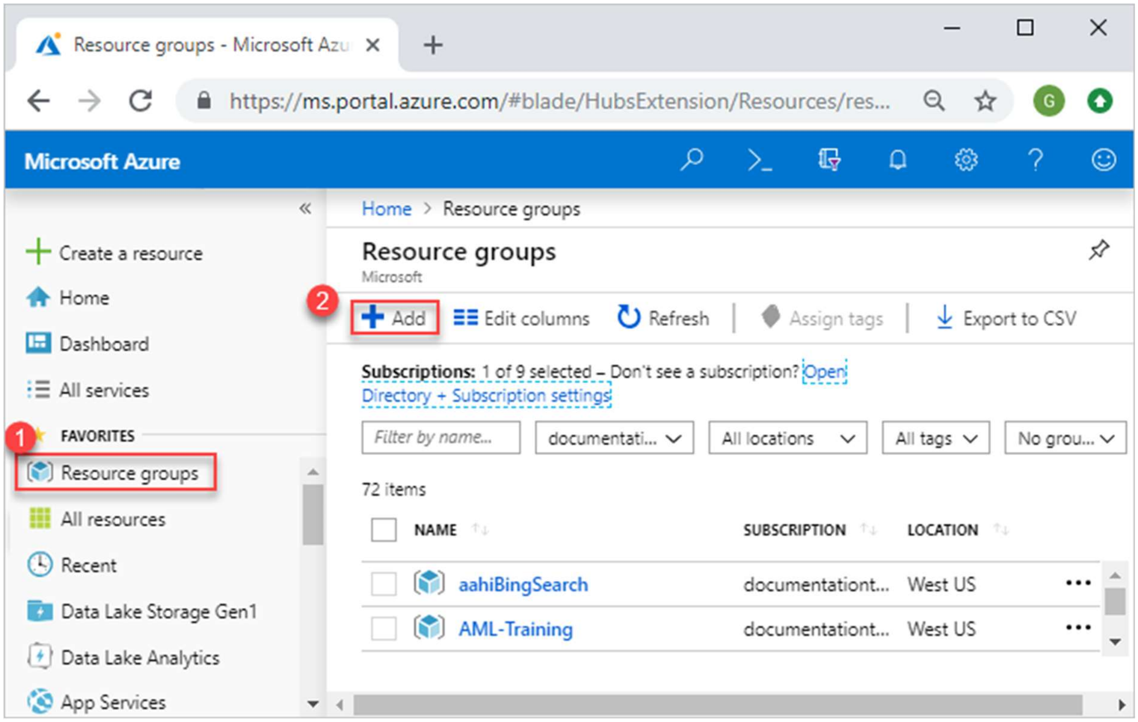
Steps for deploying Web App on Azure Web App service

Create resource groups and Web App resource:

1. Sign into the Azure portal.
2. Select Resource groups from side bar menu.



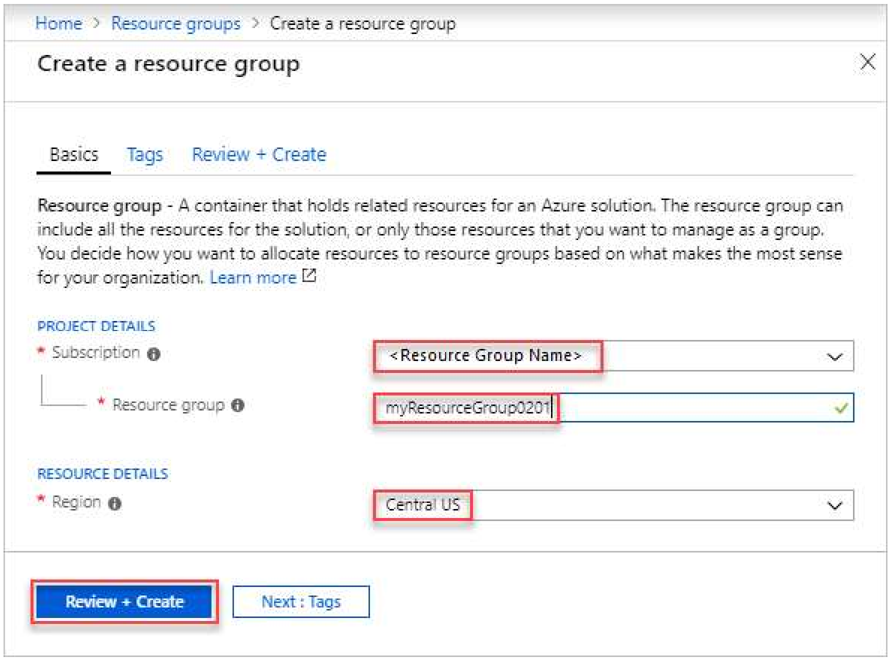
Resource groups
Microsoft

Subscriptions: 1 of 9 selected – Don't see a subscription? Open Directory + Subscription settings

