Aspect	Deploying Directly Through Code	Deploying Using Docker File
Ease of Setup	Easier to set up for simple applications	More complex setup due to containerization requirements
Configuration	Limited to what Azure App Service supports natively	Highly customizable with Dockerfile instructions
Portability	Less portable; specific to Azure App Service	Highly portable; can be deployed anywhere Docker is supported
Isolation	Limited isolation; shares environment with other apps	Full isolation; each app runs in its own container
Dependency Management	Managed by Azure; may face compatibility issues	Complete control over dependencies and their versions
Scalability	Auto-scaling available; simpler to manage	More control over scaling; requires more management
Resource Utilization	Can be less efficient; shares resources with other apps	More efficient; dedicated resources per container
Security	Managed by Azure; limited control over security policies	More control over security policies and updates
Flexibility	Limited by Azure App Service's supported languages and frameworks	Highly flexible; can run any language or framework that Docker supports
Updates & Rollbacks	Easier to update code; rollback might be more complex	Easy to update container; simple rollback to previous image
Debugging	Easier with Azure's integrated tools	Requires container-specific debugging tools
Cost	Potentially lower cost for small, simple apps	Can be higher due to additional overhead of managing containers
Learning Curve	Lower; familiar to most developers	Higher; requires knowledge of Docker and containerization
Performance	Can be limited by the underlying App Service environment	Potentially better performance due to optimized containers