

# Curated Portal Policy Application Guide

Last Updated: 10/02/2024

## TABLE OF CONTENTS

<b>Technical Background .....</b>	<b>2</b>
<b>Implementing Curated Portal policy on a new tenant (Overview) .....</b>	<b>2</b>
Implementing Curated Portal Policy on a New Tenant.....	2
Assign "Owner" Role at "Tenant Root Group" Scope.....	2
Register Microsoft.PolicyInsights provider at Tenant Root Group.....	5
Create a Custom Azure Policy Definition & Policy Assignment.....	6
<b>Implementing Curated Portal policy on a new tenant (via Script) .....</b>	<b>9</b>
Create curated portal policy using Azure Cloud Shell.....	9
Create curated portal policy using Azure Cli .....	10
<b>Update existing Curated Portal Policy Assignment (Overview).....</b>	<b>11</b>
Update existing Curated Portal Policy Assignment (Manually) .....	11
Update existing Curated Portal Policy Assignment (Azure Cloud Shell).....	12
<b>Remove Owner Role Assignment from Tenant Root Group.....</b>	<b>12</b>

## Technical Background

Azure Policy is a service which measures configuration (and optionally mitigates deviations or blocks activity) on resources according to built-in or custom sets of rules called "policies". These policies often accept arguments called parameters, and the policies can be enforced at various scopes (levels) from the tenant, management groups, subscriptions, resource groups, down to individual resources.

For Curated Portal, Microsoft provides a custom policy and AllowList in the form of Javascript Object Notation (JSON) files, which are applied to the tenant to have the desired effect. This custom policy will block the provisioning of resources not included in the contract's Catalog.

As the contractually agreed upon Curated Catalog list of approved services grows over time, the implemented policy needs to be updated to include these new services. This document outlines the various methods to apply the Curated Portal policy to your tenants and keep them updated as the contract Catalog of services grows.

## Implementing Curated Portal policy on a new tenant (Overview)

Curated Portal policy can be implemented via two means:

1. Manually via a series of steps in the Azure Portal and execution of Azure CLI commands **OR**
2. Execution of a command line tool, available in the [Curated Portal GitHub Repository](#).

Both methods require Azure PowerShell and Azure CLI to be installed on the operating system used.

The key steps are:

1. Login to Azure Portal using Global Administrator account.
2. Elevate your account to "User Access Administrator" Role.
3. Create a management group called "Tenant Root Group" if it does not exist.
4. Grant your account "Owner" role assignment at "Tenant Root Group".
5. Register the Microsoft.PolicyInsights resource provider at "Tenant Root Group".
6. Create a custom Azure Policy Definition.
7. Create an Azure Policy Assignment with current parameters.
8. You may have to wait for up to 30 minutes for Policy changes to take effect.

## Implementing Curated Portal Policy on a New Tenant

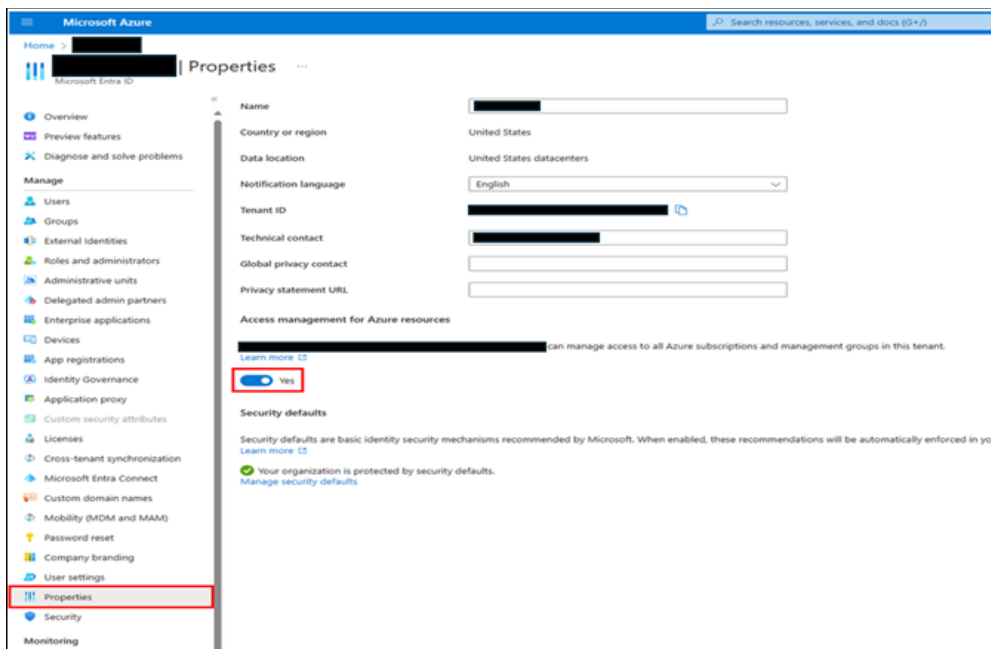
This section reviews all the steps needed to implement policy on a brand-new tenant.

