aws

User Guide

Conductor Live and Elemental Statmux

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Conductor Live and Elemental Statmux: User Guide

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

About the Conductor Live solution 1
Working with Conductor Live 1
General information 2
Software versions 3
Licenses 3
Centralized management 3
Redundancy and failover 4
Working with Elemental Live 5
Components of Elemental Live 6
Comparison of profiles 7
Working with AWS Elemental Statmux 8
Components of Elemental Statmux 8
Features of AWS Elemental Statmux 9
AWS Elemental Statmux tutorial 11
Assumptions about existing knowledge 11
Step 1: Check your cluster and redundant nodes 11
Step 2: Create the profiles 13
Step 3: Create the SPTS channels 14
Step 4: Create the MPTS 15
Step 5: Add programs to the MPTS 16
Step 6: Start the MPTS 17
Rules and limits 19
Display the Conductor Live web interface 20
Displaying the web interface 20
Viewing Conductor Live configuration information 21
Setup: Design the cluster 22
Conductor Live nodes 22
Primary Elemental Live nodes 22
Primary Elemental Statmux nodes 25
Setup: Plan resiliency 26
Conductor Live node redundancy 26
Worker node redundancy 27
N-to-M redundancy 28
1-to-1 redundancy 30

1-to-1 Plus redundancy ... 31
Organizing groups in the cluster ... 32
Resiliency in Elemental Statmux ... 33
Output listening in Elemental Statmux ... 33
Output redundancy in Elemental Statmux ... 34
Output listening combined with output redundancy ... 35
Workflows: Design an MPTS workflow ... 37
Create a standard MPTS ... 37
Step 1: Create the profiles and SPTS channels ... 38
Step 2: Create the MPTS and add channels ... 40
Handling by Elemental Statmux ... 40
Including passthrough programs ... 41
Passing through custom streams ... 42
Passing through SI/PSI tables ... 43
Daily operations: Run workflows ... 45
Start and stopping a channel ... 45
Starting and stopping one channel ... 45
Starting or stopping several channels at one time ... 45
Scheduling channels ... 46
The procedure ... 47
View active schedules ... 47
Controlling channels ... 47
Starting or stopping an MPTS ... 48
Daily operations: Monitor workflow activity ... 49
Monitoring channels ... 49
Monitoring the health of channels ... 49
Monitoring channel activity at the node ... 50
Viewing channel history ... 50
Monitoring MPTS outputs ... 50
Monitoring the progress of all MPTSes ... 50
Monitoring the muxing performance of an MPTS ... 51
Modifying the MPTS while it is running ... 51
Monitoring alerts and messages ... 51
About alerts and messages ... 51
Alerts and messages on the web interface ... 53
Monitoring nodes ... 55

Offline nodes ... 56
Failed worker nodes with worker redundancy ... 56
Failed worker nodes without worker redundancy ... 59
Failed Conductor Live nodes with Conductor Live redundancy ... 60
Failed Conductor Live nodes without Conductor Live redundancy ... 60
Monitoring the load on worker nodes ... 61
Resources: Profiles ... 62
Create profile from scratch ... 62
Design the profile ... 62
Create the profile ... 63
Work with channel parameters ... 64
Rules for channel parameters ... 65
The procedure ... 65
Plan for bulk changes ... 66
Use case: SDI direct input ... 67
Use case: SDI router input ... 68
Create profile for SPTS channel ... 68
Create profile by duplicating ... 70
Create profile using XML ... 70
Modify profile ... 71
View profile contents ... 71
Delete profile ... 71
Resource: Channels ... 73
Create channel ... 73
Create a channel by duplicating ... 74
Modify channel ... 75
Change profile for multiple channels ... 75
Step 1. Create the task ... 75
Step B. Monitor the status of the task ... 78
Delete channel ... 79
Deleting one channel ... 79
Deleting several channels at once ... 79
Resource: MPTS ... 80
Create MPTS ... 80
Step 1. Create the MPTS ... 80
Step 2. Add channels ... 82

Advanced tab – Suppressing generation of SI/PSI tables 86
Including passthrough streams 87
Including passthrough programs 89
Modify MPTS 91
Delete MPTS 92
Resources: The tasks feature 93
Reference: Supported channel parameters 94
Document History 95

About the Conductor Live solution

AWS Elemental Conductor Live lets you create and manage channels on AWS Elemental Live and/or MPTSes on AWS Elemental Statmux.

Each of the three products — AWS Elemental Conductor Live, AWS Elemental Live and AWS Elemental Statmux — runs on its own node. Conductor Live is a *management node*. Elemental Live and Elemental Statmux node are each *worker nodes*. And all the nodes are organized in a *cluster.*

A cluster contains at least one Conductor Live node and one Elemental Live node. If you want to produce MPTSes, a cluster contains at least one Conductor Live node, one Elemental Live node, and one Elemental Statmux node.

Working with Conductor Live

AWS Elemental Conductor Live lets you create and manage channels on AWS Elemental Live and/or MPTSes on AWS Elemental Statmux. Each of the three products—Conductor Live, Elemental Live and Elemental Statmux—runs on its own node. Conductor Live is a *management node*. Elemental Live and Elemental Statmux node are each *worker nodes*.

All the nodes are organized in a *cluster.*

Broadly speaking, there are two ways to work with the Conductor Live suite of products:

- Create and run events (which are called channels in Conductor Live). You use Conductor Live to create the channel on an Elemental Live node that is in the cluster. You use Conductor Live to start and control the channel.

  For this scenario, the cluster includes Conductor Live and Elemental Live nodes.
- Create and run MPTSes. You use Conductor Live to create an MPTS on an Elemental Statmux node that is in the cluster. You use the Conductor Live to add Elemental Live channels to the MPTS. You use Conductor Live to start and control the MPTS.

A cluster contains at least one Conductor Live node and one Elemental Live node. If you want to produce MPTSes, a cluster contains at least one Conductor Live node, one Elemental Live node, and one Elemental Statmux node.

Topics

- General information
- Software versions
- Licenses
- Centralized management
- Redundancy and failover

General information

Assumptions

We assume that you are already familiar with Elemental Live. You know how to create an event using Elemental Live, and you are familiar with the various features of Elemental Live.

We don't make this assumption with AWS Elemental Statmux. Therefore, this guide explains everything you need to know to create and manage MPTSes using AWS Elemental Statmux. However, we do assume that you are familiar with SPTSes (single program transport streams) and MPTSes (multiple program transport streams). You should be familiar with the business uses of SPTS and MPTS, and you should be familiar with the SI/PSI tables used in these streams.

We also assume that you are know how to log into a Linux session, in order to work via the command line interface.

Supported browsers

Most browsers that are newer than two years old will support all the features of the AWS Elemental Conductor Live web interface. Supported browsers include the current versions of Firefox and Chrome. We recommend Firefox.

JavaScript Warning

JavaScript is required for the Conductor Live web interface. If JavaScript is disabled, the web interface presents a warning and provides general guidance to resolve the error.

Related documentation

For additional information about the configuration and use of Conductor Live and Elemental Live, see the following:

- AWS Elemental Conductor Live API Reference

- AWS Elemental Conductor Live Configuration Guide
- AWS Elemental Live API Reference
- AWS Elemental Live User Guide

Software versions

This guide applies to version 2.20.3 and later of Elemental Live and Elemental Statmux, and to version 3.20.3 and later of Conductor Live.

The AWS Elemental implementation of a statmux solution was relaunched in these versions. The earlier implementation is no longer supported.

Licenses

- You need the Statmux rate control add-on package for each Elemental Live node where you will produce SPTS programs that will go into the MPTS. For more information, see the section about add-on packages in the AWS Elemental Live user guide.
- You don't need an add-on package for Elemental Statmux. The general license for Elemental Statmux is sufficient.
- You don't need an add-on package for Conductor Live. The general license for Conductor Live is sufficient.

Centralized management

Conductor Live lets you control both Elemental Live nodes and AWS Elemental Statmux nodes. These workers nodes must be in a Conductor Live cluster.

With Conductor Live, you do not work on each individual Elemental Live or AWS Elemental Statmux node. Instead, you work on all nodes from one centralized web interface on the Conductor Live.

Conductor Live and Live Nodes

Centralization via Conductor Live has several advantages for encoding work:

- You create and run events (referred to as *channels*) from Conductor Live, specifying which node the channel is to run on. It is possible to move channels from one node to another.

- You create profiles (which hold most of the data for channels) in Conductor Live. Any channel on any node can use the same profile. With Elemental Live as standalone, there is no profile sharing across nodes.
- You can view the activity on all nodes in a cluster. The Elemental Live interface lets you view activity only for the individual standalone node.
- You can start and stop channels on any node in the cluster. You can add or delete a channel on any node.
- You can perform some changes on several channels at once, even if those channels are not on the same node. For example, if several channels (distributed across several nodes) all use the same profile, you can change all those channels so that they use a new profile (that is a revised version of the original profile).

Conductor Live and AWS Elemental Statmux

Centralization via the Conductor Live also has several advantages for creation of an MPTS:

- You can create an MPTS and add channels to that MPTS from Conductor Live.
- You can start and stop an MPTS.
- You can add or remove a channel on any AWS Elemental Statmux node.
- You can change properties of an MPTS in order to change the behavior of the MPTS.
- You can view the activity on all AWS Elemental Statmux nodes in a cluster

Redundancy and failover

Running Elemental Live events and Elemental Statmux MPTSes in Conductor Live lets you implement several resiliency features. These features help reduce outages in your workflows.

To clarify the terms:

- *Resiliency* refers to the ability to continue when errors occur.
- *Redundancy* refers to duplication of hardware or software components to protect against single points of failure. Therefore, redundancy is one way to achieve resiliency.

Conductor Live offers resiliency solutions in all areas of the workflow:

- Node redundancy for Conductor Live nodes, Elemental Live nodes, and Elemental Statmux nodes. This redundancy protects against failure of an entire node.

  Node redundancy is the foundation for some resiliency options within the worker nodes. For example, within Elemental Live, there are some resiliency options that only work with specific types of redundancy. Keep this fact in mind when planning node redundancy.

  When you first deploy your Conductor Live cluster, you should plan redundancy for the nodes in the cluster.

  After your initial deployment, you should review your node redundancy design when you add more nodes or when your workflows change dramatically.

  For more information about planning node redundancy, see *Setup: Plan resiliency*.

- Resiliency features in different types of workflows:
  - Encoding workflows: workflows that involve only Elemental Live.
  - MPTS workflows: workflows that involve Elemental Live and Elemental Statmux.

  As part of the design procedure for a workflow, decide which resiliency features you want to implement. An encoding workflow and in an MPTS workflow have slightly different resiliency options.

  For more information about resiliency features, see the section called "Resiliency in Elemental Statmux".

Working with AWS Elemental Live in a cluster

Running AWS Elemental Live in a AWS Elemental Conductor Live cluster provides two key benefits that are not available when you run Elemental Live in stand-alone mode:

- Centralized creation and control of channels on the nodes.
- Resiliency through configuration of *redundancy groups* of nodes.

Topics

- Components of Elemental Live in a cluster
- Comparison of profiles

Components of Elemental Live in a cluster

When you work with Elemental Live using Conductor Live, you work with channels (events), profiles, and nodes.

Channels and Events

A *channel* is a session that decodes and encodes a live video stream or a video file and produces a live output. Video input comes into the channel and video output is the final outcome of the channel. All the encoding activity occurs within a channel.

The channel that you create using Conductor Live becomes an event on the Elemental Live node.

Profiles

The encoding activity is defined in a *profile*: the information contained in the profile includes the source of the video input, the kinds of processing that the video will undergo, the types of output protocols to produce (for example, Archive or UDP/TS), and the types of outputs (the containers).

Nodes

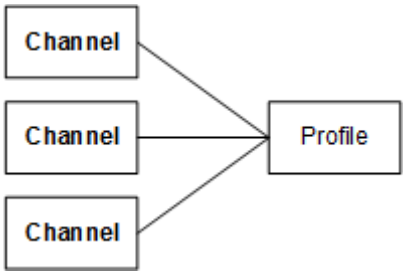
The physical computer where the video activity is handled is called a *node*. When you are deploying Conductor Live, nodes are grouped into a cluster: by adding a node to a cluster, you make it known to Conductor Live. If a node is not in a cluster, it is not being managed by Conductor Live.

Channel – Profile – Node Association

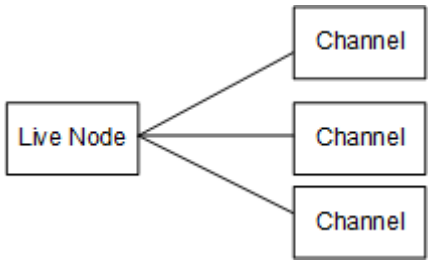
When you create a channel, you associate it with one profile and one node. So the associations between these three entities is via the channel.

In addition, keep the following in mind:

- One profile can be used by multiple channels: so profiles are multi-use.



- One node can handle multiple channels: so nodes are multi-taskers.



Comparison of profiles

The way that channels (which you create using Conductor Live) and events (which you create using Elemental Live) use their profiles is different. The way that channels use profiles has some distinct advantages in terms of visibility and maintenance.

If you have been using AWS Elemental Live in standalone mode or if you think you might occasionally run work in Elemental Live in standalone mode, you should read the information in the table to understand the differences.

Conductor Live	Elemental Live
Conductor Live profile can be used only with a channel created in Conductor Live. It cannot be used by an Elemental Live event.	An Elemental Live profile can be used only with an event created in Elemental Live. It cannot be used by an Conductor Live channel.
After a channel is created, it is linked to its profile. A link exists between a channel and the profile used. You can view the channels-profile association on the Channels page.	After an event is created, it is not linked to the profile. After the event is created, the data from the source profile exists in the event, but no link exists for that profile. You cannot query the event to find out which profile was originally used.
You cannot change a profile.	You can change a profile.

Conductor Live	Elemental Live
Instead, you can create a new profile with a new name. You can also duplicate an existing profile and then change it.	
Two channels can share a profile. If channel_A was created using profile_X and channel_B was created using profile_X, then they all have the same “profile values.”	There is no idea exists of two events “sharing” the same profile. If event_A was created using profile_X and event_B was created using profile_X, they do not automatically therefore have all the same “profile values.” (For example, if event_B was created after profile_X was modified, A and B have different values.)

Working with AWS Elemental Statmux

Include AWS Elemental Statmux nodes in your AWS Elemental Conductor Live cluster if you want to create MPTS outputs. A multi-program transport stream (MPTS) is a UDP transport stream (TS) that carries multiple programs. Conductor Live lets you create an MPTS that contains all variable bitrate programs, a mix of variable and constant bitrate programs, or all constant bitrate programs.

You use AWS Elemental Statmux to ingest SPTS outputs from AWS Elemental Live and produce MPTSes.

For Elemental Statmux, Conductor Live is a requirement. You can't run MPTSes without Conductor Live.

Topics

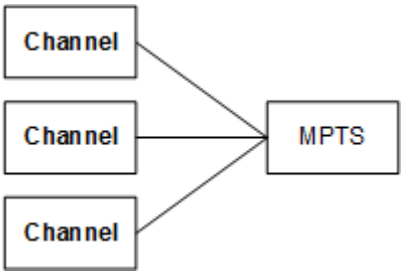
- Components of AWS Elemental Statmux in a cluster
- Features of AWS Elemental Statmux

Components of AWS Elemental Statmux in a cluster

To create an MPTS using Elemental Statmux, you need:

- Elemental Live channels that produce an SPTS output.

- An MPTS that you create using Elemental Statmux. Elemental Statmux*muxes* two or more SPTS outputs into one MPTS output.



Features of AWS Elemental Statmux

Topics

- What you can mux
- Support for multiple output configurations
- Bitrate allocation
- Closed-loop bitrate allocation
- Resiliency in a statmux workflow

What you can mux

Elemental Statmux supports muxing of the following:

- Standard muxing that includes statmuxing: An MPTS can include an SPTS that is the output of a channel (event) from an Elemental Live node that is in the same Conductor Live cluster. For more information, see the section called “Create a standard MPTS”.

