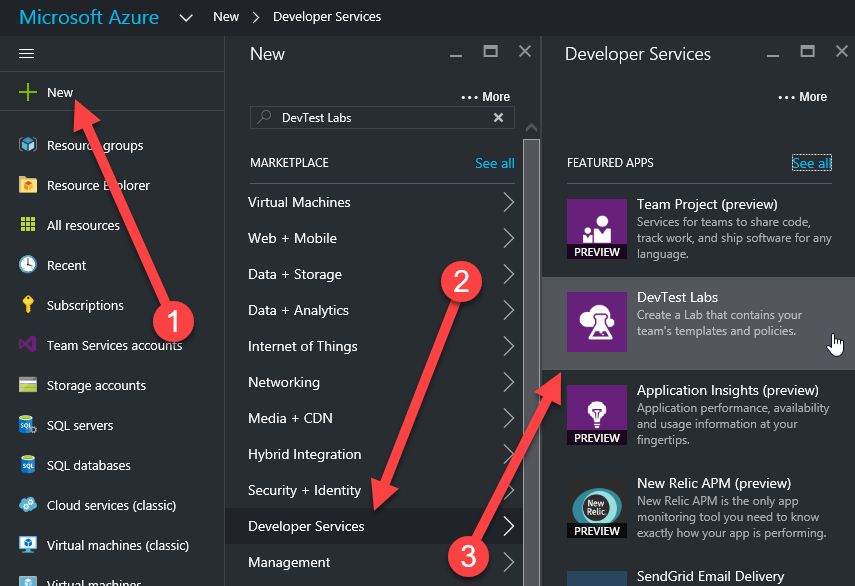
Managing the Azure for Computer Science DevTest Lab

# Initial Setup

The first step is to actually create an Azure DevTest Lab instance within an Azure subscription. These instructions assume you already have a configured Azure subscription available to use. This should generally \*not\* be a Pass/promotional account, since there will likely be billing charges accrued against this account, so it should either be an account backed by an Enterprise Agreement, a Pay-As-You-Go account with a credit card securing the account, or some other kind of account where billable resources can be provisioned.

## Crete the DevTest Lab Instance

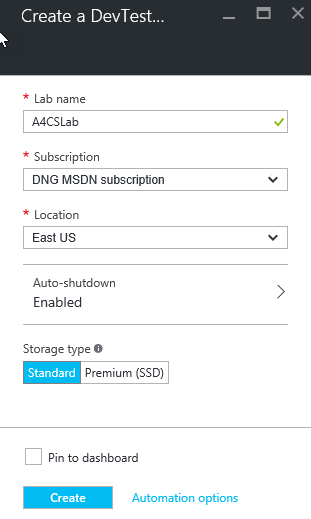
From the Portal select +New from the upper left corner, then select Developer Services from the Marketplace category listings. In the **Developer Services** blade, within the **Featured Apps** listing, select **DevTest Labs**.



In the **Create a DevTest Lab** blade, enter the configuration information for the new lab instance, including:

* A name for the lab such as ***A4CSLab*** (note – the lab name must be unique within an Azure subscription.)
* If you have more than one subscription available to your current Azure login, select the subscription which will be billed for the lab.
* Select the Location (region) for the lab
* Select the Auto-shutdown option to configure whether or not you want Auto-shutdown to be enabled by default for new VM’s provisioned within this subscription, and if so, what time to select for the shutdown.
* Select **Standard** as the **Storage Type** option. Currently Azure DevTest labs should be able to handle 50-60 VMs or so with a Standard Storage account before running into Storage Account throughput limitations. If the lab to be provisioned requires more VM’s, then consideration should be given to either using 2 DevTest lab instances, or to using Premium Storage.

Press the **Create** button to start the creation process for the DevTest lab instance.



