

Open Source Product

이 문서는 오픈 소스 (Open Source) 솔루션에 대한 다양한 정보를 공유하기 위해 작성되었습니다.

2018-03-09 20:32:54.01

Jenkins 가이드

이 문서는 Jenkins 가이드를 공유하기 위해 작성되었다.

2018-03-09 20:33:49.372

LDAP 가이드

이 문서는 LDAP 가이드를 공유하기 위해 작성되었다.

2018-03-09 20:34:25.242

Subversion 가이드

이 문서는 Subversion 가이드를 공유하기 위해 작성되었다.

2024-04-11 08:49:37.5

Eclipse Che 가이드

이 문서는 eclipse che 가이드를 공유하기 위해 작성되었다.

2024-04-09 18:49:15.743

Gradle 가이드

2018-02-05 14:20:20.341

Open Source Knowledge Base

2024-03-07 08:57:02.167

Selenium 가이드

이 문서는 Selenium 을 이용한 웹 테스트 자동화 방법을 정리한다. 개요 Selenium 은 웹 브라우저를 자동화하기 위한 강력하고 널리 사용되는 오픈 소스 프레임워크입니다. 웹 페이지와 상호 작용하기 위한 테스트 스크립트를 작성하고, 사용자 작업을 시뮬레이션하며, 다양한 브라우저와 플랫폼에서 웹 애플리케이션의 동작을 검증할 수 있는 도구와 라이브러리를 제공한다. Selenium은 Java, C#, Python, Ruby, JavaScript를 포함한 다양한 프로그래밍 언어를 지원하여 웹 애플리케이션 테스트에 있어 다재다능한 선택지를 제공한다. 제공 기능 다양한 언어 지원: Selenium은 Java, C#, Python, Ruby, JavaScript 등 여러 프로그래밍 언어를 지원하여 다양한 개발 환경에서의 사용을 가능하게 한다. 크로스 브라우저 테스트: Selenium WebDriver를 통해 Chrome, Firefox, Internet Explorer, Safari 등 다양한 웹 브라우저에서 웹 애플리케이션을 자동으로 테스트할 수 있다. 웹 애플리케이션 자동화: 사용자의 웹 브라우징 활동을 모방하여 웹 페이지와의 상호 작용을 자동화할 수 있습니다. 이는 폼 제출, 마우스 클릭, 데이터 입력 및 추출 등을 포함한다. Selenium IDE: 브라우저 확장 프로그램으로 제공되며, 코드 작성 없이 테스트 케이스를 기록하고 재생할 수 있는 기능을 제공합니다. 이를 통해 비개발자도 쉽게 테스트 스크립트를 생성할 수 있다. Selenium Grid: 여러 시스템과 브라우저에서 동시에 테스트를 실행할 수 있게 해주어, 대규모 테스트 실행과 분산 테스트 가능한다. 이는 테스트의 실행 시간을 단축시키고 효율성을 높여 준다. 오픈 소스: Selenium은 오픈 소스 소프트웨어로, 무료로 사용할 수 있으며, 활발한 커뮤니티의 지원을 받아 지속적으로 개선되고 있다. Selenium 테스트 자동화 구성 요소 Selenium은 여러 컴포넌트로 구성됩니다. Selenium WebDriver: 직접 브라우저 자동화와 상호 작용을 허용하는 핵심 컴포넌트입니다. 브라우저에 명령을 보내고 결과를 검색하여 웹 애플리케이션의 자동화를 가능하게 한다. Selenium IDE (Integrated Development Environment): 코드 작성 없이 상호 작용을 녹화 및 재생할 수 있는 Chrome과 Firefox 확장 프로그램으로, 테스트 스크립트를 빠르게 생성하는 데 유용하다. Selenium Grid: 다양한 머신과 브라우저에서 동시에 Selenium 테스트를 실행할 수 있게 하여 분산 테스트 실행을 가능하게 한다. <https://browserstack.wptestenginepowered.com/wp-content/uploads/2021/09/Selenium-WebDriver-for-Automation-Testing.webp> Java Language 예시 사전 준비 사항 Java Development Kit (JDK): 시스템에 JDK가 설치되어 있는지 확인하세요. 공식 Oracle 웹사이트 또는 다른 JDK 공급업체에서 다운로드할 수 있다. Integrated Development Environment (IDE): Eclipse, IntelliJ IDEA, NetBeans와 같은 IDE입니다. 이 가이드는 Eclipse를 사용한다고 가정한다. Maven: 프로젝트의 빌드와 의존성을 관리하는 프로젝트 관리 도구입니다. Eclipse와 IntelliJ IDEA에 통합되어 있다. WebDriver: 테스트하고자 하는 브라우저용 WebDriver 실행 파일을 다운로드해야 한다(예: Google Chrome용 ChromeDriver, Firefox용 geckodriver). Selenium 테스트 프로젝트 생성 단계 Maven 프로젝트 생성: Eclipse를 열고 파일 > 새로 만들기 > Maven 프로젝트를 선택한다. 새 Maven 프로젝트를 생성하기 위한 마법사를 따른다. 기본 아키텍처를 사용할 수 있다. Selenium 의존성 추가: pom.xml 파일을 편집하여 Selenium 의존성을 포함시킵니다. <dependencies> 섹션 내에 다음을 추가한다: LATEST_VERSION을 사용하고자 하는 최신 Selenium 버전으로 교체한다. <dependency> <groupId>org.seleniumhq.selenium</groupId> <artifactId>selenium-java</artifactId> <version>LATEST_VERSION</version> </dependency> 스트 클래스 작성: Java를 사용하여 웹 페이지를 열고, 제목을 검증하는 간단한 테스트 케이스를 작성합니다. import org.openqa.selenium.WebDriver; import org.openqa.selenium.remote.DesiredCapabilities; import org.openqa.selenium.remote.RemoteWebDriver; import org.junit.Assert; import org.junit.After; import org.junit.Before; import org.junit.Test; import java.net.URL; public class GridTitleTest { private WebDriver driver; @Before public void setUp() throws Exception { // 원하는 브라우저 구성을 설정합니다. DesiredCapabilities capabilities = DesiredCapabilities.chrome(); // 여기에서 chrome을 다른 브라우저로 변경할 수 있습니다. // Selenium Grid Hub URL을 설정합니다. 이 URL은 Selenium Grid Hub에 대한 실제 URL로 변경해야 합니다. URL hubUrl = new URL("http://localhost:4444/wd/hub"); // 원격 WebDriver 인스턴스를 생성합니다. driver = new RemoteWebDriver(hubUrl, capabilities); } @Test public void testWebTitle() { // 테스트할 웹 페이지를 엽니다. driver.get("http://www.example.com"); // 페이지 제목을 가져와서 검증합니다. String pageTitle = driver.getTitle(); Assert.assertEquals("Title of this web page", pageTitle); } @After public void tearDown() { // 테스트 후에 브라우저를 닫습니다. if (driver != null) { driver.quit(); } } 설정(Setup): setUp 메서드에서는 원격 WebDriver 인스턴스를 생성합니다. 이때, Selenium Grid Hub의 URL과 원하는 브라우저 구성(예: Chrome)을 지정한다. 테스트 실행(Test): testWebTitle 메서드에서는 웹 페이지를 열고, 그 제목이 "Title of this web page"와 일치하는지 검증한다. 정리(TearDown): tearDown 메서드에서는 테스트가 완료된 후 브라우저를 닫는다. WebDriver 설정: 테스트 코드에서 시스템 속성을 설정하여 다운로드한 WebDriver 실행 파일의 경로를 지정한다. 테스트 실행: Eclipse에서 작성한 테스트를 실행하고, 웹 페이지가 예상대로 동작하는지 확인한다. UI 요소 찾는 방법 Selenium은 다양한

방식으로 UI 구성 요소를 접근하는 방법을 제공한다. Id Links Tag names CSS Selectors XPath JQuery Text 예시) driver.findElement(By.id(<elementID>)) driver.findElement(By.linkText(<linktext>)) Id driver.findElement(By.id("name of id")) UI 요소를 찾는 가장 이상적인 솔루션 ID가 존재하지 않을 수 있음 복수의 Id가 존재 할 수 있음 Links driver.findElement(By.linkText("name of link")) driver.findElement(By.partialLinkText("name of link")) 링크에 표시된 텍스트로 요소 찾기 Tag driver.findElement(By.tagName("name of HTML tag")) 태그 이름 HTML 태그를 기반으로 웹 요소 찾기 CSS Selectors Absolute path: driver.findElement(By.cssSelector("html>body>div>p>input")) Relative path: driver.findElement(By.cssSelector("input")) *first instance found Regular attribute: driver.findElement(By.cssSelector("button[name='button name']")) Special attributes: id: driver.findElement(By.cssSelector("#save")); tag & id: driver.findElement(By.cssSelector("button#save")); class attribute: driver.findElement(By.cssSelector(".yoyo")); tag & class attribute: driver.findElement(By.cssSelector("input.username")); tag with attribute value: driver.findElement(By.cssSelector("img[alt=kuku]")); tag which has attribute: driver.findElement(By.cssSelector("img[alt]")); tag which doesn't have attribute: driver.findElement(By.cssSelector("img:not([alt])")); XPath /html/body/div[5]/div[2]/div[2]/div[2]/h2[1] ./*[@id='answers']/h2[1]/a[1] XPath는 XML에서 노드를 찾는 데 사용되는 방법임 찾으려는 요소에 적합한 ID 또는 이름 속성이 없는 경우에 적합 절대 XPath 보다 상대 XPath 사용 권장 XPath-Attributes는 ID나 이름을 사용할 수 없을 때 사용 권장 Syntax: htmltag [@attribute1='value1' and @attribute2='value2'] Example: input@id='Password' and @placeholder='Password'] WebDriver API 목록 driver.get("http://www.google.com"); To open an application driver.findElement(By.id("passwd-id")); Finding Element using Id driver.findElement(By.name("passwd")); Finding Element using Name driver.findElement(By.xpath("//input[@id='passwd-id']")); Finding Element using Xpath element.sendKeys("some text"); To type some data element.clear(); clear the contents of a text field or text area driver.findElement(By.xpath("//select")); Selecting the value driver.findElement(By.id("submit")).click(); To click on Any button/Link driver.switchTo().window("window Name"); Moving from one window to another window driver.switchTo().frame("frame Name"); swing from frame to frame (or into iframes) driver.get("http://www.google.com"); To open an application driver.switchTo().frame("frameName.0.child"); to access sub frames by separating the path with a dot, and you can specify the frame by its index too driver.switchTo().alert(); Handling Alerts driver.Navigate().to("http://www.example.com"); To Navigate Particular URL driver.Navigate().forward(); To Navigate Forward driver.Navigate().back(); To Navigate Backward driver.close(); Closes the current window driver.Quit(); Quits the driver and closes every associated window driver.switch_to_alert(); Switches focus to an alert on the page. driver.Refresh(); Refreshes the current page driver.get_screenshot_as_file('/Screenshots/foo.png'); The full path you wish to save your screenshot to driver.get_screenshot_as_base64(); Gets the screenshot of the current window as a base64 encoded string select.findElements(By.tagName("option")); Selecting the value select.deselectAll(); This will deselect all OPTIONS from the first SELECT on the page select.selectByVisibleText("Edam"); select the OPTION with the displayed text of "Edam"

2024-02-06 08:14:00.451

Eclipse

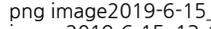
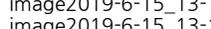
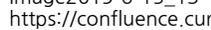
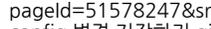
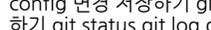
이 문서는 Eclipse 사용에 대한 가이드를 공유하기 위해 작성되었다.

2018-03-09 20:31:11.076

Redmine 가이드

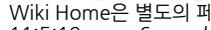
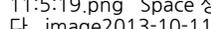
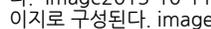
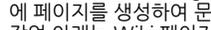
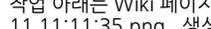
2020-04-22 12:26:15.27

Git 가이드

이 문서는 Git 가이드를 공유하기 위해 작성되었다.       1. Git 시작하기 Git 저장소 <https://confluence.curvc.com/pages/viewpage.action?pagelD=51578247&src=contextnavpagetreemode> git init git clone git config 변경 저장하기 git add git commit git diff git stash .gitignore 저장소 점검하기 git status git log git tag git blame 변경 취소하기 git 실행 취소 git clean git revert git reset git rm Rewriting history <https://confluence.curvc.com/display/ASD/Rewriting+history?src=contextnavpagetreemode> git commit --amend <https://confluence.curvc.com/display/ASD/git+commit+---amend?src=contextnavpagetreemode> git rebase <https://confluence.curvc.com/display/ASD/git+rebase?src=contextnavpagetreemode> git reflog <https://confluence.curvc.com/display/ASD/git+reflog?src=contextnavpagetreemode> 2. Git 협업하기 동기화하기 git remote <https://confluence.curvc.com/display/ASD/git+remote?src=contextnavpagetreemode> git fetch <https://confluence.curvc.com/display/ASD/git+fetch?src=contextnavpagetreemode> git pull <https://confluence.curvc.com/display/ASD/git+pull?src=contextnavpagetreemode> git push <https://confluence.curvc.com/display/ASD/git+push?src=contextnavpagetreemode> 브랜치 사용하기 git branch <https://confluence.curvc.com/display/ASD/git+branch?src=contextnavpagetreemode> git checkout <https://confluence.curvc.com/display/ASD/git+checkout?src=contextnavpagetreemode> git merge <https://confluence.curvc.com/display/ASD/git+merge?src=contextnavpagetreemode> 병합 충돌 해결하기 (Merge conflicts) <https://confluence.curvc.com/pages/viewpage.action?pagelD=51578393&src=contextnavpagetreemode> 병합 전략 (Merge strategies) Pull request 만들기

2018-09-19 11:41:39.967

XWiki 소개

이 페이지는 xWiki의 간단한 화면 덤프를 공유하기 위해 작성되었다. Wiki Home Wiki Home은 별도의 페이지이고 아래와 같이 구성할 수 있다.  Space 생성 문서를 정리하기 위해 먼저 Space(Project)를 생성한다.  스페이스 대시보드 스페이스 대시보드는 페이지로 구성된다.  페이지 생성 스페이스 내부에 페이지를 생성하여 문서를 작성한다.  페이지 작업 아래는 Wiki 페이지 작성을 위한 에디터의 예시를 보여준다.  생성된 페이지 다음은 생성된 페이지의 예시를 보여준다. 

2018-03-09 20:33:22.775

JMeter 가이드

이 문서는 JMeter 가이드를 공유하기 위해 작성되었다. Apache JMeter는 오픈 소스 소프트웨어로 기능 행위의 로드 테스트와 성능 측정을 위해 설계된 순수 Java 어플리케이션입니다. JMeter는 웹어플리케이션을 위해 설계되었지만 다른 테스트 기능으로 확장할 수 있습니다. Apache JMeter는 정적 및 동적 자원, 웹 동적 어플리케이션 모두에서 성능을 테스트하는 데 사용될 수 있습니다. 서버, 서버 그룹, 네트워크 또는 개체의 과부하를 시뮬레이션하여 강도를 테스트하거나 다양한 로드 유형에서 전반적인 성능을 분석하는데 사용할 수 있습니다. Apache JMeter 기능은 다음과 같다. 다양한 어플리케이션/서버/프로토콜 타입을 로드하고 성능 테스트할 수 있는 기능 Web - HTTP, HTTPS(Java, NodeJS, PHP, ASP.NET...) SOAP / REST Webservices FTP JDBC를 통한 데이터베이스 LDAP JMS를 통한 메시지 기반 미들웨어 메일 - SMTP(S), POP3(S), IMAP(S) 네이티브 명령 혹은 쉘 스크립트 TCP Java 오브젝트 빠른 테스트 플랜 레코딩(브라우저 혹은 네이티브 어플리케이션), 빌드, 디버깅이 지원되는 완전한 기능의 테스트 IDE 모든 Java 호환 OS(Linux, Windows, Mac OSX 등)에서 테스트를 로드하는 코멘드 라인 모드 완전하고 준비된 동적 HTML 보고서 HTML, JSON, XML 혹은 어떤 텍스트 포맷, 가장 인기있는 응답 포맷으로부터 데이터 추출 능력 풀 멀티 쓰레딩 프레임워크 많은 쓰레드로부터 병렬 샘플링 분리된 쓰레드 그룹으로 다른 기능의 동시 샘플링 캐싱과 테스트 결과의 오프라인 분석/반복수행 고도로 확장 가능한 코어 플러그인 샘플러는 무제한 테스트 수 용력을 허용 Groovy와 BeanShell과 같은 스크립트가능한 샘플러 플러그인 터미널내에 선택가능한 로드 통계 데이터 분석과 비주얼라이제이션 플러그인 동적 입력을 제공하는데 사용되는 기능 쉬운 지속적인 통합 라이브러리

2022-04-13 13:33:07.948

CentOS 가이드

이 문서는 CentOS Linux 관련 가이드를 공유하기 위해 작성되었다.

2021-03-02 17:05:18.64

GitLab 가이드

2020-10-20 15:14:23.849

Nginx 가이드

이 문서는 Nginx 가이드를 공유하기 위해 작성되었다.

2020-09-10 10:12:22.557

Open Source 설치가이드

이 문서는 개발환경과 관련된 가이드를 공유하기 위해 작성되었다.

2021-08-24 13:33:46.288

Apache Httpd

이 문서는 Apache httpd 웹 서버에 대한 가이드를 공유하기 위해 작성되었다.

2018-03-09 20:30:37.741

Docker 가이드

이 문서는 Docker 사용에 대한 가이드를 공유하기 위해 작성되었다.

2022-04-14 15:00:22.132

쿠버네티스(Kubernetes) 가이드

이 문서는 쿠버네티스(Kubernetes - k8s) 가이드를 공유하기 위해 작성되었다.

2018-03-09 20:31:26.865

Gerrit 가이드

2017-04-04 20:57:47.163

Jenkins & JBoss Integration - Publish Over SSH Plugin

이 페이지는 Jenkins의 Publish over SSH plugin 을 이용해 빌드된 산출물을 원격 JBoss 서버에 배포하는 방법을 정리한다. 사전 준비 Publish over SSH plugin 설치 System Configuration Manage Jenkins >> System Configuration >> Publish over SSH SSH 서버 등록 Screen Shot 2017-04-04 at 8.51.19 PM.png Job Configuration Job configuration >> Post Steps 등록된 SSH 서버에 target/*.war 배포하는 구성 예) base-1.0.0-SNAPSHOT.war 파일 이름을 배포 파일로 변경하기 위해 shell 사용 target directory name 제거 Exec command란에 SSH login 후 실행할 명령어 기입 Screen Shot 2017-04-04 at 8.53.56 PM.png

2018-01-19 12:11:17.691

Jenkins & JBoss Integrtaion - Wildfly Maven Plugin

이 페이지는 Wildfly Maven plugin과 Jenkins를 이용해 JBoss 용 웹 애플리케이션을 자동으로 빌드하고 배포하는 방법을 기술한다. JBoss는 Redhat 사의 WAS 이고 무료 제품인 community version과 상용 제품인 enterprise로 구성된다. JBoss Overview Homepage = <http://www.jboss.org/> <http://www.jboss.org/Documentation> home = <https://access.redhat.com/documentation/en/red-hat-jboss-enterprise-application-platform> <https://access.redhat.com/documentation/en/red-hat-jboss-enterprise-application-platform> 다운로드 회원 가입하면 가능 JBoss 설치 및 설정 JBoss 관리는 command line console과 Web console로 접근 가능 참고 문서 =https://access.redhat.com/documentation/en-us/red_hat_jboss_enterprise_application_platform/7.0/html/getting_started_guide/administering_jboss_eap#jboss_eap_installation_overview

운영 모드 (Operating modes) JBoss는 두 가지 모드로 운영할 수 있다. 본 문서는 standalone mode를 기준으로 작성되었다. Standalone Server: 하나의 독립적인 프로세스로 실행됨 Managed Domain: 여러개의 서버로 구성되며 중앙에서 관리할 수 있음 원격 접속 설정 JBoss는 로컬에서 접근하는 것이 기본이므로 원격 PC에서 관리하려면 설정이 필요함 설정파일 <install-directory>/standalone/configuration/standalone.xml 기본 설정: 1 2 3 4 5 6 <interface name="management"><inet-address value="\${jboss.bind.address.management:127.0.0.1}"/></interface><interface name="public"><inet-address value="\${jboss.bind.address:0.0.0.0}"/></interface> 위의 설정을 다음과 같이 변경 1 2 3 4 5 6 <interface name="management"><any-address/></interface><interface name="public"><any-address/></interface> JBoss 설정 로딩하여 반영 <install-directory>/bin/jboss-cli.sh --connect --command=:reload 포트 목록 및 변경 <socket-binding-group name="standard-sockets" default-interface="public" port-offset="\${jboss.socket.binding.port-offset:0}"/><socket-binding name="management-http" interface="management" port="\${jboss.management.http.port:9990}"/><socket-binding name="management-https" interface="management" port="\${jboss.management.https.port:9993}"/><socket-binding name="ajp" port="\${jboss.ajp.port:8009}"/><socket-binding name="http" port="\${jboss.http.port:8080}"/><socket-binding name="https" port="\${jboss.https.port:8443}"/><socket-binding name="txn-recovery-environment" port="4712"/><socket-binding name="txn-status-manager" port="4713"/><outbound-socket-binding name="mail-smtp"><remote-destination host="localhost" port="25"/></outbound-socket-binding></socket-binding-group> 서비스 모드 실행 설정 TBD original site = https://access.redhat.com/documentation/en-US/JBoss_Enterprise_Application_Platform/6.1/html/Installation_Guide/Install_JBoss_Enterprise_Application_Platform_6_Red_Hat_Enterprise_Linux_Service.html https://access.redhat.com/documentation/en-US/JBoss_Enterprise_Application_Platform/6.1/html/Installation_Guide/Install_JBoss_Enterprise_Application_Platform_6_Red_Hat_Enterprise_Linux_Service.html 3.9.2. Configure JBoss EAP 6 as a Service in Red Hat Enterprise Linux (Zip, Installer) Summary Use the following procedure to install JBoss EAP 6 as a service on Red Hat Enterprise Linux when the installation has been done with either the zip, text, or graphical methods. This process does not apply when the installation has been done using the RHN (RPM) method. Prerequisites Install JBoss EAP 6 using the Zip installation, Graphical Installer, or Text-based Installer: Section 3.2.2, "Install JBoss EAP 6 (Zip Installation)" https://access.redhat.com/documentation/en-US/JBoss_Enterprise_Application_Platform/6.1/html/Installation_Guide/Install_JBoss_Enterprise_Application_Platform_6_Zip_Installation.html. Section 3.3.2, "Install JBoss EAP 6 (Graphical Installer)" https://access.redhat.com/documentation/en-US/JBoss_Enterprise_Application_Platform/6.1/html/Installation_Guide/Install_JBoss_Enterprise_Application_Platform_6_Graphical_Installation.html. Section 3.4.2, "Install JBoss EAP 6 (Text-based Installer)" https://access.redhat.com/documentation/en-US/JBoss_Enterprise_Application_Platform/6.1/html/Installation_Guide/Install_JBoss_Enterprise_Application_Platform_6_Text-based_Installation.html. Administrator privileges on the server are required. Procedure 3.17. Setup the Service Locate the start-up script and configuration file The start-up script and an associated configuration file are located in the EAP_HOME/bin/init.d/ directory. Open the configuration file jboss-as.conf to edit it. Customize the start-up options in the jboss-as.conf file There are several options within the jboss-as.conf file. At the minimum, specify the correct values for JBOSS_HOME and the JBOSS_USER variables. If these variables are absent, add them. Copy files into system directories Copy the modified configuration file to the /etc/jboss-as directory. [user@host init.d]\$ sudo mkdir /etc/jboss-as [user@host init.d]\$ sudo cp jboss-as.conf /etc/jboss-as/ Copy the start-up script to the /etc/init.d directory. [user@host init.d]\$ sudo cp jboss-as-standalone.sh /etc/init.d Add the start-up script as a service. Add the new jboss-as-standalone.sh service to list of automatically started services, using the chkconfig service management command. [user@host init.d]\$ sudo chkconfig --add jboss-as-standalone.sh Start the service. Test that the service has been installed

correctly by using the standard syntax for starting Red Hat Enterprise Linux services. [user@host bin]\$ sudo service jboss-as-standalone.sh start If everything has gone correctly, you should get a green [OK]. If you get an error, check the error logs and make sure your paths are correct in the configuration file. Make the service start automatically when you restart your server. To add the service to the list of services which start automatically when your server restarts, issue the following command. [root@host ~]# chkconfig jboss-as-standalone.sh on The service will now be started when your server reaches run-level 3, and will be stopped when your server is shut down or restarted. 실행 # > <install-directory>/bin /standalone.sh 애플리케이션 빌드 및 배포 이 절은 JBoss 용 애플리케이션을 빌드 하고 배포하는 방법을 정리한다. 이 문서는 Maven 프로젝트를 기준으로 설명한다. Maven 프로젝트의 경우 Wildfly 플러그인을 이용해 배포한다. 배포 # > mvn install wildfly:deploy -Dname=admin -Dpassword=password 배포취소 # > mvn wildfly:undeploy -Dname=admin -Dpassword=password

2017-03-20 20:51:19.131

Jenkins 관리자 가이드

이 문서는 Jenkins 관리자 가이드의 정보를 공유하기 위해 작성되었다.

2017-04-14 21:26:39.32

Jenkins & Slack Integration

이 페이지는 빌드 및 배포를 상황을 프로젝트 구성원에게 전파하는 목적으로 빌드 및 배포 도구인 Jenkins와 협업 메신저인 Slack 을 연동하는 방법을 제공한다. 사전 조건 다음의 사전 조건이 만족되어야 한다. Jenkins Version 2.39 Slack Team manager 권한 계정 Jenkins 연동용 channel 생성 및 구성 Jenkins --[https]--> Slack 접속 가능 설정 호를 설정은 아래의 과정으로 진행한다. Step 1) Slack 설정 Jenkins App 설치 및 설정 Step 2) Jenkins plugin 설치 및 설정 Step 3) Jenkins job 단위로 post-build action 설정 Slack 구성 Step 1) Jenkins App 설치 Team menu >> Apps & Integrations >> Manage >> Apps 페이지로 이동하여 "Jenkins CI" 앱을 설치한다. Screen Shot 2017-04-14 at 1.41.28 PM.png Step 2) Jenkins CI 클릭 >> Add Configuration 이미 configuration이 추가했다면 연필 아이콘을 클릭하여 편집 화면으로 이동한다. Post to Channel: 메시지가 표시될 기본 channel 설정 (Jenkins job 별로 설정 가능) Token: Jenkins에서 Slack에 접속할 때 사용할 token 나머지 항목은 설정하지 않아도 됨 Screen Shot 2017-04-14 at 1.44.24 PM.png Jenkins plugin 설치 및 설정 Step 1) Slack Plugin 설치 Jenkins 관리 >> Manage Plugin >> Available 화면으로 이동 Slack Notification Plugin을 찾아 check box 선택 >> Install without restart 클릭 Jenkins 2.74의 경우 Plugin 2.0 설치 권장 Step 2) Jenkins 재시작 Slack Notification plugin은 Jenkins 재시작 을 요구한다. Restart Safely plugin이 설치된 경우 Jenkins menu에서 Restart Safely 메뉴를 선택하여 Jenkins를 재기동 시킨다. Step 3) Global 설정 Jenkins 관리 >> Configure System 메뉴를 선택하여 global 설정 페이지로 이동한다. Global Slack Notifier Settings 섹션으로 이동하여 세부 설정을 진행한다. Team Subdomain: Slack에 등록된 team URL의 subdomain 입력 (예: https://curvc.slack.com https://curvc.slack.com 이라면 "curvc"가 subdomain) Integration Token: Slack 설정의 Token을 복사하여 붙여넣기 Channel: Jenkins 메시지가 표시될 default Slack channel 설정. Job 구성에서 job 단위로 다른 channel 설정 가능 (예: #curvc-ci) Build Server URL: Slack message에 링크될 Jenkins 서버 주소 설정. 주소 마지막에 '/' 문자 입력 (예: http://jenkins-demo.curvc.com:78180) http://jenkins-demo.curvc.com:78180 image2017-4-14_21-24-11.png Jenkins Job 설정 Slack Channel에 빌드 상태 변경을 메시지를 표시하기 위해 Job 을 구성한다. Step 1) Job 설정 페이지로 이동 Jenkins >> Job 이름 >> Configure 페이지로 이동한다. Add post-build action 을 펼쳐 >> Slack Notifications 메뉴를 선택한다. Step 2) 메시지 생성할 이벤트 설정 Slack channel에 표시될 이벤트를 모두 선택한다. Step 3) 추가 정보 설정 Advanced... 버튼 클릭 Team Subdomain: Global 설정과 동일하다면 설정하지 않아도 됨 Integration Token: Slack 설정의 Token. Global 설정과 동일하다면 설정하지 않아도 됨 Project Channel: Jenkins 메시지가 표시될 default Slack channel 설정. Global 설정과 동일하다면 설정하지 않아도 됨 Screen Shot 2017-04-14 at 9.25.43 PM.png Slack에 나타나는 빌드 수행 내용 Jenkins와 Slack 연동되면 빌드 상황이 Slack channel에 메시지로 전송된다. 아래는 Slack channel에 표시된 빌드 정보의 예를 보여준다. image2017-4-14_21-20-4.png

