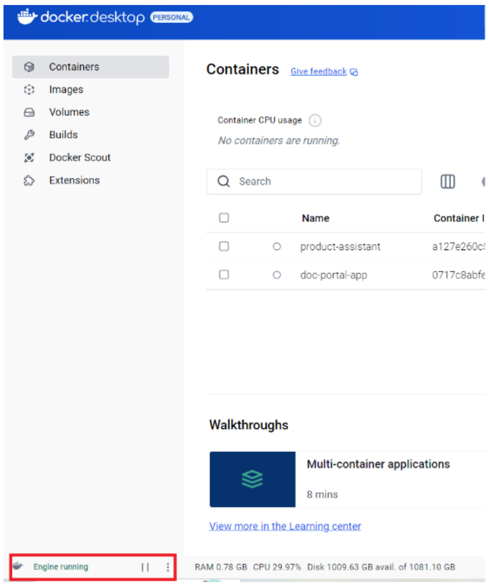
NOTE: DOCKER ENGINE MUST RUN:

1. To run follow below steps
    a. Download Docker Desktop from google then open
    b. Open the App in the bottom left you should see the "EngineRunning"
    c. 

    No containers are running.

	Name	Container I
☐ ○	product-assistant	a127e260c
☐ ○	doc-portal-app	0717c8abfe

    RAM 0.78 GB CPU 29.97% Disk 1009.63 GB avail. of 1081.10 GB
2. Install Azure CLI with below link
    a. https://azure.microsoft.com/en-us/get-started/azure-portal.
    b. Select the Microsoft Installer(MSI)
    c. To check whether the laptop is 32bit or 64bit open the "System Information"=>In the "System type" you can see x64-based PC or x32-based PC.
3. Open command prompt, Type below command to verify azure installed or not.
    a. **az –version**
4. Create the Azure account then type below commands in command prompt or bash terminal.
    a. **az login** [IF IT IS GIVING AUTHENTICATION FAILED AGAINST TENANT ID. IF YOU RUN FOR FIRST TIME YOU WILL GET THIS ERROR TO OVERCOME THE PROBLEM PERFORM BELOW STEPS]
        i. **az login --tenant <<GIVE THE TENANT ID HERE>>**
            1. EXAMPLE: az login --tenant 45eer456-e67e-89tg-8763-f234122er456
    b. **az account list --output table**
    c. **az account set --subscription <<GIVE YOUR SUBSCRIPTION ID>>**
        i. YOU WILL GET THE SUBSCRIPTION ID WITH THE ABOVE COMMAND
    d. **az provider register --namespace Microsoft.ContainerRegistry**
    e. **az provider register -n Microsoft.Storage --subscription <subscription_id>[If subscription not found says just run this command]**
    f. **az account show**
    g. **az group list**
    h. **az login**
    i. **If az is not coming in your vs terminal then do this**
        1. Write where az on your cmd
        2. Then you will get the path of az
        3. Then open the vs code
        4. Open command pallet (view>command pallet)(shortcur is ctrl+shift+p)

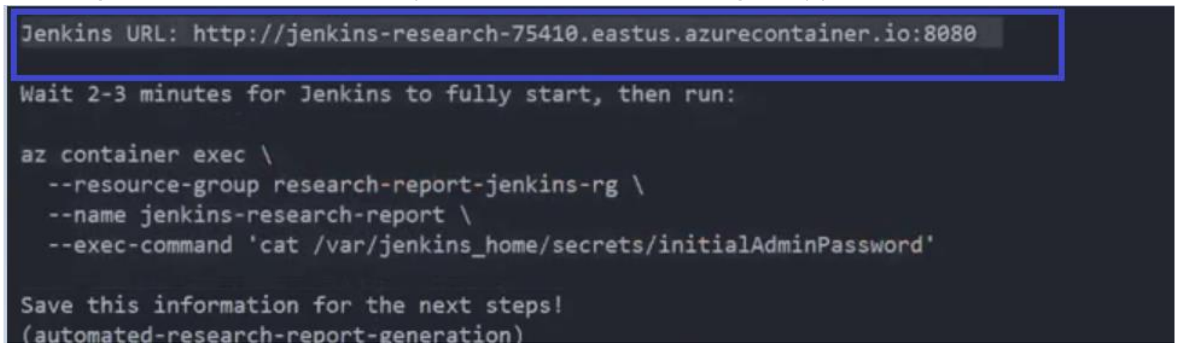
5. Then type there: **Preferences: Open Settings (JSON)**
6. {

```
    "python-envs.pythonProjects": [],
  "terminal.integrated.env.windows": {
      "PATH": "C:\\Program Files\\Microsoft
SDKs\\Azure\\CLI2\\wbin;${env:PATH}"
}
}
```

IMP NOTE: PATH should be your path from cmd

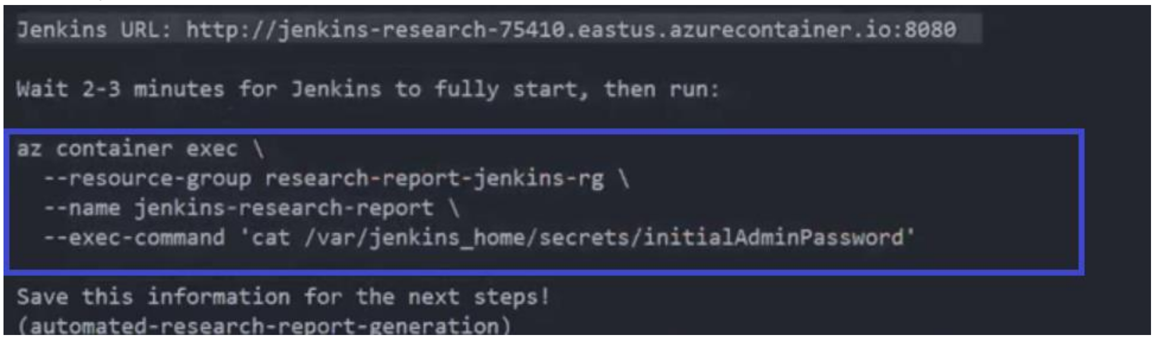
j. **az provider show --namespace Microsoft.ContainerRegistry –query "registrationState"**
   i. It should display as registered then only try the below commands which means running sh files

