Delay-Flow SCTE-35 Automated Insertion

Deployment QuickStart Guide

This guide will help you quickly deploy Delay-Flow to test video delay, matching Ad-Flow's delay, using the AWS CloudFormation service. This will deploy the container with default options (input is RTP, at port 5004).

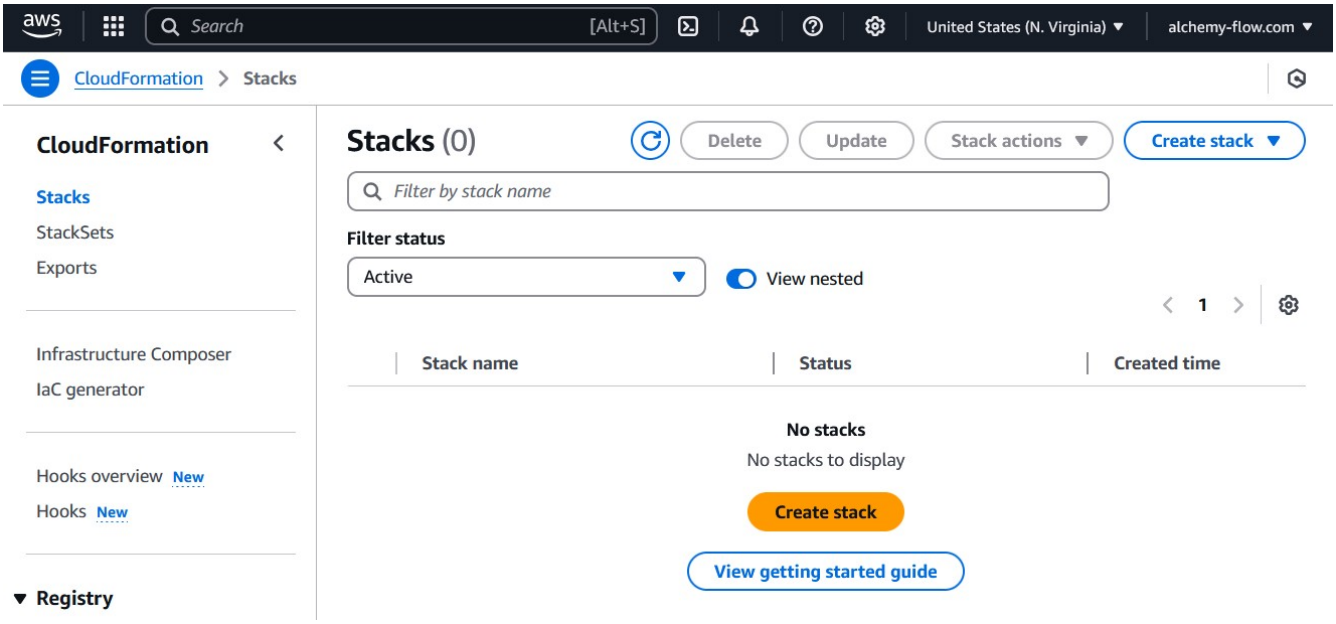
Prerequisites

- An AWS account with permissions to:
    - Create CloudFormation stacks
    - Create EC2 instances
    - Create ECS clusters
    - Create IAM roles
- A VPC with a public subnet
- An EC2 key pair (for SSH access if needed)
- Download the two CloudFormation templates for Delay-Flow:
  https://alcflow-config.s3.us-east-2.amazonaws.com/1.0/CloudFormation/ecsClusterTemplate-delayFlow.json
  https://alcflow-config.s3.us-east-2.amazonaws.com/1.0/CloudFormation/ecsTaskTemplate-delayFlow.json

