Release notes

TigerGraph Server 3.5.0 was released on Feb 8th, 2022.

§ Maintenance releases

3.5.3

TigerGraph 3.5.3 was released on May 12th, 2022 and is available on TigerGraph Cloud.

Issues fixed:

- Fixed security issues that could have allowed users to access local files on the local server file system.
- Fixed a security issue that could have allowed unprivileged users to change their access privileges or configuration settings.

Improvements:

- Enabled configurability for Executor file transfer concurrency.

3.5.2

TigerGraph 3.5.2 was released on April 25th, 2022 and is available on TigerGraph Cloud.

Issues fixed:

- Fixed an issue that could cause `Tokenbank.so` to be lost after restoring a backup in rare cases.
- Fixed an issue that could cause schema to become empty after cluster shrinking in rare cases.
- Fixed an issue that could cause the Graph Processing Engine (GPE) to crash after getting invalid responses from Zookeeper in rare cases.
- Fixed an issue that could cause the Graph Storage Engine (GSE) to crash during leader election in rare cases.
- Upgraded the dependencies of GUI server, GSQL server, Kafka, Kafka-connect, and Zookeeper to resolve related security issues.
- Fixed an issue that prevented failed `CREATE VERTEX` or `CREATE EDGE` commands from being properly rolled back.
- Fixed an issue that caused errors when a query's path pattern included edge wildcards.
- Fixed an issue that slowed the semantic checker when editing queries written in syntax V2 on GraphStudio.
- Fixed an issue that prevented GPE's memory protection from recognizing when memory usage enters the critical stage.
- Fixed an issue that could prevent vertex upsertions after the Graph Storage Engine (GSE) restarts on a single-partition (including single-node) TigerGraph instance in rare cases.

3.5.1

TigerGraph 3.5.1 was released on March 31st, 2022.

Issues fixed:

- Resolved CVE-2022-23302 and CVE-2022-23305 by removing JDBCappender and JMSsink classes from ZK and Kafka log4j jar.
- Fixed an issue that caused GraphStudio to issue too many expensive requests if the user stays on the Load Data page.
- Fixed an issue that caused Graph Storage Engine (GSE) to reuse IDs of deleted vertices.
- Fixed an issue that in rare cases prevented the Graph Processing Engine (GPE) process from restarting after being terminated.
- Fixed an issue that in rare cases caused data upsert operations to cause queries to time out.
- Fixed an issue that impacted upsert performance when upserting data to `STRING COMPRESS` data type.
- Fixed an issue that in rare cases prevented a TigerGraph cluster from recovering from a Kafka failure.
- Resolved SQL injection risks in GraphStudio.
- Added an error message when a query written in syntax V2 uses the `SAMPLE` clause of a `SELECT` statement. `SAMPLE` clauses are currently not supported in syntax V2.
- Fixed an issue that impacted performance when a query writes results to a file.
- Fixed an issue that prevented loading jobs from loading data from other disks in a file system.
- Fixed an issue that in some cases prevented GraphStudio from displaying the entire query if the query is too long.

Improvements:

- Added support for generic vertices in `INSERT INTO <edge_type>` statements.

The rest of this page provides the 3.5.0 release notes.

New Features

Below is a list of new features and improvements in TigerGraph 3.5.0:

Elasticity

- Added support for provisioning Elastic Read-only (ER) clusters to provide computation on demand.
    - Note: This feature is currently in the preview stage only available for enterprise TigerGraph Cloud customers. If you are a paid TigerGraph Cloud enterprise customer and want to set up an ER cluster for your environment, please open a support ticket.

Manageability

- Added support for removing one or more failed nodes from a cluster. After removal, users can expand the cluster back to its original size.
- Added support for backup and restore to TigerGraph deployments on Kubernetes.
- Added `gadmin configuration` parameter `KafkaStreamLL.MaxPartitionFetchBytes` to adjust Kafka Stream max partition fetch size.

Security

- Added Access Control Lists to queries for fine-grain access control.

- Added SSL/SASL support for Kafka loader.
- Masked secrets and user credentials in RESTPP logs.

Ecosystem integration

- Introduced the Data Streaming Connector with Google Cloud Storage data streaming support.

Observability

- Improved error messaging when the maximum graph size allowed by license is reached.

Reliability

- Added the `gsql-atomic-level` header to the endpoint `POST /graph/{graph_name}` to allow users to set atomicity levels in upsert requests.

