Install the AppDynamics Azure Site Extension for .NET

You can use the Windows Azure Portal to add the AppDynamics Azure Site Extension to your Azure App Service web app. Azure Site Extension is used by Ops teams that may not have access to source files, or would not like to modify or recompile them, yet still want to monitor their Azure projects and solutions.

Prepare to Install

To install the AppDynamics for Windows Azure Site Extension, you need the following:

- Connection information for your AppDynamics Controller. See also Agent and Controller Compatibility.
- A Windows Azure account.
- An Azure web app to monitor.

If you are upgrading from a previous version, see Upgrade the AppDynamics Azure Site Extension.

Add the AppDynamics Azure Site Extension

Add the AppDynamics Azure Site extension as you would any site extension for any Azure web app.

1. Log in to the Windows Azure Portal.
2. Browse to your web app.

   > If you want to configure the .NET Agent using environment variables, add the environment variables before you install the AppDynamics Azure Site Extension. See configure the agent using environment variables

3. From the **DEVELOPMENT TOOLS** list, click **Extensions**.
4. Click **+Add** to install the version of the **AppDynamics Azure Site Extension** you want to add to your web app.
   After you install the AppDynamics Azure Site Extension, it appears in the installed extensions list.

Add the AppDynamics Azure Site Extension Using an ARM Template

You can deploy the AppDynamics Azure Site Extension to Azure App Services using an Azure Resource Manager (ARM) template.

The following procedure uses Visual Studio Community 2017.

To create and deploy an ARM template:

1. From your web application in Visual Studio, choose **File > New Project**.
2. Click **Cloud**, then click **Azure Resource Group**, then click **OK**.
3. From the **Select Azure Template** dialog box, click **Web App**, then click **OK**.
4. Under your newly-created Resource Group, click the Website.json file.
5. Under JSON outline in the left pane, right-click **resources**, then choose **Add New Resource**.
6. In the Add Resource dialog box, select **Application Settings for Web Apps**, enter a name, then click **OK.**
7. Under the properties section, enter your AppDynamics Controller information. The following example shows the Website.json file with the application properties for AppDynamics.
8. Add a new apiVersion section with your Azure Site Extension details, as shown in the following example.

```
{
     "apiVersion": "2015-08-01",
     "name": "[variables('webSiteName')]",
     "type": "Microsoft.Web/sites",
     "location": "[resourceGroup().location]",
     "tags": {
       "[concat('hidden-related:', resourceGroup().id, '/providers/Microsoft.Web/serverfarms/', parameters('hostingPlanNam
       "displayName": "Website"
      },
      "dependsOn": [
        "[resourceId('Microsoft.Web/serverfarms/', parameters('hostingPlanName'))]"
      ],
      "properties": {
        "name": "[variables('webSiteName')]",
        "serverFarmId": "[resourceId('Microsoft.Web/serverfarms', parameters('hostingPlanName'))]"
      },
      "resources": [
        {
          "apiVersion": "2016-08-01",
          "name": "appsettings",
          "type": "config",
          "dependsOn": [
            "[resourceId('Microsoft.Web/Sites', variables('webSiteName'))]"
           ],
          "properties": {
            "appdynamics.controller.hostName": "<host_name>",
            "appdynamics.controller.port": "<port_name>",
            "appdynamics.controller.ssl.enabled": "true",
            "appdynamics.agent.accountName": "<account_name>",
            "appdynamics.agent.accountAccessKey": "[paraeters(<InKeyVault>)]",
            "appdynamics.agent.applicationName": "<application_name>",
            "appdynamics.agent.tierName": "<tier_name>",
            "appdynamics.agent.nodeName": "node_name>"
          }
        },
        {
          "apiVersion": "2015-08-01",
          "name": "AppDynamics.WindowsAzure.SiteExtension.4.5.release",
          "type": "siteextensions".
          "dependsOn":
            "[resourceId('Microsoft.Web/Sites', variables('webSiteName'))]"
          ]
        },
```

