LOGZILLA DOCUMENTATION

Server Migration

Migrate a LogZilla installation to a new server using the migrate-dump CLI, including version compatibility rules and post-restore license updates

Administration · Generated June 12, 2026 · logzilla.ai/docs/administration/server-migration

Migrating LogZilla to a new server involves transferring all data and configuration using the built-in `migrate-dump` command. This modern approach provides a safe, reliable method for moving LogZilla installations between servers while maintaining data integrity.

Migration Overview

The migration process uses LogZilla's dedicated `migrate-dump` command:

Create dump: Generate a complete backup on the source server
Transfer dump: Move the backup files to the destination server
Restore dump: Import the backup on the destination server
Update license: Configure licensing for the new server

Version Compatibility

The `migrate-dump` command has specific version requirements:

- **Same versions**: Works without restrictions
- **Newer dump version**: Will not work (dump version > destination version)
- **Older dump version**: Requires `--force` option (dump version < destination version)

Important: Ensure version compatibility before starting the migration process. The safest approach is using identical LogZilla versions on both servers.

Source Server Migration

Step 1: Create Migration Dump

Create a complete backup of the LogZilla installation using the migrate-dump command. Specify a directory to store the dump files:

```
logzilla migrate-dump create --dump-dir /tmp/logzilla-migration
```

This command creates a full backup including:

- All log data and archives
- Database contents (PostgreSQL, InfluxDB, Redis, etcd)
- Configuration settings
- User accounts and permissions

Note: Ensure sufficient disk space is available in the dump directory. The backup size depends on the amount of stored log data.

Step 2: Transfer Migration Files

Copy the migration dump to the destination server using scp, rsync, or another secure transfer method:

```
scp -r /tmp/logzilla-migration user@destination-server:/tmp/
```

Alternatively, use rsync for large transfers:

```
rsync -avz /tmp/logzilla-migration/ user@destination-server:/tmp/logzilla-migration/
```

Destination Server Setup

Step 1: Install LogZilla

Install LogZilla on the destination server. Ensure the version matches or is newer than the source server version:

```
logzilla install
```

Step 2: Restore Migration Dump

Restore the complete LogZilla installation from the migration dump:

```
logzilla migrate-dump restore --dump-dir /tmp/logzilla-migration
```

For version compatibility issues, use the force option if the dump version is older than the destination version:

```
logzilla migrate-dump restore --dump-dir /tmp/logzilla-migration --force
```

The restore process automatically:

- Stops LogZilla services if running
- Restores all data and configuration
- Updates internal references and paths
- Starts LogZilla services after completion

Step 3: Update License

After successful restoration, update the license for the new server:

```
logzilla license download
```

This command replaces the migrated license with a demo license valid for the new server. Contact LogZilla support to extend the license as needed.

Post-Migration Verification

Monitor Startup Progress

LogZilla initialization time depends on data volume. Monitor startup progress using the API endpoint:

```
http://your-server.com/api/monitor
```

Check System Logs

Monitor LogZilla logs for any errors during startup:

```
tail -f /var/log/logzilla/logzilla.log -n 40
```

Verify Data Integrity

Once LogZilla starts:

- Confirm historical data appears in the interface
- Verify user accounts and permissions
- Test log ingestion from sources
- Validate dashboard and trigger configurations

Troubleshooting

Common Issues

Migration Dump Creation Fails

- Verify sufficient disk space in the dump directory
- Check LogZilla service status before creating dump
- Review system logs for specific error messages
- Ensure proper file permissions on dump directory

Version Compatibility Errors

- Confirm LogZilla versions on source and destination servers
- Use `--force` option for older dump versions (with caution)
- Consider upgrading destination server if dump version is newer

Restore Process Fails

- Verify dump directory exists and contains valid migration files
- Check destination server has sufficient disk space
- Ensure network connectivity during restore process
- Review restore logs for specific error details

License Issues

- Ensure `logzilla license download` completed successfully
- Contact support if demo license expires before production license
- Verify network connectivity for license server communication

Legacy Migration Method

For older LogZilla versions without `migrate-dump` support, the manual Docker volume method may be required. Contact LogZilla support for guidance on legacy migration procedures.

Best Practices

- **Test migrations** in a staging environment first
- **Verify version compatibility** before starting migration
- **Schedule downtime** during low-usage periods
- **Monitor disk space** on both source and destination servers
- **Validate migration** thoroughly before decommissioning source server
- **Coordinate with teams** that depend on LogZilla services