DistroTV Studio Getting Started: Working with an Existing Stream

GETTING STARTED GUIDE

DISTROSCALE

Updated 2025-07-18

Copyright © 2025 DistroScale, Inc. and/or its affiliates. All rights reserved.

Table of Contents

Copyright © 2025 DistroScale, Inc. and/or its affiliates. All rights reserved.

Chapter 1. Introduction

Welcome to the DistroTV Studio Getting Started Guide.
DistroTV Studio allows you to ingest existing linear streams from different formats such as HLS, RTMP, and SRT. Once ingested, DistroTV Studio can be used to transcode the input and output these as HLS with extras such as your own ad marker insertions.

The purpose of this document is to provide a complete, hands-on walkthrough for ingesting an existing HLS stream through DistroTV Studio.
In this guide, you will learn how to:

- Correctly configure the necessary AWS services, including an S3 Bucket, IAM permissions, and Security Groups.
- Launch and initialize the DistroTV Studio EC2 instance from AWS Marketplace AMIs.
- Launch and verify that your channel is live and streaming.
- By the end of this guide, you will have a functional re-streamed channel passed through DistroTV Studio and the foundational knowledge to begin leveraging the DistroTV Studio toolkit such. For advanced configurations and detailed instructions on these features, please refer to our comprehensive **User Guide**.

Prerequisites
Before you begin, you should have:

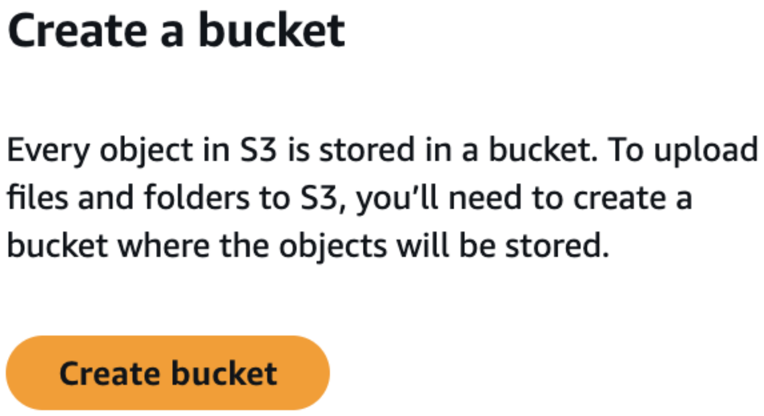
- An active AWS account with permissions to create S3 buckets, IAM roles and policies, and EC2 instances.
- Basic familiarity with navigating the AWS Management Console.

Copyright © 2025 DistroScale, Inc. and/or its affiliates. All rights reserved.

Chapter 2. Setting Up the AWS Environment

1. Creating an S3 Bucket

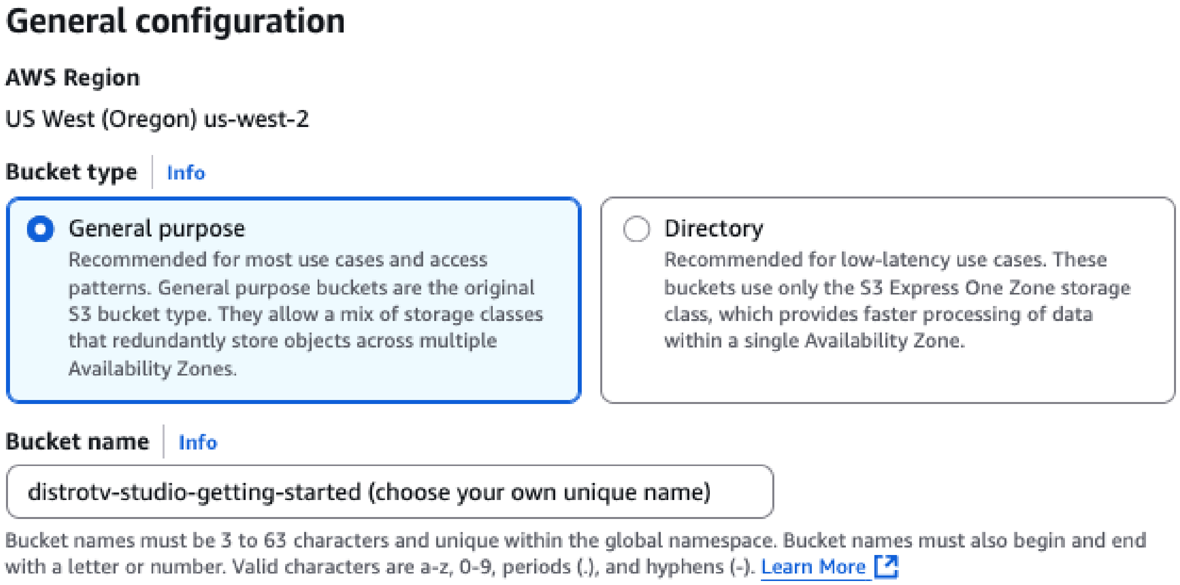
- Open AWS console and navigate to **S3** service
- Click “**Create a bucket**” button



Create a bucket

Every object in S3 is stored in a bucket. To upload files and folders to S3, you’ll need to create a bucket where the objects will be stored.

- Apply the following settings:(if an option is not mentioned, keep it on the default value)
  - Bucket type: “**General purpose**”
  - Object Ownership: “**ACLs enabled”**
  - Uncheck “**Block *all* public access**”
    - Check “**I acknowledge that the current settings might result in this bucket and the objects within becoming public.**”
- Click “**Create Bucket**” button
- **Note the name of your bucket as you will be using this in later steps.**



General configuration

AWS Region

US West (Oregon) us-west-2

Bucket type | Info

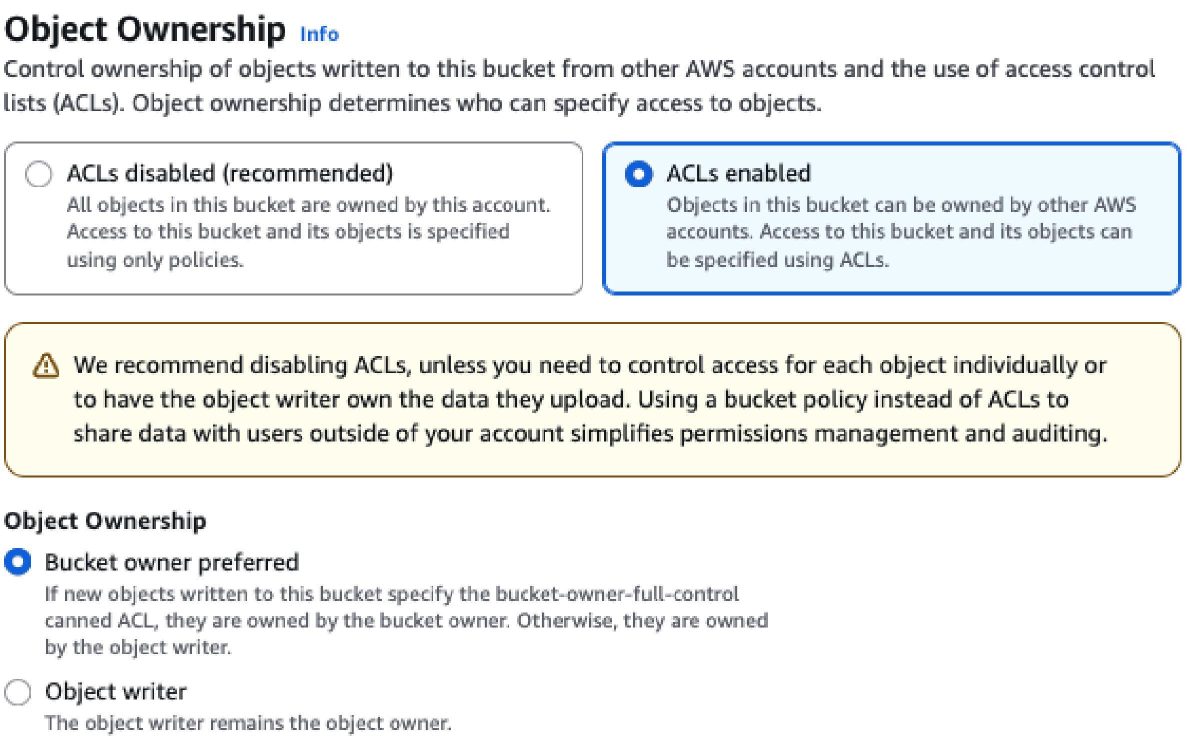
● General purpose
Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

○ Directory
Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

Bucket name | Info

distrotv-studio-getting-started (choose your own unique name)

Bucket names must be 3 to 63 characters and unique within the global namespace. Bucket names must also begin and end with a letter or number. Valid characters are a-z, 0-9, periods (.), and hyphens (-). Learn More



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.

