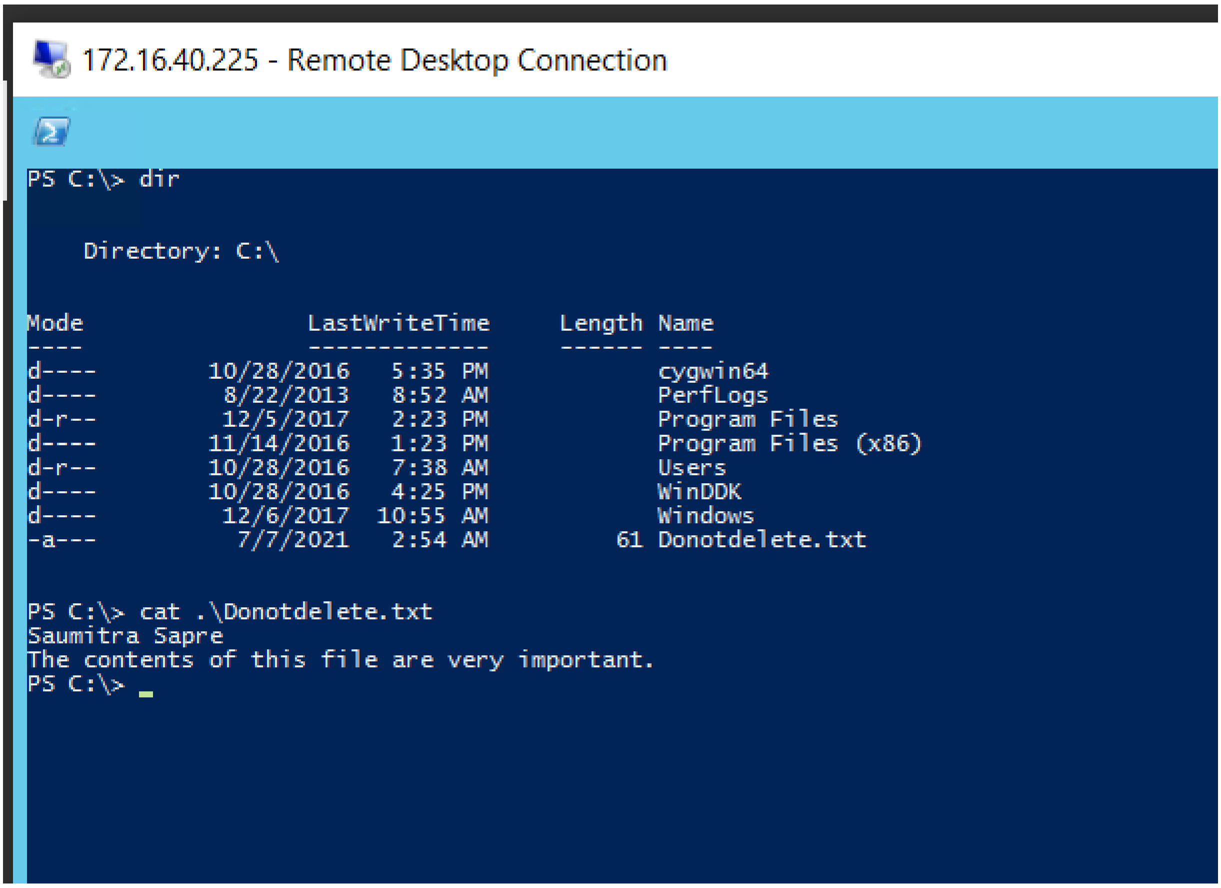
Performing VSS

Before we actually create a shadow copy of a drive and back up data to it, we must add a file to our drive in order to verify whether the restore has worked or not.

So we create a file in our drive and fill it with text in order to verify.



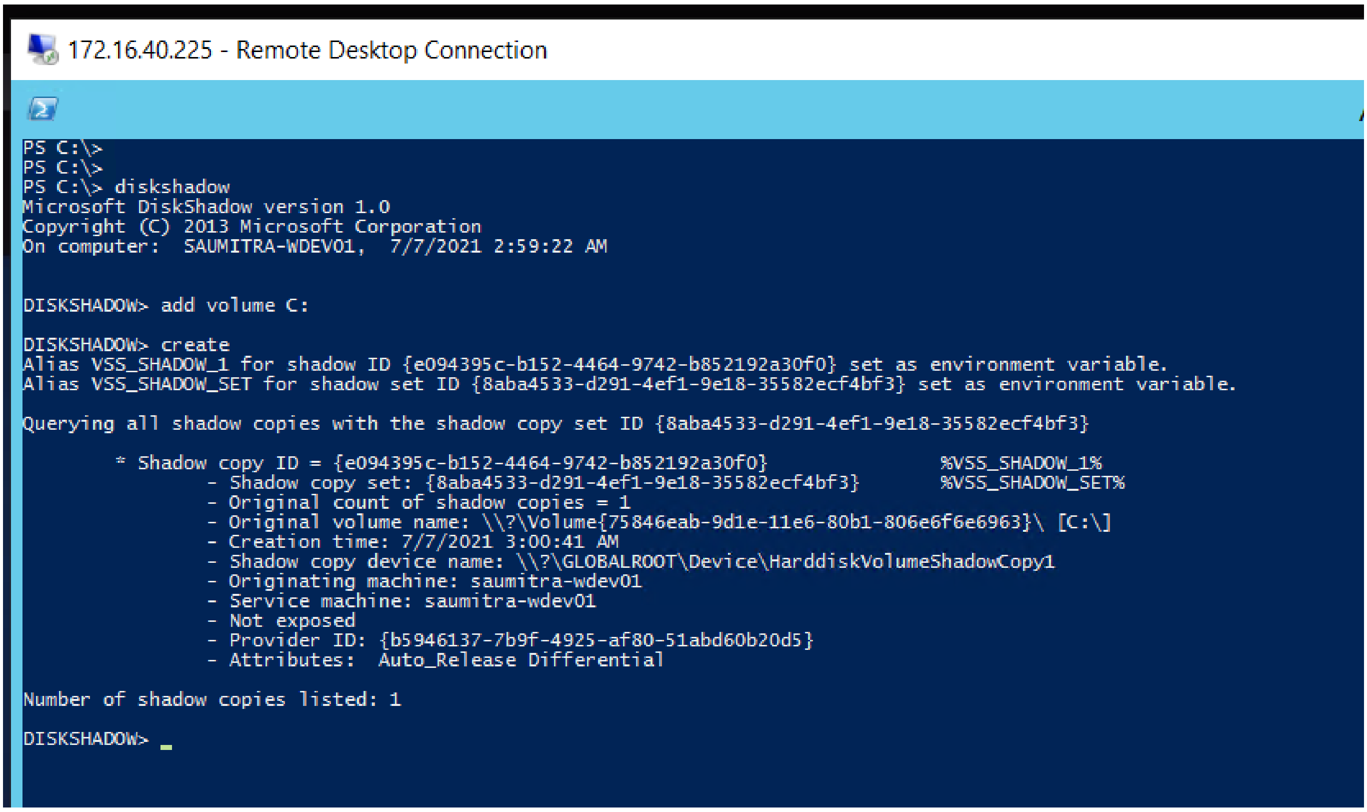
```
PS C:\> dir


    Directory: C:\


Mode                LastWriteTime     Length Name
----                -------------     ------ ----
d----        10/28/2016   5:35 PM            cygwin64
d----         8/22/2013   8:52 AM            PerfLogs
d-r--         12/5/2017   2:23 PM            Program Files
d----        11/14/2016   1:23 PM            Program Files (x86)
d-r--        10/28/2016   7:38 AM            Users
d----        10/28/2016   4:25 PM            WinDDK
d----         12/6/2017  10:55 AM            Windows
-a---          7/7/2021   2:54 AM         61 Donotdelete.txt


PS C:\> cat .\Donotdelete.txt
Saumitra Sapre
The contents of this file are very important.
PS C:\> _
```

