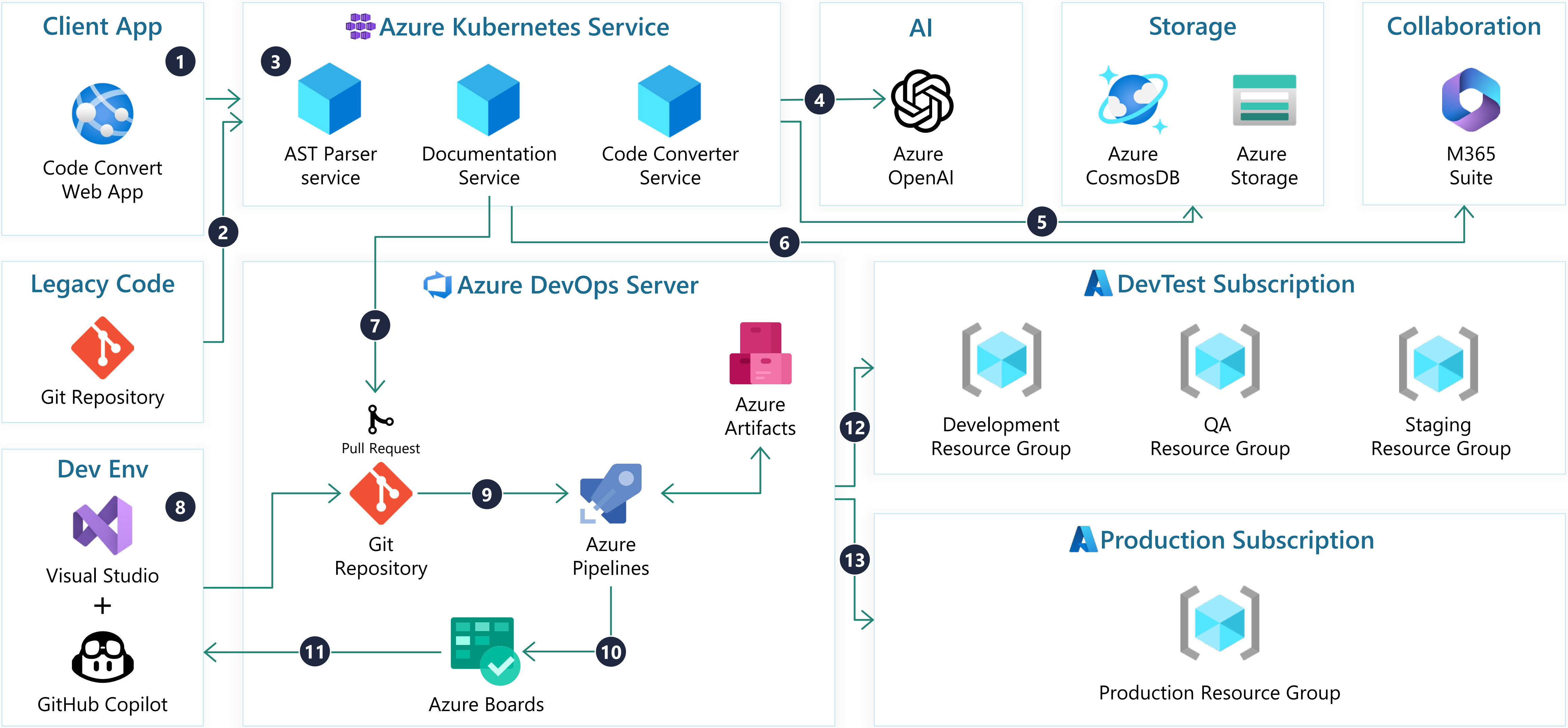
Migrate legacy applications

Migrate legacy apps Reference Architecture



Migrate legacy apps Reference Architecture

1 Developers use the Code Converter Application to translate code from the legacy COBOL language to more modern programming languages, such as Python and .NET. This facilitates the modernization of legacy systems by integrating them with contemporary technology stacks.

2 The Code Converter application is deployed in an Azure Kubernetes Service. This setup allows the application to access a repository containing legacy code, enabling the conversion process.

3 The application consists of three primary services:

- AST Parser: This service parses the Abstract Syntax Tree (AST) of the source code, which is a critical step in understanding the structure and elements of the COBOL code.
- Code Converter: This service takes the parsed COBOL code and converts it into modern programming languages like Python or .NET.
- Documentation: This service is responsible for generating documentation for the converted code, which is essential for understanding and maintaining the new codebase.

4 Integration with Azure OpenAI Service: These services leverage models from Azure OpenAI Service for various tasks. The AST Parser and Code Converter services use these models for parsing and converting the code, while the Documentation service uses them to generate comprehensive and understandable documentation.

Migrate legacy apps Reference Architecture

5 Data Storage Using CosmosDB and Azure Storage: The application uses Azure CosmosDB and Azure Storage for storing application-related information. This includes data generated during the code conversion process, ensuring organized and secure data management.

6 Documentation Storage and Collaboration via SharePoint: The generated documentation is stored in SharePoint. This allows for easy collaboration among team members, as SharePoint is a platform designed for document management and team collaboration.

7 Code Conversion Service Features: The Code Conversion Service includes a feature for creating pull requests in the code repository. This enables other developers to review the proposed changes easily. Once reviewed, these changes can be merged into the development branches, streamlining the integration of converted code into ongoing development efforts.

8 Developers use development environments through Visual Studio or Visual Studio Code, both of which are highly popular and feature-rich Integrated Development Environments (IDEs). A key enhancement in their workflow is the integration of GitHub Copilot. GitHub Copilot, powered by OpenAI, acts as an AI-powered coding assistant that suggests whole lines or blocks of code as the developer types.

9 Azure DevOps Server is a comprehensive suite of software development tools to support software development projects. The main components of Azure DevOps Server include: Boards, Pipelines and Repos. Source code of the new application is available in Git repositories builds to trigger on Git commits or pull requests. In Azure Artifacts, successful builds are systematically stored to provide a centralized repository for versioned artifacts.

10 Boards connects back to Repos to generate work items from automated and manual testing and tracks all work in the development cycle.

Migrate legacy apps Reference Architecture

11 Developers create feature or fix branches that associate work items with development sprints or cycles, tracking all code with tasks and requirements. With this approach, developers maintain an active development loop velocity that is critical to agile software development.

12 Azure Pipelines is the solution for Azure PaaS deployments. Azure Pipelines handles continuous deployment (CD) and release tasks, which consume the package versions exposed by GitHub Actions builds.

13 Azure Pipelines deployment tasks specifically support Azure PaaS resources, including web application deployments:

- With the continuous builds from Pipelines, a continuous release deploys the latest application build to a Development environment.
- The pipeline promotes the build selected for testing up to the next deployment ring, the Test environment.
- After the build passes all tests, it’s promoted and staged in the next outer ring for User Acceptance Testing (UAT).

14 When solution quality reaches a Production state, Azure Pipelines triggers a Production release. The pipeline limits Production deployments to the minimum number necessary.