

ORACLE

Oracle Content Management

Upload Content from OCI Obj Storage to OCM,
using Oracle Content Toolkit Command-Line Utility

Lemnaru Cristina

Senior BPI Specialist- OCM

Oracle – Business Process Innovation
2023





Agenda

- Oracle Content Toolkit CLu
- Migration Procedure Overview
- Setting up the environment
- Useful materials
- Q&A



Oracle Content Toolkit Command-Line Utility

Run backend commands, develop site templates, themes, custom components, and content layouts for Oracle Content Management and work in your own development environment

```
cec <command> -h' to get the detailed help for the command.

ands:

ments
c create-folder <name>                Creates a folder or folder
erarchy on OCM server.                [alias: cfd]
c copy-folder <name>                  Copies folder on OCM
rver.                                [alias: cpfd]
c share-folder <name>                 Shares folder with users
d groups on OCM server.              [alias: sfd]
c unshare-folder <name>               Deletes user or group
cess to a shared folder on OCM server. [alias: usfd]
c list-folder <path>                  Displays folder hierarchy
OCM server.                          [alias: lfd]
c download-folder <path>              Downloads folder from OCM
rver.                                [alias: dlfd]
c upload-folder <path>                Uploads folder to OCM
rver.                                [alias: ulfd]
c delete-folder <path>                Deletes folder on OCM
rver.                                [alias:    ]
c copy-file <file>                    Copies file on OCM server.
lias: cpf]
c download-file <file>                 Downloads file <file> from
M server.                            [alias: dlf]
c upload-file <file>                  Uploads file <file> to OCM
rver.                                [alias: ulf]
c delete-file <file>                  Deletes file on OCM server.
lias:    ]
c describe-file <file>                Lists the properties of a
le on OCM server.                    [alias: dsf]

onents
c create-component <name>              Creates the component
ame>.                                [alias: cc]
c copy-component <source> [<destination>] Copies an existing local or
rver component.                      [alias: cpc]
c import-component <zip>               Imports a component from
ip>.                                [alias: ic]
c export-component <name>              Exports the component
ame> as a zip file.                  [alias: ec]
c download-component <names>           Downloads the components
ames> from the OCM server.           [alias: dlcp]
```


Oracle Content Toolkit Command-Line Utility

```
cec execute-post <endpoint>           Makes an HTTP POST request
to a REST API endpoint on OCM server [alias: exeo]
cec execute-put <endpoint>            Makes an HTTP PUT request
to a REST API endpoint on OCM server [alias: exeu]
cec execute-patch <endpoint>          Makes an HTTP PATCH request
to a REST API endpoint on OCM server [alias: exea]
cec execute-delete <endpoint>         Makes an HTTP DELETE
request to a REST API endpoint on OCM server [alias: exed]
cec install                           Creates source tree.
[alias: i]
cec develop                           Starts a test server.
[alias: d]
cec sync-server                       Starts a sync server.
[alias: scs]
cec webhook-server                   Starts a webhook server.
[alias: whs]

Options:
--version, -v  Show version number
--help, -h     Show Help

[root@linux /]# cec upload-folder -h
Usage: cec upload-folder <path>

Uploads folder and all its content to OCM server. Specify the server with -s <server> or use the one specified in cec.properties
<folder> to set the parent folder on OCM server.

Options:
--help, -h      Show Help
--folder, -f <folder> The parent folder on OCM server
--server, -s <server> The registered OCM server

Examples:
cec upload-folder ~/Downloads/docs           Uploads all content from ~/Downloads/docs to folder docs
cec upload-folder ~/Downloads/docs/          Uploads all content from ~/Downloads/docs to the Home folder
cec upload-folder ~/Downloads/docs -f Mydoc   Uploads all content from ~/Downloads/docs to folder Mydoc
cec upload-folder ~/Downloads/docs/ -f Mydoc  Uploads all content from ~/Downloads/docs to folder Mydoc
cec upload-folder ~/Downloads/docs -s SampleServer1 Uploads all content from ~/Downloads/docs to folder docs
SampleServer1
cec upload-folder ~/Downloads/docs/ -f site:blog1/settings/misc Uploads all content from ~/Downloads/docs to folder settings
cec upload-folder ~/Downloads/docs -f theme:blog1Theme          Uploads all content from ~/Downloads/docs to folder docs
cec upload-folder ~/Downloads/docs -f component:Comp1           Uploads all content from ~/Downloads/docs to folder docs

[root@linux /]#
```

Execute OCM Commands using **<cec>** prefix

Upload files, folders, change permissions, create sites, repositories, and many other options available in the OCM Web interface and more

Integrated help

The **"cec"** command-line utility has integrated help, which provides information on all available commands, with examples.

