Passwordstate

Enterprise Password Management

Moving Passwordstate Web Site

And Database to a New Server

Table of Contents

1 OVERVIEW....................................3

2 PREPARATION REQUIRED....................................4

3 INSTALL PASSWORDSTATE ON THE NEW WEBSERVER....................................5

4 MIGRATING THE DATABASE....................................6

1 Overview

Please use the following instructions as a guide for moving an existing **Passwordstate Website** and **Database** to a new server.

This guide assumes you are currently running either Passwordstate with major build **7, 8 or 9**.

When this process is finished, you will be running the latest version of Passwordstate 9, so an upgrade of your database will occur unless you are already running the latest version.

This guide assumes you are intending on using a self-hosted **Microsoft SQL install**, as opposed to **Managed Cloud Database Services** such as **Azure SQL Databases** or **Amazon RDS**.

It is possible to use a cloud based virtual machine running **Windows Server** for this process, and this will require you to install **Microsoft SQL** on this virtual machine.

****Note**** You can leave your currently production instance of Passwordstate running until you can confirm that the migration has worked. Once you can confirm the new website work then you should decommission the old server.

2 Preparation Required

1. Ensure your new server is built with an operating system that meets the Passwordstate system requirements: https://www.clickstudios.com.au/passwordstate-system-requirements.aspx

2. On this server, install **SQL Server** in **Mixed Mode Authentication**. Links to SQL Server install guides can be found on Click Studios' documentation page: https://www.clickstudios.com.au/documentation/

3. If you are intending on using the same URL on the new server, take note of any **DNS record** you have for your current URL. Also check your **HTTPS bindings** on your current webserver in **IIS** (Internet Information Services) as you will be re-creating these bindings on the new server. (IIS Bindings Tutorial can be found here: https://forums.clickstudios.com.au/topic/15335-how-to-find-my-current-https-iis-binding-and-ssl-certificate-for-passwordstate/)

4. Whilst still in **IIS**, take note of the **Certificate** you are using on your **HTTPS binding**, and install it on your new server ready to assign to the new website once installed.

5. Log in to your current Passwordstate instance and add the new web server **Netbios** name to the list of authorized web servers within Passwordstate. The **Server Role** you specify for this when adding it in should be "**Primary Server**". You can do this from the screen **Administration** -> **Authorized Web Servers**.

6. Take note of your **Emergency** password, just in case you need to use this later in this migration process. This password can be found under **Administration** -> **Emergency Access**

7. Take a backup of your existing Passwordstate instance, which includes the web installation files, and the database. Passwordstate has its own backup feature you may have already configured you can use under **Administration** -> **Backups**, or you can use your own 3rd party backup solution if you have one. Another option is to back up the data manually using this guide: https://forums.clickstudios.com.au/topic/13911-manual-backup-using-sql-management-studio-tools/

3 Install Passwordstate on the New Webserver

1. On your new server, you will now need to download the appropriate Passwordstate installation files from Click Studios check sums page: https://www.clickstudios.com.au/passwordstate-checksums.aspx

    a. If your current version of Passwordstate is less than build **8995**, then you will need to download the build **8995** source files

    b. If your current version is **8995** or above, you should instead download the latest version of the **Version 9** source files.

2. Now you can follow the General Installation Instructions to install Passwordstate, stopping when you complete **Section 5**. You should not proceed to **Section 6 - 'Configuring Passwordstate for First Time Use'**

3. If you have just installed build **8995**, please grant the built in IIS account called '**Network Service**' **Modify** access to the '**c:\inetpub\passwordstate**'. Ensure these new permissions also apply to all nested files and folders. If you have installed a version later than **8995**, you can disregard this step completely.

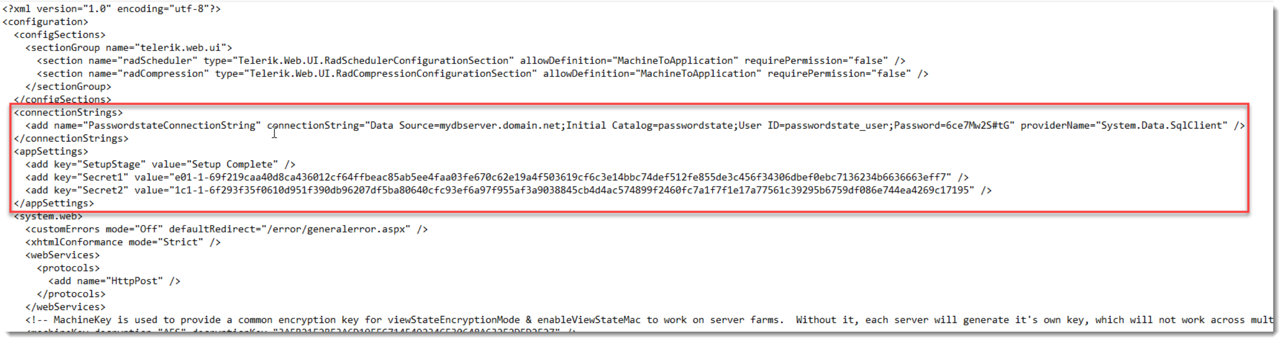
4. Next, configure your **HTTPS IIS Binding** to match your existing web server if you intend on keeping the same URL. Otherwise set your HTTPS binding to any new value you like.

