

# How to backup Oracle database to 4 Clouds

Azure, AWS, Google & Oracle

# About me...

Krešimir Škoda

[kskoda@primarykey.hr](mailto:kskoda@primarykey.hr)

Twitter: @kskoda

- Working with Oracle databases since 2004 in various environments and industries, from parking, aviation to finance and banking sectors
- Certified OCI Architect Associate
- Lately working for a client as system engineer and DevOps with Google Cloud and AWS
- Owner of primary key d.o.o., company specialized in database consultancy
- Technology and travel addict who likes cycling

# Why backup to cloud at all?

- Cloud is just some kind of someone else's computer?
- Less or even no care about the infrastructure: you get an "black box" that stores your data
- Responsibility?
- Easier?
- Less costs?
- More secure?
- Decide for yourself and your use-case (or you already have an request)

# To be able to do your backups in the cloud...

- You have a cloud subscription
- You have fast and reliable connection to internet/cloud
- You have some experience working with clouds, especially with cloud storage
- You already have a database server with installed Oracle database, either on-prem, local VM or host/server in the cloud.

# Principles and guidances

- The main principle is to mount/attach cloud storage to your host as FUSE (Filesystem USErspace) and then present this to our database/host as a disk or tape
- We will do backups and restores using RMAN
- In some cases you can also do a Data Pump export and keep it on cloud storage, but we won't focus on that
- Think about encryption:
  - Would you use Oracle's implemented RMAN encryption which requires EE with Advanced Security option
  - All cloud storage providers offer you some encryption, in that case check if your connection to cloud is encrypted as well

# Setup example (host)

For this presentation I used:

- Oracle VirtualBox 6.1, OS Oracle Linux 8.6, Oracle database 19.3
- Root container ORCLCDB and one PDB container ORCLPDB1
- Focus is on Linux, as most of my work with databases is on Linux
- At the end I got a running VM with ability to backup to all of the mentioned clouds

# Setup example (host)

## Created example schema and table:

```
sqlplus / as sysdba
  ALTER SESSION SET CONTAINER=ORCLPDB1;
  create user kskoda identified by "kskoda" default tablespace users;
  grant resource, connect to kskoda;
  alter user kskoda default role all;
  alter user kskoda quota unlimited on users;
conn kskoda/kskoda
  drop table my_table;
  create table kskoda.my_table (
    field1 number(12) constraint pk_my_table primary key,
    field2 varchar2(30),
    field3 date default sysdate) tablespace users;
  insert into kskoda.my_table (field1, field2) values (1, 'value 1');
  insert into kskoda.my_table (field1, field2) values (2, 'value 2');
  commit;
  select * from kskoda.my_table;
```

# Setup example (cloud)

- All clouds accounts are free/trial subscriptions
- Setup and used in August, September and October 2022
- Cloud providers often change their GUI look and feel, it is possible that some screenshots will not be accurate in the near future



# Backup to Azure

- The principle is to create a cloud storage/blob and mount it to our host via Blobfuse
- Blobfuse is an open-source project developed to provide a virtual filesystem backed by Azure Blob storage
- Currently available only for Ubuntu and CentOS/RedHat distributions and its derivatives, like Oracle Linux
- <https://github.com/Azure/azure-storage-fuse>
- For Windows hosts there is Azure Files NFS v4.1 available

# Backup to Azure

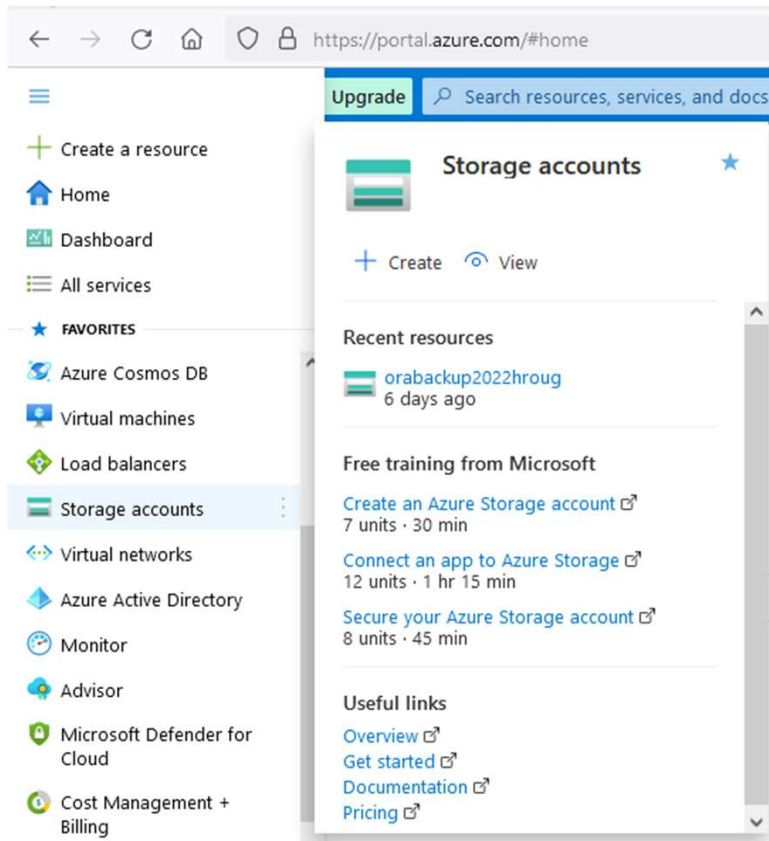
Prepare the cloud

## Create a storage account

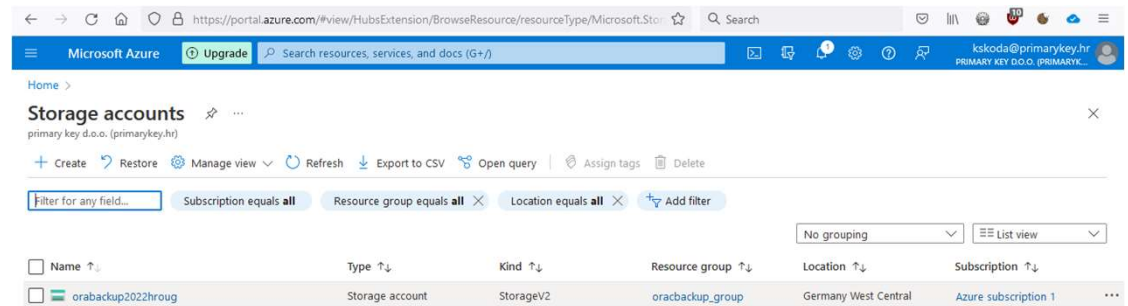
- Login to Azure Portal (<https://portal.azure.com/>)
- Choose "Storage accounts", then "+Create"
  - Select your Subscription plan and create new Resource group (`oracbackup_group`)
  - For Instance details choose your Storage account name (`orabackup2022hroug`), Region closest to your database, Performance type "Standard" and Redundancy type "Geo-redundant storage GRS", which is recommended for backup scenarios
  - Leave other settings default
  - -> Create

# Backup to Azure

## Prepare the cloud (2)



The screenshot shows the Azure portal home page. The left-hand navigation pane is open, and the 'Storage accounts' option is highlighted. The main content area displays the 'Storage accounts' overview, including a search bar, a 'Create' button, and a list of recent resources. The 'Recent resources' section shows a resource named 'orabackup2022hroug' created 6 days ago. Below this, there are links for 'Free training from Microsoft', including 'Create an Azure Storage account', 'Connect an app to Azure Storage', and 'Secure your Azure Storage account'. At the bottom, there are 'Useful links' for 'Overview', 'Get started', 'Documentation', and 'Pricing'.



The screenshot shows the 'Storage accounts' resource page in the Azure portal. The page displays a table of storage accounts with the following columns: Name, Type, Kind, Resource group, Location, and Subscription. The table contains one entry: 'orabackup2022hroug' of type 'Storage account', kind 'StorageV2', resource group 'orabackup\_group', location 'Germany West Central', and subscription 'Azure subscription 1'. The page also includes a search bar, a 'Create' button, and various management options like 'Restore', 'Manage view', 'Refresh', 'Export to CSV', 'Open query', 'Assign tags', and 'Delete'. There are also filters for 'Subscription equals all', 'Resource group equals all', and 'Location equals all'.

