**Úrbótarverkefni**

Git er frí útgáfa af Version Control System, skammstafað VCS.VCS getur virkað sem einhverskoðar ættartré fyrir kóðan þinn, þú getur séð hvaðan hann kom og frá hverjum, hægt er að búa til nýjar “greinar” eða branches til að útfæra upprunalegu greinina en getur alltaf farið til baka í ástandið sem hann var í.

GitHub er vefsíða/forrit þar sem þú getur deilt þínu Git vefsæði eða geymslu, kallað repository, með öðru fólki og valið hverjir fá aðgang.

Auðvelt er að læra á Git og ekki þarf að læra margar skipanir til að ná góðum tökum á því.

$mkdir verk

$cd verk

Notandinn býr til möppuna verk og færir sig inn í hana.

/verk $ mkdir gogn

/verk $ mkdir printf ‘Eitthvad’ > gogn/file.txt

Notandinn býr til möppuna gögn og býr til file.txt inn í þeirri möppu og í þeim file stendur “Eitthvad”.

Git init skipunin breytir möppunni í Git Repository. Skipunin gerir .git möppu með skrám sem hægt er að breyta og skoða söguna um verkefnið.

/verk $ git add gogn/file.txt.

Þegar notandinn keyrir þessa skipun þá hefur það tvenns konar áhrif. Það annarsvegar býr til nýja blob skrá í möppunni gogn/.git/objects/ og sú skrá inniheldur samanþjappað innihald af gogn/file.txt skránni. Nafnið af skránni er fengið með að hasha innihald skráarinnar breytir það nafninu í eitthvað sem einkennir skránna, Git t.d hashar Eitthvad í de20247992af0f949ae8df4fa9a37e4a03d7063e.

Þegar notandi bætir við skrá í git repositoryið geymir það innihald þess í objects möppunni. Ef að notandi eyðir t.d gogn/file.txt þá væri innihaldið enþá vistað í git.

Í öðru lagi þá bætir git add skránni til index. Index er listi sem inniheldur hverja einustu skrá sem er búið að byðja git um að halda utan um og sá listi er í verk/.git/index. Hver lína inniheldur skrá og hash að innihaldinu þegar henni var bætt við.

Þegar notandi bætir við textaskrá sem heitir t.d upplysingar.txt í gogn möppunni og gerir svo git add gogn ( það bætir bara við skrám sem eru inn í gogn möppunni ) þá bætir það við blob skrá sem inniheldur innihald skráarinnar upplysingar.txt og bætir við nýju index línu í index skránna sem tengir skránna við hash af innihaldi þess. Ef að það er breytt skrá og addar því þá er gerður ný blob skrá með innihaldinu og uppfærir index línuna svo að hún bendi á hash af innihaldinu á nýju skránni.

Commit

Git commit býr til trjáar graf sem táknar innihald af útgáfunni sem er verið að committa, býr svo til commit objectið og bendir þáverandi branchi að nýja committinu.

Tree graph

Git býr til trjáar graf af núverandi ástandi á projectinu frá indexinu. Þetta graf geymir staðsetningu og innihald á hverjum einasta file í projectinu og er samansett af blobs og trees. Blobs einkenn innihald skránna og eru geymd með git add. Þegar notandi committar er tré vistað og það táknar möppur á vinnusvæðinu. Það er ein lína fyrir hvern hlut í projectinu og getur bæði verið skrá eða önnur mappa, index geymir bara file en ekki möppur sem þýðir það að ef það er tóm mappa þá kemur hún ekki fram í grafinu. Hver lína er búin til af fjórum hlutum sem eru nauðsynlegir til að endurgera viðkomandi hlut í projectinu. 1. Item permission, 2. Tegund af objecti, hvort það sé blob eða tree, 3. Hashið af viðkomandi hlut, 4. Nafn hlutsins

t.d 10082 blob 5e40c0877058c504203932e5136051 texti.txt

Github er mikið notað meðal web developera þar sem það er svo handhægt að allir nota sama vinnusvæði án þess að skemma neitt. Ef að eitthvað er uppfært er alltaf hægt að fara yfir í gamla útgáfu af viðkomandi skjali til þess að “restorea” það. Það er líka handhægt þar sem að margir notendur geta unnið á sama stað og í hvert sinn sem að skjal er uppfært þá kemur það fram hvaða notandi uppfærði skjalið, hvað það var sem notandinn gerði og dagsetning á því þegar skjalinu var breytt.

Github gefur git meira líf og er ekki bara einangrað í command-line eins og git er. Það er bæði til vefsíða og desktop forrit fyrir github sem er þægilegt og auðvelt í notkun um leið og maður hefur lært grunnatriðin á Github.