5. Assign your **SSL Certificate** to your HTTPS binding that you previously prepared on your new server. If you intend on using a new certificate, more information about SSL Certificates can be found here: https://forums.clickstudios.com.au/topic/2978-passwordstate-certificates-explained/

8. You will most likely be using an existing **CNAME DNS** entry to resolve the URL to your old web server. You will now need to edit the DNS entry and update the host name of your new web server, or you can create a new CNAME entry pointing to the new server.

9. Next you need to copy the entire **ConnectionStrings and AppSettings** sections from your existing installation's web.config file into the new website. (The default location for the web.config file is C:\inetpub\Passwordstate\web.config)

    Example of these lines that need copying can be seen in screenshot below:



```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <configSections>
    <sectionGroup name="telerik.web.ui">
      <section name="radScheduler" type="Telerik.Web.UI.RadSchedulerConfigurationSection" allowDefinition="MachineToApplication" requirePermission="false" />
      <section name="radCompression" type="Telerik.Web.UI.RadCompressionConfigurationSection" allowDefinition="MachineToApplication" requirePermission="false" />
    </sectionGroup>
  </configSections>
  <connectionStrings>
    <add name="PasswordstateConnectionString" connectionString="Data Source=mydbserver.domain.net;Initial Catalog=passwordstate;User ID=passwordstate_user;Password=6ce7Mw2S#tG" providerName="System.Data.SqlClient" />
  </connectionStrings>
  <appSettings>
    <add key="SetupStage" value="Setup Complete" />
    <add key="Secret1" value="e01-1-69f219caa40d8ca436012cf64ffbeac85ab5ee4faa03fe670c62e19a4f503619cf6c3e14bbc74def512fe855de3c456f34306dbef0ebc7136234b6636663eff7" />
    <add key="Secret2" value="1c1-1-6f293f35f0610d951f390db96207df5ba80640cfc93ef6a97f955af3a9038845cb4d4ac574899f2460fc7a1f7f1e17a77561c39295b6759df086e744ea4269c17195" />
  </appSettings>
  <system.web>
    <customErrors mode="Off" defaultRedirect="/error/generalerror.aspx" />
    <xhtmlConformance mode="Strict" />
    <webServices>
      <protocols>
        <add name="HttpPost" />
      </protocols>
    </webServices>
    <!-- MachineKey is used to provide a common encryption key for viewStateEncryptionMode & enableViewStateMac to work on server farms.  Without it, each server will generate it's own key, which will not work across mult
```

