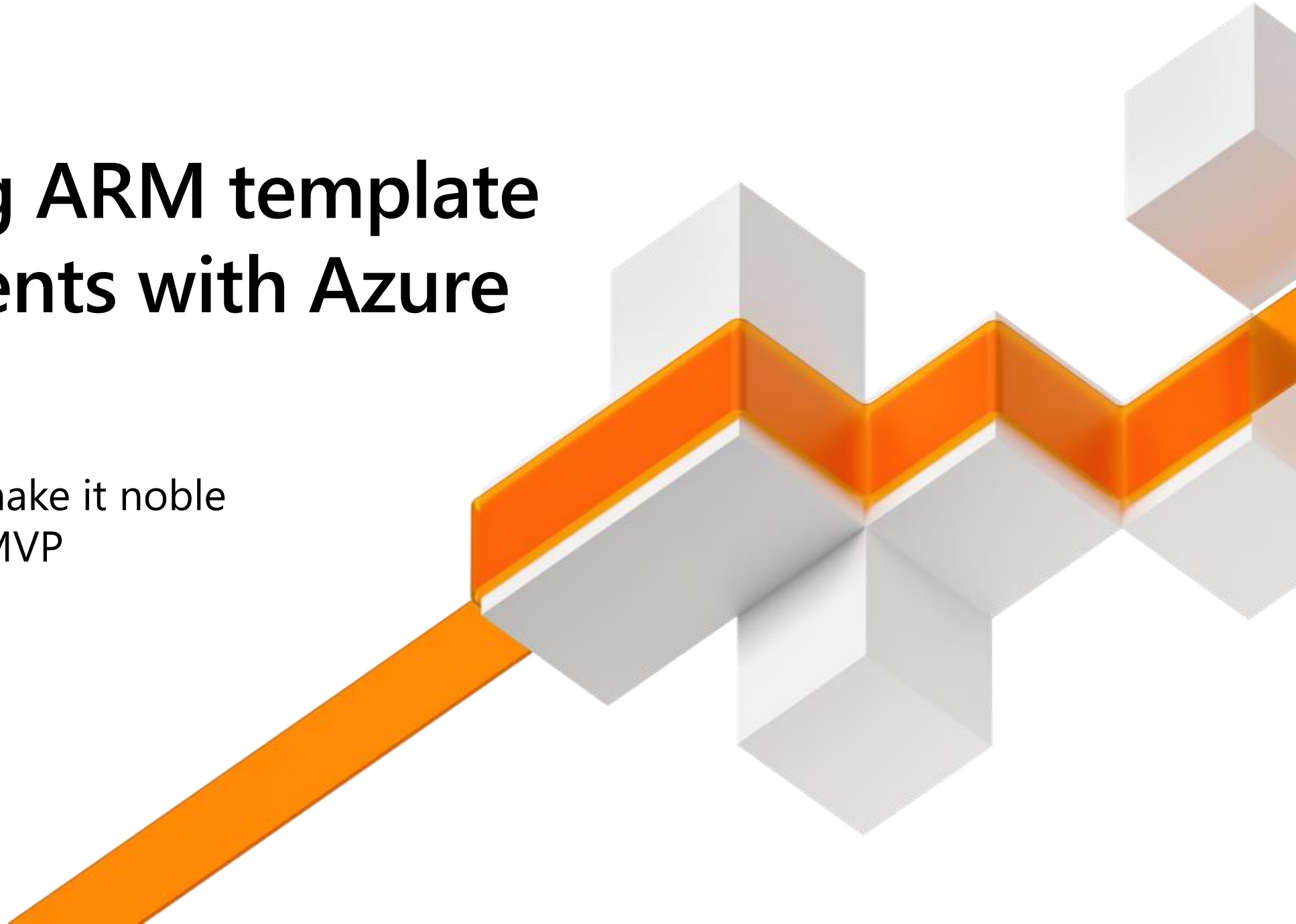




Mastering ARM template deployments with Azure DevOps

Marcel Zehner | make it noble
Microsoft Azure MVP
@marcelzehner
marcelzehner.ch



What to expect from this session

- How Azure DevOps can support your ARM template deployment process
- Repo management
- Pipeline management
- Demos

