**Accessing your On Premises data from Microservices running on IBM Bluemix**

# Meta Information

This is information about the entire document.

**Define the audience**: The document is primarily targeted to application developers and IT architects that are migrating their existing applications to the Cloud, but they are not ready yet for security concerns or regulation issues to migrate their data to the cloud.

**Title:** Securely and Reliably expose your Data to Cloud Application in a Microservice based Architecture

**Define intended results:** Learn about possible options to expose data hosted in an On Premises SQL database though Microservices running on Bluemix.

Address questions are:

* How do I connect a Microservice running on Bluemix to a SQL DB located in my On Premises datacenter using a VPN Connection?
* How do I connect a Microservice running on Bluemix to a SQL DB located in my On Premises datacenter using the Secure Gateway?

**Authors:** Kyle Brown, Arcangelo Di Balsamo, Satya Sreenivas, Federico Spiti

**Keywords**: Bluemix, Hybrid, Microservice, Database, SQL, VPN, Secure Gateway, On Premises

# Document structure and content

## Overview

IBM Bluemix offers tremendous opportunities to create Internet scale applications with first class security. Both implementation and operational costs are much lower than in traditional on-premise IT hosting. However, migrating both your data and your application all at once can be risky – especially if you are changing the technology in which your data is hosted.

In many cases, it can be safer to keep data on premise and expose that data to Cloud applications, either directly or though a set of Microservices. This poses the problem about how to connect the Microservices hosted in the Cloud to the data hosted on premises.

This document shows two options to achieve that with IBM Bluemix Public. Other options exist for IBM Bluemix Dedicated, but are outside the scope of this document.

In this document we will use IBM Softlayer as a “stand-in” for your on-premise systems. Note that in many cases IBM Softlayer is itself a good solution for hosting data in the Cloud, especially when that data has to provided on legacy technologies (such as Oracle or SQL Server) that are not directly supported from services in IBM Bluemix.

## Tools/Prerequisites

In order to fully benefit from this document, you need at the least to meet the following prerequisites:

* Basic Bluemix Knowledge
* Basic Networking Knowledge
* An IBM Bluemix account. If you do not have an existing Bluemix account, start your [free trial](https://developer.ibm.com/sso/bmregistration?lang=en_US&ca=dw-_-bluemix-_-mw-1606-crujcov1-bluemix-trs-_-article).
* Development workstation with Bluemix CLI, cf and optionally bx, and IBM Containers plugin (ic) installed. To complete the CLI setup refer to <https://console.ng.bluemix.net/docs/cli/index.html#cli>.
* Familiarity with the Online Store Sample application: <https://developer.ibm.com/bluemix/2015/03/16/sample-application-using-microservices-bluemix/>
* An IBM Softlayer Account
* A physical or virtual server located on Softlayer to host a MySQL DB (as an alternative you can also use a Docker container)
* A physical or virtual server located on Softlayer to host Docker 1.7.0 or higher.
* A physical or virtual server located on Softlayer running Vyatta and setup to route to the MySQL DB server.

## Design considerations/Planning

In order to show all the possibilities offered by Bluemix for creating hybrid connections, we will use a sample application called Online Store that was originally built as cloud native application. You can learn more about this application at the following link : <https://developer.ibm.com/bluemix/2015/03/16/sample-application-using-microservices-bluemix/>

At that link you will find the instructions on how to deploy the Online Store as a cloud native application. We recommend that you do so in order to become familiar with it before you go ahead with the Hybrid deployment.

The application simulates an On-line store where buyers can purchase items available in a catalog. When an order is submitted, a row is stored in the Orders DB. The application is composed of a User Interface written in PHP and by two Microservices: Orders and Catalog, written in Java and Node respectively.