Query Language Enhancement

- Changed default GSQL syntax to syntax V2. See Query Language Syntax Versions for a summary of the differences between V1 and V2.
- Added V2 support for single-hop `UPDATE` and `DELETE` statements.
- Added support for local tuple variables.
- Added support for multiple `POST-ACCUM` clauses in `SELECT` statements in syntax V1.
- Added 10 methods to `BitwiseAndAccum` and `BitwiseOrAccum` and increased their length limit.

GraphStudio Enhancement

- Added pagination support to AdminPortal metric charts.

Other tools

- Added `POST /gsql/queries/install` endpoint to install queries.
- Added `GET /gsql/queries/install/{request_id}` endpoint to check the status of query installation requests.

Fixed issues

- Fixed an issue where the installer did not check `AllowerUser` on all nodes for installation on OEL8.
- Fixed an issue that in rare cases led to GPE crash when the database is empty.
- Fixed an issue that caused large parallel loading to trigger race conditions.
- Fixed an issue that threw out-of-range errors when a 64-bit `BitwiseAccum` flips all bits with its `flip()` method.
- Fixed an issue where `PRINT` statements printed vertex accumulators declared after the `PRINT` statement.
- Improved the error message of the `/requesttoken` endpoint.
- Fixed an issue where in rare cases clusters with over 10 partitions fail to elect a GSE leader.
- Fixed an issue that caused connection refused errors when connecting to a TigerGraph cluster behind a load balancer through a remote GSQL client.
- Fixed an issue that caused the `getAttr()` function of vertices and edges to invalidate a clause even if it doesn’t get called.

- Fixed an issue that caused the `edgeAttribute()` method of a vertex to erroneously throw type check errors in syntax V2.
- Fixed an issue in GraphStudio query editor that produced inconsistent indentation when indents are autogenerated.
- Fixed an issue in GraphStudio where some metadata of a graph isn’t deleted after the graph is dropped

Deprecation warnings

- The `-OPTIMIZE` flag for `INSTALL QUERY` is deprecated and is planned to be dropped in the 3.7 release.

Known Issues

Cluster expansion

If you have ever deleted a vertex or edge attribute before conducting an expansion, cluster expansion may lead to issues reading attributes of that vertex or edge type.

For details, see Cluster Expansion.

Applications

- **GraphStudio**
  - Known Issues for GraphStudio
  - The No-Code Data Migration feature is in Alpha release. Your feedback would be appreciated.
  - The No-Code Visual Query Builder is in Beta release. Your feedback would be appreciated.
- **Admin Portal**
  - Known Issues for AdminPortal

GSQL

- **Multiple (Conjunctive) Path Patterns**:
  - There are no known functional problems, but the performance has not been optimized. Your feedback would be appreciated.
- **DML type check error in V2 Syntax:**
  - GSQL will report a wrong type check error for Query block with multiple POST-ACCUM clauses and Delete/Update attribute operation.
- **Turn on GSQL HA manually when upgrading from 3.0.x**
  - Users who are upgrading from 3.0.X need to manually start GSQL HA service. Please reach out to support for help with the process documented in: https://tigergraph.freshdesk.com/a/solutions/articles/5000865072
- **Stale data visible after Deletes using index**
  - Queries that use secondary index may still see the vertices being deleted until after the snapshots are fully rebuilt.
- **Concurrent read and write on the same vertex during loading**
  - Concurrent read and write operations on the same vertex and its connected edges during data loading could cause Graph Processing Engine (GPE) dysfunction in rare cases

Release notes for previous versions

- Release notes - TigerGraph 3.4
- Release notes - TigerGraph 3.3
- Release notes - TigerGraph 3.2

Compatibility with TigerGraph 3.4

- A single `POST-ACCUM` clause can no longer reference more than one vertex alias in Syntax V1.
- `GET /requesttoken` endpoint is dropped. Please use `POST /requesttoken` to request authentication tokens instead.
  - Using request body to store credentials is more secure than using query string. If you have a create a token request that puts the credentials in the query string, all you need to do is use the `POST` endpoint and move your credentials to the request body.
- `GET /gsqlserver/gsql/queryinfo` endpoint on port 14240 now returns the query input parameters in the same order as they are in the query instead of an unordered list.
- Deploying TigerGraph in Kubernetes now requires more service account permissions than previous versions. For details, see Prerequisites section in Quickstart with GKE, Quickstart with AKS, and Quickstart with EKS.

