BASTA!

by entwickler.de

Advanced Pull Requests Checks and Policies

Neno Loje

MVP für Visual Studio ALM

www.teamsystempro.de

Marc Müller

MVP für Visual Studio ALM

www.4tecture.ch

Slide Download

https://www.4tecture.ch/events/basta25-advancedpullrequest

DEMO

Pull Request

Advanced Pull Requests

PR Deployments - Why?

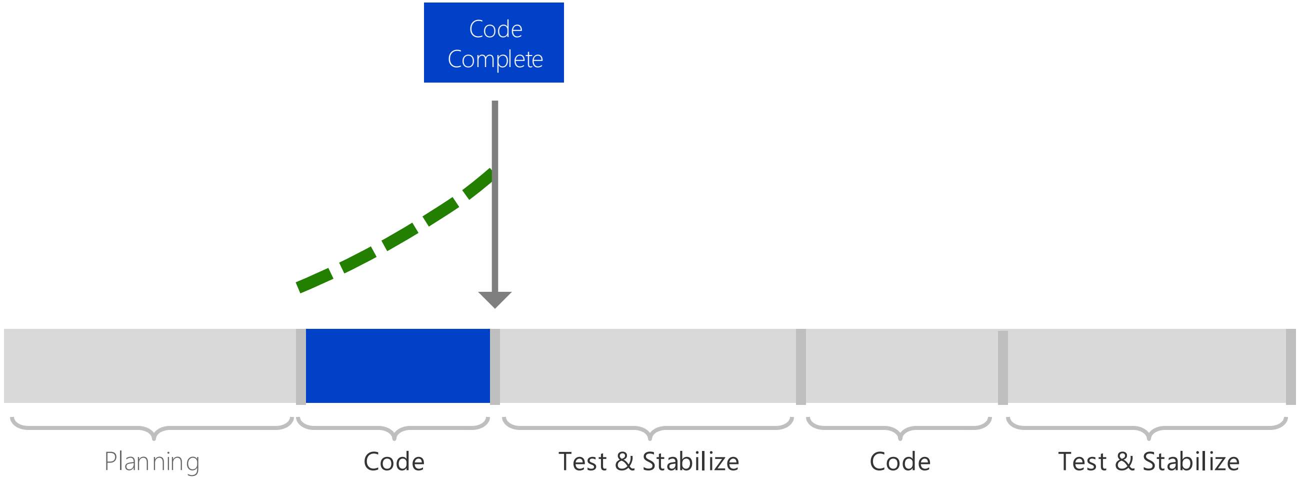
4tecture

empower your software solutions

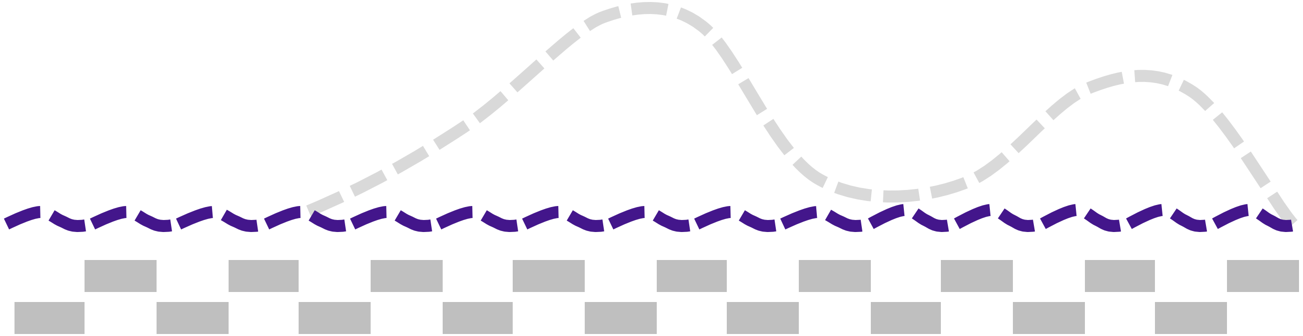
Image source: https://giphy.com/gifs/A5cMONNFFwRgA

Fail fast!

