Quick2Cloud for Cloud Systems

Built on IBM Cloud

CLOUD FOUNDRY

Quick2Cloud College - Course 101

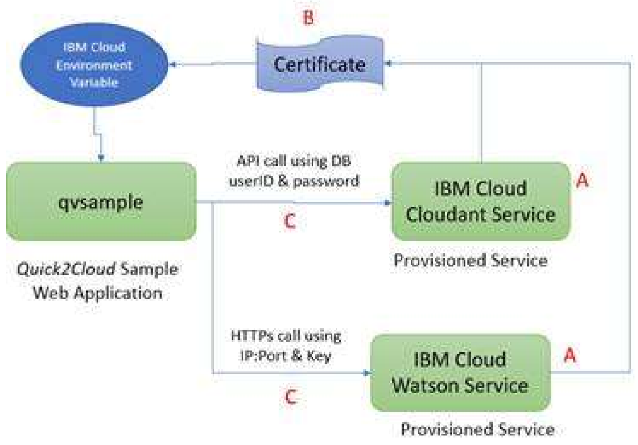
Provisioning Services to your IBM Cloud Application

Now that you have an application running in IBM□s Cloud you may consider providing it with more capability in the form of callable Services. These Services live as a set of external programs that are available as part of IBM□s Cloud offering. Such capabilities span the gambit of things you might want to have your application partake in, starting with connecting databases to communicating with IBM Watson; and everything in-between.

In this course, you will learn it□s easy, with *Quick2Cloud*□*s* assistance, to add these Services along with their credentials and application bindings. You will want to check with the IBM Service Catalog to locate various Services that might interest you plus the catalog supplies documentation on how they work. When you find Services that bolster your applications capabilities, let *Quick2Cloud* provide the guidance in getting those Services connected to your application.

Here shows what this course will accomplish. You will provision two IBM Services for your *Quick2Cloud* Web Application to use.

Provisioning and using different IBM Services


B
Certificate
IBM Cloud Environment Variable
qvsample
Quick2Cloud Sample Web Application
API call using DB userID & password
C
IBM Cloud Cloudant Service
A
Provisioned Service
HTTPs call using IP:Port & Key
C
IBM Cloud Watson Service
A
Provisioned Service

In Step A, you will provision both IBM Cloud Services. Step B, you will generate security certificates that allow communication to those services. The security information will exist and be passed to your application as Cloud environment variables. Step C, you will use that security information to call those services over Rest or HPPTs protocols.

The steps to Provision an IBM Service follows:

1. Instantiate the Service
2. Generate Credentials
3. Bind to your Application
4. Save Credentials Locally
5. Restage Application

Quick2Cloud will demonstrate provisioning two different types of IBM Services. This course will guide you into provisioning an API driven Service then tackle a slightly simpler offering using a REST interface; all the while showing the steps that are required are actually common to provisioning any IBM Service. Once you understand these two Services you should be able to provision the rest in the same fashion.

The two Services to be used in this course are:

Cloudant - object oriented json based data store

Watson Language Translation - translates English text to Spanish text

When using a Cloudant database, you will be connecting the database to your *Quick2Cloud* sample web application for the purpose of demonstration. In doing so, as a byproduct of that demonstration, your application will use the database to persist how many page hits it receives to help you understand your application□s traffic rates.

Later in this course you will learn how to programmatically extract Service information from Cloud environment variables enabling the application the use of your Services directly from its code.

IBM Cloud supports several styles of Services. In the case of the Cloudant Service, IBM will provision, on your behalf, a Cloudant instance for your application□s use in the same region your application is running making the database relatively easy to connect with via API calls. For Cloudant, language specific APIs are externally provided to your application and with those APIs you can simply access the database inline as part of your codes logic.

Another style of IBM Service exists, such as Watson□s Language Translation, which runs in a Watson Cloud Center outside your region providing your application just a URL and API key as its instance. Watson then becomes more like a secure transactional server for your application□s use. In this style of Service, you don□t need an external API interface; just a simple REST call targeted at Watson will do. Thus, depending on which IBM Service that you select from the catalog will depend on how it is provisioned and accessed from your application.

