CD3 Automation Toolkit - End to End Process

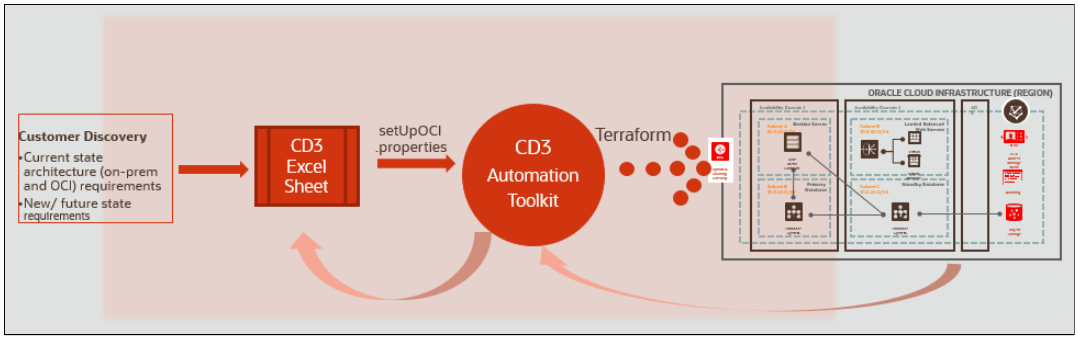
Created by Shruthi Subramanian, last modified by Dipesh Rathod 41 minutes ago

NOTE: The container also includes a copy of the confluence pages at */cd3user/tenancies/customer_name>/documentation/user_guide*

Introduction

CD3 Stands for **C**loud **D**eployment **D**esign **D**eliverable and is a structured design-level representation of the customer's OCI future state solution. The CD3 Automation toolkit is a processor that converts the detailed OCI design spec in the form of excel sheet into executable Terraform code, or takes an export of customer tenancy objects and resources, and converts it back into a design spec.

The generated TF files can be re-used at any time to build a similar infrastructure.



Automation Toolkit for OCI

Below is the list of OCI services that the CD3 Automation Toolkit can be used for the generation of Terraform files.

Identity	Compartments, Groups, Dynamic Groups, Policies
Network	VCNs, Subnets, Route Tables, Security Lists, NSGs, NAT gateways, IGWs, Service Gateways, LPGs, DRG, DRG Distributions, DRG Route Tables, LBaaS, Network Load Balancers
Compute	Instances VM/BM
Storage	Boot Volumes, Block Volumes, FSS
Governance	Tag Namespaces, Tag Keys, Default Tags, Cost Tracking Tags, Defined Tags
Developer Services	Resource Manager, OKE
Databases	ADW/ATP, Exa-Infra, Exa - Clusters, DB VM/BM
Management Services	Events, Alarms, Notifications, Logging, Service Connector Hub
CIS Features	CIS Report, OSS, KMS - Key, Vault, VCN Flow Logs and Object Storage Write Logs, Cloud Guard, Budget

Automation Toolkit Workflows

CD3 Automation Tool Kit supports 2 main workflows:

1. Greenfield Tenancies. - Empty OCI tenancy (or) do not need to modify / use any existing resources.
2. Non-Greenfield Tenancies - Need to use / manage existing resources. Export existing resources into CD3 & TF State, then use the Greenfield workflow.

Internal to Oracle
To obtain access and keep abreast of the latest releases - please join the #oci-cd3-champions channel.

Check out the videos at https://otube.oracle.com/channel/CD3%2BAutomation%2BToolkit/252278113 to learn and familiarise the toolkit before getting started.

Recommendations

- Use the **Validate** option in **SetUpOCI menu** to validate the syntax/typos in your input CD3 Excel sheet.
- For the Non-Greenfield Tenancies, please use a clean out directory (Make sure to not have any **.auto.tfvars** or **terraform.tfstate** in the outdir) and a blank CD3 file - **CD3-Blank-template**.xlsx.
- Prepping the out directory to support a newly subscribed region at a later point in time involves -

  - **Taking** a **backup** of the **existing out directory**

  - **Copying** all the terraform **modules** and **.tf** files, **except** the **.auto.tfvars** and .**tfstate** files from existing region

  - **Modifying** the **name** of **variables file** (variables_<region>.tf)

  - **Modifying** the **region parameter** in **variables_<region>.tf**

- Preparing the out directory to support a new docker image release or update involves -

  - **Taking** a **backup** of the **existing out directory (Optional)**

  - **Copying** all the terraform **modules** and **.tf** files (Except **variables_example.tf**) from **/cd3user/oci_tools/cd3_automation_toolkit/user-scripts/terraform/** to region specific directories in out directory.

