Global Knowledge®

Expert Reference Series of White Papers

Backing Up Servers to the Microsoft Cloud

Backing Up Servers to the Microsoft Cloud

Glenn Weadock, Global Knowledge Instructor, MCITP, MCSE, MCT, A+

Overview of Azure Backup

By far the easiest way to get an initial experience with Azure, "the Microsoft cloud," is by using it to back up one or more Windows servers. The procedure, while more involved than you might think, goes fairly quickly when you have a handy guide nearby (such as this white paper, I hope!). I tried out the service for my own company; the examples and screenshots here are from "real life."

Using Azure for backup isn't expensive (costs have come down since its introduction) and you get a free month, within usage limits, to experiment with it. If you like the experience, you can expand your Azure usage to include Web hosting, Active Directory, cloud-based virtual machines, and so on. And even if you don't explore Microsoft Azure further (note the recent rebranding from "Windows Azure"), you may find that the online backup piece fills a need in your disaster recovery strategy.

(Incidentally, aspirants to the Server 2012 MCSA certification will need to know about Azure Backup to pass the 70-412 exam.)

Clients and Workloads

In early 2015, Microsoft beefed up Azure Backup. Its capabilities at this writing include backing up Windows Server 2008 R2 SP1 and newer, System Center 2012 Data Protection Manager workloads (including Hyper-V, SQL Server, Sharepoint, and Exchange as long as you have DPM Update Rollup 5), 64-bit versions of Windows 8.x and Windows 7 SP1, and Windows Server Essentials 2012 and newer (which requires a different agent).

Note that without Data Protection Manager, Azure Backup is not supported for any workload other than file-and-folder backups. You cannot use Azure Backup to back up the system state, or to create a Bare-Metal-Restore (BMR) backup. You cannot back up network shares. Volumes to be backed up must be NTFS, and if you're using BitLocker, you must disable it before the backup.

The new (2015) size limit is 1.65 TB per backup. Retention policies now permit retaining backups for up to 99 years.

Data Handling

For the initial "seeding" of the cloud backup, you may not wish to back up over your network, in which case Microsoft offers the "offline backup" option that integrates with the Azure import/export service. You basically ship a disk to your friendly neighborhood Azure datacenter to create the first copy of your backup.

A key to overcoming institutional concerns about security in the cloud is *encryption*. Azure Backup requires a passphrase for encrypting data to be backed up. The agent performs encryption prior to transmission to Azure, and decryption after performing a restore operation back to the local system. Microsoft advises that it does not keep a copy of the passphrase. For additional security, servers must be registered to an Azure "vault" using verified credentials. When restoring data to the same server from which it was backed up, you don't have to supply the passphrase, but when restoring to a different server, you do. Microsoft can generate a passphrase for you, or you can create your own; in either case, you can change it in the console's Properties page later.

Azure Backup uses *compression* to reduce transmission time and storage requirements. As you might expect, the effectiveness of its compression depends on the compressibility of the content; in my test backup of 76 GB worth of ISO images, which are not very compressible, the compression ratio was very close to 1:1.

Azure Backup also uses block-level incremental backup methods so that only modified blocks are backed up in subsequent backups of the same set of files and folders. I wonder about its efficiency though: when I added a single 13KB file to my backup folder, Azure Backup's next backup was over 21MB, which seems like a lot of overhead. At each backup's conclusion, a verification step is performed, which was admirably quick in my testing.

For *redundancy*, Microsoft stores six copies of your backups, three in one datacenter and three in a geographically separate datacenter. They're all in the same country, but if an entire country goes offline, we probably have other problems.

Setting up Azure Backup

The main steps in configuring Azure Backup are as follows (optionally you would also configure integration with System Center Data Protection Manager, but we don't have space to go into that here):

- Create an Azure account, if you don't already have one.
- Create a "backup vault" online.
- Get a "vault credential."
- Download the backup agent.
- Install same, following the wizard's configuration prompts.
- Register your server (the agent installer daisy-chains into this step).
- Fire up your console of choice (there are two) and finish configuration by setting schedule and throttling options.
- Start, monitor, and manage your backups, including performing restores as necessary.

