Ex 1 - Shopping 5.5 - Installing and configuring ActiveEfficiency

Exercise Overview:

Installing and configuring ActiveEfficiency

ActiveEfficiency Server is a core component of 1E solutions and is used to discover and store information about the IT environment from various data sources. 1E Shopping uses ActiveEfficiency to obtain details of users and computers that have been discovered by ConfigMgr.
In this lab, you will learn how to install and configure ActiveEfficiency Server for use with Shopping. You will also install the ActiveEfficiency Scout and run it to gather the data that Shopping requires from ConfigMgr.

ActiveEfficiency Server is used in different ways by other 1E solutions. This course covers the basic steps required to install ActiveEfficiency with the minimal components required for use with Shopping. Please refer to the ActiveEfficiency documentation if you are planning to implement other 1E solutions in the same environment.

Install ActiveEfficiency Server

In this exercise, you will install ActiveEfficiency Server, which is comprised of the core database and web-based API used by 1E solutions.

In this lab environment, ActiveEfficiency Server is installed on the same server as Shopping. For production implementations, ActiveEfficiency Server should generally be allocated a separate server with suitable hardware specification to support all the 1E solutions in use.

Install required Windows Role Services and Features

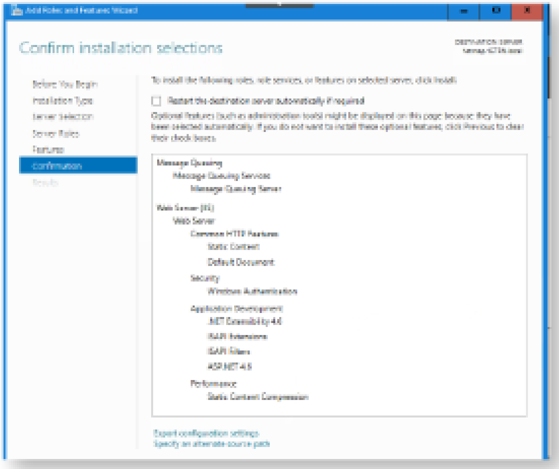
ActiveEfficiency requires the ASP.NET 4.6 and Static Content Compression Web Server Role Services and the Message Queuing Feature to be installed, which you will now do in this task.

1ETRNAP

1. Log on to **1ETRNAP** as **1ETRN\AppInstaller** and start **Server Manager**
2. From the **Manage** menu (top-right menu bar) select **Add Roles and Features** to start the Add Roles and Features Wizard
3. On the *Before you begin* page click **Next**
4. On the *Installation Type* page ensure **Role-based or feature-based installation** is selected and click **Next**
5. On the *Server Selection* page, ensure the local server (**1ETRNAP.1ETRN.LOCAL**) is selected and click **Next**
6. On the *Server Roles* page, locate and expand the **Web Server (IIS)** server role then expand the **Web Server** role service below that
7. Expand **Common HTTP Features** and select **Default Document** and **Static Content**
8. Expand **Performance** and enable **Static Content Compression**
9. Expand **Security** and select **Windows Authentication**

   Windows Authentication is not a requirement for ActiveEfficiency, although it can optionally be used to control access to data through the ActiveEfficiency web API. It is however a requirement for Shopping, which you will be installing later.

10. Expand **Application Development** and select **ASP.NET 4.6**. You will be prompted to add the ISAPI Filters, ISAPI Extensions and .NET Extensibility 4.6 role services required by ASP.NET 4.6. Click **Add Features** to include these then click **Next**
11. On the *Features* page, click **Next**
12. The *Confirmation* page should now show the following Role Services and Features



Confirm installation selections

To install the following roles, role services, or features on selected server, click Install.

□ Restart the destination server automatically if required

Optional features (such as administration tools) might be displayed on this page because they have been selected automatically. If you do not want to install these optional features, click Previous to clear their check boxes.

```
Message Queuing
    Message Queuing Services
        Message Queuing Server
Web Server (IIS)
    Web Server
        Common HTTP Features
            Static Content
            Default Document
        Security
            Windows Authentication
        Application Development
            .NET Extensibility 4.6
            ISAPI Extensions
            ISAPI Filters
            ASP.NET 4.6
        Performance
            Static Content Compression
```

13. Click **Install**
14. When the installation completes, close the wizard

Install ActiveEfficiency

In this task, you will run the ActiveEfficiency installer on the Application server (**1ETRNAP**).