72 items

NAME	SUBSCRIPTION	LOCATION
aahiBingSearch	documentationt...	West US
AML-Training	documentationt...	West US

3. Select **Add**.
4. Enter the following values:
5. Subscription: Select your Azure subscription.
6. Resource group: Enter a new resource group name.
7. Region: Select Azure location as Germany West Central. This gives more memory compared to other regions.



Create a resource group

Resource group - 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. Learn more

PROJECT DETAILS

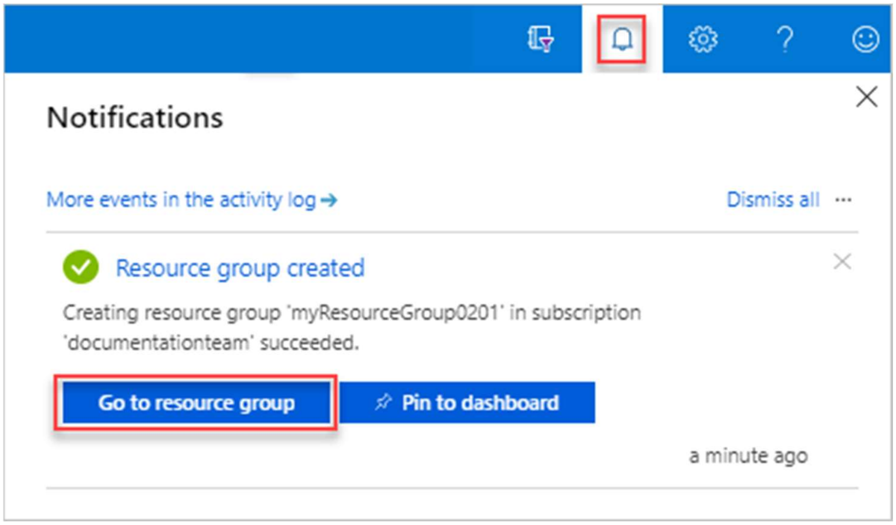
* Subscription: <Resource Group Name>

* Resource group: myResourceGroup0201

RESOURCE DETAILS

* Region: Central US

8. Select Review + Create
9. Select Create. It takes a few seconds to create a resource group.
10. Select Refresh from the top menu to refresh the resource group list, and then select the newly created resource group to open it. Or select Notification(the bell icon) from the top, and then select Go to resource group to open the newly created resource group



Notifications

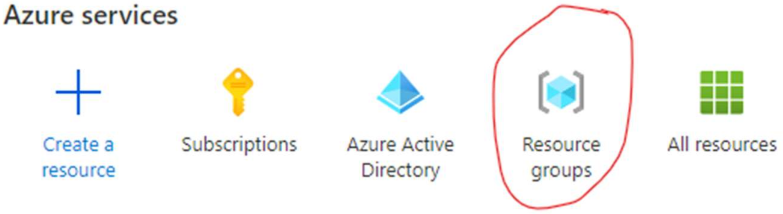
More events in the activity log →

Resource group created

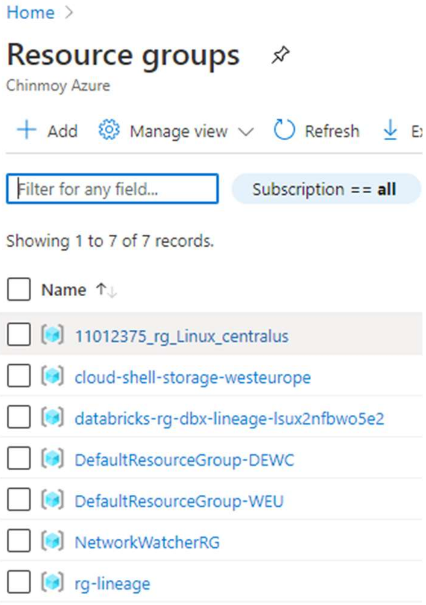
Creating resource group 'myResourceGroup0201' in subscription 'documentationteam' succeeded.

