Contents

- Azure Government Documentation
- Overview
  - What is Azure Government?
  - Available services
  - Developer guide
- Get Started
  - Virtual Machines
  - Azure App Services
  - Cognitive Services
  - Integrate Azure AD Authentication
  - GPUs
  - Kubernetes
- How To
  - Plan
    - Security
    - Compliance
    - Identity
  - Manage
    - Log Analytics
    - Subscription
    - Marketplace
  - Connect
    - Log in to the Azure Government portal
    - Connect with PowerShell
    - Connect with CLI
    - Connect to Azure Government from Visual Studio
    - Connect to Azure Government from Visual Studio Team Services
    - Connect to Azure Storage
    - Connect to Azure Government from SSMS

- Reference
  - Marketplace
    - Images
    - Extensions
    - Marketplace for partners
  - Services
    - Compute
    - Networking
    - Storage
    - Web + Mobile
    - Azure Stack
    - Media Services
    - Databases
    - Data + Analytics
    - AI + Cognitive Services
    - Internet of Things
    - Integration Services
    - Security + Identity
    - Backup
    - Monitoring + Management
    - Developer Tools
  - CSP for Azure Government
  - Certifications
    - ITAR
    - Justice and Public Safety
    - Department of Defense
- Resources
  - Azure Government Website
  - Azure Roadmap
  - Blog
  - Pricing
  - Pricing calculator

Trial

Microsoft Azure Government delivers a cloud platform built upon the foundational principles of security, privacy and control, compliance, and transparency. Public Sector entities receive a physically isolated instance of Microsoft Azure that employs world-class security and compliance services critical to U.S. government for all systems and applications built on its architecture.

Learn about Azure Government

Azure Government Video Library

Reference

Images

Marketplace

Marketplace for partners

Services

Compute

Networking

Storage

Web + Mobile

Databases

Data + Analytics

AI + Cognitive Services

Internet of Things

Security + Identity

Monitoring + Management

Welcome to Azure Government

6/27/2017 • 3 minutes to read • Edit Online

Overview

Microsoft Azure Government delivers a cloud platform built upon the foundational principles of security, privacy & control, compliance, and transparency. Public Sector entities receive a physically isolated instance of Microsoft Azure that employs world-class security and compliance services critical to U.S. government for all systems and applications built on its architecture. These services include FedRAMP and DoD compliance certifications, CJIS state-level agreements, the ability to issue HIPAA Business Associate Agreements, and support for IRS 1075. Operated by screened U.S. persons, Azure Government supports multiple hybrid scenarios for building and deploying solutions on-premises or in the cloud. Public Sector entities can also take advantage of the instant scalability and guaranteed uptime of a hyper-scale cloud service.

Azure Government includes the core components of Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS). This includes infrastructure, network, storage, data management, identity management, and many other services.

Azure Government supports most of the same great features that public Azure customers have leveraged like Geo-Synchronous data replication and auto scaling.

- See the regions page for the most up-to-date services that are generally available.
- Sign up for a trial

U.S. government organizations interested in cloud services can be confident that Azure Government provides enormous scale and rigorous security practices to meet their evolving needs.

Azure Government Documentation

This site describes the capabilities of Microsoft Azure Government services, and provides general guidance applicable to all customers. Before including specifically regulated data in your Azure Government subscription, you should familiarize yourself with the Azure Government capabilities and consult your account team if you have any questions.

You should refer to the Microsoft Azure Trust Center Compliance Page for current information on the Azure Government services covered under specific accreditations and regulations. Additional Microsoft services might also be available, but are not within the scope of the Azure Government covered services and are not addressed by this document. Azure Government services might also permit you to use various additional resources, applications, or services that are provided by third parties—or by Microsoft under separate terms of use and privacy policies—which are not included in the scope of this document. You are responsible for reviewing the terms of all such "add-on" offerings, such as Marketplace offerings, to ensure that they meet your needs regarding compliance.

Azure Government is available to entities that handle data that is subject to certain government regulations and requirements (such as NIST 800.171 (DIB), ITAR, IRS 1075, DoD L4, and CJIS) where use of Azure Government is required to comply with regulations. Azure Government customers are subject to validation of eligibility.

Entities with questions about eligibility for Azure Government should consult their account team.

General Guidance for Customers

Most of the technical content that is available currently assumes that applications are being developed for the Global Service rather than for Microsoft Azure Government, it's important for you to ensure that developers are

aware of key differences for applications developed to be hosted in Azure Government.

- First, there are services and feature differences, this means that certain features that are in specific regions of the Global Service may not be available in Azure Government.
- Second, for features that are offered in Azure Government, there are configuration differences from the Global Service. Therefore, you should review your sample code, configurations, and steps to ensure that you are building and executing within the Azure Government Cloud Services environment.
- Third, you should refer to the Azure Government Technical services documentation available from this site for information that identifies the Azure Government boundary and customer regulated/controlled data guidance and best practices.

Next Steps

For supplemental information and updates, please subscribe to the Microsoft Azure Government Blog.

If you are interested in learning more and about Azure Government please use some of the links below.

- **Sign up for a trial**
- **Acquiring and accessing Azure Government**
- **Microsoft Trust Center**

Available services in Azure Government

7/12/2018 • 2 minutes to read • Edit Online

Azure Government is continually expanding its services. These services are deployed through the same code that is used in Azure public. This section documents the services that are currently available in Azure Government, including two key types of information:

- **Variations**: Variations due to features that are not deployed yet or properties (for example, URLs) that are unique to the government environment.
- **Considerations**: Government-specific implementation detail to ensure that data stays within your compliance boundary.

For the most current list of services, see the Products available by region page.

The **services available in Azure Government** are listed by category, as well as whether they are generally available or available through preview.

> **NOTE**
> ** = Service can be accessed through PowerShell and CLI, but not yet available through the Azure Government portal.

Compute

GENERALLY AVAILABLE	PREVIEW
Virtual Machines	
Batch	
Cloud Services	
Virtual Machine Scale Sets	
Functions	
Service Fabric	

Networking

GENERALLY AVAILABLE	PREVIEW
ExpressRoute	
Virtual Network	
Load Balancer	
DNS	

GENERALLY AVAILABLE	PREVIEW
Traffic Manager	
VPN Gateway	
Application Gateway	
Network Watcher	

Storage

GENERALLY AVAILABLE	PREVIEW
Blob storage	
Table storage	
Queue storage	
File storage	
Disk storage	
StorSimple	
Import/Export	

Web + Mobile

GENERALLY AVAILABLE	PREVIEW
App Service: Web Apps	
App Service: Mobile Apps	
API Management	
Media Services	

Databases

GENERALLY AVAILABLE	PREVIEW
SQL Database	
SQL Data Warehouse	
SQL Server Stretch Database	

GENERALLY AVAILABLE	PREVIEW
Azure Cosmos DB	
Azure Redis Cache	

Data + Analytics

GENERALLY AVAILABLE	PREVIEW
HDInsight	
Power BI Pro**	
Azure Analysis Services	

AI + Cognitive Services

GENERALLY AVAILABLE	PREVIEW
	Cognitive Services**

Internet of Things

GENERALLY AVAILABLE	PREVIEW
IoT Hub	
Azure Event Hubs	
Azure Notification Hubs**	

Enterprise Integration

GENERALLY AVAILABLE	PREVIEW
Logic Apps	
Service Bus	
StorSimple	
SQL Server Stretch Database	

Security + Identity

GENERALLY AVAILABLE	PREVIEW
	Azure Security Center

GENERALLY AVAILABLE	PREVIEW
Azure Active Directory	
Azure Active Directory Premium	
Key Vault	
Azure Multi-Factor Authentication	

Monitoring + Management

GENERALLY AVAILABLE	PREVIEW
Automation	Advisor
Backup	
Policy	
Log Analytics	
Site Recovery	
Scheduler	
Monitoring and Diagnostics	
Azure Portal	
Azure Resource Manager	

Developer Tools

GENERALLY AVAILABLE	PREVIEW
DevTest Labs	

Next steps

For supplemental information and updates, subscribe to the Microsoft Azure Government blog.

Azure Government developer guide

6/29/2018 • 2 minutes to read • Edit Online

Azure Government is a separate instance of the Microsoft Azure service. It addresses the security and compliance needs of United States federal agencies, state and local governments, and their solution providers. Azure Government offers physical isolation from non-US government deployments and provides screened US personnel.

Microsoft provides various tools to help developers create and deploy cloud applications to the global Microsoft Azure service ("global service") and Microsoft Azure Government services.

When developers create and deploy applications to Azure Government services, as opposed to the global service, they need to know the key differences between the two services. The specific areas to understand are:

- Setting up and configuring their programming environment
- Configuring endpoints
- Writing applications
- Deploying applications as services to Azure Government

The information in this document summarizes the differences between the two services. It supplements the information that's available through the following sources:

- Azure Government site
- Microsoft Azure Technical Library on MSDN
- Microsoft Azure Trust Center
- Azure Documentation Center
- Azure Blogs

This content is intended for partners and developers who are deploying to Microsoft Azure Government.

Guidance for developers

Most of the currently available technical content assumes that applications are being developed for the global service rather than for Azure Government. For this reason, it's important to be aware of two key differences in applications that you develop for hosting in Azure Government.

- Certain services and features that are in specific regions of the global service might not be available in Azure Government.
- Feature configurations in Azure Government might differ from those in the global service.
    - Therefore, it's important to review your sample code, configurations, and steps to ensure that you are building and executing within the Azure Government Cloud Services environment.

Currently, US DoD East, US DoD Central, US Gov Virginia, US Gov Arizona, US Gov Texas and US Gov Iowa are the datacenters that support Azure Government. For current datacenters and available services, see Products available by region.

Quickstarts

Navigate through the links below to get started using Azure Government.

- Login to Azure Government Portal
- Connect with Visual Studio

- Connect with PowerShell
- Connect with CLI
- Connect to Azure Storage

Azure Government Video Library

The Azure Government video library contains many helpful videos to get you up and running with Azure Government.

Compliance - Azure Blueprint

The Azure Blueprint program is designed to facilitate the secure and compliant use of Azure for government agencies and third-party providers building on behalf of government.

For more information on Azure Government Compliance, refer to the compliance documentation and watch this video.

Endpoint mapping

To learn about mapping global Azure and SQL Database endpoints to Azure Government-specific endpoints, see the following table:

> **NOTE**
> The **Active Directory Authority** for Azure Government has changed from https://login-us.microsoftonline.com to https://login.microsoftonline.us. The original URL will continue to work but all applications should be updated to the new authority URL.

NAME	AZURE GOVERNMENT ENDPOINT	AZURE COMMERCIAL ENDPOINT
Portal	https://portal.azure.us	https://portal.azure.com
Active Directory Endpoint and Authority	https://login.microsoftonline.us	https://login.microsoftonline.com
Active Directory Graph API	https://graph.windows.net/	https://graph.windows.net/
Microsoft Graph API	https://graph.microsoft.com/	https://graph.microsoft.com/
Azure API	https://management.usgovcloudapi.net/	https://management.azure.com/
SQL Database DNS Suffix	*.database.usgovcloudapi.net	*.database.windows.net
Storage DNS Suffix	*.core.usgovcloudapi.net	*.core.windows.net
Traffic Manager DNS Suffix	*.usgovtrafficmanager.net	*.trafficmanager.net
Key Vault DNS Suffix	*.vault.usgovcloudapi.net	*.vault.azure.net
Service Bus DNS Suffix	*.servicebus.usgovcloudapi.net	*.servicebus.windows.net
Gallery Url	https://gallery.azure.us/	https://gallery.azure.com/

NAME	AZURE GOVERNMENT ENDPOINT	AZURE COMMERCIAL ENDPOINT
Classic Deployment Model Url	https://management.core.usgovcloudapi.net/	https://management.core.windows.net/
Publish Settings File Url	https://portal.azure.us#blade/Microsoft_Azure_ClassicResources/PublishingProfileBlade	https://portal.azure.com/#blade/Microsoft_Azure_ClassicResources/PublishingProfileBlade

Next steps

For more information about Azure Government, see the following resources:

- Sign up for a trial
- Acquiring and accessing Azure Government
- Ask questions via the azure-gov tag in StackOverflow
- Azure Government Overview
- Azure Government Blog
- Azure Compliance

Virtual Machines on Azure Government

10/23/2017 • 2 minutes to read • Edit Online

This quickstart will help you get started using Virtual Machines on Azure Government. Using VMs with Azure Government is similar to using it with the Azure commercial platform, with a few exceptions.

To learn more about Azure Virtual Machines, click here.

Part 1: Virtual Network

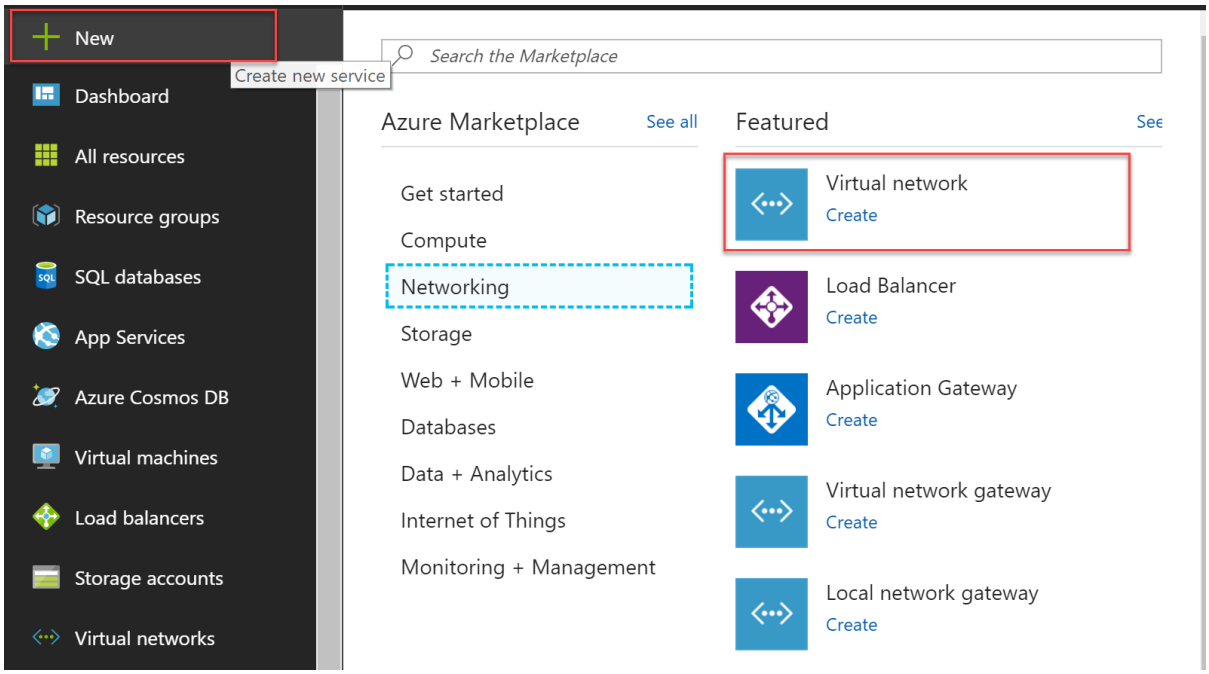
Prerequisites

Before completing this section, you must have:

- An active Azure Government subscription. If you don't have an Azure Government subscription, create a free account before you begin.

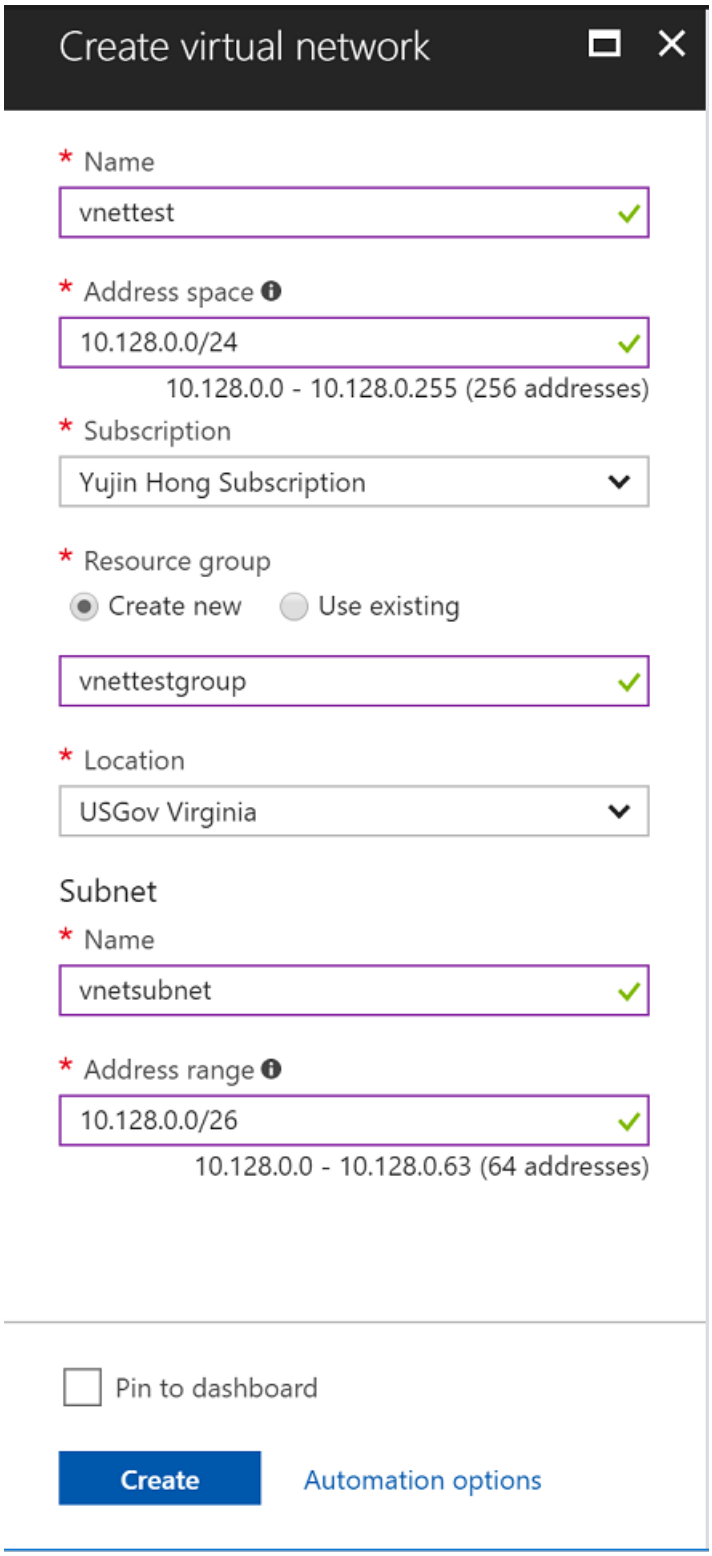
Create a new Virtual Network

1. Navigate to the Azure Government portal and login with your Azure Government credentials.
2. Click on the green + New in the upper left blade and click on Networking | Virtual Network.

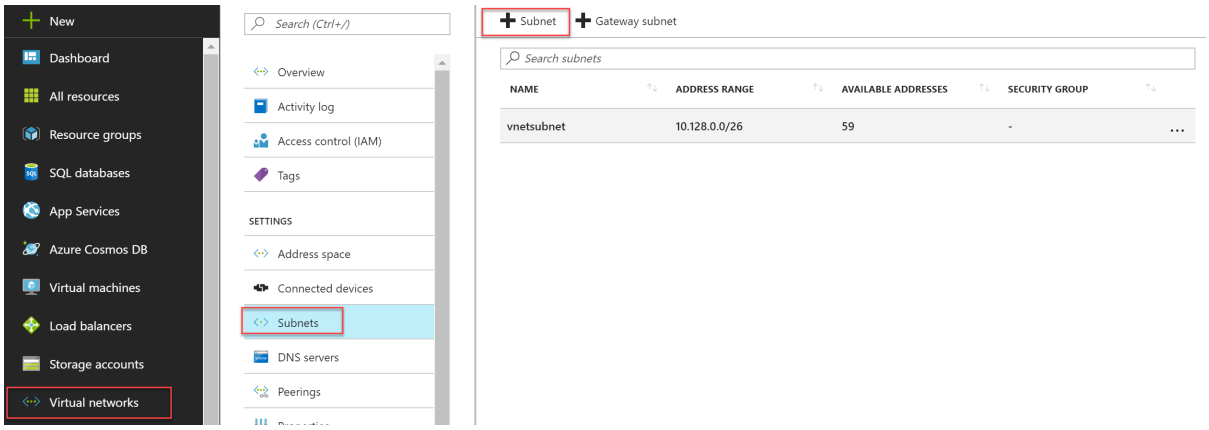


3. Make sure the deployment model is set to "Resource Manager" and click Create.
4. Fill out the following fields and click Create.

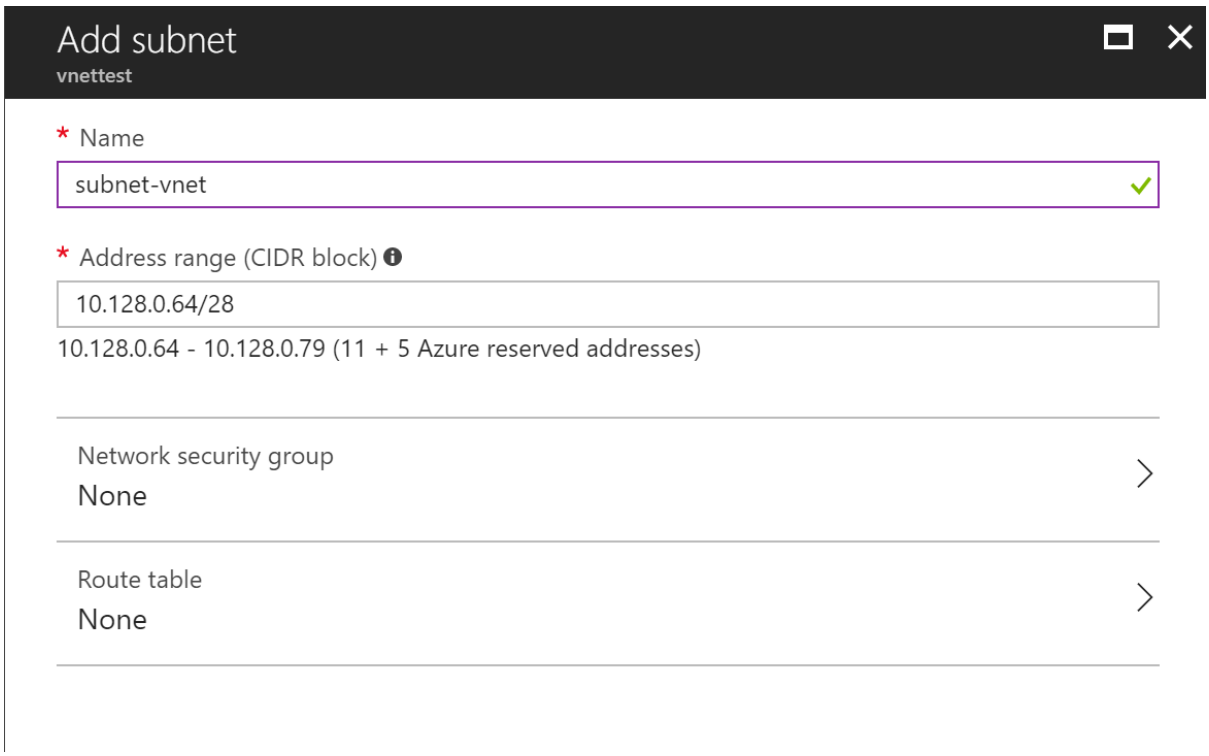
> **NOTE**
> Your Subscription box will look different from below.



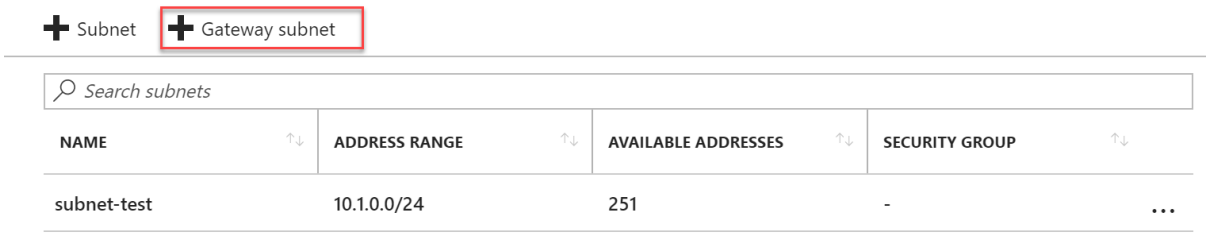
5. Navigate to "Virtual networks" from the menu on the left and click on the Virtual Network you just created. Under "Settings" click on "Subnets".



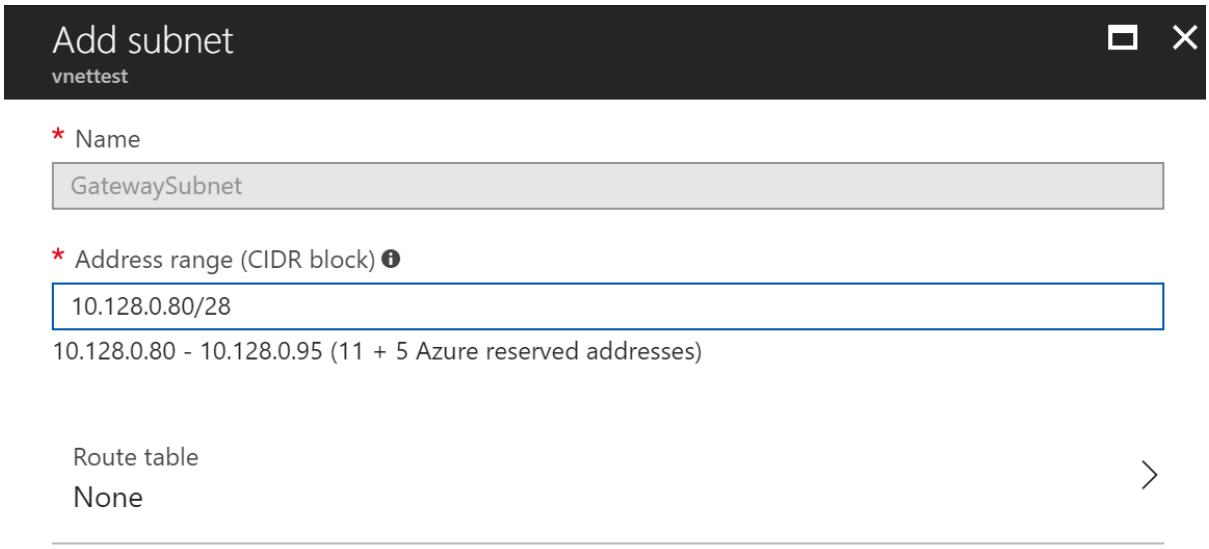
6. On the top left-hand corner of the page choose "Subnet" and fill out the following fields.



7. Click "Ok" when finished and navigate to the top left hand corner again. Click on "Gateway Subnet".



8. Enter the address range shown below ad click "Ok". You have now created a Virtual Network on Azure Government.



Part 2: Virtual Machine

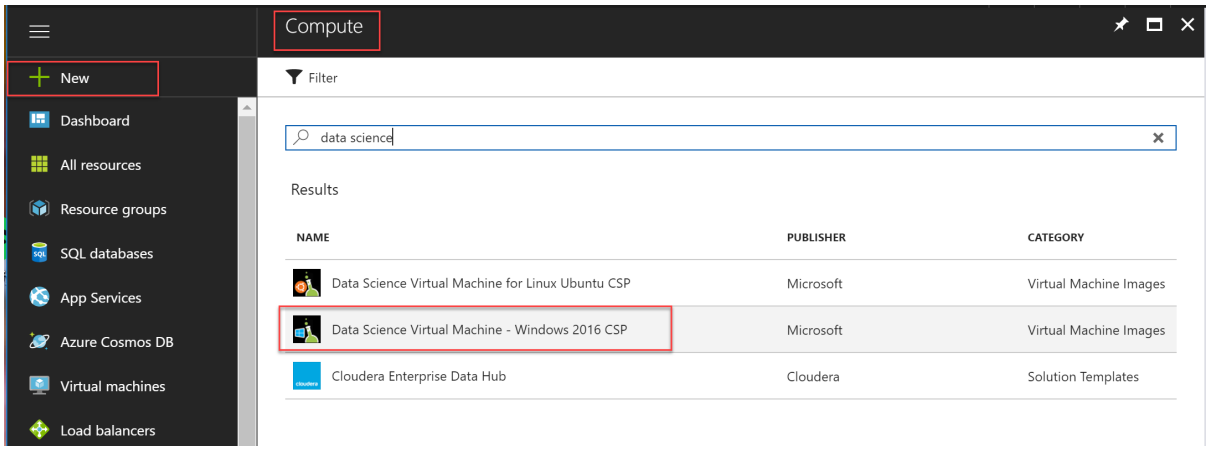
Prerequisites

Before completing this section you must have:

- An active Azure Government subscription. If you don't have an Azure Government subscription, create a free account before you begin.
- A Virtual Network running on Azure Government. If you don't already have a Virtual Network, complete the "Create a new Virtual Network" section above.

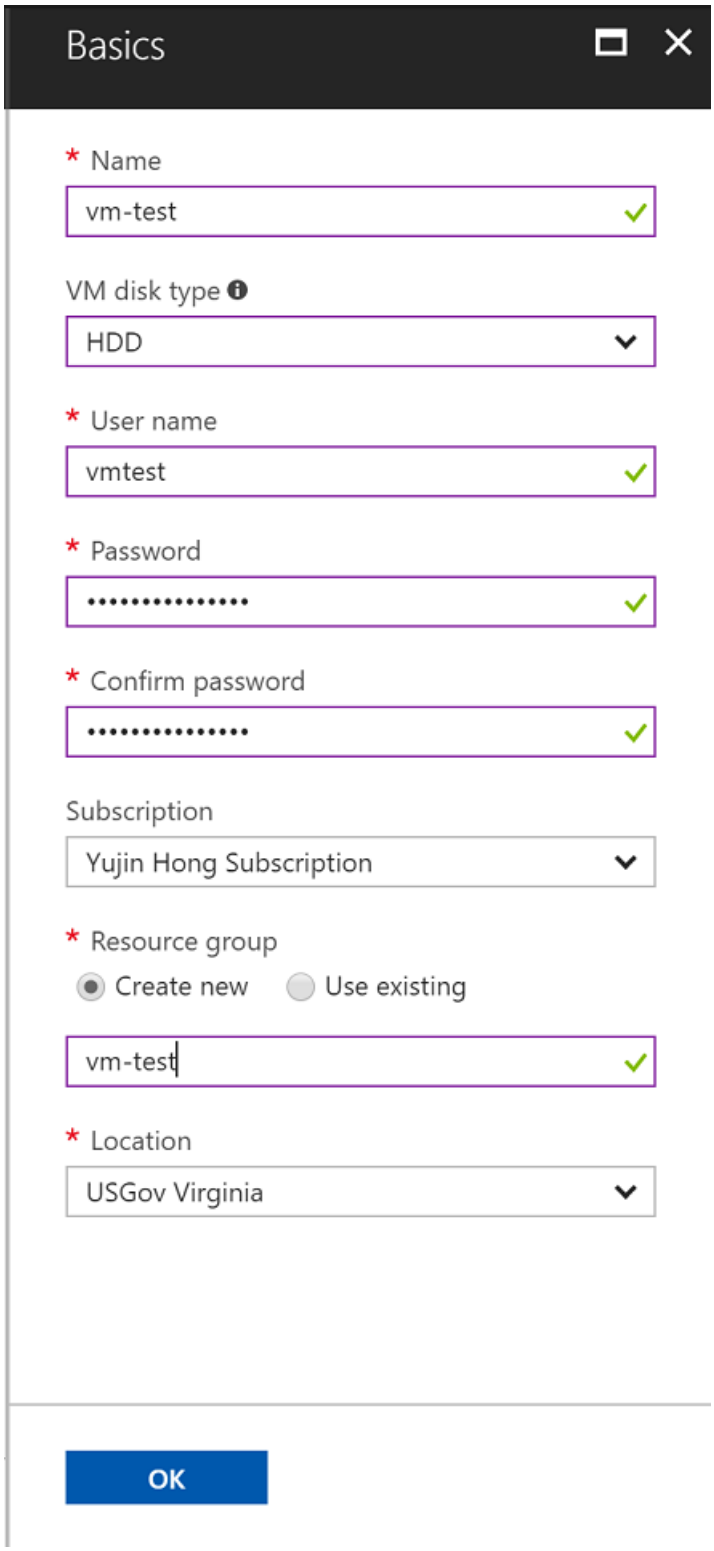
Create a new Virtual Machine

1. Navigate to the Azure Government portal and login with your Azure Government credentials.
2. Click on the green + New in the upper left corner and click on "Compute".
3. Search for "Data science" and then click on "Data Science Virtual Machine - Windows 2016 CSP".

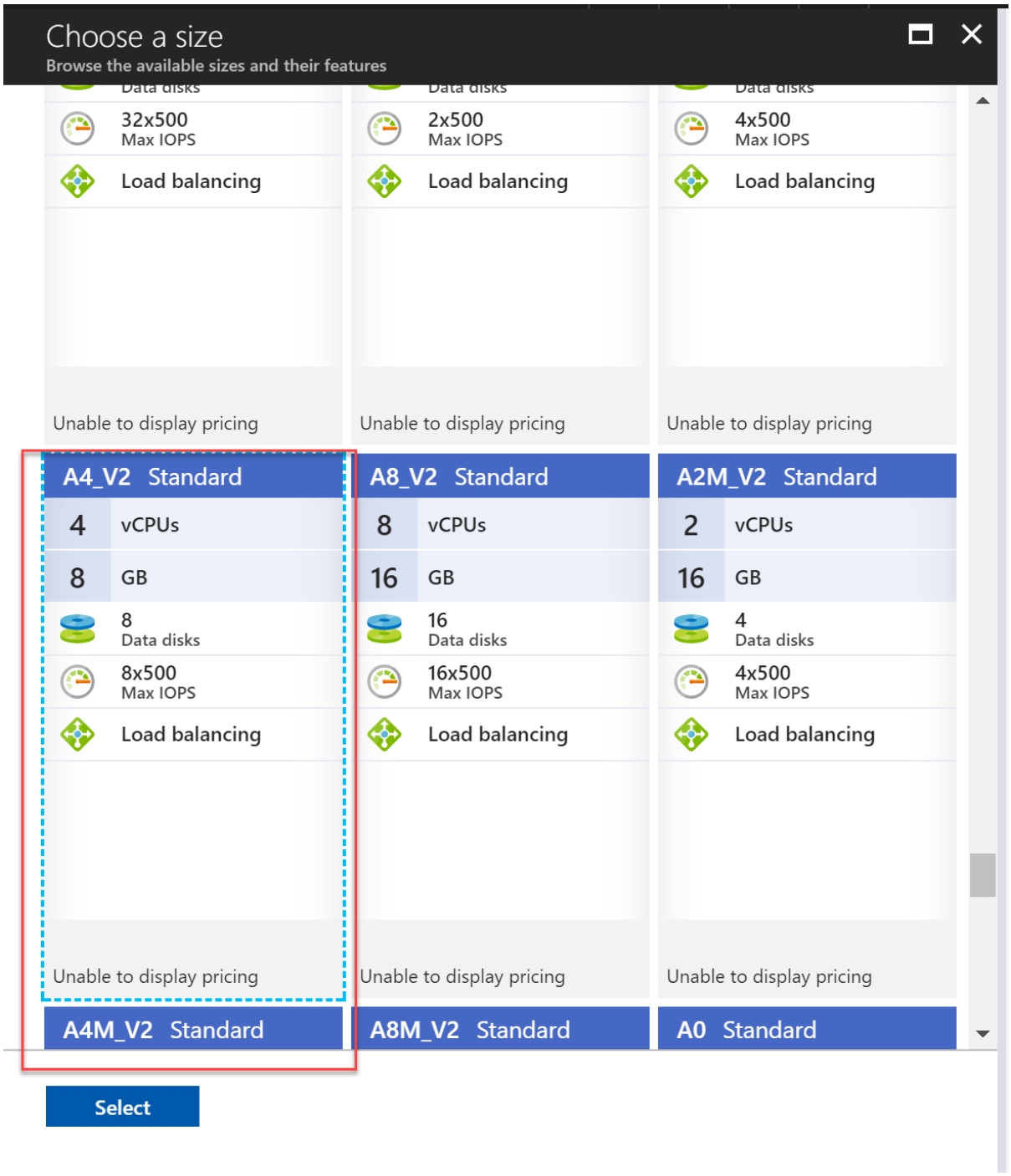


4. Click on "Create". Then fill out the fields and click "Ok".

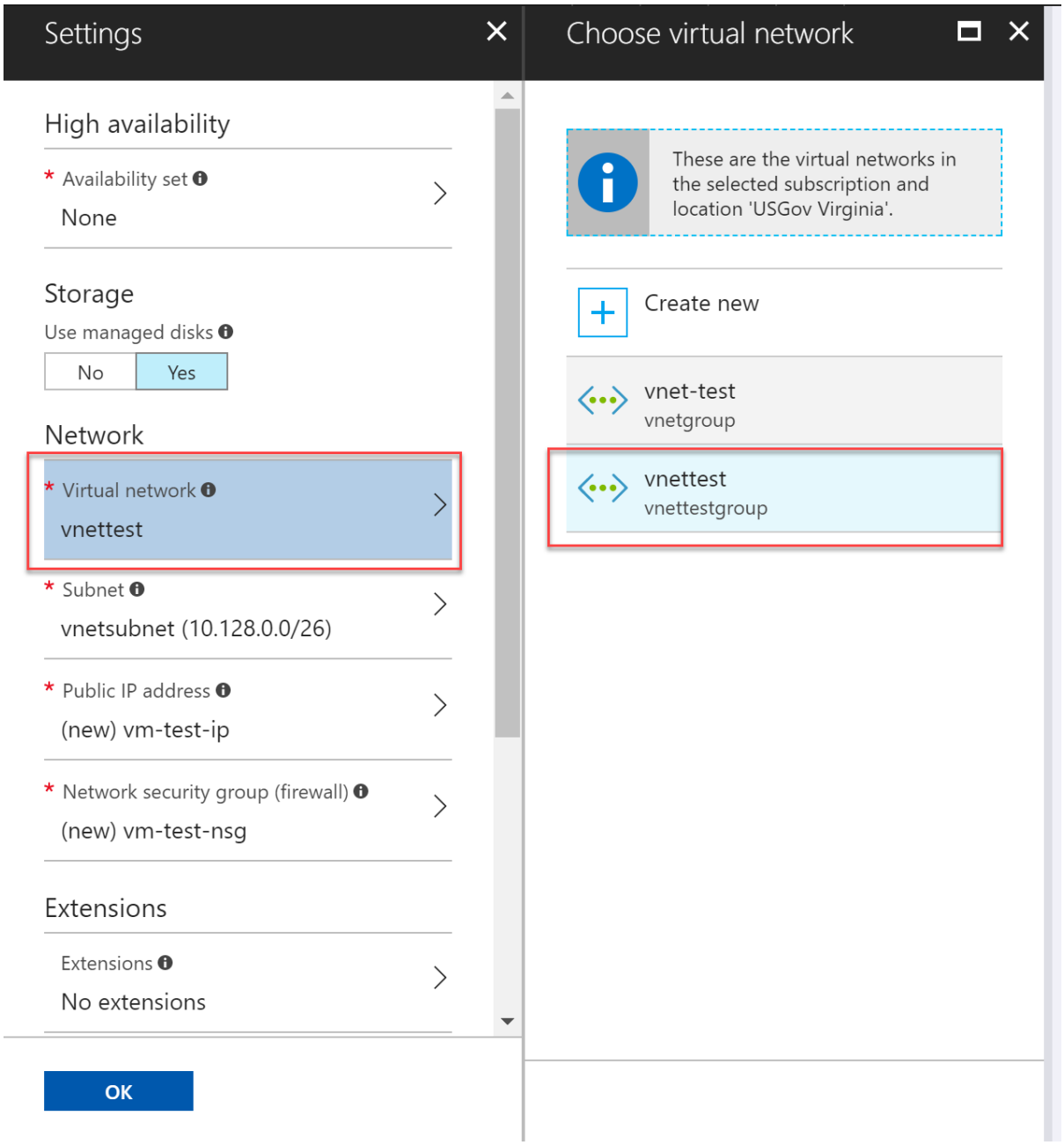
> **NOTE**
> Choose a password that you will remember!



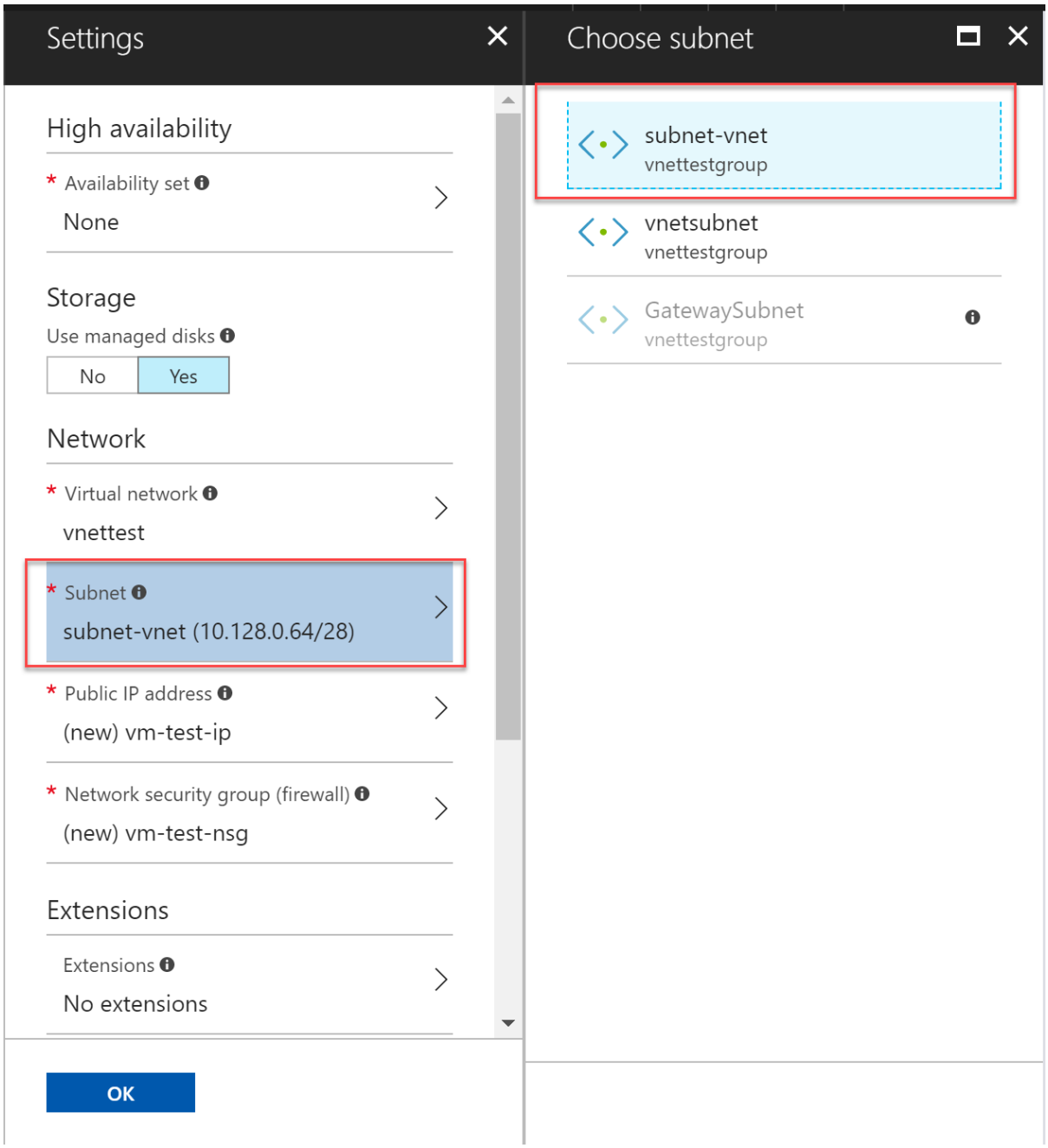
5. Open the Supported disk type dropdown box and select HDD. Click on "View All" in the options at the top right corner. Scroll down the A4_v2 size and select it. Click on Select.



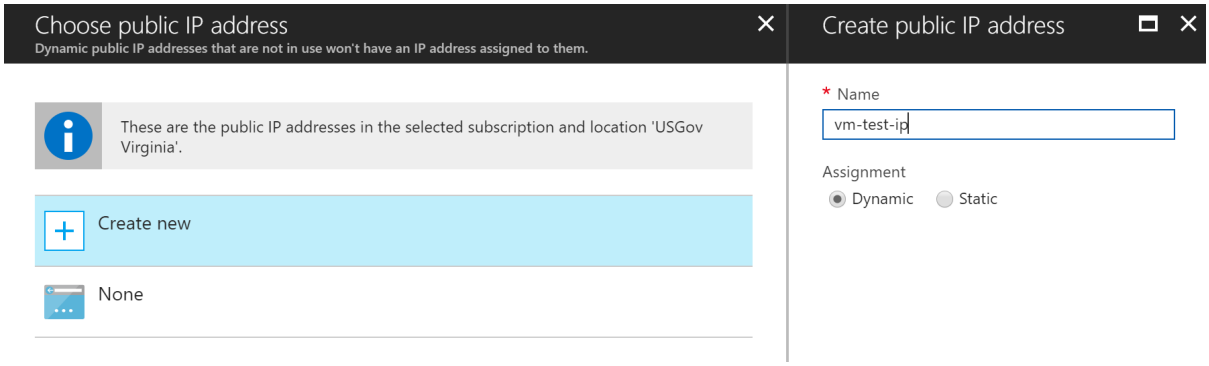
6. On the left hand "Settings" box click on "Network" and select your Virtual Network.



