Continuous Delivery with Azure Web Apps

Speaker

Vidar Kongsli

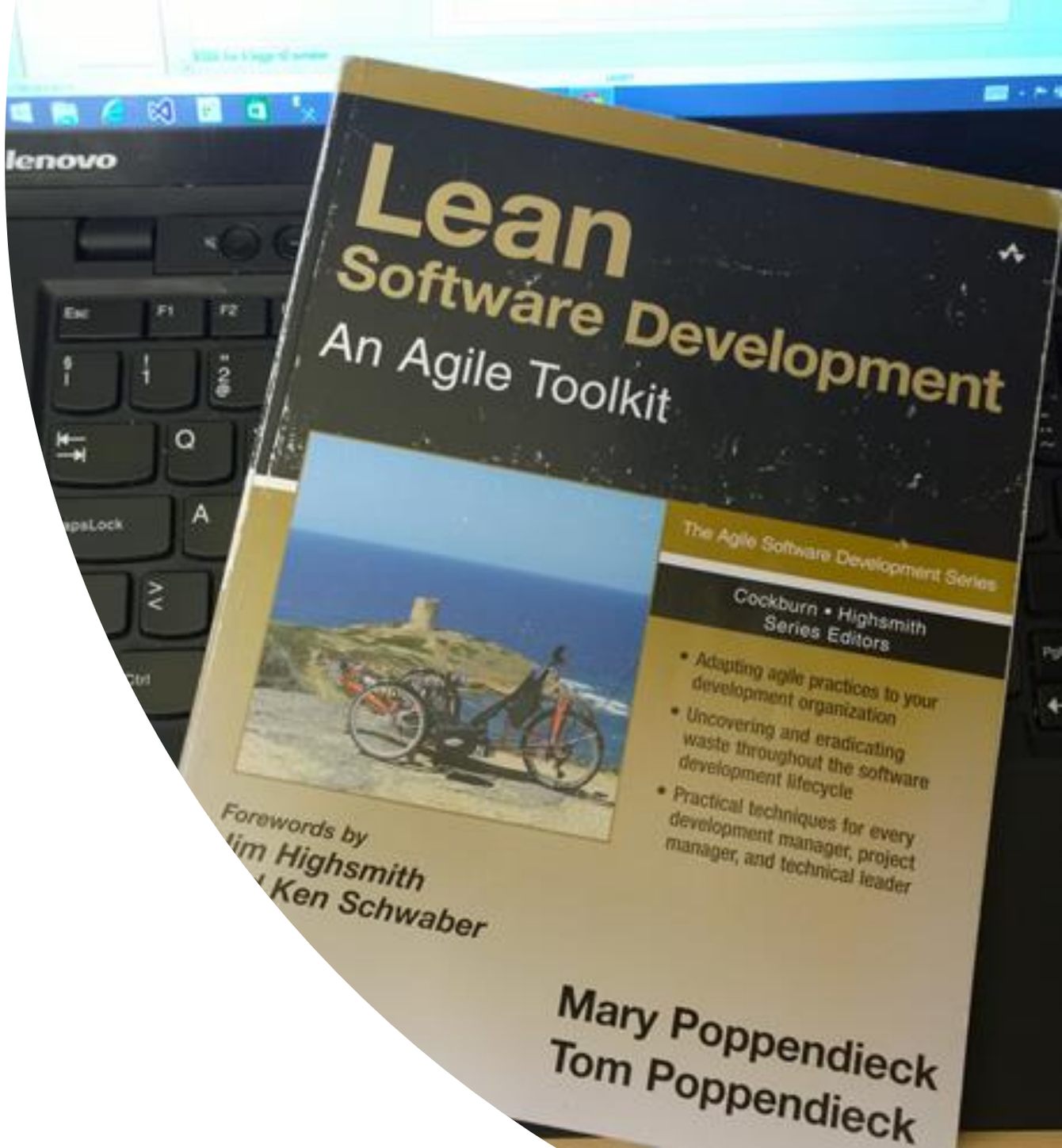
Wroc#

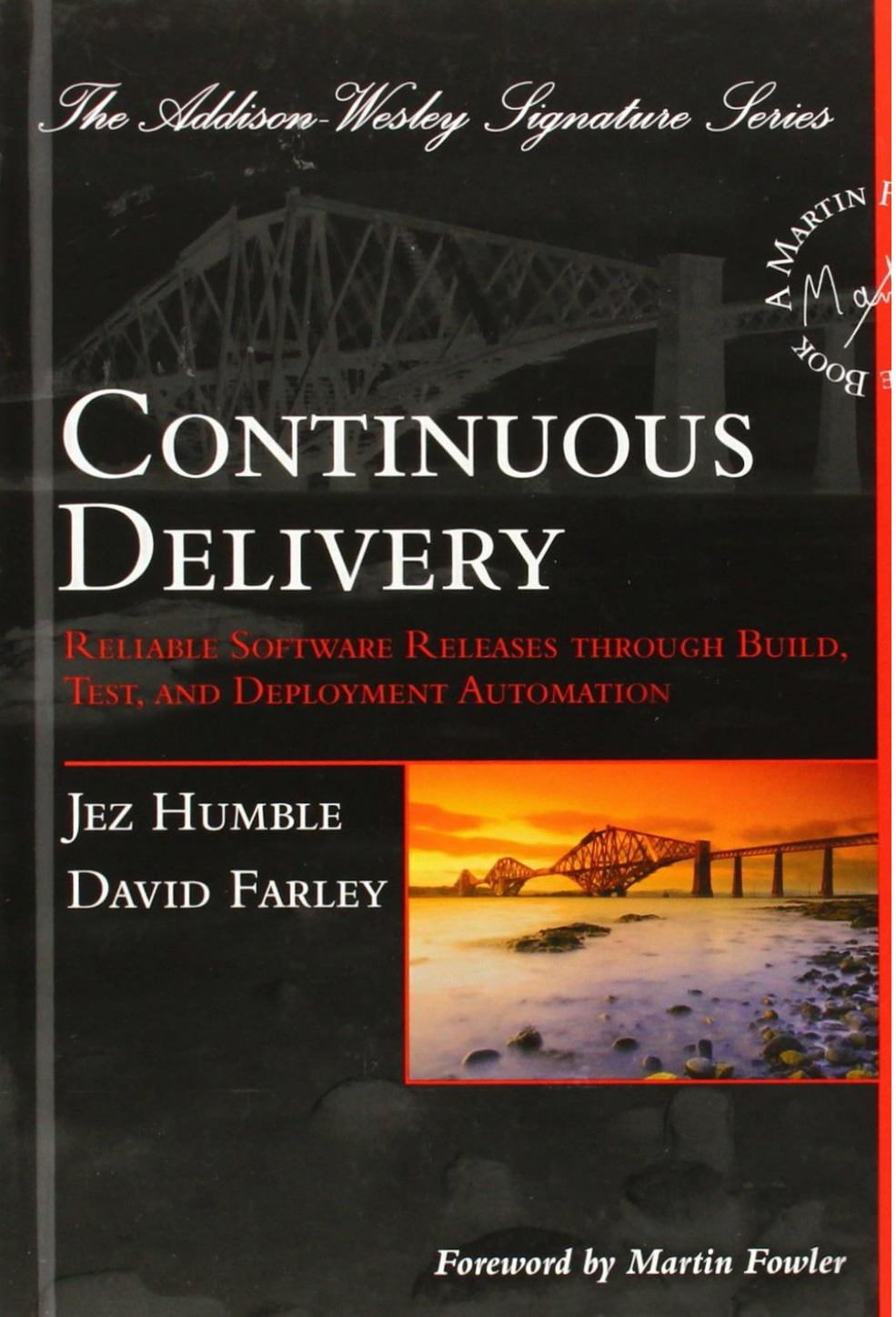
2019-03-29

bredvid

LEAN SOFTWARE DEVELOPMENT

- DELIVER AS FAST AS POSSIBLE
  - *"Rapid delivery allows some customers to delay decisions, and for others rapid delivery means quicker gratification. For customers of software development, rapid delivery often translates to increased business flexibility"*
- BUILD INTEGRITY IN
  - *"The way to build a system with high [...] integrity is to have excellent information flows both from customer to development team and between the upstream and the downstream processes of the development team"*