Example:

```
cd /cd3user/oci_tools/cd3_automation_toolkit/user-scripts/terraform/
cp -R modules /cd3user/tenancies/<customer_name>/terraform_files/<region>/
cp *.tf /cd3user/tenancies/<customer_name>/terraform_files/<region>/
cd /cd3user/tenancies/<customer_name>/terraform_files/<region>/
rm -rf variables_example.tf
```

Pre-requisites

Requirements	For Oracle Employees/Internal Users **Do NOT be connected to Oracle VPN while performing the steps.**	For External Users
Tenancy Access	Appropriate IAM policies must be in place for each of the resources that the user may try to create. Minimum requirement for the user to get started is to have the ability to read to the tenancy.	Appropriate IAM policies must be in place for each of the resources that the user may try to create. Minimum requirement for the user to get started is to have the ability to read to the tenancy.
Input	CD3 Excel Sheet (Sample CD3 templates here: **Example CD3 Excel Templates - CIS Landing Zone**)	CD3 Excel Sheet (Sample CD3 templates in Git Repo at **cd3_automation_toolkit/example/)**
Out Directory	A directory that will be shared with the docker container that will hold the generated Terraform files.	A directory that will be shared with the docker container that will hold the generated Terraform files.
Container Image	Join the Slack Channel #oci-cd3-champions for the Docker Image URL & information on the latest releases. On receiving the Object Storage URL for the Container Image, download and move the *cd3toolkit_<image_tag>.tar.gz* to the system where the Rancher Desktop/podman is installed.	Generate the container image by following the steps in Readme.md (Git Repo)
Platform	Rancher Desktop/Podman and the steps can be found at - Install Rancher or Podman.	Any docker cli compatible platform to run the toolkit.

Deploy

Create a directory in your local machine. This will be the output directory while running the toolkit to store the generated Terraform/other files.

Place any files or folder in this path to make it available inside the container. This directory will be mapped to a docker container directory and the path to this directory is identified as **<path_in_local_system_where_the_files_must_be_generated>** in the next steps.

This will allow you to upgrade the docker container in the future without losing your data.

Steps	Rancher Desktop (with Docker CLI)	Podman
Load the docker image	`sudo docker load < cd3toolkit_<image_tag>.tar.gz` Example: (Example for Linux: *sudo docker load < cd3toolkit_v5.0.tar.gz*) (Example for Windows: *docker load -i cd3toolkit_v5.0.tar.gz*)	`sudo podman load < cd3toolkit_<image_tag>.` Example: (Example for Linux: *sudo podman load < cd3toolkit_* (Example for Windows: *podman load -i cd3toolkit_v*
List the docker	`sudo docker images`	`sudo podman images`

Steps	Rancher Desktop (with Docker CLI)	Podman
images		
Create the container	`sudo docker run -it -d -v \ <path_in_local_system_where_the_files_must_be_generated>:/cd3user/tenancies \ <image_name>:<image_tag> /bin/bash` The arguments *-v <path_in_local_system_where_the_files_must_be_generated>* in the above command is used to map the local directory to a docker container directory */cd3user/tenancies*). (Example for Linux: *sudo docker run -it -d -v /home/opc/tenancies/:/cd3user/tenancies cd3toolkit:v5.0 /bin/bash*) (Example for Windows: *docker run -it -d -v D:\tenancies:/cd3user/tenancies cd3toolkit:v5.0*)	`sudo podman run -it -d -v \ <path_in_local_server_where_the_files_must <image_name>:<image_tag> /bin/bash` The arguments *-v <path_in_local_server_where_th* command is used to map the local directory to a doc (Example for Linux: *sudo podman run -it -d -v /hom cd3toolkit:v5.0 /bin/bash*) (Example for Windows: *podman run -it -d -v D:\tena* (Example for MacOS: *podman run -it -d -v <podman /bin/bash* -Replace *podman_vm_path* with the one t
List the running containers and note the container id	`sudo docker ps`	`sudo podman ps`
Exec into the docker container	`sudo docker exec -it <container_id> bash`	`sudo podman exec -it <container_id> bash`

Configuring the Docker Container to connect to OCI Tenancy

Follow the below steps to configure the docker container to connect to a tenancy:

⚠ Repeat this process for every new customer. Same docker container can be connected to multiple OCI tenancies.

