

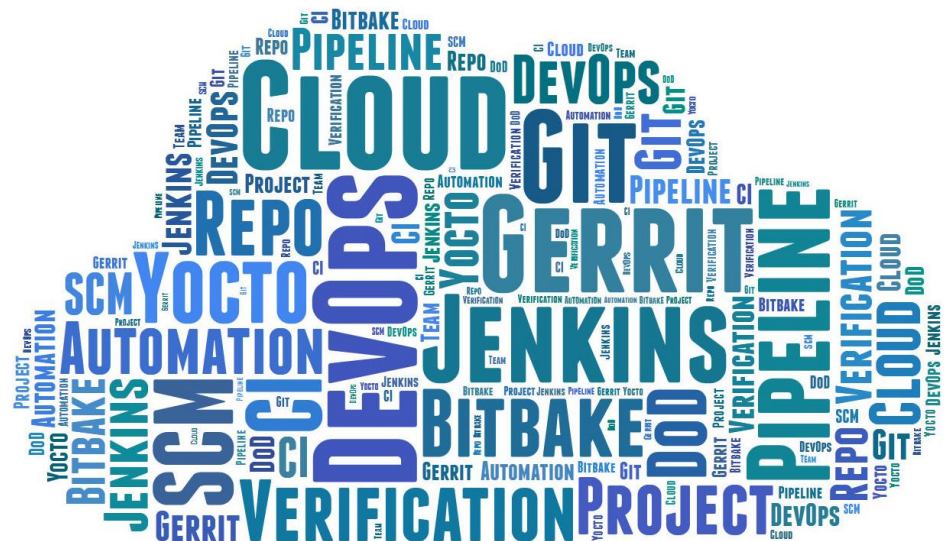
# CI with Gerrit and Jenkins in a way you would never imagine!

Miłosz Tyborowski

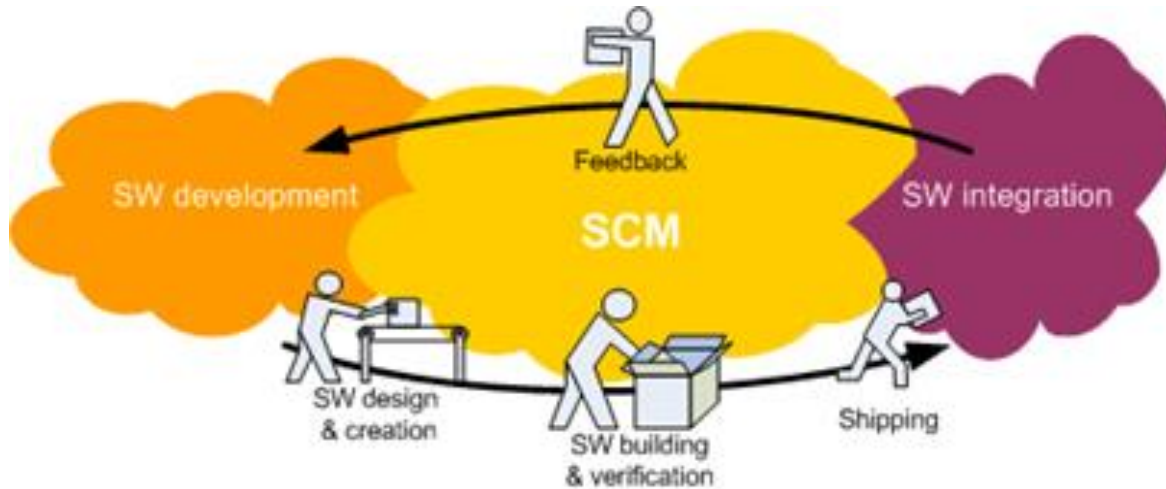
12.02.2019

# Agenda

- SCM roles overview
- Project Description
- Context & Constraints
- Jenkinsfile
- Two level build process
- Cloud
- items to do

