



Moving Passwordstate Database to a New SQL Server

Table of Contents

1	OVERVIEW.....	3
2	BACKUP PROCESS	4
3	RESTORE PROCESS	6
4	FINAL CHANGES TO YOUR PASSWORDSTATE WEBSITE	9

1 Overview

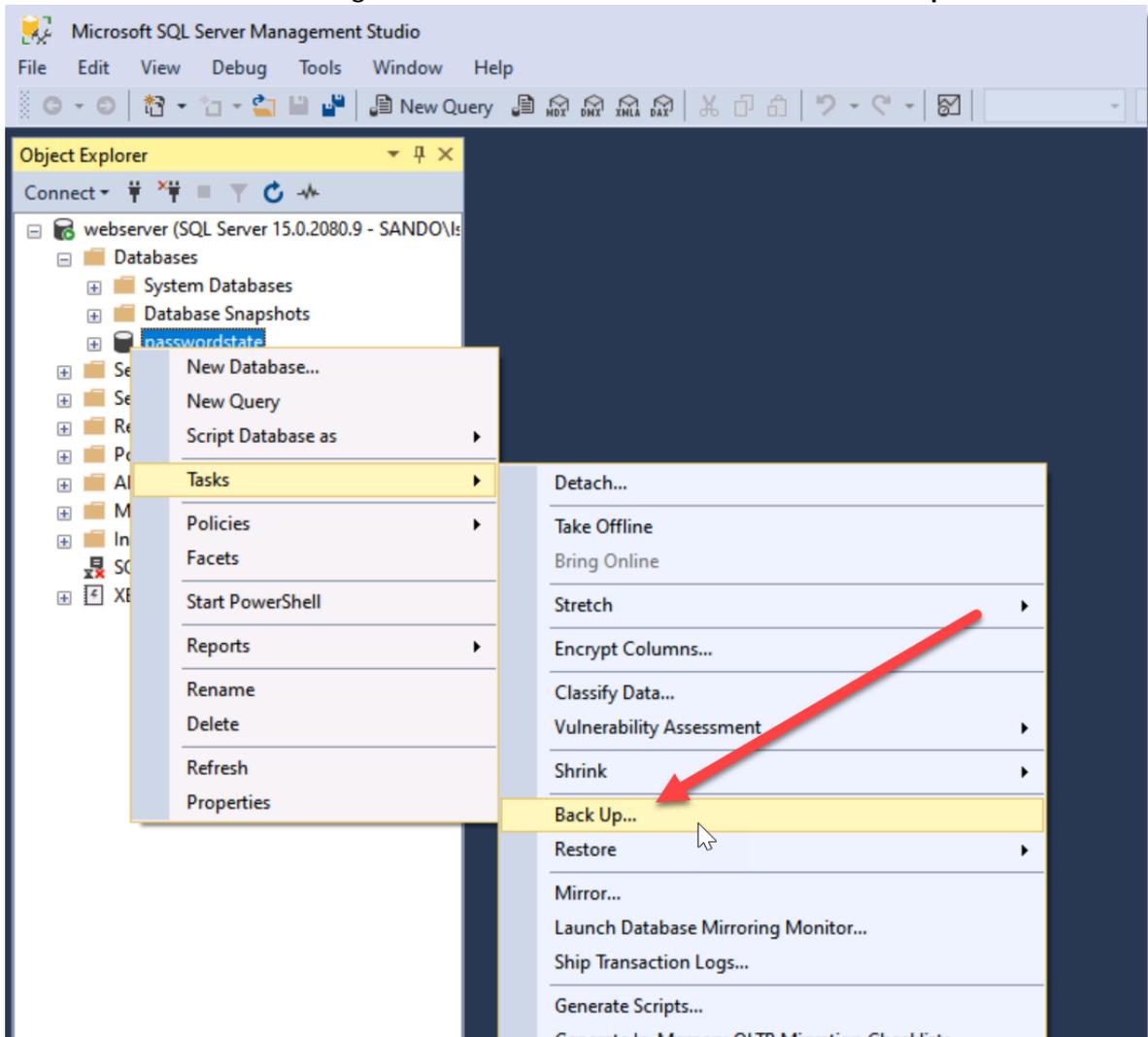
Please use the following instructions as a guide for moving an existing Passwordstate database to a new database server.

