This is configuration example for AZURE connection from VTI router / appliance.

(Source of this document - https://docs.google.com/document/d/1IY2W1q_PqblrJctCBUyx8rN8TsCQ6YKWYrB94o9swqE/edit#)
Azure specifics:

- They use isakv2, not isakv1;
- Different codecs and key life time, a lot of noise in logs if we do not specify exact codecs;
- No assigned IP on VTI interface (so we make our own and use their router IP on far end);
- No GUI support for BGP part, so part of configuration must be done by resource manager (it MAY change, but it was this way on November 2016);
- Single router on AZURE side (it makes this all less reliable); it is on virtual cluster but we can experience software level problem yet;
- IPSEC has some issues such as, in some cases, AZURE starts to re-key every few minutes. This is not resolved yet; it do not affect performance and we have a workaround to avoid it.
- No configuration download (so we use manual configuration on our end).
- BGP router != ipsec gateway, so we must use multihop BGP;
- As Azure do not understand ‘VTI tunnel IP’, we use tunnel IP as ‘local gateway’ IP and so we must create local gateway object in azure per every VTI tunnel (even if we, in reality, share VTI router between few VNET-s).

Aside of this, it all works pretty well. Azure has longer reconfiguration time, but BGP starts immediately once tunnel is open (in AWS, reconfigurations are instant, but BGP waits few minutes once tunnel is created).

Router-s preparations are THE SAME on our end, no difference. As Azure has 1 VPN gateway per vnet, we create 2 VTI appliances on our end (each with INSIDE and OUTSIDE IP) and we will create 2 VTI tunnels for redundancy (using BGP routing with AZURE).

CONFIGURATION EXAMPLE.

1. Create VNET, GatewaySubnet, and VNET gateway with ROUTE BASED policy and BGP enabled.

Step one is as usual, create VNET in AZURE, create GatewaySubnet, create Virtual Network gateway.

- Once virtual Network Gateway is created by GUI (using new portal), open it in resource manager and change it, if necessary, to allow BGP routing. It must look like this (in the end of gateway resources):

```
"gatewayType": "Vpn",
        "vpnType": "RouteBased",
        "enableBgp": true,
        "activeActive": false,
        "bgpSettings": {
        "asn": 65515,
        "bgpPeeringAddress": "10.32.1.254",
        "peerWeight": 0
        }
```

So, we created VNET gateway with AS number 65515 and internal IP 10.32.1.254, and outside IP. Write them down:

a. VNET OUTSIDE IP (you can find it by resource manager in PUBLIC IP section, or by GUI in gateway properties
b. VNET INSIDE BGP IP, we will use it as far end of VTI tunnels
c. VNET BGP AS number.

2. ASSIGN IP to our end of tunnel

If we want to be able to ping tunnel, it is important do not use our INSIDE IP (on eth0 interface) as VTI address (if we do it, we lost simple ping) but assign uniq IP onto it. I assigned 10.23.223.76.

So I have, for MY side:

- INSIDE VTI IP;
- INSIDE AS number;
- INSIDE BGP address will be the same as VTI IP.

3. Create Site-To-Site tunnel in AZURE, creating new Local Network Gateway

Use:

- Your VTI router EXTERNAL IP, as gateway IP.
- Your VTI tunnel IP as <IP>/32 in ‘Address Space’ section.
- Enter Shared key
- I use name <router-name>-vti1 as I will create 1 name per 1 VTI tunnel, not per 1 local gateway.
- OK, OK, and wait until it will be completed (few minutes).

Verify ESSENTIALS, they should look like this:

- Resource group
- Desjardins
- Data in
- 0 B
- Status
- Not connected
- Data out
- 0 B
- Location
- East US
- Virtual network
- az-vnet-dxd
- Subscription name
- Microsoft Azure Enterprise
- Virtual network gateway
- az-vnet-gw-dnd (40.11.219.26)
- Subscription ID
- 620538a2-5175-495e-af55-db32ef8bb4ee
- Local network gateway
- dev2-dxd-vti01-vti1 (209.11.73.35)

40.11.219.26 is AZURE outside IP, 209.11.73.35 is our outside IPm and IP range on local gateway is 10.23.223.76/32.

This was done by https://portal.azure.com/#resource - new AZURE portal.

4. Configure BGP routing.

All previous steps was done by https://portal.azure.com/#resource - new Azure portal.

