Azure NetApp Files

NetApp Automation

NetApp
November 18, 2025

This PDF was generated from https://docs.netapp.com/us-en/netapp-automation/solutions/anf-oracle.html on November 18, 2025. Always check docs.netapp.com for the latest.

Table of Contents

Azure NetApp Files . . . 1
Install Oracle using Azure NetApp Files . . . 1
Step 1: Install and configure Docker . . . 1
Step 2: Install Docker Compose and the NFS utilities . . . 2
Step 3: Download the Oracle installation files . . . 3
Step 4: Prepare the Docker image . . . 3
Step 5: Create an external volume . . . 4
Step 6: Install Oracle . . . 5
Step 7: Validate the Oracle installation . . . 5
Step 8: Install the azacsnap utility and perform a snapshot backup . . . 6
Step 9: Optionally migrate an on-premise PDB to the cloud . . . 6

Azure NetApp Files

Install Oracle using Azure NetApp Files

You can use this automation solution to provision Azure NetApp Files volumes and install Oracle on an available virtual machine. Oracle then uses the volumes for data storage.

About this solution

At a high level, the automation code provided with this solution performs the following actions:

- Set up a NetApp account on Azure
- Set up a storage capacity pool on Azure
- Provision the Azure NetApp Files volumes based on the definition
- Create the mount points
- Mount the Azure NetApp Files volumes to the mount points
- Install Oracle on the Linux server
- Create the listeners and database
- Create the Pluggable Databases (PDBs)
- Start the listener and Oracle instance
- Install and configure the `azacsnap` utility to take a snapshot

Before you begin

You must have the following to complete the installation:

- You need to download the Oracle using Azure NetApp Files automation solution through the NetApp Console web UI. The solution is packaged as file `na_oracle19c_deploy-master.zip`.
- A Linux VM with the following characteristics:
  - RHEL 8 (Standard_D8s_v3-RHEL-8)
  - Deployed on the same Azure Virtual Network used for the Azure NetApp Files provisioning
- An Azure account

The automation solution is provided as an image and run using Docker and Docker Compose. You need to install both of these on the Linux virtual machine as described below.

You should also register the VM with RedHat using the command `sudo subscription-manager register`. The command will prompt you for your account credentials. If needed, you can create an account at https://developers.redhat.com/.

Step 1: Install and configure Docker

Install and configure Docker in a RHEL 8 Linux virtual machine.

Steps

1. Install the Docker software using the following commands.

```
dnf config-manager --add
-repo=https://download.docker.com/linux/centos/docker-ce.repo
dnf install docker-ce --nobest -y
```

2. Start Docker and display the version to confirm the installation was successful.

```
systemctl start docker
systemctl enable docker
docker --version
```

3. Add the required Linux group with an associated user.

   First check if the group **docker** exists in your Linux system. If it doesn't, create the group and add the user. By default, the current shell user is added to the group.

```
sudo groupadd docker
sudo usermod -aG docker $USER
```

4. Activate the new group and user definitions

   If you created a new group with a user, you need to activate the definitions. To do this, you can logout of Linux and then back in. Or you can run the following command.

```
newgrp docker
```

Step 2: Install Docker Compose and the NFS utilities

Install and configure Docker Compose along with the NFS utilities package.

Steps

1. Install Docker Compose and display the version to confirm the installation was successful.

```
dnf install curl -y
curl -L
"https://github.com/docker/compose/releases/download/1.29.2/docker-
compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
sudo chmod +x /usr/local/bin/docker-compose
docker-compose --version
```

2. Install the NFS utilities package.

```
sudo yum install nfs-utils
```

Step 3: Download the Oracle installation files

Download the required Oracle installation and patch files as well as the `azacsnap` utility.

Steps

1. Sign in to your Oracle account as needed.
2. Download the following files.

File	Description
LINUX.X64_193000_db_home.zip	19.3 base installer
p31281355_190000_Linux-x86-64.zip	19.8 RU patch
p6880880_190000_Linux-x86-64.zip	opatch version 12.2.0.1.23
azacsnap_installer_v5.0.run	azacsnap installer

3. Place all the installation files in the folder `/tmp/archive`.
4. Make sure all users on the database server have full access (read, write, execute) to the folder `/tmp/archive`.

Step 4: Prepare the Docker image

You need to extract and load the Docker image provided with the automation solution.

Steps

1. Copy the solution file `na_oracle19c_deploy-master.zip` to the virtual machine where the automation code will run.

```
scp -i ~/<private-key.pem> -r na_oracle19c_deploy-master.zip
user@<IP_ADDRESS_OF_VM>
```

   The input parameter `private-key.pem` is your private key file used for Azure virtual machine authentication.