Creating an Account and a Backup Vault

If your organization doesn't already have an Azure account, creating one is your first step. Navigate to https://account.windowsazure.com/ and follow the links. You'll need a Microsoft account, a phone number, and a credit card, even if you're just signing up for the free trial; Microsoft states that it doesn't charge the card unless you subsequently convert the free trial to a pay-as-you-go subscription. (Elsewhere, Microsoft states that it will charge the card $1 which it then refunds.) You should also have your company legal eagles go over the privacy statement and online subscription agreement to make sure it's all good with them.

A note on the free trial: at this writing, you get a free month and a $200 credit (see later in this paper for notes on pricing). I experienced some glitches setting up my Azure account initially; it wasn't happy with Firefox, or IE 11 for that matter, but I got it working on a Mac (!). Amusingly, the day I created the free trial, the management portal sported a message that my free trial was "ending soon." The green "CREDIT STATUS" button at the top of the management portal lets you see how that $200 is holding up during the trial period. As a point of reference, I performed a 76 GB backup, a 125 MB restore, and used $1 of my $200 credit. Who says a dollar doesn't go as far as it used to?

Once your Azure account is set up, you need to create a "vault." Each vault can protect up to fifty machines, and you're allowed up to twenty-five vaults. Sign in to the Azure portal (https://manage.windowsazure.com) and, via the (ahem) non-alphabetized navigation menu on the left, choose **Data Services > Recovery Services > Backup Vault > Quick Create**. (Do not choose **Data Services > Storage**, as might seem logical.) See Figure 1 below. Name your vault, pick a geographical region (e.g., "West US"), and click Create Vault. The creation process took about five minutes for me.

