

Hands On Exercises

Working with Azure Repos in DevOps

Lab 1 Working with Azure Repos 2

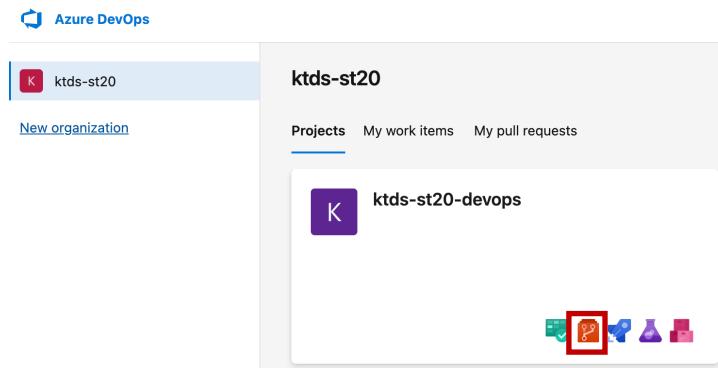
- | | |
|--|---|
| 1. Azure Repos 연결하기 | 2 |
| 2. Create notifications | 6 |
| 3. Team Explorer Everywhere plugin 사용해서 repos 연결 | 9 |

Lab 1 Working with Azure Repos

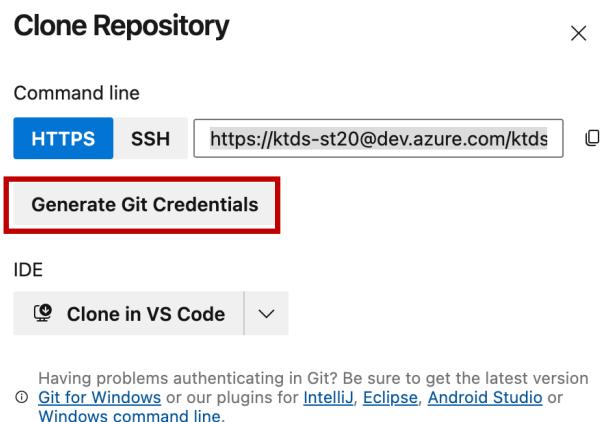
1. Azure Repos 연결하기

1.1 Security Credential 가지고 오기

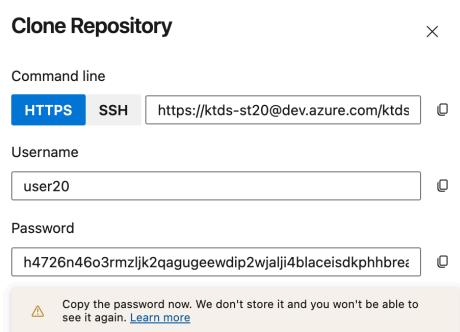
1.1.1 Azure DevOps Console에서 메뉴에서 Repos 선택 후 clone 선택



1.1.2 Clone Repository에서 HTTPS 주소를 복사하고, "Generate Git Credentials"를 선택

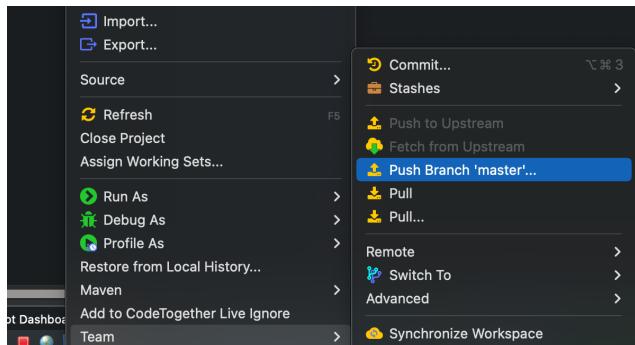


1.1.3 생성한 암호는 별도로 기록해서 Eclipse에서 원격 repos 연결 시에 사용.(1.2.2)

