

7장

Web App 운영하기

전체 내용

Website 생성 및
구성하기

Web App
활용하기

1-Website 생성 및 구성하기

- Azure에서 Web site를 생성하는 다양한 방법
- App Service에서 Web App으로 Web site 생성하기
- App Service에서 WordPress로 Web site 생성하기
- Cloud Service로 Web site 생성하기
- Web site의 Scale up 및 Scale out

1-Website 생성 및 구성하기

- Azure에서 Web site를 생성하는 다양한 방법
 - App Service에서 생성하기
 - WebApp
 - WordPress
 - Cloud Service에서 생성하기
 - VM을 직접 만들어서 Web site 생성하기

1-Website 생성 및 구성하기

- App Service에서 Web App으로 Web site 생성하기
 - Service Plan에서 **최소 S1 이상의 size를 선택하여** Web App staging을 사용하여 운영중인 Web site를 Staging으로 변경한 후 Contents를 수정하여 다시 Publish를 할 수 있다
 - Web Site의 Contents를 Visual Studio와 같은 저작도구로 작성하여 업로드한다
 - 다양한 관리 옵션을 제공한다
 - **Web App 생성하기**
 - **App Services** - Add - **Web App** - Create
 - App Service plan/Location에서 적당한 site 선택
 - Web App에 대한 FQDN이 생성된다
 - ysleeapp.azurewebsites.net
 - 생성한 Web App의 Contents를 작성
 - Visual Studio로 내용 작성 후 Azure로 업로드한다

* App name
ysleeapp ✓
azurewebsites.net

* Subscription
Azure Pass ▼

* Resource Group ⓘ
☐ Create new ☒ Use existing
RGyslee ▼

* App Service plan/Location
ysleeAppServicePlan(Japan East) >

Application Insights ⓘ

1-Website 생성 및 구성하기

- App Service에서 Web App으로 Web site 생성하기
 - Service Plan 선택하기
 - Free / Shared / Basic / Standard / Premium에 따라 지원 기능이 다르다
 - 참고: <https://azure.microsoft.com/en-us/pricing/details/app-service/plans/>

	FREE Try for Free	SHARED Host Basic Apps	BASIC More Features for Dev / Test	STANDARD Go Live with Web and Mobile	PREMIUM Enterprise Scale and Integration
Web, mobile, or API apps	10	100	Unlimited	Unlimited	Unlimited
Disk space	1 GB	1 GB	10 GB	50 GB	250 GB
Logic App Actions (per day) *	200	200	200	10,000	50,000
Maximum instances	–	–	Up to 3	Up to 10	Up to 50
SLA	–	–	99.95%	99.95%	99.95%
Auto-Scale	–	–	–	Supported	Supported
Geo-distributed deployment	–	–	–	Supported	Supported
VPN hybrid connectivity	–	–	–	Supported	Supported
Staging environments	–	–	–	5	20
Custom domain	–	Supported	Supported	Supported	Supported
SSL certificates	–	–	Unlimited SNI SSL certs	Unlimited SNI SSL certs and 1 IP SSL included ¹	Unlimited SNI SSL certs and 1 IP SSL included ¹

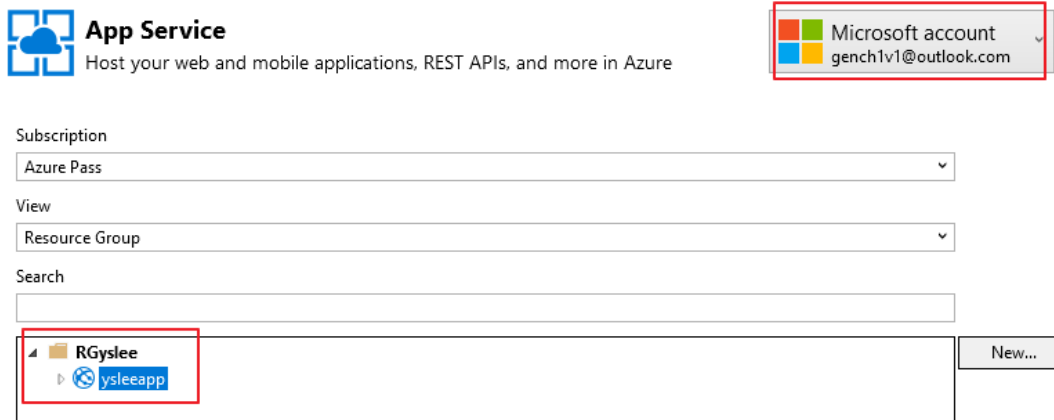
1-Website 생성 및 구성하기

- App Service에서 Web App으로 Web site 생성하기
 - **Visual Studio로 contents 작성하여 Publish하기**
 - File - New - Web Site - ASP.NET Empty Web site를 클릭 후 경로 저장
 - C:\Users\jesus\Documents\Visual Studio 2015\WebSites\ysleeapp
 - 오른쪽의 Solution Explorer에서 ysleeapp을 선택한 후 마우스 우측 버튼
 - Add- Add New Item - HTML Page 선택 후 기본 페이지 파일을 Default.html로 수정한 후 Add를 클릭한다
 - 아래 그림과 같이 내용을 입력한 후 Internet Explorer 버튼을 클릭하여 로컬 컴퓨터에 저장한 contents 파일 내용을 확인한다

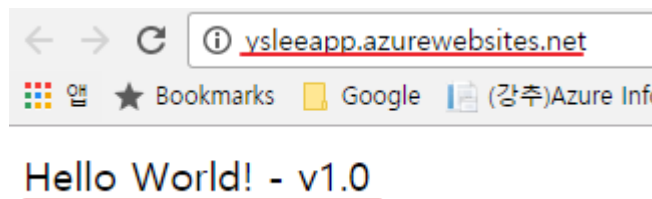
```
<body>
    Hello World! - v1.0
</body>
```
 - 내용 확인 후 반드시 "중지" 버튼을 클릭한다
 - 이제는 로컬에 저장된 내용을 Azure WebApp으로 Publish한다
 - Solution Explorer에서 ysleeapp - 마우스 우측 버튼 - Publish Web App - Microsoft Azure App Service 클릭
 - Azure에 로그인한다

1-Website 생성 및 구성하기

- App Service에서 Web App으로 Web site 생성하기
 - **Visual Studio로 contents 작성하여 Publish하기**
 - 작성한 App Service를 선택하여 OK를 클릭한다



- Validation Connection - Next - Next - Start Preview - **Publish** 클릭

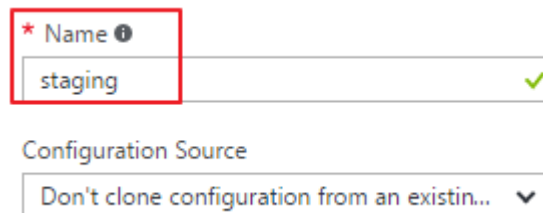


1-Website 생성 및 구성하기

- App Service에서 Web App으로 Web site 생성하기

- **Deployment slot 추가하기**

- 운영 중인 Web site를 수정할 필요가 있을 때 Deployment slot을 사용하여 Production과 Staging을 Swapping을 한다
 - App Services - ysleeapp - ADD DEPLOYMENT 항목에서 Deployment slots 클릭 - Add slot 클릭 한 후 다음과 같이 입력한다