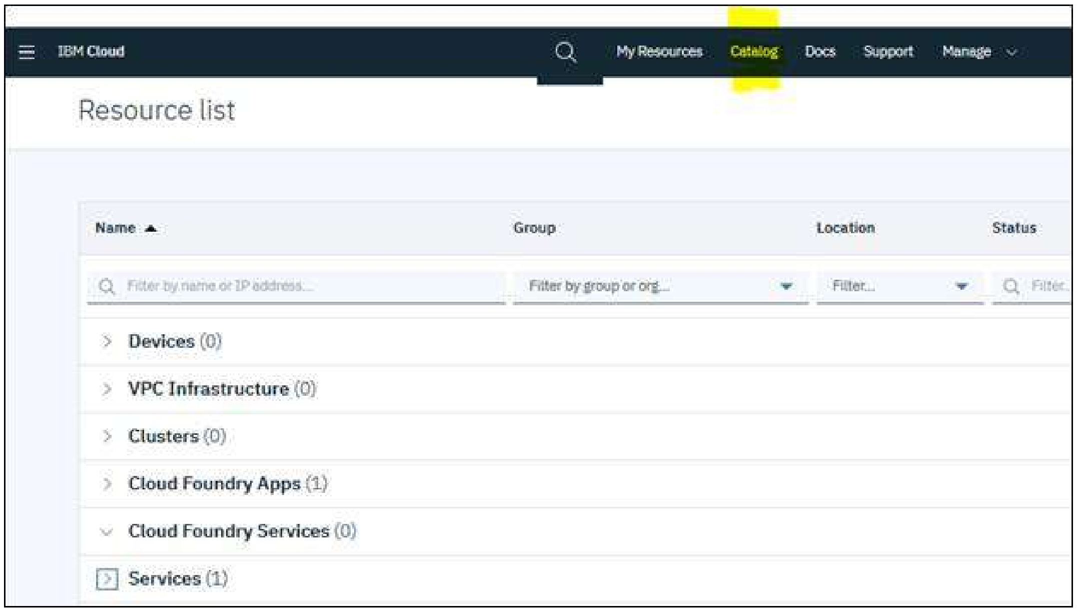
There are several steps to getting an IBM Service up and working with your application. *Quick2Cloud* will go through each step and explain their purpose so you get a good feeling on how all this all works.

Let□s start going through the steps beginning with the Cloudant Service.

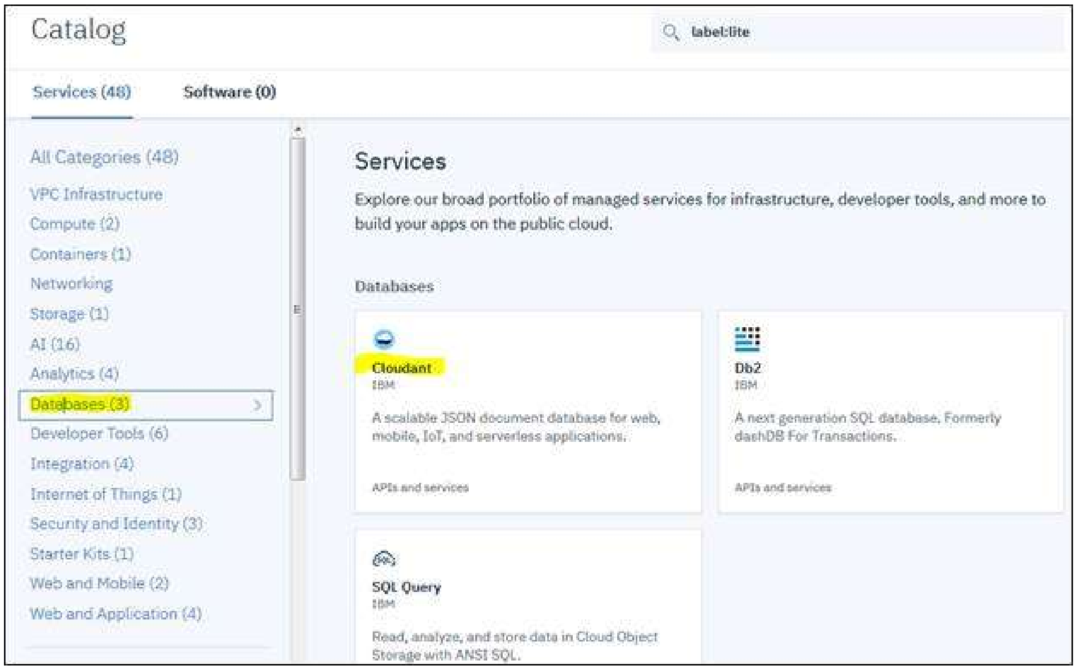
Step 0 - Provisioning from a Catalog

You will start by locating the Cloudant Service inside the IBM Service Catalog. The catalog is located on the IBM Cloud Dashboard as shown below or you can use the link provided. Some of the Services are developed directly by IBM and many others are supplied by IBM business partners. Still other Services are donated by individual users. Services can be very broad such as our robust Cloudant database with others more focused to just one capability like language translation. With each catalog entry, you will find links to documentation on how to interface and use the Service and it is suggested that you completely read the document on Cloudant before you start modifying your sample application code.