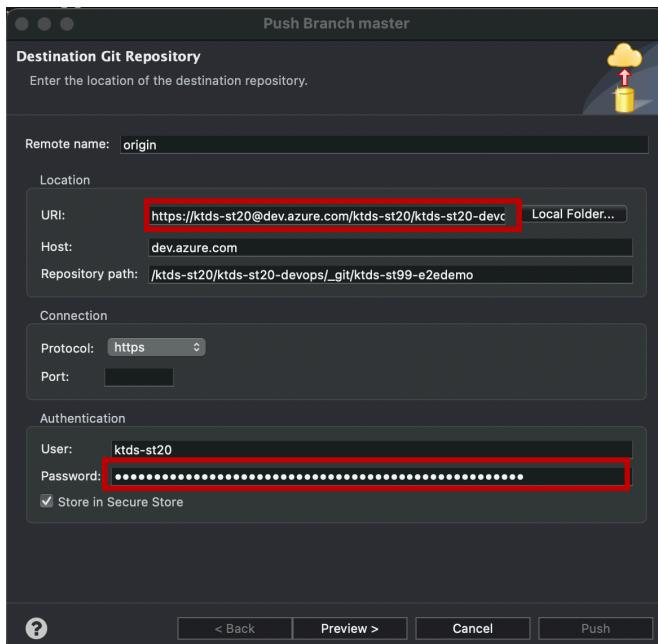


1.2 Eclipse에서 AWS CodeCommit 연결 (전제 조건 3 단계)

1.2.1 From Java view, right click on project and go to Team

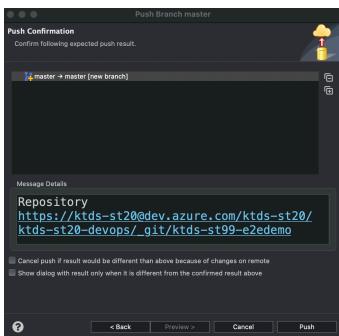


1.2.2 Select Push branch master. A pop-up window open. Enter the address you copied from above into the URI location



1.2.3 In Authentication, enter the information for credentials you copied above. Click on "Store in Secure Store"

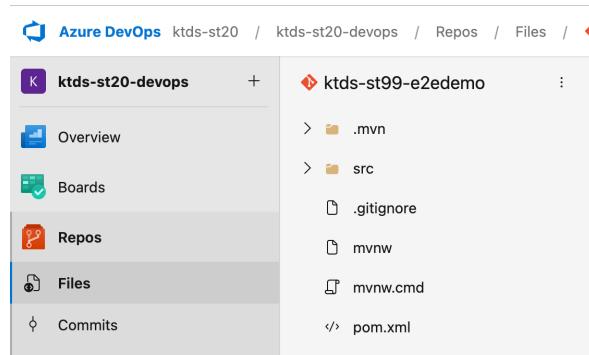
1.2.4 If the login screen pops up again, just reenter the information.[Preview]



1.2.5 [Push] - You will get a confirmation pop-up that is has been pushed. [Close]



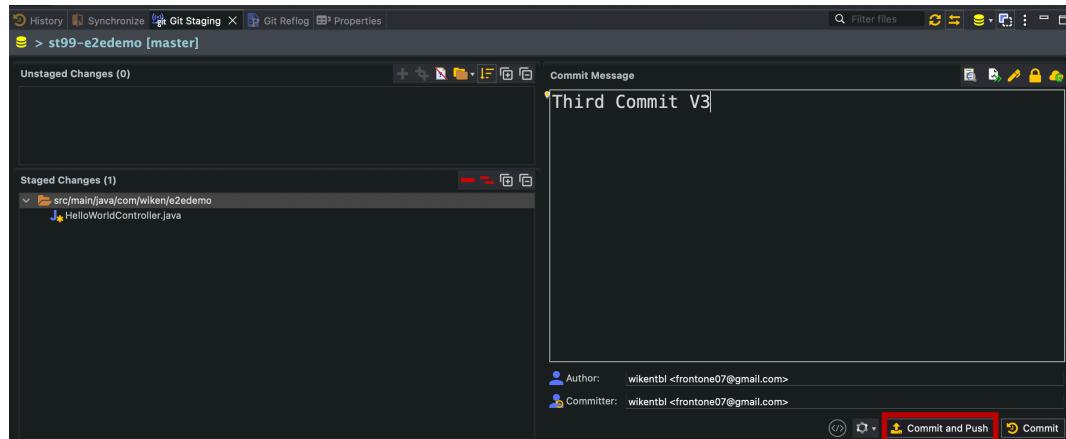
1.2.6 Azure DevOps에서 repository 선택 후 확인



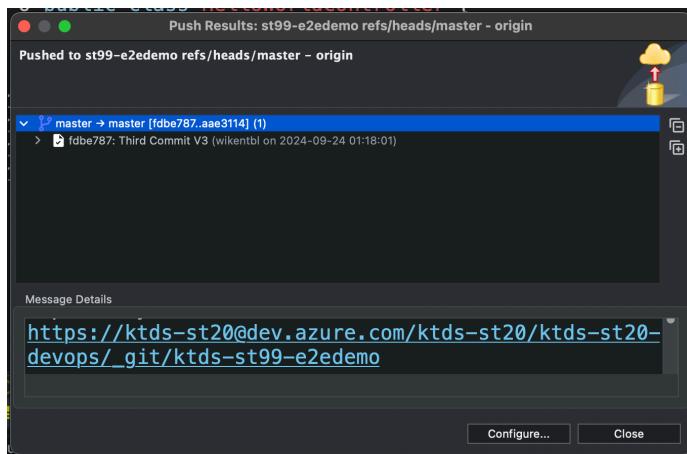
1.2.7 HelloWorldController.java에서 출력 문자를 다음과 같이 변경 후 저장

```
public String sayHello() {  
    return "Hi. Welcome to MSA apps on the Cloud - v3";
```

