Docs as Code
Documentation Management Inspired by Software Development

Alex Jitianu

alex_jitianu@oxygenxml.com

@AlexJitianu

Agenda

- What is *Docs as Code*
- Choosing a text markup language
- Version Control
- Continuous integration and delivery
- Example of a working *Docs as Code* setup

Docs as Code

- Refers to a philosophy that you should be writing documentation with the same **tools** and **workflows** as code:
  - Version Control
  - Collaboration and Review Process
  - Automated Tests, Builds, Delivery
  - Issue Trackers
  - Plain Text Markup

Choosing a Suitable Text Markup

- Markdown
- Asciidoc
- ***XML***
  - ***Docbook***
  - ***DITA***

Why should I use a Version Control?

- Storing versions
  - *The basics of version control is the ability to save changes made to files, whilst retaining the changes from all previous versions.*
- Collaboration and review
- Understand what happened

Which Version Control System?

- Git
- Mercurial
- Subversion (svn)

In the Cloud or on Premise

- On premise Git repositories
  - *GitLab Community Edition (CE) is an* ***open source*** *end-to-end software development platform with built-in version control, issue tracking, code review, CI/CD, and more. Self-host GitLab CE on your own servers, in a container, or on a cloud provider.*
- Web-based Git repositories
  - GitLab [https://about.gitlab.com/pricing]
  - GitHub [https://github.com/pricing]
  - Bitbucket [https://bitbucket.org/product/pricing]

Continuous Integration (CI)

- Changes are validated as soon as they are committed, by creating a build and running *automated tests*.
  - *Helps avoid "integration hell" where the software works on individual developers' machines, but it fails when all developers combine (or "integrate") their code*
- Puts a great emphasis on *testing automation.*

What can we automate for a documentation project?

- Quality checks
  - Business rules (Schematron, Vale)
  - Integrity checks (Validate and Check for Completeness)
- Reuse metrics
- Publishing pipelines

Continuous Delivery (CD)

- The goal of CD is to make sure the software is always ready to go to production
  - You have automated your release process
  - You can deploy your application at any point of time by clicking on a button.

CI/CD Platforms/Servers

- Jenkins
- Travis CI
- Netlify
- GitLab CI/CD

Editing tools

- Any text processor
- GitLab, GitHub built-in text editors
- Commercial XML editors
  - Oxygen Web Author
  - Oxygen XML Editor

Collaboration

- Using version control (Git)
- GitHub pull requests
  - https://github.com/features/code-review/
- Using dedicated solutions (Oxygen Content Fusion, Oxygen XML Web Author)

DEMO TIME

- Oxygen Content Fusion in action
- Oxygen Web Author in action

Issue Tracking

- GitHub Issue Tracker
  - https://github.com/oxygenxml/userguide/issues
- GitLab Issue Tracker
  - https://gitlab.com/jitianualex83/my-test-project/issues
- Atlassian Jira
  - https://www.atlassian.com/ro/software/jira

