Modify Load Balancer health probes

Overview

Platform:	**Azure**
Owner of this SOP:	Fully Managed POD A
Cloud Services:	Instance Management

Problem

When requesters are asking to modify the load balancer health probes.

tester	Reviewer
@ Ramkumar Samudram (Deactivated) - Win @ Asif Patel (Deactivated) - Linux	

Solution

Cloud Services operators will perform basic health checks and update the associated record as required. In most cases, the record will be an Incident Task (which may be assigned or manually created).

Pre-requisites

- The operator needs to have access to the **Cloud Factory** ServiceNow Dashboard. If this access is missing raise a request for Access to ServiceNow and request the group “Cloud Factory - Dashboard Access”.
- Must be a member of the “Cloud Svcs- CloudOne” assignment group to work on ServiceNow tickets. As well access to the required report in this SOP. Use the same Access to ServiceNow request to gain this group if needed.
- The operator needs to be part of **OperationsCloudOneAccess** role in AWS & **CloudOne Operator** role in Azure . If this is missing raise a Manage Cloud Access request to get this custom role.

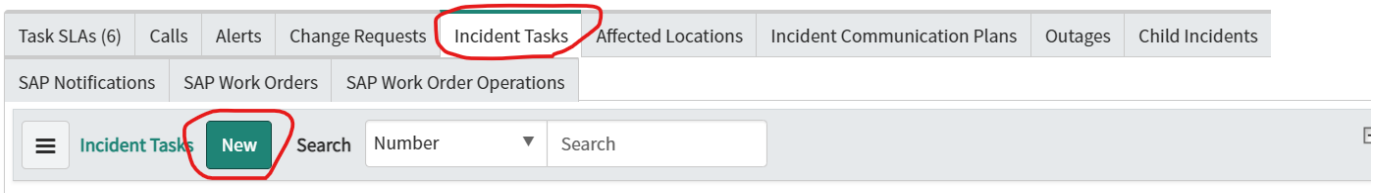
Procedure

Steps to create & manage Incident Task

Create Incident Task

Follow the below steps to create incident task. This step is required when the operator has received only an Incident.

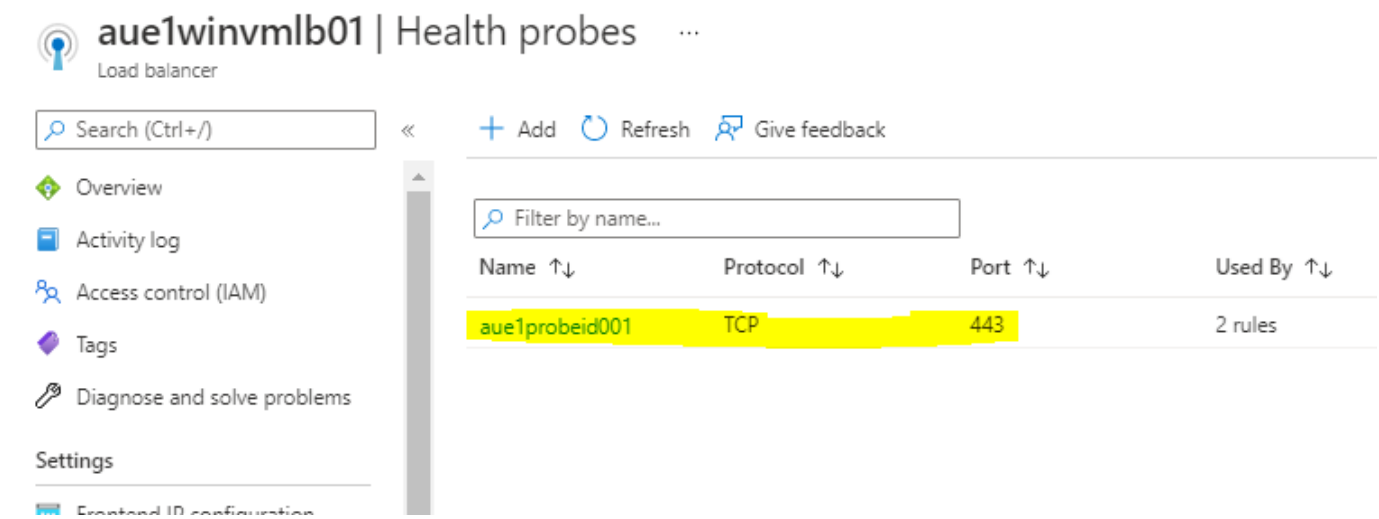
On *Incident Tasks* Tab of the Incident ticket, click *New* and assign it to “CloudOne” team. Populate the Assignment group and Configuration item (CI). (Hint the CI is the VM on which the health check is being performed)



