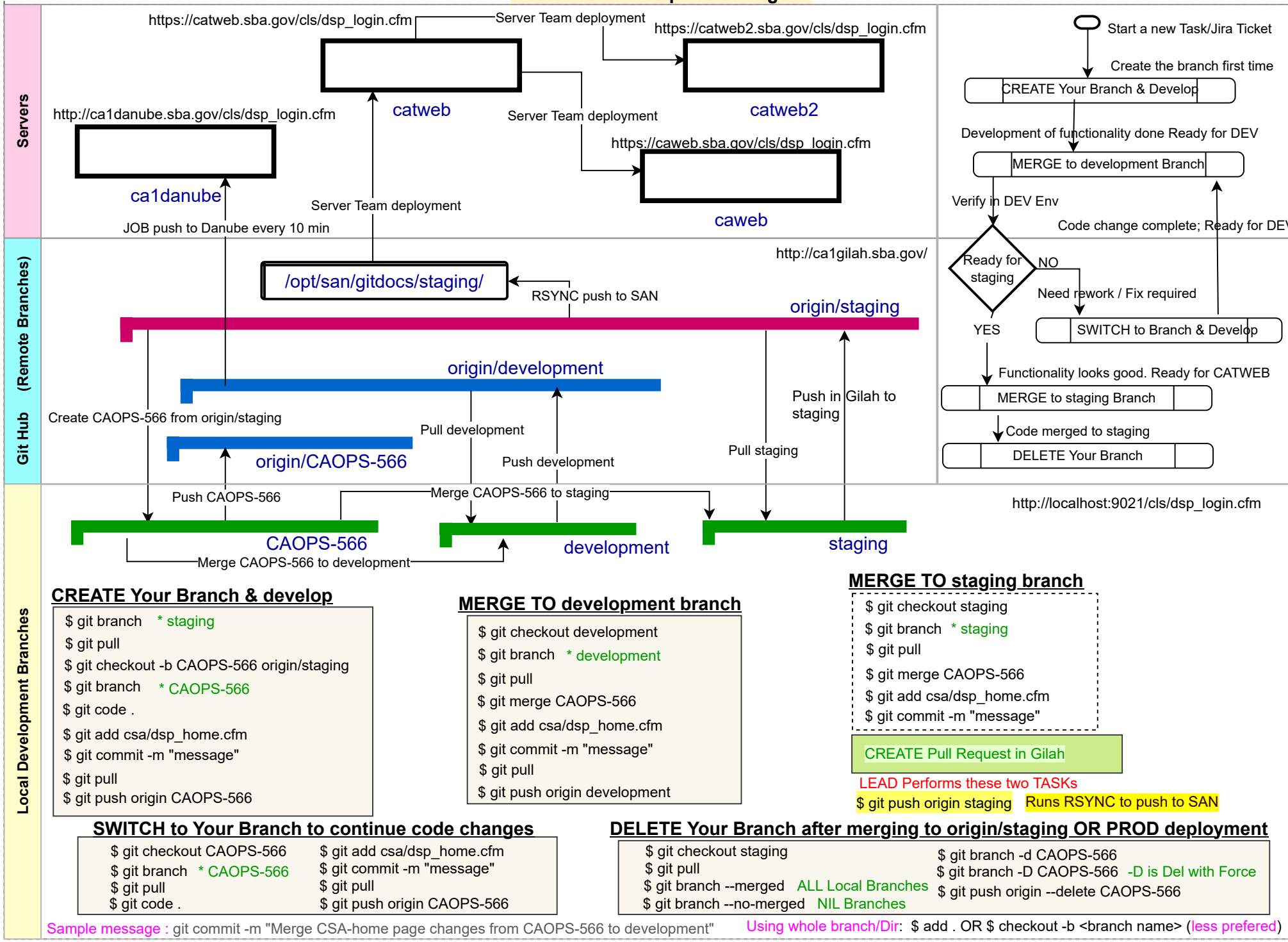
SBA Local Development using Git



CREATE Your Branch & develop

```
$ git branch    * staging
$ git pull
$ git checkout -b CAOPS-566 origin/staging
$ git branch    * CAOPS-566
$ git code .
$ git add csa/dsp_home.cfm
$ git commit -m "message"
$ git pull
$ git push origin CAOPS-566
```

MERGE TO development branch

```
$ git checkout development
$ git branch   * development
$ git pull
$ git merge CAOPS-566
$ git add csa/dsp_home.cfm
$ git commit -m "message"
$ git pull
$ git push origin development
```

MERGE TO staging branch

```
$ git checkout staging
$ git branch   * staging
$ git pull
$ git merge CAOPS-566
$ git add csa/dsp_home.cfm
$ git commit -m "message"
```

CREATE Pull Request in Gilah

LEAD Performs these two TASKs

$ git push origin staging Runs RSYNC to push to SAN

SWITCH to Your Branch to continue code changes

```
$ git checkout CAOPS-566
$ git branch   * CAOPS-566
$ git pull
$ git code .
$ git add csa/dsp_home.cfm
$ git commit -m "message"
$ git pull
$ git push origin CAOPS-566
```