Click on the Catalog link in yellow or use the link above.



Select the Database Service category to list all databases provided. Click on Cloudant for it is the one we are using in our course.



Catalog

label:lite

Services (48) Software (0)

All Categories (48)
VPC Infrastructure
Compute (2)
Containers (1)
Networking
Storage (1)
AI (16)
Analytics (4)
Databases (3)
Developer Tools (6)
Integration (4)
Internet of Things (1)
Security and Identity (3)
Starter Kits (1)
Web and Mobile (2)
Web and Application (4)

Services

Explore our broad portfolio of managed services for infrastructure, developer tools, and more to build your apps on the public cloud.

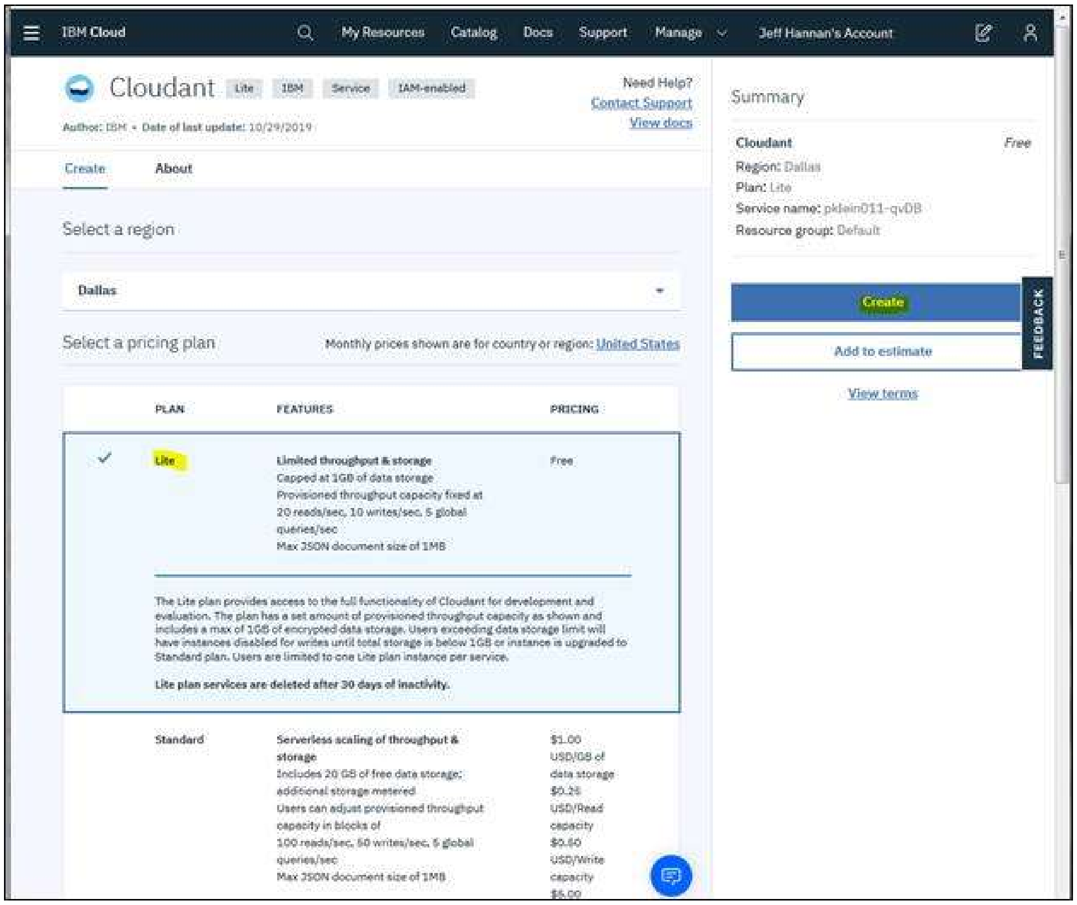
Databases

Cloudant
IBM
A scalable JSON document database for web, mobile, IoT, and serverless applications.
APIs and services

Db2
IBM
A next generation SQL database. Formerly dashDB For Transactions.
APIs and services

SQL Query
IBM
Read, analyze, and store data in Cloud Object Storage with ANSI SQL.

After selecting Cloudant, you will reach the Cloudant Plan and Provisioning page below. Each Service in the IBM Cloud has such a page for provisioning. By default the plan selected is the Lite Plan that has no-costs associated with it. Should you want to upgrade later to a more robust database based on additional read & write activity you can select one of the other plans. For this course the Lite Plan works fine.