* Name ⓘ

staging ✓

Configuration Source

Don't clone configuration from an existin... ▼

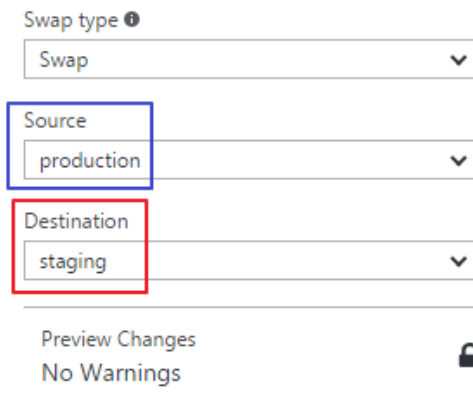
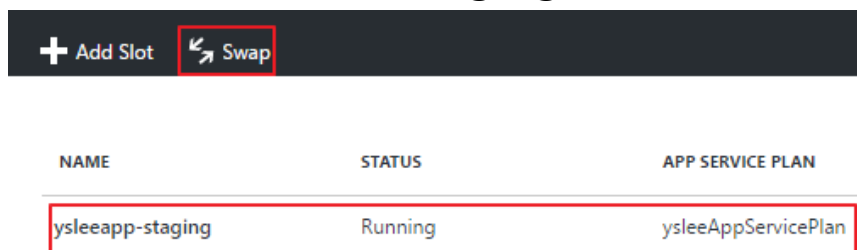
- 작성한 App Service를 선택하여 OK를 클릭한다
 - 여기서 만약 생성되지 않으면 "App Service Plan"을 클릭하여 Pricing Tier(가격 책정)를 확인한 후, 최소 S1이상이 되어야 하므로 Scale up을 클릭하여 S1 이상을 선택

1-Website 생성 및 구성하기

- App Service에서 Web App으로 Web site 생성하기

- **Web site를 Swapping(교환)하기**

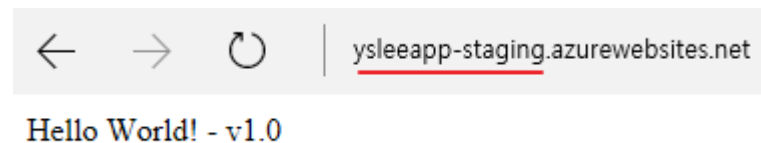
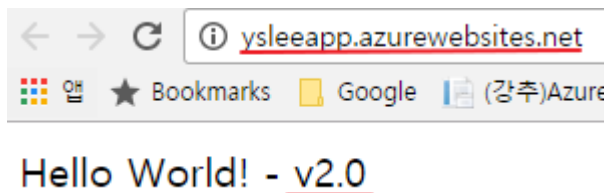
- SWAP를 클릭하여 staging으로 변경한다



- Visual Studio에서 다음과 같이 변경 한 후 다시 Publish한다

```
<body>  
  Hello World! - v2.0  
</body>
```

- Web Browser에서 <http://ysleeapp.azurewebsites.net>과 <http://ysleeapp-staging.azurewebsites.net>으로 접속해 본다



1-Website 생성 및 구성하기

- App Service에서 Web App으로 Web site 생성하기
 - **Web site swapping**하기
 - 다시 SWAP를 클릭하여 Destination을 production으로 변경한다



Swap type ⓘ

Swap

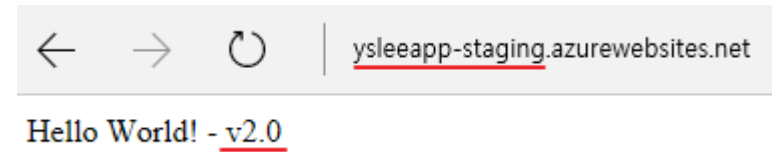
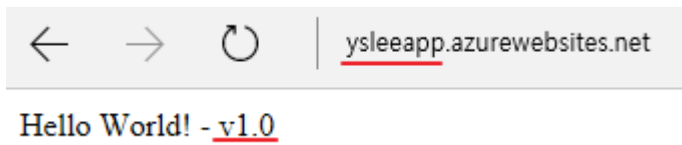
Source

staging

Destination

production

- 다시 Web Browser에서 접속해 본다
 - ysleeapp.azurewebsites.net의 내용이 v1.0으로 돌아 간다



1-Website 생성 및 구성하기

- App Service에서 WordPress로 Web site 생성하기
 - Azure에서는 다양한 상용 Web Service를 선택하여 운영할 수 있다
 - 전 세계의 Web Site 중에서 25% 정도를 차지하는 WordPress로 Web site를 운영한다
 - WordPress로 설치한 후 관리자 페이지에 로그인 한 후 다양한 Plugin을 추가하여 멋진 Web site를 구축할 수 있다
 - WordPress 생성하기
 - **App Services** - Add - **WordPress** - Create
 - App name을 입력하면 FQDN이 생성된다
 - App Service Plan은 기존 것을 선택
 - Legal Terms를 선택하여 "Purchase" 를 한다

The screenshot shows the 'Create new Web App' wizard in the Azure portal. The configuration steps are as follows:

- * App name:** ysleewp (FQDN: ysleewp.azurewebsites.net)
- * Subscription:** Azure Pass
- * Resource Group:** Create new (selected) / Use existing (unselected). Resource Group: RGyslee
- * Database Provider:** ClearDB
- * App Service plan/Location:** ysleeAppServicePlan(Japan East)
- * Database:** DefaultMySQL
- * Legal Terms (ClearDB):** Accepted

1-Website 생성 및 구성하기

- App Service에서 WordPress로 Web site 생성하기
 - 생성한 WordPress 웹 사이트에 접속하여 초기 구성을 한다
 - <http://ysleewp.azurewebsites.net>
 - 이렇게 접속하면 <http://ysleewp.azurewebsites.net/wp-admin/install.php>로 이동하여 Web site의 Language를 선택한다
 - 아래와 같이 정보를 입력한 후 Install WordPress를 클릭하여 구성을 완료

Information needed

Please provide the following information. Don't worry, you can always change these settings later.

Site Title

yslee's blog

Username

adminuser

Username can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password

UILV6s!TBE%CDovXtW

Hide

Strong

Important: You will need this password to log in. Please store it in a secure location.

Your Email

jesuswithme@gmail.com

Double-check your email address before continuing.

Search Engine
Visibility

☐ Discourage search engines from indexing this site

It is up to search engines to honor this request.

Install WordPress

1-Website 생성 및 구성하기

- Cloud Service로 Web site 생성하기
 - ASM 자원 관리 모델에서는 Cloud Service가 매우 중요하다
 - VM에 대한 보안 자동 구성
 - VM에 대한 Load balance 구성
 - Web site 생성 (사전에 package가 있어야 함)
 - Cloud Service 하나 만으로도 Web site를 운영할 수 있다
 - Cloud Service (classic) - Add

The image shows two side-by-side windows from the Azure portal. The left window is titled 'Cloud service (classic)' and contains configuration fields for a new cloud service. The right window is titled 'Upload a package' and shows the deployment details for a package.

