

Installation

- [Installing the Base Product](#)
- [Installing the Adapter](#)
- [Running DataSynapse GridServer® Director as Windows NT Service with the Adapter](#)
- [Uninstalling the Adapter](#)

Installing the Base Product

Before installing TIBCO DataSynapse™ High-Performance Computing Cloud Adapter, you must first install the TIBCO DataSynapse GridServer® product. The base product for the adapter is DataSynapse GridServer®. Read the following prerequisites to ensure that your DataSynapse GridServer® infrastructure is compatible for adapter installation.

Prerequisites

- Ensure that your DataSynapse GridServer® infrastructure is compatible for adapter installation. The adapter can be installed on one of the following:
 - A standalone Primary Director.
 - A TIBCO DataSynapse GridServer® Manager with a Primary Director and TIBCO DataSynapse GridServer® Broker enabled.

Procedure

1. Extract the DataSynapse GridServer® Manager installation archive and follow the DataSynapse GridServer® Manager installation and configuration instructions in the *TIBCO DataSynapse GridServer® Installation Guide*. Shut down the DataSynapse GridServer® Manager.
2. Install the latest hotfix for DataSynapse GridServer® Manager, if applicable to ensure your environment includes the most recent core product bug fixes and stability updates.
3. Restart the DataSynapse GridServer® Manager.

Result

You have successfully installed DataSynapse GridServer® Manager without any active engines.

Installing the Adapter

Use this procedure to install DataSynapse™ High-Performance Computing Cloud Adapter (HPCCA) and to restore configuration from the previous versions.

You can run the DataSynapse™ High-Performance Computing Cloud Adapter (HPCCA) (HPCCA) installation in the following 2 modes:

- Console Mode
- Silent Mode

Prerequisites

- Ensure that your system meets the platform and software requirements. For more information, see [Readme](#).
- DataSynapse GridServer® must be installed as described at [Installing the Base Product](#).
- If the DataSynapse GridServer® Manager is running, stop it before installing the adapter.
- DataSynapse GridServer® Engine and DataSynapse GridServer® Manager must be on the same or different Virtual Private Cloud (VPC).
- Avoid connecting DataSynapse GridServer® Engine to DataSynapse GridServer® Manager using the public internet.
- Test DataSynapse GridServer® setup with a manually installed DataSynapse GridServer® Engine.
- Take a backup of the `datasynapse/manager/webapps/livecluster/WEB-INF/lib` directory before installing DataSynapse HPCCA. If the installation fails, you can restore the lib folder using the backup folder.
- If you have previous version of `cloudprovider.properties` file in `../manager-data/hooks`, take the backup of these properties file.
- While installing the adapter, DataSynapse HPCCA also creates backup folder named as `GSBACKUP_BY_HPCCA`, which contains all libraries of GridServer from `datasynapse/manager/webapps/livecluster/WEB-INF/lib/` and certs from `../manager-data/certs`. This folder is useful when we uninstall the adapter.

Note: After HPCCA uninstallation, `GSBACKUP_BY_HPCCA` backup folder is also deleted.

- If you plan to install the adapter on a DataSynapse GridServer® Manager instance where SSL is enabled, then take a backup of the `manager-data/certs` folder. This is needed for post-installation.

Procedure

1. Extract the adapter's installation archive in the same folder that you extracted the base product's installation archive.
2. Run the batch or bash file for your system (`install.bat` or `install.sh`).

Console Mode: Example, for Linux OS run `./install.sh`. After running the `./install.sh` file, you are prompted to confirm the location of your manager and manager-data folder. If you have a manager and manager-data folder in the same directory, choose `yes`. But if manager and manager-data are in different directories, choose `no`. After this, you are prompted to provide locations of the required folders.

Silent Mode:

- a. For Windows:

```
install_silent.bat <Path_till_parent_folder_of_manager>
<Path_till_parent_folder_of_manager_data>
```

If you have a manager and manager-data folder in the same directory, then provide the same paths in both the parameters, for example:

```
install_silent.bat C:\TIBCO\DataSynapse C:\TIBCO\DataSynapse
```

- b. For Linux:

```
./install_silent.sh <Path_till_parent_folder_of_manager>
<Path_till_parent_folder_of_manager_data>
```

3. Update `./datasynapse/manager-data/hooks/Director/cloudprovider.properties` as follows:
 - a. In the first section, provide the general settings, such as which service provider you are using, the platform on which engines are to be launched, and Director details to be used for the newly launched engines.

- b. If you are using AWS Cloud Provider, update the AWS-Specific Configuration Properties.
 - c. If you are using Azure Cloud Provider, update the AzureSRG-Specific Configuration Properties.
 - d. If you are using Google Cloud Provider (GCP), update the Google Cloud Platform Specific Configuration Properties.
4. Start the DataSynapse GridServer® Manager.
 5. Verify that the adapter creates a number of engines equal to the `MINENGINES` property in the `cloudprovider.properties` file if `CONSIDER_NON_HPCCA_ENGINES` property is set to `false`.

Post-requisites

If you plan to install the adapter on a DataSynapse GridServer® Manager instance where SSL is enabled, then take a backup of the `manager-data/certs` folder. Then, you must import the certificates from the original `ssl.keystore` and `server.keystore` from that directory into the new `ssl.keystore` and `server.keystore`.

To merge the backup and the new certificate, use the following command-line example:

```
keytool -importkeystore -srckeystore PATH_OF_DIRECTOR_BACKUP_CERT/
ssl.keystore -srcstorepass changeit -destkeystore ../datasynapse/manager-
data/cert/ssl.keystore -deststorepass changeit
```

```
keytool -importkeystore -srckeystore PATH_OF_DIRECTOR_BACKUP_CERT/
server.keystore -srcstorepass changeit -destkeystore ../datasynapse/
manager-data/cert/server.keystore -deststorepass changeit
```

Running DataSynapse GridServer® Director as Windows NT Service with the Adapter

To start DataSynapse GridServer® Director as Windows NT service with the adapter and to use `AWS_AUTH_PROFILE` property in `cloudprovider.properties` file, perform the following steps so that the adapter can run smoothly over DataSynapse GridServer® Director:

Prerequisites

Note:

- If you are using custom `DS_DATA_DIR`, then for applying the DataSynapse GridServer® adapter as WIN NT service, you must set `DS_DATA_DIR` on environment variable.
- Ensure that you have installed DataSynapse HPCCA as described at [Installing the Adapter](#).

Procedure

1. Stop the Windows NT service for DataSynapse GridServer®.

```
..\datasynapse\manager\service.bat stop
```

2. Create an environment variable on the system. **Syntax:** `AWS_CREDENTIAL_PROFILES_FILE` = `PATH_TO_AWS_CREDENTIAL_FILE`

Example: `AWS_CREDENTIAL_PROFILES_FILE` = `C:\Users\Administrator\.aws\credentials`

3. Start the DataSynapse GridServer® NT service.

For more information regarding Director run as Windows NT Service, refer to "Installing DataSynapse GridServer" section in the TIBCO DataSynapse GridServer® Installation Guide.