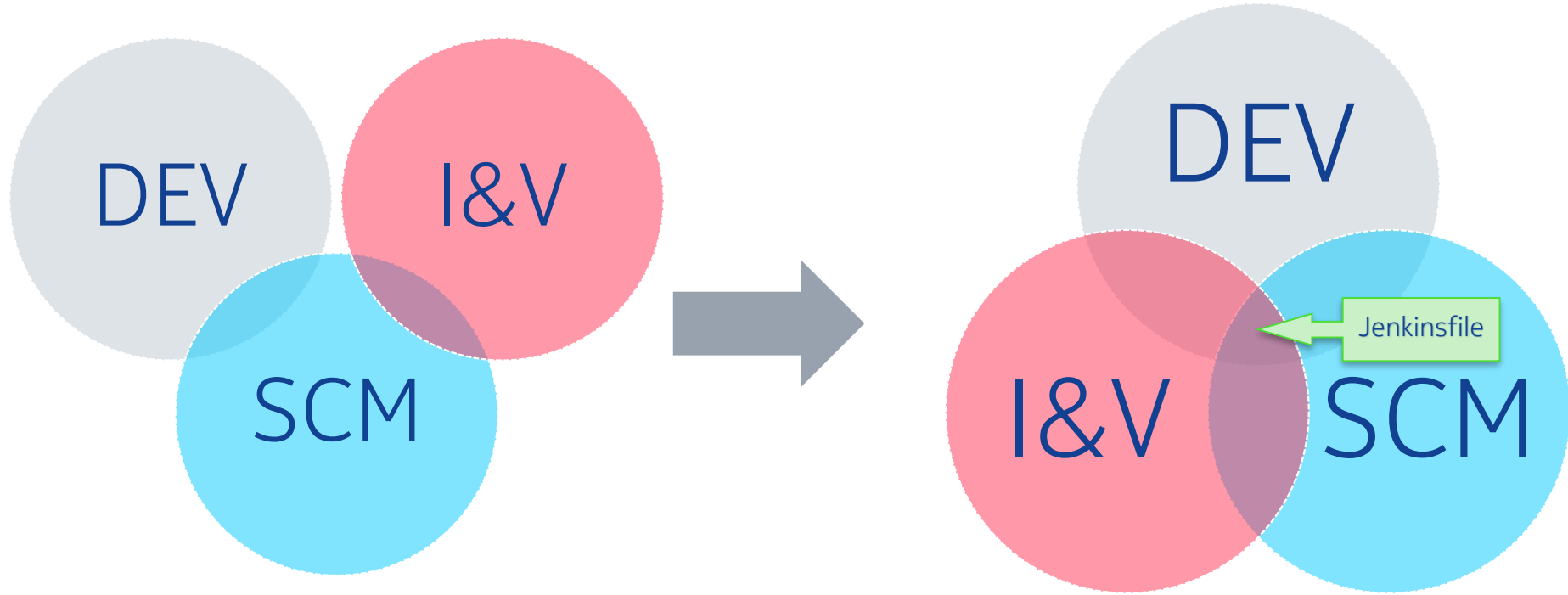


# SCM – Software Configuration Management - roles overview

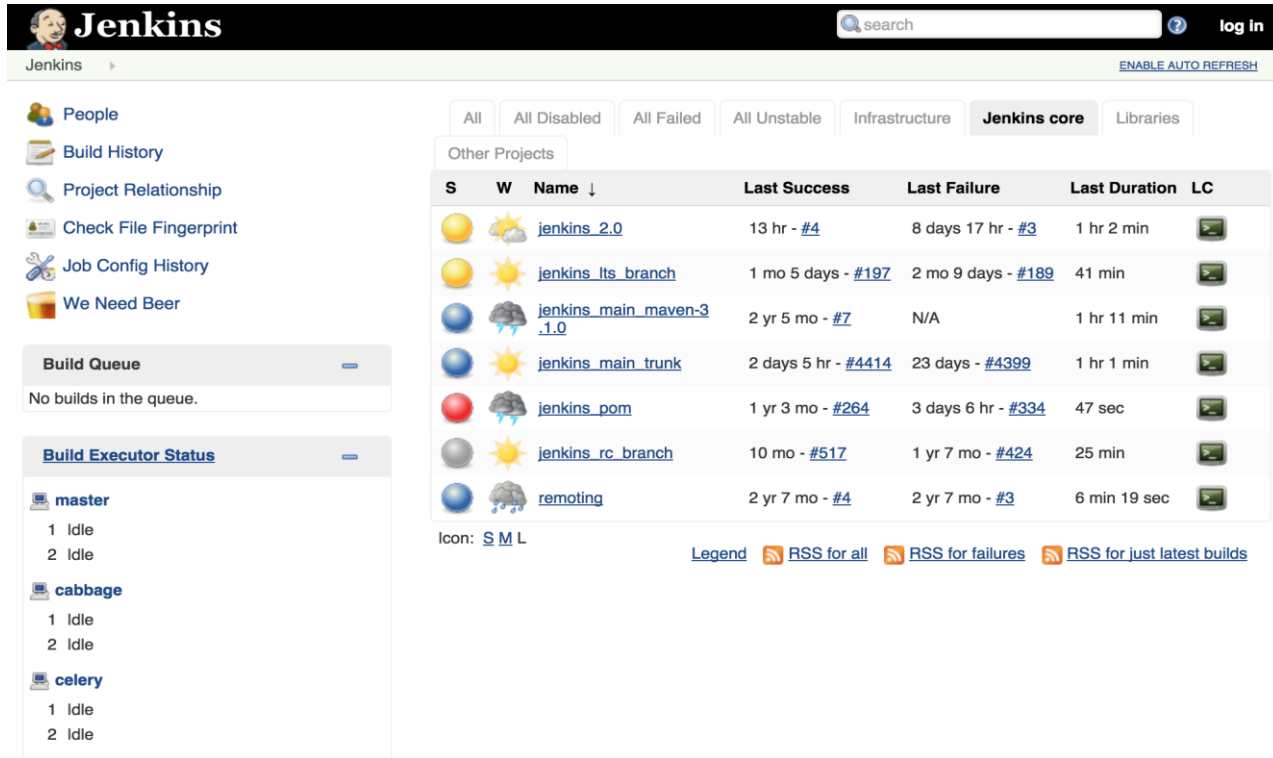


# SCM roles overview

Are you ready for DevOps?



# Context & Constraints



The screenshot shows the Jenkins web interface. The top navigation bar includes the Jenkins logo, a search bar, and a 'log in' link. Below the navigation bar, there's a sidebar on the left with links to 'People', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Job Config History', and 'We Need Beer'. The main content area displays a table of build projects under the 'Jenkins core' tab. The table has columns for 'S' (Status), 'W' (Weather icon), 'Name', 'Last Success', 'Last Failure', 'Last Duration', and 'LC' (Link icon). The table lists several projects, including 'jenkins\_2.0', 'jenkins\_its\_branch', 'jenkins\_main\_maven-3.1.0', 'jenkins\_main\_trunk', 'jenkins\_pom', 'jenkins\_rc\_branch', and 'remoting'. Below the table, there's a legend for the status icons and links for RSS feeds.

**Jenkins** search log in

Jenkins [ENABLE AUTO REFRESH](#)

[People](#) [Build History](#) [Project Relationship](#) [Check File Fingerprint](#) [Job Config History](#) [We Need Beer](#)

**Build Queue** —  
No builds in the queue.

**Build Executor Status** —

- master**
  - 1 Idle
  - 2 Idle
- cabbage**
  - 1 Idle
  - 2 Idle
- celery**
  - 1 Idle
  - 2 Idle