### Assign "Owner" Role at "Tenant Root Group" Scope

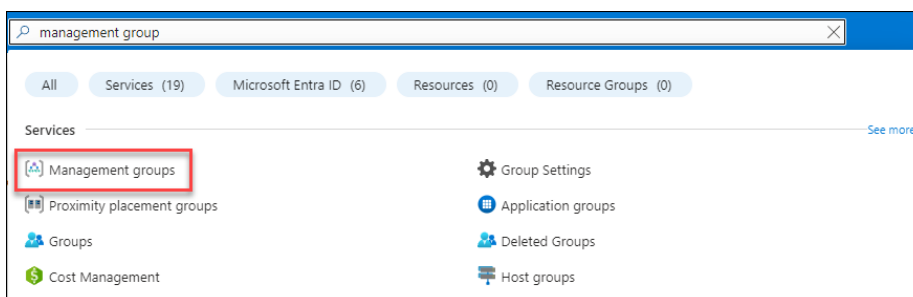
1. Log in to Azure Portal using "Global Administrator" account.
2. Configure tenant to apply Curated Portal utilizing the **Global Administrator** role. *Note – These steps only need to be completed on initial setup.*

- Go to **Microsoft Entra ID** → **Properties** and then toggle to **Yes** under **Access management for Azure Resources**.

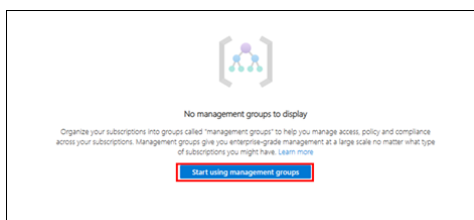
**Note** – Ensure to toggle Access Management back to **NO** once Role assignment is completed.



- Next, Search for **Management groups** in the search bar.



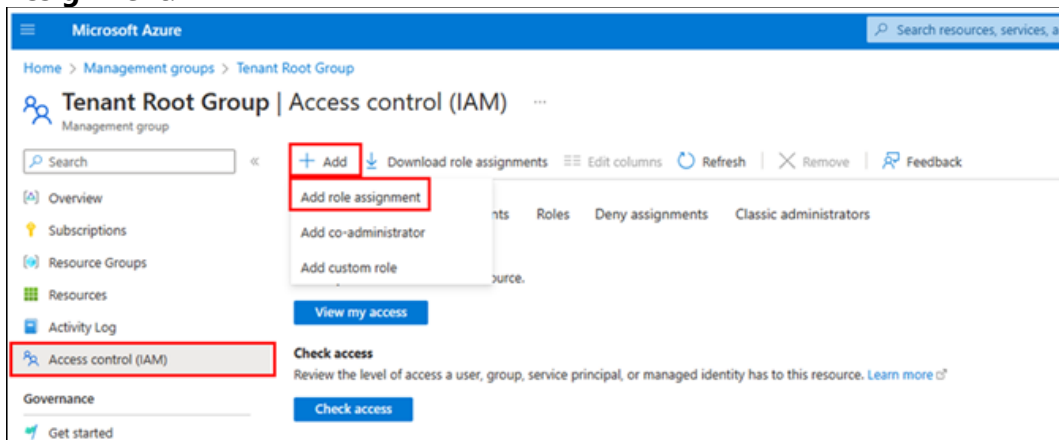
- If this is an initial tenant set up, you may see the message below. Click on **Start Using Management Groups**.



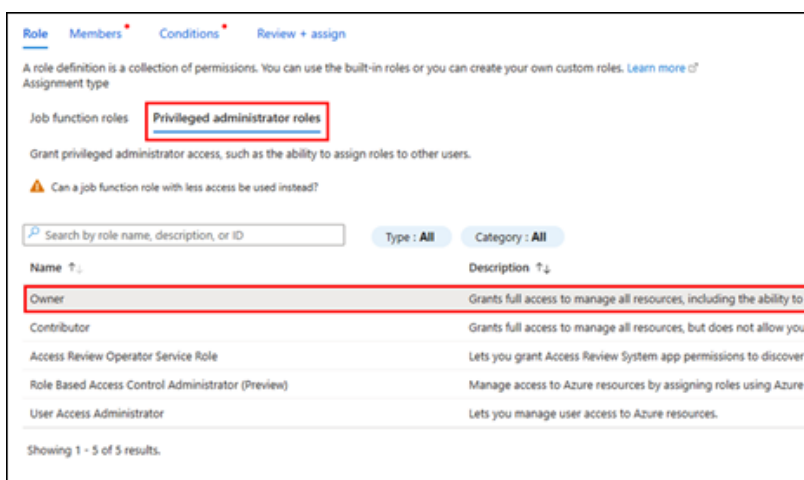
- Select **Tenant Root Group**.

