PREPARING THE SYSTEM

Contents

Introduction

Certain things are needed prepared before installing Qlik Insight Bot. The purpose of this document is to guide you to keep the required items ready before proceeding the Bot installation.

Exporting Qlik Sense Client Certificate

The Qlik Sense "Client" certificate will be used by Qlik Insight Bot Services to authenticate the Qlik Sense user for any communication.

Follow below steps to export the same –

1. Open Qlik Management Console (QMC), go to **Certificate** under **Configuration System** on left pane.

2. Click **Add machine name** and add your machine name. The machine name will be the **Host name**.

   For Multi-Node env, machine name should be the **Host name** of the respective node. You will require the certificate for all nodes.

   Note: Do not set any password.
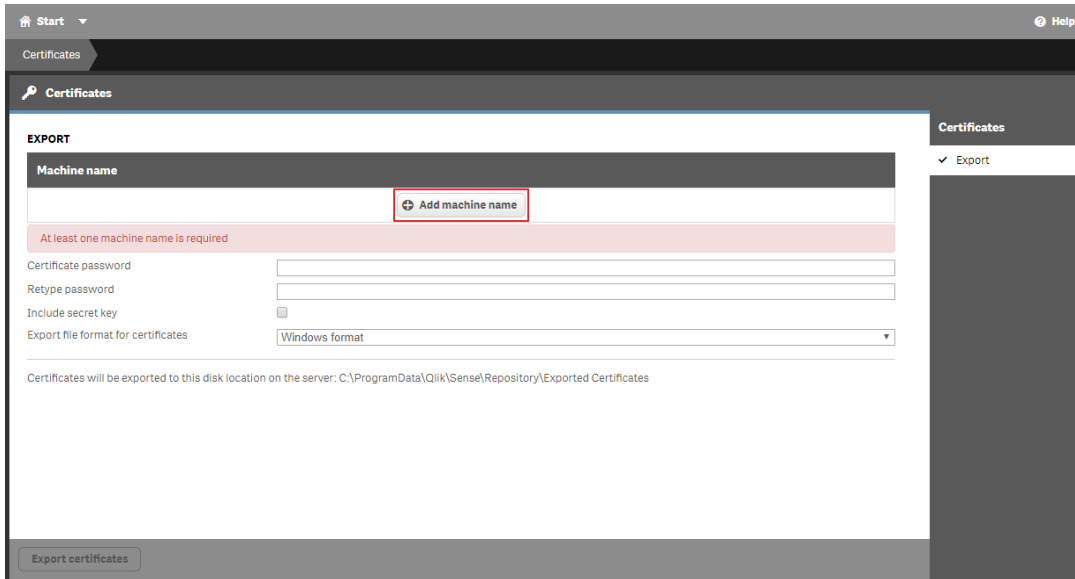
3. Check "Include secret key".

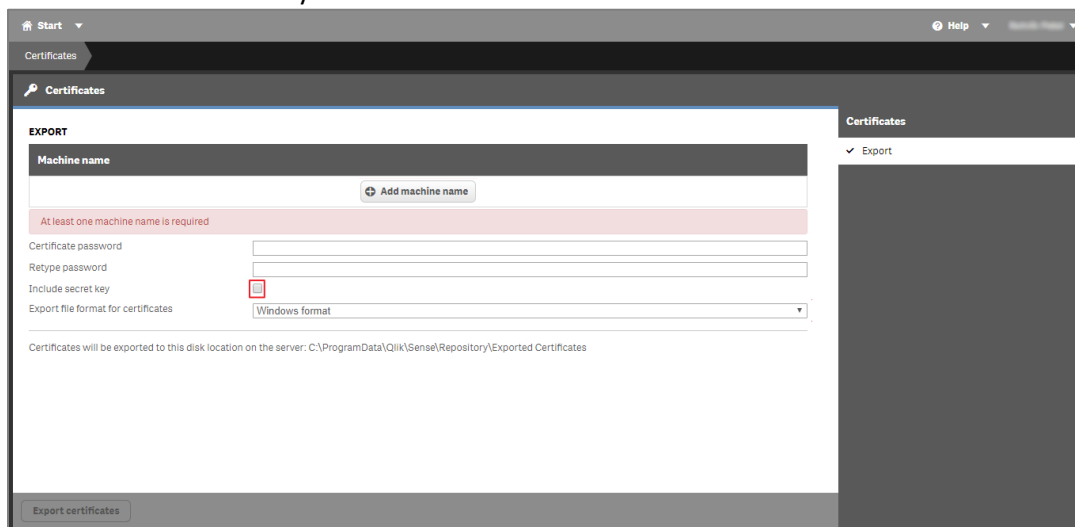4. Select **Windows Format** as "Export file format for certificates", click **Export Certificates**.
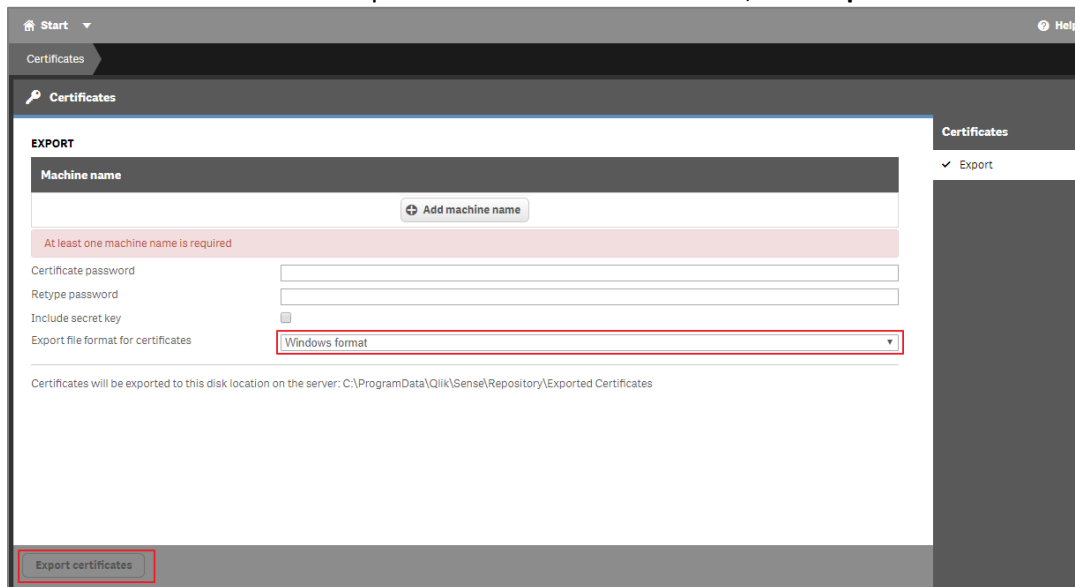
5. Certificates will be exported on the machine where Qlik Sense Server is installed. The physical path where certificates will be stored is as below -

