

How to Deploy a CloudGen Firewall via Azure Templates

<https://campus.barracuda.com/doc/72515972/>

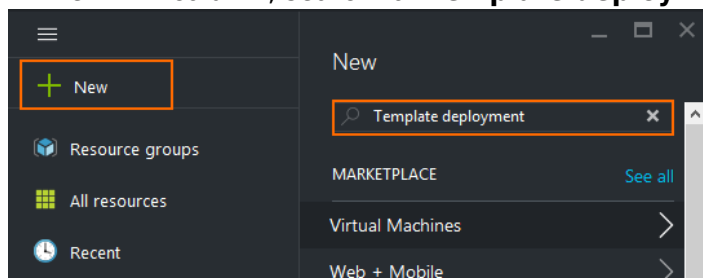
Azure templates are JSON files containing resource descriptions and parameter definitions. These parameters can be passed to the template during deployment either on the command line, through an object, or through a parameter template file. You can deploy templates using Azure Portal, PowerShell, CLI, Rest API, or directly from Visual Studio.

Before You Begin

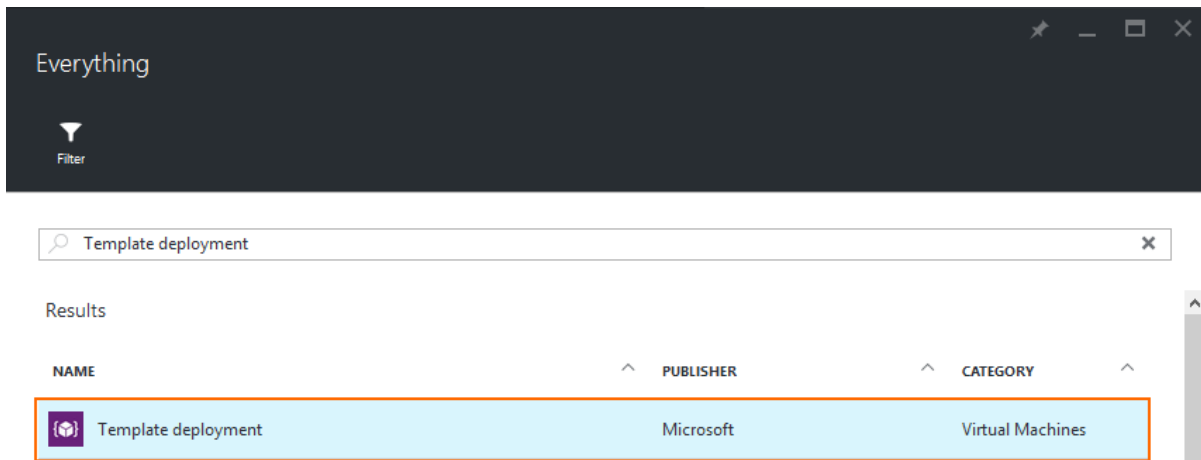
- Create an Azure template either manually or use the **Export Template** feature to export a template for an existing Resource Group.
- (optional) Modify the template to retrieve the PAR file from your Control Center on deployment.
- (PowerShell only) Install Azure PowerShell 1.3.0 or higher.
- (PowerShell only) Create a template parameter file containing the values of the template parameters.

Deploy an Azure Template via Web Portal

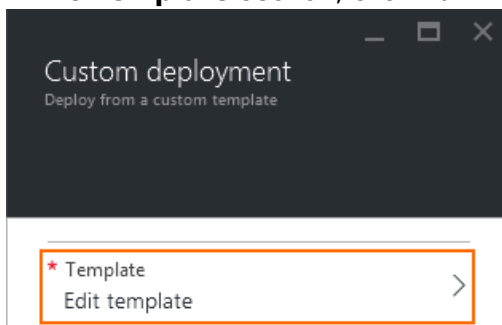
1. Go to <https://portal.azure.com>.
2. In the upper left corner, click **NEW**.
3. In the **NEW** column, search for **Template deployment**. The **Everything** column opens.



4. Click **Template deployment**.



5. In the **Template deployment** column, click **Create**. The **Custom deployment** column opens.
6. In the **Template** section, click **Edit template**.