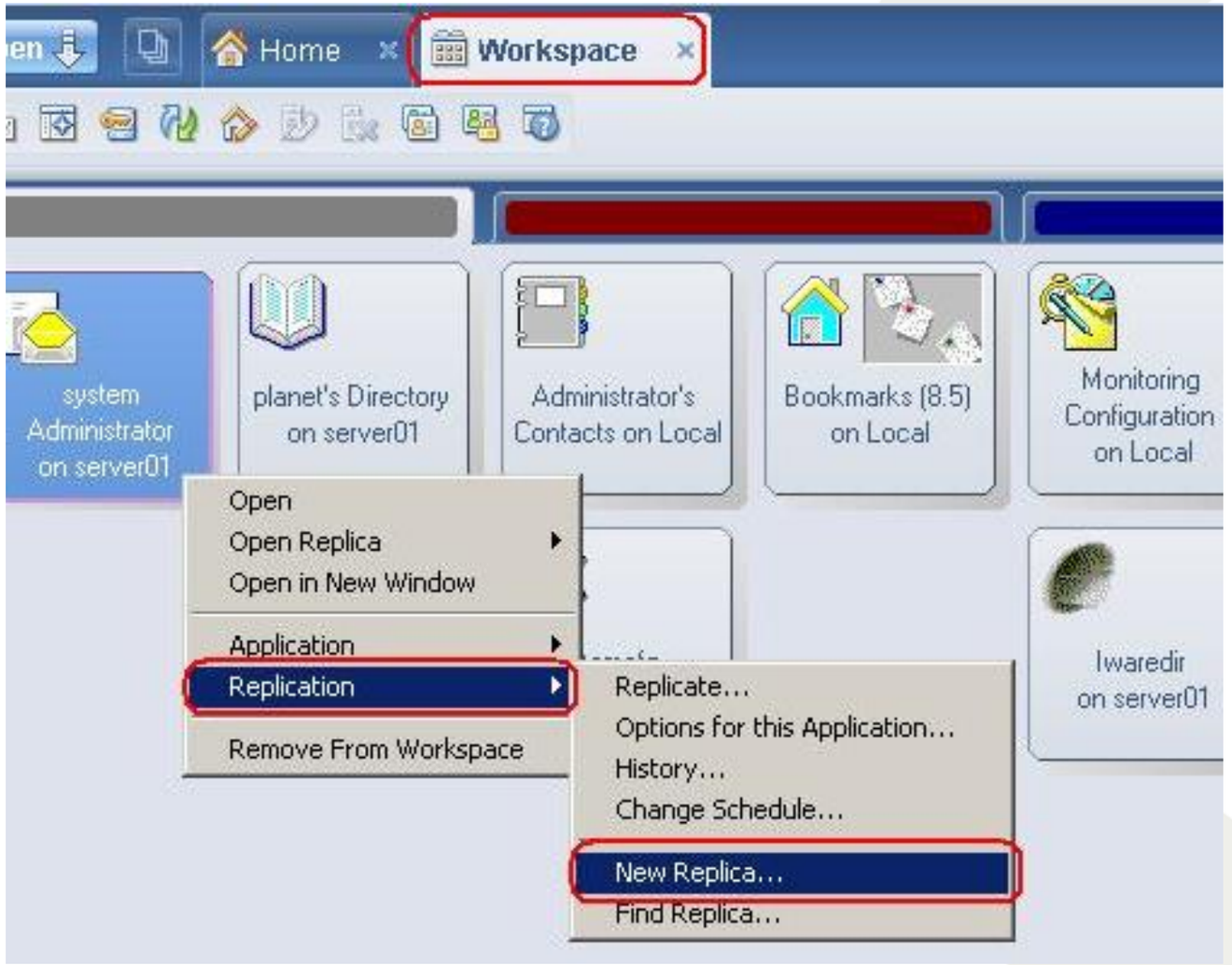


PRINCIPLES OF SOFTWARE DELIVERY

1. Create a repeatable, reliable process for releasing software
2. Automate almost everything
3. Keep everything in version control
4. If it hurts, do it more frequently, and bring the pain forward
5. Build quality in
6. Done means released
7. Everybody is responsible for the delivery process
8. Continuous improvement

DEPLOYMENT IN THE NINETIES

- Lotus Notes
- Deployment was
  - Built in
  - Quick
  - Reliable
  - Repeatable

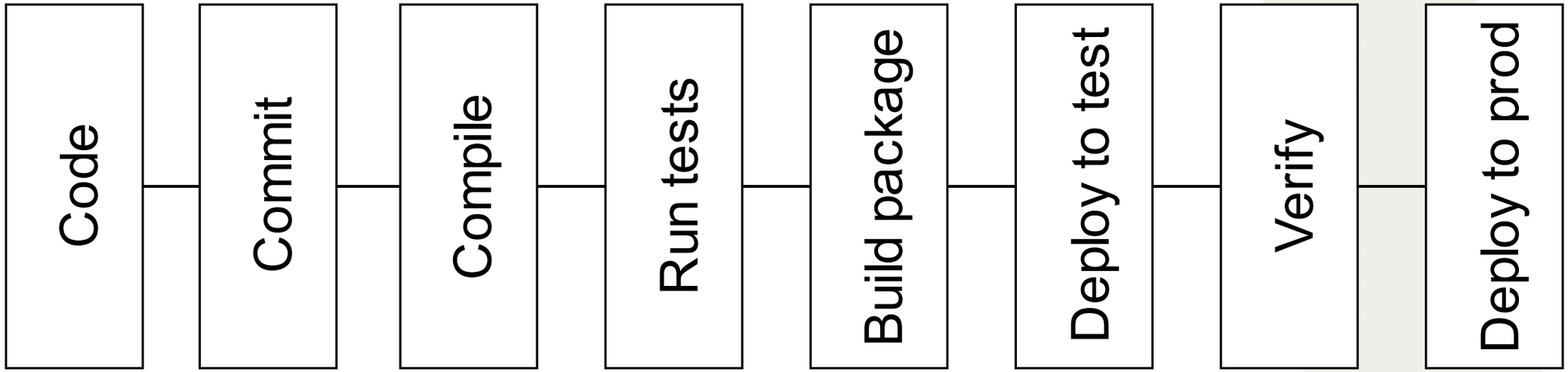


By Magnus Fröderberg/norden.org, CC BY 2.5 dk, https://commons.wikimedia.org/w/index.php?curid=25015804

