

ORACLE WHITEPAPER
SEPTEMBER 2020

DISASTER RECOVERY (DR) IN THE CLOUD – USING ORACLE DATABASE STANDARD EDITION AND THE ORACLE CLOUD

Contents

Introduction.....	3
Backups and Disaster Recovery (DR) Databases.....	3
Journey to the Cloud.....	3
Dbvisit and Oracle.....	4
Dbvisit Standby Overview	5
Graceful Switchover.....	6
Requirements.....	6
Dbvisit Standby Architecture	7
Set up for Dbvisit Standby On-Premises to Oracle Cloud Configuration	9
Switchover	23
Failover	24
Conclusion.....	26

Introduction

This whitepaper is for a technical audience that is familiar with the Oracle database and has a knowledge of Disaster Recovery (DR) databases. We will discuss Dbvisit Standby software and its compatibility with Oracle Cloud. We will also focus on the benefits of a hybrid configuration for your databases, including installation and setup of Dbvisit Standby for an on-premises to Oracle Cloud configuration.

Backups and Disaster Recovery (DR) Databases

Backups usually cannot fulfil a business' need to meet strict Recovery Time Objective (RTO) and Recovery Point Objective (RPO) if a disaster strikes. During this time you will require a standby database to be ready at a moment's notice. While backups require more time to perform restore and recovery in case of a disaster, Dbvisit Standby can help you establish a standby database environment and keep it up to date. Standby databases can also be located in a different geographical location than the primary database, this can be especially helpful in the event of a natural disaster. Standby Databases however do not replace backups, you still require backups. Regular backups are important and still fulfil an integral part of a proper disaster recovery plan, but having a standby database is more time efficient especially when strict RPO and RTO rules are in play. As the digital transformation continues with enterprises across the globe, the data of that enterprise becomes all the more important, hence the need for faster response and recovery times. When looking at those different geographical locations, sometimes a cloud option becomes more relevant. With cloud providers having multiple datacenters across the globe, they provide what was previously a more complicated issue with an easy solution.

Journey to the Cloud

Dbvisit Standby is a Disaster Recovery (DR) solution that was built to support Oracle Standard Edition databases whether on-premises or in the Cloud.

More and more organizations are placing their data in databases which are in cloud environments. Proper Disaster Recovery (DR) typically dictates that the recovery site be placed in a different geographical location than the primary production site. Placing your database with a cloud provider can save significant costs if you are talking about large data centers with associated hardware, networks and ancillary costs.

Cloud providers are filling this gap and offer options that can be cost effective, as they provide hardware and infrastructure in various geographic locations. Having your production database on-premises and your standby database in the cloud is often called a Hybrid Cloud solution. Most cloud vendors also provide different options of database and server configurations. This allows for organizations to experiment with different server configurations or allow them to have smaller DR sites than their production site. While many DR purists would say that the DR site should be an exact mirror of production, there are those who understand that there are cost and resource constraints – also taking into account that when using cloud based solutions, instance sizes can in most cases be adjusted from small/minimum to larger configurations in a short period of time. Oracle Corporation offers several options for hosting hybrid DR databases. The Oracle Cloud offers several different geographical locations where databases can be hosted and has several more locations planned in the next few years.

Many customers are exploring hosting their databases in the cloud. While these customers may have been

reluctant to do so before, they are increasingly comfortable with having their databases in the cloud. Having your DR environment in the cloud is a great first step. This will allow DBAs to learn the Oracle Cloud without the pressure of having the production database in the cloud. DBAs would learn how the Oracle database sits in the Oracle Cloud environment, increasing their Oracle Cloud skills as they set up their hybrid DR environment.

Having your standby database in the Cloud is a great DR move. It's even better if your DR environment is in a different geographic location to your production environment. This way you don't have to buy new hardware and worry about where to fit a DR environment into your data center, and you know that if disaster should strike where your production environment is, you have a copy of that critical data in your DR environment.

There are certain prerequisites to have a hybrid DR environment in the Oracle Cloud, it might seem like a big task - but in fact it is similar to setting up Dbvisit Standby for on-premises to on-premises DR.

Dbvisit and Oracle

Dbvisit Standby has become the global standard for Oracle Standard Edition database protection. Oracle customers using Oracle Enterprise Edition are used to the protection that Oracle Data Guard provides. Oracle Standard Edition does not come with Data Guard as it is a feature only available in Enterprise Edition. Dbvisit Standby has helped Standard Edition customers for over 10 years, and it is used in over 130 countries with thousands of databases under protection.

As Cloud use has grown over the years, customers of Dbvisit are looking to have that protection extended to the Cloud. Dbvisit's aim is to make easy to use disaster recovery software, whether the solution is on-premises, a hybrid configuration, or a full Cloud based solution.

The Oracle Cloud has made great strides in the past few years. There are many different cloud options available, but some options are more relevant to our discussion than others. Most SaaS (Software as a Service) and the managed versions of PaaS (Platform as a Service) are not relevant - as those are typically platforms that already host production databases which is not managed by the end customer, but by the cloud service provider (Managed Services).

Oracle also has an Exadata Cloud offering as well. While in theory, Dbvisit Standby would work on the Exadata selection, the likelihood is that those customers running Exadata will also be using Oracle Enterprise Edition (with Data Guard included) to make use of most of the Exadata features.

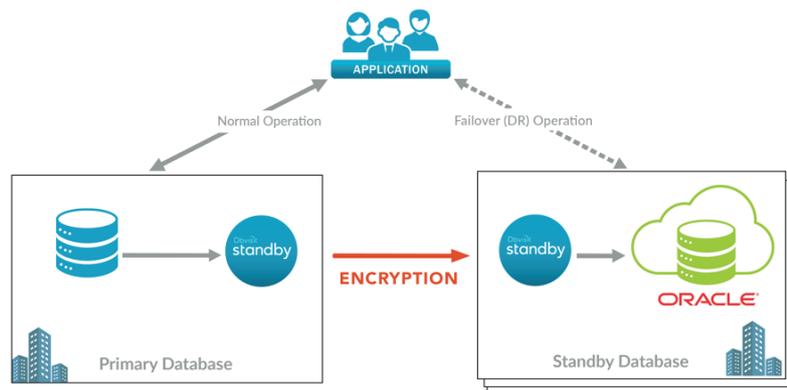
There are two Oracle Cloud offerings that can make use of the service provided by Dbvisit Standby. These two services are IaaS - Infrastructure as a Service, and DBaaS - Database as a Service offering. With IaaS you are more in control of software you would like to install. To simplify it, think of this type of configuration as you get the option to select a specific instance type (Virtual Machine) onto which you can install your software as you normally would in an on-premise configuration. With DBaaS the options are similar, but with the option that you can choose between Enterprise Edition and Standard Edition database environments. This whitepaper will focus on the DBaaS offering and the Standard Edition database option.

