

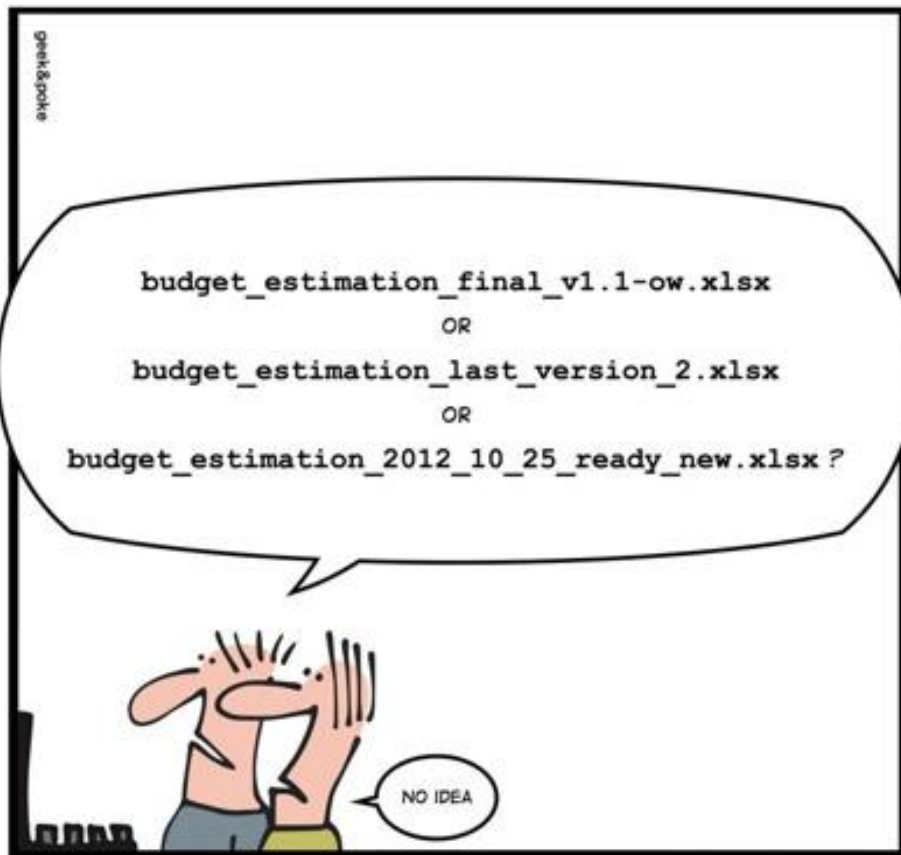
Git source control

Valdas Maksimavičius

Why source control?

- Which changes were made?
- Who made the changes?
- When were the changes made?
- Why were changes needed?

SIMPLY EXPLAINED



VERSION CONTROL

Why GIT?

1. Work Offline (local)
2. Difficult to Screw Things Up (distributed)
3. Make Useful Commits (staging)
4. Work in Your Own Way (flow)
5. Don't Mix Things Up (branching)
6. Go With the Flow (popularity)

Centralized vs. Decentralized

	Centralized (TFVC)	Decentralized (GIT)
Hard drive space required for history?	No	Yes
Who has latest revision?	Central "master version"	Variable, need to make rules
Where is full history?	Central machine	Local machine
Work offline?	No*	Yes
How fast are operations?	Network-dependent	Fast, most are local
Branching and merging?	Reliable, use with caution	Reliable, used often
Learning curve	Relatively simple	Relatively hard



Code Download : **Create Workspace >> Map & Get**

Get latest code (First Time) : **Get Latest Version**

Get latest code (After first Time) : **Get Latest Version**

Code Commit : **Check In**

Get latest after code commit: **Check In + Get Latest Version**

Check out a file for editing: **Check Out**

Review Code: **Code Review**

Shelving Code: **Shelveset**

Files to be included for commit: **Included Changes**

Files to be excluded for commit: **Exclude Changes**

Code Annotation : **Annotate**



Code Download : **Create Repository >> Clone**

Get latest code (First Time) : **Clone**

Get latest code (After first Time) : **Pull**

Code Commit : **Commit + Push**

Get latest after code commit: **Sync**

Check out a file for editing: **Just start editing**

Review Code: **Pull Request**

Shelving Code: **Stash (GUI available in VS 2019)**

Files to be included for commit: **Staged**

Files to be excluded for commit: **Unstage**

Code Annotation : **Blame**

What's a repository?

The entire collection of files and folders + each file's revision history (snapshots)

What's a commit?

Files history as a snapshot in time

What's a branch?

A linked-list of commits



Local



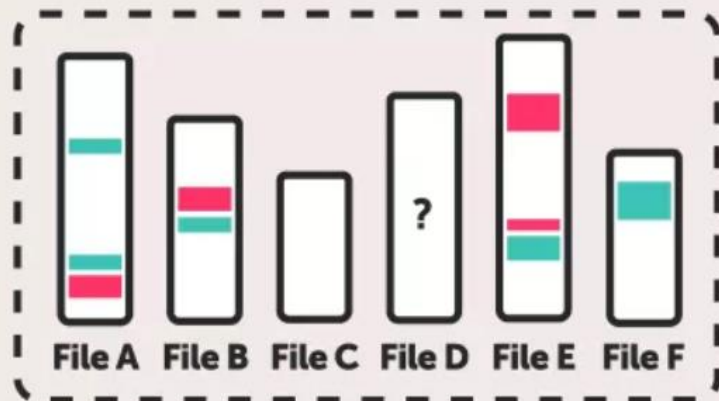
Remote




Clone a **full-fledged repository.**



View commit history. Offline.

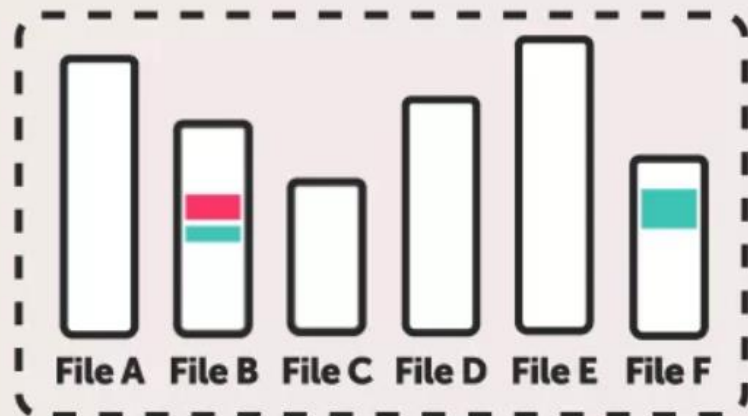
modified



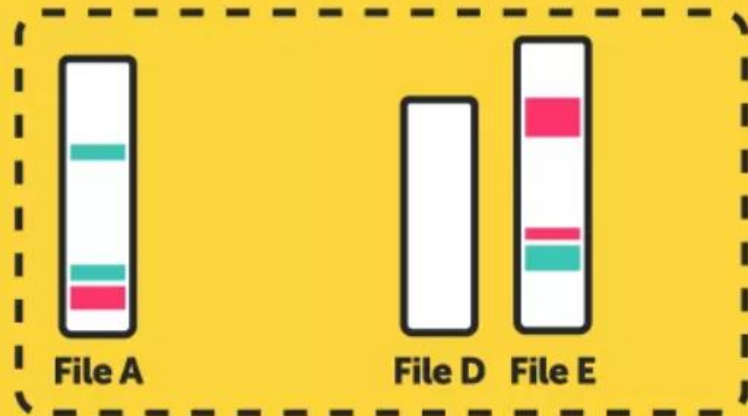
 removed line
 added line
 new file



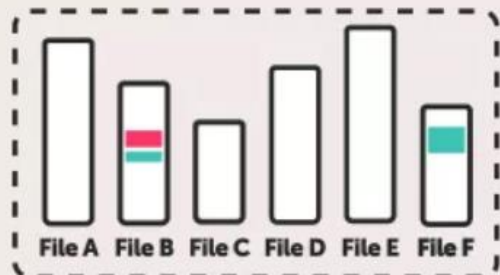
modified



staged



modified

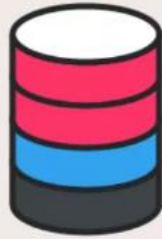


staged



committed



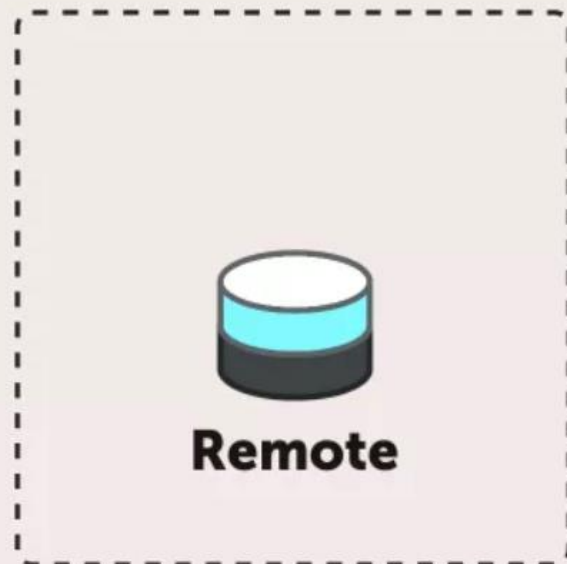
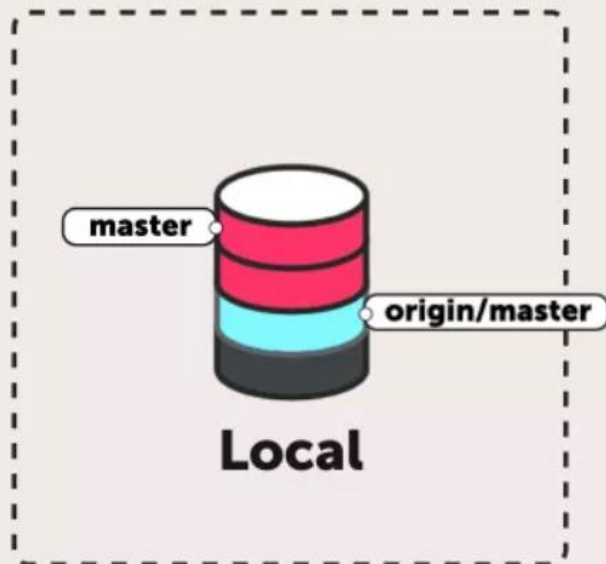


