**Azure DeVops**

**(**[**https://youtu.be/q8EevlEpQ2A?si=knBQ1SOkybzKGJ7y)Imp**](https://youtu.be/q8EevlEpQ2A?si=knBQ1SOkybzKGJ7y)Imp) **for git**

**Topics to cover-**

**Dockers-** **https://youtu.be/H8Lyj2D\_cWo?si=3DUSNQdbzyOamH2Q**

**Step1-GIT And GitHub basics https://youtu.be/uwyP5SNlQAU?si=faDhA2NYOV8sqO51**

**Step2-basic understanding of (AzureDevops ,CICD pipeline)(** **https://youtu.be/57dkyTmOCmE?si=zkNfb1gPWFwLHJqD)**

[**Azure DevOps Bootcamp: Zero to Hero (Pipelines,Boards,Repos) | Udemy Business**](https://cisco.udemy.com/course/azdevops/)

** Java: Mastering Java is essential as it's the foundation for your backend development.**

** Spring Boot: After learning Java, Spring Boot will help you build stand-alone, production-ready applications more efficiently.**

** API Creation and Integration: Understanding how to create and integrate APIs is crucial for enabling communication between services.**

** Git and GitHub: Learning version control will help you manage your code effectively and collaborate with others.**

** Azure DevOps through CI/CD: This will teach you how to automate your development workflow, from coding to deployment.**

** Microservices: Finally, learning microservices architecture will allow you to build scalable and maintainable applications, leveraging your knowledge from the previous steps.**

**jenkins**

**Index-**

**Day-by-day Table of Topics**

**Git and Githubs (3 hr) ----------------------------------------------------------------------------🡪> 19-09-2024**

| **Day** | **Topics Covered** |
| --- | --- |
| **Day 1** | **Introduction to Azure DevOps and Basic Concepts -------------------------🡪>20-09-24** |
| **Day 2** | **Azure Boards and Agile Project Management -----------------------------------------------------------** |
| **Day 3** | **Mastering Git and Source Control in Azure DevOps -------------------🡪>-21-09-24** |
| **Day 4** | **Build Pipeline (Using YouTube Clone Website)-------------------------------------------------------------** |
| **Day 5** | **Continuous Delivery with Azure DevOps Release Pipeline---------------🡪>22-09-24** |
| **Day 6** | **Azure Test Plans and Testing (Using YouTube Clone Website)** |
| **Day 7** | **Basic Project Artifacts with Azure Artifacts----------------------------------------------------------------** |
| **Day 8** | **Infrastructure as Code (IaC) with Terraform and Azure DevOps -------🡪>23-09-24** |
| **Day 9** | **Self-Hosted Agents on Azure Virtual Machine Scale Sets----------------🡪>24-09-24** |
| **Day 10** | **Managing Containers with Azure DevOps------------------------------------------------------------------** |
| **Day 11** | **Implementing end-to-end CICD using Azure DevOps on Kubernetes-----🡪>25-09-24** |
| **Day 12** | **Security and Permissions in Azure DevOps-----------------------------------------------------------------** |
| **Day 13** | **Serverless app CICD (Using Serverless QR Code Generator)-----------------🡪>26-09-24** |
| **Day 14** | **Azure DevOps Wiki----------------------------------------------------------------------🡪>27-09-24** |
| **Day 15** | **Azure DevOps Security Best Practices** |
| **Day 16** | **Issue and Troubleshooting in Azure DevOps----------------------------------------🡪>28-09-24** |
| **Day 17** | **Bonus: Azure DevOps Scenario-based Interview Questions** |

**Day 1: Introduction to Azure DevOps and Basic Concepts 🌟**

**Status: Video is live, check out 👉**[**Day1**](https://github.com/piyushsachdeva/AzureDevOps-Zero-to-Hero/tree/main/Day1)**👈 folder for notes and useful links ✅**

* **What is Cloud Computing**
* **IaaS VS PaaS VS SaaS**
* **What is a Shared Responsibility Model**
* **What is a Traditional Build and Deployment workflow**
* **What is a Waterfall model in SDLC**
* **Problems with the traditional software development life cycle (SDLC)**
* **What is Agile, and how it solve the above challenges**
* **What is DevOps and Why It Matters**
* **What is CI/CD**
* **What is Azure DevOps and a quick walkthrough**
* **Creating an Azure DevOps Organization**
* **Creating an Azure DevOps Project**
* **Azure DevOps Pricing**
* **Azure DevOps hosting options: Azure DevOps Services VS Azure DevOps Server**

**Day 2: Azure Boards and Agile Project Management 📊**

**Status: Video is live, check out 👉**[**Day2**](https://github.com/piyushsachdeva/AzureDevOps-Zero-to-Hero/tree/main/Day2)**👈 folder for notes and useful links ✅**

