# Create User Defined Routes (UDR) in the Azure CLI

This article covers the Resource Manager deployment model. You can also [create UDRs in the classic deployment model](/documentation/articles/virtual-network-create-udr-classic-cli).

The sample Azure CLI commands below expect a simple environment already created based on the scenario above. If you want to run the commands as they are displayed in this document, first build the test environment by deploying [this template](http://github.com/telmosampaio/azure-templates/tree/master/IaaS-NSG-UDR-Before), click **Deploy to Azure**, replace the default parameter values if necessary, and follow the instructions in the portal.

## Create the UDR for the front end subnet

To create the route table and route needed for the front end subnet based on the scenario above, follow the steps below.

1. Run the **azure network route-table create** command to create a route table for the front end subnet.

* azure network route-table create -g TestRG -n UDR-FrontEnd -l uswest
* Output:
* info: Executing command network route-table create  
  info: Looking up route table "UDR-FrontEnd"  
  info: Creating route table "UDR-FrontEnd"  
  info: Looking up route table "UDR-FrontEnd"  
  data: Id : /subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx/resourceGroups/TestRG/providers/Microsoft.Network/  
  routeTables/UDR-FrontEnd  
  data: Name : UDR-FrontEnd  
  data: Type : Microsoft.Network/routeTables  
  data: Location : westus  
  data: Provisioning state : Succeeded  
  info: network route-table create command OK
* Parameters:
  + **-g (or –resource-group)**. Name of the resource group where the NSG will be created. For our scenario, *TestRG*.
  + **-l (or –location)**. Azure region where the new NSG will be created. For our scenario, *westus*.
  + **-n (or –name)**. Name for the new NSG. For our scenario, *NSG-FrontEnd*.

1. Run the **azure network route-table route create** command to create a route in the route table created above to send all traffic destined to the back end subnet (192.168.2.0/24) to the **FW1** VM (192.168.0.4).

* azure network route-table route create -g TestRG -r UDR-FrontEnd -n RouteToBackEnd -a 192.168.2.0/24 -y VirtualAppliance -p 192.168.0.4
* Output:
* info: Executing command network route-table route create  
  info: Looking up route "RouteToBackEnd" in route table "UDR-FrontEnd"  
  info: Creating route "RouteToBackEnd" in a route table "UDR-FrontEnd"  
  info: Looking up route "RouteToBackEnd" in route table "UDR-FrontEnd"  
  data: Id : /subscriptions/628dad04-b5d1-4f10-b3a4-dc61d88cf97c/resourceGroups/TestRG/providers/Microsoft.Network/  
  routeTables/UDR-FrontEnd/routes/RouteToBackEnd  
  data: Name : RouteToBackEnd  
  data: Provisioning state : Succeeded  
  data: Next hop type : VirtualAppliance  
  data: Next hop IP address : 192.168.0.4  
  data: Address prefix : 192.168.2.0/24  
  info: network route-table route create command OK
* Parameters:
  + **-r (or –route-table-name)**. Name of the route table where the route will be added. For our scenario, *UDR-FrontEnd*.
  + **-a (or –address-prefix)**. Address prefix for the subnet where packets are destined to. For our scenario, *192.168.2.0/24*.
  + **-y (or –next-hop-type)**. Type of object traffic will be sent to. Possible values are *VirtualAppliance*, *VirtualNetworkGateway*, *VNETLocal*, *Internet*, or *None*.
  + **-p (or –next-hop-ip-address**). IP address for next hop. For our scenario, *192.168.0.4*.

1. Run the **azure network vnet subnet set** command to associate the route table created above with the **FrontEnd** subnet.

* azure network vnet subnet set -g TestRG -e TestVNet -n FrontEnd -r UDR-FrontEnd
* Output:
* info: Executing command network vnet subnet set  
  info: Looking up the subnet "FrontEnd"  
  info: Looking up route table "UDR-FrontEnd"  
  info: Setting subnet "FrontEnd"  
  info: Looking up the subnet "FrontEnd"  
  data: Id : /subscriptions/628dad04-b5d1-4f10-b3a4-dc61d88cf97c/resourceGroups/TestRG/providers/Microsoft.Network/  
  virtualNetworks/TestVNet/subnets/FrontEnd  
  data: Type : Microsoft.Network/virtualNetworks/subnets  
  data: ProvisioningState : Succeeded  
  data: Name : FrontEnd  
  data: Address prefix : 192.168.1.0/24  
  data: Network security group : [object Object]  
  data: Route Table : /subscriptions/628dad04-b5d1-4f10-b3a4-dc61d88cf97c/resourceGroups/TestRG/providers/Microsoft.Network/  
  routeTables/UDR-FrontEnd  
  data: IP configurations:  
  data: /subscriptions/628dad04-b5d1-4f10-b3a4-dc61d88cf97c/resourceGroups/TestRG/providers/Microsoft.Network/networkInterfaces/NICWEB1/ipConf  
  igurations/ipconfig1  
  data: /subscriptions/628dad04-b5d1-4f10-b3a4-dc61d88cf97c/resourceGroups/TestRG/providers/Microsoft.Network/networkInterfaces/NICWEB2/ipConf  
  igurations/ipconfig1  
  data:   
  info: network vnet subnet set command OK
* Parameters:
  + **-e (or –vnet-name)**. Name of the VNet where the subnet is located. For our scenario, *TestVNet*.

