Creating a highly available and scalable Webservice with OCI

Important to know for this meetup

login to OCI console:

login url: https://console.eu-frankfurt-1.oraclecloud.com

tenant : oci_core_emea_sc_dolder

user: meetociXX pw: Code4fun.meetoci, XX=01-24

login to wserver vm:

ssh opc@publicIP, authentication by ssh key, get publicIP from OCI console

NOTICE: we are all operating with the same passwords and keys, so please stay with the user 'meetociXX' that was assigned to you.

AD Distribution list for server placement by user, in case we are more than 16 participants

User	AD1	AD2	AD3
meetoci01	X	X	
meetoci02	X	X	
meetoci03	X	X	
meetoci04	X	X	
meetoci05	X	X	
meetoci06	X	X	
meeotci07	X	X	
meetoci08	X	X	
meetoci09	X		X
meetoci10	X		X
meetoci11	X		X
meetoci12	X		X
meetoci13	X		X
meetoci14	X		X
meetoci15	X		X
meetoci16	X		X
meetoci17		X	X
meetoci18		X	X
meetoci19		X	X
meetoci20		X	X

meetoci21	X	X
meetoci22	X	X
meetoci23	X	X
meetoci24	X	X

OCI Console LAB:

- Make sure you're working in the us-ashburn-1 region
- Create VCN (simple, no automatic default provisioning, name: meetociXX-VCN)
- Create subnet in VCN (either in AD-1 or AD-2, according to instructions), name: meetociXX-SN1), use default route table, default security list, default dhcp
- Create internet gateway for this VCN, name. meetociXX-IGW
- Create route to IGW in default route
- Create compute instance (shape vm1.2) in AD-1, OL 7.5 latest, name: meetociXX-WS1, configure supplied ssh-keys for subsequent access
- When ready, login to newly created vm 'opc@publicIP' using a key based login
    - Using yum, install 'httpd', create firewall rules, create simple index.html

  $ sudo yum -y install httpd

  $ sudo systemctl enable httpd

  $ sudo firewall-cmd –permanent –add-key=80/tcp

  $ sudo firewall-cmd –reload

  $ sudo vi /etc/selinux/config (➔ SELINUX=disabled)

  $ sudo echo "some text" >/var/www/html/index.html

  c$ sudo vi /var/www/cgi-bin/info.cgi ➔ copy/paste supplied cgi example into file

  $ sudo chmod +x /var/www/cgi-bin/info.cgi ➔ make .cgi executable

- Use OCI console to reboot server (needed to disable selinux)
- Test if webserver reachable and working by issuing from browser:
    - publicIP of server ➔ should return "some text" from index.html
    - publicIP/cgi-bin/info.cgi ➔ should return results of cgi script
    - HINT: if it doesn't work, checking route tables (route to IGW?) and security lists (is port 80 traffic into server allowed from the internet)
- If webpages display successfully from the internet, then repeat all steps from above to create a second server in another domain (which really means that they're geographically separated):
    - Basically you create a second subnet in domain AD-2/3, a second server using the AD-2/3 subnet. IGW, security lists, dhcp options we retain, they support region scope.
    - Then do all configuration work until you successfully can display the webpages using the 2nd server's publicIP address.

- Create a new security list which will be used along two additional subnets in which the load balancer, name: meetociXX-LBSEC, delete all rules such that this list is empty. It will be autofilled by the LB creation process.
- Create another 2 subnets in the same AD's where the server are deployed. Those will 'host' the load balancer. Attach the newly defined security list, default route and dhcp
- Create the loadbalancer, name: meetociXX-LB using the newly created subnets, ensure automatic security list setting
- Create backend set (using port 80, protocol http and url '/' for healthcheck) ,then add the OCID's of the two webserver compute instance, *tip: duplicate console tab so you can point one to the servers to use the copy OCID function,* name: meetociXX-BS
- Create the listener, http. Port 80, using the newly created 'backendset', name: meetociXX-LS
- When all objects ready, note LB publicIP and try in browser 'publicIP' and 'publicIP/cgi-bin/info.cgi' ➔ refreshing should change between the two webservers as the loadbalancer is set to 'round-robin'
    - HINT: if it doesn't work, check security list rules, is INGRESS traffic to the listener authorized

Note: before starting the 'terraform' labs, make sure the interactively created objects are deleted. Normally the sequence:

- Terminate loadbalancer
- Terminate the 2 webserver compute instances
- Terminate the VCN

Should completely remove the objects you created.

Terraform LAB:

- Login to the terraform server (located in OCI) using: **meetociXX@130.61.86.247,** pw: Code4fun.meetoci
- Besides 'terraform' your user has also configured the 'OCI CLI' interface, a quick test to check if it works for you is to query all OCI regions that are available by issuing:
    - oci iam region list
- to enable 'terraform' operation, some shell variables need to be defined, activate them by issuing in your login directory:
    - source tf-env-vars
- change to the tftest directory and issue the following commands to check proper terraform operation:
    - terraform init
    - terraform plan
    - terraform apply ➔ should return the AD-1 identifier of the actual region (no objects created)
- Lab1:

    - -quick tf real create functional test creating a VCN
    - Copy the .tbe file to the same name without .tbe (thus keeping the template if you messup somehow)
    - Apply changes as outlined below
- Lab2:
    - Create a complete loadbalancer setup with terraform.
    - Same copy operation as before to get a .tf file
    - Do all changes as outlined below
    - Create/Inspect/Change/Destroy the loadbalancer using terraform
- Lab3:
    - The lab2 terraform script does not deploy ssh keys to the webserver hosts, so no way you could log into those.
    - Try to find out (online documentation) what needs to be changed to deploy ssh keys and modify the lab2 terraform .tf accordingly
    - If you can't find the solution, the Lab3 folder contains a .tbe which includes the needed changes (a diff of Lab2 and Lab3 .tbe's will reveal)
- Lab4:
    - Further modify the lab2 script to now scale out the webserver infra by adding a third node in the AD-3 which becomes part of the loadbalancer configuration
    - If you can't find the solution, the Lab4 folder contains a .tbe which includes the needed changes (a diff of Lab3 and Lab4 .tbe's will reveal)

Changes to apply to .tbe files to personalize for your user:

replace:
XX = 2-digit user#
YY= user# * 10, no leading zeros
D#1 = 0 (index into AD's, returns definition for AD-1)
D#2 = 1 (index into AD's, returns definition for AD-2)
D#3 = 2 (index into AD's, returns definition for AD-3)

Additional things to try:

- Delete one of the servers using 'terraform delete target='resoruce'
- Use 'terraform refresh' to view new status
- Check in browser that webpage still works but without deleted server
- Check with 'terraform plan' if a subsequent 'apply' would re-add deleted objects
- Run re-add