The IaaS and the DBaaS offerings are the most alike to on-premises environments in the Oracle Cloud. DBAs who are used to dealing with their Linux operating systems located on-premises will feel right at home with the IaaS and DBaaS offerings. Once the DBAs do the initial cloud configurations, connecting and

working with the operating system and Oracle database is essentially as if they were working on-premises. That provides a great level of comfort to DBAs that are new to the Oracle Cloud and those that are leery about having their databases in the cloud.

While Dbvisit Standby supports from Oracle 10.2 up to and including Oracle 18c, the Oracle Cloud DBaaS offerings typically are only from Oracle 11.2 and above. If you need to have a lower version of the Oracle database, you may have to look at the Oracle IaaS option where you have more flexibility of installing and managing your own database software version.

Dbvisit Standby Overview



Dbvisit Standby software works in the same manner, no matter where the databases are stored, whether on-premises, in a hybrid configuration, or a full cloud configuration.

Dbvisit software is installed on the operating system where the databases reside. Dbvisit Standby software supports many different operating systems as well as many different Oracle database versions. However it is recommended that the primary and secondary (standby) systems have the same operating system and database software version installed. Remember a DR site is there to protect your precious data and taking risks with different configurations on the standby site is not recommended. The Dbvisit Standby software is installed on both the source and secondary (standby) servers. Once the software is installed and configured, Dbvisit Standby has the ability to create the standby database from scratch using the Create Standby Command.

This is a great feature and takes much of the guess work out of the DBA's hands. After the standby database has been created, Dbvisit Standby will compress, encrypt and ship the archive logs from the source primary database over to the waiting standby database. Once the archive logs are received on the target server, Dbvisit will apply the archive logs to the Oracle standby database.

The shipping and applying of archive logs can be made to run automatically as they are created, or they can be set using timed commands to run every few minutes. Once the Dbvisit Standby process is set up you are done and the processes will keep shipping and applying as you have pre determined, no further manual action is required as Dbvisit Standby will maintain the synchronization of the two databases.

Graceful Switchover

One feature that is a highlight of Dbvisit Standby is that of a switchover, also known as Graceful Switchover or role reversal. With one command, the primary database becomes the standby database and the standby database becomes the primary database. Dbvisit ships the last archive log over to the standby, and the primary database is shutdown and then put in recovery mode while the standby server is opened as a primary and then starts shipping logs to the new standby (formerly the primary) database. There is no loss of data but there will be a short outage while the role reversal takes place. This option is used for a variety of reasons:

- Testing standby hardware
- Migrating to new hardware or the Cloud
- When you have enough time to perform a Graceful Switchover instead of a Failover - in summary you have sufficient warning time before disaster strikes.

Like the Graceful Switchover feature, the Failover feature is another great feature to help DBAs in their time of need. The Failover feature does just what it sounds like, it is to be used in a time of disaster. There is a risk of data loss (whatever data is in the redo logs or hasn't been transferred over to the standby site). The Failover command basically opens up the target standby database to become the primary. Again, there is a potential risk of data loss so this command should only be done in the actual event of a disaster.

Requirements

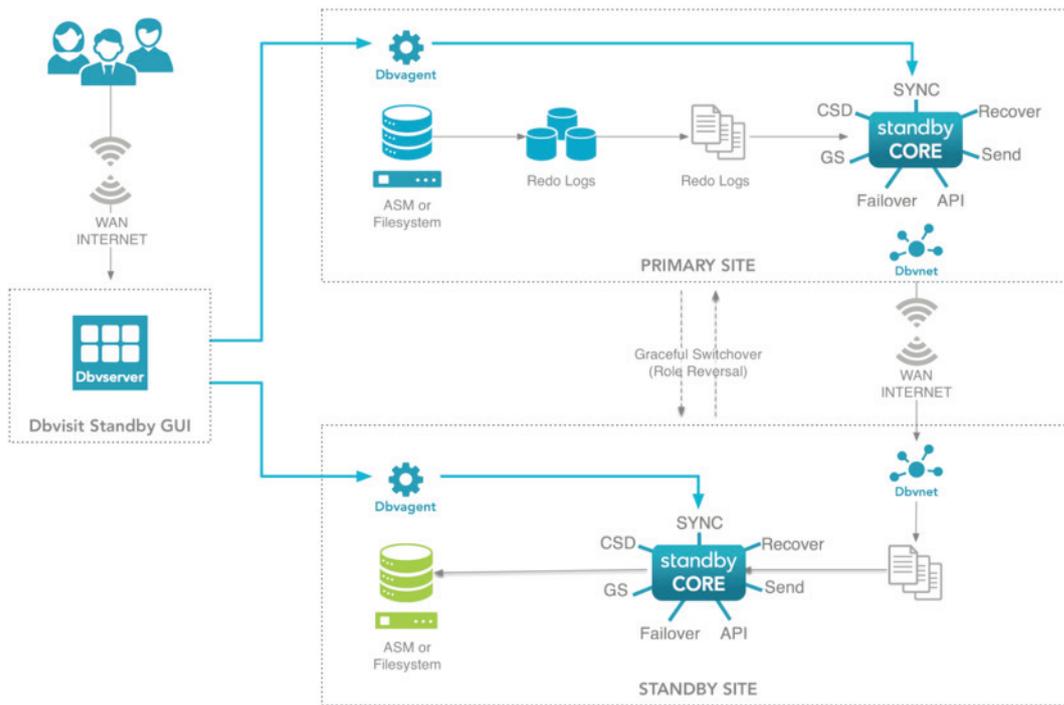
Dbvisit Standby supports a wide variety of operating systems and versions of Oracle databases. You can find a complete list of the versions in the [Dbvisit Standby documentation](#).

Dbvisit software does not require that much physical space on the operating systems, however you will need sufficient space to store a certain number of archive logs. Consideration should be taken into account regarding peak archive log creation and the number of archives that you may want retained on the system. Some customers prefer one to two days while others consider four days to be the minimum, it will often change depending on operational considerations.

The versions of Oracle, including the edition, should be the same on the primary and standby sites.

The same operating system is required on the primary and standby sites.