* **What are Azure DevOps Boards**
* **What are Azure board processes, agile, scrum, basic, and CMMI**
* **Managing work items in Azure boards**
* **Azure board implementation using basic process**
* **Working with teams, areas, and iterations**
* **Filters in backlogs and boards**
* **Azure board implementation using the scrum process**
* **Sprint planning and capacity planning**
* **Product backlog and taskboard**
* **Customizing Kanban boards**
* **Customizing dashboards**
* **Work item query**
* **Customizing team process**

**Day 3: Mastering Git and Source Control in Azure DevOps 🌿**

**Status: Video is live, check out 👉**[**Day3**](https://github.com/piyushsachdeva/AzureDevOps-Zero-to-Hero/tree/main/Day3)**👈 folder for notes and useful links ✅**

* **Introduction to Source Control and Azure Repos**
* **Git vs TFVC**
* **Configure Visual Code**
* **Cloning the repo**
* **Commit changes**
* **Reviewing history**
* **Working with branches**
* **Tagging a release**
* **Managing repository**
* **Managing Pull requests**
* **Sample application code**

**Day 4: Build Pipeline 🚀**

**Status: Video is live, check out 👉**[**Day4**](https://github.com/piyushsachdeva/AzureDevOps-Zero-to-Hero/tree/main/Day4)**👈 folder for notes and useful links ✅**

**Note: For the demo, we will be using the YouTube Clone website**

* **Provision Azure App Service to host the website.**
* **Creating Build Pipelines using the classic editor**
* **Creating build pipeline using YAML**
* **YAML pipeline structure, the difference between jobs, stages, steps, and tasks**
* **Creating a multi-stage CICD pipeline**
* **variables, triggers, Build properties, agents**
* **Publishing and Download Build Artifacts**

**Day5: 🚀 Continuous Delivery with Azure DevOps Release Pipeline**

**Status: Video is live, check out 👉**[**Day5**](https://github.com/piyushsachdeva/AzureDevOps-Zero-to-Hero/tree/main/Day5)**👈 folder for notes and useful links ✅**

**Note: this is a continuation of the previous video.**

* **Automating Deployment with a multi-stage Release Pipelines**
* **Continuous Deployment Triggers**
* **Continuous delivery using deployment slots to enable Blue-Green deployment**
* **Deployment gates such as Query Work Items and Approvals before the prod deployment**
* **Update the code to test the entire CICD process with the Build and Release pipeline**

**Day 6: Azure Test Plans and Testing 🧪**

**Status: Video is live, check out 👉**[**Day6**](https://github.com/piyushsachdeva/AzureDevOps-Zero-to-Hero/tree/main/Day6)**👈 folder for notes and useful links ✅**

**Note: We will be using the Youtube Clone website to implement the below steps**

* **Azure Test Plan Overview**
* **Features of Azure test plan**
* **Managing Test Plans, Suites and Cases**
* **Subscribe to the test plan free trial**
* **Authoring, Running, and Analyzing Manual Tests**
* **Azure Test and Feedback extension**

**Day 7: Basic Project Artifacts with Azure Artifacts 📦**

**Status: Video is live, check out 👉**[**Day7**](https://github.com/piyushsachdeva/AzureDevOps-Zero-to-Hero/tree/main/Day7)**👈 folder for notes and useful links ✅**

**Note: In this video, we will use a ✔ Nike Landing page as a sample application for CICD using Azure Artifacts**

* **Overview of Azure Artifacts**
* **Create the Azure DevOps project and check out the application code**
* **Set up the infra using Azure Web App**
* **Create Azure Artifacts feed to host the packages**
* **Create the CI pipeline that builds the package and pushes it to the feed**
* **Create the CD pipeline that consumes the package**
* **Promote the package to trigger the release pipeline**
* **Upstream packages in Azure Artifacts**

**Day 8: Infrastructure as Code (IaC) with Terraform and Azure DevOps🚀**

**Status: Video is live, check out 👉**[**Day8**](https://github.com/piyushsachdeva/AzureDevOps-Zero-to-Hero/tree/main/Day8)**👈 folder for notes and useful links ✅**

* **Introduction to IaC and Tools**
* **Various Terraform commands and workflow**
* **Creating Terraform configuration files**
* **Setting up terraform backend with Azure storage**
* **Executing Terraform commands using CLI**
* **Azure DevOps CI Pipeline to init, plan, and archive the plan file**
* **Azure DevOps CD pipeline to apply the changes**

**Day9: Self Hosted agents 👨‍🔧 on Azure Virtual machine scale sets 🧑‍💻**

**Status: Video is live, check out 👉**[**Day9**](https://github.com/piyushsachdeva/AzureDevOps-Zero-to-Hero/tree/main/Day9)**👈 folder for notes and useful links ✅**

