**Organization**

An **organization** in Azure DevOps is the top-level container that encapsulates all your projects, users, and resources. It acts as a central hub where you manage the overall configuration, permissions, and policies that apply across all projects within the organization. Key aspects of an Azure DevOps organization include:

* **Identity and Access Management**: You can add users, assign roles, and manage permissions to control who has access to various resources and what actions they can perform.
* **Billing and Subscription**: The organization level is where you manage billing information and subscriptions for Azure DevOps services.
* **Settings and Policies**: You can configure settings that affect all projects under the organization, such as security policies, repository settings, and pipeline permissions.
* **Resource Sharing**: Shared resources like agents, service connections, and extensions are managed at the organization level and can be used across multiple projects.

**Project**

A **project** in Azure DevOps is a container within an organization that encompasses all the artifacts, tools, and processes needed to plan, develop, and deliver software. Each project is isolated from others, allowing for distinct workflows and configurations. Key elements of an Azure DevOps project include:

* **Repositories**: Each project can have its own set of repositories (Git or TFVC) for version control of code and other files.
* **Pipelines**: Projects can have their own CI/CD pipelines to automate building, testing, and deploying applications.
* **Boards**: Each project can use Azure Boards to manage work items, track progress, and plan sprints with features like backlogs, Kanban boards, and dashboards.
* **Test Plans**: Projects can include their own test plans, test suites, and test cases for manual and automated testing.
* **Artifacts**: Projects can manage and store their own packages using Azure Artifacts.

**Relationship Between Organization and Project**

* **Hierarchy**: An organization can contain multiple projects, but a project belongs to a single organization.
* **Isolation and Sharing**: Projects are isolated from each other by default, which helps maintain a clean separation of concerns and security boundaries. However, resources and policies can be shared and managed at the organization level for consistency and centralized control.
* **Scalability**: This structure allows organizations to scale by adding more projects as needed, each with its own set of repositories, pipelines, boards, etc., while maintaining centralized management.

Note: Organisation is a implementation of cloud, and project is a implementation of DevOps