5. Type below commands in bash terminal to execute the Jenkins
   a. **bash ./azure-deploy-jenkins.sh**
      i. If its saying any name exists just run the **bash complete-cleanup.sh**
   b. After running the above sh file we will get the Jenkins URL which we will access the Jenkins and we will get the command to get the Jenkins password
   we will get like below which is the jenkins url click on it. I will get my jenkins



```
Jenkins URL: http://jenkins-research-75410.eastus.azurecontainer.io:8080

Wait 2-3 minutes for Jenkins to fully start, then run:

az container exec \
  --resource-group research-report-jenkins-rg \
  --name jenkins-research-report \
  --exec-command 'cat /var/jenkins_home/secrets/initialAdminPassword'

Save this information for the next steps!
(automated-research-report-generation)
```

   we will get password from below hightlighted rectangle this is the command for Jenkins password



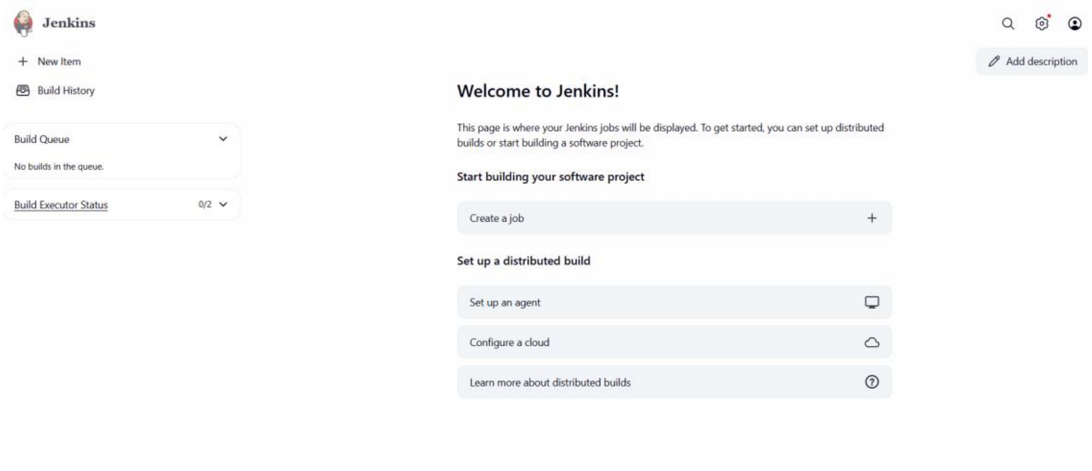
```
Jenkins URL: http://jenkins-research-75410.eastus.azurecontainer.io:8080

Wait 2-3 minutes for Jenkins to fully start, then run:

az container exec \
  --resource-group research-report-jenkins-rg \
  --name jenkins-research-report \
  --exec-command 'cat /var/jenkins_home/secrets/initialAdminPassword'

Save this information for the next steps!
(automated-research-report-generation)
```

   NOTE: Use your URL and to get the password use your command

   c. admin is your username and password you will get from above command
      i. username: admin
      ii. password: axrgjotewdvnmllydsss
      iii. Give username and password in the Jenkins url.

iv. Select the “setup plugins” in the Jenkins url

v. After completing above steps our Jenkins url looks like this



No builds in the queue.

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Create a job

Set up a distributed build

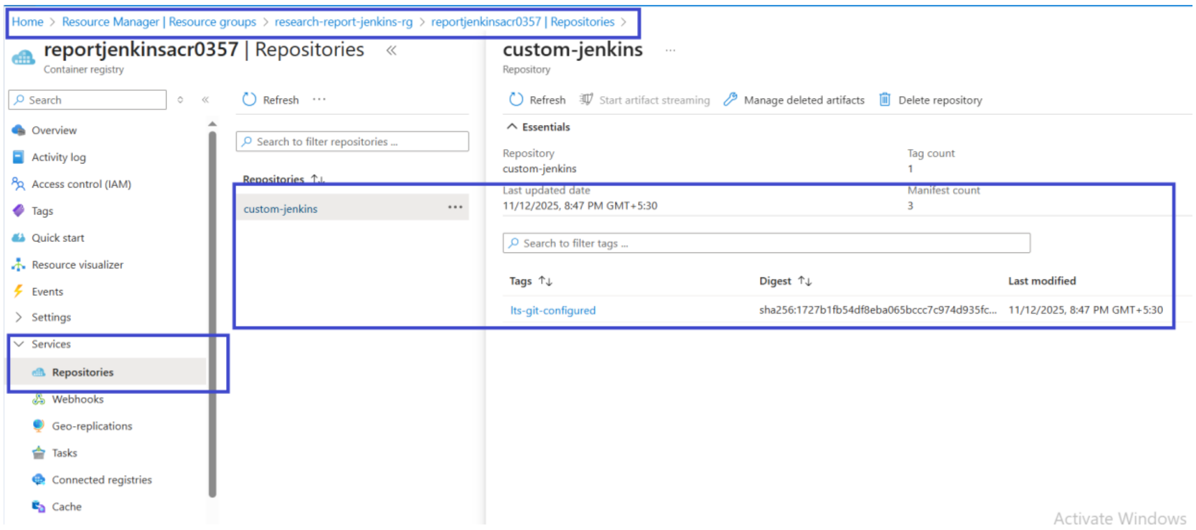
Set up an agent

Configure a cloud

Learn more about distributed builds

d. To see the repository goto Home=>ResourceManager=>ResourceGroups=>research-report-jenkins-rg=>reportjenkinsacrDDMM=>Services=>Repositories

i. This image only will run in the container



Repository: custom-jenkins

Tag count: 1

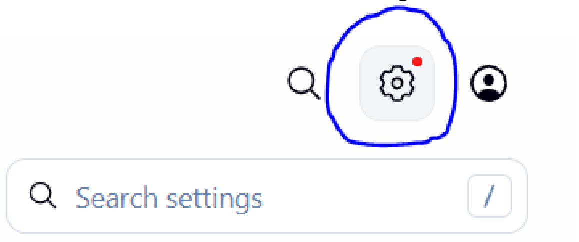
Last updated date: 11/12/2025, 8:47 PM GMT+5:30

