**What is Azure Devops Platform:** Azure Devops is a set of modern tools/services and a methodology/approach that we will practice implementing in our projects (or) applications in organizations/IT firms.

Azure Devops tools and services helps us to automate the repeated work or process for building, processing, and deploying the IT Projects in tech companies(ex: IBM, Infosys, Cognizant, Deloitte, Microsoft…etc.)

**Advantages of Azure Devops Platform(tools & services):**

**(i)Maximizes the efficiency:** with automation, like using the CI/CD pipelines services of Azure Devops we can automate the deployment process (specially for application operations team) at a particular frequency or time or as per project requirements.

**(ii)Optimizes the entire business:** With Devops we can make the most effective use of all the files, codes, logics that we have written in different repositories also to merge them with different branches as in whenever needed for different projects.

### (iii) Improves Speed and Stability of Software Development and Deployment: Devops has a very great GUI which helps us to manages hundred & thousands of files and also offers different services for the project implementation which significantly improves the speed of project development and provides stability for code deployments in multiple environments with an ease.

### (iv) Gets You to Focus on What Matters Most: In Azure Boards we are having a variety of parameters to ask, like tags, links, attachment’s, work items, description, start date, end date…..etc. with this it will become very easy for Devops Egg to focus what matters the most for the current sprint and the work items which gives huge value to the business.

**(v)Implementation of project development using Devops is cheap:**

As compare to previous approach(traditional) using Devops methodology will be very cheap for the organizations and can save huge amount of cost for the project development

**(vi)Straight forward to implement different stories, work items & projects using Devops:**

Devops approach provide plenty different services and tools which helps the developers(Devops Engg’s) to focus straight forward for the implementation of current sprint and its work items (stories, issues, task, epic….etc.)

**Why organizations(Infosys, TCS, Tech-M, NTT Data…) needs Devops Eggs??**

There are many gains, benefits & advantages that organization has if they opt Devops Engg, the Devops eggs are familiar and well aware of all benefits mentioned below (or) Devops practices for projects implementation

(i)Devops Engrs will fill all the gabs between Development team. Quality assurance team & Operations team.

(ii)It helps to develop and deliver rapidly high-quality applications to end users, with 99.9% accuracy with transparent approach & clear process to end users.

(iii)Devops eliminates communication & collaboration challenges in the IT organizations which ultimately improves the productivity for the organization and financial benefits to clients.

(iv)Customers enjoys user friendly apps that regularly updates bases on the feedback in the ongoing and upcoming sprints.

(v)If we adopt Devops methodology then it helps the organizations to improve the agility (ability to rapidly respond to the change which ultimately increases productivity, increases revenue, lower operational cost & more competitive).

**Who should learn Devops:**

1. **Developers:** Each person who is working under software development team(ex; Developers, Testers, Product owners, Scrum Masters, Delivery managers, Area DM, Account heads…etc.)
2. Whichever technology you are working on, you are must/should learn Devops, which ever technology you are working in, irrespective of technology it became necessary for each and every one to learn DevOps., it became now the need of an hour.
3. **Operations Team:** All the members who are working in Operations team are equally responsible to learn Devops, either they are responsible to code deployment, management, monitoring, Governing…etc.
4. **Individual Developers:** like you are not part of any team and you are an individual contributor on a particular project still we need to learn DevOps, because it improves our process, speed of development, speed of deployment, accuracy of the project, quality of the project…etc. All these things it is going to improve a lot.
5. **Freelancers:** if you are working as a freelancer and need to manage your code, sometimes what happens is, we write our code into system and b’coz of some reasons that system got crash then we lost all our code, all planning’s, all documents…. etc. everything will be lost, and if we manage all these things in a particular server (Devops portal is hosted on Azure servers) then everything will be available to us, every time and whenever we want we can access that server from anywhere.
6. **Management:** people working under management must/should learn Devops, bcoz what exactly the development team is doing in the project, what is the status of the project, what is sprint status, work graphs, and the other work related details of the work which is going on can be easily known by managers if they are familiar with management related functionalities of Azure Devops, hence management team should also learn DevOps…etc.
7. **Client:** If you are the client of a product and some team is working on your product using Devops method, then you must learn about the DevOps, bcoz we will get a complete picture of what is going on in the development side, we can see the progress with the help of multiple types of graphs, with the help of deployment, how many bugs are there? How many stories are there? And these points are taken care by DevOps.
8. **Everyone:** Everyone who is related to the development of something in any ways, if developing something and you have any connection to that development then we must learn DevOps.

**Why the IT industry is using/adopting to develop/execute the projects in Devops:**

1)Complex system was there and that must maintain…

2)Cost expenses were more to maintain/sustain the projects using water fall (WF) methodologies.

3)Separate teams has to be maintained (development team(5), testing team(5), application operations team(5), monitoring team(5))

4)If they need to roll back some code that they have executed it will be too complex.

5)There is a lot of dependency on development team in water fall methodologies....

6)In this WF methodologies there is a lot of coordination/discussions/meetings required to develop the project and which impacts the **Productivity** of the project development.

7)Managers/Product owners/Client managers need to attend all the meeting and follow-up all the email about the project development activities....etc.

**How Software Development works(SDLC):**

1. **Planning:** In the start of everything we have to make a plan. Once the plan is ready then we have to write the code..
2. **Coding/Development:** We have to write the code(converting the business requirement to technical requirement), once the coding is completed then we have to build the code.
3. **Code Building:** After the build is completed then we will forward or our code/functionality to testing team.
4. **Testing:** Here the test team will test the code/product then it will get deployed to the servers(QA, UAT,Prod…etc.) and once the code is deployed on the server then there are certain operations that needs to be performed
5. **Deployment:** Will do the code deployment to different environments as per the need of the project or as per the project architecture
6. **Monitor:** Here the application operations team/Administrator will monitor the application and infrastructure.

**Software Development Life Cycle:**

Plan

Code

Building

Test

Deploy

Operate

Monitor



Business team/Product Owner/Business Analyst

Development team/Devops Engg’s

Operations team/AO team

**Business team/PO:** sits with the client and works on the planning part of the product.

**Development team:** includes developers/testers, they code build and test the product.

**Operations team:** The main purpose of operations team is to deploy the code on the servers, monitor the applications and operate it.

Earlier where we were working under water fall model all the above three teams were working independently, but when Agile came into the picture the gap in b/w these teams got filled.

Business team/Product Owner(PO) and the development team were working together on the success of a product, now we have one more team i.e.; an operations team and still there was a gap in b/w these 2 teams the development team & operations team.

**What is Devops:** Devops is something where the **Development & Operations**  teams work together, here we see Dev from Development, Ops from Operations. Hence

**Development + Operations = DevOps**

Devops is something related to the development team and operations team & the **+ symbol** between them represents that they work together, basically Devops fill the gap in b/w the development team and operations team. Just like how agile did to the business team/PO and the development team the same thing is done by the Devops for the development and the operations team

Devops is a culture or practice which fills the gap between development team and operations team, if we are thinking DevOps is a tool, or a particular framework or any programming language then this is **not true**

DevOps is basically a culture, it’s a practice where both the teams(Development team & Operations team) works together in the entire Software Development Life Cycle(SDLC) to ensure the quality of product, ability of product to work in different environments and to deliver the value for the business into bits & pieces.

**Benefits of DevOps:**

**Fast Delivery:** Deploying code from Dev machine to server is very fast, basically developer has to commit the code he has to push into the particular repository and all the process of deploying the code will be automatically, so it’s a very fast deployment procedure.

**Reduce time:** Lots of things(like testing/deployment/rollback) are happening automatically, obviously in the beginning we have to make some setups/configurations, we have to write some code, but once we are done with all those things, then everything will happen automatically(like testing/build/deployment…etc.)all you have to do is we have to focus on the development and we have to push our code and rest will be taken care by the DevOps.

**Rollback:** Any release that we think should not go to end customer or should not get deployed to our servers (like Demo/POC/Production...etc.) bcoz of any reason, then we can roll it back very easily. Sometimes what happens you push the code which has a feature that we don’t want to expose it to the customer then it’s very easy that we can roll back, in another scenario like a wrong feature/logic got checked in and that is deployed on the server and still in this scenario also we can roll back to our code very easily.

**Quality:**  with lots of automatic tools we can check the quality of code before deploying it on server, there are several steps that we can apply the manual testing or automatic testing all these steps can be taken care by DevOps and we can ensure that the quality work is going on the server and if there is any problem we can fix it there itself and again we can push the code to server and can again run the entire process to make the successful deployment.

**Collaboration:**  All teams works together to build a great product, whether you are coming from the management background, development team or from the operations team.

**More Agility:** Every commit is treated as a final delivery, if we talk about the agile process then in the Agile what happen is we have to work in the sprints and at the last of the sprint we have to give something valuable product to the customer and valuable means it should be well developed, well tested, well deployed all the things should be there in that code. If we break it down into very small parts and we say that all these steps will be taken care whenever we will make single commit that’s why we say more agility, means it brings more agility to the development.

**Easy to use:** Azure Devops is very easy to use, we can learn it very fast, it is very simple to user, easy to understand.

**No Maintenance:** There is no maintenance cost if we are working with DevOps.

**Reliable:** If we are working with DevOps then will always get the quality of product, quality services from DevOps. Bcoz Azure Devops is a product of Microsoft Corporation.

**Secure:** Security is something which is the main concern, Devops is completely secure and it also depends on the tools which we are using for the Devops, we can say that Azure Devops is 100% secure, there is a guarantee of the security, there is no doubt in the security of the process of everything and also there are many other lots of benefits of the Devops.

**What is the need of Devops :**

If we are implementing our projects with Dev-ops then there is no need to have managers, project coordinators in our team to monitor all the task and status of current on-going sprint, Dev-ops platform have designed in such a way that it will reflect the complete status of the on-going sprint in the Dev-ops portal.

**What is Azure DevOps:**

Azure Devops is a set of modern services which is used to

(i)Plan smarter (ii)Collaborate better & (iii)Ship faster

**Relation B/W Azure & Azure Devops:**

**Azure:** Azure is a cloud solution which provides lots of services and features to help our organization in almost all ways during the development of any product by providing multiple Tools, Resources & technologies. There are lots of technologies that we can use in Azure, it provides lots of other things like Storage, Vnets, Virtual Machines, Kubernetes Services, Azure containers, IP’s, Backups of Resources…etc.

**Azure Devops:** It’s a feature or services of Azure, we can say like Azure is something very broad & **Azure Devops is a part of Azure**, we can say Azure Devops is a feature of Azure, if we see as per below graphical representation then we can say **Azure is something very big and Azure Devops is a part of it.**

**Azure Devops Services:**

Basically, there are Six Services, which are available in Azure Devops Projects.

**(i)Overview (ii) Boards (iii)Repos. (iv)CI/CD Pipeline. (v)Test plans. (vi)Artifacts.**

If we broadly classify then we have above six services in Azure Devops, here we can choose anyone of them, all of them, but for the success of any product/Project all these six services are required.

* **Boards:** Boards are basically used to plan our work, here we can write our stories, we can write tasks, we can write bugs, we can write epics and we can manage their progress, we can see in-progress tasks, completed tasks, to do tasks, basically all these things the Agile Board or Kanban Board all these things can manage in the Board.
* **Repos/Repository:** Repose are basically something related to our code, basically it is a repository, we can use Team Foundation Server(TFS) repository or TFS or GIT and we can manage our code, we can create branches we can create multiple repository in our project, we can use all the features which are available from the GIT.
* **CI/CD-Pipeline:** Pipeline is something related to the deployment of our code, all the tasks like the automation process, the CICD, everything will be available in CI/CD pipelines.
* **Test plans:** Testing is also one of the main features in the development (like automation testing, manual testing) all such things can be taken care by Test plans.
* **Artifacts:** It’s a collection of our packages which are getting from the private repository or from the public repository, which we need to share among your team, also these are required for the deployment, so, basically in Artifacts we can store our packages which are simply required for that development.

**Choosing Azure Devops server:** Lots of companies do not want to use cloud bcoz they do not want their project to be stored on some other server, they want store data(coding) on their own servers. There are lots of scenarios where organizations loves the idea of the cloud, they basically love how to work with the Azure resources, how to create the resources, how to manage the resources in the cloud, how to delete the resources, all these things but ultimately we are working on the cloud, then our data will be stored at some other place then they do not want this to do. They want their data to be stored in their own premises,

and If we are working with Azure Devops then we have 2 options to choose the server, 1st one is cloud & 2nd one is on-premises.

* Azure Devops provides two types of configurations. i.e.;

(i)Cloud

(ii)On-premises

If we do not have our own setup and not having our own servers then we can use cloud servers, so basically whenever we will create a project or an organization in Azure Devops then we have to choose the reason where our data will be stored, and Second option is on-premises, if you do not want to use the server of the Azure Devops then we can use our own on-premises servers and everything all the data will be stored in our on-premises.

**Relationships of VSTS & Azure Devops:** On September 10th 2018 Microsoft renamed VSTS to Azure Devops services, earlier it was known as VSTS which is Visual studio team services(VSTS)but now after 10th Sep 2018 it is known as Azure Devops services, the below are some features which everyone should know in VSTS & Azure Devops.

|  |  |
| --- | --- |
| **Visual studio team services (VSTS)** | **Azure Devops Services** |
| Build & Release | Azure Pipelines |
| Code | Azure Repos |
| Work | Azure Boards |
| Test | Azure Test Plans |
| Packages | Azure Artifacts |
| ….etc. | ….etc. |

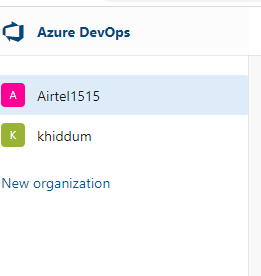
**How to create an account in Azure Devops:**

* Azure Devops portal can be accessed using the website <https://dev.azure.com>
* We can access this portal from any browser like IE, Chrome, Safari, Firefox, Microsoft Edge…..etc, the only thing we need is good internet connectivity.
* Devops is a service of Azure (or) part of Azure, Devops says plan smarter, Collaborate better & ship faster with a set of modern Dev services.
* We can create an account in Azure Devops in 2 ways i.e.;

(i)**start with GitHub:** if we are having an account in Git Hub then we can login with the help of GitHub & if we do not have an account in the GitHub then we can login with simple Email ID.

(ii)**Gmail or any email id:** Here using any email id we can create an account in Azure Devops.

* If we see on top left on Azure Devops portal then we are finding Organizations with different names



Organizations

**Diagrammatical representation of Organizations & Projects in Azure Devops:**

Organization name

Airtel1515



Project-1

Project-2

Project-3

Multiple projects placed under one organization.

**When we should create multiple organizations in Devops:**

* If we are having large number of projects in our company, like hundred & thousands of projects then managing all those projects is not an easy job then we can create multiple groups(clients) from our projects and for each group we can create an organization.
* We can treat an organization as an Account, Business unit, Groups…etc.
* Ex: When we work in big organizations (TCS, Infosys, Cognizant…. etc.) then for each client we treat client as an account. Bcoz under that account there are multiple projects which we are working on. So, for each account we can create an organization
* Each organization will have its own URL and if we see below then the organization is part of the URL

Example:

<https://dev.azure.com/khiddum/>

<https://dev.azure.com> >> Devops portal link

/khiddum/ >> Organization name

**Types of Projects in Azure Devops:** In Azure Devops there are 2 types of projects

i.e.; **(i)Public Project**

**(ii)Private Project**

**(i)Public Project:** If we create a public project then, below points to be considered.

(a) Public projects are visible to everyone.

(b)No login is required to get access for public project, we simply need the URL of that project and put the URL of that project in our browser and all the things which are related to the project will be available to everyone.

(c ) Each public project has a unique URL. In order to access the project, we need a URL and in Azure Devops each project has a unique URL.

(d)Public projects are mostly used for open-source project development.

(e) We can create unlimited number of public projects under one or more organizations.

**(ii)Private project:**

(a) If we create a new Private project then Private projects are visible to limited users.

(b)Basically these projects are visible to only those users to whom will provides some access.

© User must login into Azure DevOps portal to access the project. Each project has a unique URL.

(d)Private projects are mostly used for non-public software development.

(e) if you are working in an organization or you are the organization, where we want to keep our things private, then we choose a private project and then we can do the development of the private project.

* Now let's have look how can we create a new project on the Azure DevOps. To create a new project
* On Azure DevOps first you can open the Azure Devops portal in our browser, and we have to login by using our Azure DevOps portal, so, this is the URL <https://dev.azure.com>
* Once you login into the application. You will see the organization on the left side. The first step is we must choose an organization. So, we have to choose under what organization we are going to create a new project.
* suppose I am going to create a new project Ex: NareshIT. So, this is the organization. This organization does not have any project right now. I am going to create first project under this organization.
* suppose this is the name of the project My\_First\_Project application, and then we have to enter the description of this project over here. Then we must choose the visibility.
* Visibility means the type of the project. you have to choose whether you are going to create a public project, or you are going to create a private project.
* I am going to create a public project. So, I will click on the public. Here you will see there is one more dropdown. Which is advanced? If we click on it then you will see-.
* First, you have the version control. Let's see what is available under this version control. Here we can see there are two things.
* The first is git, and the second one is TFVC. So, basically, whenever we are working with the source code. There are two types of things to manage our code.

**(i)Git(Global Information tracker) &**

**(ii)Tfvc(Team foundation version control)**

* We can choose any one of them as per our requirement or need.
* Suppose by default I am using GIT and then on the second dropdown will see work-item process. Basically, if we click on this then will see there are four options.

**i.e.; Agile, Basic, CMMI, and then Scrum.**

* These are basically the work-item process, the board of the project depends on the options which we will choose from this dropdown, suppose by default we are clicking on the basic
* I am choosing the basic one. If I choose basic and click create. A new project will be created for us and here we have a new working project for us.
* Here on the left side we can see all the main services of this project.
* First, we have (i)Overview. (ii)Boards, (iii)Repo, (iv)Pipelines, (v) Test plan, and (vi)Artifacts. All these features are available **for every Azure Devops project.**
* This is the home page for this project and these are the services. Let's see all the services one by one So, there is nothing in the services also. If we see on the Boards, will see a completely blank board is available. If we click on the Repos we will see there is nothing in the Repos. Then we have the pipelines everything is completely blank test plan. there is nothing in the test plan. And if we click on the Artifacts. There we will see nothing in the artifacts also. basically, we are learning how to create a new project.
* Suppose we have already created a new project and now we want to see where is that project? So, suppose I am here on the organization tab Ericsson. If I want to access the project which I have created just now for that first I have to click on the organizations.
* In the organizations we will see all the project are available under the organization. So this is project which we have created just now. If I click on this one you will see we are on the same project.

**Teams Setup for Azure Devop’s Projects:**

**Team:** Team is a group of people who are responsible for the software development, when we are working on a project lots of people work on the same project with different roles and responsibilities. Whenever we create a new project, a team will be created automatically with the same name of the project and at the last word **TEAM** will be added to the name of the team.

**Example:** Developers/QA’s/Scrum Master/Product Owner/Application operations members/stake holders…etc.

* Users must have different level of access to make necessary changes in the project in Azure Devops, ex; changing the project name, changing the name of the team...etc.

**Inviting a new member in the team(1st Approach):**

Go inside the project>>click invite on top right>>search users in search box(ex: [khiddu.m@gmal.com)>>click](about:blank) on Add @ the bottom>>and finally at top we get pop up saying Added 1 user to 1 team.

* When we invite a new member in the project he/she will get an email on his email id saying You have been invited to Azure Devops and when the user click on Join now in the email and then he will redirect to our Azure Devops projects.

**Inviting a new member in the team (2nd Approach):**

Go to project settings>>Teams>>Click on the team we want>>+Add>>search the user in User or group box and to add multiple users use a semicolon>>save changes

**Email invite after adding the users in Azure Devops project team:**

Go to organization settings>>Users (left side)>>select the user (which we want)>>Resent invite>>and finally the user will get invite on his/her email.

**Setting Member Permissions/Changing Members Permissions @ Organization level:**

Goto organization settings>>users (left side)>>click on the 3 dots(extreme right, this will have for each nd every user)>>change access level>>choose any access (like Basic/Stakeholder/Visual Studio Subscriber) and finally click on save(bottom).

* In an organization there would be multiple projects and if we want to assign a particular user to multiple projects or want to remove an access of a user from the projects from an organization then.

**Removing/Granting access to users for multiple projects:**

Goto organizations settings>>User>>click on the 3 dots(extreme right)>>manage users>>un-check the projects from which you want to remove the access>>save changes.

Same as if want to provide access to different projects in same organization then we can check the multiple projects and then provide the required access(like project contributor/project administrator….etc.) and finally click on save changes

**Different types of Boards available in Azure Devops/Azure Boards types:**

Basically, there are four type of **WORK ITEM PROCES** will find while creating a new project in Azure Devops.

* When we are creating a new project in Azure Devops then in Advance section will see there is a drop down for work item process and will see 4 values in the drop down those are

(i)Basic>>

(ii)Agile>>

(iii)Scrum>>

(iv)CMMI (Capability Maturity Model Integration).

* The type of Boards depends on these work item process. Here each type of work item process (in drop down) will provide different types of work items and workflow.
* These work items and workflow depends on the selection of work item process, if we choose the Basic one will have different type of work items and work flow, if we choose Agile (or) Scrum (or) CMMI then will get different type of work items and work flows for our Azure Devops projects.

**Work item:** A work item is a unit(small or large) of work which has several characteristics, and it is a part of our product development. When we work on development of a product then we divide our tasks into small parts and that small parts or group of those parts we say it as a work item. Basically, each and every single type of work that we are doing in our project development can be consider as work item. Work item is an independent unit which can be track independently on our Azure Board

|  |  |
| --- | --- |
| Characteristics of work item | Example of work item |
| Title | Epic(an Epic is a work item) |
| Description | User story(user story is a work item) |
| Assigned To | Bug (bug is a work item) |
| Completed By…..etc. | Improvement…etc. |

**Work item in Azure Boards with Basic Process:** when we work on Azure boards there are multiple types of boards and all the boards depends on work item which we have choose at the time of project creation.

**What work items are available in Azure Boards when we work with Basic Process:**

* There are three work items in Azure Boards with basic process. When we create a project with basic process then we have only 3 types of work items and i.e.:

**(i)Epic:** An Epic is basically a kind of functionality or a kind of large work(ex: we have to implement login/sign-up/forgot password/…etc all these functionality in our application, so we can say Epic is a membership, basically all the things which are required in the membership will be a part of Epic )

**(ii)Issue:** Under this Epic we can create multiple type of issues and in Azure Boards with Basic process an issue is something that it may be a bug or a user story or an improvement…etc. all these are part of issue, and under this membership we can create multiple types of issues like login functionality /sign-up functionality /forgot password functionality…etc.

**(iii)Task:** Task is the smallest unit of work, firstly we have Epic then Issue and then Task and for each issue we can create multiple task (Ex of Task: under login functionality we can create multiple task like

Task 1: creating table in the DB

Task 2: creating an API is another task

Task3: integration of API

Task4: Design of the login page…etc.)

**Relationship of Azure Boards work items:**

Diagram, funnel chart

Description automatically generated

**Work flow:** How the work item flow in Azure Boards is called a work flow. Wok flow is the process of updating work item progress. when we work in the product will update our work items, if we talk about the work item which was implement the login functionality then first a work will be assigned to you then will work on that task and then will complete it.

**Note:** To reflect the users for the work items(like Epics, Issues & Tasks) first add the users at organization settings, then add the users at default project team or new team then click on Boards, Select the board, design the work item then will see all the users reflecting for our work items to assign.

**Managing Basic work flow in Azure Boards:**

To Do >> In-progress >> Done

Diagram, timeline

Description automatically generated

**KANBAN Board with basic process:**

In Basic Process of Kanban board will have 3 state i.e.: To Do >> Doing >> Done

* We are having multiple work items like Epic, Issue, Task…. etc. we can set any in our Kanban Board.
* On top of everything there is an Epic and under one Epic we can create multiple types of issues and under one issue we can create multiple types of tasks…
* Under one Epic we can add multiple issues as shown in below Kanban Board (just click on the three dots to add issues to this Epic)

Graphical user interface, application

Description automatically generated

* Added multiple issues to Membership Epic as shown below

Graphical user interface, application

Description automatically generated

Multiple issues like login functionality, sign up, Forgot password…etc. under Membership Epic

* To add multiple tasks under one issue, click on issue on top right as shown in image below and then we can see all the issues which has been created under the Epic

Graphical user interface, application

Description automatically generated

* Now the Kanban Board will be seen something like below and then click on the three dots on each and every issue to add multiple tasks on each issue.

