GLOC

Great Lakes Oracle Conference

AXXANA

BUILT TO LAST

Deploying Hybrid Data Guard on Oracle Cloud

ORACLE Gold Partner Cloud Standard

ORACLE DATABASE READY

ORACLE LINUX READY

ORACLE BIG DATA APPLIANCE READY

ORACLE EXADATA READY

Speaker

- Yossi Nixon
- Chief Database Architect - Axxana
- Oracle DBA since 1998
- Speaker: IOUG, ILOUG, RMOUG, GLOC, KSCOPE, NYOUG
- Specializes in Data Guard, RAC, performance tuning

@YossiNixon

www.linkedin.com/in/ynixon

oracledba.blogspot.com

http://www.axxana.com

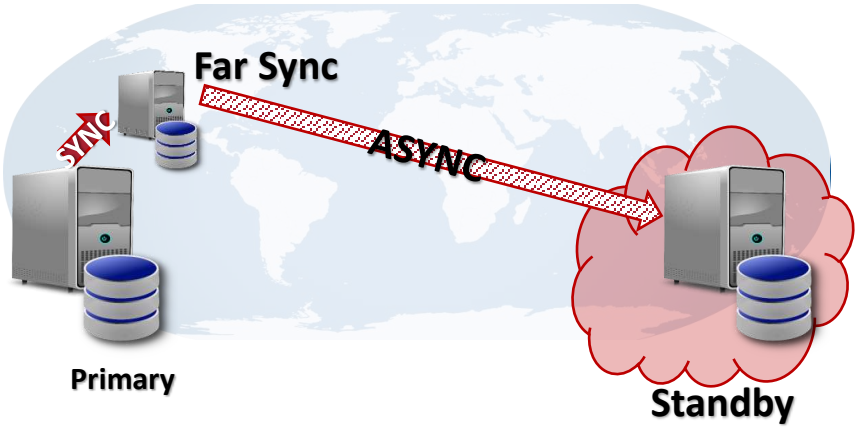
ORACLE ACE Associate

Agenda

- Introduction
- Disaster recovery on the Oracle Cloud
- Environment Prerequisites
- Deployment Process
- Active Data Guard and Far Sync
- Risk distance and a solution
- Conclusion

Introduction

- Data Guard and Active Data Guard provides
  - **Data** protection - Recovery Point Objective (RPO)
  - Availability - Recovery **time** objectives (RTO)
- Why to use Data on the cloud?
  - Disaster recovery plan is costly
  - Existing production databases remain on-premises
  - Standby databases can be used for online reporting, test & development



hybrid cloud implementation.

Hybrid DR to the Oracle Cloud

On-Premises (Production)

Clients

Applications

Java

Databases

Firewall

Active Data Guard
(encrypted)

Standby Databases on Public Cloud

Oracle Public Cloud

VPN
Gateway

Reporting

Sandbox Test/Dev
in the cloud

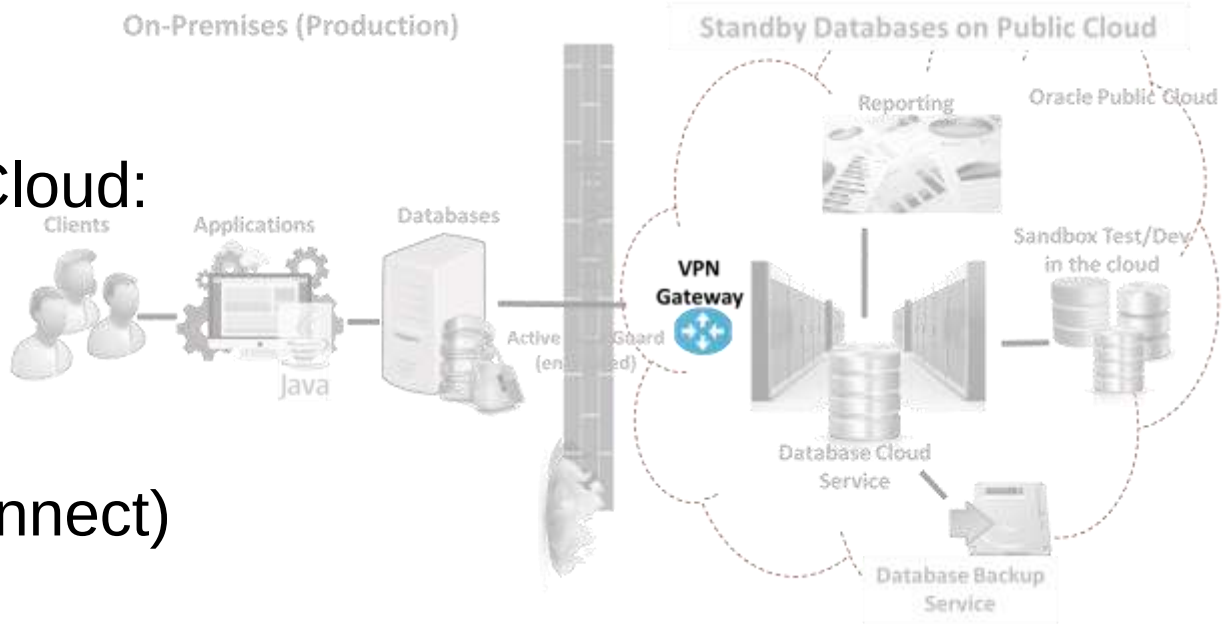
Database Cloud
Service

Database Backup
Service

Environment Prerequisites

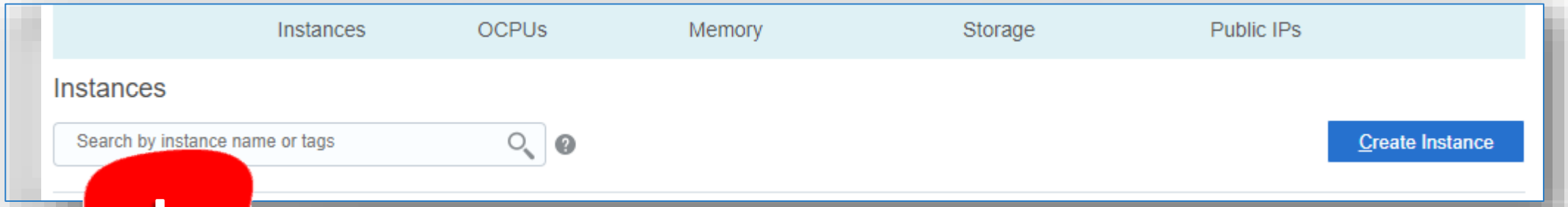
- **OS:** Linux, Windows & Solaris X86, **DB:** EE 64Bit 11.2.0.4, 12.1.0.2, 12.2.0.1 RAC or non-RAC (Note 413484.1 for Data Guard cross-platform compatibility)
- Same Oracle Database version for primary and standby databases

