George Smith

IST 615

Azure Lab 2

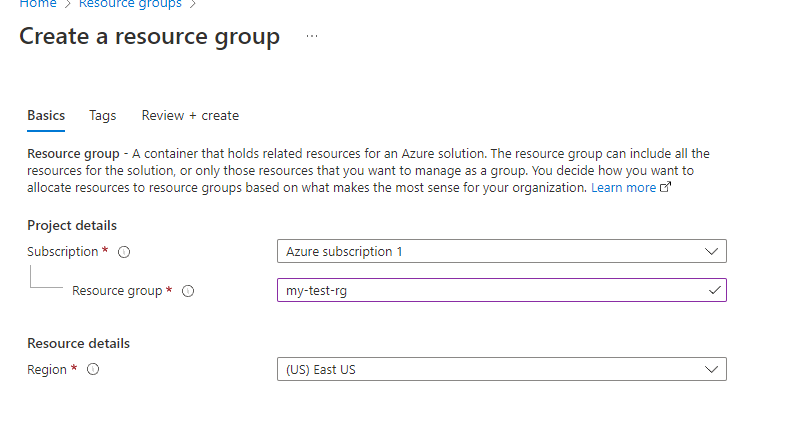
**Exercise – Protect a storage account from accidental deletion by using a resource lock**

In this lab I learned to manage resource locks from the Azure portal, PowerShell, the Azure CLI, or from an Azure Resource Manager template. I learned that I can apply locks to a subscription, a resource group, or an individual resource. The lock level can be set to CanNotDelete or ReadOnly. CanNotDelete means authorized people can still read and modify a resource, but they can't delete the resource without first removing the lock. ReadOnly means authorized people can read a resource, but they can't delete or change the resource. Applying this lock is like restricting all authorized users to the permissions granted by the Reader role in Azure RBAC. Although locking helps prevent accidental changes, you can still make changes by following a two-step process. Resource locks apply regardless of RBAC permissions. Even if you're an owner of the resource, you must still remove the lock before you can perform the blocked activity.

**Create the resource group**

**Purpose:** AResource group is a container that holds related resources for an Azure solution. The resource group can include all the resources for the solution, or only those resources that you want to manage as a group. You decide how you want to allocate resources to resource groups based on what makes the most sense for your organization.

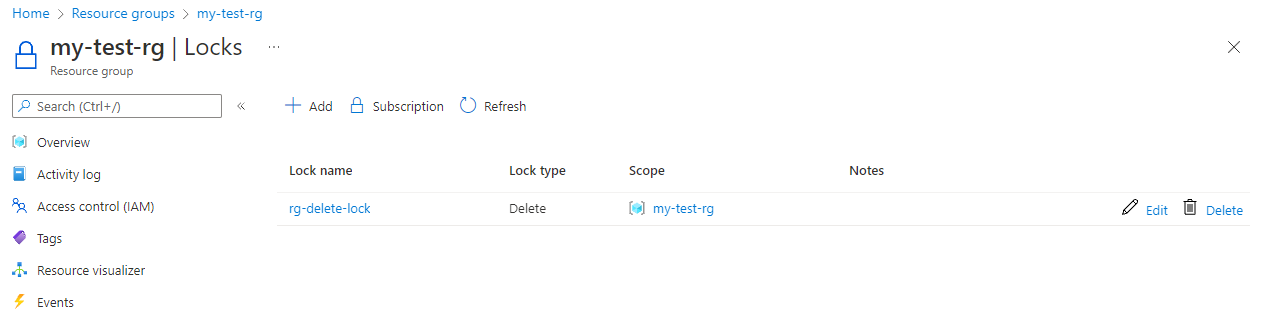
**What I did:** In this step I accessed the Azure portal and signed into my account. I found the resource group tab and created a new resource group using the details defined in the lab (seen below). I then created the resource group.



**Add a lock to the resource group**

**Purpose:** A resource lock prevents resources from being accidentally deleted or changed. Even with Azure role-based access control (Azure RBAC) policies in place, there's still a risk that people with the right level of access could delete critical cloud resources. A resource lock acts as a warning system that reminds you that a resource should not be deleted or changed.

