kenze

Experts in .NET

Continuous integration and deployment with Azure Devops

kenze
Experts in .NET

Who is Geert?

- Kenze:
  - Focus on Microsoft technologies
  - Consultancy
  - In-house projects
- Dba, software engineer, team lead, project lead, senior developer, technical architect…
- Contact: geert.schreyers@kenze.be

kenze
Experts in .NET

Agenda

- What is Azure Devops?
- Example project
- Continuous Integration using builds
- Continuous Deployment using releases
- Q & A

kenze
Experts in .NET

What is Azure Devops?

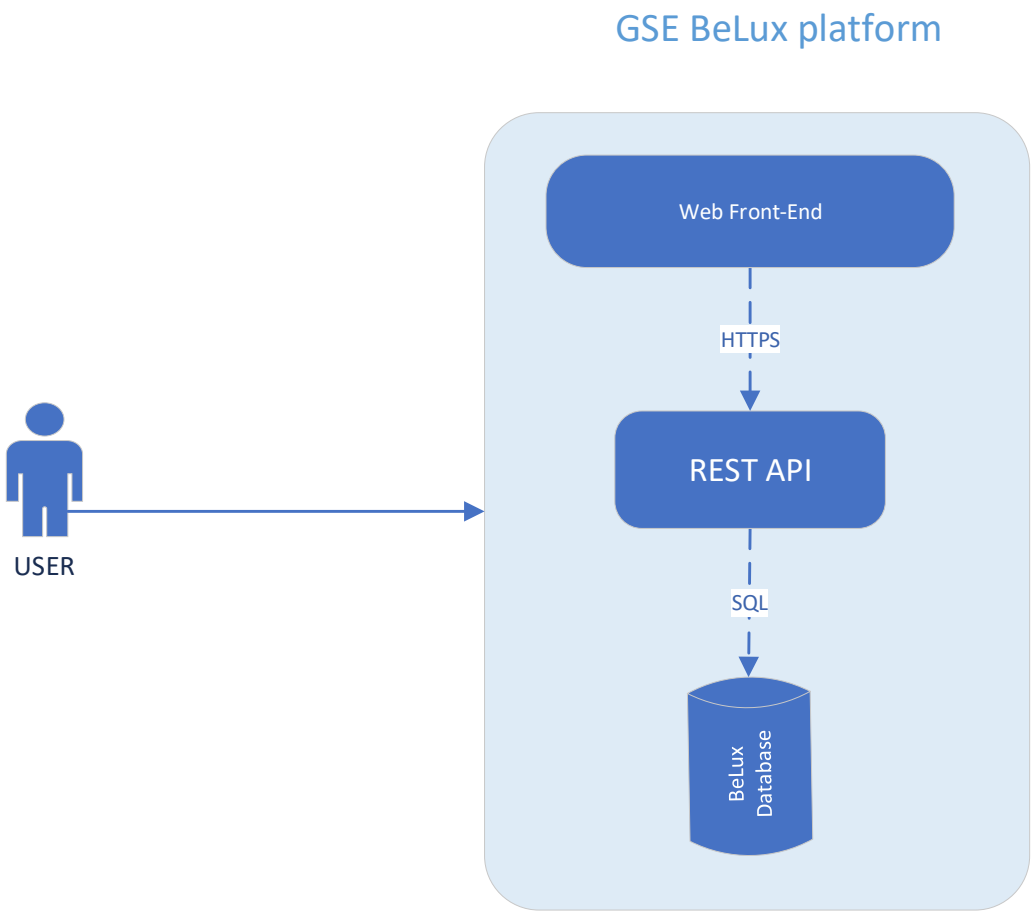
- Cloud version of TFS (Team Foundation Services)
  - Previously visualstudio.com
- “Plan smarter, collaborate better, and ship faster with a set of modern dev services”
- Services:
  - Support agile methodologies: Scrum / Kanban
  - Source Control: git (tfs)
  - Build & Release pipelines
  - Testing
  - Artifacts: npm, nuget, maven

kenze
Experts in .NET

What is Azure Devops?

- Does not only integrate with Microsoft technologies/environments:
  - On premise deployment
  - Service hooks
  - Service connections
  - Market place
- Uses agents for builds and releases
- Evolves quickly

Example project



kenze
Experts in .NET

Example project

Microsoft Azure

Acceptance

gsebeluxconferencewebacceptance.azurewebsites.net

gsebeluxconferenceapiacceptance.azurewebsites.net

BeLuxConference
gsebeluxconferenceaccepttans.database.windows.net

Production

gsebeluxconferenceweb.azurewebsites.net

gsebeluxconferenceapi.azurewebsites.net

BeLuxConference
gsebeluxconference.database.windows.net

kenze
Experts in .NET

Continuous integration using builds

- Some definitions
  - **Continuous Integration** (CI) is the practice of frequently integrating code into a shared repository so that every time a team member commits changes, those changes are verified and tested by an automated build, which allows teams to detect problems early.
  - **Continuous Integration** (CI) is a development practice that requires developers to integrate code into a shared repository several times a day. Each check-in is then verified by an automated build, allowing teams to detect problems early.
  - **Continuous integration** (CI) is the practice of routinely integrating code changes into the main branch of a repository, and testing the changes, as early and often as possible.