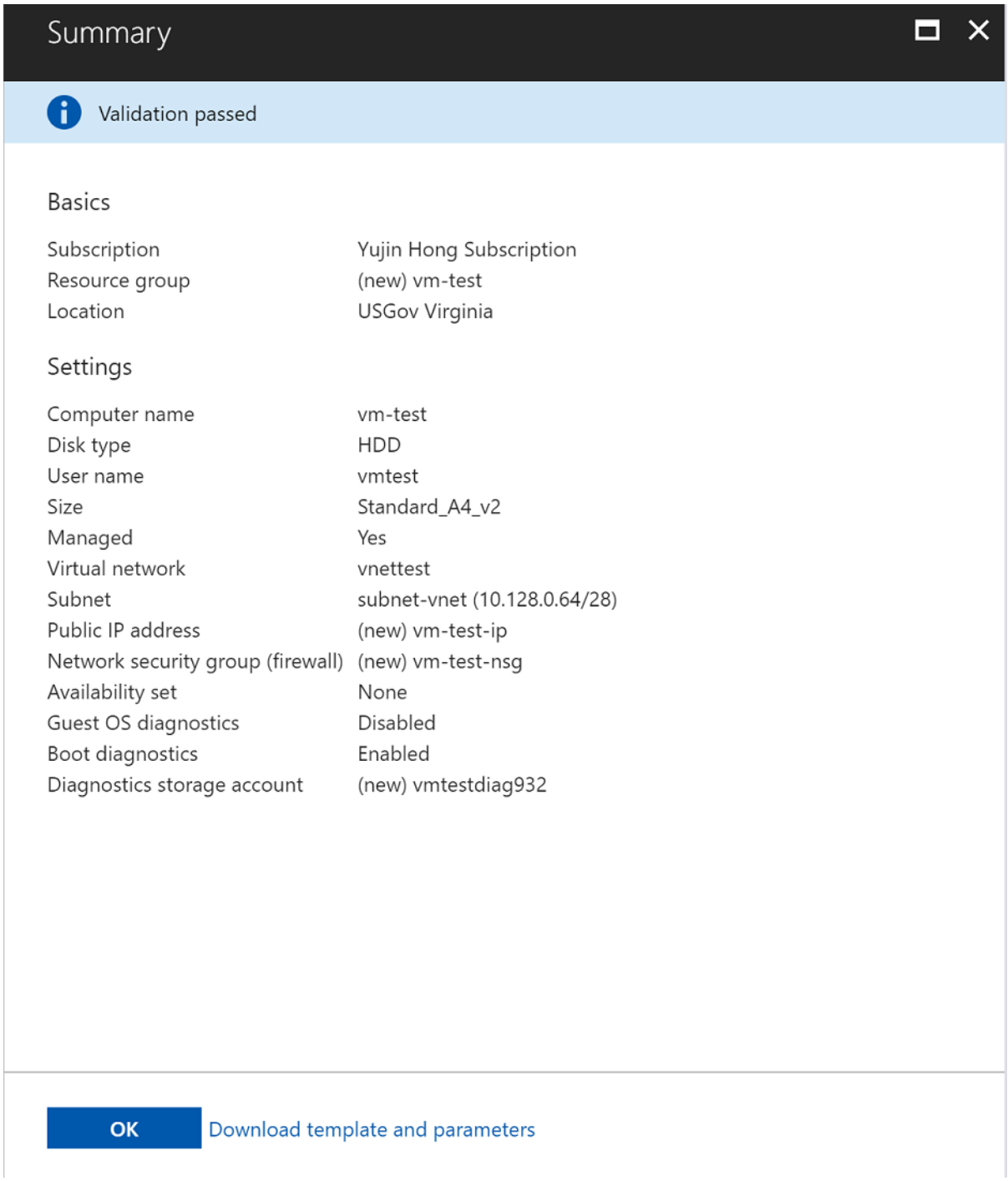
7. Click on "Subnet" and choose the subnet that you just created.



8. Click on "Public IP address" and then click on "Ok".



9. Now we can create the VM by clicking "Ok".
10. Once the validation step has completed click "Ok" and you should see the following screen.



The VM will now be provisioned. It will take several minutes to complete, but afterwards you will be able to connect to the VM with RDP using the public IP address.

Next steps

For supplemental information and updates, subscribe to the Microsoft Azure Government Blog.

Azure App Services on Azure Government

2/28/2018 • 3 minutes to read • Edit Online

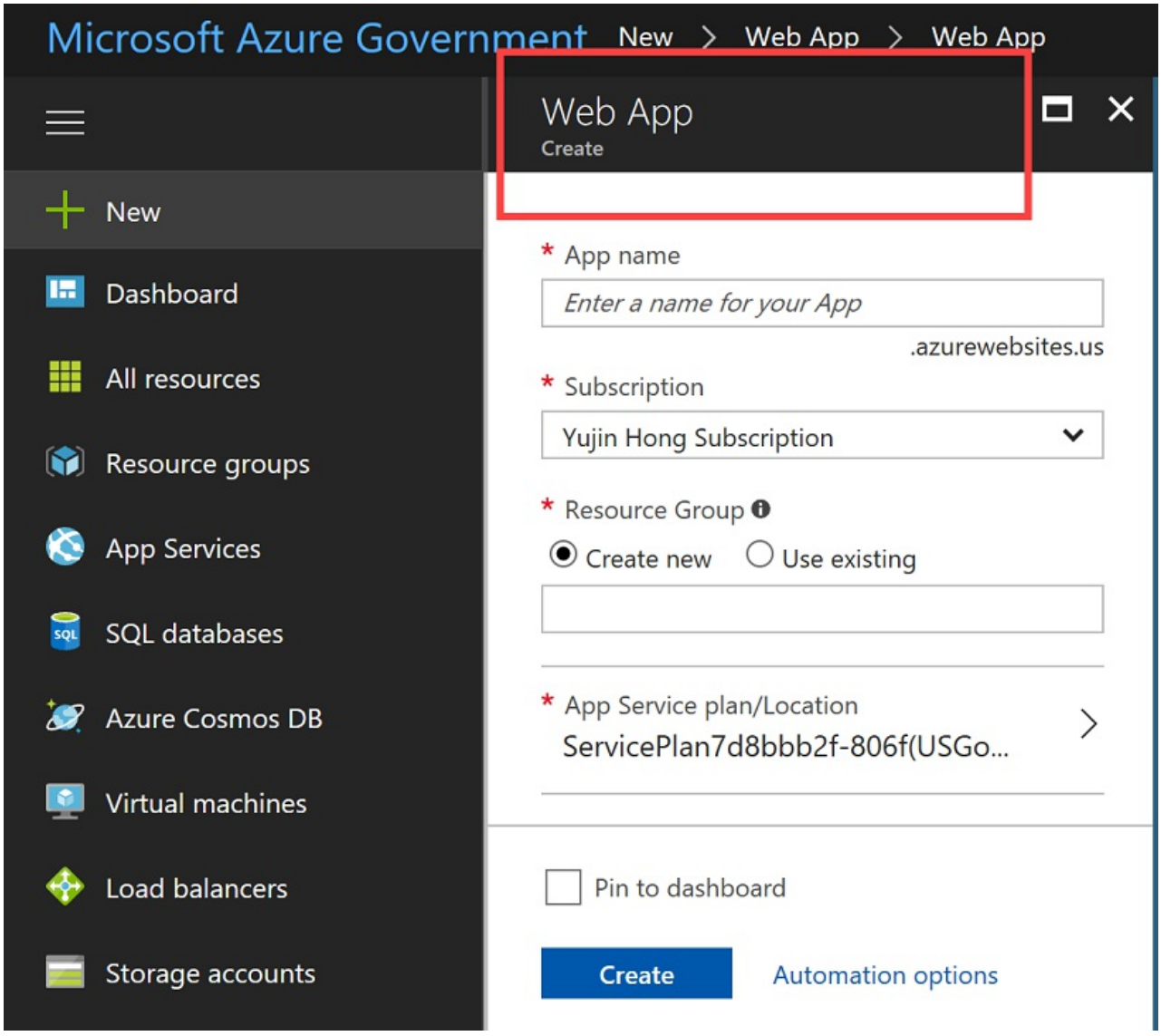
This article describes how to deploy an Azure App Services app (API App, Web App, Mobile App) to Azure Government using Visual Studio 2017.

Prerequisites

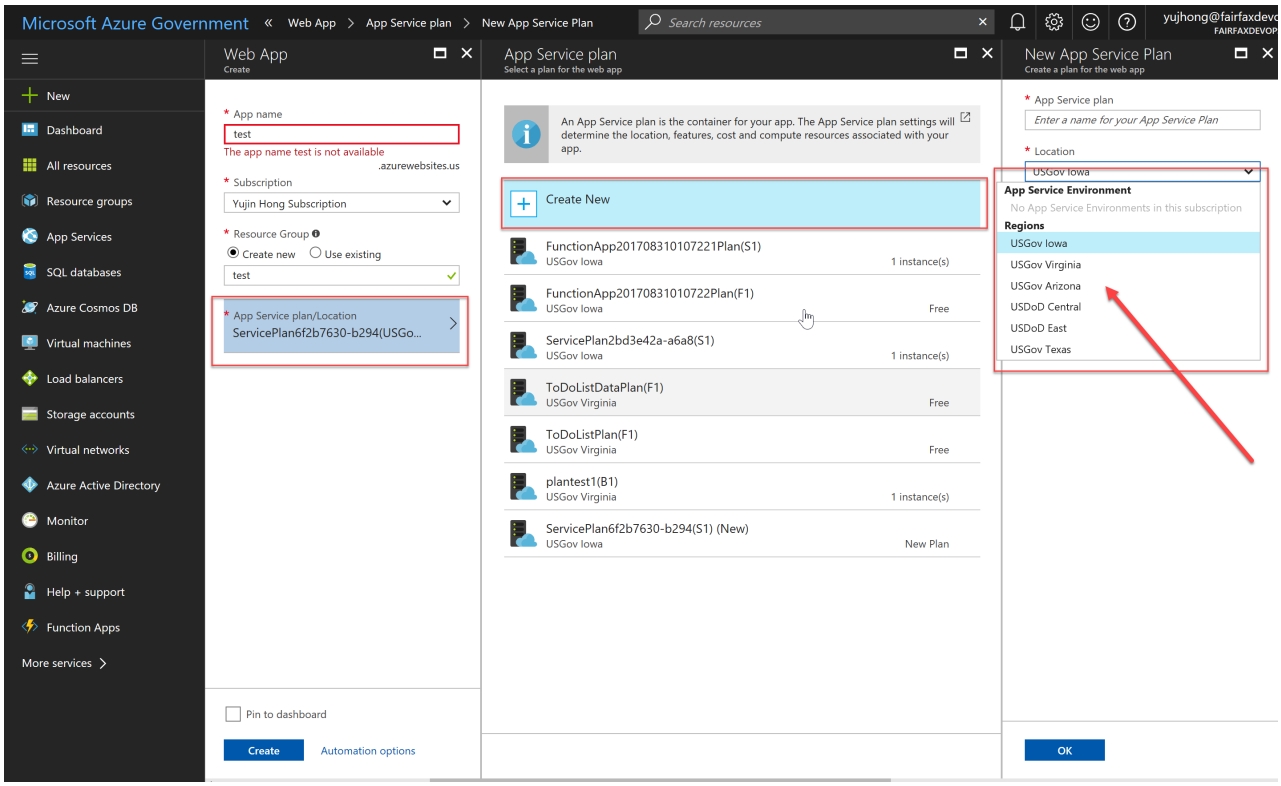
- See Visual Studio prerequisites to install and configure Visual Studio 2017 and Azure SDK.
- Follow these instructions to configure Visual Studio to connect to Azure Government account.

Provision a Web App in the Azure Government Portal

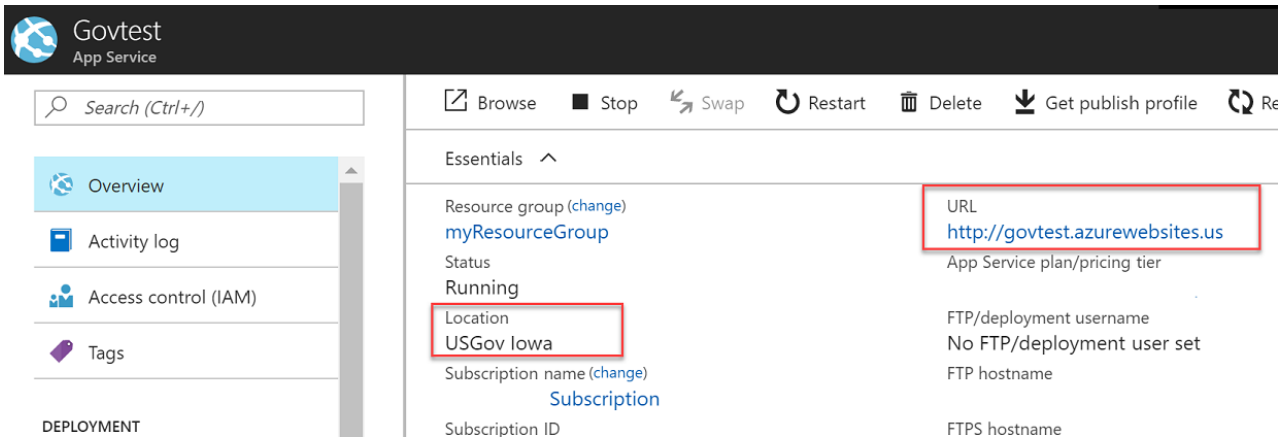
Log in to the Azure Government Portal. Click the **New** button on the top left-hand corner and choose to create **Web App**:



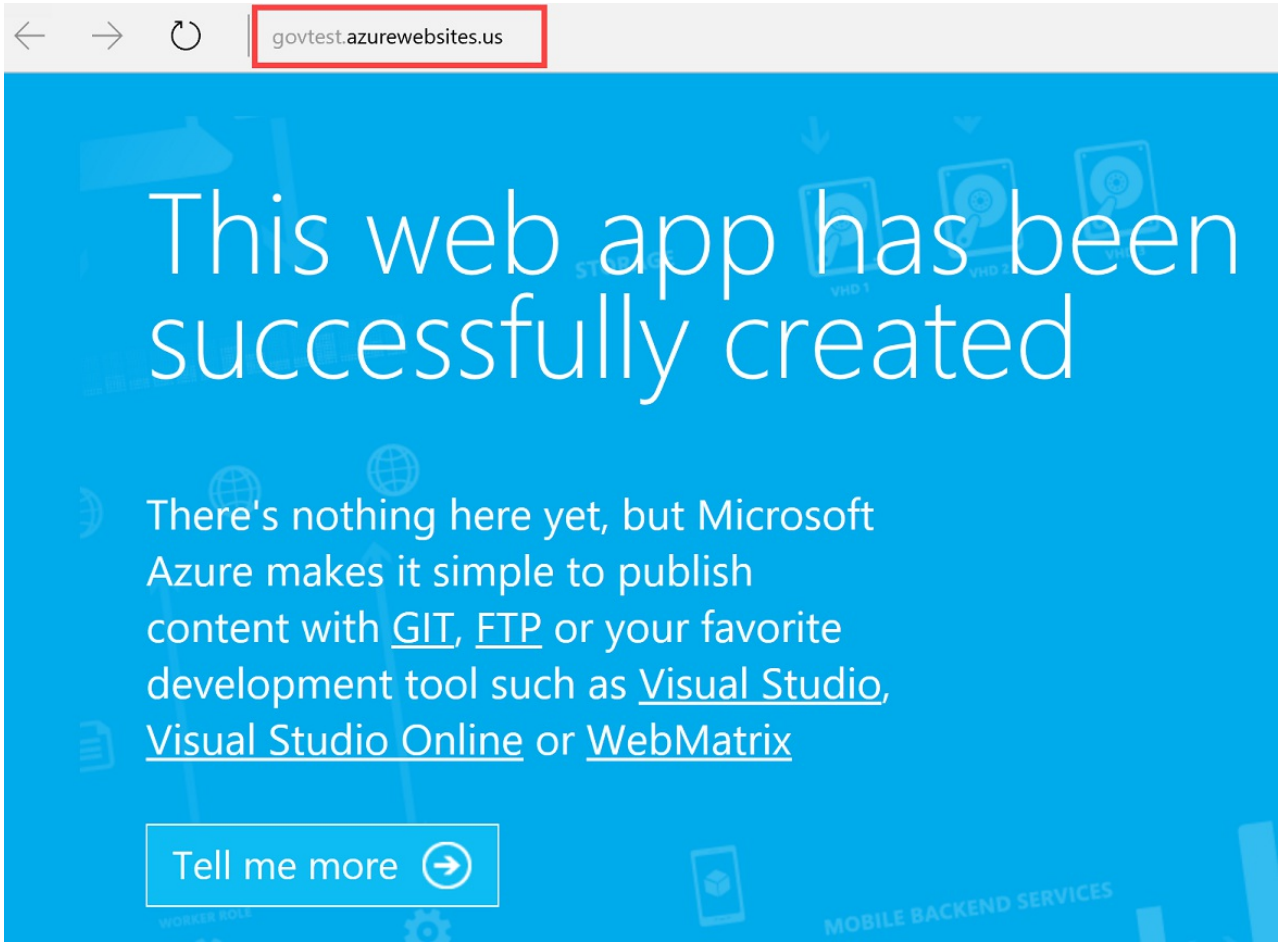
When creating the Web App, you must also have an App Service Plan. When creating a new App Service Plan, you should be able to see the different Azure Government regions in the **App Service Environment** box. If your subscription has not been approved for DoD regions, you may not see all of the DoD regions shown in the following screenshot.



Once the app has been successfully created go into the **App Services** section and you will be able to see your new web app. Click on your web app and you should see that the url ends in **azurewebsites.us**, and the location should also be an Azure Government region.



When you click on your app url, a blue page will appear:



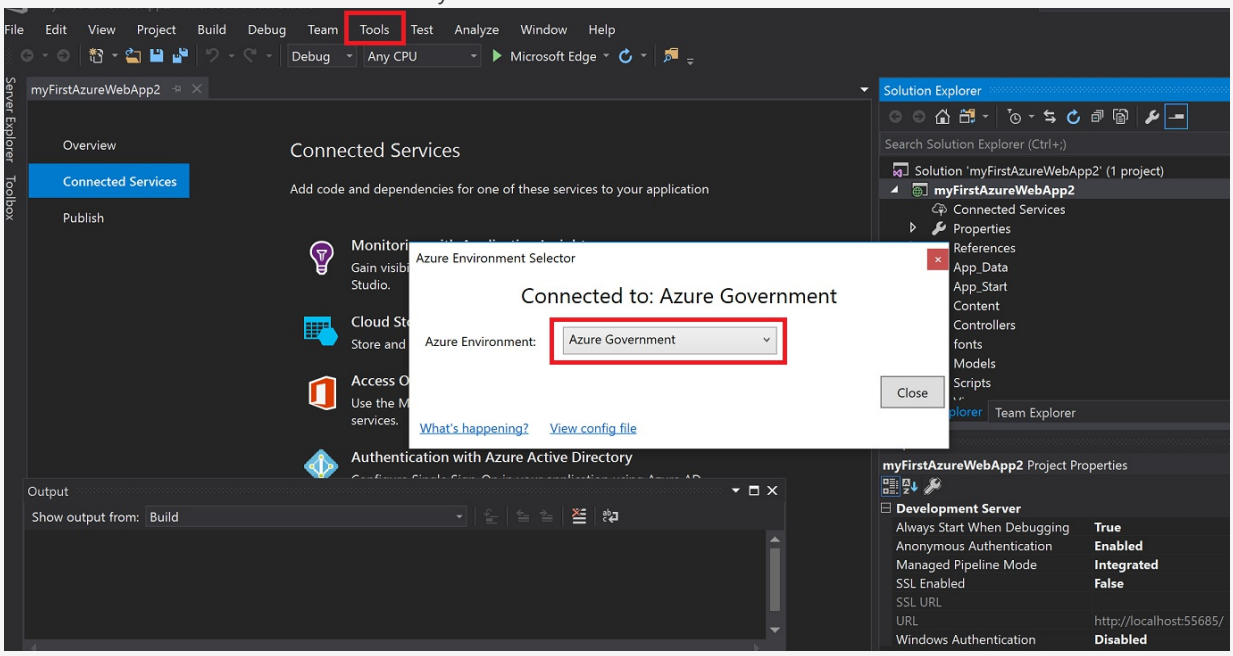
Deploy a Web App to Azure Government

Once **Visual Studio is configured to connect to Azure Government account** (already done in prerequisites section), there are two ways of deploying to Azure Government using Visual Studio:

1. Direct publish with Azure Active Directory user authentication
2. Publish with **Publish Profile** option(which can be found in the portal)

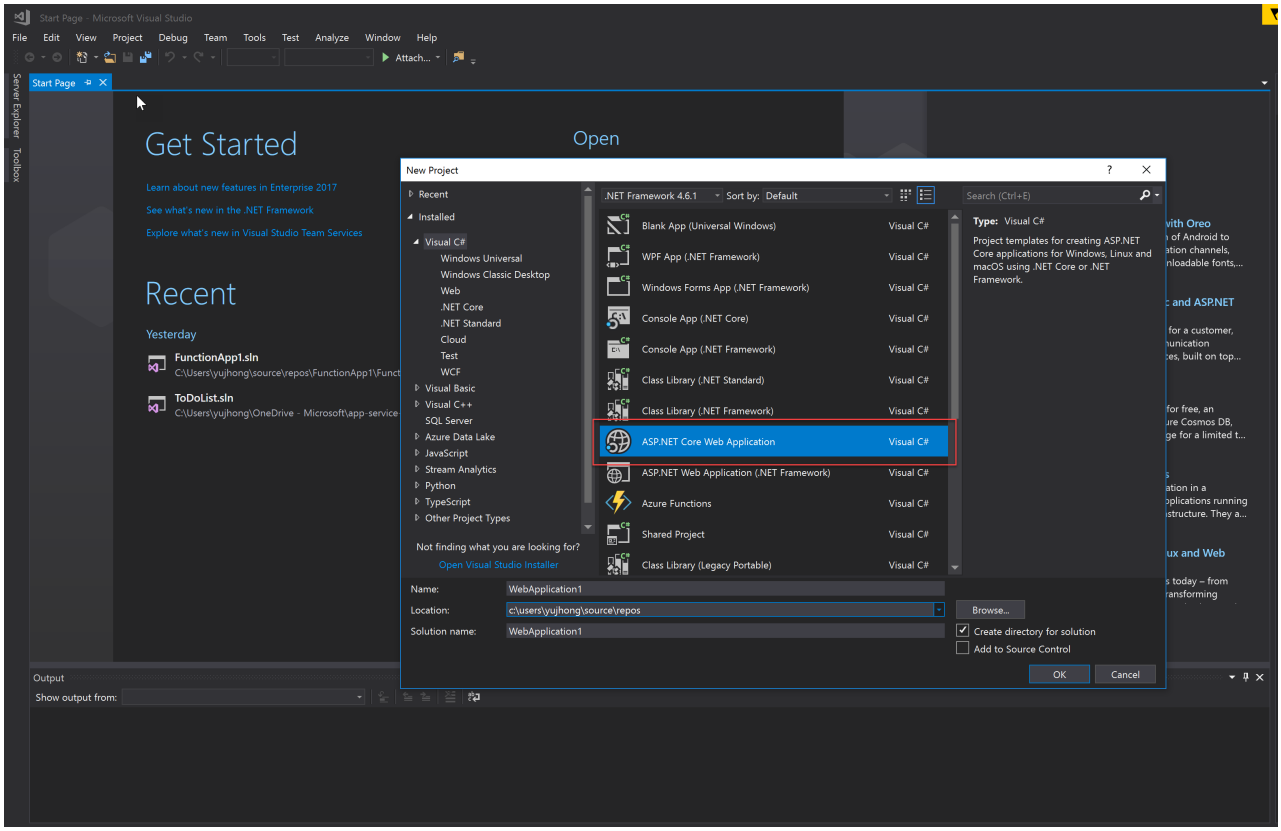
NOTE

In order to check if Visual Studio is connected to Azure Government, go to the **Tools** tab and click on the Azure Environment Selector extension to see what environment you are connected to.

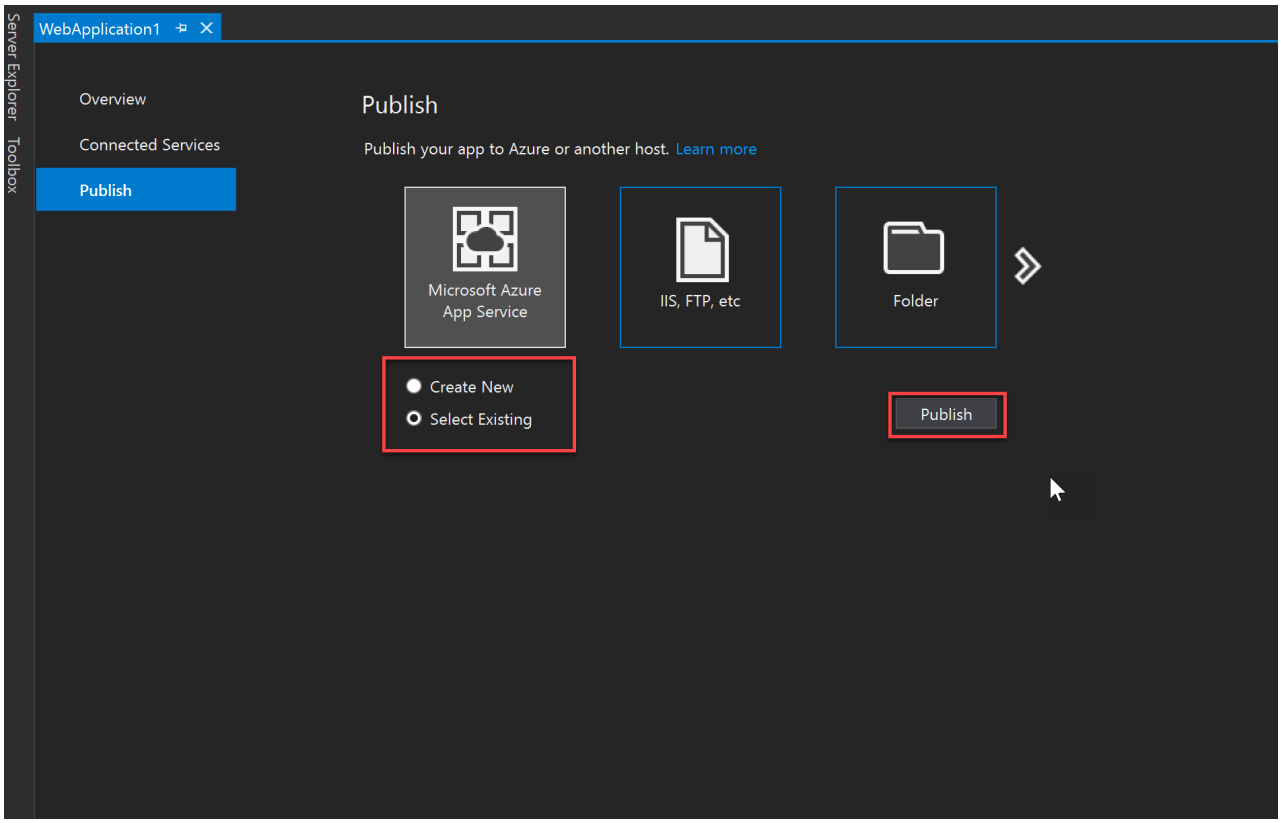


Publish using Azure Active Directory User Authentication from Visual Studio

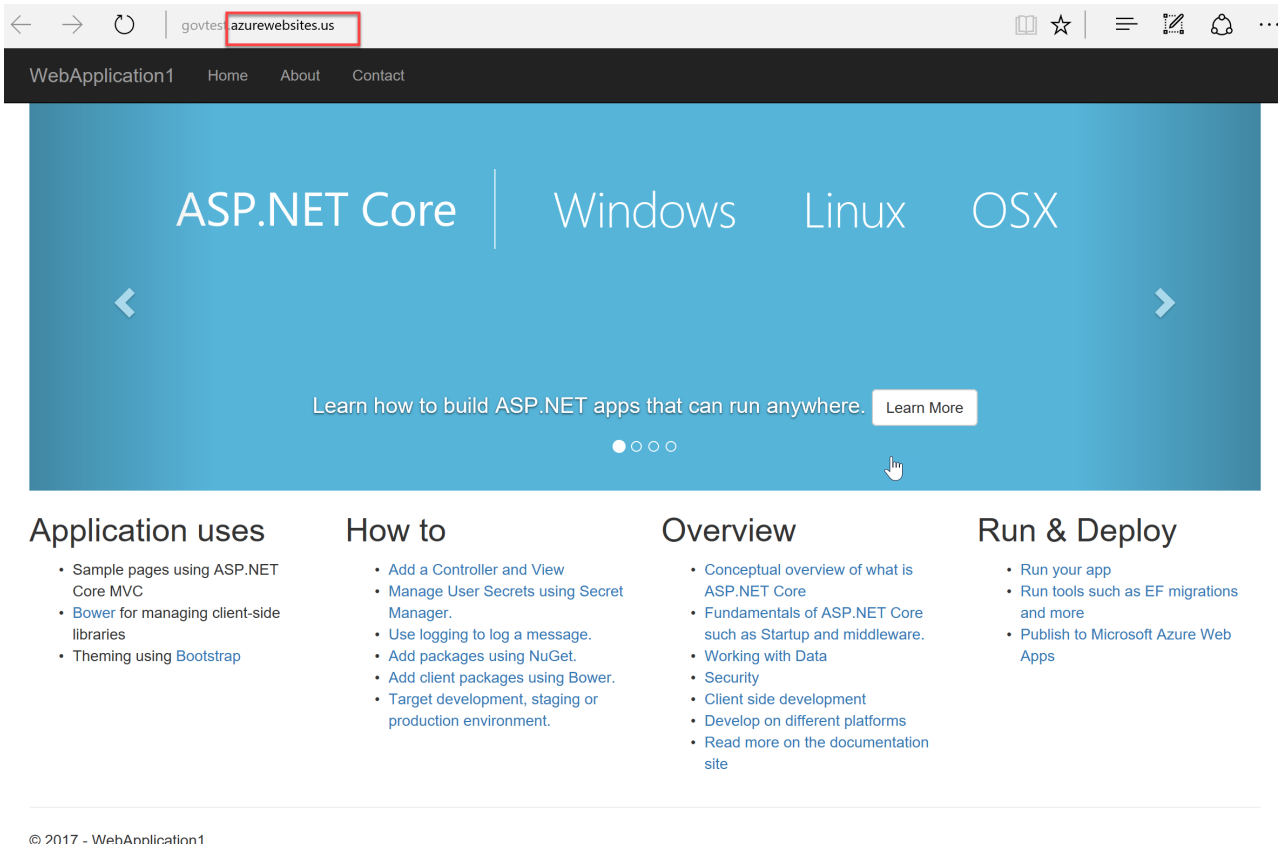
Open up Visual Studio and click File -> New -> Project. We will create an ASP.NET Core Web Application but Azure Web Apps supports a variety of web technologies as you can see on the page.



Right click on your web app and click **Publish**. There are two different options, to use an existing Web App or create a new one. We will use the existing Web App we have created in the preceding section, so choose the **select existing** box. To deploy an app and also have Visual Studio provision a new Azure Web App during the publish process, click **create new**.



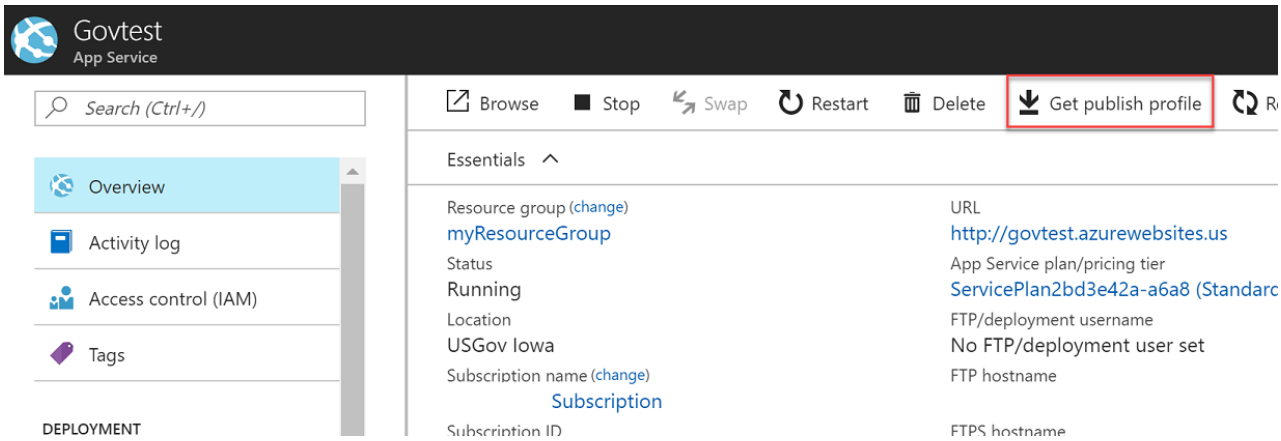
Once your web app has been published to Azure Government, you should be able to see this screen, with the URL ending in **azurewebsites.us**.



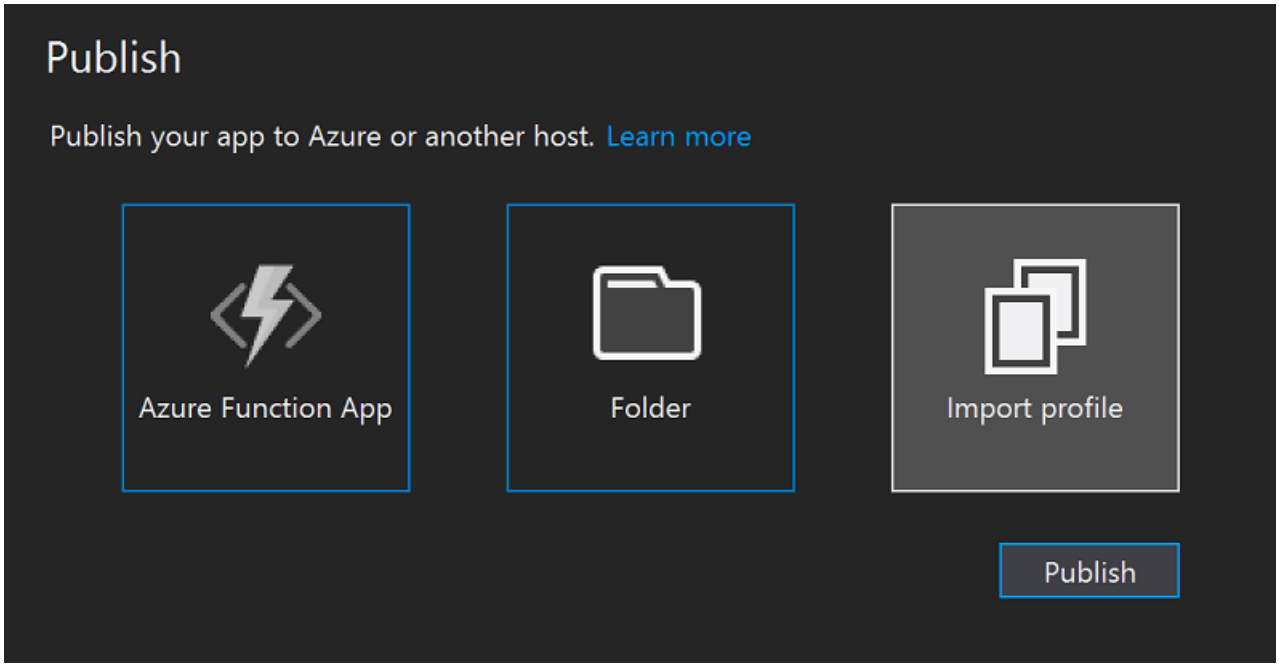
We just published our app to the existing Azure Web App that we previously created.

Deploy using Publish profile

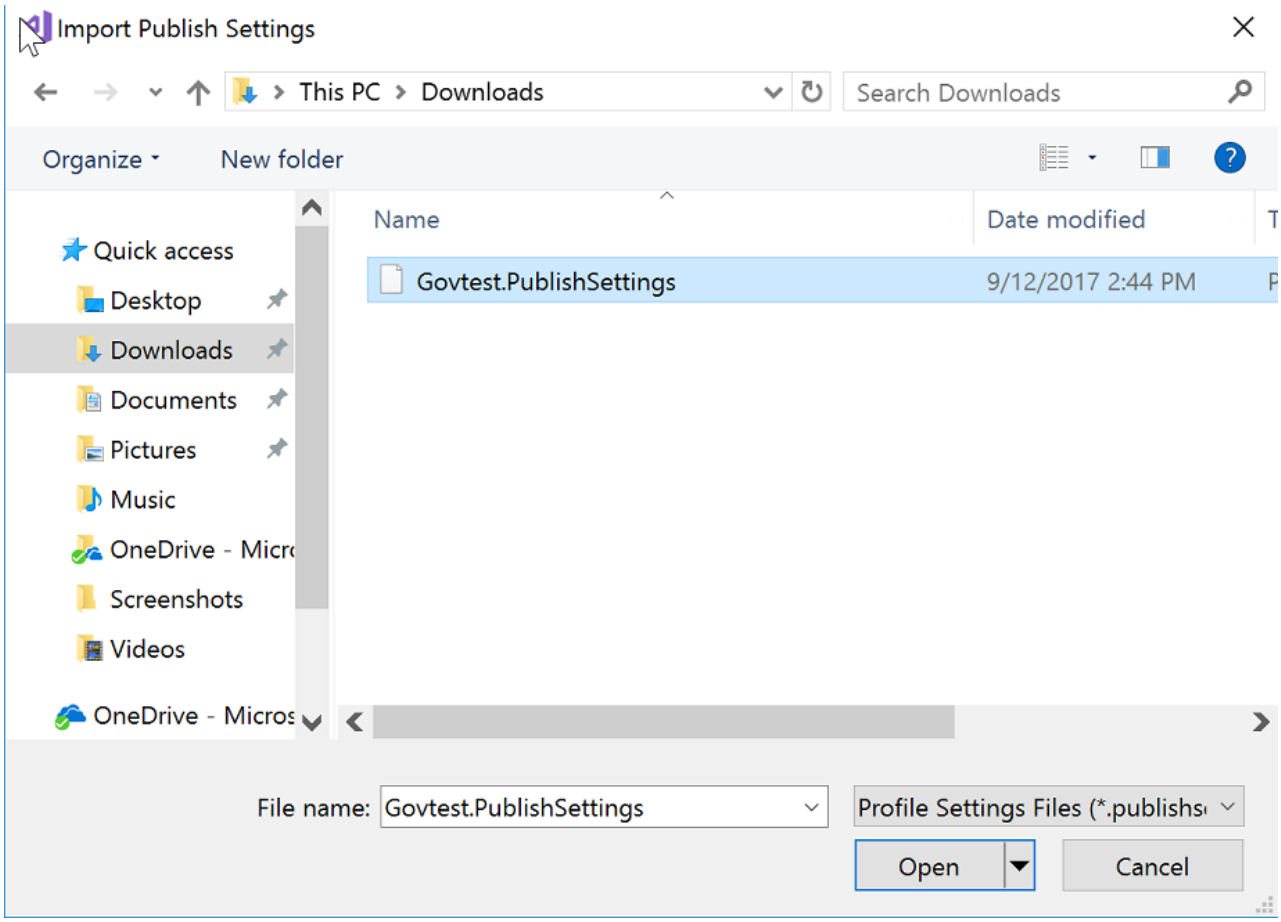
Log in to the Azure Government Portal. Click on **App Services** and choose your web app that you want to deploy. Then Click the **Get publish profile** button at the top of the page and download(take note of where the file was downloaded):



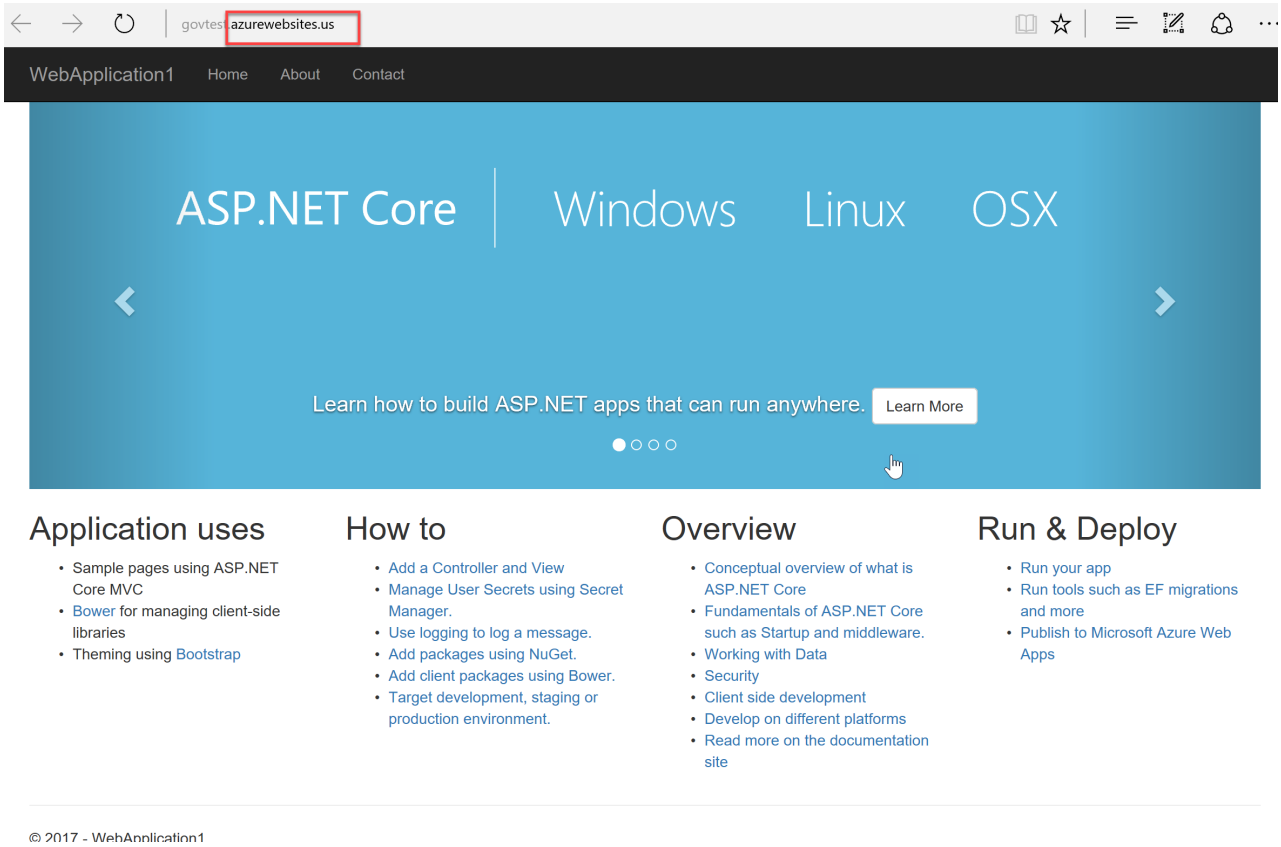
Open up Visual Studio and right click on your app solution.



Choose the **Import Profile** option and select the publish profile file from the location where you previously downloaded it. Click **publish**. Now you will be able to upload the publish profile that you downloaded from the portal.



If you navigate to the url, you should be able to see this screen.



The app has now been deployed to Azure Government.

References

- Deploy an ASP.NET web app to Azure App Service, using Visual Studio
- For general App Service documentation, see App Service - API Apps Documentation

Next steps

For supplemental information and updates, subscribe to the Microsoft Azure Government Blog.

Cognitive Services on Azure Government – Computer Vision, Face, Translator Text APIs

7/3/2018 • 12 minutes to read • Edit Online

To see an overview of Cognitive Services on Azure Government, click here.

Prerequisites

- Install and Configure Azure PowerShell
- Connect PowerShell with Azure Government

Part 1: Provision Cognitive Services Accounts

In order to access any of the Cognitive Services APIs, you must first provision a Cognitive Services account for each of the APIs you want to access. **Cognitive Services is not yet supported in the Azure Government Portal**, but you can use Azure PowerShell to access the APIs and services.

> **NOTE**
> You must go through the process of creating an account and retrieving a key(explained below) **for each** of the APIs you want to access.

1. Make sure that you have the **Cognitive Services resource provider registered on your account**.

You can do this by **running the following PowerShell command:**

```
Get-AzureRmResourceProvider
```

If you do **not see** `Microsoft.CognitiveServices`, you have to register the resource provider by **running the following command**:

```
Register-AzureRmResourceProvider -ProviderNamespace Microsoft.CognitiveServices
```

1. In the PowerShell command below, replace "rg-name", "name-of-your-api", and "location-of-resourcegroup" with your relevant account information.

   Replace the "type of API" tag with any of the three following APIs you want to access:

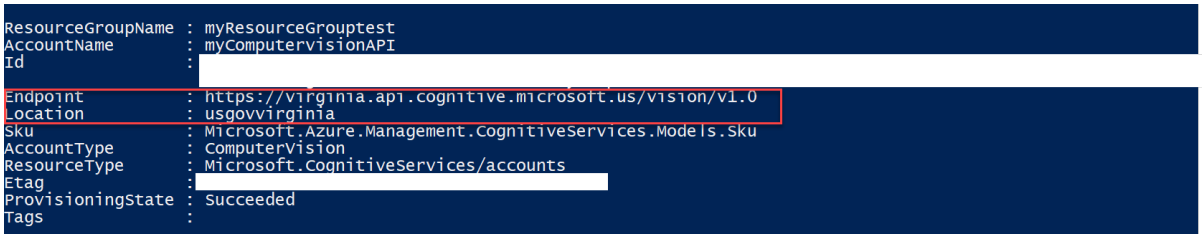
   - ComputerVision
   - Face
   - TextTranslation

   ```
   New-AzureRmCognitiveServicesAccount -ResourceGroupName 'rg-name' -name 'name-of-your-api' -Type <type of API> -SkuName S0 -Location 'location-of-resourcegroup'
   ```

   Example:

```
New-AzureRmCognitiveServicesAccount -ResourceGroupName 'resourcegrouptest' -name 'myFaceAPI' -Type Face
-SkuName S0 -Location 'usgovvirginia'
```

After you run the command, you should see something like this:



2. Copy and save the "Endpoint" attribute somewhere as you will need it when making calls to the API.

Retrieve Account Key

You must retrieve an account key to access the specific API.

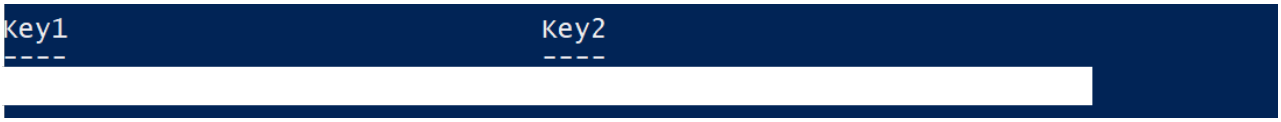
In the PowerShell command below, replace the "youraccountname" tag with the name that you gave the Account that you created above. Replace the 'rg-name' tag with the name of your resource group.

```
Get-AzureRmCognitiveServicesAccountKey -Name <youraccountname> -ResourceGroupName 'rg-name'
```

Example:

```
Get-AzureRmCognitiveServicesAccountKey -Name myFaceAPI -ResourceGroupName 'resourcegrouptest'
```

Copy and save the first key somewhere as you will need it to make calls to the API.



Now you are ready to make calls to the APIs.

Part 2: API Quickstarts

The Quickstarts below will help you to get started with the APIs available through Cognitive Services in Azure Government.

Computer Vision API

Prerequisites

- Get the Microsoft Computer Vision API Windows SDK here.
- Make sure Visual Studio has been installed:
  - Visual Studio 2017 version 15.3, including the **Azure development** workload.

> **NOTE**
>
> After you install or upgrade to Visual Studio 2017 version 15.3, you might also need to manually update the Visual Studio 2017 tools for Azure Functions. You can update the tools from the **Tools** menu under **Extensions and Updates...** > **Updates** > **Visual Studio Marketplace** > **Azure Functions and Web Jobs Tools** > **Update**.