Local

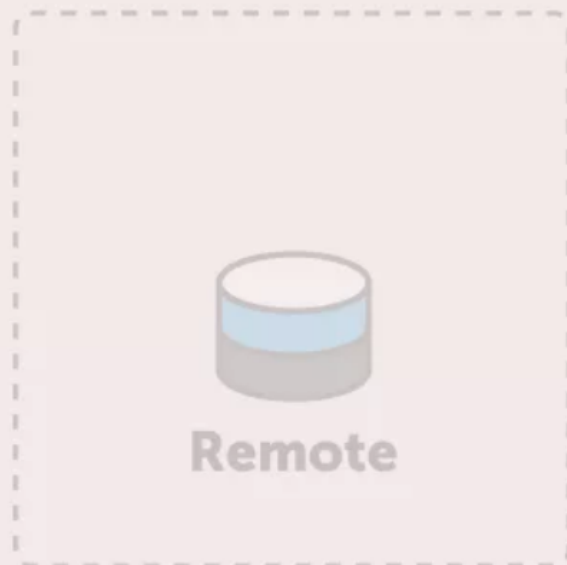
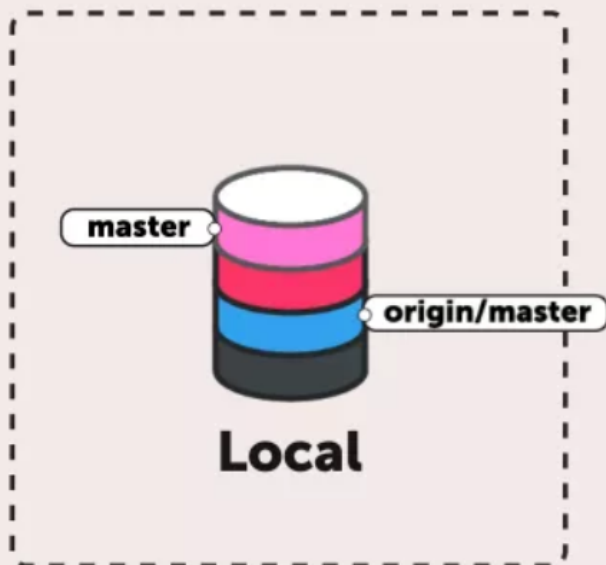


Remote

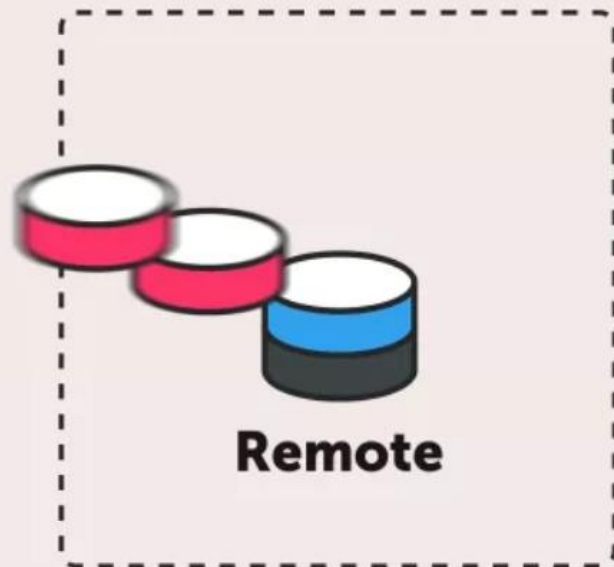
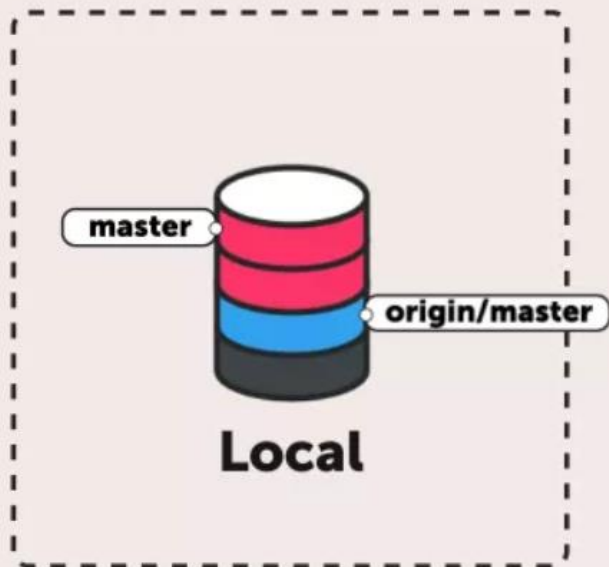
Commit locally. Push later.



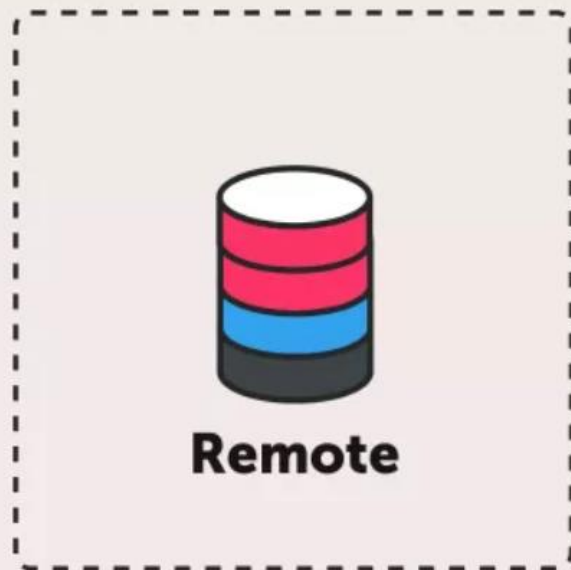
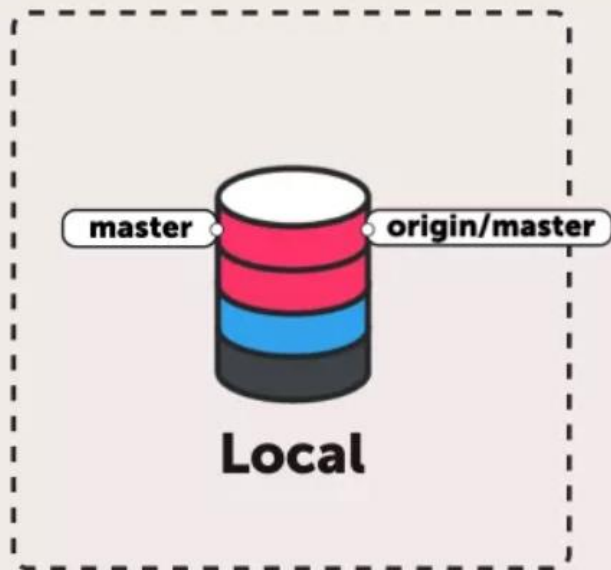
**origin/master points to
the latest *syncd* version**



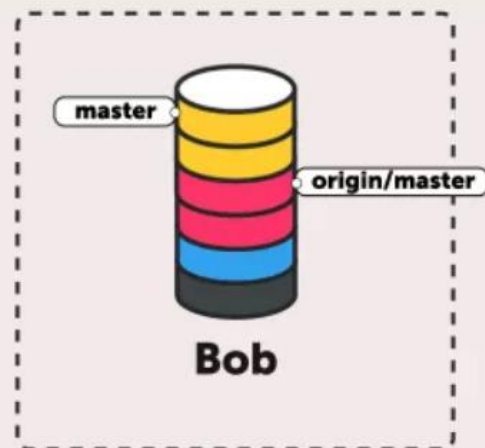
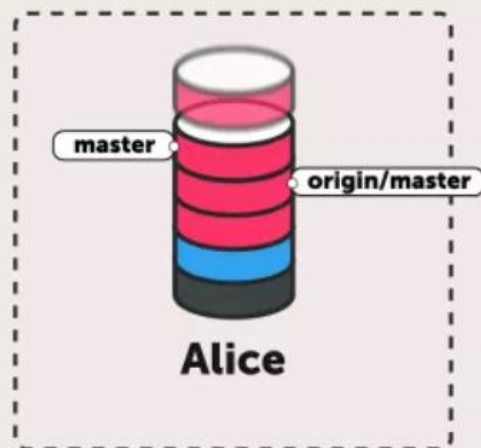
master points to
the latest **local** version



