**Lab20 – Understanding Table Storage – Azure**

**Table storage**

Azure Table storage is a service that stores structured NoSQL data in the cloud, providing a key/attribute store with a schemaless design. Because Table storage is schemaless, it's easy to adapt your data as the needs of your application evolve. Access to Table storage data is fast and cost-effective for many types of applications, and is typically lower in cost than traditional SQL for similar volumes of data.

You can use Table storage to store flexible datasets like user data for web applications, address books, device information, or other types of metadata your service requires. You can store any number of entities in a table, and a storage account may contain any number of tables, up to the capacity limit of the storage account.

**What is Table storage**

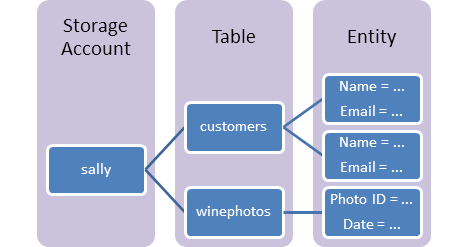
Azure Table storage stores large amounts of structured data. The service is a NoSQL datastore which accepts authenticated calls from inside and outside the Azure cloud. Azure tables are ideal for storing structured, non-relational data. Common uses of Table storage include:

* Storing TBs of structured data capable of serving web scale applications
* Storing datasets that don't require complex joins, foreign keys, or stored procedures and can be denormalized for fast access
* Quickly querying data using a clustered index
* Accessing data using the OData protocol and LINQ queries with WCF Data Service .NET Libraries

You can use Table storage to store and query huge sets of structured, non-relational data, and your tables will scale as demand increases.

**Table storage concepts**

Table storage contains the following components:



* **URL format:** Azure Table Storage accounts use this format: http://<storage account>.table.core.windows.net/<table>

Azure Cosmos DB Table API accounts use this format:http://<storage account>.table.cosmosdb.azure.com/<table>

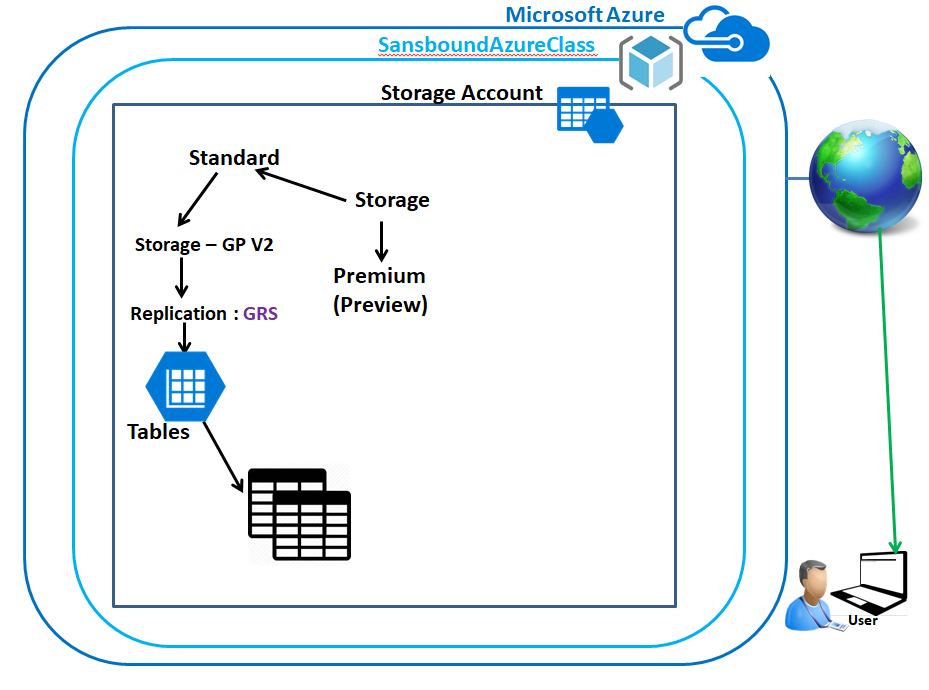
You can address Azure tables directly using this address with the OData protocol. For more information, see [OData.org](http://www.odata.org/).

* **Accounts:** All access to Azure Storage is done through a storage account. See [Azure Storage Scalability and Performance Targets](https://docs.microsoft.com/en-us/azure/storage/common/storage-scalability-targets) for details about storage account capacity.

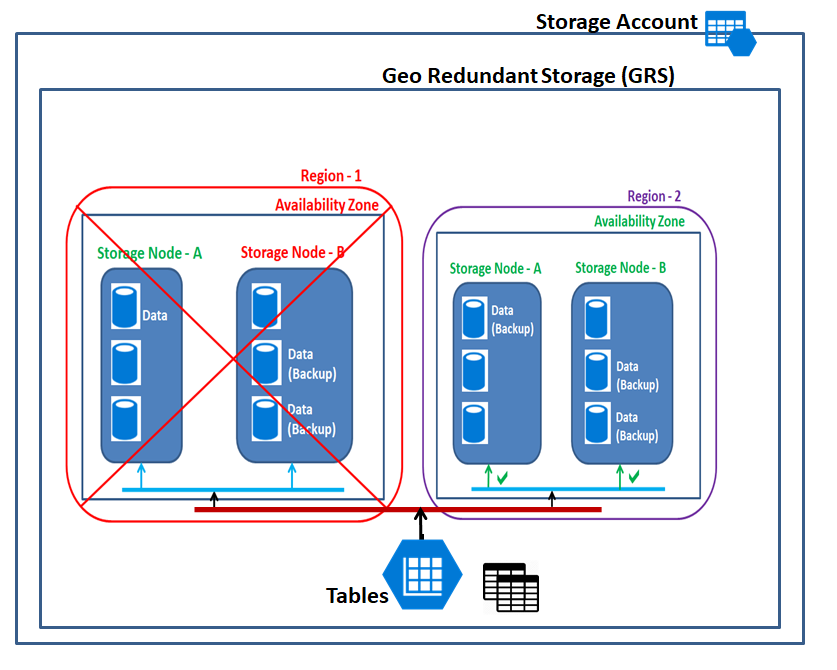
All access to Azure Cosmos DB is done through a Table API account. See [Create a Table API account](https://docs.microsoft.com/en-us/azure/cosmos-db/create-table-dotnet#create-a-database-account) for details creating a Table API account.

* **Table**: A table is a collection of entities. Tables don't enforce a schema on entities, which means a single table can contain entities that have different sets of properties.
* **Entity**: An entity is a set of properties, similar to a database row. An entity in Azure Storage can be up to 1MB in size. An entity in Azure Cosmos DB can be up to 2MB in size.
* **Properties**: A property is a name-value pair. Each entity can include up to 252 properties to store data. Each entity also has three system properties that specify a partition key, a row key, and a timestamp. Entities with the same partition key can be queried more quickly, and inserted/updated in atomic operations. An entity's row key is its unique identifier within a partition.

