

|  |
| --- |
| Azure DevOps Lab  **Microsoft Azure** |
| Module 9. Azure Automation  Home task |

TASK 1

1. To work through the [Azure Automation](file://epbyminsa0000.minsk.epam.com/Training%20Materials/EPAM%20Trainings/Microsoft%20Azure%20Program/Azure%20Automation.pptx) document and perform the task described.
2. Use the ARM template for an initial deployment created in the Home task #4, which will be used for the linked templates execution.
3. Additionally, to the previously created VNET, Key Vault and VM templates, create a linked template that will deploy an Automation Account containing the following resources:
4. Runbook with an uploaded runbook script file linked
5. Credentials that will be used to execute your runbook
6. Job that will be triggered after the creation of the runbook
7. Run the deployment of the initial ARM template and deploy required resources.
8. Use SAS tokens for accessing all templates and runbook file in this task
9. Job Name is generated dynamically
10. Key Vault secret is generated in PS and passed as a secureString or secureObject into the ARM template

Result: After deployed VM Job is executed, previously running VM is stopped.

TASK HINTS

1. Pay attention that for this task execution you must pre-create a user in your Azure AD tenant with the permissions enough for manipulating VMs in the subscription. You may create an App Registration and assign a VM Contributor role to it. Pass principal’s Id and Key as parameters in your deployment. You should also use the attached [runbook](file://epbyminsa0000.minsk.epam.com/Training%20Materials/EPAM%20Trainings/Microsoft%20Azure%20Program/Workflow_Stop-AzureVM.ps1) and update it with the creds and other required parameters.
2. You should give your trainer an Owner permission to the App Registration SPN in order to check your deployment.
3. Add your Key Vault, VNET, VM, Automation Account deployment templates to the initial template as resources.

TASK 2

1. To work through the [Azure Automation DSC](file://epbyminsa0000.minsk.epam.com/Training%20Materials/EPAM%20Trainings/Microsoft%20Azure%20Program/Azure%20Automation%20DSC.pptx) document and perform the task described.
2. Append ARM templates used previously with one more VM creation task.
3. Add the [DSC configuration](file://epbyminsa0000.minsk.epam.com/Training%20Materials/EPAM%20Trainings/Microsoft%20Azure%20Program/IISWebsite.ps1) to the Automation Account template.
4. Compile the DSC configuration, onboard your VMs, assign DIFFERENT DSC nodes configurations to your VMs via linked ARM templates.
5. DSC nodes’ names are not hardcoded in the DSC configuration
6. DSC modules to support configuration appliance are uploaded to the Automation Account via ARM
7. Automation Account is linked without hardcoding its name

Note**:** When appending your templates from the previous tasks, pay attention to the unused and unrequired code - its usage will be penalized

Result: Two VMs are assigned to the Azure Automation DSC Nodes, separate DSC configurations are applied.

REQUIREMENTS

1. One JSON file for describing all Azure resources is forbidden. **Please use linked templates.**

2. Linked templates **must be called from the initial deployment template** (main.json).

3. Main and parameters templates must be named as **main.json** and **parameters.json** accordingly.

4. Maximum **number of parameters** in Main.json is 7.

5. Main.json and parameters.json must be **executed from local folder**. Using **-TemplateParameterUri** and **-TemplateUri** options in PS script is forbidden.

6. All artifacts (JSONs and PS files) must be stored in **Azure Storage Account**. Using any GitHubs or other public repos is forbidden.  
7. Create a PowerShell/Bash (using Azure CLI) deployment script, which will be used for running all your deployments. The script should have the following functionality:

                a. Create resource group.

b. Create storage account and container within for artifacts (For example: JSONs, PS file(s), ZIP files).

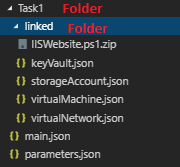
c. Upload the linked templates and other task-related artifacts to the created Storage Account.

d. Execute main.json file for deploy Azure resources.

8. Each ARM json file must have at least **1 output**.

9. All homework **artifacts must be executable** (e.g. if Mentor starts your script execution and it fails - all homework artifacts will be sent back for fixing)

10. All **resources must be deleted** after homework completion.

11. Use the next **folder structure** for storing artifacts. **Subfolder** with resources JSONs must be named **“linked”:**  


Useful links

[Get started with Azure Automation](https://docs.microsoft.com/en-us/azure/automation/automation-offering-get-started)

[Update your Automation account authentication with Run As accounts](https://docs.microsoft.com/en-us/azure/automation/automation-create-runas-account)

# String functions for [Azure](https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-template-functions-string#uniquestring) Resource Manager templates

# [How to: Use the portal to create an Azure AD application and service principal that can access resources](https://docs.microsoft.com/en-us/azure/active-directory/develop/howto-create-service-principal-portal#required-permissions)

[Microsoft.Automation automationAccounts template reference](https://docs.microsoft.com/en-us/azure/templates/microsoft.automation/2015-10-31/automationaccounts)  
  
[Onboarding machines for management by Azure Automation State Configuration](https://docs.microsoft.com/en-us/azure/automation/automation-dsc-onboarding)  
  
[Compiling DSC configurations in Azure Automation State Configuration](https://docs.microsoft.com/en-us/azure/automation/automation-dsc-compile)