Lab15 – Understanding Local Redundant Storage (LRS) - Azure

Locally redundant storage (LRS): (Low-cost data redundancy)

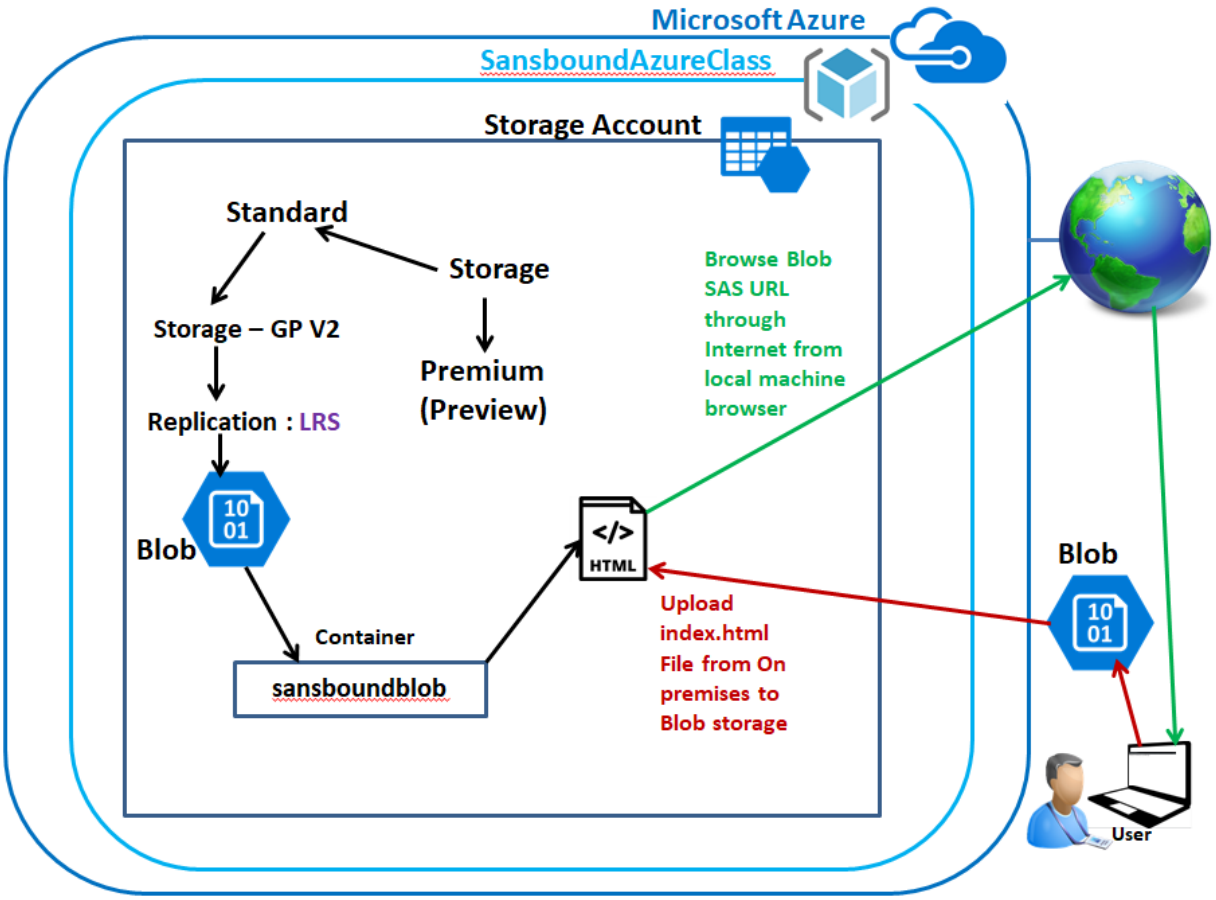
Locally redundant storage (LRS) provides at least 99.999999999% (11 nines) durability of objects over a given year. LRS provides this object durability by replicating your data to a storage scale unit. A datacenter, located in the region where you created your storage account, hosts the storage scale unit. A write request to an LRS storage account returns successfully only after the data is written to all replicas. Each replica resides in separate fault domains and update domains within a storage scale unit.

A storage scale unit is a collection of racks of storage nodes. A fault domain (FD) is a group of nodes that represent a physical unit of failure. Think of a fault domain as nodes belonging to the same physical rack. An upgrade domain (UD) is a group of nodes that are upgraded together during the process of a service upgrade (rollout). The replicas are spread across UDs and FDs within one storage scale unit. This architecture ensures your data is available if a hardware failure affects a single rack or when nodes are upgraded during a service upgrade.

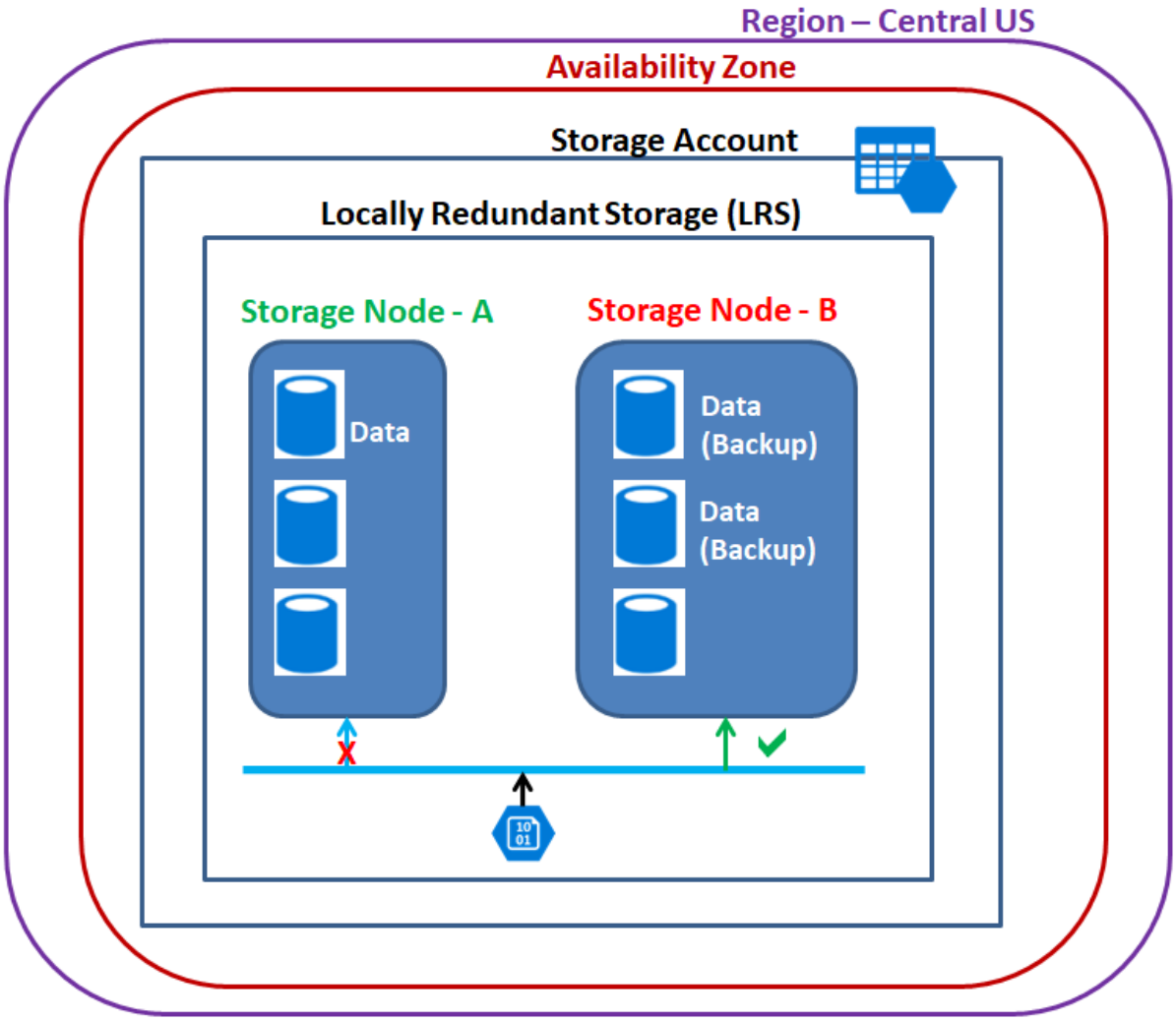
LRS is the lowest-cost replication option and offers the least durability compared to other options. If a datacenter-level disaster (for example, fire or flooding) occurs, all replicas may be lost or unrecoverable. To mitigate this risk, Microsoft recommends using either zone-redundant storage (ZRS) or geo-redundant storage (GRS).

- If your application stores data that can be easily reconstructed if data loss occurs, you may opt for LRS.
- Some applications are restricted to replicating data only within a country due to data governance requirements. In some cases, the paired regions across which the data is replicated for GRS accounts may be in another country.

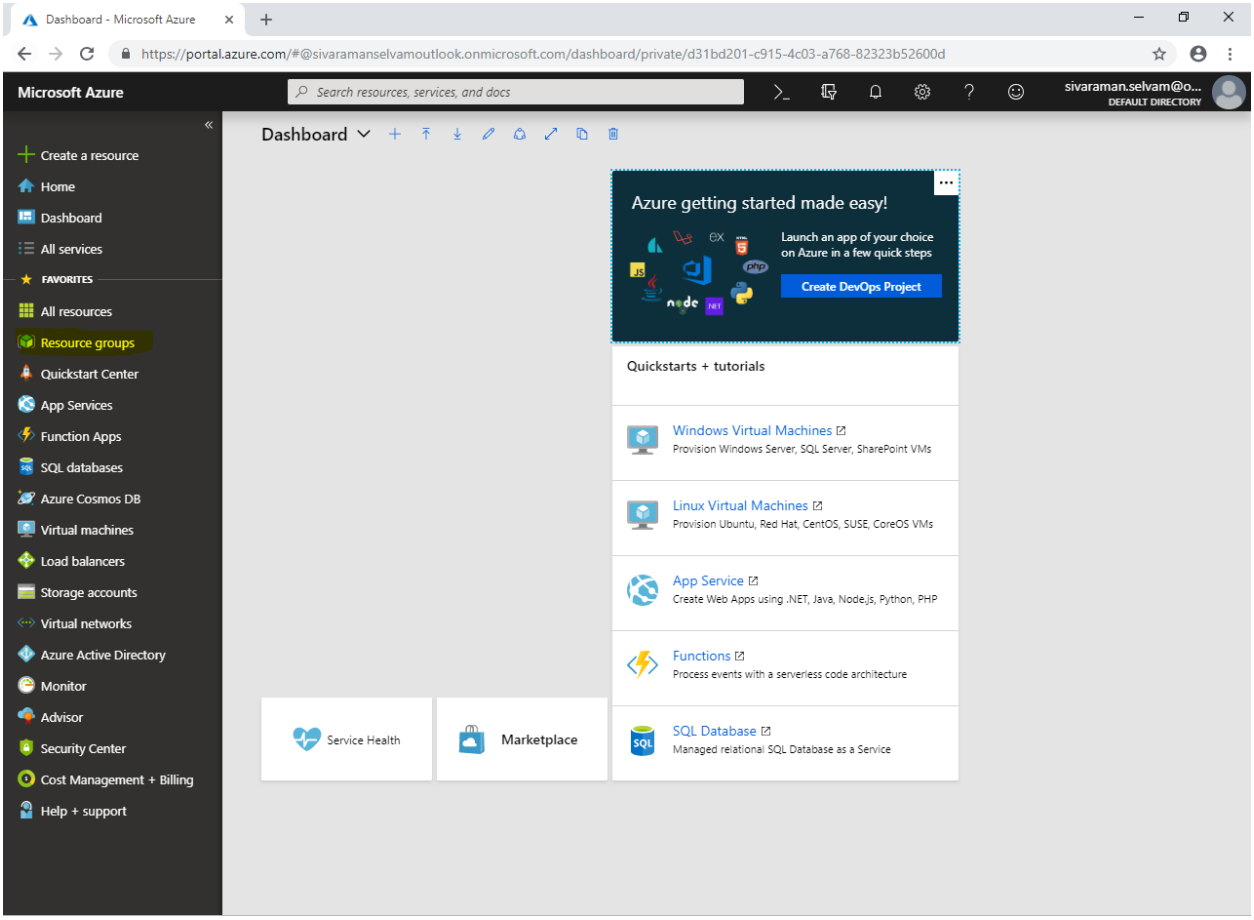
Topology


Microsoft Azure
SansboundAzureClass
Storage Account
Standard
Storage
Storage – GP V2
Premium
(Preview)
Replication : LRS
Blob
Container
sansboundblob
HTML
Browse Blob
SAS URL
through
Internet from
local machine
browser
Upload
index.html
File from On
premises to
Blob storage
Blob
User