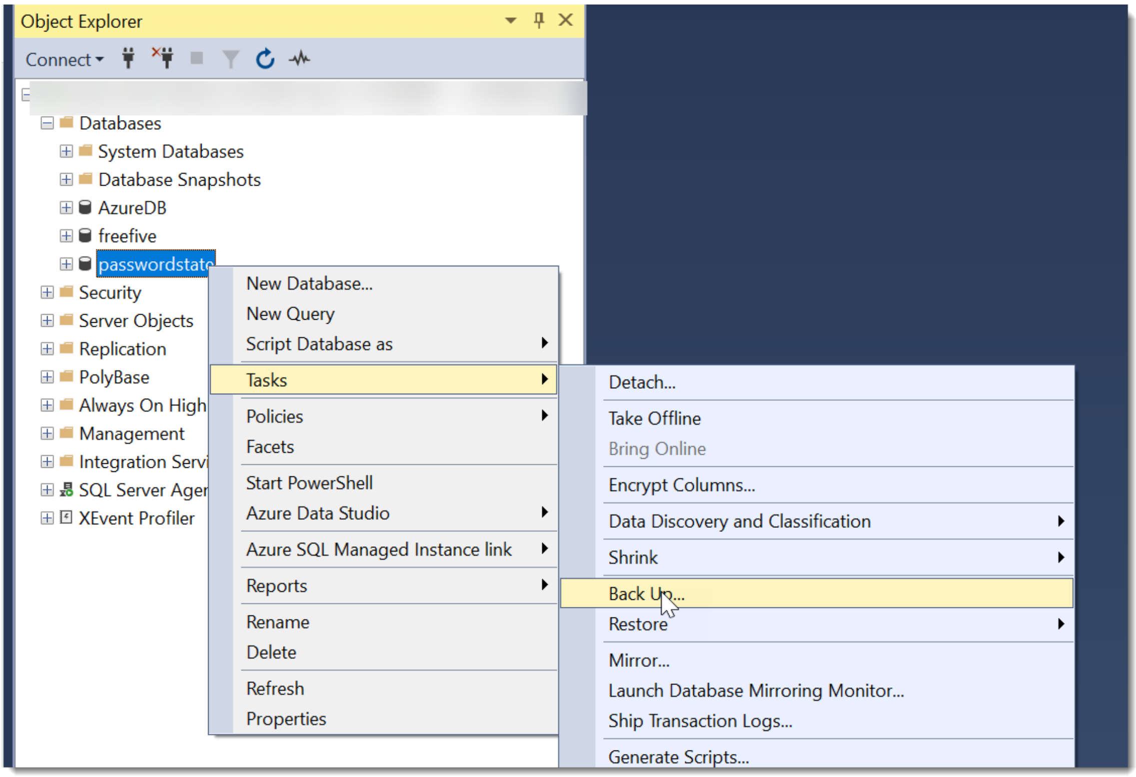
    Note If you have encrypted your database Connection string, or AppSettings section of the web.config file, you will need to decrypt these prior to moving this file. Information about what an encrypted web.config file looks like, and how to decrypt it can be found in this forum post: https://forums.clickstudios.com.au/topic/2699-encrypting-and-decrypting-the-webconfig-file/

10. At this point, *do not log* into this new website, instead follow the next section of this guide which will advise how to migrate your database.

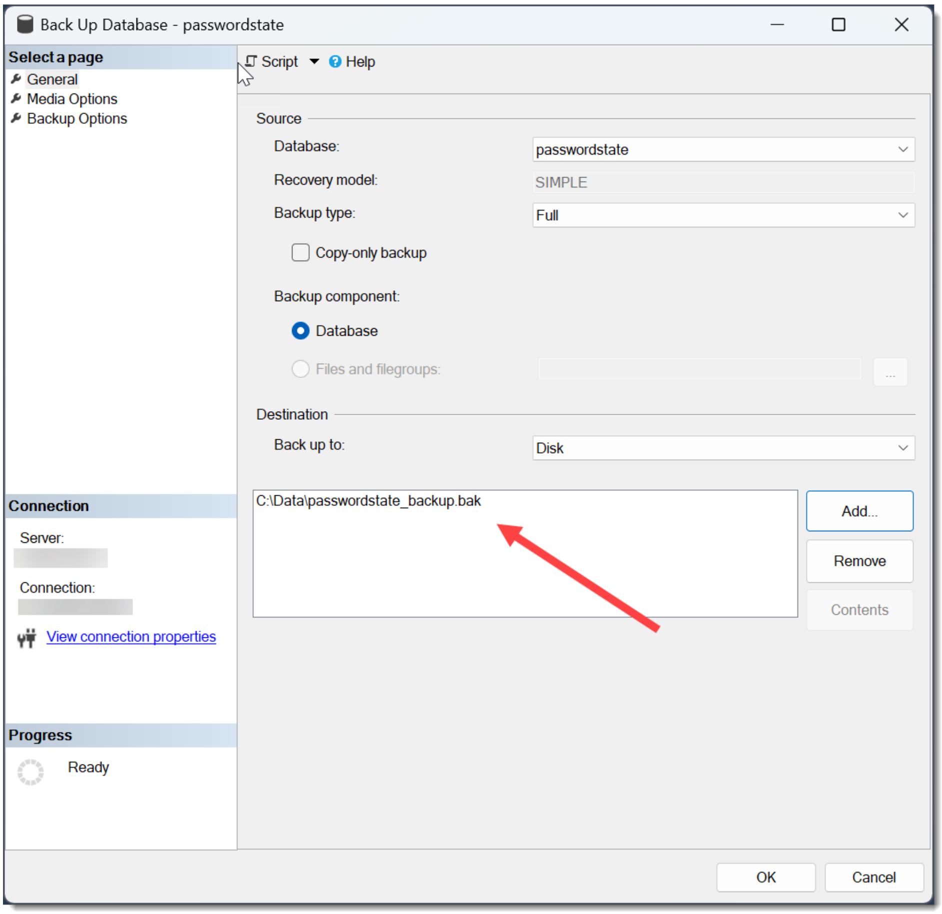
4 Migrating the Database

1. Open **SSMS** (SQL Server Management Studio) and connect to your existing 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. Ensure you append the File name with **.bak**



