An Oracle White Paper
May 2012

Multiple Public Networks in Private Database Clouds

1. Executive Overview

In a cloud deployment, all tenants share the same underlying software and hardware infrastructure. However, all tenants don't necessarily place similar consumption requirements on the underlying cloud infrastructure. For example, some tenants are business critical and therefore have strict service-level-agreements (SLAs) that must be met. Therefore, these tenants may require re-provisioning existing capacity or provisioning additional capacity to the cloud infrastructure to satisfy their SLA objectives.

Other tenants may contain governmental and regulatory compliant data and therefore must be isolated from other classes of tenants in the same cloud. Therefore, these tenants may require strong isolation capabilities, such as *Network Isolation (1)*, from the underlying cloud infrastructure.

In another scenario, as a cloud provider, to continue to meet the performance and quality-of-service (QoS) of your tenants, it may be necessary to dynamically provision additional cloud infrastructure components, such as multiple public networks, to carry out operational tasks, such as data load, backup, restore, etc., so as to minimize the performance impact on the cloud infrastructure in-use by the current tenants.

In this whitepaper, we will discuss the best practices for dynamically provisioning multiple public networks in a *Cloud Pool* [1] of a Private Database Cloud deployment. These additional public networks can be used for isolating network traffic in or adding network bandwidth to your existing cloud infrastructure.

---

[1] **CLOUD POOL** - *In an Oracle Engineered System deployment, a Cloud Pool refers to a pool of servers and storage cells carved out into a separate cluster. In all other deployments, a Cloud Pool refers to an Oracle11gR2 or later version of Grid Infrastructure deployment on a pool of servers having access to shared storage.*

2. Introduction

In the *Database Cloud Deployment Models (3)*, there are two models where multiple public networks become important due to many business and operational requirements. The business requirements may include network traffic isolation, network I/O throughput, etc. The operational requirements may include leveraging additional public network bandwidth for jobs such as Extract-Transform-Load (ETL), Data backup, Data replication, Data restore, etc. to avoid consumption of existing network bandwidth that could impact SLAs of current tenants.

2.1 Database Consolidation

In the Database consolidation model, one database can offer multiple database services to serve different business needs. For example, a database could have two services, *Online* and *Backup*. The Online service could be handling the web traffic hitting the transaction data stored in a database. The Backup service could be responsible for actively backing up the database data.

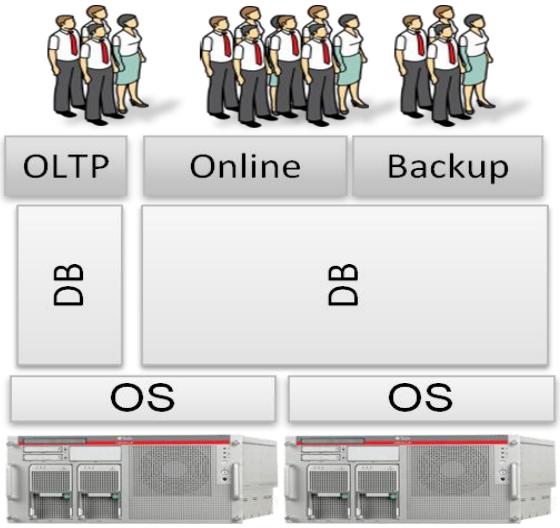


Figure 1. Database Consolidation model

If there is sufficient network bandwidth, these two services could handle all the data traffic over the same public network. However, in real world deployments and as per Oracle MAA best practices, it is always a best practice to separate the end-user/web facing traffic on its own high bandwidth and secure public network and dedicate a separate public network for internal operational needs such as backup and restore.

In another scenario of Database Consolidation model whereby multiple database workloads are consolidated onto a shared Cloud Pool, such as OLTP and non-OLTP workloads above, it may be necessary due to traffic bandwidth requirements to provision an additional public network and dedicate it to servicing the heavy OLTP traffic.

2.2 Schema Consolidation

In the Schema consolidation model, each database tenant is deployed as a schema in the same underlying database. The data belonging to each tenant is plugged in as a tablespace and schema access is offered via a database service. Therefore, as an example, one RAC database could offer three database services, HR, ERP and CRM, for three different schemas corresponding to Human Resources, Enterprise Resource Management, and Customer Relationship Management applications.

Figure 2. Schema Consolidation model

In the Schema consolidation model of database cloud deployment, it is important to isolate not only each tenant's data from that of other tenants', but also how that data is accessed by and delivered to each tenant's client applications. For example, a schema containing *Payment Card Industry* (PCI) standard compliant data needs to be appropriately masked and secured from schema containing other class of data, such as *Health Insurance Portability and Accountability Act* (HIPPA) compliant data, within the same database.

