**Problem: - To create and configure a simple CI/CD pipeline using Jenkins or any other open-source automation tool.**

**Solution: -** I have solve the problem in multiple steps on the AWS cloud.

Step1: - First I launch the Linux instance for setup the Jenkins & Docker.

Step2: In this step, I created a EKS cluster for Kubernetes.

Steps for create the EKS cluster: -

* First, I created the IAM role for EKS cluster. In the IAM, I had to give the policy permission ([AmazonEKSClusterPolicy](https://us-east-1.console.aws.amazon.com/iam/home" \l "/policies/arn:aws:iam::aws:policy/AmazonEKSClusterPolicy" \t "_blank)).
* After that I created a EKS Cluster. In this cluster, I had assigned the IAM role which I have created.
* Then I created again a IAM role for NodeGroup and give the three-policy permission. ([AmazonEKSWorkerNodePolicy](https://us-east-1.console.aws.amazon.com/iam/home" \l "/policies/arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy" \t "_blank), [AmazonEC2ContainerRegistryReadOnly](https://us-east-1.console.aws.amazon.com/iam/home#/policies/arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly), [AmazonEKS\_CNI\_Policy](https://us-east-1.console.aws.amazon.com/iam/home#/policies/arn:aws:iam::aws:policy/AmazonEKS_CNI_Policy)).
* Then I created a NodeGroup in the EKS cluster and assign the IAM role.

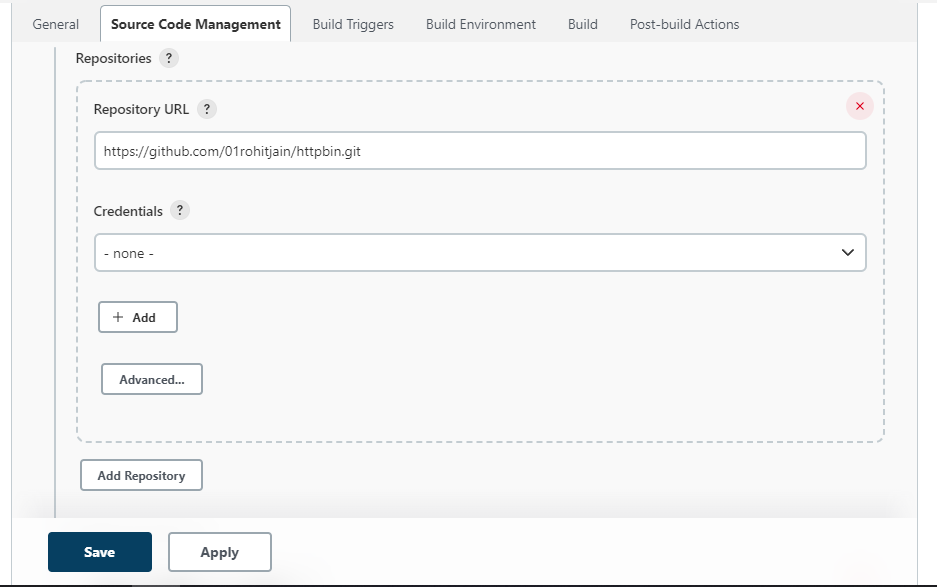
Step3: - In the next step I connected the Cluster to my launched instance.

Step4: - In the next step I have installed the kubectl in the instance.

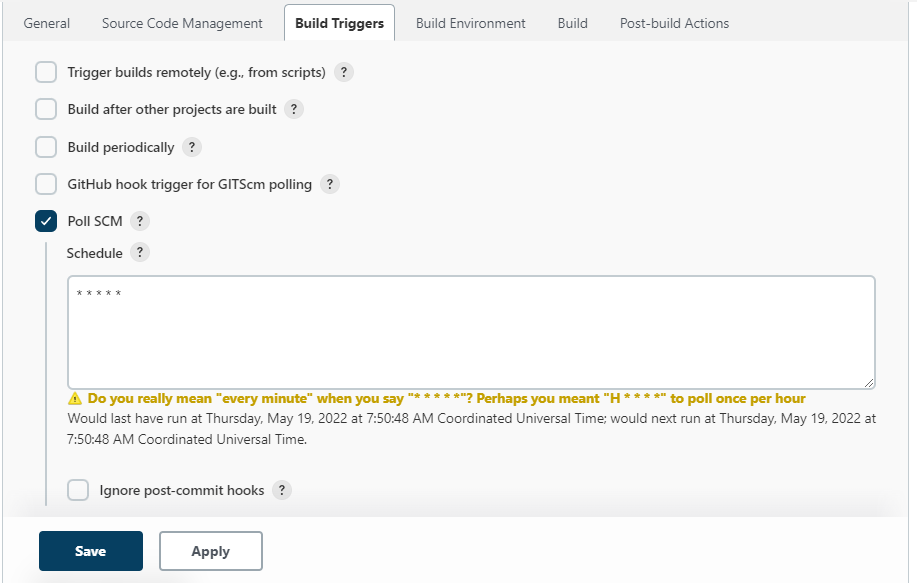
Step5: - Install the Docker and GitHub plugins in the Jenkins. And set the credential of the docker.

Step6: - In this step download the GitHub Repository from the given URL <https://github.com/postmanlabs/httpbin> and push on my GitHub account <https://github.com/01rohitjain/httpbin> then do some changes in the Docker file.