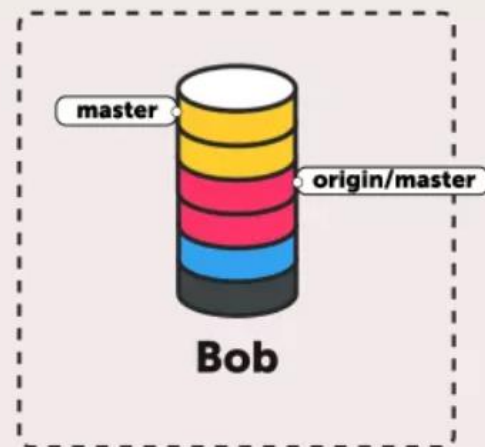
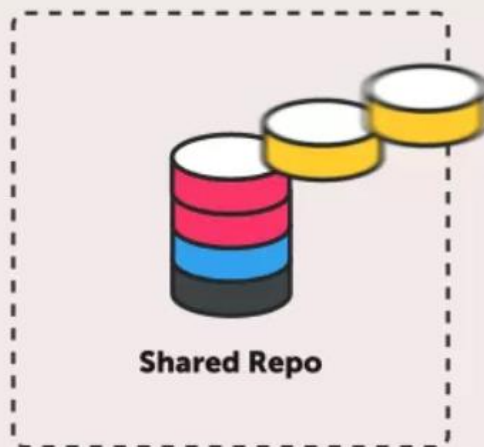
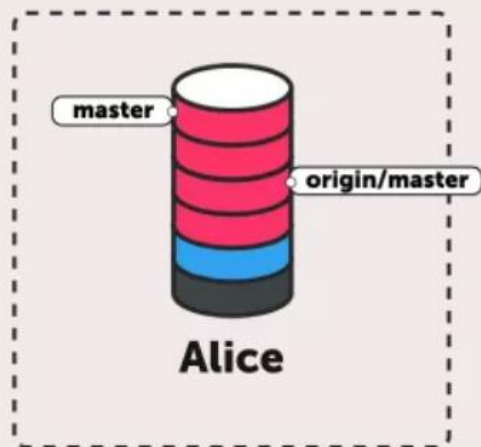
Push local changes



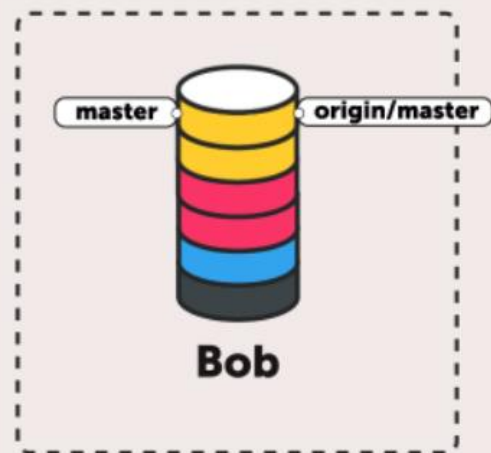
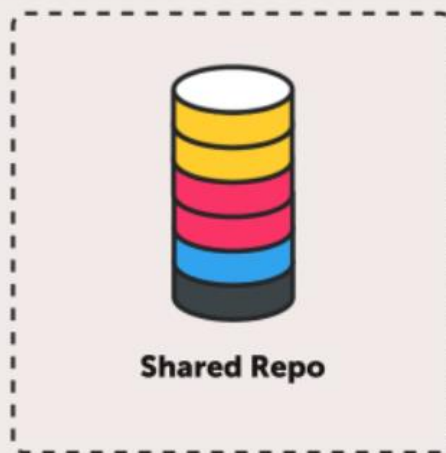
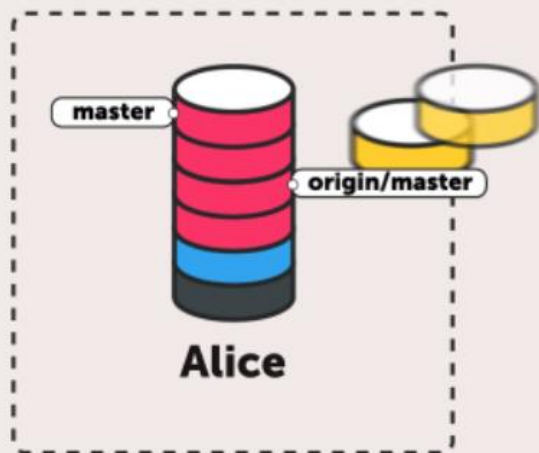
origin/master is updated



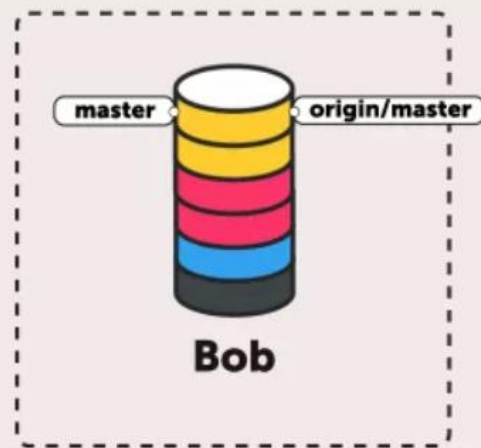
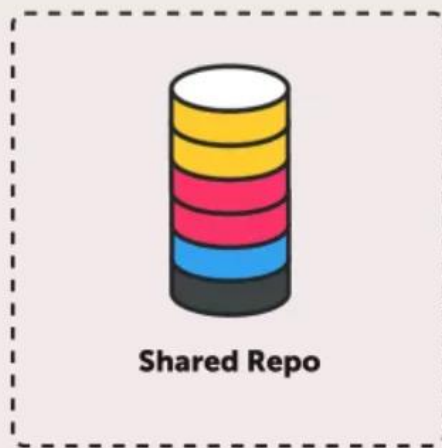
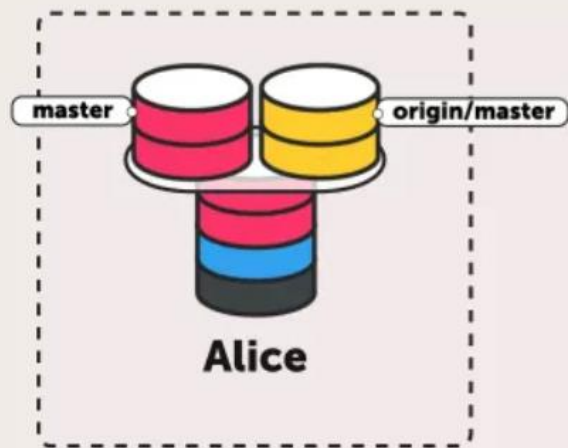
Alice and Bob **commit in parallel**



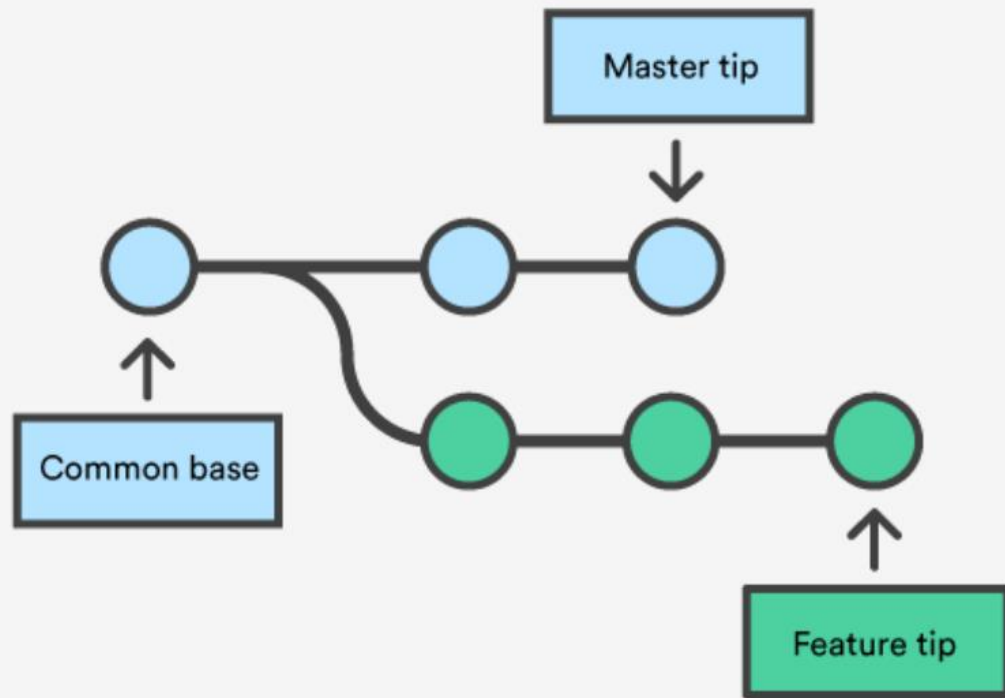
Bob pushes his work first



Alice must **fetch** the latest changes



Alice has two divergent branches



Fast forward merge (default)

It happens in a situation when there are no commits in an another branch. Then it just updates the branch pointer to the last commit.