One can use software-based logical isolation techniques to isolate data corresponding to each tenant on the same public network. However, many customers attest to the fact that IT hardware resources are cheap whereas IT personnel are expensive. Therefore, a preferred solution for isolating the access to and delivering regulatory-compliant data is to deploy multiple public networks in the database cloud environment and configure the database environment such that a database client can access its respective schema services only on the public network dedicated for that service and client.

In this whitepaper, we will describe the end-to-end best practices for configuring multiple public networks in a Cloud Pool of Private Database Cloud deployment and how database clients should be set up to access database services on their respective public networks.

3. Base Configuration

We assume that our base configuration consists of a Cloud Pool of a two node cluster with deployment of Oracle11g Release 2.x Grid Infrastructure and Oracle11g Release 2.x RAC database. Therefore, as part of the Oracle11gR2 Grid Infrastructure deployment, a default public network (managed under *ora.net1.network* CRS resource), SCAN listeners, and node local listeners (managed under *ora.LISTENER.lsnr* CRS resource) have been configured on the 10.10.10.x network.

We will also assume that second public network hardware, such as Network switches, Network adaptors, NIC cards, power cables, etc. has been provisioned and named uniformly across our two nodes Cloud Pool.

3.1 DNS or /etc/hosts Entries

Assume that the following network names are defined for a two nodes cluster (in DNS or /etc/hosts):

```
# Public host name and associated VIP on eth1
10.10.10.1            node1n1
10.10.10.11           node1n1-vip
10.10.10.2            node2n1
10.10.10.22           node2n1-vip

# scan name
10.10.10.41           racscan
10.10.10.42           racscan
10.10.10.43           racscan
```

We are going to set up second public network on the 20.20.20.x on a network adaptor named *eth2* uniformly within a Cloud Pool of two nodes cluster for a RAC database.

4. Configure Additional Public Networks

4.1 Define Host and VIP IP Addresses on new Public Network

In this example, we are going to define following host and VIP names on the second public network name *eth2* (in DNS or /etc/hosts file):

```
# host name for the 2nd public network and associated VIP on
eth2
20.20.20.1            node1n2
20.20.20.11           node1n2-vip
20.20.20.2            node2n2
20.20.20.22           node2n2-vip
```

4.2 Configure CRS Resources for Second Public Network

In order to manage the IP addresses on the second public network as HA resources, we need to define CRS resources for the network and VIP resources on this second network. This is done by using the following *srvctl* commands from GRID_HOME/bin directory.

As privileged user(root on Unix or Administrator on Windows):
 # srvctl add vip -n <node_name> ***-k <network_number>*** *-A <name|ip>/<netmask>/[if1[|if2...]] [-v]*

eg:
 # srvctl add vip -n racnode1 ***-k 2*** *-A node1n2-vip/255.255.255.0/eth2*
 # srvctl add vip -n racnode2 ***-k 2*** *-A node2n2-vip/255.255.255.0/eth2*

The first command will implicitly create the dependent *ora.net2.network* resource.

NOTE: From Oracle11gR2 11.2.0.2 onwards, network resource can be created explicitly using srvctl command.

As privileged user (root on UNIX or Administrator on Windows):
 # srvctl add network [-k <net_num>] -S <subnet>/<netmask>/[if1[|if2...]] [-w <network_type>] [-v]

eg:
 # srvctl add network ***-k 2*** *-S 20.20.20.0/255.255.255.0/eth2*

Then add vip resource on the 2nd network:
 # srvctl add vip -n racnode1 ***-k 2*** *-A node1n2-vip/255.255.255.0/eth2*
 # srvctl add vip -n racnode2 ***-k 2*** *-A node2n2-vip/255.255.255.0/eth2*

The CRS resource for the second public network, 'ora.net2.network', and two VIP resources on that network have been created now. You can verify their existence by running the following command from GRID_HOME/bin directory:

```
# crsctl stat res -t |grep -E 'net|vip'
ora.net1.network
ora.net2.network
ora.node1n1.vip
ora.node1n2-vip.vip
ora.node2n1.vip
ora.node2n2-vip.vip
...
```

5. Configure Listener on Second Public Network

5.1 Configure Listener

The following two tools can be used to configure new listener on the second public network that we configured above.

5.1.1 SRVCTL

This is a command-line tool that provides options for simple configuration of listeners. As grid home owner user, run the following commands from GRID_HOME/bin directory.

```
% ./srvctl add listener -l LISTENER_2n -s -p 1521 -k 2
% ./srvctl start listener -l LISTENER_2n
```

NOTE: The "-s" option to "srvctl add listener" command is needed as a workaround for the following known bug:

Bug 13899911 - SRVCTL ADD LISTENER FAILS ON NON-DEFAULT NETWORK

Once this bug is fixed in a particular release, then "-s" option does not need to be specified in above command.