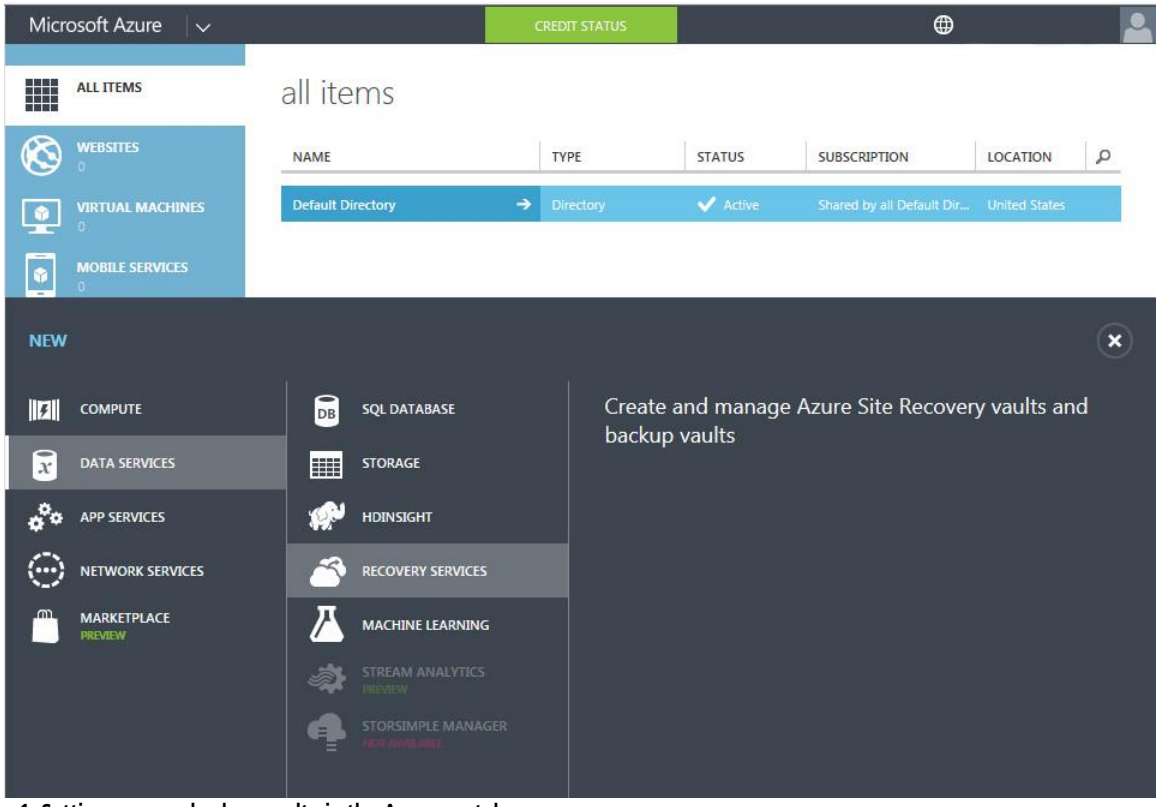

Figure 1: Setting up your backup vault via the Azure portal

Once the vault is built, you need to get a *vault credential*. In a change from 2014 and earlier, you no longer need to obtain or create a certificate before registering your server with Microsoft; this process has become easier. In the portal's "Recovery Services" area, left-click the vault to make the "Quick Start" page appear (there is no on-screen indication that this will happen). Click "Download vault credentials" and specify a location.

Alternatively, if you already have an X.509 v3 certificate that you'd like to use, you can upload it by clicking "Manage Certificate" on the command bar at the bottom of the screen.

Downloading and Installing the Agent

Via the "Quick Start" page, you can download the Azure Backup Agent installer (it's an EXE file). Make sure you pick the right one: there's one for Windows Server Essentials, and one that covers all the other supported environments.

- Run the installer on the server you plan to back up, after prepping it first with .Net Framework 4.5 if needed.
- You'll be prompted to specify a cache location (the default is *%ProgramFiles%|Microsoft Azure Recovery Services Agent|Scratch*), which should be roughly 5 percent as large as what you want to back up.
- You'll also have the opportunity to specify proxy server settings if appropriate.

Registering and Configuring the Server

The backup agent installer daisy-chains right into the server registration process, which goes as follows:

- Browse to the vault credentials you created earlier; they will be verified.
- Provide (or let Microsoft generate for you) a passphrase of at least sixteen characters that Azure will use to encrypt and decrypt backups and restores.
- Specify a location to save the passphrase (Microsoft urges that you choose a location off the volume you intend to back up, as Microsoft does not maintain a copy).

When you're finished, if you've done everything correctly, you may have my experience and see a failure message like the one in Figure 2 below. I gather from comments on the web that this is not uncommon, so rather than spending a lot of time trying to troubleshoot the error, maybe just take Microsoft's ungrammatical advice to "retry the operation after sometime," and it will likely succeed, as mine eventually did.

Figure 2: If at first you don't succeed...

After registration succeeds, you can now use the existing Windows Server Backup console (WBADMIN.MSC), which after the addition of the backup agent has a new node (see Figure 3 below) named simply "Backup" (they seriously couldn't have called it "Azure Backup" to help us out?), or you can use the dedicated Azure Backup console, which shows the same information but without the local backup capabilities.

The dedicated console is WABADMIN.MSC and you will have to find it squirreled away in *Program Files > Microsoft Azure Recovery Services Agent > bin*, because apparently putting a shortcut in with all the other administrative tools was too much trouble). Note that both of these consoles are server-centric and not useful for managing multiple servers.

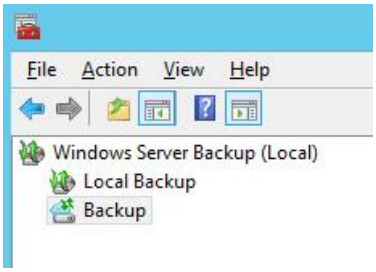

Figure 3: Azure Backup appearing as "Backup" in the Server Backup console

Your next action will be to click "Schedule Backup" in the Action pane (or menu) and launch the relevant wizard. You must do this even if you only intend to perform ad hoc backups. You'll make choices regarding the following:

- What files and/or folders to include and/or exclude
- What days and times to back up (three times per day at most)
- What retention policy you wish to specify (see Figure 4 below), remembering that the latest backup is always kept
- Whether you want to perform the initial backup over the network or offline, by shipping a disk to an Azure datacenter

