

# Graph.Build SQL Transformer Usage Instructions

These instructions help you deploy, configure, and run the Graph.Build SQL Transformer from the AWS Marketplace Quick Create CloudFormation template.

The SQL Transformer is an API service. It does not include a web user interface. After deployment, you interact with it through the API endpoint shown in the CloudFormation stack outputs.

## Launch With Quick Create

Use the AWS CloudFormation Quick Create link:

<https://us-east-1.console.aws.amazon.com/cloudformation/home?region=us-east-1#/stacks/create/review?templateURL=https://data-lens-cloudformation.s3.us-east-1.amazonaws.com/monthly-licensed-products/sql-transformer.yaml&stackName=GraphBuild-SQLTransformer>

On the Quick Create page:

1. Confirm the stack name, or enter your own name.
2. Choose the deployment region. If you change the region, set the `AWSRegion` parameter to the same region.
3. Choose an `InstanceType`. The default is suitable for normal use. Larger instance types provide more memory and CPU for transformation jobs.
4. Set `GraphDatabaseType`:
  - Use `neptune-sparql` for RDF/SPARQL output.
  - Use `neptune-gremlin` for property graph output.
5. Choose whether to create an S3 bucket:
  - Keep `CreateS3Bucket` as `true` to create a bucket named `graph-build-<account-id>`.
  - Set `CreateS3Bucket` to `false` if you have already created the bucket.
  - Optionally provide `S3BucketName` to use a specific bucket name.
6. Set `EnableCloudFront` to `true` if you want an HTTPS CloudFront URL. Leave it as `false` to use the Application Load Balancer HTTP URL.
7. Acknowledge that CloudFormation will create IAM resources.
8. Choose **Create stack**.

Deployment usually takes several minutes. Wait until the CloudFormation stack status is `CREATE_COMPLETE`.

## Prepare The S3 Workspace

The transformer reads and writes workspace files in S3. The template configures the transformer directory as:

```
None  
s3://<bucket-name>/sql/
```

If the stack creates the bucket, the bucket name is available in the CloudFormation output `GraphBuildBucketName`.

Create the `sql/` prefix and upload the files required by your transformation workflow, such as mappings, source files, connection metadata, and output locations, according to your Graph.Build documentation.

## Access The API

This product has no browser UI.

To find the API endpoint:

1. Open the CloudFormation console.
2. Select your SQL Transformer stack.
3. Open the **Outputs** tab.
4. Copy the value of `ApiEndpoint`.

If `EnableCloudFront` is `true`, the endpoint will be an HTTPS CloudFront URL. If `EnableCloudFront` is `false`, the endpoint will be an HTTP Application Load Balancer URL.

## Health Check

To verify the service is responding, append `/config` to the `ApiEndpoint` output.

Example:

```
Shell  
curl <ApiEndpoint>/config
```

For example, if the output is `https://abc123.cloudfront.net`, run:

```
Shell  
curl https://abc123.cloudfront.net/config
```

A successful response means the API container is reachable through the load balancer or CloudFront distribution.

## SSH Access

SSH access is intentionally not enabled for this product.

The stack does not ask for an EC2 key pair, does not open port 22, and places the ECS host in a private subnet. This follows AWS Marketplace guidance to avoid administrative access from the internet.

For troubleshooting, use:

- The CloudFormation **Events** tab to check deployment progress.
- The ECS console to inspect the cluster, service, task, and container status.
- CloudWatch Logs for application logs. The log group is named `/ecs/<stack-name>`.
- EC2 status checks for the ECS host instance.

If you require host-level access, create a modified template only after reviewing your organisation's security requirements. Do not open SSH to `0.0.0.0/0`.

## Upgrade Instructions

The SQL Transformer service is stateless. It does not store application data on the ECS host and does not create a persistent application volume.

Transformation inputs, mappings, and outputs should be stored in S3 under the configured transformer workspace:

None

```
s3://<bucket-name>/sql/
```

To preserve data during an upgrade:

1. Keep the same S3 bucket and `sql/` prefix when deploying the upgraded template.
2. Back up any important S3 objects before changing or deleting the stack.
3. Prefer updating the existing CloudFormation stack with the newer template or image tag.
4. After the upgrade completes, retrieve the `ApiEndpoint` output and check `<ApiEndpoint>/config`.

Because the service itself is stateless, replacing the ECS task or EC2 host does not remove transformer workspace files stored in S3. However, if the stack created the S3 bucket, deleting the stack can affect that bucket depending on its contents and deletion settings. Back up important files before deleting the stack.

## CloudFormation Delivery Information

- All information remains in your AWS account and is not shared with Graph.Build.
- IAM roles and policies are created for the ECS host and ECS task. These allow the EC2 instance to join the ECS cluster, allow the ECS task to pull container images, write logs to CloudWatch, access the configured S3 workspace, and use AWS License Manager for Marketplace license checks.
- Network resources include a new VPC, two public subnets for the Application Load Balancer, one private subnet for the ECS host, an Internet Gateway, a NAT Gateway with Elastic IP for outbound access, route tables, and security groups.
- Compute resources include one ECS cluster, one EC2 Auto Scaling Group with a single ECS-optimized EC2 host, one ECS task definition, and one ECS service running the SQL Transformer container.
- Storage resources optionally include an S3 bucket for transformer workspace files. No EFS or EBS application data volume is created for this product.
- Access resources include an internet-facing Application Load Balancer that forwards requests to the transformer API. If `EnableCloudFront` is set to `true`, the template also creates a CloudFront distribution for HTTPS access.
- Logging resources include a CloudWatch Logs log group named `/ecs/<stack-name>`.

Resource	\$ hourly	\$ month
t2.medium	0.0464	33.87
EBS Volume 30GB	0.0032	2.4
Detailed monitoring	0.0032	2.4
Application LoadBalancer	0.0225	16.43

## Deleting The Stack

Deleting the CloudFormation stack removes the AWS resources created by the template. If the template created the S3 bucket and the bucket contains objects, empty or back up the bucket before deleting the stack.

## Support And Documentation

For product documentation and support, contact [support@graph.build](mailto:support@graph.build) or visit <https://docs.graph.build>.