstructured data with regional availability
IBM

Step 2: Select Cloud Foundry Node.js

- Go to catalog and select SDK for Node.js



https://console.ng.bluemix.net/catalog/?category=runtimes&taxonomyNavigation=apps

Apps | NUS ISIS | Inbox - sunimalr | DBS iBanking | NUS Mail Sign-In | Keep It! | Save to Mendeley | ODROID | NUS Research | 3DStudio | Media | Guitar | Cyro.se - TV Serie | GPRS based data | Other bookmarks

Docs | 193 Trial Days Remaining | NUS | US South : nus.cs5224 : dev

IBM Bluemix Catalog | Catalog | Support | Account

Search | Filter

All Categories

Infrastructure: Compute, Storage, Network, Security

Apps: Boilerplates, Cloud Foundry Apps, Containers, OpenWhisk, Mobile

Services: Data & Analytics, Watson, Internet of Things, APIs, Network, Storage, Security, DevOps, Application Services, Integrate

Liberty for Java™ — Develop, deploy, and scale Java web apps with ease. IBM WebSphere Liberty Profile is a highly composable... — IBM

SDK for Node.js™ — Develop, deploy, and scale server-side JavaScript® apps with ease. The IBM SDK for Node.js™ provides an... — IBM

ASP.NET Core — Develop, deploy, and scale ASP.NET Core web apps with ease. — IBM

Runtime for Swift — A Kitura based server application that you can use as a starting point to get your own Kitura application up... — IBM

XPages — Develop, deploy and scale IBM XPages applications with ease. The IBM XPages runtime provides you with a... — IBM

Go — Develop, deploy, and scale Go web apps with ease. — Community

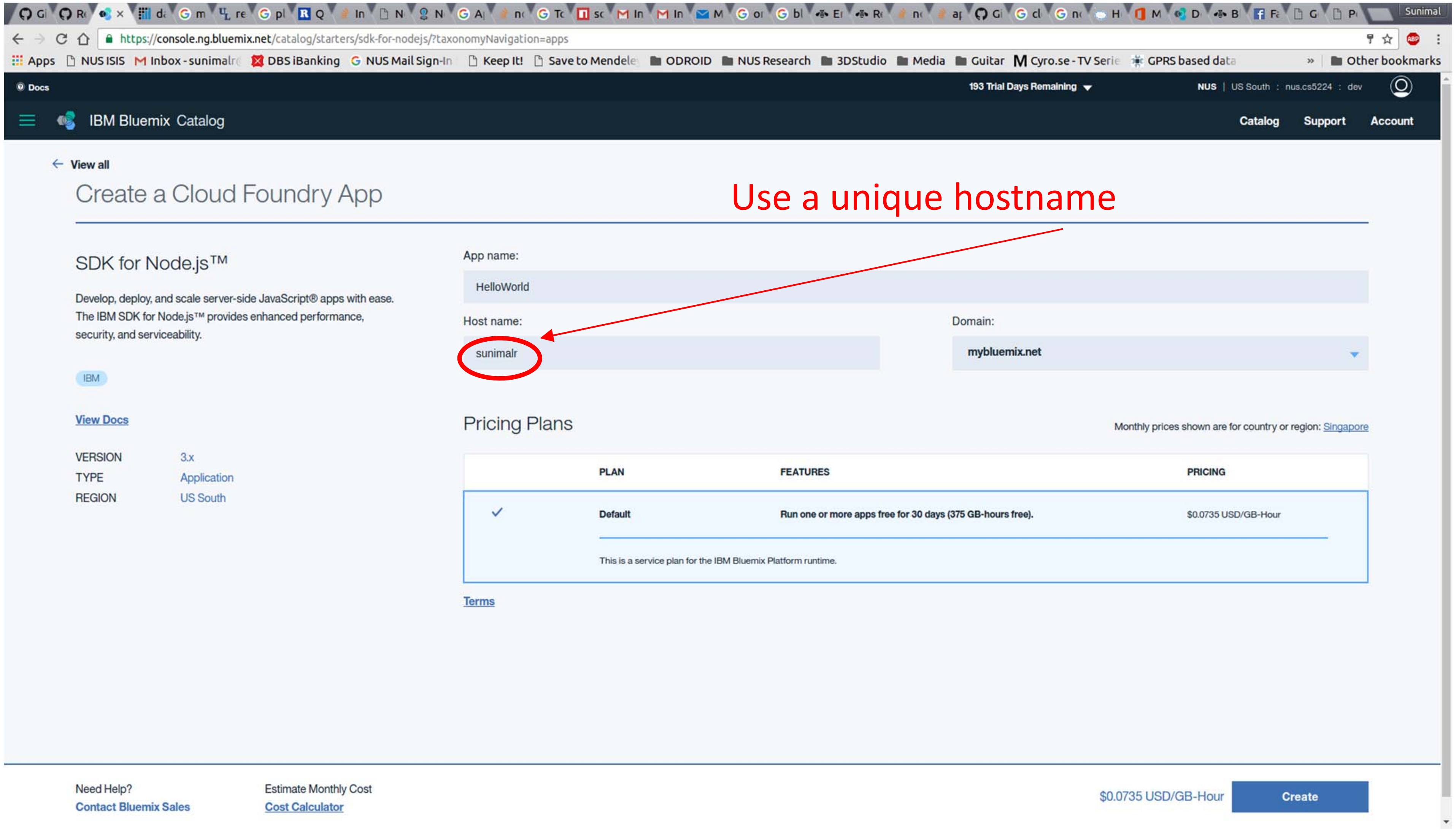
PHP — Develop, deploy, and scale PHP web apps with ease. — Community

Python — Develop, deploy, and scale Python web apps with ease. — Community

Ruby — Develop, deploy, and scale Ruby web apps with ease. — Community

Tomcat — Develop, deploy, and scale Tomcat web apps with ease. — Community

Step 3: Provide required app details



Create a Cloud Foundry App

Use a unique hostname

SDK for Node.js™

Develop, deploy, and scale server-side JavaScript® apps with ease. The IBM SDK for Node.js™ provides enhanced performance, security, and serviceability.

IBM

View Docs

VERSION	3.x
TYPE	Application
REGION	US South

App name: HelloWorld

Host name: sunimalr

Domain: mybluemix.net

Pricing Plans

Monthly prices shown are for country or region: Singapore

	PLAN	FEATURES	PRICING
✓	Default	Run one or more apps free for 30 days (375 GB-hours free).	$0.0735 USD/GB-Hour

This is a service plan for the IBM Bluemix Platform runtime.

Terms

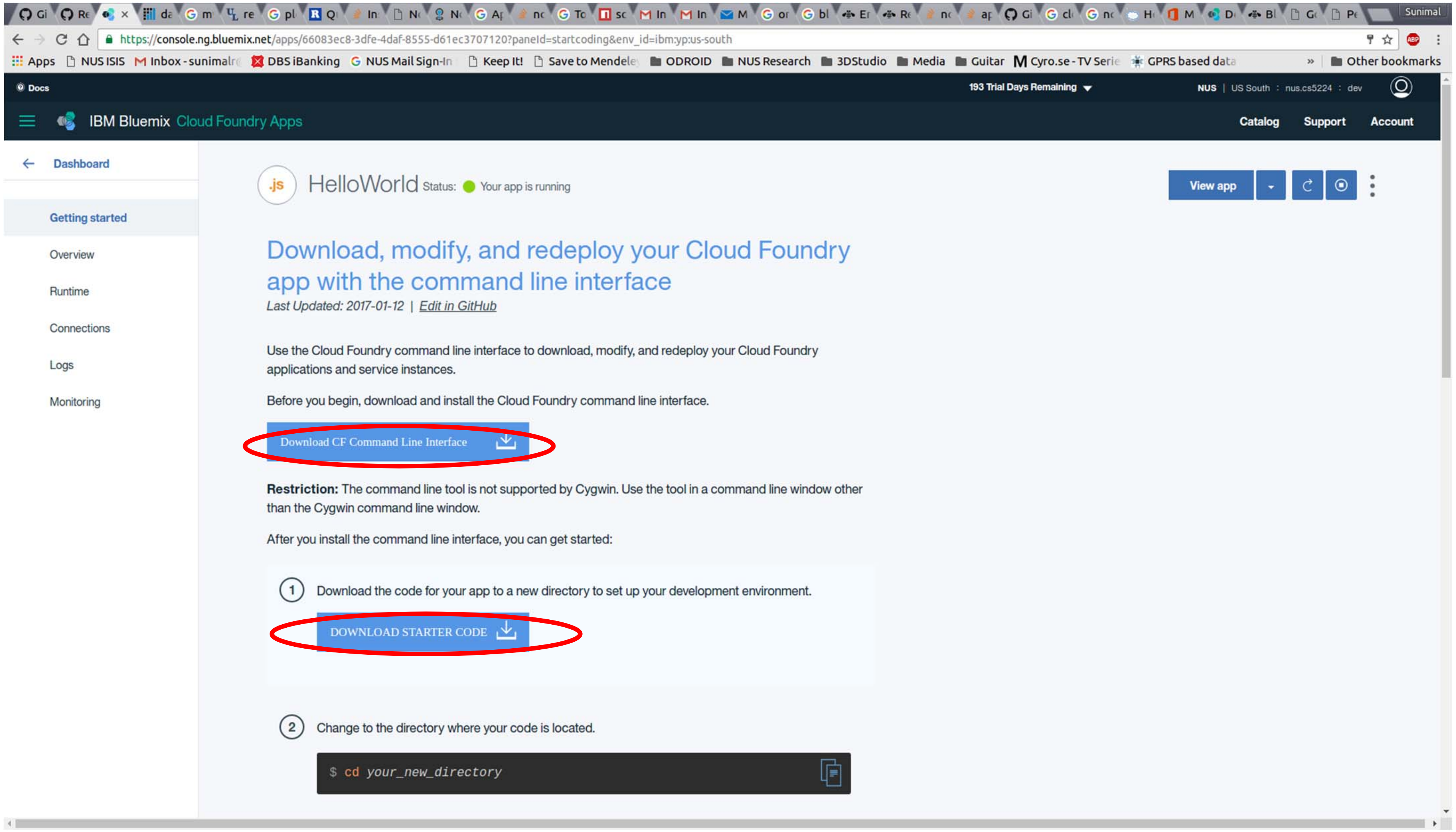
Need Help?
Contact Bluemix Sales

Estimate Monthly Cost
Cost Calculator

$0.0735 USD/GB-Hour

Create

Step 4 & 5: Install Cloud Foundry and Download Starter Code


https://console.ng.bluemix.net/apps/66083ec8-3dfe-4daf-8555-d61ec3707120?paneId=startcoding&env_id=ibm:yp:us-south
Docs
193 Trial Days Remaining
NUS | US South : nus.cs5224 : dev
IBM Bluemix Cloud Foundry Apps
Catalog
Support
Account
Dashboard
Getting started
Overview
Runtime
Connections
Logs
Monitoring
HelloWorld Status: Your app is running
View app

Download, modify, and redeploy your Cloud Foundry app with the command line interface

Last Updated: 2017-01-12 | Edit in GitHub

Use the Cloud Foundry command line interface to download, modify, and redeploy your Cloud Foundry applications and service instances.

