Contents

[Installing Ubuntu 18.04 1](#_Toc59208354)

[Install the .Net core module in our Linux environment 1](#_Toc59208355)

[Configure Nginx 1](#_Toc59208356)

[Create the service file 2](#_Toc59208357)

[Installing MySQL 3](#_Toc59208358)

[Deployment Process 4](#_Toc59208359)

[Step 1 - Publish Web API Project 4](#_Toc59208360)

[Step 2 - Delete all at deploy folder Linux Server 4](#_Toc59208361)

[Step 3 - Upload published output to Linux Server 4](#_Toc59208362)

[Step 4 - Stop kestrel web application 4](#_Toc59208363)

[Step 5 - Delete deployed contents 4](#_Toc59208364)

[Step 6 - Copy latest contents 5](#_Toc59208365)

[WEB 5](#_Toc59208366)

## Installing Ubuntu 18.04

[Create a Linux virtual machine in the Azure portal](https://docs.microsoft.com/en-us/azure/virtual-machines/linux/quick-create-portal)

[Initial Server Setup with Ubuntu 18.04 | DigitalOcean](https://www.digitalocean.com/community/tutorials/initial-server-setup-with-ubuntu-18-04)

[Install certificate for https using lets encrypt](https://letsencrypt.org/docs/)

## Install the .Net core module in our Linux environment

For that run the following commands,

* sudo apt-get update
* sudo apt-get install apt-transport-https
* sudo apt-get update
* sudo apt-get install dotnet-sdk-3.1
* sudo apt-get install dotnet-runtime-3.1
* sudo apt-get install aspnetcore-runtime-3.1

## Configure Nginx

To configure Nginx as a reverse proxy to forward HTTP requests to your ASP.NET Core app, modify /etc/nginx/sites-available/default. Open it in a text editor, and replace the contents with the following:

Server name should be the url of the domain or sub-domain url.

server {

listen 80;

server\_name phantomasp.kaz.com.bd www.phantomasp.kaz.com.bd;

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;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Proto $scheme;

}

listen 443 ssl; # managed by Certbot

ssl\_certificate /etc/letsencrypt/live/phantomasp.kaz.com.bd/fullchain.pem; # managed by Certbot

ssl\_certificate\_key /etc/letsencrypt/live/phantomasp.kaz.com.bd/privkey.pem; # managed by Certbot

include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot

ssl\_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

}

server {

listen 81;

server\_name phantomasp.kaz.com.bd www.phantomasp.kaz.com.bd;

location / {

root /var/www/unicef/web;

try\_files $uri $uri/ /index.html;

index index.html;

}

listen 444 ssl; # managed by Certbot

ssl\_certificate /etc/letsencrypt/live/phantomasp.kaz.com.bd/fullchain.pem; # managed by Certbot

ssl\_certificate\_key /etc/letsencrypt/live/phantomasp.kaz.com.bd/privkey.pem; # managed by Certbot

include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot

ssl\_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

}

server {

listen 82;

server\_name phantomasp.kaz.com.bd www.phantomasp.kaz.com.bd;

location / {

root /var/www/unicef/pwa/unicefpwa;

try\_files $uri $uri/ /index.html;

index index.html;

}

listen 445 ssl; # managed by Certbot

ssl\_certificate /etc/letsencrypt/live/phantomasp.kaz.com.bd/fullchain.pem; # managed by Certbot

ssl\_certificate\_key /etc/letsencrypt/live/phantomasp.kaz.com.bd/privkey.pem; # managed by Certbot

include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot

ssl\_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

}

## Create the service file

Create the service definition file:

sudo nano /etc/systemd/system/kestrel-unicefapp.service

The service is required to run the web api application. An example service file content may look like following:

[Unit]

Description=Unicef Education MIS .NET Web API App running on Ubuntu

[Service]

WorkingDirectory=/var/www/unicefapp

ExecStart=/usr/bin/dotnet /var/www/unicefapp/UnicefEducationMIS.Web.dll

Restart=always

# Restart service after 10 seconds if the dotnet service crashes:

RestartSec=10

KillSignal=SIGINT

SyslogIdentifier=dotnet-example

User=www-data

Environment=ASPNETCORE\_ENVIRONMENT=Production

Environment=DOTNET\_PRINT\_TELEMETRY\_MESSAGE=false

[Install]

