

AZ-400 AZURE DEVOPS



Saleh Elnaggar



linkedin.com/in/saleh-elnggar
salehelnaggar.live
saleh.elnggar@gmail.com

AZ-400 AZURE DEVOPS

- COURSE STRUCTURED
- WHAT IS DEVOPS?
- SOURCE CONTROL AND VERSION CONTROL
- CONTINUOUS INTEGRATION
- CONTINUOUS DELIVERY
- IMPLEMENTING INFRASTRUCTURE

About the certificate

Certification details

Complete one prerequisite



PREREQUISITE OPTION 1

Microsoft Certified: Azure Administrator Associate

OR



PREREQUISITE OPTION 2

Microsoft Certified: Azure Developer Associate

Take one exam



CERTIFICATION EXAM

Designing and Implementing Microsoft DevOps Solutions

Earn the certification



EXPERT CERTIFICATION

Microsoft Certified: DevOps Engineer Expert

Pre-requisite “requirement”

Familiar with Azure common services “Azure vm, vmss, azure web apps”

Simple knowledge on any development framework

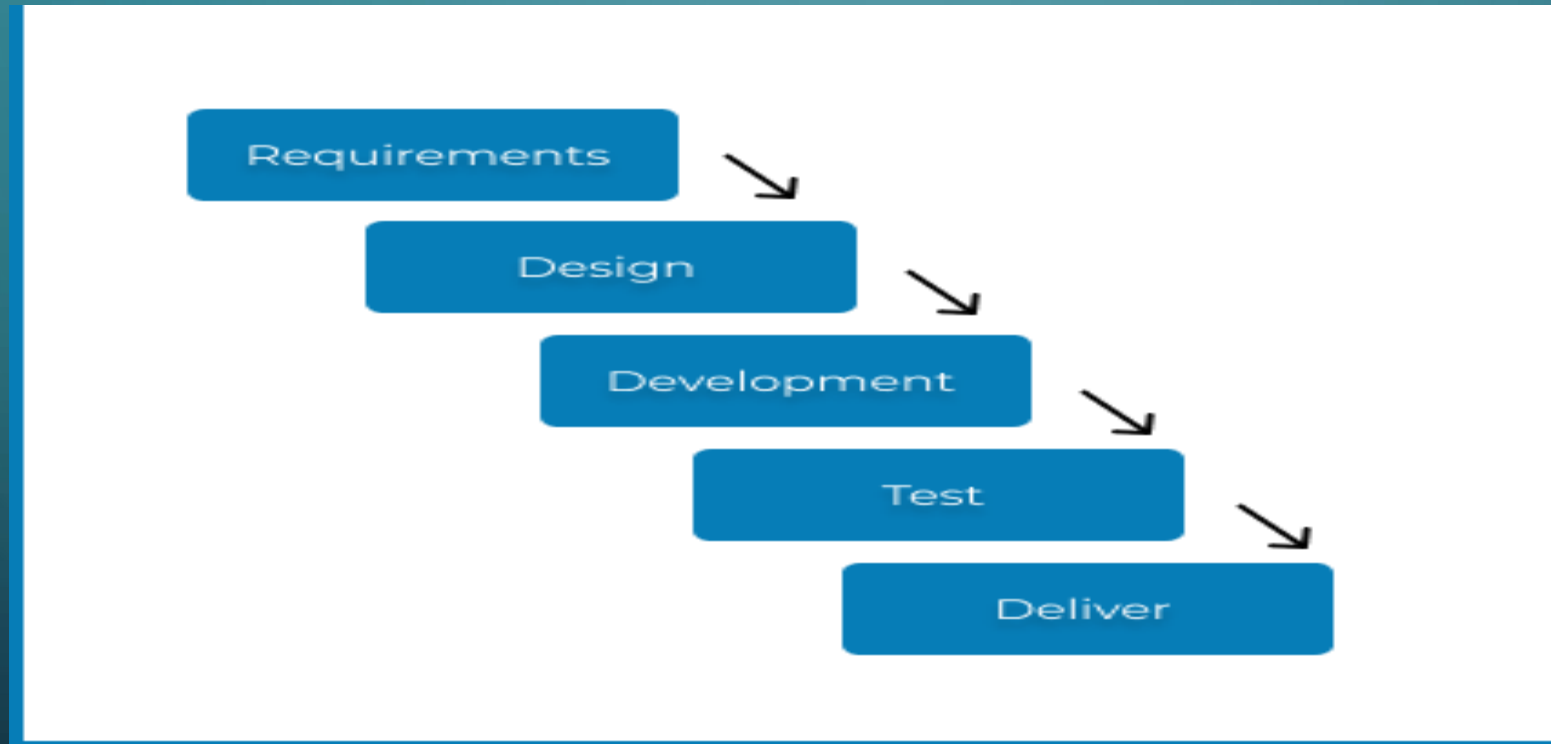
Knowledge on how applications are deployed

What do you need?

- An Azure account with Azure subscription
- An Azure DevOps account – easy to create