Creating a VSS Shadow Copy -- The wrong way 😦

Here we basically tell VSS that we want to add our drive to the VSS service. That is, we intend to backup the changes to this drive. In order to do so, we write the command `diskshadow`. The diskshadow interface helps us to create and manage shadow copies on the system. We then tell `diskshadow` to add our volume to the VSS service. This is done using: `add volume volume_path`.

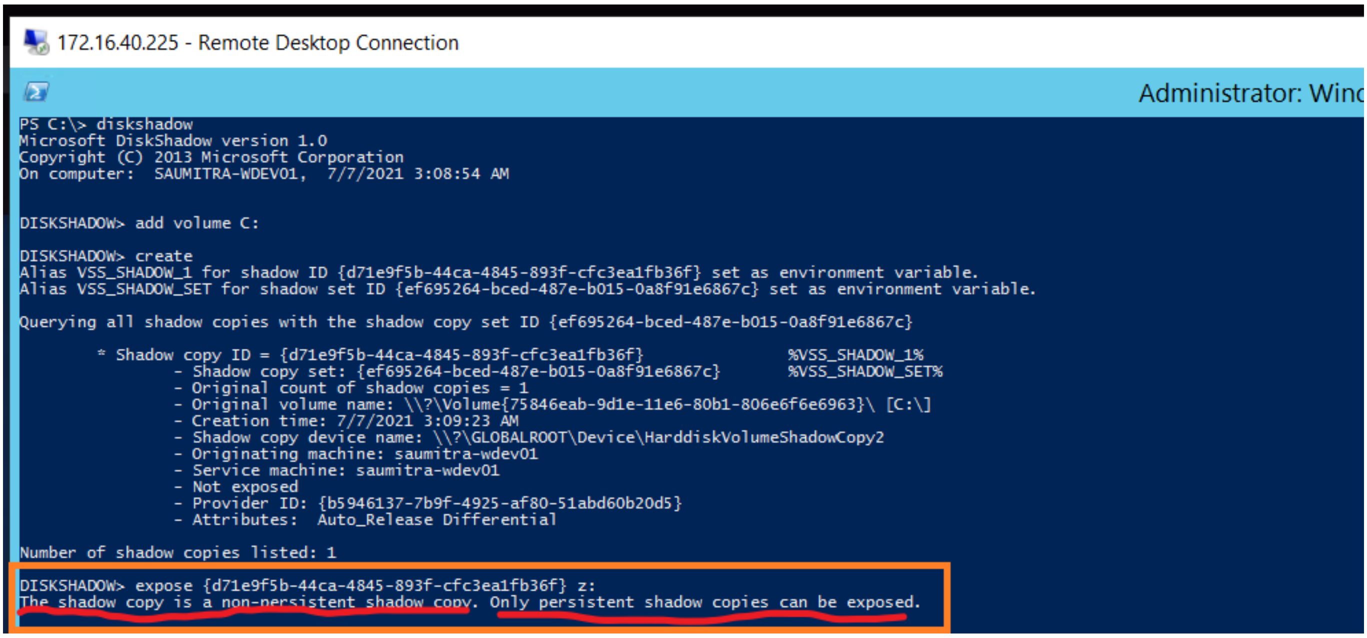
Then, we create the shadow copy (ie the reserved space for backups) using the `create` command.

Let's see the results:



```
PS C:\>
PS C:\>
PS C:\> diskshadow
Microsoft DiskShadow version 1.0
Copyright (C) 2013 Microsoft Corporation
On computer:  SAUMITRA-WDEV01,  7/7/2021 2:59:22 AM

DISKSHADOW> add volume C:

DISKSHADOW> create
Alias VSS_SHADOW_1 for shadow ID {e094395c-b152-4464-9742-b852192a30f0} set as environment variable.
Alias VSS_SHADOW_SET for shadow set ID {8aba4533-d291-4ef1-9e18-35582ecf4bf3} set as environment variable.

Querying all shadow copies with the shadow copy set ID {8aba4533-d291-4ef1-9e18-35582ecf4bf3}

        * Shadow copy ID = {e094395c-b152-4464-9742-b852192a30f0}              %VSS_SHADOW_1%
                - Shadow copy set: {8aba4533-d291-4ef1-9e18-35582ecf4bf3}      %VSS_SHADOW_SET%
                - Original count of shadow copies = 1
                - Original volume name: \\?\Volume{75846eab-9d1e-11e6-80b1-806e6f6e6963}\ [C:\]
                - Creation time: 7/7/2021 3:00:41 AM
                - Shadow copy device name: \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1
                - Originating machine: saumitra-wdev01
                - Service machine: saumitra-wdev01
                - Not exposed
                - Provider ID: {b5946137-7b9f-4925-af80-51abd60b20d5}
                - Attributes:  Auto_Release Differential

Number of shadow copies listed: 1

DISKSHADOW> _
```

In order to expose the shadow copy to the filesystem as a new drive, we need to refer to the Shadow Copy using its ID. But, if we do try this ...

