SingleStore Ingest

01/09/2025

SingleStore Ingest

Overview

SingleStore Ingest (“Ingest”) is real-time data replication software that replicates data from various sources to SingleStore. It's one of the primary components of SingleStore Flow. Ingest offers high performance, enabling real-time Change Data Capture from sources with zero load on the source systems. It captures changes and transfers them to the target system. It automates the creation of either an exact copy or a time-series copy of the data source in the target. It first performs a full initial load from the source, then incrementally merges changes to SingleStore. The entire process is fully automated.

Supported Source Databases

Ingest supports the following database sources:

- Oracle
- MS SQL Server
- MySQL
- PostgreSQL

Contact your SingleStore account team or SingleStore Sales if you want to move data from a source not listed above.

Ingest Architecture

Ingest replicates data from any supported source to a SingleStore destination database. It is a fully self-service, automated data replication tool.

SingleStore Flow, of which Ingest is a part, offers several deployment strategies for its customers, including:

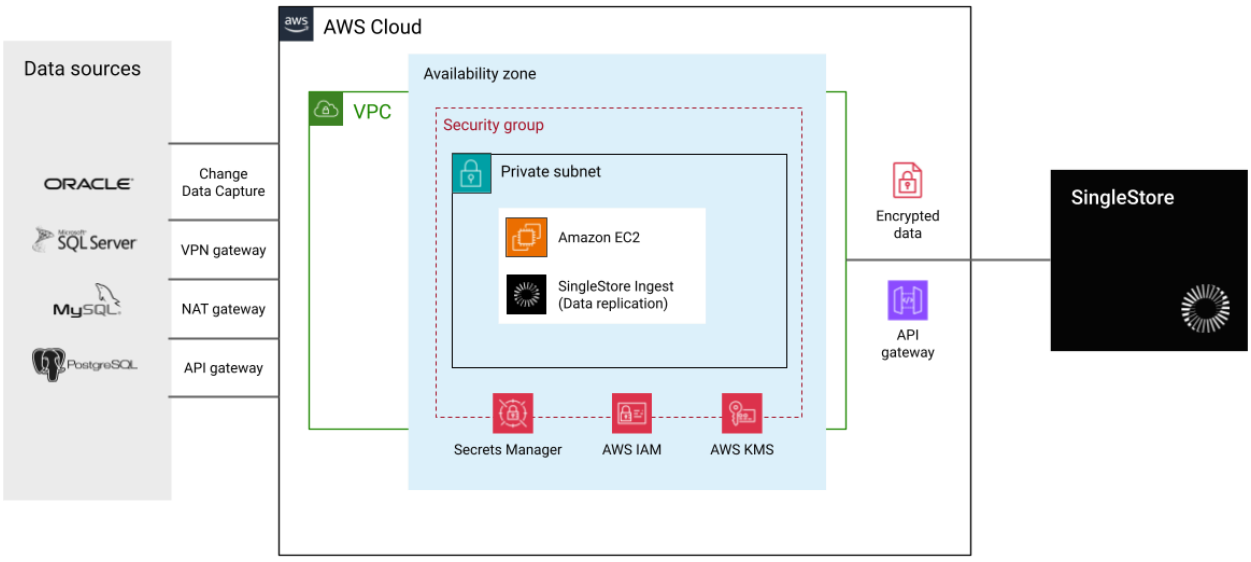
- Standard deployment in an AWS environment
- High Availability deployment in an AWS environment
- Hybrid deployment using both on-premises and cloud infrastructure
- Fully on-premises deployment

SingleStore Flow components can be deployed in Google Cloud and Microsoft Azure as well. We reference AWS components and services here as an illustration of a common type of deployment.

Ingest uses log-based Change Data Capture for data replication. The following is the technical architecture diagram that illustrates the standard setup in an AWS environment.

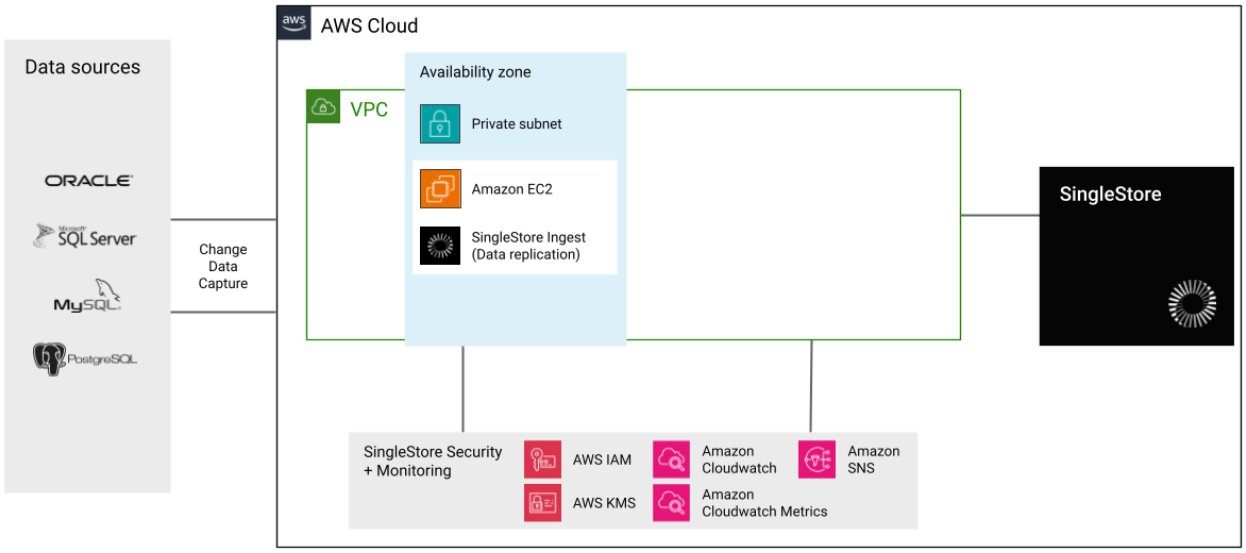
This diagram serves as the reference for all setup instructions.

Estimated deployment time: Approximately 1 hour



Ingest Architecture

The following is the Ingest architecture which showcases integration with various optional AWS services in a standard deployment.



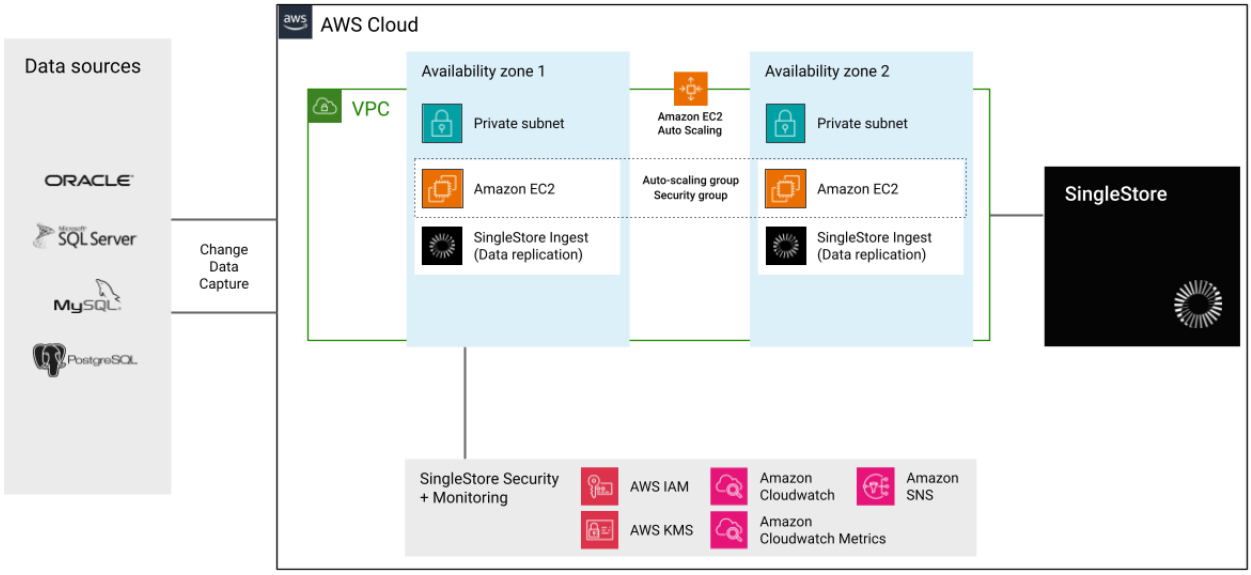
Ingest / AWS Service Integration

The above architecture diagram illustrates a standard deployment that highlights the following features:

- AWS services running alongside Ingest.
- Recommended SingleStore Flow architecture for a VPC in AWS.
- Data flow between the source database, AWS, and SingleStore destination database, including security and monitoring features.
- Security, including IAM, organized in a separate group and integrated with Ingest.

The following high availability architecture explains how Ingest is deployed in a multi-AZ setup. In the event of an instance or AZ failure, it automatically scales to another AZ without incurring any data loss.

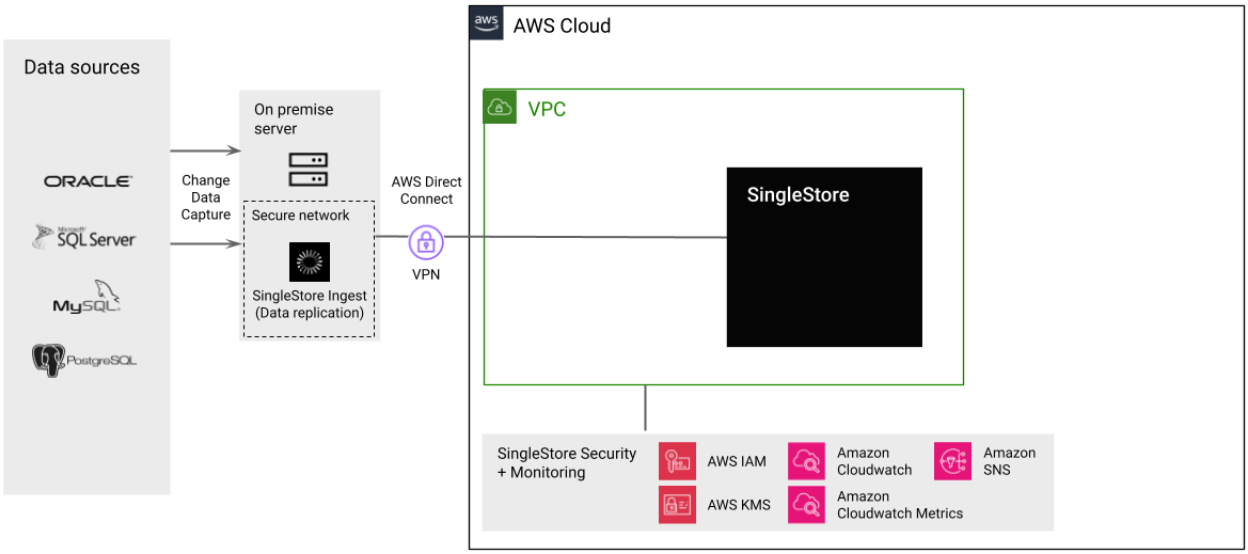
Estimated deployment time: Approximately 1 day



Ingest High Availability (HA) Architecture

Ingest also offers a hybrid deployment model to its customers that combines on-premises services with those in the AWS Cloud. Ingest can be easily set up on a Windows server in an on-premises environment. The SingleStore destination endpoint resides in the AWS Cloud that creates a hybrid model. SingleStore recommends to ensure secure connectivity between on-premises and AWS services, which can be achieved using a VPN connection or AWS Direct Connect.

Estimated deployment time: Approximately 2 hours to 1 day



Ingest Hybrid Architecture

Prerequisites

The following are the prerequisites for launching Ingest on Amazon EC2

- Selection of the Ingest volume.
- Selection of the EC2 instance type.
- Ensure connectivity between the server/EC2 hosting the Ingest software and the source. Additionally, ensure connectivity to DynamoDB if the high availability option is required.

To create the necessary AWS services, refer to Environment Preparation.

The following are the steps to take before launching SingleStore Flow in AWS via custom installation on an EC2:

1. Create a policy with a relevant name for EC2, such as `FlowEC2Policy`. Refer to the Define custom IAM permissions with customer managed policies for creating policies.
2. Refer to AWS Identity and Access Management (IAM) for SingleStore Flow for JSON policy.

3. Create an IAM role called `FlowEC2Role`. Refer to Create a role to delegate permissions to an IAM user for creating roles.
4. Attach the `FlowEC2Policy` to the role.
5. Create a Lambda policy for disk checks and attach the Lambda policy JSON. Refer to Recovery from Faults for Lambda Policy JSON.

The following are the recommended EC2 options for replicating source data volumes.

Total Data Volume	**EC2 Recommended**
< 100 GB	t2.small
100GB – 300GB	t2.medium
300GB – 1TB	t2.large
> 1TB	Contact SingleStore Support

These recommendations serve as a starting point. If you have any questions, contact SingleStore Support or your technical account team representative.

The following are the system requirements when not using the Amazon EC2:

- Port 8081 must be open on the server hosting the Ingest software.
- Google Chrome is required as the internet browser on the server hosting Ingest software.
- Java version 21 or higher is required.
- If using MS SQL Server as a source, download and install the BCP utility.
- Ensure connectivity between the server hosting Ingest software and the source, and DynamoDB (if the high availability option is required).

Recommended Hardware Configuration

The following describes the hardware configuration for a Windows server, assuming that there are a few sources and target combinations (3 medium ideally). It also depends on how intensively the data is being replicated from these sources, so this is a guide, but will need extra resources depending on the amount of data being replicated. The amount of disk space will also be dependent on the amount of data being replicated.

The following describes the hardware configuration for a Windows server; similar configuration is recommended for a Linux or Ubuntu based server. The configuration also depends on the intensity of data replication from these sources. Additional resources may be required based on the volume of data being replicated. The disk space required also depends on the amount of data being replicated.

Component	**Specification**
Processor	4 cores
Memory	16 GB
Disk requirements	Varies based on the data being extracted, with a minimum of 300 GB
Network performance	High

Prerequisites for Software on Server

The following software must be installed on the server:

- 64-bit Open JDK 21: Amazon Corretto 21 JRE
- Google Chrome
- For SQL Server Sources (Only): Install the following tools and drivers:
  - BCP Utility
  - VC++ 2017 64-bit
  - ODBC Drivers 18 64-bit
  - SQLCMD Version 15 64-bit
- For MySQL Server (Only): `mysqlbinlog.exe` must be installed on the server and included in the system path.

Required Skills

SingleStore Flow is a suite of robust applications that makes seamless data replication to the cloud. It handles large data volumes with ease, and the process is fully automated. The setup takes only three simple steps. The application does not require highly technical resources, but basic knowledge of the following is recommended for deployment:

- AWS Cloud Fundamentals
- Basic database skills, including writing and executing database queries (for RDBMS endpoints)
- Familiarity with using Microsoft Windows or Linux-based systems

Installation

For details on how to install Ingest and other Flow components, refer to SingleStore Flow Installation.

AWS Identity and Access Management (IAM) for SingleStore Flow

AWS IAM roles delegate access to AWS resources. With IAM roles, you can establish trust relationships between your trusting account and other trusted AWS accounts. The trusting account owns the resource to be accessed, while the trusted account contains the users who need access.

SingleStore's Recommendations:

1. Create an IAM User (e.g., "SingleStore_Flow_User"). Do **NOT** use the root user account to set up the application. Refer to Create an IAM user in your AWS account for creating an IAM user.
2. Create an IAM Role (e.g., "SingleStore_Flow_EC2Role"). Refer to IAM role creation for creating IAM roles.
3. Create an IAM Policy (e.g., "SingleStore_Flow_Policy") and assign custom policies (provided below) to the EC2 role. Refer to Creating IAM policies for creating a policy.
4. Instead of defining permissions for individual SingleStore Flow IAM users, it is more convenient to create groups based on job functions (e.g., administrators, developers, accounting, etc.). Define the relevant permissions for each group and assign IAM users to those groups. All users in an IAM group inherit the permissions assigned to the group. This way, you can make changes for everyone in the group in one place.
5. Only grant the minimal required permissions to the IAM role. The SingleStore Flow user requires basic permissions for S3, CloudWatch, SNS and DynamoDB.
6. SingleStore Flow needs access to the following AWS services: S3, EC2, SNS, and CloudWatch with the minimum privileges outlined below.

