

SESSION 4 – HANDSON

YOUTUBE – <https://www.youtube.com/praveensingampalli>

INSTAGRAM - <https://www.instagram.com/singam4devops/>



<https://github.com/DEVOPS-WITH-WEB-DEV/jenkins-shared-library1.git>

<https://github.com/DEVOPS-WITH-WEB-DEV/spring-cloud-kubernetes.git>

PREREQUISITES

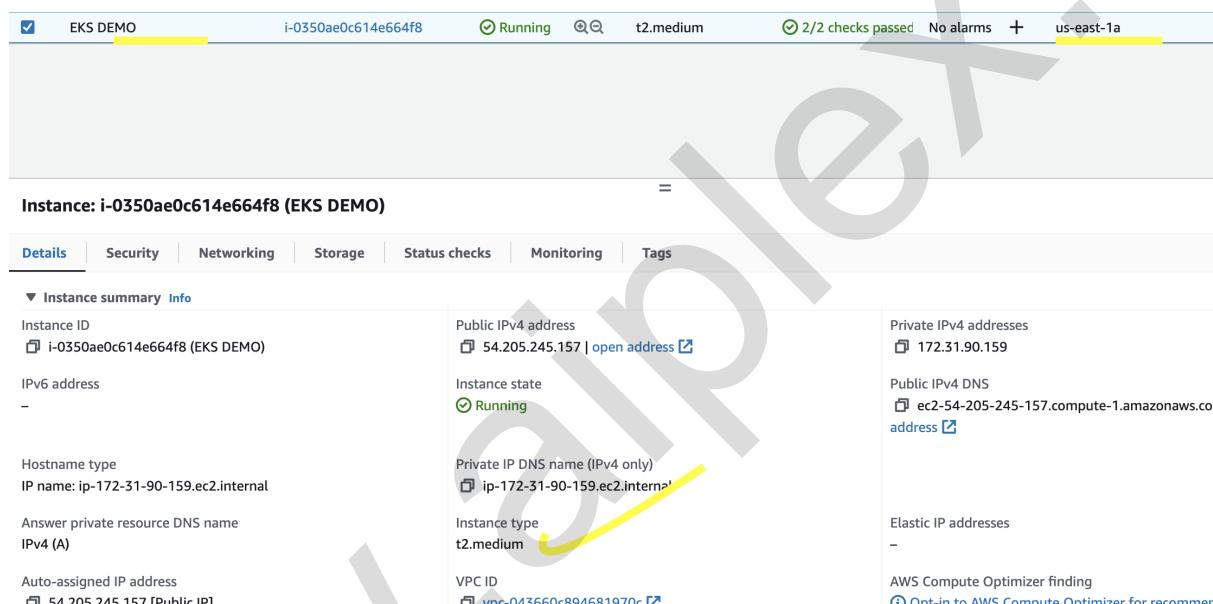
- 1) AWS LOGIN
 - 2) DOCKER HUB LOGIN
 - 3) GITHUB HUB LOGIN
- **** COST WILL OCCUR IN AWS ACCOUNT***

SESSION 4 – HANDSON

YOUTUBE – <https://www.youtube.com/praveensingampalli>

INSTAGRAM - <https://www.instagram.com/singam4devops/>

STEP 1 – CREATE A **T2.MEDIUM UBUNTU** EC2 INSTANCE IN AWS IN US-WEST-1 REGION



STEP 2 - Install JDK on AWS EC2 Instance

```
sudo apt-get update
sudo apt install openjdk-11-jre-headless -y
java --version
```

STEP3 - Install and Setup Jenkins

SESSION 4 – HANDSON

YOUTUBE – <https://www.youtube.com/praveensingampalli>

INSTAGRAM - <https://www.instagram.com/singam4devops/>

curl -fsSL

```
https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo tee /usr/share/keyrings/jenkins-keyring.asc > /dev/null
```

echo deb

```
[signed-by=/usr/share/keyrings/jenkins-keyring.asc]  
https://pkg.jenkins.io/debian-stable binary/ | sudo tee /etc/apt/sources.list.d/jenkins.list > /dev/null
```

sudo apt update

sudo apt install jenkins

sudo systemctl status jenkins

```
No containers need to be restarted.  
No user sessions are running outdated binaries.  
No VM guests are running outdated hypervisor (qemu) binaries on this host.  
● jenkins.service - Jenkins Continuous Integration Server  
  Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)  
  Active: active (running) since Fri 2023-04-14 03:46:07 UTC; 2s ago  
    Main PID: 5097 (java)  
      Tasks: 45 (limit: 4686)  
     Memory: 1.2G  
        CPU: 48.572s  
       CGroup: /system.slice/jenkins.service
```

