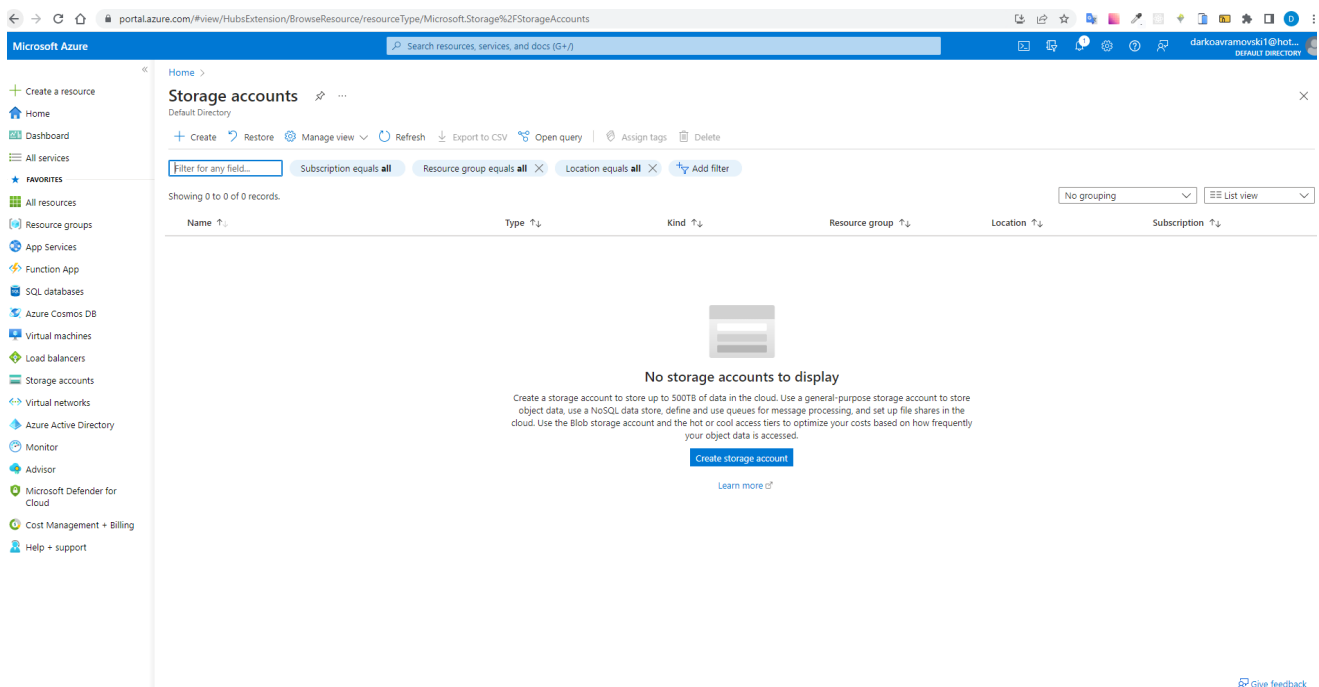


Mid-Term Task

Part I

1. Host a static website on Blob Storage: build and deploy a static Hello World website to Azure Storage.
2. Verify that the default web page has the Hello World! page.
3. Provide the steps and results.

Login to your account in [azure.portal](https://portal.azure.com) navigate to **Create a storage** > **Create a storage account** then click **Create storage account**



Fill required details **Project details** and **Instance details** and click **Create**

Microsoft Azure

Search resources, services, and docs (G+/)

Create a resource

Home

Dashboard

All services

FAVORITES

All resources

Resource groups

App Services

Function App

SQL databases

Azure Cosmos DB

Virtual machines

Load balancers

Storage accounts

Virtual networks

Azure Active Directory

Monitor

Advisor

Microsoft Defender for Cloud

Cost Management + Billing

Help + support

Home > Storage accounts >

Create a storage account

BasicsAdvancedNetworkingData protectionEncryptionTagsReview

Azure Storage is a Microsoft-managed service providing cloud storage that is highly available, secure, durable, scalable, and redundant. Azure Storage includes Azure Blobs (objects), Azure Data Lake Storage Gen2, Azure Files, Azure Queues, and Azure Tables. The cost of your storage account depends on the usage and the options you choose below. [Learn more about Azure storage accounts](#)

Project details

Select the subscription in which to create the new storage account. Choose a new or existing resource group to organize and manage your storage account together with other resources.

Subscription *

Azure Pass - Sponsorship

Resource group *

rgLearn

Create new

Instance details

If you need to create a legacy storage account type, please click [here](#).

Storage account name ⓘ *

webapp2910

Region ⓘ *

(US) East US

Deploy to an edge zone

Performance ⓘ *

☒ Standard: Recommended for most scenarios (general-purpose v2 account)

☐ Premium: Recommended for scenarios that require low latency.

Redundancy ⓘ *

Geo-redundant storage (GRS)

Review

< Previous

Next : Advanced >

Open you storage that you created

Microsoft Azure

Search resources, services, and docs (G+/)

Create a resource

Home

Dashboard

All services

FAVORITES

All resources

Resource groups

App Services

Function App

SQL databases

Azure Cosmos DB

Virtual machines

Load balancers

Storage accounts

Virtual networks

Azure Active Directory

Monitor

Advisor

Microsoft Defender for Cloud

Cost Management + Billing

Help + support

Home >

Storage accounts

Default Directory

+ Create

Restore

Manage view

Refresh

Export to CSV

Open query

Assign tags

Delete

Filter for any field...