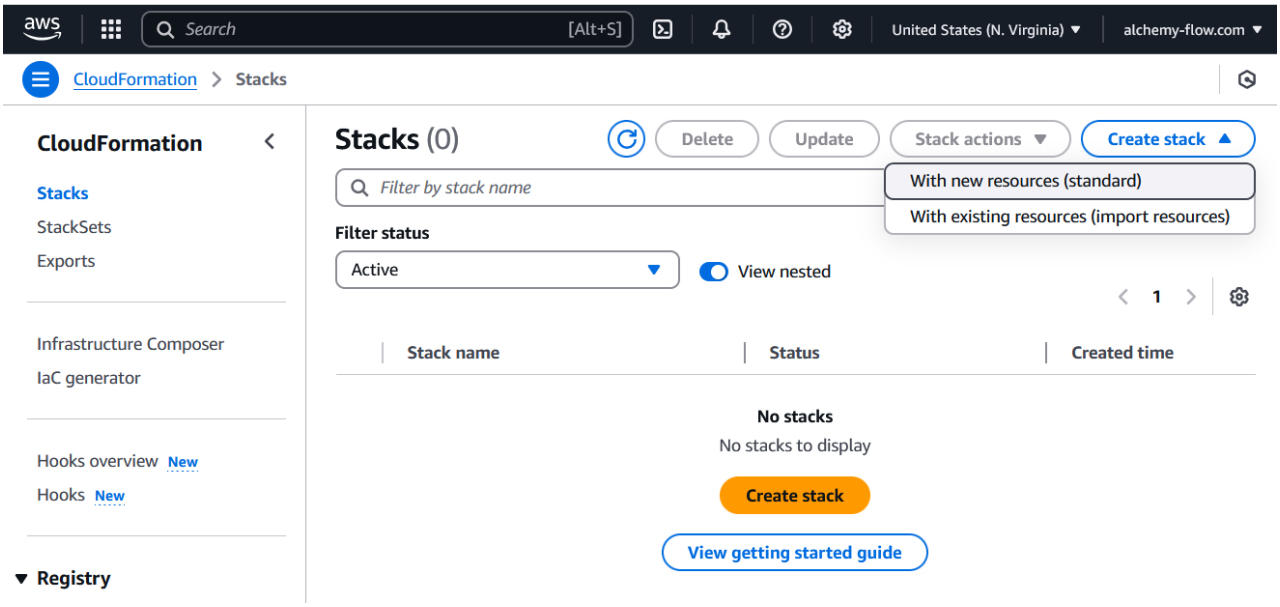
Deployment Steps

Step 1: Deploy the Cluster Template

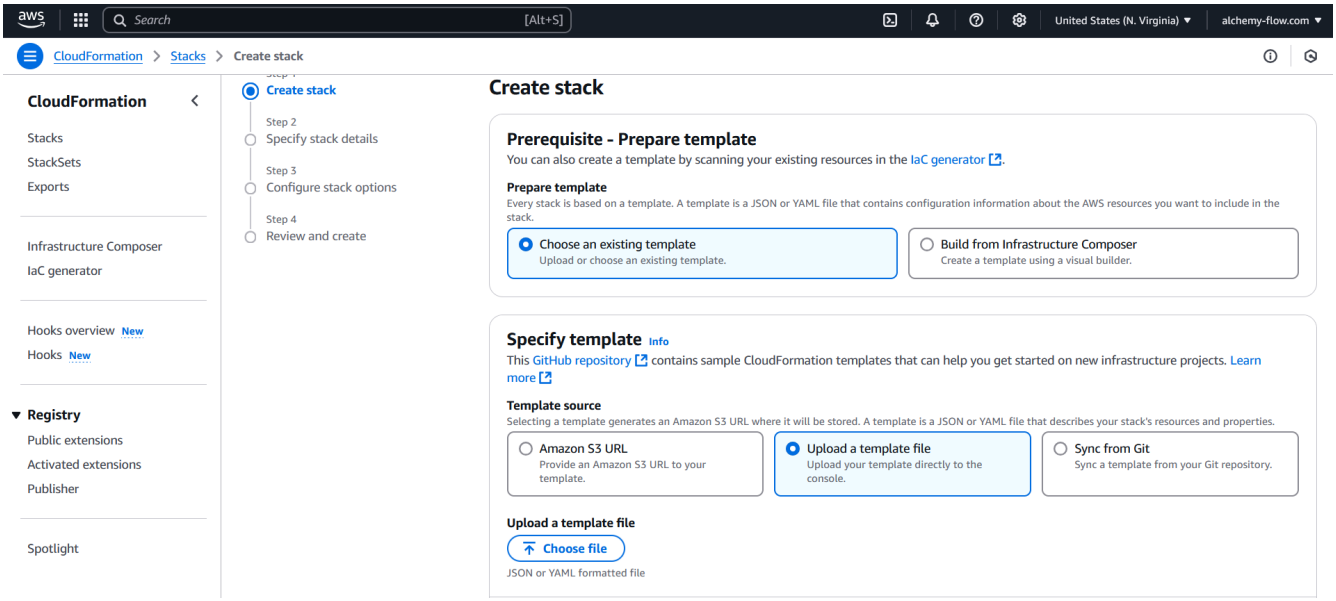
1. Log into the AWS Console and navigate to CloudFormation