Work in your own dev environment

Content Toolkit has tools to create and develop custom components and site templates, including themes and content layouts. It includes a local test harness for quick, iterative development and sample unit tests to start with.

[Visit Documentation](#)

Migration Procedure Overview



Object Storage to OCM Migration procedure overview

Main steps



Migrating External Content

The actual files should be uploaded to a bucket in Oracle's Object Storage



Mount Object Storage Bucket

Making the files accessible for OCM Toolkit in the right format



Install and configure OCM Toolkit CLu

Installing OCM Toolkit to be able to run the migration scripts



Migrate documents to OCM

Perform the migration, execute <cec> commands to migrate content from mounted Obj Storage to final destination In OCM

To be taken in consideration



Migration scenario is designed for small volumes

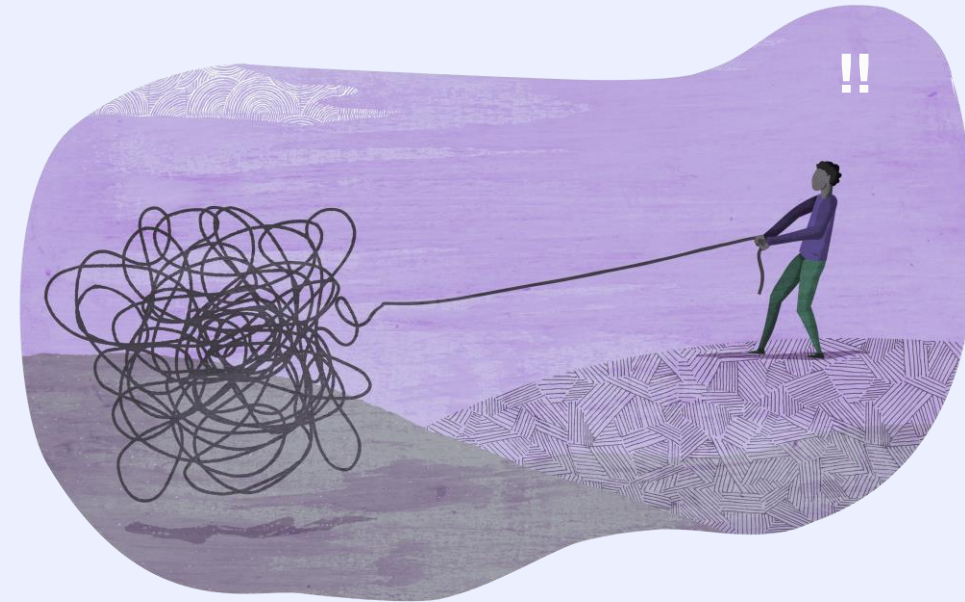


A good solution when OCM Secure Documents migration is required

It is a scenario designed when content should be migrated to Oracle Content Management Secure Documents – content organized in folders



