AZURE DEVOPS

UT-Kloud

Prepared by – Advisory Team

Table of Contents

Section 1

BASICS OF DEVOPS

WE WILL LEARN

1. What is DevOps
2. What is need of DevOps
3. Who should learn DevOps
4. What are the benefits of DevOps

DEVOPS IN GENERAL

life cycle of the software development

Planning: Begin by making a plan. **Development**: Write the code. **Building**: Build the code for testing. **Testing**: The testing team tests the product. **Deployment**: Deploy the code on the server. **Operations**: Perform necessary operations post-deployment. **Monitoring**: Monitor server performance and scaling needs. In start-ups or small teams, the same team handles all these tasks. In larger organizations: **Business/Planning Team**: Handles planning with the product owner and client. **Development Team**: Includes developers and testers; they code, build, and test the product. **Operations Team**: Deploys the code, monitors, and operates the server. **Waterfall Model**: Teams worked independently. **Agile Model**: Business team and development team work together for product success. Operations team bridges the gap between development and operations.

What is DevOps?

Introduction to DevOps: DevOps combines "Dev" from development and "Ops" from operations, indicating collaboration between these two teams. - DevOps bridges the gap between development and operations, similar to how Agile integrates business teams and development teams. **DevOps as a Culture:** DevOps is a culture or practice, not a tool, process, or methodology. - It ensures collaboration throughout the software development life cycle, enhancing product quality and compatibility across environments. - Importance of **DevOps:** Developers and operations work together to maintain consistent environments. - Addresses the common issue where code works on a developer's machine but fails in other environments. - **Automation in DevOps:** Numerous tools available to automate the DevOps process. - Tools vary in configuration, pricing, and features, but all aim to automate the development life cycle. – **Continuous Integration and Deployment (CI/CD):** Every code commit triggers automatic build, test, and deployment processes. - Ensures quality and operational aspects are maintained with each code change.

DEVOPS IN GENERAL

Who should learn DevOps?

Everyone in a Software Development Team:**Developers**: Improve development and deployment processes.**Testers**: Enhance testing efficiency and integration.**Product Owners**: Understand the development and operations workflow.**Scrum Masters**: Facilitate team collaboration and process improvement.**Operations Team**:Essential for all members responsible for deployment, management, and monitoring.**Individual Contributors**:Improves development speed and process efficiency.**Freelancers**:Ensures code management and accessibility, preventing data loss from system crashes.**Management Team**:Provides insights into development progress and operational efficiency through graphs and reports.**Clients**:Offers a clear picture of development progress, bug tracking, and story management through various metrics and graphs.**Anyone Related to Development**:If you are involved in any aspect of development, learning DevOps is beneficial.

What are the benifits of DevOps

More Agility:Treats every commit as a final delivery, ensuring continuous integration and delivery with every code change.**Easy to Use**:Simple and quick to learn, making it accessible for all team members.**No Maintenance**:Low to no maintenance cost, streamlining the development process.**Reliability**:Ensures consistent quality of products and services.**Security**:High security, especially when using tools like Azure DevOps, which guarantees secure processes.**Additional Benefits**:Enhances collaboration among teams, brings more agility to development, and ensures high-quality, well-tested, and well-deployed products.

WHAT IS AZURE DEVOPS

- What is Azure DevOps
  - Azure Devops is set of modern services which is used to
  - Plan smarter
  - Collaborate better
  - Ship faster
  - Azure DevOps provides all these faetures,you can use all of them or choose from them as per your need and role
  - Azure DevOps is developd and managed by MS
  - Azure DevOps was forerly known as Visual Studio Team Services (VSTS)
  - Azure Devops providers a setof intergrated features that you can access through your web browser
- Azure DevOps Services
  - Board
  - Repos
  - Pipelines
  - Test Plans
  - Artifacts
- Relation between Azure and Azure DevOps
  - Azure is a cloud solution that provides lots of services and features to help your organization in almost all ways during the development of any product by providing multiple tools and technologies
  - Azure DevOps is a feature or service of Azure
- VSTS to Azure DevOps
  - Sep 2018 MS renamed VSTS(Visual Studio TeamServices) to Azure DevOps Services

VSTS	Azure DevOpsServices
Build & Release	Azure Pipeline
Code	Azure Repos
Work	Azure Boards
Test	Azure Test Plans
Packages	Azure Artifacts

CREATE AN ACCOUNT ON AZURE DEVOPS

Introduction to Azure DevOps and How to create an account on Azure DevOps

Create an Azure Account

It can be accessed using the URL https://dev.azure.com

You can access it using any broswer like chrome and edge

Navigate to Azure Portal

Access the Azure portal, which is the web-based user interface for managing your Azure resources, by visiting the Azure website and logging in with your account credentials.