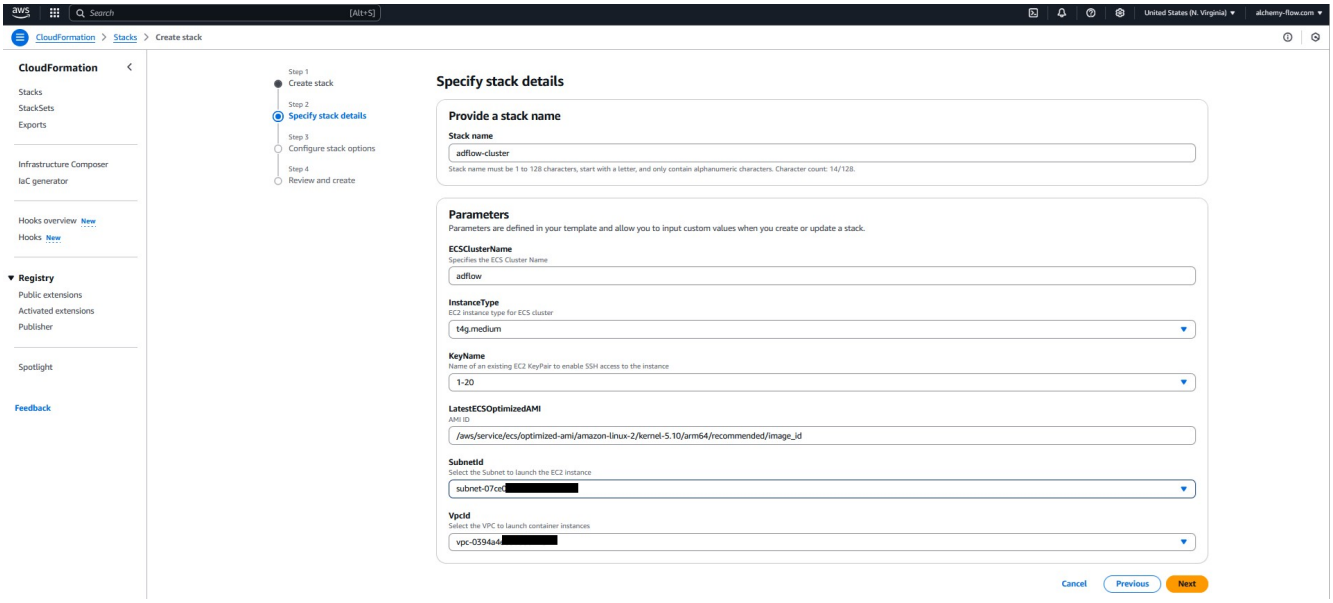
2. Click "Create stack" and select "With new resources (standard)"



3. On the "Create stack" page:
    - Select "Upload a template file"
    - Click "Choose file" and select the `ecsClusterTemplate-delayFlow.json` file
    - Click "Next"



4. Enter stack details:
    - Stack name: delay`flow-cluster` (or your preferred name)
    - Parameters:
        - KeyName: Select your EC2 key pair
        - ECSClusterName: Leave as default (delay`flow`)
        - InstanceType: Leave as default (`t4g.medium`)
        - VpcId: Select your VPC
        - SubnetId: Select a public subnet
    - Click "Next"