Limited logging and monitoring



Setting up Migration



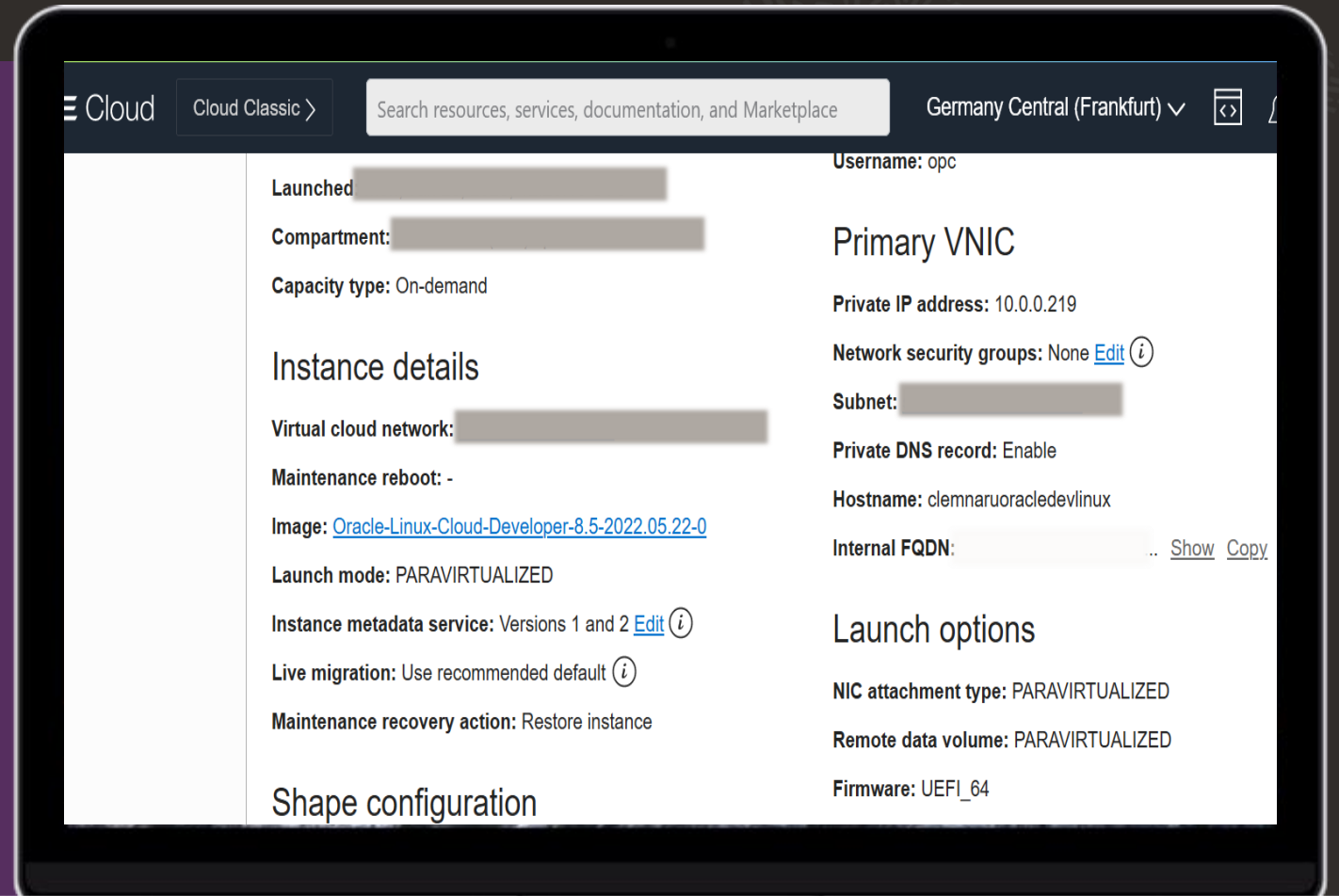


Prerequisites

Content Migrated to Object Storage Make sure that you have all the documents migrated to Object Storage

Keep everything in the same compartment. For a better performance, make sure that Object Storage, Mounted VM and OCM instance are all in the same compartment

Provision a compute instance . We need a Oracle Linux Cloud Developer instance where we will mount Obj Storage, it has a lot of tools already installed so it will simplify the process





Oracle Linux Cloud Developer instance provisioning

Compute

Overview

Instances

Dedicated Virtual Machine Hosts

Instance Configurations

Instance Pools

Cluster Networks

Scaling Configurations

Capacity Reservations

Custom Images

Instances in my-compartment Compartment

An [instance](#) is a compute host. Choose between virtual machines (VMs) and bare metal instances. The image that you use to launch an instance determines its operating system and other software.