- Program passthrough: An MPTS can include an SPTS program that you pass through from another source – either an MPTS from a third-party source, or an SPTS from an Elemental Live node that is in a different Conductor Live cluster. For more information, see the section called “Including passthrough programs”.
- Custom stream passthrough: An MPTS can include packets from a custom PID. For more information, see the section called “Passing through custom streams”.
- SI/PSI table passthrough: An MPTS can include SI/PSI tables that you have generated outside of Elemental Statmux. For more information, see the section called “Passing through SI/PSI tables”.

Support for multiple output configurations

The MPTS can include different types of SPTS channels from Elemental Live. Elemental Statmux has the ability to mux channels with the following characteristics:

- Different resolutions: SD, HD, and 4K.
- Different codecs: MPEG-2, AVC, and HEVC.
- Different color ranges: SDR and HDR.

Bitrate allocation

The following rules apply to the bitrate for an MPTS:

- The MPTS has a maximum bitrate that you specify. As much as possible, Elemental Statmux uses up this maximum. However, when necessary, it includes NULL bits in the MPTS.
- Each SPTS has either a constant bitrate (the video stream is CBR) or a bitrate range (the video is VBR). Elemental Statmux allocates the available bitrate among the SPTSes. Elemental Statmux continually adjusts the bitrate allocation among the SPTSes.

When you run the MPTS, Elemental Statmux automatically allocates a bit rate to each SPTS.

The automatic bitrate allocation deals with the video and audio and all ancillary data.

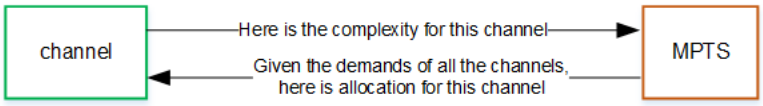
Within the MPTS, you can prioritize SPTS channels. For example, you can assign a higher priority to the SPTS channel that is the live sports event. Elemental Statmux will always assign more bitrate to more important channels, to ensure that these channels always have higher video quality.

Closed-loop bitrate allocation

The Elemental Statmux implementation of statmuxing follows a *closed-loop bitrate allocation model*.

A continual dialog is maintained between the Elemental Statmux node and the Elemental Live nodes.

For each segment in each SPTS channel, Elemental Live sends complexity information to Elemental Statmux. Elemental Statmux assesses the demands of all SPTS channels and sends a bitrate allocation response for that segment to each Elemental Live. Each Elemental Live uses the allocation response to determine the bitrate for the segment.



Resiliency in a statmux workflow

You can configure the statmux workflow for resiliency in various components of the workflow, including redundant inputs in the Elemental Live nodes and Elemental Statmux nodes.

AWS Elemental Statmux tutorial

This tutorial describes how to create a statmuxed MPTS workflow.

All the data in this tutorial is an example. You can adapt the workflow and create it yourself if, you have one AWS Elemental Conductor Live node, one AWS Elemental Statmux node, and one AWS Elemental Live node, if you have at least two sources that can be set up as inputs into Elemental Live, and if you have a multicast address where you can send an MPTS output.

Assumptions about existing knowledge

We assume the following:

- We assume that you have some familiarity with the basic features of Conductor Live. You are familiar with setting up nodes in a cluster, and with creating redundancy groups to provide resiliency for CL3 outputs.
- We assume that you are very familiar with the structure and purpose of the events (channels) that you create using Elemental Live.
- We assume that you are familiar with the uses of statmux MPTS, and with multicast and unicast. You don't need to be familiar with how to create MPTSes using Conductor Live and Elemental Statmux.

Step 1: Check your cluster and redundant nodes

We're not going to show you how to create the cluster. Assume you've already set up the following:

- Conductor Live is set up in a redundancy group as a high availability (HA) pair. This setup ensures that the MPTS workflow doesn't stop if one Conductor Live node fails.

  Assume that the redundancy group is named CL_pair.

Assume that the two nodes are called CL_1 and CL_2.

- A pair of Elemental Statmux node in a 1-to-1 redundancy group. Both nodes actively process the same MPTS, which means they are both *hot* nodes. Conductor Live manages which output gets delivered. This setup ensures that there is no interruption if one node fails.

  Assume that the redundancy group is named SM_1-to-1_hot.

  Assume that the nodes are called SM_X and SM_Y.
- Several Elemental Live nodes in an N-to-M redundancy group. Let's say there are three active nodes and one backup node serving those three active nodes. This setup ensures that encoding of the SPTS channels can quickly recover if the active Elemental Live node fails.

  Assume that the redundancy group is named Live_NM.

  Assume that the three active nodes are called EL_A, EL_B, and EL_C.

  Assume that the two backup nodes are called EL_D and EL_E.

  When you've read this tutorial, you can get more information about redundancy.

This is what you would see on the web interface for a cluster that is set up in this way:

- When you chose **Cluster**, then **Nodes** on the Conductor Live main menu, you would see the following:
  - **CL_1**, with the following information:

    A key icon that indicates that it is currently the primary node.

    The redundancy type set as **HA**.
  - **CL_2**, with the same information, but without the key icon.
  - **SM_X** and **SM_Y** with the following information:

    The name of the redundancy group.

    Note that both nodes and in the same redundancy group, and that they are the only nodes in the group. From this you can infer that the nodes are set up as a 1-to-1 redundancy group.

    The number of MPTSes that exist on each node.

Channels will always be 0 for an Elemental Statmux node.

- The five Elemental Live nodes with the following information:

  The name of the redundancy group.

  Notice that all the nodes are in the same redundancy group, and that there are more than three nodes in the group. From this you can infer that the nodes are set up as an N-to-M redundancy group.

  The number of channels that exist on each node.

  MPTS will always be 0 for an Elemental Live node.

- When you chose **Cluster**, then **Redundancy** on the Conductor Live main menu, you would see the following:
  - In the **CL_pair** redundancy group, you would see **HA**.
  - In the **SM_1-to-1_hot** redundancy group, you would see two nodes in the **Active Nodes** tab, and no nodes in the **Backup Nodes** tab. This is a 1-to-1 redundancy group, so it doesn't have backup nodes.
  - In the **Live_NM** redundancy group, you would see three nodes in the **Active Nodes** tab, and two nodes in the **Backup Nodes** tab.

After you've read this tutorial, you can get more information about redundancy.

Step 2: Create the profiles

You must create the profiles that you require for the SPTS channels that will produce the output from Elemental Live.

1. On the Conductor Live main menu, choose **Profiles**. On the **Profiles** page, choose **New Profile** (on the top right corner of the page).
2. Give the profile a name such as **`SPTS-high-resolution`**.
3. Create the profile in the usual way for everything except the output groups and outputs.
4. To set up the outputs on the profile, scroll down to **Output Groups** and choose the **UDP/TS** tab.
5. In the **New Output** section, choose the **Add Output** button.
6. For **MPTS membership**, choose **Remote**. This output section changes to display different fields. Leave the defaults for these fields.

7. In the **Streams** section, in the **Video** section, choose **Advanced** to display more fields. Set the following field:

   Rate Control Mode: Set to **Statmux**. This value indicates that the muxer will control the rate control for the output.

   Note that when you set this value, the **Bitrate**, **Max Bitrate**, and **Min Bitrate** fields don't apply, so these fields become disabled.

8. Choose **Save**.

The profile that you created is listed on the **Profiles** page. Conductor Live assigns a unique numerical ID to the profile.

Under the **Channels** column, the profile displays 0 0. This indicates that the profile is not yet being used by any channels.

Step 3: Create the SPTS channels

You must create the SPTS channels that produce the SPTS outputs. In the Conductor Live main menu, choose Channels to display the Channels screen.

1. On the Conductor Live main menu, choose **Channels**. On the **Channel** page, choose **New Channel** (on the top right corner of the page).
2. Give the channel a name such as **Program_A**. Select the profile you just created.
3. In **Node**, choose the Elemental Live node where you want the channel to run. Note that the dropdown list shows only active Elemental Live nodes. It doesn't show backup Elemental Live nodes or any Elemental Statmux nodes.
4. Choose **Save**.
5. Repeat these steps to create the second channel. Name the channel **Program_B**. Choose the same profile as you did for the first channel. Choose the same Elemental Live node.

The channels that you created are listed on the **Channels** page. Conductor Live assigns a unique numerical ID to each channel. The row for each channel specifies its profile and node.

Choose the **Profiles** page again. Note that the profile that you created now displays **0 2**, to indicate that the profile is being used by two channels but none of the channels is active.

Step 4: Create the MPTS

You are now ready to create the MPTS. You set up this MPTS to include the two channels that you created.

- On the Conductor Live main menu, choose **MPTS**. On the **MPTS** page, choose the **MPTS** button at the top right corner. The **Create a New MPTS** dialog appears.
- In the **Node** field, choose the arrow to show the dropdown list. The list shows both the Elemental Statmux nodes. In **1-to-1 redundancy** group, both nodes are active, so both nodes are listed in this dropdown. With a 1-to-1 redundancy group, there is no sense of one node being the leader and the other being the follower.

  Therefore, you can choose either node.
- Choose one of the nodes where the MPTS will run.
- Complete the other fields. Pay particular attention to these fields:
  - **Name**. Give the node a name. For example, **MyMPTS**.

    For example, choose the node **SM_X**.
  - **Transport Stream Bitrate**. Enter a value here. This value is the total bitrate for the MPTS. All the SPTS channels in the MPTS will use a portion of this bitrate.
  - **URI**. This is the address to the destination on the downstream system. Conductor Live will deliver the MPTS to this address.
  - **Output listening**. Leave this field unchecked. This field applies if you want Elemental Statmux output listening resiliency. After you've finished this tutorial, you can get more information about output listening.
  - **Add Destination**. Don't choose this button because we won't create another destination. You create a second destination for this MPTS only if you want statmux output redundancy. After you've finished this tutorial, you can get more information about output redundancy.
  - Leave the default values for other fields.
- Choose the **Advanced** tab. This tab shows information for the SI/PSI tables. Remember that when you created the profiles, you ignored the fields for the SI/PSI tables. The tables on this tab let you create tables that apply to the entire MPTS. For now, leave the defaults.
- Choose **Save**.

The MPTS appears in the list of MPTSes. The node where you created the MPTS (node **SM_X**) is part of a 1-to-1 redundancy group. Therefore, Conductor Live creates two instances of the MPTS.

The two MPTSes have identical URLs in the **Destinations** column, because the two MPTS instances are exactly identical. One node is tagged as **Primary**, the other as **Secondary**.

Conductor Live assigns a unique numerical ID. The **Channels** column displays **0 0 0**. The last 0 specifies that the MPTS doesn't have any programs yet. We'll add those in the next step.

After you've finished this tutorial, you can get more information about Elemental Statmux node redundancy.

Step 5: Add programs to the MPTS

After you create the MPTS, it has no programs in it. You must add programs before you can run the MPTS.

1. Still on the **MPTS** page, choose the MPTS by its name (**MyMPTS**). Choose the primary instance of the MPTS. The **Details** page for the MPTS appear. The page has four tabs, and the **Channels** tab is currently selected.
2. On the upper right side of the page, look for the **Add Channel** field in the gray section.
3. Choose the down arrow and choose one of the channels that you added.

   An empty row appears for the new channel.
4. In the **Basic** tab, complete the **Service Name** and **Provider Name**. Elemental Statmux uses this information to create the SDT in the MPTS.
5. Leave the other fields blank. Note that you don’t need to enter a minimum and maximum bitrate.
6. Choose **Save**.
7. Repeat to choose the other channel.
8. If you want to look at how these channels look in the MPTS, choose **MPTS** on the Conductor Live main menu.

   On the **Basic** tab, note this information:

   - **#**: A numerical ID for each program in this MPTS. The number is unique within this MPTS.
   - **ID**: A numerical ID for each program. This number is unique among all MPTSes in this Conductor Live cluster.

- **Encoders**: Specifies the number of Elemental Live encoders that are currently providing programs (channels) to this MPTS.
- **Program Number**: The PID for this program. When you added the channel to the MPTS, you had the option to specify this ID. In this tutorial, we didn't do that, so Conductor Live assigns a unique ID to each program in this field in the MPTs.
- **Min and Max Bitrate**. Remember that when we added the program, we left these fields empty. Therefore, Conductor Live automatically assigns values.

The other tabs display information that Conductor Live automatically assigns. Typically, there is no need to change these values.

Step 6: Start the MPTS

There are two steps to starting the MPTS: start all the channels, then start the MPTS. You should start processing in this order so that when the MPTS starts, it immediately starts with its full complement of programs (channels).

1. Start the channels: On the Conductor Live main menu, choose **Channels**.
2. Normally, you would identify the channels that belong to the MPTS you are starting. In our example, there is only one MPTS, so you don't need to do that.
3. In the row for each channel, choose the green arrow button on the far right.

   Look at the **Status** column and make sure that all the channels are running.
4. Start the MPTS: On the Conductor Live main menu, choose **MPTS**.
5. In the row for the first instance of **MyMPTS**, choose the **Start** (green arrow) button on the far right.
6. Repeat for the second instance of **MyMPTS**.
7. The **Status** column shows both the MPTSes as **Running**.

   They are both running, because the Elemental Statmux node is set up in a 1-to-1 redundancy group.

   One instance of the MPTS is running on one node. The other instance is running on the other node.
8. On the **MPTS** page, choose the MPTS by its name. The **MPTS Details** page appears.

9. On the left side of the page, choose the **Performance** tab. The graphic indicates the bandwidth allocation that Conductor Live is continually applying to the entire MPTS and to each program in the MPTS.

This tutorial has walked you through creating and starting an MPTS.

Conductor Live rules and limits

The following table provides a summary of the rules and constraints that apply to AWS Elemental Conductor Live features.

Feature or topic	Rule or limit
Hardware in a cluster	A Conductor Live cluster can include a maximum of 50 worker nodes.
Physical location of nodes in a cluster	Within a cluster, the Conductor Live and encoder nodes must be located in the same physical location.
	Within a cluster, the communications among cluster members shouldn't traverse the public internet.
Software versions in a cluster	We strongly recommend that all the software versions on the nodes in one cluster have the same software version, down to the patch level. So, for example, 2.21.3 on all Live nodes, and 3.21.3 on all Conductor Live nodes in the cluster. You might perform upgrades in such a way that you have two nodes with the same major version but different patch versions. For example, 2.21.3 and 2.21.5. If you experience a problem with interaction in those two nodes, AWS Elemental Support will probably request that you set up all the nodes on the same patch version.

Displaying the Conductor Live web interface

You create ECL3; encoding workflow by creating profiles and channels. When you are ready, you then start the channel. When your Elemental Live nodes are in a cluster, you should use AWS Elemental Conductor Live to create the Elemental Live encoding workflow.

Displaying the web interface

1. Obtain the following information from the person who configured the Conductor Live cluster:
   - IP address of the Conductor Live node. The correct address depends on your high-availability redundancy setup:
     - If the cluster is set up with Conductor Live high-availability redundancy, the address is the address of the VIP. This VIP is set up when you configured for high-availability redundancy, as described in the AWS Elemental Conductor Live Configuration Guide.
     - If your Conductor Live cluster is *not* set up with Conductor Live redundancy, use the IP address of the Conductor Live to set it up.
   - Your user credentials, if the cluster is set up for user login.
   - Whether the cluster is set up for worker redundancy.
2. On a web browser, enter the IP address that you obtained.

   Warning

   In a high-availability deployment, you must always access the web interface from the VIP. It is possible to access it from the primary Conductor Live, but, when a failover occurs, problems occur.

3. If the **Login** dialog appears, enter your user credentials.

The Conductor Live web interface appears. The Conductor Live main menu contains these menu items:

- **Channels**: Lets you create and work with channels that are on an Elemental Live node in the cluster.
- **Profiles**: Lets you create and work with profiles that are on an Elemental Live node in the cluster.

