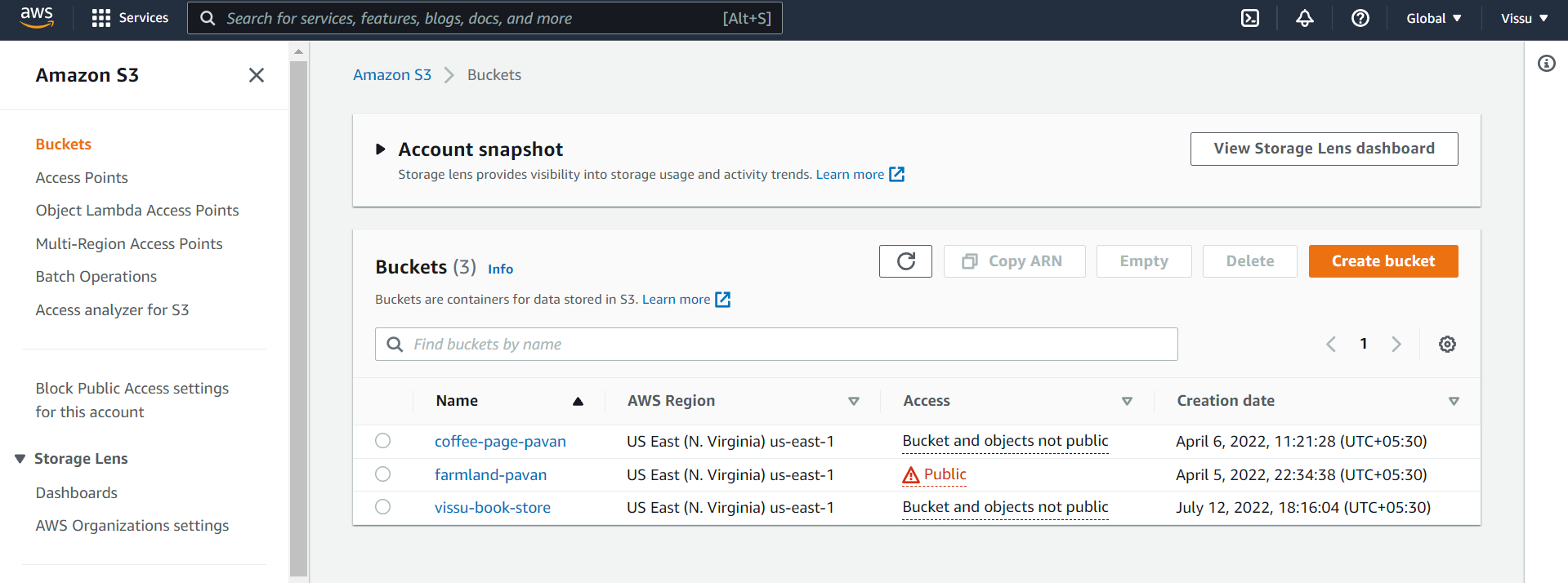
To save the build artifacts into amazon s3 we need two things

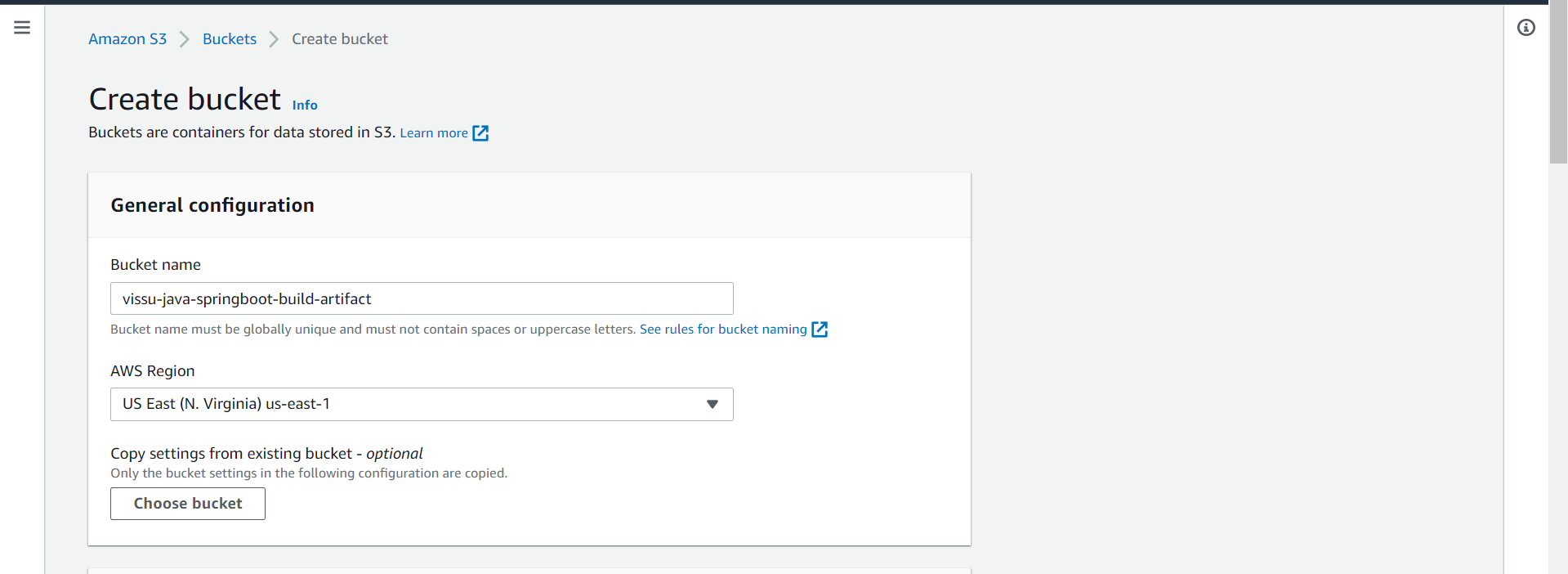
1. S3 bucket - private
2. AWS user with programmatic access.
3. We need to install a plugin called “Artifact Manager on S3” in jenkins.

**1. How to create a bucket in Amazon S3**

Go to AWS management console and login to the account and open s3 service.