Subscription equals all

Resource group equals all

Location equals all

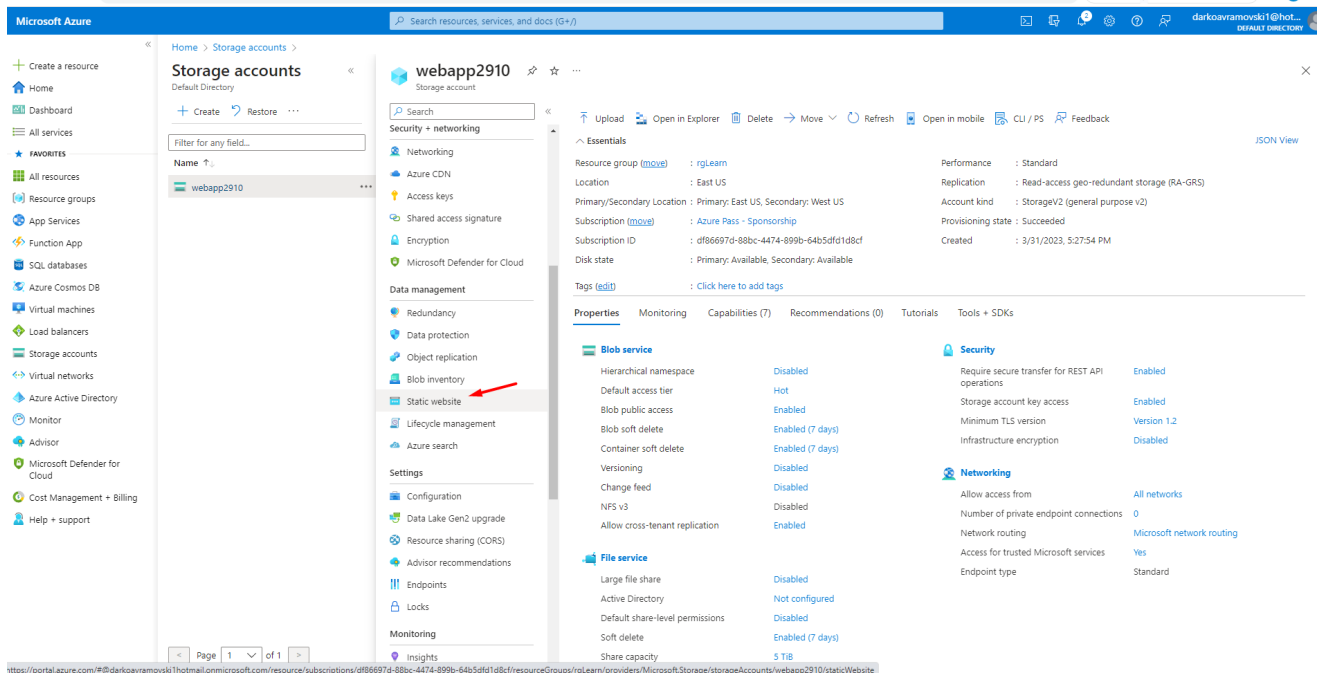
Add filter

Showing 1 to 1 of 1 records.

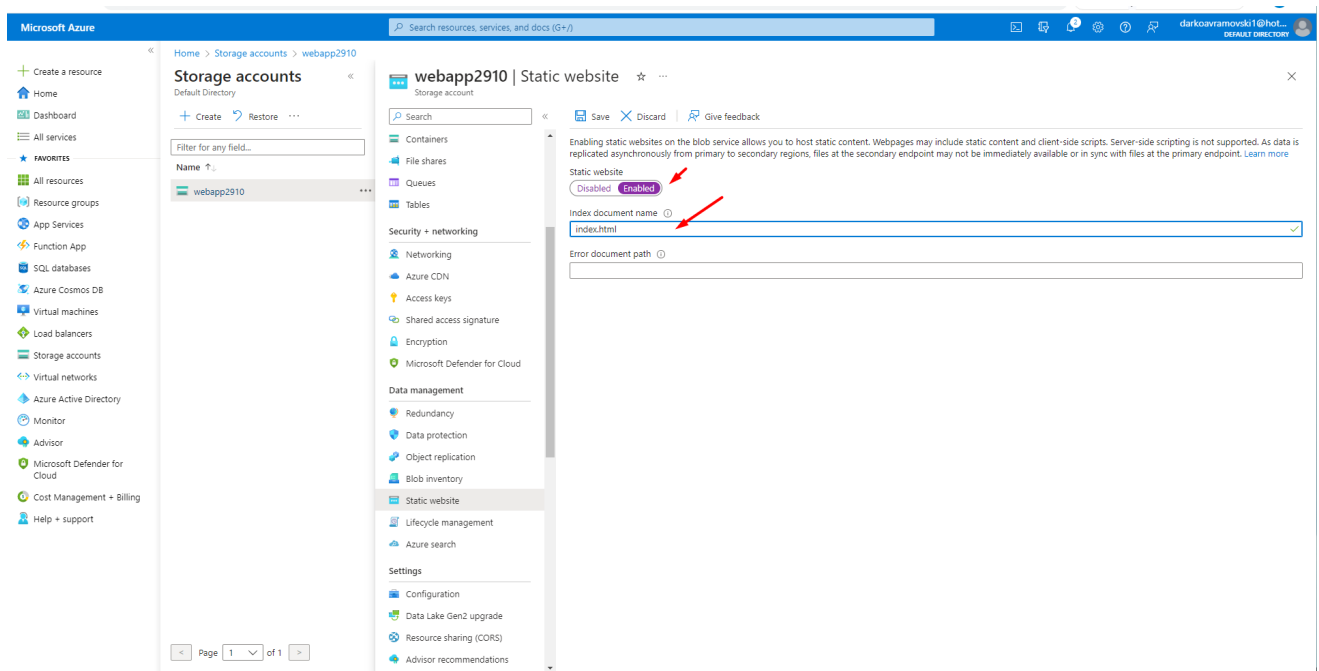
<input type="checkbox"/> Name ↑↓	Type ↑↓	Kind ↑↓	Resource group ↑↓	Location ↑↓
<input type="checkbox"/> webapp2910	Storage account	StorageV2	rgLearn	East US