1ETRNAP

15. Launch the SkyTap Shared Drive shortcut on the desktop and navigate to **1EActiveEfficiency\activeefficiency.v1.9.910.4** and copy **ActiveEfficiencyServer.msi** to **C:\Temp**
16. Start a command prompt using the **Run as administrator** option and switch to the **C:\Temp** directory
    If User Account Control is enabled, the installer may not install the components correctly if it executed by double-clicking the MSI file. It is good practice to install ActiveEfficiency from a command prompt with elevated privileges to avoid any issues with User Account Control.
17. Run the following command line to start the ActiveEfficiency installation wizard

    ```
    msiexec /i ActiveEfficiencyServer.msi /l*v AEServer_Install.log
    ```

    If your environment uses a Proxy Server for HTTP access, you will need to define the PROXYURI property on this command line, using the format PROXYURI="http://<proxyserver>:<port>".
18. On the *Welcome* page click **Next**
19. On the *License Agreement* page, accept the license agreement and click **Next**
20. On the *Prerequisite Checks* page ensure all checks are passed except MSMQ and click **Next**
    ActiveEfficiency uses MSMQ for synchronization of data used by some 1E solutions with the cloud-based ActiveEffiency service. Shopping does not use this feature of ActiveEfficiency yet, so in this lab it is not necessary to install MSMQ. If MSMQ is not installed (as is the case in this lab), the ActiveEfficiency Service (which manages the cloud synchronization) will not be installed either. In production environments, if there is a possibility that other 1E solutions that have cloud-sync functionality are to be installed, it would be best to install MSMQ and the ActiveEfficiency Service at the outset rather than modifying the installation at a later date.
21. On the *Enable 1E ActiveEfficiency Cloud Connection*, **uncheck** the option to register with 1E ActiveEfficiency and click **Next**
22. On the *Destination Folder* page click **Next**
23. On the *Database Server* page, select the **(local)** SQL server from the drop down and click **Next**
24. On the *ActiveEfficiency Website Settings* page click **Next** to use the default settings
25. On the *Nomad synchronization* page, click **Next** to accept the default value of Nomad Sync being disabled
26. On the *Ready to Install the Program* page click **Install** then click **Finish** when the installation completes
    When setup gets to the point of installing the database, it will appear as if nothing is happening for a few minutes. This is normal and the setup will continue after the background tasks are completed.

Install the ActiveEfficiency Scout

The ActiveEfficiency Scout is the component that collects information about computers and users from the ConfigMgr database and stores it in the ActiveEfficiency database. Shopping will, in turn, retrieve this data from the ActiveEfficiency database and store it locally in the Shopping database.

The ActiveEfficiency Scout is executed manually from either a command line or using a Windows Scheduled Task. While it is possible to create the Scheduled Tasks manually, the Scout includes the *ActiveEfficiency Synchronization Manager*, which provides a simple user interface and automates the creation of the necessary Scheduled Tasks.
The user account that executes the Scout must be granted read access to the ConfigMgr site database. The 1ETRN\AppInstaller account will be used to install and execute the Scout in this lab environment.

In a multi-site ConfigMgr 2012 hierarchy, the Scout would connect to the Central Administration Site (CAS) database.

In this exercise, you will grant the **1ETRN\AppInstaller** account the **db_datareader** role on the ConfigMgr database, install the ActiveEfficiency Scout and use the ActiveEfficiency Synchronization Manager to schedule the Scout to collect the necessary data from ConfigMgr.

Grant the AppInstaller account read access to ConfigMgr

The scout installation requires read access to the ConfigMgr database. As you are installing the Scout on the Application Server, you are logged in as the AppInstaller user. In this task, you will assign the AppInstaller account the db_datareader role on the ConfigMgr database.