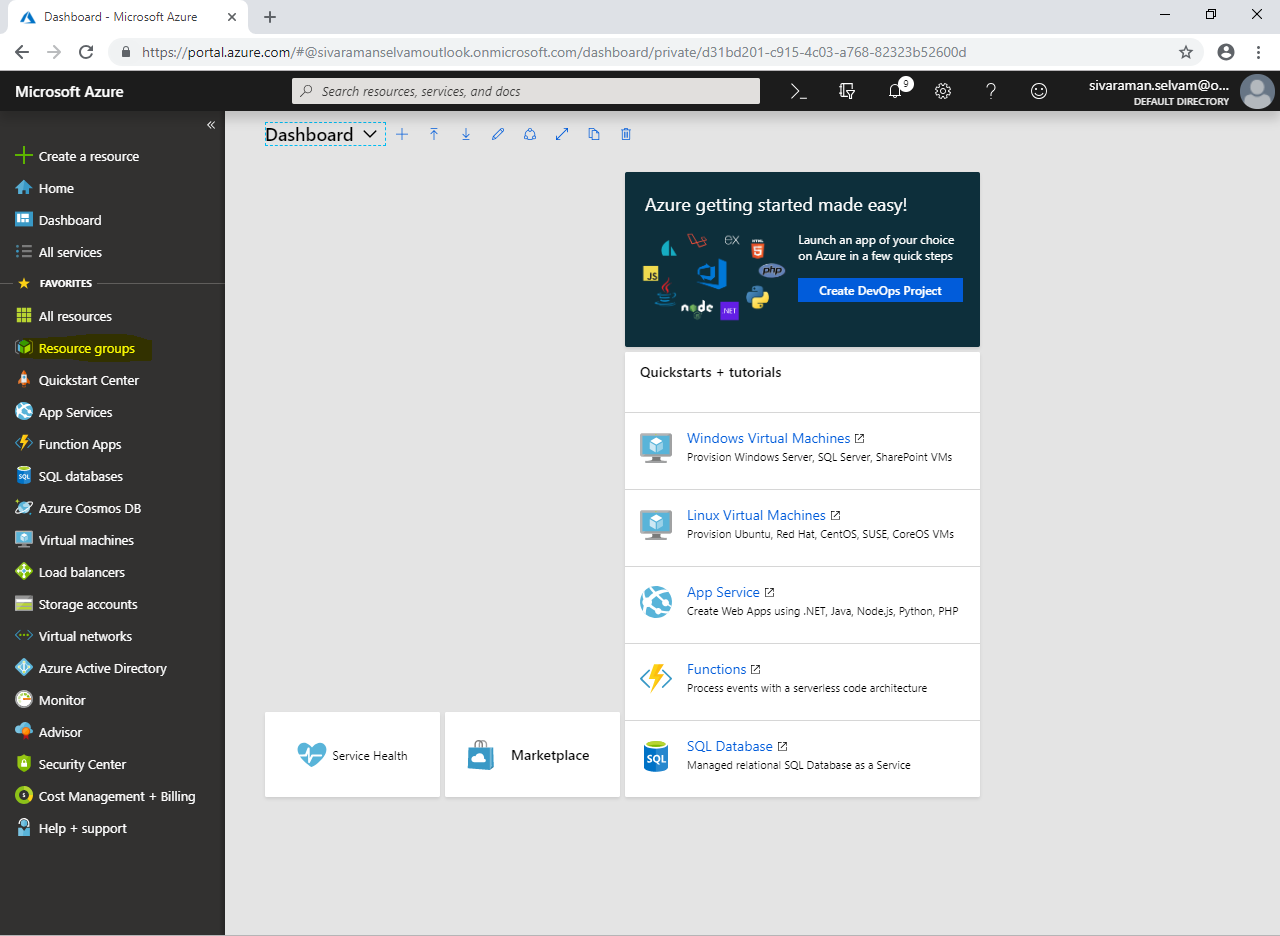
**Topology:**



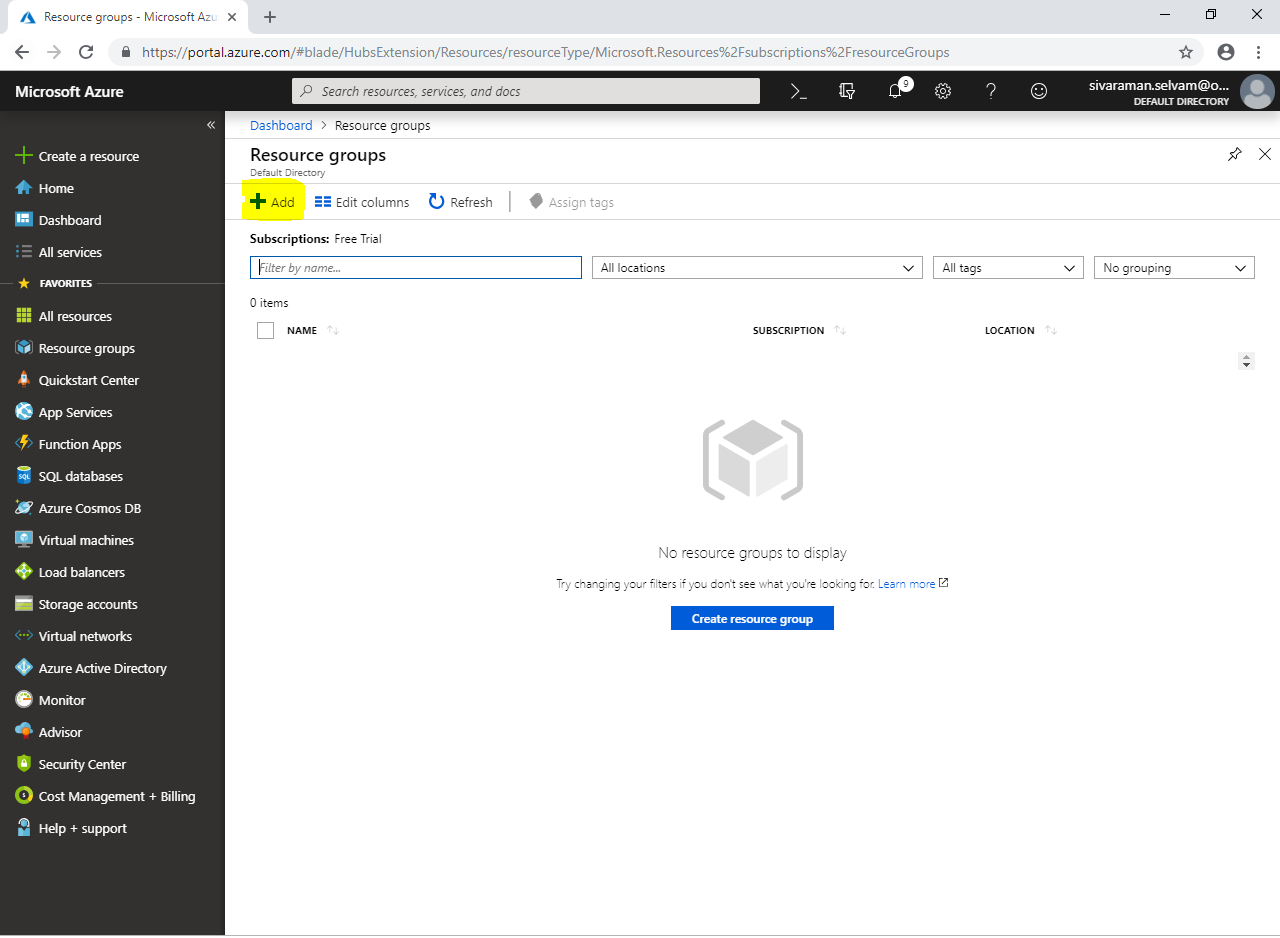
**Back-End Topology:**



Click **“Resource groups”**.



Click **“Add”**.