Now we must go to resource manager - https://resources.azure.com

Find your network there, and find your local gateway, vnet azure gateway, and connection. Using WRITE mode on top and EDIT button, enable BGP and write your BGP AS number and your IP (it is the same as on VTI, 10.23.223.76 in our example) into the resources and PUSH them.

Remember, BGP must be ‘true’ in both, VNET gateway, your local gateway, and connection. For example

First, add BGP into our local gateway. Make sure that you PUT it, then make GET and verify. Example:

```
        "gatewayIpAddress": "209.11.73.35",
        "bgpSettings": {
        "asn": "65003",
        "bgpPeeringAddress": "10.23.223.76",
        "peerWeight": "0"
```

Second, verify BGP in your vnet gateway, for example:

```
  "gatewayType": "Vpn",
        "vpnType": "RouteBased",
        "enableBgp": true,
        "activeActive": false,
        "bgpSettings": {
        "asn": 65515,
        "bgpPeeringAddress": "10.32.1.254",
        "peerWeight": 0
```

Then, make sure that BGP is allowed in connection:

```
  "connectionType": "IPsec",
        "routingWeight": 0,
        "sharedKey": "xxx111yyy222zzzz3412",
        "enableBgp": true,
        "connectionStatus": "NotConnected",
        "ingressBytesTransferred": 0,
        "egressBytesTransferred": 0
```

Write it all down.