Continuous integration using builds

- Several source control systems possible
- Runs on an Agent
- Classic version vs yaml
- Example:
  - Build for front-end
  - Build for api/database

kenze
Experts in .NET

Continuous integration using builds

- Use NuGet 4.9.1
  NuGet Tool Installer
- NuGet restore
  NuGet
- dotnet restore
  .NET Core
- Prepare analysis on SonarCloud
  Prepare Analysis Configuration
- Build Api
  MSBuild
- Build GATool database
  MSBuild
- Replace settings for GATool.Logic.Tests
  Copy Files
- Build GATool.Logic.Tests
  MSBuild
- Deploy GATool database - Release.sql
  Azure SQL Database Deployment
- Deploy GATool database - DACPAC
  Azure SQL Database Deployment
- Run tests
  Visual Studio Test
- Publish Artifact Api
  Publish Build Artifacts
- Publish Artifact Database
  Publish Build Artifacts
- Run Code Analysis
  Run Code Analysis
- Publish Quality Gate Result
  Publish Quality Gate Result

kenze
Experts in .NET

Continuous deployment using releases

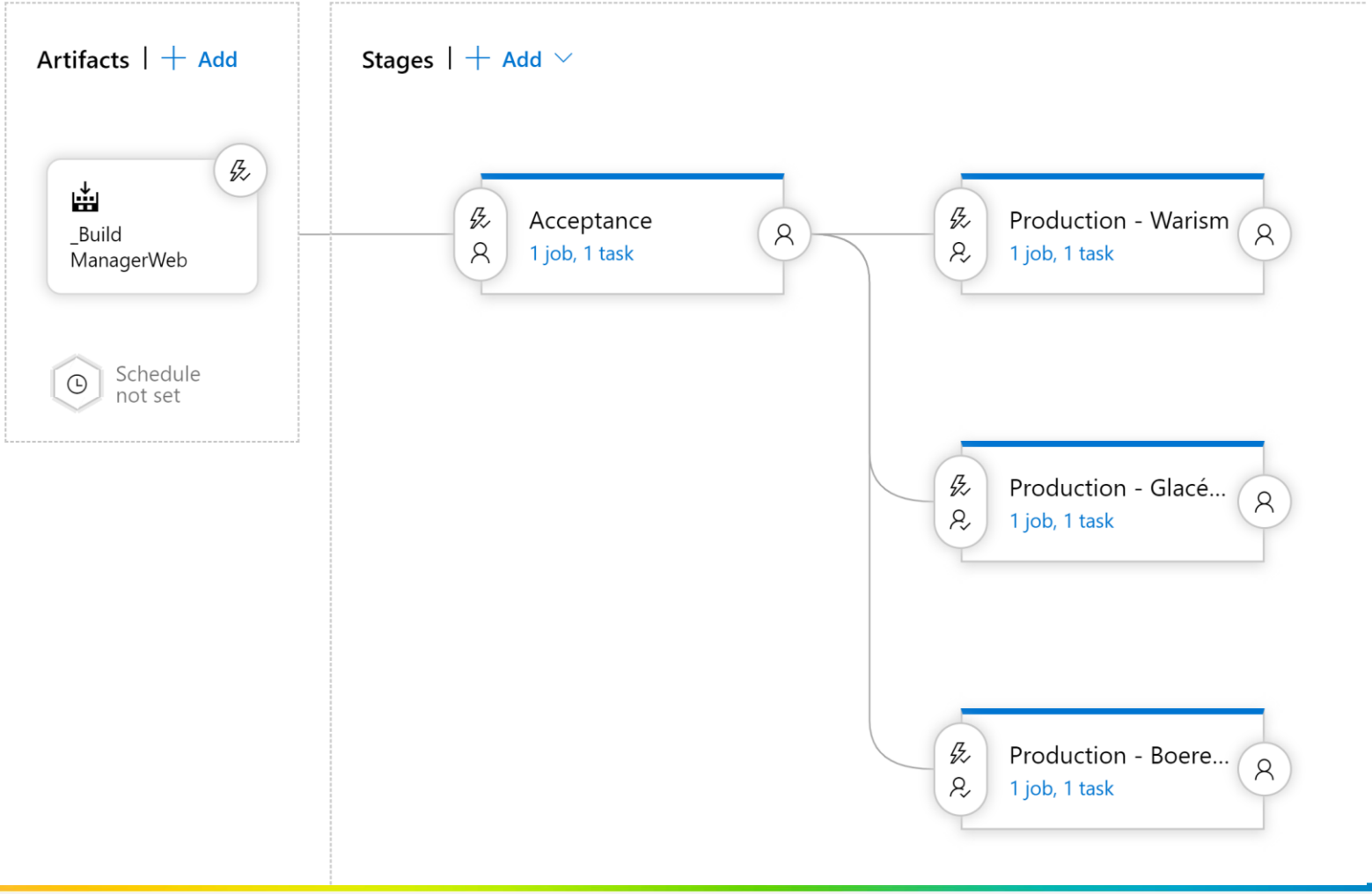
- Some definitions
  - **Continuous Deployment** (CD) is a software engineering approach in which teams produce software in short cycles, ensuring that the software can be reliably released at any time.
  - **Continuous Deployment** (CD) is the ability to get changes of all types—including new features, configuration changes, bug fixes and experiments—into production, or into the hands of users, *safely* and *quickly* in a *sustainable* way.
- Requires CI