Before


Code Complete
Planning
Code
Test & Stabilize
Code
Test & Stabilize

After



Shift left

Move the testing process to the left

- Integrate testing into the sprint / pull request
- Fast detection and fix
- Testers are part of the team

Continuous Testing

- Effective and continuous integration
- No Bulks of tests / bug-fixing

Without shift left context-switching is expensive / lowers throughput drastically

Shift Left Benefits

- Reduced costs involved in dev/test
- Early bug detection – better quality
- Effective resolution of bugs
- Massive time and effort saved

Image source: https://tenor.com/view/hamster-wheel-run-gif-10746499

If it hurts, do it more often!

Test Automation

- Reduce test time
- Have regression tests
- Focus on test design and management, rather than manual repetitive tasks

There is no better place than production!

Image source: https://giphy.com/gifs/line-downvote-production-Ta7lg6ZMBOqzu

Production-like environments

- Integration is important
- Real scenarios with real tests
- Test how it will be used in production
- Real flows, delays, latency, retries, etc.
- Reduction of no-repro bugs

Conclusion

- No “shift left” only strategy
  Combination of shift left and system testing on target system recommended
- Reducing or avoiding long circle times is crucial
- “There is no place like production”

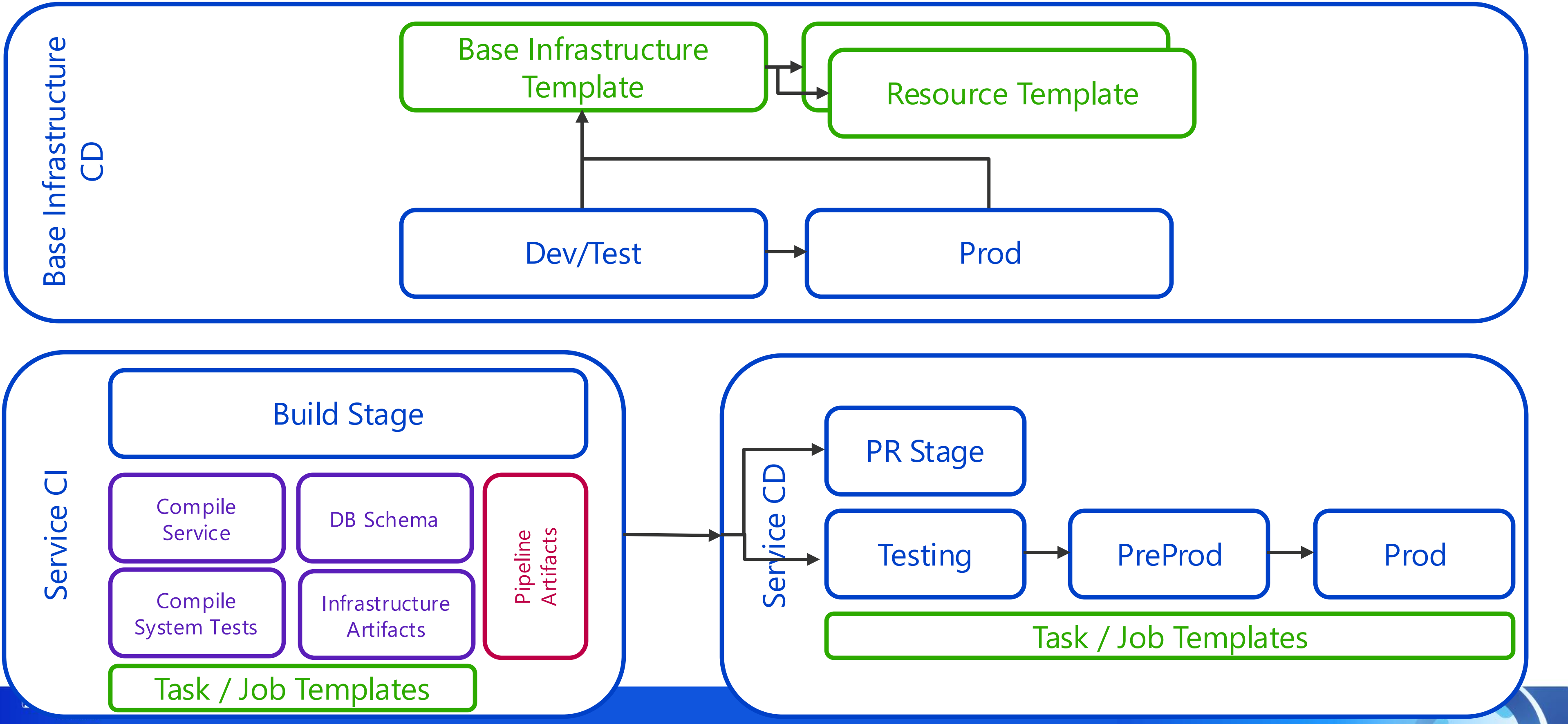
Advanced Pull Requests

Azure Pipelines Structure and Setup

4tecture

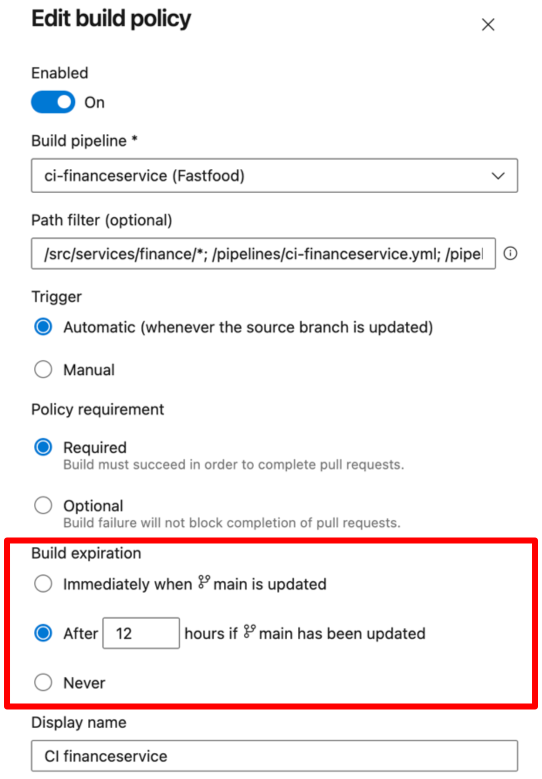
empower your software solutions

Best Practices



Is PR Validation 100% safe?

- It depends…
- Integrate multiple parallel PRs
  - Sequential
  - Parallel
- Depends on PR integration frequency / pipelines runtime



Edit build policy ×

Enabled

On

Build pipeline *

ci-financeservice (Fastfood)

Path filter (optional)

/src/services/finance/*; /pipelines/ci-financeservice.yml; /pipel

Trigger

● Automatic (whenever the source branch is updated)

○ Manual

Policy requirement

● Required
Build must succeed in order to complete pull requests.

○ Optional
Build failure will not block completion of pull requests.