Manifest count: 3

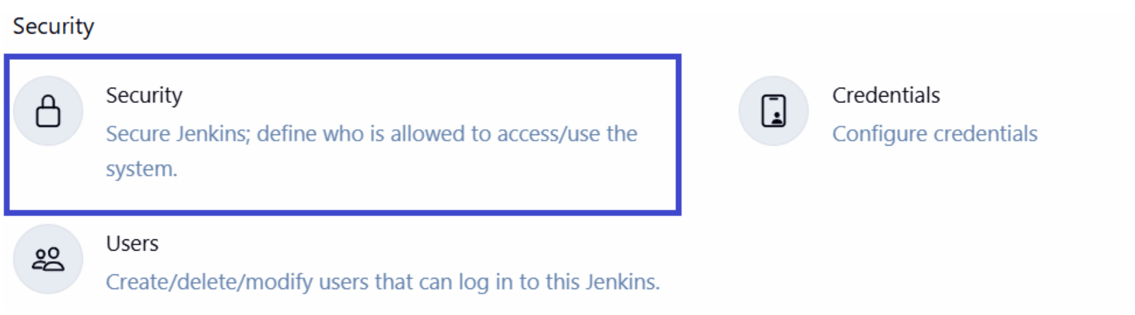
Tags	Digest	Last modified
lts-git-configured	sha256:1727b1fb54df8eba065bccc7c974d935fc...	11/12/2025, 8:47 PM GMT+5:30

e. We have to add the global credentials to make the connectivity between Jenkins and Azure to make the connectivity perform below steps.

i. Click on Settings button in Jenkins=>click on Manage Jenkins



ii. Open the Security

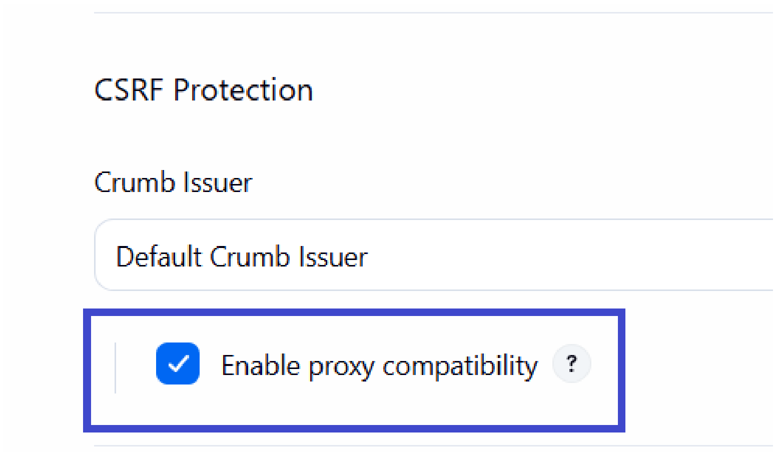

Security

Security
Secure Jenkins; define who is allowed to access/use the system.

Credentials
Configure credentials

Users
Create/delete/modify users that can log in to this Jenkins.

ii click on Enable proxy compability checkbox finally click on save then Apply

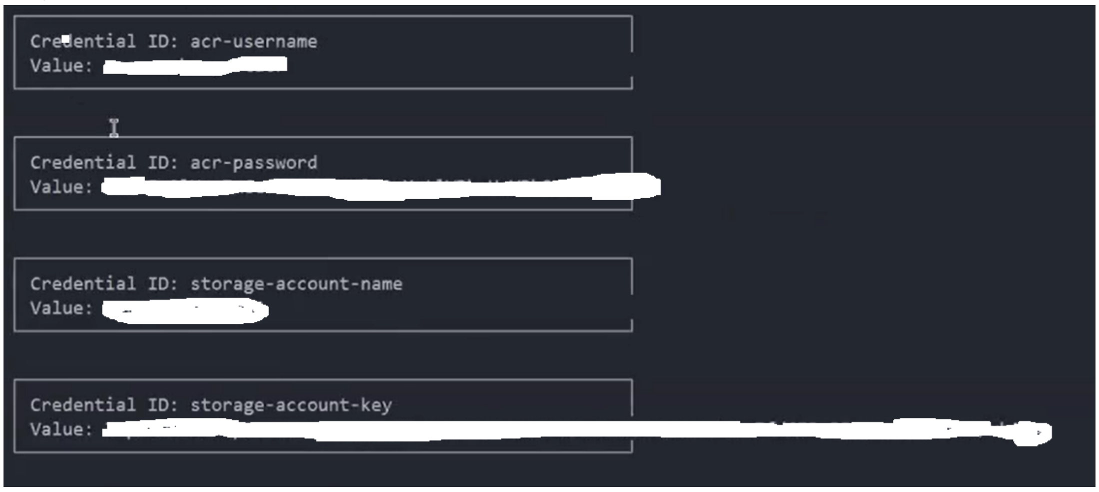

CSRF Protection

Crumb Issuer

Default Crumb Issuer

✓ Enable proxy compatibility

6. Now we will setup the infra(This is my Azure Infrastructure before we are setting up the Jenkins infrastructure)
   a. **bash setup-app-infrastructure.sh**
   b. After setting up infra we will get below credentials in bash terminal we have to setup below credentials in the Jenkins. These credentials will be used to make the connectivity between the Jenkins and Azure



