

Known Behaviour Of Automation Toolkit

Created by Suruchi Suruchi, last modified by Shruthi Subramanian on Sep 28, 2022

- Known Behaviour of Terraform** - Create a Load Balancer with Reserved IP: When you create a LBaaS with reserved ip as "Y" and do a terraform apply, everything will go smooth and be in sync for the first time. If you do a terraform plan immediately (post apply), you will find that the plan changes the private ip of load balancer to null.

```
# module.lbr-reserved-ips["LB-1-reserved-ip"].oci_core_public_ip.public_ip will be updated in-place
~ resource "oci_core_public_ip" "public_ip" {
  id           = "ocid1.publicip.oc1.phx.amaaaaaav7zeaxyatbw6k7gbtazbyxm4kkcyxh1chviazpf3g4ecackk27a"
  ~ private_ip_id = "ocid1.privateip.oc1.phx.abyhq1jrr71xrj7epwv1f2ycxcwmtgye43zj1n6s4ssuqj2qcejukhcqb6q" -> null
  # (11 unchanged attributes hidden)
}
```

Plan: 0 to add, 1 to change, 0 to destroy.

This is a behaviour of Terraform. In these scenarios, please add the private ip ocid to the auto.tfvars as shown below before you run a terraform plan again.

```
// Copyright (c) 2021, 2022, Oracle and/or its affiliates.
#####
# Network
# Create Reserved IPs for Load Balancers
# Allowed Values:
# Lifetime can be EPHEMERAL or RESERVED
# private_ip_id and public_ip_pool_id can be the ocids or the key of private_ips (map) and public_ip_pools (map) respectively
# compartment_id can be the ocid or the name of the compartment hierarchy delimited by double hiphens "--"
# Example : compartment_id = "ocid1.compartment.oc1..aaaaaaaahwwiefb56epvdl2fic6ah6jy3xf3c" or compartment_id = "Network-root-cpt--Network" where
#####
lbr_reserved_ips = {
  LB-1-reserved-ip = {
    compartment_id = "Network"
    display_name   = "LB-1-reserved-ip"
    lifetime       = "RESERVED"
    private_ip_id  = "ocid1.privateip.oc1.phx.abyhq1jtdrswxfq4t4ny2gu4a2daiskrf3lwaucdhgchab2ulqdo5duspa"
    public_ip_pool_id = ""
    defined_tags   = {}
    freeform_tags  = {}
  },
}
##Add New Load Balancer Reserved IPs for phoenix here##
}
```

Once you do the above change, and then execute a terraform plan/apply, you will get the below error and it can be ignored.

```
module.lbr-reserved-ips["LB-1-reserved-ip"].oci_core_public_ip: Modifying... [id=ocid1.publicip.oc1.phx.amaaaaaav7zeaxya4u6nzsrfwypzoi27su4zq22hfxjebzv5k6dvov56upva]

Error: 404-NotAuthorizedOrNotFound, PublicIp cannot be assigned to or unassigned from PrivateIp ocid1.privateip.oc1.phx.abyhq1jtdrswxfq4t4ny2gu4a2daiskrf3lwaucdhgchab2ulqdo5duspa as it is managed by ocid1.loadbalancer.oc1.phx.aaaaaaaabwmu3nfbhuwtekb4kzcxdtdj7ipzqnvey4dyditudsegm3qwgkq
Suggestion: Either the resource has been deleted or service Core Public Ip need policy to access this resource. Policy reference: https://docs.oracle.com/en-us/iaas/Content/Identity/Reference/policyreference.htm
Documentation: https://registry.terraform.io/providers/oracle/oci/latest/docs/resources/core_public_ip
API Reference: https://docs.oracle.com/iaas/api/#/en/iaas/20160918/PublicIp/UpdatePublicIp
Request Target: PUT https://iaas.us-phoenix-1.oraclecloud.com/20160918/publicips/ocid1.publicip.oc1.phx.amaaaaaav7zeaxya4u6nzsrfwypzoi27su4zq22hfxjebzv5k6dvov56upva
Provider version: 4.34.0, released on 2022-09-21.
Service: Core Public Ip
Operation Name: UpdatePublicIp
OPC request ID: 99e44b994e4a56f19be77337f7a8e91a/1237548353A37EEAAA16B6FAFFC994F/E91C303EFD36D0C81EC68515835407D8

with module.lbr-reserved-ips["LB-1-reserved-ip"].oci_core_public_ip.public_ip,
on modules\ip\reserved-public-ip\main.tf line 8, in resource "oci_core_public_ip" "public_ip":
8: resource "oci_core_public_ip" "public_ip" {
```

- While exporting and synching the tfstate file for LBaaS Objects, the user may be notified that a few components will be modified on apply. In such scenarios, add the attributes that the Terraform notifies to be changed to the appropriate CD3 Tab of Load Balancer and Jinja2 Templates (as a non-default attribute) and re-run the export. (To learn about how to add attributes, refer [Support for additional attributes - Flat TF Files](#))

- Add a new column - "Freeform Tags" to the CD3 Excel Sheets as per necessity, to export the tags associated with the resource as well. If executed as-is, Terraform may prompt you to modify resources based on Tags.

Example:

- ```
module.vcns["demo-vcn"].oci_core_vcn.vcn will be updated in-place
~ resource "oci_core_vcn" "vcn" {
 ~ freeform_tags = {
 - "Environment" = "Demo" -> null
 }
 id = "ocid1.vcn.oc1.iad.amaaaaaajycshisuahakjcnawhfwahuhaueckknwchw"
 # (12 unchanged attributes hidden) # (1 unchanged block hidden)
}
```

Plan: 0 to add, 1 to change, 0 to destroy.