2017-03-09 22:13:17.952

1. Jenkins 시스템 설정

Jenkins Configuring the System Jenkins의 시스템을 설정하기 위해 Jenkins 관리 > 시스템 설정 클릭한다. 1-1.Guide.png 일반 시스템 설정 다음과 같은 정보를 수정할 수 있다. 작업공간 최상위 디렉토리 빌드 기록 최상위 디렉토리 시스템 메시지 of executors Labels : 레이블 Usage Use this node as much as possible : 기본 설정 Only build jobs with label expressions matching this node : 노드와 레이블이 일치 할때 빌드 Quiet period : 젠킨스가 빌드 시작하기 전에 지정된 기간 동안 대기 SCM checkout retry count : 체크아웃 재 카운트 Restrict project naming : 프로젝트 이름 지정 제한 1-2.Guide.png Global properties Global properties에서는 환경변수, 도구 위치 등을 설정한다. Environment variables : 환경 변수 값을 추가하거나 삭제할 수 있다. Tool Locations : Tool 위치를 지정하거나 삭제할 수 있다. Help make Jenkins better by sending anonymous usage statistics and crash reports to the Jenkins project. : 젠킨스 사용에 대한 정보를 주기적 발송 1-3. Guide.png Timestamper Timestamper는 Jenkins에서 사용되는 시간 형식을 설정한다. System clock time format : 시스템 시간 형식 Elapsed time format : 경과 시간 형식 1-4.Guide.png Jenkins Location Jenkins Location에서는 Jenkins 주소 및 이메일 설정할 수 있다. Jenkins URL : 젠킨스 웹 주소 System Admin e-mail address : 시스템 관리자 이메일 주소 1-5.Guide.png SSH Server SSHD Port : SSHD 포트 1-6.Guide.png GitHub GitHub 섹션에서는 GitHub 연동을 위한 것들을 설정할 수 있다. GitHub Servers Add GitHub Server > GitHub Server : GitHub 서버 세팅 Re-register hooks for all jobs Override Hook URL Specify another hook url for GitHub configuration : GitHub 다른 hook 주소 설정 Shared secret Add > Jenkins : Jenkins가 요청을 GitHub에서 가져 왔는지 확인하기 위한 서명하는 데 사용 Additional actions Convert login and password to token : 로그인, 비밀번호를 GitHub 개인 토큰으로 변환 1-7.Guide.png GitHub Enterprise Servers API endpoint : API 끝지점 Name : 이름 1-8.Guide.png Pipeline Model Definition Docker Lable : 도커 레이블 Docker registry URL : 도커 레지스트리 URL Registry credentials > Add > Jenkins : 레지스트리 자격증명 1-9. Guide.png Global Pipeline Libraries 시스템에서 Pipeline Job을 실행할때 이용가능한 라이브러리를 등록 한다. Name : 이름 Default version : 기본 버전 Load implicitly : 암시적으로 Allow default version to be overridden : 기본 저버를 무시하도록 허용 Retrieval method Modern SCM : 최신 인터페이스를 사용하여 SCM 플러그인에서 라이브러리를 로드 Legacy SCN : Jenkins Pipeline에서 지원하는 모든 SCM 플러그인에서 라이브러리를 로드 1-10.Guide.png Build-timeout Plugin > Build Step Action Enable BuildStep Action : 빌드셋업 동작 사용 1-11. Guide.png Git plugin Jenkins와 Git 연동 설정을 위한 설정 Global Config user.name Value : 글로벌 설정 유저 이름 값 Global Config user.email Value : 글로벌 설정 유저 이메일 값 Create new accounts base on author/commiter's email : 커미터 이메일 기반으로 새 계정 만들기 1-12.Guide.png Subversion Jenkins와 Subversion 연동을 위한 설정 Subversion Workspace Version : svn workspace 버전 Exclusion revprop name : 제외 리플롭 이름 1-13. Guide.png Shell Shell executable : 만약 sh 바이너리 PATH가 외부에 있을 경우, 절대 경로를 지정 1-14. Guide.png Extended E-mail Notification SMTP server : smtp 서버 Default user E-mail suffix : 기본 이메일 사용자 User SMTP Authentication : smtp 사용자 Use SSL : SSL 사용 SMTP port : smtp 포트 Charset : 캐릭터셋 Default Content Type Plain Text (text/plain) : 일반 텍스트 HTML (text/html) : HTML 텍스트 Use List-ID Email Header : 목록 ID 이메일 헤더 사용 Add 'Precedence: bulk' Email Header : 우선순위 이메일 헤더 추가 Default Recipients : 기본 수신자 Reply To List : 응답 목록 Emergency reroute : 긴급 재 전송 라우터 Excluded Recipients : 제외된 수신자(필터링으로 수신자를 걸러냄) Default Subject : 기본 제목 Maximum Attachment Size : 최대 첨부파일 사이즈 Default Content : 기본 내용 Default Pre-send Script : 미리 보내기 스크립트 Default Post-send Script : 전송 후 스크립트 1-15. Guide.png Additional groovy classpath : 추가 그루비 클래스 스패스 Enable Debug Mode : 디버그 모드 Require Administrator for Template Testing : 관리자 권한 있는 사용자에게 이메일 템플릿 링크 표시 Enable watching for jobs : 감시 기능 Default Triggers: 기본트리거 1-16. Guide.png E-mail로 알려줌 해당 섹션에서는 이메일 알림을 위해 사용되는 서버 정보를 입력한다. SMTP 서버 : smtp 서버 Default user e-mail suffix : 기본 유저 이메일 Use SMTP Authentication : smtp 인증 사용 SSL 사용 : SSL 사용 SMTP Port : smtp 포트 Reply-To Address : 회신 주소 Charset : 캐릭터셋 Test configuration by sending test e-mail : 테스트 세팅으로 테스트 이메일 전송 1-17.Guide.png

2017-03-09 21:48:15.219

8. Jenkins Managing Users

이 문서는 Jenkins에서 사용자를 관리하는 방법에 대한 가이드를 제공한다. Jenkins 사용자 관리 Jenkins에는 다음과 같은 여러가지의 인증 방법이 지원되고 있다. Delegate to servlet container Jenkins own user database LDAP Unix user /group database 이를 설정하기 위해서는 Jenkins 관리 > Configure Global Security의 Access Control을 이용한다. 8-1.Guide.png Jenkins own user database Jenkins own user database는 Jenkins 자체적인 사용자 관리 시스템을 사용하는 방법으로 관리자가 사용자를 추가하는 방식과 일반 사용자가 가입을 하는 방법이 있다. 사용자 추가 Jenkins의 사용자를 추가하기 위해 Jenkins 관리 > Manage Users 메뉴로 이동한다. 왼쪽 메뉴에서 사용자 생성을 선택하고 적절한 정보를 입력 후 Create User 버튼을 클릭한다. image2017-3-8_12-56-0.png 사용자 가입 허용 설정 사용자 가입 허용을 위해 Jenkins 관리 > Configure Global Security > Access Control에서 사용자의 가입 허용을 체크 후 Save 버튼을 설정한다. 8-2.Guide.png 계정 생성 사용자의 가입 허용 후에 다음과 같이 로그인 화면에서 계정 생성을 확인할 수 있다. 8-3.Guide.png 회원 가입을 위한 적절한 Sign up 정보를 입력하고 Sign up 버튼을 클릭 한다. 8-4.Guide.png LDAP 방식의 사용자 관리 LDAP을 사용하여 사용자 인증을 위해 Jenkins 관리 > Configure Global Security > Access Control에서 LDAP을 선택하고 필요한 정보를 입력한다. image2017-3-8_13-1-23.png User 정보 변경 User 정보 변경 하기 위해선 admin 계정으로 접속 해야 한다. Jenkins 관리 > Manage Users 클릭 한다. 8-6.Guide.png 정보를 변경 할 유저 우측의 기어 모양을 클릭 한다. 8-7.Guide.png 아래 그림 처럼 정보를 변경 할 수 있다. 정보를 변경 하고 Save 버튼을 클릭 한다. 8-8.Guide.png User 삭제 User 삭제 하기 위해선 admin 계정으로 접속 해야 한다. Jenkins 관리 > Manage Users 클릭 한다. 삭제 하고 싶은 유저의 우측 image2017-3-7_10-52-29.png 아이콘을 클릭 한다. 8-10.Guide.png 삭제 할 것인지 한번 더 묻는다. Yes 버튼을 클릭 한다. 8-11.Guide.png 유저가 삭제 되었음을 확인할 수 있다. 8-12.Guide.png User 권한 부여 유저 권한 부여 하기 위해선 admin 계정으로 로그인 해야 한다. Jenkins 관리 > Configure Global Security 클릭 한다. Authorization > Matrix-based security 또는 Project-based Matrix Authorization Strategy > User/group to add 의 유저를 추가 해주고 권한을 부여 하고 Save 버튼을 클릭 한다. 8-13.Guide.png 만약 실수로 Admin이 Access 할 수 없을 경우, 다음과 같이 \$Jenkins_Home\$config.xml 를 설정 후 Jenkins를 재 시작 한다. 8-15.Guide.png

2017-03-09 22:53:10.78

4. Jenkins Managing Plugins

플러그인 관리 플러그인 관리에서는 Jenkins Job 설정을 돕는 다양한 플러그인들에 설치, 업데이트, 삭제를 수행한다. 이를 위해 Jenkins 관리 > 플러그인 관리로 이동 한다. 4-1.Guide.png 업데이트된 플러그인 목록 업데이트 된 플러그인 목록 탭에서는 설치되어 있는 플러그인 중에 업데이트가 가능한 플러그인의 리스트를 보여준다. 4-2.Guide.png 설치 가능 설치 가능 탭에서는 설치되지 않은 Jenkins의 플러그인 리스트를 확인할 수 있다. 상단 필터에 원하는 키워드를 입력하여 설치를 원하는 플러그인을 필터할 수 있다. 플러그인을 설치하기 위해서는 리스트에서 설치 컬럼을 체크하고 하단 설치하기 버튼을 클릭한다. 4-3.Guide.png 설치된 플러그인 목록 설치된 플러그인 목록 탭을 선택 시 설치된 플러그인 목록 리스트를 확인 및 설치 제거를 할 수 있다. 4-4.Guide.png 고급 고급 탭에서는 플러그인 설치를 위한 프록시 설정, 수동으로 플러그인 올리기, 업데이트 사이 정보등을 설정할 수 있다. HTTP 프록시 설정 : 프록시 설정을 통해 업데이트를 수행한다. 플러그인 올리기 : 다운로드 받은 플러그인 파일을 수동으로 설치한다. 업데이트 사이트 : 업데이트 사이트 경로를 입력한다. 4-5.Guide.png

2017-03-09 22:29:38.953

2. Jenkins Managing Security

Configure Global Security Enable security TCP port for JNLP agents > Fixed, Random, Disable Disable remember me : 나를 기억하지 못하게 한다. Access Control 다음 사용자 접근 권한에 대한 간략한 정보를 보여준다. 사용자 구성 및 관리는 사용자 관리 섹션을 참고한다. Security Realm : 보안 영역 Delegate to servlet container assets, cli, git, github-webhook, jnlpJars, subversion, whoAml : 서블릿 컨테이너를 이용하여 사용자 인증 Jenkins' own user database 사용자 가입 허용 : Jenkins 데이터베이스 사용 LDAP ex) ldaps://server:port : LDAP 사용자 인증 Unix user/group database Service Name > Test : 유닉스 운영체제 사용자 데이터베이스 인증 Authorization : 권한 부여 Anyone can do anything : 누구나 무엇이든 할 수 있다. Legacy mode : 레거시 모드 Logged-in users can do anything Allow anonymous read access : 로그인 한 사용자는 무엇이든 할 수 있다. Matrix-based security : 매트릭스 기반 보안 Project-based Matrix Authorization Strategy : 프로젝트 기반 매트릭스 권한 부여 2-1.Guide.png Markup Formatter Plain text, Safe HTML : 일반 텍스트, HTML Prevent Cross Site Request Firgery exploits 교차 사이트 요청 위조 공격 방지 Crumbs Crumb Algorithm : Crumbs 알고리즘 Default Crumb Issuer > Enable proxy compatibility Plugin Manager Use browser for metadata download : 메타데이터 다운로드에 브라우저 사용 Enable Slave → Master Access Control : 슬라브 활성화 2-2.Guide.png

2017-03-09 22:43:19.199

3. Jenkins Managing Tools

이 문서는 Jenkins Global Tool Configuration에 대한 가이드를 제공한다. Global Tool Configuration Global Tool Configuration에서는 Jenkins Job에서 사용되는 다양한 도구들의 설정을 구성할 수 있다. 이를 위해 Jenkins 관리 > Global Tool Configuration 클릭 한다. 3-1.Guide.png Maven Configuration Maven 설정에 대한 옵션을 선택할 수 있다. Default settings provider Use default maven settins : 기본 maven 설정 사용 Settings file in filesystem - File path : 파일 시스템의 설정 파일 Default global settings provider Use default maven global settings : 기본 글로벌 maven 설정 사용 Global settings file on filesystem : 파일 시스템의 글로벌 설정 파일 3-3.Guide.png JDK Jenkins 빌드에 사용될 수 있는 JDK 설정을 할 수 있다. JDK installations Name : 이름 Install automatically : 자동 설치 Version : 버전 Add Installer : 인스톨 3-4.Guide.png Git Jenkins 빌드에서 사용될 수 있는 Git Client 설정을 할 수 있다. Git installations Name : 이름 Path to Git executable : 실행 파일의 대한 경로 3-5.Guide.png Gradle Jenkins 빌드에서 사용될 수 있는 Gradle 설치 설정을 할 수 있다. Gradle installations Name : 이름 Install automatically : 자동 설치 Version : 버전 Add Gradle : Gradle 추가 3-6.Guide.png Ant Jenkins 빌드에서 사용될 수 있는 Ant 설치 설정을 할 수 있다. Ant installations Name : 이름 Install automatically : 자동 설치 Version : 버전 Add Ant : Ant 추가 3-7.Guide.png Maven Jenkins 빌드에서 사용될 수 있는 Maven 설치 설정을 할 수 있다. Maven installations Name : 이름 Install automatically : 자동 설치 Version : 버전 Add Maven : Maven 추가 3-8.Guide.png Docker Jenkins 빌드에서 사용될 수 있는 Docker 설치 설정을 할 수 있다. Docker installations Name : 이름 Installation root : root로 설치 Install automatically : 자동 설치 Add Docker : 독커 추가 3-9.Guide.png

2017-03-06 16:22:50.603

5. Jenkins CLI(보류)

Jenkins CLI Jenkins 관리 > Jenkins CLI 메뉴로 접속 한다. 5-1.Guide.png

2017-03-07 21:43:19.418

7. Jenkins Managing Nodes

노드관리 Jenkins 관리 > 노드 관리 클릭 한다. 7-1.Guide.png 노드 설정 설정 하기 위해 노드의 우측의 기어 아이콘 버튼을 클릭 한다. 7-2.Guide.png 설정 of executors 라벨 Usage > Use this node as much as possible 7-3.Guide.png

2017-03-09 23:00:58.839

9. Backing Up/Restoring Jenkins

Backup ThinBackup 설치 Backup을 하기 위해 ThinBackup Plugin을 설치한다. Jenkins 관리 > 플러그인 관리 9-1.Guide.png 설치 기능 탭 선택 우측의 필터란에 backup 이라고 입력 후 ThinBackup 체크 하여 재시작 없이 설치하기 버튼 클릭 한다. 9-2.Guide.png ThinBackup 설정 ThinBackup을 통해 백업 설정을 위해 Jenkins 관리 > ThinBackup 선택한다. 9-5.Guide.png 설정을 위해 Settings를 클릭한다. 9-6.Guide.png Backup 설정은 다음 정보를 참고하여 설정을 수행한다. Backup Directory : ex) /home/backup Backup schedule for full backups : * * * * * command to execute[*minute(0-59),*hour(0-23),*day(1-31),*month(1-12),*day of week(0-6 sunday to saturday) Backup schedule for differential backups : 증분백업은 풀 백업 이후 수정된 전체 파일만 백업 합니다. Max number of backup sets : 백업 세트의 최대 수 Files excluded from backup : 백업에서 제외 된 파일 Wait until Jenkins/Hudson is idle to perform a backup : 백업을 수행하기 위해 Jenkins / Hudson이 idle 상태가 될 때까지 기다립니다. > Force Jenkins to quiet mode after specified minutes : 지정한 시간이 지난 후 Jenkins는 자동모드로 된다 Backup build results : 빌드 결과 백업 Backup build archive : 아카이브 빌드 백업 Backup only builds marked to keep : 백업만 유지되도록 표시된 빌드 Backup 'userContent' folder : userContent 폴더 백업 Backup next build number file : 다음 빌드 번호 파일 백업 Backup plugins archives : 아카이브 플러그인 백업 Backup additional files : 추가 파일 백업 > Files included in backup (regular expression) : 백업에 포함 된 파일 Clean up differential backups : 증분 백업 정리 Move old backups to ZIP files : 이전 백업 파일을 ZIP 파일로 이동 9-7.Guide.png Backup 시작 백업을 시작하기 위해 Backup Now 버튼을 클릭 한다. 9-8.Guide.png Restore 백업된 파일을 복원하기 위해 Restore 버튼을 클릭 한다. 9-10.Guide.png Restore Options 설정 후 Restore 버튼을 클릭 한다. restore backup from : 복원 시점 Restore next build number file (if found in backup) : 다음 빌드 번호 파일 복원 Restore plugins : 플러그인 복원 9-14.Guide.png Jenkins 관리 > System Log 확인 9-11.Guide.png 모든 Jenkins 로그 클릭 한다. 9-12.Guide.png Restore가 완료 되었음을 확인 할 수 있다. 9-13.Guide.png

2021-09-03 22:05:56.235

백업 파일로 부터 하나의 Job 복원 방법

이 페이지는 백업 파일로 부터 하나의 Job를 복원하는 방법을 정리한다. 사전 조건 Jenkins 관리자 계정 Jenkins master 가 설치된 서버의 Jenkins 계정 thinBackup plugin이 생성한 백업 파일 존재 백업 절차 백업 폴더 확인 Jenkins 관리 > ThinBackup > Settings > Backup directory 필드 확인 Screen Shot 2021-09-03 at 9.55.06 PM.png 백업 파일 또는 폴더 확인 복원할 백업 폴더 이름 규칙: FULL-
<yyyy-MM-dd_hh-mm> Jenkins home directory 경로 확인 Jenkins 관리 > 시스템 설정 > 홈 디렉터리 Screen Shot 2021-09-03 at 10.02.21 PM.png 복원할 job home directory로 복사 예) \$ cd /data/backup-jenkins-master/FULL-2021-10-20_13-20/jobs \$ cp -r EXAMPLE-JOB /data/jenkins_home/jobs/ Job 읽어 들이기 복원한 job을 Jenkins에 반영한다. Jenkins 관리 > Reload Configuration from Disk 복원된 Job 확인

2017-04-10 14:39:56.942

백업 및 복원

이 페이지는 백업과 복원 가이드를 제공한다. 백업 대상 CentOS의 경우 백업될 데이터: /var/lib/ldap Configuration 데이터: /etc/openldap 설치폴더: 한 폴더에 모여 있지 않고 /lib64 등으로 설치되기 때문에 설치폴더는 백업하지 않음 참고 문서 : <https://www.techbrown.com/ldap-configuration-install-centos-7-rhel-7.shtml> <https://www.techbrown.com/ldap-configuration-install-centos-7-rhel-7.shtml>

2017-03-20 20:51:01.618

LDAP 관리자 가이드

2017-02-22 14:03:29.197

3. LDAP Group 관리

Groups 생성 새로운 Group을 생성 하기 앞서 여러명의 User가 생성이 되어야 한다. 2. LDAP User 관리 Organizational Unit 생성 새로운 Organizational unit을 생성하기 위해 LDAP 상위 Entry를 선택하고 오른쪽 마우스 클릭 > New > Organizational unit ...을 클릭한다. 만약 조직에서 사용하는 특별한 명명 법칙이 있다면, 거기에 준하여 Organizational Unit의 Name을 입력한다. 다음 그림은 Groups이라는 Organizational Unit을 생성하는예제를 보여준다. 22.install.png Create Organizational Unit의 name을 입력 하고 OK 버튼을 클릭 한다. 23.install.png 다음 그림은 Organizational Unit의 Groups이라는 object가 생성이 되었다. 24.install.png Group 생성 새로운 그룹을 생성하기 위해 생성을 위한 Entry를 선택하고 오른쪽 마우스를 클릭하여 New > Group를 클릭 한다. Dev Group를 생성하는 예제 그림 이다. 25.install.png Create Group의 Group name을 입력 합니다. 그리고 Members 탭의 Add 버튼을 클릭 한다. 26.install.png Dev Group에 소속되어야 할 User를 선택 하고 OK 버튼을 클릭 한다. 27.install.png 선택한 User가 속해 있는지 확인 하고 OK 버튼을 클릭한다. 28.install.png Dev Group에 추가한 User1,2,3 이 있음을 확인 할 수 있다. 29.install.png Group 의 Member 추가 및 삭제 Member를 추가 및 삭제 해야 할 Group를 선택하고 상단 메뉴의 체크 박스 아이콘 선택 또는 해당그룹의 선택 마우스 오른쪽 클릭 > Properties를 선택하여 Group의 Member를 추가 및 삭제 한다. 30.install.png Dev Group의 해당 Member를 Add 또는 Remove 할 수 있다. 31.install.png 다음 그림은 user3을 삭제 하고 user4를 추가 하는 예제 이다. user3를 선택 하고 Remove 버튼을 클릭 한다. 32.install.png user3이 삭제 되었음을 알 수 있습니다. Add 버튼을 클릭 한다. 33.install.png Choose members의 user4를 선택하고 OK 버튼을 클릭 한다. 34.install.png Dev Group의 Members의 user4이 추가 된 것을 알 수 있다. OK 버튼을 클릭하여 최종 적용 한다. 35.install.png Dev Group의 User1,2,4가 추가가 되었음을 알 수 있다. 36.install.png Group 삭제 Group를 삭제하기 위해 해당 그룹을 선택 후 상단의 X 아이콘 버튼 클릭 또는 마우스 오른쪽 클릭 > Delete 버튼을 클릭한다. 37.install.png 다음 그림은 해당 되는 entry를 지우겠다고 물어 봅니다. Yes 버튼을 클릭 하면 Group이 삭제가 된다. 38.install.png Dev Group이 삭제 되었음을 알 수 있다. 39.install.png

2017-02-22 00:47:14.66

1. LDAP 소개

이 문서는 LDAP에 대한 간단한 정보를 공유하기 위해 작성되었다. LDAP 이란 LDAP (Lightweight Directory Access Protocol)은 네트워크 상에서 조직이나 개인정보 혹은 파일이나 디바이스 정보 등을 찾아 보는 것을 가능하게 만든 소프트웨어 프로토콜이다. LDAP은 X.500 디렉토리 서비스를 위한 DAP(Directory Access Protocol)의 일종이다. DAP는 OSI Protocol stack에서 작동되고, 컴퓨팅의 많은 자원을 사용하는 무거운 프로토콜이다. 이러한 문제점을 해결한것이 LDAP이다. LDAP은 TCP/IP에서 작동되며, 보다 저렴한 비용으로 DAP의 기능을 대부분 지원할 수 있다. image2017-2-21_11-8-58.png LDAP 구조 일반적으로 LDAP은 다음과 같이 구성된다. cn (Common Name) : GilDong Hong 와 같은 일반적인 이름 sn (Surname Name) : 우리나라 성에 해당 ou (Organization Unit) : 그룹에 해당 dc (Domain Component) : 도메인의 요소 w http://ara.kaist.ac.kr/www.google.com http://www.google.com 의 dc는 dc=www,dc=google,dc=com dn (Distinguished Name) : 고유한 이름 o : organization c : country uid : user id image2017-2-21_14-56-56.png

2017-02-22 14:01:05.256

2. LDAP User 관리

이 문서는 LDAP Admin을 통해 Open LDAP을 관리하는 방법에 대해서 가이드하기 위해 작성되었다. Windows LDAP Admin 다운로드 다음 링크를 통해 LdapAdmin을 다운로드 받는다. LdapAdmin 1.7.2(<http://www.ldapadmin.org/download/ldapadmin.html>) LDAP 설치 만약 Open LDAP이 설치되어 있지 않다면, 다음 링크를 통해 설치를 진행할 수 있다. Ldap 설치 가이드 LDAP 연결 설정 Ldap 서버로 연결 설정하기 위해 다운로드 받은 LdapAdmin.exe 실행한다. 메뉴의 start > connect를 선택한다. 또는 New connection을 선택한다. 1.install.png Connection properties 창이 나타나면 다음과 같이 정보를 입력한다. Connection name : LDAP connection을 위한 이름을 입력한다. General 탭 Host : LDAP 서버의 주소를 입력 Port : 389 (기본) Base : ex) dc=curvc, dc=com Username : ex) cn=admin, dc=curvc, dc=com Password : 패스워드 입력 2.install.png 모든 정보가 입력되면 하단 Test connection을 통해 접속을 Test 하고 OK 버튼을 클릭한다. 3.install.png 새롭게 생성한 연결 정보로 접속하기 위해 해당 연결 정보를 클릭하고 OK 버튼을 클릭한다. 4.install.png 다음 그림은 Ldap 구조가 어떻게 되어 있는지 확인 가능하다. 5.install.png Organizational Unit 생성 새로운 Organizational unit을 생성하기 위해 LDAP 상위 Entry를 선택하고 오른쪽 마우스 클릭 > New > Organizational unit ...을 클릭한다. 만약 조직에서 사용하는 특별한 명명 법칙이 있다면, 거기에 준하여 Organizational Unit의 Name을 입력한다. 다음 그림은 people이라는 Organizational Unit을 생성하는예제를 보여준다. 6.install.png Create Organizational Unit의 name을 입력 하고 OK 버튼을 클릭 한다. 7.install.png 다음 그림은 Organizational Unit의 people이라는 object가 생성이 되었음. 8.install.png User 생성 새로운 사용자를 생성하기 위해 생성을 위한 Entry를 선택하고 오른쪽 마우스를 클릭하여 New > User를 클릭한다. 9.install.png Create User 창이 나타나면, 다음과 같이 필수 정보를 입력한다. First name : 이름 Second name : 성 Display name : 보여지는 이름 Username : 로그인 이름 Home Directory : 홈디렉토리 10.install.png Business 탭으로 이동하여 하단에 E-Mail Addresses에서 Add 버튼을 클릭한다. Add Address 창이 나타나면 SMTP Address 필드에 이메일 정보를 입력하고 OK 버튼을 클릭한다. 11.install.png 다음 그림은 people이라는 Organizational unit 아래의 Hong GilDong이라는 user가 생성이 된것을 보여 주고 있다. 12.install.png User 패스워드 설정 사용자 패스워드를 설정하기 위해서 해당 유저를 선택하고 Set Password를 선택한다. 13.install.png New password와 Confirm password를 입력합니다. Encryption method type를 선택하고 OK 버튼을 클릭한다. 14.install.png 다음 그림은 패스워드가 생성되었음을 보여주는 그림이다. 15.install.png User 정보 수정 사용자 정보를 수정하기 위해 수정을 원하는 사용자를 선택하고 상단 메뉴의 체크 박스 아이콘 선택 또는 해당유저 선택 마우스 오른쪽 클릭 > Properties를 선택하여 사용자 정보를 변경한다. 16.install.png 해당 유저의 정보를 변경 할 수 있다. 17.install.png 다음 그림은 정보가 변경 되었음을 알 수 있다. 18.install.png User 삭제 사용자 삭제하기 위해 해당 유저 선택 후 상단의 X 아이콘 버튼 클릭 또는 마우스 오른쪽 클릭 > Delete 버튼을 클릭한다. 19.install.png 다음 그림은 해당 되는 entry를 지우겠냐고 물어 봅니다. Yes 버튼을 클릭 하면 User 삭제가 된다. 20.install.png User가 삭제 되었음을 확인 할 수 있다. 21.install.png

2017-02-23 14:25:12.848

CollabNet Subversion

2018-02-27 09:10:11.478

Set Up CollabNet Edge

CollabNet Edge는 administration console은 jetty 그리고 apache httpd로 svn 서비스가 구성된다. 설치 환경 Windows, Linux, MacOS 지원 사전 설치: Java 설치 가이드 CollabNet에서 제공하는 설치 가이드를 참조한다. Set up Subversion Edge <http://help.collab.net/topic/csvn/action/setupcsvn.html>

2018-03-24 07:13:58.187

CollabNet Subversion Edge의 관리자 암호 초기화 방법

Edge의 관리자 암호 기준이 까다롭기 때문에 복잡한 암호를 설정해야 하기 때문에 자칫 설정했던 암호가 생각나지 않는 경우가 있다. 간단하게 암호를 초기화 할 수 있으니 다행 :) Screen Shot 2018-03-24 at 6.45.41 AM.png Step 1) Subversion Edge 서비스 중지 사용하는 Linux, Windows의 서비스 종료 방법에 따라 서비스 종료 Step 2) HSQL db 파일 찾기 csvn-production-hsqldb.script 파일의 위치 찾기 <csvn 설치 폴더>/data 하위에 위치 Step 3) 관리자 암호 수정 csvn-production-hsqldb.script 파일을 편집기로 열어서 Super Administrator 문자열을 찾아 암호 수정 INSERT INTO USER VALUES(1,2,'admin user','admin@example.com', TRUE, 'curvcasdef7898abde4252aedcb4352','Super Administrator','admin') curvc... 부분이 비밀번호 설정하는 부분이고 다음 문자열로 대체 (암호: admin 에 해당) 21232f297a57a5a743894a0e4a801fc3 Step 4) 파일 저장 후 서비스 재기동 id / password = admin admin 으로 로그인