Graphical user interface, application

Description automatically generated Graphical user interface, application

Description automatically generated

* Just assigning multiple tasks to one issue as shown below.

Graphical user interface, application

Description automatically generated

**Fields/Characteristics available in a work item:**

There are many fields available in a work item i.e.:

**(i)Title:** Here we can enter a description of 225(max) characters or less, we can also modify this title later.

**(ii)Assigned To:** Assign the work item to the team member responsible for performing the work.

**(iii)State:** Represent the current state(column of Kanban Board) of work item.

**(iv)Reason:** Represents the reason for state change

**(v)Area:** Represents the area path associated with the product or team

**(vi)Iteration/Sprint:** Represents the sprint or the iteration in which the work is to be completed. Whenever we create a work item then it must have an end date or must have a target, like what is the target to complete that work item.

**(vii)Description:** The details of the work item. Like we can write paragraphs here.

**(viii)Efforts:** A relative estimate of time required to complete a work flow. basically, for each work item there is a work flow and to complete that work flow how much time is required that is represented by Effort.

**(ix)Discussion:** This field is use to add a comment to give or ask the information (or) to start a discussion regarding that work item and if particularly you want to ask someone for something then we can tag that team member in discussion using @, like wise if we mention **#** then we can see all the work items related to the project and can ask to someone related to that work item.

**(x)StartDate:** Suppose we have created a work item and the work has not been started yet, when we want to start the work or planning to start that work, we can mention that in start date.

**(xi)Target Date:** Target Date/End Date indicates the successful workflow of work item

**(xii)Tag:** it is a feature which is use to search the work item, suppose the work item is related to payment, then in tag we give as Payment, likewise we can add multiple tags to our work items (epic, issue, task…etc.)

**(xiii)Save & Close:** Finally, after filling all the details for the fields we click on Save & Close

**(xiv)Priority:** Represents a number which specify the value of this work item to business, like how important is this story to the business, if we give priority 1 to this work item it means the product/sprint cannot ship or complete without the successful build of this work item and it should be addressed ASAP. Likewise, we can give the priority as 1, 2, 3, 4…

**(xv)Details:** Here we can see the details of our work item.

**(xvi)History:** Here we see what all changes are taking place for this work item. we can see the complete history of our work item like when we change the state/start date/Tags/priority No/attachment’s…etc. all the discussions have been happened what are the questions has been asked and answered, what info is been seeking…etc.

**(xvii)Attachments:** If want to add any attachment to this work item we can attach here in this field

**(xviii)Link:** if any link is needed to get the additional information about this work item or its related then we can put the links here in this field.

**(xix)Follow/Following:** If we want to receive an email for each and every state(like To Do>>Doing>>Done) then click follow then will received an emails for each and every state or any change we do in this work item.

**Backlog:** A Backlog is a collection of work items which will be used for future development or a backlog is a kind of collection where we create all the work items which we think that is required for development in future, will find

**Backlogs** **under Boards services** in Azure Devops projects

(or)

A collection of all those work items before starting the development is called as a **Backlog**

**There are two types of Backlogs:**

**i.e.:**

(i)Product/Project Backlog

(ii)Sprint Backlog

**(i)Product Backlog:** Product Backlog is an order list of everything that is known to be needed in the product, everything that we think that is required for the development of our project/sprint then create a work item and the collection of all those work items in an order list.

**(ii)Sprint Backlog:** Sprint Backlog is a collection of work items which are in TODO state. When we assign multiple work items to a particular Devops Engg and b’coz of any reason that developer unable to start/complete the work item then everything which is in pending state then these are called Sprint Backlogs.

**Used of Backlogs:**

**(i)quickly defined the task (user story, bug, product backlog…. etc.):** when we get a requirement for our product and based on that requirement will create multiple types of work items and sometimes the work item may be a bug or improvement, so quickly defining the task is very easy in the Backlog.

**(ii)Manage priority:** What thing is valuable for the business, what is the top priority feature which needs in the development for the product that we can manage from the Backlog.

**(iii)Add details and estimate to the backlog items in advance:** As each work item needs a detail Product owner(PO) or any other member which is responsible for the creation of work item that can write the detail of each work item prior to the development, so when the development will be started everything will be clear to the developer as what exactly he is going to develop in that particular work item.

**(iv)Quickly assign work items to team members:** When we start sprint planning meeting we can assign all the work items quickly to the concerned developer as every thing is ready in the Backlog we just need to pull the stories and work item in the Backlog and can simply assign to the responsible person

**(v)Group work items in a hierarchy:** All the things which need to be combined in a particular group then we can create an Epic for that and we create a hierarchy for all those work items.

**(vi)Procrastination of work to estimate what can be delivered within a sprint:** If we have the Backlog ready then we can pull all the work items from backlog to our sprint and based on that work item we can identify what exactly we are going to deliver in a particular sprint

**(vii)Save time in Sprint planning:** It saves lots of time in sprint planning if all the backlog items are ready.

**Implementation of Backlogs with pictorial representation:** Goto project>>Boards>>Backlogs>>New Work Item(that could be either Epic/Issue)>>Add to top

Graphical user interface, text, application, email

Description automatically generated

* We can create issues under Epic (as shown in below image) or as standalone issue.

Graphical user interface, application

Description automatically generated

* We can also add multiple tasks under one issue as shown in below image.

Table

Description automatically generated with medium confidence

**What is a Sprint/Iteration:** Sprint is a period of time in which a scrum team works to complete amount of work defined in the sprint, before we start the sprint the scrum team will have sprint planning meeting and the complete team will sit and discuss what are Backlogs, stories, items epics that we have to deploy in this sprint and is there any dependencies or backlogs that we have to consider on top based on business values.

* When we create a new project with basic process then by default one sprint is created for us and that Sprint name is Sprint 1 by default.
* To view all the sprints, we can click on sprints one left side under board services and on right side we can see what all the sprints available under our project and as shown in below image.

Graphical user interface, text, application

Description automatically generated

* We can also add multiple Sprints under Planning panel as also shown in above image.
* When we create an Azure Devops projects with Agile process then each Sprint must have the Start Date & End Date to assign the start & end dates to our sprints we can click on set dates on top right
* Once we set the start & end date to our sprint/iteration then we can see the start & end date to our sprint.
* To create a new sprint just click on New Sprint under Planning panel and pass the details of the sprint like Name, Start Date, End Date & Project name and finally click on create, like wise we can create as many sprints we want as shown in below image.

Graphical user interface

Description automatically generated

* Now to drag and drop the work items(like Epic, issues, task) to Sprint 2 or so then simply come to backlogs(under Boards) and drag and drop the work items to sprint 2 which is present under planning panel.
* If we want to see the planning panel in Devops portal then on top right we have to click on View Option and then click on Planning.

**Creating Azure Devops project with Agile process:** When we create a project **with Agile Process** then we have different work items and those are

**(i)Bug:** Something which is missed or implemented in wrong way or logic, as every developer is a human being when we works upon a particular feature then there are chances that he might miss something or by a wrong understanding and when the tester will test the functionality he will find the problem and lock a bug against that particular problem

**(ii)Epic :** An Epic represents a business initiative to be accomplished.

**(iii)Feature:** A feature typically represents a shippable component of software; assume we are having an application which is working fine on the server and now we want to improve the user experience in that application then this is an EPIC that we have to improve the user experience in this application. What all things are required to improve the user experience in that application that will come under Feature. Now for this Epic we have to improve the user experience and hence we create multiple feature (Ex of feature: (i)we might have to improve the HTML part of the particular module (ii)have to improve the process where user has to fill some less details…etc. and under one feature we can create multiple user stories)

**(iv)Issue:** Suppose there are chances where we have to implement any kind of improvement or something which is not listed under these work items then that will come under issue.

**(v)Task:** It is a small unit of work comes under issue (Ex: changing logo on home page or creating a DB…etc.)

**(vi)Test case:** If we need to write the test case for a particular feature then we use Test Case work item for that

**(vii)User story:** Implementation of new work, for each feature or Epic will create multiple user stories and the actual development will happen in the suer stories.

* To see all the work items that we have in Agile process project of Azure Devops then we have to click on top of the project name plus(+) option as shown in below image.
* Here in Agile project we are having lots of choices of work item as highlighted above and shown in below image.
* These work items we assign to any one of our sprint as per the scope of the work item to which sprint it fix to.
* All these work items can be design and put it into different sprints by any senior team member in the team or scrum master or even any member from the scrum team, all the work items are **shown below** for Agile process project of Azure Devops.

Graphical user interface, text, application

Description automatically generated

**Work items relationship in Agile process in below pictorial:**

Chart

Description automatically generated

* Here now lets understand the relationship between these work items if these relationships are clear to us then the existence of this work items would be clear to us in our Azure Devops projects & Sprints even.
* On top of everything we have an Epic(referring above image) and under one Epic we can create multiple features then inside one feature we can create multiple user stories and again under one user story we can create multiple tasks, this is the very basic relationship between these work items.

**Bug:** When we work on user stories then definitely there will be something which will be developed in wrong way or wrong manner(in quite few cases) and when the tester will test this particular feature, he will log the bug against that particular user story, so basically we have bug in the work item and when the developer work under that bug and based on the time required and definition of that bug the developer will create multiple task under that bug that’s why we have Task under the Bug and one bug may have multiple tasks.

**Test Plan:** Test plan is something which is referred by the QA and these test plans are used to write proper test steps for a particular feature. Under Test Plan we write Test Suites, under Test Suites we can write Test Cases and under Test Cases we write shared steps and shared parameters.

**Workflow in Agile Process for User Story:**

Diagram

Description automatically generated

**Workflow in Agile Process for Bug:**

Diagram

Description automatically generated

* To view all the work items(like Epic, User Stories or features…etc) under Backlogs then firstly go inside Backlogs(under Boards)>>click on settings(on top right)>>Backlogs(left side on an open window)>>Click on Epics/Features/Boards which ever you want as shown in below pic

A picture containing background pattern

Description automatically generated

* Now if we want to add multiple features under one Epic then we can click on + Add: Feature to add multiple features and under this feature we can create multiple user stories and we can also create multiple tasks or bugs (as shown in below image)

Graphical user interface, application

Description automatically generated

**What is Repository/Repos:** It is basically an Azure Devops Service which is a container/folder which is used to store and manage our code(project code) in a systematic way, this Repo is available on the server of Azure Devops, all our entire code with all the details about all the changes in that code is available in Azure Repo

(or)

Repo is a version control tool which is used to manage the code.

**Version Control:**

Version control systems are a type of software which are used to track and manage each and every change in the code, done by an individual or a team member, like when an entire team or an individual working on a particular project on a daily bases then he make some changes in the code and if we want to know what changes has been done by whom then all these things are manage by the version control.

67.Multiple team members or an individual can work in a very efficient manner if we are working with the version control system.

68.Version control is a must use software to manage our code, when we are working in any type of project and if we have a code or if we have some data related to that particular project then we must use the version control.

**Use of Azure Repo’s/why we use Repo’s in our Devops Projects:**

* To store the code of a project, as in every project we have some code and to store that particular code we use Azure Repo’s
* Track and manage each and every change done in code by team or team members, all the team members can work on the code without any issue, basically we are having a common container for all the developers and all the members in a team can work on that particular common container without having any issue then merging of the code will be done very easily and all these can be handled by version control.
* We can create several copies(by using branching system) to work on different tasks, suppose 1st person is working on a particular feature called authentication, 2nd person is working on a feature called payments, then here all the developers can create several copies using branching system and all those developer can work independently on their own branches
* If we are not maintaining the version control then suppose we have made some changes in the code and now your code is ready for the production deployment, and deploy the code in the production now for the backup we want to store that code in file system, then will create a new copy of our code and will save it at a particular place in our system again after the new sprint its time to deploy the code on the production then will again create a copy of our code and again will save the entire code in a separate place….so basically with each build will create lots of copies of our code and we are saving lots of copies in our system and this leads to loss and wastage of memory of our system and if we are using version control then there is no need to create multiple copies for each build we can handle all these things using the Tag feature which is provided by version control.

**Who should use Repo’s:**

* Everyone who has some data(code, file, image…etc) related to a project.
* If an individual is working on a project, then he/she also must use version control.
* Every team to manage their code.
* Scrum Master or Product Owner(PO) can also use Repo’s when he wants to track their changes in the report or to manage their reports and documents.

**Types of Version controls in Repo’s:**

In Azure Repo’s there are **2 types** of version control

**(i)Git (Distributed version control system):** This is the **most popular** version control which is used in today’s software development.

**(ii)TFVC-Team Foundation version control (Centralized version control system)**

**How TFVC (Centralized version control system) works:**

Timeline

Description automatically generated with medium confidence

* As shown in above image we have a repository/container where will put our code and assume this repository is available on Azure Devops servers(somewhere at Azure Devops servers) and now we can access this repository using Azure Devops portal, and now we got a new developer in the team and he has to work on the code which is available in the repository folder/container…etc assume this repository is a big folder where the entire project code(which we are developing from 5-10 years) is available, so ultimately the developers has to work on the code which is available in the repository folder, so now he has to make a connection with the repository, so here now he/she has multiple ways, i.e.:

(i)The code might be available under developer laptop/machine

(ii)The code might be available in the repository/repo

* So in above 2 ways, in either ways we have to create a connection b/w Repo & the developer system, so right now we are saying the code is available in the Repository and now a new(one more developer) has join the team, now this developer will clone the entire code from repo to his own local PC, so what’s happening here the entire code from Repo will come to the machine of the developer, he will make some changes and will simply send back those changes to the repository, like wise this process have to follow for each and every developer who is working in the project, like wise this centralized control system works(TFVC)
* But with this centralised control system we might have to face some problems, example: Suppose both the developers are working on the same file, so in that scenario both the developers are making the changes @ the same file/places, since version control provides a mechanism to merge the changes and if everything is happening on the same place/file they will commit the changes and the entire code will send back to the repo since other developers are also connected with the repository, so the latest code will be send back to the developers machine and then everything will stop working.

**How GIT (Distributed version control system) works:**

* Git is basically a distributed work system and in GIT also we have a common repository and since the functionality of GIT is distributed work system means disconnected work system and this doesn’t mean we have multiple Repos on the server, means we have only one repo on the server (ex; folder/container/Repo) when a new developer joins the team and to work on the code simply he has to fetch all the code from repo to the local system and a local repo will be created on the machine of the developer this is the big difference between the centralized version control system and distributed work system Here in distributed work system the developer got the entire code from repository to local system and now if he has to make any change, then he will simply make all the changes in this local repository, so at this place both the repositories are disconnected to each other and, yes, but ultimately we have to send all the code to the main repository which is available on the server.
* Now what will happen at a place when developer thinks that his entire code from local repo to server repo then he can push all the changes from local repo to server/main repo, firstly they will fetch the entire code from main repo to local repo and make all the changes in local repo, like they will make server commits(may be 5 or 10 commits) and finally when they feel like now they have to send the code from local repo to server repo then simply push the code from local repo to server repo, this is the main difference between disconnect and centralized functioning.
* In centralized when we are making any changes in the file and when we simply committing those changes then automatically all the change were sent back to the repository (server/main repo). But in disconnected system all the changes will remain in our local repo and it is in our hand when we want to send this code changes to main/server repository with the help of one command i.e.: Git push.
* Here in disconnected system it is not needed for us to depend on main repo we can work independently on the local repo and if we do not have internet connection then no worries as everything will be available in our local repo.
* we can switch the branches in local repo and can make all the changes in local repo but at the time when we want to connect the changes from local repo to main repo or main repo to local repo the other developer might also be working on the code and he might also have sent some code from local to main repo then in this scenario we have to get the code from main repo to local repo or vice versa….in disconnected system all the developments from every developer is going in disconnected manner that’s the advantage of GIT(distributed system).

**GIT & TFVC Common Features:**

(i)Both are used to manage the code

(ii)Both can be used by individual or team

(iii)Both provides branching system

(iv)Branching systems are very important when we are working with the version control and both the version controls provides excellent feature.

(v)Both provides history of changes(suppose on a particular file both the developers are working and they are making some changes in the file then with the help of one feature which is called history we can find out what are all the changes made in that particular file.

(vi)Both are used to merge the code.

**Which version control we should use:**

(i)Operations in GIT are fast as everything happens in a distributed mode.

(ii)In GIT, entire replica (i.e.: local repo) is available on developer machine, while in TFVC it is not available(here everything will be managed at a centralized place, so the replica of the code will not be available in TFVC)

(iii)Merging, Pull request, code review is easy in GIT

(iv)Git is used by most(99.9%) of the software development team.

|  |  |  |
| --- | --- | --- |
| **To download Git** | **To Download Tortise Git** | **To Download VS 2022** |
| <https://git-scm.com/> (or)  [Git for Windows](https://gitforwindows.org/) | <https://tortoisegit.org/download/> | [Download Visual Studio Tools - Install Free for Windows, Mac, Linux (microsoft.com)](https://visualstudio.microsoft.com/downloads/) |

* The name of the repository is coming from the project name when we create a new project immediately repository with the same name of the project created automatically.

**Repository pictorial format in Azure Devops:**

Graphical user interface, application, Teams

Description automatically generated

* Now there are multiple ways to clone a repository in our local system, the git commands to clone or to push our projects from local repo to server repo as follows…

**Step1:** goto your local laptop(which is your local repo) create a folder(with any name) and open git bash.

**Step2:** use the command **git init** >>this command is to initialize the git.

**Step3:** **dotnet new mvc**>>this command is to generate a new mvc project in our local repo.

**Step4:** **git add .**>>this command is to add the file in our gilt repo.

**Step5:** **git commit -m "Created a new project"**>> this command is to make a commit of all the files and folders in our gilt repo/repository.

**Step6:** **git push --set-upstream ServerRepoURL master**

The above command is to push the complete project(folders & files) from our local repo to server repo.

* So, basically the meaning of gitinit command is we are creating a connection b/w Azure devops repository & our local system(local repo) so with this we make changes in our local repo folder like will create new files or update the files or delete those files and then will sync all the changes from our local system to Azure Devops repository.
* There are 2 ways to add the code in repository, either we can add new files from the server or from the local, now to create a new project in this path E:\devAzure\azure%20sample%20project for a .net core project then we use command
* **dotnet new mvc** (type this command in Git bash) and then a new dotnet core project of mvc will get created in above folder path(which is our local repo actually) with multiple files and folders.
* Now add the mvc project files to local repo by giving this command in git bash
* **Git add –all/git add .** >>for all files(if we want to add all the files in our local repo)
* **Git add --file\_name** >>for one file(if we want to add one file in our local repo).
* Now we have to make a commit, and commit is a command which has an information about a particular change, what change??. Basically a commit has a unique id also a commit has a message like what changes we are making in this particular change also the information of all the files where we are making those changes.
* So right now we are adding lots of files & folders in the repository and after that will make a commit
* The command to commit the files and folders in our local repo is
* **Git commit -m “created new dotnet core project”**
* We have to now use push command to push all the changes from our local repo to server repo, we can make N number of commits, but when we want to update our server repository and when we are using git push command then all these commits which we have made will send from local repo to server repo.
* **Git push**
* So finally with Git push command we are moving all our code from local repo to server repo. Hence using Git Bash approach we have finally placed all the mvc dotnet core project files in Azure Devops repo.
* Now to update the local repo from server repo we use git pull command after going to that path(E:\devAzure\repowithtortisegit) in git bash
* **git pull followed with server Repo URL** (after going inside the repo on the extreme right top will find clone, click on that, and will find the link copy the link for HTTPS and paste it in Gitbash) as shown example below
* Git pull <https://Eclasses1515@dev.azure.com/Eclasses1515/A_Project_With_Git/_git/My_First_Repo>
* when you are having a project in your local laptop>>local repo
* when we move or push this project from our local laptop to devops repo>>server repo

**Adding & Enhancing the existing files @ local Repo and pushing back to Server Repo:**

**1st Business case scenario:**

The 1st developer pulls the code from Server repo and make the changes in the existing code, add a new files…etc in his local repository (local laptop) and pushes the code back to Server Repo and to do this he performs the below steps.

**Step1:** Firstly the code was in server repo

**Step2:** Developer pulled the code in his local laptop

**Step3:** He made the changes/add the code in his local laptop

**Step4:**He opened git bash in his local laptop

**Step5:** He performed the below commands

**git init**

**git add .**

**git commit -m "pass comments as per your understanding”**

**git push --set-upstream ServerRepoURL master**

and Finally, he could able to see the code in the server repository.

**2nd Business case scenario:**

Now a 2nd developer pulls the code from Server repo with all the changes what the first developer has done from server repo after a week or month time and he also starts making the changes in the existing code, add a new files…etc in his local repository (local laptop) and pushes the code back to Server Repo and to do this he performs the below steps.

**Step1:** Developer is pulling the code first from Server repo

**Step2:** He is making the changes in the code as per the Business requirements

**Step3:** He can either add the files or can make the changes in the code or to an existing file he can add the code

**Step4:** He performed the below commands

git init

git add .

git commit -m "pass comments as per your understanding”.

git push --set-upstream ServerRepoURL master

**Step5:** Finally, now he should be able to see his code in the server repository(after refresh)

**Adding files and folders @ server repository direct from Azure Devops:**

* If we want to add folders or files directly @ Azure Devops projects as shown below

Timeline

Description automatically generated with low confidence

* When we are creating a new folder under folders then we cannot create only folder, we must have to add one file in that folder.
* To add the content/add the code in an existing file(as shown below a piece of code) then we open the file click on Edit add the code and click on commit, to see the No of commits that we make to any file we click on History.

**Sample code to add in Devops Pipelines:**

Class test

{

//we write logic here to fulfil the business requirement

}

* if we want to download the entire project from server repo then select the repo>>click on 3 dots (extreme top right)>>download as Zip
* IF we want to upload a single file from our local to server then select the particular folder on which we want to upload the file>>click on 3 dots(beside folder on root folders pane)>>upload file.

**Branches in Azure Devops Projects:**

* When we create a new repository, add some code into that repository then first we create a new commit then there are few details which are related to this particular commit (like id, user info, who has made the changes, name of that user, the date and time of that particular commit, details of changes…etc.).
* Git is tracking the changes with the help of a particular pointer, basically it will create a snapshot of all this information of all the changes, all the commits & storing all the snapshots in its own Database. All the changes which we are making in the repository those changes are stored in the git DB & all the files will be also stored inside the git repository.
* Branch is nothing, it is just a name to the pointer (pointer keeps the track of all those commits which we are making in the repository).
* When we create a new repository then automatically a new branch is created for us and the name of that particular branch is master & if there is no branch in our repository then there is only one branch we are having and that is master.
* If we are having no other branches and if we are making many commits on our different files then all those commits is happening actually on Master Branch.
* Let us we have got now a new requirement like to do a new POC using the existing code then simply we can work on the master branch, we can make the changes for the POC’s, and we can push those changes and if we work on this manner then there are many problems will face, like in the POC code there might be some broken changes which we push inside our master branch and that will break the entire code in our repository and to avoid this breakable changes inside our code we can create a new branch from this particular branch(master).

Arrow

Description automatically generated with medium confidence

* Now this new branch is just a new pointer which has information about previous commits so at this point when we create a new branch then we are having 2 pointers’.: 1st pointer is still pointing to the master branch and the 2nd pointer is pointing to the new branch and when we create a new branch, we have to give a meaningful name to that branch.
* Now to work from Master branch to POC branch we have to switch our branch and point to POC branch & here making commits in POC branch, since now we are making changes in POC branch the code which we have pushed inside the POC branch is not available in the master branch and master branch is still having the code till the last commit of its, but POC branch will have an extra information which we have done the extra commits, so here the master branch is absolutely secure and safe and un-breakable…likewise we can create one branch from another branch as per the project need and module implementation functionality.
* In the git we can create a new branch from a particular commit, in fact in git we can create a new branch from any commit which we have inside the repository.

**Merging/Creation of Pull Request:** If we want to merge one branch code after commits to another branch then we use an operation called merging, after merging what all the code changes & commits we have done we can see all those in the master branch(if we merge the POC branch and master branch)

Diagram

Description automatically generated

* Branch is nothing but just a moveable pointer from a commit, it has all the information about all previous commits.