Before you begin, download and install the Cloud Foundry command line interface.

Download CF Command Line Interface

Restriction: The command line tool is not supported by Cygwin. Use the tool in a command line window other than the Cygwin command line window.

After you install the command line interface, you can get started:

1. Download the code for your app to a new directory to set up your development environment.

2. Change to the directory where your code is located.

```
$ cd your_new_directory
```

Step 6: Change the downloaded content as desired

- Extract the downloaded file (HelloWorld.zip)
  - This will create a new directory named “HelloWorld”

- Edit content as necessary
  - You can edit public/index.html to display something you like. Eg. Change to “Hello World from Singapore”
  - Downloaded code can be edited/replaced by custom content in order to host custom web applications

- Take note of manifest.yml which sets deployment parameters when deploying applications via Cloud Foundry and app.js file which sets up server variables and initiates the web server

index.html

```
<!DOCTYPE html>
<html>

  <head>
    <title>NodeJS Starter Application</title>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="stylesheets/style.css">
  </head>

  <body>
    <table>
      <tr>
        <td style= "width:30%;">
          <img class = "newappIcon" src="images/newapp-icon.png">
        </td>
        <td>
          <h1 id="message">Hello World!</h1>
          <p class='description'></p> Thanks for creating a <span class = "blue">NodeJS Starter Application</span>.
        </td>
      </tr>
    </table>
  </body>

</html>
```

Note: index.html is located in the ‘public’ folder

app.js

```
//------------------------------------------------------------------------------
// node.js starter application for Bluemix
//------------------------------------------------------------------------------

// This application uses express as its web server
// for more info, see: http://expressjs.com
var express = require('express');

// cfenv provides access to your Cloud Foundry environment
// for more info, see: https://www.npmjs.com/package/cfenv
var cfenv = require('cfenv');

// create a new express server
var app = express();

// serve the files out of ./public as our main files
app.use(express.static(__dirname + '/public'));

// get the app environment from Cloud Foundry
var appEnv = cfenv.getAppEnv();

// start server on the specified port and binding host
app.listen(appEnv.port, '0.0.0.0', function() {
  // print a message when the server starts listening
  console.log("server starting on " + appEnv.url);
});
```

package.json

```
{
		"name": "NodejsStarterApp",
		"version": "0.0.1",
		"private": true,
		"scripts": {
			"start": "node app.js“ //The javascript script to start the application
		},
		"dependencies": {
			"express": "4.13.x", //express web famework and cfenw framework are used
			"cfenv": "1.0.x"
		},
		"repository": {},
		"engines": {
			"node": "4.x“ //node.js engine version
		}
}
```

manifest.yml

```
applications:
  # look for the files to 'push' in the current directory
- path: .
  # max amount of memory to be used by the application
  memory: 256M
  # number of cloud instances to run the application
  instances: 1
  # domain name, mybluemix.net is the domain provided by Bluemix
  domain: mybluemix.net
  # the name of the application
  name: HelloWorld
  # the URL for the application will be <host>.<domain>
  # in this example, that's sunimalr.au-syd.bluemix.net
  host: sunimalr
  # storage limit for this application
  disk_quota: 1024M
```

Steps 7, 8, 9: Cloud Foundry and Bluemix

- Open command line and change directory into the folder with code downloaded in the previous step
- Use the CF CLI to connect to the Bluemix API (Depending on the region you are logged on to. ‘ng’ is for US South)
  - cf api https://api.ng.bluemix.net
- Login to Bluemix
  - cf login
  - use the IBM ID and password
- upload your application to Bluemix
  - cf push

Step 10: Hello World

- Go to your favorite browser and enter the URL:

  http://your_host_name.mybluemix.net

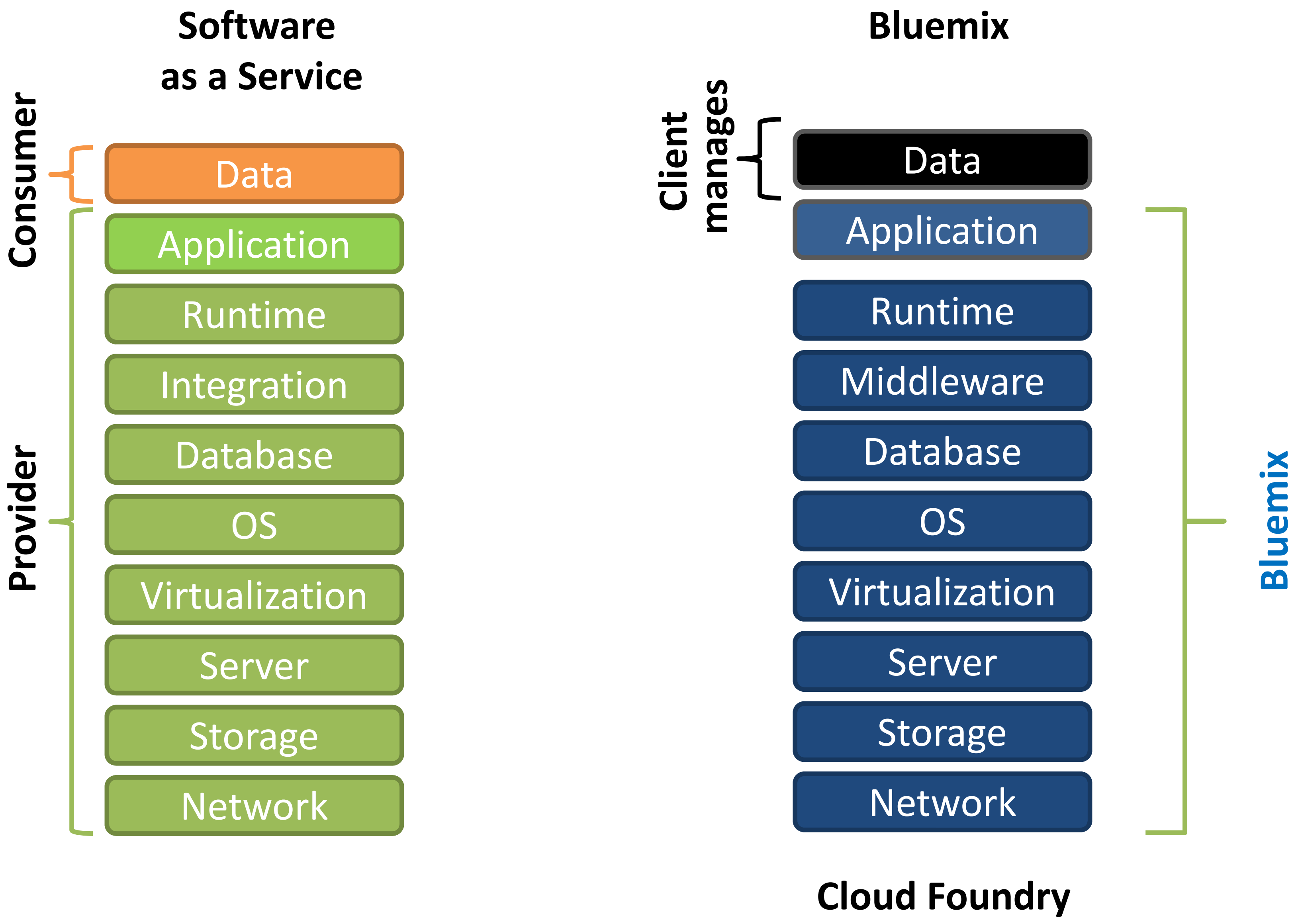
- Alternatively, you can try mine at

  http://sunimalr.mybluemix.net

SaaS: Analytics using dashDB Service

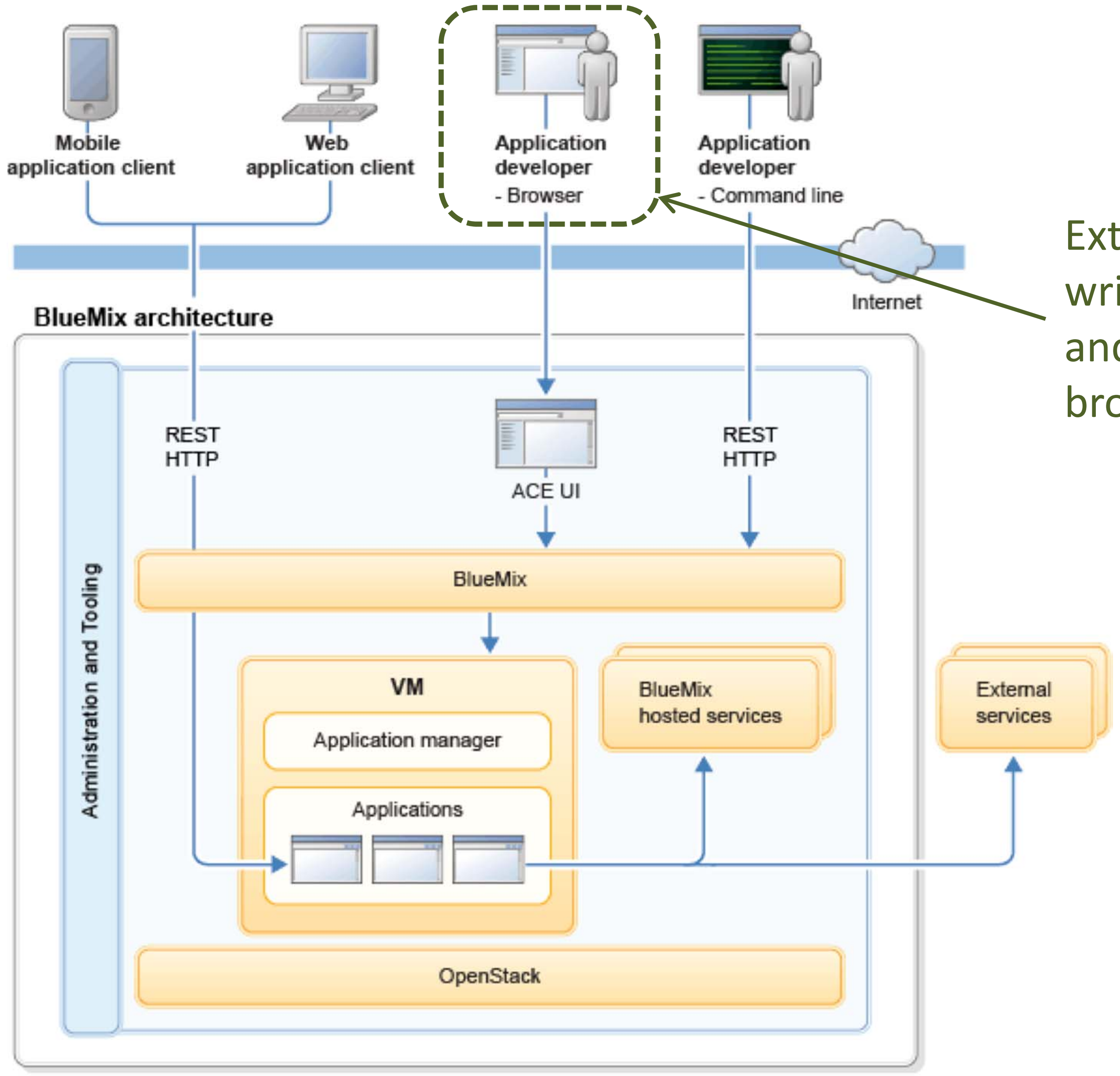
- Objective: use SaaS service (Data & Analytics) to perform data analysis
- SaaS service: dashDB is a “data warehousing and analytics solution”; stores relational data, supports SQL query and analytics using R
- Data: World development indicators for financial sector from World Bank; data is listed by countries with fields including currency, income groups, etc. Excel file: EdStats-Country
- Example: store data on dashDB, run SQL queries and visualize data using R

What is SaaS?



Software as a Service
Consumer
Provider
Data
Application
Runtime
Integration
Database
OS
Virtualization
Server
Storage
Network
Bluemix
Client manages
Data
Application
Runtime
Middleware
Database
OS
Virtualization
Server
Storage
Network
Bluemix
Cloud Foundry

SaaS: Analytics using dashDB Service

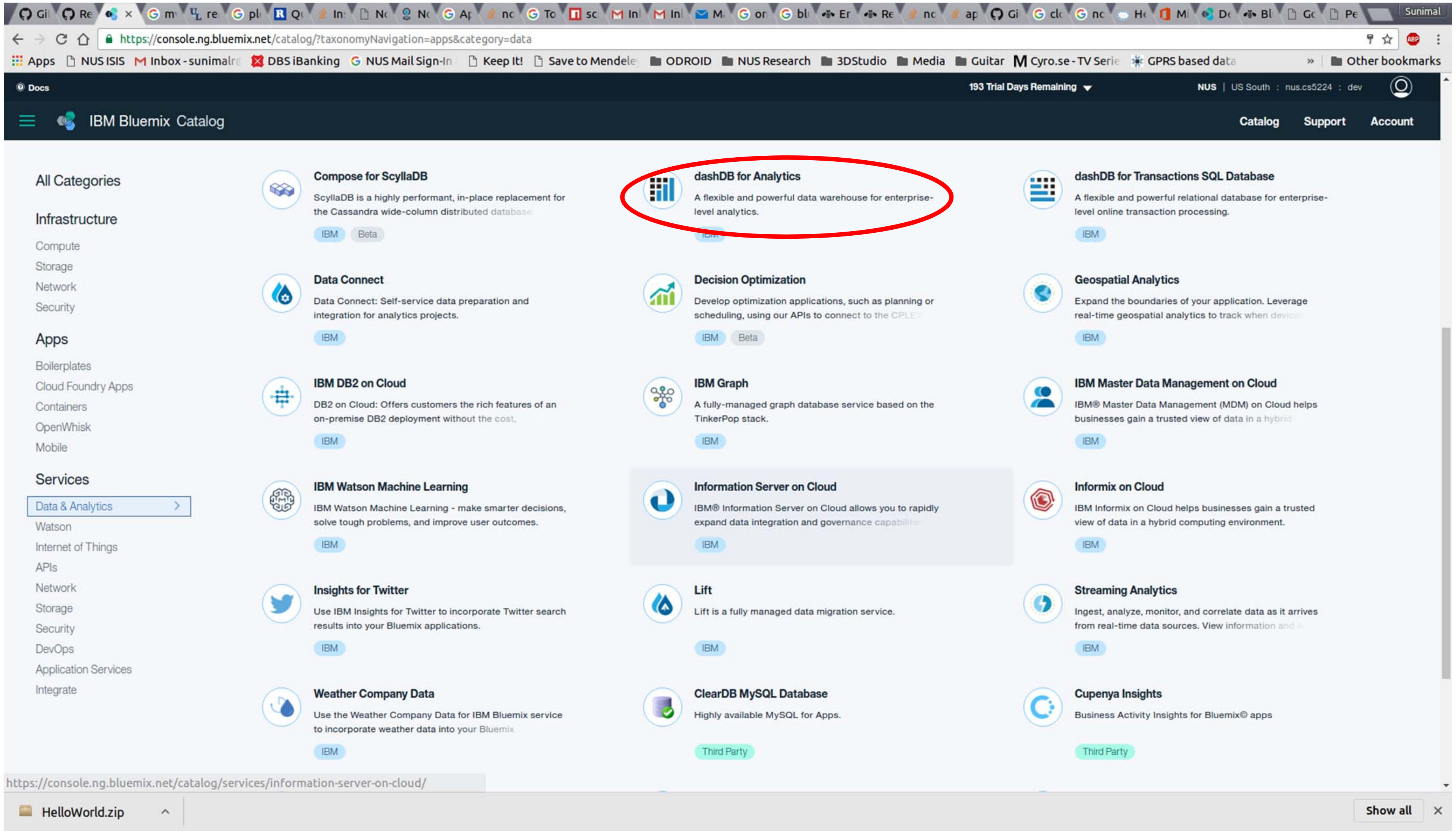


Extend SaaS to write SQL query and R code in web-browser

Main Steps: Analytics using dashDB

1. Login to Bluemix and go to Catalog
2. Select dashDB under Data & Analytics and setup
3. Connect web app to the dashDB service
4. Open dashDB console
5. Upload data (CSV or Excel)
6. Run queries (SQL)
7. Write R script to analyze data
8. Plot using R

Step 2: dashDB Setup



https://console.ng.bluemix.net/catalog/?taxonomyNavigation=apps&category=data

Apps | NUS ISIS | Inbox - sunimalr | DBS iBanking | NUS Mail Sign-In | Keep It! | Save to Mendeley | ODROID | NUS Research | 3DStudio | Media | Guitar | Cyro.se - TV Serie | GPRS based data | Other bookmarks

Docs | 193 Trial Days Remaining | NUS | US South : nus.cs5224 : dev

IBM Bluemix Catalog | Catalog | Support | Account

All Categories

Infrastructure: Compute, Storage, Network, Security

Apps: Boilerplates, Cloud Foundry Apps, Containers, OpenWhisk, Mobile

Services: Data & Analytics, Watson, Internet of Things, APIs, Network, Storage, Security, DevOps, Application Services, Integrate

Compose for ScyllaDB — ScyllaDB is a highly performant, in-place replacement for the Cassandra wide-column distributed database. IBM, Beta

dashDB for Analytics — A flexible and powerful data warehouse for enterprise-level analytics. IBM

dashDB for Transactions SQL Database — A flexible and powerful relational database for enterprise-level online transaction processing. IBM

Data Connect — Data Connect: Self-service data preparation and integration for analytics projects. IBM

Decision Optimization — Develop optimization applications, such as planning or scheduling, using our APIs to connect to the CPLEX IBM, Beta

Geospatial Analytics — Expand the boundaries of your application. Leverage real-time geospatial analytics to track when devices IBM

IBM DB2 on Cloud — DB2 on Cloud: Offers customers the rich features of an on-premise DB2 deployment without the cost, IBM

IBM Graph — A fully-managed graph database service based on the TinkerPop stack. IBM

IBM Master Data Management on Cloud — IBM® Master Data Management (MDM) on Cloud helps businesses gain a trusted view of data in a hybrid IBM

IBM Watson Machine Learning — IBM Watson Machine Learning - make smarter decisions, solve tough problems, and improve user outcomes. IBM

Information Server on Cloud — IBM® Information Server on Cloud allows you to rapidly expand data integration and governance capabilities IBM

Informix on Cloud — IBM Informix on Cloud helps businesses gain a trusted view of data in a hybrid computing environment. IBM