Merge

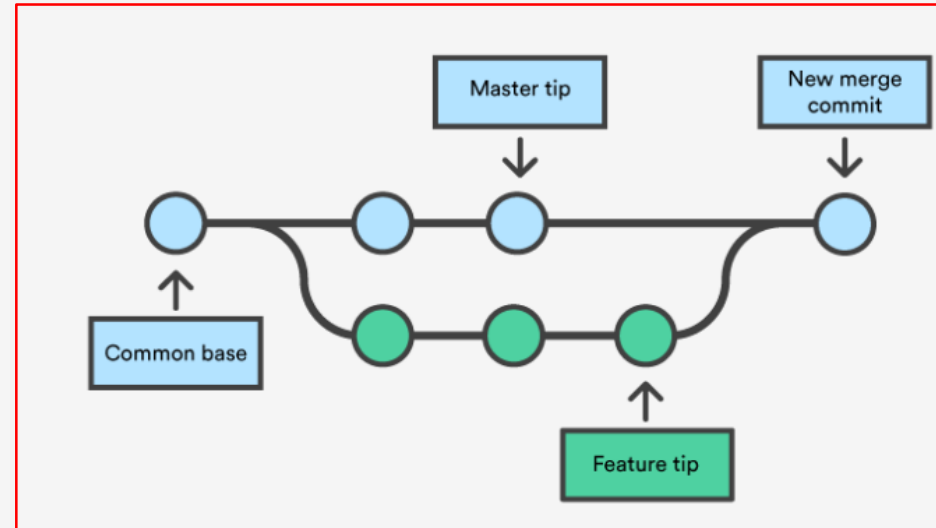
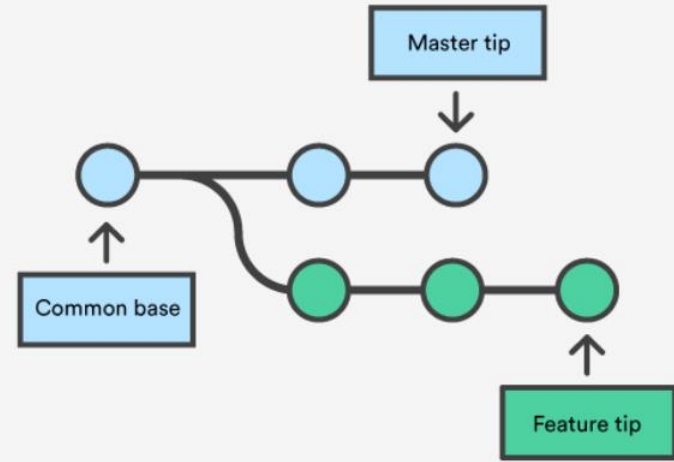
Merge is a new commit. It is a simple commit with one difference – it has two parents. Possible to commit immediately if there is no conflicts

Rebase (hiding the sausage)

A user can simulate a fast-forward by rebasing rather than merging. Rebase is recreating your work of one branch onto another. For every commit that you have on the feature branch and not in master, new commit will be created on top of the master. **New commit for every old one, with the same changes**

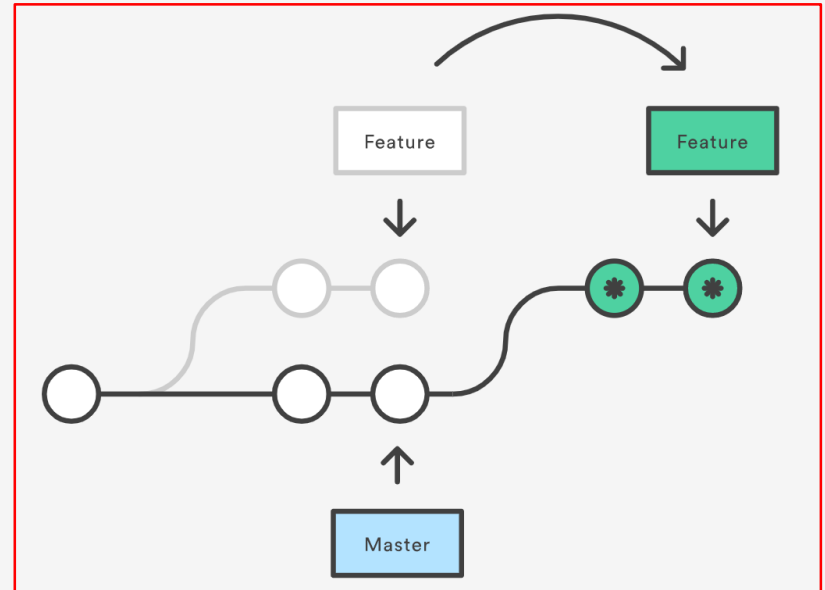
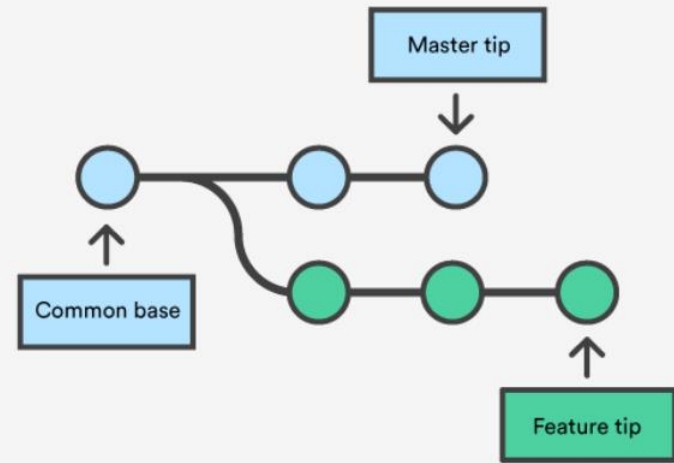
Merge

Merge is a new commit. It is a simple commit with one difference – it has two parents. Possible to commit immediately if there is no conflicts



Rebase

A user can simulate a fast-forward by rebasing rather than merging. Rebase is recreating your work of one branch onto another. For every commit that you have on the feature branch and not in master, new commit will be created on top of the master. **New commit for every old one, with the same changes**



Conflict markers

If you have questions, please

<<<<<<< HEAD

ask your question on Skype.

=====

ask your question on Teams.

>>>>>>> branch-a

Git **prevents** you from **overwriting** the central repository's **history** by refusing push requests when they result in a non-fast-forward merge.

**PUSH REJECTED, REBASE OR
MERGE**

GIT PUSH --FORCE

memegenerator.net

git fetch

Downloads the remote content but doesn't update your local repo's working state, leaving your current work intact. Safe option

git pull

More aggressive alternative, it will download the remote content for the active local branch and immediately execute git merge to create a merge commit for the new remote content. Eventually, it runs two commands: git fetch + git merge

git push

It's used to write to a remote repository. Pushes local commits to remote branches

git reset

A simple way to undo commits that haven't been shared with anyone else.
Rewrites history.

git revert

Reverting undoes a commit by creating a new commit. This is a safe way to undo shared with others changes, as it has no chance of re-writing the commit history

git checkout

Switch between branches or inspect old snapshots. Don't add new commits on top of old snapshots!!!

DANGER – git checkout

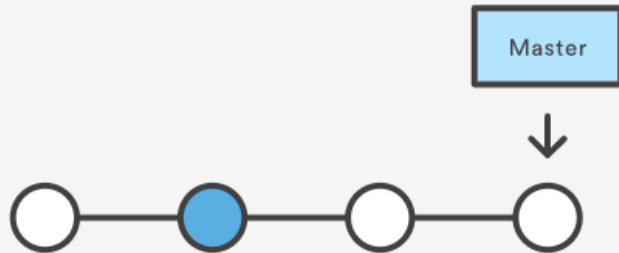
- `git checkout <branchname>`
- `git checkout 911e8c9 #detached head`
- `git checkout <branchname> #remember to pick latest`