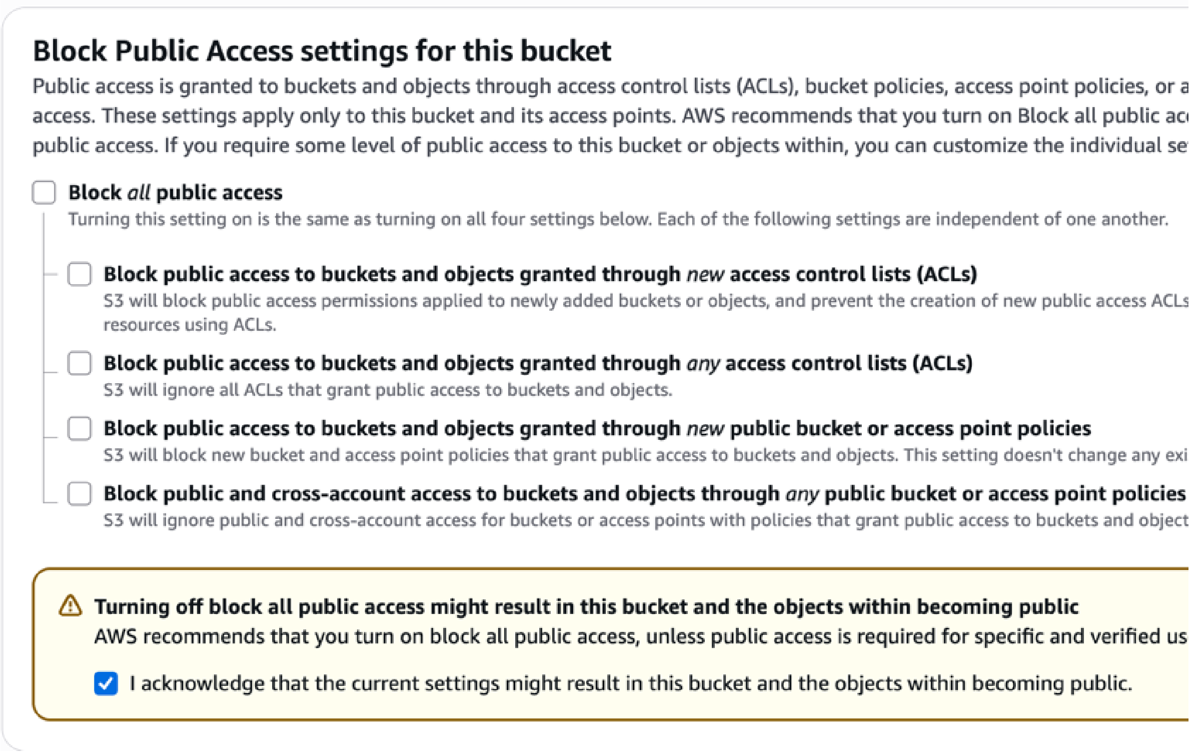
● ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

⚠ We recommend disabling ACLs, unless you need to control access for each object individually or to have the object writer own the data they upload. Using a bucket policy instead of ACLs to share data with users outside of your account simplifies permissions management and auditing.

Object Ownership

● Bucket owner preferred
If new objects written to this bucket specify the bucket-owner-full-control canned ACL, they are owned by the bucket owner. Otherwise, they are owned by the object writer.

○ Object writer
The object writer remains the object owner.



Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or a access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public ac public access. If you require some level of public access to this bucket or objects within, you can customize the individual se

□ **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 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 exi
- □ **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 object

⚠ **Turning off block all public access might result in this bucket and the objects within becoming public**
AWS recommends that you turn on block all public access, unless public access is required for specific and verified us

☑ I acknowledge that the current settings might result in this bucket and the objects within becoming public.

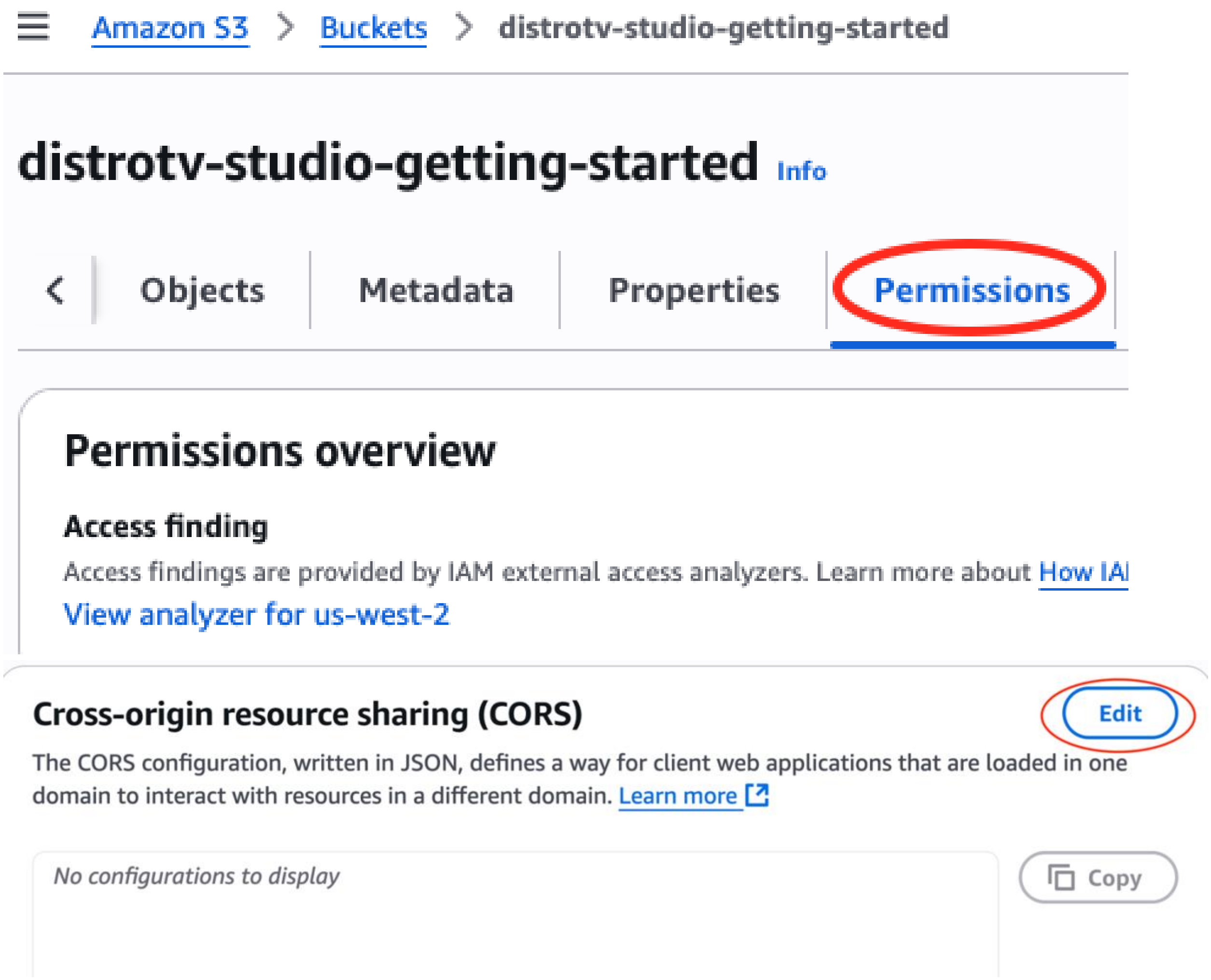
Copyright © 2025 DistroScale, Inc. and/or its affiliates. All rights reserved.