Data transfers from on-premises to Oracle Cloud:



- Public network
- High bandwidth option (Oracle FastConnect)
- Oracle Network Cloud Service – Site to Site VPN or VPN as a Service (VPNaaS)

Deployment Process – Create an Oracle Instance



Technical

Create Instance

Cancel

Instance — Details — Confirm

Next

Instance

Provide basic service instance information.

* Instance Name	stby	* Service Level	Oracle Database Cloud Service
Description		* Metering Frequency	Hourly
Notification Email	yossi.nixon@axxana.com	* Software Release	Oracle Database 12c Release 2
Region	No Preference	* Software Edition	Enterprise Edition
Tags		* Database Type	Single Instance

Database Configuration

Previous | Cancel | Instance | Details | Confirm | Next

Selecting 'None' for Backup Destination may result in no backups for your service instance.

Instance Details

Provide details for this Oracle Database Cloud Service instance.

Selection Summary

Database Configuration

* DB Name: stby
* PDB Name: PDB1
* Administration Password: ••••••••
* Confirm Password: ••••••••
* Usable Database Storage (GB): 25
Total Data File Storage (GB): 88.5
* Compute Shape: OC3 - 1.0 OCPU, 7.5 GB RAM
* SSH Public Key: YossiCloudKey.pub Edit
Use High Performance Storage