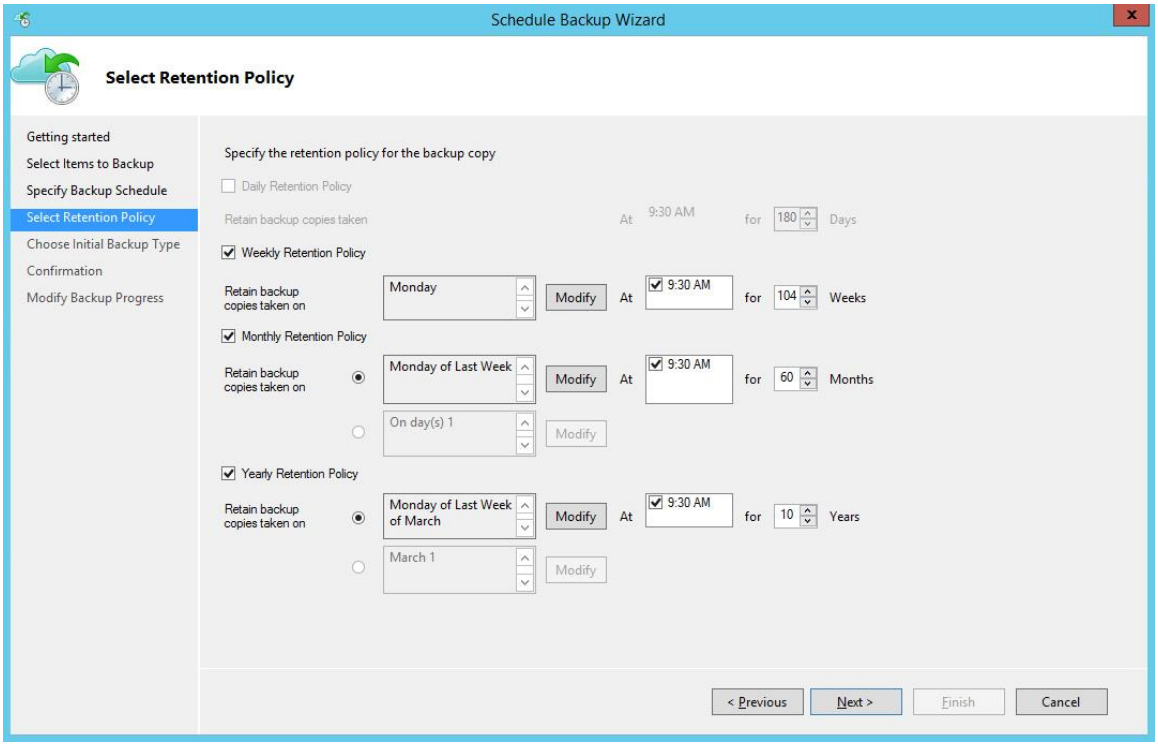

Figure 4: As of early 2015, Azure Backup retention policy options have expanded.

After you've created a schedule, you'll see a new option in the console's Action pane to "Backup Now." You can delete ("pause") the schedule if you like, but if you do so, the "Backup Now" option disappears again. If you only want to perform backups on demand without a schedule, Azure Backup is probably not for you.

If you're running Server 2012 or newer, you can also configure bandwidth throttling using the "Change Properties" link in the console. You can set different target speeds for work hours and non-work hours (see Figure 5). The "Schedule Backup" wizard doesn't prompt you to do this, but you should try to remember, because once you start a backup operation, you won't be allowed to change the throttling properties on the fly. Perhaps some good advice is to start with a small backup set, see how the traffic affects your network, and refine your throttling settings based on those observations.

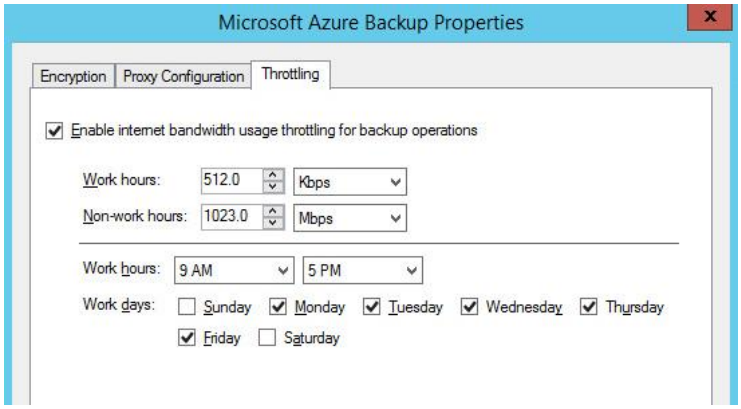

Figure 5: Throttling and other properties cannot be set during an in-progress backup.