Commands to execute inside the container:

Steps	Command
Change Directory to that of user-scripts	`cd /cd3user/oci_tools/cd3_automation_toolkit/user-scripts/`

API PEM keys:
If the key pair does not exist, create them using the script.
	`python /cd3user/oci_tools/cd3_automation_toolkit/user-scripts/createAPIKey.py`
	In case you already have the keys, rename the private key file to ***oci_api_private.pem*** and place it at **/cd3user/tenancies/keys** for smooth functio
	ERROR: While executing the above steps on a Linux VM in OCI -
	If the outdir is on the root, you may get a permission denied error.
	In such scenarios, please follow the steps given below -
	› Error: Permission Denied
	• **Error Screenshot**
	`[opc@shrsubra-linux ~]$ sudo docker run -it -d -v /var/tmp/docker-container/:/cd3user/tenancies/ localhost/cd3toolkit:v5.0.2 /sbin/init`
	`Emulate Docker CLI using podman. Create /etc/containers/nodocker to quiet msg.`
	`3ea722fa13ba2136481953fa59fe2de096968622054b54ff0d5d9dd980743353`
	`[opc@shrsubra-linux ~]$ cd /var/tmp/docker-container`
	`[opc@shrsubra-linux docker-container]$ ls -ltra`
	`total 0`
	`drwxrwxrwx. 2 opc opc 6 May 4 15:15 .`
	`drwxrwxrwt. 5 root root 232 May 4 15:24 ..`
	`[opc@shrsubra-linux docker-container]$ sudo docker exec -it 3ea bash`
	`Emulate Docker CLI using podman. Create /etc/containers/nodocker to quiet msg.`
	`[cd3user@3ea722fa13ba ~]$ cd tenancies/`
	`[cd3user@3ea722fa13ba tenancies]$ ls`
	`ls: cannot open directory .: Permission denied`
	• **Solution**
	- In such scenarios, please change
	1. **selinux mode** from **Enforcing** to **Permissive**
	2. **change the owner of folders** in **/cd3user/tenancies** to that of **cd3user.**
	Please refer the screenshots below -

```
[opc@shrsubra-linux ~]$ mkdir /outdir
[opc@shrsubra-linux ~]$ sudo podman ps
CONTAINER ID  IMAGE       COMMAND     CREATED     STATUS      PORTS       NAMES
[opc@shrsubra-linux ~]$ sudo podman images
REPOSITORY            TAG         IMAGE ID      CREATED     SIZE
localhost/cd3toolkit  v5.0.2      7080bf808554  5 days ago  3.24 GB
[opc@shrsubra-linux ~]$ sudo podman run -it -d -v /outdir/:/cd3user/tenancies localhost/cd3toolkit:v5.0.2 /sbin/init
3b472c910b3833d31753b91f25e1ff38641481faafff9f0110ef0612635e8074
[opc@shrsubra-linux ~]$ sudo podman exec -it 3b47 bash
[cd3user@3b472c910b38 ~]$ cd tenancies/
[cd3user@3b472c910b38 tenancies]$ mkdir keys
mkdir: cannot create directory 'keys': Permission denied
[cd3user@3b472c910b38 tenancies]$ ls -ltra
total 0
drwxrwxr-x. 2 opc     opc      6 May  9 09:34 .
drwxrwx---. 1 cd3user cd3user 23 May  9 09:34 ..
[cd3user@3b472c910b38 tenancies]$ exit
exit
```

```
[opc@shrsubra-linux ~]$ getenforce
Enforcing
[opc@shrsubra-linux ~]$ sudo setenforce 0
[opc@shrsubra-linux ~]$ getenforc
-bash: getenforc: command not found
[opc@shrsubra-linux ~]$ getenforce
Permissive
[opc@shrsubra-linux ~]$ sudo podman exec -it 3b47 bash
[cd3user@3b472c910b38 ~]$ cd tenancies/
[cd3user@3b472c910b38 tenancies]$ ls -ltra
total 0
drwxrwxr-x. 2 opc     opc      6 May  9 09:34 .
drwxrwx---. 1 cd3user cd3user 44 May  9 09:35 ..
[cd3user@3b472c910b38 tenancies]$ mkdir keys
mkdir: cannot create directory 'keys': Permission denied
[cd3user@3b472c910b38 tenancies]$ sudo mkdir keys
sudo: unable to send audit message: Operation not permitted
[cd3user@3b472c910b38 tenancies]$ ls -ltra
total 0
drwxrwx---. 1 cd3user cd3user 44 May  9 09:35 ..
drwxr-xr-x. 2 root    root     6 May  9 09:35 keys
drwxrwxr-x. 3 opc     opc     18 May  9 09:35 .
[cd3user@3b472c910b38 tenancies]$ sudo chown -R cd3user:cd3user .
sudo: unable to send audit message: Operation not permitted
[cd3user@3b472c910b38 tenancies]$ ls -ltra
total 0
drwxrwx---. 1 cd3user cd3user 44 May  9 09:35 ..
drwxr-xr-x. 2 cd3user cd3user  6 May  9 09:35 keys
drwxrwxr-x. 3 cd3user cd3user 18 May  9 09:35 .
[cd3user@3b472c910b38 tenancies]$ mkdir test-folder
[cd3user@3b472c910b38 tenancies]$ ls -ltra
total 0
drwxrwx---. 1 cd3user cd3user 44 May  9 09:35 ..
drwxr-xr-x. 2 cd3user cd3user  6 May  9 09:35 keys
drwxrwxr-x. 2 cd3user cd3user  6 May  9 09:36 test-folder
drwxrwxr-x. 4 cd3user cd3user 37 May  9 09:36 .
[cd3user@3b472c910b38 tenancies]$
```