   C:\ProgramData\Qlik\Sense\Repository\Exported Certificates\<Machine Name>

Configuring IIS

To host the Qlik Insight Bot Web Services on IIS, it is required to enable certain features of IIS. Follow the steps below for the same –

For Windows 8 or 10

1. Open **Control Panel**.
2. Select **Programs**.
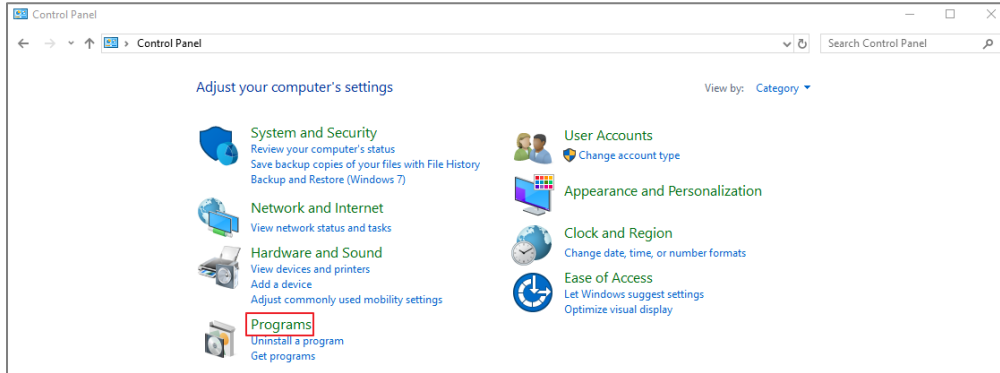
Or **Programs and Features**.

3. Select **Turn Windows features on or off**.
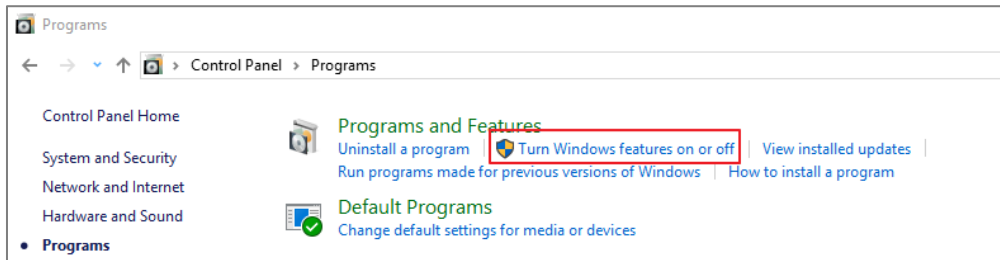
-OR-
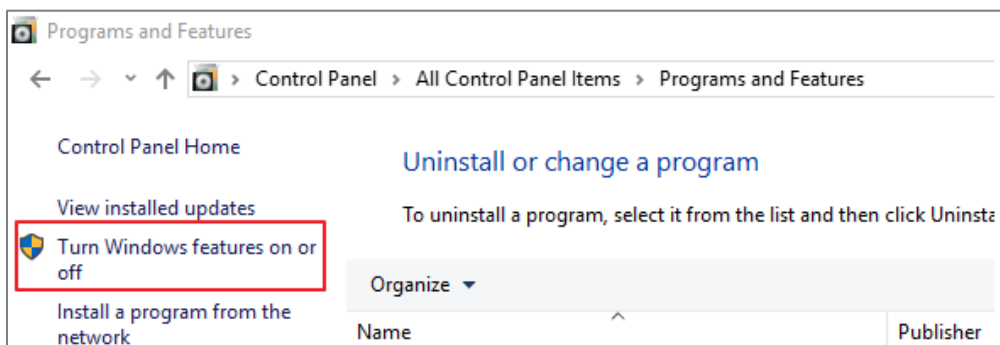
4. Select below listed features.
   - In Internet Information Services
     - o In Web Management Tools
       - ▪ IIS Management Console
     - o In World Wide Web Services
       - ▪ In Application Development Features
         - a) .NET Extensibility 4.5
         - b) ASP
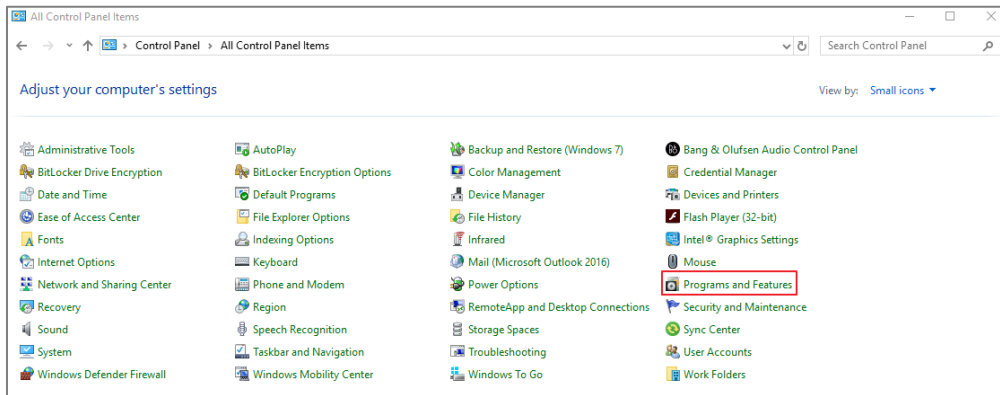   - c) ASP.NET 4.5
         - d) CGI
         - e) ISAPI Extensions
         - f) ISAPI Filters
         - g) WebSocket Protocol
       - ▪ In Common HTTP Features
         - a) Default Document
         - b) Direct Browsing
         - c) HTTP Errors
         - d) Static Content
       - ▪ In Health and Diagnostics
         - a) HTTP Logging
         - b) Request Monitor
         - c) Tracing
       - ▪ In Performance Features
         - a) Static Content Compression
       - ▪ In Security
         - a) Request Filtering
         - b) Windows Authentication
5. Click **OK**.