Dbvisit Standby Architecture



Dbvisit Standby consists of the following components:

<p>Dbvnet (Dbvisit Networking)</p>	<p>This component is responsible for the secure communication between the primary and standby systems. It consists of the following executable:</p> <ul style="list-style-type: none"> • dbvnet - This is the main Dbvnet executable used to start the networking communication between the primary and standby server. <p>This component should run on both the primary and standby servers and is configured during the installation process.</p> <p>The default port used by Dbvnet is 7890 and this port must be opened on both the primary and standby database servers to allow communication.</p>
<p>Dbvserver (Dbvisit Standby GUI)</p>	<p>This is a Web server that provides the web interface to Dbvisit. It consists of the following executable:</p> <ul style="list-style-type: none"> • dbvserver - The main Dbvserver executable for the Dbvisit Web server, and is used to start/stop the Dbvisit Web server. <p>Dbvserver is using port 4433 by default and you must use HTTPS to connect to it. The default username is "admin" and password is "admin". It is recommended to change this password.</p> <p>The Dbvserver process will communicate with the Dbvagent (Agent) process on port 7891 by default. It is important to note though that the Agent will communicate with Dbvserver on port 4433.</p> <p>To allow for communication between the webserver and the agent, port 7890 must be open on the agent side and the agent started (service running in Windows and Dbvagent started from the command line on Unix)</p>

<p>Dbvagent (Dbvisit Agent)</p>	<p>The Dbvisit Agent is used to manage the Standby core background processes as well as managing the communication between the Dbvserver (Web-based interface) and the Dbvisit Standby core.</p> <p>The Agent is making use of port 7891 by default. During installation, a passphrase is provided which is used to allow the Dbvserver (webserver) to establish a secure connection to the Agent.</p> <p>The Agent will communicate with the webserver on port 4433.</p>
<p>Standby Core (Dbvisit Standby Core and CLI)</p>	<p>Dbvisit Standby Core also referred to as Standby Core, is the core of the Dbvisit Standby product.</p> <p>This consists of the following executables (commands):</p> <ul style="list-style-type: none"> • dbvctl - The main Dbvisit Standby control utility (executable) <p>The standby core component will be invoked by either the user manually, Dbvnet or Dbvagent depending on which process is executed. There are no specific port requirements for the Standby Core.</p>
<p>Dbvisit Database Configuration (DDC file)</p>	<p>This is similar to the init.ora parameter file in the Oracle database and contains the Dbvisit Standby settings for a specific primary and standby database pair.</p> <p>The DDC file is generated during setup for each database.</p> <p>The DDC file should only be edited on the primary server.</p>
<p>Dbvisit Database Repository (DDR)</p>	<p>This repository contains the Dbvisit Standby tables used to manage the standby process.</p> <p>It is located outside the Oracle database in a small file under the DBVISIT_BASE/conf directory called <ddc>.db, for example if DDC name is PROD the file name will be "prod.db". This repository contains the Dbvisit Standby tables used to manage the standby process.</p> <p>The DDR is created during setup.</p>
<p>Dbvisit Standby trace files</p>	<p>Each time the Dbvisit Standby executes, a trace file is generated. This trace file contains information about Dbvisit Standby processing and its environment.</p> <p>When an alert or error notification is sent by Dbvisit Standby, the trace file will be attached to the email so that this can be forwarded to Dbvisit support if needed.</p> <p>This trace file is used by the Dbvisit support team to diagnose issues created. This trace file contains information about Dbvisit Standby processing and its environment.</p> <p>When an alert or error notification is sent by Dbvisit Standby, the trace file will be attached to the email so that this can be forwarded to Dbvisit support if needed.</p>

Set up for Dbvisit Standby On-Premises to Oracle Cloud Configuration

The following steps will be used to guide you through a Dbvisit Standby configuration of an Oracle Database on-premises to an Oracle database in the Oracle Cloud. This hybrid solution presented below may not include all of the steps and will be noted if they are abbreviated.

1. Create your Oracle DBaaS account and activate the appropriate service. Please see Oracle documentation for the needed steps.
2. Configure the sharing of key encryptions and networking between the source on-premises database and the Oracle Cloud. As there are a myriad of options we have not included the details here.
3. Ensure that the source database is in archive log mode.

Install the Dbvisit software on the source server. The installation of Dbvisit software is simply unzipping, untarring and then running the Dbvisit installer. You can follow the instructions in our [Dbvisit Standby installation guide](#)

```
[oracle@on-prem ~]$ unzip dbvisit-standby8.0.xx-el6.zip
Archive:  dbvisit-standby8.0.xx-el6.zip
  inflating: dbvisit-standby8.0.xx-el6.tar
[oracle@on-prem~]$ tar -xvf dbvisit-standby8.0.12-el6.tar
[oracle@on-prem ~]$ cd dbvisit
[oracle@on-prem dbvisit]$ cd installer/
[oracle@on-prem installer]$ ls -lrt
total 2768
-rwxr-x--- 1 oracle oinstall 2833364 Oct 11 02:16 install-dbvisit
[oracle@on-prem installer]$ ./install-dbvisit
```

4. Now install the Dbvisit software on the Oracle Cloud server

```
[oracle@ocloud ~]$ unzip dbvisit-standby8.0.xx-el6.zip
Archive:  dbvisit-standby8.0.xx-el6.zip
  inflating: dbvisit-standby8.0.xx-el6.tar
[oracle@ocloud~]$ tar -xvf dbvisit-standby8.0.12-el6.tar
[oracle@ocloud ~]$ cd dbvisit
[oracle@ocloud dbvisit]$ cd installer/
[oracle@ocloud installer]$ ls -lrt
total 2768
-rwxr-x--- 1 oracle oinstall 2833364 Oct 11 04:16 install-dbvisit
[oracle@ocloud installer]$ ./install-dbvisit
```

5. The installation wizard will walk you through the necessary steps. This would be done on both the on-premises server as well as the cloud server. The one difference would be that the dbvserver component (to run the Dbvisit Standby Console) * would typically only be installed in one location; on the source on-premises server, on the target cloud server, or on a 3rd tier machine. In our example, we will install it on the cloud server. For brevity we will only show the install output from the cloud server.

```
Welcome to the Dbvisit software installer.

-----

It is recommended to make a backup of our current Dbvisit software
location (Dbvisit Base location) for rollback purposes.

Installer Directory /home/oracle/dbvisit

>>> Please specify the Dbvisit installation directory (Dbvisit Base).

The various Dbvisit products and components - such as Dbvisit
Standby, Dbvisit Dbvnet will be installed in the appropriate
subdirectories of this path.

Enter a custom value or press ENTER to accept default [/usr/dbvisit]:
> /usr/dbvisit
DBVISIT_BASE = /usr/dbvisit

-----

Component          Installer Version   Installed Version
-----
standby            8.0.12.19052       not installed
dbvnet             2.0.12.19052       not installed
dbvagent          2.0.12.19052       not installed
dbvserver          2.0.12.19052       not installed
-----

What action would you like to perform?
  1 - Install component(s)
  2 - Uninstall component(s)
  3 - Terminate

Your choice: 1

Which component do you want to install?
  1 - Core Components (Dbvisit Standby Cli, Dbvnet, Dbvagent)
  2 - Dbvisit Standby Core (Command Line Interface)
  3 - Dbvnet (Dbvisit Network Communication)
  4 - Dbvagent (Dbvisit Agent)
  5 - Dbvserver (Dbvisit Central Console)
  6 - Exit Installer

Your choice: 1

-----

Summary of the Dbvisit STANDBY configuration
-----
```

DBVISIT_BASE /usr/dbvisit

Press ENTER to continue

About to configure DBVISIT DBVNET

>>> Please specify the Local host name to be used by Dbvnet on this server.