Backup and Recovery Configuration

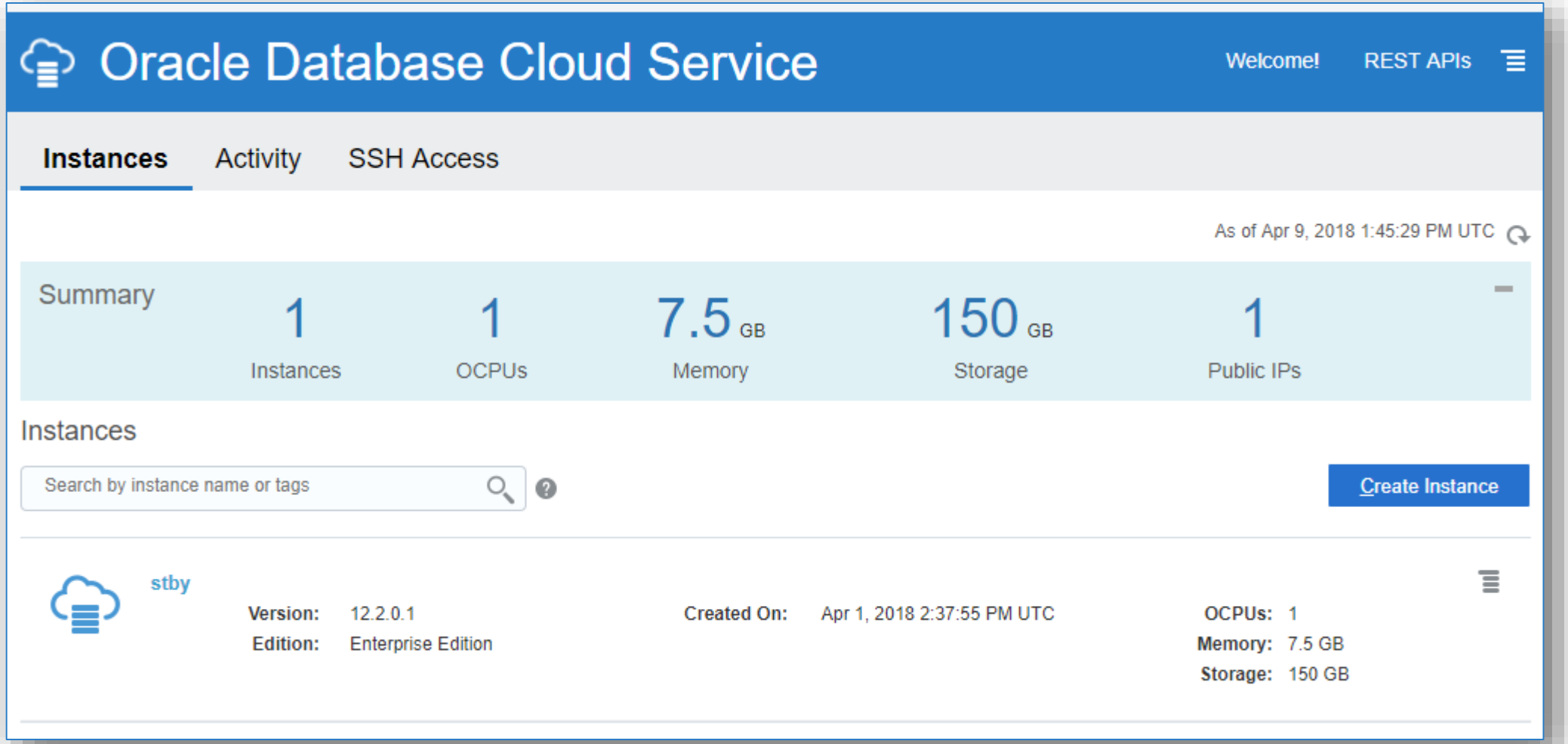
* Backup Destination: None

Initialize Data From Backup

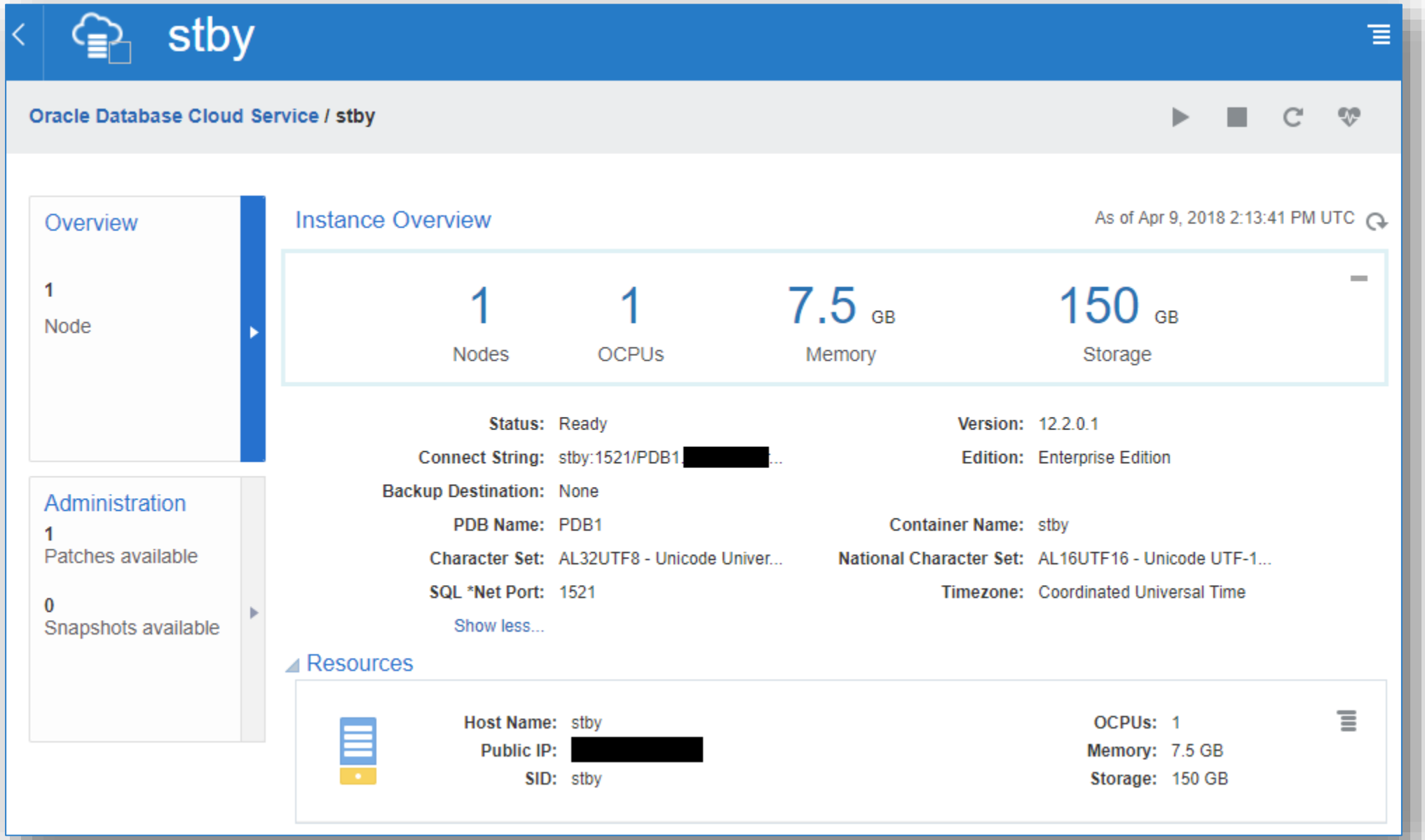
* Create Instance from Existing Backup: No