- **MPTS**: Lets you create and work with MPTSes (MPTS outputs) that are on an Elemental Statmux node in the cluster.
- **Status**: Lets you monitor activity on all the nodes in the cluster.
- **Cluster**: Lets you set up the cluster and set up redundancy groups in the cluster.
- **Settings**: Lets you configure hardware components (such as network cards) on the Conductor Live node, and configure the Conductor Live software.

Viewing Conductor Live configuration information

On the Conductor Live web interface, choose the **Globe** icon at the top right of the page. A drop-down menu appears showing system information:

- If redundancy is enabled on the cluster, the information includes the following:
  - The term **High Availability Enabled**.
  - The hostname beside the **Globe** icon shows the hostname of the Conductor Live that is currently the primary Conductor Live.
  - The **VIP** field shows the IP address of the VIP. This address does not match either the primary or the backup Conductor Live node.
- If redundancy is not enabled, the information appears as:

  The hostname beside the **Globe** icon shows the hostname of the Conductor Live.

  The **IP** field shows the IP address of the Conductor Live node.

For more information about Conductor Live redundancy, see the section called “Conductor Live node redundancy”.

Setup: Designing the cluster

You must design the cluster to suit the number of workflows you plan to create. You could be creating the following types of workflows:

- Encoding workflows. These workflows require only Elemental Live.
- MPTS workflows. These workflows require both Elemental Live and Elemental Statmux.

Topics

- Number of Conductor Live nodes
- Number of primary Elemental Live nodes
- Number of primary Elemental Statmux nodes

Number of Conductor Live nodes

You need two Conductor Live nodes if you plan to implement Conductor Live node redundancy. Otherwise, you need only one node.

We recommend that you implement resiliency in Conductor Live nodes. For more information, see the section called “Conductor Live node redundancy”.

Number of primary Elemental Live nodes

Determine the number of *primary nodes* you need:

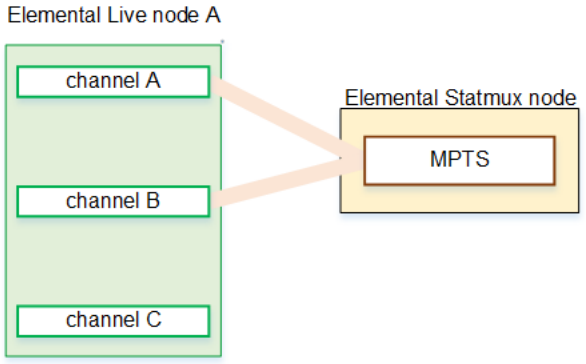
- You need at least sufficient Elemental Live nodes to run the channels for all the encoding workflows and MPTS workflows.
- You don't need to run each SPTS channel on its own node. A node can run multiple channels, including multiple SPTS channels.

After you have determined the number of primary nodes, you should identify your redundant node requirements. See the section called “Worker node redundancy”.

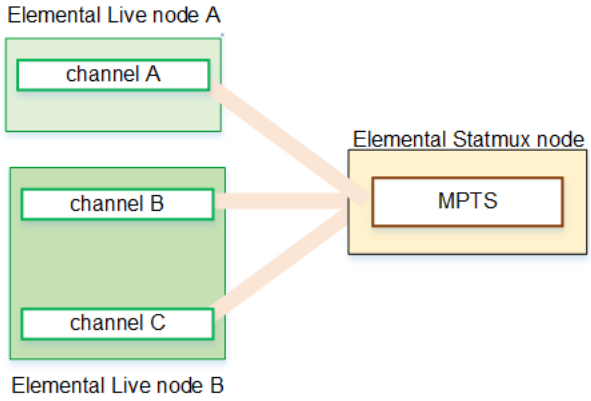
Rules for association between SPTS channels and the MPTS

The following rules help you identify the number of nodes that you need.

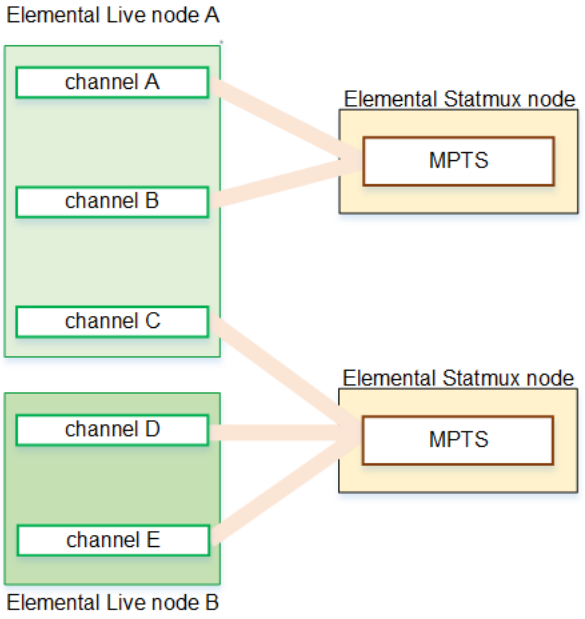
The SPTS channels for a single MPTS can originate from one node.



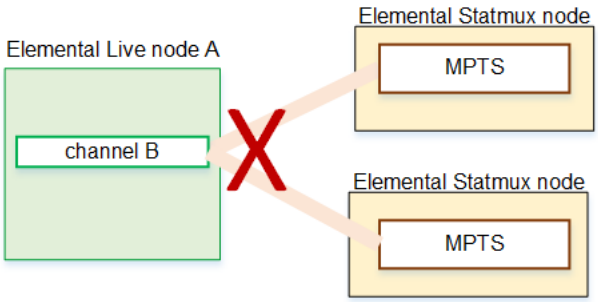
Or the SPTS channels can originate from two or more nodes.



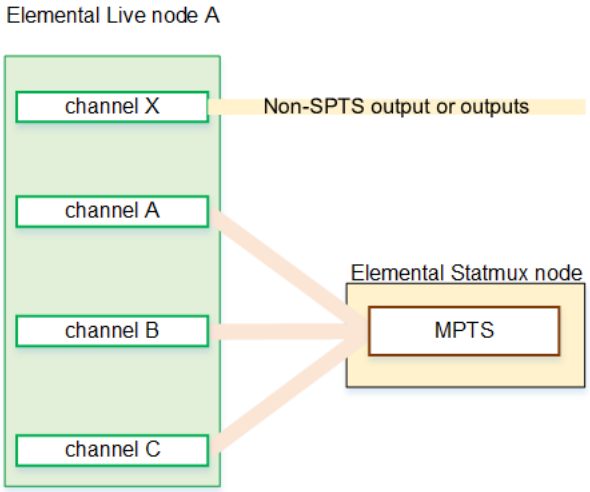
A node can contain SPTS channels that go to different MPTSes. There is no requirement for a node to be dedicated to one MPTS. In the following diagram, node A contains SPTS channels for two different MPTSes.



An SPTS channel can't be used by two different Statmux MPTS.



A node that has SPTS channels can also produce other channels (events). The node doesn't have to be dedicated to producing SPTSes.



Number of primary Elemental Statmux nodes

Determine the number of *primary nodes* you need:

- Identify the density of all the SPTS channels that you want to mux into MPTSes. Then consult with your AWS Elemental sales person for help to identify your node requirements.
- Keep in mind that you can run more than one MPTS on a node.

> **Note**
>
> You might have acquired a high-compute-power Elemental Statmux node with the intention of implementing Simulcrypt encryption, when it becomes available in Elemental Statmux.
> Simulcrypt has high compute-power requirements. You might want to plan fewer MPTSes on the node. In this way, you will be able to implement Simulcrypt later without moving any MPTSes to another node.

After you have determined the number of primary nodes, you should identify your redundant node requirements. See the section called “Worker node redundancy”.

Setup: Planning resiliency in a cluster

This section describes the resiliency features that are available on a AWS Elemental Conductor Live cluster.

Topics

- Conductor Live node redundancy
- Worker node redundancy
- Organizing redundancy groups in the cluster
- Resiliency features in Elemental Statmux

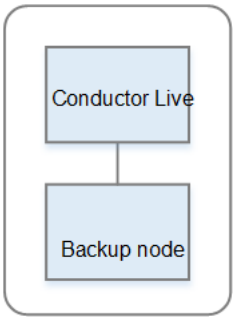
Conductor Live node redundancy

You need two Conductor Live nodes to implement Conductor Live node redundancy. Otherwise, you need only one node.

We recommend that you set up the cluster with redundant nodes. If you do, you obtain resiliency in the Conductor Live nodes. In addition, there are some resiliency features available to Elemental Live and Elemental Statmux that only apply if you have redundant Conductor Live nodes.

Redundant Conductor Live nodes are known as a *high availability* (HA) pair.

To set up redundant nodes, add two nodes to the cluster, then create an HA redundancy group. When a problem occurs on the active node, the backup node automatically takes over control of activity in the cluster.



How Failover works

If the leader node fails, the backup automatically takes over management of the cluster. The leader Conductor Live maintains the Conductor Live database; the backup database is a copy of that

leader database and is continually being synchronized. The backup Conductor Live is continually monitoring the leader.

As soon as the backup can no longer detect the leader on the network, it assumes that the leader has failed and its takes over the leader role. This change in role takes a few seconds.

After failover has occurred, the two Conductor Live nodes continue in their new roles. Even when the failed Conductor Live comes back online, it does not take back the leader role.

Worker node redundancy

This section describes redundancy options for worker nodes in an AWS Elemental Conductor Live cluster. Worker nodes are Elemental Live nodes and Elemental Statmux nodes. The same redundancy options are available to both types of worker nodes.

You can set up worker nodes in a group in order to provide node redundancy. When a problem occurs on an active node, a backup node takes over.

- For Elemental Live nodes, we recommend that when you have statmux workflows, you set up Live nodes for redundancy, even if your cluster requires only one Elemental Live node.
- For Elemental Statmux nodes, we recommend that you always set up the nodes for redundancy.

You set up node redundancy by setting up redundancy groups. There are three types of groups:

- N-to-M
- 1-to-1
- 1-to-1 Plus

You can set up multiple redundancy groups in the cluster, of the same or different types. For example, some nodes in two N-to-M redundancy groups, and more important nodes in a 1-to-1 Plus redundancy group. The redundancy groups always operate separately from each other.

Node Failure Detection

Conductor Live maintains contact with the worker nodes in the cluster. If Conductor Live can no longer communicate with the node, its assumes that the worker node has failed.

Nodes that are not part of a redundancy group will not fail over, but Conductor Live will still detect a failure.

Node failure detection is always enabled in Conductor Live. You don't need to configure it.

Topics

- N-to-M redundancy
- 1-to-1 redundancy
- 1-to-1 Plus redundancy

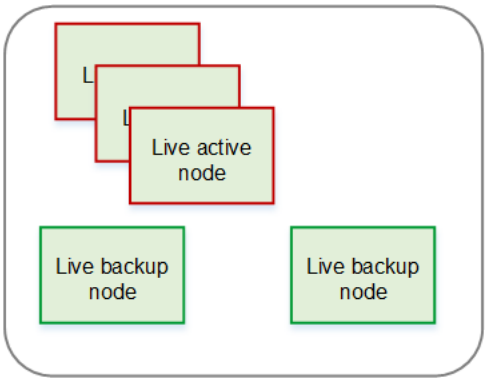
N-to-M redundancy

Setup

The redundancy group contains one or more active nodes and one or more backup (inactive) nodes. In the group, you can have the same number of active and backup nodes, or more active nodes, or more backup nodes.

In one redundancy group, all the active nodes share the backup nodes.

This diagram is an example of an N-to-M redundancy group for Elemental Live nodes. The same design applies to Elemental Statmux nodes.



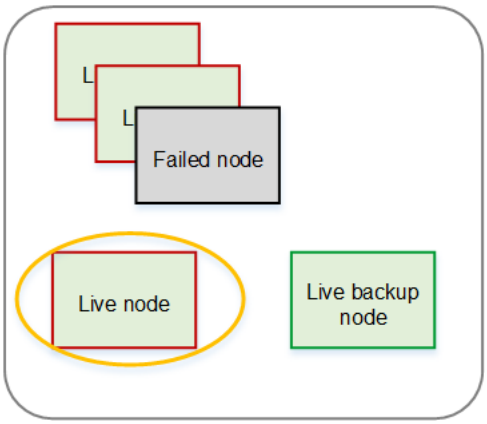
What happens in a failure

If an active Elemental Live node fails, Conductor Live automatically moves both the active and idle channels to a backup node, then starts all the active channels. There is a slight delay while the restart occurs.

If an active Elemental Statmux node fails, Conductor Live automatically moves all the active and idle MPTSes to a backup node, then starts all the active MPTSes. There is a slight delay while the restart occurs. In addition, Conductor Live ensures that the Elemental Live nodes send to the new Elemental Statmux node.

There is a delay while the backup node starts up because Conductor Live must copy the data from the failed node to the backup node. During the delay, there is no output for the affected channels or MPTSes.

This diagram illustrates the change in the group after one node fails. This diagram is for Elemental Live but the same pattern applies to Elemental Statmux.



Considerations

- You must consider the capabilities of the different nodes in the redundancy group. For example, think about the repercussions if you have a backup node that isn't as powerful as the nodes that are usually your active nodes. Think about whether you want to take the risk of having less powerful nodes as backups.

  Also consider how you will handle failure of a node that has SDI cards installed. Ideally, there will be a backup node with the same card configuration, espeically if your deployment includes a router handling the SDI input. You might want to consider organizing nodes that have SDI cards in their own redundancy group.
- You should have a policy in place for handling node failure. Decide whether you will immediately try to get the failed node back into production.
- Keep in mind that it is possible to have so many nodes in a failed state that you have no backup nodes in the redundancy group.

1-to-1 redundancy

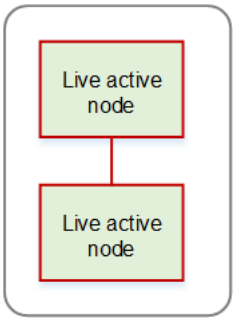
Setup

The redundancy group contains one pair of nodes that are both active. You designate one node as the primary node, and the other as the secondary node.

When you create a channel or MPTS, you assign the node to the primary node. As soon as you save the channel, Conductor Live automatically duplicates the channel (or MPTS) onto the secondary node. If you later make changes to the channel or MPTS, Conductor Live automatically applies those changes to the channel or MPTS on the secondary node.

You start the channel or MPTS on the primary node. Conductor Live automatically starts the channel or MPTS on the secondary node. In this way, the two nodes are both *hot.*

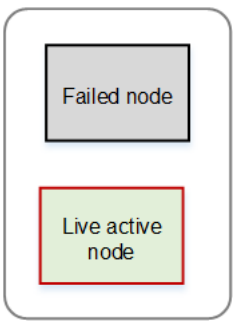
This diagram is an example of a 1-to-1 redundancy group for Elemental Live nodes. The same design applies to Elemental Statmux nodes.



What happens in a failure

If one of the nodes fails, the other node continues to process the content. There is a delay of a few seconds before the output resumes.

This diagram illustrates the change in the group after one node fails. This diagram is for Elemental Live but the same pattern applies to Elemental Statmux.



Considerations

- The two nodes must have identical capabilities.
- You should have a policy in place for recovering after a node failure. Decide whether you will immediately try to get the failed node back into production.
- When you get a failed node back into production, you must restart each channel or MPTS that was running on that node. You will then be back to a redundant setup for the nodes.

1-to-1 Plus redundancy

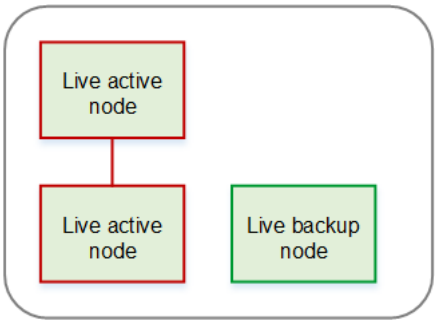
Setup

The 1-to-1 Plus redundancy group is the same as a 1-to-1 redundancy group except that it adds one backup (inactive) node.

The behavior for starting and running a channel (or MPTS) is identical to the behavior in a 1-to-1 redundancy group.

