**Microsoft Azure**

**CLOUD COMPUTING SERVICES PROVIDED BY AZURE AS WELL AS GENERAL INFORMATION**

* Cloud computing platform
* Cloud computing is delivery of computing services over the internet I.e. virtual machines, databases
* Can add additional computing power to an application as well as storage
* User of cloud is responsible for the data and information stored in the cloud, as well as security measures
* What needs to be done to maintain a cloud depends on what the cloud is being used for
* Cloud models define the deployment type of cloud resources
* Three main kinds are private, public, and hybrid
* A private cloud is a cloud that is used by a single entity
* A public cloud is built, controlled, and maintained by a third party cloud provider
* A hybrid cloud has both public and private clouds and is hosted in an inter-connected environment
* You can also use different features from different cloud providers
* Cloud computing runs on a consumption based model, which means you pay for what you use
* An application must always be available, which is where cloud services come in
* Can look in the service-level agreements for what the projected uptime of various services are
* Microsoft Well-Architected Framework for info on cost and performance predictions
* Cloud features also support governance and compliance
* Need to figure out which form of cloud service we will be utilizing (SaaS, PaaS, IaaS)
* IaaS the cloud provider is responsible for hardware, network connectivity, and physical security
* PaaS provides complete development environment without having
* PaaS is often used with forecasting applications
* SaaS allows for the most help from the cloud provider, and only has the user provide data as well as selected devices
* Need to get AI resources from Azure at some point most likely
* Can begin by migrating existing app to VM
* In azure power shell the Get-date button returns the date
* Most azure commands start with "az"
* Az interactive command makes the shell into more of an IDE
* Physical infrastructure for azure begins with the datacenters
* Datacenters are located around the world
* Datacenters are grouped into Azure Regions or Azure Availability Zones
* Availability zones are physically separate datacenters within each Azure Region
* Utilizing Azure availability zones can help to make application more available
* Management infrastructure includes Azure resources and resource groups, subscriptions, and accounts
* Resources are anything created deployed, etc. Within azure
* Resource groups allow you to group resources together
* One account can have multiple resources
* Create resources by using the create resource button on the home screen
* Using Azure virtual machines allows for the user to have a lot of flexibility, as it is an IaaS
* Virtual Machines can also be grouped together
* Should use VMS during testing and development, extending datacenter to cloud, and other related events\
* You can create a VM in the azure shell using the command az vm create
* Need to pipe in a good amount of arguments to create the VM
* Can also add extension sets to the VM using az extension set command
* Can also utilize a virtual desktop with windows running on it
* Separates information from the actual user's device, increasing security
* Azure containers allow you to run multiple instances of an application on a single host machine
* Docker is an example of a container engine
* Azure allows you to upload containers and then the service will run it for you
* It functions as a PaaS
* The Azure Kubernetes service is a container orchestration service, that allows you to deploy multiple containers at once
* Could potentially use containers for load balancing/ modularizing the frontend, backend etc.
* Azure also has a function service, which allows you to run a serverless compute on various functions without maintaining virtual machines or containers
* Good to use this when the underlying platform or infrastructure of the code does not matter
* Also scale automatically based on demand
* Can also have a function that keeps track of context
* Also only pays for each calling of the function
* Azure App Service allows you to build and host web apps, without managing the infrastructure
* Does not support MacOS 🙁
* Can also be used to host web applications
* Can also run Mobile applications
* Azure virtual networks allow for azure resources to communicate with eachother and with users
* Can also connect virtual networks
* Can also segment the IP addresses
* Can enable incoming connections by assigning public IP address
* Allows you to create a network that can span the on premises resources as well as the Azure resources
* Can also link Virtual Networks together
* Az VM list command in the cloud shell lists all of the currently running vms in the cloud
* **Tutorial: Configure network access may be helpful for connecting to the azure VMS in the future**
* Command az network nsg list in the cloud shell lists the network security groups associated with your vms
* Every VM on azure is connected to a network security group
* Az network nsg rule list command lists all of the regulations of the network security group
* Allowing inbound connections on port 80 allows for the public to access over HTTP
* An network nsg rule create allows you to create a rule that allows for access on specific ports and such
* Nsg rules is how you can configure who can connect to a VM
* Azure also has Virtual Private Network services (VPN)
* This allows for you to connect multiple private networks to one another over the internet
* Connect resources using the VPN gateways
* Can connect different virtual networks together
* Can use either a Policy-based VPN gateway or an in Route-based gateway
* Route-based VPN gateway can be used to connect between multiple virtual networks
* Many options for maintaining high availability on these resources
* Azure also provides a DNS hosting service that provides name resolution
* This DNS is very inclusive and easy to use
* Also supports virtual networks with private domains
* CANNOT USE AZURE DNS TO buy a domain name
* You can create an Azure storage account that can store data and be accessed from anywhere in the world
* Blob storage stores unstructured data
* Can also utilize file sharing systems using azure
* Azure table storage can store structured data
* When creating a storage account, make sure to have a unique identifying name
* There are also many options for data redundancy to keep your data available, with each type having specific use cases
* Locally redundant storage is the cheapest but most susceptible to becoming unavailable
* Also pricing tiers for blob storage, depending on how often it is used
* Azure Tables use NoSQL: ideal for storing structured, non-relational data
* Change access levels for data storage such as blobs to change who can see their contents
* This allows you to access the blobs content from the internet
* Azure allows for real time migration of data, infrasturcture, and applications via Azure Migrate
* Also supports asynchronous migration of data using Azure Data Box
* Azure migrate is used to transfer on-premise data to the cloud
* Azure Data box can be used to move large amounts of offline data to Azure, (this is used for physical data)
* Azure also has services to move individual files or small file groups
* AzCopy is a command line utility that can be used to copy blobs or files to or from storage account
* AzCopy does not synchronize bi-directionally
* Azure Storage Explorer is standalone app which provides a GUI for the manage files and blobs in storage account
* Azure File sync can be used to centralize file shares in Azure files and keep benefits of a Windows file server
* DevOps is a software development practice that promotes collaboration between development and operations