Starting, Monitoring and Managing Backups

You'll use the backup console—either WBADMIN or WABADMIN—to start, manage, and track your backup and restore operations. For example, to see the amount of data transferred in a current backup job, double-click the job in either of these consoles. (Unlike the online portal, this information is up-to-date.) Perhaps in a future release we'll get an at-a-glance progress bar.

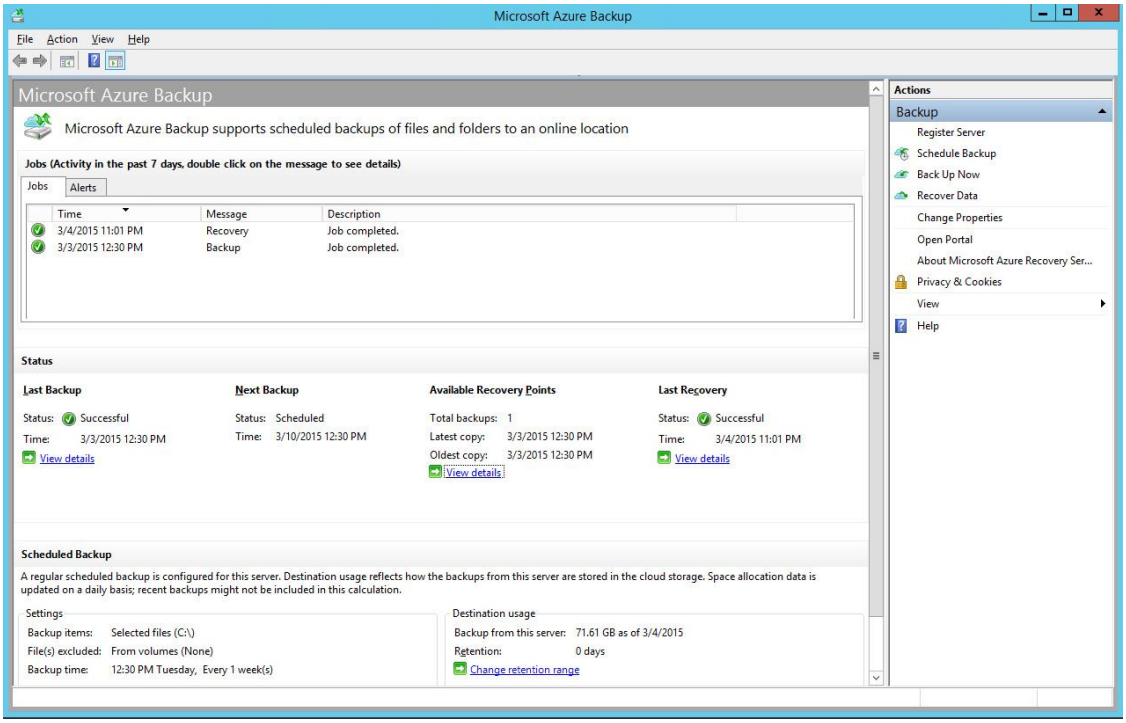

Figure 6: The standalone Azure Backup console, after a backup and restore operation.

You can also take a look at Resource Monitor (RESMON.EXE) to verify the network bandwidth. The relevant task will be CBENGINE.EXE (presumably "CB" stands for "Cloud Backup") and the target address will probably begin with "blob" as Azure uses blob storage for backups (see Figure 7 below).

By looking at the KB/sec transfer rate, you can do a bit of math and estimate the transfer time for your backup. (My estimate of thirty-one hours for my test backup was fairly close to the actual time of thirty-two hours.)

Figure 7: Resource Monitor's Network Activity display showing CBENGINE activity

You can take a peek at the event logs under "Application and Services Logs > CloudBackup > Operational" but you won't get a lot of information there, basically just "start" and "finish" notations.