What NOT to expect from this session

- Enough content to leave this session as an expert
- Swiss chocolate giveaways

About me

- Marcel Zehner
- make it noble
- Microsoft Azure MVP
- Speaker, writer & blogger
- Community champion
- Experts Live CH & EU



@marcelzehner



marcelzehner



marcelzehner.ch



Manage your ARM templates code

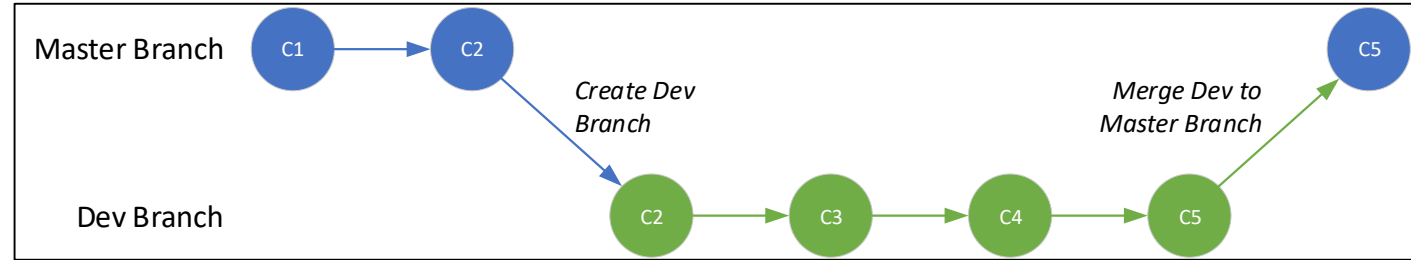


Manage your code

- Use source control management (SCM)
 - Git
 - Protect your code with policies
 - Track code changes
- Use a distributed repo approach
 - Store repo on Azure DevOps
 - Clone/pull it to your dev box
 - Change, commit and push back

Branches

- Master branch
 - Contains code version that is production-ready
 - Must be protected
- Dev branch
 - Based on master branch
 - Used to change your code
 - Test and test-deploy your code
 - Merge into master in a controlled way
 - Destroy dev branch





Branch	Commit	Author	Authored Date	Behind Ahead
dev	b12fb824	Marcel Zehner	Just now	0 1
master Default Compare	ff083573	bigboss	Friday	

Protect the master branch

- Store Git repo on Azure DevOps
- Configure a branch policy
 - No direct commits to the master branch
 - Dev needs to create a pull request
 - Pull request is analyzed, discussed and (maybe) approved

Branch policies for master

 Save changes  Discard changes

Protect this branch

- Setting a Required policy will enforce the use of pull requests when updating the branch
- Setting a Required policy will prevent branch deletion
- Manage permissions for this branch on the [Security page](#)

☒ **Require a minimum number of reviewers**
Require approval from a specified number of reviewers on pull requests.

Minimum number of reviewers

☐ Allow requestors to approve their own changes

☐ Allow completion even if some reviewers vote to wait or reject

☐ Reset code reviewer votes when there are new changes

☐ **Check for linked work items**
Encourage traceability by checking for linked work items on pull requests.

☒ **Check for comment resolution**
Check to see that all comments have been resolved on pull requests.