* **Microsoft-hosted vs. self-hosted agents**
* **Use case of self-hosted agents**
* **Ways to setup self-hosted agents: VM, VMSS, container**
* **What is a Virtual machine scale set**
* **Set up a self-hosted agent using VMSS**
* **Register the agent on an agent pool**
* **Install custom utilities on the agent**
* **Use the self-hosted agent on a pipeline**
* **Comparison between self-hosted and Microsoft-hosted agents**
* **work folder walkthrough on agent**

**Day 10 Managing Containers with Azure DevOps**

**Status: Video is live, check out 👉**[**Day10**](https://github.com/piyushsachdeva/AzureDevOps-Zero-to-Hero/tree/main/Day10)**👈 folder for notes and useful links ✅**

* **What is a container**
* **Understanding Virtual machine V/s Containers.**
* **Challenges with the non-containerized applications**
* **Docker Architecture**
* **Containerize a sample To-Do list web app written in React JS.**
* **Benefits of a multi-stage docker file**
* **What are Azure container instances(ACI)**
* **Azure DevOps CICD Pipeline to deploy to ACI**

**Day 11 Implementing end-to-end CICD using Azure DevOps on Kubernetes.**

**Status: Video is live, check out 👉**[**Day11**](https://github.com/piyushsachdeva/AzureDevOps-Zero-to-Hero/tree/main/Day11)**👈 folder for notes and useful links ✅**

* **Basic Introduction of Kubernetes and its benefits**
* **Kubernetes Architecture**
* **What is the control plane and its components**
* **What are Nodes and types of Nodes**
* **What is a Pod/Deployment/Service**
* **Azure DevOps CICD Pipeline for a web app running on Kubernetes**
* **Sample application: My Health Care - Microservices-based Healthcare management app**

**Day 12 Security and Permissions in Azure DevOps 🔐**

**Status: Video is live, check out 👉**[**Day12**](https://github.com/piyushsachdeva/AzureDevOps-Zero-to-Hero/tree/main/Day12)**👈 folder for notes and useful links ✅**

* **Enabling advanced security in Azure DevOps**
* **Dependency Scanning**
* **Secret scanning and managing alerts**
* **How to use secrets in your pipeline**
* **Code scanning for vulnerabilities**
* **Sample Application: My Health Care - Microservices-based Healthcare management app**

**Day 13: Serverless app CICD 🐳**

**Status: Video is live, check out 👉**[**Day13**](https://github.com/piyushsachdeva/AzureDevOps-Zero-to-Hero/tree/main/Day13)**👈 folder for notes and useful links ✅**

* **Introduction to Azure functions**
* **Use case and benefits of an Azure function**
* **Introduction to the sample app to be used for this demo: Serverless QR Code Generator**
* **Demo creating the Azure function and deploying locally**
* **Publishing the function to Azure using CLI tools**
* **Build and release pipeline for building and deploying the code to Azure Functions**

**Day 14: Azure DevOps wiki**

**Status: Video is live, check out 👉**[**Day14**](https://github.com/piyushsachdeva/AzureDevOps-Zero-to-Hero/tree/main/Day14)**👈 folder for notes and useful links ✅**

* **Overview of wiki**
* **Creating and editing a project Wiki**
* **Publishing code as Wiki**
* **How we can use Azure DevOps wiki to collaborate on a project**

**Day 15: Azure DevOps Security best practices 🚢**

**Status: Video is live, check out 👉**[**Day15**](https://github.com/piyushsachdeva/AzureDevOps-Zero-to-Hero/tree/main/Day15)**👈 folder for notes and useful links ✅**

* **Azure DevOps Access Control**
* **Organization Settings**
* **Agent pools Management**
* **Pipeline settings**
* **Project-level Settings**
* **Pipeline security**
* **Repo settings**
* **Authentication and Authorization**
* **Secrets and credentials access**

**Day 16: Issue and troubleshooting Azure DevOps**

* **In this video, we will discuss the most common issues you have faced throughout the series and solutions to those. Status: Video is live, check out 👉**[**Day16**](https://github.com/piyushsachdeva/AzureDevOps-Zero-to-Hero/tree/main/Day16)**👈 folder for notes and useful links ✅**

**Day 17: Bonus Video: Azure DevOps Scenario-based Interview Questions**

* **In this video, we will discuss the most asked scenario-based interview questions and answers for Azure DevOps Status: Video is live, check out 👉 [InterviewQuestions](https://github.com/piyushsachdeva/AzureDevOps-Zero-to-Hero/tree/main/InterviewQuestions) 👈 folder for notes and useful links ✅**

**What is cloud computing 🡪**

* Cloud computing is the delivery of computing services-such as storage, Servers, database, networking, Software and more over the internet
* instead of physical hardware or data centers ,organization can access and use resources on demand and pay accordingly
* cloud provides virtualized infrastructure, platform and applications

**Types Of cloud computing🡪**

There are Three types of Cloud

1. **IaaS(Infrastructure As a service)**

* Provide virtualized Computing resources like virtual machine, Storage, Networks

**Cloud Provider Responsibility**:

* **Physical Infrastructure**: Security of the data center, servers, storage, and network.
* **Networking**: Ensuring network availability and security at the infrastructure level.