These instructions will assume you have some knowledge for using the Microsoft's SQL Server Management Studio tool, and have appropriate permissions to restore databases.

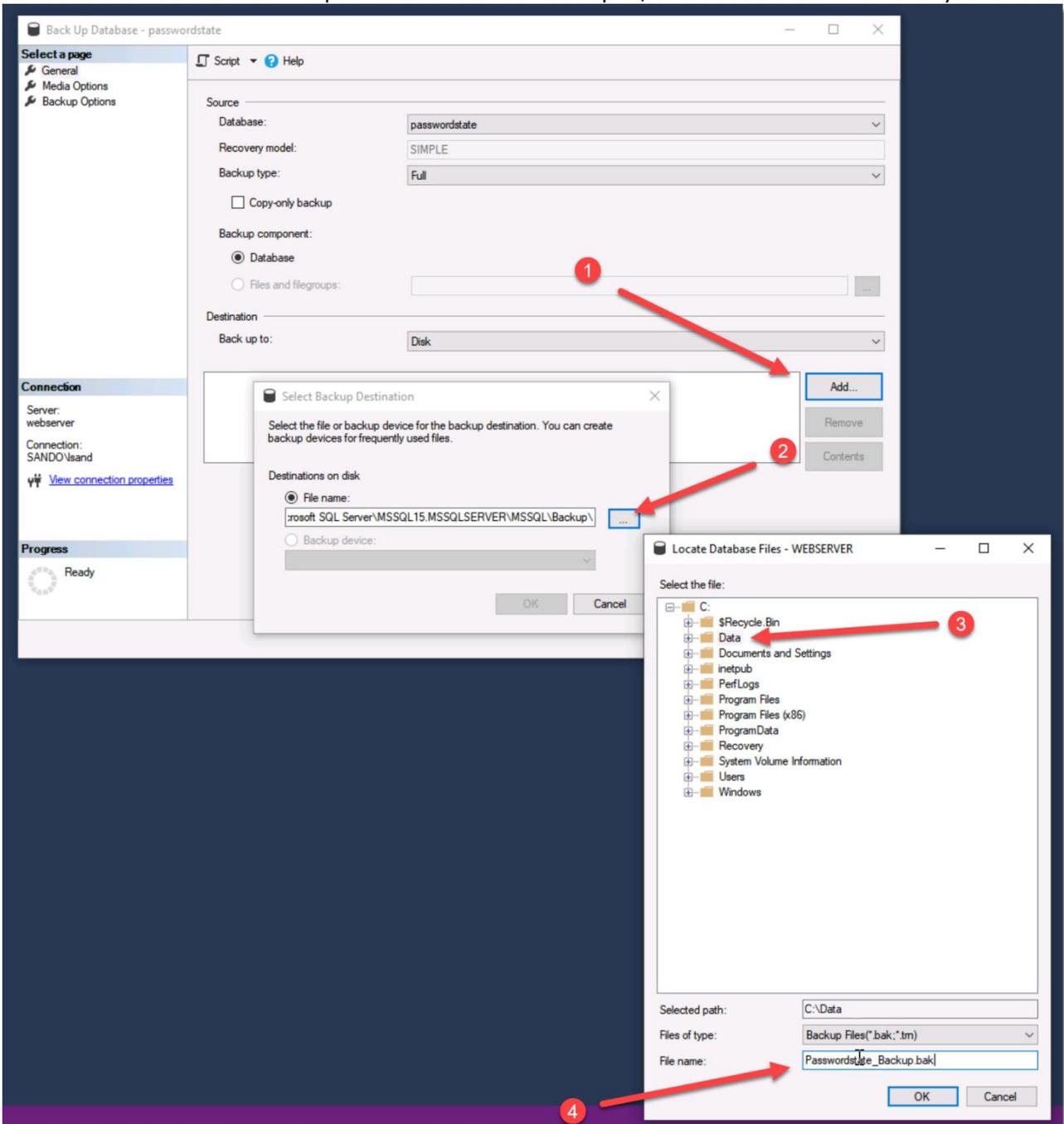
2 Backup Process

Passwordstate has a built-in backup feature that can backup your Passwordstate database, but if you have not configured backups within Passwordstate, you can instead perform this process using SQL Management Studio Tools (SMSS).