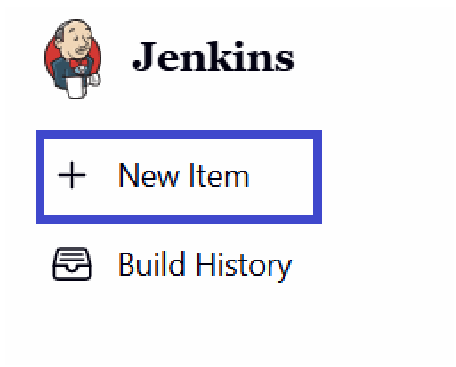
```
Credential ID: acr-username
Value: [redacted]

Credential ID: acr-password
Value: [redacted]

Credential ID: storage-account-name
Value: [redacted]

Credential ID: storage-account-key
Value: [redacted]
```

   c. To setup Goto Jenkins URL=> Goto Settings=>under the Security you will see Credentials=> click on credentials=>click on global=>click on “Add Credentials”=> under the Kind=>select the secret text
      i. Give all above key and value as secret then click on Secret
   d. Type below command in bash terminal

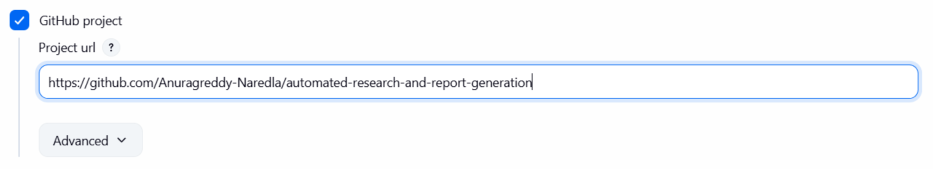
```
az ad sp create-for-rbac \
  --name "jenkins-research-report-sp" \
  --role Contributor \
  --scopes /subscriptions/$(az account show --query id -o tsv)
```

- With the above command we will get the "appId","displayName","password","tenant" add these in Jenkins credentials in the below way
  - azure-client-id: appId
  - azure-tenant-id: tenant
  - azure-client-secret: password
  - azure-subscription-id:
- This will create a service priniciple with the role based access
- For running our application we need below credentials also
  - Openai-api-key
  - Google-api-key
  - Groq-api-key
  - Tavily-api-key
  - Llm-provider=openai

7. Now we have to run our Jenkins pipeline and add the webhook in the github along with this we will build the dockerimage
   a. For configure the Jenkins pipeline go with the Jenkins page there we will get the NewItem
      i.

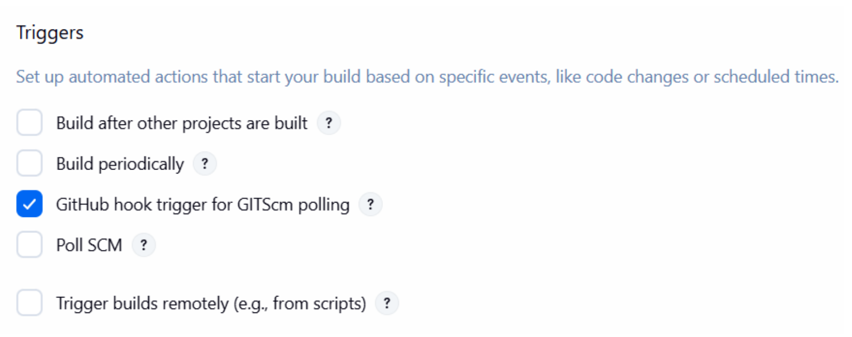


      ii. Click on New Item give Name
      iii. Choose the "pipeline" then finally click on "ok"
      iv. After click on "ok", Select the "Github Project"



https://github.com/Anuragreddy-Naredla/automated-research-and-report-generation

         1.

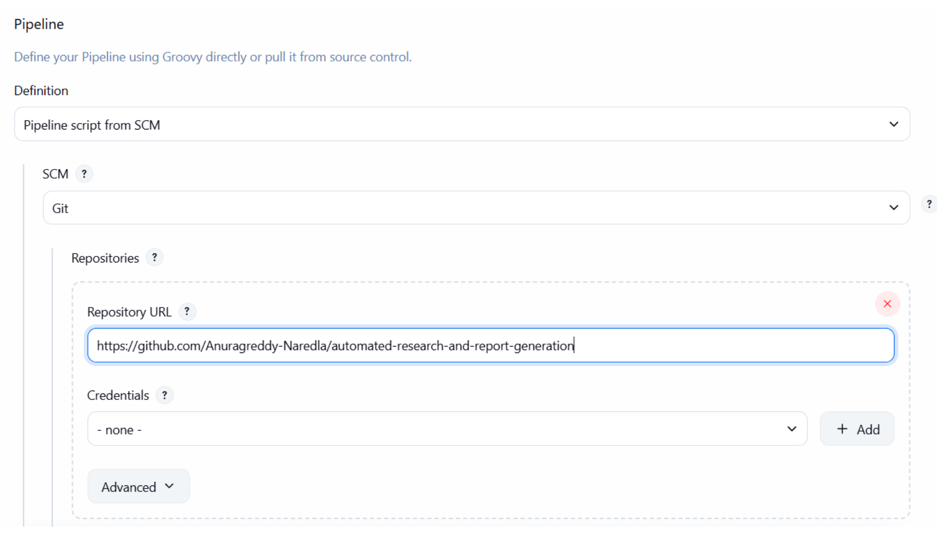


Triggers

Set up automated actions that start your build based on specific events, like code changes or scheduled times.

- [ ] Build after other projects are built
- [ ] Build periodically
- [✓] GitHub hook trigger for GITScm polling
- [ ] Poll SCM
- [ ] Trigger builds remotely (e.g., from scripts)

2.



Pipeline

Define your Pipeline using Groovy directly or pull it from source control.

