

Project Github: <https://github.com/sunnysavita10/automated-research-report-generation>

Command for UV:

1. `uv --version`
2. If uv is not there then install the uv
3. `pip install uv`
4. Once you got the uv
5. Write this command- `uv init <project_name>`
6. Open the project in VS Code

You can also take an alternative approach

1. First, create a new folder
2. Open it inside VS Code
3. Over the VS Code terminal, write this command: `uv init`

To create a virtual environment, follow the command below

You can simply write-> `uv venv` for creating a virtual environment

It will create a virtual env with the default name `.venv` (this will be the name of the env)

If you want to create with a custom name, follow the command below

`uv python list`

`uv venv <your-env-name> --python <your-python-version>`

Activate the environment

`.venv\Scripts\activate.bat`

Note: If it is a custom env then copy the path from that folder

`uv add -r requirements.txt`

`uv add ipykernel`

Website for getting Tavily API key for search the web

<https://app.tavily.com/home>

#26-Oct-2025

##steps for running the application

1. Install the requirements.txt and update your virtual env
uv pip install -r requirements.txt
2. command for running the api
uvicorn research_and_analyst.api.main:app --reload
uvicorn research_and_analyst.api.main:app --host 0.0.0.0 --port 8000

TO check the record in the [user.db](#) install this extension in vscode: **SQLite Viewer**

#1st of NOV

Installation link for the Azure CLI: <https://learn.microsoft.com/en-us/cli/azure/install-azure-cli-windows?view=azure-cli-latest&pivots=msi>

Verify the installation of Azure CLI by writing below in your terminal

az --version

Deployment command:

Microsoft Azure Portal: <https://azure.microsoft.com/en-us/get-started/azure-portal>

az login

or

az login --username <your_mail>

It will give you the error

Error in the red character

It will ask to set up the subscription

az login --tenant <KEEP_YOUR_TENANT_ID> <you will get it from console open the subscription, and copy from there the name of the tenant on the console is **Parent management group**>

az login --tenant <KEEP_YOUR_TENANT_ID> --use-device-code

az account set --subscription <KEEP_YOUR_SUBSCRIPTION_ID>

az account list --output table

az account list

az account show

az group list

az login

If nothing is working, run this

```
az logout
az account clear
rmdir /s /q "%USERPROFILE%\azure"
az login --use-device-code
az account list -o table
```

```
bash ./azure-deploy-jenkins.sh
bash ./build-and-push-docker-image.
Sh
```

Stage 1 of the deployment:

1. Setup the azure account
2. Setup the azure cli
3. Create a file and write the config inside

Stage 2 of the deployment:

1. Run the shell script
2. Set up the Azure resource
3. Set up the Jenkins server
4. Build the image
5. Push it to the ACR
6. Deploy it to the container

##2nd NOV

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

Follow the steps below for deploying your application:

Jenkins→ open source self-hosted CI/CD tool

1. [azure-deploy-jenkins.sh](#)
2. Dockerfile.jenkins
3. These are the two files that are required for us to set up Jenkins over Azure

What is this .sh→ sh stand for shell script

In .sh file we write all the command which we want to execute over the terminal

Terminal→azure cli

Why bez we have configure az login

Make sure your docker engine is running

Run the .sh file: bash <path-of-your-sh-file>

After running this file you will get two things:

1. **URL: url of your jenkins which is running under the container over the azure**
2. **Password for login:**
3. **If you will do it first time then copy the one time password from terminal and paste it over the jenkins UI**
4. **Then setup the plugins**
5. **It will atleast 5 - 10 mints**
6. **If you will do it second time then**
7. **Use username: admin**
8. **And password you will over the terminal**

If you are getting error: subscription id is not there then check with this command

az login --tenant <KEEP_YOUR_TENANT_ID> <you will get it from console open the subscription, and copy from there the name of the tenant on the console is **Parent management group**>