Variations

- The URI for accessing the Face API in Azure Government is :
  - `https://(resource-group-location).api.cognitive.microsoft.us/face/v1.0`
  - The main difference between this URI and the URI used in Commercial Azure is the ending of **.us** and the location at the beginning of the uri

Analyze an Image With Computer Vision API using C#

With the Analyze Image method, you can extract visual features based on image content. You can upload an image or specify an image URL and choose which features to return, including:

- A detailed list of tags related to the image content.
- A description of image content in a complete sentence.
- The coordinates, gender, and age of any faces contained in the image.
- The ImageType (clip art or a line drawing).
- The dominant color, the accent color, or whether an image is black & white.
- The category defined in this taxonomy.
- Does the image contain adult or sexually suggestive content?

Analyze an image C# example request

1. Create a new Console solution in Visual Studio.
2. Replace Program.cs with the following code.
3. Change the `uriBase` to the "Endpoint" attribute that you saved from Part 1, and keep the "/analyze" after the endpoint.
4. Replace the `subscriptionKey` value with your valid subscription key.
5. Run the program.

```
using System;
using System.IO;
using System.Net.Http;
using System.Net.Http.Headers;
using System.Text;

namespace VisionApp1
{
        static class Program
        {
            // **********************************************
            // *** Update or verify the following values. ***
            // **********************************************

            // Replace the subscriptionKey string value with your valid subscription key.
            const string subscriptionKey = "<subscription key>";

            //Copy and paste the "Endpoint" attribute that you saved before into the uriBase string "/analyze"
at the end.
            //Example: https://virginia.api.cognitive.microsoft.us/vision/v1.0/analyze

            const string uriBase = "<endpoint>/analyze";

            static void Main()
            {
                // Get the path and filename to process from the user.
                Console.WriteLine("Analyze an image:");
                Console.Write("Enter the path to an image you wish to analyze: ");
                string imageFilePath = Console.ReadLine();

                // Execute the REST API call.
                MakeAnalysisRequest(imageFilePath);
```

```
            Console.WriteLine("\nPlease wait a moment for the results to appear. Then, press Enter to
exit...\n");
            Console.ReadLine();
        }


        /// <summary>
        /// Gets the analysis of the specified image file by using the Computer Vision REST API.
        /// </summary>
        /// <param name="imageFilePath">The image file.</param>
        static async void MakeAnalysisRequest(string imageFilePath)
        {
            HttpClient client = new HttpClient();

            // Request headers.
            client.DefaultRequestHeaders.Add("Ocp-Apim-Subscription-Key", subscriptionKey);

            // Request parameters. A third optional parameter is "details".
            string requestParameters = "visualFeatures=Categories,Description,Color&language=en";

            // Assemble the URI for the REST API Call.
            string uri = uriBase + "?" + requestParameters;

            HttpResponseMessage response;

            // Request body. Posts a locally stored JPEG image.
            byte[] byteData = GetImageAsByteArray(imageFilePath);

            using (ByteArrayContent content = new ByteArrayContent(byteData))
            {
                // This example uses content type "application/octet-stream".
                // The other content types you can use are "application/json" and "multipart/form-data".
                content.Headers.ContentType = new MediaTypeHeaderValue("application/octet-stream");

                // Execute the REST API call.
                response = await client.PostAsync(uri, content);

                // Get the JSON response.
                string contentString = await response.Content.ReadAsStringAsync();

                // Display the JSON response.
                Console.WriteLine("\nResponse:\n");
                Console.WriteLine(JsonPrettyPrint(contentString));
            }
        }


        /// <summary>
        /// Returns the contents of the specified file as a byte array.
        /// </summary>
        /// <param name="imageFilePath">The image file to read.</param>
        /// <returns>The byte array of the image data.</returns>
        static byte[] GetImageAsByteArray(string imageFilePath)
        {
            FileStream fileStream = new FileStream(imageFilePath, FileMode.Open, FileAccess.Read);
            BinaryReader binaryReader = new BinaryReader(fileStream);
            return binaryReader.ReadBytes((int)fileStream.Length);
        }


        /// <summary>
        /// Formats the given JSON string by adding line breaks and indents.
        /// </summary>
        /// <param name="json">The raw JSON string to format.</param>
        /// <returns>The formatted JSON string.</returns>
        static string JsonPrettyPrint(string json)
        {
            if (string.IsNullOrEmpty(json))
                return string.Empty;
```

```
            json = json.Replace(Environment.NewLine, "").Replace("\t", "");

            StringBuilder sb = new StringBuilder();
            bool quote = false;
            bool ignore = false;
            int offset = 0;
            int indentLength = 3;

            foreach (char ch in json)
            {
                switch (ch)
                {
                    case '"':
                        if (!ignore) quote = !quote;
                        break;
                    case '\'':
                        if (quote) ignore = !ignore;
                        break;
                }

                if (quote)
                    sb.Append(ch);
                else
                {
                    switch (ch)
                    {
                        case '{':
                        case '[':
                            sb.Append(ch);
                            sb.Append(Environment.NewLine);
                            sb.Append(new string(' ', ++offset * indentLength));
                            break;
                        case '}':
                        case ']':
                            sb.Append(Environment.NewLine);
                            sb.Append(new string(' ', --offset * indentLength));
                            sb.Append(ch);
                            break;
                        case ',':
                            sb.Append(ch);
                            sb.Append(Environment.NewLine);
                            sb.Append(new string(' ', offset * indentLength));
                            break;
                        case ':':
                            sb.Append(ch);
                            sb.Append(' ');
                            break;
                        default:
                            if (ch != ' ') sb.Append(ch);
                            break;
                    }
                }
            }

            return sb.ToString().Trim();
        }
    }
}
```

Analyze an Image response

A successful response is returned in JSON. Following is an example of a successful response:

```
{
   "categories": [
      {
         "name": "people_baby",
         "score": 0.52734375
      },
      {
         "name": "people_young",
         "score": 0.4375
      }
   ],
   "description": {
      "tags": [
         "person",
         "indoor",
         "clothing",
         "woman",
         "white",
         "table",
         "food",
         "girl",
         "smiling",
         "posing",
         "holding",
         "black",
         "sitting",
         "young",
         "plate",
         "hair",
         "wearing",
         "cake",
         "large",
         "shirt",
         "dress",
         "eating",
         "standing",
         "blue"
      ],
      "captions": [
         {
            "text": "a woman posing for a picture",
            "confidence": 0.460196158842535
         }
      ]
   },
   "requestId": "7c20cc50-f5eb-453b-abb5-98378917431c",
   "metadata": {
      "width": 721,
      "height": 960,
      "format": "Jpeg"
   },
   "color": {
      "dominantColorForeground": "Black",
      "dominantColorBackground": "White",
      "dominantColors": [
         "White"
      ],
      "accentColor": "7C4F57",
      "isBWImg": false
   }
}
```


For more information, please see public documentation and public API documentation for Computer Vision API.

Face API

Prerequisites

- Get the Microsoft Face API Windows SDK here
- Make sure Visual Studio has been installed:
  - Visual Studio 2017 version 15.3, including the **Azure development** workload.

> **NOTE**
> After you install or upgrade to Visual Studio 2017 version 15.3, you might also need to manually update the Visual Studio 2017 tools for Azure Functions. You can update the tools from the **Tools** menu under **Extensions and Updates...** > **Updates** > **Visual Studio Marketplace** > **Azure Functions and Web Jobs Tools** > **Update**.

Variations

- The URI for accessing the Face API in Azure Government is :
  - `https://(resource-group-location).api.cognitive.microsoft.us/face/v1.0`
  - The main difference between this URI and the URI used in Commercial Azure is the ending of **.us** and the location at the beginning of the uri

Detect Faces in images with Face API using C#

Use the Face - Detect method to detect faces in an image and return face attributes including:

- Face ID: Unique ID used in several Face API scenarios.
- Face Rectangle: The left, top, width, and height indicating the location of the face in the image.
- Landmarks: An array of 27-point face landmarks pointing to the important positions of face components.
- Facial attributes including age, gender, smile intensity, head pose, and facial hair.

Face detect C# example request

The sample is written in C# using the Face API client library.

1. Create a new Console solution in Visual Studio.
2. Replace Program.cs with the following code.
3. Replace the `subscriptionKey` value with the key value that you retrieved above.
4. Change the `uriBase` value to the "Endpoint" attribute you retrieved above.
5. Run the program.
6. Enter the path to an image on your hard drive.

```
using System;
using System.IO;
using System.Net.Http;
using System.Net.Http.Headers;
using System.Text;

namespace FaceApp1
{

    static class Program
    {
        // **********************************************
        // *** Update or verify the following values. ***
        // **********************************************

        // Replace the subscriptionKey string value with your valid subscription key.
```

```
        // Replace the subscriptionKey string value with your valid subscription key.
        const string subscriptionKey = "<subscription key>";

        //Copy and paste the "Endpoint" attribute that you saved before into the uriBase string "/detect" at
the end.
        //Example: https://virginia.api.cognitive.microsoft.us/face/v1.0/detect
        const string uriBase ="<endpoint>/detect";

        static void Main()
        {
            // Get the path and filename to process from the user.
            Console.WriteLine("Detect faces:");
            Console.Write("Enter the path to an image with faces that you wish to analzye: ");
            string imageFilePath = Console.ReadLine();

            // Execute the REST API call.
            MakeAnalysisRequest(imageFilePath);

            Console.WriteLine("\nPlease wait a moment for the results to appear. Then, press Enter to
exit...\n");
            Console.ReadLine();
        }


        /// <summary>
        /// Gets the analysis of the specified image file by using the Computer Vision REST API.
        /// </summary>
        /// <param name="imageFilePath">The image file.</param>
        static async void MakeAnalysisRequest(string imageFilePath)
        {
            HttpClient client = new HttpClient();

            // Request headers.
            client.DefaultRequestHeaders.Add("Ocp-Apim-Subscription-Key", subscriptionKey);

            // Request parameters. A third optional parameter is "details".
            string requestParameters =
"returnfaceId=true&returnfaceLandmarks=false&returnfaceAttributes=age,gender,headPose,smile,facialHair,glasses,
emotion";

            // Assemble the URI for the REST API Call.
            string uri = uriBase + "?" + requestParameters;

            HttpResponseMessage response;

            // Request body. Posts a locally stored JPEG image.
            byte[] byteData = GetImageAsByteArray(imageFilePath);

            using (ByteArrayContent content = new ByteArrayContent(byteData))
            {
                // This example uses content type "application/octet-stream".
                // The other content types you can use are "application/json" and "multipart/form-data".
                content.Headers.ContentType = new MediaTypeHeaderValue("application/octet-stream");

                // Execute the REST API call.
                response = await client.PostAsync(uri, content);

                // Get the JSON response.
                string contentString = await response.Content.ReadAsStringAsync();

                // Display the JSON response.
                Console.WriteLine("\nResponse:\n");
                Console.WriteLine(JsonPrettyPrint(contentString));
            }
        }


        /// <summary>
        /// Returns the contents of the specified file as a byte array.
        /// </summary>
```

```
/// </summary>
/// <param name="imageFilePath">The image file to read.</param>
/// <returns>The byte array of the image data.</returns>
static byte[] GetImageAsByteArray(string imageFilePath)
{
    FileStream fileStream = new FileStream(imageFilePath, FileMode.Open, FileAccess.Read);
    BinaryReader binaryReader = new BinaryReader(fileStream);
    return binaryReader.ReadBytes((int)fileStream.Length);
}


/// <summary>
/// Formats the given JSON string by adding line breaks and indents.
/// </summary>
/// <param name="json">The raw JSON string to format.</param>
/// <returns>The formatted JSON string.</returns>
static string JsonPrettyPrint(string json)
{
    if (string.IsNullOrEmpty(json))
        return string.Empty;

    json = json.Replace(Environment.NewLine, "").Replace("\t", "");

    StringBuilder sb = new StringBuilder();
    bool quote = false;
    bool ignore = false;
    int offset = 0;
    int indentLength = 3;

    foreach (char ch in json)
    {
        switch (ch)
        {
            case '"':
                if (!ignore) quote = !quote;
                break;
            case '\'':
                if (quote) ignore = !ignore;
                break;
        }

        if (quote)
            sb.Append(ch);
        else
        {
            switch (ch)
            {
                case '{':
                case '[':
                    sb.Append(ch);
                    sb.Append(Environment.NewLine);
                    sb.Append(new string(' ', ++offset * indentLength));
                    break;
                case '}':
                case ']':
                    sb.Append(Environment.NewLine);
                    sb.Append(new string(' ', --offset * indentLength));
                    sb.Append(ch);
                    break;
                case ',':
                    sb.Append(ch);
                    sb.Append(Environment.NewLine);
                    sb.Append(new string(' ', offset * indentLength));
                    break;
                case ':':
                    sb.Append(ch);
                    sb.Append(' ');
                    break;
                default:
```

```
                            if (ch != ' ') sb.Append(ch);
                            break;
                    }
                }
            }

            return sb.ToString().Trim();
        }
    }
}
```

Face detect response

A successful response is returned in JSON. Following is an example of a successful response:

```
Response:
[
   {
      "faceId": "0ed7f4db-1207-40d4-be2e-84694e42d682",
      "faceRectangle": {
         "top": 60,
         "left": 83,
         "width": 361,
         "height": 361
      },
      "faceAttributes": {
         "smile": 0.284,
         "headPose": {
            "pitch": 0.0,
            "roll": -12.2,
            "yaw": -16.7
         },
         "gender": "female",
         "age": 16.5,
         "facialHair": {
            "moustache": 0.0,
            "beard": 0.0,
            "sideburns": 0.0
         },
         "glasses": "NoGlasses",
         "emotion": {
            "anger": 0.003,
            "contempt": 0.001,
            "disgust": 0.001,
            "fear": 0.002,
            "happiness": 0.284,
            "neutral": 0.694,
            "sadness": 0.012,
            "surprise": 0.004
         }
      }
   }
]
```

For more information, please see public documentation, and public API documentation for Face API.

Text Translation API

Prerequisites

- Make sure Visual Studio has been installed:
    - Visual Studio 2017 version 15.3, including the **Azure development** workload.

> **NOTE**
>
> After you install or upgrade to Visual Studio 2017 version 15.3, you might also need to manually update the Visual Studio 2017 tools for Azure Functions. You can update the tools from the **Tools** menu under **Extensions and Updates...** > **Updates** > **Visual Studio Marketplace** > **Azure Functions and Web Jobs Tools** > **Update**.

Variations

- The URI for accessing the Text Translation API in Azure Government is:
  - `https://dev.microsofttranslator.us/translate?api-version=3.0` ### Text Translation Method This sample will use the Text Translation - Translate method to translate a string of text from a language into another specified language. There are multiple language codes that can be used with the Text Translation API.

Text Translation C# example request

The sample is written in C#.

1. Create a new Console solution in Visual Studio.
2. Replace Program.cs with the corresponding code below.
3. Replace the `subscriptionKey` value with the key value that you retrieved above.
4. Replace the `text` value with text that you want to translate.
5. Run the program.

You can also test out different languages and texts by replacing the "text", "from", and "to" variables in Program.cs.

```
using System;
using Microsoft.Azure.CognitiveServices.Language.TextAnalytics;
using Microsoft.Azure.CognitiveServices.Language.TextAnalytics.Models;
using System.Collections.Generic;
using Microsoft.Rest;
using System.Net.Http;
using System.Threading;
using System.Threading.Tasks;
using System.Net;
using System.IO;
using Newtonsoft.Json;
using System.Text;

namespace TextTranslator
{
    class Program
    {
        static string host = "https://dev.microsofttranslator.us";
        static string path = "/translate?api-version=3.0";
        // Translate to German.
        static string params_ = "&to=de";

        static string uri = host + path + params_;

        // NOTE: Replace this example key with a valid subscription key.
        static string key = "PASTE KEY HERE";

        static string text = "Hello world!";

        async static void Translate()
        {
            System.Object[] body = new System.Object[] { new { Text = text } };
            var requestBody = JsonConvert.SerializeObject(body);

            using (var client = new HttpClient())
            using (var request = new HttpRequestMessage())
            {
                request.Method = HttpMethod.Post;
                request.RequestUri = new Uri(uri);
                request.Content = new StringContent(requestBody, Encoding.UTF8, "application/json");
                request.Headers.Add("Ocp-Apim-Subscription-Key", key);

                var response = await client.SendAsync(request);
                var responseBody = await response.Content.ReadAsStringAsync();
                var result = JsonConvert.SerializeObject(JsonConvert.DeserializeObject(responseBody),
Formatting.Indented);

                Console.OutputEncoding = UnicodeEncoding.UTF8;
                Console.WriteLine(result);
            }
        }

        static void Main(string[] args)
        {
            Translate();
            Console.ReadLine();
        }
    }
}
```

For more information, please see public documentation and public API documentation for Translator Text API.

Next Steps

- Subscribe to the Azure Government blog
- Get help on Stack Overflow by using the "azure-gov" tag

- Give us feedback or request new features via the Azure Government feedback forum

Integrate Azure AD Authentication with Web Apps on Azure Government

2/16/2018 • 3 minutes to read • Edit Online

The following quickstart helps you get started integrating Azure AD Authentication with applications on Azure Government. Azure Active Directory (Azure AD) Authentication on Azure Government is similar to the Azure commercial platform, with a few exceptions.

Learn more about Azure Active Directory Authentication Scenarios.

Integrate Azure AD login into a web application using OpenID Connect

This section shows how to integrate Azure AD using the OpenID Connect protocol for signing in users into a web app.

Prerequisites

- An Azure AD tenant in Azure Government. You must have an Azure Government subscription in order to have an Azure AD tenant in Azure Government. For more information on how to get an Azure AD tenant, see How to get an Azure AD tenant
- A user account in your Azure AD tenant. This sample does not work with a Microsoft account, so if you signed in to the Azure Government portal with a Microsoft account and have never created a user account in your directory before, you need to do that now.
- Have an ASP.NET Core application deployed and running in Azure Government

Step 1: Register your web application with your Azure AD Tenant

1. Sign in to the Azure Government portal.
2. On the top bar, click on your account and under the **Directory** list, choose the Active Directory tenant where you wish to register your application.
3. Click on **All Services** in the left-hand nav, and choose **Azure Active Directory**.
4. Click on **App registrations** and choose **Add**.
5. Enter the name for your application, and select 'Web Application and/or Web API' as the Application Type. For the sign-on URL, enter the base URL for your application, which is your Azure App URL + "/signin-oidc."

   > **NOTE**
   > If you have not deployed your application and want to run it locally, your App URL would be your local host address.

   Click on **Create** to create the application.
6. While still in the Azure portal, choose your application, click on **Settings**, and choose **Properties**.
7. Find the Application ID value and copy it to the clipboard.
8. For the App ID URI, enter https://<your_tenant_name>/<name_of_your_app>, replacing <your_tenant_name> with the name of your Azure AD tenant and <name_of_your_app> with the name of your application.

Step 2: Configure your app to use your Azure AD tenant

Azure Government Variations

The only variation when setting up Azure AD Authorization on the Azure Government cloud is in the Azure AD

Instance:

- "https://login.microsoftonline.us"

Configure the InventoryApp project

1. Open your application in Visual Studio 2017.
2. Open the `appsettings.json` file.
3. Add an `Authentication` section and fill out the properties with your Azure AD tenant information.

```
//ClientId: Azure AD->  App registrations -> Application ID
//Domain: <tenantname>.onmicrosoft.com
//TenantId: Azure AD -> Properties -> Directory ID

"Authentication": {
    "AzureAd": {

    "Azure ADInstance": "https://login.microsoftonline.us/",
    "CallbackPath": "/signin-oidc",
    "ClientId": "<clientid>",
    "Domain": "<domainname>",
    "TenantId": "<tenantid>"
    }
}
```

4. Fill out the `ClientId` property with the Client ID for your app from the Azure Government portal. You can find the Client ID by navigating to Azure AD -> App Registrations -> Your Application -> Application ID.
5. Fill out the `TenantId` property with the Tenant ID for your app from the Azure Government portal. You can find the Tenant ID by navigating to Azure AD -> Properties -> Directory ID.
6. Fill out the `Domain` property with ".onmicrosoft.com."
7. Open the `startup.cs` file.
8. In your `ConfigureServices` method, add the following code:

```
    public void ConfigureServices(IServiceCollection services)
    {
        //Add Azure AD authentication
        services.AddAuthentication(options => {
            options.DefaultScheme = CookieAuthenticationDefaults.AuthenticationScheme;
            options.DefaultChallengeScheme = OpenIdConnectDefaults.AuthenticationScheme;
        })
        .AddCookie()
        .AddOpenIdConnect(options => {
            options.Authority = Configuration["Authentication:AzureAd:Azure ADInstance"] +
Configuration["Authentication:AzureAd:TenantId"];
            options.ClientId = Configuration["Authentication:AzureAd:ClientId"];
            options.CallbackPath = Configuration["Authentication:AzureAd:CallbackPath"];
        });

    }
```

   In the same file, add this one line of code to the `Configure` method:

```
app.UseAuthentication();
```

9. Navigate to your **Home** controller or whichever controller file is your home page, **where you want your users to log in**. Add the `[Authorize]` tag before the class definition.

Next steps

- Navigate to the Azure Government PaaS Sample to see Azure AD Authentication as well as other services being integrated in an Application running on Azure Government.
- Subscribe to the Azure Government blog
- Get help on Stack Overflow by using the "azure-gov" tag
- Give feedback or request new features via the Azure Government feedback forum

GPUs on Azure Government

11/30/2017 • 2 minutes to read • Edit Online

This page will help you get started using GPUs on Azure Government.

Prerequisites

To get started with GPUs and Data Science VMs on Azure Government, you must have an active Azure Government subscription. If you don't have an Azure Government subscription, create a free account before you begin.

Variations

NC-Virtual Machines powered by NVIDIA Tesla® K80 GPUs are available in the following regions:

- US Gov Arizona
- US Gov Texas

If you want to start deploying GPUs, navigate to the Marketplace.

Windows offerings: The following are supported in Azure Government:

- Windows Server 2016
- Windows Server 2012 R2

  Once the VM has been created, connect to the VM and install the NVIDIA Tesla drivers.

Linux offerings: The following are supported in Azure Government:

- Ubuntu 16.04 LTS
- Red Hat Enterprise Linux 7.3
- CentOS-based 7.3

  Once the VM has been created, connect to the VM and install the NVIDIA Tesla drivers.

Data Science VMs

For those new to Azure we recommend using the Data Science Virtual Machines which support Ubuntu and Windows Server 2016 DSVM solutions.

> **NOTE**
> A DSVM has many VM sizes, but you will need to select “HDD” and an NC* size.

The Data Science Virtual Machine(DSVM) has many popular data science and deep learning tools already installed and configured. A list of tools available is located here.

Create a Data Science VM

In order to create the Data Science VMs navigate to the Azure Government Portal and click "New" to access the Azure Government Marketplace.

- Provision a Windows Data Science VM

- Provision a Linux Data Science VM

Using a Data Science VM

- Ten things you can do on the Windows Data Science VM
- Ten things you can do on the Linux Data Science VM

Next steps

For supplemental information and updates, subscribe to the Microsoft Azure Government Blog.

Kubernetes on Azure Government

5/29/2018 • 2 minutes to read • Edit Online

This article describes how to deploy a Kubernetes cluster to Azure Government using acs-engine.

Prerequisites

- Download acs-engine. Make sure you download **release v.0.14.0 or greater**, previous versions don't work properly with Azure Government.
- Download kubectl.

Define your Kubernetes cluster configuration

1. Download the sample acs-engine `apimodel.json` for Kubernetes 1.8.

   > **NOTE**
   > Only use Kubernetes version 1.8 or greater to if you intend to use Azure Files with Azure Government.

2. Modify the following values in your `apimodel.json` file:

   - `dnsPrefix`: The dns name you want for the cluster. For example, `contoso` will result in `https://contoso.usgovvirginia.cloudapp.usgovcloudapi.net`
   - `keyData`: The public SSH key to SSH into the Kubernetes cluster. See How to create and use an SSH public and private key pair for Linux VMs in Azure.
   - `clientId` and `secret`: The client ID and secret for the Azure AD service principal that Kubernetes uses to communicate with Azure Government (for example, to create load balancers, request public IPs and access Azure storage).

     > **NOTE**
     > Make sure this service principal is set up with the correct scope. See ACS-Engine: Service Principals.

Deploy your Kubernetes cluster using acs-engine

1. Obtain your Subscription ID. The subscription ID is available in the Azure portal, via Powershell and via the Azure CLI:

   Via Azure CLI:

   ```
   az cloud set --n AzureUSGovernment
   az login
   az account list
   ```

2. Use acs-engine to deploy your template to Azure Government. This operation takes up to 30 minutes for three nodes.

```
acs-engine deploy --azure-env AzureUSGovernmentCloud --location usgovvirginia --subscription-id
<YOUR_SUBSCRIPTION_ID> --api-model apimodel.json
```

Connect to your Kubernetes cluster

1. Configure your kubectl context. This configuration is per bash session. You'll need to run this command for every session:

   ```
   export KUBECONFIG=$(pwd)/_output/<DNS-PREFIX>/kubeconfig/kubeconfig.usgovvirginia.json
   ```

   Alternatively, you can replace your kubectl config file for your configuration to persist across sessions.

   > **WARNING**
   > Any existing configurations will be replaced.

   ```
   cp $(pwd)/_output/<DNS-PREFIX>/kubeconfig/kubeconfig.usgovvirginia.json ~/.kube/config
   ```

2. Test your kubectl connectivity with the cluster

   ```
   kubectl get pods
   ```

3. (Optional) Deploy a PHP Guestbook application with Redis in your Kubernetes cluster

References

- Microsoft Azure Container Service Engine - Kubernetes
- Configure Access to Multiple Clusters

Next steps

- Subscribe to the Azure Government blog
- Get help on Stack Overflow by using the "azure-gov" tag
- Give feedback or request new features via the Azure Government feedback forum

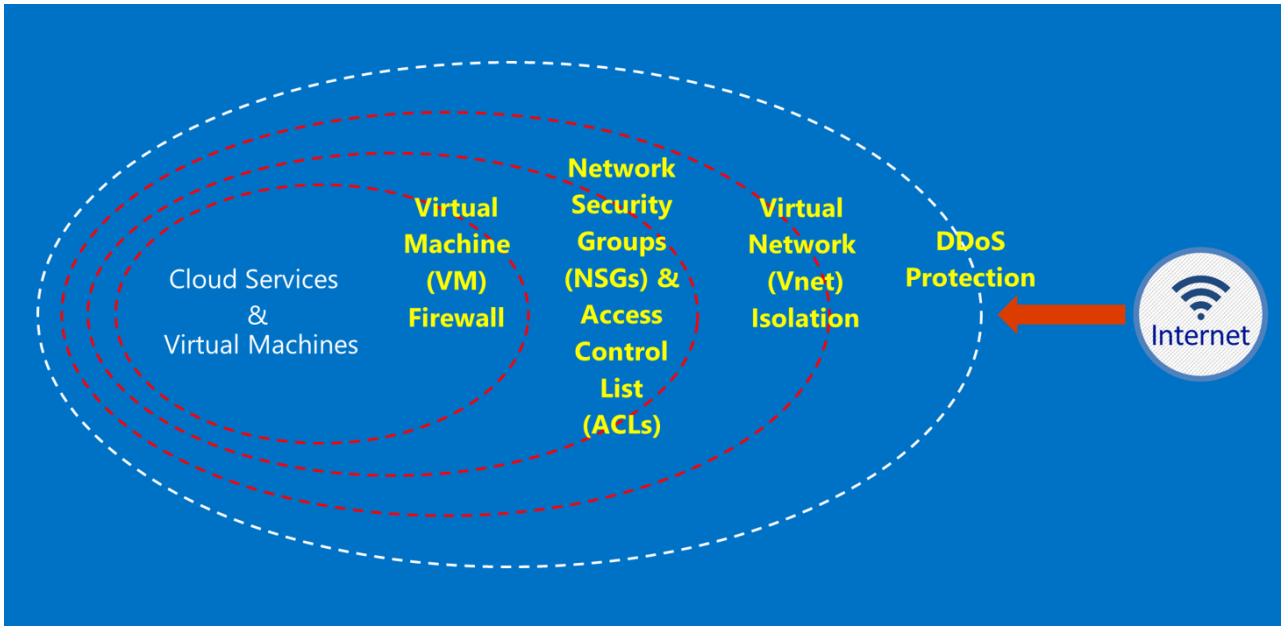
Azure Government security

7/12/2018 • 8 minutes to read • Edit Online

Azure Government provides a range of features and services that you can use to build cloud solutions to meet your regulated/controlled data needs. A compliant customer solution is nothing more than the effective implementation of out-of-the-box Azure Government capabilities, coupled with a solid data security practice.

When you host a solution in Azure Government, Microsoft handles many of these requirements at the cloud infrastructure level.

The following diagram shows the Azure defense-in-depth model. For example, Microsoft provides basic cloud infrastructure DDOS, along with customer capabilities such as security appliances for customer-specific application DDOS needs.



This page outlines the foundational principles for securing your Services and applications, providing guidance and best practices on how to apply these principles; in other words, how customers should make smart use of Azure Government to meet the obligations and responsibilities that are required for a solution that handles ITAR information.

The overarching principles for securing customer data are:

- Protecting data using encryption
- Managing secrets
- Isolation to restrict data access

Protecting customer data using encryption

Mitigating risk and meeting regulatory obligations are driving the increasing focus and importance of data encryption. Use an effective encryption implementation to enhance current network and application security measures—and decrease the overall risk of your cloud environment.

Encryption at rest

The encryption of data at rest applies to the protection of customer content held in disk storage. There are several ways this might happen:

Storage Service Encryption

Azure Storage Service Encryption is enabled at the storage account level, resulting in block blobs and page blobs being automatically encrypted when written to Azure Storage. When you read the data from Azure Storage, it will be decrypted by the storage service before being returned. Use this to secure your data without having to modify or add code to any applications.

Client-Side encryption

Client-Side Encryption is built into the Java and the .NET storage client libraries, which can utilize Azure Key Vault APIs, making this straightforward to implement. Use Azure Key Vault to obtain access to the secrets in Azure Key Vault for specific individuals using Azure Active Directory.

Encryption in transit

The basic encryption available for connectivity to Azure Government supports Transport Level Security (TLS) 1.2 protocol, and X.509 certificates. Federal Information Processing Standard (FIPS) 140-2 Level 1 cryptographic algorithms are also used for infrastructure network connections between Azure Government datacenters. Windows Server 2016, Windows 10, Windows Server 2012 R2, and Windows 8.1, and Azure File shares can use SMB 3.0 for encryption between the VM and the file share. Use Client-Side Encryption to encrypt the data before it is transferred into storage in a client application, and to decrypt the data after it is transferred out of storage.

Best practices for encryption

- IaaS VMs: Use Azure Disk Encryption. Turn on Storage Service Encryption to encrypt the VHD files that are used to back up those disks in Azure Storage, but this only encrypts newly written data. This means that, if you create a VM and then enable Storage Service Encryption on the storage account that holds the VHD file, only the changes will be encrypted, not the original VHD file.
- Client-Side Encryption: This is the most secure method for encrypting your data, because it encrypts it before transit, and encrypts the data at rest. However, it does require that you add code to your applications using storage, which you might not want to do. In those cases, you can use HTTPs for your data in transit, and Storage Service Encryption to encrypt the data at rest. Client-Side Encryption also involves more load on the client—you have to account for this in your scalability plans, especially if you are encrypting and transferring a lot of data.

Protecting customer data by managing secrets

Secure key management is essential for protecting data in the cloud. Customers should strive to simplify key management and maintain control of keys used by cloud applications and services to encrypt data.

Best practices for managing secrets

- Use Key Vault to minimize the risks of secrets being exposed through hard-coded configuration files, scripts, or in source code. Azure Key Vault encrypts keys (such as the encryption keys for Azure Disk Encryption) and secrets (such as passwords), by storing them in FIPS 140-2 Level 2 validated hardware security modules (HSMs). For added assurance, you can import or generate keys in these HSMs.
- Application code and templates should only contain URI references to the secrets (which means the actual secrets are not in code, configuration or source code repositories). This prevents key phishing attacks on internal or external repos, such as harvest-bots in GitHub.
- Utilize strong RBAC controls within Key Vault. If a trusted operator leaves the company or transfers to a new group within the company, they should be prevented from being able to access the secrets.

For more information Azure Key Vault public documentation.

Understanding isolation

Isolation in Azure US Government is achieved through the implementation of trust boundaries, segmentation, and containers to limit data access to only authorized users, services, and applications. Azure US Government supports

environment, and per-customer isolation controls and capabilities.

Environment isolation

The Azure Government multitenant cloud platform environment (FedRAMP HIGH / DoD L4) is a physical instance that is an Internet standards-based autonomous system separately administered from the rest of Microsoft's networks. This Autonomous System (AS) as defined by IETF RFC 4271 is a set of switches and routers under a single technical administration, using an interior gateway protocol and common metrics to route packets within the AS, and using an exterior gateway protocol to route packets to other ASs though a single and clearly defined routing policy. In addition, the specific DoD named region pairs (DoD L5) within Azure Government are geographically separated physical instances of compute, storage, SQL, and supporting services that store and/or process customer content (in accordance with DoD SRG 5.2.2.3 requirements).

The isolation of the Microsoft Azure Government environment is achieved through a series of physical and logical controls, and associated capabilities that include: physically isolated hardware, physical barriers to the hardware using biometric devices and cameras; conditional access (RBAC, workflow), specific credentials and multifactor authentication for logical access; infrastructure for Azure Government is located within the United States.

Within the Microsoft Azure Government network, internal network system components are isolated from other system components through implementation of separate subnets and access control policies on management interfaces. Azure Government does not directly peer with the public internet or with the Microsoft corporate network. Microsoft Azure Government directly peers to the commercial Microsoft Azure network which has routing and transport capabilities to the Internet and the Microsoft Corporate network. Azure Government limits its exposed surface area by leveraging additional protections and communications capabilities of our commercial Azure network. In addition, Microsoft Azure Government Express Route (ER) leverages peering with our customer's networks over non-Internet private circuits to route ER customer "DMZ" networks using specific Border Gateway Protocol (BGP)/AS peering as a trust boundary for application routing and associated policy enforcement.

Per-Customer isolation

Separation between customers/tenants is an essential security mechanism for the entire Azure Government multitenant cloud platform. Microsoft Azure Government provides base per-customer or tenant isolation controls including Isolation of Hypervisor, Root OS, and Guest VMs, Isolation of Fabric Controllers, Packet Filtering, and VLAN Isolation.

Customer/tenants can manage their isolation posture to meet individual requirements through network access control and segregation through virtual machines, virtual networks, VLAN isolation, ACLs, load balancers and IP filters. Additionally, customers/tenants can further manage isolation levels for their resources across subscriptions, resource groups, virtual networks, and subnets. The customer/tenant logical isolation controls help prevent one tenant from interfering with the operations of any other customer/tenant.

Screening

The recently announced FedRAMP High and Department of Defense (DoD) Impact Level 4 accreditation. This has raised the security and compliance bar across the Azure Government environment.

We are now screening all our operators at National Agency Check with Law and Credit (NACLC) as defined in section 5.6.2.2 of the DoD Cloud Computing Security Requirements Guide (SRG):

> **NOTE**
>
> The minimum background investigation required for CSP personnel having access to Level 4 and 5 information based on a "noncritical-sensitive" (e.g., DoD's ADP-2) is a National Agency Check with Law and Credit (NACLC) (for "noncritical-sensitive" contractors), or a Moderate Risk Background Investigation (MBI) for a "moderate risk" position designation.

The following table summarizes our current screening for Azure Government operators:

AZURE GOV SCREENINGS AND BACKGROUND CHECKS	DESCRIPTION
US citizenship	Verification of US citizenship.
Microsoft cloud background check (every two years)	Social Security number search, criminal history check, Office of Foreign Assets Control list (OFAC), Bureau of Industry and Security list (BIS), Office of Defense Trade Controls Debarred Persons list.
National Agency Check with Law and Credit (NACLC) (every five years)	Adds fingerprint background check against FBI databases. For additional information, go to the Office Personnel Management Site.
Criminal Justice Information Services (CJIS)	CJIS is a state, local and FBI government screening which processes fingerprint records and validates criminal histories on operational staff who could be provided access to critical criminal justice information (CJI) data. Each state does their own background check and subsequent approval of all employees with potential access to CJI.

For Azure operations personnel, the following access principles apply:

- Duties are clearly defined, with separate responsibilities for requesting, approving and deploying changes.
- Access is through defined interfaces that have specific functionality.
- Access is just-in-time (JIT), and is granted only on a per-incident basis or for a specific maintenance event, and always for a limited duration.
- Access is rule-based, with defined roles that are only assigned the permissions required for troubleshooting.

Screening standards include the validation of US citizenship of all Microsoft support and operational staff before access is granted to Azure Government-hosted systems. Support personnel who need to transfer data use the secure capabilities within Azure Government. Secure data transfer requires a separate set of authentication credentials to gain access. For example, to access system metadata, operations personnel use specific web-based internal management tools, read-only APIs, and JIT elevation.

Next steps

For supplemental information and updates please subscribe to the Microsoft Azure Government Blog.

Azure Government compliance

6/27/2018 • 3 minutes to read • Edit Online

Azure Security and Compliance Blueprint

Azure Security and Compliance Blueprints are designed to facilitate the secure and compliant use of Azure for government agencies and third-party providers building on behalf of government. Azure Government customers may leverage Azure Government's FedRAMP JAB Provisional Authority to Operate (P-ATO) or DoD Provisional Authorization (PA), reducing the scope of customer-responsibility security controls in Azure-based systems. Inheriting security control implementations from Azure Government allows customers to focus on control implementations specific to their IaaS, PaaS, or SaaS environments built in Azure.

> **NOTE**
> Within the context of Azure Security and Compliance Blueprints, "customer" references the organization building directly within Azure. Azure customers may include third-party ISVs building on behalf of government or government agencies building directly in Azure.

Azure Security and Compliance Blueprint Customer Responsibilities Matrix

The Azure Security and Compliance Blueprint Customer Responsibilities Matrix (CRM) is designed to aid Azure Government customers implementing and documenting system-specific security controls implemented within Azure. The CRM explicitly lists all NIST SP 800-53 security control requirements for FedRAMP and DISA baselines that include a customer implementation requirement. This includes controls with a shared responsibility between Azure and Azure customers, as well as controls that must be fully implemented by Azure customers. Where appropriate, controls are delineated at a control sub-requirement granularity to provide more specific guidance.

The CRM format is designed for utility and is conducive to focused documentation of only the customer portions of implemented security controls.

For example, control AC-1 requires documented access control policies and procedures for the system seeking an ATO. For this control, Microsoft has internal Azure-specific policies and procedures regarding access control mechanisms used to manage the Azure infrastructure and platform. Customers must also create their own access control policies and procedures used within their specific system built in Azure. The CRM documents control parts AC-1a, which requires the policies and procedures to include specific content, as well as AC-1b, which requires customers to review and update these documents on an annual basis.

The CRM is available as Microsoft Excel workbook for the FedRAMP Moderate and High baselines, the DISA Cloud Computing SRG L4 and L5 baselines, and the NIST Cybersecurity Framework (CSF).

The CRM is available for download from the Service Trust Portal.

Azure Security and Compliance Blueprint System Security Plan

The Azure Security and Compliance Blueprint System Security Plan (SSP) template is customer-focused and designed for use in developing an SSP that documents both customer security control implementations as well as controls inherited from Azure. Controls which include a customer responsibility, contain guidance on documenting control implementation with a thorough and compliant response. Azure inheritance sections document how

security controls are implemented by Azure on behalf of the customer.

The SSP is available for the FedRAMP Moderate and High baselines, and the DISA Cloud Computing SRG L4 and L5 baselines.

The SSP is available for download from the Service Trust Portal.

Azure Security and Compliance Blueprint implementation guidance

Azure Security and Compliance Blueprint implementation guidance is designed to help cloud solution architects and security personnel understand how Azure Government services and features can be deployed to implement a subset of customer-responsibility FedRAMP and DoD security controls. An array of documentation, tools, templates, and other resources are available to guide the secure deployment of Azure services and features. Azure resources can be deployed using Azure Resource Manager template building blocks, community-contributed Azure Quickstart Templates, or through use of customer-authored JSON-based Resource Manager templates. The architecture of a basic deployment in Azure includes compute, networking, and storage resources. This implementation guidance addresses how these resources can be deployed in ways that help meet security control implementation requirements.

The NIST SP 800-53 implementation guidance is available for download from the Service Trust Portal.

General Data Protection Regulation (GDPR) Data Subject Requests (DSR) on Azure Government

Azure government customers can submit DSR starting May 25, 2018 by going to Azure Portal Help + Support and submit a **New Support Request** with the following information:

- **Issue Type:** Subscription Management
- **Problem Type:** Security and Compliance Request
- **Category:** Privacy Blade and GDPR requests
- **Description:** *{Include list of all users within the tenant}*

For more information about how Microsoft can help you with the GDPR, see Get Started: Support for GDPR Accountability in the Service Trust Portal.

Next steps

For inquiries related to Azure Security and Compliance Blueprints, FedRAMP, DoD, or Agency ATO processes, or other compliance assistance; or to provide feedback, email AzureBlueprint@microsoft.com.

Microsoft Trust Center

Microsoft Azure Government Blog

Planning identity for Azure Government applications

6/14/2018 • 8 minutes to read • Edit Online

Microsoft Azure Government provides the same ways to build applications and manage identities as Azure Public. Azure Government customers may already have an Azure Active Directory (Azure AD) Public tenant or may create a tenant in Azure AD Government. This article provides guidance on identity decisions based on the application and location of your identity.

Identity models

Before determining the identity approach for your application, you need to know what identity types are available to you. There are three types: On-Premises Identity, Cloud Identity, and Hybrid Identity.

ON-PREMISES IDENTITY	CLOUD IDENTITY	HYBRID IDENTITY
On-Premises Identities belong to on-premises Active Directory environments that most customers use today.	Cloud identities originate, only exist, and are managed in Azure AD.	Hybrid identities originate as on-premises identities, but become hybrid through directory synchronization to Azure AD. After directory synchronization they exist both on-premises and in the cloud, hence hybrid.

> **NOTE**
> Hybrid comes with deployment options (Synchronized Identity, Federated Identity, etc.) that all rely on directory synchronization and mostly define how identities are authenticated as discussed in Choose a Hybrid Identity Solution.

Selecting identity for an Azure Government application

When building any Azure application, a developer must first decide on the authentication technology:

- **Applications using modern authentication** – Applications using OAuth, OpenID Connect, and/or other modern authentication protocols supported by Azure Active Directory. An example is a newly developed application built using PaaS technologies (**for example**, Web Sites, Cloud Database as a Service, etc.)
- **Apps using legacy authentication protocols (Kerberos/NTLM)** – Applications typically migrated from on-premises (**for example**, Lift-n-Shift).

Based on this decision there are different considerations when building in Azure Government.

Applications using modern authentication in Azure Government

Integrating Applications with Azure Active Directory shows how you can use Azure AD to provide secure sign-in and authorization to your applications. This process is the same for Azure Public and Azure Government once you choose your identity authority.

Choosing your identity authority

Azure Government applications can use Azure AD Government identities, but can you use Azure AD Public identities to authenticate to an application hosted in Azure Government? Yes! Since you can use either identity authority, you need to choose which to use:

- **Azure AD Public** – Commonly used if your organization already has an Azure AD Public tenant to support Office 365 (Public or GCC) or another application.

- **Azure AD Government** - Commonly used if your organization already has an Azure AD Government tenant to support Office 365 (GCC High or DoD) or are creating a new tenant in Azure AD Government.

Once decided, the special consideration is where you perform your app registration. If you choose Azure AD Public identities for your Azure Government application, you must register the application in your Azure AD Public tenant. Otherwise, if you perform the app registration in the directory the subscription trusts (Azure Government) the intended set of users cannot authenticate.

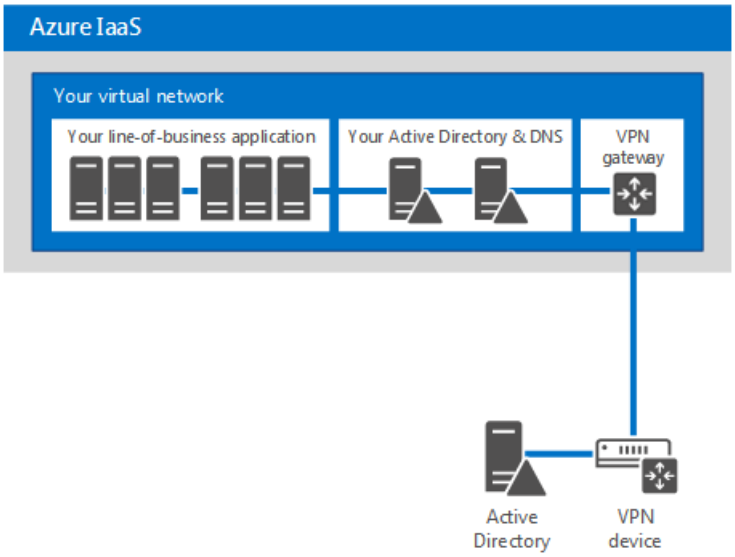
> **NOTE**
> Applications registered with Azure AD only allow sign-in from users in the Azure AD tenant the application was registered in. If you have multiple Azure AD Public tenants, it's important to know which is intended to allow sign-ins from. If you intend to allow users to authenticate to the application from multiple Azure AD tenants the application must be registered in each tenant.

The other consideration is the identity authority URL. You need the correct URL based on your chosen authority:

- **Azure AD Public** = login.microsoftonline.com
- **Azure AD Government** = login.microsoftonline.us

Applications using legacy authentication protocols (Kerberos/NTLM)

Supporting IaaS cloud-based applications dependent on NTLM/Kerberos authentication requires On-Premises Identity. The aim is to support logins for line-of-business application and other apps that require Windows Integrated authentication. Adding Active Directory domain controllers as virtual machines in Azure IaaS is the typical method to support these types of apps, shown in the following figure:



> **NOTE**
> The preceding figure is a simple connectivity example, using site-to-site VPN. Azure ExpressRoute is another and more preferred connectivity option.

The type of domain controller to place in Azure is also a consideration based on application requirements for directory access. If applications require directory write access, deploy a standard domain controller with a writable copy of the Active Directory database. If applications only require directory read access, we recommend deploying a RODC (Read-Only Domain Controller) to Azure instead. Specifically, for RODCs we recommend following the guidance available at Deployment Decisions and Factors for Read-Only DCs.

We have documentation covering the guidelines for deploying AD Domain Controllers and ADFS (AD Federation Services) at these links:

- Guidelines for Deploying Windows Server Active Directory on Azure Virtual Machines
  - Answers questions such as:
    - Is it safe to virtualize Windows Server Active Directory Domain Controllers?
    - Why deploy AD to Azure Virtual Machines?
    - Can you deploy ADFS to Azure Virtual Machines?
- Deploying Active Directory Federation Services in Azure
  - Provides guidance on how to deploy ADFS in Azure.

Identity scenarios for subscription administration in Azure Government

First, see Managing and connecting to your subscription in Azure Government, for instructions on accessing Azure Government management portals.