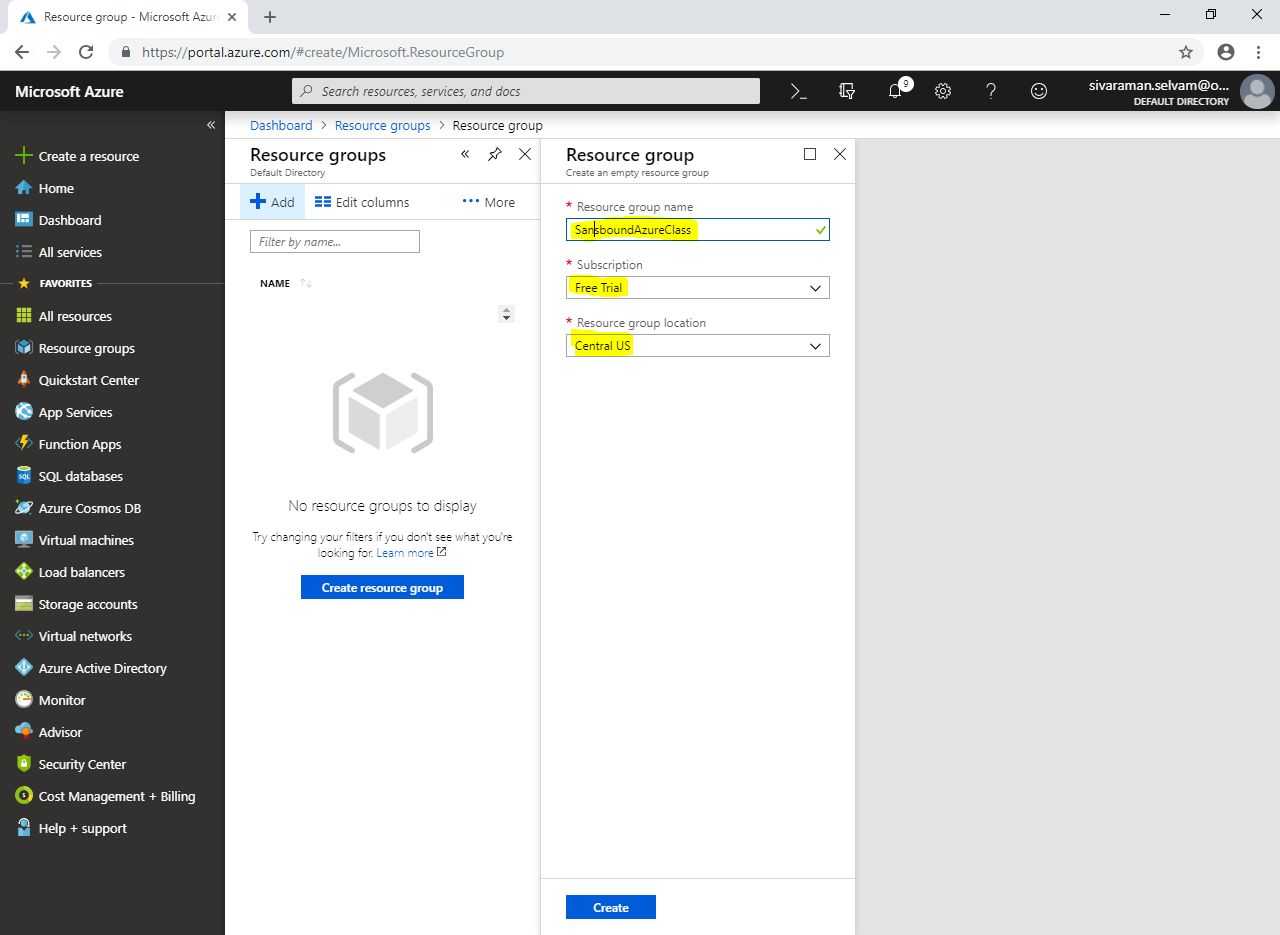


While create **“Resource group”,**

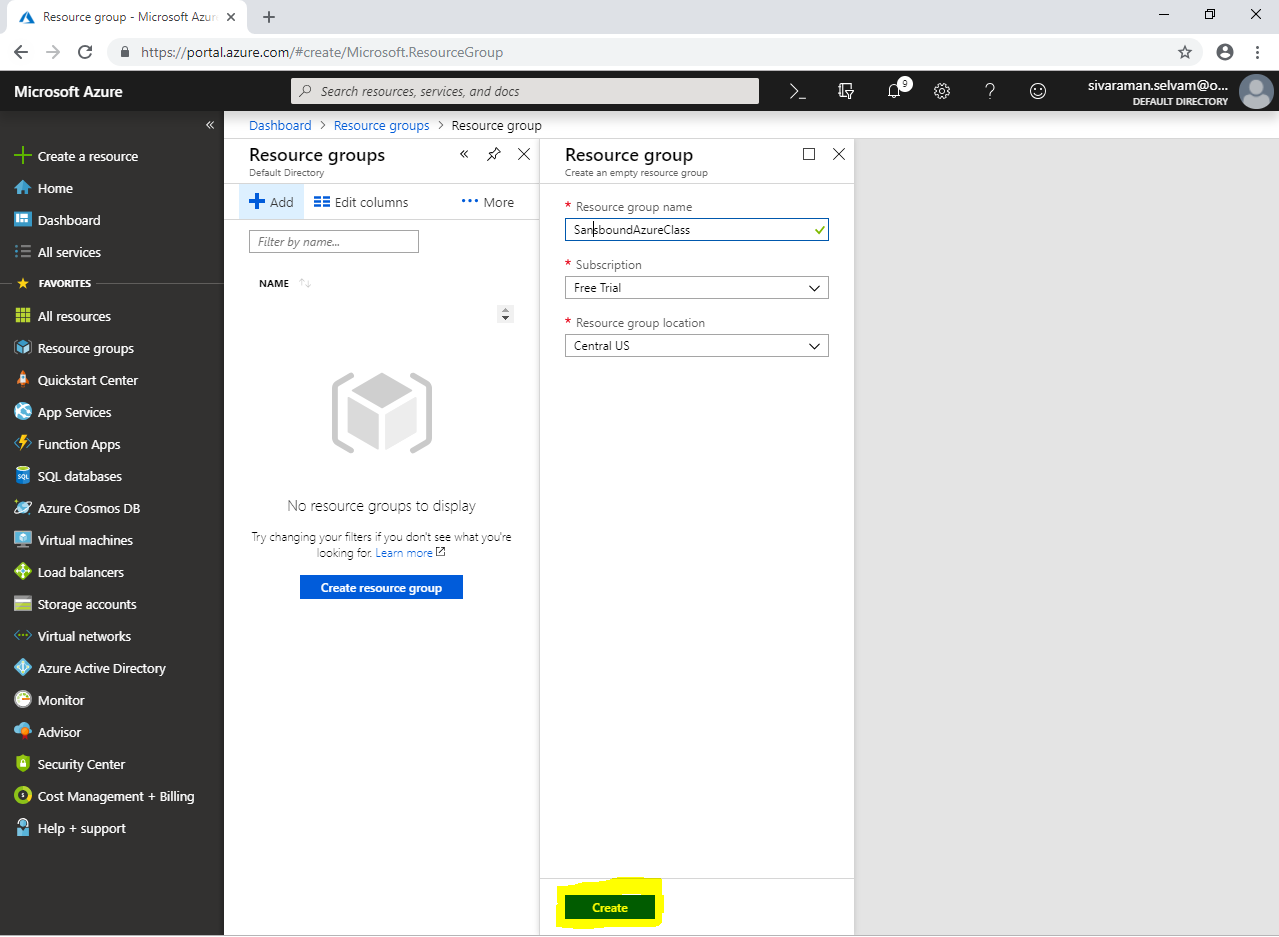
It requires **“Resource group name”,** type as **“SansboundAzureClass”**.

Select **“Subscription”** as **“Free Trial”**.

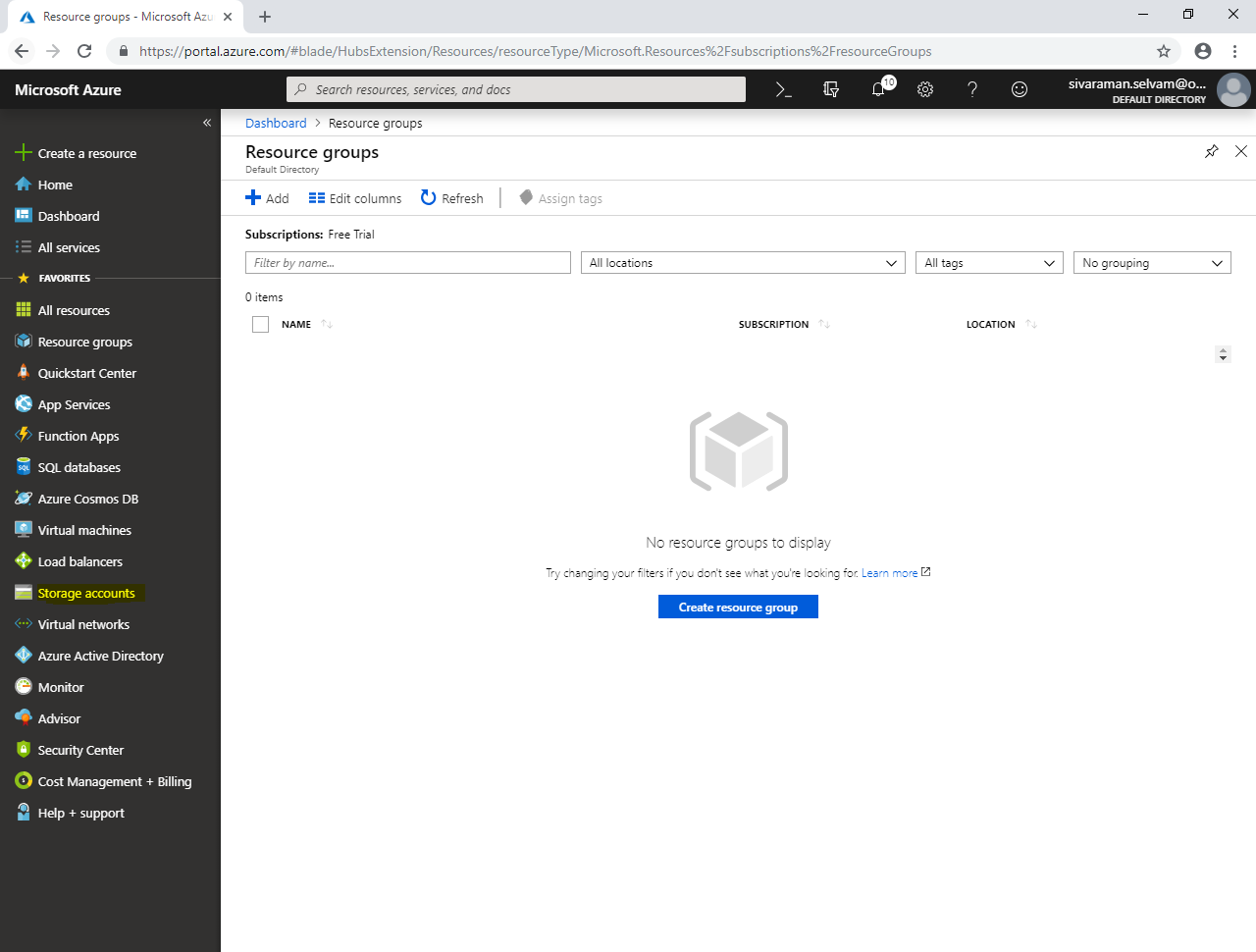
Select **“Resource group location”** as **“Central US”**.