From the left side menu select **Data management/Static website**

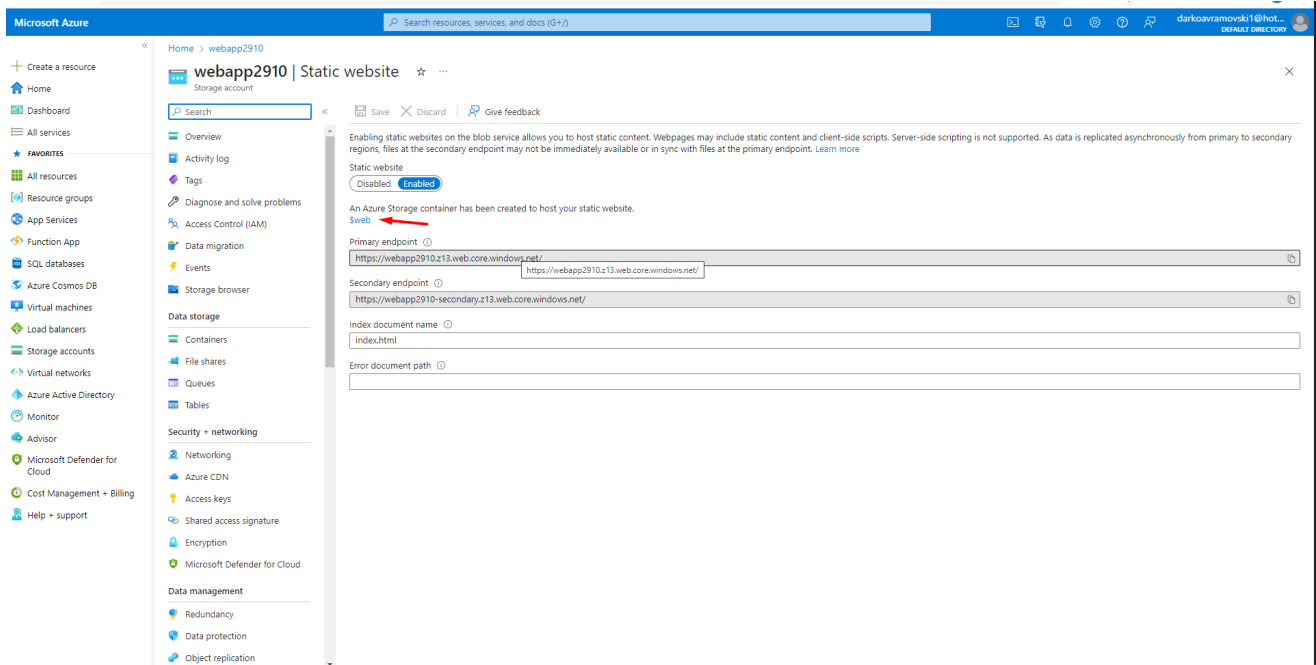
2/12



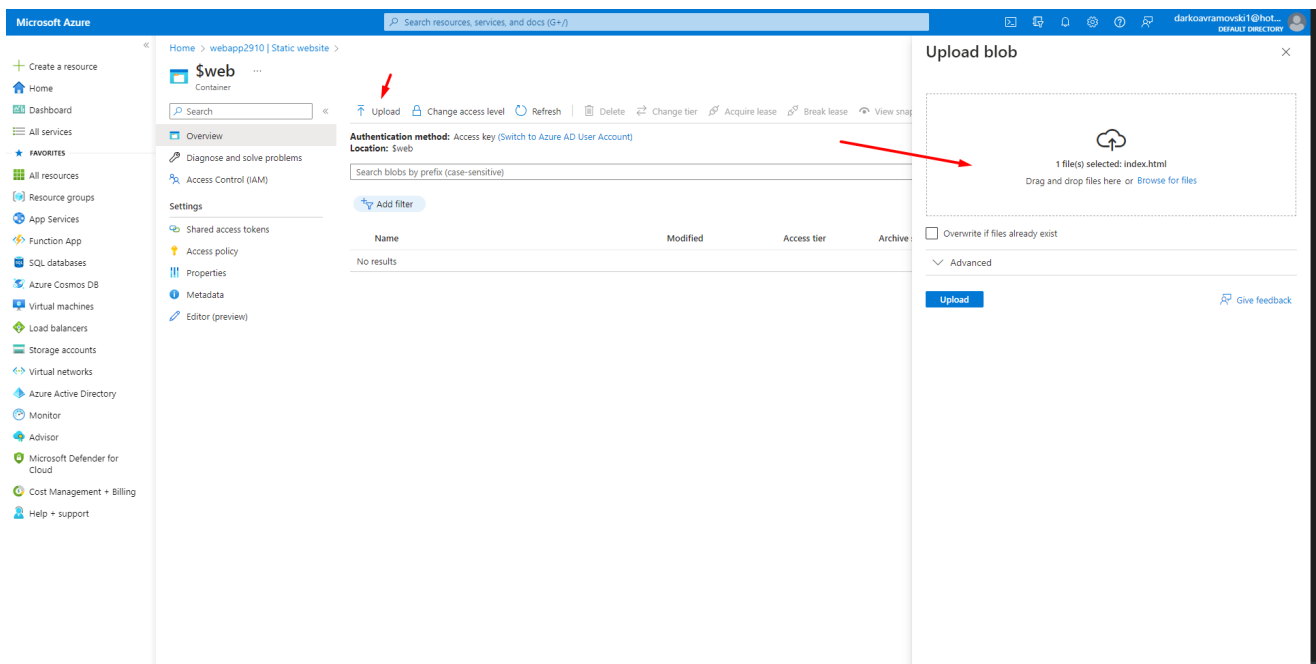
On toggle button click **Enable Static website** and fill Index document name *index.html* and click on the Save button