Incidentally, Azure Backup seems to lack any pause/resume functionality. Your options are to let the backup run, or cancel it and start over later.

In addition to the lack of a multiserver administration console, there are no Group Policy settings for Azure Backup. Microsoft seems to be taking the tacit position that larger organizations will integrate Azure Backup with System Center 2012 Data Protection Manager to fill in the missing administration puzzle pieces.

If you need to perform a restore operation, the wizard-based procedure (launched from either MMC console's Action pane or menu) is straightforward if you are recovering to the same server:

- Select the server on which the backup was created
- Choose the recovery "mode" (browse or search, depending on whether you remember the location of the file or folder you wish to restore)
- Select the volume
- Select the item(s) to recover
- Specify recovery behavior (original or different location, copy or overwrite existing items, restore ACLs or not)

I found the restore performance to be quick, and the GUI slightly more informative than the backup GUI (see Figure 8). By the way, you can recover to a different server if the server is registered to the same backup vault, and if you can supply the encryption passphrase.

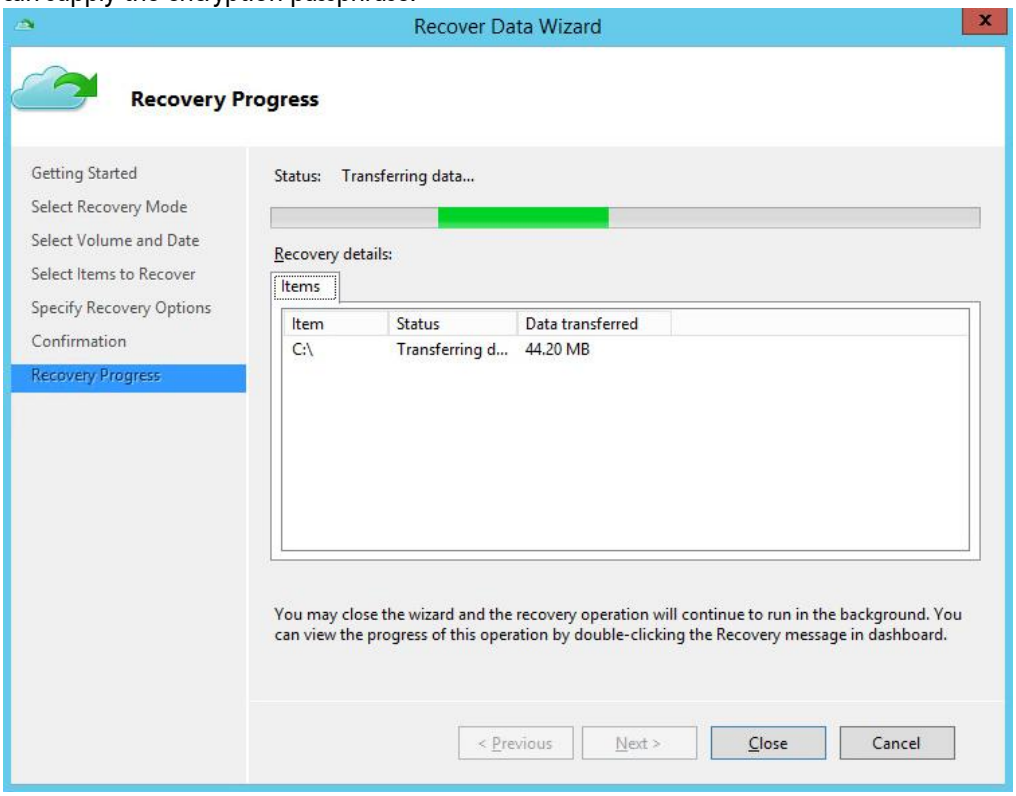

Figure 8: This screen doesn't show percent complete, but at least it shows MB transferred.

If you choose to make a copy of existing items, the restored file will have a naming convention similar to that shown below in Figure 9.

Figure 9: A restored copy with date information and "Copy" in the file name

Pricing and Administration

Price Structure

You pay for what you use, which sounds like a good concept, but the level of granularity is a bit daunting, in my opinion. Here's the structure as I write this, as best I can understand it from the multiple relevant pages on Microsoft's website.