2018-02-27 08:52:18.366

CollabNet Subversion Manage Repositories

이 문서는 CollabNet Subversion에서 저장소를 관리하는 가이드를 제공한다. 저장소 관리 Repositories를 생성 하기 위해 <http://ipaddress:3343> 으로 접속 및 로그인 한다. 저장소 리스트 보기 좌측 메뉴의 Repository List를 클릭 하면 모든 Repositories의 정보를 확인 할 수 있다. 해당 Repository의 아래 메뉴의 기능을 지원 한다. 생성 Discover 정보 Dump 검증 Load 삭제 저장소 생성 상단의 Repositories를 클릭 한다. 생성 버튼을 클릭한다. 14.Guide.png Repository를 생성하기 위해 다음 정보를 입력하고 Create 버튼을 클릭한다. Name : 저장소 명(영문, 빈칸 없이 생성) Initialize Empty repository : 빈 저장소 생성 Create standard trunk/branches/tags structure Backup : 백업으로 생성 15.Guide.png 저장소 정보 새로운 저장소가 생성되며, 해당 저장소에 대한 정보를 확인할 수 있다. Status : 상태정보 Filesystem Format : 파일 시스템 포맷 정보 Revisions : 리비전 정보 Repository Format : Repository 정보 Size : 사이즈 정보 Sharding : 공유 정보 Packed : 패키지 정보 Representation Sharing Supports : Supports 정보 UUID : UUID 정보 Hook Scripts : Hook Scripts Backup Files : 백업 파일의 대한 정보 Scheduled Jobs : 스케줄 작업 정보 Size Report : 저장소 사이즈의 대한 정보 16.Guide.png 저장소 덤프 저장소를 덤프 하기 위해 해당 저장소를 선택 하고 Dump 버튼을 클릭 한다. 29.Guide.png 덤프 옵션 Revisions : 콜론을 사용하여 수정 범위를 입력 한다, 모든 수정본을 포함 하려면 입력란을 비워둔다. Incremental : 덤프의 첫번째 수정된 파일 디렉토리만 포함 Use deltas : 이전 버전의 델타로 표시되므로 파일을 작게 만들 수 있지만 필터링이나 편집에는 적합하지 않다. Compress : zip 파일 형식으로 압축 Apply filter : 덤프되는 내용을 제한하는 경로 기반포함 또는 제외 필터 덤프 버튼을 클릭 한다. 30.Guide.png 덤프 파일이 생성이 되었다. 31.Guide.png 새로고침으로 백업 파일이 생성 되었음을 확인 할 수 있다. 덤프파일을 다운 받을 수 있거나 삭제가 가능하다. 32.Guide.png 저장소 검증 저장소 검증을 하기 위해 우선 Mail Server를 설정 해야 한다. SMTP 정보를 입력하고 저장 한다. 상단 메뉴의 Administrator > Mail Server Mail Server Name Port Username Password Connection Security Protocol From Address 33.Guide.png 저장소 리스트 화면으로 돌아 와서 해당 저장소를 선택하고 검증 버튼을 클릭한다. 34.Guide.png 저장소 검증의 대한 메일이 해당 메일로 전송이 되었음을 확인 할 수 있다. 35.Guide.png 저장소 Load 저장소 Load 하기 위해서 Dump파일이 있어야 한다. Dump 만드는 방법은 위에 저장소 덤프 설명을 참고 하기 바란다. 36.Guide.png zip으로 만든 Dump 파일을 업로드 한다. Schedule load 버튼을 클릭 한다. Load Repository from Dump File repository name Status Head revision UUID 37.Guide.png 덤프파일이 저장소로 업로드 되도록 예약이 되었다. 진행사항은 Jobs 화면에서 볼 수 있다. 38.Guide.png Jobs 화면으로 이동 한다. Administrator > Jobs 작업이 완료 되었음을 확인 할 수 있다. 39.Guide.png 저장소 삭제 저장소 삭제 하기 위해 저장소 List로 이동 합니다. 삭제 할 저장소를 선택 하고 삭제 버튼을 클릭 합니다. 40.Guide.png 마지막으로 삭제 할 것인지? 묻는다. 계속 진행 하려면 화면에서 보여주는 텍스트를 입력 하고 OK 버튼을 클릭 한다. 41.Guide.png 저장소가 삭제 되었음을 확인 할 수 있다. 42.Guide.png 저장소 접근 권한 설정 저장소의 접근 권한을 설정하기 위해 저장소 메뉴에서 Access Rules을 선택한다. 편집 버튼을 클릭하여 편집을 수행한다. 18.Guide.png 그룹 설정을 위해서 [groups]을 사용한다. 다음과 같이 그룹명 = 사용자, 사용자2 와 같은 형태로 그룹에 사용자를 할당할 수 있다. [groups] mgrs = user1, user2 engrs = engr1, engr2 allgroups = @mgrs, @engrs 설명 /tags 라는 디렉토리는 전체 유저가 읽을 수 있는 권한이 있고 @mgrs 라는 그룹 유저들은 읽고 쓰기 권한이 있는 설정이다. [dev:/tags] * = r @mgrs = rw 설명 /branches /components/xxx 라는 디렉토리는 사용자1은 읽고 쓰기 권한이 있고 사용자2는 읽을 수 있는 권한이 있다. [dev:/branches/components/xxx] user1 = rw user2 = r user3 19.Guide.png Access Rules이 설정 되었음을 확인 할 수 있다.20.Guide.png 저장소 백업 설정 저장소 백업을 수행하기 위해 백업 스케줄을 설정한다. 이를 위해 상단 메뉴의 Repositories 클릭 > 좌측 메뉴 Backup Schedule을 클릭한다. 새로운 백업 잡 생성 새로운 잡을 생성하기 위해 다음 그림과 같이 New Jobs를 클릭 한다. 21.Guide.png Repository List가 나온다 백업 스케줄을 설정 할 대상 Repository를 선택하고 다음과 같이 스케줄 타입을 설정한다. Type of Job : Full Dump Backup, Hotcopy Backup When to Run : Hourly, Daily, Weekly Number to Keep : Use deltas 22.Guide.png 해당 백업을 적용할 Repository를 선택 하고 Create 버튼을 클릭하여 스케줄을 생성한다. 23.Guide.png 백업 스케줄이 생성이 되었고 하단의 Repository의 백업 스케줄 설정한 정보를 확인 할 수 있다. 24.Guide.png 저장소 백업 설정 변경 저장소 백업 설정 변경을 수행하기 위해 백업 스케줄을 변경한다. 이를 위해 상단 메뉴의 Repositories 클릭 > 좌측 메뉴 Backup Schedule을 클릭한다. 아래 그림은 토요일에 AM 2:00에 백업 스케줄로 변경한다고 가정하고 만든 예제 그림이다. 변경이 되었으면 Replace 버튼을 클릭한다. 25.Guide.png 저장소 백업 설정이 변경 되었음을 확인 할 수 있다. 26.Guide.png 저장소 백업 삭제 저장소 백업 삭제를 수행하기 위해 백업 스케줄을 삭제한다. 이를 위해 상단 메뉴의 Repositories 클릭 > 좌측 메뉴 Backup Schedule을 클릭한다. 저장소 백업 스케줄을 삭제 하고 싶은 저장소를 선택 하고 Delete 버튼을 누른다. 27.Guide.png 저장소 백업 스케줄이 삭제 되었음을 확인 할 수 있다. 28.Guide.png 저장소 Hook 설정 사전 조건: Hook 관리 권한을 가진 사용자 계정 필요 csvn/data/conf /security.properties에 편집 가능 설정 hook.script.editor.enabled=true Step 1) Hook 관리 화면 진입 Repositories 화면 >> 저장소 오른쪽의 "OK" 버튼 클릭 Screen Shot 2018-02-20 at 1.20.32 PM.png Step 2) 관리할 hook 선택 원하는 설정에 맞는 버튼 선택 create: 새로운 hook script upload Edit: 등록된 script 수정 Copy: 등록된 script 복사 Rename: 등록된 script 이름 변경 Download: 등록된 script 다운로드 Delete: 등록된 script 삭제 업로드, 이름변경, 복사되는 script는 executable 속성이 지정됨 Screen Shot 2018-02-20 at 1.23.59 PM.png 참고 문서 : Manage repositories <http://help.collab.net/topic/csvn/action/managerrepositories.html>

2018-02-27 08:52:31.179

CollabNet Subversion Manage Users

User 생성 User 생성을 하기 위해 <http://ipaddress:3343> <http://ipaddress:3343> 접속을 한다. 상단의 메뉴 Users 클릭 생성 아이콘을 클릭 한다. 1.Guide.png Create User 아래의 그림은 User를 생성 하는 화면 이다. 각 필드의 대한 설명이다. User를 생성할때 적절한 롤 권한을 설정하여 User를 생성 한다. Login Name Full Name Password Confirm Password Email Description Roles Granted ROLE_ADMIN - Super/Root Administrator (Full Privileges) ROLE_ADMIN_HOOKS - Repository Hook Scripts Administrator ROLE_ADMIN_REPO - Repositories Administrator ROLE_ADMIN_SYSTEM - System/Server Administrator ROLE_ADMIN_USERS - User Account Administrator ROLE_USER - Basic User Authority, required for console access 3.Guide.png User가 생성이 된 것을 확인 할 수 있다. 4.Guide.png User 정보 및 권한 변경 User 정보를 변경 하거나 권한 변경을 하기 위해서 ROLE_ADMIN_USERS 권한을 가진 User가 필요하다. 또는 슈퍼 권한을 가진 User가 일반 유저의 정보나 권한 변경이 가능하다. 아래 그림은 일반 유저를 생성하여 일반 유저를 정보를 변경하거나 권한을 변경하는 예제 이다. 먼저 일반 유저를 생성한다. 5.Guide.png User List의 새로 생성한 일반 user1를 클릭한다. 6.Guide.png 패스워드 변경시 Change Password를 클릭 한다. 7.Guide.png Password 및 정보를 변경하고 권한을 추가 해주고 Update 버튼을 클릭한다. 8.Guide.png User가 변경 되었음을 확인 할 수 있다. 9.Guide.png User 삭제 User 삭제하기 위해서 ROLE_ADMIN_USERS 권한을 가진 User가 필요하다. 또는 슈퍼 권한을 가진 User가 일반 유저를 삭제가 가능하다. 상단 Users 클릭 User List의 삭제 해야 할 유저를 클릭 한다. 10.Guide.png Delete 버튼을 클릭 한다. 11.Guide.png OK 버튼을 클릭 한다. 12.Guide.png 선택한 User가 삭제 되었음을 확인 할 수 있다. 13.Guide.png 참고 문서 Manage users http://help.collab.net/topic/csvn/action/manageusers_csvn.html

2018-02-27 09:45:23.719

CollabNet Edge Reverse Proxy 구성 방법

이 문서는 CollabNet Edge를 reverse proxy로 구성하는 방법을 제공한다. Overview Reverse proxy를 구성하려면 base URL (외부 포트)과 내부 URL (내부 포트)를 다르게 구성할 수 있어야 한다. CollabNet management console은 내부/외부 포트 구성을 제공하지 않는다. 제한 사항 CollabNet이 운영되는 machine에 80 포트를 사용하는 서비스가 없어야 할 CollabNet이 80과 추가의 내부 포트를 사용하게 되기 때문임 Reverse proxy란? 클라이언트가 service.com 웹 서비스에 데이터를 요청하면 Reverse Proxy는 이 요청을 받아서 내부 서버에 (internal.service.com)서 데이터를 받은후에 이 데이터를 클라이언트에 전달하는 방식이다. 공인 IP가 하나인 환경에서 여러개의 서비스가 80 또는 443 포트를 공유할 경우 유용하다. 80 또는 443 포트 전체를 수신하는 proxy server 구성 proxy server는 domain name (또는 IP 주소)을 기준으로 내부 서버에서 데이터를 받아서 요청한 외부 client로 전송 Forward proxy는? 내부 망에서 외부 서버 (service.com)에 연결을 요청하면 proxy server가 외부 서버로 부터 데이터를 수신하여 내부 client에 전달하는 방식이다. 정보보안을 위해 외부로 나가는 http(s) 연결 허용을 제어하거나 패킷을 감시할 때 사용하는 방식이다. 우리가 browser에 설정하는 proxy 서버가 예 Configuration 서비스 base URL과 포트 설정 외부에서 접속하는 관점으로 Hostname (IP도 가능), Port 설정 Screen Shot 2018-02-27 at 9.24.31 AM.png Reverse Proxy 서버에 제공할 포트 설정 Step 1) httpd.conf 에 reverse proxy가 사용할 내부 포트 정의 <CollabNet SVN 설치 경로>/data/conf/httpd.conf Listen <포트> 예) Listen 18080 Step 2) HTTP service 재시작 Administration console에서 제공하는 서비스 제어 방법을 사용하여 HTTP 서비스를 재 시작 Screen Shot 2018-02-27 at 9.34.47 AM.png 구성 확인 내부 URL, 외부 URL 로 정상 접속되는지 확인 내부 URL 접속 확인 Browser로 http(s)://<CollabNet ip 주소>:<내부 포트>/<Browser repositories path: viewvc> 예) http://192.168.0.50:18080/viewvc http://192.168.0.50:18080/viewvc 외부 URL 접속 확인 Administration console >> Repositories 화면에서 제공되는 repository 주소를 browser 또는 svn client 로 접속 시도 Screen Shot 2018-02-27 at 9.37.58 AM.png

2018-03-12 23:06:37.584

Eclipse Che Docker 설치

이 문서는 eclipse che docker 설치 가이드를 공유한다. 먼저 eclipse che 데이터를 저장할 폴더 생성한다. mkdir data cd data mkdir eclipse 다음 명령을 통해 docker container 실행한다. docker run -ti -v /var/run/docker.sock:/var/run/docker.sock -v ~/data/eclipse:/data eclipse/che start docker run -it -e CHE_MULTIUSER=true -v /var/run/docker.sock:/var/run/docker.sock -v ~/data/eclipse:/data eclipse/che start 최종적으로 다음과 같이 실행을 확인할 수 있다. [root@localhost /]# docker run -it --rm -v /var/run/docker.sock:/var/run/docker.sock -v /data/eclipse:/data eclipse/che start WARN: Bound 'eclipse/che' to 'eclipse/che:6.2.0' INFO: (che cli): 6.2.0 - using docker 18.02.0-ce / native INFO: (che config): Generating che configuration... INFO: (che config): Customizing docker-compose for running in a container INFO: (che start): Preflight checks mem (1.5 GiB): [OK] disk (100 MB): [OK] port 8080 (http): [AVAILABLE] conn (browser => ws): [OK] conn (server => ws): [OK] INFO: (che start): Starting containers... INFO: (che start): Services booting... INFO: (che start): Server logs at "docker logs -f che" INFO: (che start): Booted and reachable INFO: (che start): Ver: 6.2.0 INFO: (che start): Use: http://192.168.0.146:8080 INFO: (che start): API: http://192.168.0.146:8080/swagger

2020-05-20 16:27:24.233

Gradle 버전 지정 방법

개요 목적 빌드 서버에 두 가지 이상의 Gradle 버전이 필요한 경우 빌드 할 때 버전 지정 방법을 정리한다. 해결 방안 Gradle 은 어떤 버전을 실행하는지에 따라 다른 버전이 실행된다 방법 1) GRADLE_HOME 경로 구성 원하는 버전을 GRADLE_HOME 변수로 지정할 수 있다 방법 2) Gradle wrapper 구성 gradlew은 gradle/wrapper/gradle-wrapper.properties에 정의된 gradle version을 실행한다. 따라서 gradle-wrapper.properties 파일에 원하는 gradle version을 설정하여 gradle version을 지정할 수 있다. 다음과 같이 지정하면 이후 gradlew를 수행하면 version 3.3이 수행된다. \$ wrapper --gradle-version 3.3

2018-03-02 20:00:13.246

Private DNS 구성 (bind9)

이 페이지는 내부 서버를 위한 DNS 환경 구성을 설명한다. DNS 서버는 Ubuntu를 고려하고 단일 DNS 서버를 구성한다. 패키지 설치 \$ sudo apt-get install bind9 bind9utils bind9-doc /etc/default/bind9 수정 "-f -4" 추가 # startup options for the server OPTIONS="-f -4 -u bind" Options 파일 설정 options { directory "/var/cache/bind"; // ISP's domain name server forwarders { 168.126.63.1; }; dnsec-validation auto; auth-nxdomain no; # conform to RFC1035 listen-on-v6 { any; }; allow-transfer { none; }; }; Local 파일 설정 외부 domain: "curvc.com", local 대신 다른 이름 사용해도 됨 내부 IP 주소: 192.168.0.xxx로 구성된 예 zone "local.curvc.com" { type master; file "/etc/bind/zones/db.local.curvc.com"; }; zone "0.168.192.in-addr.arpa" { type master; file "/etc/bind/zones/db.192.168.0"; //192.168.0.0/24 subnet }; Forward Zone 설정 host.local.curvc.com이 요청될 때 참조하는 설정 설정이 변경할 때 Serial 숫자가 증가되어야 함 \$TTL 604800 @ IN SOA ns1.local.curvc.com. admin.local.curvc.com. (4 ; Serial 604800 ; Refresh 86400 ; Retry 2419200 ; Expire 604800) ; Negative Cache TTL ; name servers - NS records IN NS ns1.local.curvc.com. ; IN NS ns2.local.curvc.com. ; name servers - A records ns1.local.curvc.com. IN A 192.168.0.104 ; ns2.local.curvc.com. IN A 192.168.0.105 ; 192.168.0.0/24 - A records ni.local.curvc.com. IN A 192.168.0.50 Reverse Zone 파일 설정 IP 주소 ("192.168.0.100")으로 DNS를 요청받으면 참조하는 설정 설정이 변경할 때 Serial 숫자가 증가되어야 함 \$TTL 604800 @ IN SOA ns1.local.curvc.com. admin.local.curvc.com. (2 ; Serial 604800 ; Refresh 86400 ; Retry 2419200 ; Expire 604800) ; Negative Cache TTL ; name servers IN NS ns1.local.curvc.com. ; IN NS ns2.local.curvc.com. ; PTR records 104 IN PTR ns1.local.curvc.com. ; 192.168.0.104 ; 105 IN PTR ns2.local.curvc.com. ; 192.168.0.105 50 IN PTR ni.local.curvc.com. ; 192.168.0.50 설정 확인 \$ sudo named-checkconf #named.conf* 설정 확인 \$ sudo named-checkzone local.curvc.com /etc/bind/zones/db.nyc3.example.com #forward zones 설정 확인 \$ sudo named-checkzone 0.168.192.in-addr.arpa /etc/bind/zones/db.192.168.0 서비스 재기동 \$ sudo service bind9 restart Client 설정 내부 DNS 서버 주소를 공인 DNS보다 먼저 설정한다. search local.curvc.com #local domain name nameserver 192.168.0.104 #local DNS 서버 주소 nameserver 203.248.252.2 #공인 DNS 서버 nameserver 164.124.101.2 Example (one dns server) Configuration Files bind9.tar.gz

2022-05-08 20:40:53.308

Linux에서 Non Root 권한으로 MSSQL 서버 접속하기

이 문서는 root 권한이 없는 시스템에서 MS SQL 등의 데이터베이스 서버에 command line으로 접속하는 방법을 정리한다. 바이너리 준비 실제 Linux 환경에 root 권한이 없어 패키지 설치가 불가하므로 별도의 임시 서버에서 바이너리를 준비한다. 실제 구성할 서버와 동일한 OS 버전의 VM 준비 mssql tool 설치 curl <https://packages.microsoft.com/config/rhel/8/prod.repo> > /etc/yum.repos.d/msprod.repo 설치된 파일 묶음 만들기 - MSSQL 도구는 CentOS의 경우 /opt에 설치될 cd /opt tar -zcvf /tmp/mssql.tar.gz mssql_tools microsoft - odbc library 묶음 만들기 cd /usr/lib64 tar -zcvf /tmp/libodbc.tar.gz libodbc.tar.gz libodbc* odbcinst binary 복사 /usr/bin/odbcinst 실제 환경 구성 임시 서버에서 준비된 바이너리를 이용해 실제 환경을 구성한다. 파일 압축 해제 mkdir ~/mssql cd ~/mssql tar -zxvf ~/Downloads/mssql.tar.gz mkdir lib64 cd lib64 tar -zxvf ~/Downloads/libodbc.tar.gz odbc driver 설정 vi ~/mssql/microsoft/msodbcsql17/etc/odbcinst.ini [ODBC Driver 17 for SQL Server] Description=Microsoft ODBC Driver 17 for SQL Server Driver=/home/username/mssql/microsoft/msodbcsql17/lib64/libmsodbcsql-17.9.so.1.1 <-- 바이너리 환경에 맞게 경로 수정 실행 환경 변수 설정 export PATH="\$PATH:\$HOME/mssql/mssql-tools/bin" export LD_LIBRARY_PATH="\$LD_LIBRARY_PATH:\$HOME/mssql/lib64" export ODBC_SYSINI=\$HOME/mssql/microsoft/msodbcsql17/etc 서버 접속 시험 쿼리 파일 생성 USE <database name> GO SELECT TOP (100) * FROM MY_TABLE GO 쿼리 실행 sqlcmd -U <username> -P <password> -S <SQL sever IP>,<port number> -i <쿼리 파일 이름> 예) sqlcmd -U curvc -P 비밀번호 -S 10.100.10.10,1433 -i query.sql 쿼리 결과 저장 CSV 출력 HEADER 포함 sqlcmd -U <username> -P <password> -S <SQL sever IP>,<port number> -i <쿼리 파일 이름> -W -f 65001 -s "," -o "<OutputFilePath>" HEADER 제외 sqlcmd -U <username> -P <password> -S <SQL sever IP>,<port number> -i <쿼리 파일 이름> -W -f 65001 -h -1 -s "," -o "<OutputFilePath>" 참고 <https://www.easysoft.com/support/kb/kb00635.html> <https://www.easysoft.com/support/kb/kb00635.html>

2019-07-04 17:48:12.485

Private DNS 구성 (powerdns + Web)

이 문서는 powerdns / MySQL / web으로 DNS를 구성하는 방법을 제공합니다. Web으로 domain name 관리할 수 있어서 관리자의 수고가 줄어듭니다. Overview 목표 구성 내부 망에서 사용하는 도메인과 공인 도메인 혼용 지원 이를 위해 DHCP 서버 dns 설정에 내부 (private) dns 서버 추가 (primary dns: 첫 번째 dns) 고정 IP 를 사용하는 PC의 경우 dns 서버 추가 (primary dns: 첫 번째 dns) 적용 솔루션 (powerdns) PDNS는 PowerDNS BV사에서 만든 네임서버입니다. 도메인 관리를 database를 통해 도메인 관리를 하기 때문에 대량의 domain 운영이 가능하고 캐쉬 잘못에 의한 오류 가능성이 없습니다. MySQL 설치 (PostgreSQL 도 지원) powerdns는 MySQL과 PostgreSQL 그리고 SQLite를 지원합니다. 이 문서는 MySQL database을 고려합니다. PDNS 설치 전 MySQL을 설치합니다. Database 구성 (MySQL) powerdns database 생성 powerdns database 사용자 생성 powerdns database 사용자에게 powerdns database 액세스 권한 부여 \$ mysql -uroot -p mysql > create database powerdns; mysql > GRANT ALL ON powerdns.* TO 'powerdns'@'localhost' IDENTIFIED BY 'password'; mysql > FLUSH PRIVILEGES; PDNS 설치 설치 과정에서 database 생성을 위해 미리 configuration 파일 생성 Database table 생성이 안되었다면 "/usr/share/doc/pdns-backend-mysql/README.Debian"을 참조하여 수동으로 table 생성 # MySQL Configuration file launch=gmysql gmysql-host=localhost gmysql-database=powerdns gmysql-user=powerdns gmysql-password=password # yum -y install pdns pdns-backend-mysql Database table 생성이 안되었다면 "/usr/share/doc/pdns-backend-mysql/README.Debian"을 참조하여 수동으로 table 생성 PDNS DB 구성 pdns backend (MySQL) 연결 설정 # MySQL Configuration file launch=gmysql gmysql-host=localhost gmysql-database=powerdns gmysql-user=powerdns gmysql-password=password PDNS 서비스 재시작 pdns 서비스가 자동 시작되도록 등록하고 서비스 재시작 \$ chkconfig pdns on \$ service pdns start PDNS 서비스 검증 dig 명령 수행 결과기 다음 예와 유사한지 확인 root@ns1:~# dig @127.0.0.1 <<> DiG 9.9.5-3-Ubuntu <<> @127.0.0.1 : (1 server found) ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 27248 ;; flags: qr rd: QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1 ;; WARNING: recursion requested but not available ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags: udp: 2800 ;; QUESTION SECTION: ; IN NS ;; Query time: 1 msec ;; SERVER: 127.0.0.1#53(127.0.0.1) ;; WHEN: Sun Nov 02 18:58:20 EST 2014 ;; MSG SIZE rcvd: 29 Apple 장치 호환성 웹 서버 설치 PowerAdmin은 web 서버를 포함하지 않고 PHP로 제작된 웹 기능을 제공하기 때문에 웹서버 (Nginx, Apache, ...)를 설치해야 한다. 이 문서는 Apache를 웹 서버로 기준으로 설명한다. \$ yum -y install httpd httpd-devel php php-mcrypt php-pdo php-mysql php-devel php-gd php-imap php-ldap php-odbc php-pear php-xmlrpc php-mbstring gettext wget httpd 서비스 재시작 Apache 서비스가 자동시작되도록 등록하고 재시작 한다. \$ service httpd restart \$ chkconfig httpd on Pear 패키지 설치 \$ yum -y install php-pear-MDB2 php-pear-MDB2-Driver-mysql PowerAdmin (Web 인터페이스) 설치 powerdns의 웹 인터페이스를 제공하는 PowerAdmin을 설치한다. 다운로드 및 압축 풀기 \$ cd /tmp/ \$ wget https://github.com/downloads/poweradmin/poweradmin/poweradmin-2.1.6.tgz \$ tar xvfz poweradmin-2.1.6.tgz \$ mv /var/www/html /var/www/html.bak \$ mv poweradmin-2.1.6 /var/www/html \$ cd poweradmin/inc/ \$ mv config-me.inc.php config.inc.php \$ chown -R www-data:www-data /var/www/html Database 접속 정보 설정 Modify db_pass and session_key to your own values : \$db_host = 'localhost'; \$db_port = '3306'; \$db_user = 'powerdns'; \$db_pass = 'password'; \$db_name = 'powerdns'; \$db_type = 'mysql'; \$session_key = 'Passw0rd'; 설정 적용을 위해 httpd 서비스 재기동 \$ service httpd restart Web을 통해 PowerAdmin 설정 브라우저를 통해 http://<IP> 접속 (예: http://192.168.0.3) Click on Install -> Choose I prefer to proceed in English -> Go to Step2 -> Go to Step3 -> Username: powerdns Password: password Database Type: MySQL Hostname: localhost DB Port: 3306 Database: powerdns PowerAdmin Administrator Password: Passw0rd Go to Step4 -> This step is optional (SKIP) -> Go to Step5 -> Go to Step6 -> Go to Step7. 설치 폴더 제거 /var/www/html/install 폴더를 다른 이름으로 변경 \$ mv /var/www/html/install /var/www/html/install.bak 도메인 관리 로그인 브라우저를 통해 http://<서버 IP> 접속 User name: admin Password: curvc1004 master zone 추가 Zone (domain)을 추가한다. Screen Shot 2018-03-02 at 9.45.07 PM.png reverse zone (PTR) 추가 (옵션) Problem with Record (PTR)은 IP를 기준으로 domain name을 추적할 때 사용되는 기록이다. 등록할 domain의 network (subnet)에 해당하는 zone 생성 예) 192.168.0.0/24 네트워크 이라면 -> 0.168.192.in-addr.arpa 예) 10.0.0.0/8 네트워크 이라면 -> 10.in-addr.arpa Screen Shot 2018-03-02 at 9.49.49 PM.png 도메인 등록 Step 1) list zone >> 원하는 zone의 연필 모양 icon 클릭 Screen Shot 2018-03-02 at 10.01.34 PM.png Step 2) 추가할 도메인 이름과 IP 주소, 그리고 추가 정보 설정 IP 주소에 해당하는 reverse DNS zone을 구성했다면, Add also reverse record 선택 -> 자동으로 reverse DNS (PTR) zone에 추가됨 Screen Shot 2018-03-02 at 10.08.53 PM.png Step 3) 등록 결과 메시지 Screen Shot 2018-03-02 at 10.11.58 PM.png Step 4) 등록 결과 확인 (찾기) Screen Shot 2018-03-02 at 10.14.22 PM.png Troubleshooting Apple 장치에서 외부 주소 얻을 수 없는 경우 구성 후 Apple 장치 (Mac, iOS)에서 정상동작 하지 않을 수 있다. Apple 장치에서 외부 도메인 (예: google.com) 주소를 얻어오지 못하는 현상이 발생할 수 있다. 이 문제는 Recursor 구성을 통해 문제를 해소할 수 있다. Step 1) recursor 설치 \$ yum install dns pdns-recursor Step 2) pdns 설정 변경 recursor 서비스 추가 (recursor=127.0.0.1:5678) launch+=gmysql # gmysql parameters gmysql-host=localhost gmysql-

```
port= gmysql-dbname=powerdns gmysql-user=powerdns gmysql-
password=curvc**** #gmysql-dnssec=yes # gmysql-socket= recursor=127.
0.0.1:5678 Step 3) recursor 설정 forward-zones=i.curvc.com=127.0.0.1:
5678;8.8.8.8 #forward-zones=<내부 전용 도메인, 외부와 공용이라면 '>=<내
부 도메인 서버 주소>;<외부 도메인 서버 주소> local-port=5678 serve-
rfc1918=no trace=on Step 4) 서비스 재시작 pdns와 pdns-recursor 서비스 재시
작
```

2018-03-04 14:29:06.602

SELinux 환경에서 MySQL 데이터 폴더 변경하기

충분한 저장공간 또는 특별한 디스크에 MySQL 데이터 폴더를 위치시키는 경우가 많
다. 이 때 폴더 접근 권한 설정은 SELinux를 고려해야한다. 이 페이지는 SELinux 에
대해 조금 소개하고 MySQL 데이터 폴더를 기본 위치에서 다른 경로로 이동하는 방법
을 소개한다. SELinux 맞보기 Security-Enhanced Linux (SELinux)는 보안 강화를
위한 리눅스 커널 모듈로서 1998년 NSA와 RedHat이 제안하고 레드햇이 개발을 담
당했다. 이 보안 모듈은 2003년 8월 8일 버전 2.6을 기점으로 주류 리눅스 커널에 동
합되었고, CentOS 4에 처음 적용되었다. 해결하고 싶은 문제 SELinux는 아래의 문
제 해결에 관심이 있다: 사용자 제어 불가: 사용자가 임의로 모든 유저자가 읽고 쓸
수 있는 폴더 또는 파일을 생성할 수 있음 프로세스가 보안 특성을 변경 가능: 사용자
의 메일 파일은 해당 사용자만 읽을 수 있지만, 메일 client 프로세스가 해당 파일을
모든 사용자가 읽을 수 있도록 변경 가능 동작 모드 세 가지 동작 모드로 구성된다.
Enforcing: SELinux 보안정책 사용을 강제하고 접근 위반 행위를 기록함
Permissive: SELinux가 사용되지만 보안정책을 강제하지는 않고 접근 위반에 대해
경고하고 위반 행위를 기록함 Disable: SELinux 사용 안함 SELinux 동작 상태 확인
방법: \$ setstatus SELinux status: enabled SELinuxfs mount: /selinux Current
mode: enforcing Mode from config file: enforcing Policy version: 21 Policy
from config file: targeted 정책 SELinux는 기본적으로 접근을 거부하는 least-
privilege 정책을 따른다. SELinux에서는 보안 정책을 "targeted policy"라는 용어
로 정의하고 서로다른 보안정책간의 변환을 허용한다. 접근 제어 SELinux는 4가지 접
근제어 형식 제공: Type Encorcement (TE): primary mechanism Role-Based
Access Control (RBAC): SELinux 사용자 기준의 접근제어 Multi-Level Security
(MLS) Multi-Category Security (MCS): MLS의 확장으로 virtual machine과
container 분류 (compartment)에 사용됨 SELinux context 정보 확인 방법: \$ ls -
Z /var/lib/ drwxr-x--x. mysql mysql system_u:object_r:mysql_db_t:s0 mysql
\$ ps axZ | grep mysqld system_u:system_r:mysql_t:s0 1675 ? Sl 0:00 /usr
/sbin/mysqld MySQL Data 폴더 변경 사전 조건 semanage 설치 \$ yum install
policycoreutils-python Step 1) MySQL 서비스 종료 \$ systemctl stop mysqld
Step 2) 데이터 폴더 이동 설정 변경을 최소화하기 위해 Linux의 symbolic link 사용
새로운 데이터 폴더 위치: /data/mysql-data \$ mv /var/lib/mysql /data/mysql-
data \$ ln -s /data/mysql-data /var/lib/mysql Step 3) SELinux context 설정 \$
semanage fcontext -a -t mysqld_db_t "/data/mysql-data(/.*)?" \$ restorecon
-Rv /data/mysql-data Step 4) MySQL 서비스 시작 \$ systemctl start mysqld

2022-10-01 09:41:11.851

리눅스 루트 파일시스템 복구

루트 파일 시스템이 손상되어 부팅 자체가 안될 경우 파일시스템 복구를 시도하는 방
법을 정리한다. 요약 복구 모드로 부팅 파티션 확인 복구 시도 파일 시스템 확인 재부
팅 복구 복구 모드로 부팅 파일 시스템 오류가 발생하여 마운트에 실패하면 자동으로
복구 모드로 진입한다. 부팅이 가능한 상태라면, 부트 화면에서 "linux rescue" 를 입
력한다. 파티션 확인 lvm 명령어로 진입하여 volume 정보를 확인한다. \$ lvm
(lvm) vgscan Screen Shot 2022-10-01 at 9.34.40 AM.png (lvm) quit 복구 시
도 \$ xfs_repair -L /dev/centos/root 만약, 파일 시스템 마운트가 가능하다면 '-L' 옵션
을 제거한다. \$ xfs_repair /dev/centos/root 파일 시스템 확인 마운트하여 파일시
스템 내용을 확인한다. \$ mkdir /tmp/mnt \$ mount /dev/centos/root /tmp/mnt
재부팅 파란버튼을 눌러 시스템을 재기동한다.