Note: Make sure that you have all necessary permissions for AWS files like config, credentials, and the folder permissions, which contains AWS files.

Uninstalling the Adapter

Prerequisites

Before uninstalling the adapter, perform the following steps:

1. Stop the DataSynapse GridServer® Director.
2. Take a backup of `cloudprovider.properties` file.

3. Navigate to the extracted folder of the adapter's installation archive.

Procedure

1. Depending on your operating system, run the batch or bash file for your system (`uninstall.bat` or `uninstall.sh`) to uninstall HPCCA:

Console Mode: Example, for Linux OS run `./uninstall.sh` . After running the `uninstall.sh` file, you are prompted to confirm the location of your manager and manager-data folder. If you have a manager and manager-data folder in the same directory, choose `yes` . But if manager and manager-data are in different directories, choose `no` . After this, you are prompted to provide locations of the required folders.

Silent Mode:

- a. For Windows:

```
uninstall_silent.bat <Path_till_parent_folder_of_manager>
<Path_till_parent_folder_of_manager_data>
```

If you have a manager and manager-data folder in the same directory, then provide the same paths in both the parameters. For example:

```
uninstall_silent.bat C:\TIBCO\DataSynapse C:\TIBCO\DataSynapse
```

- b. For Linux:

```
./uninstall_silent.sh <Path_till_parent_folder_of_manager>
<Path_till_parent_folder_of_manager_data>
```

User Guide

- [Overview](#)
- [Requirements](#)
- [Setting Up the Engine](#)
 - [Creating an AWS Engine AMI](#)
 - [Creating an Azure VM Engine Image](#)
 - [Creating a Google Cloud Platform \(GCP\) Engine VM Image](#)
- [Configurations for the Adapter](#)
 - [General Configuration Properties](#)
 - [AWS-Specific Configuration Properties](#)
 - [Azure-Specific Configuration Properties](#)
 - [Google Cloud Platform \(GCP\) Specific Configuration Properties](#)

Overview

TIBCO DataSynapse™ High-Performance Computing Cloud Adapter extends the functionality of TIBCO DataSynapse GridServer® by enabling it to create dynamically virtual machines and provision them with the Engine Daemon.

You can use DataSynapse™ High-Performance Computing Cloud Adapter (HPCCA) to increase temporarily your compute capacity during periods of unusually high load on their DataSynapse GridServer® deployment. The adapter supports 64-bit Linux and Windows Engines, and a simple cloud management algorithm. DataSynapse HPCCA is implemented as a Manager Hook for the Director component of the grid. It uses DataSynapse GridServer® APIs to react and determine current cloud compute needs. The adapter uses a standard Java properties file that defines:

- The cloud service provider that it should interact with.
- The critical elements of information necessary for that interaction.

When new engines are needed, the adapter spins up new virtual machines, based on an image the customer creates, configures for the current grid. This process allows the adapter to remain cloud provider-agnostic, so new cloud solutions can be implemented faster.

Cloud Providers

The following cloud providers are supported: Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP). In all cases, the adapter uses the Java SDK for the service provider to connect to that provider's API.

Requirements

Before using DataSynapse™ High-Performance Computing Cloud Adapter (HPCCA), ensure that the following requirements are met:

- The adapter is installed on an existing implementation of a DataSynapse GridServer® Manager. This implementation of the DataSynapse GridServer® Manager needs to connect to the cloud service provider you want it to use. Each cloud service has instructions for testing, usually using a CLI implementation of their API. For information about supported DataSynapse GridServer® Manager versions, see the readme.
- The installation of the DataSynapse GridServer® Manager should be updated with the latest hotfix appropriate for that version. There are often fixes related to engines that are very beneficial for engine performance.

Test Your Cloud Network Infrastructure

The adapter is designed and tested with all components in the same or multiple networks/VPC, they are required when you want to open engines from different cloud providers. The DataSynapse GridServer® Manager's security group allows connections from the internet to port 8080 and SSH or RDP depending on the platform in which the manager is running and whether the engine's security group allows connections from the internet. This means that the engines and managers communicate through the private DNS and IP. To resolve the private DNS of the manager, create a DNS private zone and map the private IP with the private DNS, which is attached to a network. You must set up DataSynapse GridServer® Manager with its private DNS name during installation, even though you use the public DNS to access the manager administration tool.

Work with your cloud service provider to set up the correct network infrastructure. The possible configurations and the instructions for setting up VPN connections or security groups may vary from provider to provider. To test your network infrastructure, manually install an engine VM by using the cloud service account. Ensure that the engine VMs that you create in the cloud can connect to the brokers and directors for the particular grid.

Cloud service providers often use separate public and private DNS and IP. Contact your cloud service provider to set up the network architecture for your solution. Additionally, ensure that the manager and clients can connect to it unless you have disabled Direct Data Transfer (DDT). See the [TIBCO DataSynapse GridServer® Manager](#) documentation for more information.

Ensure that you test your network architecture as it might change the behavior of manager-engine communication.

Setting Up the Engine

DataSynapse HPCCA uses a prepared VM scale set/auto scaling group or Instance Group based on VM images with security settings aligned with your environment, Grid Libraries preinstalled, and the Engine Configuration set and tested. This provides you with the opportunity to know that the Engine VM is set up the way you need.

The following instructions for DataSynapse GridServer® Engine setup are a general case. Due to the inclusion of customer-editable startup scripts (User Data on AWS, VM Extension scripting on Azure, and startup script on Google Cloud Platform (GCP)), the specific setup of the Engine VM's image depends on your particular use case. A good way to test your particular scenario is to set the `MINENGINES` to `1` in the `cloudprovider.properties` file and see if the Engine starts normally when the Director starts.

Creating VM Scale Set on Azure

For details about creating a VM scale set on Azure, see [Create a virtual machine scale set in the Azure portal](#). Ensure that the following conditions are met when creating a VM scale set:

1. Ensure that the VM size selected has vCPUs same as `AZURE_VM_CORES` property in the `cloudprovider.properties` file.
2. For Orchestration mode, select Uniform mode.
3. Select the same Virtual Network on the **Networking** tab as the one you have used when creating the image for Engine VM.
4. On the scaling tab, set the scaling policy to **Manual**.

Creating Auto scaling Group on AWS

For details about creating an auto scaling group on AWS, see [Creating an Auto Scaling group using a launch template](#). Ensure that the following conditions are met when creating an auto scaling group:

1. Instance weights must be specified for the instance types. For more information about instance weighting, see [Configuring instance weighting for Amazon EC2 Auto Scaling](#).
2. Weight of the instance type must be equal to the number of vCPUs present in the instance type. You can select instance types with different vCPUs.
3. You must create an auto scaling group with an allocation strategy as **capacity-optimized**. For more information, see [Auto Scaling groups with multiple instance types and purchase options](#).
4. When creating an auto scaling group on the AWS portal, ensure that the **Minimum capacity** property and the **Desired capacity** property are set to **0**. DataSynapse HPCCA maintains the minimum VMs in auto scaling group by using the `MINENGINES` property in the `cloudprovider.properties` file.
5. Select the same security group in Launch Template as that used when creating the image for Engine VM.
6. Set the scaling policies as **None**.
7. Select the same network and subnet in the auto scaling group as that used when creating the image for DataSynapse GridServer® Engine VM.