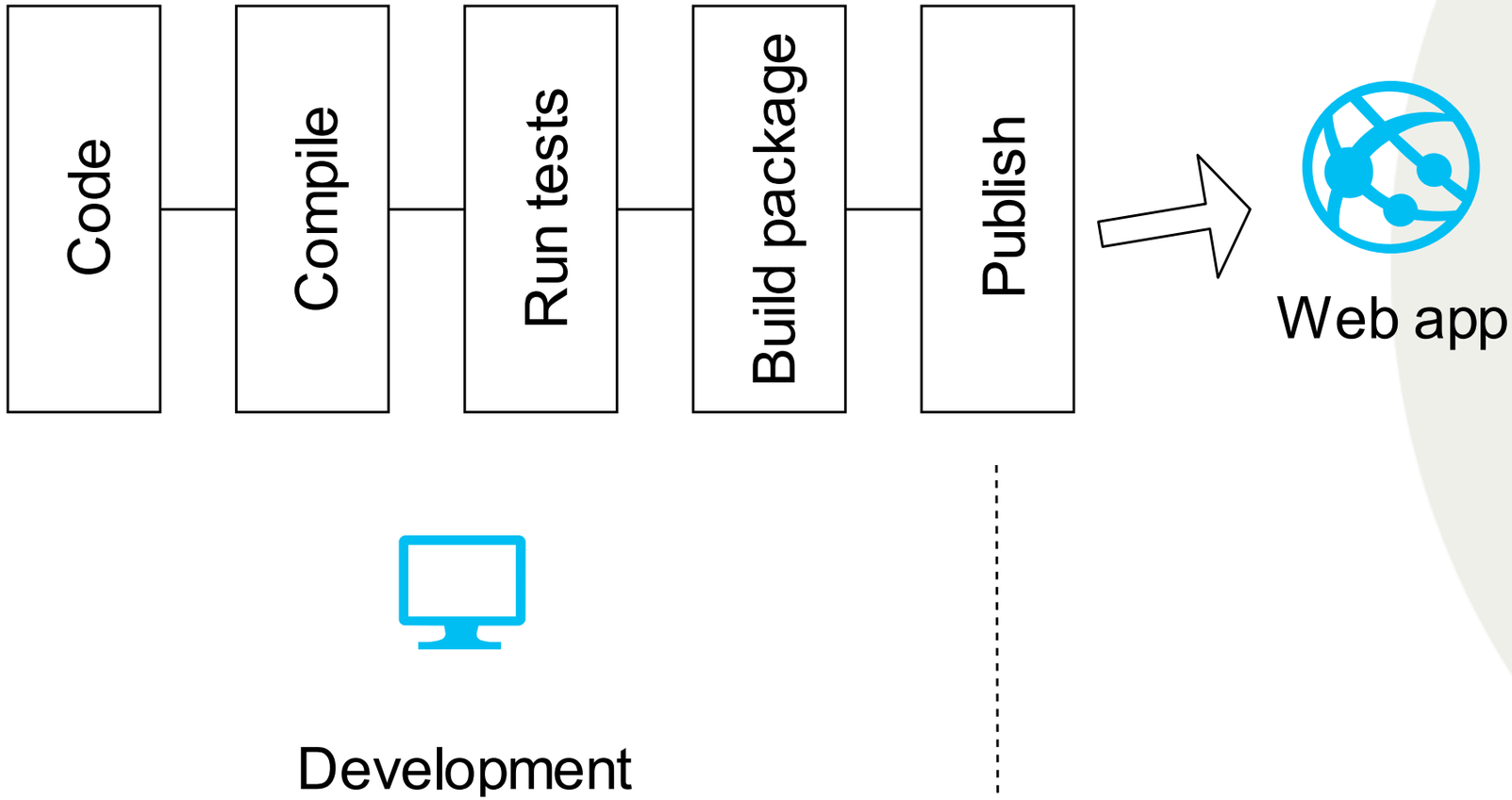
DEPLOYING TO AZURE WEB APPS

- Visual Studio
- Via build servers / infrastructure
  - Azure DevOps Pipelines
  - Third party solutions
    - Team City + Octopus deploy
    - Jenkins
- Version control deployment