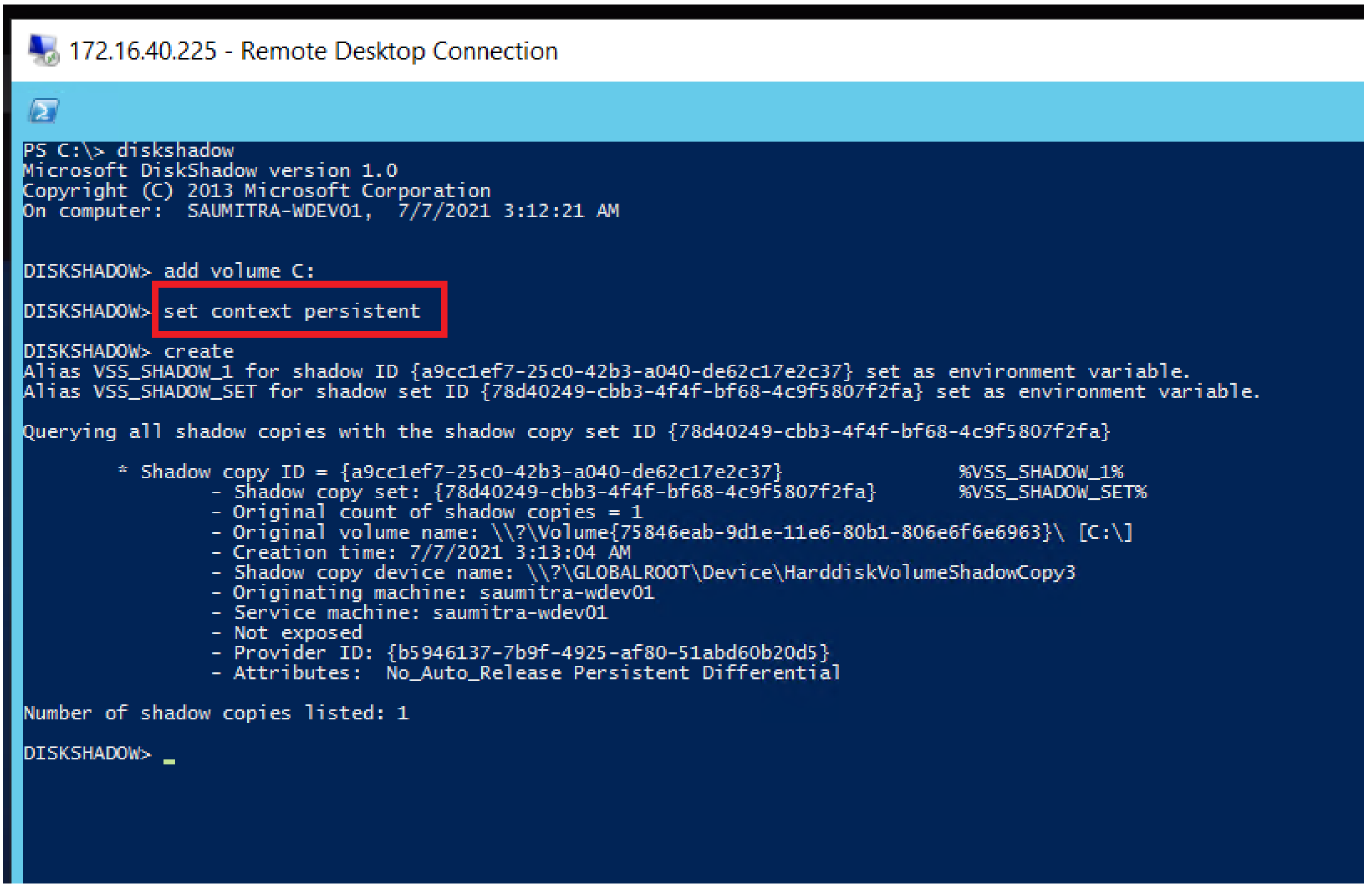


```
PS C:\> diskshadow
Microsoft DiskShadow version 1.0
Copyright (C) 2013 Microsoft Corporation
On computer:  SAUMITRA-WDEV01,  7/7/2021 3:08:54 AM

DISKSHADOW> add volume C:

DISKSHADOW> create
Alias VSS_SHADOW_1 for shadow ID {d71e9f5b-44ca-4845-893f-cfc3ea1fb36f} set as environment variable.
Alias VSS_SHADOW_SET for shadow set ID {ef695264-bced-487e-b015-0a8f91e6867c} set as environment variable.

Querying all shadow copies with the shadow copy set ID {ef695264-bced-487e-b015-0a8f91e6867c}

        * Shadow copy ID = {d71e9f5b-44ca-4845-893f-cfc3ea1fb36f}              %VSS_SHADOW_1%
                - Shadow copy set: {ef695264-bced-487e-b015-0a8f91e6867c}      %VSS_SHADOW_SET%
                - Original count of shadow copies = 1
                - Original volume name: \\?\Volume{75846eab-9d1e-11e6-80b1-806e6f6e6963}\ [C:\]
                - Creation time: 7/7/2021 3:09:23 AM
                - Shadow copy device name: \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy2
                - Originating machine: saumitra-wdev01
                - Service machine: saumitra-wdev01
                - Not exposed
                - Provider ID: {b5946137-7b9f-4925-af80-51abd60b20d5}
                - Attributes:  Auto_Release Differential

Number of shadow copies listed: 1

DISKSHADOW> expose {d71e9f5b-44ca-4845-893f-cfc3ea1fb36f} z:
The shadow copy is a non-persistent shadow copy. Only persistent shadow copies can be exposed.
```

Looks like we cannot explore our newly created shadow file. This is because we did not specify that we want to create a persistent shadow copy. A non-persistent shadow copy is not browsable and will eventually be deleted from the drive after the termination of the backup operation.

Creating a VSS Shadow Copy -- The correct way 😃

Here we follow all the same previous steps. The only difference here is that we explicitly specify that we do want a persistent shadow copy. This is done by using the command: `set context persistent`.