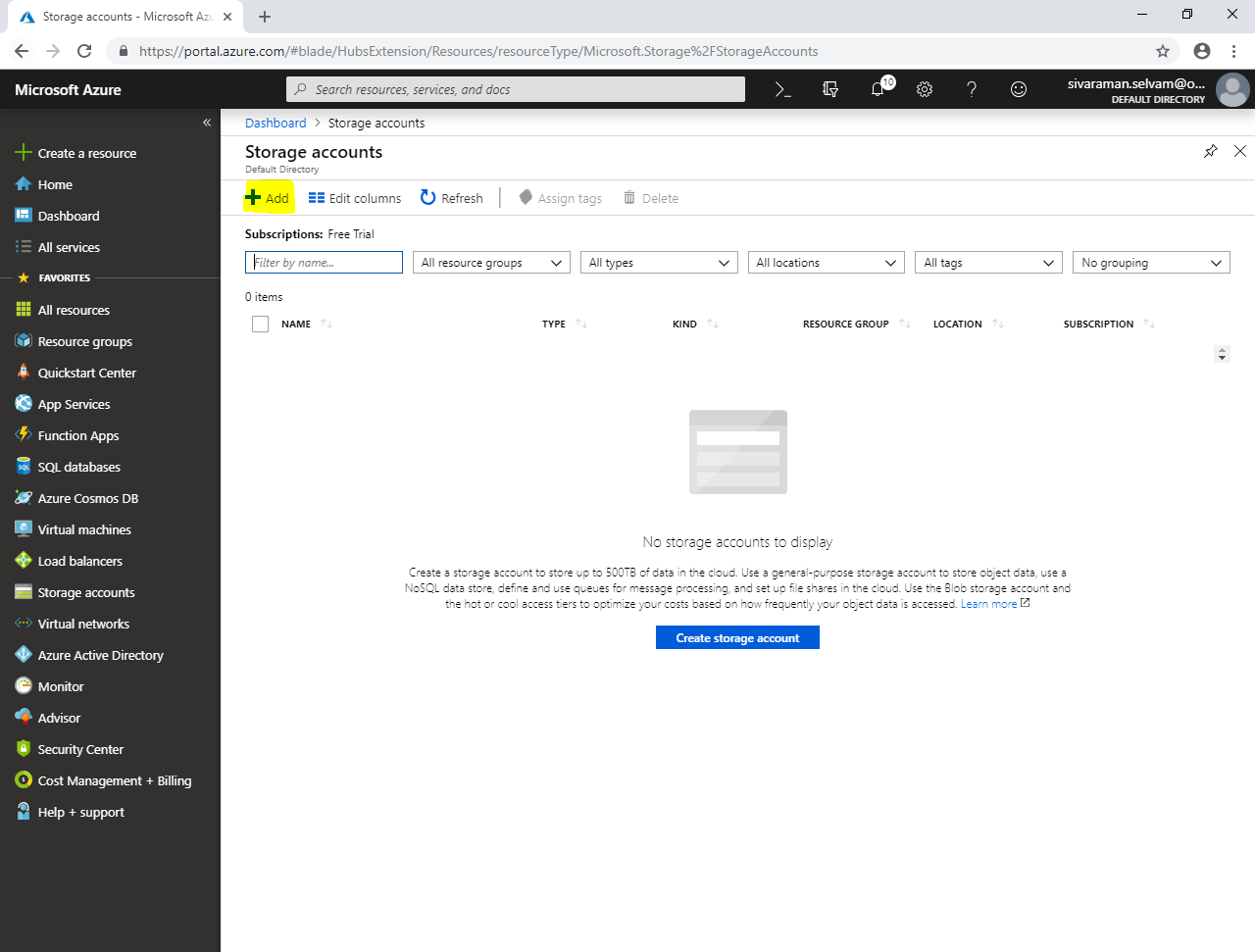
Click **“Create”.**



Click **“Storage accounts”** in left side of the panel.



Click **“Add”**.



Select **“Subscription”** as **“Free Trial”**.

Select **“Resource group”** as **“SansboundAzureClass”**.

Type **“Storage account name”** as **“sansboundstorage”**.

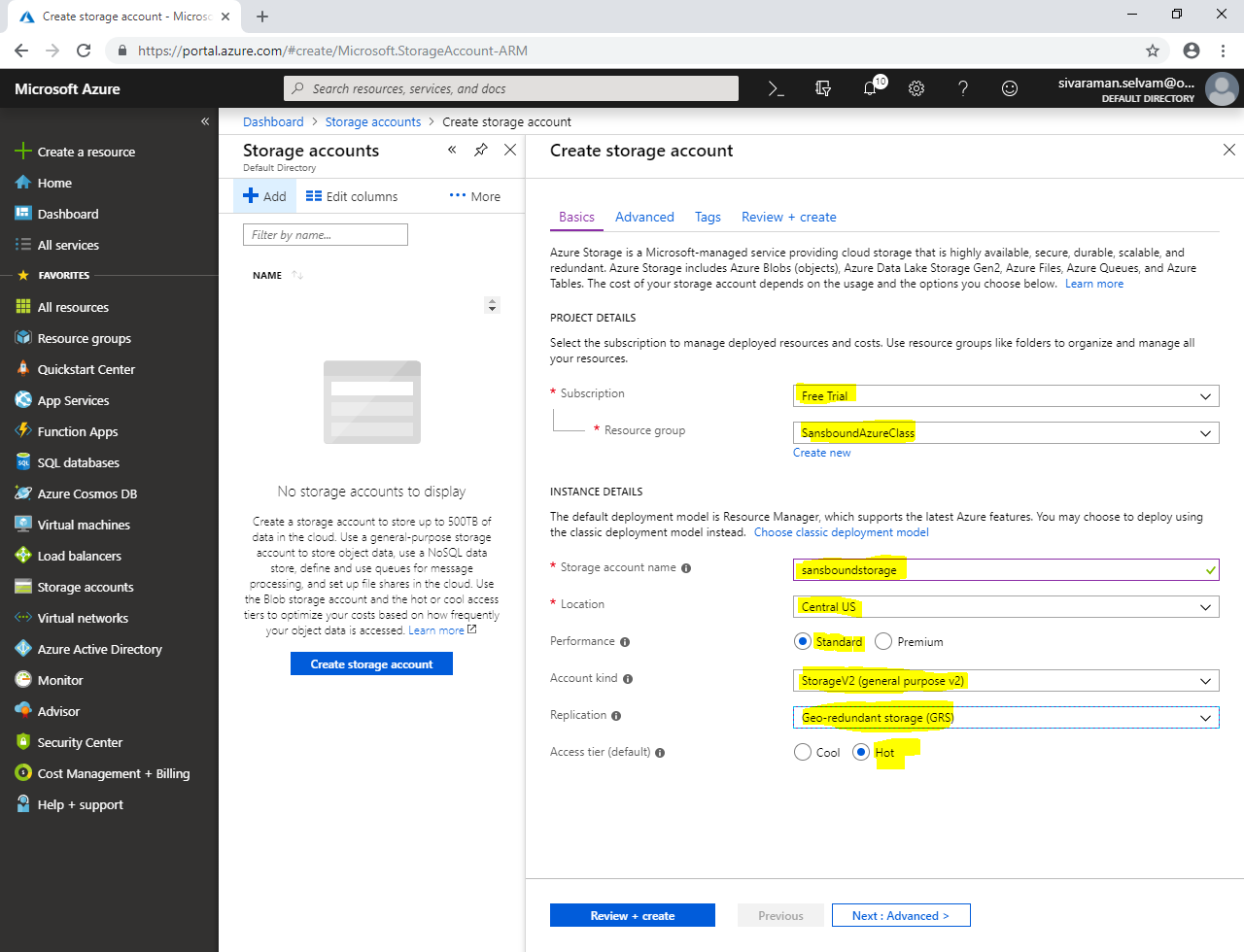
Select **“Location”** as **“Central US”**.

Select **“Performance”** as **“Standard”**.