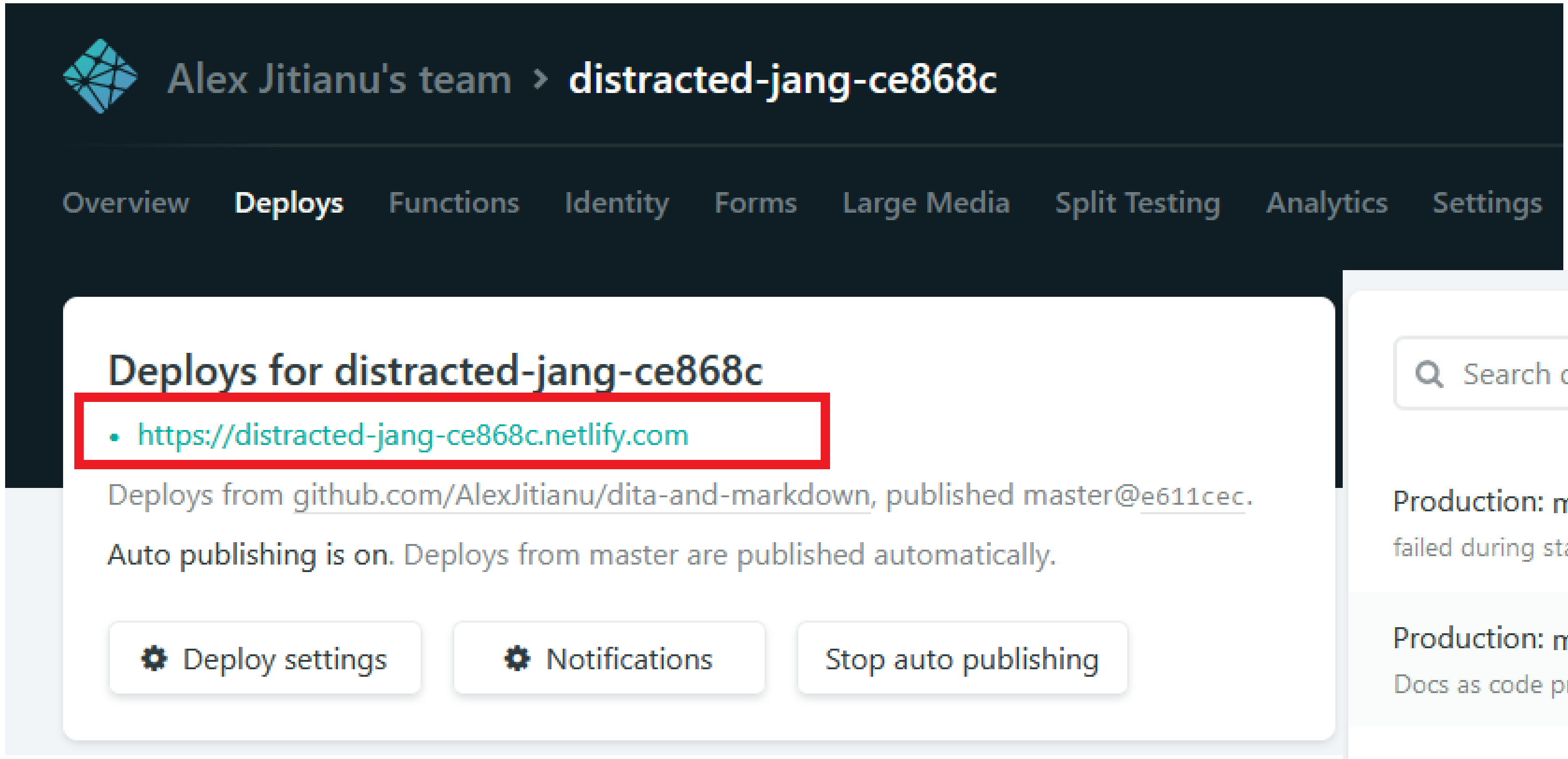
Proposed *Docs as Code* setup

- Text Markup: *DITA + Markdown*
- Version control: *GitHub*
- Issues tracker: *GitHub*
- CI/CD : *Netlify + SonarCloud*
- Collaboration: *Oxygen Web Author* [links in published output]

What can we automate for a documentation project?

- Quality checks
  - Business rules (Schematron, Vale)
  - Integrity checks (Validate and Check for Completeness)
- Reuse metrics
- Publishing pipelines

Netlify


Alex Jitianu's team › distracted-jang-ce868c

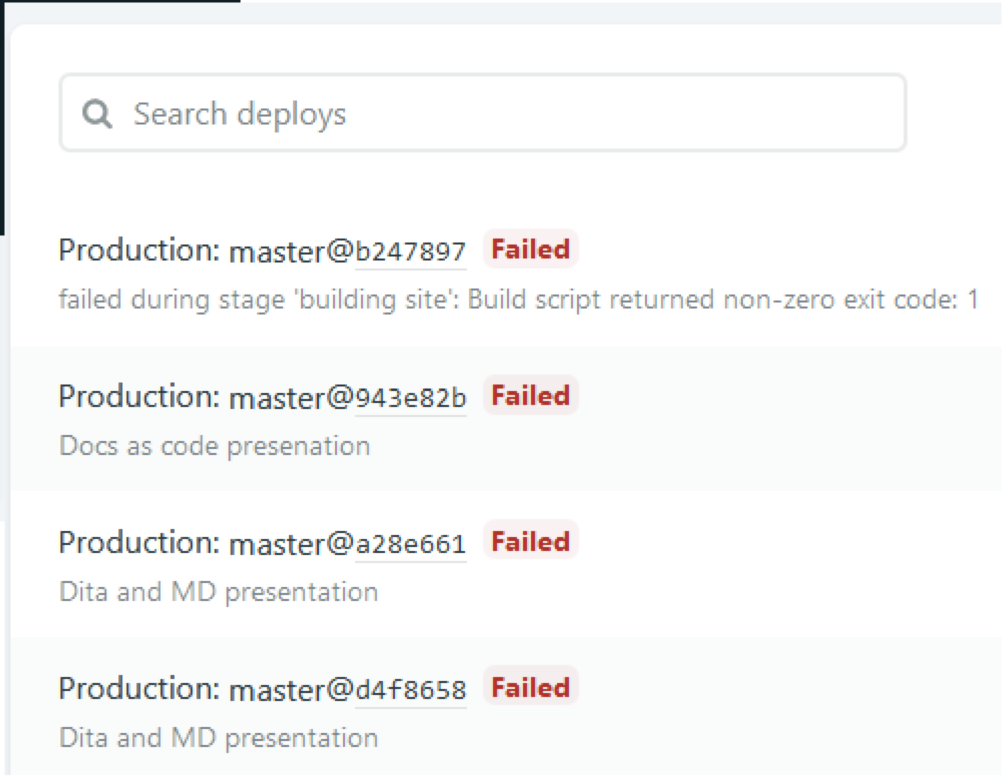
Deploys for distracted-jang-ce868c

• https://distracted-jang-ce868c.netlify.com

Deploys from github.com/AlexJitianu/dita-and-markdown, published master@e611cec.