For Windows Server 2012 or 2016

1. Open **Server Manager**.
2. Click **Add Roles and Features** in Manage menu.
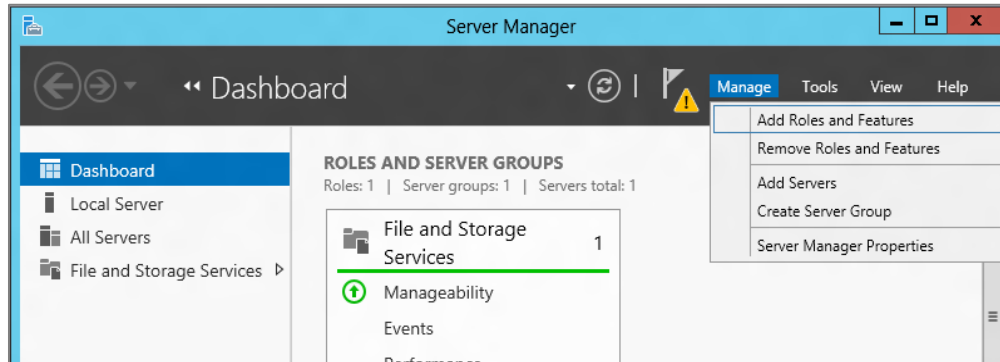
3. Click **Next**.
4. Select **Role-Based or Feature-Based Installation** and click **Next**.
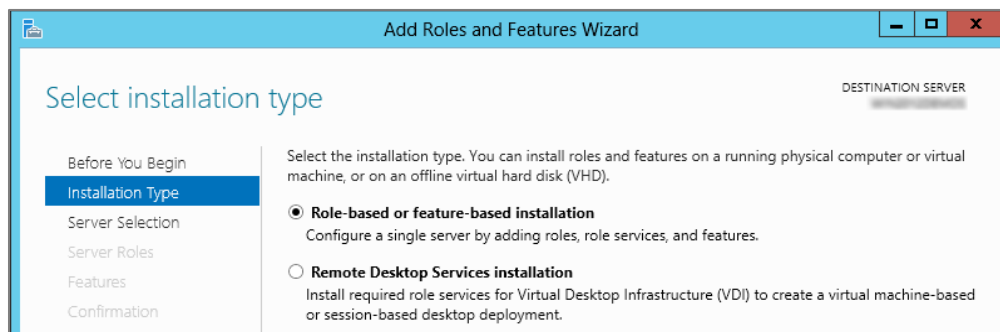
5. Select the appropriate server (local is selected by default) as shown and click **Next**.
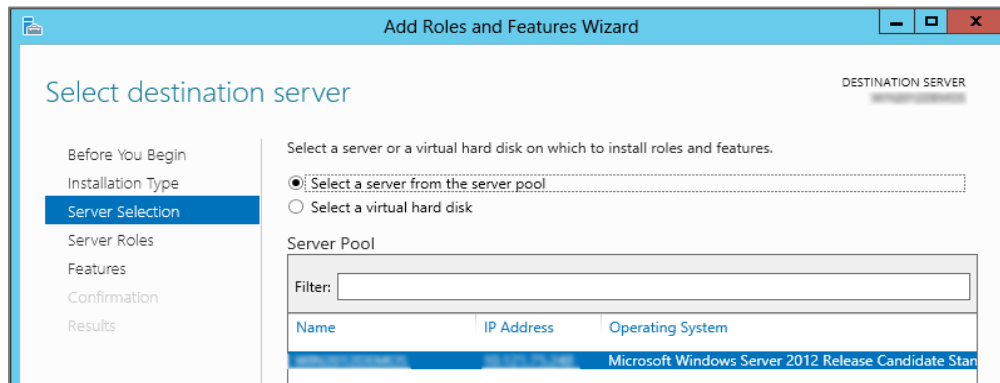
6. Select **Web Server (IIS)** and click **Next**.
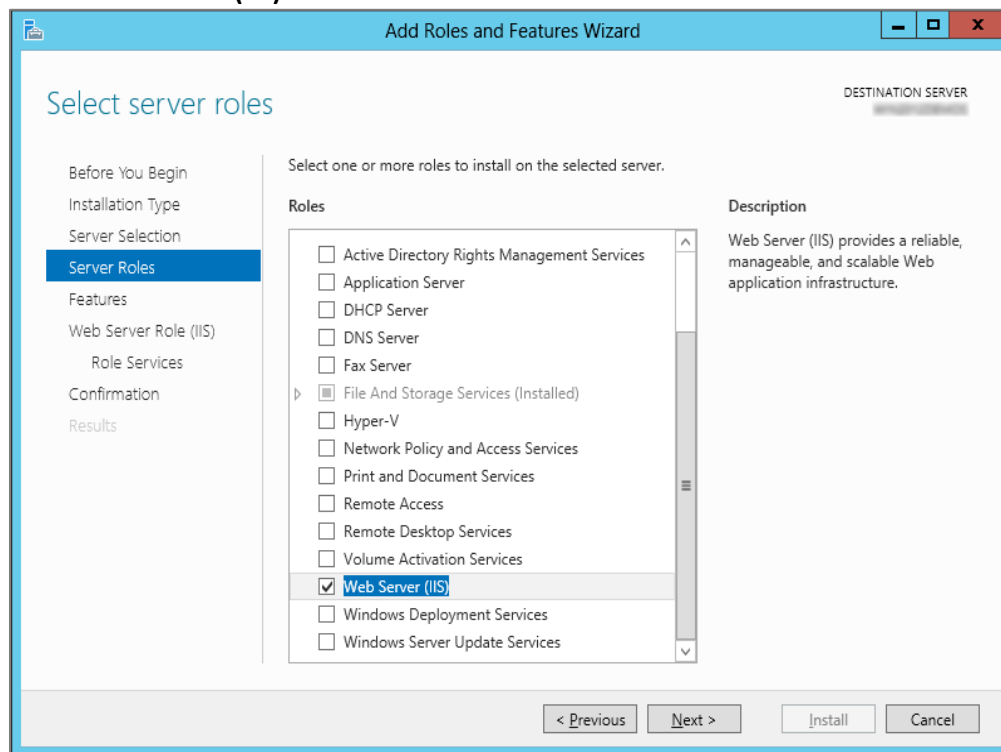
7. Ignore the **Features** tab and go on.
8. Click **next**.
9. Select below listed features and click **next**.
   - In Web Server
     - In Common HTTPS Features
       - a) Default Document
       - b) Direct Browsing
       - c) HTTP Errors
       - d) Static Content
     - In Health and Diagnostics
       - a) HTTP Logging
       - b) Request Monitor
       - c) Tracing
     - In Performance Features
       - a) Static Content Compression
     - In Security
       - a) Request Filtering
       - b) Windows Authentication
     - In Application Development Features
       - a) .Net Extensibility 4.5
       - b) ASP
       - c) ASP.NET 4.5
       - d) CGI
       - e) ISAPI Extensions
       - f) ISAPI Filters
       - g) WebSocket Protocol
   - In Management Tools
     - IIS Management Console