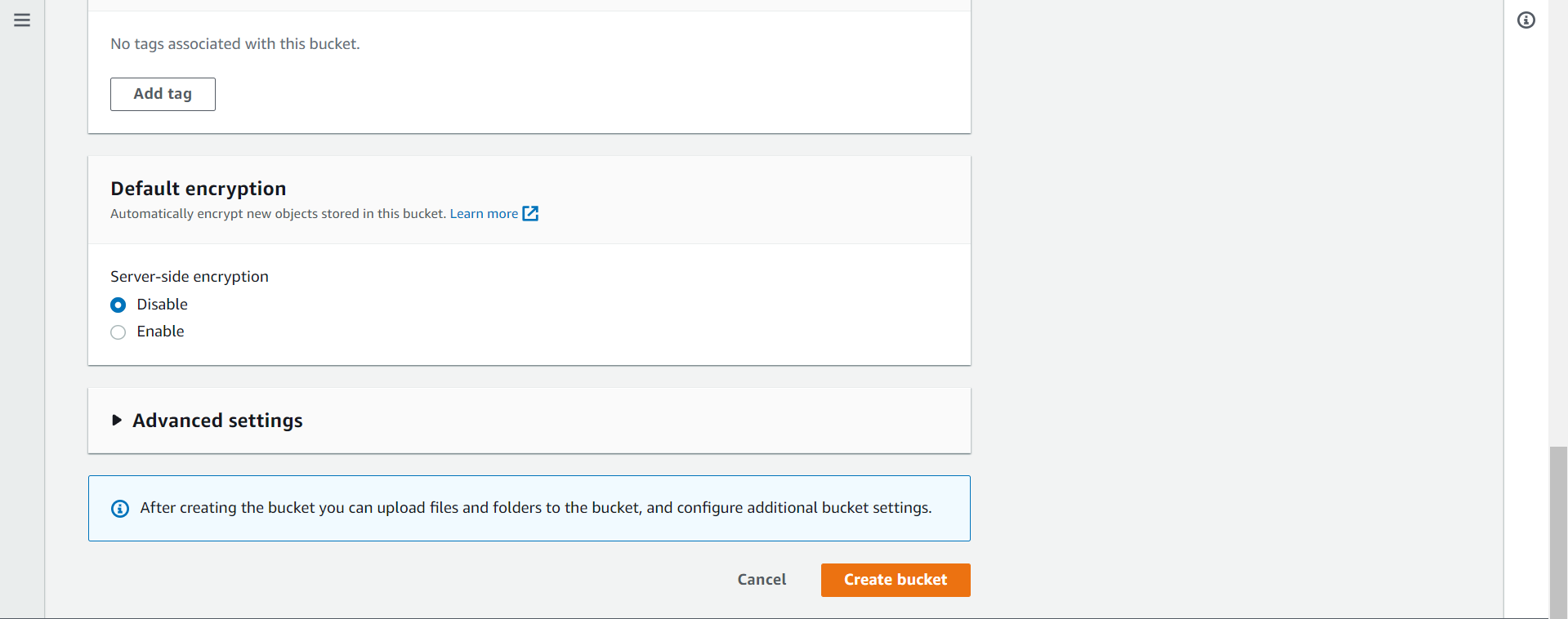


Click on “Create Bucket”

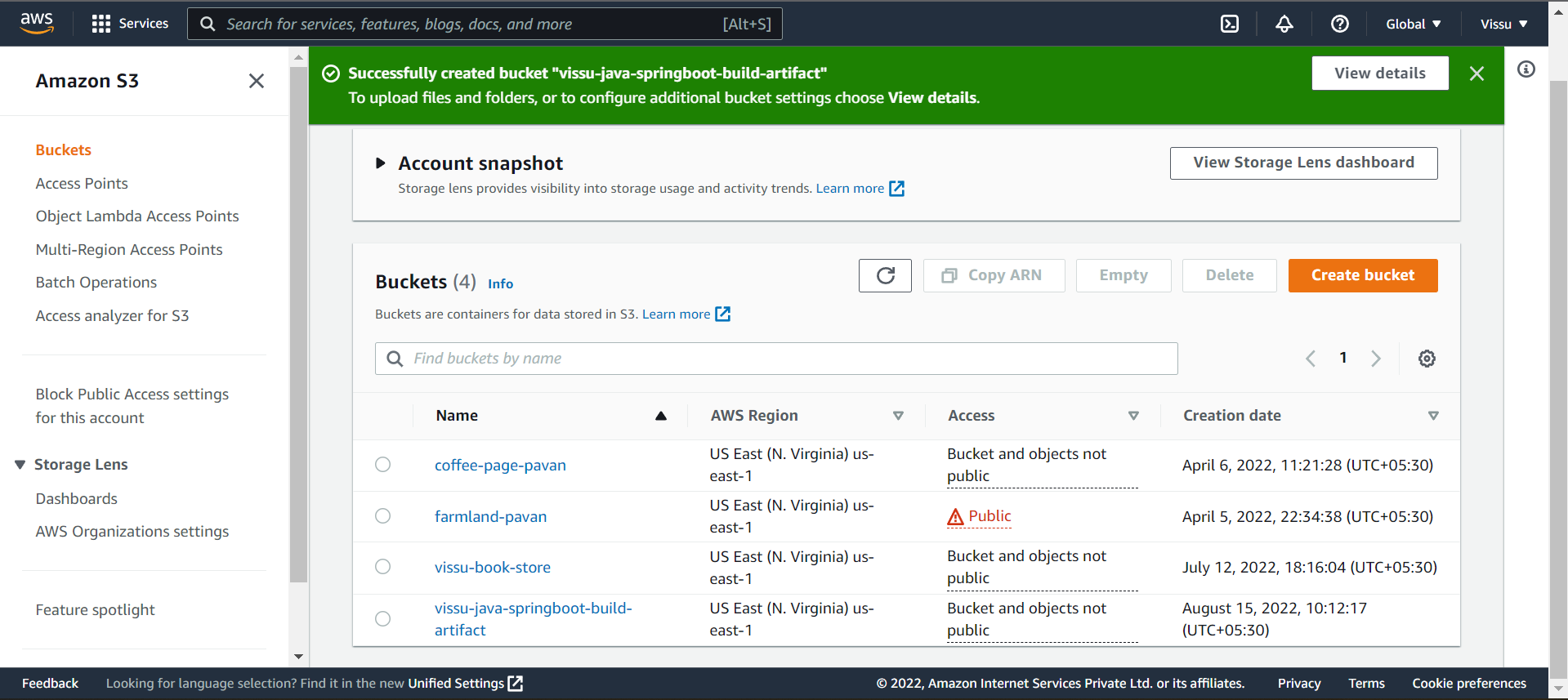


Enter the name for bucket.

Leave all other settings as default.



Click on “Create bucket”



Bucket created successfully.

Creating the policy for the IAM user with programmatic access to grant access to our s3 bucket.

Graphical user interface, application

Description automatically generated

Either we can give s3 full access or can create a new custom policy.

Here I am going for new custom policy.

Click on “Create Policy”

It will open a new tab for the policy creation

Graphical user interface, text, application, email

Description automatically generated

Select “JSON” to create the policy

Graphical user interface, text, application, email

Description automatically generated

Copy the below policy in the above JSON editor box.