Cloud service (classic) configuration:

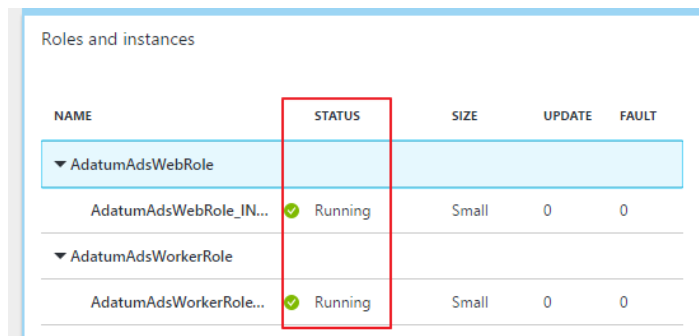
- DNS name:** ysleecs (with a green checkmark and .cloudapp.net suffix)
- Subscription:** Azure Pass
- Resource group:** RGyslee (with 'Use existing' selected)
- Location:** Japan East
- Package:** (Optional) Select a package (highlighted with a red box)
- Certificates:** (Optional) Add certificates

Upload a package configuration:

- Storage account:** sapeace
- Deployment label:** webapp_for_cloudservice (with a green checkmark)
- Package (.cspkg, .zip):** AdatumAds.cspkg (with a green checkmark and a blue icon)
- Configuration (.csfsg):** ServiceConfiguration.Cloud.csfsg (with a green checkmark and a blue icon)
- Environment:** Production (selected) / Staging
- Deploy even if one or more roles contain a single instance:** (checked)
- Start deployment:** (checked)

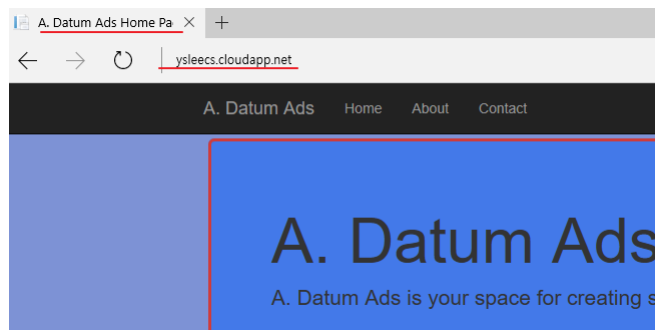
1-Website 생성 및 구성하기

- Cloud Service로 Web site 생성하기
 - 생성한 Web site에 접속하기
 - Cloud Service (classic) - ysleecs 클릭
 - 약 10분 정도 기다린 후에 Status가 running인 경우에 접속한다



NAME	STATUS	SIZE	UPDATE	FAULT
▼ AdatumAdsWebRole				
AdatumAdsWebRole_IN...	Running	Small	0	0
▼ AdatumAdsWorkerRole				
AdatumAdsWorkerRole...	Running	Small	0	0

- <http://ysleecs.cloudapp.net> 으로 접속한다



1-Website 생성 및 구성하기

- Web site의 Scale up 및 Scale out
 - Scale up의 종류에 따라 지원되는 기능이 다르다

The screenshot shows the 'Choose your pricing tier' window in the Azure portal for the application 'ysleeapp'. The left sidebar has 'SETTINGS' highlighted, and 'Scale up (App Service plan)' is selected. The main area displays three premium tiers: P1, P2, and P3. P1 is highlighted with a red box.

P1 Premium	P2 Premium	P3 Premium
1 Core	2 Core	4 Core
1.75 GB RAM	3.5 GB RAM	7 GB RAM
BizTalk Services	BizTalk Services	BizTalk Services
250 GB Storage	250 GB Storage	250 GB Storage
Up to 20 instances * Subject to availability	Up to 20 instances * Subject to availability	Up to 20 instances * Subject to availability
20 slots Web app staging	20 slots Web app staging	20 slots Web app staging
50 times daily Backup	50 times daily Backup	50 times daily Backup
Traffic Manager Geo availability	Traffic Manager Geo availability	Traffic Manager Geo availability
267,840.00 KRW/MONTH (ESTIMATED)	535,680.00 KRW/MONTH (ESTIMATED)	1,071,360.00 KRW/MONTH (ESTIMATED)
S1 Standard	S2 Standard	S3 Standard

- 부하 문제를 해결하기 위해 Scale out을 선택한다

The screenshot shows the 'Scale out (App Service plan)' window in the Azure portal for the application 'ysleeapp'. The left sidebar has 'SETTINGS' highlighted, and 'Scale out (App Service plan)' is selected. The main area shows a graph of average instances over time, with a red box highlighting the 'Scale by' dropdown menu set to 'an instance count that I enter manually'. Below the graph, the 'Instances' slider is set to 2.

Scale by: an instance count that I enter manually

Description: Manual setup means that the number of instances you choose won't change, even if there are changes in load.

Instances: 2

2-Web App 활용하기

- Custom Domain 설정하기
- FTP Client를 사용하여 **contents** 업로드하기
- Web App에 접속할 때 **사용자 인증하기**
- **GitHub**에 Web App 내용 저장하기
- **Docker Hub**에 Web App 내용 저장하기
- **Azure Container Registry**에 Web App 내용 저장하기

2-Web App 활용하기

- Custom Domain 설정하기

- Azure Web App을 생성하면 myazurekr.azurewebsites.net으로 FQDN이 생성되기 때문에 회사 도메인을 사용하는데 문제가 있다
- 이 문제를 해결하는 것이 Custom Domain 설정을 하는 것이다
- Web App에 Custom Domain을 사용하는 방법은 2가지이다
 - Azure에서 직접 도메인 구입하기
 - 외부에서 도메인을 구입한 후 Azure DNS Zone 이용하기(##강추)
- Custom Domain을 설정하여 웹사이트에 접속하기
 - 먼저 Web App을 생성한다
 - myazurekr.azurewebsites.net
 - Azure DNS Zone에서 myazure.kr 영역에서 **www** 레코드를 **CNAME**으로 설정하여 myazurekr.azurewebsites.net으로 연결한다
 - **App Services -> myazurekr -> Custom Domain -> Add Hostname -> www.myaazure.kr** 입력 후 Validate를 클릭 -> **Add Hostname**

2-Web App 활용하기

- Custom Domain 설정하기

- VS2015를 실행하여 다음과 같이 진행한다

- **File -> New -> Web site -> ASP.NET Empty Web Site -> Solution Explore에서 프로젝트의 우측 마우스 클릭 -> Add -> Add New Item -> HTML Page -> index.html 입력 -> Add**

- index.html을 만들어 업로드하기

- 이것의 내용은 내 블로그인 myazurekr.wordpress.com으로 Redirection 하도록 한다

```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="refresh" content="0; URL='http://myazurekr.wordpress.com'" />
</head>
<body>

</body>
</html>
```

- Solution Explore에서 프로젝트의 우측 마우스 클릭 -> **Publish Web App** 클릭 -> **Microsoft Azure Web Service** 클릭하여 업로드