7. Replace the default template with your custom Azure template.
8. Click **Save**.

Edit template
 Edit your Azure Resource Manager template

↑ Quickstart template
 ↓ Download template

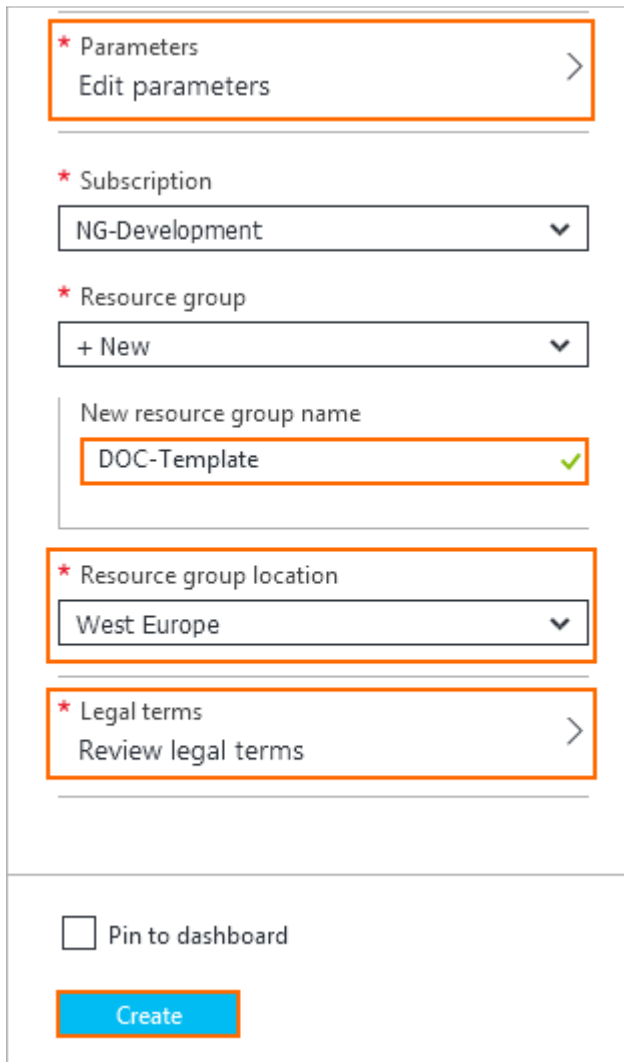
- ▶ Parameters (12)
- ▶ Variables (7)
- ▼ Resources (4)
 - [parameters('availabilitySetName')]...
 - [variables('publicIPAddressName')]...
 - [variables('nicName')] (Microsoft....
 - [parameters('vmName')] (Microsof...

```

170     "hardwareProfile": {
171         "vmSize": "[parameters('vmSize')]"
172     },
173     "osProfile": {
174         "computername": "[parameters('vmName')]",
175         "adminUsername": "azureuser",
176         "adminPassword": "[parameters('adminPasswor
177     },
178     "storageProfile": {
179         "osDisk": {
180             "name": "[concat(parameters('vmName'), '-
181             "osType": "linux",
182             "caching": "ReadWrite",
183             "createOption": "FromImage",
184             "image": {
185                 "uri": "[variables('userImageName')]"
186             },
187             "vhd": {
188                 "uri": "[variables('osDiskVhdName')]"
189             }
190         }
191     },
192     "networkProfile": {
193         "networkInterfaces": [
194             {
195                 "id": "[resourceId('Microsoft.Network/n
196             }
          
```

Save
Discard

9. In the **Parameters** section, click **Edit parameters**.
10. Enter the **Parameters** required by the template, and click **OK**.
11. Select an **Resource group** or enter the **New resource group name**.
12. Click **Review legal terms**.
13. Click **Purchase**.



* Parameters
Edit parameters >

* Subscription
NG-Development v

* Resource group
+ New v

New resource group name
DOC-Template ✓

* Resource group location
West Europe v

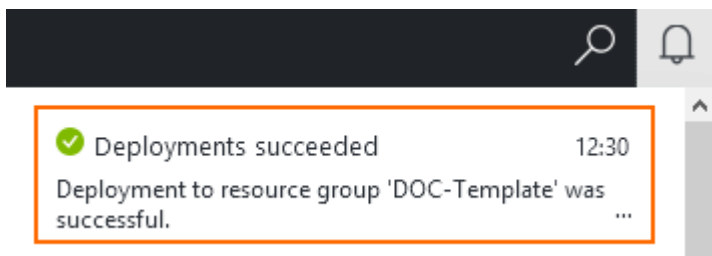
* Legal terms
Review legal terms >

Pin to dashboard

Create

14. Click **Create**.

Wait for the deployment to be completed, and then use the public IP address to log in to your CloudGen Firewall or Control Center VM.