Create instance

Table settings

Name	State	Public IP	Private IP	Shape	OCPU count	Memory (GB)	Availability domain	Fault domain	Compartment
my-instance-1	Running	130.61.130.232	10.0.0.220	VM.Standard2.2	1	16	AD-1	FD-2	my-compartment
my-instance-2	Running	130.61.46.94	10.0.0.169	VM.Standard2.2	1	16	AD-1	FD-1	my-compartment
my-instance-3 OracleDevLinux	Running	130.61.239.186	10.0.0.219	VM.Standard2.2	1	16	AD-1	FD-2	my-compartment

1 / 1

- From your OCI Menu navigate to Compute -> Instances -> Create instance



Oracle Linux Cloud Developer instance provisioning

Create compute instance

Image

Oracle Linux 8
Image build: 2022.12.15-0

Shape

VM.Standard.E4.Flex
Virtual machine, 1 core OCPU, 16 GB memory, 1 Gbps network bandwidth

Networking

[Networking](#) is how your instance connects to the internet and other resources in the Console. To

Primary network

☒ Select existing virtual cloud network ☐ Create new virtual cloud network ☐ Enter subnet

Virtual cloud network

[compartment](#)

Create Save as stack Cancel

Browse all images

Platform images

Compartment

oraemeaint (root)/specialists/clemnaru

[Platform images](#) are pre-built operating systems for Oracle Cloud Infrastructure.

Image name	OS version	Image build	Security	
<input type="checkbox"/> Canonical Ubuntu	22.04	2022.11.06-0		Advanced options
<input type="checkbox"/> CentOS	8 Stream	2022.10.26-0		Advanced options
<input type="checkbox"/> Oracle Autonomous Linux	7.9	2022.12.16-0		Advanced options
<input type="checkbox"/> Oracle Linux	9	2022.12.16-0		Advanced options
<input checked="" type="checkbox"/> Oracle Linux Cloud Developer	8	2022.05.22-0		Advanced options
<input type="checkbox"/> Windows	Server 2022 Standard	2022.09.06-0		Advanced options

1 Selected Showing 6 items

☐ I have reviewed and accept the following documents: [Oracle Linux Cloud Developer Image Terms of Use](#)

Select image Cancel

- Make sure you have selected an “Oracle Linux Cloud Developer” image and create it
- Wait till the instance will be provisioned and open it using the “opc ” user



Mount Object Storage

```
login as: opc
Authenticating with public key "imported-openssh-key"
Activate the web console with: systemctl enable --now cockpit.socket
```

- Now you are ready to Mount ObjectStorage Bucket as Filesystem in your Oracle Linux 8 VM
- Create a folder where you will keep the mounted Obj Storage and will create the OCM server

```
cec upload-folder ~/Downloads/docs -f component.v
[root@devlinux /]# mkdir mnt
```

```
[root@devlinux /]# ls
bin  dev  home  lib64  mnt  ocm-projects  proc  run  srv  tmp  var
boot  etc  lib  media  ocm  opt  root  sbin  sys  usr
```

- First of all, we will need to install **s3fs fuse On Oracle Linux Server**

[please use the following guide](#)

- Only after, we are ready to mount our Object Storage

[Mount ObjectStorage Bucket as filesystem](#)



Mount Object Storage

- After completing all the previous steps, your mounted ObjStorage will look as following