Insights for Twitter — Use IBM Insights for Twitter to incorporate Twitter search results into your Bluemix applications. IBM

Lift — Lift is a fully managed data migration service. IBM

Streaming Analytics — Ingest, analyze, monitor, and correlate data as it arrives from real-time data sources. View information and IBM

Weather Company Data — Use the Weather Company Data for IBM Bluemix service to incorporate weather data into your Bluemix IBM

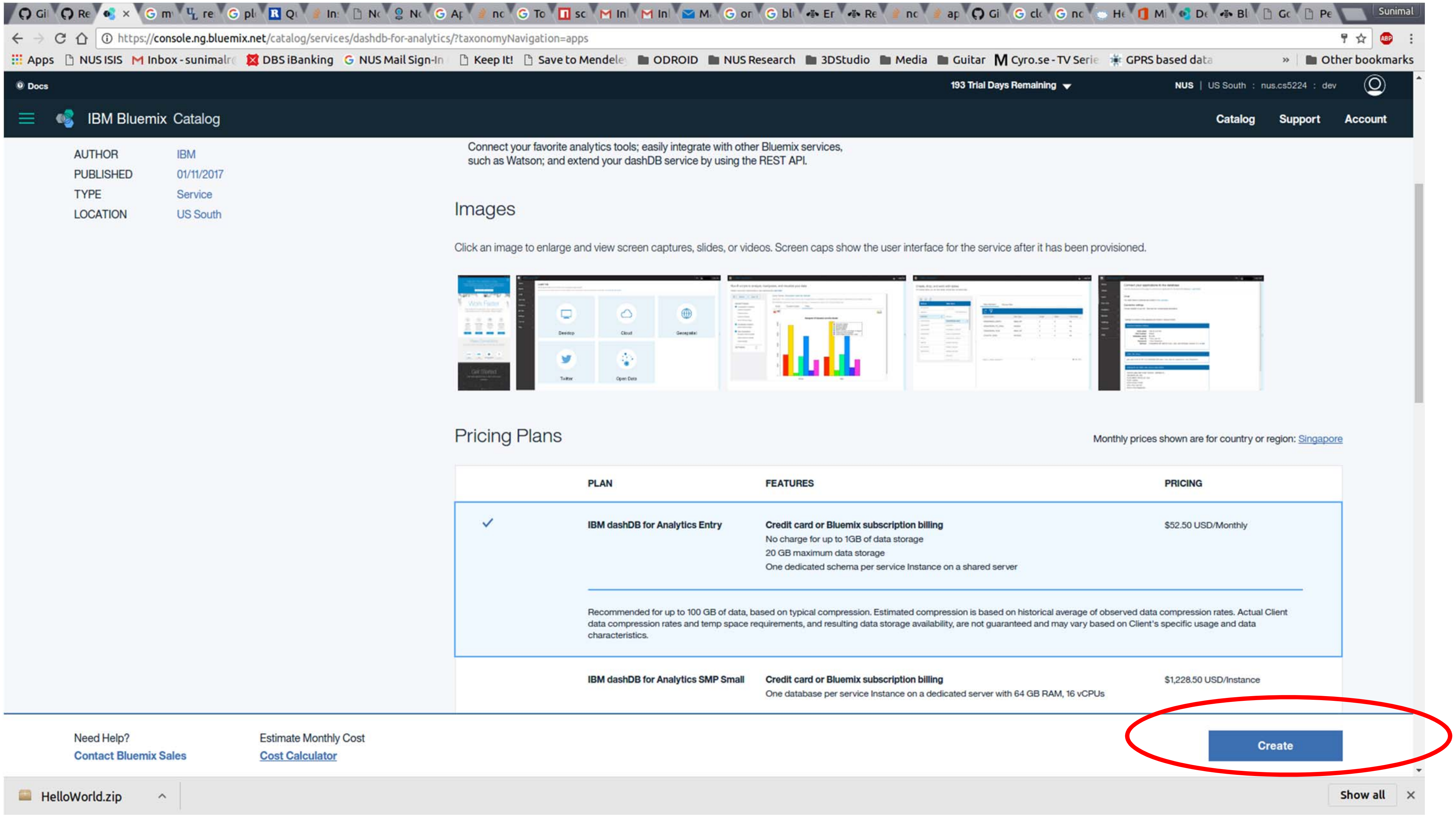
ClearDB MySQL Database — Highly available MySQL for Apps. Third Party

Cupenya Insights — Business Activity Insights for Bluemix© apps. Third Party

https://console.ng.bluemix.net/catalog/services/information-server-on-cloud/

HelloWorld.zip | Show all

Step 2: dashDB Setup



AUTHOR	IBM
PUBLISHED	01/11/2017
TYPE	Service
LOCATION	US South

Connect your favorite analytics tools; easily integrate with other Bluemix services, such as Watson; and extend your dashDB service by using the REST API.

Images

Click an image to enlarge and view screen captures, slides, or videos. Screen caps show the user interface for the service after it has been provisioned.

Pricing Plans

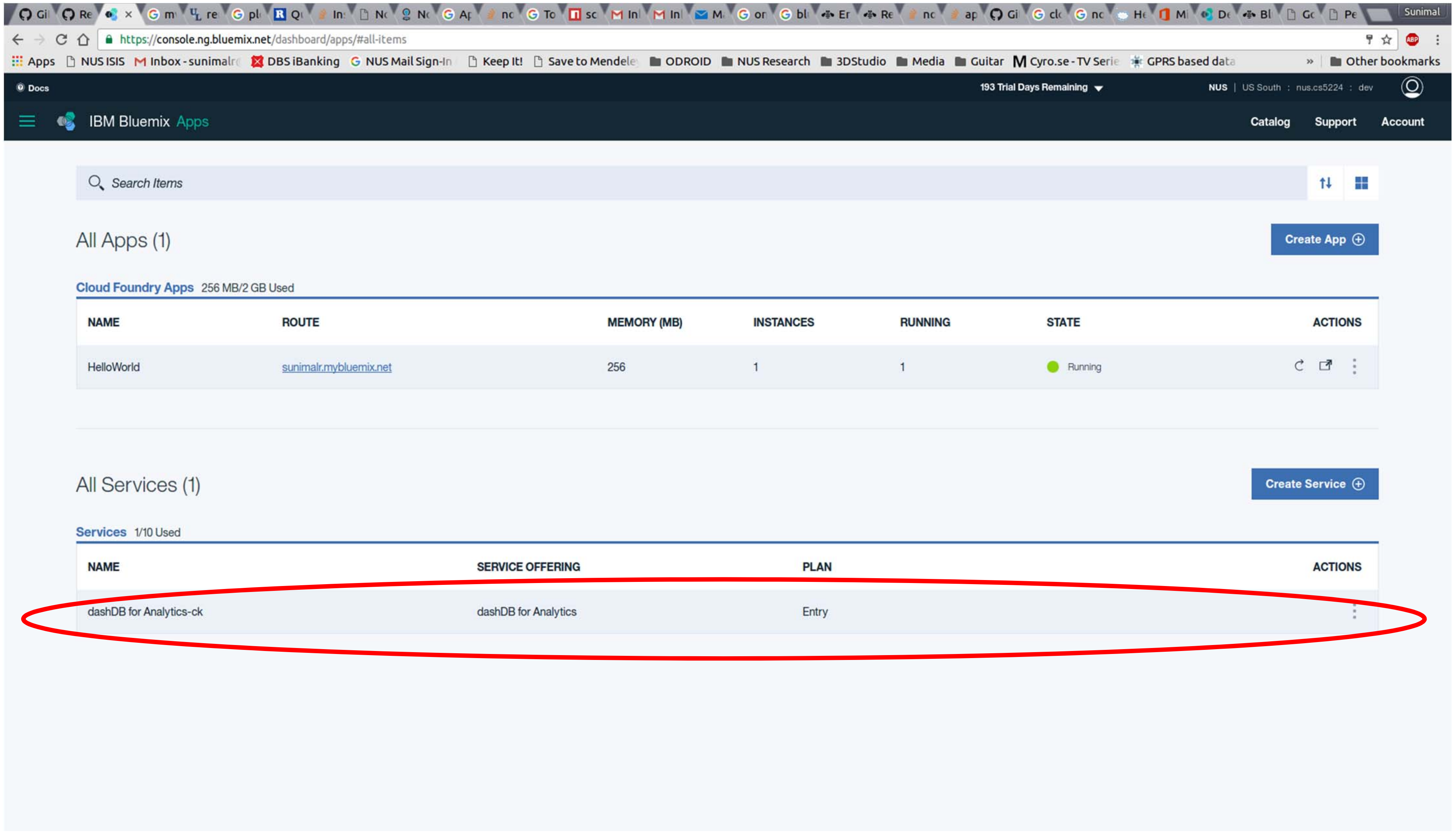
Monthly prices shown are for country or region: Singapore

	PLAN	FEATURES	PRICING
✓	IBM dashDB for Analytics Entry	**Credit card or Bluemix subscription billing** No charge for up to 1GB of data storage 20 GB maximum data storage One dedicated schema per service Instance on a shared server	$52.50 USD/Monthly
	Recommended for up to 100 GB of data, based on typical compression. Estimated compression is based on historical average of observed data compression rates. Actual Client data compression rates and temp space requirements, and resulting data storage availability, are not guaranteed and may vary based on Client's specific usage and data characteristics.		
	IBM dashDB for Analytics SMP Small	**Credit card or Bluemix subscription billing** One database per service Instance on a dedicated server with 64 GB RAM, 16 vCPUs	$1,228.50 USD/Instance

Need Help?
Contact Bluemix Sales

Estimate Monthly Cost
Cost Calculator

Step 2: dashDB Setup



All Apps (1)