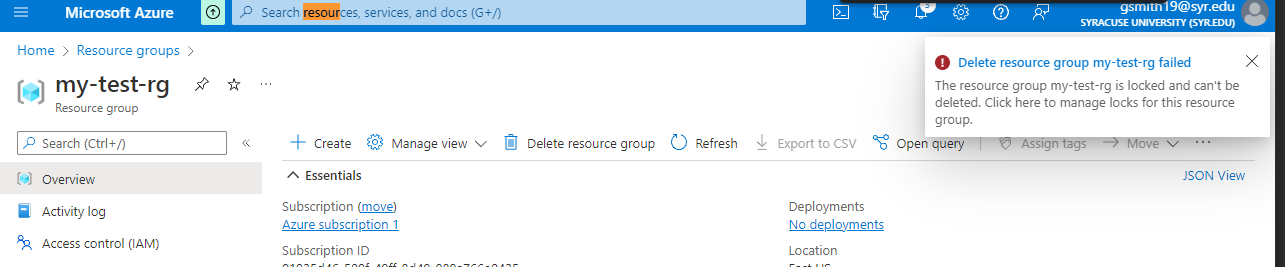
**What I did:** I accessed my recently created resource group, selected locks and added a new resource lock.



**Verify that the resource group is protected from deletion**

**Purpose:** Verify protection by attempting to delete the resource group

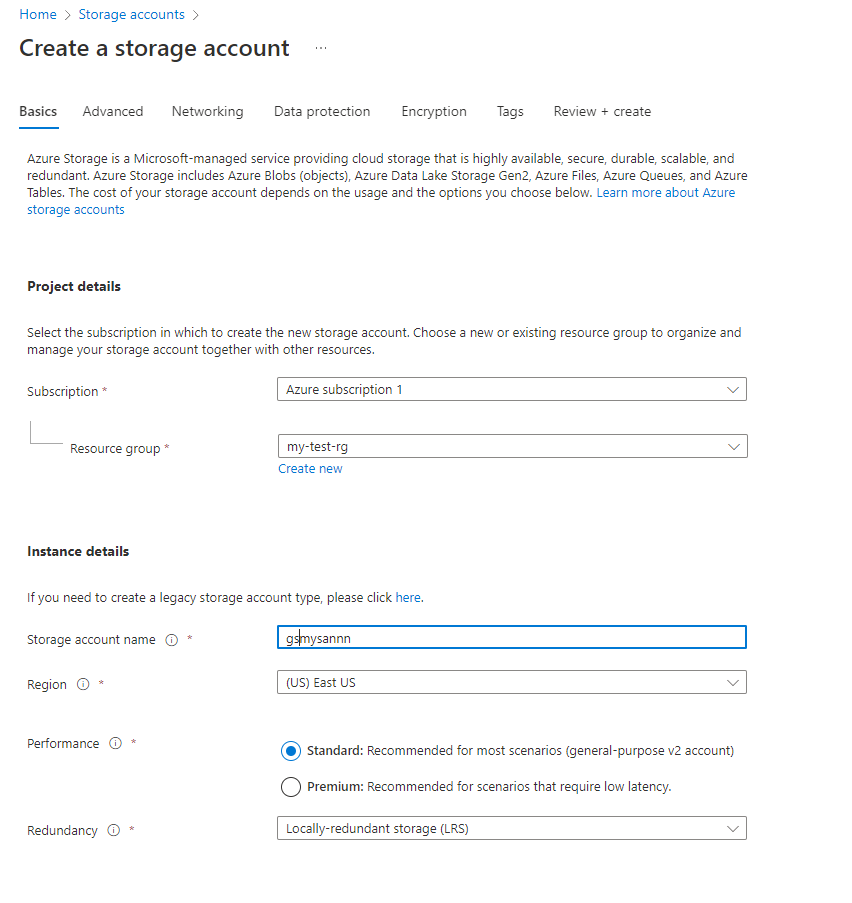
**What I did:** I attempted to delete the resource group by was met with an error message not allowed me to delete the resource group as it is locked. This was the expected outcome**.**

