



## Usage instructions:

1. Launch the product via 1-click. **Please wait until** the instance passes **all** status checks and is running. You can connect using your Amazon private key and '**ubuntu**' login via SSH client.

- To update software, use: **sudo apt update** and **sudo apt upgrade**

---

## Quick Start Guide

The following is a step-by-step tutorial to get you started using FFMPEG. In this example, we will convert a mp4 video file into a .avi file using your AWS S3 bucket.

1. Go to the AWS Console and select S3 Dashboard. Next select “Create a bucket”. Change the Object Ownership to: “**ACLs enabled**” and **grant public access** to this bucket by “**unchecking**” for Block all public access. Save changes.

Only the bucket settings in the following configuration are copied.

Choose bucket

### Object Ownership [Info](#)

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

- ACLs disabled (recommended)**  
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

- ACLs enabled**  
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

Object Ownership

## Edit Block public access (bucket settings) [Info](#)

### Block public access (bucket settings)

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

**Block all public access**

Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

**Block public access to buckets and objects granted through *new* access control lists (ACLs)**

S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.

**Block public access to buckets and objects granted through *any* access control lists (ACLs)**

S3 will ignore all ACLs that grant public access to buckets and objects.

**Block public access to buckets and objects granted through *new* public bucket or access point policies**

S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.

**Block public and cross-account access to buckets and objects through *any* public bucket or access point policies**

S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

Cancel

Save changes

2. Next, upload a (mp4) file...(aka. a media file you want to convert) into your new bucket.
3. After the file has been uploaded successfully, click on the file and click "Permissions" then "Edit".

Properties

Permissions

Versions

### Access control list (ACL)

Grant basic read/write permissions to AWS accounts. [Learn more](#)

Grantee

Properties			Permissions	Versions	
<b>Access control list (ACL)</b> Grant basic read/write permissions to AWS accounts. <a href="#">Learn more</a>					<b>Edit</b>
Grantee	Object	Object ACL			
Object owner (your AWS account) Canonical ID:  4348853c94742493ce0b96e1401d1eb30fe5d421b0b1b5e2f5d529bb7430b612	Read	Read, Write			
Everyone (public access) Group:  http://acs.amazonaws.com/groups/global/AllUsers	-	-			
Authenticated users group (anyone with an AWS account) Group:  http://acs.amazonaws.com/groups/global/AuthenticatedUsers	-	-			

- Check “Read” & “Read” Write” in the “**Everyone (public access)**” access section. Agree to the notice and save.

Access control list (ACL)		
Grant basic read/write permissions to AWS accounts. <a href="#">Learn more</a>		
Grantee	Objects	Object ACL
Object owner (your AWS account) Canonical ID:  4348853c94742493ce0b96e1401d1eb30fe5d421b0b1b5e2f5d529bb7430b612	<input checked="" type="checkbox"/> Read	<input checked="" type="checkbox"/> Read <input checked="" type="checkbox"/> Write
Everyone (public access) Group:  http://acs.amazonaws.com/groups/global/AllUsers	<input checked="" type="checkbox"/> Read	<input checked="" type="checkbox"/> Read <input type="checkbox"/> Write
Authenticated users group (anyone with an AWS account) Group:  http://acs.amazonaws.com/groups/global/AuthenticatedUsers	<input type="checkbox"/> Read	<input type="checkbox"/> Read <input type="checkbox"/> Write

- Be sure to **Copy the file path url**. “Object URL” is found under properties tab of your file.
- Next go back to your [AWS Console Home](#) and log into the to the “IAM” section. Create an IAM role for your Instance.
  - Go to: “**Roles**” under “access management”

- b. Click **“Create role”**
- c. Select **“AWS service”** & **“EC2”** for the “use case”
- d. Click **“Next”**
- e. In search for permission policies, type **“AmazonS3FullAccess”**
- f. Check “AmazonS3FullAccess” and click “Next”

