ORACLE

Oracle Siebel CRM 22.3 Installation

Oracle Private Cloud Appliance

January 25, 2023 | Version 1.01

PURPOSE STATEMENT

This technical paper provides a methodology and workflow that solution architects and system administrators can use to successfully install, configure and deploy Oracle Siebel CRM 22.3 in an Oracle Private Cloud Appliance environment.

DISCLAIMER

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

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

CONTENTS

INTRODUCTION

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

SCOPE AND CONTENT

This technical paper provides a methodology and workflow that solution architects and system administrators can use to successfully install, configure and deploy Oracle Siebel CRM 22.3 in an Oracle Private Cloud Appliance environment.

For this qualification exercise, the installation of a minimal set of components was validated post-deployment. However, automation tooling referenced in this document does not include the installation of specific components.

ADVANTAGES OF ORACLE PRIVATE CLOUD APPLIANCE

Oracle Private Cloud Appliance (PCA) is an Oracle Engineered System designed for implementing the application and middleware tiers. PCA is an integrated hardware and software system that reduces infrastructure complexity and deployment time for virtualized workloads in private clouds. It is a complete platform that provides optimal performance for a wide range of application types and workloads, with built-in management, compute, storage, and networking resources.

Oracle Private Cloud Appliance X9-2 is the latest member of the Oracle Private Cloud Appliance product family. PCA provides cloud and administrative services for modernized cloud native applications. It makes use of a modern microservices architecture, Kubernetes, and related technologies, for a future-proofed software stack.

Oracle Private Cloud Appliance delivers private cloud infrastructure and architecture consistent with Oracle Cloud Infrastructure (OCI). PCA brings APIs and SDKs compatible with Oracle Cloud Infrastructure (OCI) to an on-premises implementation at rack scale, making workloads, user experience, tool sets, and skills portable between private and public clouds. PCA can also be directly connected to Oracle Exadata to create an ideal infrastructure for scalable, multitier applications. Customers preferring or requiring an on-premises solution can realize the operational benefits of public cloud deployments using Oracle Private Cloud Appliance.

ORACLE SIEBEL CRM MULTINODE INSTALLATION ON ORACLE PRIVATE CLOUD APPLIANCE

This technical paper describes a process for creating a sample Oracle Siebel CRM multinode It is broken down into 2 parts: Manual preparatory steps and utilizing Ansible playbook to install Oracle Siebel CRM.

REFERENCE ARCHITECTURE

The image below lays out the reference architecture utilized during this qualification exercise.