10. Click **Install.**
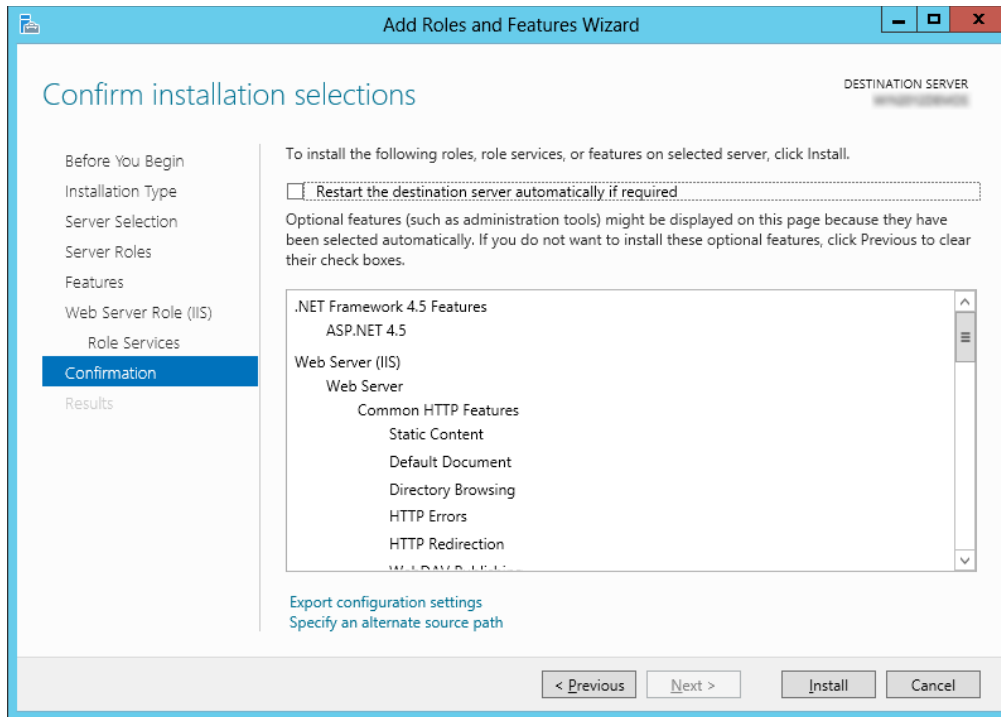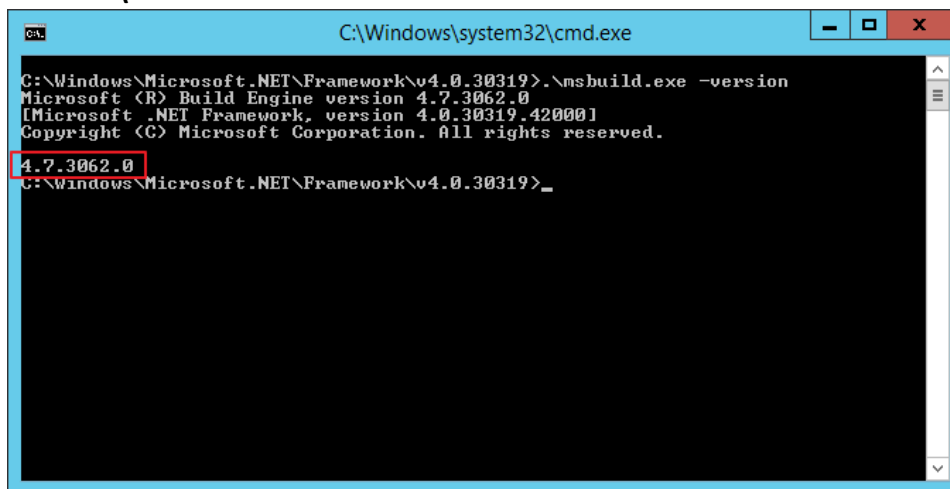
11. Once installation completed, click **Close**.
    For detailed guide visit [this](#).

.NET Framework 4.7.1

Check which .NET Framework is installed on your machine.

1. Open **CMD**.
2. Go to **C:\Windows\Microsoft.NET\Framework\v4.0.30319** directory.
3. Execute **.\MSBuild.exe -version** command.