Dbvnet will be listening on the local IP Address on this server which resolve to the host name specified here.

If using a cluster or virtual IP make sure the host name or alias specified here resolve to the IP address local to where dbvnet is installed.

Enter a custom value or press ENTER to accept default [ocloud]:
>

>>> Please specify the Local Dbvnet PORT to be used.

Dbvnet will be listening on the specified port for incoming connections from remote dbvnet connections. Please make sure that this port is not already in use or blocked by any firewall. You may choose any value between 1024 and 65535, however the default of 7890 is recommended.

Enter a custom value or press ENTER to accept default [7890]:
>

>>> Please specify the Remote host name to be used by Dbvnet.

By default Dbvnet will use this remote hostname for any remote connections. Dbvnet must be installed and configured on the specified remote host. If using a cluster or virtual IP make sure the host name or alias specified here resolve to the IP address local to where dbvnet is installed.

If you are unsure about the remote host name during installation, use the default value which will be the current local hostname.

Enter a custom value or press ENTER to accept default [ocloud]:
> on-prem

>>> Please specify the Remote Dbvnet PORT to be used.

Dbvnet will connect to the remote server on this specified port. On the remote host Dbvnet will be listening on the specified port for incoming connections. Please make sure that this port is not already in use or blocked by any firewall. You may choose any value between 1024 and 65535, however the default of 7890 is recommended.

Enter a custom value or press ENTER to accept default [7890]:
>

>>> Please specify the Dbvnet Passphrase to be used for secure connections. The passphrase provided must be the same in both the local and remote

Dbvnet installations. It is used to establish a secure (encrypted) Dbvnet connections.

Enter a custom value:

> sample

Summary of the Dbvisit DBVNET configuration

DBVISIT_BASE /usr/dbvisit
DBVNET_LOCAL_HOST ocloud
DBVNET_LOCAL_PORT 7890
DBVNET_REMOTE_HOST on-prem
DBVNET_REMOTE_PORT 7890
DBVNET_PASSPHRASE sample

Press ENTER to continue

About to configure DBVISIT DBVAGENT

>>> Please specify the host name to be used for the Dbvisit Agent.
The Dbvisit Agent (Dbvagent) will be listening on this local address.
If you are using the Dbvserver (GUI) - connections from the GUI will
be established to the Dbvisit Agent. The Dbvisit Agent address must
be visible from the Dbvserver (GUI) installation.
If using a cluster or virtual IP make sure the host name or
alias specified here resolve to the IP address local to where dbvnet
is installed.

Enter a custom value or press ENTER to accept default [ocloud]:
>

>>> Please specify the listening PORT number for Dbvagent.

The Dbvisit Agent (Dbvagent) will be listening on the specified
port for incoming requests from the GUI (Dbvserver). Please make
sure that this port is not already in use or blocked by any
firewall. You may choose any value between 1024 and 65535, however
the default of 7891 is recommended.

Enter a custom value or press ENTER to accept default [7891]:
>

>>> Please specify passphrase for Dbvagent
Each Dbvisit Agent must have a passphrase specified. This passphrase
does not have to match between all the servers. It will be used to
establish a secure connection between the GUI (Dbvserver) and the
Dbvisit Agent.

Enter a custom value:
> sample

Summary of the Dbvisit DBVAGENT configuration

DBVISIT_BASE /usr/dbvisit
DBVAGENT_LOCAL_HOST ocloud
DBVAGENT_LOCAL_PORT 7891
DBVAGENT_PASSPHRASE sample

Press ENTER to continue

About to install Dbvisit STANDBY

Component standby installed.

Press ENTER to continue

About to install Dbvisit DBVNET

Component dbvnet installed.

Press ENTER to continue

About to install Dbvisit DBVAGENT

Component dbvagent installed.

Press ENTER to continue

Component Installer Version Installed Version

standby 8.0.12.19052 8.0.12.19052
dbvnet 2.0.12.19052 2.0.12.19052
dbvagent 2.0.12.19052 2.0.12.19052
dbvserver 2.0.12.19052 not installed

What action would you like to perform?

- 1 - Install component(s)
- 2 - Uninstall component(s)
- 3 - Terminate

Your choice: 1

Which component do you want to install?

- 1 - Core Components (Dbvisit Standby Cli, Dbvnet, Dbvagent)
- 2 - Dbvisit Standby Core (Command Line Interface)
- 3 - Dbvnet (Dbvisit Network Communication)
- 4 - Dbvagent (Dbvisit Agent)
- 5 - Dbvserver (Dbvisit Central Console)
- 6 - Exit Installer

Your choice: 5

About to configure DBVISIT DBVSERVER

>>> Please specify the host name to be used for Dbvserver

The Dbvisit Web Server (Dbvserver) will be listening on this local address. If using a cluster or virtual IP make sure the host name or alias specified here resolve to the IP address local to where Dbvserver is installed.

If you are unsure about the remote host name during installation, use the default value which will be the current local hostname.

Enter a custom value or press ENTER to accept default [dbvcloud02]:
>

>>> Please specify the listening port number for Dbvserver on the local server

You may choose any value between 1024 and 65535.
The default recommended value is 4433.

Note: if you can not access this port after the installation has finished, then please double-check your server firewall settings to ensure the selected port is open.

Enter a custom value or press ENTER to accept default [4433]:
>

Summary of the Dbvisit DBVSERVER configuration

DBVISIT_BASE /usr/dbvisit
DBVSERVER_LOCAL_HOST ocloud
DBVSERVER_LOCAL_PORT 4433

Press ENTER to continue

About to install Dbvisit DBVSERVER

Component dbvserver installed.