2. Edit Bucket Permissions

- Navigate to the S3 bucket you just created
- Select the “**Permissions”** tab to configure its access settings.
- In the “**Permissions**” tab, scroll down to “**Cross-origin resource sharing (CORS)**”
- Select “**Edit**” in the top right of the **CORS** box
- Paste the following into the text box

```
[
    {
        "AllowedHeaders": [
            "*"
        ],
        "AllowedMethods": [
            "GET",
            "HEAD"
        ],
        "AllowedOrigins": [
            "*"
        ],
        "ExposeHeaders": [],
        "MaxAgeSeconds": 3000
    }
]
```

- Click “**Save changes**” button
- Consider leaving this AWS S3 tab open in your browser as you will need to reference your bucket name in the following steps

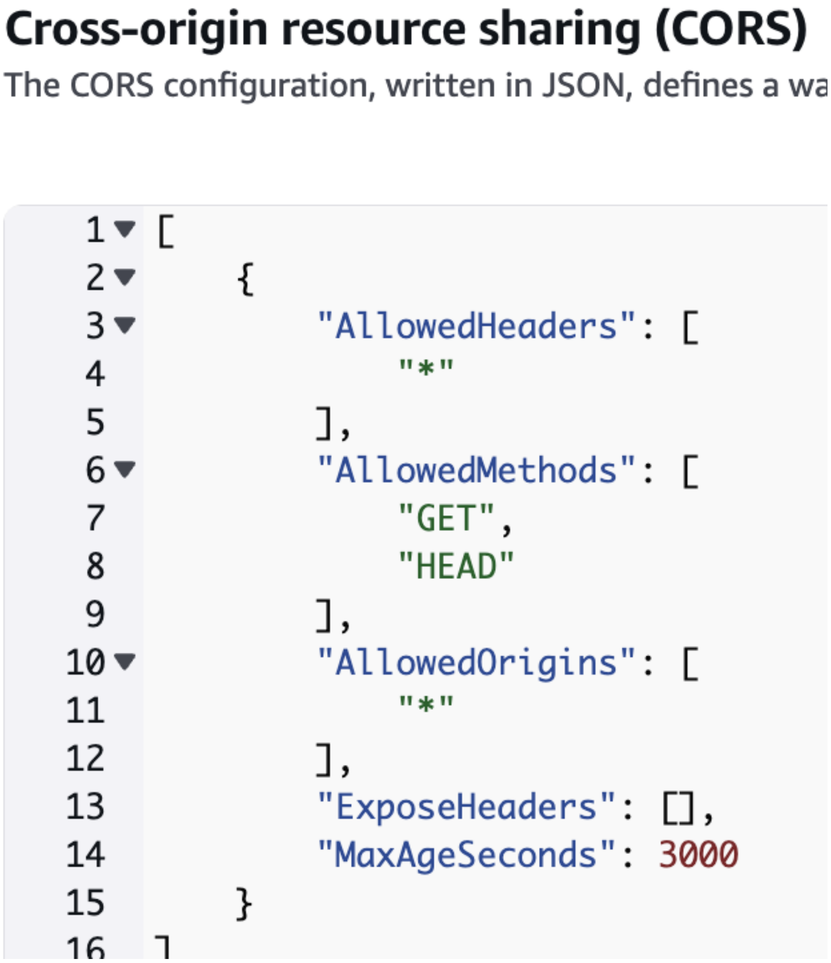

Amazon S3
Buckets
distrotv-studio-getting-started
distrotv-studio-getting-started
Info
Objects
Metadata
Properties
Permissions
Permissions overview
Access finding
View analyzer for us-west-2
Cross-origin resource sharing (CORS)
Edit
Copy

Access findings are provided by IAM external access analyzers. Learn more about How IAI

The CORS configuration, written in JSON, defines a way for client web applications that are loaded in one domain to interact with resources in a different domain. Learn more

No configurations to display

Copyright © 2025 DistroScale, Inc. and/or its affiliates. All rights reserved.



Cross-origin resource sharing (CORS)

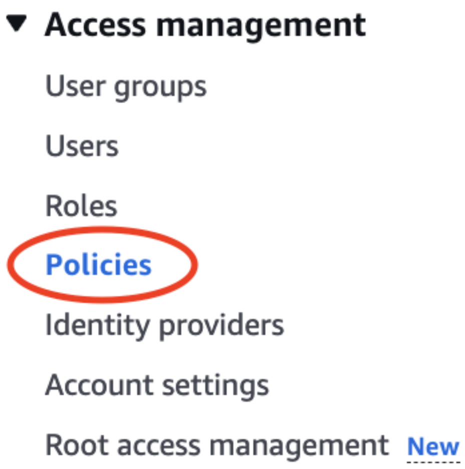
The CORS configuration, written in JSON, defines a wa

```
[
    {
        "AllowedHeaders": [
            "*"
        ],
        "AllowedMethods": [
            "GET",
            "HEAD"
        ],
        "AllowedOrigins": [
            "*"
        ],
        "ExposeHeaders": [],
        "MaxAgeSeconds": 3000
    }
]
```

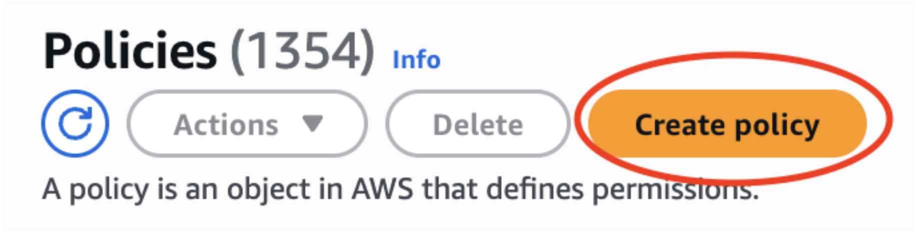
Chapter 3. Access Management Setup

1. Create a New IAM Policy

- Open AWS console and navigate to **IAM** service
- Navigate to “**Policies**” menu item under the “**Access management”** on the left menu



- Click on “**Create policy**”



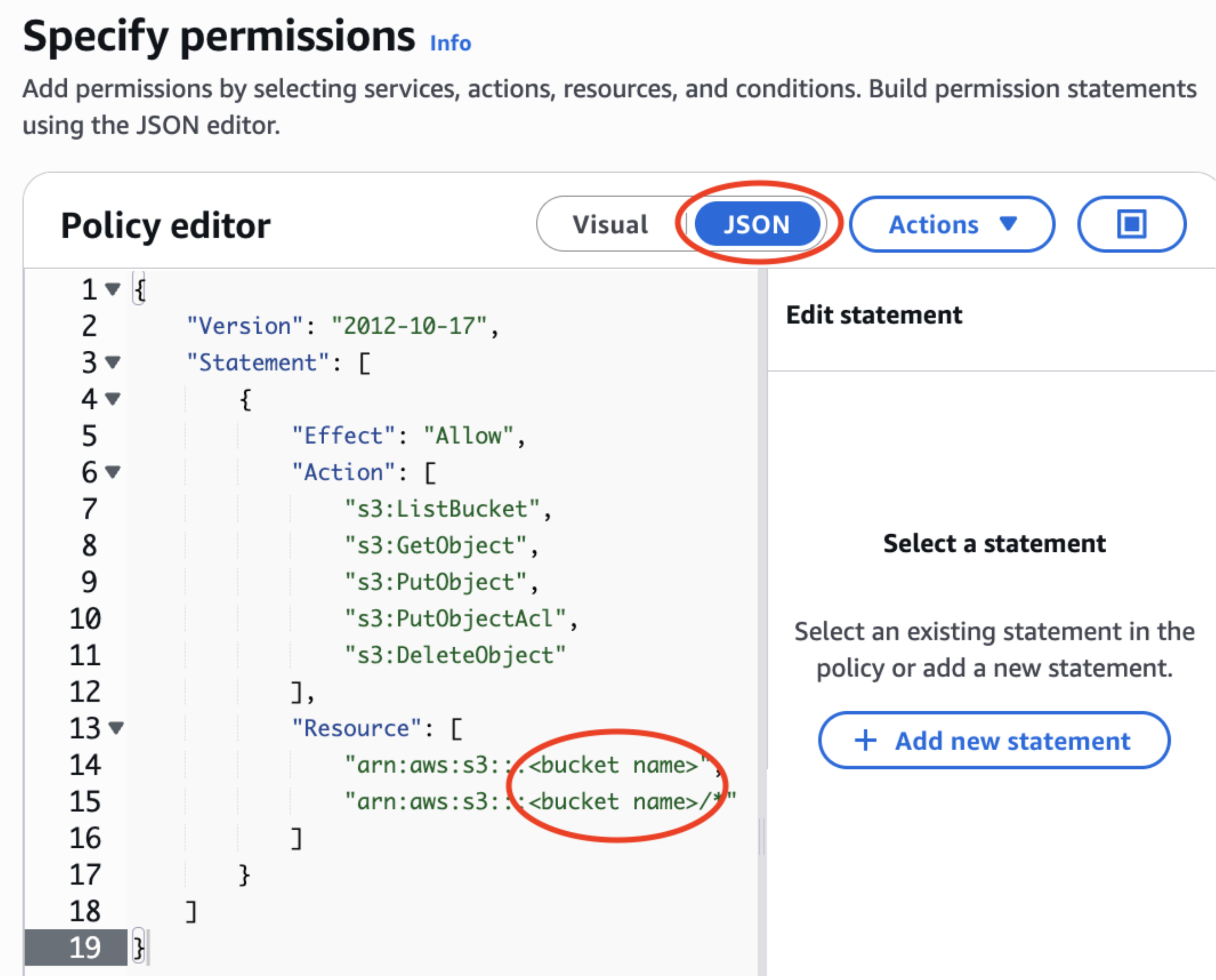
A policy is an object in AWS that defines permissions.