Policy requirement

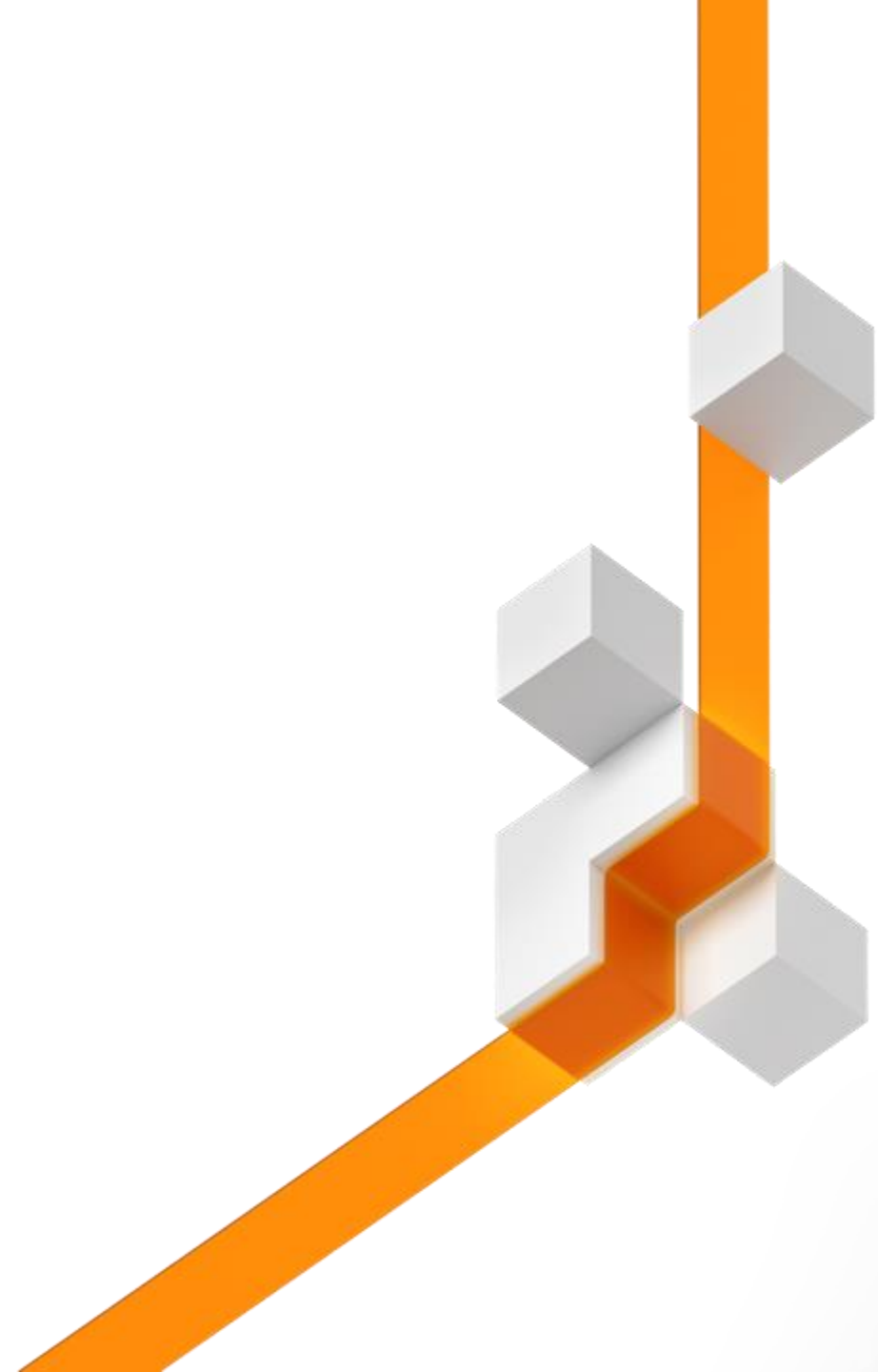
☒ **Required**
Block pull requests from being completed while any comments are active.

☐ **Optional**
Warn if any comments are active, but allow pull requests to be completed.

Create a pull request

- Compare files
- Comment and discuss
- Approve changes
- Merge dev into master branch
- Delete dev branch
 - Create a new one if you plan your next code changes