a minute ago

11. To view the resource groups, goto the homepage and click on **Resource Group** icon



12. This shows all the resource groups created under the cloud subscription.



Home >

Resource groups

Chinmoy Azure

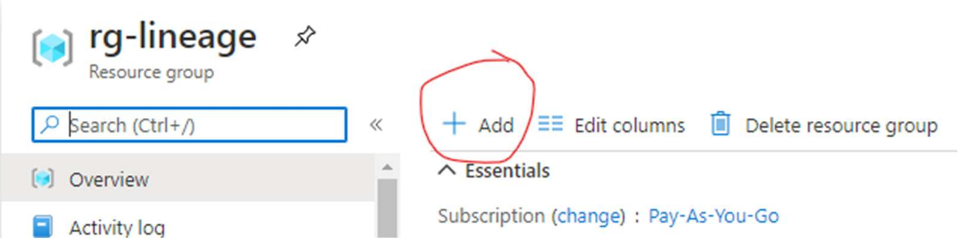
Filter for any field...

Subscription == **all**

Showing 1 to 7 of 7 records.

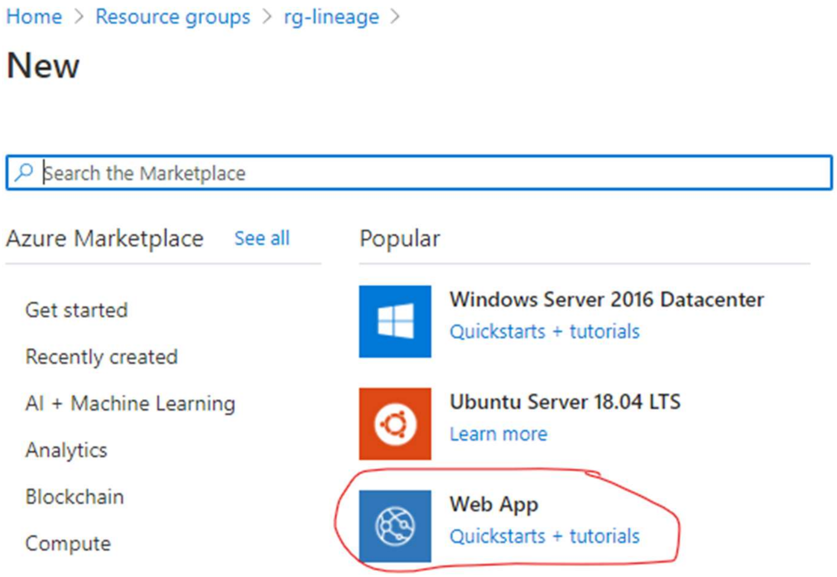
□ Name ↑↓
□ 11012375_rg_Linux_centralus
□ cloud-shell-storage-westeurope
□ databricks-rg-dbx-lineage-lsux2nfbwo5e2
□ DefaultResourceGroup-DEWC
□ DefaultResourceGroup-WEU
□ NetworkWatcherRG
□ rg-lineage