- In the “**Policy selector**”, select the “**JSON**” tab

Copyright © 2025 DistroScale, Inc. and/or its affiliates. All rights reserved.

- Edit statement and paste the following JSON, **ensuring you replace the bucket name with the actual name of the S3 bucket you created in Chapter 2**

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket",
                "s3:GetObject",
                "s3:PutObject",
                "s3:PutObjectAcl",
                "s3:DeleteObject"
            ],
            "Resource": [
                "arn:aws:s3:::<bucket name>",
                "arn:aws:s3:::<bucket name>/*"
            ]
        }
    ]
}
```



Specify permissions

Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.

Policy editor

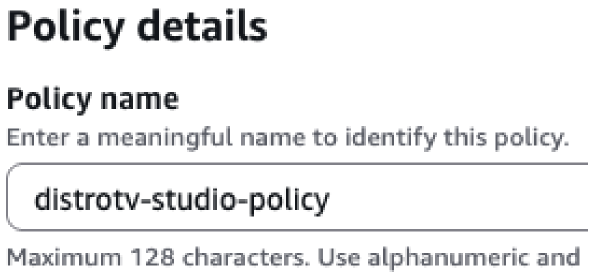
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket",
                "s3:GetObject",
                "s3:PutObject",
                "s3:PutObjectAcl",
                "s3:DeleteObject"
            ],
            "Resource": [
                "arn:aws:s3:::<bucket name>",
                "arn:aws:s3:::<bucket name>/*"
            ]
        }
    ]
}
```

Edit statement

Select a statement

Select an existing statement in the policy or add a new statement.

- Save by clicking the “**Next**” button
- Name this “**distrotv-studio-policy**”, or another name of your choice, then click “**Create policy**” button



Policy details

Policy name

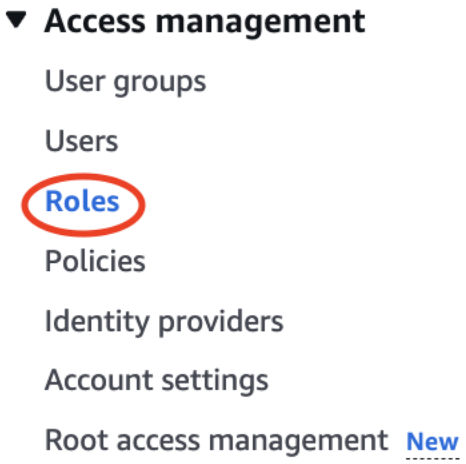
Enter a meaningful name to identify this policy.

distrotv-studio-policy

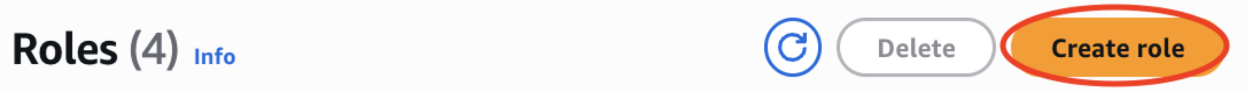
Maximum 128 characters. Use alphanumeric and

2. Create a New Role

- Open AWS console and navigate to **IAM** service
- Navigate to “**Roles**” menu item under the Access management on the left menu

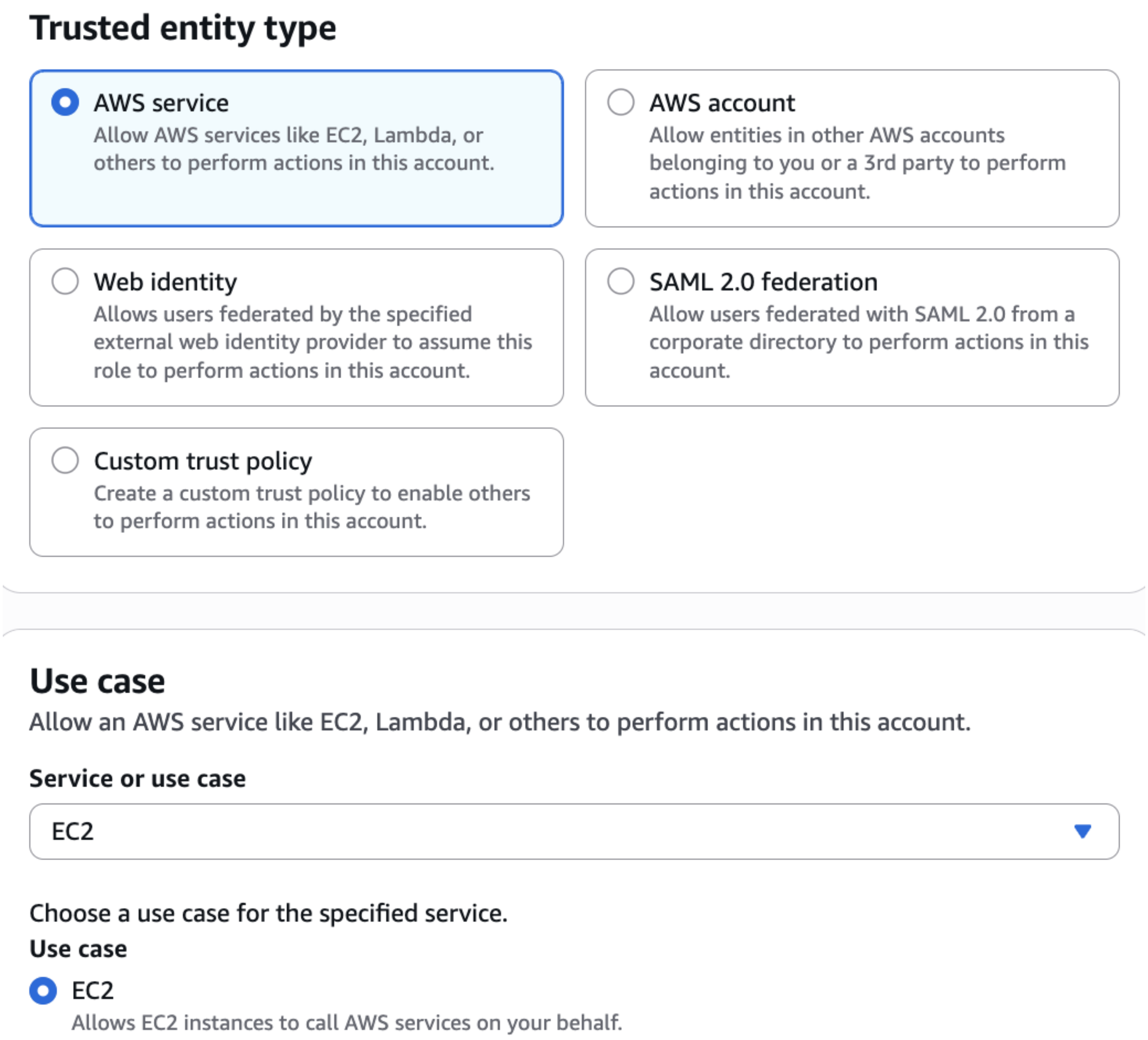


- Click on “**Create role**” button



- For “**Trusted entity type**”, ensure “**AWS service**” is selected
- Under “**Use case**”, select the “**EC2**” radio button option

Copyright © 2025 DistroScale, Inc. and/or its affiliates. All rights reserved.



Trusted entity type

● AWS service
Allow AWS services like EC2, Lambda, or others to perform actions in this account.

○ AWS account
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

○ Web identity
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

○ SAML 2.0 federation
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

○ Custom trust policy
Create a custom trust policy to enable others to perform actions in this account.

