



# Kom igång med git

*Versionshantering - Distribuerat, enkelt, effektivt*

# Sätt upp ett repo

- Du behöver en server med ssh-inloggning
- Skapa en särskild användare för git
- Lägg till public keys för alla användare
- Lägg upp ett repo
- Sätt rättigheter på repot
- Initiera repot
- Klart - börja jobba

# Sätta upp en git-server

```
# adduser git
# su git
$ mkdir .ssh
$ chmod 700 .ssh
$ cd .ssh
$ touch authorized_keys
$ chmod 600 authorized_keys
```

# Nycklar

- Skapa dina ssh-nycklar med ssh-keygen
- Lägg till din publika nyckel till git-kontot
- Den skall appendas till ~/.ssh/authorized\_keys
- Upprepa för andra användare som skall kunna pusha till dit GIT repo...

# Skapa repot

## Skapa en katalog för ditt repo

```
# mkdir /opt/git
```

## Sätt rättigheterna

```
# chown git:git /opt/git
```

```
# chmod 700 /opt/git
```

# Skapa repot

## Bli användaren GIT

```
# su git
```

## Byt till katalogen

```
$ cd /opt/git
```

## Skapa repo och initiera

```
$ mkdir project.git
```

```
$ git init --bare
```

# Ladda upp till repot

## Skapa ditt lokala projekt

```
$ cd project
```

## Initiera git

```
$ git init
```

## Lägg till lite filer och adda dem till lokal git

```
$ git add .
```

# Ladda upp till repot

## Commita dina filer

```
$ git commit -am "Initial commit"
```

## Lägg till din remote upstream

```
$ git remote add origin git@server:/opt/git/project
```

## Pusha filerna till din upstream

```
$ git push origin master
```



# Klona ditt repo till ny katalog

```
$ git clone git@host.net:/opt/git/project project
```

Skapar en lokal git med senaste versionen redo för dig att börja jobba med. Kör sedan commit/push när du har lust mot ditt repo

```
$ git commit -am "Meddelande"
```

```
$ git push origin
```

Ändringarna är nu pushade

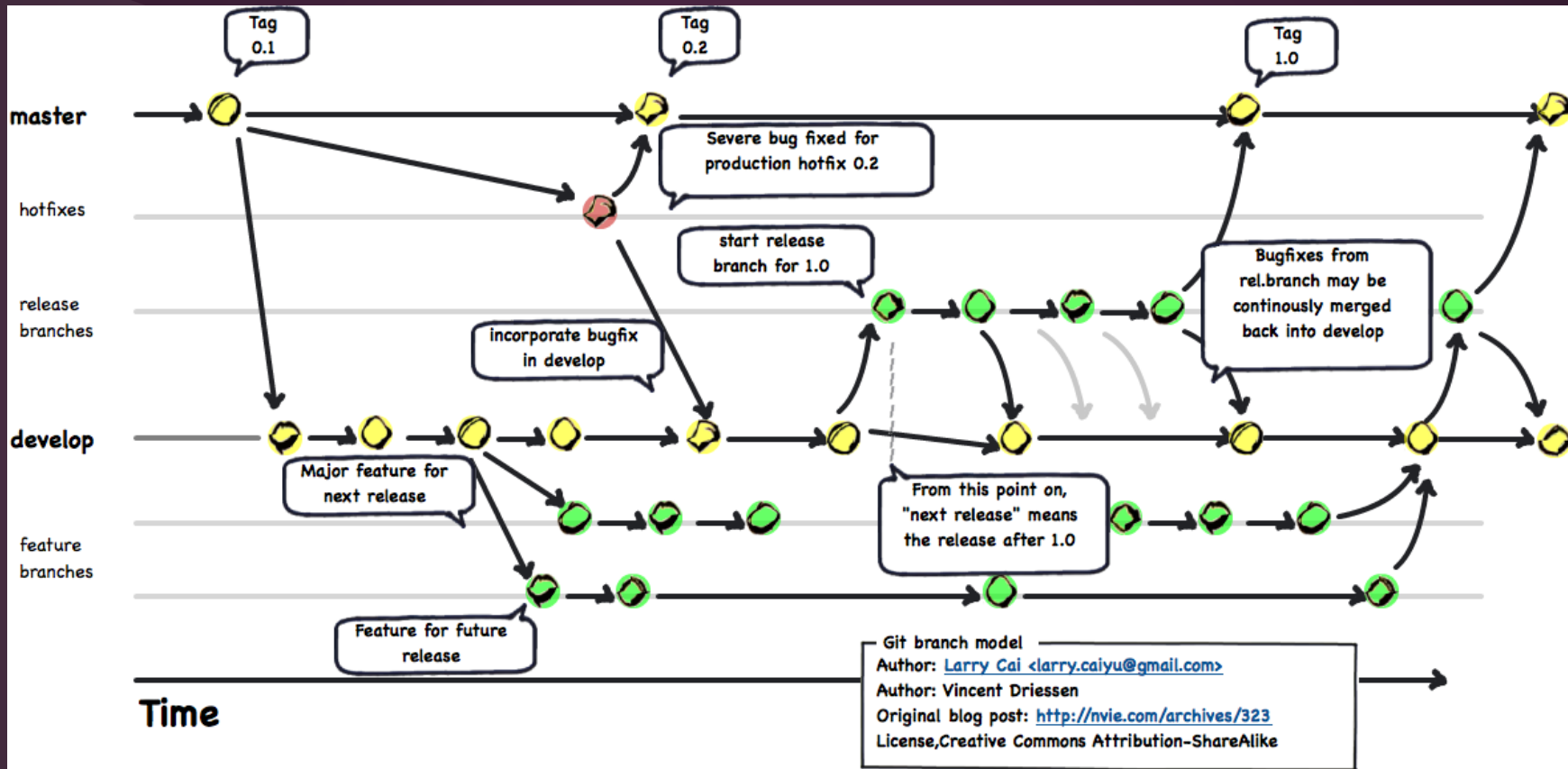
# Synka från repo till lokalt

```
$ git pull
```