**Why do we need Branches:**

* The main purpose of branch is to avoid breaking changes.
* To create new branch its totally depends on our work style and new requirements to implement the logic.
* Mostly when we are developing a new feature/logic/module/POC then its always a best practice to create a new branch to avoid breaking changes and this new branch we can merge with any other branch or master branch (depend again)
* In some projects we create branches per story development (suppose if we have 10 stories then for each story will create a branch and merge it with main branch after all completion, testing, building is done)
* In some projects we create branches per developer, we always think that Master branch code is always production ready branch, so every developer will have its own branch & once everything is done then finally, we can merge these developers branch to master branch.
* We can also create branch from any commit and new branches can be created from other branches.
* Git(global information tracker) tracks current working branch using HEAD(suppose we have 3 branches inside git repository, how git will track which is our current working branch and this can be track with the help of HEAD, if we are working with Master branch then HEAD will be pointing to our master branch and if we switch from master to POC branch then the HEAD is now pointing to the POC branch if we commit and push it means we are pushing those changes inside the POC branch & all these changes are track by the HEAD & this HEAD is available inside the .get hidden folder which we are having in our local repo)
* When we switch our branch then the HEAD points to the new branch

**Different ways to create the Branches:**

* We can create the new branch either from server repository or local repository
* The Default Branch Master will be created when we create a new Repository in server.
* In git bash type **git branch** and it shows total how many branches we are having for our git Azure Devops project and Star(\*) represents that we are currently in that branch.
* To see how many branches we have in our Azure Devops projects, click on branches(left side)>>All>>here it will show how many branches that we have.

**Creation of new Branch using Server Repo:**

* There are multiple ways that we can create a new branch, while creating new branch we can choose how we want to create a new branch based on another branch (or) based on Tags (or) based on Commits as shown in below pic

Graphical user interface, text, application

Description automatically generated

* Here the below pics shows that POC branch is 1 commit behind the master branch.

Graphical user interface, text, application, chat or text message

Description automatically generated

* To switch the branch from one branch to another branch there are multiple ways. i.e.; Using git, using devops portal or VS IDE
* The git command to switch the branch is **git checkout Branch\_Name**(here pass the branch name which we want to move in or switch into)
* Deleting the branch from our local doesn’t mean that we are deleting the branch from our server repo, after deleting our local branch what all the changes we have made that will be lost.

**Creation of Branch using local Repo**

* To create a new branch in local repo, open git bash in correct repo which you want and pass this command **git branch practicelocal**
* Here if we see in below image the head of the branch is pointing to **Practice local** branch

A picture containing text

Description automatically generated

* To move other branch within the local repo then pass the below command

**Git checkout followedwithbranchname**

* To check the existing branch in your local pass the below command

**git branch(where ever the astrik is pointing that branch we are in.)**

* Since we have created the branch in local repository, now to push this branch to server repo use this git command.

**git branch -d branch\_name >>** command to delete the branch locally from Git is correct and tested practically.

**Migrating branches from Local Repo to Server Repo:**

**Step 1:** firstly get into the branch in git which we want to push/migrate in server repository.

**Step 2:** git remote add origin “followed with repo link”

**Step 3:** git push -u origin followed with branch name

**Branches that can be seen in Azure Devops portal as shown below:**

Graphical user interface, text, application, email

Description automatically generated

**Pull request in Azur Repository:**

* Let us assume we are working on the master branch, from the master branch I got a new requirement for that requirement I created a new branch and suppose the name of the branch is MyPoc(imaginary) and we are working on this new branch and have committed some few changes in this new branch and our work is done and want to merge this new branch(BranchfromVS) to master branch then this can be done with the help of **Pull Request**
* **Pull Request** is used to merge from one branch to another branch with some extra additional commits (or) features implemented, if we raise a pull request then will see all the changes in a file one by one, we can also add some reviewer who can review those changes whether the changes we have done is valid or not or is there any chance to implement the coder further more. **Pull request** also acts as a code review as well.

**Implementation of Pull Request:**

Goto devops portal>>Pull request(left side)

Graphical user interface, text, application, chat or text message

Description automatically generated

**Mine:** Here we see all the pull request which are created by you or assigned to you

**Active:** Here we see all the active pull request

**Completed:** Here will see all the completed pull request

**Abandoned:** We have created a new pull request and that pull request marked as abandoned by the reviewer then all the pull request we can see here.

* Now to merge both the newly created branch & master branch we have click on New pull request on extreme top right (shown in pic above)
* After clicking on New pull request, we have to fill the below form as shown below, in reviewer we can add N number of members as reviewers and by filling all the other details and click on Create button.

Graphical user interface, text, application, email

Description automatically generated

* If we click on the files tab then here, we can see what all the changes that we have done in the files, all those files along with the changes will be reflected here. The code that we see in green back ground will be considered as a new code and the code we are seeing in red back ground will be considered as removed/deleted code
* Likewise if we click on Commit’s tab, then here, we can see what all the commits that we have done in the files, all those files along with the changes will be reflected here.
* If we want make our pull request as create as draft then in above create button click on the nob and say create as draft, else finally click on create>>approve (if any)>>Complete
* Reviewer also has multiple options to approve with below option, we can see this in below pic

(i)Approve (ii)Approve with suggestions (iii)wait for author (iv)Reject

Graphical user interface, application, chat or text message

Description automatically generated

Finally, we click on complete to merge both the branches.

**What is Pipelines (PL):**

* Azure pipeline is a cloud service that we can use to automatically build & test our project code and make it available for other developers also, when we run the pipeline and if there is any problem or issue in the code then it will notify us saying error @ so & so line and file and we have to resolve it before publishing that code to a particular server.
* Pipe line in Devops is a set of process (automated or can be triggered manually) which is used to make available your project code to developers, it helps us to send our code from one place to another place which will be available for all the users, if we deploy our code to particular server then our code will be available to all the users, fetching code from one place and deploying on another server/place which help all the peoples to access all the websites/applications and all the steps which is used to perform these kind of task is Pipelines.
* Azure pipeline is used to automatically build the process and to test(unit testing) our project code, can be used for unit testing of our project

**Process in Pipelines (PL):**

* Pipelines in Azure Devops is designed as per project need, every project has its own strategy to build the project code using pipeline, as per the need of the project we design the pipeline
* We can increase or decrease the number of steps while designing the pipeline, the pic below shows the general steps involved while running the pipelines, here if we don’t need testing as per our project need then we can remove this step, hence as per project need we can increase/decrease the steps as we want.



There might be some extra steps which might be used in the pipeline, so it all depends on the need of our project.

**Key points on Azure Devops Pipelines:**

* Azure Devops pipeline works with any programming language and with any platform.
* We can use any language like .Net .Net core, Python, C#, Java, JavaScript, Angular, PHP, Ruby, C++, ARM templates…. etc. the list can just go on…..
* In Azure devops our project code or project source code could be in version control system.
* Azure pipelines can be integrated with GitHub, GitHub Enterprise, Azure Repo’s, GIT, TFVC, Bitbucket cloud & subversions…. etc. all these are repositories where our project code can be place or stored in.

**Deployment Target:**

* We can use Azure devops pipelines to deploy our code on multiple targets (ex: Containers, VM’s. Azure Services if we have an azure account, any on-prem server…etc)
* Azure pipelines provides a complete package which is used to fetch the code from any of the repositories which are available in the market and can deploy the code on any of the servers (Dev/QA/PROD/POC…. etc.)

**Pipeline Pricing:**

For Public Repo pipeline is **always free**

For Private Repo pipeline is free for 30 hrs in a month, if we are using more than 30 hrs in a month then we have to pay extra amount to Azure Devops and that based on location where we have deployed our project and No of lines of code we are storing.

**What is CI/CD in Pipelines:**

**CI (Continuous integration):**

* CI is used to automate, tests & build the project code/projects, it is used to find bugs or other build issues in early phase.
* CI/CD are the two main important process in the Pipelines
* Here in CI Azure Devops pipeline will fetch the code from repository & build the code and runs the tests, these steps are covered in CI

**CD (Continuous delivery/deployment):**

* CD is used to automatically deploy the code in multiple stages like if we are working in a particular project then we are having multiple stages or environment like Dev, QA, Staging, Prod….etc. to maintain the quality and deploy the code.
* CD is used to deploy our code one by one with or without permissions as per the settings that we define in the pipelines to different targets..

**When CI/CD will triggered:**

**Manually:** means, any time we can run the CI/CD pipelines to build and test our code

**Automatically:** we can automate this as for

**(i)on each commit:** Suppose we are working on a particular project and we are pushing the code in a repository, so we have made the commit and push the code in the repository when we push the code then automatically CI will run and fetch the code from Repo, it will build the code & if everything is fine will get a notification that everything is successful & if there is some problem in building the code then also immediately will get notification and we have to fix the issue.

**(ii)on a fix interval(after every 5 hrs) :** We can also automate the CI/CD on a particular interval, suppose if we want to run our CI/CD at a particular time in a day @ 9PM every day or 2 times or 3 times in a day then we can also do these types of settings in Azure Devops.

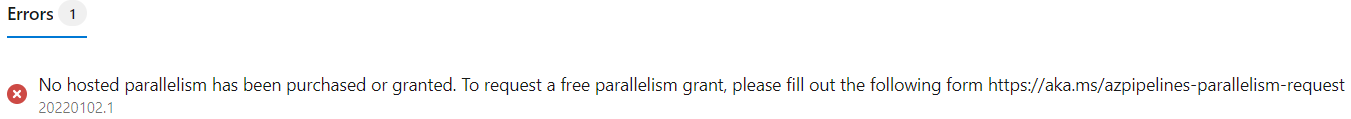
**(iii)@ a particular time:** Suppose we want to run our build today @ 7PM then we can also define that setting in Azure Devops.

* Hence we can make it manually or automatically, everything is define as per our requirement, we can do the setup as per the project need or project requirements also if we want to test the code on each commit then we can set the automatic version of CI/CD with which on each commit the code will build and deployed on the server.
* Build Pipeline is also known as continuous integration(CI)

**Running of CI/CD pipelines:**

**Note:** Whenever we want to run the build or release we always has to seek an approval from Microsoft corporation upfront(in advance), if we run the build or release without an approval then will get an error as shown in below image, here it will ask us to fill the form in link : <https://aka.ms/azpipelines-parallelism-request>

* Here we have to fill this form by mentioned all the details like Organization name, Project name, Project type…etc it might take some 3-4 business/working days to get an approval from Microsoft, once we get an approval then we can run the pipelines only for that organization & project for which we got the access….



**Note:**

After receiving approval from Microsoft, we have to disable(toggle off) below options at organization settings to see Release, Deployment Group options under pipeline else it will not show by default

1) Disable creation of classic build pipeline [Toggle: OFF]

2) Disable creation of class release pipeline [Toggle: OFF]

**Building new CI/CD pipeline in Devops:**

* To build the pipeline from the scratch, will create new organization and a new project in that organization and visibility we set as private rest of the setting set as default and then follow the steps as below

Goto to any of the folder in our local system and open git bash there and type the command

**dotnet new mvc**

Now the above command creates a new mvc project with all the required folders & files in our local system

Then below git command to initiate the git in our local repository.

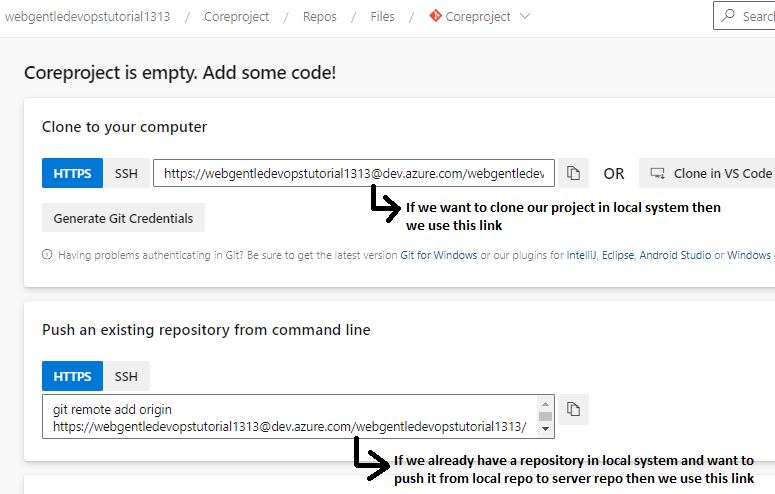
**git init**

Run the below git command to add all the files in the git

**Git add –all/git add .**

Run the below git command to commit all the files in git local repository

**Git commit -m “created a new dotnet project in local repo”**

* Run the below command to sync our local repo to server repo and there are 2 ways, either we can clone the repository to local system then we use 1st link as shown in below image (or) if we already have a repository in our local system and want to push it from there (means from local) then we can use the 2nd link as shown in below 
* So we are already having the repository in our local just have to push to server, finally copying the below link and paste it in git bash, hence when we copy and paste the above link in git bash then the below command will get generate in our git bash

**git push -u origin –all (or) git push followed with repo url**

* Then it will ask us to make a login by passing the credentials of our azure devops account and finally it will push all the files and code that we have in our local to server repo, we can see all the files and folders in server repo after we refresh the page in azure Devops.
* Now after the code has been pushed to server repo, there are 2 ways to create a new pipeline

(i)goto pipeline (left side) and click on New pipeline

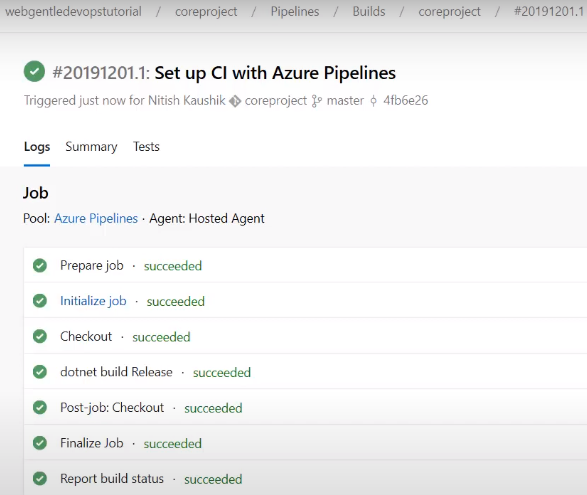
(ii)goto repo(left side) >>files>>click on Set up build(extreme top right in blue color) & if we are choosing this option then by default this particular repository will be used as source for the build

else

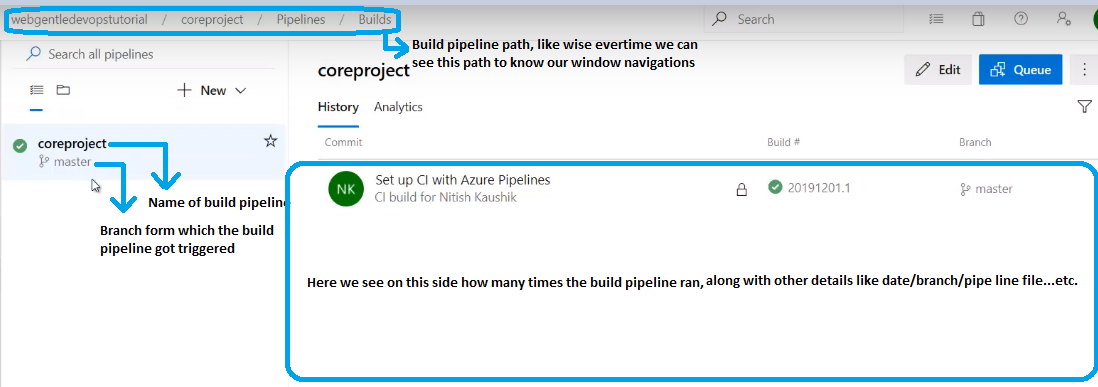
* Click on pipeline>>create pipeline>>Azure Repos Git(any version control we can choose or where ever our code is placed, as our code is placed in Azure Repo, that’s why choosing Azure Repo)>>Core project(this is our organization name and will also get the same project name as per org name that’s why choosing)>>ASP .Net core>>Now here Azure Devops is creating a new YAML file in our project

**YAML: YAML file** is a basic configuration file where we have to enter some commands and, in those commands, will do all the work that is required to build the pipeline, we can use any type of virtual machine image(window, Linux, ubuntu…etc.) to build the pipeline, Purpose is to build a new project it will create a new instance of this type of virtual machine which will enter over here in YAML it will build the project in the Virtual Machine & once the build is completed Azure devops will destroy that VM that’s why we have image of VM in YAML config file. If we are using any variables in the code then we can define those variables over here>> click on Save and run >>Save and run >>will see an agent is ready for the job.

Then will see in Build pipeline window and it will execute sequence of steps one after the other as shown in below image.



* Now in Repo section we see a new file with a name **azure-pipeline.yml** after the build succeeds.
* Now come to local repo>>open git bash>> and pass the command **git pull** to pull the new **azure-pipeline.yml** file in our local repo.
* After the pipeline got succeeded, we can see build pipeline got succeeded as shown in below image



* If we want to run this pipeline again then we can click on Queue(shown in above image top right) followed with branch and variable
* To know the details of each step(mentioned below) that get involved in build pipeline we can click on it to have the details.
* Build pipeline execution job steps.

|  |  |
| --- | --- |
| **Job steps** | **Explanations of steps** |
| **Prepare Job: Succeeded** | Here it will prepare the job for build pipeline |
| **Initialize Job: Succeeded** | Here it will initialize the build pipeline job |
| **Checkout: Succeeded** | Here it will get the code from Repo |
| **Dotnet build Release: Succeeded** | Here will get the configuration of code release |
| **Post-Job Checkout: Succeeded** | Here will do the cleaning of any cached credentials from repository |
| **Finalize Job: Succeeded** | Here finalizes the job with all details |
| **Report build status: Succeeded** | Here it will report build succeeded status |

* Goto repository files in server and purposely remove some code and commit it to test the failed build, the moment we click on commit and run the build again after some steps the build will get failed and there we have to debug why & how it got failed and it will give us full details of the issue @ which step it got failed…etc.
* Rectify the error by making the change again in the file repository code and again run the build and this time the build will get succeeded with all the step and now to see the logs of each step we have to click on right side @ the top.
* If we want to send an email of our build succeed/failed to a particular member in our team or the entire team or if we want to disable then goto the project setting>>Notifications>>Build completes: Disable/Enable, then here who all the members are part of our project they will get the emails if its Enabled or won’t get Emails if its disabled.

**There are 2 ways that we can do this notification settings.**

i.e. @ (i)Project level settings &

(ii)Organization level settings

If we want to send this build email notifications to outsider email id’s then goto project settings>>Notifications>>New subscriptions>>A build completes>>Next>>write some description as per project standards>>Delivery to :Custom email address>>Address : khiddu.m@gmail.com >>A specific team project>>Field : Status >>Operator : = >>Value: Succeeded>>Finish

**Key Concepts of Azure Pipelines:**

**Agent:**

1. An agent is an installable software that runs one job at a time and when our build or deployment runs then the system begins one or more jobs, there will be at least one job in the pipeline or more than one job, an agent is used to run one job at a time and it is used to run the job
2. To build our code or deploy our software using Azure pipelines we need at least one agent, bcoz to deploy the release pipeline or build pipeline we need at least one job and to run that job we need at least one agent.
3. In Azure DevOps we can use two types of agents

**(i)Microsoft hosted agent**: these agents are hosted and installed by Microsoft automatically, when we create a release or build pipeline then a new VM which is defined in your YAML file is created for us and these agent run on those VM’s and the basic task for this agent is to get the code from repository build it and process the tasks further.

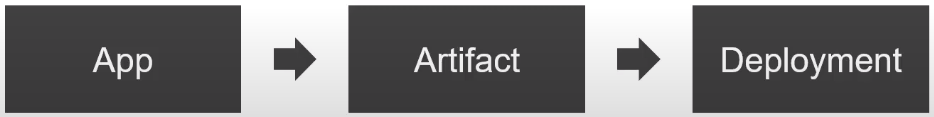
**(ii)Self-hosted agent:** these are the agent which are hosted by us, if we want to have full control on our agents in Pipeline then we can use this self-hosted agent

**Approvals:**

Approvals are set of validations which are required before a deployment can be performed. Deployment is basically a release pipeline and these are some set of validations required before a deployment can be performed. Ex: getting permission from someone in a team before the deployment on production, as first we deploy the build in Dev, then QA then staging then at the last Prod, as production is the critical environment for our projects, so we cannot deploy any code on the production and if we add some set of validations before deploying the code on production and that validation is known as approval and in the process of this approval we need permissions from specific person or more than one person and once its get approved from them the we can deploy the code on production.

**Artifacts:**

An artifact is a collection of files or packages which are created by a build run. When we run a build pipeline then there are some files which are created after running the build pipeline **Example;** we have a dotnet core application when we run this build pipeline on dotnet core application then internally its actually running a publish command and when we run a publish command on dotnet core application(an example as dotnet core application, we can use any other tools like java, python frameworks..etc) then we get some files which are required for the release or to run that particular application, Hence the collection of all those files which we get after running the publish command of dotnet core application then that is called as **Artifact** & finally these artifacts are then made available for the next task i.e., deployment



Suppose we have an application, this application could be in PhP/Java/Python/angular….etc on any other frame work which we have and once we run the build pipe line then the output of build pipeline is artifact and when we run the release pipeline on this particular application then this artifact will be treated as source and this source will be deployed on a particular server that is called as deployment. Hence the output of the build pipeline (or) a source of release pipeline is known as Artifact and these artifacts are the collection of all the files which are created from the build & everything which is required to run that particular application (ex; package or anything else which is required to run that particular application).

**Environment:**

An environment is a place where we deploy our application, when we run a release pipeline then there is a particular server where we send our code to make it available for the public, an environment is a collection of resources (VM’s, Containers, Web Apps…etc.). A release pipeline can deploy the code on one or more VM’s after the build pipeline is completed, once the build pipeline is completed then the release pipeline will work and that release pipeline will deploy our code on multiple servers and these servers, we called it as environments.

**Job:**

A job represents an execution of boundary for set of steps, all the steps run together on the same agent, Ex; we might build two configurations -x32(bits) & -x64(bits), in this case we have one build stage and two jobs(one job is for 32 bits and one job is for 64 bits), mostly importantly one job can ran only by one agent at a time.

**Run:**

One execution of pipeline (build or release) is know as run, when we execute a particular pipeline (build pipeline or release pipeline) then that is known as Run, the success or failure of a pipeline doesn’t matter in the run just the starting point of the pipeline is called as Run.

**Stage:**

Stage is used to mark separation of concerns, Ex; creating build for QA, Staging, Production….etc.so before publishing our code on production server we create some different stages(ex; one for dev environment, one for QA environment, third for staging & forth for production…)these are some basic four stages that we create in a general application, here we can increase/decrease the Number of stages as per the project need or project standards, Each stage contains one or more jobs, stage is very important an adding an extra level of validation before our code goes to production.

**Trigger:**

A trigger is a setup that tell the pipeline when to run, we can configure a pipeline when we have a

(i)New push in repo

(ii)@ schedule time

(iii)Upon completion of another build

We can create a build pipeline without creating a YAML file, means, we can use some other type of editors to create a build pipeline, like classic editor

**Creating Build Pipeline for ASP.Net Web Applications using classic editor instead of Azure Repos Git:**

**Steps:**

**Steps on high level:**

* First will create a .net project in local laptop
* We will build and run this project in my local laptop in VS 2022
* We will push this project in my Azure Devops and **open the git bash where the .sln folder is present and give the project/solution name as NareshMVC5App**(or any name as per your choice)
* We will do the Build (CI)>>continuous integration) of this project in Azure Devops
* We will do the Deployment (CD) of this project in Azure VM using Azure Devops pipelines.

To download the visual studio in our laptop use the link <https://visualstudio.microsoft.com/vs/> >> Mouse over on Download Visual Studio >> click on professional 2022(download VS 2022 professional version more precisely)>> automatically the download will start in our local laptop, after the exe get downloaded >> open the exe and install the visual studio in local laptop.

