

**PostgreSQL Upgradation Procedure Documentation**

Manual prepared for:

S2 Global Support and IT Team

Submitted on: **Sep 2024**



|  |  |
| --- | --- |
| S2 Global, Inc.  1800 Eller Drive, Suite 400  Fort Lauderdale, Florida 33316 USA |  |

Information Statement

**Copyright Notice**

Copyright © 2024 S2 Global. All rights reserved. No part of this document may be reproduced, transmitted, transferred, modified or translated, in any form or by any means, without the express written permission of S2 Global.

**Proprietary Information Notification**

This document and the information contained herein proprietary to S2 Global and is protected by all applicable laws throughout the world. Any use, disclosure or duplication of this document or any information contained herein is strictly prohibited, except to the extent expressly agreed to in writing in advance, in each instance, by S2 Global.

**Trademark Notice**

S2 Global, CertScan®, Rapiscan®, AS&E®, Eagle® and other trademarks that may appear throughout this document are owned by S2 Global or its affiliates and are used to distinguish its/their products in the United States and other countries throughout the world. Unauthorized use of any such trademarks is strictly prohibited.

**Content of this Document Statement**

S2 Global has endeavored to ensure that the information in this document is accurate and complete as of the date of creation. Through its provision of this document, S2 Global provides no warranties of any kind, and to the maximum extent permitted by applicable law disclaims all warranties, express, implied or statutory, including, without limitation, any implied warranties of merchantability, fitness for a particular purpose, title, non-infringement or accuracy, and any warranties that may arise from course of dealing or performance or usage of trade.

Table of Contents

[1. General Support Information 5](#_Toc177378301)

[1.1 Overview 5](#_Toc177378302)

[1.2 Intended Audience 5](#_Toc177378303)

[1.3 Pre-requisites 5](#_Toc177378304)

[2. PostgreSQL Version Upgrade in Docker 6](#_Toc177378305)

[2.1 Backup Database 6](#_Toc177378306)

[2.2 Upgrade PostgreSQL & Restore Database 6](#_Toc177378307)

[2.3 Rollback to Previous Version 7](#_Toc177378308)

[3. PostgreSQL Version Upgrade in Kubernetes 9](#_Toc177378309)

[3.1 Backup Database 9](#_Toc177378310)

[3.2 Upgrade PostgreSQL 9](#_Toc177378311)

[3.3 Restore Steps 10](#_Toc177378312)

[3.4 Rollback to Previous Version 12](#_Toc177378313)

[4. PostgreSQL Version Upgrade in AWS RDS 13](#_Toc177378314)

[4.1 Upgrade PostgreSQL 13](#_Toc177378315)

[4.2 How to install extensions in RDS DB Instance 13](#_Toc177378316)

[4.2.1 Create DB Parameter Group 13](#_Toc177378317)

[4.2.2 Modify DB Parameter Group Configurations 13](#_Toc177378318)

[4.2.3 Associate DB Parameter Group to a DB Instance 14](#_Toc177378319)

[4.2.4 Verify Extension from PG Admin 14](#_Toc177378320)

Revision Record

| Rev. | Description | Author | Date |
| --- | --- | --- | --- |
| 1.0 | Draft Release | Maruthi | 16th Sep 2024 |
|  |  |  |  |
|  |  |  |  |

1. General Support Information
   1. Overview

This document briefly describes the procedure of upgrading PostgreSQL v16 in different environments such as Docker, Kubernetes, and Docker with AWS RDS.

**IMPORTANT:** This document is for internal use only and not for external use**.**

* 1. Intended Audience

This document is intended especially for the internal user(s) of **S2 Global Support** and **IT Team** to upgrade PostgreSQL v16 in various environments.