Build expiration

○ Immediately when main is updated

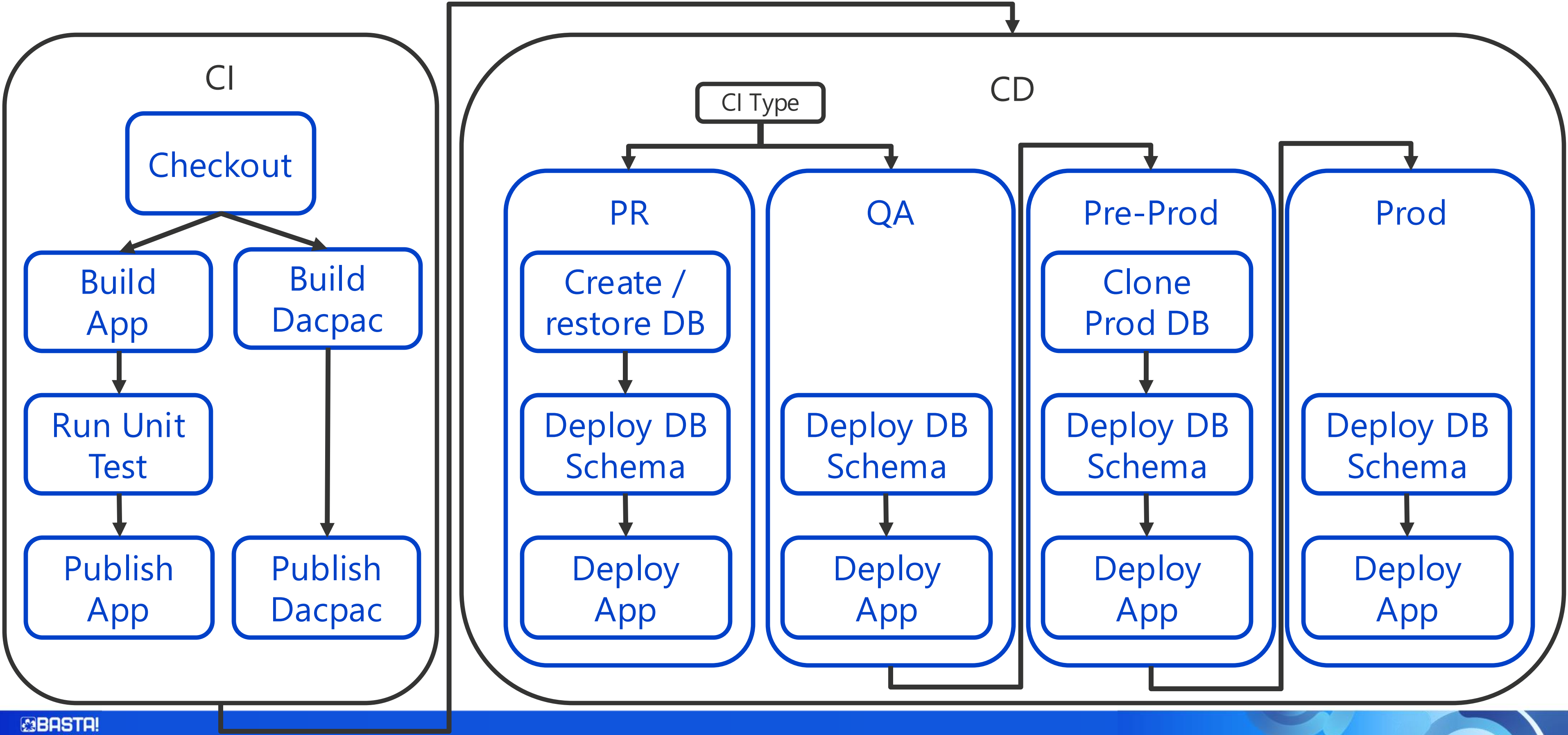
● After 12 hours if main has been updated

○ Never

Display name

CI financeservice

CI / CD Pipeline



Advanced Pull Requests

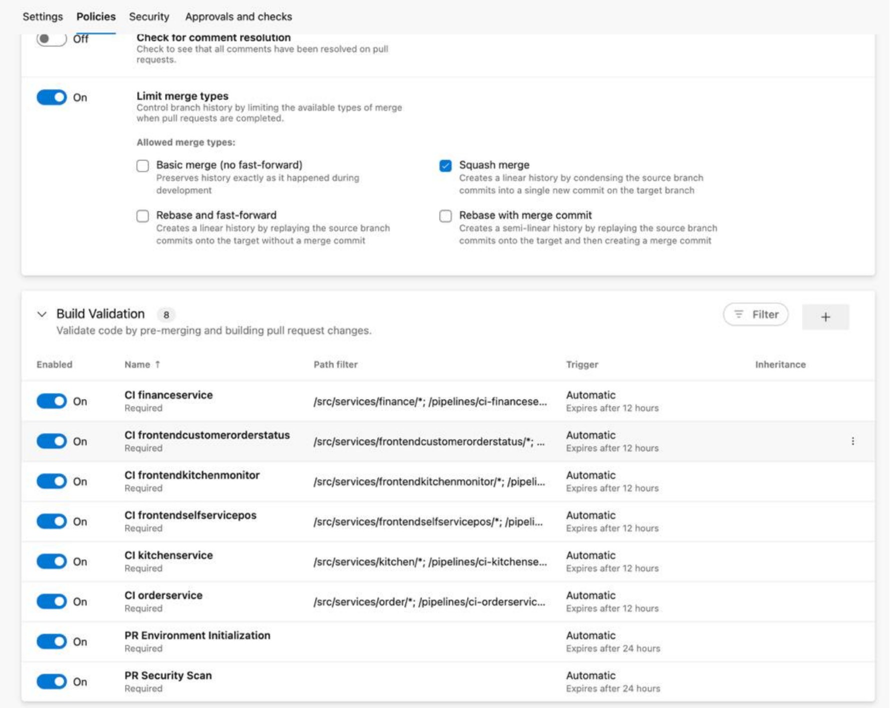
Branch Policies

4tecture

empower your software solutions

Branch Policies

- Forced policies to be able to integrate into target branch
- General Policies
- Build Validation Policies
- Status Checks
- Automatic Reviewers



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

On **Limit merge types**
Control branch history by limiting the available types of merge when pull requests are completed.