Cloud Foundry Apps 256 MB/2 GB Used

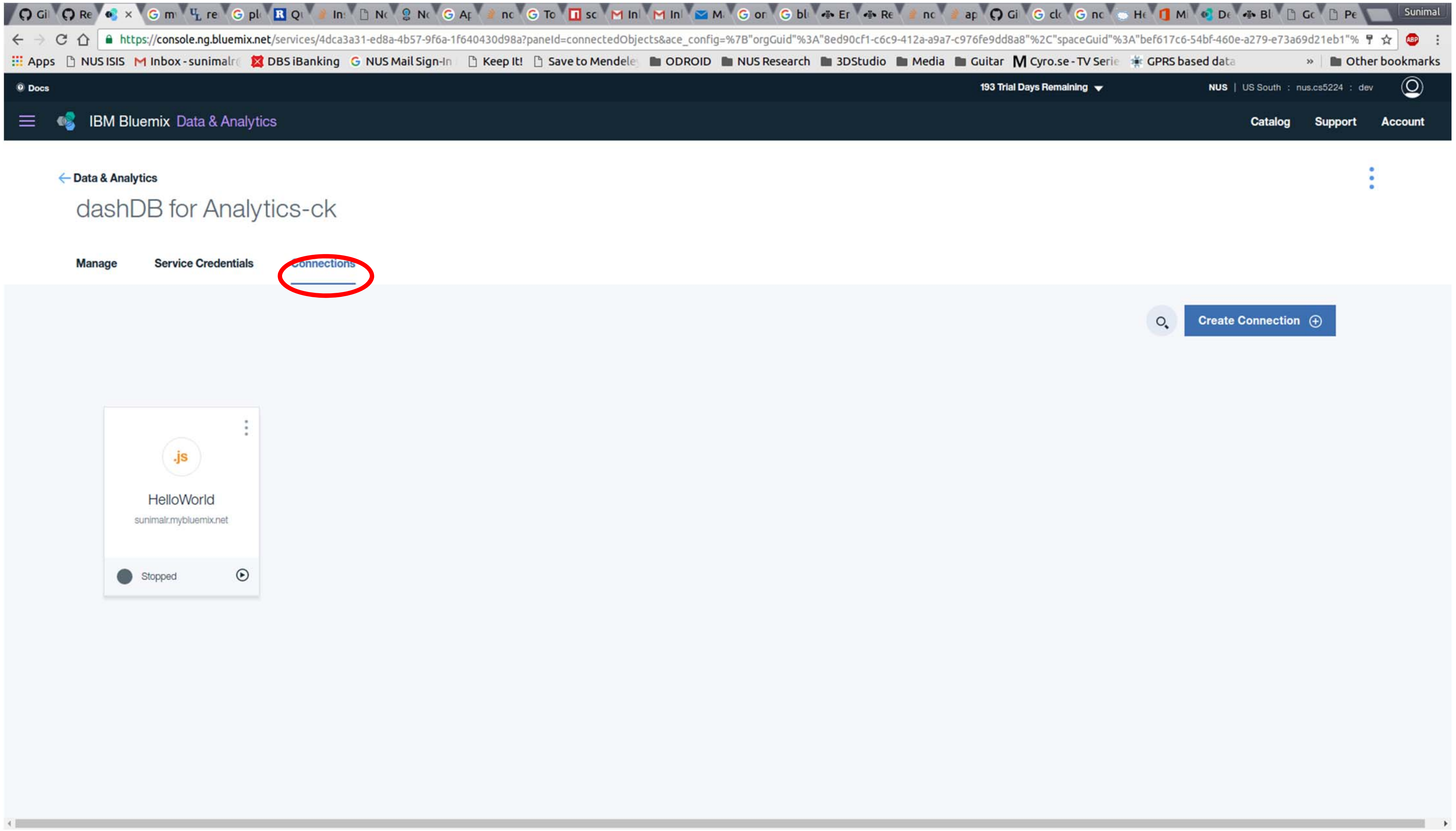
NAME	ROUTE	MEMORY (MB)	INSTANCES	RUNNING	STATE	ACTIONS
HelloWorld	sunimalr.mybluemix.net	256	1	1	● Running	

All Services (1)

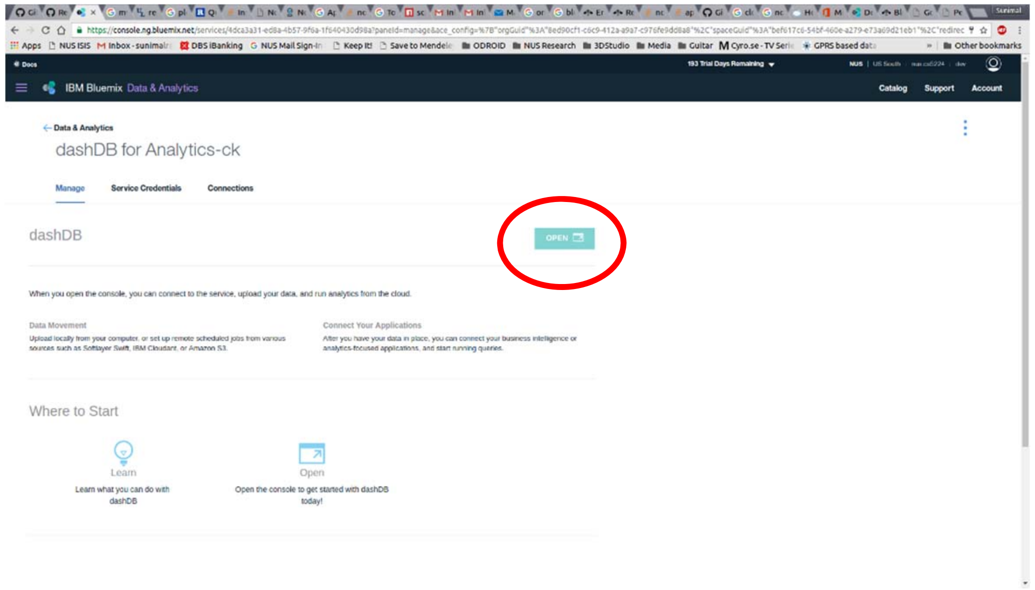
Services 1/10 Used

NAME	SERVICE OFFERING	PLAN	ACTIONS
dashDB for Analytics-ck	dashDB for Analytics	Entry	

Step 3: Connect Web App



Step 4: Open dashDB Console


IBM Bluemix Data & Analytics

Data & Analytics

dashDB for Analytics-ck

Manage | Service Credentials | Connections

dashDB

OPEN

When you open the console, you can connect to the service, upload your data, and run analytics from the cloud.

Data Movement

Upload locally from your computer, or set up remote scheduled jobs from various sources such as Softlayer Swift, IBM Cloudant, or Amazon S3.

Connect Your Applications

After you have your data in place, you can connect your business intelligence or analytics-focused applications, and start running queries.

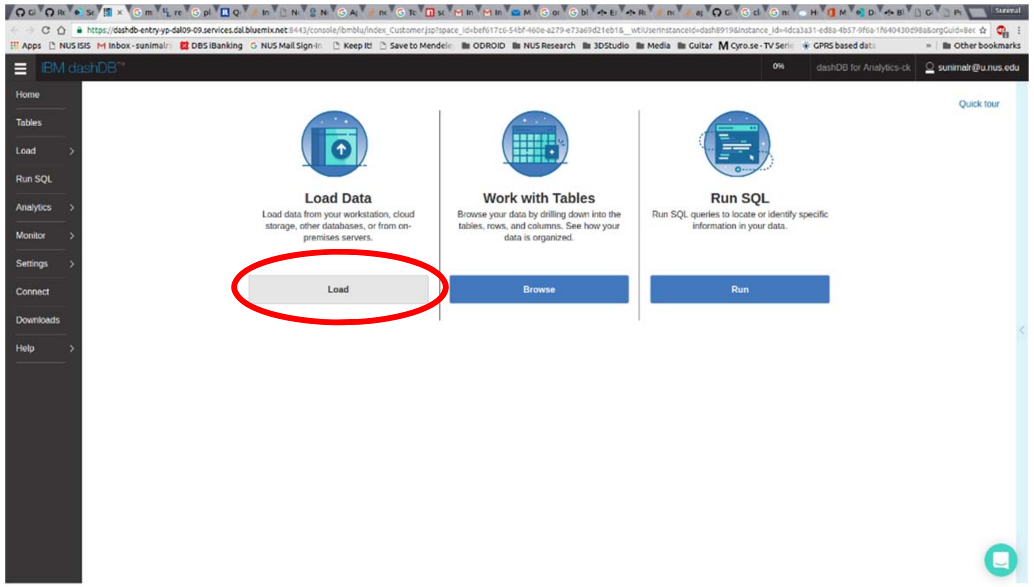
Where to Start

Learn

Learn what you can do with dashDB

Open

Open the console to get started with dashDB today!


IBM dashDB™

dashDB for Analytics-ck | sunimalr@u.nus.edu

Home, Tables, Load, Run SQL, Analytics, Monitor, Settings, Connect, Downloads, Help

Quick tour

Load Data

Load data from your workstation, cloud storage, other databases, or from on-premises servers.

Load

Work with Tables

Browse your data by drilling down into the tables, rows, and columns. See how your data is organized.