1. Open **SMSS** and connect to your current database server, using an account that has permissions to perform backups.
2. Expand out the **Databases** and then right click "**Passwordstate**". Select **Tasks** -> **Back Up...**



3. You can either leave the default path to back the database up to, or choose a custom folder of your choice:



4. Ensure you append the File name with **.bak**.

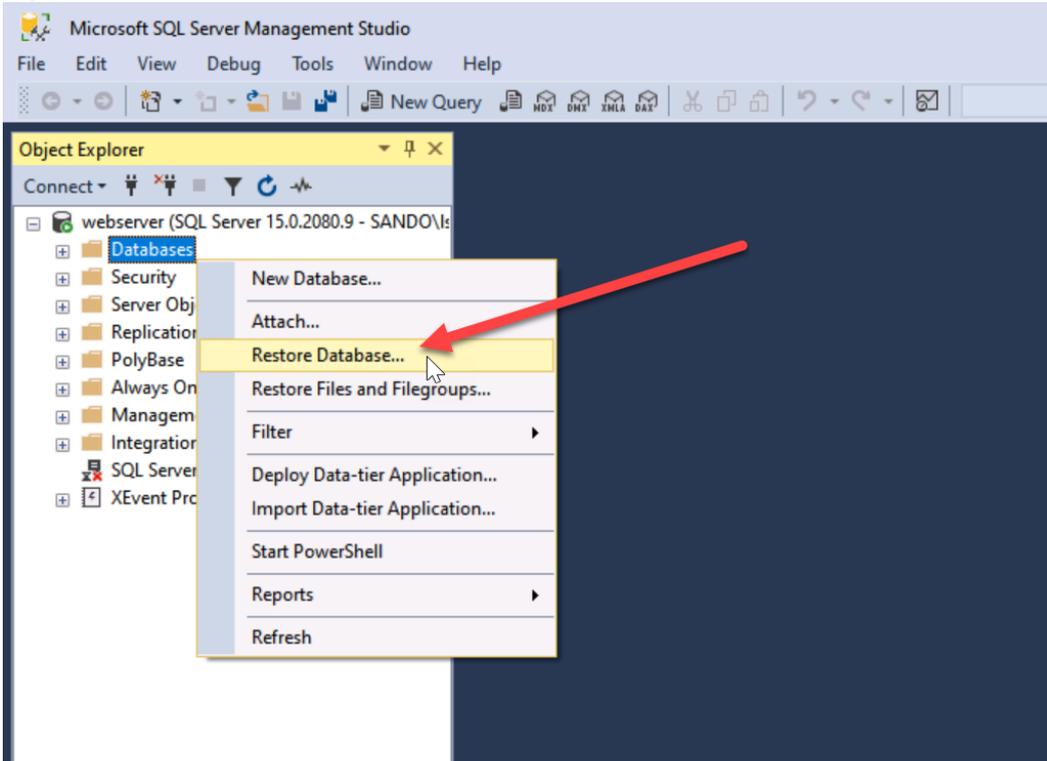
5. Transfer this file to your new SQL Server into a folder of your choice

****Note**** this file is being backed up to your database server, not the computer from where you are launching SMSS from.