JavaScript

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "1",
      "Action": [
        "s3:DeleteObject",
        "s3:GetObject",
        "s3:ListBucket",
```

```
        "s3:PutObject"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::<bucket_name>"
    },
    {
      "Sid": "2",
      "Action": [
        "ec2:AcceptVpcEndpointConnections",
        "ec2:AcceptVpcPeeringConnection",
        "ec2:AssociateIamInstanceProfile",
        "ec2:CreateTags",
        "ec2:DescribeTags",
        "ec2:RebootInstances"
      ],
      "Effect": "Allow",
      "Resource":
"arn:aws:ec2:<region>:<account_id>:instance/<ec2_instance_id>"
    },
    {
      "Sid": "4",
      "Action": [
        "sns:Publish"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:sns::<region>:<account_id>:<sns_name>"
    },
        {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "dynamodb:CreateTable",
        "dynamodb:PutItem",
        "dynamodb:Update*",
        "dynamodb:Get*",
        "dynamodb:Scan"
      ],
      "Resource": "arn:aws:dynamodb::<region>:table/SingleConnectTable"
    },

      ]
}
```

Roles Defined for SingleStore Flow

The following are the roles and permissions required to launch and manage the SingleStore Flow suite of applications.

Role	Type	Permissions/Policies	Purpose
EC2Admin	AWS Custom Role for EC2	List-DescribeInstanceStatus Directory Service List,Write-DescribeDirectories,CreateComputer Systems Manager List,Read,Write ListAssociations, ListInstanceAssociations, DescribeAssociation, DescribeDocument, GetDeployablePatchSnapshotForInstance, GetDocument, GetManifest, GetParameters, PutComplianceItems, PutInventory, UpdateAssociationStatus, UpdateInstanceAssociationStatus, UpdateInstanceInformation	Create and Manage EC2 instance
DBAdmin	AWS Custom Role	cloudwatch:DeleteAlarms cloudwatch:Describe* cloudwatch:DisableAlarmActions cloudwatch:EnableAlarmActions cloudwatch:Get* cloudwatch:List* cloudwatch:PutMetricAlarm dynamodb:CreateTable dynamodb:BatchGetItem dynamodb:BatchWriteItem dynamodb:ConditionCheckItem dynamodb:PutItem dynamodb:DescribeTable dynamodb:DeleteItem dynamodb:GetItem dynamodb:Scan dynamodb:Query dynamodb:UpdateItem ec2:DescribeAccountAttributes ec2:DescribeAddresses ec2:DescribeAvailabilityZones ec2:DescribeInternetGateways ec2:DescribeSecurityGroups ec2:DescribeSubnets ec2:DescribeVpcs iam:ListRoles iam:GetRole	Manage DB access and privileges

		kms:ListKeys logs:DescribeLogGroups logs:DescribeLogStreams logs:FilterLogEvents logs:GetLogEvents logs:Create* logs:PutLogEvents sns:Get sns:List* sns:SetTopicAttributes	
Network Admin	Custom Role	autoscaling:Describe*, directconnect:*, ec2:AcceptVpcEndpointConnections, ec2:AllocateAddress, ec2:AssignIpv6Addresses, ec2:AssignPrivateIpAddresses, ec2:AssociateAddress, ec2:AssociateDhcpOptions, ec2:AssociateRouteTable, ec2:AssociateSubnetCidrBlock, ec2:AssociateVpcCidrBlock, ec2:AttachInternetGateway, ec2:AttachNetworkInterface, ec2:AttachVpnGateway, ec2:CreateCarrierGateway, ec2:CreateCustomerGateway, ec2:CreateDefaultSubnet, ec2:CreateDefaultVpc, ec2:CreateDhcpOptions, ec2:CreateEgressOnlyInternetGateway, ec2:CreateFlowLogs, ec2:CreateInternetGateway, ec2:CreateNatGateway, ec2:CreateNetworkAcl, ec2:CreateNetworkAclEntry, ec2:CreateNetworkInterface, ec2:CreateNetworkInterfacePermission, ec2:CreatePlacementGroup, ec2:CreateRoute, ec2:CreateRouteTable, ec2:CreateSecurityGroup, ec2:CreateSubnet, ec2:CreateTags, ec2:CreateVpc, ec2:CreateVpcEndpoint, ec2:CreateVpcEndpointConnectionNotification, ec2:CreateVpcEndpointServiceConfiguration, ec2:CreateVpnConnection, ec2:CreateVpnConnectionRoute, ec2:CreateVpnGateway, ec2:DeleteCarrierGateway, ec2:DeleteEgressOnlyInternetGateway, ec2:DeleteFlowLogs, ec2:DeleteNatGateway,	Manage Network access and firewall settings

		ec2:DeleteNetworkInterface, ec2:DeleteNetworkInterfacePermission, ec2:DeletePlacementGroup, ec2:DeleteSubnet, ec2:DeleteTags, ec2:DeleteVpc, ec2:DeleteVpcEndpointConnectionNotifications, ec2:DeleteVpcEndpointServiceConfigurations, ec2:DeleteVpcEndpoints, ec2:DeleteVpnConnection, ec2:DeleteVpnConnectionRoute, ec2:DeleteVpnGateway, ec2:DescribeAccountAttributes, ec2:DescribeAddresses, ec2:DescribeAvailabilityZones, ec2:DescribeCarrierGateways, ec2:DescribeClassicLinkInstances, ec2:DescribeCustomerGateways, ec2:DescribeDhcpOptions, ec2:DescribeEgressOnlyInternetGateways, ec2:DescribeFlowLogs, ec2:DescribeInstances, ec2:DescribeInternetGateways, ec2:DescribeKeyPairs, ec2:DescribeMovingAddresses, ec2:DescribeNatGateways, ec2:DescribeNetworkAcls, ec2:DescribeNetworkInterfaceAttribute, ec2:DescribeNetworkInterfacePermissions, ec2:DescribeNetworkInterfaces, ec2:DescribePlacementGroups, ec2:DescribePrefixLists, ec2:DescribeRouteTables, ec2:DescribeSecurityGroupReferences, ec2:DescribeSecurityGroupRules, ec2:DescribeSecurityGroups, ec2:DescribeStaleSecurityGroups, ec2:DescribeSubnets, ec2:DescribeTags, ec2:DescribeVpcAttribute, ec2:DescribeVpcClassicLink, ec2:DescribeVpcClassicLinkDnsSupport, ec2:DescribeVpcEndpointConnectionNotifications, ec2:DescribeVpcEndpointConnections, ec2:DescribeVpcEndpointServiceConfigurations, ec2:DescribeVpcEndpointServicePermissions, ec2:DescribeVpcEndpointServices, ec2:DescribeVpcEndpoints, ec2:DescribeVpcPeeringConnections, ec2:DescribeVpcs, ec2:DescribeVpnConnections, ec2:DescribeVpnGateways, ec2:DescribePublicIpv4Pools, ec2:DescribeIpv6Pools, ec2:DetachInternetGateway,	

		ec2:DetachNetworkInterface, ec2:DetachVpnGateway, ec2:DisableVgwRoutePropagation, ec2:DisableVpcClassicLinkDnsSupport, ec2:DisassociateAddress, ec2:DisassociateRouteTable, ec2:DisassociateSubnetCidrBlock, ec2:DisassociateVpcCidrBlock, ec2:EnableVgwRoutePropagation, ec2:EnableVpcClassicLinkDnsSupport, ec2:ModifyNetworkInterfaceAttribute, ec2:ModifySecurityGroupRules, ec2:ModifySubnetAttribute, ec2:ModifyVpcAttribute, ec2:ModifyVpcEndpoint, ec2:ModifyVpcEndpointConnectionNotification, ec2:ModifyVpcEndpointServiceConfiguration, ec2:ModifyVpcEndpointServicePermissions, ec2:ModifyVpcPeeringConnectionOptions, ec2:ModifyVpcTenancy, ec2:MoveAddressToVpc, ec2:RejectVpcEndpointConnections, ec2:ReleaseAddress, ec2:ReplaceNetworkAclAssociation, ec2:ReplaceNetworkAclEntry, ec2:ReplaceRoute, ec2:ReplaceRouteTableAssociation, ec2:ResetNetworkInterfaceAttribute, ec2:RestoreAddressToClassic, ec2:UnassignIpv6Addresses, ec2:UnassignPrivateIpAddresses, ec2:UpdateSecurityGroupRuleDescriptionsEgress, ec2:UpdateSecurityGroupRuleDescriptionsIngress, logs:DescribeLogGroups, logs:DescribeLogStreams, logs:GetLogEvents, route53:*, route53domains:*, sns:CreateTopic, sns:ListSubscriptionsByTopic, sns:ListTopics, ec2:AcceptVpcPeeringConnection, ec2:AttachClassicLinkVpc, ec2:AuthorizeSecurityGroupEgress, ec2:AuthorizeSecurityGroupIngress, ec2:CreateVpcPeeringConnection, ec2:DeleteCustomerGateway, ec2:DeleteDhcpOptions, ec2:DeleteInternetGateway, ec2:DeleteNetworkAcl, ec2:DeleteNetworkAclEntry, ec2:DeleteRoute, ec2:DeleteRouteTable, ec2:DeleteSecurityGroup, ec2:DeleteVolume,	

		ec2:DeleteVpcPeeringConnection, ec2:DetachClassicLinkVpc, ec2:DisableVpcClassicLink, ec2:EnableVpcClassicLink, ec2:GetConsoleScreenshot, ec2:RejectVpcPeeringConnection, ec2:RevokeSecurityGroupEgress, ec2:RevokeSecurityGroupIngress, ec2:CreateLocalGatewayRoute, ec2:CreateLocalGatewayRouteTableVpcAssociation, ec2:DeleteLocalGatewayRoute, ec2:DeleteLocalGatewayRouteTableVpcAssociation, ec2:DescribeLocalGatewayRouteTableVirtualInterfaceGroupAssociations, ec2:DescribeLocalGatewayRouteTableVpcAssociations, ec2:DescribeLocalGatewayRouteTables, ec2:DescribeLocalGatewayVirtualInterfaceGroups, ec2:DescribeLocalGatewayVirtualInterfaces, ec2:DescribeLocalGateways, ec2:SearchLocalGatewayRoutes, s3:GetBucketLocation, s3:GetBucketWebsite, s3:ListBucket, iam:GetRole, iam:ListRoles, iam:PassRole, ec2:AcceptTransitGatewayVpcAttachment, ec2:AssociateTransitGatewayRouteTable, ec2:CreateTransitGateway, ec2:CreateTransitGatewayRoute, ec2:CreateTransitGatewayRouteTable, ec2:CreateTransitGatewayVpcAttachment, ec2:DeleteTransitGateway, ec2:DeleteTransitGatewayRoute, ec2:DeleteTransitGatewayRouteTable, ec2:DeleteTransitGatewayVpcAttachment, ec2:DescribeTransitGatewayAttachments, ec2:DescribeTransitGatewayRouteTables, ec2:DescribeTransitGatewayVpcAttachments, ec2:DescribeTransitGateways, ec2:DisableTransitGatewayRouteTablePropagation, ec2:DisassociateTransitGatewayRouteTable, ec2:EnableTransitGatewayRouteTablePropagation, ec2:ExportTransitGatewayRoutes, ec2:GetTransitGatewayAttachmentPropagations, ec2:GetTransitGatewayRouteTableAssociations, ec2:GetTransitGatewayRouteTablePropagations, ec2:ModifyTransitGateway, ec2:ModifyTransitGatewayVpcAttachment, ec2:RejectTransitGatewayVpcAttachment, ec2:ReplaceTransitGatewayRoute, ec2:SearchTransitGatewayRoutes	
SingleStore FlowAdmin	Custom Role	s3:ListBucket, s3:PutObject,	Able to manage

		s3:GetObject, s3:DeleteObject, s3:GetBucketLocation, s3:PutObjectAcl, secretsmanager:GetSecretValue, secretsmanager:DescribeSecret, secretsmanager:PutSecretValue, secretsmanager:UpdateSecret	SingleStore Flow Configurations
AmazonS3	Resource Based Policy	s3:PutObject, s3:GetObject, s3:DeleteObject, s3:GetBucketLocation, s3:PutObjectAclResource: arn:aws:s3:::<bucket-name>, arn:aws:s3:::<bucket-name>/*	To manage bucket level permissions, resource-based policy for S3 must be applied to restrict the bucket level access. The policy is attached to the bucket, but the policy controls access to both the bucket and the objects in it.
AmazonEC2	Resource Based Policy	ec2:AcceptVpcEndpointConnections, ec2:AcceptVpcPeeringConnection, ec2:AssociateIamInstanceProfile, ec2:CreateTags, ec2:DescribeTags, ec2:RebootInstancesResource: arn:aws:ec2:<ec2_instance_id>	To manage instance level permissions, resource-based policy for EC2 must be applied to restrict the access for the EC2 instance.

Environment Preparation

The following are the steps to prepare an environment for SingleStore Flow in AWS :

1. **Create an AWS Account**: To prepare the environment for SingleStore Flow in AWS, you need an AWS account. If you already have an AWS account, proceed to the next step. If not, refer to How do I create and activate a new AWS account? to create one.

2. **Create an IAM User**: SingleStore recommends creating a separate IAM user for managing all AWS services. Do not use the root user for any tasks. Refer to Setting up your AWS account to create an IAM admin user.

3. **Create and Assign Policies to the User**: Use the AWS Management Console to create a customer-managed policy and attach it to the IAM user based on their role. This policy enables the IAM user to sign in directly to the AWS Management Console with the assigned permissions.

4. **Sign in to AWS**:
    a. **As an IAM User**: Sign in to the AWS Management Console using your Account ID or account alias, along with your username and password. Refer to Sign in to the AWS Management Console for more information.
    b. **AWS SSO**: Sign in using IAM Identity Center (AWS SSO). Refer to How to create and manage users within AWS IAM Identity Center for more information.

5. **Create a VPC**: A Virtual Private Cloud (VPC) is a dedicated virtual network for your AWS account. It is logically isolated from other virtual networks in the AWS Cloud. You can launch your SingleStore Flow application and related AWS resources, such as Amazon EC2 instances, into this VPC. Refer to Configure a virtual private cloud for instructions on creating a VPC.

6. **Create a Private Subnet in Your VPC**: Since Ingest needs to be set up in your VPC, SingleStore recommends creating a new private subnet within the VPC for SingleStore Flow. Refer to Create a VPC to create a subnet.

7. **Create a Security Group**: A security group acts as a virtual firewall for your instances that controls inbound and outbound traffic. Security groups apply at the instance level, not the subnet level. Each instance in a subnet can be assigned different security groups. SingleStore does not recommend using the default security group, which AWS assigns if you don't specify a security group at launch. You can add rules to each security group to control inbound and outbound traffic. Refer to Control traffic to your AWS resources using security groups for more information.

8. **Configure Security Group Rules**: You can add or remove rules to a security group to authorize or revoke inbound or outbound access. A rule applies either to inbound traffic (ingress) or outbound traffic (egress). You can grant access to a specific CIDR range or to another security group in your VPC or in a peer VPC (with a VPC peering connection).

9. **Create an IAM Role**: SingleStore Flow uses an IAM role assigned to the EC2 instance where the application is hosted. The EC2 role must have the necessary policies attached. Refer to Create a role to delegate permissions to an IAM user for instructions on creating an IAM role for SingleStore Flow. Attach the required policies to the newly created IAM role.

10. **Assign the Role to Users or Groups**: The IAM role must be assigned to an AWS Directory Service user or group. The role must have a trust relationship with AWS Directory Service. Refer to Assigning users or groups to an existing IAM role to assign

users or groups to an IAM role.

11. **Create Access Key ID and Secret Access Key**: SingleStore Flow uses an access key ID and secret access key to connect to AWS services from an on-premises server. SingleStore recommends generating a set of access keys for the SingleStore Flow user account. The following are the steps to create access keys from the admin user account:

    a. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
    b. In the navigation pane, select **Users**.
    c. Select the **separate SingleStore Flow** user whose access keys you want to manage, then select the **Security credentials** tab.
    d. In **Access keys**, select **Create access key**. Then, select **Download .csv file** to save the access key ID and secret access key to a CSV file. Store this file securely, as the secret access key cannot be accessed again after closing the dialog.
    e. After downloading the CSV file, select **Close**. The access key is active by default, and you can use it. Refer to AWS security credentials for more information.
    Note: SingleStore recommends rotating access keys every 90 days. Refer to Manage Access Keys for more information.

12. **Create an Auto-Scaling Group**: When deploying Ingest in a high-availability (HA) environment, SingleStore recommends configuring your EC2 instances with an Auto Scaling Group. Follow the steps mentioned in Create an Auto Scaling group using the Amazon EC2 launch wizard to launch an Auto Scaling group via the AWS Console. The recommended parameters for configuring the Auto Scaling group are:

    a. Select the instance type recommended by SingleStore under the Prerequisites section.
    b. Set the minimum number of instances to 2 for a high-availability deployment.
    c. Select the IAM role as `SingleConnectEc2Role`.
    d. Configure the storage as recommended under the Additional AWS Services section.
    e. Select the security group created for SingleStore Flow in the previous steps.
    f. Select the key pair for the launch configuration.
    g. Refer to Create an Auto Scaling group using the Amazon EC2 launch wizard for remaining steps.

Create an EC2 System

Refer to Get started with Amazon EC2 for more information.

Recommended Network ACL Rules for EC2

The following table presents the recommended rules for your EC2. These rules block all traffic except for what is explicitly required.

The EC2 security group must include the necessary inbound and outbound rules as per the following:

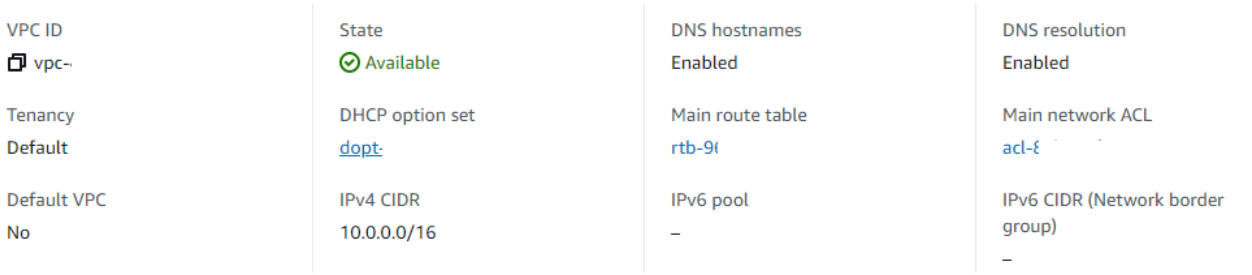
Inbound					
Rule #	**Source IP**	**Protocol**	**Port**	**Allow / Deny**	**Comments**
1	Custom IP which requires access to the SingleStore Ingest Application	TCP	80	ALLOW	Allows inbound HTTP traffic only from known/custom IPv4 addresses.
2	Public IPv4 address range of your home network	TCP	22	ALLOW	Allows inbound SSH traffic from your home network over the Internet gateway.
3	Public IPv4 address range of your home network	TCP	3389	ALLOW	Allows inbound RDP traffic from your home network over the Internet gateway.
4	0.0.0.0/0	all	all	DENY	Denies all inbound IPv4 traffic not handled by a preceding rule (not modifiable).
Outbound					
Rule #	**Dest IP**	**Protocol**	**Port**	**Allow / Deny**	**Comments**
1	Source DB Host IP address	TCP	Custom port (port specific to source database ports)	ALLOW	Allows connections to the source database.

2	Singlestore Cluster Host IP address	TCP	8080 (port specific to SingleStore destination database)	ALLOW	Allows connections to the SingleStore destination database.
3	0.0.0.0/0	all	all	DENY	Denies all outbound IPv4 traffic not handled by a preceding rule (not modifiable).

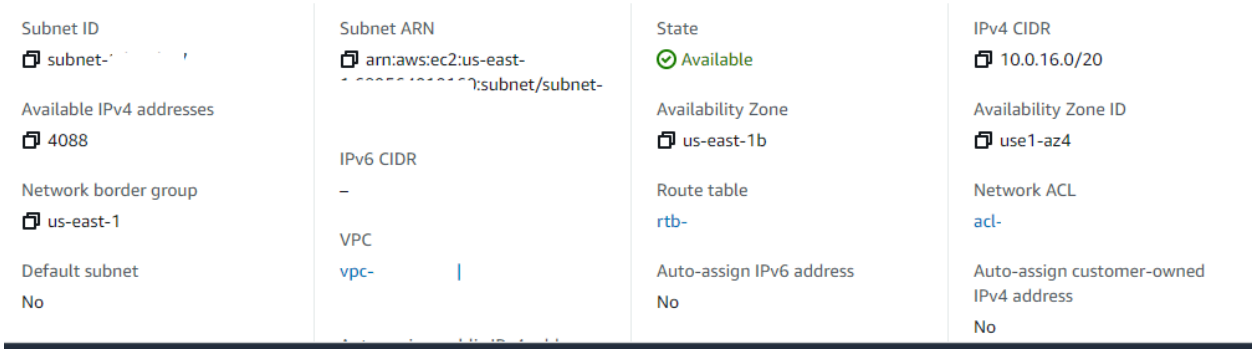
To open ports on the Amazon Console, refer to What is Amazon EC2? and follow the steps to allow inbound traffic to your Amazon instance.

To open ports on Windows Server, refer to Opening Ports in the Firewall and follow the steps to allow inbound traffic to your server.

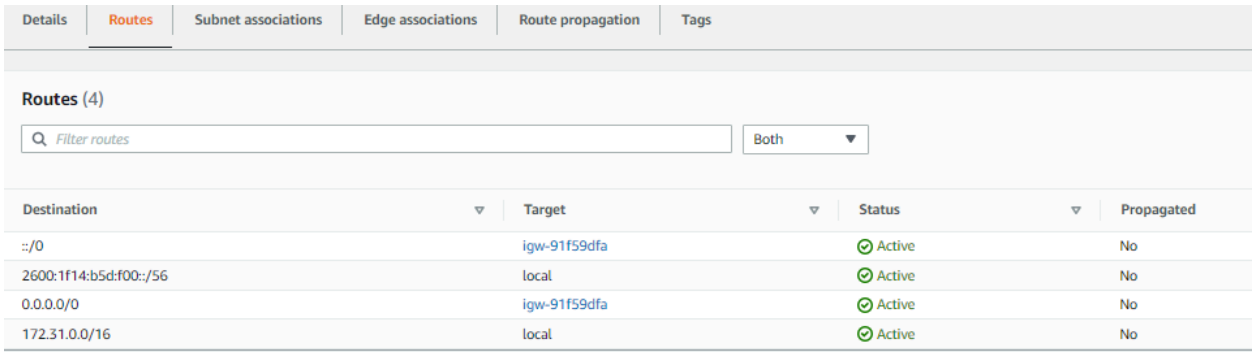
VPC Details

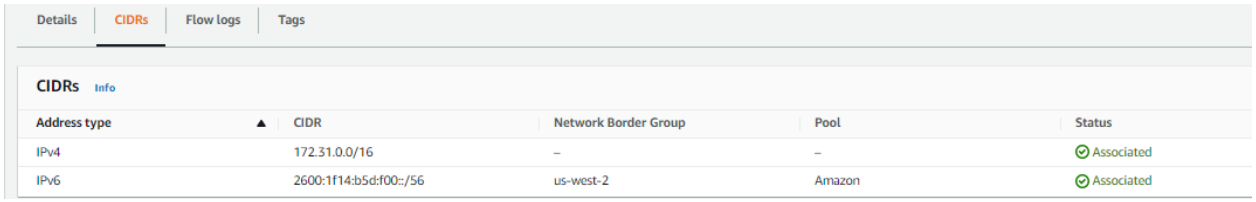


Related Subnet Details



The following are the reference details for the route table and CIDRs:





Outbound Connections

SingleStore Ingest connects to any source endpoints outside its VPC using NAT, VPN, or API Gateways.

A NAT gateway is a Network Address Translation (NAT) service. You can use a NAT gateway to allow instances in a private subnet to connect to services outside your VPC, while preventing external services from initiating a connection with those instances. Refer to NAT gateways for more information.

To connect the VPC to a remote network and enable source/destination endpoint connections, use AWS VPN. For more details, refer to NAT gateways for more information.

Additional AWS Services

SingleStore Flow uses several AWS resources to fulfill user requirements. The costs of these services are separate from SingleStore Flow charges and are billed by AWS to your account.

The following list provides an overview of other billable services within SingleStore Flow. Use the AWS Pricing Calculator to estimate the cost of additional AWS resources.

A sample estimate for a high availability setup with a source data volume of 100 GB is provided for reference. Note that

Note: Not all services are mandatory, and the size and number of services varies for each customer environment. The sample is for reference purposes only. All AWS services have service limits. Check for sufficient resources before launching services, and if necessary, refer to Request a quota increase to request an increase in quota. Refer to Service endpoints and quotas to check the service limits for each service.

Service	Mandatory	Billing Type	Service Limits
AWS EC2	Y	Pay-as-you-go	Check EC2 quota here
Additional EBS storage attached to EC2	Y	Based on size	
AWS S3	N	Pay-as-you-go	Check Amazon S3 quota here
AWS CloudWatch Logs and metrics	N	Pay-as-you-go	Check EC2 quota here
AWS SNS	N	Pay-as-you-go	Check AWS SNS quota here
AWS Dynamo DB (5 WCUs /5 RCUs)	N	Pay-as-you-go	Check Dynamo DB quota here
AWS Lambda	N	Pay-as-you-go	Check AWS Lambda quota here
AWS KMS	N	Pay-as-you-go	Check Amazon KMS quota here

SingleStore recommends to use the following mentioned instance types for EC2 with EBS volumes attached:

EC2 Instance Type	SingleStore Flow Enterprise Edition	Recommended EBS volumes	EBS Volume Type
t3.large	Volume < 100 GB	100 GB	General Purpose SSD (gp2) Volumes
t3.xlarge	Volume >100 and < 300 GB	500 GB	General Purpose SSD (gp2) Volumes
t3.2xlarge	Volume > 300 GB and < 1 TB	500 GB	General Purpose SSD (gp2) Volumes

IMDS Settings and Recommendations

SingleStore Flow uses the latest version of the AWS SDK in each release. It uses IMDSv2 for all API calls to AWS services. SingleStore recommends disabling IMDS after deployment if required. Refer to Use the Instance Metadata Service to access instance metadata for more information on how to disable IMDS.

To modify this using the AWS CLI, run the following command:

```
aws ec2 modify-instance-metadata-options \
--instance-id <instance-id> \
--http-endpoint disabled
```

Manage Access Keys

SingleStore Flow uses an access key and secret key to authenticate to AWS services like S3, CloudWatch, DynamoDB, and others. It requires the AWS access key ID and secret key to access S3 and other services from on-premises. AWS IAM roles are used when installing on an EC2 instance.

For security reasons, SingleStore recommends rotating access keys and KMS keys periodically, for example, every 90 days. After generating new keys, update them in Ingest's configuration. Follow these steps:

- Open the Ingest instance that needs the new key in a web browser.
- Go to the **Schedule** tab and stop the replication schedule for Ingest by disabling **Schedule**.
- Navigate to **Connections > Destination File System**.
- Enter the new **Access Key** and **Secret Access Key** and select **Apply**.
- Once the keys are saved, resume the replication by enabling **Schedule**.

Refer to Rotate AWS KMS keys for more information on key rotation.

The IAM role for SingleStore Flow must have the recommended policies attached. Refer to AWS Identity and Access Management (IAM) for SingleStore Flow for the list of policies and permissions.

Data Security and Encryption

SingleStore ensures data security through various mechanisms, including encryption.

With AWS KMS, Ingest uses a customer-specified KMS key to encrypt customer data on AWS services such as EC2, Secrets Manager, and DynamoDB. You can configure the customer KMS ID in SingleStore Flow, which is used to encrypt data stored on these AWS services.

Ingest also supports server-side encryption using AES-256. Amazon S3 server-side encryption employs 256-bit Advanced Encryption Standard (AES-256), one of the strongest block ciphers available, to encrypt your data. SingleStore Flow supports this encryption by default.

Ingest does not store any data outside the customer's designated environment. It stores data in the following AWS services, depending on customer requirements:

- **Amazon EC2**: Used only for temporary staging and pipeline configuration. Refer to Data protection in Amazon EC2 for enabling encryption.
- **EBS Storage**: Used only for temporary staging and pipeline configuration. Refer to Amazon EBS data security for enabling encryption.
- **Amazon Aurora DB**: Used for exporting pipeline metadata. Refer to Encrypting Amazon Aurora resources for enabling encryption.
- **Amazon DynamoDB**: Used when configured for High Availability. Refer to DynamoDB encryption at rest for enabling encryption.
- **AWS Secrets Manager**: Used for storing all pipeline credentials. Refer to AWS Key Management Service for encryption details.

For non-AWS destination endpoints:

- **SingleStore**: Data encryption is handled within SingleStore.

Key Rotation

SingleStore recommends rotating all keys configured in Ingest every 90 days for security reasons. This includes credentials for all source and destination endpoints. The following are some references for AWS services for more details:

- Rotate AWS KMS keys
- Encryption key rotation

Follow the recommendations for all non-AWS sources and destinations:

External Applications	Reference for Key Rotation
Oracle	Oracle Password Rotation
MS SQL Server	MS SQL Server password rotation
MySQL	MySQL password rotation
PostgreSQL	PostgreSQL password rotation

Configure Data Encryption

SingleStore follows AWS recommendations for encrypting data both at rest and in transit. This is achieved by creating keys and certificates used for encryption.

AWS Secrets Manager uses encryption through AWS KMS. Refer to Secret encryption and decryption in AWS Secrets Manager for more information.

Encryption In-Transit

SingleStore Flow uses SSL to establish secure connections (e.g., with AWS services, databases, etc.) for data flow that ensures secure communication in transit.

SSL involves managing security certificates. It is important to keep these certificates active at all times to ensure uninterrupted service.

AWS Certificate Manager (ACM) handles the complexity of creating and managing public SSL/TLS certificates. Customers can set up notifications to be alerted before the certificate expiration date and can renew certificates in advance to ensure uninterrupted service. Refer to Managed certificate renewal in AWS Certificate Manager for more information on managing ACM.

Testing the Connections

Verify if the connectivity to remote services is available.

To test the remote connections, you need the Telnet utility, which must be enabled from the Control Panel under **Turn On Windows Features**.

Follow these steps:

1. Open the **Start** menu, then select **Run**.
2. Type **CMD** and select **OK**.
3. At the command prompt, run the following command:

   ```
   telnet <IP address or Hostname> <Port number>
   ```

   For example:

   ```
   telnet 192.168.1.1 8081
   ```

   a. If the connection is unsuccessful, an error message appears.
   b. If the command prompt window is blank with only the cursor visible, the connection is successful, and the service is available.

Connection error to source or destination database server.

If there is a connectivity issue with the source or destination database, check if the SingleStore Flow server can reach the remote host and port.

You can test the connection to the IP address and port using the following command:

```
telnet <IP address or Hostname> <Port number>
```

Alternatively, run the PowerShell command to verify the connection:

```
tnc <IP address or Hostname> <Port number>
```

Unable to start Windows service

Error: Unable to start the Windows service 'SingleStore Ingest'.

Resolution: If Java is not installed or the system path is not updated, the Ingest service fails to start. Install Java 21 or add the Java path to the system path. To verify, open CMD and run the following command:

```
java -version
```

If the response is ‘`unable to recognize command`’, check the Java path in the Environment Variables under `'Path'` and update it to the correct path.

Application not able to launch

Error: The SingleStore Ingest service is installed and started, but the application does not launch in the browser.

Resolution: SingleStore Flow requires Java21 to function. Please install the correct version of Java and restart the service.

If Java 11 is installed, the Ingest service starts, but the page displays an error message.

To verify the Java version, open CMD and run the following command:

```
java -version
```

Expected result:

```
java version "any_build"
```

For example:

```
openjdk version "21.0.5" 2024-10-15 LTS

OpenJDK Runtime Environment Corretto-21.0.5.11.1 (build
21.0.5+11-LTS)

OpenJDK 64-Bit Server VM Corretto-21.0.5.11.1 (build 21.0.5+11-LTS,
mixed mode, sharing)
```

If the Java version is lower, uninstall Java and install the required version (Java 21).

Grants not available on the database

Error: Cannot open database `'demo'` requested by the login.

Resolution: The user does not have the necessary grants to connect to the database. Apply the correct grants to the user and try again.

Login failed for user

Error: Login failed for user `'Demo'`.

Resolution: The user does not exist, or there is a typo in the username. Alternatively, the password may be incorrect. Verify the credentials and try again.

MS SQL Server as a Source Connector

Follow the recommended steps to set up your MS SQL source connector.

Preparing MS SQL Server

The MS SQL Server setup depends on the selected replication option: Change Tracking or Change Data Capture. Refer to Prerequisites for MS SQL Server to set up the selected replication option.

Security for MS SQL Server

The Ingest database replication login user must have `VIEW CHANGE TRACKING` permission to view the Change Tracking information.

To review all change tracking tables (enabled or disabled), run the following command:

```
SELECT *

FROM sys.all_objects

WHERE object_id IN (SELECT object_id

FROM sys.change_tracking_tables

WHERE is_track_columns_updated_on = 1);
```

Verification of MS SQL Server Source

To verify if Change Tracking is already enabled on the database, run the following SQL command:

```
--Review all change tracking tables that are = 1 enabled, or = 0
disabled

SELECT *

FROM sys.change_tracking_databases

WHERE database_id = DB_ID('databasename');
```

The following SQL command lists all tables with Change Tracking enabled for the selected database:

```
USE databasename;
SELECT sys.schemas.name AS schema_name,
       sys.tables.name AS table_name
FROM sys.change_tracking_tables
JOIN sys.tables ON sys.tables.object_id =
sys.change_tracking_tables.object_id
JOIN sys.schemas ON sys.schemas.schema_id = sys.tables.schema_id;
```

Data Types in MS SQL Server

Ingest supports most MS SQL Server data types. The following is a list of supported data types:

BIGINT	REAL	VARCHAR (max)
BIT	FLOAT	NCHAR
DECIMAL	DATETIME	NVARCHAR (length)
INT	DATETIME2	NVARCHAR (max)
MONEY	SMALLDATETIME	BINARY
NUMERIC (p,s)	DATE	VARBINARY
SMALLINT	TIME	VARBINARY (max)
SMALLMONEY	DATETIMEOFFSET	TIMESTAMP
TINYINT	CHAR	UNIQUEIDENTIFIER
VARCHAR	HIERARCHYID	XML

Oracle DB as a Source Connector

Follow the recommended steps to set up your Oracle source connector.

Preparing Oracle on Amazon RDS

Enable Change Tracking for a Database on Amazon Oracle RDS

In Oracle on Amazon RDS, you must enable supplemental logging at the database level. Supplemental logging is required to log additional details in the archive logs.

To turn on supplemental logging at the database level, run the following command:

```
exec rdsadmin.rdsadmin_util.alter_supplemental_logging('ADD', 'ALL');
```

To retain archived redo logs on your DB instance, run the following command (for example, 24 hours):

```
exec rdsadmin.rdsadmin_util.set_configuration('archivelog retention
hours', 24);
```

To enable supplemental logging at the table level, execute the following statement:

```
ALTER TABLE <schema>.<tablename> ADD SUPPLEMENTAL LOG DATA (ALL)
COLUMNS;
```

Preparing On-premises Oracle

Enable Change Tracking for an On-Premises Oracle Server

To enable change tracking on an on-premises Oracle server, the Oracle database must be in `ARCHIVELOG` mode, and supplemental logging must be enabled at the database level to log additional details in the archive logs.

To enable supplemental logging at the database level, run the following command:

```
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (ALL) COLUMNS;
```

Alternatively, to enable minimal database supplemental logging, run the following command:

```
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;

ALTER DATABASE FORCE LOGGING;
```

To enable supplemental logging at the table level, run the following command:

```
ALTER TABLE <schema>.<tablename> ADD SUPPLEMENTAL LOG DATA (ALL)
COLUMNS;
```

Security for Oracle

The Oracle user running Ingest must have the following security privileges:

- `SELECT` access on all tables to be replicated.

The following command must return records:

```
SELECT * FROM V$ARCHIVED_LOG;
```

If no records are returned, ensure that `SELECT` access on `V_$ARCHIVED_LOG` is granted or verify that the database is in `ARCHIVELOG` mode.

The following security permissions must be assigned to the user:

- `CREATE SESSION`

- SELECT access on V_$LOGMNR_CONTENTS
- SELECT access on V_$LOGMNR_LOGS
- SELECT access on ANY TRANSACTION
- SELECT access on DBA_OBJECTS
- EXECUTE access on DBMS_LOGMNR

Run the following grant commands for <user>:

```
GRANT SELECT ON V_$ARCHIVED_LOG TO <user>;

GRANT SELECT ON V_$LOGMNR_CONTENTS TO <user>;

GRANT EXECUTE ON DBMS_LOGMNR TO <user>;

GRANT SELECT ON V_$LOGMNR_LOGS TO <user>;

GRANT SELECT ANY TRANSACTION TO <user>;

GRANT SELECT ON DBA_OBJECTS TO <user>;
```

Verification of Oracle Source

To verify if Oracle is set up correctly for change detection, run the following commands and check the expected results.

Condition to be checked	**SQL Command**	**Expected Result**
Is ArchiveLog mode enabled?	`SELECT log_mode FROM V$DATABASE;`	ARCHIVELOG
Is Supplemental logging turned on at database level?	`SELECT supplemental_log_data_min FROM V$DATABASE;`	YES

Is Supplemental Logging turned on at table level?	SELECT log_group_name, table_name, always, log_group_type FROM dba_log_groups;	RESULT <log group name>, <table name>, ALWAYS, ALL COLUMN LOGGING

Data Types in Oracle

Ingest supports most Oracle data types. The following table lists the supported data types:

BINARY_DOUBLE	BINARY_FLOAT	CHAR
DATE	INTERVAL DAY TO SECOND	LONG
LONG RAW	NCHAR	NUMBER
NVARCHAR	RAW	REF
TIMESTAMP	TIMESTAMP WITH LOCAL TIME ZONE	VARCHAR2

MySQL as a Source Connector

Preparing On-premises MySQL

To prepare MySQL for change tracking, enable binary logging. Configure the following parameters in the `my.ini` file on MySQL for Windows or in the `my.cnf` file on MySQL for UNIX:

Parameter	**Value**
server_id	Any value from 1. E.g. server_id = 1
log_bin=<path>	Path to the binary log file. E.g. log_bin = D:\MySQLLogs\BinLog

`binlog_format`	`binlog_format=row`
`expire_logs_days`	To avoid disk space issues it is strongly recommended not to use the default value (`0`). E.g. `expire_log_days = 4`
`binlog_checksum`	This parameter can be set to `binlog_checksum=none`. SingleStore Flow does support `CRC32` as well
`binlog_row_image`	`binlog_row_image=full`

Preparing MySQL on Amazon RDS

Enabling Change Tracking on MySQL on Amazon RDS. To enable change tracking for MySQL on Amazon RDS, follow these steps:

1. In the AWS Management Console, create a new DB parameter group for MySQL.
2. Configure the following parameters:

Parameter	**Value**
`binlog_format`	`binlog_format=row`
`binlog_checksum`	Set to `binlog_checksum=none` or `binlog_checksum=CRC32`

The MySQL RDS DB instance must use the newly created DB parameter group for binary logging to be enabled.

Security for MySQL

The Ingest user must have the following privileges:

- `REPLICATION CLIENT`
- `REPLICATION SLAVE`
- `SELECT` privileges on the source tables designated for replication

To grant the necessary permissions to a MySQL user, run the following commands:

```
CREATE USER 'bflow_ingest_user' IDENTIFIED BY '*****';

GRANT SELECT, REPLICATION CLIENT, SHOW DATABASES ON *.* TO
bflow_ingest_user;
```