Deploy an Azure Template via Azure PowerShell

1. Open Azure PowerShell.

2. Log into your Azure account

```
Login-AzureRmAccount
```

3. Create a new Resource Group.

```
New-AzureRmResourceGroup -Name RESOURCE_GROUP_NAME -Location "YOUR_LOCATION"
```

```
PS E:\> New-AzureRmResourceGroup -Name DOC-Template -Location westeurope
WARNING: The output object type of this cmdlet will be modified in a future release.

ResourceGroupName : DOC-Template
Location           : westeurope
ProvisioningState  : Succeeded
Tags               :
ResourceId         : /subscriptions/bde58b49-9951-466e-90e2-592c0920ce77/resourceGroups/DOC-Template
```

4. Test your template and parameter template JSON file.

```
Test-AzureRmResourceGroupDeployment -ResourceGroupName RESOURCE_GROUP_NAME -TemplateFile YOUR_TEMPLATE_FILE -TemplateParameterFile YOUR_PARAMETER_FILE
```

```
PS E:\> Test-AzureRmResourceGroupDeployment -ResourceGroupName DOC-Template -TemplateFile E:\azure\NextGenFirewallTemplate\NextGenFirewallAzureTemplate.json -TemplateParameterFile E:\azure\NextGenFirewallTemplate\NextGenFirewallAzureParameterFile.json
PS E:\>
```

5. Deploy the template. If the Mode parameter is not set, incremental mode is used.

```
New-AzureRmResourceGroupDeployment -Name DEPLOYMENT_GROUP_NAME -ResourceGroupName RESOURCE_GROUP_NAME -TemplateFile YOUR_TEMPLATE_FILE -TemplateParameterFile YOUR_PARAMETER_FILE
```

```
PS E:\>
PS E:\> New-AzureRmResourceGroupDeployment -Name NGF-Deployment -ResourceGroupName DOC-Template -TemplateFile E:\azure\NextGenFirewallTemplate\NextGenFirewallAzureTemplate.json -TemplateParameterFile E:\azure\NextGenFirewallTemplate\NextGenFirewallAzureParameterFile.json
WARNING: The output object type of this cmdlet will be modified in a future release.

DeploymentName      : NGF-Deployment
ResourceGroupName   : DOC-Template
ProvisioningState   : Succeeded
Timestamp           : 28.04.2016 11:53:33
Mode                : Incremental
TemplateLink        :
Parameters          :
                    Name                Type                Value
                    =====
userImageUhdName    String              GWAY-7.0.0-PAYG.vhd
userImageResourceGroupName String              DOC-Storage
userImageStorageAccountName String              docstorage0
userImageStorageContainerName String              vhd
vmName              String              NGF-Template
virtualNetworkName String              DOC-UNET
virtualNetworkResourceGroup String              DOC-Networking
subNetName          String              NGF
adminPassword       SecureString
location            String              westeurope
vmSize              String              Standard_A1
availabilitySetName String              AU-Set-Template

Outputs
DeploymentDebugLogLevel :
```

You can now log in to the CloudGen Firewall or Control Center VM you just deployed with your template.

Next Steps

Configure a user defined routing table for the backend VMs to send traffic through the firewall, and enable Azure Cloud Integration to allow the firewall VM to directly connect to the Azure service fabric.

For more information, see [How to Configure Azure Route Tables \(UDR\) using Azure Portal and ARM](#).

Figures

1. template_web_portal_01.png
2. template_web_portal_02.png
3. template_web_portal_04.png
4. template_web_portal_03.png
5. template_web_portal_05.png
6. template_web_portal_06.png
7. template_ps_01.png
8. template_ps_02.png
9. template_ps_03.png

© Barracuda Networks Inc., 2019 The information contained within this document is confidential and proprietary to Barracuda Networks Inc. No portion of this document may be copied, distributed, publicized or used for other than internal documentary purposes without the written consent of an official representative of Barracuda Networks Inc. All specifications are subject to change without notice. Barracuda Networks Inc. assumes no responsibility for any inaccuracies in this document. Barracuda Networks Inc. reserves the right to change, modify, transfer, or otherwise revise this publication without notice.