Objects		
<div>Upload More Actions</div>		
<input type="checkbox"/>	Name	
	> F1	
	> LargeFiles	
<input type="checkbox"/>	<input type="checkbox"/> Contract to sign - editable.docx	
<input type="checkbox"/>	<input type="checkbox"/> Contract to sign.docx	Wed, Jan 11, 2023, 09:19:56 UTC
<input type="checkbox"/>	<input type="checkbox"/> Contract to sign.pdf	Wed, Jan 11, 2023, 09:19:56 UTC
<input type="checkbox"/>	<input type="checkbox"/> David Won CV.docx	Wed, Jan 11, 2023, 09:19:56 UTC
<input type="checkbox"/>	<input type="checkbox"/> David Won ID.png	Wed, Jan 11, 2023, 09:19:56 UTC
<input type="checkbox"/>	<input type="checkbox"/> David Won Medical Exam.png	Wed, Jan 11, 2023, 09:19:56 UTC
<input type="checkbox"/>	<input type="checkbox"/> Hiring Contract David Won.pdf	Wed, Jan 11, 2023, 09:19:56 UTC
<input type="checkbox"/>	<input type="checkbox"/> Hiring Contract Sandra Phi.pdf	Wed, Jan 11, 2023, 09:19:56 UTC
<input type="checkbox"/>	<input type="checkbox"/> Medical Certificate 1.png	Wed, Jan 11, 2023, 09:19:56 UTC
<input type="checkbox"/>	<input type="checkbox"/> Medical Certificate 2.png	Wed, Jan 11, 2023, 09:19:56 UTC
<input type="checkbox"/>	<input type="checkbox"/> Medical Certificate 3.png	Wed, Jan 11, 2023, 09:19:56 UTC
<input type="checkbox"/>	<input type="checkbox"/> New Microsoft Word Document.docx	Wed, Jan 11, 2023, 09:19:56 UTC
<input type="checkbox"/>	<input type="checkbox"/> Sandra PHI CV.docx	Wed, Jan 11, 2023, 09:19:56 UTC
<input type="checkbox"/>	<input type="checkbox"/> Sandra PHI Medical Exam.png	Wed, Jan 11, 2023, 09:19:56 UTC

```
[root@clelnaruoracledvlinux mnt]# cd obj_bucket_ocm_migration/
[root@clelnaruoracledvlinux obj_bucket_ocm_migration]# ls
'Contract to sign.docx'      'David Won Medical Exam.png'  'Medical Certificate 1.png'    'organization policy.txt'
'Contract to sign - editable.docx'  F1  'Medical Certificate 2.png'    'Sandra PHI CV.docx'
'Contract to sign.pdf'      'Hiring Contract David Won.pdf'  'Medical Certificate 3.png'    'Sandra Phi ID.png'
'David Won CV.docx'        'Hiring Contract Sandra Phi.pdf'  'new employee checklist.docx'  'Sandra PHI Medical Exam.png'
'David Won ID.png'        LargeFiles  'New Microsoft Word Document.docx'

[root@clelnaruoracledvlinux obj_bucket_ocm_migration]# ls li
ls: cannot access 'li': No such file or directory
[root@clelnaruoracledvlinux obj_bucket_ocm_migration]# ls
'Contract to sign.docx'      'David Won Medical Exam.png'  'Medical Certificate 1.png'    'organization policy.txt'
'Contract to sign - editable.docx'  F1  'Medical Certificate 2.png'    'Sandra PHI CV.docx'
'Contract to sign.pdf'      'Hiring Contract David Won.pdf'  'Medical Certificate 3.png'    'Sandra Phi ID.png'
'David Won CV.docx'        'Hiring Contract Sandra Phi.pdf'  'new employee checklist.docx'  'Sandra PHI Medical Exam.png'
'David Won ID.png'        LargeFiles  'New Microsoft Word Document.docx'

[root@clelnaruoracledvlinux obj_bucket_ocm_migration]#
```



Oracle Content Management Toolkit Command Line Utility Installation

- Download the OCM Toolkit “sites” Package from [GitHub](#)
- In the root create a folder called “ocm”, then download and unzip the package there

```
[root@edevlinux /]# ls  
bin boot dev etc home lib lib64 media mnt ocm ocm-projects opt proc root run sbin srv sys tmp usr var
```

```
>>>> sudo wget https://github.com/oracle/content-and-experience-toolkit/archive/master.zip
```

```
>>>> unzip master.zip
```

- Navigate to folder >>>> `cd ocm/content-and-experience-toolkit-master/sites`

- Run here >>>> `npm install`

- Set the cec path variable in the `.bashrc` file (make sure you are in the root)

```
>>> Vi .bashrc
```

```
>>> Add the line with the following cec path:
```

```
>>> export PATH=/ocm/content-and-experience-toolkit-master/sites/node_modules/.bin:$PATH
```

```
>>> save the file
```




Oracle Content Management Toolkit Command Line Utility Instalation

- Now you are ready to test your OCM toolkit commands by just simply tapping “cec” you should be able to see all cec commands.