Advanced Settings

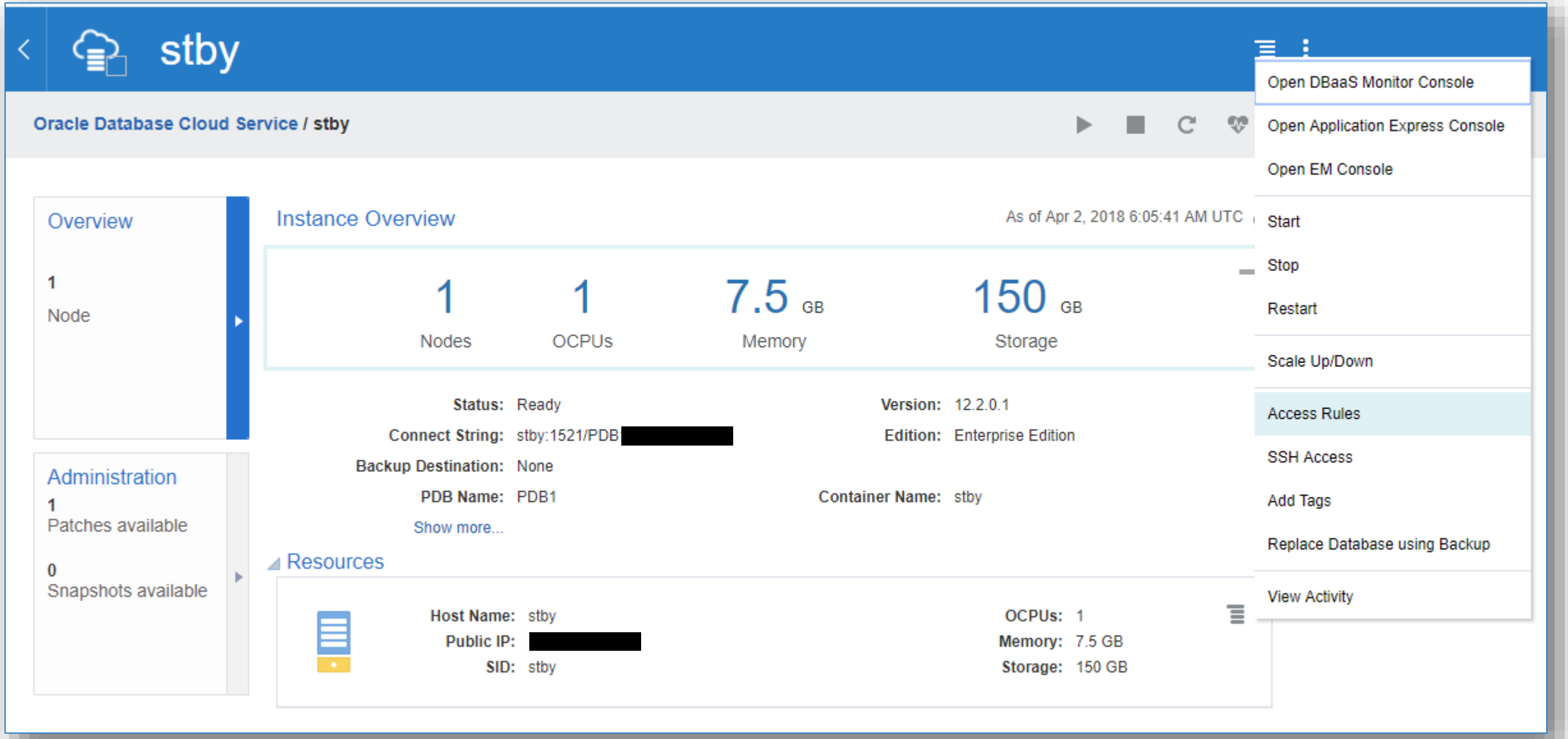
Database Cloud Configured



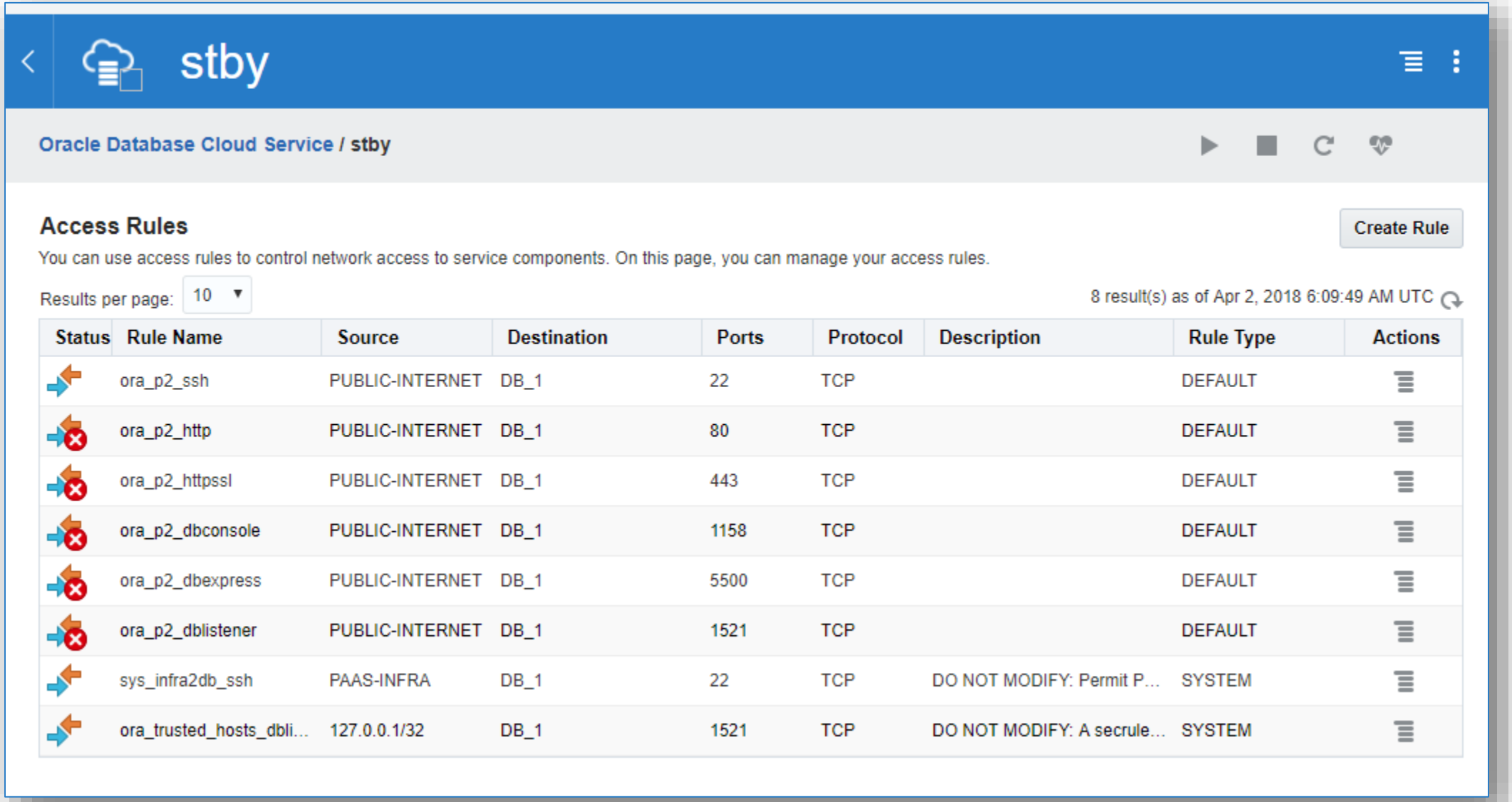
Instance Overview



Create Access Rules



Access Rules List



Adding new Access Rule

Create Access Rule

* Rule Name:	DGonPREM
Description:	allow access from on-prem primary database server
* Source:	<custom>
	5/32
* Destination:	DB_1
* Destination Port(s):	1521
* Protocol:	TCP

This operation may take some time.