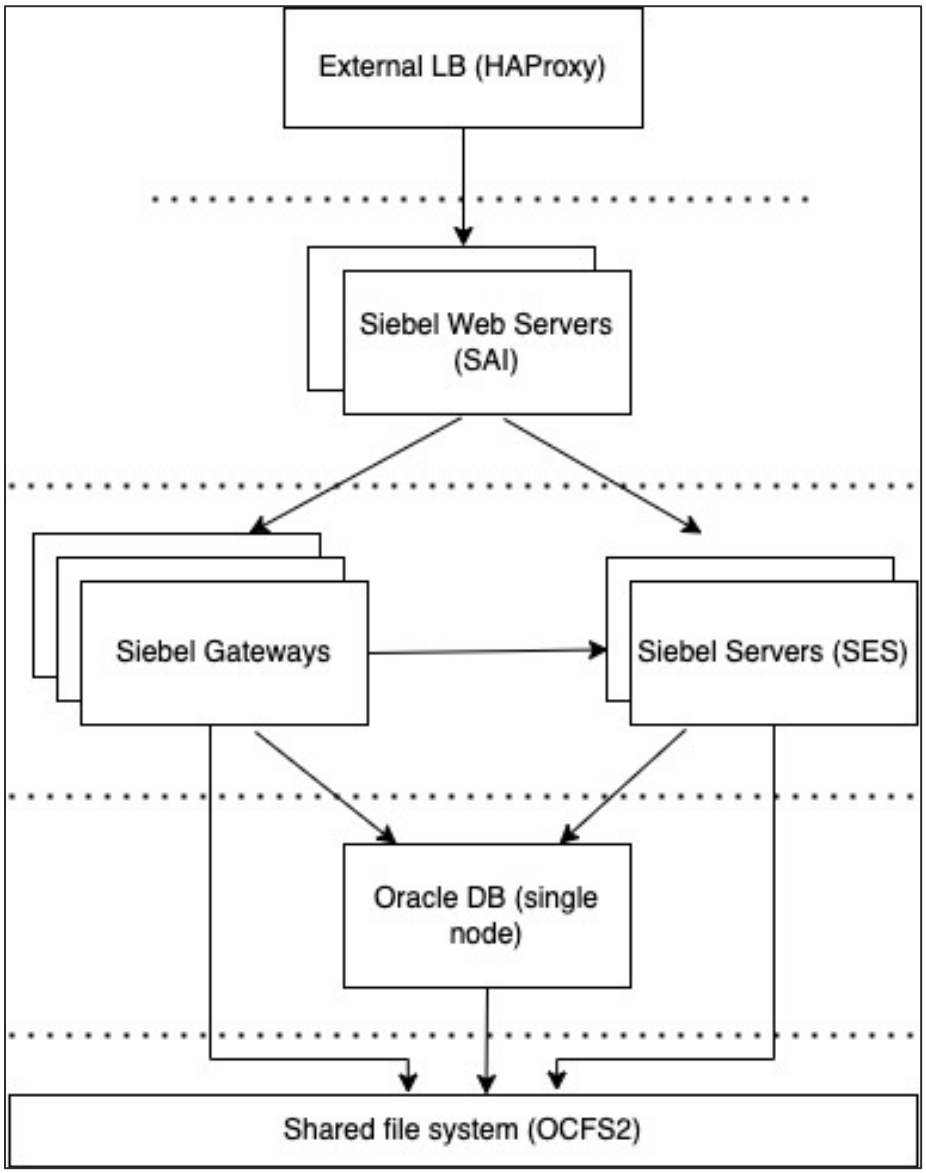

Fig 1: Oracle Siebel Reference Architecture

The architecture on Oracle Private Cloud Appliance should be provisioned according to the table below:

SIEBEL COMPONENT	SHAPE	PLATFORM IMAGE	NODE COUNT	CPU CORES	RAM
Siebel Application Interface (SAI) – Siebel Management Console	VM.PCAStandard1.4	Oracle Linux 7.9	1	16	32 GB
Siebel Gateway	VM.PCAStandard1.4	Oracle Linux 7.9	3	16	32 GB
Siebel Server (SES)	VM.PCAStandard1.4	Oracle Linux 7.9	2	16	32 GB
Siebel Database	VM.PCAStandard1.4	Oracle Linux 7.9	1	16	32 GB

Note:

- The Ansible playbook used in this guide can be used to deploy additional nodes as needed by adding them into the playbook's hosts configuration file. Ideally, this is best before running the deployment at the environment preparation phase and considering that all nodes should have access to the shared filesystem.
- One single SAI node is defined by default in the playbook. More SAI nodes can be added if needed (e.g. to load balance (LB) via an external LB).

INSTALLATION PROCEDURE

For this exercise, the binary installer files are relied on to provide for on-premises installation of Oracle Siebel CRM 22.3 and Oracle Database. The installation process has been automated in an Ansible playbook.

The deployment process consists of 4 high-level stages:

1. Environment preparation
2. Oracle Database software installation
3. Oracle Siebel CRM 22.3 installation (deployment of all needed binaries in all nodes)
4. Siebel database configuration (data seeding)

The environment preparation is expected to be performed manually, or by external tooling like Terraform (not covered here). Stages 2, 3 and 4 are executed in an automated fashion via an Ansible playbook.

For more information on Oracle Siebel's installation process refer to the Installation Guide.

Assumptions

- The environment provisioned for this deployment will be exclusively used for this purpose and not shared with any other applications.
- Load Balancer configuration is not covered in this guide.

Prerequisites

- Access to an Oracle Private Cloud Appliance environment.
- Infrastructure to be provisioned as described in the reference architecture. If the required infrastructure is not already present, please contact your tenancy administrator.
- Use a DNS enabled VCN in all the compute instances.
- All the nodes in each layer of the infrastructure should be reachable from all nodes in that layer (e.g. Siebel Gateway hostnames should be resolvable from all Gateway nodes).
- A shared filesystem should be created and accessible from all the nodes in the infrastructure with at least 60 Gb. of disk space. For this exercise, OCFS2 was used on top of a block storage volume.