**Customer Responsibility**:

* **Operating System**: Installing, updating, and securing the OS.
* **Applications**: Deploying and managing applications.
* **Data**: Data security, encryption, backups, and compliance.
* **Firewalls and Network Configurations**: Configuring firewalls, security groups, and virtual networks within the cloud environment.

1. **PaaS (Platform as a service)**

* Provide a platform to develop ,test, and deploy applications without Worring about underlying infrastructure

**Cloud Provider Responsibility**:

* **Physical Infrastructure**: Data center, servers, networking, and storage.
* **Operating System**: OS updates and patches.
* **Platform Runtime**: Runtime environment for applications (e.g., Java, .NET).
* **Middleware**: Frameworks and tools for application development (e.g., databases, web servers

**Customer Responsibility**:

* + **Applications**: Development, deployment, and management of their own applications.
  + **Data**: Securing, managing, and backing up their data.

1. **SaaS(software As service)**

* Provide access to fully managed applications via the internet

** Cloud Provider Responsibility:**

* Everything: Infrastructure, networking, servers, storage, OS, applications, and updates.
* Application Security: Securing the application, ensuring availability, updates, and compliance.

** Customer Responsibility:**

* User Data: Managing, securing, and backing up their own data.
* User Access and Identity: Configuring user accounts, roles, and permissions**.**

**What is OnPrem Application🡪**

* The software is hosted locally on companies own server like Cisco, rather than on cloud
* The company has full control over its data ,software, And servers
* The company IT team must take care Of al Aspects

Comparison With Cloud🡪

on-prem applications require more maintenance and upfront costs but offer more control. No cloud provider is involved.

**Customer Responsibility:**

* **Physical Security**: Protecting the physical servers and data center.
* **Networking**: Managing firewalls, routers, and connections.
* **Servers**: Installing, configuring, and maintaining servers and storage.
* A screenshot of a table

  Description automatically generated**Operating System (OS)**: Installing and updating the OS.
* **Applications**: Deploying, managing, and securing applications.
* **Data**: Backups, recovery, security, and compliance

**Shared Responsibility Model with a Traditional Workflow Example (Developer, Support, Ops, Testing)**:

**1. On-Premises (On-Prem)🡪**In the on-premises model, everything is managed by the company’s internal IT teams. The company owns the physical infrastructure (servers, networks) and manages the entire software stack, from hardware to applications.

Workflow:

* Developers: Write code for the banking application, including front-end, back-end, and database components.
* Operations (Ops): Manage the servers, networking, and storage. They install and configure the operating systems, middleware, and runtime environments.
* Testing: Test the application in staging environments hosted on internal servers.
* Support: Provides post-deployment support and deals with infrastructure or application issues.

Challenges:

* High costs for hardware and infrastructure maintenance.
* Slow scalability and difficulty in handling peak loads.

**2. IaaS (Infrastructure as a Service)🡪** With IaaS, the company moves the physical infrastructure (servers, storage, networking) to a cloud provider like AWS or Azure, but they are still responsible for the operating system, middleware, application, and data.

Workflow:

* Developers: Still write the code, but now they deploy the application to virtual machines in the cloud.
* Operations (Ops): No longer manage physical servers. Instead, they configure virtual machines (VMs) and handle the OS, networking (firewalls, security groups), and storage.
* Testing: Test the application in cloud-hosted environments, using cloud-based staging servers.
* Support: Support continues to monitor the virtual machines and application performance, dealing with issues related to the OS and the application.

**3. PaaS (Platform as a Service)🡪** In PaaS, the cloud provider handles the infrastructure, OS, runtime, and middleware, while the company is only responsible for the application code and data.

Workflow:

* A white text with black text

  Description automatically generatedDevelopers: Focus entirely on writing and deploying the code. They use cloud services like Azure App Service to host the application.
* Operations (Ops): Minimal involvement since the platform manages the infrastructure and middleware. They focus on configuring platform settings and scaling.
* Testing: Testing teams use automated testing services within the PaaS environment.

Support: Support focuses more on monitoring the application itself, as infrastructure management is done by the cloud provider.

**A waterfall with many colorful icons

Description automatically generated with medium confidenceWaterfall Model**

**A purple and pink infinity symbol with white text

Description automatically generatedA diagram of a system

Description automatically generatedAgile Model**

**S**