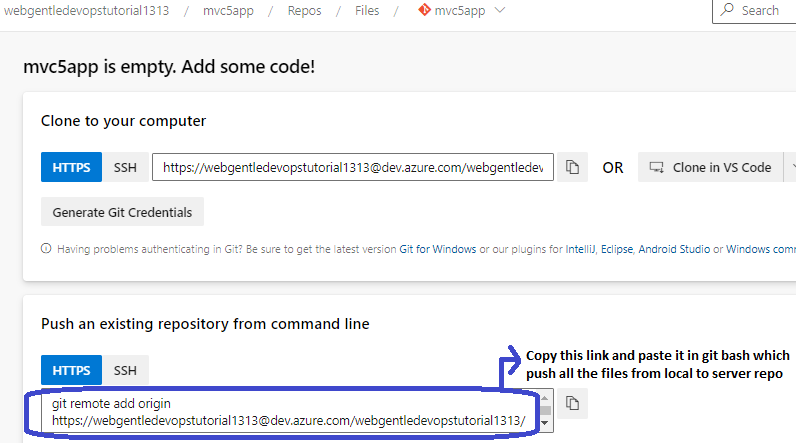
* Open VS 2022 from windows start>>Create a new project>>ASP.Net Web Application(.NET Framework)>>Next>>Fill Project name/Location/SolutionName/Framework>>Create>> uncheck configure for HTTPS(right side under Advance)>>Create.
* Let’s run this application first by clicking on top IIS Express Fire (Microsoft Edge)>>the application will get open in another window and seems the application is working fine.
* Now we will be using this application to create the build pipeline & then the release pipeline, so now let’s goto Azure Devops portal>>create a new project with name mvc5app with public visibility>>goto repo where it will be empty and now will push the mvc5 project (files, folders, code…. etc.) from our local repo to server repo
* Now open the git bash in local repo (where our project is) and pass the below git commands in sequential order

git init (to initialize the git)>>

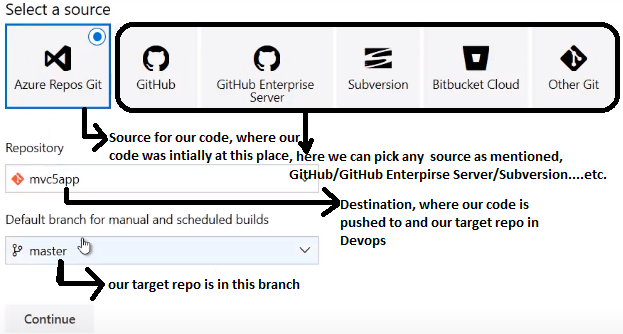
git add .(to add all the file to git)>>

git commit -m “committing all the files in git”

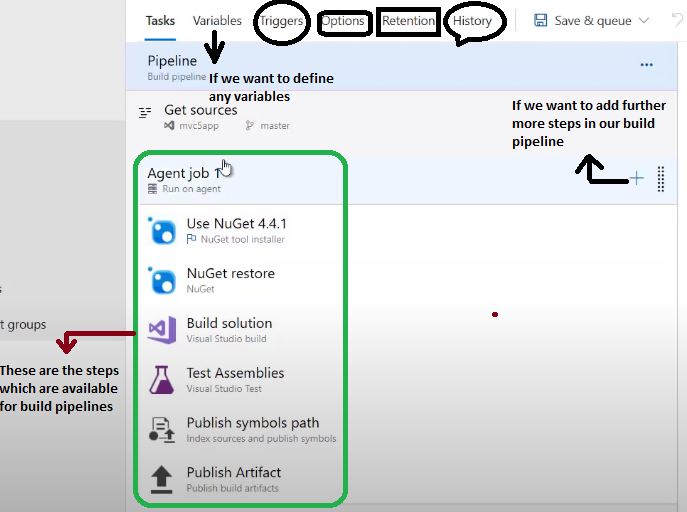
* After above commands copy the link from Repo file as shown in below image and paste it, in git bash then hit enter and now finally will see all the folders/files/code…etc in our server repository.



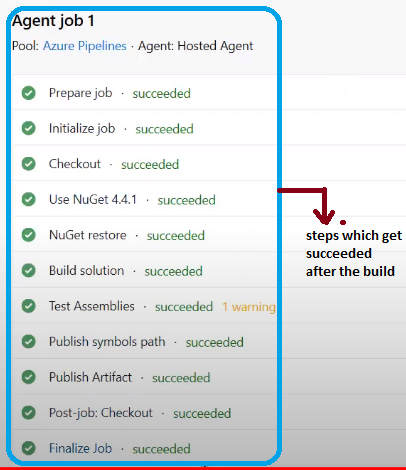
* Now the code/project has been moved from local repo to server repo(Azure Devops) after performing all steps as mentioned above.
* Goto the pipelines (left side)>>New pipeline>>**use the classic editor**
* And now when we are using classic editor then Graphical user interface(GUI) will be available for us, we don’t need to write the code in the YAML file, we simply have to choose the code in the form of tabs and everything will work automatically.
* Firstly, we have to choose where is our repository like source, destination branch…etc, and finally click on continue, as shown in below image.



* After click on continue, we have to choose the template lets choose ASP.NET(Build and test an ASP.NET web application)>>Apply>>then here we can see steps which apply on build pipeline as shown below.
* Among building the steps for Azure pipelines building the code and publishing the artifacts are the most import steps to work for release pipeline and to work on build pipeline we can use only the build portion but when we are working with release pipeline then we must use this publish artifacts task as well.
* If we are using the YAML version of Azure pipeline then by default the publish artifact is not available over there, we have to write the code manually to YAML file in order to publish your artifacts to the artifacts folder.
* If we want to see the YAML file for a particular step then click on Publish Artifacts(in the steps of build pipelines) and click on view YAML button.
* If we want to add any further more steps in our build pipeline then we can add that step by click on + symbol as shown in below image.
* If we want to add any variable to our build pipeline then we can define the variables in variable tab as shown in below image
* If we want to Enable continuous integration then we have to check the check box of Enable continuous integration as shown in below image then the build will get triggered automatically on each check in the master branch, this we can do in trigger tab in below image
* To give build description, build number format, the option related to that build can be entered here, like the Number of minutes the build job will be timed out in 60 mins(just an example), build job cancel timeout in minutes in 5 mins, this we can do in Option tab in below image
* How many days we want to retain this build we can set this configuration in Retain tab as shown in below image



* All the changes for build pipelines we can verify in history tab and finally doing all the configurations in multiple tabs we click on Save & queue
* Now our build will start and it will take some time to execute all the steps based upon the size of our project, after all the steps has been succeeded successfully will get to see the complete status of each and every step as shown in below image.



**Release Pipeline/Continuous Delivery(CD) :**

* The release pipeline is also know as CD, here will perform Release pipeline demo by deploying the code/project/application using Azure Virtual Machine.
* Now deploy a VM using Azure(<https://portal.azure.com/>) and login to the VM, then come to our azure devops portal and goto **deployment group** (left side under pipeline)>>Add a deployment group (**Deployment Group:** it is basically a connection in between Azure Devops and our Virtual Machine)>>Enter the name of Deployment group: mvc5appgroup>>Description: any description as per project need>>Create
* After we click on create will find a power shell script on right hand side>>check the checkbox as Use a personal access token in the script for authentication and click on copy script to the clipboard (to the entire script)>>Come to the VM>>open PowerShell as administrator>>paste the entire script, hit enter>>again enter>>again enter>> wait for some time and now the connection between Azure Devops and our VM has been established after running the script.
* Now come to release pipeline (left side to deploy our mvc5app code to our virtual machine)>>New pipeline>>Click on Artifacts(here we are using the artifacts from our build, if we want to use the artifacts from other places then we can use other places as a source like Azure Repo GitHub,….etc.)>>Build>>choose the project>>Source(build pipeline for a project we may have several build pipeline we have to choose it appropriately the correct build pipeline which ever we want)>>Default version: Latest>>Source alias :leave the default value>>Add
* Artifacts: it is a source for the release pipeline (here only we are checking what build pipelines we want to use to run the release pipeline) in other ways artifacts is the destination for the build pipeline, so the build pipeline will get the code from repository, it will build the solution and publish the entire code into the artifacts, then this artifact will be treated as source for release pipeline. Hence in release pipeline we are choosing the source through artifacts and in artifacts we are choosing the build pipeline.
* In general standard for a particular application, we create multiple environments (like Dev, QA, Prod…etc), we can increase or decrease the environments as per our project need, here stages refers to that particular environment.
* Click on +Add a stage>>IIS website deployment >>Apply (Here we have lots of templates to choose)>>Pass stage name: Dev>>close it by clicking on cross button>>
* Now adding the task to the release pipeline, so click on the stage>>Stage name : Dev>>Configuration type : IIS website>>Action : Create or Update(means for the first time it will create the application and for 2nd 3rd or 4th time if we run then it will update the website>>Website name : MVC5App(this name will be treated as application name in the IIS web server)>>check Add bindings>>click on three dots to update the port>>Add Bindings(top right)>>Pass the port No>>Ok>>click on IIS Deployment(top centre)>>Deployment Group : mvc5application>>Required tags if we need to pass>>time out : 15>>Job cancel timeout : 15 >>Artifacts download : \_mvc5app-ASP.NET-CI(this is our artifact name with latest)>>Check Allow scripts to access the OAuth token>>click on IIS Web App Manage(top centre)>>check Enable IIS(if IIS web server is not enabled in our VM)>>click on IIS Web App Deploy (top center) and just verify all the values all the values let it be by default what all we are having as is
* Now click on variable tab(top center), if we want to run our pipeline on different environments
* Now click on Retention tab(top center), here if we want to retain our release for some Number of days then we can do that configurations here following with Options and History tabs and finally click on Save button(top center right) and pass the folder path and comments if needed and then click ok
* Now click on Release pipeline again>>Edit(extreme top right)>>click on continuous deployment trigger in Artifacts box>>Enable continuous deployment trigger, and if this is disabled then the build will not get triggered automatically then we have to run our build manually, if we want to trigger our release pipeline automatically and once the build pipeline is completed then we have to enable it from here under Artifacts box and click on the cross button on top and finally click on save
* Now click on create release (top centre right)>>pass the version as 0>>leave all default values as is and finally click on create button>>then will see Release -1 has been created

Goto release (left side)>>click on create release (top right)>>Pass the artifact version number>>click on create and then here will after some time the release will get succeeded and now if we login to the VM and see then we find the IIS server has been hosted in the VM and MVC5 application has also been running under our VM.

* Now if we want to make some changes in our MVC5app application website in our VM then open the VS make the changes in the code, commit the code and the CI/CD pipeline will run/execute automatically and the changes that we have made can be directly seen in our website which is hosted in our VM.

**Pull Request Trigger:**

If we want to trigger our build on each pull request then we canenable that also from here i.e.; under Artifact’s box

**Steps to Implement the Release Pipelines/Continuous Delivery(CD As IAAS):**



**Learning of some more multiple stages in Release Pipeline:**

* So till now we have created a new project in Azure devops where we have added the code for ASP.net MVC5 application and then by using the ASP.net MVC5 code we have created a build pipeline and this build pipeline builds the code automatically and creates the artifacts and once the artifacts are ready then using a release pipeline we are getting these artifacts as a source and we are publishing the entire code on a virtual machine that we have provisioned in Azure portal.
* In general development projects we have multiple stages(like Dev, QA, Prod…etc.)let us assume we have development stage and since we are working on the code and to test it in a particular environment we create a development environment, we create the release on development environment, we test it & if everything is working fine then we release the code to the QA environment and if everything goes fine in QA environment then we release the same code in production environment finally, so these 3 environments(Dev, QA, Prod) are called the stages in Azure devops
* So in above lab we have created only one stage i.e.: development environment, now will learn how can we add more stages to our release pipeline.
* Setting the triggers, triggers are basically the permissions which are required for a particular stage, suppose before publishing the code to QA environment or on production environment, we want some permission from particular user from our team(like manager, team lead….etc.) and that particular user give us the permission then we can deploy the code on production else we cannot deploy
* Now to add one more stage to our release pipe line which we have done in above demo then go to release pipeline >>Edit>>Here we have one stage which is deploying our code in dev environment, here we have to add one more stage to it and that stage could be QA or Prod and @ this point we might have 2 scenarios that is for development environment we might use a VM(this we did) but for QA environment I might need to deploy the code on some Azure service or some other environment or the same machine that we are using for the dev environment, but to keep it simple we may have the same type environment that we are using for the development or we might have some different type of environment for other stages
* Now depending on the nature of the type if we have same environment then we can clone this entire stage(which we did in above demo the same stage) to create a new stage and if we have all together a new environment, just like here in dev environment we are using a Virtual Machine and for the QA environment we may need to deploy the code in some Azure Services then we have to use the add feature of the stages
* Now to add a new stage or to clone the existing stage we have to first click on the stage and then we choose clone as shown in pic below
* Once we click on clone a new stage will arise and the name of the stage will be (like Copy of Dev) as we are copying from Dev, here we give the new name to the stage like QA as shown in pic below

Graphical user interface, application

Description automatically generated

* Here if we see in above pic the Artifact as a source and from this Artifact we have one stage for the Dev(which is working fine) and from this stage we have launched one more stage and named as QA
* Now if we want to add a new stage for a complete new environment then click on Add and then choose the template based on where we want to deploy our code like if we want to deploy our code on App service deployment or want to deploy our code on Kubernetes cluster, depending on the nature of our application or depending the server requirement we can choose the template and can add the new stage accordingly.
* Now for demonstration purpose, we can add the new website on the same virtual machine for QA environment, so, for this we can clone the entire stage, to clone it first click on the stage and mouse over to nearby add and click on clone.
* Now to setup the configuration for CD (release pipeline), click on the stage>>then go to task(centre top)>>click on QA>>keep all the setting/values as is>>check the checkbox to Add binding>>click the 3 dots to and change the port No as 87(we can pass any port No)this we are doing to update the binding, and if we are using other virtual machines then there is no need to update this binding number(port No) as here in our case we are deploying our code/MVCApp to same VM that’s why we are updating this binding No
* Now click IS Deployment (top centre)>>Deployment group will keep same as we are using the same VM to deploy our code, if we are deploying our code to new VM then we have to create a new deployment group and from the drop down we have to select that new deployment group>>rest all settings keep as is.
* Now click on IIS Web App Manage(top center)>>just verify all the settings are filled and keep all settings as is.
* Now click on IIS Web App Deploy(top center)>> just verify all the settings are filled and keep all settings as is>>finally click on Save button
* Now click on Create release(top right) then will see, first the release will run on Dev Environment then on QA Environment>>pass the build CI version>>Give Release description>>click on create then finally on top we see Release has been created as shown in below image and click on this Release

Graphical user interface, text, application, email

Description automatically generated

* Lets go to release pipeline(left side)then will see two stages 1st is Dev & 2nd is QA, first the build will work on Dev then next on QA, something like shown in below image

Graphical user interface, text, application, email

Description automatically generated

* Click on Release on above image then it will navigate us to view the details of the execution of the pipelines as shown in below figure, where it is showing us Dev stage is executed successfully and QA stage is in-progress and which will get succeeded in some time and to view the log of each stage we can click on log if want to see the details, then finally we login to the VM and can see the QA website deployed under IIS in our VM

Graphical user interface, application, PowerPoint

Description automatically generated

* Now lets make a change in the code again with VS.net and commit the code then the build pipeline will get triggered automatically as we have automated the execution of CI/CD pipelines
* Now again the CI(build pipeline)/CD(release pipeline) will trigger automatically and deploy the code in our VM and we can verify the code what we have changes in our MVC5app.
* Now if we want to create the stages in parallel of particular stage then we can implement even this as well, here in our above all the stages are in queue (means one after the other, first Dev stage execution then QA execution) but if we want to run both the stages in parallel then we can desing some parallel stage also and to create the parallel stage in dev, just mouse over on dev stage at below then +Add option will pop up automatically as shown in below pic

Graphical user interface, text

Description automatically generated

* If we want to get the permissions to execute the QA, suppose we are making some changes in the code then automatically the build pipelines will get trigger and the code should be deployed on the Dev machine, but for QA deployment we need some permission for the deployment and to make these kind of permission settings then we click on the pre-deployment conditions as shown in pic below and once we click on this pre-deployment condition then a window will get popup on right side with After stage check when means after Dev Stage, QA Stage will execute and if we set to After release then both Dev stage and QA stage will execute in parallel.
* Suppose if we have multiple stages at first level (like Dev in our case) and from the completion of particular stage we want to run our current stage then we can choose those stage from Stages option as shown in pic below.
* We can check the check box saying Trigger even when the selected stages partially succeeded

Graphical user interface, application, Word

Description automatically generated

* Now to seek the approvals from someone to execute the stage we have to enable Pre-deployment approvals and then pass choose the Approvers who are going to approve, **Timeout: 30 days** means by this time this request will be in pending state and after this time this request will be rejected automatically.
* Suppose I created a new release, Dev environment is complete now I have to get the approval request for this particular QA environment release and if the approver is not approving the request in a given No of days then this request will be rejected automatically, so here we can put the Number of days as per our project standard and the maximum time is 365 days allows as shown in pic below, like wise we find many other setting in the right side window, we can configure all these as per our project requirements.

Graphical user interface, text

Description automatically generated

* Here we go to our VS.net framework and make some changes in the code like adding a new link all together in our code and as the moment we commit the CI/CD pipeline will trigger automatically and now if we see the Dev stage is completed but the execution stop to execute the QA stage bcoz it is waiting for the approval as shown in pic below, the person can approve it by clicking on approve below with some specific time mentioning in Defer deployment for later followed with time/date/time zone…etc and with some comments or he will get an email to approve it else he can click on the cancel button to cancel the deployment as shown in pic below and the QA stage will start getting executed after the approval then here finally we can see the changes in IIS web server in our MVC5App application in our VM

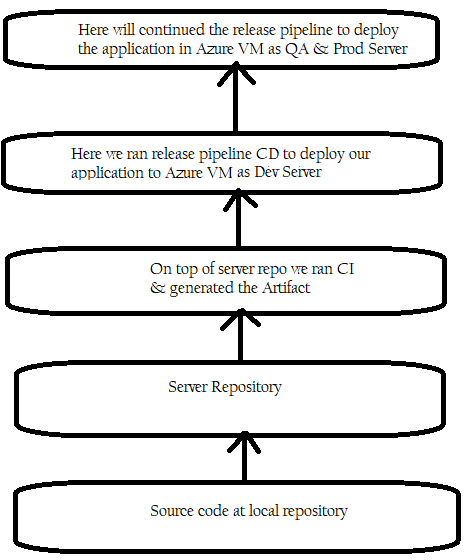
Graphical user interface, application

Description automatically generated

**Steps to Implement the multistage in Release pipeline with approvals:**

****

**CI/CD Architecture from scratch:**

****

**Deploying Web Apps in Azure App services(As PAAS):**

* Azure App Service lets us create apps faster with one-of-a kind cloud service to create enterprise-ready web app and mobile apps quickly and easily for any platform or device and deploy them on a scalable and reliable cloud infrastructure.
* *Azure App Service* is an HTTP-based service for hosting web applications, REST APIs, and mobile back ends. we can develop in our favourite language, be it .NET, .NET Core, Java, Ruby, Node.js, PHP, or Python. Applications run and scale with ease on both Windows and [Linux](https://learn.microsoft.com/en-us/azure/app-service/overview#app-service-on-linux)-based environments.
* App Service not only adds the power of Microsoft Azure to our application, such as security, load balancing, autoscaling, and automated management. We can also take advantage of its DevOps capabilities, such as continuous deployment from Azure DevOps, GitHub, Docker Hub, and other sources, staging environments, custom domain, and TLS/SSL certificates.
* With App Service, we pay for the Azure compute resources that we use. The compute resources we use are determined by the *App Service plan* that we can run our apps on.

**Implementation steps to deploy .Net Webapps in Cloud Computing(as PAAS):**

**Step1(Creating .Net Web Application in on-prem):**

Open VS-2022 in laptop>>create a new project>>ASP.Net web application(.Net Framework)>>set the path in local laptop>>select mvc frame>>create>>build the application in local laptop to verify here the application should be up and running.

**Step2(Migrating/Hosting .Net web Application in cloud):**

Search for App Services in azure portal and deploy an app service in Azure portal(ensure that you deploy an app service plan that will support the custom domain)>>now click on browse after the App service got deployed to check whether the App services is up and running in Azure portal>>right click on the solution>>publish>>Azure>>Next>>Azure App Service>>Next>>Sign in>>select existing>>Pass the creds>>click on publish>>Select here the subscription/Resource Group/App service(that we have created in Azure)>>finish>>Publish>>Now come in Azure portal inside the App service and click on browse to verify our website is published on Azure App services.

**Step3(making enhancements in the project code @ on-prem & deploying directly in cloud):**

Now in VS expand Views(folder)>>Shared>>\_Layout.cshtml>>in line No:13 instead of Application name mention as NareshIT>>click Save and Save All(at the top)>>Build the project again>>ensure build should get succeed>>right click the solution publish>>+New>>Azure>>Next>>Azure App Service(Windows)>>Next>>Select the respective RG>>select the respective App service>>Finish>>wait for some 5 mins>>Close>>Now come to the website and check the changes what we have done is reflecting in the web browser.

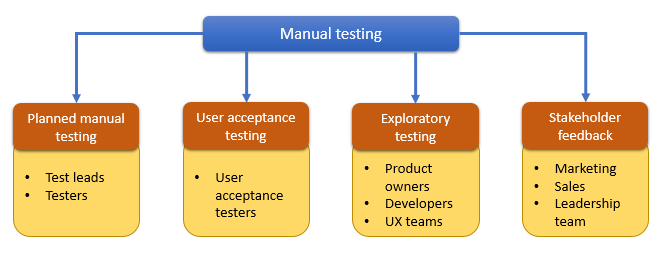
Hence here we have deployed/migrate our .Net web application on Azure App services which is offered as Platform as service(PAAS) in cloud platform..

**Test Plans:**

* Quality is an important aspect of software systems, manual testing and exploratory testing continues to be an important technique for maximizing this. In today's software development processes, everybody in the team owns quality - including developers, managers, product owners, end-user…etc.
* Azure DevOps provide rich and powerful tools to everyone in the team that can be used to drive quality and collaboration throughout the development process. They are easy-to-use, browser-based test management solution provides all the capabilities required for planned manual testing, user acceptance testing, exploratory testing, and gathering feedback from stakeholders/End-users.

**Different types of testing:**

1. **(i)**[**Planned manual testing**](https://docs.microsoft.com/en-us/azure/devops/test/overview?view=azure-devops#manual-testing). Manual testing by organizing tests into test plans and test suites by designated testers and test leads.
2. **(ii)**[**User acceptance testing**](https://docs.microsoft.com/en-us/azure/devops/test/overview?view=azure-devops#user-acceptance). Testing carried out by designated user acceptance testers to verify the value delivered meets customer requirements, while reusing the test artifacts created by engineering teams.
3. **(iii)**[**Exploratory testing**](https://docs.microsoft.com/en-us/azure/devops/test/overview?view=azure-devops#exploratory-testing). Testing carried out by development teams, including developers, testers, product owners and more, by exploring the software systems without using test plans or test suites.
4. **(iv)**[**Stakeholder feedback**](https://docs.microsoft.com/en-us/azure/devops/test/overview?view=azure-devops#stakeholder-feedback). Testing carried out by stakeholders outside the development team, such as users from marketing and sales divisions.

**Holistic approach to manual testing, types of manual testing, and persons involved.**

**Note:**

To design the test plans, test suites, test cases we always have to enable the **Basic+Test plan access** for a user @ an organization level and then at project level, this is a onetime configuration that we have to setup @ every project level if we want to design the test plans, test suites, test cases, the trial will be enabled for us for a month where we can do the hands-on on how to design the test plans, test suites, test cases for different work items that we are having in our Azure Boards.

**Steps to Enable the Basic+Test plan access:**

**Step1:** Go to the organization settings(where our project is present)>>Billing(left side)>>Basic+Test Plan: Start free trial>>click on this Start free trial>>click again on Start free trial>>then will see a message saying Trial expires in 30 days.

**Step2:** Go to the organization settings>>Users>>click on 3 dots(extreme right) for which of the users you want to provide the access>>change access level>>Access level: Basic + Test Plans>>Save

**Step3:** Now goto the project>>click on Test plans(left side)>>+New Test plan>>Design a new test plan(give any name of your choice) then a Testplan and a testsuite will also get created with the same name.

**Testing from Kanban boards:**

Start testing easily using the Kanban board in the Work hub. Add, view, interact with test cases directly from the cards on the Kanban board, and then progressively monitor status directly from the card. Developers and testers can use the search capabilities to simplify maximizing quality within their teams. In Azure DevOps, you need just Basic access to use these features.

**Here we are going to learn managing of**

(i)Managing Test Plans,

(ii)Managing Test Suites: here we can create test suites of different types as mentioned below.

(a)Static Suites (b)Requirement Based Suites (c)Query Based Suites

(iii)Test Suites & assigning configuration to test suites for different environments of Test cases for different browser, different operating system…etc.

(iv)Run with options manual tests using web browser-based runner.

(v)Shared steps and managing shared steps.

(vi)Test case parameterization using shared parameters values and non-shared parameter values.

(vii)Test Plan Grid view to add test cases within no time.

(viii)Analysing of Test results.

**Static Suite:** Static suite is an empty suite where we are adding either a new test cases or an existing test cases.

**Requirement based suites:**

Here we can fetch all the test cases which we have created for our different work items by running a query.