Creating an Instance Group on Google Cloud Platform (GCP)

For details about creating an instance group on GCP, see [Creating managed instance groups](#). Ensure that the following conditions are met when creating an instance group from the GCP console:

1. In Create an instance group, select the New Managed Instance Group (stateless) option.
2. Set location to multiple zones and select the required region in which engine VMs are to be launched.
3. Select an instance template. For more information, see [Creating a Google Cloud Platform \(GCP\) Engine VM Image](#).
4. Disable Instance redistribution.
5. Click the **delete auto scaling configuration** button.

Creating an AWS Engine AMI

To use the adapter with Amazon AWS, you must first create an Engine AMI and an auto scaling group based on that AMI.

Prerequisites

Before creating an Engine Image, ensure that the following requirements are met:

1. Disable the firewall on the machine or open all the appropriate ports for the Engine.
2. Install the Engine. You can install the Engine with the Director you plan to use for production or for a Director you use for a development or QA grid. The adapter submits user data to AWS to reconfigure the Engine to use the Director for the grid.
3. Create an auto scaling group. For more information, see [Creating an Auto scaling Group on AWS](#).

Creating an Engine AMI on Linux

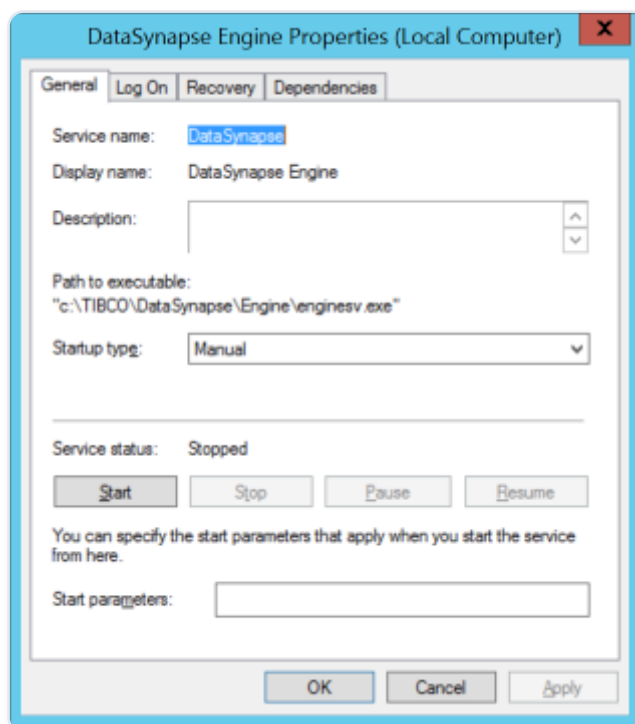
Procedure

1. Shut down the Engine Daemon after the Engine starts running against the first Director.
2. Verify that the Engine Daemon does not start automatically at boot.
3. Save the AMI.
4. Create a launch template using the AMI created above and User Data as given in [User Data for AWS](#).
5. Update `cloudprovider.properties` with auto scaling group details.