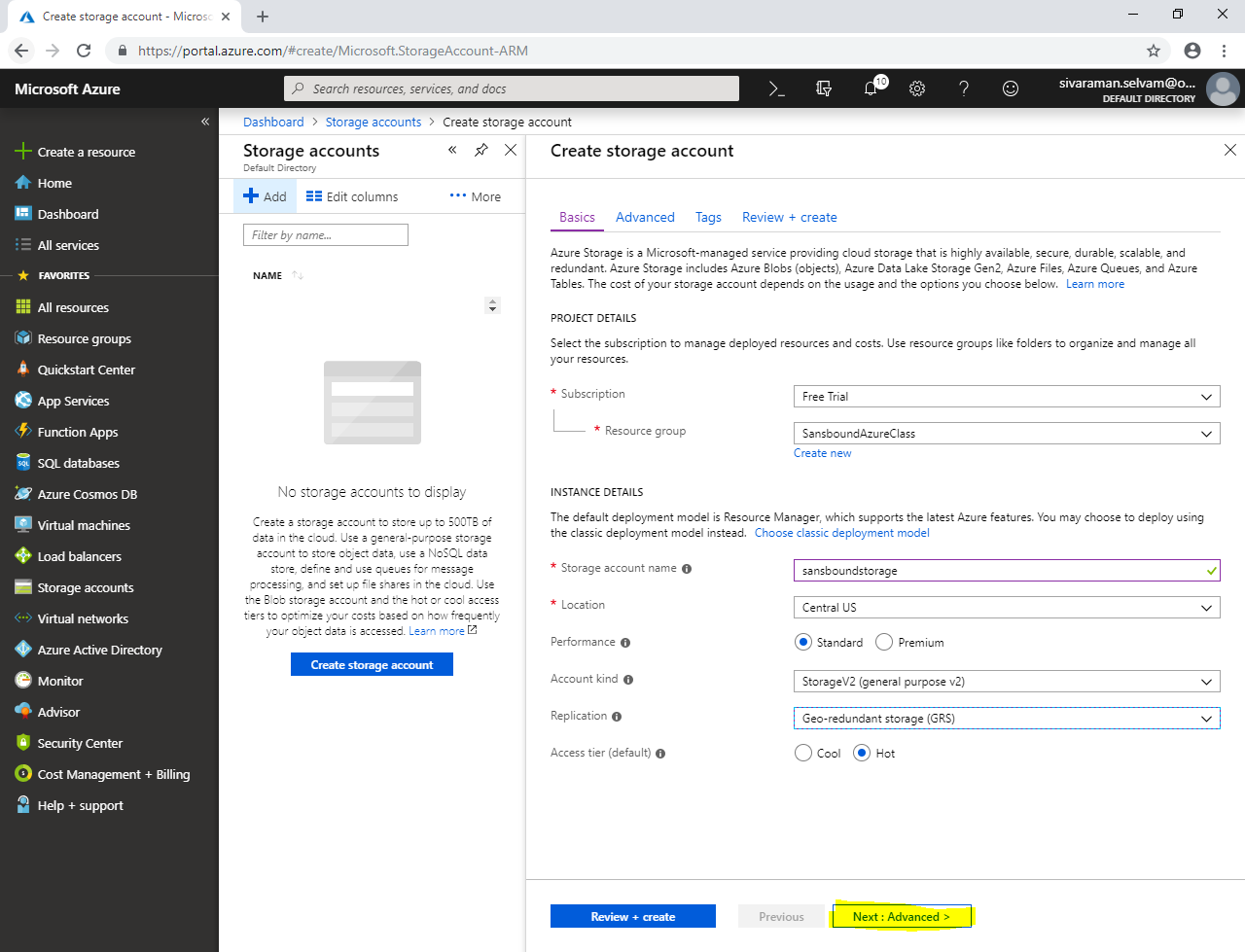
Select **“Account kind”** as **“Storage V2”**.

Select **“Replication”** as **“Geo-redundant Storage (GRS)”**.

Click **“Access Tier”** as **“Hot”**.

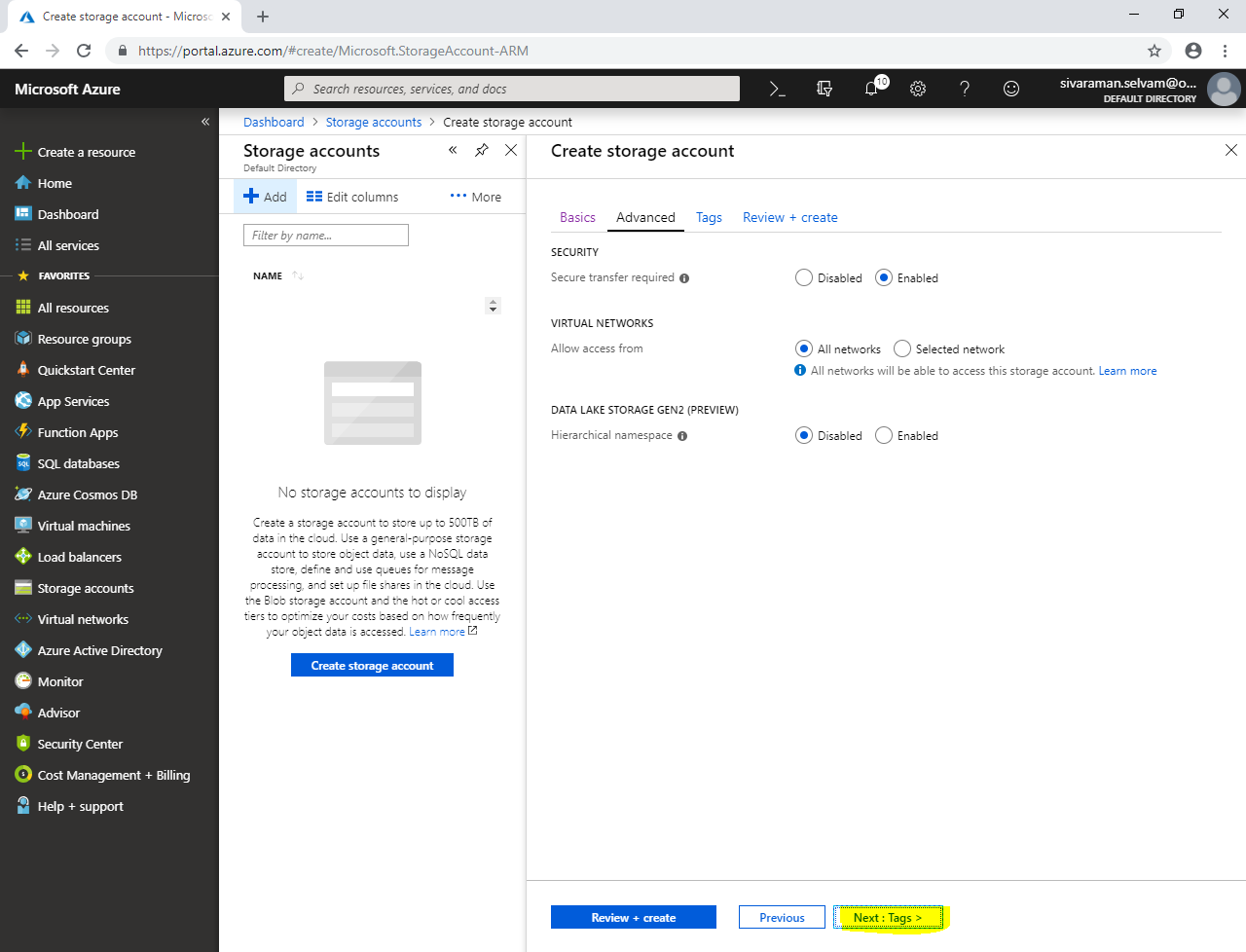


Click **“Next : Advanced >”**.