- When you execute export of Security Rules, Route Rules, DRG Route Rules and run a Terraform plan, it should have synced up TF state with OCI console and should only show changes to add default security list of each VCN to the terraform.

From now on, you can use cd3 to add rules to the default security list and default route tables as well.

Similarly, it will create TF for only those DRGs which are part of CD3 and skip Route Tables for the DRGs created outside of CD3. This will also synch DRG rules in your tenancy with the terraform state.

**NOTE:** When there are changes made in the OCI console manually, the above options of export and modify can be helpful to sync up the contents/objects in OCI to TF.

- Match All criteria specified for Route Distribution Statement in DRGs sheet will show below output each time you do terraform plan:

```
Terraform will perform the following actions:

oci_core_drg_route_distribution_statement.Hub-nprddrg_DRG-VCNs-RT-DN3_statement1 will be updated in-place
~ resource "oci_core_drg_route_distribution_statement" "Hub-nprddrg_DRG-VCNs-RT-DN3_statement1" {
 id = "drgRouteDistributions/ocid1.drgroutedistribution.oc1.phx.aaaaaaaayvicr73xulntve45w7leggh7qdtvg1f2xr7vaosjl35gni4qhzzq/statements/132"
 # (3 unchanged attributes hidden)

 + match_criteria {}

 # (1 unchanged block hidden)
}
```

The service api is designed in such a way that it expects an empty list for match all. And it sends back an empty list in the response every time. Hence this behaviour from terraform side. This can be safely ignored.

- Export process for non greenfield tenancies v6.0 or higher doesn't support OCI objects having duplicate names like for Policies, for VCNs having same names within a region or subnets having same names within a VCN. You will get output similar to below when terraform plan is run (Option 3 with non-gf\_tenancy set to true)

Error: Duplicate resource "oci\_identity\_policy" configuration

- Export process for non greenfield tenancies v6.0 or higher will try to revert SGW for a VCN to point to all services if it was existing for just object storage. You will get output similar to below when terraform plan is run (Option 3 with non-gf\_tenancy set to true)

```
oci_core_service_gateway.VCN_sgw will be updated in-place
~ resource "oci_core_service_gateway" "VCN_sgw" {

 block_traffic = false
 compartment_id = "ocid1.compartment.oc1..aaaaaaaahsesjfw5hhftccsvndbufdlf5ca2c3q3clyvwg4wngj4ej26i3ya"

 display_name = "VCN_sgw"
 freeform_tags = {}

 id = "ocid1.servicegateway.oc1.iad.aaaaaaajqtpjyq7ihgikmug5kbz55pzyrmt7m6t4ijlqek5ujqg3qxeaxma"
 state = "AVAILABLE"

 time_created = "2019-03-19 16:46:33.859 +0000 UTC"
 vcn_id = "ocid1.vcn.oc1.iad.aaaaaaaazjup6ahpesjgrjyaxr2bcnx44tpn3ygvx2tjlytqkub5ikl6rha"

 - services {
 - service_id = "ocid1.service.oc1.iad.aaaaaaa74z6sqsezqf6znyomdp5jkrfwb4j2ol33abgosvnhxcqphy3eaq" -> null
 - service_name = "OCI IAD Object Storage" -> null
 }

 + services {
 + service_id = "ocid1.service.oc1.iad.aaaaaaaam4zfmy2rjue6fmglumm3czgisxzmrvrwqeodtzg7hwa272mlfna"
 + service_name = (known after apply)
 }

 timeouts {}
}
```

- Modification Process - When there is a change to any of the instance parameters, the volume policy assignment will get recreated. Example: If you want to add a NSG to the instance, you will be notified on a read to the boot volume ocids, modification to the instance and, a create and destruction of the policy assignment. Screenshot below for reference. This behaviour is expected and in order to not have the backup policy recreated, comment the depends\_on statement in modules/compute/data.tf at line around 35.

```
+ filter {
+ name = "state"
+ values = [
+ "AVAILABLE",
+]
+ }
}

module.instances["Test1-2"].oci_core_instance.instance will be updated in-place
~ resource "oci_core_instance" "instance" {
 id = "oci1.instance.oc1.uk-london-1.anwgiljsv7zeaxyc72tptggcsgfgry2rik13n76tl5zyquiujd5wjuvsfzla"
 # (19 unchanged attributes hidden)

 ~ create_vnic_details {
 ~ nsg_ids = [
 "oci1.networksecuritygroup.oc1.uk-london-1.aaaaaaatj4rdxpdjrcfbm6t453falramt3rdubrzfga2k35tfuy27v3m3va",
]
 # (9 unchanged attributes hidden)
 }

 # (7 unchanged blocks hidden)
}

module.instances["Test1-2"].oci_core_volume_backup_policy_assignment.volume_backup_policy_assignment[0] must be replaced
- resource "oci_core_volume_backup_policy_assignment" "volume_backup_policy_assignment" {
 ~ asset_id = "oci1.bootvolume.oc1.uk-london-1.abwgiljs3hhhe7lw64rrpyntvd2snavjksbsjddjv36ribouqa3au2dnysa" -> (known after apply) # forces replacement
 ~ id = "oci1.volumebackuppolicyassign.oc1.uk-london-1.abwgiljs7bbk75erjz6xogb4g5pqskkiiifbsjlogwifqmxku52iefjwgzba" -> (known after apply)
 ~ time_created = "2022-05-25 12:16:00.956 +0000 UTC" -> (known after apply)
 # (1 unchanged attribute hidden)

 - timeouts {}
}
```

Plan: 1 to add, 1 to change, 1 to destroy.

No labels