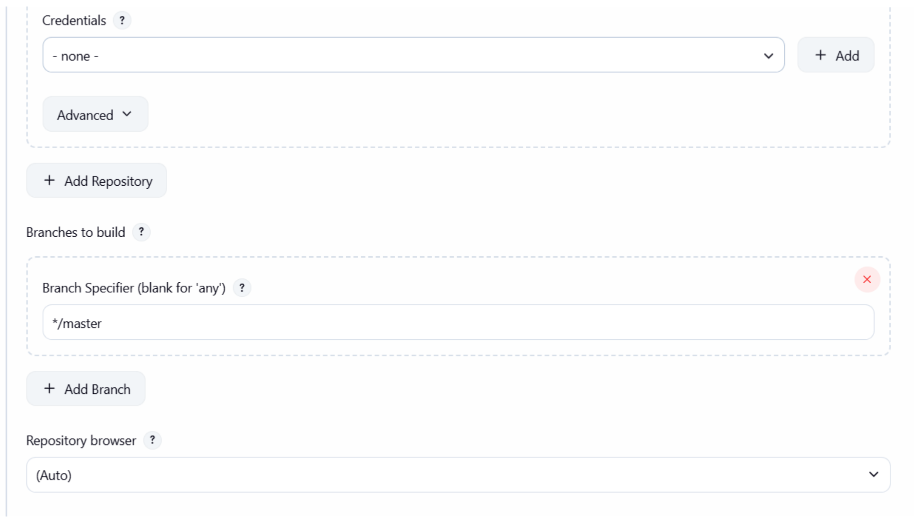
Definition: Pipeline script from SCM

SCM: Git

Repositories

Repository URL: https://github.com/Anuragreddy-Naredla/automated-research-and-report-generation

Credentials: - none -



Credentials: - none -

Branches to build

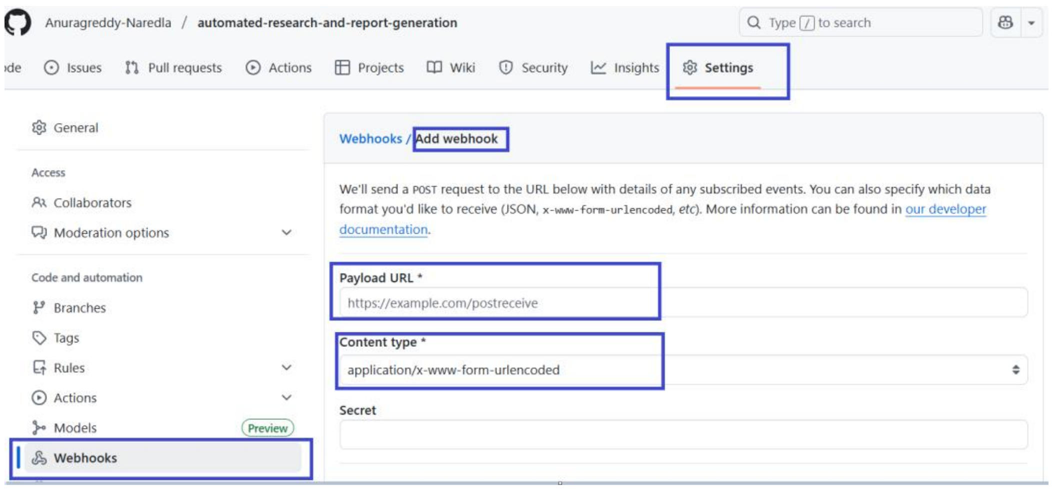
Branch Specifier (blank for 'any'): */master

Repository browser: (Auto)

3.

4. Finally click on “Apply” and “Save”

b. Add the webhook in the github

i. 
Anuragreddy-Naredla / automated-research-and-report-generation

Webhooks / Add webhook

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in our developer documentation.

Payload URL *: https://example.com/postreceive

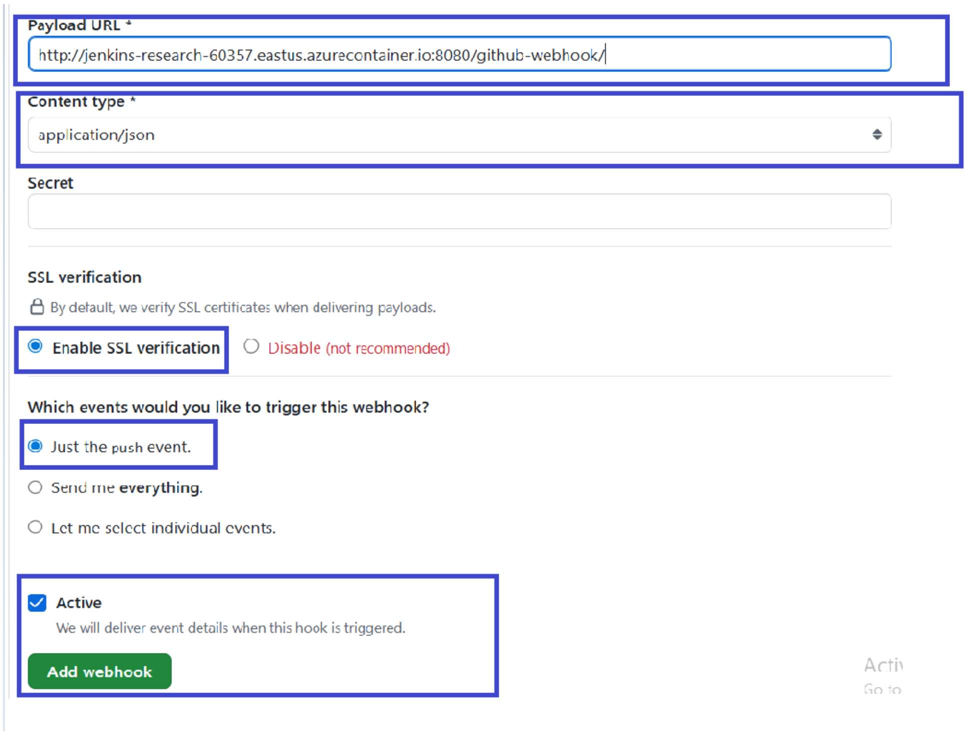
Content type *: application/x-www-form-urlencoded

Secret:

ii. Goto your Project URL
iii. Goto Settings
iv. Select the webhooks
v. Under the webhook select the Add webhook
vi. Now we have to write the payload URL and content type
vii. Run the build-and-push-docker-image.sh

   1. bash ./build-and-push-docker-image.sh [This will build the dockerimage for our application and push to it ACR hub.]

viii. After running the sh file successfully go and perform below steps
ix. Payload URL will be your Jenkins url along with this add the github-webhook
x. Content type should be application/json
xi. Finally click on AddWebhook

   1. 