Allowed merge types:

- □ Basic merge (no fast-forward)
  Preserves history exactly as it happened during development
- ☑ Squash merge
  Creates a linear history by condensing the source branch commits into a single new commit on the target branch
- □ Rebase and fast-forward
  Creates a linear history by replaying the source branch commits onto the target without a merge commit
- □ Rebase with merge commit
  Creates a semi-linear history by replaying the source branch commits onto the target and then creating a merge commit

Build Validation 8

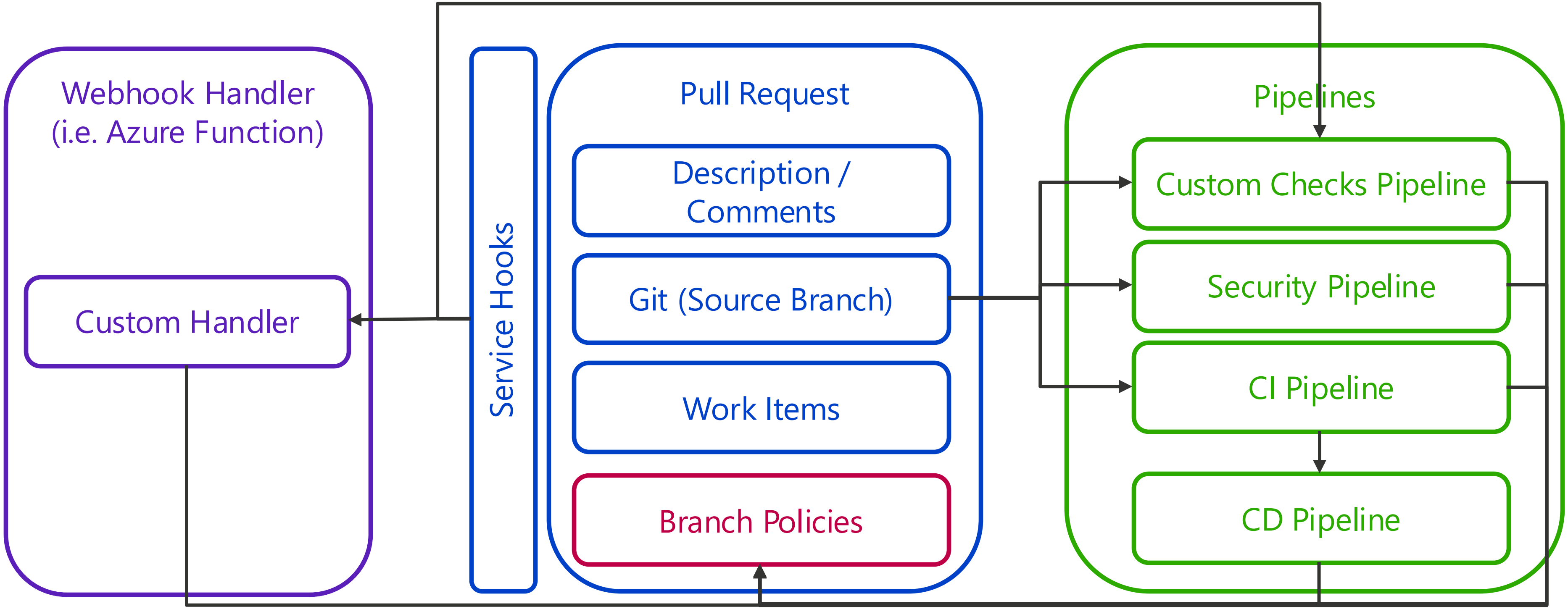
Validate code by pre-merging and building pull request changes.

