



\*\*\*Unable to locate subtitle\*\*\*

# Serverless 3D Data Optimization Pipelines on AWS



# Serverless 3D Data Optimization Pipelines on AWS: \*\*\*Unable to locate subtitle\*\*\*

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

<b>Serverless 3D Data Optimization Pipelines on AWS</b> .....	<b>1</b>
Abstract .....	1
<b>Introduction</b> .....	<b>2</b>
<b>Reference Architecture</b> .....	<b>3</b>
<b>Prerequisites</b> .....	<b>5</b>
<b>AWS CloudFormation Template</b> .....	<b>6</b>
<b>Automated Deployment</b> .....	<b>7</b>
<b>Launch the Stack</b> .....	<b>8</b>
<b>Understanding the Solution</b> .....	<b>10</b>
<b>Scaling</b> .....	<b>13</b>
<b>Costs</b> .....	<b>14</b>
Example cost .....	14
<b>Troubleshooting</b> .....	<b>16</b>
AWS Batch .....	16
Amazon CloudWatch Logs .....	16
<b>Cleaning up</b> .....	<b>17</b>
<b>Contributors</b> .....	<b>19</b>
<b>Further Reading</b> .....	<b>20</b>
<b>Document Revisions</b> .....	<b>21</b>
<b>Appendix A: Subscribing to the PiXYZ Scenario Processor</b> .....	<b>22</b>
<b>Notices</b> .....	<b>25</b>

# Serverless 3D Data Optimization Pipelines on AWS

Publication Date: **March 8, 2021**

## Abstract

This implementation guide describes an architecture for scalable 3D Data Optimization Pipelines on AWS using containers, and can be used with any containerized conversion software. An AWS Marketplace Container product, the [PiXYZ Scenario Processor](#), is used as a specific containerized conversion software example. This product enables customers to decimate and simplify 3D Computer Aided Design (CAD) models, and convert them to different industry-standard file formats.

The architecture is designed for horizontal scalability using AWS serverless technologies. Fully-managed AWS services automatically expand and shrink resources based on the customer's workload demands, so customers don't need to manage nor provision compute capacity or servers. This allows customers to control costs by only paying for the resources they use, and focus their attention on their 3D Data Optimization requirements instead of infrastructure management. More importantly, customers benefit from automatic scale-out features to handle peak volumes, parallelizing large conversion jobs across the AWS compute cloud. This guide shows how this reduces turn-around times from approximately 14 hours down to approximately 10 minutes.

# Introduction

Augmented Reality (AR), Virtual Reality (VR), and interactive 3D customers are showing an increased demand for interactive 3D applications across industries including Enterprise, Manufacturing, Education, Internet of Things, Aerospace and Automotive. Interactive 3D applications can include immersive training, visualization, and design review on mobile devices, AR headsets, and VR headsets.

Before customers can create an interactive 3D application, they need 3D model files to represent virtual objects at the right level of detail for target devices, such as tablets, laptops, or AR/VR headsets. The 3D files are sometimes created from scratch, but many industrial customers have access to Computer Aided Design (CAD) files for the objects in their applications. However, these CAD files are too detailed for interactive 3D, and need to be converted to file formats suitable for 3D application authoring tools such as Unity, Unreal, Maya, or Blender. The process of removing unnecessary detail is referred to as *decimation* or more generally *optimization*, and the process of converting file formats is referred to as *conversion*.

[PiXYZ Software](#) is an AWS Marketplace partner specializing in CAD file optimization and conversion. The PiXYZ Scenario Processor is a Docker container image available on the AWS Marketplace that enables customers to optimize and convert CAD files. Customers pay only for the conversion processing time they use and associated AWS resources. There are no upfront costs or commitments to get started. For more information, see [Appendix A: Subscribing to the PiXYZ Scenario Processor](#).

This guide includes a reference architecture that uses the PiXYZ Scenario Processor in combination with [Amazon Simple Storage Service](#) (Amazon S3), [AWS Batch](#), [Amazon Simple Queue Service](#) (Amazon SQS) and [AWS Lambda](#) to create a serverless 3D optimization and conversion pipeline that scales with customer needs. This guide also includes an AWS CloudFormation template that can be used to test this reference architecture in your AWS account.

