**1-bob: Git bilan ishlashni boshlash**

Versiya Ishlab chiqarilish sanasi

2.13 2017-05-10

2.12 2017-02-24

2.11.1 2017-02-02

2.11 2016-11-29

2.10.2 2016-10-28

2.10 2016-09-02

2.9 2016-06-13

2.8 2016-03-28

2.7 2015-10-04

2.6 2015-09-28

2.5 2015-07-27

2.4 2015-04-30

2.3 2015-02-05

2.2 2014-11-26

2.1 2014-08-16

2.0 2014-05-28

1.9 2014-02-14

1.8.3 2013-05-24

1.8 2012-10-21

1.7.10 2012-04-06

1.7 2010-02-13

1.6.5 2009-10-10

1.6.3 2009-05-07

1.6 2008-08-17

1.5.3 2007-09-02

1.5 2007-02-14

1.4 2006-06-10

1.3 2006-04-18

1.2 2006-02-12

1.1 2006-01-08

1.0 2005-12-21

0.99 2005-07-11

**1.1-bo'lim: Birinchi omboringizni yarating, so'ngra fayllarni qo'shing va topshiring**

Buyruqlar satrida avval Git o'rnatilganligini tekshiring:

Barcha operatsion tizimlarda:

git --version

UNIX-ga o'xshash operatsion tizimlarda:

which git

Hech narsa qaytarilmasa yoki buyruq tan olinmasa, o'rnatuvchini yuklab olish va ishga tushirish orqali tizimingizga Gitni o'rnatishingiz kerak bo'lishi mumkin. O'rnatish uchun juda aniq va oson ko'rsatmalar uchun Git bosh sahifasiga(https://git-scm.com/book/en/v2/Getting-Started-Installing-Git) qarang.

Git-ni o'rnatganingizdan so'ng, foydalanuvchi nomingiz va elektron pochta manzilingizni sozlang. Qaror qabul qilishdan *oldin* buni bajaring.

Git o'rnatilgandan so'ng, versiya nazorati ostida joylashtirmoqchi bo'lgan katalogga o'ting va bo'sh Git omborini yarating:

git init

Bu Git ishlashi uchun zarur bo'lgan sanitariya-tesisatni o'z ichiga olgan yashirin .git papkasini yaratadi.

Keyin, Git yangi omboringizga qanday fayllar qo'shishini tekshiring; Ushbu bosqich alohida e'tiborga loyiqdir:

git status

Olingan fayllar ro'yxatini ko'rib chiqing; Git-ga qaysi fayllarni versiya boshqaruviga joylashtirishni aytishingiz mumkin (parollar kabi maxfiy ma'lumotlarga ega fayllarni yoki omborni bezovta qiladigan fayllarni qo'shmang):

git add <file/directory name #1> <file/directory name #2> < ... >

Agar ro'yxatdagi barcha fayllar omborga kirish huquqiga ega bo'lgan har bir element bilan bo'lishish kerak bo'lsa, bitta buyruq sizning joriy katalogingiz va uning pastki kataloglaridagi hamma narsani qo'shadi:

Git add .

Bu versiya boshqaruviga qo'shiladigan barcha fayllarni "bosqichga" qo'yadi va ularni birinchi majburiyatingizda bajarilishiga tayyorlaydi.

Hech qachon versiya nazorati ostida boʻlishini istamaydigan fayllar uchun qoʻshish buyrugʻini ishga tushirishdan oldin “.gitignore” nomli faylni yarating va toʻldiring.

Qo'shilgan barcha fayllarni ‘commit message’ bilan birga ‘commit’ qiling:

git commit -m "Initial commit"

Bu ‘commit message’ bilan yangi ‘commit’ yaratadi. ‘commit’ butun loyihangizni saqlash yoki suratga olish kabidir. Endi siz uni ‘remote repository’("masofaviy ombor")- ga ‘push’ qilishingiz yoki yuklashingiz mumkin, agar kerak bo'lsa, keyinroq unga qaytishingiz mumkin. Agar siz -m parametrini o'tkazib yuborsangiz, standart muharriringiz ochiladi va siz u erda ‘commit message’-ni tahrirlashingiz va saqlashingiz mumkin.