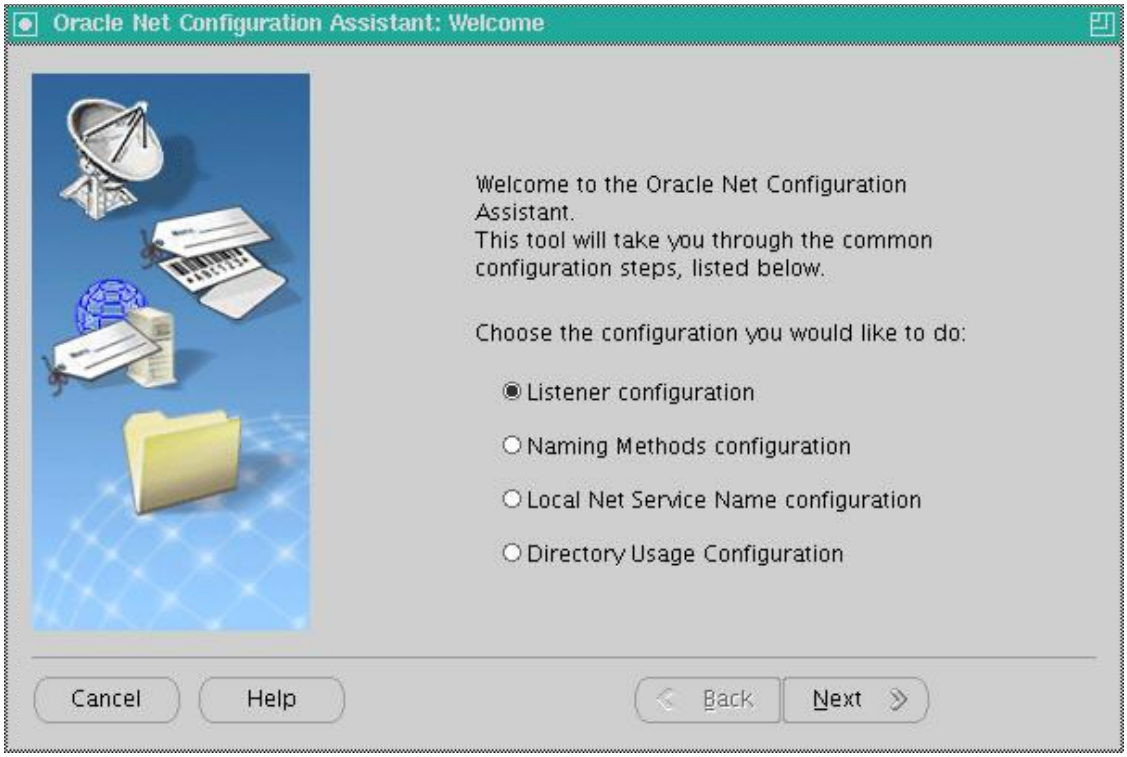
5.1.2 NETCA

The NetCA is a Graphical User Interface (GUI) tool that offers choices from simple configuration to very advanced configuration of listeners. As the grid home owner user, invoke "netca" from the GRID_HOME/bin directory.

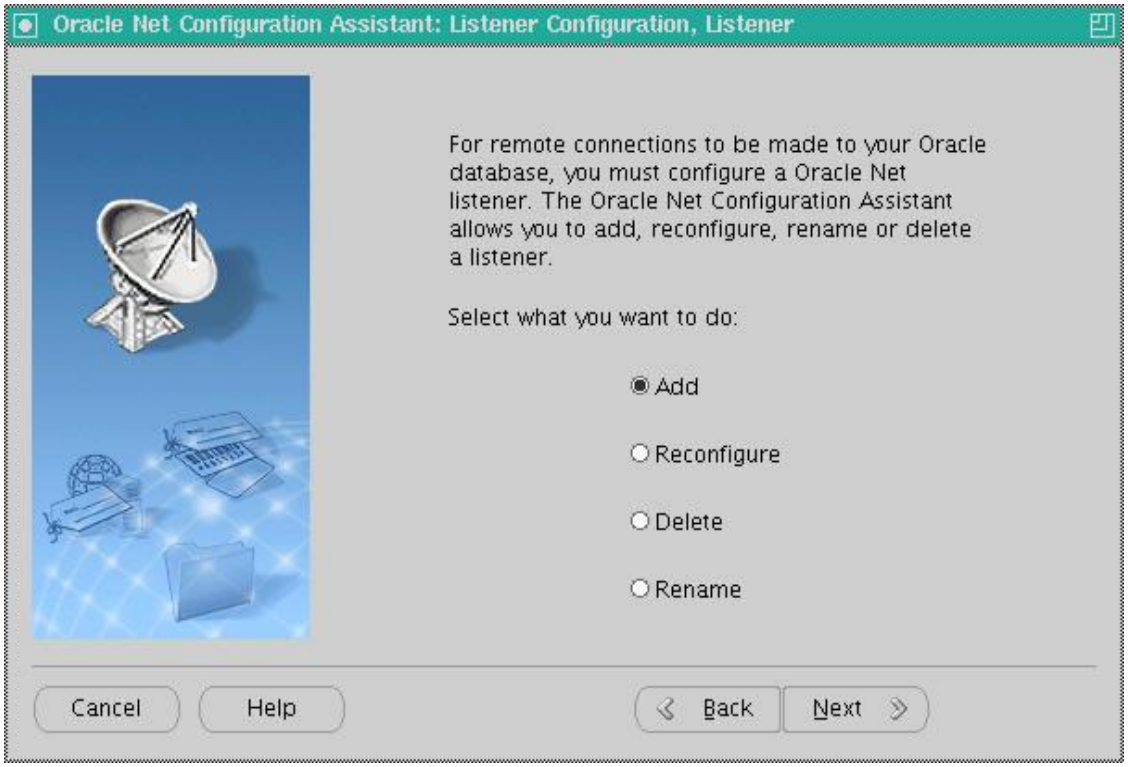
NOTE: Because of the current known bug-13899911 mentioned above, if there is an existing listener listening on the port (e.g. 1521), then netca cannot be used to configure another listener to listen on the same port irrespective of the underlying public network. Since we already have an existing listener, LISTENER, listening on port 1521 on the default public network (managed under ora.net1.network CRS resource), the following screenshots are for informational purpose only as there is no workaround for the bug-13899911 in netca.

```
% ./netca
```