1ETRNCM

27. Log on to **1ETRNCM** as **1ETRN\SCCMAdmin**
28. Start **SQL Server Management Studio** and connect to the local SQL server
29. Expand the **Security** node and expand the **Logins** node
30. Double-click on **1ETRN\AppInstaller** to open its properties
31. Select the *User Mapping* Page. Select the **CM_PS1** database from the list in the *Users mapped to this login* section
32. Select **db_datareader** form the list in the *Database role membership for: CM_PS1* section of the *User Mapping* page and click **OK**
    If the account didn't exist on this SQL server instance, you would have to create the account in SQL and provision the rights listed above.

Install the ActiveEfficiency Scout

1ETRNAP

33. Log on to **1ETRNAP** as **1ETRN\AppInstaller**
34. Launch the SkyTap Shared Drive shortcut on the desktop and navigate to **1EActiveEfficiency\activeefficiency.v1.9.910.4** and copy **ActiveEfficiencyScout.msi** to **C:\Temp**
35. Start a command prompt using the **Run as administrator** option and switch to the **C:\Temp** directory
    If User Account Control is enabled, the installer may not install the components correctly if it executed by double-clicking the MSI file. It is good practice to install the Scout from a command prompt with elevated privileges to avoid any issues with User Account Control. We can also specify a log file for the installed via the command line.
36. Run the following command line to start the Scout installation wizard

    ```
    msiexec /i ActiveEfficiencyScout.msi /l*v AEScout_Install.log
    ```

37. On the *Welcome* page click **Next**
38. On the *License Agreement* page, accept the license agreement and click **Next**
39. On the *Component Selection* page leave the **ConfigMgr** component checked using the **Standard** option, ensure **Server** and **iQSonar** are unchecked and click **Next**
40. On the *Destination Folder* page click **Next** to use the default location
41. On the *1E ActiveEfficiency Server* page click **Next** (ActiveEfficiency is installed on this server)
42. On the ConfigMgr Database page enter the following details and click **Next**
    ConfigMgr Database Server: **1ETRNCM**
    ConfigMgr Database Name: **CM_PS1**
43. On the *1E Shopping Configuration Manager Setup* page click **Next**
    Here you can set a different Configuration Manager database server and database name to use when running the Scout for retrieving information for Shopping. For example, you may want to configure the Scout to retrieve information for AppClarity once a night from an offline ConfigMgr reporting server that itself is updated once a day, but configure the Scout for Shopping to retrieve information every 60 minutes from a live ConfigMgr database server. In this lab environment, there is only one ConfigMgr database, so no configuration is required on this page.
44. On the *Ready to Install the Program* page click **Install**
45. When the installation wizard completes, ensure that the **Launch 1E AE Sync Manager** option is selected and click **Finish**

Schedule the Scout using the ActiveEfficiency Synchronization Manager

The ActiveEfficiency Synchronization Manager provides a simple interface to automate the creation of the necessary scheduled tasks required to execute the scout periodically and keep the ActiveEfficiency data in sync with the ConfigMgr data.

The ActiveEfficiency Scout is typically scheduled to run once per day and retrieves a lot of inventory data from ConfigMgr that is required by other 1E products (such as AppClarity) but not needed by Shopping. The Scout can be executed on a more frequent schedule using a special configuration that only retrieves a much smaller subset of data required by Shopping.
In this task, the ActiveEfficiency Synchronization Manager will be configured to run the Scout using this alternate schedule with the configuration that only collects the specific data required by Shopping.