There are a few important points that set the foundation of this section:

- Azure subscriptions only trust one directory, therefore subscription administration must be performed by an identity from that directory.
- Azure Public subscriptions trust directories in Azure AD Public and Azure Government subscriptions trust directories in Azure AD Government.
- If you have both Azure Public and Azure Government subscriptions, separate identities for both are required.

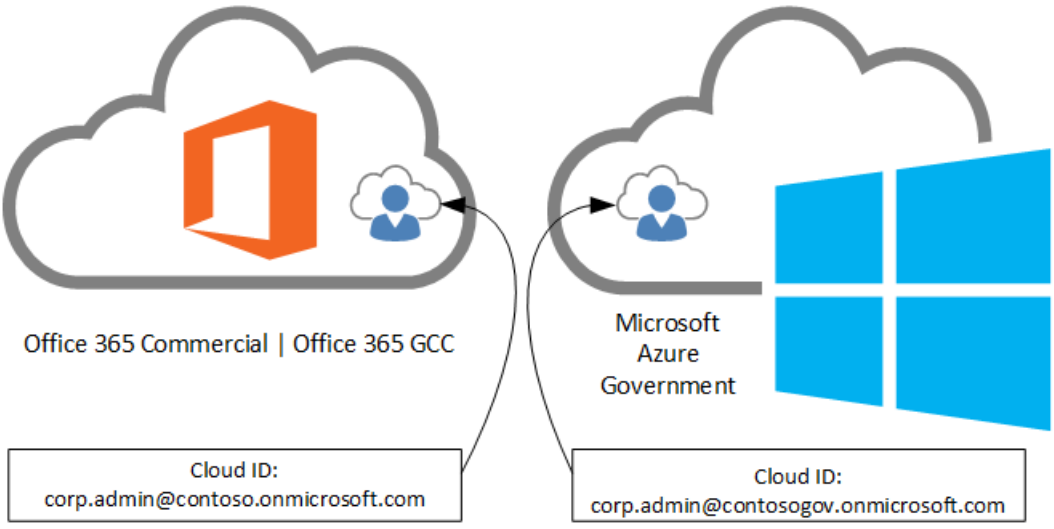
The currently supported identity scenarios to simultaneously manage Azure Public and Azure Government subscriptions are:

- Cloud identities - Cloud identities are used to manage both subscriptions
- Hybrid and cloud identities - Hybrid identity for one subscription, cloud identity for the other
- Hybrid identities - Hybrid identities are used to manage both subscriptions.

A common scenario, having both Office 365 and Azure subscriptions, is conveyed in each of the following scenarios.

Using cloud identities for multi-cloud subscription administration

The following diagram is the simplest of the scenarios to implement.

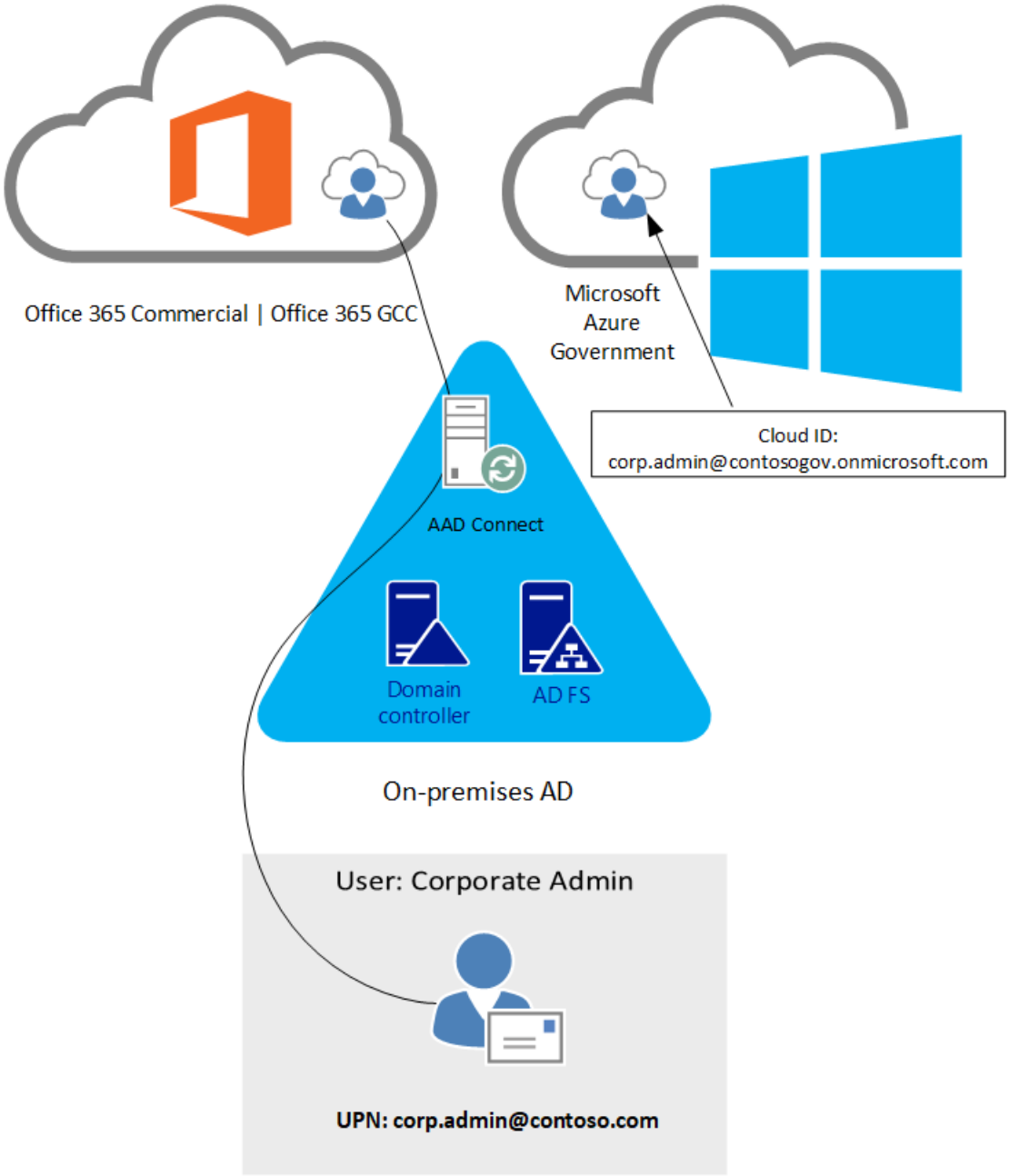


While using cloud identities is the simplest approach, it is also the least secure because passwords are used as an authentication factor. We recommend Azure Multi-Factor Authentication, Microsoft's two-step verification solution, to add a critical second layer of security to secure access to Azure subscriptions when using cloud identities.

See How Azure Multi-Factor Authentication works to learn more about the available methods for two-step verification.

Using hybrid and cloud identities for multi-cloud subscription administration

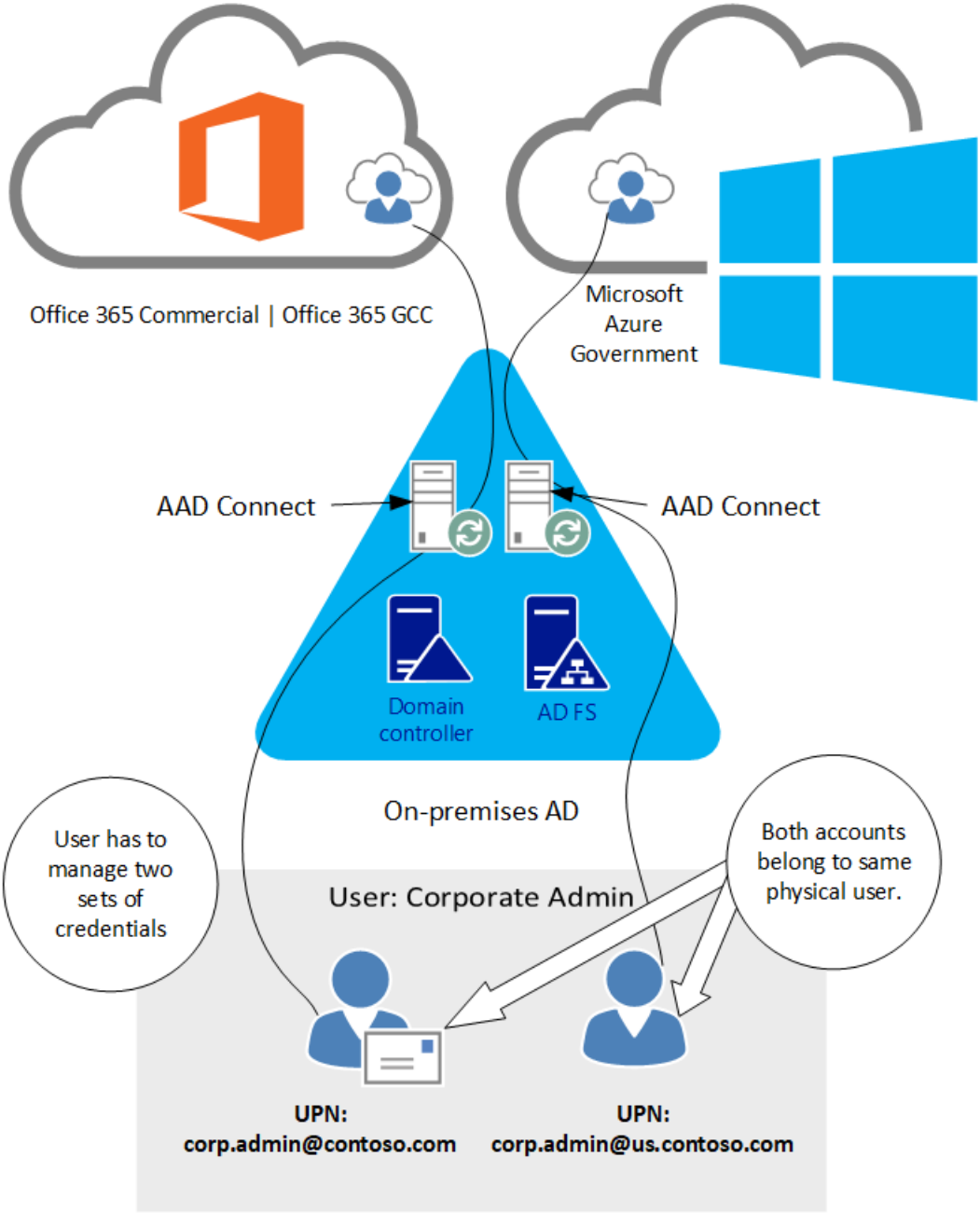
In this scenario, we include administrator identities through directory synchronization to the Public tenant while cloud identities are still used in the government tenant:



Using hybrid identities for administrative accounts allows the use of smartcards (physical or virtual). Government agencies using Common Access Cards (CACs) or Personal Identity Verification (PIV) cards benefit from this approach. In this scenario ADFS serves as the identity provider and implements the two-step verification (**for example**, smart card + PIN).

Using hybrid identities for multi-cloud subscription administration

In this scenario, hybrid identities are used to administrator subscriptions in both clouds:



Frequently asked questions

Why does Office 365 GCC use Azure AD Public?

The first Office 365 US Government environment, Government Community Cloud (GCC), was created when Microsoft had a single cloud directory. The Office 365 GCC environment was designed to use Azure AD Public while still adhering to controls and requirements outlined in FedRAMP Moderate, CJIS (Criminal Justice Information Services), IRS 1075, and National Institute of Standards and Technology (NIST) publication 800-171. Azure Government, with its Azure AD infrastructure was created later. By that time, GCC had already secured the necessary compliance certifications (for example, FedRAMP Moderate and CJIS) to meet Federal, State, and Local government requirements while serving hundreds of thousands of customers. Now, many Office 365 GCC customers have two Azure AD tenants: one from the Azure AD subscription that supports Office 365 GCC and the other from their Azure Government subscription with identities in both.

How do I identify an Azure Government tenant?
Here's a way to find out using your browser of choice:

- Obtain your tenant name (**for example**, contoso.onmicrosoft.com) or a domain name registered to your Azure AD tenant (**for example**, contoso.gov).

- Navigate to https://login.microsoftonline.com/<domainname>/.well-known/openid-configuration
  - <domainname> can either be the tenant name or domain name you gathered in step 1.
  - **An example URL**: https://login.microsoftonline.com/contoso.onmicrosoft.com/.well-known/openid-configuration
- The result posts back to the page in attribute/value pairs using Java Script Object Notation (JSON) format that resembles:

```
{
  "authorization_endpoint":"https://login.microsoftonline.com/b552ff1c-edad-4b6f-b301-
5963a979bc4d/oauth2/authorize",
  "tenant_region_scope":"USG"
}
```

- If the **tenant_region_scope** attribute's value is **USG** as shown, you have yourself an Azure Government tenant.
  - The result is a JSON file that's natively rendered by more modern browsers such as Microsoft Edge, Mozilla Firefox, and Google Chrome. Internet Explorer doesn't natively render the JSON format so instead prompts you to open or save the file. If you must use Internet Explorer, choose the save option and open it with another browser or plain text reader.
  - The tenant_region_scope property is exactly how it sounds, regional. If you have a tenant in Azure Public in North America, the value would be **NA**.

If I'm an Office 365 GCC customer and want to build solutions in Azure Government do I need to have two tenants?
Yes, the Azure AD Government tenant is required for your Azure Government Subscription administration.

If I'm an Office 365 GCC customer that has built workloads in Azure Government, where should I authenticate from, Public or Government?
See "Choosing your Identity Authority" earlier in this article.

I'm an Office 365 customer and have chosen hybrid identity as my identity model. I also have several Azure subscriptions. Is it possible to use the same Azure AD tenant to handle sign-in for Office 365, applications built in my Azure subscriptions, and/or applications reconfigured to use Azure AD for sign-in?

Yes, see How Azure subscriptions are associated with Azure Active Directory to learn more about the relationship between Azure subscriptions and Azure AD. It also contains instructions on how to associate subscriptions to the common directory of your choosing.

Can an Azure Government subscription be associated with a directory in Azure AD Public?

No, the ability to manage Azure Government subscriptions requires identities sourced from a directory in Azure AD Government.

Next steps

- Check out the Azure Government developer guide and build your first application!
- For supplemental information and updates, subscribe to the Microsoft Azure Government blog.

Azure Government cybersecurity: Monitoring and securing your assets with Log Analytics

4/25/2018 • 5 minutes to read • Edit Online

Cybersecurity in the cloud

A crucial concern for our customers who are moving to the cloud is retaining asset management and security of the Azure Government services that they've deployed to the cloud. Virtual machine firewalls need to be configured correctly. Virtual networks need to have the right network security groups applied to them. Access to your assets needs to be locked down at the right time. All these necessary work streams need to be planned, designed, and provisioned to enable a secure infrastructure for your agency to use.

Setting up this kind of environment can be challenging. Onboarding your fleet of servers to any monitoring service is a hard operation to scale, and it can also be challenging to update the monitoring service. Monitoring infrastructure on different cloud providers as well as across the cloud and on-premises is difficult. Finally, keeping your monitoring up-to-date and enabling Azure Application Insights to monitor, detect, alert, and counter cybersecurity threats require time, resources, and computing power.

Azure Log Analytics

Log Analytics, now available in Azure Government, uses hyperscale log search to quickly analyze your data and expose threats in your environment. This article focuses on using Log Analytics which uses hyperscale log search to quickly analyze your data and expose threats in your environment.

Azure Log Analytics can:

- Deploy agents to individual VMs (Linux and Windows) on Azure, other cloud providers, and on-premises.
- Connect your existing logs via an Azure Government storage account or System Center Operations Manager endpoint with existing logging data.

Let's explore how we can get Log Analytics integrated into your fleet and look at some of the out-of-box solutions that address the concerns that we've described here.

Onboarding servers to Log Analytics

The first step in integrating your cloud assets with Log Analytics is installing the Log Analytics agent across log sources. For virtual machines, this is very simple because you can manually download the agent from the Log Analytics portal.

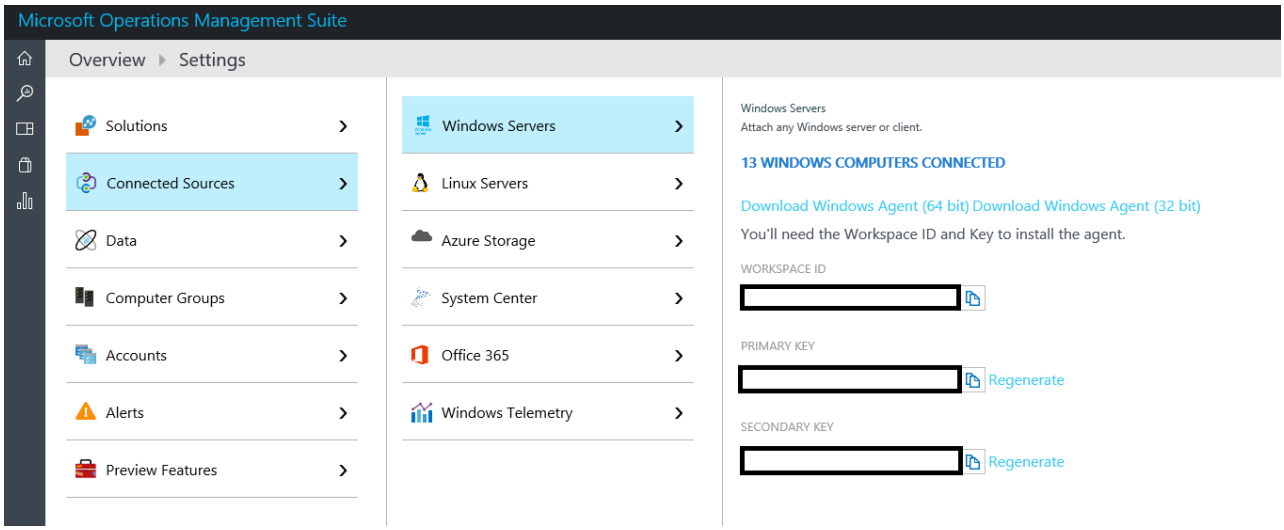


Figure 1: Windows servers connected to Log Analytics

You can connect Azure VMs to Log Analytics directly through the Azure portal. For instructions, see New ways to enable Log Analytics on your Azure VMs.

You can also connect them programmatically or configure the Log Analytics virtual machine extension right into your Azure Resource Manager templates. See the instructions for Windows-based machines at Connect Windows computers to Log Analytics and for Linux-based machines at Connect Linux computers to Log Analytics.

Onboarding storage accounts and Operations Manager to Log Analytics

Log Analytics can also connect to your storage account and/or existing System Center Operations Manager deployments to offer you operations management in hybrid scenarios (across cloud providers or in cloud/on-premises infrastructures).

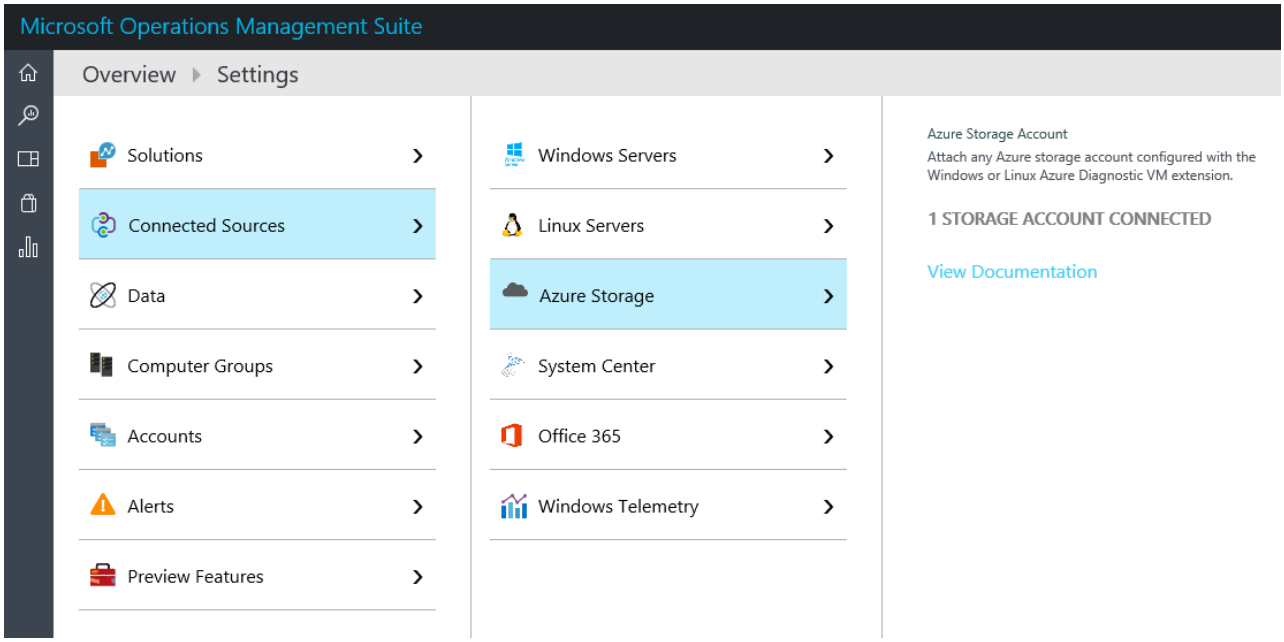


Figure 2: Connecting Azure Storage and Operations Manager to Log Analytics

Log Analytics also supports collecting logging information from other monitoring services like Chef or Puppet. Furthermore, for Azure deployments, we have VMs with Log Analytics-enabled Azure Resource Manager templates so you can deploy compute and onboard to your Log Analytics workspace at the same time.

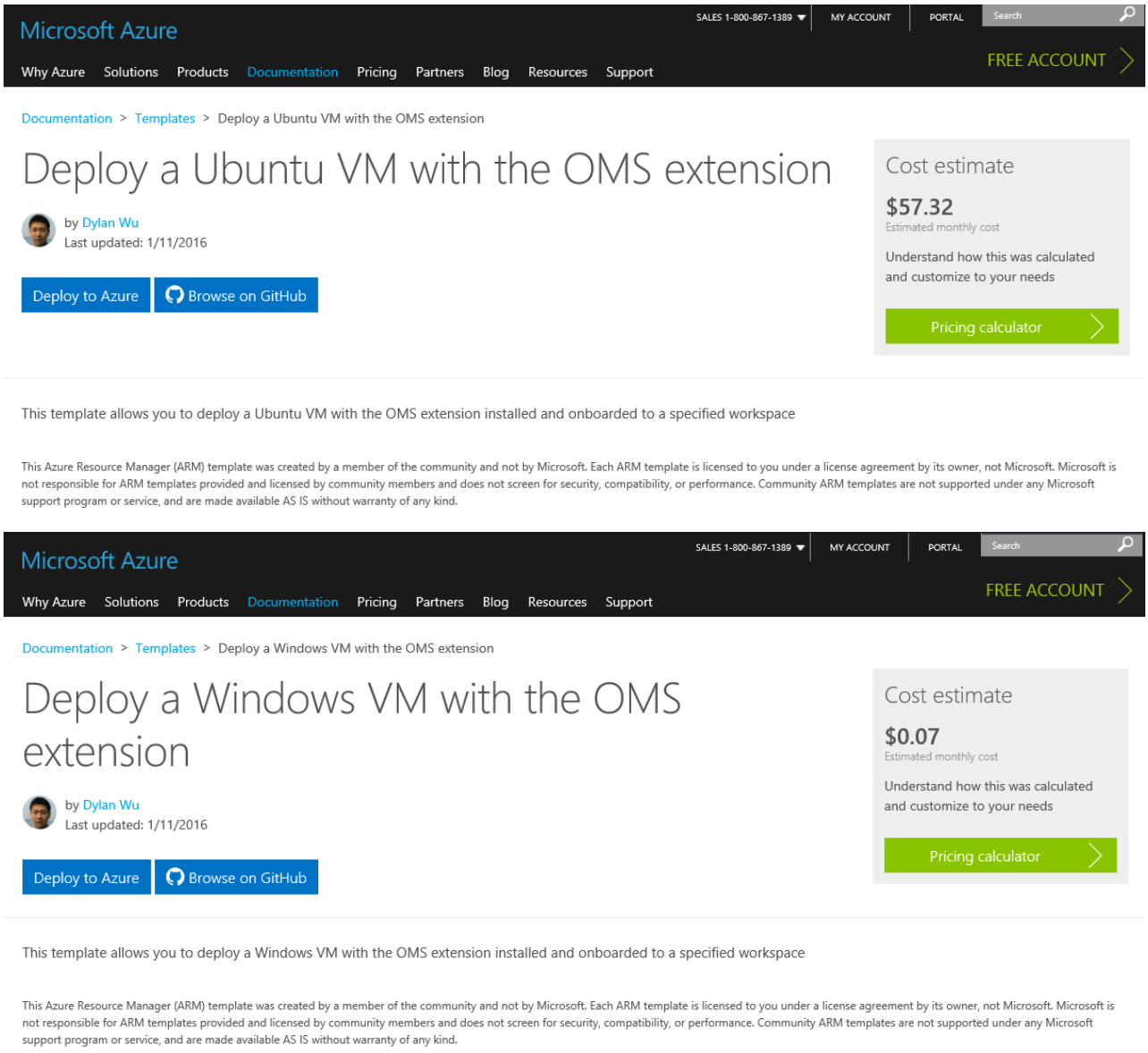


Figure 3: Azure Resource Manager templates for Azure VMs with Log Analytics VM extension

Information about setting up Log Analytics with your existing Operations Manager implementation on-premises can be found in Connect Operations Manager to Log Analytics.

Applying intelligence through management solutions

Now that you have various sources for logging data, you have to make sense of all this data.

Log Analytics, at its core, is a log search service that lets you write powerful queries to quickly search across thousands or even millions of logs. However, discovering the issues that you need to write queries can be difficult.

Enter Log Analytics solutions. These are packs of queries that are natively integrated with Log Analytics to proactively give you insights into your Log Analytics-managed fleet.

On the theme of cyber security, I briefly discuss three cybersecurity scenarios that Log Analytics can solve out of the box for you.

Antimalware assessment

Antimalware assessments give you a canned set of queries, notifications, and monitoring dashboards to tell you at a glance how well your fleet is protected against malware.

This dashboard gives you a list of four things:

- Any servers that have active and/or remediated threats.

- Currently detected threats.
- Computers that aren't being sufficiently protected. Log Analytics finds this information by crawling the logs of your computers to look for any site of FWs that are being opened, or for improperly configured rules in common web browsers.
- Analysis of how your protected servers are being protected, for example by native Windows OS virus protection or a solution such as System Center Endpoint Protection.

For example, you can see that the following threat was caught and automatically triaged by System Center:

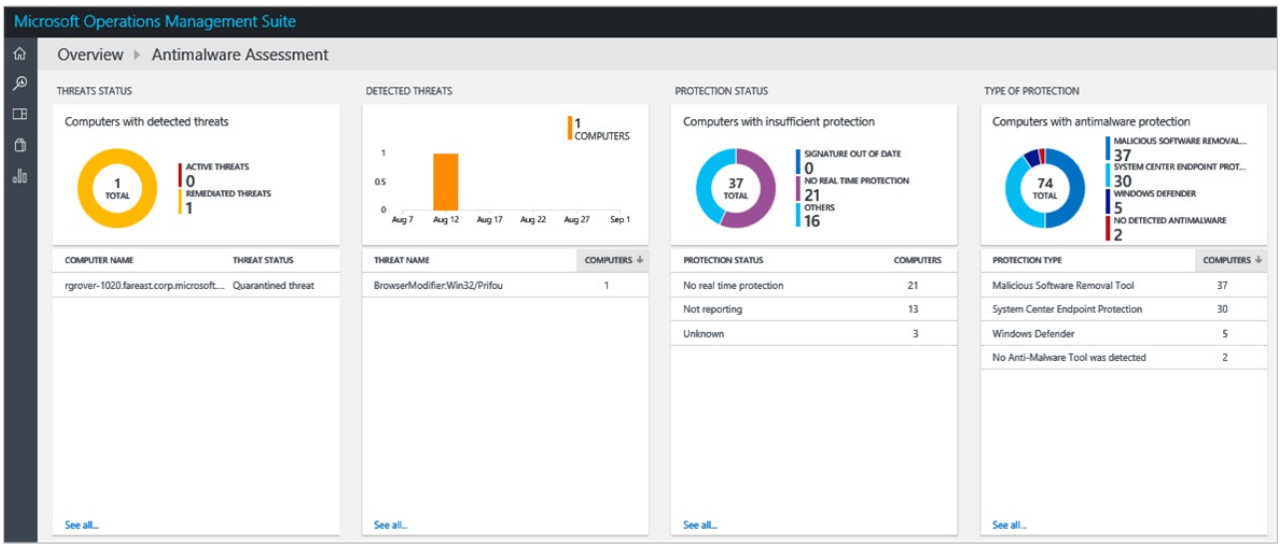


Figure 4: Log Analytics antimalware assessment solution

More information about antimalware assessment can be found in the article Malware assessment solution in Log Analytics.

Identity and access

Another common cybersecurity scenario in the cloud revolves around credential compromise. Not only does your cloud subscription have credentials, but each individual VM has a user and/or secret (usually a certificate or password) that's associated with it.

Log Analytics organizes all sign-in attempts in your fleet and buckets them depending on type (remote, local, username, and so on). For example, in the following example, I can see a large amount of unsuccessful sign-in attempts from largely random strings as usernames. This indicates that it's highly likely that my computers have been exposed and not properly protected by firewalls and access control lists.

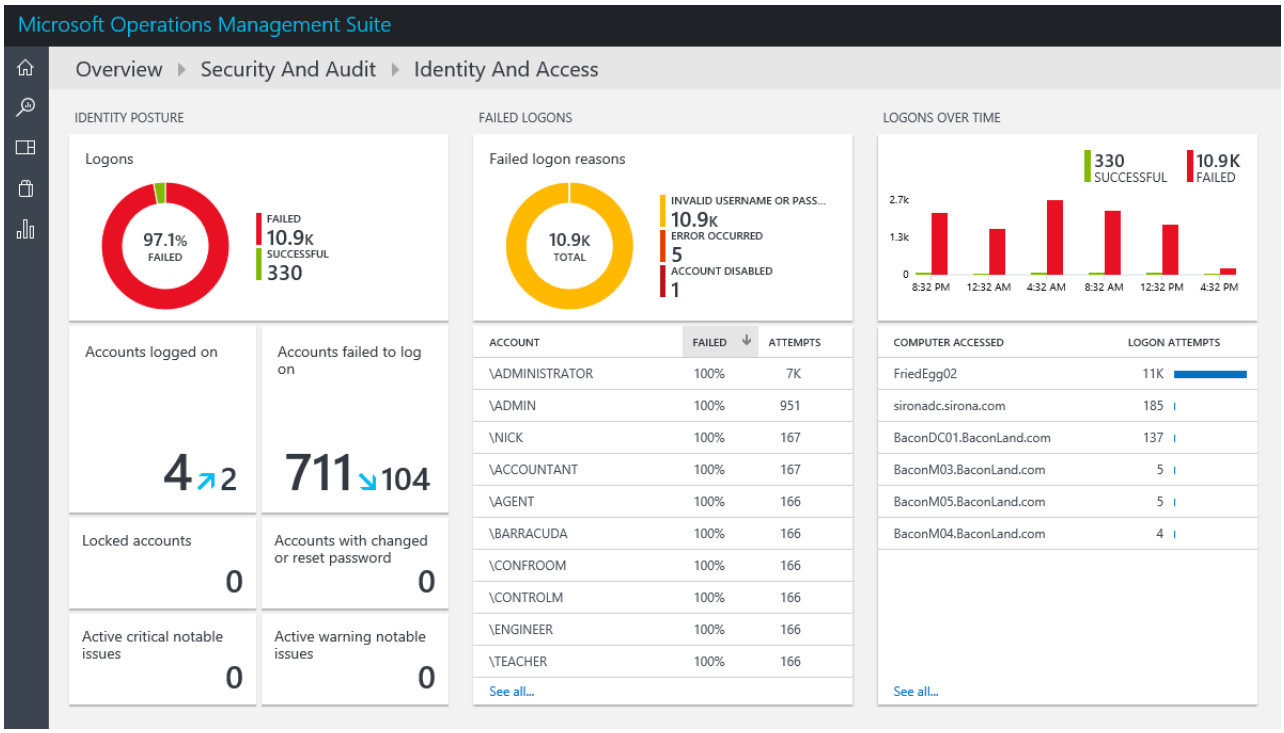


Figure 5: 97.3% sign-ins failed in the last 24 hours

Threat intelligence

Log Analytics also provides protection against malicious insider scenarios, when there's a security compromise inside your organization and a malicious user is trying to exfiltrate data.

Log Analytics threat intelligence looks at all the network logs on your computer and automatically searches for and notifies you about inbound/outbound network connections to known malicious IPs (for example, IP addresses on the unindexed dark net).

For example, in the following screenshot, I can see that there are both inbound and outbound network connections to the People's Republic of China.

By double-clicking the inbound tag, I discover that a Linux VM that is being managed by Log Analytics is making outbound connections to a known dark net IP address in China.

You can also set up alerts to Log Analytics solutions like threat intelligence. In the following screenshot, I've set up an alert so that if Log Analytics detects more than 10 outbound connections to a known malicious IP address, it sends an alert out to me via email. I then configure that alert to fire an Azure Automation job, which is set up to automatically shut down that VM.

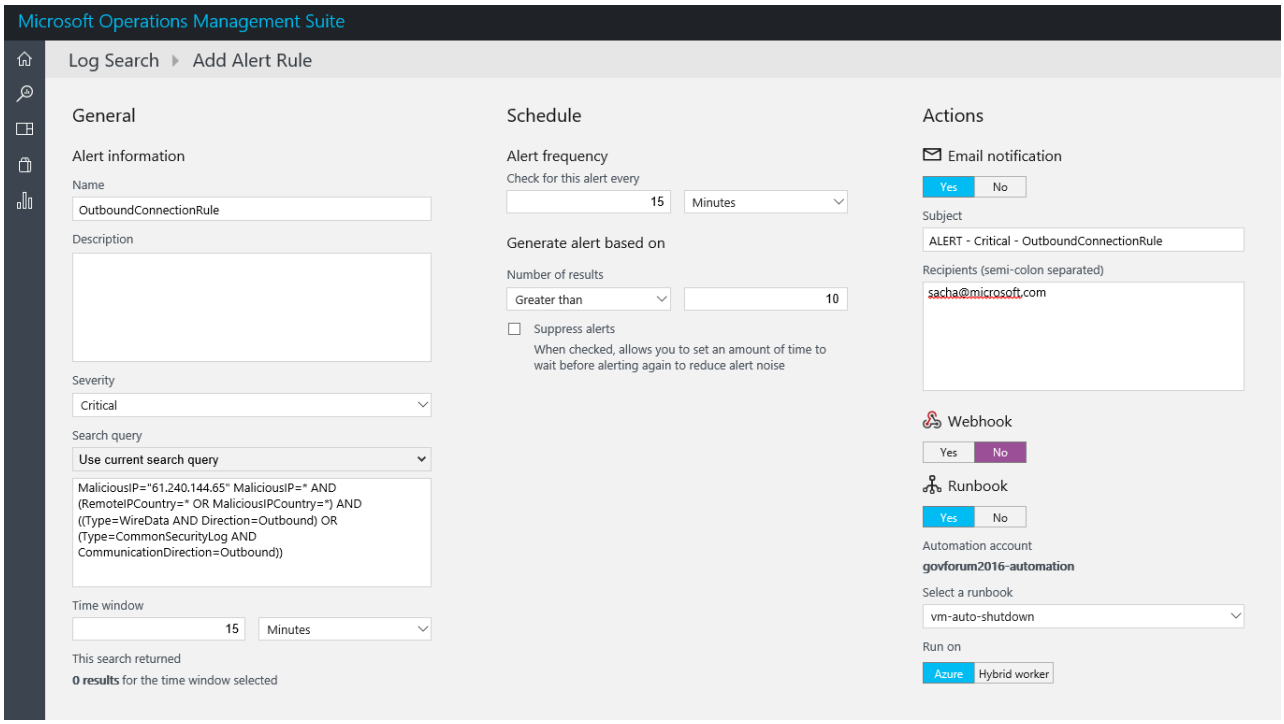


Figure 6: Log Analytics alerts and automation

This is just one example of an out-of-box Log Analytics solution that can be applied to your fleet, whether it's running on Azure, another cloud service provider, or on-premises.

Log Analytics continues to update its machine learning to fight the latest threats automatically for you, and we continue to roll out new solutions to the Azure marketplace as well.

For more information about Log Analytics, see our documentation page.

Managing and connecting to your subscription in Azure Government

9/28/2017 • 2 minutes to read • Edit Online

Azure Government has unique URLs and endpoints for managing your environment. It is important to use the right connections to manage your environment through the portal or PowerShell. Once you are connected to the Azure Government environment, the normal operations for managing a service works if the component has been deployed.

Connecting via the portal

The portal is the primary way that most people connect to Azure Government. To connect, browse to the portal at https://portal.azure.us. The legacy version of the Azure portal can be accessed via https://manage.windowsazure.us.

Subscriptions can be created for your account:

- Via the Azure Government Account Portal - Select the "Subscriptions" option in the top menu and then click on "add subscription".
- Via the Azure Government Portal - Select the "Subscriptions" option in the left side navigation then click on "Add".

Next steps

If you are looking for more information, you can check out:

- PowerShell docs on GitHub
- Step-by-step instruction on connecting to Resource Management
- Azure PowerShell docs on MSDN

For supplemental information and updates, subscribe to the Microsoft Azure Government Blog

Azure Government Marketplace

7/16/2018 • 2 minutes to read • Edit Online

The Azure Government Marketplace helps connect government agencies and partners with independent software vendors (ISVs) and start-ups that are offering their solutions in Azure Government.

> **NOTE**
> For information on making your images available in Azure Government, see the partner onboarding guidelines.

Variations

The Azure Government Marketplace differs from the Azure Marketplace in the following ways:

- Only Bring Your Own License (BYOL) and Pay-as-you-Go (PayGo) images are available.
- A different set of images is available. You can find the list of available images here

> **NOTE**
> Red Hat Enterprise Linux is available in Azure Government with Azure Marketplace billing. This is a special case exception to the above statement about license options in Azure Government.

Enable the Azure Government Marketplace

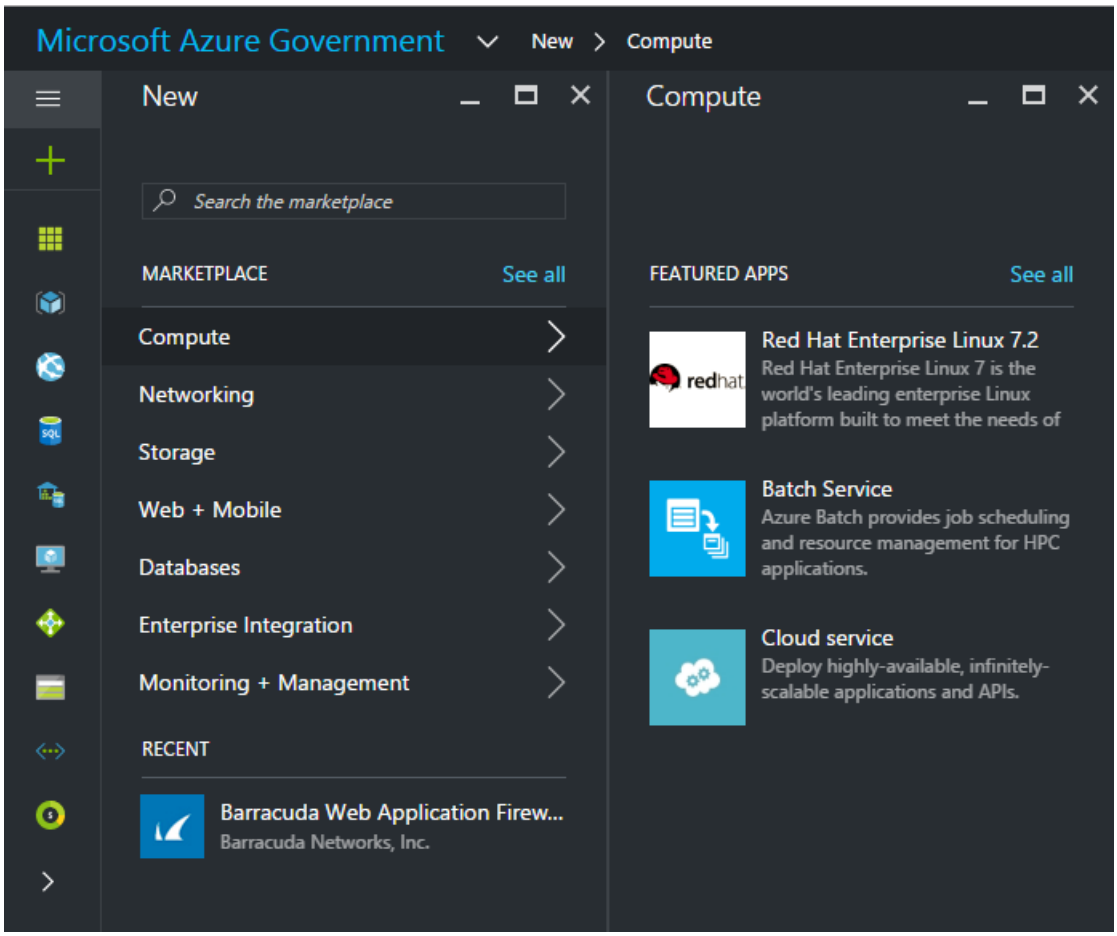
If your subscription is under an Enterprise Agreement (EA), the Azure Government Marketplace must be enabled before you can deploy a Marketplace solution to your subscription.

1. Log in to the Enterprise Account Portal as an Enterprise Administrator
2. Navigate to **Manage**
3. Under **Enrollment Details**, click the pencil icon next to the **Azure Marketplace** line item
4. Toggle **Enabled/Disabled** as appropriate
5. Click **Save**

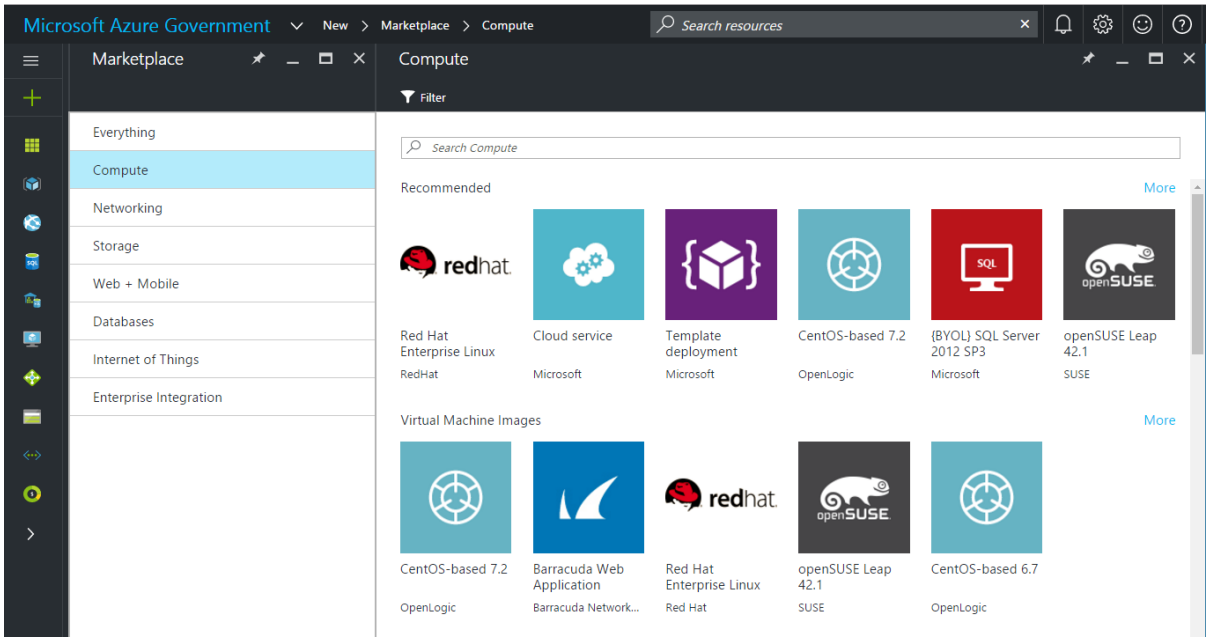
> **NOTE**
> It can take up to 24 hours for the change to take effect.

Deploy a Solution to your Subscription

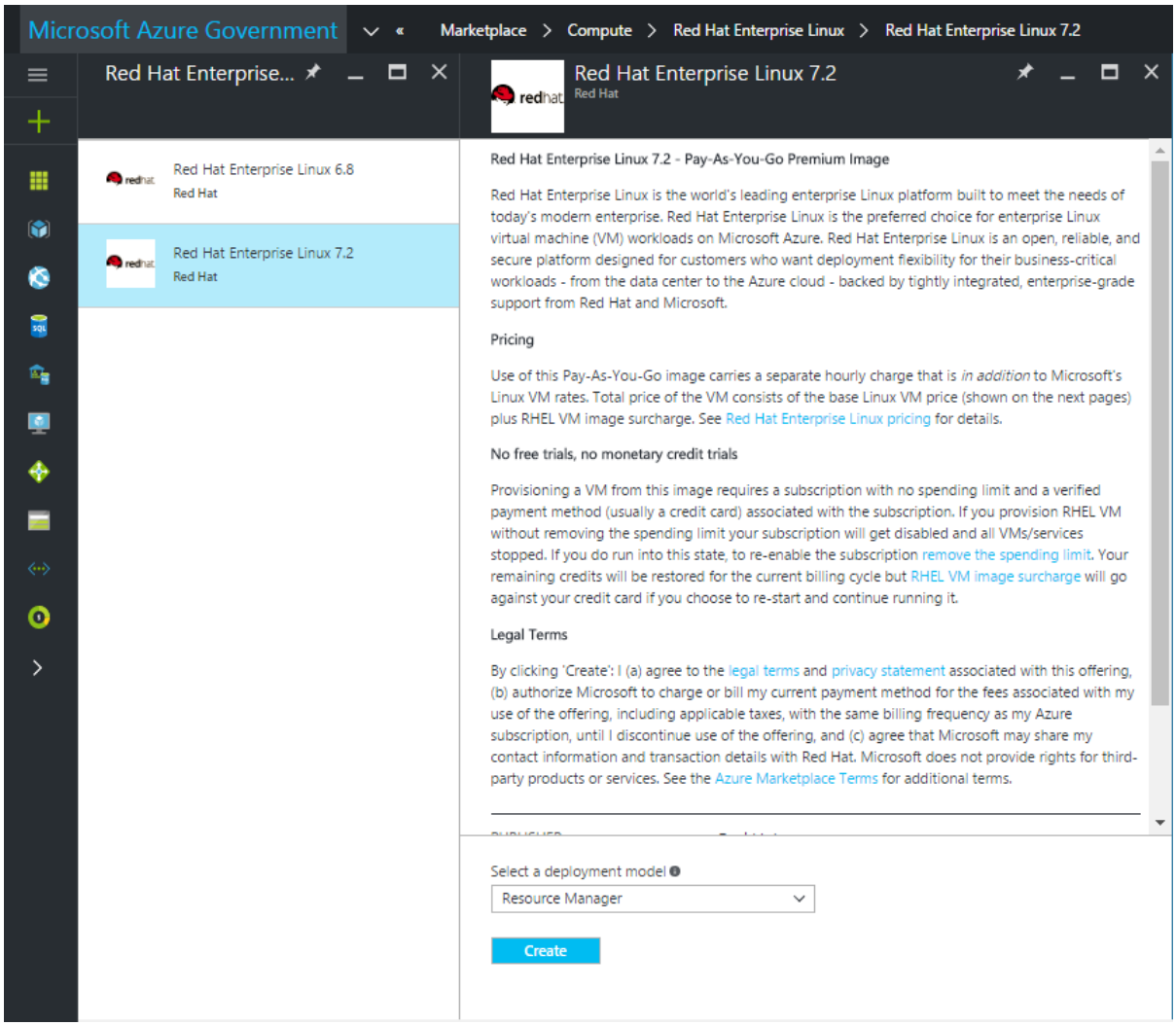
1. **Log in** to the Azure Government portal.
2. Click on **+New**.



