



Automating VM Backup on the Oracle Private Cloud Appliance

Oracle Private Cloud Appliance

August 1st, 2024 | Version 2

Copyright © 2024, Oracle and/or its
affiliates Public

PURPOSE STATEMENT

The document summarizes how customers can utilize a series of bash scripts to enable VM Backup on the Private Cloud Appliance.

DISCLAIMER

This document in any form, software, or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced, or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement, nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle.

CONTENTS

| | |
|--|-----------|
| Purpose Statement | 1 |
| Disclaimer | 1 |
| Introduction | 3 |
| Scope and content | 3 |
| Advantages of Oracle Private Cloud Appliance | 3 |
| VM BACKUP on Oracle Private Cloud Appliance | 4 |
| Automating Off System Backup methods | 4 |
| Single Instance Backup | 4 |
| Multiple Instance Backup | 6 |
| List All Backups | 7 |
| Get backups | 8 |
| Single Instance Restore | 9 |
| Multiple Instance Restore | 11 |
| Deleting the Backups | 12 |
| Download from Object Store | 14 |
| Transferring an Instance Backup | 14 |
| Transferring an Instance Backup to Another System | 14 |
| Transferring an Instance Backup from Another System to Private Cloud Appliance | 14 |
| RESOURCES | 15 |

INTRODUCTION

Oracle Private Cloud Appliance is uniquely compatible with OCI providing fast and efficient infrastructure for modern software and business applications. Private Cloud Appliance has the same infrastructure constructs (including APIs and SDKs) as OCI. This enables customers to adopt a “develop once and deploy anywhere—on-premises or on OCI” approach to rapidly design and develop high-performance applications and middleware.

SCOPE AND CONTENT

This document provides multiple ways in which to use bash scripts that enable VM Backup on Oracle Private Cloud Appliance, transfer the backup to another object store and restore from the backup on an Oracle Private Cloud Appliance.

The steps documented here are applicable to the following systems referred to as Oracle Private Cloud Appliance or Private Cloud Appliance:

- Oracle Private Cloud Appliance X9-2
- Oracle Private Cloud Appliance X10

ADVANTAGES OF PRIVATE CLOUD APPLIANCE

Oracle Private Cloud Appliance is a rack-scale engineered system delivering Oracle Cloud Infrastructure compute, storage, and networking constructs on-premises. It enables rapid deployment of applications, middleware and workloads that are cloud-compatible via automation in an OCI-like environment while being disconnected from the public cloud. Private Cloud Appliance is the ideal platform alongside Oracle Exadata and Oracle Database Appliance platforms offering lowest latency and highest performance between the application and database layers. Private Cloud Appliance is designed for customers who want a cloud-like development and deployment experience while also meeting data residency requirements.

VM BACKUP ON THE PRIVATE CLOUD APPLIANCE

Private Cloud Appliance offers customers the capability to backup and restore VM instances. The instance backup is created in a Private Cloud Appliance Object Storage Bucket where a customer can choose to either copy the backup to another server or object store in the customer datacenter. When required, a customer can import the backup into any Oracle Private Cloud Appliance Object Storage Bucket and restore the VM.

AUTOMATING OFF SYSTEM BACKUP METHODS

A customer can utilize the sample bash scripts as below to automate the full workflow of Off System

Backup: **Note:**

- The bash script in this post is structured for Linux. If you are trying to follow this post on Windows, you will need to modify escape sequences and possibly some commands.
- Fill bash script with your values and as per your environment as below added are sample scripts for how off system backup feature can be automated.
 - Users can also put in required profile/value in a separate script instead of individual script as below are just sample scripts
 - Object Store bucket has to be pre-created for the off-system backup scripts

