# Prerequisites for Gradle Build

## On-premises TFS Server

* + TFS 2013
  + TFS Build 2013
  + Code should be checked into GIT Source Control

## VS Online / TFS Service

* + Hosted Build Controller cannot be used. Hence configure your on-premises Build Server to connect to VSOnline.
  + Code should be checked into GIT Source Control

## Client Machine

* + Team Explorer 2013

## Server hosting Team Foundation Build Agent

* + Java to be installed.
    - JDK version 🡪 1.5 or higher
    - JAVA\_HOME environment variable to be set
  + Gradle version 🡪 1.9 or higher
  + For ‘gradle’ command:
    - Gradle to be installed
    - GRADLE\_HOME environment variable to be set
  + For ‘gradlew’ command:
    - Needs Internet connection, esp. for first time run – Gradle’s characteristic

# Prerequisites for Uploading Build Binaries to Azure Storage

## Server hosting Team Foundation Build Agent

* + AzCopy to be installed.
    - AZCOPY environment variable to be set to the path where AzCopy.exe is available
    - AZCOPY 2.0.0.0 or higher

## Azure Storage

* + Storage Account should be created
  + Access Key to be provided to Build Template
  + Container should be created

**NOTE:** If any of the above mentioned Environment variables are set newly, then the TFS Build Service should be restarted. Else TFS Build cannot resolve the Environment variables

# Prerequisites for Deploying to Azure VMs

## Server hosting Team Foundation Build Agent

* + **For Linux VM:**
    - OpenSSH or any tool that provides an SSH client for Windows
    - Path of ssh.exe to be added to environment variable %PATH% on Build Agent.
    - SSH private key to be saved on the File system.
      * If there are multiple Build Agents configured, you can choose to store the private key in a shared drive to which all Build Agents have access to.
      * The full path to this file should be provided to the Build Definition.
      * See <http://www.windowsazure.com/en-us/manage/linux/how-to-guides/ssh-into-linux/> to know how to create SSH private key file.
  + **For Windows VM:**
    - Windows Azure PowerShell
    - WinRM Config for Client as follows:
      * winrm quickconfig
      * winrm set winrm/config/client '@{AllowUnencrypted="true"}' # Not Required for HTTPS connection
      * winrm set winrm/config/client/auth '@{Basic="true"}'
      * winrm set winrm/config/client '@{TrustedHosts*="<DNS of VMs comma separated>*"}'
    - Download the SSL Certificate of the Azure Cloud Service hosting the Windows VMs and store it on the File System
      * If there are multiple Build Agents configured, you can choose to store the Certificate in a shared drive to which all Build Agents have access to.
      * The full path to this file should be provided to the Build Definition.
      * Use the below Azure PowerShell script to download the certificate:

$WinRMCert = (Get-AzureVM -ServiceName $CloudServiceName -Name $vmName | select -ExpandProperty vm).DefaultWinRMCertificateThumbprint

$AzureX509cert = Get-AzureCertificate -ServiceName $CloudServiceName -Thumbprint $WinRMCert -ThumbprintAlgorithm sha1

$certFile = $certFilePath

$AzureX509cert.Data | Out-File $certFile

* + The BuildProcess.dll should be checked in to TFS Version Control and provided to the Build Controller as the path to find assemblies. Or it should be added to the GAC

## Azure VMs

* + All the Azure VMs required for this Gradle Build & Continuous Deploy, should be provisioned under the same Azure Cloud Service
  + All the Azure VMs provisioned under this Cloud Service should have the same username

## Azure Linux VM

* + SSH server to be running on the Linux VM
  + SSH Endpoint private port should be the default SSH port i.e. port 22.
  + Standalone SSH endpoint needs to be created for every VM, which is not behind the load balancer. You can create it on the Azure Management portal / via Azure PowerShell cmdlet.
  + If the VM is behind a load balancer, then Standalone SSH endpoints need to be created for every VM. Each VM will have a different public port. You can create it on the Windows Azure Management portal or from the Windows Azure PowerShell CLI. For details on how to use the Azure PowerShell CLI, see the ‘Client Machine’ section.
  + The VM should use SSH Key for Authentication. See <http://www.windowsazure.com/en-us/manage/linux/how-to-guides/ssh-into-linux/> to know how to use SSH Keys for authentication while the VM is being provisioned.
  + In case your VM is already provisioned and uses Username/password authentication, then generate the SSH Keys as mentioned in the link above. Then add the data in myCert.pem to /home/<username>/.ssh/authorized\_keys file.

## Azure Windows VM

* + WinRM HTTPS endpoint private port should be the default WinRM HTTPS port i.e., port 5986
  + Standalone WinRM HTTPS endpoint needs to be created for every VM, which is not behind the load balancer. The private port of this endpoint should be the same as the WinRM HTTPS listener port.
  + If the VMs are behind a load balancer, then Standalone WinRM HTTPS endpoints need to be created for every VM. Each VM will have a different public port. You can create it on the Windows Azure Management portal or from the Windows Azure PowerShell CLI. For details on how to use the Azure PowerShell CLI, see the ‘Client Machine’ section.
  + WinRM Config as follows (The following should be executed on Windows Azure Elevated PowerShell / Command Prompt window on the VM. You can do it via RDP):
    - winrm quickconfig
    - winrm set winrm/config/service '@{AllowUnencrypted="true"}' # Not Required for HTTPS connection
    - winrm set winrm/config/service/auth '@{Basic="true"}'
    - winrm set winrm/config/service/auth '@{CredSSP="True"}' #Not Required
    - winrm Certificate, HTTPS
  + Change the Remote Management Exception on Windows Firewall for Remote IPs. You can do this from Advanced Windows Firewall snap-in.

## Client Machine

* + Windows Azure PowerShell (In case Azure endpoints will be created via Windows Azure PowerShell CLI). Alternately, you can also use the Management Portal / create endpoints at the time of VM creation.
    - **For Linux VM:**

If the VMs are behind load balancers, then create a new endpoint with a new public port, which maps to the load-balanced endpoint’s SSH private port. Ex:

Get-Azurevm -ServiceName $serviceName -Name $hostName

| Add-AzureEndpoint -Name "directSsh" -Protocol tcp

-LocalPort 22

-PublicPort $newPublicPort

| Update-AzureVM

* + - **For Windows VM:**

If the VMs are behind load balancers, then create a new endpoint with a new public port, which maps to the load-balanced endpoint’s WinRM HTTPS private port. Ex:

Get-Azurevm -ServiceName $serviceName -Name $hostName

| Add-AzureEndpoint -Name "winRMHTTPS" -Protocol tcp

-LocalPort 5986

-PublicPort $newPublicPort

 | Update-AzureVM