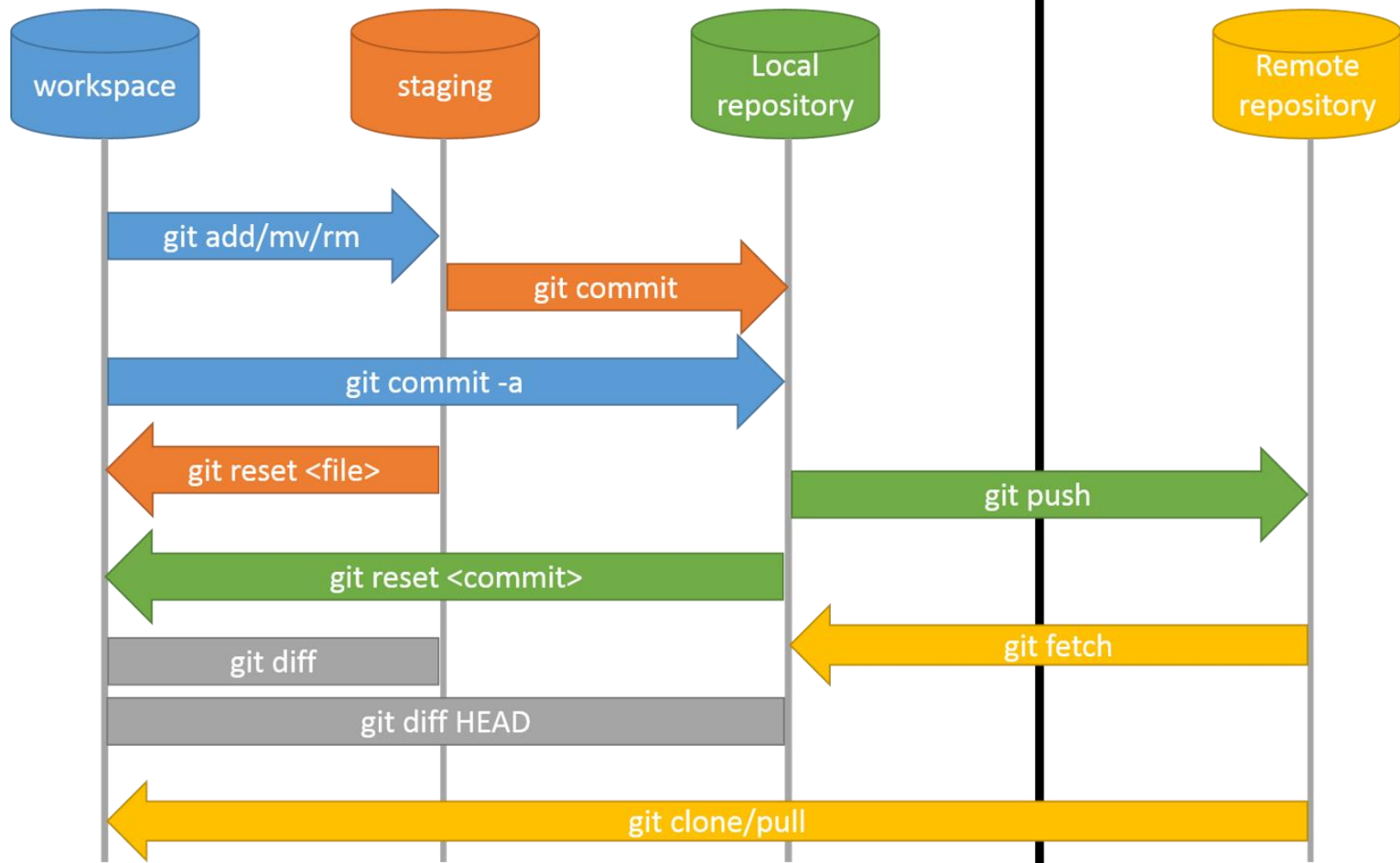
The point is, your development should always take place on a branch - never on a detached HEAD.

Attached HEAD



Detatched HEAD

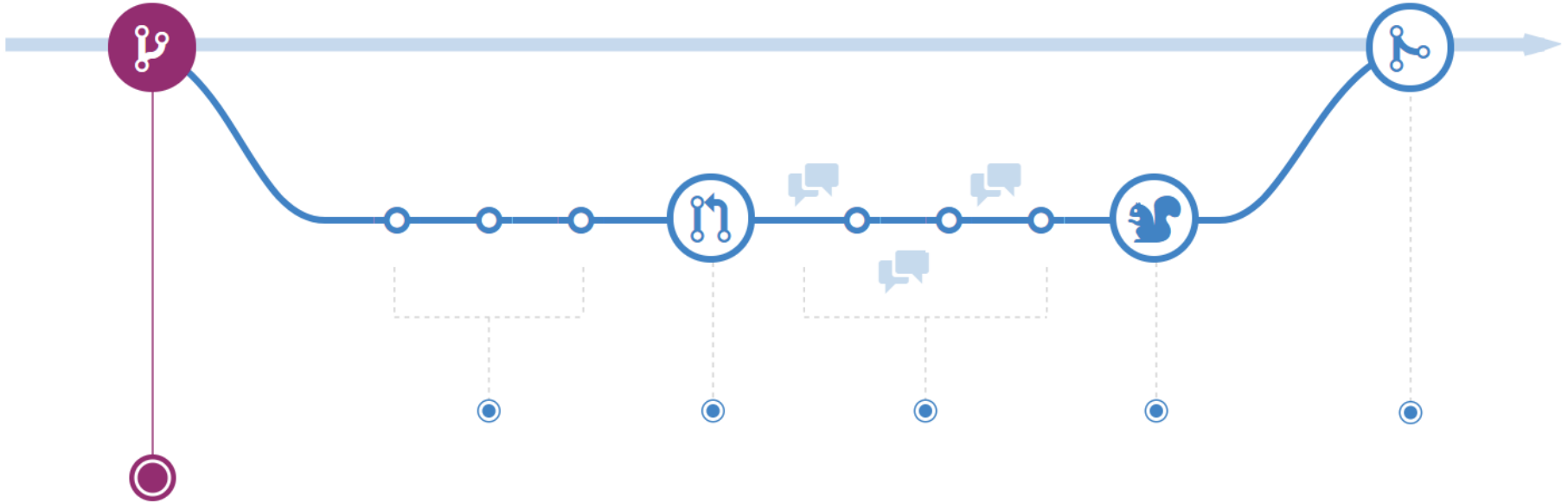




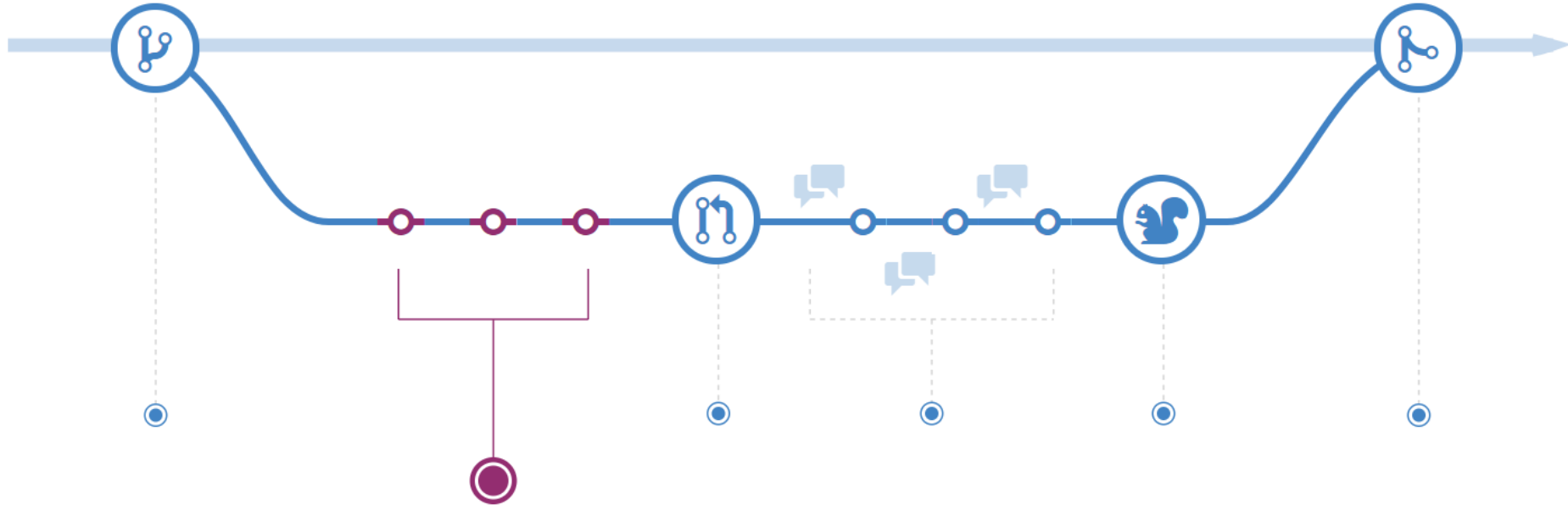
GitHub flow

1. **Create a branch:** Topic branches created from the canonical deployment branch (usually `master`) allow teams to contribute to many parallel efforts. Short-lived topic branches, in particular, keep teams focused and results in quick ships.
2. **Add commits:** Snapshots of development efforts within a branch create safe, revertible points in the project's history.
3. **Open a pull request:** Pull requests publicize a project's ongoing efforts and set the tone for a transparent development process.
4. **Discuss and review code:** Teams participate in code reviews by commenting, testing, and reviewing open pull requests. Code review is at the core of an open and participatory culture.
5. **Merge:** Upon clicking merge, GitHub automatically performs the equivalent of a local 'git merge' operation. GitHub also keeps the entire branch development history on the merged pull request.
6. **Deploy:** Teams can choose the best release cycles or incorporate continuous integration tools and operate with the assurance that code on the deployment branch has gone through a robust workflow.

