ORACLE

Oracle Secure Backup – Marketplace Image Deployment Guide

Quick guide to setting up Oracle Secure Backup in OCI from the Marketplace

Click or tap to enter a date, Version 2.0

Dropdown Options

Purpose statement

This document provides an overview on how to deploy Oracle Secure Backup from the OCI Marketplace

Disclaimer

Table of contents

Introduction

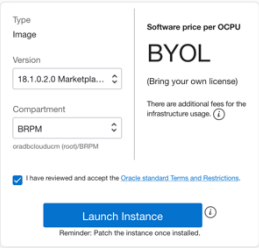
Oracle Secure Backup 18.1 is Oracle's media management solution, it provides centralized backup management for heterogeneous file systems and the Oracle Database to disk, tape, and cloud.

Oracle Secure backup is an enterprise class backup and recovery software that can help to protect your workloads on-premises, in Oracle Cloud Infrastructure (OCI) and in hybrid environments. OSB can backup compute nodes and databases and comes with built in features such as encryption, compression and many more. OSB can use OCI object storage buckets as backup target.

The OSB marketplace image allows you to quickly deploy and configure an OSB admin server in your OCI environment. The image comes bundled with Ansible playbooks that automate the process of deploying OSB media agent or client software to target compute nodes.

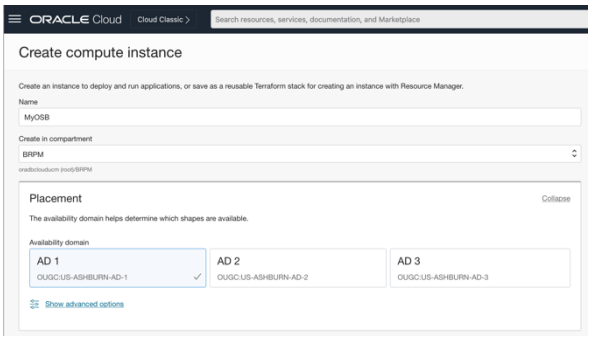
Oracle Secure Backup is a licensed product, this is a BYOL (Bring Your Own License) image, so you need to obtain the appropriate licenses. Contact your Oracle License Sales Representative for more information.

Deploying the Oracle Secure Backup Image



The actual image deployment process is simple and guided. Select the latest version and the compartment in which you want to deploy the Oracle Secure Backup Admin Serve and click "Launch Instance.

The rest of the process is identical to provisioning a new OCI Compute Instance. Assign a name, confirm the compartment, and select the Availability Domain of your choice.



Continue the guided process selecting the shape of the instance, and the networking parameters. For OSB to communicate properly with the client hosts, a set of ports (or all ports) must be opened. The "OCI Networking Configuration for Oracle Secure Backup" section below provides more information on OSB Networking requirements about VCN and local host firewall configurations.

Next add the ssh public key or create a new key pair. Again, this is a standard procedure when deploying new compute instances in OCI.

When the deployment completes, you are ready to log in to your OSB Admin Server

First Login

Connect to the new OSB Admin Server via ssh using the keys you supplied previously using the oci user.

Once you login, you are greeted with following message:

```
Welcome to Oracle Secure Backup marketplace deployment.
Press enter to continue setup. If you abort this, you
need to run the setup manually.
```

You can press any key to continue with the setup. If you abort the setup process, you can logout and re-login and setup will start again.

After continuing with the setup, OSB loads required binaries and creates some configuration files. The software is installed at the default location of "/usr/local/oracle/backup".

After the OSB software is loaded, you are presented with following prompt to specify the role for the current compute node.

```
Loading of Oracle Secure Backup software is complete.

Choose from one of the following options. The option you choose defines
the software components to be installed.
Configuration of this host is required after installation is complete.
You can install the software on this host in one of the following ways:

    (a) administrative server and client
    (b) client

If you are not sure which option to choose, please refer to the Oracle
Secure Backup Installation Guide. (a or b) [a]? :
```

As we are setting up an OSB admin server, specify "a" and press enter to continue with the setup.