Press ENTER to continue

Component Installer Version Installed Version

standby 8.0.12.19052 8.0.12.19052
dbvnet 2.0.12.19052 2.0.12.19052
dbvagent 2.0.12.19052 2.0.12.19052
dbvserver 2.0.12.19052 2.0.12.19052

What action would you like to perform?

- 1 - Install component(s)
- 2 - Uninstall component(s)
- 3 - Terminate

Your choice: 3

>>> Installation completed

Install log /tmp/dbvisit_install.log.201706200938.

At this stage, the installation of the Dbvisit Standby software is complete.

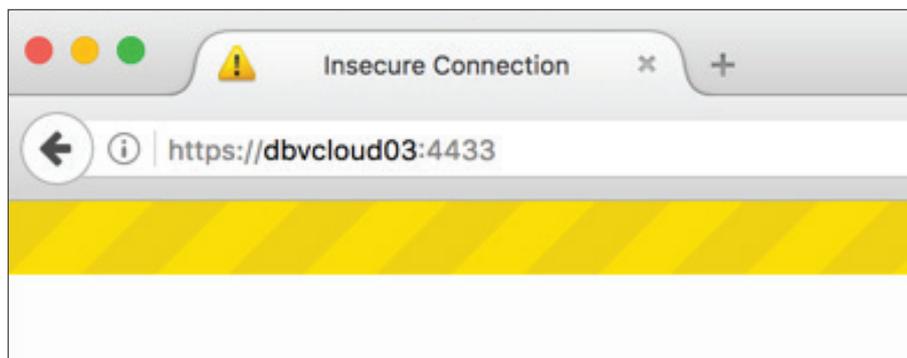
6. With the software installed it is time to start the same processes on the on-premises primary server.

```
[oracle@on-prem installer]$ cd /usr/dbvisit/dbvnet
[oracle@on-prem dbvnet]$ ./dbvnet -d start
[oracle@on-prem dbvnet]$ cd ../dbvagent/
[oracle@on-prem dbvagent]$ ./dbvagent -d start
[oracle@on-prem dbvagent]$ ps -ef|grep -i dbv
oracle    9372      1  0 20:34 ?        00:00:00 ./dbvnet -d start
oracle    9379      1  0 20:34 ?        00:00:00 ./dbvagent -d start
```

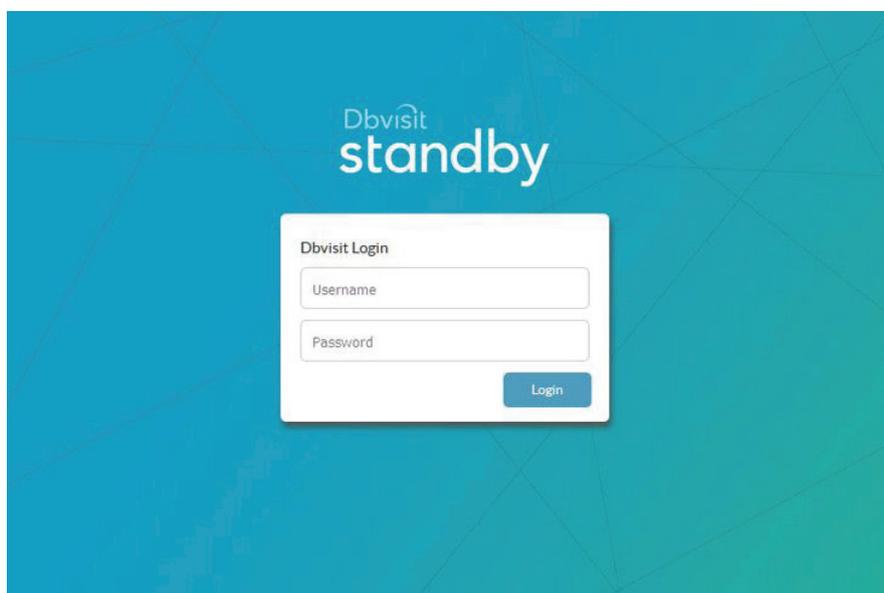
And then start the Dbvisit Server process on the cloud server.

```
[oracle@ocloud dbvagent]$ cd ../dbvserver/
[oracle@ocloud dbvserver]$ ./dbvserver -d start
[oracle@ocloud dbvserver]$ ps -ef|grep -i dbv
oracle    30354     1  0 09:44 ?        00:00:00 ./dbvnet -d start
oracle    30374     1  0 09:44 ?        00:00:00 ./dbvserver -d start
oracle    30549     1  0 09:45 ?        00:00:00 ./dbvagent -d start
```

7. We need to create the DDC (Dbvisit Database Configuration) file. This will contain a series of parameters to configure the Dbvisit software. We will do this by connecting to the Dbvisit Standby Central Console (GUI). We set up the Console on the Oracle Cloud above and now we need to connect to it. Start a web browser and connect to the server (in our example ocloud) and the port designated during the installation process.



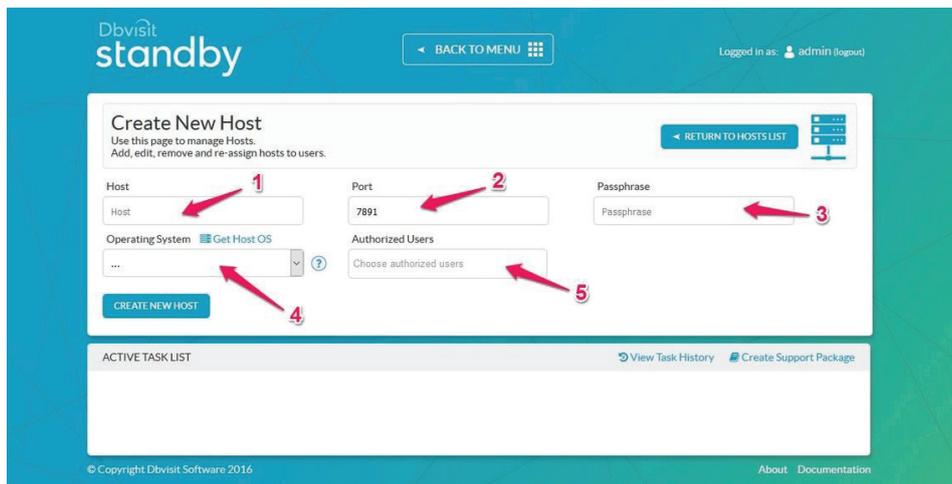
8. Enter the default user of admin, with the default password of admin, on the Dbvisit Standby Central Console's login screen.