**Test Cases/Test Plans:**

Test case describes the steps to take when we run a test. Test suites are groups of test cases, and a test plan is the collection of test suites that needs to be run for a particular iteration or release. Test points are the pairings of test cases with test configurations that needs to be run for the test plan. A test run consists of a set of test points. The results are the outcome of running the tests in the test run

**Test Suites:**

Test suites help in organizing test cases in a test plan. A test suite can contain child test suites, helping you build a folder structure under a test plan, or it can contain test cases. Leaf test suites typically contain test cases, where are intermediate suites represent a folder hierarchy. Intermediate suites can run on Static type, whereas leaf suites can be of static suites, requirement-based suites, or query-based suites.

**Testcase Parameterization Using Shared & Non-Shared Parameter Values:**

Click on Parameters under Test Plans>>click on **+** symbol (@ top) and then an excel sheet like get opens for us, here we fill the details of the users, like below.

|  |  |  |
| --- | --- | --- |
| Username | Password | Browser |
| Rahul | Test@123 | Chrome |
| Rajesh | Test@123 | Mozilla |
| Rakesh | Test@123 | Microsoft Edge |

And after filling these details click on Save by clicking on top above>>now click on Test case pane(extreme right top) to switch **on** >>click on + symbol(to add a new test case)>>give the title to the test case and fill in Action & Expected result as

Action Expected Result

Enter @validusername Succeeded

Enter @validpassword Succeeded

Use @Browser Succeeded

* And if we see below its creating a table for us and we have to fill this table by choosing the username, password and browser accordingly and click on Save & Close(@ top right side)
* Click on Test suites>>click on extreme top right and say Add existing test cases>>window will open>>Run query>>and here @ bottom we find Shared Parameter Test>>click on Add test cases(@ bottom)
* Click on select the Shared Parameter Test>>Execute>>select the Shared Parameter Test>>Run for web application(extreme top right)>>a window will get open and on top we can see the Iterations 1, 2, 3(if we have give 3 users for testing, here it is parameterizing all those parameter values for Username, Password & Browser)>>select the iterations one after the other and pass/fail all the test cases by clicking on right side circle as checked>>and then finally click on save & close.
* Hence here we are parameterizing the test case called shared parameter test with different input set.

**Artifacts:**

When you are running your builds(CI/CD) and you are making the build succeeded then an artifact will get generated for your build and this artifact contains all the information about your build, the output of our build is called as an Artifact.....like

* what framework has been used?
* what agent (Microsoft or self-hosted agent) has been used
* On what branch we ran the build?
* What Image we have used?
* If there is any variables has been used?
* What are the steps we are going to execute??
* What operating system has been used to run the build…etc etc.
* what language has been used (.Net, Python, Php, Java, ARM templates...etc) .... all this kind of information we can find in an artifact and this artifact file will get generated in our repo's only if the build gets succeeded.
* Artifacts are basically a collection of files or packages which are created by a build run. When we run a build pipeline then there are some files which are created after running the build pipeline Example; we have a dotnet core application when we run this build pipeline on dotnet core application then internally its actually running a publish command and when we run a publish command on dotnet core application(an example as dotnet core application, we can use any other tools like java, python frameworks. Etc) then we get some files which are required for the release or to run that particular application, Hence the collection of all those files which we get after running the publish command of dotnet core application then that is called as Artifact & finally these artifacts are then made available for the next task i.e., deployment

**Example of Artifacts:**

Suppose we have an application, this application could be in PhP/Java/Python/angular….etc on any other frame work which we have and once we run the build pipe line then the output of build pipeline is artifact and when we run the release pipeline on this particular application then this artifact will be treated as source and this source will be deployed on a particular server that is called as deployment. Hence the output of the build pipeline (or) a source of release pipeline is known as Artifact and these artifacts are the collection of all the files which are created from the build & everything which is required to run that particular application (ex; package or anything else which is required to run that particular application).

**Public Cloud:** Now the entire IT industry is moving and migrating their applications to Public cloud, we are having many cloud vendors (ex; Microsoft Azure, AWS, GCP, Oracle cloud, Salesforce, IBM…etc.) these cloud providers are offering their cloud computing platform with public cloud concept, till now the entire IT industry, the organizations/IT firms were having their own cloud (private cloud) and now the public cloud is available for all. **Public cloud means the platform is available and accessible for all, not our resources/services & data**. We have to create an account/subscription in the vendors platform and then will get an identity & with authentication/authorization process will happen to login to our subscription in the cloud platform then it will allow us to create the resources/services which all we need as per our application and project requirements.

**Private Cloud/On-prem:** Till now the entire IT industry(from decades), the IT firms/Organizations were having their own cloud and that is called a private cloud, the IT firms were maintaining these infrastructure/Datacenters on their own expenses and responsibilities and this was **accessible only for them**. And they are responsible and accountable if some misshaped/disaster either natural disaster or manmade debacle have occurred.

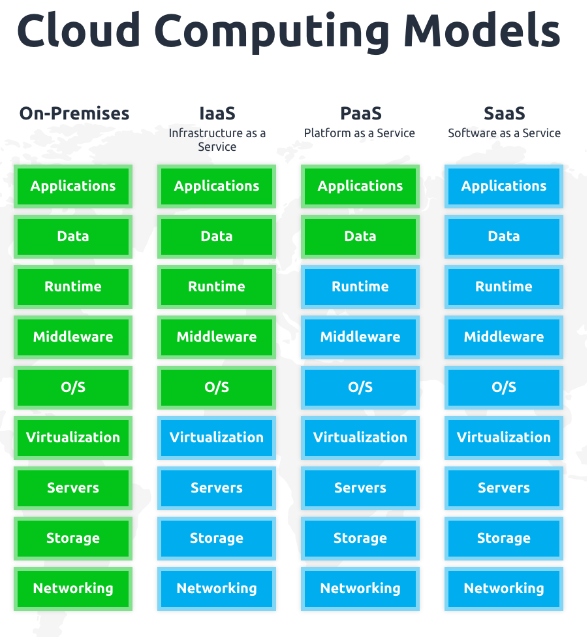
**Hybrid Cloud:** It is a combination of Public and Private cloud. (Ex : let us say our frontend application servers are hosted in cloud as public and DB servers are in on-prem, now using VPN will connect from our back end server(i.e.. On-prem) to front end server(ie. On cloud) this type of Setup/Scenarios is called Hybrid cloud)

**Advantages of Cloud Computing:**

* **24/7 availability & accessibility:** It is available & accessible from anywhere & at any time we only need to have device and internet connectivity to access the Azure resources/services.
* **Scalability:** The face of transformation is very simple and ease in cloud computing (ex; resizing of resources/workloads/services…etc.) and that can be done just with click of a mouse.
* **Security:** it is using a very high security algorithms and Hash functions to protect our data & resources.
* **Enhanced collaboration:** Just in one platform(portal.azure.com) you are going to get all the resources/services that what you need for your project or application requirement.
* **Cost effective**: It is very much cheap an economical as compared to private cloud or an old legacy system.
* **Reliable:** Cloud services are consistently good in quality with equal performance even if we perform multiple enhancements on them.

**Different cloud services/cloud service models::** In cloud service model, the cloud providers gives it services in three different ways(i.e. IAAS, PAAS,SAAS)

* IAAS >>Infrastructure as a Service(Ex of IAAS is : VM, SA, Vnet, AAD…etc.)
* PAAS >>Platform as a Service(Ex of PAAS is : SQL DB, Cosmo DB, Web apps, App services, Azure Bastion host, Azure Firewalls, Logic Apps, Function Apps…….etc.)
* SAAS >> Software as a Service(Ex of SAAS is : Skype, Gmail, FB, WhatsApp…etc.)



**What is Microsoft Azure:** Microsoft Azure is a cloud computing service created by Microsoft for building, testing, deploying, and managing applications (Business Applications) and services through **Microsoft-managed data centers.**

**Advantages/Features of Azure Cloud Computing:**

* As It is a product of Microsoft as Microsoft has launched many frameworks, tools, IDE’s, languages for the applications development and all the applications are doing great business from decades, hence clients in the market has got that faith & trust saying Microsoft products are reliable, reasonable, efficient and even economical for software applications development.
* Compare to AWS the learning/working curve of Microsoft Azure Cloud Computing is small. Azure is easy to work, easy to learn, easy to manage, there is no such pre-requisites required to learn Azure, no programming language understanding is required to work in Azure cloud computing.
* Azure is cheap as compare to other cloud providers (4-12%)
* If you are making Azure as your cloud computing partner then it is offering you MS office, WPS office, Lync, skype, share point...etc. and other platform available at cheaper cost which ultimately needed for our applications/project developments.
* As compared with other cloud providers Azure is offering you many regions/places to deploy/provisioning/creating your resources (VM's, SA's, DB's, Vnet, NSG’s, Backup’s...etc.)for our software applications.
* Azure is using a very high security algorithms and Hash functions to protect your data and resources what all is been provisioned in different regions.
* Azure is providing default encryption for all your services that you are provisioning in cloud computing platform, with which it is not at all easy for any ethical hacker to hack / hijack the resources which are hosted in Azure Cloud Computing Platform.

**Azure Resources/Services:**

A public cloud computing platform, [Microsoft Azure](https://www.whizlabs.com/blog/introduction-to-microsoft-azure/) offers infrastructure as a service (IaaS), software as a service (SaaS), platform as a service (PaaS), and a serverless model. A constant hybrid cloud, Microsoft Azure is growing in demand with approximately 90% of the Fortune 500 companies using Azure services.

The Azure cloud services are trained and created to deploy and manage even complex apps, through virtual infrastructure. It supports various programming languages, devices, databases, operating systems, and extensive frameworks. Therefore, Azure services intended for the professionals and enterprises offer all-around alternatives to the traditional means of organizational processes, with top Azure services greatly improving the performance.

(or)

**Azure Resources/Services:** Anything that you are creating/deploying as part of your application need or project requirement is called as Resources or services.

**Example of Resources/Services:**

(i)Virtual machine>>IAAS (ii)Databases>>PAAS (iii)Storage Accounts>>IAAS

(iv)Virtual Networks>>IAAS (v) Load Balancers>>IAAS

(vi)Container instances>>PAAS (vii)Container Registry>>PAAS (viii)NSG's>>IAAS

(ix)App services>>PAAS (x)Azure Repo's>>PAAS (xi)Azure Pipelines>>PAAS

(xii)Azure CI/CD>>PAAS

**Azure Resource Groups & Configuration and management of Azure Resource groups for hosting Azure services**

It is a place holder/name/folder basically which hold all our resources in azure. It is a logical container which holds all our resources in Azure portal (or) in Azure resource manager, for each resource that you are creating in Azure **must and should** be in any of the resource group. **We cannot create a resource in Azure without a resource group.**

* We can also move the resources from one resource group to another resource group (or) from one subscription to another subscription, but the tools & Scripts associated with moved resources will not work until we update them to use new resource ID’s.
* If there are plenty of resources or big size resources in resource group then it might lead a downtime (15-20 mins), so planning of moving resources should be down in non-business hours.
* If already an operation of moving resources is in-progress then at the same time we cannot proceed to move further resources to move to the same resource group. We have to wait until this operation has been completed.

**What is a virtual machine(VM):**

It is a software computer that runs an operating system(o/s) & applications in it. It also contains set of resources in it (ex : O/S disk, Storage disks, Network, IP’s, NSG, NIC… other supporting files, other networks….etc.)

When we are creating a VM then the below components we have to consider broadly

(i)Hardware

(ii)Subscription

(iii)Resource group

(iv)Network

(v)Storage account

(vi) O/s

(vii)Load Balancer (viii)VPN….etc.

When we are creating a VM via portal then we have to fill the below tabs.

(i)Basics (ii)Disks(iii)Networking(iv)management(v)Advanced(vi)Tags(vii)Review+create.

**Note:**

**We can configure availability set only when we are creating a new VM, we can’t add an existing VM to an availability set(AS).**

* Azure has 64 region(data centres) all over the globe to deploy our VM’s. as of now. And Microsoft is keeps adding for better user experience.
* The data centres are broadly classified into Availability zones/Azure Regions/Announced region.

**Difference between Availability Set & Availability Zone:**

**Availability Set**

It is a concept within a datacentre. In AS the servers in a datacentre are divided into multiple physical and logical groups. The physical grouping is called fault domain and the logical grouping is called update domain. if there are 10 racks of server in a datacentre, it's like you have 10 different fault domains. let's say in a datacentre, we have 100 servers, and all these servers are connected by a single power supply and network switch. What if that single power supply or network switch fails? Well, all the 100 servers will fail as well.

* If there are 100 servers in a datacentre, they are logically grouped into 10, 15 or 20 update domains. So, the important point to keep in mind is, an update domain is a logical grouping. Why are update domains required? Well, from time to time, server patches and software updates need to be applied. Some updates require servers to be rebooted. Now, we don't want all the servers to be rebooted at the same time. Only one update domain is rebooted at a time, So an update domain is a group of servers that can be updated and rebooted at the same time.
* Availability Zone: Availability Zone's also have the concept of fault domains and update domains. First, let's understand, what is an availability zone. availability Zone is a unique physical location within an Azure region. Each Availability Zone is made up of one or more datacentres with independent power, cooling, and networking. Not all Regions have Availability Zones, but regions that do have availability zones, have a minimum of three separate zones. You can think of each availability zone as a separate fault domain and update domain. So in a given azure region if you have 3 availability zones, then it's like you have 3 fault domains and 3 update domains. if you create three VMs across three availability zones in an Azure region, your VMs are effectively distributed across three fault domains and three update domains. If one of the Availability Zones has gone down for some reason, we still have 2 VMs from the rest of the 2 availability zones.
* Availability sets are used to protect applications from hardware failures within an Azure data centre, availability zones, protect applications from complete Azure data centre failures. An availability zone is a unique physical location that exists within an Azure region.

The SLA for Availability Set: 99.95%

The SLA for Availability Zone: 99.99%

**Availability Zone(AZ):**

* Availability zone is not available in every location or region, it is an another type of data centres which has very low latency network(n/w), means if we keep our VM under availability zones then the chances of losing the connection will be less and the n/w connection is much better.
* Availability zones protects our application and data from data center failures, it is the unique physical location with in azure region, each AZ is equipped with in-dependent power supply, network supply, & cooling supply.
* with AZ azure offers industry best uptimes for VM’s
* AZ is a combination of fault domain and update domain.
* AZ makes high availability of our application (or) data by co-locating our VM’s with another zones.

**Fault Domain(FD) :**

Fault domain has a common cooling/power and network supply for the protection of fault scenarios. If any of the things goes down(like power/cooling/network)then all the servers in FD gets impacted. FD are those machines which are getting same power supply, cooling supply and network supply in the data center.

**Update Domain(UD):** In UD we are logically grouping the under lying hardware that can undergo for maintenance (or) rebooted @ the same time, when we create a VM with in an availability set the azure platform automatically distribute our VM’s across the FD & UD. UD ensures that at least one instance of our application always be running . when Azure platform undergoes with periodic maintenance the order of update domain may not be sequential but only one update domain @ a time. In UD the machines (means VM’s) will be updated(like Windows, hardware, firmware, patching….etc.)simultaneously and as part of the update they may need to be rebooted simultaneously.

(i)if we are planning high availability of our application then we need to have multiple other VM’s (M/c’s) running @ the background to support the outage of one VM. We cannot put multiple VM’s which we want to be available@ one time in the same Fault domain & Update domain

* While creating a VM in **In Availability options** we have
* (i)No infrastructure redundancy required

(ii)Availability Zone

(iii)Availability set

(iv)Virtual machine scale set

* The fault domain and update domain depend on region & subscription where we are deploying the VM.
* **We can only put a VM in an Availability set(AS) when we are creating a VM, once the VM is created we do not have an option keep that VM in AS**
* If we are selecting an AS for a VM, then by default will get FD & UD for our VM.
* If we want to put a VM in AS later then will have to design a new VM from the beginning all together.
* When we are creating a VM then will get an option as image, this image is nothing but the O/s that we have to choose, the different O/s that we can see in Azure portal @ VM creations are Windows Server 2019 Datacentre/Windows 2016 Data Centre/Red Hat Enterprises Linux/Ubuntu Server…. etc.
* In Azure the VM’s are categorized into many types or many families like

General Purpose/Compute Optimize/Memory optimize/Storage Optimize/High performance compute…. etc….in Azure the VM’s are also categorized into series like

A series/B Series/D Series/DC Series/E Series/F Series/G Series/H Series……etc. etc.

* once the VM gets created we will get both public and private IP addresses this IP addresses will get assign to our VM’s based upon the location/region we are choosing to create our VM, the IP addresses are defined based upon the geographic location
* **If we take an RDP of a VM then the default port No 3389 should be open, unless this port No is open we cannot take the RDP of a VM & the Port No for RDP is 3389.**
* While creating a VM if we want to allow any port no then we can mention that in Public inbound ports: (i) None (ii)Allow selected ports.
* By default, will get temporary disk in our VM to store temporary data and once we reboot the VM the data will be lost there in the temporary disk.
* If want, we can create and attach a new disk after the VM gets created, we can attach either a new disk or an old disk which we have imported from on-premises.
* In networking tab for NIC Network Security Group option : (i)None (iii)Basic (iii) Advance and when we choose advance it will help us to create network security group, which will help us creating in-bound port rules and out-bound port rules for the VM Yes , means what type of connections(ports) should be allowed and what type of connections(ports) should not be allowed and this can be configured with the help of network security group (NSG)
* **Extensions :** this feature we use to install SDK’s(ex: .net framework, acronis backup…etc.) and other platforms in our VM’s. these software need additional configuration and setting that we have to do in our VMs’, post licenses purchasing.
* The charges will get started once the VM gets ready and running. Azure VM has one O/s disk and temporary disk for short term storage, we also can add additional disk.
* The size of the VM tells us the type of storage/disk capacity that we can have or we can use and the No of data disks is allowed to attach to a VM..
* **Tags :** Tags consist of name & value **Ex:** we can apply the name as Environment & value as Production to fetch all the resources of production. After we apply tags to the resources in our subscription we can fetch with name and value, Tags makes it easy to organize the resources for billing & management purpose.

**Ex of Tag** Name : Environment Value : Production

* After creating a VM if we want to add disks to a VM then we have to goto disks(left side under VM) and click on +Add data disks.

**Manage Disks/Data Disks:**  Manage disks attached to a VM to store the application data or other data that we want, data disks are registered as SCSI drivers and labelled with a letter that we choose, it has a max capacity of 32,767 GB, the size of VM determines how many DD we can attach to it

**O/S Disks :** Every VM has one attached operating system (O/s) disks and this O/s disk has preinstalled o/s(which ever we have choosed while creating the VM)and this disk has a max capacity of 2,048GB /2TB

**Temporary Disk:** Every VM contains a temporary disks and which is not a managed disk. The data on temporary disk will be lost during a maintenance event or when we redeploy a VM, when we create a VM then temporary disk is provided automatically, this temporary disk is **“D”** on windows VM’s by default and temporary storage in Azure VM is SSD.

* When we provision a VM then in temporary driver will get a notepad as “Data loss warning Read Me” and will get this notepad by default.
* Many people think that if we shut down the VM by connecting with RDP console then it stop charging us, but that’s not the case. We have to stop the VM in portal then only it will stop charging us.
* In overview of VM or any resource, we can see the complete details of the resource.

Like Resource Group/Status/Location/Subscription/Subscription ID/Computer name/O/s/Size/Public IP/Private IP….etc.

* It will stop charging us only when we stop the VM from the portal and it shows us the status : stopped(deallocated) and when we are stopping a VM from portal and it says public IP address will be lost if we stop the VM. If we start the VM then again an new public IP address will get assign to our VM, if we are assigning a dynamic public IP to our VM.
* A private IP will get assign when we create a virtual network and when we create a virtual network, we define range of IP’s, Subnet and from that subnet it is supposed to pick up the private IP.
* If we have created a VM and don’t want to give an internet connection to the VM then what will do is, we have to disable the http & https ports for the VM, we disable this ports in network security group.
* To disable the RDP of VM we have to disable the port 3389, this port No is for RDP of a VM, even if we disable the port still the VM have an option to connect to the internet.
* If we want the VM to be accessible only by private network and when we are creating that VM then we do not assigs that VM to a public IP, so when we don’t assign a public IP to a VM then it becomes a DMZ VM’s (DMZ VMs means the VM’s which only connects to the private networks)
* Inbound & outbound rules comes when we want to block the ports@ VM level, but if we do not assign a public IP then we cannot even RDP using a public internet, we cannot even access the internet on that VM…there are a lot many things which we cannot do.
* We can enable the IP forwarding for a VM from VM NIC card on NIC card setting, we can enable IP forwarding and also have to enable from registry by modifying certain registry key.
* The network virtual appliances is not connected to the internet when we don’t assign an IP address to the VM.
* In-bound & out-bound, opening/blocking the ports all these comes under network security groups.

**How does a VM will get Public IP address assigned? :** The public IP address assigned to our VM is on region wise, so the region we choose while creating a VM the public IP address will get assigned to our VM.

* All the disks that we choose while creating a VM will be a part of Storage account. The premium disk that we select while creating the VM that can be a part of manage data disks(which is managed by Azure itself)
* If we delete a resource group then what all resources we have in this resource group will get deleted permanently.
* **Size**(left side under VM)using this feature we can modify the size of our VM, we can change the virtual machine(VM) size any time.

**Properties(left side, under VM):** It contains all the information of the VM like Status/computer name/public IP address/private IP/Operating system/Resource ID/Location…etc.

**Reset Password(left side, under VM):**  if we want to create another user who can login on this VM (or) if we have forgotten the password or our existing VM’s then we can reset our passwords here and also can create a new user here who can access our VM.

* @ one specific time only one user can connect to a VM, unless we buy an additional RDP licenses, VM can have multiple RDP but we should have multiple RDP license to connect to VM.
* When we de-allocate a VM the resources allocated to the VM remains the same, only the public IP will get disappeared, even we can get the same public IP, if we create the static public IP.
* When we De-allocate/stop the VM then the software(s/w) will get shut down and it will not charge us, if we shut down the VM then still it will be using the resources, once we de-allocate the VM then we could not able to use any of the resources of the VM
* We are not able to use the resources of the VM until we start the VM, by stopping the VM it is simply moving the resources to vault.
* **RDP**>> it a utility which helps us to login to a VM, it will by default get installed in our VM.
* When we delete a VM in portal then will get a prompt as “This action will permanently delete the VM” associated resources(like disk, vnet. SA…etc.) **will not be deleted** and has to be removed/deleted manually.
* After creating a VM if we could not able to connect to the VM then go into the VM>>networking(left side)and we have to ensure that the RDP port is open.
* We can login to the VM using RDP from any internet or from anywhere in the world, until the restrictions are not provided.
* The common outages/downtime are due to power supply, network supply, cooling supply or any maintenance activity(planned or un-planned)
* Basically the purpose of creating a FD & UD is to overcome with an outage.
* We can access connect/login to our VM’s is in 2 ways i.e.: (i)RDP (ii)SSH(secure shell)
* While creating a VM we do not have an option to choose the FD (or) UD instead we have an option to chose Availability Set and Availability Zone and by this FD & UD are automatically set by Azure portal for our VM’s
* If we have one VM to address the request of the user and if anything happen to the first VM then we should have 2nd VM ready to address that request and how many 2nd VM’s we need that depends on the size of our application.

**Different High Availability option in Azure:**

(i)Load Balancer (ii)More than 1 VM or Backup VM’s (iii)Traffic Manager (iv)VMSS (v)Multiple availability zone (v)Application gateway (vi)Availability set….etc.

* We can add a Load Balancer(LB), Traffic manager, application gateway to a VM after we create a VM, **but Availability Zone(AZ), Availability Set(AS) only possible while creating a VM**
* **Availability set(AS):** is created by us and by creating this AS we are making sure that our VM’s or applications we are going to put in separate FD & separate UD and this is called as AS.
* Separate FD & UD make sure that our VMs are highly available and to overcome with outage problems.
* Before deleting a RG it will show what all the resources this RG contains below the RG name before deleting it.
* We don’t have Availability zone for all regions, we have AZ only for few regions.

**Key Points on Availability Sets(AS):**

* Availability set ensures that our VM’s are deployed across multiple isolated hardware nodes in a cluster.
* By deploying our VM’s across multiple hardware’s nodes azure ensures if any hardware or software failure happens within azure then only a subset of our VM will get impacted but our over all solution is safe and in working condition and also provides redundancy to our VM’s
* It spread our VM’s across multiple fault domains and update domains
* If we want to leverage 99.95% SLA from Microsoft then we must place our VMs inside the AS, except the VM’s having premium storage.

**Key Points on Update Domain :**

* VM’s get update domain automatically once they put inside the AS
* All VM’s with in an update domain will get rebooted together.
* Update domains are used for patching of the VMs.
* Only one update domain will get updated @ a time.