Single Instance Backup

```
#!/bin/bash
##### Fill these in with your values ##### #OCID of the tenancy calls are being
made in to
tenancy_ocid="ocid1.tenancy.XXXX.XXXX.gidclawftnq3vo572t4cb46tmqxmalg4aivv7ttoa1a7y0mibcak0009 0146"

# OCID of the user making the rest call
user_ocid="ocid1.user.XXXX.XXXX.zknu1e9y13efvmmpu6doo8arto2eogrds6abbx0zdsygaz9qglksdqxxq|0m"

# path to the private PEM format key for this user
privateKeyPath="/home/sampleuser/.oci/oci_api_key.pem"

# fingerprint of the private key for this
user fingerprint="99:aa....."

# The REST api you want to call, with any required
parameters. rest_api="/20160918/instances"

# The PCA you want to make the call against example
iaas.<pcaname>.domainname> host="iaas.pca.oraclecloud..." # The Instance ID for
backup
instance_id="/ocid1.instance.XXXX.XXXX.7kh649y8kd9kgxcmidbmdhvj1eh0ukb3gnd5ejvxm85f7ybm1o2qtrk 0p91g"
```

```

# Export action for backup
rest_action="/actions/export"

#Object Store data for export
objectstore_payload="{\"bucketName\": \"Image\", \"destinationType\": \"objectStorageTuple\",
\"namespaceName\": \"backup_tenant\", \"compartmentId\": \"${tenancy_ocid}\"}"

#data load for token gathering
data_login="{\"username\": \"admin\", \"password\": \"XXXXXX\", \"tenancy\":
\"backup_tenant\"}"

# Get login tokens
TOKEN=$(curl --no-proxy "*" -k -X POST --header "Content-Type: application/json" -d "$data_login" -sS
https://$host/20160918/login | grep token | awk '{print $2}' | sed -E 's/(^)|(..$)//g')

# Export the Instance echo
"=====

set -x

curl --no-proxy "*" -k -H "AUTHORIZATION: Bearer $TOKEN" -X POST --data "$objectstore_payload"
-sS https://$host$rest_api$instance_id$rest_action

```

Multiple Instance Backup

```
#!/bin/bash
##### Fill these in with your values #####
#OCID of the tenancy calls are being made in to
tenancy_ocid="ocid1.tenancy.XXXX.XXXX.gidclawftnq3vo572t4cb46tmqxmlg4aivv7ttoa1a7y0mibcak0009 0146"

# OCID of the user making the rest call
user_ocid="ocid1.user.XXXX.XXXX.zknu1e9y13efvmmpu6doo8arto2eogrds6abbx0zdsygaz9qglksdqxxql0m"

# path to the private PEM format key for this user privateKeyPath="/home/sampleuser/.oci/oci_api_key.pem"

# fingerprint of the private key for this user
fingerprint="99:aa....."

# The REST api you want to call, with any required parameters. rest_api="/20160918/instances"

# The PCA you want to make the call against example
iaas.<pcaname>.<domainname> host="iaas.pca.oraclecloud. "

# Multiple Instance ID for backup with space between the instance id
instance_id_list="/ocid1.instance.XXXX.XXXX.7kh649y8kd9kgxcmidbmdhvj1eh0ukb3gnd5ejvxn85f7ybm10
2qtrkOp91g
/ocid1.instance.XXXX.XXXX.bjzvv4oks8iiu2zreked5l5ff529bzug9nq9w6cvdv14ygbr7kx4v778nx78"

# Export action for backup
rest_action="/actions/export"

#Object Store data for export
objectstore_payload="{\"bucketName\": \"Image\", \"destinationType\": \"objectStorageTuple\",
\"namespaceName\": \"backup_tenant\", \"compartmentId\": \"${tenancy_ocid}\"}"

#data load for token gathering
data_login="{\"username\": \"admin\", \"password\": \"XXXXXX\", \"tenancy\":
\"backup_tenant\"}"
```

```

# Get login tokens
TOKEN=$(curl --no-proxy "*" -k -X POST --header "Content-Type: application/json" -d "$data_login" -sS
https://$host/20160918/login | grep token | awk '{print $2}' | sed -E 's/(^.)|(..$)//g')

# Export Multiple Instances
echo "=====
set -x
for instance_id in $instance_id_list do
    echo "Exporting $instance" set -x
    curl --no-proxy "*" -k -H "AUTHORIZATION: Bearer $TOKEN" -X POST --data "$subjectstore_payload" -sS
https://$host$rest_api$instance_id$rest_action
done

```

List All Backups

```

#!/bin/bash
##### Fill these in with your values ##### #OCID of the tenancy calls are being
made in to
tenancy_ocid="ocid1.tenancy.XXXX.XXXX.gidclawftnq3vo572t4cb46tmqxmalg4aivv7ttoa1a7y0mibcak0009 0146"

# OCID of the user making the rest call
user_ocid="ocid1.user.XXXX.XXXX.zknu1e9y13efvmmpu6doo8arto2eogrds6abbx0zdsygaz9qglksdqxxq|0m"

# path to the private PEM format key for this user
privateKeyPath="/home/sampleuser/.oci/oci_api_key.pem"

# fingerprint of the private key for this
user fingerprint="99:aa....."

# The REST api you want to call, with any required parameters.
rest_api="/20160918/instanceBackups"

# The PCA you want to make the call against example
iaas.<pcaname>.<domainname> host="iaas.pca.oraclecloud..."