9. You will now be at the main screen of the Dbvisit Standby Central Console pictured below



The Console has only two options that are available at this point. You could create new users or manage the host configurations. Clicking on the Manage Hosts will allow us to create New Host. Enter the host, the port and the passphrase that we listed during the installation portion. This will be done once for the on-premises host and once for the Oracle Cloud host.



11. After the two hosts have been added it will look like this:



12. Create a directory for the ARCHSOURCE and ARCHDEST parameters. The ARCHSOURCE parameter is only required in the event of a Graceful Switchover when the primary (on-premises) database becomes the standby database. This would be done on each server.

```
[oracle@ocloud ~]$ mkdir /u01/app/oracle/dbvisit_arch
```

13. On the main screen of the Console, click the Manage Configurations icon and proceed to create a new configuration.

14. With the configuration (DDC) file created, you are ready to create the standby database on the standby Oracle Cloud server. Click the Create Standby Database (CSD) icon. Ensure the parameters are correct and submit. You can follow the progress in the activity bar. Alternatively, you could create the standby database from the command line.

```
oracle@on-prem[/home/oracle]: cd /usr/dbvisit/standby/
oracle@on-prem[/usr/dbvisit/standby]:
oracle@on-prem[/usr/dbvisit/standby]: ./dbvctl -d SAM --csd
```

```
>>> Running pre-checks please wait... done
```

```
What would you like to do:
```

- 1 - Create standby database (and optionally save settings in template)
- 2 - Create standby database using existing template (previously saved)
- 3 - Help
- 4 - Terminate processing

```
Please enter your choice [1]:
```

Answer Yes to transfer the datafiles directly to the standby server (not recommended for large database or slow networks)
Note tablespaces will be placed in backup mode for direct copy.
Answer No (recommended) to create an RMAN backup of primary datafiles in a temporary location and transfer the backup to the standby server

=> Do you want to copy database files directly to the standby server?

[N]:

Your input: N

Is this correct? <Yes/No> [Yes]:

=> Do you want to use TRANSPORTABLE MEDIA to transfer the database backup to the standby server? Transportable media is an external device such as a USB drive that is first plugged into the primary server and then manually transferred to the standby site and plugged into the standby server to continue the process.

It can be used for large databases or slow networks.

Specifying No means the network will be used to transfer the database backup.

[N]:

Your input: N

Is this correct? <Yes/No> [Yes]:

A temporary location must be specified on dbvlab01 where the database will be backed up to first.

This location must be big enough to hold RMAN backup of the whole database

(1.38GB).

=> Specify the location on this server: [/usr/tmp]:

Your input: /usr/tmp

Is this correct? <Yes/No> [Yes]:

A temporary location must be specified on dbvcloud03 where the database backup will be copied to before moving to specified locations.

=> Specify location on remote server: [/usr/tmp]:

Your input: /usr/tmp

Is this correct? <Yes/No> [Yes]:

The following Oracle database parameters will be set in the standby database pfile or spfile:

SID NAME VALUE
* audit_file_dest /u01/app/oracle/
admin/SAM/adump
* compatible 11.2.0.4.0

```

*          control_files                /u01/app/oracle/
oradata/SAM/control01.ctl,/u01/app/oracle/fast_recovery_area/SAM/
control02.ctl
*          db_block_size                8192
*          db_domain
*          db_name                      SAM
*          db_recovery_file_dest        /u01/app/oracle/
fast_recovery_area
*          db_recovery_file_dest_size   4007657472
*          db_unique_name              SAM
*          diagnostic_dest              /u01/app/oracle
*          dispatchers                  (PROTOCOL=TCP)
(SERVICE=SAMXDB)
*          log_archive_format           %t_%s_%r.arc
*          open_cursors                 300
*          pga_aggregate_target         268435456
*          processes                    150
*          remote_login_passwordfile    EXCLUSIVE
*          sga_target                   805306368
*          spfile                       OS default
*          undo_tablespace              UNDOTBS1

```

What would you like to do:

- 1 - Proceed with creating the standby database
 - 2 - Edit Oracle database parameters for the standby database pfile/
spfile
 - 3 - Terminate processing
- Please enter your choice [1]: 2

WARNING: Do NOT use single or double quotes to enclose parameter values!

Separate multiple values with a comma. If a comma is part of a value, escape it with a backslash.

=> Enter full parameter name: []: control_files

Leave blank to create Oracle Managed Control File(s) if creating of OMF files is enabled

- 1 - Remove from the standby parameter file
(parameter will be set to default value)
- 2 - New value in the standby parameter file

=> Please enter your choice: []: 2

=> Enter new value (leave blank to set to null) []: /u01/app/oracle/
oradata/SAM/control01.ctl,/u03/app/oracle/fast_recovery_area/SAM/
control02.ctl

Validating ... please wait

The following Oracle database parameters will be set in the standby database pfile or spfile:

```

-----
SID                NAME                VALUE
*
*      audit_file_dest                /u01/app/oracle/
admin/SAM/adump
*      compatible                    11.2.0.4.0
*      control_files                  /u01/app/oracle/
oradata/SAM/control01.ctl,/u03/app/oracle/fast_recovery_area/SAM/
control02.ctl
*      db_block_size                  8192
*      db_domain
*      db_name                        SAM
*      db_recovery_file_dest          /u01/app/oracle/
fast_recovery_area
*      db_recovery_file_dest_size     4007657472
*      db_unique_name                SAM
*      diagnostic_dest                /u01/app/oracle
*      dispatchers                    (PROTOCOL=TCP)
(SERVICE=SAMXDB)
*      log_archive_format             %t_%s_%r.arc
*      open_cursors                   300
*      pga_aggregate_target           268435456
*      processes                      150
*      remote_login_passwordfile      EXCLUSIVE
*      sga_target                     805306368
*      spfile                          OS default
*      undo_tablespace                UNDOTBS1
-----

```

What would you like to do:

- 1 - Proceed with creating the standby database
- 2 - Edit Oracle database parameters for the standby database pfile/spfile
- 3 - Terminate processing

Please enter your choice [1]: 2

WARNING: Do NOT use single or double quotes to enclose parameter values!
Separate multiple values with a comma. If a comma is part of a value,
escape it with a backslash.