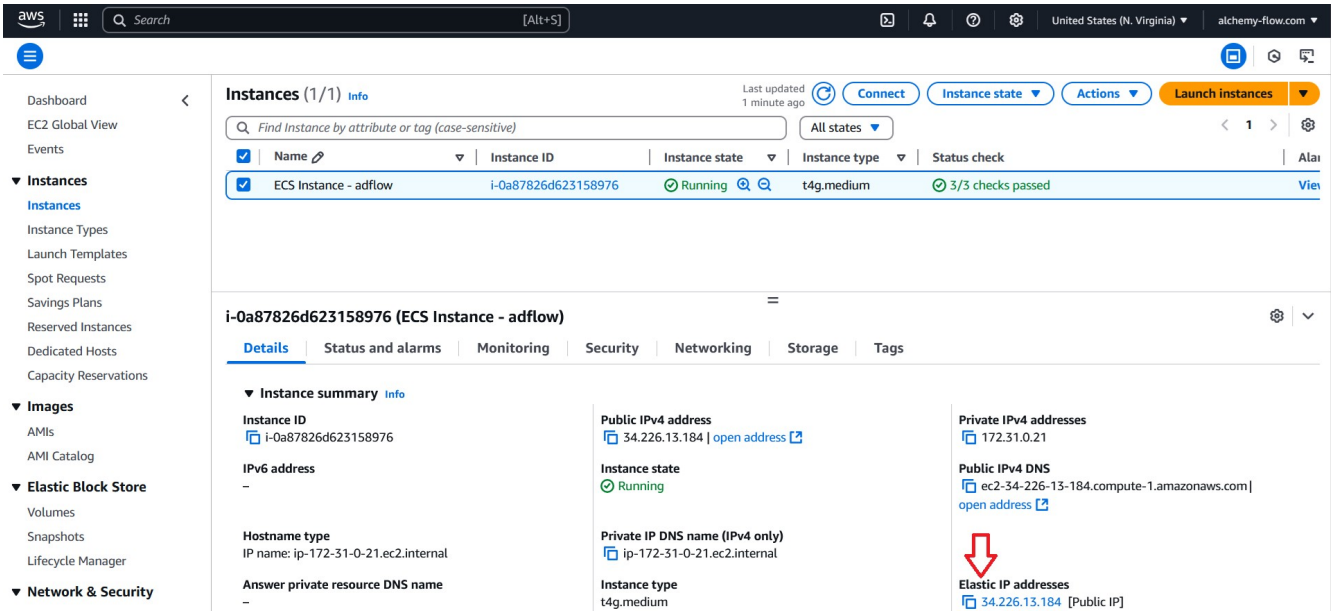
5. On the "Configure stack options" page:
    - Leave all settings at their defaults
    - Click "Next"
6. Review the configuration:
    - Check the acknowledgment for IAM resource creation
    - Click "Create stack"
7. Wait for the stack creation to complete (approximately 5 minutes)
    - The status will change to "CREATE_COMPLETE" when finished