IBM Cloud | My Resources | Catalog | Docs | Support | Manage | Jeff Hannan's Account

Cloudant Lite IBM Service IAM-enabled

Author: IBM • Date of last update: 10/29/2019

Need Help?
Contact Support
View docs

Create | About

Select a region

Dallas

Select a pricing plan

Monthly prices shown are for country or region: United States

	PLAN	FEATURES	PRICING
✓	Lite	**Limited throughput & storage** Capped at 1GB of data storage Provisioned throughput capacity fixed at 20 reads/sec, 10 writes/sec, 5 global queries/sec Max JSON document size of 1MB	Free
		The Lite plan provides access to the full functionality of Cloudant for development and evaluation. The plan has a set amount of provisioned throughput capacity as shown and includes a max of 1GB of encrypted data storage. Users exceeding data storage limit will have instances disabled for writes until total storage is below 1GB or instance is upgraded to Standard plan. Users are limited to one Lite plan instance per service. **Lite plan services are deleted after 30 days of inactivity.**	
	Standard	**Serverless scaling of throughput & storage** Includes 20 GB of free data storage; additional storage metered Users can adjust provisioned throughput capacity in blocks of 100 reads/sec, 50 writes/sec, 5 global queries/sec Max JSON document size of 1MB	$1.00 USD/GB of data storage $0.25 USD/Read capacity $0.50 USD/Write capacity $5.00

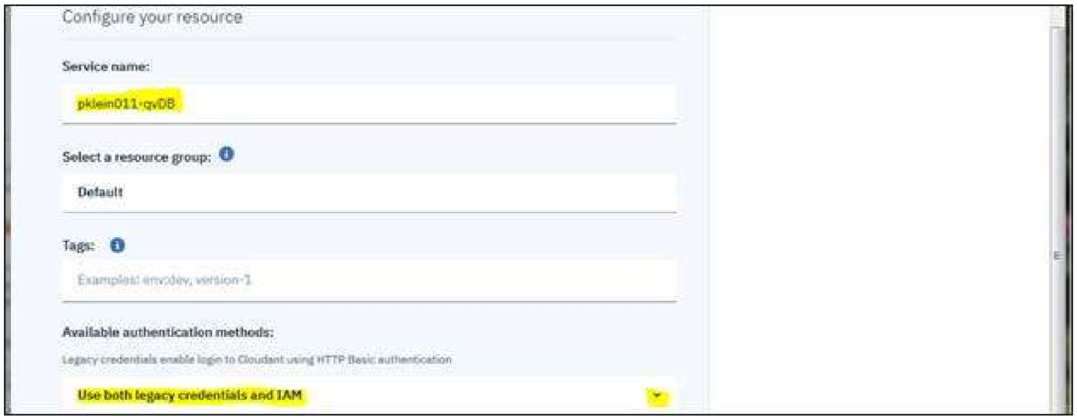
Summary

Cloudant *Free*
Region: Dallas
Plan: Lite
Service name: pklein011-qvDB
Resource group: Default

Create
Add to estimate
View terms
FEEDBACK

<u>Be sure to scroll down</u>. At the bottom of the Provisioning Page you will need to select your Service Name and authorization method. Select your Service Name by appending your ibmID in front like ibmID-qvDB. Select legacy and IAM as the

authorization method then click on **Create**. This will begin the provisioning process.



Configure your resource

Service name: pklein011-qvDB

Select a resource group: Default

Tags: Examples: env:dev, version-1

Available authentication methods:

Legacy credentials enable login to Cloudant using HTTP Basic authentication

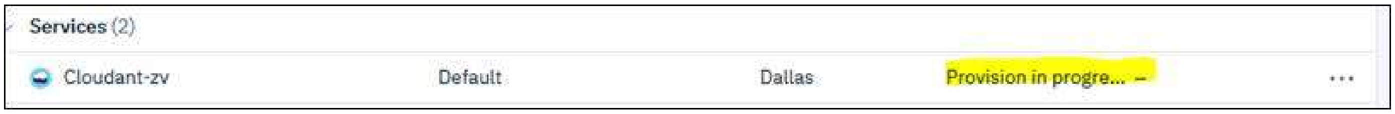
Use both legacy credentials and IAM

Note: It may take several minutes for this kind of Service to provision. Other Services such as Watson will provision immediately

---

Very Good! You have just finished the first step in getting an IBM Service working in conjunction with your sample application. You will have to wait for the provisioning to finish before you continue. The best way to wait is to view the Service status on the IBM Cloud Dashboard; then wait until the status changes from Provisioning to Provisioned. Don□t worry, it won□t take that long.