OSB performs an initialization process and prompts you to specify an email for the "admin" user. OSB uses this email to send job summaries and other alerts.

```
You should now enter an email address for the Oracle Secure Backup 'admin'
user. Oracle Secure Backup uses this email address to send job summary
reports and to notify the user when a job requires input. If you leave this
blank, you can set it later using the obtool's 'chuser' command.

Please enter the admin email address: test@osbmail.com
```

The next prompt asks you to specify if want to change any advanced setting. For this setup, we do not need to modify any advanced configuration. Press “enter” to continue with setup.

```
Please enter the admin email address: test@osbmail.com

Do you want to change any advanced settings? (y or n) [n]:
```

The last two prompts ask you to specify the keystore and admin password.

The keystore password is used to encrypt the backup data encryption keys. It is important to specify a strong password for this. The admin password is used to login and manage the OSB domain. Refer to the OSB Administrator’s Guide to understand more about keystore and password management.

```
The keystore password must be known and safeguarded by
the Oracle Secure Backup Administrator.  In the event of
a disaster, the keystore password is required for
recovering your Oracle Secure Backup Administrative
Server.  Oracle Secure Backup cannot be prompted to
retrieve the password.

Please enter keystore password:
Re-type password for verification:
Please enter admin password:
Re-type password for verification:
Oracle Secure Backup was installed
[opc@instance-20230104-1139 ~]$
```

After you specify the passwords, the installation process completes and shows a message indicating Oracle Secure backup was successfully installed.

You can run the following command to verify that OSB services are up and running as desired.

```
[opc@instance-20230104-1139 ~]$ ps -eaf | grep obs
root       17404       1  3 17:28 pts/0    00:00:00 observiced -s
root       17408   17404  0 17:28 pts/0    00:00:00 /usr/local/oracle/backup/etc/obscheduled
opc        17582   16466  0 17:28 pts/0    00:00:00 grep --color=auto obs
[opc@instance-20230104-1139 ~]$
```

Now, you can login and use OSB commands to perform backup and other tasks.

Launch the “obtool” Command Line Interface to manage OSB.

The following screenshot shows the long listing of host.

```
[opc@instance-20230104-1139 ~]$ obtool -u admin
Password:
ob> lshost -l
instance-20230104-1139:
    Access mode:            OB
    IP names:               instance-20230104-1139.subnet09211721.vcn09211721.oraclevcn.com
    Disable RDS:            not set (system default)
    TCP/IP buffer size:     not set (global policy)
    S/w compression:        (not set)
    Algorithm:              aes192
    Encryption policy:      allowed
    Rekey frequency:        1 month
    Key type:               transparent
    In service:             yes
    Roles:                  admin,client
    Trusted host:           yes
    Certificate key size:   3072
    UUID:                   20f49ac8-6e83-103b-b444-0200170d9482
ob>
```

The marketplace OSB image comes bundled with “Ansible” playbooks and required dependent software to help you deploy OSB on client and media server. The Ansible playbooks are located in the `osb_playbooks` directory which contains a readme file with instructions on how to use them.

```
[opc@instance-20230104-1139 ~]$ ls -lR
.:
total 0
drwxrwxr-x. 2 opc opc 53 Dec  7 17:36 osb_home
drwxrwxr-x. 2 opc opc 66 Dec  7 17:51 osb_playbooks

./osb_home:
total 104796
-rw-r--r--. 1 opc opc 107308880 Dec  7 17:36 osb_18.1.0.2.0_linux.x64_mkt_120722.zip

./osb_playbooks:
total 20
-rw-r--r--. 1 opc opc  727 Dec  7 17:51 hosts
-rw-r--r--. 1 opc opc 6044 Oct 28 11:49 osb_install_linux.yml
-rw-r--r--. 1 opc opc 5192 Oct 28 11:50 Readme.txt
[opc@instance-20230104-1139 ~]$
```

At this point the Admin Server is ready!

Creating Cloud Storage Devices

Before you can create a target Cloud Storage Device you must have at least a Media Server in your Domain. You can add the Media Server Role to the Domain Admin Server by running the following command:

```
#obtool chhost --addrole mediaserver <hostname>
```

Oracle Secure Backup uses Object Storage to store your backups. Object Storage Buckets are configured in Oracle Secure Backup as Cloud Devices.