Pipelines

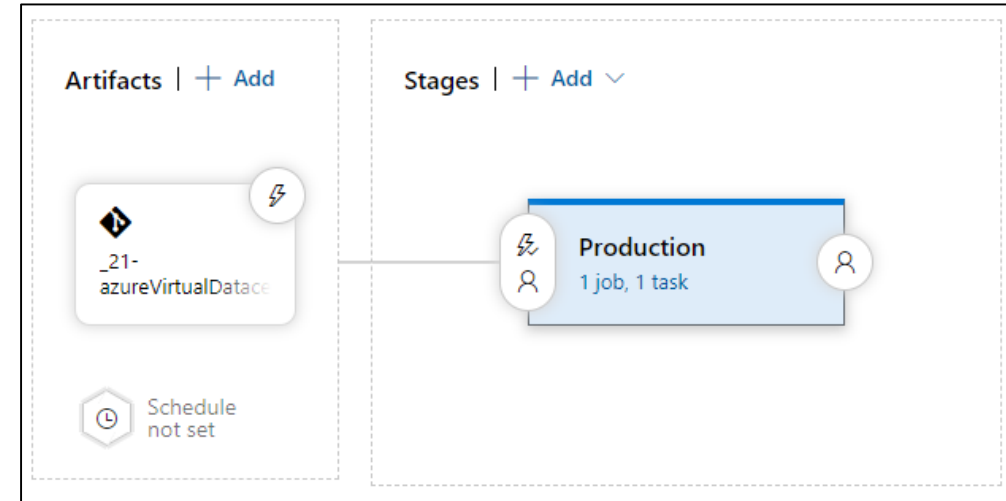


Release your code

- We need a consistent way to release our code
 - Dev box is OK to test your code
 - For int, stage, prod environment this is a no-go!
- Use Azure DevOps pipelines instead
 - Structured approach to prepare and release your code
 - Process is identical, no matter who triggers it
- Gives you good control
 - Multiple stages, steps, tasks etc.
 - Quality gates and approvals

Types of pipelines

- Classic
 - Build pipelines (test and prepare code)
 - Release (release code to Azure)
 - Graphics designer
- Modern
 - All in one (build & release)
 - Code designer
 - Pipeline as code
 - Allows to re-use code (templates)

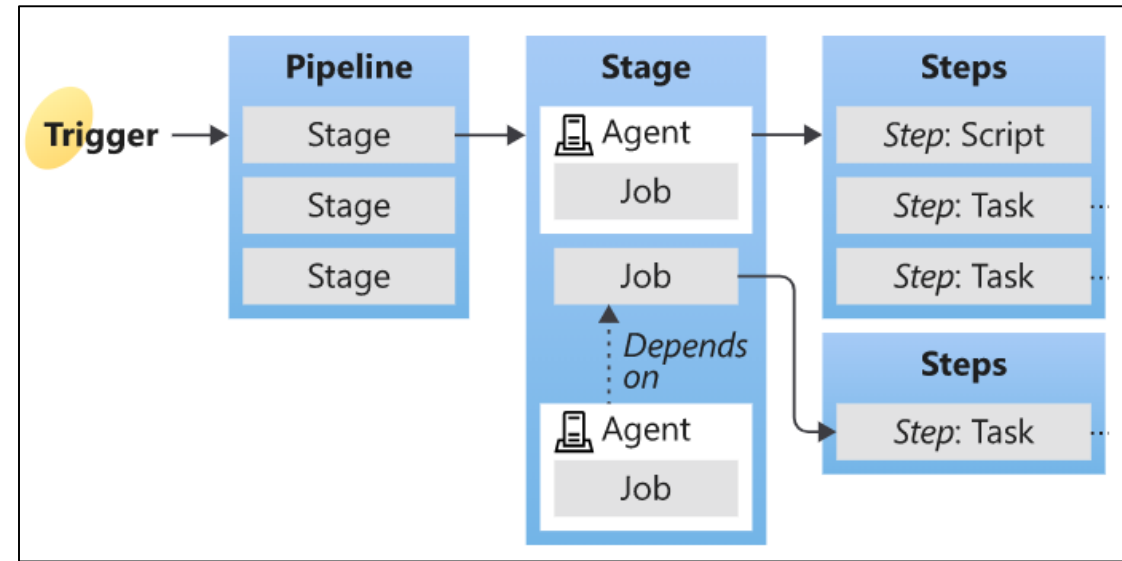


```
stages:
- stage: Preparation
  jobs:
  - job: ARMTemplatesQualityCheck
    pool: Hosted VS2017
    steps:
    - checkout: 11ResourceTemplates
    - checkout: 21AzureVirtualDatacenter
    # - script: dir $(System.DefaultWorkingDirectory)

    Settings
    - task: RunARMTTKTests@1
      displayName: "ARM Template Test for Resource Templates"
      inputs:
      - templateLocation: '$(System.DefaultWorkingDirectory)/11-resourceTemplates/*'
      - resultLocation: '$(System.DefaultWorkingDirectory)/11-resourceTemplates/'
```

