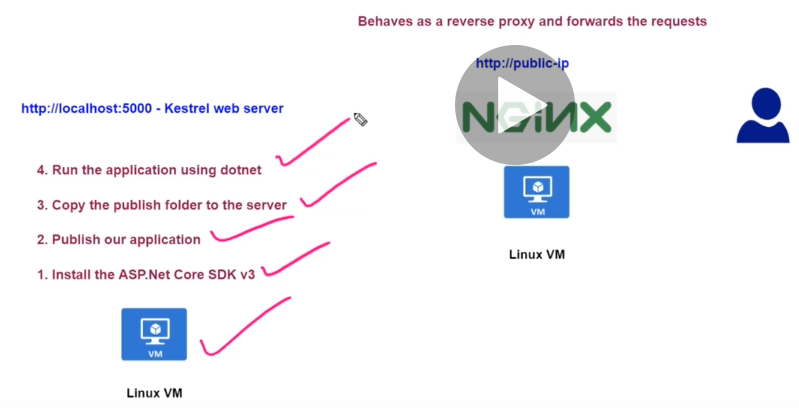
**Lesson16 Deploy Asp.net core application to Linux Server through Docker image**

**Notes: -**



**We will create .net core application that can work on any O.S that will be hosted using NGINX (Apache server) that works as proxy server with the kestrel web server on Linux**

**(we ensure that we install .net core SDK v3)**

**Steps: -**

**1-we create VM on ubuntu as done in previous lesson and install docker and upload nginx web server as reverse web server between the kestrel web server and the interner**

**2-we have to make sure that in .net core application the forward middleware is applied**

**//to apply the forward middleware**

**app.UseForwardedHeaders(new ForwardedHeadersOptions{**

**ForwardedHeaders = ForwardedHeaders.XForwardedFor | ForwardedHeaders.XForwardedProto});**

**3-on VS 2019 command promput we type the following commands**

**cd C:\Users\m.enbeh\Desktop\coreapp - 1\coreproj**

**dotnet publish //it will publish the project**

**4-by using PUTTY tool we connect to ubuntu VM and apply the following commands**

**sudo apt-get install nginx //to install nginx if not installed**

**//to install the .net core SDK on ubuntu server apply the following commands**

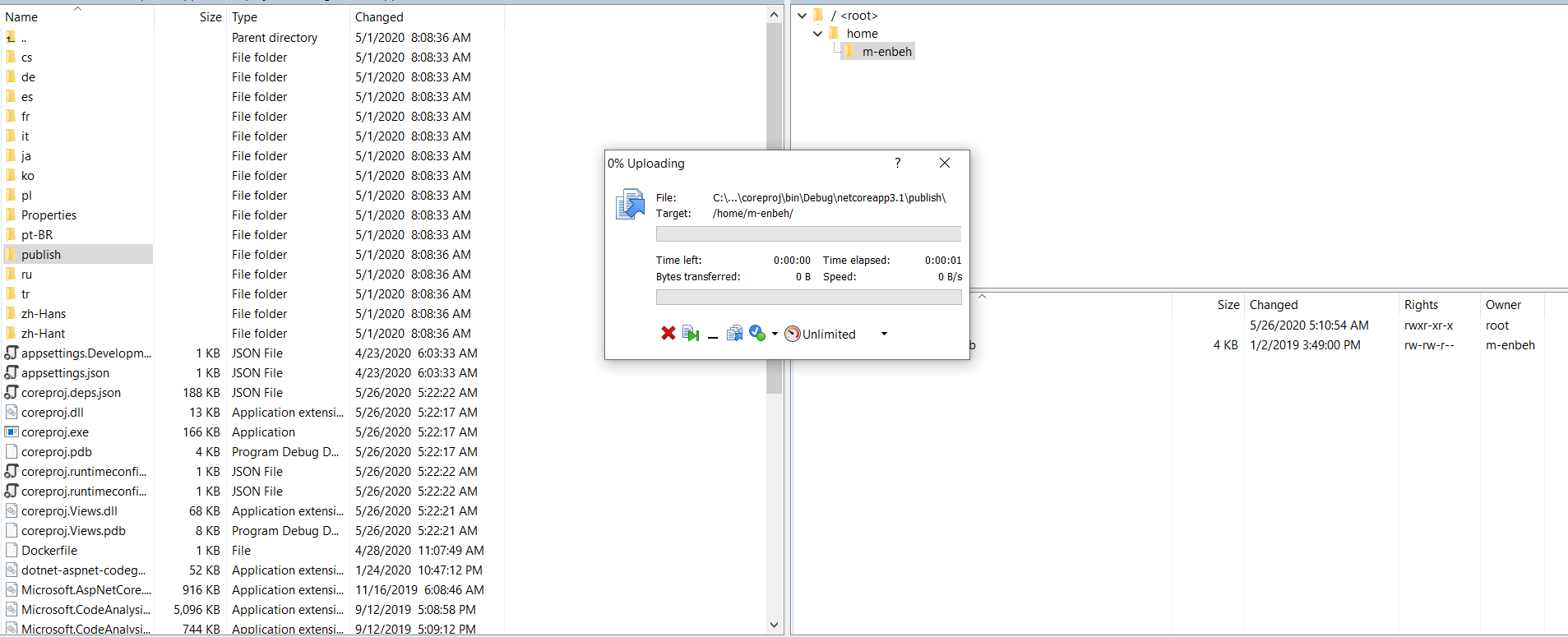
**wget https://packages.microsoft.com/config/ubuntu/18.04/packages-microsoft-prod.deb -O packages-microsoft-prod.deb**