**Azure DevOps**

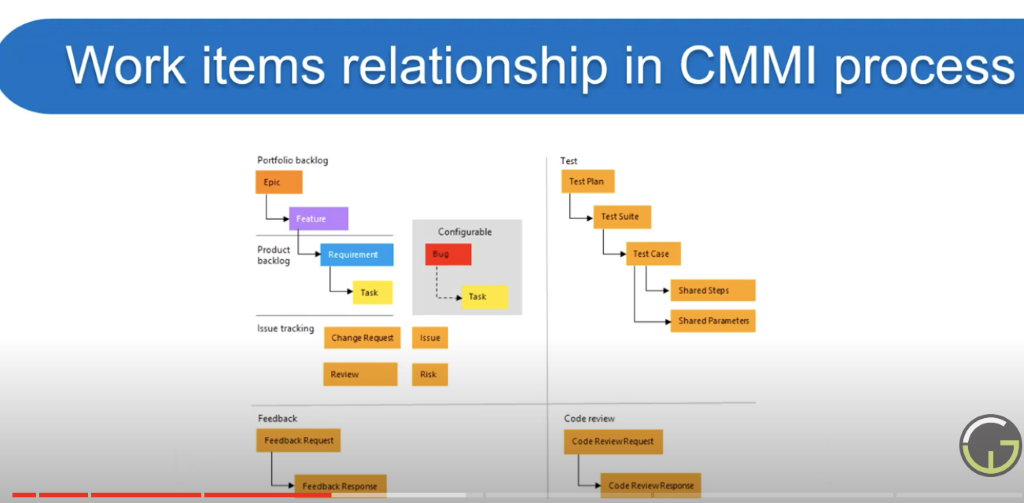
**What is DevOps??????**

* Joining of development team and operations team is DevOps
* Development team is often responsible for Coding, building, and testing the code
* Operations team is responsible for Deploying, operating, and monitoring the deployed code
* DevOps is a work culture
* Makes it much easier to deploy code to various deployments
* DevOps runs the lifecycle on each commit or particular interval by using an automate process
* Essentially when code is pushed it is automatically built test and deployed
* Also can be used to test the quality of the code before it goes on the server
* No maintenance cost for devOps
* Azure devOps is a set of modern services whch streamlines software development lifecycle
* It is a website
* Azure DevOps is a feature on Azure
* Features of Azure DevOps:
* Boards: Boards are used to plan your work using AGILE methodology
* Repos: allows you to manage your code via git
* Pipeline: related to deployment of code, contains all automation processes for CI/CD
* Test plans: Can set up both manual and automatic testing for code using this service
* Artifacts: A collection of packages from repository that are required for deployment
* Azure DevOps provides two types of server configuration
* Cloud: will be used by us most likely
* Azure DevOps structure consists of Organizations, Projects and Teams
* Organizations are used for organizing and connecting groups of related projects
* Must have at least one organization
* Each organization will have its own URL
* To create a new organization hit the "+ new organization" button
* Must choose a cloud data center to host the projects, depends on project requirements
* Statistics for gambling
* Fixed cost, variable costs, find out where the profit margins begin
* <5 people on devOps organization allows for free use of DevOps services
* Project provides a repository for the source code and a place for people to plan and track progress on software
* All data related to a software is stored under the project
* Project provides all the main services of Azure DevOps
* Public project is used for open source development
* Each project has a unique url
* Private project requires access on a user-to-user basis
* Can manage team code using git/TFS
* Project settings button takes you to a place to update the project settings
* Administrators have the highest privilege level
* Can also see which teams are working on the project and add more
* User must have admin level access to make changes to the team
* To resend a user's invitation go to the organization settings and then click on the user you would like to reinvite

