AARHUS UNIVERSITET

Microservices and DevOps

DevOps and Container Technology

Build Pipelines

Henrik Bærbak Christensen

AARHUS UNIVERSITET

Pipeline

- *Deployment Pipeline:* An automated implementation of your application's build, deploy, test, and release process.

Humble et al.

Figure 1.1 The deployment pipeline

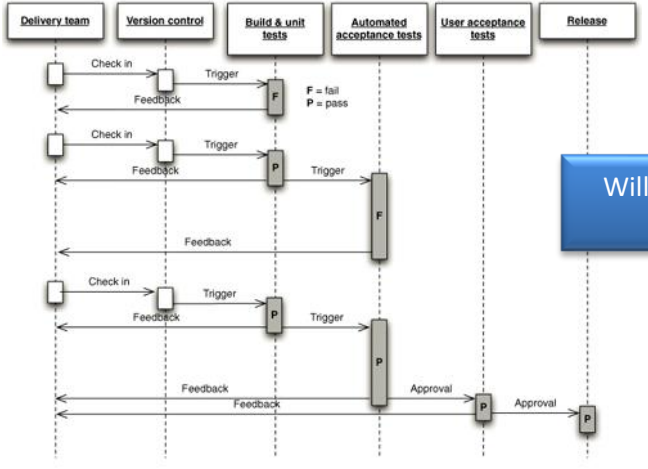
- Every change that is made in any artefacts of the application *triggers* the creation of a new instance of the pipeline.
  - Series of tests, each more demanding, each giving confidence in that *it will work*. Passing all tests means ready for release.

AARHUS UNIVERSITET

Pipeline

- Newman (as usual) waves his hand a bit…
  - *To have different stages in your build, creating what is known as a build pipeline [p 107]*
  - Fast test followed by slow tests
    - Why run the slow ones if the fast tests fail?

- Pattern: *Fail fast*



Will return to this figure later…

AARHUS UNIVERSITET

Pipeline

- Pipeline = stages of build, test, deploy, and release
- Modern build tools have aspects of this
  - Gradle ‘jar’ = compile, test, build deployment unit
  - The pipeline is predefined by gradle
- And – you could easily handcode (parts) of it!
  - A bash script that does gradle jar, docker build, docker push
- Of course, better with good tool support

AARHUS UNIVERSITET

Tech Choice?

- Lots of Build servers out there
  - Jenkins, Hudsun, GoCD, Bamboo, Concourse, ...
- Git and Bitbucket have their own system
  - *Rather* well integrated with the SCM system!!!
    - *Had a lot of issues with Concourse and Jenkins about checking out of a private repo...*
- Bitbucket in MSDO because...
  - History – I started there…
  - **TestContainers work well within BitBucket !**
- You are free to pick any you like...
  - But I can help more if you choose BitBucket ☺

AARHUS UNIVERSITET

Key Concepts

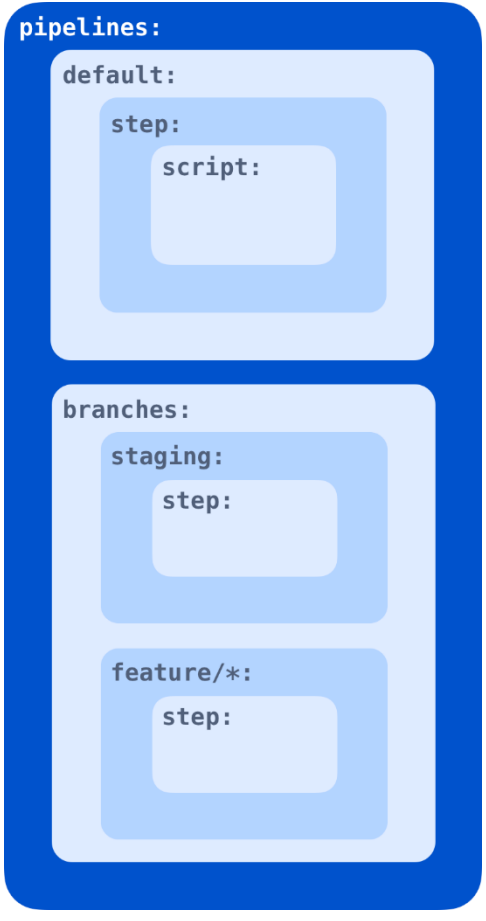
Key concepts

A pipeline is made up of a set of steps.