**sudo dpkg -i packages-microsoft-prod.deb**

**sudo apt-get update**

**sudo apt-get install dotnet-sdk-3.1 //to install .net core version 3.1**

**5-we use WinScp to connect the ubuntu VM through SFTP with set username and password m-enbeh , Mohammed1993**



**6-on Putty tool type the following commands**

**Dir**

**Cd publish**

**dotnet coreproj.dll //it will be running your app on the kestrel web server**

**(we can with docker file running your app in kestrel web server in lesson 17)**

**7-we have to make change on the etc/nginx/sites-avaiable/default file but we have to make allow permission edit the file**

**(on the Putty right click and select duplicate batch window)**

**cd /etc/nginx/sites-available**

**sudo chmod 667 default //to allow update permission for default file**

**please update the location section to apply reverse server forwarding as below**

**##**

**# You should look at the following URL's in order to grasp a solid understanding**

**# of Nginx configuration files in order to fully unleash the power of Nginx.**

**# http://wiki.nginx.org/Pitfalls**

**# http://wiki.nginx.org/QuickStart**

**# http://wiki.nginx.org/Configuration**

**#**

**# Generally, you will want to move this file somewhere, and start with a clean**

**# file but keep this around for reference. Or just disable in sites-enabled.**

**#**

**# Please see /usr/share/doc/nginx-doc/examples/ for more detailed examples.**

**##**

**# Default server configuration**

**#**

**server {**

**listen 80 default\_server;**

**listen [::]:80 default\_server;**

**# SSL configuration**

**#**

**# listen 443 ssl default\_server;**

**# listen [::]:443 ssl default\_server;**

**#**

**# Note: You should disable gzip for SSL traffic.**

**# See: https://bugs.debian.org/773332**

**#**

**# Read up on ssl\_ciphers to ensure a secure configuration.**

**# See: https://bugs.debian.org/765782**

**#**

**# Self signed certs generated by the ssl-cert package**

**# Don't use them in a production server!**

**#**

**# include snippets/snakeoil.conf;**

**root /var/www/html;**

**# Add index.php to the list if you are using PHP**

**index index.html index.htm index.nginx-debian.html;**

**server\_name \_;**

**//Part Section start**

**location / {**

**proxy\_pass http://localhost:5000;**

**proxy\_http\_version 1.1;**

**proxy\_set\_header Upgrade $http\_upgrade;**

**proxy\_set\_header Connection keep-alive;**

**proxy\_set\_header Host $host;**

**proxy\_cache\_bypass $http\_upgrade;**

**}**

**//Part Section End**

**# pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000**

**#**

**#location ~ \.php$ {**

**# include snippets/fastcgi-php.conf;**

**#**

**# # With php7.0-cgi alone:**

**# fastcgi\_pass 127.0.0.1:9000;**

**# # With php7.0-fpm:**

**# fastcgi\_pass unix:/run/php/php7.0-fpm.sock;**

**#}**

**# deny access to .htaccess files, if Apache's document root**

**# concurs with nginx's one**

**#**

**#location ~ /\.ht {**

**# deny all;**

**#}**

**}**

**# Virtual Host configuration for example.com**

**#**

**# You can move that to a different file under sites-available/ and symlink that**

**# to sites-enabled/ to enable it.**

**#**

**#server {**

**# listen 80;**

**# listen [::]:80;**

**#**

**# server\_name example.com;**

**#**

**# root /var/www/example.com;**

**# index index.html;**

**#**

**# location / {**

**# try\_files $uri $uri/ =404;**

**# }**

**#}**

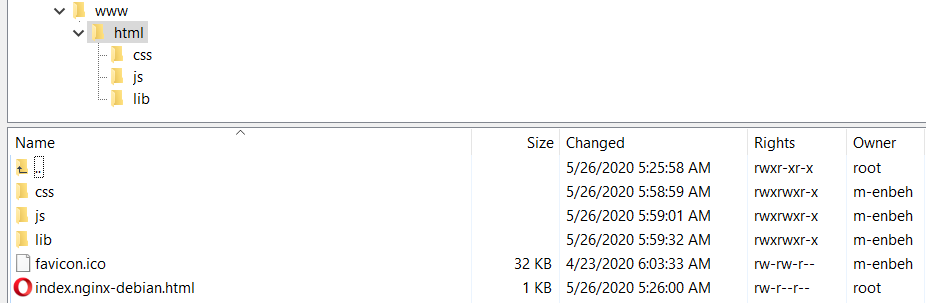
**8-on the Putty run the following commands on batch**

**cd /var/www**

**sudo chmod 667 html //to allow update permission on folder html**

**(then copy and paste the files inside the below path using winscp)**





**9-run the following command to refresh the nginx web server to respond the updates**

**sudo nginx -s reload**

**10-run on the browser**