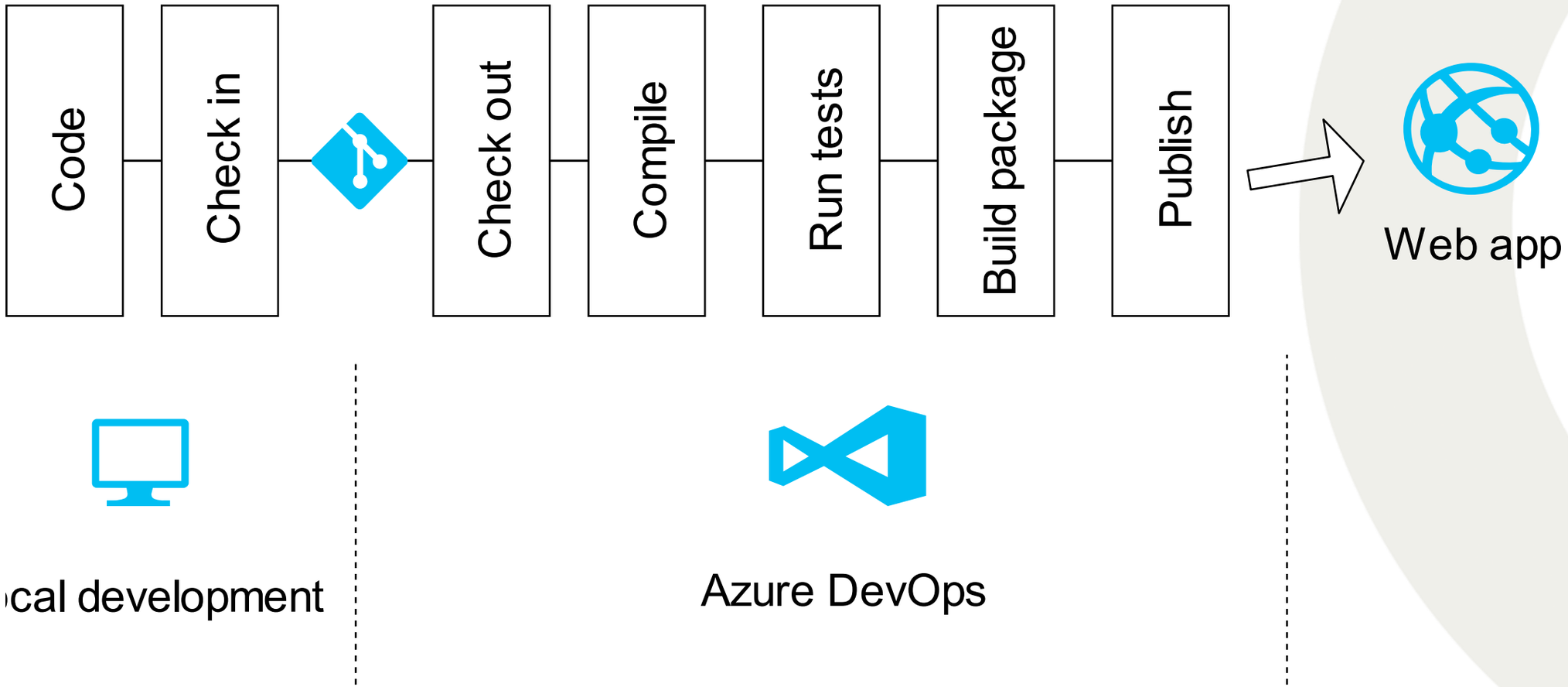
Generic build/deployment pipeline



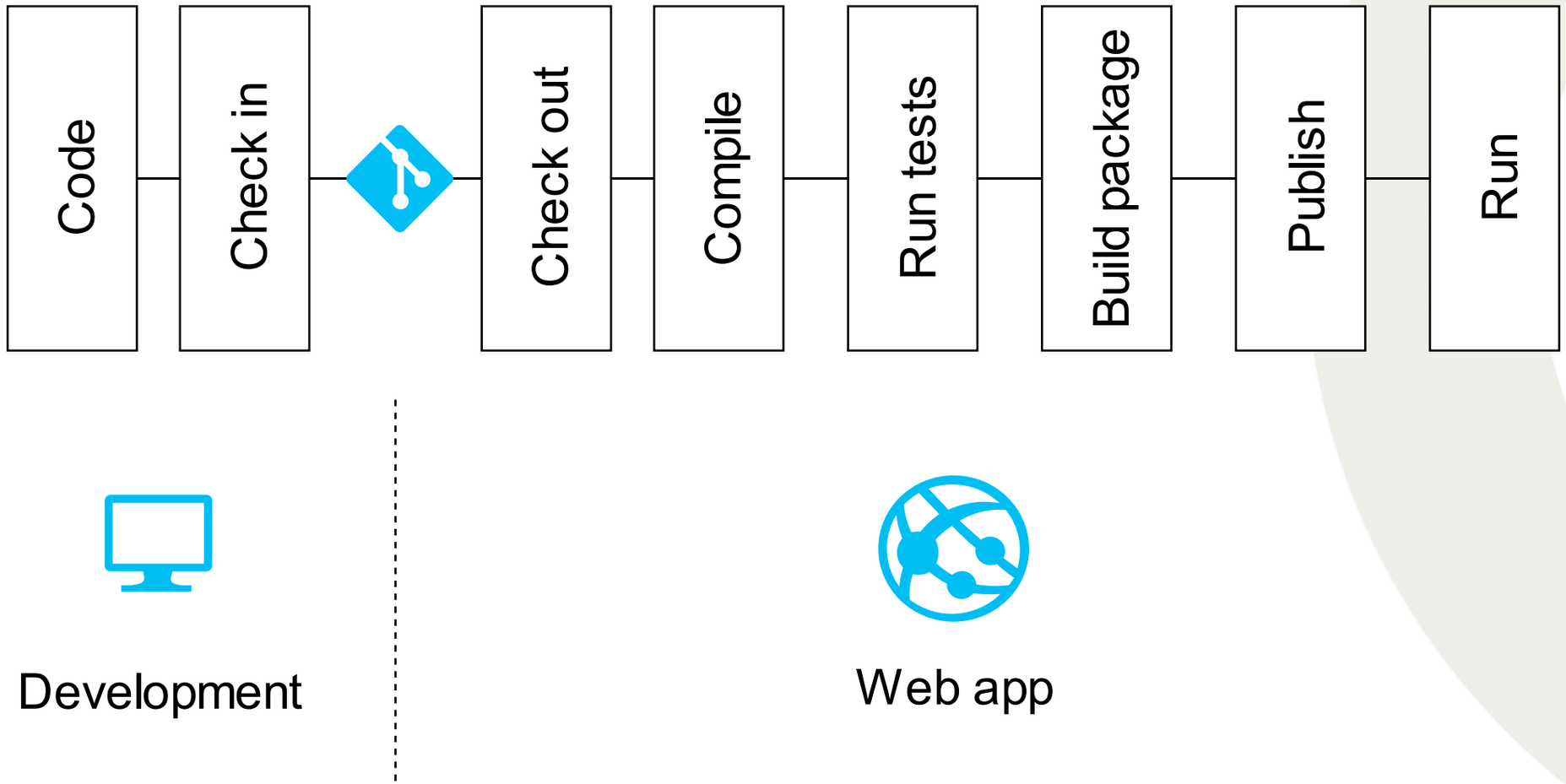
Visual Studio deployment



Azure DevOps Pipelines



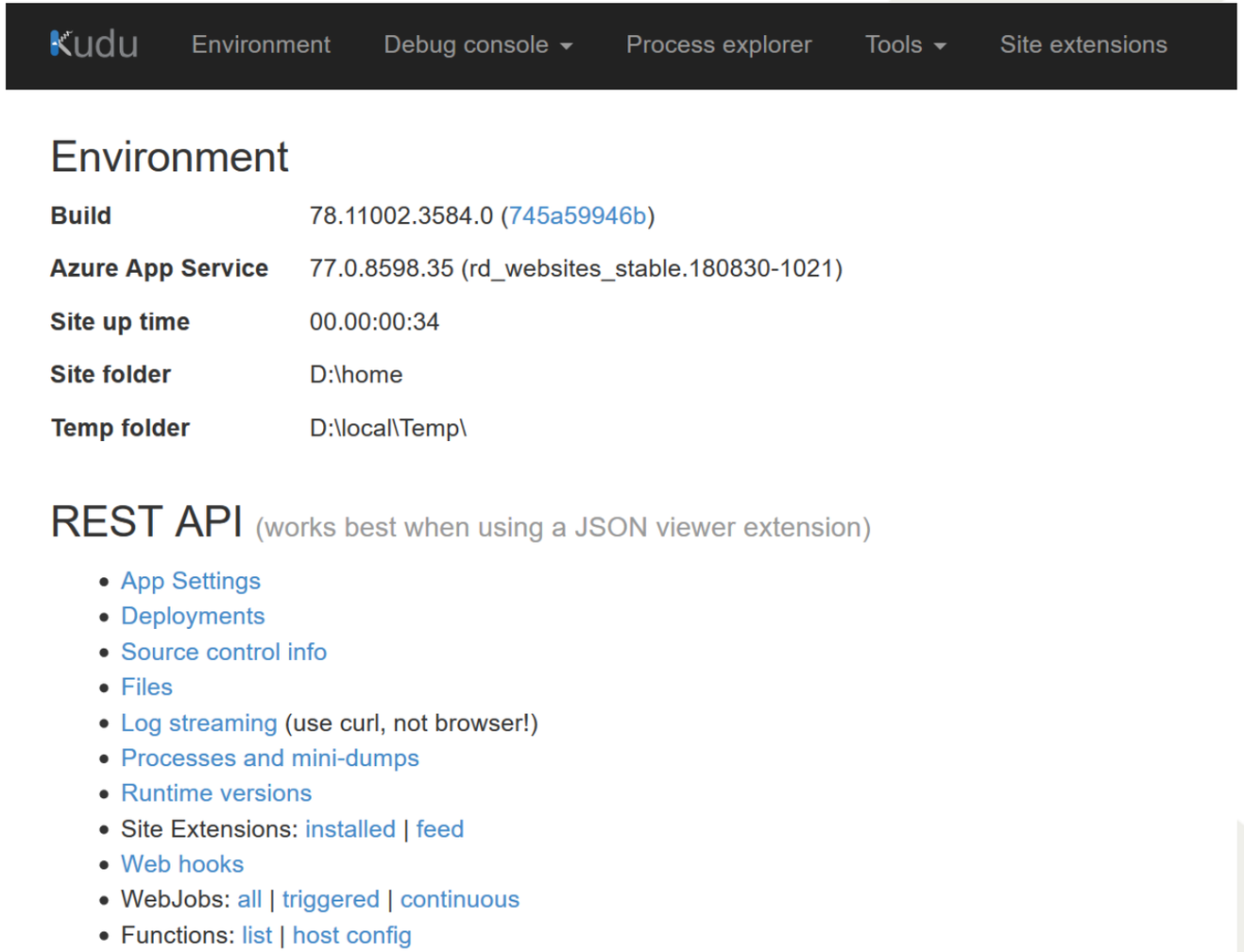
Version control deployment



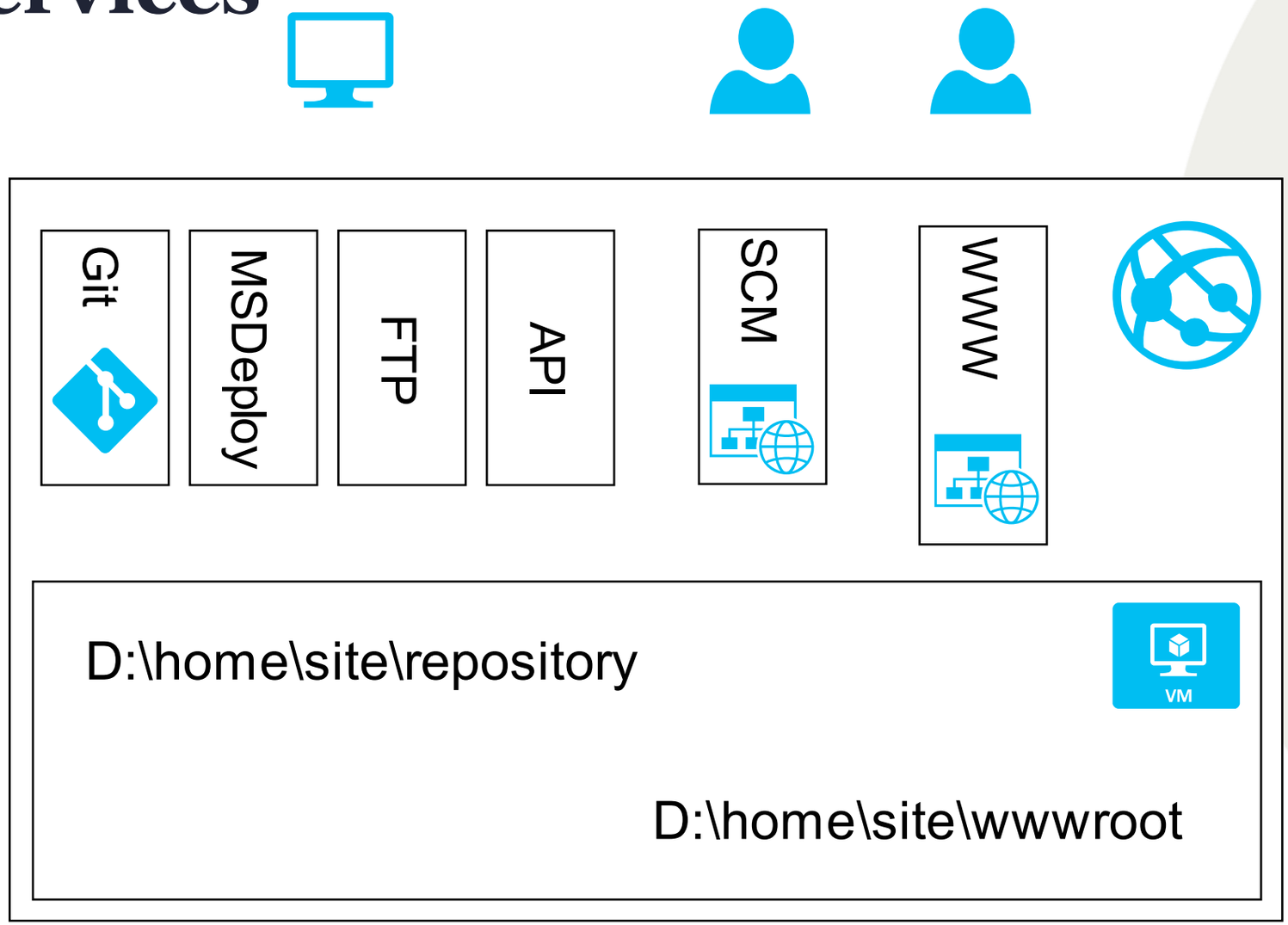
By Stefan Schäfer, Lich - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=32291896

Kudu

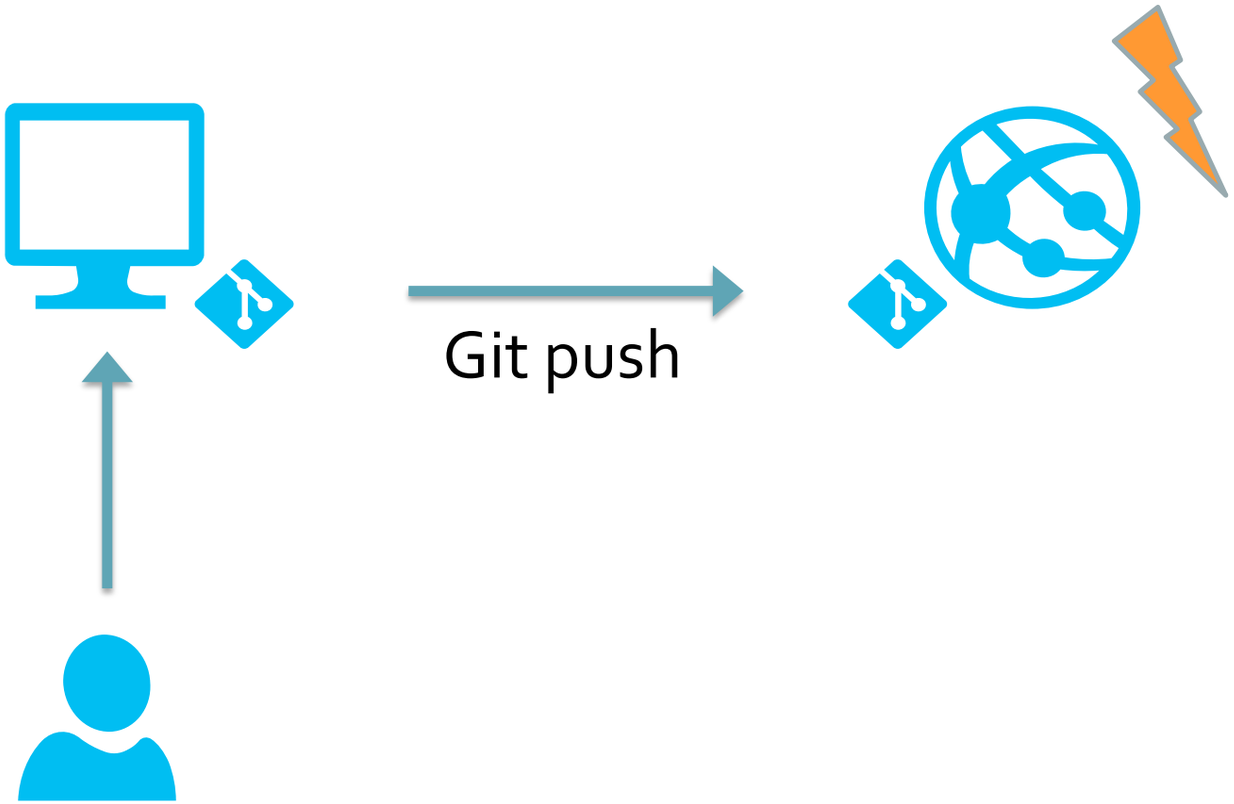
- Available on all Azure Service Plan SKUs
- https://<app-service-name>.**scm**.azurewebsites.net
- Open source at https://github.com/projectkudu/kudu