Explore Azure Services

Browse the various Azure services and features available, such as virtual machines, storage accounts, databases, and more, to find the resources you need for your project.

Configure Settings

Customize your Azure account settings, such as your profile information, billing preferences, and access permissions, to personalize your experience and secure your account.

By following these steps, you can successfully create an Azure account and navigate the Azure portal to access the cloud services and tools needed for your project or organization.

CREATE AN ORGANIZATION (ORG) IN AZURE DEVOPS

- Organization on Azure DevOps

  In Azure DevOps, there is a structure of your work that depends on organization, Projects, and team

  An organization is a mechanism for organization and connecting groups of related projects.

  Each project must belong to an organization

  To work with Azure DevOps you must have at least one organization

  You get one default organization while creating the account, you can make more than one organization also

  Youcan create organizations for business Units, Sub compnies, Clients

  Each company will have its own and unique URL hence org name should be unique- https://dev.azure.com/suhailkhan2/UT-Kloud

- Create and Configure Organization Settings

  Once you login to the Azure DevOps page, in the upper left-hand corner we will see the default Organization, and in the bottom left-hand corner an option to add a new Organization

  Select unique name and region based on data saving location

  Choose the Org and click on Organization settings in the left bottom

  Overview- gives the option to change the Org name, the region can not be changed

  Org owner can be changed, Org can be deleted and billing info

  Project- All projetct can be seen here

  Users- All users will be available here, Azure DevOps is free if the team size is less than 5 members

  Global Notifications- Notifications like Build completed, Pull request,

  Extension mgmt, Deployment

  Usage-Shows the utilization

  Extensions- Allows third party extenstion like Slack, Code Search

  Azure AD- To connect Azure AD with Azure DevOps account

PROJECT IN AZURE DEVOPS 1/2

What is an Azure Devops Project

A project is a place where the actual complete productDevlopment is done by the team

It provides a Repository for source code and a place for groups of people to plan, track progress, and collaborate on building solutions

all the data related to software is stored under a project, each project must have an Org

A Project provides us all the main services of Azure DevOps. Repo, Boards, Pipeline, Test plan, Artifacts

By default a team with the same as project is created with a new project

Types of Projects in Azure DevOps

Public- Visible to everyone on the internet without login just by using a URL in the browser

Mostly used for open-source development

We can unlimited number of Public project

Private- Visible to limited users with access

Mostly use for non-public software development

PROJECT IN AZURE DEVOPS 2/2

How to Create an Azure DevOps Project

On the main page select the desired Org and click on Create Project

Select Public or Private

Under advacned Select Version control and Work item Progress

Project Settings

Project Settings in the bottom left- We can view Process and update Name and visibility

We can see and add project administrator

We can see services, Enable/disable, delete project

Teams- We can see default team and its members

Security- allteams and AzureDevOps groups are visible here

Notification- Tomanage notifications

SETTING UP A TEAM IN AZURE DEVOPS

1. Introduction to Team

A team is a group of people who are responsible for the software development

A team includes Developers, QA, Scrum Master, PO, Person responsible for deployment etc

A user must have Administrator level access to make changes in team

2. Invite a new member to work on the project

Invite the members of your team to the project by adding them to the project settings and assigning relevant permissions.

Use invite button at the top right hand corner and search members from teams (which got created at the time of project creation) or by their email address

We can add new team members using Teams option under project setting -- click add and we can add single or multiple users

Incase email needs to be resent then go to Org Settings --Users-- Select the user--ellipsis--Resend Invite

3. Set member permission

Utilize Azure DevOps Boards to manage your team's work items, such as tasks, bugs, and user stories, and to track the progress of your project.

To change access level- Org Settings --Users-- Select the user--ellipsis--Manage User--Select Project and assign role -- Project Reader-, Project Contributor-, Project Administrator-.

To manage user- Org Settings --Users-- Select the user--ellipsis--Change access level--

Basic-, Stakeholder-, Visual studio subscriber-.