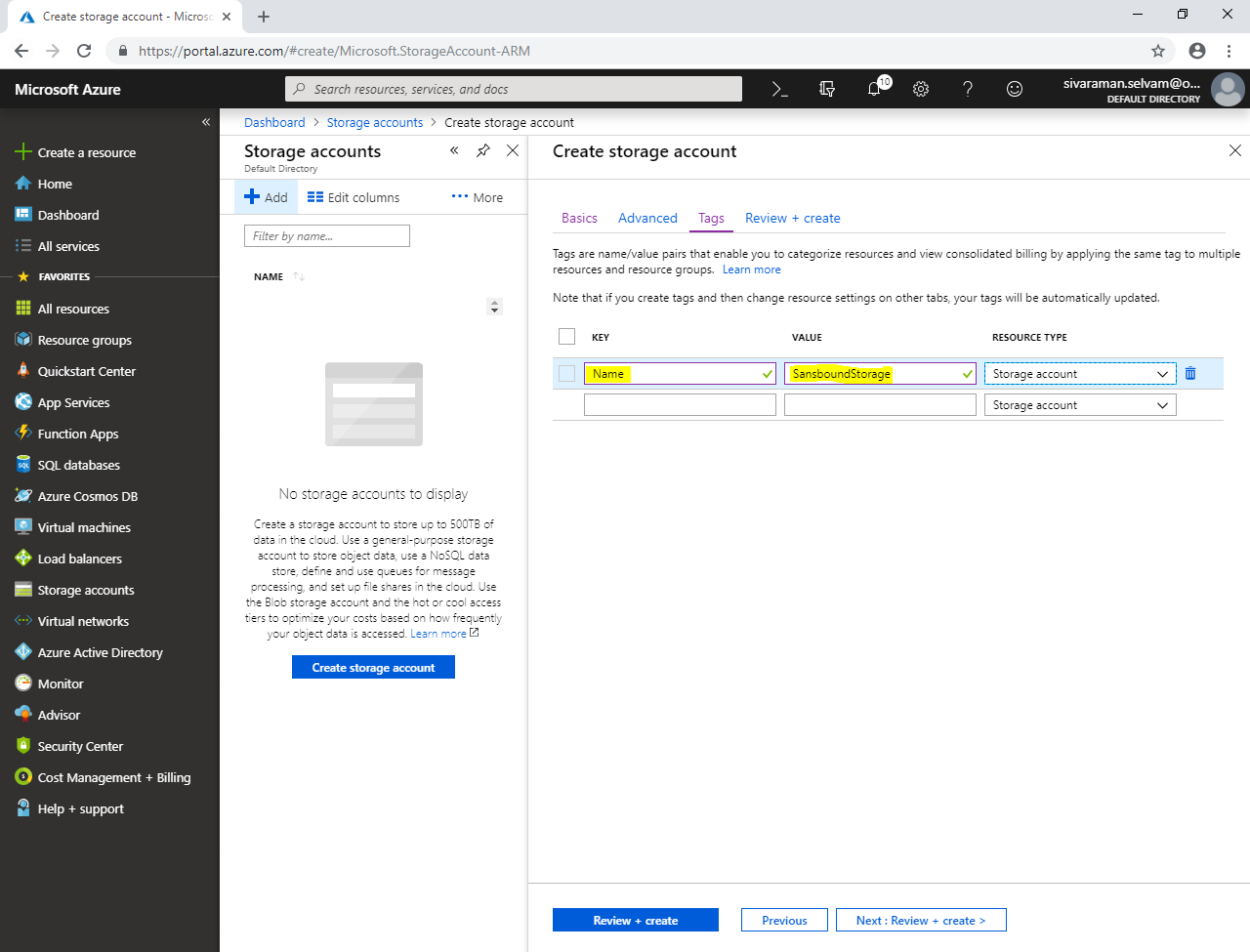
In **“Advanced”**

Click **“Next : Tags >”**.

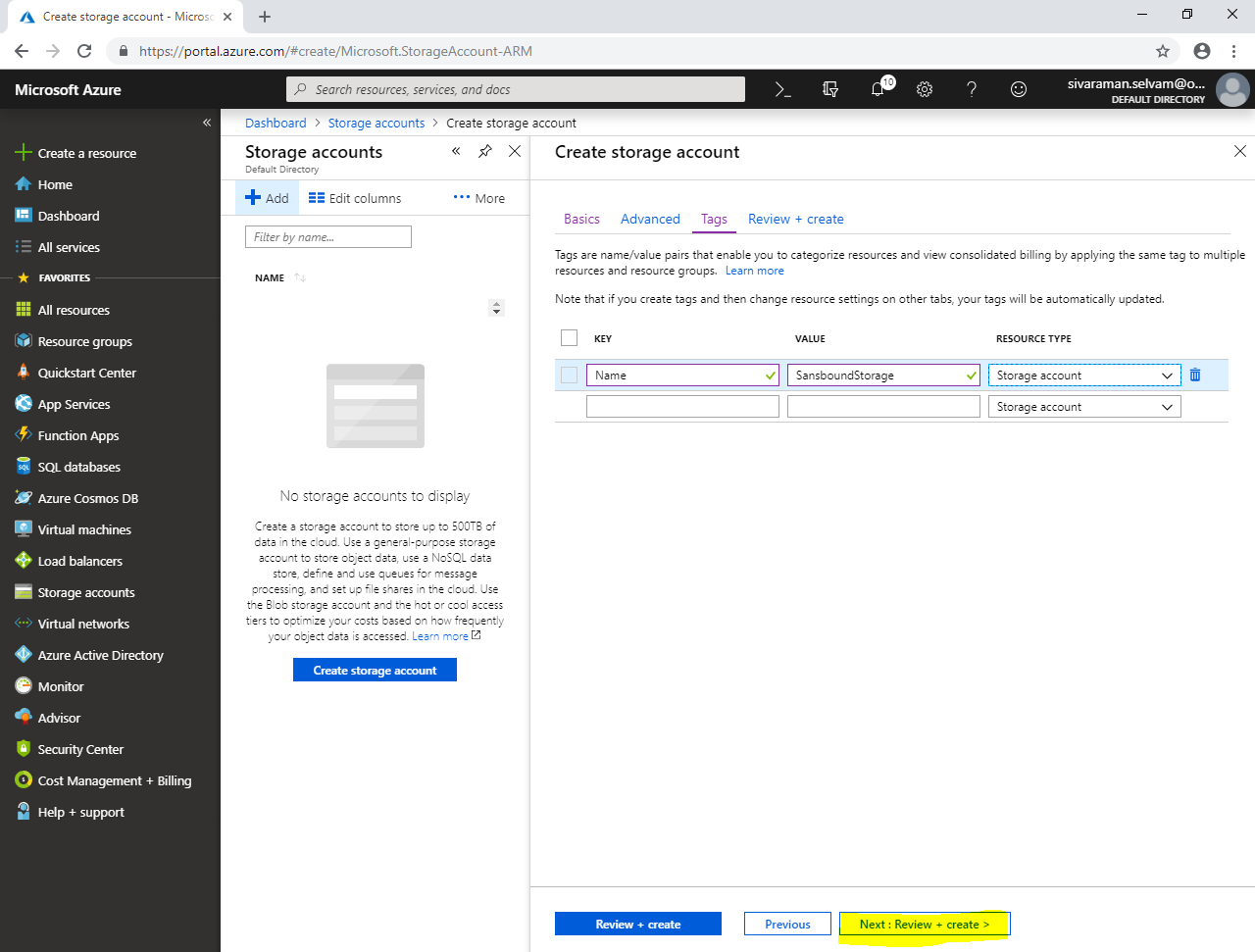


In **“Tags”**.

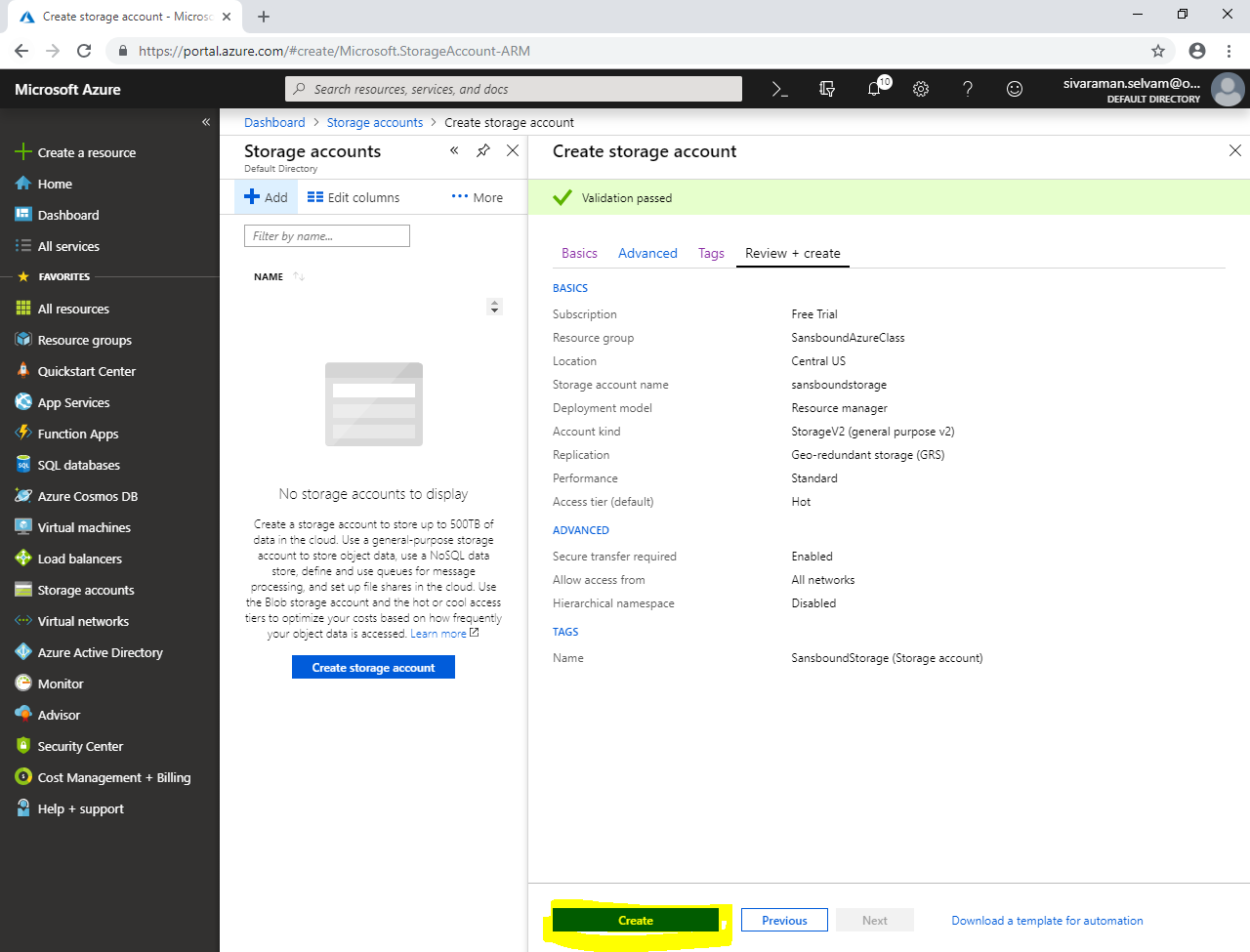
Click **“KEY”** as **“Name”** and **“VALUE”** as **“SansboundStorage”**.