3. Browse through different products to find the right one. The marketplace publisher provides a list of certifications as part of the product description to help you make the right choice.

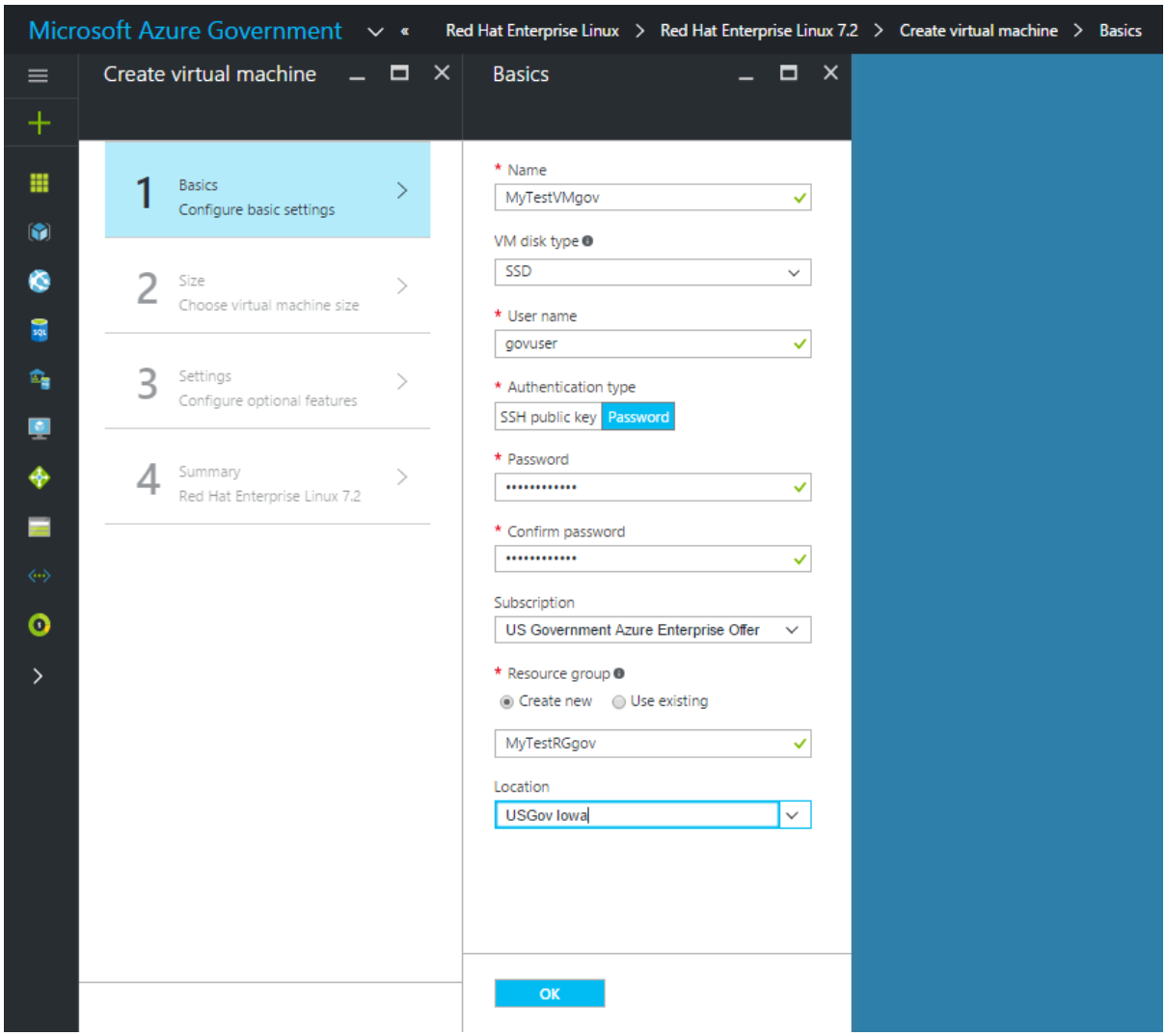


4. Choose an product\image and click **Create**.



5. Enter the required parameters for deployment.

> **NOTE**
>
> In the Location dropdown, only Azure Government locations are visible



6. To start the provisioning process, click **Ok**.

Next steps

- Subscribe to the Azure Government blog
- Get help on Stack Overflow by using the azure-gov tag
- Give feedback or request new features via the Azure Government feedback forum

Connecting with the Azure Government Portal

6/27/2017 • 2 minutes to read • Edit Online

The portal is the primary way that most people connect to Azure Government. To connect, browse to the portal at https://portal.azure.us. The classic Azure portal can be accessed via https://manage.windowsazure.us.

Subscriptions can be created for your account by connecting to https://account.windowsazure.us.

Once you log in, you should see "Microsoft Azure Government" in the upper left of the main navigation bar.

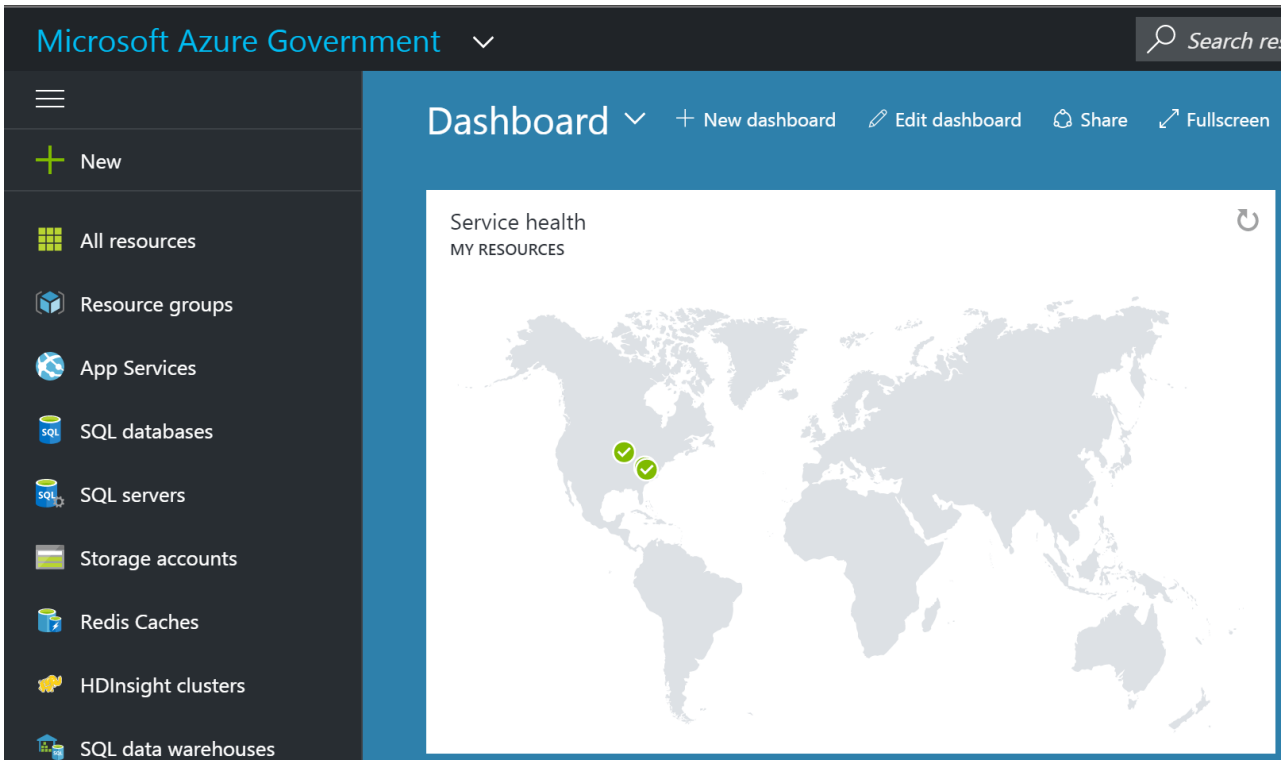


Figure 1: Azure Government Portal

Next steps

For more information about Azure Government, see the following resources:

- Connect to Azure Government with PowerShell
- Connect to Azure Government with Azure CLI

Connect to Azure Government with PowerShell

4/18/2018 • 2 minutes to read • Edit Online

To use Azure PowerShell with Azure Government, you need to connect to Azure Government instead of Azure Public. Azure PowerShell can be used to manage a large subscription through script or to access features that are not currently available in the Azure portal. If you have used PowerShell in Azure Public, it is mostly the same. The differences in Azure Government are:

- Specifying Azure Government as the *environment* to connect to
- Determining Azure Government regions

> **NOTE**
> If you have not used PowerShell yet, check out the Introduction to Azure PowerShell.

Specifying Azure Government as the *environment* to connect to

When you start PowerShell, you have to tell Azure PowerShell to connect to Azure Government by specifying an environment parameter. The parameter ensures that PowerShell is connecting to the correct endpoints. The collection of endpoints is determined when you connect log in to your account. Different APIs require different versions of the environment switch:

CONNECTION TYPE	COMMAND
Azure commands	`Connect-AzureRmAccount -EnvironmentName AzureUSGovernment`
Azure Active Directory commands	`Connect-AzureAD -AzureEnvironmentName AzureUSGovernment`
Azure (Classic deployment model) commands	`Add-AzureAccount -Environment AzureUSGovernment`
Azure Active Directory (Classic deployment model) commands	`Connect-MsolService -AzureEnvironment UsGovernment`

You may also use the `Environment` switch when connecting to a storage account using `New-AzureStorageContext` and specify `AzureUSGovernment`.

If you are curious about the available environments across Azure, you can run:

```
Get-AzureRMEnvironment

Get-AzureEnvironment # For classic deployment model
```

Determining Azure Government regions

Once you are connected, there is one additional difference – The regions used to target a service. Every Azure cloud has different regions. You can see them listed on the service availability page. You normally use the region in the `Location` parameter for a command.

There is one catch. The Azure Government region display names have different formatting than their common

names:

COMMON NAME	DISPLAY NAME	LOCATION NAME
US Gov Virginia	`USGov Virginia`	`usgovvirginia`
US Gov Iowa	`USGov Iowa`	`usgoviowa`
US Gov Texas	`USGov Texas`	`usgovtexas`
US Gov Arizona	`USGov Arizona`	`usgovarizona`
US DoD East	`USDoD East`	`usdodeast`
US DoD Central	`USDoD Central`	`usdodcentral`

NOTE

There is no space between `US` and `Gov` or `US` and `DoD` when using the `Location` parameter.

NOTE

As is the case with PowerShell for Azure Public, you can use either the Display Name or the Location Name for the `Location` parameter.

If you ever want to validate the available regions in Azure Government, you can run the following commands and print the current list:

```
Get-AzureRMLocation

Get-AzureLocation # For classic deployment model
```

Connect to Azure Government with Azure Command Line Interface (CLI)

9/6/2017 • 2 minutes to read • Edit Online

To use Azure CLI, you need to connect to Azure Government instead of Azure public. The Azure CLI can be used to manage a large subscription through script or to access features that are not currently available in the Azure portal. If you have used Azure CLI in Azure Public, it is mostly the same.

Azure CLI 2.0

There are multiple ways to install the Azure CLI 2.0.

To connect to Azure Government, you set the cloud:

```
az cloud set --name AzureUSGovernment
```

After the cloud has been set, you can continue logging in:

```
az login
```

> **NOTE**
> The above login command is recommended, but for simple Azure AD setups/scenarios you can use the `az login --username your-user-name@your-gov-tenant.onmicrosoft.com` and optionally the `--password` parameter. However, if you have configured Azure AD for federation you need to use `az login` and go through the device login flow.

To confirm the cloud has correctly been set to AzureUSGovernment, run this command:

```
az cloud list
```

or

```
az cloud list --output table
```

and verify that the `isActive` flag is set to `true` for the AzureUSGovernment item.

Connecting via Visual Studio

10/9/2017 • 3 minutes to read • Edit Online

Visual Studio is used by developers to easily manage their Azure subscriptions while building solutions. Visual Studio does not currently allow you to configure a connection to Azure Government in the user interface.

Visual Studio 2017

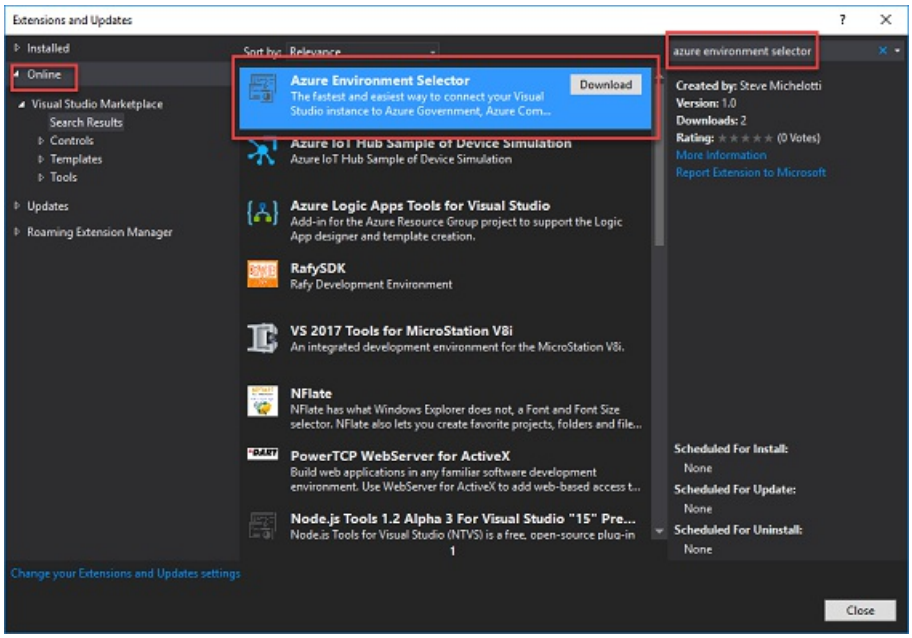
Visual Studio 2017 requires a configuration file for Visual Studio to connect to Azure Government. With this file in place, Visual Studio connects to Azure Government instead of Azure Public. You can do this by using a Visual Studio Extension, or through manual configuration.

Automatically configuring your target using a Visual Studio Extension

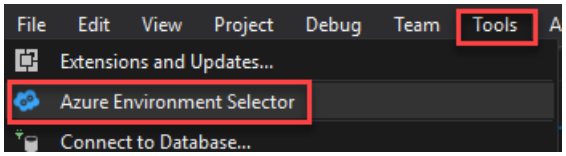
> **NOTE**
> This is the recommended way to connect to Azure Government through Visual Studio.

The Visual Studio extension allows for quickly and easily switching between Azure environments. This can be installed like any other extension in Visual Studio:

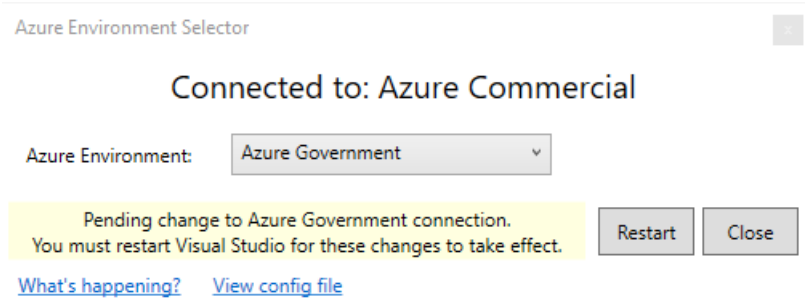
1. In the **Tools** menu, open **Extension and Updates**
2. Click the **Online** tab on the left and **Search "*azure environment selector*"**.
3. **Download** the package, as shown in the screenshot below.



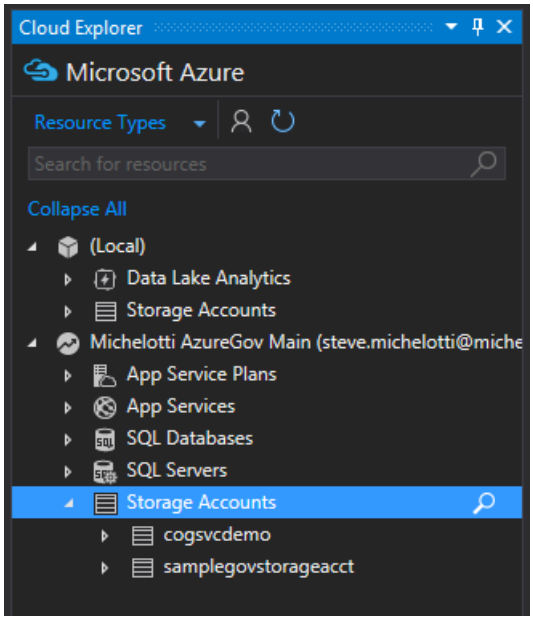
4. **Restart Visual Studio** to complete the installation of the extension.
5. Once Visual Studio restarts, in the **Tools** menu, open the newly available **Azure Environment Selector**:



6. In the Azure Environment Selector dialog, select **Azure Government** from the dropdown:



From here you can restart Visual Studio and the change will take effect. Once Visual Studio restarts, you will now be able to connect to other environments with VS tools such as the Cloud Explorer (shown below connected to Azure Government), Server Explorer, the main Visual Studio login, and the Visual Studio Solution Explorer.



This short video shows the extension in action – walking you through installation and showing how easy it is to connect to Azure Government.

Manually configuring your target

> **NOTE**
> If you have successfully completed the extension installation above, you do not need to complete this section.

Manually creating a configuration file for Azure Government

Create a file named **AadProvider.Configuration.json** with the following content:

```
{
  "AuthenticationQueryParameters": null,
  "AsmEndPoint": "https://management.core.usgovcloudapi.net/",
  "Authority": "https://login.microsoftonline.us/",
  "AzureResourceManagementEndpoint": "https://management.usgovcloudapi.net",
  "AzureResourceManagementAudienceEndpoints": [ "https://management.core.usgovcloudapi.net" ],
  "ClientIdentifier": "872cd9fa-d31f-45e0-9eab-6e460a02d1f1",
  "EnvironmentName": "AzureUSGovernment",
  "GraphEndpoint": "https://graph.windows.net",
  "MsaHomeTenantId": "f8cdef31-a31e-4b4a-93e4-5f571e91255a",
  "NativeClientRedirect": "urn:ietf:wg:oauth:2.0:oob",
  "PortalEndpoint": "https://portal.azure.us/",
  "ResourceEndpoint": "https://management.core.usgovcloudapi.net",
  "ValidateAuthority": true,
  "VisualStudioOnlineEndpoint": "https://app.vssps.visualstudio.com/",
  "VisualStudioOnlineAudience": "499b84ac-1321-427f-aa17-267ca6975798"
}
```

Manually updating Visual Studio for Azure Government

1. Close Visual Studio
2. Place **AadProvider.Configuration.json** created in the previous step into **%localappdata%\.IdentityService\AadConfigurations**. Create this folder if not present.
3. Launch Visual Studio and begin using your Azure Government account.

> **NOTE**
> With the configuration file, only Azure Government subscriptions are accessible. You still see subscriptions that you configured previously but they do not work because Visual Studio is now connected to Azure Government instead of Azure Public. Remove the file to connect to Azure Commercial.

Manually reverting Visual Studio Connection to Azure Government

To enable Visual Studio to connect to Azure Public, you need to remove the configuration file setting that enables connection to Azure Government.

1. Close Visual Studio
2. Delete this folder: **%localappdata%.IdentityService\AadConfigurations**
3. Restart Visual Studio and begin using your Azure Public account.

> **NOTE**
> Once this configuration has been reverted, your Azure Government subscriptions no longer accessible.

Visual Studio 2015

Visual Studio 2015 requires a registry change for Visual Studio to connect to Azure Government. Once this registry key is set, Visual Studio connects to Azure Government instead of Azure Public.

Updating Visual Studio for Azure Government

To enable Visual Studio to connect to Azure Government, you need to update the registry.

1. Close Visual Studio
2. Create a text file named **VisualStudioForAzureGov.reg**
3. Copy and paste the following text into **VisualStudioForAzureGov.reg**:

```
Windows Registry Editor Version 5.00

[HKEY_CURRENT_USER\Software\Microsoft\VSCommon\ConnectedUser]
"AadInstance"="https://login.microsoftonline.us/"
"adaluri"="https://management.core.usgovcloudapi.net"
"AzureRMEndpoint"="https://management.usgovcloudapi.net"
"AzureRMAudienceEndpoint"="https://management.core.usgovcloudapi.net"
"EnableAzureRMIdentity"="true"
"GraphUrl"="graph.windows.net"
```

4. Save and then run the file by double-clicking it. You are prompted to merge the file into your registry.
5. Launch Visual Studio and begin using Cloud Explorer with your Azure Government account.

> **NOTE**
> Once this registry key is set, only Azure Government subscriptions are accessible. You still see subscriptions that you configured previously but they do not work because Visual Studio is now connected to Azure Government instead of Azure Public. See the following section for steps to revert the changes.

Reverting Visual Studio Connection to Azure Government

To enable Visual Studio to connect to Azure Public, you need to remove the registry settings that enable connection to Azure Government.

1. Close Visual Studio
2. Create a text file named **VisualStudioForAzureGov_Remove.reg**
3. Copy and paste the following text into **VisualStudioForAzureGov_Remove.reg**:

```
Windows Registry Editor Version 5.00

[HKEY_CURRENT_USER\Software\Microsoft\VSCommon\ConnectedUser]
"AadInstance"=-
"adaluri"=-
"AzureRMEndpoint"=-
"AzureRMAudienceEndpoint"=-
"EnableAzureRMIdentity"=-
"GraphUrl"=-
```

4. Save and then run the file by double-clicking it. You are prompted to merge the file into your registry.
5. Launch Visual Studio

> **NOTE**
> Once this registry key has been reverted, your Azure Government subscriptions show but are not accessible. They can safely be removed.

Connecting from Visual Studio Team Services

4/25/2018 • 4 minutes to read • Edit Online

The tutorial below will help you set up continuous deployment to your web app running in Azure Government using Visual Studio Team Services (VSTS). Continuous deployment (CD) means starting an automated deployment process whenever a code change is made to your application or whenever a new successful build is available. Visual Studio Team Services is used by teams to configure continuous deployment for their applications hosted in their Azure subscriptions. Refer to CI/CD for newbies for an overview of CI/CD with Team Services.

Release Management in Visual Studio Team Services is a service that enables continuous deployment for various applications. We can use this service for applications running in Azure Government by defining service endpoints for Azure Government.

> **NOTE**
> Visual Studio Team Services itself is not available in Azure Government Clouds. When CD is configured using Team Services to deploy apps to Azure Government clouds, artifact storage, build, and (or) deployment orchestration for the app would execute outside the government cloud.

To learn more about Visual Studio Team Services, click here.

Prerequisites

Before starting this tutorial, you must have the following:

- An active Azure Government subscription. If you don't have an Azure Government subscription, create a free account before you begin.
- Have a VSTS account and Team Project
- Installed and set up Azure Powershell

Create Azure Government app service

Create an App service in your Azure Government subscription. The following steps will set up a CD process to deploy to this Web App.

Set up Build and Source control integration

Follow through one of the quickstarts below to set up a Build for your specific type of app:

- ASP.NET Core app
- ASP.NET 4 app
- Node.js app with Gulp

Generate a service principal

1. Download or copy and paste this powershell script into an IDE or editor.
2. Open up the file and navigate to the `param` parameter. Replace the `$environmentName` variable with AzureUSGovernment." This sets the service principal to be created in Azure Government.
3. Open your Powershell window and run the following command:
   `Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass` Setting this policy enables running local files.

Enter "A" when you are shown the following:

```
PS C:\Users\yujhong>  Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass

Execution Policy Change
The execution policy helps protect you from scripts that you do not trust. Changing the execution policy might expose
you to the security risks described in the about_Execution_Policies help topic at
https:/go.microsoft.com/fwlink/?LinkID=135170. Do you want to change the execution policy?
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is "N"): A
```

4. Navigate to the directory that has the edited script above.
5. Edit the following command with the name of your script and run: `./<name of script file you saved>`
6. The "subscriptionName" parameter can be found by logging into your Azure Government subscription with `Connect-AzureRmAccount -EnvironmentName AzureUSGovernment` and then running `Get-AzureSubscription`.
7. When prompted for the "password" parameter, you can enter your desired password.

```
cmdlet vststest.ps1 at command pipeline position 1
Supply values for the following parameters:
(Type !? for Help.)
subscriptionName: Microsoft Azure Internal Offer - US Government
password: ********
```

8. After providing your Azure Government subscription credentials you should see the following:

> **NOTE**
>
> The Environment variable should be "AzureUSGovernment"

```
subscriptionName: [redacted]
password: ********
Provide your credentials to access Azure subscription [redacted]


Account          : [redacted]
SubscriptionName : [redacted]
SubscriptionId   : [redacted]
TenantId         : [redacted]
Environment      : AzureUSGovernment
```

9. After the script has run you should see your service connection values. Copy these values as we will need them when setting up our endpoint.

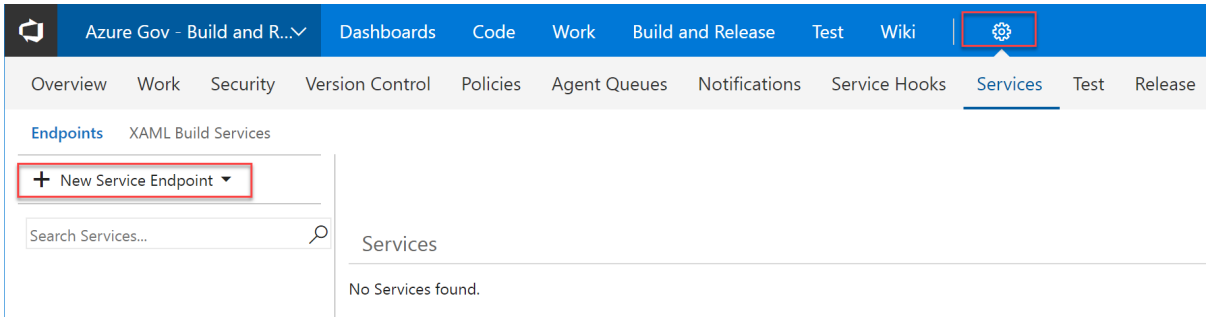
```
SPN role assignment completed successfully

Copy and Paste below values for Service Connection
***************************************************************************
Connection Name: (SPN)
Environment: AzureUSGovernment
Subscription Id: [redacted]
Subscription Name:
Service Principal Id: [redacted]
Service Principal key: <Password that you typed in>
Tenant Id: [redacted]
***************************************************************************
```

Configure the VSTS Endpoint

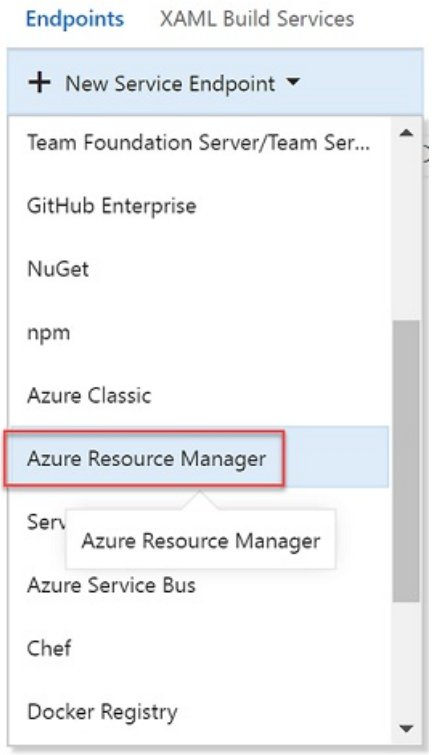
NOTE

Make sure that you add the endpoint soon after running the Powershell script above, as the key expires. If not, you navigate to the Azure Government portal -> AAD -> App registrations -> Add Key

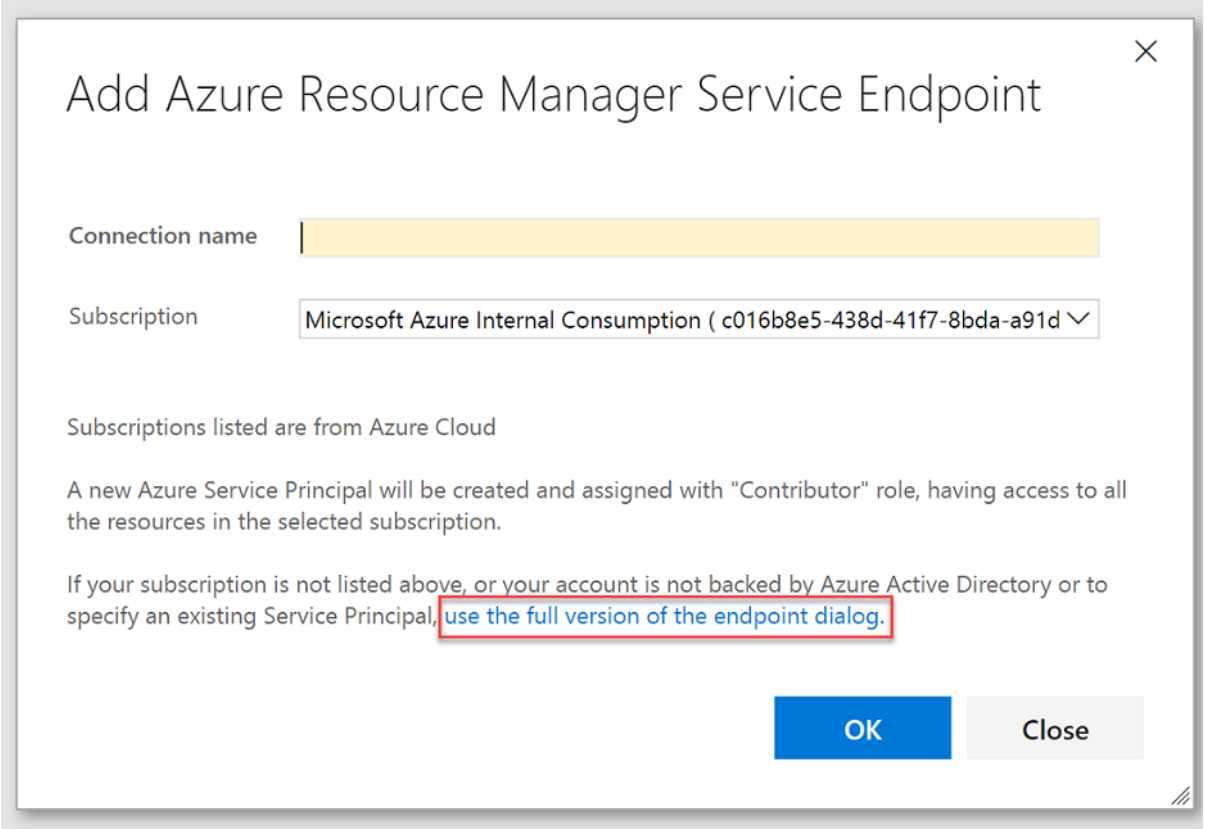
1. Navigate to your Team Project from your Visual Studio Account.
2. Navigate to the Services tab and click on "New Service Endpoint".



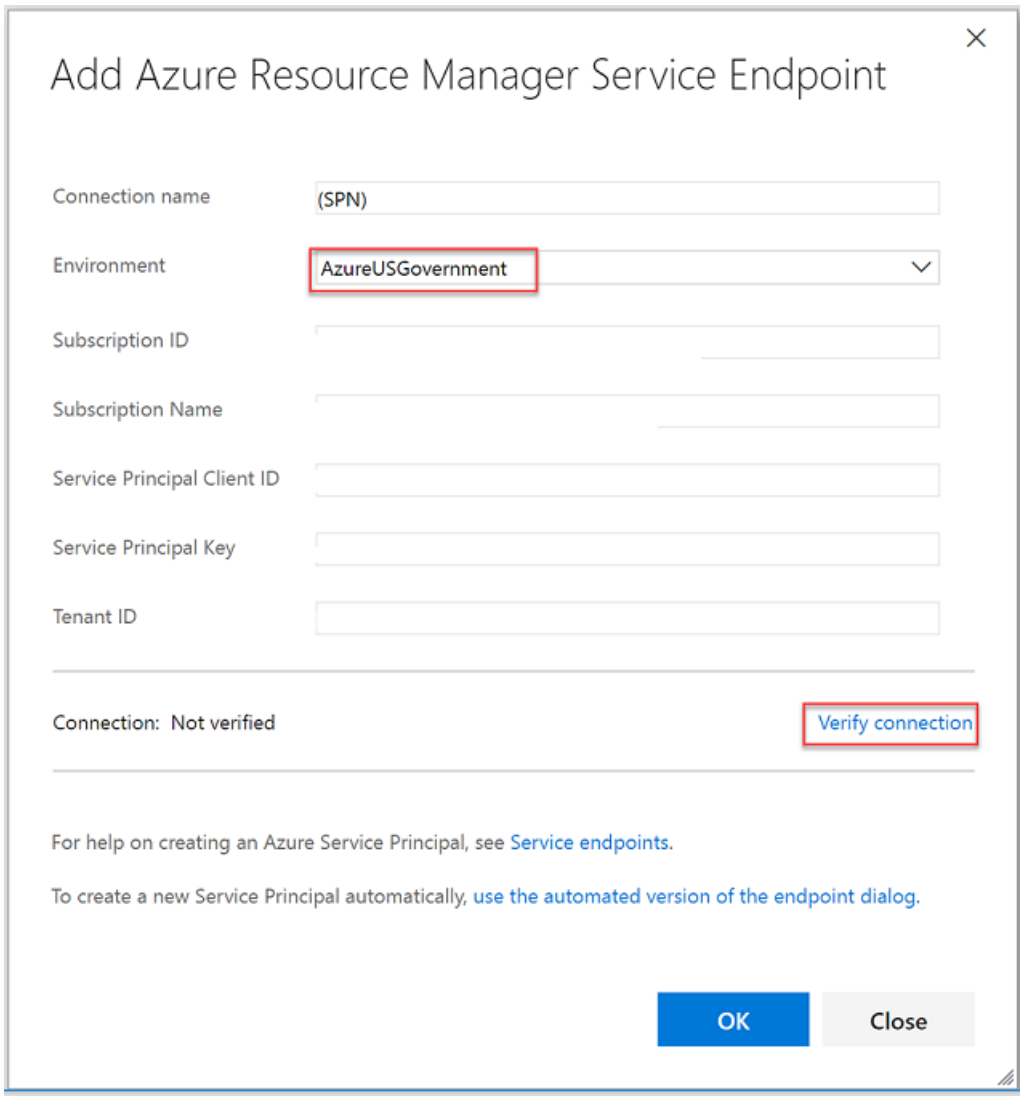
3. Choose "Azure Resource Manager" from the dropdown menu.



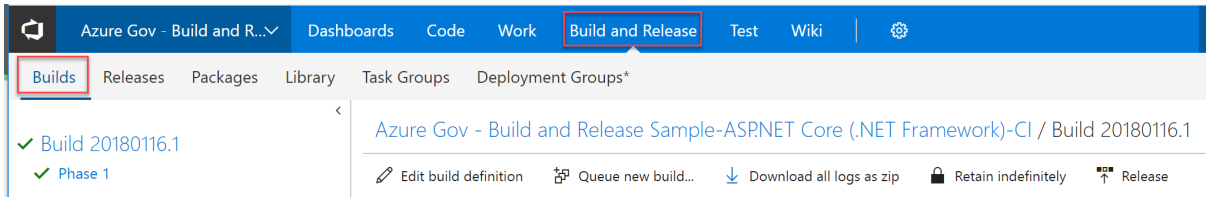
4. Click on the blue text in order to set up the service principal.



5. Fill out the dialog with "AzureUSGovernment" for the "Environment" parameter and choose a friendly name for "Connection Name". Fill out the rest of dialog with the Service Connection values generated in **Step 9 of the "Generate a Service Principal" section above**. Click "Verify Connection" in the bottom right corner, make sure it says "Verified", and then click "Ok".



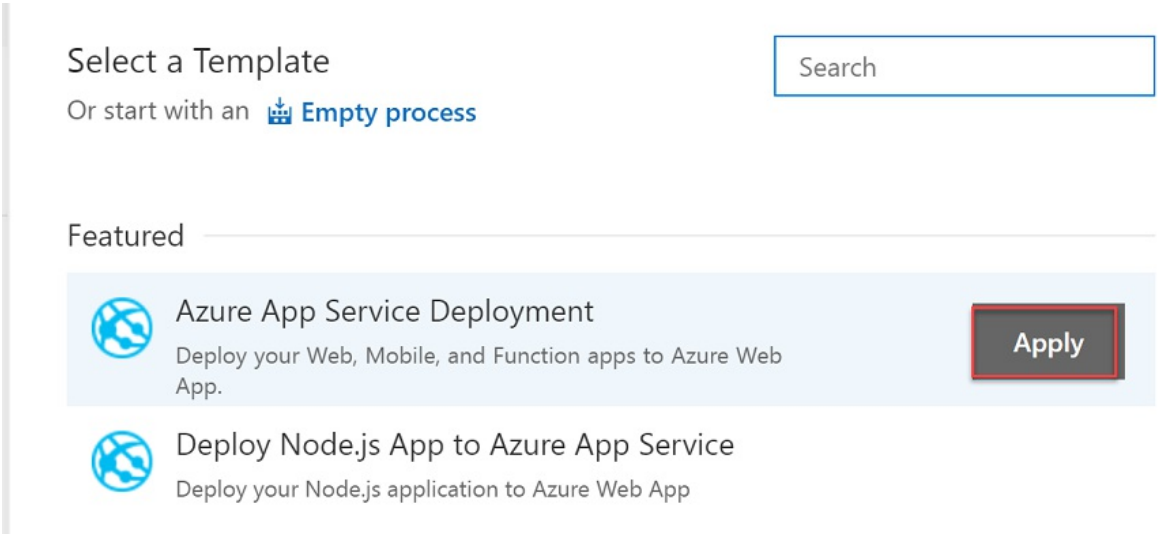
6. Confirm your build has been created successfully.



Define a Release Process

1. After you have completed the steps above, we can now define the release process for our build.
2. Choose the link to the completed build (for example, Build 1634). In the build's Summary tab under Deployments, choose "Create release". This starts a new release definition that's automatically linked to the build definition.

3. Select the Azure App Service Deployment template and choose Next.



   In "Source..." make sure your CI build definition for the Web deploy package is selected as the artifact source.

4. Select the Continuous deployment check box, and then choose Create.
5. Select the Deploy Azure App Service task and configure it as follows:

   - Azure Subscription: Select the endpoint configured earlier
   - App Service Name: the name of the web app (the part of the URL without .azurewebsites.us).
   - Deploy to Slot: make sure this is cleared (the default)
   - Virtual Application: leave blank
   - Web Deploy Package: $(System.DefaultWorkingDirectory)***.zip (the default)
   - Advanced: Take App Offline: If you run into locked .DLL problems when you test the release, as explained below, try selecting this check box.

6. Edit the name of the release definition, choose Save, and choose OK. The default environment is named Environment1, which you can edit by clicking directly on the name.

Now that your pipeline has been constructed, you can deploy changes to your applications in Azure Government.

Q&A

- Do I need a build agent? You need at least one agent to run your deployments. By default, the build and deployment processes are configured to use the hosted agents. Configuring a private agent would limit data sharing outside of Azure Government.
- I use Team Foundation Server on-premises. Can I configure CD on my TFS Server to target Azure Government? Currently, Team Foundation Server cannot be used to deploy to an Azure Government Cloud. This capability will be added in the next update of TFS 2017.

Next steps

- Subscribe to the Azure Government blog
- Get help on Stack Overflow by using the "azure-gov" tag
- Give us feedback or request new features via the Azure Government feedback forum

Connect to Storage in Azure Government

4/9/2018 • 5 minutes to read • Edit Online

Azure Government uses the same underlying technologies as commercial Azure, enabling you to use the development tools you're already familiar with. In order to use these services in Azure Government, you must define different endpoint mappings, as shown below for the Storage service.

Connecting Storage Explorer to Azure Government

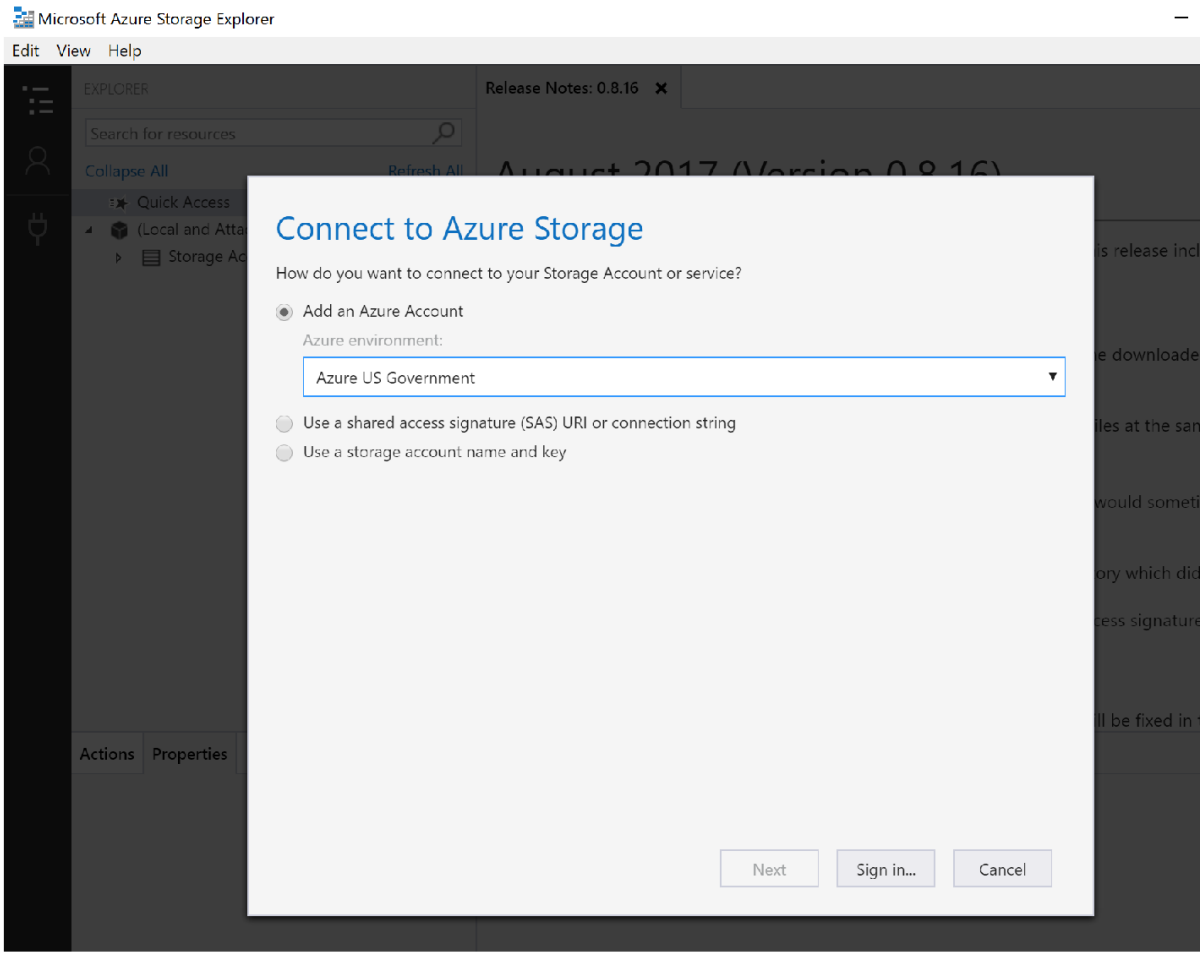
The Microsoft Azure Storage Explorer is a cross-platform tool for working with Azure Storage. Government customers will now be able to take advantage of all the latest features of the Azure Storage Explorer such as being able to create and manage blobs, queues, tables, and file shares.

Prerequisites

- Download and install the latest version of Azure Storage Explorer here.
- Have an active Azure Government subscription. If you don't have an Azure Government subscription, create a free account before you begin.

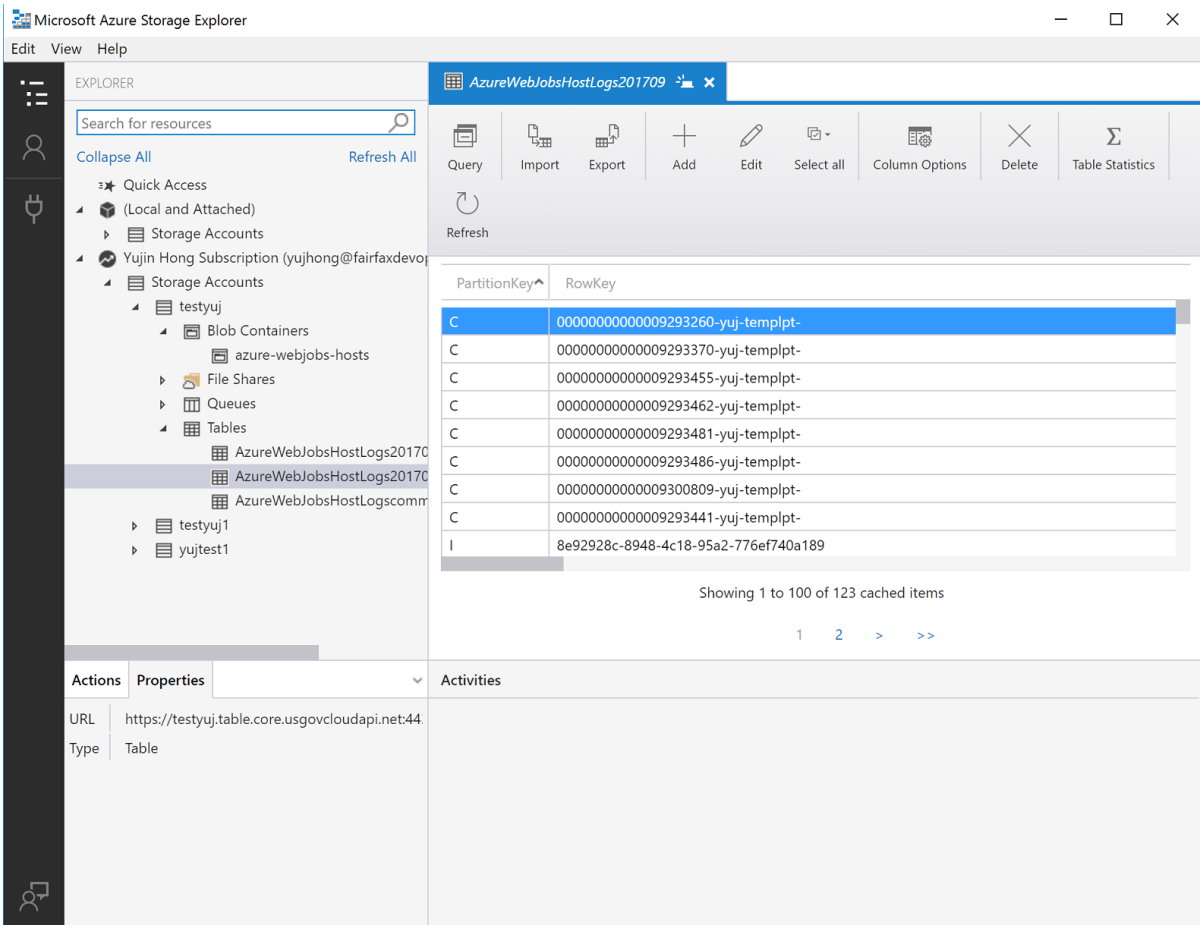
Getting Started with Storage Explorer

1. Open the Azure Storage Explorer desktop application.
2. You will be prompted to add an Azure account; in the dropdown choose the "Azure US Government" option:



3. Log in to your Azure Government account and you will be able to see all of your resources. The Storage Explorer should look similar to the screenshot below. Click on your Storage Account to see the blob

containers, file shares, Queues, and Tables.