Name	Type	ID
✓ [Icon] Tenant Root Group	...	Management group [Redacted]
> [Icon] 3 subscriptions		
> [Icon] [Redacted]	...	Management group [Redacted]
> [Icon] [Redacted]	...	Management group [Redacted]

7. Select the **Access Control (IAM)** blade on the left side then select **Add** and **Add Role Assignment**.



8. Choose the **Privileged Administrator Roles** tab and select the **Owner** role. Click **Next**.



9. Choose **Select Members** and choose your user account from the right hand drop down. Use the **Select** button to confirm user selected and select **Next**.



10. Select **Allow user to assign all role** and select **Review and Assign**.

11. Confirm role assignment by navigating to the **Role Assignments** Tab.

Name	Type	Role	Scope	Condition
Owner (2)				
[User]	User	Owner	Root (inherited)	None
[Group]	Group	Owner	Root (inherited)	None
Reader (1)				
[Group]	Group	Reader	Root (inherited)	None
User Access Administrator (1)				
[User]	User	User Access Administrator	Root (inherited)	None

## Register Microsoft.PolicyInsights provider at Tenant Root Group

Resource provider Microsoft.PolicyInsights can be registered using one of the two methods. Use one of these methods to register provider.

- Register Microsoft.PolicyInsights provider at Tenant Root Group using Azure CLI **OR**
- Register Microsoft.PolicyInsights provider at Tenant Root Group using Azure Cloud Shell

### Register Microsoft.PolicyInsights provider at Tenant Root Group Using Azure CLI

To run the following command via Azure CLI, make sure that you are logged in with Azure credentials:

1. Launch Azure Cli on your local machine.
2. To login using your user account to Azure, run the following command.

#### az login

3. To register Microsoft.PolicyInsights resource provider at Tenant Root Group, run following 3 commands.

```
$aadTenantId = az account list --query "[?isDefault].tenantId" -o tsv
echo $aadTenantId
az provider register --namespace "Microsoft.PolicyInsights" --management-group-id $aadTenantId --debug
```

### Register Microsoft.PolicyInsights provider at Tenant Root Group Using Azure Cloud Shell

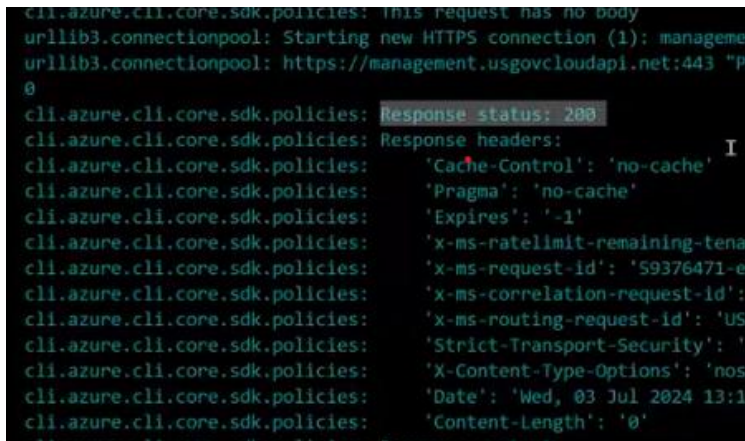
1. If you have never configured Azure Cloud Shell, then follow instructions in one of the links below. If you already have Azure Cloud Shell configured, then skip to next step.