This diagram is an example of a 1-to-1 Plus redundancy group for Elemental Live nodes. The same design applies to Elemental Statmux nodes.

The backup node is dedicated to one redundancy group. One backup node can't act as backup for two 1-to-1 redundancy groups.

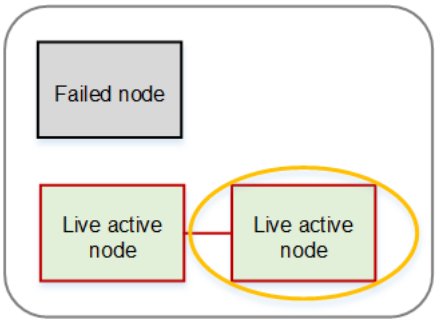


What happens in a failure

If one of the nodes fails, the other node continues to process the content. There is a delay of a few seconds before the output resumes. In addition, if the failure is in an Elemental Statmux node, Conductor Live redirects the Elemental Live output to the new Elemental Statmux node.

After Conductor Live has switched to delivering from the second node, the backup node becomes an active node. Therefore, immediately after the failure, there are three nodes in the Active nodes list—the two active nodes and the failed node.

This diagram illustrates the change in the group after one node fails. This diagram is for Elemental Live but the same pattern applies to Elemental Statmux.



Considerations

- The two nodes must have identical capabilities.
- You should have a policy in place for recovering after a node failure. Decide whether you will immediately try to get the failed node back into production.
- When you get a failed node back into production, you must restart each channel or MPTS that was running on that node. You will then be back to the desired redundant setup for the nodes.

Organizing redundancy groups in the cluster

These rules and guidelines apply to organizing multiple redundancy groups in the cluster.

- You can create as many redundancy groups as you want. For example, you can create several Elemental Live redundancy groups, and these groups can be different types.
- You can mix and match different types of redundancy groups. For example, you can organize all your Elemental Live nodes in one or more N-to-M redundancy groups, but organize your Elemental Statmux nodes in a 1-to-1 redundancy group.
- You should think about how you want to associate redundancy groups of Elemental Live nodes with redundancy groups of Elemental Statmux nodes.

  There is no rule in Conductor Live that forces you set have all the channels in one Elemental Live redundancy group serve only the MPTSes in a single Elemental Statmux redundancy group. But

you might find it is easier to manage failures, and to track what has happened, if you do enforce associations yourself.

Resiliency features in Elemental Statmux

Elemental Statmux includes two features that provide resiliency for Elemental Statmux – output listening and output redundancy.

Topics

- Output listening in Elemental Statmux
- Output redundancy in Elemental Statmux
- Output listening combined with output redundancy

Output listening in Elemental Statmux

If your MPTS is on a node that is in a 1:1 (or 1:1 Plus) redundancy group, you can set up the MPTS for output listening.

Output listening works only with multicast delivery to the system that is downstream of Elemental Statmux. It protects against the following problems:

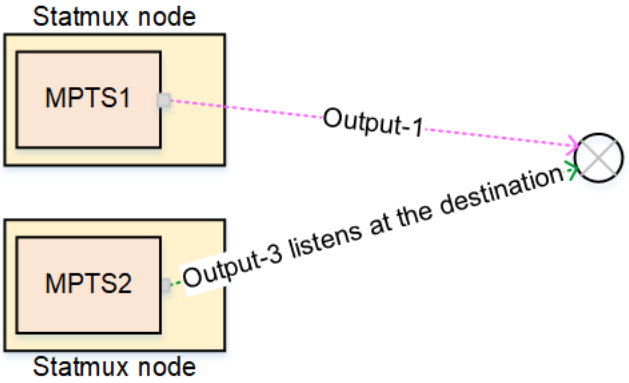
- Muxing problems within the MPTS.
- Failure on an Elemental Statmux node.

Setup

You set up for output listening by checking the **Output Listening** field when you create the MPTS. See the section called “Output tab”.

Conductor Live replicates the MPTS on the two nodes in the redundancy group. The two MPTSes adopt roles. One MPTS is the primary MPTS, the other is the secondary MPTS. When you start the MPTS, Conductor Live automatically starts it on both nodes. Both MPTSes mux the output, but only the primary MPTS delivers the output to the destination. The secondary MPTS continually listens on the multicast destination, to monitor the health of the other MPTS.

The following diagram illustrates the setup.



What happens in a failure

A failure might occur on the primary MPTS, either because there is a problem in the muxer, or because the node fails.

In either case, the secondary MPTS detects that the first MPTS is not delivering, and it automatically starts to deliver to the same destination. The secondary MPTS is already muxing, so there is minimal disruption in delivery.

Output redundancy in Elemental Statmux

You can set up the MPTS with two destinations. Elemental Statmux delivers the MPTS to two different addresses.

Output redundancy works on any type of redundancy setup. The node can be in any type of redundancy group, or it can be outside a redundancy group.

Output redundancy protects against the following:

- Failure of the output interface on Elemental Statmux.
- Failure in the network path to the downstream system.

This resiliency feature works with either unicast or multicast delivery to the system that is downstream of Elemental Statmux.

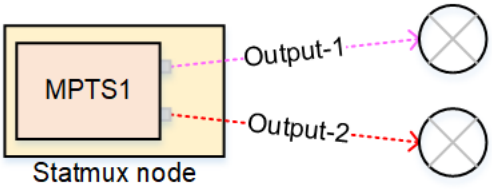
Output redundancy with N-to-M redundancy

With this redundancy setup, you set up for output redundancy by specifying two destinations when you create the MPTS output. The two addresses can be identical or different. Usually the

interfaces are different (as shown in the diagram), to protect again switch failure in the node. The downstream system must be able to handle the type of delivery.

The MPTS continually delivers two outputs. If a failure occurs, the downstream system must be set up to detect problems and react appropriately.

The following diagram illustrates the setup. One MPTS has two destinations to the downstream system.

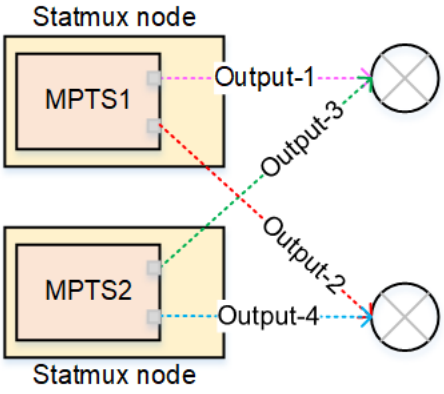


Output redundancy with 1-to-1 redundancy or 1-to-1 Plus redundancy

With this redundancy setup, you set up for output redundancy by specifying two destinations when you create the MPTS output. The two addresses can be identical or different. Usually the interfaces are different (as shown in the diagram), to protect again switch failure in the node. The downstream system must be able to handle the type of delivery.

The MPTS continually delivers two outputs from each node in the redundancy group. If a failure occurs, the downstream system must be set up to detect problems and react appropriately.

The following diagram illustrates the setup where there is 1-to-1 redundancy or 1-to-1 Plus redundancy.

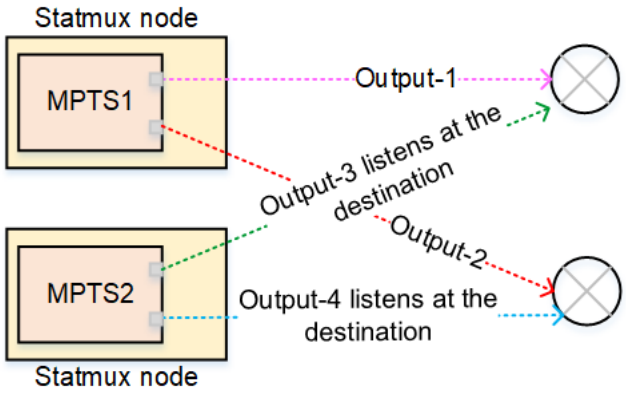


Output listening combined with output redundancy

If your MPTS is on a node that is in a 1:1 (or 1:1 Plus) redundancy group, you can combine output listening and output redundancy for an MPTS.

With this setup, you have two outputs from each node. The secondary MPTS continually listens on the two multicast destinations, to monitor the health of the other MPTS. Compare the diagram below to the diagram for output redundancy with 1-to-1 redundancy. There is a slight difference in the role of the secondary MPTS at each downstream destination.

This setup combines all the failure protection of output listening and output redundancy.



Designing an MPTS workflow

This section describes how to design a standard MPTS, and how to augment that standard MPTS by including passthrough elements.

You can configure the MPTS to include multiple programs.

You can configure the MPTS to generate the following SI/PSI tables:

- PAT. Required if you want to create a compliant MPTS.

- PMT for each program. Required if you want to create a compliant MPTS.
- NIT. Always optional.
- SDT. Always optional.
- TDT. Always optional.

You can also configure the MPTS to pass through any SI/PSI tables that you pass in, both the tables the Elemental Statmux can generate, and those that it never generates.

Topics

- Creating a standard MPTS
- Including passthrough programs
- Passing through custom streams
- Passing through SI/PSI tables

Creating a standard MPTS

This section describes how to create a standard MPTS. A standard MPTS is one where each STPS programs comes from a channel that is run on an Elemental Live node in the cluster. The SPTS programs can be any mix of VBR and CBR programs. The MPTS is fixed bandwidth, but the programs can be a mix of VBR and CBR programs.

Assumptions

This section assumes the following:

- You know how to create profiles and channels using AWS Elemental Conductor Live.
- You are familiar with the encoding features of AWS Elemental Live.
- You are familiar with the purpose of SPTS and MPTS, and of muxing and statmuxing.
- You are familiar with the structure of an SPTS and MPTS, and specifically with SI/PSI tables, with programs, and with handling of packets in the transport stream.

Topics

- Step 1: Create the profiles and SPTS channels
- Step 2: Create the MPTS and add channels
- Handling by Elemental Statmux

Step 1: Create the profiles and SPTS channels

You perform these steps in Conductor Live. You don't perform them on Elemental Live.

1. Design the profile for each SPTS channel.
   - Identify the input or inputs. The SPTS channel can use any Elemental Live inputs.
   - Identify the features you want to enable. For example, ad avails via SCTE-35 and motion overlay. There are no special rules about features that can be enabled in an MPTS channel. The channel follows the same rules as a regular Live event.
   - Identify the outputs you want to create:
     - You must always include one UDP/TS output that you configure for the MPTS. This output is called an *MPTS output*.
     - You can also include other UDP/TS outputs, for delivery to a regular UDP server.
     - You can include any number of outputs of another type. For example, you can include an HLS output group in order to produce an ABR stack for an OTT workflow.
   - Identify fields in the profile that you must set up as profile parameters. For more information about profile parameters, see the section called “Work with channel parameters”.
2. As part of the design of the *MPTS* output (in the UDP/TS output group), consider the following:
   - You must set up to include the SI/PSI tables that Elemental Statmux requires — the PAT and the PMT.
   - You can include or exclude the NIT, SDT, and TDT tables. In the MPTS, you have the opportunity to configure them again, for the entire MPTS.

- You don't set the PIDs for the video, the audio, most captions, or the PCR.
- You can choose to include ancillary data such as SCTE 35 and Nielsen ID3 data. But in all cases, you don't set the PIDs.

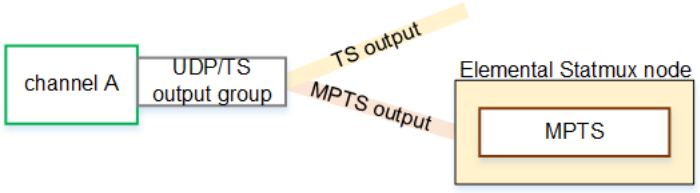
When Elemental Live creates the output, it creates a PMT that references all the included streams, and it creates a PAT. It creates other tables according to the channel instructions.

3. Create the profile. For more information, see the section called “Create profile from scratch”.
4. Create the channels for all the profiles that you have created. Create the channel in the way that you create any channel using Conductor Live. You can create the channel from scratch, or you can duplicate an existing channel.

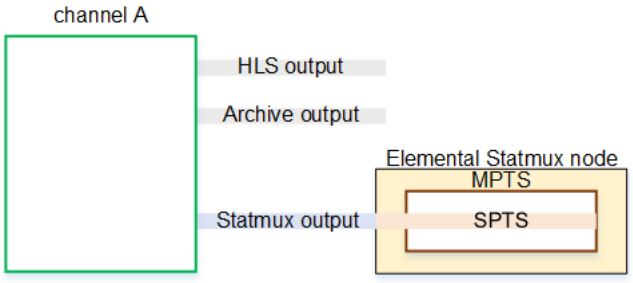
Rules

The following rules apply to the Elemental Live profiles and channels:

- Everything about the profile can be identical to a regular profile used by a non-SPTS channel, except for the outputs. The profile must include one UDP/TS output that is set up for MPTS.
- The channel can produce a UDP/TS output group that produces two outputs, one that is a regular SPTS (not a statmux SPTS), and one that is a statmux output. You might create this TS output as a *monitoring output*. Creating this output doesn't add to the workload on the channel.



- The channel can include both statmux outputs and non-statmux outputs. These non-statmux outputs can be of any type, including other UDP/TS outputs (that go to other destinations).



Step 2: Create the MPTS and add channels

Create an MPTS. To create the MPTS, see the section called “Create MPTS”. The MPTS appears in the list of MPTSes.

Then add channels (SPTS programs) to the MPTS. Specify a channel that exists in the cluster. This channel only ever has one program, so there is no need to tell MPTS which program to extract.

For each program you add, you can assign the following:

- Data to use in the SI/PSI tables in the output MPTS.
- PIDs to assign to this program in the output MPTS.
- The bitrate range for this program in the MPTS.
- Locations for the different types of communications that occur between the Elemental Live node and the Elemental Statmux node.

For much of this information, if you don't specify values, Elemental Statmux automatically assigns values when you save the MPTS. Elemental Statmux ensures that valid PIDs are assigned throughout the MPTS.

Handling by Elemental Statmux

When Elemental Statmux ingests each SPTS program, it handles the data as follows:

It uses the PAT and PMT to extract all the streams from the program—the video, audio, and so on.

It reads any of the optional tables, such as the SDT.

It uses the extracted data to create one set of new tables for the MPTS. If a program didn't include an optional table (such as the SDT), Elemental Statmux uses the information that you might have specified directly in the MPTS. Elemental Statmux assigns the standard PIDs to the tables.

Elemental Statmux assigns new numbers to program streams, in order to avoid conflict when two SPTS programs use the same PID for the same stream. Elemental Statmux provides fields where you can assign new numbers. But typically in a standard MPTS, you let Elemental Statmux automatically assign numbers.

Elemental Statmux also generates its own NULL packets, to pad the MPTS and ensure a constant bitrate.

Including passthrough programs

In any MPTS, you can include any number of *passthrough program*s. A passthrough program is a source program that doesn't come from an Elemental Live node that is in the Conductor Live cluster. The program could be either of the following:

- An MPTS produced by another encoder. That MPTS could include one or more SPTSes. Elemental Statmux lets you extract only the SPTSes that you want.
- An MPTS produced by an Elemental Live node that is not in the Conductor Live cluster. In this case, the MPTS is an output from a UPD/TS output group, where the MPTS Membership of the output hasn't been set up as remote.

Setting up

You include the program by setting it up in the MPTS as a *passthrough program*. These rules apply:

- The source programs must be well-formed MPTSes. They must contain all the tables that Elemental Statmux expects, so that Elemental Statmux can process the program in the same way as it processes standard Elemental Live SPTSes.
- The source programs must have CBR video streams. They can't have VBR streams.

To include passthrough programs, design the workflow in the regular way. When you create the MPTS, add passthrough programs. For each program, you must specify the following information:

- The location where the upstream system is sending the source program. Elemental Statmux listens for the stream at that location.
- The specific program to extract from the source MPTS.
- Data to use in the SI/PSI tables in the output MPTS.
- PIDs to assign to this program in the output MPTS.

For much of this information, if you don't specify values, Elemental Statmux automatically assigns values when you save the MPTS. Elemental Statmux ensures that valid PIDs are assigned throughout the MPTS.