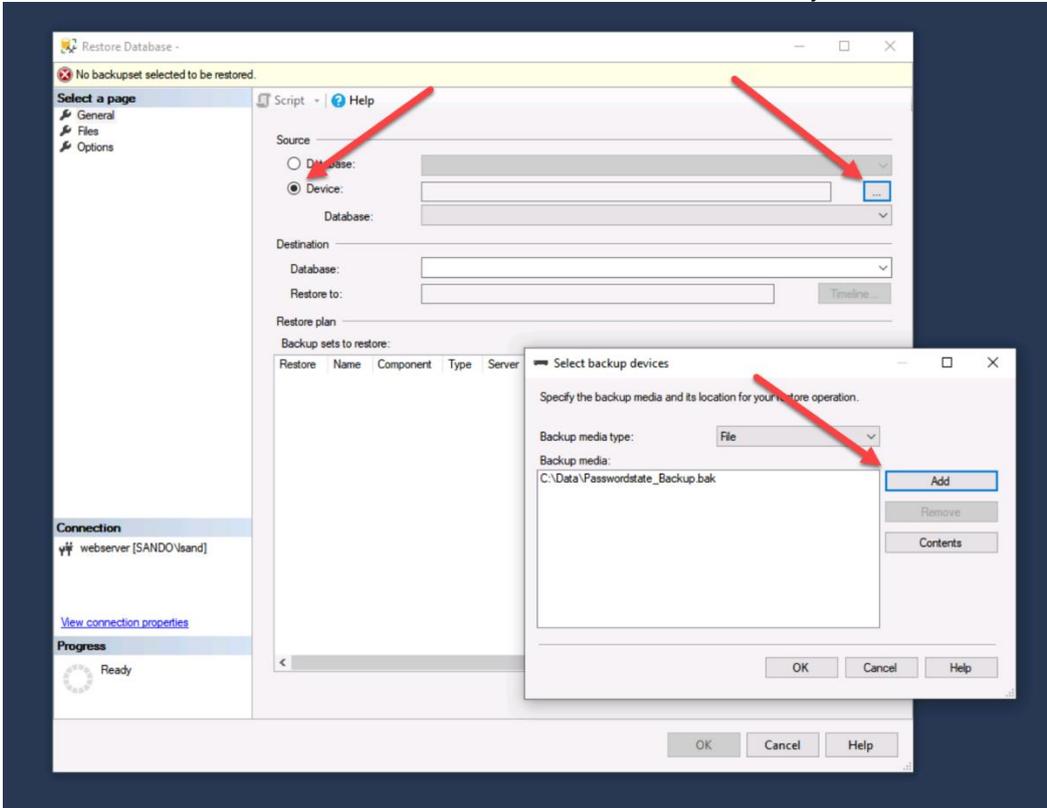
3 Restore Process

Now we will restore the database to a new server where you have already installed Microsoft SQL Server.