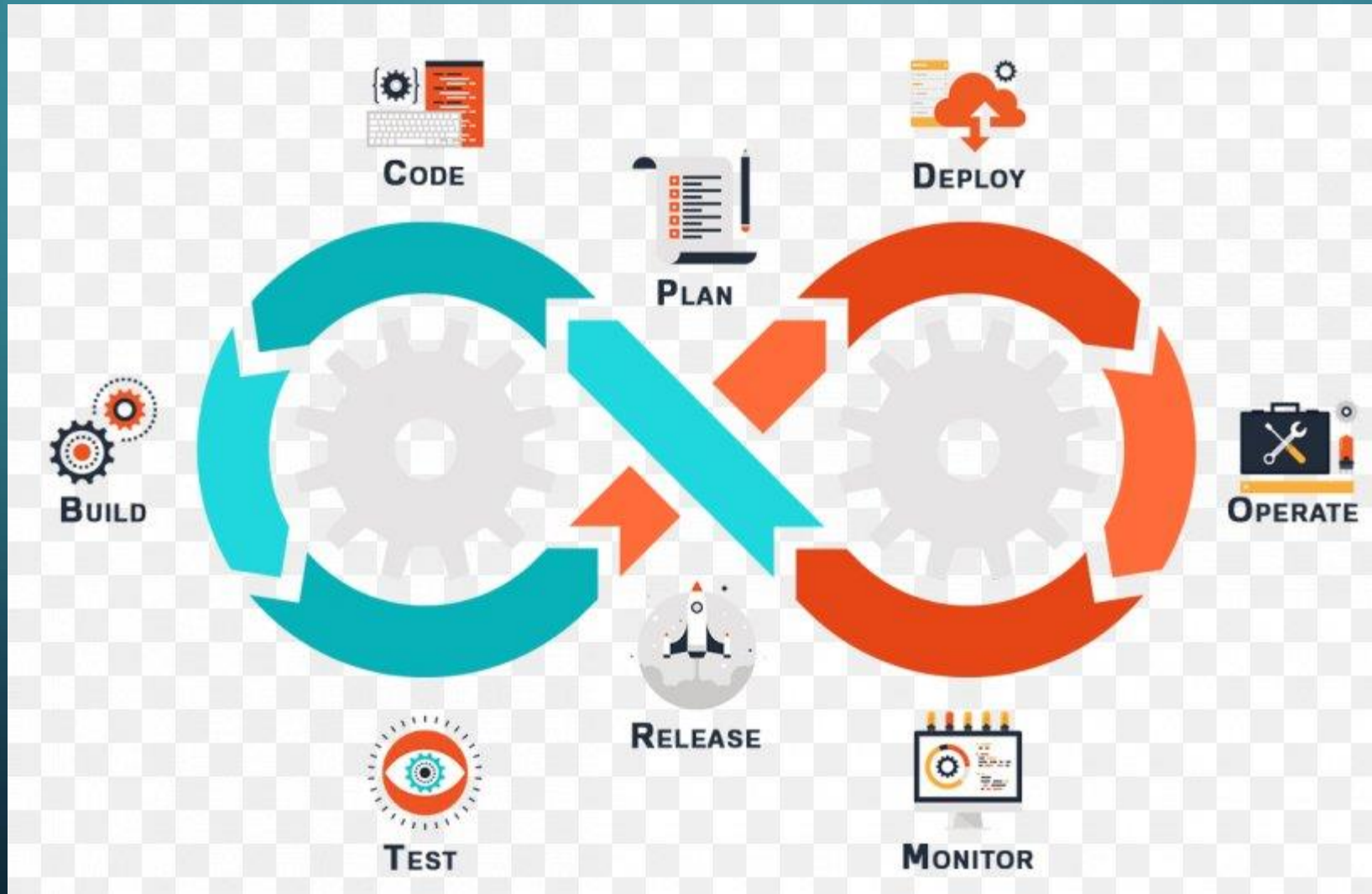
Before DevOps concept

- Traditional Project lifecycle



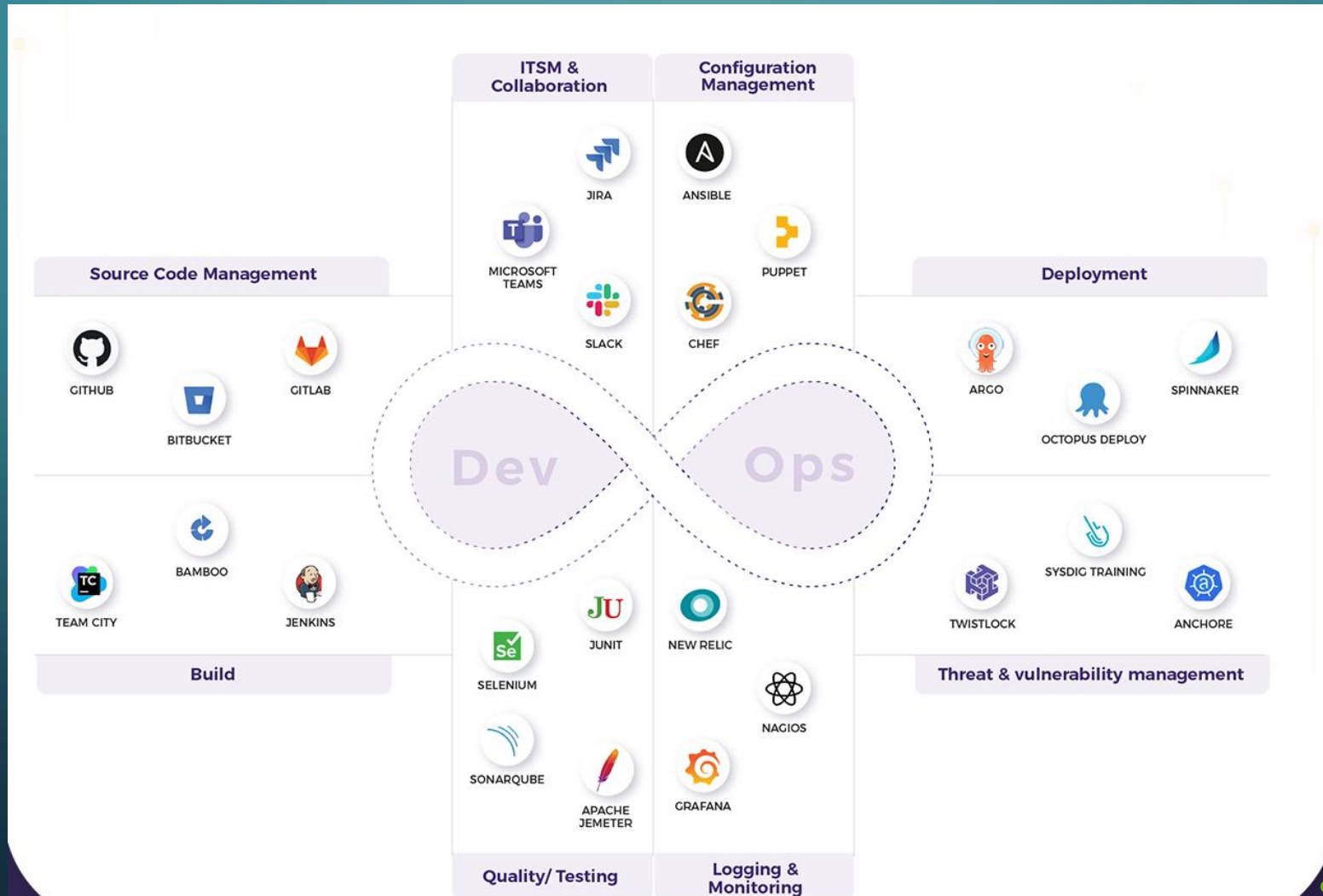
Why DevOps?

- Agile Project lifecycle



Why DevOps?

- DevOps tools



Azure DevOps tools



Azure Boards

Plan, track, and discuss work across teams, deliver value to your users faster.



Azure Repos

Unlimited cloud-hosted private Git repos. Collaborative pull requests, advanced file management, and more.



Azure Pipelines

CI/CD that works with any language, platform, and cloud. Connect to GitHub or any Git provider and deploy continuously to any cloud.



Azure Test Plans

The test management and exploratory testing toolkit that lets you ship with confidence.



Azure Artifacts

Create, host, and share packages. Easily add artifacts to CI/CD pipelines.