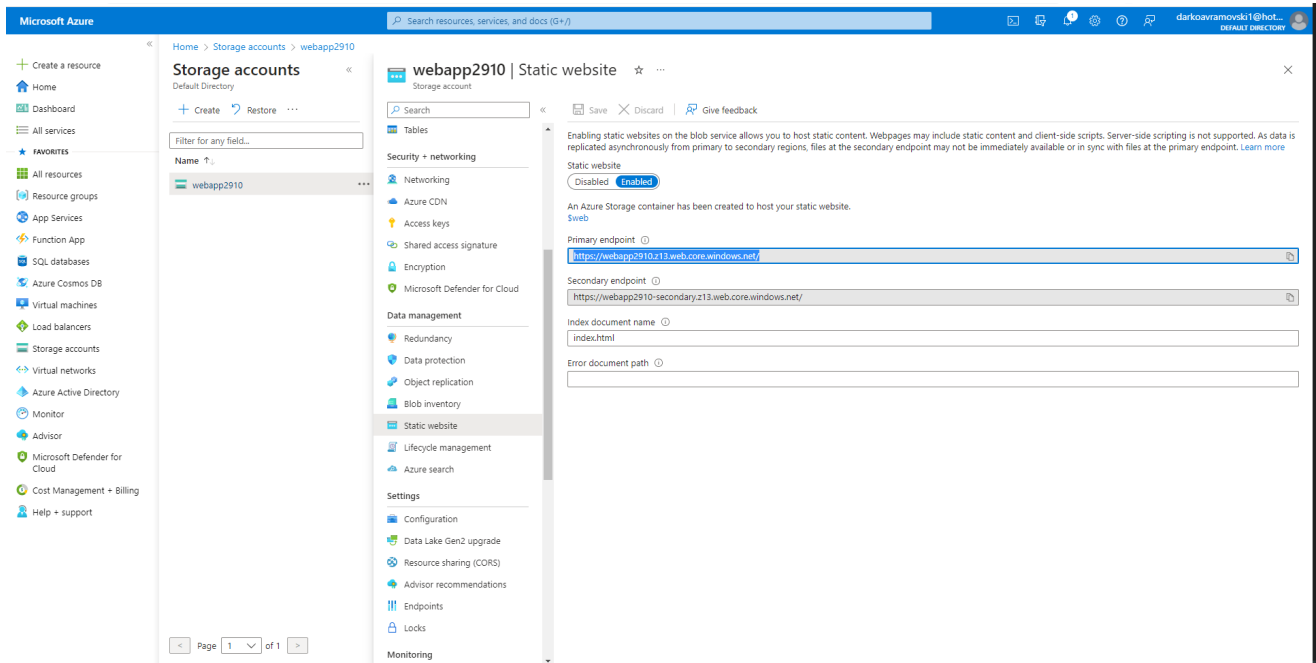
After we Successfully we creted the static page we can click on \$web to upload new file



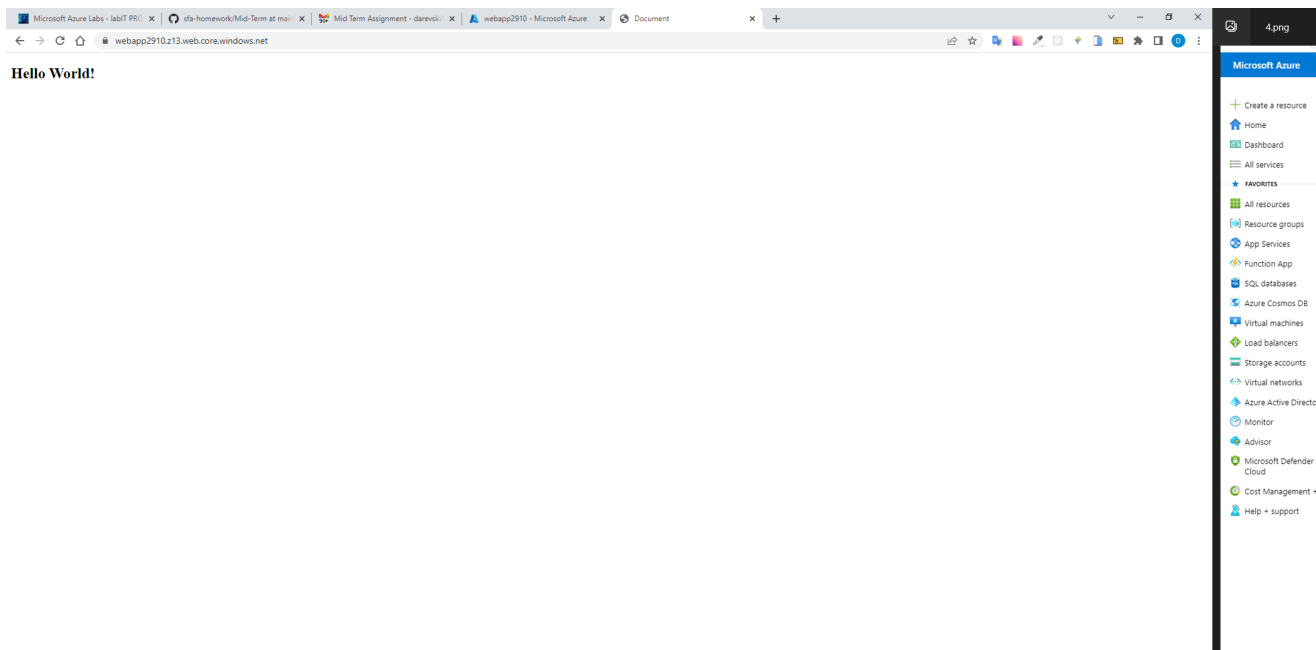
On the next screen click **upload** and new popup windows from the right side will appear select the file that we created **index.html** and click upload



