Amazon Elastic File System

Scalable,elastic,cloud-native NFS File system for $0.08/GB-Month

Amazon Elastic File System (Amazon EFS) provides a simple, scalable, fully managed elastic NFS file system for use with AWS Cloud services and on-premises resources. It is built to scale on demand to petabytes without disrupting applications, growing and shrinking automatically as you add and remove files, eliminating the need to provision and manage capacity to accommodate growth.

Amazon EFS offers two storage classes: the Standard storage class, and the [Infrequent Access storage class](https://aws.amazon.com/efs/features/infrequent-access/) (EFS IA). EFS IA provides price/performance that's cost-optimized for files not accessed every day. By simply enabling EFS Lifecycle Management on your file system, files not accessed according to the lifecycle policy you choose will be automatically and transparently moved into EFS IA. The EFS IA storage class costs only $0.025/GB-month\*.

While workload patterns vary, customers typically find that 80% of files are infrequently accessed (and suitable for EFS IA), and 20% are actively used (suitable for EFS Standard), resulting in an effective storage cost as low as $0.08/GB-month\*. Amazon EFS transparently serves files from both storage classes in a common file system namespace.

Amazon EFS is designed to provide massively parallel shared access to thousands of Amazon EC2 instances, enabling your applications to achieve high levels of aggregate throughput and IOPS with consistent low latencies.

Amazon EFS is well suited to support a broad spectrum of use cases from home directories to business-critical applications. Customers can use EFS to lift-and-shift existing enterprise applications to the AWS Cloud. Other use cases include: big data analytics, web serving and content management, application development and testing, media and entertainment workflows, database backups, and container storage.

Amazon EFS is a regional service storing data within and across multiple Availability Zones (AZs) for high availability and durability. Amazon EC2 instances can access your file system across AZs, regions, and VPCs, while on-premises servers can access using AWS Direct Connect or AWS VPN.

\*pricing in US East (N. Virginia) region, assumes 80% of your storage in EFS IA

Amazon EFS overview (2:13)

[What is Cloud File Storage?](https://aws.amazon.com/what-is-cloud-file-storage/)

Learn more about cloud file storage and how it compares to other cloud storage solutions.

[When to Choose Amazon EFS](https://aws.amazon.com/efs/when-to-choose-efs/)

Cloud storage options provide a simple and scalable way to store, access, and share data over the Internet. Here, we compare Amazon EFS to other AWS Cloud storage offerings, helping you understand when to choose Amazon EFS.

Benefits

POSIX-compliant shared file storage

Amazon EFS provides secure access for thousands of connections for Amazon EC2 instances and on-premises servers simultaneously using a traditional file permissions model, file locking capabilities, and hierarchical directory structure via the NFSv4 protocol. Amazon EC2 instances can access your file system across AZs, regions, and VPCs, while on-premises servers can access using AWS Direct Connect or AWS VPN.

Scalable performance

Amazon EFS is designed to provide the throughput, IOPS, and low latency needed for Linux workloads. Throughput and IOPS scale as a file system grows and can burst to higher throughput levels for short periods of time to support the unpredictable performance needs of file workloads. For the most demanding workloads, Amazon EFS can support performance over 10 GB/sec and more than 500,000 IOPS.

Dynamic elasticity

Amazon EFS automatically and instantly scales your file system storage capacity up or down as you add or remove files without disrupting your applications, dynamically providing the storage capacity you need as you need it. You simply create your file system and start adding files with no need to provision storage in advance.

Fully managed

Amazon EFS is a fully managed service providing shared file system storage for Linux workloads. It provides a simple interface allowing you to create and configure file systems quickly and manages the file storage infrastructure for you, removing the complexity of deploying, patching, and maintaining the underpinnings of a file system.

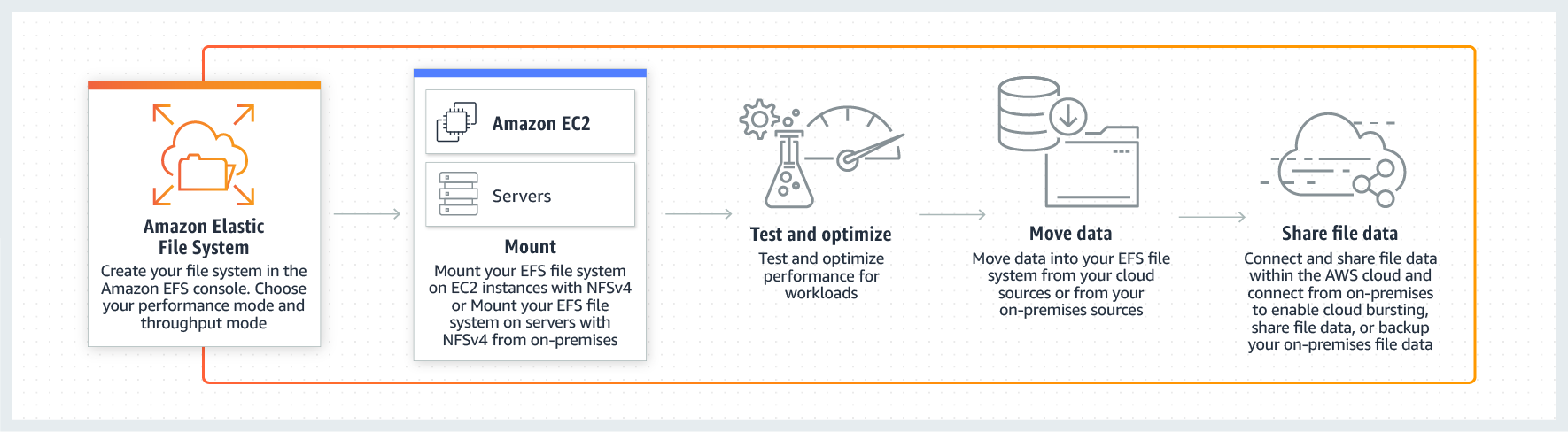
Cost-effective

With Amazon EFS storage, you pay only for what you use. There is no need to provision storage in advance and there are no minimum commitments or up-front fees. With EFS Lifecycle Management, you can automatically move files accessed less frequently to a cost-optimized storage class, reducing file storage costs by up to 92%. You can also use AWS Budgets to monitor your file system costs.

Security and compliance

Amazon EFS allows you to securely access your files using your existing security infrastructure. Control access to your Amazon EFS file systems with POSIX permissions, [Amazon VPC](https://aws.amazon.com/vpc/), and [AWS IAM](https://aws.amazon.com/iam/). Secure your data by encrypting your data at rest and in transit. Amazon EFS also meets many eligibility and compliance requirements to help you meet your regulatory needs. [Click here](https://aws.amazon.com/compliance/services-in-scope/) for a list of compliance programs in scope for Amazon EFS.

How it works



Use cases

Containers and serverless persistent file storage

Amazon EFS enables customers to persist data and state from their containers and serverless functions, providing fully managed, elastic, highly-available, scalable, and high-performance, cloud-native shared file systems. These same attributes are shared by Amazon Elastic Container Service (Amazon ECS), Amazon Kubernetes Service (Amazon EKS), AWS Fargate, and AWS Lambda, so developers don’t need to design for these traits, the services are simply ready for modern application development with data persistence. Amazon EFS allows data to be persisted separately from compute, and enables applications to have cross-AZ availability and durability. Amazon EFS provides a shared persistence layer that allows stateful applications to elastically scale up and down, such as for DevOps, web serving, web content systems, media processing, machine learning, analytics, search index, and stateful microservices applications.

Move to managed file systems

Amazon EFS provides the scalability, elasticity, availability, and durability to be the file store for enterprise applications and for applications delivered as a service. Its standard file system interface, file system permissions, and directory hierarchy make it easy to migrate enterprise applications from on-premises to the AWS cloud, and to build new ones. Move your business critical, Linux-based applications to managed file systems with Amazon EFS, while lowering your total cost of ownership (TCO).

Analytics & machine learning

Amazon EFS provides the ease of use, scale, performance, and consistency needed for machine learning and big data analytics workloads. Data scientists can use EFS to create personalized environments, with home directories storing notebook files, training data, and model artifacts. [Amazon SageMaker](https://aws.amazon.com/sagemaker/) integrates with EFS for training jobs, allowing data scientists to iterate quickly.

Web serving & content management

Amazon EFS provides a durable, high throughput file system for content management systems and web serving applications that store and serve information for a range of applications like websites, online publications, and archives. Since Amazon EFS adheres to the expected file system directory structure, file naming conventions, and permissions that web developers are accustomed to, it can easily integrate with web applications.