Masofaviy omborni qo'shish

Yangi ‘masofaviy ombor’ qo'shish uchun terminalda git remote add buyrug'idan foydalaning, sizning omboringiz saqlanadigan katalogda.

git remote add buyrug'i ikkita argumentni oladi:

1. ‘remote’ nomi, masalan, origin
2. ‘remote’ URL manzili, masalan, https://<your-git-service-address>/user/repo.git

git remote add origin https://<your-git-service-address>/owner/repository.git

DIQQAT: ‘remote’-ni qo'shishdan oldin siz ‘git xizmati’-da kerakli omborni yaratishingiz kerak, ‘remote’-ni qo'shgandan so'ng siz ‘commit’-larni ‘push’/’pop’ qilish imkoniyatiga ega bo'lasiz.

**1.2-bo'lim: ‘Repozitariy’(‘Ombor’)-ni klonlash**

git clone buyrug'i mavjud Git omborini serverdan mahalliy mashinaga nusxalash uchun ishlatiladi.

Masalan, ‘GitHub’-dagi loyihani klonlash uchun:

cd <klonning katalog yaratishini xohlagan yo'l>

git clone https://github.com/username/projectname.git

‘BitBucket’-dagi loyihani klonlash uchun:

cd <klonning katalog yaratishini xohlagan yo'l>

git clone https://yourusername@bitbucket.org/username/projectname.git

Bu ‘masofaviy Git ombori’-dagi barcha fayllarni o'z ichiga olgan ‘mahalliy mashina’-da projectname deb nomlangan katalog yaratadi. Bunga loyiha uchun ‘manba fayllari’, shuningdek, loyihaning butun tarixi va konfiguratsiyasini o'z ichiga olgan ‘.git’ ‘pastki katalog’-i kiradi.

Katalogning boshqa nomini belgilash uchun, masalan. MyFolder:

git clone https://github.com/username/projectname.git MyFolder

Yoki joriy katalogda klonlash uchun:

git clone https://github.com/username/projectname.git .

Eslatma:

1. Belgilangan katalogga klonlashda katalog bo'sh yoki mavjud bo'lmasligi kerak.
2. Buyruqning ssh versiyasidan ham foydalanishingiz mumkin:

git clone git@github.com:username/projectname.git

‘https’ versiyasi va ‘ssh’ versiyasi ekvivalentdir. Biroq, GitHub kabi ba'zi xosting xizmatlari ‘ssh’ o'rniga ‘https’-dan foydalanishni tavsiya qiladi.

**1.3-bo'lim: Kodni almashish**

Kodingizni almashish uchun siz ‘mahalliy omboringiz’-dan nusxa ko'chiradigan ‘masofaviy server’-da ombor yaratasiz.

‘Masofaviy server’-da bo'sh joydan foydalanishni minimallashtirish uchun siz ‘yalang'och ombor’ni yaratasiz: unda faqat .git ob'ektlari mavjud va fayl tizimida ishlaydigan nusxa yaratmaydi. Bonus sifatida siz ushbu ‘masofaviy ombor’-ni boshqa dasturchilar bilan yangilanishlarni osongina almashish uchun ‘yuqori oqim serveri’ sifatida o'rnatasiz.

Masofaviy serverda:

git init --bare /path/to/repo.git

Mahalliy mashinada:

git remote add origin ssh://username@server:/path/to/repo.git

(E'tibor bering, ssh: masofaviy omborga kirishning mumkin bo'lgan usullaridan biri.)

Endi ‘mahalliy omboringiz’-ni ‘masofaviy omboringiz’-ga nusxalash:

git push --set-upstream origin master

yoki

git push -u origin main

--set-upstream (yoki -u) qo'shilishi argumentsiz Git buyruqlari tomonidan ishlatiladigan yuqori oqim (kuzatuv) mos yozuvini yaratdi, masalan, git pull.