Enabled	Name ↑	Path filter	Trigger	Inheritance
On	CI financeservice Required	/src/services/finance/*; /pipelines/ci-financese...	Automatic Expires after 12 hours	
On	CI frontendcustomerorderstatus Required	/src/services/frontendcustomerorderstatus/*; ...	Automatic Expires after 12 hours	
On	CI frontendkitchenmonitor Required	/src/services/frontendkitchenmonitor/*; /pipeli...	Automatic Expires after 12 hours	
On	CI frontendselfservicepos Required	/src/services/frontendselfservicepos/*; /pipeli...	Automatic Expires after 12 hours	
On	CI kitchenservice Required	/src/services/kitchen/*; /pipelines/ci-kitchense...	Automatic Expires after 12 hours	
On	CI orderservice Required	/src/services/order/*; /pipelines/ci-orderservic...	Automatic Expires after 12 hours	
On	PR Environment Initialization Required		Automatic Expires after 24 hours	
On	PR Security Scan Required		Automatic Expires after 24 hours	

Pull Request Status and Comments

- Status checks perform automated verification
- PR deployment should be added as PR status (as required policy)
  - Only deployable apps / services should be integrated
  - System integration tests should be part of the PR deployment
- Work Item Validation, PR Content Validation, External System Integration
- Automated comments guide the reviewers and product owners

Custom Checks and Policies



Custom Checks

- Security Checks
- Work Items → content, type, status, links
- PR Deployments → tests, monitoring, …
- PR Description → Check boxes checked?

Additional Automation

- Pull Request Tags
- AI description and work item text verification
- …

DEMO

Branch Policies

Advanced Pull Requests

Automatic Code Reviewers

4tecture

empower your software solutions

DEMO

Automatic Code Reviewers

Advanced Pull Requests

PR Templates

4tecture

empower your software solutions

PR Templates - About

- ## Definition
  - A PR template is a .md or .txt file whose contents are injected into the pull-request description when the PR is first created
- ## Built-in template types:
  - **Default template**
    Applied to *every* new PR unless a branch-specific file overrides it
  - **Branch-specific template**
    Automatically used when the PR's *target* branch matches the file name, e.g., dev.md for all dev/* PRs
  - **Additional/optional templates**
    Selectable from **Add a template** drop-down so authors can append extra guidance

PR Templates – Content, Benefits

- Typical content
  - check-lists (unit tests, docs updated, work-item links)
  - acceptance criteria
  - “Definition of Done” reminders
  - security or performance gates, etc.
- Benefits
  - Standardizes hand-overs
  - reduces missing info
  - accelerates reviews
  - enforces internal conventions & compliance and pairs well with branch policies for true “shift-left” quality

DEMO

PR Templates

Advanced Pull Requests

Azure Pipelines Incoming Webhook

4tecture

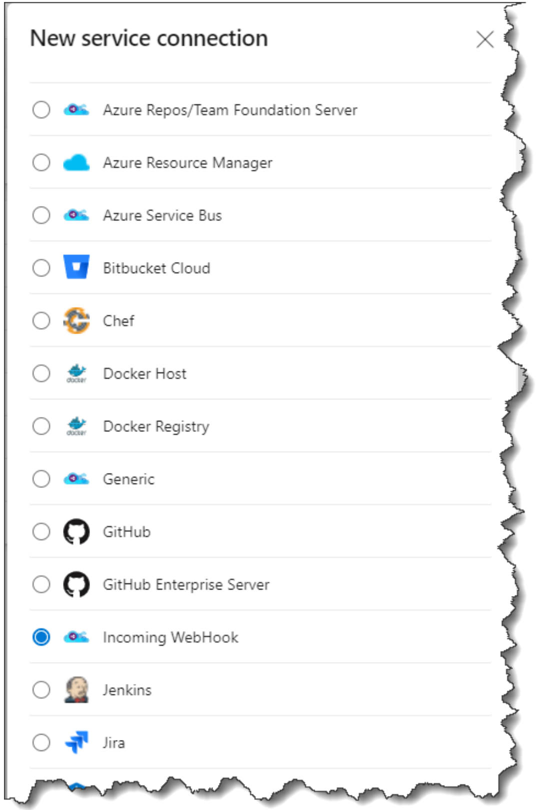
empower your software solutions

Trigger

- Main triggers are repository / code related
- Many automation scenarios trigger from other events
  - Work Item Update
  - Pull Request Update