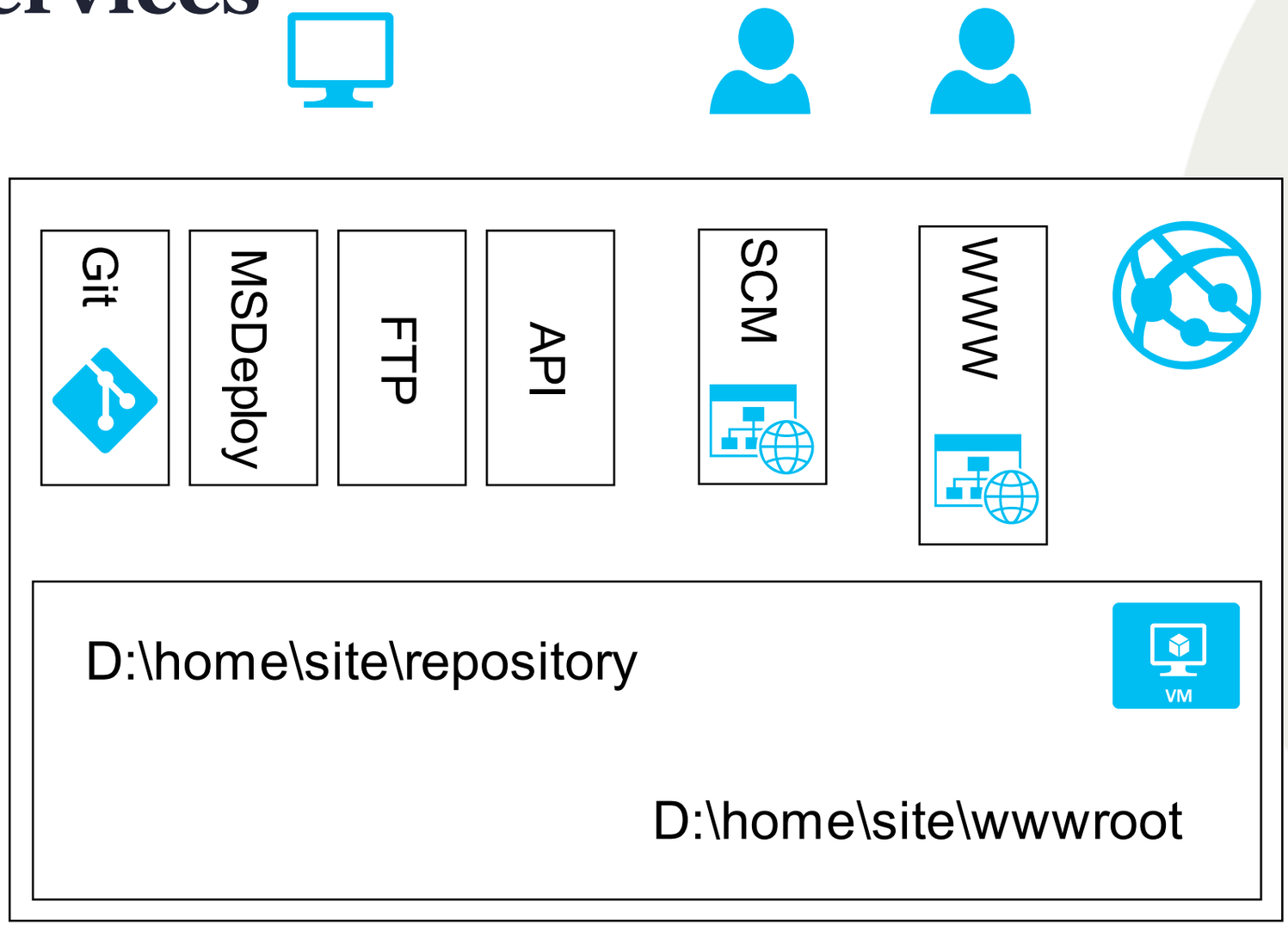
Kudu services



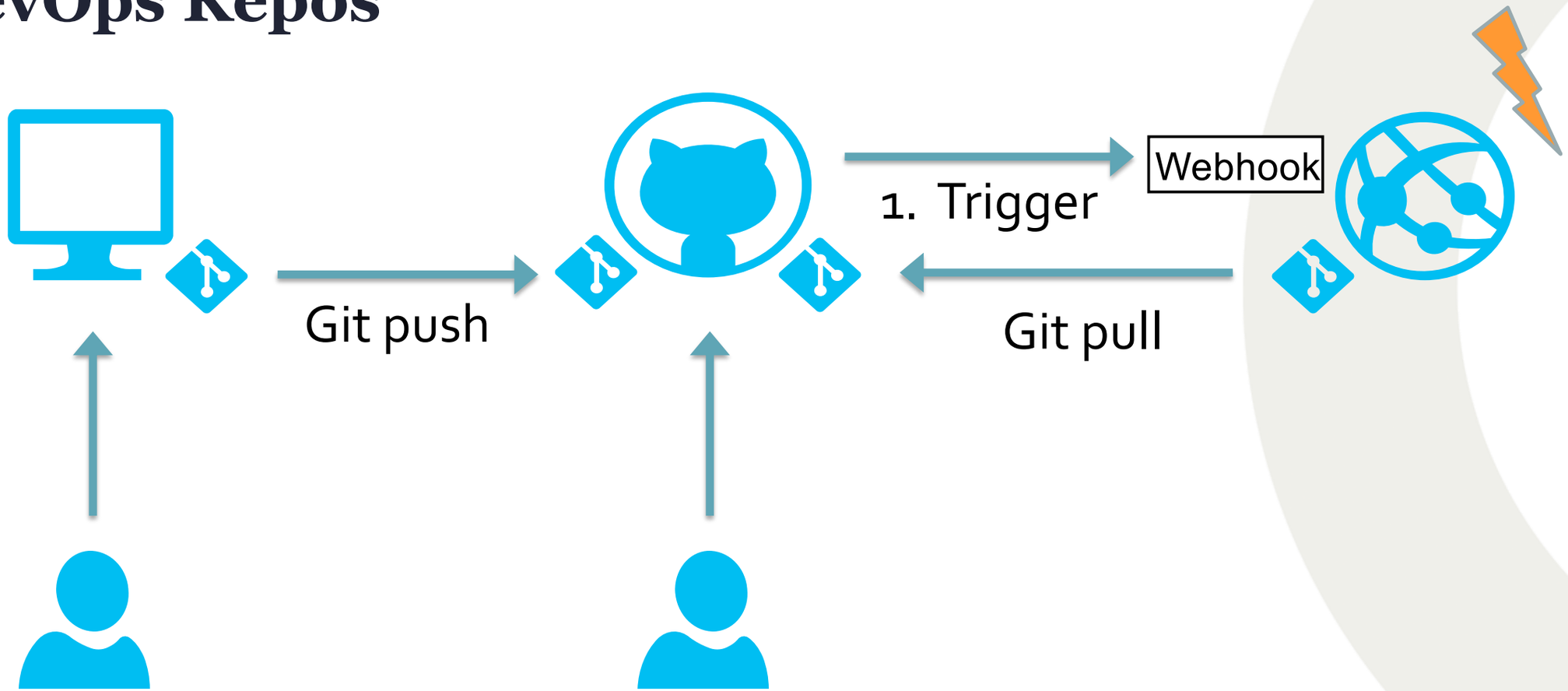
Source: Local git deployment



Kudu services

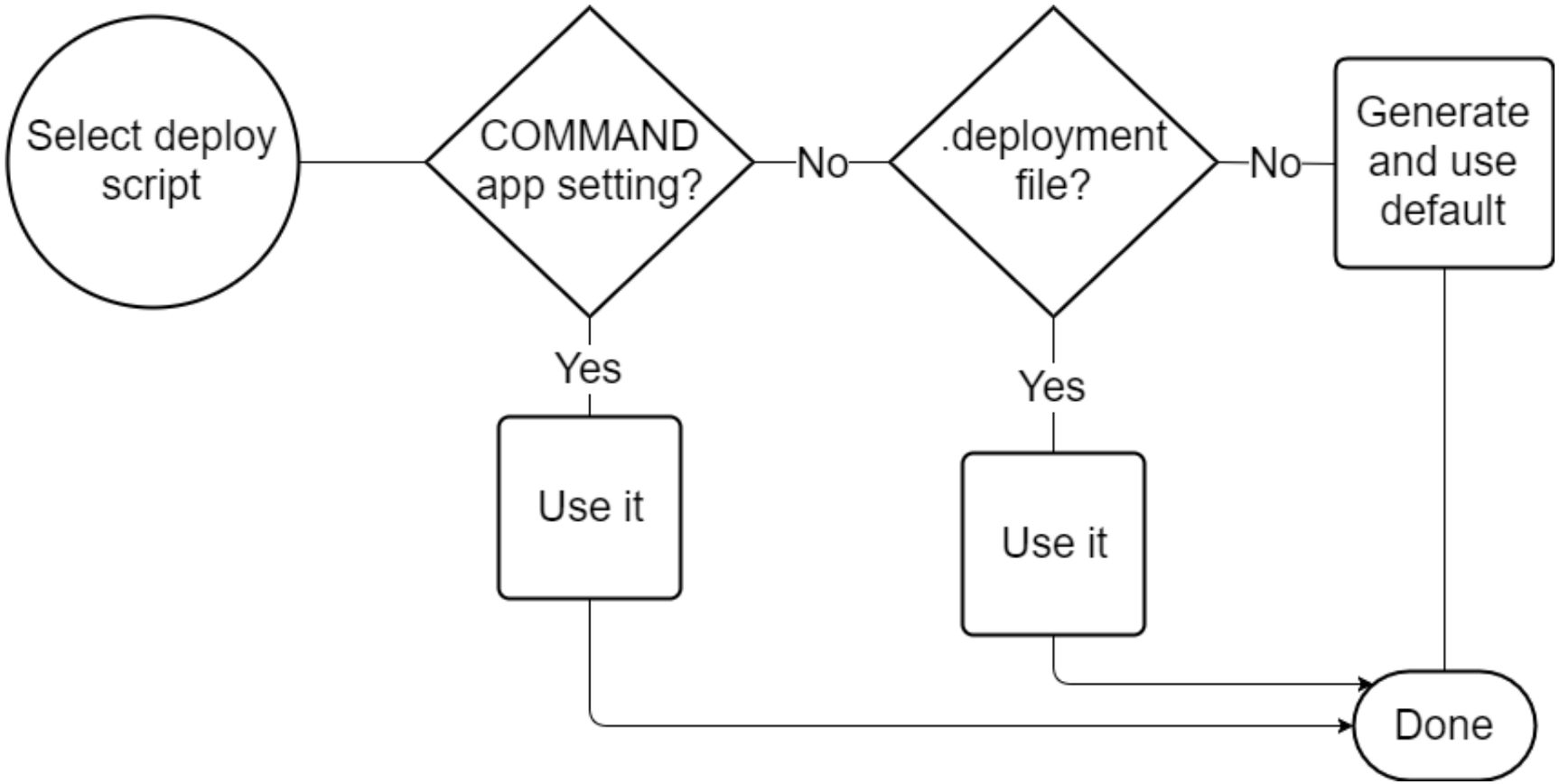


Source: GitHub / BitBucket / Azure DevOps Repos



Version control deployments

Select deployment script



Summary – Local git deploy

- Local git push
- Locally runnable build script
- Customized with quality assurance steps
- Use run from package feature
- Good for early development, one or two developers
- Swift feedback loops.

Summary – Deploy via GitHub

- Good solution for small teams
- Incorporate development workflow
  - Branching strategies
  - Quality assurance via pull requests
- Deploy to dev / test / production from different branches.