1.2.8 Commit to local as Third Commit and push to Azure DevOps.



1.2.9 This time, after setting the staged changes, enter “Third Commit”, you will select [Commit and Push] since we now have Azure DevOps to push to. A pop up window will confirm success



1.3 Go back to Azure DevOps and verify the Third Commit

1.3.1 On the left tab menu, select Commits. You should see three commits. 확인 후에 아래
Third commit v3 를 클릭하여 내용 확인

1.3.2 Repository에서 내용을 update 확인 (inline/side-by-side)

```

    7   7
    8   8     @GetMapping("/hello")
    9   9     public String sayHello() {
10  10     +         return "Hi. Welcome to QNA apps on the Cloud - v2";
11  11     }
12  12   }
13  13
  
```

2. Create notifications

2.1 Azure DevOps 에 로그인

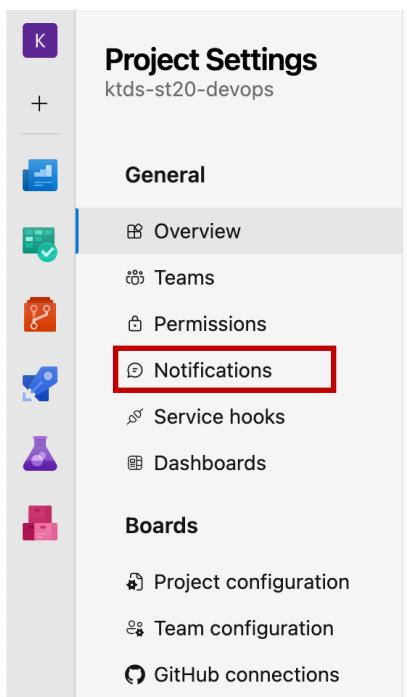
2.1.1 먼저 Azure DevOps 포털에 로그인합니다.

2.1.2 프로젝트 대시보드로 이동하여 알림을 설정할 프로젝트를 선택합니다.

2.2 프로젝트 설정으로 이동

2.2.1 화면 왼쪽 하단에 있는 Project Settings (프로젝트 설정) 메뉴를 클릭합니다.

2.2.2 Notifications (알림) 메뉴 선택



2.3 프로젝트 설정 화면에서 Notifications 또는 Service Hooks 를 활용

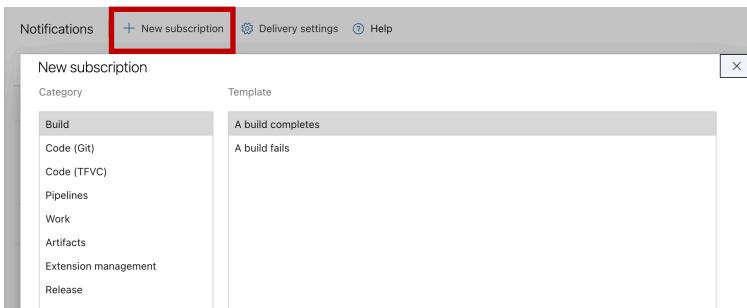
2.3.1 Notifications 메뉴는 프로젝트에서 발생하는 다양한 이벤트에 대한 알림 설정을 관리할 수 있는 곳입니다.

Description	Type	Notifies	State
Build	Build completed (any project)	ktds-st20-devops Team Individual members	<input checked="" type="checkbox"/>
Code (Git)	Pull request (any project)	ktds-st20-devops Team Individual members	<input checked="" type="checkbox"/>
	Pull request changes (any project)	ktds-st20-devops Team Individual members	<input checked="" type="checkbox"/>

2.4 새 알림 구독 만들기

2.4.1 **Notifications** 페이지에서 기본 제공되는 알림 템플릿을 사용할 수 있습니다.

2.4.2 상단에 있는 New subscription (새 구독) 버튼을 클릭합니다.



2.5 **New subscription** 창이 나타나면, 알림을 받을 이벤트 유형을 선택합니다.

2.5.1 **Git Pushes** 이벤트를 선택합니다. 이 항목은 **새로운 소스가 푸시될 때** 발생하는 이벤트를 의미합니다.