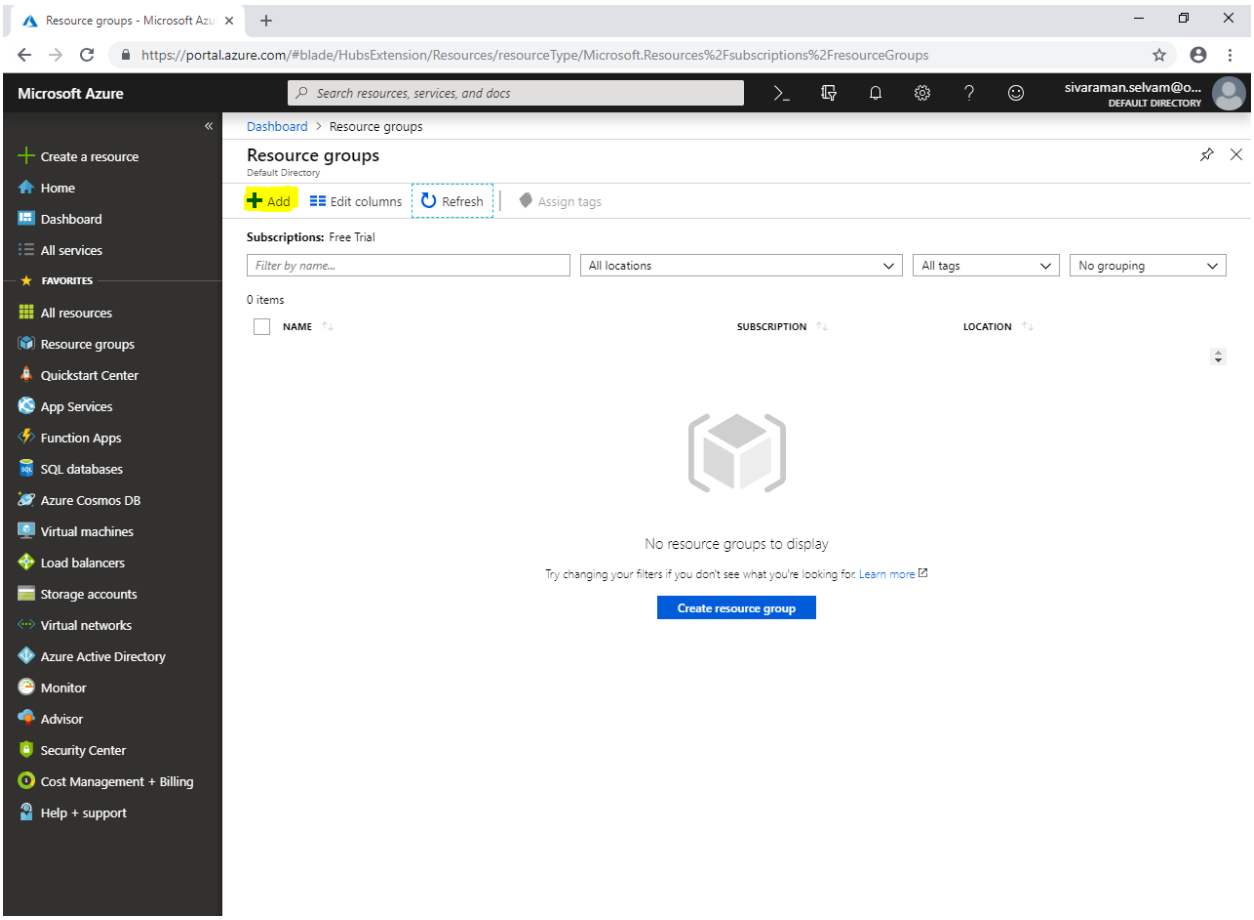
Locally Redundant Storage (Back-End):



In Azure portal, click **“Resource groups”**.



Click **"Add"**.


Resource groups - Microsoft Azu
https://portal.azure.com/#blade/HubsExtension/Resources/resourceType/Microsoft.Resources%2Fsubscriptions%2FresourceGroups
Microsoft Azure
Search resources, services, and docs
sivaraman.selvam@o...
DEFAULT DIRECTORY
Create a resource
Home
Dashboard
All services
FAVORITES
All resources
Resource groups
Quickstart Center
App Services
Function Apps
SQL databases
Azure Cosmos DB
Virtual machines
Load balancers
Storage accounts
Virtual networks
Azure Active Directory
Monitor
Advisor
Security Center
Cost Management + Billing
Help + support
Dashboard > Resource groups
Resource groups
Default Directory
Add
Edit columns
Refresh
Assign tags
Subscriptions: Free Trial
Filter by name...
All locations
All tags
No grouping
0 items
NAME
SUBSCRIPTION
LOCATION
No resource groups to display
Try changing your filters if you don't see what you're looking for. Learn more
Create resource group

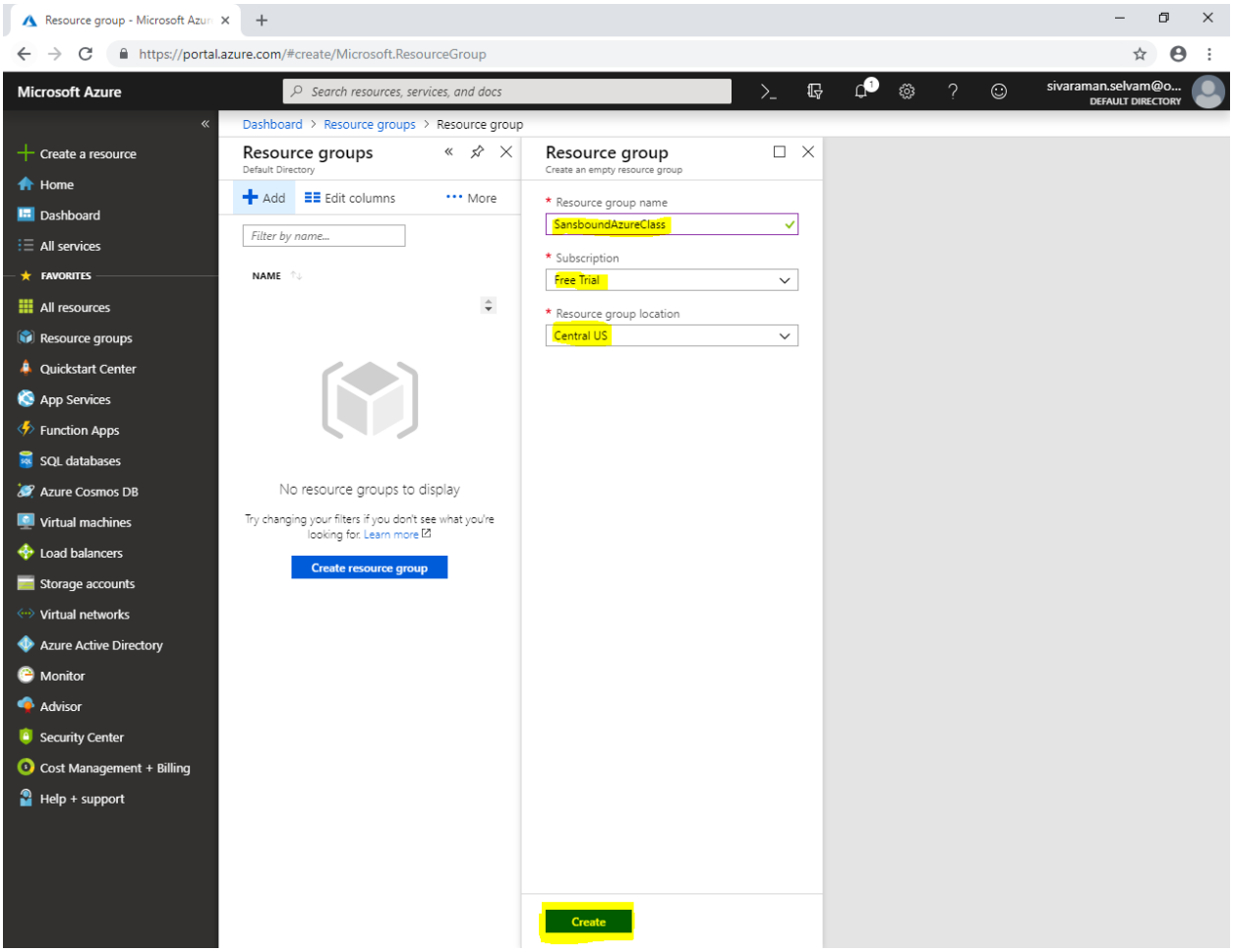
While create **“Resource group”**

Type **“Resource group name”** as **“SansboundAzureClass”**.

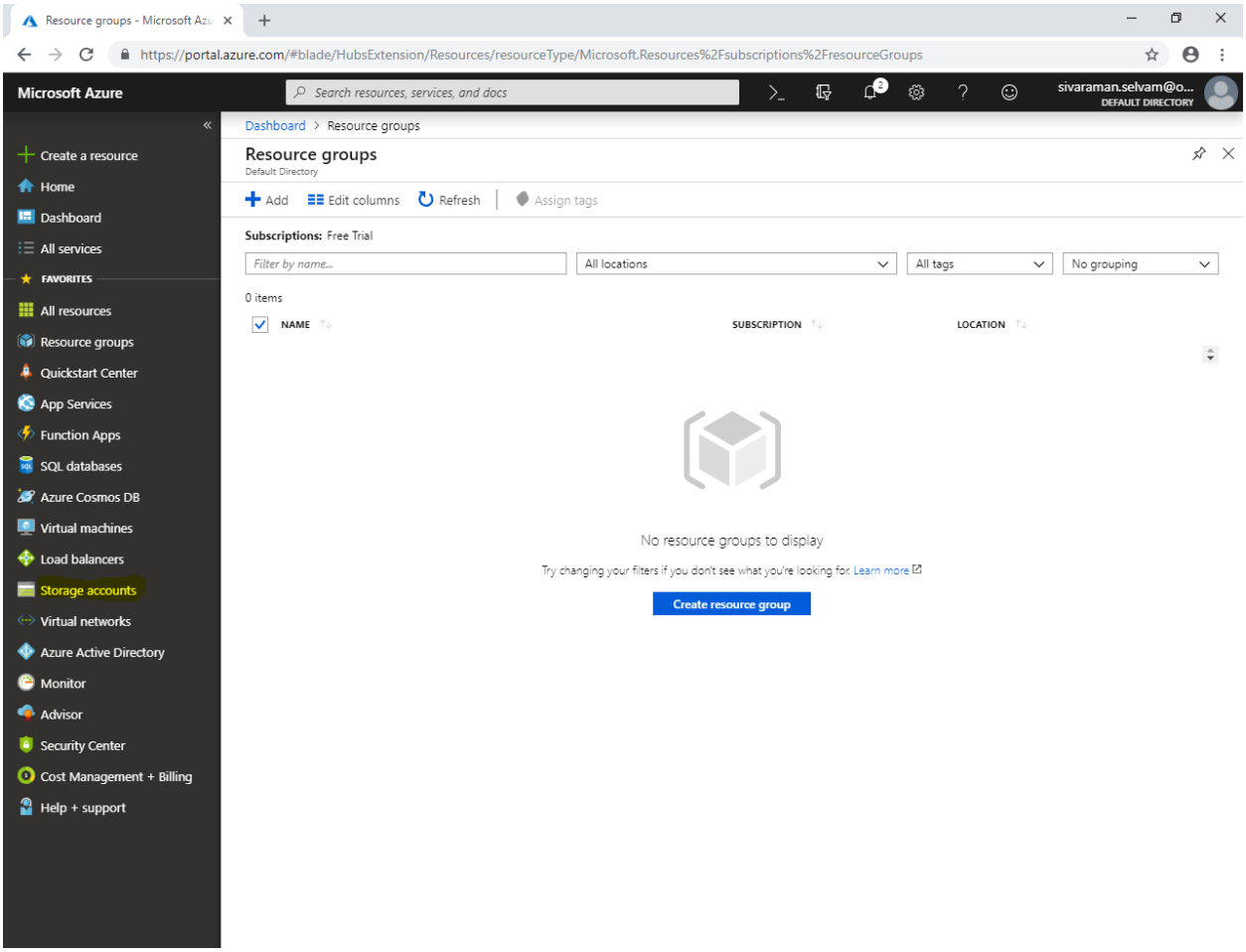
Select **“Subscription”** as **“Free Trial”**.

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

Click **“Create”**.