This figure (Figure 1: Online Store) shows the architecture of the application:

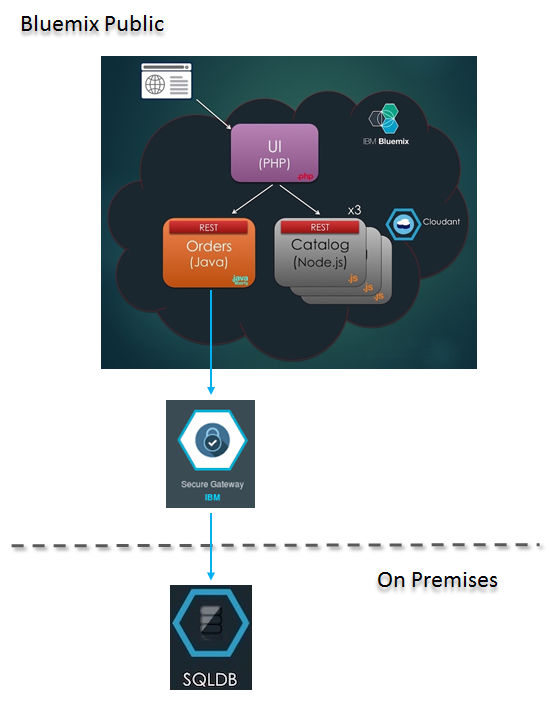


Figure : Online Store

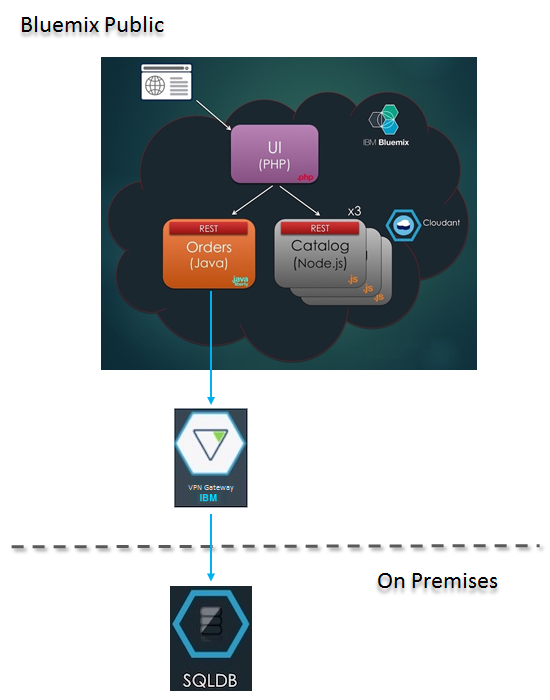
In our deployment we will focus on the Orders Microservice, which is implemented as a Java Liberty application connected to a SQL database. In our tests we will use MySQL as that SQL database.

We will show how to create a bridge between the MySQL DB running on-premise (in our case, using Softlayer to simulate that on-premise deployment) behind a firewall and an Orders Microservice running in Bluemix using two of the mechanisms offered by Bluemix: Secure Gateway and VPN.

In the case of the Secure Gateway the deployment diagram changes as follows:



While in the case of the VPN Gateway the deployment diagram changes to the following:



The Secure Gateway and VPN Services are both used to provide secure and reliable ways to cross a customer’s firewall. They have both pros and cons that we will discuss later in this document.

## Deploy your application/solution

This section contains step-by-step instructions that will guide you through the setup of the environment using both the Secure Gateway and VPN Services. The Secure Gateway is required to connect Bluemix Cloud Foundry Applications to on-premise services, whereas VPN is required to connect IBM Containers to on-premise services.

If you are not interested in performing a comparative analysis between the two compute options, or if you have already determined your preferred option, you may implement only one of the two possible choices.

### Setup MySQL Database Server in SoftLayer

MySQL database server can be setup in SoftLayer by placing an order to provision a Virtual Server Instance (VSI) pre-installed with MySQL server software.

1. Log into SoftLayer Portal. Place Order for an Hourly/Monthly VSI using following options. Default selections can be made for other options.