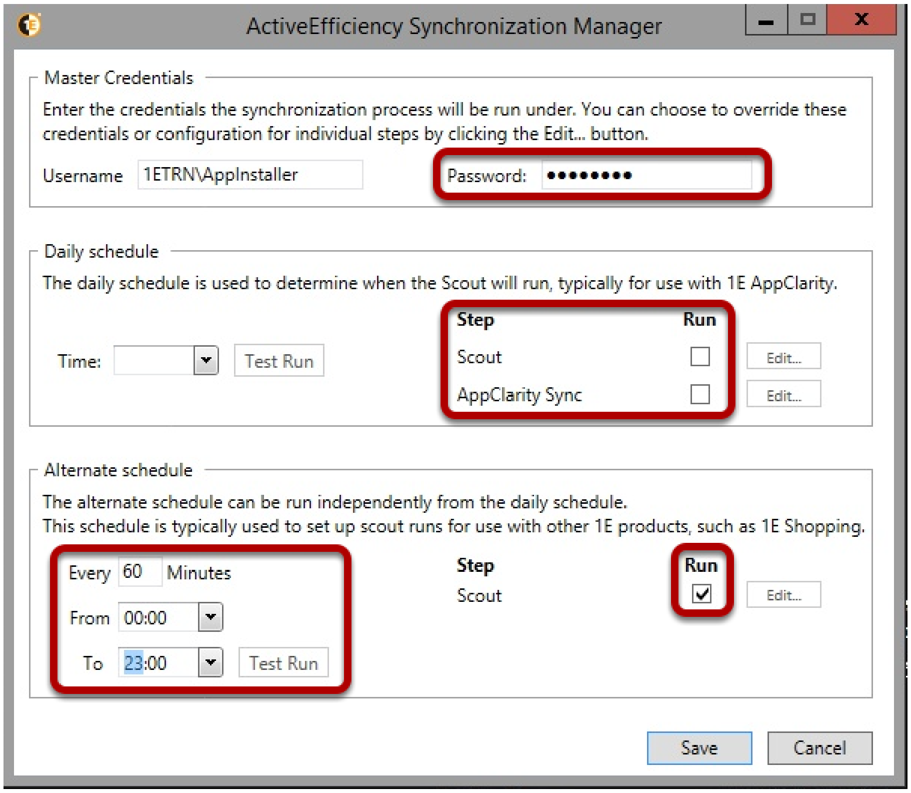
The ActiveEfficiency Synchronization Manager was launched by default after the Scout installation completed in the previous task. You can run it at any time from a command line **C:\Program Files (x86)\1E\ActiveEfficiency\Scout\N1E.ActiveEfficiency.SyncManager.exe**. The **1E AE Sync Manager** shortcut will also be available on the Start Menu when all applications are displayed.

1ETRNAP

46. In the *Master Credentials* section of the ActiveEfficiency Synchronization Manager, ensure the Username is **1ETRN\AppInstaller** and enter **Pass w0rd** in the Password field

Note that the Daily and Alternate schedules may be configured to override the master credentials when configuring the individual steps by clicking the **Edit…** button to the right of the Scout Run. If using the alternate credentials, you must ensure that the accounts used have a SQL Login on the ConfigMgr database server with the **db_datareader** role enabled.

47. In the *Daily schedule* settings, **uncheck** the Scout and AppClarity Sync steps so they do not run
    As we are only using ActiveEfficiency for Shopping there is no need to execute the full Scout on a daily schedule. If AppClarity were also in use, you would need to define a schedule for the Scout to do a 'full' run.
48. Configure the *Alternate schedule* to run every **30** minutes between **00:00** and **23:00** (be sure to check the **Run** checkbox)
49. Check that the settings reflect those shown in the screenshot below and click **Save**. You will see a dialog box pop up indicating the 1ETRN\AppInstaller account has been granted additional privileges to run the scheduled task. Click **OK**



ActiveEfficiency Synchronization Manager

Master Credentials

Enter the credentials the synchronization process will be run under. You can choose to override these credentials or configuration for individual steps by clicking the Edit... button.

Username 1ETRN\AppInstaller Password: ••••••••

Daily schedule

The daily schedule is used to determine when the Scout will run, typically for use with 1E AppClarity.

Time: Test Run

Step	Run
Scout	□
AppClarity Sync	□

Alternate schedule

The alternate schedule can be run independently from the daily schedule.
This schedule is typically used to set up scout runs for use with other 1E products, such as 1E Shopping.

Every 60 Minutes
From 00:00
To 23:00 Test Run

Step	Run
Scout	☑

Note that when you click Save, your settings will be saved but the Synchronization Manager will remain open (the Save button will be greyed out indicating there are no changes to save.