Name	Type	Kind	Resource group	Location	Subscription
orabackup2022hroug	Storage account	StorageV2	orabackup_group	Germany West Central	Azure subscription 1

# Backup to Azure

Prepare the cloud (3)

- On "Storage accounts" accounts list click on our newly created account
  - Choose "Data storage", then "Containers", then "+Create"
  - Name your container (orablob1) and "Public access level" Blob
  - -> Create

# Backup to Azure

## Prepare the cloud (4)

The screenshot displays the Microsoft Azure portal interface. The main content area shows the 'Containers' view for the storage account 'orabackup2022hroug'. A search bar is present at the top of the container view. Below it, a table lists the containers:

Name	Last modified	Public access level	Lease state
<input type="checkbox"/> slogs	10/7/2022, 4:20:17 PM	Private	Available
<input type="checkbox"/> orablob1	10/7/2022, 4:33:23 PM	Blob	Available

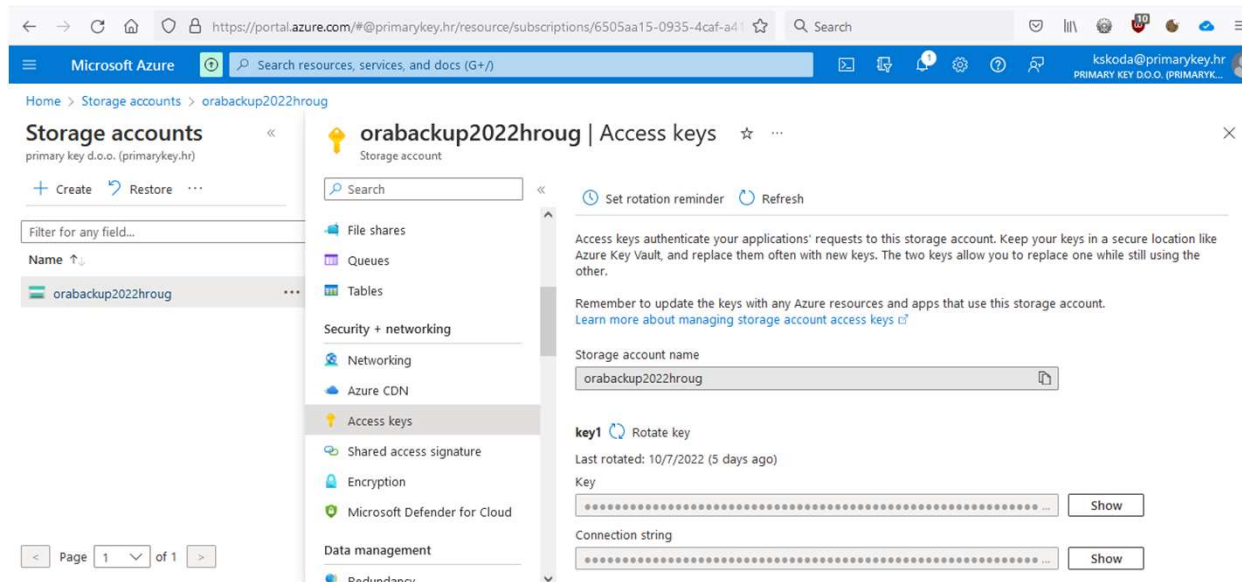
The left navigation pane includes options like 'Overview', 'Activity log', 'Tags', 'Diagnose and solve problems', 'Access Control (IAM)', 'Data migration', 'Events', 'Storage browser', 'Data storage', 'Containers', 'File shares', and 'Queues'. The 'Containers' option is currently selected.

# Backup to Azure

Prepare storage on the cloud

On "Storage accounts" accounts list click on our newly created account

- Choose "Security + networking", then "Access keys"
- Copy key, this will be the "accountKey" property later in `connection.cfg` file



# Backup to Azure

## Prepare the host

- **Install Microsoft repo:**

```
sudo rpm -Uvh https://packages.microsoft.com/config/rhel/8/packages-microsoft-prod.rpm
```

- **Install Blobfuse:**

```
dnf install blobfuse
```

- **Create mountpoints and set ownerships:**

```
mkdir -p /mnt/bkp_azure/orabackup  
mkdir /mnt/blobfusetmp  
chown -R oracle:oinstall /mnt/bkp_azure/  
chown oracle:oinstall /mnt/blobfusetmp/
```

- **Create configuration file for mounting:**

- **As oracle user create file connection.cfg in home directory:**

```
vi ~/connection.cfg  
  
    accountName orabackup2022hroug  
    accountKey VDW11gXKUTCa6aDZ8iUM881DGV0swEE5/1qMpYJ+GBDKVaqwBmUjm1+hpjCG60/yMRp48+9WSHHM+AStSeQCyQ==  
    containerName orablob1  
chmod 600 ~/connection.cfg
```

# Backup to Azure

Prepare the host (2)

- **Mount the storage as oracle user:**

```
blobfuse /mnt/bkp_azure/orabackup \  
  --tmp-path=/mnt/blobfusetmp \  
  --use-attr-cache=true -o attr_timeout=240 \  
  -o entry_timeout=240 -o negative_timeout=120 \  
  --config-file=/home/oracle/connection.cfg
```

- **Check the mount:**

```
df -h | grep blobfuse  
blobfuse          70G   18G   53G   26% /mnt/bkp_azure/orabackup
```



# Backup to Azure

## RMAN backup

### Perform RMAN backup as oracle user:

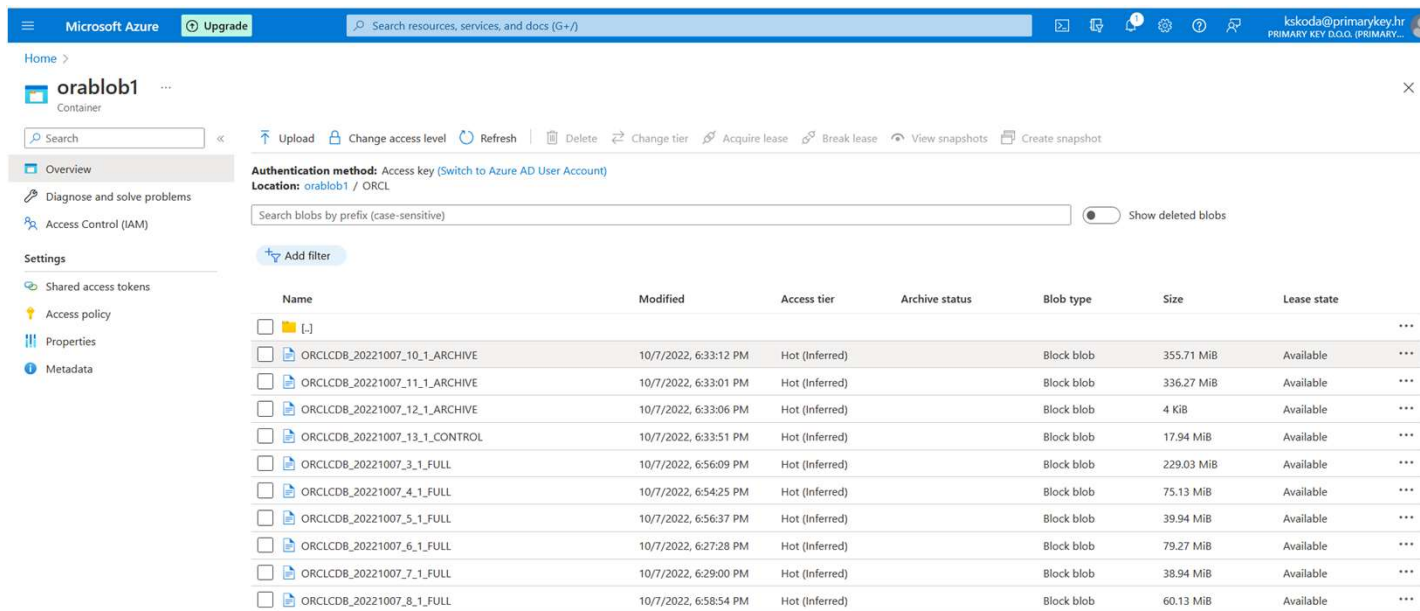
```
. oraenv
  ORCLCDB
export NLS_DATE_FORMAT="yyyy-mm-dd hh24:mi:ss"
rman target /
  sql 'alter system archive log current';
  sql "alter session set nls_date_format='yyyy-mm-dd hh24:mi:ss'";
  run {
  configure controlfile autobackup on;
  allocate channel c1 device type disk;
  allocate channel c2 device type disk;
  backup as compressed backupset full database tag AZURE_ORCL_FULL format
  '/mnt/bkp_azure/orabackup/ORCL/%d_%T_%s_%p_FULL' ;
  sql 'alter system archive log current';
  backup tag AZURE_ORCL_ARCHIVE format '/mnt/bkp_azure/orabackup/ORCL/%d_%T_%s_%p_ARCHIVE'
  archivelog all delete all input ;
  backup tag AZURE_ORCL_CONTROL current controlfile format
  '/mnt/bkp_azure/orabackup/ORCL/%d_%T_%s_%p_CONTROL';
  release channel c1;
  release channel c2;
  }
```