2. Navigate to the correct folder with the solution file and unzip the file.

```
unzip na_oracle19c_deploy-master.zip
```

3. Navigate to the new folder `na_oracle19c_deploy-master` created with the unzip operation and list the files. You should see file `ora_anf_bck_image.tar`.

```
ls -lt
```

4. Load the Docker image file. The load operation should normally complete in a few seconds.

```
docker load -i ora_anf_bck_image.tar
```

5. Confirm the Docker image is loaded.

```
docker images
```

You should see the Docker image `ora_anf_bck_image` with the tag `latest`.

```
   REPOSITORY          TAG       IMAGE ID      CREATED      SIZE
ora_anf_bck_image    latest   ay98y7853769   1 week ago   2.58GB
```

Step 5: Create an external volume

You need an external volume to make sure the Terraform state files and other important files are persistent. These files must be available for Terraform to run the workflow and deployments.

Steps

1. Create an external volume outside of Docker Compose.

   Make sure to update the volume name before running the command.

```
docker volume create <VOLUME_NAME>
```

2. Add the path to the external volume to the `.env` environment file using the command:

   `PERSISTENT_VOL=path/to/external/volume:/ora_anf_prov`.

   Remember to keep the existing file contents and colon formatting. For example:

```
PERSISTENT_VOL= ora_anf _volume:/ora_anf_prov
```

3. Update the Terraform variables.
   a. Navigate to the folder `ora_anf_variables`.
   b. Confirm the following two files exist: `terraform.tfvars` and `variables.tf`.
   c. Update the values in `terraform.tfvars` as required for your environment.

Step 6: Install Oracle

You can now provision and install Oracle.

Steps

1. Install Oracle using the following sequence of commands.

```
docker-compose up terraform_ora_anf
bash /ora_anf_variables/setup.sh
docker-compose up linux_config
bash /ora_anf_variables/permissions.sh
docker-compose up oracle_install
```

2. Reload your Bash variables and confirm by displaying the value for `ORACLE_HOME`.
   a. `cd /home/oracle`
   b. `source .bash_profile`
   c. `echo $ORACLE_HOME`
3. You should be able to login to Oracle.

```
sudo su oracle
```

Step 7: Validate the Oracle installation

You should confirm the Oracle installation was successful.

Steps

1. Log in to the Linux Oracle server and display a list of the Oracle processes. This confirms the installation completed as expected and the Oracle database is running.

```
ps -ef | grep ora
```

2. Log in to the database to examine the database configuration and to confirm the PDBs were created properly.

```
sqlplus / as sysdba
```

   You should see output similar to the following:

```
SQL*Plus: Release 19.0.0.0.0 - Production on Thu May 6 12:52:51 2021
Version 19.8.0.0.0

Copyright (c) 1982, 2019, Oracle. All rights reserved.

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.8.0.0.0
```

3. Execute a few simple SQL commands to confirm the database is available.

```
select name, log_mode from v$database;
show pdbs.
```

Step 8: Install the azacsnap utility and perform a snapshot backup

You need to install and run the `azacsnap` utility to perform a snapshot backup.

Steps

1. Install the container.

```
docker-compose up azacsnap_install
```

2. Switch to the snapshot user account.

```
su - azacsnap
execute /tmp/archive/ora_wallet.sh
```

3. Configure a storage backup detail file. This will create the `azacsnap.json` configuration file.

```
cd /home/azacsnap/bin/
azacsnap -c configure --configuration new
```

4. Perform a snapshot backup.

```
azacsnap -c backup --other data --prefix ora_test --retention=1
```

Step 9: Optionally migrate an on-premise PDB to the cloud

You can optionally migrate the on-premise PDB to the cloud.

Steps

1. Set the variables in the `tfvars` files as needed for your environment.
2. Migrate the PDB.

```
docker-compose -f docker-compose-relocate.yml up
```

Copyright information

Copyright © 2025 NetApp, Inc. All Rights Reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

LIMITED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data -Noncommercial Items at DFARS 252.227-7013 (FEB 2014) and FAR 52.227-19 (DEC 2007).

Data contained herein pertains to a commercial product and/or commercial service (as defined in FAR 2.101) and is proprietary to NetApp, Inc. All NetApp technical data and computer software provided under this Agreement is commercial in nature and developed solely at private expense. The U.S. Government has a non-exclusive, non-transferrable, nonsublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b) (FEB 2014).

Trademark information

NETAPP, the NETAPP logo, and the marks listed at http://www.netapp.com/TM are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.