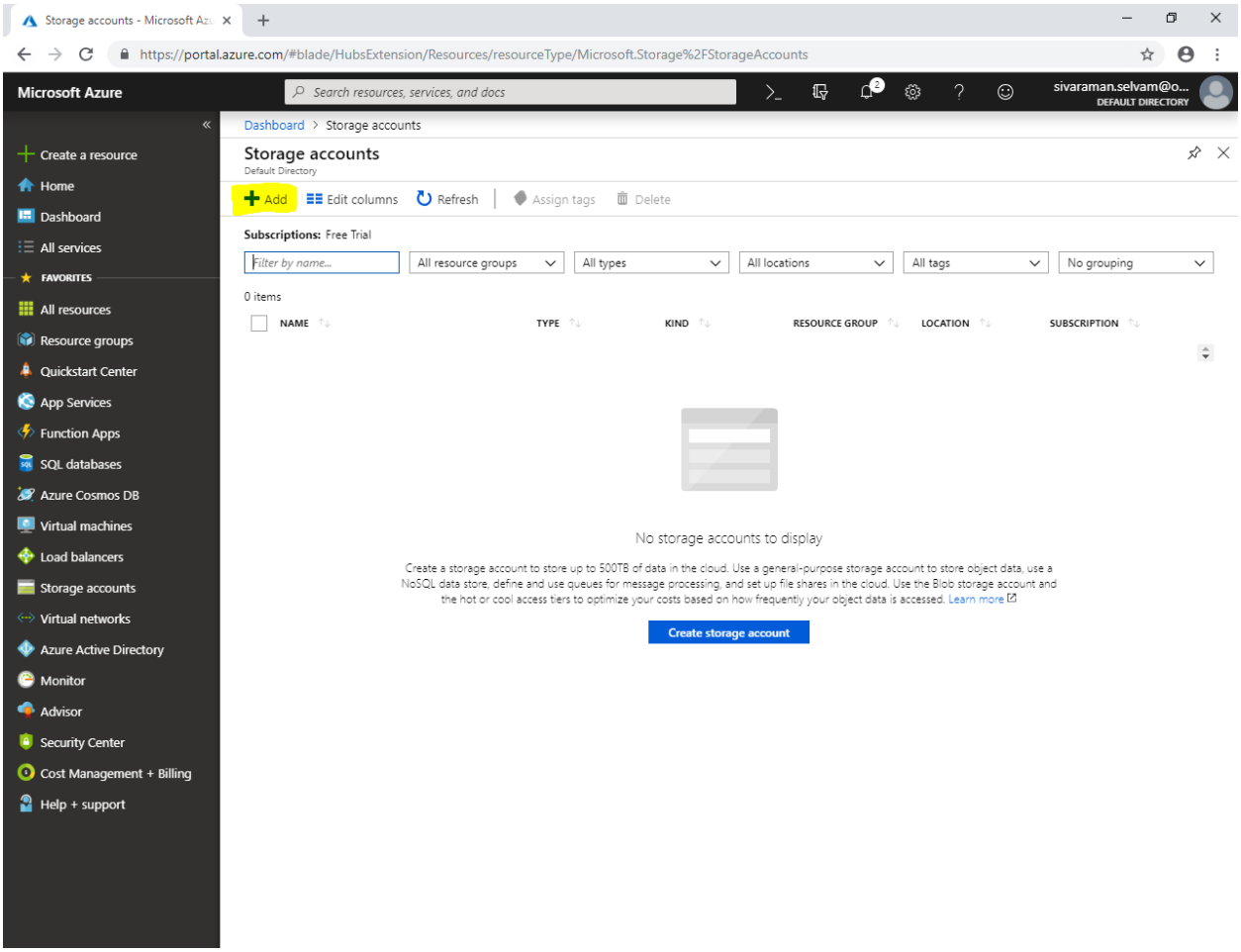


Click **“Storage accounts”** in left side panel.



In **“Storage accounts”**.

Click **“Add”**.



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

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

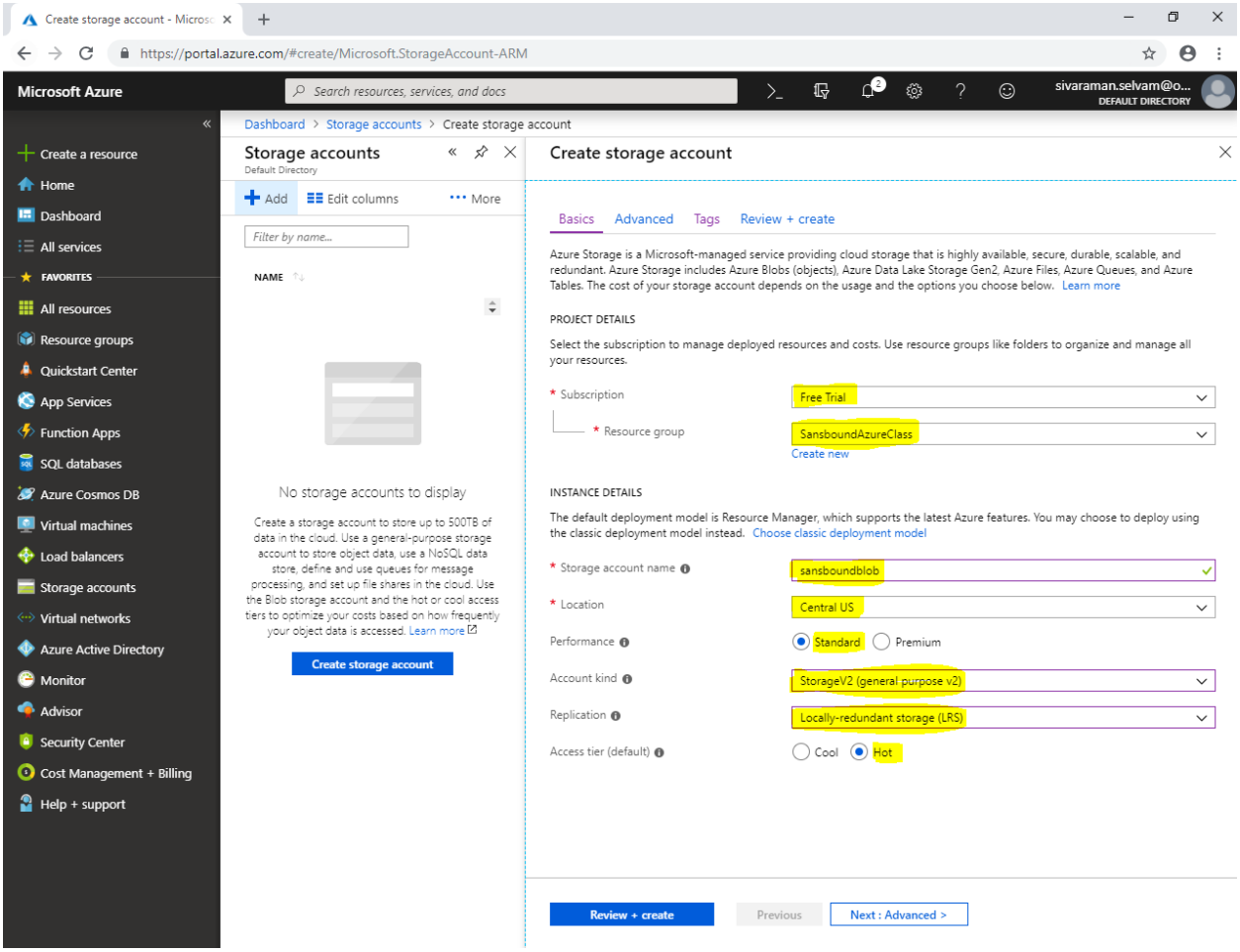
Type **“Storage account name”** as **“sansboundlrs”.**

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

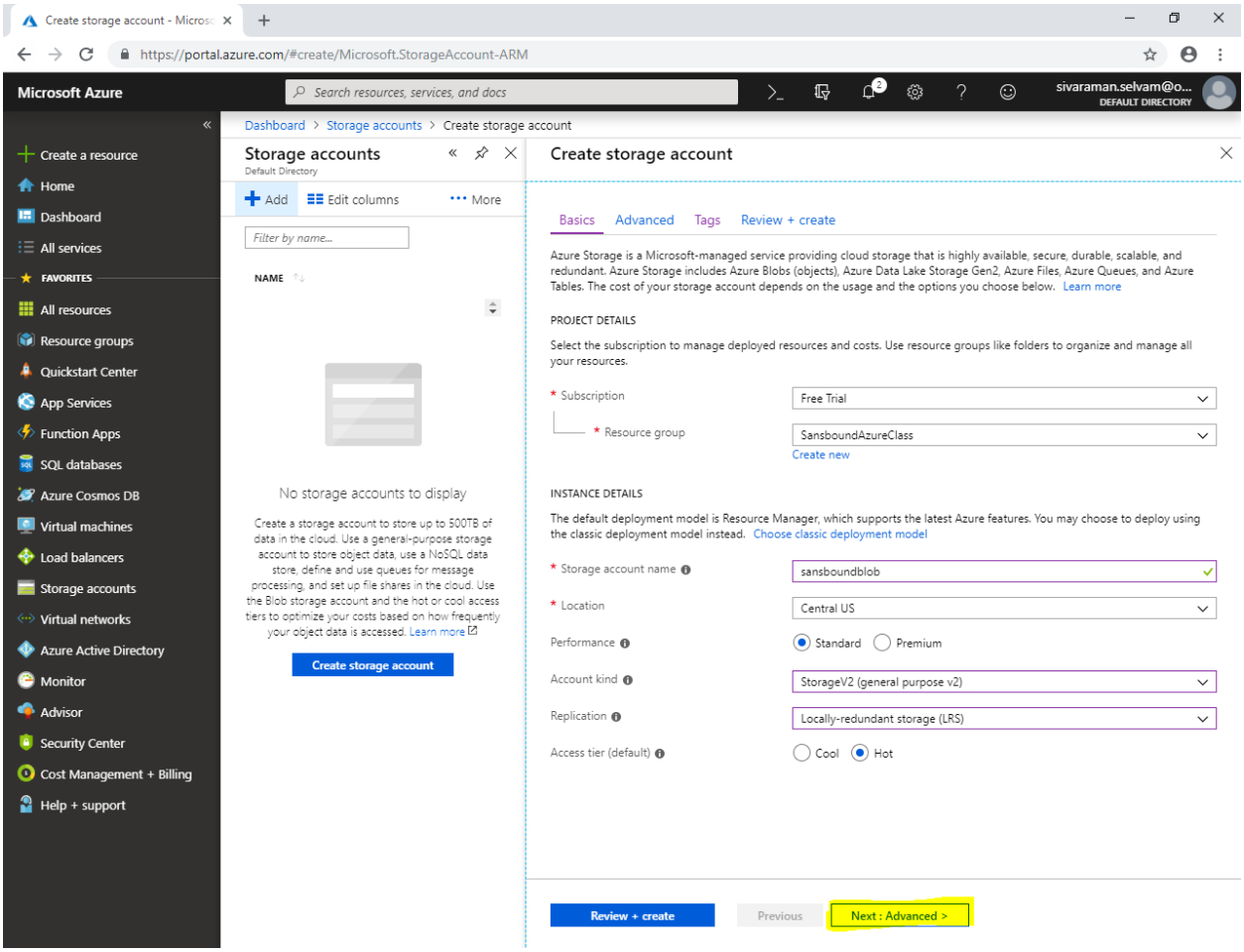
In **“Performance”** click **”Standard”**.

Select **“Account kind”** as **“Storage (general purpose v2)”**.

Select **“Replication”** as **“Locally-redundant-storage (LRS)”**.

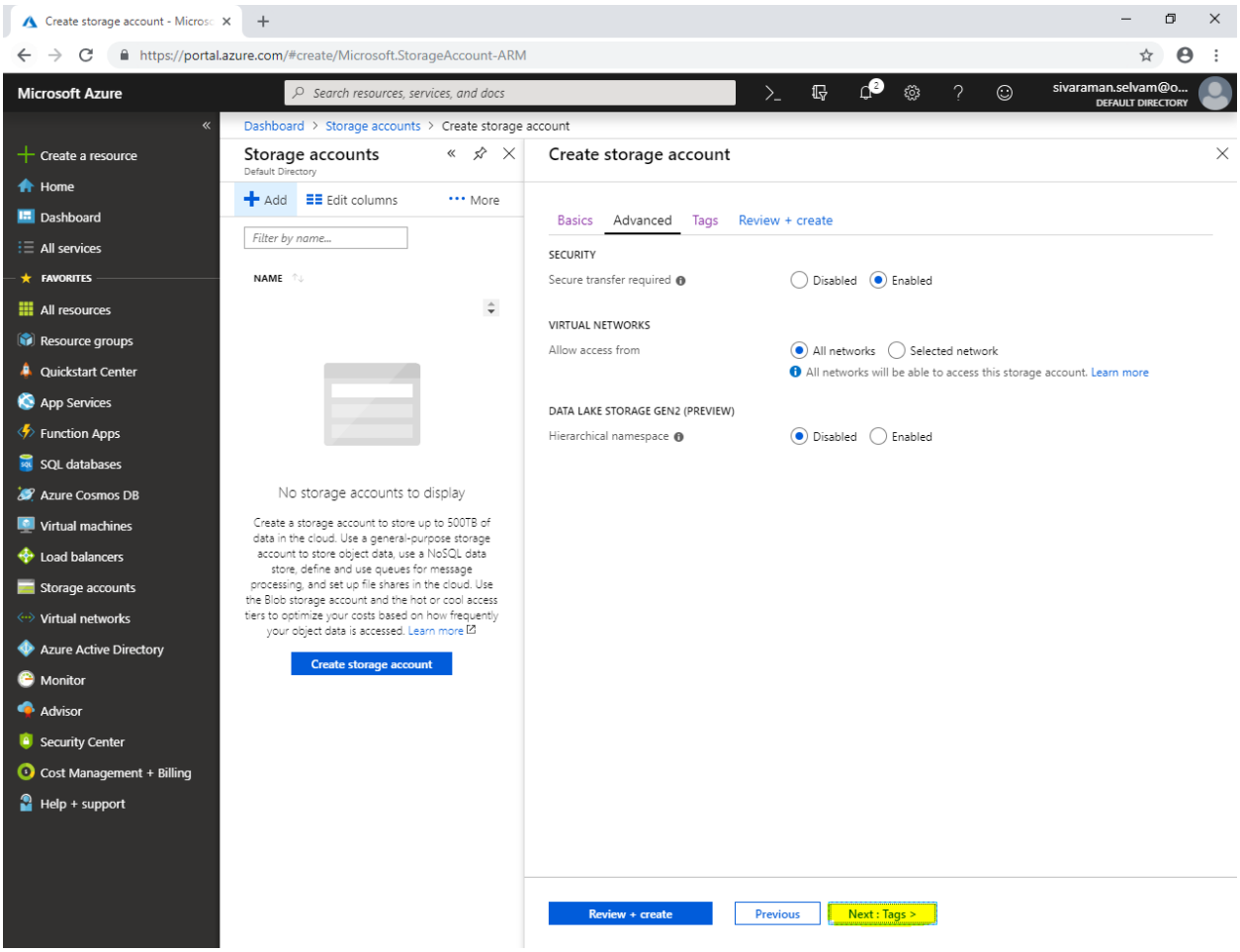


Click **“Next : Advanced >”**.