Azure DevOps services pricing

Basic Plan



First 5 users free,
then \$6 per user per month

Start free

- **Azure Pipelines:** Includes the free offer from INDIVIDUAL SERVICES
- **Azure Boards:** Work item tracking and Kanban boards
- **Azure Repos:** Unlimited private Git repos
- **Azure Artifacts:** 2 GiB free per organization

First 5 users free

Basic + Test Plans



\$52 per user
per month

30 day free trial

- Includes all Basic plan features
- Test planning, tracking & execution
- Browser-based tests with annotation
- Rich-client test execution
- User acceptance testing
- Centralized reporting

[more details](#)

salehelnaggar.live







Azure Boards

PRODUCT BACKLOG



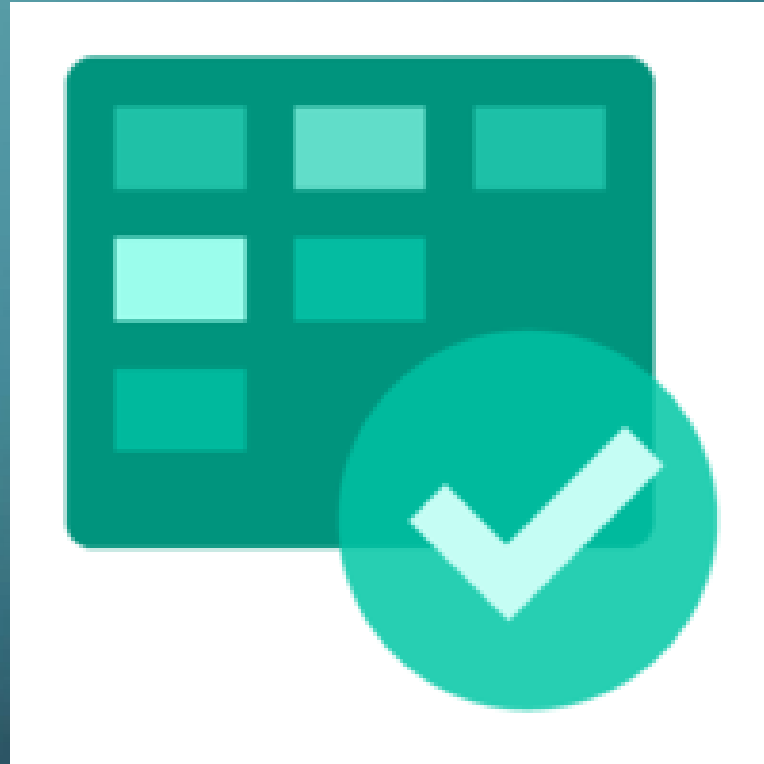
Azure Boards

- Sprint

Backlog	In Progress (3)	Peer Review (3)	In Test (1)	Done	Blocked
					
Fast Track/ Defect					

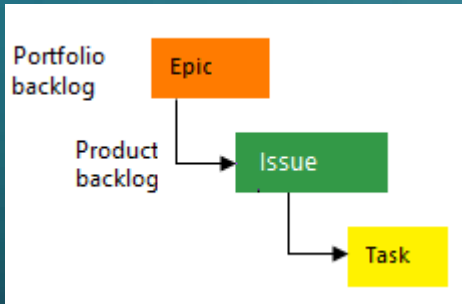
Azure Boards

Demo lab

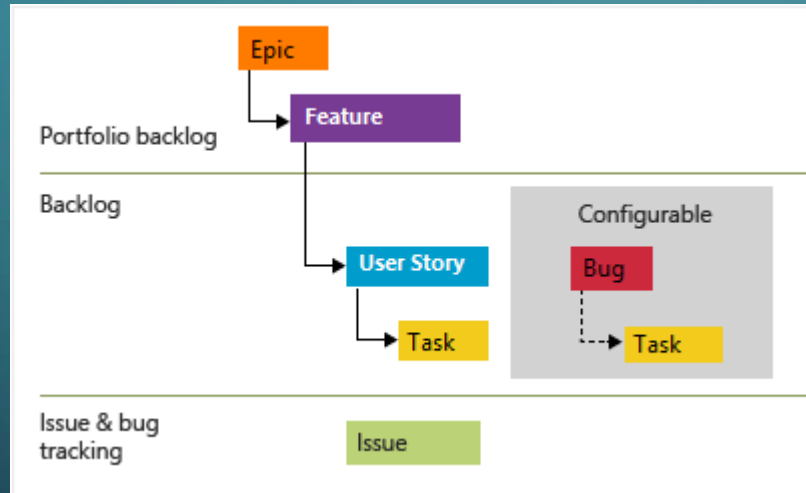


Azure Boards – Project types

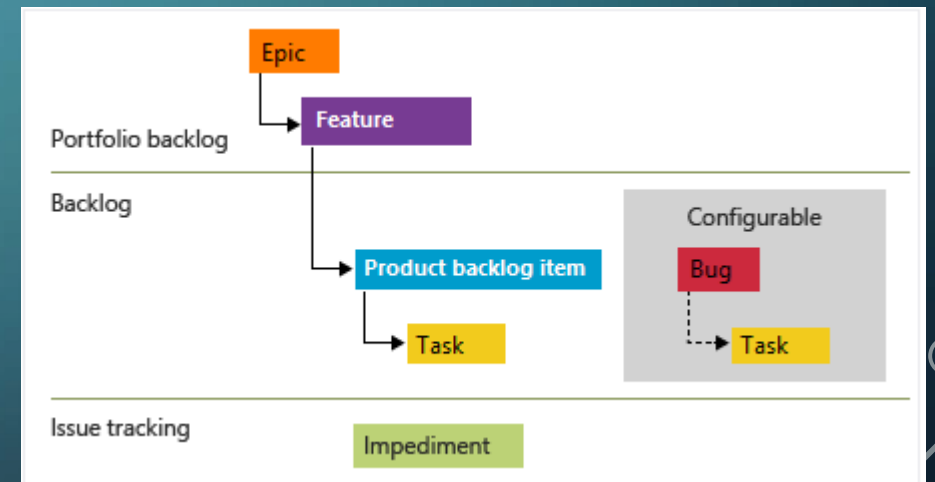
Basic



Agile



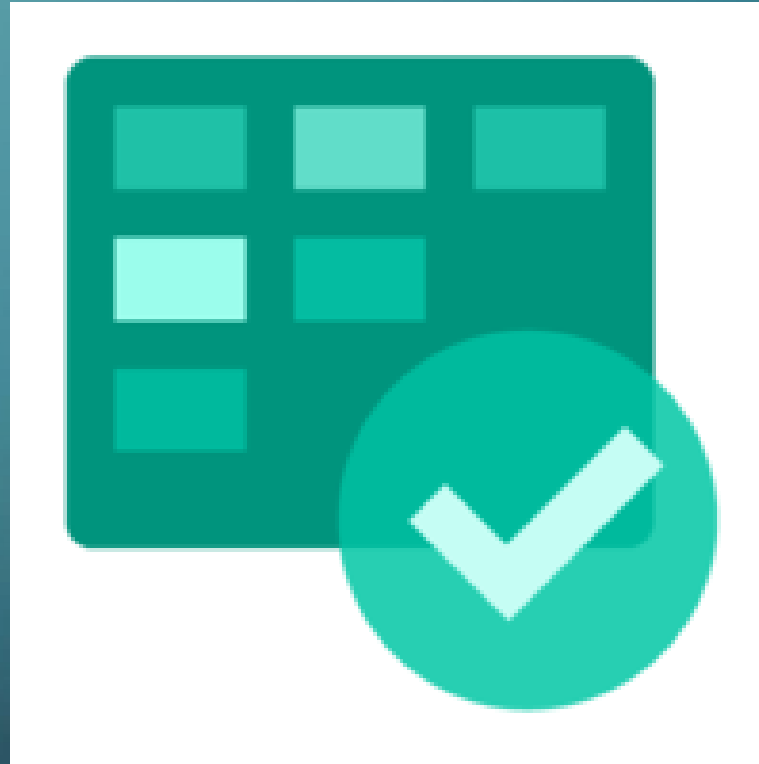
Scrum



[more details](#)

Azure Boards – use sprint

Demo lab



Azure Boards – integration with slack

Demo lab



Azure Boards – Azure AD integration

Demo lab



Azure Boards – add users to project

Demo lab



Azure Boards – different charts



Burndown

Displays burndown across multiple teams and multiple sprints. Create a release burndown or bug burndown.