2.6 알림 세부 사항 설정

2.6.1 알림 구독의 세부 사항을 설정할 수 있습니다:

2.6.2 **Name:** 구독에 적절한 이름을 지정합니다. 예: "New Push Alert".

2.6.3 **Filter:** 모든 푸시 이벤트에 대해 알림을 받거나, 특정 브랜치에 대한 알림을 받을 수 있습니다. 여기서 브랜치를 지정하려면 **Branch** 옵션을 설정하여 알림을 받을 브랜치를 선택합니다.

2.6.4 **Deliver to :** 알림을 받을 팀원이나 이메일 주소를 설정합니다.

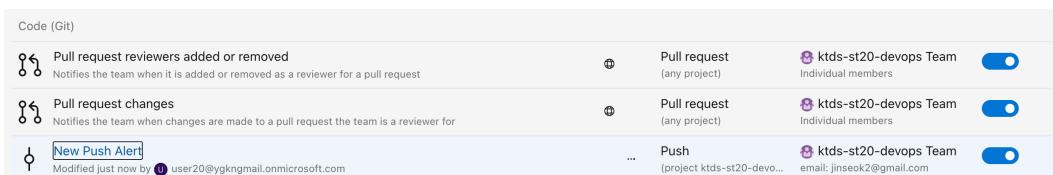
2.6.5 특정 사용자 또는 그룹을 추가할 수 있습니다.

2.6.6 모든 팀 구성원에게 알림을 보내려면 프로젝트 팀을 선택할 수 있습니다.

2.7 설정 완료 및 저장

2.7.1 모든 세부 사항을 입력한 후 **Create Subscription (구독 생성)** 버튼을 클릭합니다.

2.7.2 이제 새로운 소스 코드가 **푸시(push)**될 때, 지정된 팀원에게 이메일을 통해 알림을 받게 됩니다.



2.8 알림 설정 테스트

2.8.1 설정이 제대로 작동하는지 확인하려면, 프로젝트에 새로운 코드를 푸시하거나 커밋한 후 이메일을 수신하는지 확인합니다.

The screenshot shows an email from Azure DevOps. At the top, it displays the commit message: "master updated with 1 commit" followed by "user20@ygkngmail.onmicrosoft.com pushed new changes". Below this, there's a link to "View push". The main body of the email shows two sections: "Files" (with one item: "src/main/java/com/wiken/e2edemo/HelloWorldController.java") and "Commits" (with one item: "9143e01e: 4th commit. Mail Alert testing").

2.9 추가 설정 (선택 사항)

2.9.1 **Service Hooks**를 사용하여 더 고급 알림을 설정할 수 있습니다. 예를 들어, 다른 통합 서비스(Slack, Microsoft Teams 등)와 연동도 가능합니다.

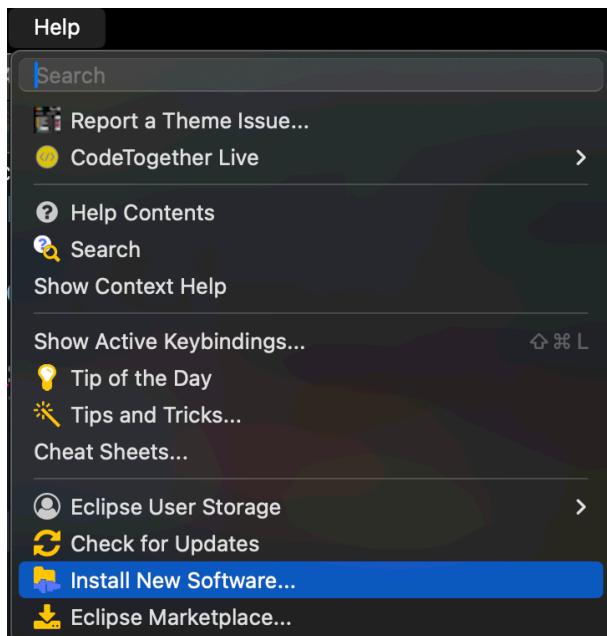
3. Team Explorer Everywhere plugin 사용해서 repos 연결

리포지토리가 생성되면 Git URL을 제공받습니다. 이 URL은 로컬 Git 리포지토리와 연결하는 데 사용됩니다. 제공한 TFSEclipsePlugin-updateSiteArchive-14.zip 설치해서 Eclipse에 **Team Explorer Everywhere (TEE)** 플러그인을 수동으로 설치합니다.