A Cloud Storage Device is an OSB object representing an OCI Object Storage target. Each Cloud Storage Device is associated with only one Bucket, and the Bucket must be reserved for OSB usage. Multiple Cloud Storage Devices can be created in OSB.

To create the first Cloud Storage Device perform the following steps:

- Identify the region, compartment, and the user account (with appropriate privileges) to be used to access your Object Storage Bucket(s).
- Create a new ssh key pair, or use an existing one
- Create the OSB Authentication Object
- Create the Cloud Storage Device(s)

You must choose the region your buckets will be located in and identify the compartment to use. It is recommended to create the buckets in the same region where your clients reside. If the environment is distributed in different regions, create at least a media server per region so the data do not travel across regions. You must make a note of the tenancy, compartment, and user OCIDs as you will need them later on. You also need to identify the storage namespace for your tenancy. You can find the storage namespace on the OCI console in your tenancy properties.

Creating the OSB Authentication Object

The first step in configuring OSB to access the OCI Object Storage Service is to create the OSB Authentication Object that will store the authentication information. To do this you can use the OSB Web interface or the obtool mkauth command.

This is the syntax for the mkauth obtool command

```
#obtool mkauth -t oci
[--comment/-c <comment>]
            {--fingerprint/-f  <key-finger-print>}
            [--iddomain/-d <identity-domain>]
            [--inputcomment/-i]
            {--keyfile/-k <key-file-path>}
            {--tenancyocid/-o <tenancy-ocid>}
            [--url/-r <cloud-url>]
            {--userocid/-u <user-ocid>}
            <authobj-name>
```

```
For example:
#obtool mkauth --type oci \
--fingerprint 3f:6e:a8:df:39:b6:5d:e0:51:fd:33:b6:54:b2:32:8d  \
--tenancyocid ocid1.tenancy.oc1..aaaaaaaaj4ccq454di45442s5x7ufvmmojd24smd4xwxyv3gda \
--keyfile /home/opc/.ssh/mykey.key   \
--url "objectstorage.us-ashburn-1.oraclecloud.com"    \
--userocid ocid1.user.oc1..aaaaaaaav3hetqe35pkl5ds4fuu2nmffgfdxpoxesdfxxpw2oj67xxx  \
--iddomain "mynamespace" \
myauthobj
```

You can create multiple authentication objects to use different accounts if needed. You can use the lsauth command to list the authentication objects in your OSB domain:

```
#obtool lsauth -long
myauthoci:
    Type:            oci
    Tenancy ocid:    ocid1.tenancy.oc1..xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
    User ocid:       ocid1.user.oc1..aaaaaaa.yyyyyyyyyyyyyyyyyyyyyyyyyyyy
    Key fingerprint: e3:2b:c1:22:cf:3c:32:ed:30:a2:35:26:d6:8a:f9:09
    Identity domain: mynamespace
    URL:             objectstorage.us-ashburn-1.oraclecloud.com
    UUID:            55699aaa-5fe2-1037-b5bb-fa163e0eda8f
```

Creating the Cloud Storage Device

Once the authentication object is created, we are ready to create the first cloud device. Before doing that, you must decide the storage tier you want to use. There are three tiers available in OCI Object Storage:

1. Standard: For frequently used data, or data that you want to be always immediately available for restore. This is the most expensive storage tier. There are no storage costs associated with downloading data from Standard tier buckets. The storage charges are only based on consumed capacity.

2. Infrequent Access: For infrequently accessed data, this is less expensive than the Standard tier and data is also immediately available for restore, but in addition to the consumed capacity charges there are also costs associated with the volume of data downloaded from the buckets. There is a minimum 31-day retention. If you delete objects before the 31 days has passed, you will still be charged for 31 days.

3. Archive: This is the least expensive tier and storage charges are capacity based only, but there is a delay of at least 1-hour before data are ready for download, so restores will be delayed. Also, there is a 90-day minimum retention in this tier.