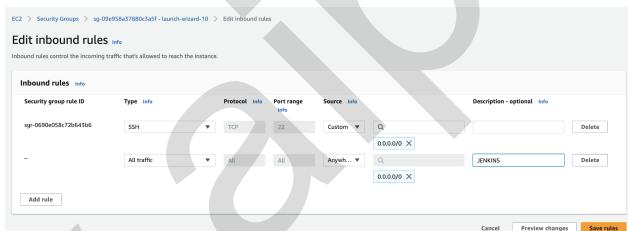
SESSION 4 – HANDSON

YOUTUBE – <https://www.youtube.com/praveensingampalli>

INSTAGRAM - <https://www.instagram.com/singam4devops/>

Step 4: Setup Jenkins

- Now go to AWS dashboard -> EC2 -> Instances(running) and click on Jenkins-EC2
- Copy Public IPv4 address.
- Change the SG to open for Jenkins.



- Alright now we know the public IP address of the EC2 machine, so now we can access Jenkins from the browser using the public IP address followed by port 8080.
- Copy the below key and paste it on JENKINS

SESSION 4 – HANDSON

YOUTUBE – <https://www.youtube.com/praveensingampalli>

INSTAGRAM - <https://www.instagram.com/singam4devops/>

```
cat: /var/lib/jenkins/secrets/initialAdminPassword: Permission denied
ubuntu@ip-172-31-90-159:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
4a7db9ed7fed46e7a11a305ade8a5d5b
ubuntu@ip-172-31-90-159:~$
```

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password is stored in the log (**not sure where to find it?**) and this file on the server:

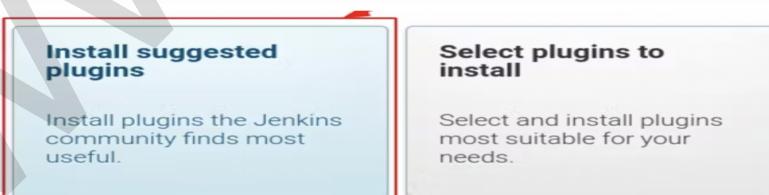
`/var/lib/jenkins/secrets/initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password

.....

- After completing the installation of the suggested plugin you need to set the **First Admin User** for Jenkins.



The image shows the 'Create First Admin User' form. It includes fields for 'Username' (admin), 'Password' (admin), 'Confirm password' (admin), 'Full name' (Sunkal), and 'E-mail address' (sunkal123@gmail.com). Below the form is a note: 'This user will have full administrative access to your Jenkins instance.'

SESSION 4 – HANDSON

YOUTUBE – <https://www.youtube.com/praveensingampalli>

INSTAGRAM - <https://www.instagram.com/singam4devops/>

STEP 5 - Update visudo and assign administration privileges to jenkins user

- Open the file /etc/sudoers in **vi** mode
sudo vi /etc/sudoers
- Add the following line at the end of the file

jenkins ALL=(ALL) NOPASSWD: ALL

-After adding the line save and quit the file.Now we can use Jenkins as root user and for that run the following command

sudo su - jenkins

SESSION 4 – HANDSON

YOUTUBE – <https://www.youtube.com/praveensingampalli>

INSTAGRAM - <https://www.instagram.com/singam4devops/>

```
# Members of the admin group may gain root privileges
%admin  ALL=(ALL)  ALL

# Allow members of group sudo to execute any command
%sudo    ALL=(ALL:ALL)  ALL

# See sudoers(5) for more information on "@include" directives

@includedir /etc/sudoers.d
-- INSERT --
jenkins  ALL=(ALL)  NOPASSWD:  ALL

@includedir /etc/sudoers.d
:wq!
```

STEP6 - Install Docker with user jenkins

```
sudo apt install docker.io
docker --version
docker ps
sudo usermod -aG docker jenkins
sudo reboot
```