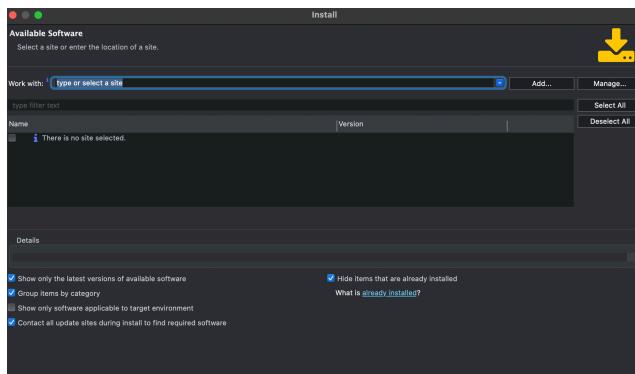
3.1 Spring Tool Suite(STS)에서 Git 연결

Azure Repos 와 Spring Tool Suite 4를 연결하려면 로컬 프로젝트를 Git과 연동해야 합니다.

3.2 로컬 아카이브 추가



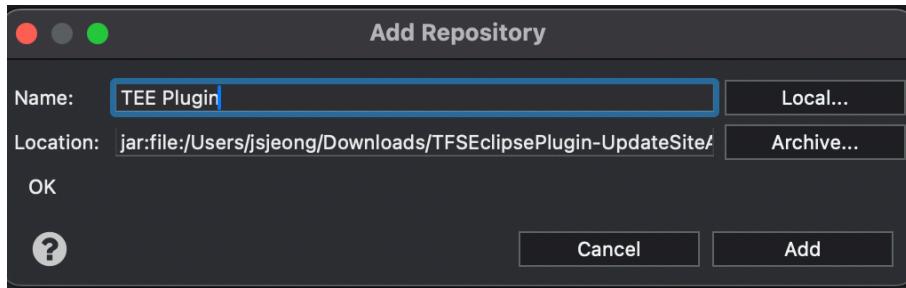
3.2.1 Install New Software 창에서 Add 버튼을 클릭합니다.



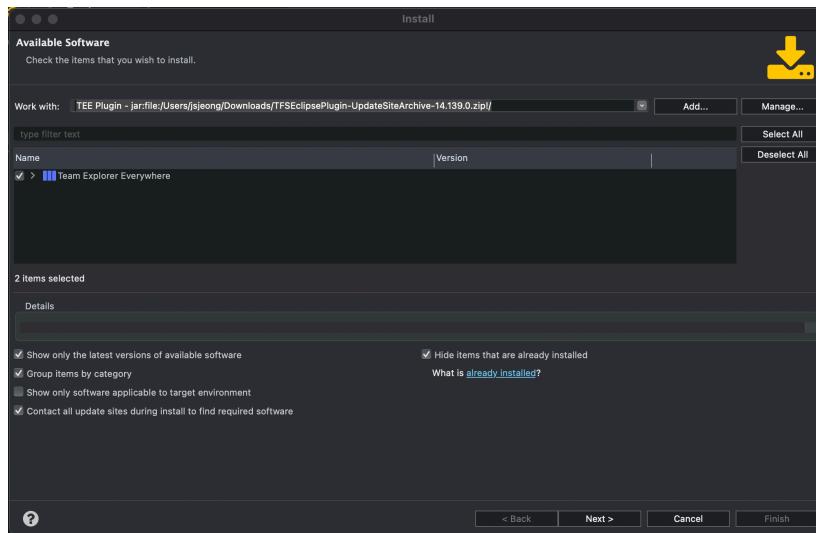
3.2.2 Archive 버튼을 선택하여, 로컬 TEE 위치를 선택합니다.

3.2.3 Add Repository 창이 나타나면, 아래 단계를 따릅니다:

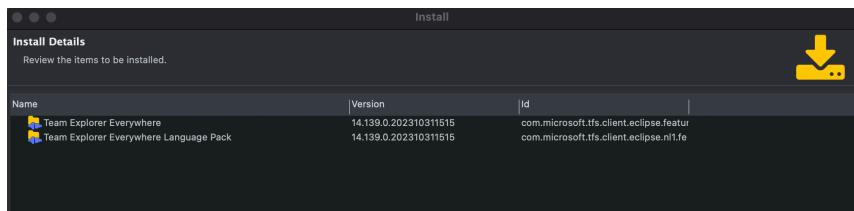
- **Name:** 적절한 이름을 입력합니다. 예: TEE Plugin
- **Location:** 오른쪽에 있는 Archive 버튼을 클릭하여, 다운로드한 TFSEclipsePlugin.zip 파일을 선택합니다.