Section 2

AZURE BOARDS

AZURE BOARDS IN AZURE DEVOPS 1/2

What is a board in Azure DevOps

It is one of the main services of Azure DevOps which is used in project to

Track the work with Kanban Board

Work with Backlog

Team Dashboard

Reporting

Azure board provides drag and drop feature to update Kanban Board https://shorturl.at/bwb7J

Azure boards are fully compatible with SCRUM

What is the use of boards

Azure boards are useful in-

Managing and tracking work

It provides a clear picture of work done/doing by team member

You can give access of board to your stakeholder allow him to see the progress of your development

You can manage scrum, sprint with Azure boards

Manage your product backlog

Get different types of reports

AZURE BOARDS IN AZURE DEVOPS 2/2

Who will work on board

The Team

Development team will work on managing their work (task, stories, bug etc)

PO(Product Owner) will work to manage the backlog

Scrum Master will work on Azure board to get different types of reports and to see the progress of them

What type of bards are available in Azure DevOps

There are four types of **work item process** while creating a new project

The type of board depends on these work item prcess

Each type provide different types of work item and work flow

While creating the project under advance we see this option (Basic is default)

AZURE DEVOPS WITH BASIC PROCESS

- We will Discuss
  - What is work item
  - What type of work items are available in Azure board with basic process
  - Introduction to Kanban board
  - Create a work item
  - Update work item status
  - Add task to an issue

WHAT IS A WORK ITEM 1/2

A work item is a unit (small or large) of work which has several characteristics and is a part of your product development

Characteristics Like-

- Title
- Description
- Assigned to
- Completed by
- etc

Example of work Item

- **Bug**-Something which is missed or implemenered in wrong way
- **Epic**-An epic reprsents a business initiave to beacomplished
- **Feature-** A feature typically represnts a shippalbe component of software
- **User Story-** Implenetation of new work
- **Issue-** Any other custom type
- Improvemnt
- etc

WHAT IS A WORK ITEM 2/2

Sprint

Each sprint must have start and end date in Agile.

Smallest unit of the work

Task

we can create multiple tasks under issues

Login functionality issue can have multiple tasks like - Creating table in DB, creating API, integration of API, designed, all the tasks to complete an issue.

Product Backlog item

Definition: A Product Backlog item (PBI) is a unit of work that needs to be completed in order to develop a product.

Example: As a user, I want to be able to search for products by name.

Product Backlog-: The Product Backlog is a prioritized list of Product Backlog items (PBIs). Purpose: The Product Backlog is used to communicate the product vision and roadmap, and to guide the development team in their work.

Sprint Backlog-The Sprint Backlog is a list of the work that the development team will complete during a sprint. Purpose: The Sprint Backlog is used to plan and track the progress of the sprint, and to ensure that the team is working on the most important tasks.

Impediment-

Definition: An impediment is anything that blocks or slows down progress (Something which has dependency, unless task A is completed we can not start task B).

Purpose: Impediments are identified and tracked so that they can be resolved and the team can continue to make progress.

AZURE BOARD

Work items in Azure board with different process				
Work item process in Azure DevOps (Need to select this at the time of project creation) Always Check the portal for updated options.	Basic	Scrum	Agile	CMMI
	Epic	Bug	Bug	Bug
	Issue	Epic	Epic	Change Request
	Task	Feature	Feature	Epic
		Impediment	Issue	Feature
		Product Backlog item	Task	Issue
		Task	Test Case	Requirement
		Test Case	User Story	Review
				Risk
				Task
				Test Case

AZURE BOARD --> KANBAN BOARD

Work flow

Workflow is the process if updating work item progress, Eg- Login button function
Basic Flow - TO DO --> In Progress --> Done (Works both ways)

Kanban Board

Azure DevOps Portal --> Boards --> Boards
States - To Do --> Doing --> Done
We can create work items here, all work item will have unique IDs -- We can assign this any team member
We can move the state simply by drag and drop option
From To Do to Doing, Doing to Done
We can add a bug and move Back the item from Done to To Do or doin

AZURE BOARDS WITH ALL PROCESSES

- We will discuss and practice on portal
  - Create a new work item on Azure Board
  - What fields are available in a work item
  - How to add attachment to a work item
  - How to see history of work item
  - Change column options on azure board

COLUMN OPTIONS ON AZURE BOARD

- Agenda
  - Azure board is used to see lots of details related to work items
  - These details have lots of columns(Information)
  - Example- Work items, Backlog, Queries etc
  - What is column options
  - Add column options
  - Remove column options
  - Sort by column

QUERY IN AZURE DEVOPS

- We will discuss and practice on portal
  - What is a query
  - How to see query dashboard
  - How to view a query
  - How to rename, delete a query
  - How to send a query on email