ENVIRONMENT PREPARATION

- Create the required infrastructure as outlined in the reference architecture. At the creation of compute nodes, make sure to use a DNS enabled VCN so hostnames are resolvable from nodes in the same layer.
- The Ansible playbook will be executed from the Siebel Application Interface node. This will be named the *Ansible's control node*.
- Get Oracle Siebel standard install zip files from My Oracle Support using your Oracle account:

```
p33947735_2200_Linux-x86-64_1of6.zip

p33947735_2200_Linux-x86-64_2of6.zip

p33947735_2200_Linux-x86-64_3of6.zip

p33947735_2200_Linux-x86-64_4of6.zip

p33947735_2200_Linux-x86-64_5of6.zip
```

- Get Oracle Database 19c software installer from here.
  - LINUX.X64_193000_db_home.zip
- Configure proxy and update all packages in each one of the hosts:

```
$ sudo sh -c "echo 'proxy=http://<your-proxy>:<your-proxy-port>' >> /etc/yum.conf"
$ sudo yum update -y
```

- Ensure that FQDNs for each compute host are resolvable as follows:

1. All hostnames should resolve from SAI (SMC and Ansible's control node)
2. Siebel Gateway hostnames should resolve from all Siebel Gateway nodes
3. Siebel Servers hostnames should resolve from all Siebel Server nodes
4. Oracle Database hostname should resolve from all nodes

- Mount the shared volume on a directory called /storage. For this exercise, an OCFS2 cluster was created. For more details on how to perform this step refer to the section "Shared filesystem cluster creation (OCFS2)"
- Create an 'oracle' user in each one of the hosts. Ensure that the user is added to sudoers file:

```
$ adduser oracle
$ passwd <pwd>
 #Give it sudo privileges:
$ sudo visudo
```

Add the following line to the bottom of the file for adding sudo power to members of the oracle group:

```
oracle        ALL=(ALL)     NOPASSWD: ALL
```

Make sure that NOPASSWD is included to avoid password-related issues while running the Ansible playbook later

- Generate and copy private ssh keys to the different compute instances as needed.
- Enable Passwordless authentication for the 'oracle' user
- Run:

```
$ sudo vi /etc/ssh/sshd_config
```

Add:

AllowUsers oracle

Set:

PasswordAuthentication yes

Run:

```
$ sudo systemctl restart sshd

 # Add your public key to authorized_keys:

$ cat .ssh/id_rsa.pub >> .ssh/authorized_keys

 # Restrict permissions to ssh keys

$ sudo chmod 700 .ssh/

$ sudo chmod 600 .ssh/authorized_keys
```

```
$ sudo chmod 400 /home/oracle/.ssh/id_rsa

$ sudo chmod 644 .ssh/id_rsa.pub
```

- Add private key into the ssh authentication agent (avoids asking for passphrase on ssh login):

```
$ eval $(ssh-agent)

$ ssh-add
```

- From the Ansible control node (where the playbook will be run), ensure that all relevant hostnames are added to known_hosts. This should happen after testing ssh connectivity.
- Ensure that the shared filesystem (e.g. OCFS2) is mounted on the **/storage** path in each one of the nodes.

```
$ df -Th /storage

Filesystem     Type   Size  Used Avail Use% Mounted on

/dev/sdb1      ocfs2  250G   46G  205G  19% /storage
```

- Create required directories:

```
$ mkdir -p /storage/installer/22.3.0.0/linux

$ mkdir -p /storage/installer/dbstage

$ mkdir -p /storage/installer/siebelcerts

$ mkdir -p /storage/siebelfs

$ sudo chmod -R 755 /storage
```

- Copy installer files to their respective locations:

```
$ scp LINUX.X64_193000_db_home.zip oracle@<instance_ip>:/storage/installer/dbstage

$ scp p33947735_2200_Linux-x86-64_*.zip
oracle@<instance_ip>:/storage/installers/22.3.0.0/linux
```

- Generate a Siebel JKS keystore and store it in /storage/installer/siebelcerts

  (this is a simple example of a self-signed certificate quick creation for demo purposes, but it is recommended that you use your company's JKS keystore and truststore).