Use case

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Service or use case

EC2 ▼

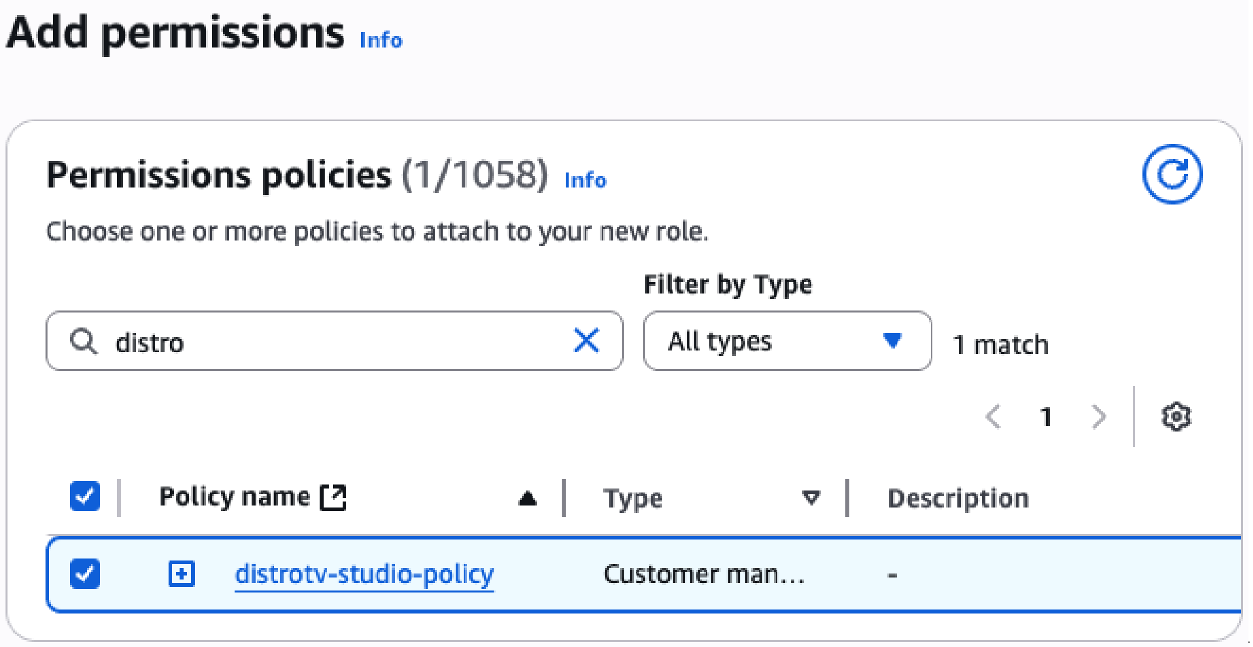
Choose a use case for the specified service.

Use case

● EC2
Allows EC2 instances to call AWS services on your behalf.

- Click on “**Next**” button
- Add permissions: In the “**Permissions policies**” search box, search for the policy you created in Step 1 (e.g., `distrotv-studio-policy` or the name you chose). Check the box next to this policy to select it.

Copyright © 2025 DistroScale, Inc. and/or its affiliates. All rights reserved.



Add permissions Info

Permissions policies (1/1058) Info

Choose one or more policies to attach to your new role.

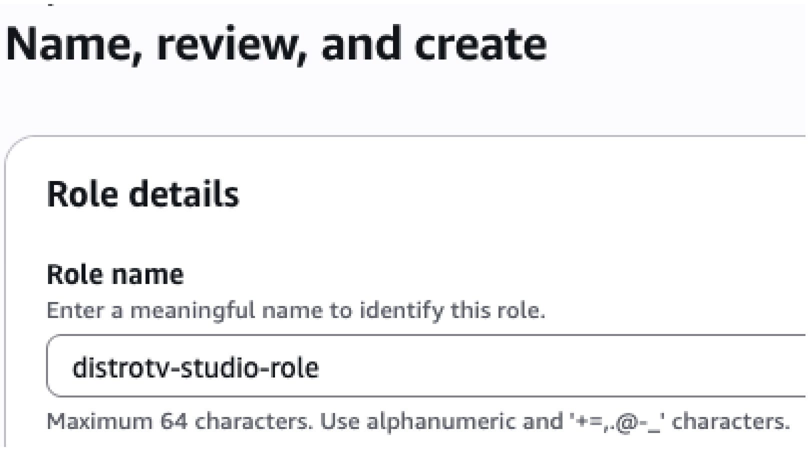
Filter by Type

distro | All types ▼ | 1 match

< 1 >

☑	Policy name	Type	Description
☑	⊞ distrotv-studio-policy	Customer man…	-

- Name, review, and create
  - Under "**Role name**" box "**Name**" section, enter '**distrotv-studio-role**' or another name of your choice
  - Description can also be left default or changed to your liking.



Name, review, and create

Role details

Role name

Enter a meaningful name to identify this role.

distrotv-studio-role

Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.

- Click on "**Create role**" button

Copyright © 2025 DistroScale, Inc. and/or its affiliates. All rights reserved.

Chapter 4. Create a Global Conf File

The global conf file is a configuration file which helps point the DistroTV Studio AMI to access the correct public S3 directories for files.

It takes 4 required parameters:
s3_meta is the folder where the channel configuration files are being placed.
s3_output is the folder where the m3u8s and ts files will be uploaded. they will be uploaded into this folder /strm/ for linear and built channels and in /vid/ for content generated by the DistroTV Studio Transcoder.
s3_transcode is the folder where the preramp will communicate with the DistroTV Studio Transcoder when new videos are to be encoded.
channel_list are the names of the configuration files the DistroTV Studio AMI will scan for within your S3 directory. In this guide we assume the name of the channel file will be myrestreamchanel.

1. Create Global Conf File

- Download the sample global conf file here and be sure to replace the <bucket name> with your bucket: https://docs.distro.tv/samples/restream/globals/global_1.conf
    - HINT: After opening the link, right-click the page and ‘Save As’ **global_1.conf**

```
[General]
s3_meta=s3://<bucket name>/meta/
s3_output=s3://<bucket name>/
s3_transcode=s3://<bucket name>/transcode/
channel_list=myrestreamchannel
```

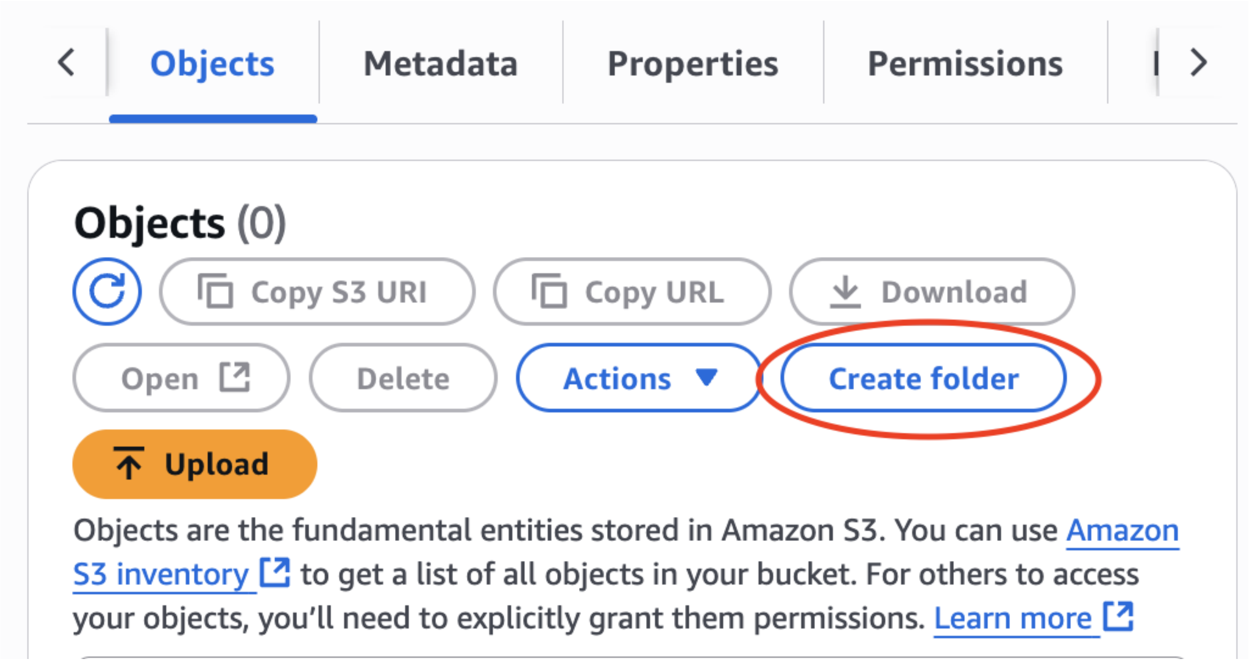
- Edit the file and be sure to replace the <bucket name> with your bucket
    - Note: the /meta/ and /transcode/ folders will be created in later steps

- Save the file as “**global_1.conf**” on your local machine. You will be uploading this file in a later step.