13. Goto the resource group just created and Click on Add option to create a resource.

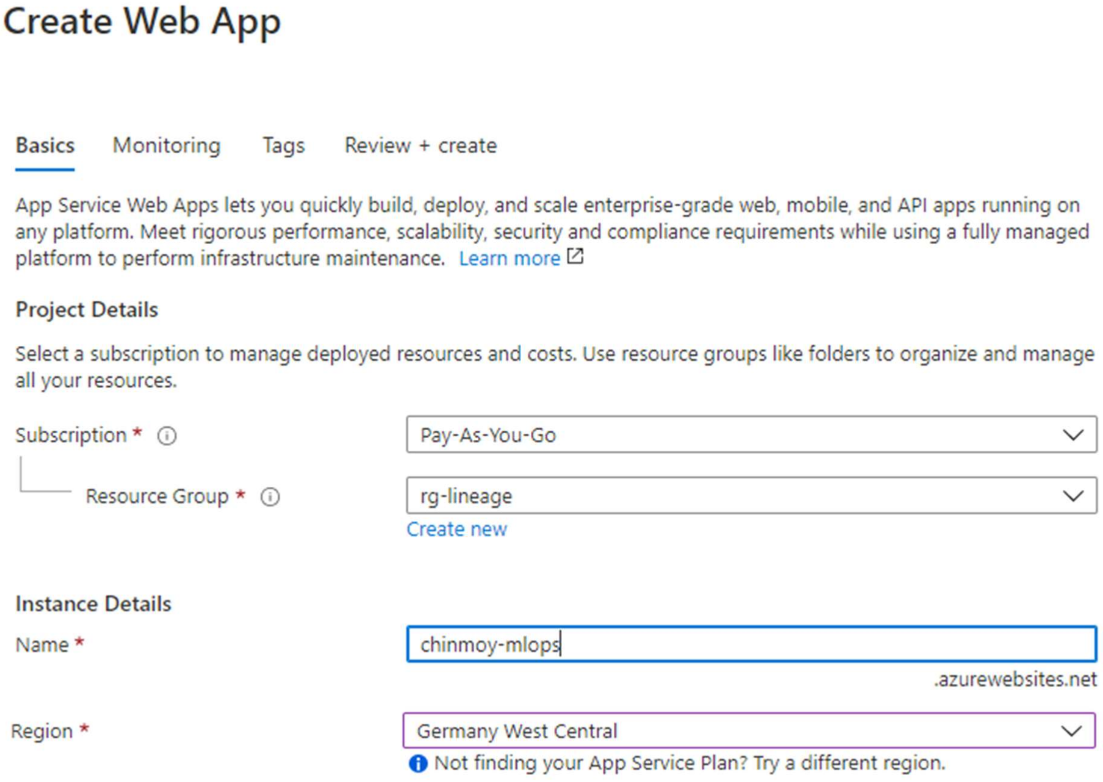


Subscription (change) : Pay-As-You-Go

14. Then select Web App



15. In the next page give a name for the app and select the region as suggested above:



App Service Web Apps lets you quickly build, deploy, and scale enterprise-grade web, mobile, and API apps running on any platform. Meet rigorous performance, scalability, security and compliance requirements while using a fully managed platform to perform infrastructure maintenance. Learn more

Project Details

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * : Pay-As-You-Go

Resource Group * : rg-lineage

Create new

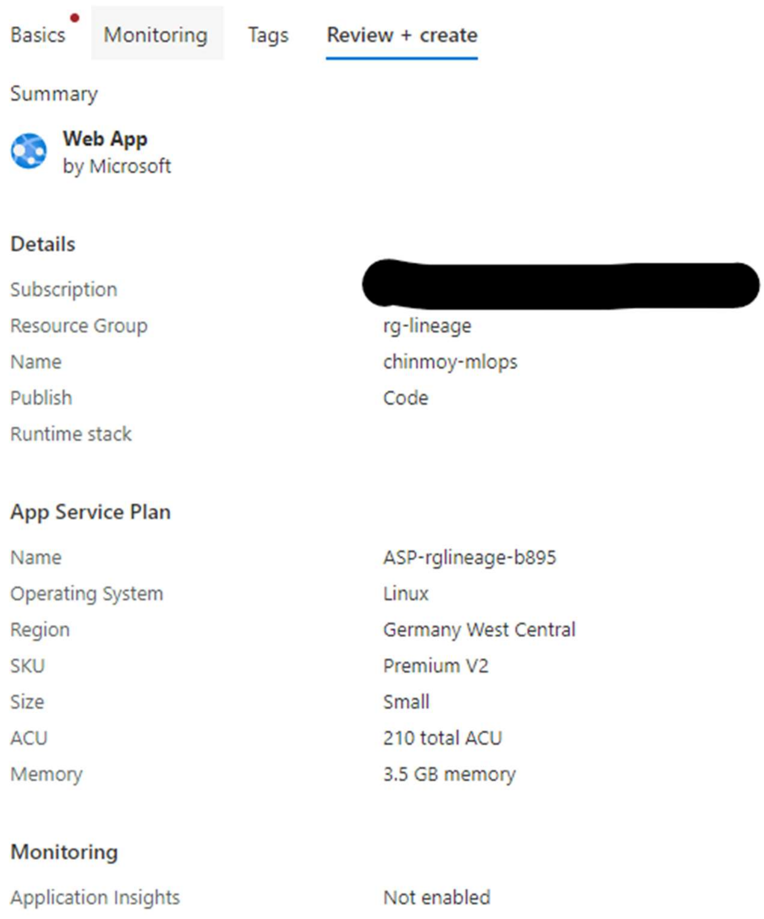
Instance Details

Name * : chinmoy-mlops .azurewebsites.net

Region * : Germany West Central

Not finding your App Service Plan? Try a different region.