STEP 7 - Install and Setup AWS and EKS CLI

```
sudo apt install awscli
curl
"https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
```

SESSION 4 – HANDSON

YOUTUBE – <https://www.youtube.com/praveensingampalli>

INSTAGRAM - <https://www.instagram.com/singam4devops/>

```
unzip awscliv2.zip  
sudo ./aws/install --update  
aws --version
```

EKSCTL INSTALLATION

```
curl --silent --location  
"https://github.com/weaveworks/eksctl/releases/latest/download/e  
ksctl_$(uname -s)_amd64.tar.gz" | tar xz -C /tmp  
  
sudo mv /tmp/eksctl /usr/local/bin  
  
eksctl version
```

STEP 8 - Configure the AWS CLI so that it can authenticate and communicate with the AWS environment.

COMMAND : aws configure

Once you execute the above command it will ask for the following information -

1. AWS Access Key ID [None]: YOUR ACCESS KEY
2. AWS Secret Access Key [None]: YOUR SECRET KEY

SESSION 4 – HANDSON

YOUTUBE – <https://www.youtube.com/praveensingampalli>

INSTAGRAM - <https://www.instagram.com/singam4devops/>

3. Default region name [None]: REGION

4. Default output format [None]:

You can click on the **Create New Access Key** and it will let you generate - AWS Access Key ID, AWS Secret Access Key.

```
jenkins@ip-172-31-90-159:~$  
jenkins@ip-172-31-90-159:~$  
jenkins@ip-172-31-90-159:~$ aws configure  
AWS Access Key ID [None]: [REDACTED]  
AWS Secret Access Key [None]: [REDACTED]  
Default region name [None]: us-east-1  
Default output format [None]:  
jenkins@ip-172-31-90-159:~$ [REDACTED]
```

STEP 9 - Install and Setup Kubectl

curl -LO

<https://storage.googleapis.com/kubernetes-release/release/v1.23.6/bin/linux/amd64/kubectl>

chmod +x ./kubectl

sudo mv ./kubectl /usr/local/bin/kubectl

SESSION 4 – HANDSON

YOUTUBE – <https://www.youtube.com/praveensingampalli>

INSTAGRAM - <https://www.instagram.com/singam4devops/>

kubectl get pods – WILL GIVE ERROR

STEP 10 - Creating an Amazon EKS cluster using eksctl

IT WILL TAKE 15-20 MINUTES TIME TO CREATE THE CLUSTER

Now in this step, we are going to create Amazon EKS cluster using eksctl

You need the following in order to run the eksctl command

- 1. Name of the cluster : --name first-eks-cluster**
- 2. Version of Kubernetes : --version 1.24**
- 3. Region : --region us-west-1**
- 4. Nodegroup name/worker nodes
: --nodegroup-name worker-nodes**
- 5. Node Type : --nodegroup-type t2.micro**
- 6. Number of nodes: --nodes 2**

SESSION 4 – HANDSON

YOUTUBE – <https://www.youtube.com/praveensingampalli>

INSTAGRAM - <https://www.instagram.com/singam4devops/>

```
eksctl create cluster --name first-eks-cluster --version  
1.24 --region us-west-1 --nodegroup-name  
worker-nodes --node-type t2.micro --nodes 2
```

The screenshot shows the AWS EKS service dashboard. On the left, there's a sidebar with 'Clusters' (New), 'Related services' (Amazon ECR, AWS Batch), and 'Documentation'. The main area is titled 'Clusters' and shows a table with one entry: 'first-eks-cluster' (Status: Active, Kubernetes version: 1.24). A yellow arrow points to the cluster name. Below this, another table shows 'Cluster: Nodes (2)' with two entries: 'ip-192-168-48-227.ec2.internal' and 'ip-192-168-5-98.ec2.internal', both of which are 't2.micro' instances in the 'worker-nodes' group and are 'Created 3 hours ago'.

