
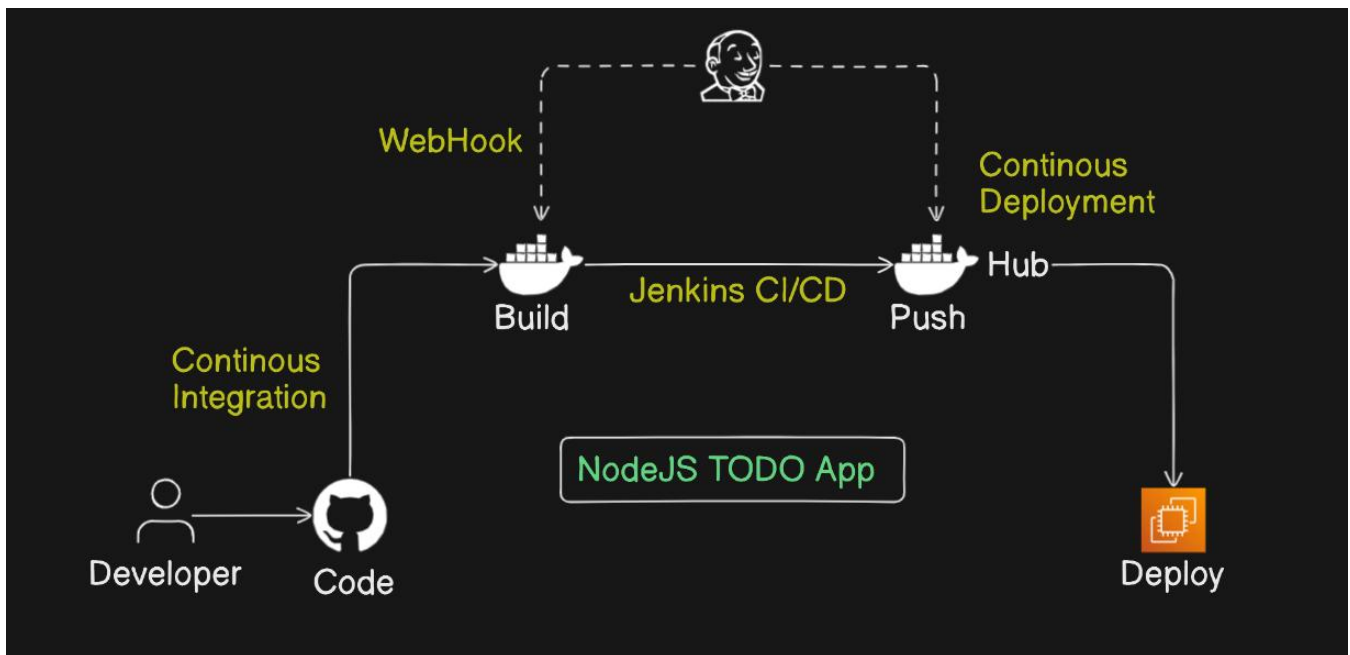


NodeJS application deployment using Jenkins CI/CD

Find detailed blog at -> 

1. Architecture of Project-



2. Setup GitHub repository for source code and CI

- GitHub repository used -> [node-todo](#)
- Note : If you need detailed overview you can also view this [Blog](#).

3. Setup Job on Jenkins for application

- Create EC2 instance for Jenkins server
- Update repository packages
- Create Job,

Instances | EC2 | us-east-1 x New Item [Jenkins] x +

100.24.9.64:8080/view/all/newJob

Jenkins

Search (CTRL+K)

Dashboard > All > New Item

New Item

Enter an item name

node-todo-app

Select an item type

- Freestyle project**
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
- Pipeline**
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
- Multi-configuration project**
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
- Folder**
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different

OK

Instances | EC2 | us-east-1 x node-todo-app Config [Jenkins] x +

100.24.9.64:8080/job/node-todo-app/configure

Dashboard > node-todo-app > Configuration

Configure

General

Advanced Project Options

Pipeline

General

Enabled

Description

This is an demo application build using NodeJS

Plain text [Preview](#)

☐ Discard old builds ?