2-Web App 활용하기

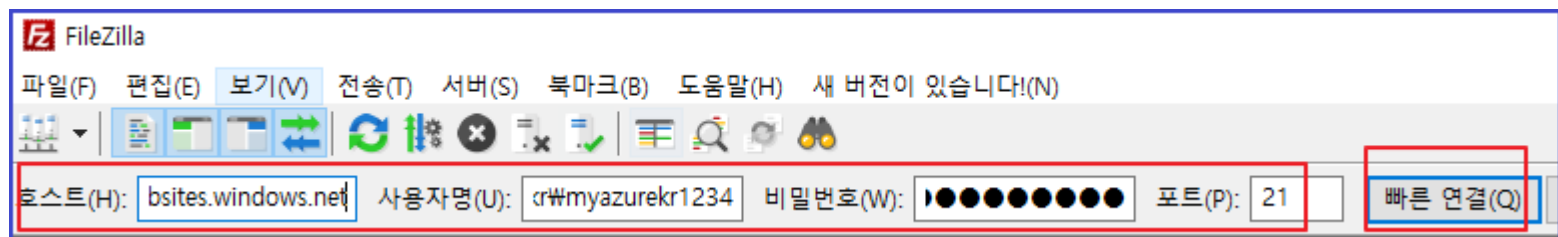
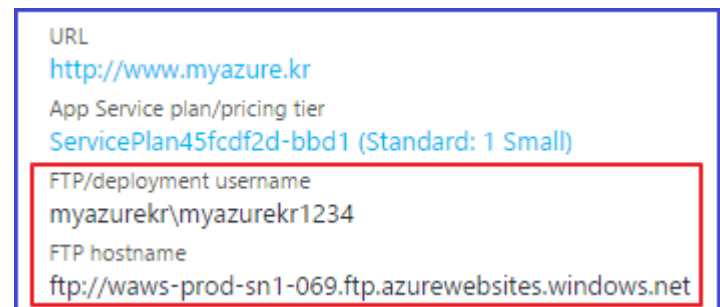
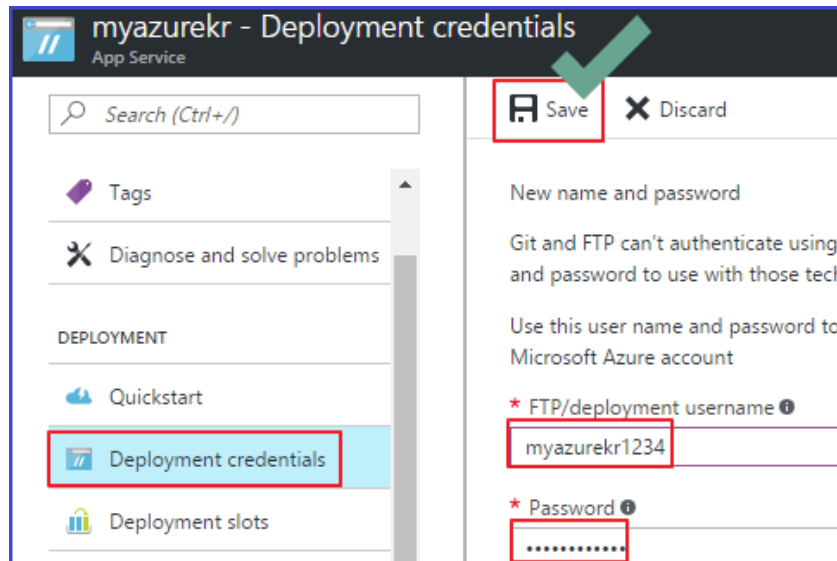
- Custom Domain 설정하기
 - 생성한 웹사이트에 접속하기
 - **http://www.myazure.kr**
 - 이 결과 URL이 Redirection이 되어 myazurekr.wordpress.com에 접속한다

The image shows two overlapping screenshots. The top screenshot is the Azure App Service portal for 'myazurekr'. The 'Overview' tab is selected in the left sidebar. In the 'Essentials' section, the 'URL' is highlighted with a red box and shows 'http://www.myazure.kr'. The bottom screenshot is a web browser window showing the WordPress site 'myazurekr.wordpress.com'. The page title is 'Azure에 Docker Swarm 구현하기' (Implementing Docker Swarm on Azure). The date 'January 6, 2017' is displayed. The text '참고: 꼭 한 번 해 볼 것' (Reference: Definitely worth trying once) is visible. The source is cited as 'Source: [Azure에 Docker Swarm 구현하기](#)'.

2-Web App 활용하기

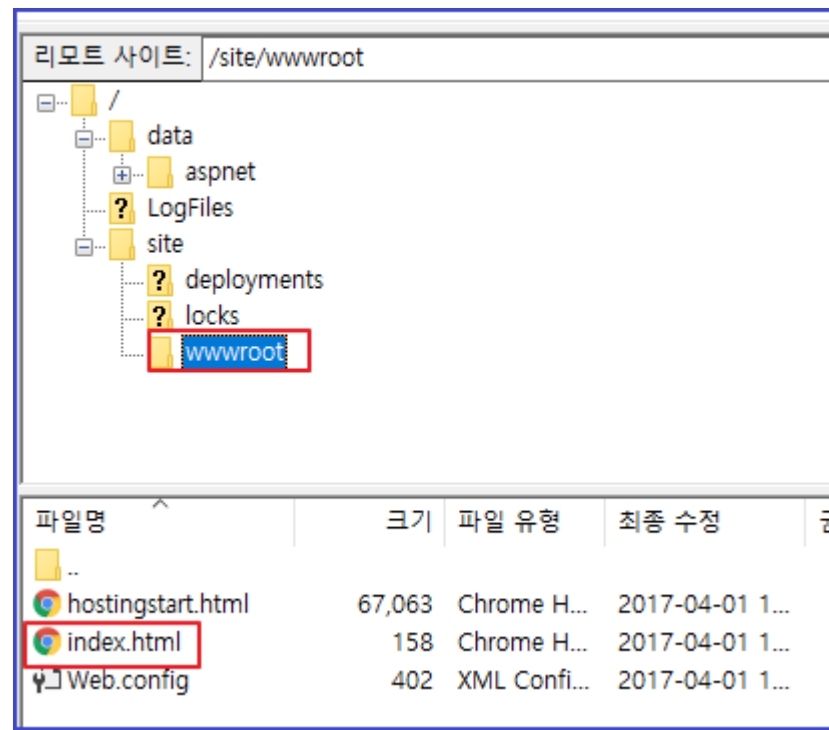
- FTP Client를 사용하여 **contents** 업로드하기

- 개발자가 생성한 코드를 FTP Client를 사용하여 한꺼번에 업로드한다
 - 이 때 Web App에서는 FTP 접속을 허용하도록 사용자와 암호를 생성한다
 - FTP Client는 Filezilla를 사용하면 된다



2-Web App 활용하기

- FTP Client를 사용하여 **contents** 업로드하기
 - 개발자가 생성한 코드를 FTP Client를 사용하여 한꺼번에 업로드한다
 - Drag & drop으로 업로드하면 된다