STEP 11 - Add Docker and GitHub Credentials on Jenkins

- Setup Docker Hub Secret Text in Jenkins

You can set the docker credentials by going into -

SESSION 4 – HANDSON

YOUTUBE – <https://www.youtube.com/praveensingampalli>

INSTAGRAM - <https://www.instagram.com/singam4devops/>

Goto -> Jenkins -> Manage Jenkins -> Manage Credentials -> Stored scoped to jenkins -> global -> Add Credentials [GIVE YOUR DOCKER HUB CREDENTIALS]

Update credentials

Scope ?

Global (Jenkins, nodes, items, all child items, etc)

Username ?

praveensingam1994

Treat username as secret

Password ?

Concealed Change Password

ID ?

docker_cred

Description ?

- Setup GitHub Username and password into Jenkins

Now we add one more username and password for GitHub.

Goto -> Jenkins -> Manage Jenkins -> Manage Credentials -> Stored scoped to jenkins -> global -> Add Credentials

 GIT_HUB_CREDENTIALS	praveen1994dec/*****	Username with password	
 docker_cred	praveensingam1994/*****	Username with password	

SESSION 4 – HANDSON

YOUTUBE – <https://www.youtube.com/praveensingampalli>

INSTAGRAM - <https://www.instagram.com/singam4devops/>

STEP 11.1 -> ADD THE EKS IAM ROLE TO EC2

GO TO IAM -> CREATE ROLE

The screenshot shows the 'Create role' wizard in the AWS IAM console. The steps are:

- Step 1: Select trusted entity
- Step 2: Add permissions (highlighted)
- Step 3: Name, review, and create

In the 'Add permissions' step, the 'Permissions policies' section shows the 'AdministratorAccess' policy selected. A yellow checkmark is placed over the checkbox next to 'AdministratorAccess'. The table below lists three policies:

Policy name	Type	Description
AdministratorAccess	AWS managed	Provides full access to AWS services
PowerUserAccess	AWS managed	Provides full access to AWS services
ReadOnlyAccess	AWS managed	Provides read-only access to AWS services

STEP 12 – ADD MAVEN IN GLOBAL TOOL CONFIGURATION



STEP 13 – ADD JENKINS SHARED LIBRARY

SESSION 4 – HANDSON

YOUTUBE – <https://www.youtube.com/praveensingampalli>

INSTAGRAM - <https://www.instagram.com/singam4devops/>

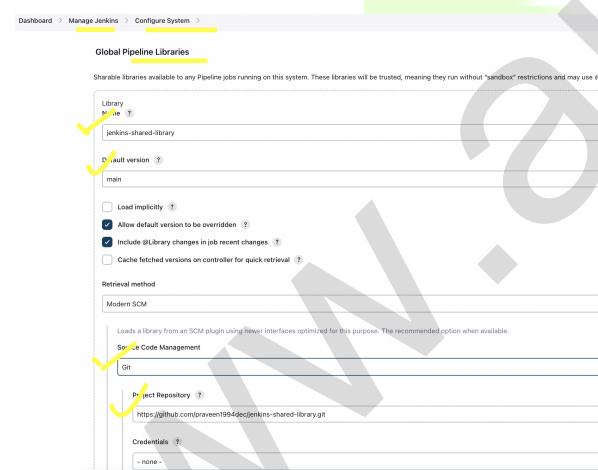
Go to Manage Jenkins -> Configure System -> Global Pipeline Libraries ->

Give Library name – jenkins-shared-library

Default Version - main

Project Repository -

<https://github.com/DEVOPS-WITH-WEB-DEV/jenkins-shared-library1.git>



STEP 14 - Build, deploy and test CI/CD pipeline

Create new Pipeline: Go to Jenkins Dashboard or Jenkins home page click on New Item

SESSION 4 – HANDSON

YOUTUBE – <https://www.youtube.com/praveensingampalli>

INSTAGRAM - <https://www.instagram.com/singam4devops/>

Pipeline Name: Now enter Jenkins pipeline name and select Pipeline

Add pipeline script:

Click on Configure -> Select Pipeline ->
<https://github.com/praveen1994dec/spring-cloud-kubernetes/blob/main/kubernetes-configmap-reload/Jenkinsfile> -> Copy that Jenkinsfile -> Paste it in Pipeline Script in Jenkins