**Other Projects**

S	W	Name ↓	Last Success	Last Failure	Last Duration	LC
🟡	☀️	<a href="#">jenkins_2.0</a>	13 hr - <a href="#">#4</a>	8 days 17 hr - <a href="#">#3</a>	1 hr 2 min	🔗
🟡	☀️	<a href="#">jenkins_its_branch</a>	1 mo 5 days - <a href="#">#197</a>	2 mo 9 days - <a href="#">#189</a>	41 min	🔗
🟢	☁️	<a href="#">jenkins_main_maven-3.1.0</a>	2 yr 5 mo - <a href="#">#7</a>	N/A	1 hr 11 min	🔗
🟢	☀️	<a href="#">jenkins_main_trunk</a>	2 days 5 hr - <a href="#">#4414</a>	23 days - <a href="#">#4399</a>	1 hr 1 min	🔗
🔴	☁️	<a href="#">jenkins_pom</a>	1 yr 3 mo - <a href="#">#264</a>	3 days 6 hr - <a href="#">#334</a>	47 sec	🔗
🔴	☀️	<a href="#">jenkins_rc_branch</a>	10 mo - <a href="#">#517</a>	1 yr 7 mo - <a href="#">#424</a>	25 min	🔗
🔴	☁️	<a href="#">remoting</a>	2 yr 7 mo - <a href="#">#4</a>	2 yr 7 mo - <a href="#">#3</a>	6 min 19 sec	🔗

Icon: [S](#) [M](#) [L](#)

[Legend](#) [RSS for all](#) [RSS for failures](#) [RSS for just latest builds](#)

# Testing environment

## What we had

Readability was poor

There was no easily available feedback

No Notification

No automated feedback for review proces

No way of verifying single delivery

Jobs running serially due to infrastructure underneath

Build History		<a href="#">trend</a>
find		x
	#841	Mar 2, 2018 1:18 PM
	#840	Mar 2, 2018 11:32 AM
	#839	Mar 2, 2018 9:31 AM
	#838	Mar 2, 2018 6:31 AM
	#837	Mar 2, 2018 3:31 AM
	#836	Mar 2, 2018 12:31 AM
	#835	Mar 1, 2018 9:31 PM
	#834	Mar 1, 2018 6:31 PM
	#833	Mar 1, 2018 3:31 PM
	#832	Mar 1, 2018 12:31 PM
	#831	Mar 1, 2018 9:31 AM
	#830	Mar 1, 2018 6:31 AM
	#829	Mar 1, 2018 3:31 AM
	#828	Mar 1, 2018 12:31 AM
	#827	Feb 28, 2018 9:31 PM
	#826	Feb 28, 2018 6:31 PM
	#825	Feb 28, 2018 3:31 PM
	#824	Feb 28, 2018 12:31 PM
	#823	Feb 28, 2018 9:31 AM
	#822	Feb 28, 2018 6:31 AM
	#821	Feb 28, 2018 3:31 AM
	#820	Feb 28, 2018 12:31 AM
	#819	Feb 27, 2018 9:31 PM
	#818	Feb 27, 2018 6:31 PM
	#817	Feb 27, 2018 3:31 PM
	#816	Feb 27, 2018 12:31 PM
	#815	Feb 27, 2018 9:31 AM
	#814	Feb 27, 2018 6:31 AM
	#813	Feb 27, 2018 3:31 AM
	#812	Feb 27, 2018 12:31 AM
	#811	Feb 26, 2018 9:31 PM

RSS for all RSS for failures

# Testing environment

## What we had

Our initial attempts at utilizing cloud infrastructure were a bit rusty.

We statically linked Jenkins jobs and cloud instances to perform tests and verifications

# Context & Constraints

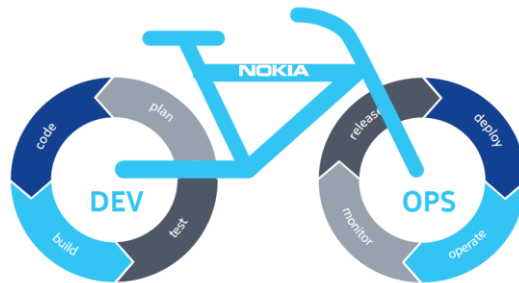
- Context:
  - Adopt More Efficient, Reliable and Uniform Practices
  - Utilize IT hosted Cloud infrastructure
  - Scripts and Tooling to verification source code
  - Relieving Jenkins; Jenkins as a schedule and workflow manager, not a system for everything (collecting reports, analyzing logs, ...)
- Constraints:
  - Planning and discussions of common practices and implementations must include stakeholders from all collaborating organizations. The investment in time is likely to be significant and needs to be accounted for in the project planning.