Auto publishing is on. Deploys from master are published automatically.

A platform offering:
- Hosting (free!)
- Continuous Deployment from a Git Repo



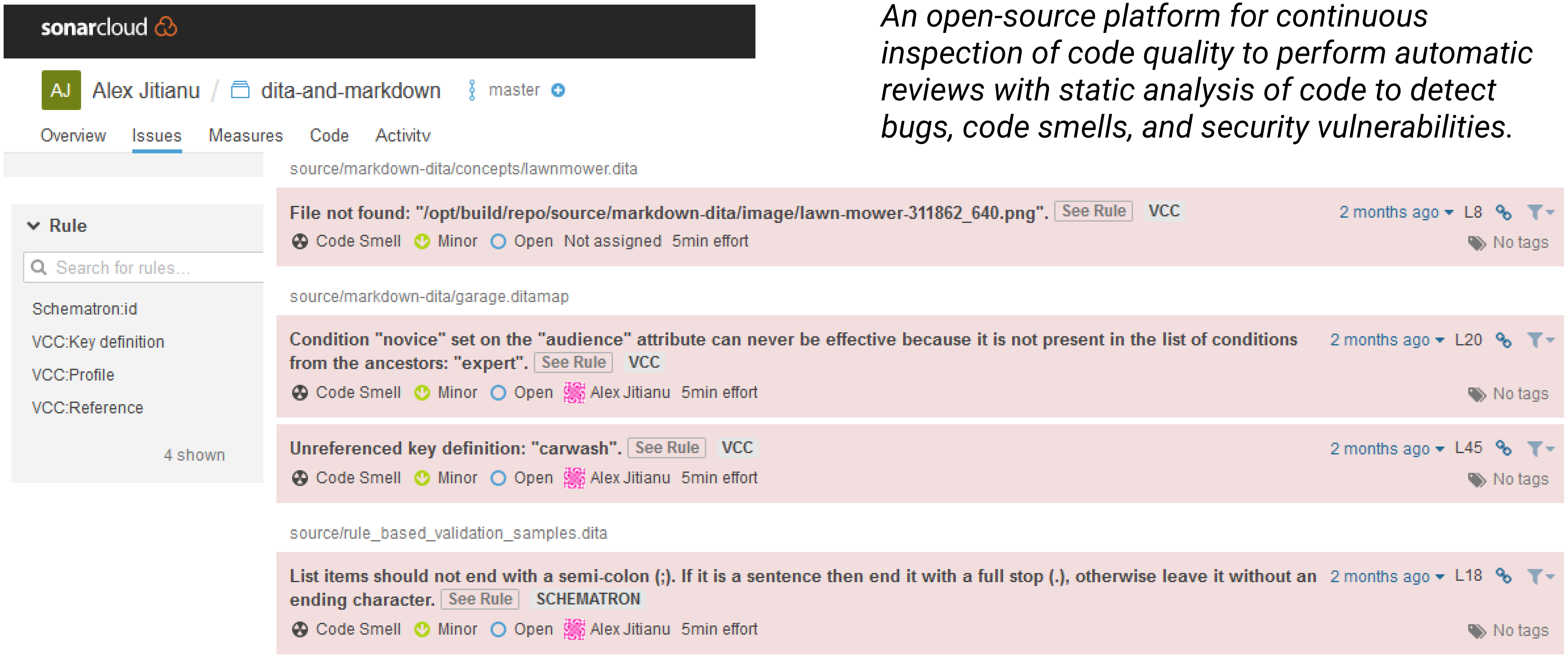
Production: master@b247897 Failed
failed during stage 'building site': Build script returned non-zero exit code: 1

Production: master@943e82b Failed
Docs as code presenation

Production: master@a28e661 Failed
Dita and MD presentation

Production: master@d4f8658 Failed
Dita and MD presentation

SonarCloud



source/markdown-dita/concepts/lawnmower.dita

File not found: "/opt/build/repo/source/markdown-dita/image/lawn-mower-311862_640.png". See Rule VCC 2 months ago L8
Code Smell Minor Open Not assigned 5min effort No tags

source/markdown-dita/garage.ditamap

Condition "novice" set on the "audience" attribute can never be effective because it is not present in the list of conditions from the ancestors: "expert". See Rule VCC 2 months ago L20
Code Smell Minor Open Alex Jitianu 5min effort No tags

Unreferenced key definition: "carwash". See Rule VCC 2 months ago L45
Code Smell Minor Open Alex Jitianu 5min effort No tags

source/rule_based_validation_samples.dita

List items should not end with a semi-colon (;). If it is a sentence then end it with a full stop (.), otherwise leave it without an ending character. See Rule SCHEMATRON 2 months ago L18
Code Smell Minor Open Alex Jitianu 5min effort No tags

An open-source platform for continuous inspection of code quality to perform automatic reviews with static analysis of code to detect bugs, code smells, and security vulnerabilities.

