

```

[CmdletBinding()]
Param (
    [Parameter(Mandatory = $true)]
    [string]
    [ValidateNotNullOrEmpty()]
    $LicenseXmlPath,

    # We do not need to use [SecureString] here since the value will be stored unencrypted in .env,
    # and used only for transient local example environment.
    [string]
    $SitecoreAdminPassword = "Password12345",

    # We do not need to use [SecureString] here since the value will be stored unencrypted in .env,
    # and used only for transient local example environment.
    [string]
    $SqlSaPassword = "Password12345"
)

$errorActionPreference = "Stop";

if (-not (Test-Path $LicenseXmlPath)) {
    throw "Did not find $LicenseXmlPath"
}

if (-not (Test-Path $LicenseXmlPath -PathType Leaf)) {
    throw "$LicenseXmlPath is not a file"
}

# Check for Sitecore Gallery
Import-Module PowerShellGet

$SitecoreGallery = Get-PSRepository | Where-Object { $_.SourceLocation -eq
"https://sitecore.myget.org/F/sc-powershell/api/v2" }

```

```
if (-not $SitecoreGallery) {  
    Write-Host "Adding Sitecore PowerShell Gallery..." -ForegroundColor Green  
    Register-PSRepository -Name SitecoreGallery -SourceLocation https://sitecore.myget.org/F/sc-  
powershell/api/v2 -InstallationPolicy Trusted  
    $SitecoreGallery = Get-PSRepository -Name SitecoreGallery  
}  
  
# Install and Import SitecoreDockerTools  
$dockerToolsVersion = "10.0.5"  
Remove-Module SitecoreDockerTools -ErrorAction SilentlyContinue  
if (-not (Get-InstalledModule -Name SitecoreDockerTools -RequiredVersion $dockerToolsVersion -  
ErrorAction SilentlyContinue)) {  
    Write-Host "Installing SitecoreDockerTools..." -ForegroundColor Green  
    Install-Module SitecoreDockerTools -RequiredVersion $dockerToolsVersion -AllowPrerelease -  
Scope CurrentUser -Repository $SitecoreGallery.Name  
}  
Write-Host "Importing SitecoreDockerTools..." -ForegroundColor Green  
Import-Module SitecoreDockerTools -RequiredVersion $dockerToolsVersion  
  
#####  
# Populate the environment file  
#####  
  
Write-Host "Populating required .env file variables..." -ForegroundColor Green  
  
# SITECORE_ADMIN_PASSWORD  
Set-DockerComposeEnvFileVariable "SITECORE_ADMIN_PASSWORD" -Value  
$SitecoreAdminPassword  
  
# SQL_SA_PASSWORD  
Set-DockerComposeEnvFileVariable "SQL_SA_PASSWORD" -Value $SqlSaPassword  
  
# TELERIK_ENCRYPTION_KEY = random 64-128 chars
```

```
Set-DockerComposeEnvFileVariable "TELERIK_ENCRYPTION_KEY" -Value (Get-SitecoreRandomString 128)
```

```
# SITECORE_IDSECRET = random 64 chars
```

```
Set-DockerComposeEnvFileVariable "SITECORE_IDSECRET" -Value (Get-SitecoreRandomString 64 -DisallowSpecial)
```

```
# SITECORE_ID_CERTIFICATE
```

```
$idCertPassword = Get-SitecoreRandomString 12 -DisallowSpecial
```

```
Set-DockerComposeEnvFileVariable "SITECORE_ID_CERTIFICATE" -Value (Get-SitecoreCertificateAsBase64String -DnsName "localhost" -Password (ConvertTo-SecureString -String $idCertPassword -Force -AsPlainText))
```

```
# SITECORE_ID_CERTIFICATE_PASSWORD
```

```
Set-DockerComposeEnvFileVariable "SITECORE_ID_CERTIFICATE_PASSWORD" -Value $idCertPassword
```

```
# SITECORE_LICENSE
```

```
Set-DockerComposeEnvFileVariable "SITECORE_LICENSE" -Value (ConvertTo-CompressedBase64String -Path $LicenseXmlPath)
```

```
#####
```

```
# Configure TLS/HTTPS certificates
```

```
#####
```

```
Push-Location traefik\certs
```

```
try {
```

```
    $mkcert = ".\mkcert.exe"
```

```
    if ($null -ne (Get-Command mkcert.exe -ErrorAction SilentlyContinue)) {
```

```
        # mkcert installed in PATH
```

```
        $mkcert = "mkcert"
```

```
    } elseif (-not (Test-Path $mkcert)) {
```

```
        Write-Host "Downloading and installing mkcert certificate tool..." -ForegroundColor Green
```

```
Invoke-WebRequest "https://github.com/FiloSottile/mkcert/releases/download/v1.4.1/mkcert-
v1.4.1-windows-amd64.exe" -UseBasicParsing -OutFile mkcert.exe
```

```
if ((Get-FileHash mkcert.exe).Hash -ne
"1BE92F598145F61CA67DD9F5C687DFEC17953548D013715FF54067B34D7C3246") {
```

```
    Remove-Item mkcert.exe -Force
```

```
    throw "Invalid mkcert.exe file"
```

```
}
```

```
}
```

```
Write-Host "Generating Traefik TLS certificates..." -ForegroundColor Green
```

```
& $mkcert -install
```

```
& $mkcert -cert-file xp0cm.localhost.crt -key-file xp0cm.localhost.key "xp0cm.localhost"
```

```
& $mkcert -cert-file xp0id.localhost.crt -key-file xp0id.localhost.key "xp0id.localhost"
```

```
}
```

```
catch {
```

```
    Write-Host "An error occurred while attempting to generate TLS certificates: $_" -
    ForegroundColor Red
```

```
}
```

```
finally {
```

```
    Pop-Location
```

```
}
```

```
#####
```

```
# Add Windows hosts file entries
```

```
#####
```

```
Write-Host "Adding Windows hosts file entries..." -ForegroundColor Green
```

```
Add-HostsEntry "xp0cm.localhost"
```

```
Add-HostsEntry "xp0id.localhost"
```

```
Write-Host "Done!" -ForegroundColor Green
```