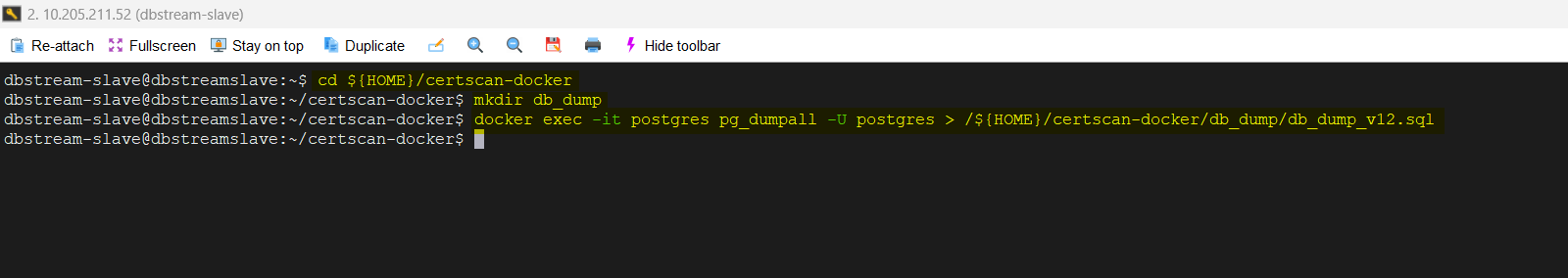
* 1. Pre-requisites

Before performing the PostgreSQL upgradation procedure, ensure the following pre-requisites are met:

* Download the Deployment Bundle to Server
  + Before copying the latest release bundle to your server, rename your existing “**db-init-compose.yml**” file as “**db-init-compose\_12.yml**”. (applicable only for Docker environment)
  + Before copying the latest release bundle to your server, rename your existing release bundle folder as **“certscan-kubernetes-kustomize\_backup”** (applicable for Kubernetes environment)
* Install AWS CLI and Configure
* Login to AWS ECR
* Down all the Application services except the database

**NOTE:** All the commands in this document are represented in **‘Courier New’** font. If any of the commands doesn’t work as expected (i.e., as shown in the screenshots), then please add **sudo** as a prefix to the command and execute them again.

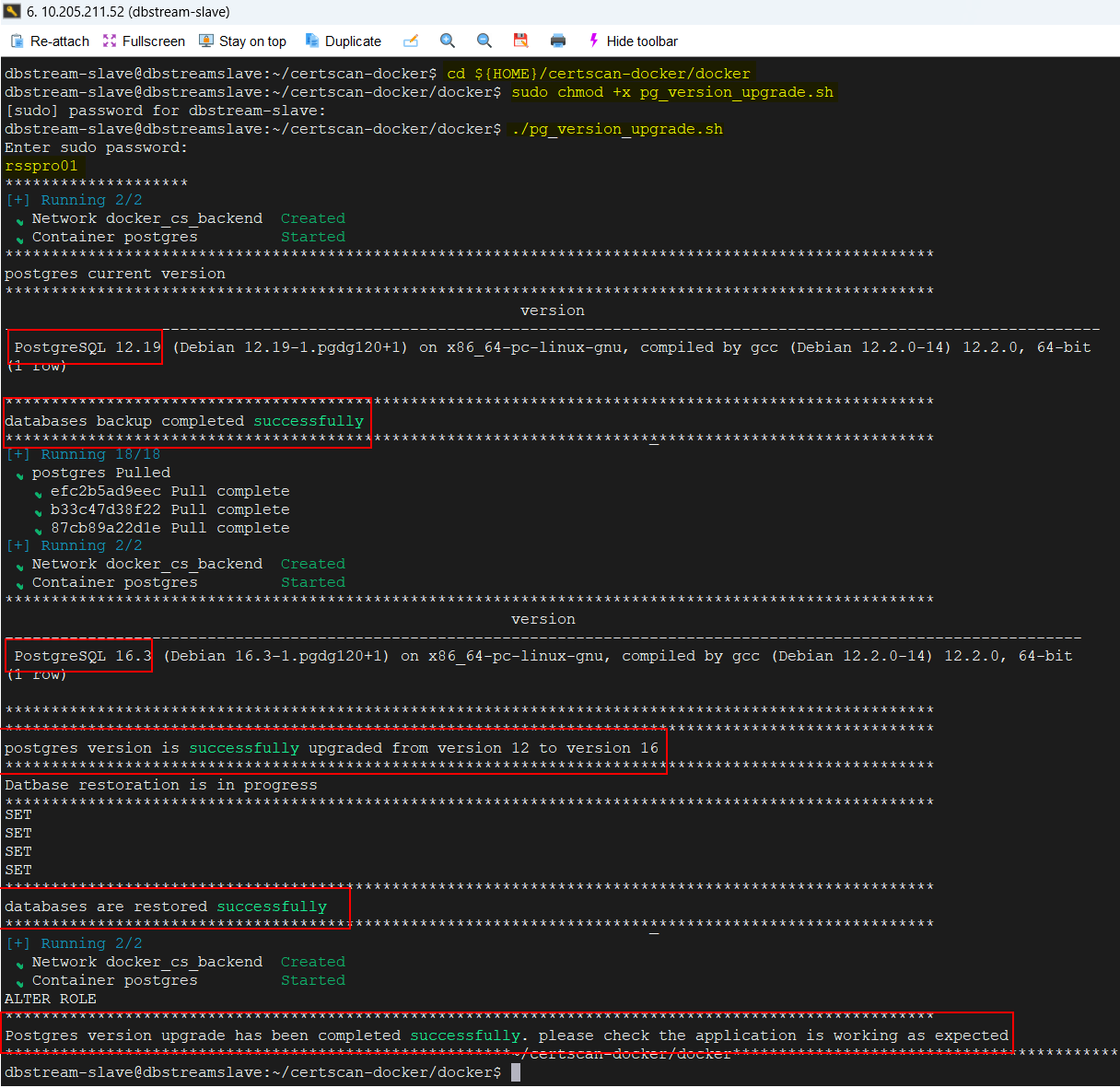
1. PostgreSQL Version Upgrade in Docker
   1. Backup Database
2. Run **cd ${HOME}/certscan-docker** command to navigate to the certscan-docker folder.
3. Run **mkdir db\_dump** command to create a directory named “db\_dump”.
4. Run **docker exec -it postgres pg\_dumpall -U postgres > /${HOME}/certscan-docker/db\_dump/db\_dump\_v12.sql** command to take the backup of current running database. It is used for restoration after upgrading the PostgreSQL.