2-Web App 활용하기

- Web App에 접속할 때 **사용자 인증하기**
 - 생성하여 운영하고 있는 <https://peaceapp.azurewebsites.net/>에 접속할 때 보안을 위하여 사용자 인증을 요구할 수 있다
 - 사전에 peaceapp이라는 Web App을 생성해 둔다
 - 이 때 Azure AD 사용자를 사용하도록 설정한다
 - **App Services -> peaceapp -> Authentication/Authorization -> App Service Authentication**에서 **ON** 클릭 -> **Log in with Azure Active Directory -> Management mode**에서 **Express** 클릭 -> **Current Active Directory**에서 "기본 디렉터리" 선택 -> **Azure AD app**에서 **peaceapp** 선택
 - Web Browser에서 https://peaceapp.azurewebsites.net를 입력하여 로그인한다
 - 그러면 사용자 인증창이 나온다 (**성공!!**)
 - Azure Active Directory를 사용하는 App 확인하기
 - **Azure Active Directory - Enterprise applications -> All applications**에서 생성한 peaceapp이 추가된 것을 알 수 있다

GW	Outlook Groups
PE	peaceapp https://peaceapp.azurewebsites.net

2-Web App 활용하기

- **GitHub**에 Web App 내용 저장하기
 - Azure Web App의 Contents를 저장하는 곳
 - **Azure**
 - Windows, Linux
 - **GitHub**
 - **Docker**
 - **Docker Hub**
 - **Azure Container Register**

The screenshot shows the 'Web App' creation wizard in the Azure Portal, specifically the 'Container settings' tab. The left pane shows the 'Web App' configuration with the following details:

- App name:** jesuswithme (with a green checkmark)
- Subscription:** MSDN Platforms (34de01eb-88a0-4743-9c...)
- Resource Group:** classRG (with 'Use existing' selected)
- OS:** Docker (selected from Windows, Linux, Docker)
- App Service plan/Location:** ServicePlan5fa6df28-bc41(West ...)
- Configure container:** (link to further configuration)

The right pane shows the 'Container settings' section with the following details:

- Container settings:** Single Container (selected), Docker Compose (Preview), Kubernetes (Preview)
- Image source:** Quickstart, Azure Container Registry (selected), Docker Hub, Private registry
- Repository Access:** Public (selected), Private
- Image and optional tag (eg 'image:tag'):** jesuswithme/gosmall

2-Web App 활용하기

- Git 명령어 사용하기

- Git이란?

- Git(재수없고 멍청한 놈/자식)은 GitHub를 관리하는 명령어이다

- Git을 왜 사용하는가?

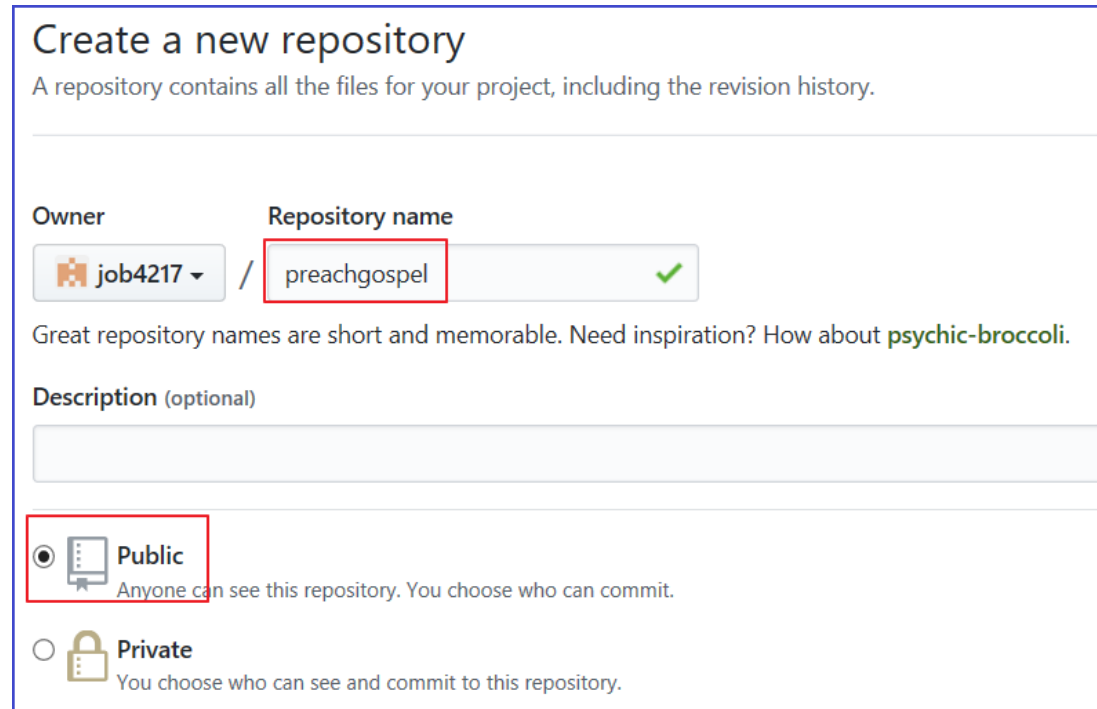
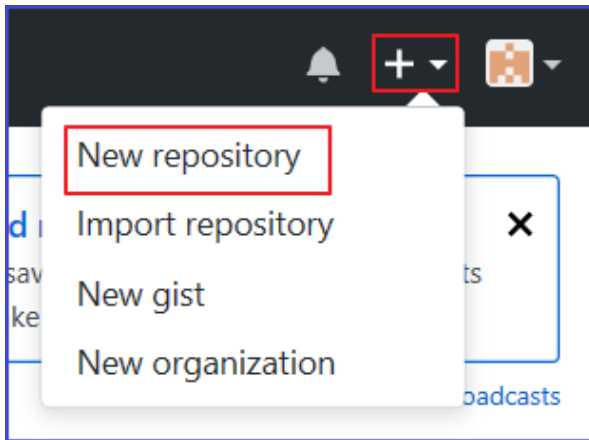
- 2명 이상의 개발자가 각각 다른 컴퓨터에서 coding을 하여 동일한 Web Site에 code를 upload하는 경우에 예상하지 못한 결과를 초래할 수 있다
 - Code를 잘못 수정한 경우에 이전 내용으로 돌아가고자 한다
 - 이럴 때 git을 사용하면 로컬 컴퓨터에서 commit을 하거나 GitHub.com으로 code를 push하면 항상 먼저 snapshot(checkpoint)을 수행하고 저장하도록 하는 것이 git이다
 - 그 결과 code의 충돌 방지 및 이전 내용으로 돌아갈 수 있어서 실수를 줄일 수 있다

- Git을 사용하려면?