☐ Do not allow concurrent builds

☐ Do not allow the pipeline to resume if the controller restarts

☒ GitHub project

Project url ?

https://github.com/ojasjawale/node-todo-cicd.git

Advanced

Save Apply

- Add basic script for testing,

```
pipeline{
  agent any;

  stages{
    stage("Code"){
      steps{
        echo "Source code fetched....."
      }
    }
    stage("Build"){
      steps{
        echo "Image created SUCCESS....."
      }
    }
  }
}
```

```

    }
  }
  stage("Push"){
    steps{
      echo "Pushing to dockerhub....."
    }
  }
  stage("Deploy"){
    steps{
      echo "Deployment done....."
    }
  }
}
}
}

```

The screenshot shows the Jenkins web interface for a pipeline named 'Node-App'. The left sidebar contains navigation links: Status, Changes, Build Now, Configure, Delete Pipeline, GitHub, Stages, Rename, Pipeline Syntax, and GitHub Hook Log. The main area displays the pipeline status as 'Node-App' with a green checkmark. Below this, it shows the project name 'node-todo-app' and a description 'This is an demo application build using NodeJS'. A section titled 'Permalinks' lists four links: 'Last build (#2), 3 min 14 sec ago', 'Last stable build (#2), 3 min 14 sec ago', 'Last successful build (#2), 3 min 14 sec ago', and 'Last completed build (#2), 3 min 14 sec ago'. At the bottom left, a 'Builds' section shows a list of builds with a filter input and two builds listed: '#3 3:25 PM' and '#2 3:21 PM', both with green status icons.