Payload URL *: http://jenkins-research-60357.eastus.azurecontainer.io:8080/github-webhook/

Content type *: application/json

Secret:

SSL verification

By default, we verify SSL certificates when delivering payloads.

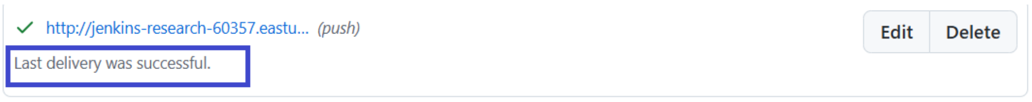
[✓ Enable SSL verification] Disable (not recommended)

Which events would you like to trigger this webhook?

[✓ Just the push event.]
Send me everything.
Let me select individual events.

[✓ Active]
We will deliver event details when this hook is triggered.

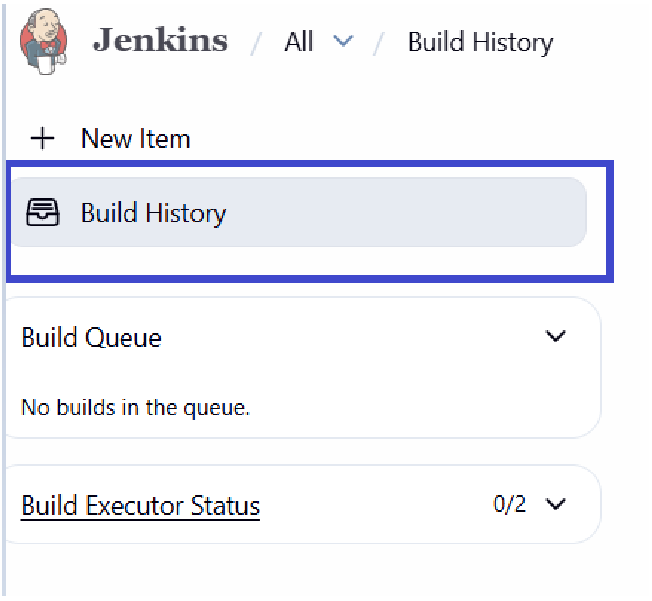
8. Change any code then commit the code to github
    a. You should see like below



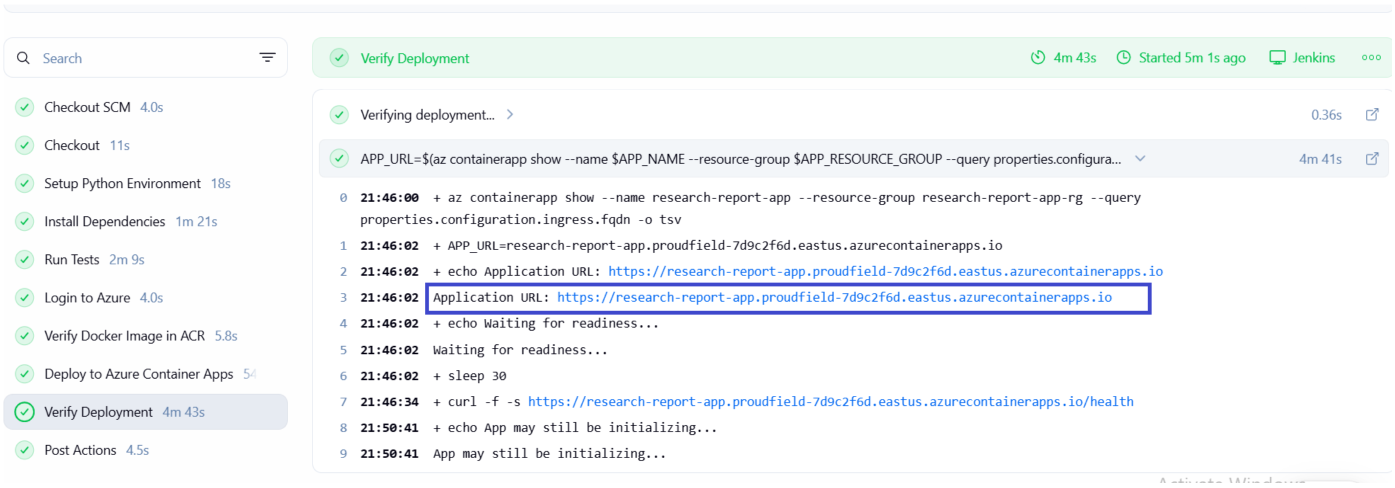
http://jenkins-research-60357.eastu... (push)

Last delivery was successful.

        i.
    b. To check



        i.
    c. Below is the application URL from Azure we will get in the Jenkins pipeline.