* 1. Upgrade PostgreSQL & Restore Database

Steps to be followed to upgrade PostgreSQL v16 and restore the database backup:

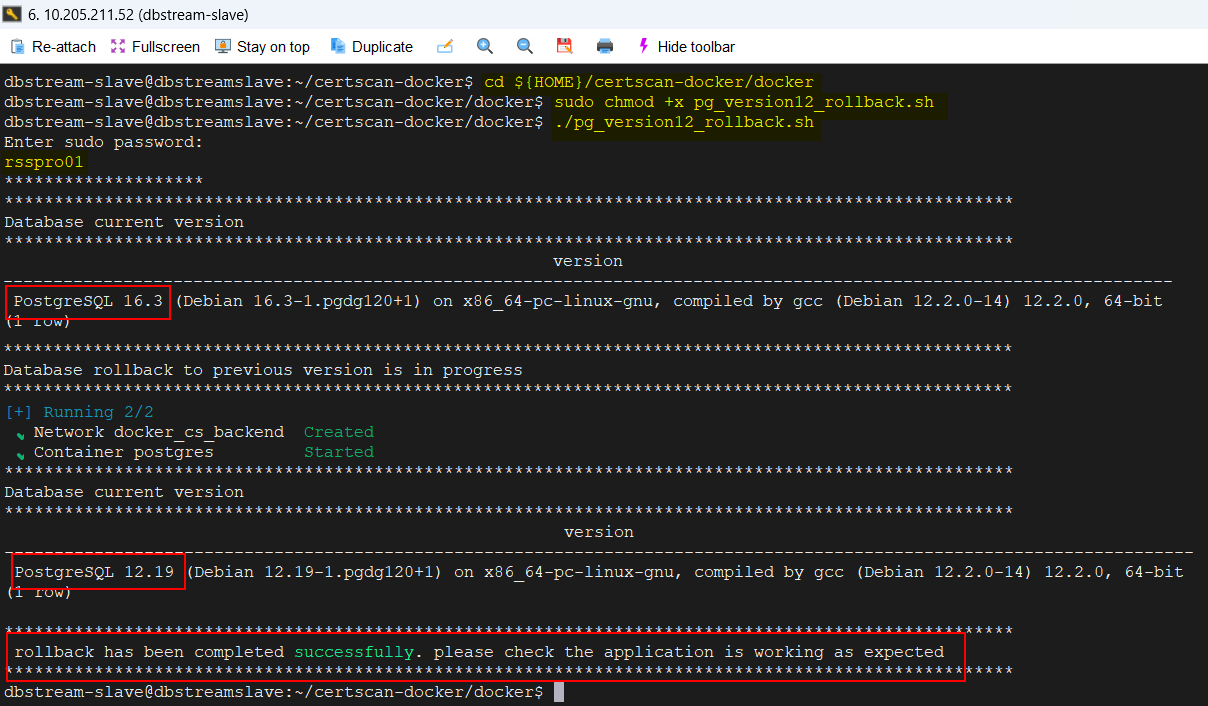
1. Run **cd ${HOME}/certscan-docker/docker** commandto navigate to the docker folder.
2. Run **sudo chmod +x pg\_version\_upgrade.sh** command to provide execution permissions to the shell script. Provide the root user password associated with the username whenever required.
3. Run **./pg\_version\_upgrade.sh** command to execute the shell script. Enter the root user password associated with the username to proceed further with upgrading the PostgreSQL from v12 to v16 and restoring the database backup.