Once you have identified all the information you need you can use the mkdev obtool command to create the Cloud Device in your OSB Domain

```
ob> obtool mkdev -t cloudstorage
--mediaserver <media server host name>
--storageclass <storage tier>
--inservice
--container <bucket name, the bucket will be created>
--concurrentjobs <number of concurrent jobs allowed to use this device>
--authobj <authentication object name>
--servicetype oci
--compartment <compartment OCID>
<Cloud Device Name>
```

For Example:

```
ob> obtool mkdev -t cloudstorage \
--mediaserver osb-ms-2 \
--storageclass infrequentaccess \
--inservice \
--container mynewbucket \
--concurrentjobs 10 \
--authobj myauthobj \
--servicetype oci \
--compartment ocid1.compartment.oc1..aaaaaaxslr4ksberu67gdfssd5iljdmydfolgfygwdpnrq \
myclouddevice
```

Once the Cloud Storage Device is created, you can list it using “lsdev” command as below

```
ob> lsd -l cloud
myclouddevice:
    Device type:          cloud storage
    Enable checksum:      (system default)
    In service:           yes
    Debug mode:            no
    Capacity:              (not set)
    Consumption:           0
    Free space goal:       (system default)
    Concurrent jobs:       1
    Blocking factor:       (default)
    Max blocking factor:   (default)
    UUID:                  5bbef4ae-5fe2-1037-b5bb-fa163e0eda8f
```

```
Attachment 1:
    Host:                osb-ms-2
Staging:                 no
Container:               mynewbucket
Storage class:           infrequentaccess
Identity domain:         mystoragenamespace
Segment size:            (system default)
Streams per job:         (system default)
Service type:             oci
Auth object:              myauthobject
```

At this point, device is ready to be used for the backup and restore.

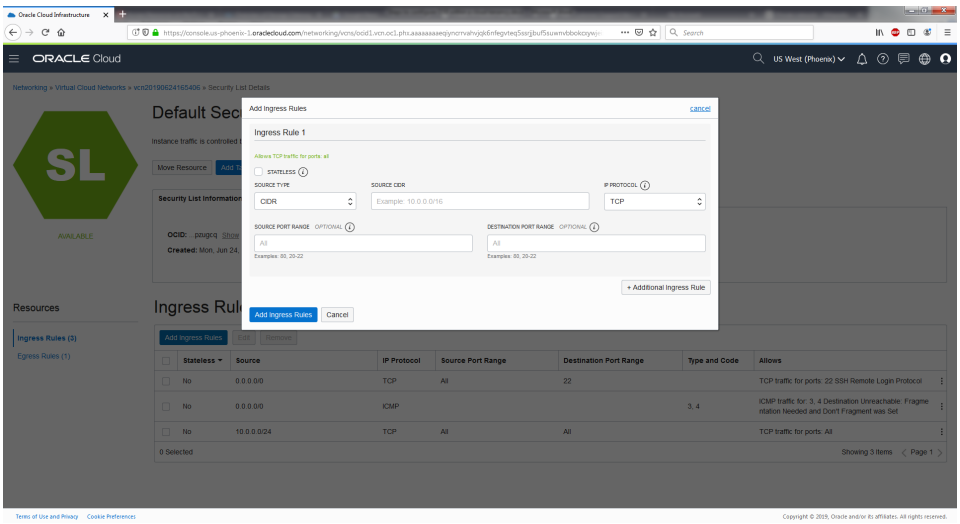
NOTE:	The bucket specified in the mkdev command must not exist, it will be created. The Concurrent Jobs value determines the number of OSB licenses required. The default value is 1.