```



```

#data load for token gathering
data_login="{\"username\": \"admin\", \"password\": \"XXXXX\", \"tenancy\":
\"backup_tenant\"}"

#Get login tokens
TOKEN=$(curl --no-proxy "*" -k -X POST --header "Content-Type: application/json" -d "$data_login" -sS
https://$host/20160918/login | grep token | awk '{print $2}' | sed -E 's/(^.)|(..$)//g')

# List All the Instance backup
echo "=====
set -x
curl --no-proxy "*" -k -H "AUTHORIZATION: Bearer $TOKEN" -X GET -sS https://$host$rest_api?compartmentId=$tenancy_ocid

```

Get backups

```

#!/bin/bash
##### Fill these in with your values ##### #OCID of the tenancy calls are being
made in to
tenancy_ocid="ocid1.tenancy.XXXX.XXXX.gidclawftnq3vo572t4cb46tmqxmalg4aivv7ttoa1a7y0mibcak0009 0146"

# OCID of the user making the rest call
user_ocid="ocid1.user.XXXX.XXX.zknu1e9y13efvmmpu6doo8arto2eogrds6abbx0zdsygaz9qglksdqxxql0m"

# path to the private PEM format key for this user
privateKeyPath="/home/sampleuser/.oci/oci_api_key.pem"

# fingerprint of the private key for this
user fingerprint="99:aa ..... "

# The REST api you want to call, with any required parameters.
rest_api="/20160918/instanceBackups"

# The PCA you want to make the call against example
iaas.<pcaname>.<domainname> host="iaas.pca.oraclecloud..."

#exported instance id

```

```

export_instance_id="/ocid1.exportinstance.XXXX.XXXX.loikk3scba58td7dkljmnaj1moztqo7ucikr45n9bx
6s6kgzp2do5b559614
/ocid1.exportinstance.XXXX.XXXX.loikk3scba58td7dkljmnaj1moztqo7ucikr45n9bx6s6kgzp2do5b559614"

#data load for token gathering
dataloadlogin="{\"username\": \"admin\", \"password\": \"XXXX\", \"tenancy\":
\"backup_tenant\"}"

# Get login tokens
TOKEN=$(curl --no-proxy "*" -k -XPOST --header "Content-Type: application/json" -d "$dataloadlogin" -sS
https://$host/20160918/login | grep token | awk '{print $2}' | sed -E 's/(^)(.)/g')

# Get the Instance backup echo
"=====
for export_id in $export_instance_id do
    echo "Get $export_id details" set -x
    curl --no-proxy "*" -k -H "AUTHORIZATION: Bearer $TOKEN" -XGET -sS https://$host$rest_api$export_id

```

Single Instance Restore

```

#!/bin/bash
##### Fill these in with your values ##### #OCID of the tenancy calls are being
made in to
tenancy_ocid="ocid1.tenancy.XXXX.XXXX.gidclawftnq3vo572t4cb46tmqxmalg4aivv7ttoa1a7y0mibcak0009 0146"

# OCID of the user making the rest call
user_ocid="ocid1.user.XXXX.XXXX.zknu1e9y13efvmmpu6doo8arto2eogrds6abbx0zdsygz9qglksdqxxql0m"

# path to the private PEM format key for this user
privateKeyPath="/home/sampleuser/.oci/oci_api_key.pem"

# fingerprint of the private key for this
user fingerprint="99:aa ..... "

# The REST api you want to call, with any required parameters.

```

```

rest_api="/20160918/instanceBackups"

# The PCA you want to make the call against example
iaas.<pcaname>.<domainname> host="iaas.pca.oraclecloud..."

# Export action for backup
rest_action="/actions/import"

#exported instance id
exported_instance_id="/ocid1.exportinstance.XXXX.XXXX.uowdqdmY1bw0xsr4tfmmz9srthp9r4ed36esfdz4
ffd5hz9qm2z6hy0431wdv"

#Object Store data for import
objectstore_payload="{\"bucketName\": \"Image\", \"sourceType\": \"objectStorageTuple\",
\"namespaceName\": \"backup_tenant\", \"compartmentId\": \"${tenancy_ocid}\"}"

#data load for token gathering
data_login="{\"username\": \"admin\", \"password\": \"XXXX\", \"tenancy\": \"backup_tenant\"}"

# Get login tokens
TOKEN=$(curl --no-proxy "*" -k -X POST --header "Content-Type: application/json" -d "$data_login" -sS
https://$host/20160918/login | grep token | awk '{print $2}' | sed -E 's/(^)|(..$)//g')

# Importing the Instance echo
"=====

set -x

curl --no-proxy "*" -k -H "AUTHORIZATION: Bearer $TOKEN" -X POST --data "$objectstore_payload"
-sS https://$host$rest_api$exported_instance_id$rest_action