Navigate back to Static website select storage account / static website from the left side menu and copy **Primary endpoint**



Verify that the static web page is working.



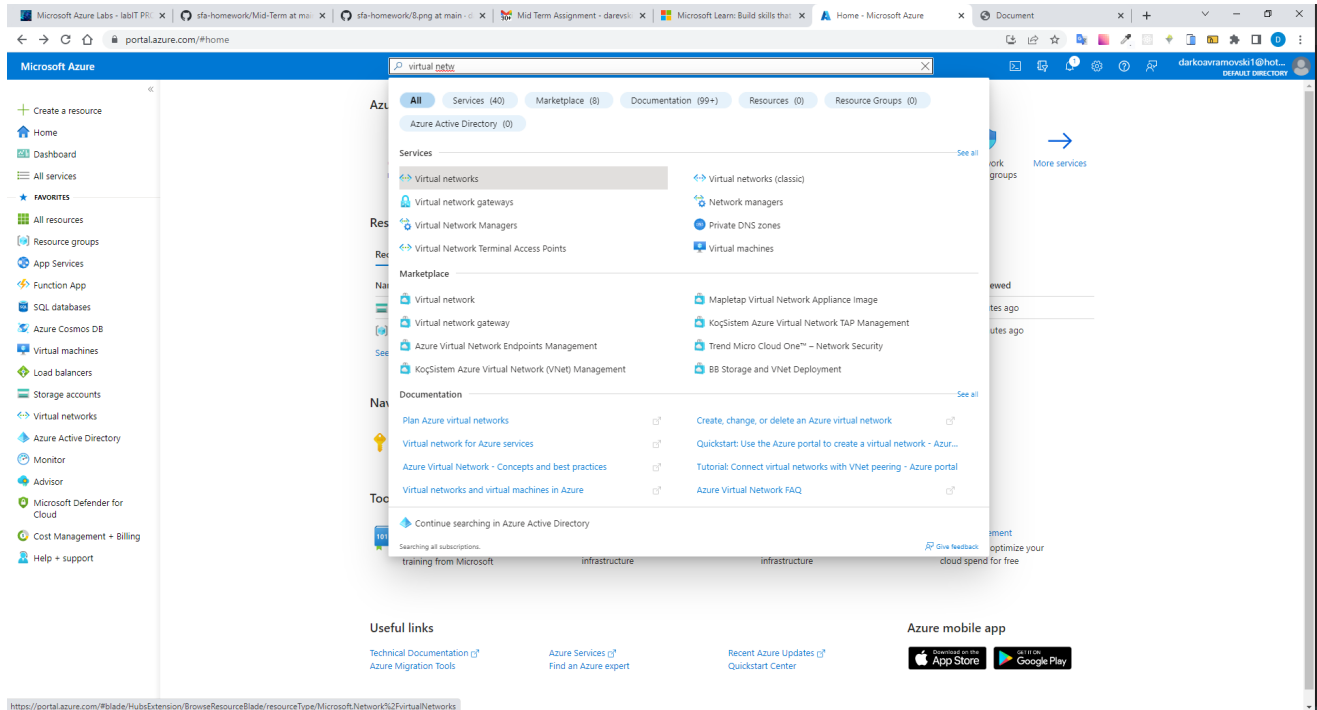
Part II

In this exercise we will setup a Linux based web server and will deploy a web page on it.

1. Create a Virtual Network where you will deploy your Linux Based Web Server.
2. Modify the network security group for your virtual machine that will allow you to remotely manage your machine only from your local machine and nowhere else.
3. Create a Linux Virtual Machine that will be your Web Server which is publicly available for web publishing (not SSL) only from your machine and nowhere else.
4. Connect to the VM.
5. Install Apache Web Server.