- Each step in your pipeline runs a separate Docker container. If you want, you can use different types of container for each step, by selecting different images.
- The step runs the commands you provide in the environment defined by the image.
- A single pipeline can have up to 10 steps.

Infrastructure-as-code
bitbucket-pipelines.yml
(YAML format of course)

https://support.atlassian.com/bitbucket-cloud/docs/get-started-with-bitbucket-pipelines/



AARHUS UNIVERSITET

Example Pipeline

- From Bitbucket sample code

```yaml
# You can specify a custom docker image from Docker Hub as your build environment.
image: maven:3.3.9

pipelines:
  default:
    - step:
        caches:
          - maven
        script: # Modify the commands below to build your repository.
          - mvn -B verify # -B batch mode makes Maven less verbose

        services:
          - mongo

definitions:
  services:
    mongo:
      image: mongo
```

```yaml
image: openjdk:8
pipelines:
  default:
    - step:
        script:
          - bash ./gradlew build
```

https://bitbucket.org/bitbucketpipelines/pipelines-guide-java/src/master/bitbucket-pipelines.yml

AARHUS UNIVERSITET

Strong Hint for SkyCave

- Potential first step in solution

```
image: openjdk:11

pipelines:
  default:

    - step: # === Unit test
        name: Unit Test
        caches:
          - gradle

        script:
          - bash ./gradlew test jacocoRootReport
```

- Huh? Why ‘./gradlew’ ???

AARHUS UNIVERSITET

Caches

- From buttom up is expensive wrt dependencies
  - Download everything every time ☹

What is dependency caching?

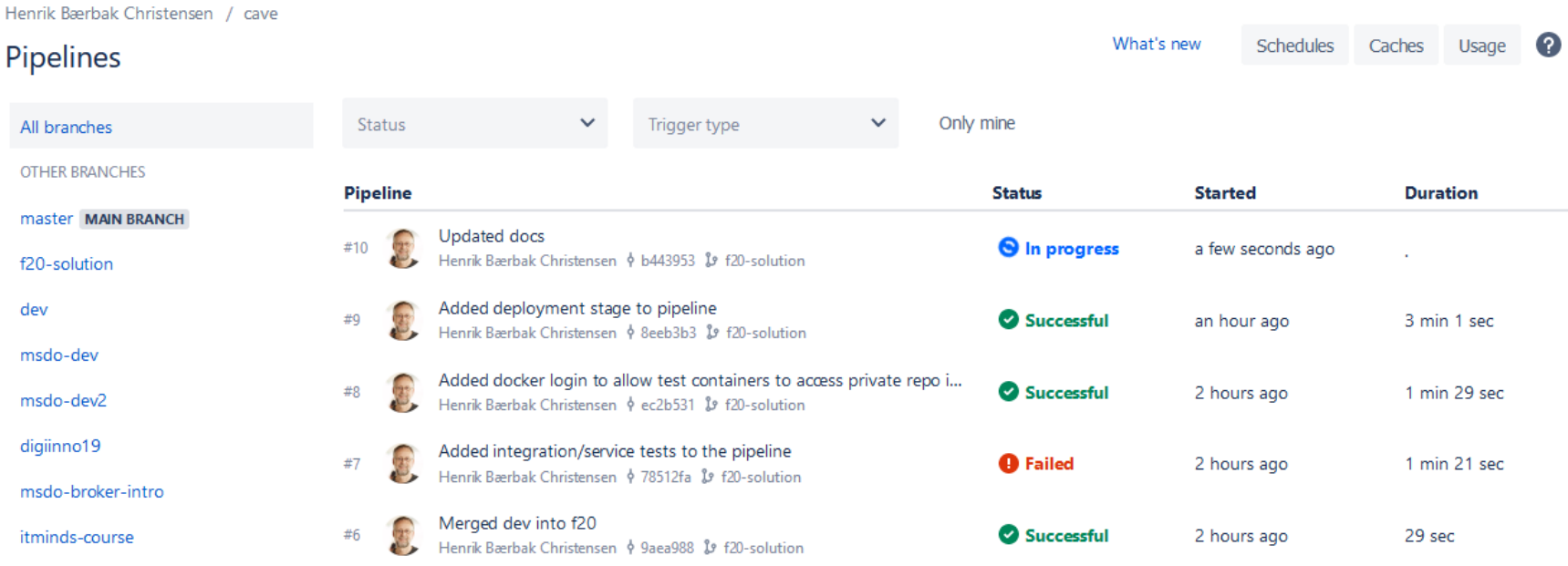
Most builds start by running commands that download dependencies from the internet, which can take a lot of time for each build. As the majority of dependencies stay the same, rather than download them every time, we recommend downloading them once into a cache which you can reuse for later builds.