Continuous deployment using releases

- Runs on an agent
- Example:
  - Release for front-end
  - Release for api / database

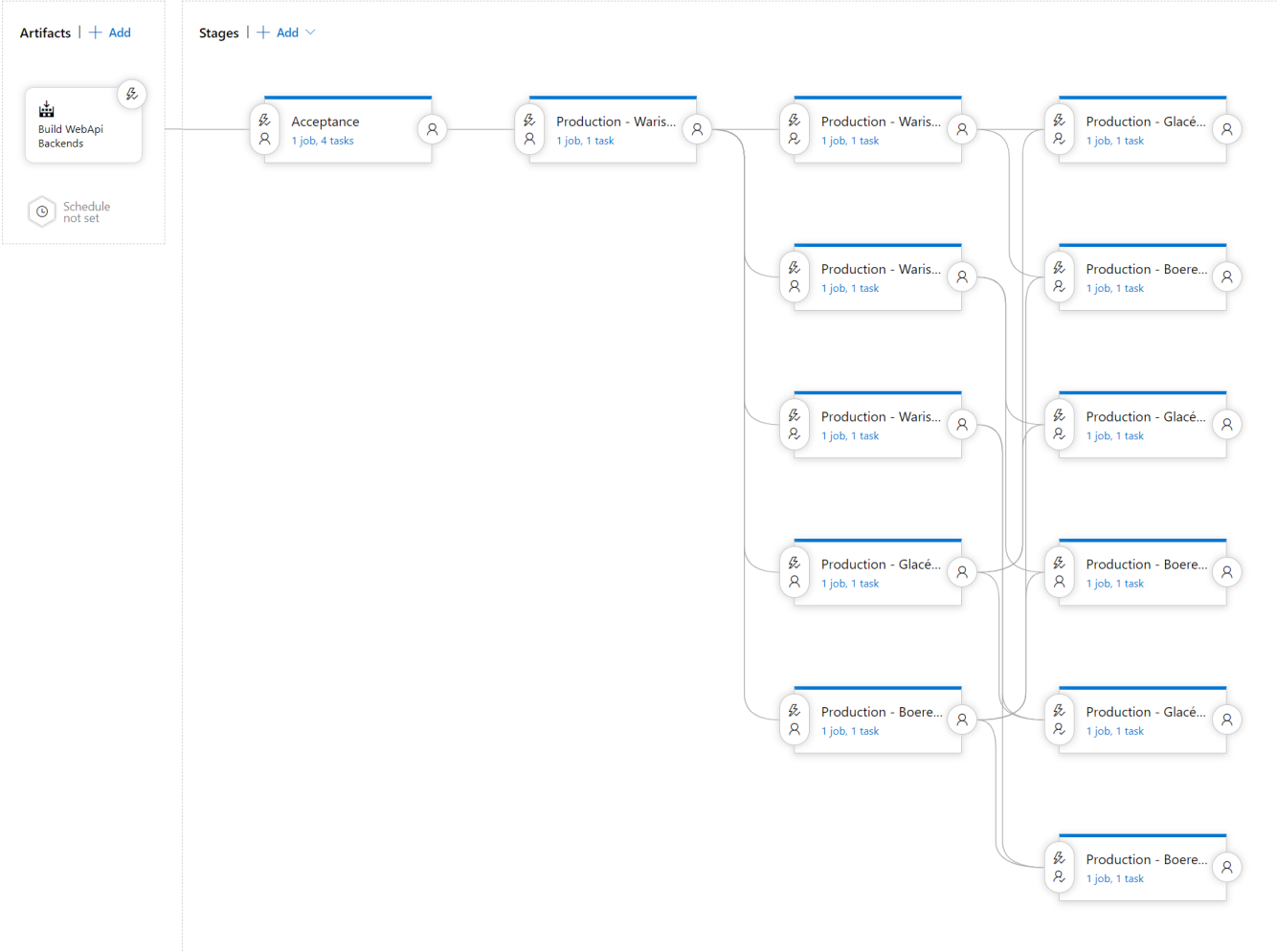
kenze
Experts in .NET

Continuous deployment using releases



kenze
Experts in .NET

Continuous deployment using releases



Questions?