```
GRANT SELECT, REPLICATION SLAVE, SHOW DATABASES ON *.* TO
bflow_ingest_user;
```

Note:

If the source DB is an Amazon RDS MySQL DB, download `mysqlbinlog.exe` and add its directory path to the `PATH` variable in the Windows Environment Variables on the SingleStore Flow machine.

PostgreSQL DB as a Source Connector

Preparing PostgreSQL DB

1. Use a PostgreSQL database version 9.4.x or later.
2. Add the IP address of the SingleStore Flow machine to the `pg_hba.conf` configuration file, using the **replication** keyword in the database field. For example:

   ```
   host replication all 189.452.1.212/32 trust
   ```

3. Set the following parameters and values in the `postgresql.conf` configuration file:
   a. `wal_level = logical`
   b. `max_replication_slots:` Set this to a value greater than 1. The value must match the number of tasks you want to run. For example, to run four tasks, set this to at least 4. Slots open automatically as soon as a task starts and remain open even when the task ends. You need to manually delete unused slots.
   c. `max_wal_senders`: Set this to a value greater than 1. This parameter controls the number of concurrent replication tasks.
   d. `wal_sender_timeout = 0` : This parameter terminates replication connections that remain inactive for more than the specified time. Although the default is 60 seconds, SingleStore recommends setting this to 0 to disable the timeout mechanism.

   Note: After modifying these parameters, restart PostgreSQL for the changes to take effect.

4. Grant superuser permissions to the user account specified for the PostgreSQL source database. Superuser privileges are necessary to access replication-specific functions.

Start and Stop Ingest

To Start Ingest:

1. Start the Ingest service using Windows Services or Windows Task Manager.
2. Open Chrome and enter the URL: `localhost:8081`

To Stop Ingest:

1. Go to Windows Services
2. Stop the Ingest service.

Configuration of Ingest

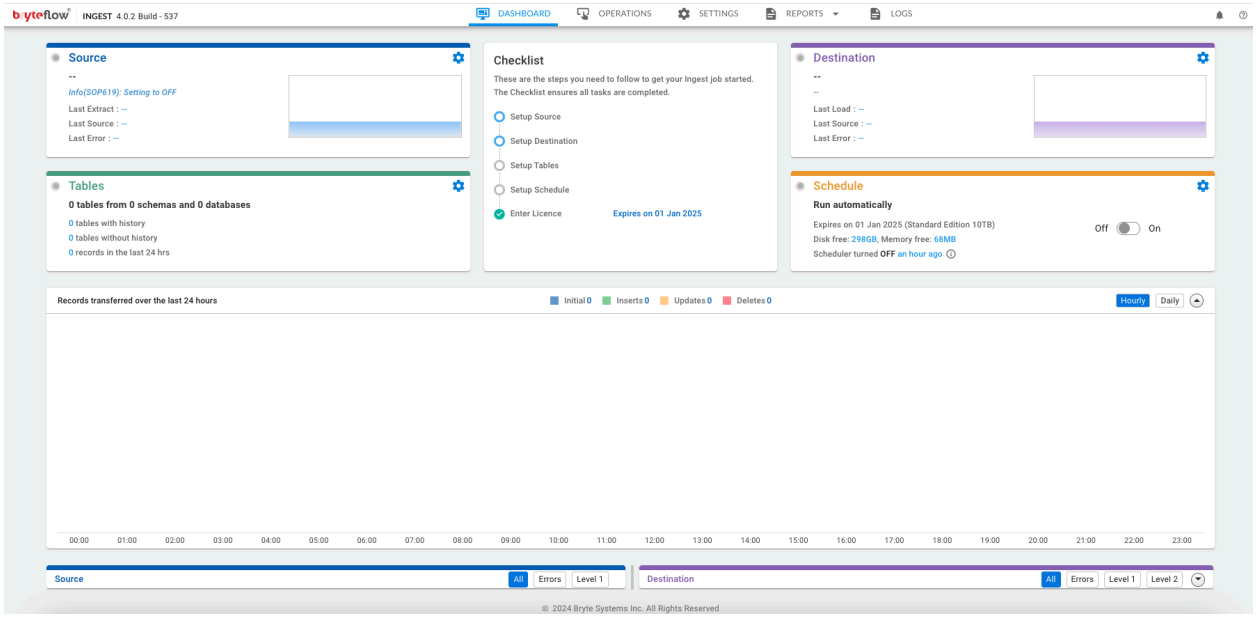
You can configure Ingest through the web console. To access the console, enter the following URL in Chrome: `localhost:8081`

The screen displays the following tabs:

- Dashboard
- Operations
- Settings
- Reports
- Logs

Dashboard

The Dashboard provides a central screen to monitor the overall status of the Ingest instance.



- Dashboard provides quick access to configure Ingest, including the following:
    - **Source** Database
    - **Destination** Database
    - **Tables**
    - **Schedule** Tasks

- The **Checklist** gives the summary of the steps that are needed to complete to start an Ingest job.
- The graph summary displays the number of records transferred.
    - When **Hourly** is selected, you can view transfer statistics for the past 24 hours.
    - When **Daily** is selected, the monthly statistics are shown.
    - The bar graph illustrates the process status:
        - Initial Extract is represented by Blue.
        - Inserts are represented by Green.
        - Updates are represented by Yellow.
        - Deletes are represented by Red.
    - Hovering over the bar graph shows the exact number of records transferred.
- You can view the log for **Source** and **Destination** database at the bottom.

Source Database

You can configure the following databases as a source: Oracle, MS SQL Server, MySQL, and PostgreSQL. To configure the source database, navigate to **Dashboard > Source**, and then select the gear icon.

Oracle DB Configuration

1. In the **Source Database** dialog, select **Oracle** from the **Source Database** list.
2. Select the **Extract Type** from the following:
    a. **Log Miner**
    b. **Fast Log Miner**
    c. **Remote Log Miner**
    d. **Continuous Log Miner**
    e. **Full Extracts**
    f. **Timestamps**

    Note: Tabs are different for **Remote Log Miner** extract type.
3. In the **Database** tab, enter the following configuration details:
    a. **Host Name**: Enter the IP address or hostname of the database server.
    b. **Port**: Enter the port number on which the database server is listening (default port for Oracle is 1521).
    c. **Database Name**: Enter the name of your database.
    d. **User Id**: Enter a valid Oracle database user ID to be used with Ingest.
    e. **Password**: Enter the password and confirm it by re-entering it in the **Confirm Password** field.
    Note: Passwords are encrypted within Ingest.
    f. **JDBC Options** (Optional): JDBC options are optional and can be used to extend the JDBC URL for accessing the database.
4. Enter the configuration details for **Remote Log Miner** extract type:
    a. Select the **Source Database** tab, and enter the following configuration details:
        i. **Host Name**: Enter the IP address or hostname of the database server.

ii. **Port**: Enter the port number on which the database server is listening (default port for Oracle is 1521).
iii. **Database Name**: Enter the Oracle SID.
Note: When using SID to connect to a dedicated Oracle server instance, use **:SID**.
iv. **User Id**: Enter a valid Oracle database user ID to be used with Ingest.
v. **Password**: Enter the password and confirm it by re-entering it in the **Confirm Password** field.
Note: Passwords are encrypted within Ingest.
vi. **JDBC Options** (Optional): JDBC options are optional and can be used to extend the JDBC URL for accessing the database.

b. Select the **Mining Database** tab, and enter the following configuration details:
i. **Host Name**: Enter the IP address or hostname of the database server.
ii. **Port**: Enter the port number on which the database server is listening (default port for Oracle is 1521).
iii. **Database Name**: Enter the Oracle SID.
Note: When using SID to connect to a dedicated Oracle server instance, use **:SID**.
iv. **User Id**: Enter a valid Oracle database user ID to be used with Ingest.
v. **Password**: Enter the password and confirm it by re-entering it in the **Confirm Password** field.
Note: Passwords are encrypted within Ingest.
vi. **JDBC Options** (Optional): JDBC options are optional and can be used to extend the JDBC URL for accessing the database.

5. Select the **Advanced Options** tab. The configuration details vary for each **Extract Type**.

Extract Type	**Configurations in Advanced Options**
Log Miner	• Log file catchup count • Log catchup time (mins) • Log catchup offset (mins) • Log file look-ahead • Convert RAW to Hex
Fast Log Miner	• Extract Threads • Log file catchup count • Log catchup time (mins) • Log catchup offset (mins) • Log file look-ahead • Convert RAW to Hex
Remote Log Miner	• Extract Threads • Log file catchup count • Log catchup time (mins) • Log catchup offset (mins)

	• Log file look-ahead • Convert RAW to Hex
Continuous Log Miner	• Convert RAW to Hex
Full Extracts	• Extract Threads • Convert RAW to Hex
Timestamps	• Extract Threads • Convert RAW to Hex

Refer to Appendix: Additional Configurations for more information.

6. Select **Test** to verify the connectivity.
7. Select **Apply** to confirm and save the details.

Oracle Pluggable DB Configuration

1. In the **Source Database** dialog, select **Oracle(Pluggable Database)** from the **Source Database** list.
2. Select the **Extract Type** from the following:
    a. **Fast Log Miner**
    b. **Continuous Log Miner**
3. In the **Database** tab, enter the following configuration details:
    a. **Host Name**: Enter the IP address or hostname of the database server.
    b. **Port**: Enter the port number on which the database server is listening (default port for Oracle is 1521).
    c. **Database Name**: Enter the Oracle SID.
    Note: When using SID to connect to a dedicated Oracle server instance, use **:SID**.
    d. **User Id**: Enter a valid Oracle pluggable database user ID to be used with Ingest.
    e. **Password**: Enter the password and confirm it by re-entering it in the **Confirm Password** field.
    Note: Passwords are encrypted within Ingest.
    f. **JDBC Options** (Optional): JDBC options are optional and can be used to extend the JDBC URL for accessing the database.
4. In the **Root Database** tab, enter the following configuration details:
    a. **Host Name**: Enter the IP address or hostname of the database server.
    b. **Port**: Enter the port number on which the database server is listening (default port for Oracle is 1521).
    c. **Database Name**: Enter the root container database name.
    d. **User Id**: Enter a valid root container user ID to be used with Ingest.
    e. **Password**: Enter the password and confirm it by re-entering it in the **Confirm Password** field.
    Note: Passwords are encrypted within Ingest.

f. **JDBC Options** (Optional): JDBC options are optional and can be used to extend the JDBC URL for accessing the database.

5. Select the **Advanced Options** tab. The configuration details vary for each **Extract Type**.

Extract Type	Configurations in Advanced Options
Fast Log Miner	• Extract Threads • Log file catchup count • Log catchup time (mins) • Log catchup offset (mins) • Log file look-ahead • Convert RAW to Hex
Continuous Log Miner	• Convert RAW to Hex

Refer to Appendix: Additional Configurations for more information.

6. Select **Test** to verify the connectivity.
7. Select **Apply** to confirm and save the details.

Oracle RAC Configuration

1. In the **Source Database** dialog, select **Oracle(RAC)** from the **Source Database** list.
2. Select the **Extract Type** from the following:
   a. **Log Miner**
   b. **Continuous Log Miner**
3. In the **Database** tab, enter the following configuration details:
   a. **Host Name**: Enter the IP address or hostname of the database server.
   b. **Port**: Enter the port number on which the database server is listening (default port for Oracle is 1521).
   c. **Database Name**: Enter the Oracle SID.
   Note: When using SID to connect to a dedicated Oracle server instance, use **:SID**.
   d. **User Id**: Enter a valid Oracle pluggable database user ID to be used with Ingest.
   e. **Password**: Enter the password and confirm it by re-entering it in the **Confirm Password** field.
   Note: Passwords are encrypted within Ingest.
   f. **JDBC Options** (Optional): JDBC options are optional and can be used to extend the JDBC URL for accessing the database.
4. Select the **Advanced Options** tab. The configuration details vary for each **Extract Type**.

Extract Type	Configurations in Advanced Options
Log Miner	• Extract Threads • Log file catchup count

	• Log catchup time (mins) • Log catchup offset (mins) • Log file look-ahead • Convert RAW to Hex
Continuous Log Miner	• Convert RAW to Hex

Refer to Appendix: Additional Configurations for more information.

5. Select **Test** to verify the connectivity.
6. Select **Apply** to confirm and save the details.

Oracle RAC Pluggable DB Configuration

1. In the **Source Database** dialog, select **Oracle(RAC) (Pluggable Database)** from the **Source Database** list.
2. In the **Extract Type** list, select **Log Miner**.
3. In the **Database** tab, enter the following configuration details:
    a. **Host Name**: Enter the IP address or hostname of the database server.
    b. **Port**: Enter the port number on which the database server is listening (default port for Oracle is 1521).
    c. **Database Name**: Enter the Oracle SID.
    Note: When using SID to connect to a dedicated Oracle server instance, use **:SID**.
    d. **User Id**: Enter a valid Oracle pluggable database user ID to be used with Ingest.
    e. **Password**: Enter the password and confirm it by re-entering it in the **Confirm Password** field.
    Note: Passwords are encrypted within Ingest.
    f. **JDBC Options** (Optional): JDBC options are optional and can be used to extend the JDBC URL for accessing the database.
4. In the **Root Database** tab, enter the following configuration details:
    a. **Host Name**: Enter the IP address or hostname of the database server.
    b. **Port**: Enter the port number on which the database server is listening (default port for Oracle is 1521).