---------------------------------------------------------------------------------------

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Action": [

"s3:GetBucketLocation",

"s3:ListAllMyBuckets"

],

"Resource": "arn:aws:s3:::\*"

},

{

"Effect": "Allow",

"Action": "s3:ListBucket",

"Resource": "arn:aws:s3:::vissu-java-springboot-build-artifact"

},

{

"Effect": "Allow",

"Action": [

"s3:PutObject",

"s3:GetObject",

"s3:DeleteObject"

],

"Resource": "arn:aws:s3:::vissu-java-springboot-build-artifact/\*"

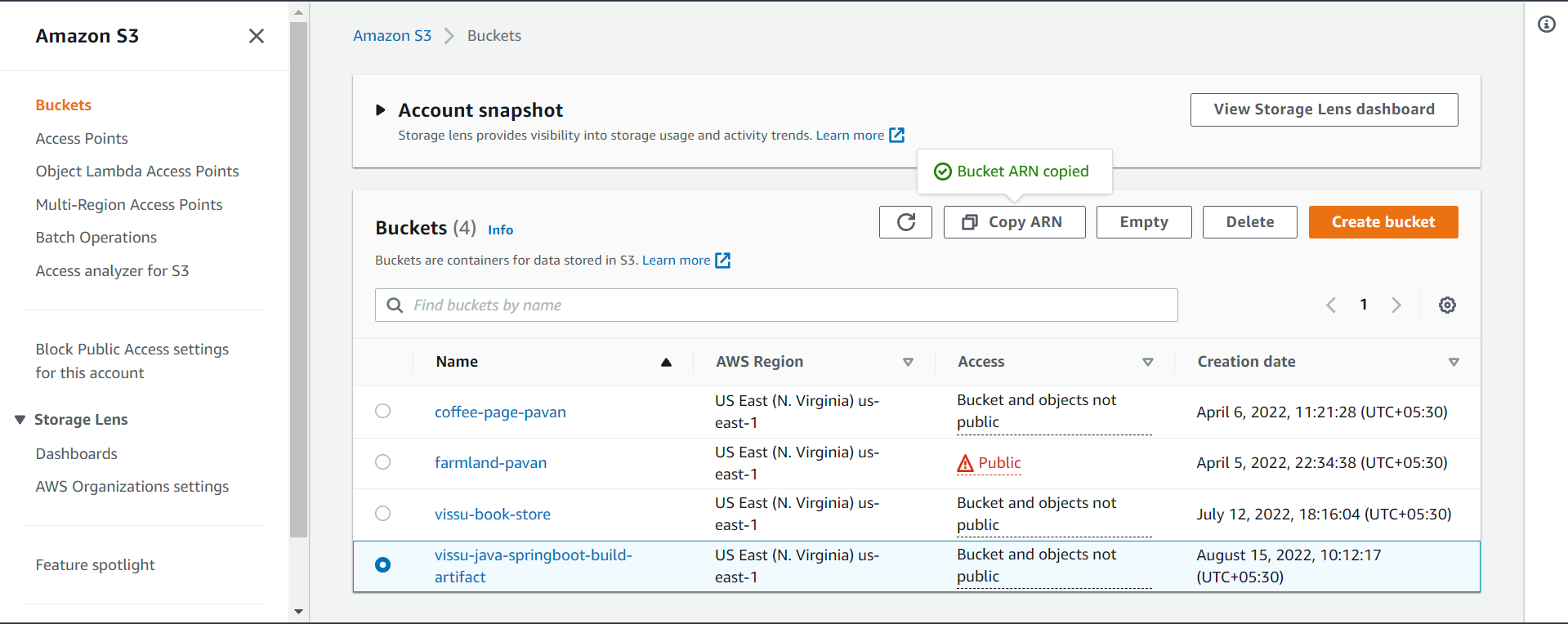
}

]

}

Note:

In the above policy replace the “Resource” value with your bucket ARN.

How to check the bucket ARN?  


Select your bucket and click on “Copy ARN”.

Graphical user interface, text, application, email

Description automatically generated

I just copied and pasted the policy in the JSON editor.

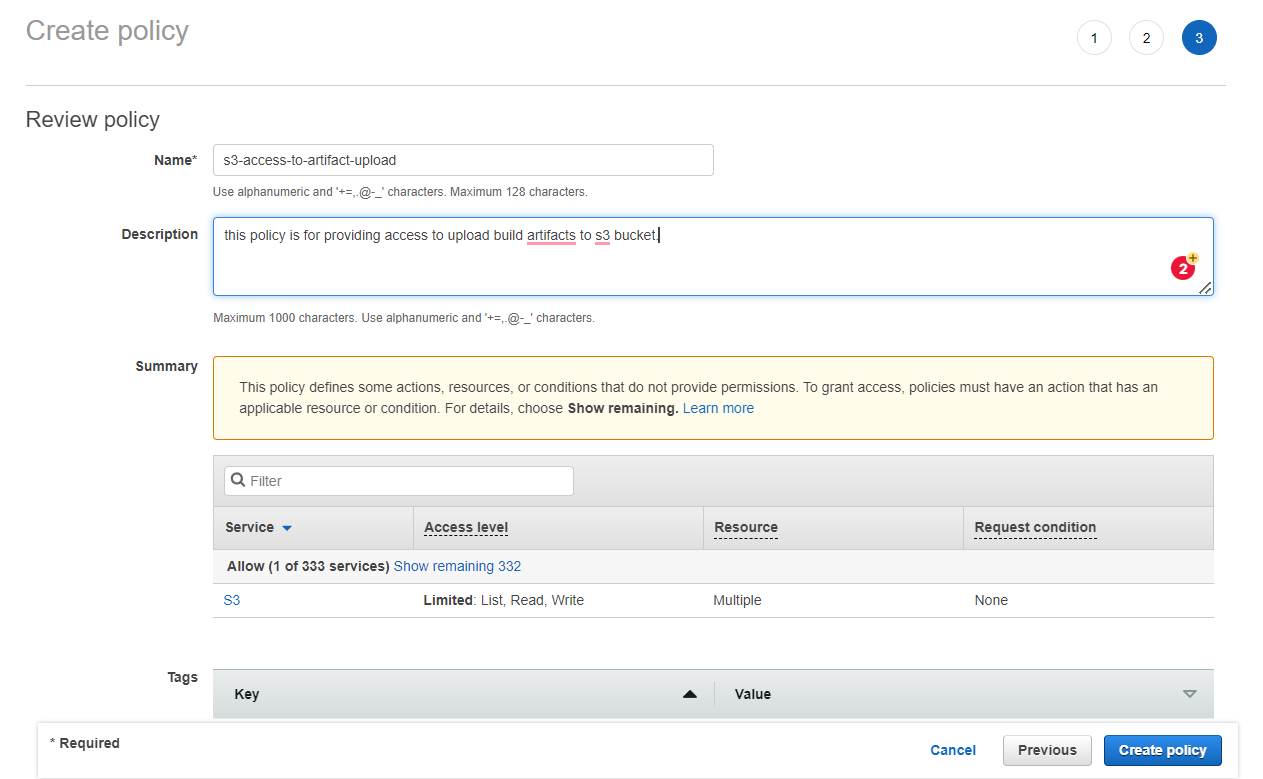
Next click on Tags

Graphical user interface, text, application

Description automatically generated

Adding tag is optional you can skip it.