Focus on the remaining work within the specified period of time



Burnup

Displays burnup across multiple teams and multiple sprints. Create a release burnup or bug burnup.

Focuses on the completed work



Chart for Work Items

Visualize work items like bugs, user stories, and features using shared work item queries.

Are we on track to complete the set of work by the end date



Cumulative Flow Diagram (CFD)

Visualize the flow of work and identify bottlenecks in the software development process.

This helps to see the items as they move through the different states

Azure Boards – different charts



Cycle Time

Visualize and analyze your team's cycle time using a control chart.

Measures the time taken for the team to complete work items once they have begun actively working on them



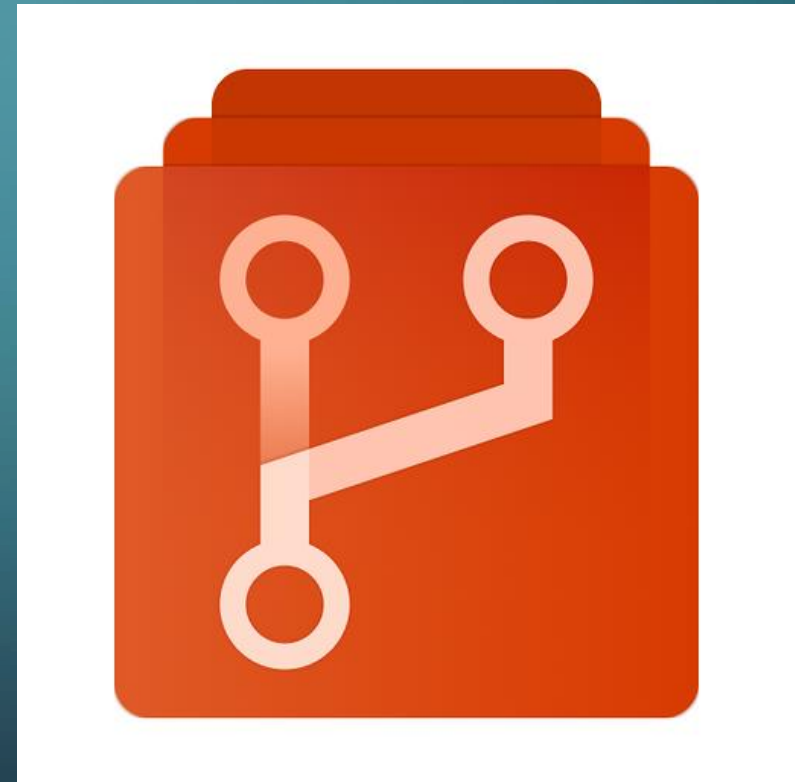
Lead Time

Visualize and analyze your team's lead time using a control chart.

Measures the total time elapsed from the creation of work items to their completion

Source code tool – version control

- What is Git?
- Azure repos



Source code tool – version control

- Version control categories:

- Centralized system

- Subversion control
 - Team foundation

- Decentralized system

- git

Source code tool – version control

- Version control categories:
 - Centralized system
 - Subversion control
 - Team foundation



Source code tool – version control

- Decentralized system
 - git

		Git Version 1	Git Version 2
	FileA	Version 1	Version 1
	FileB	Version 1	Version 2
	FileC	Version 1	Version 2

Git

Demo lab

1. Install git.
2. Initialize an empty repository.
3. Playing with git locally.



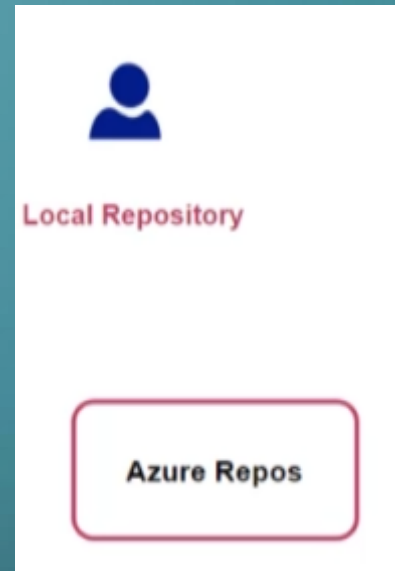
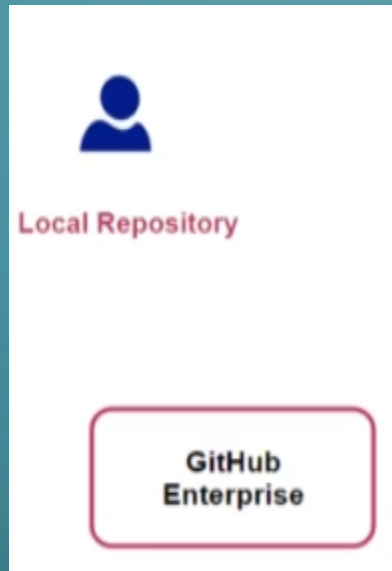
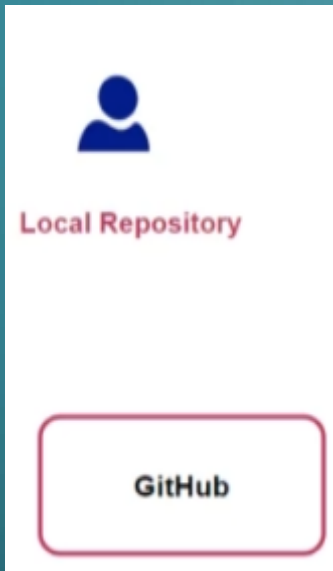
Git