# Backup to Azure

Backup files in cloud

Backup files are visible in:

Azure Portal -> Storage accounts -> orabackup2022hroug -> Data storage -> Containers  
(or in Azure Storage Explorer)



The screenshot displays the Microsoft Azure portal interface for a storage container named 'orablob1'. The top navigation bar includes the Microsoft Azure logo, an 'Upgrade' button, and a search bar. The left sidebar shows navigation options like 'Overview', 'Diagnose and solve problems', 'Access Control (IAM)', and 'Settings'. The main content area shows the container's details, including the authentication method (Access key) and location (orablob1 / ORCL). Below this is a search bar for blobs and a 'Show deleted blobs' toggle. The central part of the screen features a table listing the blobs in the container.

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
[...]						...
ORCLCDB_20221007_10_1_ARCHIVE	10/7/2022, 6:33:12 PM	Hot (Inferred)		Block blob	355.71 MiB	Available
ORCLCDB_20221007_11_1_ARCHIVE	10/7/2022, 6:33:01 PM	Hot (Inferred)		Block blob	336.27 MiB	Available
ORCLCDB_20221007_12_1_ARCHIVE	10/7/2022, 6:33:06 PM	Hot (Inferred)		Block blob	4 KiB	Available
ORCLCDB_20221007_13_1_CONTROL	10/7/2022, 6:33:51 PM	Hot (Inferred)		Block blob	17.94 MiB	Available
ORCLCDB_20221007_3_1_FULL	10/7/2022, 6:56:09 PM	Hot (Inferred)		Block blob	229.03 MiB	Available
ORCLCDB_20221007_4_1_FULL	10/7/2022, 6:54:25 PM	Hot (Inferred)		Block blob	75.13 MiB	Available
ORCLCDB_20221007_5_1_FULL	10/7/2022, 6:56:37 PM	Hot (Inferred)		Block blob	39.94 MiB	Available
ORCLCDB_20221007_6_1_FULL	10/7/2022, 6:27:28 PM	Hot (Inferred)		Block blob	79.27 MiB	Available
ORCLCDB_20221007_7_1_FULL	10/7/2022, 6:29:00 PM	Hot (Inferred)		Block blob	38.94 MiB	Available
ORCLCDB_20221007_8_1_FULL	10/7/2022, 6:58:54 PM	Hot (Inferred)		Block blob	60.13 MiB	Available

# Backup to Azure

## RMAN restore

### Stop Oracle database (as root):

```
/etc/init.d/oracledb_ORCLCDB-19c stop  
echo 0 > /opt/oracle/oradata/ORCLCDB/ORCLPDB1/users01.dbf  
/etc/init.d/oracledb_ORCLCDB-19c start
```

### As oracle user mount the database:

```
sqlplus / as sysdba  
startup mount
```

### Perform RMAN restore:

```
. oraenv  
ORCLCDB  
export NLS_DATE_FORMAT="yyyy-mm-dd hh24:mi:ss"  
rman target /  
run {  
allocate channel c1 device type disk;  
allocate channel c2 device type disk;  
set until scn= 2770796;  
restore database;  
recover database;  
release channel c1;  
release channel c2;  
}  
sqlplus / as sysdba  
alter database open resetlogs;  
alter session set container=ORCLPDB1;  
select * from kskoda.my_table;
```

# Backup to AWS

- The principle is to create a S3 cloud storage/bucket and mount it to our host
- We will use Oracle Secure Backup Module for Amazon S3, which needs Java version 1.7 or higher to be installed
- Alternative option is to use AWS Storage Gateway, a hybrid cloud service that offers cloud storage and mounts it to the host like a NFS share mapped to an S3 bucket
- AWS offers several S3 buckets, differentiated by the speed of getting data for restore

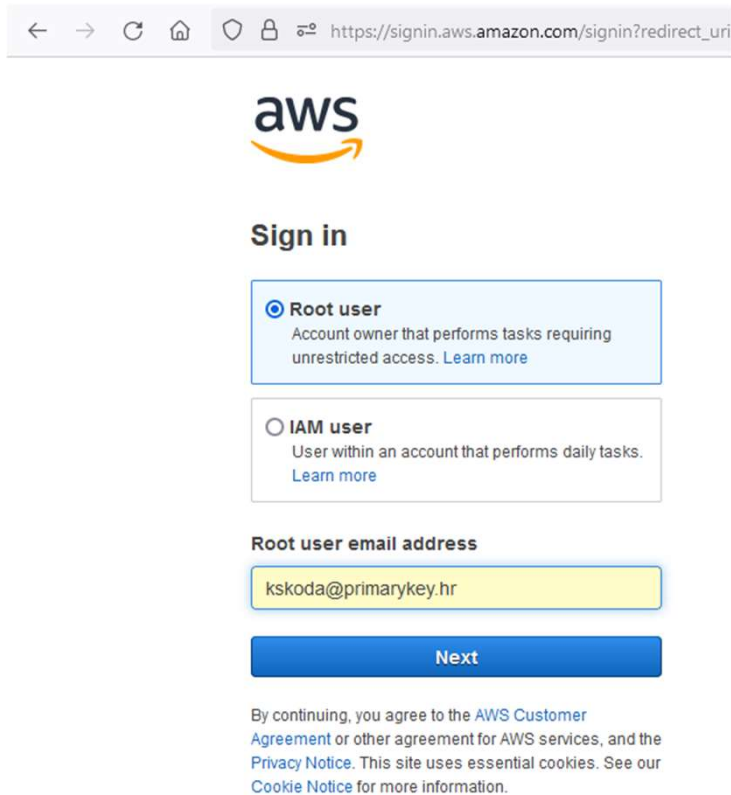
# Backup to AWS

Prepare the cloud

- Login to AWS Console (<https://aws.amazon.com/console/>) with your root account
- From "Services" choose "Security, Identity, & Compliance", then "IAM"
- Create IAM user `orabackup` with access key ID and secret access key, for policies select existing policy `AmazonS3FullAccess`

# Backup to AWS

## Prepare the cloud (2)



A screenshot of the AWS sign-in page. The browser address bar shows the URL `https://signin.aws.amazon.com/signin?redirect_uri`. The page features the AWS logo and a "Sign in" section. Two user types are available: "Root user" (selected) and "IAM user". The "Root user" option is described as the account owner that performs tasks requiring unrestricted access. Below the user type selection, there is a field for "Root user email address" containing the email `kskoda@primarykey.hr`. A blue "Next" button is positioned below the email field. At the bottom of the page, a disclaimer states: "By continuing, you agree to the [AWS Customer Agreement](#) or other agreement for AWS services, and the [Privacy Notice](#). This site uses essential cookies. See our [Cookie Notice](#) for more information."

