

Project Name: Static Website in storage account and FileShare.

To do:Launch static website using storage accounts & create fileshare between multiple linux vm's

Description:

Azure Storage Accounts offer various features and capabilities to store and manage data in the Azure cloud. Two of these features are File Shares and Static Websites. Here's a brief description of each:

1. **File Share**:

- **What is a File Share?**: A File Share is a feature within Azure Storage Accounts that allows you to create and manage a file storage system in the cloud, similar to a traditional file server. It provides a convenient way to store and share files across multiple Azure Virtual Machines or even on-premises systems over the SMB (Server Message Block) protocol.

- **Use Cases**:

- **Shared File Storage**: You can use Azure File Shares to store configuration files, scripts, and data that need to be shared among multiple virtual machines or application instances.

- **Application Data**: Azure File Shares can serve as a shared storage location for application data, making it accessible to different parts of your application running in the cloud.

- **Data Migration**: You can use Azure File Shares for temporary storage during data migration or backup operations.

- **Access Control**: Azure File Shares support role-based access control (RBAC) and shared access signatures (SAS) to control who can access and modify the files stored in the share.

- **Scalability**: Azure File Shares can scale up to handle large amounts of data and can be accessed from anywhere with an internet connection.

2. **Static Website**:

- **What is a Static Website?**: A Static Website is a feature within Azure Storage Accounts that enables you to host static web content such as HTML, CSS, JavaScript, and media files in the form of a website. This feature simplifies web hosting for simple websites that do not require server-side processing.

- **Use Cases**:

- **Personal Websites**: You can use Azure Storage's Static Website feature to host personal blogs, portfolios, or small informational websites.

- **Documentation Sites**: Host static documentation or help sites.

- **Single-Page Apps**: For single-page applications (SPAs) that primarily run on the client-side, Azure Storage can serve as a cost-effective hosting solution.

- **Scalability**: Azure Storage Accounts provide high availability and scalability for static websites, making it suitable for websites with varying traffic loads.

- **Custom Domain**: You can configure a custom domain name for your static website hosted in Azure Storage, allowing you to use your own domain for branding.

- **HTTPS Support**: Azure Storage static websites can be accessed via HTTPS by default, providing secure access to your static content.

- **Serverless**: Since static websites don't require server-side processing or infrastructure management, they are cost-effective and require minimal maintenance.

In summary, Azure Storage Accounts offer versatile features like File Shares for shared file storage needs and Static Websites for hosting simple, serverless, and cost-effective web content. These features are part of Azure's broader ecosystem for data storage and web hosting, making it easier for developers and businesses to manage and deploy data and web applications in the cloud.

Steps to deploy a static website in storage account:

Step 1: Create a storage account.

The screenshot shows the Azure portal interface for a storage account named 'samstorage01'. The main pane displays the storage account's properties, including its resource group (rg1), location (France Central), and subscription (Azure for Students). The 'Properties' tab is selected, showing details like performance (Standard), replication (Locally-redundant storage (LRS)), and account kind (StorageV2 (general purpose v2)). The 'Blob service' section shows settings for hierarchical namespace, default access tier (Hot), blob anonymous access, blob soft delete, container soft delete, versioning, change feed, NFS v3, and allow cross-tenant replication. The 'Security' section includes options for require secure transfer for REST API operations, storage account key access, minimum TLS version (Version 1.2), and infrastructure encryption. The 'Networking' section indicates allow access from all networks and no private endpoint connections. The left sidebar lists other management options like Activity log, Tags, Diagnose and solve problems, Access Control (IAM), Data migration, Events, Storage browser, and Storage Mover. The bottom status bar shows the date (09-09-2023), time (11:50), and weather (33°C, Partly sunny).

The screenshot shows the Azure portal interface for managing storage accounts. The top navigation bar includes a search bar and a user profile. The main content area is titled 'Storage accounts' and shows a single record: 'samstorage01'. The table includes columns for Name, Type, Kind, Resource group, Location, and Subscription. The 'samstorage01' row is highlighted. The bottom status bar shows the date (09-09-2023), time (11:50), and weather (33°C, Partly sunny).