1) On Welcome screen, select *Listener configuration* and click *Next*.



2) On Listener configuration screen, select *Add* and click *Next*.

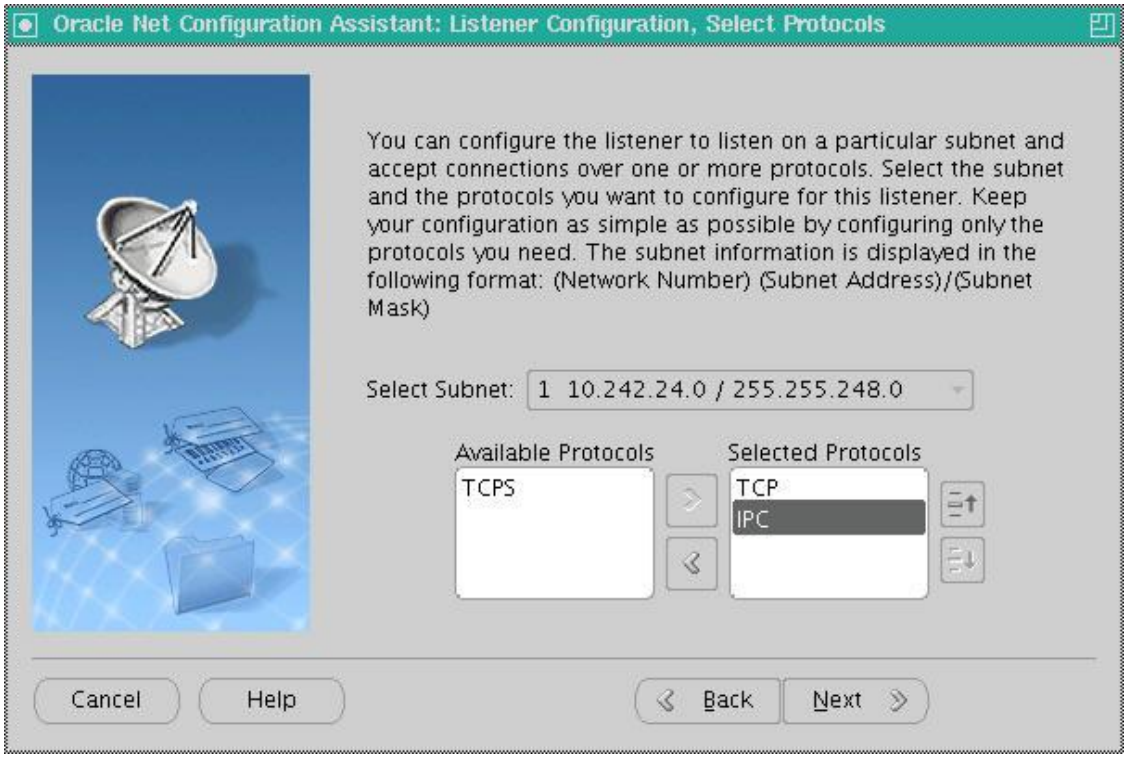


3) Enter the listener name, *LISTENER_2N*, and click *Next.*

4) Select the Subnet, the subnet "*2*" will appear in the drop down list. For example:

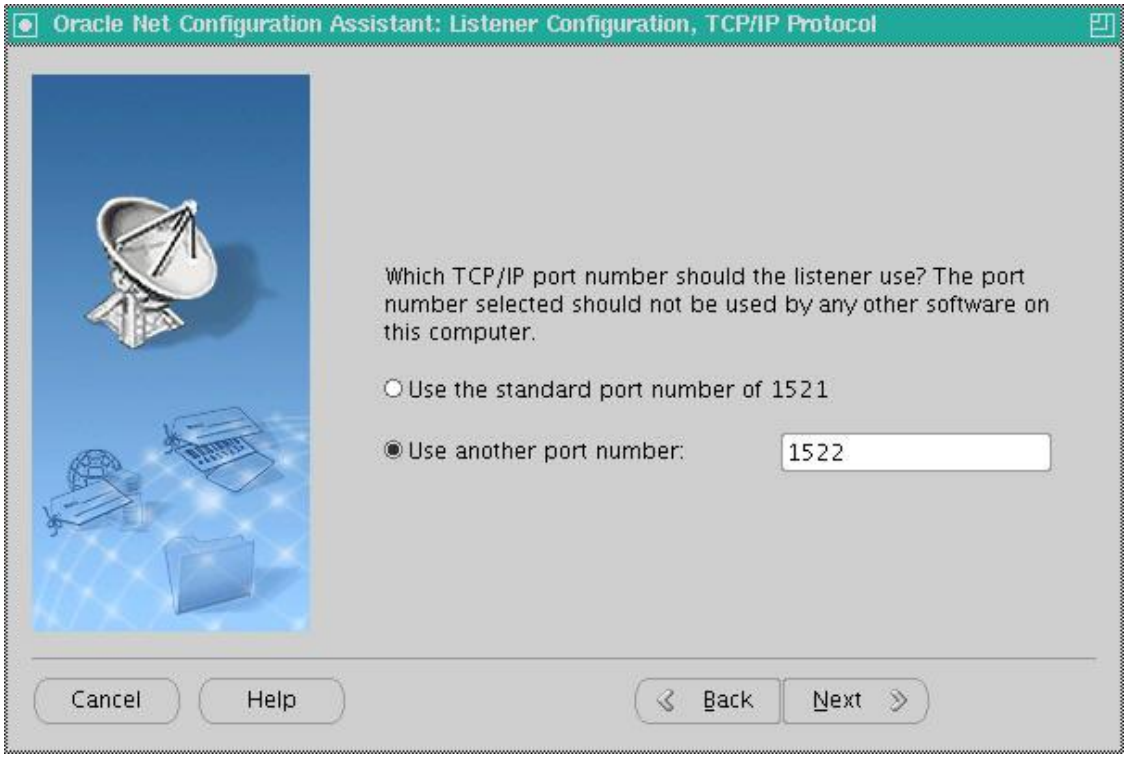
```
2 20.20.20.0/255.255.255.0
1 10.10.10.0/255.255.255.0
```

Also select the listener Network protocols, such as *TCP and IPC*, and click *Next*.