CUSTOMIZE PROJECT USING INHERITED PROCESS

- We will discuss and practice on portal
  - What is an inherited process
  - How to create inherited process
  - How to create a project using inherited process

CREATE CUSTOM WORK ITEM AND FIELDS AND CUSTOMIZE THEM

- We will discuss and practice on portal
  - Create custom work item in inherited process
  - Create custom field under work item
  - Customize work item's layout
  - Add new custom field
  - Set location and type of field
  - Set a default value of field
  - Make field required
  - Add new tab to work item
  - Verify result

ADD A COLUMN IN KANBAN BOARD

- Add a column in the Kanban board
  - How to add new column
  - Rename a column
  - Move order of columns

UPDATE FIELDS FROM THE CARDS, DEFINE STYLE RULES TO HIGHLIGHT CARDS & ADD A COLOR TAG TO WORK ITEM CARD

- Update fields from the cards
  - Customize visible card's fields on Kanban board
  - Show more fields
  - Hide fields
  - Lots of other costomization
  - **Define style rules to highlight cards and Add color tag to work item card**
  - Add and Delete style rules
  - Add condition in rules
  - Set color for tag

Section 3

AZURE DEVOPS REPO AND VERSION CONTROL

WHAT IS REPO IN AZURE DEVOPS | WHAT IS VERSION CONTROL

- What is Repo in Azure DevOps | What is Version control
  - What is Repo
  - What is version control
  - What is the use of Repo
  - Who should use Repo
  - What are the types of version control in Repo

WHAT IS REPO IN AZURE DEVOPS | WHAT IS VERSION CONTROL

- We will learn
  - Git vs TFVC
  - How git works
  - How TFVC works
  - Common features in Git & TFVC
  - Which one should we use for our project

DISTRIBUTED VS CENTRALIZED SOURCE CONTROL

1. **GIT Distributed version control**
   - Both are used to manage code
   - Both provide branching system
   - Both are used to merge code
   - Operations in Git are fast as everything happens in a distributed mode. (Eg: Branch switching is easy)
   - In Git, an entire replica (i.e. Local Repo) is available on the developer's machine
   - Merging, Pull request, Code review is easy in Git
   - Git is used by most of the software development team

1. **TFVC Team Foundation version control (Centralized version control)**
   - Both can be used by indivdual or team
   - Both provide history of changes
   - Both are used to merge code
   - TFVC does not have the entire replica of the local machine (Code will be available however features like History, branches, etc are not available locally)

SETUP AND WORKING ON GIT

1.What will we learn on the portal

- Create project with Git or TFVC
- Setup Git on local system
- Clone Azure DevOps Repo
- Git Commit, Push & Pull in Azure DevOps Repo
- Working directly on azure devops server repository
- Branch in azure devops repo | What is branch in git
- Create new branch in azure devops using server repository
- Create new branch in azure devops using local repository
- How to create Pull Request in azure devops

Section 4

AZURE DEVOPS PIPELINES

WHAT IS PIPELINES IN AZURE DEVOPS | WHAT IS CI & CD

1.What will we learn

What is Pipelines in Azure DevOps

What processes are avaiable in azure DevOps pipelines

What type of application and programming langauage can we used in pipeline

Where can I store my project code to use piplie

Wheere can I deploy (Server, VM, etc) my project code

Is pipeline free to use

What is CI Continuous integration | Build Pipeline (for developers) & CD Continuous delivery | Release Pipeline (for operations)

When CI & CD will be triggered

CREATE A BUILD (CONTINUES INTEGRATION) PIPELINE IN AZURE DEVOPS

- Create Build (CI - continues Integration) pipeline using YAML (for developers)
- Create Build (CI - continues Integration) pipeline using classic editor if not familiar with YAML

KEY CONCEPT OF AZURE DEVOPS PIPELINES

1.What will we learn

- Key concepts in Pipeline
- What is Agent
- Approvals
- Artifacts
- Environment
- Job
- Run
- Stage
- Trigger

AGENT

- When your build or deployment runs, the system begins one or more jobs. An agent in installable software that runs one job at a time (There will be at least one job or more than one job in the pipeline).
- To build your code or deploy your software using Azure Pipelines, you need at least one agent. Because to deploy software you need at least one job and to run that job we need one agent.
- In Azure DevOps we can use two types of agents

Microsoft-hosted agents—These agents are installed by MS automatically. Whenever you run a pipeline, a new VM (which is defined in your YAML file) gets created, and these agents run on it. The basic job of these agents is to get the code from the repository and build it.

