

Now just to really cement that idea about the types of deployment, I have one of two modes. Incremental is the default. With incremental, resources defined in the template are deployed. If the resource already exists in my target, i.e. the resource group, and it matches the settings, the configuration in the template, it's left alone. If the settings are different, it is updated if it can. Resources that exist in that target, i.e. that resource group, but are not defined in a template, are ignored. So the first time I deployed that virtual network, it left the storage account alone. It wasn't defined in the template, but because it ran in incremental mode, it didn't touch it. In complete mode, it's the same as incremental in that if the resource already exists in the desired configuration, it's left alone. If it exists and it doesn't match, it's made to meet it. If it doesn't exist, it's created. But if there are things in the target, either resource group that are not defined in the template, they are removed. Complete is saying, make that target, i.e. the resource group, match the template. If there's stuff in the resource group that's not in the template, we need to remove it.

# **Azure Resource Manager deployment modes**

When deploying your resources, you specify that the deployment is either an incremental update or a complete update. The difference between these two modes is how Resource Manager handles existing resources in the resource group that aren't in the template.

For both modes, Resource Manager tries to create all resources specified in the template. If the resource already exists in the resource group and its settings are unchanged, no operation is taken for that resource. If you change the property values for a resource, the resource is updated with those new values. If you try to update the location or type of an existing resource, the deployment fails with an error. Instead, deploy a new resource with the location or type that you need.

**The default mode is incremental.**

## **Complete mode**

In complete mode, Resource Manager deletes resources that exist in the resource group but aren't specified in the template.

**Can I preview the changes that will happen before deploying a template?**

Yes, use the [what-if feature](https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/template-deploy-what-if). It evaluates the current state of your environment and compares it to the state that will exist after deployment. What-if shows you which resources will be created, deleted, or modified. Use what-if to avoid unintentionally deleting resources.

You can examine the summarized changes to make sure the template doesn't have any unexpected results

## **What-if commands**

### **Azure PowerShell**

To preview changes before deploying a template, use [New-AzResourceGroupDeployment](https://docs.microsoft.com/en-us/powershell/module/az.resources/new-azresourcegroupdeployment) or [New-AzSubscriptionDeployment](https://docs.microsoft.com/en-us/powershell/module/az.resources/new-azdeployment). Add the -Whatif switch parameter to the deployment command.

* New-AzResourceGroupDeployment -Whatif for resource group deployments
* New-AzSubscriptionDeployment -Whatif and New-AzDeployment -Whatif for subscription level deployments

You can use the -Confirm switch parameter to preview the changes and get prompted to continue with the deployment.

* New-AzResourceGroupDeployment -Confirm for resource group deployments
* New-AzSubscriptionDeployment -Confirm and New-AzDeployment -Confirm for subscription level deployments
* Be careful using complete mode with [copy loops](https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/copy-resources). Any resources that aren't specified in the template after resolving the copy loop are deleted.
* If you deploy to more than one resource group in a template, resources in the resource group specified in the deployment operation are eligible to be deleted. Resources in the secondary resource groups aren't deleted.
* There are some differences in how resource types handle complete mode deletions. Parent resources are automatically deleted when not in a template that's deployed in complete mode. Some child resources aren't automatically deleted when not in the template. However, these child resources are deleted if the parent resource is deleted. **For example**, if your resource group contains a DNS zone (Microsoft.Network/dnsZones resource type) and a CNAME record (Microsoft.Network/dnsZones/CNAME resource type), the DNS zone is the parent resource for the CNAME record. If you deploy with complete mode and don't include the DNS zone in your template, the DNS zone and the CNAME record are both deleted. If you include the DNS zone in your template but don't include the CNAME record, the CNAME isn't deleted.

For a list of how resource types handle deletion, see [Deletion of Azure resources for complete mode deployments](https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/complete-mode-deletion).

**If the resource group is** [**locked**](https://docs.microsoft.com/en-us/azure/azure-resource-manager/management/lock-resources)**, complete mode doesn't delete the resources.**

**Note:**

* Only root-level templates support the complete deployment mode. For [linked or nested templates](https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/linked-templates), you must use incremental mode.
* [Subscription level deployments](https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/deploy-to-subscription) don't support complete mode.
* Currently, the portal doesn't support complete mode.

## **Incremental mode**

In incremental mode, Resource Manager leaves unchanged resources that exist in the resource group but aren't specified in the template. Resources in the template are added to the resource group.