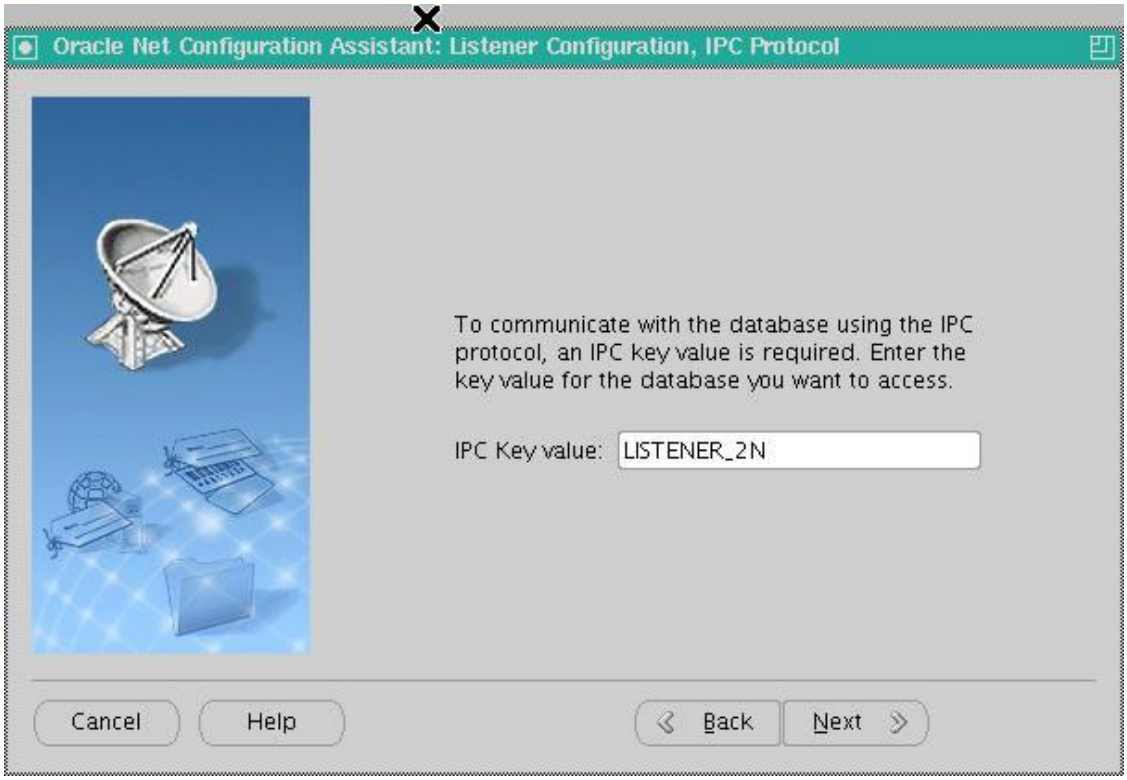


5) Select the listener port number to use. Because of the known bug, **Bug 13899911 - SRVCTL ADD LISTENER FAILS ON NON-DEFAULT NETWORK**

we have to select “Use another port number” option as we already have an existing listener listening on port 1521. Enter a port number, *1522*, and click *Next*.



6) Enter IPC key. It can be same as the listener name, LISTENER_2N, and click *Next*.

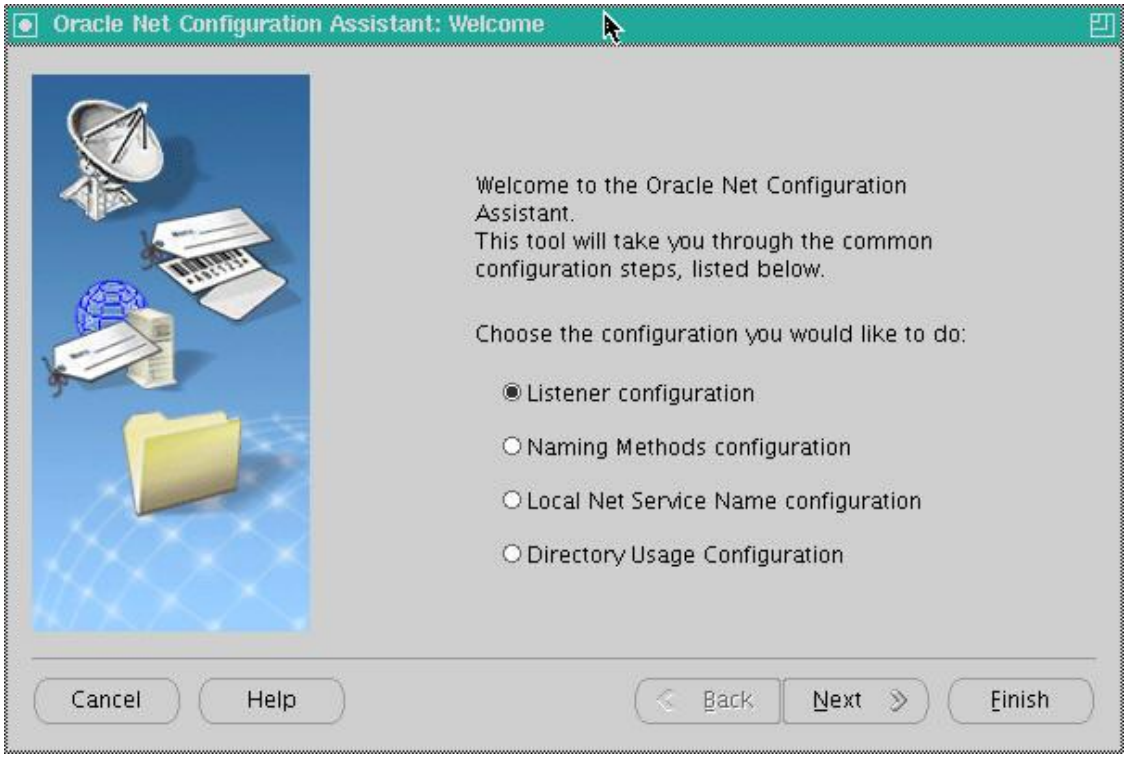


7) Select *No* on Configure more listeners screen and click *Next*.

8) From the drop down list, select the listener to start, *LISTENER_2N*, and click *Next.*

9) Once listener LISTENER_2N has been started successfully on all nodes of the Cloud Pool, the following screen will appear. Click *Next* to go to the beginning, the step 1.

10) Click *Finish* to exit the netca GUI.