Operating System: **CentOS 7.x - Minimal Install (64 bit)**

Uplink Port Speeds: **Private Only**

Database Software: **MySQL for Linux**

1. After the CentOS VSI VSI is successfully provisioned, note down the private ip-address and password for user-id **root**.
2. Connect to SoftLayer VPN, and ssh to the VSI as **root**.
3. Connect to MySQL (MariaDB) server CLI by typing **mysql** at Linux command prompt.

Instructions here are limited to creating a database and a user to access the database, refer to MySQL documentation to setup a secure production grade MySQL database server.

1. Create orders database and orders\_dbuser to use the database.

MariaDB [(none)]> CREATE DATABASE orders;

MariaDB [(none)]> GRANT ALL PRIVILEGES ON orders.\* TO 'orders\_dbuser'@'%' IDENTIFIED BY 'Pass4OrdersUs3r';

MariaDB [(none)]> FLUSH PRIVILEGES;

MariaDB [(none)]> quit;

1. Note down the database server's private IP address, database name (orders), database user (orders\_dbuser) and password (Pass4OrdersUs3r). These will be used to create a Bluemix user provided service to connect to this database.

### Connect to Bluemix Org using CLI

Use values from table below for bluemix-api-endpoint and bluemix-domain.

|  |  |  |
| --- | --- | --- |
| **Region** | **bluemix-api-endpoint** | **bluemix-domain** |
| US South | api.ng.bluemix.net | mybluemix.net |
| London | api.eu-gb.bluemix.net | eu-gb.mybluemix.net |
| Sydney | api.au-syd.bluemix.net | au-syd.mybluemix.net |

1. Log in to your Bluemix account.

# cf login -a <bluemix-api-endpoint> -u <your-bluemix-user-id>

1. Set target to use your Bluemix Org and Space.

# cf target -o <your-bluemix-org> -s <your-bluemix-space>

## Access On-premise Data using Secure Gateway Connection

This section will deploy the Orders Microservice as a Cloud Foundry application in Bluemix, this application will access the Orders database in MySQL server in SoftLayer using Secure Gateway service.

### Setup Secure Gateway Connection

Implementing a Secure Gateway connection requires configuration on both the Server side (in Bluemix) and the Client side (On Premises).

The Secure Gateway service can be created using Dashboard in Bluemix web console, or Bluemix CLI. Also, there are multiple runtime options available to setup the secure gateway client. In this document we will demonstrate setting up the Secure Gateway client as a docker container running in SoftLayer.

#### Setup Secure Gateway service in Bluemix

1. Create Secure Gateway service instance in your Bluemix Space, this service will be used to establish connectivity between the Orders Microservice in Bluemix and Database server in SoftLayer.

# cf create-service SecureGateway securegatewayplan My-SecureGateway

1. Log into [Bluemix Dashboard](https://new-console.ng.bluemix.net/" \l "all-items). Click on your user avatar icon located at the top right corner of the dashboard and set the **Region**, **Organization**, and **Space** used for this deployment. From list of services double-click on **My-SecureGateway** service to launch the **Secure Gateway Dashboard**.
2. In the **Secure Gateway Dashboard**, click **Add Gateway**. The Add Gateway page is displayed. Enter **Connect SoftLayer Devices** in **Gateway Name** input field.

##### For this sample application clear the checkbox for **Require security token to connect clients**. If this option is left selected, it means that you will need to enter a security token each time you start the Secure Gateway client.

##### For this sample application clear the checkbox for **Token Expiration:.** If selected, this option lets you set the token expiration time. When the security token expires, it is not automatically regenerated. You need to regenerate the token to receive a new one.

##### Click **ADD GATEWAY** to add the gateway. The gateway named **Connect SoftLayer Devices** should now be displayed on the **Secure Gateway Dashboard**.

##### On the **Secure Gateway Dashboard** double-click on **Connect SoftLayer Devices** gateway. Notice this gateway is in **Disconnected** state. Click on icon to **Copy Gateway ID** and paste it to a text file. This ID will be used later by the Secure Gateway client to connect to this gateway service.