2019-02-16 08:29:37.235

Let's Encrypt 인증서 발급 방법

이 페이지는 Let's Encrypt에서 운영하는 무료 SSL 인증서를 받는 방법에 대해 정리한다. 기존 OS CentOS 7 사전 준비 EPEL repository 설정 Certbot이 EPEL package 이므로 EPEL repository를 사용할 수 있도록 설정한다 \$ yum install epel-release 도메인 관리자 권한 SSL을 받을 때 _acme-challenge.<domain> 에 TXT record를 생성해야 하므로 도메인 관리자 계정이 필요하다. Certbot 설치 \$ yum install python2-certbot-apache 인증서 받기 Certbot은 인증서를 받기 그리고 web server 설정에 반영하기를 지원한다. 본 정리에서는 인증서를 받는 방법에 대해 정리한다. Step 1) certbot 실행 \$ certbot certonly -d <발급하려는 도메인 이름> -d <발급하려는 또 다른 도메인 이름> --manual --preferred-challenges dns-01 --force-renewal --manual-public-ip-logging-ok 발급하려는 도메인 이름 예) *.curvc.com *.almdemo.curvc.com jira.almdemo.curvc.com Step 2) _acme-challenge 레코드 생성 이 단계는 certbot (letsencrypt)가 도메인 소유주가 맞는지 확인하는 절차로써 certbot이 지정한 문자열을 지정된 도메인 (_acme-challenge.-) 레코드로 생성해 하고 certbot이 이를 확인하는 과정이다. ----- Please deploy a DNS TXT record under the name _acme-challenge.almdemo.curvc.com with the following value: ZFETHF7O9RWoct4tk8jkgrXlqjbZMBA0J1WLW5OITv4 Before continuing, verify the record is deployed. ----- Press Enter to Continue 도메인 서비스의 zone edit menu에서 _acme-challenge.<발급하려는 도메인 이름>에 TXT 형식의 레코드를 생성한다. 예) *.curvc.com 이라면: _acme-challenge.almdemo.curvc.com 에 TXT 값 (ZFETHF7O9RWoct4tk8jkgrXlqjbZMBA0J1WLW5OITv4) 레코드 생성 (기존에 존재하면 TXT 값 수정) 일반적으로 도메인 서버에 반영되기까지 시간이 필요하므로 충분히 (2분 이상) 기다리거나, 아래 방법으로 반영 여부를 확인후 Enter를 입력해 다음 단계로 진행한다. \$ dig -t TXT _acme-challenge.almdemo.curvc.com ; <<>> DiG 9.10.6 <<>> -t TXT _acme-challenge.almdemo.curvc.com ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<< opcode: QUERY, status: NOERROR, id: 26540 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0 ;; QUESTION SECTION: ;_acme-challenge.almdemo.curvc.com. IN TXT ;; ANSWER SECTION: _acme-challenge.almdemo.curvc.com. 301 IN TXT "RJULHF7O9RWoct4tk8jkgrXlqjbZMBA0J1WLW5OITv4" ;; Query time: 298 msec ;; SERVER: 10.0.0.3#53(10.0.0.3) ;; WHEN: Thu Sep 13 18:31:11 KST 2018 ;; MSG SIZE rcvd: 107 13 line과 같이 설정했던 값이 확인되면 다음 단계로 진행한다. Step3) 인증서 확인 다음 문구와 같이 발급된 인증서가 저장된 위치를 알려준다. 예) /etc/letsencrypt/live/almdemo.curvc.com IMPORTANT NOTES: - Congratulations! Your certificate and chain have been saved at: /etc/letsencrypt/live/almdemo.curvc.com/fullchain.pem Your key file has been saved at: /etc/letsencrypt/live/almdemo.curvc.com/privkey.pem Your cert will expire on 2018-12-12. To obtain a new or tweaked version of this certificate in the future, simply run certbot again. To non-interactively renew *all* of your certificates, run "certbot renew" Step 4) 인증서 사용 저장된 인증서를 필요한 곳에 사용하면 됨 인증서 갱신 \$ certbot renew 참고 <https://certbot.eff.org/lets-encrypt/centosrhel7-apache> <https://certbot.eff.org/lets-encrypt/centosrhel7-apache>

2018-05-16 12:37:58.171

CentOS 6 Mac Terminal에서 Copy/paste 시 0~ 1~ 문자 추가되는 등

환경 Mac의 terminal App을 이용해 ssh 프로토콜로 CentOS 6 접속 CentOS 6.9 Mac: High Sierra 현상 command를 복사하여 terminal에 붙여넣으면 0~ 1~ 추가 될 \$ > 0~command 1~ 원인 Terminal의 bracket 붙여넣기 모드로 설정되어 있기 때문 해결 Terminal에 아래 명령 입력 printf "we[?2004]"

2020-11-01 09:27:51.028

복수의 Java 버전 운영하는 방법

서버에 여러가지 Java 버전을 설치하는 경우가 자주 발생하는데, alternatives 도구를 이용해 간편하게 버전을 전환하는 방법을 정리한다. 적용 대상 CentOS, Red Hat Linux Ubuntu의 경우 alternative 대신 update-alternatives 명령어 사용 Java 설치 후 다음 절차를 통해 java, javac 등의 java 실행 파일 경로 등록 java 등록 \$ sudo alternatives --install /usr/bin/java java <java 설치 폴더>/bin/java <우선 순위> 예) /usr/lib/jvm/open-jdk-11 에 설치한 경우 \$ sudo alternatives --install /usr/bin/java java /usr/lib/jvm/open-jdk-11/bin/java 100 javac 등록 \$ sudo alternatives --install /usr/bin/javac javac <java 설치 폴더>/bin/javac <우선 순위> 예) /usr/lib/jvm/open-jdk-11 에 설치한 경우 \$ sudo alternatives --install /usr/bin/javac javac /usr/lib/jvm/open-jdk-11/bin/javac 100 jar 등록 \$ sudo alternatives --install /usr/bin/jar jar <java 설치 폴더>/bin/jar <우선 순위> 예) /usr/lib/jvm/open-jdk-11 에 설치한 경우 \$ sudo alternatives --install /usr/bin/jar jar /usr/lib/jvm/open-jdk-11/bin/jar 100 JAVA_HOME, JRE_HOME 등록 환경 설정 파일에 JAVA_HOME 환경변수를 지정한다.

```

JAVA_HOME=$(dirname $(dirname $(readlink $(readlink $(which java))))))
export JAVA_HOME export CLASSPATH=.:$JAVA_HOME/jre/lib:$JAVA_HOME/lib:$JAVA_HOME/lib/tools.jar Linux의 경우 JAVA_HOME, JRE_HOME 아래의 방법으로 환경설정이 가능하다. 본 문서는 profile.d 방식을 예로 설명한다. 시스템 수준 설정 /etc/profile /etc/profile.d/java.sh ← bash login shell에 적용됨 사용 수준 설정 ~/.bashrc ~/.bash_profile Java 버전 전환 설치된 java 버전 중 특정 버전을 선택할 수 있다. $ sudo alternatives --config java
There are 3 programs which provide 'java'. Selection Command
----- * 1 /usr/java/jdk1.8.0_151/jre/bin/java
+ 2 /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.212.b04-0.el7_6.x86_64/bin/java
+ 3 /opt/jdk1.8.0_144/bin/java + : 사용자 지정 default, 수동 설정이 우선
* : 자동으로 설정된 default Alternatives 부가 기능 등록된 경로 제거 $ alternatives --remove <제거할 대상> 예) $ alternatives --remove /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.212.b04-0.el7_6.x86_64/bin/java

```

2022-03-30 05:55:55.651

리눅스 팁

이 페이지는 Linux와 관련된 팁을 정리한다.

2018-06-02 17:39:37.961

MySQL Resource Monitoring

이 페이지는 모니터링 해야하는 MySQL metric에 대해 정리한다. Monitoring Metrics MySQL provides a few good metrics for monitoring your connections: Variable What it represents Why you should monitor it
 Threads_connected This variable indicates the total number of clients that have currently open connections to the server. It provides real-time information on how many clients are currently connected to the server. This can help in traffic analysis or in deciding the best time for a server re-start.
 Threads_running The number of threads that are not sleeping. It's good for isolating which connected threads are actively processing queries at any given time, as opposed to connections that are open but are currently idle.
 Connections The number of connection attempts (successful or not) to the MySQL server. It can give you a good idea of how many people and applications are accessing the database. Over time, these numbers reveal busiest times and average usage numbers.
 Connection_errors_internal The number of connections refused due to internal server errors, such as failure to start a new thread or an out-of-memory condition. Although MySQL exposes several metrics on connection errors, Connection_errors_internal is probably the most useful, because it is incremented only when the error comes from the server itself. Internal errors can indicate an out-of-memory condition or an inability to start a new thread.
 How to monitor We can use the MySQL show status command to show MySQL variables and status information. Here are a few examples: SHOW GLOBAL STATUS LIKE '%Threads_connected%'; +-----+-----+ | Variable_name | Value | +-----+-----+ | Threads_connected | 2 | +-----+-----+
 SHOW GLOBAL STATUS LIKE '%Threads_running%'; +-----+-----+ | Variable_name | Value | +-----+-----+ | Threads_running | 1 | +-----+-----+
 SHOW GLOBAL STATUS LIKE 'Connections'; +-----+-----+ | Variable_name | Value | +-----+-----+ | Connections | 20 | +-----+-----+

2020-03-16 17:32:23.127

Apache Disabling TLS 1.0

개요 2018. 7. 30에 TLS 1.0 지원이 종료되었다. 이 페이지는 Apache webserver 에서 TLS 1.0 을 사용하지 않도록 설정하는 방법을 정리한다. /etc/httpd/conf.d/ssl.conf 변경 전 # SSL Engine Switch: # Enable/Disable SSL for this virtual host. SSLEngine on # List the protocol versions which clients are allowed to connect with. # Disable SSLv3 by default (cf. RFC 7525 3.1.1). TLSv1 (1.0) should be # disabled as quickly as practical. By the end of 2016, only the TLSv1.2 # protocol or later should remain in use. SSLProtocol all -SSLv3 SSLProxyProtocol all -SSLv3 \$ nmap --script ssl-enum-ciphers -p 443 localhost | grep TLSv | TLSv1.0: | TLSv1.1: | TLSv1.2: 변경 후 # SSL Engine Switch: # Enable/Disable SSL for this virtual host. SSLEngine on # List the protocol versions which clients are allowed to connect with. # Disable SSLv3 by default (cf. RFC 7525 3.1.1). TLSv1 (1.0) should be # disabled as quickly as practical. By the end of 2016, only the TLSv1.2 # protocol or later should remain in use. SSLProtocol TLSv1.2 SSLProxyProtocol TLSv1.2 \$ nmap --script ssl-enum-ciphers -p 443 localhost | grep TLSv | TLSv1.2:

2018-03-29 17:19:05.593

VMWare 환경에서 CentOS 7 Root Partition Resizing

Created by Fredrik Mikker <https://kb.op5.com/display/~fmikker>, last modified by Martin Sharples <https://kb.op5.com/display/~msharples> on Jul 25, 2017 <https://kb.op5.com/pages/diffpagesbyversion.action?pageld=5375707&selectedPageVersions=15&selectedPageVersions=16> Go to start of metadata <https://kb.op5.com/display/COMMUNITY/Extend+root+partition+using+LVM#page-metadata-start> Version This article was written for version 6 of OP5 Appliance System, it could work on both lower and higher version if nothing else is stated. Articles in community space is not covered by OP5 Support When running virtual machines there might be a need to add space your virtual machine when it's running low, in this example we will add 10GB to the root filesystem of a virtual instance of OP5 Appliance System 6.5. Prerequisites To follow this

article the following will be needed: OP5 Appliance System 6 or CentOS 6
 As always when working with storage, make sure you have a recent backup of the virtual machine to be safe if anything should go wrong. No snapshots can be present on the VM when resizing the disk Basic knowledge about Linux and LVM2 Shut down your virtual machine Change the setting "Provisioned Size" in VMware to the total size that you want to have available in the operating system of your virtual server: <https://kb.op5.com/download/attachments/5375707/2016-11-15-001-lvm-provision-vdisk.png?version=1&modificationDate=1479224135000&api=v2> Start the virtual machine and log in as root to check that size on your disk has changed: # fdisk -l /dev/sda Disk /dev/sda: 107.4 GB, 107374182400 bytes 255 heads, 63 sectors/track, 13054 cylinders Units = cylinders of 16065 * 512 = 8225280 bytes Sector size (logical/physical): 512 bytes / 512 bytes I/O size (minimum/optimal): 512 bytes / 512 bytes Disk identifier: 0x000076f1 Device Boot Start End Blocks Id System /dev/sda1 * 1 17 131072 83 Linux Partition 1 does not end on cylinder boundary. /dev/sda2 17 8355 66976768 8e Linux LVM In this example, the disk /dev/sda has been extended to 107.4 GB and there are only two primary partitions currently in use, this is good because there are room for two more primary partitions. What you need to do now is to create the new partition sda3 that disk size can be taken from when extending the logical volume later on. This will be done with fdisk The default values here should be sane, just press enter after the command input. # fdisk /dev/sda WARNING: DOS-compatible mode is deprecated. It's strongly recommended to switch off the mode (command 'c') and change display units to sectors (command 'u'). Command (m for help): n Command action e extended p primary partition (1-4) p Partition number (1-4): 3 First cylinder (8355-13054, default 8355): Using default value 8355 Last cylinder, +cylinders or +size{K,M,G} (8355-13054, default 13054): Using default value 13054 In the previous step, a new partition was created from the space that was added from VMware, and now we need to set the correct partition type before writing the partition table and saving the changes: Command (m for help): t Partition number (1-4): 3 Hex code (type L to list codes): 8e Changed system type of partition 3 to 8e (Linux LVM) Command (m for help): w The partition table has been altered! Calling ioctl() to re-read partition table. WARNING: Re-reading the partition table failed with error 16: Device or resource busy. The kernel still uses the old table. The new table will be used at the next reboot or after you run partprobe(8) or kpartx(8) Syncing disks. The warning message above can be addressed in two ways: 1) Reboot the system, 2) Re-scan the device and add the new partition. We will do the latter: # partx -v -a /dev/sda device /dev/sda: start 0 size 209715200 gpt: 0 slices dos: 4 slices # 1: 2048- 264191 (262144 sectors, 134 MB) # 2: 264192-134217727 (133953536 sectors, 68584 MB) # 3: 134217728-209712509 (75494782 sectors, 38653 MB) # 4: 0- -1 (0 sectors, 0 MB) BLKPG: Device or resource busy error adding partition 1 BLKPG: Device or resource busy error adding partition 2 added partition 3 The partx application will give some errors for the existing partitions, but that's nothing to worry about. The important part is that partition 3 was successfully added. If partition 3 isn't added you need to reboot the machine to make the changes take effect. Now we need to create a new physical volume that can be used for adding space to the volume group # pvcreate /dev/sda3 Physical volume "/dev/sda3" successfully created Check what name your volume group your root filesystem uses: # vgdisplay --- Volume group --- VG Name 1 System ID Format lvm2 [...] In this case the Volume group is called 1 Now, lets extend the volume group with the space we added with vgextend: # vgextend 1 /dev/sda3 Volume group "1" successfully extended With the utility pvscan you can see what physical volumes are available, and how much free space there is within them. The free space can be used to extend your root partion, or other volumes that are short on space. # pvscan PV /dev/sda2 VG 1 lvm2 [63,87 GiB / 29,87 GiB free] PV /dev/sda3 VG 1 lvm2 [36,00 GiB / 36,00 GiB free] Total: 2 [99,87 GiB] / in use: 2 [99,87 GiB] / in no VG: 0 [0] Now in this step we will at 10G to the root partition. First find out what logical volume that the root partition belongs to: # df -h Filesystem Size Used Avail Use% Mounted on /dev/mapper/1-sys 7,8G 1,4G 6,0G 20% /tmpfs 246M 0 246M 0% /dev/shm /dev/sda1 120M 27M 88M 24% /boot /dev/mapper/1-opt 7,8G 181M 7,2G 3% /opt /dev/mapper/1-tmp 7,8G 19M 7,4 G 1% /tmp /dev/mapper/1-var 7,8G 475M 6,9G 7% /var In this case, the root partition originates from the logical volume "sys", and the total size is 7.8 GB. The following command will add 10 GB to the Logical Volume "sys" in the Volume Group "1" and the disk area will be taken from the Physical Volume sda3. # lvextend -L+10G /dev/mapper/1-sys /dev/sda3 Size of logical volume 1/sys changed from 8,00 GiB (2048 extents) to 18,00 GiB (4608 extents). Logical volume sys successfully resized. Run the following command to expand the ext4 filesystem inside of the Logical Volume: # xfs_growfs -d /dev/mapper/cl-root meta-data=/dev/mapper/cl-root isize=512 agcount=4, agsize=2293504 blks = sectsz=512 attr=2, projid32bit=1 = crc=1 finobt=0 spinodes=0 data = bsize=4096 blocks=9174016, imaxpct=25 = sunit=0 swidth=0 blks

```

naming =version 2          bsize=4096  ascii-cj=0  ftype=1
log      =internal        bsize=4096  blocks=4479,
version=2          =          sectsz=512  sunit=0  blks, lazy-count=1
realtime =none           extsz=4096  blocks=0, rtextents=0 data blocks
changed from 9174016 to 28865536 And voli! Now your root partition is
extended with 10 GB: # df -h Filesystem      Size Used Avail Use%
Mounted on /dev/mapper/1-sys 18G 1,5G 16G 9% /
tmpfs      246M  0 246M  0% /dev/shm /dev
/sda1      120M  27M  88M  24% /boot /dev/mapper/1-opt 7,8
G 181M  7,2G  3% /opt /dev/mapper/1-tmp 7,8G 19M 7,4G  1% /tmp
/dev/mapper/1-var 7,8G 475M 6,9G  7% /var

```

2022-03-30 05:58:01.898

쓰래드 개수 확인하기

특정 프로세스의 thread 개수 확인하는 방법 `ps -o thcount <pid>` 지속적으로 개수 모니터링 하는 방법 `watch` 명령은 주어진 명령어를 주기적으로 실행하여 출력을 화면에 표시한다. default 주기: 2초 종료 방법: `Ctrl + C` `watch ps -o thcount <pid>`

2024-02-06 08:35:38.181

Command 실행 메뉴 추가하기

이 문서는 Eclipse 에 Windows batch 스크립트 실행 메뉴를 추가하는 방법을 정리한다. 배경 반복적으로 Windows batch 스크립트 실행이 필요한 경우 Eclipse에서 제공하는 External Tool 구성을 통해 실행 메뉴를 추가하여 실행할 수 있다. 구성 `Run > External Tools > External Tools Configurations` Screenshot 2024-02-05 at 11.09.31PM.png New Configuration Screenshot 2024-02-06 at 8.25.18AM.png 실행 내용 구성 Name: 표시할 메뉴 이름 설정 Main Tab Location: 실행 파일 (스크립트 포함) 경로 지정 Workspace 또는 특정 파일 시스템 경로 지정 가능 Working Directory: 실행 경로 지정 Workspace 또는 특정 파일 시스템 경로 지정 가능 Arguments: 실행 파일에 전달할 변수 지정 가능 Screenshot 2024-02-06 at 8.26.54AM.png Refresh Tab 실행 후 갱신할 범위를 지정한다. Screenshot 2024-02-05 at 11.10.34PM.png 실행 추가한 구성은 Run 메뉴에 노출된다. Screenshot 2024-02-06 at 8.33.12AM.png

2017-02-22 20:37:01.764

Redmine 사용자 가이드

이 문서는 Redmine의 일반 사용자에게 대한 가이드 제공한다. 1. Personal Information 1.1 My Account 나의 계정(My Account)에서는 계정의 기본정보, 이메일 알림 설정, 언어 등의 설정을 수행할 수 있다. 나의 계정으로 접속하기 위해서 상단 자신의 아이디 옆 My Account 계정을 선택한다. Information First name : 이름 Last name : 성 Email : 이메일 Language : 사용하고자 하는 언어 선택 Email notifications For any event on all my projects 나의 모든 프로젝트의 알림을 받음 For any event on the selected projects only... 선택된 프로젝트만 알림을 받음 Only for things I watch or I'm involved in 내가 포함되어 있거나 보고 있는 이슈 Only for things I am assigned to 내가 할당된 이슈 Only for things I am the owner of 내가 소유자인 이슈 No events 받지 않음 Preferences Hide my email address : 이메일 주소 숨김 Time zone : 표시를 원하는 시간 대역 Display comments : 코멘트 출력 Warn me when leaving a page with unsaved text : 저장 알림 image2017-2-21_16-33-10.png 1.2 My Page 마이 페이지(My Page)는 프로젝트에 대한 다양한 정보를 보여준다. 기본적으로 다음 두개의 정보 블록을 제공한다. Issue assigned to me 전체 프로젝트에서 현재 사용자에게 할당된 이슈 리스트를 보여준다. Reported issues 전체 프로젝트에서 현재 사용자가 보고 받는 모든

이슈 리스트를 보여준다. 상단 Personalize this page 버튼을 클릭하면, 개인이 원하는 불록을 추가할 수 있다. Watched Issues 현재 사용자가 watch하고 있는 이슈 리스트 제공 Latest news 최근 뉴스 overview 제공 Calendar Weekly Calendar overview 제공 Documents 최신 문서의 overview 제공 Spent time 현재 사용자의 최근 7일 동안의 전체 소비한 시간 제공 image2017-2-22_14-23-53.png 2. Project Overview 프로젝트 Overview 페이지는 프로젝트의 전체 Overview를 보여준다. Issue tracking 얼마나 많은 업무들이 Open되고 Close되어 있는지를 나타낸다. Member 해당 프로젝트의 멤버가 누군지를 나타낸다. Latest news 해당 프로젝트의 최신 뉴스를 보여준다. Progress 프로젝트 Roadmap의 각 버전별 진행상황을 보여준다. Spent time 프로젝트의 전체 작업 소비 시간을 보여준다. image2017-2-22_14-23-1.png 3. Issue Tracking 이슈는 Redmine 프로젝트를 구성하는 핵심 사항이다. 하나의 이슈는 프로젝트에 종속되며, 한명의 사용자에게 소유되고 하나의 버전과 연관된다. 3.1 Issue List 프로젝트 메뉴의 Issues를 선택하면 현재 프로젝트의 이슈 리스트를 확인할 수 있다. image2017-2-22_14-39-8.png 3.1.1 Filters Filters에서는 이슈를 다양한 조건으로 필터하여 리스트를 보여준다. image2017-2-22_14-42-44.png 3.1.2 Options Options은 이슈 리스트에 몇몇 옵션들을 추가하여 리스트를 보여준다. Columns : 이슈 리스트에 보여줄 컬럼 항목을 설정한다. Group results by : 설정한 항목에 따라 이슈를 그룹하여 보여준다. Show : 이슈 리스트에 Description을 포함한다. image2017-2-22_14-44-8.png 3.2 Issue 생성 이슈를 생성하기 위해 프로젝트 메뉴에서 New issue 링크를 선택한다. 기본적으로 제공되는 Issue의 항목은 다음 표와 같다. 필드 필수여부 설명 Private 옵션 개인 이슈로 만든다. Tracker 필수 이슈의 타입을 선택한다. Subject 필수 이슈의 주제(제목)를 입력한다. Description 옵션 이슈의 세부적인 설명을 입력한다. Status 필수 이슈의 상태를 선택한다. Priority 필수 이슈의 중요도 혹은 우선순위를 선택한다. Assignee 옵션 이슈의 담당자를 선택한다. Target version 옵션 이슈가 적용될 버전을 선택한다. Parent task 옵션 만약 Sub Task일 경우 부모 Task를 선택한다. Start date 옵션 시작 날짜를 입력한다. Due date 옵션 종료 날짜를 입력한다. Estimated time 옵션 일의 시간을 추정한다. % Done 옵션 업무의 완료 정도를 %로 입력한다. Files 옵션 이슈와 관련된 파일을 업로드 한다. Watchers 옵션 이슈를 함께 볼 사용자를 선택한다. 다음은 이슈 생성화면의 예시를 보여주고 있다. image2017-2-22_20-31-1.png 이슈 생성 화면의 이슈 생성을 위한 버튼은 3가지가 존재한다. Create : 해당 이슈를 생성한다. Create and continue : 이슈 생성 후 다른 이슈를 다시 생성한다. Preview : 이슈의 Description에 대한 미리보기를 제공한다. 3.3 이슈 세부사항 이슈 리스트에서 해당 이슈를 선택하면, 이슈의 세부사항을 확인할 수 있다. image2017-2-22_15-9-0.png 이슈 화면의 상단 메뉴에서는 다음을 수행할 수 있다. Edit : 이슈 내용을 수정한다. Log time : 이슈에 대한 작업 시간을 기록한다. Watch : 해당 이슈를 Watch한다. 이슈에 대한 변경 사항이 발생하면 알림 메일을 받을 수 있다. Copy : 이슈를 복제하여 새로운 이슈를 만든다. Delete : 이슈를 삭제한다. 3.4 이슈 링크 레드마인에서는 이슈간의 관계를 다양한 형태로 정의하고 설정할 수 있다. 특정 이슈와의 링크가 설정된 이슈는 Related issues에 나타나며, 이슈의 링크 관계는 다음과 같이 표현될 수 있다. Related to - 관련된 이슈를 연결 Duplicates - 중복되는 이슈를 연결 : 만약 B 이슈가 duplicates A이슈라면, A가 Close되면 B가 자동으로 Close된다. Duplicated by - 중복되는 이슈를 연결 : 만약 A 이슈가 duplicated by B 이슈라면, A가 Close되면 B가 자동으로 Close된다. Blocks - A이슈가 block B이슈라면, A는 B가 Close되기전에 Close할 수 없다. Blocked by - B이슈가 blocked by A이슈라면, A는 B가 Close되기전에 Close할 수 없다. Precedes / Follows - 이슈의 우선순위 관계를 나타내는 링크 Copied from / to - 특정 이슈를 복사했을 경우 image2017-2-22_15-32-42.png 4. Activity Activity에서는 프로젝트에 일어나고 있는 전체 활동을 시간 순으로 나타낸다. image2017-2-22_15-50-46.png Activity의 범위는 좌측 Activity 메뉴에서 선택 후 적용하면 확인할 수 있다. 다음은 Activity에 추가할 수 있는 항목을 보여준다. Issues Changesets News Documents Files Wiki edits Messages Spent time 5. Roadmap 5.1 Roadmap List 레드마인의 로드맵은 프로젝트의 개발을 계획하고 관리하기 위해 버전 기반으로 이슈를 구성하는 기능을 제공한다. 현재 프로젝트의 로드맵을 확인하고 버전을 설정하기 위해 프로젝트 메뉴에서 Roadmap 메뉴를 선택한다. 다음은 Roadmap에 대한 예제 화면을 보여주고 있다. image2017-2-22_16-15-4.png 로드맵 페이지에는 각 버전에 대한 제목, 종료일, 설명, 진행상태, 이슈를 제공한다. 좌측 메뉴에서 Version에 대한 특정 이슈를 보거나 특정 버전만을 볼 수 있는 필터를 제공한다. 5.2 New Version 프로젝트에 새로운 버전을 추가하기 위해 Roadmap 화면 상단의 New version 링크를 클릭한다. 아래 그림과 같이 New version 페이지가 나타나면 다음을 참고하여 필요한 정보를 입력한다. Name : 버전의 이름으로 프로젝트 혹은 조직의 정책에 따라 명명한다. Description : 버전에 대한 간단한 설명을 입력한다. Status open : 버전이 열려 있는 상태 locked : 버전이 lock 되어 있는 상태 - 버전에 이슈를 할당할 수 없다. close : 버전이 종료된 상태 - 버전에 이슈를 할당할 수 없고 reopen 할 수 없다. Wiki page : 버전에서 참고할 수 있는 위키 페이지 Date : 버전 종료 예정일 Sharing Not shared : 버전을 공유하지 않는다. With subprojects : 버전을 하위 프로젝트와 공유한다. With project hierarchy : 버전을 hierarchy 관계에 있는 모든 프로젝트와 공유한다. (조상과 자식) With project tree : 프로젝트 트리 상의 모든 프로젝트와 공유한다. (조상, 자식, 조상의 자식) With all projects : 모든 프로젝트와 버전을 공유한다. image2017-2-22_20-32-13.png 6. Gantt 간트는 할당된 버전을 기반으로 이슈들의 시작일과 종료일을 간트 차트에 보여준다. 간트차트의 일정을 수정하기 위해서는 각 이슈의 시작일과 종료일을 수정함으로써 수정할 수 있다. Filters에서는 Gantt에 보여지는 이슈의 리스트를 필터할 수 있으며, Options에서는 이슈의 Blocks과 Precedes와 Progress line을 설정할 수 있다. 간트차트 상단 메뉴를 통해 Gantt차트의 범위 설정과 간트차트를 확대 및 축소할 수 있다. image2017-2-22_16-45-3.png 7. Calendar Calendar는 현재 프로젝트의 월별 이슈의 Overview를 제공한다. 프로젝트의 Calendar에 접근하기 위해 프로젝트 메뉴에서 Calendar 메뉴를 선택한다. Calendar에서는 모든 이슈의 시작일과 종료일을 나타내며 모든 버전의 종료일을 표시한다. 다음 그림은 Calendar의 예시를 보여준다. image2017-2-22_16-54-33.png 8. News, Documents, Wiki, File 8.1 News News는 프로젝트에 관련된 문서를 게시판과 같은 형태로 출력할 수 있게 해주는 모듈이다. News로 접근하기 위해 프로젝트 메뉴에서 News 메뉴를 선택한다. 다음 그림은 News의 예시를 보여주고 있다. image2017-2-22_20-33-1.png News 페이지의 상단 메뉴를 통해 News를 추가하

