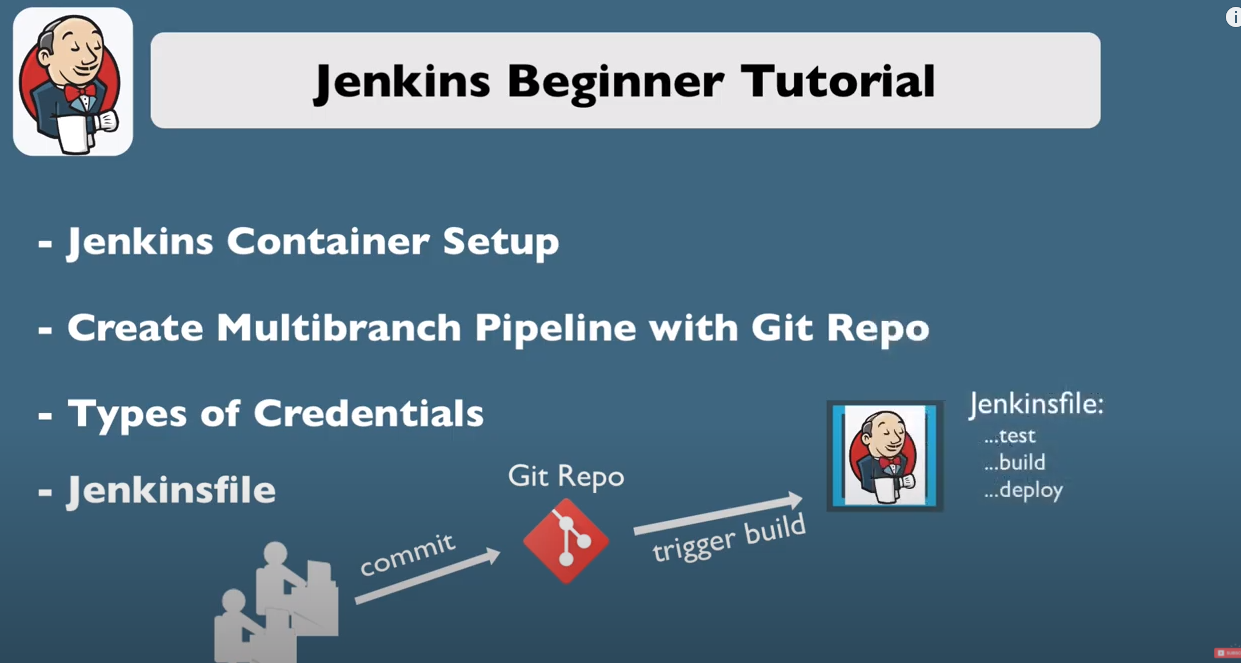
**Jenkins**

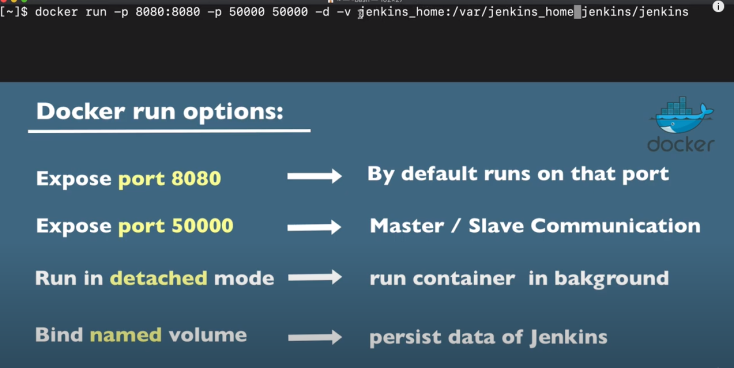
**https://www.youtube.com/watch?v=pMO26j2OUME**



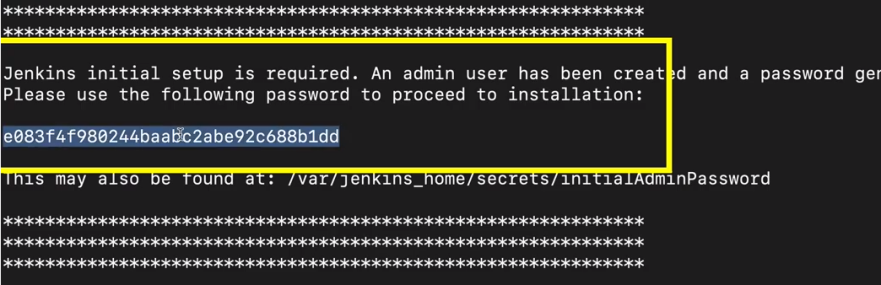
Step 1

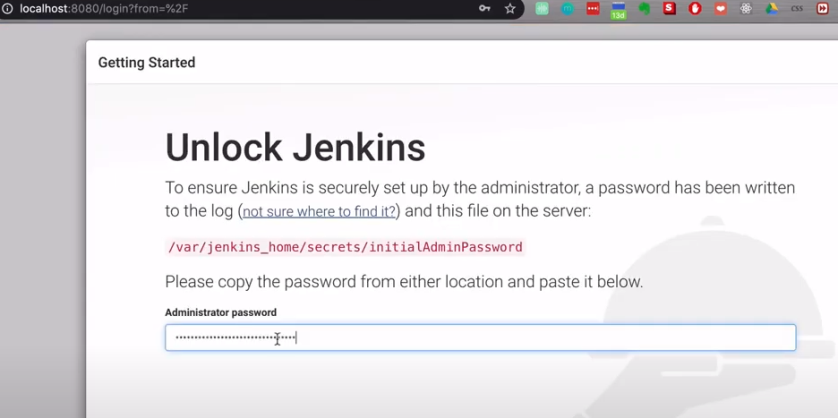


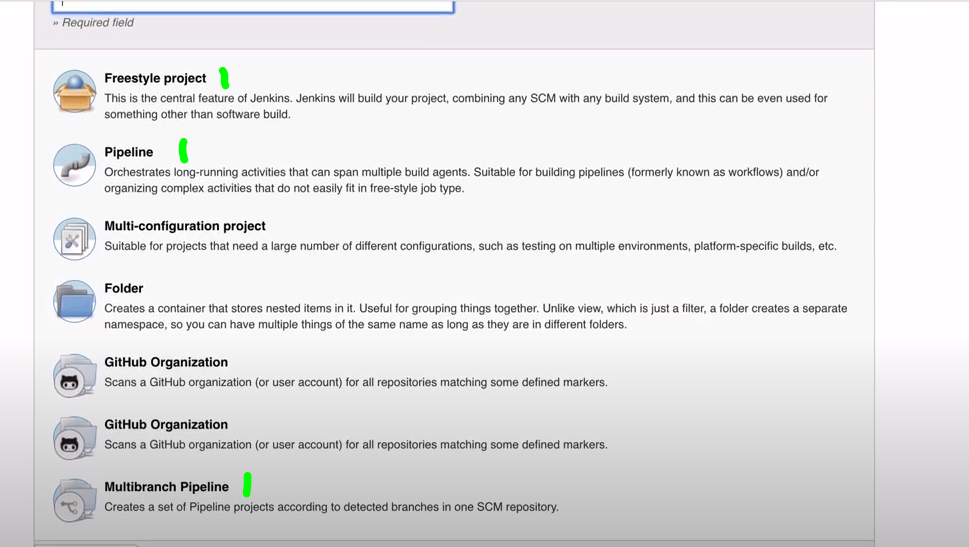


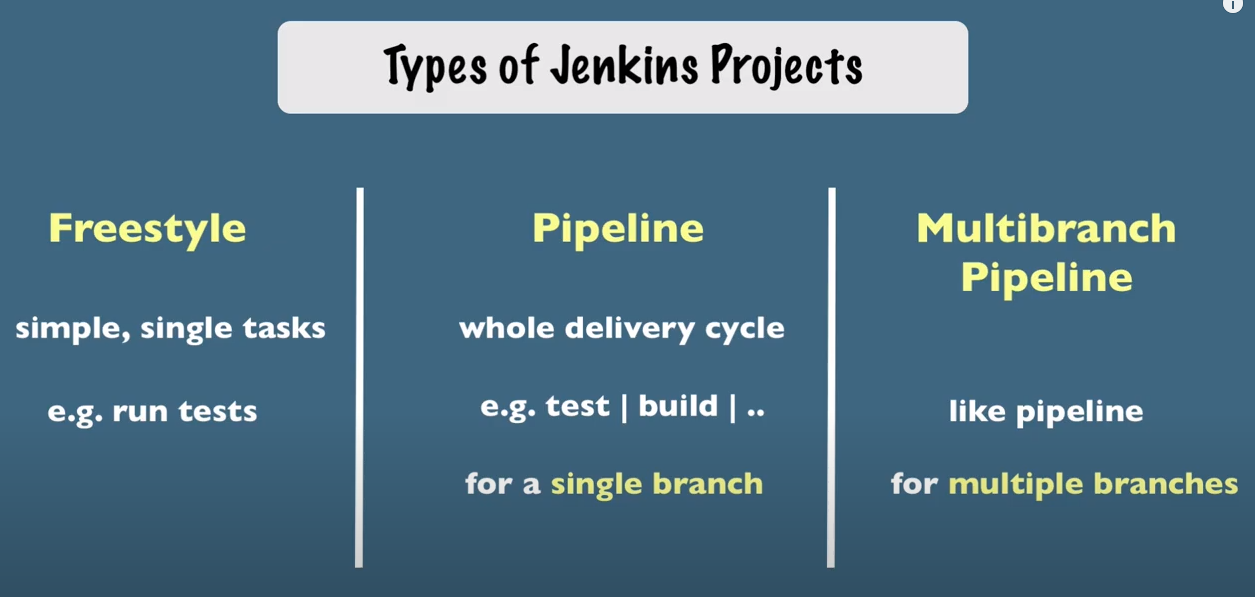


Chiedendo il log ci dice di usare la password per il setup iniziale:



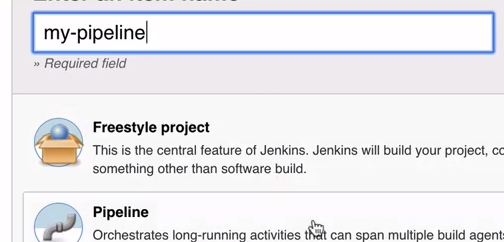


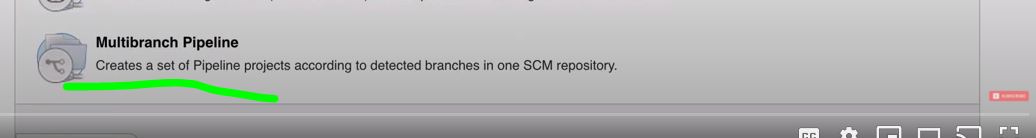


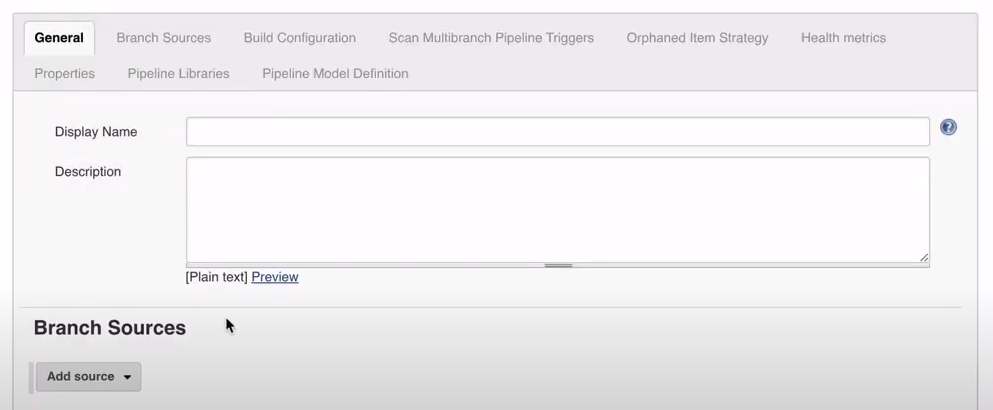


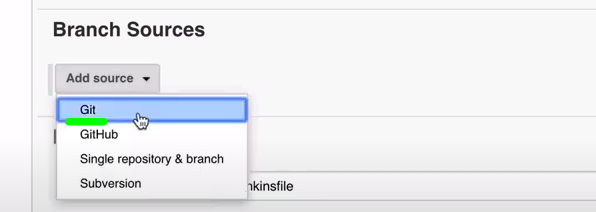
[**https://www.youtube.com/watch?v=tuxO7ZXplRE**](https://www.youtube.com/watch?v=tuxO7ZXplRE)

Ora creiamo il multybranch pipeline

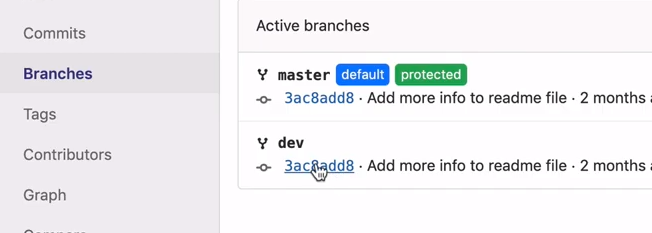




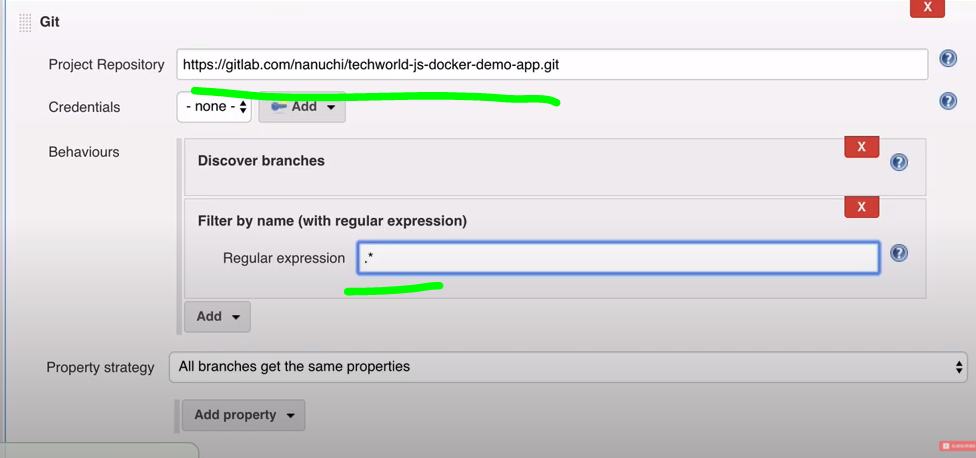




Sul progetto di git abbiamo due branch



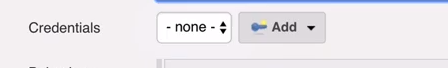
Faccio il clone della directory e la incollo su project repository e poi .\* significa prendere in considerazione e quindi fare la build di tutti i branch:



La build configuration deriva da un file jenkins

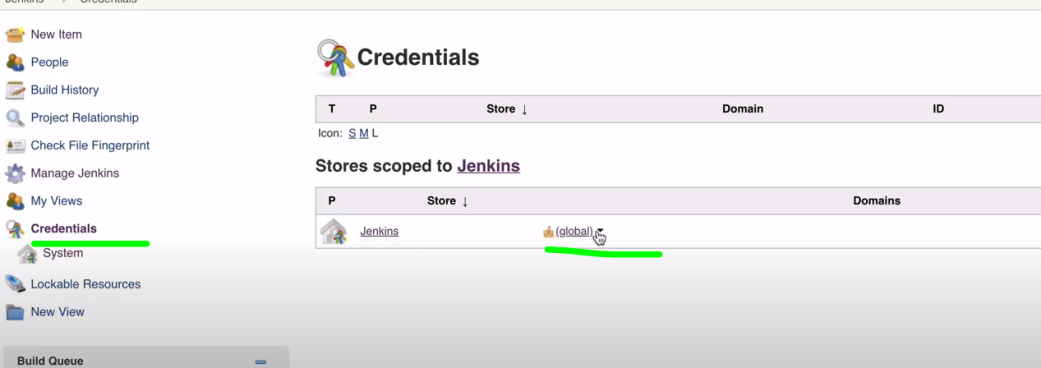


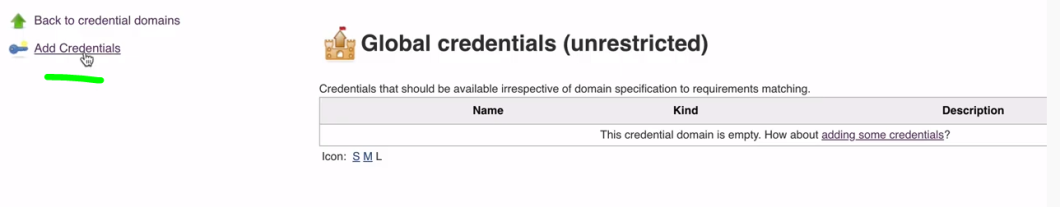
Occorre inserire le credenziali della repository:

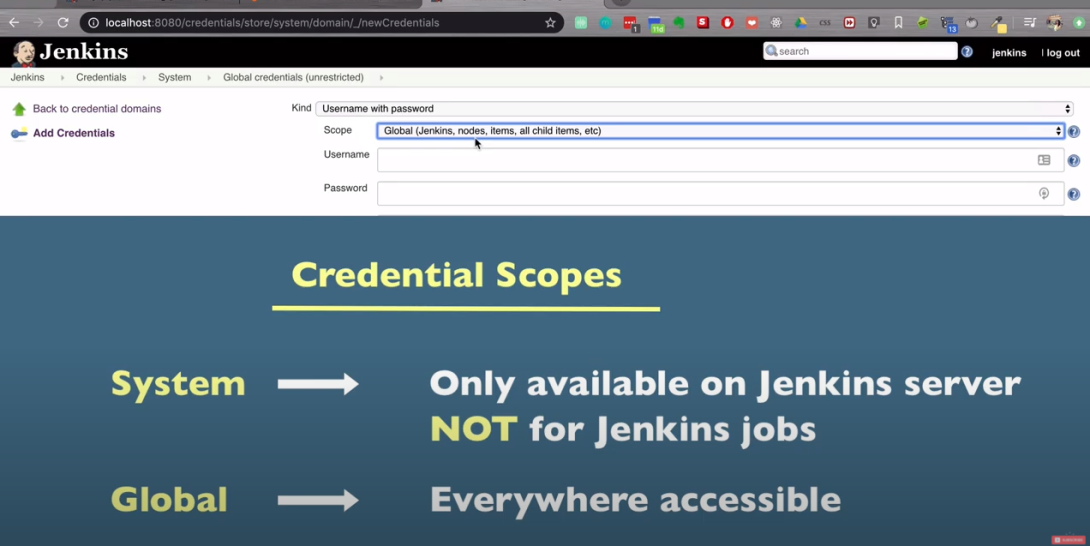


Apriamo una parentesi sulle credenziali in jenkins e come utilizzarle:

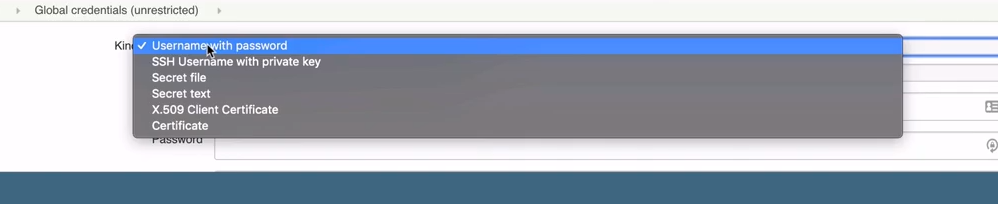
In jenkins abbiamo di default uno scope globale e dentro nessuna credenziale, andiamo ad aggiungerne una.