Next click on Review.



Enter name for the policy

Enter description

Click on “Create policy”

Graphical user interface, text, application, email

Description automatically generated

Policy created successfully.

**2. How to create IAM User with programmatic access.**

Go to aws and open IAM Service.

Go to users and click on create user.

Graphical user interface, text, application, email

Description automatically generated

Enter name for the user

Select the Access Type = “Programmatic access”.

Click on next permissions

Graphical user interface, text, application, email

Description automatically generated

Choose the “Attach existing policies”

Search for our policy we have created above and select it.

Next click on Tags

Graphical user interface, text, application

Description automatically generated

Tags are optional, we can skip it

Next click on review

Graphical user interface, text, application, email

Description automatically generated

Click on create user.

Graphical user interface, text, application, email

Description automatically generated

Download the user credentials. We will use these credentials to connect our jenkins server with aws S3.

**3. How to install Artifact Manager on S3 plugin on jenkins?**

Open jenkins server

Go to Manage Jenkins 🡺 Manage plugins

Graphical user interface, application

Description automatically generated

Search for the plugin in available plugins, select it and click on “Install without restart”.

Graphical user interface, table

Description automatically generated

Installation started.

Graphical user interface, application

Description automatically generated

Installation completed.

Restart the Jenkins once.

Graphical user interface, application, Teams

Description automatically generated

Jenkins is restarting.

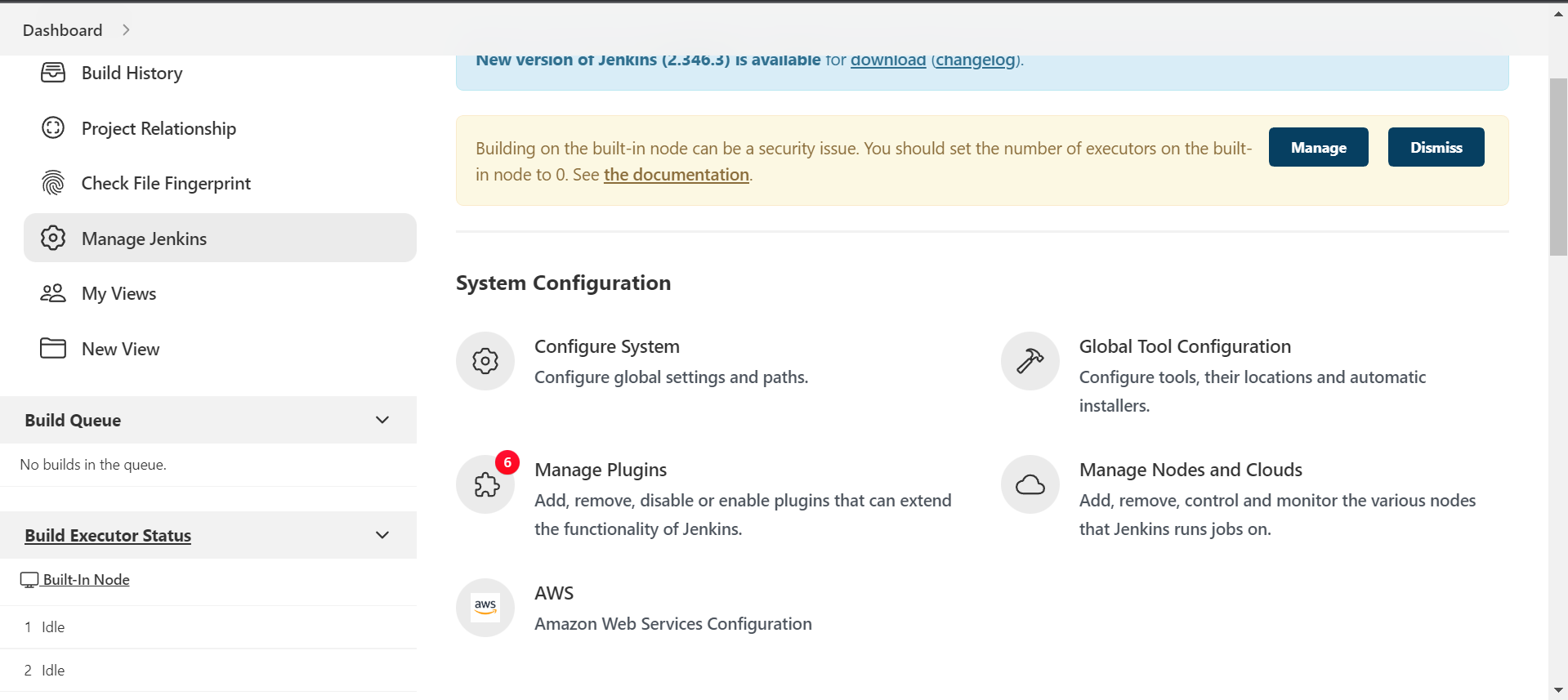
Graphical user interface, application, Teams

Description automatically generated

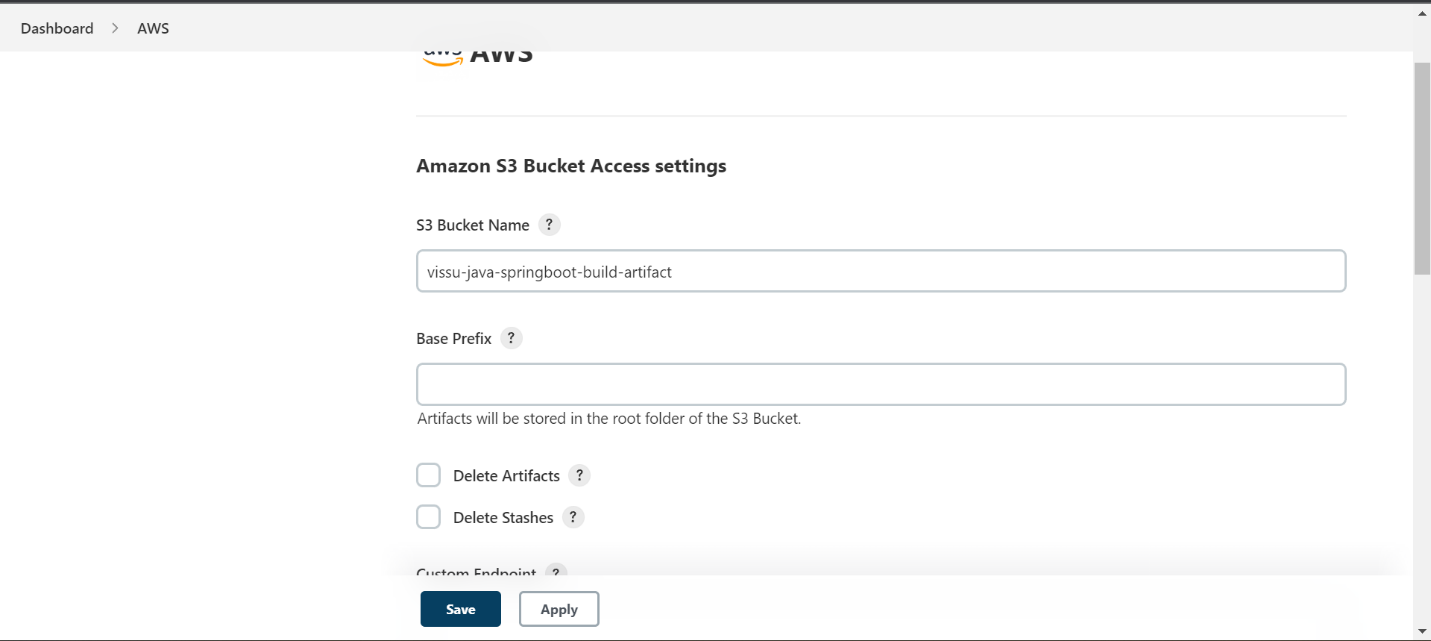
Jenkins restarted, login to the jenkins.