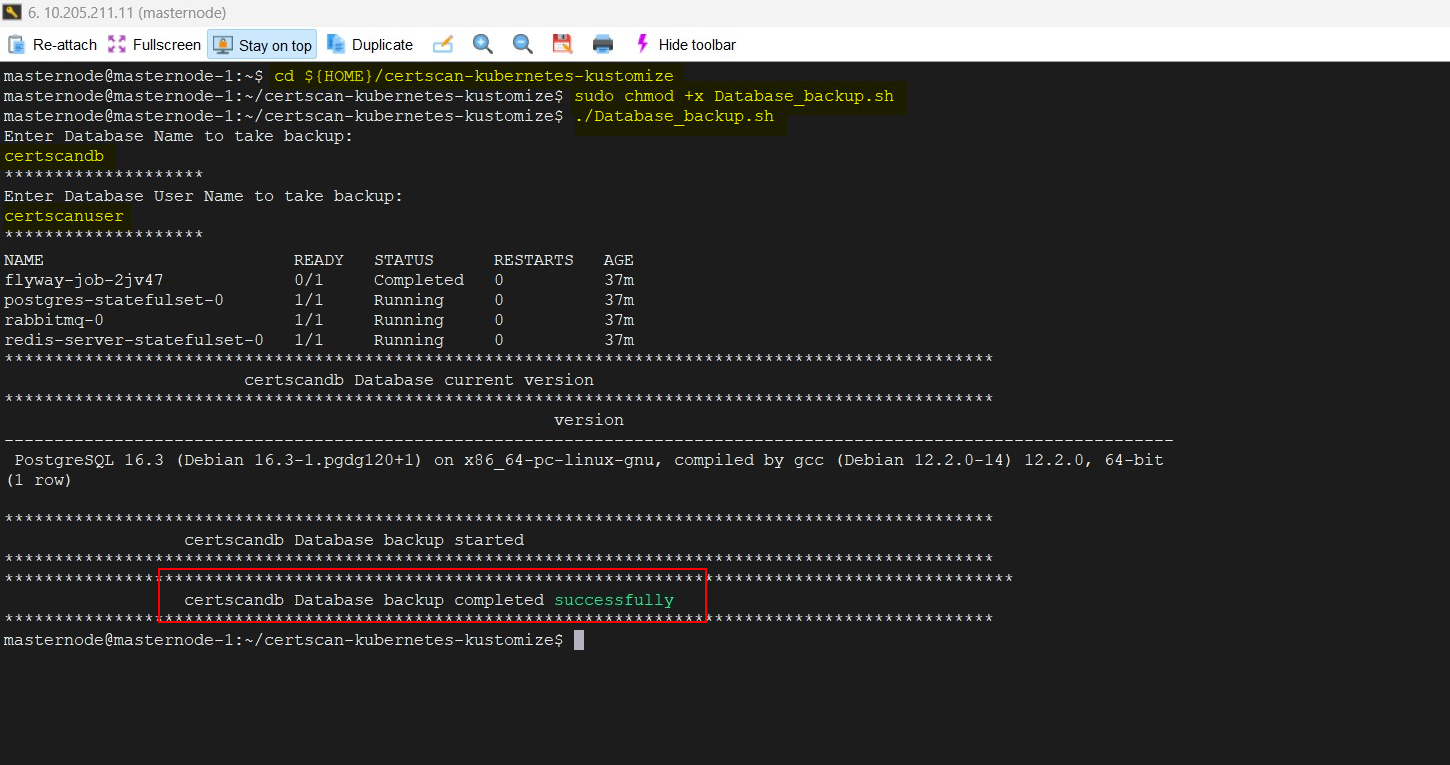
* 1. Rollback to Previous Version

In case the database upgrade is unsuccessful, you can rollback to the previous version as we have taken the backup files from the previous version.

1. Run **cd ${HOME}/certscan-docker/docker** commandto navigate to the docker folder.
2. Run **sudo chmod +x pg\_version12\_rollback.sh** command to provide execution permissions to the shell script. Provide the root user password associated with the username whenever required.
3. Run **./pg\_version12\_rollback.sh** command to execute the shell script. Enter the sudo password associated with the username to proceed further to rollback to the previous PostgreSQL version.