Source

Database: passwordstate

Recovery model: SIMPLE

Backup type: Full

☐ Copy-only backup

Backup component:

● Database

○ Files and filegroups:

Destination

Back up to: Disk

C:\Data\passwordstate_backup.bak

Connection

Server: [redacted]

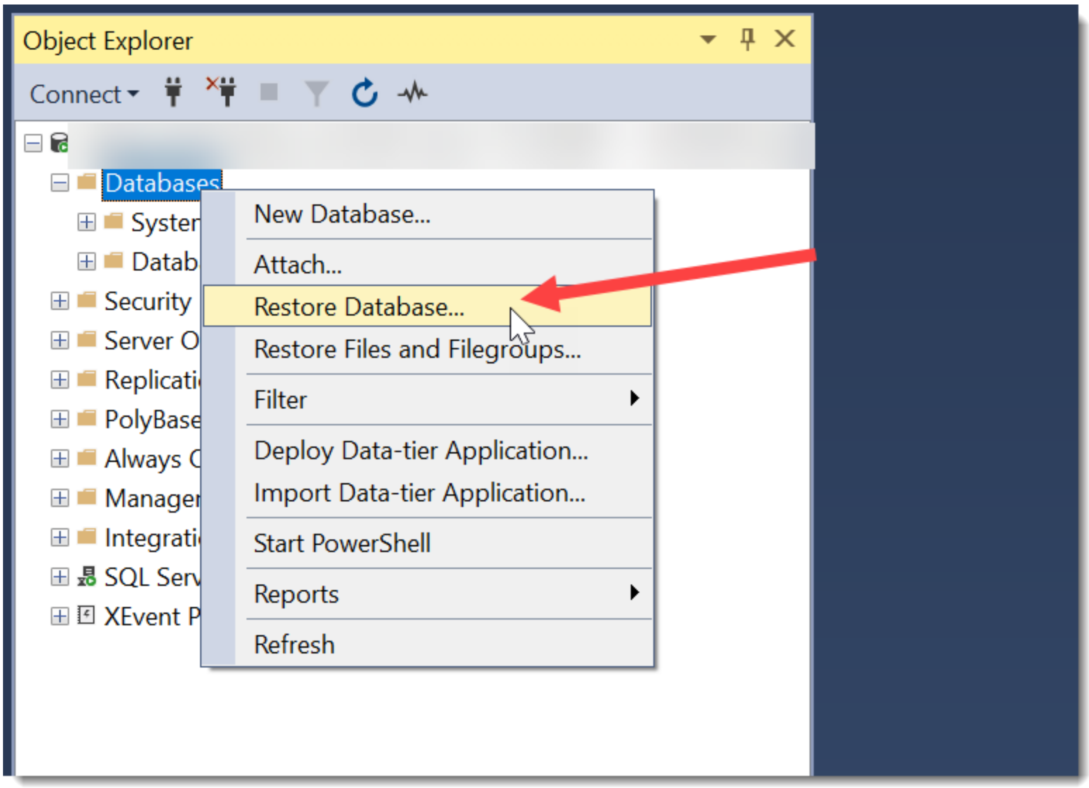
Connection: [redacted]