4. Setup security credentials in Jenkins

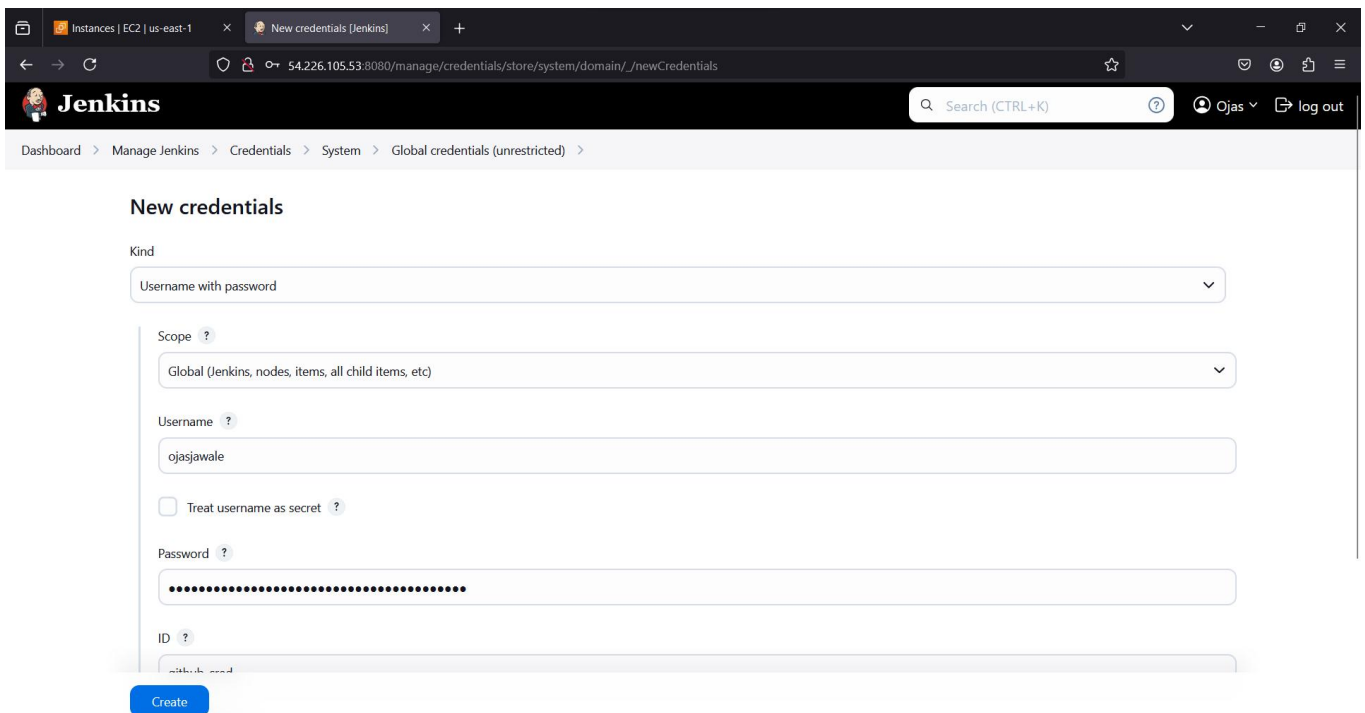
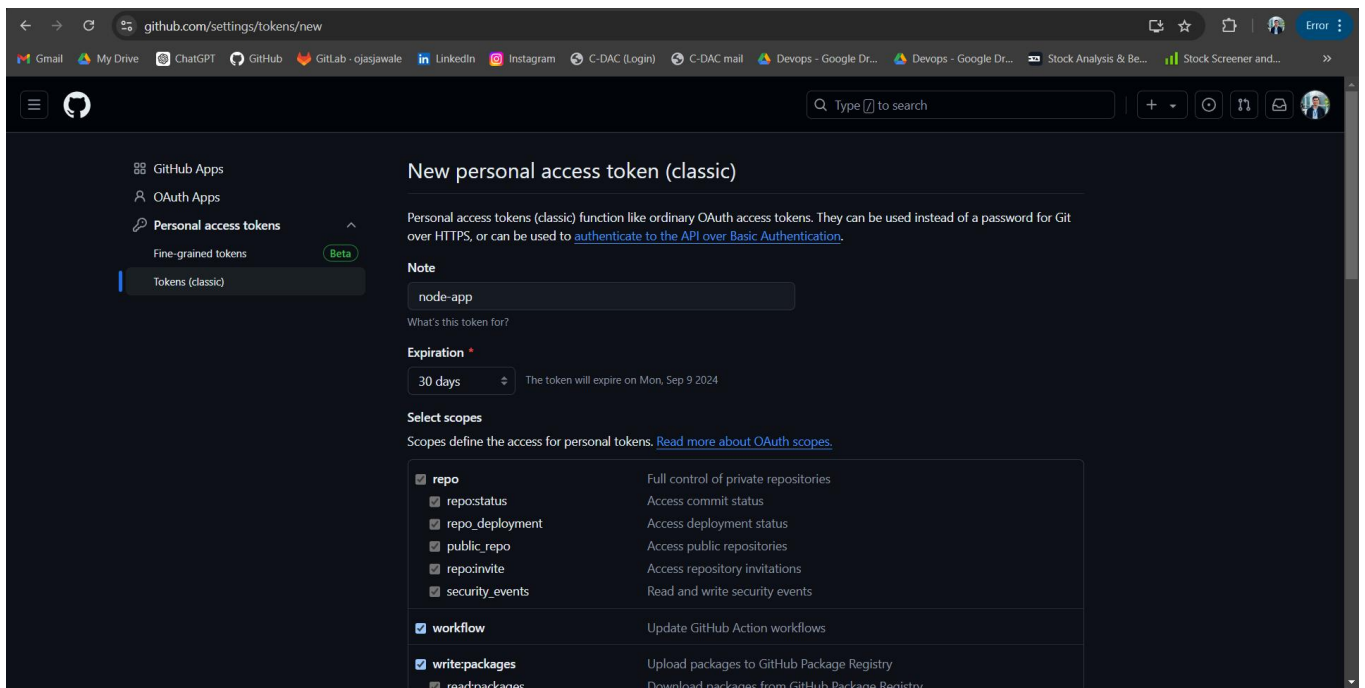
I. GitHub credentials

The screenshot shows the Jenkins 'Global credentials (unrestricted)' page. The breadcrumb trail is 'Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted)'. The page title is 'Global credentials (unrestricted)' with an 'Add Credentials' button. Below the title, it states 'Credentials that should be available irrespective of domain specification to requirements matching.' A table lists the credentials:

ID	Name	Kind	Description
node1	node1	SSH Username with private key	node1

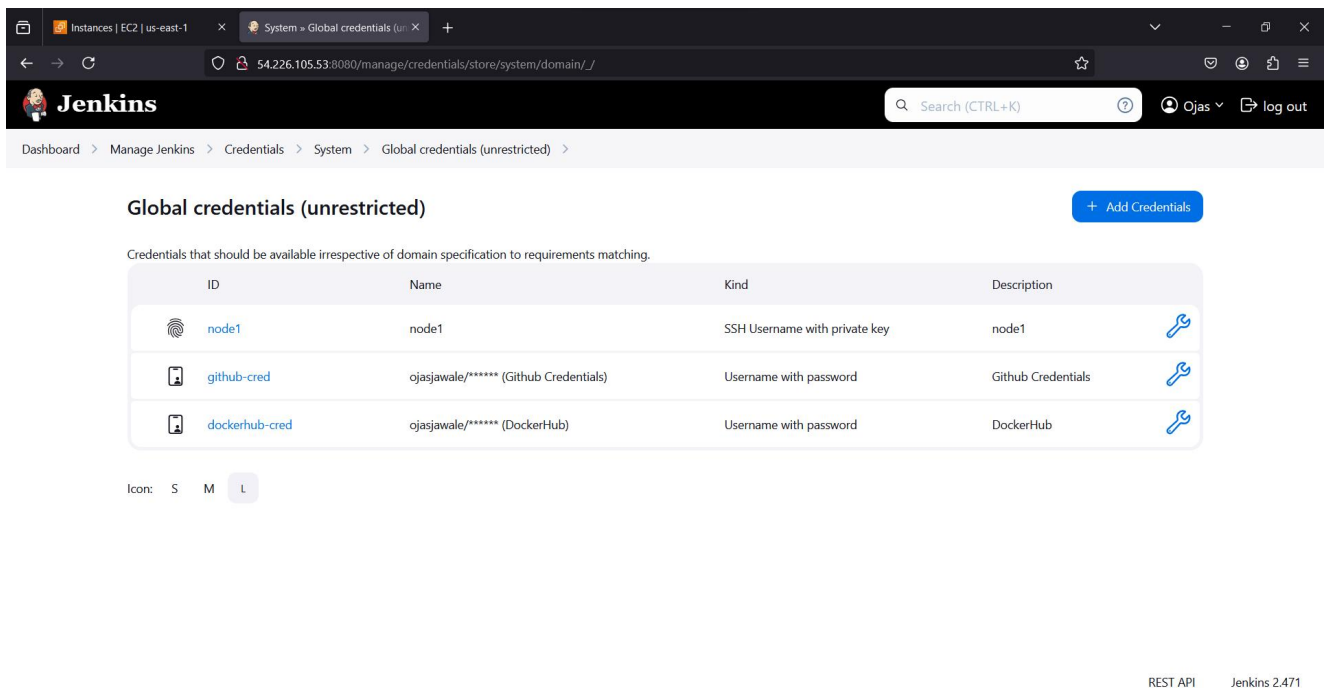
At the bottom left, there are icons for 'Icon: S M L'.

- Go to Github.com and create Personal Access Token.