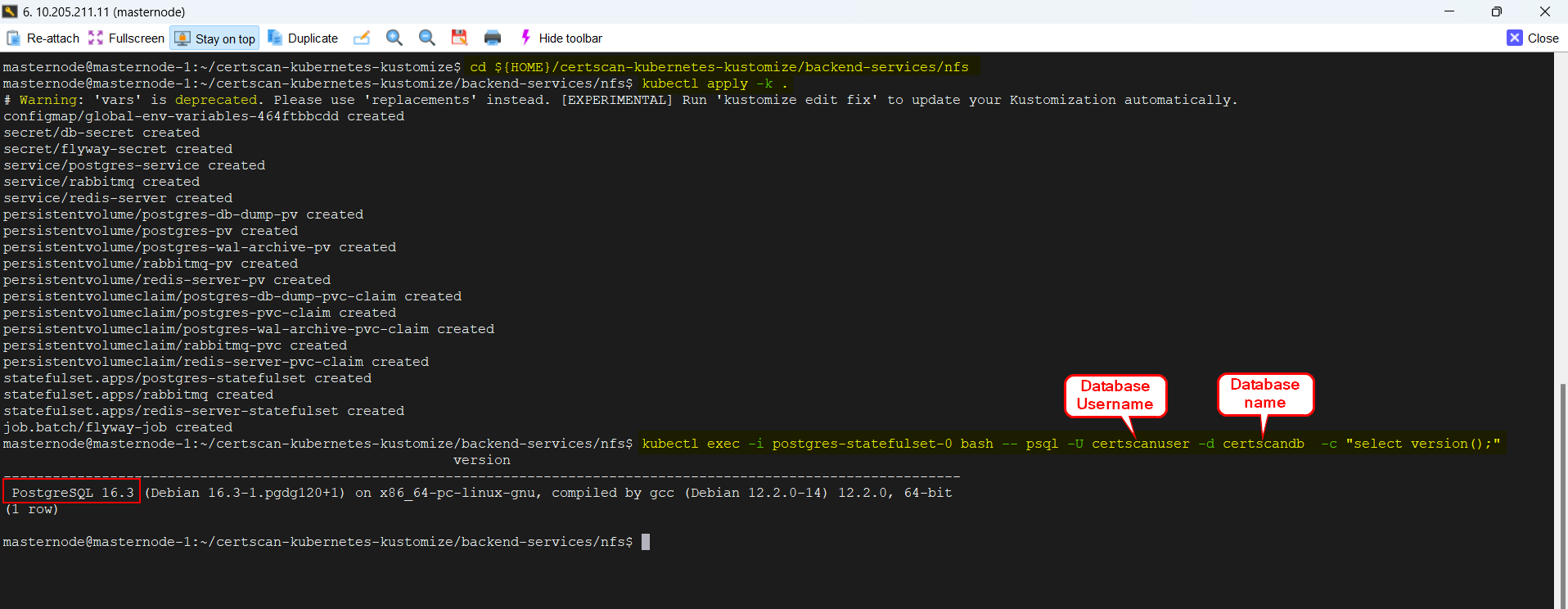
1. PostgreSQL Version Upgrade in Kubernetes
   1. Backup Database
2. Run **cd ${HOME}/**c**ertscan-kubernetes-kustomize** command to navigate to the deployment folder.
3. Run **sudo chmod +x Database\_backup.sh** command to provide execution permissions to the shell script.Note that it is mandatory to provide the password associated with the username whenever you run any **sudo** command.
4. Run **./Database\_backup.sh** command to execute the shell script. Provide the following necessary details to backup the selected database. This backup file can be used for restoration purpose after upgrading the PostgreSQL.
   1. Database Name
   2. Database Username