The screenshot shows the Windows taskbar at the bottom of the screen. It includes icons for the Start button, File Explorer, Microsoft Edge, Mail, and other system icons. The Azure portal icon is visible in the taskbar, indicating it is currently running or pinned. The bottom status bar shows the date (09-09-2023), time (11:50), and weather (33°C, Partly sunny).

Step 2 : In the navigation click on Static website and create it.

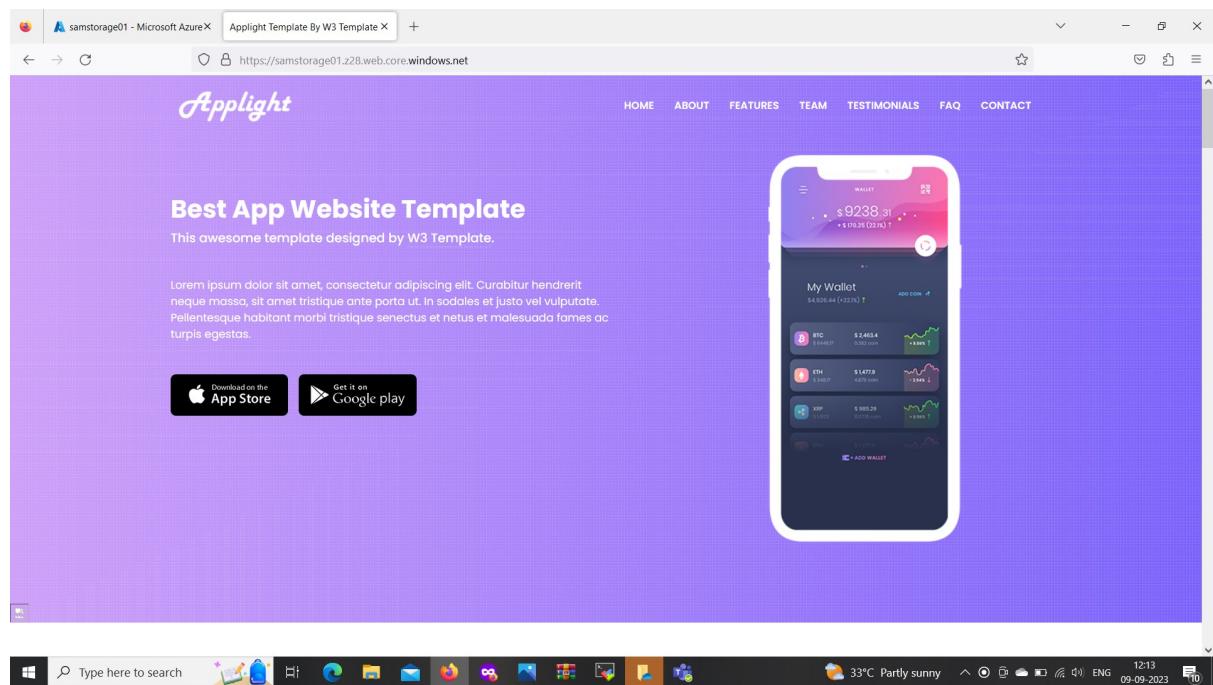
The screenshot shows the Microsoft Azure portal interface. The user is navigating through the 'Storage accounts' section under 'Storage accounts'. They have selected the 'samstorage01' account. On the left sidebar, under 'Static website', they have chosen the 'Enabled' option. The main pane displays configuration options for the static website, including the primary endpoint set to <https://samstorage01.z28.web.core.windows.net/>, the index document name set to 'index.html', and an error document path. The status bar at the bottom indicates the date and time as 09-09-2023 12:13.

Step 3 : Upload the website files in \$web folder.