**Azure Board Information**

* Azure boards are compatible with scrum/AGILE
* Work items can be various different things ranging from a user story to a bug
* Hierarchy of things to do go epic > issue > task
* Work flow is the process of updating work item progress
* Can add issues under epic work items and so on
* Can also progress which state it is currently in (to do, doing, done)
* Can also assign various levels of priority to work items
* Can also add attachments to work items
* A backlog is a collection of work items which will be used for future development
* Product backlog is ordered list of everything that is known to be needed in the product
* Sprint backlog of work items which are in TOTO state
* Very intuitive to create hierarchy in the backlog
* Sprints separate out a project's lifecycle in distinct phases
* Can set working days for timeline's sake in sprints
* Can edit what kind of methodology board is utilized ex. Agile, scrum
* Agile has test cases and more variety of work item types
* Work items relationship in Agile process 
  Test 
  Feature 
  Story 
  backlog 
  Issue tracking 
  0 3:33 / 10:34 
  Confi 
  Task 
  •o 
* After finding a bug you can set multiple tasks related to dealing with the bug
* Very streamlined testing work items
* Bug is a poorly implemented feature
* Epic represents a business initiative to be accomplished
* Feature is a shippable component of software
* Task is the smallest unit of work
* Begin with user stories and then code the features required for the story to actually happen
* Workflow of user bug - 
  Apgoved to 
  backlog 
  Fixed fixed 

Workflow of user task - 
Work 
started 
Cornpleted 
Reconsidering 
the task 
from the 
Work 
halted 
Reactivated 
o 

* Backlogs contain a collection of tasks to be completed
* Team settings can be tweaked to add more navigation levels to the backlogs (such as epics)
* Utilize the hierarchy of work items when creating the backlog
* Iteration specifies which sprint the work item will be created in
* Can email work items as well as create copies of them
* Risk specifies how much the completion of a work item will potentially fuck up the project
* Can also utilize a development link in items to take you directly to the code which should be worked on
* System info can utilize the pipeline information which the repository is set up
* Can set start and end dates for sprints
* Can drag work items from the backlog into the sprints
* CMMI = Capability Maturity Model Integration (workflow framework such as agile)
* CMMI has requirements, not user stories
* 
* CMMI also utilizes change request which means the work is done but needs to implement some other functionality
* Can also create a project utilizing Scrum Methodology
* Work items have their unique classifications under the Scrum Methodology
* Work items relationship in Scrum process 
  Portfolio backlog 
  Feature 
  Product 
  Produ 
  backlog 
  tracking 
  0 3:33 / 15:03 
  Task 
  • Chapter 3 > 
  ap er 
* Impediment is a classification that shows that a work item cannot be continued to be worked on due to an impediment
* Product backlog item is equivalent of user story or requirement
* On scrum you have to approve various work items
* Can see one sprint at a time for the scrum processes
* On the Kanban board there are lots of various options that can be worked with for the columns and other board items
* By clicking on column options you can then change the name of the column names and such
* By dragging and dropping column names you can change the order of the columns
* Can also add a sorting order to the column titles
* A query is a combination of logic which is applied on work items
* Can use this to filter work items based on various logic example: work items assigned to a specific person
* Queries is a sub menu of the boards category on the web portal
* You can follow specific work items to keep tabs on them
* Very easy to delete/edit queries
* Can also create folders to further organize queries
* Can send a query on email there is an option on the three dots to email query
* Query editor allows you to edit syntax logic to create a new query
* Click on run query button to run the query
* Can add multiple logic criteria to queries to further specify results
* Azure Devops allows you to customize projects using inherited processes
* Existing work processes cannot be edited
* You are able to create custom processes yourself
* Inherited process allows you to create any custom process type
* Example a new process can be inherited from the basic process, and customized as needed
* Inherited processes are fully customizable
* To create inherited process go to organization settings and click on the process settings
* To create an inherited process you have to choose an existing starting point (Agile, Scrum, etc), and then choose the create inherited process button
* Allows you to use custom process in work item process drop down menu
* Work items and fields are also completely customizable in inherited processes
* To customize fields in custom work items click on the custom work item in the settings
* Can add any existing fields or create your own
* Can also change the location of a field via customization
* To edit work items click on it while in the settings menu
* Can also add groups to custom fields
* Fields can also be hidden from the user's view
* Kanban boards of inherited processes can be customized
* Go to team settings of the kanban board to gain access to the column customization
* Can also split specific columns between doing and done
* You are also able to customize the fields on cards, such as adding new fields or hiding them
* Kanban board cards can be further customized via style rules
* Custom styling can be tied to logical statements
* Can also add coloring rules to tags on the kanban board tags