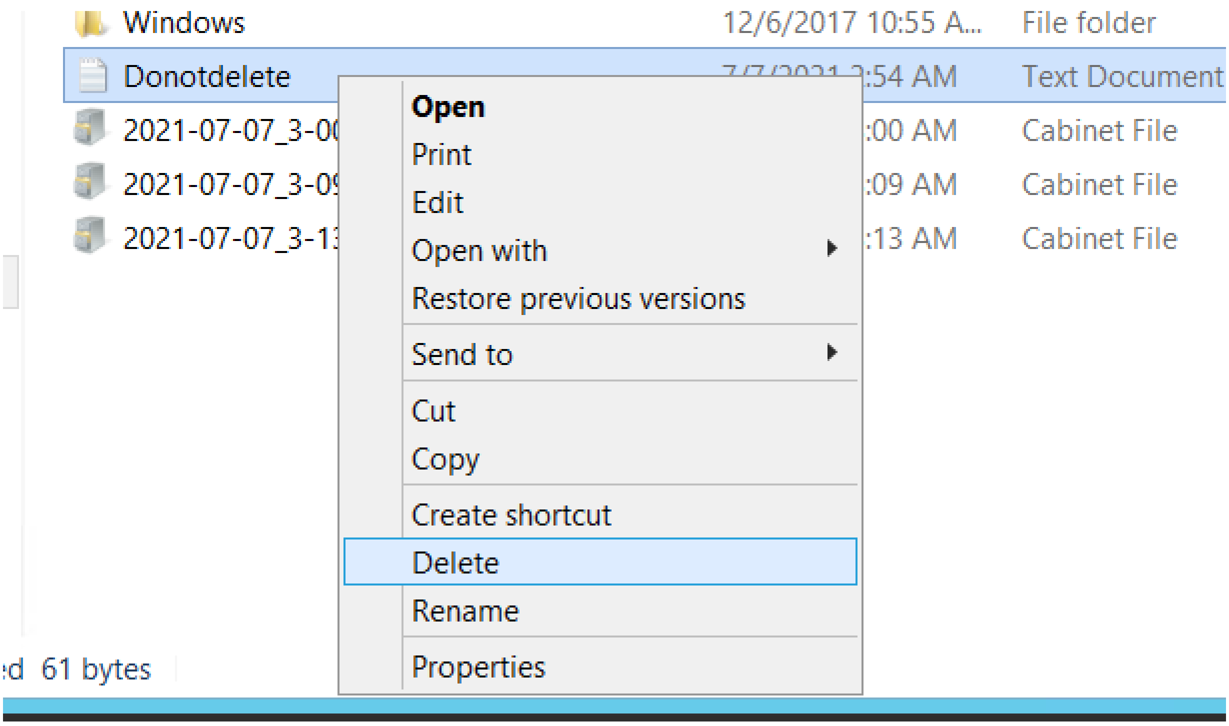


```
PS C:\> diskshadow
Microsoft DiskShadow version 1.0
Copyright (C) 2013 Microsoft Corporation
On computer:  SAUMITRA-WDEV01,  7/7/2021 3:12:21 AM

DISKSHADOW> add volume C:

DISKSHADOW> set context persistent

DISKSHADOW> create
Alias VSS_SHADOW_1 for shadow ID {a9cc1ef7-25c0-42b3-a040-de62c17e2c37} set as environment variable.
Alias VSS_SHADOW_SET for shadow set ID {78d40249-cbb3-4f4f-bf68-4c9f5807f2fa} set as environment variable.

Querying all shadow copies with the shadow copy set ID {78d40249-cbb3-4f4f-bf68-4c9f5807f2fa}

        * Shadow copy ID = {a9cc1ef7-25c0-42b3-a040-de62c17e2c37}               %VSS_SHADOW_1%
                - Shadow copy set: {78d40249-cbb3-4f4f-bf68-4c9f5807f2fa}       %VSS_SHADOW_SET%
                - Original count of shadow copies = 1
                - Original volume name: \\?\Volume{75846eab-9d1e-11e6-80b1-806e6f6e6963}\ [C:\]
                - Creation time: 7/7/2021 3:13:04 AM
                - Shadow copy device name: \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy3
                - Originating machine: saumitra-wdev01
                - Service machine: saumitra-wdev01
                - Not exposed
                - Provider ID: {b5946137-7b9f-4925-af80-51abd60b20d5}
                - Attributes:  No_Auto_Release Persistent Differential

Number of shadow copies listed: 1

DISKSHADOW> _
```

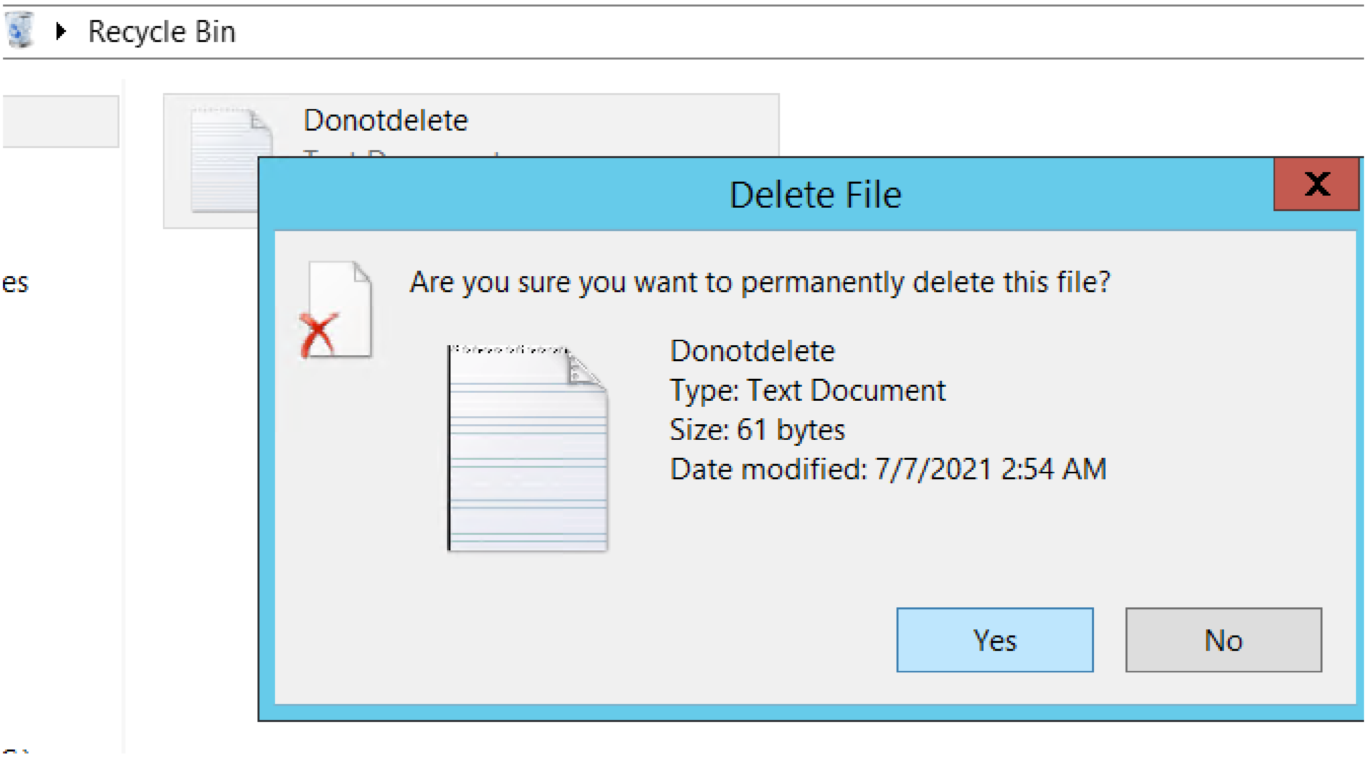
Such type of a shadow copy will *persist* on the disk even after we exit `diskshadow`.

