RDS

The RDS instance will host the SQL Server database for the installation's cloud-based operations. The exact nature of these operations depends on the deployment type that was selected for the client:

- Kensium Hosted – the RDS database is the client's corporate/hub database, and acts as the central database for integrated POS functions.
- Client Hosted – the RDS database behaves more like a POS store database, enabling public transactions (such and mobile and ecommerce orders) to be served through cloud infrastructure.

> For more information about deployment types, review the internal POS Deployment Scenarios document.

The table structure of the database does not differ according to the deployment type. However, data flows and potential data sizes are different, and have an impact on determining the hosting requirements of the database.

Database Version

The database version to use for the client database on RDS. This is one of:

- SQL Server 2019 Express
  - Use this version for smaller clients that use the Client Hosted deployment model, with no databases that will exceed 10 GB.
- SQL Server 2019 Web
  - Use this version for other clients.
- Custom

> Avoid custom SQL versions. The version of the database in the RDS environment does not need to be the same as the SQL Server version that is deployed at client environments.

There are a few ways that Kensium can host the client's database in RDS:

- Shared – the client database can be located on an existing RDS instance, along with other clients' databases.
- Shared (New) – the client database can be located on an RDS instance, but it must be provisioned first. Once created, the RDS instance can also be used to host future client databases.
- Dedicated – the client database should be created on a new RDS instance. The instance will not be used to host future client databases.

> Dedicated instances should normally not be created. Only create a dedicated instance if it is requested by the Kensium services manager.

Indicate the name of the RDS instance that will host the client database. The RDS instance database edition must match the the Database Version selected for the client.

Depending on the Database Version and RDS Hosting Type you selected above, and the load on existing RDS instances, you may need to create a new RDS instance first.

An example RDS instance name is db-pos-01.

Enter the availability zone (AZ) where the RDS instance is located. Confirm that this AZ matches the AZ assigned to the client.

The next step is to create the client database on the RDS instance. This can be done one of two ways:

- If the client has an existing Kensium database, ***and*** the deployment type (from the client pre-checklist) is Kensium Hosted, the schema and data should be copied from the client's existing Kensium Corporate database into a new database on RDS.

- Otherwise, create an empty database on RDS.

In either case, the new database should follow standard POS naming conventions:

- Follow the pattern pos_orgid
- For example, pos_xmsqa1
- The name should be lower-case.

> To access the RDS instance, you may need to configure a rule for your IP address in the SG-POS-OPS security group for the MSSQL port. If so, remember to add your name beside the rule to make it easier to update the rule later if your IP address changes.

Create a new database login on the RDS instance's SQL Server:

- Specify the login name.
  - The login name should follow the pattern pos_orgid_user.
  - For example, pos_xmsqa1_user.
- Use SQL Server authentication.
- Create a random, secure password.
  - Use the Kensium Password Manager to create a secure password.
- Uncheck Enforce password expiration.
- Set the default database to the client's database you just created, e.g pos_xmsqa1.
- Map the database login to a new user on the client database (with the same name), and give that user dbowner role membership.

> The client's database user should only have access to their client database. The user must not be able to access other client databases.

KENSIUM POS ⇄ Acumatica

Record an entry into the Kensium password manager:

- The entry name should follow the pattern orgid – Cloud (MSSQL).
- For example, xmsqa1 – Cloud (MSSQL).
- Record the username and password.
- In the Notes field, enter the connection string to the database.
  - Use the full public hostname of the RDS instance as the data source.
  - Use the name of the database you created above.
  - Use the database user and password you created above.
  - Example: Data Source=db-pos-01.c8hkiftu3gwj.us-west-2.rds.amazonaws.com;Database=pos_xmsqa1;User ID=pos_xmsqa1_user;Password=XXXXXXXXX
- Save the record to the Client Credentials collection in the Kensium organization.

> Always use the Kensium password manager to access client credentials. Do not store or transmit these in files, email or other communications.

Next, you will need to provide access to the RDS instance for the client's local network(s).

The preferred approach to this is to use a VPN. That approach is not documented in this guide, however. An alternative approach is to create an inbound rule entry in the AWS SG-POS-DB security group:

- Port is MSSQL (1433).
- Source is the IP address of the client network, e.g. 192.0.0.1/32.
  - Use a /32 mask when possible to identify specific IP addresses.
  - Use other subnet masks only if the client owns the IP address space.
- Enter the organization ID for the description, and if the client network is for a specific store, the ID of the store.
  - e.g. orgid or orgid / Store1

Once configured, verify that the client network can access the RDS instance.

Consider reviewing and using the VPN approach for client connectivity. Using Amazon security groups to white-list IP addresses has the following drawbacks:

- It can be time-consuming to maintain.
- It is prone to errors if the client IP addresses change.
- For Kensium-hosted deployments, where stores communicate directly with the RDS instance, it introduces a security vulnerability as store networks may not be as secure as corporate networks.

M

- Creating a new RDS instance