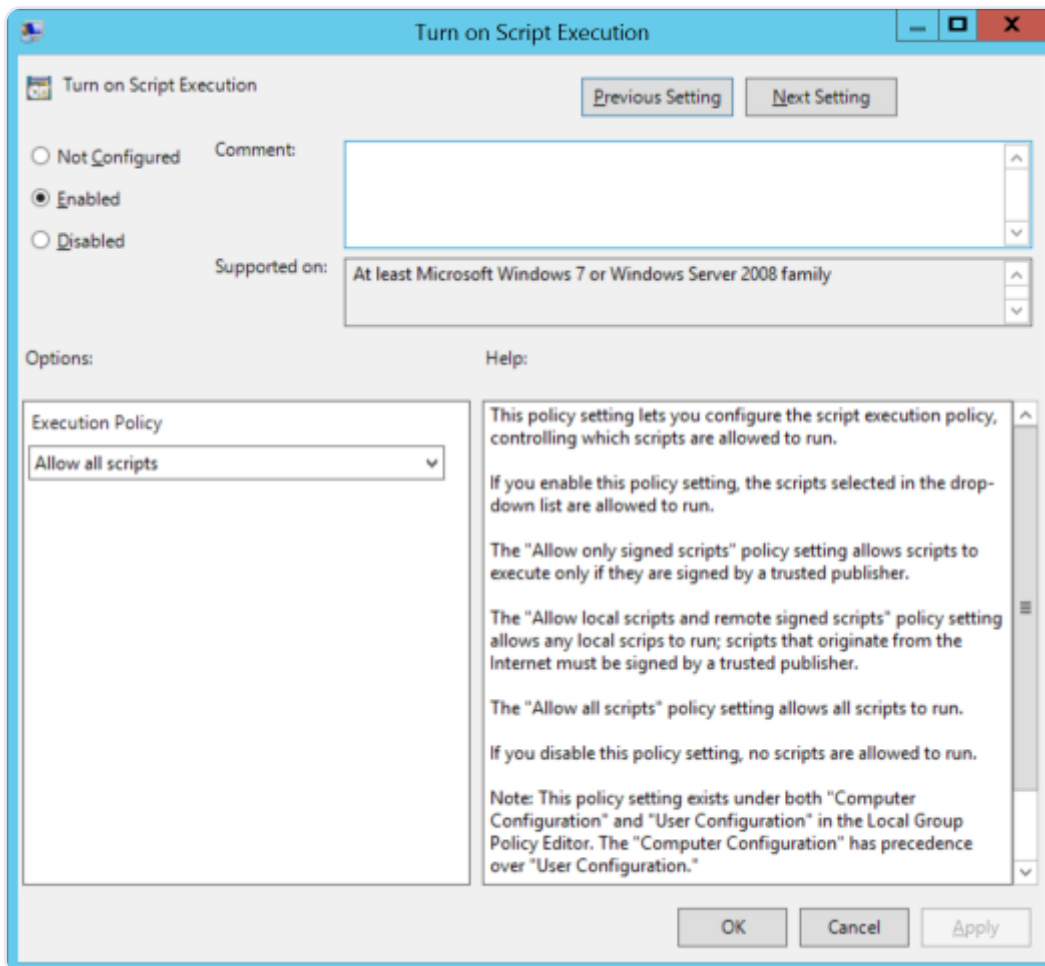
Creating an Engine AMI on Windows

Procedure

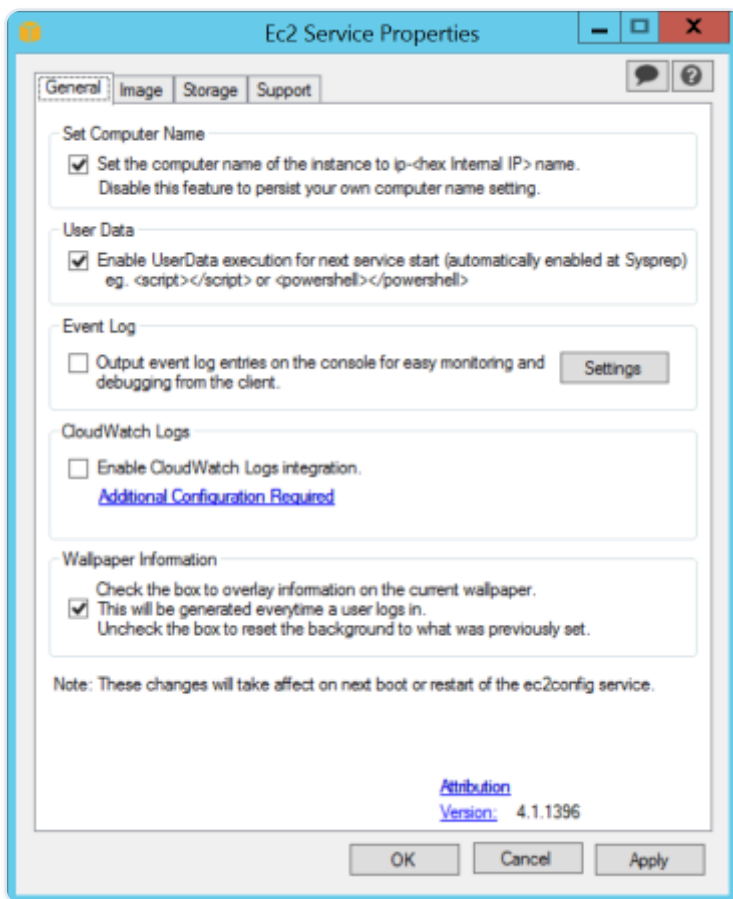
1. Shut down the Engine Daemon after the Engine starts running against the first Director.
2. Verify that the Engine Daemon does not start automatically at boot. This is very important for Windows since it installs as a service by default. The service needs to be set for Manual start.



3. Access `gpedit.msc` to enable PowerShell scripts. Go to **Local Computer Policy > Computer configuration > Administrative Templates > Windows Components > Windows PowerShell > Turn on Script Execution**.
4. In the **Turn on Script Execution** window, select **Enabled** in the upper-left, then in the **Options** section, select **Allow all scripts**.



5. Go to **Start > All Programs > EC2ConfigService > Settings**. On the **General** tab, select the checkbox next to **Enable** in the **User Data** section. Also select **Set the computer name of the instance...** in the **Set Computer Name** section.



6. Save the AMI.
7. Create a launch template using the AMI created from the previous step and User Data as given in [User Data for AWS](#).
8. Update `cloudprovider.properties` with the auto scaling group details.

User Data for AWS

Linux

```
#!/bin/bash

//Stopping the engine

cd /opt/Datasynapse/engine

./engine.sh stop
```

```
//running the configure utility

./configure.sh -s DIRECTOR_HOST:DIRECTOR_MESSAGE_PORT


//Re-Starting the engine

./engine.sh start
```

If you are configuring the adapter over an SSL-enabled Manager, use the following configure command instead:

```
//running the configure utility

./configure.sh -s DIRECTOR_HOST:DIRECTOR_MESSAGE_PORT -l y
```

Windows

```
<powershell>

Stop-Service DataSynapse

$dsfolder="C:\Tibco\Datasynapse\Engine"

$director="ip-172-31-0-39"

$directorPort="8000"

Remove-Item -Recurse -Force "$($dsfolder)\profiles\"

$datfile="$($dsfolder)\intranet.dat"

$contents = [IO.File]::ReadAllText($datfile)

$httpPrefixDelim="://"
```

```

        $endpos=$contents.IndexOf

($httpPrefixDelim)+$httpPrefixDelim.Length

        $newPrefix=$contents.Substring(0,$endpos)

        "$($newPrefix)$($director):$($directorPort)/livecluster

/public_html/register/register.jsp".Trim() | Out-File -Encoding Ascii
$datfile

        Start-Service DataSynapse

</powershell>

```

Creating an Azure VM Engine Image

To use the adapter with Azure, you must first create an Azure VM Engine Image for DataSynapse GridServer® Engine for your choice of OS.

Prerequisites

Before creating an Engine Image, ensure that the following requirements are met:

1. Disable the firewall on the machine or open all the appropriate ports for the Engine.
2. Install the Engine. You can install the Engine with the Director you plan to use for production or for a Director you use for a development or QA grid.

Creating an Engine Image on Linux

Procedure

1. Open any Linux VM on Azure and use Managed disk to open the VM.
2. Shut down the Engine Daemon after the Engine starts running against the first Director.
3. Verify that the Engine Daemon does not start automatically at boot time.

4. Create Azure VM scale sets (VMSS) by selecting the appropriate engine image. For more information, see [Creating VM Scale Set on Azure](#). For more information about saving VM Image disks in Azure, see [How to create a managed image of a virtual machine or VHD](#).
5. Once the VM scale set is created, add a custom script in extension for configuring engine machines that is launched in the VM scale set. For more information, see [Virtual machine extensions and features for Linux](#). Sample script:

```
#!/bin/bash

#Stopping the engine

cd /opt/datasynapse/engine

./engine.sh stop

#running the configure utility

./configure.sh -s DIRECTOR_HOST:DIRECTOR_MESSAGE_PORT

#Re-Starting the engine

./engine.sh start
```

If you are configuring the adapter over an SSL enabled Manager, then configure by using the following command:

```
//running the configure utility

./configure.sh -s DIRECTOR_HOST:DIRECTOR_MESSAGE_PORT -l y
```

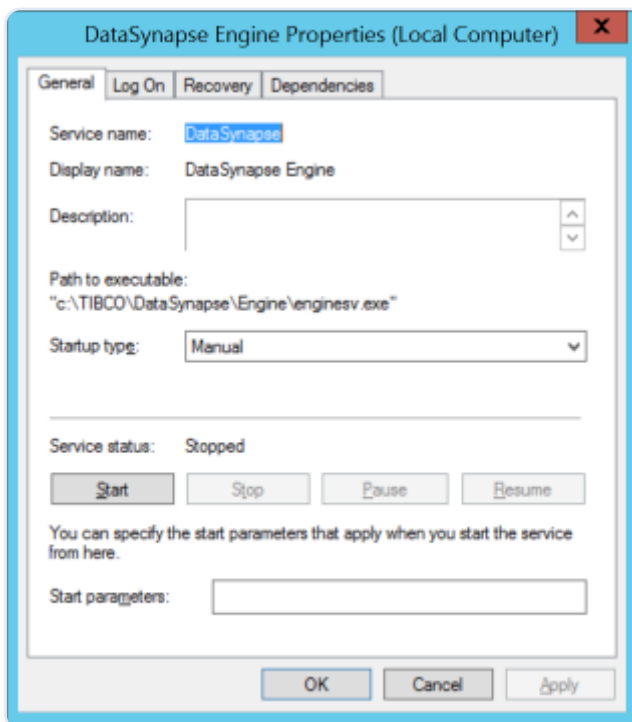
6. Update the VM scale set name in a `cloudprovider.properties` file for `VM_SCALESET_NAME` property.