* 1. Upgrade PostgreSQL

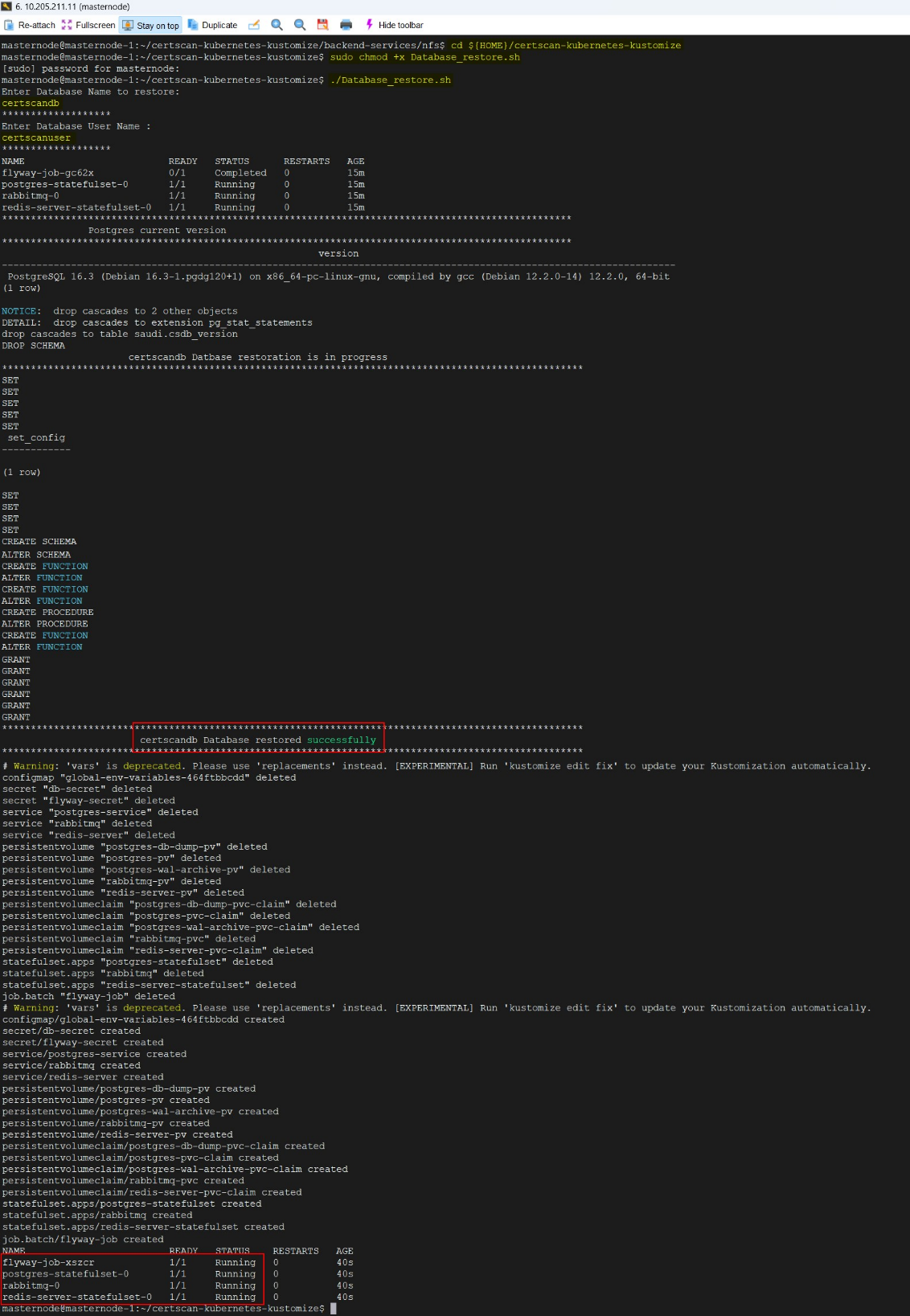
Steps to be followed to upgrade PostgreSQL v16:

1. Run **cd ${HOME}/**c**ertscan-kubernetes-kustomize/backend-services/<storage volume>** command to navigate to the backend-services folder.   
   **NOTES:**
   * Make sure to update your respective storage volume (NFS / Tanzu / AWS) in the above command before running it.
   * As of now, we have validated the PostgreSQL upgrade process only on the NFS storage volume.
2. Run **kubectl apply -k .** command to execute the backend services necessary for deploying CertScan application alongside the upgradation of the PostgreSQL version from v12 to v16.
3. Run **kubectl exec -i postgres-statefulset-0 bash -- psql -U <Databaseusername> -d <Databasename> -c "select version();"** command to verify whether the PostgreSQL version is upgraded to 16.   
   **NOTE:** Make sure to update your respective Database Username and Database Name in the above command before running it.



* 1. Restore Steps

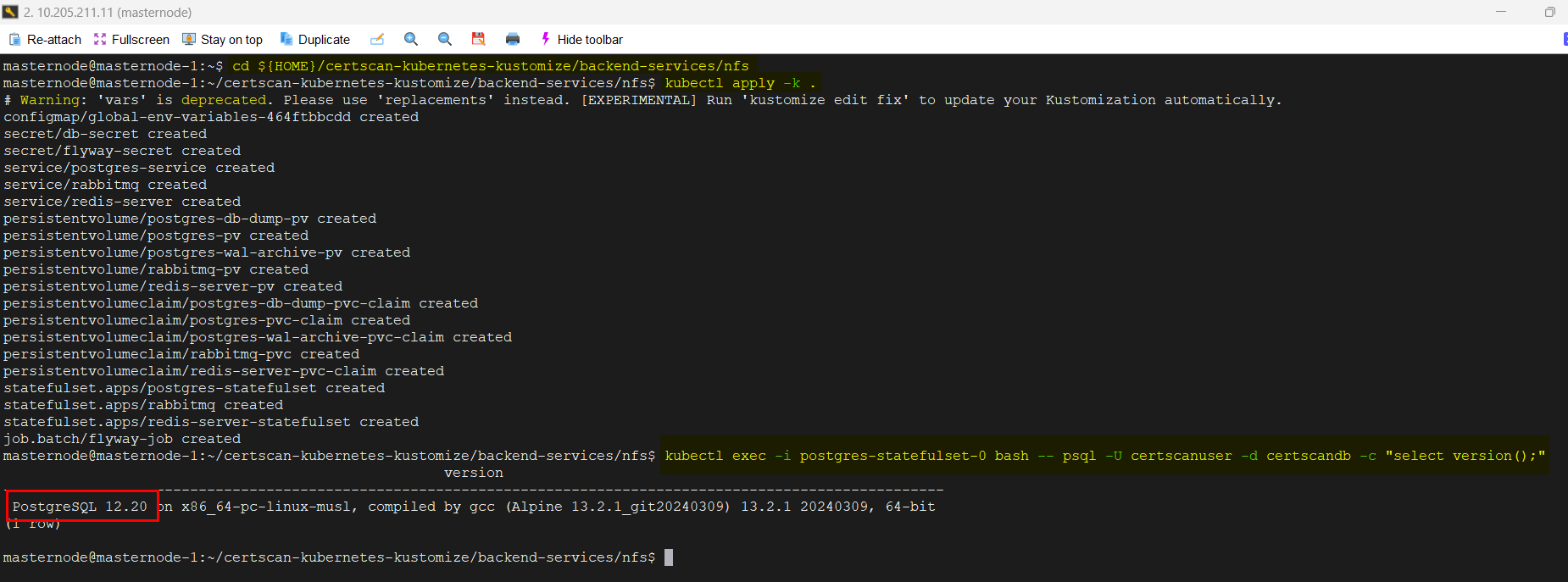
1. Run **cd ${HOME}/**c**ertscan-kubernetes-kustomize** command to navigate to the deployment folder.
2. Run **sudo chmod +x Database\_restore.sh** command to provide execution permissions to the shell script. Note that it is mandatory to provide the password associated with the username whenever you run any **sudo** command.
3. Run **./Database\_restore.sh** command to execute the shell script. Provide the following necessary details to restore the database backup.
   1. Database Name
   2. Database Username