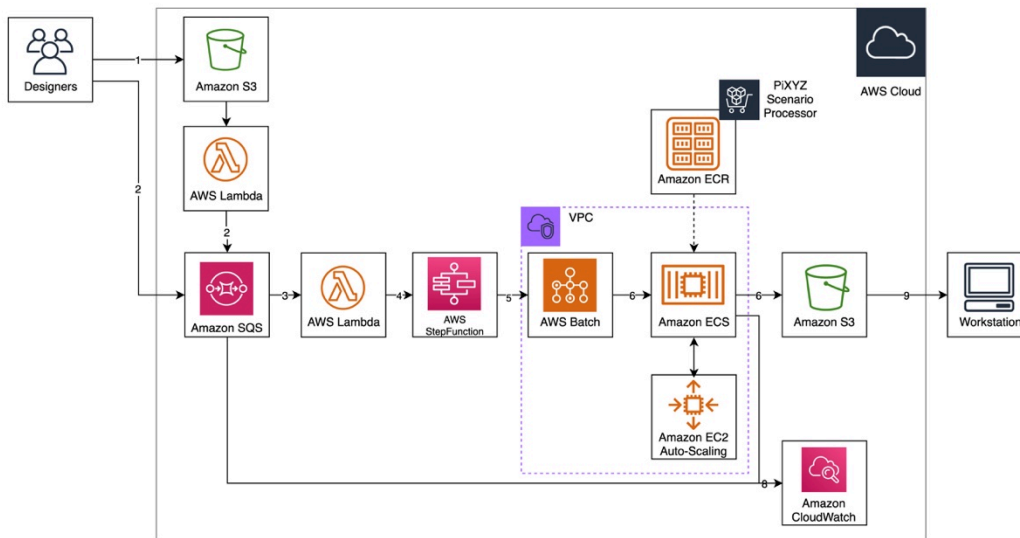
# Reference Architecture

To enable horizontal scaling (the ability to add compute resources to get things done faster), we need a way to replicate the software environment and dependencies needed for each task in a job. In the context of this guide, we need to encapsulate the PiXYZ converter software and its associated dependencies. [OS-level virtualization](#) and containers encapsulate and isolate a software environment to address this need. [Docker](#) is a popular container implementation supported on many AWS services. The PiXYZ Scenario Processor is a Docker Container offering of PiXYZ's CAD optimization and conversion software.

## Scalable 3D CAD Conversion Pipeline with PiXYZ on AWS

Build a Scalable 3D Data Conversion Pipeline to meet the demands of your business

This diagram shows a 3D data conversion from CAD to GLTF on AWS using the PiXYZ Scenario Processor from the AWS Marketplace running in a containers using AWS Batch to manage the provisioning of compute resources, Amazon Simple Queue Service (SQS) and AWS Lambda to orchestrate the processing, and Amazon Simple Storage Service (S3) to store the results.



1. Designers upload CAD files to Amazon S3
2. Designer or S3 event pushes conversion task item to Amazon SQS
3. A long-polling AWS Lambda function detects new task items
4. AWS Lambda triggers AWS StepFunction execution
5. AWS StepFunction submits AWS Batch Jobs
6. AWS Batch automatically provisions EC2 Instances through Amazon Elastic Container Service (ECS), which pulls the PiXYZ AWS Marketplace container from Amazon Elastic Container Registry (ECR)
7. Converted Files are uploaded to Amazon S3
8. Logs and Metrics are written to Amazon CloudWatch

## Scalable 3D CAD Conversion Pipeline with PiXYZ on AWS

The workload is triggered when a customer uploads their CAD files to Amazon S3. The S3 bucket is configured with an event trigger to call an AWS Lambda function on upload events. The AWS Lambda function uses a simple python script which pushes a *task item* onto a queue managed by Amazon SQS. Each task item contains the parameters needed for converting a single file, such as the PiXYZ optimization algorithm, the input and output S3 file keys, and options such as the targeted polygon count.

In addition to inserting a task item via an S3 create event, you can also use a python script running on your workstation to push task items into the SQS queue. Then, the SQS queue is polled by a second AWS Lambda function. When items arrive on the queue, the AWS Lambda function translates them into an AWS Step Function execution, and submits a task request to AWS Batch. AWS Batch provisions the [Amazon Elastic Compute Cloud](#) (Amazon EC2) instances and distributes tasks among the instances. It also leverages [Amazon Elastic Container Service](#) (Amazon ECS) to orchestrate the jobs, pulling the PiXYZ Scenario Processor container image from the [Amazon Elastic Container Registry](#) (Amazon ECR) as specified by a Job Definition.

Under periods of high load, AWS Batch will scale-out EC2 instances, including finding appropriate instance types, and packing as many containers in each instance as it can, based on the CPU and memory requirements you specify. Each container task writes its converted file back to S3 in the location specified in the SQS task item. From there the customer can use the result with their interactive 3D authoring packages.

## Prerequisites

You must first subscribe to the PiXYZ Scenario Processor AWS Marketplace container as detailed in [Appendix A](#).



# AWS CloudFormation Template

This reference architecture uses AWS CloudFormation to create and configure all the necessary services and permissions in your account. It includes the following AWS CloudFormation template which you can download before deployment:



Server

**AWS-Batch-pipeline-with-PiXYZ.yaml:** Use this template to launch this solution and all associated components. The default configuration deploys an Amazon S3 bucket, an Amazon SQS queue, AWS Lambda and Step-Functions, a Job Definition and Compute Environment in AWS Batch, an Amazon VPC, and IAM Roles for running the jobs.

