# Setting a personal RPAD for MSTeamGW development.

1. Create a bot under your resource group as documented before. In this example the group is called jkskypebot.
2. Add a classic vNet wrapping the bot: Using the Azure portal go to: “Virtual networks (classic)” under more service. Click the + and fill as follows:

Name: Anything you want. For convention use YourNameBot-vnclassic

Address space: 10.200.0.0/16 this space falls nicely with Aqua addresses convention.

Subnet name: sub1

Subnet address range: 10.200.0.0/24

Subscription: AQ-Beater-1

Resource Group: Use existing and select the same resource group of your bot (Cloud service).

Location: Same as your bot (Cloud service).

1. Open your bot solution in visual studio go to ServiceConfiguration.Cloud.cscfg and set the Network Configuration as follows (Notice the PublicIPs section was removed):

<NetworkConfiguration>

<!—Group and Name of the target classic Virtual Network -->

<VirtualNetworkSite name="Group NameOfTheClassicvNetResourceGroup NameOfTheClassicvNet" />

<AddressAssignments>

<InstanceAddress roleName="WorkerRole">

<Subnets>

<Subnet name="sub1" />

</Subnets>

</InstanceAddress>

</AddressAssignments>

</NetworkConfiguration>

Make the same changes to ServiceConfiguration.Local.cscfg.

1. Open the file AzureConfiguration.cs under the worker role go to the GetInstancePublicIpAddress (Around line 338). Change the function to be:

private static IPAddress GetInstancePublicIpAddress(string publicFqdn)

    {

        int instanceNumber;

//get the instanceId for the current instance. It will be of the //form  XXMediaBotRole\_IN\_0. Look for IN\_ and then extract the number after it

        //Assumption: in\_<instanceNumber> will the be the last in the instanceId

        string instanceId = RoleEnvironment.CurrentRoleInstance.Id;

        int instanceIdIndex = instanceId.IndexOf(InstanceIdToken, StringComparison.OrdinalIgnoreCase);

if (!Int32.TryParse(instanceId.Substring(instanceIdIndex + InstanceIdToken.Length), out instanceNumber))

        {

Log.Error(new CallerInfo(), LogContext.FrontEnd, "Couldn't extract Instance index from {0}", instanceId);

                throw new Exception("Couldn't extract Instance index from " + instanceId);

        }

//for example: instance0 for fooservice.cloudapp.net will have hostname as //pip.0.fooservice.cloudapp.net

       string instanceHostName = DomainNameLabel + "." + instanceNumber + "." + publicFqdn;

       IPAddress[] instanceAddresses = null;

       try

       {

            instanceAddresses = Dns.GetHostEntry(instanceHostName).AddressList;

       }

       catch (System.Net.Sockets.SocketException e)

       {

             Log.Error(new CallerInfo(), LogContext.FrontEnd, "Failed to resolve host");

       }

       if (instanceAddresses == null || instanceAddresses.Length == 0)

       {

            return IPAddress.Parse("40.1.1.1");

       }

         return instanceAddresses[0];

      }

      #endregion Helper methods

    }

The main change there is a try catch over the DNS lookup, because it will fail. With the new network configuration, the bot will not have a public IP. To make this code work it is just hard code it to 40.1.1.1. The bot will still work as before even without a public IP.

**Rebuild the solution.**

1. Select Publish the solution. In your Publish pop up go to Settings->AdvancedSetting, see that the “Deployment update” is checked, click the setting… next it, and set popup as shown:



And click Publish to publish.

1. Create a Resource vNet:

On the Azure portal go to “Virtual Networks” and click the “+”

Name: Anything you want. For convention use YourNameBot-vnresource

Address Space: 10.255.0.0/29

Subscription: AQ-Beater-1

Resource Group: Use existing and select the same resource group of your bot (Cloud service).

Location: Same as your bot (Cloud service).

Subnet name: sub1

Address range: 10.255.0.0/29

Service endpoints(Preview): Disabled

1. Peer the networks: Go to the resource vNet just created: Select Peering and click +

Name: Anything you want. For convention use YourNameBot-resourceToClassic

Virtual Network deployment model : classic

Check “I know my Resource my ID”.

Set the resource ID to classic vNet resource id, go to the classic vNet and you find it under properties.

Configuration: Enabled

Allow forwarded traffic check. (The other uncheck)

1. Optional, copy a vhd to the region needed: The vhd is already copied in 3 places for south central us under gatewayskypebotscus, east us under gatewayskypebot and east Asia under gatewayskypeboteasia. In case you need it in another place or need a different version here is the copy command used to copy it to east asia:

IMAGE=plcm-rpp-rpad-4.2.5-241367-fcd5f998-azure.vhd;

az storage blob copy start -c vhds --account-name gatewayskypeboteasia --source-account-name aqppi1scus --source-container vhds --source-blob $IMAGE -b $IMAGE --source-account-key 1MS5lfaYqklid3WWfOI8RpkBWDI6lGggGFPtkO4gbGxRAh7UzEF5XH5ue71PGV6Rh9oqw7iqJhhoDxoYF/eg0g==

If you get an authorization error. Make sure the key is right, copy it again from the Azure portal under the container account aqppi1scus.

1. Create an rpad disk from the vhd under yourResourceGroup (in this example the vhd is in gatewayskypeboteasia):

az disk create --resource-group yourResourceGroup --name managedDisk-rpad-yourbot --source <https://gatewayskypeboteasia.blob.core.windows.net/vhds/plcm-rpp-rpad-4.2.5-241367-fcd5f998-azure.vhd>

1. Create a vm from the disk:

az vm create --resource-group yourResourceGroup --location eastasia --name vm-rpad-yourbot --os-type linux --attach-os-disk managedDisk-rpad-yourbot

1. Fix the hosts file of the vm.

Look at the details of the vm the azure portal and fint its ip. Ssh to the vm using the command ssh rpad@vmIp, type in the password in the email.

Then su to it (Type su at the prompt) and enther the other password in the email.

vi /etc/hosts

Change the ip of the local host to point to 127.0.0.1, so it looks like this:

127.0.0.1 localhost.localdomain localhost

And the change the private ip to point to yourNameInitialraad for example:

10.255.0.4 rbrpad rbrpad

Reboot the vm you can do it through the portal.

1. Add access to the vm:

On the azure portal under your resource go to the NSG for example: vm-rpad-rbbotNSG

Go to inbound security rules and one for the UI:

Name :UI Port:8443 Protocol: tcp, Priority:1001 the rest can stay no change.

And add one more for SIP

Name: sip Port:5060 Protocol any, Priority:1002 the rest can stay no change.

1. Point the rpad next hope to the worker role (of the bot) private ip:

Go to the aqua portal and look for the bot worker role private ip.

Go the rpad ui: Go to <https://rpadPuplicIP:8443/edge>. Log in: admin\Polycom12#$

Go to configuration->sip change the “SIP proxy (Next hop) address:” change it to the worker role private ip (you just found). And the port to 5060.

1. Test it by calling fron a regular end point over SIP to the rpad public ip.

Go to the call history on the rpad UI and see that the call was forwarded to the bot private IP.