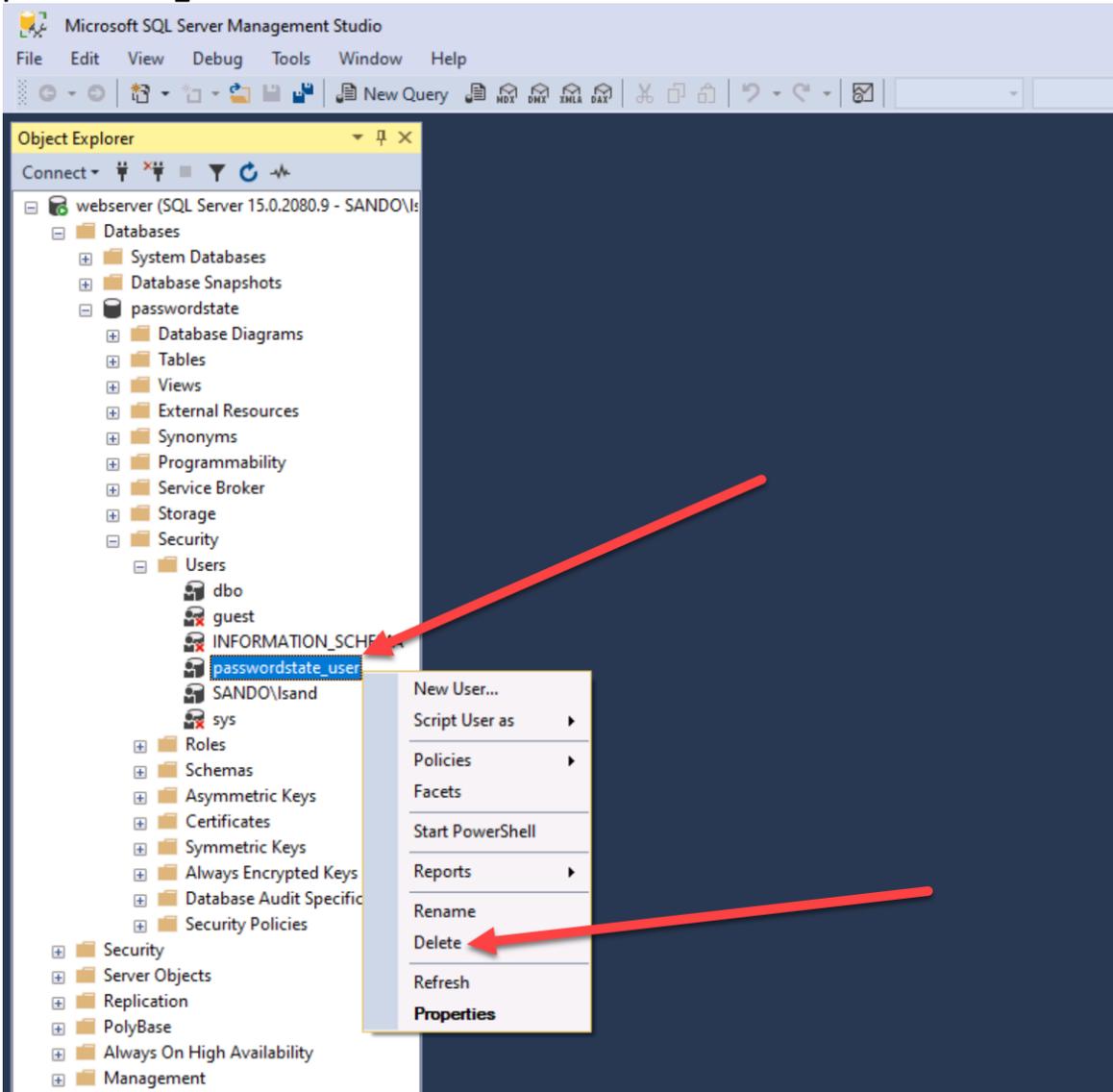
1. Log into the new database server using an account that has permissions to restore databases.
2. Right click **Databases**, and click **Restore Database...**



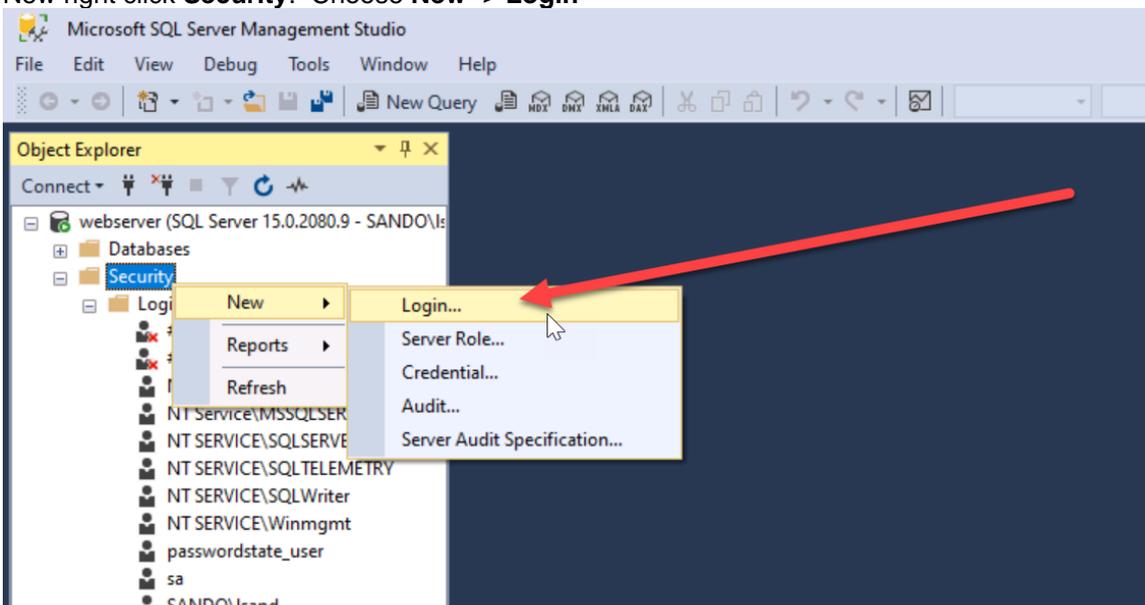
3. Select **Device**, click the **three dots** and then browse to the **.bak** file you wish to restore:



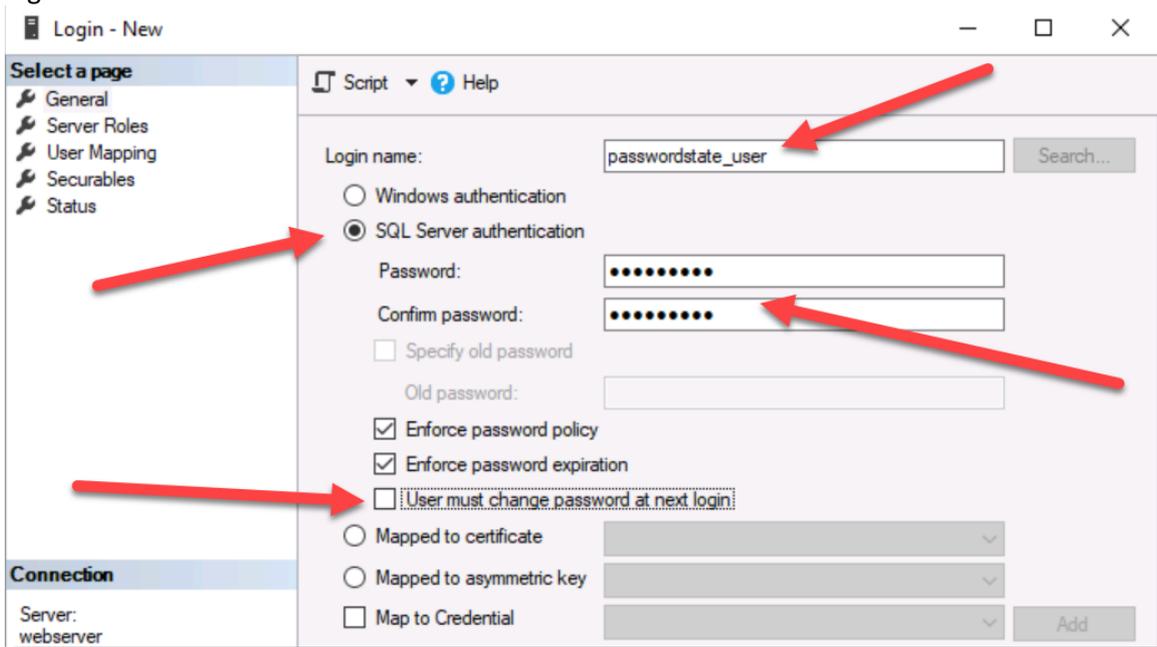
4. You should now see your Passwordstate database under the **Databases** folder. Expand out your Passwordstate database, and then expand out your **Security** -> **Users** folder. Right Click and **Delete** the **passwordstate_user**:



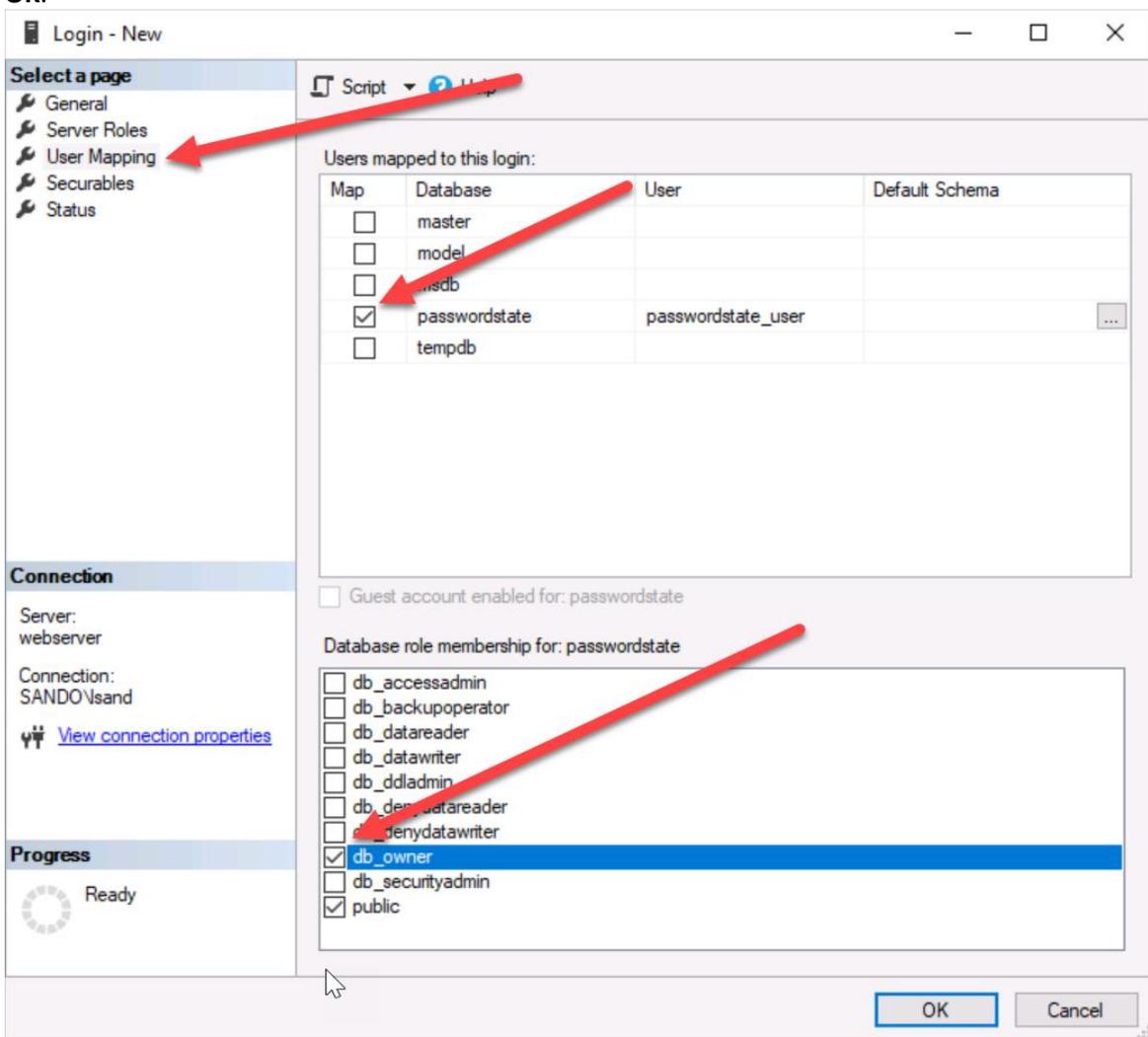
5. Now right click **Security**. Choose **New** -> **Login**



6. Set the username as **passwordstate_user**, tick the **SQL Server Authentication** option, set a password of your choice and then uncheck the option so the user does not need to change their password upon next login:



7. Under the **User Mapping** menu, tick the **Passwordstate** database, and check the **db_owner** option. Click **OK**.