Application testing & development

Amazon EFS provides your development environments a common storage repository that gives you the ability to share code and other files in a secure and organized way. You can provision, duplicate, scale, or archive your test, development, and production environments with a few clicks, enabling your organization to be more agile and responsive to customer needs. Amazon EFS delivers a scalable and highly available solution that is ideal for testing and development workloads.

Media & entertainment

Media workflows like video editing, studio production, broadcast processing, sound design, and rendering often depend on shared storage to manipulate large files. Amazon EFS provides a strong data consistency model with high throughput and shared file access which can cut the time it takes to perform these jobs and consolidate multiple local file repositories into a single location for all users.

Database backups

Amazon EFS presents a standard file system that can be easily mounted with NFSv4 from database servers. This provides an ideal platform to create portable database backups using native application tools or enterprise backup applications. Many businesses want to take advantage of the flexibility of storing database backups in the cloud either for temporary protection during updates or for development and test.

Application of EFS storage :

Perform using EFS instead of EBS service on the AWS as,

Create/launch Application using Terraform

1. Create Security group which allow the port 80.

2. Launch EC2 instance.

3. In this Ec2 instance use the existing key or provided key and security group which we have created in step 1.

4. Launch one Volume using the EFS service and attach it in your vpc, then mount that volume into /var/www/html

5. Developer have uploded the code into github repo also the repo has some images.

6. Copy the github repo code into /var/www/html

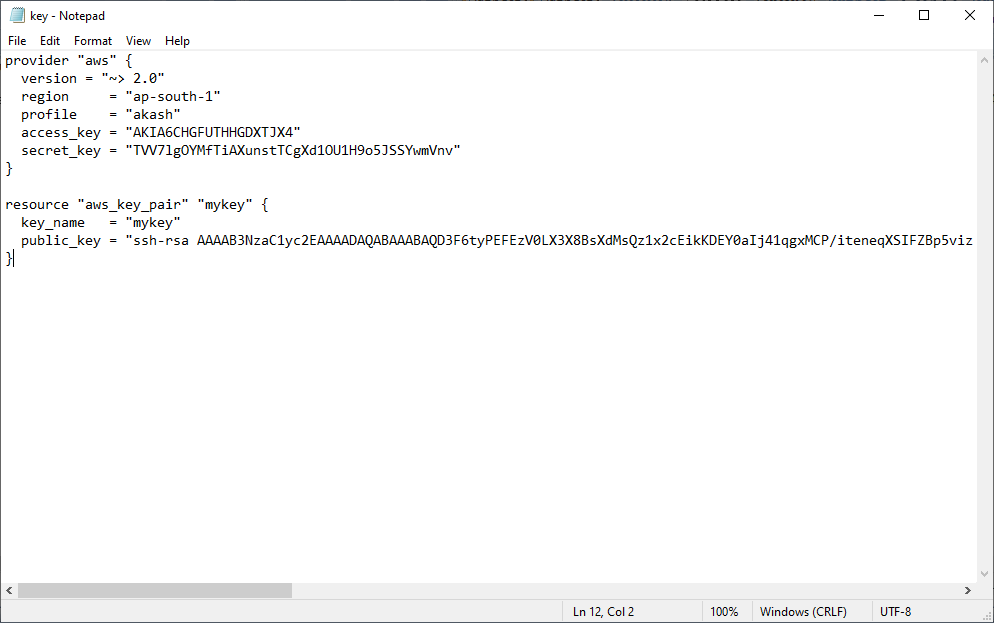
7. Create S3 bucket, and copy/deploy the images from github repo into the s3 bucket and change the permission to public readable.

8 Create a Cloudfront using s3 bucket(which contains images) and use the Cloudfront URL to update in code in /var/www/html

Optional

1. Those who are familiar with jenkins or are in devops AL have to integrate jenkins in this task wherever you feel can be integrate

Create key-pair for launch the EC2 instances: A **key pair**, consisting of a private **key** and a public **key**, is a set of security credentials that you **use** to prove your identity when connecting to an instance. Amazon **EC2** stores the public **key**, and you store the private **key**. You **use** the private **key**, instead of a password, to securely access your instances.

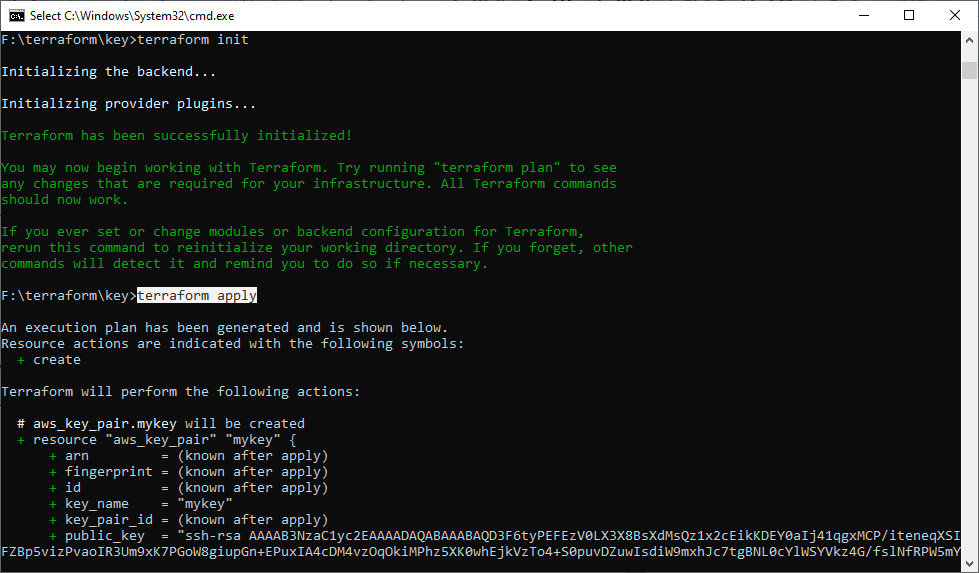


First run terraform init cmd:

F:\terraform\key>terraform init

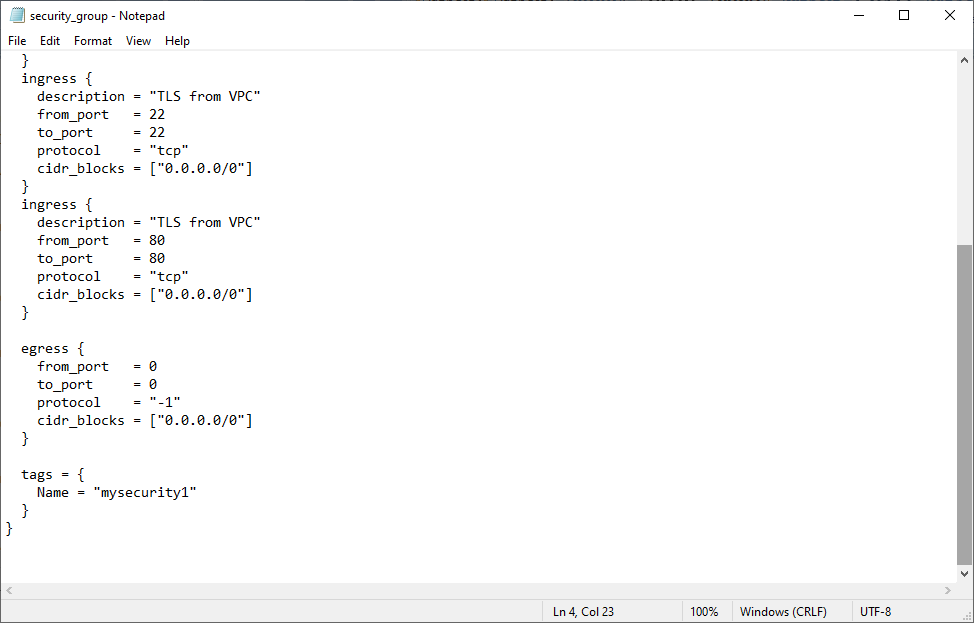
This cmd use for initializing the provider plugins.

The use the below highlight cmd for creating the resources.



Create security group : A network **security group** contains **security** rules that allow or deny inbound network traffic to, or outbound network traffic from, several types of Azure resources. For each rule, you can specify source and destination, port, and protocol.