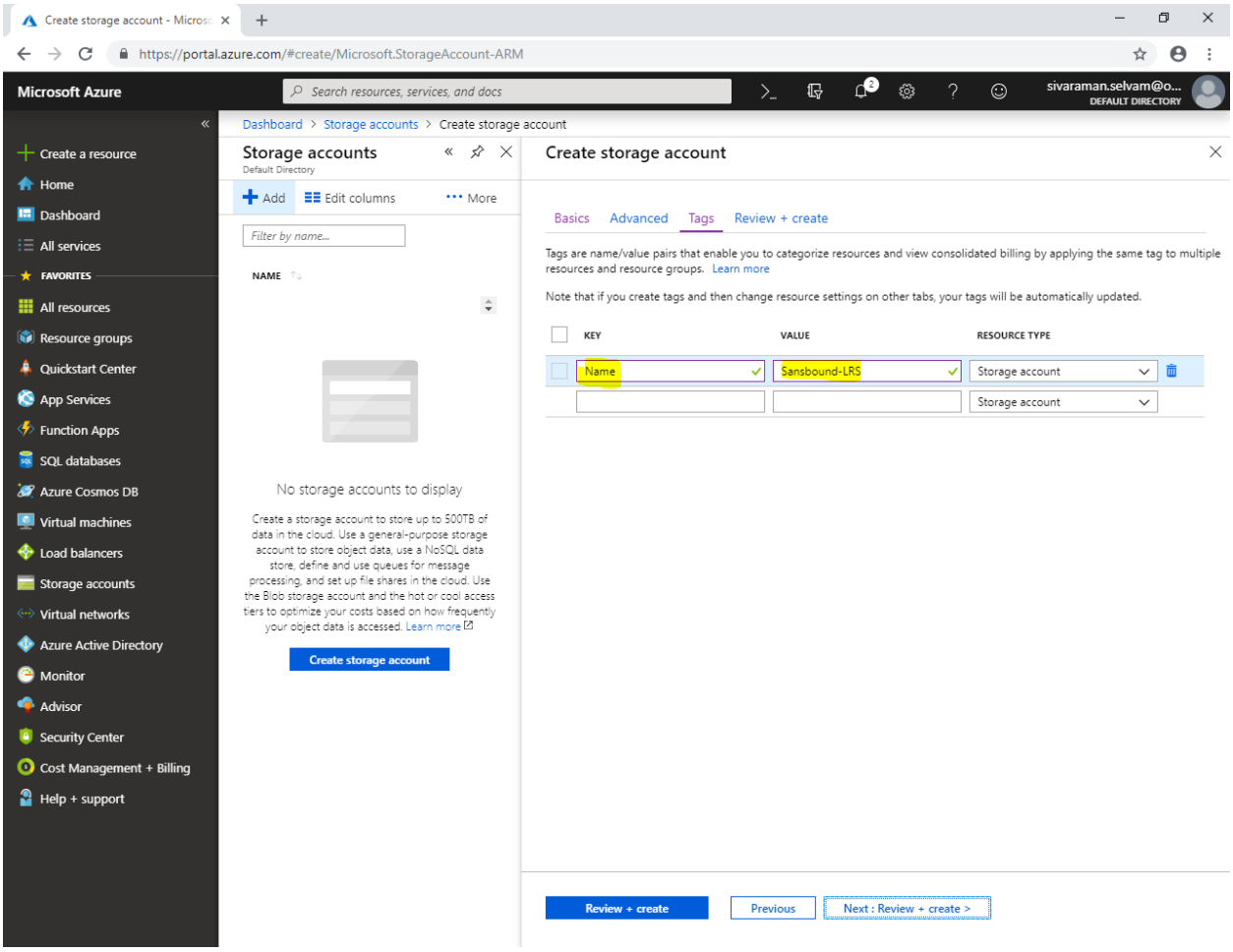
In **“Advanced”**,

Click **“Next : Tags >”** .

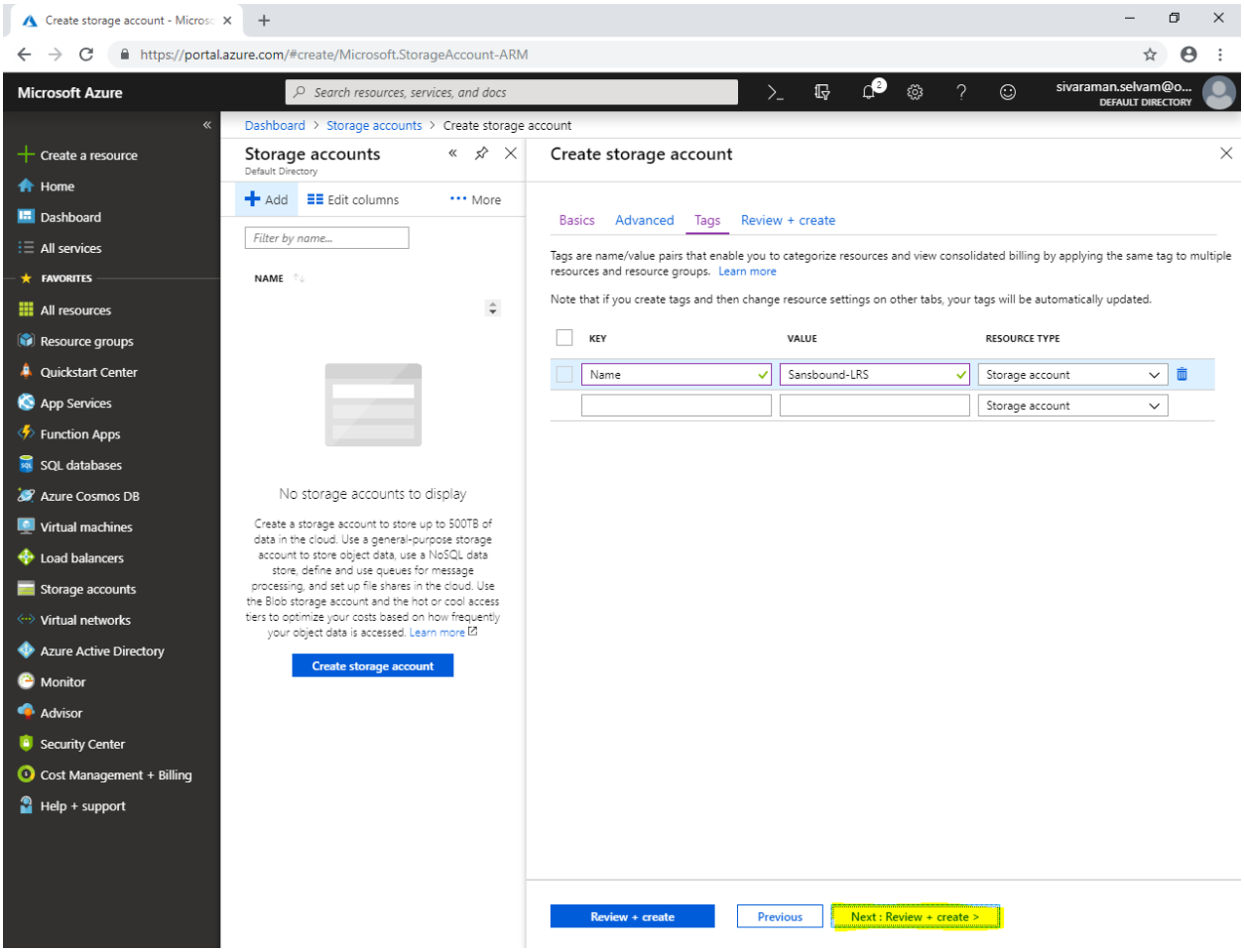


In **“Tags”**,

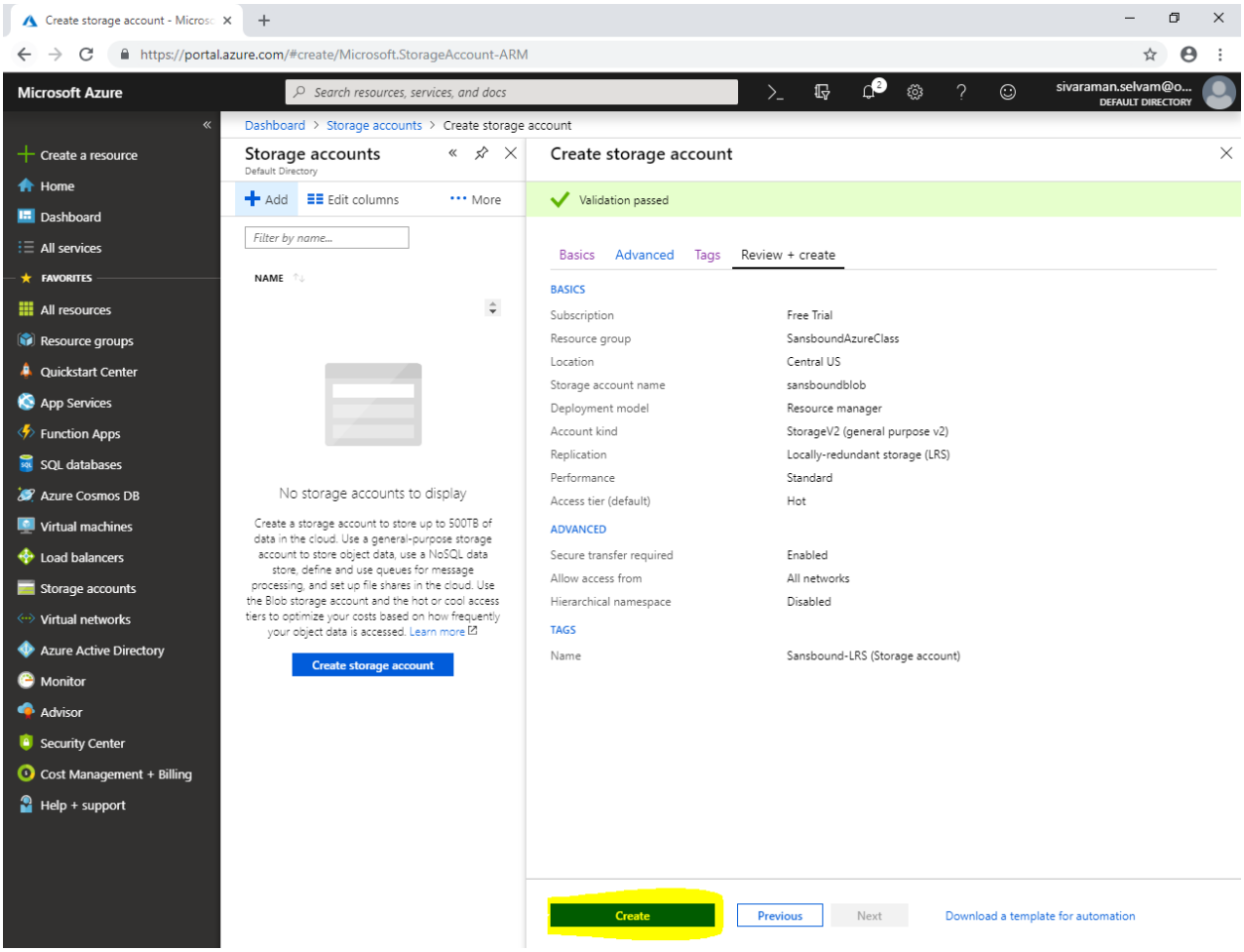
Type **“KEY”** as **“Name”** and **“VALUE”** as **“Sansbound-LRS”**.