50. Close the ActiveEfficiency Synchronization Manager via the red X on the top right corner
    You will now observe the scheduled tasks that were created by the ActiveEfficiency Synchronization Manager.
51. Start the Windows **Task Scheduler** (from the Start screen, start typing **Task Scheduler** then click on **Task Scheduler** in the search results)
52. Expand the **Task Scheduler Library** and note the *ActiveEfficiency* folder created by the ActiveEfficiency Synchronization Manager
53. Note that the ActiveEfficiency Synchronization Manger has created a task in the **ActiveEfficiency** folder named **ActiveEfficiencySyncAlternate**.
    Note this task is scheduled to run at 12:00AM (00:00) every day and repeat every 60 minutes until 23:00
54. Select the **ActiveEfficiencySyncAlternate** task then select the **Actions** tab in the lower half of the Task Scheduler window and observe the command line that this task executes
    You will observe that the scheduled task does not actually run **scout.exe**. Instead, it runs **taskRunner.exe** with a **file** parameter that defines a configuration file (in this case **C:\ProgramData\1e\ActiveEfficiency\AlternateScheduledTask.xml**). Depending on the configuration options selected in the ActiveEfficiency Synchronization Manager, the scheduled task may have multiple operations to complete (such as completing an AppClarity Sync following the scout execution). These operations are defined in XML configuration files stored in **C:\ProgramData\1e\ActiveEfficiency**.
55. Open **C:\ProgramData\1e\ActiveEfficiency\AlternateScheduledTask.xml** in Notepad. Observe that this configuration file defines a command line to execute the Scout with the command-line argument **ApplicationConfigFile=Scout.Alternate.config**
56. Open **C:\Program Files (x86)\1E\ActiveEfficiency\Scout\Scout.Alternate.config** in Notepad. Note that the items related to ConfigMgr software usage and inventory will be skipped when the scout is executed with this config file. This data is not required by Shopping

```
<add key="ConfigMgrSoftwareUsageItem" value="skip"/>
<add key="ConfigMgrSoftwareInventoryItem" value="skip"/>
<add key="ConfigMgrSoftwareUsageProductItem" value="skip"/>
<add key="ConfigMgrSoftwareInventoryProduct" value="skip"/>
```

57. Close **AlternateScheduledTask.xml** and **Scout.Alternate.config** files, taking care that you have not made any accidental changes while they were open (if you are prompted to save the file, do not!). Leave the Task Scheduler open, as you will use it in the next exercise
    Note that the scheduled task will always run using the user account defined in the Master Credentials settings in the ActiveEfficiency Synchronization Manager. If you specified alternate credentials for the Alternate schedule in the ActiveEfficiency Synchronization Manager, these will be defined (in encrypted form) in the <Credentials> node in AlternateScheduledTask.xml and used for impersonation when scout.exe is executed by taskRunner.exe.

Getting data from ConfigMgr

In the previous task, you used the ActiveEfficiency Synchronization Manager to create a scheduled task to run the Scout using the alternate configuration every 60 minutes to get data that Shopping needs from ConfigMgr.

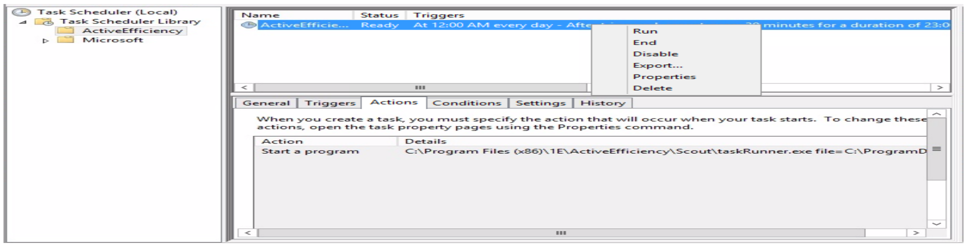
In this exercise, you will manually trigger that scheduled task (rather than waiting up to 60 minutes for the schedule to kick in) to retrieve properties of devices and users from ConfigMgr to support Shopping.

Run the Scout with the alternate configuration

In this task, you will manually trigger the **ActiveEfficiencySyncAlternate** scheduled task observed in the previous exercise to run the scout and get data from the ConfigMgr database.