## Create the UDR for the back end subnet

To create the route table and route needed for the back end subnet based on the scenario above, follow the steps below.

1. Run the **azure network route-table create** command to create a route table for the back end subnet.

* azure network route-table create -g TestRG -n UDR-BackEnd -l westus

1. Run the **azure network route-table route create** command to create a route in the route table created above to send all traffic destined to the front end subnet (192.168.1.0/24) to the **FW1** VM (192.168.0.4).

* azure network route-table route create -g TestRG -r UDR-BackEnd -n RouteToFrontEnd -a 192.168.1.0/24 -y VirtualAppliance -p 192.168.0.4

1. Run the **azure network vnet subnet set** command to associate the route table created above with the **BackEnd** subnet.

* azure network vnet subnet set -g TestRG -e TestVNet -n BackEnd -r UDR-BackEnd

## Enable IP forwarding on FW1

To enable IP forwarding in the NIC used by **FW1**, follow the steps below.

1. Run the **azure network nic show** command, and notice the value for **Enable IP forwarding**. It should be set to *false*.

* azure network nic show -g TestRG -n NICFW1
* Output:
* info: Executing command network nic show  
  info: Looking up the network interface "NICFW1"  
  data: Id : /subscriptions/628dad04-b5d1-4f10-b3a4-dc61d88cf97c/resourceGroups/TestRG/providers/Microsoft.Network/  
  networkInterfaces/NICFW1  
  data: Name : NICFW1  
  data: Type : Microsoft.Network/networkInterfaces  
  data: Location : westus  
  data: Provisioning state : Succeeded  
  data: MAC address : 00-0D-3A-30-95-B3  
  data: Enable IP forwarding : false  
  data: Tags : displayName=NetworkInterfaces - DMZ  
  data: Virtual machine : /subscriptions/628dad04-b5d1-4f10-b3a4-dc61d88cf97c/resourceGroups/TestRG/providers/Microsoft.Compute/  
  virtualMachines/FW1  
  data: IP configurations:  
  data: Name : ipconfig1  
  data: Provisioning state : Succeeded  
  data: Public IP address : /subscriptions/628dad04-b5d1-4f10-b3a4-dc61d88cf97c/resourceGroups/TestRG/providers/Microsoft.Network/  
  publicIPAddresses/PIPFW1  
  data: Private IP address : 192.168.0.4  
  data: Private IP Allocation Method : Static  
  data: Subnet : /subscriptions/628dad04-b5d1-4f10-b3a4-dc61d88cf97c/resourceGroups/TestRG/providers/Microsoft.Network/  
  virtualNetworks/TestVNet/subnets/DMZ  
  data:   
  info: network nic show command OK

1. Run the **azure network nic set** command to enable IP forwarding.

* azure network nic set -g TestRG -n NICFW1 -f true
* Output:
* info: Executing command network nic set  
  info: Looking up the network interface "NICFW1"  
  info: Updating network interface "NICFW1"  
  info: Looking up the network interface "NICFW1"  
  data: Id : /subscriptions/628dad04-b5d1-4f10-b3a4-dc61d88cf97c/resourceGroups/TestRG/providers/Microsoft.Network/  
  networkInterfaces/NICFW1  
  data: Name : NICFW1  
  data: Type : Microsoft.Network/networkInterfaces  
  data: Location : westus  
  data: Provisioning state : Succeeded  
  data: MAC address : 00-0D-3A-30-95-B3  
  data: Enable IP forwarding : true  
  data: Tags : displayName=NetworkInterfaces - DMZ  
  data: Virtual machine : /subscriptions/628dad04-b5d1-4f10-b3a4-dc61d88cf97c/resourceGroups/TestRG/providers/Microsoft.Compute/  
  virtualMachines/FW1  
  data: IP configurations:  
  data: Name : ipconfig1  
  data: Provisioning state : Succeeded  
  data: Public IP address : /subscriptions/628dad04-b5d1-4f10-b3a4-dc61d88cf97c/resourceGroups/TestRG/providers/Microsoft.Network/  
  publicIPAddresses/PIPFW1  
  data: Private IP address : 192.168.0.4  
  data: Private IP Allocation Method : Static  
  data: Subnet : /subscriptions/628dad04-b5d1-4f10-b3a4-dc61d88cf97c/resourceGroups/TestRG/providers/Microsoft.Network/  
  virtualNetworks/TestVNet/subnets/DMZ  
  data:   
  info: network nic set command OK
* Parameters:
  + **-f (or –enable-ip-forwarding)**. *true* or *false*.