# Automated Deployment

Before you launch the automated deployment, please review the reference architecture, configuration, and other considerations discussed in this guide.

**Time to deploy:** Approximately 5 minutes

# Launch the Stack

This automated AWS CloudFormation template deploys the Serverless CAD Conversion on AWS solution on the AWS Cloud.

## Note

You are responsible for the cost of the AWS services used while running this solution. See the [Costs](#) section for more details. For full details, see the pricing webpage for each AWS service you will be using in this solution.

1. Sign in to the AWS Management Console and click the button below to launch the `Serverless-AWS-Batch-pipeline-with-PiXYZ` AWS CloudFormation template.

[Launch Solution](#)

You can also [download the template](#) as a starting point for your own implementation.

2. The template is launched in the US East (N. Virginia) Region by default. To launch the solution in a different AWS Region, use the region selector in the console.
3. Under **Stack name**, assign a name to your solution stack.
4. Under **Parameters**, review the parameters for the template and modify them as necessary. This solution uses the following default values.

Parameter	Default	Description
Bucket Name	converted-files	Amazon S3 bucket names must be unique globally. The template will suffix the bucket and with a unique hash string from the CloudFormation stack.

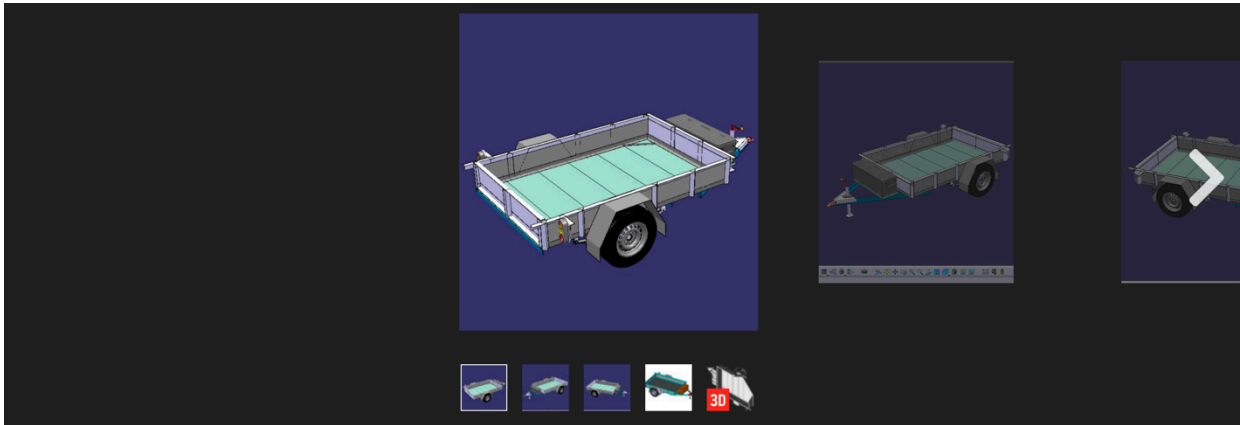
Parameter	Default	Description
Public Subnet CIDR1	10.192.10.0/24	Enter the public IP range for the public subnet in the first availability zone.
VPC CIDR	10.192.10.0/24	Enter the IP range for the VPC.

5. Review and confirm the settings. Be sure to check the box acknowledging that the template will create AWS Identity and Access Management (IAM) resources.
6. Choose **Create stack** to deploy the stack.

You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should see a status of **CREATE\_COMPLETE** in approximately five minutes.

# Understanding the Solution

Once the solution is successfully deployed in your account, you can now use the solution to decimate and simplify 3D CAD data. If you don't have CAD data to test with, you can get free test CAD models from [GrabCad.com](https://www.grabcad.com). In this test we are using this [utility trailer](#). Currently, the deployed CloudFormation template only supports .step, .stp, .jtt, .fbx, and .obj files. For a full list of supported file formats by the PiXYZ processor, see [Supported File Formats](#). Use the following procedure to begin testing in your account.



## Free CAD Data from GrabCad.com

1. Navigate to Amazon S3, and select the bucket created by the CloudFormation stack.

Amazon S3 > converted-files-ac648220 > Create folder

### Create folder

Use folders to group objects in buckets. When you create a folder, S3 creates an object using the name that you specify followed by a slash (/). This object then appears as folder on the console. [Learn more](#)

**Your bucket policy might block folder creation**  
If your bucket policy prevents uploading objects without specific tags, metadata, or access control list (ACL) grants, you will not be able to create a folder using this configuration. Instead, you can use the [upload configuration](#) to upload an empty folder and specify the appropriate settings.