# Pipeline 2.0

## Project Description

- The aim of the "Pipeline 2.0" project is to introduce a uniform and transparent code verification framework
- Continuous and automated monitoring and collection of feedback
- Workflow automation (e.g. lifecycle management)
- Continuous testing, delivery and deployment
- Renewing SW architecture to be cloud-native and modularized



A wide-angle photograph of a suspension bridge spanning a deep, dense forest. The bridge is made of wood with metal railings. A group of people, many wearing bright yellow safety vests, are walking across the bridge. The forest is lush with green trees, and the background is slightly hazy. The text "Let's get ready to rumble!" is overlaid in white, sans-serif font across the middle of the image.

Let's get ready to rumble!

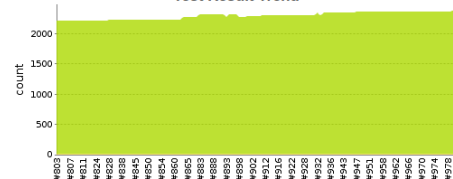
# Testing environment

Then came the pipelines...



Stage View

Test Result Trend



[\(just show failures\)](#) [enlarge](#)

		Fetch SCM scripts	Prepare environment	Download sources	Apply change	Execute tests	Execute CTC	Execute Valgrind	Execute clang static analysis	Build targets	Prepare statistics	Cleanup	Send notifications
Average stage times: (Average full run time: ~1h 0min)		9s	495ms	22s	3s	44min 58s	1min 19s	24ms	23ms	15min 43s	113ms	29s	3s
#983	Mar 02 12:15 No Changes	9s	407ms	23s	4s	49min 41s	26ms	26ms	25ms	15min 57s	114ms	31s	2s
#982	Mar 02 12:13 No Changes	10s	399ms	22s	4s	51min 29s	26ms	26ms	24ms	17min 7s	115ms	24s	4s
#981	Mar 02 11:48 No Changes	8s											



# Testing environment

## What we have

And together with Pipelines 1st\_lvl we got:

- Good readability
- relatively fast verification
- Quick and automatic feedback
- Code-review partly automated

Not so long after introducing first Pipelines we managed to get them to run **parallelly**.

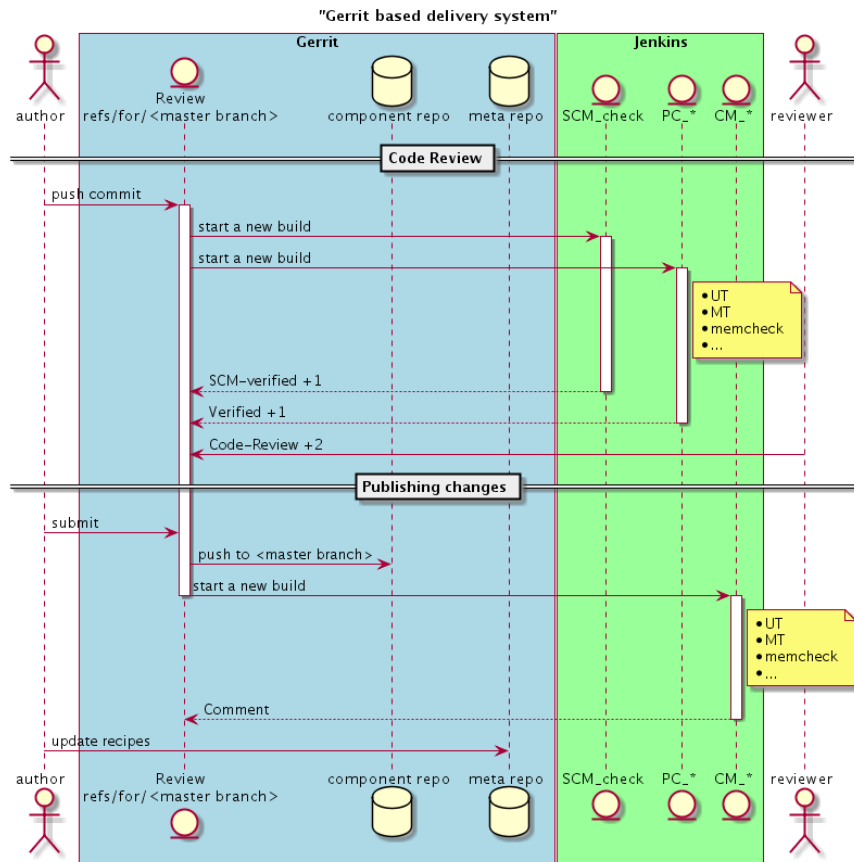
Thus making **verification** and feedback even **faster**.