Git leyfir þér að hafa margar branches og þær eru allar alveg óháðar hvor annari sem gerir þér kleyft að búa til nýtt branch, commitað og prufað nýja hugmynd og fært þig aftur þar sem þú byrjaðir. Það sem einkennir git frá öðrum version control systems er það að þú committað sumar af skránnum þínum án þess að uppfæra allar sem þú hefur breytt á vinnusvæðinu sem gerir þér kleyft að committa aðeins hluta af skrá sem þú hefur breytt. Byrjað er að adda frá working directory yfir í staging area og þaðan committað inn í repositoryið. Þetta gerir Git allt á skotstundu.

Clone

Ef þú vilt afrita núverandi Git Repository til dæmis verkefni sem þú vilt breyta þá er notað git clone. Git fær copy af nánast öllum gögnum sem serverinn hefur. Allar útgáfur af öllum skrám í sögu verkefnisins og jafnvel ef serverinn krassar þá er hægt að nota cloneana til að endurræsa servernum á sama ástand og hann var í þegar cloneinn var gerður. Git clone <https://github.com/verk2/verk2> býr til möppuna verk2, býr til .git möppu inn í henni og tekur afrit af öllum gögnum úr því repository og hægt er að fara strax að vinna í afritinu. Ef þarf að skýra verkefnið öðru nafni er hægt að skilgreina það með því að láta nýja nafnið fylgja eftir slóðinni með bili.

Hver skrá í verkefninu getur verið í einu af tveim ástöndum, tracked eða untracked. Tracked skrár eru skrár sem voru í seinasta snapshotti, þau geta verið unmodified, modified eða staged. Untracked skrár eru allt annað, allar skrár í verkefninu sem voru ekki í seinasta snapshotti og eru ekki í staging area. Þegar þú clonear repository verða allar skrár tracked og unmodified vegna þess að Git hefur athugað á þeim og notandinn hefur engu breytt. Þegar þú hefur breytt skrá sér Gitþau sem modified þar sem þú hefur breytt þeim eftir síðasta commit þá stagear maður modified skrárnar og committar allar staged breytingar.

Skoða staged og unstaged skrár.

Hægt er að nota git diff til að sjá nákvæmlega hverju var breytt ekki aðeins hvaða skrám var breytt. Þannig er hægt að sjá hverju hefur verið breytt en ekki stageað enþá og hvað hefuru stageað sem þú ert að fara committa. Git status getur svarað þessum spurningum en git diff gefur þér mun ítarlegri útgáfu af því, t.d hvaða línur voru breyttar, eyddar eða bætt við. Git diff sýnir ekki allar breytingar sem hafa verið gerðar síðan seinasta committ, heldur bara breytingar sem eru enþá unstaged. Svo að ef þú hefur stageað allar breytingar þá gefur git diff þér ekkert output.

Committa breytingar.

Þegar þú hefur sett upp stage geturu byrjað að committa. Allar þær skrár sem þú hefur breytt eða búið til en hefur ekki keyrt git add á síðan þeim var breytt fara ekki í næsta commit. Þau munu haldast sem modified skrá. Auðveldasta leiðin til að committa er að nota git commit. Þá kemur upp editor og sýnir þér hvað var breytt bætt við með nafni. Útkoman sem kemur þegar git commit er keyrt geymir síðustu útkomu af git status commandinu. Hægt er að eyða þessum commentum og skrifað sitt eigið commit skilaboð. Commit geymir snapshot sem sett var upp í staging area, allt sem ekki er stageað er enþá í modified. Í hvert sinn sem er committað geymir snapshot af verkefninu sem er hægt að fara aftur í seinna eða bera saman ef þess þarf. Með því að bæta við -a í git commit skipuninni lætur git sjálfkrafa stagea allar skrár sem eru trackaðar áður en committið er framkvæmt sem gerir þér kleyft að sleppa git add hlutanum.

Til að fjarlægja skrá frá Git þarf að fjarlægja hana frá tracked skrám, eða í raun fjarlægja hana úr staging area og svo committa. Git rm gerir þér kleyft að gera það og fjarlægjir líka skrár frá vinnusvæðinu svo að þú sérð þau ekki untracked næst. Ef notandi fjarlægjir skrá kemur hann fram sem unstaged ef notandinn gerir git status. Svo næst þegar þú committar verður skráin farin og ekki lengur tracked en ef skráin hefur verið breytt og núþegar verið sett í index þarf að neyða git að fjarlægja það með -f, þetta er gert til þess að eyða ekki óvart skrá sem er ekki vistuð í snapshot og getur ekki verið endurheimt.