**Folder**

Folder name

 /

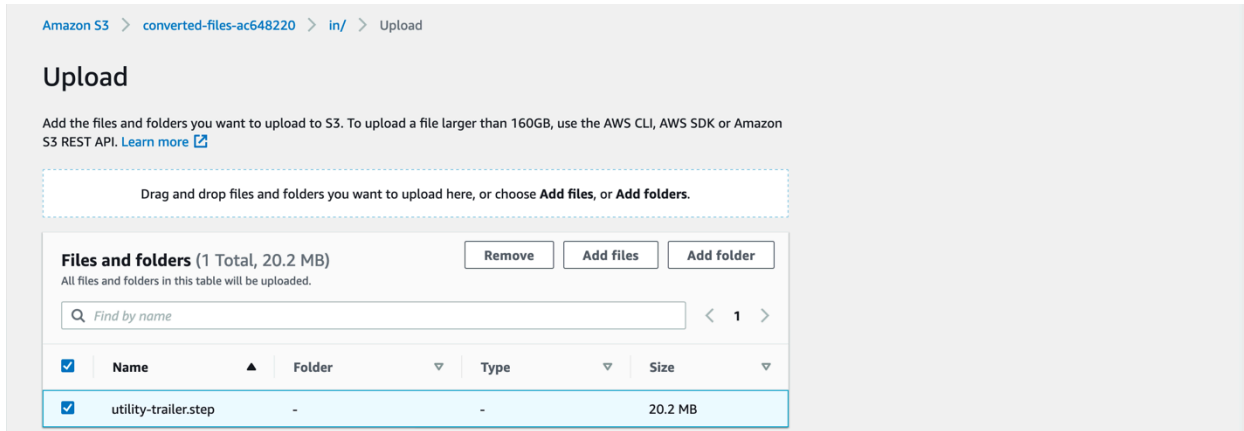
Folder names can't contain "/". [See rules for naming](#)

## Create a New Folder

7. Upload your CAD model file into the newly created in folder. Once uploaded, the job will be triggered automatically.

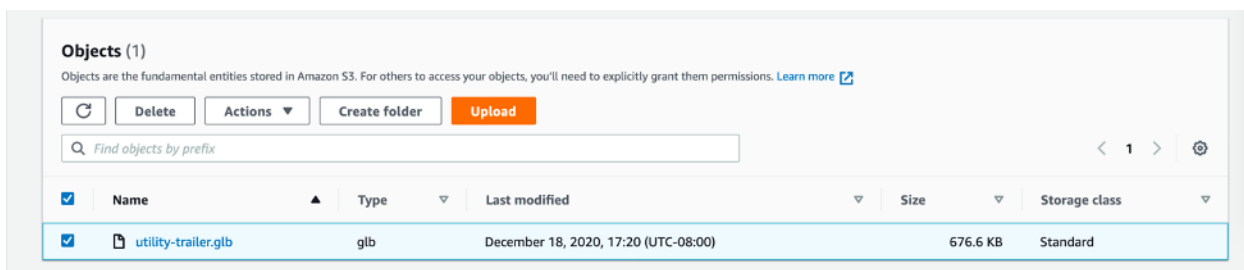
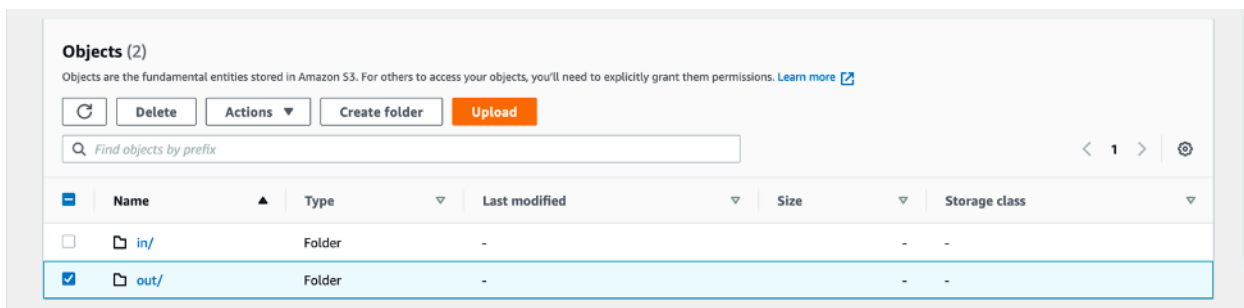
The file name shouldn't have any spaces in it. If no spaces are found, it will trigger a series of events described previously, and will submit a job to AWS Batch.

**Warning:** If you don't place a file in the in folder, the job won't start and will fail. For more information about debugging failed jobs, see [Troubleshooting](#).



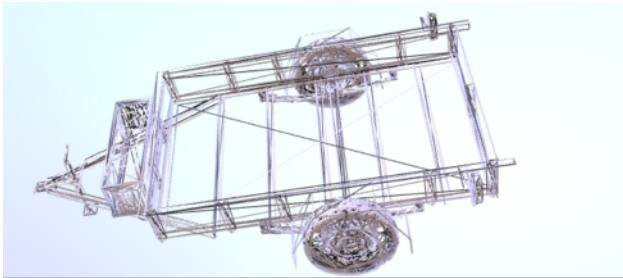
### Upload CAD file

- Once the job has successfully completed, an exported .glb file will automatically be placed in an out folder in the same S3 bucket used previously.