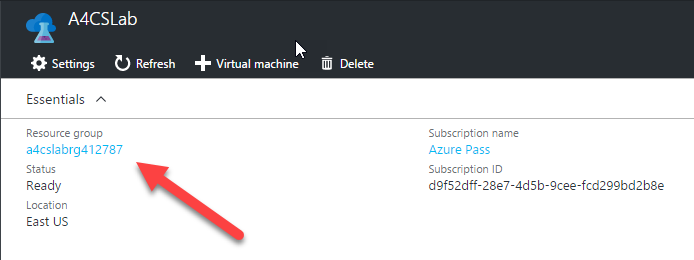
## Upload the VM Image

The next step in setting up the lab is to upload the custom VM Image that will be used as the template for VM’s created to support the Azure for Computer Science labs. This step will be time consuming (may be as long as an hour or longer, depending on tools being used.)

The first step is to locate the VM Storage account being used by the DevTest Lab that was just created. When the DevTest lab instance is created, 2 Storage Accounts are created for it within the same resource group. One Storage Account is used for VM’s and the other is for other Artifacts. The key to locating the correct one for VM’s is to find the one that was created with an “Uploads” container pre-provisioned.

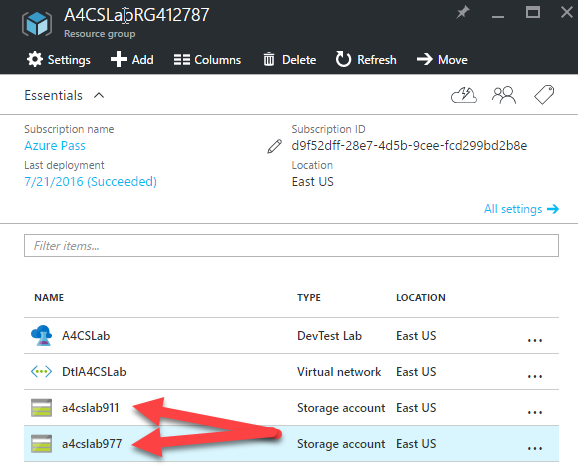
*Note – the following steps use the Azure Portal to locate the correct account. It can also be done with tools like Azure Storage Explorer.*

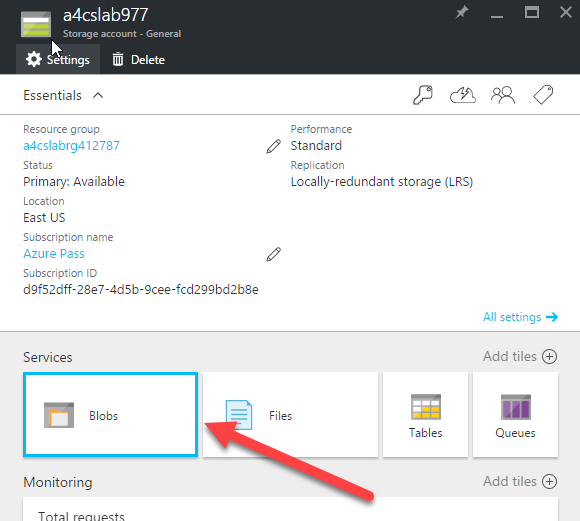
Open the lab’s Resource Group blade in the Azure Portal (one way to do this is to open the blade for the created DevTest lab in the Azure Portal and click on the name/link of the Resource Group there.)

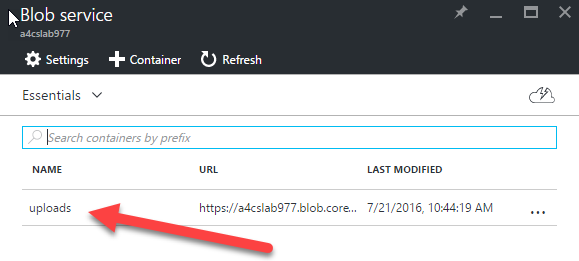


From the **Resource Group** blade, open each Storage Account in turn to check to see which one has the content that marks it as the one that DevTest labs uses for VM storage. To do this, click on the Storage Account entry in the resource table, which will open the blade for that Storage Account. Click on the Blobs button, and in the **Blob service** blade that is displayed, look for a container named ***uploads.*** The Storage Account that you are interested in is the one which has this container pre-provisioned.

Make a note of the storage account name and access keys to use to copy content into this storage account.