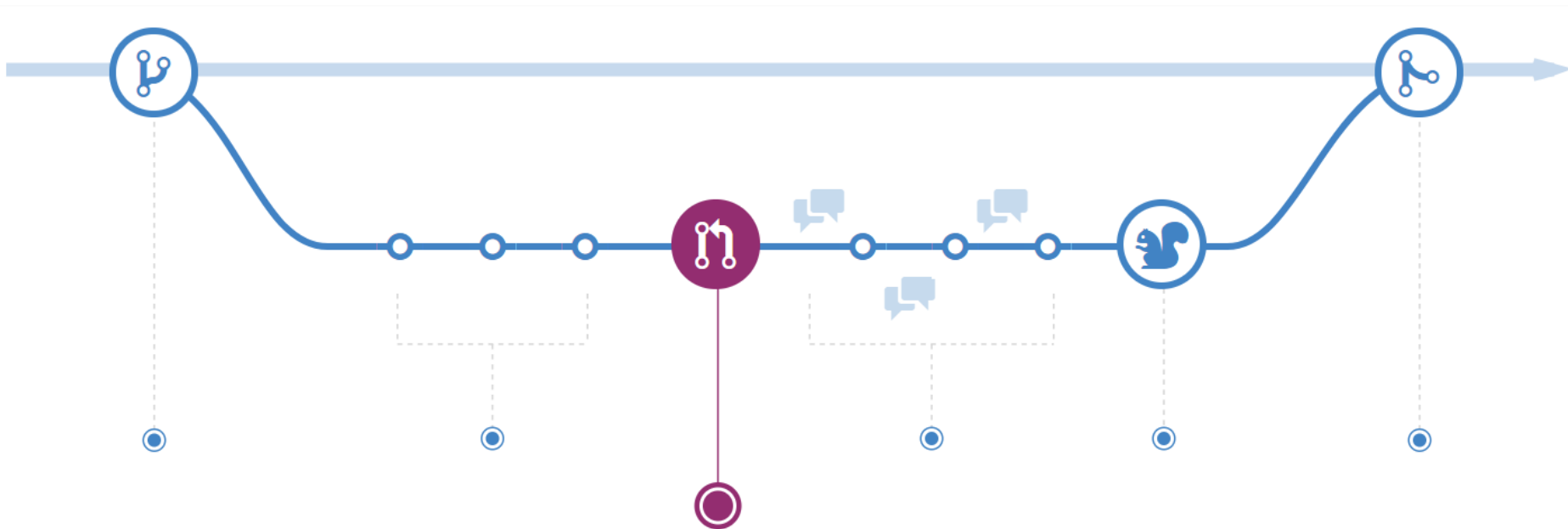
Create a branch



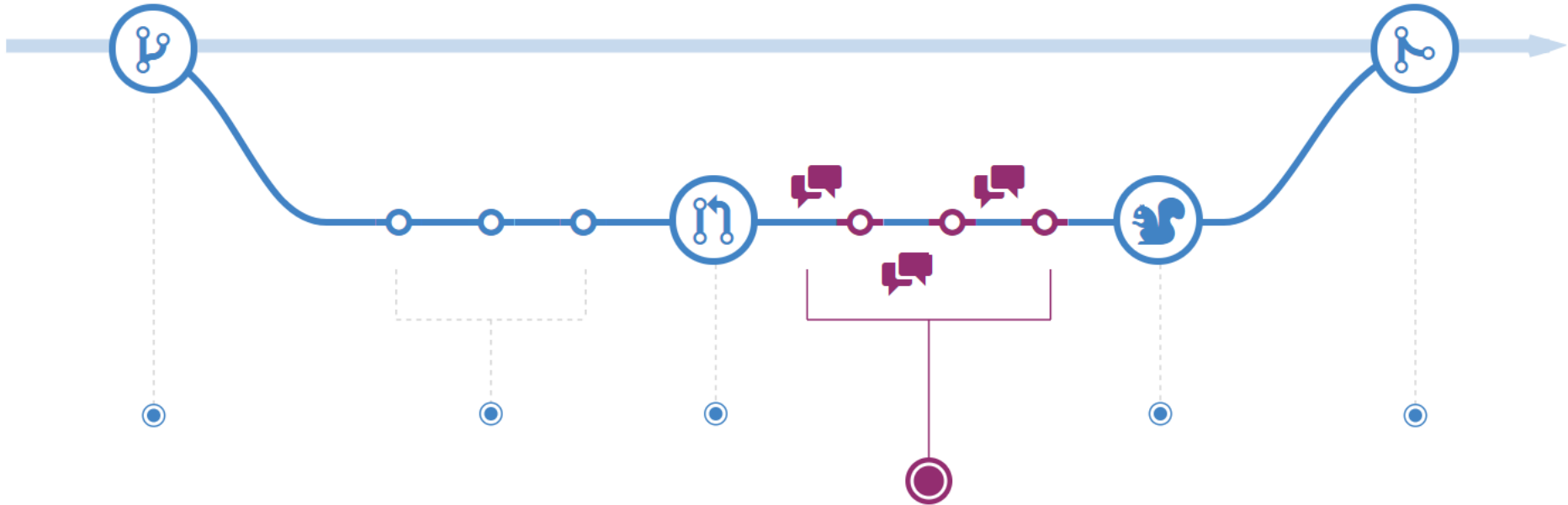
Add commits



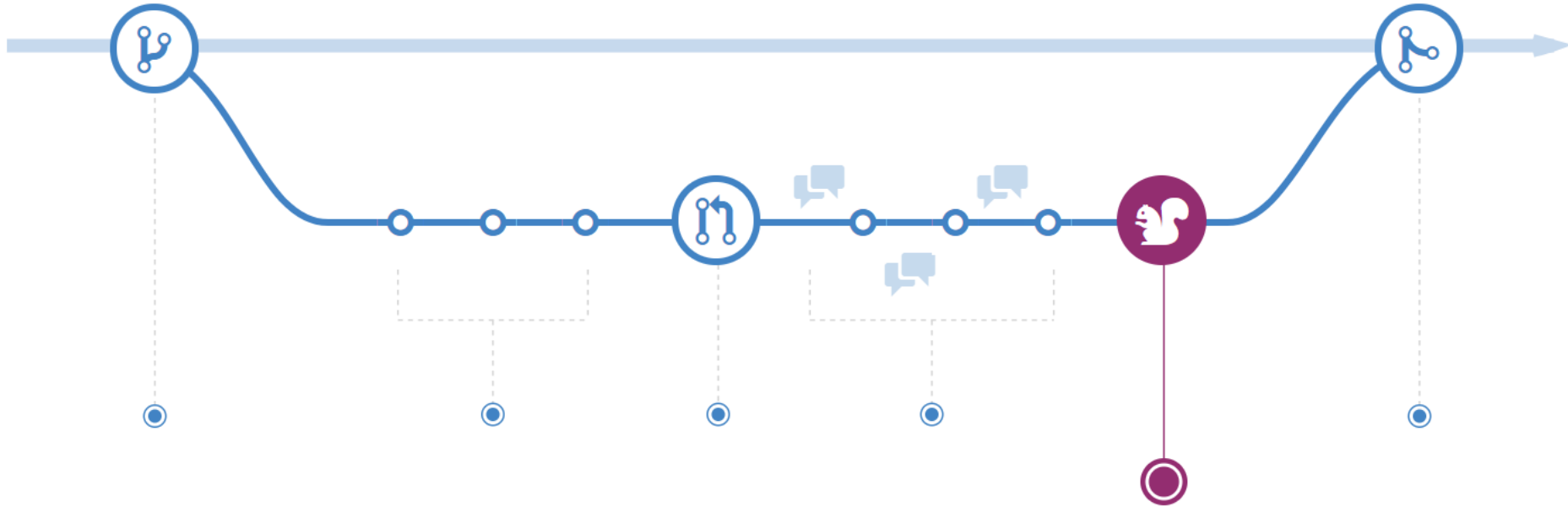
Open a Pull Request



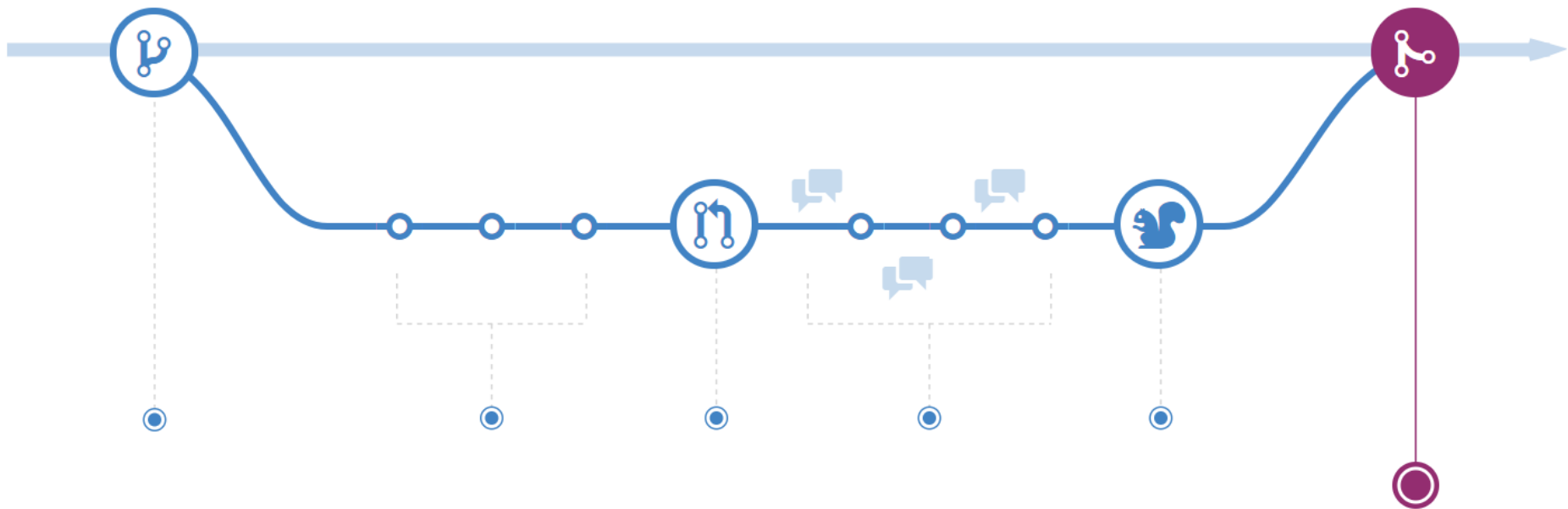
Discuss and review your code



Final testing



Merge



Single Responsibility Principle – does that ring a bell?

- Fix misspelling (separate, dedicated commit)
- Fix line ending to CR / LF (separate, dedicated commit)
- Little refactoring (separate, dedicated commit)
- Do commit early and often
- `git add --patch` to select parts of file what you need to commit

GIT UBER COMMIT

COUNTING OBJECTS: 500



	COMMENT	DATE
○	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
○	ENABLED CONFIG FILE PARSING	9 HOURS AGO
○	MISC BUGFIXES	5 HOURS AGO
○	CODE ADDITIONS/EDITS	4 HOURS AGO
○	MORE CODE	4 HOURS AGO
○	HERE HAVE CODE	4 HOURS AGO
○	AAAAAAA	3 HOURS AGO
○	ADKFJSLKDFJSDKLFJ	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAAAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT
MESSAGES GET LESS AND LESS INFORMATIVE.

How to Write a Git Commit Message

1. Use the imperative mood in the subject line, e.g. “Add ...”
2. Include issue tracker references
3. Limit the subject line to 50 characters
4. Capitalize the subject line
5. Do not end the subject line with a period
6. Use the body to explain what and why vs. how
7. Separate subject from body with a blank line
8. Wrap the body at 72 characters

<https://chris.beams.io/posts/git-commit/>

Dealing with passwords in git repositories

- Your config: `foobar.config` <- add to `.gitignore`
- Instead commit `foobar.config.example`, without sensitive info
- There are tricks to remove sensitive data from previous commits
<https://help.github.com/articles/removing-sensitive-data-from-a-repository/>

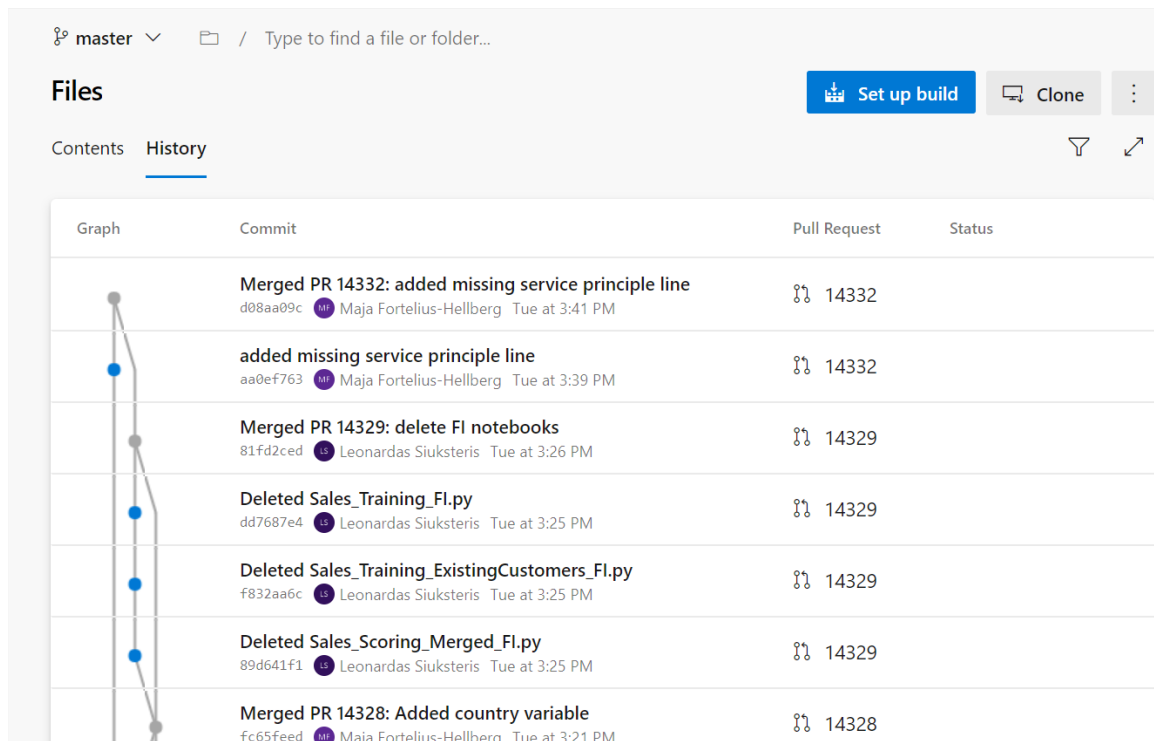
Markdown for Readme files

- # This is an <h1> tag
- ****This text will be bold****
- ![GitHub Logo](/images/logo.png)
- ```
```javascript
function fancyAlert(arg) {
  }
```
```
- - [ ] this is an incomplete item








<https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet>

# Git implementations – hosting your repos

- Azure DevOps Repos
- GitHub
- BitBucket



