Accessing your on-premises data from microservices running on IBM Bluemix

1 DOCUMENT DETAILS

1.1 AUTHORS
Authors of this document include:
1.2 **INTENDED AUDIENCE**
The document primarily targets application developers and IT architects who are migrating their existing applications to the cloud, but they are not yet ready for security concerns or regulation issues to migrate their data to the cloud.

1.3 **PURPOSE**
The purpose of this document is to introduce you to the possible options to expose data hosted in an on-premises SQL database through microservices running on Bluemix.

After reading the document, you should be able to answer the following questions:

- How do I connect a microservice running on Bluemix to a SQL database located in my on-premises datacenter using a VPN connection?
- How do I connect a microservice running on Bluemix to a SQL database located in my on-premises datacenter using the Secure Gateway?

2 **INTRODUCTION**

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-premises IT hosting. However, migrating both your data and your applications at the same time can be risky – especially if you are changing the technology in which your data is hosted.

In many cases, it is safer to keep data on premises and expose that data to cloud applications, either directly or through a set of microservices. But now you’re left wondering how to connect the microservices hosted in the cloud to the data hosted on premises. This document shows you two ways to achieve that with IBM Bluemix Public. Other options exist for IBM Bluemix Dedicated, but are outside the scope of this document.

In this tutorial, we use IBM Softlayer as a stand-in for your on-premise systems. Note that, in many cases, IBM Softlayer is a great solution for hosting data in the cloud, especially when that data is provided on legacy technologies (such as Oracle or SQL Server) that are not directly supported from services in IBM Bluemix.
3 PREREQUISITES

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.
- Development workstation with Bluemix CLI, cf and optionally bx, and IBM Containers plugin (ic) installed. Refer to the Bluemix documentation to complete the CLI setup.
- Familiarity with the Online Store Sample application in the article, “Microservices Store Sample on 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 configured to route to the MySQL DB server.

4 DESIGN CONSIDERATIONS AND PLANNING

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 a cloud native application. You can learn more about this application in the article, “Microservices Store Sample on Bluemix.”

The article gives you instructions for how to deploy the Online Store as a cloud native application. We recommend that you follow the steps to familiarize yourself with the Online Store before you go ahead with the hybrid deployment.

The application simulates an online store where buyers purchase items available in a catalog. When an order is submitted, a row is stored in the Orders database. The application is composed of a user interface (UI) written in PHP and by two microservices: Orders and Catalog, written in Java and Node respectively.

Figure 1 shows the architecture of the application:
In our deployment, we focus on the Orders microservice, which is implemented as a Java Liberty application connected to a SQL database. In our tests, we use MySQL as that SQL database.

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

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

![Secure Gateway Diagram]

Figure 2: Secure Gateway

Here the Secure Gateway service acts as a bridge to the Secure Gateway client, which is co-located in Softlayer with our MySQL Database. The Secure Gateway client runs in a Docker container behind the Softlayer firewall, which then connects to the MySQL database running in a VM in Softlayer.

In the case of the VPN Gateway, the deployment diagram changes to the following:
A Vyatta VPN Gateway creates the firewall that separates the microservices from the MySQL DB, adding an extra layer of security and protection of your data.

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.

## 5 DEPLOY YOUR APPLICATION OR SOLUTION

Follow the step-by-step instructions in this section to set up an environment that uses 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.

### 5.1 SET UP MYSQL DATABASE SERVER IN SOFTLAYER

You can set up a MySQL database server in SoftLayer by placing an order to provision a virtual server instance (VSI) pre-installed with your MySQL server software. The steps for provisioning this VSI are:

1. Log in to SoftLayer Portal. Place an order for an hourly/monthly VSI using the following options. All other options can be left at the default selections.
   - Operating System: **CentOS 7.x - Minimal Install (64 bit)**
   - Uplink Port Speeds: **Private Only**
   - Database Software: **MySQL for Linux**

2. After the CentOS VSI VSI is successfully provisioned, make a note of the private IP address and password for user ID **root**.
3. Connect to the SoftLayer VPN, and ssh to the VSI as root.