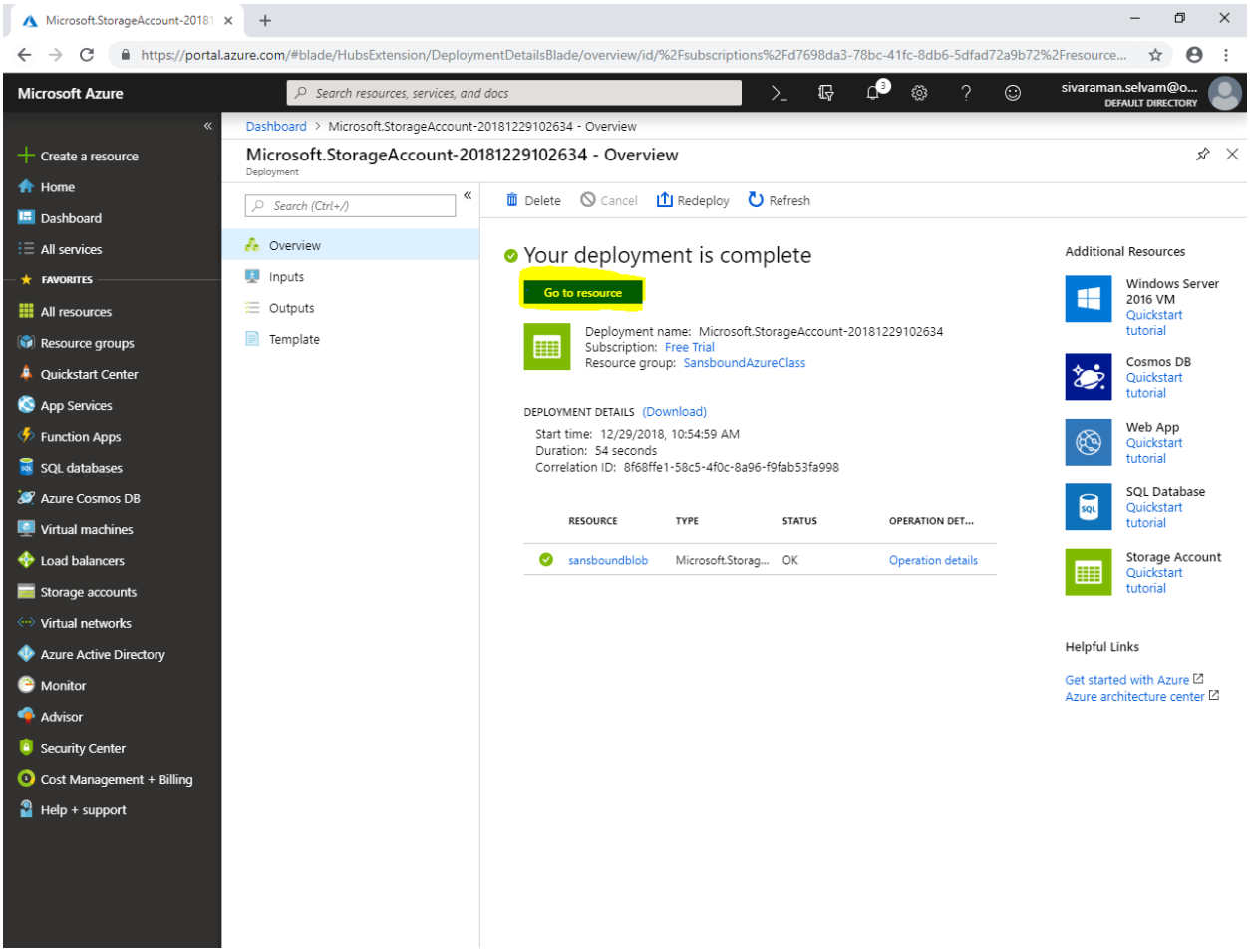
Click **“Next : Review + create”.**



Click **"Create"**.

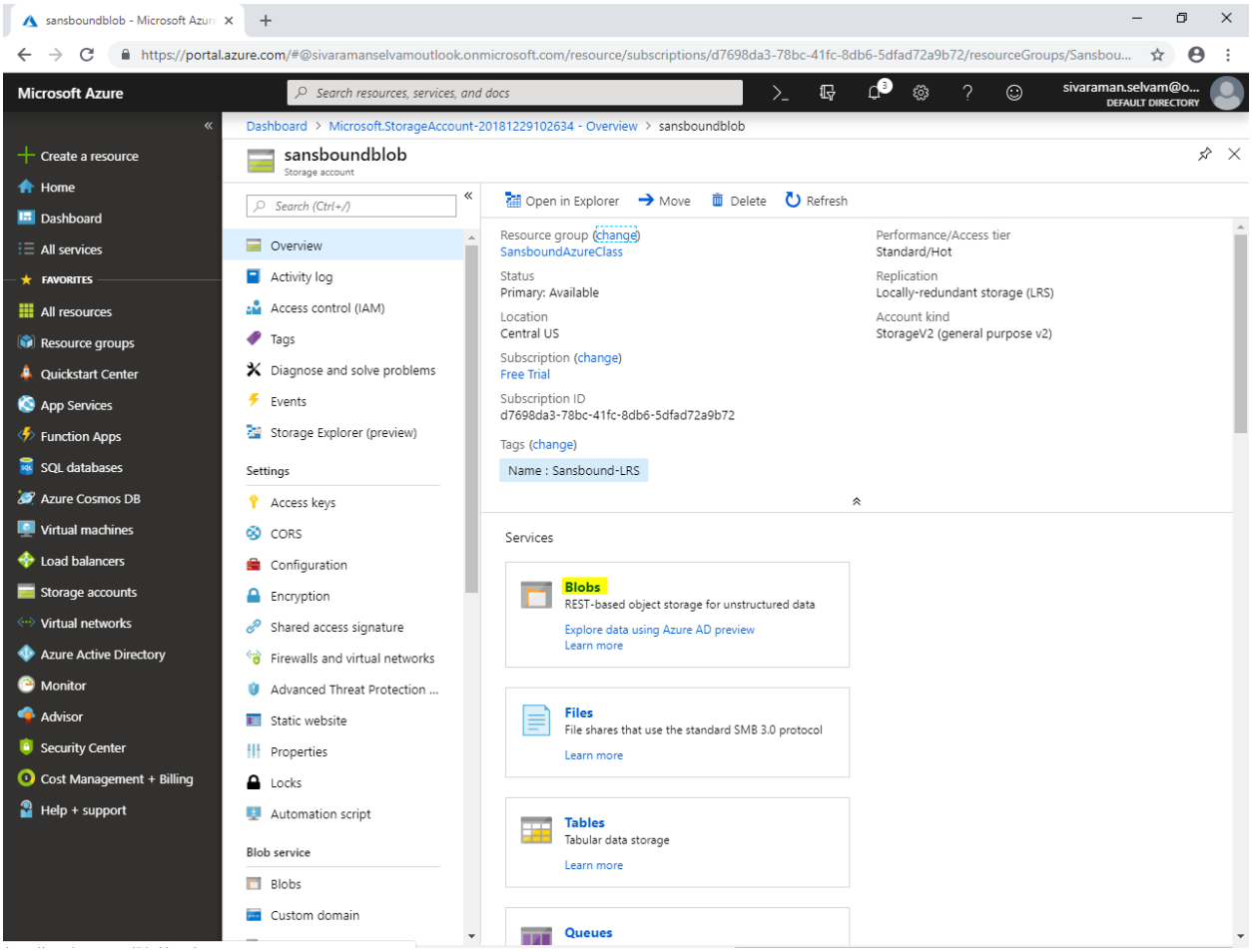


Click **"Go to resource".**



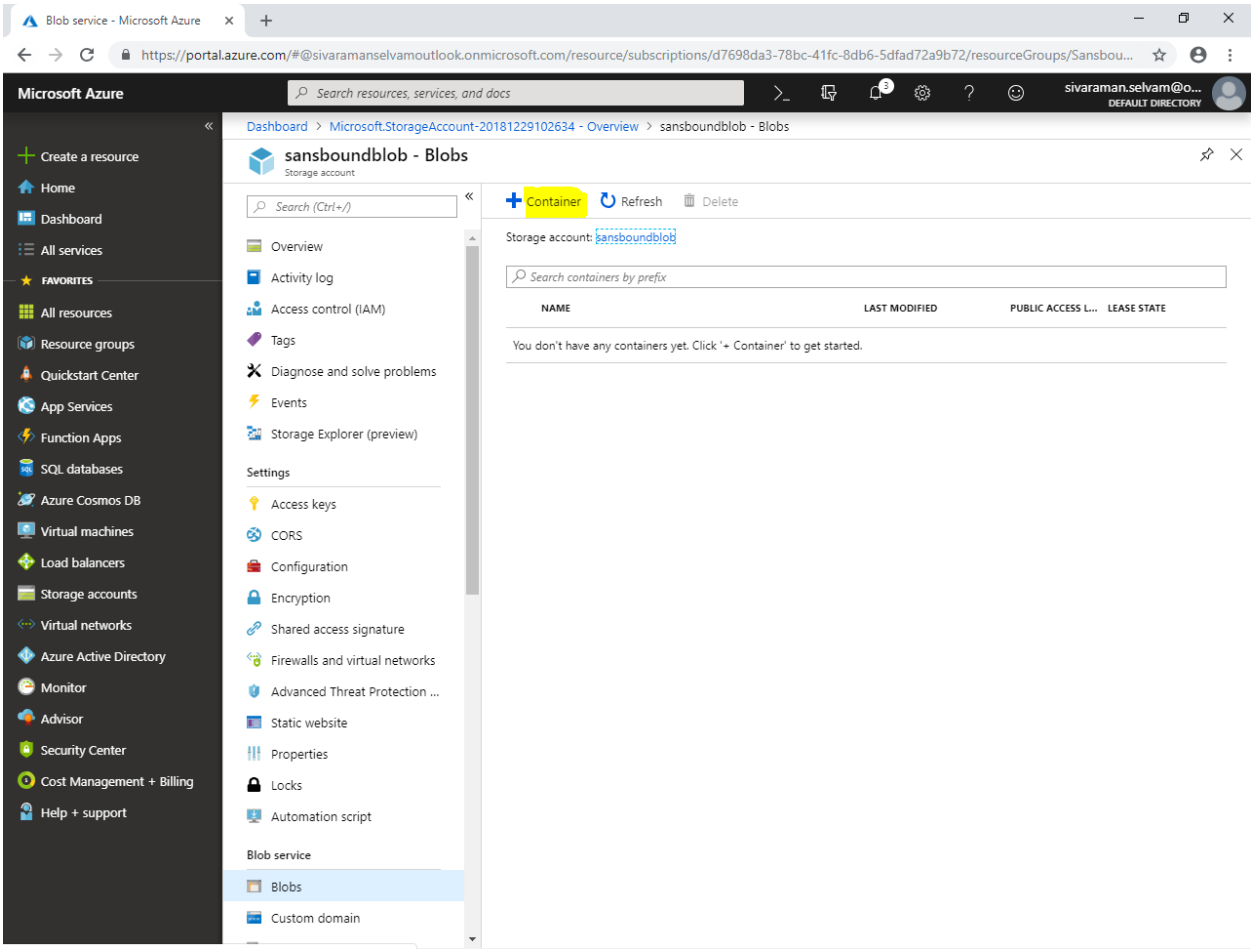
In **“sansboundblob”**.

Click **“Blobs”**.



In **“Blobs”**.

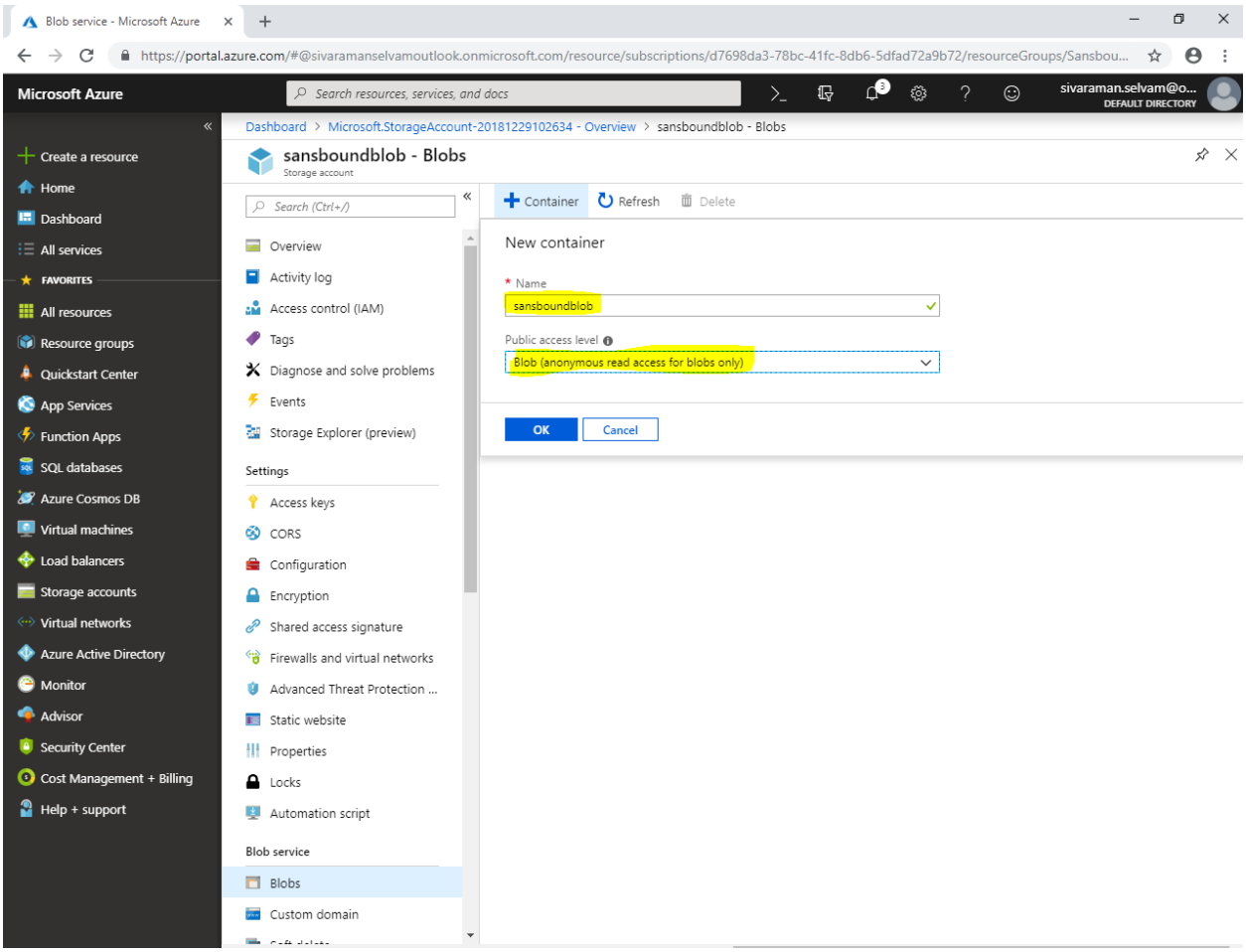
Click **“Container”** to create container.