Important: Take note of the password used in the keystore. It will be used later in the Ansible playbook configuration steps.

```
$ keytool -keystore siebelkeystore.jks -genkey -alias siebel

Enter keystore password:

Re-enter new password:

What is your first and last name?

  [Unknown]: <your-name>

What is the name of your organizational unit?

  [Unknown]:  <your-org-unit>

What is the name of your organization?

  [Unknown]:  <your-org-name>

What is the name of your City or Locality?

  [Unknown]:  <your-city>

What is the name of your State or Province?

  [Unknown]:  <your-state>

What is the two-letter country code for this unit?

  [Unknown]:  US

Is CN=Unknown, OU=my-custom-ca, O=my-custom-ca, L=my-city, ST=my-state, C=US correct?




Enter key password for <client>

    (RETURN if same as keystore password):



Warning:

The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS12 which is an
industry standard format using "keytool -importkeystore -srckeystore clientkeystore -
destkeystore clientkeystore -deststoretype pkcs12".



# -----------------------
```

```
# Check keystore contents:

$ keytool -list -v -keystore siebelkeystore.jks

Enter keystore password:

Keystore type: JKS

Keystore provider: SUN


Your keystore contains 1 entry


Alias name: client

Creation date: Oct 28, 2022

Entry type: PrivateKeyEntry

Certificate chain length: 1

Certificate[1]:

Owner: CN=my-name, OU=my-custom-ca, O=my-custom-ca, L=my-city, ST=my-state, C=US

Issuer: CN=my-name, OU=my-custom-ca, O=my-custom-ca, L=my-city, ST=my-state, C=US

...

...
```

Verify the keystore contents (make sure the format is JKS and not PKCS12):

```
$ keytool -list -v -keystore siebelkeystore.jks
```

- At the control node (the host from which you will execute the Ansible playbook) install Ansible:

```
$ sudo yum-config-manager --enable ol7_developer_EPEL
$ sudo yum repolist
$ sudo yum install ansible
```

EXECUTION PROCESS

The execution of the Ansible playbook referenced in this document consists of 4 main steps:

1. Playbook preparation
2. Oracle Database 19c software installation
3. Oracle Siebel CRM software installation
4. Siebel database seeding and configuration

Each of the steps will be run separately as indicated below. After these steps, users should be able to access Siebel's Management Console and configure the environment by creating Application Interface, Gateway and Enterprise Server profiles and deployments based on the reference architecture (while details on this are not covered in this document, they can be found in Oracle Siebel documentation).

The steps to execute the playbook are the following:

1. Playbook preparation

- Navigate to the Ansible playbook path.
- Edit <ansible-playbook-path>/inventories/siebel-inv/hosts and update with the host FQDNs corresponding to your PCA compute instances:
- E.g.

```
$ sed -i 's/<your\-siebel\-sai\-server1\.yourdomain\.com>/ses1\.my\.company.com/g' hosts
$ sed -i 's/<your\-siebel\-server1\.yourdomain\.com>/ses1\.my\.company.com/g' hosts
$ sed -i 's/<your\-siebel\-server2\.yourdomain\.com>/ses2\.my\.company.com/g' hosts
...
...
```

Note: Siebel MDE binaries will be deployed to the Oracle Database host too, since Siebel libraries are required for the database seeding (configuration) step.

- Edit the file *<ansible-playbook-path>/required_vars.yml* and change passwords or default usernames as needed.

  Notes:

  - If any of the required variables is not defined, the playbook will return an error and you will be required to define it in the file.
  - *PKI_PWD* variable should be the same password for the Siebel JKS keystore (see reference to JKS keystore creation above).
- Make sure that your ssh key is added to ssh's authentication agent before running the playbook steps below.

```
$ ssh-add
Enter passphrase for /home/oracle/.ssh/id_rsa:
Identity added: /home/oracle/.ssh/id_rsa (XXXXXXXX)
```

- Optionally, you may encrypt *required_vars.yaml* with Ansible-vault to keep usernames/passwords hidden. If you do this, you will need to run the Ansible playbook commands with the "--adk-vault-pass" flag.

```
$ ansible-vault encrypt required_vars.yaml [WARNING]: log file at /var/log/ansible/ansible.log
is not writeable and we cannot create it, aborting


New Vault password:
```

```
Confirm New Vault password:
Encryption successful ...

$ cat required_vars.yaml
$ANSIBLE_VAULT;1.1;AES256
393638363664363761303862663734356239646637653165633138623863323538653939373863 38
396238663530373036613163336537623566343531623763 0a6231666264643335356661366634 31
323838393663336362393736383339393261306564313766313766663939636635656533313064 38
336637643833656263 0a363062633865626165353636313938393137363363396165633934636635
...
...
```