- 로컬 컴퓨터에 git 프로그램을 설치해야 한다
 - <https://git-scm.com/download/win>에 접속하여 설치한다
 - Windows에 git을 설치하여 Linux 명령어를 사용하게 된다

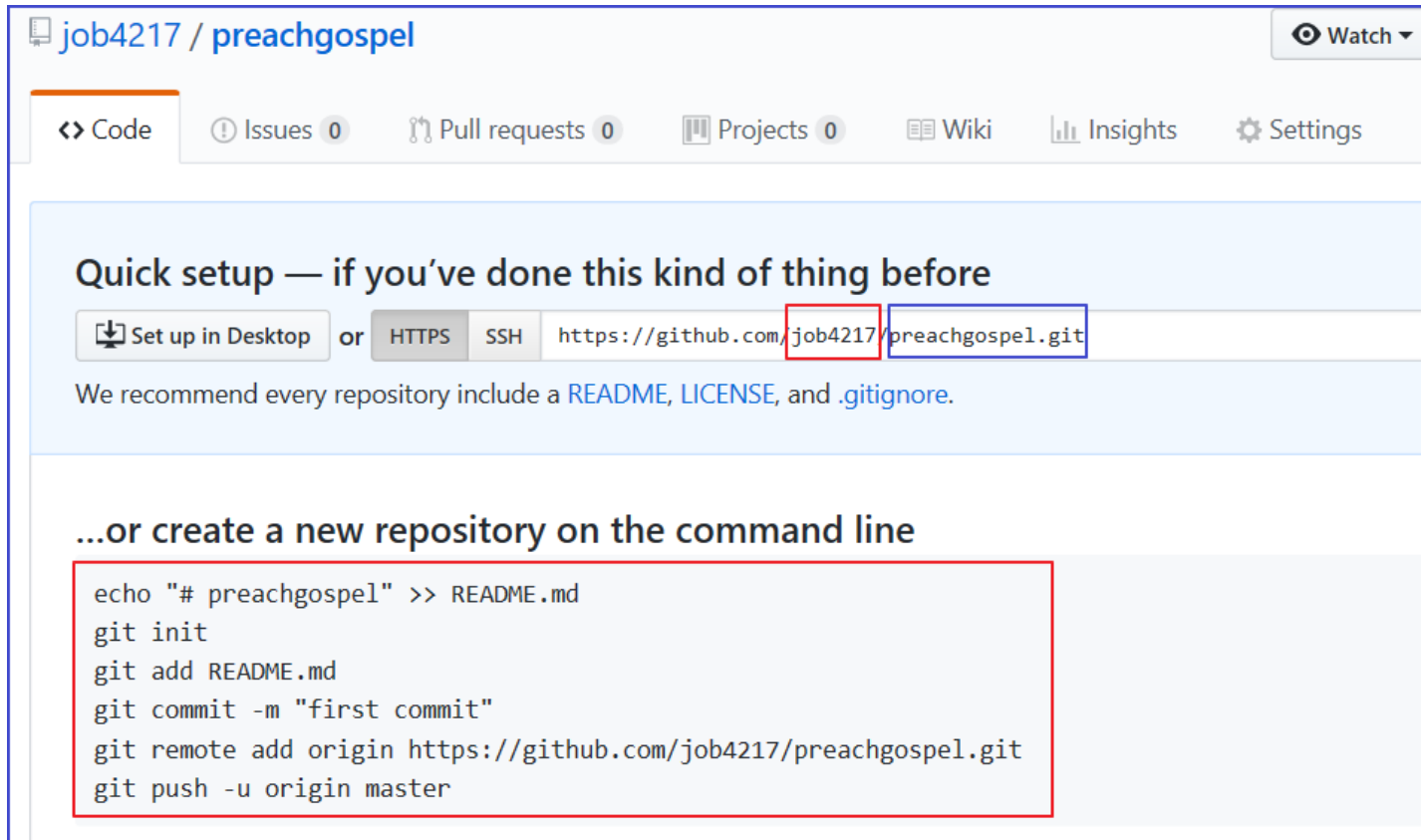
2-Web App 활용하기

- Git 명령어 사용하기
 - GitHub.com에 회원 가입하기
 - github.com에 접속하여 회원 가입한다
 - 이 때 Email 주소와 ID를 꼭 기억해 둔다
 - GitHub에 Repository 생성하기

A screenshot of the 'Create a new repository' form on GitHub. The form title is 'Create a new repository' with a subtitle 'A repository contains all the files for your project, including the revision history.' Below this, there are two main sections: 'Owner' and 'Repository name'. The 'Owner' field shows 'job4217' with a dropdown arrow. The 'Repository name' field shows 'preachgospel' with a green checkmark icon to its right. Below these fields, there is a text prompt: 'Great repository names are short and memorable. Need inspiration? How about **psychic-broccoli**.' Underneath is a 'Description (optional)' text input field. At the bottom, there are two radio button options for repository visibility: 'Public' (selected, with a red box around it) and 'Private'. The 'Public' option has a subtext: 'Anyone can see this repository. You choose who can commit.' The 'Private' option has a subtext: 'You choose who can see and commit to this repository.'

2-Web App 활용하기

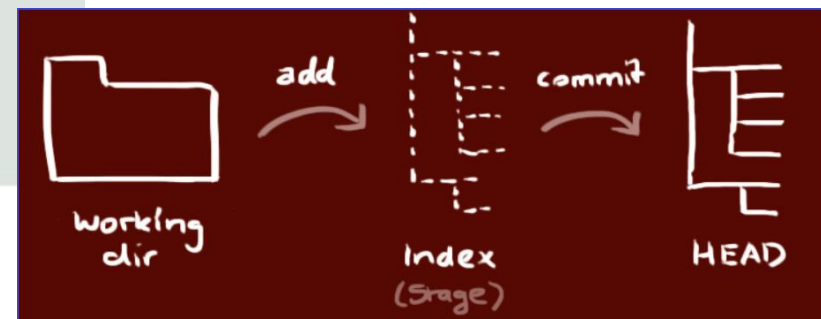
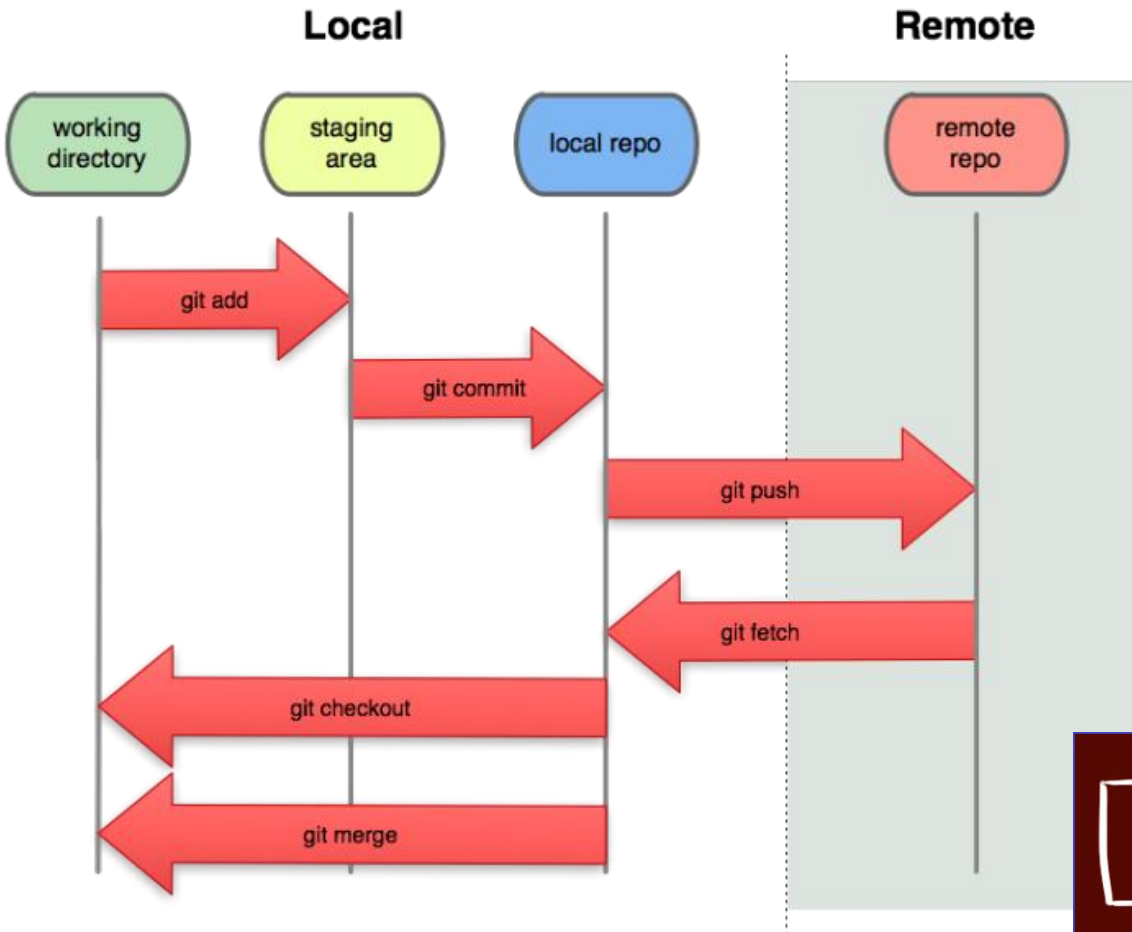
- Git 명령어 사용하기
 - GitHub에 Repository 생성하기(계속)