For more information on Azure Storage Explorer, click here.

Connecting to the Storage API

Prerequisites

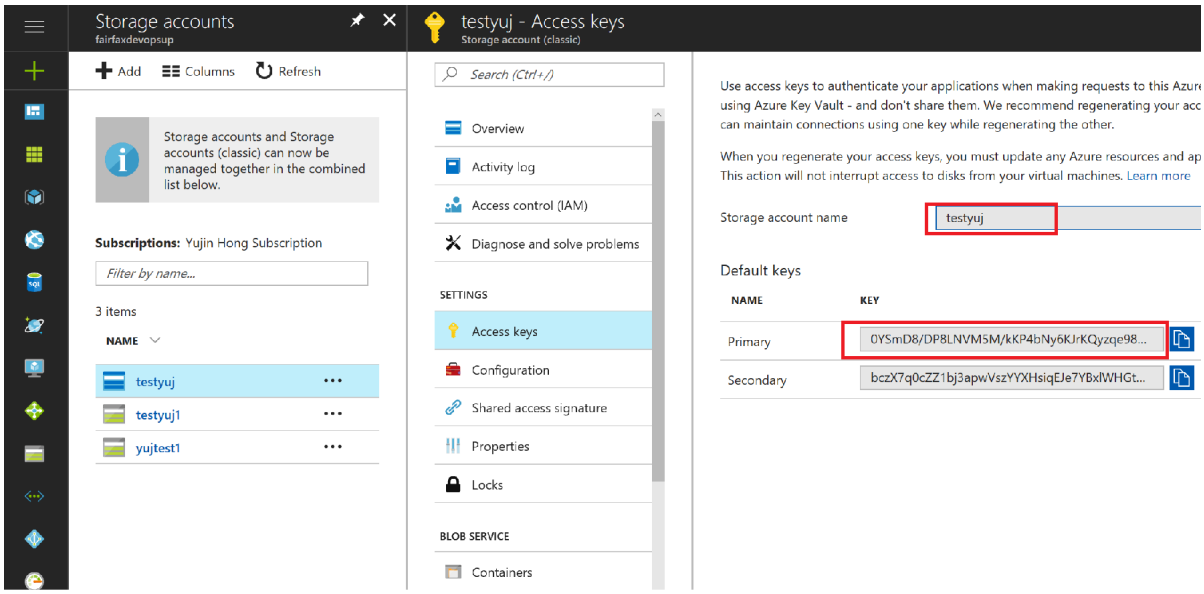
- Have an active Azure Government subscription. If you don't have an Azure Government subscription, create a free account before you begin.
- Download Visual Studio 2017 and Connect to Azure Government.

Getting Started with Storage API

One important difference to note when connecting with the Storage API is that the URL for storage is different than the URL for storage in commercial Azure – specifically, the domain ends with "core.usgovcloudapi.net", rather than "core.windows.net".

These endpoint differences must be taken into account when you connect to storage in Azure Government with C#.

1. Go to the Azure Government portal and select your storage account and then click the "Access Keys" tab:



2. Copy/paste the storage account connection string.

C#

1. Open up Visual Studio and create a new project. Add a reference to the WindowsAzure.Storage NuGet package. This NuGet package contains classes we will need to connect to your storage account.

2. Add these two lines of C# code to connect:

```
var credentials = new StorageCredentials(storageAccountName, storageAccountKey);

var storageAccount = new CloudStorageAccount(credentials, "core.usgovcloudapi.net", useHttps: true);
```

   - Notice on the second line we had to use a particular constructor for the CloudStorageAccount – enabling us to explicitly pass in the endpoint suffix of "core.usgovcloudapi.net". This constructor is the **only difference** your code requires to connect to storage in Azure Government as compared with commercial Azure.

3. At this point, we can interact with storage as we normally would. For example, if we want to retrieve a specific record from our table storage we could do it like this:

```
 var tableClient = storageAccount.CreateCloudTableClient();

 var table = tableClient.GetTableReference("Contacts");
 var retrieveOperation = TableOperation.Retrieve<ContactEntity>("gov-partition1", "0fb52a6c-3784-4dc5-
aa6d-ecda4426dbda");
 var result = await table.ExecuteAsync(retrieveOperation);
 var contact = result.Result as ContactEntity;
 Console.WriteLine($"Contact: {contact.FirstName} {contact.LastName}");
```

Java

1. Download the Azure Storage SDK for Java and configure your project accordingly.
2. Create a `CustomerEntity` class in your project and paste the code below:

```
import com.microsoft.azure.storage.table.TableServiceEntity;

public class CustomerEntity extends TableServiceEntity {
        public CustomerEntity(String lastName, String firstName) {
            this.partitionKey = lastName;
            this.rowKey = firstName;
        }

        public CustomerEntity() { }

        String email;

        public String getEmail() {
            return this.email;
        }

        public void setEmail(String email) {
            this.email = email;
        }

    }
```

3. Create a "test" class where we will access Azure Table Storage using the Azure Storage API. Copy and paste the code below, and **paste** your Storage Account connection string into the storageConnectionString variable.

```
import com.microsoft.azure.storage.*;
import com.microsoft.azure.storage.table.*;

public class test {

    public static final String storageConnectionString = //Paste in your Storage Account connection
string

    public static void main(String[] args) {

    try
    {
        // Retrieve storage account from connection-string.
        CloudStorageAccount storageAccount =
        CloudStorageAccount.parse(storageConnectionString);

        // Create the table client.
        CloudTableClient tableClient = storageAccount.createCloudTableClient();

        // Create the table if it doesn't exist.
        String tableName = "Contacts";
        CloudTable cloudTable = tableClient.getTableReference(tableName);
        cloudTable.createIfNotExists();
        // Create a new customer entity.
        CustomerEntity customer1 = new CustomerEntity("Brown", "Walter");
        customer1.setEmail("Walter@contoso.com");

        // Create an operation to add the new customer to the people table.
        TableOperation insertCustomer1 = TableOperation.insertOrReplace(customer1);

        // Submit the operation to the table service.
        cloudTable.execute(insertCustomer1);
    }
    catch (Exception e)
    {
        // Output the stack trace.
        e.printStackTrace();
    }
  }
}
```

Node.js

1. Download the Azure Storage SDK for Node.js and configure your application accordingly.
2. The following code below connects to Azure Blob Storage and creates a Container using the Azure Storage API. **Paste** your Azure Storage account connection string into the storageConnectionString variable below.

```
var azure = require('azure-storage');
var storageConnectionString = //Paste Azure Storage connection string here
var blobSvc = azure.createBlobService(storageConnectionString);
blobSvc.createContainerIfNotExists('testing', function(error, result, response){
if(!error){
// Container exists and is private
}
});
```

Python

1. Download the Azure Storage SDK for Python.
2. When using the Storage SDK for Python to connect to Azure Government, you **must separately define an "endpoint_suffix" parameter**. **Paste** in your Azure storage account name and key in the placeholders below.

```
# Create the BlockBlockService that is used to call the Blob service for the storage account
block_blob_service = BlockBlobService(account_name='#your account name', account_key='#your account
key', endpoint_suffix="core.usgovcloudapi.net")
container_name ='ml-gov-demo'
generator = block_blob_service.list_blobs(container_name)
for blob in generator:
    print(blob.name)
```

PHP

1. Download the Azure Storage SDK for PHP.
2. The code below accesses Azure Table Storage using the Azure Storage API. In the `connectionString` variable you will notice that there is a `TableEndpoint` parameter. Depending on which service you are using, you must define the parameter and set it to the endpoint for that service:

   - BlobEndpoint= //ends with 'blob.core.usgovcloudapi.net'
   - QueueEndpoint= //ends with 'queue.core.usgovcloudapi.net'
   - TableEndpoint= //ends with 'table.core.usgovcloudapi.net'

   > **NOTE**
   > You can find these endpoints by navigating to your Storage Account from the portal. **Paste** in your storage account name, key, and service endpoint in the `connectionString` variable.

   ```
   <?php
   require_once "vendor/autoload.php";
   use WindowsAzure\Common\ServicesBuilder;
   use MicrosoftAzure\Storage\Common\ServiceException;
   $connectionString = 'DefaultEndpointsProtocol=http;AccountName=<accountname>;AccountKey=
   <accountkey>;TableEndpoint=http://<storageaccountname>.table.core.usgovcloudapi.net/';

   $tableRestProxy = ServicesBuilder::getInstance()->createTableService($connectionString);
   try {
   // Create table.
   $tableRestProxy->createTable("test");
   }
   catch(ServiceException $e){
   $code = $e->getCode();
   $error_message = $e->getMessage();
   }
   ?>
   ```

Next steps

- Read more about Azure Storage.
- Subscribe to the Azure Government blog
- Get help on Stack Overflow by using the "azure-gov" tag
- Give us feedback or request new features via the Azure Government feedback forum

Connect to Azure Government with SQL Server Management Studio

12/15/2017 • 2 minutes to read • Edit Online

To use SQL Server Management Studio (SSMS) with Azure Government, specify Azure Government as the environment to connect to, rather than Azure Public. To connect to computers that are running SQL Server in your Azure Government subscription, you must configure SSMS to connect to the Azure Government cloud.

For general information about SSMS, see the SSMS documentation.

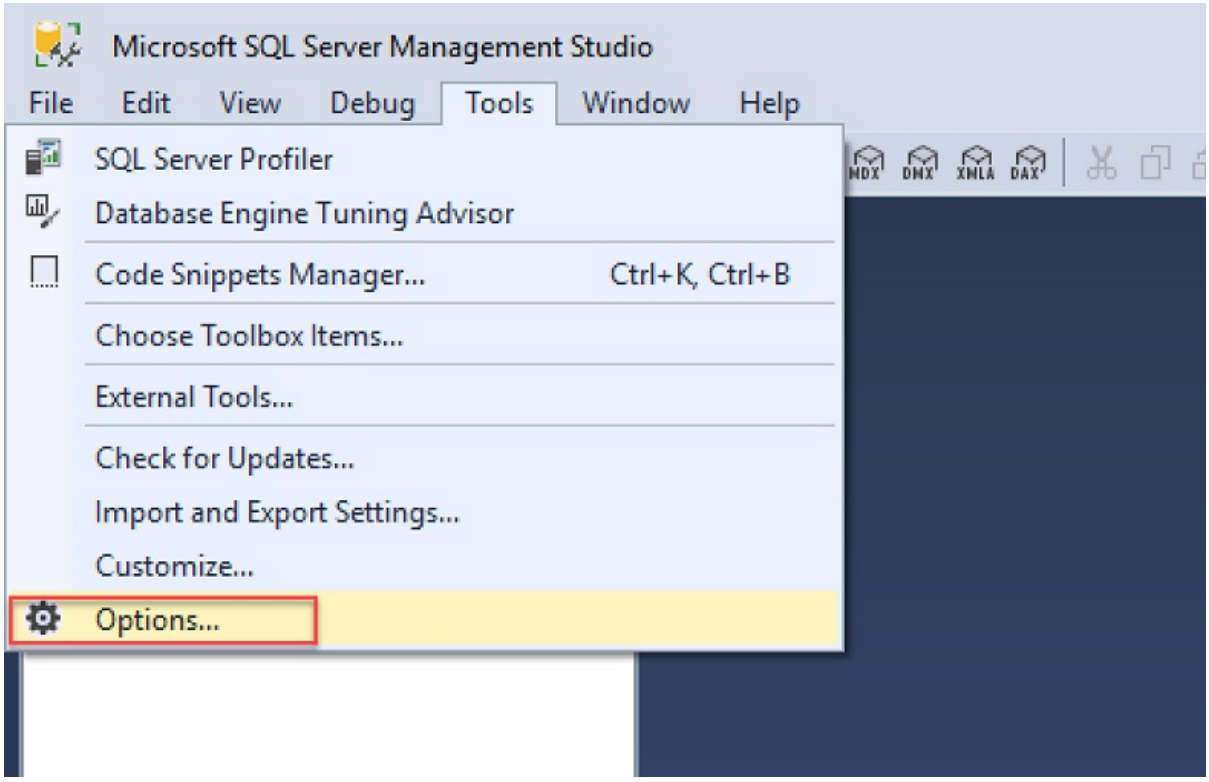
Set up an Azure SQL Server firewall rule

Before you connect to Azure Government from SSMS, you must set up an Azure SQL Server firewall rule to allow your local IP address to access your computer that's running SQL Server.

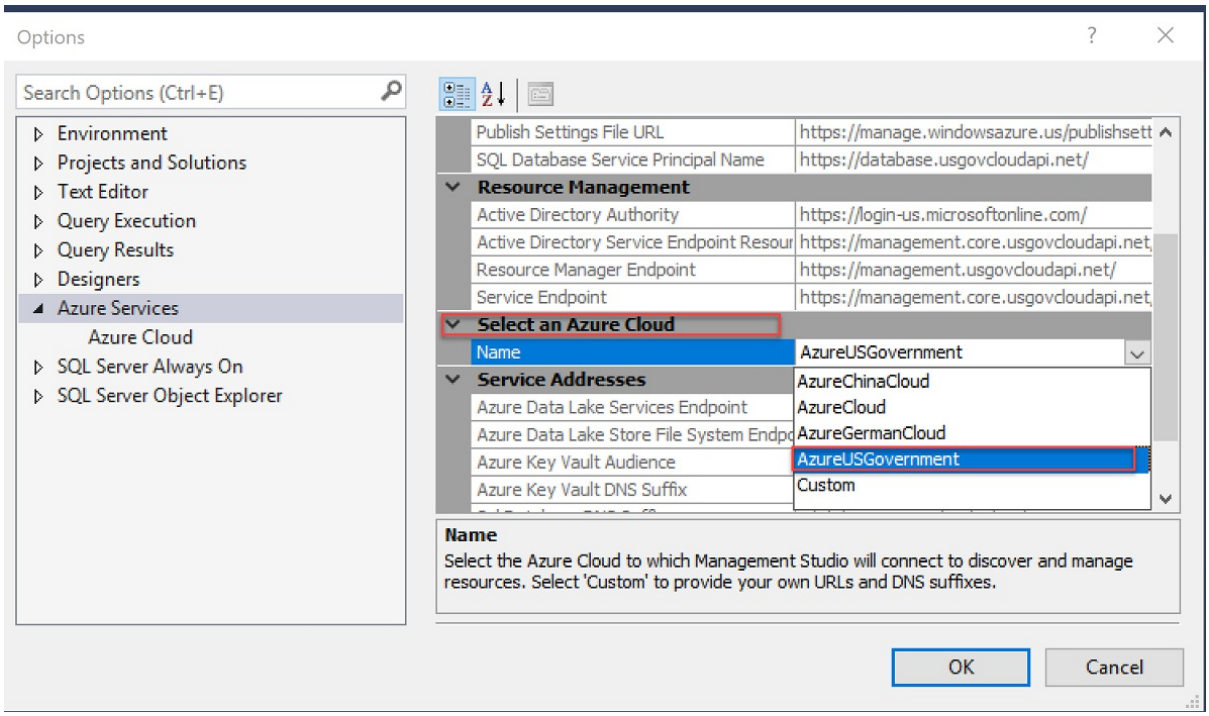
Follow these steps to Manage firewall rules by using the Azure portal.

Specify Azure Government as the environment to connect

1. Open SSMS. Browse to **Tools** > **Options** > **Azure Services**.

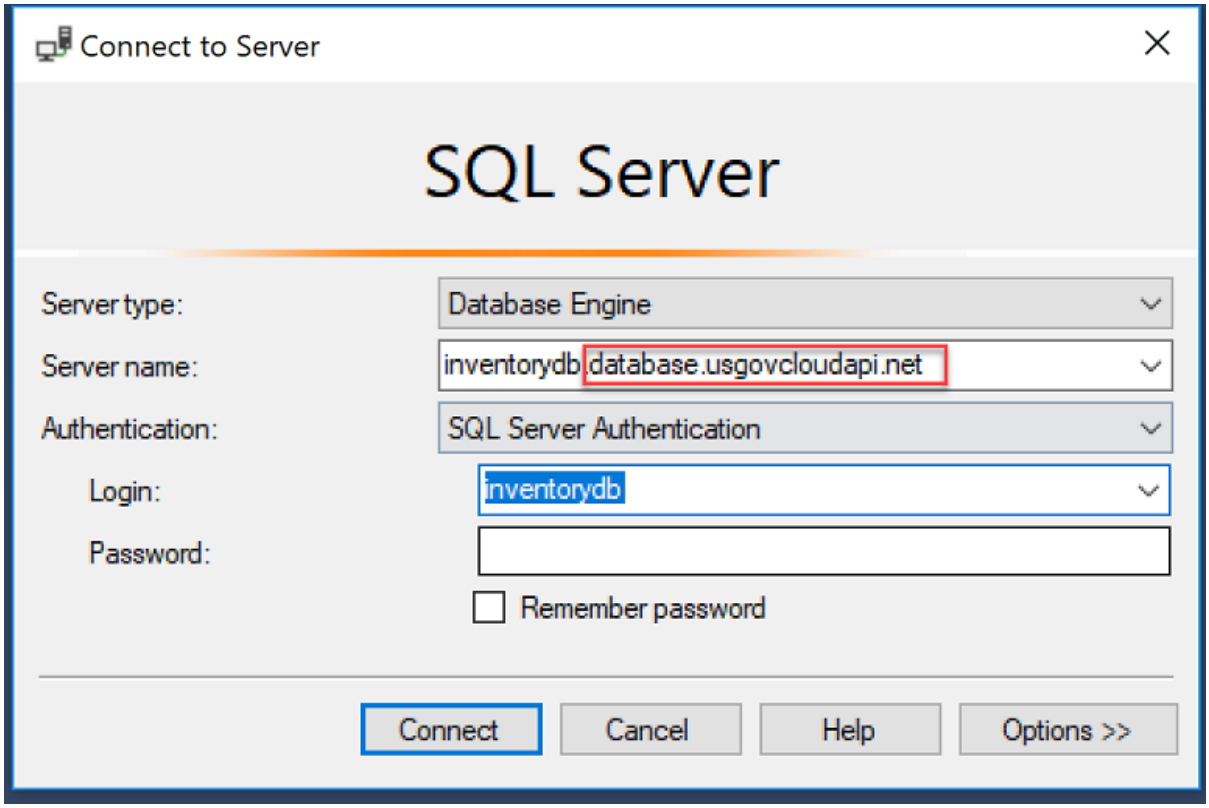


2. In the **Select an Azure Cloud** drop-down, select **AzureUSGovernment**.



3. Browse to **File** > **Connect Object Explorer**. Enter the name of your computer that's running SQL Server. Enter your authentication information.

> **NOTE**
>
> The name of the computer that's running SQL Server ends with **.usgovcloudapi.net**.



SSMS is now connected to your Azure Government subscription.

Next steps

- Read more about Azure Storage.
- Subscribe to the Azure Government blog.

- Get help on Stack Overflow by using the `azure-gov` tag.
- Share feedback or request new features by using the Azure Government feedback forum.

Azure Government Marketplace images

7/13/2018 • 10 minutes to read • Edit Online

The Azure Government Marketplace provides a similar experience as the public Azure portal. You can choose to deploy prebuilt images from Microsoft and our partners, or upload your own VHDs. This gives you the flexibility to deploy your own standardized images if needed.

The following table shows a list of available images within the Azure Government Marketplace. If you'd like to see other images in Azure Government, please request them via the Azure Government Feedback Forum.

Some of the prebuilt images include pay-as-you-go licensing for specific software. Work with your Microsoft account team or reseller for Azure Government-specific pricing. For more information, see Virtual machine pricing.

Images

The list of virtual machine images available in Azure Government can be obtained by connecting to Azure Government via PowerShell and running the following commands:

```
Connect-AzureRmAccount -Environment AzureUSGovernment

Get-AzureRmVMImagePublisher -Location USGovVirginia | `
Get-AzureRmVMImageOffer | `
Get-AzureRmVMImageSku
```

The table below contains a snapshot of the list of virtual machine images available in Azure Government via Resource Manager as of July 13, 2018.

PUBLISHER	OFFER	SKU
a10networks	a10-vthunder-adc	vthunder_410_byol
a10networks	a10-vthunder-adc	vthunder_byol
akumina	akumina-interchange	akam101
alertlogic	alert-logic-tm	20215000100-tmpbyol
alertlogic	alert-logic-wsm	20216000100-wsmpbyl
altamira-corporation	lumify	lumify
asigra	asigra-on-azure	asigra-evaluation-vm
aviatrix-systems	aviatrix-cloud-services	av-csg-byol
aviatrix-systems	aviatrix-companion-gateway-v2	aviatrix-companion-gateway-v2
barracudanetworks	barracuda-app-sec-control-center	byol
barracudanetworks	barracuda-email-security-gateway	byol

PUBLISHER	OFFER	SKU
barracudanetworks	barracuda-email-security-gateway	hourly
barracudanetworks	barracuda-message-archiver	byol
barracudanetworks	barracuda-ng-cc	byol
barracudanetworks	barracuda-ng-firewall	byol
barracudanetworks	barracuda-ng-firewall	hourly
barracudanetworks	barracuda-ng-firewall-90-day-trial	hourly-90-day-trial
barracudanetworks	barracuda-spam-firewall	byol
barracudanetworks	barracuda-waf-90-day-trial	hourly-90-day-trial
barracudanetworks	waf	byol
barracudanetworks	waf	hourly
batch	rendering-centos73	rendering
batch	rendering-windows2016	rendering
beyondtrust	beyondinsight	uvm-azm
bitnami	abantecart	1-2
bitnami	activemq	5-13
bitnami	activemq	default
bitnami	akeneo	1-4
bitnami	alfrescocommunity	201602
bitnami	apachesolr	5-5
bitnami	artifactory	4-5
bitnami	canvaslms	2016-02
bitnami	cassandra	3-7
bitnami	cassandra	cassandra
bitnami	cassandra	default
bitnami	chyrp	2-5

PUBLISHER	OFFER	SKU
bitnami	civicrm	4-7
bitnami	cmsmadesimple	2-1
bitnami	codiad	2-7
bitnami	concrete5	5-7
bitnami	coppermine	1-5
bitnami	couchdb	1-6
bitnami	couchdb	couchdb
bitnami	diaspora	0-5
bitnami	discourse	1-4
bitnami	djangostack	1-8
bitnami	dokuwiki	20150810a
bitnami	dolibarr	3-8
bitnami	dreamfactory	2-1
bitnami	drupal	8-0
bitnami	eclipseche	4-4
bitnami	elastic-search	2-2
bitnami	elk	4-6
bitnami	erpnext	6-21
bitnami	espocrm	3-9
bitnami	exoplatform	4
bitnami	exoplatformenterprise	4-2
bitnami	ezpublish	2014-11
bitnami	fatfreecrm	0-13
bitnami	ghost	0-7
bitnami	ghost	default

PUBLISHER	OFFER	SKU
bitnami	gitlab	8-5
bitnami	hadoop	2-7
bitnami	hadoop	default
bitnami	hhvmstack	3-9
bitnami	hordegroupwarewebmail	5-2
bitnami	jasperreports	6-2
bitnami	jbossas	7-2
bitnami	jenkins	1-650
bitnami	joomla	3-5
bitnami	jrubystack	9-0
bitnami	kafka	0-1
bitnami	kafka	default
bitnami	kafka	kafka
bitnami	kong	default
bitnami	kubernetessandbox	default
bitnami	lampstack	5-6
bitnami	lappstack	5-6
bitnami	letschat	0-4
bitnami	liferay	6-2
bitnami	limesurvey	20160228
bitnami	livehelperchat	2-44v
bitnami	magento	2-0
bitnami	mahara	15-10
bitnami	mantis	1-2
bitnami	mariadb	default

PUBLISHER	OFFER	SKU
bitnami	mariadb	mariadb
bitnami	mattermost	3-6
bitnami	mautic	1-2
bitnami	mean	3-2
bitnami	mediawiki	1-26
bitnami	memcached	1-4
bitnami	memcached	default
bitnami	memcached	memcached
bitnami	modx	2-4
bitnami	mongodb	3-2
bitnami	mongodb	default
bitnami	moodle	3-0
bitnami	moodle	moodle-free-byol
bitnami	multicraft	public
bitnami	mybb	1-8
bitnami	mysql	5-6
bitnami	mysql	default
bitnami	neos	2-0
bitnami	nginxstack	1-9
bitnami	noalyss	6-9
bitnami	nodejs	4-3
bitnami	ocportal	9
bitnami	odoo	9-0
bitnami	openatrium	2-54
bitnami	opencart	2-1

PUBLISHER	OFFER	SKU
bitnami	openedx	cypress
bitnami	openfire	4
bitnami	openproject	5-0
bitnami	orangehrm	3-3
bitnami	orocrm	1
bitnami	osclass	3-6
bitnami	osqa	1-0rc
bitnami	owncloud	8-2
bitnami	oxid-eshop	4-9
bitnami	parseserver	2-1
bitnami	parseserver	default
bitnami	phabricator	20160208
bitnami	phpbb	3-1
bitnami	phplist	3-2
bitnami	pimcore	3-1
bitnami	piwik	2-16
bitnami	plone	5-0
bitnami	pootle	2-7
bitnami	postgresql	9-5
bitnami	postgresql	default
bitnami	postgresql	postgresql
bitnami	prestashop	1-6-1
bitnami	processmakerenterprise	3-1
bitnami	processmakeropensourceedition	3-0
bitnami	processwire	2-7

PUBLISHER	OFFER	SKU
bitnami	publify	8-2
bitnami	rabbitmq	3-6
bitnami	rabbitmq	default
bitnami	rabbitmq	rabbitmq
bitnami	railo	4-2
bitnami	redash	0-10
bitnami	redis	3-2
bitnami	redis	default
bitnami	redis	redis
bitnami	redmine	3
bitnami	redmineplusagile	public
bitnami	refinerycms	2-1
bitnami	reportserver	2-2
bitnami	reportserverenterprise	3-0
bitnami	resourcespace	7-5
bitnami	reviewboard	2-5
bitnami	reviewboardpowerpack	public
bitnami	roundcube	1-1
bitnami	rubystack	2-0
bitnami	seopanel	3-8
bitnami	shopware	default
bitnami	silverstripe	3-2
bitnami	simpleinvoices	2013-1
bitnami	simplemachinesforum	2-0
bitnami	sonarqube	6-4

PUBLISHER	OFFER	SKU
bitnami	spree	3-0
bitnami	squash	20151209
bitnami	subversion	1-8
bitnami	suitecrm	7-4
bitnami	tensorflowserving	default
bitnami	testlink	1-9
bitnami	tikiwikicmsgroupware	14-2
bitnami	tinytinyrss	20160220
bitnami	tom-cat	7-0
bitnami	trac	1-0
bitnami	typo3	7-6
bitnami	weblate	2-4
bitnami	webmailpro	public
bitnami	wildfly	10-0
bitnami	wordpress	4-4
bitnami	wordpress-multisite	4
bitnami	wordpresspro	default
bitnami	x2enginesalescrm	5-5
bitnami	xoops	2-5
bitnami	youtrack	7-0
bitnami	zurmo	3-1
Canonical	UbuntuServer	12.04.5-LTS
Canonical	UbuntuServer	14.04.4-LTS
Canonical	UbuntuServer	14.04.5-LTS
Canonical	UbuntuServer	16.04-LTS

PUBLISHER	OFFER	SKU
Canonical	UbuntuServer	16.04.0-LTS
Canonical	UbuntuServer	16.10
Canonical	UbuntuServer	17.04
Canonical	UbuntuServer	17.04-DAILY
Canonical	UbuntuServer	17.10
checkpoint	check-point-r77-10	SG-BYOL
checkpoint	check-point-vsec-r80	sg-byol
checkpoint	sg2	sg-byol2
chef-software	chef-automate-vm-image	byol
cisco	cisco-asav	asav-azure-byol
cisco	cisco-csr-1000v	16_5
cisco	cisco-csr-1000v	16_6
cisco	cisco-csr-1000v	16_7
cisco	cisco-csr-1000v	3_16
cisco	cisco-csr-1000v	csr-azure-byol
cisco	cisco-ftdv	ftdv-azure-byol
cisco	cisco_cloud_vedge_17_2_4	cisco_vedge_azurecloud_18_2_0
citrix	netscaler-sd-wan	netscalersd-wanstandardedition
citrix	netscalervpx-120	netscalerbyol
citrix	netscalervpx-121	netscalerbyol
citrix	netscalervpx110-6531	netscalerbyol
citrix	netscalervpx111	netscalerbyol
citrix	xenapp-server	coldfireserver
citrix	xenapp-vda-rdsh	coldfirerdsh
citrix	xenapp-vda-rdsh	server2016rdsh

PUBLISHER	OFFER	SKU
citrix	xenapp-vda-vdi	coldfirevdi
citrix	xenapp-vda-vdi	server2016vdi
clouber	cws	cuber
cloud-checkr	cloudcheckr-gov	cloudcheckr-gov
cloudera	cloudera-centos-6	cloudera-centos-6
cloudera	cloudera-centos-os	6_7
cloudera	cloudera-centos-os	6_8
cloudera	cloudera-centos-os	7_2
cloudera	cloudera-centos-os	7_4
codelathe	codelathe-filecloud-ubuntu	filecloud_ubuntu_byol
codelathe	codelathe-filecloud-win2012r2	filecloud_byol
codelathe	filecloud-efss-windows2016	filecloud_windows2016
cohesive	vns3_4x_network_security	cohesive-vns3-4x-byol
commvault	commvault	commvaulttrial
composable	composable	composable-govt
connecting-software	cb-replicator-byol	cbrep-gov-byol
CoreOS	CoreOS	Stable
couchbase	couchbase-server-enterprise	byol
couchbase	couchbase-sync-gateway-enterprise	byol
credativ	Debian	7
credativ	Debian	8
credativ	Debian	8-backports
credativ	Debian	9
credativ	Debian	9-beta
datacore	datacore-maxparallel_sql2012	1b-sq12e13-w1220-mp200

PUBLISHER	OFFER	SKU
datacore	datacore-maxparallel_sql2012	1b-sq12s13-w1220-mp200
datastax	datastax-enterprise	datastaxenterprise
dellemc	dell-emc-datadomain-virtual-edition	ddve-31-ver-060100
dellemc	dell-emc-datadomain-virtual-edition	ddve-31-ver-060101
dell_software	uccs	uccs
delphix	delphix_dynamic_data_platform	dynamic_data_platform_for_azure_5-1-8-0
derdack	enterprisealert	enterprisealert-2017-datacenter-byol
docker	docker-ee	docker-ee
docker	docker4azure-cs	docker4azure-cs-1_12
docker	docker4azure-cs	docker4azure-cs-1_1x
dynatrace	ruxit-managed-vm	byol-managed
enterprise-ethereum-alliance	quorum-demo	quorum-demo
esri	arcgis-10-4-for-server	cloud
esri	arcgis-desktop	desktop-byol-106
esri	arcgis-desktop	desktop-byol-1061
esri	arcgis-enterprise	byol
esri	arcgis-enterprise	byol-1051
esri	arcgis-enterprise-106	byol-106
esri	arcgis-enterprise-106	byol-1061
esri	arcgis-for-server	cloud
eventtracker	eventtracker-siem	etlm
eventtracker	eventtracker-siem	etsc
f5-networks	f5-big-ip-adc	f5-bigip-virtual-edition-better-byol
f5-networks	f5-big-ip-adc	f5-bigip-virtual-edition-good-byol
f5-networks	f5-big-ip-advanced-waf	f5-bigip-virtual-edition-1g-waf-hourly

PUBLISHER	OFFER	SKU
f5-networks	f5-big-ip-advanced-waf	f5-bigip-virtual-edition-200m-waf-hourly
f5-networks	f5-big-ip-advanced-waf	f5-bigip-virtual-edition-25m-waf-hourly
f5-networks	f5-big-ip-best	f5-bigip-virtual-edition-1g-best-hourly
f5-networks	f5-big-ip-best	f5-bigip-virtual-edition-200m-best-hourly
f5-networks	f5-big-ip-best	f5-bigip-virtual-edition-25m-best-hourly
f5-networks	f5-big-ip-best	f5-bigip-virtual-edition-best-byol
f5-networks	f5-big-ip-better	f5-bigip-virtual-edition-1g-better-hourly
f5-networks	f5-big-ip-better	f5-bigip-virtual-edition-200m-better-hourly
f5-networks	f5-big-ip-better	f5-bigip-virtual-edition-25m-better-hourly
f5-networks	f5-big-ip-better	f5-bigip-virtual-edition-better-byol
f5-networks	f5-big-ip-good	f5-bigip-virtual-edition-200m-good-hourly
f5-networks	f5-big-ip-good	f5-bigip-virtual-edition-25m-good-hourly
f5-networks	f5-big-ip-good	f5-bigip-virtual-edition-good-byol
f5-networks	f5-big-iq	f5-bigiq-virtual-edition-byol
flashgrid-inc	flashgrid-ol7-g	fg-17-05-ol74-g
flashgrid-inc	flashgrid-ol7-g	fg-rh-gc
flashgrid-inc	flashgrid-racnode	fg-1709-ol
flashgrid-inc	flashgrid-racnode	fg-1709-rh
flashgrid-inc	flashgrid-racnode	fg-1709-rh-mc
flashgrid-inc	flashgrid-racnode	fg-ol7-priv-byol
flashgrid-inc	flashgrid-racnode	fg-rh7-priv-byol
fortinet	fortinet-fortianalyzer	fortinet-fortianalyzer

PUBLISHER	OFFER	SKU
fortinet	fortinet-fortimanager	fortinet-fortimanager
fortinet	fortinet_fortigate-vm_v5	fortinet_fg-vm
fortinet	fortinet_fortimail	fortinet_fortimail
fortinet	fortinet_fortiweb-vm_v5	fortinet_fw-vm
gigamon-inc	gigamon-fm-5_3_01	gfm-azure
gigamon-inc	gigamon-fm-5_3_01	gvtap-cntlr
gigamon-inc	gigamon-fm-5_3_01	vseries-cntlr
gigamon-inc	gigamon-fm-5_3_01	vseries-node
hanu	hanu-insightv2	hanu-insight-v2-enterprise-byol
hanu	hanu-insightv2	hanu-insight-v2-standard-byol
infoblox	infoblox-vnios-te-v1420	vnios-cp-v1400
infoblox	infoblox-vnios-te-v1420	vnios-cp-v2200
infoblox	infoblox-vnios-te-v1420	vnios-cp-v800
infoblox	infoblox-vnios-te-v1420	vnios-te-v1420
infoblox	infoblox-vnios-te-v1420	vnios-te-v2220
infoblox	infoblox-vnios-te-v1420	vnios-te-v820
infoblox	infoblox-vnios-te-v1420	vsot
jamcracker	4632d5b4-feb0-4332-8452-f2e66133672f	jamcracker_cloud_control_appliance_version5
jamcracker	jamcracker-cloudanalytics-version4	jamcracker-cloud-analytics-version4
jamcracker	jamcracker-cloudanalytics-version5	jamcracker-cloudanalytics-version5
jamcracker	jamcracker-csb-service-provider	jc-csbsp-version5
jamcracker	jamcracker-csb-serviceprovider	jc-csbsp-version5
jamcracker	jamcracker-csb-standard	jamcracker-csb-standard-version5
jamcracker	jamcracker-csb-standard-v3	jamcracker-csb-standard-v3
jamcracker	jamcracker-csb-standard-version4	jamcracker-csb-standard-version4

PUBLISHER	OFFER	SKU
jamcracker	jamcracker-hybrid-cloud-management-version4	jamcracker-hybrid-cloud-management-version4
jamcracker	jamcracker_cloud_control_appliance_version4	jamcracker-cloud-control-appliance-version4
jamcracker	jsdnapp_csb_serviceprovider-version4	jc-csbsp-version4
jamcracker	jsdnapp_hybrid_v3	jamcracker-hybrid-cloud-management-version5
juniper-networks	vmx-services-gateway-byol	vmx-services-gateway-byol
juniper-networks	vsrx-next-generation-firewall	vsrx-byol-azure-image
juniper-networks	vsrx-next-generation-firewall-solution-template	vsrx-byol-azure-image-solution-template
kali-linux	kali-linux	kali
kemptech	kemp360central-byol	kemp360central-byol
kemptech	kemp360central-byol	kemp360central-spla
kemptech	vlm-azure	basic-byol
kemptech	vlm-azure	freeloadmaster
kemptech	vlm-azure	vlm-byol-lts
kemptech	vlm-azure	vlm-spla
kemptech	vlm-azure	vlm-spla-lts
kinetica	kineticadbbyol	centos73-601
kinetica	kineticadbbyol	centos75-620
liebsoft	enterprise_random_password_manager	redim5521
mapr-technologies	mapr52-base-dev	5202
marklogic	marklogic-9-byol	ml9031_centos_byol
marklogic	marklogic-developer-9	ml9031_centos
mico	mobile-impact-platform	mipvm
microsoft-ads	linux-data-science-vm-ubuntu	linuxdsvmubuntubyol
microsoft-ads	windows-data-science-vm	windows2016byol

PUBLISHER	OFFER	SKU
microsoft-dsvm	dsvm-windows	server-2016
microsoft-dsvm	linux-data-science-vm-ubuntu	linuxdsvmubuntu
microsoft-hyperv	rs5_preview	2019-datacenter
MicrosoftAzureSiteRecovery	Process-Server	Windows-2012-R2-Datacenter
MicrosoftHybridCloudStorage	StorSimple	StorSimple-Garda-8000-Series
MicrosoftHybridCloudStorage	StorSimple	StorSimple-Garda-8000-Series-BBUpdate
MicrosoftHybridCloudStorage	StorSimpleVA	StorSimpleUpdate3RC
MicrosoftOSTC	FreeBSD	10.3
MicrosoftOSTC	FreeBSD	11
MicrosoftOSTC	FreeBSD	11.0
MicrosoftRServer	MLServer-CentOS	Enterprise
MicrosoftRServer	MLServer-RedHat	Enterprise
MicrosoftRServer	MLServer-Ubuntu	Enterprise
MicrosoftRServer	MLServer-WS2016	Enterprise
MicrosoftRServer	RServer-CentOS	Enterprise
MicrosoftRServer	RServer-RedHat	Enterprise
MicrosoftRServer	RServer-Ubuntu	Enterprise
MicrosoftRServer	RServer-WS2016	Enterprise
MicrosoftSharePoint	MicrosoftSharePointServer	2016
MicrosoftSQLServer	SQL2008R2SP3-WS2008R2SP1	Enterprise
MicrosoftSQLServer	SQL2008R2SP3-WS2008R2SP1	Express
MicrosoftSQLServer	SQL2008R2SP3-WS2008R2SP1	Standard
MicrosoftSQLServer	SQL2008R2SP3-WS2008R2SP1	Web
MicrosoftSQLServer	SQL2012SP3-WS2012R2	Enterprise
MicrosoftSQLServer	SQL2012SP3-WS2012R2	Express

PUBLISHER	OFFER	SKU
MicrosoftSQLServer	SQL2012SP3-WS2012R2	Standard
MicrosoftSQLServer	SQL2012SP3-WS2012R2	Web
MicrosoftSQLServer	SQL2012SP3-WS2012R2-BYOL	Enterprise
MicrosoftSQLServer	SQL2012SP3-WS2012R2-BYOL	Standard
MicrosoftSQLServer	SQL2012SP4-WS2012R2	Enterprise
MicrosoftSQLServer	SQL2012SP4-WS2012R2	Express
MicrosoftSQLServer	SQL2012SP4-WS2012R2	Standard
MicrosoftSQLServer	SQL2012SP4-WS2012R2	Web
MicrosoftSQLServer	SQL2012SP4-WS2012R2-BYOL	Enterprise
MicrosoftSQLServer	SQL2012SP4-WS2012R2-BYOL	Standard
MicrosoftSQLServer	SQL2014SP1-WS2012R2	Enterprise
MicrosoftSQLServer	SQL2014SP1-WS2012R2	Express
MicrosoftSQLServer	SQL2014SP1-WS2012R2	Standard
MicrosoftSQLServer	SQL2014SP1-WS2012R2	Web
MicrosoftSQLServer	SQL2014SP1-WS2012R2-BYOL	Enterprise
MicrosoftSQLServer	SQL2014SP1-WS2012R2-BYOL	Standard
MicrosoftSQLServer	SQL2014SP2-WS2012R2	Enterprise
MicrosoftSQLServer	SQL2014SP2-WS2012R2	Express
MicrosoftSQLServer	SQL2014SP2-WS2012R2	Standard
MicrosoftSQLServer	SQL2014SP2-WS2012R2	Web
MicrosoftSQLServer	SQL2014SP2-WS2012R2-BYOL	Enterprise
MicrosoftSQLServer	SQL2014SP2-WS2012R2-BYOL	Standard
MicrosoftSQLServer	SQL2016-WS2012R2	Enterprise
MicrosoftSQLServer	SQL2016-WS2012R2	Express
MicrosoftSQLServer	SQL2016-WS2012R2	SQLDEV

PUBLISHER	OFFER	SKU
MicrosoftSQLServer	SQL2016-WS2012R2	Standard
MicrosoftSQLServer	SQL2016-WS2012R2	Web
MicrosoftSQLServer	SQL2016-WS2012R2-BYOL	Enterprise
MicrosoftSQLServer	SQL2016-WS2012R2-BYOL	Standard
MicrosoftSQLServer	SQL2016-WS2016	Enterprise
MicrosoftSQLServer	SQL2016-WS2016	SQLDEV
MicrosoftSQLServer	SQL2016-WS2016	Standard
MicrosoftSQLServer	SQL2016-WS2016	Web
MicrosoftSQLServer	SQL2016-WS2016-BYOL	Enterprise
MicrosoftSQLServer	SQL2016-WS2016-BYOL	Standard
MicrosoftSQLServer	SQL2016SP1-WS2016	Enterprise
MicrosoftSQLServer	SQL2016SP1-WS2016	Express
MicrosoftSQLServer	SQL2016SP1-WS2016	SQLDEV
MicrosoftSQLServer	SQL2016SP1-WS2016	Standard
MicrosoftSQLServer	SQL2016SP1-WS2016	Web
MicrosoftSQLServer	SQL2016SP1-WS2016-BYOL	Enterprise
MicrosoftSQLServer	SQL2016SP1-WS2016-BYOL	Standard
MicrosoftSQLServer	SQL2016SP2-WS2016	Enterprise
MicrosoftSQLServer	SQL2016SP2-WS2016	Express
MicrosoftSQLServer	SQL2016SP2-WS2016	SQLDEV
MicrosoftSQLServer	SQL2016SP2-WS2016	Standard
MicrosoftSQLServer	SQL2016SP2-WS2016	Web
MicrosoftSQLServer	SQL2016SP2-WS2016-BYOL	Enterprise
MicrosoftSQLServer	SQL2016SP2-WS2016-BYOL	Standard
MicrosoftSQLServer	SQL2017-RHEL73	Evaluation

PUBLISHER	OFFER	SKU
MicrosoftSQLServer	SQL2017-WS2016	Enterprise
MicrosoftSQLServer	SQL2017-WS2016	Express
MicrosoftSQLServer	SQL2017-WS2016	SQLDEV
MicrosoftSQLServer	SQL2017-WS2016	Standard
MicrosoftSQLServer	SQL2017-WS2016	Web
MicrosoftSQLServer	SQL2017-WS2016-BYOL	Enterprise
MicrosoftSQLServer	SQL2017-WS2016-BYOL	Standard
MicrosoftVisualStudio	VisualStudio	VS-2015-Comm-VSU3-AzureSDK-29-WS2012R2
MicrosoftVisualStudio	VisualStudio	VS-2015-Comm-VSU3-AzureSDK-291-WS2012R2
MicrosoftVisualStudio	VisualStudio	VS-2015-Ent-VSU3-AzureSDK-29-WS2012R2
MicrosoftVisualStudio	VisualStudio	VS-2017-Comm-Latest-Preview-WS2016
MicrosoftVisualStudio	VisualStudio	VS-2017-Comm-Latest-WS2016
MicrosoftVisualStudio	VisualStudio	VS-2017-Comm-WS2016
MicrosoftVisualStudio	VisualStudio	VS-2017-Ent-Latest-Preview-WS2016
MicrosoftVisualStudio	VisualStudio	VS-2017-Ent-Latest-WS2016
MicrosoftVisualStudio	VisualStudio	VS-2017-Ent-WS2016
MicrosoftWindowsDesktop	Windows-10	RS2-Pro
MicrosoftWindowsDesktop	Windows-10	RS2-ProN
MicrosoftWindowsDesktop	Windows-10	RS3-Pro
MicrosoftWindowsDesktop	Windows-10	RS3-ProN
MicrosoftWindowsDesktop	Windows-10	rs4-pro
MicrosoftWindowsDesktop	Windows-10	rs4-pron
MicrosoftWindowsServer	WindowsServer	2008-R2-SP1
MicrosoftWindowsServer	WindowsServer	2012-Datacenter