[https://signin.aws.amazon.com/signin?redirect\\_uri](#)

**aws**

**Sign in**

**Root user**  
Account owner that performs tasks requiring unrestricted access. [Learn more](#)

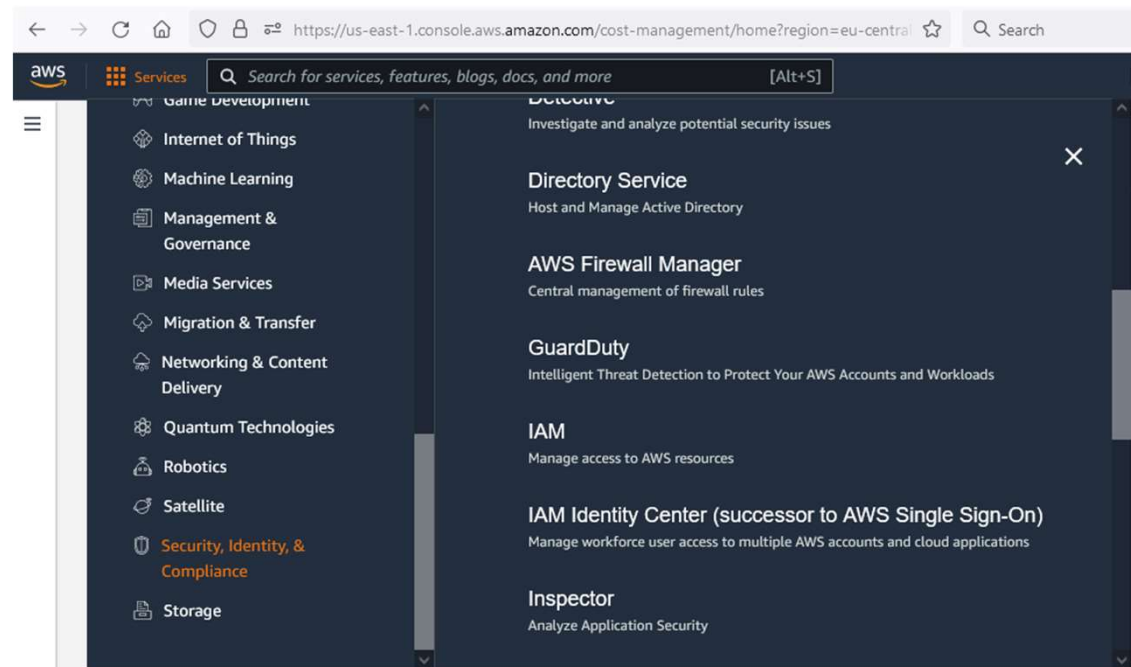
**IAM user**  
User within an account that performs daily tasks. [Learn more](#)

**Root user email address**

`kskoda@primarykey.hr`

**Next**

By continuing, you agree to the [AWS Customer Agreement](#) or other agreement for AWS services, and the [Privacy Notice](#). This site uses essential cookies. See our [Cookie Notice](#) for more information.



# Backup to AWS

## Prepare the cloud (3)

The screenshot shows the AWS IAM console interface. At the top, there is a navigation bar with the AWS logo, a search bar, and the user's name 'primarykeyhr'. Below the navigation bar, a blue banner announces the new Users list experience. The main content area is titled 'IAM > Users' and features a 'Users (1) Info' section with a refresh button, a 'Delete' button, and an 'Add users' button. A search bar is provided to find users by username or access key. Below the search bar is a table with the following columns: 'User name', 'Groups', 'Last activity', 'MFA', and 'Password ...'. The table contains one entry for the user 'orabackup'.

<input type="checkbox"/>	User name	Groups	Last activity	MFA	Password ...
<input type="checkbox"/>	orabackup	None	✓ 4 days ago	None	✓ 4 days ago

# Backup to AWS

Prepare the host

- Install Java as root user:

```
sudo dnf install java
```

- Create Oracle Secure Backup (OSB) directory and set ownerships:

```
mkdir -p /mnt/bkp_aws/osb
```

```
chown -R oracle:oinstall /mnt/bkp_aws
```

- As oracle user download and install Oracle Secure Backup from

[https://download.oracle.com/otn/other/osbws\\_installer.zip](https://download.oracle.com/otn/other/osbws_installer.zip)

- Create wallet directory

```
mkdir ~/owbws_wallet
```



# Backup to AWS

## Prepare the host (2)

- Install OSB:

```
cd /mnt/bkp_aws/osb
unzip ~/Downloads/osbws_installer.zip
java -jar osbws_install.jar \
  -AWSID AKIAW3W4YOQ7PAAXBXR \
  -AWSKey t0wGECd/hY3khn6L5rZd2FgBLKof06z6ogYhLyeb \
  -walletDir /home/oracle/osbws_wallet \
  -libDir $ORACLE_HOME/lib \
  -location eu-central-1 \
  -awsEndPoint s3-eu-central-1.amazonaws.com \
  -otnUser kskoda@primarykey.hr \
  -otnPass password
```

Oracle Secure Backup Web Service Install Tool, build 12.2.0.1.0DBBKPCSBP\_2018-06-12

AWS credentials are valid.

Oracle Secure Backup Web Service wallet created in directory /home/oracle/osbws\_wallet.

Oracle Secure Backup Web Service initialization file /opt/oracle/product/19c/dbhome\_1/dbs/osbwsORCLCDB.ora created.

Downloading Oracle Secure Backup Web Service Software Library from file osbws\_linux64.zip.

Download complete.

# Backup to AWS

Prepare the host (2)

- Installation creates an `osb<ORACLE_SID>.ora` file with the following contents:

```
cd $ORACLE_HOME/dbs
less osbwsORCLCDB.ora
OSB_WS_HOST=http://s3-eu-central-1.amazonaws.com
OSB_WS_LOCATION=eu-central-1
OSB_WS_WALLET='location=file:/home/oracle/osbws_wallet CREDENTIAL_ALIAS=s3_aws'
```

# Backup to AWS

## RMAN backup

### Perform RMAN backup as oracle user:

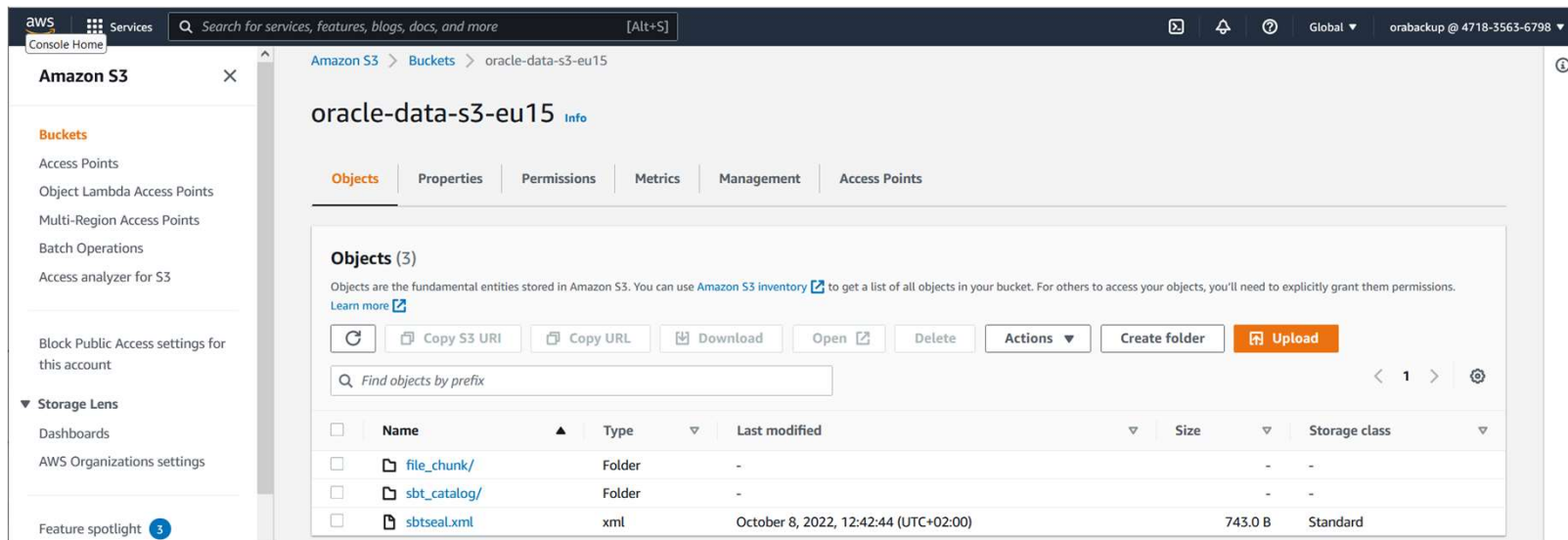
```
. oraenv
  ORCLCDB
export NLS_DATE_FORMAT="yyyy-mm-dd hh24:mi:ss"
rman target /
  sql 'alter system archive log current';
  run {
  configure controlfile autobackup on;
  allocate channel c1 device type sbt
  parms='SBT_LIBRARY=/opt/oracle/product/19c/dbhome_1/lib/libosbws.so,SBT_PARMS=(OSB_W
  S_PFILE=/opt/oracle/product/19c/dbhome_1/dbs/osbwSORCLCDB.ora) ' ;
  allocate channel c2 device type sbt
  parms='SBT_LIBRARY=/opt/oracle/product/19c/dbhome_1/lib/libosbws.so,SBT_PARMS=(OSB_W
  S_PFILE=/opt/oracle/product/19c/dbhome_1/dbs/osbwSORCLCDB.ora) ' ;
  backup as compressed backupset full database tag AWS_ORCL_FULL ;
  sql 'alter system archive log current' ;
  backup tag AWS_ORCL_ARCHIVE archivelog all delete all input ;
  backup tag AWS_ORCL_CONTROL current controlfile ;
  release channel c1;
  release channel c2;
  }
```