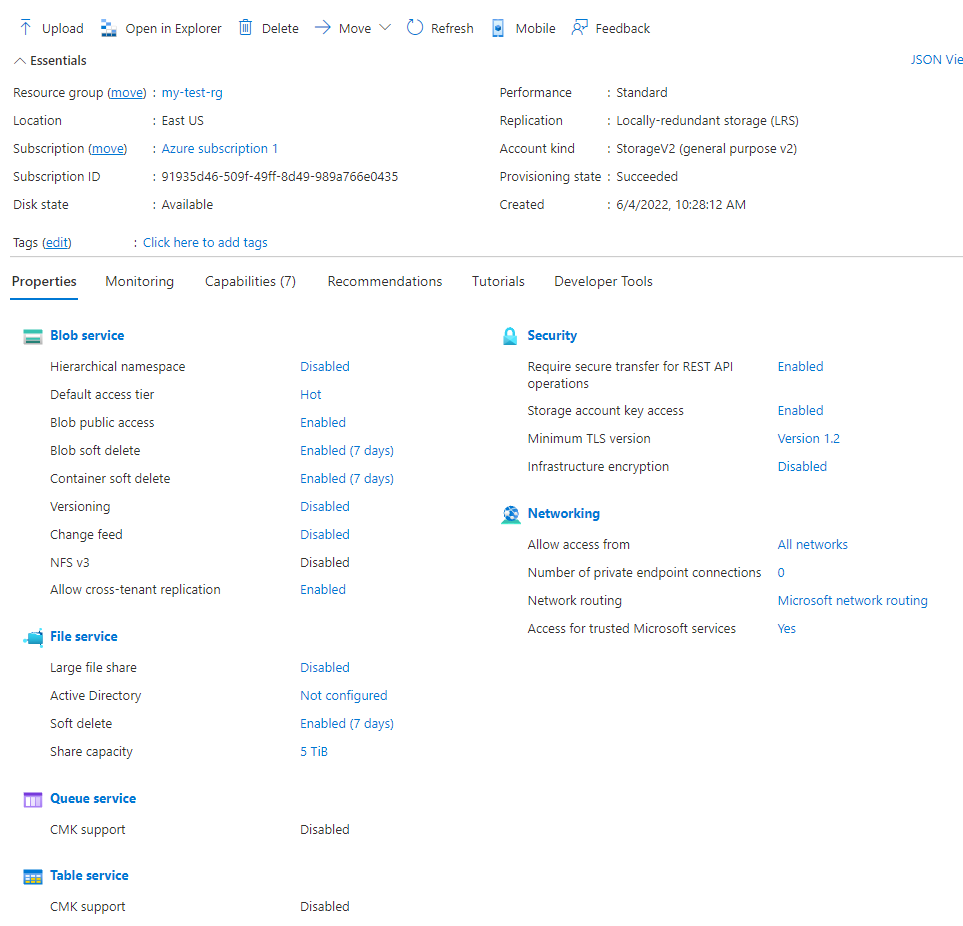


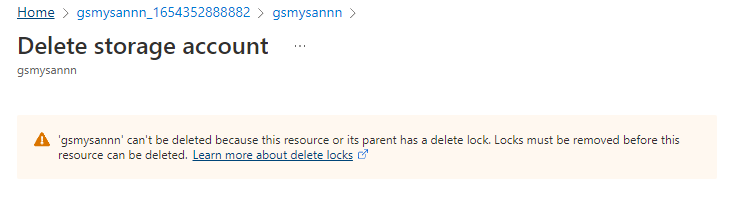
**Protect a storage account from accidental deletion**

**Purpose:** Add a storage account to my resource group and see how the lock from the parent resource group prevents the storage account from being deleted

**What I did:** From the Azure portal I selected storage accounts and created a new storage accounts using the details from the lab (seen below). I then attempted to delete the resource and found that the lock created for the parent resource group was inherited to the storage account. As a result, I was unable to delete the storage account.



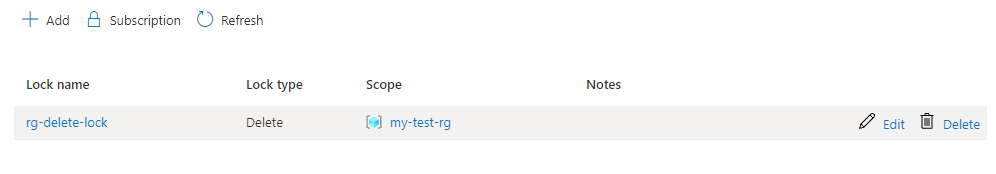




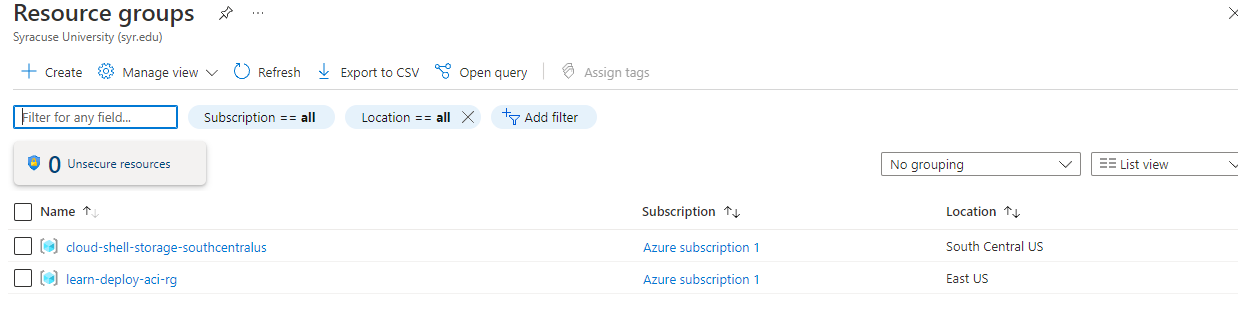
**Delete the resource group and the storage account**

**Purpose:** I no longer need the resource group or storage account so I removed both. To do this I first need to remove resource lock

**What I did:** I went back to my resource group and selected locks under settings. I located my lock and deleted it. I then deleted my resource group. I then confirmed that my resource group was deleted



Resource group created in exercise no longer exists



**Restrict deployments to a specific location by using Azure Policy**

This exercise allowed me to create a policy in Azure policy that restricts the deployment of Azure resource to a specific location. Azure Policy allowed me to define both individual policies and groups of related policies, known as initiatives. Azure Policy evaluates your resources and highlights resources that aren't compliant with the policies created. Azure Policy can also prevent noncompliant resources from being created. Azure Policy comes with built-in policy and initiative definitions for Storage, Networking, Compute, Security Center, and Monitoring. Additional, Azure Policy also evaluates and monitors all current VMs in my environment. Azure Policy can automatically remediate noncompliant resources and configurations to ensure the integrity of the state of the resources. Azure Policy also integrates with Azure DevOps by applying any continuous integration and delivery pipeline policies that pertain to the pre-deployment and post-deployment phases of your applications.

**Create the resource Group**

**Create the resource group**

**Purpose:** AResource group is a container that holds related resources for an Azure solution. The resource group can include all the resources for the solution, or only those resources that you want to manage as a group. You decide how you want to allocate resources to resource groups based on what makes the most sense for your organization.

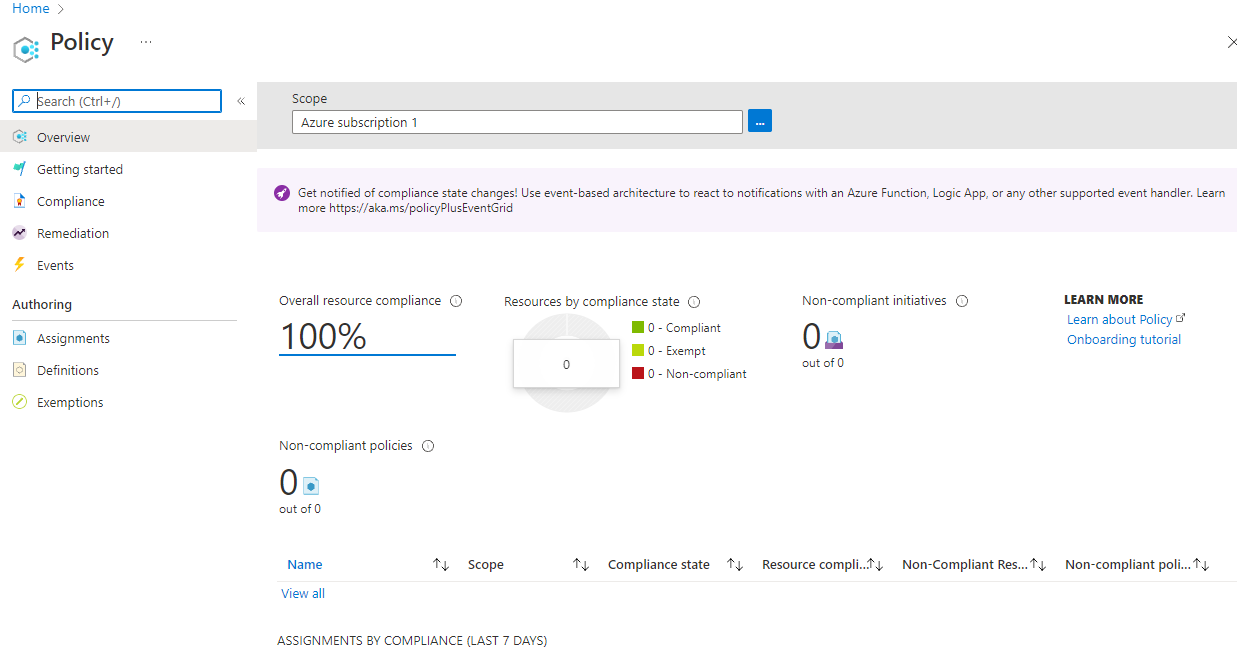
**What I did:** In this step I accessed the Azure portal and signed into my account. I found the resource group tab and created a new resource group using the details defined in the lab (seen below). I then created the resource group.