2. Oracle Database 19c installation

Approximate execution time: 20 min

```
# Use --ask-vault-pass if required_vars was encrypted using Ansible-vault
$ ansible-playbook db19c_deploy.yml -i inventories/siebel-inv/hosts -u oracle <--ask-vault-pass>
```

3. Siebel CRM software installation

Approximate execution time: 10 min

```
# Use --ask-vault-pass if required_vars was encrypted using Ansible-vault
$ ansible-playbook siebel-install.yml -i inventories/siebel-inv/hosts -u oracle <--ask-vault-pass>
```

4. Oracle Database 19c configuration

Approximate execution time: 45 min

```
# Use --ask-vault-pass if required_vars was encrypted using Ansible-vault
$ ansible-playbook db19c_sblconfig.yml -i inventories/siebel-inv/hosts -u oracle <--ask-vault-pass>
```

Note: By default, Ansible will send the output of all these commands to STDOUT (console). You can change this to send output to a log file as specified in Ansible's logging documentation.

SETUP VALIDATION

Validate the deployment by following the next steps:

- Verify the output of each one of the stages
  - Each one of the steps above should generate a summary of the execution. Look out for any issues in the execution.

```
PLAY RECAP
*********************************************************************************************
```

```
**********************************************************************************************
************************

sbddb2.ext.psft.oraclevcn.com : ok=20   changed=8    unreachable=0    failed=0    skipped=12
rescued=0    ignored=0

sblnode1.ext.psft.oraclevcn.com : ok=39   changed=0    unreachable=0    failed=0    skipped=9
rescued=0    ignored=0


Monday 31 October 2022  09:13:43 -0500 (0:01:37.219)       0:45:54.273 ********

===============================================================================

db19c_sblconfig : Run Server upgrade wizard ---------------------------------------------
-----------------------------------------------------------------------------------------
-------------------------- 2620.05s

/home/oracle/workspace/pcaapps/pca-siebel/roles/db19c_sblconfig/tasks/main.yml:139 -----------
-----------------------------------------------------------------------------------------
---------------------------------

db19c_sblconfig : Validate Siebel schema ------------------------------------------------
-----------------------------------------------------------------------------------------
---------------------------- 97.22s

/home/oracle/workspace/pcaapps/pca-siebel/roles/db19c_sblconfig/tasks/main.yml:221 -----------
-----------------------------------------------------------------------------------------
---------------------------------

pause -----------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------
----------------------------- 3.59s

/home/oracle/workspace/pcaapps/pca-siebel/siebel-init-vars.yml:22 ---------------------------
-----------------------------------------------------------------------------------------
-------------------------

...

...

...

...
```

- Verify the outcome of the Oracle Database 19c configuration step

  If database configuration completed correctly, the last line in the last step's output should indicate the following:

```
dbchck: Checking completed: no checks failed.
```

- Verify access to Siebel's Management Console

  From a browser, navigate to: https://<your-siebel-sai-server.yourdomain.com>:9111/siebel/smc/index.html

  Login with sadmin/<Siebel_sadmin_password> credentials.

- Verify connectivity to Oracle Database

  From the Oracle Database host, ensure you can connect:

```
$ sqlplus siebel/<siebel_dbowner_pwd>@SIEBEL

SQL*Plus: Release 12.2.0.1.0 Production on Mon Oct 31 11:08:38 2022

Copyright (c) 1982, 2016, Oracle.  All rights reserved.

Last Successful login time: Mon Oct 31 2022 09:10:35 -05:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

SQL>
```

SHARED FILESYSTEM CLUSTER CREATION (OCFS2)

This section provides a brief explanation of the deployment of a OCFS2 shared filesystem cluster. This would be a requirement for mounting the /storage directory on a shared block volume and creating all the necessary directories for the Siebel deployment outlined above.