OCI Networking Configuration for Oracle Secure Backup

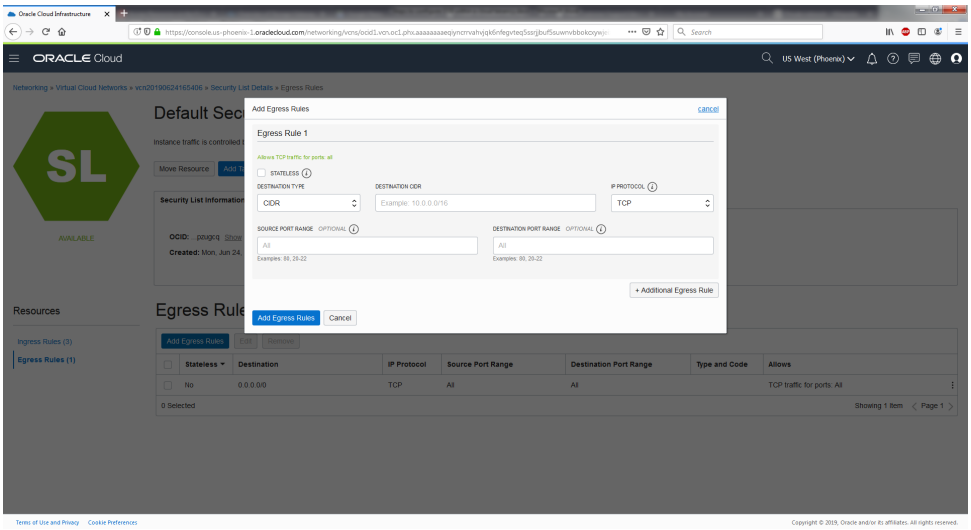
Updating VCNs (Virtual Cloud Networks) to open TCP ports for OSB traffic

Before deploying the OSB agents to the clients, you will need to update the VCN settings to allow TCP traffic between the VMs. You can choose to open all the ports for internal traffic or only select a limited set of ports as described below. The ports must be opened both at the VCN/Security Lists level and on the local firewall on each compute instance that you want to protect with OSB.

1. Select from main menu Networking -> Virtual Cloud Networks
2. Select the VCN for which you created instances
3. Select the appropriate subnet
4. Select “Default Security List”
5. Add ingress rule
    a. Source type “CIDR”
    b. Source CIDR “10.0.0.0/24” ← Specify the CIDR you are using
    c. IP Protocol “TCP”
    d. All ports



6. Add Egress Rule
    a. Source type “CIDR”
    b. Source CIDR “0.0.0.0/0”
    c. IP Protocol “TCP”
    d. All ports



You can choose to restrict traffic to limited ports. If you decide to limit the ports you open, please follow the following guidelines.

- Port 400 and port 10000 must be open for OSB control messages to flow through different hosts
- Whatever range you open for the data communication, please make sure you set the same range on all clients and media servers for open ports for application at OS level. The OSB Admin Server deployed via the Marketplace is configured to use the 50000-51000 port range.

Updating firewall settings for all instances

You will need to update your firewall settings for all instances to allow TCP traffic over the private network.

For each VM that you need to configure with OSB, please add the following firewall rules

a. iptables -I INPUT 1 -p tcp -s 10.0.0.0/24 -j ACCEPT
b. iptables -I OUTPUT 1 -p tcp -s 10.0.0.0/24 -j ACCEPT

This step enables the TCP communication over the private network within the tenancy on all the ports.

You must save these iptables changes to make sure they are persistent over reboot.

If you chose to only open certain ports, in addition to 400 and 10,000, that are mandatory, you need to adapt the iptables command to reflect that.