Making changes

Let us delete the `Donotdelete.txt` file that we created at the beginning. If we are able to recover this file later using the shadow copy, then we shall say that we have successfully saved a snapshot of the volume in its previous step.



Also, deleting the file from the recycle bin to ensure that it permanently deleted.



Are you sure you want to permanently delete this file?

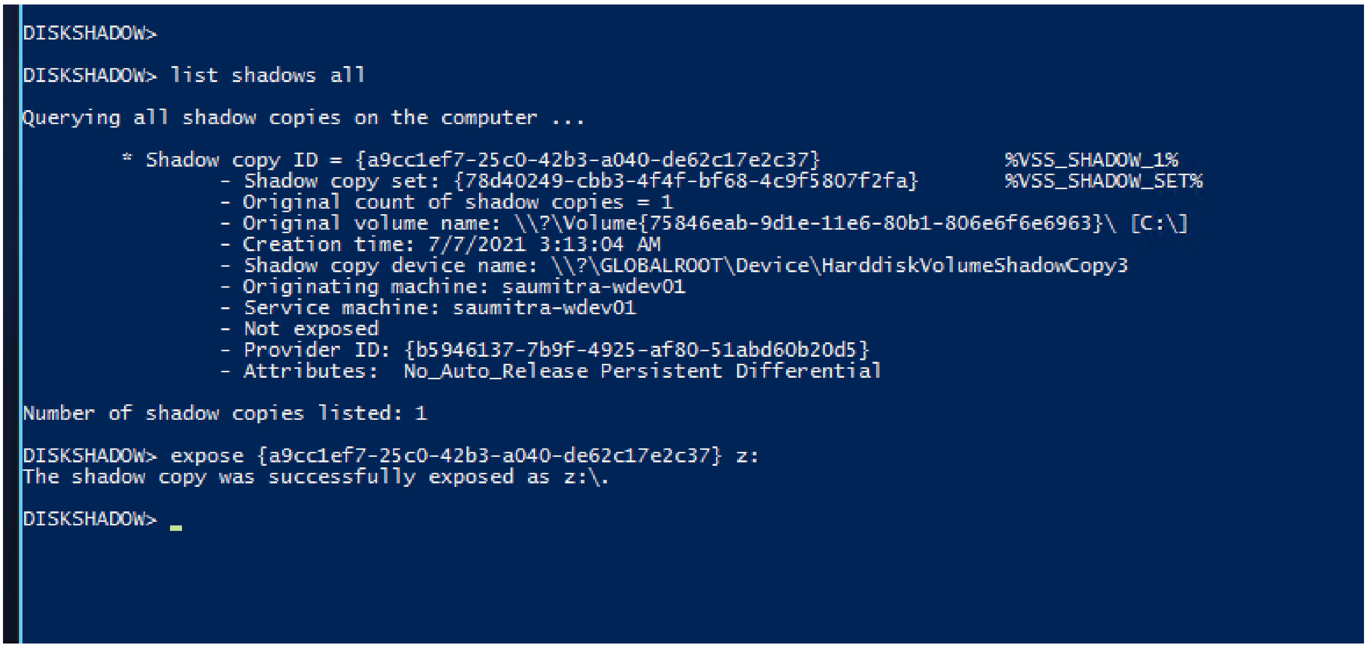
Donotdelete
Type: Text Document
Size: 61 bytes
Date modified: 7/7/2021 2:54 AM

Exposing Shadow Copies to the file system

Now, lets expose the Shadow Copy that we created to the filesystem. We do this by first entering the `diskshadow` interface. We then list all shadow copies that are present on the system using `list shadows all`.

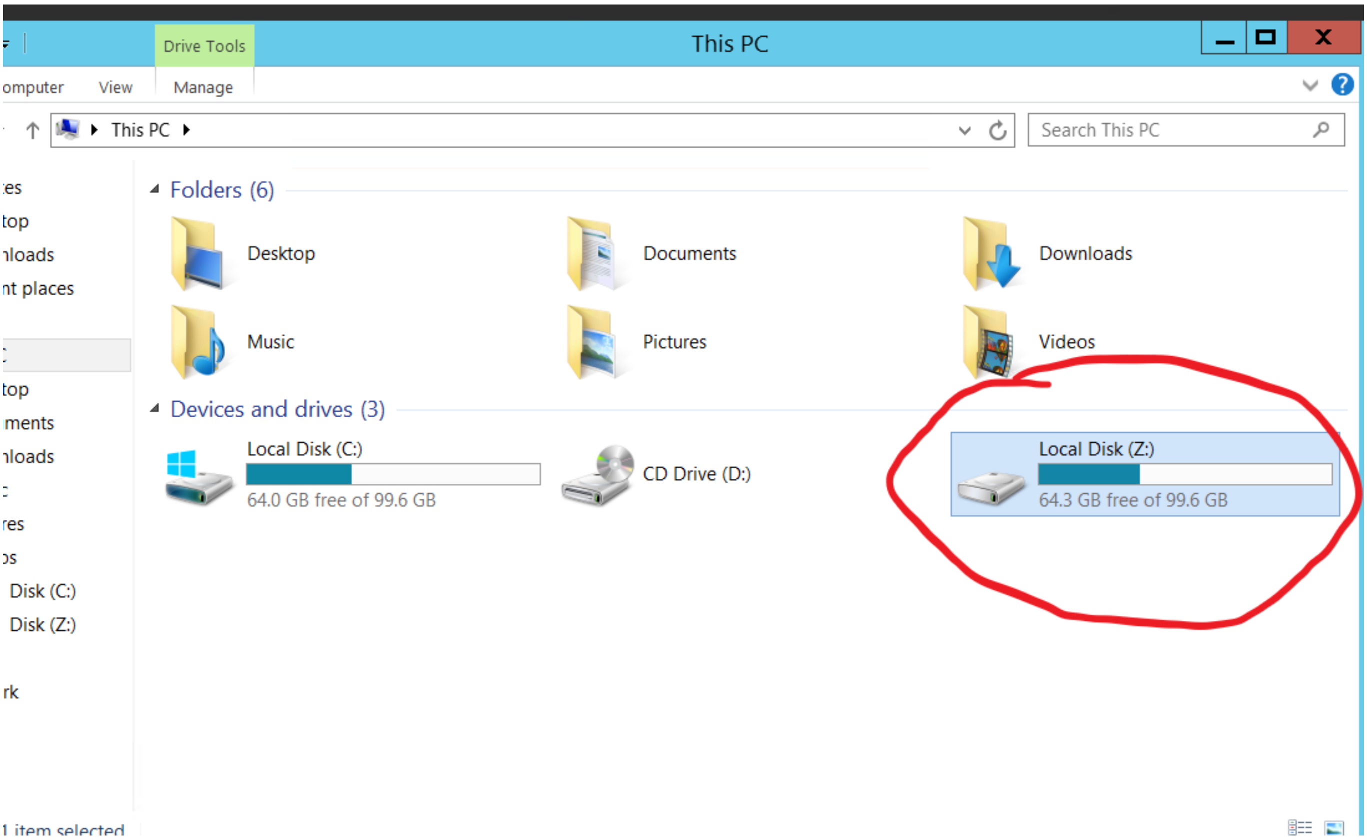
Then, we refer to our Shadow Copy using its ID, and then use the `expose` command by specifying a new drive letter.

```
expose shadow_copy_id new_drive_letter
```