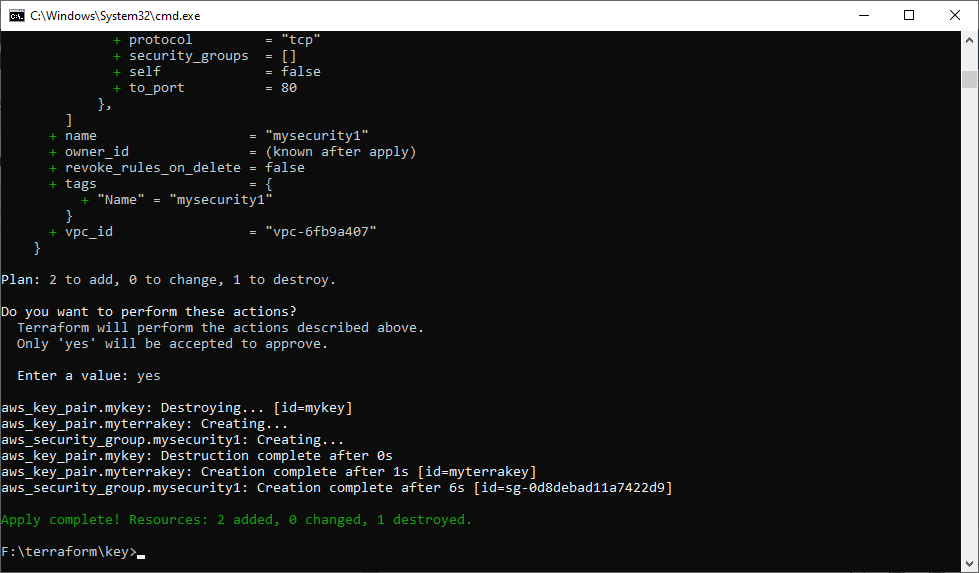




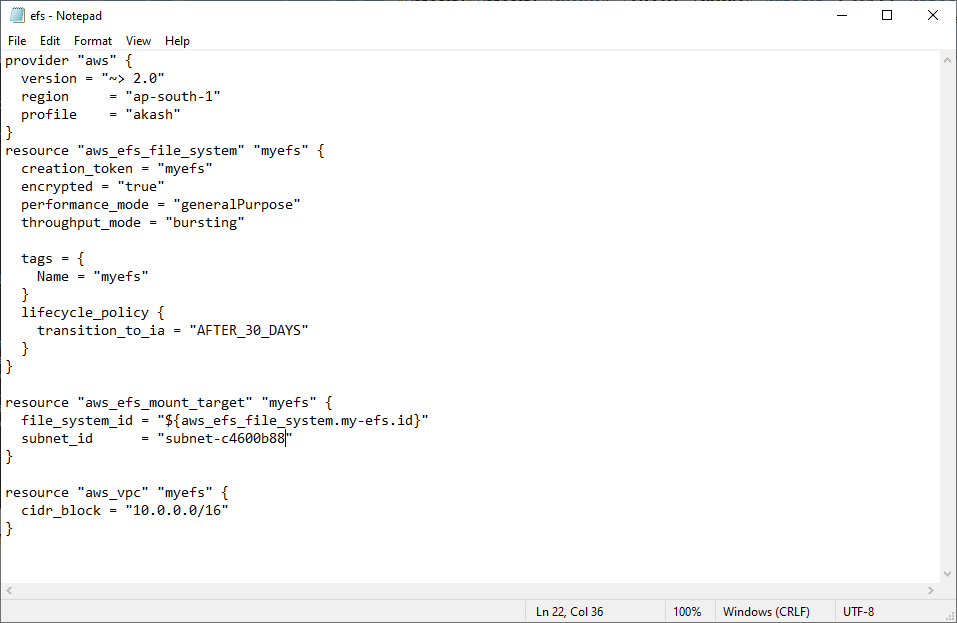
Use same cmd to create security-group resources:

F:\terraform\security\_group>terraform init

F:\terraform\security\_group>terraform apply



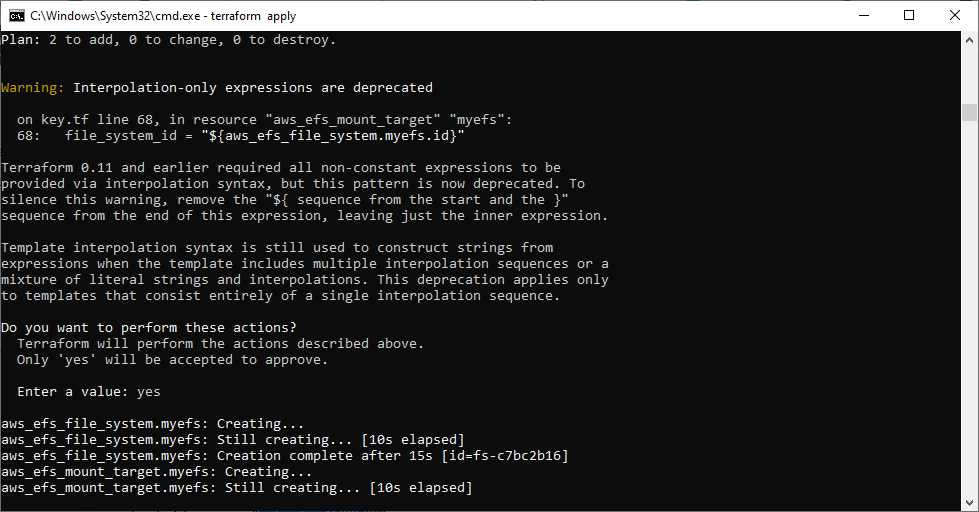
Create efs storage and mount it :

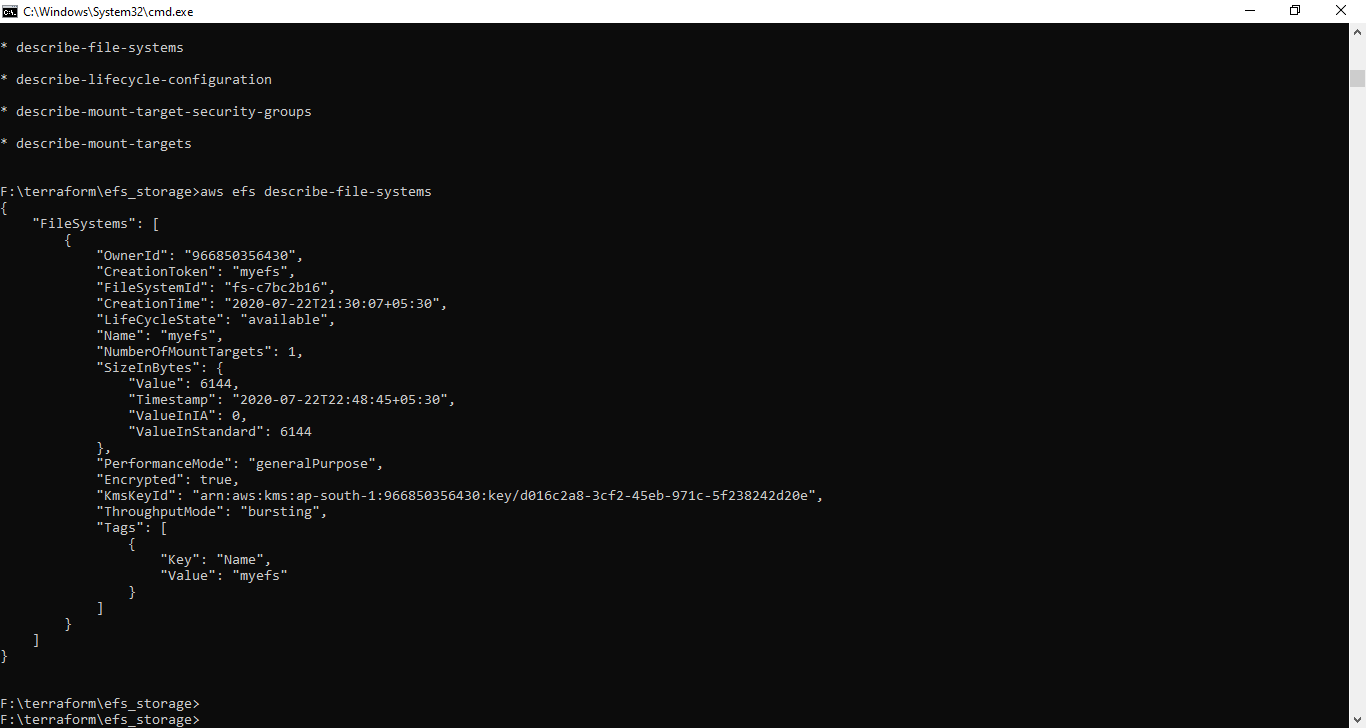


Run both cmds sequentially:

F:\terraform\efs\_storage>terraform init

F:\terraform\efs\_storage>terraform apply

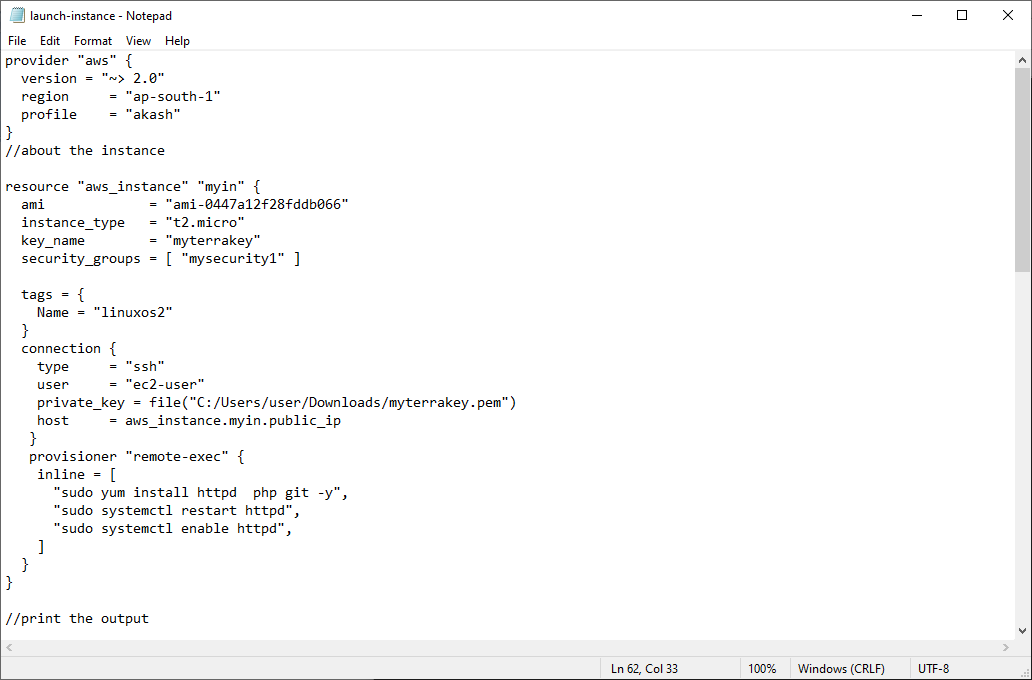


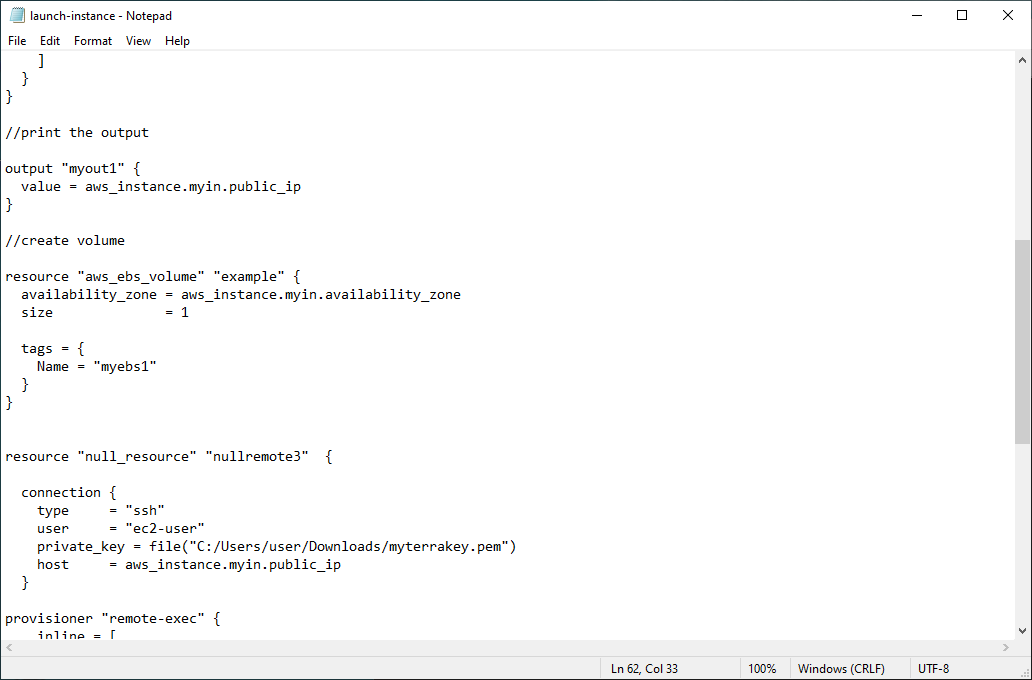


Launch the ec2 instance using above key & security group .and mount efs storage with it .you don’t need to do anything this code do everything for you.you need you run only this cmd .