Creating an Engine Image on Windows

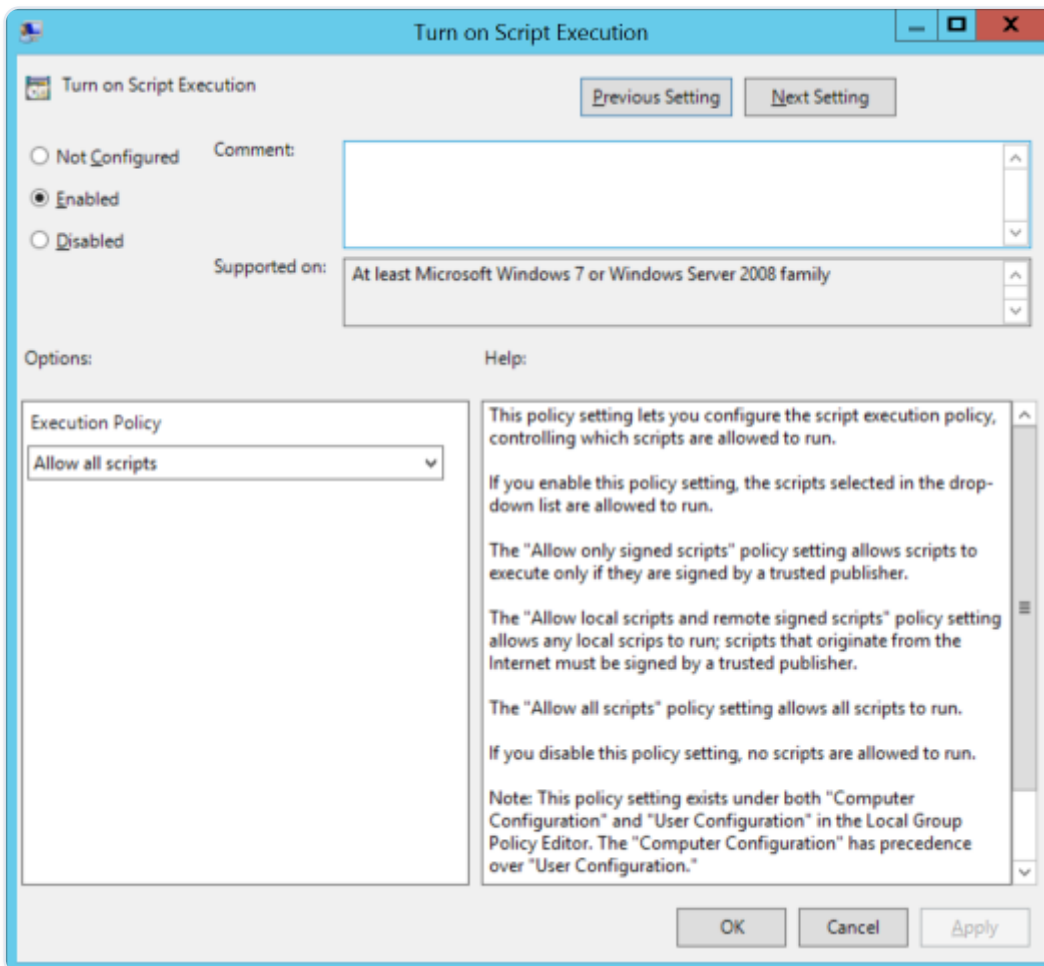
Procedure

1. Shut down the Engine Daemon after the Engine starts running against the first Director and delete the Daemon entry from the Director's **Daemon Admin** page.

2. Verify that the Engine Daemon does not start automatically at boot time. This is very important for Windows since it installs as a service by default. The service needs to be set for a manual start.



3. Access `gpedit.msc` to enable PowerShell scripts. Go to **Local Computer Policy > Computer configuration > Administrative Templates > Windows Components > Windows PowerShell > Turn on Script Execution**.
4. In the **Turn on Script Execution** window, select **Enabled** in the upper-left, then in the **Options** section, select **Allow all scripts**.



5. Create Azure VM scale sets (VMSS) by selecting the appropriate engine image. For more information, see [Creating VM Scale Set on Azure](#). For more information about saving VM Image disks in Azure, see [How to create a managed image of a virtual machine or VHD](#).
6. Once the VM scale set is created, add a custom script in extension for configuring engine machines that is launched in the VM scale set. Sample script:

```
Stop-Service DataSynapse

$directorIP="gridDirectorIp"

$directorPort="8000"

$datfile="C:\TIBCO\DataSynapse\engine\intranet.dat"

$newPrefix="http://"

"$($newPrefix)$($directorIP):$($directorPort)/livecluster/public_
html/register/register.jsp".Trim() | Out-File -Encoding Ascii $datfile
```

Start-Service DataSynapse

7. Update the VM scale set name in a `cloudprovider.properties` file for `VM_SCALESET_NAME` property.

Creating a Google Cloud Platform (GCP) Engine VM Image

To use the adapter with Google Cloud Platform (GCP), you must first create a Google Cloud Platform VM Engine Image for GridServer Engine for your choice of OS.

Prerequisites

Before creating an Engine Image, ensure that the following requirements are met:

1. Disable the firewall on the machine or open all the appropriate ports for the Engine.
2. Install the Engine. You can install the Engine with the Director you plan to use for production or for a Director you use for a development or QA grid.

Creating an Engine Image on Linux

Procedure

1. Shut down the Engine Daemon after the Engine starts running against the first Director.
2. Verify that the Engine Daemon does not start automatically at boot.
3. Create the image on GCP. For more information, see [Creating GCP Image](#).

Note: Ensure that you create an image using the `--guest-os-features` `MULTI_IP_SUBNET` flag by using Google Cloud.

4. Create instance templates by using the newly created image and provide a script in the Startup script field. The user must provide the startup script at the automation section during the creation of instance templates. Sample value script:

```
#!/bin/bash

//Stopping the engine

cd /opt/datasynapse/engine
```

```
./engine.sh stop

//running the configure utility

./configure.sh -s DIRECTOR_HOST:DIRECTOR_MESSAGE_PORT

//Re-Starting the engine

./engine.sh start
```

If you are configuring the adapter over an SSL enabled Manager, then configure by using the following command:

```
//running the configure utility

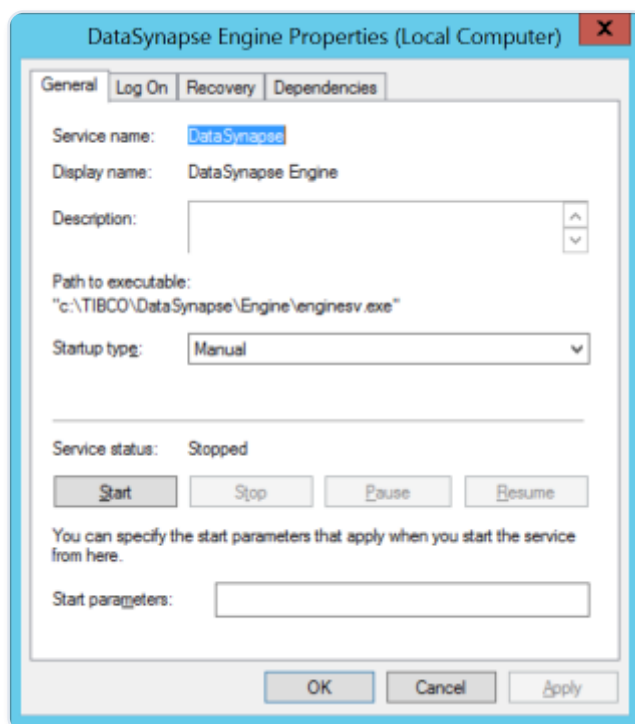
./configure.sh -s DIRECTOR_HOST:DIRECTOR_MESSAGE_PORT -l y
```

5. Create an Instance Group. For more information, see [Creating an Instance Group on Google Cloud Platform \(GCP\)](#).
6. Update the `cloudprovider.properties` file with the Instance Group details.