Hämtat från origin till lokalt.

# Branching

- En branch är en förgrening av ditt projekt
- Användbart när man jobbar på större ändringar
- Ibland arbetar olika personer på olika branches för att undvika att man måste göra merge ofta
- Eller om man vill frysa koden mot en särskild release
- När du är klar gör du oftast merge med din master, gren för gren, tills alla är mergade och du har en release
- Därefter kan du tagga den och pusha den taggen
- Då har du alltid ett snapshot att jobba med om du behöver checka ut tidigare version enkelt



# Branching

## Skapa ny branch

```
$ git branch <name>
```

## Checka ut den nya branchen

```
$ git checkout <name>
```

## Pusha branch upstream

```
$ git push origin <name>
```

# Exempel på branch

```
anders@lyta:~/dev/ipta$ git branch
```

```
* argp
```

```
master
```

```
anders@lyta:~/dev/ipta$
```

# Merge

När du vill merga din branch med huvudspåret “main” igen gör du bara följande:

```
$ git checkout master
```

```
$ git merge <branch>
```

# Merge

- Om du också gjort ändringar i main samtidigt kan du få konflikter.
- Dessa markeras i merge.
- Du får nu chansen att gå in i filerna och välja vilken kod som du skall behålla.
- Merging går oftast faktiskt rätt smärtfritt



# Tags

Tagga och pusha till upstream:

```
$ git tag tagname
```

```
$ git push tagname
```

Vill du checka ut en tag? Inga problem:

```
$ git checkout tagname
```

# Loggar

```
anders@zathras:~/www/ichimusai.org$ git log
commit 4652c456a50fcafca2402859e9ce84ba198736e5
Author: Anders Sikvall <anders@sikvall.se>
Date:   Wed Jul 8 23:40:51 2015 +0200
```

Update

```
commit dcd5044b4bf8cdf36d7bf858e1f740f6f3ba1a36
Author: Anders Sikvall <anders@sikvall.se>
Date:   Fri May 8 12:50:16 2015 +0200
```

First initial commit

# Flytta fil, radera fil

- I en git-kontrollerad värld använder vi inte `rm` och `mv`
- I stället använder vi `git rm` och `git mv`
- De fungerar ungefär likadant men uppdaterar gits interna strukturer så det fungerar som tänkt
- Därefter, som vanligt - `commit`, `push`

# Grunderna...

- Det finns förstås mycket mer att lära om git
- Detta är dock tillräckligt för att komma igång
- Vill du hosta ditt projekt för flera att jobba med, kolla t. ex. github eller någon annan tjänst
- Eller sätt upp en egen server, helt vanilj Linux ok, ingen server side programvara behövs!
- Git pratar många protokoll som ssh, https osv...

# Tack

*Ut och lek med git*