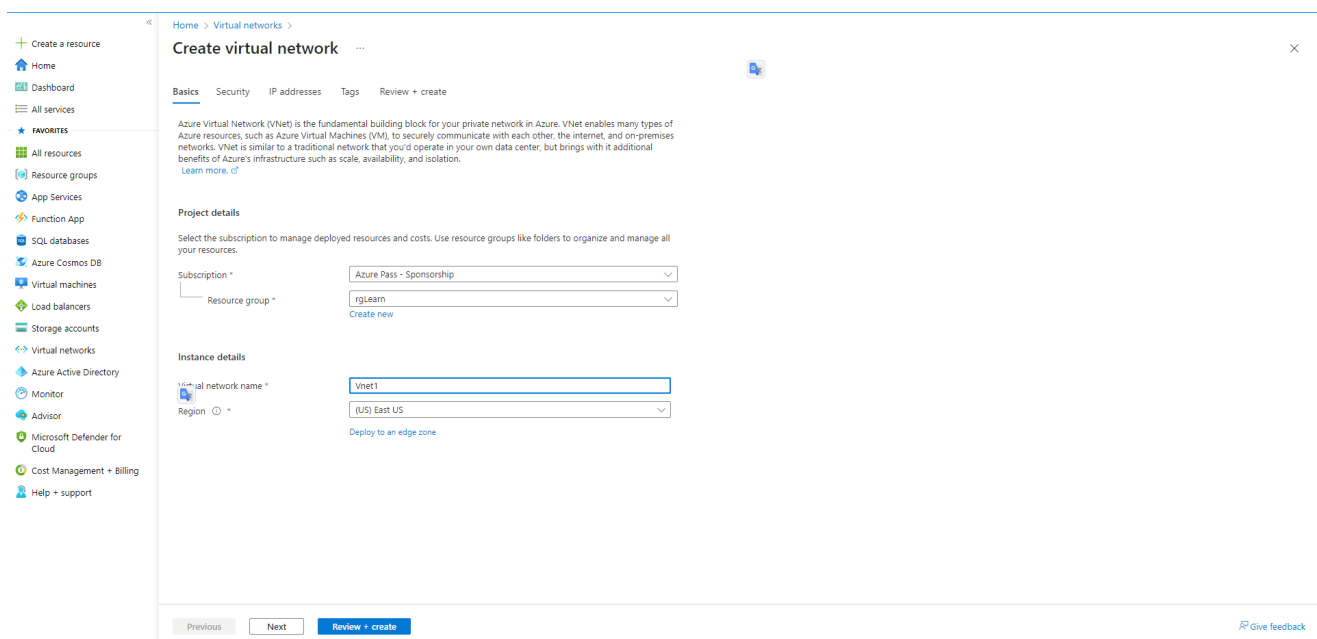
6. Deploy the "Hello World" web page.
7. Provide Testing from your cellphone.
8. Provide the steps and results.

Login to your account, in the search bar type **Virtual networks** open virtual network and click create new virtual network now we are going to create new virtual network