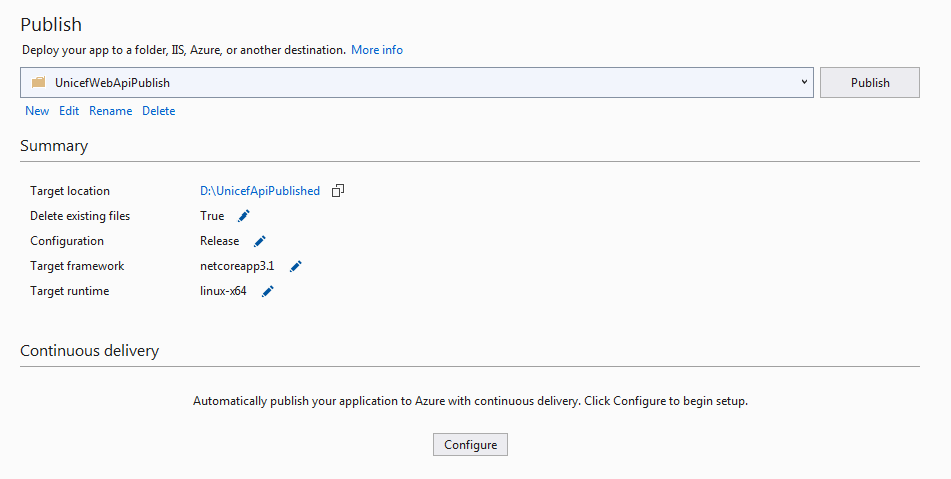
WantedBy=multi-user.target

## Installing MySQL

* wget https://dev.mysql.com/get/mysql-apt-config\_0.8.21-1\_all.deb
* wget https://dev.mysql.com/get/mysql-apt-config\_0.8.21\_all.deb
* wget https://dev.mysql.com/get/mysql-apt-config\_0.8.15-1\_all.deb
* sudo dpkg -i mysql-apt-config\_0.8.15-1\_all.deb
* sudo apt update
* sudo apt install mysql-server
* apt policy mysql-server
* sudo systemctl enable --now mysql
* systemctl status mysql
* mysql\_secure\_installation
* mysql -u root –p

## Deployment Process

### Step 1 - Publish Web API Project



### Step 2 - Delete all at deploy folder Linux Server

Using Filzilla delete all files and folders from linux server location

For Api: /home/mosfiq/Deploy/Unicef/api.

For Web App: /home/mosfiq/Deploy/Unicef/front-end

For PWA: /home/mosfiq/Deploy/Unicef/pwa/unicefpwa

### Step 3 - Upload published output to Linux Server

Using Filezilla, upload published output to Linux server at location

For Api: /home/mosfiq/Deploy/Unicef/api.

For Web App: /home/mosfiq/Deploy/Unicef/front-end

For PWA: /home/mosfiq/Deploy/Unicef/pwa

### Step 4 - Stop kestrel web application

Using MobaXterm, run following command: (API)

sudo systemctl stop kestrel-unicefapp.service

### Step 5 - Delete deployed contents

Using MobaXterm, run following command:

For Api: sudo rm -r /var/www/unicefapp/\*

For Web App: sudo rm -rf /var/www/unicef/web/\*

For PWA: sudo rm -rf /var/www/unicef/pwa/unicefpwa/\*

### Step 6 - Copy latest contents

Using MobaXterm, run following command:

For Api: sudo cp -R /home/mosfiq/Deploy/Unicef/Api/\* /var/www/unicefapp

For Web App : sudo cp -R /home/mosfiq/Deploy/Unicef/front-end/unicefweb/\* /var/www/unicef/web

For PWA: sudo cp -R /home/mosfiq/Deploy/Unicef/pwa/unicefpwa/\* /var/www/unicef/pwa/unicefpwa  
Step 7 - Start web application

Using MobaXterm, run following command:

sudo systemctl start kestrel-unicefapp.service  
API

sudo systemctl stop kestrel-unicefapp.service && sudo rm -r /var/www/unicefapp/ && sudo cp -R /home/mosfiq/Deploy/Unicef/Api/ /var/www/unicefapp/ && sudo systemctl start kestrel-unicefapp

### WEB

sudo rm -rf /var/www/unicef/web/ && sudo cp -R /home/mosfiq/Deploy/Unicef/front-end/unicefweb/ /var/www/unicef/web/