PUBLISHER	OFFER	SKU
MicrosoftWindowsServer	WindowsServer	2012-Datacenter-smalldisk
MicrosoftWindowsServer	WindowsServer	2012-R2-Datacenter
MicrosoftWindowsServer	WindowsServer	2012-R2-Datacenter-smalldisk
MicrosoftWindowsServer	WindowsServer	2016-Datacenter
MicrosoftWindowsServer	WindowsServer	2016-Datacenter-Server-Core
MicrosoftWindowsServer	WindowsServer	2016-Datacenter-Server-Core-smalldisk
MicrosoftWindowsServer	WindowsServer	2016-Datacenter-smalldisk
MicrosoftWindowsServer	WindowsServer	2016-Datacenter-with-Containers
MicrosoftWindowsServer	WindowsServer	2016-Datacenter-with-RDSH
MicrosoftWindowsServer	WindowsServer	2016-Nano-Server
MicrosoftWindowsServer	WindowsServer-HUB	2008-R2-SP1-HUB
MicrosoftWindowsServer	WindowsServer-HUB	2012-Datacenter-HUB
MicrosoftWindowsServer	WindowsServer-HUB	2012-R2-Datacenter-HUB
MicrosoftWindowsServer	WindowsServer-HUB	2016-Datacenter-HUB
MicrosoftWindowsServer	WindowsServerSemiAnnual	Datacenter-Core-1709-smalldisk
MicrosoftWindowsServer	WindowsServerSemiAnnual	Datacenter-Core-1709-with-Containers-smalldisk
MicrosoftWindowsServer	WindowsServerSemiAnnual	Datacenter-Core-1803-with-Containers-smalldisk
MicrosoftWindowsServerRemoteDesktop	WindowsServer	RDSH-Office365P
MicrosoftWindowsServerRemoteDesktop	WindowsServer	Remote-Desktop-Session-Host
nasuni	nasuni-nmc	nasuni_nmc_7_10_6_prod
nasuni	nasuni_edge_appliance	nasuni_edge_appliance_7_10_6_prod
netapp	netapp-oncommand-cloud-manager	occm-byol
netapp	netapp-ontap-cloud	ontap_cloud_byol
noobaa	noobaa-hybrid-s3-archive-05	pay-per-usage

PUBLISHER	OFFER	SKU
nubeva-inc	controller	byol
nuxeo	nuxeo-6-lts	nuxeo-6-lts
nuxeo	nuxeo-lts	nuxeo-lts-2015
nuxeo	nuxeo-lts	nuxeo-lts-2016
onyx-point-inc	op-bnf-v1	bnfcv1
onyx-point-inc	op-bnf1_6-v1	bnf1_6cv1
onyx-point-inc	op-dfi-v1	dfiv1
onyx-point-inc	op-scc-v1	op-scc-v1
OpenLogic	CentOS	6.7
OpenLogic	CentOS	6.8
OpenLogic	CentOS	6.9
OpenLogic	CentOS	7.2
OpenLogic	CentOS	7.2n
OpenLogic	CentOS	7.3
OpenLogic	CentOS	7.4
OpenLogic	CentOS	7.5
OpenLogic	CentOS-CI	7-CI
OpenLogic	CentOS-HPC	6.8
OpenLogic	CentOS-HPC	7.1
OpenLogic	CentOS-HPC	7.4
OpenLogic	CentOS-LVM	7-LVM
opentext	opentext-content-server-16	ot-cs16
Oracle	Oracle-Database-Ee	12.1.0.2
Oracle	Oracle-Database-Se	12.1.0.2
Oracle	Oracle-Linux	6.7

PUBLISHER	OFFER	SKU
Oracle	Oracle-Linux	6.8
Oracle	Oracle-Linux	7.2
orfast-technologies	orfast-mam-1	orasft_mam_01
paloaltonetworks	vmseries1	byol
panzura-file-system	azura-freedom-filer-v7110	fd-vm-azure-byol
panzura-file-system	panzura-cloud-filer	fd-vm-azure-byol
panzura-file-system	panzura-freedom-filer-v7020	fd-vm-azure-byol
pivotal	bosh-windows-server	2012r2gov
pivotal	pivotal-ops-manager	pivotal-ops-manager
qlik	qlik-sense	qliksense
qualysguard	qualys-virtual-scanner-v24	qvsa-24
quest	rapid-recovery-core-vm	quest_rapid_recovery_core_vm
radiant-logic	radiantone-vms	node-centos-7-5
radiant-logic	radiantone-vms	node-redhat-7-4
radiant-logic	radiantone-vms	node-redhat-7-5
radiant-logic	radiantone-vms	node-ubuntu-16-04-lts
radiant-logic	radiantone-vms	node-ubuntu-18-04-lts
radiant-logic	radiantone-vms	node-ws-2016
rapid7	nexpose-scan-engine	nexpose-scan-engine
rapid7	rapid7-vm-console	rapid7-vm-console
RedHat	RHEL	6.8
RedHat	RHEL	6.9
RedHat	RHEL	6.9-LVM
RedHat	RHEL	7-LVM
RedHat	RHEL	7-RAW

PUBLISHER	OFFER	SKU
RedHat	RHEL	7.2
RedHat	RHEL	7.3
RedHat	RHEL	7.3-LVM
RedHat	RHEL	7.4
RedHat	RHEL	7.4-LVM
RedHat	RHEL	7.4-RAW
RedHat	RHEL	7.4.Beta
RedHat	RHEL	7.4.Beta-LVM
RedHat	rhel-ocp-marketplace	rhel74
RedHat	rhel-ocp-marketplace	rhel75
RedHat	RHEL-SAP-APPS	6.8
RedHat	RHEL-SAP-APPS	7.3
RedHat	RHEL-SAP-HANA	6.7
RedHat	RHEL-SAP-HANA	7.2
riverbed	riverbed-sccm-5-5-1	riverbed-sccm-5-5-1
riverbed	riverbed-steelhead-9-1-3	steelhead-9-1-3
riverbed	riverbed-steelhead-9-2	riverbed-steelhead-9-2
riverbed	riverbed-steelhead-9-5-0	riverbed-steelhead-9-5-0
riverbed	riverbed-steelhead-9-6-0	riverbed-steelhead-9-6-0
scalegrid	centos	free
silver-peak-systems	silver_peak_edgeconnect	silver_peak_edgeconnect_8_1
silver-peak-systems	silver_peak_vx	silver-peak-vx-8-1
softnas	mp_nas_byol	mp_enterprise_byol
sophos	sophos-xg	byol
splunk	splunk-enterprise-base-image	splunk-on-ubuntu-14-04-lts

PUBLISHER	OFFER	SKU
starwind	starwindvirtualsan	starwindbyol
starwind	starwindvtl	starwindvtl
stonefly	stonefly-cloud-drive	byol_stonefly
SUSE	openSUSE-Leap	42.3
SUSE	SLES	11-SP4
SUSE	SLES	12-SP3
SUSE	SLES-BYOS	11-SP4
SUSE	SLES-BYOS	12-SP3
SUSE	SLES-HPC	12-SP3
SUSE	SLES-SAP-BYOS	12-SP1
SUSE	SLES-SAP-BYOS	12-SP2
SUSE	SLES-SAP-BYOS	12-SP3
SUSE	SLES-SAPCAL	11-SP4
SUSE	SUSE-CaaSP-Admin-BYOS	2.1
SUSE	SUSE-CaaSP-Cluster-BYOS	2.1
SUSE	SUSE-Manager-Proxy-BYOS	3.0
SUSE	SUSE-Manager-Proxy-BYOS	3.1
SUSE	SUSE-Manager-Server-BYOS	3.0
SUSE	SUSE-Manager-Server-BYOS	3.1
suse-byos	sles-byos	12-sp1
tableau	tableau-server	bring-your-own-license
talon	talon-fast	talon-azure-byol
tenable	tenable-nessus-6-byol	tenable-nessus-byol
tenable	tenablecorenessus	tenablecorenessusbyol
teradata	teradata-data-mover	teradata-data-mover-agent-byol

PUBLISHER	OFFER	SKU
teradata	teradata-data-mover	teradata-data-mover-byol
teradata	teradata-data-stream-controller	teradata-data-stream-controller-byol
teradata	teradata-database-1510-byol	teradata-database-advanced-1510-byol
teradata	teradata-database-1510-byol	teradata-database-base-1510-byol
teradata	teradata-database-1510-byol	teradata-database-enterprise-1510-byol
teradata	teradata-database-1620-byol	teradata-database-advanced-1620-byol
teradata	teradata-database-1620-byol	teradata-database-base-1620-byol
teradata	teradata-database-1620-byol	teradata-database-enterprise-1620-byol
teradata	teradata-querygrid-manager	teradata-querygrid-manager
teradata	teradata-querygrid-manager-intellisphere	teradata-querygrid-manager-intellisphere
teradata	teradata-rest-services	teradata-rest-services-byol
teradata	teradata-server-management	teradata-server-management-byol
teradata	teradata-viewpoint	teradata-viewpoint-multiple-systems-byol
teradata	teradata-viewpoint	teradata-viewpoint-single-system-byol
teradata	teradata-viewpoint	teradata-viewpoint-single-system-data-lab-byol
teradata	teradata-viewpoint-intellisphere	teradata-viewpoint-intellisphere
thales-vormetric	ciphertrust-ckm	ciphertrust-ckm
thales-vormetric	vormetric-dsm	dsm-6-0-2-5162
thales-vormetric	vts-2_2_0_2604	vts-2_2_0_2604
veeam	veeam-backup-replication	veeam-backup-replication-95
veeam	veeam-cloud-connect-enterprise	veeamcloudcconnectenterprise
veeam	veeamcloudconnect	veeambackup
velocitydb-inc	velocitydb	velocitydb

PUBLISHER	OFFER	SKU
veritas	netbackup-8-0	netbackup_8-standard
veritas	netbackup-8-0	netbackup_8_1-standard
vidizmo	c962d038-826e-4c7f-90d9-a2d7ebb50d0c	vidizmo-appdb-single
vidizmo	vidizmo-highavailability-servers	vidizmo-application
vidizmo	vidizmo-separate-servers	vidizmo-application
vidizmo	vidizmo-separate-servers	vidizmo-database
websense-apmailpe	forcepoint-email-security-85beta	forcepoint_email_security_v85_beta
winmagic_securedoc_cloudvm	seccuredoc_cloudvm_5	winmagic_securedoc_cloudvm_byol
wowza	wowzastreamingengine	linux-byol
wowza	wowzastreamingengine	windows-byol
zerto	zerto-cloud-appliance-50	zerto60ga
zerto	zerto-cloud-appliance-50	zerto60u1ga

Next steps

- Create a Windows virtual machine with the Azure portal
- Create a Windows virtual machine with PowerShell
- Create a Windows virtual machine with the Azure CLI
- Create a Linux virtual machine with the Azure portal

Azure Government virtual machine extensions

7/13/2018 • 2 minutes to read • Edit Online

This document contains a list of available virtual machine extensions in Azure Government. If you'd like to see other extensions in Azure Government, please request them via the Azure Government Feedback Forum.

Virtual machine extensions

The list of virtual machine extensions available in Azure Government can be obtained by connecting to Azure Government via PowerShell and running the following commands:

```
Connect-AzureRmAccount -Environment AzureUSGovernment

Get-AzureRmVmImagePublisher -Location USGovVirginia | `
Get-AzureRmVMExtensionImageType | `
Get-AzureRmVMExtensionImage | Select Type, Version
```

The table below contains a snapshot of the list of extensions available in Azure Government as of July 13, 2018.

EXTENSION	VERSIONS
ADETest	1.4.0.2
AzureCATExtensionHandler	2.2.0.68
AzureDiskEncryption	1.1.0.1
AzureDiskEncryptionForLinux	0.1.0.999195; 0.1.0.999196; 0.1.0.999283; 0.1.0.999297
AzureDiskEncryptionForLinuxTest	0.1.0.999321
AzureEnhancedMonitorForLinux	2.0.0.2; 3.0.1.0
BGInfo	2.1
ChefClient	1210.12.110.1000
CustomScript	2.0.2
CustomScriptExtension	1.2; 1.3; 1.4; 1.7; 1.8; 1.9.1
CustomScriptForLinux	1.0; 1.1; 1.2.2.0; 1.3.0.2; 1.4.1.0; 1.5.2.0
DSC	2.19.0.0; 2.22.0.0; 2.23.0.0; 2.24.0.0; 2.26.0.0; 2.26.1.0; 2.71.0.0; 2.72.0.0; 2.73.0.0; 2.76.0.0
DSCForLinux	1.0.0.0; 2.0.0.0; 2.70.0.4
IaaSAntimalware	1.3.0.0; 1.5.4.4

EXTENSION	VERSIONS
IaaSAutoPatchingForWindows	1.0.1.14
IaaSDiagnostics	1.4.3.0; 1.7.4.0; 1.9.0.0
JsonADDomainExtension	1.3; 1.3.2
Linux	1.0.0.9101; 1.0.0.9102
LinuxChefClient	1210.12.109.1005; 1210.12.110.1000
LinuxDEBIAN7	1.0.0.9101; 1.0.0.9102
LinuxDEBIAN8	1.0.0.9100; 1.0.0.9101; 1.0.0.9102
LinuxDiagnostic	2.0.9005; 2.1.9005; 2.2.9005; 2.3.9005; 2.3.9007; 2.3.9011; 2.3.9013; 2.3.9015; 2.3.9017; 2.3.9021
LinuxOL6	1.0.0.9101; 1.0.0.9102
LinuxRHEL6	1.0.0.9101; 1.0.0.9102
LinuxRHEL7	1.0.0.9101; 1.0.0.9102
LinuxSLES11SP3	1.0.0.9101; 1.0.0.9102
LinuxSLES11SP4	1.0.0.9101; 1.0.0.9102
LinuxSLES12	1.0.0.9102
LinuxUBUNTU1404	1.0.0.9100; 1.0.0.9101; 1.0.0.9102
LinuxUBUNTU1604	1.0.0.9100; 1.0.0.9101; 1.0.0.9102
MicrosoftMonitoringAgent	1.0.11030.0; 1.0.11030.1; 1.0.11030.2; 1.0.11049.1
NetworkWatcherAgentLinux	1.4.270.0; 1.4.306.5; 1.4.411.1; 1.4.493.1; 1.4.526.2; 1.4.585.2
NetworkWatcherAgentWindows	1.4.270.0; 1.4.306.5; 1.4.411.1; 1.4.493.1; 1.4.526.2; 1.4.585.2
OmsAgentForLinux	1.2.75.0; 1.4.45.2
OSPatchingForLinux	1.0.1.1; 2.0.0.5; 2.1.0.0; 2.2.0.0; 2.3.0.1
RDMAUpdateForLinux	0.1.0.9
SqlIaaSAgent	1.2.11.0; 1.2.15.0; 1.2.16.0; 1.2.17.0; 1.2.18.0
VMAccessAgent	2.0; 2.0.2; 2.3; 2.4.2; 2.4.4
VMAccessForLinux	1.0; 1.1; 1.2; 1.3.0.1; 1.4.0.0; 1.4.5.0

EXTENSION	VERSIONS
VMBackupForLinuxExtension	0.1.0.995; 0.1.0.993
VMJITAccessExtension	1.0.0.0; 1.0.1.0
VMSnapshot	1.0.22.0; 1.0.23.0; 1.0.26.0; 1.0.27.0; 1.0.40.0; 1.0.41.0; 1.0.42.0
VMSnapshotLinux	1.0.9111.0; 1.0.9112.0; 1.0.9117.0; 1.0.9118.0; 1.0.9128.0; 1.0.9131.0
VSRemoteDebugger	1.1.3.0
Windows	1.0.0.9100; 1.0.0.9101; 1.0.0.9102

Next steps

- Deploy a Windows virtual machine extension
- Deploy a Linux virtual machine extension

Publishing to the Azure Government Marketplace

7/16/2018 • 4 minutes to read • Edit Online

This article is provided to help partners create, deploy, and manage their solutions listed in the Azure Government Marketplace for Azure Government customers and partners to use.

Why publish to Azure Government

Azure Government is a dedicated instance of Azure that employs world-class security and compliance services critical to U.S. government for all systems and applications built on its architecture. This makes the cloud a viable option for thousands of US federal, state, local and tribal governments, and their partners.

Publishing your solution in the Azure Government Marketplace is as simple as publishing to Azure global and checking an extra box. There are no compliance requirements to publish your solution to Azure Government and making it available in the Azure Government Marketplace makes it easier for these government customers to gain exposure to your solution and get up and running quickly.

Compliance considerations

There are no initial Microsoft compliance requirements to publish solutions to the Azure Government Marketplace.

Once a solution has been published, customers can deploy it into their own subscription as part of a broader operational environment or business solution. The customer might then opt to certify the overarching environment. As part of that certification process, they might reach out to the publisher with extra requirements, which the publisher can then evaluate and triage with the customer.

Marketplace offer support

Currently, the Azure Government Marketplace only supports the following offers:

- Virtual Machines > Bring your Own License
- Virtual Machines > Pay-as-you-Go
- Azure Application > Solution Template

If there are other offer types you'd like to see supported in Azure Government, let us know via the Azure Government feedback forum.

Publishing

> **NOTE**
> These steps assume you have already published a solution in Azure Global. If you haven't, please check out the Azure Marketplace Publisher Guide documentation before proceeding.

1. **Sign in to the Azure Cloud Partner Portal**.
2. **Open the offer** you want to publish to the Azure Government Marketplace.
3. The Editor tab is opened by default, click on the **SKUs** entry in the left menu of the editor.
4. **Click on the sku**.
5. In the **SKU Details section**, **Cloud Availability option**, check the **Azure Government Cloud** box. Remember that this option isn't available for all offers.

6. ***Optionally***, click the **+ Add Certification** link to add links to any certifications that are relevant for your product and that you want to make available to customers.
7. ***Optionally***, add your Azure Government subscription to preview your marketplace offering before it is broadly available.
   a. Click on **Marketplace** entry in the left menu
   b. In the **Preview Subscription Ids section**, click on **Add subscription** and add your Azure Government subscription ID.
8. **Publish** your solution once again.

Testing

If you want to confirm that your solution has been published or test it, you need to request an Azure Government account. This is a separate account from any account in Azure Global that is used to log in to the Azure Government portal.

To obtain an account:

1. Request an Azure Government trial account.
   - Indicate that your organization is a *Solution Provider Serving U.S. Federal, State, Local or Tribal Government Entities*.
2. Wait for 3 - 5 business days for your account to be provisioned.
3. Log in to the Azure Government portal with your newly created account.
4. Eventually you can convert your trial account to a paid account

Troubleshooting

Generally, virtual machines and solution templates work across both Azure and Azure Government, however there are a few instances when this is not the case. The following section outlines the most common reasons why a virtual machine or solution template would work in the Azure Marketplace but not the Azure Government Marketplace.

Not available after publish

If you've completed all the steps outlined above and your virtual machine is still not available in the Azure Government Marketplace, make sure that your Virtual's Machine *Hide this SKU* setting is not set to *Yes*. If it is set to yes, there's probably also a solution template that you also need to publish to Azure Government. If there is no solution template and you want to make the standalone Virtual Machine available, flip that switch to *No* and republish.

Hardcoded endpoints

Verify endpoints are not hard-coded into your solution Template for Azure Global as they will not be valid for any other Azure clouds (Azure Government, Azure China, Azure Germany). Instead modify the Solution template to obtain the endpoint from the resource, for example:

- Incorrect VHD uri (hard coded)

```
"uri": "[concat('https://', variables('storageAccountName'), '.blob.core.windows.net/',
'/osdisk.vhd')]",
```

- Correct VHD uri (referenced)

```
"uri": "[concat(reference(resourceId('Microsoft.Storage/storageAccounts/',
variables('storageAccountName'))).primaryEndpoints.blob, 'osdisk.vhd')]",
```

Hardcoded list of locations

Make sure your solution template supports the Azure Government locations. See the list of Azure Government regions

Unavailable resources

Verify that resources, API versions, extensions and VM images used in your solution template are available in Azure Government.

Images

Make sure that the image that your solution template relies on is available in Azure Government. If this is a Virtual Machine you own, need to also publish that to the Azure Government Marketplace. Check out the Azure Government Marketplace images documentation to obtain the list of images available.

Resource providers and API versions

You can obtain the full list of resource providers and their API versions by logging in to the Azure Government portal using your Azure Government account and following the steps listed in the Resource providers and types documentation.

Extensions

Make sure that your any virtual machine extensions that your solution template relies on is available in Azure Government. Check out the Azure Government virtual machine extensions documentation to obtain the list of extensions available.

Next steps

- Subscribe to the Azure Government blog
- Get help on Stack Overflow by using the azure-gov tag
- Give us feedback or request new features via the Azure Government feedback forum

Azure Government Compute

4/9/2018 • 2 minutes to read • Edit Online

Virtual Machines

For details on this service and how to use it, see Azure Virtual Machines Sizes.

Variations

For available virtual machine sizes in Azure Government, see Products Available by Region

Data Considerations

The following information identifies the Azure Government boundary for Azure Virtual Machines:

REGULATED/CONTROLLED DATA PERMITTED	REGULATED/CONTROLLED DATA NOT PERMITTED
Data entered, stored, and processed within a VM can contain export-controlled data. Binaries running within Azure Virtual Machines. Static authenticators, such as passwords and smartcard PINs for access to Azure platform components. Private keys of certificates used to manage Azure platform components. SQL connection strings. Other security information/secrets, such as certificates, encryption keys, master keys, and storage keys stored in Azure services.	Metadata is not permitted to contain export-controlled data. This metadata includes all configuration data entered when creating and maintaining your Azure Virtual Machine. Do not enter Regulated/controlled data into the following fields: Tenant role names, Resource groups, Deployment names, Resource names, Resource tags

Virtual Machine Scale Sets

For details on this service and how to use it, see Azure Virtual Machine Scale Sets documentation.

Variations

The only variation is the available sizes of Virtual Machines in Azure Government.

Batch

For details on this service and how to use it, see Azure Batch documentation.

Variations

The URLs for accessing and managing the Batch service are different:

SERVICE TYPE	AZURE PUBLIC	AZURE GOVERNMENT
Batch	*.batch.azure.com	*.batch.usgovcloudapi.net

Cloud Services

For details on this service and how to use it, see Azure Cloud Services documentation.

Variations

The DNS for the Cloud Services is different:

SERVICE TYPE	AZURE PUBLIC	AZURE GOVERNMENT
Batch	*.cloudapp.net	*.usgovcloudapp.net

Azure Functions

The Azure Functions service is now available (General Availability) for the Azure Government environment, with some differences, which you can read about below.

Variations

The following Functions features are not currently available in Azure Government:

- The App Service plan is available in Azure Government. The Consumption plan is not available yet. To learn more about the two hosting plans, click here
- Monitoring via Application Insights is not available yet.

The URLs for Function are different:

SERVICE TYPE	AZURE PUBLIC	AZURE GOVERNMENT
Functions	.azurewebsites.net	.azurewebsites.us

Service Fabric

For details on this service and how to use it, see Azure Service Fabric documentation.

Next Steps

For supplemental information and updates, subscribe to the Microsoft Azure Government Blog.

Azure Government Networking

5/24/2018 • 6 minutes to read • Edit Online

ExpressRoute (Private Connectivity)

ExpressRoute is generally available in Azure Government. For more information (including partners and peering locations), see the ExpressRoute public documentation.

Variations

ExpressRoute is generally available (GA) in Azure Government.

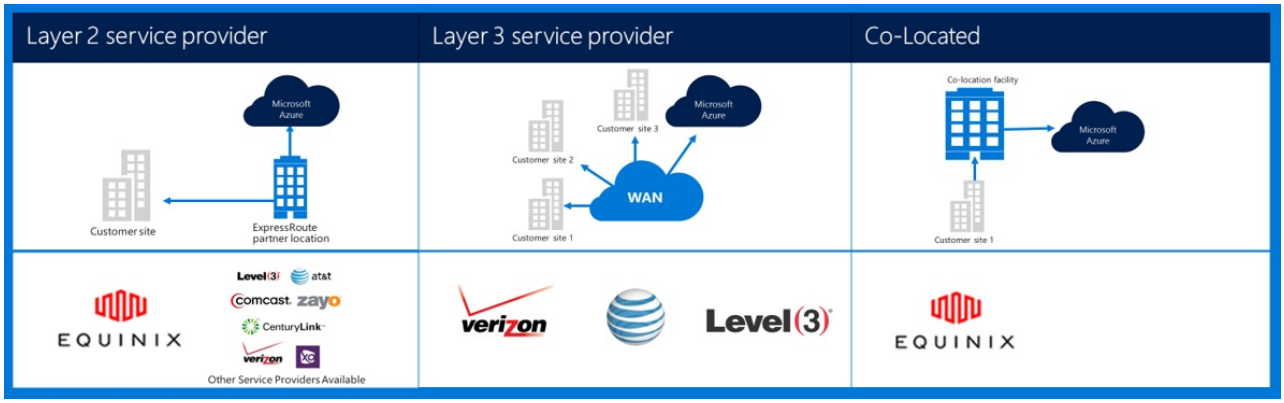
- Government customers connect to a physically isolated capacity over a dedicated Azure Government (Gov) ExpressRoute (ER) connection
- Azure Gov provides Increased availability & durability by leveraging multiple region pairs located a minimum of 500 miles apart
- By default all Azure Gov ER connectivity is configured active-active redundant with support for bursting and delivers up to 10G circuit capacity (smallest is 50 MB)
- Azure Gov ER locations provide optimized pathways (shortest hops, low latency, high performance, etc.) for customers and Azure Gov geo-redundant regions
- The Azure Gov ER private connection does not utilize, traverse, or depend on the Internet
- Azure Gov physical and logical infrastructure are physically dedicated and separated, and access is restricted to U.S. persons
- Microsoft owns and operates all fiber infrastructure between Azure Gov Regions and Azure Gov ER Meet-Me locations
- Azure Gov ER provides connectivity to Microsoft Azure, O365, and CRM cloud services

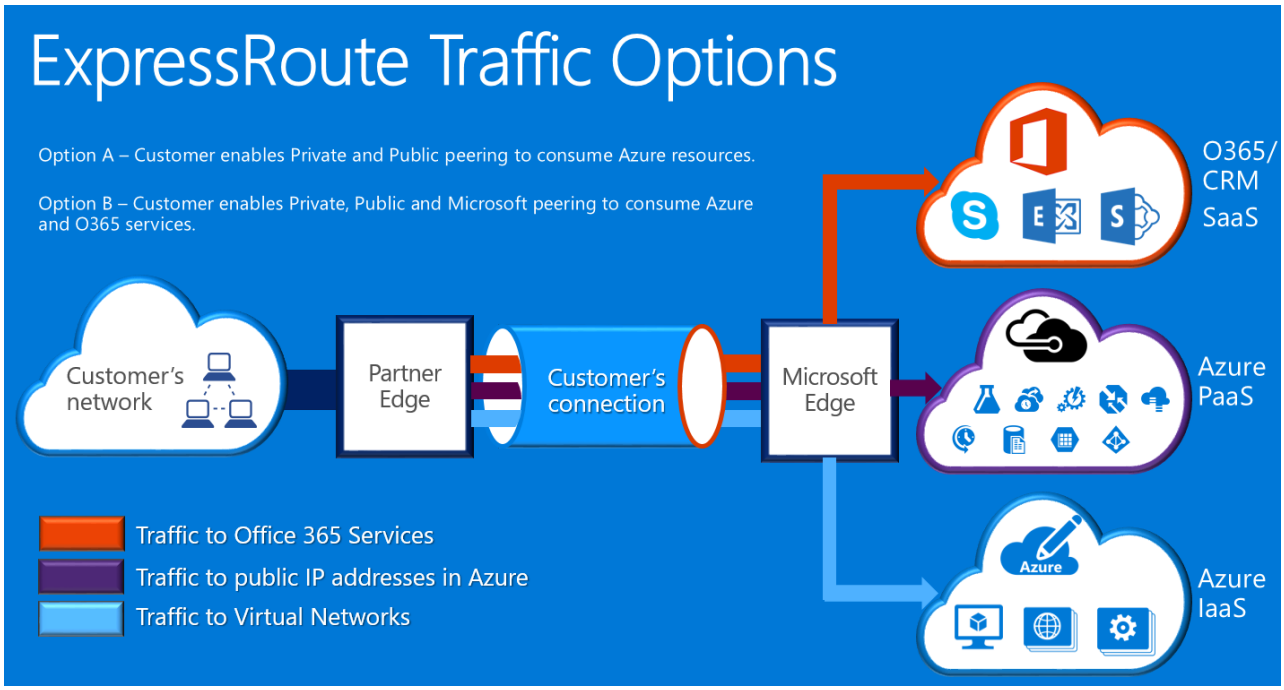
Considerations

There are two basic services that provide private network connectivity into Azure Government: VPN (site-to-site for a typical organization) and ExpressRoute.

Azure ExpressRoute is used to create private connections between Azure Government datacenters, and your on-premises infrastructure, or in a colocation environment. ExpressRoute connections do not go over the public Internet—they offer more reliability, faster speeds, and lower latencies than typical Internet connections. In some cases, using ExpressRoute connections to transfer data between on-premises systems and Azure yields significant cost benefits.

With ExpressRoute, you establish connections to Azure at an ExpressRoute location (such as an Exchange provider facility), or you directly connect to Azure from your existing WAN network (such as a multiprotocol label switching (MPLS) VPN, supplied by a network service provider).





For network services to support Azure Government customer applications and solutions, it is strongly recommended that ExpressRoute (private connectivity) is implemented to connect to Azure Government. If VPN connections are used, the following should be considered:

- Customers should contact their authorizing official/agency to determine whether private connectivity or other secure connection mechanism is required and to identify any additional restrictions to consider.
- Customers should decide whether to mandate that the site-to-site VPN is routed through a private connectivity zone.
- Customers should obtain either an MPLS circuit or VPN with a licensed private connectivity access provider.

All customers who utilize a private connectivity architecture should validate that an appropriate implementation is established and maintained for the customer connection to the Gateway Network/Internet (GN/I) edge router demarcation point for Azure Government. Similarly, your organization must establish network connectivity between your on-premises environment and Gateway Network/Customer (GN/C) edge router demarcation point for Azure Government.

Data Considerations

The following information identifies the Azure Government International Traffic in Arms Regulations (ITAR) boundary for Azure ExpressRoute:

REGULATED/CONTROLLED DATA PERMITTED	REGULATED/CONTROLLED DATA NOT PERMITTED
Data entered, transmitted, and processed within ExpressRoute dedicated connections can contain export-controlled data.	Azure ExpressRoute metadata is not permitted to contain export-controlled data. This metadata includes configuration data entered when creating and maintaining your ExpressRoute circuit. Do not enter regulated/controlled data into the Circuit name field when configuring the initial ExpressRoute circuit.

Support for BGP communities

This section provides an overview of how BGP communities will be used with ExpressRoute in AzureGov. Microsoft will advertise routes in the public and Microsoft peering paths with routes tagged with appropriate community values. The rationale for doing so and the details on community values are described below. Microsoft, however, will not honor any community values tagged to routes advertised to Microsoft.

If you are connecting to Microsoft through ExpressRoute at any one peering location within the AzureGov region, you will have access to all Microsoft cloud services across all regions within the government boundary.

For example, if you connected to Microsoft in Washington D.C. through ExpressRoute, you will have access to all Microsoft cloud services hosted in AzureGov.

Refer to the "Overview" tab on ExpressRoute public documentation for details on locations and partners, and a detailed list of ExpressRoute for AzureGov peering locations.

You can purchase more than one ExpressRoute circuit. Having multiple connections offers you significant benefits on high availability due to geo-redundancy. In cases where you have multiple ExpressRoute circuits, you will receive the same set of prefixes advertised from Microsoft on the public peering and Microsoft peering paths. This means you will have multiple paths from your network into Microsoft. This can potentially cause sub-optimal routing decisions to be made within your network. As a result, you may experience sub-optimal connectivity experiences to different services.

Microsoft will tag prefixes advertised through public peering and Microsoft peering with appropriate BGP community values indicating the region the prefixes are hosted in. You can rely on the community values to make appropriate routing decisions to offer optimal routing to customers. For additional details, refer to the "How-to guides > Best practices" tab on ExpressRoute public documentation and click on "Optimize routing."

NATIONAL CLOUDS AZURE REGION	BGP COMMUNITY VALUE
US Government	
US Gov Arizona	12076:51106
US Gov Iowa	12076:51109
US Gov Virginia	12076:51105
US Gov Texas	12076:51108
US DoD Central	12076:51209
US DoD East	12076:51205

All routes advertised from Microsoft will be tagged with the appropriate community value.

In addition to the above, Microsoft will also tag prefixes based on the service they belong to. This applies only to the Microsoft peering. The table below provides a mapping of service to BGP community value.

SERVICE IN NATIONAL CLOUDS	BGP COMMUNITY VALUE
US Government	
Exchange Online	12076:5110
SharePoint Online	12076:5120
Skype For Business Online	12076:5130
Dynamics 365	12076:5140

SERVICE IN NATIONAL CLOUDS	BGP COMMUNITY VALUE
Other Office 365 Online services	12076:5200

> **NOTE**
> Microsoft does not honor any BGP community values that you set on the routes advertised to Microsoft.

Support for Virtual Network

Virtual Network is generally available in Azure Government. For more information, see the Virtual Network public documentation.

Support for Load Balancer

Load Balancer is generally available in Azure Government. For more information, see the Load Balancer public documentation.

Support for DNS

DNS is generally available in Azure Government. For more information, see the DNS public documentation.

Support for Traffic Manager

Traffic Manager is generally available in Azure Government. For more information, see the Traffic Manager public documentation.

The **IP addresses for Azure Government from which Traffic Manager health checks can originate are here**. Review the IPs listed in the JSON file to ensure that incoming connections from these IP addresses are allowed at the endpoints to check its health status.

Support for VNet Peering

VNet Peering is generally available in Azure Government. For more information, see the VNet Peering public documentation.

Support for VPN Gateway

VPN Gateway is generally available in Azure Government. For more information, see the VPN Gateway public documentation.

Support for Application Gateway

Application Gateway is generally available in Azure Government. For more information, see the Application Gateway public documentation.

Support for Network Watcher

Network Watcher is generally available in Azure Government. For more information, see the Network Watcher public documentation.

Support for Service Bus

Service Bus is generally available in Azure Government. For more information, see the Service Bus public documentation.

Variations

The URLs for accessing and managing the Service Bus service are different:

SERVICE TYPE	AZURE PUBLIC	AZURE GOVERNMENT
Service Bus	*.servicebus.windows.net	*.servicebus.usgovcloudapi.net

Next Steps

For supplemental information and updates please subscribe to the Microsoft Azure Government Blog.

Azure Government storage

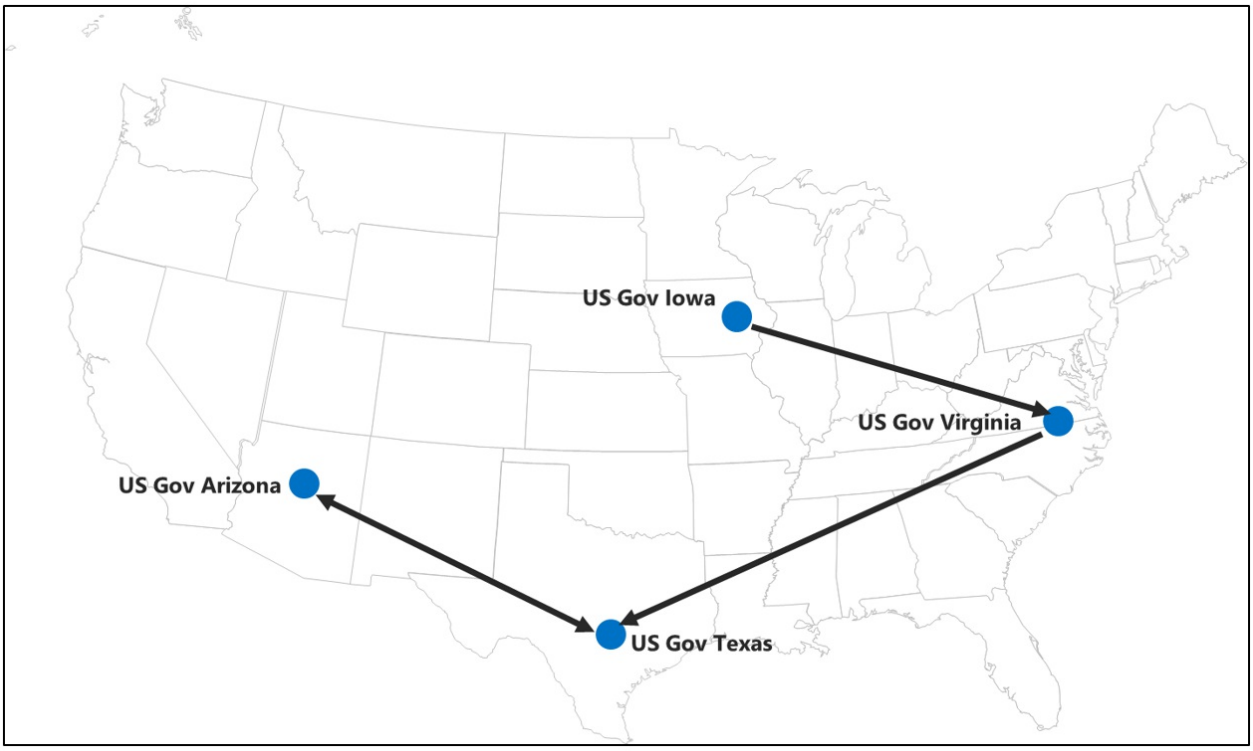
4/9/2018 • 4 minutes to read • Edit Online

Azure Storage

Azure Storage is generally available in Azure Government. For a Quickstart that will help you get started with Storage in Azure Government, click here. For general details on Azure Storage, see Azure Storage public documentation.

Storage pairing in Azure Government

The following map shows the primary and secondary region pairings used for geo-redundant storage and read-access geo-redundant storage accounts in Azure Government.

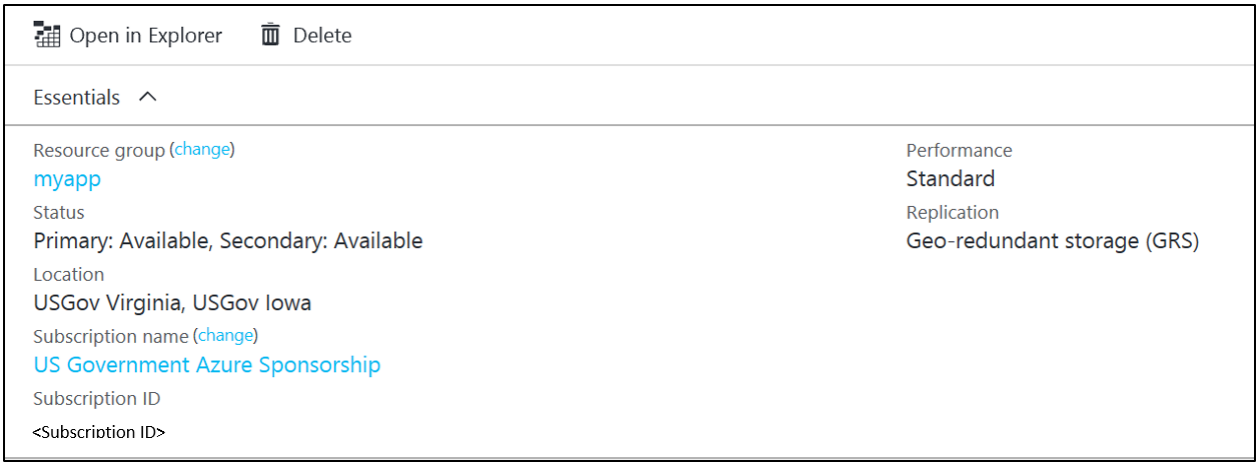


NOTE

The USGov Virginia secondary region is USGov Texas. Previously, USGov Virginia used USGov Iowa as a secondary region. Storage accounts with USGov Iowa as a secondary region are being migrated to USGov Texas as a secondary region.

Checking the secondary region for a storage account

To view the current secondary region of your geo-redundant storage or read-access geo-redundant storage account through the Azure portal, select the storage account on the left. Select the name of the storage account to bring up the storage account overview that lists the primary and secondary regions.



Storage service and feature availability by Azure Government region

SERVICE OR FEATURE	USGOV VIRGINIA	USGOV IOWA	USGOV ARIZONA	USGOV TEXAS	USDOD EAST	USDOD CENTRAL
Blob storage	GA	GA	GA	GA	GA	GA
Azure Files	GA	GA	GA	GA	GA	GA
Table storage	GA	GA	GA	GA	GA	GA
Queue storage	GA	GA	GA	GA	GA	GA
Hot/cool blob storage	GA	-	GA	GA	-	-
Locally redundant storage	GA	GA	GA	GA	GA	GA
Geo-redundant storage	GA	GA	GA	GA	GA	GA
Read-access geo-redundant storage	GA	GA	GA	GA	GA	GA
Zone-redundant storage	-	GA	GA	GA	GA	GA
Storage Service Encryption	GA	GA	GA	GA	GA	GA
Premium Storage	GA	-	GA	GA	GA	GA
StorSimple	GA	GA	GA	GA	GA	GA

Variations

These are the URLs for storage accounts in Azure Government:

SERVICE TYPE	AZURE PUBLIC	AZURE GOVERNMENT
Blob storage	*.blob.core.windows.net	*.blob.core.usgovcloudapi.net
Queue storage	*.queue.core.windows.net	*.queue.core.usgovcloudapi.net
Table storage	*.table.core.windows.net	*.table.core.usgovcloudapi.net
File storage	*.file.core.windows.net	*.file.core.usgovcloudapi.net

> **NOTE**
> All your scripts and code need to account for the appropriate endpoints. See Configure Azure Storage Connection Strings.

For more information on APIs, see the Cloud Storage Account Constructor.

The endpoint suffix to use in these overloads is *core.usgovcloudapi.net*.

> **NOTE**
> If error 53 ("The network path was not found") is returned while you're mounting the file share, a firewall might be blocking the outbound port. Try mounting the file share on VM that's in the same Azure subscription as the storage account.

When you're deploying the StorSimple Manager service, use the https://portal.azure.us/ URL for the Azure Government portal. For deployment instructions for StorSimple Virtual Array, see StorSimple Virtual Array system requirements. For the StorSimple 8000 series, see StorSimple software, high availability, and networking requirements and go to the **Deploy** section from the left menu. For more information on StorSimple, see the StorSimple documentation.

Considerations

The following information identifies the Azure Government boundary for Azure Storage:

REGULATED/CONTROLLED DATA PERMITTED	REGULATED/CONTROLLED DATA NOT PERMITTED
Data that's entered, stored, and processed within an Azure Storage product can contain export-controlled data. This data includes static authenticators, such as passwords and smartcard PINs for access to Azure platform components. It also includes private keys of certificates used to manage Azure platform components. And it includes other security information/secrets, such as certificates, encryption keys, master keys, and storage keys stored in Azure services.	Azure Storage metadata cannot contain controlled data. This metadata includes all configuration data that's entered when you're creating and maintaining your storage product. Do not enter regulated/controlled data in the following fields: **Resource groups**, **Deployment names**, **Resource names**, **Resource tags**.

Azure Import/Export

Azure Import/Export is generally available for Azure Government. All Azure Government regions are supported. To create Import/Export jobs, see the Azure Import/Export documentation.

Variations

With Import/Export jobs for USGov Arizona or USGov Texas, the mailing address is for USGov Virginia. The data is loaded into selected storage accounts from the USGov Virginia region.

Considerations

For DoD L5 data, use a DoD region storage account to ensure that data is loaded directly into the DoD regions.

For all jobs, we recommend that you rotate your storage account keys after the job is complete to remove any access granted during the process. For more information, see Managing storage accounts.

REGULATED/CONTROLLED DATA PERMITTED	REGULATED/CONTROLLED DATA NOT PERMITTED
Data copied to the media for transport and the keys used to encrypt that data.	Azure Import/Export metadata cannot contain controlled data. This metadata includes all configuration data that's entered when you're creating your Import/Export job and shipping information that's used to transport your media. Do not enter regulated/controlled data in the following fields: **Job name**, **Carrier name**, **Tracking number**, **Description**, **Return information (Name, Address, Phone, E-Mail)**, **Export Blob URI**, **Drive list**, **Package list**, **Storage account name**, **Container name**.

Azure Backup Service

For detailed documentation on using the Azure Backup Service in Azure Government, click here.

Next steps

For supplemental information and updates, subscribe to the Microsoft Azure Government blog.

Azure Government Web + Mobile

5/21/2018 • 2 minutes to read • Edit Online

App Services

Variations

Azure App Services is generally available in Azure Government.

The Address for Azure App Service apps created in Azure Government is different from those apps created in the public cloud:

SERVICE TYPE	AZURE PUBLIC	AZURE GOVERNMENT
App Service	*.azurewebsites.net	*.azurewebsites.us
Service Principal ID	abfa0a7c-a6b6-4736-8310-5855508787cd	6a02c803-dafd-4136-b4c3-5a6f318b4714

Some App Service features available in Azure Government have variations:

- Deployment Options are limited to local git and external git.

Some App Service features available in the public cloud are not yet available in Azure Government:

- App Service Certificates
- Settings
  - Managed service identity > Vote for this
  - Push notifications
  - Security scanning
- Development Tools
  - Performance test
  - Resource explorer
  - PHP Debugging
- Monitoring
  - Application Insights
  - Metrics per instance
  - Live HTTP traffic
  - Application events
  - FREB logs
- Support & Troubleshooting
  - App Service Advisor
  - Failure History
  - Diagnostics as a Service
  - Mitigate

Considerations

The following information identifies the Azure Government boundary for App Service:

REGULATED/CONTROLLED DATA PERMITTED	REGULATED/CONTROLLED DATA NOT PERMITTED
Data entered, stored, and processed within Azure App Service can contain export-controlled data. Binaries running within Azure App Service. Static authenticators, such as passwords and smartcard PINs for access to Azure platform components. Private keys of certificates used to manage Azure platform components. SQL connection strings. Other security information/secrets, such as certificates, encryption keys, master keys, and storage keys stored in Azure services.	Metadata is not permitted to contain export-controlled data. This metadata includes all configuration data entered when creating and maintaining your Azure App Service. Do not enter Regulated/controlled data into the following fields: Resource groups, Resource names, Resource tags

API Management

For details on this service and how to use it, see Azure API Management documentation.

Variations

Azure API Management service is generally available in Azure Government. Features that are not currently available in API Management service for Azure Government are:

- Azure AD B2C Integration
  - Integration with Azure AD B2C is not available in Azure Government

The URLs for accessing Azure API Management in Azure Government are different:

SERVICE TYPE	AZURE PUBLIC	AZURE GOVERNMENT
API Management gateway	*.azure-api.net	*.azure-api.us
API Management portal	*.portal.azure-api.net	*.portal.azure-api.us
API Management management	*.management.azure-api.net	*.management.azure-api.us

Considerations

The following information identifies the Azure Government boundary for Azure API Management service:

REGULATED/CONTROLLED DATA PERMITTED	REGULATED/CONTROLLED DATA NOT PERMITTED
All data stored and processed in Azure API Management service can contain Azure Government-regulated data.	Azure API Management service metadata is not permitted to contain export-controlled data. Do not enter regulated/controlled data into the following fields: API Management service name, Subscription name, Resource groups, Resource tags.

Next Steps

For supplemental information and updates, subscribe to the Microsoft Azure Government Blog.

Azure Stack for Azure Government Customers

4/25/2018 • 2 minutes to read • Edit Online

With Azure Stack now available, it is important to understand what capabilities it brings to our Azure Government community. Understanding the features and capabilities and their alignment existing services and offerings, will help guide some important decisions in the cloud transformation journey.

Usage scenarios – What can I do with Azure Stack?

Below are some examples scenarios that describe what capabilities exist today for Azure Government customers. You can get started with these as soon as your hardware is available. These will enable planning, development, and implementation of on-premises or hybrid workloads. This documentation will be updated as additional features, capabilities, and scenarios become available.