# Backup to AWS

Backup files in cloud

Backup files are visible in:

AWS Console -> login with orabackup user -> Services -> Storage -> S3 -> Buckets



# Backup to AWS

## Backup files in cloud (2)

The screenshot shows the AWS S3 console interface. The breadcrumb navigation path is: Amazon S3 > Buckets > oracle-data-s3-eu15 > file\_chunk/ > 2857570486/ > ORCLCDB/ > backuppiece/ > 2022-10-08/ > Of19ools\_1\_1/ > 4gNQBhF6KmEC/. The bucket name 4gNQBhF6KmEC is displayed prominently. A 'Copy S3 URI' button is visible in the top right. The 'Objects' tab is selected, showing a list of two objects:

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	0000000001	-	October 8, 2022, 13:13:09 (UTC+02:00)	228.8 MB	Standard
<input type="checkbox"/>	metadata.xml	xml	October 8, 2022, 13:13:54 (UTC+02:00)	1.7 KB	Standard

Below the table, there is a search bar with the placeholder text 'Find objects by prefix' and a pagination control showing '1' of 1 objects.

# Backup to AWS

## RMAN restore

### Stop Oracle database (as root):

```
/etc/init.d/oracledb_ORCLCDB-19c stop
echo 0 > /opt/oracle/oradata/ORCLCDB/ORCLPDB1/users01.dbf
/etc/init.d/oracledb_ORCLCDB-19c start
```

### As oracle user mount the database:

```
sqlplus / as sysdba
startup mount
```

### Perform RMAN restore:

```
. oraenv
ORCLCDB
export NLS_DATE_FORMAT="yyyy-mm-dd hh24:mi:ss"
rman target /
run {
  allocate channel c1 device type sbt
  parms='SBT_LIBRARY=/opt/oracle/product/19c/dbhome_1/lib/libosbws.so,SBT_PARMS=(OSB_WS_PFILE=/opt/oracle/product/19c/dbhome_1/dbs/osbwsORCLCDB.ora)' ;
  allocate channel c2 device type sbt
  parms='SBT_LIBRARY=/opt/oracle/product/19c/dbhome_1/lib/libosbws.so,SBT_PARMS=(OSB_WS_PFILE=/opt/oracle/product/19c/dbhome_1/dbs/osbwsORCLCDB.ora)' ;
  restore database;
  recover database;
  release channel c1;
  release channel c2;
}
sqlplus / as sysdba
alter database open;
alter session set container=ORCLPDB1;
select * from kskoda.my_table;
```

# Backup to Google Cloud

- The principle is to create a GCS bucket (Google Cloud Storage bucket) and mount it to our host via gcsfuse
- gcsfuse is an open-source project FUSE adapter that allows you to mount a virtual filesystem backed by Google Cloud Storage
- <https://github.com/GoogleCloudPlatform/gcsfuse>
- Very similar to backup on Azure

# Backup to Google Cloud

Prepare the cloud

## Create a storage account

- Login to Google Cloud Console (<https://console.cloud.google.com/>)
- Choose "IAM & Admin" -> "Service accounts", then "+CREATE SERVICE ACCOUNT"
  - Create `orabackup2022hroug` and select role "Storage Admin"
  - Click on newly created account and on "KEYS" tab choose "ADD KEY" option
  - The key should be JSON type, and automatically downloaded through your browser
  - Save the downloaded JSON file to oracle account on your host under `path/name /home/oracle/gcs_key.json`



# Backup to Google Cloud

## Prepare the cloud (2)

A screenshot of the Google Cloud console navigation menu. The menu is open, showing various options. The 'IAM & Admin' option is highlighted at the bottom. The menu items include: IAM, Identity & Organization, Policy Troubleshooter (with a 'NEW' badge), Policy Analyzer, Organization Policies, Service Accounts, Workload Identity Federation, Labels, Tags, Settings, Privacy & Security, Identity-Aware Proxy, Roles, Audit Logs, and Manage Resources. The left sidebar shows 'Google Cloud' logo, 'Cloud overview', 'View all products', 'PINNED' section with 'Pin your top products here', and 'MORE PRODUCTS' section with 'Marketplace', 'Billing', 'APIs & Services', 'Support', and 'IAM & Admin'.

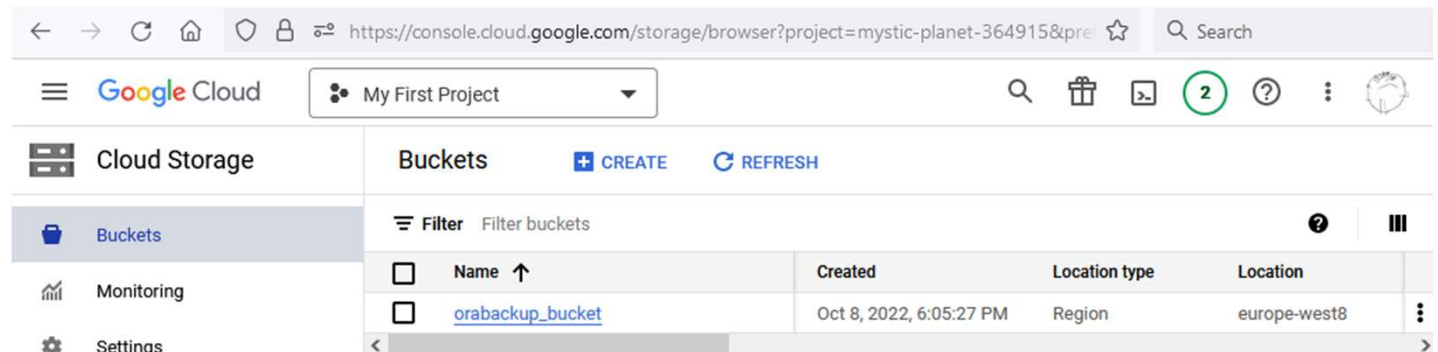
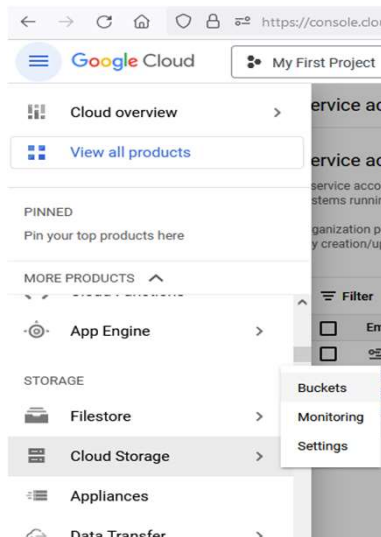
A screenshot of the Google Cloud IAM & Admin console. The page title is 'Service accounts' for 'My First Project'. The main content area shows 'Service accounts for project "My First Project"'. Below this, there is a table of service accounts. The table has columns for 'Email', 'Status', 'Name', and 'Description'. One service account is listed: 'orabackup2022hroug' with a status of '✓' and a description that includes the email address 'orabackup2022hroug@mystic-planet-364915.iam.gserviceaccount.com'. The left sidebar shows the 'IAM & Admin' menu with 'Service Accounts' selected.