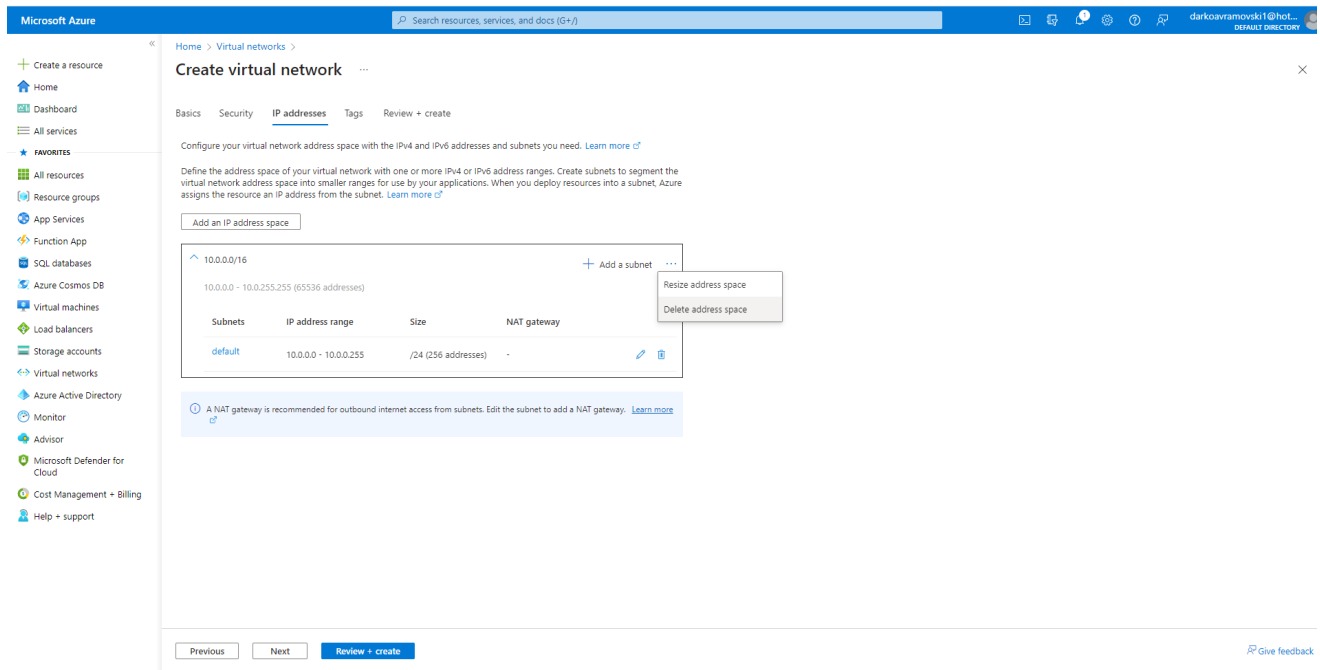


on the next screen fill required fields

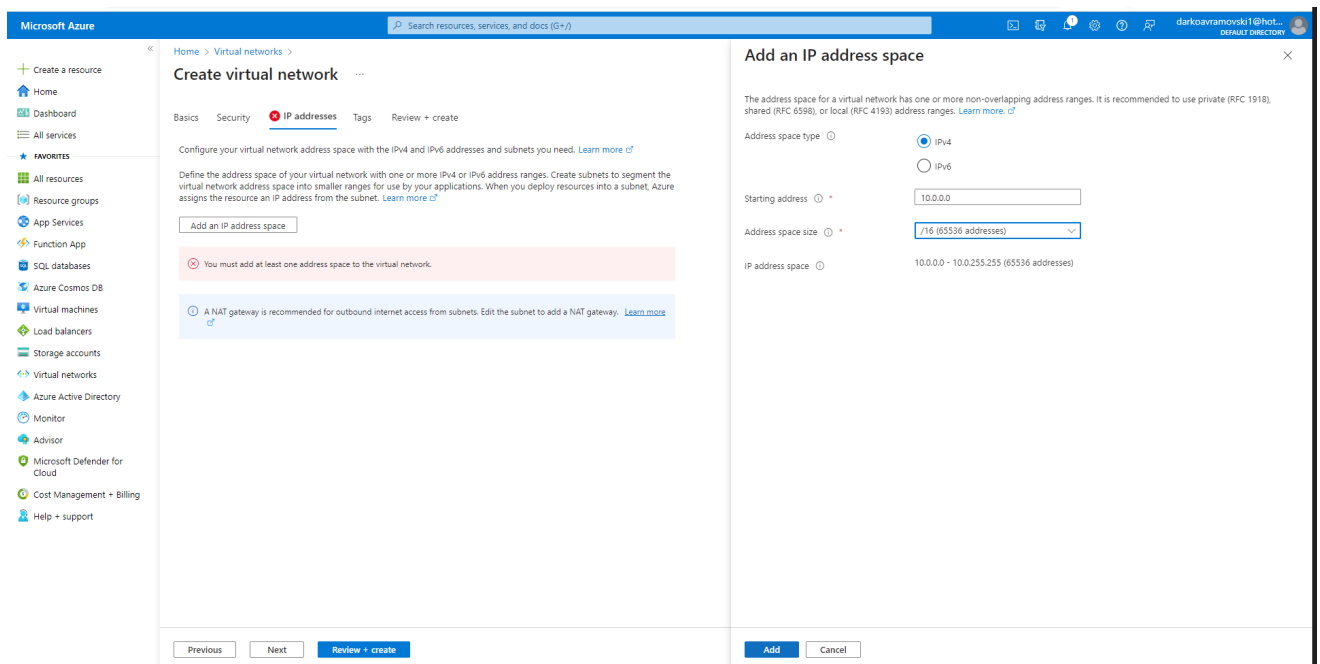
- Subscription: Choes Subscription
- Resource group: Select resource group or create new one Instance details
- Virtual network name: Type virtual network name
- Region: Select Region



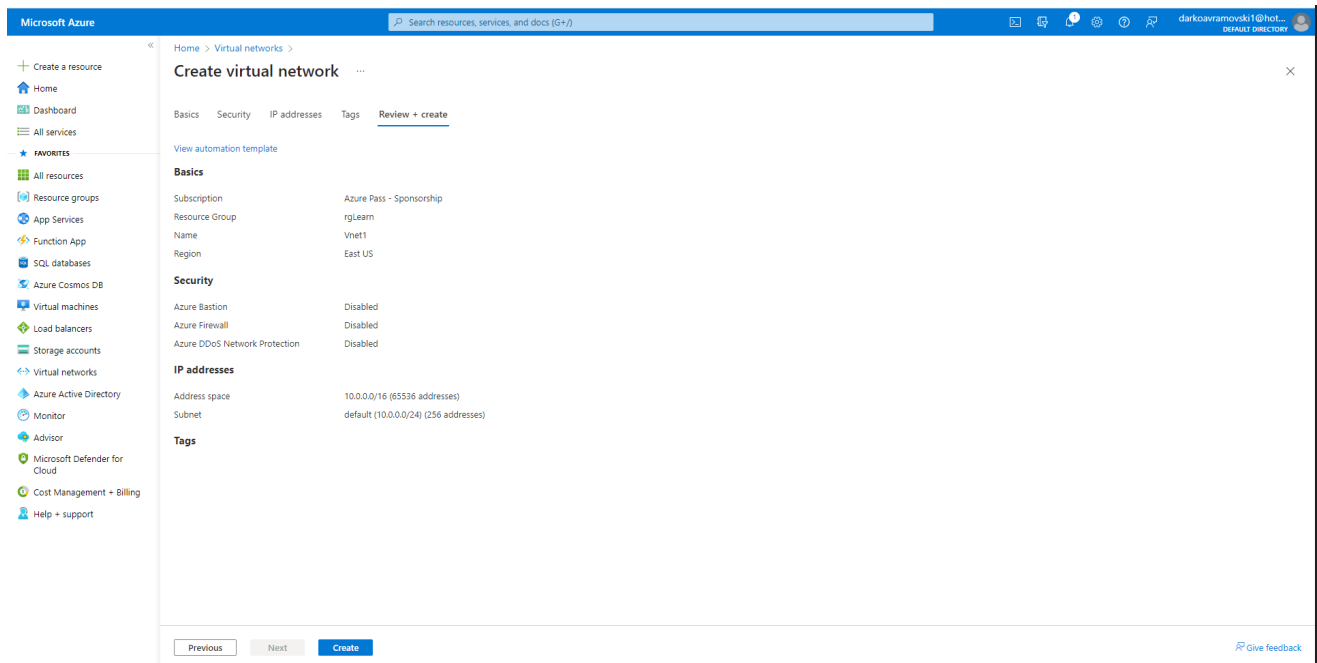
Click next Securty tab will appear click Next **IP addresses** delete default address space that already appears.