3.2.4 Add 버튼을 클릭하면, Name에 항목에 Team Explorer Everywhere 를 선택하고, next

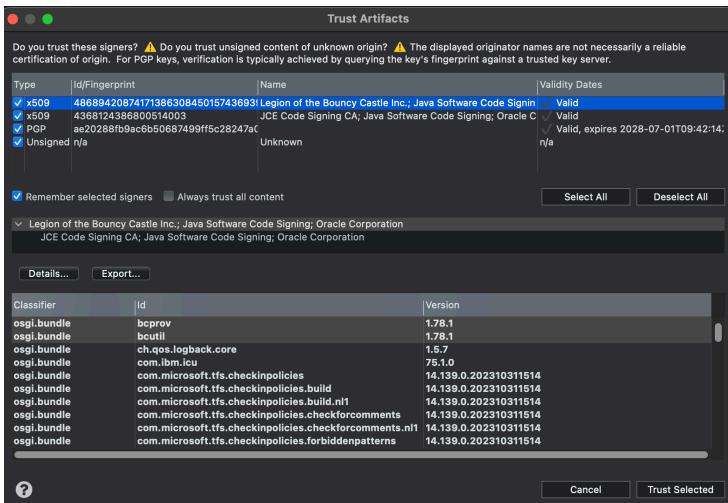


3.2.5 Install details 에 설치될 내용이 리스트 되고, Next

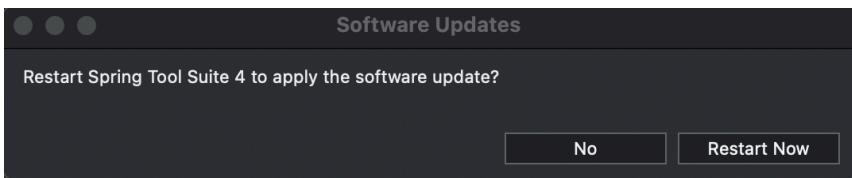


3.2.6 Review Licenses 동의하고 finish 를 선택하면 설치 작업이 진행됨. (하단에 표시)

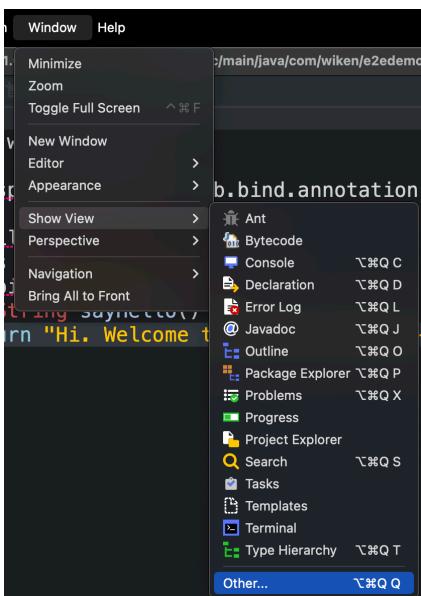
설치 중에 Trust Artifacts 팝업 창이 나타나면 전체 선택하고 “Trust Selected”



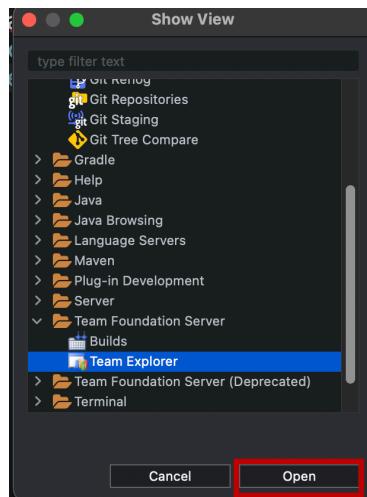
3.2.7 선택 후에 Eclipse 를 다시 시작하여 변경 사항을 적용 할 수 있는 팝업이 나타나고, Restart Now 를 선택한다



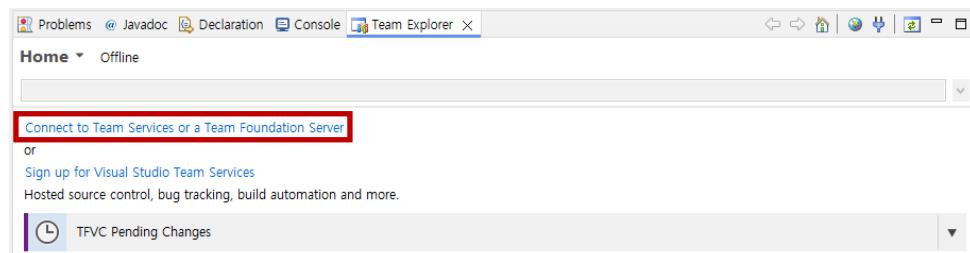
3.2.8 Eclipse 가 다시 시작되면 Windows > Show View 표시로 이동하여 Other...를 선택한다.



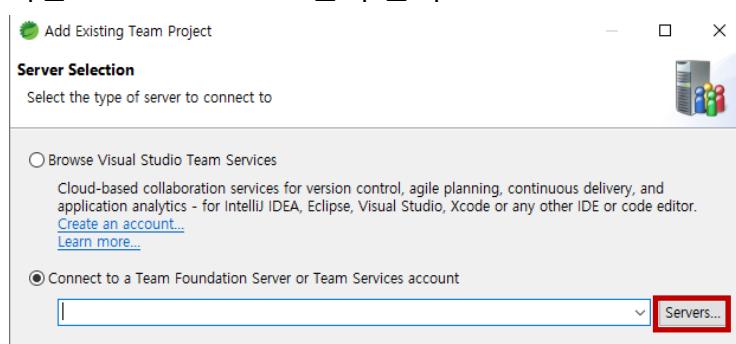
3.2.9 Team Foundation Server 를 검색하고 Team Explorer 보기 를 선택한 다음 Open 을 클릭한다.

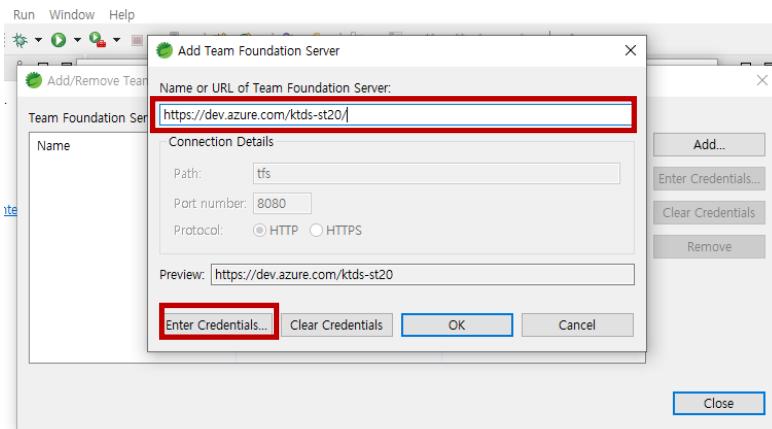


3.2.10 Team Explorer 보기에서 Connect to a Team Foundation Server or Team Services account 할 링크를 클릭한 다음 Azure DevOps 조직(<https://dev.azure.com/>**your-organization-name**)의 이름을 입력하고 Next 버튼을 누른다.

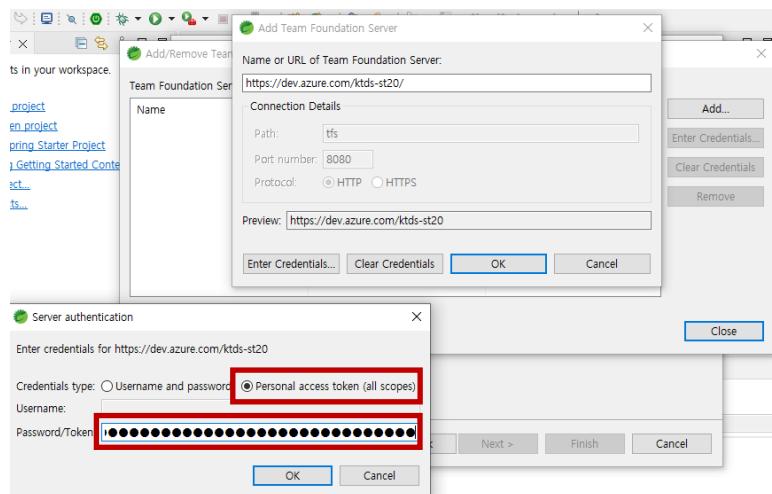


3.2.11 “Connect to a Team Foundation Server or Team Services account” 옆 “Servers...” 버튼을 누른다. “Add/Remove Team Foundation Server” 패널에서 “Add...”를 누르고 “Add Team Foundation Server” 패널에 Azure DevOps 조직의 이름을 입력한다. 그런 다음 Enter Credentials 를 누른다.