    c. **Database Name**: Enter the root container database name.
    d. **User Id**: Enter a valid root container user ID to be used with Ingest.
    e. **Password**: Enter the password and confirm it by re-entering it in the **Confirm Password** field.
    Note: Passwords are encrypted within Ingest.
    f. **JDBC Options** (Optional): JDBC options are optional and can be used to extend the JDBC URL for accessing the database.
5. Select the **Advanced Options** tab. Enter the following configuration details:
    a. Extract Threads
    b. Log file catchup count
    c. Log catchup time (mins)

    d. Log catchup offset (mins)
    e. Log file look-ahead
    f. Convert RAW to Hex

   Refer to Appendix: Additional Configurations for more information.
6. Select **Test** to verify the connectivity.
7. Select **Apply** to confirm and save the details.

Oracle 19c DB Configuration

1. In the **Source Database** dialog, select **Oracle 19c** from the **Source Database** list.
2. In the **Extract Type** list, select **Continuous Log Miner**.
3. In the **Database** tab, enter the following configuration details:
    a. **Host Name**: Enter the IP address or hostname of the database server.
    b. **Port**: Enter the port number on which the database server is listening (default port for Oracle is 1521).
    c. **Database Name**: Enter the Oracle SID.
    Note: When using SID to connect to a dedicated Oracle server instance, use **:SID**.
    d. **User Id**: Enter a valid Oracle pluggable database user ID to be used with Ingest.
    e. **Password**: Enter the password and confirm it by re-entering it in the **Confirm Password** field.
    Note: Passwords are encrypted within Ingest.
    f. **JDBC Options** (Optional): JDBC options are optional and can be used to extend the JDBC URL for accessing the database.
4. Select the **Advanced Options** tab. Enable **Convert RAW to Hex**. Refer to Appendix: Additional Configurations for more information.
5. Select **Test** to verify the connectivity.
6. Select **Apply** to confirm and save the details.

Oracle 19c Pluggable DB Configuration

1. In the **Source Database** dialog, select **Oracle 19c (Pluggable Database)** from the **Source Database** list.
2. In the **Extract Type** list, select **Continuous Log Miner**.
3. In the **Database** tab, enter the following configuration details:
    a. **Host Name**: Enter the IP address or hostname of the database server.
    b. **Port**: Enter the port number on which the database server is listening (default port for Oracle is 1521).
    c. **Database Name**: Enter the Oracle SID.
    Note: When using SID to connect to a dedicated Oracle server instance, use **:SID**.
    d. **User Id**: Enter a valid Oracle pluggable database user ID to be used with Ingest.
    e. **Password**: Enter the password and confirm it by re-entering it in the **Confirm Password** field.
    Note: Passwords are encrypted within Ingest.

f. **JDBC Options** (Optional): JDBC options are optional and can be used to extend the JDBC URL for accessing the database.

4. In the **Root Database** tab, enter the following configuration details:
    a. **Host Name**: Enter the IP address or hostname of the database server.
    b. **Port**: Enter the port number on which the database server is listening (default port for Oracle is 1521).
    c. **Database Name**: Enter the root container database name.
    d. **User Id**: Enter a valid root container user ID to be used with Ingest.
    e. **Password**: Enter the password and confirm it by re-entering it in the **Confirm Password** field.
    Note: Passwords are encrypted within Ingest.
    f. **JDBC Options** (Optional): JDBC options are optional and can be used to extend the JDBC URL for accessing the database.
5. Select the **Advanced Options** tab. Enable **Convert RAW to Hex**. Refer to Appendix: Additional Configurations for more information.
6. Select **Test** to verify the connectivity.
7. Select **Apply** to confirm and save the details.

MS SQL Server Configuration

1. In the **Source Database** dialog, select **Microsoft SQL Server** from the **Source Database** list.
2. Select the **Extract Type** from the following:
    a. **Change Tracking**
    b. **CDC**
    c. **Full Extracts**
    d. **Timestamps**
3. In the **Database** tab, enter the following configuration details:
    a. **Host Name**: Enter the IP address or hostname of the database server.
    b. **Port**: Enter the port number on which the database server is listening (default port for MS SQL Server is 1433).
    c. **Database Name**: Enter the name of your database.
    d. **User Id**: Enter a valid MS SQL Server user ID to be used with Ingest. If a Windows user is required, contact SingleStore Support for guidance.
    e. **Password**: Enter the password and confirm it by re-entering it in the **Confirm Password** field.
    Note: Passwords are encrypted within Ingest.
    f. **JDBC Options** (Optional): JDBC options are optional and can be used to extend the JDBC URL for accessing the database.
4. Select the **Advanced Options** tab. Enter the **Extract Threads**. Refer to Appendix: Additional Configurations for more information.
5. Select **Test** to verify the connectivity.
6. Select **Apply** to confirm and save the details.

MySQL DB Configuration

1. In the **Source Database** dialog, select **MySQL 5.1 or higher** from the **Source Database** list.
2. In the **Extract Type** list, select **Log**.
3. In the **Database** tab, enter the following configuration details:
    a. **Host Name**: Enter the IP address or hostname of the database server.
    b. **Port**: Enter the port number on which the database server is listening (default port for MySQL is 3306).
    c. **Database Name**: Enter the name of your database.
    d. **User Id**: Enter a valid MySQL user ID to be used with Ingest.
    e. **Password**: Enter the password and confirm it by re-entering it in the **Confirm Password** field.
    Note: Passwords are encrypted within Ingest.
    f. **JDBC Options** (Optional): JDBC options are optional and can be used to extend the JDBC URL for accessing the database.
4. Select the **Advanced Options** tab. Enable **Handle zero length strings**. Refer to Appendix: Additional Configurations for more information.
5. Select **Test** to verify the connectivity.
6. Select **Apply** to confirm and save the details.

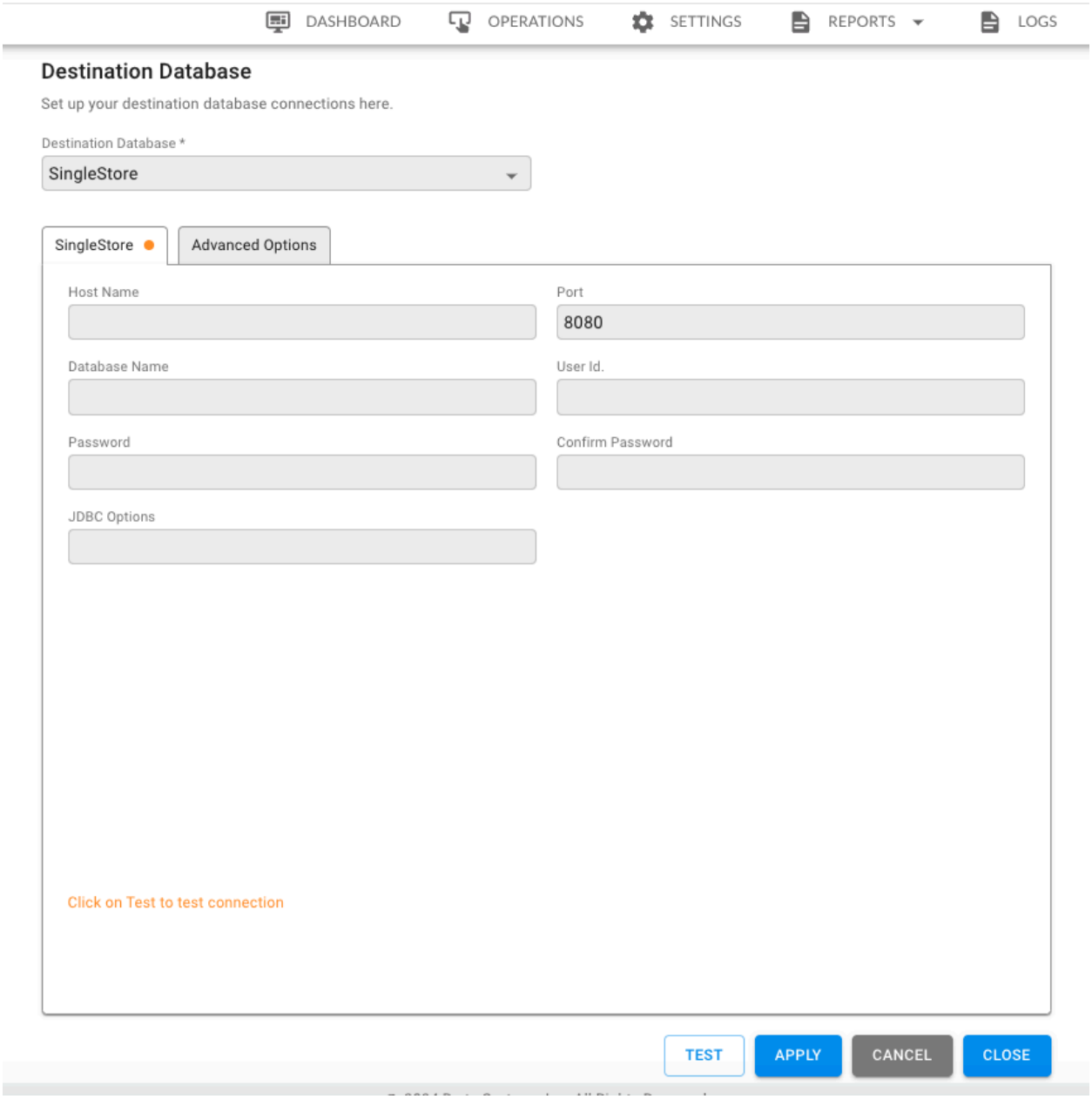
PostgreSQL DB Configuration

1. In the **Source Database** dialog, select **Postgres** from the **Source Database** list.
2. In the **Extract Type** list, select **Logs**.
3. In the **Database** tab, enter the following configuration details:
    a. **Host Name**: Enter the IP address or hostname of the database server.
    b. **Port**: Enter the port number on which the database server is listening (default port for Postgres is 5432).
    c. **Database Name**: Enter the name of your database.
    d. **User Id**: Enter a valid Postgres user ID to be used with Ingest.
    e. **Password**: Enter the password and confirm it by re-entering it in the **Confirm Password** field.
    Note: Passwords are encrypted within Ingest.
    f. **JDBC Options** (Optional): JDBC options are optional and can be used to extend the JDBC URL for accessing the database.
4. Select **Test** to verify the connectivity.
5. Select **Apply** to confirm and save the details.

Destination Database

You can configure Singlestore databases as a destination. To configure the destination database, navigate to **Dashboard > Destination**, and then select the gear icon.

1. In the **Destination Database** dialog, select **SingleStore** from the **Destination Database** list.
2. In the **SingleStore** tab, enter the following configuration details:
    a. **Host Name**: Enter the IP address or hostname of the database server.
    b. **Port**: Enter the port number on which the database server is listening (default port for SingleStore is 8080).
    c. **Database Name**: Enter the name of your database.
    d. **User Id**: Enter a valid SingleStore user ID to be used with Ingest.
    e. **Password**: Enter the password and confirm it by re-entering it in the **Confirm Password** field.
    Note: Passwords are encrypted within Ingest.
    f. **JDBC Options** (Optional): JDBC options are optional and can be used to extend the JDBC URL for accessing the database.



3. Select the **Advanced Options** tab. Enter the following configuration details:
    a. Max Updates (Default maximum update is 4)
    b. Load Threads
    c. Add Database Prefix
    d. Truncate table instead of drop
    e. Schema for all tables
    f. Ignore database name in schema
    g. Schema for staging tables
    h. Retain staging tables

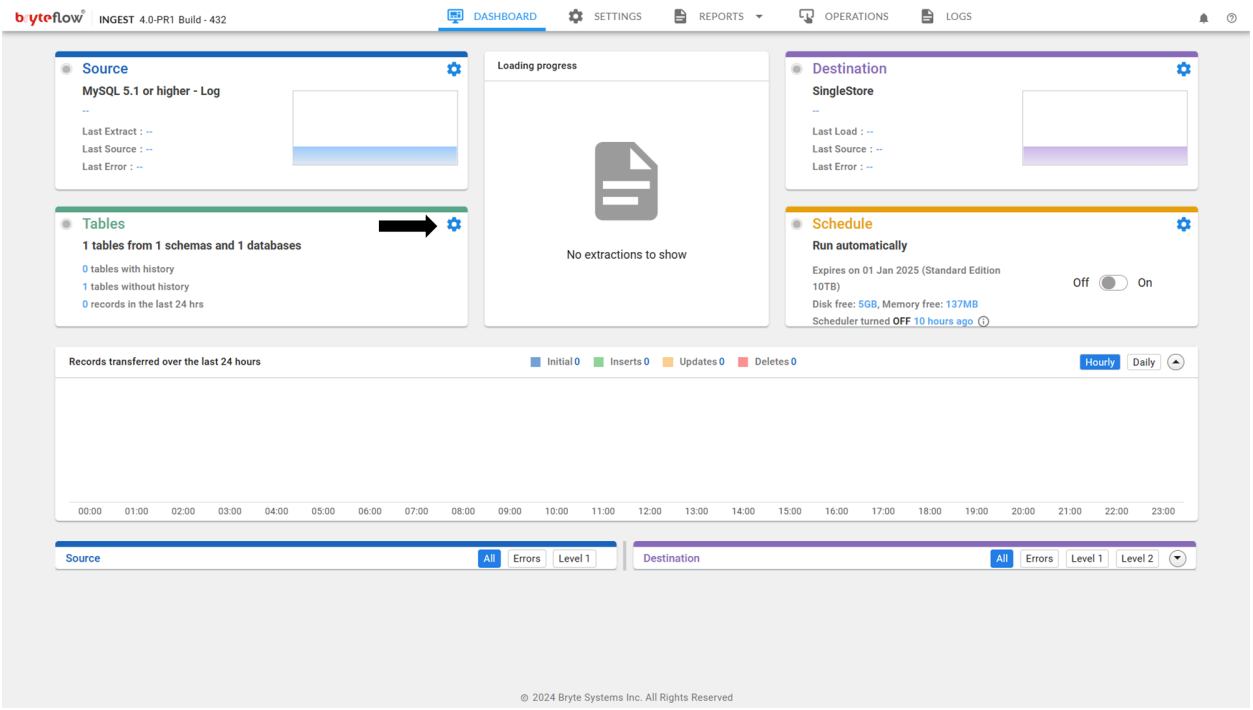
    Refer to Appendix: Additional Configurations for more information.
4. Select **Test** to verify the connectivity.

5. Select **Apply** to confirm and save the details.

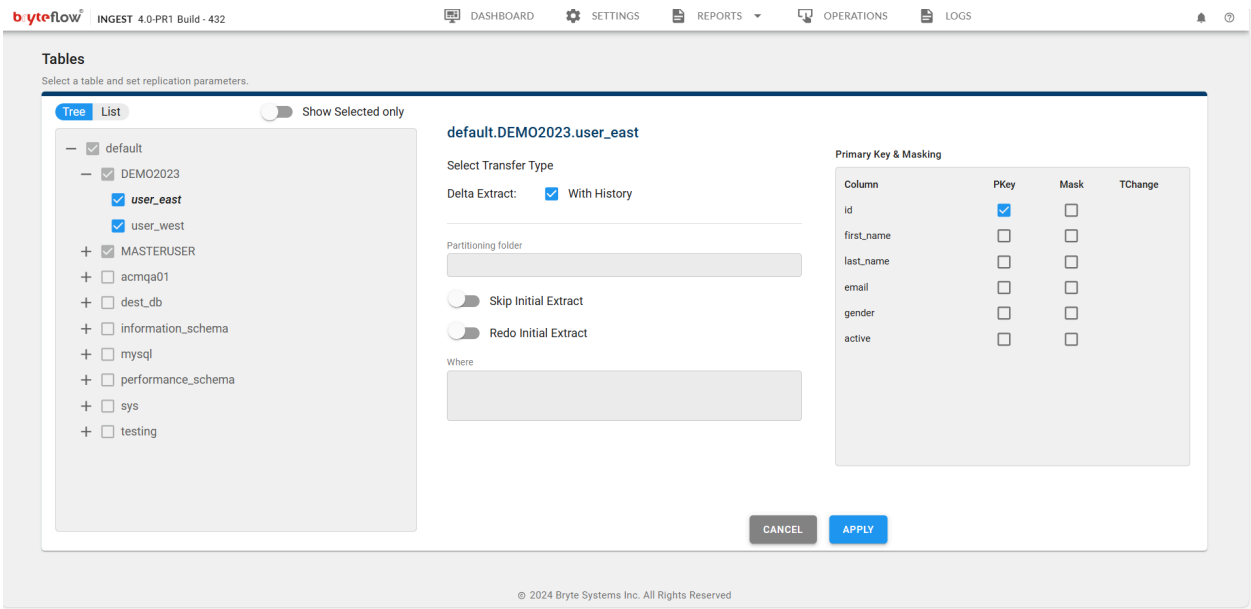
Tables

Note: Review this section in conjunction with the Appendix: Understanding the Extraction Process.

You can configure tables to ingest data from source database to destination database. To select a table for transferring to SingleStore database, navigate to **Dashboard > Tables**, and then select the gear icon.



1. The **Table** page contains two views:
    a. Tree view
    b. List view.



2. In the **List** view, expand **default** to view the schemas under the connected database.
3. Expand the schema to view the tables.
4. Select the box next to the table to extract using Ingest.
5. You can view the details for the selected table in the middle column in the page.
6. Select the **With History** checkbox to create a history of records in the destination table. Leave the checkbox unselected if you want a mirror copy of the table instead.
7. **Partitioning** is not applicable to SingleStore destination.
8. Enable **Skip Initial Extract** to capture changes to a table without performing the initial bulk load.
9. Enable **Redo Initial Extract** to perform the initial bulk load of a previously extracted table.
10. You can add a simple **`WHERE`** clause to filter records during the initial bulk load. This option does not apply to delta extracts.
11. Select the primary key column or unique indexed column by enabling **PKey** under **Primary Key & Masking**. If your table does not have a primary key or a unique indexed column, select multiple columns to form a natural key.
12. To prevent the extraction of values from a specific column, you can mask it by enabling **Mask** under **Primary Key & Masking**. Ingest does not extract values for a masked column, and the column is created as `varchar(1)` in the SingleStore table.
13. If necessary, select the **TChange** checkbox next to columns that require a datatype conversion.
14. Select **Apply** to confirm and save the details.

Repeat this process for each table. Once completed, proceed with the next steps:

1. Navigate to the **Operations** tab.
2. Select **Full Extract** to initiate the initial extract and load process.

Column Type Change

This feature is commonly used in SAP environments to allow datatype changes for columns or fields, such as converting from character or numeric formats to Integer, Long, Float, Date, or Timestamp.

Ingest automatically converts data types during data replication or CDC to the appropriate destination formats. The destination data types are:

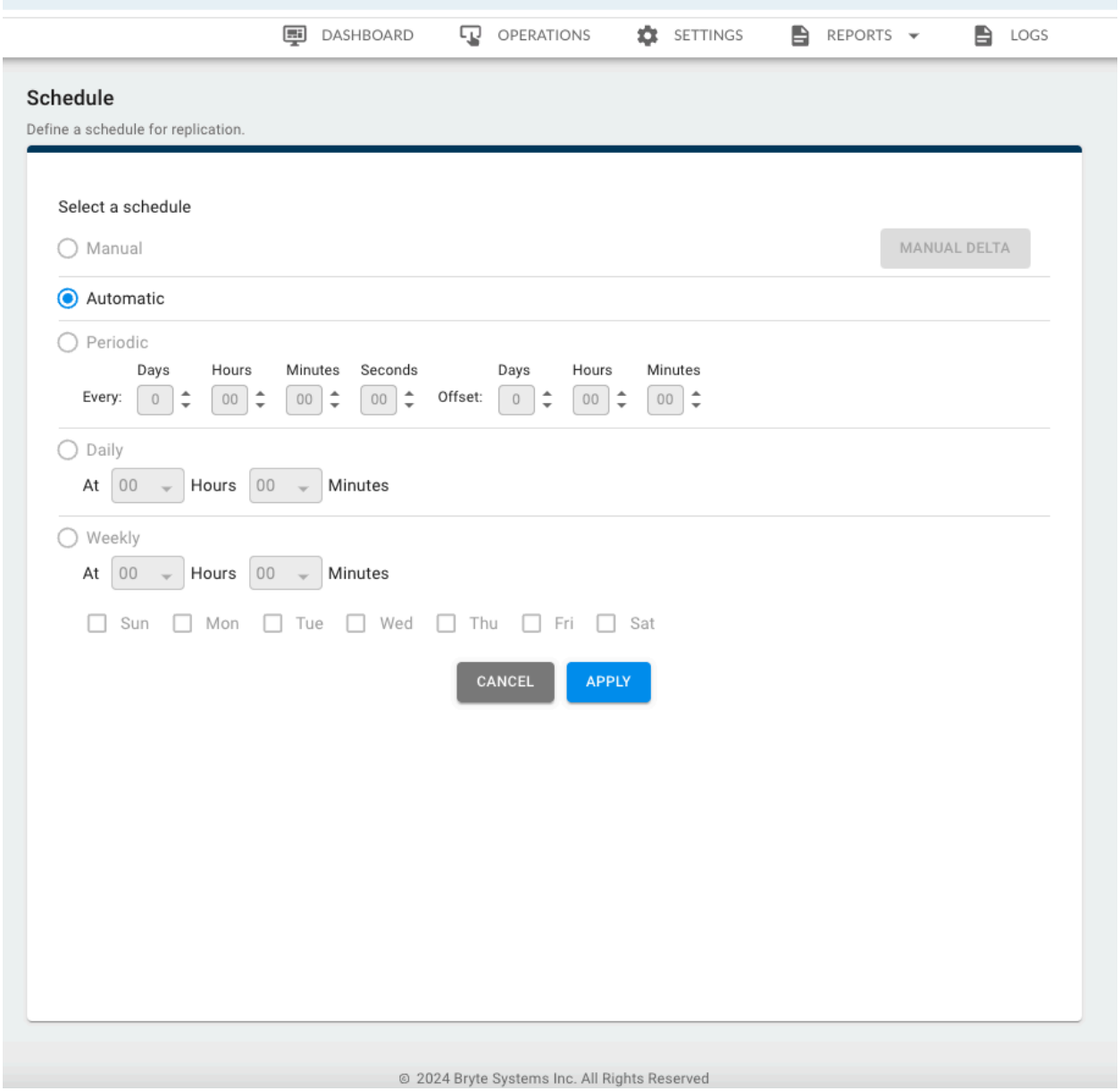
`INTEGER`	`@I`
`LONG`	`@L`
`FLOAT`	`@F`
`DATE` (including format clause e.g. `yyyyMMdd`)	`@D(format)`
`TIMESTAMP` (including format clause e.g. `yyyy-MM-dd HH:mm:ss`)	`@T(format)`

Note: The `(format)` part can vary based on the value in the source column.

Schedule an Ingest Job

You can schedule an Ingest job after setting up the tables. To schedule an Ingest job, enable the scheduler, navigate to **Dashboard > Schedule**, and then select the gear icon.