View connection properties

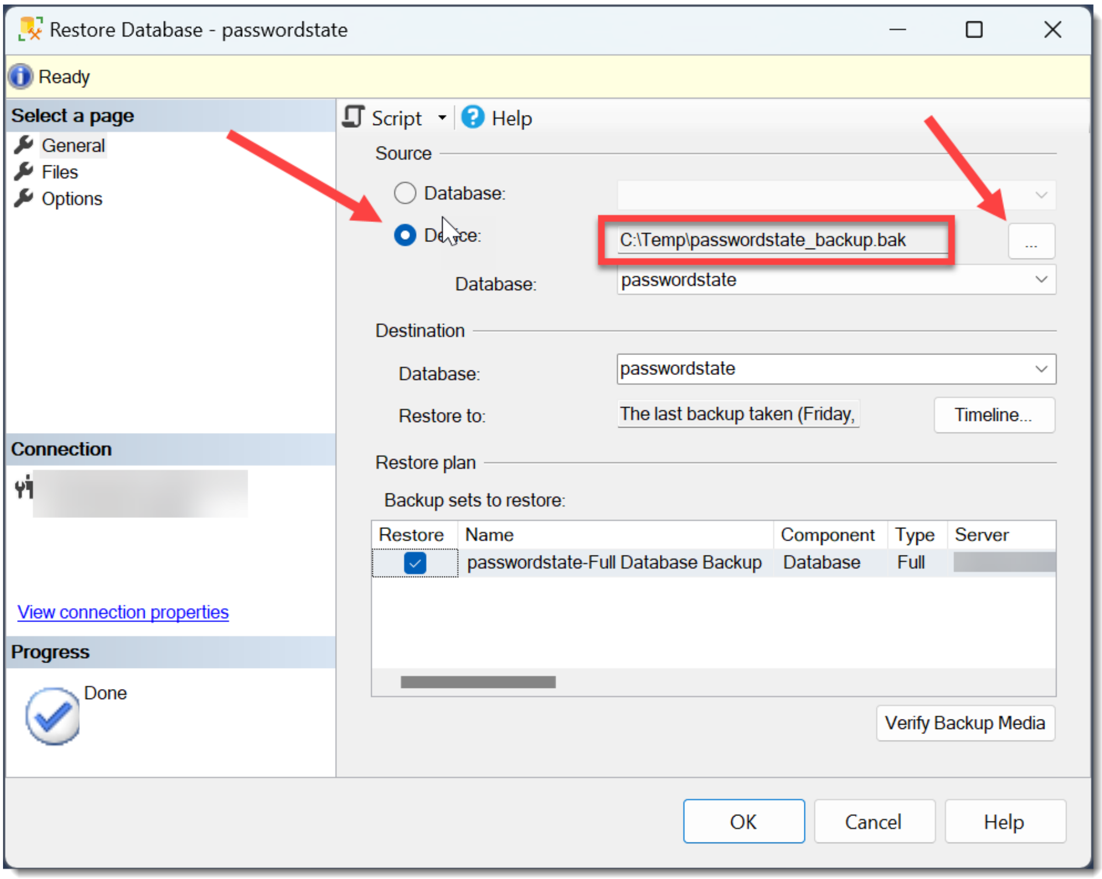
Progress

Ready

4. Transfer this **.bak** file to your new server into a folder of your choice suck as **C:\Data**
5. Using **SSMS**, connect to the new server using an account that has permissions to restore databases.
6. Right click **Databases**, and click **Restore Database…**



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



Source

○ Database:

● Device: C:\Temp\passwordstate_backup.bak ...

Database: passwordstate

Destination

Database: passwordstate

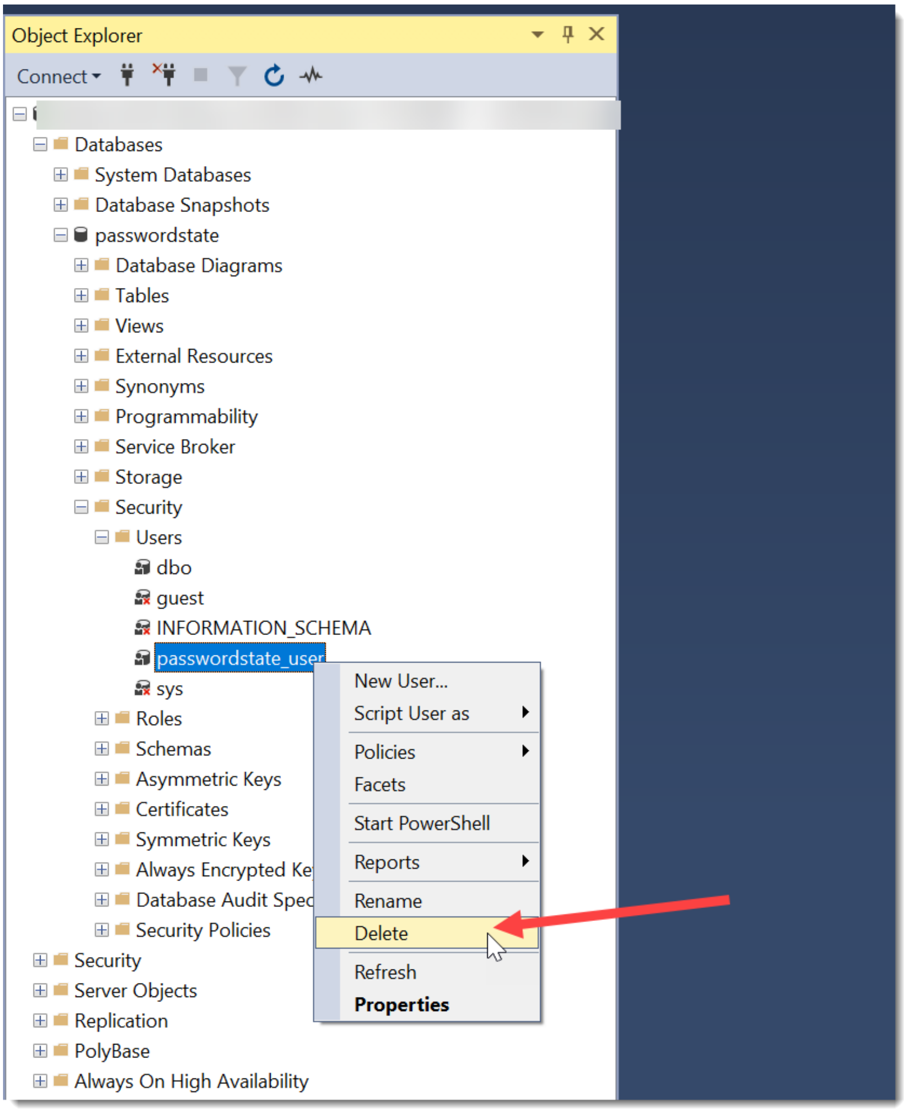
Restore to: The last backup taken (Friday,