4. Connect to the MySQL (MariaDB) server CLI by typing `mysql` at a Linux command prompt.

   *Note: Instructions here are limited to creating a database and a user to access the database, refer to MySQL documentation to set up a secure production grade MySQL database server.*

5. Use the following CLI commands to create an 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;
   ```

6. Make a note of the database server’s private IP address, database name (orders), database user (orders_dbuser) and password (Pass4OrdersUs3r). You will use these to create a Bluemix user provided service to connect to this database.

5.2 CONNECT TO BLUEMIX ORG USING CLI

Since Bluemix Public is available in different regions, use the appropriate values from the table below for bluemix-api-endpoint and bluemix-domain.

<table>
<thead>
<tr>
<th>Region</th>
<th>bluemix-api-endpoint</th>
<th>bluemix-domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>US South</td>
<td>api.ng.bluemix.net</td>
<td>mybluemix.net</td>
</tr>
<tr>
<td>London</td>
<td>api.eu-gb.bluemix.net</td>
<td>eu-gb.mybluemix.net</td>
</tr>
<tr>
<td>Sydney</td>
<td>api.au-syd.bluemix.net</td>
<td>au-syd.mybluemix.net</td>
</tr>
</tbody>
</table>

1. Log in to your Bluemix account.
   
   ```
   # cf login -a <bluemix-api-endpoint> -u <your-bluemix-user-id>
   ```

2. Set the target to use your Bluemix Org and Space.
   
   ```
   # cf target -o <your-bluemix-org> -s <your-bluemix-space>
   ```

6 ACCESS ON-PREMISES DATA USING SECURE GATEWAY CONNECTION

This section shows you how to deploy the Orders microservice as a Cloud Foundry application in Bluemix. This application accesses the Orders database in the MySQL server in SoftLayer through the Secure Gateway service.

6.1 SET UP THE SECURE GATEWAY CONNECTION

Implementing a Secure Gateway connection requires configuration on both the server side (in Bluemix) and the client side (on premises).

You can create the Secure Gateway service using the dashboard in the Bluemix web console, or through the Bluemix CLI. Also, there are multiple runtime options available to set up the Secure Gateway client. In this document, we demonstrate setting up the Secure Gateway client as a Docker container running in SoftLayer.
6.2 **SET UP THE SECURE GATEWAY SERVICE IN BLUEMIX**

1. Create a Secure Gateway service instance in your Bluemix space. This service will be used to establish connectivity between the Orders microservice in Bluemix and the database server in SoftLayer.

   ```
   # cf create-service SecureGateway securegatewayplan My-SecureGateway
   ```

2. Log in to the [Bluemix dashboard](https://console.ng.bluemix.net). 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 the list of services, double-click on **My-SecureGateway** service to launch the Secure Gateway dashboard.

3. In the Secure Gateway Dashboard, click **Add Gateway**. The Add Gateway page is displayed. Enter **Connect SoftLayer Devices** in the **Gateway Name** input field.

4. 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.

5. 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.

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

7. On the Secure Gateway Dashboard double-click **Connect SoftLayer Devices** gateway. Notice this gateway is in a disconnected state. Click the **Copy Gateway ID** icon and paste it to a text file. The Secure Gateway client uses this ID later to connect to this gateway service.

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

<table>
<thead>
<tr>
<th>Select Radio-button</th>
<th>On-Premises Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination Name</td>
<td>SoftLayer MySQL Server</td>
</tr>
<tr>
<td>Resource Hostname</td>
<td>Private IP Address of MySQL Database server</td>
</tr>
<tr>
<td>Resource Port</td>
<td>3306</td>
</tr>
<tr>
<td>Protocol</td>
<td>TCP</td>
</tr>
</tbody>
</table>

9. **SoftLayer MySQL Server** is now added as a Destination with state **Enabled**. Click on the **Settings** (gear-wheel icon) to display the connection details. Make a note of the **Cloud Host : Port** values and close the details dialog.

10. Create a Bluemix service (**mysql-OrdersDBService**) to connect to the MySQL database server running in SoftLayer. Use the following values from the table below.
6.3 Set up the Secure Gateway client in SoftLayer

You can install the Secure Gateway client in Windows, Red Hat Enterprise Linux, SuSE, Ubuntu, and OS X, and also be run on IBM DataPower, and within Docker. This document uses a Docker container:

1. Log into the SoftLayer Portal. Place an order for an hourly or monthly VSI running CoreOS Stable. This will be used as Docker host to run the Secure Gateway client container. Select CoreOS Stable for the Operating System. You can accept the default selections for all the other options.

2. After the CoreOS VSI is successfully provisioned, make a note of the private ip-address and password for user-id core.