Create Cancel

On-Premises - Access & prompt-less SSH

- Configure Access
- Configure Name Resolution
- prompt-less SSH:

  Oracle Cloud: Generate the ssh key and copy the file to the On-Premises

```
$ ssh-keygen
$ scp ~/.ssh/id_rsa.pub oracle@<onpremisesIP>:~/.ssh/id_rsa.pub_cloud
```

- **On-Premises**: Copy the generated key to the `authorized_users` file

```
$ cat ~/.ssh/id_rsa.pub_cloud >> ~/.ssh/authorized_users
$ chmod 700 ~/.ssh/authorized_users
```

On-Premises – SSH login

Connect to the provisioned cloud instance:

```
$ ssh opc@<cloudip> [-i <key>]
```

Switch to root:

```
[opc@stby ~]$ sudo su -
```

On-Premises - TCP Socket Buffer Size

- Check the TCP socket sizes on Oracle Cloud & On-Premises:

  run as root

```
# /sbin/sysctl -a | egrep net.core.[w,r]mem_max
net.core.wmem_max = 2097152
net.core.rmem_max = 4194304
# /sbin/sysctl -a | egrep net.core.[w,r]mem_max
net.core.wmem_max = 1048576
net.core.rmem_max = 4194304
```

- If needed adjust sockets size maximums to 10MB on Oracle Cloud:

```
# sysctl -w net.core.rmem_max=10485760
# sysctl -w net.core.wmem_max=10485760
```

On-Premises - Oracle Net

On-premises and cloud machines

- sqlnet.ora

```
SQLNET.ENCRYPTION_SERVER = requested
SQLNET.CRYPTO_CHECKSUM_SERVER = requested
SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER = (SHA1)
SQLNET.ENCRYPTION_TYPES_SERVER = (AES256, AES192, AES128)

SQLNET.ENCRYPTION_CLIENT = requested
SQLNET.CRYPTO_CHECKSUM_CLIENT = requested
SQLNET.ENCRYPTION_TYPES_CLIENT = (AES256, AES192, AES128)
```

On-Premises - Oracle Home and Patches

Patch differences between databases software homes must be:

Standby-First compatible.

```
[oracle@stby ~]$ $ORACLE_HOME/OPatch/opatch lspatches
26569225;
24701882;
26389300;
26272761;
24401351;
26635944;OJVM RELEASE UPDATE: 12.2.0.1.171017 (26635944)
26710464;Database Release Update : 12.2.0.1.171017 (26710464)

OPatch succeeded.
```

Note 1265700.1 on Data Guard Standby-First patch apply

On-Premises - Standby Redo Logs

SRLs – Standby Redo Logs

ORLs – Online Redo Logs

- SRLs size = largest of ORLs (preferred to be uniform size)
- Number of SRLs = number of ORLs + 1 for each thread (Per Instance)
- SRLs should have same number of threads as ORLs
- Best practice is that SRLs are not duplexed like ORLs

On-Premises - Standby Redo Logs

```
alter system set DB_CREATE_ONLINE_LOG_DEST_1='&DISK_GROUP.';
declare
  log_num number;
  log_size number;
  log_num_standby number;
begin
for j in  (select THREAD# i from gv$log group by THREAD#) loop
     select count(*) into log_num from gv$log where THREAD#=j.i ;
     select count(*) into log_num_standby from gv$standby_log where THREAD#=j.i ;
     select max(BYTES) into log_size from gv$log where THREAD#=j.i ;
    for i in 1..(log_num+1-log_num_standby) loop
    execute immediate 'ALTER DATABASE ADD STANDBY LOGFILE THREAD '||j.i ||' SIZE
'||log_size;
    end loop;
end loop;
end;
/
```

SRLs are not duplexed

Number of Threads

Logs Per Tread

Maximum Size of ORLs

Bypass Already Created SRLs

On-Premises: Grid Infrastructure, TDE & Archive mode

- Grid infrastructure / Oracle Restart has become an integral part of the application failover features for Oracle Data Guard
- The installation software for Oracle Grid infrastructure is not present on the cloud service
- Enable Archive Log Mode
- Convert Database to use Transparent Data Encryption

Prepare the Cloud Environment

- Verify Oracle Home and Patches
- Remove the Default Database

Database

```
dbca -silent -deleteDatabase -sourceDB STBY -sysDBAUserName sys
-sysDBAPassword <passwd>
```

- Install the Grid Infrastructure (if not already installed)
- Set TCP Socket Buffer Size
- Cloud Oracle Net Encryption Configuration

Network

Setting tnsnames.ora

- Configure TNS entries for redo transport – Primary/Standby

```
<primary/standby db_unique_name> =
 (DESCRIPTION =
  (SDU=65536)
  (RECV_BUF_SIZE=10485760)
  (SEND_BUF_SIZE=10485760)
  (ADDRESS = (PROTOCOL = TCP)(HOST = <primary/standby IP address>)(PORT =
{<port#>}))
 (CONNECT_DATA =
  (SERVER = DEDICATED)
  (SERVICE_NAME = <primary/standby db_unique_name>)
 )
)
```

Setting listener.ora

- Configure static listeners on the Cloud

```
SID_LIST_LISTENER =
(SID_LIST =
 (SID_DESC =
  (GLOBAL_DBNAME = <Local Instance name>_DGMGRL)
  (ORACLE_HOME = <Local Oracle Home>)
  (SID_NAME = <Local Instance Name>)
 )
)

$ORACLE_HOME/bin/lsnrctl reload <listener name>
```

On 11.2 - a static listener is required for Data Guard Broker

Preparing Auxiliary Database

- Create Audit Directory

```
mkdir -p /u01/app/oracle/admin/<STANDBY DBNAME>/adump
```

- Create Auxiliary Database, Password File and init.ora

```
$ $ORACLE_HOME/bin/orapwd file='$ORACLE_HOME/dbs/orapw<INSTANCE_NAME>'
password=${passwd} force=y
$ echo "db_name=<primary db_name>" > /tmp/aux.pfile
$ echo "db_unique_name=<standby db_name>" >> /tmp/aux.pfile
$ echo "sga_target=800M" >> /tmp/aux.pfile
```

- Start the Auxiliary Instance

```
$ export ORACLE_SID=<standby instance name (STBY)>
$ sqlplus "/ as sysdba"
SQL> startup nomount pfile='/tmp/aux.pfile'
ORACLE instance started.
```