**1.4-bo'lim: Foydalanuvchi nomi va elektron pochtangizni o'rnatish**

Har qanday ‘topshirish’ yaratishdan oldin kimligingizni belgilashingiz kerak. Bu ularga tegishli ‘foydalanuvchi nomi’ va ‘elektron pochta manzili’-ga ega bo'lish imkonini beradi.

Masofaviy omborga o'tishda (masalan, GitHub, BitBucket yoki GitLab ‘hisob qaydnoma’-ngiz yordamida masofaviy omborga o'tishda) buning ‘autentifikatsiya’ bilan hech qanday aloqasi yo'q.

Ushbu identifikatsiyani *barcha* omborlar uchun e'lon qilish uchun git config --global dan foydalaning. Bu sozlamani foydalanuvchining ".gitconfig" faylida saqlaydi: masalan, "$HOME/.gitconfig" yoki Windows uchun "%USERPROFILE%\.gitconfig" .

git config --global user.name "Sizning nomingiz"

git config --global user.email [mail@example.com](mailto:mail@example.com)

Bitta ombor uchun identifikatorni e'lon qilish uchun "katalog" ichida "git config" dan foydalaning.

Bu sozlamani alohida ombor ichida, ‘$GIT\_DIR/config’ faylida saqlaydi. Masalan, "/path/to/your/repo/.git/config".

cd /path/to/my/repo

git config user.name "Your Login At Work"

git config user.email mail\_at\_work@example.com

Repozitariy konfiguratsiya faylida saqlangan sozlamalar ushbu ombordan foydalanganda global konfiguratsiyadan ustun bo'ladi.

Maslahatlar: agar sizda turli xil ‘identifikator’-lar bo'lsa (biri ochiq manbali loyiha uchun, biri ish joyida, biri shaxsiy reposlar uchun, ...) va siz ustida ishlayotgan har bir turli repos uchun to'g'risini o'rnatishni unutishni xohlamasangiz. :

* **Global identifikatsiyani olib tashlang**

git config --global --remove-section user.name

git config --global --remove-section user.email

Version ≥ 2.8

* Gitni identifikatoringizni global konfiguratsiyada emas, faqat ombor sozlamalarida qidirishga majburlash uchun:

git config --global user.useConfigOnly true

Shunday qilib, agar siz ma'lum bir ombor uchun user.name va user.email-ni o'rnatishni unutib qo'ysangiz va ‘commit’-ni bajarishga harakat qilsangiz, quyidagilarni ko'rasiz:

no name was given and auto-detection is disabled

no email was given and auto-detection is disabled

**1.5-bo'lim: ‘The upstream remote’-ni sozlash**

Agar siz ‘vilka’ klonlagan bo'lsangiz (masalan, Github-dagi ochiq kodli loyiha) sizda yuqori oqim omboriga ‘push’ ruxsati bo'lmasligi mumkin, shuning uchun sizga ikkala ‘vilka’ kerak, lekin yuqori oqim omborini olishingiz mumkin.

Avval ‘masofaviy omborlar’ nomlarini tekshiring:

git remote -v

origin https://github.com/ilmfan/git.git (fetch)

origin https://github.com/ilmfan/git.git (push)

upstream # bu chiziq bu erda bo'lishi mumkin yoki bo'lmasligi mumkin

Agar yuqori oqim allaqachon mavjud bo'lsa (u ba'zi Git versiyalarida mavjud) URL manzilini o'rnatishingiz kerak (hozirda u bo'sh):

git remote set-url upstream https://github.com/projectusername/repo.git

Agar yuqori oqim bo'lmasa yoki siz ham do'stingiz/hamkasbingizning vilkasini qo'shmoqchi bo'lsangiz (hozirda ular mavjud emas):

git remote add upstream https://github.com/projectusername/repo.git

git remote add dave https://github.com/dave/repo.git