- Alternately, please attach a **data disk** and map the container (/cd3user/tenancies) on a folder that is created on the data disk (your local folde

Upload the Public key to "**API keys"** under user settings in OCI Console.

Pre-requisite to use the complete functionality of the Automation Toolkit is to have the user as an administrator to the tenancy.

Steps:

1. Open the Console, and sign in as the user:
   View the details for the user who will be calling the API with the key pair:
2. Open the Profile menu (User menu icon) and click User Settings.
3. Click Add Public Key.
4. Paste the contents of the PEM public key in the dialog box and click Add.

Enter the details to ***tenancyconfig.properties*** file

tenancyconfig.properties

```
[Default]
#Mandatory Fields
#Friendly name for the Customer Tenancy eg: gctenancy;
#The generated .auto.tfvars will be prefixed with this customer name
customer_name=
tenancy_ocid=
fingerprint=
user_ocid=
#Path of API Private Key (PEM Key) File; If the PEM keys were generated by running createAPI.py,leave th
#Defaults to /cd3user/tenancies/keys/oci_api_private.pem when left empty.
key_path=
#Region ; defaults to us-ashburn-1 when left empty.
region=

#Optional Fields
#SSH Key to launched instances
ssh_public_key=
```

```
python /cd3user/oci_tools/cd3_automation_toolkit/user-scripts/createTenancyConfig.py tenancyconfig.properties
```

Initialise your environment to use the Automation Toolkit	*If the API Keys were generated and added to the OCI console using the previous steps, it might take a couple of seconds to reflect. Thus, running t immediately might result in Authentication Errors.* *In such cases, please retry after a minute.*

Files created on successfully completing the above steps -

1. A customer specific Config file
2. Customer specific setUpOCI.properties file
3. Region based directories along with Variables File, Provider File, Root and Sub modules
4. Public and Private Key Pairs
5. cmds.log file that contains a copy of the *Commands to execute* section of the *console output*
6. A documentation directory with two sub directories containing .pdf and .md files providing instructions on how to use the toolkit and edit the .auto.tfvars

› Description of the Generated files

Description of the Generated files:

Files Generated	At File Path	Comment/Purpose
Config File	/cd3user/tenancies/<customer_name>/<customer_name>_config	Customer specific Config file is required for OCI API calls.
setUpOCI.properties	/cd3user/tenancies/<customer_name>/<customer_name>_setUpOCI.properties	Customer Specific properties files will be created.
Region based directories	/rcd3user/tenancies/<customer_name>/terraform_files	Tenancy's subscribed regions based directories for the generation and segregation of terraform files.
Variables File, Provider File, Root and Sub modules	/cd3user/tenancies/<customer_name>/terraform_files/<region>	Required for terraform to work.
Public and Private Key Pairs	Copied from /cd3user/tenancies/keys/ to /cd3user/tenancies/<customer_name>/	API Keys that were previously generated are moved to customer specific out directory locations for easy access.
A log file with the commands to execute	/cd3user/tenancies/<customer_name>/cmds.log	This file contains a copy of the *Commands to execute* section of the *console output*.
Documentation folder	/cd3user/tenancies/<customer_name>/documentation/user_guide/ /cd3user/tenancies/<customer_name>/documentation/terraform/	These folders contain the PDF and .md files with instructions on how to use the toolkit and edit the .auto.tfvars .

Running the Automation Toolkit

Once the above scripts are executed successfully, choose the appropriate CD3 Excel Sheet template from below and update the setUpOCI.properties file at /cd3user/tenancies/<customer_name>/<customer_name>_setUpOCI.properties. Finally, run the commands displayed in the console output. These commands are also made available in the cmds.log file of the output directory for future reference.

Excel Sheet Templates - CIS Landing Zone

Below are the CD3 templates for the latest release having standardised IAM Components (compartments, groups and policies), Network Components and Events & Notifications Rules as per CIS Landing Zone and the CIS Foundations Benchmark for Oracle Cloud.

Details on how to fill the data into the excel sheet can be found in the Blue section of each sheet inside the excel file. Please refer CD3 Excel Information for additional details on each tab of the excel sheets.

Excel Sheet	Purpose
CD3-Blank-template.xlsx	Choose this template while exporting the existing resources to the CD3 and Terraform.
CD3-CIS-template.xlsx	This template has auto-filled in data of CIS Landing Zone. Choose this template while using a tenancy that supports DRGV2.
CD3-HubSpoke-template.xlsx	This template has auto-filled in data for a Hub and Spoke model of networking. The user must only modify the region according to requirement and execute the toolkit to generate the terraform files.
CD3-CIS-ManagementServices-template.xlsx	This template has auto-filled in data of CIS Landing Zone. Choose this template while creating the components of Events, Alarms and Notifications.
CD3-SingleVCN-template.xlsx	This template has auto-filled in data for a Single VCN model of networking. The user

Excel Sheet	Purpose
	must only modify the region according to requirement and execute the toolkit to generate the terraform files.

Here is the CIS Landing Zone quick start template by NACE Security Team also:

https://www.ateam-oracle.com/cis-oci-landing-zone-quickstart-template

ⓘ The Excel Templates can also be found at ***cd3_automation_toolkit/example*** in the ***Git repository*** or at ***/cd3user/oci_tools/cd3_automation_toolkit/example*** inside the ***container***.

setUpOCI.properties

Before we start with the steps to execute the Automation Toolkit, kindly update the properties file which is the input to the Toolkit.

Current Version: **setUpOCI.properties v10**

Example File: (This file can be found at /cd3user/oci_tools/cd3_automation_toolkit/. Make sure to use/modify the properties file at /cd3user/tenancies/<customer_name>/<customer_name>_setUpOCI.properties during executions)

```
[Default]

#Input variables required to run setUpOCI script

#path to output directory where terraform file will be generated. eg /cd3user/tenancies/<customer_tenancy_name>/terraform_files when running from cd3toolkit docker container
outdir=/cd3user/tenancies/demotenancy/terraform_files

#prefix for output terraform files eg client name
prefix=demotenancy

#input config file for Python API communication with OCI eg example\config; Leave it blank if code is being executed from OCS Work VM
config_file=

#params required if input data format is cd3
#path to cd3 excel eg example\CD3-template.xlsx
cd3file=/cd3user/tenancies/demotenancy/CD3-demotenancy-template.xlsx


#Is it a Non Green Field tenancy
non_gf_tenancy=false
```

Variable	Description	Example
outdir	Path to output directory where terraform files will be generated	/cd3user/tenancies/<customer_name>/terraform_files
prefix	Prefix for output terraform files	<customer_name>
config_file	Python config file	/cd3user/tenancies/<customer_name>/config
cd3file	Path to the CD3 input file	/cd3user/tenancies/<customer_name>/testCD3.xlsx
non_gf_tenancy	Specify if its a Non Green field tenancy or not (**True** or **False**)	False