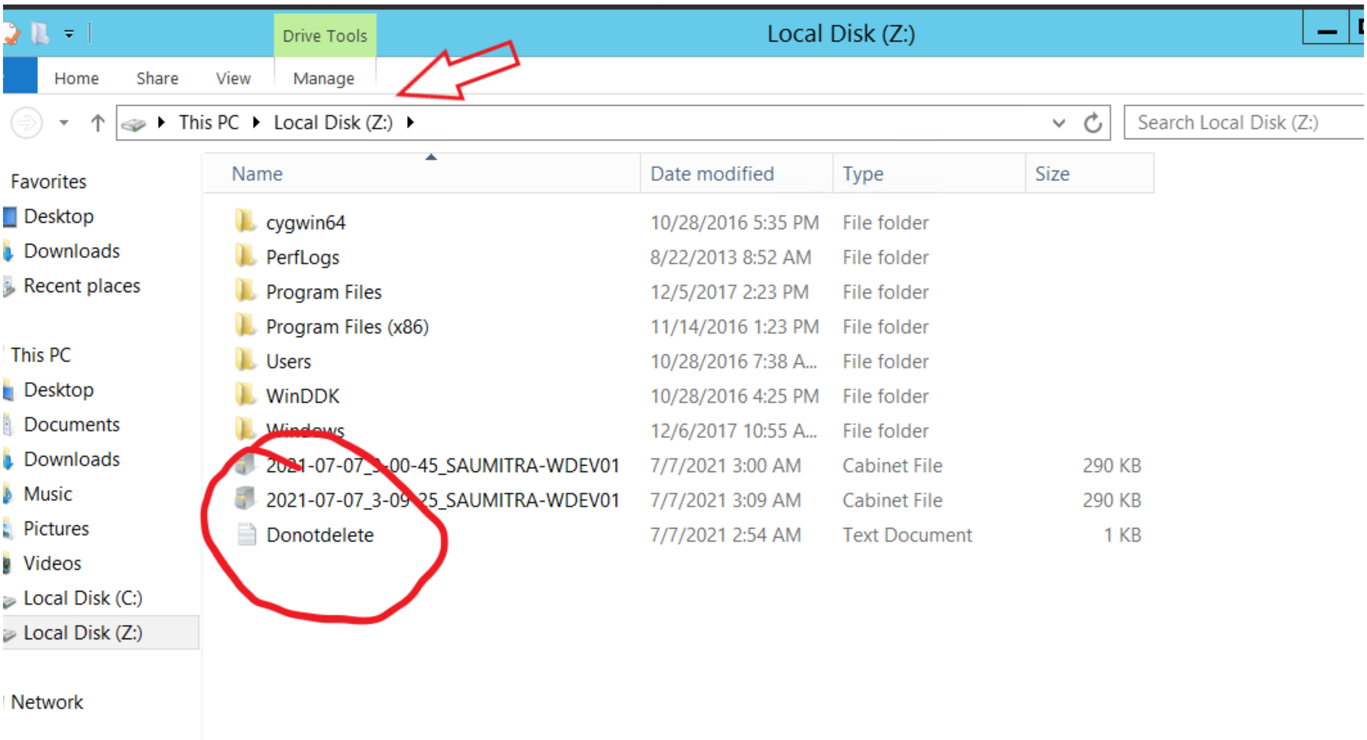


```
DISKSHADOW>

DISKSHADOW> list shadows all

Querying all shadow copies on the computer ...

        * Shadow copy ID = {a9cc1ef7-25c0-42b3-a040-de62c17e2c37}              %VSS_SHADOW_1%
                - Shadow copy set: {78d40249-cbb3-4f4f-bf68-4c9f5807f2fa}      %VSS_SHADOW_SET%
                - Original count of shadow copies = 1
                - Original volume name: \\?\Volume{75846eab-9d1e-11e6-80b1-806e6f6e6963}\ [C:\]
                - Creation time: 7/7/2021 3:13:04 AM
                - Shadow copy device name: \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy3
                - Originating machine: saumitra-wdev01
                - Service machine: saumitra-wdev01
                - Not exposed
                - Provider ID: {b5946137-7b9f-4925-af80-51abd60b20d5}
                - Attributes:  No_Auto_Release Persistent Differential

Number of shadow copies listed: 1

DISKSHADOW> expose {a9cc1ef7-25c0-42b3-a040-de62c17e2c37} z:
The shadow copy was successfully exposed as z:\.

DISKSHADOW> _
```

We can now see that our system has a new drive `Z:/` mounted. It is the exact snapshot of our `C:/` drive before we made changes.



Going into the drive `Z:/`, we can see that our deleted file is indeed present.