Browse

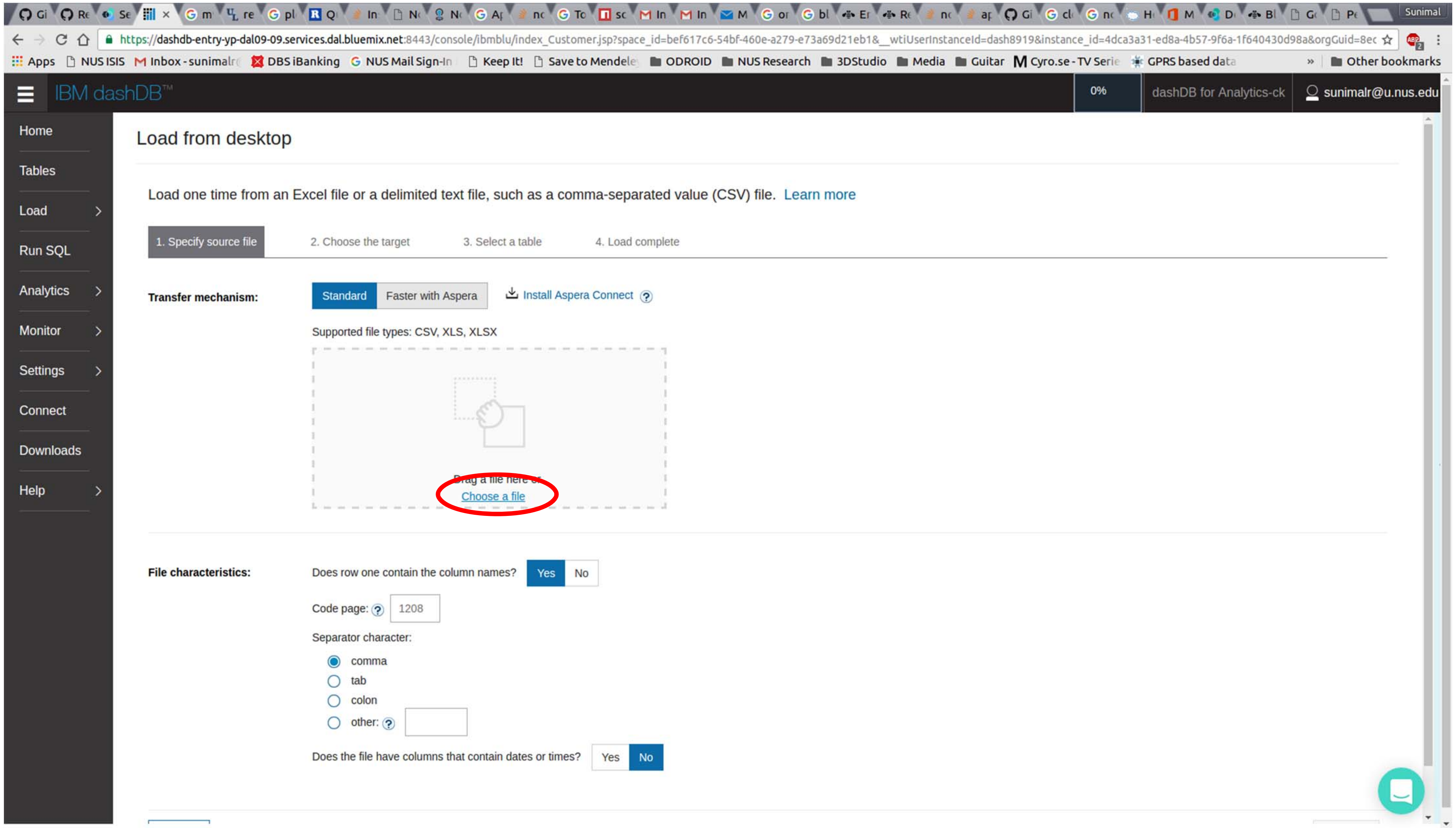
Run SQL

Run SQL queries to locate or identify specific information in your data.

Run

Step 5: Upload data

- First download .csv data file from the course webpage, then upload to dashDB



Load from desktop

Load one time from an Excel file or a delimited text file, such as a comma-separated value (CSV) file. Learn more

1. Specify source file 2. Choose the target 3. Select a table 4. Load complete

Transfer mechanism: [✓ Standard] Faster with Aspera

Supported file types: CSV, XLS, XLSX

Drag a file here or Choose a file

File characteristics:

Does row one contain the column names? [✓ Yes] No

Code page: 1208

Separator character: [✓ comma] tab colon other:

Does the file have columns that contain dates or times? Yes [✓ No]

Step 5: Upload Data

- Browse files, select the necessary parameters, upload file
- Preview
- Next
  - First time define a new table and load
  - For second file, use existing table and load
- Click the column headings to rename the columns, if needed
- Click Finish

Step 6: Run SQL Query

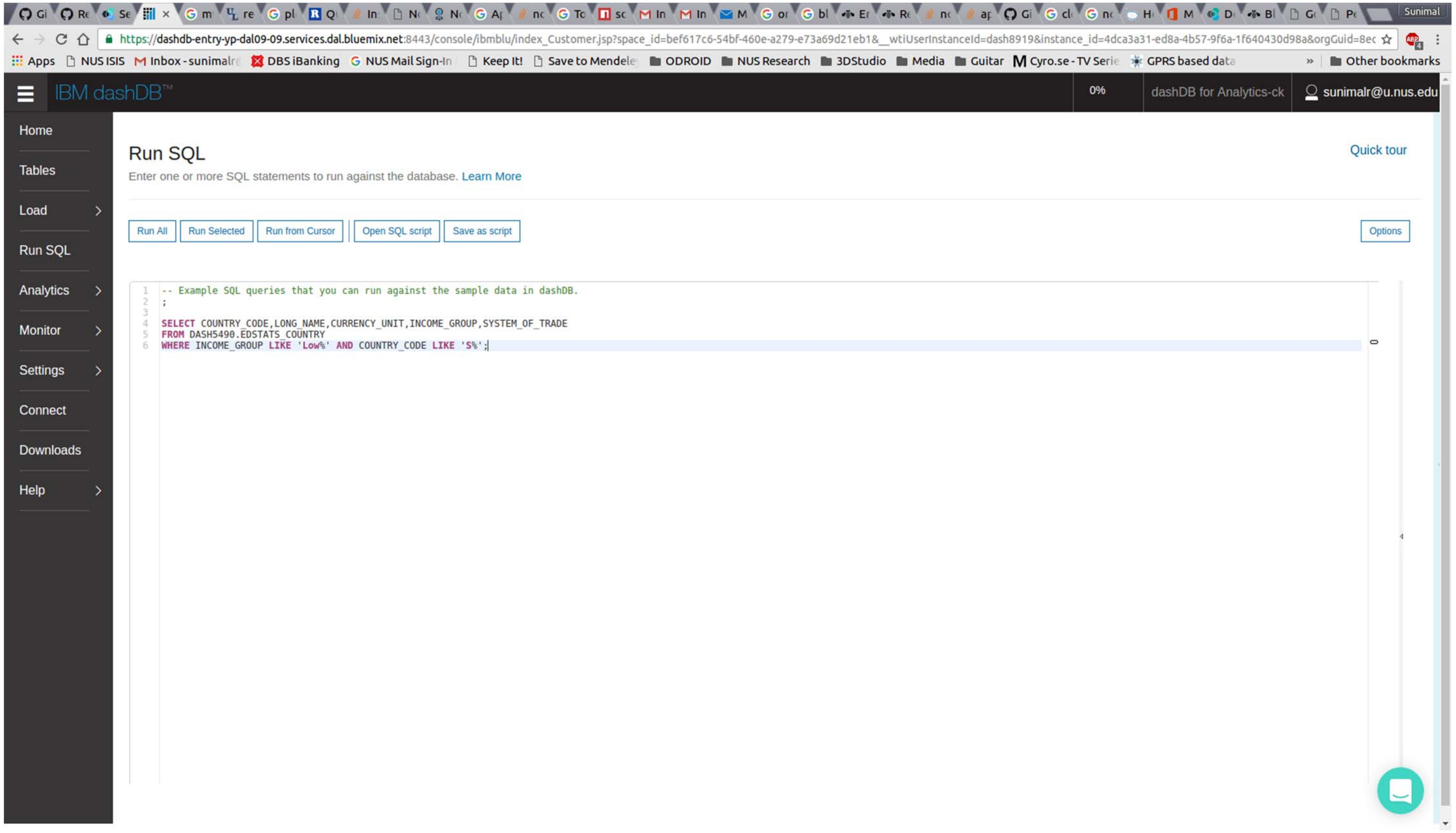
- Find out countries that fall under the low income group and has their country code starting with S

```
SELECT
COUNTRY_CODE,LONG_NAME,CURRENCY_UNIT,INCOME_GROUP,SYSTEM_OF
_TRADE
FROM DASH5490.EDSTATS_COUNTRY
WHERE INCOME_GROUP LIKE 'Low%' AND COUNTRY_CODE LIKE 'S%';
```

- Data file: EdStats-Country

Database name should be set as required by referring to the dashdb tables.

Step 6: Run Query (SQL)