F:\terraform\key>terraform init

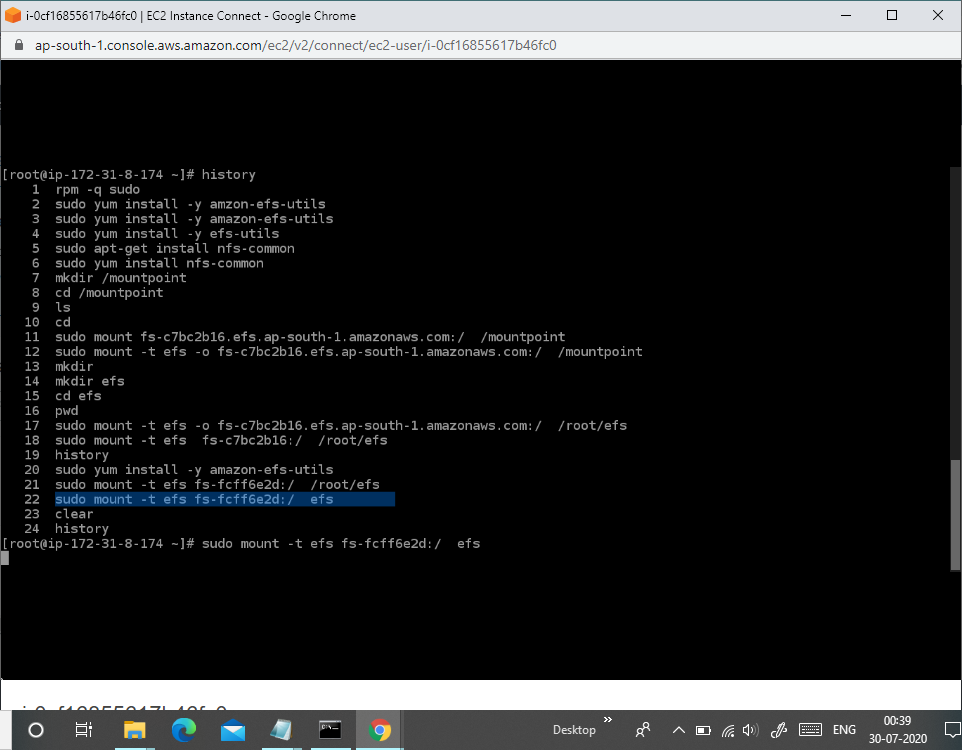
F:\terraform\key>terraform apply





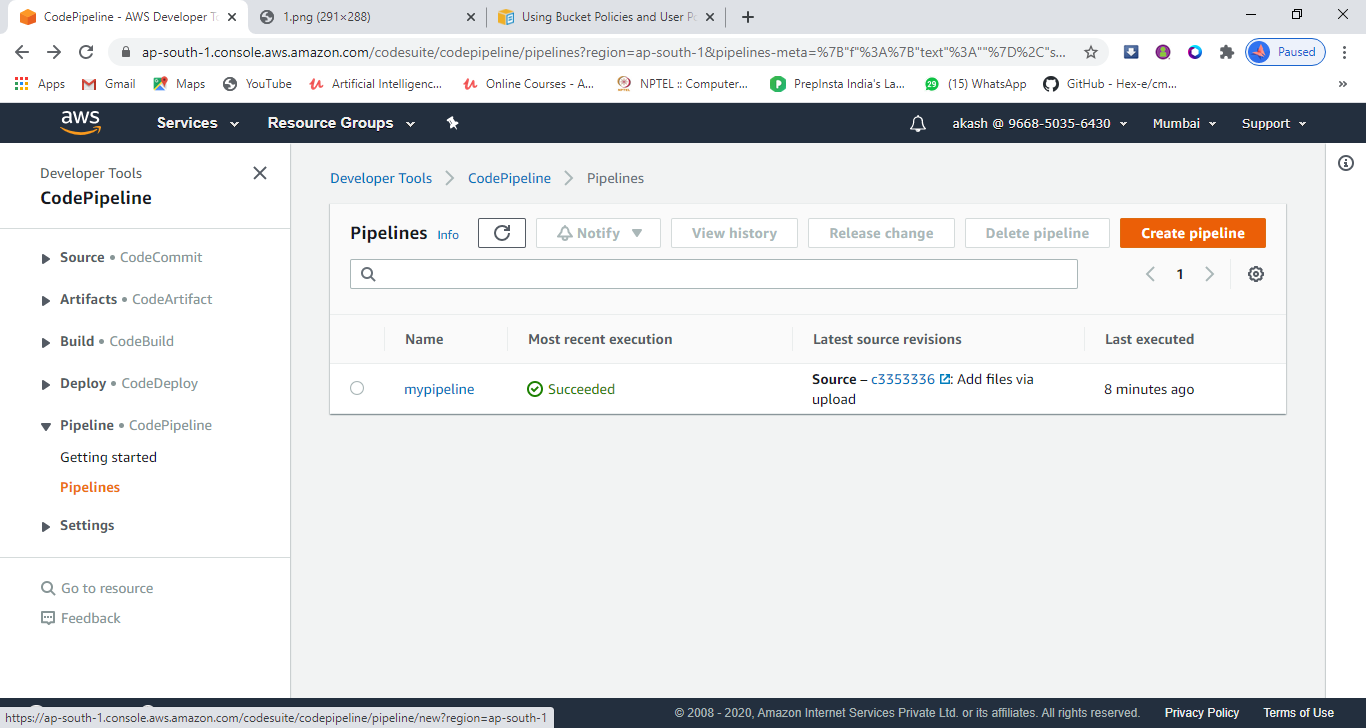


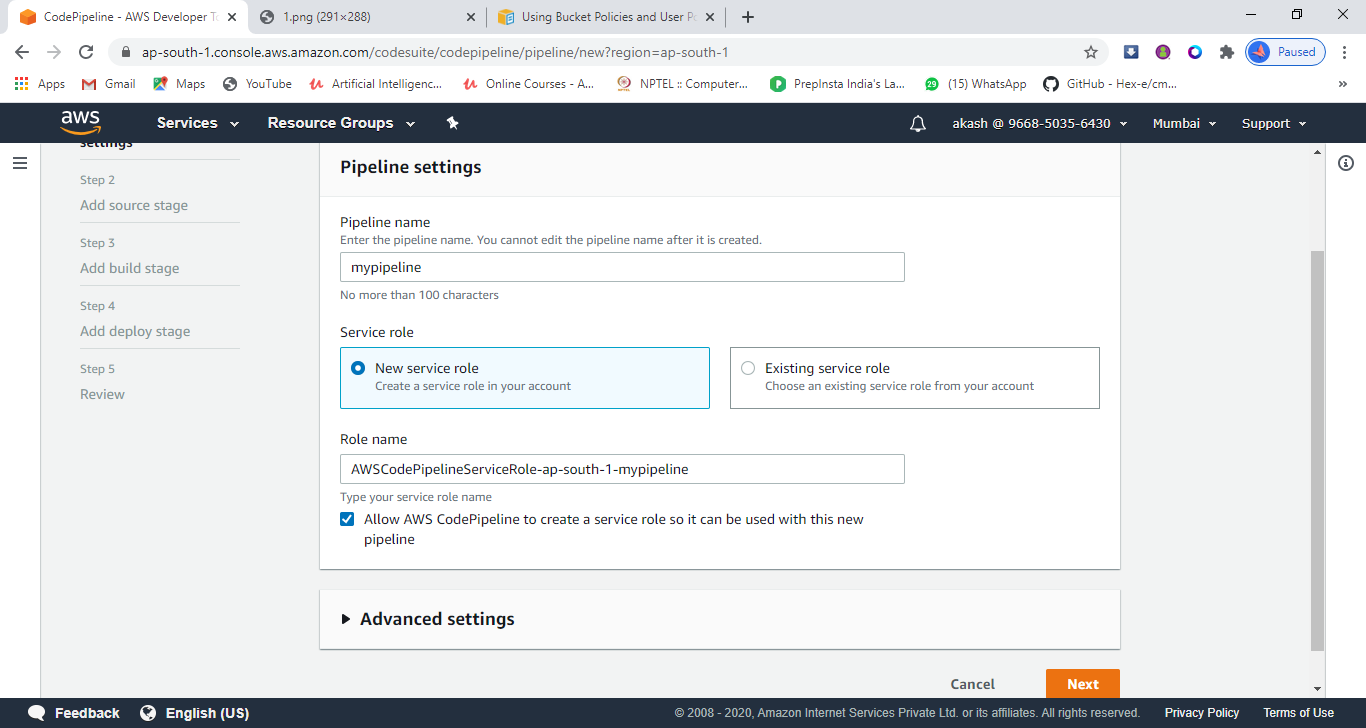
This is the basic cmd for do this thing manually.