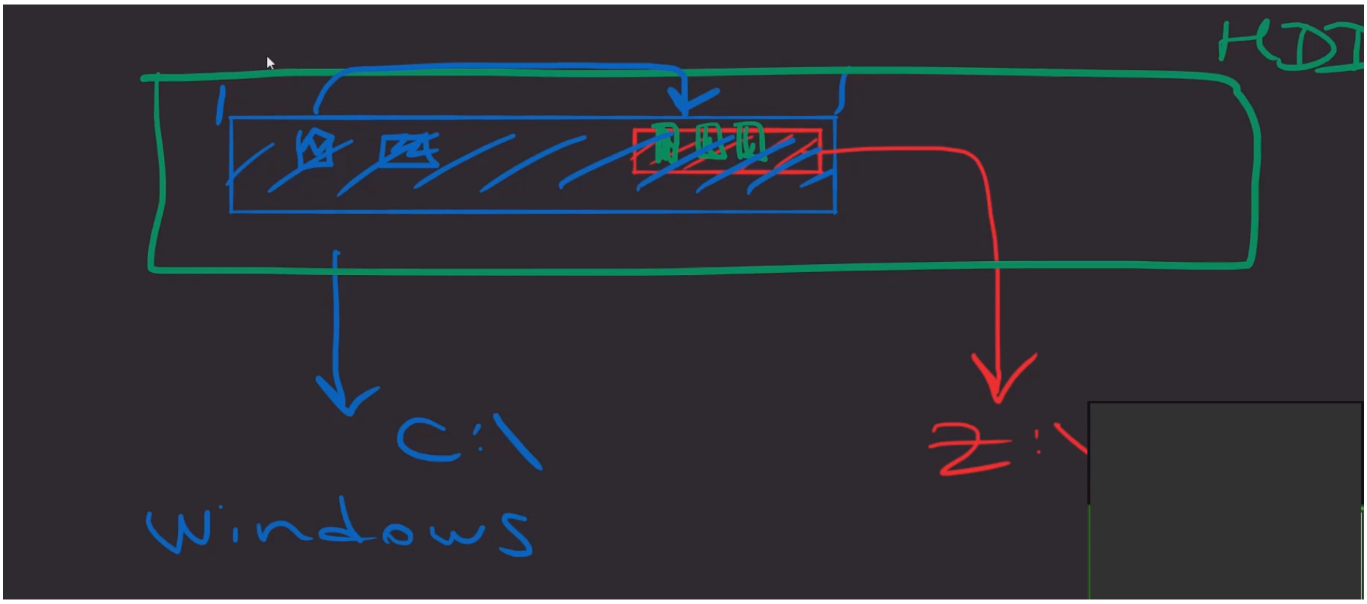


Name	Date modified	Type	Size
cygwin64	10/28/2016 5:35 PM	File folder	
PerfLogs	8/22/2013 8:52 AM	File folder	
Program Files	12/5/2017 2:23 PM	File folder	
Program Files (x86)	11/14/2016 1:23 PM	File folder	
Users	10/28/2016 7:38 A...	File folder	
WinDDK	10/28/2016 4:25 PM	File folder	
Windows	12/6/2017 10:55 A...	File folder	
2021-07-07_3-00-45_SAUMITRA-WDEV01	7/7/2021 3:00 AM	Cabinet File	290 KB
2021-07-07_3-09-25_SAUMITRA-WDEV01	7/7/2021 3:09 AM	Cabinet File	290 KB
Donotdelete	7/7/2021 2:54 AM	Text Document	1 KB

Thus, we can see that we have successfully restored our original file with the help of Windows Volume Shadow Copy Service.

How it works

VSS works on the concept of differential backups. This means that whatever files remain unchanged are taken from the original drive. Only the originals of the changed files (*the differences*) are backed up in the Shadow Copy space.



This means that whatever we are seeing in drive `Z:/` is partly coming from `C:/` drive (the original unchanged files), and partly from the Shadow Copy Volume (*the changes*).

Cleaning Up

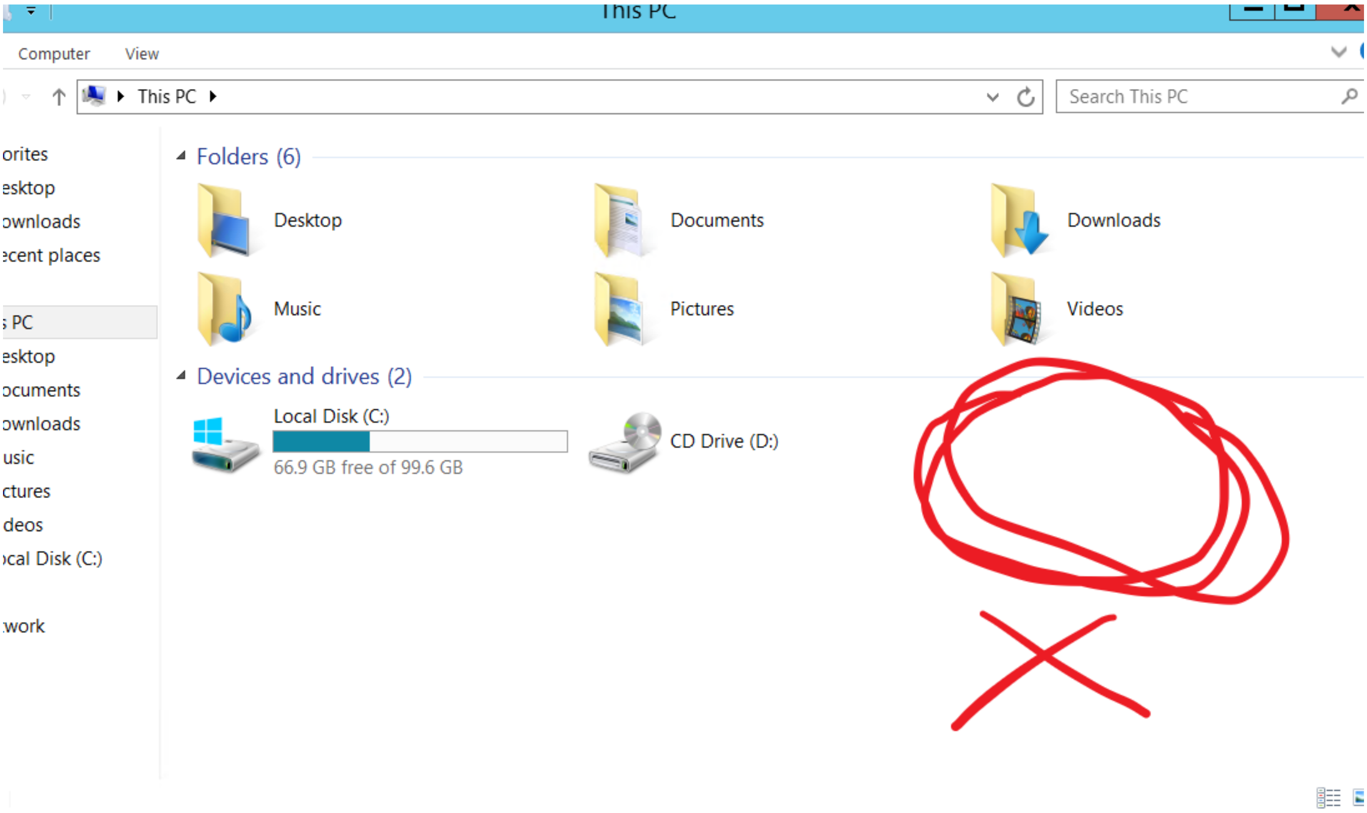
After we copy all the files we want to restore back into `C:/`, we can delete the shadow copy, as it is of no use to us now.For that we need to unexpose our drive `Z:`. First we enter the `diskshadow` interface. Then we list all available shadow copies available.

```
list shadows all
```

Then, we unexpose the shadow copy we want to by referring to it using its ID.

```
unexpose shadow_copy_id
```