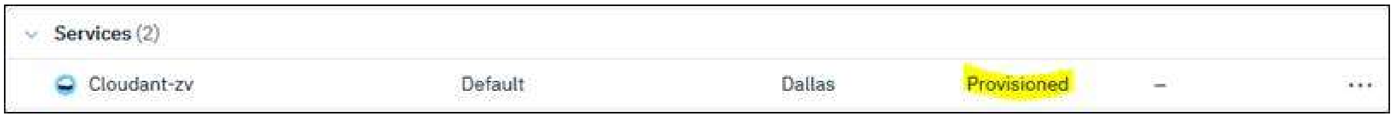
Display the IBM Cloud Dashboard and click on the Services row to expand.



Services (2)

Cloudant-zv	Default	Dallas	Provision in progre... –

Once the status changes to Provisioned as shown below in yellow:



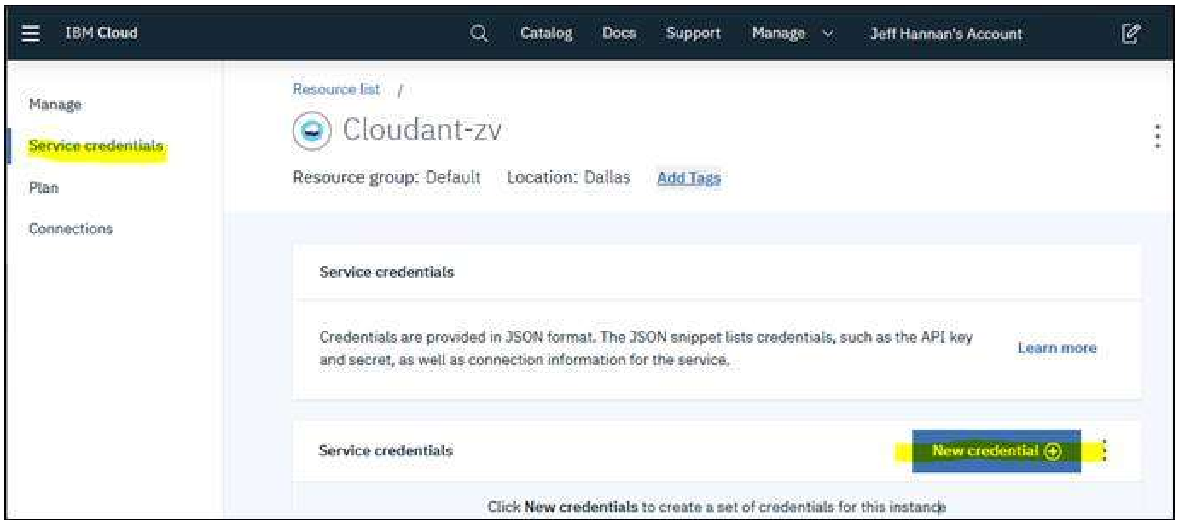
Services (2)

Cloudant-zv	Default	Dallas	Provisioned	–

you will begin the step of generating a set of credentials providing all the information about how to access the database including the important URL, userID and password required for your application to connect and manipulate the database objects.

Step 1 - Generating Credentials

Click on the Cloudant Service name again from the IBM Cloud Dashboard then click on the Service Credentials on the far left side menu. This will bring you to the credentialing facility for the Service you selected. Click on New Credentials to begin the process.



Resource group: Default Location: Dallas

Service credentials

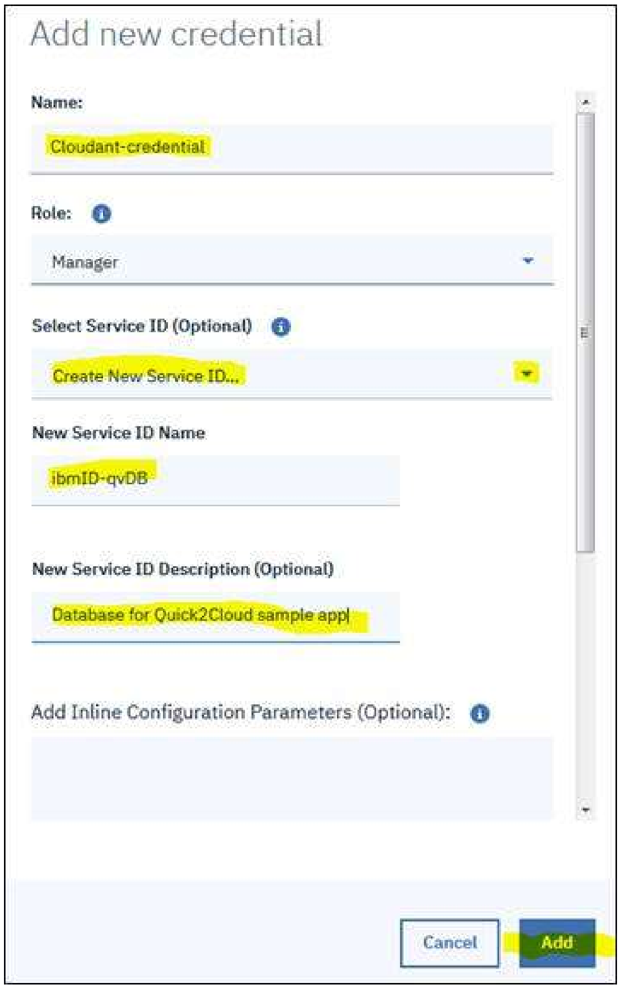
Credentials are provided in JSON format. The JSON snippet lists credentials, such as the API key and secret, as well as connection information for the service.