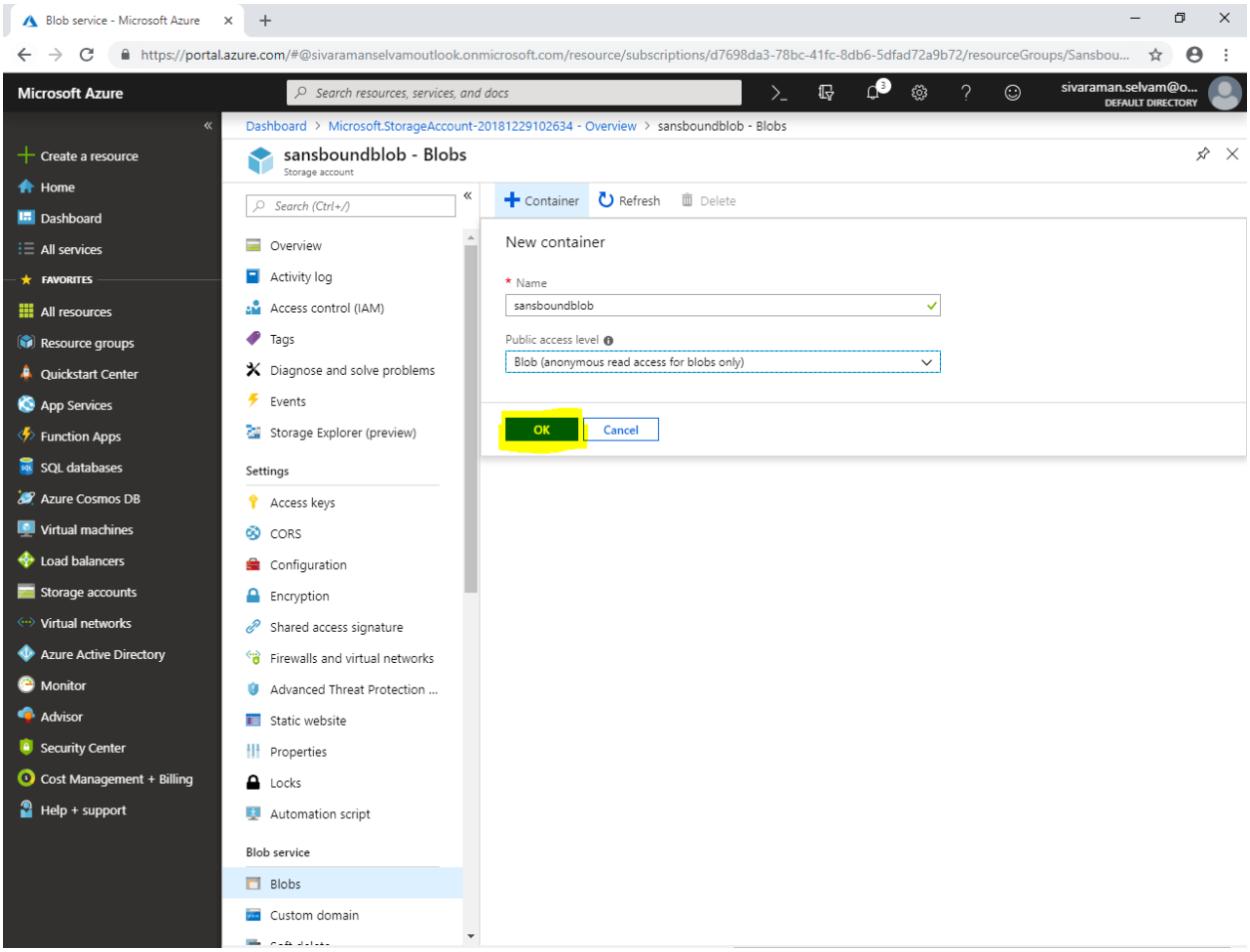
While create new container,

In **“Name”** type name as **“sansboundblob”**.

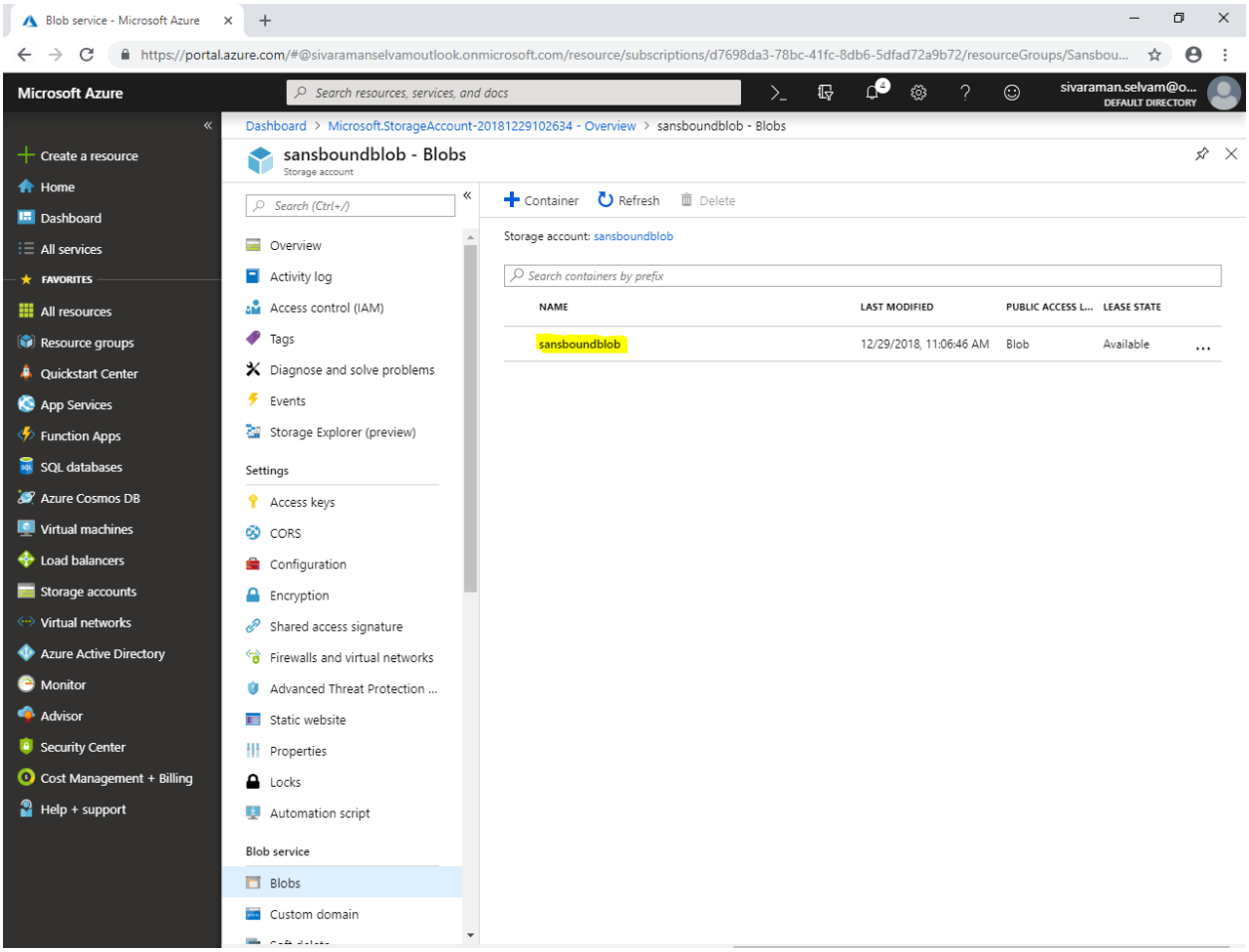
In **“Public access level”** select as **“Blob”**.



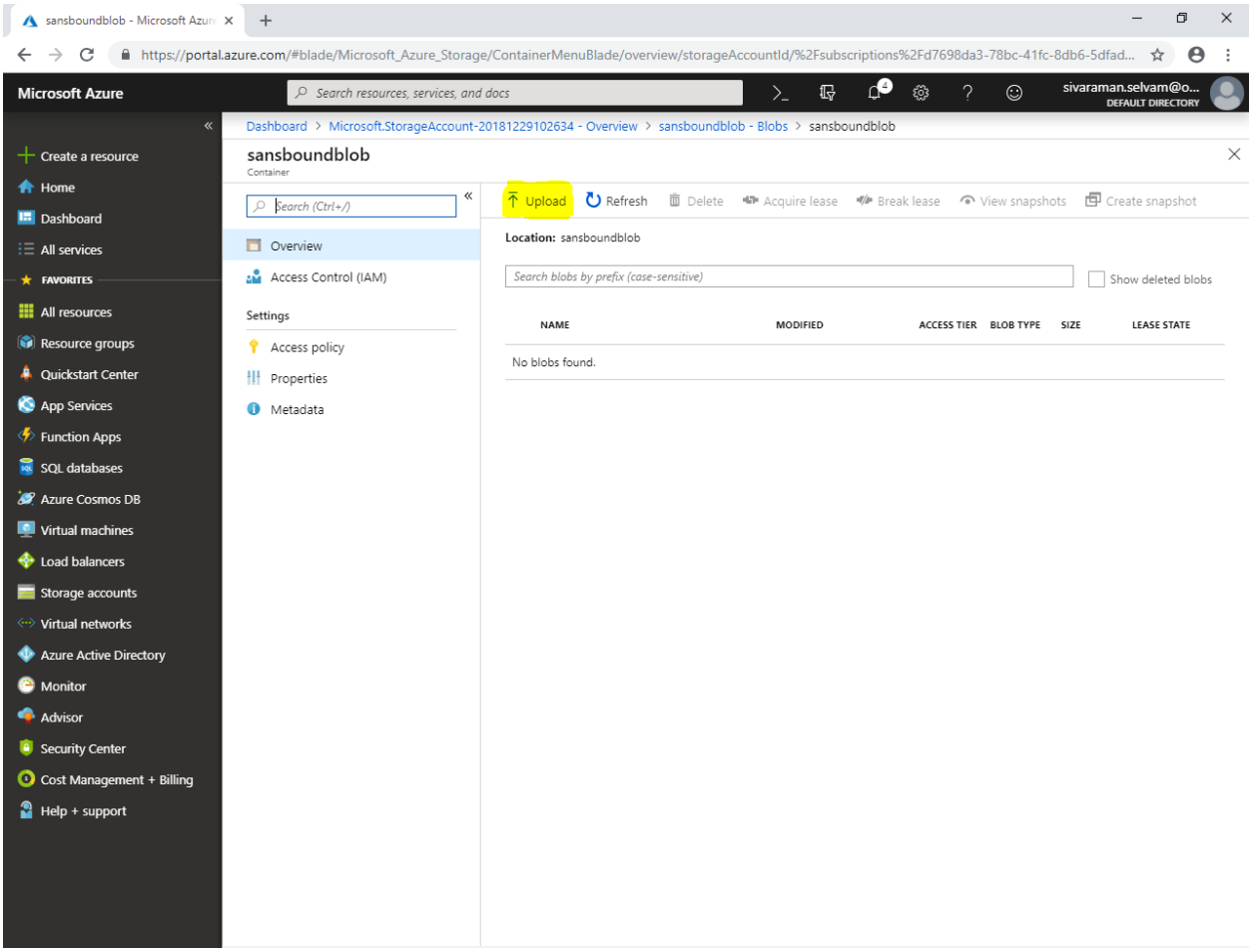
Click **"Ok".**


Blob service - Microsoft Azure
https://portal.azure.com/#@sivaramanselvamoutlook.onmicrosoft.com/resource/subscriptions/d7698da3-78bc-41fc-8db6-5dfad72a9b72/resourceGroups/Sansbou...
Microsoft Azure
Search resources, services, and docs
sivaraman.selvam@o...
DEFAULT DIRECTORY
Dashboard > Microsoft.StorageAccount-20181229102634 - Overview > sansboundblob - Blobs
sansboundblob - Blobs
Storage account
Search (Ctrl+/)
Container
Refresh
Delete
New container
* Name
sansboundblob
Public access level
Blob (anonymous read access for blobs only)
OK
Cancel
Create a resource
Home
Dashboard
All services
FAVORITES
All resources
Resource groups
Quickstart Center
App Services
Function Apps
SQL databases
Azure Cosmos DB
Virtual machines
Load balancers
Storage accounts
Virtual networks
Azure Active Directory
Monitor
Advisor
Security Center
Cost Management + Billing
Help + support
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Events
Storage Explorer (preview)
Settings
Access keys
CORS
Configuration
Encryption
Shared access signature
Firewalls and virtual networks
Advanced Threat Protection ...
Static website
Properties
Locks
Automation script
Blob service
Blobs
Custom domain

Click container named **“sansboundblob”**.