3. Connect to the SoftLayer VPN, and ssh to the CoreOS VSI as core. Run the Secure Gateway client container using the 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.

4. Go to the Bluemix dashboard. From the list of services, double-click My-SecureGateway service to launch the Secure Gateway dashboard.

5. On the Secure Gateway dashboard, double-click Connect SoftLayer Devices gateway. It should now be in a connected state.

---

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

<table>
<thead>
<tr>
<th>hostname</th>
<th>Cloud Host name from step 9</th>
</tr>
</thead>
<tbody>
<tr>
<td>port</td>
<td>Port number from step 9</td>
</tr>
<tr>
<td>user</td>
<td>orders_db user</td>
</tr>
<tr>
<td>password</td>
<td>Pass4OrdersUs3r</td>
</tr>
<tr>
<td>jdbcUrl</td>
<td>jdbc:mysql://&lt;hostname&gt;:&lt;port&gt;/orders</td>
</tr>
<tr>
<td>uri</td>
<td>mysql://&lt;hostname&gt;:&lt;port&gt;/orders</td>
</tr>
</tbody>
</table>
6.4 DEPLOY ORDERS MICROSERVICE TO BLUEMIX CONNECTING OVER SECURE GATEWAY

Follow these steps to deploy the Orders microservice to Bluemix connecting over Secure Gateway.

1. Clone the git repository to your local machine.
   
   ```
   # cd <your-working-dir>
   # git clone https://github.com/ibm-solution-engineering/hybrid.git
   ```

2. Deploy the Orders microservice application as a pre-packaged Liberty server, giving it a desired name `<app-name>`:
   
   ```
   # cd hybrid/onprem-connectivity/orders-microservice-container
   # zip -r hybrid-ordersapi-ng.zip wlp/
   # cf push -d <bluemix-domain> <app-name> -p hybrid-ordersapi-ng.zip --no-start
   ```

3. Go to the Bluemix dashboard. From the list of _Cloud Foundry Applications_ locate the app with `<app-name>`. It should be in a stopped state.

4. Bind your MySQL database service to the Orders microservice application.
   
   ```
   # cf bind-service <app-name> mysql-OrdersDBService
   ```

5. Start the Orders microservice.
   
   ```
   # cf start <app-name>
   ```

6. Get the URL to access the application and confirm the application is now running.
   
   ```
   # cf app <app-name> | grep 'state\|url'
   ```

7. Copy the application URL and launch it in a web browser. If the web page loads successfully, then the Orders microservice deployed successfully.

8. Click on the GET URI to see any existing orders in the _orders_ database hosted in SoftLayer.

9. To create a new order, send a POST request using a cURL command with a JSON string containing the order information. Make sure to replace `<URI>` with the URL retrieved from the above command.

   ```
   # echo '{"itemid":"3242", "customerid":"4656", "count":1}' | curl -i -X POST -H "Content-Type: application/json" -d@- http://<URI>/rest/orders
   ```

   This will return HTTP 201 and an order ID in the response body. Make sure the order was created correctly using the following:

   ```
   $ curl http://<URI>/rest/orders/<order id>
   ```

This completes the deployment of a hybrid microservices sample application running in Bluemix connected a MySQL database server running in SoftLayer.
7 ACCESS ON-PREMISES DATA USING VPN

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

7.1 DEPLOY ORDER MICROSERVICE CONTAINER IN BLUEMIX

First, create a container hosting the Orders microservice starting from an IBM Liberty image and start it on Bluemix.

7.1.1 Create Docker image

1. Clone the git repository to your local machine.
   
   # cd <your-working-dir>
   # git clone https://github.com/ibm-solution-engineering/hybrid.git

2. Log in to the Bluemix container service.
   
   # cf ic login

3. Find the private namespace of your Bluemix registry.
   
   # cf ic namespace get

4. Build the docker image locally, but tag it with the private namespace, and give it the desired image name <image name>.

Pass the credentials for the MySQL user on the commandline, which will update server.xml.

   # cd hybrid/onprem-connectivity/orders-microservice-container
   # docker build --build-arg db_ip=<private server IP> --build-arg db_user=<mysql db user> --build-arg db_passwd=<mysql db password> --build-arg db_name=<db name> --build-arg db_port=<db port> -t registry.ng.bluemix.net/<namespace>/<image_name>.

By default, most of the arguments will use what is defined in the steps above. For example, if the MySQL server was configured as it is documented in the above steps, only the IP build argument needs to be passed (in this case the MySQL server is at 10.121.235.100):

   # docker build --build-arg db_ip=10.121.235.100 -t registry.ng.bluemix.net/<namespace>/<image_name>.

The default values are specified at the top of the Dockerfile.

Once the image has been built, verify that the image is in the local Docker repository:

   # docker images | grep <image_name>
7.1.2 Publish Docker image to Bluemix
1. Push the container into Bluemix using the following command:

```bash
# docker push registry.ng.bluemix.net/<namespace>/<image_name>
```

2. Verify that the container made it into Bluemix by using the following command:

```bash
# cf ic images | grep <image_name>
```

7.1.3 Start Docker container in Bluemix
Use the following command to discover what public IPs are available in the Bluemix space:

```bash
# cf ic ip list
```

If no public IPs exist, use the following command to request one. This command may require your account administrator to increase the public IP quota for the space.

```bash
# cf ic ip request
```

Run the container in Bluemix using the following command, passing the desired container name `<container name>`, and binding the desired public IP address to the container. This command exposes the ports defined in the liberty container (9080, 9443) with memory size 1024M, and executes the container in the background (-d):

```bash
# cf ic run -p <Public IP>:9080:9080 -p <Public IP>:9443:9443 -m 1024 --name <container name> -d registry.ng.bluemix.net/<namespace>/<image_name>
```

The web application should be accessible at the URL http://<Public IP>:9080 or https://<Public IP>:9443.

7.2 Set up VPN connection
In this section, you will establish a secure peer-to-peer IPsec tunnel between the IBM VPN Service in Bluemix and the Vyatta Gateway in SoftLayer.

7.2.1 Set up IBM VPN service in Bluemix
1. Create a VPN service instance in Bluemix