4 Final Changes to your Passwordstate website

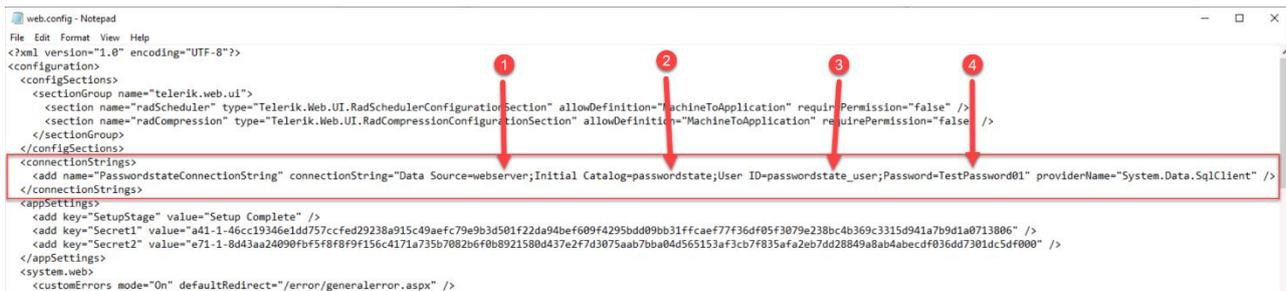
Now the Database is migrated to the new SQL server, we need to configure the Passwordstate website to connect it to this new location for the database. We do this by editing the **web.config** file on the Passwordstate web server.

The **web.config** file is located in the **c:\inetpub\passwordstate** folder on your web server. This is the default location, but it's possible to install Passwordstate in any location of your choice. This forum post describes how to find the location of your Passwordstate install folder:

<https://forums.clickstudios.com.au/topic/1668-how-to-find-where-passwordstate-is-installed/>

When opening the **web.config** file to modify it, you should see a section called **connectionStrings**. Inside this tag is 4 pieces of information:

1. The database server name (which includes the instance name if it is **sqlexpress** or a custom instance)
2. The database name (by default it's "**passwordstate**" unless you change it)
3. The SQL user which has **db_owner** rights to the Passwordstate database
4. And the password for that SQL user.



```
web.config - Notepad
File Edit Format View Help
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <configSections>
    <sectionGroup name="telarik.web.ui">
      <section name="radScheduler" type="Telarik.Web.UI.RadSchedulerConfigurationSection" allowDefinition="MachineToApplication" requirePermission="false" />
      <section name="radCompression" type="Telarik.Web.UI.RadCompressionConfigurationSection" allowDefinition="MachineToApplication" requirePermission="false" />
    </sectionGroup>
  </configSections>
  <connectionStrings>
    <add name="PasswordstateConnectionString" connectionString="Data Source=webserver;Initial Catalog=passwordstate;User ID=passwordstate_user;Password=TestPassword01" providerName="System.Data.SqlClient" />
  </connectionStrings>
  <appSettings>
    <add key="SetupStage" value="Setup Complete" />
    <add key="Secret1" value="a41-1-46cc19346e1dd757ccfed29238a915c49aefc79e9b3d501f22da94bef609f4295bdd09bb31ffcae77f36df05f3079e238bc4b369c33150941a7b9d1a0713806" />
    <add key="Secret2" value="e71-1-8d43aa24090fbf5f8f8f9f156c4171a735b7082b6f0b89215800437ezf7d3075aab7bba04d565153af3cb7f835afa2eb7dd28849a8ab4abecd036dd7301dc5df000" />
  </appSettings>
  <system.web>
    <customErrors mode="On" defaultRedirect="/error/generalerror.aspx" />
  </system.web>
</configuration>
```

You should update the information in your web.config file to be relevant to your new database server, and click **Save**. You should now be able to log into your Passwordstate website successfully, and it will be connected to your new database server.

****note**** if you cannot see the **connectionStrings** section in your **web.config** file, this most likely means it is encrypted for extra security. In this scenario you will need to decrypt it in order to see it in clear text, and make the necessary changes.

More information on decrypting/encrypting the **web.config** file can be found here:

<https://forums.clickstudios.com.au/topic/2699-encrypting-and-decrypting-the-webconfig-file/>