II. DockerHub Credentials

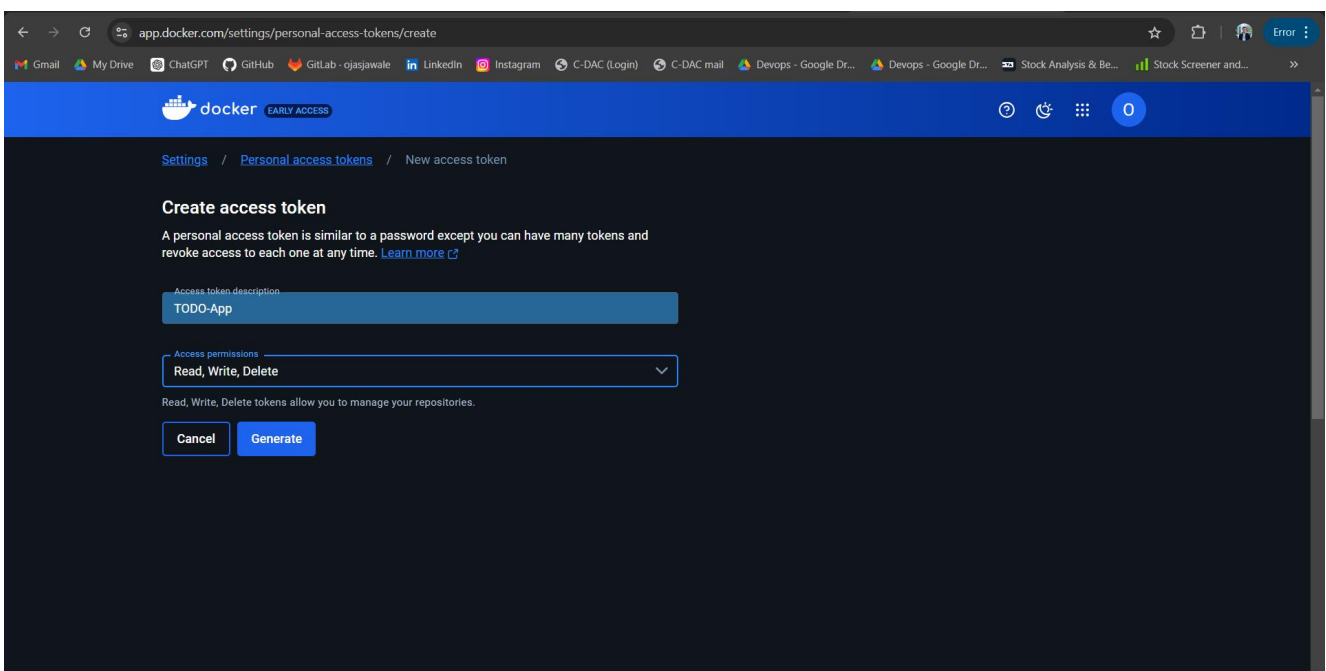
Go to, Dashboard -> Manage Jenkins -> Credentials -> System -> Global credentials -> Add Credentials



The screenshot shows the Jenkins web interface. The browser address bar displays the URL `54.226.105.53:8080/manage/credentials/store/system/domain/_/`. The Jenkins logo and navigation menu are visible at the top. The breadcrumb trail is: Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >. The page title is "Global credentials (unrestricted)". A blue button labeled "+ Add Credentials" is in the top right. Below the title, a message states: "Credentials that should be available irrespective of domain specification to requirements matching." A table lists the credentials:

ID	Name	Kind	Description
node1	node1	SSH Username with private key	node1
github-cred	ojasjawale/***** (Github Credentials)	Username with password	Github Credentials
dockerhub-cred	ojasjawale/***** (DockerHub)	Username with password	DockerHub

At the bottom right, the text "REST API Jenkins 2.471" is visible.