Step 2: Deploy the Task Template

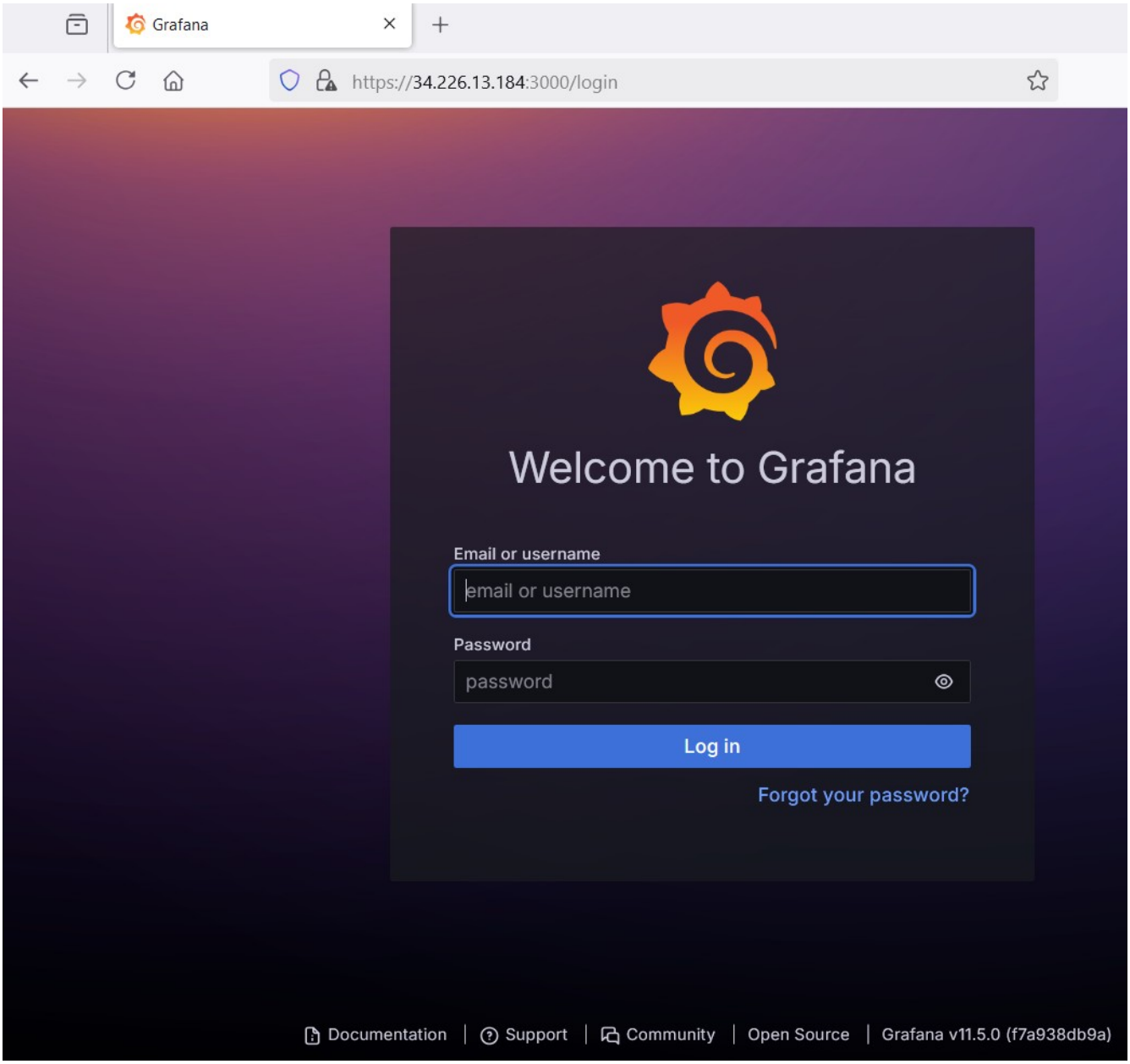
1. In CloudFormation, click "Create stack" again
2. On the "Create stack" page:
    - Select "Upload a template file"
    - Click "Choose file" and select the `ecsTaskTemplate-delayFlow.json` file
    - Click "Next"
3. Enter stack details:
    - Stack name: `delayflow-task` (or your preferred name)
    - Parameters:
        - ECSClusterName: Enter the same name used in Step 1 (`delayflow`)
    - Click "Next"
4. On the "Configure stack options" page:
    - Leave all settings at their defaults
    - Click "Next"
5. Review and create:
    - Check the acknowledgment for IAM resource creation
    - Click "Create stack"
6. Wait for the stack creation to complete (approximately 5 minutes)