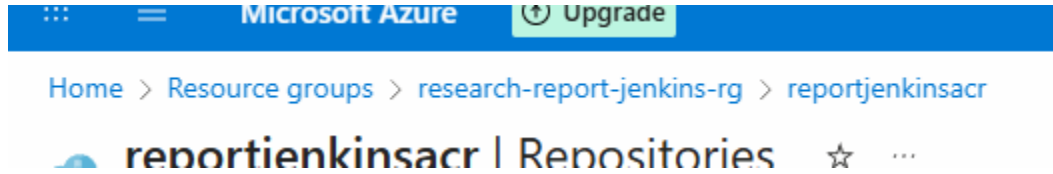
az login --tenant <KEEP_YOUR_TENANT_ID> --use-device-code

az account set --subscription <KEEP_YOUR_SUBSCRIPTION_ID>

az provider register --namespace Microsoft.ContainerRegistry
az provider register -n Microsoft.Storage --subscription <subscription_id>

```
az account list --output table
az account list
az account show
```

After completing everything you will get a image of custom jenkins under this location:

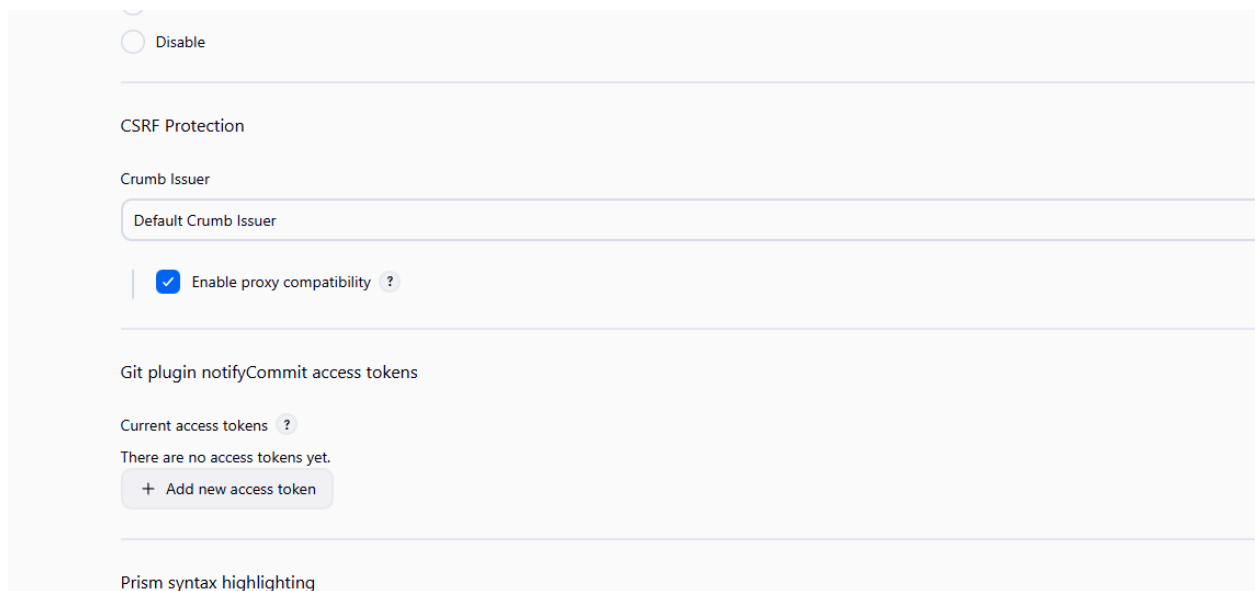


After reaching to this location open service/repository

After open the jenkins tick mark this check box:

Where you will get it:

You will get under **settings**<it is gear icon at right top>/security



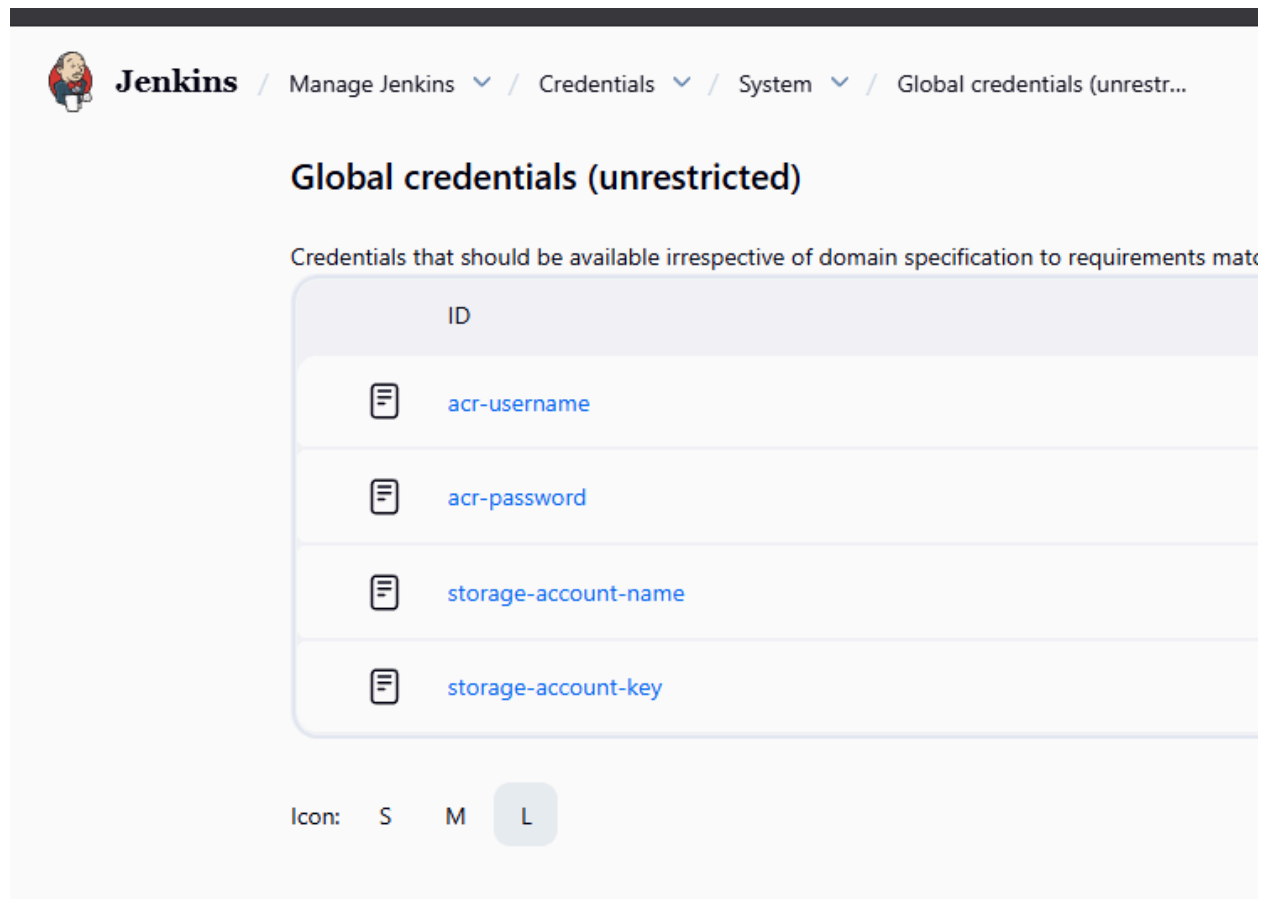
Now the next step setup the infra:

So command is:

```
bash setup-app-infrastructure.sh
```

After this you will get a secrate for making connection between jenkins and azure

Below is the secrate



Jenkins / Manage Jenkins / Credentials / System / Global credentials (unrestricted)

Global credentials (unrestricted)

Credentials that should be available irrespective of domain specification to requirements matching

ID
acr-username
acr-password
storage-account-name
storage-account-key

Icon: S M L

You will get a key and a value both over the terminal

Add it at the appropriate location

Add few more credentials:

You will get those credentials using these commands

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













Use of this command: It will create a Service Principle in azure with role based access

After running you will get a json:

azure-client-id: appld
azure-tenant-id: tenant
azure-client-secret: password
azure-subscription-id: az account show --query id -o tsv

Openai-api-key
Google-api-key
Groq-api-key
Tavily-api-key
Llm-provider

Credentials that should be available irrespective of domain specification to requirements

ID	
	acr-username
	acr-password
	storage-account-name
	storage-account-key
	azure-client-id
	azure-tenant-id
	azure-client-secret
	azure-subscription-id
	OPENAI_API_KEY
	TAVILY_API_KEY
	GROQ_API_KEY
	LLM_PROVIDER

Configure these keys in the Jenkins global variable

The next setup would be to

Build the Docker image from our application and push to it to ACR hub

Below is the command

build-amd-push-docker-image.sh

configure the Jenkins pipeline:

new-item/pipeline

Now follow the video for further configuration

Then add the

Add the webhook in the github: