**Configuration Management**

**ServerManager Guide for Git**

**스마트워크 2, 3단계 구축\_형상관리\_가이드**

****Version 1.0

2016.09.05

작성자 : 문해민



**Table of Contents**

[**1.** **개요** - 2 -](#_Toc462142702)

[**1.1.** **한화 S&C 형상관리 서버** - 3 -](#_Toc462142703)

[**2.** **형상관리 서버 접속** - 3 -](#_Toc462142704)

[**2.1.** **Putty 설치** - 4 -](#_Toc462142705)

[**2.2.** **SSH 원격 접속** - 5 -](#_Toc462142706)

[**3.** **서버 관리하기** - 6 -](#_Toc462142707)

[**3.1.** **Git 저장소 생성** - 6 -](#_Toc462142708)

[**3.1.1.** **최초 저장소 생성** - 6 -](#_Toc462142709)

[**3.1.2.** **기존 저장소 복제** - 7 -](#_Toc462142710)

[**3.1.3.** **스마트HTTP에 저장소 인증 등록** - 7 -](#_Toc462142711)

[**3.2.** **Git 사용자 계정 관리 [ ncp, ncp\_workplace ]** - 9 -](#_Toc462142712)

[**3.2.1.** **Git 사용자 생성 및 추가** - 10 -](#_Toc462142713)

[**3.2.2.** **Git 사용자 삭제** - 13 -](#_Toc462142714)

[**3.3.** **작업 이련 관리** - 13 -](#_Toc462142715)

1. **개요**

**Abstract**

본 문서는 형상관리 서버를 관리하기 위한 초기 설정 및 방법들을 설명합니다. 형상관리 서버안에 존재하는 Git 서버에서는 다양한 툴과 설정 등을 통해 디버깅, 코드 인스펙션, 로그를 통한 충돌 감지 등 여러가지 작업들이 가능합니다. 하지만 본 문서에서는 **‘동시다발적인 협업, 인터넷 환경이나 장소에 제약 받지 않는 작업 환경 구축’**에 초점을 맞춰 **‘Git 사용자 인증 및 관리에 대한 전반적인 가이드에 범위’**를 두고 있습니다.

* 1. **한화 S&C 형상관리 서버**

Git은 다양한 프로토콜을 제공하기 때문에 Git 기반 형상관리 서버에는 다양한 사용자 접근 제어 방법을 적용할 수 있습니다. 각 프로젝트의 특성과 팀의 상황을 고려하여 ‘느슨한 제어부터 세밀한 제어’까지 다양한 방식이 존재합니다. 하지만 한화 S&C의 형상관리 서버에서는 스마트HTTP 프로토콜을 사용하여 ‘Git 사용자 관리 환경’이 구축되어 있습니다.

**스마트HTTP 사용 이유 ( 장점 )**

* 읽기와 쓰기에 하나의 URL만 사용함
* 사용자에게 익숙한 아이디와 비밀번호 방식의 인증을 사용함
* 사용자 측면에서, SSH 인증에 비해 인증 방식이 간단함 [SSH 인증 방식은 사용자가 키를 만들고 공개키를 서버에서 올려야함]
* SSH 만큼 빠른 속도를 보장함
* HTTP대신 HTTPS를 이용하면 전송 데이터를 암호화하거나 클라이언트에게 서명된 SSL 인증서를 요구하는 것도 가능함
* HTTP는 보편적인 프로토콜이기 때문에 회사 방화벽에 쉽게 통과할 수 있음

이러한 이유로 S&C에서는 스마트HTTP를 사용합니다. 즉, 본 문서에서는 스마트HTTP 프로토콜을 기반으로 한 사용자 관리 방법에 대해, 소개할 것입니다.

1. **형상관리 서버 접속**

형상관리 서버에 접속하기 위해선 SSH(Secure Shell)를 이용해야 합니다. Putty라는 SSH 접속 프로그램으로 서버에 원격 접속하도록 하겠습니다.

* 1. **Putty 설치**

**Putty Download URL**

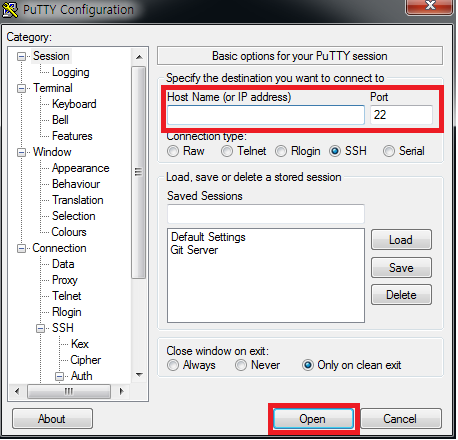
[**http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html**](http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html)

* 해당 URL에 접속하시면 다음과 같은 화면이 보여지게 됩니다.



* 위와 같은 화면에서 본인의 PC환경에 맞는 “putty.exe” 파일을 다운로드 합니다.
  1. **SSH 원격 접속**

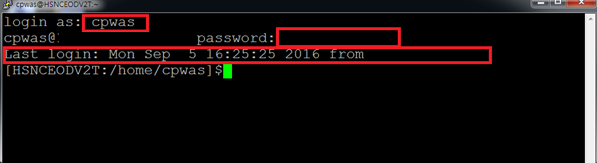
‘putty.exe’ 프로그램을 실행하시면 다음과 같은 화면이 보여집니다.



* Host Name의 Input box에 접속하고자 하는 IP 주소를 기입합니다.

***[S&C 협업포탈 원격 저장소의 경우 xxx.xxx.xxx.xxx / SSH 접속 개방 포트 22]***

* IP 주소 기입 후 Open 버튼을 클릭하면 아래와 같은 화면이 보여집니다.



서버에는 사전에 원격 접속을 허용하기 위해 SSH Key 인증이 완료된 Admin 아이디가 존재하는데, 이를 통해 로그인 절차를 진행하시면 됩니다. 로그인이 완료되면 “Last login : 접속 날짜와 시각 from 본인의 IP” 문구를 확인 하실 수 있습니다.

***[ S&C 협업포탈 “Admin ID / password : 문의 / 문의 ]***

1. **서버 관리하기**
   1. **Git 저장소 생성**

서버의 원격 저장소에는 기본적으로 워킹 디렉토리가 존재하지 않습니다. [ 로컬 저장소와 상이함 ] 그래서 서버에 원격 저장소를 생성하거나 기존 저장소를 복제할 때 명령어에 옵션을 추가해야 합니다.

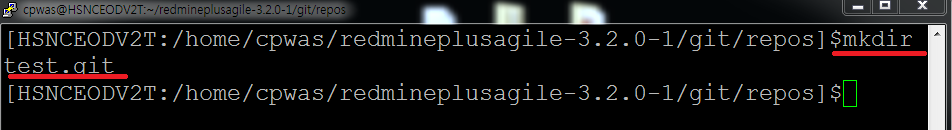
**Option 명령 설명**

--bare는 워킹디렉토리가 없는 저장소를 생성하는 옵션

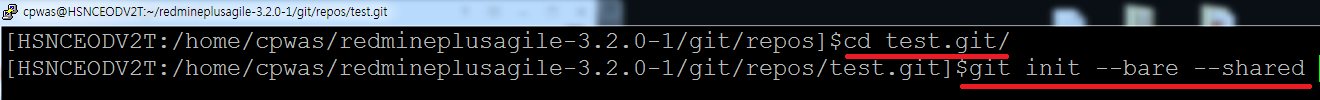
--share는 자동으로 그룹쓰기 권한을 추가하는 옵션

**위와 같은 옵션들을 명령어와 함께 입력하시면 됩니다.**

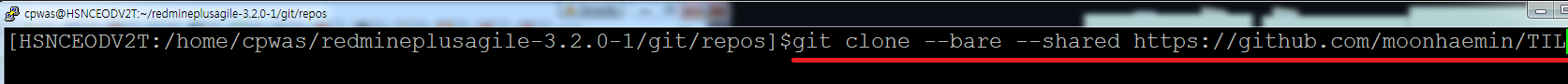
* cd 명령어를 이용하여 저장소를 생성하고자 하는 디렉토리로 이동 후, **.git** 형태로 디렉토리를 생성합니다. **[ 기본적으로 원격 저장소 생성시 .git 규약을 따라야 함 ]**



* + 1. **최초 저장소 생성**
* cd명령어를 이용하여 본인이 생성한 test.git 디렉토리로 이동 후, **git init --bare --shared** 명령어를 입력하여 최초 저장소를 생성합니다.



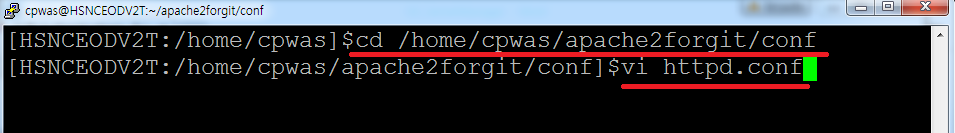
* + 1. **기존 저장소 복제**
* cd명령어를 이용하여 본인이 생성한 디렉토리로 이동한 후, **git clone --bare --shared** **URL** 을 입력하여 기존의 저장소를 복제합니다.

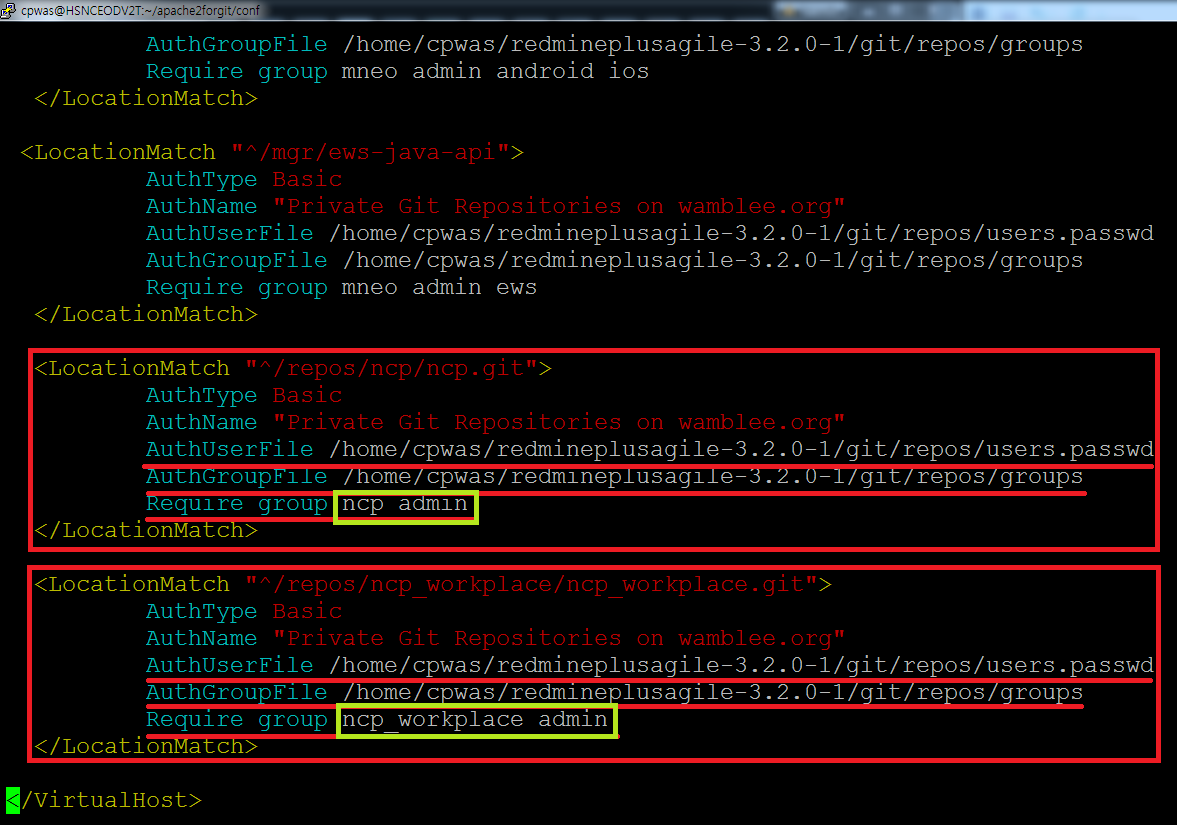


* + 1. **스마트HTTP에 저장소 인증 등록**

위 과정까지 진행하시면, 저장소 생성에는 성공하였지만 아직 생성한 원격 저장소가 서버에 등록이 되지 않았기 때문에 다른 Git 사용자들이 접근할 수 없는 상태입니다. 서버에 원격 저장소를 등록하는 과정을 진행해 보도록 하겠습니다.

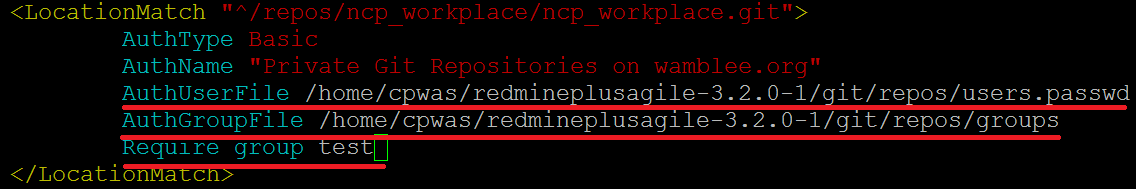
* **cd /home/cpwas/apache2forgit/conf** 명령어를 활용하여 디렉토리를 이동 후, **httpd.conf** 파일을 열어줍니다.

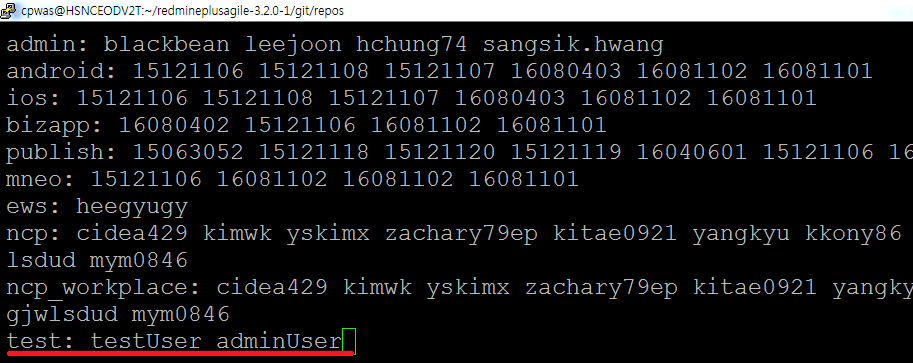




**위 설정 파일의 지시어를 살펴보도록 하겠습니다.**

* **AuthType** 지시어는 사용자를 인증할 방법을 선택합니다. 가장 일반적인 방법은 Basic인증 입니다. Basic 인증은 브라우저가 서버로 암호를 암호화하지 않고 보내기 때문에, 기밀 자료를 보호하기 위해서는 Basic 인증방식을 추천하지 않습니다.
* **AuthName** 지시어는 인증에 사용할 영역을 지정합니다. 영역은 두 가지 역할을 하는데, 첫 번째는 클라이언트가 보통 이 정보를 암호 대화창에 보여줍니다. 두 번째는 영역 정보를 사용하여 클라이언트가 특정 인증구역에 어떤 암호를 보낼지 결정합니다.
* **AuthUserFile** 지시어는 htpasswd로 만든 암호파일의 경로를 설정합니다. 사용자가 많다면 요청마다 매번 사용자를 인증하기 위해 일반 문서파일을 검색하는데 시간이 상당이 많이 걸릴 수 있기에, 아파치의 DB모듈을 활용하기도 합니다. [S&C는 일반 문서파일을 사용함]
* **Require** 지시어는 서버의 특정 영역에 접근할 수 있는 사용자를 지정하여 권한을 부여합니다. 다음 Chapter에서 require 지시어를 활용하여 Git 사용자 계정을 관리하는 방법에 대해 설명합니다.
* 즉, 위와 같은 형태로 설정파일에 새로 추가한 원격 저장소를 추가해주시고, **cd /home/cpwas/redmineplusagile-3.2.0-1/git/repos** 이동 후 **groups** 파일을 오픈하여 원격 저장소의 계정 그룹을 추가시켜 줍니다.





위의 작업내용을 풀어 설명하자면, 생성한 원격 저장소를 스마트HTTP에 경로와 함께 등록하고 원격 저장소에 접근 가능한 사용자 그룹을 추가시켜주는 과정입니다. 즉, testUser와 adminUser가 test라는 원격 저장소에 접근할 수 있는 권한을 갖게 되는 것입니다. [ testUser와 adminUser는 계정 생성이 완료됐다는 가정 ]

* 1. **Git 사용자 계정 관리 [ ncp, ncp\_workplace ]**

기본적으로 ‘S&C 협업포탈’ Git 서버에는 “ncp, ncp\_workplace” Repository가 존재합니다.

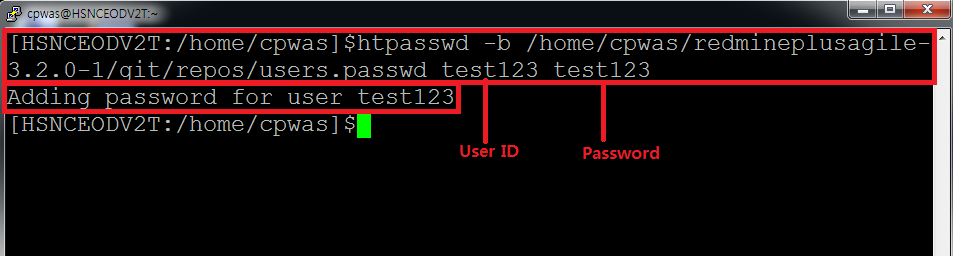
**ncp 프로젝트는 협업포탈 中 워크플레이스를 제외한 모든 기능**

**ncp\_workplace 프로젝트는 협업포탈 中 워크플레이스 기능**

두 개의 프로젝트가 하나의 협업포탈을 구축하게 되는 것 입니다. 서비스하는 것은 하나의 서비스지만 프로젝트가 두 개로 분리되어 있기 때문에 Git 사용자 계정을 부여할 시 두 개의 인증을 필요로 합니다.

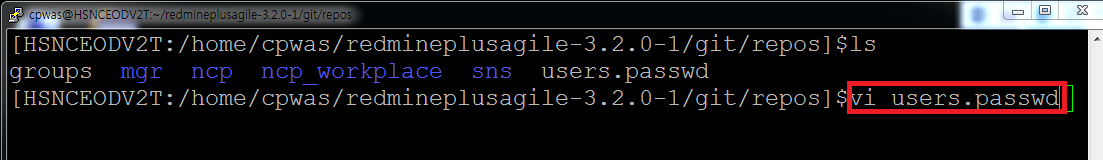
* + 1. **Git 사용자 생성 및 추가**
* 서버에 접속이 완료된 상태에서 다음과 같은 명령어를 입력합니다.

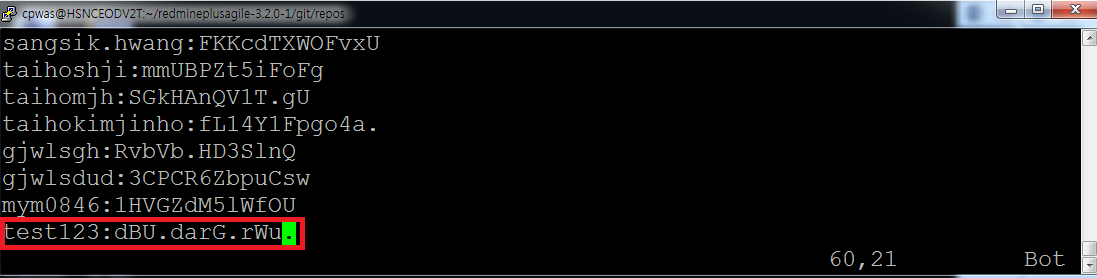
**htpasswd -b /home/cpwas/redmineplusagile-3.2.0-1/git/repos/users.passwd <user> <passwd>**



위 이미지에서는 **“test123/test123”** 이라는 ID/PW로 사용자 계정을 생성하고 있습니다. 계정이 성공적으로 생성되면 **“Adding password for user test123”** 라는 문구를 확인하실 수 있습니다.

계정은 **‘/home/cpwas/redmineplusagile-3.2.0-1/git/repos’** 라는 디렉토리의 **‘users.passwd’** 파일안에 암호화되어 추가되는데, 확인을 위해 cd 명령어를 이용하여 해당 디렉토리로 이동한 후, vi편집기로 파일을 열어보도록 하겠습니다.





위 이미지와 같이 htpass 명령어를 사용해 생성한 test123 계정이 users.passwd 파일안에 암호화되어 추가된 것을 확인 하실 수 있습니다.

**users.passwd 파일은 암호화된 파일인데, 사전에 서버를 설정할 때**

**htpasswd –c /home/cpwas/redmineplusagile-3.2.0-1/git/repos/users.passwd rbowen**

**명령어를 활용하여 암호화 파일을 미리 생성해 놓은 것입니다.**

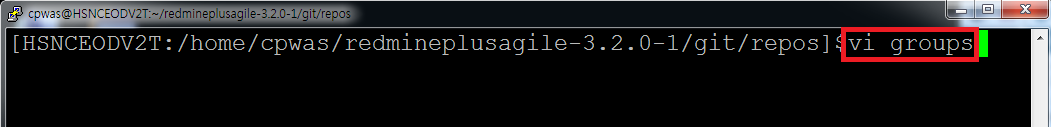
계정 생성은 완료되었으나, 아직 Repository group에 생성된 계정을 추가해주지 않았기 때문에 현재 상태에서는 사용자들이 자신의 로컬 저장소에 원격 저장소를 복제해 올 수 없는 상태입니다.

Repository group에 생성한 계정을 추가하기 위해서는 다음과 같은 작업을 수행해야 합니다.

* ‘/home/cpwas/redmineplusagile-3.2.0-1/git/repos’ 디렉토리로 이동한 후, ls 명령어로 디렉토리를 조회하시면 ‘groups’ 라는 파일이 존재하는 것을 확인 하실 수 있습니다.

C:\Users\한화에스엔씨\Desktop\putty4.PNG

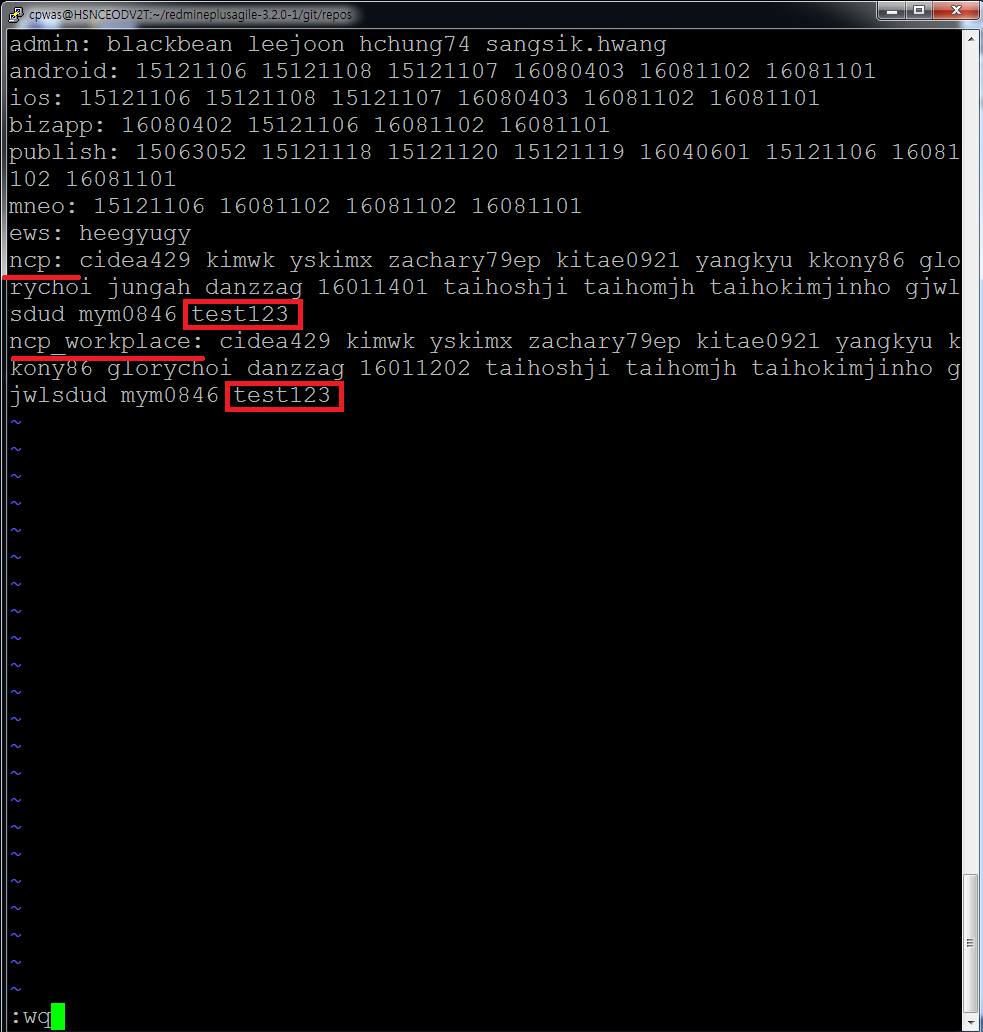
* Vi 편집기를 이용해서 groups 파일 편집을 준비합니다.



* ‘groups’ 파일이 오픈되면, 아래 그림과 같이 파일을 편집하시면 됩니다.

**ncp, ncp\_workplace 라인 모두 space(띄어쓰기) 구분자로 순서대로 기입해주시면 됩니다.**

ex) Group Name : userId1 userId2 userId3



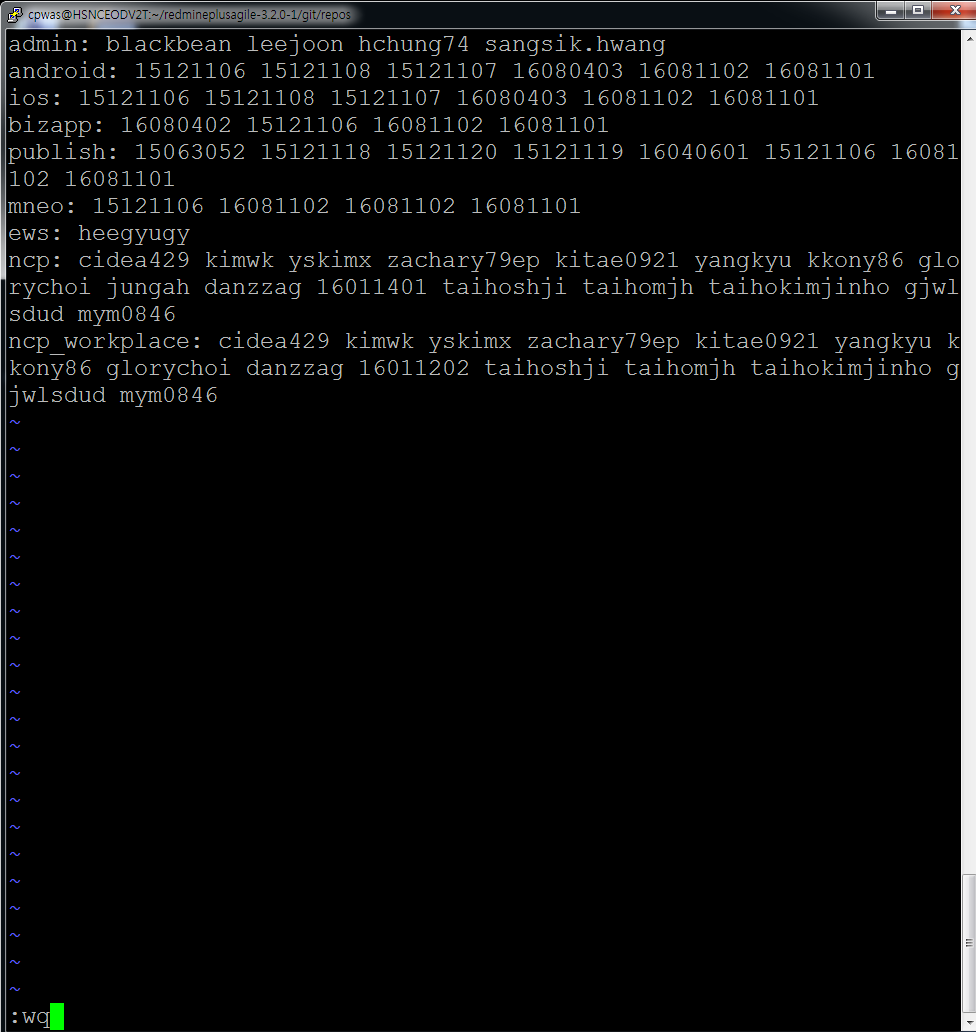
파일 편집을 완료하시고 vi편집기를 종료하시면 **‘Git 사용자 계정 생성 및 인증’** 절차가 모두 완료됩니다. 여기까지의 프로세스가 완료되면 사용자는 원격 저장소에 접근하여 협업할 수 있는 권한을 갖게 됩니다. **[Repository 복제, Push, Pull……]**

* + 1. **Git 사용자 삭제**

사용자 권한 회수는 vi 편집기를 이용해서 사전의 **‘계정 생성 및 인증 추가’** 작업했던 것을 그대로 지워준 후 저장해주시면 됩니다.

**vi users.passwd -> test123:암호화** 부분 삭제

**vi groups -> ncp, ncp\_workplace : ……. test123** 부분 삭제

****

* 1. **작업 이련 관리**

관리자는 서버에 대한 관리능력도 필요로 하지만 서버의 핵심인 ‘Git Repository’에 대한 관리 지식도 필요합니다.

Commit, Branch, Merge 등의 작업들이 Git 사용자들로부터 동시적으로 발생하게 되는데 이때 여러 가지 충돌상황들이 발생할 수 있습니다.

**git log**

**git branch**

**git status**

기본적으로 충돌이 발생하면 위와 같은 로그 조회 명령어를 통해 문제를 인식하고 발단 부분을 검출하여 문제 해결을 신속하게 할 수 있어야 합니다.

**Git Reference URL**

<http://backlogtool.com/git-guide/kr/reference/>

<https://git-scm.com/doc>