16. Click on **Review and Create** to goto the validation page which provides all information about the service just created.



Summary

Web App
by Microsoft

Details

Subscription [redacted]
Resource Group rg-lineage
Name chinmoy-mlops
Publish Code
Runtime stack

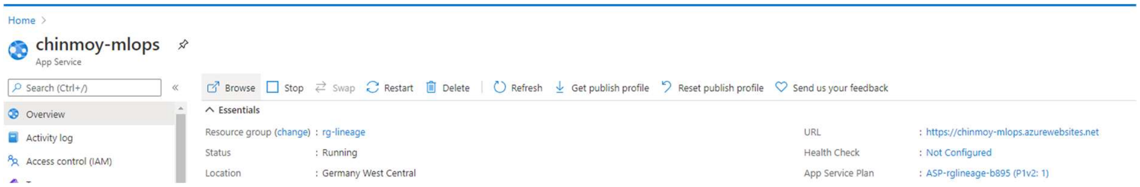
App Service Plan

Name ASP-rglineage-b895
Operating System Linux
Region Germany West Central
SKU Premium V2
Size Small
ACU 210 total ACU
Memory 3.5 GB memory

Monitoring

Application Insights Not enabled

17. Click on Create
18. Once completed, following page for the app will come up:



Resource group (change) : rg-lineage
Status : Running
Location : Germany West Central
URL : https://chinmoy-mlops.azurewebsites.net
Health Check : Not Configured
App Service Plan : ASP-rglineage-b895 (P1v2: 1)

19. The URL provided is the link for webpage.

Publishing the local Flask app using Git:

1. To push the flask application code to web app, install Git bash from the Git website: https://git-scm.com/download/win

2. Create your local Git repo for your code. If your code is already in a local Git repo, you can skip this step.
3. Navigate to the folder where your code is on the command line:

   ```
   cd /home/fabrikam/fiber
   ```

4. Create a Git repo on your machine to store your code. You will connect this repo to Azure Repos in the next section.

   ```
   git init .
   ```

5. Commit your code into the local Git repo by running below commands:

   ```
   git add --all
   git commit -m "first commit of my code"
   ```

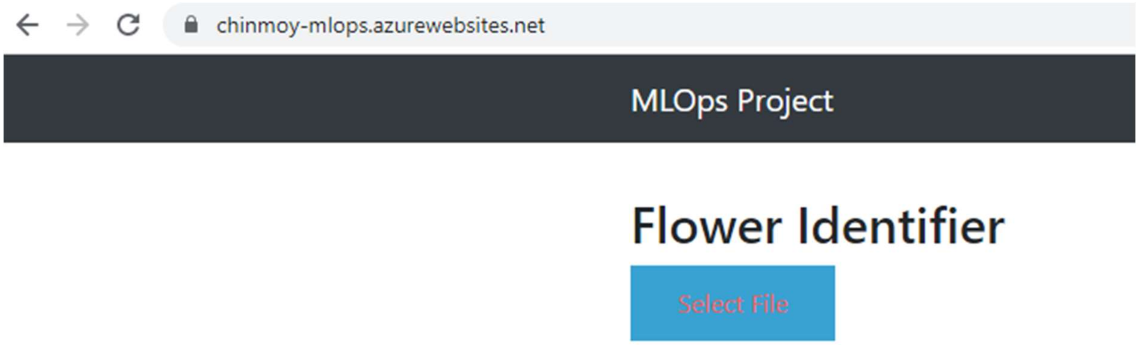
6. Connect your local repo to the Git repo in Azure Repos using the copied clone URL in the git remote command:

   ```
   git remote add origin https://mlops@chinmoy-mlops.scm.azurewebsites.net/chinmoy-mlops.git
   ```

7. Push your code
8. Before pushing your code, set up authentication with credential managers or SSH before continuing.

   ```
   git push origin main
   ```

9. After the push is done, goto the Azure portal and click on the URL. This open the website below where you can browse for images and identify them:


chinmoy-mlops.azurewebsites.net
MLOps Project
Flower Identifier
Select File