### Exported 3D CAD Mesh

- You can view exported file in your preferred model viewer, and see the results of the PiXYZ decimation.



*3D Mesh Review*

## Scaling

This solution scales to meet the demands of the input files placed in your S3 bucket. For example, if you placed 350 files in the S3 bucket's in folder, AWS Batch jobs would be started in parallel as the files arrived. During testing, we tried this experiment with a set of 350 files that took nearly 14 hours of total compute time to convert, by using this solution we received the results back in just 10 minutes through AWS Batch's fully managed fan-out parallelization.



## Costs

You are responsible for the costs of the AWS services used while running this solution. The total cost for running this solution depends on the size and complexity of the input test model. For full details, refer to the pricing webpage for each AWS service you will be using in this solution.

## Example cost

With the solution deployed in the US East (N. Virginia) Region and using the default configuration, a 41MB input step file that takes 15 seconds for the PiXYZ scenario processor to convert, and a m4.large instance, costs approximately **\$0.07**.

AWS Batch manages the on-demand EC2 instances for the processing. For this cost example, a single m4.large instance, scaling in after 10m21s was used. The delayed scale-in allows for subsequent jobs to use the started EC2 instances, amortizing the cold-start times for the instances.

**Note:** There are additional negligible service costs associated, such as \$0.0009 for S3 and \$0.00013 for SQS).

Table 2 - Summary of Costs

<b>PiXYZ Scenario Processor</b> (15 sec @ \$15/hr)	\$ 0.25
<b>EC2</b> (10m21s m4.large @ \$.1/hr)	\$ 0.01
<b>Total</b>	<b>\$ 0.26</b>

In this example we are using the on-demand pricing for the PiXYZ Scenario Processor. This enables customers to innovate quickly with low risk since there are no upfront costs. However, some customers prefer to have annual fixed costs. For this option, PiXYZ offers a yearly fixed cost subscription (\$20,000 per year) which allows unlimited use for a single container instance. If purchased, parallel tasks would first be metered to the yearly subscription until they are all consumed, then fallback the hourly metering. For example, if one yearly subscription was purchase and AWS Batch parallelized 64 tasks, the first task would be run at no charge (other than the yearly subscription), but the other 63 tasks would be metered at the hourly rate.

The advantages of fixed costs and unlimited usage from the yearly subscription must be balanced against rapid turn-around times through parallelization. If you have a constant, non-urgent

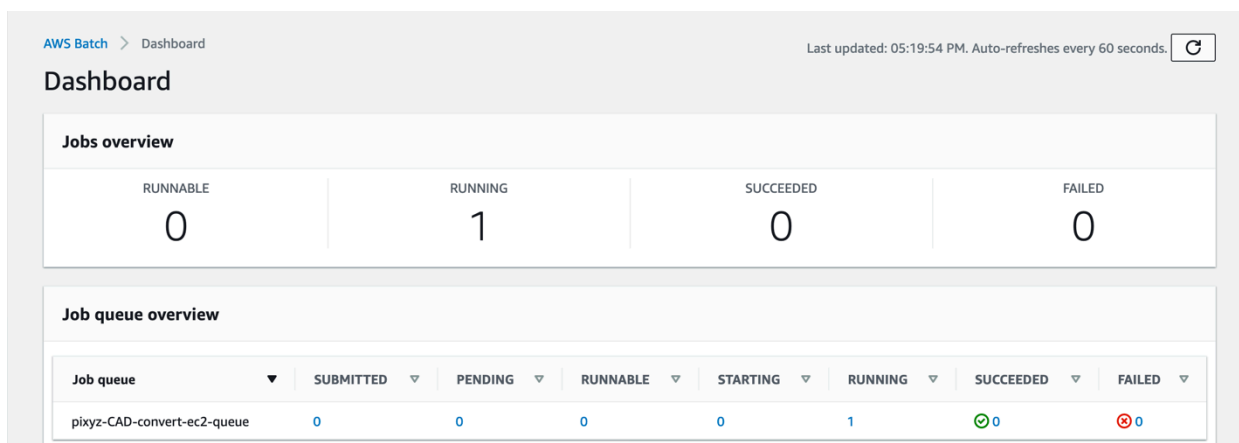
workload of serializable conversions that exceeds 1333.3 hours of conversions per year (the equivalent of \$20,000 at the hourly rate), we recommend using the hourly pricing so that you pay for only what you use and can aggressively parallelize the workloads for faster turn-around times.

# Troubleshooting

If the job fails, you can run the following troubleshooting procedures to debug steps.

## AWS Batch

1. In the AWS Management Console, navigate to AWS Batch dashboard.
2. After the upload to S3 is completed, you will see the job status progress through Submitted, Pending, Runnable, Starting, Running, Succeeded and Failed states.
3. If a job fails, select the Failed job number, then selecting on the Name of the job, then selecting the Log Stream Name to see the associated CloudWatch logs which can be used to debug the failure.