* 1. Rollback to Previous Version

In case the database upgrade is unsuccessful, you can rollback to the previous version as we have taken the backup of the previous release bundle.

1. Rename the latest release bundle folder as **“certscan-kubernetes-kustomize-new”** and the previous release bundle folder as **“certscan-kubernetes-kustomize**”.
2. Repeat the same steps (Step 1 to Step 3) listed in [**Section 3.2**](#UpgradePostgreSQL) to rollback to the previous PostgreSQL version.



1. PostgreSQL Version Upgrade in AWS RDS
   1. Upgrade PostgreSQL

Refer to the below link to find the PostgreSQL version upgrade procedure.

[Performing a major version upgrade - Amazon Aurora](https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/USER_UpgradeDBInstance.PostgreSQL.MajorVersion.html#USER_UpgradeDBInstance.Upgrading.Manual)

* 1. How to install extensions in RDS DB Instance
     1. Create DB Parameter Group

1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Parameter groups**.
3. Click on **Create parameter group** buttonunderCustom tab.
4. Provide the following required details for creating a new group.
   1. *Parameter group name* ***-*** Enter the name of your new DB parameter group.
   2. *Description -* Enter a description for your new DB parameter group.
   3. *Engine type -* Choose your DB engine (Aurora PostgreSQL).
   4. *Parameter group family -* Choose a DB parameter group family (aurora-postgresql-16).
   5. *Type -*If applicable, choose **DB Parameter Group**.
5. Click on **Create** button to create a new group with the provided details.



* + 1. Modify DB Parameter Group Configurations

The RDS console shows the status of the DB parameter group associated with a DB instance on the **Configuration** tab. For example, suppose that the DB instance isn't using the latest changes to its associated DB parameter group. If so, the RDS console shows the DB parameter group with a status of **pending-reboot**. To apply the latest parameter changes to that DB instance, manually reboot the DB instance.

1. In the navigation pane, choose **Parameter Groups**.
2. From the groups list, select the newly created parameter group.
3. Go to **Parameter group 🡪 Actions** and choose **Edit**.
4. Search for the **shared\_preload\_libraries** and add **pg\_cron** to its value.
5. Search for the **cron.database\_name** and update your certscan database name (certscandb) as its value.
6. Click on the **Save Changes** button to save your changes for the selected parameter group.
7. After the modifications, select a modified DB instance and choose **Actions 🡪 Reboot**.



* + 1. Associate DB Parameter Group to a DB Instance

1. In the navigation pane, choose **Databases**.
2. Select a DB instance that you want to modify and click on **Modify** button.
3. Go to **Additional Configuration 🡪 Database Options** and select the newly created DB parameter group.
4. Click on **Continue** button and check the summary of modifications.
5. Choose **Apply immediately** to apply the changes immediately.
6. On the confirmation page, review your changes and click on **Modify DB instance** button to save your changes.
7. After the modifications, select the DB instance and choose **Actions 🡪 Reboot**.



* + 1. Verify Extension from PG Admin

1. Connect to the database from pg\_Admin.
2. Run **select \* from pg\_available\_extensions;** queryto verify whether the pg\_cron is available in the query results.