- First, you will pay $10 per month per protected "instance" defined as a 500GB chunk of data, although Microsoft advises that this will be pro-rated if you don't use the service for part of the month. (What "don't use the service" actually means is not made entirely clear.)
- Second, you pay separately for storage space (storage meaning from Microsoft's standpoint, i.e., compressed size). The first 5GB is free; after that, the cost is $0.20 per month per GB.
- Third, you pay for "outbound data transfers" if you ever have to perform a data restore operation. The first 5GB per month is free; after that, up to 10TB per month, the price ranges between $0.087 and $0.181 per GB, depending on which geographical "zone" you are in (there are three zones covering the world).

To throw another variable into the mix, if you enter into an enterprise agreement, available from Microsoft licensing partners, you commit to paying at least $1,200/year and you get some discounts off the above-listed fees, but unused funds vanish at the end of the year.

The online price calculator at https://azure.microsoft.com/en-us/pricing/calculator may be of some help as you try to get a handle on what your costs will be.

Ongoing Management

To manage Azure Backup, use the online portal (manage.windowsazure.com) for anything pertaining to your Azure account: cost information, backup vaults, certificates, etc. (See Figure 10 below.) For example, to create a new backup vault, highlight "Recovery Services" then click the big plus sign on the command bar at the bottom left. (If you get tired of the Quick Start screen and prefer seeing the Dashboard right away when you click on a backup vault, click "Skip Quick Start the next time I visit.")

Be aware that the Azure portal is nowhere close to a realtime display; it may take a day or longer to reflect activity, such as the "TOTAL STORAGE PROTECTED" indicator. Also, the "CREDIT STATUS" button at the top takes a few seconds to appear after the rest of the page has loaded.