- Create a shared block volume in your PCA tenancy and compartment.
- Attach the volume to all the compute nodes in PCA (can be done through UI or via command line)
- In one of the compute instances, create a primary partition for the new device (e.g. /dev/sdb)

```
sudo fdisk /dev/sdb
```

```
# Set Linux Secure to permissive:
# Change to
#   SELINUX=permissive

$ sudo vi /etc/selinux/config
$ sudo setenforce permissive

# Open up ports in OS firewall:
# sudo firewall-cmd --zone=public --permanent --add-port=7777/tcp
# sudo firewall-cmd --zone=public --permanent --add-port=3260/tcp
# sudo firewall-cmd --complete-reload

# Install OCFS2 dependencies:
$ sudo yum install ocfs2-tools-devel ocfs2-tools -y

# Create a cluster definition
$ sudo o2cb add-cluster ociocfs2
```

```
# In each of your nodes:
$ sudo o2cb add-node ociocfs2 <sai-node-shortname> --ip <sai-node-ip>
$ sudo o2cb add-node ociocfs2 <ses-node1-shortname> --ip <ses-node1-ip>
$ sudo o2cb add-node ociocfs2 sbdsai1 --ip 10.0.1.25
$ sudo o2cb add-node ociocfs2 sbdsai2 --ip 10.0.1.26
$ sudo o2cb add-node ociocfs2 sbdgw1 --ip 10.0.1.27
$ sudo o2cb add-node ociocfs2 sbdgw2 --ip 10.0.1.28
$ sudo o2cb add-node ociocfs2 sbdgw3 --ip 10.0.1.29
$ sudo o2cb add-node ociocfs2 sbddb1 --ip 10.0.1.30

# Verify cluster definition:
$ sudo cat /etc/ocfs2/cluster.conf
cluster:
        heartbeat_mode = local
        node_count = 2
        name = ociocfs2

node:
        number = 0
        cluster = ociocfs2
        ip_port = 7777
        ip_address = <sai-node-ip>
        name = sai-node-shortname

node:
        number = 1
        cluster = ociocfs2
        ip_port = 7777
        ip_address = <ses-node1-shortname>
        name = ses-node-shortname
...
...

# Configure the cluster
# In each of the nodes run:
$ sudo /sbin/o2cb.init configure
$ sudo /sbin/o2cb.init status
```

```
$ sudo systemctl enable o2cb
$ sudo systemctl enable ocfs2

# Set panic reboot kernel parameters (required for OCFS2 cluster operation)
$ sudo sysctl kernel.panic=30
$ sudo sysctl kernel.panic_on_oops=1

# Add following lines to /etc/sysctl.conf:
-
    kernel.panic=30
    kernel.panic_on_oops=1
-

# Create an OCFS2 volume in the attached shareable device (primary partition)
# This step is only needed in any one of the cluster nodes
$ sudo mkfs.ocfs2 -L "storage" /dev/sdb1

# Create /storage directory and set permissions
# In each of the nodes
$ sudo mkdir /storage
$ sudo chown oracle:oracle /storage/
$ sudo chmod -R 755 /storage/

# Add the following line to /etc/fstab
-
  /dev/sdb1 /storage ocfs2     _netdev,defaults   0 0
-

# Run the following:
$ sudo mount -a

# Verify the cluster has been properly mounted
$ df -Th /storage
Filesystem     Type   Size  Used Avail Use% Mounted on
/dev/sdb1      ocfs2  250G   46G  205G  19% /storage

$ sudo o2cb.init status
Driver for "configfs": Loaded
Filesystem "configfs": Mounted
```

```
Stack glue driver: Loaded
Stack plugin "o2cb": Loaded
Driver for "ocfs2_dlmfs": Loaded
Filesystem "ocfs2_dlmfs": Mounted
Checking O2CB cluster "ociocfs2": Online
  Heartbeat dead threshold: 31
  Network idle timeout: 30000
  Network keepalive delay: 2000
  Network reconnect delay: 2000
  Heartbeat mode: Local
Checking O2CB heartbeat: Active
Debug file system at /sys/kernel/debug: mounted
```

RESOURCES

For additional information review

- Oracle Siebel Installation Guide
- Oracle Private Cloud Appliance Release Notes
- Oracle Systems Blog

CONNECT WITH US

Call +1.800.ORACLE1 or visit oracle.com.
Outside North America, find your local office at oracle.com/contact.

blogs.oracle.com facebook.com/oracle twitter.com/oracle

Oracle Siebel CRM 22.3 Installation
Jan-23
Author: Oracle Corporation