- Classic UI - [Get started with Azure Cloud Shell | Microsoft Learn](#)
- New UI - [Get started with Azure Cloud Shell ephemeral sessions | Microsoft Learn](#)

2. Launch **Azure Cloud Shell** in **Bash** mode and then run following 3 commands.

```
aadTenantId=$(az account list --query "[?isDefault].tenantId" -o tsv)
echo $aadTenantId
az provider register --namespace "Microsoft.PolicyInsights" --management-group-id $aadTenantId --debug
```

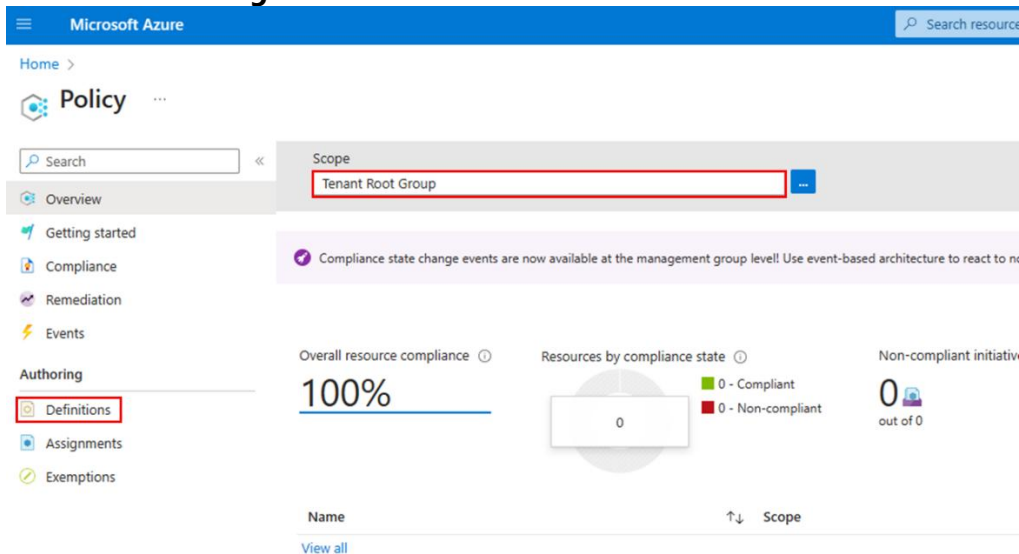
3. Verify successful registration with a **Response status: 200** in the output.



```
cli.azure.cli.core.sdk.policies: This request has no body
urllib3.connectionpool: Starting new HTTPS connection (1): management.usgovcloudapi.net:443
urllib3.connectionpool: https://management.usgovcloudapi.net:443 "POST /providers/Microsoft.PolicyInsights/registration?api-version=2023-07-01 HTTP/1.1" 200
cli.azure.cli.core.sdk.policies: Response status: 200
cli.azure.cli.core.sdk.policies: Response headers:
cli.azure.cli.core.sdk.policies: 'Cache-Control': 'no-cache'
cli.azure.cli.core.sdk.policies: 'Pragma': 'no-cache'
cli.azure.cli.core.sdk.policies: 'Expires': '-1'
cli.azure.cli.core.sdk.policies: 'x-ms-ratelimit-remaining-tenant': '1'
cli.azure.cli.core.sdk.policies: 'x-ms-request-id': '59376471-e...'
cli.azure.cli.core.sdk.policies: 'x-ms-correlation-request-id': 'US...'
cli.azure.cli.core.sdk.policies: 'x-ms-routing-request-id': 'US...'
cli.azure.cli.core.sdk.policies: 'Strict-Transport-Security': 'max-age=31536000; includeSubDomains'
cli.azure.cli.core.sdk.policies: 'X-Content-Type-Options': 'nosniff'
cli.azure.cli.core.sdk.policies: 'Date': 'Wed, 03 Jul 2024 13:13:10 GMT'
cli.azure.cli.core.sdk.policies: 'Content-Length': '0'
```

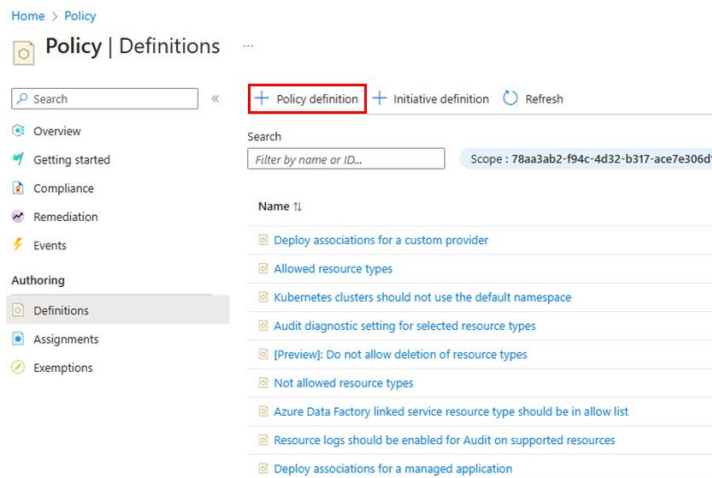
## Create a Custom Azure Policy Definition & Policy Assignment

1. Search for **Policy** in the search bar. Set **Scope** to **Tenant Root Group** and select **Definitions** from the **Authoring** section.