<input type="checkbox"/>	Email	Status	Name ↑	Description
<input type="checkbox"/>	<a href="mailto:orabackup2022hroug@mystic-planet-364915.iam.gserviceaccount.com">orabackup2022hroug@mystic-planet-364915.iam.gserviceaccount.com</a>	✓	orabackup2022hroug	

# Backup to Google Cloud

Prepare the cloud (3)

- Choose "Cloud Storage" -> "Buckets", then "+CREATE"
  - Name for the bucket: `orabackup_bucket`
  - Location: region closest to your location
  - Choose a default storage class for your data: Nearline (for backups)



# Backup to Google Cloud

Prepare the host

- **Setup repo and install gcsfuse:**

```
vi /etc/yum.repos.d/gcsfuse.repo
[gcsfuse]
name=gcsfuse (packages.cloud.google.com)
baseurl=https://packages.cloud.google.com/yum/repos/gcsfuse-el7-x86_64
enabled=1
gpgcheck=1
repo_gpgcheck=0
gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg
      https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
```

- **Install gcsfuse:**

```
dnf install gcsfuse
```

- **Create mountpoints and set ownerships:**

```
mkdir -p /mnt/bkp_google/orabackup
chown -R oracle:oinstall /mnt/bkp_google/
```

# Backup to Google Cloud

Prepare the host (2)

- **Mount the storage as oracle user:**

```
export GOOGLE_APPLICATION_CREDENTIALS=/home/oracle/gcs_key.json  
gcsfuse orabackup_bucket /mnt/bkp_google/orabackup
```

- **Check:**

```
df -h | grep orabackup_bucket  
orabackup_bucket      1.0P      0  1.0P    0% /mnt/bkp_google/orabackup
```

# Backup to Google Cloud

## RMAN backup

### Perform RMAN backup as oracle user:

```
. oraenv
  ORCLCDB
mkdir /mnt/bkp_google/orabackup/ORCL
export NLS_DATE_FORMAT="yyyy-mm-dd hh24:mi:ss"
rman target /
    sql 'alter system archive log current';
    run {
        configure controlfile autobackup on;
        allocate channel c1 device type disk;
        allocate channel c2 device type disk;
        backup as compressed backupset full database tag GOOGLE_ORCL_FULL format
        '/mnt/bkp_google/orabackup/ORCL/%d_%T_%s_%p_FULL' ;
        sql 'alter system archive log current';
        backup tag GOOGLE_ORCL_ARCHIVE format '/mnt/bkp_google/orabackup/ORCL/%d_%T_%s_%p_ARCHIVE'
        archivelog all delete all input ;
        backup tag GOOGLE_ORCL_CONTROL current controlfile format
        '/mnt/bkp_google/orabackup/ORCL/%d_%T_%s_%p_CONTROL';
        release channel c1;
        release channel c2;
    }
```

# Backup to Google Cloud

RMAN backup

Backup files are visible in:

Google Cloud Console -> Cloud Storage -> Buckets -> orabackup\_bucket

The screenshot shows the Google Cloud Console interface for the 'orabackup\_bucket'. The breadcrumb navigation is 'Buckets > orabackup\_bucket > ORCL'. The bucket details section shows the following metadata:

Location	Storage class	Public access	Protection
eu-west-8 (Milan)	Nearline	Not public	None

Below the metadata, there are tabs for 'OBJECTS', 'CONFIGURATION', 'PERMISSIONS', 'PROTECTION', and 'LIFECYCLE'. The 'OBJECTS' tab is active, showing a list of objects with the following columns: Name, Size, Type, Created, Storage class, Last modified, Public access, Version history, Encryption, Retention expiration date, and Holds. The objects listed are:

Name	Size	Type	Created	Storage class	Last modified	Public access	Version history	Encryption	Retention expiration date	Holds
ORLCDB_20221008_57_1_FULL	229 MB		Oct 8, 202...	Nearline	Oct 8, 202...	Not public	—	Google-managed key	—	None
ORLCDB_20221008_58_1_FULL	76.4 MB		Oct 8, 202...	Nearline	Oct 8, 202...	Not public	—	Google-managed key	—	None
ORLCDB_20221008_59_1_FULL	40 MB		Oct 8, 202...	Nearline	Oct 8, 202...	Not public	—	Google-managed key	—	None
ORLCDB_20221008_60_1_FULL	79.3 MB		Oct 8, 202...	Nearline	Oct 8, 202...	Not public	—	Google-managed key	—	None
ORLCDB_20221008_61_1_FULL	38.9 MB		Oct 8, 202...	Nearline	Oct 8, 202...	Not public	—	Google-managed key	—	None
ORLCDB_20221008_62_1_FULL	60.2 MB		Oct 8, 202...	Nearline	Oct 8, 202...	Not public	—	Google-managed key	—	None
ORLCDB_20221008_64_1_ARCH...	11.9 MB		Oct 8, 202...	Nearline	Oct 8, 202...	Not public	—	Google-managed key	—	None
ORLCDB_20221008_65_1_ARCH...	159 KB		Oct 8, 202...	Nearline	Oct 8, 202...	Not public	—	Google-managed key	—	None
ORLCDB_20221008_66_1_ARCH...	6 KB		Oct 8, 202...	Nearline	Oct 8, 202...	Not public	—	Google-managed key	—	None
ORLCDB_20221008_69_1_CONT...	18 MB		Oct 8, 202...	Nearline	Oct 8, 202...	Not public	—	Google-managed key	—	None

# Backup to Google Cloud

## RMAN restore

### Stop Oracle database (as root):

```
/etc/init.d/oracledb_ORCLCDB-19c stop  
echo 0 > /opt/oracle/oradata/ORCLCDB/ORCLPDB1/users01.dbf  
/etc/init.d/oracledb_ORCLCDB-19c start
```

### As oracle user mount the database:

```
sqlplus / as sysdba  
startup mount
```

### Perform RMAN restore:

```
. oraenv  
ORCLCDB  
export NLS_DATE_FORMAT="yyyy-mm-dd hh24:mi:ss"  
rman target /  
run {  
allocate channel c1 device type disk;  
allocate channel c2 device type disk;  
restore database;  
recover database;  
release channel c1;  
release channel c2;  
}  
sqlplus / as sysdba  
alter database open resetlogs;  
alter session set container=ORCLPDB1;  
select * from kskoda.my_table;
```

# Backup to Oracle Cloud (OCI)

- There are two common ways to backup Oracle database to OCI:
  - 1) Same as other clouds: create cloud storage object, mount it and do the backups as on disk
  - 2) Use Oracle Database Cloud Backup Module
- As the first way is somewhat same as previous cloud backups, we will focus on backup using Oracle Database Cloud Backup Module
- Backups must be encrypted through RMAN



# Backup to Oracle Cloud (OCI)

Prepare the host

- Download Oracle Cloud Backup Module from the link <https://www.oracle.com/database/technologies/oracle-cloud-backup-downloads.html> as oracle user or alternatively:

```
cd ~/Downloads
wget https://objectstorage.us-ashburn-1.oraclecloud.com/p/VEKec7t0mGwBkJX92Jn0nMptuXIIEpJ5XJA-A6C9PymRgY2LhKbjWqHeB5rVBbaV/n/c4u04/b/live-labs-files/o/data-management-library-files/opc_installer.zip
unzip opc_installer.zip; cd opc_installer/
cd oci_installer/
```

- Check that Java is 1.7 or higher version:

```
java -version
openjdk version "1.8.0_345"
OpenJDK Runtime Environment (build 1.8.0_345-b01)
OpenJDK 64-Bit Server VM (build 25.345-b01, mixed mode)
```

# Backup to Oracle Cloud (OCI)

Prepare the host (2)

For the Oracle Database Backup Cloud Service, you need to have the identifiers and credentials below. You will need an OCI user able to call APIs with these credentials