거나 자신을 Watch로 추가할 수 있다. Add news : 새로운 뉴스를 추가한다.
 Watch : 뉴스 페이지를 Watch 한다. 8.2. Documents Documents는 프로젝트의 사용자 문서 혹은 기술 문서와 같은 문서를 작성할 수 있는 모듈이다. Documents로 접근하기 위해 프로젝트 메뉴에서 Documents 메뉴를 선택한다. New document : 새로운 Document를 생성하기 위해 Document 페이지의 상단에서 New document 를 선택한다. Document category : Document category는 기본적으로 사용자 문서와 기술 문서 두 가지로 설정되어 있으며, 필요에 따라 관리자 페이지에서 enum 값을 수정하거나 추가할 수 있다. 8.3 Wiki 8.3.1 새로운 페이지 생성 레드마인 Wiki에서 새로운 페이지 생성하기 위해서는 생성하고자 하는 페이지에서 다음 link 마크업을 추가하면 된다. [[MyNewWikiPage]] 8.3.2 Wiki 톨바 레드마인 Wiki의 상단에는 다음 톨바들을 제공한다. Edit : 현재 페이지의 콘텐츠를 수정한다. Watch : 자신을 현재 페이지에 대한 Watcher로 추가한다. 이를 통해 변경사항에 대한 알림 메일을 받을 수 있다. Rename : 위키페이지의 이름을 변경하거나 이동할 수 있다. Lock : Lock을 걸고 페이지를 수정할 수 있다. Delete : 해당 위키 페이지를 삭제한다. History : 위키 페이지에 대한 수정 사항을 확인하고 비교할 수 있다. 다음 그림은 위키 페이지의 수정 히스토리에 대한 예제 화면을 보여준다. image2017-2-22_19-9-3.png 8.3.3 Sidebar 레드마인 Wiki의 사이드바는 다음 링크를 제공한다. Start page : 설정된 시작 페이지로 이동한다. Index by title : 위키 페이지를 이름 순으로 정렬하여 보여준다. Index by date : 위키 페이지를 작성일 순으로 정렬하여 보여준다. 8.3.4 Wiki Markdown 이변절에서는 레드마인에서 사용할 수 있는 기본적인 Wiki 마크업에 대해서 설명한다. Ckeditor 혹은 HTML을 지원하는 에디터의 경우에는 HTML 문법을 사용하여 Wiki 페이지를 작성할 수 있다. 텍스트 스타일은 다음과 같은 방식으로 적용할 수 있다. = Heading 1 = == Heading 2 == === Heading 3 === ===== Heading 4 ===== ===== Heading 5 ===== ===== Heading 6 ===== 불릿 리스트는 다음과 같은 방식으로 적용할 수 있다. * Item1 * Item2 * Item3 * Item4 ** Sub-item 4 a) *** Sub-item 4 a) 1. **** Sub-item 4 a) 1. i) **** Sub-item 4 a) 1. ii) ** Sub-item 4 b) * Item5 넘버 리스트는 다음과 같은 방식으로 적용할 수 있다. # Item1 # Item2 # Item3 # Item4 ## Sub-item 1 ### Sub-sub-item #### Sub-sub-sub-item ## Sub-item 2 # Item5 리치 텍스트는 다음과 같은 방식으로 적용할 수 있다. "italicize text" "bold the text" 레드마인의 Markdown에 대한 자세한 정보는 다음 링크를 참조한다. http://www.redmine.org/projects/redmine/wiki/RedmineTextFormattingMarkdown http://www.redmine.org/projects/redmine/wiki/RedmineTextFormattingMarkdown 8.4 Files 레드마인의 Files 모듈에서는 프로젝트에 필요한 파일들을 첨부하고 공유할 수 있다. Files 모듈의 파일 리스트는 다음 정보를 함께 제공한다. File - 파일명 Date - 업로드 날짜 Size - 파일 사이즈 D/L - 다운로드 수 MD5 - MD5 해시 정보 9. Repository 레드마인의 Repository 모듈에서는 프로젝트의 소스 코드 저장소와 최신 Commit을 확인할 수 있게 해준다. 9.1 코드 탐색 디렉토리의 + 버튼을 통해 확장하여 브라우징할 수 있으며, 디렉토리를 더블 클릭하거나 Enter 버튼을 클릭하여 디렉토리로 진입할 수 있다. image2017-2-22_20-33-48.png 9.2 Revision 하단 Latest revisions에서는 변경 히스토리를 확인할 수 있고 각 revision 버전을 선택하여 세부사항을 확인할 수 있다. revision 간의 차이점은 비교를 원하는 revision을 선택하고 View differences 버튼을 클릭하여 비교할 수 있다. 다음은 리버전의 차이점을 side by side view로 보여주는 예시이다. image2017-2-22_20-34-36.png 9.3 Statistics Repository 페이지 상단의 Statistics를 선택하여 월간 Commit 추이와 사용자별 Commit 수에 대한 차트를 확인할 수 있다. 10. Project Settings 10.1 Information 프로젝트 세팅의 Information 탭은 프로젝트의 대한 일반적인 사항에 대한 설정을 제공한다. Name : 프로젝트 이름을 설정한다. Description : 프로젝트에 대한 설명을 설정한다. Identifier : 프로젝트 식별자를 확인한다. Homepage : 관련 홈페이지의 링크를 설정한다. Public : 프로젝트를 공용으로 설정한다. Subproject of : 서브프로젝트로 설정한다. Inherit members : 사용자를 상속 받는다. Trackers : 사용할 이슈 타입을 선택한다. image2017-2-22_20-35-58.png 10.2 Modules Modules 탭에서는 현재 프로젝트에서 사용할 모듈들을 선택할 수 있게 해준다. 모듈을 플러그인 설치 여부에 따라 다르게 나타날 수 있다. 10.3 Members Members 탭에서는 현재 프로젝트의 멤버와 역할을 추가/수정/삭제 할 수 있다. 다음 그림은 Members 탭의 예제 화면을 보여준다. image2017-2-22_19-40-8.png 10.4 Versions 버전 탭에서는 현재 프로젝트에서 사용되고 있는 전체 버전의 정보를 보여준다. 리스트에서 함께 제공되는 정보는 Version 명, 릴리즈 날짜, 설명, 상태, 공유, 위키 링크 등이며 해당 정보를 수정하거나 삭제할 수 있다. 10.5 Issue categories 이슈 카테고리에는 이슈에 대한 카테고리들을 지정하고 기본 Assignee를 할당할 수 있다. 10.6 Wiki Wiki 탭에서는 Wiki의 시작 페이지를 설정할 수 있다. 10.7 Repositories Repositories 탭에서는 프로젝트의 소스코드 저장소를 추가할 수 있다. 10.8 Forums Forums 탭에서는 프로젝트에서 사용할 수 있는 간단한 Forums를 생성할 수 있다. 다음 그림은 Forums 탭에 대한 예제 화면을 보여준다. image2017-2-22_20-28-29.png 10.9 Activities Activities 탭에서는 Time Tracking 기록을 위한 활동을 선택할 수 있다.

2017-03-01 00:55:48.69

Redmine 관리자 가이드

이 문서는 Redmine의 관리자에 대한 일반적인 가이드를 제공한다. 1. Projects Administration > Projects에서는 기본적으로 모든 Active 프로젝트의 리스트를 보여준다. 모든 프로젝트를 확인하기 위해서는 Filters의 Status를 All로 변경할 수 있다. 프로젝트 리스트에서 프로젝트를 선택하면 각 프로젝트의 settings로 이동하며,

프로젝트에 대한 정보를 수정할 수 있다. image2017-2-22_21-33-10.png 1.1 프로젝트 상태 레드마인에서 프로젝트 상태는 다음과 같이 분류 될 수 있다. Active : 프로젝트가 정상적으로 운영되고 있는 상태 Closed : 프로젝트가 Read Only인 상태 Archived : 사용자에게 더이상 보이지 않는 상태 1.2 프로젝트 생성 새로운 프로젝트를 만들기 위해 프로젝트 메뉴의 상단의 New project 링크를 선택한다. 혹은 프로젝트 리스트에서 복사를 희망하는 프로젝트에서 Copy 링크를 선택하면 해당 프로젝트의 정보를 복사하여 프로젝트를 생성할 수 있다. Name : 프로젝트 이름을 설정한다. Description : 프로젝트에 대한 설명을 설정한다. Identifier : 프로젝트 식별자를 설정한다. Homepage : 관련 홈페이지의 링크를 설정한다. Public : 프로젝트를 공용으로 설정한다. Subproject of : 서브프로젝트로 설정한다. Inherit members : 사용자를 상속 받는다. Modules : 프로젝트에 활성화하기를 원하는 모듈을 선택한다. Tracker : 프로젝트에서 사용하기 원하는 이슈 타입을 선택한다. Custom fields : 프로젝트에서 사용하기 원하는 커스텀 필드를 선택한다. image2017-2-22_21-35-4.png 1.3 프로젝트 삭제 프로젝트를 삭제 위해 프로젝트 리스트에서 삭제를 원하는 프로젝트의 Delete 링크를 선택해준다. 2. Users Administration > Users에서 사용자 리스트와, 상태 관리, 정보수정, 추가, 삭제 등을 수행할 수 있다. 2.1 사용자 검색 기본적으로 active되어 있는 사용자가 표시되며, 전체 사용자 리스트를 보기위해서는 Filters에서 Status를 변경하면 된다. 다음 그림은 사용자 리스트 조회 화면을 보여준다. image2017-2-22_20-58-32.png 2.2 사용자 정보 수정 사용자 리스트에서 특정 사용자 Login 아이디를 선택하면, 해당 사용자의 정보를 수정할 수 있다. Information Login : 로그인 아이디 First name : 이름 Last name : 성 Email : 이메일 Language : 사용하고자 하는 언어 선택 Administrator : 관리자 권한 여부 Email notifications For any event on all my projects 나의 모든 프로젝트의 알림을 받음 For any event on the selected projects only... 선택된 프로젝트만 알림을 받음 Only for things I watch or I'm involved in 내가 포함되어 있거나 보고 있는 이슈 Only for things I am assigned to 내가 할당된 이슈 Only for things I am the owner of 내가 소유자인 이슈 No events 받지 않음 I dont want to be notified of changes that I make myself 내가 만든 수정에 대한 알림을 받지 않음 Preferences Hide my email address : 이메일 주소 숨김 Time zone : 표시를 원하는 시간 대역 Display comments : 코멘트 출력 방식 Warn me when leaving a page with unsaved text : 저장 알림 Authentication mode 내부 사용자 관리를 사용할지? 아니면 외부 Directory를 선택 2.3 사용자 추가 새로운 사용자를 추가하기 위해서는 사용자 리스트 페이지에서 New user 링크를 선택한다. 2.4 사용자 삭제 혹은 Lock 사용자를 삭제하기 위해서 사용자 리스트이 끝 컬럼에서 Delete 버튼을 선택한다. 사용자를 Lock하기 위해서는 사용자 리스트의 끝 부분 컬럼에서 Lock 버튼을 선택한다. 사용자 Lock은 해당 사용자의 로그인 및 접근 제한, 이슈 할당 제한, Watcher 제한, 이메일 noti 제한, 그리고 프로젝트 멤버에서 보이지 않게 한다. 3. Groups Administration > Groups에서는 레드마인에서 사용될 수 있는 그룹을 생성 및 삭제하고 해당 그룹에 사용자를 할당할 수 있다. 다음 그림은 Groups 페이지에 대한 예제를 보여준다. image2017-2-22_21-0-19.png 3.1 Group 생성 새로운 그룹을 생성하기 위해 Groups 리스트의 상단의 New group 링크를 선택한다. 적절한 그룹 명을 입력하고 하단 Create 버튼을 클릭한다. 생성한 그룹명은 리스트에서 변경을 원하는 특정 그룹을 선택하고 General 탭에서 그룹명 변경 후 Save 버튼을 클릭한다. 3.2 Group에 사용자 할당 그룹에 사용자를 할당하기 위해, 리스트에서 특정 그룹을 선택하고 Users 탭에서 New user 링크를 통해 사용자를 추가한다. 다음 그림은 Group에 사용자를 추가하는 예제를 보여준다. image2017-2-22_21-0-45.png 특정 그룹을 프로젝트로 할당하기 위해 Projects 탭에서 Add Projects 링크를 클릭 후 적용을 원하는 프로젝트와 Roles을 선택한다. 3.3 Group 삭제 특정 그룹을 삭제하기위해 프로젝트 리스트에 Delete 버튼을 클릭한다. 4. Roles and Permissions Administration > Roles and Permissions에서는 프로젝트에서 사용되는 사용자의 역할과 해당 역할에 대한 권한을 설정할 수 있다. image2017-2-22_21-1-17.png 4.1 새로운 역할 추가 새로운 역할을 추가하기 위해 상단 New role 링크를 선택한다. 공통 Name : 새로운 역할 이름 Issues can be assigned to this role : 역할에 이슈 할당 Issue visibility : 볼 수 있는 이슈들 Time logs visibility : 볼 수 있는 타임로그 Users visibility : 볼 수 있는 사용자 Copy workflow from : 워크플로우 복사 대상 Permissions Project : 프로젝트 관련 권한 Forums : 포럼 관련 권한 Calendar : 달력 관련 권한 Documents : 문서 관련 권한 Files : 파일 관련 권한 Gantt : Gantt 관련 권한 Issue tracking : 이슈 트래킹 관련 권한 News : 뉴스 관련 권한 Repository : 저장소 관련 권한 Time tracking : 타임 트래킹 관련 권한 Wiki : 위키 관련 권한 image2017-2-22_21-1-38.png 4.3 Permissions Report Permissions Report를 통해 각 역할 별로 권한을 표로 보거나 부여할 수 있다. image2017-2-22_21-2-14.png 4.4 역할 삭제 각 역할은 역할 리스트에서 Delete 버튼을 통해 삭제할 수 있다. 5. Tracker Administration의 Tracker에서는 레드마인에서 사용하는 이슈 타입(Tracker)의 리스트를 확인할 수 있다. image2017-2-22_21-2-43.png 5.1 새로운 이슈타입 추가 새로운 이슈타입을 만들기 위해 상단 New tracker 링크를 선택한다. 공통 Name : 이슈타입의 이름 Default status : 이슈가 생성되었을 때 기본 상태 Issues displayed in roadmap : 로드맵에 표시 여부 Standar fields : 이슈에 사용할 표준 필드 Custom fields : 이슈에 사용할 커스텀 필드 Copy workflow from : 워크플로우를 복사할 대상 프로젝트 프로젝트에 영역에서는 해당 이슈를 적용할 프로젝트를 선택한다. 5.2 Summary 상단 Summary 링크를 선택하면, 다음 그림과 같이 이슈 타입 별로 사용하는 필드의 리스트를 보거나 선택할 수 있다. image2017-2-22_21-3-22.png 6. Issue statuses Administration의 Issue statuses에서는 Redmine 이슈에서 사용하는 이슈의 상태의 리스트를 확인할 수 있다. image2017-2-22_21-3-54.png 6.1 새로운 상태 추가 새로운 상태를 추가하기 위해서 상단 New status 링크를 선택한다. New status 화면이 나타나면 이슈 제목을 선택하고 해당 상태가 Closed를 나타내는지 여부를 체크해서 Create를 선택한다. 7. Workflow Administration의 Workflow에서는 역할 별 이슈 타입의 워크플로우를 선택할 수 있다. 7.1 Workflow 수정 워크플로우를 수정하기 위해 상단 Role에서 수정하고자하는 역할을 선택하고 Tracker에서 수정을 원하는 이슈를 선택한다. 좌측은 현재 상태이고 우측 상단의 상태들은 이동할 수 있는 상태를 나타낸다. 만약 진행에서 해결로 이동하기를 원하면 진행을 체크하면 된다. image2017-2-22_21-5-24.png 7.2 Fields Permissions Workflow 상태별로 접근할 수 있는 필드를 설정하기 위해서는 Fields permissions 탭을 선택한다. 이 역시 상태 별로 해당 필드를 Edit 모드로 할지? Read only로 할지?를 선택한다.

image2017-2-22_21-5-48.png 8. Custom fields 레드마인에서는 기본 레드마인 표준 필드 외에 추가적인 정보를 포함할 수 있는 커스텀 필드를 제공한다.

image2017-2-22_21-10-21.png 8.1 Custom fields 생성 새로운 커스텀 필드를 생성하기 위해 Administration > Custom fields > New custom field를 선택한다. New custom field 페이지가 나타나면 먼저 커스텀 필드의 범주를 선택한다. Select the type of object to which the custom field is to be attached : Issues : 이슈 Spent time : 시간 Projects : 프로젝트 Versions : 버전 Documents : 문서 Users : 사용자 Groups : 그룹 Activities : 활동 Issue priorities : 이슈 우선순 위 Document categories : 문서 카테고리 image2017-2-22_21-10-35.png 다음 정보를 참고하여 필요한 정보를 선택한다. Format : 지원되는 Format은 다음과 같다. Boolean : 체크박스 Date : 날짜 Float : floating 포인트 수 Integer : 양수 혹은 음수 Link : URL List : 드롭다운 리스트 Long Text : 멀티라인 텍스트 User : 사용자 Version : 버전 Name : 필드명 Description : 설명 Min - Max length : 최소 - 최대 길이 Regular expression : 정규표현식 적용 Text formatting : 선택 시, 위키 문법 지원 Default value : 기본가치 Link values to URL : 다음과 같은 변수를 사용하는 URL 생성 %value% : 커스텀 필드 값 %id% : 커스텀 오브젝트의 아이디 %project_id% : 커스터마이징된 오브젝트의 프로젝트 아이디 %project_identifier% : 커스터마이징된 오브젝트의 식별자 %m1%, %m2% : 커스텀 필드 정규식에 매치되는 그룹 캡처 Required : 이슈를 생성하거나 저장하기 위해 요구되는 필드 For all projects : 만약 체크된다면, 모든 프로젝트를 위해 사용된다. Searchable : 레드마인에서 검색 가능한지 여부 Visible : 사용자 프로파일 여부 image2017-2-22_21-7-40.png 9. Enumerations Enumeration은 Document categories, Issue priorities, Activities(time tracking)에서 사용되는 Enumerations 값을 설정한다. image2017-2-22_21-11-41.png 9.1 Enumerations 추가 각 Enumerations에 값을 추가하기 위해 하위에 New value 링크를 선택한다. 다음 항목을 참고하여 입력하고 Create 버튼을 클릭한다. Name : 새로운 Enumerations 값을 입력 Active : 체크 시 활성화 된다. Default value : 기본값으로 설정한다. 9.2 Enumerations 삭제 리스트에서 삭제를 원하는 아이템에서 Delete 버튼을 클릭한다. 10. Settings Settings에서는 레드마인의 전역에 적용될 수 있는 다양한 설정을 변경할 수 있게 해준다. 설정을 위해 Administration > Settings를 선택한다. 10.1 General General 탭에서는 다음에 대한 설정을 적용할 수 있다. Application title : 레드마인 헤딩에 나타나는 제목 Welcome text : 레드마인 홈페이지에 표시되는 텍스트로 HTML을 지원한다. Attachment max. size : 업로드 파일의 최대 사이즈 Objects per page options : Issues, Comments 등 페이지에 보여지는 기본 숫자 Days displayed on project activity : 프로젝트 엔티비티 내의 보여주는 일 수 Host name and path : 레드마인 서버의 호스트 이름과 경로 Protocol : 접속을 위한 프로토콜 http 혹은 https Text formatting : 텍스트 필드의 에디터를 위한 포맷 Cache formatted text : 포맷 텍스트의 캐싱 활성화 Wiki history compression : 위키 히스토리 저장소의 압축 활성화 Feed content limit : RSS 피드에 포함되는 최대 피드 수 Max size of text files displayed inline KB : 텍스트 파일의 최대 사이즈 Max number of diff lines displayed : 변경 라인 최대 수 image2017-2-22_21-13-8.png 10.2 Display Display 탭에는 다음에 대한 설정을 적용할 수 있다. Theme : 커스텀 테마를 선택할 수 있게 해준다. Default language : 기본 언어 설정 Date format : 날짜 포맷 Based on user's language : 사용자 언어에 따라 설정됨 Other formats : 사용자가 설정한 포맷 Time format : 타임 포맷 Based on user's language : 사용자 언어에 따라 설정됨 Other formats : 사용자가 설정한 포맷 Users display format : username이 보여지는 방식 Use Gravatar user icons : Gravatars 사용자 아이콘 사용 Default Gravatar image : 기본 Gravatar 이미지 image2017-2-22_21-14-16.png 10.3 Authentication Authentication 탭에는 다음에 대한 설정을 적용할 수 있다. Authentication required : 체크되어 있다면, anonymous 사용자의 페이지 접근을 차단 Autologin : 자동으로 로그인 기간 혹은 기능 사용 여부 Self-registration : 이 옵션은 회원가입을 활성화 하거나 비활성화할 수 있음 Minimum password length : 최소 패스워드 길이 Lost password : 이 옵션이 체크되어 있다면, 패스워드 찾기 기능이 활성화 된다. Allow OpenID login and registration : OpenID 로그인과 등록 가능 여부 Session expiration : Session maximum lifetime : 세션의 최대 라이프 타임 설정 Session inactivity timeout : 비활동 시 세션 타임아웃 설정 image2017-2-22_21-14-30.png 10.4 Projects Project 탭에서는 다음에 대한 설정을 적용할 수 있다. New projects are public by default : 새로운 프로젝트의 공개 여부 Default enabled modules for new projects : 새로운 프로젝트를 위한 기본적으로 활성화 되는 모듈 설정 Default trackers for new projects : 새로운 프로젝트를 위한 기본 이슈 타입 설정 Generate sequential project identifiers : 이 옵션을 사용 시, 레드마인의 식별자를 순차적으로 생성함 Role given to a non-admin user who creates a project : 관리자가 아닌 역할에 프로젝트 생성 권한을 부여함. image2017-2-22_21-14-44.png 10.5 Issue tracking Issue tracking 탭에서는 다음에 대한 설정을 적용할 수 있다. Allow cross-project issue relations : 설정 시 다른 프로젝트로부터 이슈들 사이의 관계를 생성할 수 있다. Link issues on copy : 복사한 이슈의 링크 여부 Allow cross-project subtasks : 서브 태스크에 대한 제한 Allow issue assignment to groups : 이슈를 그룹에 할당하는 것을 허락 Use current date as start date for new issues : 새로운 이슈를 생성 시, 현재 시점을 시작 날짜로 사용 Display subprojects issues on main projects by default : 설정 시, 서브 프로젝트의 이슈는 메인 프로젝트의 리스트에 보여짐 Calculate the issue done ratio with : 이슈 완료 % 설정 Non-working days : 휴일 설정 Issues export limit : 이슈 Export 제한 Maximum number of items displayed on the gantt chart : 간트차트에 출력되는 최대 아이템의 수 Default columns displayed on the issue list : 이슈 리스트에 보여지는 기본 컬럼 image2017-2-22_21-15-0.png 10.6 Email notifications Email notifications 탭에서는 이메일 알림과 관련된 다음 설정을 적용할 수 있다. Emission email address : 알림 이메일의 From에 사용되는 이메일 주소 Blind carbon copy recipients(bcc) : 설정 시, 이메일은 숨은 참조로 알림을 보낸다. Plain text mail (no HTML) : 설정 시, 이메일은 text 모드로 보내진다. Default notification option : 기본 알림 설정 Select actions for which email notifications should be sent : 이메일 알림 액션 설정 Email header : 이메일 헤더 설정 Email footer : 이메일 푸터 설정 image2017-2-22_21-15-17.png 10.7 Incoming emails Incoming emails 탭에는 다음 설정을 적용할 수 있다

다. Truncate emails after one of these lines : incoming 메일로부터 signatures 제거 사용 Exclude attachments by name : 첨부파일 제거 Enable WS for incoming emails API key : 레드마인은 이메일로부터 이슈 생성 혹은 커멘트 작성을 구성할 수 있으며, 이를 위해 받는 메일의 API를 활성화해야 한다. image2017-2-22_21-16-15.png 10.8 Repositories Repositories 탭에는 다음 설정을 적용할 수 있다. Enabled SCM : 프로젝트에 연결할 수 있는 SCM 설정 Fetch commits automatically : 저장소로부터 자동으로 새로운 버전의 리비전을 가져온다. Enable WS for repository management : 자동 SVN 저장소 생성 스크립트가 설치되어 있다면, 활성화할 수 있다. Maximum number of revisions displayed on file log : 파일 로그에 보여지는 최대 리비전 수 Referencing keywords : Commit 메시지에서 이슈와 연계를 위해 참조할 키워드 Allow issues of all the other projects to be referenced and fixed : 다른 프로젝트의 이슈와의 연계 Enable time logging : Commit 메시지에서 타임 로깅 허락 ex) Implement feature #1234 @3h15m Activity for logged time : Time logging을 위한 Activity 설정 image2017-2-22_21-16-31.png 11. LDAP authentication 레드마인은 LDAP을 통한 사용자 인증 기능을 제공하고 있다. 이를 설정하기 위해 Administration > LDAP authentication 메뉴를 선택한다. image2017-2-22_21-16-55.png 11.1 New authentication mode 새로운 LDAP 인증 모드를 생성하기 위해 상단 New authentication mode를 선택한다. Name : 디렉토리 별칭 Host : LDAP 호스트 네임 Port : LDAP 포트 Account : LDAP에 읽기 접근을 위한 사용자 이름을 Password : 계정을 위한 패스워드 Base DN : LDAP 디렉토리 트리의 최상위 레벨의 DN LDAP filter : LDAP 적용을 위한 Filter Timeout : 타임아웃 시간 On-the-fly user creation : 체크 시, 특정 LDAP 사용자가 레드마인에 로그인 시 레드마인 계정이 생성됨 Login attribute : 로그인에 사용될 아이디 매핑 Firstname attribute : 이름 매핑 Lastname attribute : 성 매핑 Email attribute : 이메일 매핑 image2017-2-22_21-17-37.png 12. Plugins Plugins 메뉴에는 현재 설치되어 있는 플러그인의 리스트와 구성을 설정할 수 있게 해준다. image2017-2-22_21-18-16.png 13. Information 설치된 레드마인의 정보를 확인할 수 있다. image2017-2-22_21-18-42.png 14. 알림 메일 구성 이메일 구성은 configuration.yml에서 설정할 수 있다. 다음은 구글 이메일을 보내는 메일로 설정한 예제를 보여주고 있다. production: email_delivery: delivery_method: :smtp smtp_settings: enable_starttls_auto: true address: "smtp.gmail.com" port: '587' domain: "smtp.gmail.com" authentication: :plain user_name: "your_email@gmail.com" password: "your_password" 다음은 지원되는 이메일 인증 (authentication) 방식을 보여준다. nil plain login cram_md5 다음은 지원되는 이메일 발송 방식(delivery_method)을 보여준다. smtp sndmail async smtp async_sendmail 15. 참고도서 다음은 참고할 수 있는 레드마인 도서를 보여준다.