The screenshot shows the Microsoft Azure Policy portal interface. At the top, there's a search bar with 'Policy' entered. Below the search bar, the 'Scope' is set to 'Tenant Root Group'. The left sidebar shows the 'Authoring' section with 'Definitions' selected. The main content area displays a '100%' overall resource compliance bar chart. Below this, there's a 'Resources by compliance state' section showing 0 compliant and 0 non-compliant resources. A 'Non-compliant initiatives' section shows 0 out of 0. At the bottom, there's a table with columns for 'Name' and 'Scope'.

2. Select **+ Policy Definition** to create new policy definition.



- Set **Definition location** to **Tenant Root Group** and provide appropriate **Name** for the policy. For example - **"JWCC Curated Portal Policy"**

**Policy definition** ...

New Policy definition

**BASICS**

Definition location \*

Tenant Root Group ✓

Name \* ⓘ

JWCC Curated Portal Policy ✓

- Copy code from [Allowed resource types.json](#) file from Curated Portal GitHub repository and **replace** the existing code under the **POLICY RULE** section. Click **Save**.

Category ⓘ

☒ Create new ☐ Use existing

Category

**POLICY RULE**

Import sample policy definition from GitHub

Learn more about policy definition structure

```

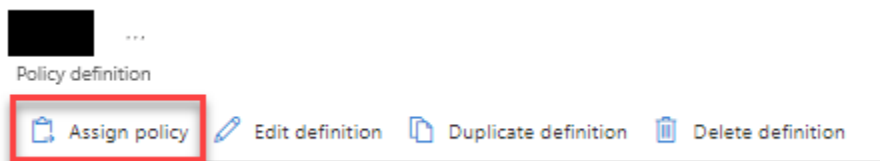
1 {
2   "properties": {
3     "displayName": "Allowed resource types",
4     "policyType": "Builtin",
5     "mode": "All",
6     "description": "This policy enables you to specify the resource types that your organization can deploy. Only resource types that support",
7     "metadata": {
8       "version": "1.0.0",
9       "category": "General"
10    },
11    "parameters": {
12      "listOfResourceTypesAllowed": {
13        "type": "Array",
14        "metadata": {
15          "description": "The list of resource types that can be deployed.",
16          "displayName": "Allowed resource types"
17        }
18      }
19    },
20    "policyRule": {
21      "if": {
22        "not": {
23          "field": "type",
24          "in": "[parameters('listOfResourceTypesAllowed')]"
25        }
26      },
27      "then": {
28        "effect": "deny"
29      }
30    }
31  },
32  "id": "/providers/Microsoft.Authorization/policyDefinitions/a08ec900-254a-4555-9bfs-e42af0405c5c",
33  "type": "Microsoft.Authorization/policyDefinitions",
34  "name": "a08ec900-254a-4555-9bfs-e42af0405c5c"
35 }

```

Save Cancel

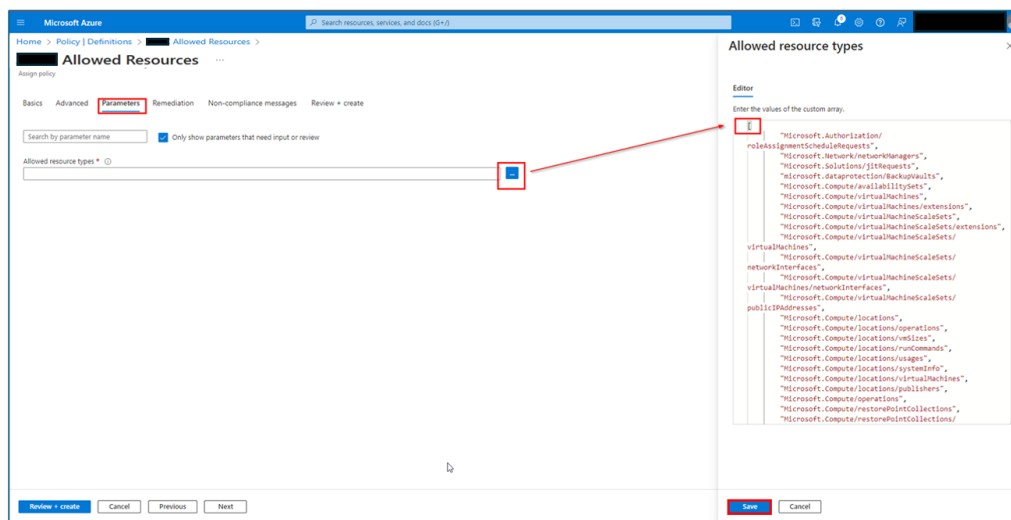
- Click on **Assign Policy** to continue with policy assignment.