If you don't have .NET Framework 4.7.1 installed on your machine, download the same from here and follow the installation steps.

Opening the Required Ports in Firewall

Ports 443, 4431, 4434, 4435, 5000 are required to open in firewall to access Qlik Insight Bot web services from outside the Network.

If Qlik Insight Bot is installed on a different machine from Qlik Sense Server, ports 443, 4242, 4747 and 4243 are required to open in firewall where Qlik Sense Server is installed.

Follow the steps below –

1. Open Windows firewall. Click **Advanced settings** from left pane.
2. Click **Inbound Rules**.
3. Click **New Rule…** under **Actions** in right pane.
4. Select **Port** option, click **Next**.
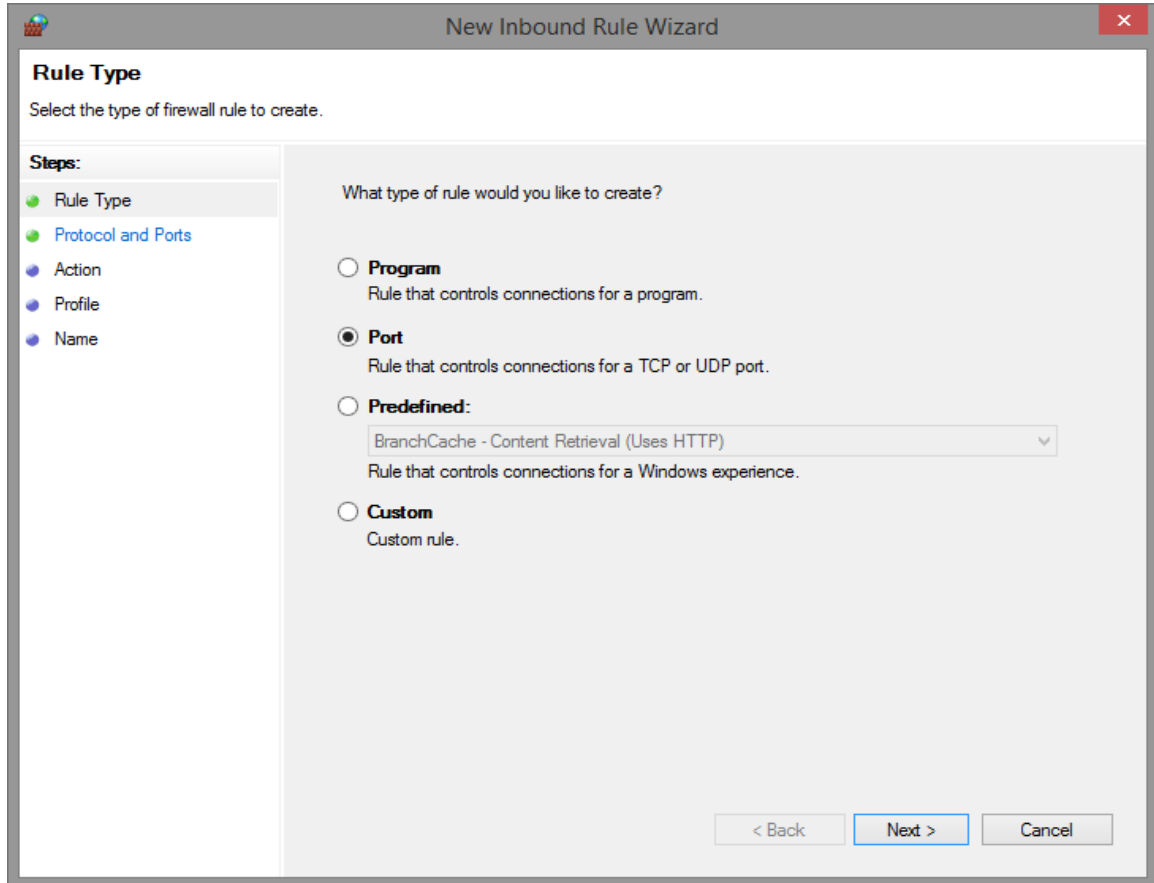