2017-02-20 22:11:32.8

4. Git On The Server

2019-06-15 16:13:02.25

2. Git 협업하기

2017-08-17 14:31:39.849

Migration

2020-05-29 22:05:37.8

Git Tips

Bare 저장소를 일반 저장소로 변환하기 1. bare 저장소에 .git 폴더 생성 mkdir .git
 2. 모든 파일/폴더를 .git 폴더로 이동 mv branches/ .git/ mv config .git/ mv
 FETCH_HEAD .git/ mv HEAD .git/ mv hooks/ .git/ mv logs/ .git/ mv objects/ .
 git/ mv packed-refs .git/ mv refs/ .git/ 3. core.bare 설정 git config --local --
 bool core.bare false 4. master 브랜치 checkout git reset --hard mkdir .git mv
 branches/ .git/ mv config .git/ mv FETCH_HEAD .git/ mv HEAD .git/ mv
 hooks/ .git/ mv logs/ .git/ mv objects/ .git/ mv packed-refs .git/ mv refs/ .git/
 git config --local --bool core.bare false git reset --hard Git username,
 password 저장하기 영구 저장 방법 ~/.git-credentials 파일에 plain text로 저장됨
 (파일에 대해 access 권한 관리 필요) \$ git config --global credential.helper
 store 임시 저장 방법 캐시에 임시 저장됨 \$ git config credential.helper cache
 <timeout> Commit 개수 얻기 To get a commit count for a revision
 (HEAD, master, a commit hash): \$ git rev-list --count <revision> 예) \$ git rev-
 list --count branch-1 \$ git rev-list --count 3930540717b How to
 authenticate user via git pre-receive hook [https://stackoverflow.com
 /questions/38015205/how-to-authenticate-user-via-git-pre-receive-hook](https://stackoverflow.com/questions/38015205/how-to-authenticate-user-via-git-pre-receive-hook)
 read old new ref author=\$(git log -1 \$ref --pretty=%an) committer=\$(git
 log -1 \$ref --pretty=%cn) echo author:\$author echo committer:\$committer
 fifty_first_commits = list(repo.iter_commits('master', max_count=50))
 assert len(fifty_first_commits) == 50 # this will return commits 21-30 from
 the commit list as traversed backwards master ten_commits_past_twenty =
 list(repo.iter_commits('master', max_count=10, skip=20)) assert len
 (ten_commits_past_twenty) == 10 assert fifty_first_commits[20:30] ==
 ten_commits_past_twenty

2017-02-20 22:10:25.149

3. Git Branching

2019-06-15 14:14:09.475

1. Git 시작하기

2018-07-30 15:29:55.298

Git 교육자료

2019-06-16 12:18:51.827

Pull Request 만들기

Git 가이드

2020-04-22 08:13:04.096

브랜치 사용하기

Git 가이드

2019-06-16 12:17:05.256

동기화하기

Git 가이드

2020-04-22 09:40:37.978

Git Merge

머지는 분기된 이력을 병합하는 방법으로 하나의 브랜치 변경 이력을 다른 브랜치에 병합한다. 다음은 병합의 일반적인 절차를 설명한다. Step 1) 병합될 브랜치로 전환
 본 예는 Feature → mastre 병합을 고려한다. git checkout master Step 2) origin 동기화 git pull origin Step 3) 머지 머지 명령을 이용해 Feature 브랜치를 master 브랜치로 병합한다. --no-ff 옵션은 머지 커밋을 생성하고 싶은 경우 지정한다. git merge [--no-ff] Feature Screen Shot 2020-04-22 at 9.08.22 AM.png Setp 4) Feature branch 삭제 git branch -d Feature Fast-Forward Merge 아래 그림과 같이 머지 대상 (master)에 변경 이력이 없는 상태에서 Feature의 변경 이력을 병합하는 경우를 의미한다. Screen Shot 2020-04-22 at 9.23.43 AM.png 3-way Merge 머지 대상 (본 예의 경우 master)에 변경 이력이 존재하는 상태에서 Feature의 변경 이력을 병합하는 경우를 의미한다. Screen Shot 2020-04-22 at 9.22.55 AM.png Merge Conflict 3-way merge의 경우 동일한 파일에 같은 지점이 변경된 경우 병합 대상 (본 예의 경우 Feature 브랜치)와 master를 병합할 때 충돌이 발생할 수 있다. 이 때 git은 고유한 방법으로 해당 파일 위치에 출동을 표시한다. \$ cat merge.txt <<<<<< HEAD this is some content to mess with (현재 브랜치 변경 내용) content to append ===== totally different content to merge later >>>>>> new_branch_to_merge_later 일반적인 해소 절차는 관련 페이지를 참조한다. Git 가이드

2020-04-22 09:49:47.836

병합 전략 (Merge Strategies)

Git 가이드

2020-04-22 09:49:35.562

병합 충돌 해결하기 (Merge Conflicts)

충돌 해소 절차는 다음과 같다. Step 1) 파일 확인 \$ git status On branch master
 You have unmerged paths. (fix conflicts and run "git commit") (use "git merge --abort" to abort the merge) Unmerged paths: (use "git add <file>..." to mark resolution) both modified: merge.txt (충돌 파일) Step 2) 파일 수정 편집기를 통해 파일을 수정한다. \$ cat merge.txt <<<<<< HEAD this is some content to mess with content to append ===== totally different content to merge later >>>>>> new_branch_to_merge_later Step 3) Add to index & commit git add <conflicted files> git commit -m <commit message> Step 4) Push to origin git push origin Git 가이드

2020-04-22 08:33:06.023

Git Branch

Git 은 강력한 브랜치 기능을 제공하므로 브랜치 기능을 이해하는 것이 중요하다. 브랜치 생성 브랜치는 원격 또는 로컬 저장소에서 생성할 수 있고 저장소간 동기 가능하다. git 은 현재 브랜치의 HEAD를 기반으로 브랜치가 생성된다. git branch <branch> 브랜치 목록 보기 로컬 브랜치 목록 로컬 브랜치 목록을 표시한다. git branch 전체 브랜치 목록 로컬 및 원격 브랜치 목록을 표시한다. git branch -a 브랜치 삭제 git은 안전한 삭제와 강제 삭제 두 가지 삭제 옵션을 제공한다. 안전 옵션 머지 되지 않은 변경사항이 있는 경우 삭제 되지 않는다. git branch -d <branch> 강제 옵션 이 옵션은 머지 상태와 관계 없이 브랜치 삭제가 가능하다. git branch -D <branch> 브랜치 이름 변경 git은 브랜치 이름 변경을 지원한다. git branch -m <branch> Screen Shot 2020-04-22 at 8.13.33 AM.png Git 가이드

2020-04-22 09:07:22.629

Git Checkout

git checkout은 파일, 커밋 그리고 브랜치를 전환하는 방법을 제공한다. 새로운 브랜치로 전환 신규 브랜치를 생성하여 해당 브랜치로 전환하는 방법을 제공한다. existing-branch가 주어진 경우 주어진 브랜치 기반으로 브랜치를 생성하며 생략된 경우 현재 브랜치 기반으로 생성한다. git checkout -b <new-branch> [existing-branch] 존재하는 브랜치 전환 기존에 존재하는 브랜치로 전환하는 방법을 제공한다. git checkout <branch> 파일 전환 브랜치의 특정 파일 또는 폴더 전환도 지원한다. git checkout master -- <path 또는 파일 경로> Detached HEAD Git은 HEAD 을 이용해 현재 snapshot를 관리한다. 내부적으로 checkout은 HEAD이 가리키는 commit 또는 branch 을 변경한다. HEAD가 아닌 commit을 checkout하면 git은 detached HEAD 상태로 전환한다. detached HEAD 상태는 참조하는 브랜치가 없기 때문에 과거 커밋 내용을 확인하는 등의 용도에서 사용고 정상적인 기능 개발을 위해 사용하지 않도록 주의한다. Screen Shot 2020-04-22 at 8.50.30 AM.png Git 가이드

2020-04-22 07:12:26.777

Git Fetch

git fetch는 원격 저장소 변경사항을 로컬 저장소에 동기화 한다. -all 옵션은 모든 원격 저장소 정보 동기화 한다. git fetch --all <remote>를 지정하면 특정 remote의 변경 내용을 동기화 하며, branch가 지정되면 특정 브랜치 정보만 동기화 한다. git fetch <remote> [branch] --dry-run 옵션은 로컬 저장소에 반영하지 않는다. git fetch <remote> [branch] Screen Shot 2020-04-22 at 6.50.43 AM.png Git 가이드

2020-04-22 08:00:41.018

Git Push

일반 기능 로컬 저장소의 커밋 내용은 push 동작을 통해 원격 저장소에 전송한다. branch가 지정되지 않으면 현재 로컬 저장소와 동일한 원격 브랜치에 정보를 업로드 (동기화) 한다. git push <remote> [branch] Screen Shot 2020-04-22 at 7.27.55 AM.png 강제 업로드 Git은 기본적으로 fast-forward 만 허용한다. --force 옵션을 사용해 non-fast-forward 로 push 할 수 있다. 이력에 대해 정확한 이해가 있을 때만 --force 옵션을 사용한다. git push <remote> --force 모든 로컬 브랜치 업로드 git push <remote> --all 태그 정보 업로드 git push 기본 동작은 태그 정보를 업로드 하지 않는다. git push <remote> --tags fast-forward vs. non-fast-forward git push는 원격 저장소의 보호를 위해 기본적 동작은 fast-forward 모드를 권장한다. fast-forward: 로컬 저장소가 원격 저장소와 동기가 이루어진 상태에서 업로드하는 모드 non-fast-forward: 로컬 저장소가 원격 저장소에 뒤쳐진 상태에서 업로드하는 모드 fast-forward.png Git 가이드

2020-04-22 07:26:30.879

Git Pull

git pull은 원격 저장소 정보를 로컬 working folder에 반영한다. 상태1) 로컬 저장소의 master 브랜치가 원격 저장소에 뒤쳐진 상태 Screen Shot 2020-04-22 at 7.16.31 AM.png pull 후) git pull origin master 원격 저장소의 A-B-C 커밋이 단 일 머지 커밋 (H)로 저장된다. Screen Shot 2020-04-22 at 7.19.22 AM.png rebase 옵션을 이용한 pull) rebase 옵션이 주어진 경우 원격 저장소의 A-B-C 커밋이 로컬 저장소의 브랜치에 rebase 된다. git pull --rebase origin Screen Shot 2020-04-22 at 7.22.40 AM.png Git 가이드

2020-04-22 07:00:40.959

Git Remote

origin remote는 로컬 저장소의 원본 저장소를 지칭하며 origin은 clone 절차에서 자동으로 설정된다. 리모트 저장소 목록 보기 \$ git remote [-v] origin upstream other_users_repo 원격 저장소 추가 아래 명령으로 원격 저장소를 추가한다. \$ git remote add fake_test https://bitbucket.com/upstream_user/reponame.git; [remote "remote_test"] url = https://bitbucket.com/upstream_user/reponame.git fetch = +refs/heads/*:refs/remotes/remote_test/* 원격 저장소 점검 git remote show upstream * remote upstream Fetch URL: https://bitbucket.com/upstream_user/reponame.git Push URL: https://bitbucket.com/upstream_user/reponame.git HEAD branch: master Remote branches: master tracked simd-deprecated tracked tutorial tracked Local ref configured for 'git push': master pushes to master (fast-forwardable) 원격 저장소와 로컬 저장소 동기 방법 Git은 원격 저장소와 로컬 저장소 동기 방법을 제공한다. 세부 사용법은 해당 설명 페이지를 참고한다. git fetch git pull git push Screen Shot 2020-04-22 at 6.50.43 AM.png Git 가이드

2017-08-17 14:51:49.397

Migrating A CVS Repository To GIT

Importing from a CVS repository can be done using git-cvsmimport either by direct access to the remote CVS server or from a snapshot tarball. The procedure includes the following steps: Import a CVS repository to a git repository (usually on a local machine) Create a bare clone of the latter to be used as a shared read/write repository (again locally) Set a new project repository on the server through gitosis (see above "Starting a new project repository) Push the cloned one in 2. to the newly created one in 3. All steps above should be done on a local machine. 1. CVS repository import cvsmimport 설치 필요 user@local# sudo yum install git-cvsmimport Direct import from a server user@local# cvs -z3 -d :pserver:anoncvs@curvc.com:/home/gmx/cvs login user@local# git cvsmimport -A ./authors.txt -v -a -i -k -d :pserver:anoncvs@curvc.com:/home/gmx/cvs -C gromacs-cvs.git cmx This way is very slow, and it is much faster to use a tarball of a CVS snapshot. authors.txt: CVS 사용자를 Git 사용자로 매핑 형식 예) username1_in_cvs = Git User1 <gituser1@curvc.com> username2_in_cvs = Git User2 <gituser2@curvc.com http://curvc.com> (Alternative) Import from a CVS snapshot tarball rossen@curvc.com# cd /home/gmx rossen@curvc.com# tar cvfz /home/rossen/gromacs-cvs-20090218.tar.gz cvs rossen@local# pwd /home/rossen/temp rossen@local# scp cvs.curvc.com:gromacs-cvs-20090218.tar.gz . rossen@local# tar xvfz gromacs-cvs-20090218.tar.gz rossen@local# git-cvsmimport -a -i -k -d /home/rossen/temp /cvs -C gromacs-cvs.git gmx 2. Create a bare clone of the imported GIT rossen@local# mkdir gromacs rossen@local# cd gromacs rossen@local# git --bare init --shared rossen@local# git --bare fetch ../gromacs-cvs.git master: master rossen@local# git remote add gromacs git@git.curvc.com:gromacs.git 3. Set a new repository on the server Git 서버에서 gromacs repository 생성 4. Push the clone to the server rossen@local# git push gromacs --mirror Links git-cvsmigration <http://www.kernel.org/pub/software/scm/git/docs/gitcvs-migration.html> git-cvsserver <http://www.kernel.org/pub/software/scm/git/docs/git-cvsserver.html> git-cvsmimport <http://www.kernel.org/pub/software/scm/git/docs/git-cvsmimport.html>

2021-07-07 17:09:11.724

Git Push Store In Secure Store가 동작하지 않을 때

현상 Git push 동작 시 계정정보를 계속 해서 물어보는 현상 (store in secure store 를 체크했을때도) 원인 어떠한 원인으로 인하여 git 저장소의 key가 꼬여버림 해결방 안 Windows > Preferences > General > Security > Secure Storage > Contents 에서 git 저장소 선택 후 Delete image2021-7-7_17-5-20.png Eclipse 재시작 다시 commit&push하면서 계정정보 입력 및 store in secure store 체크하여 push하면 정상적으로 저장됨

2023-08-03 13:17:54.674

Git 개행처리 설정 방법

이 문서는 Git 사용중 OS 간의 개행 처리 방법이 다른 문제를 해결하는 개행 처리 설정 방법에 대한 가이드를 공유하기 위해 작성되었다. 개요 Git config 적용 범위는 시스템 설정, 글로벌 설정, git 저장소별 설정이 가능하고 우선순위는 git 저장소 → 글로벌 → 시스템 순이다. (git 저장소가 가장 높음) 특별하게 저장소에 별도로 설정할 일이 있는 경우를 제외하고 일반적으로 사용자 레벨인 글로벌에 config를 설정한다. Config 파일 위치 : git저장소 : 저장소*.git\config 글로벌 : C:\Users\사용자계정*.git\config 시스템 : C:\ProgramData\Git\config autocrlf 설정 사용자 각자가 git bash에서 아래 config 설정 입력. gitattributes의 경우 eclipse의 egit에서 해당 파일을 읽을 수 없어서 미사용. Windows 환경 윈도우에서는 CRLF를 사용하므로 저장소에서 가져올 때 LF를 CRLF로 변경하고 저장소로 보낼 때는 CRLF를 LF로 변경하도록 true로 설정한다. git config --global core.autocrlf true Linux, MAC, Unix 환경 리눅스, 맥, 유닉스는 LF만 사용하므로 input으로 설정한다. git config --global core.autocrlf input 파일모드 false 설정 git이 리눅스의 파일모드를 감시하기 때문에 파일 권한으로 인하여 파일이 git status에 변경된 이력으로 노출될 수 있음. 다음 설정을 통해서 filemode를 false 설정한다. git config --global core.filemode false git config 설정 적용 확인 git config --list Eclipse에서 Git config 적용 일반적으로 Eclipse의 git config는 git의 config를 바라보고 있으나 Eclipse에서도 git config 설정이 가능하다. 경로: 메뉴 > Window > Preference > Team > Git > Configuration User Settings 탭에서 Add Entry 등록 key : autocrlf / Value : true Key : filemode / Value : false crlf 변경하여 다시 적용하는 방법 git repository를 새로 Clone 받았을때 아무런 변경도 하지 않았는데 Changes에 파일들이 변경이 되었다고 나오면 개행이 달라서 변경이 된 것으로 노출되는 것이다. git에서 관리하는 파일의 개행이 깨졌을 경우 복구하는 방법은 다음과 같다. 본인의 작업 환경 crlf 설정 확인(Window의 경우 crlfauto = true) eclipse에서 clone import (git bash나 다른 클라이언트에서 clone 진행시 클라이언트의 기능에 의해서 자동으로 개행처리해줘서 문제가 발생하지 않을 수 있음) workspace에서 git bash로 접속 git status 실행 시 개행으로 인하여 change로 변경된 파일들 리스트가 출력 git add. (.project파일을 제거) git commit -m "crlf change" git push commit change에서 + -가 개수 동일인지 확인 CRLF / LF 보기가상 차이점 git diff를 입력하면 파일의 개행 변경 내용을 확인 할 수 있다. [CRLF -> LF로 변경된 예시] - This is line 1^M -This is line 2^M -This is line 3^M +This is line 1 +This is line 2 +This is line 3 [LF -> CRLF로 변경된 예시] -This is line 1 -This is line 2 -This is line 3 +This is line 1^M +This is line 2^M +This is line 3^M

2022-04-25 21:33:52.426

Git Diff 두 점(..)과 세점(...) 옵션 차이

이 페이지는 Git의 diff 명령의 옵션 중 두 점 (..) 과 세 점 (...)의 차이점에 대해 정리한다. A---B---C topic / D---E---F---G master 커밋 A: a.txt 파일 수정 커밋 B: b.txt 파일 추가 커밋 C: c.txt 파일 삭제 커밋 F: f.txt 파일 변경 커밋 G: g.txt 파일 추가 위와 같이 커밋 이력이 발생한 경우, 두 점 (..) 또는 점 없는 옵션 master → topic git diff master topic --name-status 또는 git diff master..topic --name-status 은 F, G 커밋을 포함한 master 브랜치와 A, B, C 커밋을 포함한 topic 브랜치 사이의 차이점을 비교한다. 즉, 결과는 master (커밋 G)를 기준으로 topic 관점에서 차이점을 나타낸다. M a.txt A b.txt D c.txt M f.txt D g.txt topic → master git diff topic master --name-status 또는 git diff topic..master --name-status 은 F, G 커밋을 포함한 master 브랜치와 A, B, C 커밋을 포함한 topic 브랜치 사이의 차이점을 비교한다. 즉, 결과는 topic (커밋 C)를 기준으로 master 브랜치 관점에서 차이점을 나타낸다. M a.txt D b.txt A c.txt M f.txt A g.txt 세 점 (...) 옵션 master → topic git diff master...topic --name-status 은 master 브랜치와 topic 브랜치의 가장 최근 공통 커밋을 기준으로 topic 브랜치 관점에서 차이점을 비교한다. 즉, 결과는 커밋 E를 기준으로 topic 브랜치 (커밋 C) 관점에서 차이점을 나타낸다. M a.txt A b.txt D c.txt topic → master git diff topic...master --name-status 은 topic 브랜치와 master 브랜치의 가장 최근 공통 커밋을 기준으로 master 브랜치 관점에서 차이점을 비교한다. 즉, 결과는 커밋 E를 기준으로 master 브랜치 (커밋 G) 관점에서 차이점을 나타낸다. M f.txt A g.txt

2021-07-05 12:44:12.31

특정 파일 이력 원복하기 (Eclipse)

파일 단위 배포에서 변경목록 작업 중 Commit 또는 Push 된 파일들을 원복 시키는 방법을 가이드 합니다. 원복 방법 방법1) 수동으로 소스 수정하여 다시 commit&push하면 변경목록에서 파일 제거 가능 SR 브랜치가 생성된 시점과 완전히 동일해야 함 Whitespace, EOL도 일치해야함 방법2) Eclipse 가 제공하는 기능 활용하여 파일 내용 원복 원복 대상 파일 선택 Replace 메뉴선택 원복 대상 파일에 대해 마우스 우클릭 > Replace With 메뉴 선택 원복 대상 종류 선택 Replace With 메뉴 선택 > Commit... 선택 작업 중인 파일 삭제 확인 창 Replace With 기능은 지정된 이력으로 로컬 파일을 원복하므로 선택한 파일의 백업이 필요하다면 취소하여 백업 후 절차를 다시 시작한다. 진행 확인 파일이 복원되어도 좋다면 "Discard Changes" 클릭 복원할 커밋 선택 변경 목록의 파일 제거 목적이므로 SR 브랜치가 생성된 부모 커밋 (또는 SR의 최초 커밋의 이전 커밋) 선택 확인 OK 버튼을 클릭하여 파일 복원 완료 복원한 파일을 Git Add, Commit, Push 하여 삭제 중인 변경목록 내의 파일 제거 image2021-7-5_12-44-5.png

2021-07-05 12:46:51.156

특정 파일 이력 원복하기 (IntelliJ)

파일 단위 배포에서 변경목록 작업 중 Push 된 파일들을 원복 시키는 방법을 가이드 합니다. 원복 방법 방법1) 수동으로 소스 수정하여 다시 commit&push하면 변경목록에서 파일 제거 가능 방법2)File history에서 get으로 원복 후 commit&push하면 변경목록에서 파일 제거 가능 image2021-7-1_17-12-29.png image2021-7-1_17-13-10.png

2021-07-07 12:38:48.093

Git 기본 환경 설정

이 페이지는 개발 환경 중 Git의 기본 환경 설정하는 방법을 가이드한다. Git 환경 설정 파일 위치 Git 환경 파일은 다음과 같이 구별된다. --system: 시스템 default (\${prefix}/etc/gitconfig), 모든 사용자에게 적용 --global: 사용자 default (~/.gitconfig), 사용자에게 적용 --local: 현재 repository default(.git/config), 현재 작업 중인 repository에 적용 사용자 정보 설정 커밋을 기록하는 사용자 정보를 설정한다. Git 환경 설정 파일에 아래와 같은 형식으로 설정한다. [user] name = username email = username@khmnc.or.kr 라인 개행 문자 처리 설정 checkout 과 add 시에 라인 개행 문자 (CR 또는 CRLF) 처리 방법을 설정한다. 설정하지 않으면 시스템에서 사용하는 개행 문자가 적용되어 개발자 환경에 따라 파일을 구성하는 라인의 개행 문자가 섞일 수 있다 (Linux: LF, Windows: CRLF). LF 로 저장소의 개행 문자를 통일하는 것을 권장한다. 저장소의 개행 문자를 LF로 유지할 때의 aotocrlf 설정: input: 입력 대로 처리 (Linux, Mac, Cygwin 환경) true: 자동으로 LF로 변환 (Windows) [core] autocrlf = true

2020-11-26 07:49:54.47

(j) 이력에서 파일 삭제하기

original site = <https://rtyley.github.io/bfg-repo-cleaner/> <https://rtyley.github.io/bfg-repo-cleaner/> an alternative to git-filter-branch The BFG is a simpler, faster alternative to git-filter-branch <http://git-scm.com/docs/git-filter-branch> for cleansing bad data out of your Git repository history:

Removing Crazy Big Files Removing Passwords, Credentials & other Private data The git-filter-branch command is enormously powerful and can do things that the BFG can't - but the BFG is much better for the tasks above, because: Faster <https://rtyley.github.io/bfg-repo-cleaner/#speed> : 10 - 720x faster Simpler <https://rtyley.github.io/bfg-repo-cleaner/#examples> : The BFG isn't particularly clever, but is focused on making the above tasks easy Beautiful : If you need to, you can use the beautiful Scala language to customise the BFG. Which has got to be better than Bash scripting at least some of the time. Usage First clone a fresh copy of your repo, using the --mirror <http://stackoverflow.com/q/3959924/438886> flag: \$ git clone --mirror <http://example.com/some-big-repo.git> This is a bare http://git-scm.com/docs/gitglossary.html#def_bare_repository repo, which means your normal files won't be visible, but it is a full copy of the Git database of your repository, and at this point you should make a backup of it to ensure you don't lose anything. Now you can run the BFG to clean your repository up: \$ java -jar bfg.jar <https://repo1.maven.org/maven2/com/madgag/bfg/1.13.0/bfg-1.13.0.jar> --strip-blobs-bigger-than 100M [some-big-repo.git](http://example.com/some-big-repo.git) The BFG will update your commits and all branches and tags so they are clean, but it doesn't physically delete the unwanted stuff. Examine the repo to make sure your history has been updated, and then use the standard git gc <http://git-scm.com/docs/git-gc> command to strip out the unwanted dirty data, which Git will now recognise as surplus to requirements: \$ cd [some-big-repo.git](http://example.com/some-big-repo.git) \$ git reflog expire --expire=now --all && git gc --prune=now --aggressive Finally, once you're happy with the updated state of your repo, push it back up (note that because your clone command used the --mirror flag, this push will update all refs on your remote server): \$ git push At this point, you're ready for everyone to ditch their old copies of the repo and do fresh clones of the nice, new pristine data. It's best to delete all old clones, as they'll have dirty history that you don't want to risk pushing back into your newly cleaned repo. Examples In all these examples bfg is an alias for java -jar bfg.jar. Delete all files named 'id_rsa' or 'id_dsa' : \$ bfg --delete-files id_{dsa,rsa} [my-repo.git](http://example.com/my-repo.git) Remove all blobs bigger than 50 megabytes : \$ bfg --strip-blobs-bigger-than 50M [my-repo.git](http://example.com/my-repo.git) Replace all passwords listed in a file (prefix lines 'regex:' or 'glob:' if required) with ***REMOVED*** wherever they occur in your repository : \$ bfg --replace-text passwords.txt [my-repo.git](http://example.com/my-repo.git) Remove all folders or files named '.git' - a reserved filename <https://github.com/git/git/blob/d29e9c89d/fsck.c#L228-L229> in Git. These often become a problem <http://stackoverflow.com/q/16821649/438886> when migrating to Git from other source-control systems like Mercurial : \$ bfg --delete-folders .git --delete-files .git --no-blob-protection [my-repo.git](http://example.com/my-repo.git) For further command-line options, you can run the BFG without any arguments, which will output text like this <https://repository.sonatype.org/service/local/artifact/maven/redirect?r=central-proxy&g=com.madgag&a=bfg&v=LATEST&e=txt>. Your current files are sacred... By default the BFG doesn't modify the contents of your latest commit on your master (or 'HEAD') branch, even though it will clean all the commits before it. That's because your latest commit is likely to be the one that you deploy to production, and a simple deletion of a private credential or a big file is quite likely to result in broken code that no longer has the hard-coded data it expects - you need to fix that, the BFG can't do it for you. Once you've committed your changes- and your latest commit is clean with none of the undesired data in it - you can run the BFG to perform it's simple deletion operations over all your historical commits. Note: Cleaning Git repos is about completely eradicating bad stuff from history. If something 'bad' (like a 10MB file, when you're specifying --strip-blobs-bigger-than 5M) is in a protected file, it won't be deleted - it'll persist in your repository, even if the BFG deletes it from earlier commits <https://github.com/rtyley/bfg-repo-cleaner/issues/53#issuecomment-50088997>. If you want the BFG to delete something you need to make sure your current commits are clean. Note that although the files in those protected commits won't be changed, when those commits follow on from earlier dirty commits, their commit ids will change, to reflect the changed history - only the SHA-1 id of the filesystem-tree will remain the same. If you want to turn off the protection (in general, not recommended) you can use the --no-blob-protection flag: \$ bfg --strip-biggest-blobs 100 --no-blob-protection [repo.git](http://example.com/repo.git)

2019-06-15 16:27:10.384

변경 저장하기

'체크인'은 중앙 서버로 원격 푸시하는 작업입니다. 이것은 SVN 커밋이 프로젝트 변경 사항을 완전히 '저장'하기 위해 인터넷 액세스가 필요하다는 것을 의미합니다. Git 커밋은 로컬에서 캡처 및 빌드 한 다음 필요에 따라 git push -u origin master 명령을 사용하여 원격 서버에 푸시 할 수 있습니다. 두 가지 방법의 차이점은 아키텍처 설계 간의 근본적인 차이입니다. Git은 분산 애플리케이션 모델이지만 SVN은 중앙 집중식 모델입니다. 분산 응용 프로그램은 중앙 집중식 서버와 같이 단일 지점에서 장애가 발생하지 않으므로 일반적으로 더 강력합니다. git add, git status, git commit 명령은 모두 Git 프로젝트의 현재 상태의 스냅 샷을 저장하는 데 사용됩니다. Git은 'stash'라는 추가 저장 메커니즘을 가지고있다. stash는 커밋 할 준비가되지 않은 변경 사항을 임시로 저장하는 영역입니다. stash는 세 개의 트리 중 첫 번째 트리 인 작업 디렉토리에서 작동하며 광범위한 사용 옵션이 있습니다. 자세한 내용은 git stash 페이지를 참조하십시오. Git 저장소는 특정 파일이나 디렉토리를 무시하도록 설정할 수 있습니다. 이것은 자식이 무시한 내용의 변경 사항을 저장하지 못하도록합니다. 힘 내는 무시 목록을 관리하는 여러 가지 설정 방법을 가지고있다. Git ignore configure에 대해서는 git ignore 페이지에서 자세히 설명합니다. Git 가이드

