Azure Repos is a source code version control service hosted by Microsoft Azure, offering Git repositories for managing and collaborating on code. It provides several features that facilitate team collaboration, version control, and integration with Azure DevOps services. Here’s an overview of Azure Repos:

**Key Features of Azure Repos:**

1. **Git and TFVC Support**:
   * Azure Repos supports both Git (distributed version control system) and Team Foundation Version Control (TFVC, centralized version control).
2. **Collaboration and Code Reviews**:
   * Teams can collaborate on code by branching, merging, and reviewing code changes.
   * Pull Requests facilitate code reviews and approval workflows.
3. **Integration with Azure DevOps Services**:
   * Azure Repos integrates seamlessly with Azure DevOps Services, enabling comprehensive DevOps capabilities such as CI/CD pipelines, agile planning, and project management.
4. **Security and Permissions**:
   * Granular access controls ensure that only authorized users have access to repositories and specific branches.
   * Secure connectivity using HTTPS or SSH protocols.
5. **Scalability and Performance**:
   * Azure Repos is designed for scalability, supporting large repositories and distributed teams.
   * It offers robust performance and reliability, backed by Azure’s infrastructure.
6. **Extensibility**:
   * Integrates with various development tools and services, including IDEs (Integrated Development Environments) like Visual Studio and Visual Studio Code.
   * Supports extensions and APIs for custom integrations and automation.

**Getting Started with Azure Repos:**

To start using Azure Repos, follow these steps:

1. **Create a Project**:
   * Navigate to Azure DevOps and create a new project. Azure Repos is part of Azure DevOps Services.
2. **Set up Repositories**:
   * Choose Git or TFVC as the version control system when setting up your project.
   * Create repositories to host your codebase.
3. **Clone Repositories**:
   * Clone repositories locally using Git commands (git clone) or directly from IDEs that support Azure Repos integration.
4. **Collaborate and Manage Code**:
   * Use branches for feature development (git branch and git checkout) and merge changes (git merge).
   * Conduct code reviews using Pull Requests (git pull-request), and manage code versions effectively.
5. **Integrate with CI/CD Pipelines**:
   * Set up CI/CD pipelines using Azure Pipelines to automate builds, testing, and deployments directly from Azure Repos.
6. **Monitor and Manage**:
   * Monitor repository activity, track changes, and manage permissions and security settings as needed.

**Azure Repos vs. GitHub:**

* **Azure Repos** is part of Azure DevOps Services, tightly integrated with Azure’s ecosystem, and offers robust DevOps capabilities.
* **GitHub**, owned by Microsoft, is a separate platform known for its social coding features, extensive community, and integration with third-party tools.

Both platforms support Git repositories and offer similar basic version control functionalities but differ in their additional features, integrations, and community aspects.

**Conclusion:**

Azure Repos provides a reliable and scalable platform for version control and collaborative software development, particularly suited for teams leveraging Azure DevOps services. Its integration with Azure services makes it a compelling choice for organizations using Azure’s cloud infrastructure.