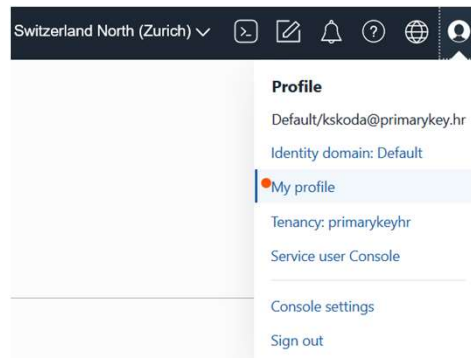
```
mkdir ~/.oci
# Generate the private key:
openssl genrsa -out ~/.oci/oci_api_key.pem 2048
chmod go-rwx ~/.oci/oci_api_key.pem
# Generate the public key:
openssl rsa -pubout -in ~/.oci/oci_api_key.pem -out
~/.oci/oci_api_key_public.pem
# public key:
cat ~/.oci/oci_api_key_public.pem
# fingerprint:
openssl rsa -pubout -outform DER -in ~/.oci/oci_api_key.pem |
openssl md5 -c
```

# Backup to Oracle Cloud (OCI)

Prepare the cloud

## Setup your OCI account

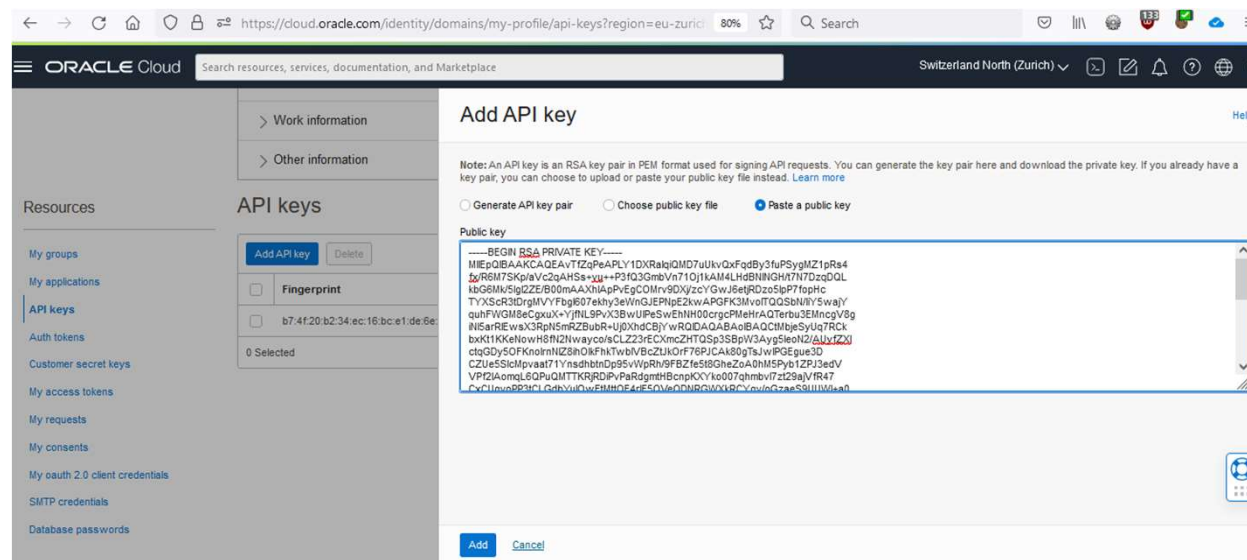
- Login to OCI Console (<https://cloud.oracle.com/>)
- From your profile choose "My profile":



# Backup to Oracle Cloud (OCI)

## Prepare the cloud (2)

- Under Resources, choose "API keys", then "Add API key", and paste the public key from slide before (contents of the file `oci_api_key_public.pem`)
- Copy the user OCID from the "User information" tab



# Backup to Oracle Cloud (OCI)

## Prepare the host (3)

- Syntax for installing the backup module:

```
java -jar oci_install.jar -host <host> -pvtKeyFile ~/.oci/oci_api_key.pem \  
-pubFingerPrint <fingerprint> -uOCID <user-ocid> -tOCID <tenancy-ocid> \  
-cOCID <compartment-ocid> -walletDir $ORACLE_HOME/dbs/oci_wallet \  
-libDir $ORACLE_HOME/lib -bucket db_backups
```

- Parameters explanation can be found in below table:

Parameter	Description	Required or Optional
-host	End point of the Oracle Cloud Infrastructure account. Constructed as <code>https://objectstorage.&lt;region&gt;.oraclecloud.com</code> . See <a href="#">this reference</a> for the region codes.	Required
-vtKeyFile	File that contains the private key used to authenticate Oracle Cloud Infrastructure API requests. The key file must be in PEM format.	Required
-pubFingerPrint	Finger print of the public key paired with the specified private key.	Required
-tOCID	Tenancy OCID for the Oracle Cloud Infrastructure account.	Required
-uOCID	User OCID for the Oracle Cloud Infrastructure account.	Required
-cOCID	Resource compartment ID for the Oracle Cloud Infrastructure account.	Required
-bucket	Name of the bucket in which backups are stored. If this bucket does not exist, then the installer creates it. When this parameter is omitted, a default bucket is automatically created to store backups	Optional
-walletDir	Directory in which Oracle Cloud Infrastructure Object Storage credentials are stored.	Required
-libDir	Directory in which the system backup to tape (SBT) library used for backups and restores with Oracle Cloud Infrastructure is stored.	Required

# Backup to Oracle Cloud (OCI)

## Prepare the host (4)

- As oracle user install the backup module:

```
. oraenv
  ORCLCDB
cd ~/Downloads/opc_installer/oci_installer
java -jar oci_install.jar -host https://objectstorage.eu-zurich-1.oraclecloud.com \
  -pvtKeyFile ~/.oci/oci_api_key.pem \
  -pubFingerPrint b7:4f:20:b2:34:ec:16:bc:e1:de:6e:4b:7e:01:ed:f5 \
  -uOCID ocid1.user.oc1..aaaaaaaadzekvu26xktbewu2tsnx7mimxwwpkxasf46nmrgdrnbd2apofi5a \
  -tOCID ocid1.tenancy.oc1..aaaaaaaagbho7w54bioojkg3kea5epee5oillw3ixp6qmepogaqjfkf2szf3q \
  -cOCID ocid1.tenancy.oc1..aaaaaaaagbho7w54bioojkg3kea5epee5oillw3ixp6qmepogaqjfkf2szf3q \
  -walletDir $ORACLE_HOME/dbs/oci_wallet -libDir $ORACLE_HOME/lib -bucket db_backups \
  -configFile $ORACLE_HOME/dbs/opcORCL.ora
```

```
[oracle@localhost oci_installer]$ java -jar oci_install.jar -host https://objectstorage.eu-zurich-1.oraclecloud.com -pvtKeyFile ~/.oci/oci_api_key.pem -pubFingerPrint b7:4f:20:b2:34:ec:16:bc:e1:de:6e:4b:7e:01:ed:f5 -uOCID ocid1.user.oc1..aaaaaaaadzekvu26xktbewu2tsnx7mimxwwpkxasf46nmrgdrnbd2apofi5a -tOCID ocid1.tenancy.oc1..aaaaaaaagbho7w54bioojkg3kea5epee5oillw3ixp6qmepogaqjfkf2szf3q -cOCID ocid1.tenancy.oc1..aaaaaaaagbho7w54bioojkg3kea5epee5oillw3ixp6qmepogaqjfkf2szf3q -walletDir $ORACLE_HOME/dbs/oci_wallet -libDir $ORACLE_HOME/lib -bucket db_backups -configFile $ORACLE_HOME/dbs/opcORCL.ora
Oracle Database Cloud Backup Module Install Tool, build 19.3.0.0.DBKPCSBP_2019-10-16
Oracle Database Cloud Backup Module credentials are valid.
Backups would be sent to bucket db_backups.
Oracle Database Cloud Backup Module wallet created in directory /opt/oracle/product/19c/dbhome_1/dbs/oci_wallet.
Oracle Database Cloud Backup Module initialization file /opt/oracle/product/19c/dbhome_1/dbs/opcORCL.ora created.
Downloading Oracle Database Cloud Backup Module Software Library from Oracle Cloud Infrastructure.
Download complete.
[oracle@localhost oci_installer]$
```