Execution Steps:

Steps	Command
Change Directory to that of cd3_automation_toolkit	`cd /cd3user/oci_tools/cd3_automation_toolkit/`
Edit the setUpOCI.properties at location *<u>/cd3user/tenancies/<customer_name>/<customer_name>_setUpOCI.properties</u>* with appropriate values.	Place Excel sheet at appropriate location in your docker and provide the corresponding path in *<u>/cd3user/tenancies/<customer_name>/<customer_name>_setUpOCI.properties</u>* file
Execute the setUpOCI Script	`python setUpOCI.py /cd3user/tenancies/<customer_name>/<customer_name>_set`
	Choose "**Fetch Compartments OCIDs to variables file**" from **CD3 Services** in setUpOCI menu.

Additional Information:

Execute the command to fetch the details of the compartments if it already exists/created in OCI.

These details will be written to the terraform variables file.

Choose the right option by setting the property **non_gf_tenancy** of **setUpOCI.properties** , to toggle between the two workflows:

1. Set the property **non_gf_tenancy** to **false** for supporting **Green Field Tenancies**

   → this will help to **create** new resources

2. Set the property **non_gf_tenancy** to **true** for supporting **Non - Green Field Tenancies**
   → this will help to **export** existing resources **from OCI to CD3,**
   → create the terraform configuration files for them and
   → a shell script containing the import commands to import the state of exported components to the tfstate file.

Once the export (including the execution of ***tf_import_commands_<resource>_nonGF.sh***) is complete, switch the value of **non_gf_tenancy** back to **false**.

This allows the Tool Kit to support the tenancy as Green Field from this point onwards.

Steps to execute Automation Toolkit Workflows

Green Field Tenancies

Below are the steps that will help to configure the Automation Tool Kit to support the Green Field Tenancies:

Step 1: The CD3 Template can be found at location - /cd3user/oci_tools/cd3_automation_toolkit/example or can be downloaded from cd3

For the Core OCI Objects (IAM, Tags, Networking, Instances, LBR, Storage, Databases) - use the **CD3-SingleVCN-template.xlsx** file or **CD3-HubSpoke-template.xlsx** or **CD3-CIS-template.xlsx** based on the requirement.

For Events, Notifications, Alarms and Service Connector Hub - use the **CD3-CIS-ManagementServices-template.xlsx** file.

Step 2: Fill the CD3 file with appropriate values specific to the client and put at the appropriate location**.**

Modify/Review **setUpOCI.properties**: (**non_gf_tenancy** set to **false**)

```
[Default]

#Input variables required to run setUpOCI script

#path to output directory where terraform file will be generated. eg /cd3user/tenancies/<customer_name>/terraform_files when running from OCS VM
outdir=/cd3user/tenancies/demotenancy/terraform_files

#prefix for output terraform files eg client name
prefix=demotenancy

#input config file for Python API communication with OCI eg example\config; Leave it blank if code is being executed from OCS Work VM
config_file=

#params required if input data format is cd3
#path to cd3 excel eg example\CD3-template.xlsx
cd3file=/cd3user/tenancies/demotenancy/CD3-demotenancy-template.xlsx

#Is it Non GreenField tenancy
non_gf_tenancy=false

```

Step 3: Execute the SetUpOCI.py script to start creating the terraform configuration files.

Command to Execute:

python setUpOCI.py <path_to_setupOCI.properties>

Example execution of the wrapper script:

```
0.  Validate CD3
1.  Identity
2.  Tags
3.  Network
4.  Compute
5.  Storage
6.  Database
7.  Load Balancers
8.  Management Services
9.  Developer Services
10. CIS Compliance Features
11. CD3 Services
q.  Press q to quit

See example folder for sample input files

Enter your choice (specify comma separated to choose multiple choices): 0,1,2
```

Choose the resources by specifying a single option (for choosing one of these resources) or comma-separated values (to choose multiple resources) as shown in the sample screenshot above.

Step 4: Change your directory to /cd3user/tenancies/<customer_name>/terraform_files/<region>/.

Execute ***terraform init*** - To initialize and prepare your working/out directory so Terraform can run the configuration.

terraform plan - To preview any changes before you apply them.

terraform apply - To make the changes defined by Terraform configuration to create, update, or destroy resources in OCI.