For detailed instructions about adding a passthrough program, see the section called “Including passthrough programs”.

Handling by Elemental Statmux

When the MPTS starts, Elemental Statmux connects to the source transport stream that you specified and extracts the program that you specified.

- Elemental Statmux uses the PAT and SDT to identify the PMT, then uses the PMT to identify the streams (the video, audio, and data).
- It then discards all the SI/PSI tables from the source.
- It assigns new output PIDs to the streams. You could specify the output PIDs that you want Elemental Statmux to use, but that step is optional.
- It restamps the PCR in each program as it inserts the packets.
- It creates a new PMT for the program, and includes that PMT in the PAT and SDT for the MPTS.

Passing through custom streams

You can passthrough any stream from a source TS to the output MPTS. You can pass through any number of streams.

This section describes the procedure for passing through a stream whose PID isn't reserved for an SI/PSI table. For example, you might want to pass through a stream that isn't part of any program because your downstream system can use the data.

If you want to pass through an SI/PSI table rather than a custom stream, see the next section.

Setting up

You include the stream by setting it up in the MPTS as a *passthrough stream*. The source of the stream can be any well-formed transport stream.

To include a passthrough stream, design the workflow in the regular way. When you create the MPTS, add the passthrough stream. You must specify the following information for the stream:

- The location where the upstream system is sending the source transport stream. Elemental Statmux listens for the stream at that location.
- The PID to assign to this stream in the output MPTS.

For detailed instructions about passing through a stream, see the section called “Including passthrough streams”.

Handling by Elemental Statmux

When the MPTS starts, Elemental Statmux connects to the location where the upstream system is publishing the transport stream, and extracts the packets for the specified PID.

- Elemental Statmux ignores and discards any SI/PSI tables that are in the transport stream. It extracts only the specified stream.
- Elemental Statmux includes the stream in the MPTS.
- Elemental Statmux doesn't include the stream PID in any of the SI/PSI tables for the MPTS.

Passing through SI/PSI tables

You can pass through any stream from a source TS to the output MPTS. You can pass through any number of streams.

This section describes the procedure for passing through one or more SI/PSI tables that have been created outside of Elemental Statmux. If you want to pass through a custom stream rather than an SI/PSI table, see the previous section.

You can pass through the following:

- An SI/PSI table that Elemental Statmux doesn't implement. For example, an EIT.
- An SI/PSI table that Elemental Statmux does implement. For example, a PAT.

You include the table by setting it up in the MPTS as a *passthrough stream*.

You can combine SI/PSI table passthrough with custom stream passthrough and program passthrough.

Setting up

You pass through one or more SI/PSI tables by setting it up in the MPTS as a *passthrough stream*.

To include passthrough streams, follow these steps:

- Design the MPTS workflow in the regular way.
- When you create the MPTS, add the passthrough streams.

  You must specify the following information for the stream:

- The location where the upstream system is sending the source program. Elemental Statmux listens for the stream at that location.
- The PID to assign to this stream in the output MPTS.

For detailed instructions about passing through a stream, see the section called “Including passthrough streams”.

- You might want to suppress some or all of the tables that Elemental Statmux usually generates.
  - You must suppress table generation for a specific table if you are also passing through that table. Elemental Statmux won't allow two tables with the same PID.
  - You can optionally suppress table generation even if you aren't passing through the table. For example, you can suppress the PAT, even if you aren't passing it through. The MPTS won't include a PAT.

  For detailed instructions about suppressing table generation, see the section called “Advanced tab – Suppressing generation of SI/PSI tables”.

Handling by Elemental Statmux

When passing through this type of stream, Elemental Statmux reads the PID of a passthrough table, to ensure that it doesn't conflict with other PIDs. But it treats the contents as a *black box*. Elemental Statmux doesn't perform any validation on the contents.

It is your responsibility to make sure that all the SI/PSI tables are acceptable to the downstream system. It is your responsibility to ensure that the SI/PSI tables correctly reference the program and stream PIDs.

Daily operations: Running channels and MPTSes

To start and stop channels on AWS Elemental Live nodes that are in a cluster, use AWS Elemental Conductor Live. When the channel is running, you can control the inputs and the outputs.

Topics

- Starting and stopping a channel
- Starting channels using a schedule
- Controlling channels
- Starting or stopping an MPTS

Starting and stopping a channel

Starting and stopping one channel

On the Conductor Live main menu, choose **Channels**. Then choose the green **Start** button to the right of the channel that you want to start.

To stop a channel, choose the **Stop** button for that channel. When you stop a channel, it is displayed at the point at which it was stopped on the Conductor Live web interface. When the channel stops on the Conductor Live web interface, it also stops and is archived on the AWS Elemental Live web interface.

Starting or stopping several channels at one time

Step A. Create the task

You can use the Tasks feature to start or stop several channels at once.

1. On the Conductor Live main menu, choose **Channels**.
2. On the **Channels** page, choose **Tasks**, then choose **Start Channels** or **Stop Channels**.
3. Select the channels for the action. On the **Start Channels** page, only channels that are currently not running appear in the list. On the **Stop Channels** page, only channels that are currently running appear.
4. On the final page, choose **Process Now** or **Save for Later**.

Process Now: Conductor Live applies the change. The **Channels** page reappears, showing the change.

Save for Later: This option lets you queue up several tasks and then perform them in one pass.

> **Warning**
>
> **Save for Later** is intended to queue for a short time.
> Don't use **Save for Later** and then delay process the task in a few hours. Doing so might create undesired consequences.

Step B. Monitor the status of the task

When you choose **Process Now** to run the task, you can monitor the status of each stop or start in the task.

1. On the Conductor Live main menu, choose **Status**, then choose **Task Reports**. The **Task Reports** page appears, with a list of tasks in the panel on the left. Newer tasks appear first.
2. Select a task from the list. The details for the task appear. A line appears for each action in the task. Each line shows the current status of the action: Pending, Success, or an explanation of why the task failed.

   As the status of an action changes, the page automatically refreshes to display the latest information.

Starting channels using a schedule

You can schedule channels to run once or repeatedly, and for a specified duration. Schedules must be at least 60 seconds in duration, with at least one minute between scheduled runs.

To schedule a channel to run

1. On the Conductor Live main menu, choose **Channels**.
2. Choose the channel by selecting its ID or name.
3. Select the **Schedules** tab on the left. Then choose the **New Schedule** button. Complete the **New Schedule** dialog:

- **Duration**: Enter a duration. The minimum duration is 60 seconds. Or leave the duration empty to create a schedule without an end time.
- **Run once** or **Repeat**: Complete the fields to specify the number of times to run the schedule.

4. Choose **Save**. The new schedule is added to the list of schedules.
5. Choose **Enable**. The channel will run as specified.

See also the section called “View active schedules”.

The procedure

View active schedules

After you have enabled at least one schedule, choose **All Active Schedules** to see a calendar view of all active schedules. You can change the display to show one day, week, or month.

Controlling channels

When a channel is running, you can control its behaviour from the Conductor Live web interface in several ways. These features work in the same way as they work on the Elemental Live node.

1. On the Conductor Live main menu, choose **Channels**, then select the channel by ID or by name. The **Channel Details** page appears.
2. On the **Channel Details** page, choose the **Status** tab.

Switching inputs

If the channel is configured with more than one input, you can switch to a different input.

The **Status** tab shows the inputs in the channel. The input that is currently being ingested has an active icon. It also shows a green line going from the input to the output panel.

To switch, choose the green **Start** button to the left of another input. The selected input turns blue and is represented by a rotating circle of dots while it connects. The buttons dim and additional actions are disabled until the input has switched.

Controlling ad avail blanking for the channel

If the channel has been configured for ad avail blanking, an Ad avail blanking viewer shows whether blanking is currently active.

To start or stop blanking, select the window.

Starting, stopping, and pausing the output

You can start, stop, and pause each output that is in the channel.

In the output panel, there is a section for each output. In each section, buttons appear for the actions that apply to that output type.

Select the button.

Starting or stopping an MPTS

On the Conductor Live main menu, choose **MPTS**.

To start an MPTS, select the **Start** icon. Note that starting an MPTS doesn't automatically start the programs (channels) on the Elemental Live nodes.

To stop an MPTS, select the **Stop** icon at the right side of the MPTS. Note that the sources in the MPTS (the channels, passthrough programs, and passthrough streams) do not stop.

Monitoring workflow activity on Conductor Live

This topic covers the operations you use to run and manage nodes in AWS Elemental Conductor Live.

Topics

- Monitoring channels
- Monitoring MPTS outputs
- Monitoring alerts and messages
- Monitoring nodes
- Monitoring the load on worker nodes

Monitoring channels

Topics

- Monitoring the health of channels
- Monitoring channel activity at the node
- Viewing channel history

Monitoring the health of channels

You can monitor the status of channels as they run.

1. In the AWS Elemental Conductor Live main menu, choose **Channels** . Information is color-coded as follows:
   - Yellow background shading indicates that there are active alerts on the channel that you have not yet read and suppressed.
   - Red background shading indicates that the status of the channel is Error.
2. Display more information if you want:
   - Choose any red icon to go to the **Status – Messages** page. This page shows all messages for this channel. The error message is shaded red and have the same red icon.
   - Choose any orange icon to go to the **Status – Alerts** page. This page shows detailed information about any alerts for this channel.

Monitoring channel activity at the node

You can view information about the channel activity that is happening at any worker node.

1. In the Conductor Live main menu, choose **Channels** .
2. Select any channel by its ID or name. The **Channels Details** page appears.

When a channel is running, information appears in three tabs: **Status**, **Parameters**, and **Logs**.

Elemental Live constantly forwards _eme and _ eme_ve logs to Conductor Live.

Note that channel logs are displayed for 24 hours. Logs that are from 24 hours to one-week old are held in zip files that you can unzip if needed.

Viewing channel history

You can view a summarized history of the channel.

On the **Channel Details** page, choose **History**.

Time is shown in the time zone currently configured on the Conductor Live. The timeline captures when a channel gets created, started, and stopped, and also includes node information.

Monitoring MPTS outputs

Topics

- Monitoring the progress of all MPTSes
- Monitoring the muxing performance of an MPTS
- Modifying the MPTS while it is running

Monitoring the progress of all MPTSes

You should monitor every MPTS constantly to ensure that none has failed.

1. On the AWS Elemental Conductor Live main menu, choose **Status**, then choose **Overview**. Information about the MPTS appears.
2. If at least one MPTS is in error, go to the **MPTS** page.
3. Look for any MPTS that has an orange icon in the **Status** column.

4. Choose an orange icon to go to the **Status – Alerts & Messages** page. This page appears with the filter set to show only the information for this MPTS output.
5. Review the alerts and messages to determine why the MPTS failed.

Monitoring the muxing performance of an MPTS

To monitor the muxing of an individual MPTS, display the **MPTS** page and choose **Performance** (graph icon) beside the item.The MPTS **Details** appears with the **Performance** tab on top.

Modifying the MPTS while it is running

You can modify the MPTS ouput even when it is running:

- You can modify its properties.
- To add or remove channels.

See the section called “Modify MPTS”.

Monitoring alerts and messages

AWS Elemental Conductor Live generates alerts and messages to provide information about the status of the nodes in the cluster and about the encoding channels. This section covers how to monitor alerts and messages via the web interface.

For information about setting up automatic email or web callback alert notifications, and about using the SNMP and REST interfaces for alerts and messages, see AWS Elemental Conductor Live Configuration Guide.

Topics

- About alerts and messages
- Alerts and messages on the web interface

About alerts and messages

In the following table, read down the first column to find the type of information that you're interested in. Then read across to find the interfaces that provide alerts about that information and that provide messages about that information.

Type of information	Alerts	Messages
Access Options	Web Interface Automatic email notification Web callback notification SNMP trap SNMP poll REST calls	Web Interface SNMP poll REST calls
Information Conveyed	Alerts are feedback on a problem that must be fixed. The **Channel Error** alert informs you that a channel has moved to an Error state. This can help when you are receiving automatic email notifications, to let you know to check for related messages on the web interface.	Three types of messages are: **AuditMessage:** Informational messages that you do not need to react to. Often, these messages are feedback to actions you performed. **WarningMessage:** Messages that advise you that there is a risk that a future activity will fail unless you take action to prevent it. **ErrorMessage**: Messages that indicate that a planned activity has failed or an unexpected system error has occurred.
Active/Inactive	Alerts remain active until the underlying problem is resolved. When the cause of the alert is no longer present,	Messages are neither active nor inactive. They are defined as “recent” when they are fewer than 24 hours old.

Type of information	Alerts	Messages
	the system clears the alert so that it becomes inactive.	
Visibility	You can toggle the visibility of active alerts. Suppressing an alert this way is similar to marking an email as read. The section below describes where you can see suppresse d and unsuppressed alerts on the web interface.	Only messages of the type **Error** are visible in the header. You can toggle the visibilit y of recent error messages, which is similar to marking an email as **read**. The section below describes where you can see suppresse d and unsuppressed messages on the web interface.

Alerts and messages on the web interface

Conductor Live provides information about alerts and messages in two places:

- On the header of every page.
- In more detail on the pages **Status – Alerts** and **Status – Messages**.

Web interface header: Alerts

The web interface header, located at the top of all pages of the web interface, shows a count of alerts that are both active and visible:

- Active: the condition that is causing the alert still exists.
- Visible: no user has marked the alert as *read*.

The count is in a red circle to the right of the information (**i**) icon.

1. Select the red circle to display a pop-up list of the ten most recent active, visible alerts.

2. Optionally, choose the suppress (**x**) icon to dismiss this alert. The alert will remain active until the underlying cause is resolved. It won't appear in the popup list. But it is still listed in the **Status – Alerts** page, under the **Active** tab.

   You can unsuppress the alert on the **Status – Messages** page.

Web interface header: Messages

The web interface header, located at the top of all pages of the web interface, shows a count of error messages that are both recent and visible:

- Active: the messages was created in the last 24 hours.
- Visible: no user has marked the alert as *read*.

The count is in a red circle to the right of the information (**i**) icon.

1. Select the red circle to display a pop-up list of the ten most recent, visible alerts.
2. Optionally, choose the **Suppress** (**x**) icon to dismiss this alert. The alert will remain active until the underlying cause is resolved. It won't appear in the popup list. But it is still listed in the **Status – Alerts** page, under the **Active** tab.

   You can unsuppress the alert on the **Status – Messages** page.

Status – Alerts page

On the Conductor Live main menu, choose **Status**. Then choose **Alerts** in the left panel.

The **Alerts** page contains three tabs, for active, inactive, and all alerts.

Each tab shows the same information:

- The unique code for the alert
- The type and the message wording

- Whether the alert is visible. On the active tab, you can select this icon to change the alert between visible and invisible.

- The node and associate for this alert. The association identifies the target of the alert, for example, a channel.

You can choose the **Alert Filters** button at the top right corner to filter alerts.

Status - Messages page

On the Conductor Live main menu, choose **Status**. Then choose **Messages** in the left panel.

The page shows the following information:

- The unique code for the message.
- The type. Messages are error messages, warning messages, and audit messages.
- Error messages have red shading and a red triangle icon. Only error messages are included in the message count on the web interface header.
- The message wording.

- Whether the message is visible. You can select this icon on an error message, to change the message between visible and invisible.

- The node and associate for this message. The association identifies the target of the message, for example, a channel.

You can choose the **Message Filters** button at the top right corner to filter messages.

Monitoring nodes

You should monitor the nodes regularly to ensure that they are still all online.

1. On the Conductor Live main menu, choose **Status**, then choose the **Overview** tab.

   This page shows a summary of the status of the nodes, channels, and MPTSes in the cluster.
2. If a node is shown as failed or offline, you can obtain more information. On the Conductor Live main menu, choose **Cluster**, then choose **Nodes**.
3. To identify the problem node or nodes, look for nodes that have a red or yellow background and an orange icon in the Status column.