**How many FD & UD we can have??**

In Azure Resource manager portal, we have 3 FD & 5 UD, but we can also upgrade/increase or UD up to 20**(depends on the region where we are creating our VM).** VM’s are assigned sequentially in the update domain and fault domain.

We have to create a VM in the same resource group as Availability set, we cannot set different resource group for VM’s and availability set. And we should also create a separate storage account for each VM.

**Vertical Scaling/Scale up:**

Addition (or) adding an additional resource to an existing VM like (Disk, CPU, RAM, increasing the VM size …etc.) is called scale up or Vertical scaling.

**Practical implementation of Vertical scaling:**

Create a new VM>>Login to VM>>Server Manager>>File and storage services>>Disks>>Here will see the new disks>right click and say initialize>>again right click and say new volume>>Next>>Next>>allocate the disk size(ex:4GB)>>Next>>pick the drive letter>>Next>>Next>>Create>>then finally come to this PC window and refresh to see the newly created disk

**Expanding the existing disk size in Azure VM:**  There are many scenarios in our real time projects that we received a request from clients to expand the size of existing disk of a VM or attach and launch the new disk in an existing VM’s, as the project & DB sizes grows up over time we have to enhanced the resources of our Azure workloads and this activity comes under vertical scaling.

**Steps to expand the size of existing Disk in Azure VM’s:**

Stop the VM>>goto disk>>click the disk which we want to increase the size(OS disk or data disk)>>click on size+performance(of left side)>>increase some 10GB on existing disk size by mentioning as 138GB(example) in Custom disk size(GiB) (or) select the size we want>>Resize(@ the bottom)

Start the VM>>Connect the VM>>if we see in VM still the drive is showing the old size>>open run(windows+r) and type diskmgmt.msc>>ok>>right click on allocated part>>extend volume>>Next>>Next>>Finish>>that’s it done.

Now we see in our VM the C drive will show and extended size, same configurations we can apply for Data disk in azure Virtual Machines.

**Azure Key Vault:**

*Azure Key Vault is a cloud service for securely storing and accessing secrets(passwords).* A secret is anything that we want to tightly control access to, such as API keys, passwords, certificates. Key Vault service supports two types of containers: vaults and managed hardware security module (HSM) pools. Vaults support storing software and HSM-backed keys, secrets, and certificates. Managed HSM pools only support HSM-backed keys. See [Azure Key Vault REST API overview](https://learn.microsoft.com/en-us/azure/key-vault/general/about-keys-secrets-certificates) for complete details.

**Why we use Azure Key Vault:**

Centralizing storage of application secrets in Azure Key Vault allows us to control their distribution. Key Vault greatly reduces the chances that secrets may be accidentally leaked. When application developers use Key Vault, they no longer need to store security information in their application. Not having to store security information in applications eliminates the need to make this information part of the code. For example, an application may need to connect to a database. Instead of storing the connection string in the app's code, we can store it securely in Key Vault.

Our applications can securely access the information they need by using URIs. These URIs allow the applications to retrieve specific versions of a secret. There's no need to write custom code to protect any of the secret information stored in Key Vault….for further information about Azure Key Vault please refer the link [Azure Key Vault Overview - Azure Key Vault | Microsoft Learn](https://learn.microsoft.com/en-us/azure/key-vault/general/overview)

**Vault URI:**

If we are having a web application and DB and the DB password is stored in our Key Vault and we want our application to get the DB password, connect and communicate with the DB and for this in our application setting will give the Vault URI .

Access policies (left side)>>Here will see the global admin user details of our subscription who has created the Key and have admin access rights on our subscription.

Access Configuration (left side)>>Here we can see in permission model as

(i)Vault access policy>>By default Key vault will get created with this permission.

(ii)Azure role-based access control.

**Keys:** Here we can store keys example if we want to encrypt the disk of our VM then we can store such keys here to do the VM disk encryptions.

**Secrets:** Here we can store the passwords, example a DB password. Profile password, SharePoint site passwords…etc.

**To store the secrets in key vault:**

Click on the Secrets(left side)>>Name: NareshDBPassword (this is just a secret name not a userid or username or DB server name)>>Value: Shaikpet@123>>Create

**Certificates:** If we have applied SSL certificates on our webapps either on App services or on Azure VM’s and if we want to retrieve that certificate from key vault then such certificates, we can store it here in Azure key vault.

After creating a keyvault and adding a secret in the KV and if we login to the Azure Subscription with different user credentials the user unable to see the keyvault and other resources in the subscription and now let us give the owners permission to that user at subscription after that if we login with the user credentials then the user can see the resources in that subscription along with key vault but not the secrets in the key vault.

**Now in order to access the secrets/keys/certificates from key vault the user must be added in Access Policies with some permissions** then after that if we login with that user access then we can see the secrets with that user credentials.

**Azure Virtual Machine for Deployment:** If we want to retrieve any secret/key/certificate from Virtual machine then we can use this option.

**Azure Resource manager for template deployment:** If we are deploying a resource through ARM templates and integrating the key vault in ARM template and wanted from key vault that any key/secret/certificate to retrieve from ARM template then we can use this option.

**Azure Disk encryption for Volume encryption:** At the time of disk encryption for Virtual machine and we want the key to retrieve from the key vault for disk encryption for our VM then we can use this option.

**VM Re-deploy:** Re-deploy is one of the feature we have for Azure VM’s, if we are unable to connect/login to the VM, and we have done the troubleshooting and found nothing as why we are not able to login to the VM or there is some issue/errors that are occurring in our VM’s related to network ports, inbound outbound or application related issues…etc. then will do the VM redeploy, when we do VM redeploy then all the configuration and associated resources of the VM will remain same, like LB, RG, ASG, AS, NSG, Subnets….etc only the data which we are having in our temp drive will be lost, if we are having dynamic IP allocation for our VM then after Re-deploy the IP will also get changed/updated.

When we do Re-deploy it is going to restart and re-launch our VM in some other node with in the Azure Infrastructure, the data, configurations, resources all will be as is and safe, no worries, only the data in temp drive will be lost, if all seems good but still not able to login/connect to the VM then finally the last option we have is Re-Deploy

**VM Re-Apply:** When we are performing any action on the VM like start/stop/restart…. Etc. and that got failed then in that condition we can do Re-apply

**ARM templates/JSON templates:**

The resources which we create in azure are based on JSON template….every resource that we create/provision is based on template and that is JSON template.

**ARM Templates : Azure Resource Manager templates:**

**ARM = Azure Resource Manager.**

**Key points on ARM templates:**

(i)ARM templates specify the deployment parameters and helps the admins/developers/Devops Engr’s to configure the settings to create the resources(VM, SA, Vnet, CDN, LB, VMSS…etc)

(ii)Whatever we are creating in Azure cloud everything(resources and services) is based on ARM templates only.

(iii)Whatever we create/deploy using Azure cloud portal then behind the scenes cloud portal will create /write ARM templates scripting. We can get this ARM template scripting in export template option.

(ii)ARM templates are used for automation like creating multiple resources of same/different configuration.

(iii)when we click on export template on left side for any resource like(VM, SA…etc.) then we can see the coding of template in multiple languages(ex: JSON)

(iv)We can set the parameters of ARM templates(ex: VM name, VM image, Location…etc.)

**(v)Schema/parameters/content version…..These are mandatory fields of ARM templates.**

(vi)we can download/Add to library/deploy this ARM templates.

(vii)If we must create a VM which are 10 and identical in nature or different, with same Operating system(O/s) system or different O/s, so we can fulfil this requirement with ARM templates.

(viii)ARM templates scripts are case sensitive, if there is a single miss of comma, bracket, quotes, colons….etc then this script will get failed in our Azure portal.

**Syntax of ARM template:**

{

"$schema": <https://schema.management.azure.com/schemas/2015-> 01-01/deploymentTemplate.json#",

"contentVersion": "",

"apiProfile": "",

"parameters": { },

"variables": { },

"functions": [ ],

"resources": [ ],

"outputs": { }

}

**VM creation using ARM templates:**

This template allows you to create 2 Virtual Machines under an Internal Load balancer. This template also deploys a Storage Account, Virtual Network, Availability Set and Network Interfaces. The Azure Load Balancer is assigned a static IP in the Virtual Network and is configured to load balance on Port 80.

* To validate our JSON code we can go to <https://jsonlint.com/> paste our code in the tool and click on **validate JSON** and can check whether all is good and no errors in our code.
* If all the code is appropriate/good then it will pop up a message in Result panel saying **Valid JSON** with green popup Else will get an error.
* This tool validate the code even if it is for 10 VM’s even or 10000 lines of code.

**Deploying Linux VM in Azure Cloud Computing:**

We can also create virtual machine of different Operating systems in Azure cloud computing, to deploy a VM of Linux O/S have to follow the steps below.

**Implementation Steps of Linux VM’s.**

1. **Subscription:** pick any
2. **Resource Group:** pick any
3. **Virtual machine name:** give any name
4. **Region:** choose any region

**Image:** Ubuntu Server 20.04 LTS -x64 Gen2(20.04 is just a version of Linux O/s)

1. **Authentication type:** Password (choose password)
2. **Username:** Myvm
3. **Password & Confirm password:** Shaikpet@123
4. Fill rest of the values accordingly or leave to default and deploy the Linux VM
5. After the Linux VM gets deployed>>open the cloud shell>>Bash>>Create a storage account>>wait for some 2 mins, until the bash terminal gets ready>>pass the below commands.

10) **ssh username@VMPublicIP>>hit enter** (here pass the VM username and VMPublic IP accordingly and this command is to connect the Linux VM)

11) **Are you sure you want to continue connecting:** Yes(type yes and hit enter)

12) **Give the password** (what we gave @ the time of VM creation) and hit enter.

Now we successfully connected to Linux VM and here now we can execute different commands to perform the operations on Linux servers.

**Linux commands handbook for practice:**



**What is Azure Storage Account/Storage Services:**

Azure Storage is Microsoft's cloud storage solution for modern data storage scenarios.

Azure storage account contains all our storage data objects like blobs, files, queues & table storage account provides a unique namespace for our Storage data that is accessible from anywhere in the world over HTTP or HTTPS.

Container/Blobs and File storage are the main concepts on which Admin/infrastructure specialist works on.

Table and Queue storage are used and worked by Developers.

**Container/Blob (Binary large object) :** It is used to store binary large objects, In blob we can store unstructured data and it is a part of storage service.

* In Blob we have different types of storages. i.e.: (i)Page blob. (ii)Append blob (iii)Block blob.

**Page blob:** it is used to keep the VM disks, the data which we are using very frequently keep under page blob, its pricing is cheaper then block blob, basically we are storing here UN-structure data(ex: video files(2-3 hrs), VM disks, DB files, Unstructured DB files....etc.)

**Append blob:** Used for logging purpose such as VM logs, diagnostics logs...etc.

**Block Blob:** It gives us the URL access of the data which helps us to keep the data such as docs, videos, images, pdf’s…etc.

* Storage account (SA) is just a name space (or) place holder once we have or created a Storage Account (SA) then we will get the access of Blob/Queue/Table/Fileshare storages…etc.
* When we are creating a Blob & file in Storage Account then we create a container, which is nothing but a folder.
* Normally we upload .vhd files in page blob. When we create a page blob or block blob and if we go into it then will get the URL, this URL is private by default and it won’t open in browser.
* When we create a SA then we basically fills the below details.

(i)Subscription

(ii)Resource Group

(iii) Storage account name

(iv)Location

(v)performance : standard & premium

(vi)Account kind

(vii)Replication

(viii) Access tier

* Cluster technology is being used by MS in azure to make sure our data is being replicated, means if one server is down then in other regions/zones, the other servers are there as backup to provide the data to us.
* Depending on requirement we can choose our offering and later on we can upgrade as well.

**Replication:**  While creating a Storage account we have an option called replication and in replication we have 4-5 different types i.e.:

**(i)Locally Redundant Storage(LRS) >>**Here the data is going to kept in same region which we have selected. Here data is replicated 3 times and it gives SLA of 99.9 times 9. Here data gets replicated in same region what we have choose.

**(ii)zone redundant storage (ZRS) :** It gives an SLA of 99.12 times 9. It does the replication of data 3 times and this replication could be in same region or different region

**(iii)Geo Redundant Storage (GRS) :** Here the data will get replicated 3 times in primary region and secondary region. It gives an SLA of 99.16 times 9.

**(iv)Read access geo redundant storage (RA-GRS) :** Here the data is replicated 3 times both in primary region and secondary region, but secondary region data is a ready only data.

We never get to know in which regions or zones the data centers are located by MS.

* If we select Blob storage in account kind, then we won’t get the option of ZRS in replication.
* If we select standard in performance, then we won’t get the option of ZRS in replication and will get access tier as Cool or Hot
* Blob is more advanced than Files share/Table/Queue storages.
* Click on Next: Advance >>Secure transfer required : (i)disabled (ii)enabled
* If we select enable then we can’t upload/download the data using HTTP, but with HTTPS we can.
* If there is an HTTP connection means Non-secure and if its HTTPS connection means that is secured.

Storage account(SA) will keep the data of our VM.

**Virtual Network:**

(i)all networks (ii)Selected networks.

If we want to download/upload the data from all the networks then we can choose this option as All networks else if we want to restrict for any network then we can choose as selected networks.

**Data Protection:**

(i)Disable (ii)Enable.

It allows us to recover our blob data to save when blobs and blob snapshots are deleted, if we override the blob data by any chance then by enabling this option it saves our data for specific time like (7 or 9 or 20 or 50 days based on our choice).

**File Share Storage:**

We create a file share and then we can help the users to map the file share with the team.

* We can create a directory or folder in file share that we have created to any machine then it will give us an option to which o/s or m/c we want to connect like(windows, Linux, MacOS) and this file share will map to our m/c.
* The letter we choose for Drive then accordingly for every letter will get a different script to run in PowerShell to map the file share to our M/c or VM.
* To Map the file share to our VM we need a port No 443(this is an SMB port no or CIFS) this port No should be open in our environment by our internet service provider, if this port is open then only we can able to map it .
* In Azure VM this 443 port No is open or not, if its open then we can map our file share to any VM.
* When we are creating a folder/container in blob then we have 3 options for public access level and i.e.: (i)Private (ii)Blob (iii)container.

**(i)Private:**

Only accessible to the owner/subscriber who has created the blob storage or accessible to the users whom the access has been granted by owner.

**(ii)Blob:**

Means any person can read (blobs only, blobs means files only)blob is under the container, blob is just one file and container can have many blobs under it.

**(iii)container:**

Anonymous read access for containers and blobs.

* If we open the link in browser then after we upload the files/docs in Storage account(SA) for BLOB, then the link is somewhat like this.

[https://Practice11.blob.core.windows.net/manish/1st%20March(5).jpg](https://practice11.blob.core.windows.net/manish/1st%20March(5).jpg)

Practice11 >> Storage account name

Blob.core.window.net >> blob storage

Manish >>folder name that we gave in the blob

1st March>>file name that we have uploaded.

* When we are working with files under the SA and when we click on files, here we have to create a +Fileshare and this is nothing but a folder/directory how we have in blob and under this FileShare we can place/upload files and folders what all we want.

<https://practice11.file.core.windows.net/shared11/wasay%20files/azure%20admin%20job%20description>

practice11 >>Storage account name

file.core.windows.net >> file storage

shared11 >>file share name

Naresh files >> folder name

Azure admin job description >> file name that we have uploaded under Naresh files folder.

* While creating a SA if we create account kind as Blob Storage then in performance, we can find only Standard as enabled option and premium will get disappeared.

**The Difference: Azure Blob Storage vs File share Storage vs Disk Storage:**

| **Storage Category** | **Azure Blob Storage** | **Azure File Storage** | **Azure Disk Storage** |
| --- | --- | --- | --- |
| **Object Storage** | Primarily used for storing unstructured data such as text, images, videos, and backups | Not suitable for storing unstructured data; mainly used for sharing files across multiple machines in a distributed environment | Not suitable for storing  unstructured data; mainly  used for hosting VM disks and  persistent storage for Kubernetes |
| **Durability and Availability** | Provides high durability and availability of data with built-in redundancy and replication across multiple regions | Provides durability and availability of data with built-in redundancy across multiple nodes in the same region | Provides durability and  availability of data with  built-in redundancy within the same availability set or availability zone |
| **Performance** | Provides high throughput and scalability for read-intensive workloads | Provides low latency and high throughput for read/write operations on small files | Provides low latency and  high throughput for  read/write operations on  large disks |
| **Cost** | Cost-effective for storing large amounts of data that is not accessed frequently.  Cost depends on the amount of data stored per month, the number and type of operations performed on the data, and the data transfer costs. Blob Storage also offers different tiers of storage for different access frequencies and performance needs:**hot, cool, and archive**. | Cost-effective for sharing files across multiple machines; may be more expensive than Blob storage for large amounts of data,  Cost depends on whether you’re deploying [premium](https://learn.microsoft.com/en-us/azure/storage/files/understanding-billing#provisioned-model)  or [standard](https://learn.microsoft.com/en-us/azure/storage/files/understanding-billing#pay-as-you-go-model) file shares. | More expensive than Blob  and File storage; suitable  for performance-critical  workloads.  Pricing depends on the size  of the disks per month, the  number of transactions  performed on the disks, and  the data transfer costs. Disk  Storage also offers different  types of disks for different  performance and cost  requirements: ultra-disks,  premium SSDs, standard  SSDs, and standard HDDs. |
| **Disaster recovery capability** | Supports geo-redundant storage (GRS) or read-access geo-redundant storage (RA-GRS), which replicates your data to a secondary region that is hundreds of miles away from the primary region. You can also initiate a storage account failover to switch the secondary endpoint to the primary endpoint in case of a regional outage. | Supports GRS or RA-GRS for disaster recovery. However, you cannot initiate a storage account failover for file shares. Instead, you have to use Azure Backup or Azure Site Recovery to protect your file shares from regional disasters. | Uses GRS or RA-GRS to  replicate your disks to a  secondary region. You can  also initiate a storage  account failover for disks.  Additionally, you can use  Azure Backup or Azure  Site Recovery to back up  and restore your disks in  case of a disaster. |
| **Integration** | Provides integration with various Azure services such as Azure Functions, Logic Apps, Azure Stream Analytics, and Azure Data Factory. This makes it easy to build data processing pipelines in the cloud. | Provides integration with various Azure services such as Azure VMs and Azure Kubernetes Service and on-premise applications. Easy to migrate on-premises applications to the cloud and share files across different platforms. | Provides integration with  Azure VMs and Azure  Kubernetes Service.  This makes it easy to  attach high-performance  storage to VMs running on  Azure. |
| **Reliability** | Data stores in Blob Storage are highly durable and reliable with very low chances of data loss.  Blob storage automatically creates multiple replicas of your data, and each replica is stored in a different storage scale unit. | The data stored in File Storage are highly durable and reliable with very low chances of data loss. The service automatically stores multiple replicas of your data in different storage scale units to ensure data durability. | Data stores in Disk Storage  are not durable and reliable  as compared to Blob Storage  and File Storage. Azure Disk  Storage provides durability  based on the type of disk you  choose. For example: Premium SSD disks offer an  SLA of 99.9% and provide  durability by replicating your  data within the same data  centre. Standard SSD disks and  Standard HDD disks offer an  SLA of 99.5% and store three  replicas of your data within  the same region, but across  different fault domains. |

* While creating a SA if we select performance as standard and account kind as Blob storage then after deploying the SA we can only see Blobs under Storage Account means we can only see blobs service that’s it.
* While creating a SA if we select Account kind as General purpose then we can see all kind of storage services(like containers, Files, Table & Queue storage)
* Azure storage has 4 separate storage offerings(i.e: Containers, Files, Table & Queue) in which Blob is one of the storage offering, Blob is the most used by infrastructure specialist, platform eggs and Azure admins.

**Azure Function Apps:**

Azure Functions is a serverless solution that allows you to write less code, maintain less infrastructure, and save on costs. Instead of worrying about deploying and maintaining servers, the cloud infrastructure provides all the up-to-date resources needed to keep your applications running.

Azure functions is a serverless concept of cloud native design that allows a piece of code deployed and execute without any need of server infrastructure, web server, or any configurations. Azure functions can be written in multiple languages such as C#, Java, JavaScript, TypeScript, and Python.

Azure functions are scalable. When demand of execution increases, more resources are allocated automatically to the service and when requests fall, all extra resources and application instances drop off automatically.

**Advantages/Benefits of Azure Functions:**

* Azure functions are lightweight and serverless.
* Azure functions are easier to write and deploy.
* Azure functions are fast to execute because there is no large application, start-up time, initialization, and other events fired before the code is executed.
* Azure functions’ execution is triggered when an event is fired.
* Azure functions are compute-on-demand and that is scalable. When demand of execution increases, more resources are allocated automatically to the service and when requests fall, all extra resources and application instances drop off automatically.
* Azure functions support multiple programming languages including C#, F#, Java, JavaScript, TypeScript, and Python. You choose your choice of language.
* Azure functions do not need any infrastructure and have 0 maintenance.
* Azure function can be build, tested, and deployed in Azure portal using a browser or using any framework.
* Azure functions are easy to upgrade and doesn’t affect other parts of the website.
* Azure functions use industry standard and can communicate with other APIs, databases, and libraries.

**Implementation Steps:**

In Azure portal search for Function App>>Create and fill the details as below

**Subscription:** Free trial/any

**Resource Group:** NareshRG/any

**Function App name:** NareshFunctionApp/any of your choice

**Publish:** code

**Runtime stack:** .Net

**Version:** 6(LTS)

**Note:** don’t user lower version which is not supporting an app development in Azure portal directly.

**Region:** EastUS/any of your choice

**Operating system:** Windows

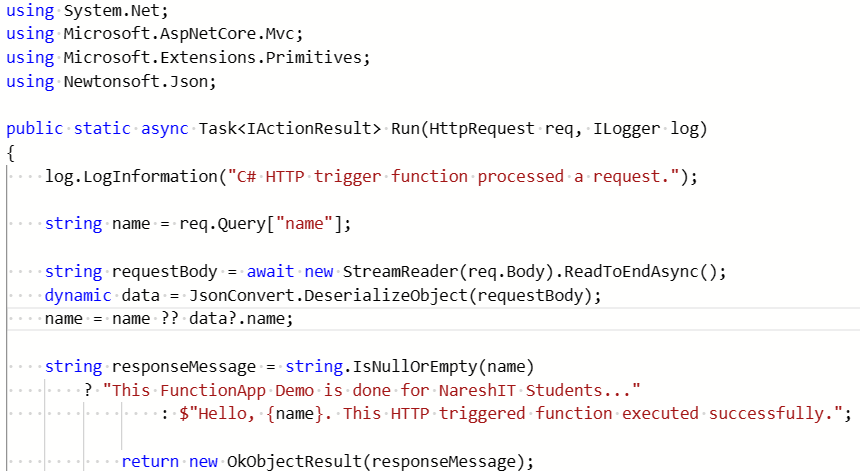
**Plan type:** consumption(serverless) and then leave rest of the values to default as is and deploy the function App.

Click on Function App>>Create in Azure portal(below)>>**Development environment:** Develop in portal>>**Template:** HTTP trigger>>**New Function:** NareshStudentsPracticeFunction>>**Authorization level:** Anonymous>>Create.

Click on Code+Test(left side under NareshStudentsPracticeFunction) and now here will see a code written in C#>>click on Test/Run(top centre)>>Run>>and here at the bottom we can we see logs that the function has been called and we can see the execution message as **“Hello, Azure. This HTTP triggered function executed successfully.”**

If we want to see the same execution message of the function in browser, then click on Get Function URL>>copy the URL from here and paste it in a new browser the same message we can see it in URL as well.

If we want to show the customized message as per our need, then in our C# code we can pass the custom message as shown in below image and after passing the custom message click on **save on top for sure** and again click on Get function URL>>copy the URL and paste it in a new browser and with this we can see the custom message is getting populated in our browser URL with Http



Hence the Azure functions is a serverless concept of cloud native design which allows us a piece of code deployed and execute without any need of server infrastructure, web server, or any configurations. Azure functions can be written in multiple languages such as C#, Java, JavaScript, TypeScript, and Python.

**Virtual Network:**

An **Azure Virtual Network** (VNet) is a representation of your own **network** in the cloud. ... When we create a Virtual Network(Vnet), then all our services and VMs within our Vnet can communicate directly and securely with each other in the cloud.