Now, configure the aws credentials in jenkins

**How to configure AWS in S3?**



Go to manage jenkins 🡺 AWS



Enter your S3 bucket name.

Graphical user interface, application

Description automatically generated

Select the “region” in which your bucket is created. Or you can leave it to auto.

Graphical user interface, text, application, email

Description automatically generated

Click on “Add” to add the credentials.

This will open a pop-up window.



Select the kind as “AWS Credentials”.

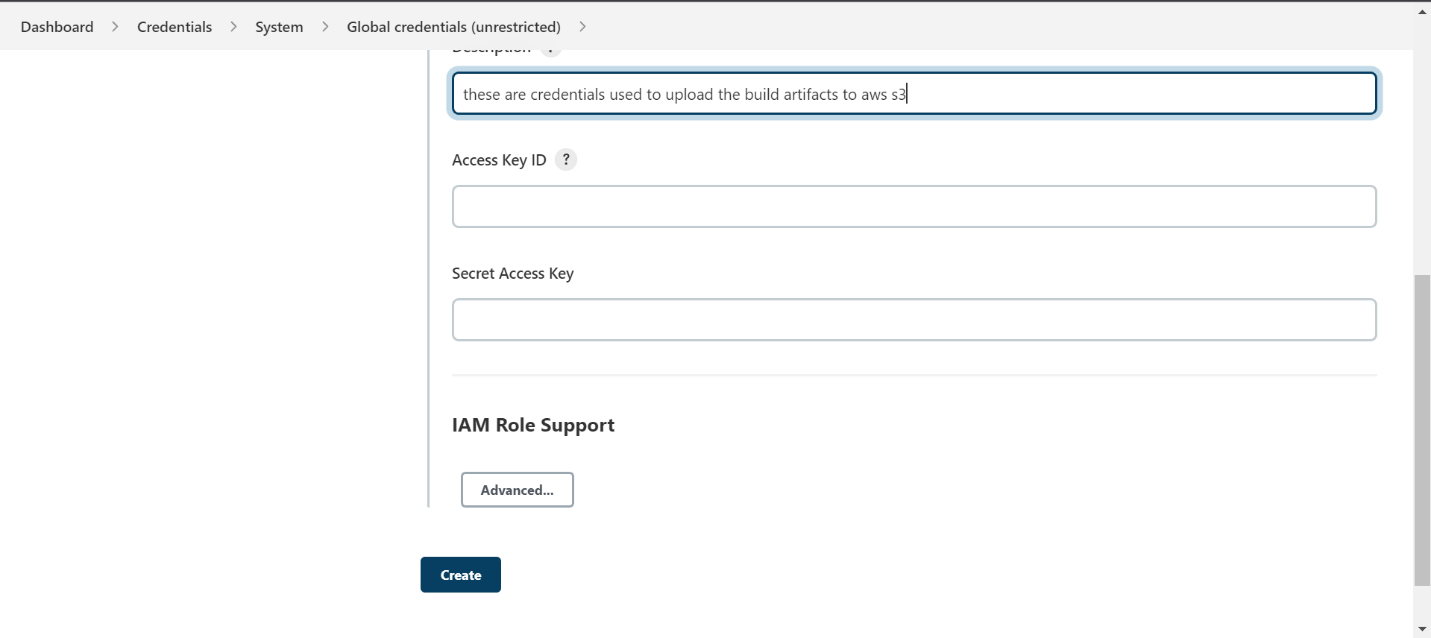
Graphical user interface, text, application, email