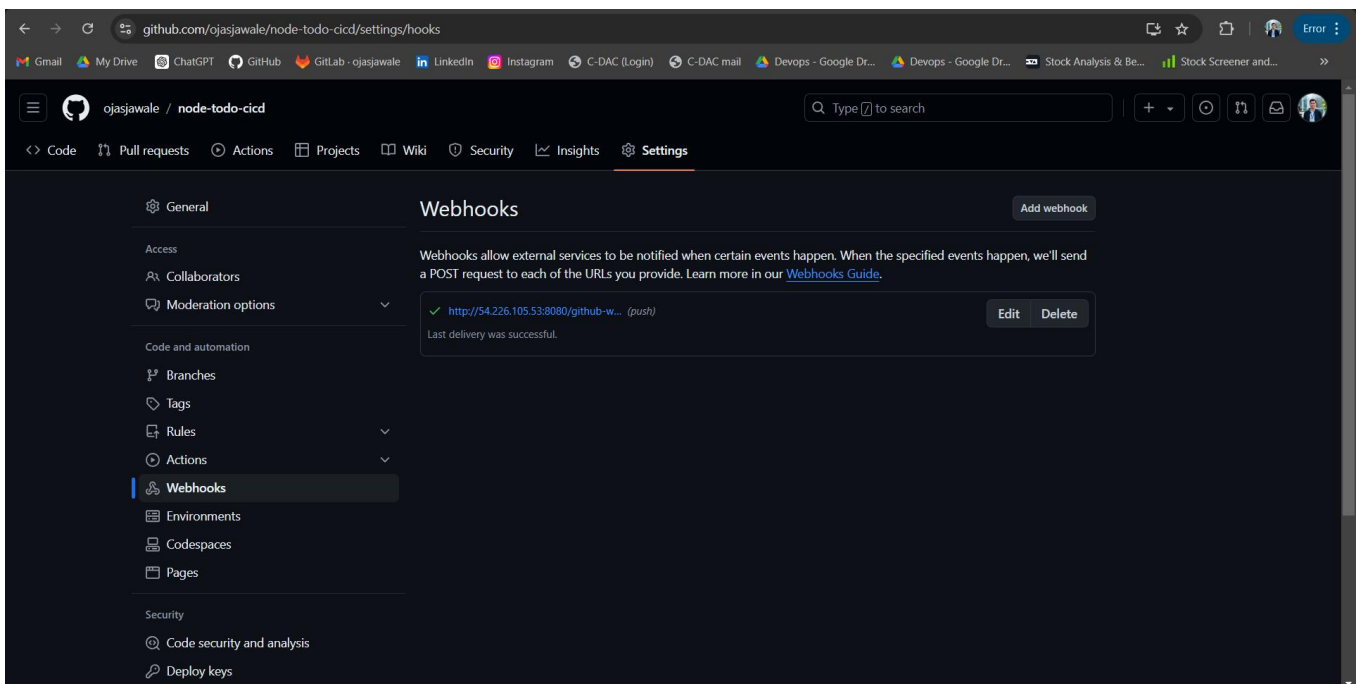
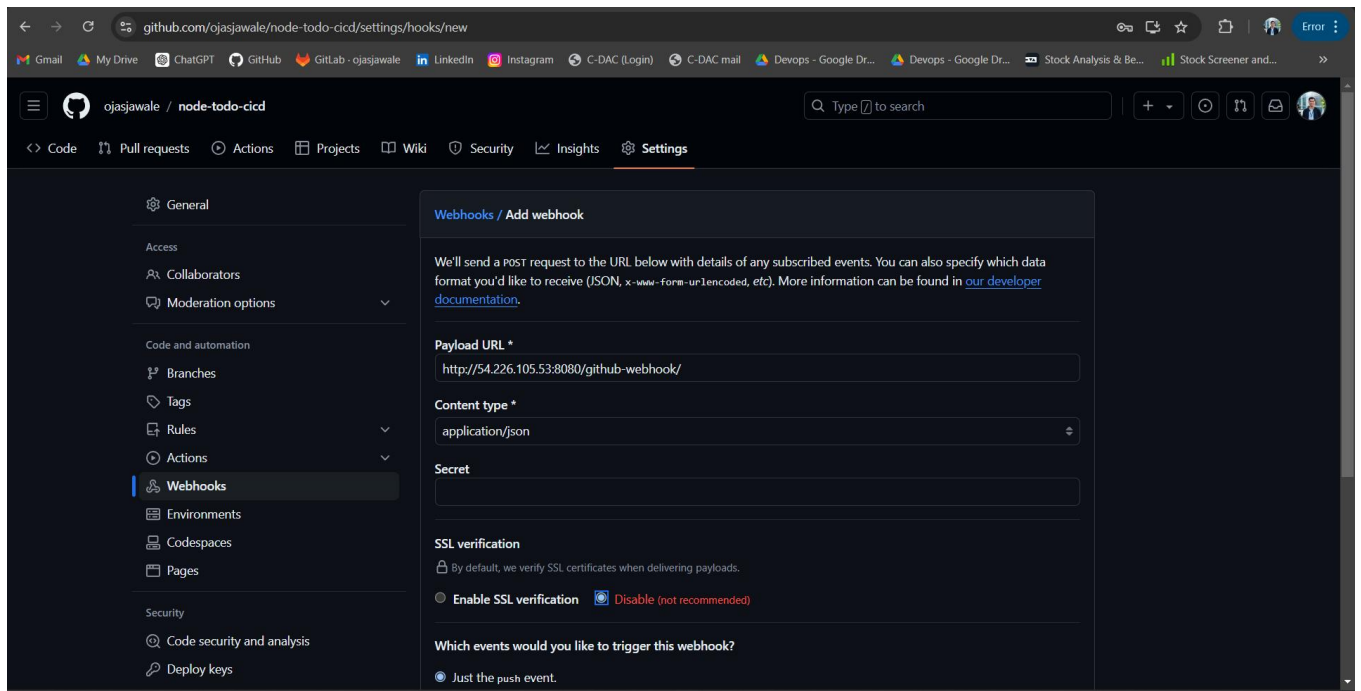