**Virtual network(Vnet):**

It is basically an extension of our physical network, the resources like network, routing, switching….this all

IP patterns for creation of multiple subnets:

**Address Space for Vnet creation.**

10.0.0.0/16

**Address range for creation of multiple subnets:**

10.0.1.0/24

10.0.2.0/24

10.0.3.0/24

10.0.4.0/24

10.0.5.0/24…………etc

* Configuration is done by us, but it is being managed by our cloud provider and this comes under IAAS settings.
* All the devices in networking are managed by cloud provider but we are free to define our own network(n/w), free to create the address space, free to select the resource group, location, subnet….etc.
* Basically we have 5 classes of IP’s. The 5 classes of IP is the first 5 Alphabet

i.e: (i)Class A (ii)Class B (iii) Class C (iv) Class D (v) Class E.

10.0.0.0 >> Class A (1 octet \* 8 bits = 8 network bits in Class A)

172.16.1.0 >> Class B (2 octet \* 8 bits = 16 network bits in Class B)

192.168.2.8>> Class C(3 octet \* 8 bits = 24 network bits in class C)

* The octet in IP address are separated by dot(.) Ex172.16.14.18

Here 172 is one octet 14 is one octet

16 is one octet 18 is one octet.

* The IP address is having 32 bits and each octet is having 8 bits, so how come the IP address is having 32 means 4\*8 = 32 bits(for IPV4).
* These bits are binaries so they are having 0 & 1
* **We have network bits and host bits in any IP address**
* How many network bits and host bits we have in class A ?? the answer is we have 8 network bits and 24 host bits in Class A.
* How many IP’s we can have in Class C?? i.e: 2^24 -2 i.e: 16,777,214 IPs we can have in class C

**In traditional networking** will do -2 and this **-2 is basically for broadcasting and networking domain**. The first IP & Last IP we do not assign to any machines, so generally we go ahead and subtract it from total

**In cloud networking** will do -5 and this -5 is basically for

(i)Multicast domain

(ii)Broadcasting domain

(iii)Loopback domain

(iv)Link-local

(v)internal DNs

* The **Class D & Class E is useful for scientific and multicast purpose.**
* Generally we refer to Class A, B & C for our common use.
* By definition the **Class A** IP address has 8 network bits and 24 host bits.
* By definition the **Class B** IP address has 16 network bits and 16 host bits.
* By definition the **Class C** IP address has 24 network bits and 8 host bits.
* The Total No of IPs that we can have in Class A. Class B & Class C

|  |  |
| --- | --- |
| Traditional Networking | Cloud **Computing Networking** |
| Class A is 2^8-2 = 254  Class B is 2^16-2 =65,534  Class C is 2^24-2 = 16,777,214 | Class A is 2^8-5 = 251  Class B is 2^16-5 =65,531  Class C is 2^24-5 = 16,777,211 |

When we write an IP address with structure as 10.1.0.0/16(this /16 we call it as mask bits)

* When we are following an IP address with mask bits such as /8, /16, /24 then it is called as Classful inter domain routing.
* The pattern of IP address with **classful inter domain routing** is (i)10.2.8.16/8 (ii) 172.16.12.10/16 (iii) 192.56.23.10/24
* The pattern of IP address with **classless inter domain routing** is (i)10.2.8.16/9 (ii) 172.16.12.10/11 (iii) 192.56.23.10/19….etc.
* **Public IP address range for Class A, B & C as follows:**

**Class A :** 1.0.0.0 to 9.255.255.255 & 11.0.0.0 to 126.255.255.255

**Class B :** 128.0.0.0 to 172.15.255.255 & 173.32.0.0 to 191.255.255.255

**Class C :** 192.0.0.0 to 192.167.255.255 & 192.169.0.0 to 223.255.255.255

* The range for Class A IP address starts actually from 0 to 127, but we do -2 and this -2 is common for all classes IP address and so when we do -2 means removing the first IP & last IP. The first IP is 0.0.0.0 and the last IP is 127.255.255.255, so this 2 ranges(0.0.0.0 & 127.255.255.255) we do not assign to anything & that’s why we say 1 to 126 is the IP address range for class A.
* If asked which IP address pattern you have followed in your project ?? We have followed Classless inter domain routing (CIDR) IP patterns, b’coz with this pattern we can increase and decrease the IP address as per our requirements and convenience.
* If the IP is starting from 1.0.0.0 to 126.255.255.255 then we say it is a **Class A IP address**.
* If the IP is starting from 128.0.0.0 to 191.255.255.255 then we say it is a **Class B IP address.**
* If the IP is starting from 192.0.0.0 to 223.255.255.255 then we say it is a **Class C IP address.**

**Private IP address ranges for Class A,B, & C:**

**Class A:** 10.0.0.0 to 10.255.255.255

**Class B:** 172.16.0.0 to 172.31.255.255

**Class C :**192.168.0.0 to 192.168.255.255

* These all IP addresses are defined by IANA.
* For classful IP addressing it is suppose to have post prefix as /8, /16 & /24
* For classless IP addressing which is called as classless interdomain routing(CIDR) and the post prefix could be anything except 8,16 & 24.xxxz
* If we say 10.1.0.0/16(Means it is class B IP address, b’coz we always take network bits in (/) consideration and this IP address range will give us 65,534 how means 10.1.0.0 = 2^16-2 = 65,534).
* The pattern or format of IP address which do not follow any class that IP address is called as **Classless Inter domain routing(CIDR)**
* We also have many subnet calculators(search in google)
* When we are doing a classless interdomain routing and when we are putting IP in CIDR format then we have to ensure that all IPs has been utilized correctly.
* Though we are using an IP address of Class A, Class B or Class C but free to borrow the network bits, free to increase the n/w bits, free to decrease the n/w bits.
* The pattern or format of IP address which is not following any class then that is called as classless or CIDR IPs where we can able to modify and change the IP addresses as per the requirement.
* When we click to create a Vnet in Azure portal then we are putting multiple resources in the Vnet and we fill the below template for Vnet.i.e: Name/AddressSpace/Subscription/Resource Group/Location/Address range for subnet/DDOS protection/Service end point….etc.
* In address space if we mention IP as 10.1.0.0/16 then below it shows how many IP address do we get(ex: 65536) it means in this address space we can create these many resources which we can assign a unique IP address from this range.
* When we create a subnet then those subnet can divide these IP addresses which we have created in Address space then we can create a multiple subnets as per our choice….when we are creating a first subnet we should not assign all the IP address creating a first subnet, we should not assign all the IP address to first subnet only. Probably we can give around 254 IP’s to first subnet, then 320 IP’s to 2nd subnet…like wise we can spread the IP’s to multiple subnets and this process of dividing the IP address is called as proper or correct networking planning.
* The basic thing to put in mind is when creating a Vnet then how many IP’s do we actually need and that depends on how many resources we need….so all the resources which we are creating are really need or some IP and if we are creating a VM and that will need a private IP.
* Subnet(left side under Vnet): In subnet we can see address range/Available addresses(IPs)/security group/subnet name….etc.
* After defining the IP addresses in Vnet, if we want to add more subnet or further then go to Vnet>>subnet(left side, under Vnet) we can add subnet is here to whichever Vnet we want .
* We can create as many subnets we want provided it should not exceed the No of total IPs we have in our address space.
* The public IP gets from Azure subscription by default IP list, so where ever we are putting the resource based upon the location/region of the resource then then IP address will get assign to the resource and same goes for Vnet, the public IP we don’t need to worry about but for private IP we need to take care of.
* The IP’s which we are assigning are private and after our Vnet gets created we are free to increase it and it is scalable and we can add another address space later on to allocate more resources and it is scalable.
* The address space feature is for us to understand the networking & in address space the IP’s we are defining only that much IP’s we can allocate to the resources.
* We have to be very smart and cautious b’coz each IP is going to cost us.
* In Azure 5 IPs are reserved(that’s why we are doing -5) & this 5 reserved IP’s also include the other 2 IP’s i.e: Broadcasting & networking.
* If we want to create a subnet from the address space then we have to define (or) Pass the values for subnet, and while creating a Vnet then it is must to pass the values for subnet and address space.
* We can assign al the IP’s to only one subnet or else we can assign the address space IP’s to multiple subnets(its our choice) few we can define to subnet and rest we can keep for later use or for additional subnet, we can also create additional subnet.

**Network Security Group(NSG)/Checking the ports for Azure Resources:**

* NSG is basically designed to filter traffic to (inbound) and from(outbound) Azure resources located in Azure Virtual Network, here we create the in-bound and out-bound traffic rules for our VM’s or Azure resources which are hosted on our Virtual network’s
* Here in NSG the filtering is controlled by rules and in NSG we have an option to have multiple inbound & outbound rules.
* The rules are implemented in NSG by passing multiple parameters as follows

(i)Source/Destination (IP addresses, service tags, application security groups)

(ii)Protocol (TCP/UDP/any)

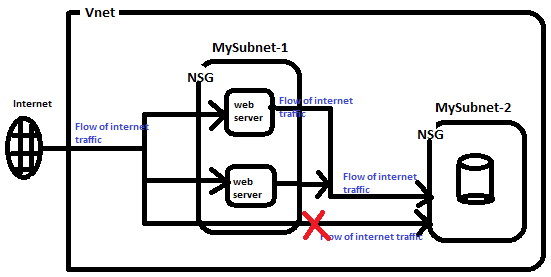
(iii)Port (Port ranges: RDP:3389; SSH:22; HTTP:80; HTTPS:443)

(iv)Direction: inbound or outbound

(v)Action: Allow or Deny.

(vi)…. Etc

**Diagrammatical representation of Network Security Group (NSG):**



* Let’s say we have 2 servers, to handle web application traffic and 1 more server to host our DB and depending on our architectural decisions we cand divide those into subnets, Example: one subnet for our web application traffic and another subnet to handle our DB/DB server and since all these are subnets they need to reside within a Vnet and if we create infrastructure like this and dont do anything else then all the traffic coming from the internet will be allowed to all of these servers additionally all the traffic between these servers will be allowed, so here everything can communicate with everything. But we want our internet traffic should not be reaching our DB/DB server and not all of these services should be able to communicate with each other and now to fulfil these kind of requirement’s NSG comes into picture.
* After implementing the rule as part of NSG we can see exclamation mark before our rule name, this is based on Microsoft recommendation and in general the security recommendations we should not open our RDP
* connectivity to the entire world, we should target always specific servers and specific IP ranges, Yah but for testing this is fine.
* Networking (left side, under VM) when we click on networking, we can see all the details of a VM specially related to network resources like (i)Network interface: vm249(this is NIC card) (ii)Public IP (iii)Private IP (iv)Accelerated networking (v)Inbound port rules (vi)outbound port rules (vii)application security groups (viii)Load balancing.
* In NSG we have an option to add inbound port rules and out bound port rules, application security groups, load balancing…etc.
* **In-bound Port rule:** Here we are allowing the incoming traffic/inbound port rule to our VM Ex: which ports/services to allow which ports/services not to allow or deny…etc.
* **Out-bound Port rule:** If the VM is trying to access an internet then basically it means it is trying to make an outbound rule/request, whether we want to allow or deny the outbound request by going through a particular port No, we have this liberty to modify this setting.
* In networking (left side, under VM) we have priority like (1000,1001,65000,65001….)the way this priority works is lets say we have a priority to allow an RDP connection as 1000, if we want to disable this priority(means block/deny and connect to our VM using RDP) then click on add inbound port rule>>then click on Basic(on very top)>>service : RDP>>priority : 100>>protocol : any>>Action : Deny then click on add.
* The lower the priority No the higher the impact, the higher the sequences.

**(a).** When we take an RDP from my laptop to the VM then basically we are making an inbound request to the VM…. for the VM it is an incoming request.

**(b).** After making the setting from above steps and clicks on Add then if we try to take the RDP, by giving the public IP None of us would get connect to this machine…this exactly called as network security group.

**(c)** If the priority is 100 then that is bigger than the priority 101,1001,201…. whatever the No is, if it is greater than 100 then the priority is less (No the smallest, priority is the highest).

* Hence in NSG we can do the setting for allowing (or) denying the ports of RDP/HTTP/HTTPS/FTP/SMTP/MSSQL/POSTGRESQL/LDAP…many as such.
* We can find all these options in services. By implementing NSG it is an another way of securing (or) making sure that our connections and VM’s are secure, by creating an in-bound and out-bound port rule
* When the priority No is big then the impact is less….it is like a severity as P1, P2 has huge impact compare to P3 & P4.
* We can also delete this in-bound rules that we have created for our RDP, by simply clicking on delete button,
* HTTP & HTTPS are the protocols that we use to connect to the internet, http is not secure https is secure, we can even set the http & https request to allow or deny these protocols in our in-bound and out-bound requests.
* Implementing this in-bound port rule and out-bound port rule for VMs is called network security group (NSG).
* If we do not want our M/c to connect to http, we want to deny it, that means we only want to connect to the secure sites (i.e.: https).
* After implementing the http as deny, if we go ahead and connect our machine then our http internet will not work, we implement this http or https in out-bound port rules and in-bound port rules as well.
* So basically, by using the NSG we can manage and create some port rules and with which we restrict the usage of communication coming in or going from our VMs.
* Once we are creating a resource (like VM) then it is by default picking a public & private IPs based upon the region of resource.

**Layer 4 v/s Layer 7 Load Balancers:** Basically, we are having 2 types of Load Balancers to do the Load Balancing of network traffic and that is

(i)Azure Load Balancer(VM’s for IAAS Services) &

(ii)Traffic Manager Profile(Web Apps for PAAS Services)

**(i)Azure Load Balancer (LB):**

Azure Load Balancer is an Azure component/service which delivers a high availability and network performance to our applications, it distributes the network traffic to the backend resources using Load Balancing rules and Health probes, Load Balancing rules determines how the traffic is to be distributed to the backend resources and Health probe indicates the resources in the backends are healthy up and running and accessible through the load balancer.

**Key points on Azure LB:**

* LB can be used for inbound as well as outbound scenarios, it can scale up to millions of TCP/UDP applications flows
* Distributes inbound traffic to backend resources using load balancing rules and health probes
* Even protect on-premises web applications with secure remote access
* Basically there are **2 types of Load balancer’s** that we can deploy i.e.:

**(i) Public/External LB (ii)Private/Internal Load Balancer**

**Public/External Load Balancer:** When there is an incoming internet traffic coming to our VM’s/Workloads then we can use public/external LB to load balance this traffic (or) it maps the public IP address and port No of incoming traffic to the VM's private IP addresses

**Internal Load Balancer:**

To route the traffic with in our Vnet to different VM’s (or) work loads then we use internal/private load balancer.

**(i)**It directs the traffic only to the resources inside a virtual Network or the resources which are accessible through a VPN Azure infrastructure.

**(ii)**The frontend IP addresses and virtual networks are never directly exposed to an internet endpoint

**(iii)**This enables the Load Balancing with in a virtual network for cross premises virtual networks for multi-tier applications.

**(iv)**When we are planning to implement a LB then all the VM’s should be in same region, same Vnet and must be in an Availability Set, if the VM’s are not in an Availability Set then we cannot add the VM’s inside the LB.

**(v)**When we are planning to add the VM’s in a LB then the VM’s should not be carrying the Public IP addresses.

**(vi)**We always add VM’s in a LB, and when we deploy a Standard LB then that LB can manage up to 1000 VM’s.

**Diagrammatical representation of Load Balancers:**

Diagram

Description automatically generated

**Steps to Implement Load Balancer:**

**Step1:** Create a Resource Group.

**Step2:** Deploy 2 VM's with name as Webserver1 & Webserver2 and ensure this VM's should deployed under the same Vnet, an Availability Set & Same region..

**Step3:** Install IIS web server in both the VM's.

**Step4:** Perform the below steps in both the VM’s

Create a web page to perform the testing/trouble shooting of LB whether it is balancing the load equally on both the Servers/VM's or not and for this

Go to C drive on both the servers>>inetpub>>wwwroot>>Delete both the pages(iisstart) which we got created by default>>create a new text file and mention the content in the file as Webserver1(1st VM)>>then do Save as and pass the file name as “index.html”(including the double quotes while saving the files for sure) & save type as All Files

Go to C drive on both the servers>>inetpub>>wwwroot>>Delete both the pages(iisstart) which we got by default>>create a new text file and mention the content in the file as Webserver2(2nd VM)>>then do Save as and pass the file name as “index.html”(including the double quotes while saving the files for sure) & save type as All Files

**Step5:** Disconnect the VM's RDP’s and Disassociate the Public IP’s for both the VM’s.

**Step6:** Provision a new Load Balancer of type Public

**Step7:** Add this 2 VM's in Backend pool and to add the VM's in backend Pool the VM's should not contain an IP address (for this will stop the VM's, then go to networking and click on Public IP and say Disassociate)

**Step8:** Perform the same step 7 for the 2nd VM also to add the VM's in Backend Pool.

**Step9:** Now go to Backend Pool feature of the Load Balancer to add both the VM’s.

**Step10:**Now go to Health Probe feature of the Load Balancer to add the HealtProbe

**Step11:** Now go to Load Balancing Rule and pass the values accordingly

(i)Name: Pass any name

(ii)IP Version: IPv4

(iii)Frontend IP address : Here select the LoadBalancerFrontEnd

(iv)Protocol: TCP

(v)Port: 80

(vi)Backend port : 80

(vii)Backend pool : Select the Backend pool what we have crated for our LB

(viii)Healthprobe: Select the Healthprobe what we have crated for our LB

(ix)Session persistence : None

(x)idle timeout : 4

(xi)Rest all default values and finally click on Add

**Step12:** Copy the Load Balancer IP from Overview, paste it in browser and keep refreshing to check how the LB is balancing the load on different VM's that we have added in the Backend Pool

**Key points on Azure Load Balancer:**

* We can use Azure LB to load balance the incoming traffic to our VM’s and this configuration is known as public LB.
* When there is an incoming internet traffic to our VM’s then we can use public/External LB, this LB is facing the traffic from public internet.
* To route the traffic with in our Vnet to different VM’s (or) workloads then we use Internal/Private LB.
* We cannot do the Load Balancing in different VM’s until we put them in one Availability Set, all the VM’s should be in same availability set if we want them to shared the load equally via load balancer
* We cannot change the SKU of an existing Load Balancer nor we can add an on-prem VM’s to Azure load balancer.
* A stand Alone VM, AS or VMSS can refer to only one SKU (either Basic or Standard)but never both.
* When we create a LB independently there, we can add the VM’s or VMSS to the LB, but when we are creating a LB as part of VMSS or VM then that LB will be dedicated or responsible to manage those workloads or VM’s only.
* All the VM’s should be hosted in common Vnet, and Availability set to add those VM’s to a Load Balancer.
* We can add VM’s to a Load Balancer even after creation of VM’s, but Availability Set, Availability Zone is only possible while creating a Virtual Machine (VM)
* Load Balancer will charge us the No of rules we implement for the Load Balancer.
* We can add the VM’s to a Load Balancer only if the VMs are not carrying a public IP, hosted on the same Vnet and Availability set.
* If the VM’s region and LB region is same, then we select tier as Regional and if the VM’s and LB regions are different when we select tier as global while provisioning the LB
* While deploying the VM’s for LB ensure that VM should be **deployed under an Availability Set.**

**Properties of LB:**

(i)Frontend IP Configuration, (ii)Backend pool (iii)Health probe(ii) (iii)Load balancing rule…. etc.

* Load Balancer load balance the traffic @ layer 4 of OSI model
* Application Gateway load balance the traffic @ layer 7 of OSI model
* Application gateways do the load balance @ layer 7 of OSI model which is the application layer of OSI model. AG only works with HTTP & HTTPS request.
* AG & LB, these 2 can only do the load balance for the resource which are part of the same region, but traffic manager can do the load balancing from one region to another region and another region resources as well.
* Traffic manager works on DNS ways and AG is layer 7 load balancer which works with web traffic only like HTTP/HTTPS/web socket…. etc.
* AG comes with a feature called URL based routing.
* LB again comes with 2 different offerings ie (i) Basic & (ii)Standard. When we are creating a basic LB then public IP address also comes with 2 offering i.e.: Basic & Standard.
* When we are deploying a LB, will get an option to choose the SKU as Standard or Basic Load balancer.
* Among multiple High availability options in Azur, Load Balancer is also one of the High availability options we have which does the Load Balancing for our workloads which are added in the backend pool.
* Load Balancer works on layer 4(transport layer) of Open system interconnection (OSI) model.

|  |  |  |
| --- | --- | --- |
| **Features** | **Standard Load Balancer** | **Basic Load Balancer** |
| Backend pool Size | Supports up to 1000 instances | Supports up to 100 instances |
| Health probe | TCP, HTTP, HTTPS | TCP, HTTP |
| Backend pool end points | Any VM or VMSS hosted in a single Vnet and Availability Set | VM’s in a single availability set or VMSS |
| SLA | 99.99% | Not documented as such |
| Secure by default | Yes | No |
| Availability Zone | Zone redundant, cross zone load balancing | No supported |
| Pricing | Charged on based on the No of rules for inbound & outbound | Free of cost |

* When we are balancing the network to different VM with the help of LB then all the VM’s should be placed in common Availability Set, then only the work load/network traffic can be equally distributed to all the VM’s from the LB
* The Idle Timeout for Azure LB is 4 mins, means if any of our request for the first time reaches to VM1 and if we refresh the IP in our browser within 4 mins then the request won’t go to another server, it will be there for the same VM until 4 mins, and if you have not done any activity for more than 4 mins and refresh your IP then in that case it might go to the different VM, this is the basic condition that we have in Azure is that we have to be on one server for at least 4 mins
* When we are creating a Basic LB then it will create a Basic public IP address and if we are using a standard LB then it will create a standard public IP address.
* Traffic manager works on DNS-to-DNS mapping, and it balance the load from one region to another region as well.
* In VMSS basically we are creating a No of identical VMs which are identical in O/s, carrying the same configuration and same application data. In VMSS we are doing autoscaling and defining No of VM’s which we are going to deploy.
* When we want to see the inbound port rule/outbound port rule /application security groups/load balancing for a VM then go to VM and click on networking.
* In an inbound port rule/outbound port rule we can see the details like priority/name/port/protocol/source/destination/action…. etc.
* We can choose Std LB for creating 1000 instances, greater backend pool flexibility and High availability ports.
* While creating the LB, if we select SKU as Basic then it will create a basic Public IP. If we select SKU as standard, then it will create a standard public IP and here in standard LB it is always a static public IP and if we create a dynamic public IP then we have to select SKU as basic.
* Public LB is going to manage the traffic which is coming from public VM’s. Now here in backend pool we have to add the VM’s and private n/w IP to LB which we have to create and can be associated to 4 things i.e.: (i)Unassociated(default) (ii)Availability Set (iii)Single VM (iv)VMSS.
* If we want to associate a VM and network IP for a LB then we always do this activity in back-end pool and we can find this backend pool option in LB.
* It may be possible some time the VM is not working/Decommissioned/Deallocated or some technical issue then the VM is not able to address any request or users request from LB…then in such scenarios the LB needs to be identify whether the VM is up and running. So to get the heartbeat of the VM we must use the feature Health probe.
* This Health probe helps the LB to know whether the VM is capable enough to answer & acknowledge our request. In Health probe (HP) we check the heartbeat of our VM.
* In HP we fill the below details Name/Protocol/Port No/Interval/Unhealthy threshold…. etc.
* If we set the interval as 5 or 10 secs then it means for every 5 secs the LB probe with the back-end pool VM’s…means LB is checking whether it is getting healthy heartbeat of the VM’s for every 5 secs.
* LB only supports VM’s/VMSS/Availability sets
* If we are setting the unhealthy threshold as 2 means if 2 consecutive times the LB has received the unhealthy heartbeat of the VM and then it will consider that VM is unhealthy.