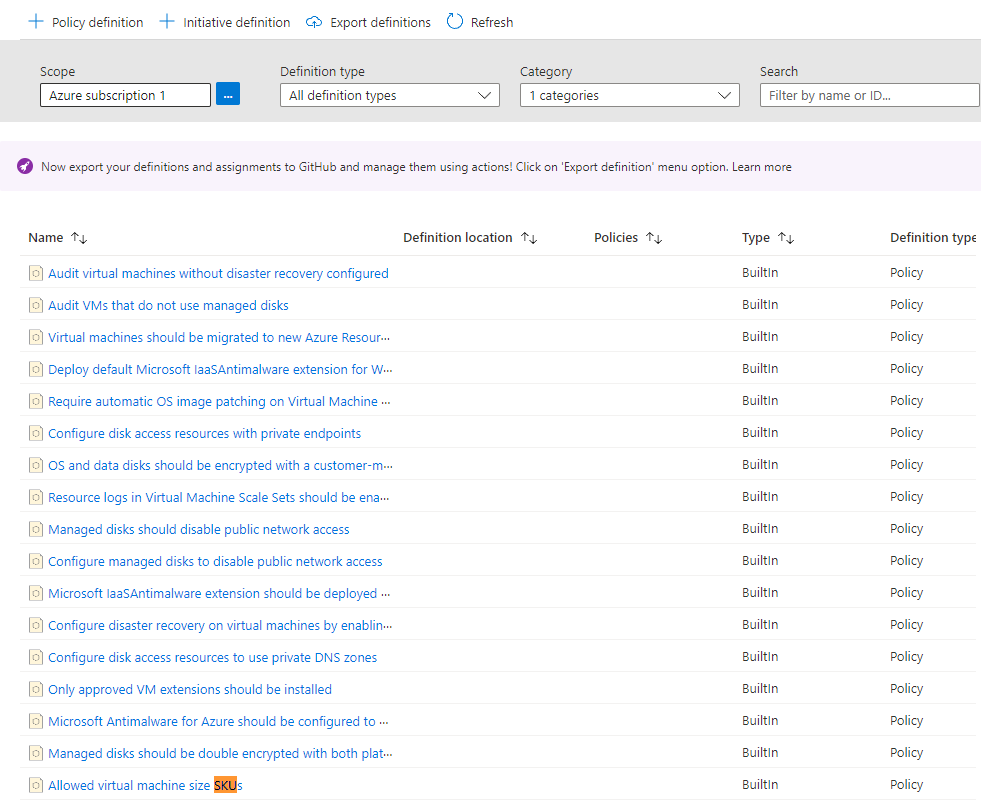


**Explore predefined policies**

**Purpose:** Before configuring the location policy I took a look at some predefined policies.

**What I did:** I entered policy at the top of the search bar and selected policy from the list of results. Under authoring I selected definitions. From the category drop down list I selected only compute and viewed the policies.