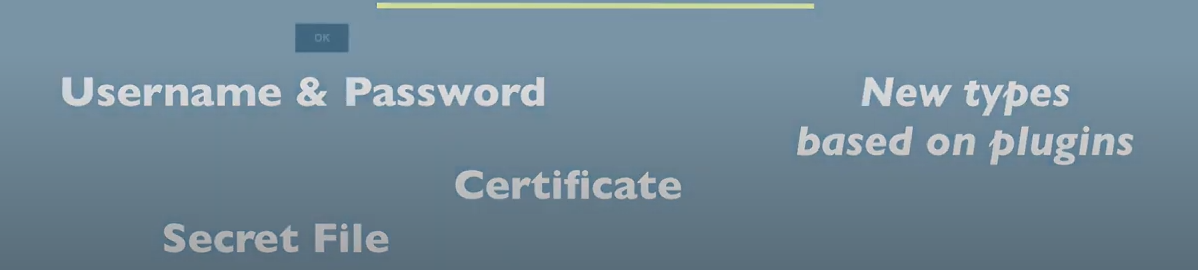




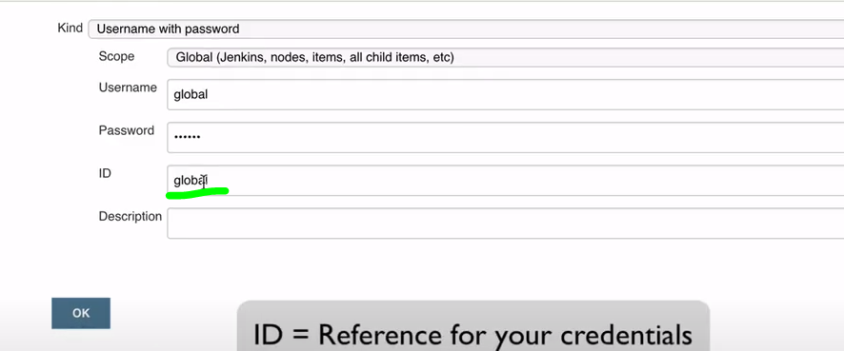


Poi abbiamo il type delle credenziali:

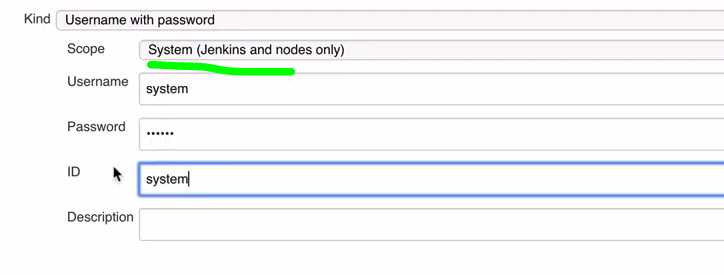




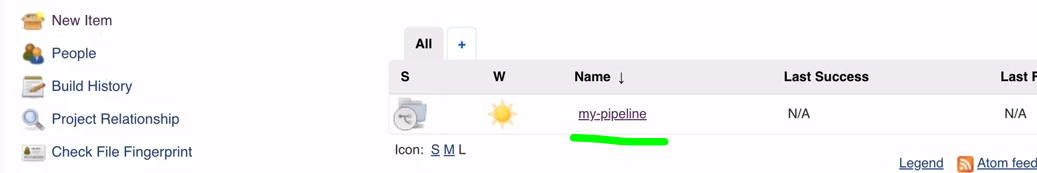
Id è molto importante perché con quello referenziamo le credenziali ovunque compresi i file jenkins:



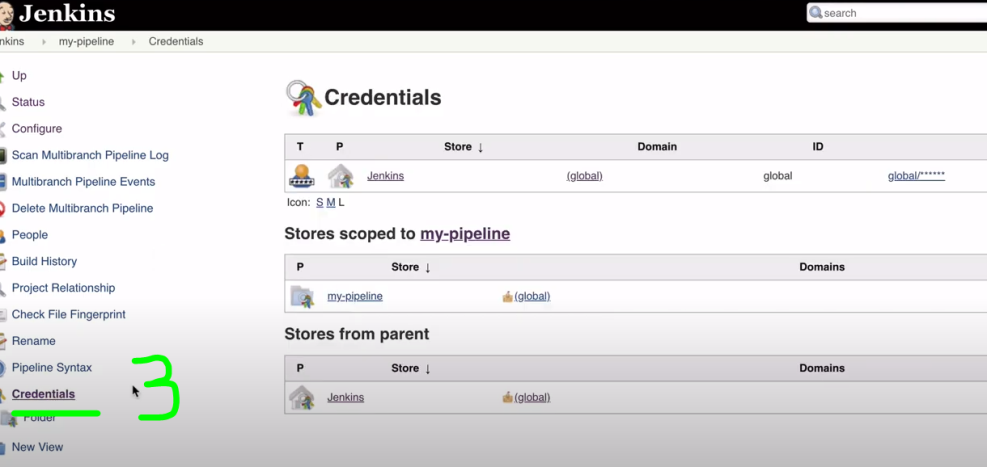
Poi creiamo anche una credenziale di sistema:



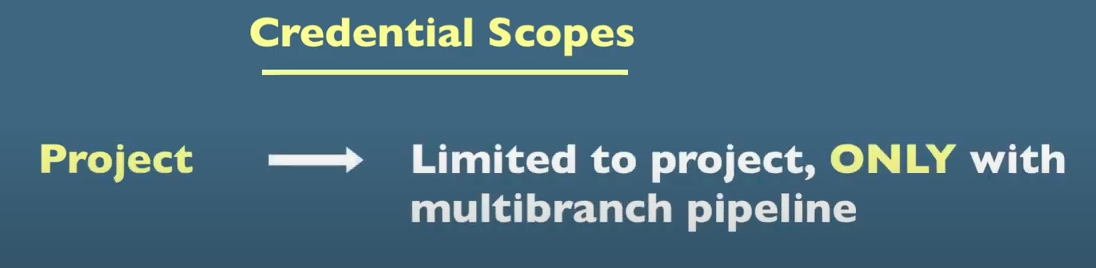
Se ora vado dentro la mia pipeline:



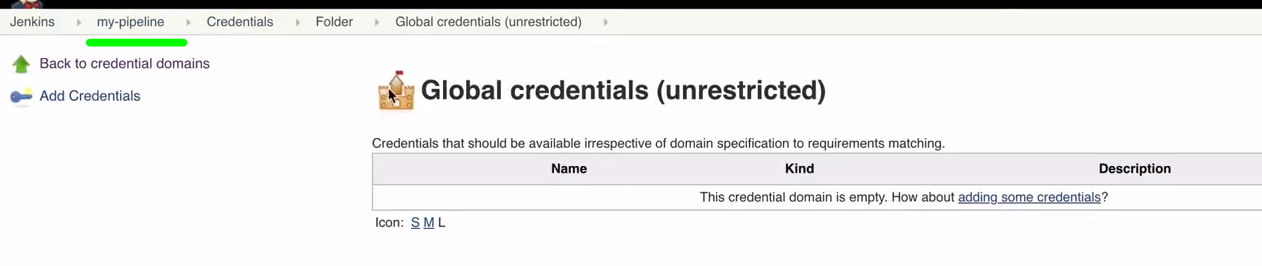
Se entriamo dentro abbiamo un terzo livello di credenziali:



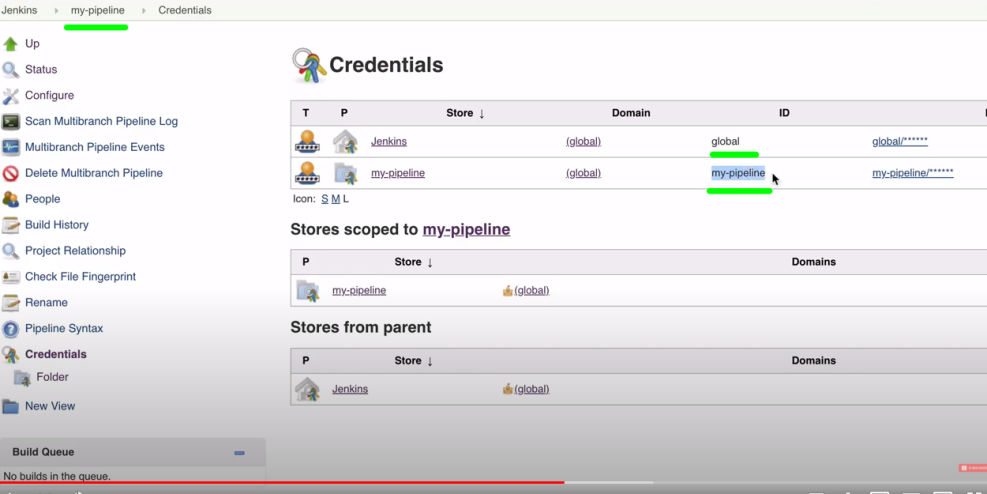
Questo scope è limitato solo al job corrente ed è solo per il multybranch:



Qui ad esempio siamo in jenkis pipeline credential (fatto click su store scoped to my-pipeline:

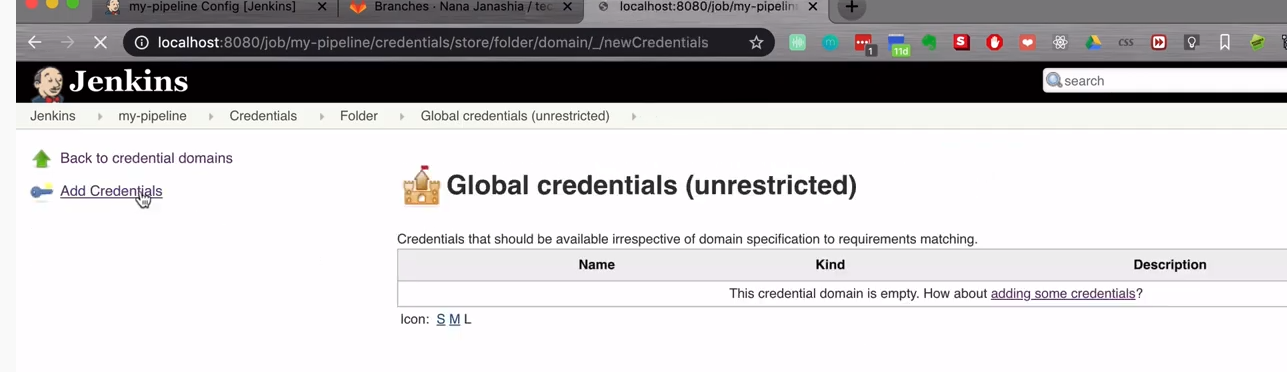


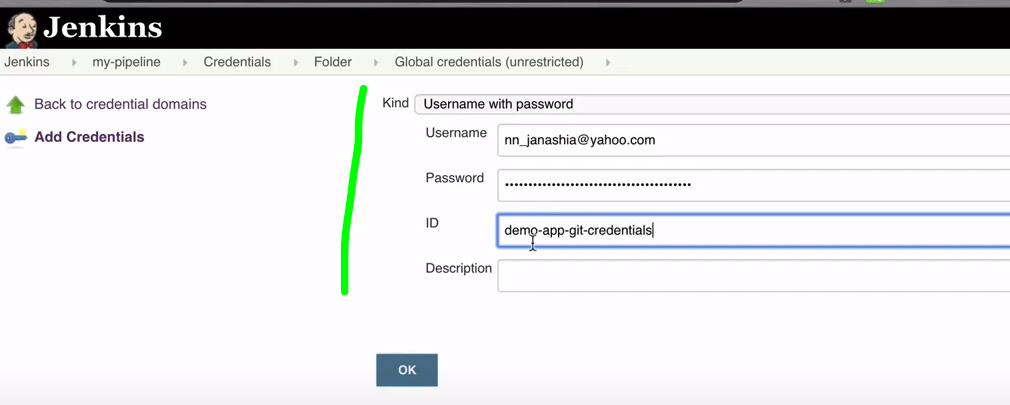
Creiamo una credenziale e dopo averlo fatto vediamo che ora il nostro progetto ha due credenziali, quella creata precedentemente e quella creata ora:

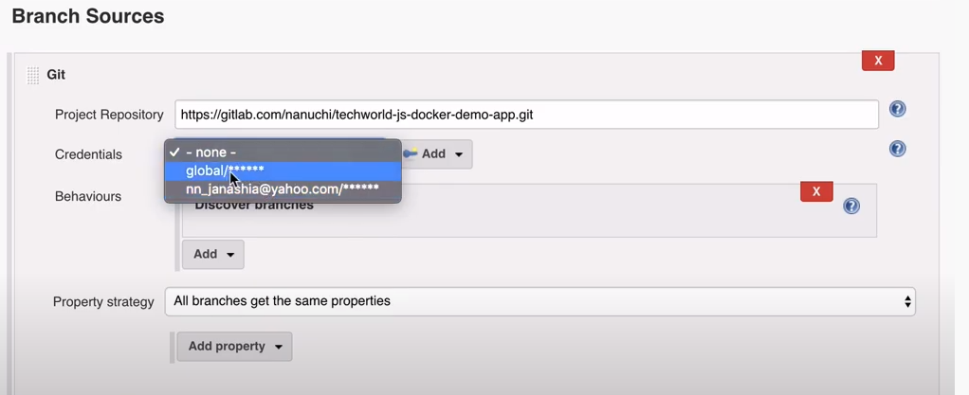


Questa creazione di credenziale a livello di progetto serve quando hai più team che lavorano a parti separate del progetto..

Quindi tornando a prima ora creiamo dentro la pipeline scope delle credenziali con username e pwd con cui accediamo a git:

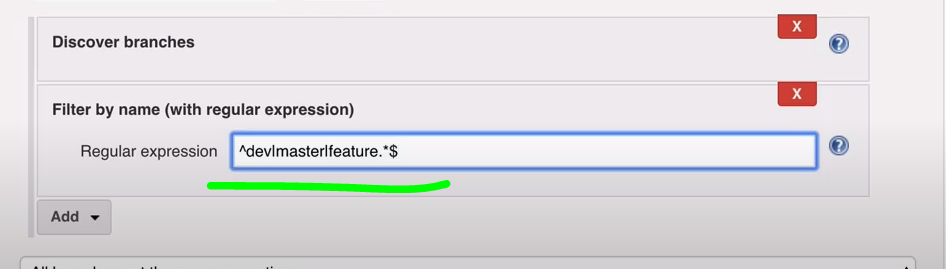






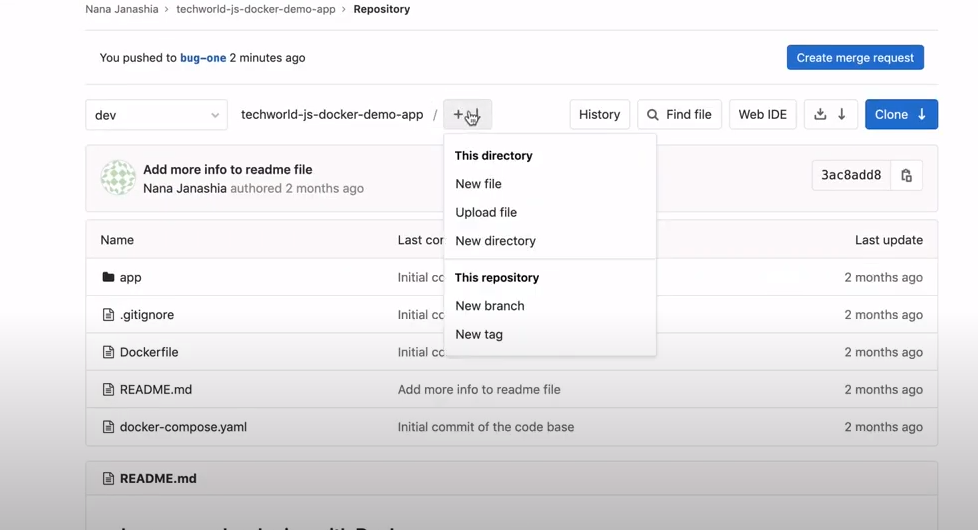
Salvate le credenziali noteremo dal log che si è connesso, ha trovato i branch ma non trova il file jenkins, questo perché dobbiamo crearlo.

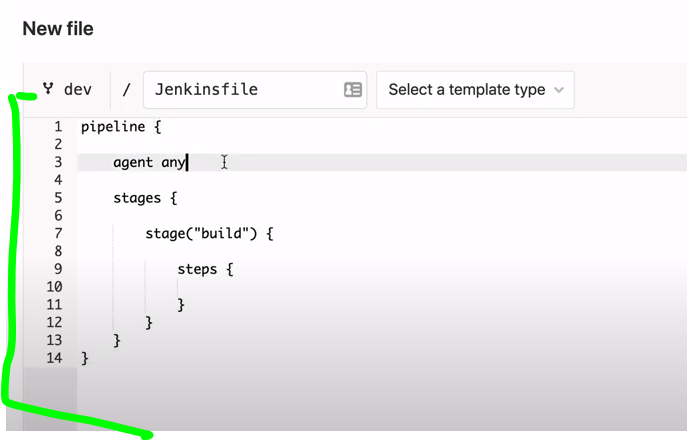
Per prendere in considerazione solo alcuni branch sostituiamo la regular expression generica.



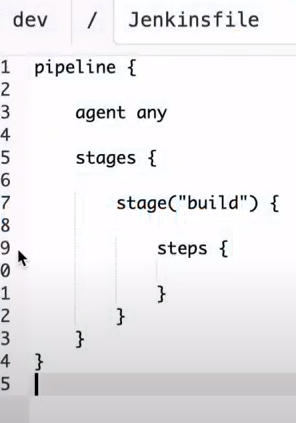
**https://www.youtube.com/watch?v=MY1w7sWW5ms**

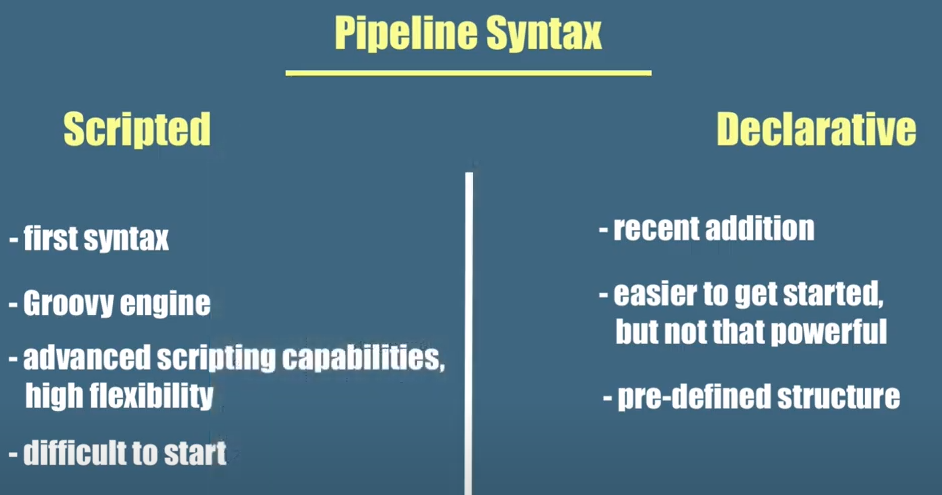
Ora creiamo un jenkins in dev branch

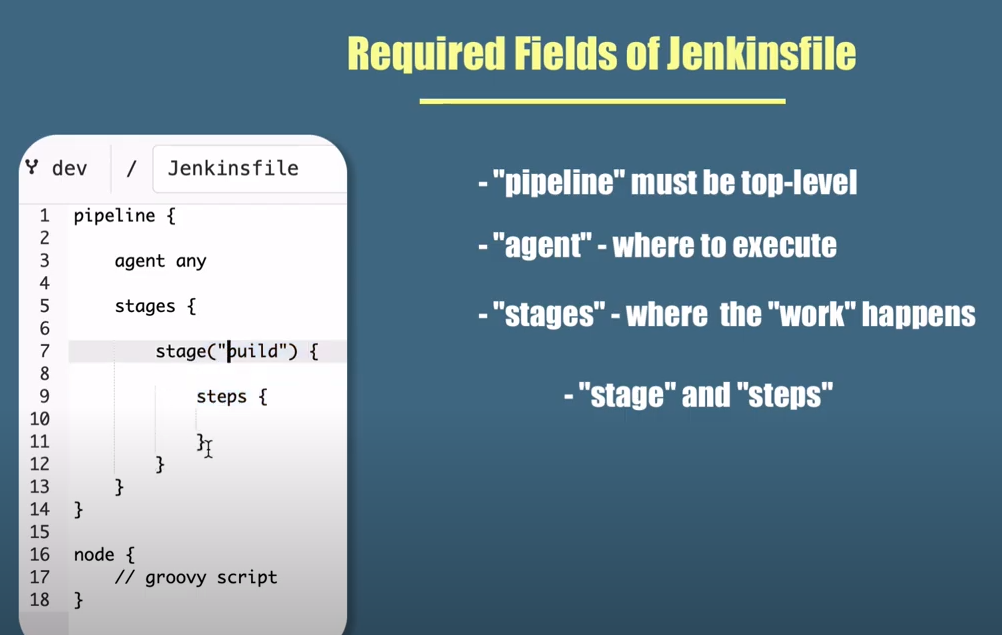
  
questa la sintassi base:



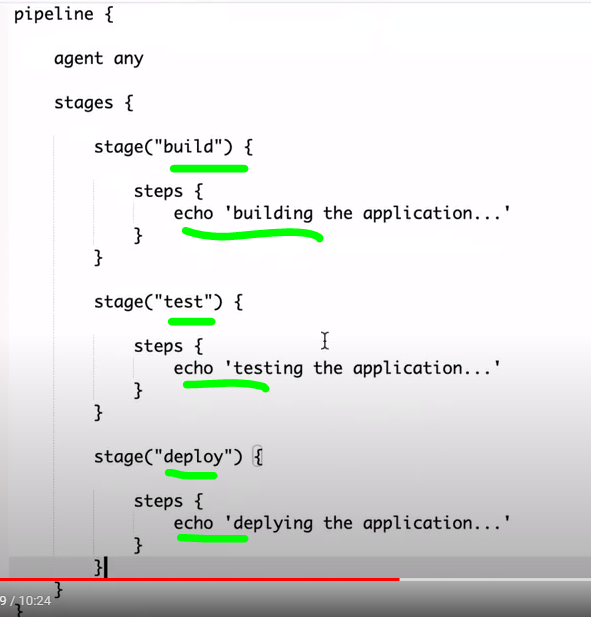
Jankins può utilizzare due sintassi la prima è la più avanzata e flessibile e fa uso di groovy script la seconda più semplice. La seconda è stata aggiunta più di recente ed è un po’ più semplce perché ha una struttura predefinita di partenza:



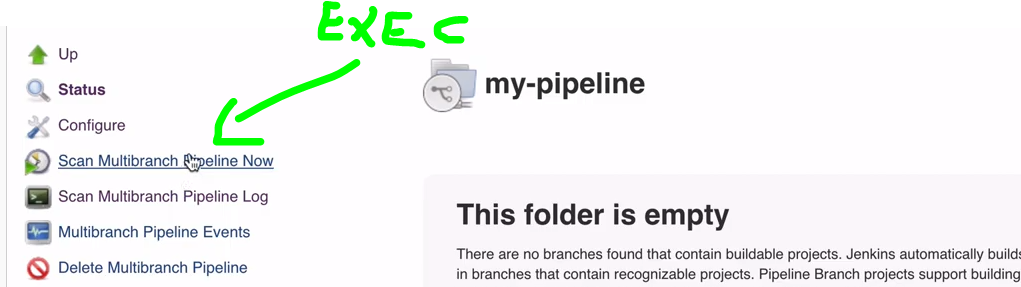




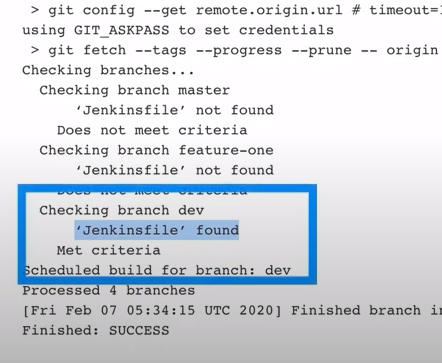
Una volta creato un file jenkins di esempio e mandato in esecuzione vediamo dal log che trova il file e lo processa:



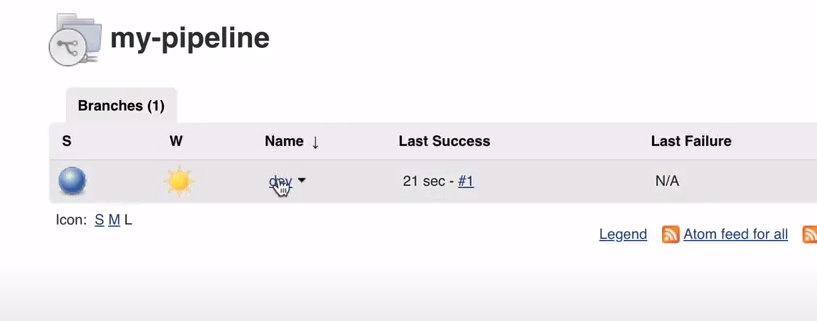
Eseguiamo:



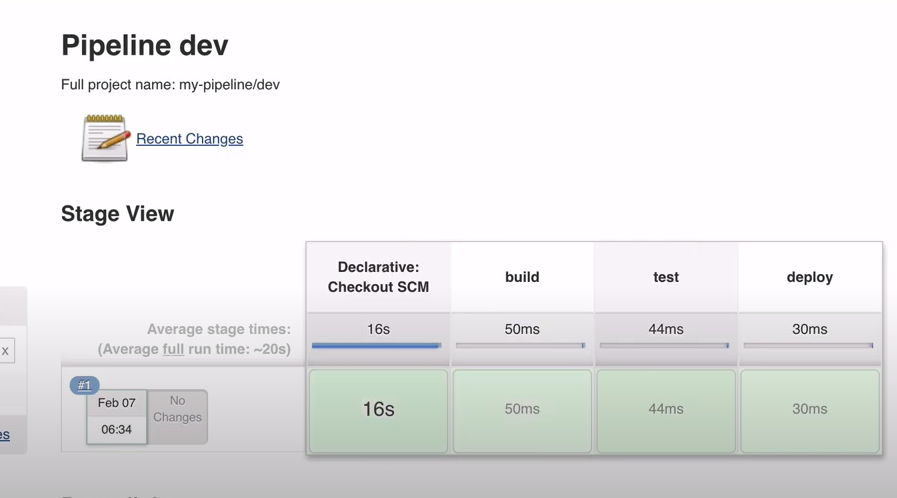
Questo il log



Se andiamo nella pipeline view vediamo che il branch dev è stato compilato:



E se entriamo dentro nel record vediamo l’esecuzione in dettaglio:

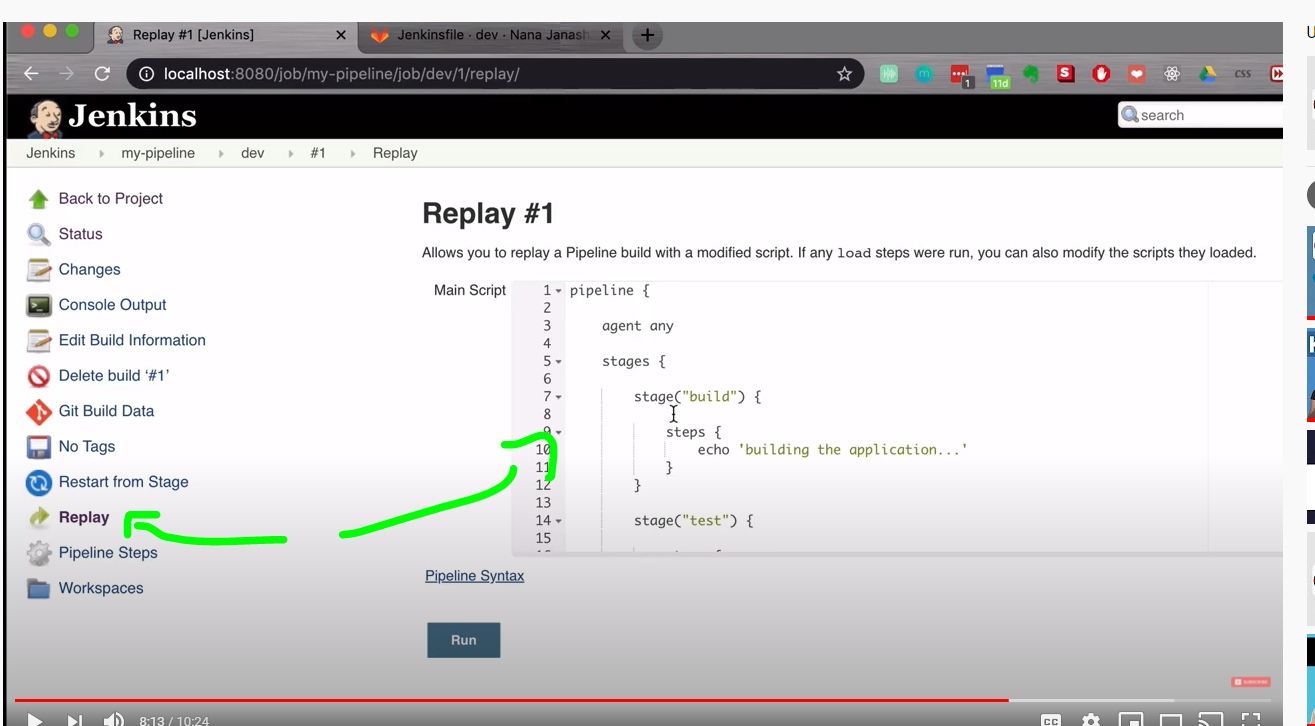


Il primo step qui sopra (Checkout SCM) non è stato definito nel file di pipeline ma è il check iniziale definito nella UI quando abbiamo impostato la connessione a git e le credenziali per l’accesso (si veda sopra)

Se vogliamo modificare una pipeline di quelle eseguite andiamo nell’esecuzione dello specifico branch, qui troviamo lo storico delle esecuzioni:

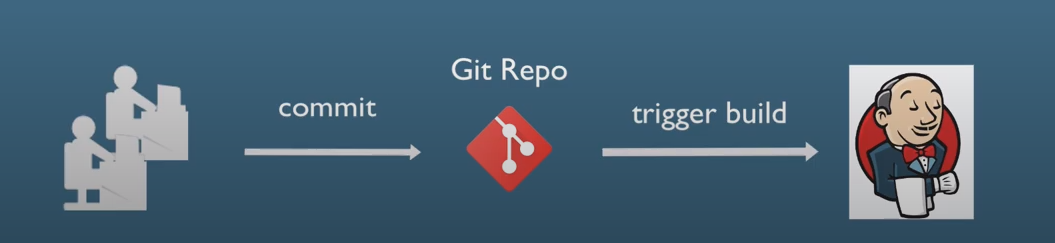


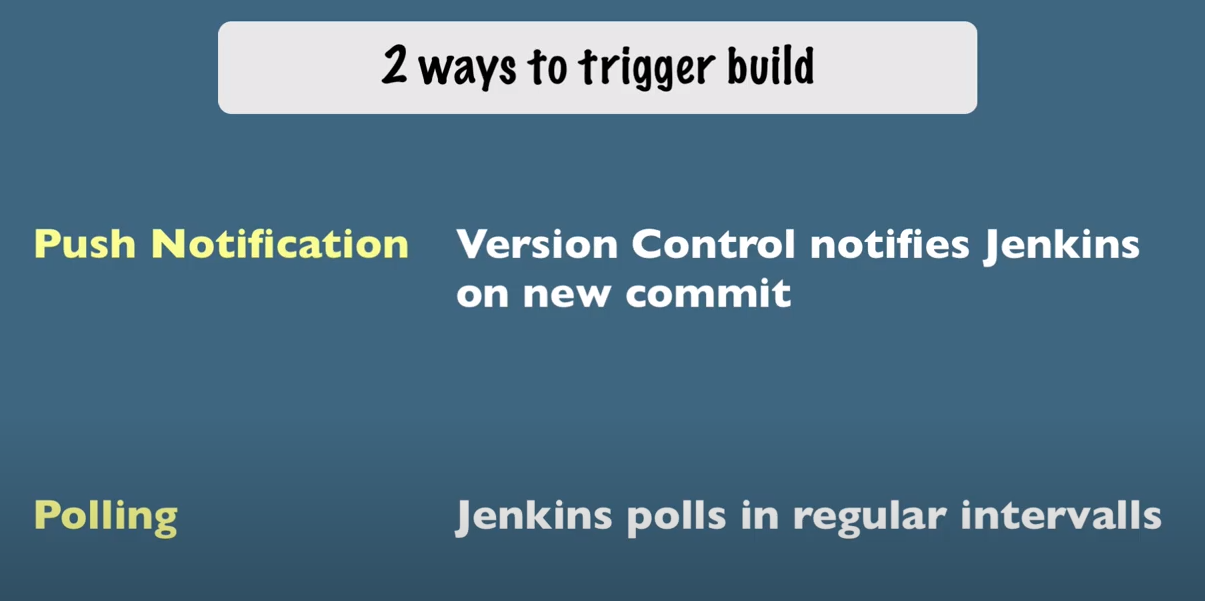
Entriamo dentro una specifica esecuzione e potremo modificare il file di pipeline e rieseguire:



[**https://www.youtube.com/watch?v=CmwTPxdx24Y**](https://www.youtube.com/watch?v=CmwTPxdx24Y)

Ora come facciamo a eseguire jenkins automaticamente su commit di git?

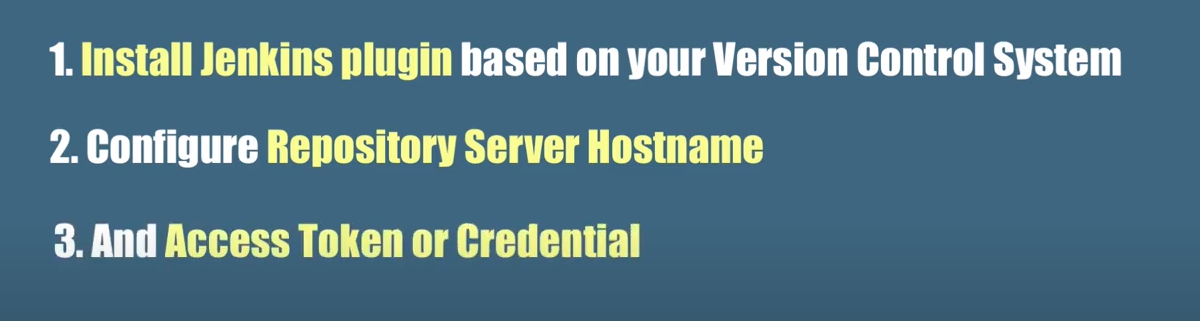


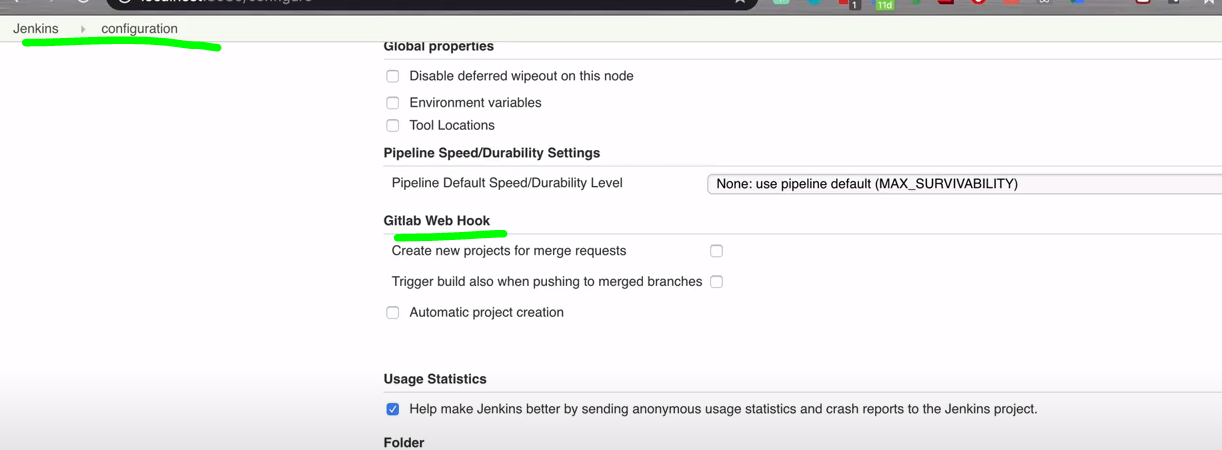


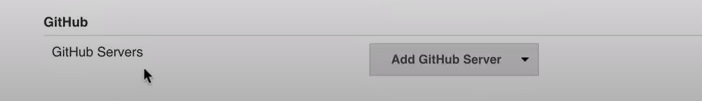
La push notification è più efficiente ma vediamo come configurare entrambe:

Push notification:

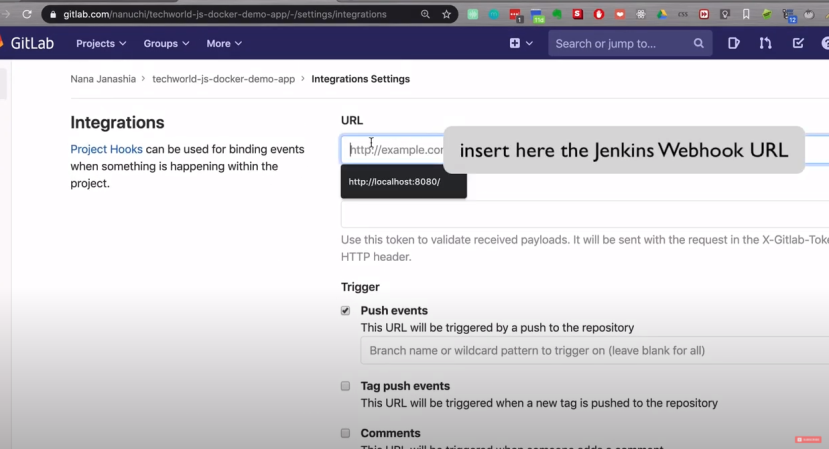
A seconda che si usi github, gita occorre installare il plugin specifico in jenkins. E poi occorre l’access token e il repository hostname:



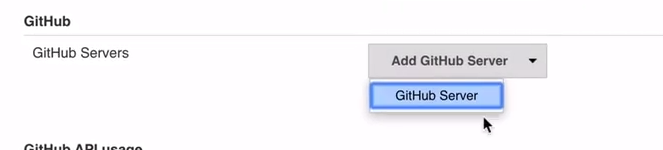


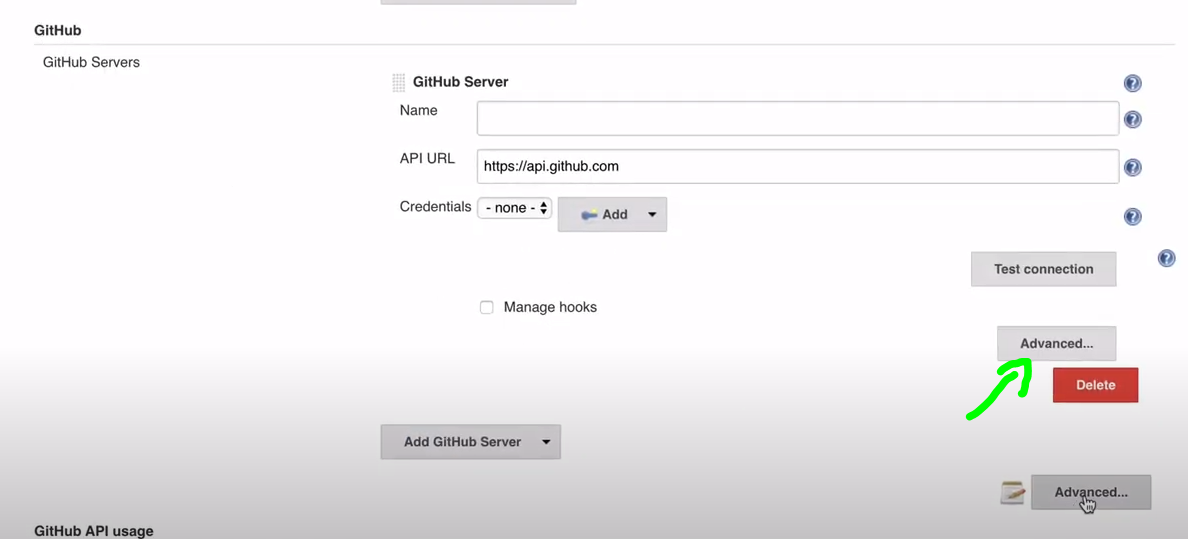


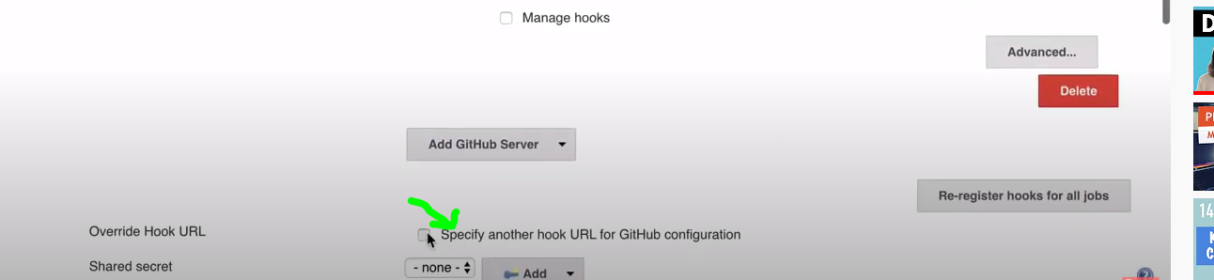
Su gitlab i project hooks:



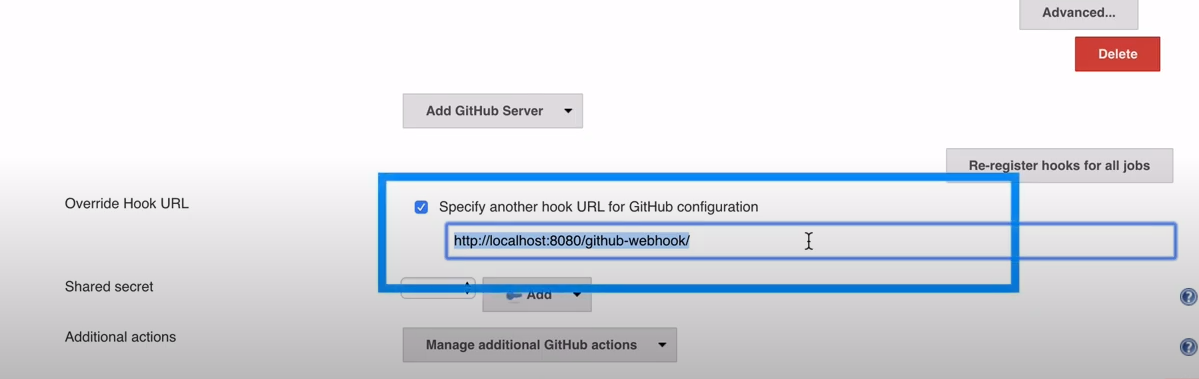
Per recuperare in Jenkins il webhook:



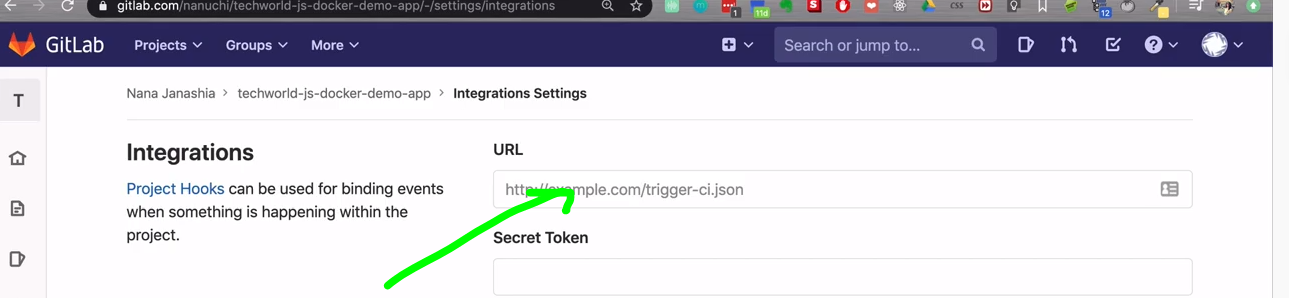




Questo è l’url di jenkins per il webhook di github



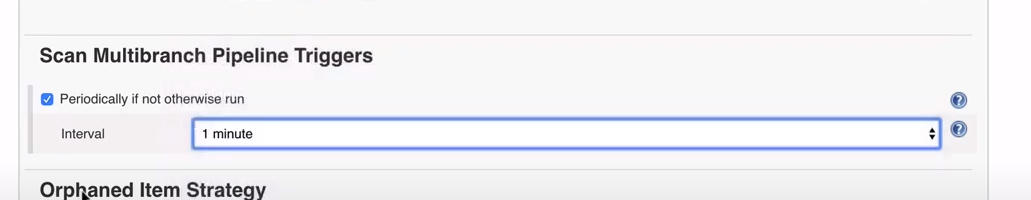
Il webhook andrebbe inserito qui in gitab ma non funzionerebbe perché è localhost:

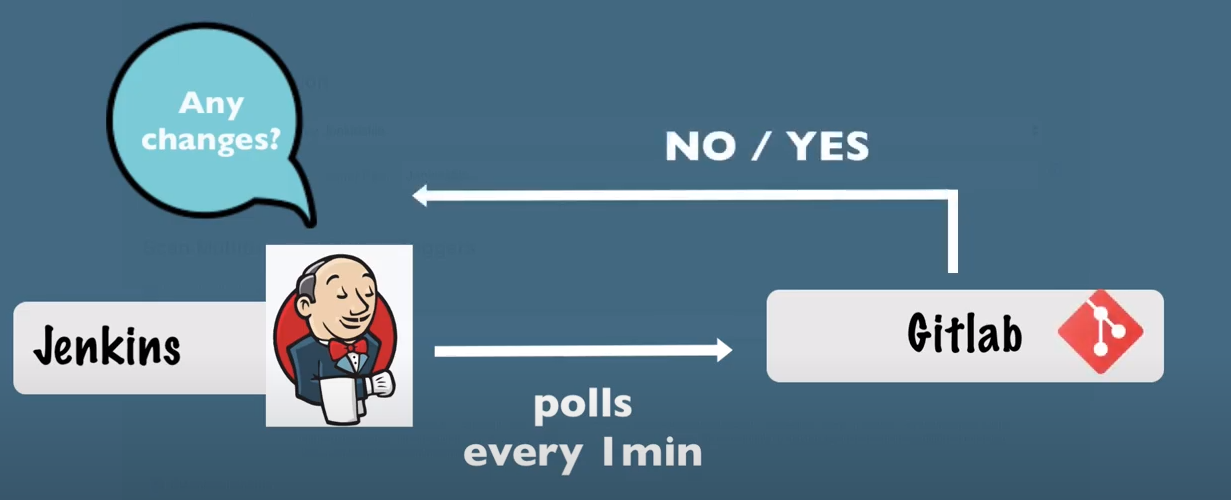


Funziona così:

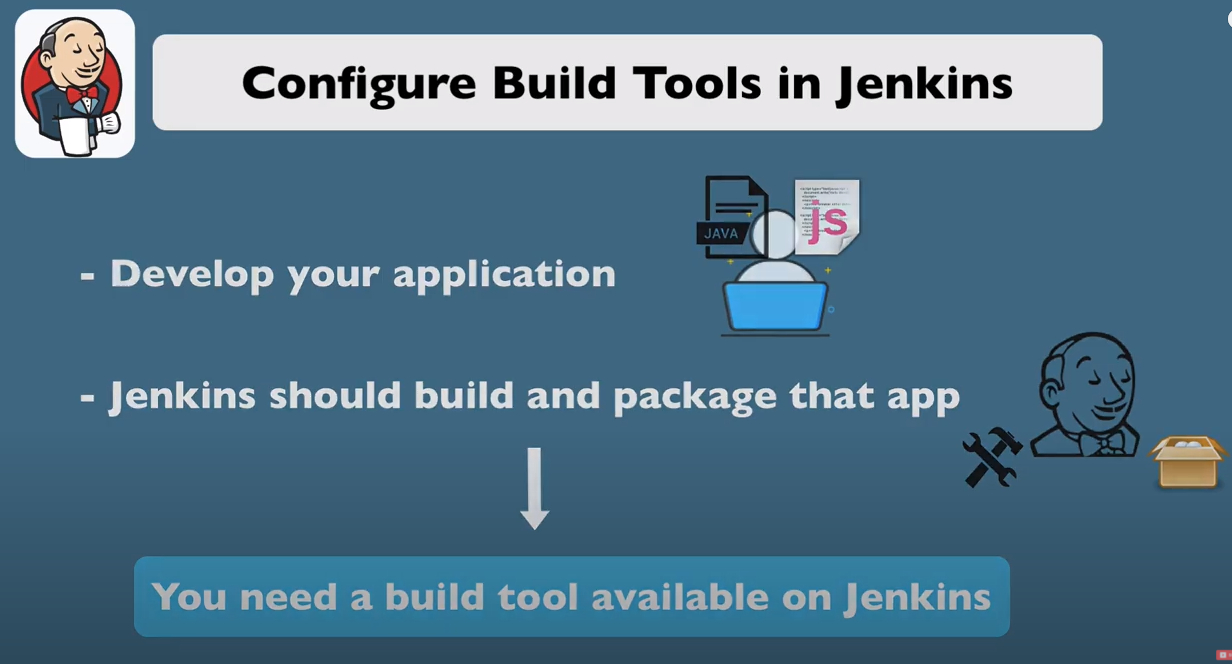






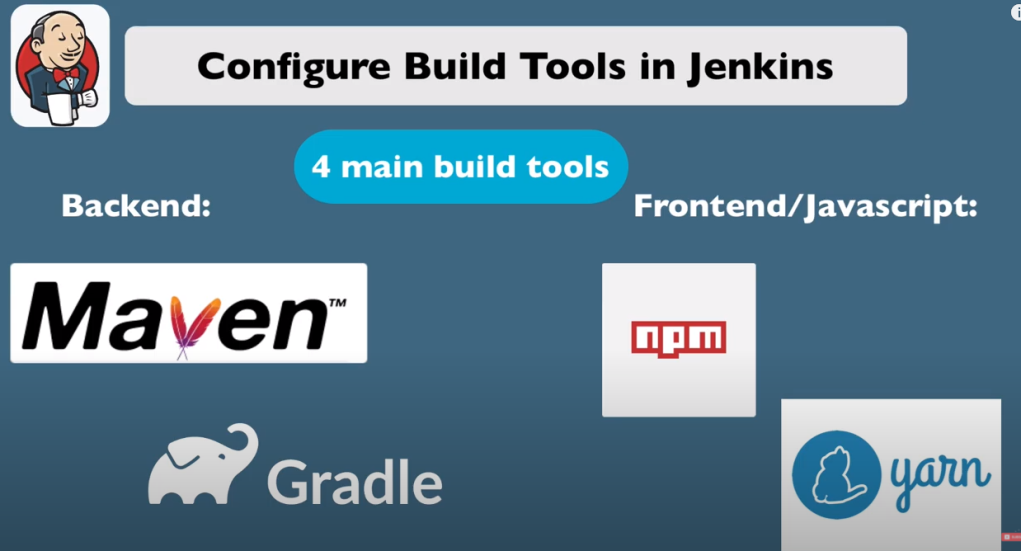


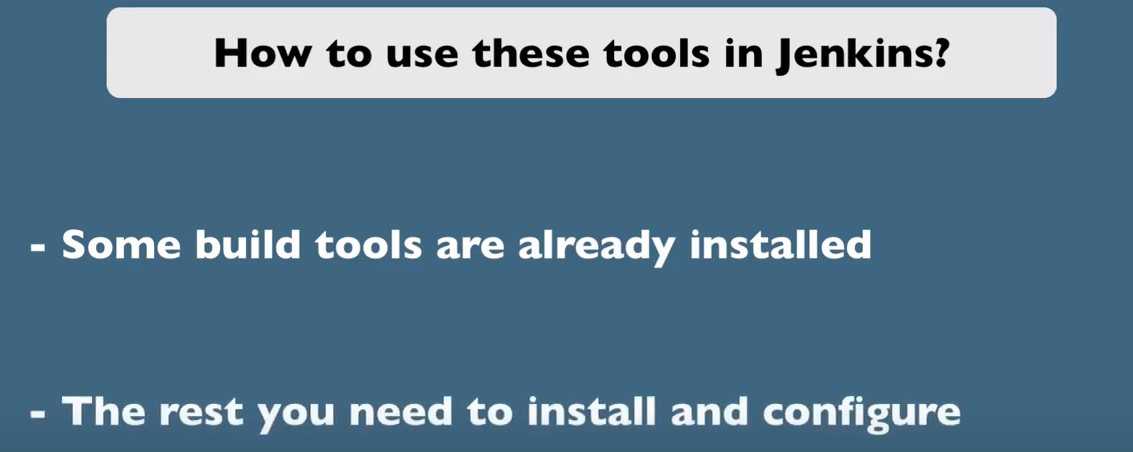
[**https://www.youtube.com/watch?v=L9Ite-1pEU8**](https://www.youtube.com/watch?v=L9Ite-1pEU8)

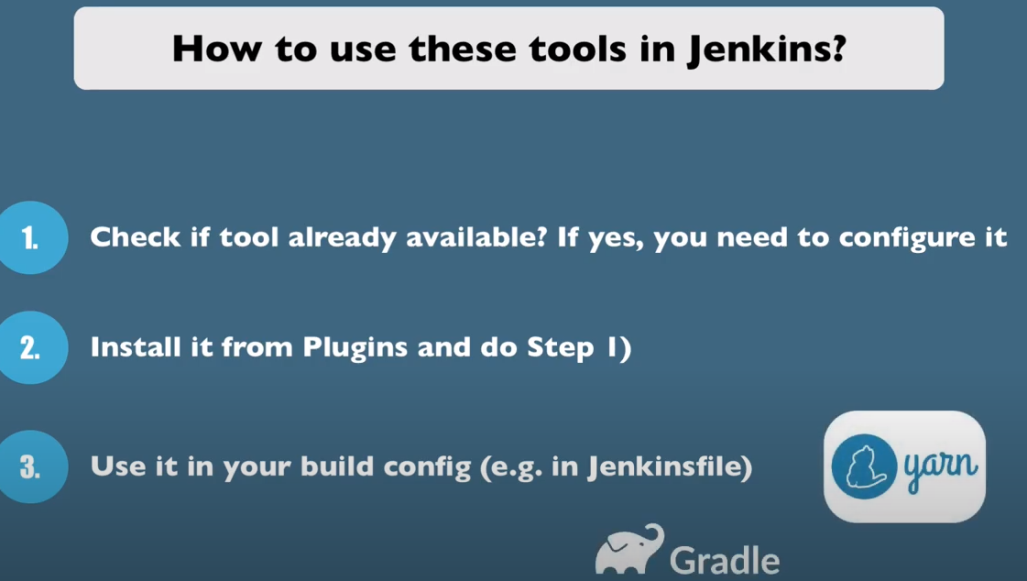


Per compilare l’applicazione jenkins ha bisogno di uno specifico tool.

Per esempio per compilare java (a sinistra) e javascript (a destra) si usano i seguenti tool:

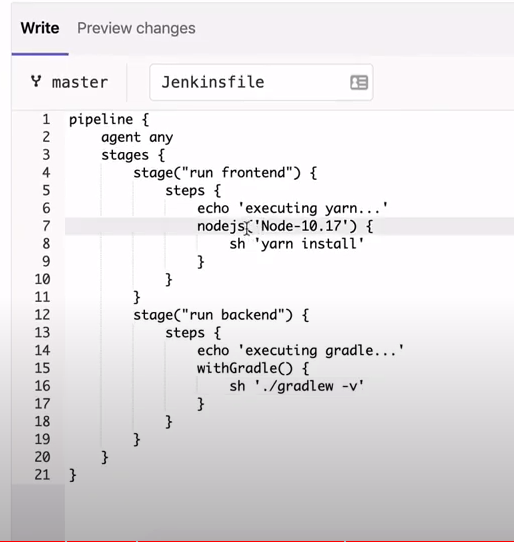






Vediamo un esempio:

dopo aver installato i tools specifici scrivo questo file jenkins:



MSBuild è il tool per microsoft

Costruire un file jenkins:

https://www.youtube.com/watch?v=7KCS70sCoK0