Demo lab

1. Making changes to your files
2. Go back to previous commit.



Central git repository



Using GitHub

Demo lab

1. Create new repo
2. Add remote repo to local repo.



Using GitHub

Demo lab

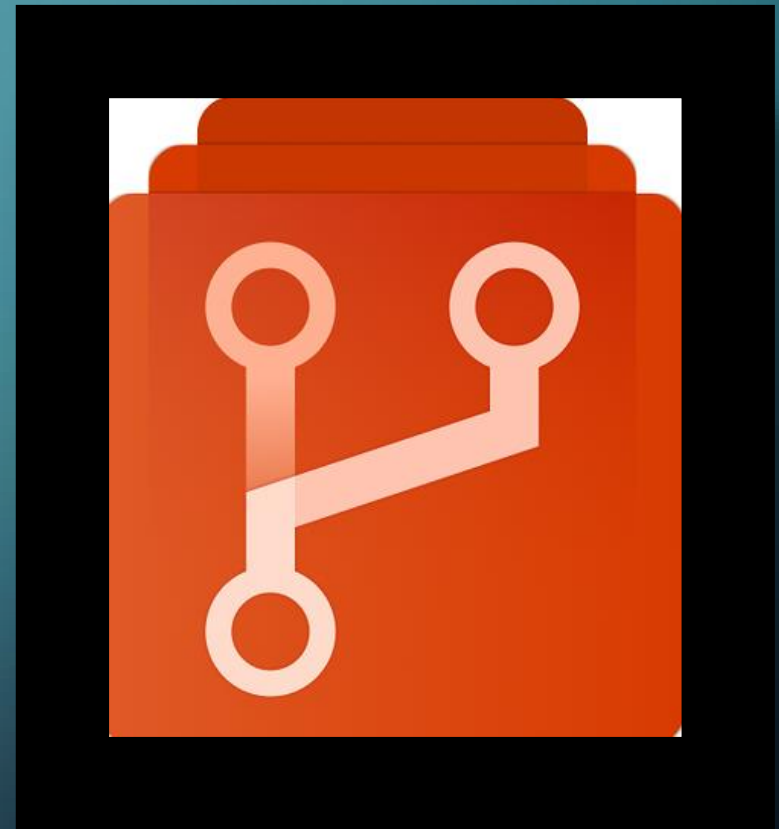
1. Make changes on repo locally
2. See the different pointers.
3. Check it in the remote repo.



Azure Repos

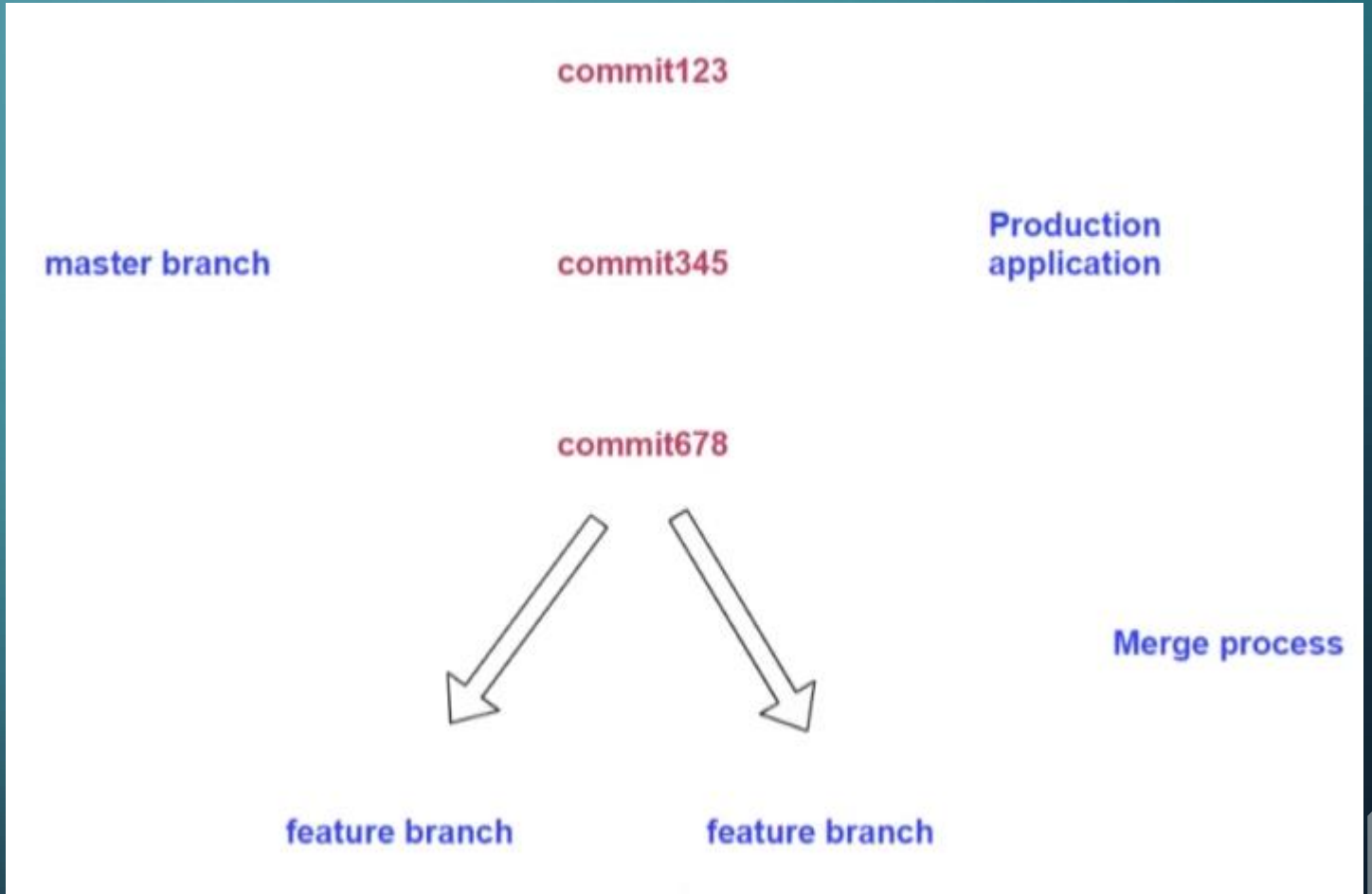
Demo lab

1. Check the default repo.
2. Create new repo.
3. Add Azure repo to local repo.
4. Make changes locally and push it.



Understanding branches

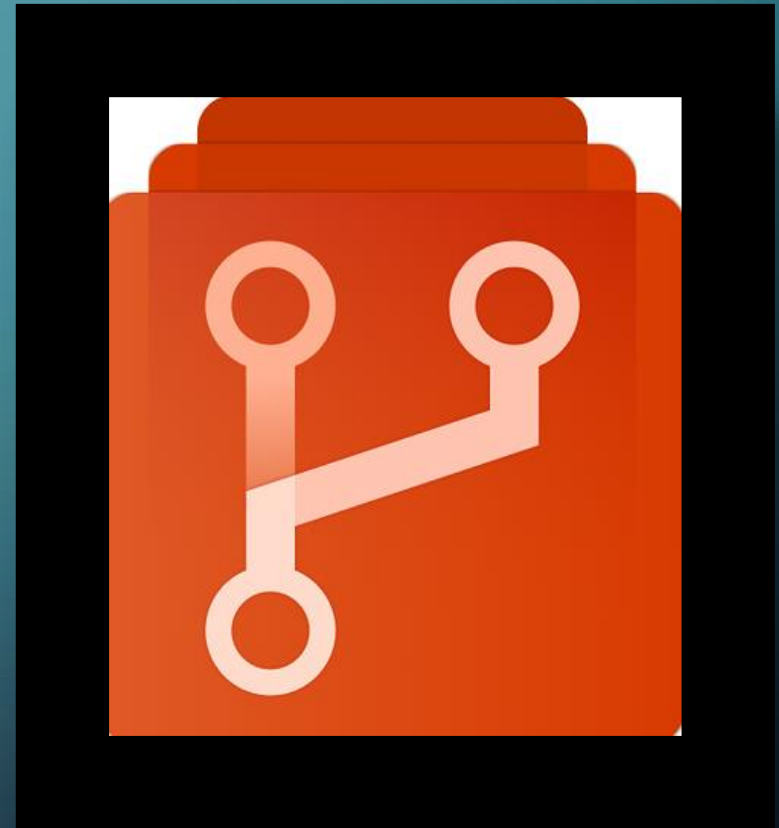
- Good practices:
 - Create many short feature branches
 - Delete once they are not required.