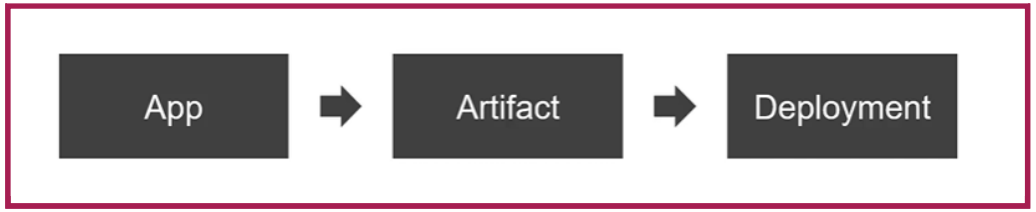
Self-hosted agent— This agent is installed by you. If you need more control over your agent, then use this option. However, if you do not want much administration on the agent, then use an MS-hosted agent.

APPROVALS

- Approvals are a set of validations that are required before a deployment can be performed, Deployment is a release pipeline and a set of validations required before Deployment can be performed. Example- Getting permission from someone in a team before deployment on production.
- In general setup we create multiple environments like deploying the code in Test, Dev, Stage, and then production, since prod is critical hence we need some validation/approval before deploying anything in a production environment, we need approval from one or more than one person from the team. This approval can be used in multiple stages like staging and production.

ARTIFACTS

- An artifact is a collection of files or packages that are created by a build run, when we run a build pipeline then some files get created after running the pipeline, for example: If a .not core application is there --> When we run build pipeline --> we run publish command --> we get some files which are required for the release to run the application, collection of these files called artifacts.
- These artifacts are then made available for the next task ie: deployment: These can be used for release pipeline/deployment. Example: App --> run build pipeline --> outcome is Artifact --> when we run the release pipeline on this particular application then this artifact will be used as a source --> and source will be deployed on a particular server that is called deployment. The output of the build pipeline or the source of the release pipeline is called the artifact, this artifact is the collection of all the files that are created from the build and everything that is required to run that 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 something where we Deploy our application.
- An environment is a collection of resources like- VMs, containers, Web ap, etc, if you have an angular application and you want to deploy that to a VM there must be a setup for this, this setup is called an environment.
- A release pipeline can deploy the code on one or more VMs (environments) after the build pipeline is completed.

JOB AND RUN

Job

- A job represents an execution boundary of a set of steps All of the steps run together on the same agent Example- You might build two configurations -x86 and x64. In this case, you have one build stage(build application) and two jobs one for x84 bit and one for x64 bit. One job can be run by only one agent at a time.

Run

- One execution of a pipeline (build or release) is known as run, success or failure of that pipeline doesn't matter, just the starting point is called RUN

STAGE

- A stage is used to mark the separation of concerns, it is used to create separation inside the deployment/build pipeline. Example- Creating a build for QA, Staging, Production, etc Before publishing the code on production deployment/server we create some different stages like Dev, QA, Staging, and Production - these are the basic four stages we create for a general application, stages can be increase or decrease as per your application need.
- First, we start building and releasing on the Dev environment then QA and Staging then the Production environment - all these are called stages and need approval after every stage. Each stage contains one or more jobs. The stage is very important in adding an extra level of validation before your code goes to production.

TRIGGER

- A trigger is a setup that tells the pipeline when to run We can configure a pipeline when

A new push in the repo (build pipeline will run automatically)

At a scheduled time (build pipeline will run automatically)

Upon completion of another build (build pipeline will run automatically) -- once the build pipeline gets completed then the release pipeline will run automatically

We can also run these manually.

CREATE A RELEASE (CONTINUES DELIVERY) PIPELINE IN AZURE DEVOPS

1.What will we learn

How to create a Release (Continues Delivery) pipeline

Deploy code on VM, we will use a VM with Windows OS with an IIS role, we will deploy our code from the Azure repository to the VM

Deploy code automatically- We will make some changes in the code then the build pipeline should get triggered automatically

Once the build pipeline gets successfully completed then the release pipeline will get triggered automatically

once the release pipeline is completed then we should be able to see the changes that we made in the code reflected on our site hosted on IIS

Deployment group

MULTI-STAGES IN RELEASE (CD) PIPELINE WITH PRE-DEPLOYMENT APPROVALS

1.What will we learn

How to create multiple stages in relese (Continues Delivery) pipeline (for operations)

Set triggers

Deployment group