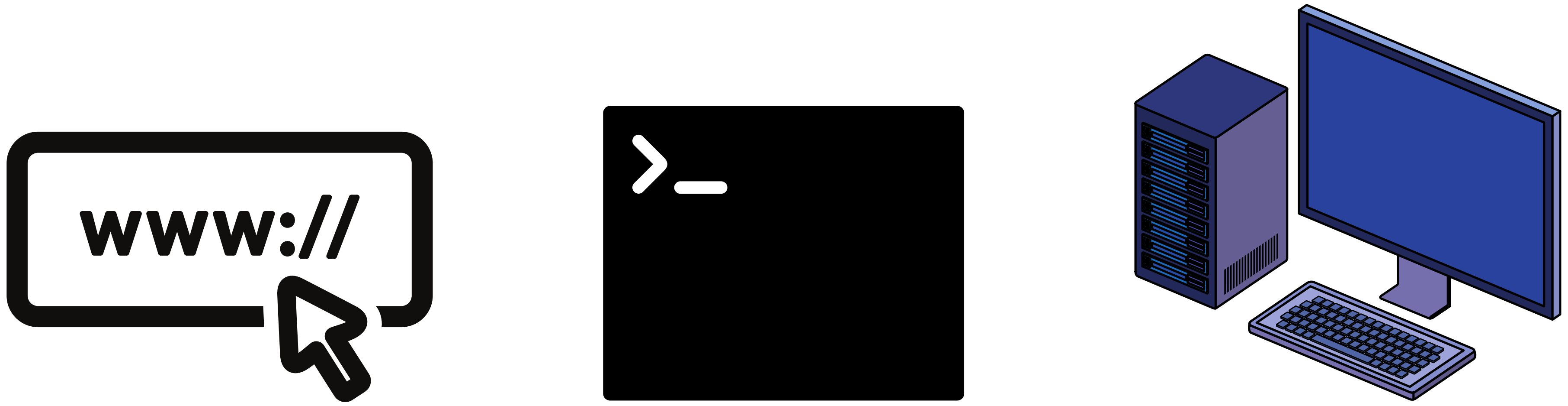
Deploying .NET Applications



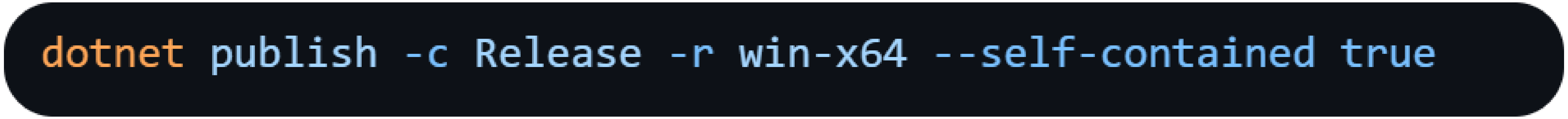
SAHIL MAKVANA - 8023058064 (ROLL NO:12)

Introduction

- Deployment refers to distributing a .NET application so users can run it on their machines or servers.
- .NET applicationscan be deployed in different ways depending on the type:
- Console Applications (Command-line apps)
- Web Applications (ASP.NET apps, APIs)
- Desktop Applications (WinForms, WPF, MAUI)
- Choosing the right deployment method depends on platform, dependencies, and target audience.

Deploying .NET Console Applications

- Console apps are lightweight and don’t need a GUI.
- Can be deployed as:
- Self-Contained Deployment (SCD) – Includes .NET runtime
- Framework-Dependent Deployment (FDD) – Requires .NET installed on the machine
- Steps to Publish a Console App

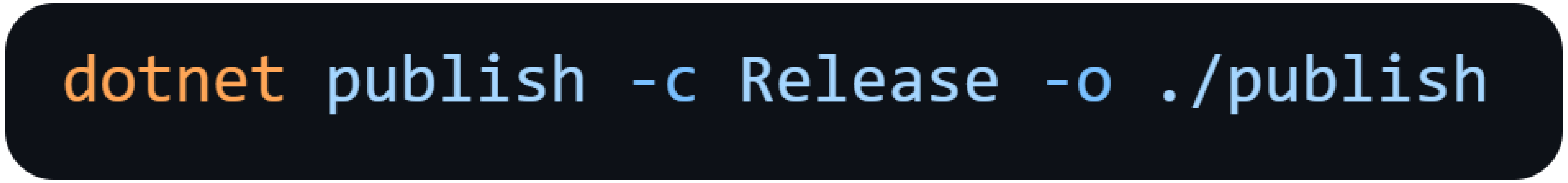


```
dotnet publish -c Release -r win-x64 --self-contained true
```