5. Configure VTI tunnel on our router.

To do it, run

```
cd /usr/local/scripts && ./VTI_ADD.sh
```

Answer questions.
N - vtiN, use default (it is first so it will be 1)
OIP_REMOTE - address of AZURE GW,
Default gateway - where to forward packets in OUTSIDE network, it is our outside gateway
IIP local - out local VTI IP, we choose it as 10.23.223.76
IIP remote - we use their BGP gateway IP to be able to ping
VTI_IIP_PREFIX - enter /32 as we want it to be host IP
BGP remote AS - AS of AZURE
BGP local AS - it is our AS, it can be actually any AS as it is configurable for each connection separately

```
Results will be in WORK/dev2-djd-vti01.d
Enter Connection name, will be added to connection name in IPSEC
Recommended format: local-gw_remote-gw
ID= []_az-vnet-dnd-dev2-djd-vti01
Enter Interface number (1 - 99)
N= [1]_
Enter MTU of this VTI
VTI_MTU= [1420]_
Enter OUTSIDE IP of tunnel, local
VTI_OIP_LOCAL= [209.11.73.35]_
Enter OUTSIDE IP of tunnel, remote (no default)
VTI_OIP_REMOTE= []_40.11.219.26
Enter Default gateway for 40.11.219.26 on local gateway
LOCAL_GW= [209.11.73.1]_
Enter Inside IP of tunnel, local, without /
VTI_IIP_LOCAL= []_10.23.223.76
Enter Inside IP of tunnel, remote, without /
VTI_IIP_REMOTE= []_10.32.1.254
Enter Netmask prefix with / (/30 means 255.255.255.252, and so on)
VTI_IIP_PREFIX= [/30]_/32
Enter Shared Secret
VTI_PSK= []_xxxsssaaaqwweeedfssaa
Enter MARK for this VTI, must be different on different VTI
VTI_MARK= [12]_
Enter Enter provider - AZURE, AWS - for pre-configured settings. Enter to use defaults
VTI_PROVIDER= []_AZURE
 Added
        VTI_O1=ike=aes256-sha1-modp1024
        VTI_O2=esp=aes128-sha1,aes256-sha1!
        VTI_O3=keyexchange=ike
        VTI_O4=ikelifetime=10800s
        VTI_O5=keylife=3600s
        VTI_O6=keyingtries=%forever
        VTI_BGP1=route-map from_azure in
        VTI_BGP2=ebgp-multihop

        You can change them later or ADD extra options.
You can add up to 5 connection options in format key=value . No syntax check here.
Enter Option 1. Enter # to skip
VTI_O1= [ike=aes256-sha1-modp1024]_
Enter Option 2. Enter # to skip
VTI_O2= [esp=aes128-sha1,aes256-sha1!]_
Enter Option 3. Enter # to skip
VTI_O3= [keyexchange=ike]_
Enter Option 4. Enter # to skip
VTI_O4= [ikelifetime=10800s]_
```

```
Enter Option 5. Enter # to skip
VTI_O5= [keylife=3600s]_
Enter Option 6. Enter # to skip
VTI_O6= [keyingtries=%forever]_
Now enter BGP information if we use it
Enter Enter AS of remote neighbor (ENTER if no BGP)
VTI_BGP_REMOTE_AS= []_65515
Enter Enter AS of our system
VTI_BGP_LOCAL_AS= []_65003
Enter Enter IP of neighbor
VTI_BGP_REMOTE_IP= [10.32.1.254]_
Enter Enter local IP for BGP
VTI_BGP_LOCAL_IP= [10.23.223.76]_
You can add up to 5 bgp options  here (they added with neighbor 10.32.1.254). No syntax
check here.
Enter neighbor 10.32.1.254 Option 1. Enter # to skip
VTI_BGP1= [route-map from_azure in]_
Enter neighbor 10.32.1.254 Option 2. Enter # to skip
VTI_BGP2= [ebgp-multihop]_
Enter neighbor 10.32.1.254 Option 3. Enter # to skip
VTI_BGP3= []_
Enter neighbor 10.32.1.254 Option 4. Enter # to skip
VTI_BGP4= []_
Enter neighbor 10.32.1.254 Option 5. Enter # to skip
VTI_BGP5= []_
We adjusted addresses as VTI_IIP_LOCAL=10.23.223.76/32 and VTI_IIP_REMOTE=10.32.1.254/32
Created WORK/dev2-djd-vti01.d/vti1.properties
### MANUALLY CREATED Mon Nov 14 20:38:13 PST 2016
ID="az-vnet-dnd-dev2-djd-vti01"
VTI="vti1"
VTI_MTU="1420"
VTI_OIP_LOCAL="209.44.73.35"
VTI_OIP_REMOTE="40.76.219.26"
LOCAL_GW="209.44.73.1"
VTI_IIP_LOCAL="10.23.223.76/32"
VTI_IIP_REMOTE="10.32.1.254/32"
VTI_IIP_PREFIX="/32"
VTI_PSK="AZURE103200eis102317"
VTI_MARK="12"
VTI_PROVIDER="AZURE"
VTI_O1="ike=aes256-sha1-modp1024"
VTI_O2="esp=aes128-sha1,aes256-sha1!"
VTI_O3="keyexchange=ike"
VTI_O4="ikelifetime=10800s"
VTI_O5="keylife=3600s"
VTI_O6="keyingtries=%forever"
VTI_BGP_REMOTE_AS="65515"
VTI_BGP_LOCAL_AS="65003"
VTI_BGP_REMOTE_IP="10.32.1.254"
VTI_BGP_LOCAL_IP="10.23.223.76"
VTI_BGP1="route-map from_azure in"
VTI_BGP2="ebgp-multihop"
VTI_BGP3=""
VTI_BGP4=""
VTI_BGP5=""
```

6. Review and edit property file

VTI -list

vi WORK/`hostname -s`.d/vti1.properties

(You can edit properties once they are created. Do not change MARK and vti name).

For example, I edit bgp properties as they are slightly different in our network (we use different route map names).

7. Add interface and check router configuration.

You can now see vti1 as available when you run VTI -list. Add it into the system
VTI -add vti1

Review results:

- vtysh - allows to see router configuration. If in doubts save config and check files in /etc/quagga manually. For example, I verify that we have now in BGP configuration lines

```
router bgp 65003
 bgp router-id 10.23.23.4
 network 10.0.0.0/8
 network 192.168.0.0/16
 neighbor 10.32.1.254 remote-as 65515
 neighbor 10.32.1.254 ebgp-multihop 255
 neighbor 10.32.1.254 route-map from_aws in
```

-
- cat /etc/strongswan/vti.d/vti1.init
- cat /etc/strongswan/vti.d/vti1.secrets
- cat /etc/strongswan/vti.d/.DISABLED/vti1.conf