2. Create globals Folder

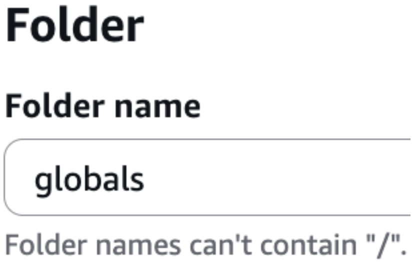
- Open AWS console and navigate to **S3** service and navigate to the bucket you created in Chapter 1 (root folder)
- Click on “**Create folder**” button

Copyright © 2025 DistroScale, Inc. and/or its affiliates. All rights reserved.



Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

- Under “**Folder**” box “**Folder name**” section, name this “**globals**”



Folder

Folder name

globals

Folder names can't contain "/".

- Create a **globals** folder inside the path defined in the **s3_output** folder.
  - If you kept the default filepath from the sample global conf file, the file directory path would be s3://<bucket name>/globals/

3. Upload Your Global Conf File

- Upload the global conf file you just created to your S3 bucket /**globals**/ folder.

Copyright © 2025 DistroScale, Inc. and/or its affiliates. All rights reserved.

Chapter 5. Creating S3 Bucket Directories

- In your S3 bucket, ensure that the folders & directories listed in your global conf file exist
  - **s3_meta**=s3://<bucket name>/meta/
  - **s3_output**=s3://<bucket name>/
  - **s3_transcode**=s3://<bucket name>/transcode/
  - If they do not exist, create them and name them according to how you defined the paths in your global conf file by following the steps below

1. Create **meta** Folder

- Navigate to your S3 bucket root folder
- Click on "**Create folder**" button
- Under "**Folder**" box "**Folder name**" section, name this "**meta**"
  - If you kept the default filepath from the sample global conf file, the file directory path would be **s3://<bucket name>/meta**
- Click on "**Create folder**" button

2. Create **transcode** Folder

- Navigate to your S3 bucket root folder
- Click on "**Create folder**" button
- Under "**Folder**" box "**Folder name**" section, name this "**transcode**"
  - If you kept the default filepath from the sample global conf file, the file directory path would be **s3://<bucket name>/transcode**
- Click on "**Create folder**" button

3. Create **content** Folder

- Navigate to your S3 bucket root folder
- Click on "**Create folder**" button
- Under "**Folder**" box "**Folder name**" section, name this "**content**"
  - If you kept the default filepath from the sample global conf file, the file directory path would be **s3://<bucket name>/content/**
- Click on "**Create folder**" button

4. Create **channels** Folder

- Navigate to your S3 bucket **meta** folder
- Click on "**Create folder**" button

Copyright © 2025 DistroScale, Inc. and/or its affiliates. All rights reserved.

- Under “**Folder**” box “**Folder name**” section, name this “**channels**”
    - If you kept the default filepath from the sample global conf file, the file directory path would be **s3://<bucket name>/meta/channels/**
- Click on “**Create folder**” button

5. Create **schedule** Folder

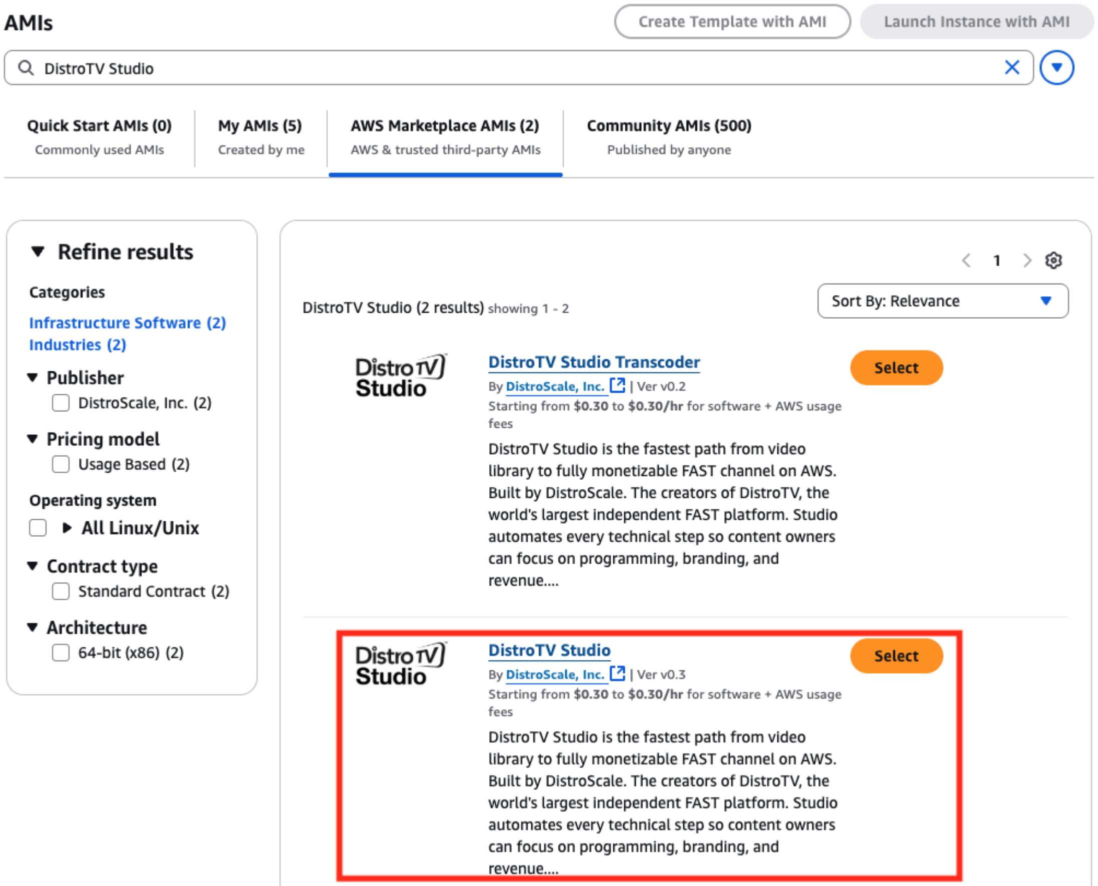
- Navigate to your S3 bucket **meta** folder
- Click on “**Create folder**” button
- Under “**Folder**” box “**Folder name**” section, name this “**schedule**”
    - If you kept the default filepath from the sample global conf file, the file directory path would be **s3://<bucket name>/meta/schedule/**
- Click on “**Create folder**” button

After this chapter, your S3 bucket should have the following structure:

```
<bucket-name>/
├── content/                           (Created in Chapter 5)
│
├── globals/                           (Created in Chapter 4)
│   └── global_1.conf                  (Uploaded in Chapter 4)
│
├── meta/                              (Created in Chapter 5)
│   ├── channels/                      (Created in Chapter 5)
│   └── schedule/                      (Created in Chapter 5)
│
└── transcode/                         (Created in Chapter 5)
```

Chapter 6. Launch a DistroTV Studio Image

- Navigate to **AWS EC2** service
- Navigate to “**AMI Catalog**” menu item under the “**Images”** section on the left menu
- Navigate to “**AWS Marketplace AMIs**” and search for “**DistroTV Studio**”



DistroTV Studio

DistroTV Studio (2 results) showing 1 - 2

DistroTV Studio Transcoder

By DistroScale, Inc. | Ver v0.2

Starting from $0.30 to $0.30/hr for software + AWS usage fees

DistroTV Studio is the fastest path from video library to fully monetizable FAST channel on AWS. Built by DistroScale. The creators of DistroTV, the world's largest independent FAST platform. Studio automates every technical step so content owners can focus on programming, branding, and revenue....