5. In **Protocol and Ports** section, select **TCP**, **Select Specific local ports** and enter **443, 4431, 4434, 4435, 5000**, click **Next**.
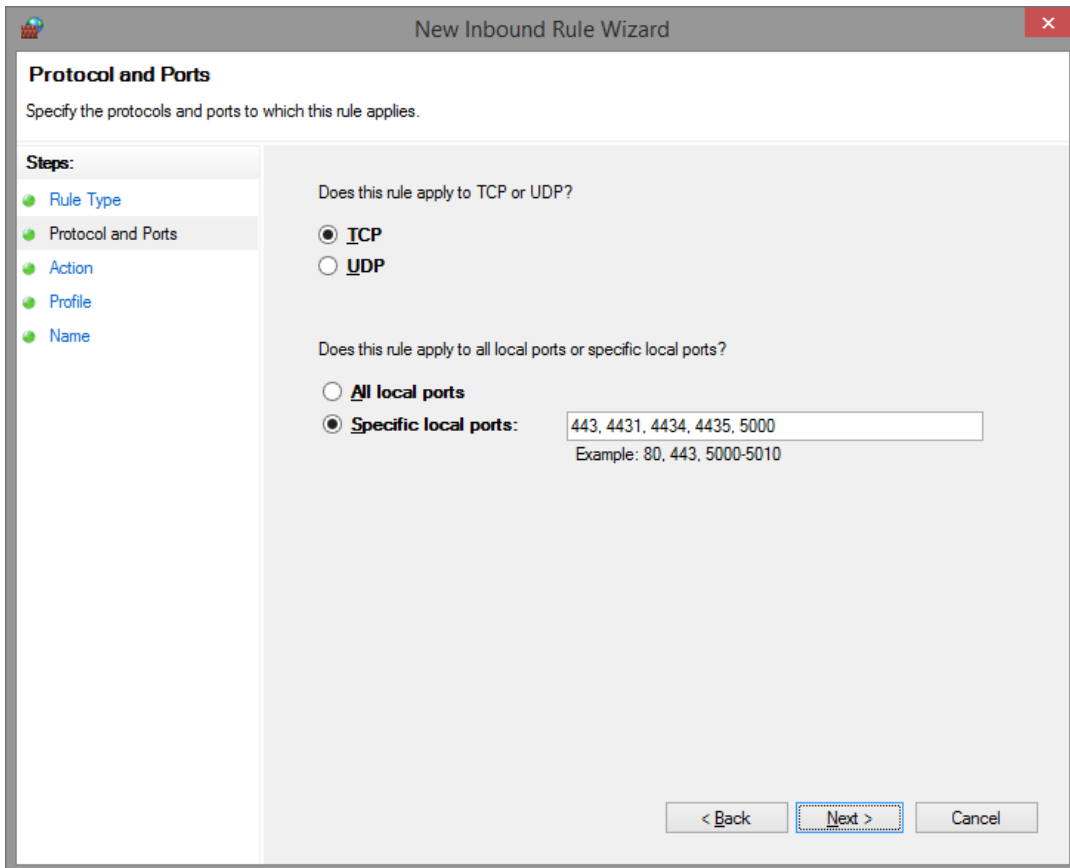
6. In **Action** section, select **Allow the connection**, click **Next**.
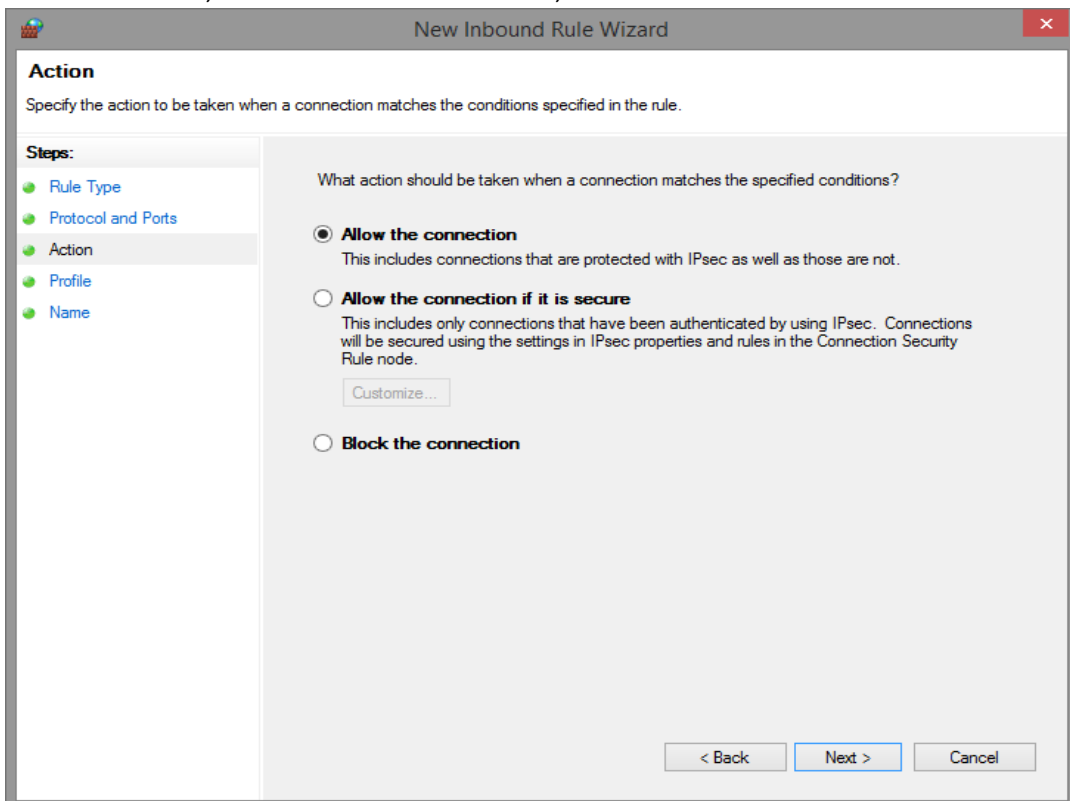
7. In **Profile** section, check for **Domain**, **Private** and **Public**, click **Next**.
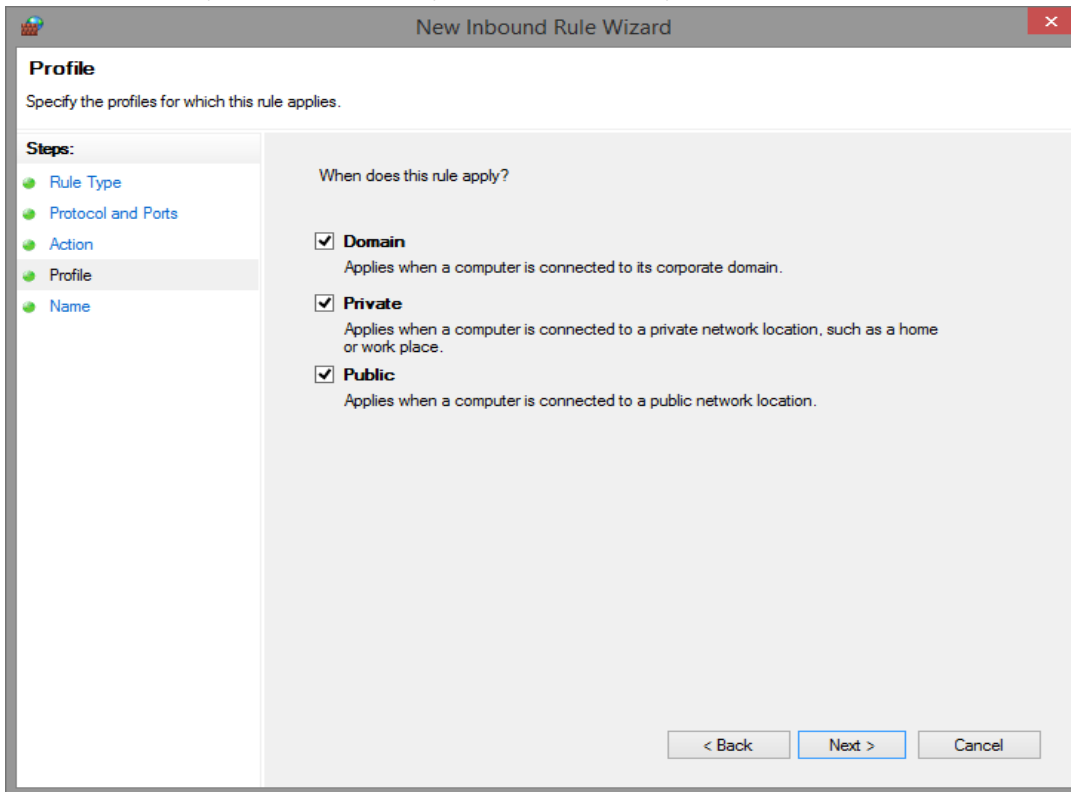
8. Give Appropriate **Name** and **Description,** click **Finish**.
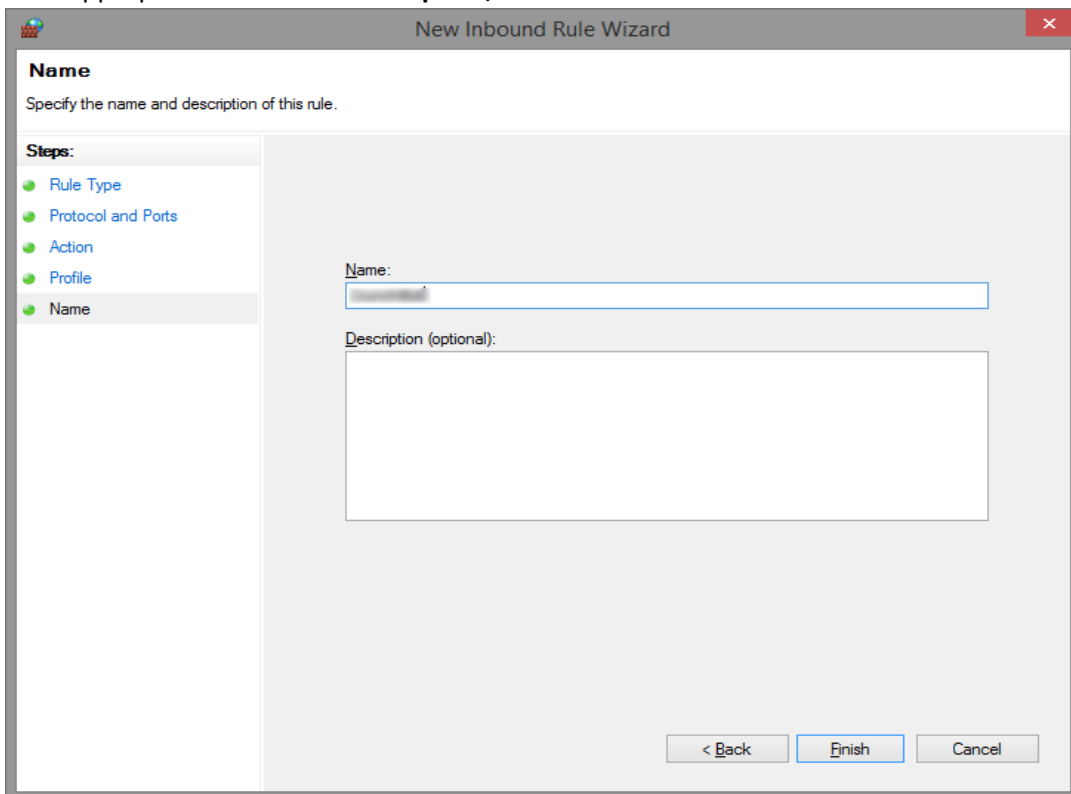
9. Follow the same steps to set **Outbound Rules** for the ports mentioned above. Also, change the action to **Allow the connection** at the time of creating Outbound Rule.
   *Important: Make sure the ports used by **Qlik Services** must be opened in Firewall.*