Description automatically generated

Select the scope

Give an ID for the credentials

Enter the description

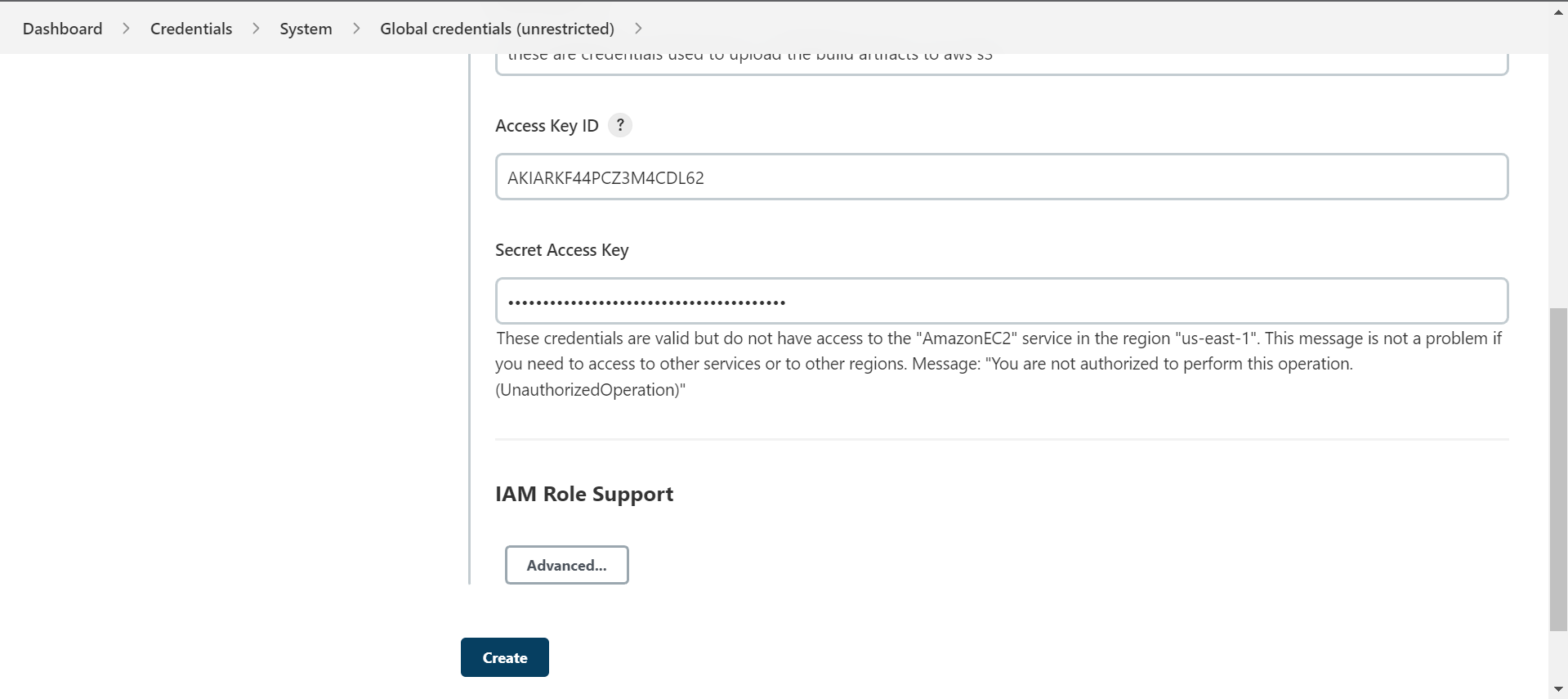


Enter the “Access key ID” and “Secret Access Key” for the IAM User we created above.

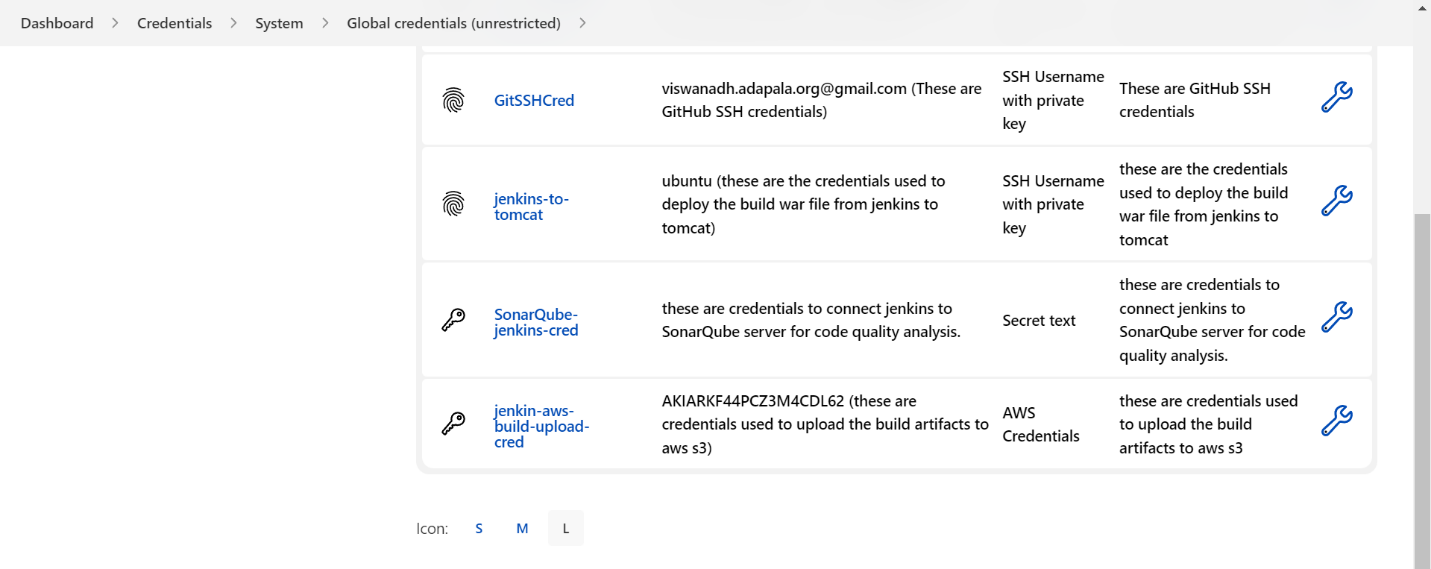
Text

Description automatically generated

Copy this access key id and access key and paste in the above fields.

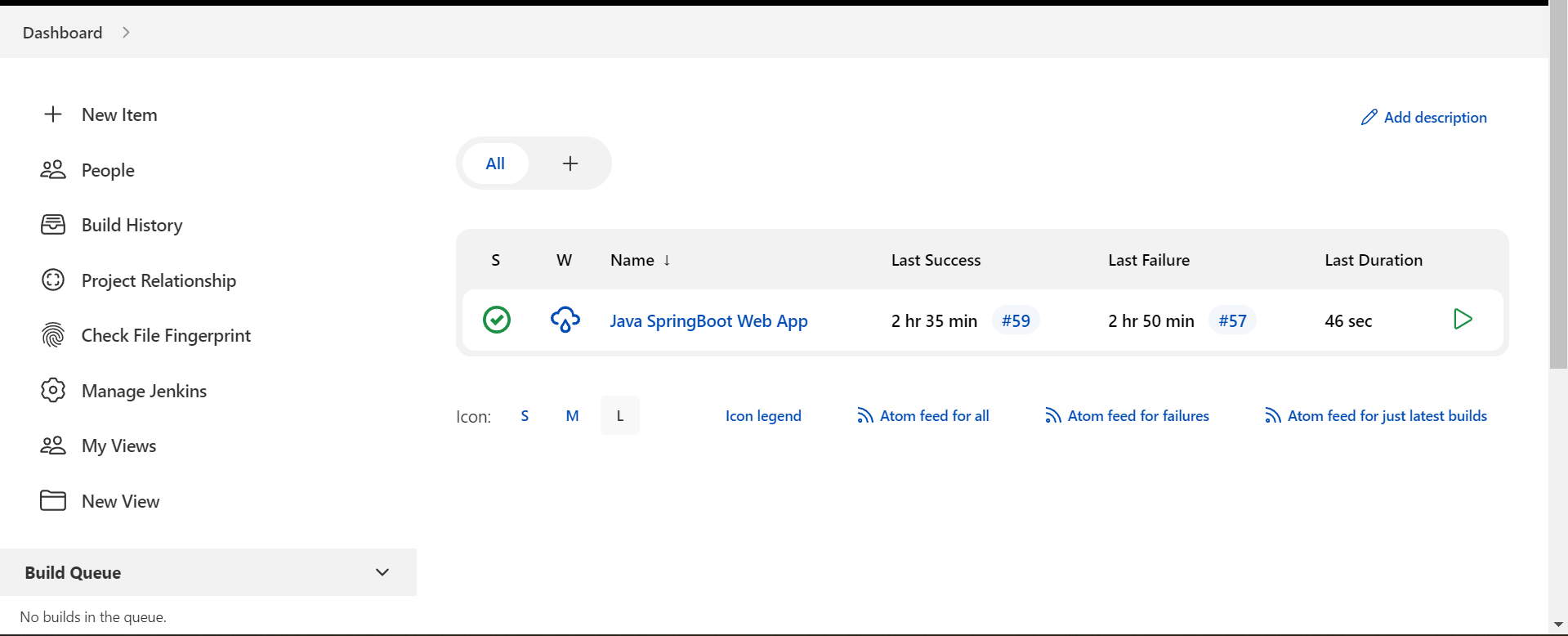


Click on create.



