|  |  |  |  |
| --- | --- | --- | --- |
| **Enterprise Business Process** | Optum Connect | **Last Updated Date** | 04/18/2023 |
| **Revision #** | 1.0 | **Process Owner** | Charles Colstrom |
| **Written/Revised By** | Tejinder Singh | | |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **The following are approved changes incorporated into the revision numbers indicated below** | | | |
| **Version No.** | **Date** | **Description of Change** | **Modified By** |
| 1.0 | 04/18/2022 | *Initial Draft* | Tejinder Singh |

Table of Contents

[1. Purpose 3](#_Toc132722988)

[2. Scope 3](#_Toc132722989)

[3. Roles & Responsibilities 3](#_Toc132722990)

[Role 3](#_Toc132722991)

[Responsibilities 3](#_Toc132722992)

[4. Prerequisites 3](#_Toc132722993)

[5. Backup 4](#_Toc132722994)

[6. Release Upgrade 5](#_Toc132722995)

[7. Glossary 8](#_Toc132722996)

[Figure 1 App-config export 4](#_Toc132670507)

[Figure 2 Parameter Export 5](#_Toc132670508)

[Figure 3 App release fetch 5](#_Toc132670509)

[Figure 4 List tasks 6](#_Toc132670510)

[Figure 5 App status 7](#_Toc132670511)

[Figure 6 Friendly name version 7](#_Toc132670512)

# Purpose

The purpose of this document is to serve as a guide for the process of Terraform Enterprise release upgrade.

# Scope

The Scope of this document is limited to upgrade procedurals for the Terraform Enterprise application.

# Roles & Responsibilities

Refer to the following table for OCC Engineering Team responsibilities relating to the process.

|  |  |
| --- | --- |
| Role | Responsibilities |
| OCC Engineering Team | Develop and maintain |

# Prerequisites

* Access to the Terraform Enterprise (TFE) Virtual Machine Scale-Set (VMSS) instance
* Export Terraform Enterprise Configuration

Prior to the upgrade process, it is strongly recommended to export existing application configurations for reference and in case they are needed for an emergency situation. The steps below are required to export the application configuration:

* 1. Navigate into Terraform Enterprise host machine via Bastion Host
  2. Execute command **replicatedctl app-config export --hidden**
  3. Save the JSON output into a file and store in a safe location
  4. Ensure there is a backup of **/etc/replicated.conf** and the required TLS certificates and license file.
* Review the general [Terraform Enterprise requirements](https://developer.hashicorp.com/terraform/enterprise/requirements/credentials), including credentials, hardware, operating system, data storage, and network
* Review the upgrade requirements and release notes for the specific [Terraform Enterprise release](https://developer.hashicorp.com/terraform/enterprise/releases)

# Backup

It is imperative to backup the data of the current Terraform Enterprise installation to avoid unexpected failures which may cause system outage or, in the worst-case scenario, data loss. The backup and restore API back-up all of the data stored in a Terraform Enterprise installation, including both the blob storage and the PostgreSQL database. It does not back up the installation configuration. This backup can then be restored to a new installation of Terraform Enterprise.

Please note the following when using the backup and restore API:

* The version of Terraform Enterprise cannot be changed between a backup and restore. That is, a backup taken from one version of Terraform Enterprise cannot be restored to an installation running a different version of Terraform Enterprise
* The version of PostgreSQL being used cannot be changed between a backup and restore. That is, a backup taken from a Terraform Enterprise installation using one version of PostgreSQL cannot be restored to an installation using a different version of PostgreSQL
* The Terraform Enterprise installation that will be restored to must be a new, running installation with no existing application data
* Once a restore is completed, the Terraform Enterprise application will need to be restarted before it can use the restored data

The backup API uses an authentication token which can be generated using the following commands on the VMSS instance

**replicatedctl app-config export | jq '.backup\_token'**

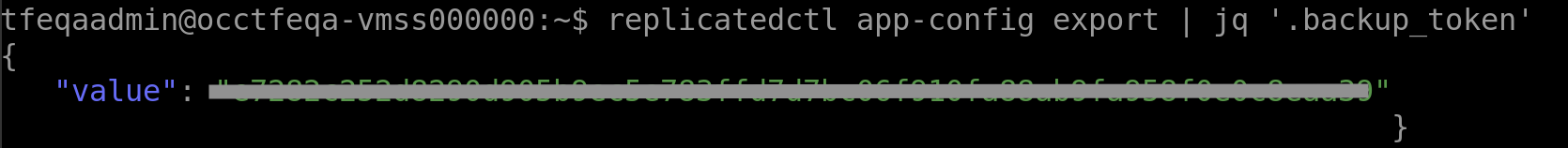


Figure 1 App-config export

To initiate a backup, make a POST request to the backup endpoint on a running Terraform Enterprise installation as shown below