SSL Certificate

A valid SSL certificate, a certificate with certificate authority (CA), will be assigned to Qlik Insight Bot Web services and to Qlik Insight Bot NLU in order to exchange the information securely between client-server and Qlik Insight Bot NLU-other components respectively.

Right to create a sheet in Qlik Sense

When user asks a question to get a chart which is not already present in master visualizations, QIB creates on-the-fly chart. The on-the-fly chart object is created in a new sheet of Qlik Sense app. Hence, it is required to have a sheet creation permission to the user who is interacting with QIB.

Below rules needs an update permission:

1. Stream
2. CreateAppObjectsPublishedApp

Installing Docker and Docker Compose on Linux

Docker Compose is a tool for defining and running multi-container Docker applications. With Compose, you can use a YAML file to configure application's services. Then, with a single command, you can create and start all the services from the configuration.

Below are the steps to download Docker installation package from the official Docker repository. Connect to Linux shell and execute following commands.
Important: Make sure the command is in a single line only.

On Ubuntu

Installing Docker

1. First, let's ensure if the download is valid using the command below.

    ```
    $ curl -fsSL https://download.docker.com/linux/ubuntu/gpg |
    sudo apt-key add -
    ```

2. Now, you need to add Docker repository to APT sources.

    ```
    $ sudo add-apt-repository "deb [arch=amd64]
    https://download.docker.com/linux/ubuntu $(lsb_release -cs)
    stable"
    ```

3. After adding Docker repository to APT sources, update the package database with the Docker packages from the newly added repository.

    ```
    $ sudo apt-get update
    ```

4. Before installing make sure that you are installing from the Docker repository instead of the default Ubuntu repository.