##### Click on **Add Destination** to create connection to the MySQL database server in SoftLayer. **Add Destination** wizard will launch, click on **Advanced Setup**. On the Next screen enter following values and then click **ADD DESTINATION**.

|  |  |
| --- | --- |
| Select Radio-button | On-Premises Destination |
| Destination Name | SoftLayer MySQL Server |
| Resource Hostname | Private IP Address of MySQL Database server |
| Resource Port | 3306 |
| Protocol | TCP |

1. **SoftLayer MySQL Server** is now added as a Destination with state **Enabled**. Click on the **Settings** (gear-wheel icon) to display the connection details. Note down **Cloud Host : Port** values and close the details dialog.
2. Create Bluemix service (**mysql-OrdersDBService**) to connect to MySQL database server running in SoftLayer. Use following values.

# cf create-user-provided-service mysql-OrdersDBService -p "hostname, port, user, password, jdbcUrl, uri"

|  |  |
| --- | --- |
| hostname | Cloud Host name from step 9 |
| port | Port number from step 9 |
| user | orders\_dbuser |
| password | Pass4OrdersUs3r |
| jdbcUrl | jdbc:mysql://<hostname>:<port>/orders |
| uri | mysql://<hostname>:<port>/orders |

#### Setup Secure Gateway Client in SoftLayer

The Secure Gateway client can be installed in Windows, RedHat, SuSE, Ubuntu, and OS X, and also be run on IBM DataPower, and within Docker. This document will use a Docker Container.

1. Log into SoftLayer Portal. Place Order for an Hourly/Monthly VSI running CoreOS stable. This will be used as Docker host to run Secure Gateway client container. Select CoreOS Stable for Operating System, default selections can be accepted for all the other options.
2. After the CoreOS VSI is successfully provisioned, note down the private ip-address and password for user-id core.
3. Connect to SoftLayer VPN, and ssh to CoreOS VSI as core. Run the Secure Gateway client container using following commands.

Pull the secure gateway client image from IBM

# sudo docker pull ibmcom/secure-gateway-client

Run the secure gateway docker container

# sudo docker run -itd ibmcom/secure-gateway-client <gateway-id> --service -A "acl allow :3306"

Make sure that you substitute <gateway-id> above with the Gateway ID value from Step 7 in the previous section.

1. Go to Bluemix Dashboard. From list of services double-click on **My-SecureGateway** service to launch the Secure Gateway Dashboard.
2. On the Secure Gateway Dashboard double-click on **Connect SoftLayer Devices** gateway, it should now be in **Connected** state.

### Deploy Orders Microservice to Bluemix

1. Clone repository to your local machine.

# cd <your-working-dir>

# git clone https://github.com/ibm-solution-engineering/hybrid.git

1. Deploy the Orders Microservice application.

# cd hybrid/onprem-connectivity/orders-microservice-app

# cf push -d <bluemix-domain> --no-start

1. Go to [Bluemix Dashboard](https://new-console.ng.bluemix.net/#all-items).. From the list of \_Cloud Foundry Applications\_ locate **Hybrid-Orders-Microservice**, it should be in **Stopped** state.
2. Bind MySQL database service to Orders Microservice application.

# cf bind-service mysql-OrdersDBService Hybrid-Orders-Microservice

1. Start Orders Microservice.

# cf start Hybrid-Orders-Microservice

1. Get the URL to access the application and confirm the application is now running.

# cf app Hybrid-Orders-Microservice | grep 'state\|url'

1. Copy the application URL and launch it in a web browser, if the web page loads successfully then the Orders Microservice deployed successfully.
2. Click on the GET uri to see any existing orders in the \_\_orders\_\_ database hosted in SoftLayer. To create a new order send a POST request using cURL command with JSON string containing the order information.

This completes the deployment of a Hybrid Microservice sample application running in Bluemix connected a MySQL database server running in SoftLayer.

## Access On-premise Data using VPN

This section will deploy the Orders Microservice as an IBM Container in Bluemix, this container will access the Orders database in MySQL server in SoftLayer using a secure VPN between IBM VPN Service in Bluemix and Vyatta Gateway appliance in SoftLayer.