```bash
# cf create-service VPN_Service_Broker Standard My-VPNService
```

2. Go to the Bluemix dashboard. From list of services, double-click My-VPNService service to launch the IBM Virtual Private Network dashboard.

3. Click Create Gateway to create the default gateway. Make a note of the IP address of the gateway. In the SoftLayer Vyatta configuration when you create the IPsec peer, replace <BMX-VPN-GW-IP> with this value.

4. Also make a note of the subnets for all single containers and all scalable groups. In the SoftLayer Vyatta configuration when you create the IPsec peer, replace <BMX-IC-Subnet> with this value.
7.2.2 **Set up Vyatta Gateway in SoftLayer**

1. Log in to the SoftLayer portal. Place an order for a Vyatta Gateway Appliance.

Note the following hardware specifications are not recommended for a production-grade setup, these are minimum specifications to run sample workloads.

<table>
<thead>
<tr>
<th>Server</th>
<th>Single Intel Xeon E3-1270</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAM</td>
<td>4 GB</td>
</tr>
<tr>
<td>Operating System</td>
<td>Vyatta 6.x Subscription Edition (64 bit)</td>
</tr>
<tr>
<td>Disk</td>
<td>1TB JBOD</td>
</tr>
</tbody>
</table>

2. Go to the Device Details for your MySQL Database server. Disconnect the Public interface, then click on **VLAN** of the **Private** interface and make a note of the VLAN number.

3. Make a note of the Subnet at the bottom of the page, and replace `<Local-Subnet>` with this value in Vyatta configuration. Click on the subnet, and then make a note of the Gateway address that is shown. Next, replace `<vif-gateway>` in the Vyatta configuration with this value. Also make a note of the Mask Bits of the subnet—it’s the numeric value after the forward slash (for example /26).

4. SSH into MySQL server and add route to the Containers network in Bluemix via the Vyatta Gateway.

   ```
   # ip route add default via <vif-gateway>
   ```

5. After the Vyatta is provisioned, connect to SoftLayer VPN and ssh to the Vyatta using it’s private IP address as user **vyatta**.

6. Switch to the configuration mode and run the 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>'
   ```

7. 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>'

8. Commit and Save the configuration.
   # commit
   # save

9. Go to the SoftLayer portal, browse to Network > Gateway Appliances. Click on the Vyatta Gateway configured for this setup to launch the Details page.

10. Under Associate a VLAN, select the VLAN Number saved from step 2 and click Associate. The VLAN will be added to Associated VLANs.

11. 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.

7.2.2.1 Create site connection in IBM VPN Service in Bluemix

1. Go to the Bluemix dashboard. From list of services, double-click My-VPNService service to launch the IBM Virtual Private Network dashboard.

2. Click Create Connection to create a new site-to-site connection with the Vyatta Gateway in SoftLayer. Use the following values to create a new connection. Accept defaults for other input fields.

<table>
<thead>
<tr>
<th>Preshared Key String</th>
<th>Sharedsecretstring</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer Gateway IP</td>
<td>&lt;Vyatta-Public-Address&gt;</td>
</tr>
<tr>
<td>Customer Subnet</td>
<td>&lt;Local-Subnet&gt; in Vyatta Configuration</td>
</tr>
</tbody>
</table>

3. The connection should be created with the status ACTIVE.
7.2.3 Verify the VPN connection

1. Ping from the Order microservice container to the MySQL DB server in SoftLayer.
   # cf ic login
   # cf ic ps exec –it <container-id> ping –t3 <MySQL-Private-IP-Address>

2. Get the IP address of the Orders microservice container.
   # cf ic inspect 547c6d9e-3b8 | grep -m IPAddress

3. Log in to the MySQL server and ping to the Orders microservice container in Bluemix.
   # ping –t3 <IP-Address-of-Container>

8 MANAGE YOUR APPLICATION AND SOLUTION

The following table indicates which solution we recommend considering specific aspects of the different options.

<table>
<thead>
<tr>
<th></th>
<th>Secure Gateway</th>
<th>VPN</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Performance</strong></td>
<td>Up to 50 transactions per second</td>
<td>Up to 1000 transactions per second</td>
</tr>
<tr>
<td><strong>Reliability</strong></td>
<td>Poor</td>
<td>Poor</td>
</tr>
<tr>
<td><strong>Costs</strong></td>
<td>Very Cheap</td>
<td>Cheap</td>
</tr>
<tr>
<td><strong>Time to Setup</strong></td>
<td>&lt; 1 h</td>
<td>&lt; 4 hours</td>
</tr>
<tr>
<td><strong>Complexity</strong></td>
<td>Low</td>
<td>Medium</td>
</tr>
<tr>
<td><strong>Monitoring</strong></td>
<td>Limited</td>
<td>Adequate</td>
</tr>
</tbody>
</table>