Credentials are added successfully.

Now, enable to store the build artifacts into S3 in jenkins.



Go to manage jenkins

Graphical user interface, text, application, chat or text message

Description automatically generated

Go to “configure system”

Graphical user interface, text, application, email, Teams

Description automatically generated

Go to “Artifactory Management for Builds”

Click on “Add”

Select “Cloud Artifact Storage”

Graphical user interface, application

Description automatically generated

Select the cloud provider as “Amazon S3”

Click on apply and save.

Now, add the build archive stage in our pipeline job.

How, to add build archive stage in jenkins pipeline job

Graphical user interface, application

Description automatically generated

Write the build archive code inside the steps.

Code: archiveArtifacts artifacts: '\*\*/\*.war', followSymlinks: false

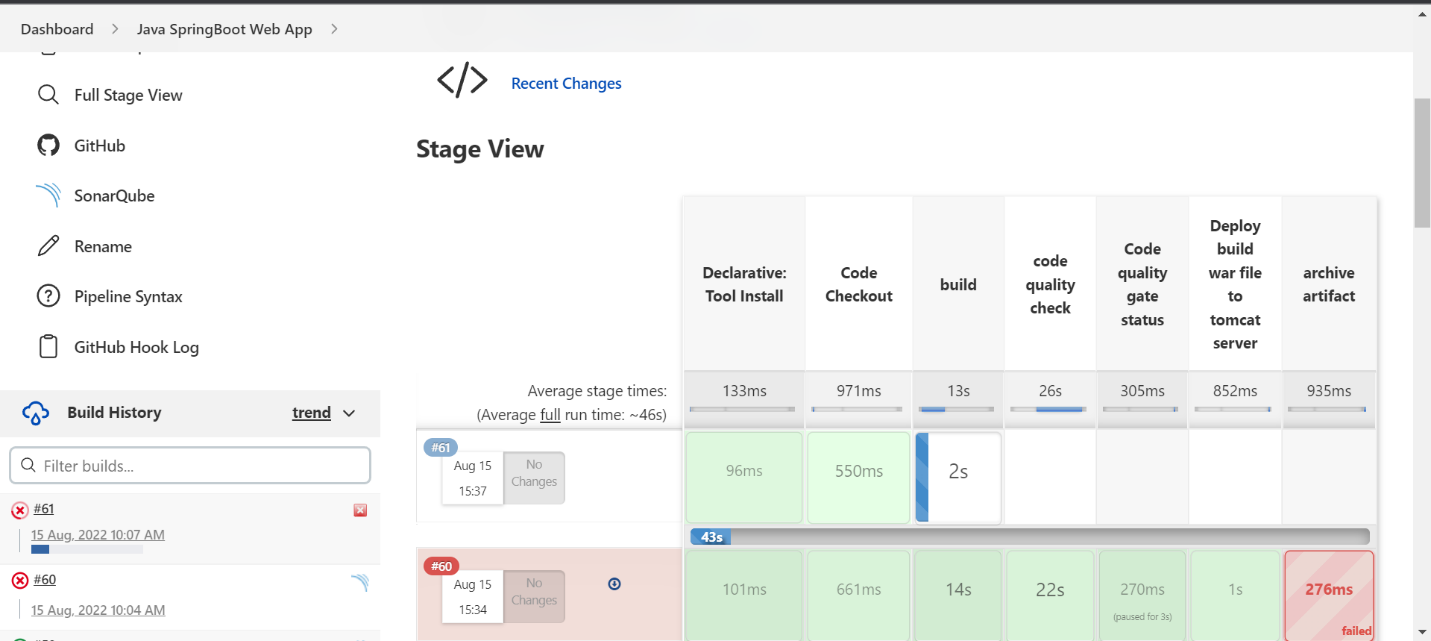
Write this code inside steps.

Graphical user interface, text, application

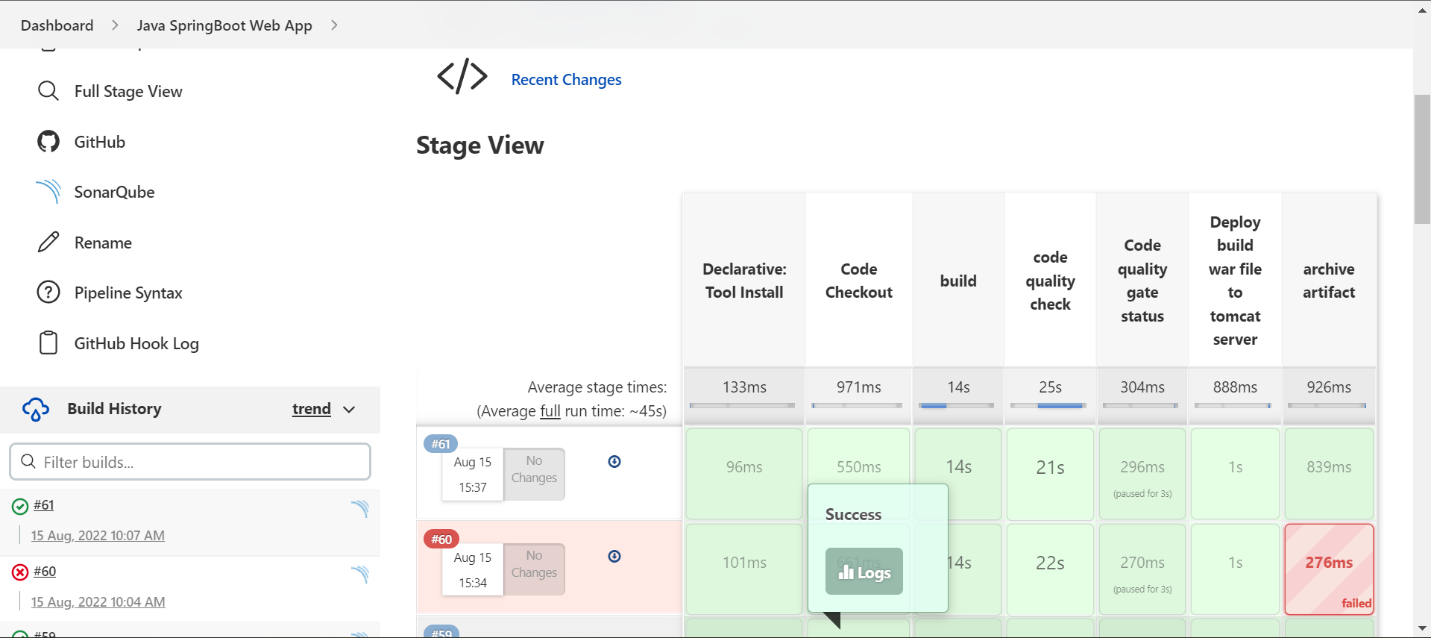
Description automatically generated

Click on Apply and Save.