### AWS Batch Job Status

## Amazon CloudWatch Logs

If the job wasn't added to the job queue in AWS Batch, you can review the Lambda execution logs.

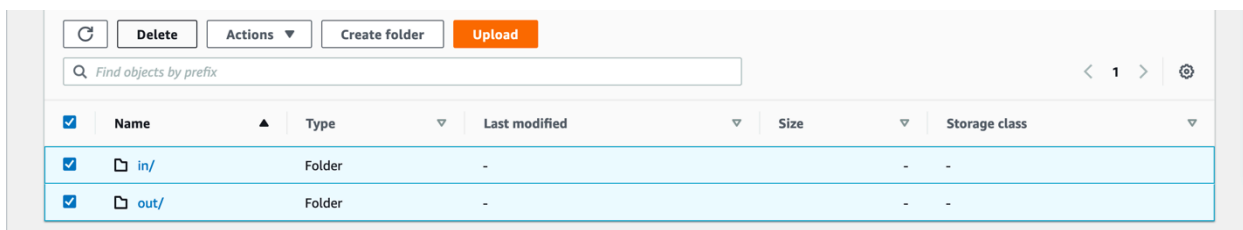
1. In the AWS Management Console, navigate to Amazon CloudWatch .
2. In the left-hand navigation panel, select **Logs**, then **Log Groups**, and check the StepFunction execution. The log group will start with the name `/aws/lambda/<CloudFormation Stack Name>-LambdaPollSQSExecuteStep`.

# Cleaning up

If the solution is no longer required, it is recommended that you remove associated resources and avoid unnecessary charges. Use the following procedure, to remove the solution from your account. Since the resources are part of a CloudFormation stack, clean-up is mostly handled by removing that stack, except for the additional step of emptying your S3 bucket.

1. Navigate to your Amazon S3 bucket, select the top-level folders, and choose delete.

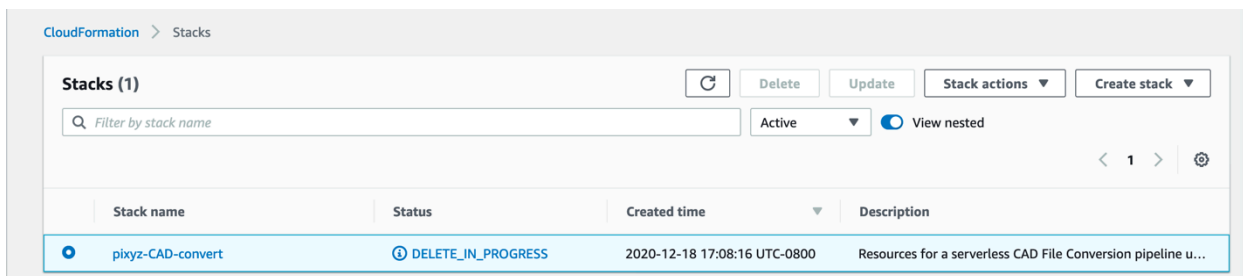
**Warning:** If you must empty and delete your S3 bucket before removing the CloudFormation stack, or you will receive an error during the deletion of the stack.



## Clean up Amazon S3 Contents

2. In the AWS Management Console, navigate to the AWS CloudFormation dashboard.
3. In the list of stacks, select the created stack, then choose delete.

You can view the progress of the deletion, by selecting the stack name, then events tab.



## Delete AWS CloudFormation Stack

This guide shows how to horizontally scale 3D Data Optimization and Conversion pipelines. Using the PiXYZ Scenario Processor container from the AWS Marketplace as an example, we were able to turn-around almost 14 hours of processing in just over 10 minutes without having to worry about managing the compute infrastructure for the processing.

You can begin building your own serverless, scalable 3D Data Optimization and Conversion pipelines using this document as a starting point to test on a larger, more production-representative data set.

# Contributors

Contributors to this document include:

- Leo Chan, Worldwide Tech Lead, AWS Spatial Computing
- Adam Harder, R&D Spatial Computing Solutions Architect

## Further Reading

For additional information, see:

- [AWS Batch](#)
- [Amazon Marketplace](#)
- [Amazon S3](#)
- [Amazon SQS](#)
- [Amazon EC2](#)
- [Amazon ECR](#)
- [Amazon ECS](#)
- [Amazon EC2 Auto Scaling](#)
- [The PiXYZ Scenario Processor Usage Guide](#)
- For technical support and PiXYZ questions, visit the [Pixyz website](#).

# Document Revisions

To be notified about updates to this whitepaper, subscribe to the RSS feed.