The screenshot displays the Azure DevOps Repos interface for the 'master' branch. The 'Files' tab is active, and the 'History' sub-tab is selected. The interface shows a commit history table with columns for Graph, Commit, Pull Request, and Status. The commit history is as follows:

| Graph                                                                               | Commit                                                                                                          | Pull Request | Status |
|-------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|--------------|--------|
|    | <b>Merged PR 14332: added missing service principle line</b><br>d08aa09c Maja Fortelius-Hellberg Tue at 3:41 PM | 14332        |        |
|    | <b>added missing service principle line</b><br>aa0ef763 Maja Fortelius-Hellberg Tue at 3:39 PM                  | 14332        |        |
|    | <b>Merged PR 14329: delete FI notebooks</b><br>81fd2ced Leonardas Siuksteris Tue at 3:26 PM                     | 14329        |        |
|    | <b>Deleted Sales_Training_FI.py</b><br>dd7687e4 Leonardas Siuksteris Tue at 3:25 PM                             | 14329        |        |
|    | <b>Deleted Sales_Training_ExistingCustomers_FI.py</b><br>f832aa6c Leonardas Siuksteris Tue at 3:25 PM           | 14329        |        |
|    | <b>Deleted Sales_Scoring_Merged_FI.py</b><br>89d641f1 Leonardas Siuksteris Tue at 3:25 PM                       | 14329        |        |
|  | <b>Merged PR 14328: Added country variable</b><br>fc65feed Maia Fortelius-Hellberg Tue at 3:21 PM               | 14328        |        |

# Tools – making commits + tree visualization

Up to developer:

- Git bash – command line
- Visual Studio
- Source Tree

The screenshot shows the SourceTree application interface. On the left is a sidebar with navigation options: File status, History (selected), Search, BRANCHES, BOOKMARKS, TAGS, REMOTES, SHELVED, and SUBREPOSITORIES. The main area displays a commit history table with columns for Commit, Author, Description, and Date. The commit history is visualized as a graph on the left side of the table. The current commit selected is b7358c7 by Rahul Chhabra, titled 'Removing ol...'. The graph shows a series of commits connected by lines, with colors indicating different branches or states.