Here is the sample text for copying or pasting:

```
{
      "apiVersion": "2015-08-01",
      "name": "[variables('webSiteName')]",
      "type": "Microsoft.Web/sites",
      "location": "[resourceGroup().location]",
      "tags": {
        "[concat('hidden-related:', resourceGroup().id, '/providers/Microsoft.Web/serverfarms/',
parameters('hostingPlanName'))]": "Resource",
        "displayName": "Website"
      },
      "dependsOn": [
        "[resourceId('Microsoft.Web/serverfarms/', parameters('hostingPlanName'))]"
      ],
      "properties": {
        "name": "[variables('webSiteName')]",
        "serverFarmId": "[resourceId('Microsoft.Web/serverfarms', parameters('hostingPlanName'))]"
      },
      "resources": [
        {
          "apiVersion": "2016-08-01",
          "name": "appsettings",
          "type": "config",
          "dependsOn": [
            "[resourceId('Microsoft.Web/Sites', variables('webSiteName'))]"
          ],
          "properties": {
            "appdynamics.controller.hostName": "leargasstage.saas.appdynamics.com",
            "appdynamics.controller.port": "443",
            "appdynamics.controller.ssl.enabled": "true",
            "appdynamics.agent.accountName": "leargasstage",
            "appdynamics.agent.accountAccessKey": "[parameters('danielAppDAccessKeyInKeyVault')]",
            "appdynamics.agent.applicationName": "HelloWorldSecureAppDKey",
            "appdynamics.agent.tierName": "TestTier",
            "appdynamics.agent.nodeName": "TestNode"
          }
        },
        {
          "apiVersion": "2015-08-01",
          "name": "AppDynamics.WindowsAzure.SiteExtension.4.5.Release",
          "type": "siteextensions",
          "dependsOn": [
            "[resourceId('Microsoft.Web/Sites', variables('webSiteName'))]"
          ]
        },
```

Configure the Controller Connection

You have the following options to configure the .NET Agent to connect to the AppDynamics Controller:

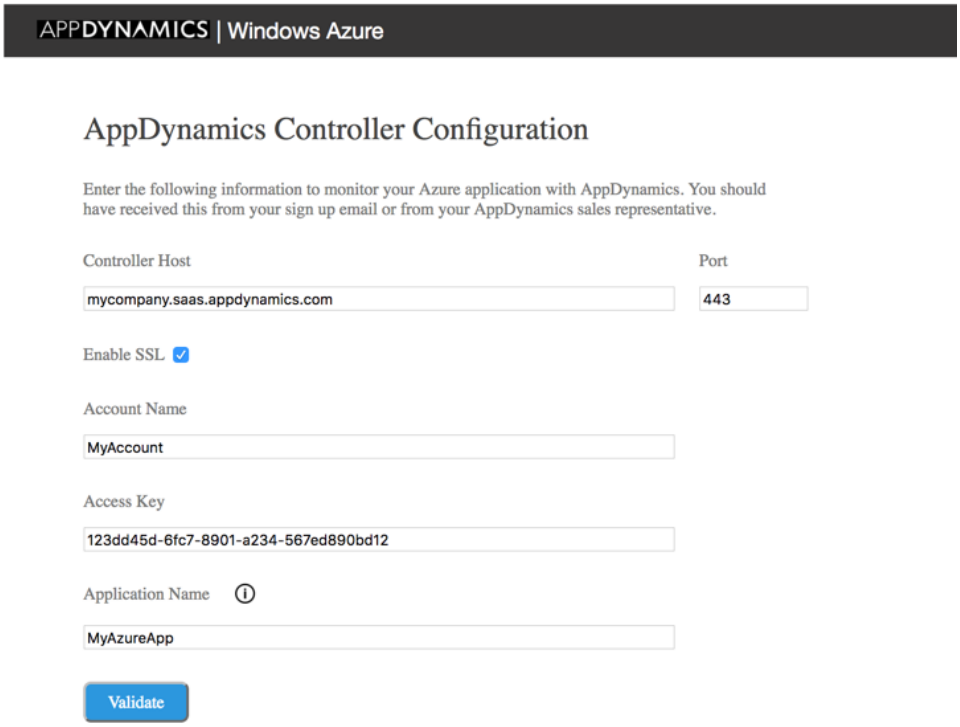
- Configure the Controller connection using the Kudu console.
- Configure the Controller connection settings using environment variables.

For more detail, see Agent-to-Controller Connections.

Configure the Agent with the Kudu Console

When you add the AppDynamics Azure Site Extension to your web app, you can interactively configure the .NET Agent using the Kudu console.