=> Enter full parameter name: []: db_recovery_file_dest

- 1 - Remove from the standby parameter file
(parameter will be set to default value)
- 2 - New value in the standby parameter file

=> Please enter your choice: []: 2

=> Enter new value (leave blank to set to null) []: /u03/app/oracle/
fast_recovery_area

Validating ... please wait

The following Oracle database parameters will be set in the standby
database pfile or spfile:

```

-----
SID                NAME                VALUE
*
*      audit_file_dest                /u01/app/oracle/
admin/SAM/adump
-----

```

```

*          compatible                                11.2.0.4.0
*          control_files                            /u01/app/oracle/
oradata/SAM/control01.ctl,/u03/app/oracle/fast_recovery_area/SAM/
control02.ctl
*          db_block_size                           8192
*          db_domain
*          db_name                                  SAM
*          db_recovery_file_dest                    /u03/app/oracle/
fast_recovery_area
*          db_recovery_file_dest_size               4007657472
*          db_unique_name                           SAM
*          diagnostic_dest                          /u01/app/oracle
*          dispatchers                             (PROTOCOL=TCP)
(SERVICE=SAMXDB)
*          log_archive_format                       %t_%s_%r.arc
*          open_cursors                             300
*          pga_aggregate_target                     268435456
*          processes                                150
*          remote_login_passwordfile                EXCLUSIVE
*          sga_target                               805306368
*          spfile                                   OS default
*          undo_tablespace                          UNDOTBS1

```

What would you like to do:

- 1 - Proceed with creating the standby database
- 2 - Edit Oracle database parameters for the standby database pfile/spfile
- 3 - Terminate processing

Please enter your choice [1]:

Validating Oracle database parameters... please wait

=> SUCCEEDED

=> Create standby database template for SAM using provided answers?

[Y]: Y

Your input: 1

Is this correct? <Yes/No> [Yes]:

=> Continue with creating a standby database? (If No processing will terminate, the saved template will be available for future use) [Y]:

Your input: 1

Is this correct? <Yes/No> [Yes]:

>>> dbvctl will now run a pre-flight check for standby database creation. An attempt will be made to create a standby (s)pfile using Oracle standby database parameters, followed by trying to start the standby instance. If this step fails, then please double-check the following items before re-running dbvctl again:

1. Review the standby database parameters you have supplied and provide valid values unless a template is used.
2. Recreate the template to provide valid values for standby database parameters if a template is used.

```
>>> Running pre-flight check for standby creation, please wait... done
```

```
>>> Total database size for SAM is 1.38GB
```

```
>>> Backing up primary database...
```

```
    Backing up datafile 1... done
```

```
    Backing up datafile 2... done
```

```
    Backing up datafile 3... done
```

```
    Backing up datafile 4... done
```

```
>>> Creating standby control file... done
```

```
>>> Transferring backup from dbvlab01 to dbvcloud03...
```

```
    Transferring /usr/tmp/dbv_SAM_csd_dbf_1_07s7bhj4_1_1.rman... done
```

```
    Transferring /usr/tmp/dbv_SAM_csd_dbf_1_08s7bhjt_1_1.rman... done
```

```
    Transferring /usr/tmp/dbv_SAM_csd_dbf_2_09s7bhjv_1_1.rman... done
```

```
    Transferring /usr/tmp/dbv_SAM_csd_dbf_3_0as7bhke_1_1.rman... done
```

```
    Transferring /usr/tmp/dbv_SAM_csd_dbf_4_0bs7bhkg_1_1.rman... done
```

```
>>> Restoring standby control files... done
```

```
>>> Starting standby database SAM on ocloud mount... done
```

```
>>> Restoring datafiles on ocloud...
```

```
    Restoring datafile 1... done
```

```
    Restoring datafile 2... done
```

```
    Restoring datafile 3... done
```

```
    Restoring datafile 4... done
```

```
>>> Renaming standby redo logs and tempfiles on ocloud... done
```

```
>>> Performing checkpoint and archiving logs... done
```

```
>>> Completing standby database creation...
```

```
>>> Standby database created.
```

To complete creating standby database please run dbvctl on the primary server first, then on the standby server, to ensure the standby database is in sync with the primary database.

```
PID:5691
```

```
TRACE:5691_dbvctl_csd_SAM_2012206211045.trc
```

15. The standby database is now created in the Oracle Cloud environment. The last step to be performed is having the logs automatically shipped and applied on the standby database. A daemon (background) process will monitor and ship as soon as the archive log is created. The daemon process would have to be started on each side.

```
[oracle@on-prem]$ ./dbvctl -d SAM -D status
Dbvisit Daemon process is not running.
[oracle@on-prem]$ ./dbvctl -d SAM -D start
Starting Dbvisit Daemon... Started successfully.
[oracle@on-prem]$ ./dbvctl -d SAM -D status
Dbvisit Daemon process is running with pid 1232.
```

```
[oracle@ocloud]$ ./dbvctl -d SAM -D status
Dbvisit Daemon process is not running.
[oracle@ocloud]$ ./dbvctl -d SAM -D start
Starting Dbvisit Daemon... Started successfully.
[oracle@ocloud]$ ./dbvctl -d SAM -D status
Dbvisit Daemon process is running with pid 8421.
```

As shown above, setting up Dbvisit Standby is a straightforward affair. Once properly installed, testing the software and going over the major features is recommended to familiarize yourself with them.

Switchover

The Graceful Switchover feature or role reversal is certainly one feature to be tested. As mentioned above, this nondestructive command, takes the primary database and makes it the standby database while the standby database becomes the primary database. This action does require a small outage on the primary database while the reversal takes place. One thing to note is that Dbvisit Standby does NOT point your application to the standby database and this action will have to take place outside of Dbvisit.

Steps needed to execute a Graceful Switchover (GS):

You will need to stop the Dbvisit daemon process to stop shipping archive logs

```
[oracle@test standby]$ ./dbvctl -d test -D stop
Stopping Dbvisit Daemon... Successfully stopped.
```

Before you execute a Graceful Switchover you must make sure that the Archive Log Gap between the primary and standby is 0. The command is run with the `/dbvctl -d <ddc_filename> -i`. Below it is shown with a DDC filename of test.

```
oracle@test:/u01/app/dbvisit/standby/conf//u01/app/dbvisit/standby/
dbvctl -d test -i
=====
Dbvisit Standby Database Technology (8.0.xx_0_) (pid 21204)
dbvctl started on test: Wed Dec 28 13:31:47 2018
=====

Dbvisit Standby log gap report for test thread 1 at 203203221670:
-----
Destination database on dbv102 is at sequence: 221.
Source database on dbv101 is at log sequence: 222.
Source database on dbv101 is at archived log sequence: 221.
Dbvisit Standby last transfer log sequence: 221.
Dbvisit Standby last transfer at: 2018-12-28 13:29:14.

Archive log gap for thread 1: 0.
```

```
Transfer log gap for thread 1: 0.
Standby database time lag (DAYS-HH:MI:SS): +00:01:22.
```

```
=====
dbvctl ended on DBVP: Wed Dec 28 13:31:47 2018
=====
```

