

SnowMirror | Maersk Case Study

Maersk IT's reporting needs

Maersk IT went live with our ServiceNow installation in 2015. To enable bespoke reporting, we had previously integrated a BI tool with the database using our previous service management tool. This presented us with performance and database issues, so we decided to stick with the native reporting capability in ServiceNow and Performance Analytics. We also decided to minimise the number of interfaces into ServiceNow to export data, to maintain data integrity and performance.

Our original reason for procuring SnowMirror was to enable tools direct access to a database of CMDB information to enrich data within those tools - without them needing to directly query the ServiceNow database.

Over time, it became clear that ServiceNow reporting was not optimal for certain reporting requirements. Users were taking regular manual exports from ServiceNow to develop their own reports in dashboards within a BI tool. This caused a lot of overhead for teams, so we endeavoured to create a replica database which we could connect to BI tools. We wanted to be able to store data within the database on a requirement basis rather than making all ServiceNow tables available.

Comparative advantages of SnowMirror over other replication tools

SnowMirror was recommended to us, so we did not consider any other replication tool. SnowMirror stands out due to its easy setup and API capabilities.

How Maersk IT uses SnowMirror

We implemented SnowMirror on cloud infrastructure and use a tiered structure for account access to the tables within the database. Each tool with an interface to

the database has its own account with the access level defined, so that we can ensure users/tools only have access to consume the data that they require. We have connections to multiple security tools consuming CMDB information and our chosen BI tooling.

We have created a service offering within ServiceNow for managing the synchronizations within SnowMirror along with access to the data. This includes the use of Catalog items within ServiceNow for requesting new and updates to current synchronizations. These items tie into a custom module that we have created to maintain synchronizations directly from ServiceNow, leveraging the SnowMirror REST API.

We also use the REST API to run daily status updates and automatically generate incident tickets if there are any failures - pulling back synchronization run information with detailed logs. We can then conduct debugging and re-run the synchronization without having to access the SnowMirror Application Server.

Level of satisfaction

We have been very happy with SnowMirror's performance and have had no issues so far. We have decided to stick with a daily import for all data, so the performance impact on ServiceNow is minimal.

We contacted SnowMirror support and the response was quick and extremely helpful in debugging any issues. The detailed logging in SnowMirror helps with this. We found the documentation and guidance on the SnowMirror website very helpful. The API document was useful, but it did lack some detail on the authentication header that needs to be sent, and the fact that creating a new sync requires the "sys_id" column to be sent.

Key take-aways

- Clear and easy setup for SnowMirror
- REST APIs key for operationalising SnowMirror
- Fantastic SnowMirror support