DistroTV Studio

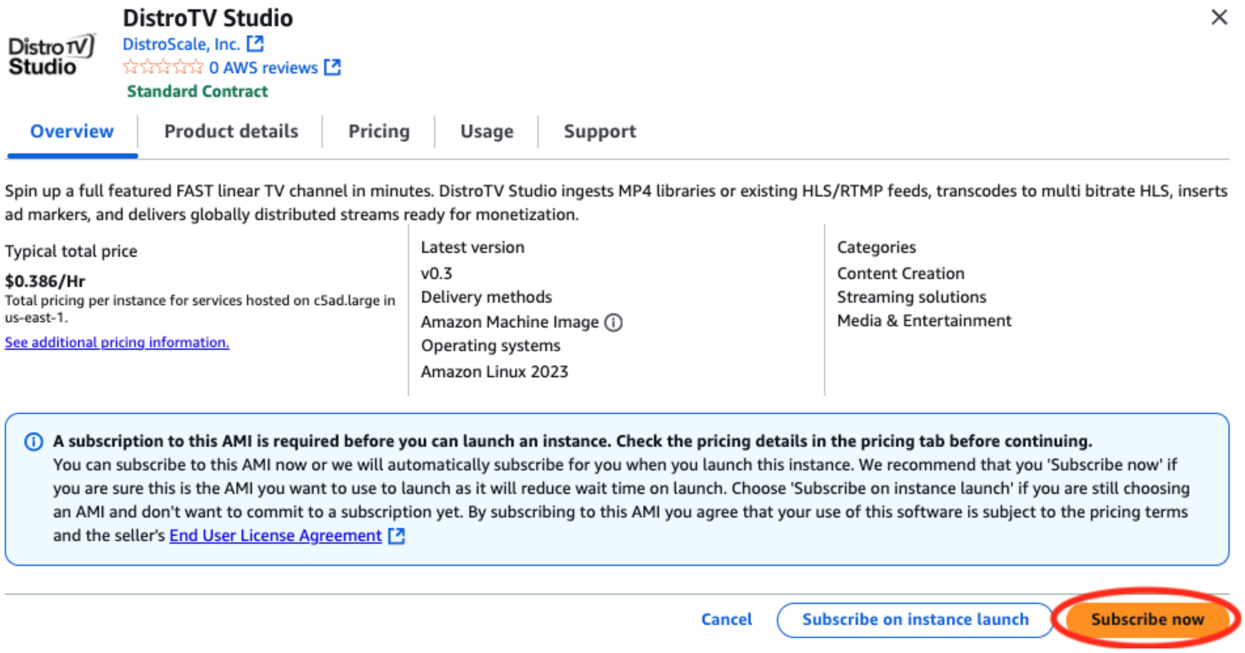
By DistroScale, Inc. | Ver v0.3

Starting from $0.30 to $0.30/hr for software + AWS usage fees

DistroTV Studio is the fastest path from video library to fully monetizable FAST channel on AWS. Built by DistroScale. The creators of DistroTV, the world's largest independent FAST platform. Studio automates every technical step so content owners can focus on programming, branding, and revenue....

- Select the **DistroTV Studio** image and click on “**Subscribe now**”

Copyright © 2025 DistroScale, Inc. and/or its affiliates. All rights reserved.



DistroTV Studio

DistroScale, Inc.

☆☆☆☆☆ 0 AWS reviews

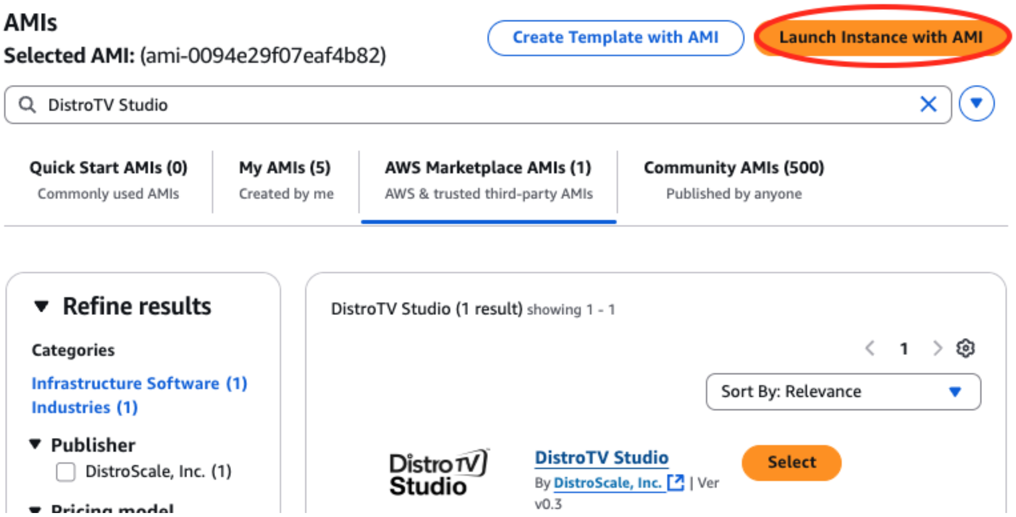
Standard Contract

Spin up a full featured FAST linear TV channel in minutes. DistroTV Studio ingests MP4 libraries or existing HLS/RTMP feeds, transcodes to multi bitrate HLS, inserts ad markers, and delivers globally distributed streams ready for monetization.

Typical total price	Latest version	Categories
$0.386/Hr Total pricing per instance for services hosted on c5ad.large in us-east-1. See additional pricing information.	v0.3 Delivery methods Amazon Machine Image Operating systems Amazon Linux 2023	Content Creation Streaming solutions Media & Entertainment

A subscription to this AMI is required before you can launch an instance. Check the pricing details in the pricing tab before continuing. You can subscribe to this AMI now or we will automatically subscribe for you when you launch this instance. We recommend that you 'Subscribe now' if you are sure this is the AMI you want to use to launch as it will reduce wait time on launch. Choose 'Subscribe on instance launch' if you are still choosing an AMI and don't want to commit to a subscription yet. By subscribing to this AMI you agree that your use of this software is subject to the pricing terms and the seller's End User License Agreement

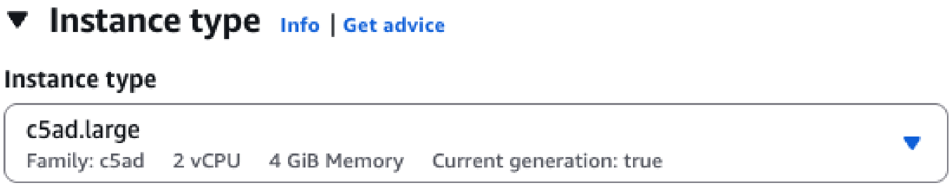
- After subscribing, you click on “**Launch Instance with AMI**”



AMIs

Selected AMI: (ami-0094e29f07eaf4b82)

  - Name – to your liking
  - Instance type
    - **c5ad.large, c5ad.xlarge, or c5ad.2xlarge**



