|  |
| --- |
| Code deployment |
| Update Frame work |
| Data base procedures Timeout |
| Disaster Recovery Plan |
| Bug traking |

Deploying a Dockerized ASP.NET Core 6 API into different cloud platforms involves several key considerations, such as the deployment process, framework updates, database procedure timeouts, disaster recovery plans, and bug tracking. Below is a summary of the best and easiest ways to deploy on popular cloud platforms, each with a brief example in a tabular format.

**1. Azure**

Azure provides comprehensive support for Dockerized applications, including Azure App Service and Azure Kubernetes Service (AKS).

| **Property** | **Description** | **Example** |
| --- | --- | --- |
| **Code Deployment** | Use Azure App Service for Containers or Azure Kubernetes Service for more control. | az webapp create --resource-group myResourceGroup --plan myAppServicePlan --name myApp --deployment-container-image-name mydockerimage |
| **Framework Update** | Update Docker image and push to Azure Container Registry (ACR). | docker build -t mydockerimage . <br> docker push myacr.azurecr.io/mydockerimage |
| **Database Timeout** | Configure connection strings and retry policies in Azure SQL Database. | az sql server firewall-rule create --resource-group myResourceGroup --server myServer --name AllowYourIp --start-ip-address <your\_ip> --end-ip-address <your\_ip> |
| **Disaster Recovery** | Use Azure Site Recovery and configure geo-redundant backups. | az backup vault create --resource-group myResourceGroup --name myBackupVault |
| **Bug Tracking** | Integrate with Azure DevOps for continuous integration and monitoring. | az pipelines create --name myPipeline --repository-url https://github.com/your-repo.git --branch main --yaml-path azure-pipelines.yml |

**2. AWS**

Amazon Web Services (AWS) offers multiple services like Elastic Beanstalk, ECS, and EKS for deploying Dockerized applications.

| **Property** | **Description** | **Example** |
| --- | --- | --- |
| **Code Deployment** | Use Elastic Beanstalk for simple setups or ECS/EKS for more complex applications. | eb init -p docker my-app <br> eb create my-app-env |
| **Framework Update** | Update Docker image and push to Amazon Elastic Container Registry (ECR). | docker build -t mydockerimage . <br> aws ecr create-repository --repository-name mydockerimage <br> docker push myaccountid.dkr.ecr.us-west-2.amazonaws.com/mydockerimage |
| **Database Timeout** | Configure RDS parameters and use connection pooling libraries. | aws rds create-db-instance --db-instance-identifier mydb --allocated-storage 20 --db-instance-class db.t2.micro --engine mysql --master-username myuser --master-user-password mypassword |
| **Disaster Recovery** | Use AWS Backup and enable Multi-AZ deployment for RDS. | aws backup create-backup-vault --backup-vault-name myBackupVault |
| **Bug Tracking** | Integrate with AWS CodePipeline and CloudWatch for CI/CD and monitoring. | aws codepipeline create-pipeline --cli-input-json file://pipeline.json |

**3. Google Cloud Platform (GCP)**

Google Cloud offers services like Google Kubernetes Engine (GKE) and Cloud Run for containerized applications.

| **Property** | **Description** | **Example** |
| --- | --- | --- |
| **Code Deployment** | Use Cloud Run for serverless deployment or GKE for managed Kubernetes clusters. | gcloud run deploy my-service --image gcr.io/my-project/mydockerimage |
| **Framework Update** | Update Docker image and push to Google Container Registry (GCR). | docker build -t gcr.io/my-project/mydockerimage . <br> docker push gcr.io/my-project/mydockerimage |
| **Database Timeout** | Configure Cloud SQL with appropriate settings and retries. | gcloud sql instances create myinstance --database-version=POSTGRES\_9\_6 --cpu=1 --memory=4GB |
| **Disaster Recovery** | Use Google Cloud Backup and configure regional backups for Cloud SQL. | gcloud sql backups create --instance=myinstance |
| **Bug Tracking** | Integrate with Google Cloud Build and Stackdriver for continuous integration and monitoring. | gcloud builds submit --tag gcr.io/my-project/mydockerimage |

**Summary Table**

| **Cloud Provider** | **Service for Deployment** | **Docker Image Registry** | **CI/CD Integration** | **Database Solution** | **Disaster Recovery** | **Bug Tracking** |
| --- | --- | --- | --- | --- | --- | --- |
| **Azure** | Azure App Service / AKS | Azure Container Registry (ACR) | Azure DevOps Pipelines | Azure SQL Database | Azure Site Recovery | Azure DevOps |
| **AWS** | Elastic Beanstalk / ECS/EKS | Amazon Elastic Container Registry (ECR) | AWS CodePipeline | Amazon RDS | AWS Backup | AWS CodePipeline + CloudWatch |
| **GCP** | Cloud Run / GKE | Google Container Registry (GCR) | Google Cloud Build | Google Cloud SQL | Google Cloud Backup | Stackdriver Monitoring |