**Azure PowerShell, ARM template, DSC**

Before you begin the tasks, review the following training materials and articles:

* [Azure Resource Manager DevOps Jump Start](https://mva.microsoft.com/en-US/training-courses/azure-resource-manager-devops-jump-start-8413?l=rqSZQ7Jz_7304984382)
* [Microsoft Azure for IT Pros Content Series: Storage](https://mva.microsoft.com/en-US/training-courses/microsoft-azure-for-it-pros-content-series-storage-17237?l=ToQqXzuZD_6306218965)
* [Azure PowerShell](https://docs.microsoft.com/en-us/powershell/azure/overview?view=azurermps-5.7.0&viewFallbackFrom=azurermps-5.2.0), [Install Az Module](https://docs.microsoft.com/en-us/powershell/azure/install-az-ps?view=azps-1.3.0) and [Azure Virtual Machine PowerShell samples](https://docs.microsoft.com/en-us/azure/virtual-machines/windows/powershell-samples?toc=%2fpowershell%2fmodule%2ftoc.json&view=azurermps-5.2.0)
* [Getting Started with PowerShell Desired State Configuration (DSC)](https://mva.microsoft.com/en-US/training-courses/getting-started-with-powershell-desired-state-configuration-dsc-8672?l=ZwHuclG1_2504984382)
* [Advanced PowerShell Desired State Configuration (DSC) and Custom Resources](https://mva.microsoft.com/en-US/training-courses/advanced-powershell-desired-state-configuration-dsc-and-custom-resources-8702?l=3DnsS2H1_1504984382)
* [Internet Information Server](https://www.youtube.com/watch?v=Laib4ynCrUk&list=PLyKWK-QrXDMCSdHJ_yiD_PO9seooO9acL)

1. Create a storage account and a BLOB-type container. Upload all local folders and files to that container. (Azure PowerShell)
2. Create Azure SQL and create 3 databases on the server. For testing purpose do insert for each database. [Export full backup databases](https://github.com/Huachao/azure-content/blob/master/articles/sql-database/sql-database-export-powershell.md) and [import to another Azure SQL server](https://docs.microsoft.com/en-us/azure/sql-database/sql-database-import). (Azure PowerShell)
3. After logging into Azure via PowerShell, save the credentials in a file that you can use in the future in the PowerShell script for the login process. Have a look [Auto login to Azure with PowerShell](https://4sysops.com/archives/auto-login-to-azure-with-powershell/) and [Persist user credentials across PowerShell sessions](https://docs.microsoft.com/en-us/powershell/azure/context-persistence?view=azurermps-5.7.0) articles. (Azure PowerShell)
4. Create an ARM template to create a Windows Server 2016 Datacenter with a DNS name and a DSC extension. In the folder where ARM template is located add a new Artifacts folder and create a text file in this folder. Using PowerShell DSC extension copy this file to the C:\Artifacts folder on the Azure VM. Have a look [Structure and syntax of Azure Resource Manager templates](https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-authoring-templates), [Creating and deploying Azure resource groups through Visual Studio](https://docs.microsoft.com/en-us/azure/azure-resource-manager/vs-azure-tools-resource-groups-deployment-projects-create-deploy) , [Working with Azure ARM Templates in Visual Studio Code](https://cmatskas.com/working-with-azure-arm-templates-in-visual-studio-code/), [Key concepts of DSC Resources](https://docs.microsoft.com/en-us/powershell/dsc/resources/resources), [DSC xPSDesiredStateConfiguration module](https://github.com/PowerShell/xPSDesiredStateConfiguration).
5. Create an ARM template to create a Windows Server 2016 Datacenter with a DNS name, DSC and Custom Script extension. In the folder where ARM template is located add a new Shell folder and create myscript.ps1 PowerShell script in this folder. Using PowerShell DSC extension copy this script to the Azure VM. From Custom Script extension run myscript.ps1 and pass two parameters. Have a look [Custom Script Extension for Windows](https://docs.microsoft.com/en-us/azure/virtual-machines/extensions/custom-script-windows)
6. Create an ARM template to create a Windows Server 2016 Datacenter with a DNS name and a DSC extension. Configure through DSC two different sites with different ports on IIS. Have a look [xWebAdministration](https://github.com/PowerShell/xWebAdministration) and [Configuring IIS with DSC](http://www.jasonhelmick.com/2016/04/24/configuring-iis-remote-management-with-dsc/)