    ```
    $ apt-cache policy docker-ce
    ```

    you should see the output similar to the below -

    ```
    docker-ce:
        Installed: (none)
        Candidate: 18.06.1~ce~3-0~ubuntu
        Version table:
           18.06.1~ce~3-0~ubuntu 500
              500 https://download.docker.com/linux/ubuntu
    xenial/stable amd64 Packages
    ```

5. Install Docker.

    ```
    $ sudo apt-get install -y docker-ce
    ```

6. Docker should be installed now. Check that it's running.

```
$ sudo systemctl status docker
```

you should see the output similar to the below -

```
docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled;
           vendor preset: enabled)
   Active: active (running) since Thu 2018-10-18 20:28:23 UTC;
           35s ago
     Docs: https://docs.docker.com
 Main PID: 13412 (dockerd)
   CGroup: /system.slice/docker.service
           ├─13412 /usr/bin/dockerd -H fd://
           └─13421 docker-containerd --config
                        /var/run/docker/containerd/containerd.toml
```

Executing the Docker Command without Sudo

1. By default, running the docker command requires root privileges, that is, you must prefix the command with **sudo**. It can also be run by a user in the docker group, which is automatically created during the installation of Docker.

2. If you want to avoid typing sudo whenever you run the docker command, add your username to the docker group:

```
$ sudo usermod -aG docker ${USER}
```

3. To apply new group membership:

```
$ su - ${USER}
```

4. This will ask to enter password to continue. To confirm user is added to the docker group.

```
$ id -nG
```

Installing Docker Compose

1. To download and install the latest version of Docker Compose execute below commands one after another.

```
$ sudo curl -L
"https://github.com/docker/compose/releases/download/1.23.1/docker-
compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
$ sudo chmod +x /usr/local/bin/docker-compose
```

-OR-

1. If you encounter any error or an issue, execute below set of commands to do the same.

```
$ sudo apt-get -y install python-pip
$ sudo pip install docker-compose
```

2. Check the installation is successfully completed or not

```
$ docker-compose --version
```

For detailed guide visit this.

On CentOS

Setting up the repository

1. Open terminal of your Linux machine.
2. First, install required packages.

```
$ sudo yum install -y yum-utils device-mapper-persistent-
data lvm2
```

3. Use the following command to set up the stable repository.

```
$ sudo yum-config-manager --add-repo
https://download.docker.com/linux/centos/docker-ce.repo
```

Installing Docker

1. Install the latest version of the Docker CE and containerd. Also, type "Y" whenever it prompts for your permission.

```
$ sudo yum install docker-ce docker-ce-cli containerd.io
```

2. Once the docker package is installed, use below commands to start the docker demon and enable it to automatically start at boot time.

```
$ sudo systemctl start docker
```
```
$ sudo systemctl enable docker
```

3. Verify that the docker service is running.

```
$ sudo systemctl status docker
```

You should see the output similar to the below -

```
docker.service - Docker Application Container Engine
    Loaded: loaded (/usr/lib/systemd/system/docker.service;
disabled; vendor preset: disabled)
    Active: active (running) since Thu 2019-01-24 14:46:55
CET; 7h ago
      Docs: https://docs.docker.com
 Main PID: 7399 (dockerd)
     Tasks: 25
    Memory: 251.6M
    CGroup: /system.slice/docker.service
            ├─ 7399 /usr/bin/dockerd -H fd://
```

Executing the Docker Command without Sudo

1. By default, running the docker command requires root privileges, that is, you must prefix the command with **sudo**. It can also be run by a user in the docker group, which is automatically created during the installation of Docker.
2. If you want to avoid typing sudo whenever you run the docker command, add your username to the docker group:

```
$ sudo usermod –aG docker ${USER}
```

3. To apply new group membership:

```
$ su – ${USER}
```

4. This will ask to enter password to continue. To confirm user is added to the docker group.

```
$ id –nG
```

Installing Docker Compose

1. To download and install the latest version of Docker Compose execute below commands one after another.

```
$ sudo curl -L
"https://github.com/docker/compose/releases/download/1.23.1/docker-
compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
$ sudo chmod +x /usr/local/bin/docker-compose
```

-OR-

1. If you encounter any error or an issue, execute below set of commands to do the same.

```
$ sudo yum install python-pip
$ sudo pip install docker-compose
```

2. Check the installation is successfully completed or not

```
$ docker-compose -version
```

On RHEL

Setting up the repository

1. Open terminal of your Linux machine.
2. Use the following commands to set up the stable repository.

```
$ sudo yum install -y yum-utils
$ sudo yum-config-manager --add-repo
https://download.docker.com/linux/centos/docker-ce.repo
$ sudo yum makecache fast
```

Installing Docker

1. Install the latest version of the Docker CE. Also, type "Y" whenever it prompts for your permission.

```
$ sudo yum -y install docker-ce
```

2. Let's verify that the docker service is running.

```
$ sudo systemctl status docker
```

You should see the output similar to the below -

```
docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service;
disabled; vendor preset: disabled)
   Active: active (running) since Thu 2019-01-24 14:46:55
CET; 7h ago
     Docs: https://docs.docker.com
 Main PID: 7399 (dockerd)
    Tasks: 25
   Memory: 251.6M
   CGroup: /system.slice/docker.service
           ├─ 7399 /usr/bin/dockerd -H fd://
```

Executing the Docker Command without Sudo

1. By default, running the docker command requires root privileges, that is, you must prefix the command with **sudo**. It can also be run by a user in the docker group, which is automatically created during the installation of Docker.
2. If you want to avoid typing sudo whenever you run the docker command, add your username to the docker group:

```
$ sudo usermod -aG docker ${USER}
```

3. To apply new group membership:

```
$ su - ${USER}
```

4. This will ask to enter password to continue. To confirm user is added to the docker group.

```
$ id -nG
```

Installing Docker Compose

1. To download and install the latest version of Docker Compose execute below commands one after another.

```
$ sudo curl -L
"https://github.com/docker/compose/releases/download/1.23.1/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
$ sudo chmod +x /usr/local/bin/docker-compose
```

-OR-

1. If you encounter any error or an issue, execute below set of commands to do the same.

```
$ sudo yum install python-pip
$ sudo pip install docker-compose
```

2. Check the installation is successfully completed or not

```
$ docker-compose –version
```

--- EOD ---