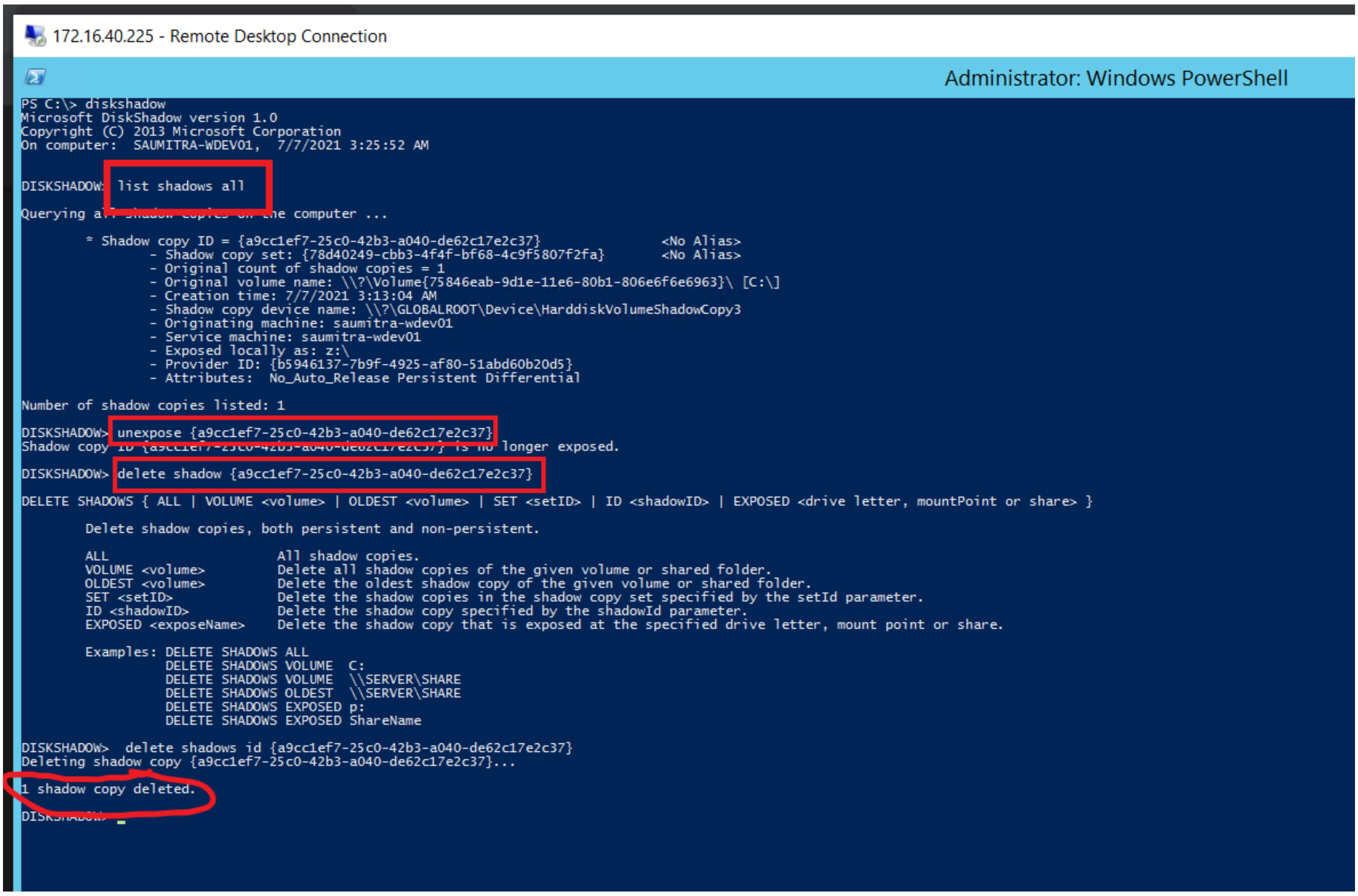
This will make the drive `Z:/` disappear from the file explorer menu.



Finally, we delete the shadow copy by referring to it using its ID.

`delete shadows id shadow_copy_id`

Thus, all shadow copies we created are deleted.


172.16.40.225 - Remote Desktop Connection
Administrator: Windows PowerShell

```
PS C:\> diskshadow
Microsoft DiskShadow version 1.0
Copyright (C) 2013 Microsoft Corporation
On computer:  SAUMITRA-WDEV01,  7/7/2021 3:25:52 AM

DISKSHADOW> list shadows all

Querying all shadow copies on the computer ...

        * Shadow copy ID = {a9cc1ef7-25c0-42b3-a040-de62c17e2c37}               <No Alias>
                - Shadow copy set: {78d40249-cbb3-4f4f-bf68-4c9f5807f2fa}       <No Alias>
                - Original count of shadow copies = 1
                - Original volume name: \\?\Volume{75846eab-9d1e-11e6-80b1-806e6f6e6963}\ [C:\]
                - Creation time: 7/7/2021 3:13:04 AM
                - Shadow copy device name: \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy3
                - Originating machine: saumitra-wdev01
                - Service machine: saumitra-wdev01
                - Exposed locally as: z:\
                - Provider ID: {b5946137-7b9f-4925-af80-51abd60b20d5}
                - Attributes:  No_Auto_Release Persistent Differential

Number of shadow copies listed: 1

DISKSHADOW> unexpose {a9cc1ef7-25c0-42b3-a040-de62c17e2c37}
Shadow copy ID {a9cc1ef7-25c0-42b3-a040-de62c17e2c37} is no longer exposed.

DISKSHADOW> delete shadow {a9cc1ef7-25c0-42b3-a040-de62c17e2c37}

DELETE SHADOWS { ALL | VOLUME <volume> | OLDEST <volume> | SET <setID> | ID <shadowID> | EXPOSED <drive letter, mountPoint or share> }

        Delete shadow copies, both persistent and non-persistent.

        ALL                     All shadow copies.
        VOLUME <volume>         Delete all shadow copies of the given volume or shared folder.
        OLDEST <volume>         Delete the oldest shadow copy of the given volume or shared folder.
        SET <setID>             Delete the shadow copies in the shadow copy set specified by the setId parameter.
        ID <shadowID>           Delete the shadow copy specified by the shadowId parameter.
        EXPOSED <exposeName>    Delete the shadow copy that is exposed at the specified drive letter, mount point or share.

        Examples: DELETE SHADOWS ALL
                  DELETE SHADOWS VOLUME  C:
                  DELETE SHADOWS VOLUME  \\SERVER\SHARE
                  DELETE SHADOWS OLDEST  \\SERVER\SHARE
                  DELETE SHADOWS EXPOSED p:
                  DELETE SHADOWS EXPOSED ShareName

DISKSHADOW>  delete shadows id {a9cc1ef7-25c0-42b3-a040-de62c17e2c37}
Deleting shadow copy {a9cc1ef7-25c0-42b3-a040-de62c17e2c37}...

1 shadow copy deleted.

DISKSHADOW> _
```