Restore plan

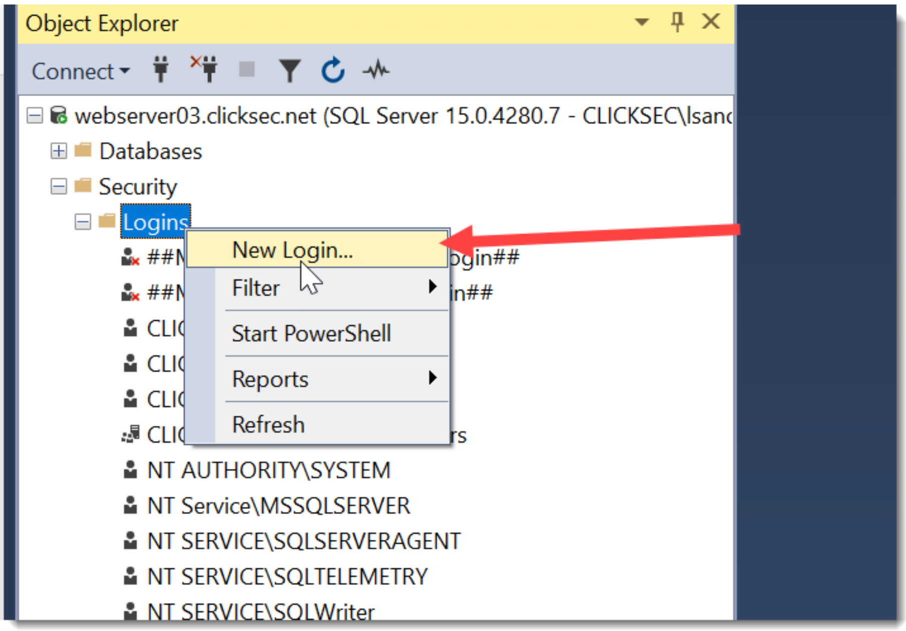
Backup sets to restore:

Restore	Name	Component	Type	Server
☑	passwordstate-Full Database Backup	Database	Full	