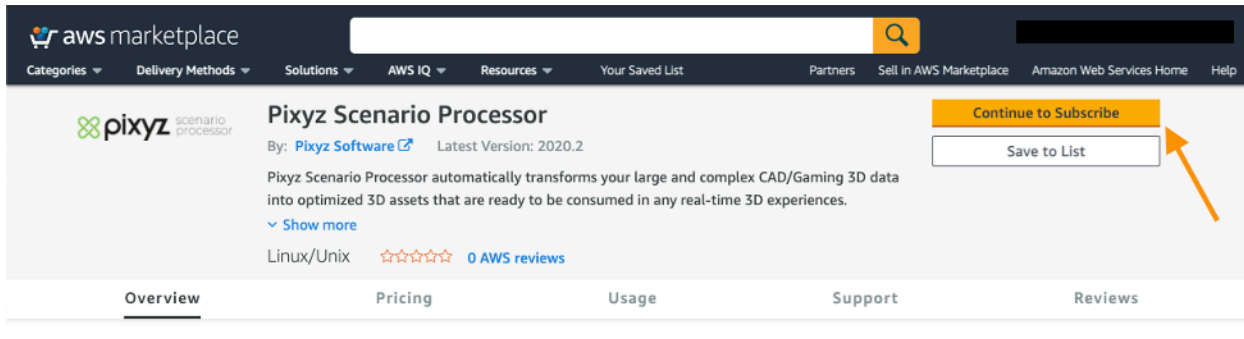
Change	Description	Date
<a href="#">Initial Publication</a>	First Publication	March 8, 2021



# Appendix A: Subscribing to the PiXYZ Scenario Processor

You must subscribe to the PiXYZ Scenario Processor container before you can use it.

1. In the AWS Marketplace, navigate to the [Pixyz Scenario Processor](#).
2. In the upper right corner, select **Continue to Subscribe**.

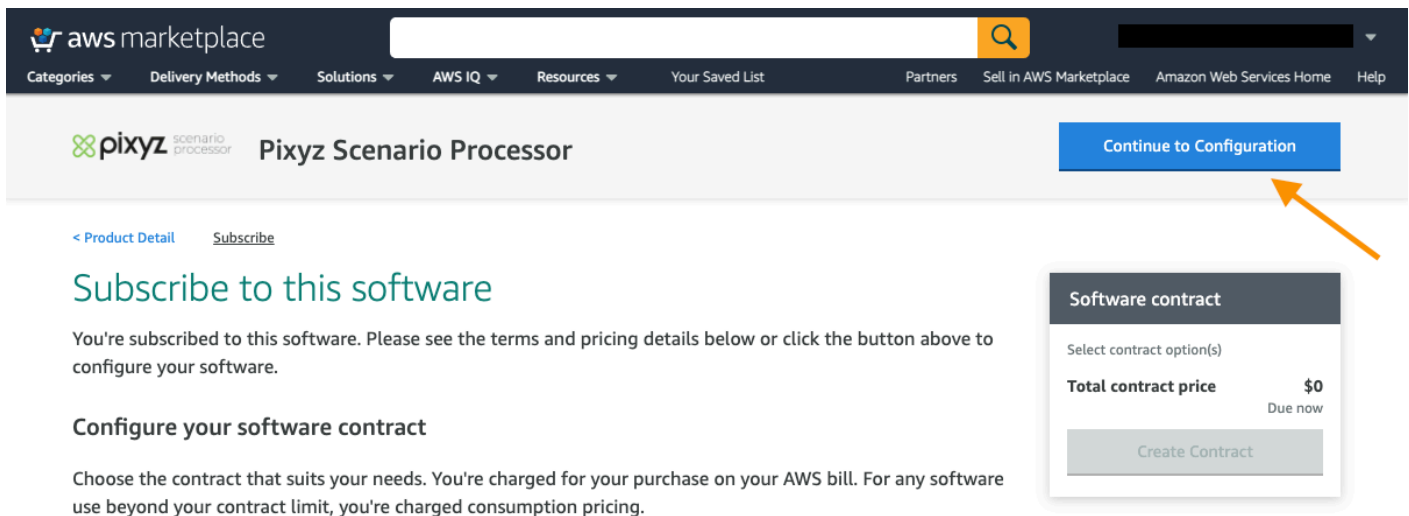


## *Continue to Subscribe*

3. Read the Terms and Conditions and choose **Accept Terms**.

It will take a few minutes for your subscription request to process, and the product subscription page will refresh where you can continue to configuration. If you have navigated away from the product subscription page you can return to it by starting at step 1.