```

Multiple Instance Restore

```
#!/bin/bash
##### Fill these in with your values #####
#OCID of the tenancy calls are being made in to
tenancy_ocid="ocid1.tenancy.XXXX.XXXX.gidclawftnq3vo572t4cb46tmqxmlg4aivv7ttoa1a7y0mibcak0009 0146"
# OCID of the user making the rest call
user_ocid="ocid1.user.XXXX.XXXX.zknu1e9y13efvmmpu6doo8arto2eogrds6abbx0zdsygaz9qgklsdxxxql0m"
# path to the private PEM format key for this user privateKeyPath="/home/sampleuser/.oci/oci_api_key.pem"

# fingerprint of the private key for this user
fingerprint="99:aa....."

# The REST api you want to call, with any required parameters.
rest_api="/20160918/instanceBackups"

# The PCA you want to make the call against example
iaas.<pcaname>.<domainname> host="iaas.pca.oraclecloud. "

# Export action for backup
rest_action="/actions/import"

#exported backup id for backup with space between the backup id
backup_list_id="/ocid1.exporthinstance.XXXX.XXXX.rtzmmjmsee2d5wn6sem4v34zp3zzisyonrrd1sz1224rua
rcd9eof7n9qre7
/ocid1.exporthinstance.XXXX.XXXX.5.nmz962biefic3msq7bz5x7agp5g86eeano2pbbsdspklectzy0a5ozuk3m
"

#Object Store data for import
objectstore_payload="{\"bucketName\": \"Image\", \"sourceType\": \"objectStorageTuple\",
\"namespaceName\": \"backup_tenant\", \"compartmentId\": \"${tenancy_ocid}\"}"

#data load for token gathering
data_login="{\"username\": \"admin\", \"password\": \"XXXX\", \"tenancy\": \"backup_tenant\"}"
# Get login tokens
TOKEN=$(curl --noproxy "*" -k -XPOST --header "Content-Type: application/json" -d "$data_login" -sS
https://$host/20160918/login | grep token | awk '{print $2}' | sed -E 's/(^)|(..$)//g')
# Importing the Instance
```

```

echo
"=====
set -x
for backup_id in $backup_list_id do
    echo "Importing $instance" set -x
    curl --noproxy "*" -k -H "AUTHORIZATION: Bearer $TOKEN" -X POST --data "$subjectstore_payload" -sS
    https://$host$rest_api$backup_id$rest_action
done

```

Deleting the Backups

```

#!/bin/bash
##### Fill these in with your values ##### #OCID of the tenancy calls are being
made in to
tenancy_ocid="ocid1.tenancy.XXXX.XXXX.gidclawftnq3vo572t4cb46tmqxmalg4aivv7ttoa1a7y0mibcak0009 0146"

# OCID of the user making the rest call
user_ocid="ocid1.user.XXXX.XXXX.zknu1e9y13efvmmpu6doo8arto2eogrds6abbx0zdsygaz9qg|ksdqxxq|0m"

# path to the private PEM format key for this user
privateKeyPath="/home/sampleuser/.oci/oci_api_key.pem"

# fingerprint of the private key for this
user fingerprint="99:aa....."

# The REST api you want to call, with any required
parameters. rest_api="/20160918/instanceBackups"

# The PCA you want to make the call against example
iaas.<pcaname>.<domainname> host="iaas.pca.oraclecloud..."

#exported instance id
exported_instance_id="/ocid1.exportinstance.XXXX.XXXX.rtzmmjmsee2d5wn6sem4v34zp3zsisyonrrd1sz1
224ruarcd9eof7n9qre7
/ocid1.exportinstance.XXXX.XXXX.rnmz962biefic3msq7bz5x7agp5g86eeano2pbbsdspklecotzy0a5ozuk3m"

```

```

#data load for token gathering
data_login="{\"username\": \"admin\", \"password\": \"XXXX\", \"tenancy\": \"backup_tenant\"}"

#Get login tokens
TOKEN=$(curl --no-proxy "*" -k -X POST --header "Content-Type: application/json" -d "$data_login" -sS
https://$host/20160918/login | grep token | awk '{print $2}' | sed -E 's/(^.)|(..$)//g')

# Delete Multiple Exports
echo "=====
set -x
for export_id in $exported_instance_id do
    echo "Deleting $export_id" set -x
    curl --no-proxy "*" -k -H "AUTHORIZATION: Bearer $TOKEN" -X DELETE -sS
https://$host$rest_api$export_id
done

```

Launching Instance from Restore

For creating as multiple instances from the instance backup, follow these steps:

- Create an instance in the usual fashion. While doing so, perform these actions:
 - For the image source, specify the backup boot volume.
 - If the instance requires access using SSH, ensure that you include an SSH public key.