1ETRNAP

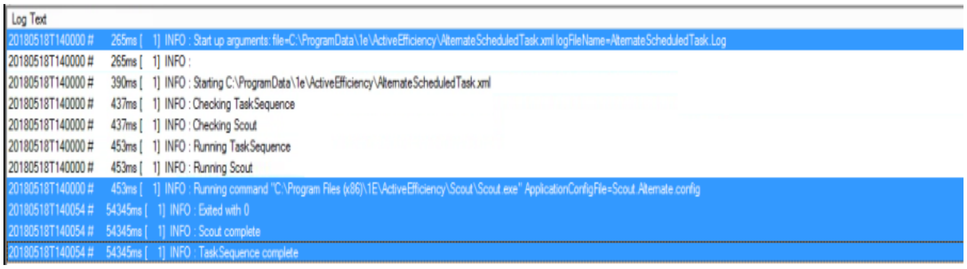
58. On **1ETRNAP**, open Task Scheduler from the Start menu, right-click the **ActiveEfficiencySyncAlternate** task and select **Run**



When you create a task, you must specify the action that will occur when your task starts. To change these actions, open the task property pages using the Properties command.

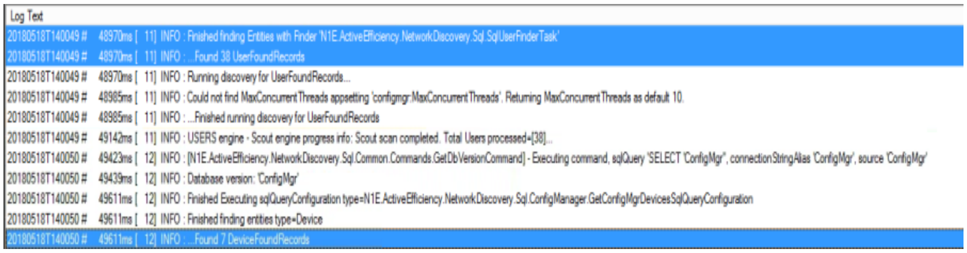
Action	Details
Start a program	C:\Program Files (x86)\1E\ActiveEfficiency\Scout\taskRunner.exe file=C:\ProgramD

59. Open **C:\ProgramData\1E\ActiveEfficiency\AlternateScheduledTask.log** and observe the activity as TaskRunner executes
    Remember the scheduled task executes **taskRunner.exe**. TaskRunner.exe logs its activity in a log file specified by the **logFileName** parameter on the **taskRunner.exe** command line. The ActiveEfficiency Synchronization Manager creates the ActiveEfficiencySyncAlternate task with **logFile Name=C:\ProgramData\1E\ActiveEfficiency\AlternateScheduledTask.log**.



```
Log Text
20180518T140000 #    265ms [   1] INFO : Start up arguments: file=C:\ProgramData\1e\ActiveEfficiency\AlternateScheduledTask.xml logFileName=AlternateScheduledTask.Log
20180518T140000 #    265ms [   1] INFO :
20180518T140000 #    390ms [   1] INFO : Starting C:\ProgramData\1e\ActiveEfficiency\AlternateScheduledTask.xml
20180518T140000 #    437ms [   1] INFO : Checking TaskSequence
20180518T140000 #    437ms [   1] INFO : Checking Scout
20180518T140000 #    453ms [   1] INFO : Running TaskSequence
20180518T140000 #    453ms [   1] INFO : Running Scout
20180518T140000 #    453ms [   1] INFO : Running command "C:\Program Files (x86)\1E\ActiveEfficiency\Scout\Scout.exe" ApplicationConfigFile=Scout.Alternate.config
20180518T140054 #  54345ms [   1] INFO : Exited with 0
20180518T140054 #  54345ms [   1] INFO : Scout complete
20180518T140054 #  54345ms [   1] INFO : TaskSequence complete
```