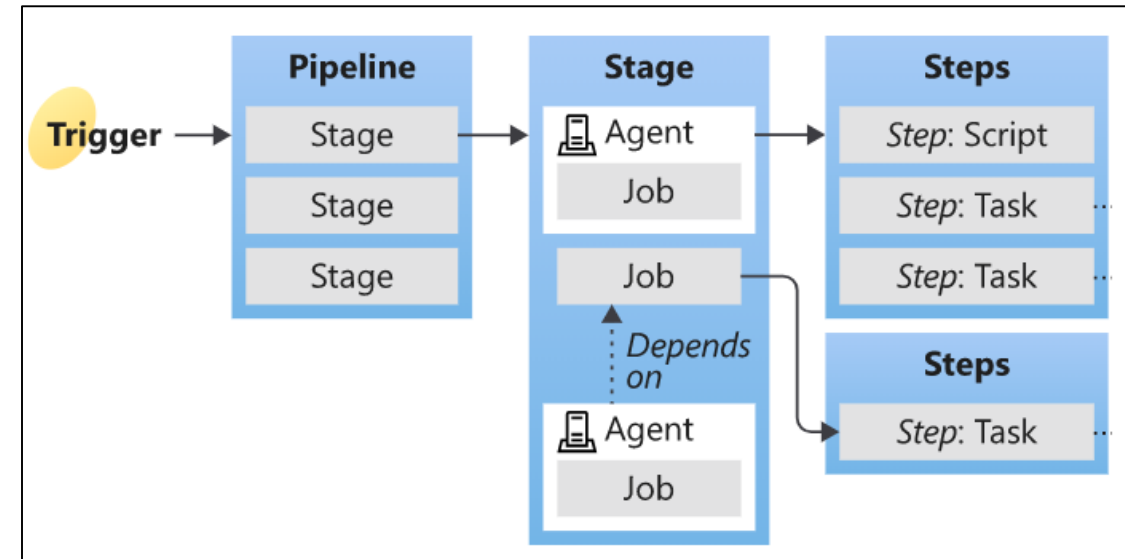
Components 1

- Artifacts
 - Code that needs to be released
 - Azure DevOps Git repo
- Stages
 - Major division inside a pipeline to build a logical boundary
 - Pre- and post actions



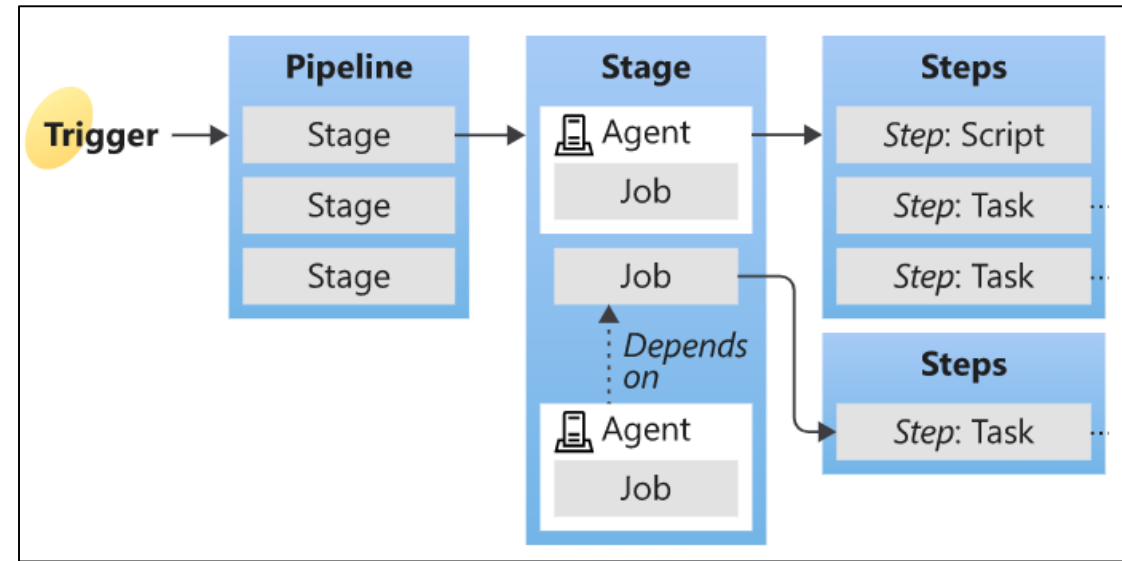
Components 2

- Agent Jobs
 - Part of a stage
 - Contains steps that need to be executed
 - Executed by an agent
 - Agent pool (hosted or own)
 - Server (Azure DevOps Server)
 - Containerized