Duplicate Database from On-Premisses to Cloud

- Distribute TDE wallets

Copy the TDE wallet files from on-premises to the cloud to ENCRYPTION_WALLET_LOCATION directory (defined in sqlnet.ora file)

- Instantiate, creating standby database using RMAN DUPLICATE

```
RMAN> ... duplicate target database for standby from active database
spfile
PARAMETER_VALUE_CONVERT= '<PREMISES DB NAME>', '<CLOUD DB NAME>'
set db_unique_name='<CLOUD DB NAME>' …
```

- Set these additional parameters

```
alter system set DB_FLASHBACK_RETENTION_TARGET=120 scope=both sid='*';
alter system set remote_login_passwordfile='exclusive' scope=spfile sid='*';
alter system set DB_BLOCK_CHECKSUM=FULL; (FULL for inmemory checksum, performance impact)
alter system set DB_BLOCK_CHECKING=MEDIUM; (FULL if performance allows it)
alter system set DB_LOST_WRITE_PROTECT=TYPICAL;
alter system set LOG_BUFFER=256M scope=spfile sid='*';
alter system set STANDBY_FILE_MANAGEMENT=AUTO;
```

Setting Grid Infrastructure + preparing for Broker

- Register the standby database Oracle Restart

```
$ srvctl add database -d <standby db_unique_name> -c SINGLE <result of
hostname -s> -oh <oracle home> -r physical_standby -s <mount|open>
```

- Configure client Failover

Clients can reconnect to the active primary database after a failure

- Configure Data Guard Broker

```
SQL> alter system set dg_broker_start=FALSE;
SQL> alter system set dg_broker_config_file1='<PREMISES_FILE_LOC>/<CLOUD DB NAME>/dr1.dat';
SQL> alter system set dg_broker_config_file2='<PREMISES_RECOV_LOC>/<CLOUD DB NAME>/dr2.dat';
SQL> alter system set dg_broker_start=TRUE;
```

Configuring Data Guard Broker

- Data Guard Broker Configuration

```
dgmgrl sys/<passwd>@<PREMISES>

create configuration 'DGconfig' as primary database is <PREMISES> connect identifier
is <PREMISES>;

add database <CLOUD> as connect identifier is <CLOUD>;

edit database <PREMISES> set property RedoRoutes='(LOCAL: <CLOUD> ASYNC)';
edit database <CLOUD> set property RedoRoutes='(LOCAL:<PREMISES> ASYNC)';
RedoRoutes - not relevant for 11g

EDIT CONFIGURATION SET PROTECTION MODE AS MaxPerformance;

enable configuration;
```

Data Guard Health Check

- Data Guard specific queries

```
SQL> select thread#,count(group#) from v$log group by thread#;        SRL >= ORL per Thread
SQL> select distinct bytes from v$log;                                Same single value
SQL> select distinct bytes from v$standby_log;                        Same single value
SQL> select group#,count(member) from v$logfile where type='STANDBY' group by
group#;                                                               one for all groups
SQL> select flashback_on from v$database;                             YES
SQL> select force_logging from v$database;                            YES
SQL> show parameter checking
SQL> show parameter log_archive_max_processes    remote destinations +  threads/instances
```

Data Guard Health Check

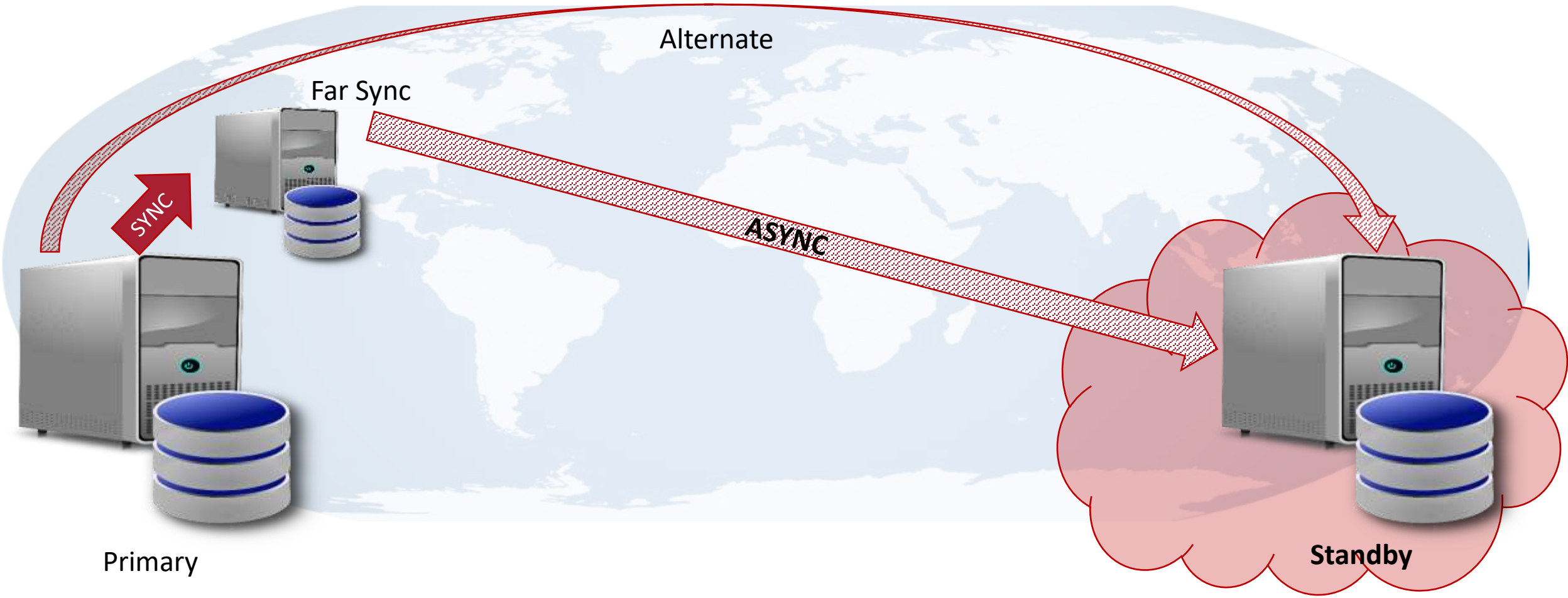
- Data Guard VALIDATE DATABASE

```
DGMGRL> validate database stby;

Database Role: Physical standby database
Protection Mode:   MaxAvailability
Primary Database: pdb

Ready for Switchover: Yes
Ready for Failover: Yes (Primary Running)
```