**curl --location --request POST '**[**https://tfe.optum.com/\_backup/api/v1/backup**](https://tfe.optum.com/_backup/api/v1/backup)**' \**

**--header 'Authorization: Bearer <backup\_token>' \**

**--header 'Content-Type: application/json' \**

**--output backup.blob \**

**--data-raw '{"password": "password\_to\_encrypt\_backup"}'**

No manual backup is required to be taken for the Azure infrastructure components

* Storage Account
* PostgreSQL Database
* Redis Cache

# Release Upgrade

Once the pre-requisites and backup tasks have been successfully completed, refer to the steps given below to upgrade the Terraform Enterprise release version

1. Connect to the Terraform Enterprise VMSS instance using the Bastion Host
2. Post a successful login, verify the current release sequence number for the TFE application by executing the below command on the VMSS instance:

**replicatedctl params export | jq '.ReleaseSequence'**

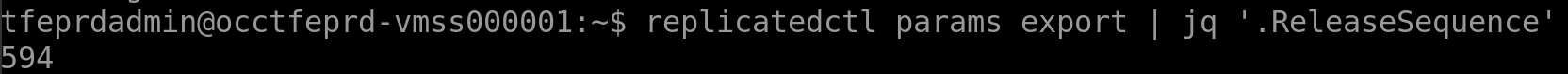


Figure 2 Parameter Export

1. Once the current release has been identified, execute the following command on the VMSS instance to fetch the available release versions

**replicatedctl app-release ls --fetch**

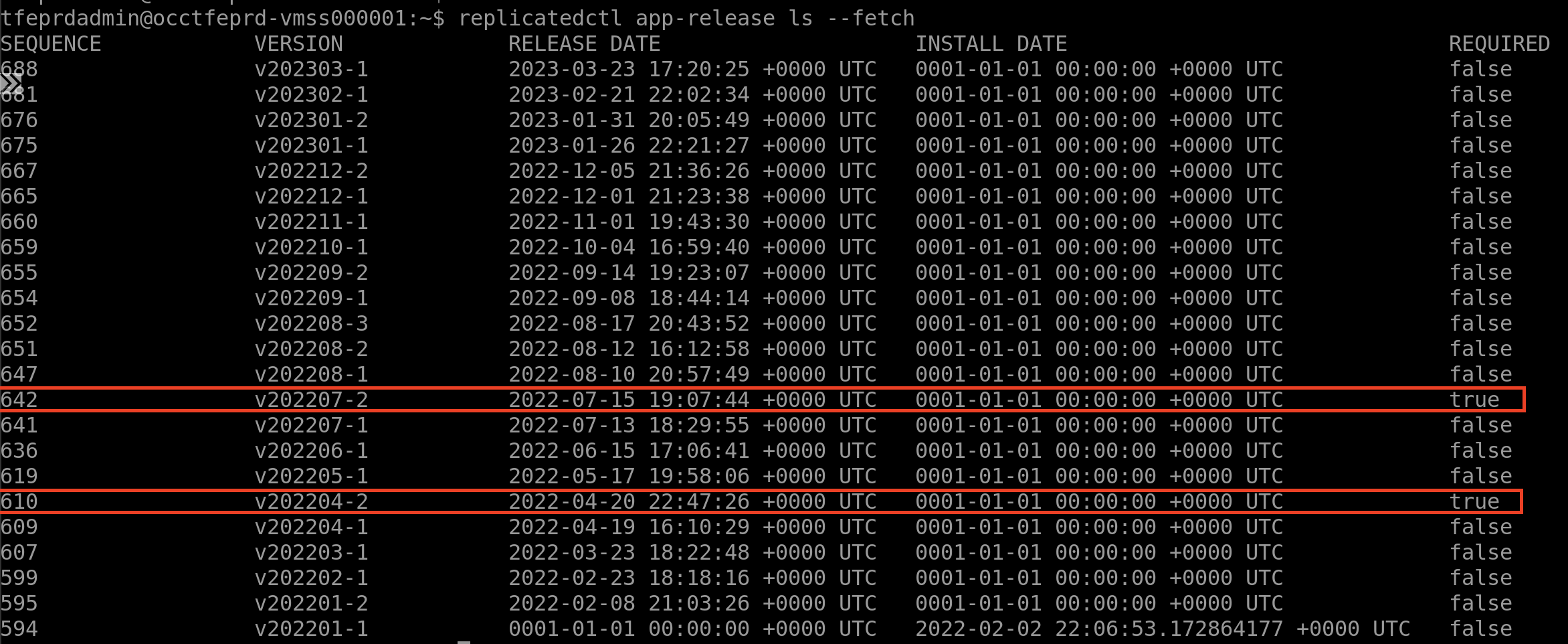


Figure 3 App release fetch

Upgrading from a much older release of Terraform Enterprise than the target release may require a stepped upgrade. This will involve upgrading to the release sequence versions which have the required parameter marked as true.