**AZURE REPOSITORY INFORMATION**

* Azure repo is a container hosted by the server of Azure devOps containing the project code
* In Azure you can either use Git version control or TFVC (Team foundation version control)
* Git is a distributed version control system and TFVC is a centralized version control system
* Git works by disconnecting the repository and making a local copy
* TFVC makes a new repository for each developer on the team
* Both are version control systems
* Git operations are faster
* In Git an entire local repository is available on the developer machine
* Utilize visual studio code to work with code using Azure repository
* Tortoise git provides lot of GUI options to work with git in command line
* Git add adds all the changes
* Git commit adds info pertaining to what change has been made in the repository git commit –m "{Enter Message Here}"
* Git push sends the changes to the web server for the repository
* Synchronize between the local repositories as well as the one hosted by Azure
* You are also able to work directly on the Azure devops server repository, can add new files, folders, commits, etc.
* Can also click on cancel button to revert a change that you accidentally started
* Git pull on local machine to bring changes from server to local machine
* Can also upload files directly to the repository
* **Branching on the DEVOPS, this is most likely VERY IMPORTANT SO REFER TO THIS WHEN NECESSARY**
* Default branch is called the master branch.
* Branch allows for pointers to point to commits besides the master branch
* Branching does not alter the size of the repository (hard copy all of it)
* Can also switch between branches and make various commits in various branches
* Can create a new branch at any commit within the repository
* Merging merges the pointers from one branch to another
* Branch = movable pointer to a commit
* Can also create a branch per developer
* Git tracks the current working branch using the HEAD pointer
* Can create branches via the server repository or the local repository (click on the add new branch button)
* Git branch lists all of the branches available in the repository
* Current working branch is also listed on VSCode
* To create a new branch go to the repo menu, and click on the new branch button in the branches sub-area
* DevOps portal shows what commits are ahead/behind the master
* Git pull to gain access to new branch on local repository
* Git checkout allows you to switch from branch to branch
* Run explanation on how github connects to devOps (comprehensive definition)
* Git hub desktop can deal with the merge conflicts and branching and such
* Deleting a local branch does not delete it from the server, only the changes on the local branch are lost
* Git branch {branch\_name}, allows you to create a new branch
* To sync local repository and azure repository: Might have to look into tortoise git, otherwise creating a pipeline to auto synchronize the two repositories could be fire.
* Git push –set-upstream???
* Git push --set-upstream origin {branch-name}
* Look more into how upstream branches work for command line tool (this is what I'm going to use)
* Pull requests are used when specific features are local to a branch, then the branch will get merged with the master branch with the pull request
* Pull request merge two different files
* Creating a pull request on azure involves going to the pull requests menu under the repos subsection
* This provides a Gui to specify which branch you want to have pulled into the master branch
* Pull requests allow you to look at a list of all the
* If conflicts happen in the pull request then you have to manually resolve the conflict by editing the necessary files
* To complete the pull request you merge the branches
* Can find specific pull requests by searching for specific ids

**AZURE PIPELINES**

* Pipeline in DevOps is a set of process (automated or can be triggered manually) which is used to deploy the code to users
* Azure pipeline can be used to automatically build and test code
* Each project has its own specific pipeline
* Process goes from repository->build->test (unit testing) -> deployment to users
* Azure devops pipeline works with any programming language and any platform, so this is not a constraint
* Project source code needs to be some sort of version control system
* Integrates with GitHub, GitHub Enterprise
* Code can be deployed to containers, VMS, Azure Services, or any on cloud targets
* For a public repository the pipeline is free
* **Private project you have to pay after using 30 dollars each month (30 free hours a week)**
* CI is a part of the pipeline where the code is built , tested
* CD is used to automatically deploy the code
* Basically first and second phase of the pipeline
* The pipeline can be setup to automatically run everytime a commit is changed, a fixed interval, etc.
* Two steps to setting up a pipeline: go to repo, set up build