(Optional) Attach any block volumes that were included in the instance backup.

```

export cidr_block=<substitute-value-of-cidr_block>
export compartment_id=<substitute-value-of-compartment_id>
export availability_domain=<substitute-value-of-availability_domain> export image_id=<substitute-value-of-
image_id>
export sourcebootvolumeid =<substitute-value-of-sourcebootvolumeid-for-restored-boot-volume>
vcn_id=$(oci network vcn create --cidr-block $cidr_block --compartment-id $compartment_id -- querydata.id --raw-output)
subnet_id=$(oci network subnet create --cidr-block $cidr_block --compartment-id
$compartment_id --vcn-id $vcn_id --query data.id --raw-output)
oci compute instance launch --availability-domain $availability_domain --compartment-id
$compartment_id --source-boot-volume-id $sourcebootvolumeid --shape VM.PCAStandard1.Flex -- subnet-id $subnet_id --shape-config
'{"ocpus": 1, "memoryInGBs": 16}' --ssh-authorized-keys- file<SSH Public Key>--wait-for-state RUNNING

```

DOWNLOAD FROM OBJECT STORE

Transferring an Instance Backup

The procedures in this section describe how to transfer an instance backup to another system and how to transfer an instance backup back to any Private Cloud Appliance 3.x.

Transferring an Instance Backup to Another System

You can use this procedure to transfer the instance backup to another system in your data center for safekeeping. Instance backups are large files. Ensure that you have enough space on your system to store the backup. You can use the `oci os "object list command"` to display the size of the instance backup. See [Viewing Objects in a Bucket](#).

Using the OCI CLI

- Gather the information that you need to run the command:
 - Bucket name:
 - `oci os bucket list --namespace-name <object_storage_namespace> o` Object name:
 - `oci os object list --namespace-name <object_storage_namespace> --bucket-name <backup_object_name>`

- Run this command:

```
oci os object get
--namespace-name <object_storage_namespace>
--bucket-name <bucket_name>
--name <backup_object_name>
--file <instance_backup_pathname_with_filename>
```

Example:

```
oci os object get --namespace-name testnamespace --bucket-name backup --name
ocid1.exportinstance.XXXX.XXXX.1lzndoh06luhko5x5pyb81bba5mt5ncqs5z6uus5oailsa6wji2zkz71cu7 b -
-file backupFile
```

```
Downloading object [#####] 100%
```

Transferring an Instance Backup from Another System to Private Cloud Appliance

Use this procedure to transfer an instance backup from another system in your data center to an Object Storage bucket in Private Cloud Appliance.

Using the OCI CLI

- Gather the information that you need to run the command:
 - Bucket name:
 - `oci os bucket list --namespace-name <object_storage_namespace>`
 - Object name:
 - `oci os object list --namespace-name <object_storage_namespace> --bucket-name <backup_object_name>`
- Upload the instance backup to an Object Storage bucket in the target appliance. Use the `oci os object put` command Syntax (entered on a single line):

```
oci os object put \
--namespace-name <namespace_name> \
--bucket-name <bucket_name> \
--file <instance_backup_pathname_with_filename>
```

Example:

```
oci os object put --file backupFile --bucket-name targetbucket --namespace-name testnamespace
```

Upload ID: 5bde0dcd-14d4-442e-af4c-43ffd65b24a9 Split file into 62

parts for upload.

Uploading object [#####] 100%

```
{
  "etag": "b22a7b46db3203247a9428941ce354d1", "last-
  modified": "2022-10-15T21:33:42",
  "opc-multipart-md5": "WqmysJmye0m+LIEbZC+ow==62"
}
```

CONCLUSION

Following the above steps, Private Cloud Appliance customers can now choose Automated VM Backup while configuring their VM instances.

RESOURCES

See these reference documents for additional information:

- [Oracle Private Cloud Appliance Release Notes](#)
- [Oracle Systems Blog](#)
- [Object Storage](#)

CONNECT WITH US

Call +1.800.ORACLE1 or visit [oracle.com](https://www.oracle.com).

Outside North America, find your local office at [oracle.com/contact](https://www.oracle.com/contact).

 blogs.oracle.com

 facebook.com/oracle

 twitter.com/oracle

Copyright © 2024, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Automating VM Backup on the Oracle Private Cloud Appliance

January 2024

Author: Amardeep Dhillon