The screenshot shows the Docker web interface for creating a personal access token. The browser address bar displays the URL `app.docker.com/settings/personal-access-tokens/create`. The Docker logo and navigation menu are visible at the top. The breadcrumb trail is: Settings / Personal access tokens / New access token. The page title is "Create access token". A message states: "A personal access token is similar to a password except you can have many tokens and revoke access to each one at any time. [Learn more](#)". Below the message, there is a form with two fields: "Access token description" with the value "TODO-App" and "Access permissions" with the value "Read, Write, Delete". Below the form, there are two buttons: "Cancel" and "Generate".

5. Setup GitHub WebHook

Go to repository -> Setting -> Webhook -> Add WebHook

Payload URL -: `http://IP_of_Jenkins:8080/github-webhook/`



6. Setup CI/CD pipeline

```
pipeline{
  agent any;

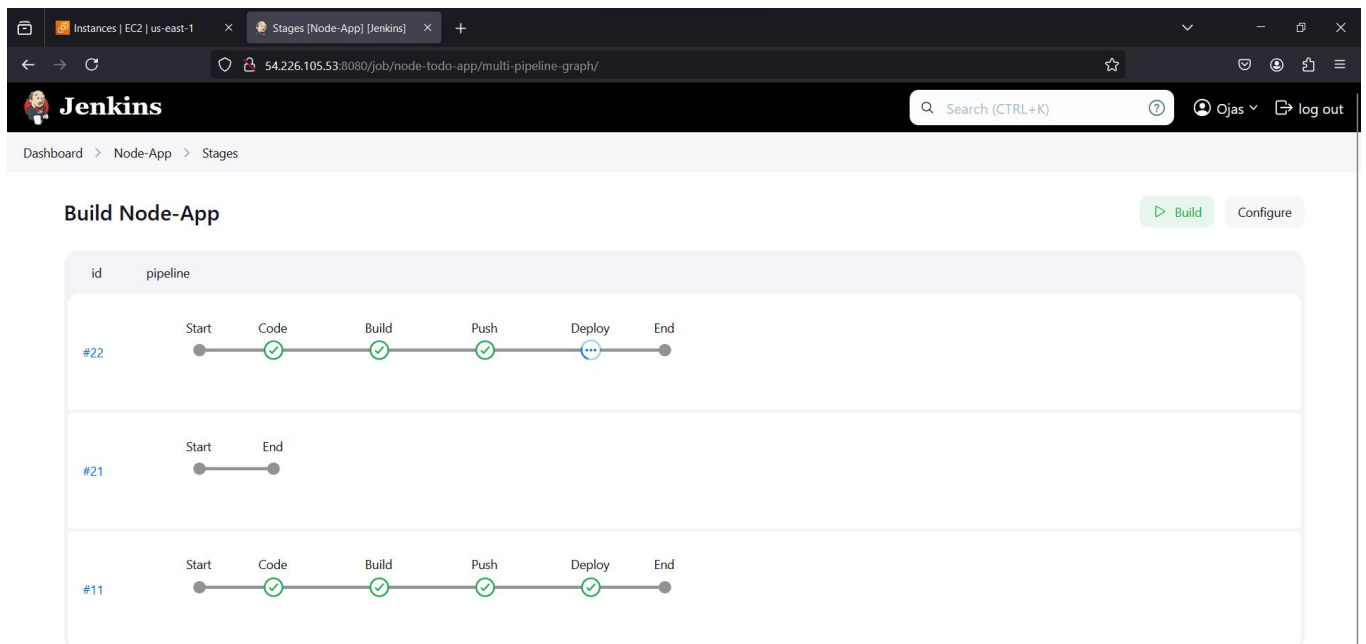
  stages{
    stage("Code"){
      steps{
        echo "Pulling source code from GitHub...."
        git url: "https://github.com/ojasjawale/node-todo-cicd.git/", branch: "master"
      }
    }
    stage("Build"){
      steps{
        echo "Image building started....."
```

```

        sh 'docker build -t node-app:v1 .'
        echo "Image Build SUCCESS....."
    }
}
stage("Push"){
    steps{
        withCredentials([usernamePassword(credentialsId: 'dockerhub-cred',
passwordVariable: 'DOCKER_PASS', usernameVariable: 'DOCKER_USER')]) {
            sh 'docker login -u ${DOCKER_USER} -p ${DOCKER_PASS}'
            sh 'docker image tag node-app:v1 ${DOCKER_USER}/node-app:v1'
            sh 'docker push ${DOCKER_USER}/node-app:v1'
            echo "Image pushed to DockerHub successfully....."
        }
    }
}
stage("Deploy"){
    steps{
        echo "Deployment In-progress....."
        sh 'docker stop node-app && docker rm node-app'
        sh 'docker run --name node-app -p 8000:8000 ojasjawale/node-app:v1'
        echo "Deployment SUCCESS....."
    }
}
}
}
}