- **`https://github.com/아이디/프로젝트이름.git`** 주소를 기억해 둔다

2-Web App 활용하기

- Git 명령어 사용하기
 - Git 작업 흐름 이해하기



2-Web App 활용하기

- Git 명령어 사용하기
 - Git은 항상 git으로 시작한다
 - **git status**
 - **git init**
 - Git 명령어 도움말 사용하기
 - **git help init**
 - **git help status**
 - Git 사용자 정보 등록하기
 - **git config --global user.name "Yongshik Lee"**
 - **git config --global user.email "job4217@outlook.com"**
 - Email은 github.com에 등록할 때 사용한 것을 입력해야 한다
 - Local Repository로 사용할 디렉터리 생성하기
 - `cd`
 - `mkdir preachgospel`
 - `cd !$`
 - `i$`: 이전 명령어의 마지막 인자(preachgospel)를 대신하여 사용한다
 - `pwd`

2-Web App 활용하기

- Git 명령어 사용하기

- 현재 directory를 git이 관리하는 디렉터리로 초기화하기(Local Repo 생성)
 - ls -la
 - **git init**
 - ls -la
 - 프로젝트 파일이 저장될 Local Directory를 생성한 것이다
- 파일 생성하기
 - **echo** "This is my web site in Azure." > **index.html**
 - **cat** index.html
- 파일 상태 확인하기
 - **git status**
 - 아무 정보가 없다
- Staging Area로 파일을 이동하기
 - **git add** index.html
 - **git status**

2-Web App 활용하기

- Git 명령어 사용하기

- Local Repo로 파일 이동하기

- **git commit -m** "1-Adding index.html file to Local Repo" **index.html**
 - 이 때 내부적으로 snapshot을 찍어 둔다
 - **echo** "God is Love." > **gospel.html**
 - **git add** gospel.html
 - **git commit -m** "2-Adding gospel.html"
 - **echo** "Good Job!" >> index.html
 - **echo** "Good news." >> gospel.html
 - **git status**
 - **git commit -a -m** "3-Adding all files to Local repo"
 - working directory에 변경된 모든 파일을 commit하기
 - **git status**

2-Web App 활용하기

- Git 명령어 사용하기

- Commit하여 Local Repo에 기록된 내용 확인하기(log 확인)

- **git log**
- **git log --oneline**

- 특정한 파일이 어떤 commit을 했는지 확인하기



- **git log -- index.html**
- **git log -- gospel.html**

- GitHub의 원격 저장소 지정하기

- 로컬 저장소에 저장된 파일을 GitHub에 올리기 전에 먼저 원격 저장소를 지정해 두어야 한다
- **git remote add origin <https://github.com/job4217/preachgospel.git>**
- **git remote -v**
 - 연결이 잘되었는지 확인한 것임

2-Web App 활용하기

- Git 명령어 사용하기
 - GitHub에 파일 업로드하기
 - **git push origin master**
 - GitHub의 ID와 암호를 입력한다. 그러면 2개의 파일이 업로드된다
 - GitHub에서 추가된 파일 확인하기

job4217 4-Adding updated files to Local Repo		Latest commit c812bd1 13 minutes ago
 gospel.html	4-Adding updated files to Local Repo	13 minutes ago
 index.html	4-Adding updated files to Local Repo	13 minutes ago

- 이전 파일로 돌아가기(수정되기 전으로 돌리기)
 - 이전으로 돌릴 파일의 commit ID 확인하기: **git log -- index.html**
 - **git reset --hard <커밋된 ID>**

2-Web App 활용하기

- GitHub에 Web App 내용 저장하기
 - 사전 설정하기(앞에서 이미 작업함)
 - Github.com에 회원 가입하여 사용자 계정 생성하기
 - Github.com에서 Windows 명령어 버전의 프로그램 다운로드하여 설치하기
 - <https://git-scm.com/downloads/>
 - 설치한 git 실행하여 확인하기
 - **git version**
- Azure Portal에서 Web App 생성하기
 - Web App 이름: #####**git** (#####:자신의 영문 Intial)
 - 생성한 Web App에 접속하기: **ysleegit.azurewebsites.net**

2-Web App 활용하기

- GitHub에 Web App 내용 저장하기
 - github.com에 프로그램 code를 저장할 Repository 생성하기
 - Repository 이름: #####git

The screenshot shows the GitHub interface for creating a new repository. The top navigation bar includes the GitHub logo, a search bar, and links for Pull requests, Issues, Marketplace, and Explore. A notification banner for Terms of Service and Privacy Statement is visible. Below the navigation bar, there are tabs for Repositories and a highlighted 'New repository' button. To the right, there's a section for 'Browse activity' showing a recent star event. The main content area is titled 'Create a new repository' and explains that a repository contains all files for a project. It features two input fields: 'Owner' (set to 'jesuswithme') and 'Repository name' (set to 'ysleegit' with a green checkmark). A note at the bottom suggests that great repository names are short and memorable.