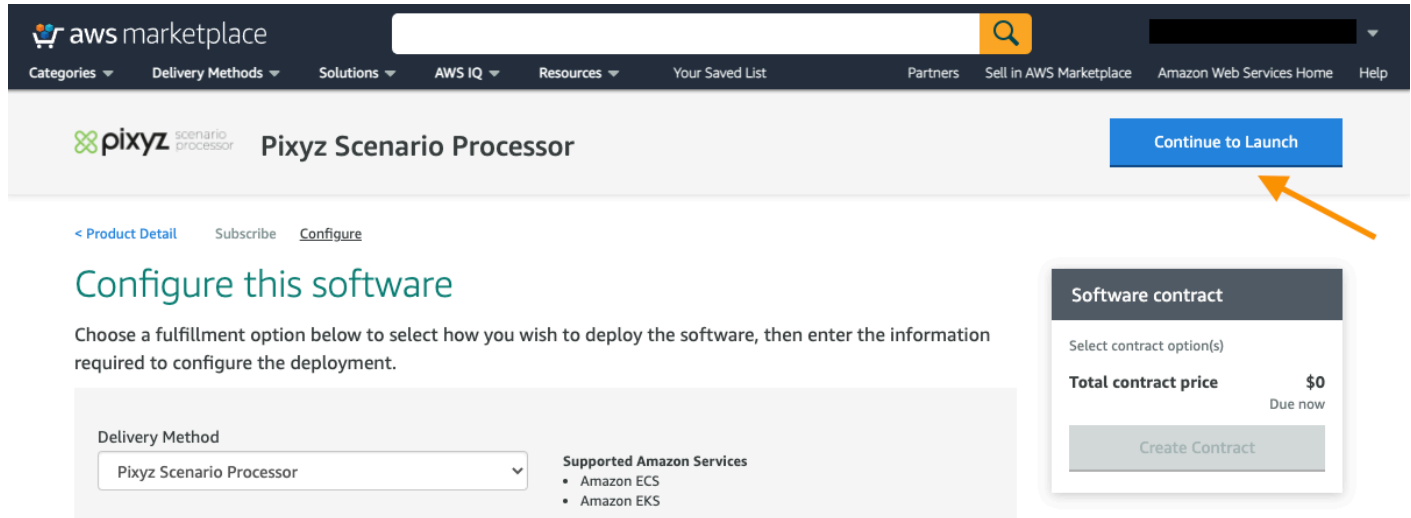
4. Select **Continue to Configuration**.



## *Continue to Configuration*

There is a **Contract options** section on this page. This is if you want to buy annual, unlimited use contracts up front. Each of these contracts is good for a single container instance. In this guide we are using the pay-per-use option instead, you don't need to fill in the Contract options section.

5. On the page that appears, select **Continue to Launch**.



The screenshot shows the AWS Marketplace interface for the Pixyz Scenario Processor. The top navigation bar includes the AWS Marketplace logo, a search bar, and various menu items like Categories, Delivery Methods, Solutions, AWS IQ, Resources, Your Saved List, Partners, Sell in AWS Marketplace, Amazon Web Services Home, and Help. The main content area features the Pixyz logo and the product name 'Pixyz Scenario Processor'. A blue 'Continue to Launch' button is prominently displayed. Below this, there are links for '< Product Detail', 'Subscribe', and 'Configure'. The 'Configure this software' section instructs users to choose a fulfillment option and enter deployment information. A 'Delivery Method' dropdown menu is set to 'Pixyz Scenario Processor', and 'Supported Amazon Services' are listed as Amazon ECS and Amazon EKS. A 'Software contract' summary box shows 'Total contract price' as '\$0 Due now' and a 'Create Contract' button.

### *Continue to Launch*

6. On the page that appears, select **View container image details**, and copy the docker image address into a text editor. You will use this image address in your AWS Batch Job Definition to specify the container for job tasks.

aws marketplace

Categories ▾ Delivery Methods ▾ Solutions ▾ AWS IQ ▾ Resources ▾ Your Saved List Partners Sell in AWS Marketplace Amazon Web Services Home Help

Ensure you have installed the latest version of the AWS CLI and Docker. For more information, see [ECR documentation](#)

**For macOS or Linux systems, use the AWS CLI:**

Step 1: Retrieve the login command to authenticate your Docker client to your registry.

```
aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin 117940112483.dkr.ecr...
```

For Windows systems, use AWS Tools for PowerShell:

```
Invoke-Expression -Command (Get-ECRLoginCommand -Region us-east-1 -RegistryId "117940112483").Command
```

Note: If you receive an 'Unknown options: -no-include-email' error when using the AWS CLI, ensure that you have the latest version installed.

**Step 2: Pull all docker images listed below.**

- 117940112483.dkr.ecr.us-east-1.amazonaws.com/0b8e0df6-0934-4d27-a551-b1f6870d21be/cg-1437830752/pixyz...

Amazon EKS

Usage Instructions

**Container Images**

This product has 1 container image. Use the following options to deploy the product.

Deployment template

[ECS Task Definition for Pixyz Software Scale-Out](#)

[Usage Instructions](#)

[View container image details](#)

## Container Image Details

You can now use the PiXYZ Scenario Processor. You will only be charged for the time used to process CAD files. You can manage your [AWS Marketplace Subscriptions](#) at any time from the [AWS console](#).

## Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents current AWS product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS products or services are provided “as is” without warranties, representations, or conditions of any kind, whether express or implied. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

© 2021 Amazon Web Services, Inc. or its affiliates. All rights reserved.