    Note that the timestamp on these log entries is in UTC.
    When the Scout is executed by TaskRunner using **Scout.Alternate.config**, the Scout logs its activity to **C:\ProgramData\1E\ActiveEfficiency\Scout_alternate.log**.
60. Open **C:\ProgramData\1E\ActiveEfficiency\Scout_alternate.log** and observe the activity as the scout runs. You should be able to identify how many users and devices were discovered in the ConfigMgr database



```
Log Text
20180518T140049 #  48970ms [  11] INFO : Finished finding Entities with Finder 'N1E.ActiveEfficiency.NetworkDiscovery.Sql.SqlUserFinderTask'
20180518T140049 #  48970ms [  11] INFO : ...Found 38 UserFoundRecords
20180518T140049 #  48970ms [  11] INFO : Running discovery for UserFoundRecords...
20180518T140049 #  48985ms [  11] INFO : Could not find MaxConcurrentThreads appsetting 'configmgr:MaxConcurrentThreads'. Returning MaxConcurrentThreads as default 10.
20180518T140049 #  48985ms [  11] INFO : ...Finished running discovery for UserFoundRecords
20180518T140049 #  49142ms [  11] INFO : USERS engine - Scout engine progress info: Scout scan completed. Total Users processed=[38]...
20180518T140050 #  49423ms [  12] INFO : [N1E.ActiveEfficiency.NetworkDiscovery.Sql.Common.Commands.GetDbVersionCommand] - Executing command, sqlQuery 'SELECT 'ConfigMgr'', connectionStringAlias 'ConfigMgr', source 'ConfigMgr'
20180518T140050 #  49439ms [  12] INFO : Database version: 'ConfigMgr'
20180518T140050 #  49611ms [  12] INFO : Finished Executing sqlQueryConfiguration type=N1E.ActiveEfficiency.NetworkDiscovery.Sql.ConfigManager.GetConfigMgrDevicesSqlQueryConfiguration
20180518T140050 #  49611ms [  12] INFO : Finished finding entities type=Device
20180518T140050 #  49611ms [  12] INFO : ...Found 7 DeviceFoundRecords
```

Observe the data returned by the Scout

The data retrieved in the previous task is stored in various tables in the ActiveEfficiency database. It can also be viewed through the ActiveEfficiency web service interface. In this task, you will observe the basic data that has been discovered by running the scout in ConfigMgr mode.

1ETRNAP

61. On **1ETRNAP**, open **Internet Explorer** and browse to **http://localhost/ActiveEfficiency* to open the *1E ActiveEfficiency web service** interface
62. Click the Uri link to the right of **Devices** and observe the devices that have been discovered from the ConfigMgr database. This information corresponds to the data in the Devices table in the ActiveEfficiency database
63. Click on the link in the **Identity** column that corresponds to the **1ETRNW71.1ETRN.local** device to view detailed properties of the device in the *Single Device resource* page
64. Click the back button in the browser until you return to the *1E ActiveEfficiency web service* page and then click the URI link to the right of **DiscoveryHistory.** This shows the start and end time of each execution of the Scout. This corresponds to the data in the DiscoveryHistory table in the ActiveEfficiency database
    In most environments, the scout will be installed just on the ActiveEfficiency server. However, the architecture allows the Scout to be installed on multiple computers, in which case you would see a row for each instance of the scout (identified by the computer name in the InstanceId column).

    You will notice that there is also an item called **Scouts** on the *1E ActiveEfficiency web service* page. This is left in for backwards compatibility so that previous Scout run information is preserved in an ActiveEfficiency upgrade scenario.
    The 1E ActiveEfficiency web service interface does not include a view of the User objects discovered by the scout. If you are interested in viewing these, you can look directly in the **Users** table in the ActiveEfficiency database. When the scout queries the ConfigMgr database, it records all the computers it discovers in the Devices table, and all the users in the Users table. Shopping will later retrieve the data from these two tables.

Lab Summary

In this lab, you have installed ActiveEfficiency Server and the ActiveEfficiency Scout and used the scout to gather data from the ConfigMgr database. You have learned how to confirm the data has been successfully gathered from the ConfigMgr database and to determine when the scout was last executed. You have learned how to use the ActiveEfficiency Synchronization Manager to create the scheduled task that runs the Scout using the alternate configuration (which skips collecting software inventory and usage data) frequently to ensure new computers and users are added to the ActiveEfficiency database in a timely manner. In the next lab, you will install Shopping and observe as it retrieves the computer and user information from ActiveEfficiency.

Next Page
Ex 2 - Shopping 5.5 - Installing Shopping