STEP 15 – ANSIBLE PYTHON SETUP

sudo apt update

sudo apt install software-properties-common

sudo add-apt-repository --yes --update
ppa:ansible/ansible

sudo apt install ansible

sudo apt install python3

sudo apt install python3-pip

pip3 install Kubernetes

SESSION 4 – HANDSON

YOUTUBE – <https://www.youtube.com/praveensingampalli>

INSTAGRAM - <https://www.instagram.com/singam4devops/>

STEP 16 - SELECT BUILD WITH PARAMETERS [IF FAILS AGAIN START THE PIPELINE]

Action – create

ImageName - kubernetes-configmap-reload

ImageTag – v1

AppName - kubernetes-configmap-reload

Docker_repo – Give your Docker Hub account ID

DOCKER LOGIN

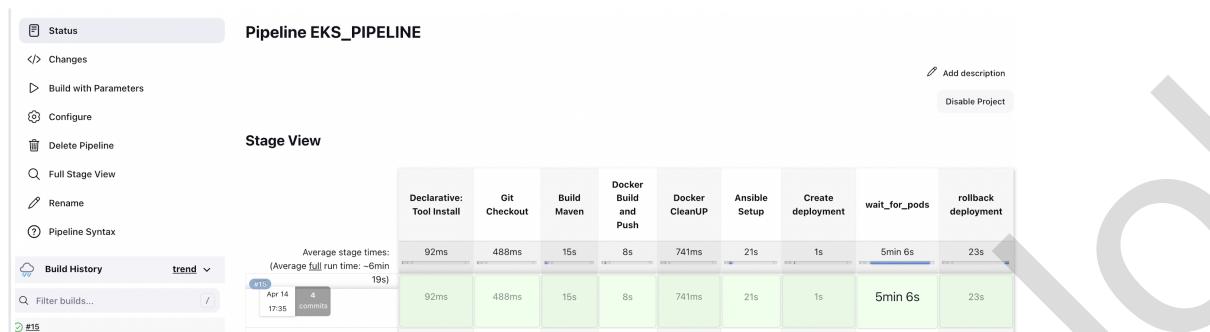
The screenshot shows the AWS Lambda Pipeline console. On the left, there's a sidebar with navigation links like 'ashboard', 'Status', 'Changes', 'Build with Parameters' (which is currently selected), 'Configure', 'Delete Pipeline', 'Full Stage View', 'Rename', and 'Pipeline Syntax'. Below this is a 'Build History' section with a dropdown menu set to 'trend' and a 'Filter builds...' input field. A list of build logs is shown, with the most recent one being '#15 Apr 14, 2023, 12:05 PM'. To the right of the sidebar, the main area is titled 'Pipeline EKS_PIPELINE'. It displays a message: 'This build requires parameters: action'. Below this, there are four input fields: 'action' (set to 'create'), 'ImageName' (empty), 'ImageTag' (empty), and 'AppName' (empty). At the bottom of the pipeline configuration area, there's a 'docke_repo' input field. The overall interface has a light gray background with some green highlights around the 'Build with Parameters' section.