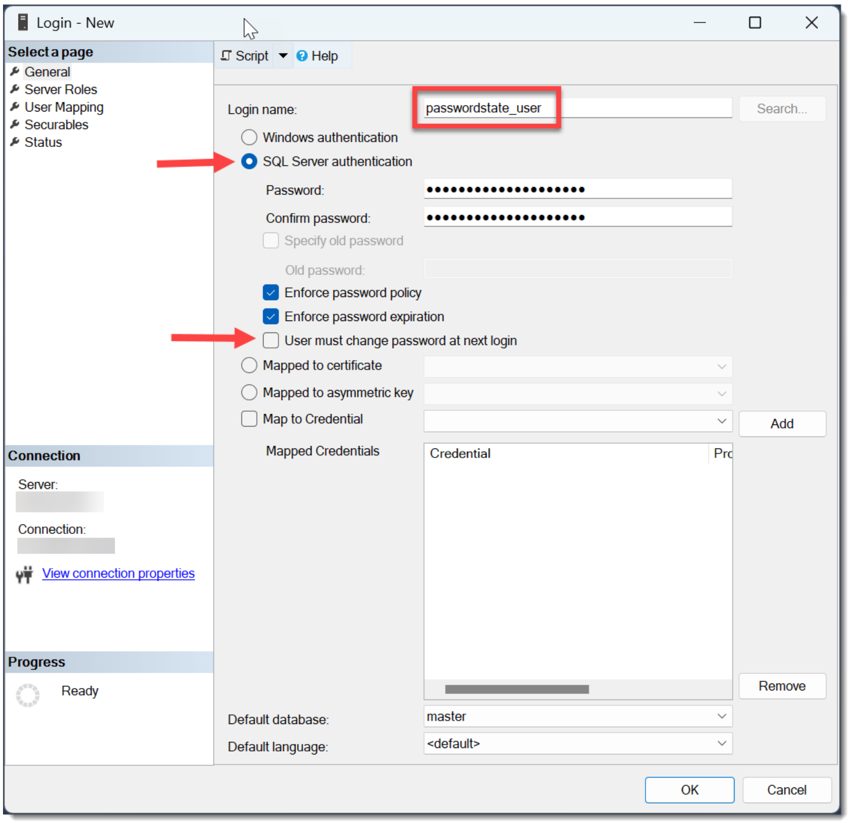
8. You should now see your Passwordstate database under the **Databases** folder. Expand out your Passwordstate database, and then expand out your **Security → Users** folder. Right Click and **Delete** the **passwordstate_user**:



9. Now right click **Security → Logins and** choose **New Login…**



10. 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:



Login name: passwordstate_user

○ Windows authentication

● SQL Server authentication

Password: ••••••••••••••••••••

Confirm password: ••••••••••••••••••••

□ Specify old password

Old password:

☑ Enforce password policy

☑ Enforce password expiration

□ User must change password at next login

○ Mapped to certificate

○ Mapped to asymmetric key

□ Map to Credential

Mapped Credentials

Credential	Pro

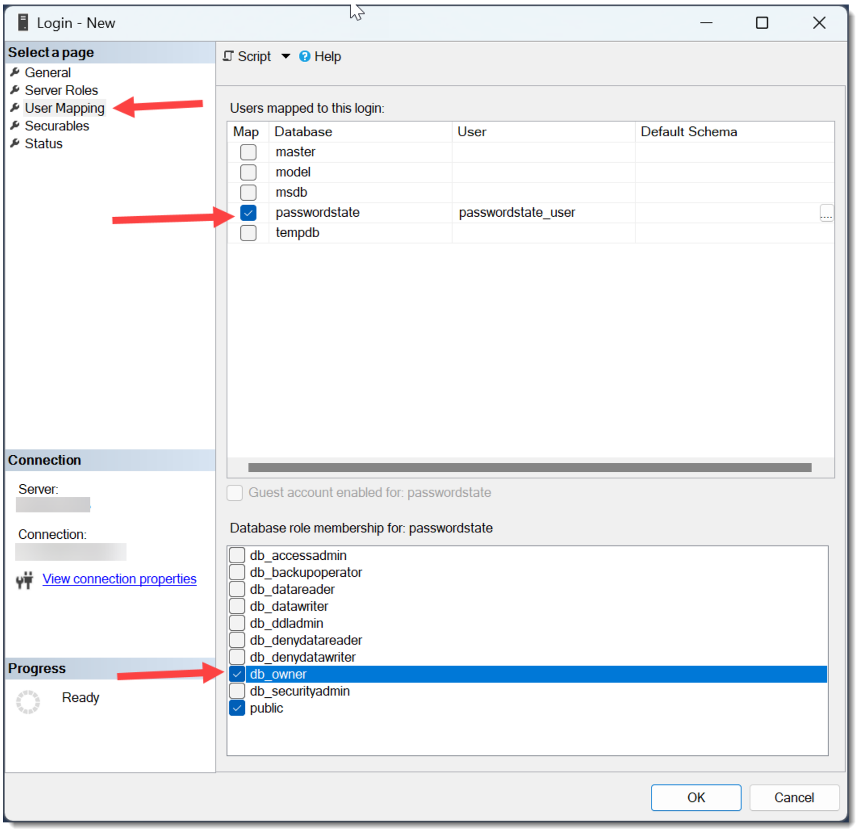
Default database: master

Default language: <default>

Server: [redacted]

Connection: [redacted]

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



Users mapped to this login:

Map	Database	User	Default Schema
□	master		
□	model		
□	msdb		
☑	passwordstate	passwordstate_user	
□	tempdb		