2019-06-15 23:11:01.843

저장소 점검하기

Git 가이드

2019-06-16 06:59:47.491

변경 취소하기

Git 가이드

2019-06-16 13:01:34.804

Rewriting History

Git 가이드

2020-04-22 05:47:33.397

Git 저장소

Git은 새로운 저장소 생성과 기존 저장소 복제를 제공합니다. image2020-4-22_5-47-3.png 저장소란? Git 저장소는 프로젝트의 가상 저장소입니다. 필요한 경우 액세스 할 수있는 코드 버전을 저장할 수 있습니다. 새로운 저장소 생성하기: git init 새 저장소를 만들려면 git init 명령을 사용합니다. git init은 새로운 repo의 초기 설정 중에 사용하는 일회성 명령입니다. 이 명령을 실행하면 현재 작업 디렉토리에 새 .git 하위 디렉토리가 만들어집니다. 그러면 새로운 마스터 브랜치가 작성됩니다. Versioning an existing project with a new git repository 이 예에서는 기존에 프로젝트 폴더를 만들었으며 프로젝트 내에 repo를 만들려한다고 가정합니다. 먼저 루트 프로젝트 폴더로 이동 한 다음 git init 명령을 실행합니다. cd /path/to/your /existing/code git init git init을 기존 프로젝트 디렉토리로 지정하면 위에서 언급

한 것과 동일한 초기화 설정이 실행되지만 해당 프로젝트 디렉토리로 범위가 지정됩니다. git init <project directory> 기존 저장소 복제하기: git clone 프로젝트가 중앙 저장소에 이미 설정된 경우 clone 명령은 사용자가 로컬 개발 복제본을 얻는 가장 일반적인 방법입니다. git init과 마찬가지로, 복제는 일반적으로 일회성 작업입니다. 개발자가 작업 복사본을 얻으면 모든 버전 제어 작업은 로컬 저장소를 통해 관리됩니다. git clone <repo url> git clone은 원격 리포지토리의 복사본이나 복제본을 만드는 데 사용됩니다. git clone에 저장소 URL을 전달합니다. Git은 몇 가지 다른 네트워크 프로토콜과 해당 URL 형식을 지원합니다. HTTP 프로토콜을 지원하는 Bitbucket 저장소 URL 형식 예: https://user@bitbucket.almdemo.curvc.com/scm/git/spring3example.git https://user@bitbucket.almdemo.curvc.com/scm/git/spring3example.git user: Bitbucket username bitbucket.almdemo.curvc.com/scm http://bitbucket.almdemo.curvc.com/scm: Bitbucket URL git: Bitbucket project key spring3example: 저장소 이름 실행되면 마스터 브랜치의 원격 repo 파일의 최신 버전이 풀다운되어 새 폴더에 추가됩니다. 새 폴더의 이름은이 경우 javascript-data-store의 REPO_NAME 다음에 지정됩니다. 이 폴더는 원격 저장소와 새로 생성된 master 브랜치의 전체 히스토리를 포함합니다. 저장소에 변경사항 저장하기: git add and git commit 저장소가 설정되면 이제 저장소를 복제하거나 초기화 했으므로 파일 버전 변경을 커밋 할 수 있습니다. 다음 예제에서는 / path / to / project에 프로젝트를 설정했다고 가정합니다. 이 예제에서 수행되는 단계는 다음과 같습니다. 디렉토리를 / path / to / project로 변경합니다. 내용이 담긴 CommitTest.txt 파일을 만듭니다 ~ "git tutorial for test content" ~ git은 CommitTest.txt를 저장소 스테이징 영역에 추가합니다. 커밋에서 수행된 작업을 설명하는 메시지로 "CommitTest.txt 파일 추가" 지정 cd /path/to/project echo "test content for git tutorial" >> CommitTest.txt git add CommitTest.txt git commit -m "CommitTest.txt 파일 추가" 이 예를 실행하면 Repo에 기록에 CommitTest.txt가 추가되고 파일의 향후 업데이트를 추적합니다. 이 예제는 두 개의 추가적인 git 명령인 add와 commit을 소개했다. 이것은 매우 제한된 예었지만, 두 명령 모두 git add 및 git commit 페이지에서 자세히 다루어집니다. git add의 또 다른 일반적인 사용 예는 --all 옵션입니다. git add --all을 실행하면 repo에서 변경되거나 추적되지 않은 파일을 가져 와서 repo에 추가하고 repo의 작업 트리를 업데이트합니다. 저장소 간 협업: git push Git의 "작업 카피"에 대한 아이디어가 SVN 저장소에서 소스 코드를 체크 아웃하여 얻은 작업 카피와 매우 다르다는 것을 이해하는 것이 중요합니다. SVN과 달리 Git은 작업 사본과 중앙 저장소를 구분하지 않습니다. 그들은 모두 본질적인 Git 저장소입니다. 이것은 Git과 SVN과 근본적으로 다른 협력을 만듭니다. SVN이 중앙 저장소와 작업 사본 사이의 관계에 의존하는 반면, Git의 공동 작업 모델은 저장소와 저장소 간의 상호 작용을 기반으로 합니다. 작업 복사본을 SVN의 중앙 저장소로 검사하는 대신 한 저장소에서 다른 저장소로 커밋을 푸시하거나 끌어 당깁니다. 물론, Git repos라는 특별한 의미를 주는 것을 막을 수 있는 방법은 없습니다. 예를 들어 Git repo를 "중앙" 저장소로 지정하면 Git을 사용하여 중앙 집중식 워크 플로를 복제 할 수 있습니다. 이는 VCS 자체에 내장되어 있지 않고 규칙을 통해 수행됩니다. Bare vs. cloned repositories 이전의 "새 저장소 초기화"절에서 git clone을 사용하여 로컬 저장소를 설정 한 경우 저장소는 원격 협업을 위해 이미 구성되어 있습니다. git clone은 복제 한 Git URL을 가리키는 리모컨으로 자동으로 저장소를 구성합니다. 즉, 파일을 변경하고 커밋하면 원격 저장소에 변경 사항을 적용 할 수 있습니다. git init을 사용하여 새로운 repo를 작성했다면 변경 사항을 푸시할 원격 저장소가 없습니다. 새로운 repo를 초기화 할 때 일반적인 패턴은 Bitbucket과 같은 호스팅 된 Git 서비스로 가서 repo를 생성하는 것입니다. 이 서비스는 Git URL을 제공하여 로컬 Git 저장소에 추가하고 호스트 된 저장소로 Push 할 수 있습니다. 선택한 서비스로 원격 저장소를 만든 후에는 매핑을 사용하여 로컬 저장소를 업데이트해야 합니다. 이 프로세스는 아래의 구성 및 설정 안내서에서 설명합니다. 자신의 원격 저장소를 호스팅하려면 "기본 저장소"를 설정해야 합니다. git init과 git clone 모두 --bare 인수를 허용합니다. 베어 레포의 가장 일반적인 사용 사례는 원격 중앙 저장소를 만드는 것입니다. 설정: git config 원격 저장소 설정이 끝나면 로컬 저장소에 원격 저장소 URL을 추가하고 로컬 분기에 대한 업스트림 브랜치를 설정해야 합니다. git remote 명령은 그러한 유틸리티를 제공합니다. git remote add <remote_name> <remote_repo_url> 이 명령은 <remote_repo_url>의 원격 저장소를 <remote_name>의 로컬 저장소에있는 참조로 매핑합니다. 원격 저장소를 매핑하면 로컬 저장소를 해당 저장소로 푸시 할 수 있습니다. git push -u <remote_name> <local_branch_name> 이 명령은 <local_branch_name> 아래의 로컬 repo 분기를 <remote_name>의 원격 저장소로 푸시합니다. git remote에 대한 자세한 내용은 Git 원격 페이지를 참조하십시오. 원격 repo URL을 설정하는 것 외에도 username 또는 email과 같은 글로벌 Git 구성 옵션을 설정해야 할 수도 있습니다. git config 명령을 사용하여 명령 행에서 Git 설치 (또는 개별 저장소)를 구성 할 수 있습니다. 이 명령은 사용자 정보에서 환경 설정, 저장소 작동에 이르기까지 모든 것을 정의 할 수 있습니다. 몇 가지 일반적인 구성 옵션은 다음과 같습니다. Git은 개별 저장소 (로컬), 사용자 (전역) 또는 전체 시스템 (시스템)에 범위 옵션을 제공하는 구성 옵션을 3 개의 개별 파일에 저장합니다. 로컬 : <repo> / .git / config - 저장소 별 설정. 전역 : /.gitconfig - 사용자 별 설정. 여기에는 --global 플래그로 설정된 옵션이 저장됩니다. 시스템 : \$ (접두사) / etc / gitconfig - 시스템 전체 설정. 현재 저장소의 모든 커밋에 사용할 작성자 이름을 정의하십시오. 일반적으로 --global 플래그를 사용하여 현재 사용자에게 대한 구성 옵션을 설정하려고 합니다. git config --global user.name <name> 현재 사용자가 모든 커밋에 사용할 작성자 이름을 정의하십시오. --local 옵션을 추가하거나 설정 수준 옵션을 전달하지 않으면 현재 로컬 저장소에 대한 user.name http://user.name이 설정됩니다. git config --local user.email <email> 현재 사용자가 모든 커밋에 사용할 작성자 전자 메일을 정의합니다. git config --global alias.<alias-name> <git-command> Git 명령에 대한 바로 가기를 만듭니다. 이것은 일반적으로 사용되는 git 명령에 대한 사용자 정의 바로 가기를 작성하는 강력한 유틸리티입니다. 단순한 예는 다음과 같습니다. git config --global alias.ci commit 이렇게하면 git commit 바로 가기로 실행할 수 있는 ci 명령이 만들어집니다. git 별칭에 대한 자세한 내용은 git config 페이지를 참조하십시오. git config --system core.editor <editor> 현재 컴퓨터의 모든 사용자에게 대해 git commit과 같은 명령이 사용하는 텍스트 편집기를 정의하십시오. <editor> 인수는 원하는 편집기 (예 : vi)를 시작하는 명령이어야 합니다. 이 예제는 --system 옵션을 소개합니다. --system 옵션은 전체 시스템의 구성을 설정합니다. 즉, 모든 사용자와 시

시스템의 repos를 의미합니다. 구성 수준에 대한 자세한 내용은 git config 페이지를 참조하십시오. git config --global --edit 수동 편집을 위해 텍스트 편집기에서 글로벌 구성 파일을 엽니다. git이 사용할 텍스트 편집기를 구성하는 방법에 대한 자세한 안내는 Git config 페이지에서 찾을 수 있습니다. Git은 구성 옵션을 세 개의 개별 파일에 저장하므로 개별 저장소, 사용자 또는 전체 시스템에 대한 옵션을 범위 지정할 수 있습니다. <repo> /. <repo> / .git / config - 저장소 별 설정. ~ / .gitconfig - 사용자 별 설정. 여기에는 --global 플래그로 설정된 옵션이 저장됩니다. \$ (접두사) / etc / gitconfig - 시스템 전체 설정 이 파일의 옵션이 충돌하면 로컬 설정이 시스템 설정보다 우선하는 사용자 설정보다 우선합니다. 이러한 파일 중 하나를 열면 다음과 같은 내용이 표시됩니다. [user] name = John Smith email = john@example.com [alias] st = status co = checkout br = branch up = rebase ci = commit [core] editor = vim 이 값을 수동으로 git config와 동일한 효과로 편집 할 수 있습니다. Git 가이드

2019-06-15 17:12:02.931

Git Diff

변경 비교하기: git diff Diffing은 두 개의 입력 데이터 집합을 가져 와서 변경 내용을 출력하는 함수입니다. git diff는 실행시 Git 데이터 소스에서 diff 함수를 실행하는 다중 사용 Git 명령어입니다. 이러한 데이터 소스는 커밋, 분기, 파일 등이 될 수 있습니다. 이 문서는 git diff와 diffing 작업 흐름 패턴의 일반적인 호출에 대해 설명합니다. git diff 명령은 git status 및 git log와 함께 사용되어 Git repo의 현재 상태를 분석합니다. Reading diffs: outputs Raw output format 다음 예제는 간단한 repo에서 실행됩니다. repo는 아래 명령으로 생성됩니다. \$ mkdir diff_test_repo \$ cd diff_test_repo \$ touch diff_test.txt \$ echo "this is a git diff test example" > diff_test.txt \$ git init. Initialized empty Git repository in /Users/kev/code/test/.git/ \$ git add diff_test.txt \$ git commit -am"add diff test file" [master (root-commit) 6f77fc3] add diff test file 1 file changed, 1 insertion(+) create mode 100644 diff_test.txt 이 시점에서 git diff를 실행하면 결과가 출력되지 않습니다. diff에 대한 변경 사항이 없으므로 이는 예상되는 동작입니다. repo가 생성되고 diff_test.txt 파일이 추가되면 diff 출력을 실행하기 위해 파일 내용을 변경할 수 있습니다. \$ echo "this is a diff example" > diff_test.txt 이 명령을 실행하면 diff_test.txt 파일의 내용이 변경됩니다. 일단 수정되면 diff를보고 출력을 분석할 수 있습니다. 이제 git diff를 실행하면 다음과 같은 결과가 출력됩니다. \$ diff --git a/diff_test.txt b/diff_test.txt index 6b0c6cf..b37e70a 100644 --- a/diff_test.txt +++ b/diff_test.txt @@ -1 +1 @@ -this is a git diff test example +this is a diff example 이제 diff 출력에 대한 자세한 분석을 살펴 보겠습니다. 1. Comparison input \$ diff --git a/diff_test.txt b/diff_test.txt 이 행은 diff의 입력 소스를 표시합니다. a / diff_test.txt와 b / diff_test.txt가 diff에 전달되었음을 알 수 있습니다. 2. Meta data index 6b0c6cf..b37e70a 100644 이 행은 내부 Git 메타 데이터를 표시합니다. 이 정보가 필요 없을 것입니다. 이 출력의 숫자는 Git 체크아웃 해시 식별자에 해당합니다. 3. Markers for changes --- a/diff_test.txt +++ b/diff_test.txt 이 행은 각 diff 입력 소스에 기호를 지정하는 범례입니다. 이 경우 / diff_test.txt의 변경 사항은 ---로 표시되고 b / diff_test.txt의 변경 사항은 +++ 기호로 표시됩니다. 4. Diff chunks 나머지 diff 출력은 diff 'chunk' 목록입니다. diff는 변경 사항이있는 파일의 섹션 만 표시합니다. 현재 시나리오에서는 간단한 시나리오로 작업하면서 하나의 체크 만 있습니다. 덩어리는 독자적인 세분화 된 출력 의미를 가지고 있습니다. @@ -1 +1 @@ -this is a git diff test example +this is a diff example 첫 번째 줄은 체크 헤더입니다. 각 체크는 @@ 기호 안에 포함 된 헤더 앞에 추가됩니다. 헤더의 내용은 파일에 대한 변경 사항 요약입니다. 우리의 단순화된 예에서 우리는 -1 +1 의미 라인 하나가 변경되었습니다. 보다 현실적인 diff에서는 다음과 같은 헤더가 표시됩니다. @@ -34,6 +34,8 @@ 이 머리글 예제에서는 34 행부터 6 행을 추출했습니다. 또한 34 행부터 8 행을 추가했습니다. diff chunk의 나머지 내용은 최근 변경 사항을 표시합니다. 변경된 각 줄 앞에는 + 또는 - 기호가 표시되어 변경 사항의 원본 버전을 나타냅니다. 앞에서 설명한 것처럼 -는 a / diff_test.txt의 변경 사항을 나타내며 +는 b / diff_test.txt의 변경 사항을 나타냅니다. 변경 사항 강조 표시 1. git diff --color-words git diff는 --color-words보다 훨씬 세분화 된 변경 사항을 강조하기 위한 특수 모드도 제공합니다. 이 모드는 추가되거나 제거 된 행을 공백으로 토큰화 한 다음 그 행과 비교합니다. \$ git diff --color-words \$ diff --git a/diff_test.txt b/diff_test.txt index 6b0c6cf..b37e70a 100644 --- a/diff_test.txt +++ b/diff_test.txt @@ -1 +1 @@ this is a git diff test example 이 출력물에는 변경된 색으로 구분 된 단어 만 표시됩니다. 2. git diff-highlight git 소스를 복제하면 contrib이라는 하위 디렉토리가 있습니다. 여기에는 여러 git 관련 도구가 포함되어 있으며 흥미로운 비트와 조각은 아직 핵심 노드로 승격되지 않았습니다. 이 중 하나는 diff-highlight라는 Perl 스크립트입니다. diff 강조 표시는 diff 출력의 일치하는 행을 짝 지어주고 변경된 하위 단어 조각을 강조 표시합니다. \$ git diff | /your/local/path/to/git-core/contrib/diff-highlight/diff-highlight diff --git a/diff_test.txt b/diff_test.txt index 6b0c6cf..b37e70a 100644 --- a/diff_test.txt +++ b/diff_test.txt @@ -1 +1 @@ -this is a git diff test example +this is a diff example 이제 우리는 가능한 가장 작은 변경 사항을 비교해 보았습니다. 바이너리 파일 비교하기 우리가 지금까지 보여준 텍스트 파일 유틸리티 외에도 git diff는 바이너리 파일에서 실행될 수 있습니다. 불행히도, 기본 출력은 별로 도움이 되지 않습니다. \$ git diff Binary files a/script.pdf and b/script.pdf differ Git에는 diff를 수행하기 전에 이진 파일의 내용을 텍스트로 변환하는 쉘 명령을 지정할 수

있는 기능이 있습니다. 그래도 약간의 설정이 필요합니다. 먼저 특정 유형의 바이너리를 텍스트로 변환하는 방법을 설명하는 textconv 필터를 지정해야 합니다. 우리는 PDF 파일을 사람이 읽을 수 있는 HTML로 변환하기 위해 pdftohtml이라는 간단한 유틸리티 (homebrew에서 사용 가능)를 사용하고 있습니다. 이 설정은 .git / config 파일을 편집하여 단일 저장소에 대해 설정할 수도 있고 ~ /.gitconfig를 편집하여 전역으로 설정할 수도 있습니다 [diff "pdfconv"] textconv=pdftohtml -stdout 그런 다음 하나 이상의 파일 패턴을 pdfconv 필터와 연결하기 만하면 됩니다. 리포지토리의 루트에 .gitattributes 파일을 만들어이 작업을 수행 할 수 있습니다. *.pdf diff=pdfconv 일단 설정되면, git diff는 먼저 구성된 변환기 스크립트를 통해 이진 파일을 실행하고 변환기 출력을 diff합니다. jips, jars 및 기타 아카이브 : pdf2html 대신 unzip -l (또는 유사)을 사용하면 추가되거나 제거 된 경로가 표시됩니다. 이미지 커밋 : exiv2는 이미지 차원 문서와 같은 메타 데이터 변경 사항을 표시하는 데 사용할 수 있습니다. 변환 도구는 .odf, .doc 및 기타 문서 형식을 일반 텍스트로 변환하기 위해 존재합니다. 핀치 (pinch)에서 문자열은 형식 변환기가없는 바이너리 파일에서 종종 작동합니다. 파일 비교하기: git diff file git diff 명령은 명시 적 파일 경로 옵션을 전달할 수 있습니다. git diff에 파일 경로가 전달되면 diff 작업의 범위가 지정된 파일로 지정됩니다. 아래 예제는 이러한 사용법을 보여줍니다. \$ git diff HEAD ./path/to/file 이 예제는 호출 될 때 ./path/to/file로 범위가 지정되며 작업 디렉토리의 특정 변경 사항을 인덱스와 비교하여 아직 준비되지 않은 변경 사항을 보여줍니다. 기본적으로 git diff는 HEAD와의 비교를 실행합니다. 위의 예에서 HEAD를 생략하면 git diff ./path/to/file과 동일한 효과가 나타납니다. \$ git diff --cached ./path/to/file git diff가 --cached 옵션과 함께 호출되면 diff는 단계적 변경 사항을 로컬 저장소와 비교합니다. --cached 옵션은 --staged와 동의어입니다. 모든 변경 비교하기 git diff를 파일 경로없이 사용하면 전체 저장소의 변경 사항을 비교할 수 있습니다. 위의 파일 별 예제는 ./path/to/file 인수없이 호출 할 수 있으며 로컬 repo의 모든 파일에서 동일한 출력 결과를 갖습니다. 마지막 커밋 이후 변경 보기 기본적으로 git diff는 마지막 커밋 이후 커밋되지 않은 변경 사항을 보여줍니다. \$ git diff 서로 다른 커밋의 파일 비교하기 git diff는 diff에 커밋을 전달할 수 있습니다. 몇 가지 예제 ref는 HEAD, 태그 및 분기 이름입니다. Git의 모든 커밋에는 GIT LOG를 실행할 때 얻을 수있는 커밋 ID가 있습니다. git diff에이 커밋 ID를 전달할 수도 있습니다. \$ git log --pretty=oneline 957fbc92b123030c389bf8b4b874522bdf2db72c add feature ce489262a1ee34340440e55a0b99ea6918e19e7a rename some classes 6b539f280d8b0ec4874671bae9c6bed80b788006 refactor some code for feature 646e7863348a427e1ed9163a9a96fa759112f102 add some copy to body \$ git diff 957fbc92b123030c389bf8b4b874522bdf2db72c ce489262a1ee34340440e55a0b99ea6918e19e7a 브랜치 비교하기 두 개의 브랜치 비교하기 브랜치는 다른 모든 ref 입력과 마찬가지로 git diff와 비교됩니다. \$ git diff branch1...other-feature-branch 이 예제는 도트 연산자를 소개합니다. 이 예에서 두 점은 diff 입력이 두 가지의 팁임을 나타냅니다. 점이 생략되고 분기 사이에 공백이 사용되는 경우에도 동일한 효과가 발생합니다. 또한 점 연산자 3 개가 있습니다. \$ git diff branch1...other-feature-branch 세개의 도트 연산자는 첫 번째 입력 매개 변수 branch1을 변경하여 diff를 시작합니다. branch1을 두 diff 입력, 즉 branch1과 other-feature-branch의 공유 조상 사이의 공유 공통 조상 커밋의 ref로 변경합니다. 마지막 매개 변수 입력 매개 변수는 other-feature-branch의 끝으로 변경되지 않습니다. 두 브랜치의 파일 비교하기 브랜치간에 특정 파일을 비교하려면 파일 경로를 git diff의 세 번째 인수로 전달하십시오 git diff master new_branch ./diff_test.txt Git 가이드

2019-06-15 16:42:22.986

Git Commit

12.png git commit 명령은 프로젝트의 현재 단계별 변경 사항에 대한 스냅 샷을 캡처합니다. 커밋 된 스냅 샷은 프로젝트의 "안전한"버전으로 생각할 수 있습니다. Git은 명시적으로 요청하지 않으면 변경하지 않습니다. git commit을 실행하기 전에 git add 명령을 사용하여 커밋에 저장 될 프로젝트의 변경 사항을 승격 또는 '수행'합니다. 이 두 명령 git commit과 git add는 가장 자주 사용되는 두 가지 명령입니다. Git commit 과 SVN commit 차이점 동일한 이름을 공유하는 동안 git commit은 svn commit과 같은 것이 아닙니다. 이 공유 용어는 svn 배경을 가진 Git 신규 사용자에게 혼란의 포인트가 될 수 있으며 그 차이를 강조하는 것이 중요합니다. git commit 대 svn commit을 비교하는 것은 중앙 집중식 애플리케이션 모델 (svn)과 분산 애플리케이션 모델 (Git)을 비교하는 것입니다. SVN에서 커밋은 변경 사항을 로컬 SVN 클라이언트에서 원격 중앙 집중식 공유 SVN 저장소로 푸시합니다. Git에서는 저장소가 배포되고 스냅 샷은 로컬 저장소에 커밋되며 다른 Git 저장소와는 전혀 상호 작용이 필요하지 않습니다. 자식 커밋은 나중에 임의의 원격 리포지토리에 푸시될 수 있습니다. How it works 높은 수준에서 Git은 타임 라인 관리 유틸리티로 생각할 수 있습니다. 커밋은 Git 프로젝트 타임 라인의 핵심 빌딩 블록 단위입니다. 커밋은 Git 프로젝트의 타임 라인을 따라 스냅 샷 또는 마일스톤으로 생각할 수 있습니다. 커밋은 git commit 명령으로 생성되어 해당 시점의 프로젝트 상태를 캡처합니다. 스냅 샷은 항상 로컬 저장소에 커밋됩니다. 이는 SVN과 근본적으로 다르며 작업 카피는 중앙 저장소에 커밋됩니다. 대조적으로, 힘내라는 준비가 끝날 때까지 중앙 저장소와 상호 작용하도록 강요하지 않습니다. 준비 영역은 작업 디렉토리와 프로젝트 기록 사이의 버퍼이므로 각 개발자의 로컬 저장소는 기여와 중앙 저장소 사이의 버퍼입니다. Git 사용자를위한 기본 개발 모델을 변경합니다. Git 개발자는 변경 사항을 중앙 repo에 직접 커밋하는 대신 로컬 레포에 커밋을 추적 할 수 있습니다. 이것은 SVN 스타일의 협업에 비해 많은 이점을 가지고 있습니다. 즉, 기능을 원자 단위로 분할하고, 관련 커밋을 그룹화하고, 중앙 저장소에 공개하기 전에 로컬 내역을 정리하는 것을 더 쉽게 만듭니다. 또한 개발자는 격리 된 환경에서 작업 할 수 있으므로 다른 사용자와 편리하게 병합 할 때까지 통합이 지연됩니다. 격리 및 지연된 통합은 개별적으로 유익하지만, 자주 및 작은 단위로 통합하는 것이 팀의 이익입니다. 힘내 팀 공동 작업에 대한 우수 사례에 대한 자세한 내용은 팀에서 어떻게 힘내 워크 플로우를 구성하는지 읽어보십시오. Snapshots, not differences SVN과 Git 사이의 실질적인 차이점을 제외하고는 기본 구현은 완전히 다른 디자인 철학을 따릅니다. SVN이 파일의 차이를 추적하는 반면 Git의 버전 제어 모델은 스냅 샷을 기반으로합니다. 예를 들어, SVN 커밋은 저장소에 추가 된 원본 파일과 비교되는 diff로 구성됩니다. Git은 모든 커밋에서 각 파일의 전체 내용을 기록한다. 14-1.png 이는 특정 버전의 파일을 diff에서 "어셈블"할 필요가 없기 때문에 많은 Git 작업을 SVN보다 훨씬 빠르게 만듭니다. 각 파일의 전체 버전은 Git의 내부 데이터베이스에서 즉시 사용할 수 있습니다. Git의 스냅 샷 모델은 버전 제어 모델의 거의 모든 측면에 광범위한 영향을 미치며 분기 및 병합 도구에서부터 협업 작업 흐름에 이르는 모든 것에 영향을 미칩니다. 공동 옵션 git commit 준비된 스냅 샷을 커밋합니다. 그러면 커밋 메시지를 묻는 텍스트 편집기가 시작됩니다. 메시지를 입력 한 후 파일을 저장하고 편집기를 닫아 실제 커밋을 만듭니다. git commit -a 작업 디렉토리의 모든 변경 사항에 대한 스냅 샷을 커밋합니다. 이것은 오직 추적 된 파일 (히스토리의 어느 시점에서 git add로 추가 된 것들)에 대한 수정 만 포함합니다. git commit -m "commit message" 전달 된 커밋 메시지로 즉시 커밋을 만드는 바로 가기 명령입니다. 기본적으로 git commit은 로컬로 구성된 텍스트 편집기를 열고 커밋 메시지를 입력하라는 메시지를 표시합니다. -m 옵션을 전달하면 인라인 메시지를 선호하는 텍스트 편집기 프롬프트가 표시되지 않습니다. git commit -am "commit message" -a 및 -m 옵션을 결합한 파워 유저 바로 가기 명령입니다. 이 조합은 즉시 모든 단계적 변경 사항의 커밋을 생성하고 인라인 커밋 메시지를받습니다. git commit --amend 이 옵션은 커밋 명령에 다른 수준의 기능을 추가합니다. 이 옵션을 전달하면 마지막 커밋이 수정됩니다. 새 커밋을 만드는 대신 이전 커밋에 단계별 변경 사항이 추가됩니다. 이 명령은 시스템에 구성된 텍스트 편집기를 열고 이전에 지정한 커밋 메시지를 변경하라는 메시지를 표시합니다. Examples 커밋으로 변경 저장하기 다음 예제는 현재 브랜치에서 hello.py라는 파일의 일부 내용을 편집 한 것으로 가정하고 프로젝트 히스토리에 커밋 할 준비가 되어 있다고 가정합니다. 먼저 git add로 파일을 준비한 다음 준비된 스냅 샷을 커밋 할 수 있습니다. git add hello.py 이 명령은 hello.py를 Git 스테이징 영역에 추가합니다. git status 명령을 사용하여 작업의 결과를 검사 할 수 있습니다. git status On branch master Changes to be committed: (use "git reset HEAD <file>..." to unstage) new file: hello.py 녹색 출력 새로운 파일 : hello.py는 hello.py가 다음 커밋과 함께 저장 될 것임을 나타냅니다. 커밋에서 다음을 실행하여 만듭니다. git commit 커밋 된 로그 메시지를 요구하는 텍스트 편집기 (git config를 통해 사용자 정의 가능)가 열립니다. # Please enter the commit message for your changes. Lines starting # with '#' will be ignored, and an empty message aborts the commit. # On branch master # Changes to be committed: # (use "git reset HEAD ..." to unstage) # #modified: hello.py Git은 특정 형식 제약 조건을 따르는 커밋 메시지를 요구하지 않지만 표준 형식은 첫 줄에 전체 커밋을 50 자 미만으로 요약하고 빈 줄을 남기고 변경된 내용에 대한 자세한 설명을 제공합니다. 예 : Change the message displayed by hello.py - Update the sayHello() function to output the user's name - Change the sayGoodbye() function to a friendlier message 전자 메일과 마찬가지로 커밋 메시지의 첫 번째 줄을 계속 줄로 사용하는 것이 일반적입니다. 나머지 로그 메시지는 본문으로 간주되어 커밋 변경 집합의 세부 정보를 전달하는 데 사용됩니다. 많은 개발자들이 커밋 메시지에 현재 시계를 사용하기를 좋아한다는 점에 유의하십시오. 이것은 리포지토리에서의 작업과 같이 더 많이 읽을 수 있게 해줌으로써 많은 역사 작성 작업을 더욱 직관적으로 만듭니다. 마지막 커밋 수정하기 위의 hello.py 예제를 계속 진행하십시오. hello.py에 대한 추가 업데이트를하고 다음을 실행합니다. git add hello.py git commit --amend 그러면 다시 구성된 텍스트 편집기가 열립니다. 그러나 이번에는 이전에 입력 한 커밋 메시지로 미리 채워질 것입니다. 이것은 우리가 새로운 커밋을 생성하는 것이 아니라 마지막 커밋을 편집하고 있음을 나타냅니다. Git 가이드