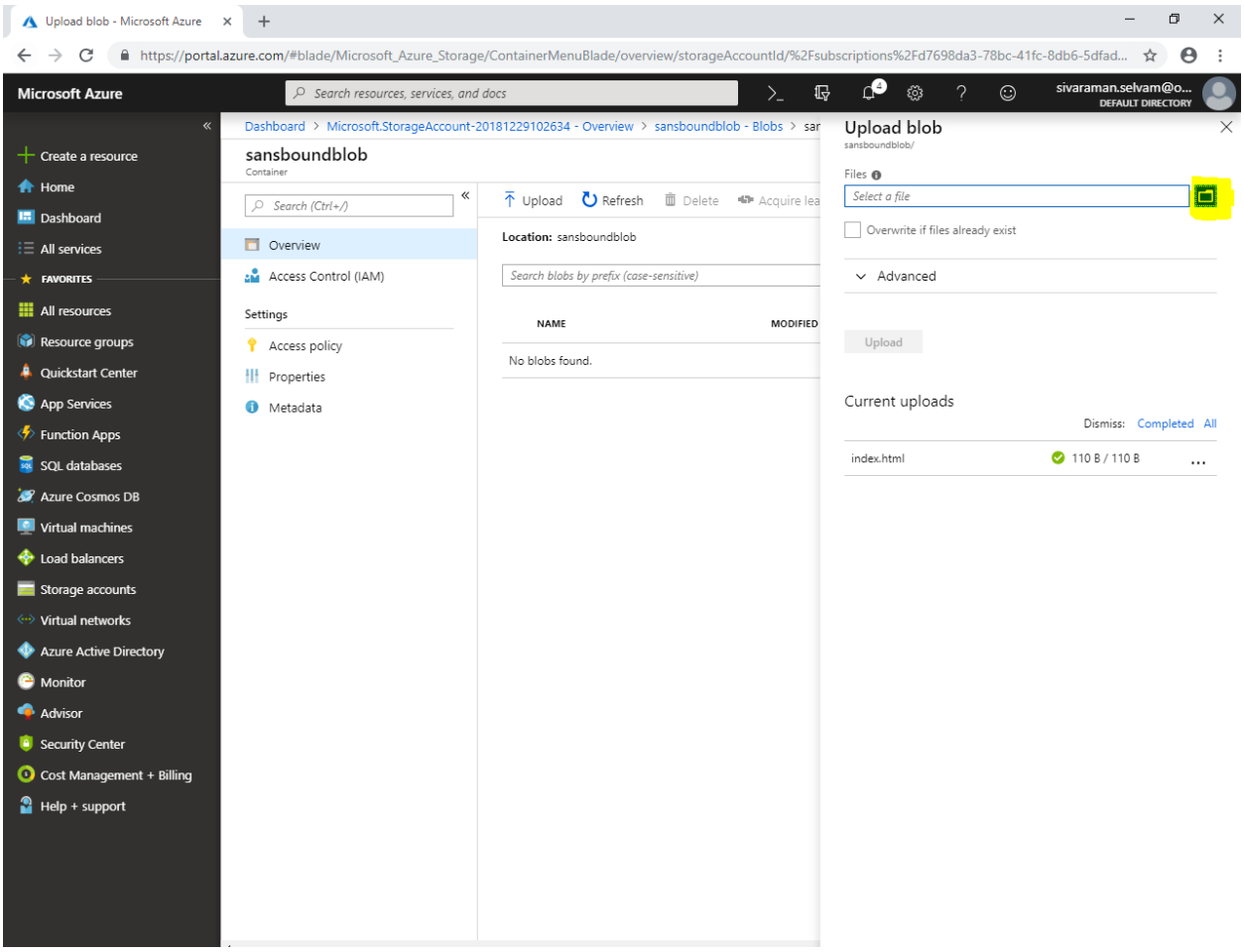


Click **“Upload”**.



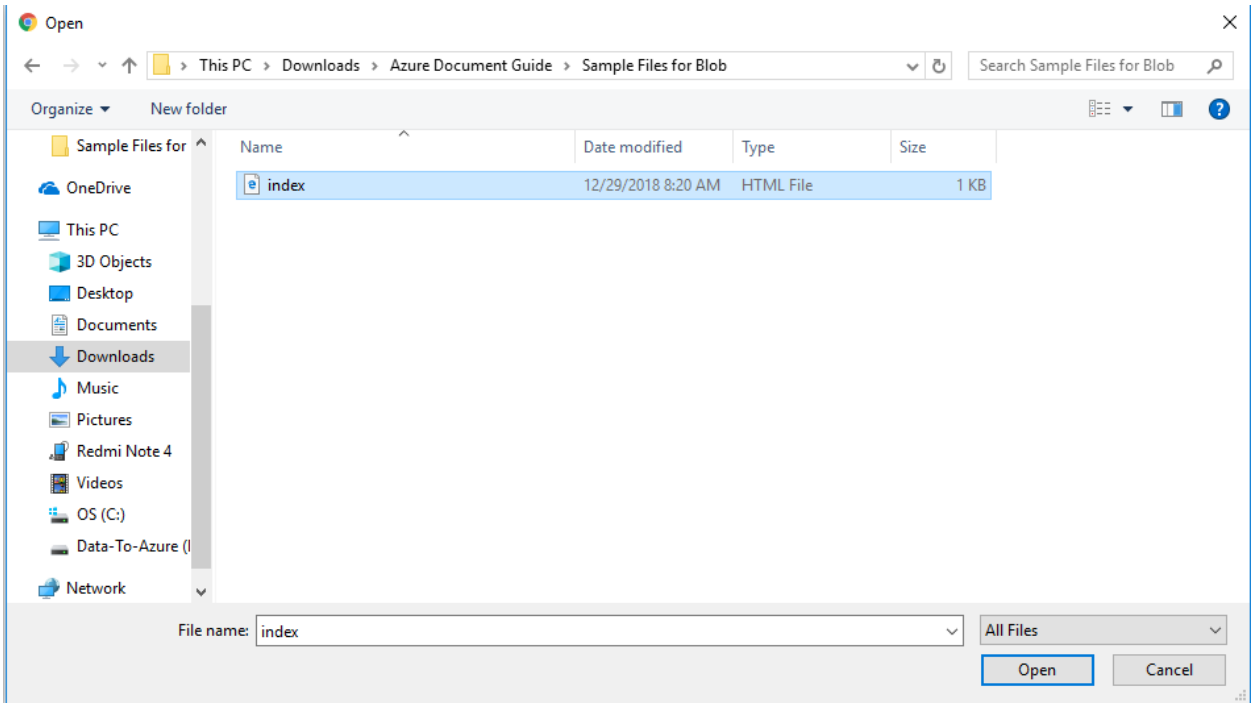
In **“Upload blob”**

Click **“Icon”**.

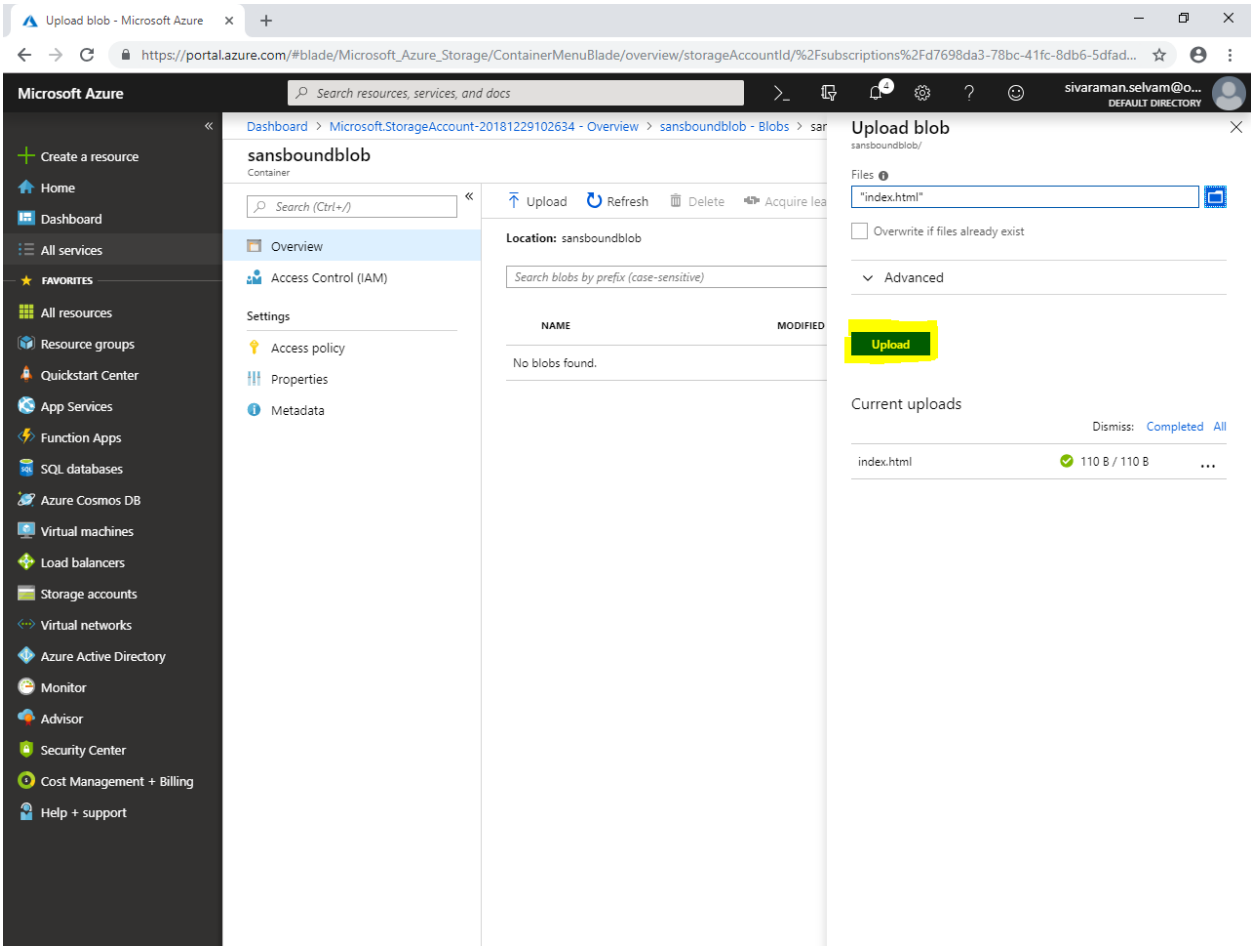


Locate the path of index.html file where you have stored and select **"index.html"** file.

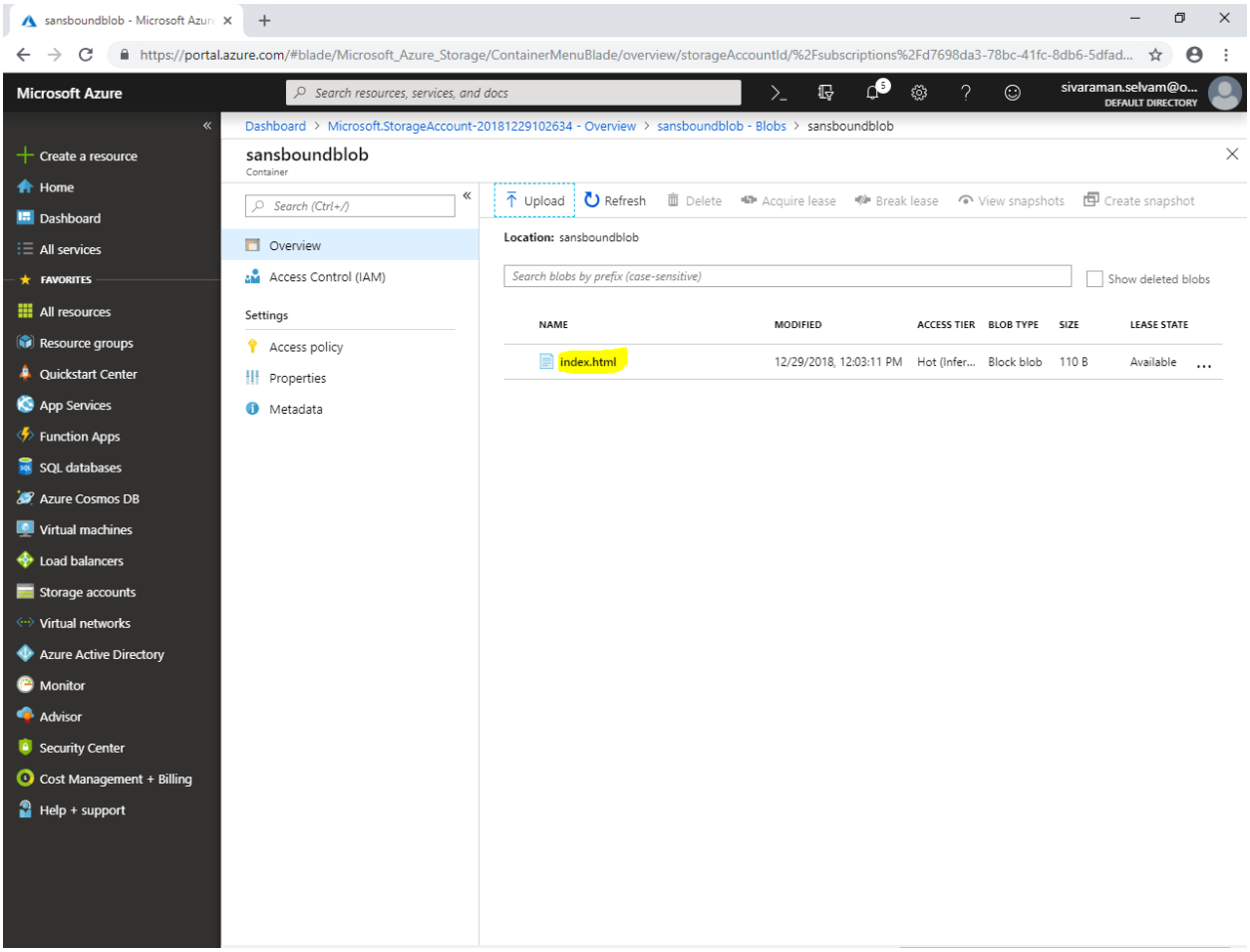
Click **"Open"**.