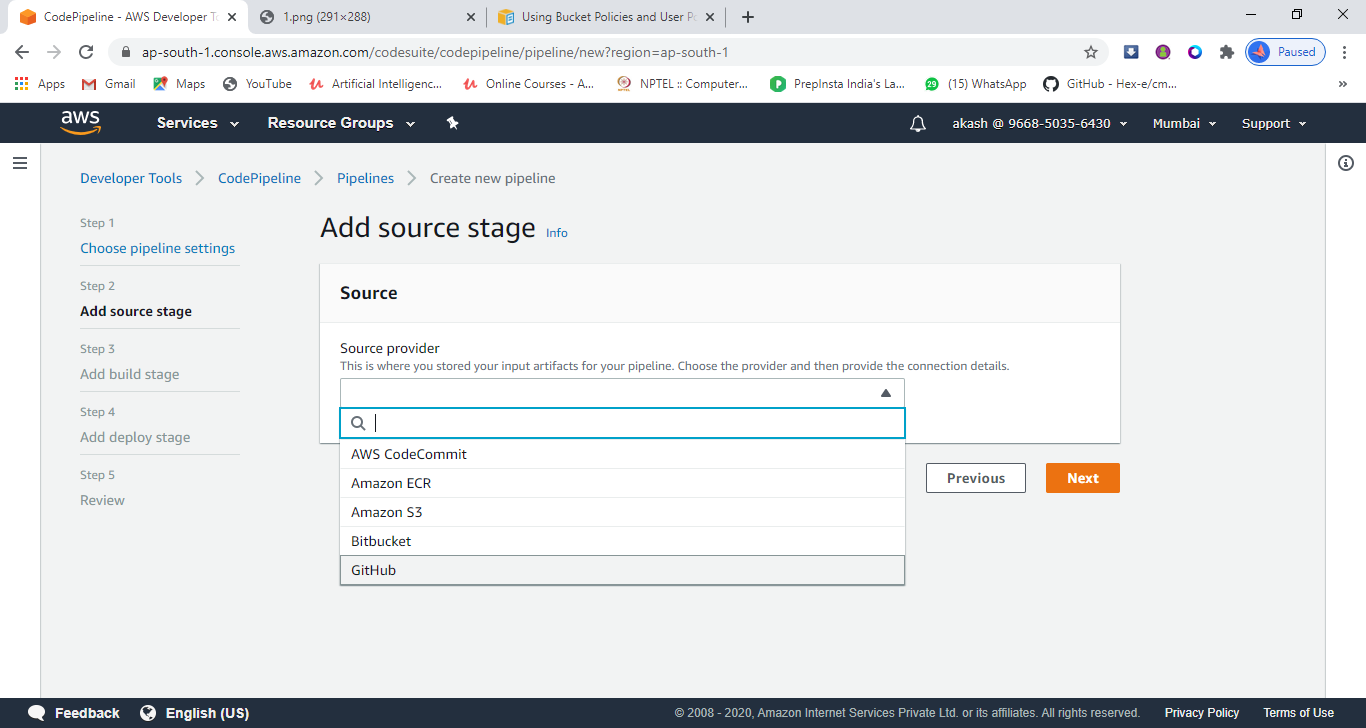


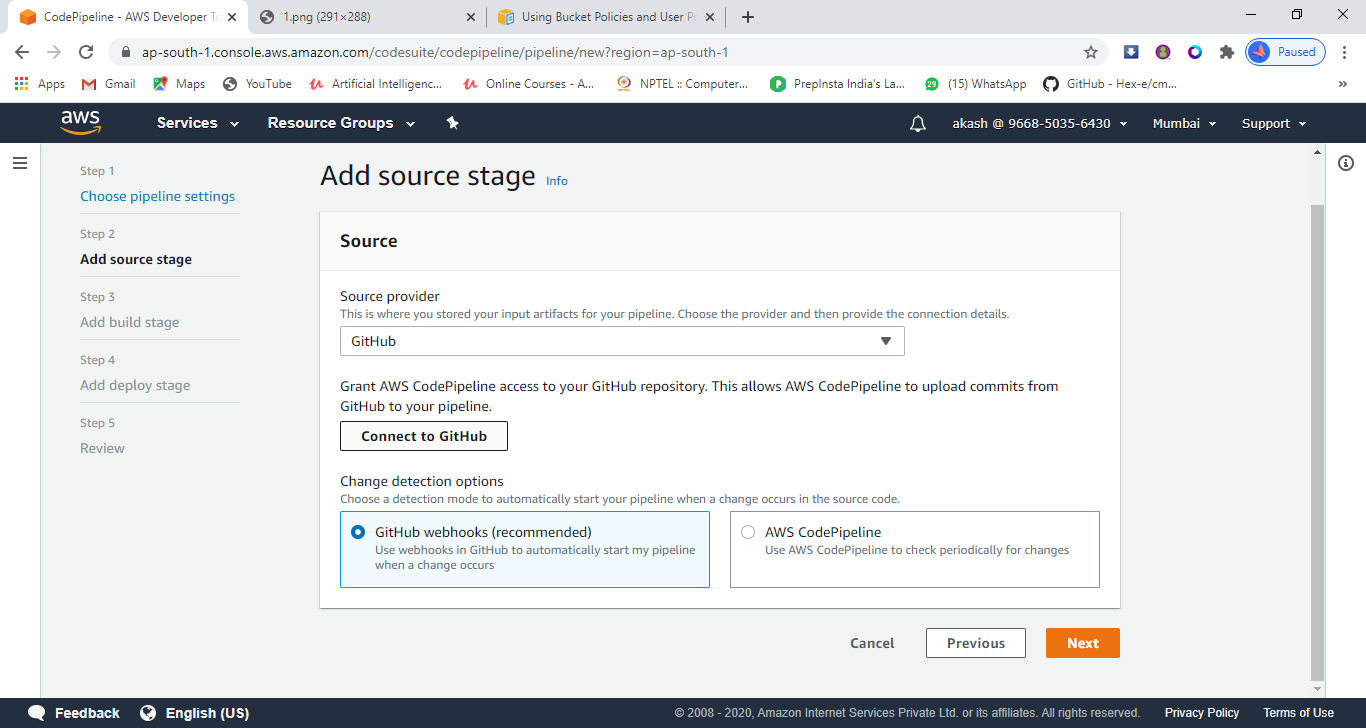
7. Create S3 bucket, and copy/deploy the images from github repo into the s3 bucket and change the permission to public readable.

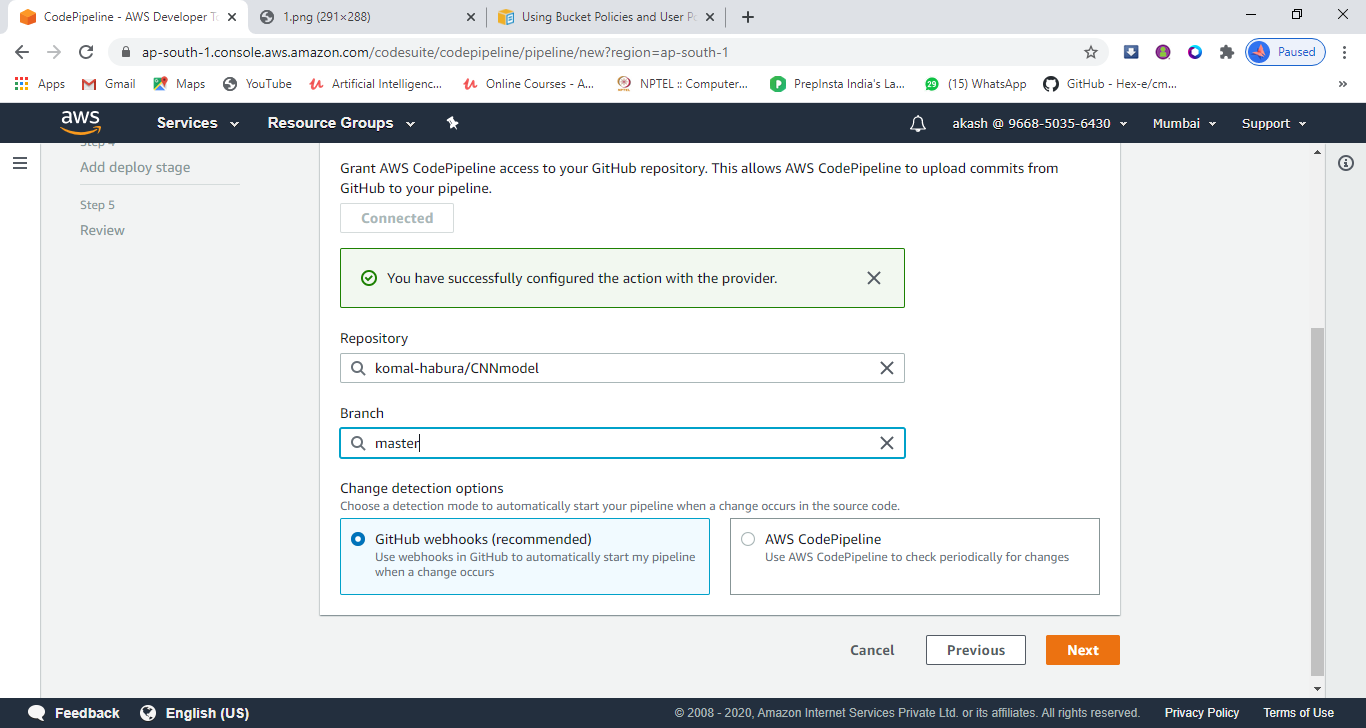
Aws provide one services is code pipeline .create pipeline to cop/deploy images from github repo into the s3 bucket

