

```

1 Lab4. GitHub과 Jenkins 연동하기
2   -Jenkins job이 GitHub의 Push Event 받기
3
4 1. Jenkins Plugin 추가 설치
5 1)GitHub Integration Plugin 설치
6   -[Jenkins 관리] > [System Configuration] > [Plugins] > [Available plugins]
7   -"Github"으로 검색
8   -목록에서 "GitHub Integration" 체크
9   -[Install without restart] 버튼 클릭하여 설치
10  -[Download progress] 페이지에서 "Github Integration"이 "성공"으로 확인
11
12 2)Python Plugin 설치
13   -[Jenkins 관리] > [System Configuration] > [Plugins] > [Available plugins]
14   -"Python"으로 검색
15   -목록에서 "Python" 체크
16   -[Install without restart] 버튼 클릭하여 설치
17   -[Download progress] 페이지에서 "Python"이 "성공"으로 확인
18
19
20 2. GitHub Setting
21 1)GitHub repository 에 소스를 push 했을때, Jenkins 서버에 webhook 을 날려서 자동으로 빌드가 되도록 세팅하는 과정이다.
22 2)Github plugin은 jenkins가 Github의 Webhook을 받을 수 있도록 수동 모드와 자동 모드를 제공한다.
23 3)수동 모드는 사용자가 Git Repository setting에서 Hook & Services를 직접 등록해 사용하는 것이고,
24 4)자동 모드는 사용자의 GitHub OAuth 토큰을 이용해 Jenkins가 자동으로 Hooks & Service를 등록한다.
25 5)자동으로 Webhook이 만들어지지 않는 경우 수동으로 Webhook을 만든다.
26 6)payload URL : Jenkins주소/github-webhook/
27
28 7)GitHub에 Deploy Key 등록하기
29   -GitHub의 해당 Repository의 [Settings]
30   -좌측 메뉴 중 [Deploy keys]
31   -[Deploy keys] 페이지에서
32     --[Add deploy key] click
33     --[Title] : jenkins_deploy_key
34     --Key
35       ---.ssh\id_ed25519.pub값 또는 id_rsa.pub의 내용을 에디터로 열어서 그 내용을 복사해서 붙여 넣는다.
36     --[Add key] click
37
38 8)GitHub 새로운 Repository(Hello-World) 생성
39   -ex)Repository url : https://github.com/gitinstructor/Hello-World.git
40
41 9)Webhook Setting
42   -Webhook 은 웹의 Trigger 같은 존재다.
43   -GitHub의 지정한 브랜치에 소스가 push되면 webhook으로 Jenkins에게 알려주어 빌드를 유발하도록 설정한다.
44   -GitHub의 해당 Repository의 [Settings]
45   -좌측 메뉴 중 [Webhooks]
46     --[Webhooks] 페이지에서 [Add webhook] 버튼 click
47     --Payload URL :
48       ex)http://ec2-3-39-227-190.ap-northeast-2.compute.amazonaws.com:8080/github-webhook/
49       ※ 주의사항
50         --EC2의 경우 주소가 변경될 수 있음.
51         --반드시 마지막에 / 추가할 것
52
53   -Content type : application/json
54   -[Add webhook] click
55
56
57 3. Jenkins에 새로운 Item 생성하기
58 1)좌측 메뉴에서 [새로운 Item] 클릭
59 2)Enter an item name : deployment_demo
60 3)[Freestyle project] 클릭 > [OK] 클릭
61
62 4)Jenkins의 왼쪽 메뉴에서 [소스 코드 관리] > [Git] 선택
63   -[Repository URL] : https://github.com/gitinstructor/Hello-World.git
64   -[Credentials] > [Add] > [Jenkins]
65   -[Jenkins Credentials Providers: Jenkins] 창에서,
66     --[Domain] : Global credentials(unrestricted)
67     --Kind : SSH Username with private key
68     --[Scope] : Global(Jenkins, nodes, items, all child items, etc)
69     --[ID] : ssh-key
70     --[Username] : ssh-key
71     --[Private Key] > Enter directly check
72       ---[Add] 버튼 클릭
73       ---.ssh\id_ed25519값 또는 id_rsa의 내용을 에디터로 열어서 내용 복사 후 붙여넣는다.
74
75     ---ex)
76     -----BEGIN OPENSSSH PRIVATE KEY-----
77     b3BlbnNzaC1rZXktdjEAAAAAAAAACmFlczI1Ni1jdHIAAAAGYmNyeXB0AAAAGAAAAABAGr/xjEB
78     LC9jwrrnzO0H0E9AAAAEAAAAEAAAAzAAAAC3NzaC1lZDI1NTE5AAAAIHhbrfY//xy/xJzk
79     1MYletTGNmiAXuVaioUPIWrbqKmXAAAAoO51CRIU6OY/IrKrGqgOqsPFbpopZsjV82edWV
80     ILK/jjY0jrobOIm54GxwT79BvKVnvDgFheecpM5TjT7XsnsVbQH3tQ6OoXDQR4PifeCCY
81     88IQ2TTXuqKK2le8NVAZmfj/KuuVyI+Ovn94E9YH9Mr9nCbQvMcL61IZZH9V5Ai0SfvjPW
82     ixz4Idv3j4hKs+NBpaG2ER2u+HsB3M/Y+kEpY=
83     -----END OPENSSSH PRIVATE KEY-----
84

```

```
85         --Add click
86     -[Credentials]에서 방금 생성한 ssh-key를 선택한다.
87
88     -[Branches to build]
89         --[Branch Specifier(blank for 'any')]
90         --*/main
91
92     -[저장] 버튼 클릭
93
94 5)좌측 메뉴의 [구성] > [빌드 유발]
95     -빌드 유발 에서는 [GitHub hook trigger for GITScm polling] 를 선택
96     -GitHub 의 hook trigger 를 받으면 빌드를 하겠다는 것
97
98 6)[Build Steps] > [Add build step]
99     -목록에서 [Execute shell] 선택
100     -[Command] 에 다음 입력
101         echo "GitHub hook trigger is working"
102
103     -[저장] 버튼 클릭
104
105
106 5. 위에서 생성한 Github에 새로운 push를 해보자.
107     1)파일 생성
108         $ nano hello.py
109
110     2)Staging에 등록
111         $ git add hello.py
112
113     3)Commit
114         $ git commit -m 'Jenkins Upload'
115
116     4)Github에 Push
117         $ git push -u origin main
118
119     5)Push 후 잠시 시간이 지난 후 자동으로 Jenkins에서 Build 확인
```