## Add permissions Info

**Permissions policies** (Selected 1/826) Info  
Choose one or more policies to attach to your new role.

Filter policies by property or policy name and press enter. 5 matches

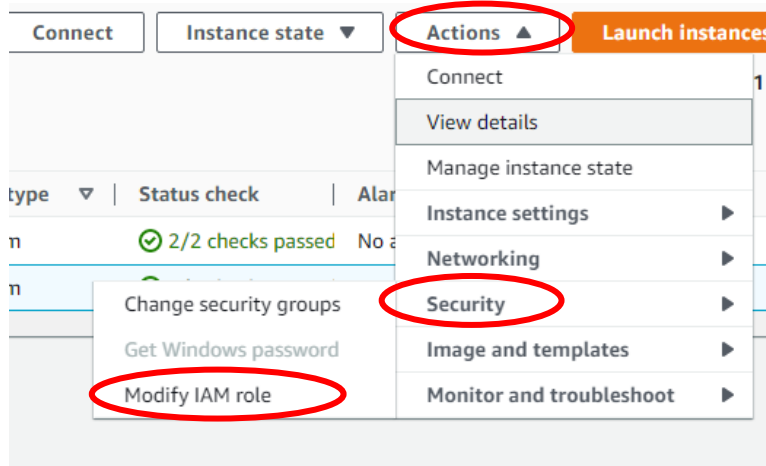
“amazons3” X Clear filters

<input type="checkbox"/>	Policy name	Type	Description
<input checked="" type="checkbox"/>	AmazonS3FullAccess	AWS ma...	Provides full access to all buckets via the AWS Management Console.
<input type="checkbox"/>	AmazonS3ReadOnlyAccess	AWS ma...	Provides read only access to all buckets via the AWS Management Console.
<input type="checkbox"/>	AmazonS3OutpostsFullAccess	AWS ma...	Provides full access to Amazon S3 on Outposts via the AWS Management Conso...
<input type="checkbox"/>	AmazonS3ObjectLambdaExecutionRolePolicy	AWS ma...	Provides AWS Lambda functions permissions to interact with Amazon S3 Object ...

- g. Name the Role and “click “Create role”

7. Go back to your **AWS console**, select > **EC2 Dashboard** > **Instances running**.

- a. Check your “Instance” that is running.
- b. Under **“Actions”**, select **“Security”** and then **“Modify IAM Role”**



- c. **Search** for the role you just created in the search tab and select it.
  - d. Then click “Update IAM role”
8. Now log back into your **Ubuntu server**.
9. Now move the uploaded file to your ubuntu server using the AWS CLI and wget command:
  - a. To view your S3 buckets, at the ubuntu command prompt, run:  
**aws s3 ls**
  - b. Now transfer the file from your bucket to your Ubuntu server using **wget**. Use the object url of the file from (**step 5 above**). Run this command:  
**wget https://your object url of the file in your S3 bucket**

*For example: [wget https://codecreatorassets.s3.amazonaws.com/myfile.mp4](https://codecreatorassets.s3.amazonaws.com/myfile.mp4)*

  - c. Wait until it is complete. To verify the file was uploaded to your server, use: “**ls -la**” command to see the file has been transferred.
  - d. “Exit” and return to Ubuntu prompt.
10. Again at the ubuntu command, type the following command to **convert the file** from **mp4 to AVI**.  
  
\*Use: “ffmpeg -i” command\*as follows:  
  
**ffmpeg -i yourfilename.mp4 yourfilename.avi**  
  
*\*This command will tell the software to change the mp4 file and convert it into an avi file.*
11. Wait until it is converted and use the “ls” to see a **new file** in your directory converted into an **.avi file**.
12. Finally send the new file back to your **S3 bucket** for download. Use the “cp” command.  
**aws s3 cp yourfilename.avi s3://yourbucketname**
13. Now return to your **AWS S3 console** and you should now see the new avi file uploaded to your s3 bucket you created earlier.

For more information please visit the FFMPEG website: <https://www.ffmpeg.org/ffmpeg.html>

## **AWS Data**

- Data Encryption Configuration: This solution does not encrypt data within the running instance.
- User Credentials are stored: /root/.ssh/authorized\_keys & /home/ubuntu/.ssh/authorized\_keys
- Monitor the health:
  - Navigate to your Amazon EC2 console and verify that you're in the correct region.
  - Choose Instance and select your launched instance.
  - Select the server to display your metadata page and choose the Status checks tab at the bottom of the page to review if your status checks passed or failed.

## **Extra Information: (Optional)**

### **Allocate Elastic IP**

To ensure that your instance **keeps its IP during restarts** that might happen, configure an Elastic IP. From the EC2 console:

1. Select ELASTIC IPs.
2. Click on the ALLOCATE ELASTIC IP ADDRESS.
3. Select the default (Amazon pool of IPv4 addresses) and click on ALLOCATE.
4. From the ACTIONS pull down, select ASSOCIATE ELASTIC IP ADDRESS.
5. In the box that comes up, note down the Elastic IP Address, which will be needed when you configure your DNS.
6. In the search box under INSTANCE, click and find your INSTANCE ID and then click ASSOCIATE.
7. Your instance now has an elastic IP associated with it.
8. For additional help: <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html>