**TBD: Arcangelo**

### Deploy Orders Microservice Container in Bluemix

#### Create Docker Image

#### Publish Docker Image to Bluemix

#### Start Docker Container

### Setup VPN Connection

Establish a secure peer to peer IPsec tunnel between IBM VPN Service in Bluemix and Vyatta Gateway in SoftLayer.

#### Setup IBM VPN Service in Bluemix

1. Create a VPN service instance in Bluemix

# cf create-service VPN\_Service\_Broker Standard My-VPNService

1. Go to Bluemix Dashboard. From list of services double-click on **My-VPNService** service to launch the IBM Virtual Private Network dashboard.
2. Click on **Create Gateway** to create the default gateway. Note down the **IP Address** of the gateway. In SoftLayer Vyatta configuration when creating the IPsec peer, replace *<BMX-VPN-GW-IP>* with this value.
3. Also note down the Subnets for **All Single Containers** and **All Scalable Groups.** In SoftLayer Vyatta configuration when creating the IPsec peer, replace *<BMX-IC-Subnet>* with this value.

#### Setup Vyatta Gateway in SoftLayer

1. Log into SoftLayer Portal. Place Order for Vyatta Gateway Appliance. Note the following hardware specifications are not recommended for a production grade setup, these are minimum specifications to run sample workloads.

|  |  |
| --- | --- |
| Server | Single Intel Xeon E3-1270 |
| RAM | 4 GB |
| Operating System | Vyatta 6.x Subscription Edition (64 bit) |
| Disk | 1TB JBOD |

1. Go to Device Details for your MySQL Database server. Disconnect the **Public** interface, and click on **VLAN** of the **Private** interface and note down the **VLAN Number**.
2. Note down the Subnet at bottom of the page, replace <Local-Subnet> with this value in Vyatta configuration. Click on the subnet note down the **Gateway** address, replace <vif-gateway> in Vyatta configuration with this value. Also note down the ***Mask Bits*** of the subnet, it is the numeric value after the forward slash (for example /26).
3. SSH into MySQL server and add route to Containers network in Bluemix via Vyatta Gateway.

# ip route add default via <vif-gateway>

1. After the Vyatta is provisioned connect to SoftLayer VPN, and ssh to the Vyatta using it’s private IP address as user **vyatta**.
2. Switch to configuration mode and run following commands to add a virtual interface to route to the VLAN containing MySQL server.