>>> cec

```
[root@ix-obj_bucket_ocm_migration]# cec
Usage: cec <command> [options]

Run cec <command> -h' to get the detailed help for the command.

Commands:

Documents
  cec create-folder <name>          Creates a folder or folder hierarchy on OCM server.      [alias: cfd]
  cec copy-folder <name>           Copies folder on OCM server.      [alias: cpfd]
  cec share-folder <name>          Shares folder with users and groups on OCM server. [alias: sfd]
  cec unshare-folder <name>        Deletes user or group access to a shared folder on OCM server. [alias: usfd]
  cec list-folder <path>           Displays folder hierarchy on OCM server.      [alias: lfd]
  cec download-folder <path>       Downloads folder from OCM server.      [alias: dlfd]
  cec upload-folder <path>         Uploads folder to OCM server.      [alias: ulfd]
  cec delete-folder <path>         Deletes folder on OCM server.      [alias: ]
  cec copy-file <file>             Copies file on OCM server.      [alias: cpf]
  cec download-file <file>         Downloads file <file> from OCM server. [alias: dlf]
  cec upload-file <file>           Uploads file <file> to OCM server. [alias: ulf]
  cec delete-file <file>          Deletes file on OCM server.      [alias: ]
  cec describe-file <file>        Lists the properties of a file on OCM server. [alias: dsf]

Components
  cec create-component <name>      Creates the component <name>.      [alias: cc]
  cec copy-component <source> [<destination>] Copies an existing local or server component. [alias: cpc]
  cec import-component <zip>       Imports a component from <zip>.      [alias: ic]
  cec export-component <name>      Exports the component <name> as a zip file. [alias: ec]
  cec download-component <names>   Downloads the components <names> from the OCM server. [alias: dlcp]
  cec upload-component <names>     Uploads the components <names> to the OCM server. [alias: ulcp]
  cec control-component <action>   Performs action <action> on components on OCM server. [alias: ctcp]
  cec share-component <name>       Shares component with users and groups on OCM server. [alias: sc]
  cec unshare-component <name>     Deletes user or group access to a component on OCM server. [alias: usc]
  cec describe-component <name>    Lists the properties of a component on OCM server. [alias: dscp]

Templates
  cec create-template <name>       Creates the template <name>.      [alias: ct]
  cec create-template-from-site <name> Creates the template <name> from site <site> on the OCM server. [alias: ctfs]
  cec download-template <name>     Downloads the template <name> from the OCM server. [alias: dlt]
  cec compile-template <source>    Compiles the site within the template. [alias: cmpt]
  cec copy-template <source> [<destination>] Copies an existing local or server template. [alias: cpt]
  cec import-template <zip>        Imports a template from <zip>.      [alias: it]
```





Oracle Content Management Toolkit Command Line Utility Installation

- Now we need to install the cec package in our **mnt** folder so just run

```
>>> cec install
```

- Once the installation is done, we need to register the OCM server, where we will going to migrate our files. Run the following command:

```
>>> cec register-server <myCustomServerName> -e <OCM URL> -u <OCM Username> -p <OCM Password>
```

- The next step is to set OAuth token , this is how you [can generate the token](#), (Access Using 2-Legged OAuth) after just run :

```
>>> cec set-oauth-token <my generated token > -s <myCustomServerName>
```



Migrate Content to OCM

- Now we are ready to migrate our files and folders from Mounted ObjStorage to OCM, let's see all the upload folder options :

>>> **cec upload-folder -h**

```
Options:
--help, -h      Show Help
--folder, -f    <folder> The parent folder on OCM server
--server, -s    <server> The registered OCM server

Examples:
cec upload-folder ~/Downloads/docs           Uploads all content from ~/Downloads/docs to folder docs on the server
cec upload-folder ~/Downloads/docs/          Uploads all content from ~/Downloads/docs to the Home folder on the server
cec upload-folder ~/Downloads/docs -f Mydoc   Uploads all content from ~/Downloads/docs to folder Mydoc/docs on the server
cec upload-folder ~/Downloads/docs/ -f Mydoc  Uploads all content from ~/Downloads/docs to folder Mydoc on the server
cec upload-folder ~/Downloads/docs -s SampleServer1 Uploads all content from ~/Downloads/docs to folder docs on the registered server
                                                SampleServer1
cec upload-folder ~/Downloads/docs/ -f site:blog1/settings/misc Uploads all content from ~/Downloads/docs to folder settings/misc of site blog1
cec upload-folder ~/Downloads/docs -f theme:blog1Theme Uploads all content from ~/Downloads/docs to folder docs of theme blog1Theme
cec upload-folder ~/Downloads/docs -f component:Comp1 Uploads all content from ~/Downloads/docs to folder docs of component Comp1

[root@edevlinux obj_bucket_ocm_migration]#
```