4. Choose an orange icon to go to the **Status** - **Alerts & Messages** page to display detailed information about the alerts and messages. The **Alerts & Message**s page appears with the filter set to show only the information for this channel.
5. Review the alerts and messages to determine why the channel failed.
6. For detailed information on dealing with problems, see the following topics.

Topics

- Offline nodes
- Failed worker nodes with worker redundancy
- Failed worker nodes without worker redundancy
- Failed Conductor Live nodes with Conductor Live redundancy
- Failed Conductor Live nodes without Conductor Live redundancy

Offline nodes

Investigate an offline node if you were not expecting nodes to be offline. Try to determine why the node has been taken offline (speak to other engineers and operators) and, if necessary, take steps to bring the node back online.

Failed worker nodes with worker redundancy

When worker redundancy is implemented on the cluster and a node switches to the failed status, any channels that are running on the worker node move to a backup node, as described in “How Worker Node Failover Occurs” below.

Setting up for Notification

We recommend that you set up Conductor Live so that it sends you an email or hits your webserver when the following alerts or messages are raised:

- 4009
- 4010
- 4018

See the AWS Elemental Conductor Live Configuration Guide for information on setting up notifications.

Dealing with a failed node

When a node goes to failed, follow this procedure to deal with the failed node and with the redundancy setup.

1. Go to the **Cluster** - **Redundancy** page and look for the redundancy group that the failed node belongs to: choose each group in the **Redundancy Groups** section and look for the node in the **Active Nodes** tab and the **Backup Nodes** tab.

   If the node appears in the **Backup Nodes** tab, see *If a Reserve Node Fails*, below. Otherwise, continue this procedure.
2. Verify if there is still at least one node listed in the Backup tab.
   - If yes, then there is no immediate need to deal with the failed node, but you should still deal with it in a timely manner.
   - If not, you can assume that when the failed node failed over, it used up the last of your backup nodes. You should solve the problem on the failed node as soon as possible and bring it back into service so that you can get back to the state of having at least one backup node.

     You receive an alert if you have a redundancy group set up but do not have any backup nodes available.
3. To investigate the failed node (either now or later):
   - Go to the **Status** - **Nodes** page. The node should have an orange icon in the **Status** column. Choose this icon; the **Status** - **Alerts & Messages** page appears, filtered to show only the information for that node.
   - Review the alerts and messages to determine why the node failed.
4. Make sure you have the desired number of backup nodes set up.

How worker node failover occurs

1. Conductor Live determines the action to attempt:
   - If the node was online/idle before it failed, Conductor Live takes no fail over action. The node simply goes to the failed status.
   - If the node was online/running, Conductor Live attempts to failover this node to one of the reserve nodes, as described in the following steps.
2. Conductor Live identifies the redundancy group that the failed node belongs to and selects a reserve node in that group.

3. Conductor Live then attempts to move all channels (in the case of a failed Elemental Live node) or MPTSes (in the case of a failed Elemental Statmux node) to node_Y and restart the previously running channels or MPTS outputs on this new node. The role for node_Y changes from **reserve** to **active**. This node is no longer eligible to be selected as a failover node if another active node fails.

If a reserve node fails

If a reserve node fails when it is currently in reserve, it stays as a reserve node but its status changes to **offline**.

If a reserve node switches to **active** and then fails, it will be eligible to fail over to another reserve, in the same way as any other active node is eligible.

When a failed node recovers

When a node that is failed is brought back into service, it returns to the status it had when it failed: Active or Backup.

Dealing with a false failure

Conductor Live may determine that node_X has failed, when in fact it has only become disconnected from the management network (and is continuing to run channels) but has not shut down.

Meanwhile, because Conductor Live has determined that a failure has occurred, it attempts to perform a fail over. The fail over routine does not include any attempt to stop the channels running on node_X. If the fail over succeeds, the channels are running on both node_X and the fail over node.

However, if the network connection is later re-established (so that Conductor Live can now view activity on node_X), Conductor Live attempts to shut down the channels or MPTSes that are running there.

If a node does not fail over

If a node fails but there is no reserve node ready to take over for it, the node remains active/offline. When the node problem is resolved and the node goes back online, it still has its original channels. Channels that were running before the failure start running again.

Monitoring the distribution of nodes in a redundancy group

After a fail over, you should check the state of the redundancy group and take steps to ensure that the distribution of active nodes to reserve nodes matches the desired redundancy type (distribution of active versus backup nodes).

For example, you need to make sure that there is always at least one reserve node in each redundancy group. Each time a node fails, a reserve node switches to **active**. It is possible for all nodes to become active, in which case you need to re-assign at least one node to reserve in order to be prepared for a possible new fail over.

On the Redundancy page, make sure that the Redundancy type has a non-zero number as the second number:

Redundancy Status Alert

Alerts are raised if a redundancy group has one or more active, online nodes but has no backup, online nodes. The alert persists until a node is restored to a backup role, or a node without channels is manually moved to a backup role.

For more information about alerts and messages, see the section called “Monitoring alerts and messages”.

Failed worker nodes without worker redundancy

When worker redundancy is not implemented on the cluster and a worker node has failed, you must do the following:

- Determine if failure of the node has caused channels to fail and then take steps to re-start those failed channels.
- Deal with the problem node.

To troubleshoot nodes

1. Go to the **Channels** page and determine if any channels have failed. If they have, then move the channels to other nodes as soon as possible. See the section called “Modify channel” and change the associated node.

2. Go to the **Status** - **Nodes** page. The node should have an orange icon in the **Status** column. Choose this icon; the **Status** - **Alerts & Messages** page appears, filtered to show only the information for that node.
3. Review the alerts and messages to determine why the node failed.
4. Take the necessary steps to resolve the problem and bring the node back into service.

Failed Conductor Live nodes with Conductor Live redundancy

When you have redundant Conductor Live nodes set up, then, when the primary fails, the backup automatically takes over management of the cluster. This change in role takes a few seconds.

After failover has occurred, the two Conductor Live nodes continue in their new roles.

Even when the failed Conductor Live comes back online, it does not take back the leader role: it comes back into service as the backup.

> **Note**
>
> Typically, there is no reason for you to switch back to the node that was previously the primary: there is no need to prefer one Conductor node as the primary for the other. In fact, there is no mechanism for you to switch the node roles.

Failed Conductor Live nodes without Conductor Live redundancy

When your cluster has only one Conductor Live node, then when it fails, you are not able to use Conductor Live to control worker nodes. The worker nodes are not affected by the Conductor Live node failure.

To troubleshoot a Conductor Live node

1. Go to the **Status** - **Nodes** page. The Conductor Live node should have an orange icon in the **Status** column. Choose this icon; the **Status** - **Alerts & Messages** page appears, filtered to show only the information for that node.
2. Review the alerts and messages to determine why the node failed.
3. Take the necessary steps to resolve the problem and bring the node back into service.

Returning a Node from Failure

When the Conductor Live node comes back online, it automatically takes over management of the cluster again. It brings itself up to date in terms of activity and status of all the nodes:

- If an alert was active when the node failed and the problem no longer exists, the alert is automatically cleared.
- If an alert was active and the problem still exists, the alert is not cleared.
- If a problem occurred on a worker node while the Conductor Live node was offline, the Conductor Live now detects this problem and displays a new alert or message.
- If a problem occurred and got resolved on a worker node while the Conductor node was offline, the Conductor Live has no knowledge of that problem ever having existed. This is really the only missing information from the outage.

Monitoring the load on worker nodes

You can view information about the overall load on any worker node in an AWS Elemental Conductor Live cluster.

On the **Nodes** page, choose the hostname of the node. (Don't choose the IP Address. Doing so will open the web interface for that node in another tab.)

The **Node Details** page appears for that node, showing these charts:

- Bandwidth
- CPU Utilization
- Disk
- GPU Frames/Second (if the node is running GPU-enabled software)
- GPU Temperature (if the node is running GPU-enabled software)
- Memory
- Realtime
- Total Frames/Second

Resources: Profiles

A *profile* is a pre-defined collection of values for most of the data in an AWS Elemental Live channel. When you use AWS Elemental Conductor Live to create a channel, you associate it with a profile in order to set up all that data.

In Conductor Live, profiles are required. To create a channel, you must use a profile.

This section describes how to use Conductor Live to create, modify or delete a profile.

Topics

- Creating a profile from scratch
- Working with channel parameters in a profile
- Creating a profile for an SPTS channel
- Creating a profile by duplicating an existing profile
- Creating a profile by uploading XML
- Modifying a profile
- Viewing the contents of a profile
- Deleting a profile

Creating a profile from scratch

Topics

- Design the profile
- Create the profile

Design the profile

When you create an AWS Elemental Live profile, you must enter a value for every field that you want to be useable by the channel.

A created profile will contain all the fields where you entered a value (or a checkmark, for example) or where you set up a profile parameter. All the other fields are deleted from the profile. The following rules apply:

- You won't be able to add fields from the profile back in.
- You won't be able to delete fields that are in the profile.
- You won't be able to modify or specify a value for any fields, except in fields that have profile parameters.
- You won't be able to leave a profile parameter empty.

Therefore, you must plan the profile fields carefully:

- Some fields are required. You must either provide a value, or accept the default.

  If you provide a value, it must be the permanent value, or you must set it up as a profile parameter (if that's possible).

  If you don't like the default, make sure that you change it, because you won't be able to change it when you use the profile to create the channel.

- Some fields are optional. If you want to include the field, you must either provide a value or set up the field as a profile parameter (if that's possible).
- Pay particular attention to the output groups section. Make sure that you include all the output groups, outputs within those groups, and streams (encodes) within all the outputs.

  You won't be able to add or delete output groups, outputs, or streams. Therefore, for example, make sure that you create all the audio encodes that you need in a specific output. For example, if you know that you will sometimes need two audio encodes and sometimes three audio encodes, you must create two profiles.

Create the profile

1. On the AWS Elemental Conductor Live main menu, choose **Profiles**.
2. On the **Profiles** page, choose **New Profile**. The **New Profile** page appears.
3. Complete the profile. You can accept all the defaults, but you must at least complete the following:
   - **Name**: Enter a profile name.
   - **Restart on Failure**: Check to restart the channel automatically if it fails (recommended).

- **Add Input**: Provide information for at least one video input: Specify the type , and the source location.

  Note the following:

  - If you do not want to specify the input source in the profile, you can specify a "channel parameter" as a placeholder that you replace with real information when you create the channel. See the section called "Work with channel parameters".
  - If your input is an SDI Direct Input, see the section called "Use case: SDI direct input" for important information.
  - If your input is an SDI Router Input, see the section called "Use case: SDI router input" for important information.
- **Output Group**: A channel must always contain at least one output group. Set up the output group with all its contents and features.
- **Output** and **Stream**: A channel must always contain at least one output and one stream in each output group that you want to create. Each output must use one stream.

4. Choose **Save** to save the profile. The profile appears in the list on the **Profiles** page.

Working with channel parameters in a profile

You can create an AWS Elemental Live profile in which some fields have values that are variable rather than absolute. Then when you create a channel using that profile, you assign real values to those fields. In AWS Elemental Conductor Live, these variable fields are called *channel parameters*.

Setting up a profile in this way makes the profile more flexible. You can use it with multiple channels simply by entering different values in the channel parameters.

Fields that can be set up with parameters have a blue background in the web interface. For a list of these fields, go to the AWS Elemental User Community and search for *supported channel parameters*. You will definitely need this list if you create profiles using the REST API.

Topics

- Rules for channel parameters
- The procedure
- Planning ahead for bulk changes
- Use case: Using SDI direct input in a profile and channel

- Use case: Using SDI router input in a profile and channel

Rules for channel parameters

There are two rules associated with channel parameters in the profile:

- Generally, if a field is blue on the profile, setting up as a channel parameter is optional. On the profile, you could also enter a permanent value or no value.

  There are two exceptions to this rule: the SDI Direct Input field and the SDI Router Input field must be set up as channel parameters. See the section called “Use case: SDI direct input” and the section called “Use case: SDI router input”.
- You shouldn't set up non-blue fields with a parameter. If you try to do so, you receive an error when you save the profile.

> **Warning**
>
> It is possible to create profiles with parameters in fields that aren't blue. But doing so can cause problems when you create channels from these profiles or when you import profiles after upgrades.

The procedure

Follow this procedure to set up a blue field with a channel parameter. This procedure assumes that you know how to create a profile and a channel.

1. Position your cursor in a blue field. The **Parameters** panel opens on the page.
2. For the value of the field, enter a name for the parameter in double curly brackets:

   `{{my_parameter}}`

   You can set up the field as a mixture of hardcoded value and parameter value. For example:

   `udp://10.20.110.106:{{port}}`

   In the above example, you've created a parameter called **`port`**.

3. As soon as you enter a correctly formatted name, the parameter appears in the **Parameters** panel.
4. In the **Parameters** panel, enter a validation value for the parameter. The validation value is a sample value that is correctly formatted.

   When you create a channel using this profile, you will enter a real value for

   For example, assume that the field always takes a URI and that the protocol must be RTP or UDP. A sample value could look like any of the following:

   `rtp://hostname.com:1234`

   `rtp://1.2.3.4.com:1234`

   `udp://hostname.com:1234`

   Note the following about these sample validation values:

   - **`rtp://`** is a real value and is correctly formatted.
   - **`hostname.com`** is not a real value, but it is correctly formatted.
   - **`1234`** is not a real value, but it is correctly formatted.
5. When you save the profile, Elemental Live verifies that the validation value you've entered is valid.

Planning ahead for bulk changes

When you create a channel parameter that you might use in another profile, note its name carefully so you can enter the exact same name in the other profile.

This careful planning will help if you ever want to make a bulk change to switch several channels to a different profile (the section called “Change profile for multiple channels”).

Example of good planning

You create profile_A that has one channel parameter called {{input_network_location}}. You create profile_B that has one channel parameter called {{input_network_location}}. You assign each profile to one channel: channel_1 and channel_2. You later want to use the task feature to change the profile of channel_1 and channel_2 to use profile_C.

The two profiles have use the same channel parameter name for the same field. You will be able to switch both channels to profile C without a problem.

Example of bad planning

This example is the same as the previous example except that you name the channel parameters differently:

- Profile_A uses the name {{input_network_location}}.
- Profile_B uses the name {{input_nw_location}} for the channel parameter for the same field.

You won't be able to change the profiles without making adjustments.

Use case: Using SDI direct input in a profile and channel

You might want to specify an SDI direct input in your profile and channel. When you choose this type of input, two supplementary fields appear—the **Device ID** field and the optional **Device Settings**.

The **Device ID** field must be set up as a channel parameter. You can't enter a real ID.

To set up this field, follow this procedure. This procedure assumes that you have already read the section called “Work with channel parameters”.

1. When you create the profile, complete the following fields in this way:
   - **Input Type**: Choose **SDI Direct Input**.
   - **Device ID**: For the value, create a channel parameter as described in the section the section called “Work with channel parameters”.

     Assign any name, but do not create a name consisting only of a number. For example, create a channel parameter with the name **`SDI_direct_input`**.

     In the **Parameters** panel, leave the validation value empty. A validation value is not required for the **Device ID** field.
   - **Device Settings**: Always leave this field empty.
2. When you create the channel, complete the following fields in this way:
   - **Node**: Choose a node that has an SDI direct input attached to it.
   - **SDI Direct Input Device Id**: Note that the field name appears with this label. It doesn't appear with the label **Device ID**.

From the list, choose one of the SDI interfaces that is attached to the node that you chose.

- **Device Settings**: Choose a value, if one is applicable to this input.

Use case: Using SDI router input in a profile and channel

You might want to specify an SDI router input in your profile and channel. This type of input goes through a router, and is only applicable if there is at least one node in the cluster that is configured with a router. When you choose this type of input, a supplementary field appears—the **Router Input ID** field.

The **Router Input ID** field must be set up as a channel parameter. You can't enter a real ID.

To set up this field, follow this procedure. This procedure assumes that you have already read the section called “Work with channel parameters”.

1. When you create the profile, complete the following fields in this way:
   - **Input Type**: Choose **SDI Router Input**.
   - **Router Input ID**: For the value, create a channel parameter as described in the section the section called “Work with channel parameters”.