5.2 Verify Listener Configuration

To continue our discussion further, we assume that LISTENER_2N was configured to listen on port 1521 on second public network using the srvctl command.
For verifying that the new listener is listening on the VIP address node1n2-vip (or node2n2-vip for node 2) associated with the second network, run the following command from GRID_HOME/bin directory.

```
% ./lsnrctl status listener_2n

LSNRCTL for Linux: Version 11.2.0.1.0 - Production on 08-FEB-2010
16:10:23

Copyright (c) 1991, 2009, Oracle. All rights reserved.

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=LISTENER_2n)))
STATUS of the LISTENER
------------------------
Alias LISTENER_2n
Version TNSLSNR for Linux: Version 11.2.0.1.0 - Production
Start Date 08-FEB-2010 16:10:20
Uptime 0 days 0 hr. 0 min. 2 sec
Trace Level off
Security ON: Local OS Authentication
```

```
SNMP OFF
Listener Parameter File
/u02/app/11.2.0/grid/network/admin/listener.ora
Listener Log File
/u02/app/oragrid/diag/tnslsnr/racnode1/listener_2n/alert/log.xml
Listening Endpoints Summary...
(DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=LISTENER_2n)))
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=10.10.10.1)(PORT=1521)))
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=20.20.20.11)(PORT=1521)))
The listener supports no services
The command completed successfully
```

Please note the "*lsnrctl status*" output for 11.2.0.2 or later version listener shows two ADDRESS endpoints (IPC and VIP endpoints) for listener configured on the non-default public network:

```
(DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=LISTENER_2n)))
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=20.20.20.11)(PORT=1521)))
```

The (HOST=*hostname*) endpoint is excluded from the endpoints of the non-default network listener because host name/IP of the node is available on the default network only.

6. Configure Database Services on Second Public Network

6.1 Server-side Service Configuration

6.1.1 Policy-managed Database Configuration

In a policy-managed database configuration, define the database services so that these services depend upon the second public network that we have configured above. As ORACLE_HOME owner user for the database, run the following command from ORACLE_HOME/bin directory to define new service(s) as follows:

% ./srvctl add service –d <db_unique_name> -s <my_service> -k <net_num> …

A concrete example of above command would be:
% ./srvctl add service -d mydb -s ***myservice2n*** *-k 2 -g my_pool -c UNIFORM*

6.1.2 Administrator-managed Database Configuration

Currently, it is not possible to define service-to-network dependencies in an administrator-managed database configuration. However, we are aware of this limitation and have reported the following enhancement request for development to help bring functional parity with policy-managed database configuration:

BUG13847318 - SUPPORT MULTIPLE NETWORKS IN ADMINISTRATOR - MANAGED CONFIGURATION

6.2 Client-side Service Configuration

Depending upon the service name resolution method that you are using for your database, LDAP, TNSNAMES, etc., you need to define the net service name entry for the service defined above on the new network. For example, if you are using tnsnames.ora for service name resolution, then add the following entry to the ORACLE_HOME/network/admin/tnsnames.ora file:

```
myservice2n.us.example.com =
(DESCRIPTION =
    (ADDRESS=(PROTOCOL=TCP)(HOST=node1n2-vip)(PORT=1521))
    (ADDRESS=(PROTOCOL=TCP)(HOST=node2n2-vip)(PORT=1521))
    (LOAD_BALANCE=yes)
    (CONNECT_DATA=
        (SERVER = dedicated)
        (SERVICE_NAME = myservice2n.us.example.com)
    )
 )
```

7. Register Database Services with new Network Listener

7.1 Configure Address Aliases for new Network Listener

Depending upon the listener address resolution method that you are using for your database deployment, LDAP, TNSNAMES, etc., we recommend that you define the local and remote listeners address entries for listeners defined above on the second public network. For example, if you are using tnsnames.ora for name resolution, then add the following entries to ORACLE_HOME/network/tnsnames.ora file.

Assuming that ORACLE_HOME is private to each node, then on node1:

```
listener_net2 =(ADDRESS = (PROTOCOL = TCP)(HOST = node1n2-vip)(PORT =
1521))

remote_net2 =
            (ADDRESS_LIST =
              (ADDRESS=(PROTOCOL=TCP)(HOST=node1n2-vip)(PORT=1521))
              (ADDRESS=(PROTOCOL=TCP)(HOST=node2n2-vip)(PORT=1521))
            )
```

And on node 2:

```
listener_net2 =(ADDRESS = (PROTOCOL = TCP)(HOST = node2n2-vip)(PORT =
1521))

remote_net2 =
            (ADDRESS_LIST =
             (ADDRESS=(PROTOCOL=TCP)(HOST=node1n2-vip)(PORT=1521))
             (ADDRESS=(PROTOCOL=TCP)(HOST=node2n2-vip)(PORT=1521))

            )
```

7.2 Server-side Configuration

In order to ensure that database registers its services with the listener on the second public network and that connection requests to the remote listener are only redirected to the local listener on that same network, the LISTENER_NETWORKS parameter needs to be set in the database parameters file (pfile or spfile) to define local and remote listeners pairs on each non-default public network.

```
SQL> alter system set
LISTENER_NETWORKS='((NAME=network2)(LOCAL_LISTENER=listener_net
2)(REMOTE_LISTENER=remote_net2))' SCOPE=BOTH SID='*';
```