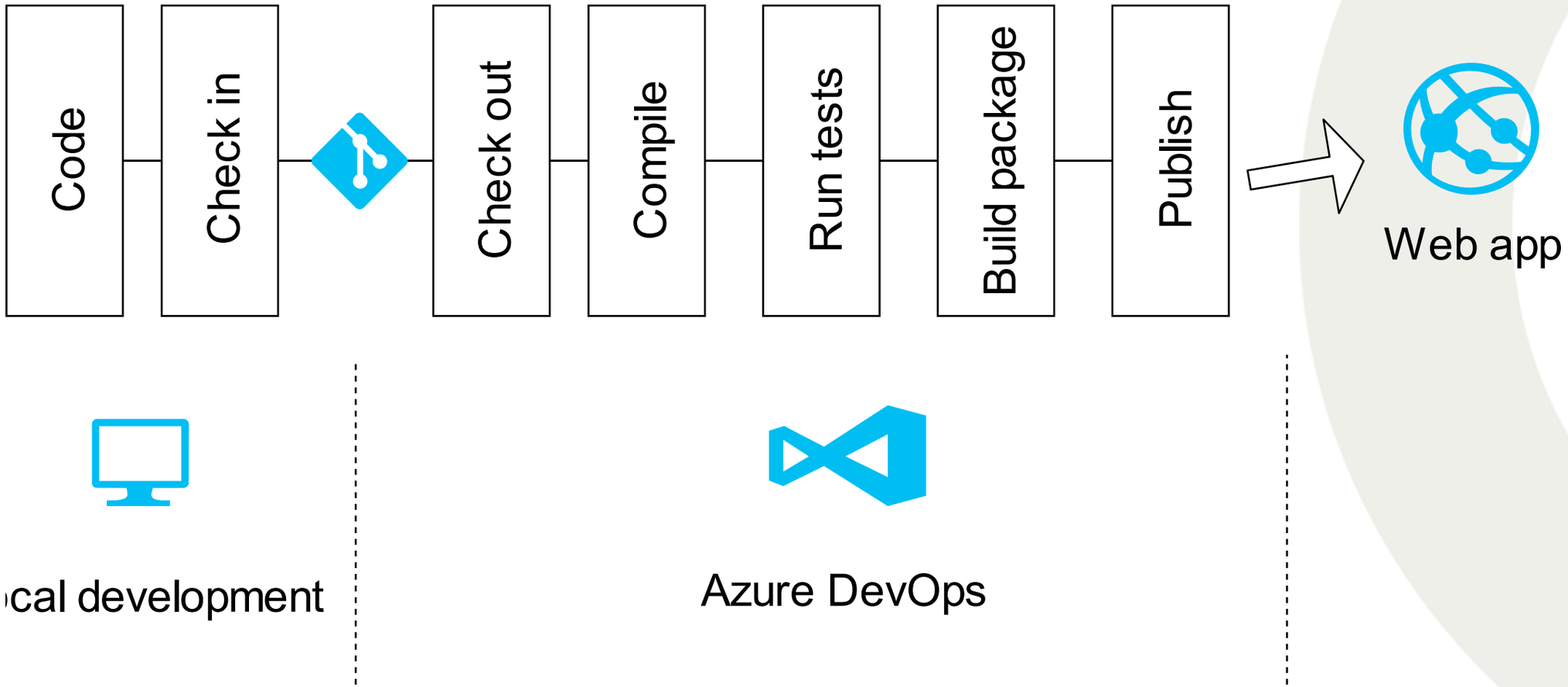
Version control deployments: possible enhancements

- Use deployment slots
  - Either manual or auto-deploy
- Develop scripts to rollback to earlier versions
- Use other build frameworks – Cake, Psake, etc.
- Add automated site verification steps
- Post deployment updates to Slack or Teams.

Shortcomings

- Difficult to coordinate large solutions with many services / applications
- Deployment history and reports not easily available
- No support for advanced deployment workflows.

Azure DevOps Pipelines



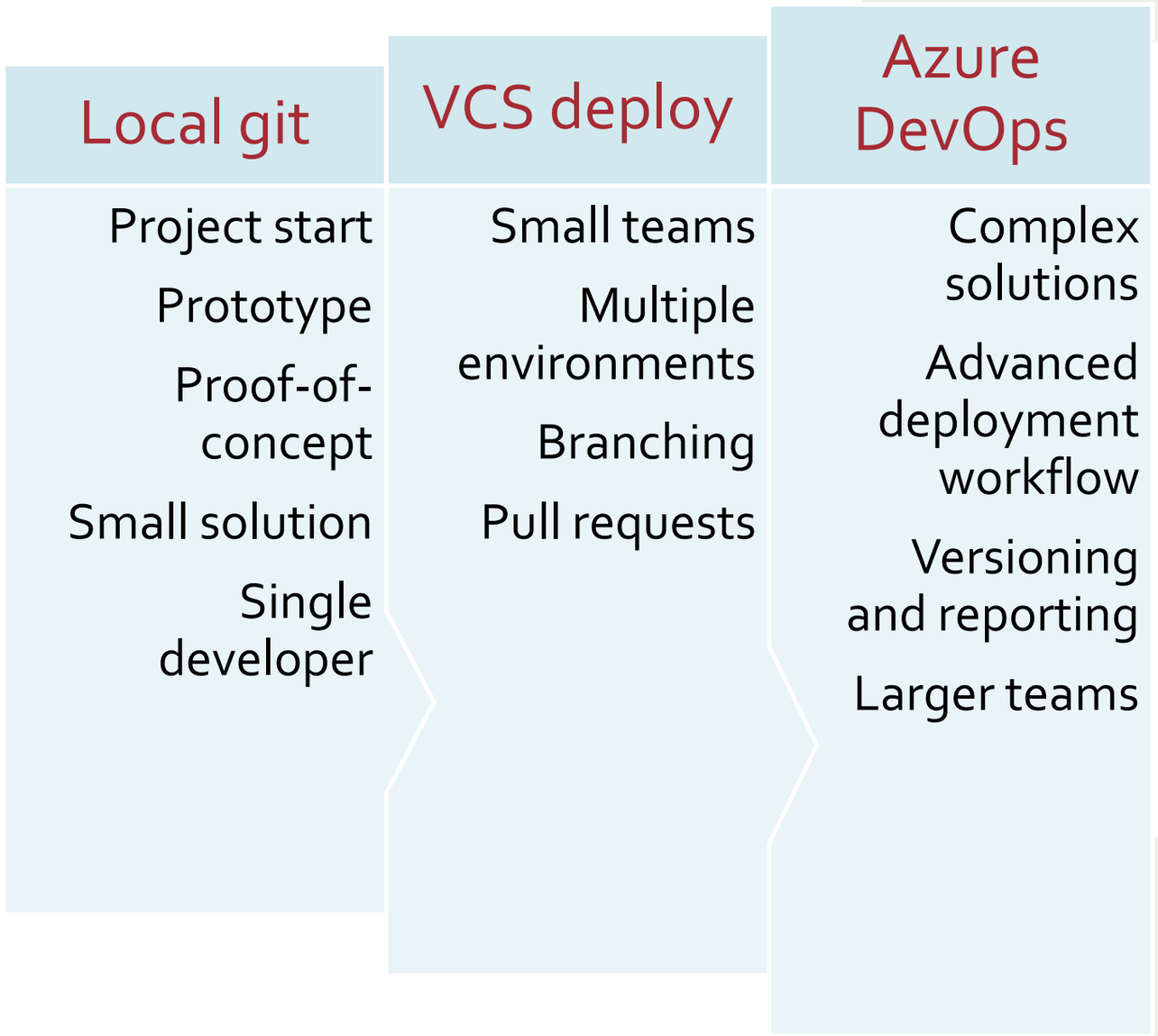
Azure DevOps Pipelines

Summary – Azure DevOps Pipelines

- Offload builds to dedicated environment
- Build policies for pull requests
- More advanced pipelines easily available
- More moving parts
- Lose the possibility to test / validate pipelines locally – sort of

Pick and choose

- Deployment approach based on your needs
- Switch approach with your project lifecycle
- Your build script
  - Is kept in VCS
  - Is in proximity with your code
  - Is reused if switching build/deployment approach



Travel light

Photo by Holly Mandarich on Unsplash

Thank you

- Contact
  - @vidarkongsli
  - vidar.kongsli@bredvid.no
- Stuff
  - Presentation files on GitHub
    - http://bit.ly/wroc19azure
  - Blog
    - https://blog.bredvid.no
    - https://vidar.kongs.li