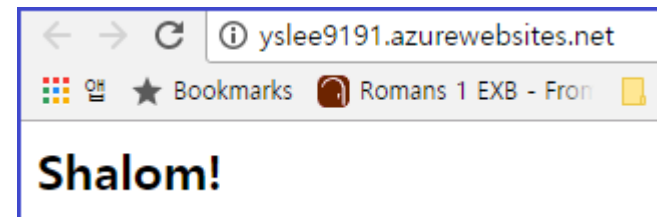
- Public을 선택하고 **Create Repository**를 선택하여 생성하면 된다
- 화면을 잘 보면 git의 기본적인 사용법을 알 수 있다

2-Web App 활용하기

- GitHub에 Web App 내용 저장하기
 - Github.com의 내용과 Azure WebApp을 연결하기
 - ysleegit.azurewebsites.net을 다음과 같이 편집하여 Github.com 내용을 불러온다

The screenshot shows the Microsoft Azure portal interface for a user named 'yslee9191'. The breadcrumb navigation at the top reads: Microsoft Azure > App Services > yslee9191 > Deployment option > Choose source. The left sidebar shows the 'Deployment options' menu item highlighted with a red box. The main content area is divided into two panes. The left pane, titled 'Deployment option', shows a red box around the 'Choose Source' link with the text 'Configure required settings'. The right pane, titled 'Choose source', lists several deployment providers: Visual Studio Team Services, OneDrive, Local Git Repository, GitHub (highlighted with a red box), and Bitbucket. To the right of the main content, a sidebar titled 'Deployment option' lists configuration steps: 'Choose Source' (with 'GitHub' in a red box), 'Authorization' (with 'jesuswithme' in a red box), 'Choose project' (with 'testwebapp' in a red box), and 'Choose branch' (set to 'master'). At the bottom of this sidebar, it says 'Performance Test Not Configured'.

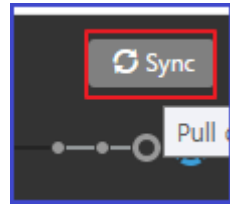
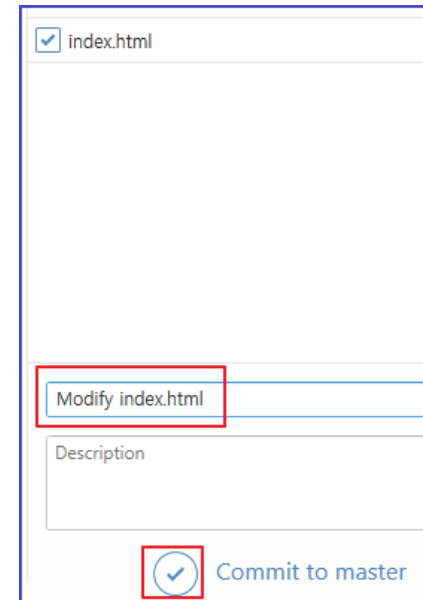
- 다시 Azure Web App에 접속해 본다
 - <http://ysleegit.azurewebsites.net>



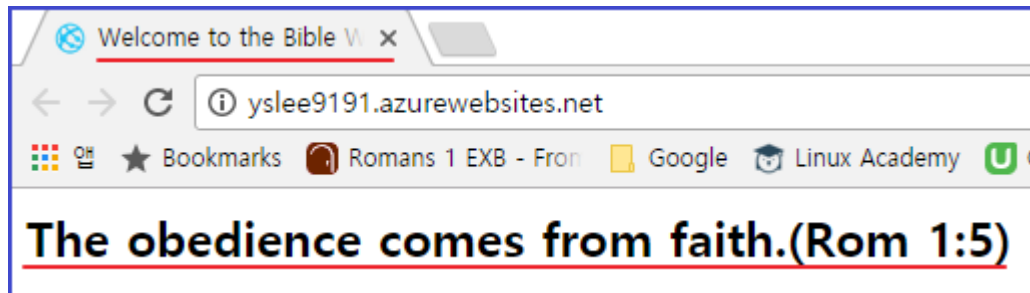
2-Web App 활용하기

- GitHub에 Web App 내용 저장하기
 - 다시 VS Code에서 Github.com에 있는 index.html 파일을 수정하여 Github Desktop을 다시 Sync를 해 본다

```
<html>
  <head>
    <title>Welcome to the Bible World</title>
  </head>
  <body>
    <h2>The obedience comes from faith.(Rom 1:5)</h2>
  </body>
</html>
```

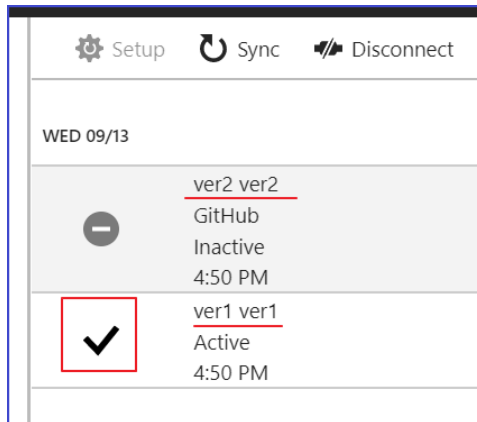


- 다시 yslee9191.azurewebsites.net에 접속하여 수정된 여부를 확인한다



2-Web App 활용하기

- WebApp을 GitHub에서 관리할 때의 장점
 - GitHub.com에 Contents를 올리면 Version 관리가 된다
 - 원하는 버전을 Redeploy를 하여 ver1, ver2를 번갈아 가면서 적용할 수 있다



- 자신이 개발한 것을 다른 사람과 공유할 수 있다
- 트래픽을 Github.com으로 보내므로 비용 절감이 된다(?)

2-Web App 활용하기

- Docker Hub에 Web App 내용 저장하기
 - Docker Hub(<http://hub.docker.com>)를 사용하는 방법은 8장에서 학습한다
 - Web App의 Contents를 docker hub를 이용하여 운영하기
 - docker hub에 있는 주소: **jesuswithme/gosmall**

Home > App Services > Web > Web App > Web App > Container settings

Web App

Create

* App name
jesuswithme ✓
.azurewebsites.net

* Subscription
MSDN Platforms (34de01eb-88a0-4743-9c' ▼

* Resource Group ⓘ
☐ Create new ☒ Use existing
classRG ▼


* OS
☐ Windows ☐ Linux ☒ Docker

* App Service plan/Location
ServicePlan5fa6df28-bc41(West ... >

* Configure container >

Container settings

Single Container Docker Compose (Preview) Kubernetes (Preview)

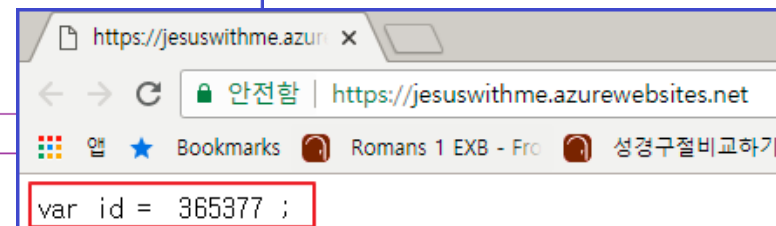
 Container Settings

Bring your own Docker formatted container images and easily deploy and

Image source
☐ Quickstart ☐ Azure Container Registry ☒ Docker Hub ☐ Private registry

Repository Access
☒ Public ☐ Private

* Image and optional tag (eg 'image:tag')
jesuswithme/gosmall



2-Web App 활용하기

- Azure Container Registry에 Web App 내용 저장하기
 - Azure Container Registry에 docker image를 업로드한 후에 Web App에서 Container Settings에서 경로를 설정하면 된다
 - 사전에 docker image를 업로드 해두어야 한다(자세한 내용은 docker에서 학습한다)