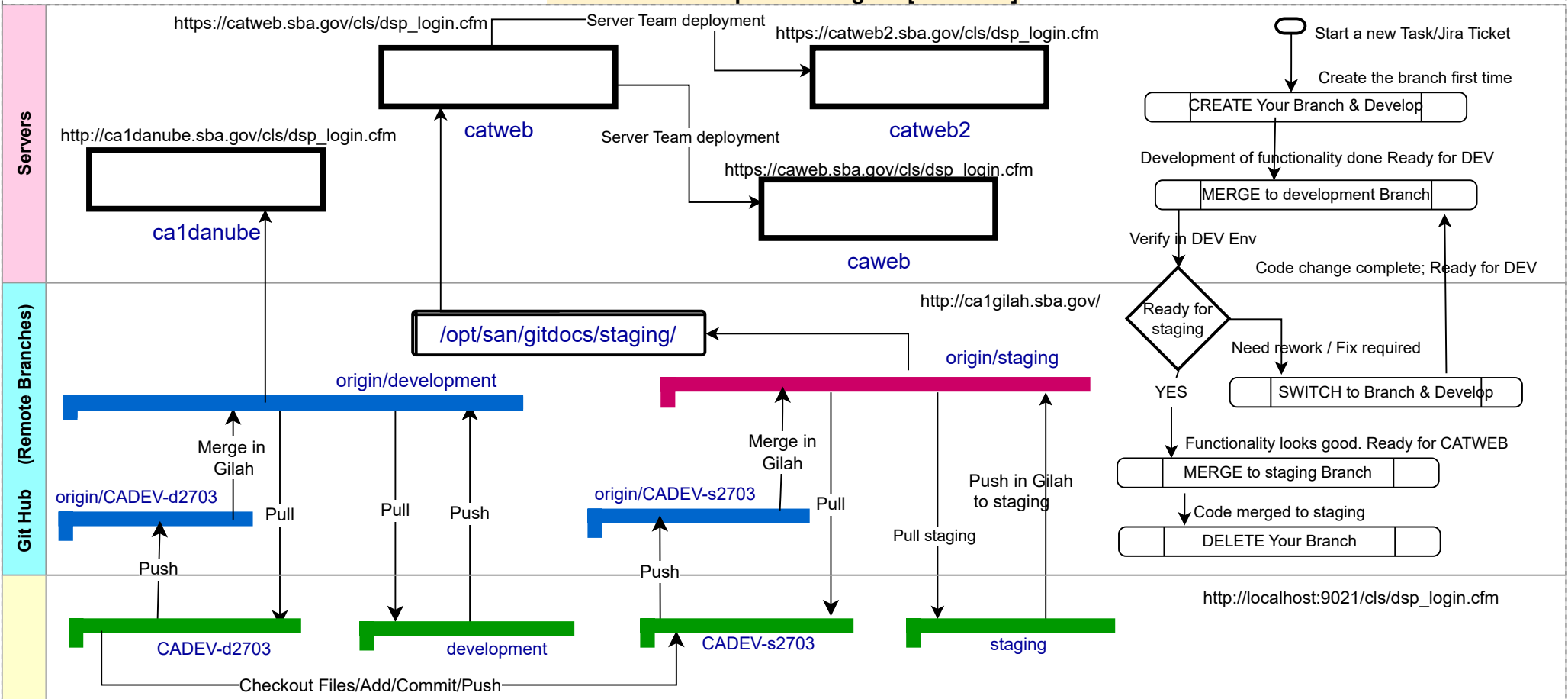
DELETE Your Branch after merging to origin/staging OR PROD deployment

```
$ git checkout staging
$ git pull
$ git branch --merged     ALL Local Branches
$ git branch --no-merged  NIL Branches
$ git branch -d CAOPS-566
$ git branch -D CAOPS-566   -D is Del with Force
$ git push origin --delete CAOPS-566
```

Sample message : git commit -m "Merge CSA-home page changes from CAOPS-566 to development"

Using whole branch/Dir: $ add . OR $ checkout -b <branch name> (less prefered)

SBA Local Development using Git [REVISED]



CREATE 2 Branches per Project, each pointing to [development] and [staging]

$ git branch
$ git branch -b CADEV-**d**2703 origin/development
$ git branch -b CADEV-**s**2703 origin/staging
$ git checkout CADEV-d2703 [Develop Here]
$ git add FILENAME [You are in CADEV-d2703]
$ git commit -m "CADEV-d2703 Your Message"
$ git pull [Stream points development]
$ git push origin CADEV-d2703

GOTO Gilah in Browser
CREATE Pull Request between development & CADEV-d2703
FILES will be in development & DEV Server/CHECK in DEV

$ git checkout CADEV-s2703
$ git pull [You are in CADEV-s2703] [Stream points staging]
$ git checkout CADEV-d2703 FILENAME [REPEAT more files]
$ git add FILENAME [REPEAT more files]
$ git commit -m "CADEV-s2703 Your Message"
$ git pull
$ git push origin CADEV-s2703

GOTO Gilah in Browser
CREATE Pull Request between staging & CADEV-s2703
WAIT for Approval and files will be in staging Location
SEND Deployment request to CATWEB/TEST

End of Project REMOVE CADEV-d2703 & CADEV-s2703
Local Branches development and staging need only be run
$ git pull
MERGE action is done in Gilah between appropriate branches
[development] ← [CADEV-d2703]
[staging] ← [CADEV-s2703]
DONOT MERGE cross stream branches
VERIFY How many files being merged during COMPARISON
STOP when you find more files than your changes
FIGURE out the issue and FIX before proceeding further
PREFERABLY Use named additions to $ git add FILENAME

v6.0 / 06-22-2023 / Siva Maniam