- Predefined caches:

gradle	`caches:` `  - gradle`	`~/.gradle/caches`

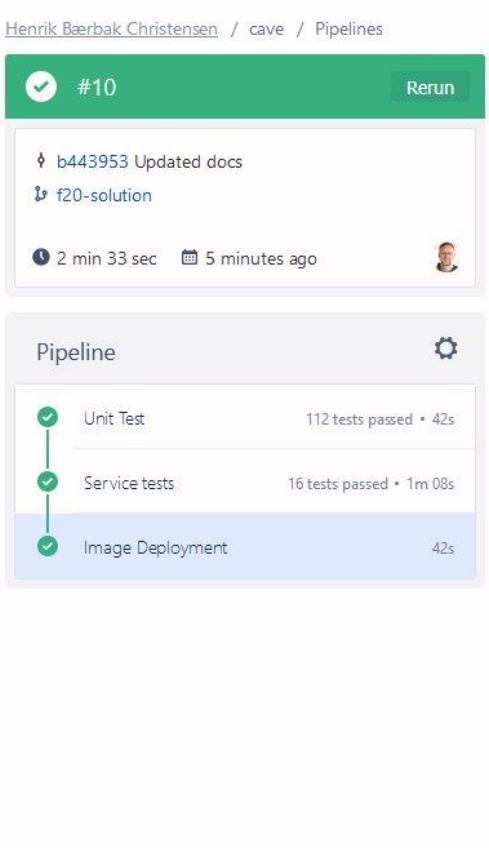
AARHUS UNIVERSITET

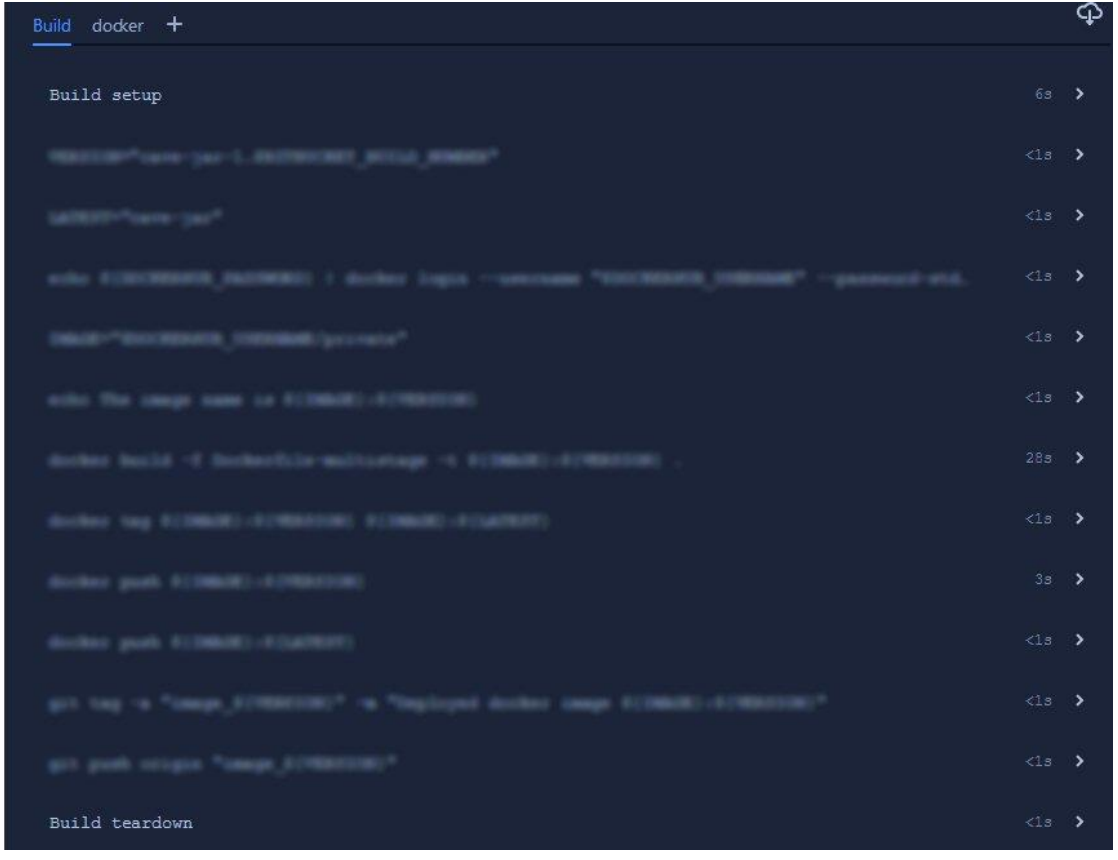
Commit Triggered



AARHUS UNIVERSITET

Pipeline Log

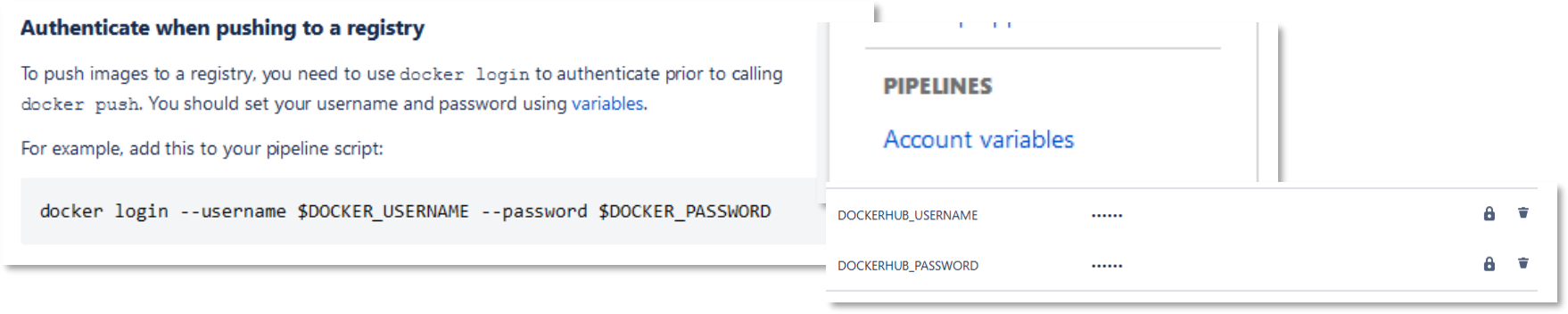