| Commit  | Author           | Description                                                  | Date                 |
|---------|------------------|--------------------------------------------------------------|----------------------|
| b7358c7 | Rahul Chhabra... | Removing ol...                                               | Mar 3, 2016, 11:...  |
| bdb8bef | Rahul Chhabra... | Merged in update-google-verification (pull request #14)      | Feb 18, 2016, 1:3... |
| dfe975d | Tyler Tadej...   | Update google verificati...                                  | Feb 11, 2016, 2:2... |
| 3bc3290 | Tyler Tadej...   | Replace outdated Atlassian logo in footer with base-64 en... | Feb 11, 2016, 2:1... |
| dba47f9 | Tyler Tadej...   | Add gitignore                                                | Feb 11, 2016, 1:3... |
| ff67b45 | Mike Minns...    | Updated Mac min-spec to 10.10                                | Feb 15, 2016, 11:... |
| 72d32a8 | Michael Min...   | Merged in hero_images (pull request #13)                     | Feb 15, 2016, 10:... |
| 246c4ff | Joel Unger...    | Used TinyPNG to c...                                         | Feb 11, 2016, 3:3... |
| 9d9438c | Joel Unger...    | Replacing hero images with new version of SourceTree         | Feb 9, 2016, 2:59... |
| ce75b3d | Michael Min...   | Merged in bug/date-https (pull request #12)                  | Feb 15, 2016, 10:... |
| 85367bb | Patrick Tho...   | fixed date and https errors                                  | Jan 7, 2016, 12:2... |
| 4f9b557 | Joel Unger...    | New Favicon                                                  | Feb 8, 2016, 3:55... |
| 384e6d5 | Rahul Chhabra... | search console google ver...                                 | Feb 3, 2016, 2:09... |
| 6fa47a9 | Mike Minns...    | updated to move supported version to OSX 10.9+               | Dec 15, 2015, 2:0... |
| 8dd87bb | Mike Minns...    | remove extra , when a line is skipped due to empty server    | Nov 23, 2015, 2:2... |
| faa195e | Mike Minns...    | Skip records with empty server/user id as gas rejects them   | Nov 23, 2015, 2:1... |
| 0cdf96  | Mike Minns...    | corrected paths after merge                                  | Nov 23, 2015, 2:0... |
| 051ab1b | Mike Minns...    | corrected column counting                                    | Nov 23, 2015, 1:5... |
| a723bc2 | Mike Minns...    | Merge branch 'au2gex'                                        | Nov 23, 2015, 1:5... |

The screenshot shows the Git GUI application interface. At the top, it displays the repository name 'common-OASIS: --all - gitk' and a menu bar with File, Edit, View, and Help. The main area shows a commit history graph with branches 'master' and 'remotes/origin/master' highlighted. The current commit selected is 'Merged PR 13823: Clean up'. Below the graph, the SHA1 ID is displayed as '2c8052c136dfc5cb93d25e1000b756f5dc166a3c'. The diff view shows the changes for the selected commit, including updates to 'meta logging enabled', 'user info logging enabled', 'Metalog update NotebookInfo class', and 'Deleted tutorial/git-demo'. The diff view also shows the commit message 'Merged PR 13823: Clean up' and the commit details for the parent and child commits.

SHA1 ID: 2c8052c136dfc5cb93d25e1000b756f5dc166a3c

Find: commit containing: Search

Diff: ☒ Diff ☐ Old version ☐ New version Lines of context: 3 Ignore space change Line diff

Author: Leonardas Siuksteris <leonardas.siuksteris@if.se> 2020-02-03 08:44:59  
Committer: Leonardas Siuksteris <leonardas.siuksteris@if.se> 2020-02-03 08:44:59  
Parent: 5f9a7940e44648aa90495c259edbe87800361dd7 (Merged PR 13822: Updated set-dbr-sp.py)  
Parent: e0f9c8cdeb8e9edf726c1e85f28c2620ee620a3 (Deleted tutorial/git-demo)  
Child: 9ad6ae96bbf02ccfd6f5f5f98baa0a89bf2a14c (Merged PR 13830: detailed meta logging enabled)  
Child: 350f2720fb33dccc35ca09d34eb21e896c85286 (Metalog update NotebookInfo class)

Branches: master, remotes/origin/master  
Follows:  
Precedes:

Merged PR 13823: Clean up

master



/ Type to find a file or folder...

## Files

Set up build

Clone



Contents

History



| Graph | Commit                                                                                                           | Pull Request | Status |
|-------|------------------------------------------------------------------------------------------------------------------|--------------|--------|
|       | <b>Merged PR 14332: added missing service principle line</b><br>d08aa09c  Maja Fortelius-Hellberg Tue at 3:41 PM | 14332        |        |
|       | <b>added missing service principle line</b><br>aa0ef763  Maja Fortelius-Hellberg Tue at 3:39 PM                  | 14332        |        |
|       | <b>Merged PR 14329: delete FI notebooks</b><br>81fd2ced  Leonardas Siuksteris Tue at 3:26 PM                     | 14329        |        |
|       | <b>Deleted Sales_Training_FI.py</b><br>dd7687e4  Leonardas Siuksteris Tue at 3:25 PM                             | 14329        |        |
|       | <b>Deleted Sales_Training_ExistingCustomers_FI.py</b><br>f832aa6c  Leonardas Siuksteris Tue at 3:25 PM           | 14329        |        |
|       | <b>Deleted Sales_Scoring_Merged_FI.py</b><br>89d641f1  Leonardas Siuksteris Tue at 3:25 PM                       | 14329        |        |
|       | <b>Merged PR 14328: Added country variable</b><br>fc65feed  Maja Fortelius-Hellberg Tue at 3:21 PM               | 14328        |        |



# Azure DevOps – Pull requests

14332 **COMPLETED** added missing service principle line

MF Maja Fortelius-Hellberg [aof\\_add\\_spline](#) into [master](#)

Overview Files Updates Commits Conflicts

Leonardas Siuksteris completed the pull request on 2/11/2020 2:41 PM.

Cherry-pick Revert

d08aa09c Merged PR 14332: added missing service principle line...

Description

added missing service principle line to abandoned offerform notebook

Show everything

Add a comment...

Leonardas Siuksteris completed the pull request 2/11/2020

Approved by Leonardas Siuksteris 2/11/2020

LS

Policies

Required  
 1 reviewer approved

Status  
 Metadata deployment succeeded

Work Items

No related work items

Reviewers

Leonardas Siuksteris Approved

Jussi Liimatainen Admin

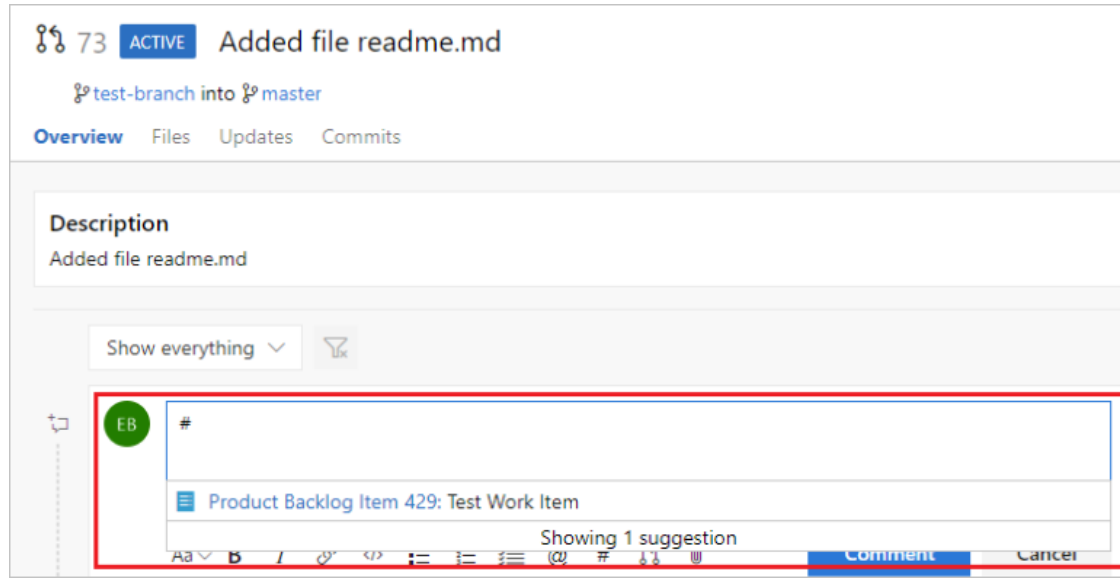
Tomas Lukas Komar

# Azure DevOps – Link tasks and user stories

<https://docs.microsoft.com/en-us/azure/devops/notifications/add-links-to-work-items?view=azure-devops>

Use the **#ID** control within the following areas:

- A work item discussion or any rich-text field
- A pull request discussion
- Commit comments
- Changeset or shelveset comments
- Wiki page.



# AzureDevOps - Pipelines

The screenshot displays the Azure DevOps Pipelines interface for a release named "Release-85". The breadcrumb navigation shows "Metadata deployment" and "Release-85". The top navigation bar includes tabs for "Pipeline", "Variables", and "History", along with action buttons: "+ Deploy", "Cancel", "Refresh", "Edit", and a menu icon. The main content area is divided into two panels: "Release" and "Stages".

**Release Panel:**

- Pull request deployment:** Triggered for pull request #14332 by Maja Fortelius-Hellbom on 2/11/2020 at 3:40 PM.
- Artifacts:** A build artifact named "\_common-OASIS" is shown with a file icon, a lock icon, and a Git icon. The artifact ID is "12683f6c4e4ff4495bfa6c803e1f74b83a7d64b7" and it is located at "refs/pull/14332/merge".

**Stages Panel:**

- Deploy metadata:** A stage task that has "Succeeded" (indicated by a green checkmark) on 2/11/2020 at 3:41 PM.

A line connects the artifact from the Release panel to the Deploy metadata stage in the Stages panel, indicating that the stage uses the artifact.

# Azure DevOps Repos - limitations

- Repositories should generally be no larger than 10GB
- Pushes are limited to 5GB at a time