Service credentials

Click **New credentials** to create a set of credentials for this instance

Name your credential and select the Create New Service ID menu item as shown in yellow below. Fill in your ibmID as part of the New Service ID, that name will be used later in your application when it is programmatically parsing for the database information. When filled out completely, click on **Add** to generate the credential for use with your sample application.

Important: Make sure you scroll the screen down and fill in all the fields



Add new credential

Name: Cloudant-credential

Role: Manager

Select Service ID (Optional): Create New Service ID...

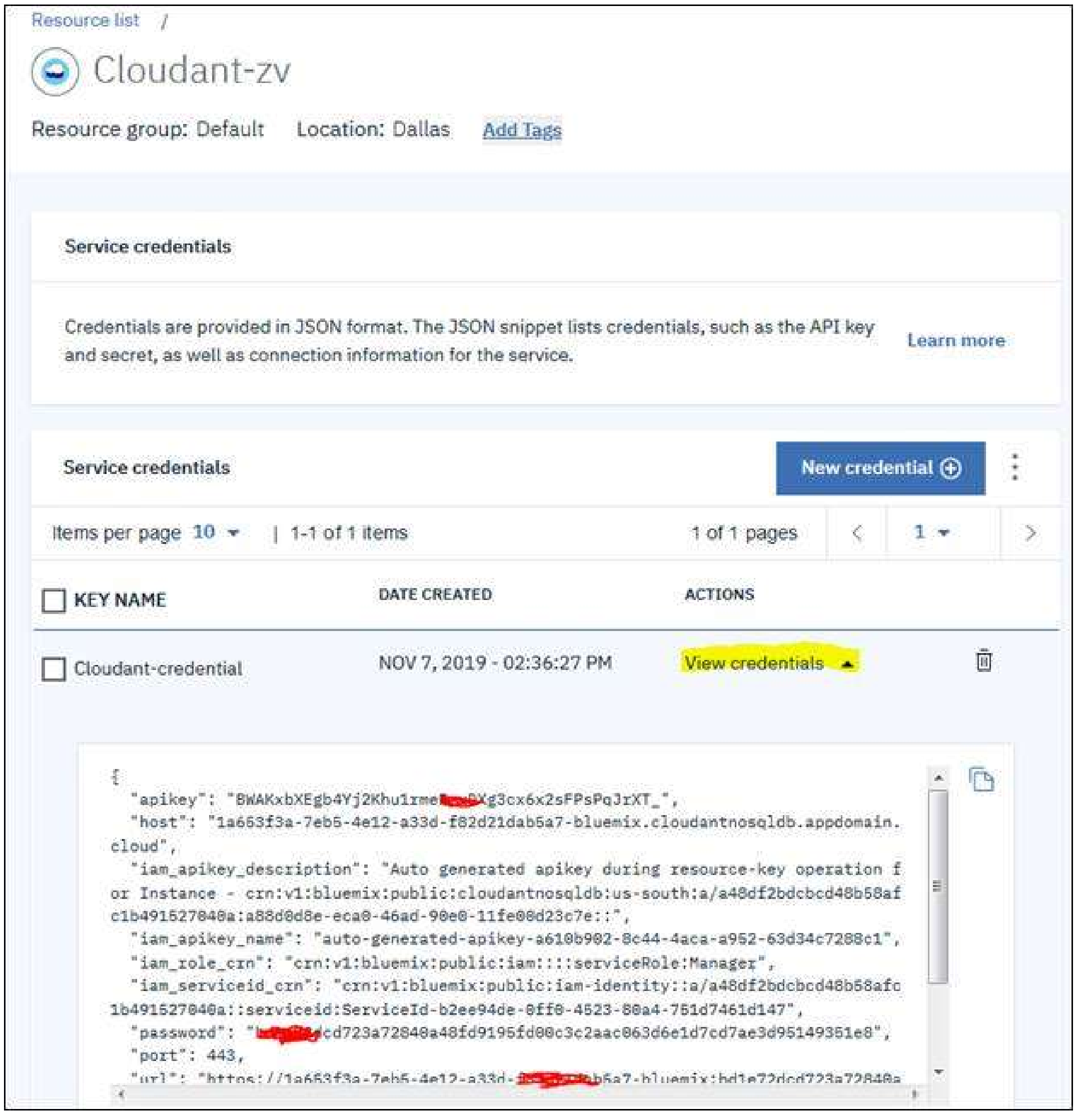
New Service ID Name: ibmID-qvDB

New Service ID Description (Optional): Database for Quick2Cloud sample appl

Add Inline Configuration Parameters (Optional):

You should now see the generated credential as shown below. By looking at the json representation of the specific keys you can get an idea on how an application may parse this and figure out everything it needs to make a database connection. A parsing technique will be introduced a bit later in the course.

Make sure you click on View Credentials when looking.


Resource list /

Cloudant-zv

Resource group: Default Location: Dallas Add Tags

Service credentials

Credentials are provided in JSON format. The JSON snippet lists credentials, such as the API key and secret, as well as connection information for the service. Learn more

Service credentials New credential

Items per page 10 | 1-1 of 1 items 1 of 1 pages 1

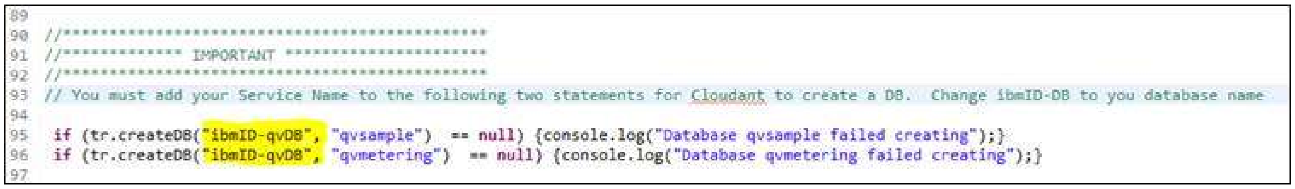
KEY NAME	DATE CREATED	ACTIONS
Cloudant-credential	NOV 7, 2019 - 02:36:27 PM	View credentials

```
{
  "apikey": "BWAKxbXEgb4Yj2Khu1rme[redacted]Xg3cx6x2sFPsPqJrXT_",