DEMO TIME

- Deploy to Netlify our *docs-as-code* solution for DITA and Markdown
    - https://github.com/AlexJitianu/dita-meets-markdown

SonarCloud Configuration

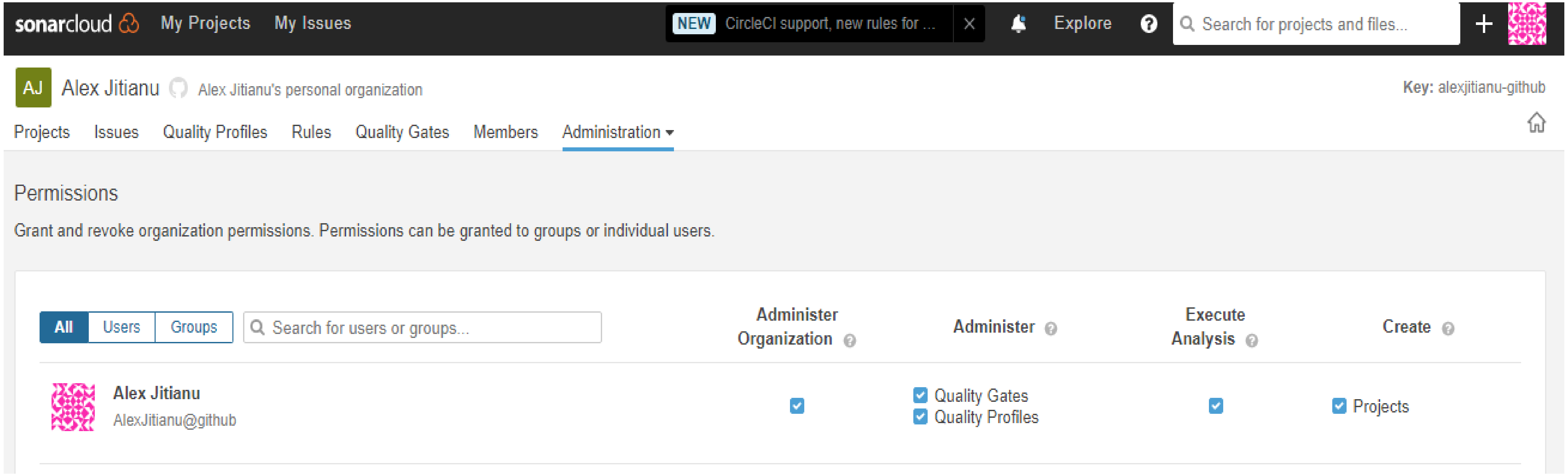
```
# These details need to be filled in on a per-project basis
sonar.organization={sonarcloud.organization.name}
sonar.login={sonarcloud.auth.token}
sonar.projectKey={unique.project.name} !!!!!

# Configration
sonar.sources=.
sonar.host.url=https://sonarcloud.io
sonar.exclusions=bin/**,scripts/**, demo-files/**
sonar.externalIssuesReportPaths=bin/tmp/sonar-schematron.json, bin/tmp-vcc/vcc-result-sonar.json
```

DEMO TIME

- Setup SonarCloud
  - Create account
    - https://sonarcloud.io/about GitHub
  - Create organization
    - https://sonarcloud.io/account/organizations
  - Set permissions (*Execute Analysis*)
    - https://sonarcloud.io/organizations/{organization-name}/permissions
  - Generate Token
    - https://sonarcloud.io/account/security/

SonarCloud permissions



sonarcloud My Projects My Issues NEW CircleCI support, new rules for ... Explore Search for projects and files...

AJ Alex Jitianu Alex Jitianu's personal organization

Key: alexjitianu-github

Projects Issues Quality Profiles Rules Quality Gates Members Administration

Permissions

Grant and revoke organization permissions. Permissions can be granted to groups or individual users.

All Users Groups Search for users or groups...

	Administer Organization	Administer	Execute Analysis	Create
Alex Jitianu AlexJitianu@github	✓	✓ Quality Gates ✓ Quality Profiles	✓	✓ Projects

THANK YOU!

Any questions?

Alex Jitianu
alex_jitianu@oxygenxml.com
@AlexJitianu