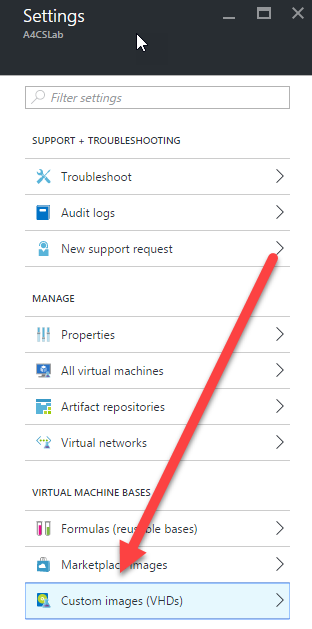


Using Azure Storage Explorer (or an Azure Storage management & copy tool of your choosing), copy the lab’s VM image (as of this writing it is A4CSVM\_20160721.vhd) into this ***uploads*** container. This is the step that may take some time. If possible, prefer sourcing the VM from another Azure Storage account within the same region.

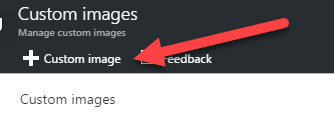
## Create/Register the Custom Image in the Lab

Once the Image file has been successfully uploaded to the lab’s VM Azure Storage Account, the lab needs to be told about the image so it can be used as a template for the VMs it creates.

In the Azure Portal blade for the DevTest Lab instance, select **Custom Images (VHDs)** from the **Settings** panel.

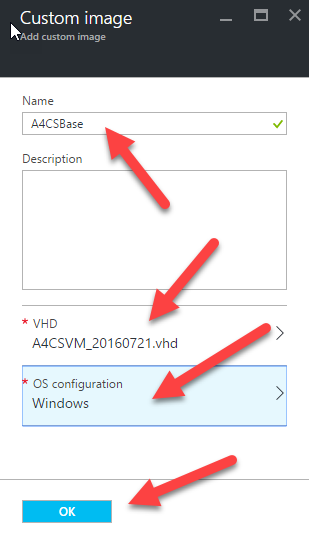


Click on the **+ Custom image** button to create a new Custom Image entry.



Configure the Custom Image in the **Custom image** blade.

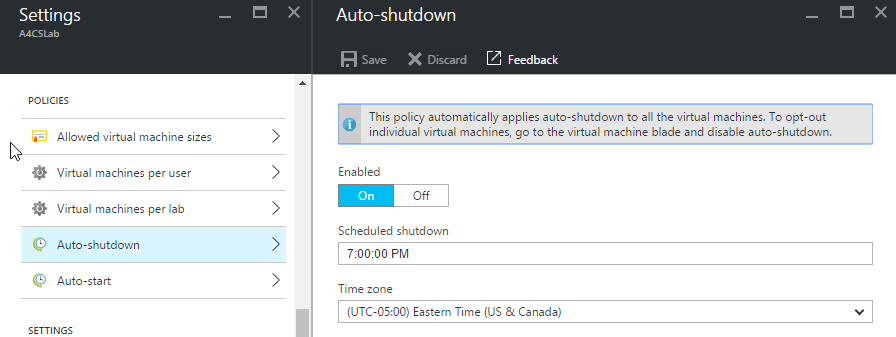
* Enter the name ***A4CSBase*** for the Custom Image.
* Select **VHD options** and click on the name of the VHD you uploaded under **Use an existing VHD** and press **OK** to accept and close the **VHD** blade.
* Select **OS configuration** and select **Windows** for the **OS type**. Check the box next to **I have run sysprep on the Virtual Machine**. Press **OK** to accept and close the **OS configuration** blade.
* Press OK to accept the stings and close the **Custom image** blade.