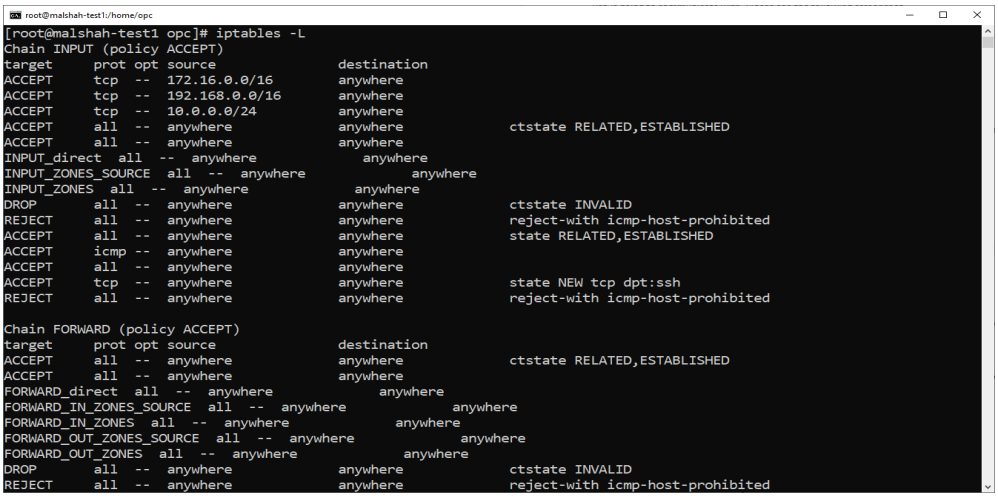
The Oracle Linux 8 image in OCI is now using firewalld. The following example shows a little script to open a specific set of ports 400, 10,000 and 50,000 to 51,000 on firewalld. This is what is used on the OSB Admin Server

```
#!/bin/bash
OB_LOW=50000
OB_HIGH=51000

# Open required ports
sudo firewall-cmd --add-port=400/tcp   --permanent
sudo firewall-cmd --add-port=10000/tcp --permanent
sudo firewall-cmd --add-port=${OB_LOW}-${OB_HIGH}/tcp --permanent
sudo firewall-cmd --reload

# Add OSB low and high ports to /etc/services
echo "ob-daemon-low   ${OB_LOW}/tcp      # OSB low port"   | sudo tee -a /etc/services
echo "ob-daemon-high  ${OB_LOW}/tcp      # OSB high port"  | sudo tee -a /etc/services
```

If you are peering multiple VCNs in your domain, please make sure to add rules for each CIDR range the VCN is going to communicate with. Please see the following screenshot.



Protecting Hosts across multiple compartments

OSB can supports hosts across multiple compartments within the same tenancy and same region. It works the same way as two VCNs that are connected through local peering within the same compartment (as described above). Users must follow the rules of local peering. Follow these steps to configure the environment

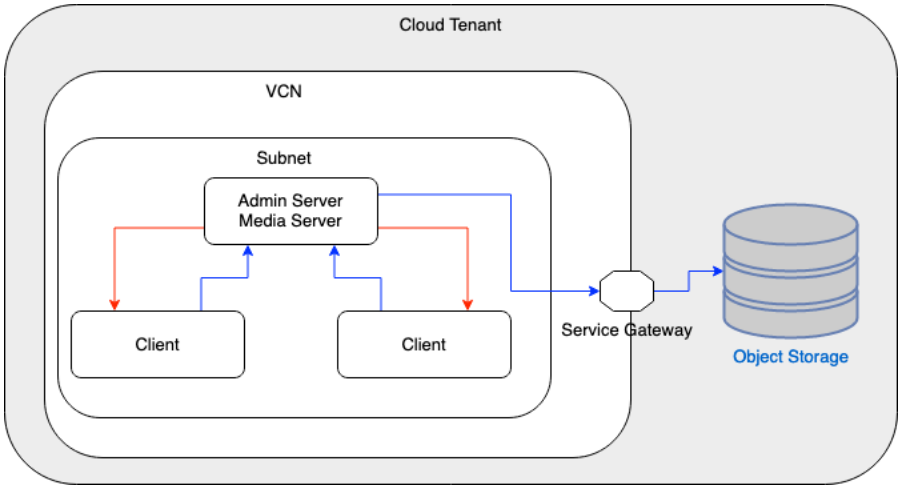
- Create VCNs/Subnets/Gateways in different compartments.
- Make sure the CIDR range does not overlap with the VCN you will be peering with.
- Configure local peering between the VCNs you want to make part of OSB domain.
- Configure firewall settings in all the instances that you would want to communicate with

Dataflow for the backup

In the above setup the data flow for the backup works as per the diagram below.

- Data between client and media server always traverse through the cloud private network
- Data between media server and object storage traverse through the VCN Service Gateway

The diagram shows a single VNC and Subnet. The different OSB components can span multiple VCNs and Subnets if the Security Lists and Routing Tables are configured to allow it.