AARHUS UNIVERSITET

Image Release

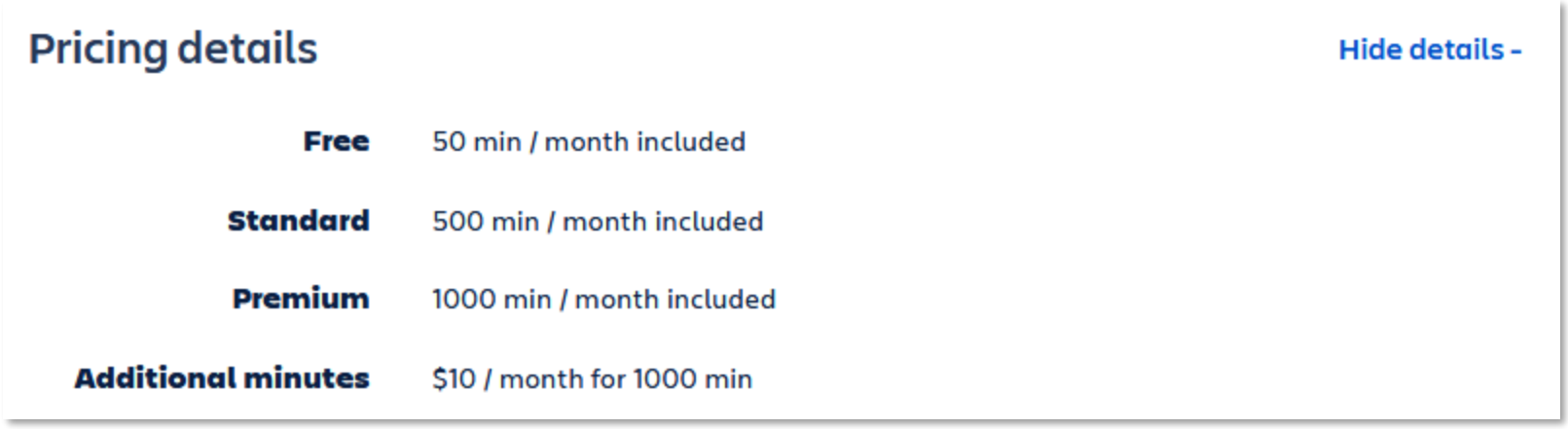
- … means *pushing the built image to Docker hub*
- … means having the credentials to Docker hub
- So you need to store the credentials in BitBucket
  - **Do not have them in the script!!!**