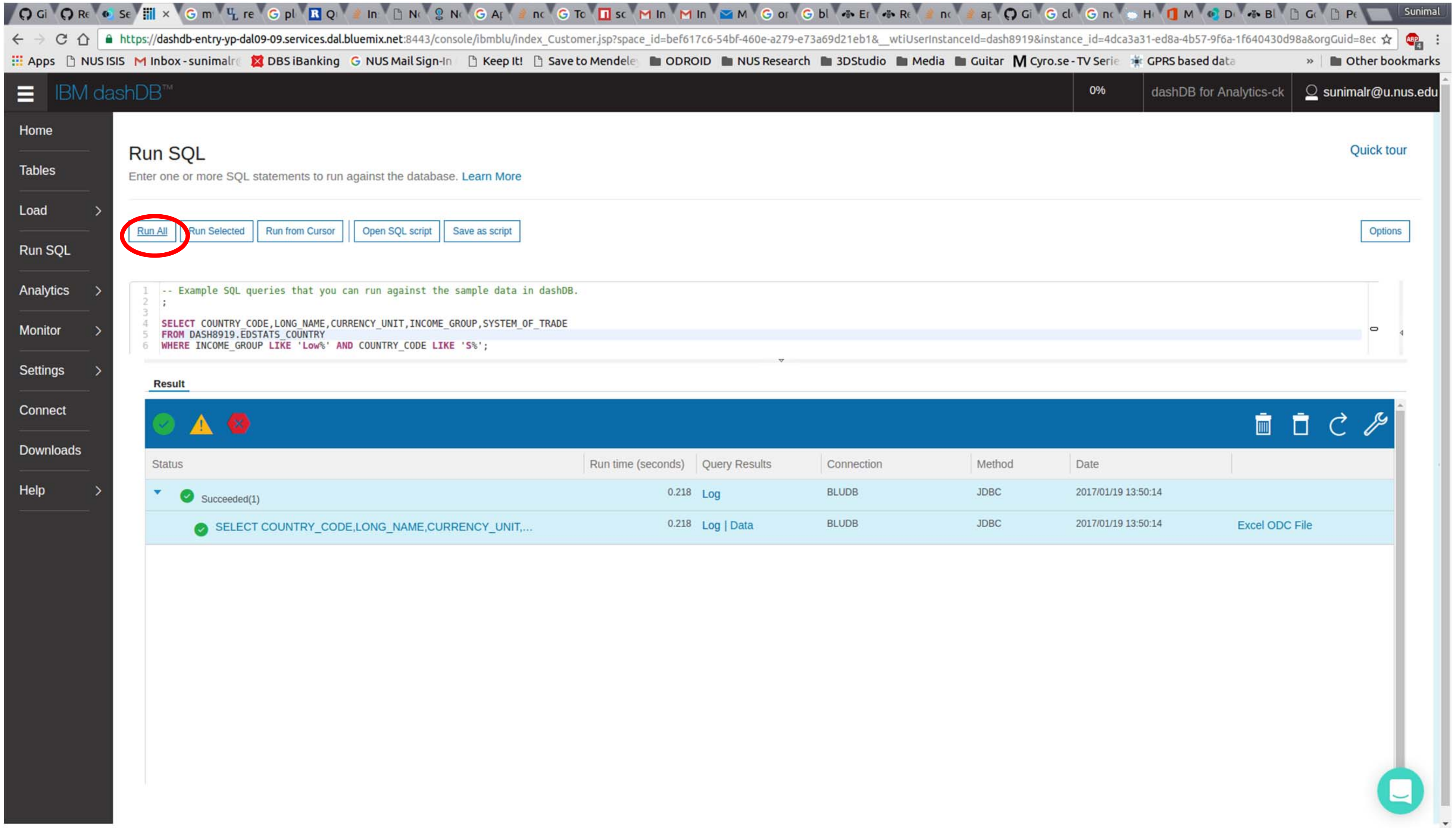


Run SQL

Enter one or more SQL statements to run against the database. Learn More

```
-- Example SQL queries that you can run against the sample data in dashDB.
;

SELECT COUNTRY_CODE,LONG_NAME,CURRENCY_UNIT,INCOME_GROUP,SYSTEM_OF_TRADE
FROM DASH5490.EDSTATS_COUNTRY
WHERE INCOME_GROUP LIKE 'Low%' AND COUNTRY_CODE LIKE 'S%';
```

Step 6: Run Query (SQL)



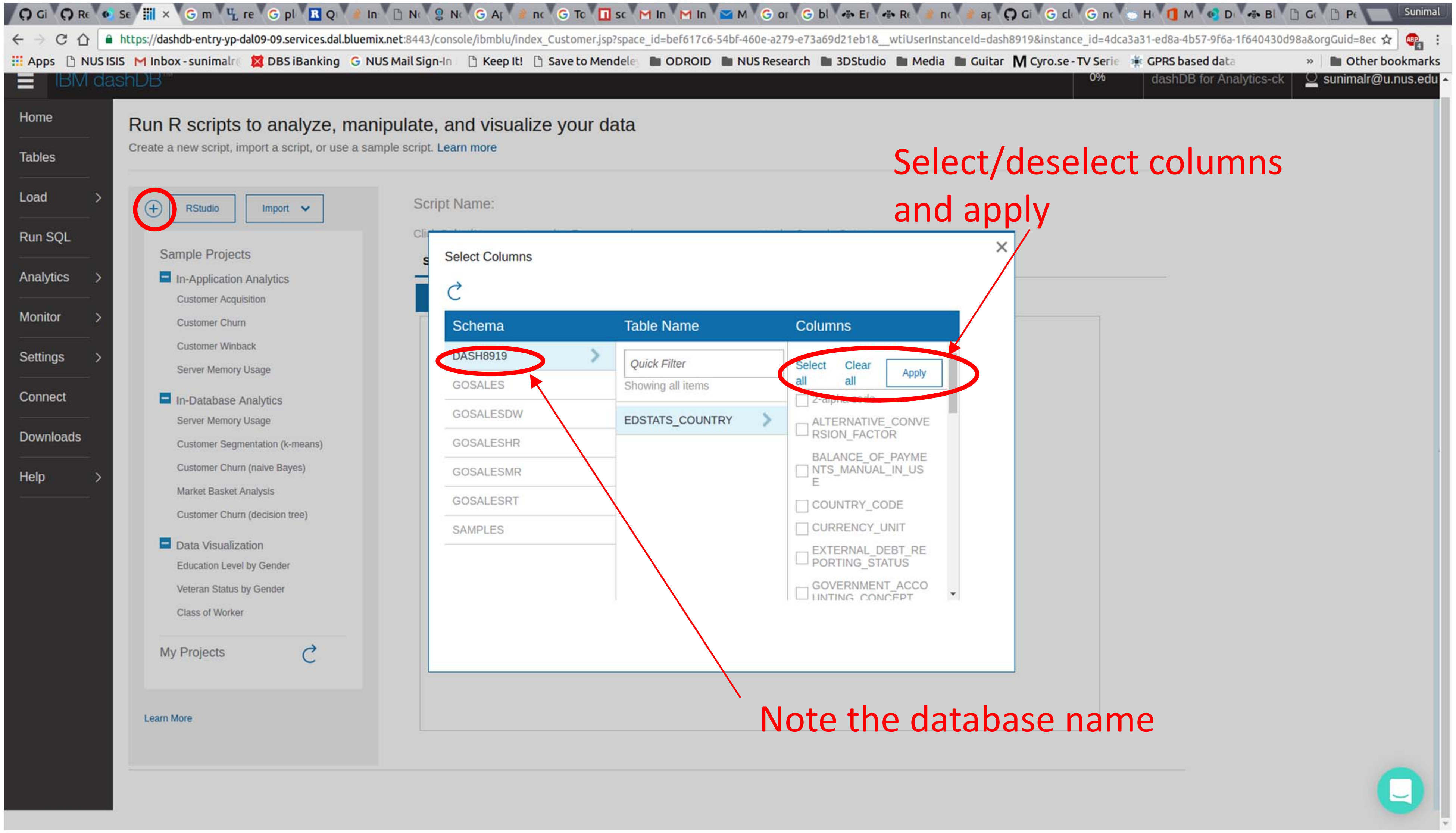
Run SQL

Enter one or more SQL statements to run against the database. Learn More

```
-- Example SQL queries that you can run against the sample data in dashDB.
;

SELECT COUNTRY_CODE,LONG_NAME,CURRENCY_UNIT,INCOME_GROUP,SYSTEM_OF_TRADE
FROM DASH8919.EDSTATS_COUNTRY
WHERE INCOME_GROUP LIKE 'Low%' AND COUNTRY_CODE LIKE 'S%';
```

Status	Run time (seconds)	Query Results	Connection	Method	Date	
Succeeded(1)	0.218	Log	BLUDB	JDBC	2017/01/19 13:50:14	
SELECT COUNTRY_CODE,LONG_NAME,CURRENCY_UNIT,...	0.218	Log | Data	BLUDB	JDBC	2017/01/19 13:50:14	Excel ODC File

Step 7: Write R Script



Run R scripts to analyze, manipulate, and visualize your data

Create a new script, import a script, or use a sample script. Learn more

Select Columns

Schema	Table Name	Columns
DASH8919	EDSTATS_COUNTRY	ALTERNATIVE_CONVERSION_FACTOR
GOSALES		BALANCE_OF_PAYMENTS_MANUAL_IN_USE
GOSALESDW		COUNTRY_CODE
GOSALESHR		CURRENCY_UNIT
GOSALESMR		EXTERNAL_DEBT_REPORTING_STATUS
GOSALESRT		GOVERNMENT_ACCOUNTING_CONCEPT
SAMPLES		

Select/deselect columns and apply

Note the database name

Step 7 : Write R script

- Click ⊕ , Select columns , click apply
  - Example template auto-generated with selected columns

- # load the ibmdbR library
  - library(ibmdbR)
- #create a connection object for the database
  - mycon <- idaConnect("BLUDB", "", "")
  - idaInit(mycon)

- # and create an R data frame based on the columns that you selected.
  - data_frame1<- as.data.frame(ida.data.frame('"DASH5490"."EDSTATS_COUNTRY"')[,c('CURRENCY_UNIT', 'INCOME_GROUP', 'REGION', 'SHORT_NAME')])

Change autogenerated name to data_frame1

Step 8: Write R script to Plot Data

- # Load the plotting package
  - library(ggplot2)

- # Plot the count of countries across different income groups
  - #Classify the levels and label them: Create a factor

    ```
    data_frame1<-
    as.data.frame(ida.data.frame('"DASH5490"."EDSTATS_COUNTRY"')[
    ,c('CURRENCY_UNIT', 'INCOME_GROUP', 'REGION', 'SHORT_NAME')])

    data_frame1$"INCOME_GROUP" <-
    factor(data_frame1$"INCOME_GROUP",levels=c("Low income","Lower
    middle income","Upper middle income","High income: nonOECD","High
    income: OECD"), labels=c("LI","LMI","UMI","HIN","HIO"))
    ```