```

- Build the Job,



The screenshot shows the Jenkins web interface for a pipeline named 'Build Node-App'. The pipeline is currently in progress, with the 'Deploy' stage being the active stage. The pipeline consists of six stages: Start, Code, Build, Push, Deploy, and End. The 'Code', 'Build', and 'Push' stages are marked with green checkmarks, indicating they have completed successfully. The 'Deploy' stage is marked with a blue circle containing three dots, indicating it is currently in progress. The 'Start' and 'End' stages are marked with grey dots. The pipeline is displayed in a graph view with a table-like structure below it.

id	pipeline
#22	Start → Code (✓) → Build (✓) → Push (✓) → Deploy (⋯) → End
#21	Start → End
#11	Start → Code (✓) → Build (✓) → Push (✓) → Deploy (✓) → End

Instances | EC2 | us-east-1 x Build log [#22] [Jenkins] x +

54.226.105.53:8080/job/node-todo-app/22/pipeline-console/

Jenkins Search (CTRL+K) Ojas log out

Dashboard > Node-App > #22 > Pipeline Console

Build #22 Rebuild Overview Configure ...

In progress 7 min 42 sec ago in 7 min 42 sec and counting

- Code
- Build
- Push
- Deploy

Stage 'Deploy'

- Started 7 min 26 sec ago
- Queued 0 ms
- Took 7 min 26 sec
- Running
- [View as plain text](#)

Deployment In-progress..... 23 ms [Print Message](#)

docker run --name node-app -p 8000:8000 ojasjawale/node-app:v1 7 min 26 sec [Print Message](#)

Shell Script

```
0 + docker run --name node-app -p 8000:8000 ojasjawale/node-app:v1
1 Todolist running on http://0.0.0.0:8000
```

Jenkins 2.471

- Access application,

Instances | EC2 | us-east-1 x Build log [#22] [Jenkins] x Todo List APP test x +

54.226.105.53:8000/todo

Todo List - Made for Batch 7

Hey, What are you doing? [Print Message](#)

What should I do?

Add

Ojas Jawale