  - And I always spend 15 minutes trying to find the variables ☹

Caveat

- Free offer limit is 50 build minutes per month



Pricing details

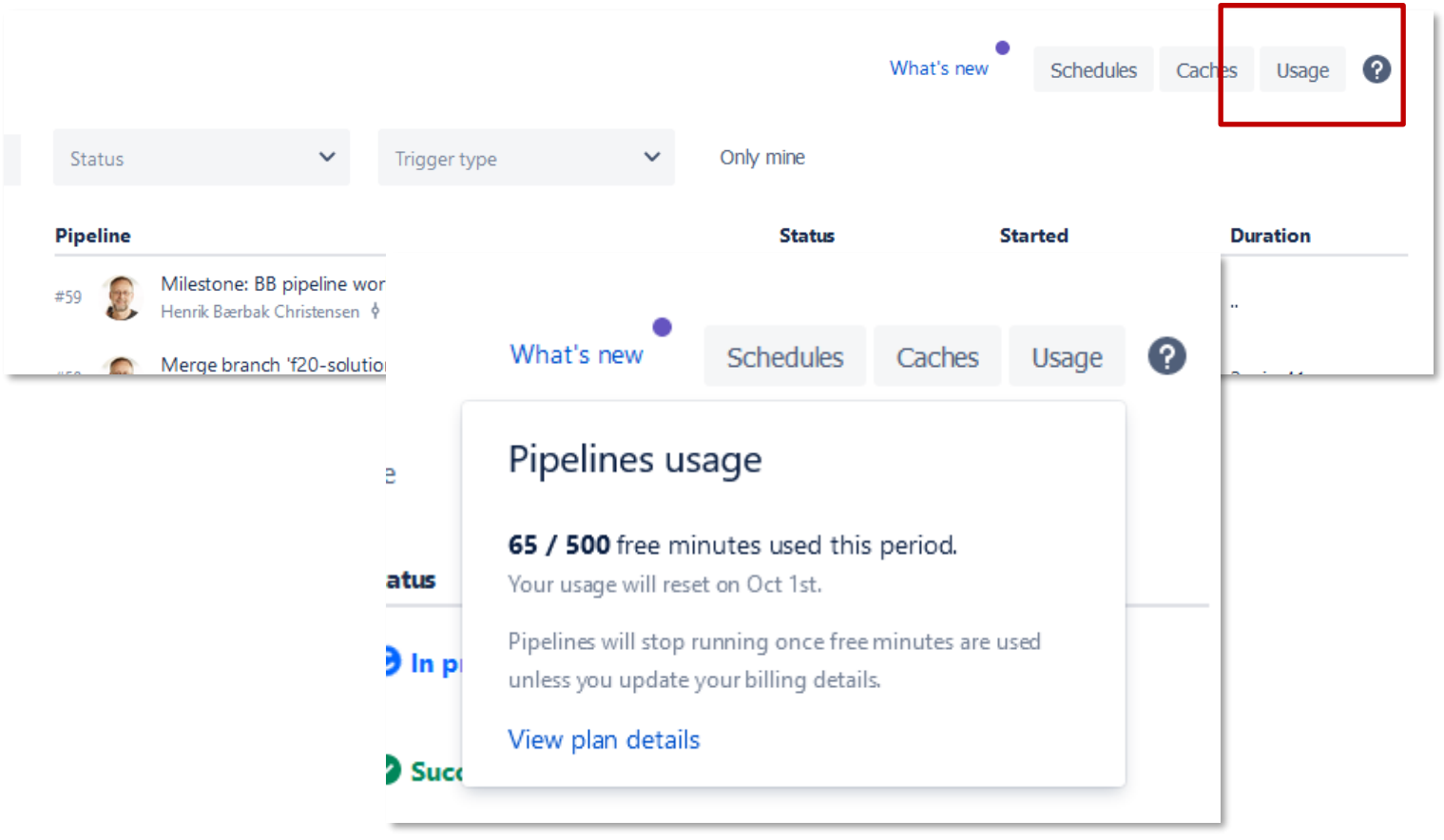
Hide details -

Free	50 min / month included
Standard	500 min / month included
Premium	1000 min / month included
Additional minutes	$10 / month for 1000 min

- So…
  - Keep the pipeline in your ‘deploy’ branch only!
    - In ‘dev’/’master’ you will run out of build time very soon…