The screenshot shows the Microsoft Azure portal interface. The user is viewing the contents of the '\$web' container within the 'samstorage01' storage account. The left sidebar shows various settings like 'Overview', 'Diagnose and solve problems', and 'Access Control (IAM)'. The main pane lists the uploaded files: 'index.html', 'style.css', 'License.txt', 'css', and 'images'. Each file entry includes details such as name, modified date, access tier, archive status, blob type, size, and lease state. The status bar at the bottom indicates the date and time as 09-09-2023 12:13.

Save the folder.

Step 4: Now copy the link present under Primary endpoint and paste it in the browser.



Steps to create a fileshare.

Step 1: Create a storage account with a unique name.

The screenshot shows the Azure Storage Account Overview page for 'samstorage01'. The left sidebar includes sections for Overview, Activity log, Tags, Diagnose and solve problems, Access Control (IAM), Data migration, Events, Storage browser, and Storage Mover. Under Data storage, there are options for Containers, File shares, Queues, and Tables. The main content area displays the following details:

Setting	Value
Resource group (move)	rg1
Location	France Central
Subscription (move)	Azure for Students
Subscription ID	2dc11c06-dff8-4f53-ae3a-f17d684e9b52
Disk state	Available
Tags (edit)	: Add tags
Performance	Standard
Replication	Locally-redundant storage (LRS)
Account kind	StorageV2 (general purpose v2)
Provisioning state	Succeeded
Created	9/9/2023, 11:50:12 am

The Properties tab is selected, showing Blob service and Security settings. The Blob service section includes options like Hierarchical namespace, Default access tier (Hot), Blob anonymous access (Disabled), Blob soft delete (Enabled (7 days)), Container soft delete (Enabled (7 days)), Versioning (Disabled), Change feed (Disabled), NFS v3 (Disabled), and Allow cross-tenant replication (Disabled). The Security section includes Require secure transfer for REST API operations (Enabled), Storage account key access (Enabled), Minimum TLS version (Version 1.2), and Infrastructure encryption (Disabled). The Networking section shows Allow access from (All networks) and Number of private endpoint connections (0). The bottom status bar indicates it's 33°C Partly sunny at 11:50 on 09-09-2023.

Step 2: Create a fileshare by clicking on fileshare in the left navigation window.

The screenshot shows the Azure File Share Overview page for 'fs01'. The left sidebar includes sections for Overview, Connect, Diagnose and solve problems, Access Control (IAM), Browse, Operations, Snapshots, and Backup. The main content area displays the following details:

Setting	Value
Storage account	samstorage01
Resource group (move)	rg1
Location	France Central
Subscription (move)	Azure for Students
Subscription ID	2dc11c06-dff8-4f53-ae3a-f17d684e9b52
Share URL	https://samstorage01.file.core.windows.net/fs01
Redundancy	Locally-redundant storage (LRS)
Configuration modified	9/9/2023, 11:52:37 am

The Properties tab is selected, showing Size, Feature status, Performance, Active Directory, and SMB protocol settings. The Size section includes Maximum capacity (5 TiB), Used capacity (0 B), and Tier (Transaction optimized). The Feature status section includes Soft delete (7 days) and Large file shares (Disabled). The Performance section includes Maximum IO/s (1000), Ingress rate (60 MiB / s), and Egress rate (60 MiB / s). The Active Directory section shows Directory service (Not configured) and Domain (-). The SMB protocol settings section shows Maximum compatibility (Maximum compatibility). The bottom status bar indicates it's 33°C Partly sunny at 11:52 on 09-09-2023.

Step 3: Create 3 Linux Servers.

The screenshot shows the Microsoft Azure portal interface for a virtual machine named "PublicServer1". The main pane displays the "Essentials" section with details such as Resource group (rg1), Status (Running), Location (West US 3 (Zone 1)), Subscription (Azure for Students), and Operating system (Linux (ubuntu 20.04)). The "Properties" tab is selected, showing the VM configuration including Computer name (PublicServer1), Operating system (Linux (ubuntu 20.04)), and Image publisher (canonical). The "Networking" tab shows the VM has a Public IP address and is connected to a Virtual network/subnet. The "Size" tab indicates it's a Standard D2s v3 (2 vcpus, 8 GiB memory) VM. The left sidebar contains navigation links like Overview, Activity log, Tags, and Diagnose and solve problems. The bottom taskbar shows the Windows Start button, search bar, and various pinned icons.