Components 3

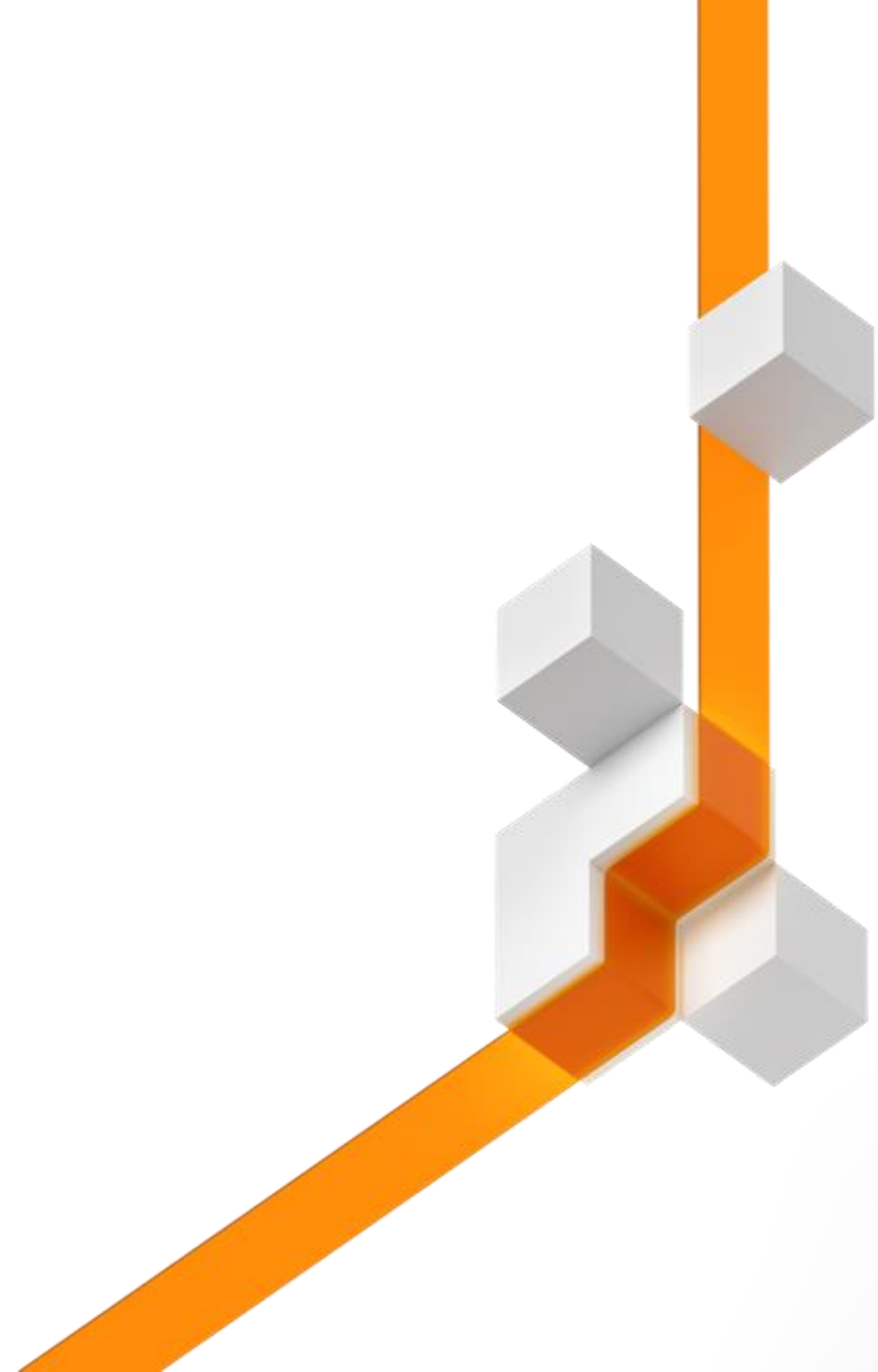
- Steps
 - Effective work
 - Different types of steps
 - Script > Powershell, bash etc. script
 - Task > Packaged script/procedure that can be used from a catalog