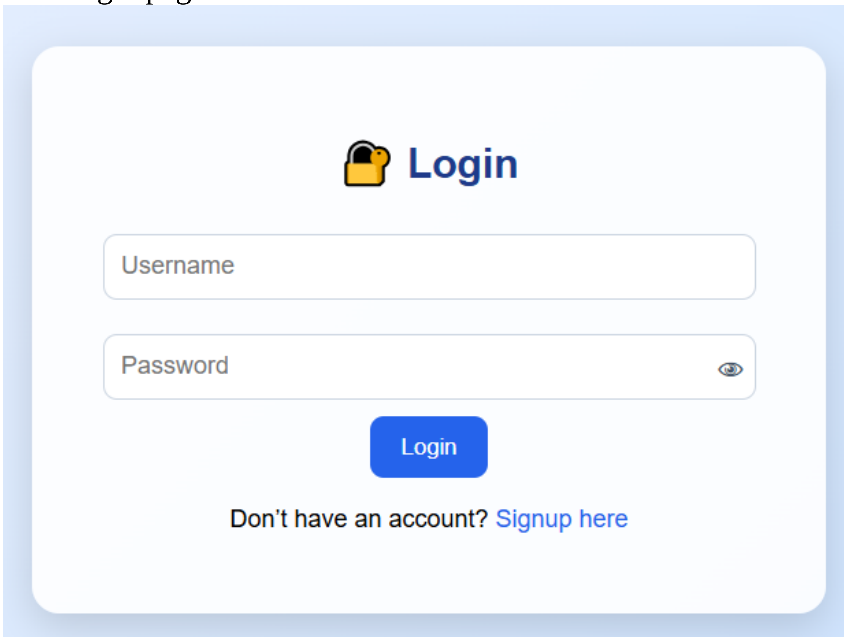


APP_URL=$(az containerapp show --name $APP_NAME --resource-group $APP_RESOURCE_GROUP --query properties.configura...

```
21:46:00  + az containerapp show --name research-report-app --resource-group research-report-app-rg --query
properties.configuration.ingress.fqdn -o tsv
21:46:02  + APP_URL=research-report-app.proudfield-7d9c2f6d.eastus.azurecontainerapps.io
21:46:02  + echo Application URL: https://research-report-app.proudfield-7d9c2f6d.eastus.azurecontainerapps.io
21:46:02  Application URL: https://research-report-app.proudfield-7d9c2f6d.eastus.azurecontainerapps.io
21:46:02  + echo Waiting for readiness...
21:46:02  Waiting for readiness...
21:46:02  + sleep 30
21:46:34  + curl -f -s https://research-report-app.proudfield-7d9c2f6d.eastus.azurecontainerapps.io/health
21:50:41  + echo App may still be initializing...
21:50:41  App may still be initializing...
```

d. Below are the screenshots of my application.

i. User login page



Login

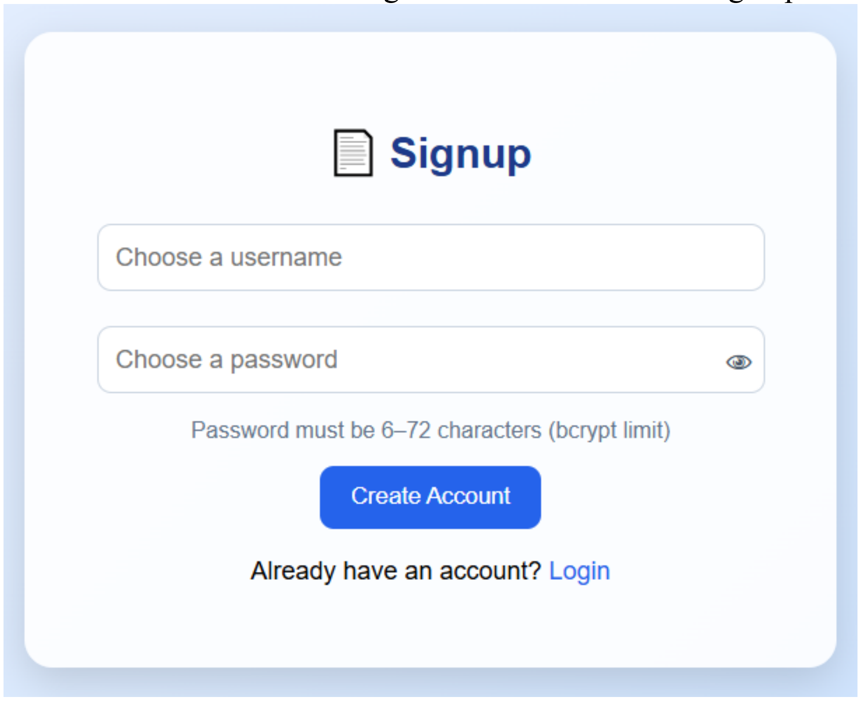
Username

Password

Login

Don't have an account? Signup here

ii. If the user doesn't have the login credentials. User can sign up.



Signup

Choose a username

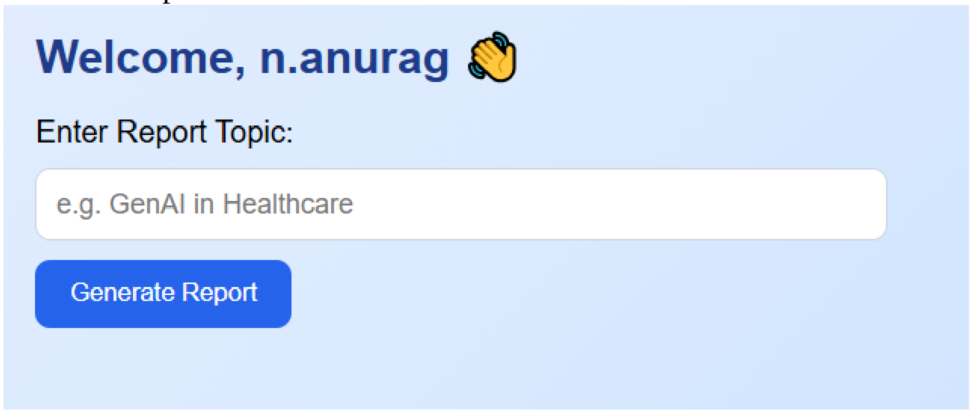
Choose a password

Password must be 6–72 characters (bcrypt limit)