The screenshot shows the Microsoft Azure portal interface for a virtual machine named "PublicServer2". The main pane displays the "Essentials" section with details such as Resource group (rg2), Status (Running), Location (East US 2 (Zone 1)), Subscription (Azure for Students), and Operating system (Linux (ubuntu 20.04)). The "Properties" tab is selected, showing the VM configuration including Computer name (PublicServer2), Operating system (Linux (ubuntu 20.04)), and Image publisher (canonical). The "Networking" tab shows the VM has a Public IP address and is connected to a Virtual network/subnet. The "Size" tab indicates it's a Standard D2s v3 (2 vcpus, 8 GiB memory) VM. The left sidebar contains navigation links like Overview, Activity log, Tags, and Diagnose and solve problems. The bottom taskbar shows the Windows Start button, search bar, and various pinned icons.

The screenshot shows the Microsoft Azure portal interface. The main page displays the details of a virtual machine named "PublicServer3". The "Essentials" section provides key information such as the resource group (rg3), status (Running), location (France Central), subscription (Azure for Students), and VM ID (2dc11c06-dff8-4f53-ae3a-f17d684e9b52). The "Properties" tab is selected, showing the virtual machine's configuration (Computer name: PublicServer3, Operating system: Linux (ubuntu 20.04), Image publisher: canonical, etc.) and networking details (Public IP address: 98.66.137.237, Virtual network/subnet: vnet3/subnet31). The portal header includes the user's email (21A91A6121@aec.edu.in) and the date (09-09-2023).

Step 4: Go to the fileshare created and click on connect and copy the script according to the os used.

The screenshot shows the Microsoft Azure portal interface, specifically the "File shares" section. It displays the details of an SMB file share named "fs01". The "Connect" section contains a command-line script for connecting from a Linux computer:

```
sudo mkdir /mnt/fs01
if [ ! -d "/etc/smbcredentials" ]; then
    sudo mkdir /etc/smbcredentials
fi
if [ ! -f "/etc/smbcredentials/samstorage01.cred" ]; then
    sudo bash -c 'echo "username=samstorage01" > /etc/smbcredentials/samstorage01.cred'
    sudo bash -c 'echo "password=HcFubdQM/nStIlgCplHeDitjxSeauCZ+ASTC2/3Uw==" >> /etc/smbcredentials/samstorage01.cred'
fi
sudo chmod 600 /etc/smbcredentials/samstorage01.cred
sudo bash -c 'echo "/samstorage01/file.core.windows.net/fs01 /mnt/fs01 cifs
nofail,credentials=/etc/smbcredentials
/samstorage01.cred.dir_mode=0777,file_mode=0777,serverino,nosharesock,actimeo=30" >> /etc/fstab'
```

A tooltip indicates that the copied text has been copied. The portal header includes the user's email (21A91A6121@aec.edu.in) and the date (09-09-2023).

Step 5: Paste the script in the 3 linux servers you can see the created fileshare in the server.

The screenshot shows a MobaXterm interface with two terminal sessions. The left terminal window (root@PublicServer2) displays a file tree under the /mnt directory. The right terminal window (root@PublicServer1) shows the result of running 'ls' in the root directory, which includes 'DATALOSS WARNING README.txt' and 'fs01 lost+found'. A sidebar on the left side of the interface lists files in the current directory, including '.Name', '.cache', '.ssh', '.bash_history', '.bash_logout', '.profile', '.sudo_as_admin_successful', and '.xauthority'.

```

root@PublicServer3:/mnt#
Terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Package Settings Help
Quick connect...
[home/SaiSamitha/
Name
cache
ssh
bash_history
bash_logout
bashrc
profile
.sudo_as_admin_successful
.xauthority
]
3. root@PublicServer1 /mnt/
4. root@PublicServer2: /mnt/fs01
5. root@PublicServer3 /mnt/
SaiSamitha@PublicServer3:~$ sudo mkdir /mnt/fs01
[sudo] password for SaiSamitha: 
SaiSamitha@PublicServer3:~$ [ -d "/etc/smbcredentials" ]; then
> sudo mkdir /etc/smbcredentials
> fi
SaiSamitha@PublicServer3:~$ if [ ! -f "/etc/smbcredentials/samstorage01.cred" ]; then
> sudo bash <- echo "username=samstorage01" >> /etc/smbcredentials/samstorage01.cred
> sudo bash <- echo "password=NNGNzZOKLbBcg4rEptFl0YNEm1jc23TD5apEMghfcubd0M/nsStIIgrCplHeDtxjlx5eauCZ+AstC2/3Uw==" >> /etc/smbcredentials/samstorage01.cred
> fi
SaiSamitha@PublicServer3:~$ sudo chmod 600 /etc/smbcredentials/samstorage01.cred
ho //samstorage01.file.core.windows.net/fs01 /mnt/fs01 cifs nofail,credentials=/etc/smbcredentials/samstorage01.cred,dir_mode=0777,file_mode=0777,serverino,nosharesock,actimeo=30 >> /etc/fstab'
age01.file.core.windows.net/fs01 /mnt/fs01 -o credentials=/etc/smbcredentials/samstorage01.cred,dir_mode=0777,file_mode=0777,serverino,nosharesock,actimeo=30
SaiSamitha@PublicServer3:~$ sudo su
root@PublicServer3:~# cd mnt
root@PublicServer3:/# ls
DATALOSS_WARNING_README.txt  fs01  lost+found
root@PublicServer3:/mnt# 

```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

Step 6: Create a file in the Public Server 2 and you can see the same file in the remaining two servers.

```

root@PublicServer2:/mnt/fs01
Terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Package Settings Help
Quick connect...
[home/SaiSamitha/
Name
cache
ssh
bash_history
bash_logout
bashrc
profile
.sudo_as_admin_successful
.xauthority
]
3. root@PublicServer1 /mnt/
4. root@PublicServer2: /mnt/fs01
5. root@PublicServer3 /mnt/
root@PublicServer2:~# cd mnt
root@PublicServer2:/mnt# ls
DATALOSS_WARNING_README.txt  fs01  lost+found
root@PublicServer2:/mnt# cd fs01
root@PublicServer2:/mnt/fs01# cat > file
This file is created in PublicServer2 and it will be visible in both PublicServer1 and PublicServer3
root@PublicServer2:/mnt/fs01# 

```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

The file created in public server 2 is present in public server 1.

```
root@PublicServer1:/mnt/fs01
root@PublicServer2:/mnt/fs01
root@PublicServer3:/mnt/fs01

SaiSamitha@PublicServer1:~$ sudo mkdir /mnt/fs01
ho bash <- 'echo "username=samstoragedir: cannot create directory '/mnt/fs01': File exists
SaiSamitha@PublicServer1:~$ if [ ! -d "/etc/smbcredentials" ]; then
> sudo mkdir /etc/smbcredentials
> fi
SaiSamitha@PublicServer1:~$ if [ ! -f "/etc/smbcredentials/samstorage01.cred" ]; then
> sudo bash <- 'echo "username=samstorage01" >> /etc/smbcredentials/samstorage01.cred'
> sudo bash <- 'echo "password=NGNzZ0Klb0cg4rEptFLOYNEm1jcz3ZTD5apEMghcFubdQM/nstlIIgrCpLHeDitxjlx5eauCZ+AstC2/3Uw==" >> /etc/smbcredentials/samstorage01.cred'
> fi
SaiSamitha@PublicServer1:~$ sudo chmod 600 /etc/smbcredentials/samstorage01.cred
ho "/samstorage01.file.core.wnSaiSamitha@PublicServer1:~$"
SaiSamitha@PublicServer1:~$ sudo bash <- 'echo //samstorage01.file.core.windows.net/fs01 /mnt/fs01 cifs nofail,credentials=/etc/smbcredentials/samstorage01.cred,dir_mode=0777,file_mode=0777,serverino,nosharelock,actimeo=30' >> /etc/fstab'
SaiSamitha@PublicServer1:~$ sudo mount -t cifs //samstorage01.file.core.windows.net/fs01 /mnt/fs01 -o credentials=/etc/smbcredentials/samstorage01.cred,dir_mode=0777,file_mode=0777,serverino,nosharelock,actimeo=30
SaiSamitha@PublicServer1:~$ sudo su
root@PublicServer1:/home/SaiSamitha# cd /
root@PublicServer1:/# ls
bin boot dev etc home lib lib32 lib64 libx32 lost+found media mnt opt proc root run sbin snap srv sys tmp usr var
root@PublicServer1:/# mnt
root@PublicServer1:/mnt# ls
DATALOSS WARNING README.txt [fs01] lost+found
root@PublicServer1:/mnt# cd fs01
root@PublicServer1:/mnt/fs01# ls
file
root@PublicServer1:/mnt/fs01# cat file
This file is created in PublicServer2 and it will be visible in both PublicServer1 and PublicServer3
root@PublicServer1:/mnt/fs01#
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

The file created in public server 2 is present in public server 3.

```
root@PublicServer3:/mnt/fs01
root@PublicServer2:/mnt/fs01
root@PublicServer1:/mnt/fs01

SaiSamitha@PublicServer3:~$ sudo mkdir /mnt/fs01
ho /etc/smbcredentials"; then
Sudkdir: cannot create directory '/mnt/fs01': File exists
SaiSamitha@PublicServer3:~$ if [ ! -d "/etc/smbcredentials" ]; then
> sudo mkdir /etc/smbcredentials
> fi
SaiSamitha@PublicServer3:~$ if [ ! -f "/etc/smbcredentials/samstorage01.cred" ]; then
> sudo bash <- 'echo "username=samstorage01" >> /etc/smbcredentials/samstorage01.cred'
> sudo bash <- 'echo "password=NGNzZ0Klb0cg4rEptFLOYNEm1jcz3ZTD5apEMghcFubdQM/nstlIIgrCpLHeDitxjlx5eauCZ+AstC2/3Uw==" >> /etc/smbcredentials/samstorage01.cred'
> fi
SaiSamitha@PublicServer3:~$ sudo chmod 600 /etc/smbcredentials/samstorage01.cred
ho "/samstorage01.file.core.wnSaiSamitha@PublicServer3:~$"
SaiSamitha@PublicServer3:~$ sudo bash <- 'echo //samstorage01.file.core.windows.net/fs01 /mnt/fs01 cifs nofail,credentials=/etc/smbcredentials/samstorage01.cred,dir_mode=0777,file_mode=0777,serverino,nosharelock,actimeo=30' >> /etc/fstab'
SaiSamitha@PublicServer3:~$ sudo mount -t cifs //samstorage01.file.core.windows.net/fs01 /mnt /fs01 -o credentials=/etc/smbcredentials/samstorage01.cred,dir_mode=0777,file_mode=0777,serverino,nosharelock,actimeo=30
SaiSamitha@PublicServer3:~$ sudo su
root@PublicServer3:/home/SaiSamitha# cd /
root@PublicServer3:/# mnt
root@PublicServer3:/mnt# ls
DATALOSS WARNING README.txt [fs01] lost+found
root@PublicServer3:/mnt# cd fs01
root@PublicServer3:/mnt/fs01# ls
file
root@PublicServer3:/mnt/fs01# cat file
This file is created in PublicServer2 and it will be visible in both PublicServer1 and PublicServer3
root@PublicServer3:/mnt/fs01#
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>