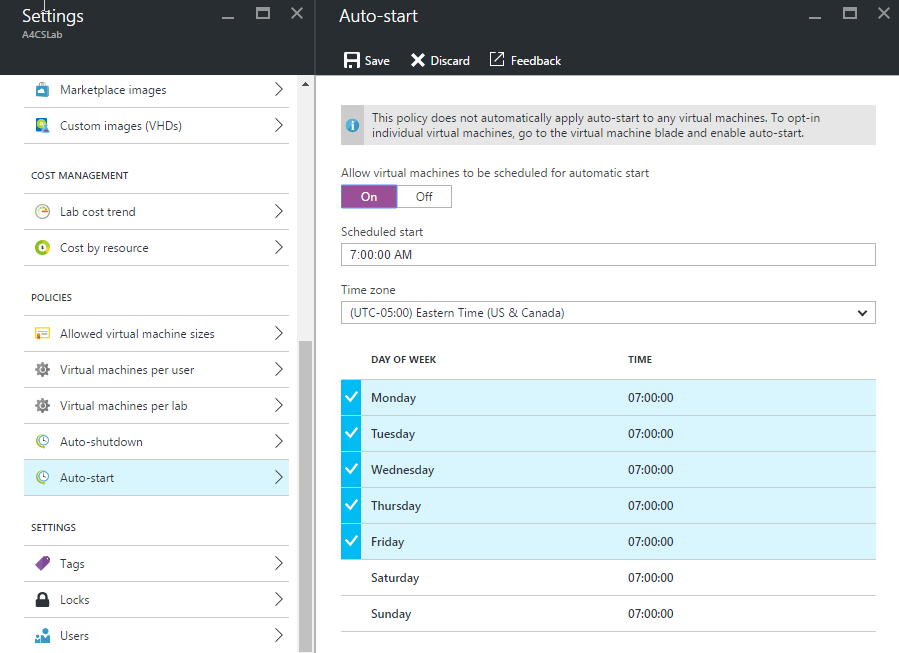
## Configure Automatic Startup & Shutdown

DevTest labs allows setting several different kinds of “policies” which govern the behavior of the lab and the VMs it contains. These policies include “Auto-Start” and “Auto-Shutdown” policies. By default, your lab was created with an Auto-Shutdown policy set to stop running VM’s at 7PM in the lab’s local time zone, and no auto-start policy.

To configure or manage the lab’s auto-shutdown policy, select **Auto-shutdown** from the **Policies** section of the lab instance’s **Settings** blade. In the **Auto-shutdown** blade, you can enable or disable the availability of the policy and set the scheduled shutdown time and its time zone.



To configure or manage the lab’s auto-start policy, select **Auto-start** from the **Policies** section of the lab instance’s **Settings** blade. In the **Auto-start** blade, you can enable or disable the availability of the policy, set the scheduled start time and its time zone, and select on which days of the week the machines will be started.



Note that auto-start allows choosing on which days the machines that opt into the policy in the lab will be started, but auto-shutdown runs every single day of the week.

Once the policies are defined, you can apply them to individual VM’s in the lab. Instructions for doing so are included below in the ***Starting and Stopping VMs*** section.

*Note – Bear in mind that the Startup and Shutdown times are set in a given time zone. Setting and opting into an auto-shutdown time of 7PM in the US Eastern time zone might pose a problem if the next lab is being done on the US West coast – resulting in a 4PM shutdown of the VMs. (Similar challenges exist for the auto-start time.)*

## Core Count

Azure Subscriptions include a default limit on the number of cores that can be provisioned at any given time within a given region. This core count can be increased via a request to Azure Support. So part of the initial (and ongoing) lab setup should be to ensure that the Azure Subscription being used has a high-enough core count to support the desired number and size of VM’s. Additional information about the applicable limits can be found at <https://azure.microsoft.com/en-us/documentation/articles/azure-subscription-service-limits/>

*Note – This request could take several hours to process. This is not something that should be postponed until just before the lab is getting ready to start.*

# Hosting an Event

## (Re)Provisioning VMs

Prior to the event, the VM’s for the lab will need to be provisioned within the DevTest lab (if they haven’t been already). You should \*not\* reuse the same VM’s from a prior lab, as they will include the changes that previous users made to the VM’s.

A PowerShell module has been created which includes scripts for several of the key features required to manage the lab VM’s. The commands provided include:

|  |  |
| --- | --- |
| Reset-Azure4CSLab | Deletes any existing VM’s in the lab and creates new VM’s (basically a combination of Remove and Add) |
| Add-Azure4CSLabVMs | Creates a desired number of Lab VM’s from a VM Image |
| Remove-Azure4CSLabVMs | Clears the Lab VM’s |
| Start-Azure4CSLabVMs | Starts any stopped/deallocated Lab VM’s |
| Stop-Azure4CSLabVMs | Stops any running Lab VM’s (moves to Stopped/deallocated state where the Azure subscription is not billed for VM time.) |

To use the PowerShell scripts, start PowerShell, navigate to the directory that contains the PowerShell module, and import the “ManageLab.psm1” module using the following command.