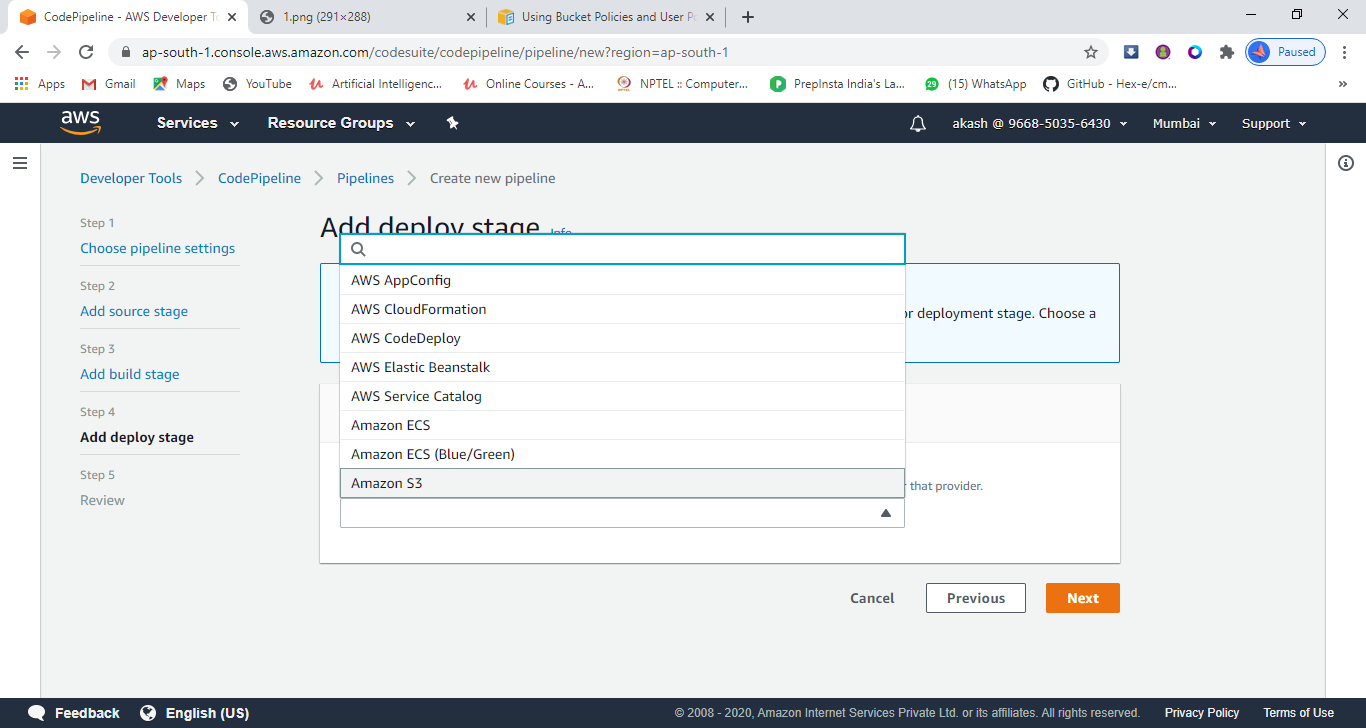






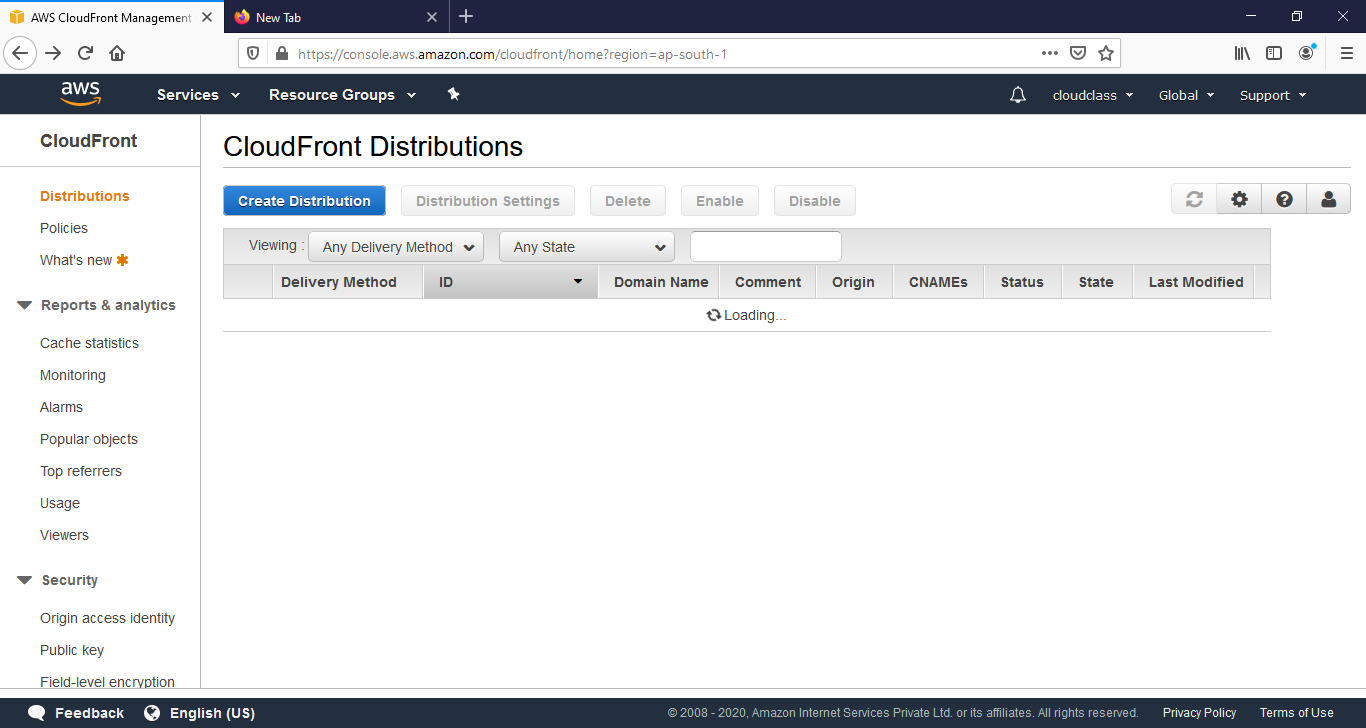


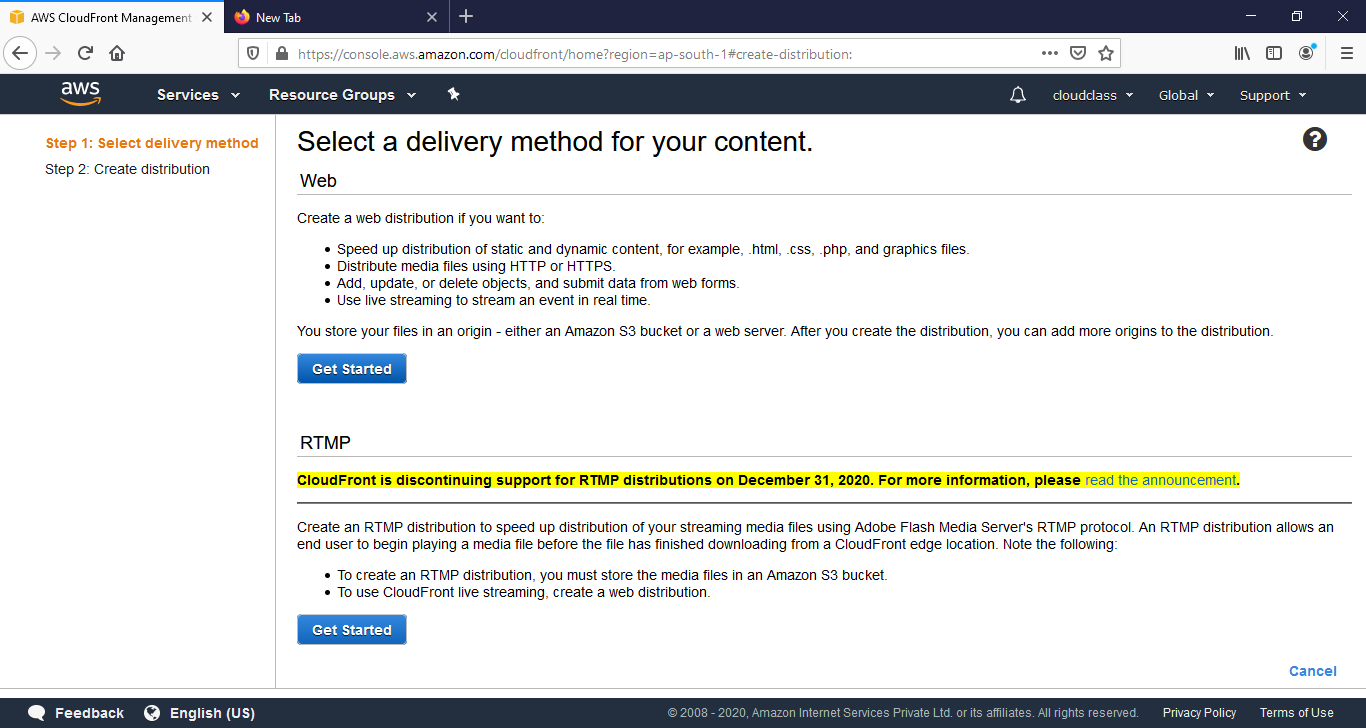


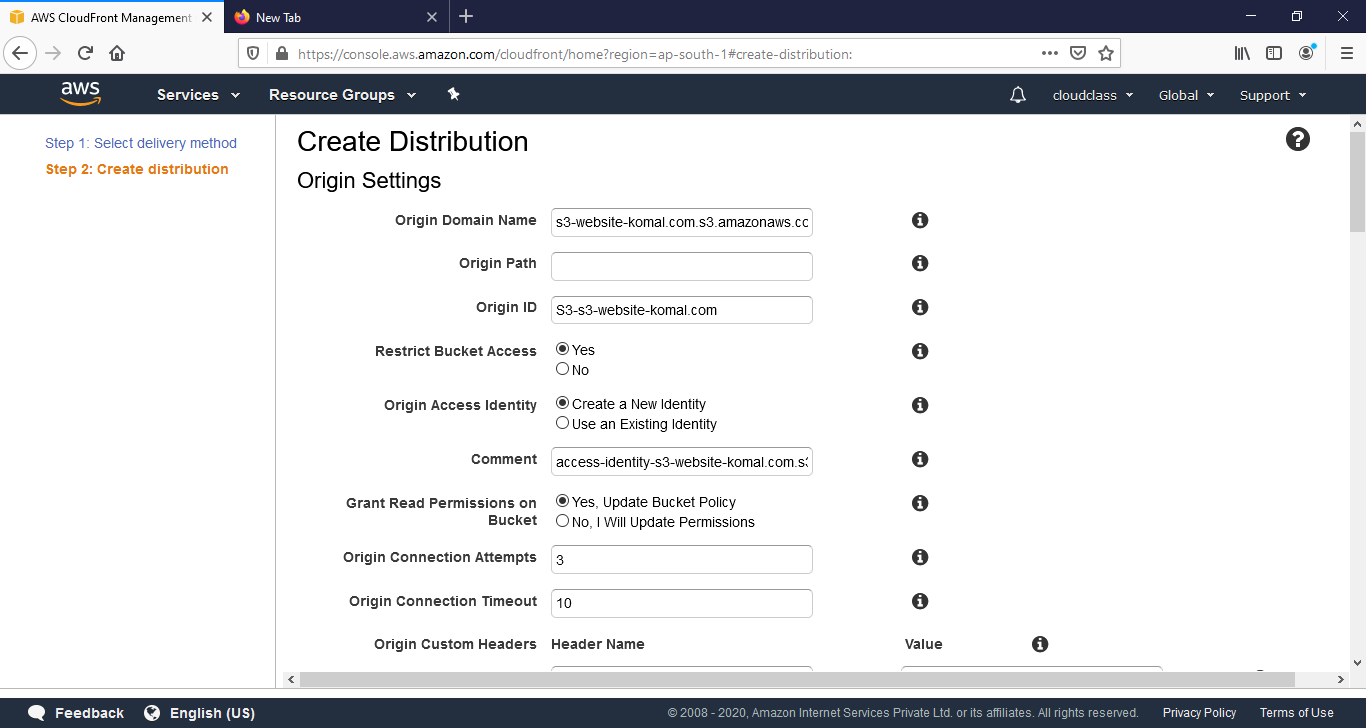


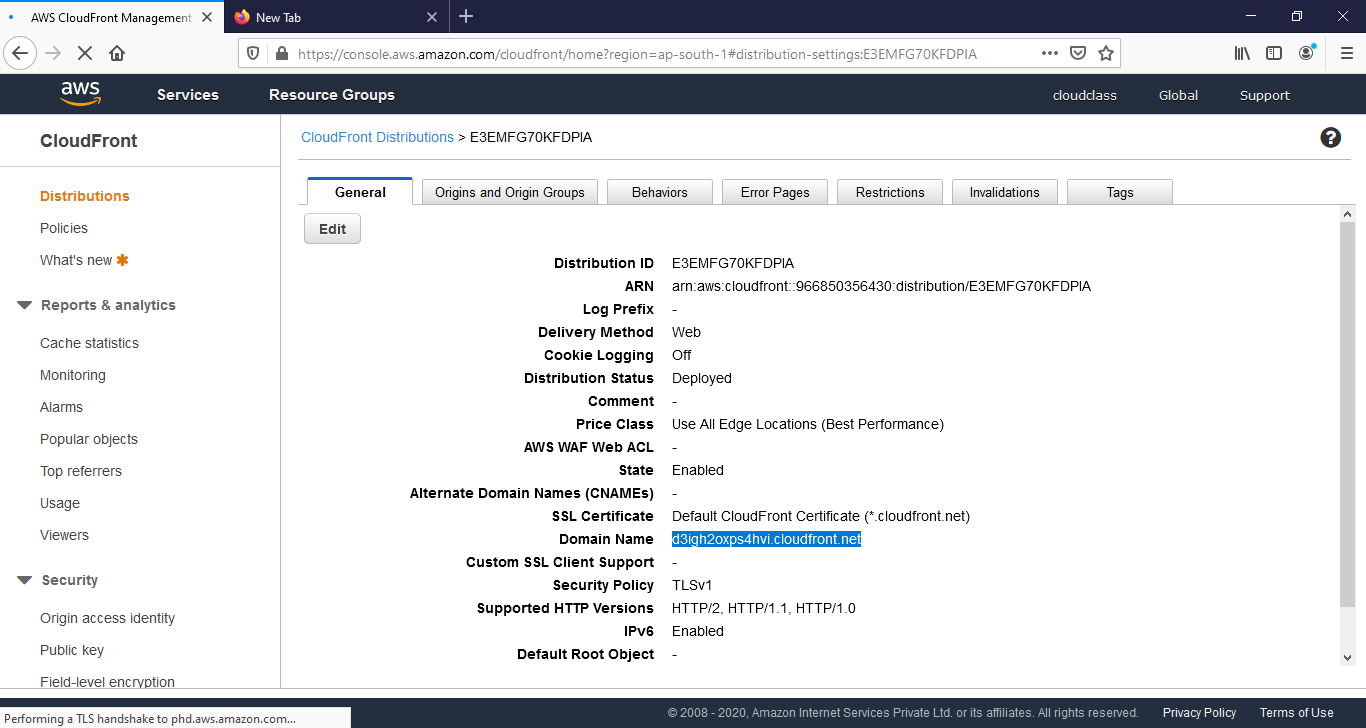
Now after at all your pipeline will be created .

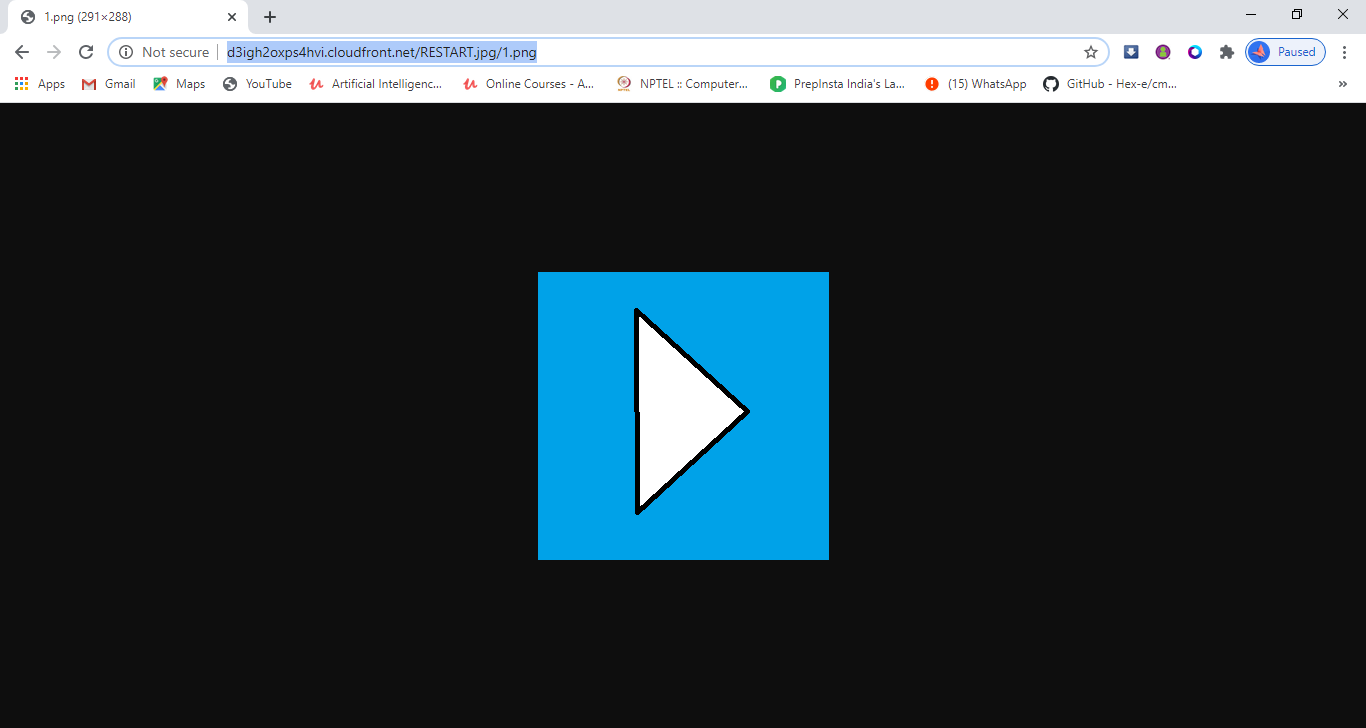
8 Create a Cloudfront using s3 bucket(which contains images) and use the Cloudfront URL to update in code in /var/www/html









 complete url to check it’s working or not.past the CloudFront domain name inside the html code of image.

https://www.linkedin.com/posts/komal-habura-b8446a1a2\_hybrid-multi-cloud-computing-task2-activity-6702160546743021568-GU7a