Deployment Methods:
✅ **Copying Executables – Simple for small apps**
✅ **Creating an Installer – Makes it easy for end users**

Deploying .NET Web Applications

- Web apps built with ASP.NET Core can be deployed on:
- IIS (Windows Server) – Used for enterprise apps
- Azure Web App – Cloud-based hosting
- Docker Containers – For scalability
- Linux with Nginx/Apache
- Self-Hosting (Kestrel) – Directly run via dotnet run
- Steps to Publish a Web App:

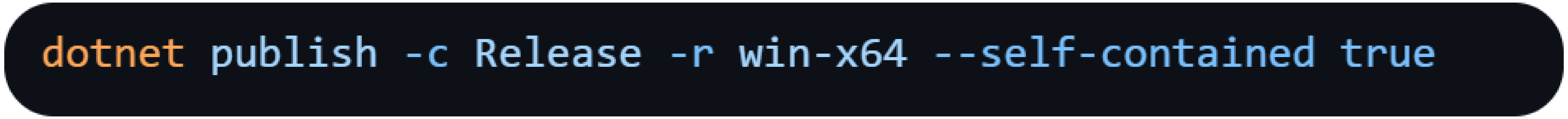


```
dotnet publish -c Release -o ./publish
```

✅ **Environment Variables for configurations**
✅ **CI/CD Pipelines for automated deployment**

Deploying .NET Desktop Applications

- Desktop apps are typically WinForms, WPF, or MAUI apps.
- Can be deployed using:
- MSIX Packaging – Modern Windows installer
- ClickOnce Deployment – Simple auto-updating installer
- Standalone Executables – Publish as a single .exe file
- Steps to Publish a WinForms/WPF App:



```
dotnet publish -c Release -r win-x64 --self-contained true
```

✅ **Dependency Bundling – Ensure required libraries are included**
✅ **Code Signing – Prevents security warnings**

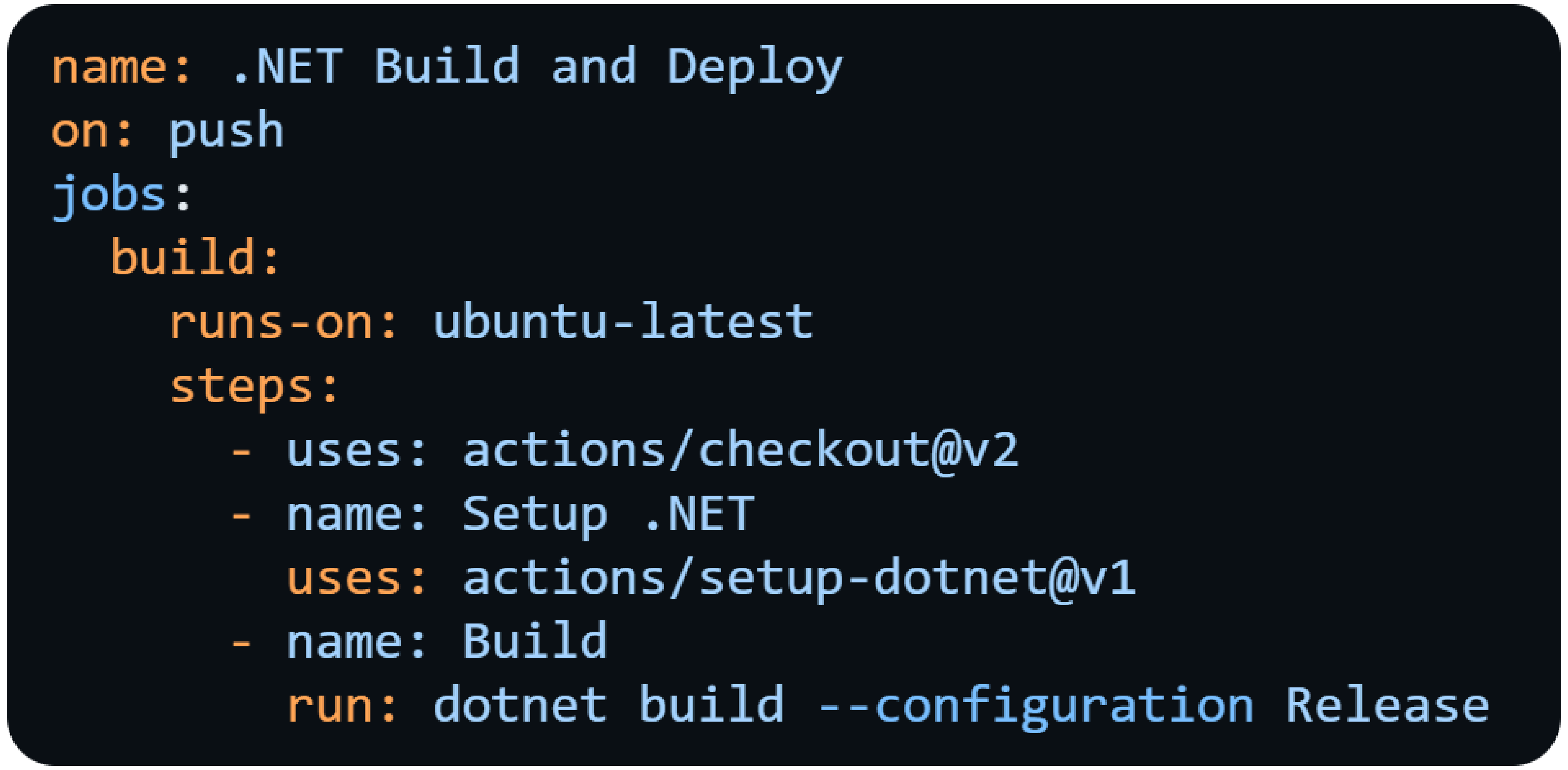
Self-Contained vs Framework-Dependent Deployment