Web tool

OSB web tool works within this setup with following settings.

- OSB Admin Server must have port 80 and port 443 open on the firewall settings.
- OCI VCN containing the OSB admin must open incoming traffic on TCP port 80 and 443
- The Web Browser client must be able to connect to the Admin Server on ports 80 and 443

Ansible Playbooks

These instructions assume SSH passwordless private/public key authentication mechanism setup between controller and other hosts.

Customizing the Ansible inventory file

The Ansible *hosts* inventory file located in /home/opc/osb_playbooks must be customized as shown below, before running the playbooks.

```
# OSB Ansible inventory template

# Variable Section

  [all:vars]

# Mandatory

  ob_adminhost=<your osb Admin Server hostname here> #OSB admin hostname

  ob_lin_temp=/home/opc/osb_home  #Local temporary path on the target host where the OSB shiphome
will be copied and extracted.

  ob_lin_shiphomepath=/home/opc/osb_home/osb_18.1.0.1.0_linux.x64_cdrom201123.zip #Path where OSB
shiphome stored on the controller node.

  ob_lin_shiphome=osb_18.1.0.1.0_linux.x64_cdrom201123 #OSB shiphome name

# Optional

  ob_lin_instpath=/usr/local/oracle/backup  #Path where OSB software will be installed

  up_var=18.1.0.2.0  #OSB upgrade version (future)


# Ansible related parameters(Mandatory).

  ansible_ssh_private_key_file=<full path of the ssh private key file> #Private key used for SSH
passwordless connection between admin server and hosts

# ansible_ssh_user=osbuser  #for connections using ssh authentication

# ansible_ssh_pass=osbpwd  #for connections using ssh authentication, recommend ansible vault for
better security

  [Linuxclient] # Enter the hostnames of the hosts where OSB client needs to be installed under this
section/group.

  <client1 hostname>

  <client2 hostname>

  <...>

  [Linuxmediaserver] # Enter the hosts where OSB mediaserver needs to be installed under this
section/group. For smaller environment a single combined admin/media server can be used, in that case
this list will be left empty

  <mediaserver1 hostname>

  <mediaserver2 hostname>

  <...>


  [Linuxclient:vars]

  ob_role=client  # osb host role


  [Linuxmediaserver:vars]

  ob_role=mediaserver # osb host role
```

Using the installation playbook

This playbook copies the agent software and installs it to clients and media servers and adds them to the OSB Domain

commands are executed from the /home/opc/osb_playbooks, if not, use the hosts file full pathname

To deploy the OSB software and configure the hosts belonging to Linuxclient group only use the –Limit parameter.

```
ansible-playbook osb_install_linux.yml  --limit "Linuxclient" -i hosts
```

If you want to deploy the OSB software to the Linuxmediaserver hosts and configure them as media servers in OSB

```
ansible-playbook  osb_install_linux.yml  --limit "Linuxmediaserver" -i hosts
```

A single command can be used for deploying both the clients and media servers at once

```
ansible-playbook  osb_install_linux.yml  --limit "Linuxclient,Linuxmediaserver" -i
hosts # OSB will be installed on hosts belonging to the Linuxclient and
Linuxmediaserver groups.
```

On successful completion, a summary of installed hosts will be displayed.

During installation, the failure log will be copied to <ob_lin_temp>/faillogs/hostname on the admin host, and the client host will be cleaned up.

Failed host details will be stored in the faillogs/Install_Failed_Hosts.log file.

Using the uninstall playbook

This playbook uninstalls the agent software from clients and media servers and removes them from the OSB Domain

Similarly to the installation example above, you can run the uninstallation playbook for clients only, mediaservers only or both

```
ansible-playbook osb_uninstall_linux.yml --limit "Linuxclient" -i hosts

ansible-playbook osb_uninstall_linux.yml --limit "Linuxmediaserver" -i hosts

ansible-playbook osb_uninstall_linux.yml --limit "Linuxclient,Linuxmediaserver" -i
hosts
```

On successful completion summary of uninstalled hosts will be displayed.

On failure host details will be stored in the faillogs/Uninstall_Failed_Hosts.log file.

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