Compatibility with TigerGraph 3.1

The following changes were made to the built-in roles in TigerGraph’s Role-based Access Control

- The built-in role `queryreader` can no longer run queries that include updates to the database.
  - To emulate the old `queryreader` role, create a role with all `queryreader` privileges, and also grant the `WRITE_DATA` privilege to the new role.
- The built-in role `admin` can no longer create users
  - To emulate the old `admin` role, create a global role with all `admin` privileges, and also grant the `WRITE_USER` privilege to the new role.
- To learn more about role management and the privileges of built-in roles, see:
  - Role Management
  - List of Privileges

Compatibility with TigerGraph 2

Major revisions (e.g., from TigerGraph 2 to TigerGraph 3) are the opportunity to deliver significant improvements. While we make every effort to maintain backward compatibility, in selected cases APIs have changed or deprecated features have been dropped, in order to advance the overall product.

Data migration: A tool is available to migrate the data in TigerGraph 2.6 to TigerGraph 3.0. Please contact TigerGraph Support for assistance.

Query and API compatibility:

- Some gadmin syntax has changed. Notably. `gadmin set config` is now `gadmin config set`. Please see Managing with gadmin.
- Some features which were previously deprecated have been dropped. Please see V3.0 Removal of Previously Deprecated Features for a detailed list.

V3.0 Removal of Previously Deprecated Features

TigerGraph 2.x contained some features which were labeled as deprecated. These features are no longer necessary because they have been superseded already by improved approaches for using the TigerGraph platform. The new approaches were developed because they use more consistent grammar, are more extensible, or offer higher performance. Therefore, TigerGraph 3.0 has streamlined the product by removing support for some of these deprecated features, listed below:

Data Types

See Data Types in GSQL Language Reference

Deprecated Type	Alternate Approach
REAL	Use FLOAT or DOUBLE
INT_SET	Use SET<INT>
INT_LIST	Use LIST<INT>
STRING_SET_COMPRESS	Use SET<STRING COMPRESS>
STRING_LIST_CONPRESS	Use LIST<STRING COMPRESS>
UINT_SET	Use SET<INT>
UINT32_UINT32_KV_LIST	Use MAP<UINT, UINT>
INT32_INT32_KV_LIST	Use MAP<INT, INT>
UINT32_UDT_KV_LIST	Use MAP<UINT, UDT_type>, where UDT_type is a user-defined tuple type
INT32_UDT_KV_LIST	Use MAP<INT, UDT_type>, where UDT_type is a user-defined tuple type

Syntax for Control Flow Statements

See Control Flow Statements in GSQL Language Reference

Deprecated Statement	Alternate Statement
FOREACH ... DO ... DONE	FOREACH... DO... END

FOREACH (condition) {
 	
body
 	
}
	
	FOREACH condition DO
	body
	END

Deprecated Statement	Alternate Statement
IF (condition) {	IF condition THEN
body1	body1
} else {	ELSE
body2	body2
}	END
WHILE (condition) {	WHILE condition DO
body	body
}	END

Vertex Set Variable Declaration

See Vertex Set Variable Declaration and Assignment

If a vertex type is specified, the vertex type must be within parentheses.

Deprecated Statement	Alternate Statement
MySet Person = …	MySet (Person) = …

Query, Job, and Token Management

Deprecated Operation	Alternate Operation
CREATE JOB	CREATE LOADING JOB
RUN JOB	Specify the job type: RUN LOADING JOB RUN SCHEMA_CHANGE JOB RUN GLOBAL SCHEMA_CHANGE JOB
CREATE / SHOW/ REFRESH TOKEN	To create a token, use the REST endpoint GET /requesttoken
offline2online	The offline loading job mode was discontinued in v2.0. Do not write loading jobs using this syntax.

Output

See PRINT Statement in 'Output Statements and File Objects'

Deprecated Syntax	Alternate Syntax
JSON API v1	v2 has been the default JSON format since TigerGraph 1.1. No alternate JSON version will be available.
PRINT … TO_CSV [filepath]	Define a file object, then PRINT … TO_CSV [file_object]

Built-in Queries

Deprecated Statement	Alternate Statement

SELECT count() FROM ...
// count may be out of date
	SELECT approx_count(*) FROM ...
	// same behavior as count(); may not include all the latest data updates
	
	SELECT count(*) FROM ...
	// exact, but slower than approx_count(*)

3 Twin Dolphin Drive, Ste 225
Redwood City, CA 94065

Resources

Support

Community

Developer Site

Test Drive

Social

Linkedin

Facebook

Twitter

Legal

Privacy Policy

Terms of Use

Sitemap