Branches

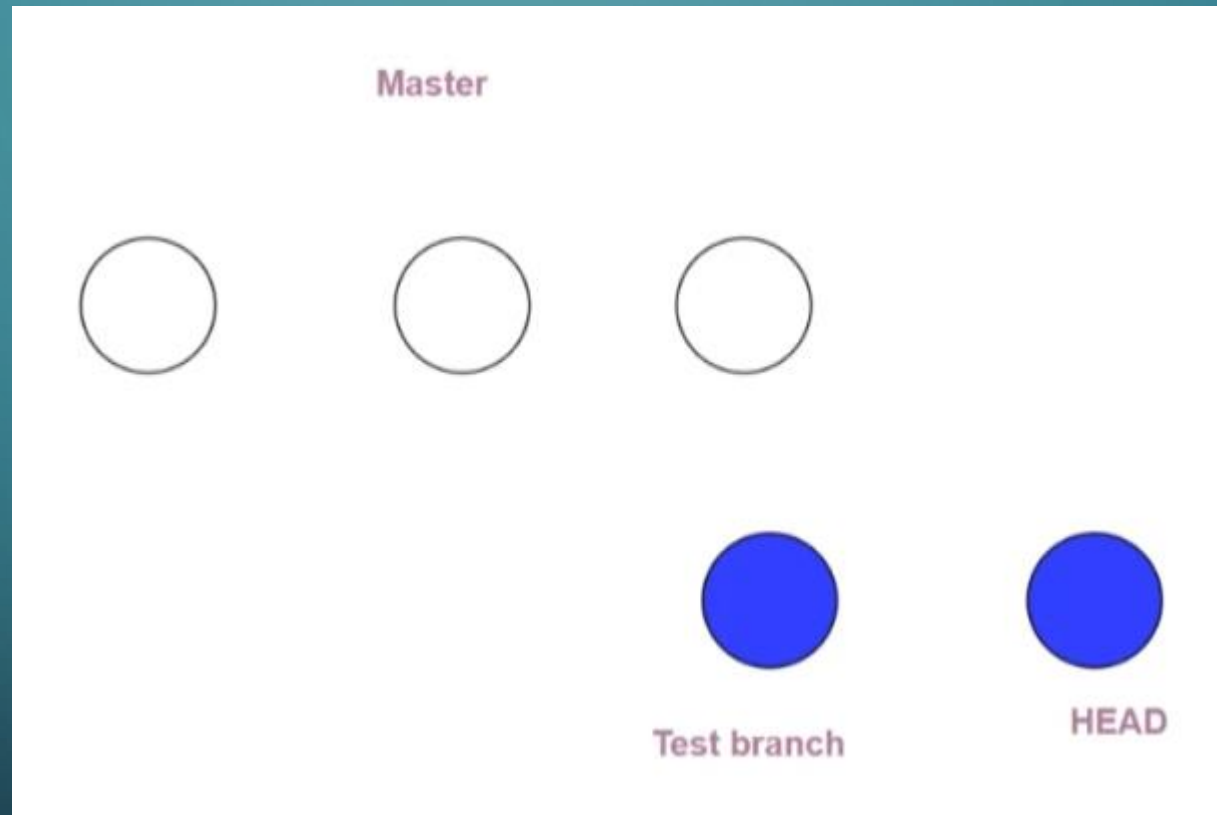
Demo lab

1. Show all branches.
2. Create new branch.
3. Work with the new branch.



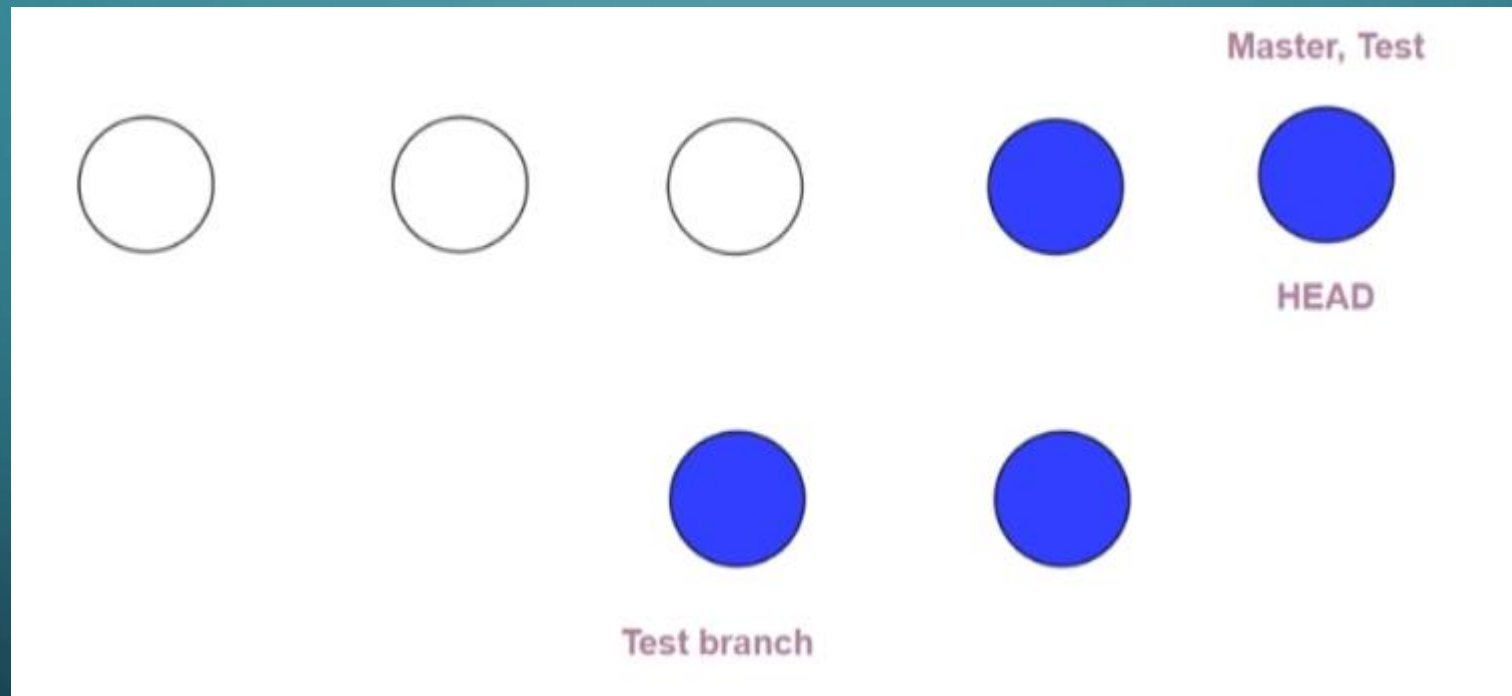
Merges in git

Implicit Merge "Fast forward merge"