**Note**

When redeploying an existing resource in incremental mode, all properties are reapplied. The properties aren't incrementally added. A common misunderstanding is to think properties that aren't specified in the template are left unchanged. If you don't specify certain properties, Resource Manager interprets the deployment as overwriting those values. Properties that aren't included in the template are reset to the default values. Specify all non-default values for the resource, not just the ones you're updating. The resource definition in the template always contains the final state of the resource. It can't represent a partial update to an existing resource.

In rare cases, properties that you specify for a resource are actually implemented as a child resource. For example, when you provide site configuration values for a web app, those values are implemented in the child resource type Microsoft.Web/sites/config. If you redeploy the web app and specify an empty object for the site configuration values, the child resource isn't updated. However, if you provide new site configuration values, the child resource type is updated.

## **Example result**

To illustrate the difference between incremental and complete modes, consider the following scenario.

Resource Group contains:

* Resource A
* Resource B
* Resource C

Template contains:

* Resource A
* Resource B
* Resource D

When deployed in incremental mode, the resource group has:

* Resource A
* Resource B
* Resource C
* Resource D

When deployed in complete mode, Resource C is deleted. The resource group has:

* Resource A
* Resource B
* Resource D

## 

## 

## **Set deployment mode**

To set the deployment mode when deploying with PowerShell, use the Mode parameter.

Azure PowerShell

New-AzResourceGroupDeployment `

-Mode Complete `

-Name ExampleDeployment `

-ResourceGroupName ExampleResourceGroup `

-TemplateFile c:\MyTemplates\storage.json

The following example shows a linked template set to incremental deployment mode:

JSON

"resources": [

{

"type": "Microsoft.Resources/deployments",

"apiVersion": "2017-05-10",

"name": "linkedTemplate",

"properties": {

"mode": "Incremental",

<nested-template-or-external-template>

}

}

]

# **Rollback on error to successful deployment**

When a deployment fails, you can automatically redeploy an earlier, successful deployment from your deployment history. This functionality is useful if you've got a known good state for your infrastructure deployment and want to revert to this state. **There are a number of caveats and restrictions**:

* The redeployment is run exactly as it was run previously with the same parameters. You can't change the parameters.
* The previous deployment is run using the [complete mode](https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/deployment-modes#complete-mode). Any resources not included in the previous deployment are deleted, and any resource configurations are set to their previous state.
* The redeployment only affects the resources, any data changes aren't affected.
* You can use this feature only with resource group deployments, not subscription or management group level deployments.
* You can only use this option with root level deployments. Deployments from a nested template aren't available for redeployment.

To use this option, your deployments must have unique names so they can be identified in the history. If you don't have unique names, the current failed deployment might overwrite the previously successful deployment in the history.

## **PowerShell**

To redeploy the **last** successful deployment, add the -RollbackToLastDeployment parameter as a flag.

New-AzResourceGroupDeployment -Name ExampleDeployment02 `

-ResourceGroupName $resourceGroupName `

-TemplateFile c:\MyTemplates\azuredeploy.json `

-RollbackToLastDeployment

To redeploy a **specific** deployment, use the -RollBackDeploymentName parameter and provide the name of the deployment. The specified deployment must have succeeded.

New-AzResourceGroupDeployment -Name ExampleDeployment02 `

-ResourceGroupName $resourceGroupName `

-TemplateFile c:\MyTemplates\azuredeploy.json `

-RollBackDeploymentName ExampleDeployment01

# 

# **View deployment history with Azure Resource Manager**

Azure Resource Manager enables you to view your deployment history. You can examine specific operations in past deployments and see which resources were deployed. This history contains information about any errors.

The deployment history for a resource group is limited to 800 deployments. As you near the limit, deployments are automatically deleted from the history.

## **Get deployments and correlation ID**

You can view details about a deployment through the Azure portal, PowerShell, Azure CLI, or REST API. Each deployment has a correlation ID, which is used to track related events. If you [create an Azure support request](https://docs.microsoft.com/en-us/azure/azure-portal/supportability/how-to-create-azure-support-request), support may ask you for the correlation ID. Support uses the correlation ID to identify the operations for the failed deployment.

1. Select the resource group you want to examine.
2. Select the link under Deployments.
3. Select one of the deployments from the deployment history.
4. A summary of the deployment is displayed, including the correlation ID.

## **Get deployment operations and error message**

Each deployment can include multiple operations. To see more details about a deployment, view the deployment operations. When a deployment fails, the deployment operations include an error message.

1. On the summary for a deployment, select Operation details.
2. You see the details for that step of the deployment. When an error occurs, the details include the error message.