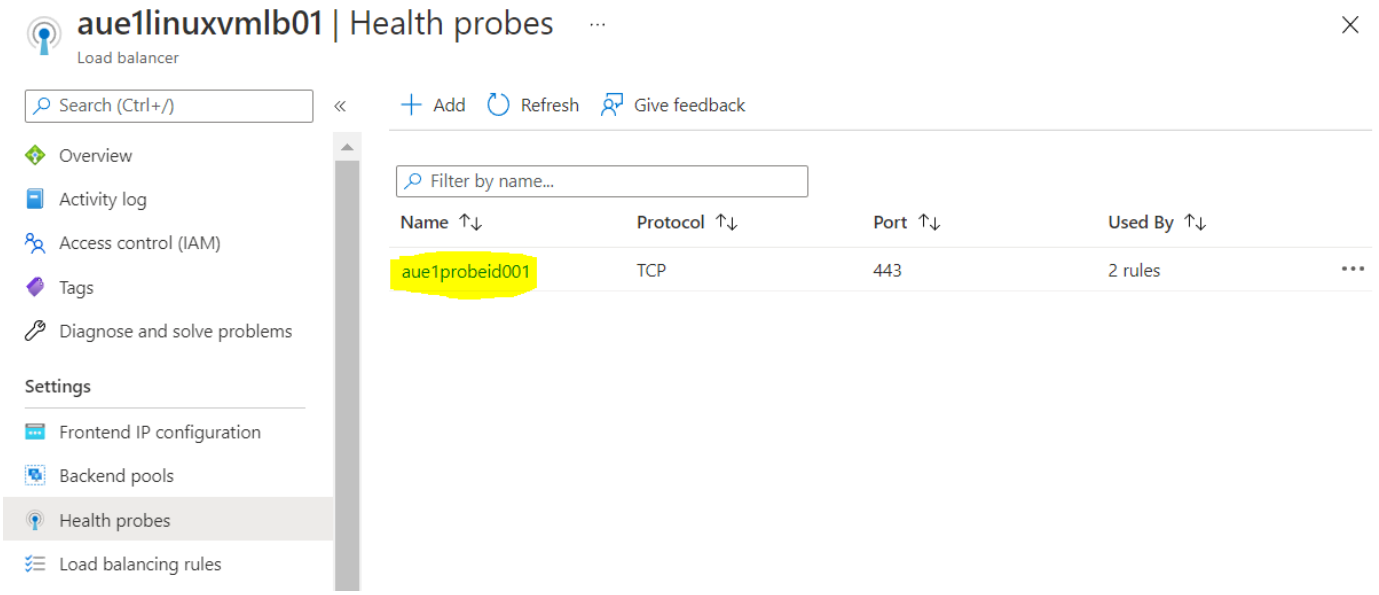
Steps

- Log on to Azure Portal and check the existing health probe.

Windows:

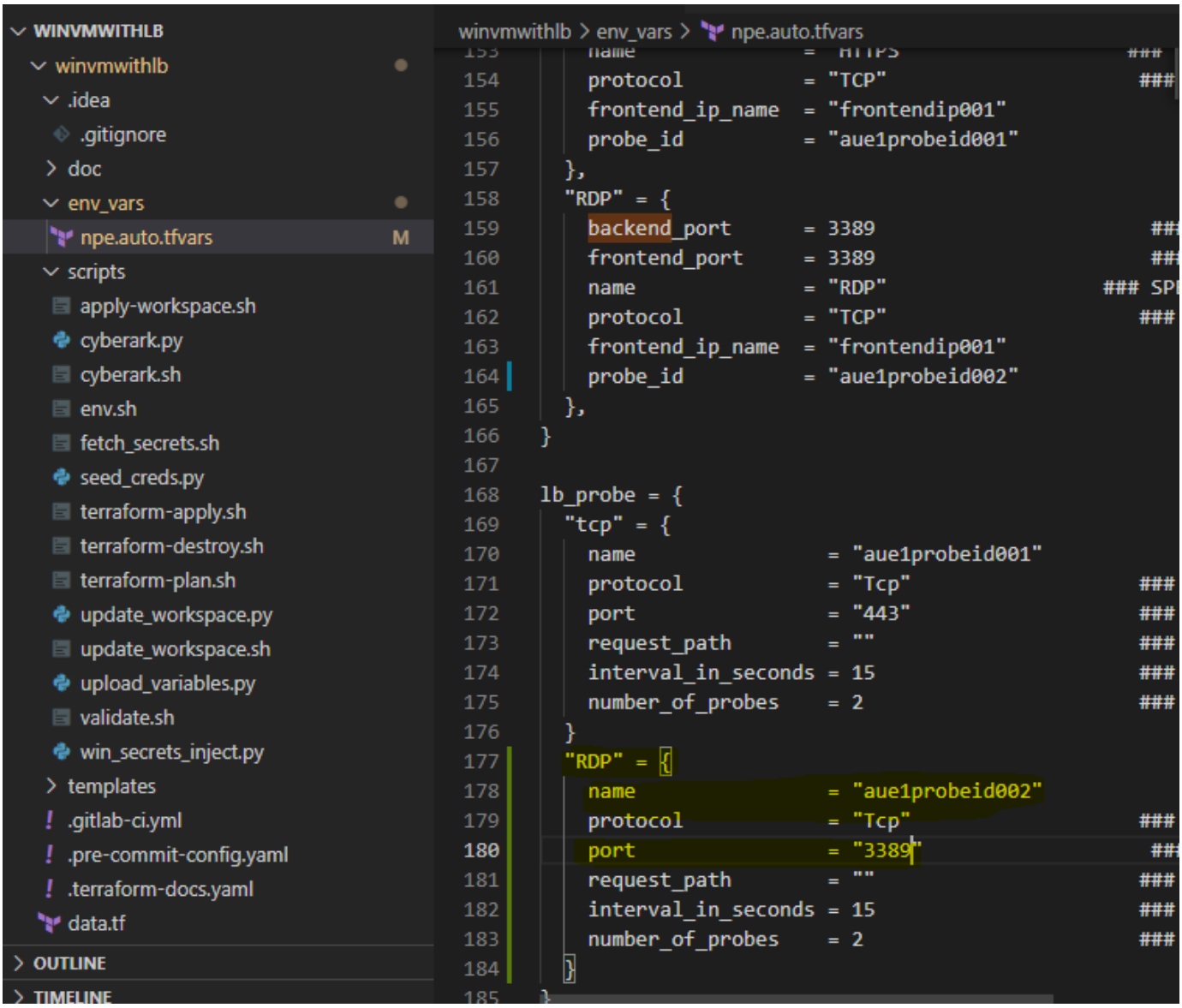


Linux:

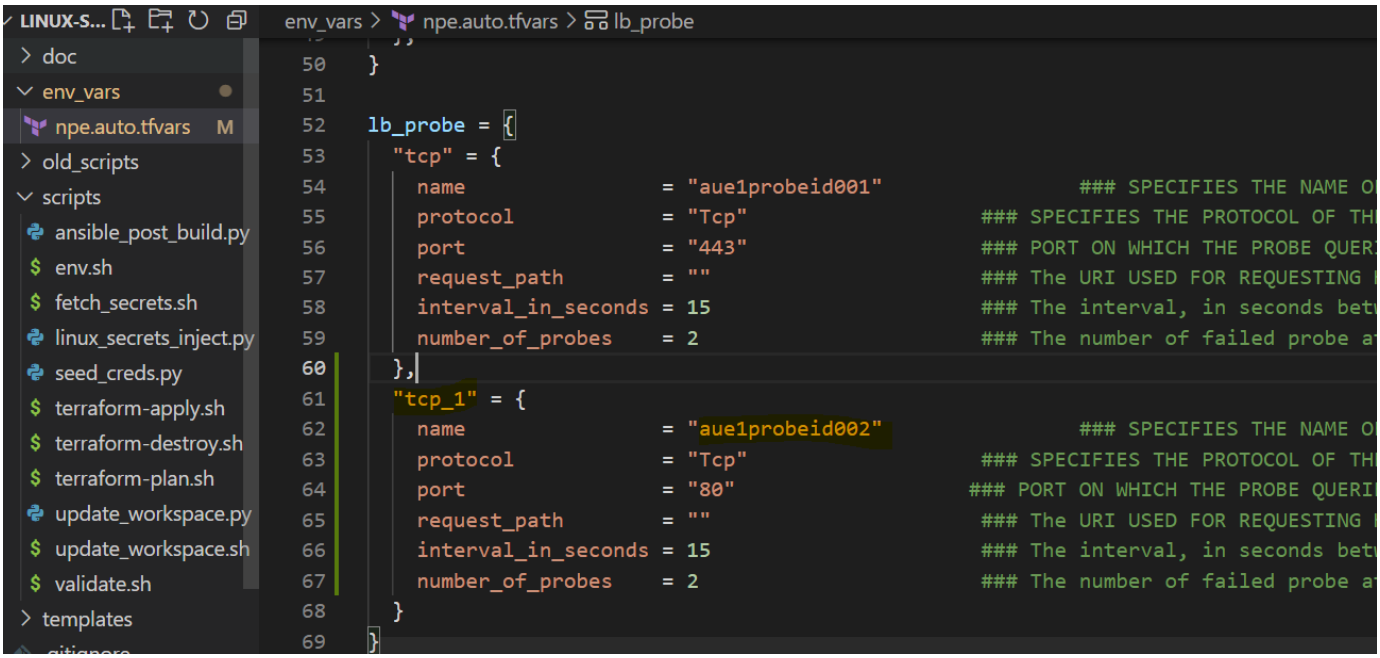


- Clone the respective repo and change the required variables into .tfvars file.

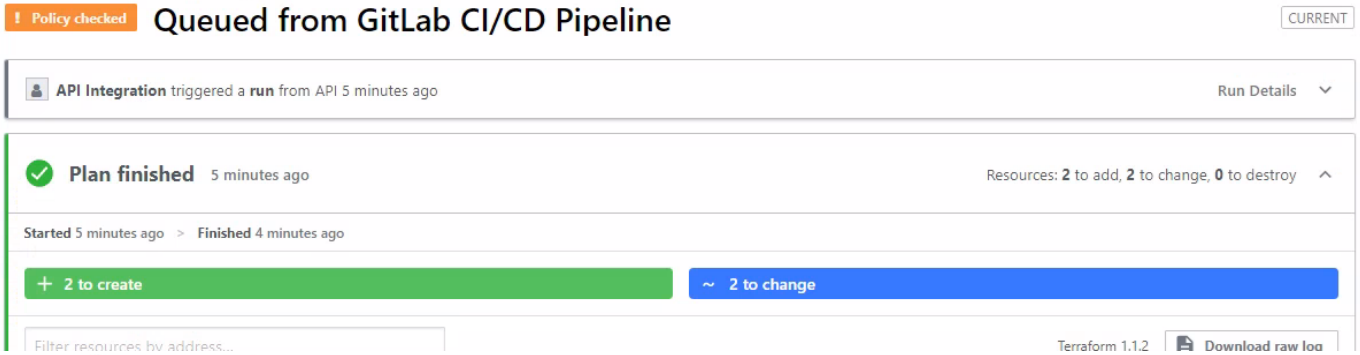
Windows:

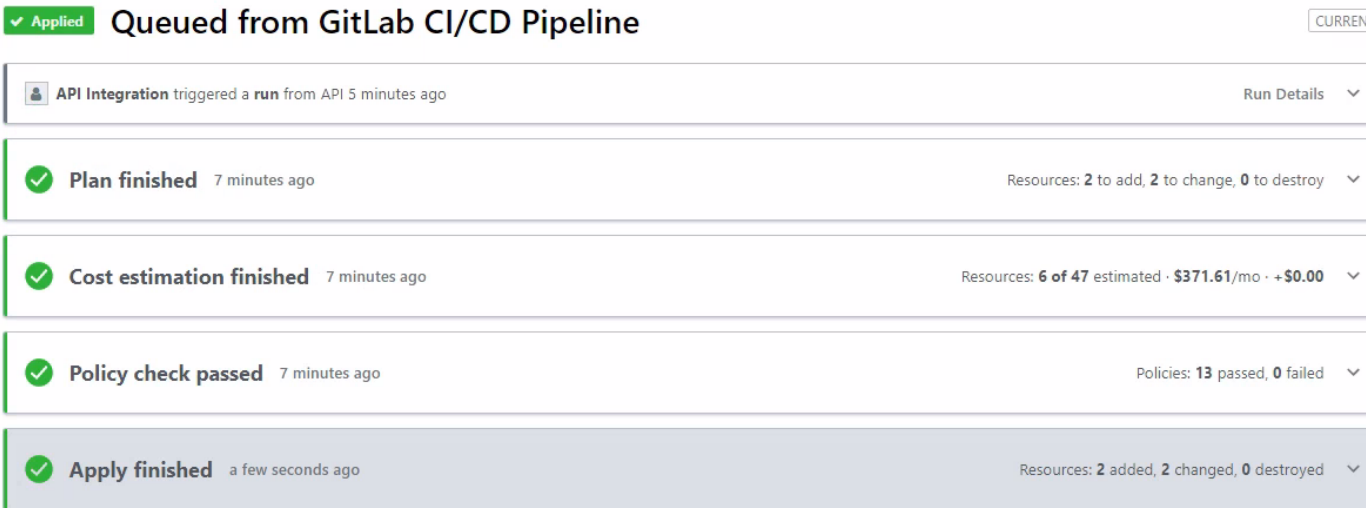

WINVMWITHLB
winvmwithlb
.idea
.gitignore
doc
env_vars
npe.auto.tfvars
M
scripts
apply-workspace.sh
cyberark.py
cyberark.sh
env.sh
fetch_secrets.sh
seed_creds.py
terraform-apply.sh
terraform-destroy.sh
terraform-plan.sh
update_workspace.py
update_workspace.sh
upload_variables.py
validate.sh
win_secrets_inject.py
templates
.gitlab-ci.yml
.pre-commit-config.yaml
.terraform-docs.yaml
data.tf
OUTLINE
TIMELINE
winvmwithlb > env_vars > npe.auto.tfvars
protocol = "TCP"
frontend_ip_name = "frontendip001"
probe_id = "aue1probeid001"
"RDP" = {
backend_port = 3389
frontend_port = 3389
name = "RDP"
protocol = "TCP"
frontend_ip_name = "frontendip001"
probe_id = "aue1probeid002"
lb_probe = {
"tcp" = {
name = "aue1probeid001"
protocol = "Tcp"
port = "443"
request_path = ""
interval_in_seconds = 15
number_of_probes = 2
"RDP" = {
name = "aue1probeid002"
protocol = "Tcp"
port = "3389"
request_path = ""
interval_in_seconds = 15
number_of_probes = 2

Linux:



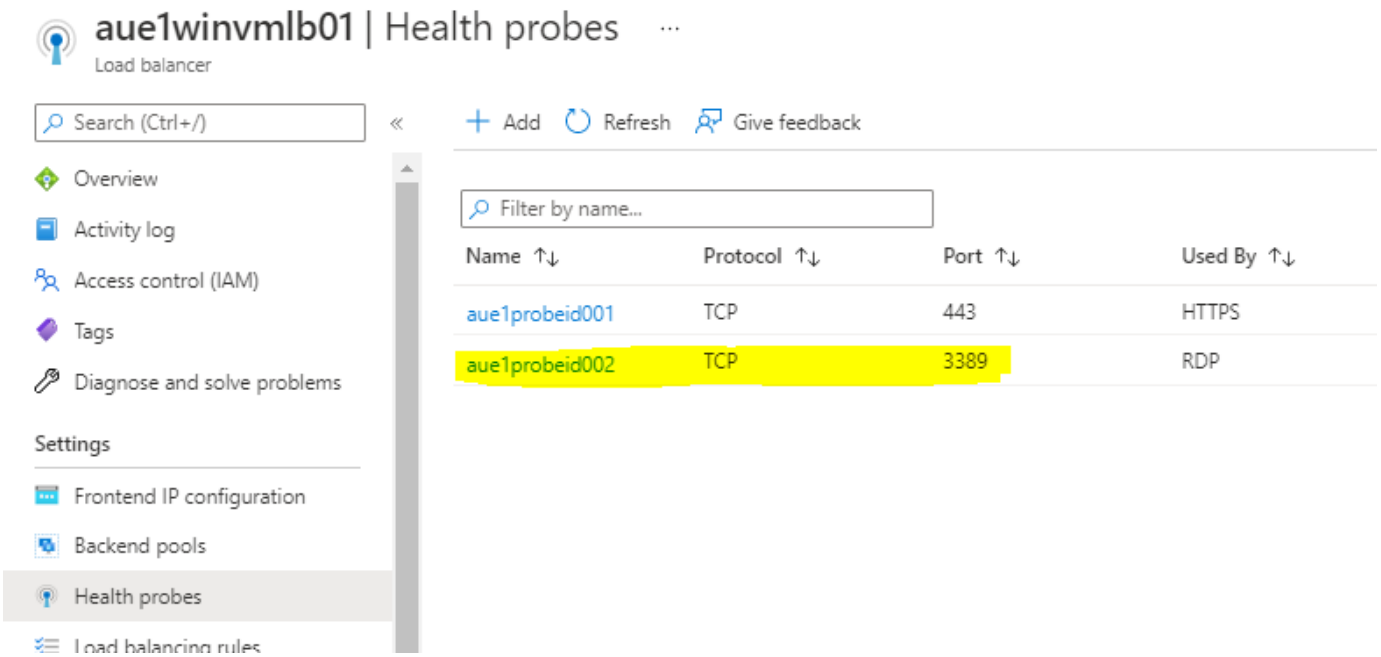
- commit and push your changes
- verify that the pipeline executes as expected and that a terraform workspace is created and run
- verify the terraform plan in your TF workspace
- if you are satisfied with the plan output, Initiate the apply stage in pipeline



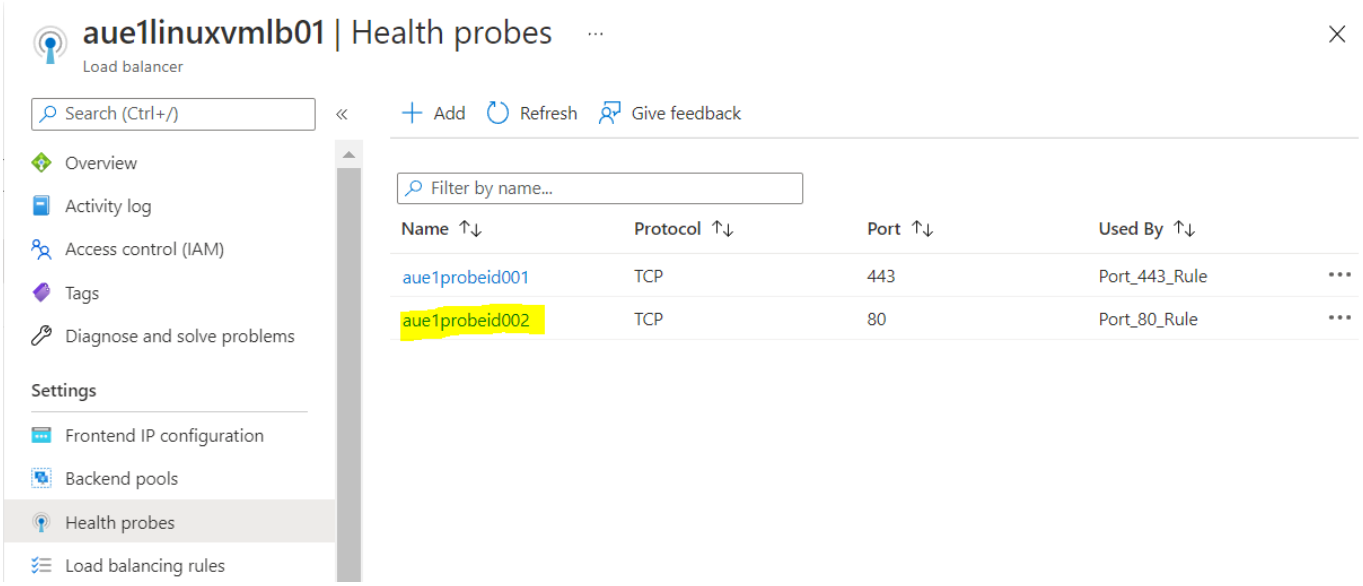


- Log on to the Azure portal and verify that whether the new port is added to the health probe as per requirement.

Windows:



Linux:



Related articles

- AWS VPC Endpoints configuration change
- Renewal / Upload SSL certificate for existing Application
- Monitoring GitLab Access Audit Report
- Azure Storage Account Blueprint SOP - BPAZR022
- Azure App Service FTP Credential Sharing and Reset Procedure