□ Guest account enabled for: passwordstate

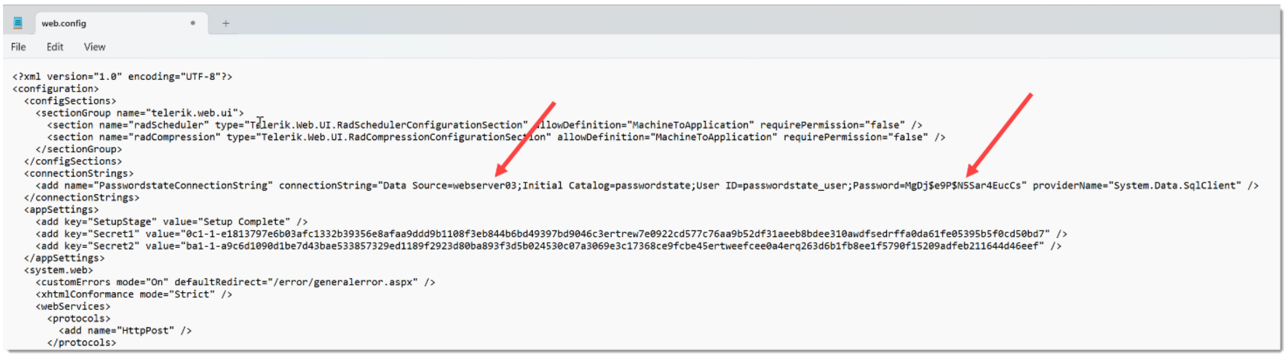
Database role membership for: passwordstate

□ db_accessadmin
□ db_backupoperator
□ db_datareader
□ db_datawriter
□ db_ddladmin
□ db_denydatareader
□ db_denydatawriter
☑ db_owner
□ db_securityadmin
☑ public

Server: [redacted]

Connection: [redacted]

12. Now you need to make some final changes to your web.config file on your new server. Open the file and change the **Data Source** and **Password** in the connection string to be relevant to your new server. The password is what you set in **Item #10** above.



```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <configSections>
    <sectionGroup name="telerik.web.ui">
      <section name="radScheduler" type="Telerik.Web.UI.RadSchedulerConfigurationSection" allowDefinition="MachineToApplication" requirePermission="false" />
      <section name="radCompression" type="Telerik.Web.UI.RadCompressionConfigurationSection" allowDefinition="MachineToApplication" requirePermission="false" />
    </sectionGroup>
  </configSections>
  <connectionStrings>
    <add name="PasswordstateConnectionString" connectionString="Data Source=webserver03;Initial Catalog=passwordstate;User ID=passwordstate_user;Password=MgDj$e9P$N5Sar4EucCs" providerName="System.Data.SqlClient" />
  </connectionStrings>
  <appSettings>
    <add key="SetupStage" value="Setup Complete" />
    <add key="Secret1" value="0c1-1-e1813797e6b03afc1332b39356e8afaa9ddd9b1108f3eb844b6bd49397bd9046c3ertrew7e0922cd577c76aa9b52df31aeeb8bdee310awdfsedrffa0da61fe05395b5f0cd50bd7" />
    <add key="Secret2" value="ba1-1-a9c6d1090d1be7d43bae533857329ed1189f2923d80ba893f3d5b024530c07a3069e3c17368ce9fcbe45ertweefcee0a4erq263d6b1fb8ee1f5790f15209adfeb211644d46eef" />
  </appSettings>
  <system.web>
    <customErrors mode="On" defaultRedirect="/error/generalerror.aspx" />
    <xhtmlConformance mode="Strict" />
    <webServices>
      <protocols>
        <add name="HttpPost" />
      </protocols>
```

13. Restart the **Passwordstate Windows Service** on your new web server (after the Service has been restarted, it will recreate any custom Logos or images you have uploaded into Passwordstate - you may need to give it a minute or two to do this)

14. Now you can attempt to log into Passwordstate for the first time after this migration, and you will most likely be prompted to upgrade the database. This is normal, and you can continue at this point until the upgrade process finishes.

15. If you now log in you should see all your data as it was when on premises. Everything including passwords, users, settings and license keys. If you find the current build is **8995**, you will need to follow **Section 3** in this guide to upgrade to the latest build of **Version 9**: https://www.clickstudios.com.au/downloads/version9/Upgrade_Instructions.pdf

At this point, your Passwordstate instance has now been moved to your new server and your migration is complete. We encourage you to do some user acceptance testing and when complete, you can safely decommission the old server.

This concludes the migration process, and if you have any queries about this, you are more than welcome to log a support call with Click Studio via our support page: https://www.clickstudios.com.au/support.aspx