Note: Make sure to execute "**Fetch Compartments OCIDs to variables file**" from **CD3 Services** in setUpOCI menu after you create Compartments. This will ensure that the variables file in *outdir* is updated with the OCID information of all the compartments.

Non-Green Field Tenancies

Note:

1. Course of actions involved in Exporting objects from OCI -
   - Automation Tool Kit fetches the data for the cd3 supported services from all the regions the tenancy is subscribed to. Data is written to appropriate sheets of the CD3 based on the resources being exported.
   - Tool Kit then generates the TF configuration files/auto.tfvars files for these exported resources.
   - It also generates a shell script - **tf_import_commands_<resource>_nonGF.sh** that has the import commands, to import the state of the resources to tfstate file. (This helps to manage the resources via Terraform in future).

Below are the steps that will help to configure the Automation Tool Kit to support the Non - Green Field Tenancies:

Step 1: Choose the right CD3 format for exporting the contents from OCI.

Two different formats of CD3 to be used : (An example of these files can be found at location - /cd3user/oci_tools/cd3_automation_toolkit/example or can be downloaded from cd3)

CD3-Blank-template.xlsx - Use this format of the Excel sheet to export objects like Network Components, Identity Components, Core Infra Components , DB Components and Tags.

CD3-CIS-ManagementServices-template.xlsx - Use this format of the Excel sheet to export Events, Notifications, Alarms and Service Connector Hub.

Step 2: Fill up/review the **setUpOCI.properties** file.

Once the CD3 format is chosen, fill the sheets with appropriate values and put it at the appropriate location.

Modify **setUpOCI.properties** as shown below: (**non_gf_tenancy** set to **true**)

```
[Default]

#Input variables required to run setUpOCI script

#path to output directory where terraform file will be generated. eg /cd3user/tenancies/<customer_name>/terraform_files when running from OCS VM
outdir=/cd3user/tenancies/demotenancy/terraform_files

#prefix for output terraform files eg client name
prefix=demotenancy

#input config file for Python API communication with OCI eg example\config; Leave it blank if code is being executed from OCS Work VM
config_file=

#params required if input data format is cd3
#path to cd3 excel eg example\CD3-template.xlsx
cd3file=/cd3user/tenancies/demotenancy/CD3-demotenancy-template.xlsx

#Is it Non GreenField tenancy
non_gf_tenancy=true
```

Step 3: Execute the SetUpOCI.py script to start exporting the resources to CD3 and creating the terraform configuration files.

Command to Execute:

python setUpOCI.py <path_to_setupOCI.properties>

Example execution of the wrapper script:

```
1.  Export Identity
2.  Export Tags
3.  Export Network
4.  Export Compute
5.  Export Storage
6.  Export Databases
7.  Export Load Balancers
8.  Export Management Services
9.  Export Developer Services
10. CD3 Services
q.  Press q to quit
Enter your choice (specify comma separated to choose multiple choices): 1,2,3
```

Choose the resources by specifying a single option (for choosing one of these resources) or comma-separated values (to choose multiple resources) as shown in the sample screenshot above.

Make sure to execute "**Fetch Compartments OCIDs to variables file**" from **CD3 Services** in setUpOCI menu at least once. This will ensure that the variables file in *outdir* is updated with the OCID information of all the compartments.

Tabs- Exported OCI data will over-write to the specific CD3 sheets while the other sheets remain intact.

Expected Outputs:

a. *Excel sheet with the resource details from OCI*
b. *Terraform Configuration files - *.auto.tfvars*
c. *Shell Script with import commands - tf_import_commands_<resource>_nonGF.sh*

Action: Execute the *tf_import_commands_<resource>_nonGF.sh* files that are generated in the outdir.

The terraform plan should show that infrastructure is up-to-date with no changes required for all regions.

```
Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan, but will not be
persisted to local or remote state storage.


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

No changes. Infrastructure is up-to-date.

This means that Terraform did not detect any differences between your
configuration and real physical resources that exist. As a result, no
actions need to be performed.
```

Output

Terraform Modules.

Releases

Latest - (Date - Jan 13th, 2023)

Automation Toolkit Release v10

Docker Image Release v6.0

Explore

Scenario Based Workflows

Support for Additional Attributes

Support for New Region or New Protocol

Support for CD3 Validator

OCI Resource Manager Upload

Known Behaviour Of Automation Toolkit

No labels