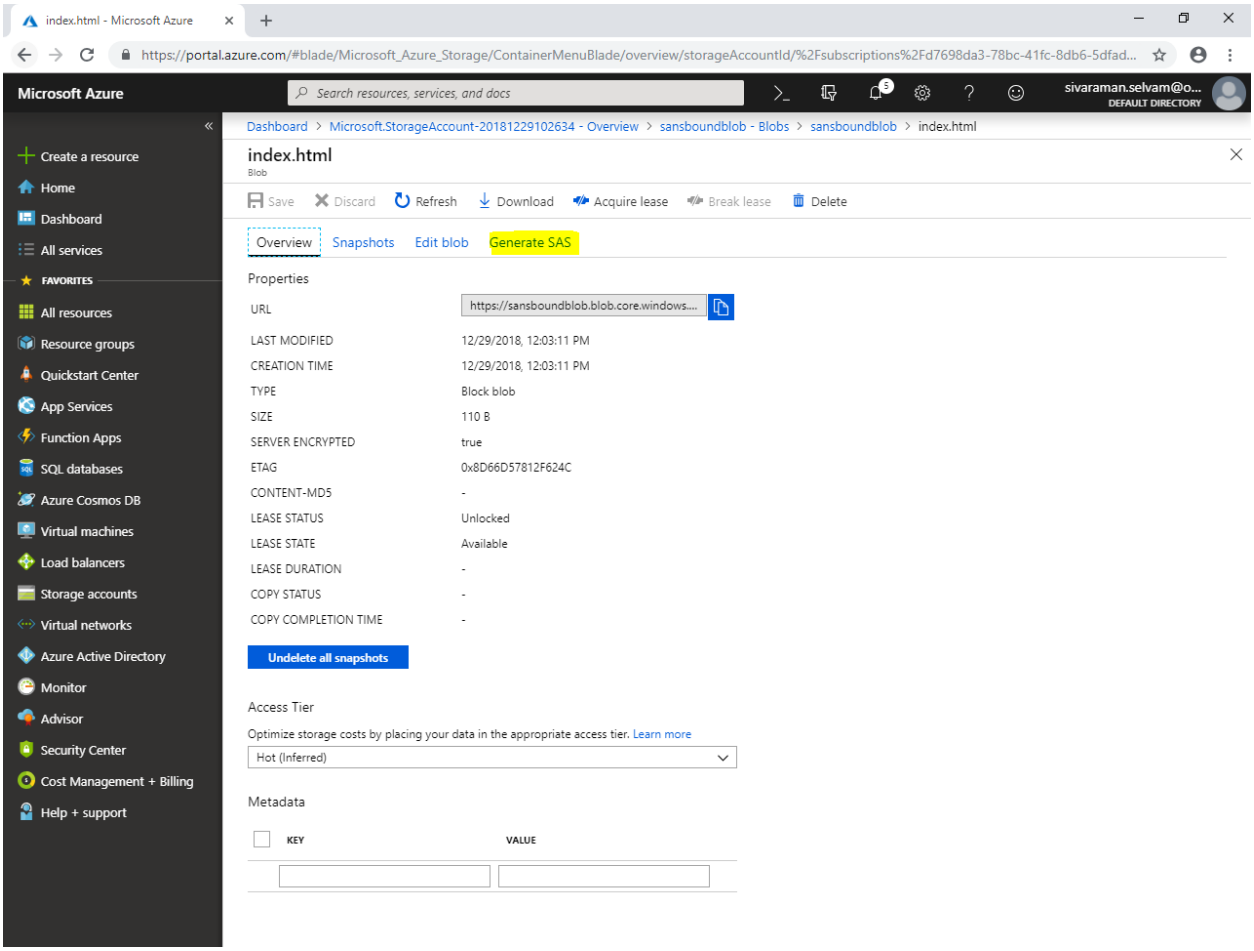
Click **"Upload"**.



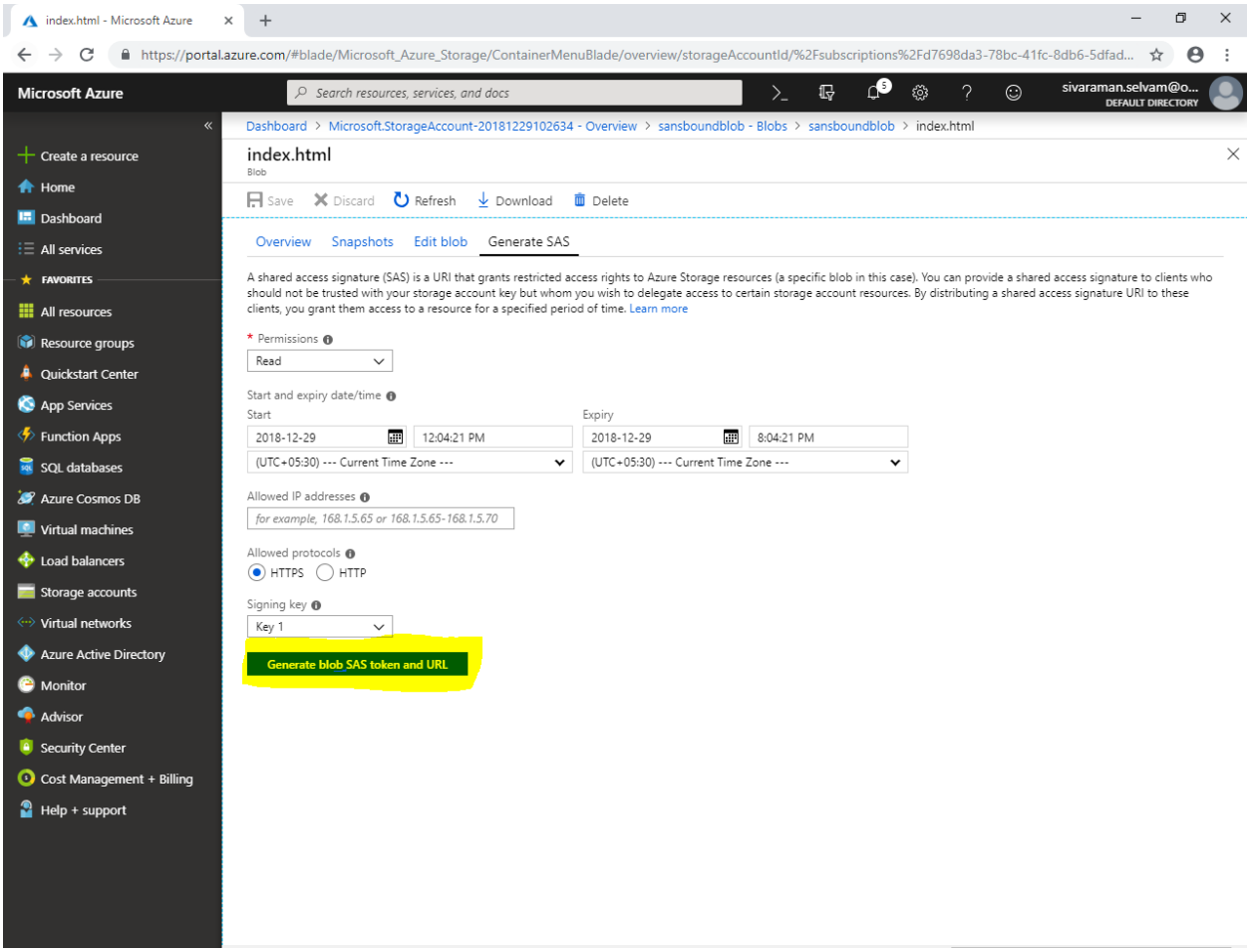
Click **“index.html”**.



Click **"Generate SAS"**.

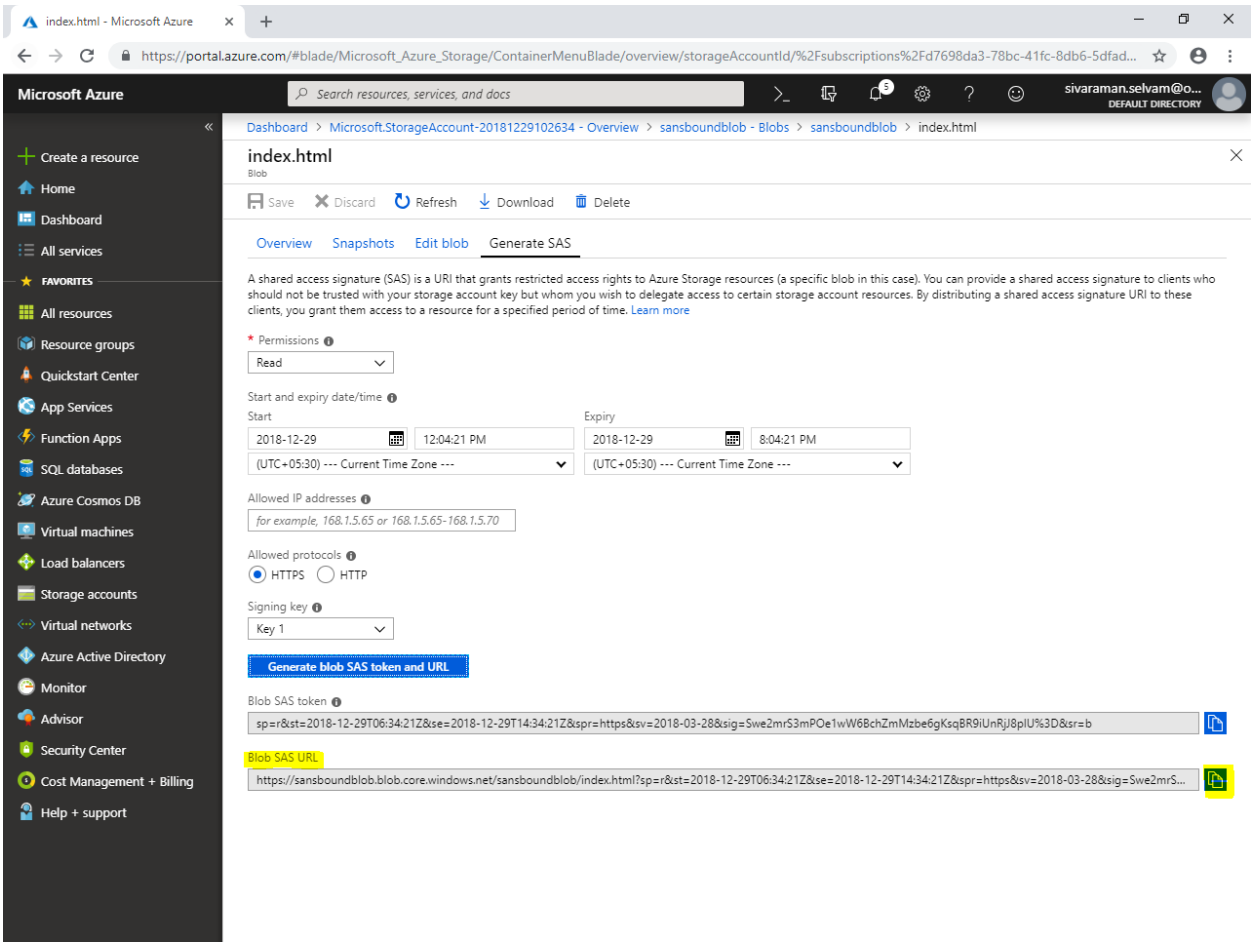


Click **"Generate blob SAS token and URL"**.

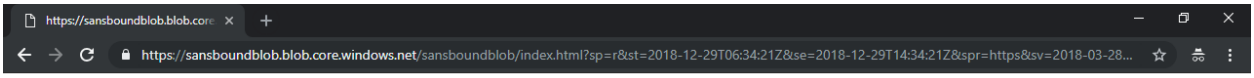


In **“Blob SAS URL”**

Click **“Icon”** to copy Blob SAS URL path.



Paste “Blob URL” in browser and press **“Enter”**



Welcome to Sansbound Azure Blob page

Note: Your data has been stored in same region in same availability zone in different storage nodes. If primary node failed, then you can able to access data from Secondary node.