Merges in git

Implicit Merge "Fast forward merge"



Fast forward merge

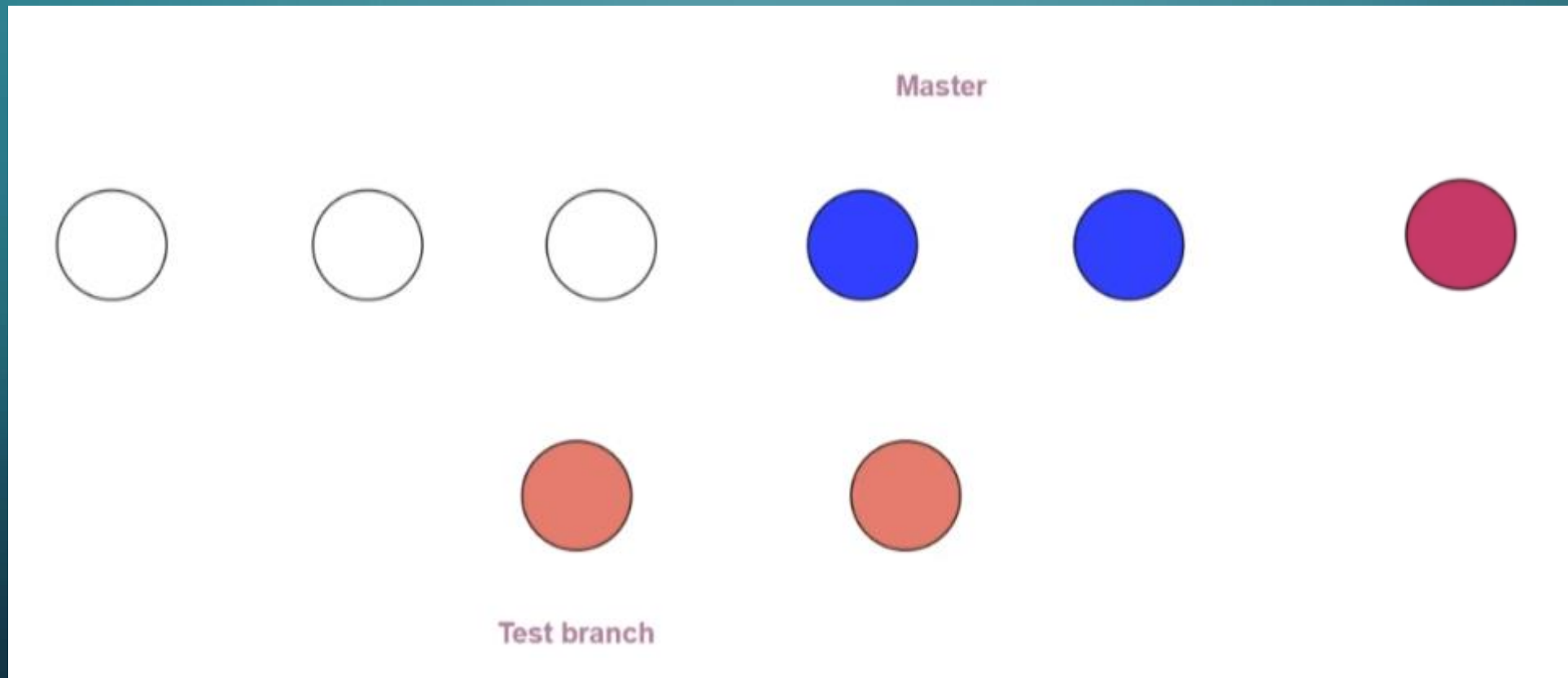
Demo lab

1. Do fast forward merge
2. Check the pointer



Merges in git

Recursive merge "3-way merge"



Recursive merge

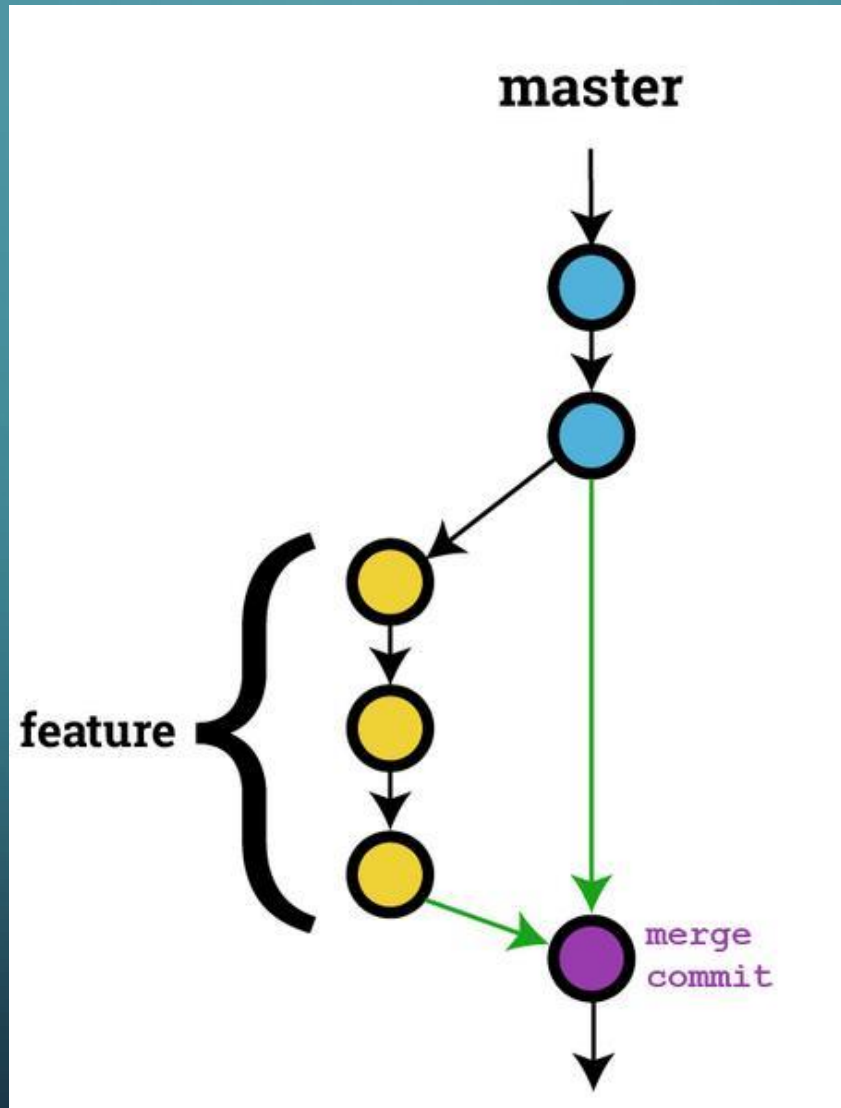
Demo lab

1. Make changes in the main.
2. Add new file to feature branches.
3. Try to do merge “recursive merge”.



Merges in git

Squash merge



Squash merge

Demo lab

1. Make more than two commit in the feature.
2. Try to do merge “squash merge”.



Conflict in the merge

Demo lab

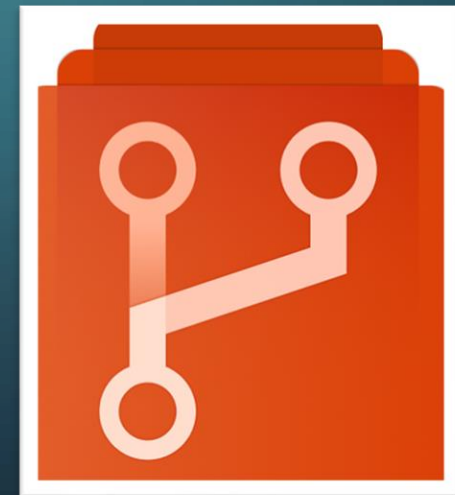
1. Make changes in the main.
2. Make changes in the feature.
3. Try to make merge.
4. Solve the conflict.
5. How to avoid that in the real live?