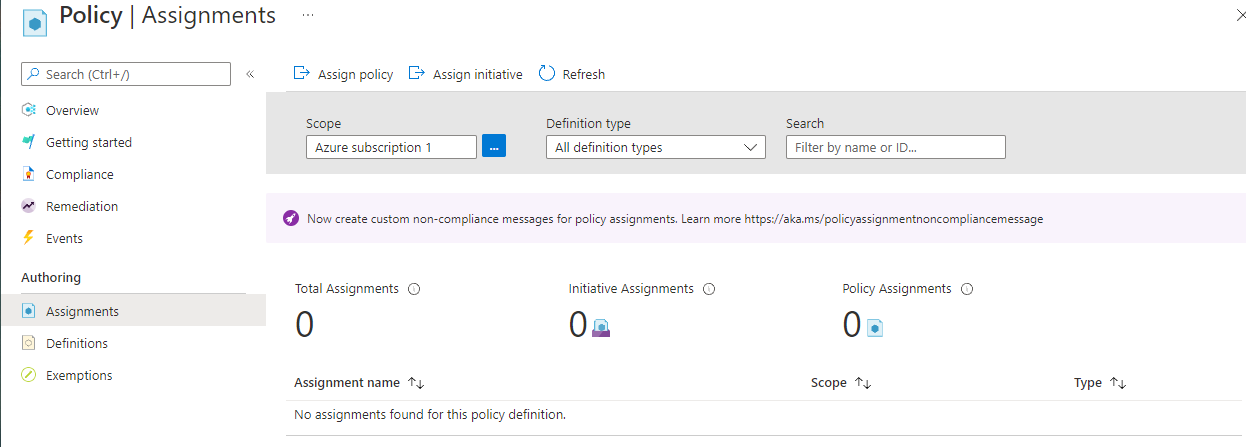


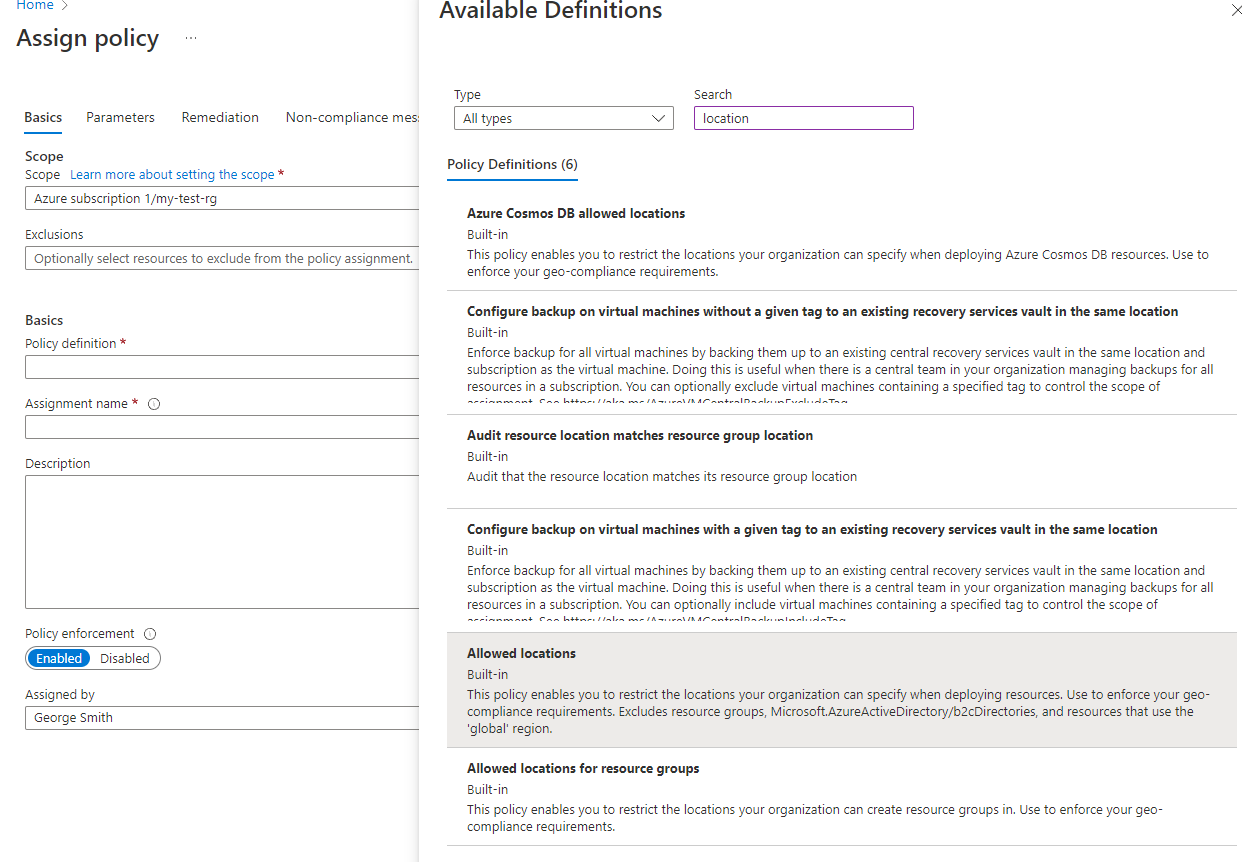


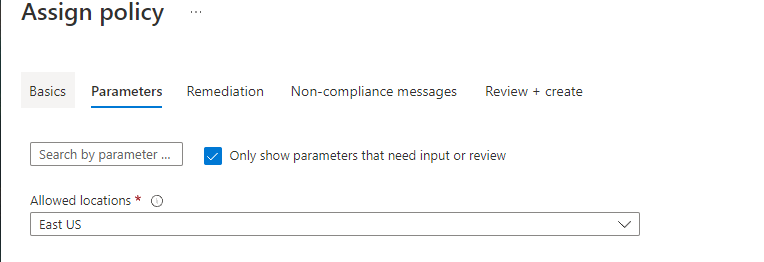
**Configure the location policy**

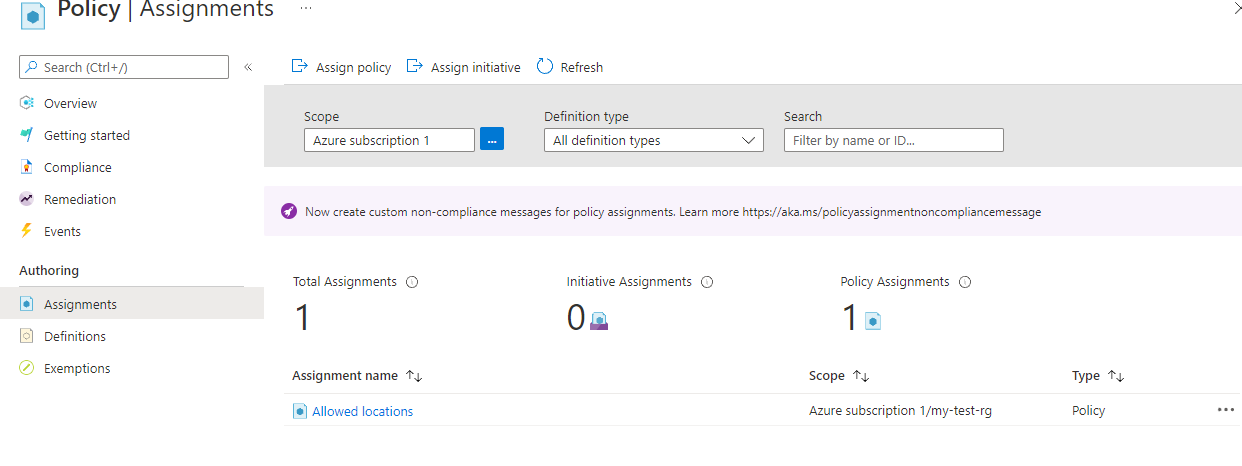
**Purpose:** Configure the allowed location policy by using Azure Policy

**What I did:** I went to the policy pane and under authoring selected assignments. I assigned a policy. Under scope I selected the ellipsis and set the subscription field and resource group field. I then selected the allowed locations. As US East. I then created the allowed location policy and verified that it worked.