1. Create a schedule as per your requirement.



2. Select **Apply** to save the changes.

Add a New Table to Existing Extracts

If replication is running and you need to add a new table to the extraction process, perform the following:

1. Disable the scheduler (top right of the screen under the **Schedule** tab).
2. Navigate to **Dashboard > Tables**.
    a. Select the new table(s) by browsing to the database instance name, schema name, and table name(s).
    b. Configure the table with the following options:

        i. Transfer type
        ii. Partitioning folder (refer to the Partitioning section for details)
        iii. Primary key column(s)
        iv. Columns to be masked (optional; masked columns are excluded from replication, e.g., salary data)
    c. Select **Apply**.
    d. Repeat the process for each additional table.
3. Navigate to the **Operations** tab.
    a. Select **Sync New Tables**.

This initiates the full extract for the new table(s). Once completed, Ingest automatically resumes processing deltas for both the new and previously configured tables.

Resync Data for Existing Tables

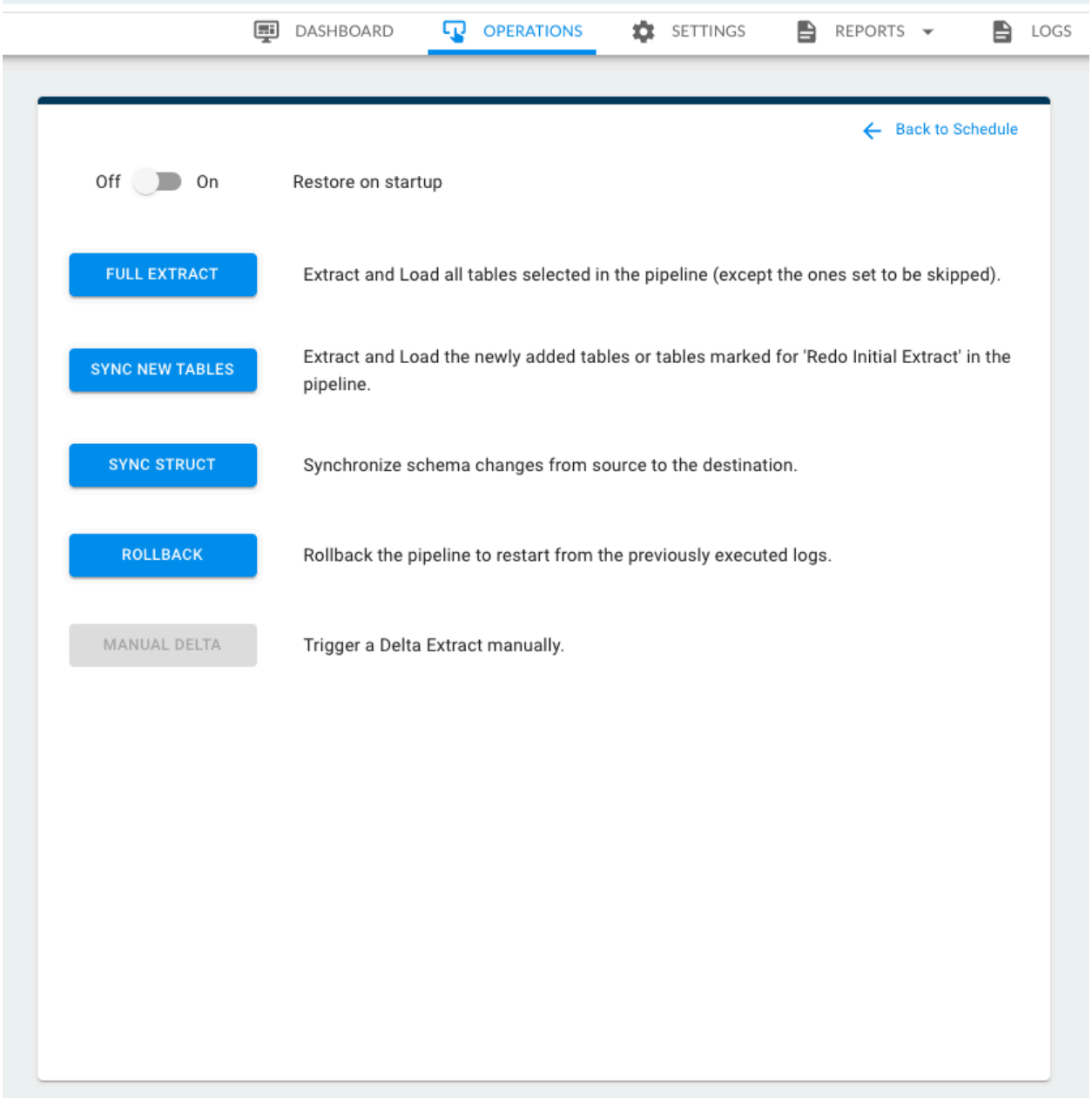
To resync data from the source, follow these steps depending on your requirements:

For **Primary Key with History**:

1. Disable the scheduler (top right of the screen under the **Schedule** tab).
2. For resyncing data for all configured tables:
    a. Navigate to the **Operations** tab.
    b. Select **Full Extract**.
3. For resyncing data for selected tables:
    1. Navigate to **Dashboard > Tables**.
        a. Select the table(s) by browsing to the database instance name, schema name, and table name(s).
        b. Select **Redo Initial Extract**.
        c. Repeat the process for each table that requires resyncing.
    2. Navigate to the **Operations** tab.
        a. Select **Sync New Tables** to resume processing deltas.

Operations

Operations allows to manage and troubleshoot the data pipeline, control the synchronization process, and handle the schema changes. To manage the data ingestion pipeline, navigate to the **Operations** tab.



Select the following according to your requirements.

1. **Restore on Startup (On/Off toggle):** Enable or disable to determine whether the pipeline must automatically restore its previous state when the system starts.
2. **Full Extract**: Trigger the initial bulk load for all the selected tables, except those marked as **Skip Initial Extract**. Every Ingest pipeline must undergo a full extract at least once to begin with.
3. **Sync New Tables**: Trigger the initial bulk load for tables marked as **Redo Initial Extract** and newly added tables in an ongoing replication. This operation cannot replace a full extract.

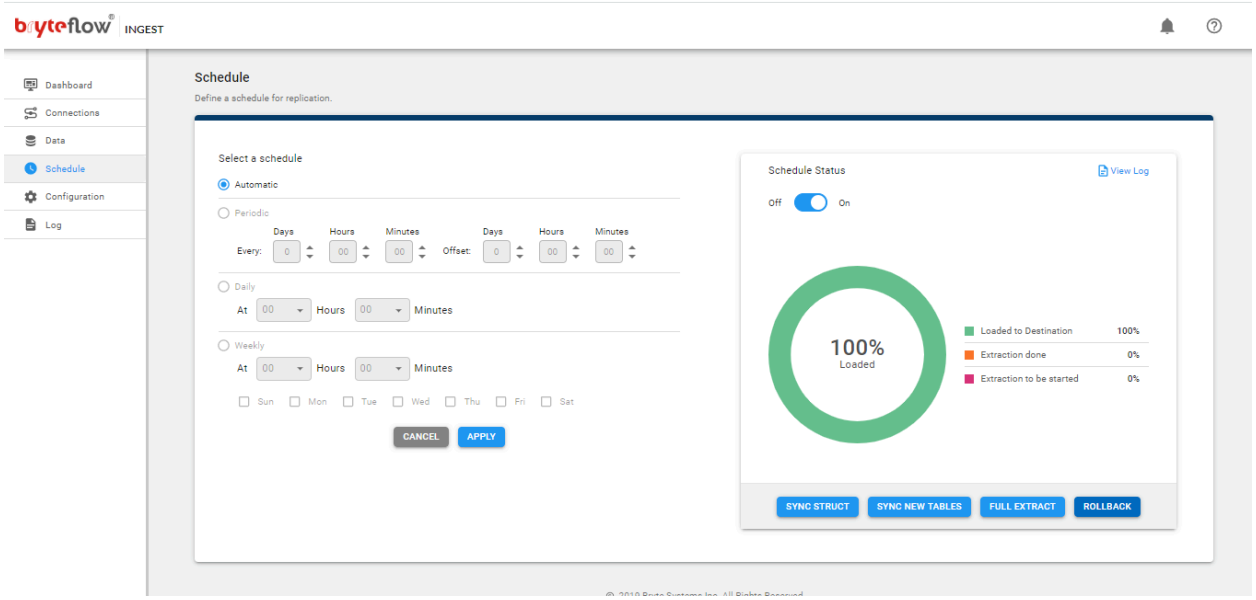
4. **Sync Struct**: Compare and fix the table structure in the destination database. Load fails if the schema changes. The operator identifies when the load started failing and rolls back to a point just before the initial failure. During this operation, Ingest compares and updates the structure of the destination table without losing data. If a significant amount of time has passed or the log file is unavailable, triggers a **Redo Initial Extract** to sync the table.
5. **Rollback**: Replay logs from a specific point to reapply changes to the destination. If the primary key columns are correctly selected, there should be no duplicates in the destination table. Rollback affects all tables in the selected list.
6. **Manual Delta** : Triggers a manual delta extract to process incremental changes.

Rollback

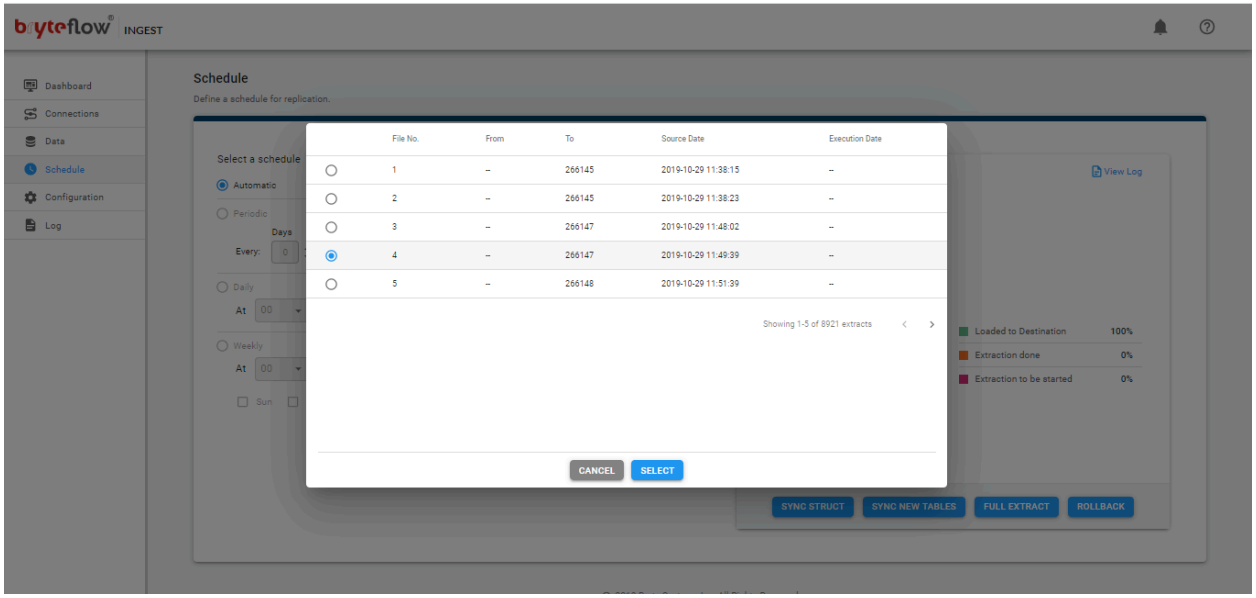
In the event of unexpected issues, such as intermittent source database outages or network connectivity problems, you can rollback Ingest to a point in time before the issue occurred and replay the changes.

To perform the rollback:

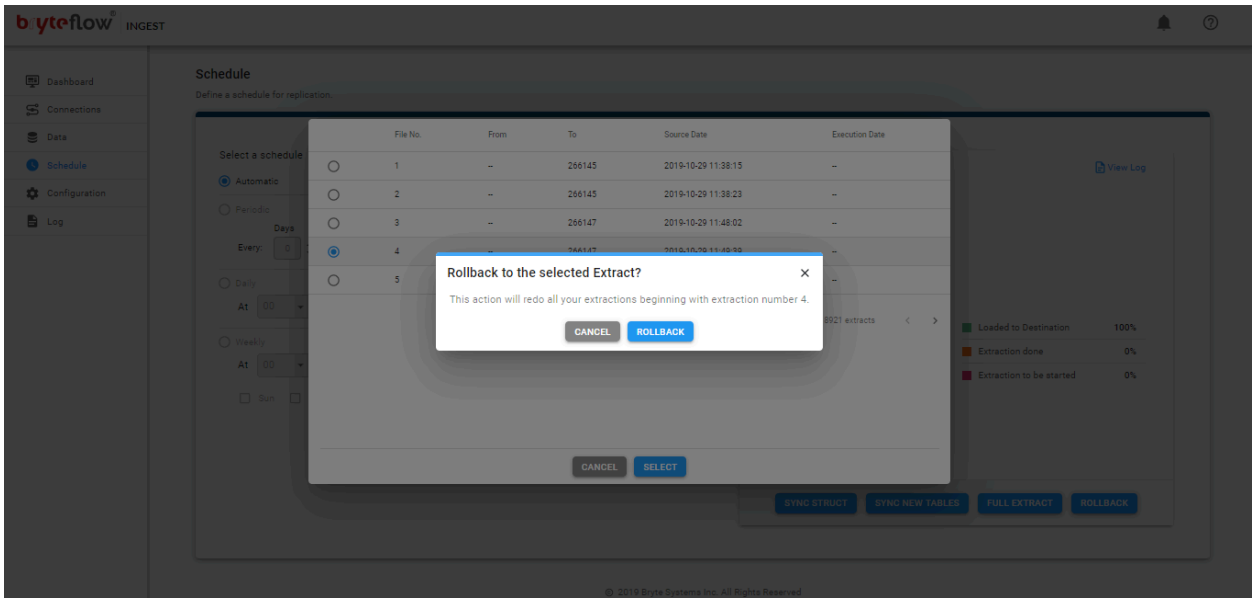
1. Navigate to the **Schedule** tab.



2. Select **Rollback**.
3. The rollback screen appears, displaying a list of available rollback points in descending order, depending on the source database log retention policy.
4. Select the desired date (radio button) and **Select**.



5. Select **Rollback** to initiate the rollback.
6. Ingest automatically catches up from the selected point in time to the current time that replays all log entries and applies them to the destination.



Settings

Instance Details

Navigate to **Settings > Instance Details**, and enter the following configuration details:

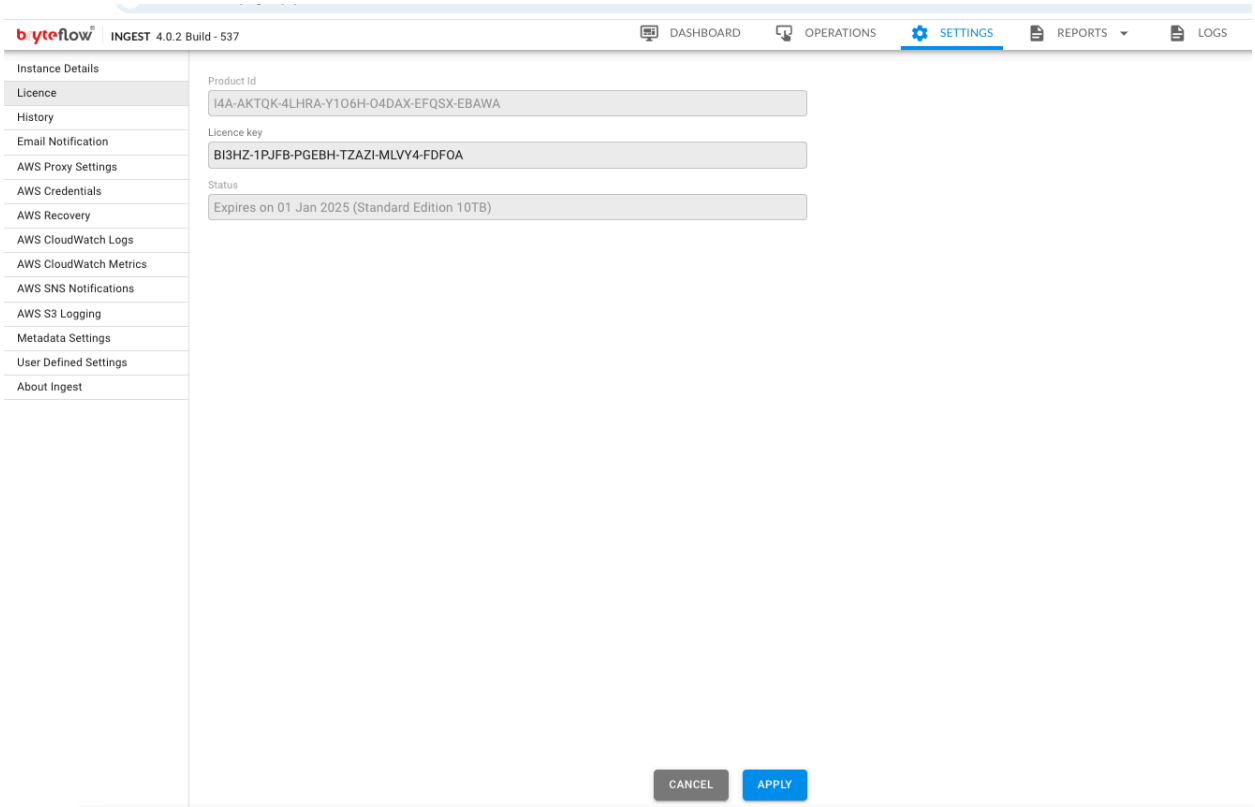
1. **Ingest Instance Name**: Enter a name for the current instance of Ingest.

2. **Web Port**: Enter the port on which the Ingest server runs.
3. **Icon Text**: Enter the text for your instance icon. The text takes only 2 characters.
4. **Icon Color**: Select the color for your instance icon.

Select **Apply** to save the settings.

License

To obtain a valid license, contact SingleStore Sales or your SingleStore technical account team representative with the **Product ID**. Navigate to **Settings > Licence**. In **Licence Key**, enter the license key, and then select **Apply** to save the settings.



History

Navigate to **Settings > History**, and enter the following configuration details:

1. Enable **Keep history by default**, if you want to keep history always.
2. In **History Start Date Column**, enter the source date for type-2 SCD records.
3. In **History End Date Column**, enter the history end date for type-2 SCD records.
4. In **Open End Date Column (YYYY-MM-DD)**, enter the end date used for history records.
5. In **Max. Updates in batch**, enter the value that determines when combined updates exceed this threshold in a batch.

Select **Apply** to save the settings.

Email Notification

To configure email updates, navigate to **Settings > Email Notification**, and then perform the following steps:

1. To receive email notifications, enable **Enable Email Notifications**.
2. In **Host Name**, enter the address of your SMTP server.
3. In **Port**, enter the port number that the SMTP server is listening on.
4. In **Sender**, enter the email address from which the notifications are sent. This must be a valid email address on the server.
5. In **Recipient**, enter the email address to which the notifications are sent.
6. In **User ID**, enter the full email address to authenticate with the SMTP server.
7. Enter the **Password** for the email, then confirm it in the **Confirm Password** field.

   Note: Passwords are encrypted within Ingest.

8. Select **Apply** to save the settings.

AWS Proxy Settings

Navigate to **Settings > AWS Proxy Settings**, enter the following configuration details:

1. **Proxy Host**: Enter the S3 proxy host name.
2. **Proxy Port**: Enter the S3 proxy port.
3. **Proxy User Id**: Enter the S3 proxy user ID.
4. **Proxy Password**: Enter the S3 proxy password.

Select **Apply** to save the settings.

AWS Credentials

Ingest can access AWS services using either IAM roles or access keys when deployed on-premises. You must configure the access method and credentials in Ingest.

To access AWS services from an on-premises SingleStore Flow server, navigate to **Settings > AWS Credentials**, and perform the following steps:

1. Enable **Use EC2 IAM Role**, if you are using.
2. Select the **AWS Region** from the list.
3. Enter the AWS access key ID in **AWS Key** and **AWS Secret Key** to access the S3 service (if installed on-premises). For installations using IAM roles, these keys are not required. **Note**: Keys are encrypted within Ingest.
4. If you are using KMS, enter the KMS Key in **AWS Key**. **Note**: Keys are encrypted within Ingest.
5. Select **Apply** to confirm and save the details.

AWS Recovery

Ingest supports high availability, which means it automatically saves the current configuration and execution state to S3. This enables recovery of the Ingest instance (including its current state) if it is catastrophically lost.

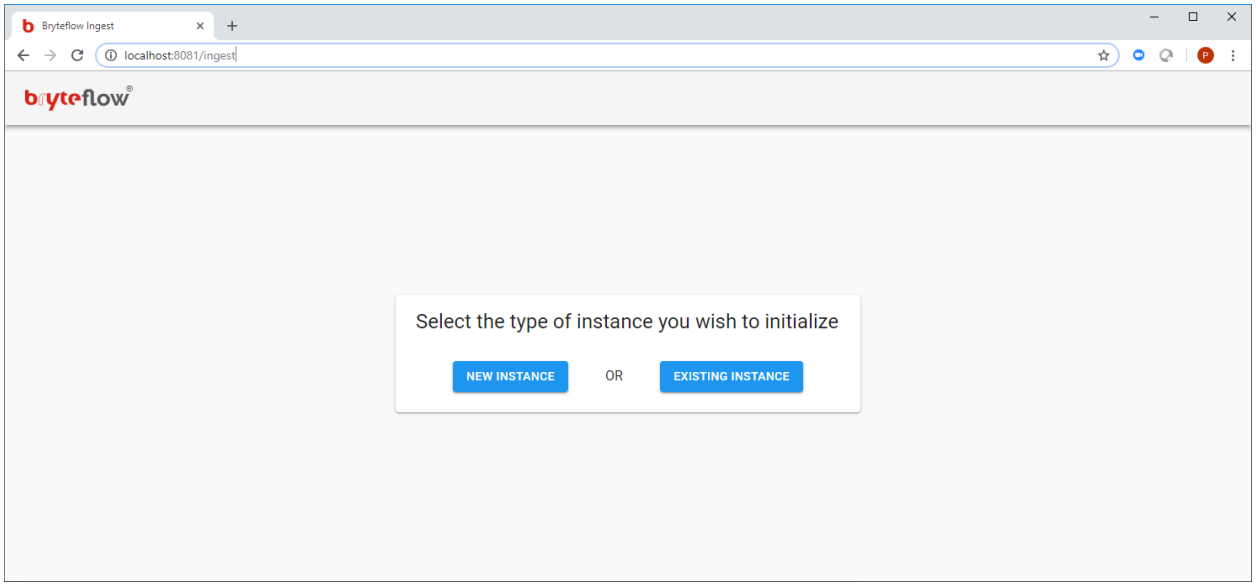
SingleStore Flow keeps a backup of every successful job execution in S3 that makes the latest version available for recovery. To configure it, navigate to **Settings > AWS Recovery**, and perform the following steps:

1. In **Ingest Instance Name**, enter a name for the current instance of Ingest.
2. Check **Enable Recovery**.
3. In **S3 Recovery Location**, enter the destination for recovery data in S3, for example: `s3://your_bucket_name/your_folder_name/Your_Ingest_name`.
4. In **S3 KMS Id**, enter the KMS ID to encrypt logs in S3. This is optional but recommended. When provided, logs are encrypted using KMS encryption.
5. Select **Apply** to save the settings.

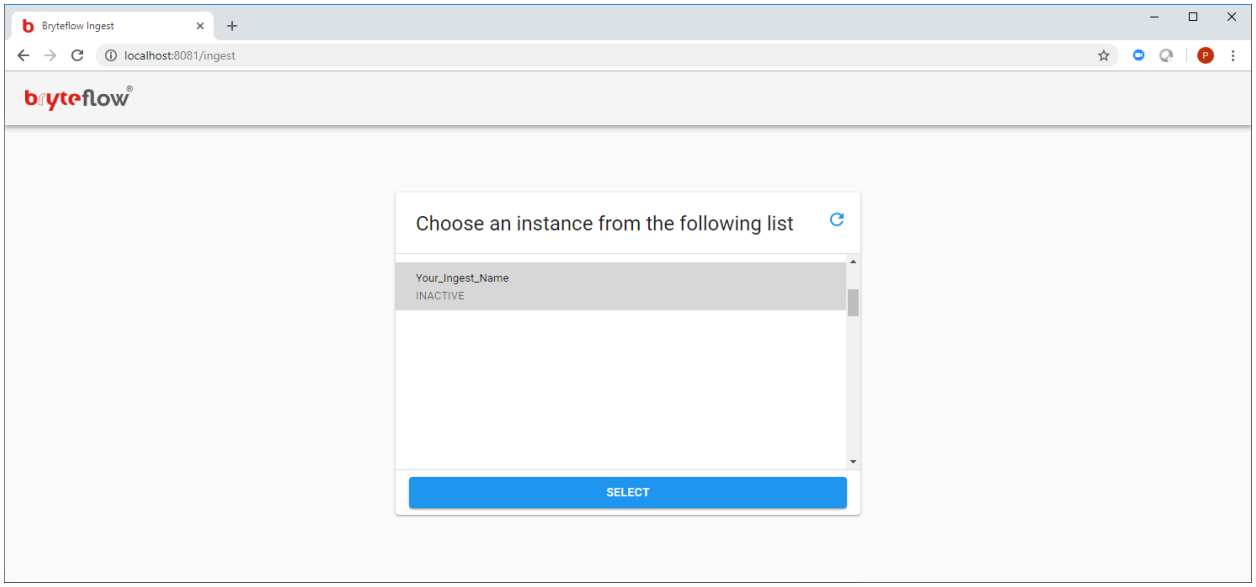
Recovery Utilization

To recover an instance of Ingest, source a new instance of Ingest on a new server or Amazon EC2 server:

1. Download and install Ingest in the newly launched EC2. Refer to the SingleStore Flow installation guide.
2. Once the EC2 instance is running, the endpoint is an EC2 instance hosting Ingest as a Windows service on port `8081`. The EC2 instance's assigned role must include the required policies. Refer to the AWS Identity and Access Management (IAM) for SingleStore Flow.
    a. The list of instances is stored in DynamoDB. Ensure that the EC2 role has the necessary permissions to access DynamoDB and the underlying table containing saved instances.
3. Once the EC2 instance is launched, open the Ingest web console by entering `localhost:8081` in a Chrome browser.
4. Select **Existing Instance**.



5. Select the instance you want to restore from the list (e.g., "Your_Ingest_Name").
6. Select **Select**.



7. Ingest retrieves the configuration and saved execution state of the selected instance and restores it.
8. After restoration, SingleStore recommends to stop the faulty EC2 instance (the previous installation):
    a. Go to the AWS EC2 console.
    b. Search for the Ingest tag (or the tag used to launch the application).
    c. Select the older EC2 instance.
    d. Navigate to **Actions > Instance State**, and then select **Terminate**.

Note: Recovery can also be used as a method for partial migration between environments (e.g., from `DEV` to `PROD` stacks). Since the restore clones the source environment and state, further configuration may be needed (e.g., updating configuration options for the `PROD` stack S3 location, etc.). This method can reduce the workload when migrating hundreds of tables to a new EC2 instance.

Recovery from Faults

SingleStore Flow supports high availability and auto-recovery mechanisms in case of faults or failures. These include:

- **EC2 Instance Failures**: If the EC2 instance is terminated or affected, SingleStore Flow saves the last successful loads as a savepoint. It resumes from the savepoint when restarted on another EC2 instance or AZ.
- **SingleStore Connection Issues**: SingleStore Flow retries until successful.
- **Source DB Connection Issues**: SingleStore Flow retries until successful.
- **AWS S3 Connection Issues**: SingleStore Flow retries until successful.

Customers seeking high availability must configure their Ingest instance for high availability and recovery. Refer to AWS Recovery for setup details.

In the event of application faults or failures:

- Enable CloudWatch Logs and metrics to get notified. AWS CloudWatch events can be used for alerting by writing custom Lambda functions.
- Enable SNS notifications in setup details Ingest and subscribe to the SNS topic via the AWS console.

To monitor EC2 instances:

- Enable CloudWatch monitoring for EC2 or set up health checks with proper alerts.
- For disk space, setup details Ingest sends status updates to CloudWatch Logs, including free disk space in GB. Users can create Lambda functions to raise alarms when space is low.

Refer to Remote Monitoring for more information.

Disk Space Monitoring via Lambda

Prerequisites:

1. Create an IAM role with the following policy to allow Lambda functions to call AWS services:

```
JavaScript
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1",
      "Action": [
        "logs:FilterLogEvents",
        "logs:GetLogEvents",
        "logs:PutLogEvents"
      ],
      "Effect": "Allow",
      "Resource":
"arn:aws:logs:<region>:<account_ID>:log-group:<log-group-name>:log-stream:<log-
stream-name>"
    },
    {
      "Sid": "Stmt2",
      "Action": [
        "sns:Publish",
        "sns:TagResource"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:sns:<region>:<account_ID>:<topic_name>"
    }
  ]
}
```

2. Create an SNS topic (refer to What is Amazon CloudWatch? for more information) and use the ARN in your Lambda function code.

Lambda Function Steps:

1. Log in to the AWS console and go to the AWS Lambda dashboard.
2. Select **Create Function** and enter a function name.
3. Select Python 3.8 as the runtime.
4. Select **Create Function**.
5. Add a trigger for **CloudWatch Log** and select your log group.
6. Add the Lambda code for disk check and select the **Lambda Execution Role**.
7. Select **Save**.

Here is sample Lambda code to check disk space:

```python
import json
import boto3

def lambda_handler(event, context):
    freeGb = 100
    cloudwatch = boto3.client('logs')
    response = cloudwatch.get_log_events(
        logGroupName='Oracle_LogGroup',
        logStreamName='Oracle_LogStream',
        startFromHead=False,
        limit=100
    )
    for i in response['events']:
        msg = json.loads(i['message'])
        if msg['type'] == "Status" and msg['diskFreeGB'] < freeGb:
            sns = boto3.client('sns')
            sns.publish(
                TopicArn='arn:aws:sns:us-west-2:689564010160:LambdaTrigger',
                Message='Your free disk size is getting low, please contact the
concerned team!'
            )
```

Note: Customize the email body as needed.

Time to Recover

Recovery Point Objective (RPO)

Ingest automatically recovers from most failure scenarios that leverages durable services like S3 for unlimited data retention. Data recovery depends on the customer's source DB retention settings.

Note: No full reload is required, unlike other solutions.

Recovery Testing

After recovery, perform the following tasks before resuming replication:

1. Start the Ingest service.
2. Open the web console in Chrome.
3. Navigate to **Dashboard > Source** and test the connection for the source database.
4. Navigate to **Dashboard > Destination** and test the connection for the destination database.
5. If there are any connection issues, troubleshoot and resolve them.

6. Once successful, navigate to Dashboard, under **Schedule**, enable the Scheduler, and resume replication.

AWS CloudWatch Logs

Ingest comes pre-configured with remote monitoring capabilities that leverage existing AWS technologies, such as CloudWatch Logs and Events. CloudWatch, along with other AWS services, can monitor the execution of Ingest and trigger appropriate alarms in case of errors or failures. The events from Ingest application flows to CloudWatch Logs. These events provide detailed application statistics that can be used for custom monitoring.

In addition to integration with CloudWatch, Ingest writes internal logs directly to S3, including execution and error logs from the Ingest console. Navigate to **Settings > AWS CloudWatch Logs**, and perform the following steps:

1. Check **Enable CloudWatch Logs** and metrics to get notified. AWS CloudWatch events can be used for alerting by writing custom Lambda functions.
2. Enter the name of the **CloudWatch Log Group.** This needs to be created first in the AWS console.To create a CloudWatch Log Group, refer to CreateLogGroup for more information.
3. Enter the name of the **CloudWatch Log Stream**. This also needs to be created first in the AWS console. To create a CloudWatch Log Stream, refer to CreateLogStream for more information.
4. Select **Apply** to save the settings.

Refer to Appendix: SingleStore Flow Events for AWS CloudWatch Logs and SNS for more information.

AWS CloudWatch Metrics

Navigate to Settings > AWS CloudWatch Metrics, and perform the following steps:

1. Check **Enable CloudWatch Metrics** if required.
2. In **CloudWatch Metrics Namespace,** enter the name of CloudWatch metric. Refer to View available metrics for more information.
3. Select **Apply** to save the settings.

The events that Ingest pushes to AWS CloudWatch Logs and the metrics console are as follows. Refer to Appendix: Understanding the Extraction Process and Appendix: SingleStore Flow Events for AWS CloudWatch Logs and SNS for more information.

AWS SNS Notifications

Navigate to **Settings > AWS SNS Notifications**, and perform the following steps:

1. Check **Enable SNS notifications** and subscribe to the SNS topic via the AWS console.

2. In **SNS Topic**, enter the Topic ARN or SNS Topic Name. To create an SNS Topic, refer to Creating an Amazon SNS Topic for more information.
3. Select **Apply** to save the settings.

Refer to Appendix: SingleStore Flow Events for AWS CloudWatch Logs and SNS for more information.

AWS S3 Logging

Navigate to **Settings > AWS S3 Logging**, and perform the following steps:

1. Check **Enable S3 Logging** if you want to record data to S3 (console/execution logs).
2. In **S3 Log Location**, enter the destination of the logging data in S3, for example: `s3://your_bucket_name/your_folder_name`.
3. In **S3 KMS Id**, enter the KMS ID to encrypt logs in S3. This is optional but recommended. When provided, logs are encrypted using KMS encryption.
4. Select **Apply** to save the settings.

Metadata Settings

Navigate to Settings > Metadata Settings, and perform the following steps:

1. Check **Enable Operational Metadata**, if required.
2. Select metadata database from the following **Database Type** list:
    a. AWS Aurora
3. In **Host Name**, enter the IP address or hostname of the database server.
4. In **Port**, enter the port number on which the database server is listening.
5. In **Database Name**, enter the name of your database.
6. In **Database Schema**, enter the schema name of your database.
7. In **User Id**, enter a valid database user ID to be used with Ingest.
8. Enter the **Password** and confirm it by re-entering it in the **Confirm Password** field.
9. **JDBC Options** are optional and can be used to extend the JDBC URL for accessing the database.
10. Select **Apply** to save the settings.

User-Defined Settings

To add custom fields, navigate to **Settings > User Defined Settings**. Enter the **Name** and **Value** of your custom field, and select the **Save** icon. You can also **Delete** and **Undo** the name and value of your custom field.
Note: You can add only one custom field.

About Ingest

To view the **Version**, **Build**, **Host Name**, and **Ingest Location** of your installed Ingest Instance, navigate to **Settings > About Ingest**.

Reports

To run the reports and view the performance of your tables month wise and day wise, navigate to the **Reports** tab.

Run Reports for Month

To run the reports of ingestion for a particular month, from the **Reports** list, select **Run Reports for Month**. From the **Tables** list, select the table and from the **Months** list, select the month. You can see the following in the report of the selected month:

1. Date
2. No. of Runs
3. Records
4. Extract Duration
5. Load Duration

You can modify the report by selecting **Modify** and toggle off the metric that you do not want in your report. By default, all metrics are toggled on.

Run Reports for Day

To run the reports of ingestion for a particular date, from the **Reports** list, select **Run Reports for Day**. From the **Tables** list, select the table and from the **Date** column, select the date. You can see the following in the report of the selected date:

1. Run Id
2. File Id
3. Extract Start
4. Extract End
5. Extract Duration
6. Load Start
7. Load End
8. Load Duration
9. Total Tables
10. Loaded Tables
11. Error Tables
12. Initial
13. Inserts
14. Updates
15. Deletes
16. Status

You can modify the report by selecting **Modify** and toggle off the metric that you do not want in your report. By default, all metrics are toggled on.

Performance for Month

To view the performance of the ingestion for a particular month, from the **Reports** list, select **Performance for Month**. From the **Months** list, select the month. You can see the following for the selected month:

1. A graph illustrating
    a. Total
    b. DB Delay is represented by Orange.
    c. Extract is represented by Blue.
    d. Load Wait is represented by Yellow.
    e. Load is represented by Green.
2. Date
3. No. of Runs
4. Total Duration
5. DB Delay Duration
6. Extract Duration
7. Load Wait Duration
8. Load Duration

You can modify the performance view by selecting **Modify** and toggle off the metric that you do not want in your performance view. By default, all metrics are toggled on.

Performance for Day

To view the performance of the ingestion for a particular date, from the **Reports** list, select **Performance for Day**. From the **Date** column, select the date. You can see the following for the selected date:

1. A graph illustrating
    a. Total
    b. DB Delay is represented by Orange.
    c. Extract is represented by Blue.
    d. Load Wait is represented by Yellow.
    e. Load is represented by Green.
2. Run Id
3. Load Finish
4. Total Duration
5. DB Delay Duration
6. Extract Duration
7. Load Wait Duration
8. Load Duration

You can modify the performance view by selecting **Modify** and toggle off the metric that you do not want in your performance view. By default, all metrics are toggled on.

Logs

To monitor the progress of your extracts and loads, navigate to the **Logs** tab. The log displays the progress and current activity of the Ingest instance. You can apply filters to view specific logs, such as errors. You can view the logs of the following:

- **Source**: Displays the logs of the source database.
- **Destination**: Displays the logs of the destination database.
- **Others**: Displays the logs of the Ingest instance.
- **Errors**: Displays all errors related to the source, destination, and Ingest instance.

Ingest stores log files in the install folder, under the `\log` folder. The path to the log files is: `<install folder of Ingest>\log\sirus*.log`. For example, `c:\SingleStore-Flow\Ingest\log\sirus-2019-01.log`.

Error files are also stored in the `\log` folder. The path to these error files is: `<install folder of Ingest>\log\error*.log`. For example, `c:\SingleStore-Flow\Ingest\log\error-2019-01.log`.

These logs can also be reviewed or stored in S3. Refer to Remote Monitoring for more information.

Optimize Usage of AWS Resources / Save Costs

Tagging AWS Resources

AWS allows customers to assign metadata to their AWS resources through tags. SingleStore highly recommends tagging all AWS resources created for and by SingleStore Flow to manage and organize resources, control access, track costs, and automate processes.

For clarity and organization, SingleStore recommends to use tags with names specific to the instances being created. For example, for a SingleStore Flow instance replicating a production database server for Billing and Finance, the tag name must reflect the database name it is dedicated to, such as `SingleStoreIngest_BFS_EC2_Prod`. Similarly, for a user acceptance testing (UAT) environment, the tag name can be `SingleStoreIngest_BFS_EC2_UAT`. This approach helps customers differentiate between various AWS resources in their environment. Use consistent tag names across all services.

SingleStore recommends tagging AWS EC2 services with unique and identifiable tag names. Refer to Tag your Amazon EC2 resources for more information on tagging AWS EC2 resources.

Appendix: Understanding the Extraction Process

Extraction Process

The extraction process consists of two parts:

- Initial Extract
- Delta Extract

Initial Extract

An initial extract is performed the first time Ingest connects to a database. During this extract, the entire table is replicated from the source database to the destination.

Delta Extract

After the initial extract, Ingest performs delta extracts. Delta extracts capture only the changes made since the last extraction and merge them with the destination.

A typical delta extract log file looks like this:

```
Extracting 2

Delta Extract database_name:table_name

Info (ME188): Stage pre-BCP

Info (ME190): Stage post-BCP

Info (ME260): Stage post-process

Delta Extract database_name complete (10 records)

Extracted 2

Load file 2

Creating table dbname_schemaname.table_name...

Created table dbname_schemaname.table_name

Loading table dbname_schemaname.table_name with x records(n bytes)

Created new connection org.mariadb.jdbc.Connection@4ca52dc7

Replace data...
```

```
Loading ./spool/dbname_schemaname.table_name_2.dat into
dbname_schemaname.table_name

Loaded ./spool/dbname_schemaname.table_name_2.dat

Deleted ./spool/dbname_schemaname.table_name_2.dat

Replace data completed

Loaded table dbname_schemaname.table_name(0 of 1 left)

Loaded file 2(Source=2025-01-07 10:01:56 IST)
```

First Extract

The first extract always needs to be a **Full Extract**. This extracts the entire table, and future extractions are delta extracts that run periodically based on your desired frequency.

Appendix: Additional Configurations

Source Database

While configuring the source database, there are additional configurations for each **Extract Type**.

- Handle zero length strings: Load zero-length strings directly from the source to the destination.
- Extract Threads: The number of extracting threads to use.
- Log file catchup count: The number of Oracle archive logs processed in one instance.
- Log catchup time (mins):
- Log catchup offset (mins):
- Log file look-ahead:
- Convert RAW to Hex: Convert raw columns to hex strings instead of treating them as `CHAR(1)`.

Destination Database

While configuring the destination database, there are additional configurations for each **Extract Type**.

- Max Updates: Combine updates that exceed this value.
- Load Threads: The number of loading threads to use.
- Add Database Prefix:
- Truncate table instead of drop:
- Schema for all tables: Ignore the source schema and place all tables in this schema on the destination.
- Ignore database name in schema: Check this option to ignore the database name as part of the schema prefix for destination tables.
- Schema for staging tables: Specify the schema name to be used for staging tables in the destination.

- Retain staging tables: Check this option to retain staging tables in the destination.

Appendix: SingleStore Flow Events for AWS CloudWatch Logs and SNS

Ingest supports connections to AWS CloudWatch Logs, CloudWatch Metrics, and SNS. These integrations enable monitoring of Ingest operations and facilitate interaction with other assets utilizing the AWS infrastructure. AWS CloudWatch Logs can capture event logs, such as load completion or failure, from Ingest. These logs can also help monitor error conditions and trigger alarms.

The following is a list of events that Ingest pushes to the AWS CloudWatch Logs console and AWS SNS:

SingleStore Flow Events	**Description**
`LogfileProcessed`	Archive log file processed (Oracle only)
`TableExtracted`	Source table extraction complete for MS SQL Server and Oracle (initial extracts only)
`ExtractCompleted`	Source extraction batch is complete
`TableLoaded`	Destination table load complete
`LoadCompleted`	All destination table loads in a batch complete
`HaltError`	Unrecoverable error occurred, disabled the Scheduler
`RetryError`	Error occurred, but process will retry

The following are the details for each of the SingleStore Flow events:

Event: LogfileProcessed

Attribute	Is Metric(Y/N)?	Description
type	N	“LogfileProcessed”
generated	N	Timestamp of message
source	N	Instance name
sourceType	N	“CDC”
fileSeq	N	File sequence
file	N	File name
dictLoadMS	Y	Time taken to load dictionary in milliseconds
CurrentDBDate	N	Current database date
CurrentServerDate	N	Current SingleStore Flow server date
parseMS	Y	Time taken to parse file in milliseconds
parseComplete	N	Timestamp when parsing is complete
sourceDate	N	Source date

Event: TableExtracted

Attribute	Is Metric(Y/N)?	Description
type	N	“TableLoaded”
subType	N	Table name
generated	N	Timestamp of message
source	N	Instance name
sourceType	N	“CDC”
tabName	N	Table name
success	N	true/false
message	N	Status message
sourceTS	N	Source date time
sourceInserts	Y	Number of Inserts in source
sourceUpdates	Y	Number of Updates in source
sourceDeletes	Y	Number of Deletes in source

Event: ExtractCompleted

Attribute	Is Metric(Y/N)?	Description
type	N	“ExtractCompleted”
generated	N	Timestamp of message
source	N	Instance name
sourceType	N	“CDC”
jobType	N	“EXTRACT”
jobSubType	N	Extract type
success	N	Y/N
message	N	Status message
runId	N	Run ID
sourceDate	N	Source date
dbDate	N	Current database date
fromSeq	N	Start file sequence
toSeq	N	End file sequence
extractId	N	Run ID for extract
tableErrors	Y	Count of table errors
tableTotals	Y	Count of total tables

Event: TableLoaded

Attribute	Is Metric(Y/N)?	Description
type	N	“TableLoaded”
subType	N	Table name
generated	N	Timestamp of message
source	N	Instance name
sourceType	N	“CDC”
tabName	N	Table name
success	N	true/false
message	N	Status message
sourceTS	N	Source date time
sourceInserts	Y	Number of Inserts in source
sourceUpdates	Y	Number of Updates in source
sourceDeletes	Y	Number of Deletes in source
destInserts	Y	Number of Inserts in destination
destUpdates	Y	Number of Updates in destination
destDeletes	Y	Number of Deletes in destination

Event: LoadCompleted

Attribute	Is Metric(Y/N)?	Description
type	N	“LoadCompleted”
generated	N	Timestamp of message
source	N	Instance name
sourceType	N	“CDC”
jobType	N	“LOAD”
jobSubType	N	Subtype of the “LOAD”
success	N	Y/N
message	N	Status message
runId	N	Run ID
sourceDate	N	Source date
dbDate	N	Current database date
fromSeq	N	Start file sequence
toSeq	N	End file sequence
extractId	N	Run ID for extract
tableErrors	Y	Count of table errors
tableTotals	Y	Count of total tables

Event: HaltError

Attribute	Is Metric (Y/N)?	Description
type	N	“HaltError”
generated	N	Timestamp of message
source	N	Instance name
sourceType	N	“CDC”
message	N	Error message
errorId	N	Short identifier

Event: RetryError

Attribute	Is Metric (Y/N) ?	Description
type	N	“RetryError”
generated	N	Timestamp of message
source	N	Instance name
sourceType	N	“CDC”
message	N	Error message
errorId	N	Short identifier