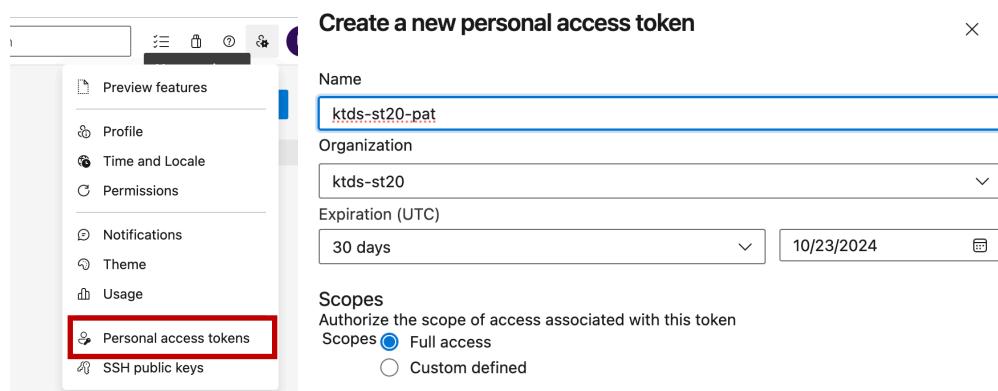




3.2.12 “Server”를 선택하여 Team Foundation Server 정보를 입력하고, “Enter Credentials” 서버 인증을 위한 PAT(1.2.13에서 생성)를 입력한다.

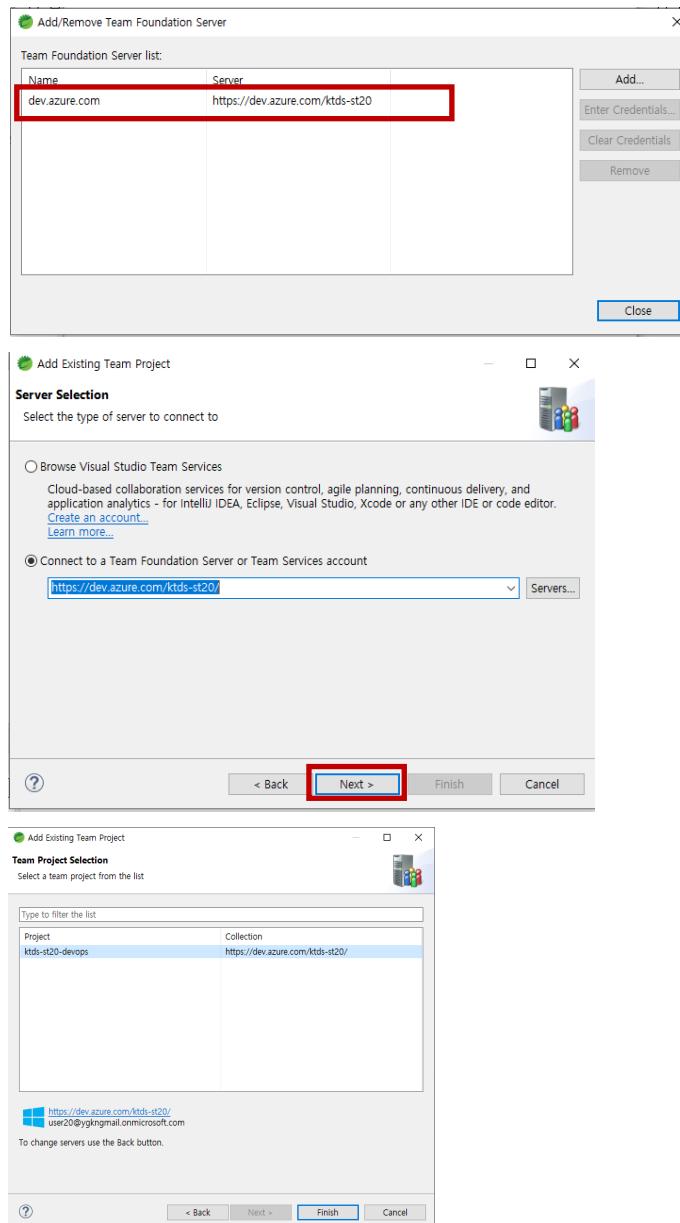


3.2.13 Azure DevOps Server에서 **Personal Access Token (PAT)**을 생성하는 방법은 DevOps 서버 페이지 우측 상단 “user setting” 설정의 Personal access tokens 을 클릭



- Name=ktds-st##-pat 로 하여 Full access 선택하여 PAT를 생성한다. PAT가 생성되면 시스템에서 해당 토큰 값을 표시합니다. 이때, 해당 토큰을 반드시 복사해 둡니다. 한 번 생성된 PAT는 나중에 다시 확인할 수 없으므로, 안전한 장소에 저장해 두어야 합니다.
- 복사를 못한 경우에는 revoke 하고 다시 생성할 수 있다.

3.2.14 Eclipse로 돌아가서 팝업 창에서 확인 버튼을 누른다. 이제 Azure DevOps 조직이 연결할 서버 목록에 표시되어야 한다. 현재 창을 닫으려면 "Close"버튼을 누른 다음 "Next" 버튼을 눌러 서버에 연결한다.



3.2.15 설정이 완료 되면 Repos 리포지토리와 연결하여 로컬에서 코드 변경을 진행한 후에 원격 서버에도 Repos 을 확인 할 수 있다.

