Load balancer

Creating a two tier load balancer application for showcasing use of internal and external load balancer

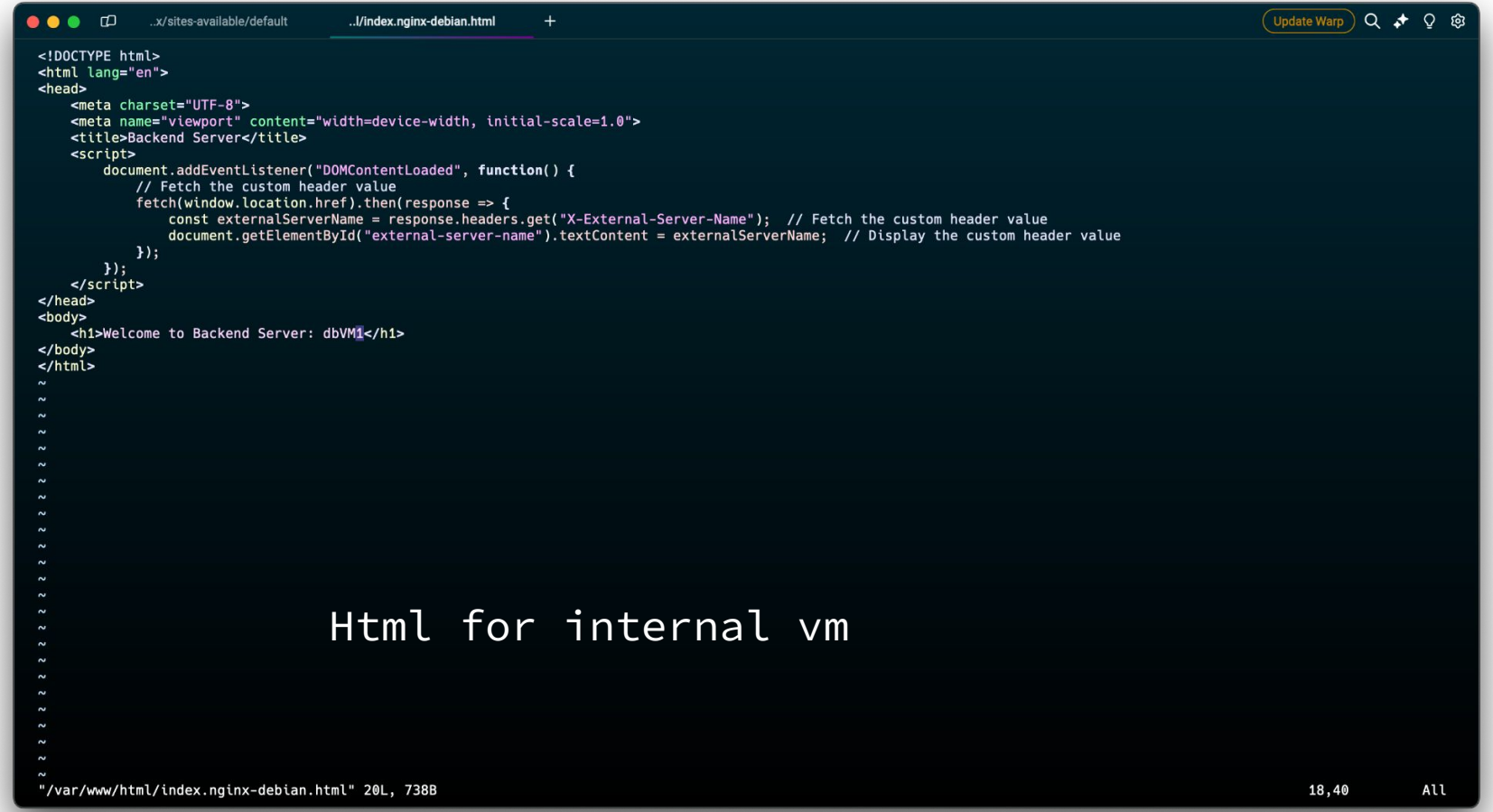
lb.sh | temp.sh

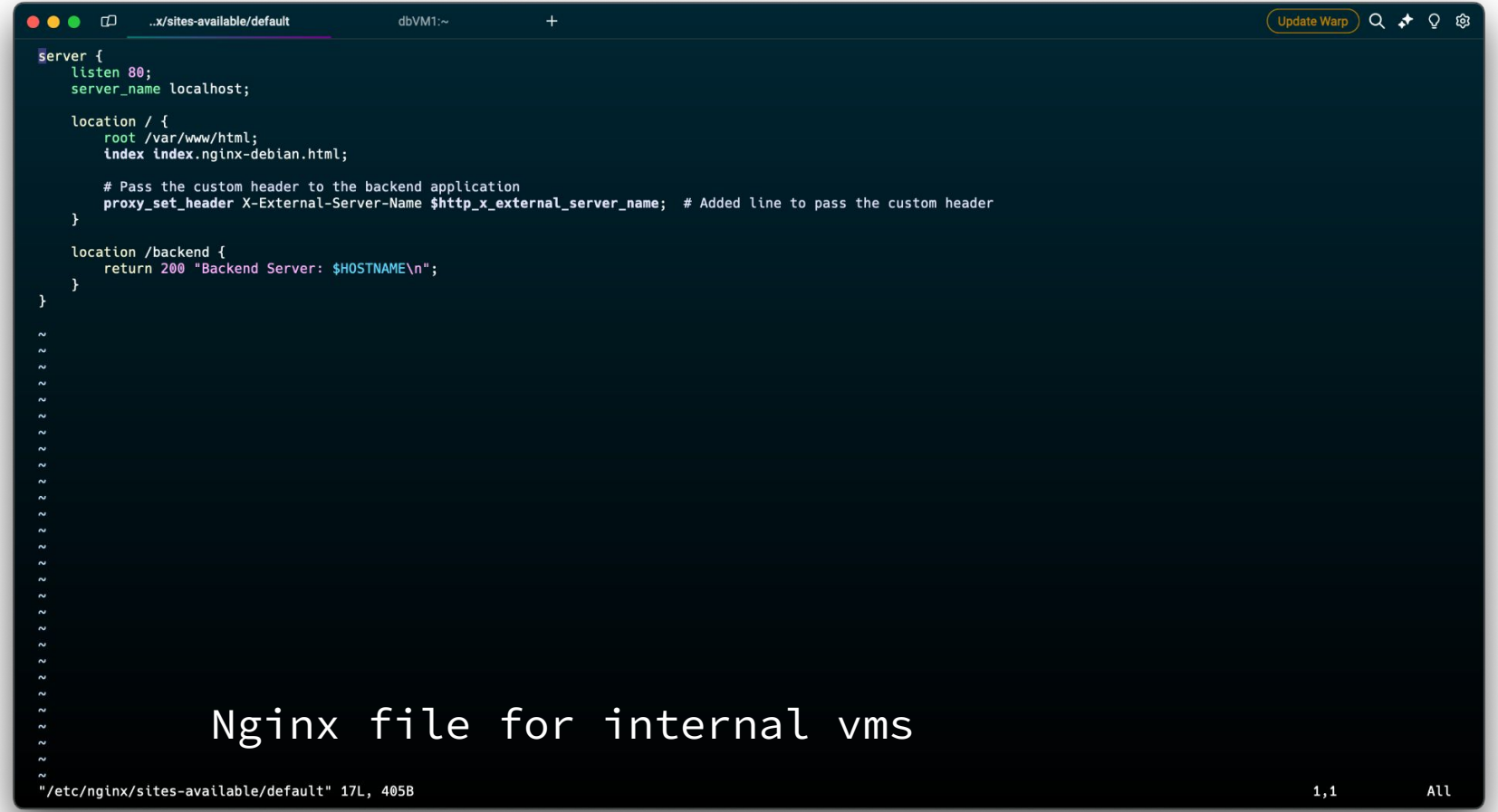
```
31  }
32
33  # Create resource group
34  create_resource "az group create \
35      --name $RESOURCE_GROUP \
36      --location $LOCATION \
37      --output table" "Resource Group"
38
39  # Create VNet and subnets
40  create_resource "az network vnet create \
41      --resource-group $RESOURCE_GROUP \
42      --name $VNET_NAME \
43      --address-prefix 10.0.0.0/16 \
44      --subnet-name $SUBNET_FRONTEND \
45      --subnet-prefix 10.0.1.0/24 \
46      --output table" "VNet and Frontend Subnet"
47
48  create_resource "az network vnet subnet create \
49      --resource-group $RESOURCE_GROUP \
50      --vnet-name $VNET_NAME \
51      --name $SUBNET_BACKEND \
52      --address-prefix 10.0.2.0/24 \
53      --output table" "Backend Subnet"
54
55  create_resource "az network vnet subnet create \
56      --resource-group $RESOURCE_GROUP \
57      --vnet-name $VNET_NAME \
58      --name $SUBNET_DB \
59      --address-prefix 10.0.3.0/24 \
60      --output table" "DB Subnet"
61
62  # Create NSG
63  create_resource "az network nsg create \
64      --resource-group $RESOURCE_GROUP \
65      --name $NSG_NAME \
66      --output table" "NSG"
67
68  # Allow HTTP traffic
```

Using the script in the github you can provision all the things needed for This two tier application

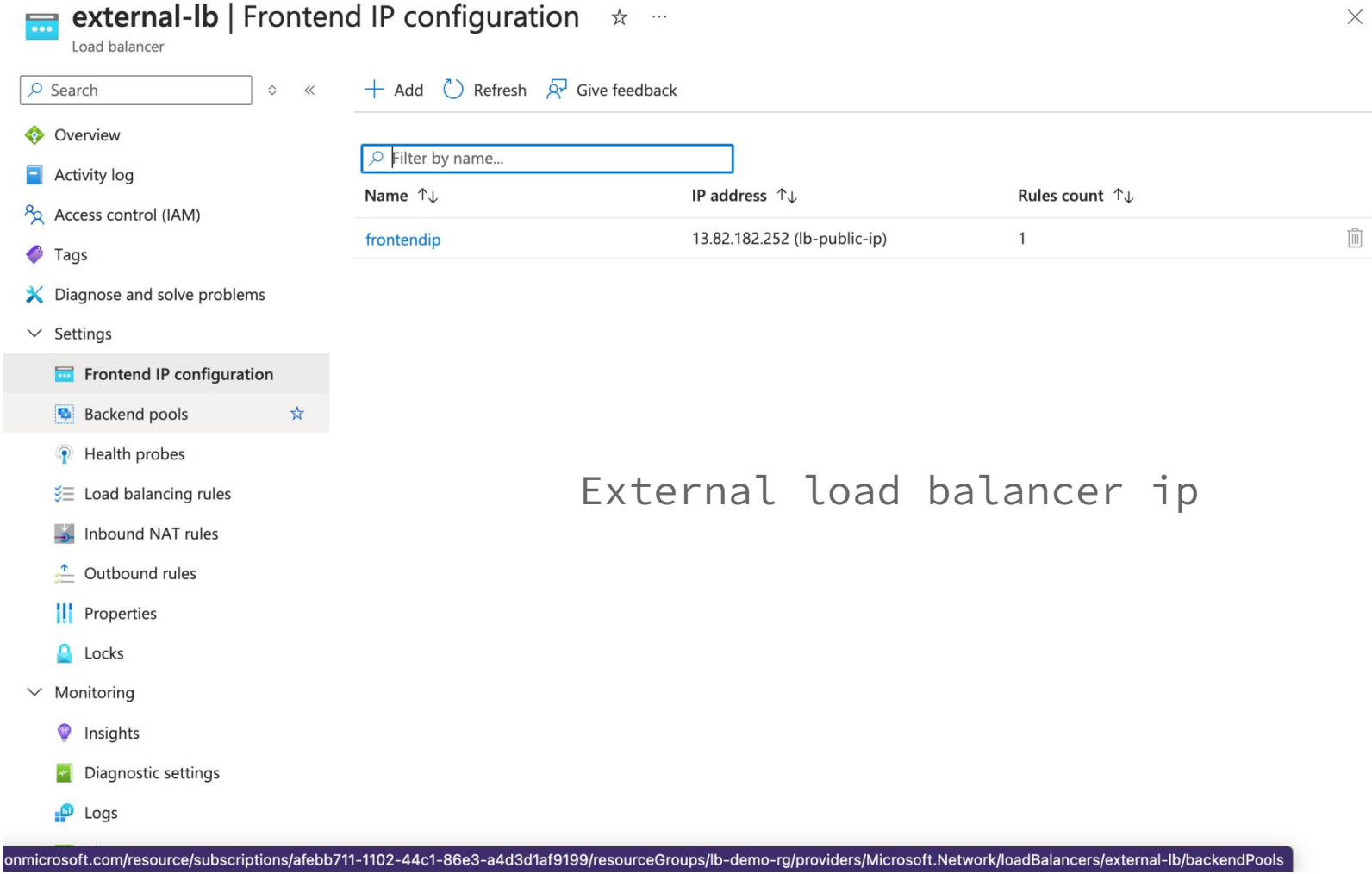
```
server {
    listen 80;
    server_name localhost;

    location / {
        add_header X-Frontend-Server "Frontend $HOSTNAME" always;
        proxy_pass http://10.0.2.4;  # Internal Load Balancer IP
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
    }

    location /frontend {
        return 200 "Frontend Server: $HOSTNAME\n";
    }
}
```

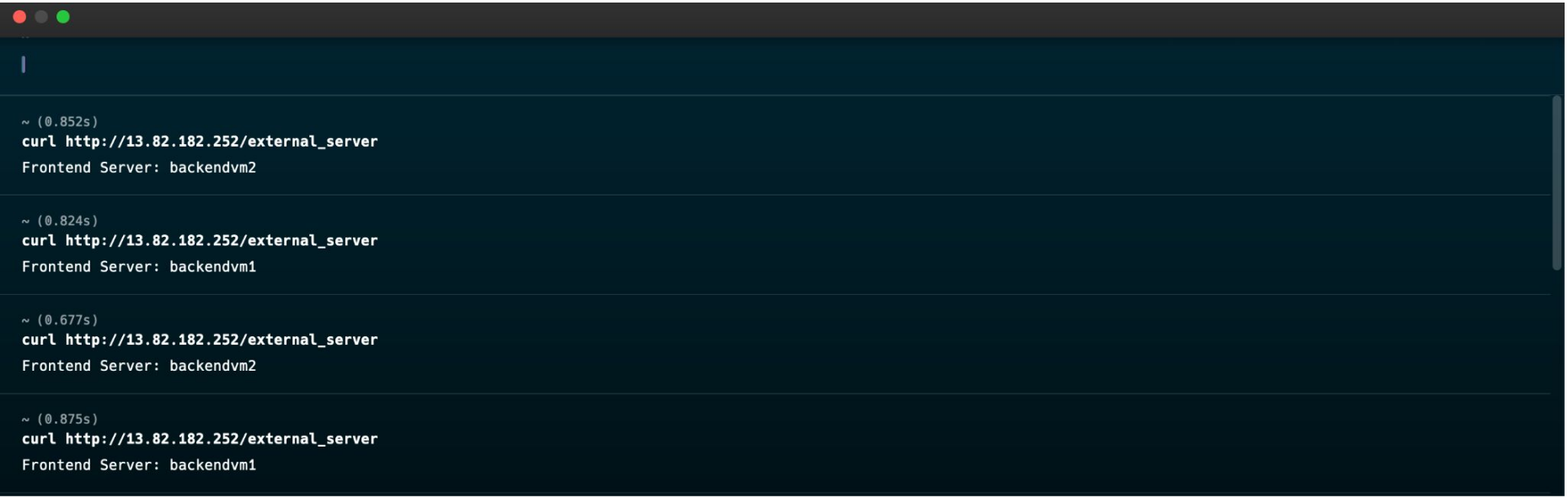
Nginx for setting up proxy at external vm


../sites-available/default
../l/index.nginx-debian.html
Update Warp
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Backend Server</title>
<script>
document.addEventListener("DOMContentLoaded", function() {
// Fetch the custom header value
fetch(window.location.href).then(response => {
const externalServerName = response.headers.get("X-External-Server-Name"); // Fetch the custom header value
document.getElementById("external-server-name").textContent = externalServerName; // Display the custom header value
});
});
</script>
</head>
<body>
<h1>Welcome to Backend Server: dbVM1</h1>
</body>
</html>
Html for internal vm
"/var/www/html/index.nginx-debian.html" 20L, 738B
18,40
All



Nginx file for internal vms


external-lb | Frontend IP configuration
Load balancer
Search
Add
Refresh
Give feedback
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Settings
Frontend IP configuration
Backend pools
Health probes
Load balancing rules
Inbound NAT rules
Outbound rules
Properties
Locks
Monitoring
Insights
Diagnostic settings
Logs
Filter by name...
Name
IP address
Rules count
frontendip
13.82.182.252 (lb-public-ip)
1
External load balancer ip
onmicrosoft.com/resource/subscriptions/afebb711-1102-44c1-86e3-a4d3d1af9199/resourceGroups/lb-demo-rg/providers/Microsoft.Network/loadBalancers/external-lb/backendPools



Accessing external vm for checking if working or not

Welcome to Backend Server: dbVM1

Welcome to Backend Server: dbVM2

Demonstration of working of both internal vm inside internal load balancer

Thank you