Edge and Disconnected Scenarios

These are completely isolated and secure environments, containing critical/sensitive mission workloads while retaining Azure's platform capabilities. A demo with additional information can be viewed here.

Cloud Apps on-premises

Build applications that run on the Azure platform and implement consistent engineering and DevOps processes. Transform your secure mission workload and begin moving to the cloud with the comfort of on-premises isolation.

Useful Resources

Below are links to useful content, guidance, and planning resources to help become familiar with Azure Stack.

- Azure Stack Use Cases
- Azure Stack - Whitepaper
- Azure Stack Documentation Home Page
- Plan for your Azure Stack Journey
- Key Features, Concepts, and Terms
- How to add an Image (Custom or 3rd Party

Azure Government Media Services

5/7/2018 • 2 minutes to read • Edit Online

For details on this service and how to use it, see the Azure Media Services documentation.

Azure Media Services (AMS) is currently generally available in Azure Government.

Connecting

For information on how to connect to AMS, see connecting to AMS

When connecting to Media Services in Azure Government, use the following values:

ACS scope

The context scope should be set to "urn:WindowsAzureMediaServices".

ACS

The ACS base address should be set to "https://ams-usge-0-acs-global-1-1.accesscontrol.usgovcloudapi.net"

REST endpoints

The API server address depends on what region you are deployed to:

- US Gov Virginia: "https://ams-usge-1-hos-rest-1-1.usgovcloudapp.net/API/"
- US Gov Iowa: "https://ams-usgc-1-hos-rest-1-1.usgovcloudapp.net/API/"

Analyzing

The "Azure Media Indexer 2 Preview" Azure Media Analytics media processor is not available in Azure Government.

CDN integration

There is no CDN integration with streaming endpoints in Azure Government DCs.

Next Steps

For supplemental information and updates, subscribe to the Microsoft Azure Government Blog.

Azure Government Databases

4/25/2018 • 2 minutes to read • Edit Online

SQL Database

For more information, see the Microsoft Security Center for SQL Database Engine and Azure SQL Database documentation for additional guidance on metadata visibility configuration, and protection best practices.

Variations

SQL V12 Database is generally available in Azure Government.

The Address for SQL Azure Servers in Azure Government is different:

SERVICE TYPE	AZURE PUBLIC	AZURE GOVERNMENT
SQL Database	*.database.windows.net	*.database.usgovcloudapi.net

Considerations

The following information identifies the Azure Government boundary for Azure SQL:

REGULATED/CONTROLLED DATA PERMITTED	REGULATED/CONTROLLED DATA NOT PERMITTED
All data stored and processed in Microsoft Azure SQL can contain Azure Government-regulated data. Use database tools for data transfer of Azure Government-regulated data.	Azure SQL metadata is not permitted to contain export-controlled data. This metadata includes all configuration data entered when creating and maintaining your storage product. Do not enter regulated/controlled data into the following fields: Database name, Subscription name, Resource groups, Server name, Server admin login, Deployment names, Resource names, Resource tags

SQL Data Warehouse

For details on this service and how to use it, see Azure SQL Data Warehouse documentation.

SQL Server Stretch Database

For details on this service and how to use it, see Azure SQL Server Stretch Database documentation

Azure Cosmos DB

For details on this service and how to use it, see Azure Cosmos DB documentation.

Variations

Azure Cosmos DB is generally available in Azure Government. Features that are not currently available in Cosmos DB for Azure Government are:

- **Add Azure Search** - Does not work because Azure Search is not yet deployed in Azure Government.
- **Gremlin API (Graph)** - Cosmos DB accounts using Gremlin cannot be created at this time.

The URLs for accessing Cosmos DB in Azure Government are different:

SERVICE TYPE	AZURE PUBLIC	AZURE GOVERNMENT
Cosmos DB	*.documents.azure.com	*.documents.azure.us

Considerations

The following information identifies the Azure Government boundary for Azure Cosmos DB:

REGULATED/CONTROLLED DATA PERMITTED	REGULATED/CONTROLLED DATA NOT PERMITTED
All data stored and processed in Azure Cosmos DB can contain Azure Government-regulated data.	Azure Cosmos DB metadata is not permitted to contain export-controlled data. Do not enter regulated/controlled data into the following fields: **DB name, Subscription name, Resource groups, Resource tags**.

Azure Redis Cache

For details on this service and how to use it, see Azure Redis Cache documentation.

Variations

The URLs for accessing and managing Azure Redis Cache in Azure Government are different:

SERVICE TYPE	AZURE PUBLIC	AZURE GOVERNMENT
Cache endpoint	*.redis.cache.windows.net	*.redis.cache.usgovcloudapi.net

> **NOTE**
> All scripts and code need to account for the appropriate endpoints and environments. For more information, see How to connect to other clouds.

Considerations

The following information identifies the Azure Government boundary for Azure Redis Cache:

REGULATED/CONTROLLED DATA PERMITTED	REGULATED/CONTROLLED DATA NOT PERMITTED
All data stored and processed in Azure Redis Cache can contain Azure Government-regulated data.	Azure Redis Cache metadata is not permitted to contain export-controlled data. Do not enter regulated/controlled data into the following fields: **Cache name, Subscription name, Resource groups, Resource tags, Redis properties**.

Next Steps

For supplemental information and updates subscribe to the Microsoft Azure Government Blog.

Azure Government Data + Analytics

11/2/2017 • 2 minutes to read • Edit Online

This article outlines the data and analytics services, variations, and considerations for the Azure Government environment.

HDInsight

HDInsight on Linux Standard is generally available in Azure Government. You can see a demo on how to build data-centric solutions on Azure Government using HDInsight.

Variations

The following HDInsight features are not currently available in Azure Government.

- HDInsight is not available on Windows.
- Azure Data Lake Store is not currently available in Azure Government. Azure Blob Storage is the only available storage option currently.

The URLs for Log Analytics are different in Azure Government:

SERVICE TYPE	AZURE PUBLIC	AZURE GOVERNMENT
HDInsight Cluster	*.azurehdinsight.net	*.azurehdinsight.us

For secured virtual networks, you will want to allow Network Security Groups (NSGs) access to certain IP addresses and ports. For Azure Government, you should allow the follow IP addresses (all with an Allowed port of 443):

REGION	ALLOWED IP ADDRESSES	ALLOWED PORT
USGov Virginia	13.72.49.126 13.72.55.55	443
USGov Iowa	13.72.184.124 13.72.190.110	443

For more information, see HDInsight public documentation.

Power BI

Power BI US Government is generally available as part of the Office 365 US Government Community subscriptions. You can learn about Power BI US Government here.

You can see a demo on how to build data-centric solutions on Azure Government using Power BI

Variations

Power BI does not yet have Portal support in the Azure Government Portal.

The URLs for Power BI are different in US Government:

SERVICE TYPE	POWER BI COMMERCIAL	POWER BI US GOVERNMENT
Power BI URL	app.powerbi.com	app.powerbigov.us

Power BI Embedded

For details on this service and how to use it, see Azure Power BI Embedded Documentation.

Variations

Power BI Embedded does not yet have Portal support in the Azure Government Portal.

Azure Analysis Services

For information on this service and how to use it, see Azure Analysis Services Documentation.

Next Steps

For supplemental information and updates, subscribe to the Microsoft Azure Government Blog.

Azure Government AI and Cognitive Services

6/6/2018 • 2 minutes to read • Edit Online

Cognitive Services

The following Cognitive Services APIs are currently in Public Preview in Azure Government.

- Computer Vision API
- Face API
- Translator Speech API
- Translator Text API

Variations

- Provisioning and management for the following APIs are available through PowerShell and CLI only (**no Portal Support**).

Quickstarts

The Azure Government Cognitive Services Quickstart will guide you through getting started with provisioning an account and accessing the APIs.

Vision

Computer Vision API

The following variations exist for Computer Vision API from Commercial Azure:

- Endpoint URL: https://virginia.api.cognitive.microsoft.us/vision/v1.0/
- Available SKUs: F0, S0, S1

For more information, please see public documentation and public API documentation for Computer Vision API.

Face API

The following variations exist for Face API from Commercial Azure:

- Endpoint: https://virginia.api.cognitive.microsoft.us/face/v1.0/
- Available SKUs: F0, S0

For more information, please see public documentation and public API documentation for Face API.

Emotion API

Emotion API is being deprecated and all of its technology has been incorporated to Face API.

Speech

Translator Speech API (Speech Translation):

The following variations exist for Translator Speech API from Commercial Azure:

- Endpoint: https://dev.microsofttranslator.us
- Auth Token Service: https://virginia.api.cognitive.microsoft.us/sts/v1.0/issueToken
- Available SKUs: F0, S1, S2, S3, S4

For more information, please see public documentation and public API documentation for Bing Speech API.

Language

Translator Text API (Text Translation):

The following variations exist for Translator Text API from Commercial Azure:

- Endpoint: https://api.microsofttranslator.us
- Auth Token Service: https://virginia.api.cognitive.microsoft.us/sts/v1.0/issueToken

- Available SKUs: F0, S1, S2, S3, S4
- Translator Hub, Web Widget, and Collaboration Translation Framework (CTF) are not supported.

For more information, please see public documentation and public API documentation for Translator Text API.

Data Considerations

Data considerations for Cognitive Services are not yet available.

Next Steps

- Subscribe to the Azure Government blog
- Get help on Stack Overflow by using the azure-gov tag
- Give us feedback or request new features via the Azure Government feedback forum

Azure Government Internet of Things

5/7/2018 • 2 minutes to read • Edit Online

Azure IoT Hub

Azure IoT Hub is generally available in Azure Government.

For more information, see Azure IoT Hub commercial documentation.

Variations

The following URL for Azure IoT Hub is different in Azure Government:

AZURE PUBLIC	AZURE GOVERNMENT
*.azure-devices.net	*.azure-devices.us

If you are using the IoT Hub connection string (instead of the Event Hub-compatible settings) with the Microsoft Azure Service Bus .NET client library to receive telemetry or operations monitoring events, then be sure to use WindowsAzure.ServiceBus NuGet package version 4.1.2 or higher.

Azure Event Hubs

For details on this service and how to use it, see Azure Event Hubs documentation.

Variations

The following URL for Azure Event Hubs is different in Azure Government:

AZURE PUBLIC	AZURE GOVERNMENT
*.servicebus.windows.net	*.servicebus.usgovcloudapi.net

Azure Notification Hubs

Azure Notification Hubs is generally available in Azure Government.

For details on this service and how to use it, see Azure Notification Hubs documentation.

Variations

The URLs for accessing and managing Azure Notification Hub in Azure Government are different:

AZURE PUBLIC	AZURE GOVERNMENT
*.servicebus.windows.net	*.servicebus.usgovcloudapi.net

Next steps

For supplemental information and updates, subscribe to the Microsoft Azure Government Blog.

Azure Government Integration Services

6/14/2018 • 2 minutes to read • Edit Online

This article outlines the integration services variations and considerations for the Azure Government environment.

Logic Apps

Logic Apps is generally available in Azure Government. For more information, see Logic Apps public documentation.

Variations

- The Azure-based Connectors are scoped to connect to resources in Azure Government. If the Azure service isn't yet available in Azure Government, the connector for that service isn't available, for example:
  - Data Lake Store
  - Data Factory
  - Event Grid
  - Application Insights
  - Content Moderator
- For other missing connectors, request them via the Azure Government feedback forum and the Logic Apps feedback forum. If you need to use any missing connectors, you can call a logic app hosted in Azure Commercial that uses them.
- The creation experience for custom connectors via the portal isn't yet available. If you need to use the portal experience to create a custom connector, you can leverage the portal experience in Azure Commercial. Create the resource in Azure Commercial, download it as an Azure Resource Manager deployment template, and deploy it to Azure Government. You can download a custom connector by selecting the Download button on the Logic Apps Custom Connector overview blade. To deploy resources in a Resource Manager deployment template from the Azure portal, see the resource group deployment documentation.

Next steps

- Subscribe to the Azure Government blog
- Get help on Stack Overflow by using the azure-gov
- Give feedback or request new features via the Azure Government feedback forum

Azure Government Security + Identity

7/20/2018 • 5 minutes to read • Edit Online

This article outlines the security and identity services variations and considerations for the Azure Government environment.

Azure Security Center

Azure Security Center is available for public preview in Azure Government.

For details on this service and how to use it, see the Azure Security Center public documentation.

Variations

The following variations and limitations are present in the Azure Security Center offering in Azure Government:

- **Windows Defender Advanced Threat Protection (WDATP) alerts**
  - WDATP installation on Windows VMs via Security Center and the associated alerts are not available in Azure Government.
- **Security incidents**
  - The aggregation of alerts for a resource, known as a security incident, is not available in Azure Government.
- **Custom alerts**
  - The ability to create custom alerts from raw data is not available in Azure Government.
- **Vulnerability assessments**
  - The Qualys Vulnerability Assessment agent is not available in Azure Government.
- **Email notifications for high severity alerts and JIT access**
  - Alerts and just-in-time access will function normally. However, email notifications are not available in Azure Government.
- **Adaptive application controls**
  - Application whitelisting is not available in Azure Government. Other cloud defense capabilities such as just-time-access (JIT) are available. Standard Azure RBAC roles will function normally.
- **Specific detections**
  - Detections based on VM logs, Azure core router network logs, threat intelligence reports, and detections for app services are not available in Azure Government.
- **Azure activity logs**
  - Auditing insights from Azure activity logs are not available in Azure Government.
- **Baseline content server details**
  - Recommendations and details, such as the potential impact and countermeasures for baseline configuration vulnerabilities, are not available in Azure Government.
- **Security playbooks**
  - Playbooks for automated orchestration and response are not available in Azure Government.
- **Investigation**
  - The investigation feature linking security alerts, users, computers, and incidents is not available in Azure Government.
- **Threat intelligence enrichment**
  - Geo-enrichment and the threat intelligence option are not available in Azure Government.
- **Management Groups**

- Azure management groups are not available in Azure Government and cannot be utilized by Azure Security Center in Azure Government.

Azure Security Center FAQs

For Azure Security Center FAQs, see Azure Security Center frequently asked questions public documentation. Additional FAQs for Azure Security Center in Azure Government are listed below.

What will customers be charged for Azure Security Center in Azure Government?
The Standard tier of Azure Security Center is free for the first 60 days. Should you choose to continue to use public preview or generally available Standard features beyond 60 days, we automatically start to charge for the service.

What features are available for Azure Security Center government customers?
A detailed list of feature variations in the Azure Security Center government offering can found in the variations section of this article. All other Azure Security Center capabilities can be referenced in the Azure Security Center public documentation.

What is the compliance commitment for Azure Security Center in Azure Government?
Azure Security Center engineering has committed to the FedRAMP-High compliance standard with planned audit participation no later than February 2019. We will provide additional updates on this process and certification as this progresses.

Is Azure Security Center available for DoD customers?
Azure Security Center is deployed on Azure Government regions but not DoD regions. Azure resources created in DoD regions can still utilize Security Center capabilities. However, using it will result in Security Center collected data being moved out from DoD regions and stored in Azure Government regions. By default, all Security Center features which collect and store data are disabled for resources hosted in DoD regions. The type of data collected and stored varies depending on the selected feature. Customers who want to enable Azure Security Center features for DoD resources are recommended to consider data residency before doing so.

Key Vault

Key Vault is generally available in Azure Government.

For details on this service and how to use it, see the Azure Key Vault public documentation.

Variations

The URLs for accessing Key Vault in Azure Government are different:

SERVICE TYPE	AZURE PUBLIC	AZURE GOVERNMENT
Key Vault	*.vault.azure.net	*.vault.usgovcloudapi.net
Service Principal ID	cfa8b339-82a2-471a-a3c9-0fc0be7a4093	7e7c393b-45d0-48b1-a35e-2905ddf8183c
Service Principal Name	Azure Key Vault	Azure Key Vault

Data considerations

The following information identifies the Azure Government boundary for Azure Key Vault:

REGULATED/CONTROLLED DATA PERMITTED	REGULATED/CONTROLLED DATA NOT PERMITTED

REGULATED/CONTROLLED DATA PERMITTED	REGULATED/CONTROLLED DATA NOT PERMITTED
All data encrypted with an Azure Key Vault key may contain Regulated/controlled data.	Azure Key Vault metadata is not permitted to contain export-controlled data. This metadata includes all configuration data entered when creating and maintaining your Key Vault. Do not enter Regulated/controlled data into the following fields: **Resource group names, Key Vault names, Subscription name**

Azure Active Directory

Azure Active Directory is generally available in Azure Government.

For details on this service and how to use it, see the Azure Active Directory Documentation.

Variations

The URLs for accessing Azure Active Directory in Azure Government are different:

SERVICE TYPE	AZURE PUBLIC	AZURE GOVERNMENT
Active Directory Endpoint and Authority	https://login.microsoftonline.com	https://login.microsoftonline.us
Active Directory Graph API	https://graph.windows.net/	https://graph.windows.net/

Azure Active Directory Premium P1 and P2

Azure Active Directory Premium is available in Azure Government. For details on this service and how to use it, see the Azure Active Directory Documentation.

For a list of features in Azure Active Directory Premium P1, see Azure Active Directory Features for a list of all capabilities available. This same feature list applies to the US Government cloud instance. All features covered in the above list are available in the US Government cloud instance, with the following known limitations:

Variations

The following Azure Active Directory Premium P1 features are currently not available in Azure Government:

- B2B Collaboration (vote for this feature)
- Group-Based Licensing
- Azure Active Directory Domain Services
- Intune enabled Conditional Access scenarios
- Cloud App Security

The following features have known limitations in Azure Government:

- Limitations with the Azure Active Directory App Gallery:
  - Pre-integrated SAML and password SSO applications from the Azure AD Application Gallery are not yet available. Instead, use a custom application to support federated single sign-on with SAML or password SSO.
  - Rich provisioning connectors for featured apps are not yet available. Instead, use SCIM for automated provisioning.
- Limitations with Multi-factor Authentication:
  - Oath tokens, SMS, and Voice verification can be used as factors, though SMS and Voice traverse outside

the Azure Government Cloud.
    - Trusted IPs are not supported in Azure Government. Instead, use Conditional Access policies with named locations to establish when Multi-Factor Authentication should and should not be required based off the user's current IP address.
- Limitations with Azure AD Join:
    - Joining cloud and hybrid devices to Azure AD is not yet available
    - Features related to Azure AD joined devices not yet available are Desktop SSO, Windows Hello and Self-service BitLocker recovery
    - MDM auto-enrollment for Windows 10 devices in Azure AD is not yet available
    - Enterprise State Roaming for Windows 10 devices is not available

Azure Multi-Factor Authentication

For details on this service and how to use it, see the Azure Multi-Factor Authentication Documentation.

Next Steps

For supplemental information and updates, subscribe to the Microsoft Azure Government Blog.

Azure Government Backup

10/23/2017 • 2 minutes to read • Edit Online

This article provides an overview of the Azure Backup service and lists the Backup features available in Azure Government. Azure Backup is the Azure-based service you can use to back up (or protect) your data to the Microsoft cloud. Protecting your data in Azure not only means backing it up to the cloud, but restoring the data either to the cloud, or to an on-premises installation. Azure Backup provides these key benefits:

- Automatic storage management
- Unlimited scaling
- Multiple storage options
- Unlimited data transfer
- Data encryption
- Application-consistent backup
- Long-term retention

If you're new to Azure Backup and would like an overview of the available features, read the article, What is Azure Backup.

The Azure Backup service had two types of vaults - the Backup vault and the Recovery Services vault. The Backup vault came first. Then the Recovery Services vault came along to support the expanded Resource Manager deployments. Microsoft recommends using Resource Manager deployments unless you specifically require a Classic deployment. By the end of 2017, all Backup vaults were converted to Recovery Services vaults.

> **NOTE**
> Backup vaults could not protect Resource Manager-deployed solutions. However, Recovery Services vaults can protect classically-deployed servers and VMs.

Azure Backup components available in Azure Government Backup

You can use Azure Backup to protect: files, folders, volumes, virtual machines, applications, and workloads. Depending on what you want to protect, and where that data exists, you use a different Azure Backup component. The following sections have links to articles in the Azure Backup public documentation for each component.

Using Windows Server and Windows computers in Azure portal

- Back up Windows Server and Windows client computers
- Restore Windows Server and Windows client computers
- Manage Windows Server and Windows client computer backups
- Using PowerShell to back up Windows Server

Using Virtual Machines in Azure portal

- Prepare your virtual machine environment
- Back up virtual machines
- Restore virtual machines
- Manage virtual machines
- Using PowerShell to back up virtual machines

Using System Center Data Protection Manager in Azure portal

- Back up System Center Data Protection Manager

Using Azure Backup Server in Azure portal

Azure Backup Server is an Azure Backup component that functions similarly to System Center Data Protection Manager (DPM) with one exception - Azure Backup Server cannot save data to tape. Azure Backup Server can protect application workloads such as: Hyper-V VMs, Microsoft SQL Server, SharePoint Server, Microsoft Exchange, and Windows clients to the cloud from a single console. Azure Backup Server does not require a System Center license.

- Azure Backup Server

Upgrade a Backup vault to a Recovery Services vault

- Upgrade now

Next steps

If you aren't sure where to begin, start with the article, Back up Windows Server and Windows client computers. This tutorial leads you through the steps for setting up a backup project on a Windows Server or computer.

If you already know that you could use Azure Backup, but want to know the costs, see the Backup Pricing page. There is a list of Frequently Asked Questions that may provide useful information. Also note there are multiple Azure Government regions in the **Region** dropdown menu.

Azure Government Monitoring + Management

7/12/2018 • 5 minutes to read • Edit Online

This article outlines the monitoring and management services variations and considerations for the Azure Government environment.

Advisor

Advisor is in public preview in Azure Government.

For more information, see Advisor public documentation.

Variations

The following Advisor recommendations are not currently available in Azure Government:

- Security
  - Security recommendations from Security Center
- Cost
  - Optimize virtual machine spend by resizing or shutting down underutilized instances
  - Eliminate unprovisioned ExpressRoute circuits
- Performance
  - Improve App Service performance and reliability
  - Improve Redis Cache performance and reliability

Automation

Automation is generally available in Azure Government.

For more information, see Automation public documentation.

Backup

Backup is generally available in Azure Government.

For more information, see Azure Government Backup.

Policy

Policy is generally available in Azure Government.

For more information, see Azure Policy.

Site Recovery

Azure Site Recovery is generally available in Azure Government.

For more information, see Site Recovery commercial documentation.

Variations

The following Site Recovery features are not currently available in Azure Government:

- Email notification

SITE RECOVERY	CLASSIC	RESOURCE MANAGER
VMWare/Physical	GA	GA
Hyper-V	GA	GA
Site to Site	GA	GA

The following URLs for Site Recovery are different in Azure Government:

AZURE PUBLIC	AZURE GOVERNMENT	NOTES
*.hypervrecoverymanager.windowsazure.com	*.hypervrecoverymanager.windowsazure.us	Access to the Site Recovery Service
*.backup.windowsazure.com	*.backup.windowsazure.us	Access to Protection Service
*.blob.core.windows.net	*.blob.core.usgovcloudapi.net	For storing the VM Snapshots
http://cdn.mysql.com/archives/mysql-5.5/mysql-5.5.37-win32.msi	http://cdn.mysql.com/archives/mysql-5.5/mysql-5.5.37-win32.msi	To download MySQL

Monitor

Azure Monitor is generally available in Azure Government.

For more information, see Monitor commercial documentation.

Variations

The following sections detail differences and workarounds for features of Azure Monitor in Azure Government:

Action Groups

Action Groups are generally available in Azure Government with no differences from commercial Azure.

Activity Log Alerts

Activity Log Alerts are generally available in Azure Government with no differences from commercial Azure.

Alerts Experience

The unified alerts UI experience is not available in Azure Government.

Autoscale

Autoscale is generally available in Azure Government.

If you are using PowerShell/ARM/REST calls to specify settings, set the "Location" of the Autoscale to "USGov Virginia" or "USGov Iowa". The resource targeted by Autoscale can exist in any region. An example of the setting is below:

```
$rule1 = New-AzureRmAutoscaleRule -MetricName "Requests" -MetricResourceId
"/subscriptions/S1/resourceGroups/RG1/providers/Microsoft.Web/sites/WebSite1" -Operator GreaterThan -
MetricStatistic Average -Threshold 10 -TimeGrain 00:01:00 -ScaleActionCooldown 00:05:00 -ScaleActionDirection
Increase -ScaleActionScaleType ChangeCount -ScaleActionValue "1"
$rule2 = New-AzureRmAutoscaleRule -MetricName "Requests" -MetricResourceId
"/subscriptions/S1/resourceGroups/RG1/providers/Microsoft.Web/sites/WebSite1" -Operator GreaterThan -
MetricStatistic Average -Threshold 10 -TimeGrain 00:01:00 -ScaleActionCooldown 00:10:00 -ScaleActionDirection
Increase -ScaleActionScaleType ChangeCount -ScaleActionValue "2"
$profile1 = New-AzureRmAutoscaleProfile -DefaultCapacity 2 -MaximumCapacity 10 -MinimumCapacity 2 -Rules
$rule1, $rule2 -Name "MyProfile"
$webhook_scale = New-AzureRmAutoscaleWebhook -ServiceUri https://example.com?mytoken=mytokenvalue
$notification1= New-AzureRmAutoscaleNotification -CustomEmails myname@company.com -
SendEmailToSubscriptionAdministrator -SendEmailToSubscriptionCoAdministrators -Webhooks $webhook_scale
Add-AzureRmAutoscaleSetting -Location "USGov Virginia" -Name "MyScaleVMSSSetting" -ResourceGroup sdubeys-usgv -
TargetResourceId /subscriptions/s1/resourceGroups/rg1/providers/Microsoft.Web/serverFarms/ServerFarm1 -
AutoscaleProfiles $profile1 -Notifications $notification1
```

If you are interested in implementing autoscale on your resources, use PowerShell/ARM/Rest calls to specify the settings.

For more information on using PowerShell, see public documentation.

Diagnostic Logs

Diagnostic Logs are generally available in Azure Government with no differences from commercial Azure.

Metrics

Metrics are generally available in Azure Government. However, multi-dimensional metrics are supported only via the REST API. The ability to show multi-dimensional metrics is in preview in the Azure Government portal.

Metric Alerts

The first generation of metrics alerts is generally available in both Azure Government and commercial Azure. The first generation is called *Alerts (Classic)*. A second generation of alerts is available only in commercial Azure.

When using PowerShell/ARM/Rest calls to create Metric Alerts, you will need to set the "Location" of the metric alert to "USGov Virginia" or "USGov Iowa". An example of the setting is below:

```
$actionEmail = New-AzureRmAlertRuleEmail -CustomEmail myname@company.com
$actionWebhook = New-AzureRmAlertRuleWebhook -ServiceUri https://example.com?token=mytoken
Add-AzureRmMetricAlertRule -Name vmcpu_gt_1 -Location "USGov Virginia" -ResourceGroup myrg1 -TargetResourceId
/subscriptions/s1/resourceGroups/myrg1/providers/Microsoft.ClassicCompute/virtualMachines/my_vm1 -MetricName
"Percentage CPU" -Operator GreaterThan -Threshold 1 -WindowSize 00:05:00 -TimeAggregationOperator Average -
Actions $actionEmail, $actionWebhook -Description "alert on CPU > 1%"
```

For more information on using PowerShell, see public documentation.

Log Analytics

Log Analytics is generally available in Azure Government.

Variations

- Solutions that are available in Azure Government include:
  - Network Performance Monitor (NPM) - NPM is a cloud-based network monitoring solution for public and hybrid cloud environments. Organizations use NPM to monitor network availability across on-premises and cloud environments. Endpoint Monitor - a subcapability of NPM, monitors network connectivity to applications.

The following Log Analytics features and solutions are not currently available in Azure Government.

- Solutions that are in preview in Microsoft Azure, including:

    - Service Map
    - Windows 10 Upgrade Analytics solution
    - Application Insights solution
    - Azure Networking Security Group Analytics solution
    - Azure Automation Analytics solution
    - Key Vault Analytics solution
- Solutions and features that require updates to on-premises software, including:
    - Surface Hub solution
- Features that are in preview in public Azure, including:
    - Export of data to Power BI
- Azure metrics and Azure diagnostics
- Operations Management Suite mobile application

The URLs for Log Analytics are different in Azure Government:

AZURE PUBLIC	AZURE GOVERNMENT	NOTES
mms.microsoft.com	oms.microsoft.us	Log Analytics portal
workspaceId.ods.opinsights.azure.com	*workspaceId*.ods.opinsights.azure.us	Data collector API
*.ods.opinsights.azure.com	*.ods.opinsights.azure.us	Agent communication - configuring firewall settings
*.oms.opinsights.azure.com	*.oms.opinsights.azure.us	Agent communication - configuring firewall settings
*.blob.core.windows.net	*.blob.core.usgovcloudapi.net	Agent communication - configuring firewall settings
portal.loganalytics.io	portal.loganalytics.us	Advanced Analytics Portal - configuring firewall settings
api.loganalytics.io	api.loganalytics.us	Advanced Analytics Portal - configuring firewall settings
docs.loganalytics.io	docs.loganalytics.us	Advanced Analytics Portal - configuring firewall settings
*.azure-automation.net	*.azure-automation.us	Azure Automation - configuring firewall settings

The following Log Analytics features behave differently in Azure Government:

- To connect your System Center Operations Manager management server to Log Analytics, you need to download and import updated management packs.
    - System Center Operations Manager 2016
        1. Install Update Rollup 2 for System Center Operations Manager 2016.
        2. Import the management packs included as part of Update Rollup 2 into Operations Manager. For information about how to import a management pack from a disk, see How to Import an Operations Manager Management Pack.
        3. To connect Operations Manager to Log Analytics, follow the steps in Connect Operations

Manager to Log Analytics.

- System Center Operations Manager 2012 R2 UR3 (or later) / Operations Manager 2012 SP1 UR7 (or later)
  1. Download and save the updated management packs.
  2. Unzip the file that you downloaded.
  3. Import the management packs into Operations Manager. For information about how to import a management pack from a disk, see How to Import an Operations Manager Management Pack.
  4. To connect Operations Manager to Log Analytics, follow the steps in Connect Operations Manager to Log Analytics.

- To use computer groups from System Center Configuration Manager 2016, you need to be using Technical Preview 1701 or later.

Frequently asked questions

- Can I migrate data from Log Analytics in Microsoft Azure to Azure Government?
  - No. It is not possible to move data or your workspace from Microsoft Azure to Azure Government.
- Can I switch between Microsoft Azure and Azure Government workspaces from the Operations Management Suite Log Analytics portal?
  - No. The portals for Microsoft Azure and Azure Government are separate and do not share information.

For more information, see Log Analytics public documentation.

Scheduler

For information on this service and how to use it, see Azure Scheduler Documentation.

Azure portal

The Azure Government portal can be accessed here.

Azure Resource Manager

For information on this service and how to use it, see Azure Resource Manager Documentation.

Next steps

- Subscribe to the Azure Government blog
- Get help on Stack Overflow by using the azure-gov
- Give feedback or request new features via the Azure Government feedback forum

Azure Government Developer Tools

5/7/2018 • 2 minutes to read • Edit Online

This article outlines the developer tools services variations and considerations for the Azure Government environment.

DevTest Labs

DevTest Labs is generally available in Azure Government. For more information, see DevTest Labs public documentation.

Variations

The following DevTest Labs features are not currently available in Azure Government:

- Azure Resource Manager templates to create multi-VM environments and PaaS resources within a Lab.
- Connect to an external GitHub repository to add Azure Resource Manager Templates, however adding artifact repositories to leverage custom artifacts is available.
- Auto shutdown feature for Azure compute VMs, however, setting auto shutdown for Labs and Lab Virtual Machines is available.

Next steps

- Subscribe to the Azure Government blog
- Get help on Stack Overflow by using the azure-gov
- Give feedback or request new features via the Azure Government feedback forum

Azure Government CSP application process

5/11/2018 • 3 minutes to read • Edit Online

Azure Government is available for purchase via different channels, one of them is the Cloud Solution Provider channel or CSP. The following resources provide an overview of what you need to become a CSP for US Government and be able to resell Azure Government. An overview of the CSP program can be found in the Cloud Solution Provider Program page.

Basics

Before being able to apply for CSP or any other programs that run under the Microsoft Partner Network, you need to obtain a Microsoft Partner Network ID (MPN ID). To get an overview, visit the Microsoft Partner Network page. To become a Partner and obtain and ID, the Enrollment and Membership page has all the details.

Becoming a government CSP

Now that you have obtained an overview on what the CSP program is and met the basic requirement of becoming a Microsoft Partner, focus on becoming a Government CSP. Designed for the US government from the ground up, Microsoft Cloud for Government enables public sector customers in the United States—from large federal agencies to small town governments—to select from a range of cloud computing services. To best address its customers' specific needs, Microsoft has made significant investment in datacenters and is dedicated to meeting compliance with US federal and state policies, mandates, and requirements.

To initiate your application, visit the CSP Program for Microsoft Cloud for US Government page. Resources and links to all relevant pages are contained in this page.

Obtaining your government tenant

The process kicks-off with the request for an Azure Government Tenant. The form to request this tenant and begin the validation can be found at the Microsoft Government Validation CSP Request page. Once complete, you receive a Government Azure Tenant. The validation includes the following items:

- Be a Microsoft Partner (have an MPN ID)
- Verification of legitimacy of the Company, Systems Integrator, Distributor, or Independent Software Vendor (ISV) applying for the tenant
- Verification of business engagements with government customers (for example, proof of services rendered to government agencies, statements of works, evidence of being part of GSA Schedule)
- If you already have an Azure Government tenant, you can use your existing credentials to complete the CSP Application.
- Ensure that emails coming from US Government Cloud Eligibility are reaching your inbox.
- Check SPAM / JUNK as credentials and asks for further info come from this alias.

Applying for government CSP

Once the preceding process is complete, credentials are provided via email to the authorized users. Use these credentials and navigate to the CSP Partner Center for Gov application page and "Sign In" to apply to join for the CSP Government seller program. It takes 5-6 days to process the application and once approved you receive an email to log on to Partner Center to accept the Terms and Conditions.

> **NOTE**
> Terms and Conditions are not negotiated for the Cloud Solution Provider Program. If you wish to discuss customer terms that you have in place for your Commercial agreement, work with your Microsoft Account Representative to achieve so.

The application process includes:

- Credit check
- Estimation of potential revenue
- Company validation via Dun and Bradstreet
- Email Verification
- Accepting Terms and Conditions

After the validation has been completed and terms have been signed, you are Ready to Transact. At this point, you can create customers, spin up resources, and get billed as per regular CSP Terms. To learn more about CSP Billing, you can visit this page.

Additional resources

- Once you are a CSP and have an MPN ID, you can review all incentives by signing up and reviewing the Partner Incentives page.
- All agreements for end customers and partners in the CSP program are located on the CSP Resources page. The customer agreement to be flown down is the MCA or Microsoft Cloud Agreement.
- A list of the Azure Services can be found on the Azure Services Availability on CSP page.
- To learn about the most frequently asked questions related to the US Government CSP Program, visit the FAQ page.
- If you are still unclear about CSP or are looking to apply for the commercial side of the program, review the CSP Programs for Commercial, once you have elected a program that suits your business needs, apply for the one that meets your profile.

Next steps

Once you have onboarded and are ready to create your first customer, make sure to review the Azure Government CSP Quickstart to view a step by step screencast of the process. If you have any additional questions, contact the Azure Government CSP Program.

ITAR Overview for Azure Government

4/25/2018 • 2 minutes to read • Edit Online

Overview

This information is intended for Azure Government customers with obligations under the International Traffic in Arms Regulations (ITAR) who intend to use Azure Government services to store, process, or transmit regulated information. The information provided describes the capabilities of Microsoft Azure Government services, and provides general guidance applicable to ITAR customers. Before including ITAR-controlled data in your Azure Government subscription, you should familiarize yourself with the Azure Government capabilities and consult your account team if you have any questions.

You should refer to the Microsoft Azure Trust Center Compliance Page for current information on the Azure Government services covered under ITAR. Additional Microsoft services might also be available, but are not within the scope of the Azure Government covered services. Azure Government services might also permit you to use a variety of additional resources, applications, or services that are provided by third parties—or by Microsoft under separate terms of use and privacy policies. You are responsible for reviewing the terms of all such "add-on" offerings, such as Marketplace offerings, to ensure that they meet your needs regarding ITAR compliance.

Azure Government is available to entities that handle data that is subject to certain government regulations and requirements, such as ITAR, where use of Azure Government is required to comply with regulations. Azure Government customers are subject to validation of eligibility. Validation of eligibility by Microsoft will include confirmation that you are a manufacturer, exporter, or broker of defense articles under the ITAR—as shown by your registration with the US Department of State—or through a sponsorship agreement with a government entity that has specific requirements for the handling of data.

Entities with questions about eligibility for Azure Government should consult their account team.

Next steps

Microsoft Trust Center - ITAR web page

Microsoft Azure Government Blog.

Justice and Public Safety (JPS) in Azure Government

6/27/2017 • 2 minutes to read • Edit Online

Overview

Justice and Public Safety (JPS) agencies are under mounting pressure to keep communities safe, reduce crime, and improve responsiveness. From intelligent policing awareness systems, to body camera systems across the country, to day-to-day mobile police collaboration, cloud computing is transforming the way law enforcement agencies approach their work.

When they are properly planned and secured, cloud services can deliver powerful new capabilities for JPS. These capabilities include digital evidence management, data analysis, and real-time decision support--with solutions delivered on the latest mobile devices. However, not all cloud providers are equal. As law enforcement agencies embrace the cloud, they need a cloud service provider they can trust. The core of the law enforcement mission demands partners who are committed to meeting a full range of security, compliance, and operational needs.

From devices to the cloud, Microsoft puts privacy and information security first, while increasing productivity for officers in the field and throughout the department. By combining highly secure mobile devices with "anytime-anywhere" access to the cloud, JPS agencies can contribute to ongoing investigations, analyze data, manage evidence, and help protect citizens from threats.

Other cloud providers treat Criminal Justice Information Systems (CJIS) compliance as a check box, rather than a commitment. At Microsoft, we're committed to providing solutions that meet the applicable CJIS controls, today and in the future. In addition, we extend our commitment to justice and public safety through our Digital Crimes Unit, Cyber Defense Operations Center, and Worldwide Justice and Public Safety organization.

Next steps

- Microsoft Trust Center - Criminal Justice Information Services webpage
- Microsoft Azure Government blog

Department of Defense (DoD) in Azure Government

4/25/2018 • 8 minutes to read • Edit Online

Overview

Azure Government is used by Department of Defense (DoD) entities to deploy a broad range of workloads and solutions, including those workloads covered by The DoD Cloud Computing Security Requirements Guide, Version 1, Release 2 at Impact Level 4 (L4), and Impact Level 5 (L5).

Azure Government is the first and only hyperscale commercial cloud service to be awarded an Information Impact Level 5 DoD Provisional Authorization by the Defense Information Systems Agency. In addition, Azure Government regions dedicated to US Department of Defense customer workloads are now generally available.

One of the key drivers for the DoD in moving to the cloud is to enable organizations to focus on their missions and minimize the distractions of building and managing in-house IT solutions.

Azure Government-based cloud architectures allow DoD personnel to focus on mission objectives, and managing IT commodity services such as SharePoint and other application workloads. This allows for the realignment of critical IT resources to focus on application development, analytics, and cyber security.

The elasticity and flexibility delivered by Azure provides enormous benefits to DoD customers. It is simpler, quicker, and more cost-effective to scale-up a workload in the cloud than it is to go through traditional hardware and services procurement processes when working on-premises, or in DoD data centers. For example, to procure new multi-server hardware, even for a test environment, may take many months, and require the approval of significant capital expenditure. By contrast, using Azure, a test migration for an existing workload can be configured in weeks or even days, and in a cost-effective manner (when the test is over, the environment can be torn down with no ongoing costs).

This flexibility is significant. By moving to Azure, DoD customers do not just save money; the cloud delivers new opportunities. For example, it is easy to spin up a test environment to gain insights into new technologies, you can migrate an application and test it in Azure before committing to a production deployment in the cloud. Mission owners can explore more cost effective options easier, and without risk.

Security is another key area, and although any cloud deployment requires proper planning to ensure secure and reliable service delivery, in reality most properly configured cloud-based workloads (up to and including L4 workloads) in Azure Government will be more secure than many traditional deployments in DoD locations and data centers. This is because defense agencies have the experience and expertise to physically secure all assets; however, the IT surface areas present different challenges. Cyber security is a rapidly changing space, requiring specialist skills and the ability to rapidly develop and deploy counter-measures as required. The Azure platform, both commercial and Government, now supports hundreds of thousands of customers, and this scale enables Microsoft to quickly detect evolving attack vectors, and then direct its resources onto rapid development and implementation of the appropriate defenses.

DoD Region Q&A

What are the Azure Government DoD Regions?

The US DoD East and US DoD Central regions are physically separated regions of Microsoft Azure architected to meet US Department of Defense (DoD) security requirements for cloud computing, specifically for data designated as DoD Impact Level 5 per the DoD Cloud Computing Security Requirements Guide (SRG).

What is the difference between Azure Government and the Azure Government DoD Regions?

Azure Government is a US government community cloud providing services for Federal, State and Local government customers, tribal, entities subject to ITAR, and solution providers performing work on their behalf. All Azure Government regions are architected and operated to meet the security requirements for DoD Impact Level 5 data and FedRAMP High standards.

The Azure Government DoD regions are architected to support the physical separation requirements for Impact Level 5 data by providing dedicated compute and storage infrastructure for the use of DoD customers only.

What is the difference between Impact Level 4 and Impact Level 5 data?

Impact Level 4 data is controlled unclassified information (CUI) that may include data subject to export control, privacy information protected health information and other data requiring explicit CUI designation (e.g. For Official Use Only, Law Enforcement Sensitive, Sensitive Security Information).

Impact Level 5 data includes controlled, unclassified information (CUI) that requires a higher level of protection as deemed necessary by the information owner, public law or government regulation. Impact Level 5 data is inclusive of unclassified National Security Systems. More information on the SRG impact levels, their distinguishing requirements and characteristics is available in section 3 of the DoD Cloud Computing Security Requirements Guide.

What Data is categorized as Impact Level 5?

Level 5 accommodates controlled unclassified information (CUI) that requires a higher level of protection than that afforded by Level 4 as deemed necessary by the information owner, public law, or other government regulations. Level 5 also supports unclassified National Security Systems (NSSs). This level accommodates NSS and CUI information categorizations based on CNSSI-1253 up to moderate confidentiality and moderate integrity (M-M-x).

What is Microsoft doing differently to support Impact Level 5 data?

Impact Level 5 data by definition can only be processed in a dedicated infrastructure that ensures physical separation of DoD customers from non-Federal government tenants. In delivering the US DoD East and US DoD Central regions, Microsoft is providing an exclusive service for DoD customers that meets an even higher bar than DoD's stated requirements and exceeds the level of protection and capability offered by any other hyperscale commercial cloud solution.

Do these regions support classified data requirements?

These Azure Government DoD regions support only unclassified data up to and including Impact Level 5. Impact Level 6 data is defined as classified information up to Secret.

What organizations in the DoD can use the Azure Government DoD Regions?

The US DoD East and US DoD Central regions are built to support the US Department of Defense customer base. This includes:

- The Office of the Secretary of Defense
- The Joint Chiefs of Staff
- The Joint Staff
- The Defense Agencies
- Department of Defense Field Activities
- The Department of the Army
- The Department of the Navy (including the United States Marine Corps)
- The Department of the Air Force
- The United States Coast Guard
- The unified combatant commands
- Other offices, agencies, activities, and commands under the control or supervision of any approved entity named above

Are the DoD regions more secure?

Microsoft operates all of its Azure datacenters and supporting infrastructure to comply with local and international standards for security and compliance – leading all commercial cloud platforms in compliance investment and achievements. These new DoD regions will provide specific assurances and commitments to meet the requirements defined in the DoD SRG for Cloud Computing.

Why are there multiple DoD regions?

By having multiple DoD regions, Microsoft provides customers with the opportunity to architect their solutions for disaster recovery scenarios across regions to ensure business continuity and satisfy requirements for system accreditation. In addition, customers may optimize performance by deploying solutions in the geography within closest proximity to their physical location.

Are these DoD regions connected to the NIPRNet?

The DoD mandates that commercial cloud services used for CUI must be connected to customers through a Cloud Access Point (CAP). Therefore, the Azure DoD regions are connected to the NIPRNet through redundant connections to multiple geographically distributed CAPs. A DoD CAP is a system of network boundary protection and monitoring devices that offer protection to DoD information system network and services.

What Does General Availability Mean?

General Availability means that the DoD regions in Azure Government may be used to support production workloads and that financially backed SLAs for all services deployed in the regions and also generally available will be supported.

How does a DoD customer acquire Azure Government DoD services?

Azure Government DoD services may be purchased by qualified entities through the same reseller channels as Azure Government. In keeping with Microsoft's commitment to make cloud services acquisition planning and cost estimation simple, pricing for Azure Government DoD regions will be included in the Azure Pricing calculator at the time of general availability. Azure Government DoD services can quickly scale up or down to match demand, so you only pay for what you use. No contractual modifications will be required for Enterprise Agreement customers already using Azure Government.

How are the DoD regions priced?

The DoD regions utilize region based pricing. This means that service costs for validated DoD customers will be based on the Azure Government region in which you run your workloads. For more specific pricing information, please consultant your Microsoft Account Executive. Pricing for the DoD regions will be provided through the Azure.com calculator at a future date.

How does a DoD organization get validated for the Azure Government DoD regions?

In order to gain access to the Azure DoD regions, customers must complete a pre-qualification process for verifying their organization and intended use of the Azure DoD environment. After successful completion of the pre-qualification process, Microsoft will provide the organizational applicant with further instructions for creating a subscription, accessing the environment and providing role-based access control to other members of the organization.

Can independent software vendors and solution providers building on Azure deploy solutions in the Azure Government DoD regions?

Solution providers with cloud service offerings built on Azure may operate DoD-only single tenant and multi-tenant solutions in the Azure Government DoD regions. These providers must first demonstrate eligibility by providing documented evidence of a contract with an approved DoD entity or have a sponsor letter from an approved DoD entity. Providers offering services in the Azure Government DoD regions must include computer network defense, incident reporting and screened personnel for operating solutions handling Impact Level 5 information in their offering. Additional guidance for solution providers may be found in the DoD Cloud Computing Security Requirements Guide.

Will Office 365 or Microsoft Dynamics 365 be a part of this offering?

Microsoft is providing Office 365 services for the DoD at Impact Level 5 in conjunction with this offering. Dynamics 365 is planning to offer Impact Level 5 services from the Azure DoD regions at a future date.

How do I connect to the DoD Regions once I have a subscription?

The DoD regions for Azure Government are available through the Azure Government management portal. DoD customers approved for use will see the regions listed as available options when deploying available services. For general guidance on managing your Azure Government subscriptions please consult our documentation.

What services are part of your Impact Level 5 accreditation scope?

Azure is an evergreen service where new services and capabilities are being added every week, the number of services in scope is regularly expanding. For the most up-to-date information, please visit our Microsoft Trust Center.

Next steps:

Microsoft Trust Center - DoD web page

The DoD Cloud Computing Security Requirements Guide, Version 1, Release 2

Azure Government Reseller Channels

Microsoft Azure Government Blog.