Step 3: Access Grafana Dashboard

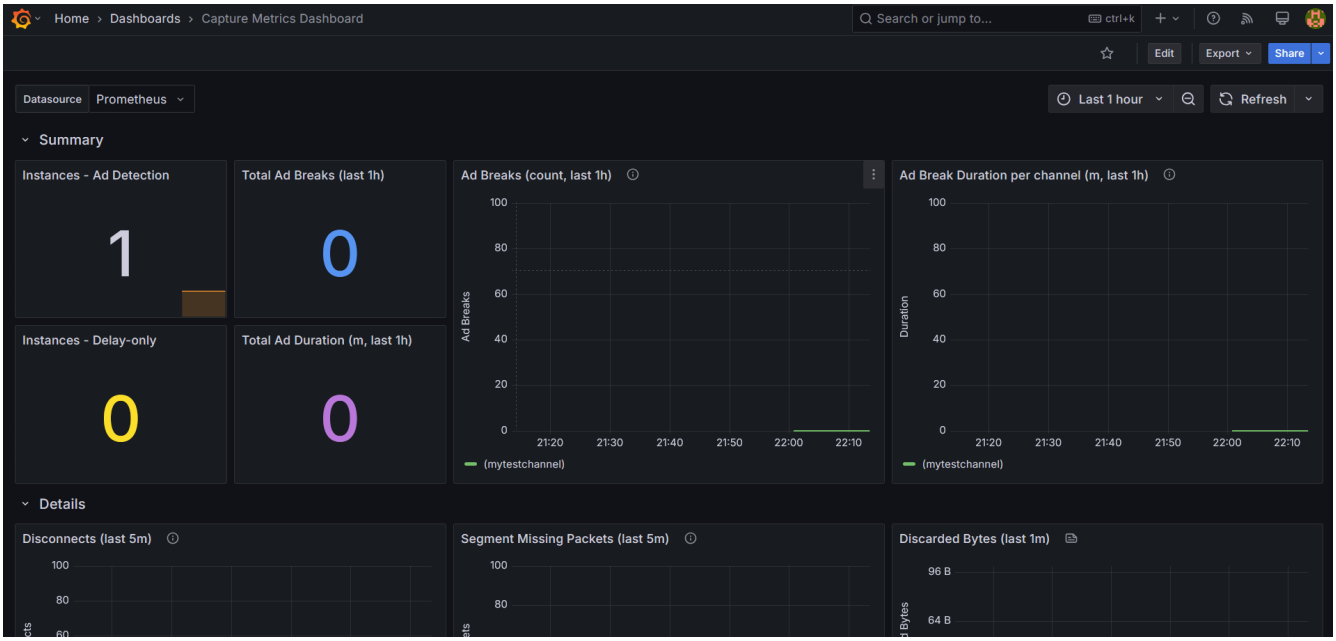
1. Find your instance's public address (Elastic IP):
    - Go to EC2 service in AWS Console
    - Select the instance named "ECS Instance - delayflow"
    - Copy the "Elastic IP addresses [public IP]" value



2. Access Grafana:
    - Open your browser
    - Navigate to `https://[Public IPv4 DNS]:3000`
    - Accept the security warning about the self-signed certificate
    - Log in with:
        - Username: `admin`
        - Password: `admin`
    - Change the password when prompted



3. View the Delay-Flow dashboard:
    - Click on "Dashboards" in the left menu
    - Select "Capture Metrics Dashboard"



Testing Your Deployment

Option 1: Using the Test Stream AMI

1. Deploy the "RTP/SRT Streamer for Ad-Flow" AMI:
    - Search for "RTP/SRT Streamer for Ad-Flow" in AWS Marketplace
    - Follow the AMI's documentation for setup
    - Configure it to stream to your Delay-Flow instance's IP address on port 5004

Option 2: Using Your Own RTP Source

Configure your RTP source to stream to:

- Destination: Your Ad-Flow instance's Elastic IP
- Port: 5004
- Protocol: UDP/RTP

Verifying Operation

1. In the Grafana dashboard, check:
    - Input transfer rate (should show data flowing)
    - Ad break detection count. Please note that unlike Ad-Flow, the Delay-Flow container will not detect any ads. Its output is only meant to be used as a backup input stream alongside with Ad-Flow’s output.
    - Stream health metrics
2. Common issues:
    - No data in dashboard: Check security group allows UDP port 5004

- Can't access Grafana: Verify port 3000 is allowed in security group
- Certificate warning: Expected with self-signed certificate

Next Steps

Once you've verified ad break detection is working:

1. Review the full documentation for production deployment options
2. Configure output streaming to your video encoder’s secondary output

Support

For issues or questions:

- Check CloudWatch logs for errors
- Review the full documentation
- Contact AWS Marketplace support, or Alchemy-Flow support