Deployment Type	Pros	Cons
Self-Contained	No need for .NET runtime on the machine	Larger file size
Framework-Dependent	Smaller size	Requires .NET installed

- Use Self-Contained for standalone apps.
- Use Framework-Dependent for enterprise environments.

Deployment Automation (CI/CD)

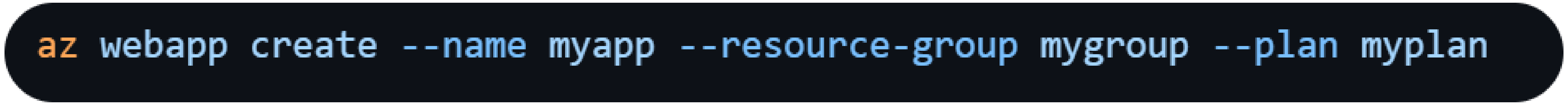
- CI/CD automates deployment using tools like:
- GitHub Actions
- Azure DevOps Pipelines
- Jenkins
- Example GitHub Action for .NET Deployment:



```
name: .NET Build and Deploy
on: push
jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
      - name: Setup .NET
        uses: actions/setup-dotnet@v1
      - name: Build
        run: dotnet build --configuration Release
```

Cloud Deployment (Azure, AWS, Docker)

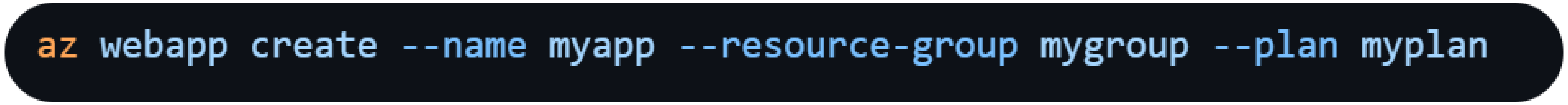
- ure App Service – For web apps
- AWS Elastic Beanstalk – Auto-scales web apps
- Docker Containers – Deploy cross-platform apps
- Deploying to Azure Example:



```
az webapp create --name myapp --resource-group mygroup --plan myplan
```

Security Considerations

- HTTPS & SSL Certificates – Encrypt web traffic
- Environment Variables – Never store secrets in code
- App Signing – Prevents tampering with executables
- Regular Updates – Patch security vulnerabilities



```
az webapp create --name myapp --resource-group mygroup --plan myplan
```

Conclusion

- Console Apps – Deploy as .exe or via script
- Web Apps – Host on IIS, Azure, or Containers
- Desktop Apps – Use MSIX, ClickOnce, or standalone .exe
- CI/CD Pipelines – Automate deployments
- Security & Cloud Considerations

THANK
YOU!