  - Or – get an ‘academic license’ – I get 500 minutes for free…

AARHUS UNIVERSITET

Review Usage



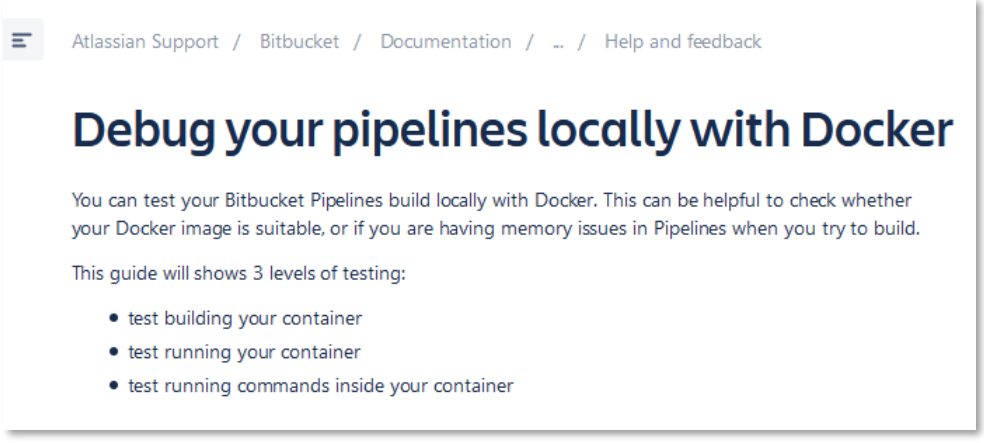
AARHUS UNIVERSITET

Take Small Steps

- Creating your first pipeline has a somewhat steep learning curve…
- Tutorial material is, well, ok.
  - Bit convoluted…
- Take small steps
  - Lots of commits, lots of waiting for pipeline to start, but debugging is not funny

Small Steps

- Maybe have a look at (though I have not tried it)