- Run the migration command by just mentioning the source and destination folders

>>> **cec upload-folder <my mounted/ObjStorage/ bucket/ location/> -f <My OCM Folder> -s <myCustomServerName>**

Migrate Content to OCM

- The result

Objects		
<div>Upload More Actions</div>		
<input type="checkbox"/>	Name	Last Modified
	> F1	-
	> LargeFiles	-
<input type="checkbox"/>	Contract to sign - editable.docx	Wed, Jan 11, 2023, 09:19:56 UTC
<input type="checkbox"/>	Contract to sign.docx	Wed, Jan 11, 2023, 09:19:56 UTC
<input type="checkbox"/>	Contract to sign.pdf	Wed, Jan 11, 2023, 09:19:56 UTC
<input type="checkbox"/>	David Won CV.docx	Wed, Jan 11, 2023, 09:19:56 UTC
<input type="checkbox"/>	David Won ID.png	Wed, Jan 11, 2023, 09:19:56 UTC
<input type="checkbox"/>	David Won Medical Exam.png	Wed, Jan 11, 2023, 09:19:56 UTC
<input type="checkbox"/>	Hiring Contract David Won.pdf	Wed, Jan 11, 2023, 09:19:56 UTC
<input type="checkbox"/>	Hiring Contract Sandra Phi.pdf	Wed, Jan 11, 2023, 09:19:56 UTC
<input type="checkbox"/>	Medical Certificate 1.png	Wed, Jan 11, 2023, 09:19:56 UTC
<input type="checkbox"/>	Medical Certificate 2.png	Wed, Jan 11, 2023, 09:19:56 UTC
<input type="checkbox"/>	Medical Certificate 3.png	Wed, Jan 11, 2023, 09:19:56 UTC
<input type="checkbox"/>	New Microsoft Word Document.docx	Wed, Jan 11, 2023, 09:19:56 UTC
<input type="checkbox"/>	Sandra PHI CV.docx	Wed, Jan 11, 2023, 09:19:56 UTC
<input type="checkbox"/>	Sandra PHI Medical Exam.png	Wed, Jan 11, 2023, 09:19:56 UTC

Documents > my_migrated_content > obj_bucket_ocm_migration				
<input type="checkbox"/>	Select All			
	Name ↑	Version	Last Updated ↕	Updated
<input type="checkbox"/>	F1		Wednesday at 3:42 PM	You
<input type="checkbox"/>	LargeFiles		Wednesday at 3:42 PM	You
<input type="checkbox"/>	Contract to sign - editable.docx	v1	Wednesday at 3:42 PM	You
<input type="checkbox"/>	Contract to sign.docx	v1	Wednesday at 3:42 PM	You
<input type="checkbox"/>	Contract to sign.pdf	v1	Wednesday at 3:42 PM	You
<input type="checkbox"/>	David Won CV.docx	v1	Wednesday at 3:42 PM	You
<input type="checkbox"/>	David Won ID.png	v1	Wednesday at 3:42 PM	You
<input type="checkbox"/>	David Won Medical Exam.png	v1	Wednesday at 3:42 PM	You
<input type="checkbox"/>	Hiring Contract David Won.pdf	v1	Wednesday at 3:42 PM	You
<input type="checkbox"/>	Hiring Contract Sandra Phi.pdf	v1	Wednesday at 3:42 PM	You
<input type="checkbox"/>	Medical Certificate 1.png	v1	Wednesday at 3:42 PM	You