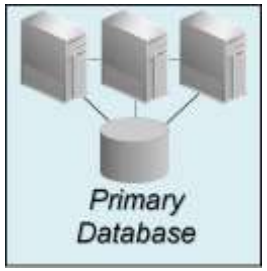
Active Data Guard with Far Sync




```
DGMGRL> EDIT CONFIGURATION SET PROTECTION MODE AS MAXAVAILABILITY;

RMAN> CONFIGURE ARCHIVELOG DELETION POLICY TO APPLIED ON ALL STANDBY;
```

Far Sync Syntax




```
SQL > ALTER DATABASE CREATE FAR SYNC INSTANCE
CONTROLFILE AS  '/tmp/farsync.ctl';
SQL > create pfile='/tmp/initfs.ora' from spfile;
```


```
SQL > ALTER DATABASE ADD STANDBY LOGFILE THREAD 2
SIZE 52428800;
SQL > create spfile from pfile='/tmp/initfs.ora';
```

```
DGMGRL> ADD FAR_SYNC fs AS CONNECT IDENTIFIER IS fs;
```

```
DGMGRL> EDIT DATABASE pdb SET PROPERTY RedoRoutes = '(LOCAL
: fs SYNC ALT =(sdb ASYNC FALLBACK))';
```

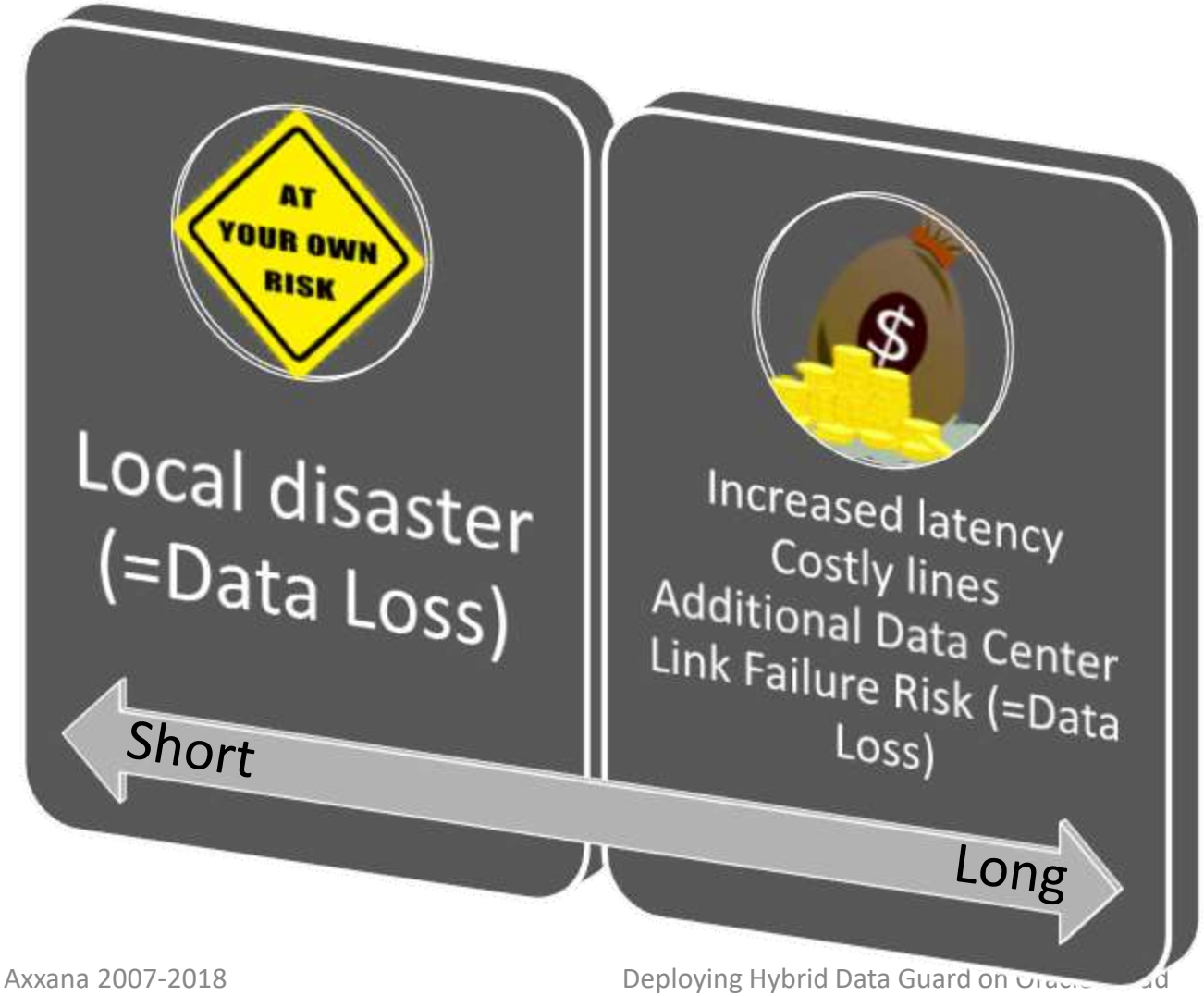
```
DGMGRL> EDIT DATABASE pdb SET PROPERTY RedoRoutes = '(LOCAL
: fs SYNC PRIORITY=1, sdb ASYNC PRIORITY=2 ) )';
```