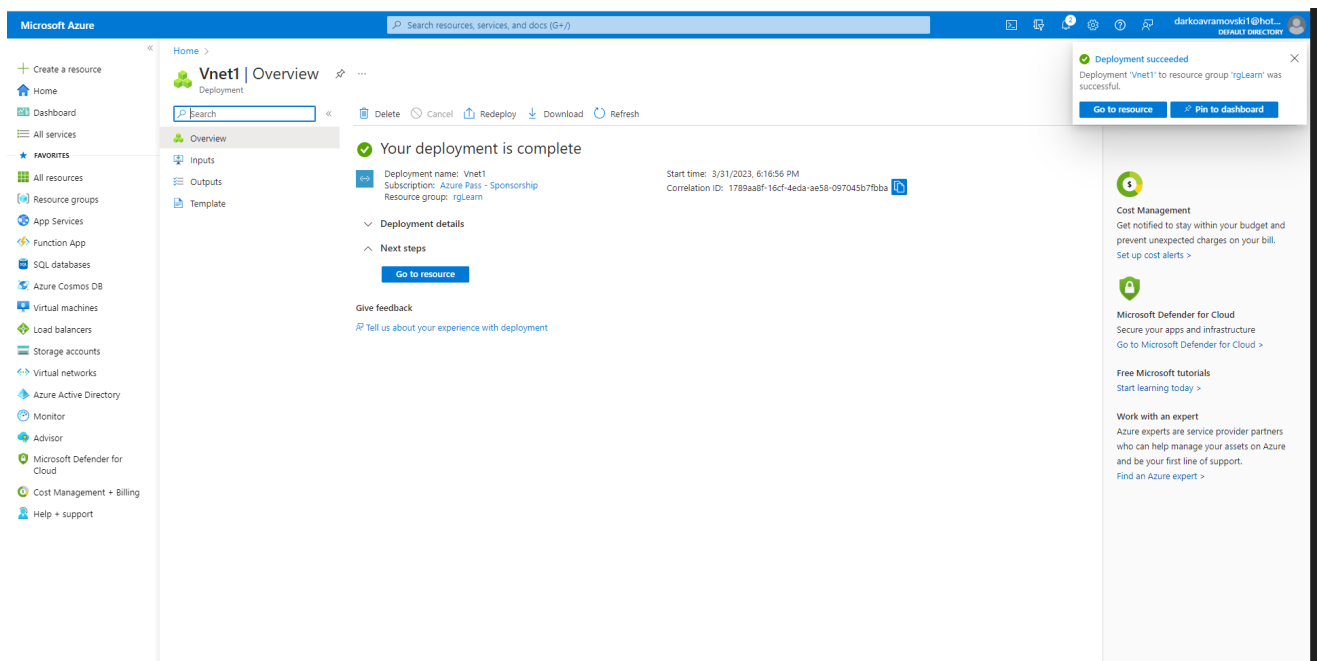
Click on the button Add an IP address space, and new window we appear on the right side on the screen and fill the address space for new Ip address



Now Select Add subnet and create new subnet for our virtual network. And click **Review and create + Create**



Your deployment is complete



Next Step: Modify the network security group for your virtual machine that will allow you to remotely manage your machine only from your local machine and nowhere else.

Create a Linux Virtual Machine that will be your Web Server which is publicly available for web publishing (not SSL) only from your machine and nowhere else.

Navigate to **Virtual machine > Create a virtual machine** and click a new virtual machine

Create new virtual machine and Connect to the VM.


```
ds@dsApp01: /var/www/html
login as: ds
ds@74.235.161.221's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1035-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Fri Mar 31 17:36:37 UTC 2023

System load:  0.5               Processes:           106
Usage of /:   5.2% of 28.89GB   Users logged in:    0
Memory usage: 8%               IPv4 address for eth0: 10.0.0.9
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ds@dsApp01:~$ sudo apt update
Hit:1 http://azure.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://azure.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://azure.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:4 http://azure.archive.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:5 http://azure.archive.ubuntu.com/ubuntu focal/universe amd64 Packages [8628
```

install apache web server

```
sudo apt update
sudo apt install apache2
```

```

ds@dsApp01: ~
Setting up libaprutil1-dbd-sqlite3:amd64 (1.6.1-4ubuntu2.1) ...
Setting up apache2-utils (2.4.41-4ubuntu3.14) ...
Setting up apache2-bin (2.4.41-4ubuntu3.14) ...
Setting up apache2 (2.4.41-4ubuntu3.14) ...
Enabling module mpm_event.
Enabling module authz_core.
Enabling module authz_host.
Enabling module authn_core.
Enabling module auth_basic.
Enabling module access_compat.
Enabling module authn_file.
Enabling module authz_user.
Enabling module alias.
Enabling module dir.
Enabling module autoindex.
Enabling module env.
Enabling module mime.
Enabling module negotiation.
Enabling module setenvif.
Enabling module filter.
Enabling module deflate.
Enabling module status.
Enabling module reqtimeout.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /lib/systemd/system/apache-htcacheclean.service.
Processing triggers for ufw (0.36-6ubuntu1) ...
Processing triggers for systemd (245.4-4ubuntu3.20) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.9) ...
ds@dsApp01:~$ sudo apt install apache2

```

cd /var/www/html/

Apache is working on port 80

Not secure | 74.235.161.221

Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```

/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf

```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets that manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf`. See their respective man pages for detailed information.
- The binary is called `apache2`. Due to the use of environment variables, in the default configuration, `apache2` needs to be started/stopped with `/etc/init.d/apache2` or `apache2ctl`. Calling `/usr/bin/apache2` directly will not work with the default configuration.

```

cd /var/www
sudo mkdir hello_world
cd /var/www/hello_world
nano index.html

```

A screenshot of a web browser window. The address bar at the top shows a 'Not secure' warning and the URL '74.235.161.221'. The main content area of the browser displays the text 'Hello World!' in a large, bold, black font.