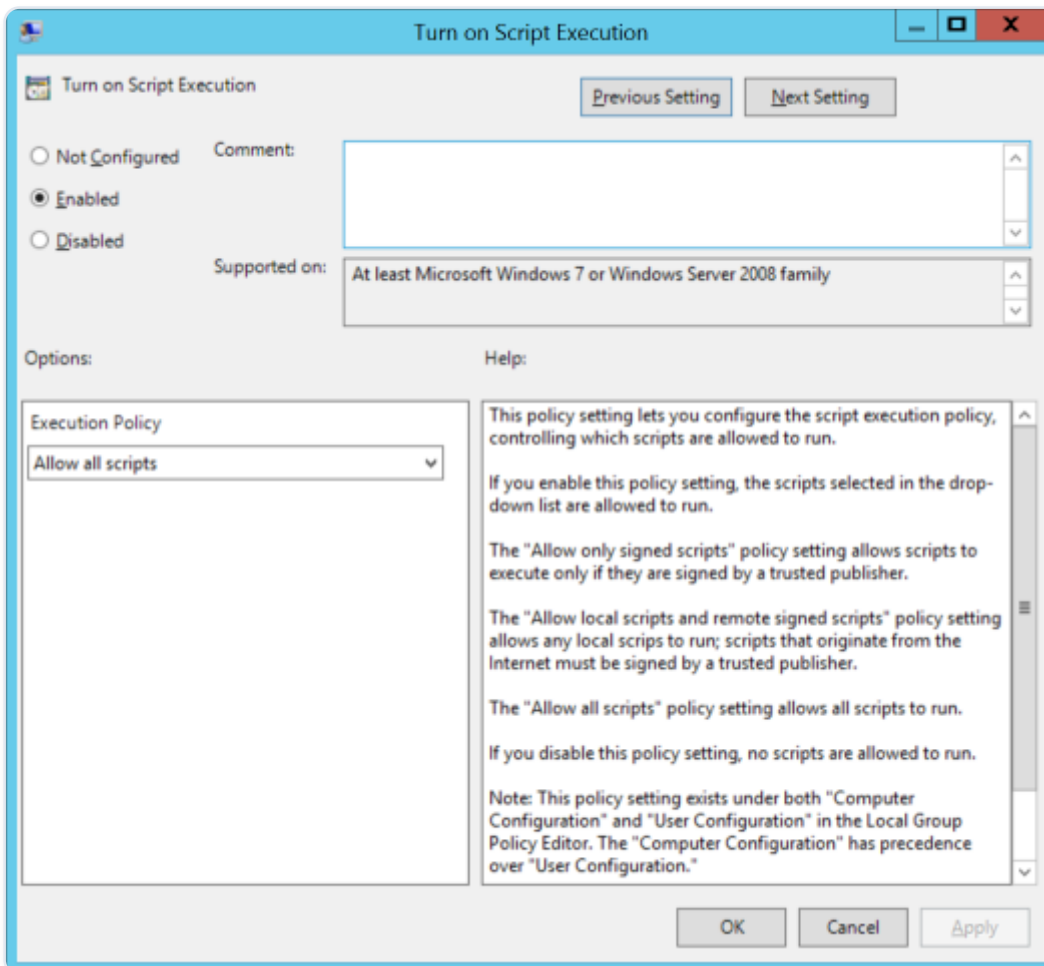
Creating an Engine Image on Windows

Procedure

1. Shut down the Engine Daemon after the Engine starts running against the first Director.
2. Verify that the Engine Daemon does not start automatically at boot time. This is very important for Windows since it installs as a service by default. The service needs to be set for a manual start.



3. Access `gpedit.msc` to enable PowerShell scripts. Go to **Local Computer Policy > Computer configuration > Administrative Templates > Windows Components > Windows PowerShell > Turn on Script Execution**.
4. In the **Turn on Script Execution** window, select **Enabled** in the upper-left, then in the **Options** section, select **Allow all scripts**.



5. Create the image on GCP. For more information, see [Creating Windows OS Image](#). Ensure that you perform the following steps during the creation of the image:

- Run `GCESysprep` utility.
- Create an image using the flag `--guest-os-features MULTI_IP_SUBNET` from the Google Cloud.

6. Create instance templates:

- Use the newly created image.
- Use the following key and script as a value in the custom metadata subsection under the Automation section.
- Key = `windows-startup-script-ps1`

Sample value script:

```
$director="gs-manager-rsm-director"

$messagePort= 8000
```

```

echo "Director options: $director " >> "C:\gsengine.log"

echo "messagePort: $messagePort " >> "C:\gsengine.log"

$datfile="$($dsfolder)\intranet.dat"

"http://$($director):$($messagePort)/livecluster/public_
html/register/register.jsp".Trim() | Out-File -Encoding Ascii $datfile

echo "Starting datasynapse service....." >> "C:\gsengine.log"

Start-Service DataSynapse

Start-Sleep -seconds 45

$datasynapseService = Get-Service -Name DataSynapse

if ($datasynapseService.Status -ne 'Running'){

    "Datasynapse Service is not running, trying to rerun it." >> "C:
\gsengine.log"

    Start-Service DataSynapse

    Start-Sleep -seconds 15

    $datasynapseService.Refresh()

    if ($datasynapseService.Status -eq 'Running')

    {

        "Datasynapse Service is running" >> "C:\gsengine.log"

```

```

    }

}

```

7. Create an Instance Group. For more information, see [Creating an Instance Group on Google Cloud Platform \(GCP\)](#).
8. Update the `cloudprovider.properties` file with the Instance Group details.

Configurations for the Adapter

The adapter is configured with properties set in the `cloudprovider.properties` file. The properties file is at, `TIBCO_HOME/datasynapse/manager-data/hooks/Director`. Here, `TIBCO_HOME` is defined as the directory where your TIBCO software installations reside. (For example, on Windows machines, it is typically `C:\TIBCO`.)

After making updates in the `cloudprovider.properties` file, you must restart the Director for the updates to take effect.