# Backup to Oracle Cloud (OCI)

Prepare the host (5)

- Check if installation went well:

```
ls $ORACLE_HOME/lib/libopc.so
    /opt/oracle/product/19c/dbhome_1/lib/libopc.so
ls $ORACLE_HOME/dbs/opc*
    /opt/oracle/product/19c/dbhome_1/dbs/opcORCL.ora
ls $ORACLE_HOME/dbs/oci_wallet
    cwallet.sso  cwallet.sso.lck
```

- Now bounce the database (as root):

```
/etc/init.d/oracledb_ORCLCDB-19c stop
/etc/init.d/oracledb_ORCLCDB-19c start
```

# Backup to Oracle Cloud (OCI)

RMAN backup

## Configure RMAN backup:

```
. oraenv
  ORCLCDB
export NLS_DATE_FORMAT="yyyy-mm-dd hh24:mi:ss"
rman target /
  configure encryption for database on;
  configure compression algorithm 'MEDIUM' as
    of release 'default'
  optimize for load true;
```



# Backup to Oracle Cloud (OCI)

## RMAN backup (2)

Perform RMAN backup:

```
rman target /
  set encryption on identified by 'myEncPassword' only;
  sql 'alter system archive log current';
  run {
    configure controlfile autobackup on;
    allocate channel c1 device type sbt tape
    parms='SBT_LIBRARY=/opt/oracle/product/19c/dbhome_1/lib/libopc.so,SBT_PARMS=(
      OPC_PFILE=/opt/oracle/product/19c/dbhome_1/dbs/opcORCL.ora)';
    allocate channel c2 device type sbt tape p
    arms='SBT_LIBRARY=/opt/oracle/product/19c/dbhome_1/lib/libopc.so,SBT_PARMS=(
      OPC_PFILE=/opt/oracle/product/19c/dbhome_1/dbs/opcORCL.ora)';
    backup as compressed backupset full database tag OCI_ORCL_FULLL ;
    sql 'alter system archive log current' ;
    backup tag OCI_ORCL_ARCHIVE archivelog all delete all input ;
    backup tag OCI_ORCL_CONTROL current controlfile ;
    release channel c1;
    release channel c2;
  }
```

# Backup to Oracle Cloud (OCI)

RMAN backup

Backup files are visible in:

OCI Cloud Console -> Object Storage -> Buckets -> db\_backups

The screenshot displays the Oracle Cloud Console interface for a bucket named 'db\_backups'. The page is divided into several sections:

- Header:** 'ORACLE Cloud' logo and 'API Keys' search bar.
- Navigation:** 'Object Storage > Bucket Details' breadcrumb.
- Alert:** A blue notification bar stating: 'You can use 10 GiB of Object Storage and 10 GiB of Archive Storage for free in your home region. You are using approximately 0 bytes of combined Object Storage and Archive Storage. If you use more than 20 GiB of combined Object Storage and Archive Storage, you will be charged for the additional storage used.' A green 'B' icon is visible on the left.
- Bucket Name:** 'db\_backups' with action buttons: 'Edit Visibility', 'Move Resource', 'Re-encrypt', 'Add Tags', and 'Delete'.
- Bucket Information:** A tabbed section with 'General' and 'Tags' tabs.
  - General:**
    - Namespace: zrywuoapuhp
    - Compartment: primarykeyhr
    - Created: Sat, Oct 8, 2022, 19:43:52 UTC
    - ETag: 355baa22-edb6-4bed-98df-4438246143be
    - OCID: ...j177ega [Show](#) [Copy](#)
  - Usage:**
    - Approximate Object Count: 6 objects
    - Approximate Size: 70.51 MB
    - Uncommitted Multipart Uploads Count: 1 uploads
    - Uncommitted Multipart Uploads Approximate Size: 100 MB
- Features:**
  - Default Storage Tier: Standard
  - Visibility: Private
  - Encryption Key: Oracle managed key [Assign](#)
  - Auto-Tiering: Disabled [Edit](#)
  - Emit Object Events: Disabled [Edit](#)
  - Object Versioning: Disabled [Edit](#)

- Resources:** A sidebar menu with 'Objects' selected, and links for 'Metrics', 'Pre-Authenticated Requests', 'Work Requests', 'Lifecycle Policy Rules', and 'Replication Policy'.
- Objects:** A table listing the contents of the bucket.

<input type="checkbox"/>	Name	Last Modified	Size
<input type="checkbox"/>	> file_chunks	-	-
<input type="checkbox"/>	> heartbeat	-	-
<input type="checkbox"/>	> sdt_catalog	-	-

# Backup to Oracle Cloud (OCI)

## RMAN restore

### Stop Oracle database (as root):

```
/etc/init.d/oracledb_ORCLCDB-19c stop
echo 0 > /opt/oracle/oradata/ORCLCDB/ORCLPDB1/users01.dbf
/etc/init.d/oracledb_ORCLCDB-19c start
```

### As oracle user mount the database:

```
sqlplus / as sysdba
startup mount
```

### Perform RMAN restore:

```
. oraenv
ORCLCDB
export NLS_DATE_FORMAT="yyyy-mm-dd hh24:mi:ss"
rman target /
    set encryption on identified by 'myEncPassword' only;
    sql 'alter system archive log current';
    run {
        allocate channel c1 device type sbt tape
        parms='SBT_LIBRARY=/opt/oracle/product/19c/dbhome_1/lib/libopc.so,SBT_PARMS=(OPC_PFILE=/opt/oracle/product/19c/dbhome_1/dbs/opcORCL.ora)';
        allocate channel c2 device type sbt tape
        parms='SBT_LIBRARY=/opt/oracle/product/19c/dbhome_1/lib/libopc.so,SBT_PARMS=(OPC_PFILE=/opt/oracle/product/19c/dbhome_1/dbs/opcORCL.ora)';
        restore database from tag=OCI_ORCL_FULL;
        sql 'alter system archive log current';
        backup tag OCI_ORCL_ARCHIVE archivelog all delete all input;
        backup tag OCI_ORCL_CONTROL current controlfile;
    }
sqlplus / as sysdba
alter database open resetlogs;
alter session set container=ORCLPDB1;
select * from kskoda.my_table;
```

# Bonus: backup to your own cloud: Nextcloud

- What is Nextcloud?
- It has advantages if:
  - You don't trust any of the mentioned cloud providers
  - You already manage your own Nextcloud instance
  - You don't want to pay for cloud storage? (Calculate your TCO first)
- Disadvantages:
  - You have to backup to local disk, and then use Nextcloud client to upload your backupsets, so the local disk space for RMAN backups needs to be the same as in Nextcloud
  - Managing your own Nextcloud instance requires additional administration, and it is also on some kind of OS with additional maintaining

# Cost comparison

The prices can vary and depend on many factors, and without exact knowing of what you need the calculation is hard to make.

To name a few factors that impact the price: type of storage, number of requests to data, amount of retrieval, region, redundancy, etc.

- AWS
  - Cost estimator: <https://calculator.aws>
  - Example: ~11 \$/monthly for 1 TB for S3 Glacier type storage in Frankfurt
- Azure
  - Cost estimator: <https://azure.microsoft.com/en-us/pricing/calculator/>
  - Example: ~67 \$/monthly for 1 TB of archive type storage with 500 GB of data retrieval
- Google Cloud
  - Cost estimator: <https://cloud.google.com/products/calculator/>
  - Example: ~50 \$/monthly for 1 TB of storage in Europe with 500 GB of data retrieval
- OCI
  - Cost estimator: <https://www.oracle.com/cloud/costestimator.html>
  - Example: Standard type storage bucket: ~27 €/monthly for 1 TB for 1000 requests, infrequent type storage bucket: ~14 €/monthly with 512 GB of data retrieval, archive type storage bucket: ~2,5 €/monthly for 1 TB

# Conclusions and Q&A

- Lot of links on the net, with cloud provider's official guidances
- My opinion: if you have an advantage to choose, go with OCI (at least their marketing promises they will always be the cheapest)
- Can I recommend backup to cloud?