Incoming Webhook Service Connection

- Incoming webhook can be defined as service connection
- Generic webhook trigger in pipelines
- Payload can be filtered



New service connection

○ Azure Repos/Team Foundation Server

○ Azure Resource Manager

○ Azure Service Bus

○ Bitbucket Cloud

○ Chef

○ Docker Host

○ Docker Registry

○ Generic

○ GitHub

○ GitHub Enterprise Server

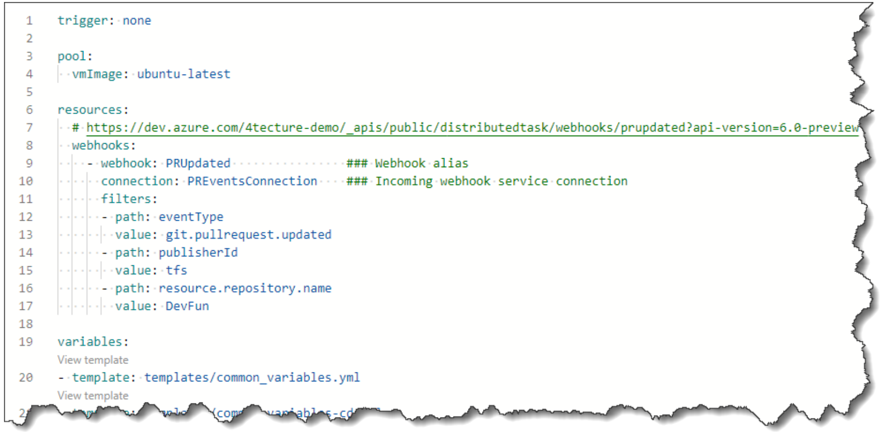
● Incoming WebHook

○ Jenkins

○ Jira

Pipeline Resource

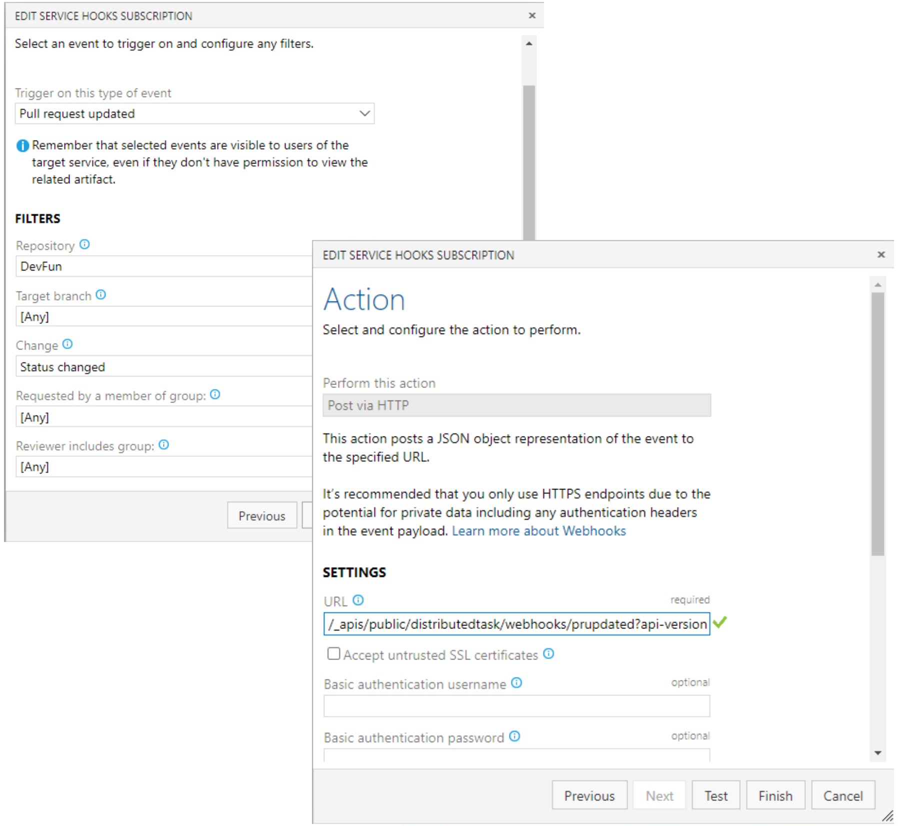
- Azure DevOps creates webhook endpoint based on alias
- Pipeline resource triggers pipeline
- Filters can be applied to trigger



```yaml
trigger: none

pool:
  vmImage: ubuntu-latest

resources:
  # https://dev.azure.com/4tecture-demo/_apis/public/distributedtask/webhooks/prupdated?api-version=6.0-preview
  webhooks:
    - webhook: PRUpdated                ### Webhook alias
      connection: PREventsConnection    ### Incoming webhook service connection
      filters:
      - path: eventType
        value: git.pullrequest.updated
      - path: publisherId
        value: tfs
      - path: resource.repository.name
        value: DevFun

variables:
View template
- template: templates/common_variables.yml
View template
```