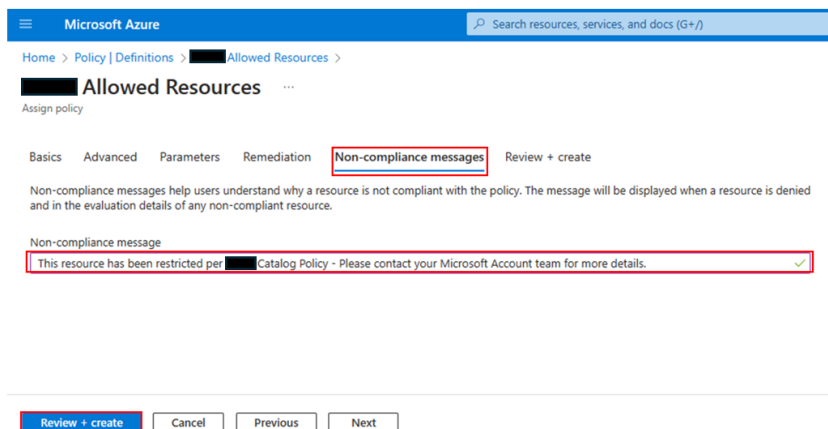


6. On **Basic** tab, make sure that **Scope** is set to **Tenant Root Group**. If your environment contains a mixture of JWCC and non-JWCC subscriptions, then make sure to exclude non-JWCC subscriptions on **Exclusions** tab. Click **Next**.
7. On **Parameters** tab, click on ellipsis to edit that **Allowed resource types** parameter. Parameter file is located in the [Curated Portal GitHub Repository](#). Copy all the Resource Providers (RPs) from the JSON file and paste them within **Editor** and click **Save**.

Parameter file name format - **JWCC\_Gov\_Final\_<Month>\_<Year>\_<Version>.json**  
**e.g.** JWCC\_Gov\_Final\_June\_2024\_4.0.json



8. Skip **Remediation** tab and go to the **Non-Compliance Messages** tab. Provide the recommended message: ***This resource has been restricted per JWCC Catalog Policy – Please contact your Microsoft Account team for more details.*** Click **Review + Create**.





# Implementing Curated Portal policy on a new tenant (via Script)

A script and relevant files are available in the [Curated Portal GitHub repository \(Scripts folder\)](#). You will find the following files at this location.

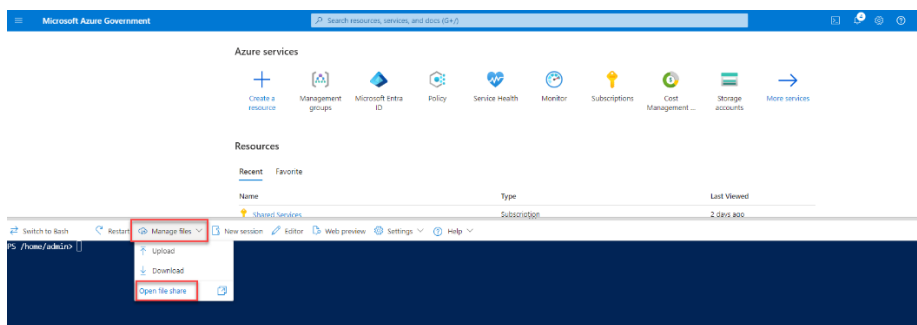
Name	Notes
Curated_Portal_PolicyDefinition.json	Contains <b>Azure Policy Definition</b> to be used for Curated Portal Policy
JWCC_Gov_Final_<Month>_<Year>_<Version>.json	Contains <b>latest JWCC Catalog</b> to be used as parameter value for Policy Assignment
Parameters.json	Contains <b>Azure Policy Parameters</b> to be used for Curated Portal Policy
curatedcatalog.ps1	PowerShell Script to create Curated Portal Policy Definition/Assignment

**Note** - Name of the file `JWCC\_Gov\_Final\_<Month>\_<Year>\_<Version>.json` will change as JWCC Catalog gets updated over time. (e.g. JWCC\_Gov\_Final\_June\_2024\_4.0.json)

