Azure DevOps

* **Continuous Integration:** integrating small bites of code at a time by running a local build and then a build on CI server that combines changes and runs several tests to ensure bug free code before being deployed to production. CI server builds are run on the main or master branch. CI builds can be automated on check-in or manually triggered.

e.g Jenkins

* **Continuous delivery:** ensure that builds are able to run on production at any time
* **Continuous deployment:** every change to the main automatically gets pushed to production. Many deployments per day! Using a deployment pipeline

Benefits of Continuous Delivery

* Small changes so no risk of major issues and better control of bug if any
* Improved project tracking
* Early feedback from stakeholders on deployed changes – if deployments are made continuously
* Artifacts are executables containing all resources required to run the application. A build generates an artifact.

Code testing

* Unit tests: testing pieces of code – function/method
* Integration tests: testing interoperations between functions or classes
* Smoke test: ensures functions work in general
* Regression test: test changes that were recently made. Did it fix the issue or introduce more bugs
* Acceptance test: Customers accept changes
* System test: treat entire application as a black box

Distributed Build

* Using multiple servers in the build pipeline. Offload workload or jobs to lessen risk of serve failures etc. Some servers can run the builds, while others run tests and deployments.

3 steps to a pipeline.

* Build
* Test
* Deploy

How Azure DevOps fits into CI/CD

* 5 components
  + Azure repos – SCM/VCS
  + Azure pipeline – build, test and deploy pipeline
  + Azure boards – track project tasks etc
  + Azure test plans (from a project management perspective, UAT, exploratory testing etc)
  + Azure artifacts: sharing code via Nuget packages etc

Azure Pipelines As Code

* Save pipeline code/script along with the project code or in separate project in SCM
* 2 ways to set up an azure pipeline
  + Visual builder
  + YAML files (can be checked in with code)

Create a pipeline from the CLI

* Create a YAML pipeline “az pipelines create”
* List and show build agents “az pipelines agent”
* Run a build “az pipelines run”
* Create a release “az pipelines release create”

Pipeline templates

* Used to quickly create pipelines for build and release

YAML

* Multi steps via stages with its own set of tasks
  + E.g Build stage then Deploy stage
* YAML code can be stored and referenced as template as part of the major pipeline YAML file.