Service Hook & Incoming Webhook

- Service hooks can trigger webhooks for many Azure DevOps Events
  - Build completed
  - Code pushed
  - Elastic agent pool resized
  - Pull request commented on
  - Pull request created
  - Pull request merge attempted
  - Pull request updated
  - Release abandoned
  - Release created
  - Release deployment approval completed
  - Release deployment approval pending
  - Release deployment completed
  - Release deployment started
  - Run stage approval completed
  - Run stage state changed
  - Run stage waiting for approval
  - Run state changed
  - Work item commented on
  - Work item created
  - Work item deleted
  - Work item restored
  - Work item updated



EDIT SERVICE HOOKS SUBSCRIPTION

Select an event to trigger on and configure any filters.

Trigger on this type of event: Pull request updated

Remember that selected events are visible to users of the target service, even if they don't have permission to view the related artifact.

FILTERS

Repository: DevFun

Target branch: [Any]

Change: Status changed

Requested by a member of group: [Any]

Reviewer includes group: [Any]

EDIT SERVICE HOOKS SUBSCRIPTION

Action

Select and configure the action to perform.

Perform this action: Post via HTTP

This action posts a JSON object representation of the event to the specified URL.

It's recommended that you only use HTTPS endpoints due to the potential for private data including any authentication headers in the event payload. Learn more about Webhooks

SETTINGS

URL (required): /_apis/public/distributedtask/webhooks/prupdated?api-version ✓

□ Accept untrusted SSL certificates

Basic authentication username (optional)

Basic authentication password (optional)

DEMO

Incoming Webhook

Advanced Pull Requests

Q & A

4tecture

empower your software solutions

Recap

- Pull Request is single point of interaction / status for developers, testers and product owners
- Fail fast – learn fast & fix fast
- Only an integrated change provides clarity if it runs successfully in production
- Use events and change triggers to verify the status of the PR and report it through PR Status / Checks

4tecture©

empower your software solutions

Marc Müller

Principal Consultant
für DevOps, ALM, TFS /VS, .NET

Microsoft Most Valuable Professional

4tecture

Webseite:	http://www.4tecture.ch
Schulungen:	http://4tecture.ch/trainings
Twitter:	@muellermarc

4tecture GmbH
Industriestr. 25
8604 Volketswil
Schweiz

+41 44 508 37 00
info@4tecture.ch

Haben Sie Fragen zu VS Team Services?

Neno Loje

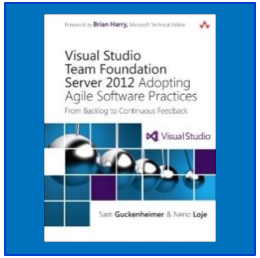
Freier Berater | Trainer | Coach
für Team Foundation Server/VSTS, Scrum, DevOps
neno@teamsystempro.ch

Webseite:	http://www.nenoloje.de
Buch:	http://www.nenoloje.de/buch
Schulungen:	http://www.tfscamp.de
Blog [EN]:	http://www.nenoloje.de/tfsblog
Blog [DE]:	http://www.nenoloje.de/meinblog

MVP
Microsoft®
Most Valuable
Professional

TFS Camp
Team Foundation
Server Camp

Neno Loje
teamsystempro.de



Foreword by Brian Harry, Microsoft Technical Fellow

Visual Studio
Team Foundation
Server 2012 Adopting
Agile Software Practices
From Backlog to Continuous Feedback

Visual Studio

Sam Guckenheimer & Neno Loje

4tecture©

empower your software solutions