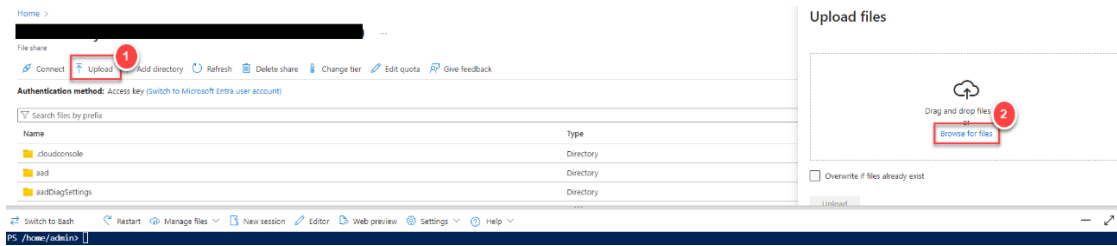
Descriptor	Notes
<Month>	Indicates Month (e.g. `June`)
<Year>	Indicates Year (e.g. `2024`)
<Version>	Indicates Version (e.g. `4.0`)

## Create curated portal policy using Azure Cloud Shell

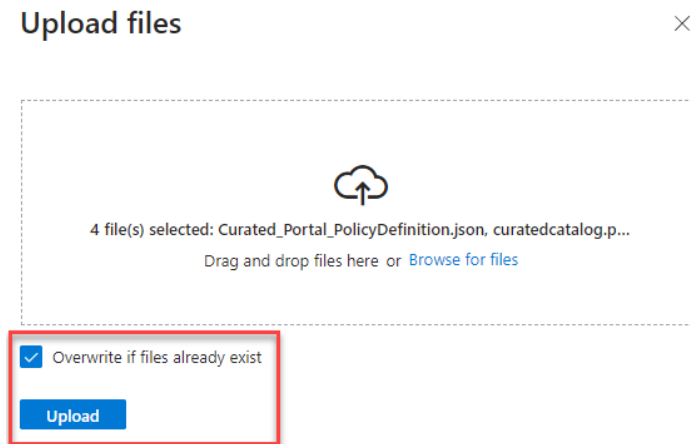
- Download following files from [Curated Portal GitHub repository \(Scripts folder\)](#) and store locally on your machine.
  - Curated\_Portal\_PolicyDefinition.json
  - JWCC\_Gov\_Final\_<Month>\_<Year>\_<Version>.json
  - Parameters.json
  - curatedcatalog.ps1
- Log in to **Azure Portal** using **Global Administrator** user account.
- Make sure this account has **Owner** role assignment at **Tenant Root Group** scope.
- Launch **Azure Cloud Shell** in **PowerShell** mode.
- Select **Manage files** and then select **Open file share**.



- Click on **Upload** and then click on **Browse for files** on **Upload files** page.



7. Select all the files downloaded in step 1 and click **Open**.
8. Select **Overwrite if files already exist** and then click on **Upload**. Now you should have all these files in your **clouddrive**.



9. In **Azure Cloud Shell**, type **cd clouddrive**.
10. Run following command to run the script to create Azure Policy Definition, Register Microsoft.PolicyInsights resource provider and Policy Assignment.

**./curatedcatalog.ps1 -folderPath <folderPath> -azureCloudName <azureCloudName> -allowListFileName <allowListFileName>**

**Note** - Before running the command replace the following placeholders with appropriate values.

Placeholder Name	Comment
<folderPath>	Provide folder path where files are uploaded in <b>clouddrive</b> (e.g. <b>/home/admin/clouddrive/</b> )
<azureCloudName>	Provide value for Cloud Name (Pick one of these options relevant to your cloud name - <b>AzureUSGovernment, AzureCloud, ussec, usnat</b> )
<allowListFileName>	Provide file name containing allow list (e.g. <b>JWCC_Gov_Final_June_2024_4.0.json</b> )

11. Successful run of above command will create Policy Definition and Policy Assignment.

## Create curated portal policy using Azure Cli

1. Download following files from [Curated Portal GitHub repository \(Scripts folder\)](#) and store locally on your machine.
  - Curated\_Portal\_PolicyDefinition.json
  - JWCC\_Gov\_Final\_<Month>\_<Year>\_<Version>.json

- Parameters.json
  - curatedcatalog.ps1
2. Launch **Azure Cli** on your local machine.
  3. Log in to Azure using **az login** command with **Global Administrator** user account.
  4. Make sure this account has **Owner** role assignment at **Tenant Root Group** scope.
  5. Run following command to run the script to create Azure Policy Definition, Register Microsoft.PolicyInsights resource provider and Policy Assignment.