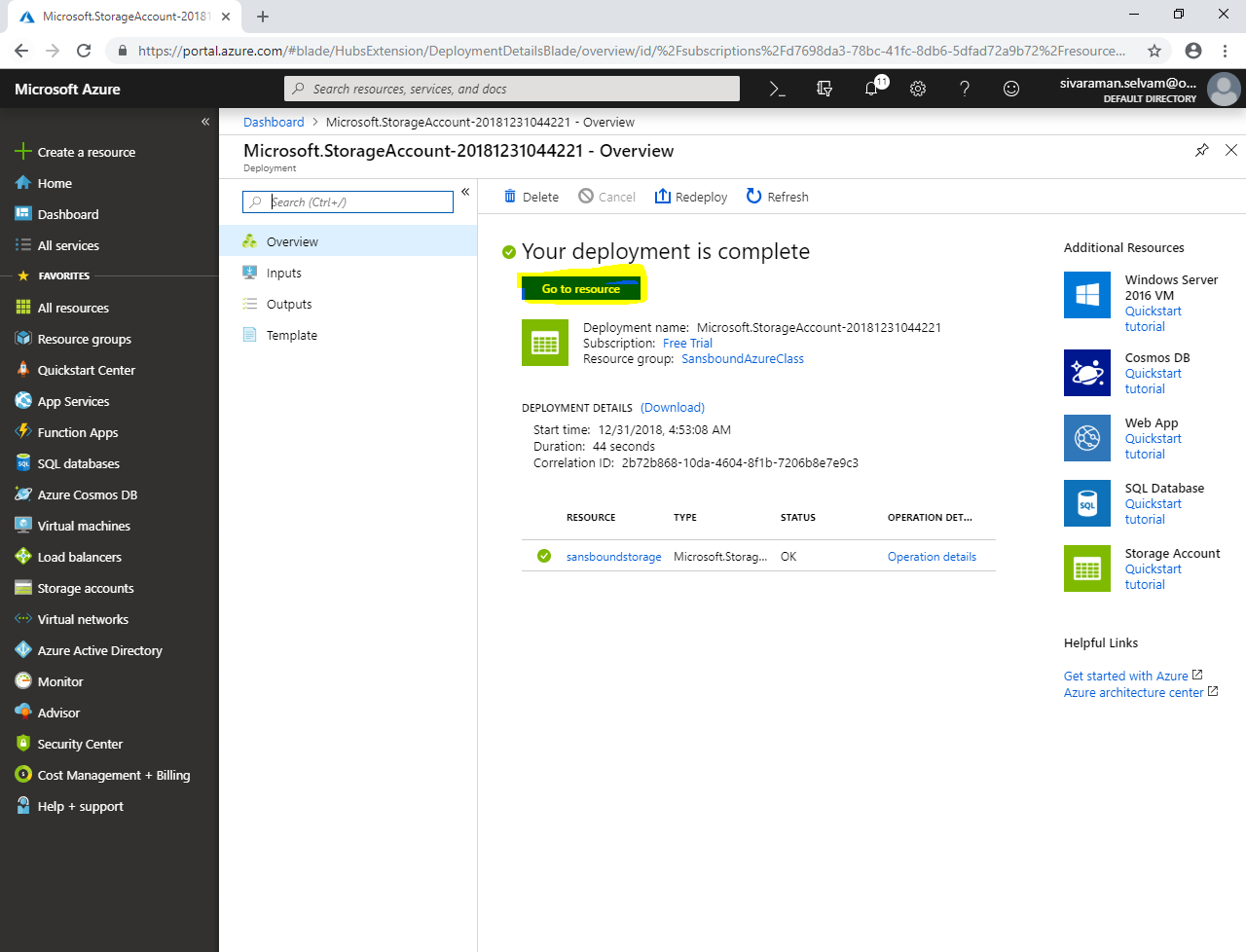
Click **“Review + create”**.