  "host": "1a663f3a-7eb5-4e12-a33d-f82d21dab5a7-bluemix.cloudantnosqldb.appdomain.
cloud",
  "iam_apikey_description": "Auto generated apikey during resource-key operation f
or Instance - crn:v1:bluemix:public:cloudantnosqldb:us-south:a/a48df2bdcbcd48b58af
c1b491527840a:a88d0d8e-eca0-46ad-90e0-11fe00d23c7e::",
  "iam_apikey_name": "auto-generated-apikey-a610b902-8c44-4aca-a952-63d34c7288c1",
  "iam_role_crn": "crn:v1:bluemix:public:iam::::serviceRole:Manager",
  "iam_serviceid_crn": "crn:v1:bluemix:public:iam-identity::a/a48df2bdcbcd48b58afc
1b491527840a::serviceid:ServiceId-b2ee94de-0ff0-4523-80a4-751d7461d147",
  "password": "[redacted]cd723a72840a48fd9195fd00c3c2aac063d6e1d7cd7ae3d95149351e8",
  "port": 443,
  "url": "https://1a663f3a-7eb5-4e12-a33d-[redacted]b5a7-bluemix:bd1e72dcd723a72840a
```

Step 2 - Edit Sample App

IMPORTANT

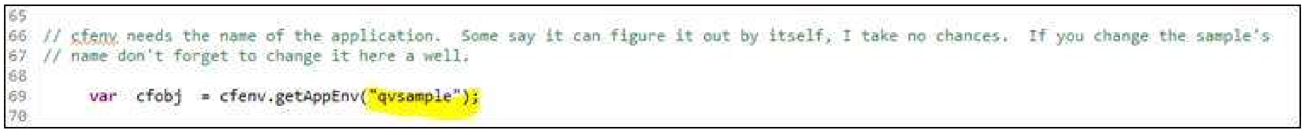
Now that you have a database Service defined you need to update your *Quick2Cloud* Sample Web Server Application with its name so it can be used. To accomplish this use your local editor and edit qvsample.js found in the q2c-web-app directory. Locate the two lines that create the databases qvsample.js uses.



```

//*************************************************
//************** IMPORTANT ***********************
//*************************************************
// You must add your Service Name to the following two statements for Cloudant to create a DB.  Change ibmID-DB to you database name

 if (tr.createDB("ibmID-qvDB", "qvsample")  == null) {console.log("Database qvsample failed creating");}
 if (tr.createDB("ibmID-qvDB", "qvmetering")  == null) {console.log("Database qvmetering failed creating");}

```

Replace the ibmID-qvDB with your database name that was just provisioned above. Save the file.