If satisfied, enable vti1

- VTI -enable vti1

```
VTI -enable vti1
vti1 enabled
Reloading strongSwan IPsec configuration...
establishing CHILD_SA vti1.az-vnet-dnd-dev2-djd-vti01
generating CREATE_CHILD_SA request 2 [ SA No TSi TSr ]
sending packet: from 209.44.73.35[4500] to 40.76.219.26[4500] (316 bytes)
received packet: from 40.76.219.26[4500] to 209.44.73.35[4500] (300 bytes)
parsed CREATE_CHILD_SA response 2 [ SA No TSi TSr ]
CHILD_SA vti1.az-vnet-dnd-dev2-djd-vti01{2} established with SPIs c49aa017_i 643a7e4a_o and
TS 0.0.0.0/0 ::/0 === 0.0.0.0/0 ::/0
connection 'vti1.az-vnet-dnd-dev2-djd-vti01' established successfully
Reloading strongSwan IPsec configuration...
```

Verify connection now. You must see INSTALLED status, not only ESTABLISHED.
VTI -status | grep vti1

```
VTI -status
vti1.az-vnet-dnd-dev2-djd-vti01: #5, ESTABLISHED, IKEv2, 8acbf080414aade6:f7e18157608f8fc0
  local  '209.44.73.35' @ 209.44.73.35[4500]
  remote '40.76.219.26' @ 40.76.219.26[4500]
  AES_CBC-256/HMAC_SHA1_96/PRF_HMAC_SHA1/MODP_1024
  established 83s ago, rekeying in 10045s
  vti1.az-vnet-dnd-dev2-djd-vti01: #1, reqid 1, INSTALLED, TUNNEL-in-UDP,
ESP:AES_CBC-256/HMAC_SHA1_96
        installed 83s ago, rekeying in 2718s, expires in 3517s
        in  ccf8e3f3,  618 bytes,     9 packets,     22s ago
        out 5aa8a7b4,  422 bytes,     6 packets,     22s ago
        local  0.0.0.0/0 ::/0
        remote 0.0.0.0/0 ::/0
  vti1.az-vnet-dnd-dev2-djd-vti01: #2, reqid 1, INSTALLED, TUNNEL-in-UDP,
ESP:AES_CBC-256/HMAC_SHA1_96
        installed 78s ago, rekeying in 2869s, expires in 3522s
        in  c49aa017,   3660 bytes,   45 packets,    22s ago
        out 643a7e4a,  40 bytes,      1 packets,     22s ago
        local  0.0.0.0/0 ::/0
        remote 0.0.0.0/0 ::/0
  vti1.az-vnet-dnd-dev2-djd-vti01: #3, reqid 1, INSTALLED, TUNNEL-in-UDP,
ESP:AES_CBC-256/HMAC_SHA1_96
        installed 74s ago, rekeying in 2764s, expires in 3526s
        in  c7fd061a,   3641 bytes,   48 packets,    22s ago
        out e1e092e0,  99 bytes,      2 packets,     22s ago
        local  0.0.0.0/0 ::/0
        remote 0.0.0.0/0 ::/0
vti1: flags=209<UP,POINTOPOINT,RUNNING,NOARP>  mtu 1420
10.32.0.0      10.32.1.254    255.255.254.0   UG            0 0            0 vti1
10.32.1.254    0.0.0.0        255.255.255.255 UH            0 0            0 vti1
```

Make sure that you propagate routing as required. if your VTI router is inside DMZ (as it is in our case), enable asymmetric TCP bypass if you need connections from inside the same DMZ and your outbound gateway is your ASA router. here is simple example, but you must read all details on Cisco sites:

```
access-list dmzXX_bypass extended permit tcp object-group DMZXX object-group
AZURE_XX
access-list acl_dmzXX extended permit ip object-group DMZXX object-group AZURE_XX

class-map dmzXX_bypass
 match access-list dmzXX_bypass
policy-map dmzxx_bypass_policy
 class dmzXX_bypass
  set connection advanced-options tcp-state-bypass
 class dcerpc
  inspect dcerpc
!
no service-policy interface_dcerpc interface dmzXX
service-policy dmz17_bypass_policy interface dmzXX
```

Alternative is to use your VTI router as default gateway and configure VRRP between pair of such routers for reliability. Or keep different DMZ for VTI routers. Or run OSPF between all members. Just do not forget about such problem - if your host is in DMZ, your VTI router to AZURE is in DMZ, and host has only default routing to firewall, then traffic from host to azure go via firewall and traffic from azure to host go directly from vti router, so firewall must understand one way pass and allow interface back to the same zone traffic. (It is outside of our scope here).