Now, let’s build the pipeline again.



Our new build 61 was started.



New build got success.

Let’s check the archive logs

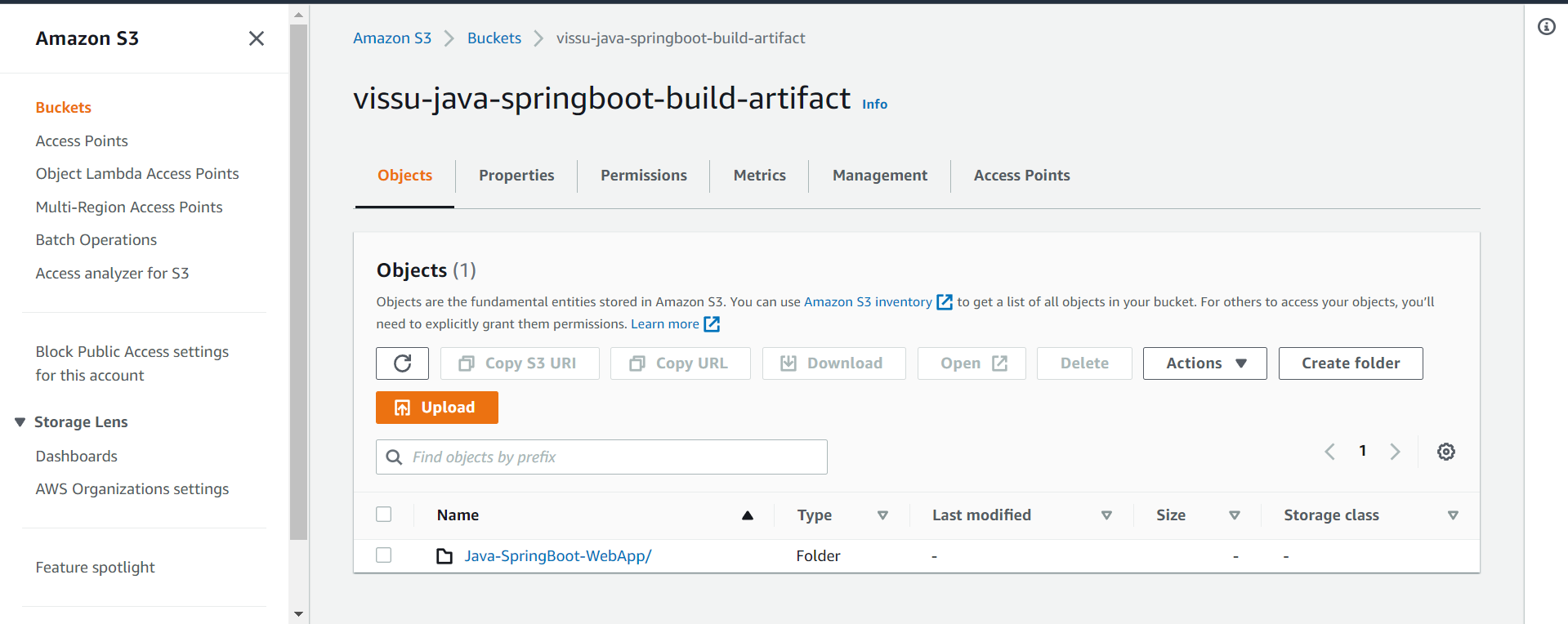
Graphical user interface, text, application, email

Description automatically generated

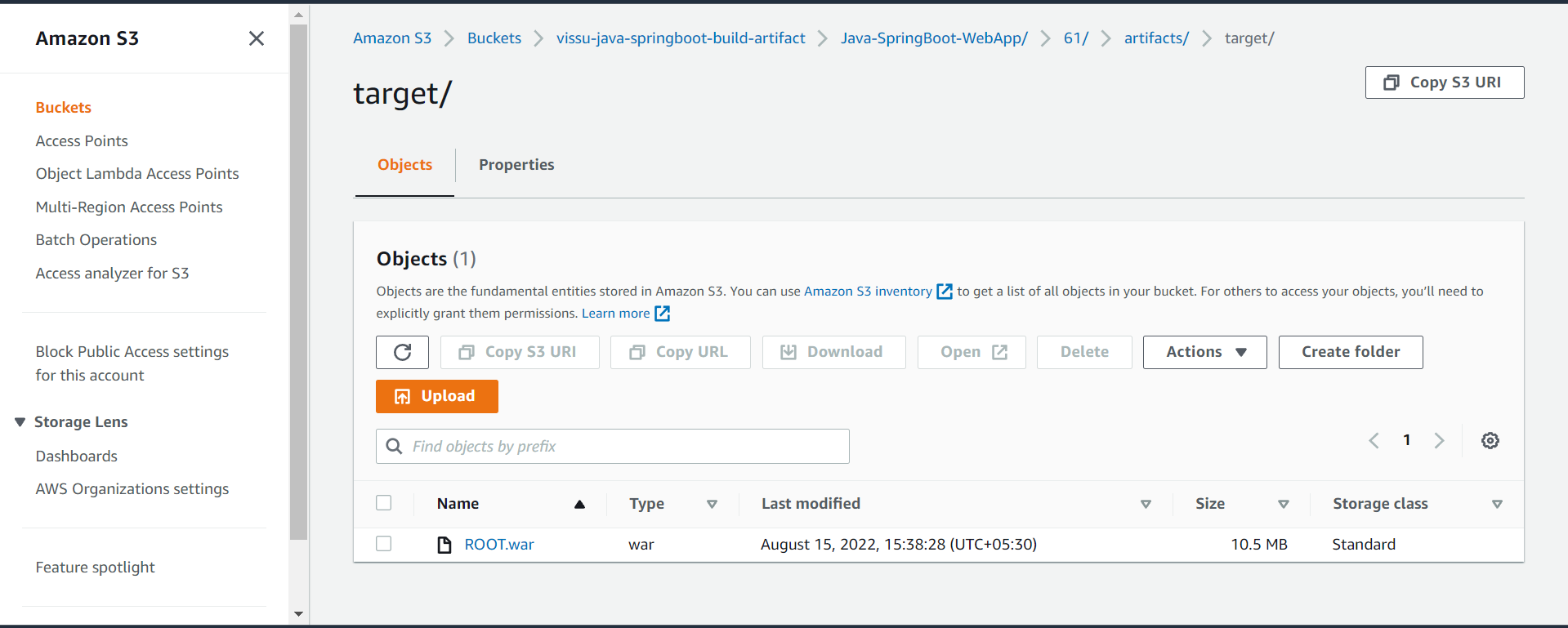
Artifacts are uploaded to aws s3 successfully.

Let’s verify.

Go to aws s3 bucket and refresh it once.



A new folder is created with our project GitHub repo name.



Our build artifact file “ROOT.war” file is created under build id 61 🡺 artifacts 🡺 target.