2019-06-15 19:41:44.771

.gitignore

17.png Git은 작업 카피의 모든 파일을 세 가지 중 하나로 본다. 추적 - 이전에 수행되었거나 확장 된 파일. untracked - 준비되거나 커밋되지 않은 파일. 또는 무시 됨 - Git이 명시 적으로 무시하도록 지시 한 파일. 무시 된 파일은 일반적으로 저장소 소스에서 파생되거나 커밋되지 않아야 하는 빌드 아티팩트 및 컴퓨터 생성 파일입니다. 몇 가지 일반적인 예는 다음과 같습니다. / node_modules 또는 / 패키지의 내용과 같은 종속성 캐시 .o, .pyc 및 .class 파일과 같은 컴파일 된 코드 / bin, / out 또는 / target과 같은 출력 디렉토리 빌드 .log, .lock 또는 .tmp와 같이 런타임에 생성되는 파일 .DS_Store 또는 Thumbs.db와 같은 숨겨진 시스템 파일 .idea / workspace.xml과 같은 개인 IDE 구성 파일 무시 된 파일은 저장소의 루트에서 체크인 된 .gitignore라는 특수 파일에서 추적됩니다. 명시 적으로 git ignore 명령은 없습니다. 대신에 .gitignore 파일을 편집하고 커밋해야 합니다. 무시할 새 파일이 있으면 수동으로 커밋해야 합니다. .gitignore 파일은 저장소의 파일 이름과 일치하는 패턴을 포함하여 무시해야 하는지 여부를 결정합니다. Git ignore 패턴 .gitignore는 globbing 패턴을 사용하여 파일 이름과 일치시킵니다. 다양한 기호를 사용하여 패턴을 구성 할 수 있습니다. Pattern Example matches Explanation* **/logs logs /debug.log logs/monday/foo.bar build/logs/debug.log You can prepend a pattern with a double asterisk to match directories anywhere in the repository. **/logs/debug.log logs/debug.log build/logs/debug.log but not logs/build/debug.log You can also use a double asterisk to match files based on their name and the name of their parent directory. *.log debug.log foo.log .log logs/debug.log An asterisk is a wildcard that matches zero or more characters. *.log !important.log debug.log trace.log but not important.log logs/important.log Prepending an exclamation mark to a pattern negates it. If a file matches a pattern, but alsomatches a negating pattern defined later in the file, it will not be ignored. *.log !important/*.log trace.* debug.log important/trace.log but not important/debug.log Patterns defined after a negating pattern will re-ignore any previously negated files. /debug.log debug.log but not logs/debug.log Prepending a slash matches files only in the repository root. debug.log debug.log logs /debug.log By default, patterns match files in any directory debug?.log debug0.log debug.log but not debug10.log A question mark matches exactly one character. debug[0-9].log debug0.log debug1.log but not debug10.log Square brackets can also be used to match a single character from a specified range. debug[01].log debug0.log debug1.log but not debug2.log debug01.log Square brackets match a single character form the specified set. debug[!01].log debug2.log but not debug0.log debug1.log debug01.log An exclamation mark can be used to match any character except one from the specified set. debug[a-z].log debuga.log debugb.log but not debug1.log Ranges can be numeric or alphabetic. logs logs/debug.log logs/latest/foo.bar build/logs build/logs/debug.log If you don't append a slash, the pattern will match both files and the contents of directories with that name. In the example matches on the left, both directories and files named logs are ignored logs/ logs/debug.log logs /latest/foo.bar build/logs/foo.bar build/logs/latest/debug.log Appending a slash indicates the pattern is a directory. The entire contents of any directory in the repository matching that name - including all of its files and subdirectories - will be ignored logs/ !logs/important.log logs/debug.log logs/important.log Wait a minute! Shouldn't logs/important.logbe negated in the example on the left Nope! Due to a performance-related quirk in Git, you can not negate a file that is ignored due to a pattern matching a directory logs/**/debug.log logs/debug.log logs/monday/debug.log logs /monday/pm/debug.log A double asterisk matches zero or more directories. logs/*day/debug.log logs/monday/debug.log logs/tuesday /debug.log but not logs/latest/debug.log Wildcards can be used in directory names as well. logs/debug.log logs/debug.log but not debug.log build/logs /debug.log Patterns specifying a file in a particular directory are relative to the repository root. (You can prepend a slash if you like, but it doesn't do anything special.) **이 설명은 .gitignore 파일이 협약과 마찬가지로 저장소의 최상위 디렉토리에 있다고 가정합니다. 저장소에 .gitignore 파일이 여러 개있는 경우 "저장소 루트"를 ".gitignore 파일이있는 디렉토리"로 정신적으로 대체하십시오 (팀의 온전한 고려를 위해 통합을 고려하십시오). * 이러한 문자 외에도 .gitignore 파일에 #을 사용하여 주석을 포함 할 수 있습니다. # ignore all logs *.log 파일이나 디렉토리가있는 경우 ##를 사용하여 .gitignore 패턴 문자를 이스케이프 처리 할 수 있습니다. # ignore the file literally named foo[01].txt foo##[01##].txt 저장소에 .gitignore 공유하기 Git 무시 규칙은 일반적으로 저장소의 루트에있는 .gitignore 파일에 정의되어 있습니다. 그러나 저장소의 여러 디렉토리에 여러 .gitignore 파일을 정의하도록 선택할 수 있습니다. 특정 .gitignore 파일의 각 패턴은 해당 파일이 들어 있는 디렉토리를 기준으로 테스트됩니다. 그러나 규칙과 가장 단순한 접근법은 루트에 단일 .gitignore 파일을 정의하는 것입니다. .gitignore 파일이 체크인되면 리포지토리의 다른 파일과 마찬가지로 버전이 지정되고 무시 할 때 팀원과 공유됩니다. 일반적으로 .gitignore에는 저장소의 다른 사용자에게 도움이 되는 패턴 만 포함시켜야 합니다. Personal Git ignore rules 특정 저장소에 대한 개인 무시 패턴을 .git / info / exclude의 특수 파일에 정의 할 수도 있습니다. 이러한 버전은 버전이 지정되지 않으며 저장소와 함께 배포되지 않으므로 이점 만 얻을 수있는 패턴을 포함하는 것이 좋습니다. 예를 들어 사용자 정의 로깅 설정이나 저장소의 작업 디렉토리에 파일을 생성하

는 특수 개발 도구가있는 경우 .git / info / exclude에 추가하여 실수로 저장소에 커밋되지 않도록 할 수 있습니다. Global Git ignore rules 또한 Git core.excludesFile 속성을 설정하여 로컬 시스템의 모든 리포지토리에 대한 전역 Git 무시 패턴을 정의 할 수 있습니다. 이 파일을 직접 만들어야합니다. 전역 .gitignore 파일을 어디에 둘지 확실치 않으면 홈 디렉토리가 나쁜 선택이 아니며 나중에 쉽게 찾을 수 있습니다. 파일을 만들었 으면 git config로 위치를 구성해야 합니다. \$ touch ~/.gitignore \$ git config --global core.excludesFile ~/.gitignore 서로 다른 파일 유형이 서로 다른 프로젝트와 관련되므로 어떤 패턴을 전역 적으로 무시할지 선택해야 합니다. 일부 개발자 도구로 만든 특별 운영 체제 파일 (예 : DS_Store 및 thumbs.db) 또는 임시 파일은 전 세계적으로 무시할 수있는 대표적인 후보입니다. Ignoring a previously committed file 과거에 커밋 한 파일을 무시하려면 저장소에서 파일을 삭제 한 다음 .gitignore 규칙을 추가해야 합니다. git rm과 함께 --cached 옵션을 사용하면 파일이 저장소에서 삭제되지만 작업 디렉토리에는 무시 된 파일로 남아있게 됩니다. \$ echo debug.log >> .gitignore \$ git rm --cached debug.log rm 'debug.log' \$ git commit -m "Start ignoring debug.log" 저장소 및 로컬 파일 시스템에서 파일을 삭제하려면 --cached 옵션을 생략 할 수 있습니다. Committing an ignored file git add와 함께 -f (또는 --force) 옵션을 사용하여 무시 된 파일을 저장소에 강제로 적용 할 수 있습니다. \$ cat .gitignore *.log \$ git add -f debug.log \$ git commit -m "Force adding debug.log" 일반적인 패턴 (예 : *.log)이 정의되어 있지만 특정 파일을 커밋하려는 경우가 작업을 고려할 수 있습니다. 그러나 더 좋은 해결책은 일반적인 규칙에 대한 예외를 정의하는 것입니다. \$ echo !debug.log >> .gitignore \$ cat .gitignore *.log !debug.log \$ git add debug.log \$ git commit -m "Adding debug.log" 이 접근법은 팀원에게보다 분명하고 혼란스럽지 않습니다. Stashing an ignored file git stash는 로컬 변경 사항을 일시적으로 저장 및 되돌리기위한 강력한 Git 기능으로, 나중에 다시 적용 할 수 있습니다. 예상대로, 기본적으로 git stash는 무시 된 파일을 무시하고 Git에 의해 추적되는 파일에 대한 변경 사항 만 숨 깁니다. 그러나, --all 옵션을 사용하여 git stash를 호출하면 무시되거나 변경되지 않은 파일에 변경 사항을 숨길 수 있습니다. Debugging .gitignore files 복잡한 .gitignore 패턴이 있거나 여러 .gitignore 파일에 패턴이 퍼져있는 경우 특정 파일을 왜 무시하는지 추적하기가 어려울 수 있습니다. git check-ignore 명령을 -v (또는 --verbose) 옵션과 함께 사용하여 어떤 패턴이 특정 파일을 무시하게 하는지 결정할 수 있습니다. \$ git check-ignore -v debug.log .gitignore:3:*.log debug.log 결과: <file containing the pattern>: <line number of the pattern>: <pattern> <file name> 원하는 경우 여러 파일 이름을 git check-ignore에 전달할 수 있으며 이름 자체는 저장소에있는 파일에 해당 할 필요조차 없습니다. Git 가이드

2019-06-15 18:41:23.002

Git Stash

작업 복사본에 대한 숨김 보관함 (또는 숨김) 변경 내용을 숨겨서 다른 작업을 할 수 있게 한 다음 나중에 다시 적용하여 나중에 다시 적용 할 수 있습니다. 숨기기는 컨텍스트를 신속하게 전환하고 다른 작업을해야 하는 경우 편리하지만 코드를 변경하는 중반에 있으며 커밋 할 준비가되지 않았습니 다. 작업 숨기기 git stash 명령은 커밋되지 않은 변경 (준비 및 언 스테이지 모두)을 취하고 나중에 사용하기 위해 저장 한 다음 작업 복사본에서 되돌립니다. 예 : \$ git status On branch master Changes to be committed: new file: style.css Changes not staged for commit: modified: index.html \$ git stash Saved working directory and index state WIP on master: 5002d47 our new homepage HEAD is now at 5002d47 our new homepage \$ git status On branch master nothing to commit, working tree clean 이 시점에서 여러분은 자유롭게 변경을 할 수 있고, 새로운 커밋을 만들고, 분기를 전환하고, 다른 Git 작업을 수행 할 수 있습니다. 그런 다음 다시 돌아와서 준비가되었을 때 은닉을 다시 적용하십시오. 숨겨진 저장소는 Git 저장소에 로컬로 저장됩니다. 밀어 넣을 때 은폐는 서버로 전송되지 않습니다. 숨겨진 변경 적용하기 git stash pop으로 이전에 숨긴 변경 사항을 다시 적용 할 수 있습니다. \$ git status On branch master nothing to commit, working tree clean \$ git stash pop On branch master Changes to be committed: new file: style.css Changes not staged for commit: modified: index.html Dropped refs/stash@{0} (32b3aa1d185dfe6d57b3c3cc3b32cbf3e380cc6a) Popping your stash removes the changes from your stash and reapplies them to your working copy. Alternatively, you can reapply the changes to your working copy and keep them in your stash with git stash apply: \$ git stash apply On branch master Changes to be committed: new file: style.css Changes not staged for commit: modified: index.html 여러 분기에 동일한 숨겨진 변경 사항을 적용하려는 경우 유용합니다. stashing의 기본 사항을 알았으므로 git stash를주의해야 합니다. 기본적으로 Git은 추적되지 않거나 무시되는 파일의 변경 사항을 숨기지 않습니다. untracked or ignored 파일 숨기기 기본적으로 git stash를 실행하면 다음 항목이 숨겨집니다. 색인에 추가 된 변경 사항 (단계별 변경 사항) 현재 Git에 의해 추적되는 파일에 대한 변경 사항 (unstaged changes) 그러나 다음 항목은 숨겨지지 않을 것입니다: 작업 복사본에서 아직 준비되지 않은 새 파일 무시 된 파일 위 예제에 세 번째 파일을 추가했지만 스테이지를 만들지 않으면 (즉, git add를 실행하지 않음) git stash가 stash하지 않습니다. \$ script.js \$ git status On branch master Changes to be committed: new file: style.css Changes not staged for commit: modified: index.html Untracked files: script.js \$ git stash Saved working directory and index state WIP on master: 5002d47 our new

homepage HEAD is now at 5002d47 our new homepage \$ git status On branch master Untracked files: script.js -u 옵션 (또는 --include-untracked)을 추가하면 git stash가 untracked 파일을 숨긴다. \$ git status on branch master Changes to be committed: new file: style.css Changes not staged for commit: modified: index.html Untracked files: script.js \$ git stash -u Saved working directory and index state WIP on master: 5002d47 our new homepage HEAD is now at 5002d47 our new homepage \$ git status On branch master nothing to commit, working tree clean git stash를 실행할 때 -a 옵션 (또는 --all)을 전달하여 무시 된 파일에 대한 변경 사항을 포함 할 수 있습니다. 15.png 숨겨진 목록 (stash) 관리하기 git stash를 여러 번 실행하여 숨겨진 목록을 만든 다음 git stash list를 사용하여 볼 수 있습니다. 기본적으로 stash는 작성된 분기 또는 커밋 위의 "진행중인 작업"(WIP)으로 식별 됩니다. 잠시 후 각 은닉 내용을 기억하는 것은 어려울 수 있습니다. \$ git stash list stash@{0}: WIP on master: 5002d47 our new homepage stash@{1}: WIP on master: 5002d47 our new homepage stash@{2}: WIP on master: 5002d47 our new homepage 좀 더 많은 문맥을 제공하기 위해 stash에 설명을 추가하고 git stash save "message"를 사용하여 주석을 달아주는 것이 좋습니다. \$ git stash save "add style to our site" Saved working directory and index state On master: add style to our site HEAD is now at 5002d47 our new homepage \$ git stash list stash@{0}: On master: add style to our site stash@{1}: WIP on master: 5002d47 our new homepage stash@{2}: WIP on master: 5002d47 our new homepage 기본적으로 git stash pop은 가장 최근에 생성된 stash를 다시 적용합니다 : stash @ {0} 식별자를 마지막 인수로 전달하여 다시 적용 할 스택을 선택할 수 있습니다 (예 : \$ git stash pop stash@{2} 숨겨진 내용 비교 (stash diffs) git stash show를 사용하여 stash 요약 볼 수 있습니다. \$ git stash show index.html | 1 + style.css | 3 +++ 2 files changed, 4 insertions(+) 또는 stash 전체 diff를 보려면 -p 옵션 (또는 - patch)을 전달하십시오. \$ git stash show -p diff --git a/style.css b/style.css new file mode 100644 index 0000000..d92368b --- /dev/null +++ b/style.css @@ -0,0 +1,3 @@ ++ { + text-decoration: blink; +} diff --git a/index.html b/index.html index 9daea6b..ebdcbd2 100644 --- a/index.html +++ b/index.html @@ -1 +1,2 @@ @ + <link rel="stylesheet" href="style.css"/> 부분 stashes 단일 파일, 파일 모음 또는 파일 내 개별 변경 사항을 숨기도록 선택할 수도 있습니다. git stash에 -p 옵션 (또는 --patch)을 전달하면 작업 복사본에서 변경된 각 "덩어리"를 반복하여 숨길지 여부를 묻습니다. \$ git stash -p diff --git a/style.css b/style.css new file mode 100644 index 0000000..d92368b --- /dev/null +++ b/style.css @@ -0,0 +1,3 @@ ++ { + text-decoration: blink; +} Stash this hunk [y,n,q,a,d,/,e,?]? y diff --git a/index.html b/index.html index 9daea6b..ebdcbd2 100644 --- a/index.html +++ b/index.html @@ -1 +1,2 @@ @ + <link rel="stylesheet" href="style.css"/> Stash this hunk [y,n,q,a,d,/,e,?]? n 15-2.png 일반적으로 유용한 것들은 다음과 같습니다 : Command Description / search for a hunk by regex ? help n don't stash this hunk q quit (any hunks that have already been selected will be stashed) s split this hunk into smaller hunks y stash this hunk 명시 적 "중단"명령은 없지만 CTRL-C (SIGINT)를 누르면 숨김 프로세스가 중단됩니다. Stash로부터 브랜치 생성하기 브랜치의 변경사항이 stash의 변경사항과 다른 경우 stash를 적용 할 때 충돌이 발생할 수 있습니다. 대신 git stash branch를 사용하여 숨겨진 변경 사항을 적용 할 새 브랜치를 만들 수 있습니다 : \$ git stash branch add-stylesheet stash@{1} Switched to a new branch 'add-stylesheet' On branch add-stylesheet Changes to be committed: new file: style.css Changes not staged for commit: modified: index.html Dropped refs/stash@{1} (32b3aa1d185dfe6d57b3c3cc3b32cbf3e380cc6a) 이것은 당신이 숨긴 곳을 만든 커밋을 기반으로 새로운 브랜치를 체크 아웃하고 숨겨진 변경 사항을 그 위에 놓습니다. stash 비우기 더 이상 stash가 필요 없다고 판단되면 git stash drop으로 삭제할 수 있습니다. \$ git stash drop stash@{1} Dropped stash@{1} (17e2697fd8251df6163117cb3d58c1f62a5e7cdb) 또는 다음과 같이 모든 숨기기를 삭제할 수 있습니다. \$ git stash clear How git stash works Stash가 어떻게 동작하는지 궁금하다면 이 절을 읽어도 좋습니다. 단지 stash 사용 방법이 알고 싶다면 이 절을 읽지 않아도 됩니다. stash는 저장소에서 실제로 커밋 객체로 인코딩됩니다. .git / refs / stash의 특별한 ref는 가장 최근에 생성된 stash를 가리키며, 이전에 생성된 stash는 stash ref의 reflog에 의해 참조됩니다. 이것이 당신이 stash @ {n}을 은닉하여 참조하는 이유입니다. 실제로는 stash ref의 n 번째 reflog 항목을 참조하고 있습니다. stash는 커밋 일 뿐이므로 git log로 검사 할 수 있습니다. \$ git log --oneline --graph stash@{0} * 953ddde WIP on master: 5002d47 our new homepage |W W | | * 24b35a1 untracked files on master: 5002d47 our new homepage | * 7023dd4 index on master: 5002d47 our new homepage | / * 5002d47 our new homepage 숨긴 것에 따라, 하나의 git stash 연산은 2 ~ 3 개의 새로운 커밋을 생성합니다. 위 다이어그램의 커밋은 다음과 같습니다. stash @ {0}, git stash를 실행할 때 작업 복사본에 있던 추적 된 파일을 저장하는 새로운 커밋 @ {0} 님의 첫 번째 부모를 숨기고 git stash를 실행할 때 HEAD에 있던 기존 커밋을 숨깁니다. @ {0}의 두 번째 부모를 숨기고, git stash를 실행할 때 색인을 나타내는 새로운 커밋 @ {0}의 세 번째 부모를 숨기고, git stash를 실행할 때 작업 복사본에 있던 untracked 파일을 나타내는 새로운 커밋. 이 세 번째 상위 항목은 다음 경우에만 생성됩니다. 작업 복사본에는 실제로 untracked 파일이 포함되어 있습니다. 과 git stash를 호출 할 때 --include-untracked 또는 --all 옵션을 지정했습니다. git stash가 작업 트리과 인덱스를 커밋으로 인코딩하는 방법은 다음과 같습니다. 숨겨지기 전에 작업 트리에 추적된 파일, 추적할 수 없는 파일 및 무시된 파일에 대한 변경 사항이 포함될 수 있습니다. 이러한 변경 사항 중 일부는 색인에 포함될 수도 있습니다. 16.png git stash를 호출하면 추적된 파일에 대한 모든 변경 사항이 DAG에서 두 개의 새로운 커밋으로 인코딩됩니다. 하나는 무단 변경에 대한 것이고 다른 하나는 인덱스에 준비된 변경에 대한 것입니다. 특수한 refs / stash ref가 그들을 가리 키도록 업데이트됩니다. 16-1.png --include-untracked 옵션을 사용하면 추적되지 않은 파일에 대한 변경 사항을 추가 커밋으로 인코딩 할 수 있습니다. 16-2.png --all 옵션을 사용하면 동일한 커밋에서 추적되지 않은 파일의 변경 사항과 함께 무시된 파일의 변경 사항이 포함됩니다. 16-3.png git stash pop을 실행하면 위 커밋의 변경 사항이 작업 복사본과 인덱스를 업데이트하는 데 사용되며 stash reflog는 셔플되어 팝된 커

밋을 제거합니다. 팝 된 커밋은 즉시 삭제되지 않지만 향후 가버지 수집을위한 후보가 됩니다. Git 가이드

2019-06-15 16:27:22.431

Git Add

git add 명령은 작업 디렉토리의 변경 사항을 스테이징 영역에 추가합니다. Git에게 다음 커밋에서 특정 파일에 대한 업데이트를 포함하길 원한다는 것을 알려줍니다. 그러나 git add는 저장소에 중요한 영향을 미치지 않습니다. 변경 사항은 git commit을 실행할 때까지 실제로 기록되지 않습니다. 이 명령들과 함께 작업 디렉토리와 스테이징 영역의 상태를 보려면 git 상태가 필요합니다. How it works git add와 git commit 명령은 근본적인 Git 작업 흐름을 구성합니다. 이 명령은 팀의 공동 작업 모델에 관계없이 모든 Git 사용자가 이해해야 하는 두 가지 명령입니다. 그것들은 저장소의 히스토리에 프로젝트 버전을 기록하는 수단이다. 프로젝트를 개발하는 것은 기본적인 편집 / 단계 / 커밋 패턴을 중심으로 이루어집니다. 먼저 작업 디렉토리에서 파일을 편집합니다. 프로젝트의 현재 상태 사본을 저장할 준비가 되면 git add 명령으로 변경 사항을 저장합니다. 준비된 스냅 샷에 만족하면 git commit을 사용하여 프로젝트 히스토리에 커밋합니다. git reset 명령은 커밋 또는 준비된 스냅 샷을 실행 취소하는 데 사용됩니다. git add 및 git commit 외에도 세 번째 명령 인 git push는 완전한 협업 작업 흐름을 위해 필수적입니다. git push는 협업을 위해 커밋 된 변경 사항을 원격 저장소에 보내는 데 사용됩니다. 이렇게하면 다른 팀 구성원이 저장된 변경 사항 집합에 액세스 할 수 있습니다. 11.png git add 명령을 저장소에 파일을 추가하는 svn add와 혼동해서는 안됩니다. 대신, git add는 더 추상적인 변경 레벨에서 작동합니다. 즉, 파일을 변경할 때마다 git add를 호출해야 하지만 svn add는 각 파일에 대해 한 번만 호출하면됩니다. 중복되는 것처럼 들릴 수도 있지만,이 워크 플로는 프로젝트를 체계적으로 정리하는 것을 훨씬 쉽게 만듭니다. The staging area git add 명령의 주요 기능은 작업 디렉토리에서 보류중인 변경 사항을 git staging 영역으로 승격시키는 것입니다. 준비 영역은 Git의 고유 한 기능 중 하나이며, SVN (또는 심지어 Mercurial) 배경에서 오는 경우에는 머리를 감싸는 데 약간의 시간이 걸릴 수 있습니다. 작업 디렉토리와 프로젝트 기록 사이의 버퍼로 생각하는 것이 도움이 됩니다. 준비 영역은 Git의 "세 나무"중 하나, 작업 디렉토리 및 커밋 기록 중 하나로 간주됩니다. 마지막 커밋 이후에 변경 한 사항을 모두 커밋하는 대신 스테이지에서 관련 변경 사항을 포커스가있는 스냅 샷으로 그룹화하여 실제로 프로젝트 기록에 커밋 할 수 있습니다. 즉, 무관 한 파일에 대한 모든 종류의 편집 작업을 수행 한 다음 관련 변경 사항을 스테이지에 추가하고이를 논리적 커밋으로 분리하여 개별적으로 커밋 할 수 있습니다. 모든 개정 관리 시스템에서와 마찬가지로 원자 커밋을 만들어 버그를 추적하고 변경 사항을 프로젝트의 나머지 부분에 미치는 영향을 최소화하면서 쉽게 되돌릴 수 있도록 하는 것이 중요합니다. 공통 옵션 git add <file> 다음 커밋을 위해 <file>의 모든 변경 사항을 준비하십시오. git add <directory> 다음 커밋을 위해 <directory>의 모든 변경 사항을 스테이지하십시오. git add -p 다음 커밋에 추가 할 파일의 부분을 선택할 수있는 대화 형 스테이징 세션을 시작하십시오. 이렇게하면 변경 사항이 표시되고 명령을 묻는 메시지가 나타납니다. 청크를 스테이지하려면 y를, 청크를 무시하려면, 작은 청크로 분할하고, 청크를 수동으로 편집하고, q를 종료하십시오. Examples 새 프로젝트를 시작할 때 git add는 svn import와 동일한 기능을 제공합니다. 현재 디렉토리의 초기 확약을 작성하려면 다음 두 명령을 사용하십시오. git add . git commit 프로젝트를 시작하고 실행하면 git add에 경로를 전달하여 새 파일을 추가 할 수 있습니다. git add hello.py git commit 위의 명령을 사용하여 기존 파일의 변경 사항을 기록 할 수도 있습니다. 다시 한번 말하지만, Git은 새로운 파일의 준비 변경과 이미 저장소에 추가 된 파일의 변경을 구분하지 않습니다. Git 가이드

2019-06-16 06:14:06.0

Git Log

git log 명령은 커밋 된 스냅 샷을 표시합니다. 프로젝트 기록을 나열하고, 필터링하고, 특정 변경 사항을 검색 할 수 있습니다. git status를 사용하면 작업 디렉토리와 스테이징 영역을 검사 할 수 있지만 git log는 커밋 된 히스토리에서만 작동합니다. 18. png 로그 출력은 커밋을 필터링하는 것에서부터 완전히 사용자 정의 된 형식으로 표시하는 것까지 여러 가지 방법으로 사용자 정의 할 수 있습니다. git log의 가장 일반적인 설정 중 일부는 다음과 같습니다. 사용법 git log 기본 포맷을 사용하여 전체 커밋 기록을 표시하십시오. 출력이 둘 이상의 화면을 차지하는 경우, 스페이스를 사용하여 스크롤하고 q를 사용하여 종료 할 수 있습니다. git log -n <limit> 커밋 수를 <limit>만큼 제한하십시오. 예를 들어, git log -n 3은 3 개의 커밋 만 표시합니다. git log --oneline 각 커밋을 한 줄로 압축하십시오. 이것은 프로젝트 히스토리에 대한 높은 수준의 개요를 얻는 데 유용합니다. git log --stat 일반 자식 로그 정보와 함께 변경된 파일과 각 파일에 추가 또는 삭제 된 행의 상대 번호가 포함됩니다. git log -p 각 커밋을 나타내는 패치를 표시합니다. 이것은 각 커밋의 전체 diff를 보여줍니다. 이것은 프로젝트 히스토리에 대해 가장 자세히 볼 수 있습니다. git log --author="<pattern>" 특정 작성자가 커밋을 검색합니다. <pattern> 인수는 일반 문자열이나 정규 표현식이 될 수 있습니다. git log --grep="<pattern>" <pattern>과 일치하는 커밋 메시지로 커밋을 검색합니다.이 커밋 메시지는 일반 문자열이나 정규 표현식이 될 수 있습니다. git log <since>..<until> <since>와 <until> 사이에 발생하는 커밋 만 표시합니다. 두 인수 모두 커밋 ID, 분기 이름, HEAD 또는 다른 종류의 수정 참조일 수 있습니다. git log <file> 지정된 파일을 포함하는 커밋 만 표시하십시오. 이것은 특정 파일의 기록을 쉽게 볼 수 있는 방법입니다. git log --graph --decorate --oneline 몇 가지 유용한 옵션을 고려해야 합니다. 커밋 메시지의 왼쪽에 커밋의 텍스트 기반 그래프를 그리는 --graph 플래그. --decorate는 표시된 커밋의 브랜치 또는 태그 이름을 추가합니다. --oneline은 커밋 정보를 한 줄에 표시하여 커밋을 한 눈에 쉽게 탐색 할 수 있게 합니다. Git 가이드