As shown in Figure 3 to upgrade from a release sequence of 594 to 688, a stepped release upgrade is required, where the instance will be first upgraded to release sequence number 610, followed by 642, and finally to the release sequence 688 to complete the upgrade process.

1. Set the new release sequence number up to the desired version using the below command on the VMSS instance

**replicatedctl params set ReleaseSequence --value 610**

1. Once the release sequence number has been set, revalidate the same by executing the below command once again:

**replicatedctl params export | jq '.ReleaseSequence'**

The output of the execution should point to the desirable version of the release sequence.

1. Execute the following command, to initiate the release upgrade of the TFE instance:

**replicatedctl app-release apply --sequence "610"**

1. Validate the status of the upgrade activity using the below command:

**replicatedctl task ls**

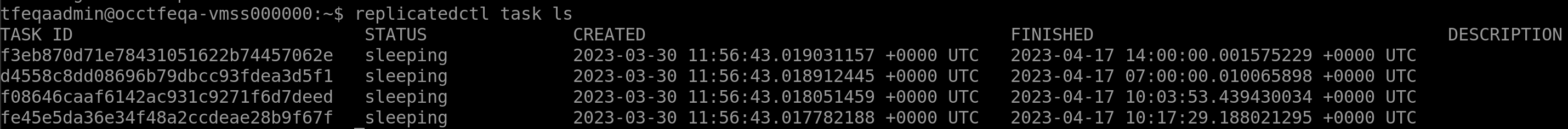


Figure 4 List tasks

The status of the tasks should be “completed”, if the status is other than “completed” wait for the process to complete and re-execute the above command to validate the completion.

1. On successful completion of the tasks, execute the below command to identify the app status

**replicatedctl app status**

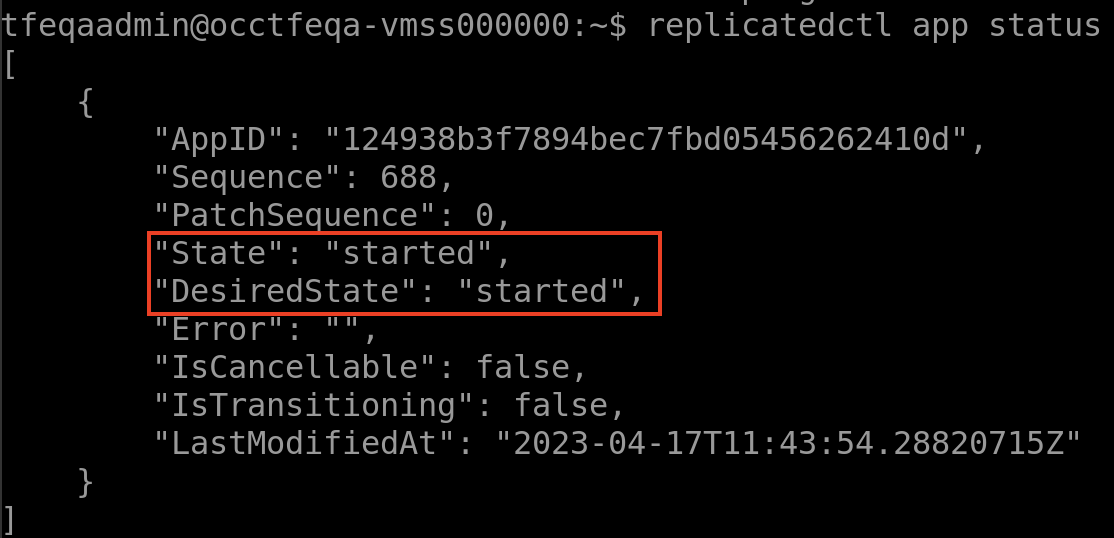


Figure 5 App status

Make sure the “**State**” field in the above output should have the value as “**started**”, if the value is “**stopped**”, execute the below command to start the TFE instance

**replicatedctl app start**

and, revalidate the app status with the following command

**replicatedctl app status**

1. Head to the TFE instance homepage by visiting the URL “https://tfe.optum.com/” on a browser and validate the release sequence number at the bottom of the webpage.

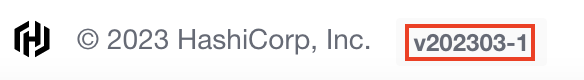


Figure 6 Friendly name version

The release sequence number should point to the friendly name version with the format “**vYYYYMM-X**” where **YYYY** denotes the year, **MM** denotes month and **X** refers to the release count.

# Glossary

Refer to the following acronyms and definitions.

|  |  |
| --- | --- |
| **Term/Acronym** | **Definition** |
| OCC | Optum Connect Cloud |
| API | Application Programming Interface |
| JSON | JavaScript Object Notation |
| TFE | Terraform Enterprise |
| VMSS | Virtual Machine Scale-Set |