  - #Count the income groups

    ```
    counts <- table(data_frame1$"INCOME_GROUP")
    ```

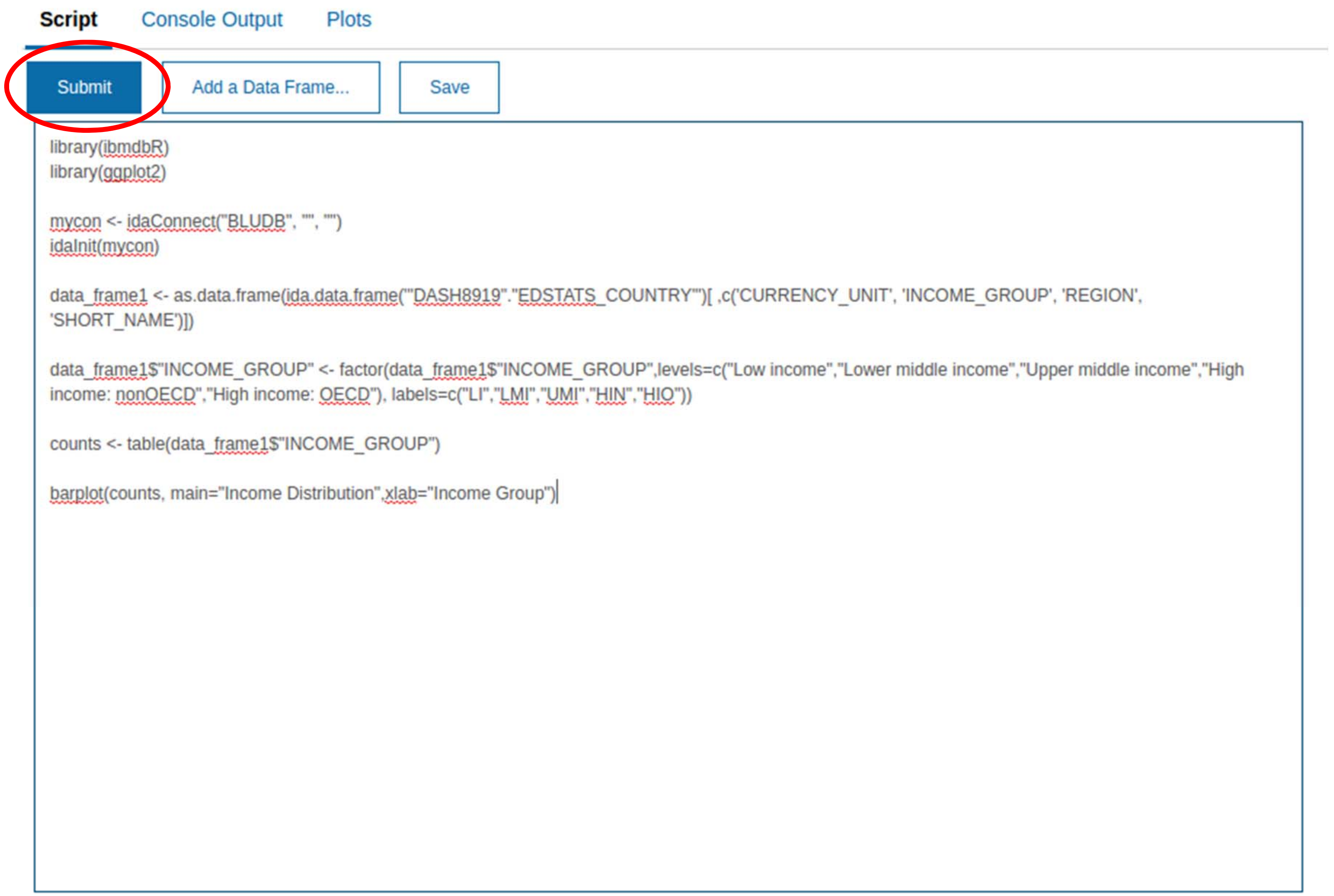
  - #Plot counts versus the labelled income groups

    ```
    barplot(counts, main="Income Distribution",xlab="Income Group")
    ```

Step 8: Complete R Script

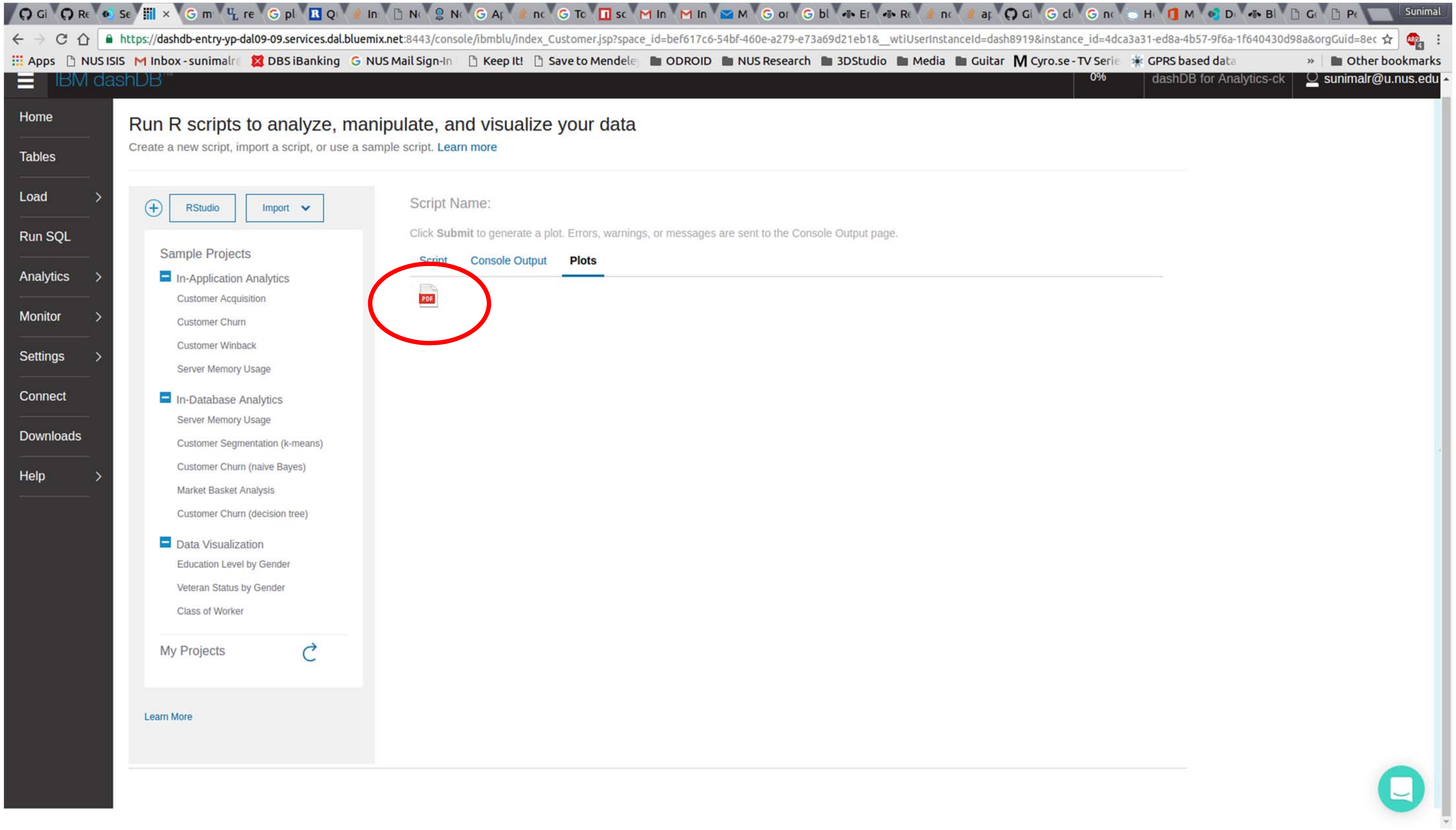
```
library(ibmdbR)
library(ggplot2)

mycon <- idaConnect("BLUDB", "", "")
idaInit(mycon)

data_frame1<-
as.data.frame(ida.data.frame('"DASH5490"."EDSTATS_COUNTRY"')[
,c('CURRENCY_UNIT', 'INCOME_GROUP', 'REGION', 'SHORT_NAME')])

data_frame1$"INCOME_GROUP" <-
factor(data_frame1$"INCOME_GROUP",levels=c("Low income","Lower
middle income","Upper middle income","High income: nonOECD","High
income: OECD"), labels=c("LI","LMI","UMI","HIN","HIO"))

counts <- table(data_frame1$"INCOME_GROUP")
barplot(counts, main="Income Distribution",xlab="Income Group")
```

Step 8: Plot



```
library(ibmdbR)
library(ggplot2)

mycon <- idaConnect("BLUDB", "", "")
idaInit(mycon)

data_frame1 <- as.data.frame(ida.data.frame('"DASH8919"."EDSTATS_COUNTRY"')[ ,c('CURRENCY_UNIT', 'INCOME_GROUP', 'REGION',
'SHORT_NAME')])

data_frame1$"INCOME_GROUP" <- factor(data_frame1$"INCOME_GROUP",levels=c("Low income","Lower middle income","Upper middle income","High
income: nonOECD","High income: OECD"), labels=c("LI","LMI","UMI","HIN","HIO"))

counts <- table(data_frame1$"INCOME_GROUP")

barplot(counts, main="Income Distribution",xlab="Income Group")
```

Step 8: Download the Result



Run R scripts to analyze, manipulate, and visualize your data

Create a new script, import a script, or use a sample script. Learn more

Script Name:

Click **Submit** to generate a plot. Errors, warnings, or messages are sent to the Console Output page.

Summary

- overview of BlueMix – main features
- hands-on with BlueMix
- web server using node.js boilerplate
- simple SQL query using dashDB service of BlueMix
- simple R script to plot analysis results of data on dashDB

References

- An Updated Overview and Demonstration of IBM Bluemix, Youtube video, Feb 2016
- IBM Bluemix The Cloud Platform for Creating and Delivering Applications, IBM Redbooks, 2015.
- Bluemix users: https://www.ibm.com/cloud-computing/bluemix/case-studies
- IBM dashDB:
  http://www-01.ibm.com/support/knowledgecenter/SS6NHC/com.ibm.swg.im.dashdb.kc.doc/welcome.html
- SQL Query: http://www.w3schools.com/sql/
- R Language:
  http://cran.r-project.org/doc/manuals/r-release/R-intro.html
- Plotting: http://docs.ggplot2.org/current/