Depending on the service provider (except AWS), you must ensure that the `AZURE_VM_CORE` or `GCP_VM_CORE` property is equal to the number of vCPUs present in the instance type mentioned in the VM Scaleset in Azure or instance group in GCP. For example, if the `AZURE_VM_CORE` value is `2`, you must select the VM size that has vCPUs equal to 2 such as `Standard_D2S_V3` while creating an Azure VM scale set.

The default properties file is broken down into the following sections: Adapter-General, AWS-Specific, Azure-Specific, and Google Cloud Provider.

Property Value Obfuscation

You can obfuscate property values contained in the `cloudprovider.properties` file. There is a Java application in the `TIBCO_HOME/datasynapse/manager-data/hooks/Director` folder called `Obfuscator.jar`. This application translates property values into and out of the obfuscated format. If you take any value here and pass it to `Obfuscator.jar`, the Obfuscator generates a hash value with `OBF:` as the prefix. You must then take this value, including the prefix, and replace the original value in this file. For example:

```
java -jar Obfuscator.jar -obfuscate yourvalue
```

For example, if the original property is `WEBSERVICEUSER=myWebServiceID`, and the user wants to obfuscate the string `myWebServiceID`, then the user must enter the following at the command line:

```
java -jar Obfuscator.jar -obfuscate myWebServiceID
```

The Obfuscator obfuscates the string `myWebServiceID` and generates output similar to the following: `OFB:1o6g1rjj1qb51sox1ta01sv01vn61vnw1swy1tae1sot1qbx1rgv1o46`

You can then set the obfuscated string in the previous example as the property for `WEBSERVICEUSER`:

```
WEBSERVICEUSER=OFB:1o6g1rjj1qb51sox1ta01sv01vn61vnw1swy1tae1sot1qbx1rgv1o46
```

If you want, you can deobfuscate an already obfuscated value. You can do that with the `Obfuscator.jar` application. For example:

```
java -jar Obfuscator.jar -deobfuscate  
OFB:1o6g1rjj1qb51sox1ta01sv01vn61vnw1swy1tae1sot1qbx1rgv1o46
```

In this case, it would generate the value `myWebServiceID`.

Other Configuration Files

The following other configuration files can be used to modify the behavior of DataSynapse HPCCA:

- `HPCCHook.xml`: This is the definition file used by DataSynapse GridServer® when loading the adapter. Normally you would not need to modify this file. However, in some cases you may

need to change the `enabled` property to `True` to force the adapter to start when the DataSynapse GridServer® Director starts.

General Configuration Properties

The following general configuration properties are available as part of the `cloudprovider.properties` file at `./datasynapse/manager-data/hooks/Director/cloudprovider.properties`. These properties are used for all cloud service providers.

Adapter - General Properties

PROPERTY	DESCRIPTION
<code>SERVICE_PROVIDER</code>	Specify the service providers to be used with DataSynapse GridServer® Director. Valid values: AWS, AZURE, or GCP.
<code>AZURE_VM_CORE</code> , <code>GCP_VM_CORE</code>	The number of CPU cores of the VM size selected for the Engine. Ensure that the VM size mentioned in the VM scale set or instance group has vCPU equal to this property.
<code>AWS_ENGINE_WEIGHT</code> , <code>AZURE_ENGINE_WEIGHT</code> , <code>GCP_ENGINE_WEIGHT</code>	Weightage of DataSynapse GridServer® Engine for a particular Cloud Provider.
<code>BROKER_PROTOCOL</code>	Protocol of DataSynapse GridServer® Broker. Example: HTTP or HTTPS.
<code>BROKER_HOSTNAME</code>	List of DataSynapse GridServer® Broker hostname on which adapter engines configure. It only contains standalone DataSynapse GridServer® Broker, not failover Brokers.
<code>AGGRESSIVE_SCALEDOWN</code>	Set this property to true to let the DataSynapse HPCCA scale down the

PROPERTY	DESCRIPTION
	DataSynapse GridServer® Engine as soon as the task is completed. If this property is set to true, then DataSynapse HPCCA skips the <code>ENGINEIDLETIMEOUT</code> check and quickly scales down the DataSynapse GridServer® Engine when it finds that no task is running. If this property is set to false, then DataSynapse HPCCA starts calculating the idle timeout for the DataSynapse GridServer® Engine, and after the idle timeout exceeds the DataSynapse HPCCA spins down the DataSynapse GridServer® Engine.
<code>MINIMUM_BUSY_ENGINE</code>	This property is used to check count of busy engine on Engine Daemon. If HPCCA finds the number of busy Engines on Engine Daemon is less than or equal to <code>MINIMUM_BUSY_ENGINE</code>, the Engines on Engine Daemon are parked and becomes candidate for aggressive scale down.
<code>AGGRESSIVE_SCALEDOWN_RECHECKINTERVAL</code>	The time interval to monitor the working state of DataSynapse GridServer® Engine for aggressive scale down.
<code>AWS_ENGINE_OS</code> , <code>AZURE_ENGINE_OS</code> , <code>GCP_ENGINE_OS</code>	It is the Engine OS for a particular Cloud Provider.
<code>WEBSERVICEUSER</code> and <code>WEBSERVICEPW</code>	The user ID and password in DataSynapse GridServer® that the adapter uses to interact with the DataSynapse GridServer® web service API.

PROPERTY	DESCRIPTION
	The adapter calls the web service API to determine if DataSynapse GridServer® Engine has connected to the director.
GRIDSTATUSRECHECKINTERVAL	GRIDSTATUSRECHECKINTERVAL controls two key operations: Grid status check - Internal algorithm interval to check the need for VMs and either open new VMs or stop the idle ones based on the calculated need. Orphan VM detection - Termination of cloud VMs that fail to register a GridServer engine daemon, preventing runaway cost from untracked instances across configured cloud providers. - A built-in grace period of <code>30 minutes</code> is applied before a VM is flagged as orphaned, allowing newly launched VMs sufficient time to register with the broker.
ENGINEIDLETIMEOUT	The period for which the DataSynapse GridServer® Engine can be idle before it is flagged for termination, assuming the number of DataSynapse GridServer® Engine exceeds the <code>MINENGINES</code> value.
MINENGINES and MAXENGINES	The minimum and maximum number of DataSynapse GridServer® Engine VMs that the adapter is allowed to create. If <code>MINENGINES</code> is greater than 0, the adapter attempts to start DataSynapse GridServer® Engine when the server is started (provided

PROPERTY	DESCRIPTION
	that it does not find any VMs that appear to be DataSynapse GridServer® Engine VMs in the available region for the cloud service provider.)
ALGORITHM	The algorithm currently in use to determine the number of DataSynapse GridServer® Engine VMs needed at any time. Only the BASIC algorithm is defined. In the basic algorithm, we take the number of Engines needed as the following: Engines needed = (number of tasks that can run on the cloud) x (the allocation percentage as a decimal) Example If we have 10 Engines at the moment, our allocation percentage is 0.1 and the number of tasks we are running increases to 200, the basic algorithm starts 10 new Engines (200 x 0.1 = 20 Engines, 20 Engines needed - 10 active Engines = 10 new Engines needed)
TASKCOUNTFLOOR	This specifies the minimum number of pending tasks before DataSynapse GridServer® Engine VMs are started. This value is subtracted from the pending task total (minimum zero) when the BASIC algorithm determines the Engine need.
ALLOCATIONPERCENTAGE	This is specific to the BASIC algorithm and determines the number of DataSynapse GridServer® Engine VMs needed by multiplying the number of pending tasks by the allocation percentage to arrive at a target number of DataSynapse GridServer® Engine VMs. It then compares this value with the current number of DataSynapse GridServer®