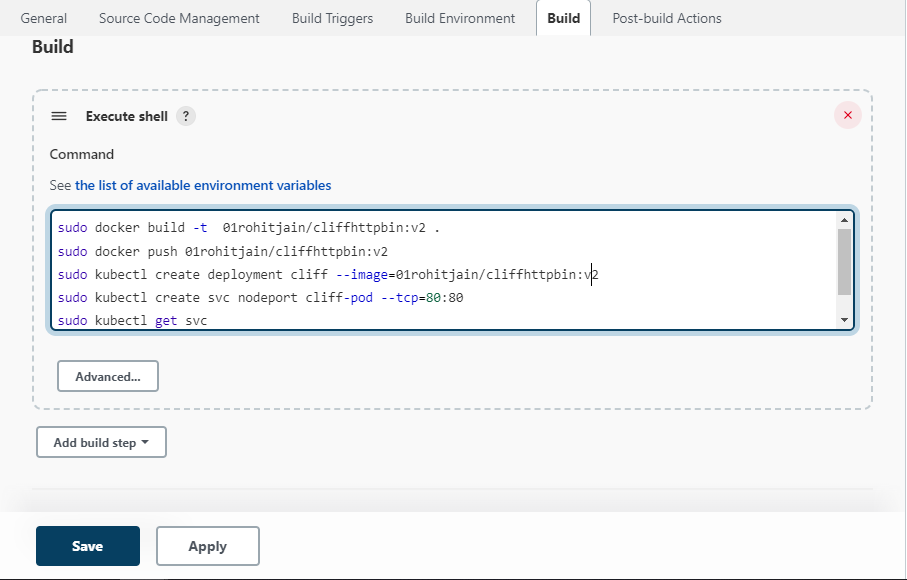
Step7: Provide the GitHub URL to the Jenkins job for get the data.



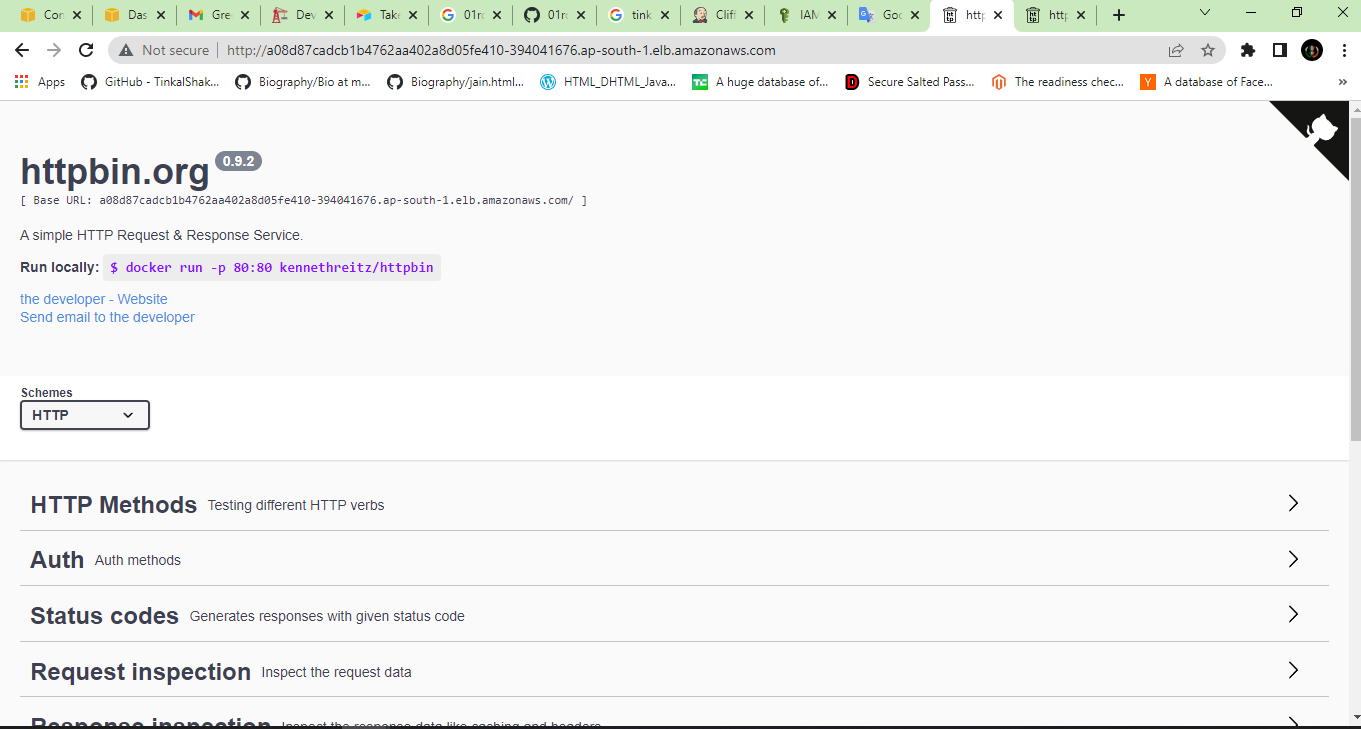
Step8: - In the next step, assign the Poll SCM that is used for getting the data for real time.

****

Step9: - In this step, I assigned the some docker, Kubernetes command in the execute shell and save the job and build.



**Output: -**



**Components used: -**

* AWS (EC2, EKS, IAM)
* Jenkins
* Docker
* Kubernetes
* GitHub

**Jenkins URL: -**

<http://43.204.107.100:8080/job/Cliff_CiCd_job>

**URI of the application host: -**

[**http://a08d87cadcb1b4762aa402a8d05fe410-394041676.ap-south-1.elb.amazonaws.com/**](http://a08d87cadcb1b4762aa402a8d05fe410-394041676.ap-south-1.elb.amazonaws.com/)

**Conclusion:** - Finally I created a CI/CD pipeline using Jenkins in this pipeline if developer commit any changes on the GitHub Repo than webapp automatically update.