1. Navigate to the AppDynamics Controller Configuration page in the Kudu Console:
   http://{web app}.scm.azurewebsites.net/appdynamics/
   For example: https://myazureexample.scm.azurewebsites.net/appdynamics/
2. On the AppDynamics Controller Configuration page, enter your Controller connection information. For example:

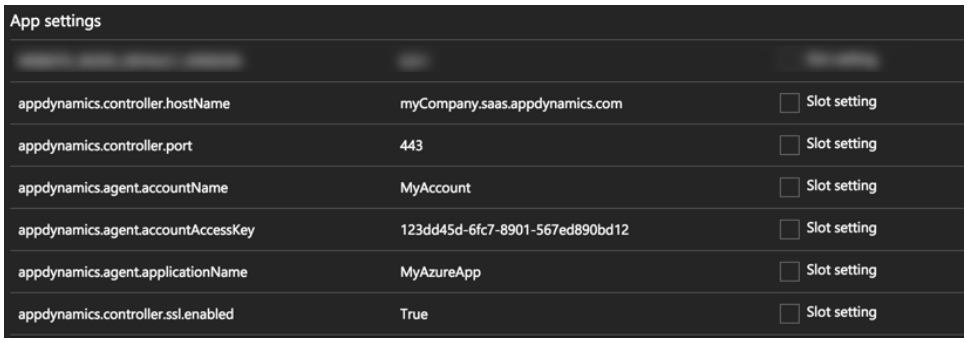


3. Click **Validate** to test the connection to the AppDynamics Controller and save your settings.
4. Restart your web app.
   After you apply some load to your web app, you can view it on flow maps in the AppDynamics Controller UI.

Configure the Agent Using Environment Variables

Configuring the .NET Agent using environment variables allows for unattended configuration. To configure agents in this manner, add the environment variables before you install the AppDynamics Azure Site Extension, as follows:

1. Navigate to **SETTINGS > Application Settings** for your web app.
2. Add the .NET Agent environment variables under **App settings:**
   - appdynamics.controller.hostName: the address for the AppDynamics Controller
   - appdynamics.controller.port: the Controller port
   - appdynamics.agent.accountName: the account name you use to log on to the Controller
   - appdynamics.agent.accountAccessKey: the account key you use to log on the Controller
   - appdynamics.agent.applicationName: the business application name in the Controller
   - appdynamics.controller.ssl.enabled: set to "True" to enable SSL connection to the Controller. Otherwise set to "False".
     For example:



3. Restart your web app. After you apply some load to your web app, you can view it on flow maps in the AppDynamics Controller UI.

Specify Which WebJobs to Monitor

Use the `APPDYNAMICS.PROCESSLIST` environment variable to specify which WebJobs the .NET Agent monitors.

Environment Variable: `APPDYNAMICS.PROCESSLIST`
Type: Strings separated by "|"
Default: None. When undefined, the agent instruments all WebJob running in the app service. When defined, the .NET Agent instruments only the specified processes.

Required: No

Usage Cases:

- Environment 1: An Azure app service with no WebJobs, where the Appdynamics .NET Agent site extension is installed and configured. After defining the environment variable `AppDynamics.ProcessList` with the value `w3wp.exe`, the `Daasrunner.exe` WebJob, which is a default WebJob that is added to all app services, is not instrumented.
- Environment 2: An Azure app service, with two WebJobs, `webjob1.exe` and `webjob2.exe`, where the Appdynamics .NET Agent site extension is installed and configured. After defining the environment variable `AppDynamics.ProcessList` with the value `w3wp.exe`, only the worker process serving the actual web app will be instrumented and no WebJobs will be instrumented. With the environment variable value set to `w3wp.exe|webjob1.exe`, the worker process serving the web app and the webjob1.exe are instrumented, but the webjob2.exe is not instrumented. With the `AppDynamics.ProcessList` removed from the App settings of the Azure app service, all the WebJobs, including `Daasrunner.exe` are are instrumented, including the worker process serving the web app.

Upgrade the AppDynamics Azure Site Extension

When you click on the **Extensions** tab for your web app, the Microsoft Azure Portal displays the currently installed version of the AppDynamics Azure Site Extension. The Update Available column of the installed extensions list indicates if there is a more recent minor release of the .NET Agent available. If so, you can click to update the extension from the list.

Upgrade a Major Version of the .NET Agent

AppDynamics maintains major release versions of the .NET Agent as separate site extensions. Therefore you need to uninstall the installed version of the AppDynamics Azure Site Extension before you upgrade to a new major release:

1. Log in to the Windows Azure Portal.
2. Stop your web app.
3. Click on the AppDynamics Azure Site Extension from the list of installed extensions and click **Delete** to uninstall it.
4. Install the new version of the AppDynamics Azure Site Extension as normal.

If you are upgrading from version 4.2 of the .NET agent and you used environment variable configuration, you must update your web app environment variables. See configure the agent using environment variables. Note that the `APPD_UNATTENDED` variable is no longer required.