ALSO, for applications that are not using the default *Quick2Cloud* College Sample Web Application name (qvsample) you need to update the code with the new application name in the qvsamplib.js file. Again, the step below is not needed if you are just taking the college course but will be needed with your own application.

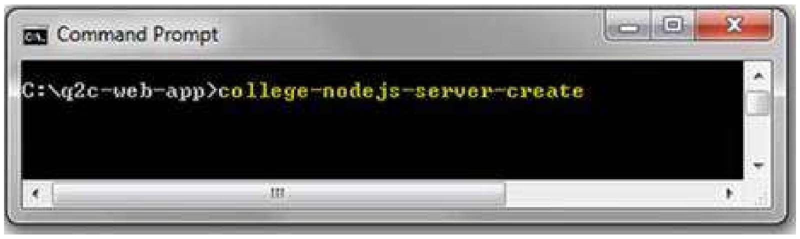


```
// cfenv needs the name of the application.  Some say it can figure it out by itself, I take no chances.  If you change the sample's
// name don't forget to change it here a well.

    var  cfobj  = cfenv.getAppEnv("qvsample");
```

Step 2A

Now that you made changes to your sample web application you need to update the IBM Cloud with these changes before you move on. Use the script created in the last course to push your updated code to the IBM Cloud, replacing the existing one running, with this new version.

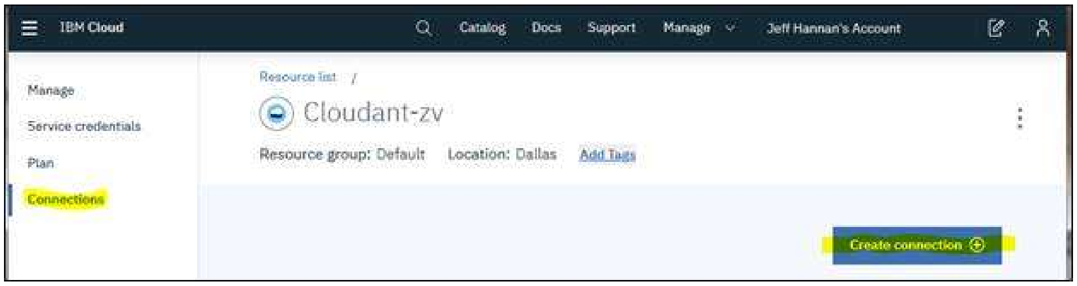
college-nodejs-server-create



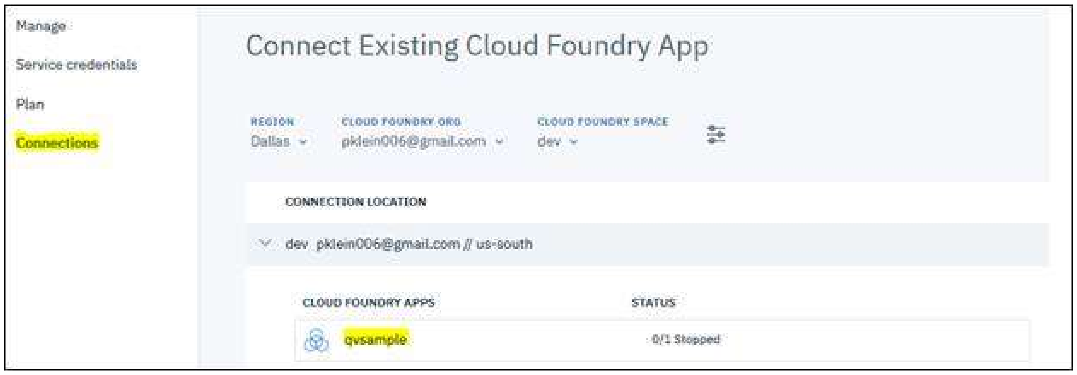
```
C:\q2c-web-app>college-nodejs-server-create
```

Step 3 - Bind Service to App

On the left display that was used for generating credentials click on the Connections link



This will bring you to the following display:



Click on the application name you want to connect this Service too. When this is complete, restart your application or use the script generated in **Course 100**. The application will now have a VCAP_SERVICES variable passed to it that contains all the Cloudant Service information for URL, userID and password allowing the sample application to connect and manipulate the database.

Continue