STEP 17 – PIPELINE WILL PASS

SESSION 4 – HANDSON

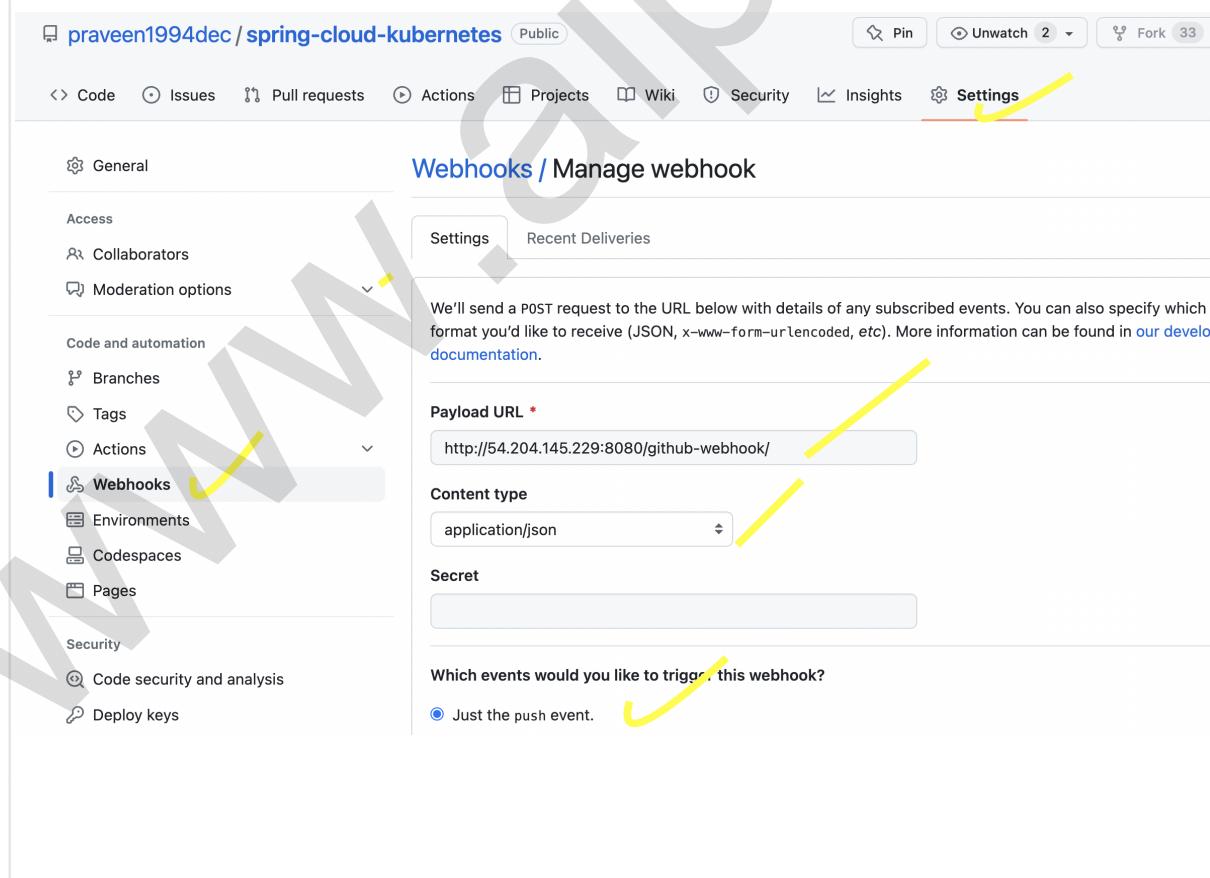
YOUTUBE – <https://www.youtube.com/praveensingampalli>

INSTAGRAM - <https://www.instagram.com/singam4devops/>



STEP 18 – ADD WEBHOOK

IN GITHUB PROJECT -> SETTINGS -> WEBHOOK -> URL
-> http://<ec2_ip>:8080/github-webhook/ -> ONLY
FOR PUSH EVENTS

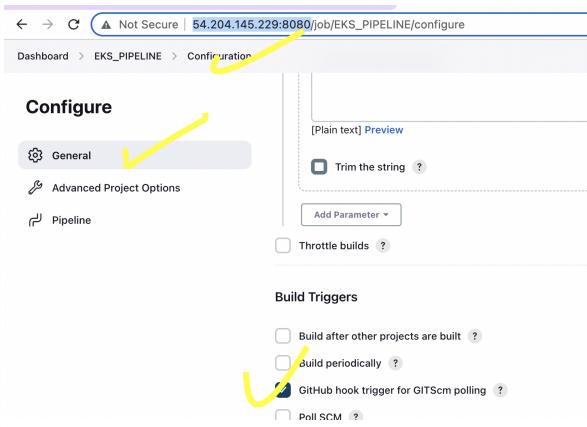


SESSION 4 – HANDSON

YOUTUBE – <https://www.youtube.com/praveensingampalli>

INSTAGRAM - <https://www.instagram.com/singam4devops/>

**IN JENKINS -> GO TO GENERAL TAB -> BUILD TRIGGERS
-> ADD GITHUB WEB HOOK TRIGGER**



STEP 19 – DELETE EKS CLUSTER [PLEASE NOTE FROM STEP5 TO STEP19 ALL STEPS HAVE TO BE DONE AS JENKINS USER IN UBUNTU SYSTEM]

eksctl delete cluster --name first-eks-cluster

If you have liked the project please do share the channel link with your friends 😊

<https://www.youtube.com/praveensingampalli>