$ configure

# set interfaces bonding bond0 vif <VLAN-Number> address '169.254.178.90/29'

# set interfaces bonding bond0 vif <VLAN-Number> vrrp vrrp-group 2 priority '254'

# set interfaces bonding bond0 vif <VLAN-Number> vrrp vrrp-group 2 sync-group 'vgroup1'

# set interfaces bonding bond0 vif <VLAN-Number> vrrp vrrp-group 2 virtual-address '<vif-gateway>/<Mask Bits>'

1. In configuration mode run the following commands to create the IPsec peer.

# set vpn ipsec esp-group bmx-esp-default compression 'disable'

# set vpn ipsec esp-group bmx-esp-default lifetime '3600'

# set vpn ipsec esp-group bmx-esp-default mode 'tunnel'

# set vpn ipsec esp-group bmx-esp-default pfs 'dh-group2'

# set vpn ipsec esp-group bmx-esp-default proposal 1 encryption 'aes128'

# set vpn ipsec esp-group bmx-esp-default proposal 1 hash 'sha1'

# set vpn ipsec ike-group bmx-ike-default dead-peer-detection action 'restart'

# set vpn ipsec ike-group bmx-ike-default dead-peer-detection interval '20'

# set vpn ipsec ike-group bmx-ike-default dead-peer-detection timeout '120'

# set vpn ipsec ike-group bmx-ike-default lifetime '86400'

# set vpn ipsec ike-group bmx-ike-default proposal 1 dh-group '2'

# set vpn ipsec ike-group bmx-ike-default proposal 1 encryption 'aes128'

# set vpn ipsec ike-group bmx-ike-default proposal 1 hash 'sha1'

# set vpn ipsec ipsec-interfaces interface 'bond1'

# set vpn ipsec site-to-site peer *<BMX-VPN-GW-IP>* authentication mode 'pre-shared-secret'

# set vpn ipsec site-to-site peer *<BMX-VPN-GW-IP>* authentication pre-shared-secret 'sharedsecretstring'

# set vpn ipsec site-to-site peer *<BMX-VPN-GW-IP>* connection-type 'initiate'

# set vpn ipsec site-to-site peer *<BMX-VPN-GW-IP>* default-esp-group 'bmx-esp-default'

# set vpn ipsec site-to-site peer *<BMX-VPN-GW-IP>* ike-group 'bmx-ike-default'

# set vpn ipsec site-to-site peer *<BMX-VPN-GW-IP>* local-address '<Vyatta-Public-Address>'

# set vpn ipsec site-to-site peer *<BMX-VPN-GW-IP>* tunnel 1 allow-nat-networks 'disable'

# set vpn ipsec site-to-site peer *<BMX-VPN-GW-IP>* tunnel 1 allow-public-networks 'disable'

# set vpn ipsec site-to-site peer *<BMX-VPN-GW-IP>* tunnel 1 local prefix '<Local-Subnet>'

# set vpn ipsec site-to-site peer *<BMX-VPN-GW-IP>* tunnel 1 remote prefix '*<BMX-IC-Subnet>*'

.

1. Commit and Save the configuration

# commit

# save

1. Go to SoftLayer portal, browse to Network > Gateway Appliances. Click on the Vyatta Gateway configured for this setup to launch the Details page.
2. Under the **Associate a VLAN**, select the **VLAN Number** saved from step-2 and click on **Associate.** The VLAN will be added to Associated VLANs.
3. Under **Associated VLANs** select the VLAN that was just added. Click on **Actions** and select **Route VLAN**. Give it a few minutes for the configuration change to take effect.

#### Create Site Connection in IBM VPN Service in Bluemix

1. Go to Bluemix Dashboard. From list of services double-click on **My-VPNService** service to launch the IBM Virtual Private Network dashboard.
2. Click on **Create Connection** to create a new site to site connection with the Vyatta Gateway in SoftLayer. Use following values to create a new connection. Accept defaults for other input fields.

|  |  |
| --- | --- |
| Preshared Key String | sharedsecretstring |
| Customer Gateway IP | <Vyatta-Public-Address> |
| Customer Subnet | <Local-Subnet> in Vyatta Configuration |

1. Connection should be created with Status **ACTIVE.**

#### Verify VPN Connection

1. Ping from Order Microservice container to MySQL DB server in SoftLayer.

# cf ic login

# cf ic ps exec –it <container-id> ping –t3 <MySQL-Private-IP-Address>

1. Get IP Address of Orders Microservice container

# cf ic inspect 547c6d9e-3b8 | grep -m1 IPAddress

1. Log into MySQL server and ping to Orders Microservice container in Bluemix.

# ping –t3 <IP-Address-of-Container>

#### Update DataSource Information in Orders application

**TBD Arcangelo**

## Manage your application/solution

The table below gives you some indications about which solution we recommend considering different aspects

|  |  |  |  |
| --- | --- | --- | --- |
|  | Secure Gateway | VPN | Direct Link |
| Performance | Up to 50 transactions per second | Up to 1000 transactions per second | Above 1000 transaction per second |
| Reliability | Poor | Poor | High |
| Costs | Very Cheap | Cheap | Expensive |
| Time to Setup | Less than 1 h | 4 hours | months |
| Complexity | Low | Medium | High |
| Monitoring | Limited | Adequate | Optimal |
| … |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Resources

Ideally, you’ll include many of these resources inline with the text. But this is a great place to link to overview content related to the paper or products.

## Scenarios

If you want to walk the reader through specific scenarios, this is the place to put that content. Diagrams and images that demonstrate what you’re talking about are great.