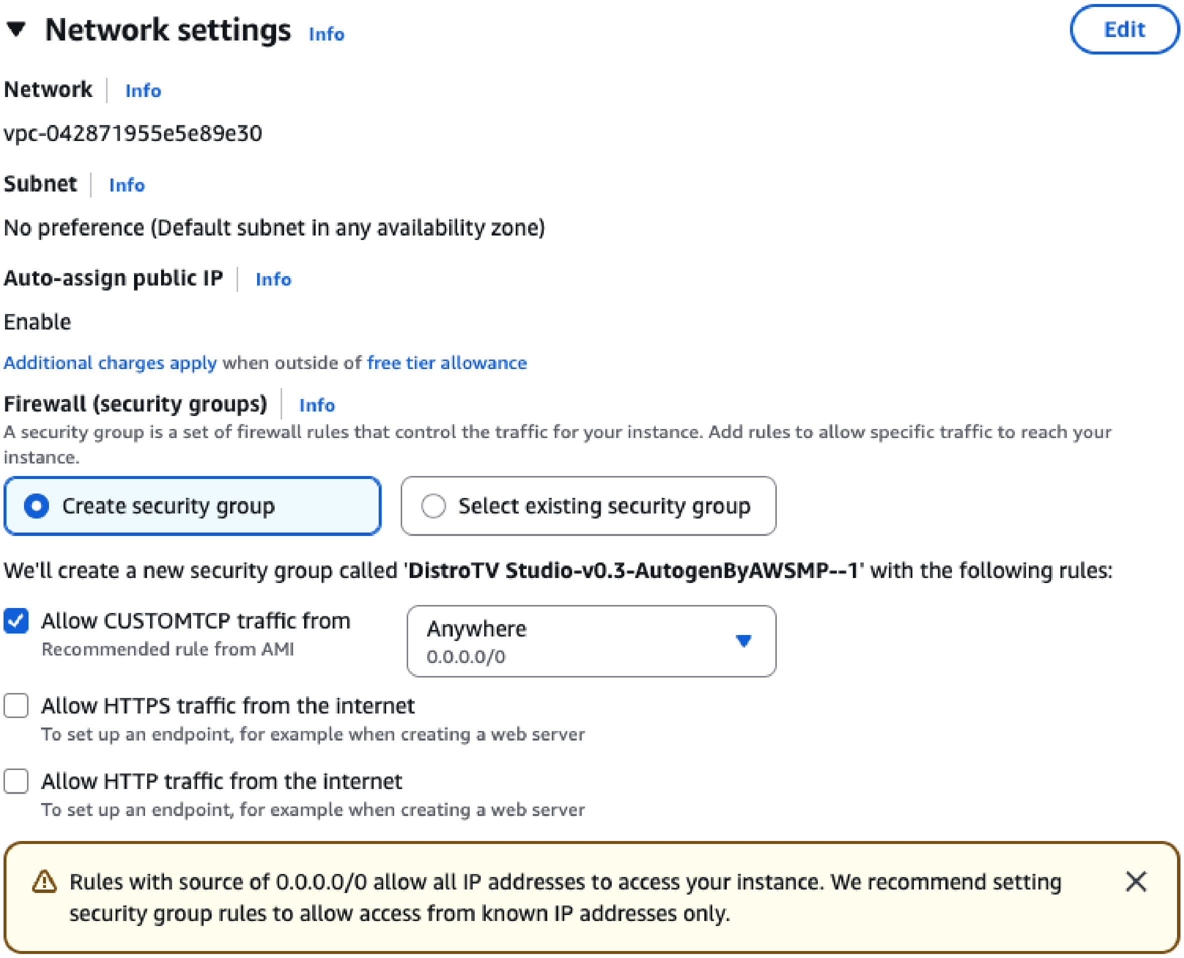
Instance type Info | Get advice

Instance type: c5ad.large — Family: c5ad 2 vCPU 4 GiB Memory Current generation: true

  - Key pair
    - Choose your preferred key pair
  - Network settings
    - Check the following:

Copyright © 2025 DistroScale, Inc. and/or its affiliates. All rights reserved.

        - Allow CUSTOMTCP traffic from (Anywhere)



▼ **Network settings** Info

Network | Info

vpc-042871955e5e89e30

Subnet | Info

No preference (Default subnet in any availability zone)

Auto-assign public IP | Info

Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) | Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

● Create security group ○ Select existing security group

We'll create a new security group called '**DistroTV Studio-v0.3-AutogenByAWSMP--1**' with the following rules:

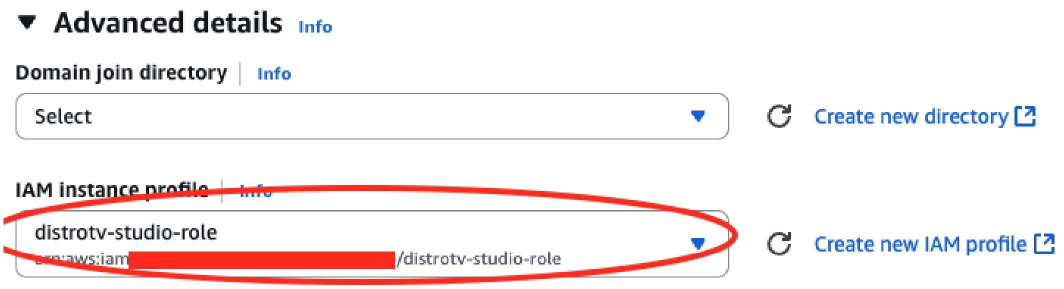
☑ Allow CUSTOMTCP traffic from — Anywhere 0.0.0.0/0
Recommended rule from AMI

☐ Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

☐ Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

- Under the "**Advanced details**" box
    - IAM instance profile
        - Select the role you created earlier (**distrotv-studio-role** if you followed the naming convention in the Chapter 2)



▼ **Advanced details** Info

Domain join directory | Info

Select

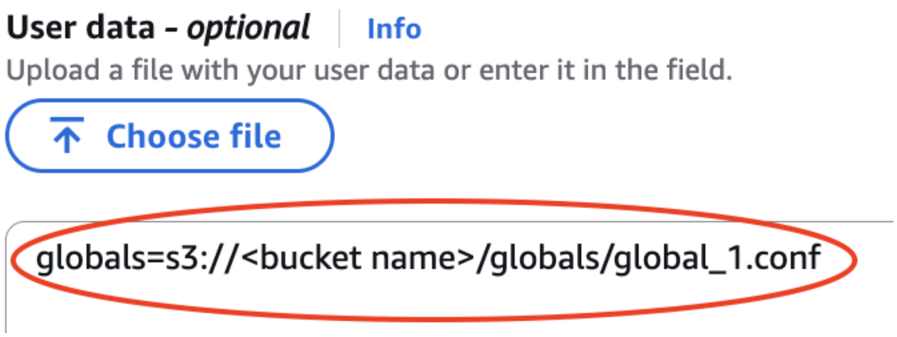
IAM instance profile | Info

distrotv-studio-role
arn:aws:iam[redacted]/distrotv-studio-role

    - "User data - optional" (near the bottom of the options)
        - Add the following line being sure to reference your S3 file path or publicly accessible URL that stores the global conf file(s)

```
globals=s3://<bucket name>/globals/global_1.conf
```

Copyright © 2025 DistroScale, Inc. and/or its affiliates. All rights reserved.



Upload a file with your user data or enter it in the field.

```
globals=s3://<bucket name>/globals/global_1.conf
```

        - Click on “**Launch instance**” button
- After waiting 2-3 minutes, validate the machine is up and running by connecting to the API
    - Open a browser and go to your EC2 instance’s API log, being sure to replace <your-instance-public-IPv4> with your newly launched EC2 instance’s Public IPv4 address: http://<your-instance-public-IPv4>:34123/?version

Copyright © 2025 DistroScale, Inc. and/or its affiliates. All rights reserved.

Chapter 7. Launch Your Channel

- Create a channel conf file by downloading the sample channel conf file below here being sure to replace **<your-existing-stream-m3u8>** with an existing HLS stream master m3u8 playlist: https://docs.distro.tv/samples/restream/meta/channels/myrestreamchannel.conf

```
[General]
stream=<your-existing-stream-m3u8>
buffer=10
cdd=1
uses3=1
awsparams=--acl public-read --recursive --cache-control max-age=3
download=1
```

- Save as **myrestreamchannel.conf**
- This channel conf file controls different channel parameters such as how much to buffer the original stream, enabling ad marker insertions, recording dynamic ad break durations, and more. You can find additional details in our **User Guide**.
- Inside the **/meta/channels/** folder, upload the channel conf file you created **myrestreamchannel.conf**
- After uploading, you can wait a few minutes to see if the channel was picked up by the machine by checking the watchdog API at: http://<your-instance-public-IPv4>:34123/?cmd=getlog&channel=watchdog&grep=
- You can also check to see if files have begun to populate in your **s3_output** folder. If you kept the default filepath from the sample global conf file, the file directory path would be **s3://<bucket name>/strm/channels/myrestreamchannel/**
- When the re-stream is active, you will be able to view your channel at https://**<bucket name>**.s3**.<region>**.amazonaws.com/strm/channels/myrestreamchannel/master.m3u8
    - Replace **<bucket name>** with your bucket name
    - Replace **<region>** with your S3 bucket region
- Congratulations on launching your first channel!

Copyright © 2025 DistroScale, Inc. and/or its affiliates. All rights reserved.

After all chapters, your S3 bucket should have the following structure:

```
<bucket-name>/
├── content/                        (Created as per Chapter 5)
|
├── globals/                        (Created as per Chapter 4)
|   └── global_1.conf               (Uploaded as per Chapter 4, referenced by EC2 instances)
|
├── meta/                           (Created as per Chapter 5)
|   ├── channels/                   (Created as per Chapter 5)
|   |   ├── myrestreamchannel.conf     (Uploaded as per Chapter 7)
|
├── strm/                           (System generated, implied by Chapter 7)
|   └── channels/                   (System generated, implied by Chapter 7)
|       └── myrestreamchannel/         (System generated, implied by Chapter 7)
|           ├── master.m3u8         (System generated, playback URL target in Chapter 7)
|           └── *.ts                (Video segment files, system generated)
|
└── transcode/                      (Created as per Chapter 5)
```

Copyright © 2025 DistroScale, Inc. and/or its affiliates. All rights reserved.