**Load Balancing Rule(left side under LB):**

It show us how the Load balancing is working in Load balancing rule we fill the below details… like Name/IP version/Frontend IP address/Protocol TCP/port/Backend port/Health probe…

* **Load Balancer will charge us the No of rules we implement in the LB. The public IP address of LB is different and the public IP of VM is different.**
* If we go to the LB which got created as part of VMSS then after going under that LB we can see all When we are creating a VMSS then along with it VM’s/LB/AG/Inbound NAT rules/Health probe/Backend pools….etc. also gets created as part of VMSS….we can skip few as per our requirements.
* details like Health Probe, Load balancing rules, Inbound NAT rules, Backend pools….etc.
* **When we create a LB with VMSS then that LB is specific only for that VMSS.**
* If we have an existing VMSS and we selected none for LB option then we can create a manual LB then go to Backend pool and can use that VMSS to be added in the backend pool and then we can create the health probe, load balancing rules…..etc.
* **Backend pool(left side, under LB)** : it is the place where we can find the list of the VM’s which are going to be managed by LB. so here in backend pool we can add the VM’s to LB.
* We can see the Private IP’s of VMSS instances in backend pool. LB is the part of VMSS that means this LB is supposed to manage the traffic going to the VMSS which we have created and it will list all the virtual machines which we have created in the VMSS.
* If we want to see the VM’s names and how many VM’s we have created while creating the VMSS then go to VMSS>> Instances(left side)
* While creating the VMSS, if we are giving name as “Vmscaleset1” then a LB also gets created with this name and VM’s gets created with name as VMSS1\_0.
* To know the IP addresses(public ip) of VM’s that we have created during VMSS then go to LB>>Inbound NAT rules(left side under LB) then here we can see all the VM’s IP addresses that we created while creating VMSS.
* We can see the complete details of the VM that we have created while creating the VMSSS with below steps…..go to VMSS>>Instances>>click on the instance>> then we can see the complete details of that VM like…. Instance ID/Status: Running (or) Stopped/Public IP/Private IP/Vnet/Subnet/Disk/Computer name/Start/Restart/Delete/Upgrade/De-allocate….etc.
* To see the private IP of the VM’s which got created while creating VMSS we can see the private IP’s in load balancer>>Backend pools.
* We can create a VMSS even without LB and application gateway by choosing Load balancing options as (i)Application gateway (ii)Load balancer (iii)None.
* Azure gives us 3 different capabilities to load balance our applications (or) Resources (or) VM’s workloads and those options are.

(i)Load Balancer (ii) Application Gateway (iii)Traffic manager.

We can use the above 3 collectively to achieve our goals and these 3 has different purpose or benefits.

**(ii)Traffic Manager Profile (TMP):**

Traffic Manager Profile (TMP) routes the traffic to different endpoints/web Apps as per our instructions. basically, here we are automating the network traffic that how it should flow.

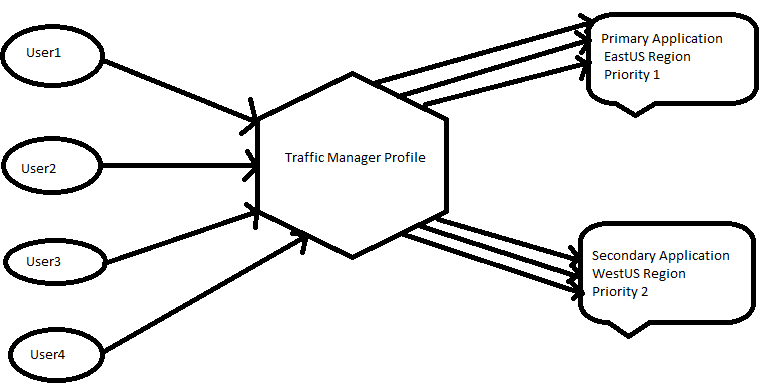
**Example1**: let’s say we have a web application which is hosted on EastUS region, now this application could be on VM or an Azure web App (PAAS),

If this application is critical to business in which the client cannot afford a downtime of even 2 mins, what happens if by chance the infrastructure in EastUS region goes down, then how can we make our application to be available then here we could have a backup available in another region, let’s say West US region and can have this azure traffic manager service available which could be used to route the traffic to another Web App in another region, if in case the primary app goes down for any of the reasons.

**Example2:** This is mostly useful when we are deploying a new web application, what we can do is we can direct 20% of the traffic to new application and 80% traffic to the old application and if new application is working and taking the load as desired then we can slowly increase the percentage of load towards to the new application and ensures that all the traffic is directed towards new application only.

We use traffic manager profile to improve the performance of our web site/web applications. Basically, TMF will redirect our network traffic to another web app which we have stated in endpoint-based priority (like priority 1, priority 2, priority 3…)

**Diagrammatical representation of TMP:**



We have different routing methods for TMP. i.e.:

**(i)Priority:** Route traffic to another endpoint in case the primary fails

**(ii)Weighted:** Route traffic to different endpoints based on weight (like load balancing)

**(iii)Performance:** If we want our end users to use the closest endpoint in terms of the lowest network latency then we can opt this option.

**(iv)Geographic:** Geographic location their DNS query originates from.

**Implementation Steps of Traffic Manager Profile(TMP):**

**Step1:** Create 2 VM's in separate regions

**Step2:** Install IIS in both the VM's and mention as Website1 & Website2 and do the below configurations in both the VM’s.

Go to C drive on both the servers>>inetpub>>wwwroot>>Delete both the pages(iisstart) which we got created by default>>create a new text file and mention the content in the file as Webserver1(1st VM)>>then do Save as and pass the file name as “index.html”(including the double quotes while saving the files for sure) & save type as All Files

**Step3:** Configure DNS for both the VM's(ex: nareshsite; b’coz TMP routes the traffic based on Domain name system(DNS))

**Step4:** Deploy Traffic Manager Profile(TMP)

**Step5:** Goto endpoint inside the TMP and do the configuration as follows.

**(i)Adding 1st Endpoint in TMP:**

Click on +Add>>Type: Azure endpoint>>Name: P1>>check the box Enable Endpoint>>Target resource type: Public IP address>>Public IP address: here choose 1st VM ip (that we have deployed)>>leave rest of the values to default and click on Add to enable the Endpoint.

**(ii)Adding 2nd Endpoint in TMP:**

Click on +Add>>Type: Azure endpoint>>Name:P2>>check the box Enable Endpoint>>Target resource type: Public IP address>>Public IP address: here choose 2nd VM ip (that we have deployed)>>leave rest of the values to default and click on Add to enable the Endpoint

**Testing of TMP:**

**Step6:** Now click on overview of TMP(left side)>>copy the DNS Name url and paste it in the URL, here it will redirect us to either VM1 website or VM2 website

**Step7:** Come to TMP>>Endpoints>>click on anyone of the endpoint>>and change the status Enabled to Disabled>>click Save on top

**Step8:** Now copy the DNS Name URL again and open a new incognito and paste the DNS URL and now here the TMP will redirect the network traffic to 2nd VM website.

Hence it is proved that if one website is down then the TMP redirects the network traffic to another website silently without any knowledge of the end users and which ultimately gives the business continuity.

**SQL DB as service in Azure:**

Basically, Azure gives us two options to run SQL Server workloads.

**IAAS** : Azure SQL Database(DBAAS)

**PAAS** : SQL Server on Azure VM’s i.e. SQL Server inside fully managed VM.

**Azure SQL Database:**

Azure SQL DB is a cloud based relational database service that is build on SQL Server Technologies, it supports T-SQL commands, tables, indexes, views, primary keys, store procedures, triggers, roles, functions…etc.

SQL Database delivers predictable performance, scalability with no downtime, business continuity and data protection with almost zero administration, with which we can focus rapid app development and accelerating our time to market rather than managing virtual machines and infrastructure as it is based on SQL server engine, SQL DB supports existing SQL server tools, libraries and API’s which makes it easier for us to move and extend to cloud.

In Azure SQL DB’s are available in two purchasing models DTU & vCore. SQL Databases is available in

**(i)Basic,**

**(ii)Standard/General Purpose,**

**(iii)Premium(Business critical & Hyperscale service tiers)** each service tier offers different level of performance and capabilities to support lightweight to heavyweight database workloads, we can build our first app on a small database for few months and then we can change the service tier manually or programmatically at anytime based upon our convenience without any downtime to our apps and customers.

**Benefits of SQL DB as Service:**

* High availability: For each SQL DB created on Azure, there are three replicas of that Database.
* On Demand: One can quickly provision the DB when needed with a few mouse clicks.
* Reduced Management Overhead: It allows you to extent your business applications into the cloud by building on core SQL server functionality

**SQL Database Deployment options:**

* **(i)Single Database:** it is an isolated single DB, it has its own guaranteed compute, memory, and storage.
* **(ii)Elastic pool:** Collection of single DB’s with fixed set of resources such CPU. Memory shared by all DB’s in pool.
* **(iii)Managed Instances:** Set of Databases which can be used together.

**Azure SQL Database Purchasing Model:**

There are two purchasing models or service tier **DTU & vCore**

1. **Database Transaction Unit(DTU):**

DTU stands for Database Transaction Unit and it is a combined measure of compute, storage, & IO resources. This DTU based model is not supported for managed instance.

1. **vCores:**

vCores are virtual core provides higher compute, memory, and storage limits and gives us the great control over the compute and storage resources that we create and pay for.

**Implementation Steps:**

Search for SQL server in Azure portal>>Create and deploy the SQL server in Azure.

If we click on SQL databases(left side) then will see no DB’s available inside this SQL server, now will provision a new DB in Azure portal..

Search for SQL DB in azure portal>>create>>fill the details accordingly and create a new DB in azure portal.

Goto SQL servers(that we have created)>>Backups(left side)>>and here will see the DB backup is already available for us in azure portal which is been taken by Azure and from here we can Restore this DB if required based upon the need.

On the extreme right it is giving us an option to restore the DB backup

In Retention policies tab we can see that the Long term retention (LTR) that we can set for weeks, months and years, select the DB and click on configure policies(if we want to change the configure policies) and here we can change

1)Point-in-time-restore to max of 35 days.

2)Take a differential backup every 12 hrs or 24 hrs.

3)Weekly LTR Backups

4)Monthly LTR Backups

5)Yearly LTR Backups

After making the changes and finally click on Apply and then yes.

After connecting to Cloud DB in our Laptop we can create and insert the data into table of cloud DB to check, the sample queries attached below

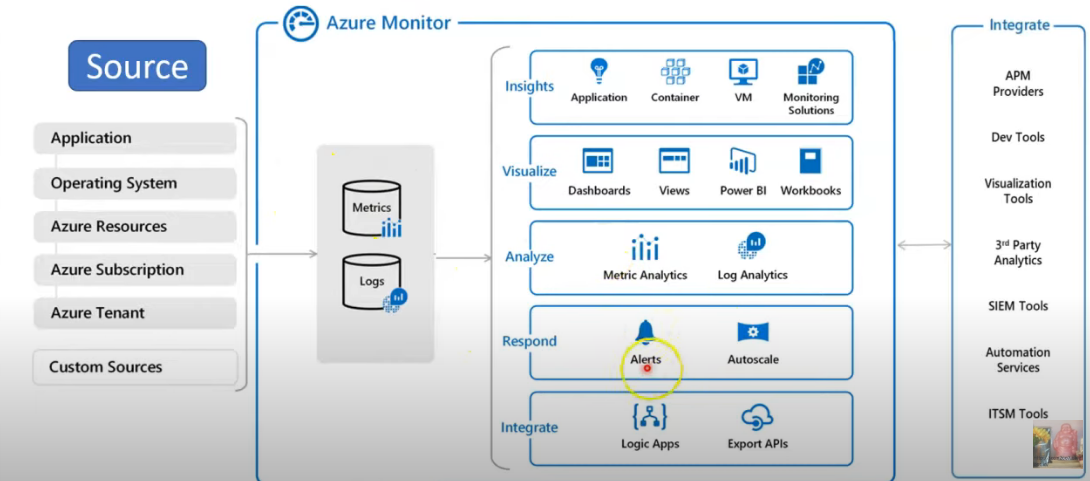
**Azure Monitor:**

Its basically an inbuild services in Azure that collects and analyse data about various azure resources(VM’s, SA, Vnet, App insights…etc) and infrastructure on which these resources run.it gives us insights(detail information) on what is happening application and services that runs on various azure resources.

Azure monitor is a tool which unifies everything into one with which we can monitor the applications, Operating system on which this application runs, we can monitor subscription of Azure resources…etc. we can have all this @ one place and finally all this data can be visualized.

**How Azure Monitor works:**

Azure monitor collects data for azure resources and for infrastructure on which these resources are hosted, Azure monitor collects metrics and logs all the data that azure monitor collects can be divided into either metrics or logs and from these metrics and logs insights are provided, visualizing the data can be happened, analysis can be done and with this we can take the response and integration.



**Logs:**

In Azure monitor logs Events, Performance, Data traces…like this thing related to specific resources/applications can be stored in Azure monitor logs.

Azure has three alert types, metric, log, and activity log.

**Alerts:**

Alerts help us detect and address issues before users notice them by proactively notifying us when Azure Monitor data indicates that there may be a problem with our infrastructure or application. we can alert on any metric or log data source in the Azure Monitor data platform.

**Metrics:**

Metrics are numerical data that define properties like CPU usage, memory usage, network usage…. Etc, this data shows what was happening @ certain point of time and it is stored in intervals, and with this we can plot graphs, charts, tables from these data and can gain information about what was happening at certain point of time.

**Azure Active Directory(AAD)/Microsoft Entra ID:**

**What is the use AD(or) why we use AD:**  With AD we can divide different businesses and we can create a domain depending on their usage, we can also put the computer/users/groups/security groups/distribution list….etc. in these domains as these all we can perform with the help of AD. The purpose of AD is to authenticate and authorize the users to give some permissions & to provide access to certain part of our resources.

* The different license version of Azure Active Directory can be

<https://azure.microsoft.com/en-in/pricing/details/active-directory/>

* When we are creating a user in AAD then for every user will get **.onmicrosoft.com**(ex:khiddubabygmail.onmicrosoft.com)
* **.onmicrosoft.com :** this is the domain name of AD whenever we create users/groups will get this.

**Access control(IAM):** it is the place where we will provide role assignments to different users for different Azure resources or Resource Groups or even to different subscriptions.

* We have different versions/licenses of AAD which ever suits for our project requirements we can opt it.

(i)Azure AD Free >> 500000 object limit

(ii)Premium P1>>No Object Limit

(iii) Premium P2>>No Object Limit

(iv)Office 365 Apps>> No Object Limit.

**Creating Groups in AD:** Click on Azure AD>>Groups>>New Group>>Group Type: office 365>>Group Name :ERR>>Group Description : Err users>> Membership type :Assigned >>click on members(right side we find members) same do for owners as well and click on create.

**Purpose for creating groups in Active Directory(AD)** Rather than assigning permissions to thousands of users we create a group and add all those users in that group and will assign the permissions to that group, which makes the job easy to do the management for all the permissions.

**Adding members to the groups:**  Goto AD>>Groups>>Click on Group(ex ERR)>>members>>Add members>>will get members automatically>>select the members which ever we want>>click on Select

**Role Based Access Control(RBAC)**  With RBAC we can segregate duties within our team and grant only the amount of access to users that they need to perform their jobs,…

**Example** : (i)Allow one user to manage virtual machines in subscription and another user to manage the virtual networks. (ii)Allowing DBA group of users to manage SQL DB’s in a subscription. (iii)Allowing users to manage all resources in resource groups such as VM’s, Subnet’s…etc.

The different roles that we have in Azure for subscriptions are

Owner/Contributor/Reader/Avere Contributor/Avere Operator/…..etc.

**Steps for creating a User in AAD:** go to AAD>>user>>New user>>Name :Rajesh>>User name:Rajesh@mehertechgmail.onmicrosoft.com. so by this steps the above user got added to AAD

**Implementing Identity access management(IAM):**

* Now got to VM >>click on VM>>Access control (AM)>>Role assignment>>Add>>Add Role Assignment>select role>>select user(Rajesh)>Save.
* With above steps the user Rajesh has been given a permission to have contributor access to the above VM.
* After creating a user in AD & assigning role(ex :contributor) to any of the resource and login-in to Azure portal with that user(**Ex:** [Rajesh@mehertech.onmicrosoft.com](mailto:Rajesh@mehertech.onmicrosoft.com)) then after giving the owner access @ subscription level then we can directly create a resource with that login-in.
* If we create a user in AD & not assigned/gave access to any of the resources & if we login-in again with that credentials then it will ask to get a free account. Hence the catch point here is after creating a user in AD. We have to assign that user to any of the resource for sure.
* A contributor(role) cannot go ahead and assign access to other users, he cannot share the azure resources to another but the owner(role) can do.
* When we create a user in AD, it basically asks 3 things i.e..: User/Global Administrator/Limited Administrator.
* We have access control even for Subscriptions/Directories/resource/users…etc.
* Whenever we subscribe Azure active directory or use the subscription it does not matter what is our license (or) what is the subscription (or) entitlement we have, but once we subscribe to AAD we use an email address & password.
* The email address which we use to login for the first time that becomes our global administrator by default to manage the subscription related queries (or) to assign permissions (or) to create a user’s.
* When we create an azure subscription (AS) for the first time the user which we use that becomes our global administrator.
* When we join a company, we get the user id & password and this user id gets authenticated with windows active directory which 95% of the companies do, they will give us AD user account and password.
* Hence AD is pretty much about authentication and authorization of users to get into the system and provide access to the resources which are part of a network.
* When we go into Azure Active Directory(AAD) then in overview we can see on left side our role.
* If we could not able to create a resource then might be the user account does not have a subscription…so the account which we have created in that all the users are supposed to be given a subscription access…to give this access, search subscription in Azure market place>> go to subscription>> select subscription(center) Access control(IAM, left side)>>then we can add the users here so that they can also use the same subscription.
* Click on Role Assignment>>Add>>Add role assignment>>select.
* Now after performing the above steps which ever users we have selected and gave subscription access, if they login to azure portal, now they can able to create all the resources which ever they want.
* **So, the thing we have to keep in mind is , if we create a user in AAD, then it doesn’t mean that by default the user will get access to create resources in Azure portal, they will get access to view the resource and can login to the VM but cant create the resources, Even a global administrator user can’t create the resources, he should be given a subscription access, until we give subscription access, he also cannot create a resources.**
* So, the above steps that we have is called giving subscription access to the users which we have created in AAD

**PowerShell**:

It is task automation and configuration management framework from Microsoft consisting of a command-line shell and associated scripting language.

* In PowerShell tasks are executed by command-lets(cmdlets)
* All PowerShell command-lets are parameterized to ensure maximum flexibility.
* Azure PowerShell is a set of cmdlets designed to help with Management of Azure Resources directly from the PowerShell command lines.

**Below link contains cheat sheet for Azure PowerShell cmdlets:**

[GitHub - andreipintica/Azure-PowerShell-CheatSheet: Cheatsheet with the most common Microsoft Azure PowerShell commands with examples.](https://github.com/andreipintica/Azure-PowerShell-CheatSheet)

Power Shell Commands(cmdlets)::



**Encryption & De-cryption of Azure VM’s Disks using Azure Powershell commands(both OS & Data disks):**

We must have to encrypt our Azure VM's disks, b’coz this disks are stored somewhere in our Microsoft infrastructure(datacenters)and what if Microsoft Infrastructure itself is compromised? that might not happened for sure, but still, nothing can be said and for our security point of view we need to encrypt all the data from our side for our safety, so we can enable the bit locker to enable the entire disks of our Azure VM

If we see before the encryption of our disks then we can find SSE with PMK(for disks) it means that encryption is happening from Azure side by default. The SSE with PMK stands for Server Side Encryption with Platform Managed Key and the Encryption that we do is ADE(Azure Disk Encryption) and this is based on bit locker.

One Azure Key Vault can have keys for multiple Azure VM's, In Key Vault under Access policy tab we have an option Current Access Policies (at the bottom) and here it is showing the name as Nazim Saifi, it means this guy (means global administrator) is going to have access for the Azure Key Vault keys

Access Policy in Key Vault is going to define who is going to have access to the keys and that will be the global administrator and also we can also provide access to others.

**Implementation Steps to Encrypt the Azure VM's disks:**

* Deploy the VM in Azure Portal while attaching a data disk.
* login to the VM and Launch the disks in Azure VM's and ensure the disks are visible in VM and which are not encrypted.
* Create a Key Vault (KV) and fill the below details.

**Subscription:** any

**Resource Group:** Any

**Key vault name:** give any name

**Region:** mostly it is recommended to give same region for key vault as VM

**Pricing tier:** standard

**Days to retain deleted vaults:** 90.

**Purge protection:** Disable purge protection.

**Click on Next:** Access policy and fill the below details.

check the box **Azure Disk Encryption for volume encryption** and leave all the values to default as is and finally.

click on Review+Create to deploy the key vault

4)Now open the cloud shell(in Azure Portal) and run the PowerShell command as mentioned below and note the VM should be in running status while executing this PowerShell commands.

**##Encrypt the Disk**

$KeyVault = Get-AzKeyVault -VaultName Techknowledge -ResourceGroupName MyRG

Set-AzVMDiskEncryptionExtension -ResourceGroupName MyRG -VMName Myvm -DiskEncryptionKeyVaultUrl $KeyVault.VaultUri -DiskEncryptionKeyVaultId $KeyVault.ResourceId

**##Check Disk Encryption Status**

Get-AzVmDiskEncryptionStatus -VMName Myvm -ResourceGroupName MyRG

**## Disable Disk Encryption**

Disable-AzVMDiskEncryption -ResourceGroupName MyRG -VMName Myvm -VolumeType OS **>> for OS disk only**

Disable-AzVMDiskEncryption -ResourceGroupName MyRG -VMName Myvm -VolumeType Data **>> for Data disk only**

Disable-AzVMDiskEncryption -ResourceGroupName MyRG -VMName Myvm -VolumeType All **>> for all(OS or Data) types of disks**

**##Delete Azure Key Vault**

Remove-AzKeyVault -VaultName "Testvm1" -PassThru

**Note:**

We always have to encrypt the OS disk(C drive) first then later we can encrypt the data disk, if we try to encrypt the data disk then will get an error.

**Automation using powershell for the installation of IIS in multiple Azure workloads/VM’s/nodes**

**Internet information services/server(IIS):**

* It is the most powerful web server from Microsoft which is used to host websites/web applications and other web content. It has its own process engine to handle the request. When a request comes from client(web application) to IIS server it takes the request process it and send the response back to client.
* IIS runs on windows server and serve the request for HTML pages and files; it accept the request from remote client computers and returns the appropriate response. IIS supports HTTP/HTTPS/FTP/FTPS/SMTP/…etc. To enable the IIS in a VM, first we have to install an IIS in a VM then we have to configure the IIS in a VM using server manager with in a VM.
* if you are having a requirement in your project where your client is saying that install one software(IIS) in all the VM's....5 VM's or 10 VM's or 20 VM's and if you to each and every VM individually and install a software then it is a kind of mechanical job...it is not recommended…instead of that we will do an automation using PowerShell Cmdlets where I will select the VM then automatically the software will be installed in my VM's whatever the VM's we have selected.

**Desired State Configuration(DSC):**

Desired State Configuration is a management platform in PowerShell that enables us to manage our resources with configuration as a code.

DSC in Azure allows you to write, manage and compile configurations, import and assign configurations to targets servers. The most important feature is that all this runs in the cloud!

**Scenario or example:**

Let’s imagine we have few servers in our environment, and we want to deploy IIS or any tool or we would like to copy some files from our Storage Accounts to a specific path on the servers.

If you want to install multiple software’s in your servers/VM's if there are 10, 20, 30 40 50 VM's but if you are having 3000 VM's to manage and install the software’s in it as per your project requirement then manually you cannot go to each and every VM and install the required software’s in your Servers/VM's.

Basically, this concept is an automation where you are creating one PowerShell script and you attach this script to your nodes/Servers/VM's, hence automatically once you attached your server to the DSC then automatically all the software’s will be installed in your VM's as per your project standards.

**Implementations steps for DSC(with automation account & PS Cmdlets for Azure virtual machines)**

**Step1:** Deploy 2 VM’s(Webserver1 & Webserver2)

**Step2:** Deploy Automation account.

**Step3:** In VM check extensions(Azure portal) and login to the VM and check if any extensions/software’s installed in server manager that IIS is been installed or not.

**Step4:** Goto automation account(Here we are installing IIS using Desired state configuration with PowerShell Cmdlets)>>State Configuration(DSC left side)>>

Configurations (@ the centre)>>+Add>>browse>>select the file>>Name: IISInstallations >>Description: Installing IIS on multiple servers using PowerShell script>>Ok



**Step5:** Now click on the IISInstallations(centre)>>Compile>>Yes.

**Step6:** Come to automation account>>State Configuration (DSC)>>Configurations>>double click on IISInstallations (centre)>>**wait for some 10 mins** and here it will show the job status as completed.

**Step7:** Come to automation account>>State Configuration (DSC)>>Nodes>>+Add>>Click on VM>>Connect>>Node configuration Name: click on and select the value>>Configuration Mode: ApplyAndAutoCorrect>>leave rest all values to default and click on OK finally.

**Step8:** Now **wait for some 8-10 mins** and come to VM extensions and will see the Microsoft.Powershell.DSC has been provisioned and the status will be showing as completed.

**Step9:** Login back to the VM and in Server manager will see the IIS has been installed.

Hence the Desired State Configuration has been implemented using Azure Automation account for azure workloads/Azure Virtual machines.

**Hence, we have deployed/installed the software IIS on multiple nodes/VM's using PowerShell scripting via Azure automation account.**