Atlassian Support / Bitbucket / Documentation / ... / Help and feedback

Debug your pipelines locally with Docker

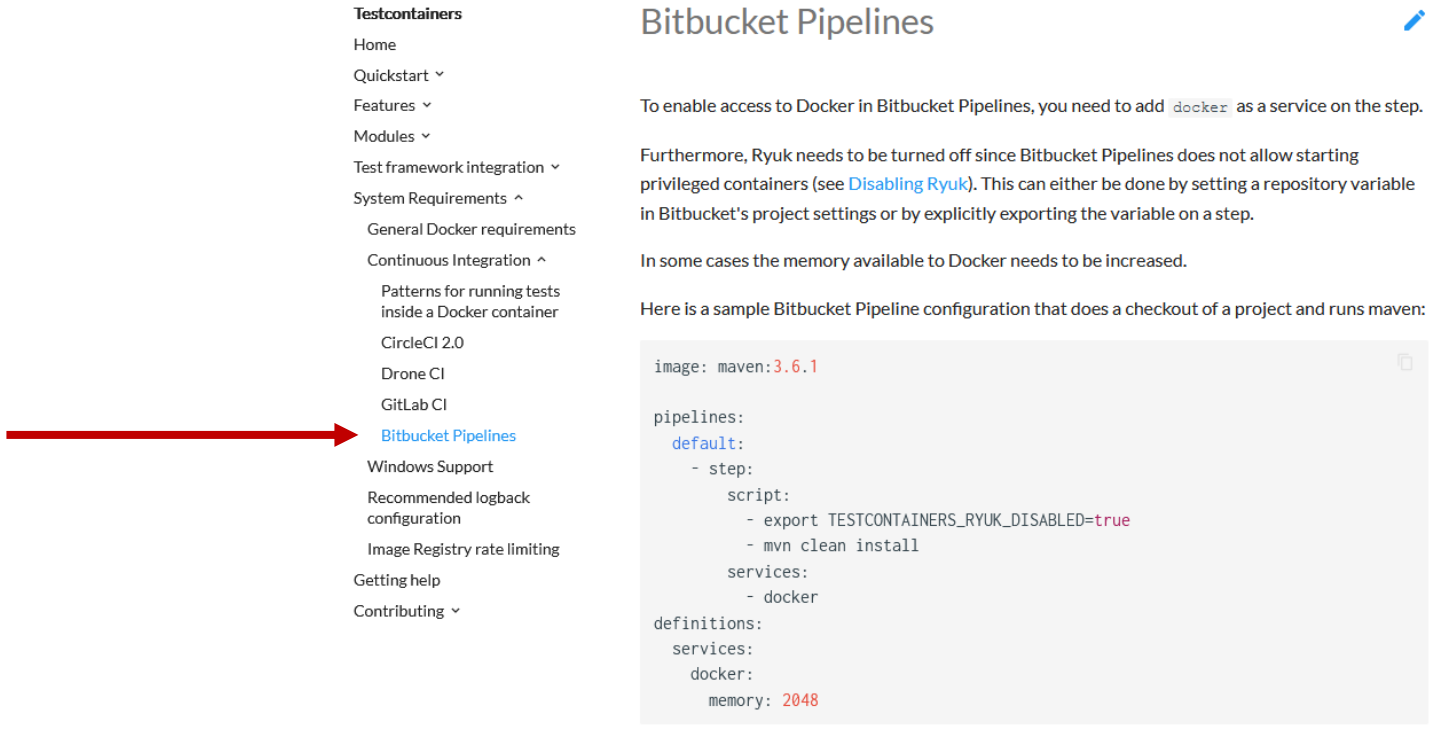
You can test your Bitbucket Pipelines build locally with Docker. This can be helpful to check whether your Docker image is suitable, or if you are having memory issues in Pipelines when you try to build.

This guide will shows 3 levels of testing:

- test building your container
- test running your container
- test running commands inside your container

Integration Tests in BitBucket

- TestContainers **works well** in Bitbucket pipelines !



AARHUS UNIVERSITET

Summary

- *Deployment Pipeline:* An automated implementation of your application's build, deploy, test, and release process.
- *Build Pipeline:* An automated implementation of your application's build, deploy, test, ~~and release~~ process.
- Uses the *Fail fast* pattern
- Tool supported – triggered by (certain) SCM commits