Create Account

Already have an account? Login

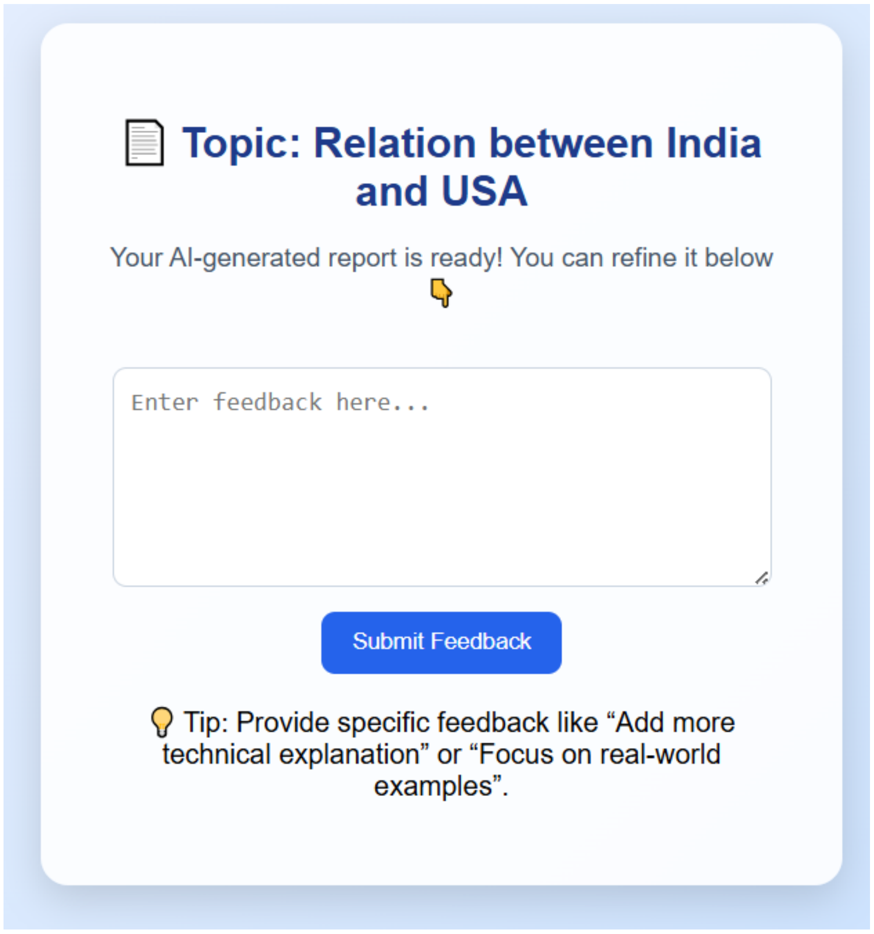
iii. User can give the specific topic in the field. Then user have to click on Generate Report


Welcome, n.anurag 👋

Enter Report Topic:

e.g. GenAI in Healthcare

iv. User can enter the feedback here then click on Submit feedback.

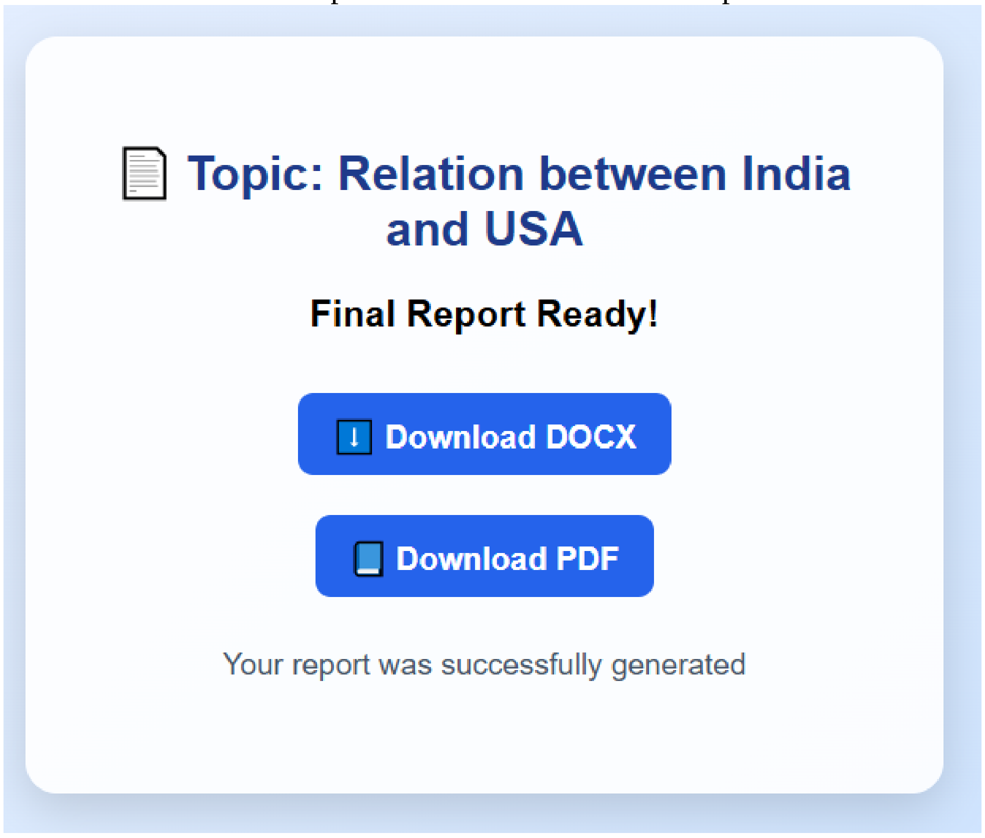

📄 Topic: Relation between India and USA

Your AI-generated report is ready! You can refine it below 👇

Enter feedback here...

💡 Tip: Provide specific feedback like “Add more technical explanation” or “Focus on real-world examples”.

v. User can download the report in the format of docx and pdf file.



📄 Topic: Relation between India and USA

Final Report Ready!

Your report was successfully generated