**.\curatedcatalog.ps1 -folderPath <folderPath> -azureCloudName <azureCloudName> -allowListFileName <allowListFileName>**

**Note** - Before running the command replace the following placeholders with appropriate values.

Placeholder Name	Comment
<folderPath>	Provide folder path where files are downloaded on your local machine (e.g. <b>C:\Temp\Scripts\</b> )
<azureCloudName>	Provide value for Cloud Name (Pick one of these options relevant to your cloud name - <b>AzureUSGovernment, AzureCloud, ussec, usnat</b> )
<allowListFileName>	Provide file name containing allow list (e.g. <b>JWCC_Gov_Final_June_2024_4.0.json</b> )


6. Successful run of above command will create Policy Definition and Policy Assignment.

## Update existing Curated Portal Policy Assignment (Overview)

Updating an existing Curated Portal policy can be done via two methods:

1. Paste replacement parameters values into the assignment parameters in the Azure Portal
2. Using Azure Cloud Shell

## Update existing Curated Portal Policy Assignment (Manually)

1. Login to the Azure Portal using **Global Administrator** user account.
1. Search and select **Policy** in the search bar.
2. Select **Assignments** from the **Authoring** section.
3. Under **Assignments**, select the existing policy assignment relating to Curated Portal policy at the **Tenant Root Group**. Record the assignment name.
4. Select **Edit Assignment**.
5. On the **Basics** section and review the assignment metadata and update the "Assigned by", "Assignment description" for versioning. Keep the same **Assignment Name**.
6. Select **Parameters** and then click the ellipsis  to open the editor on the right side of the page.
7. Select All (**Ctrl+A**) content in the editor window. Copy (**Ctrl+C**) all the content from the editor and paste (**Ctrl+V**) to an empty text editor. **Save** the copied parameter values as a backup in case you need to revert later, due to an error.
8. **Delete** the web editor content to have a blank space to enter the updated parameters.

- Open the updated parameter file located in the\_ Copy all the Resource Providers (RPs) from the JSON file and paste them within **Editor** and click **Save**.

Parameter file name format - **JWCC\_Gov\_Final\_<Month>\_<Year>\_<Version>.json**  
e.g. JWCC\_Gov\_Final\_June\_2024\_4.0.json

- Click **Review + Save**.

## Update existing Curated Portal Policy Assignment (Azure Cloud Shell)

Before you can update the existing Policy Assignment, you need to have the following two values available.

- Policy Assignment Name
- resourceId of policy assignment scope

- Log in to **Azure Portal** using **Global Administrator** user account.
- Make sure this account has **Owner** role assignment at **Tenant Root Group** scope.
- Go to **Policy** and select **Assignments**. Make sure **Scope** is selected for the **Tenant Root Group**.
- Select **Assignment Name** of the policy assignment and copy the **Name**.
- Launch **Azure Cloud Shell** in **PowerShell** mode.
- Run following command to retrieve & store **resourceId** of the **Tenant Root Group**.

```
$tenantid = "$(az account list --only-show-errors --query "[?isDefault].homeTenantId" -o tsv)"
$mgmtgroups = az account management-group list --query "[?contains(name, '$tenantid')]" -o jsonc
$mgmtgroupid = ($mgmtgroups | convertfrom-json).id
```

- Run following command to update existing Policy Assignment with updated parameter values.  
az policy assignment update --name <assignmentName> --scope \$mgmtgroupid --params <allowListFileName>

**Note** - Before running the command replace the following placeholders with appropriate values.

Placeholder Name	Comment
<assignmentName>	Provide Policy Assignment name retrieved in step 4
<allowListFileName>	Provide file name containing allow list (e.g. <b>JWCC_Gov_Final_June_2024_4.0.json</b> )

## Remove Owner Role Assignment from Tenant Root Group

After verification of registration, remove ownership over the Tenant Root Management Group to reduce privilege footprint.

- Search for **Management groups** in the search bar.
- Go to **Access Control (IAM) → Role assignments**.
- Select your user account and then **Remove** and click **Yes** to remove the selected role assignment.







+ Add   Download role assignments   Edit columns   Refresh   **Remove**   Feedback

Check access   **Role assignments**   Roles   Deny assignments   Classic administrators

All   Job function (0)   Privileged (12)

Search by name or email   Type : All   Role : All   Scope : All scopes   Group by : Role

12 items (12 Users)

Name		Type	Role
Owner (6)			
<input type="checkbox"/>	 [Redacted]	User	Owner ⓘ
<input type="checkbox"/>	 [Redacted]	User	Owner ⓘ
<input type="checkbox"/>	 [Redacted]	User	Owner ⓘ
<input type="checkbox"/>	 [Redacted]	User	Owner ⓘ
<input checked="" type="checkbox"/>	 Tim Zimmerman	User	Owner ⓘ
<input type="checkbox"/>	 [Redacted]	User	Owner ⓘ
> User Access Administrator (6)			