

Migrating Ubersmith Appliances 3.x to 4.x

Read this document completely before starting the migration process, to ensure the necessary access is available to complete all steps. For these steps, the **new appliance** is the system the Ubersmith Appliance is being migrated to, and the **old appliance** is the system the Ubersmith Appliance is being migrated from. It is assumed that the **Ubersmith host is upgraded first**, and that the Ubersmith Appliance will be installed to the following default directory: `/usr/local/ubersmith`

If you would prefer to have Ubersmith perform your migration, please contact [Ubersmith Support](#) to discuss the scope of the project and to arrange for a quote from our Managed Services team.

Preparing for the Migration

Perform these steps before migration day.



The RRD (round robin database) files that the Ubersmith Appliance uses to collect infrastructure data are architecture dependent. If you have a particularly old appliance running an `x86` architecture, and choose to migrate to an `x86_64` architecture, the RRD files will not function properly on the new appliance.

Ensuring Connectivity

- Ensure that the **old appliance** can reach the **new appliance** via SSH.
- Ensure that the Ubersmith host can reach the **new appliance**. The **new appliance** will accept connections on `8080/tcp` and `8443/tcp`.
- Configure an SSH key between the two appliances to ease file copying and database dumps. [See this online resource to perform this configuration task.](#)

Deploying the New Ubersmith Appliance

- Ensure that the **new appliance** system is [compatible](#) with Ubersmith and install the Ubersmith Installation and Upgrade Utility [prerequisites](#).
- Deploy the Ubersmith Appliance on the **new appliance** using the Ubersmith [Installation and Upgrade Utility](#).
- Once the installation is complete, disable the `app_cron` container on the **new appliance** environment. This ensures that the new system does not start polling your devices before you are ready to use it.

```
cd /usr/local/ubersmith
docker-compose rm -sf app_cron
```

Scheduling Downtime/Dry Run

To get a time estimate of how long the Ubersmith Appliance will be offline, perform a dry run by performing the migration day steps, but not the post migration steps.

- Note the timing for the following events:
 - Database dump
 - Database transfer
 - Database restore
 - RRD transfer
 - Ubersmith Appliance upgrade
- Based on the details of the dry run, schedule downtime with your users. For extended downtime, consider performing the migration after hours or on a weekend. Appliances with large data sets will experience significant downtime during the RRD transfer phase.

Migration Day

These steps perform the migration and may cause the Ubersmith Appliance to be unavailable.

Exporting the Database

- By default, the Ubersmith Appliance database is named `uberapp`.
- Complete a logical export using `mysqldump`, which is likely already installed on the system. See the `mysqldump` documentation for [MySQL 5.6](#). In most circumstances, the Ubersmith Appliance database is small, and will be at most a few hundred megabytes.
- Example usage of `mysqldump`:

```
mysqldump -u root -p uberapp > ubersmith_appliance_export.sql
```

On this page:

On this page:

- [Preparing for the Migration](#)
- [Ensuring Connectivity](#)
- [Deploying the New Ubersmith Appliance](#)
- [Scheduling Downtime/Dry Run](#)
- [Migration Day](#)
- [Exporting the Database](#)
- [Copying the Database](#)
- [Upgrading the Ubersmith Appliance](#)
- [Disabling Poll Tasks](#)
- [Migrating RRD Files](#)
- [Post Migration](#)
- [Enabling Scheduled Tasks](#)
- [Updating Ubersmith's Appliance Configuration](#)
- [Related Topics](#)

- In this command the `root` user logs into MySQL, the password is prompted for and entered on the command line, the `uberapp` database is exported, and saved to a file named `ubersmith_appliance_export.sql`.
- Once the database export is complete, use the `tail` command to examine the end of the file to ensure it completed successfully.

```
tail -n1 ubersmith_appliance_export.sql
```

- A successful completion will show the output `-- Dump completed on with a date and timestamp`.

Copying the Database

- Copy the appliance database export file from the **old appliance** to the **new appliance**, using `scp` or `sftp`.
- Copy the appliance database export file into the **new appliance's** `app_db` container, using the `docker cp` command:

```
docker cp ubersmith_appliance_export.sql ubersmith_app_db_1:/
```

- Enter a shell in the **new appliance** database container (`app_db`) to begin the restore:

```
cd /usr/local/ubersmith
docker-compose exec app_db bash
```

- Within the **new appliance's** shell in the database container, restore the database:

```
mysql -u root -p$MYSQL_ROOT_PASSWORD uberapp <
/ubersmith_appliance_export.sql
```

Upgrading the Ubersmith Appliance

- On the **new appliance**, run the `appliance_upgrade.sh` script which is part of the [Ubersmith Installation and Upgrade Utility](#).

Disabling Poll Tasks

- If simultaneous polling between two appliances is not desirable, stop the polling tasks on the **old appliance**. In a typical install, the cron tasks are owned by the `ubersmith` user. They can be deleted or commented out. To edit the `ubersmith` user's crontab, run this command as root:

```
crontab -e -u ubersmith
```

- Disabling these tasks ensures that the **old appliance** does not poll your infrastructure during the migration process. If this step is not performed now, be sure to perform it once the migration is complete.

Migrating RRD Files

- Copy (`rsync`, `scp` or `sftp`) the existing RRD files from the **old appliance** to the **new appliance**. The existing RRD files can be found on the **old appliance** in the directory:

```
/home/uberm/appliance/rrds
```

- On the **new appliance**, the RRD files are stored in a Docker volume named `ubersmith_rrds`. The simplest way to migrate the RRDs is to copy them directly into Docker's storage location for the volume:

```
/var/lib/docker/volumes/ubersmith_rrds/_data/
```

Post Migration

These steps finalize the migration and configure Ubersmith to use the new appliance.

Enabling Scheduled Tasks

- On the **new appliance**, recreate the cron container, so that the Ubersmith Appliance can perform scheduled polling tasks.

```
cd /usr/local/ubersmith
docker-compose up -d app_cron
```

- Permissions must be set on the RRD files within the Ubersmith Appliance cron container, so that the cron tasks can write to them. Execute the following command in `/usr/local/ubersmith:`

```
docker-compose exec app_cron chown -R ubersmith:ubersmith /var/www/appliance_root/rrds
```

Updating Ubersmith's Appliance Configuration

- When you are satisfied that the Ubersmith Appliance is functioning properly, renumber the **new appliance** with the IP address from the **old appliance**, or update Ubersmith's Appliance configuration (Settings > Appliances) to use the new address.
- Be sure to disable polling on the **old appliance** if it was not performed earlier in the migration process.

Related Topics

[Implementation](#)

[Ubersmith Installation and Upgrade Utility](#)

[Migrating Ubersmith 3.x to 4.x](#)