     Assign any name, but do not create a name consisting only of a number. For example, create a channel parameter with the name **`SDI_router_ID`**.

     In the **Parameters** panel, enter a validation value.
2. When you create the channel, complete the following fields in this way:
   - **Node**: Choose a node that has an SDI router attached to it.
   - **SDI Router Input Router Input ID**: Note that the field name appears with this label. It doesn't appear with the label **Router Input ID**.

     From the list, choose one of the inputs on the router. This router is attached to the node that you chose.

Creating a profile for an SPTS channel

An SPTS channel is a channel that you plan to use in an MPTS that you will run on AWS Elemental Statmux.

A profile for an SPTS channel must include a UDP/TS output group that has one output that is set up for statmux. The UDP/TS output group can also include non-statmux outputs. The profile itself can include other types of output groups.

1. On the AWS Elemental Conductor Live main menu, choose **Profiles**. On the **Profiles** page, choose **New Profile** (on the top right corner of the page).
2. Create the profile in the usual way for everything except the UDP/TS output group.
3. To set up the UDP/TS output group, scroll down to **Output Groups** and choose the **UDP/TS** tab.
4. In the **New Output** section, choose the **Add Output** button.
5. For **MPTS membership**, choose **Remote**. This output section changes to display different fields. Complete the fields as follows:
   - Note that the destination fields are removed because you set the destination in the MPTS, not in the channel profile.
   - Note that the **PAT** and **PMT** fields are enabled because Elemental Statmux always creates these tables. You can set the interval to specify how often each table is inserted in the transport stream.
   - Set the **NIT** and **TDT** fields if you want to create these tables for this program.
   - Set the **SDT** fields to specify how you want Elemental Statmux to handle the program in this table. There is an option to not include this program in the SDT.
6. In the **Streams** section, in the **Video** section, choose **Advanced** to display more fields. Set the following field:

   Rate Control Mode: Set to **Statmux**. This value indicates that the muxer will control the rate control for the output.

   Note that when you set this value, the **Bitrate**, **Max Bitrate**, and **Min Bitrate** fields don't apply, so these fields become disabled.
7. Choose **Save**.

The profile that you created is listed on the **Profiles** page:

- Conductor Live assigns a unique numerical ID to the profile.
- Under the **Channels** column, the profile displays **0 0**. This indicates that the profile is not yet being used by any channels.

Creating a profile by duplicating an existing profile

You can create an AWS Elemental Live profile by duplicating an existing profile and making any changes that you want.

To duplicate a profile

1. On the AWS Elemental Conductor Live main menu, choose **Profiles**.
2. In the list of profiles, find the profile that you want to duplicate. Choose the **Duplicate** icon.

   The **Duplicate Profile** page appears. It shows the values from the source profile, including any channel parameters.
3. Change any fields as needed.

   You must enter validation values for all the parameters that you use. For more information, see the section called “Work with channel parameters”.
4. Choose **Save**.

Creating a profile by uploading XML

You can create an AWS Elemental Live profile using the XML from an existing profile that is up to two current versions back from the current AWS Elemental Conductor Live version. When the profile is uploaded, the Conductor Live system migrates the profile to the current version while also maintaining field selections and values.

1. If you haven’t already done so, download the existing profile and save it.
   - On the Conductor Live main menu, choose **Profiles**.
   - In the list of profiles, find the profile that you want to duplicate. Choose the **Download** icon. Conductor Live copies the profile to your local computer. The profile is an XML file.
2. On the **Profiles** page, choose **New Profile**.
3. Choose the **Upload XML** button. Choose **Browse** and navigate to the downloads folder on your computer. Open the XML file, then choose **Apply** to load the XML in the **New Profile** page.
4. Update fields as required and save the profile.

Modifying a profile

You can't modify a AWS Elemental Conductor Live profile after it has been created. This rule ensures the dependability of profiles: a channel that uses profile_A and that was run two weeks ago has the same profile data as a channel that uses profile_A that was run yesterday.

If you want to modify a profile so that you do not have to recreate it from scratch, take the following steps.

To modify a profile

1. Duplicate the profile. Change values as desired and then save the profile.
2. Take the appropriate action:

   - If you duplicated the profile in order to fix errors in a profile you only just created, delete the incorrect profile.
   - If you duplicated a profile that is being used by one or more channels, remember to associate the channels with the new profile. Then delete the unused profile.
     - To change the association for only one channel, see the section called “Modify channel”.
     - To change the association for several channels that use this profile, see the section called “Change profile for multiple channels”.

Viewing the contents of a profile

To view the contents of an AWS Elemental Live profile, choose **Show** beside the profile name. An abbreviated version of the Profile page appears: it shows all the fields that you explicitly completed, plus other key fields.

Deleting a profile

To delete a profile

1. Verify that the profile is not being used:

   - On the AWS Elemental Conductor Live main menu, choose **Profiles**. Look at the **Channels** column for this profile.

- If it specifies a number, display the **Channels** page and filter the list (if necessary) to show only channels that use this profile.
- Modify the channels to use a different profile.

2. Go back to the **Profiles** page and choose **Delete** next to the profile name. Then choose **OK**.

Resources: Elemental Live Channels

This section describes how to use AWS Elemental Conductor Live to create, modify or delete a channel.

A *channel* is a session that decodes and encodes a live video stream or a video file and produces a live output. Video input comes into the channel and video output is the final outcome of the channel. All the encoding activity occurs within a channel.

When you use Conductor Live, you create a channel by first selecting a profile. If you aren't familiar with profiles, first read the section called “Create profile from scratch”.

After you have selected the profile, you complete fields that were set up as profile parameters in the profile. When you save the channel, Conductor Live creates the channel using the following data:

- The values that you entered in the profile parameters in the channel.
- The values that were set up as permanent values in the profile.

You can't add or delete fields on the channel. Instead, you must use a profile that has all the fields and sections of fields that you need. Specifically, you can't add or delete output groups, outputs, or streams.

Topics

- Creating a channel
- Creating a channel by duplicating an existing channel
- Modifying a channel
- Changing the profile used by multiple channels
- Deleting channels

Creating a channel

To create a channel

1. Decide which profile you will use as the basis for the channel.
2. On the Conductor Live main menu, choose **Channels**.

3. On the **Channels** page, choose **New Channel**. The **New Channel** page appears.
4. Complete the fields. Take note of the following:

   - **Profile**: Choose the profile to base this channel on. When you choose a profile, the page changes to show all the profile parameters for that profile.

     The page doesn't show fields from the profile that you didn't set in the profile. For those fields, the default value applies.

     The pages doesn't show fields that you did set in the profile. For those fields, the value from the profile applies. You can't change the value locally in the channel.

   - **Node**: You can choose a node now, or you can leave this field empty and assign a node later. Make sure that the node is appropriate for the density of the channel. Make sure that the node has all the licenses required for the channel. For example, make sure that it has the required codec licenses.

   - **Profile parameters**: Complete the profile parameters. Remember that some fields aren't required, even if you created a profile parameter for the field.

     For example, there might be a profile parameter for an interface. The interface field might be optional. In this field, you can enter a value. Or you can leave the field empty — Elemental Live will use the default interface.
5. Choose **Save**. The channel is created. If you specified all the parameters, then the channel is ready to run: see the section called “Start and stopping a channel”.

Creating a channel by duplicating an existing channel

To create a channel by duplicating an existing channel

1. On the Conductor Live main menu, choose **Channels**.
2. In the list of channels, find the channel that you want to duplicate. Choose the **Duplicate** icon.

   The **Duplicate Channel** page appears. It shows the values from the source channel, including any channel parameters and their current values.
3. Make changes and choose **Save**.

Modifying a channel

You can modify a channel that is in the Idle or Error state (that is, when it is not actively running).

1. On the Conductor Live main menu, choose **Channels**.
2. On the Edit Channel page, make any changes:
   - **Name**: You can change the name.
   - **Profile**: You can associate the channel with a different profile. Complete any new channel parameter that is associated with a required field.
   - **Node**: You can associate the channel with a node for the first time, with a different node, or with no node.
   - **Channel parameters**: Change the value for any parameter.

Changing the profile used by multiple channels

You can use the **Tasks** feature to change the profile so that several channels can use it. You can also change associations between channels and profiles so that each channel uses a different profile.

You can modify a channel even if it is running.

Step 1. Create the task

To create a task

1. On the **Channels** page, select **Tasks** in the top left corner, and choose **Change Channel Profiles**.

   The **Change Channel Profiles** page shows all the channels on all the nodes in the cluster.
2. Select the channels that you want to associate with one specific profile.

   You can select by choosing individual channels.

   You can select by filtering. For example, you can filter by profile or by node. After you have filtered, choose the **Select All** button.
3. Choose **Next**.
4. Select the new profile to associate with all the selected channels. Choose **Next**.

5. If the new profile includes channel parameters, complete those parameters. When you are ready, choose **Next**.

Examples of changes in channel parameters

Original Profile for a Channel	New Profile for a Channel	Action
Three channel parameters	Two channel parameters, mapped to the same two fields as the original profile. A third parameter channel which does not exist in the new profile.	None. For the two parameter channels that still exist, the values already in the channel are preserved and used. For the channel parameter that has been dropped, a value is no longer required in the channel; the value that is “hard-coded” (or implied, that is, a system default) in the new profile is used.
Three channel parameters	Four total channel parameters: • Three channel parameter s, mapped to the same three fields as the original profile. • One new channel parameter.	You are prompted to provide a value for the new channel parameter.

Original Profile for a Channel	New Profile for a Channel	Action
One channel parameter, for example, {{input_network_location}}	A channel parameter, mapped to the same field as the original profile. This channel parameter, however, has a different name. For example {{input_nw_location}}	You are prompted to provide a value for the new channel parameter.
One channel parameter, for example, {{input_network_location}}	The same channel parameter, but mapped to a different field.	Conductor Live treats the channel parameter as a new parameter. You are prompted to provide a value for the new channel parameter. You should try to avoid this problem by tying each channel parameter name to a specific field. Avoid generic names such as {{field 1}}

You might get any combination of these types of discrepancies, and you are prompted to handle all of them.

6. On the final page, choose **Process Now** or **Save for Later**.

   Process Now: Conductor Live applies the change. The **Channels** page reappears, showing the change.

   Save for Later: This option lets you queue up several tasks and then perform them in one pass.

   > **Warning**
   >
   > **Save for Later** is intended to queue for a short time.

Don't use **Save for Later** and then delay process the task in a few hours. Doing so might create undesired consequences.

Example of Save for Later

Assume that you want to move all existing channels from node A so that you can move another set of channels to node A. This is an opportunity to use bulk changes. Set up bulk changes as follows: do Bulk Change #1 to move the channels from node A to node B (change the associated node for the channels). Then do Bulk Change #2 to move the channels from node C to node A.

In order to do this two-part move with the least downtime, you could do the following:

- Set up Bulk Change #1 and choose **Save for Later**.
- Set up Bulk Change #2 and choose **Save for Later**.

  Both bulk changes appear on the left side of the page, as shown below.
- Then choose **Run Pending Actions**. Bulk Change #1 runs. As soon as it has finished, Bulk Change #2 runs.

Step B. Monitor the status of the task

When you run **Process Now** to run the task, you can monitor the status of each modify action in the task.

1. On the Conductor Live main menu, choose **Status**, then choose **Task Reports**. The **Task Reports** page appears.
2. In the **Task Reports** panel on the left, choose the task to monitor (newer tasks appear first).

   (This list shows running and completed tasks; for pending tasks, go to the **Channels** page and choose the **Pending Tasks** icon).
3. Find the relevant task and review the information:
   - The top line shows the count of total tasks, failed tasks, and successful tasks.
   - Below that, a line appears for each individual action. Each line shows the current status of the action: **Pending**, **Success**, or an explanation of why the task failed.

As the status of an action changes, the page automatically refreshes to display the latest information.

Deleting channels

Deleting one channel

You can delete a channel if it isn't running.

1. On the Conductor Live main menu, choose **Channels**.
2. On the **Channels** page, choose the **Delete** icon beside the channel. The channel is deleted immediately.

Deleting several channels at once

You can use the **Tasks** feature to delete several channels at once.

1. On the Conductor Live main menu, choose **Channels**.
2. On the **Channels** page, select **Tasks** on the top left. Then choose **Delete Channels**.
3. Select the channels that you want to delete.
4. Choose **Save for Later** or **Process Now**.

   Process Now: Conductor Live applies the change. The **Channels** page reappears, showing the change.

   Save for Later: This option lets you queue up several tasks and then perform them in one pass.

   > **Warning**
   >
   > **Save for Later** is intended to queue for a short time.
   > Don't use **Save for Later** and then delay process the task in a few hours. Doing so might create undesired consequences.

Resource: MPTS

A multi-program transport stream (*MPTS*) is a UDP transport stream (TS) that carries multiple programs. This section describes how to use AWS Elemental Conductor Live to create an MPTS and add channels (programs), and modify or delete an MPTS. For information about starting an MPTS and working with a running MPTS, see the section called "Starting or stopping an MPTS" and the section called "Monitoring MPTS outputs".

Topics

- Creating a standard MPTS
- Advanced tab – Suppressing generation of SI/PSI tables
- Including passthrough streams in an MPTS
- Including passthrough programs in an MPTS
- Modifying an MPTS
- Deleting an MPTS

Creating a standard MPTS

With Conductor Live there are two steps to creating an MPTS—you create the MPTS, then you add SPTS channels to it. You perform all these steps in Conductor Live. You don't perform these steps in Elemental Statmux.

For guidelines for creating an MPTS workflow, see *Workflows: Design an MPTS workflow*.

Topics

- Step 1: Create the MPTS
- Step 2. Add channels to the MPTS

Step 1: Create the MPTS

To create an MPTS

1. On the Conductor Live main menu, choose **MPTS**. Then choose **New MPTS**. The **Create a New MPTS** dialog appears.

2. Complete the fields in the top section of the dialog. For information about a field, hover on the upper-right corner of the field and choose the **?** icon. Pay particular attention to the following fields.

Field	Description
Name	Any name.
Transport Stream ID	The PID for the transport stream in the MPTS.
Node	The node where you want the MPTS to run. The list shows all active nodes. It doesn't include backup nodes in a redundancy group
Transport Stream Bitrate	The total bitrate for the MPTS. All the SPTS channels in the MPTS will use a portion of this bitrate.

3. Complete the fields in the **Output** tab and the **Advanced** tab. For more information about the significant fields, see the sections after this procedure.
4. Choose **Save**.

   The MPTS appears in the list of MPTSes.

   If the node you selected is part of a 1-to-1 or 1-to-1 Plus redundancy group, Conductor Live automatically creates two instance of the MPTS. One instance is on the primary Elemental Statmux node, the other is on the secondary node. Both MPTS instances have the same downstream system destination or destinations.

   The next step in creating the MPTS is to add channels (programs).

   Other optional steps are adding passthrough streams, and adding passthrough programs.

Topics

- Output tab

Output tab

Field	Description
URI, Interface, Virtual Source Address	Required. The address to the destination on the downstream system. Conductor Live will deliver the MPTS to this address.
Output listening	Check this field if you want to enable output listening, which is a form of resiliency that protects against failures within the muxing pipeline. You can only enable output listening if you have set up Elemental Statmux for 1-to-1 node redundancy because the feature requires two nodes. For information about how this feature works when the MPTS is running, see the section called "Resiliency in Elemental Statmux".
Add Destination	Choose this button if you are implement ing statmux output redundancy. With this redundancy feature, you want the Elemental Statmux node to deliver the MPTS to two different downstream systems. You must specify the destination of the second downstream system. For information about how this feature works when the MPTS is running, see the section called "Resiliency in Elemental Statmux".

Step 2. Add channels to the MPTS

After you create the MPTS, you must add channels. The channels must already exist.

1. On the Conductor Live main menu, choose **MPTS**.

2. Select the MPTS by ID or by name. If the MPTS is listed twice, it has been set up for output listening. Select the MPTS that is marked as **Primary**.