PROPERTY	DESCRIPTION
	Engine to determine how many DataSynapse GridServer® Engine VMs must be created.
CONSIDER_NON_HPCCA_ENGINES	Set this property to <code>true</code> if DataSynapse HPCCA must consider non- DataSynapse HPCCA engines while calculating the engine need. If this property is set to <code>true</code> : • DataSynapse HPCCA considers the engines not launched from DataSynapse HPCCA for calculating the total number of engines. • DataSynapse HPCCA considers that the <code>MAXENGINES</code> property includes non- DataSynapse HPCCA engines. • DataSynapse HPCCA considers non- DataSynapse HPCCA engines while maintaining the <code>MINENGINES</code> property. For example, if the DataSynapse GridServer® Broker has 20 engines (DataSynapse HPCCA + non- DataSynapse HPCCA) and <code>MAXENGINES</code> property is set to 24, then DataSynapse HPCCA can open only 4 new engines until the value of <code>MAXENGINES</code> property is increased.
BATCH_VM_TERMINATION	Set this property to <code>true</code> to terminate VMs in batch size.
BATCH_VM_COUNT	Count of VMs to terminate in batch size.
USE_HPPCA_AUTO_RESTART	Set this property to true if HPCCA needs to restart itself (after changing <code>cloudprovider.properties</code> file) without restarting DataSynapse GridServer® Manager.

PROPERTY	DESCRIPTION
<code>DETECT_CHANGES_INTERVAL</code>	Time Interval for HPCCA to check for changes in <code>cloudprovider.properties</code> file.
<code>INSTANCE_TYPE</code>	Stores instance type information, whether it is a SPOT or ON-DEMAND Instance type for AWS, Azure, or GCP. This property is set on the Engine Daemon to track whether it is running on a spot/preemptible instance.

AWS-Specific Configuration Properties

The following properties are specific to adapter configuration in AWS:

PROPERTY	DESCRIPTION
<code>AWS_REGION</code>	The region in which you want to create DataSynapse GridServer® Engine instances.
<code>AWS_ACCESSKEY</code> and <code>AWS_SECRETKEY</code>	The access key and secret key are tied to a user in AWS and allow the Manager Hook to manage instances from within AWS. For more information on how to obtain your access key and secret key from AWS, see Getting started with Amazon SQS. If these values are omitted, and the property <code>AWS_AUTH_PROFILE</code> is also omitted, the adapter assumes that you are using Instance-based authentication.
<code>AWS_AUTH_PROFILE</code>	The name of the profile, which is authorized to open new instances. This value is only used when you are utilizing profile-based authentication. See the AWS documentation for more information on the type of authentication available. If this value is omitted,

PROPERTY	DESCRIPTION
	and the properties <code>AWS_ACCESSKEY</code> and <code>AWS_SECRETKEY</code> are also omitted, the adapter assumes that you are using instance-based authentication.
<code>AWS_LINUX64_AUTO_SCALING_GROUP_NAME</code>	Existing auto scaling group name for 64-bit DataSynapse GridServer® Linux platform Engine.
<code>AWS_LINUX64_LAUNCH_TEMPLATE_NAME</code>	<code>LaunchTemplate</code> name used for creating auto scaling group for DataSynapse GridServer® Linux Engine.
<code>AWS_LINUX64_LAUNCH_TEMPLATE_VERSION</code>	<code>LaunchTemplate</code> version used for creating auto scaling group for DataSynapse GridServer® Linux Engine.
<code>AWS_WIN64_AUTO_SCALING_GROUP_NAME</code>	Existing auto scaling group name for 64-bit Windows platform DataSynapse GridServer®.
<code>AWS_WIN64_LAUNCH_TEMPLATE_NAME</code>	<code>LaunchTemplate</code> name used for creating auto scaling group for DataSynapse GridServer® Windows Engine.
<code>AWS_WIN64_LAUNCH_TEMPLATE_VERSION</code>	<code>LaunchTemplate</code> name used for creating auto scaling group for DataSynapse GridServer® Windows Engine.
<code>AWS_INSTANCE_TYPES</code>	Comma-separated list of instance types used when an auto scaling group for Engines is created. All instance types mentioned in <code>AWS_INSTANCE_TYPES</code> must have the number of cores equal to the <code>VM_CORES</code> property.

PROPERTY	DESCRIPTION
<code>AWS_ENGINE_AMI_PRODUCT_CODES</code>	Comma-separated list of product codes for Engines.

Azure-Specific Configuration Properties

The following configuration properties are specific to Azure-Platform:

PROPERTY	DESCRIPTION
<code>SUBSCRIPTIONID</code>	To find your subscription ID: 1. Log in to https://portal.azure.com. 2. From the left navigation pane, select Subscriptions. 3. In the Subscriptions window that opens, the subscription ID should be available (in GUID format) in the second column.
<code>ARM_CLIENTID</code> , <code>CLIENT_SECRET</code> , and <code>ARM_TENANT</code>	These values are specific to your account and the objects that you set up in it. The process for obtaining them can be complicated. For more information: • Refer to the Azure documentation or contact your Microsoft Azure representative for assistance with obtaining them. • Refer to Use an app identity to access Azure Stack Hub resources.
<code>RESOURCE_GROUP</code>	The resource group in which your VM scale is set is present. All the DataSynapse GridServer® Engine virtual machines are launched in this resource group.

PROPERTY	DESCRIPTION
<code>LINUX64_VM_SCALESET_NAME</code>	Azure scale set name for DataSynapse GridServer® Linux Engine.
<code>WIN64_VM_SCALESET_NAME</code>	Azure scale set name for DataSynapse GridServer® Windows Engine.

Google Cloud Platform (GCP) Specific Configuration Properties

The following configuration properties are specific to Google Cloud Platform (GCP):

PROPERTY	DESCRIPTION
<code>GCP_REGION</code>	The GCP region in which you have configured the instance group.
<code>MANAGED_LINUX64_INSTANCE_GROUP_NAME</code>	The instance group name for DataSynapse GridServer® Linux Engine.
<code>MANAGED_WIN64_INSTANCE_GROUP_NAME</code>	The instance group name for DataSynapse GridServer® Windows Engine.
<code>PROJECT_ID</code>	The Google Cloud Platform project ID.
<code>GOOGLE_APPLICATION_CREDENTIALS_FILE_PATH</code>	The path of the Google credentials file. This file is required if the <code>GOOGLE_APPLICATION_CREDENTIALS</code> environment variable is not set. For more information, see Creating and managing service account keys .