Import-Module .\Wintellect.DevTestLabManagement.psm1 -Force

*Note – the “Force” switch is optional, and instructs PowerShell to import the module even if it has been already imported, in case the module has been updated/changed.*

Before running any of the commands from this module, you will need to sign to the Azure subscription where the lab has been provisioned. This can be done with the following command, which will present a sign-in dialog.

Add-AzureRmAccount

Prior to a lab, the most likely/useful command to use is Reset-Azure4CSLabVMs. For parameters, this function requires the Lab Name and the Custom Image Name. It will optionally accept the size of the VMs to provision, the quantity of VM’s to provision, the Root Name to use for the VM’s that are provisioned (the index of the VM is tacked onto the name – such as A4CSVM001, A4CSVM002, etc.), and the username and passwords to be used to log into the VM’s (all of the machines share the same username and password.) An optional flag value of InitiallyStopVMs can also be set to instruct the VM’s to be transitioned to a Stopped state (so the VM compute time isn’t billing until it is needed). There are also several other parameters available. Documentation for the full set of available parameters can be seen using the get-help PowerShell command.

*Note - if the VM count is omitted or set to zero and the reset is being done on a lab that already contains VM’s, the number of VM’s being deleted will be counted and that will be used as the number of VM’s to provision, basically resetting the lab in-place.*

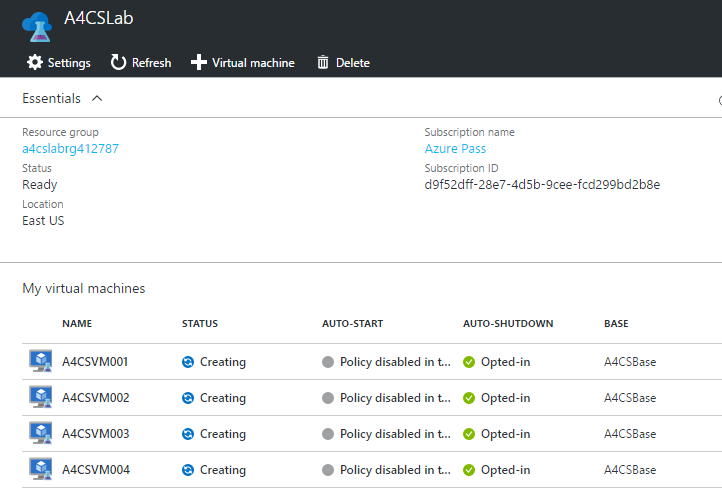
To provision a lab to a desired number of VM’s, with everything else set to the default, and the VM’s initially in a Stopped State, use the following command:

Reset-Azure4CSLabVMs -LabName A4CSLab -NumberOfCopies 4 -InitiallyStopVMs

Alternatively, to re-provision a lab that already contains VM’s (and keep the exact same number of VM’s), also resulting in stopped VMs, the command can be simplified to:

Reset-Azure4CSLabVMs -LabName A4CSLab -InitiallyStopVMs

In the Azure Portal, you will be able to see the new VM’s being provisioned within the lab.