   The **Details** page for the MPTS appear. The page has four tabs, and the **Channels** tab is currently selected.

3. On the **MPTS Details** page, select the **Channels** tab. Note that the area at the bottom of the page lists the channels in the MPTS. This list is currently empty.

4. Choose **Add a Channel** and choose from the list of channels that appears. To appear in this list, a channel must have these characteristics:

   - It must be set up as an SPTS channel with MPTS membership set to **Remote**.
   - It can't already be in another MPTS.
   - It can't already be assigned to an MPTS unless using CBR rate control.

     When a channel is using CBR for the Rate Control Mode, it can be used in unlimited MPTSes.

5. When you select the channel, it is added to the list of channels for this MPTS. The list has several tabs.

6. Complete the fields in the tabs as appropriate. For more information about the significant fields, see the sections after this procedure.

7. Choose **Save**.

The MPTS is now ready to start. See the section called “Starting or stopping an MPTS”.

Topics

- Basic tab
- PID Controls tab
- TS Endpoints tab
- Complexity Endpoints tab
- RateAllocation Endpoints tab

Basic tab

Field	Description
#	Read only, numerical ID for each program in this MPTS. Unique within this MPTS.
ID	Read only, numerical ID for each program. Must be unique among all MPTSes in this Conductor Live cluster.
Encoders	Read only, the number of Elemental Live encoders that are currently providing programs (channels) to this MPTS.
Outgoing Program Number	The PMT PID for this program. Must be unique within this MPTS. If you let Elemental Statmux assign this number for all the programs, then all the numbers are guaranteed to be unique. You can change this number, but you must make sure that the numbers are unique.
Service Name, Provider Name	Enter this information if the source program doesn't include it. Or leave the field empty to use the information from the source program. Or leave the field empty if you don't want this program to appear in the SDT for the MPTS.
Quality Priority Offset	Complete this field in one or more programs to assign relative priorities to the programs.
Min and Max Bitrate	Typically leave these fields empty. Elemental Statmux will assign values for all the programs when you save the MPTS.

PID Controls tab

On this tab, you can assign the output PIDs for the streams that are in the source program. You can leave one, several or all the fields empty, and Elemental Statmux will assign a unique ID.

If you do enter a PID, you must make sure that each PID is unique in the entire MPTS. You should also assign PIDs only to streams that you know are in the stream, otherwise you are wasting a number.

If you leave the fields empty, Elemental Statmux assigns PIDs after you save. It makes sure that each stream has a PID that is unique in the entire MPTS.

TS Endpoints tab

This tab contains fields for the endpoints used for communications between the Elemental Live node that is the source of the program and the Elemental Statmux node that receives the program.

Typically, there is no need to set these fields manually. Leave the fields empty so that Elemental Statmux can assign values when you save the MPTS. These fields are made available only for troubleshooting communications problems.

Complexity Endpoints tab

This tab contains fields for the endpoints used for the complexity/allocation communications between the Elemental Live node and the Elemental Statmux node.

Typically, there is no need to set these fields manually. Leave the fields empty so that Elemental Statmux can assign values when you save the MPTS. These fields are made available for troubleshooting communications problems.

RateAllocation Endpoints tab

This tab contains fields for the endpoints used for the complexity/allocation communications between the Elemental Live node and the Elemental Statmux node.

Typically, there is no need to set these fields manually. Leave the fields empty so that Elemental Statmux can assign values when you save the MPTS. These fields are made available for troubleshooting communications problems.

Advanced tab – Suppressing generation of SI/PSI tables

Many of the fields on this tab let you control generation of all the SI/PSI tables that Elemental Statmux can generate. For guidance for disabling generation of these tables, see the section called "Passing through SI/PSI tables".

Field	Description
Suppress PMT Generation	Check this field if you don't want Elemental Statmux to generate PMTs for any of the programs in the MPTS. For a standard MPTS, you leave this field unchecked.
PAT Interval	If you want Elemental Statmux to generate this table for the MPTS, check the field and set the interval. For a standard MPTS, you check this field.
SDT Interval	If you want Elemental Statmux to generate this table for the MPTS, check the field and set the interval. Elemental Statmux creates the SDT table. For a standard MPTS, you check this field. For each program in the MPTS, Elemental Statmux creates the SDT information as follows: • If you entered values for the service name and provider name on the **Basic** tab for each channel, Elemental Statmux uses that data. • If you didn't enter values, Elemental Statmux uses the data passed from the source program. • If there is no data from either location, the information is left empty for that program.

Field	Description
TDT Interval	If you want Elemental Statmux to generate this table for the MPTS, check the field and set the interval. For a standard MPTS, this field is optional.
Enable NIT Information Table (NIT)	If you want Elemental Statmux to generate this table for the MPTS, check the field and complete the data fields. For a standard MPTS, this field is optional.

Including passthrough streams in an MPTS

Read this section if you want to include passthrough streams in the MPTS. For guidelines and rules about the streams that you can pass through, see the section called “Passing through custom streams” and the section called “Passing through SI/PSI tables”

To pass through a stream, you instruct Elemental Statmux to extract a specific PID from a *source MPTS* that is provided by the upstream system. Elemental Statmux. The PID must identify a single stream, not a program.

Get ready

To pass through one or more streams, you need the following information:

- The source MPTS locations: The location where each upstream system source will push each source MPTS. The upstream system can decide on each location and provide it to you. Or you can let Elemental Statmux create a multicast address and then you can provide the upstream system with that location.

- The PID of each stream that you want to extract from each source MPTS.

The procedure

You can add passthrough streams while you are creating the MPTS or after you have created it.

Follow this procedure to extract one or more streams from the same source MPTS:

1. On the **MPTS Details** page where you want to add the streams, display the **Passthrough Streams** tab.
2. Select the **Add Passthrough Stream** button. The **Add Passthrough Stream** dialog appears. Complete the fields:
   - **Name**: Give the program a nickname for your internal use. This name doesn't appear in the outputs MPTS.
   - **PID Controls – Incoming**: Identify the PID that you want to extract from the program. This PID must be discoverable — it must be included in a PMT that is included in the PAT for the source MPTS.
   - **PID Controls – Outgoing**: The PID to assign to the stream in the output MPTS.

     You must specify this PID, which means it must be unique among all the PIDs that you are manually specifying in the entire MPTS. You don't need to worry about it being unique among the PIDs that Elemental Statmux will assign because Elemental Statmux will always work around the PIDs you've assigned manually.
3. Select the **Add PID Mapping** button if you want to extract another stream from the same source MPTS.
4. Select the **Stream Endpoints** tab: Complete these fields if the upstream system has provided you with the location of the source MPTS.
5. Choose **Done**. One new entry appears in the list of passthrough streams in the MPTS.
6. Choose **Save**.

Review the information:

- Select the **Stream Endpoints** tab. This tab shows the location of the source MPTS:
  - If you completed the location fields the dialog, this tab shows the information you entered.
  - If you left the location fields empty on the dialog, Elemental Statmux has automatically generated a primary and backup multicast address. Give this information to your contact at the upstream system so that they can push the source MPTS to that location.
- Select the **PID Controls** tab. All the PIDs that you entered on the dialog appear in the same entry.

Including passthrough programs in an MPTS

Read this section if you want to include passthrough programs in the MPTS. For guidelines and rules about including these programs in an MPTS, see the section called “Including passthrough programs”.

To pass through a program, you instruct Elemental Statmux to extract a specific program from a *source MPTS* that is provided by the upstream system.

Get ready

To pass through one or more programs, you need the following information:

- The source MPTS locations: The location where each upstream system source will push each source MPTS. The upstream system can decide on each location and provide it to you. Or you can let Elemental Statmux create a multicast address and then you can provide the upstream system with that location.

- The PID of each program that you want to extract from each source MPTS.

The procedure

You can add passthrough programs while you are creating the MPTS or after you have created it.

Follow this procedure to extract one or more programs from the same source MPTS:

1. On the **MPTS Details** page where you want to add
2. Select the **Add Program Selector** button if you want to extract another program from the same source MPTS.
3. the programs, display the **Passthrough Programs** tab.
4. Select the **Add Passthrough Program** button. The **Add Passthrough Program** dialog appears. Complete the fields:
   - **Name**: Give the program a nickname for your internal use. This name doesn't appear in the outputs MPTS.
   - **Stream Endpoints** tab: Complete these fields if the upstream system has provided you with the location of the source MPTS.
5. Choose **Done**. One new entry appears in the list of passthrough programs in the MPTS. The entry has information over three tabs.

6. Select the **Basic** tab. On this tab, you must identify the following:
   - The program that you want to extract from the source MPTS. Complete the Incoming Program Number.
   - The PID for this program in the output MPTS. Complete the Outgoing Program Number.
   - The optional data to extract from the program: Complete Provider Name, Service Name, or Service Descriptor Passthrough for this program in the output MPTS. Or leave these three fields empty, to omit this information from the output MPTS.
7. Still on the **Basic** tab, if you want to extract another program from the same MPTS, select the Add Program Selector button on the far right. Complete the fields.
8. Select the **Stream Endpoints** tab. This tab shows the location of the source MPTS:
   - If you completed the location fields the dialog, this tab shows the information you entered. You can change any of the fields.
   - If you want Elemental Statmux to generate a primary and backup multicast address, leave these fields empty.
9. Select the **PID Controls** tab. All the programs that you entered in the **Basic** tab appear. This tab lets you remap the stream PIDs that are in the source program, so that the streams have different PIDs in the output program.

   A field appears for every stream type that Elemental Statmux supports.

   If you leave a field empty,
10. Select the **Add Program Selector** button if you want to extract another program from the same source MPTS.
11. When you've added all the programs, choose **Save**.

Review the information:

- Select the **Stream Endpoints** tab. This tab shows the location of the source MPTS:
  - If you completed the location fields the dialog, this tab shows the information you entered.
  - If you left those fields empty, Elemental Statmux has automatically generated a primary and backup multicast address. Give this information to your contact at the upstream system so that they can push the source MPTS to that location.

Modifying an MPTS

1. On the Conductor Live main menu, choose **MPTS**.
2. Look for the MPTS you want. If the MPTS is listed twice, it has been set up for output listening. Find the MPTS that is marked as primary, and select it by ID or by name.

   The **Details** page for the MPTS appear.

Modify the properties of the MPTS

You can change any of the fields on this tab even if the MPTS is running. For information on the fields, see the section called "Create MPTS". The change takes effect immediately. For example, if you change the transport stream bitrate, the muxer starts using the new value immediately.

1. Choose the **Configuration** tab.
2. Change any fields and choose **Save**.

Add or delete channels

You can add or delete channels, even if the MPTS has been started and is running.

1. Choose the **Channels** tab.
2. Choose the action:
   - To add a channel, choose **Add a Channel**.
   - To delete a channel, select the **Delete** icon to the right of the channel.

Change the properties of one or more channels

You can modify the properties of any channel, even if the MPTS has been started and is running.

1. Choose the **Channels** tab.
2. Choose **Edit Channels**.
3. Select the tab, then change the fields for one or more channels.
4. Choose **Save**. The changes take effect immediately, on all affected channels in the MPTS.

Deleting an MPTS

1. On the Conductor Live main menu, choose **MPTS**.
2. Look for the MPTS you want. If the MPTS is listed twice, it has been set up for output listening. Find the MPTS that is marked as Primary, and select it by ID or by name.
3. If necessary, select **Stop** to stop the MPTS.
4. Select the **Delete** icon to the right of the MPTS.

Resources: Using the Conductor Live tasks feature

The tasks feature of AWS Elemental Conductor Live lets you perform the following batch actions:

- Change several channels so that they all the same specified profile. See the section called "Change profile for multiple channels".
- Start several channels at once or stop several channels at once. See the section called "Starting or stopping several channels at one time".
- Delete several channels. See the section called "Delete channel".

Reference: Supported channel parameters

In an AWS Elemental Liveprofile, you can create set up some fields as channel parameters. When a field has a channel parameter, its value is a variables instead of a real value.

For information about setting up profile fields with channel parameters, see the section called “Work with channel parameters”.

For a list of fields that support channel parameters, go to the AWS Elemental User Community and search for *supported channel parameters*. You will definitely need this list if you create profiles using the REST API.

Document History for AWS Elemental Conductor Live

The following table describes important changes to this documentation.

- **API version:** 3.22.0 and later
- **Release notes:** current Release Notes

The following table describes the documentation for this release of AWS Elemental Conductor Live. For notification about updates to this documentation, you can subscribe to an RSS feed.

Change	Description	Date
Section on planning resiliency	The guide now includes information about planning for resiliency in a cluster. The information node redundancy for AWS Elemental Conductor Live, AWS Elemental Liveand AWS Elemental Statmux nodes, and resiliency features available in AWS Elemental Statmux.	April 12, 2022
Section on designing a cluster	The guide now includes information about designing a AWS Elemental Conductor Live cluster so that it contains sufficient nodes to handle your encoding and statmuxin g requirements.	April 12, 2022
Passing through programs	The guide now includes information about including third-party programs in an Elemental Statmux MPTS. This program passthrou	April 12, 2022

	gh feature was introduced in version AWS Elemental Statmux 2.23.4 (Conductor Live version 3.23.4).	
Passing through custom streams	The guide now includes information about including custom streams in an Elemental Statmux MPTS. This stream passthrough feature was introduced in version AWS Elemental Statmux 2.22.1 (Conductor Live version 3.22.1).	April 12, 2022
Passing through custom SI/PSI tables	The guide now includes information about passing through SI/PSI tables for inclusion in an Elemental Statmux MPTS. These tables are tables that have been created outside of AWS Elemental Statmux.	April 12, 2022
Designing an MPTS workflow	This section has been revised to better describe the features available in the relaunch of AWS Elemental Statmux.	April 12, 2022
Major revision to the guide	We have made a major revision to the guide to bring it up to date with the latest features, particularly with the new launch of AWS Elemental Statmux.	April 11, 2022

Cross-version release of the guide	This guide has been modified so that it isn't for a specific version of Elemental Live. The configuration procedure doesn't change from version to version.	November 11, 2021
Cross-version release of the guide	This guide has been modified so that it isn't for a specific version of Elemental Live. The configuration procedure doesn't change from version to version.	November 11, 2021
New section on rules and limits	The guide now includes a section that summarizes the rules and limits that apply to AWS Elemental Conductor Live. More information will be added to this section in the future.	September 28, 2021
Version 3.22 release	First release of the 3.22 software version.	February 6, 2021
Version 3.22 release	First release of the 3.22 software version.	February 6, 2021

Note

- AWS Elemental Conductor Live is not designed or intended for use with applications or in situations requiring fail-safe performance, such as life safety operations, navigation or communication systems, air traffic control, or life support machines in which the unavailability, interruption or failure of the services could lead to death, personal injury, property damage or environmental damage.

- A component of Conductor Live is licensed under the AVC patent portfolio license for the personal and non-commercial use of a consumer to (i) encode video in compliance with the AVC standard ("AVC video") and/or (ii) decode AVC video that was encoded by a consumer engaged in a personal and non-commercial activity and/or was obtained from a video provider licensed to provide AVC video. No license is granted or shall be implied for any other use. A component of Conductor Live is licensed under the mpeg-4 patent portfolio license for the personal and non-commercial use of a consumer for (i) encoding video in compliance with the mpeg-4 visual standard (“mpeg-4 video”) and/or (ii) decoding mpeg-4 video that was encoded by a consumer engaged in a personal and non-commercial activity and/or was obtained from a video provider licensed to provide AVC video. No license is granted or shall be implied for any other use. Additional information may be obtained from MPEG-LA, LLC. See http://www.mpegla.com.
- Conductor Live may contain Dolby Digital and Dolby Digital Plus, which are protected under international and U.S. copyright laws as unpublished works. Dolby Digital and Dolby Digital Plus are confidential and proprietary to Dolby Laboratories. Their reproduction or disclosure, in whole or in part, or the production of derivative works therefrom without the express permission of Dolby Laboratories is prohibited. © Copyright 2003-2015 Dolby Laboratories. All rights reserved.