OCICTL User Guide

1. Introduction

OCICTL is a tool simplifying using OCI Cli. It consists of the main bash script (ocictl.sh) and set of additional bash and python scriplets. The tool allows for

- using display names of services instead of their less descriptive and longer ids
- easy management of composite services, like, for example, Base
 Database Service, where we have multiple subcomonents, like databases, nodes, etc.
- easy management of group of services used together like, for example, application server working on top of database
- easy management of Object Storage buckets and their contents Note:

This tool supports currently only basic operations, like start/stop of a service or group of services belonging to the following categories

- Base Database Service
- Autonomous Database Service
- Compute instances

and basic operations on Object Storage

2. Prerequsities

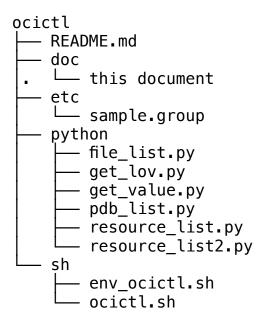
To use OCICTL the following software needs to be installed and configured

- Bash shell
- Python 3
- OCI Cli

OCICTL supports all main Linux/UNIX distributions, where all above requirements can be met, including MacOS and WSL configurations. It DOES NOT support native Windows command prompt.

3. Configuration

During cloning the OCICTL repository from github the following directory structure is created



sh directory contains two the most important scripts:

env_ocictl.sh	This script can be used to set required	
	environment values	
ocictl.sh	The main shell script	

To configure the tool there is need to set the following environment values:

OCI_NS	OCI namespace ID
OCI_TID	OCI tenancy ID
OCI_CID	OCI compartment ID
OCICTL_HOME	Directory, where OCICTL is installed
OCICTL_CONFIG	Directory, where different configuration files can be stored.
PATH	Should be extended by adding to it \$OCICTL_HOME/sh

All these environment variables can be set in env_ocictl.sh script, which is automatically called by ocictl.sh

4. Usage

Below table presents basic usage of ocictl.sh script:

ocictl.sh adb list	Lists all autonomous databases
	(*)
ocictl.sh adb start stop <service-name></service-name>	Starts/stops a particular
	autonomous database
ocictl.sh db list	Lists all base database services
	(*)
ocictl.sh db start stop <service-name></service-name>	Starts/stops a particular base
	database service
compute list	Lists all compute instances (*)
compute start stop <service-name></service-name>	Starts stops a particular
	compute instance
ocictl.sh os bucket list	Lists all bucket
ocictl.sh os bucket create <bucket-name></bucket-name>	Creates a bucket
ocictl.sh os bucket delete <bucket-name></bucket-name>	Deletes a bucket (**)
ocictl.sh os file list <bucket-name></bucket-name>	Lists content of a bucket
ocictl.sh os file put <bucket-name> <file-name></file-name></bucket-name>	Uploads a file into a bucket
ocictl.sh os file put <bucket-name> <file-name></file-name></bucket-name>	Downloads a file from a bucket

Managing groups of services

In some cases a particular solution consists of several separate services, like application servers, compute nodes and/or databases. In that case to simplify their startup/shutdown OCICTL introduces groups. A group can be defined as a named set of services: base database, autonomous database or compute. To create a group there is need to create a file with .group extension in \$OCICTL_CONFIG directory. Clear installation of the tool provides a sample group file, which can be used as a template:

compute-vm1 compute
compute-vm2 compute
db01 db
adb01 adb

In the first column we need to provide the name of a service (its display name), in the second – its type: **compute** for a compute instance, **db** for a base database service or **adb** for an autonomous database. To manage groups OCICTL has the following commands implemented.

ocictl.sh group list	Lists all existing groups with
	their contents
ocictl.sh group start stop <group></group>	Starts/stops a particular group