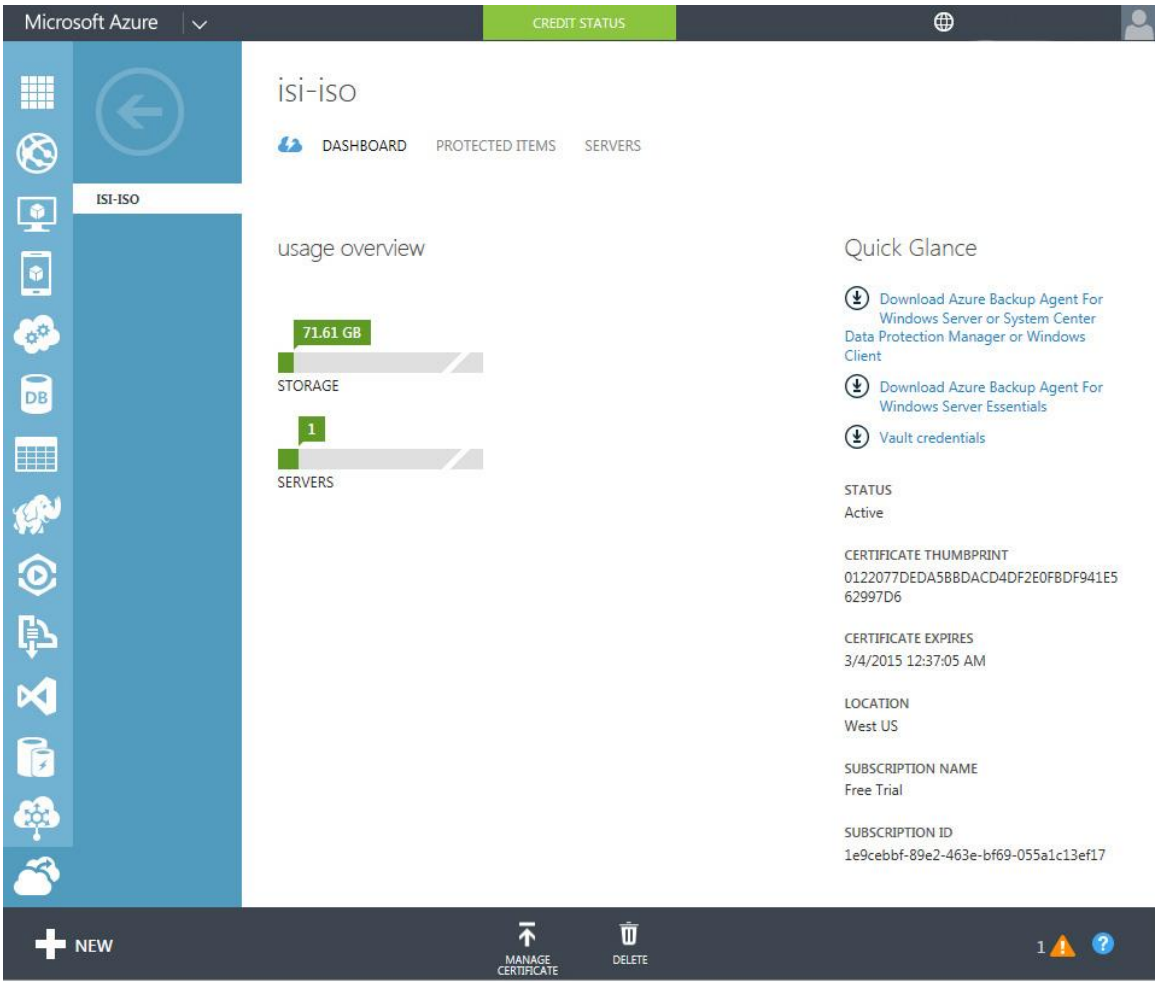

Figure 10: The Microsoft Azure management portal, showing the Dashboard for vault ISI-ISO. Note "Credit Status" link at top, and "New" and "Manage Certificate" buttons at bottom.

In addition to managing Azure Backup with the portal and the backup console(s), you can also use PowerShell. The secret to recognizing PowerShell commands pertaining to Azure Backup is to look for "OB" in the command name, presumably standing for "Online Backup. "

Here are a few, by way of example; learn about them all by visiting *http://msdn.microsoft.com/en-us/library/azure/hh831590.aspx*.

- **Start-OBRegistration** registers a server.
- **Set-OBMachineSetting** can set a passphrase, set proxy details, and/or set throttling parameters.
- **New-OBSchedule** creates a schedule object.
- **New-OBPolicy** creates a policy object (including schedule, files, and retention settings)
- **Get-OBPolicy | Start-OBBackup** makes a backup based on a policy.

Conclusion

On the plus side, Microsoft Azure Backup appears to be reasonably secure, reliable, and inexpensive; it can perform file-and-folder backups for most modern flavors of Windows, both server and client; and it integrates with System Center Data Protection Manager to provide support for a much wider variety of workloads.

On the minus side, the online and local tools could use a thorough Quality Control review; the pricing structure feels needlessly complex; a lot of capabilities are missing (pause-and-resume, *ad hoc* backups without a schedule, system state backup, bare metal restore, Group Policy settings, etc.); and the consoles are server-centric, making multi-server management tedious.

For cloud-based data backup, then, Azure Backup therefore seems well suited for the ends of the company-size spectrum: small companies (like mine) that don't have many servers, and large companies that will integrate it with System Center DPM. If Microsoft provides a multi-server management tool that doesn't require DPM, it would become more appealing to midsized organizations as well.

By the way: If you experiment with Microsoft Azure Backup and would like to share your impressions with other IT professionals, or just read about what others have to say, visit http://feedback.azure.com/forums/258995-azure-backup.

Learn More

Learn more about how you can improve productivity, enhance efficiency, and sharpen your competitive edge through training.

- Configuring Advanced Windows Server 2012 Services (M20412)
- Monitoring and Operating a Private Cloud with System Center 2012 R2 (M20246)
- Upgrading Your Skills to MCSA Windows Server 2012 (M20417)

Visit **www.globalknowledge.com** or call **1-800-COURSES (1-800-268-7737)** to speak with a Global Knowledge training advisor.

About the Author

Glenn Weadock is a longtime instructor for Global Knowledge and teaches Windows 8, Windows 7, Vista, Server 2012, Server 2008, and Active Directory. He co-developed with Mark Wilkins two advanced Server 2008 classes in the Microsoft Official Curriculum. Glenn also consults and provides expert witness services in patent infringement cases through his Colorado-based company Independent Software, Inc. (www.i-sw.com).