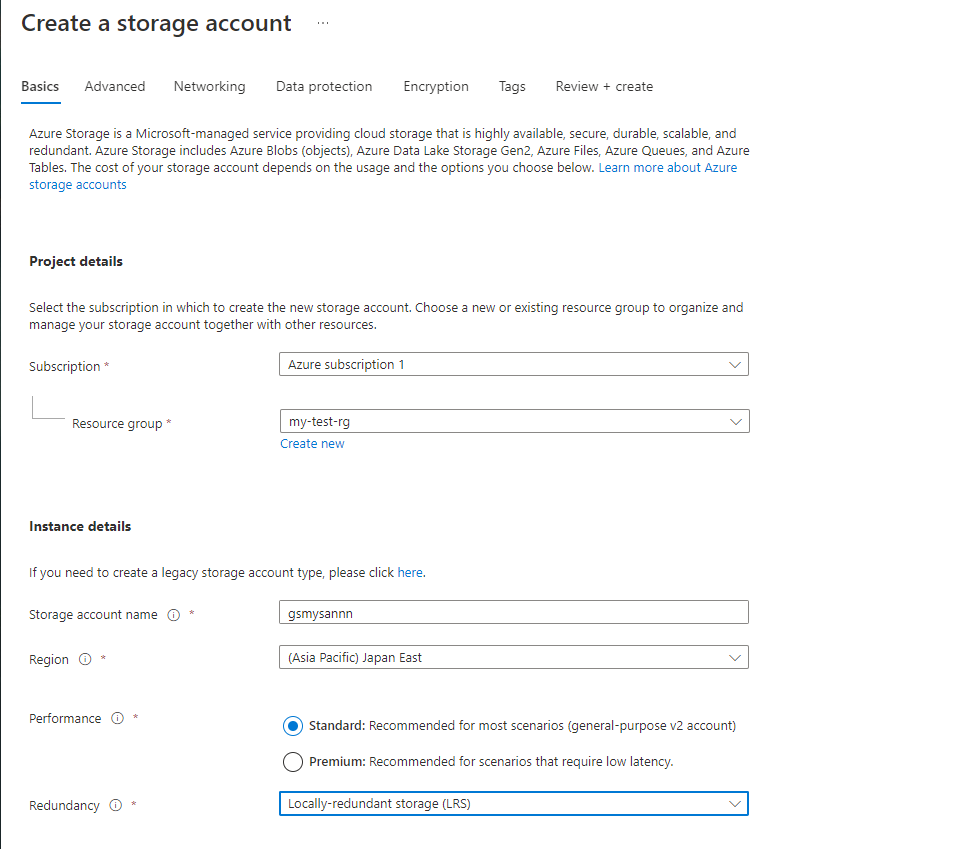


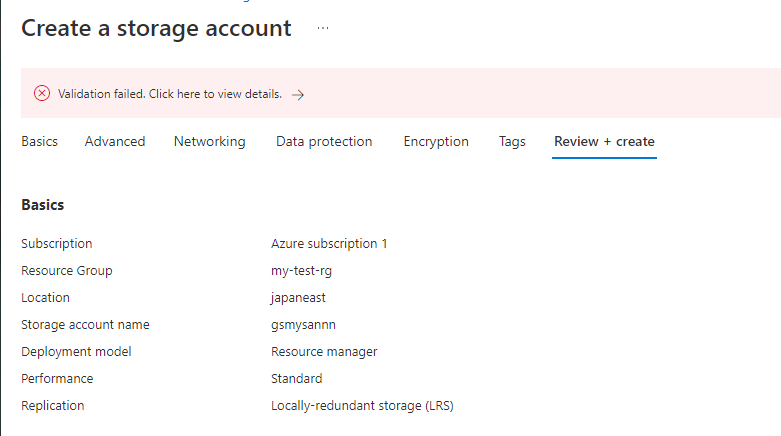


**Verify the location policy**

**Purpose:** To attempt to add a storage account to my resource group at a location that violates my location policy

**What I did:** Accessed the storage account window from the search bar. Crated a storage account using the setting defined in the lab (detailed below). Created the storage account, and found that the deployment failed because of the policy violation.

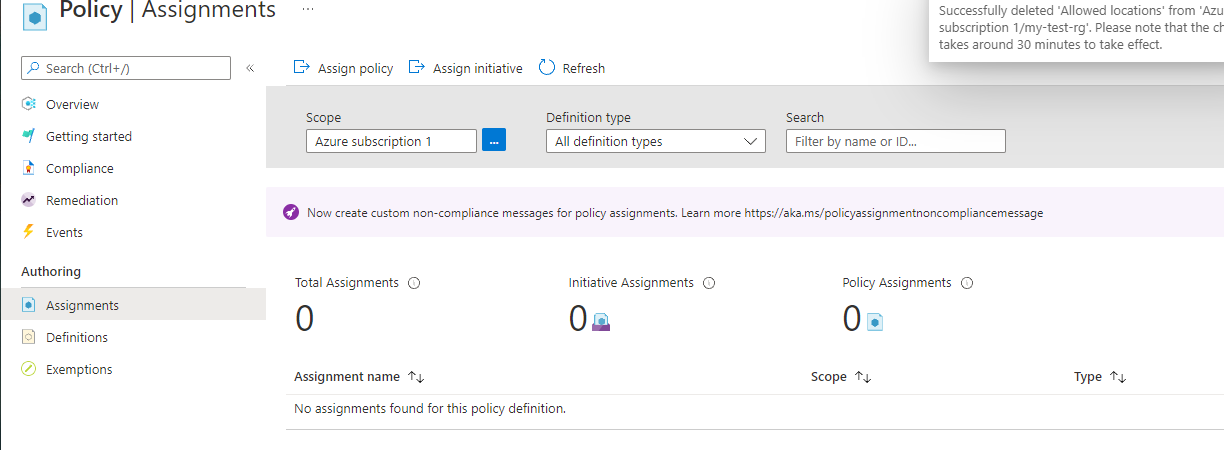




**Delete the policy assignment**

**Purpose:** I no longer need my policy assignment, as a result I removed it from my subscription

**What I did:** I selected policy and went back to assignment under authoring. On the allowed locations row, I selected delete assignment.The below confirms I no longer have any assignments



**Delete the resource group**

**Purpose:** As I no longer need my resource group, I removed it from my subscription

**What I did:** I accessed my-test-rg resource group. I selected overview and hit delete resource group. I entered my resource group name in order to delete it. I confirmed that my resource group had been deleted.The below confirms my-test-rg was deleted**.**