NOTE: If you had defined LISTENER_NETWORKS parameter, then you should query the existing value of this parameter and concatenate it to the new network name-value pairs. For example, if we were to define third public network for our RAC database, then we would execute the following commands:

```
Query existing value of LISTENER_NETWORKS parameter:

SQL> select value from v$parameter where name='listener_networks';

VALUE

--------------------------------------------------------------------------------

((NAME=network2) (LOCAL_LISTENER=listener_net2)
(REMOTE_LISTENER=remote_net2))

Assume that we have defined address entries for local and
remote listeners in tnsnames.ora for listener on third
public network like those for our second network listener.
Then we'd set LISTENER_NETWORKS as follows:
```

```
SQL> alter system set
LISTENER_NETWORKS='((NAME=network2)(LOCAL_LISTENER=listener_net2)(RE
MOTE_LISTENER=remote_net2))',
'((NAME=network3)(LOCAL_LISTENER=listener_net3)(REMOTE_LISTENER=remote_
net3))' SCOPE=BOTH SID='*';
```

8. Access Database Services via Multiple Public Networks

At this stage of our discussion, we have two public networks in our two node RAC database deployment. Therefore, our two instance RAC database is offering its services on multiple public networks configured on the cluster. However, there are differences in how these services should be accessed. The following two cases describe how services should be accessed on these public networks.

8.1 Access Services via Default Public Network

The public network managed under the *ora.net1.network* resource in an Oracle11g Release 2.x Grid Infrastructure deployment is called the default public network. This is the network on which the cluster SCAN (Single Client Access Name) is configured. All databases register with SCAN listeners by default. Therefore, all clients that need to access database services on the default public network must use the SCAN listeners. This is done by using the EZConnect syntax as follows:

```
        <scan_name>[:<scan_port>]/<service_name>
```

For our running example, an OCI client (e.g. SQL*Plus) connects to *myservice* using SCAN on the default network as follows:

```
% sqlplus
system/manager@racscan:1521/myservice.us.example.com
```

A JDBC client accesses *myservice* using SCAN on default network by using the following connect URL:

```
jdbc:oracle:thin:@racscan:1521/myservice.us.example.com
```

8.2 Access Services via Non-Default Public Networks

The public networks that are managed under CRS resources other than the *ora.net1.network* resource in an Oracle11g Release 2.x Grid Infrastructure deployment are called the non-default public networks. This includes the *ora.net2.network* resource that we defined above. In an Oracle11g Release 2.x deployment, the cluster SCAN is defined only on the default public network and SCAN can only redirect service connection requests to the database listeners that are on the same public network. Therefore, database services that are registered on the non-default network must not be accessed using cluster SCAN. Instead, these services must be accessed using the pre-Oracle11g Release 2.x syntax.

For our running example, an OCI client (e.g. SQL*Plus) connects to *myservice2n* on the second public network as follows:

```
% sqlplus system/manager@myservice2n.us.example.com
```

A JDBC client accesses *myservice2n* on the second public network by using the following connect URL:

```
jdbc:oracle:thin:@(DESCRIPTION =
    (ADDRESS=(PROTOCOL=TCP)(HOST=node1n2-vip)(PORT=1521))
    (ADDRESS=(PROTOCOL=TCP)(HOST=node2n2-vip)(PORT=1521))
    (LOAD_BALANCE=yes)
    (CONNECT_DATA=
        (SERVER = dedicated)
        (SERVICE_NAME = myservice2n.us.example.com)
    )
 )
```

9. Conclusion

To ensure that your cloud infrastructure is agile and continues to meet your business objectives, it is necessary to re-provision the existing infrastructure components or even provision brand new infrastructure components into existing cloud environment. This introduces the following two challenges:

1) Re/provision the infrastructure components so that there is no or minimal impact on existing cloud infrastructure components.
2) Dynamically exploit the new capacity made available by the new infrastructure components to achieve even greater performance levels.

In this white paper, we have discussed how organizations can provision multiple public networks in a Cloud Pool of their Private Database Cloud deployments and the configuration changes necessary on the server- and client-tiers to exploit the new public network capacity for meeting network throughput and traffic isolation objectives.

10. References

(1) Network Isolation in Private Database Clouds
http://www.oracle.com/technetwork/database/database-cloud/ntwk-isolation-pvt-db-cloud-1587225.pdf

(2) Private Database Cloud on Oracle Database Appliance
http://www.oracle.com/technetwork/database/database-cloud/privatedbcloudonoda-1522348.pdf

(3) Database Consolidation onto Private Clouds
http://www.oracle.com/us/products/database/database-private-cloud-wp-360048.pdf

Multiple Public Networks in Private Database Clouds
May 2012
Author: Raj K. Kammend
Contributing Authors: Tim Read, Burt Clouse, John McHugh, Nitin Vengurlekar, Troy Anthony

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200

oracle.com

Oracle is committed to developing practices and products that help protect the environment

Hardware and Software, Engineered to Work Together