As shown above there is no gap.

```
[oracle@dbv101 /usr/dbvisit/standby]$ ./dbvctl -d test -o switchover
=====
Dbvisit Standby Database Technology (8.0.xx) (pid 577) dbvctl started
on dbv101: Wed Dec 28 13:37:49 2018
=====
>>> Starting Switchover between dbv101 and dbv102
Running pre-checks          ... done
Pre processing              ... done
Processing primary          ... done
Processing standby          ... done
Converting standby         ... done
Converting primary         ... done
Completing                  ... done
Synchronizing              ... done
Post processing             ... done
>>> Graceful switchover completed.
Primary Database Server: dbv102
Standby Database Server: dbv101
  >>> Dbvisit Standby can be run as per normal:      dbvctl -d test
PID:577 TRACE:/usr/dbvisit/standby/trace/578_dbvctl_switchover_
test_201849071037.trc
=====
dbvctl ended on dbv101: Wed Dec 28 13:39:42 2018
=====
[oracle@dbv101 /usr/dbvisit/standby]$
```

At this point the databases have reversed roles. The above action time may vary depending on archive log size, network speed and other factors.

Failover

When disaster strikes and the primary database is no longer available, the standby database must be activated to become the new primary database so it can continue operation. This is called failover or failover to the standby database.

The steps to activate the standby database as the new primary database are:

1. Stop the scheduling of Dbvisit Standby.
2. Change the network configuration (or DNS) so that users will connect to the standby database (or server) instead of the primary database (or server). This is done outside of Dbvisit.
3. Activate and open the standby database for normal operation as per instructions below. Note that as soon as the standby database is activated and becomes the new primary database, that is open read/write. The link to the original primary database is lost and it is no longer possible to apply new logs to the original standby database. This is a destructive operation.

Prior to activating the standby database, you may want to run a test. You can do this by seeing if you can open the standby database in read only mode first. If this is successful, you can proceed to opening it up in read write mode. If it is unsuccessful you may need to have more archive logs applied.

```
[oracle@test /usr/dbvisit/standby]$ ./dbvctl -d DEV -o read
=====
dbvctl started on dbv2: Dec Nov 28 13:51:23 2018
=====
Open Standby Database DEV in READ ONLY mode...
Standby Database DEV opened in READ ONLY mode.
Log files cannot be applied to Database while in READ ONLY mode.
Database tempfile(s) may need to be added to this database.
=====
dbvctl ended on dbv2: Wed Dec 28 13:51:24 2018
=====
```

Once successful, set it back:

```
[oracle@dbv2 /usr/dbvisit/standby]$ ./dbvctl -d DEV -o restart
=====
Dbvisit Standby Database Technology (8.0.xx) (pid 3335)
dbvctl started on dbv2: Wed Dec 13 13:52:11 2018
=====
Stopping instance DEV...
Standby Instance DEV shutdown successfully on dbv2.
Starting instance DEV... Standby Instance DEV started
=====
dbvctl ended on dbv2: Dec 28 13:52:19 2018
=====
```

Once it is back to recovery mode, it is time to activate the standby database:

```
[oracle@dbv2 /usr/dbvisit/standby]$ ./dbvctl -d DEV -o activate
=====
Dbvisit Standby Database Technology (8.0.xx) (pid 3420)
dbvctl started on dbv2: Dec 28 13:52:38 2018
=====
Activating means this database will become a Primary Database. It will no
longer be a Standby Database for DEV on dbv1. Activation cannot be reversed.
=> Activate Standby Database on dbv2? <Yes/No> [No]: Yes
Are you sure? <Yes/No> [No]: Yes
>>> Activating now... File /usr/dbvisit/standby/conf/dbv_DEV.env copied to /
usr/dbvisit/standby/conf/dbv_DEV.env.201611131052.
>>> Activation complete. Please ensure a backup of this Database is made
>>> Old archives from before the activation should be removed to avoid mix-up
between new and old archive logs
>>> If the Dbvisit Standby process is to be reversed, then database on dbv1
will need to be rebuilt as a Standby Database
=====
dbvctl ended on dbv2: Dec 28 13:53:35 2018
=====
```

Note that it will ask you if you are sure you want to action this. Because the activation is a destructive action, this should only be done in a true disaster. It also makes note of recommending that you backup your database and make a new standby database as soon as possible. Outside of Dbvisit Standby, the application will need to be pointed to this new primary database.

Conclusion

Disasters, whether naturally occurring, man-made, human errors or hardware failures, do happen. With so much of the business value tied up in the data of organizations, it is paramount that this data be backed up and protected. Backups, while important, can not offer the RTOs and RPOs that modern digital enterprises require. For this reason, standby databases are essential to meet the needs of organizations so that the databases, filled with critical information, are available at all times. Dbvisit Standby provides that level of readiness for Oracle Standard Edition Databases.

Having your standby database in the Oracle Cloud makes perfect sense. You can place your DR environment in a different geographic location at a cost effective price, without having to buy new hardware and worry about where to fit it into your data center.



 www.dbvisit.com

 info@dbvisit.com

USA: 1-800-933-8007

Rest of the world: +64 9 950 3301

 **Gold Partner**


DISASTER RECOVERY SOFTWARE

ÜBER HUNKLER

Das Karlsruher Systemhaus HUNKLER wurde 1988 erster offizieller Partner von Oracle in Deutschland. Ein Team von rund 20 Mitarbeitern unterstützt Kunden aus Industrie, öffentlicher Verwaltung, Gesundheits- und Finanzwesen mit Beratung, Lösungsentwicklung und Managed Services.

Im Fokus von HUNKLER stehen leistungsfähige, wirtschaftliche Infrastrukturen für Oracle-datenbanken mit den Schwerpunkten Hochverfügbarkeit, Ausfallsicherheit und Zero Downtime Migration. Die integrierten Komplettlösungen der Produktfamilie Oracle Engineered Systems sowie der Datenbank-/Anwendungsbetrieb in der Oracle Cloud sind weitere Themenfelder, die das Unternehmen umfassend abdeckt.

HUNKLER
GmbH & Co. KG

Hauptsitz Karlsruhe

Bannwaldallee 32, 76185 Karlsruhe
Tel. 0721-490 16-0, Fax 0721-490 16-29

Geschäftsstelle Bodensee

Fritz-Reichle-Ring 6a
78315 Radolfzell
Tel. 07732-939 14-00, Fax 07732-939 14-04
info@hunkler.de, www.hunkler.de