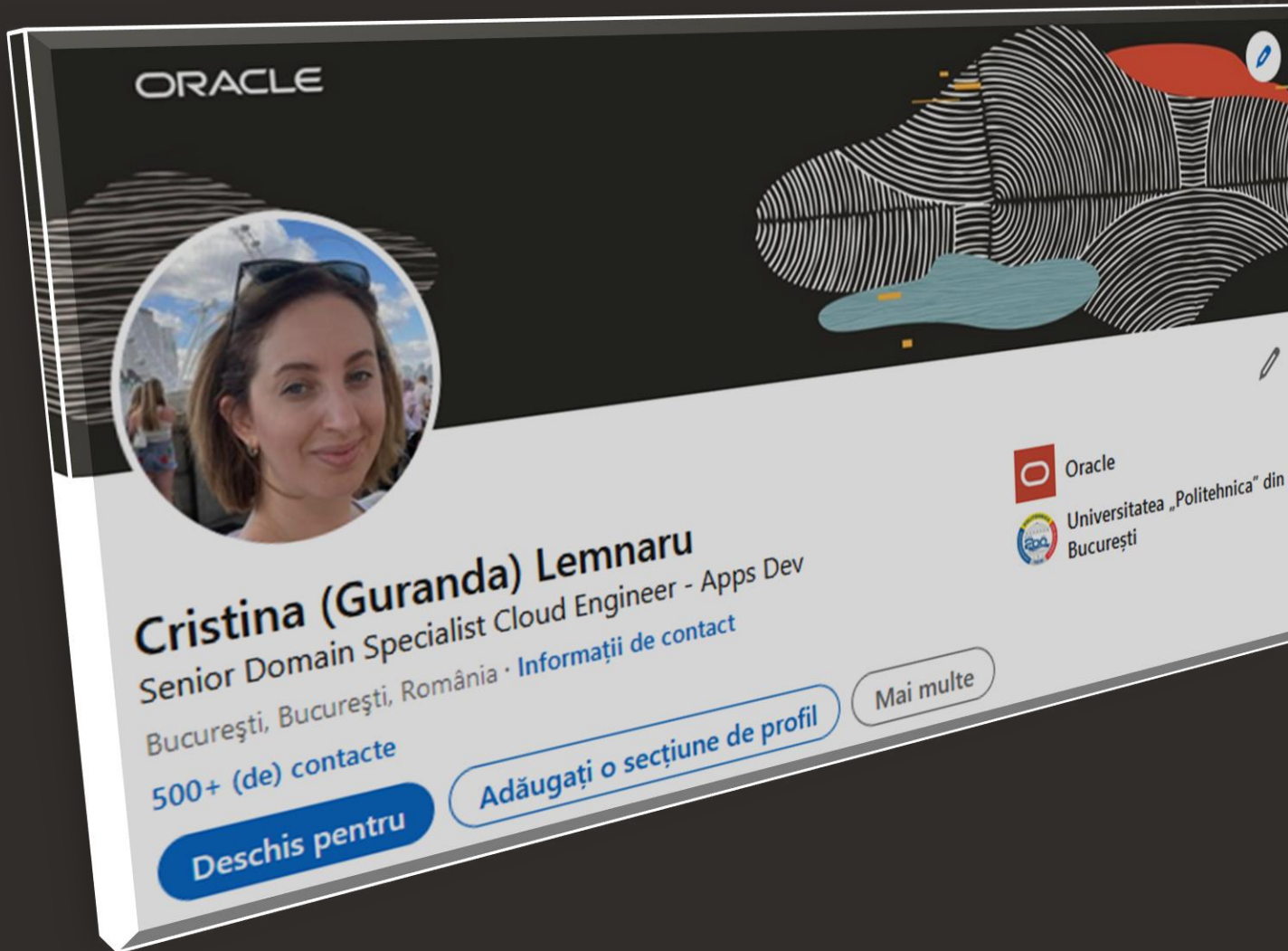
Useful links



Useful Links

- [Oracle Content Management Toolkit](#)
- [s3fs- fuse Github](#)
- [Access Using 2-Legged OAuth \(Identity Domain\)](#)

Happy to connect



Thank You!