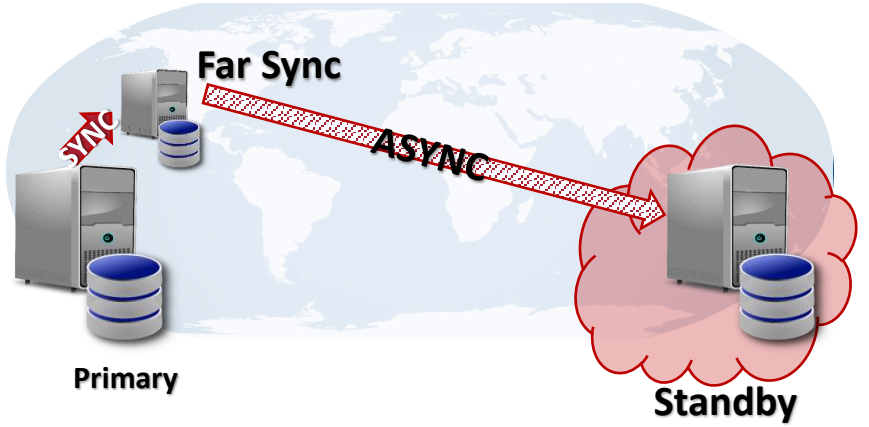
NEW IN 12.2

https://oracledba.blogspot.com/2017/08/creating-far-sync-instance-step-by-step.html

Distance Risk

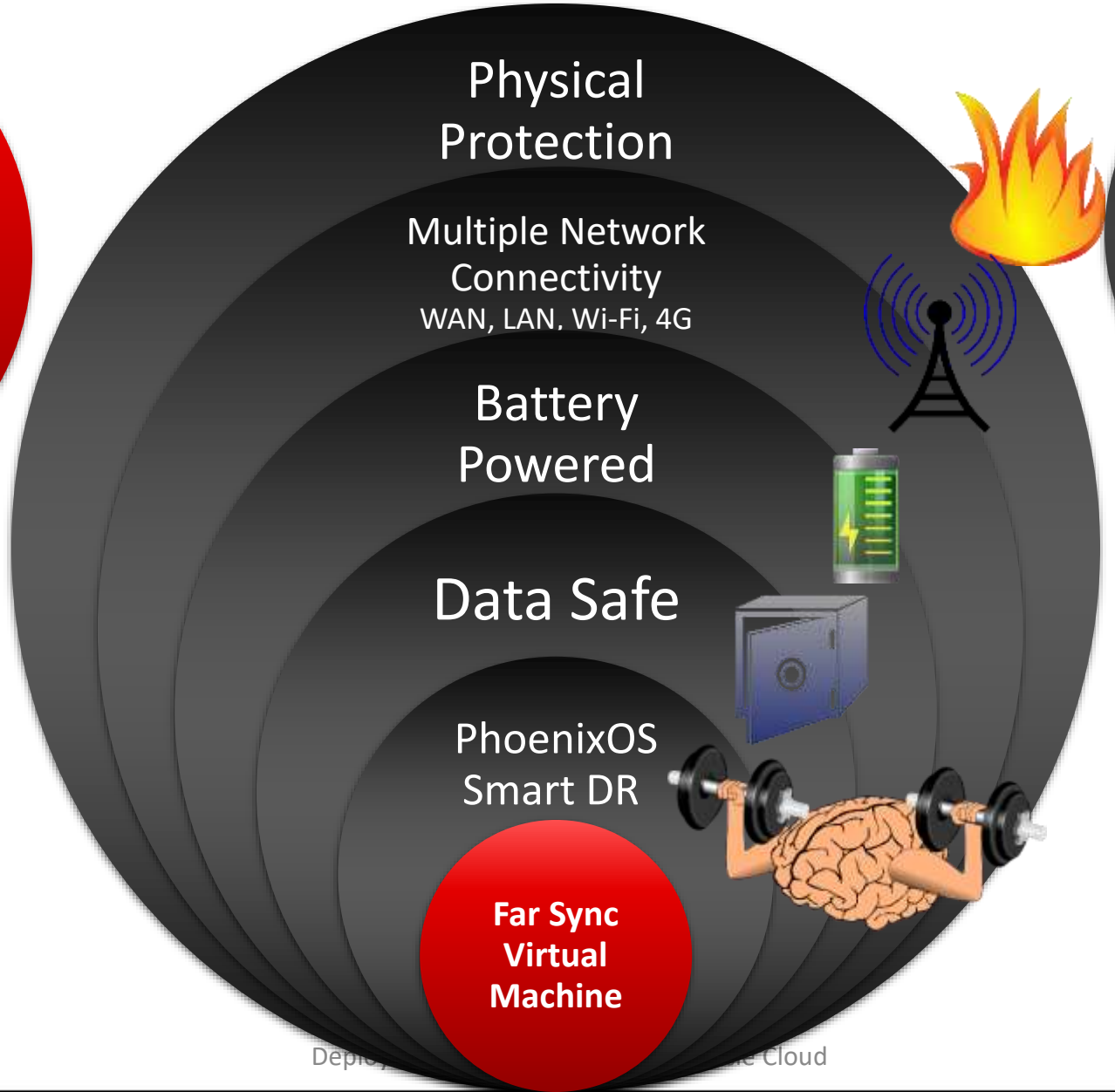
Distance Between Primary and Standby/Far Sync





Axxana – Transparent Layered Protection

Managed by the User



Transparently Managed by Axxana

Active Data Guard Far Sync with Axxana

Short Video

Video : The Phoenix by Axxana - Multi Layered Extensive Protection

Conclusion

Disaster Recovery on the Oracle Cloud

☑ Eliminates costs and complexity

Data Guard or Active Data Guard

☑ Eliminates downtime potential risk

Axxana's Solution

☑ True zero data loss in the hybrid cloud

References

Hybrid DR to Oracle Cloud https://www.oracle.com/assets/dr-to-oracle-cloud-2615770.pdf

Oracle Data Guard – http://www.oracle.com/goto/dataguard

Maximum Availability Architecture (MAA) – http://www.oracle.com/goto/maa

MAA Blogs – http://blogs.oracle.com/maa

Oracle Cloud Infrastructure – getting started https://docs.us-phoenix-1.oraclecloud.com/pdf/gsg/OCI_Getting_Started.pdf

Questions?

AXXANA
BUILT TO LAST