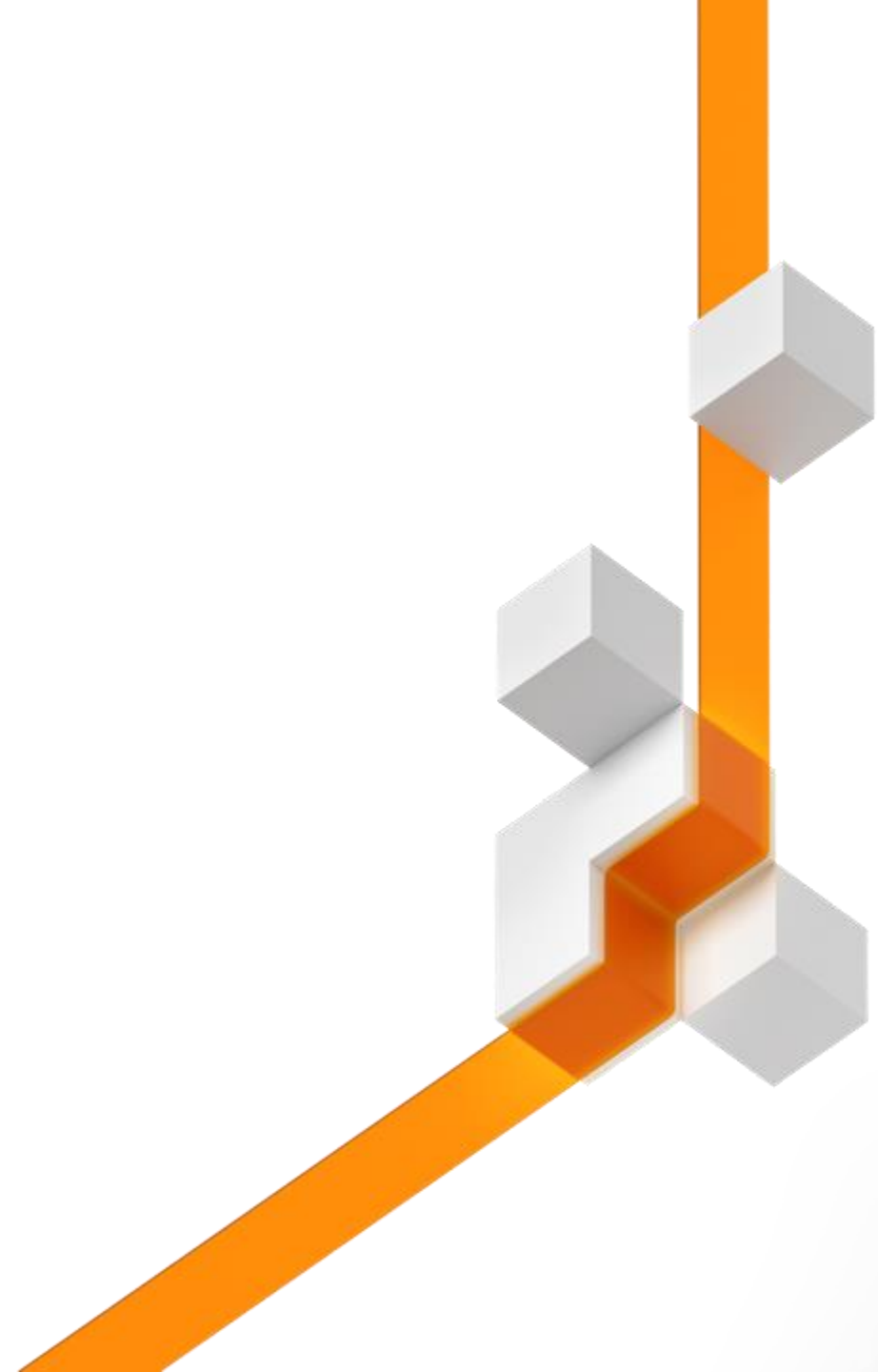
Other important stuff

- Approval & gates
 - Stage pre- and post deployment conditions
 - Approvals by persons
 - Quality gates (e.g. Azure governance status, monitoring etc.)
- Conditions
 - When to continue with a step/stage
- Triggers
 - Automatically (Git commit, pull request)
 - Manual

Demo



Key takeaways



Key takeaways

- Azure DevOps is your friend to manage your code and releases
- Protect and control your code
- Use structured approaches to release your code
- In a perfect world, only the pipeline can modify your resources



Mastering ARM template deployments with Azure DevOps

Marcel Zehner | make it noble
Microsoft Azure MVP
@marcelzehner
marcelzehner.ch