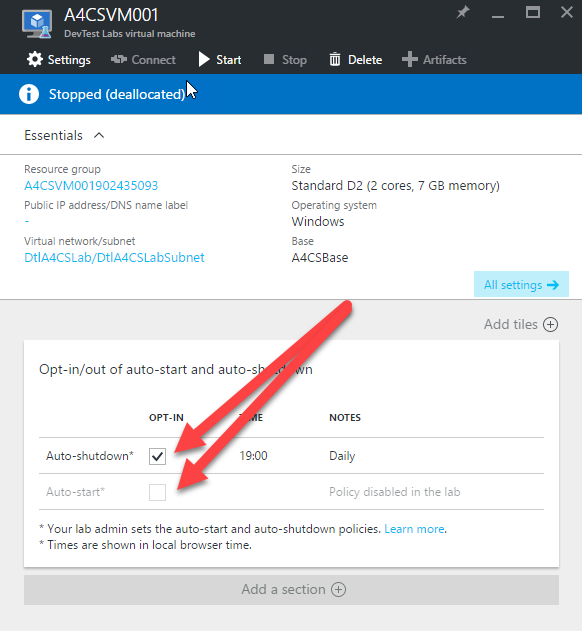


*Note – see prior note about core limits before provisioning VMs.*

## Starting and Stopping VMs

As mentioned previously, DevTest Labs supports setting Auto-Start and Auto-Stop policies which will automatically start and stop the selected VMs in a DevTest Lab instance. When set & enabled, an individual VM can opt in to either or both of these policies.

When VMs are created within a DevTest lab, they are automatically enrolled into any active policies. To manage VM enrollments in these policies (add them to a new policy, remove them from an existing one, etc.), select the VM instance in the lab instance’s blade in the Azure Portal. This will open the Lab VM blade for the selected VM. In this blade, there are checkboxes available for opting in or out of the individual policies.



In addition to leveraging policies to automatically start or stop VM’s, the provided PowerShell scripts include functions that can be called to immediately either start or stop all of the VM’s in a given lab.

*Note – An Azure Subscription will be billed for any & all VM’s that are not in a Stopped/Deallocated state. It is very important to remember to shut down unused VM’s to prevent unwanted billing.*

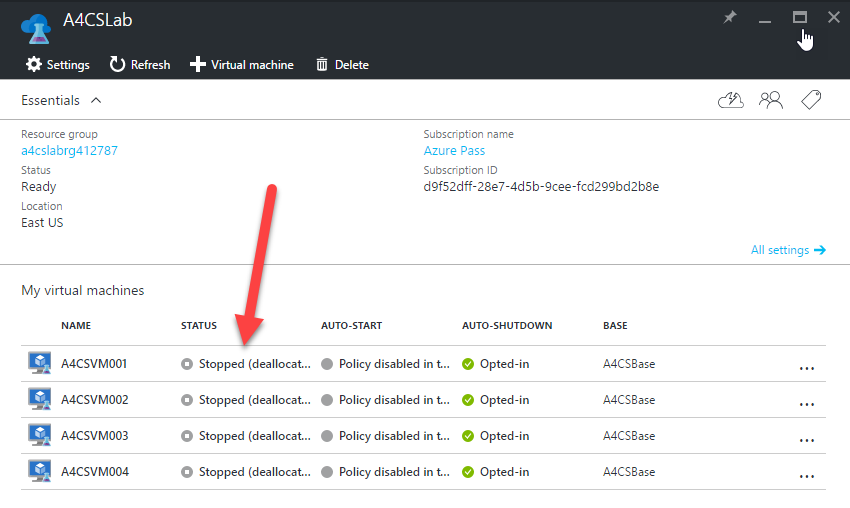
To manually stop all of the VM’s in a lab, use the following PowerShell command:

Stop-Azure4CSLabVMs -LabName A4CSLab

Likewise, to manually start all of the VM’s in a lab, use the following PowerShell command:

Start-Azure4CSLabVMs -LabName A4CSLab

The started/stopped state of VM’s in the lab can be seen in the Azure Portal in the DevTest lab blade.



In addition to the Auto-Start & Auto-Stop policies and the PowerShell scripts, individual VM’s can also be started and stopped through the Azure Portal in the DevTest lab blade. By selecting the ellipses next to the lab’s name and choosing **Start** or **Stop** from the menu that is shown.

## At the Conclusion of an Event

As was previously mentioned, the lab should be reset at the conclusion of an event. There are basically three options that can be used to prevent unnecessary billing:

* Reset the lab & stop all of the lab’s VM’s (use Reset-Azure4CSLabVMs)
* Clear the lab and add the new VM’s prior to the next event (use Remove-Azure4CSLabVMs)
* Stop all of the lab’s VM’s and reset it later (use Stop-Azure4CSLabVMs)