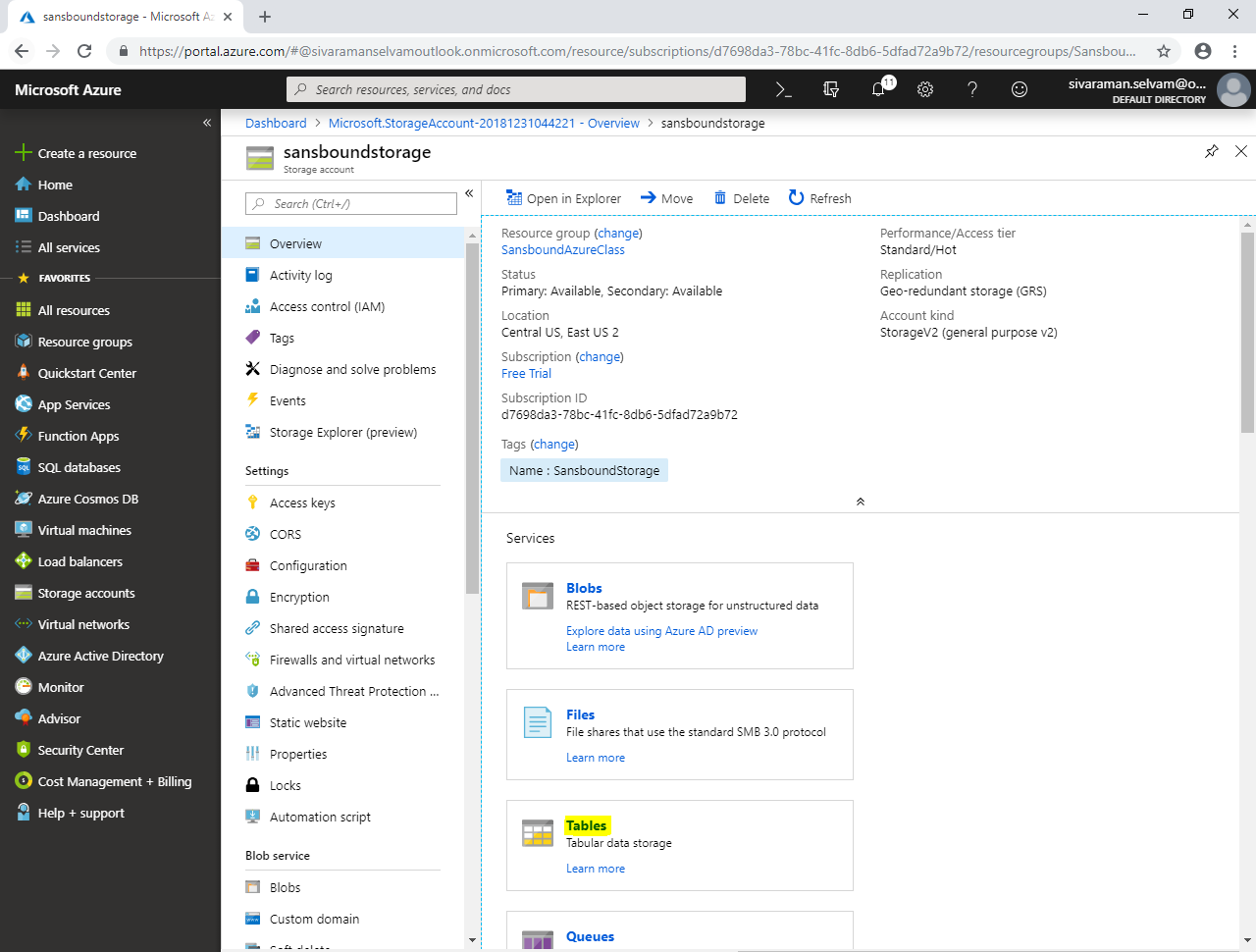
Click **“Create”.**



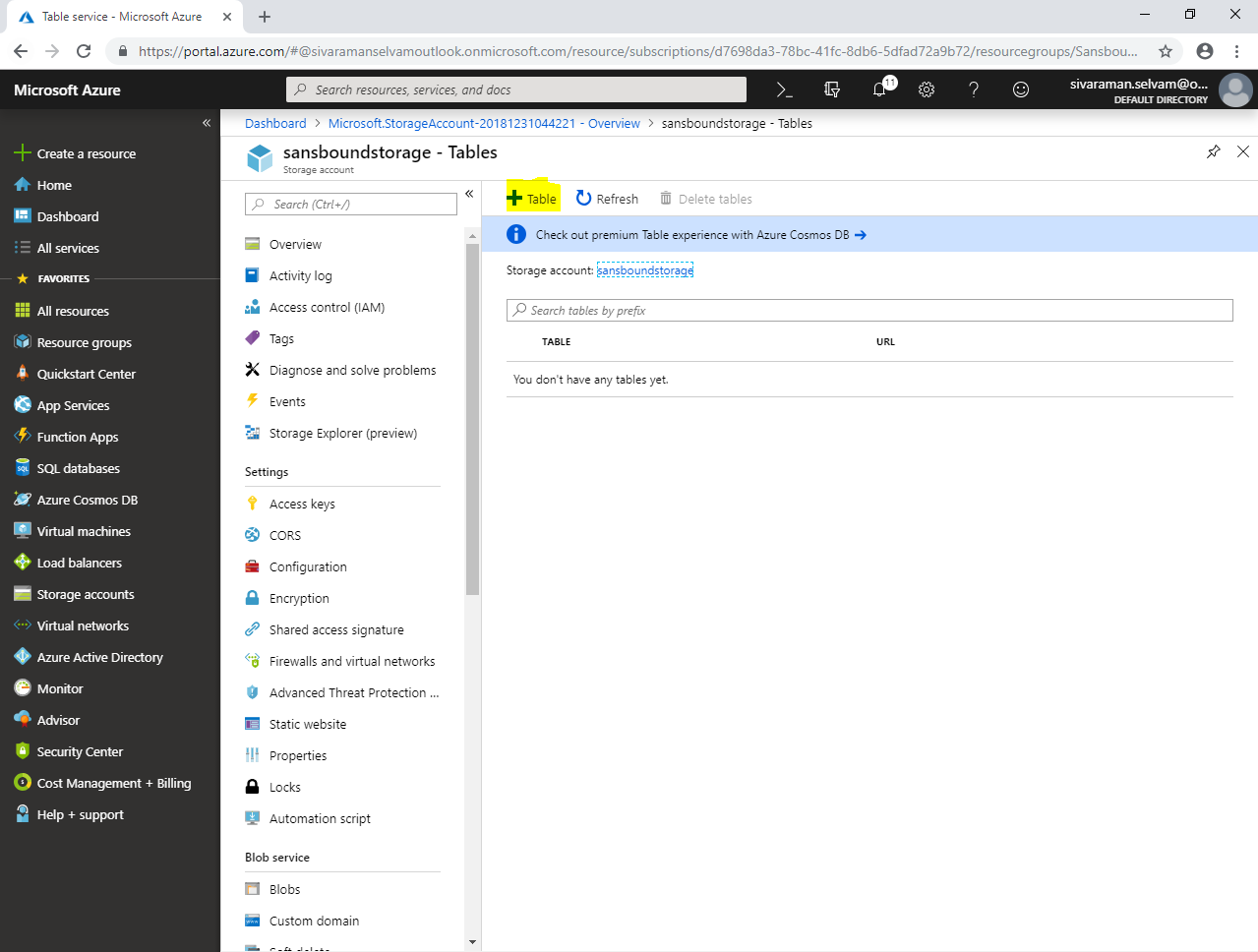
Click **“Go to resource”.**



Click **“Tables”**.

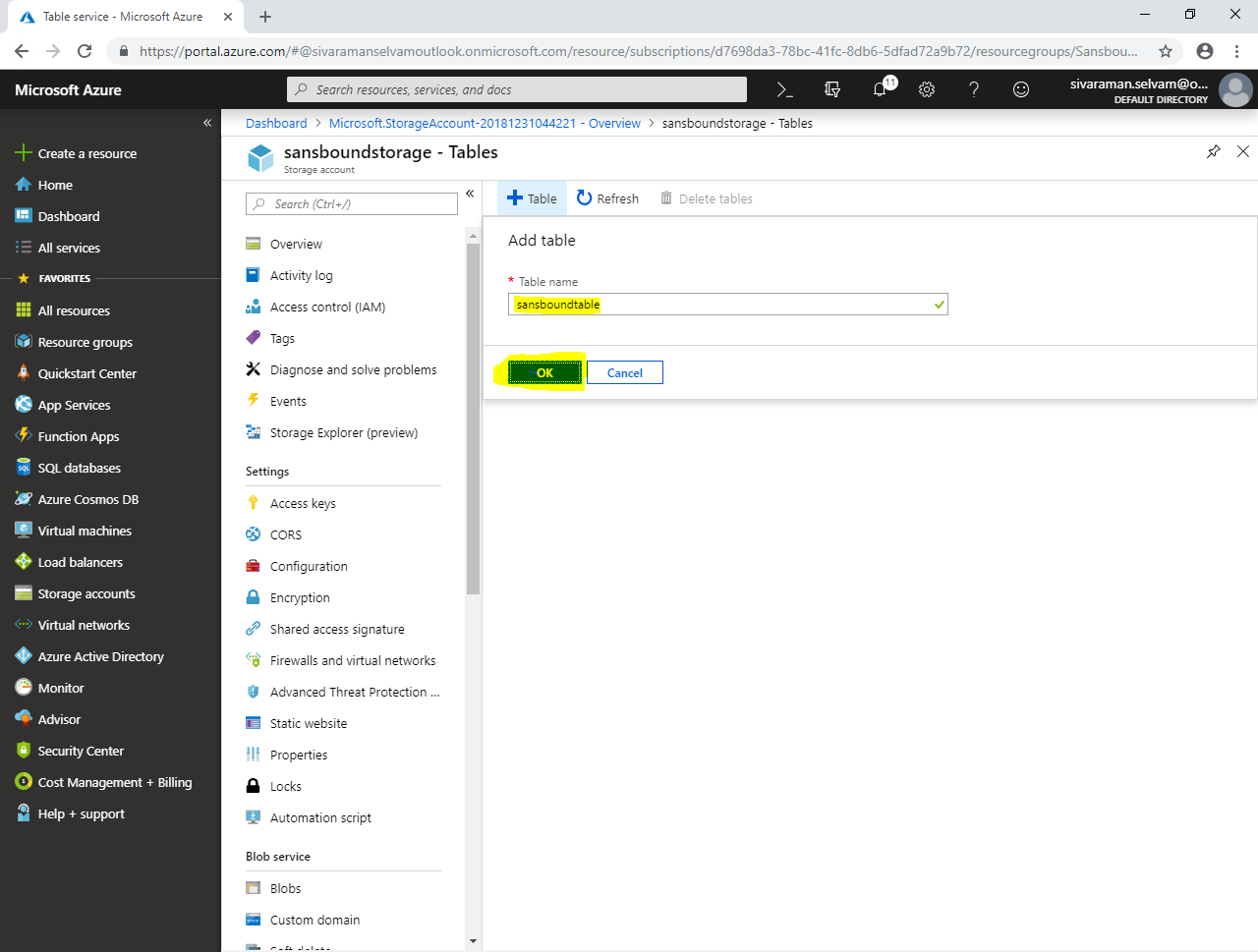


Click **“Table”**.



While add table, type **“Table name”** as **“sansboundtable”**.

Click **“Ok”**.



You can apply policy for this tables, we will see it later.