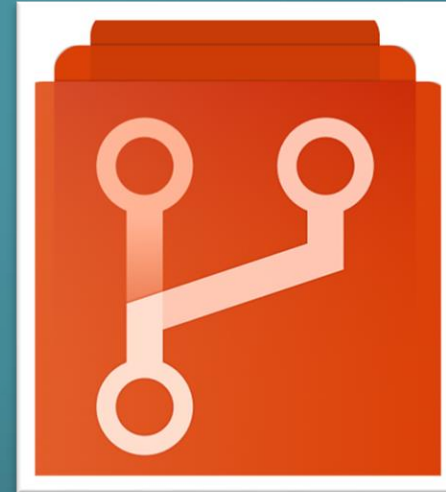
Pushing branches

Demo lab

1. Check the repo branches
2. Use the git push command “push the main”
3. Check again your branches.
4. Push the new branch also.
 1. `git push --all origin`
 2. `git push --u origin feature`



Pull requests



Pull request

Demo lab

1. Enable any of branch policies
2. Make changes in the new branch
3. Merge to main locally
4. Try to push to main branch
5. Go to pull requests
 1. Create new pull request
 2. Approve and complete reviewing the changes



Pulling changes from the repo

Demo lab

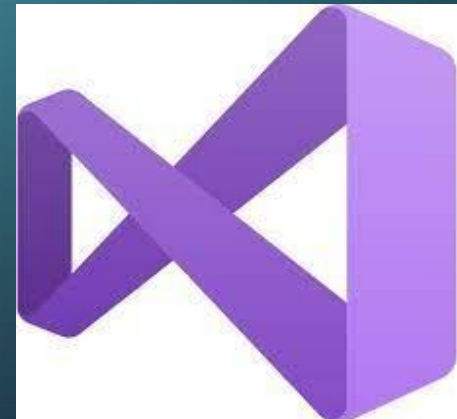
1. Make change in the remote repo.
2. If the developer in the local git try to push after some modification in the same file!
3. His push will rejected.
4. Need to make pull first from the remote repo.



GitHub with Visual Studio

Demo lab

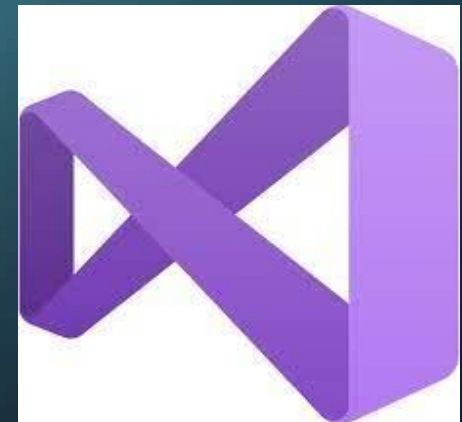
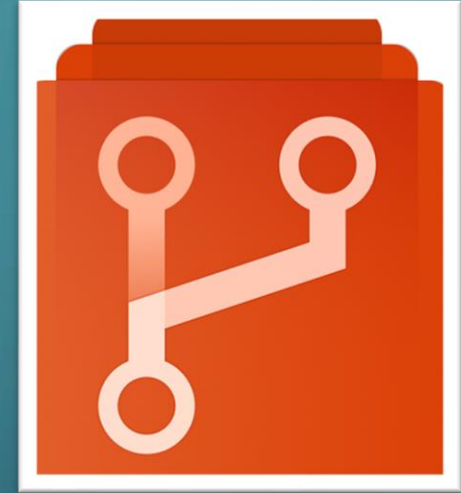
1. Create new .NET core project.
2. Make sure that the source control in your VS is git.
3. Create new GitHub repo.
4. Make changes on the code and commit and push it.
5. Check the changes on GitHub.



Azure repos with Visual Studio

Demo lab

1. Create new .NET core project.
2. Make sure that the source control in your VS is git
3. Add Azure repo to your project.
4. Push the code to Azure repo.
5. Make changes and check it remotely.



Git – .gitignore file

Demo lab

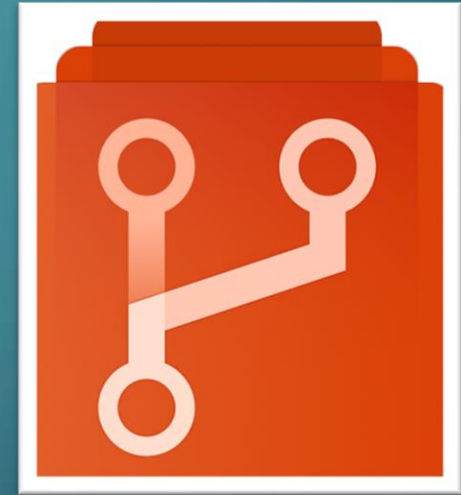
1. Make a new folder to try gitignore.
2. Create more than 3 files
3. Add which files you need to .gitignore
 1. Manually
 2. echo command “echo yourFile >> .gitignore”



Team foundation version control with Visual Studio

Demo lab

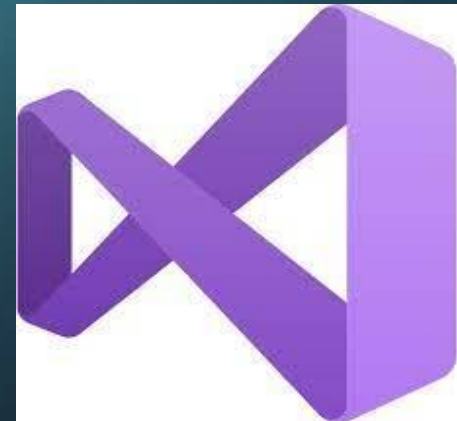
1. Create new repo with TFVC
2. Manage connection to browse your repos
3. Choose your TFVC repo and map & Get
4. Add your project to source control.
5. Check in your project to the TFVC repo.
6. Check out for edit and check in again.



Team foundation version control with Visual Studio

Demo lab

1. Create new repo with TFVC
2. Manage connection to browse your repos
3. Choose your TFVC repo and map & Get
4. Add your project to source control.
5. Check in your project to the TFVC repo.
6. Check out for edit and check in again.



Integration GitHub with Azure Boards

Demo lab

1. Make GitHub connection
2. Using the specific keyword which integrate with Azure Boards “Fixed AB#taskNumber”