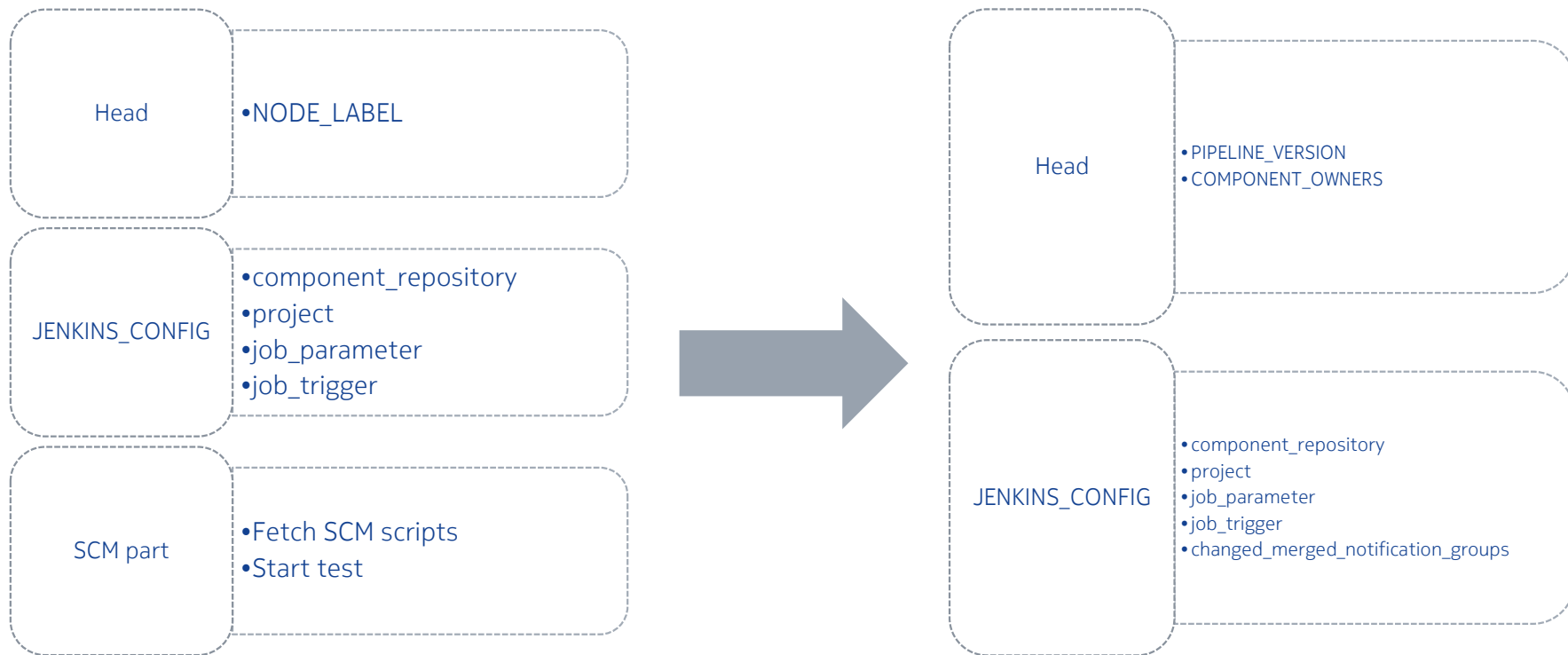
# Jenkinsfile

## What can be determined there?

- Jenkinsfile is a text file that contains the definition of a Jenkins Pipeline and is checked into source control.

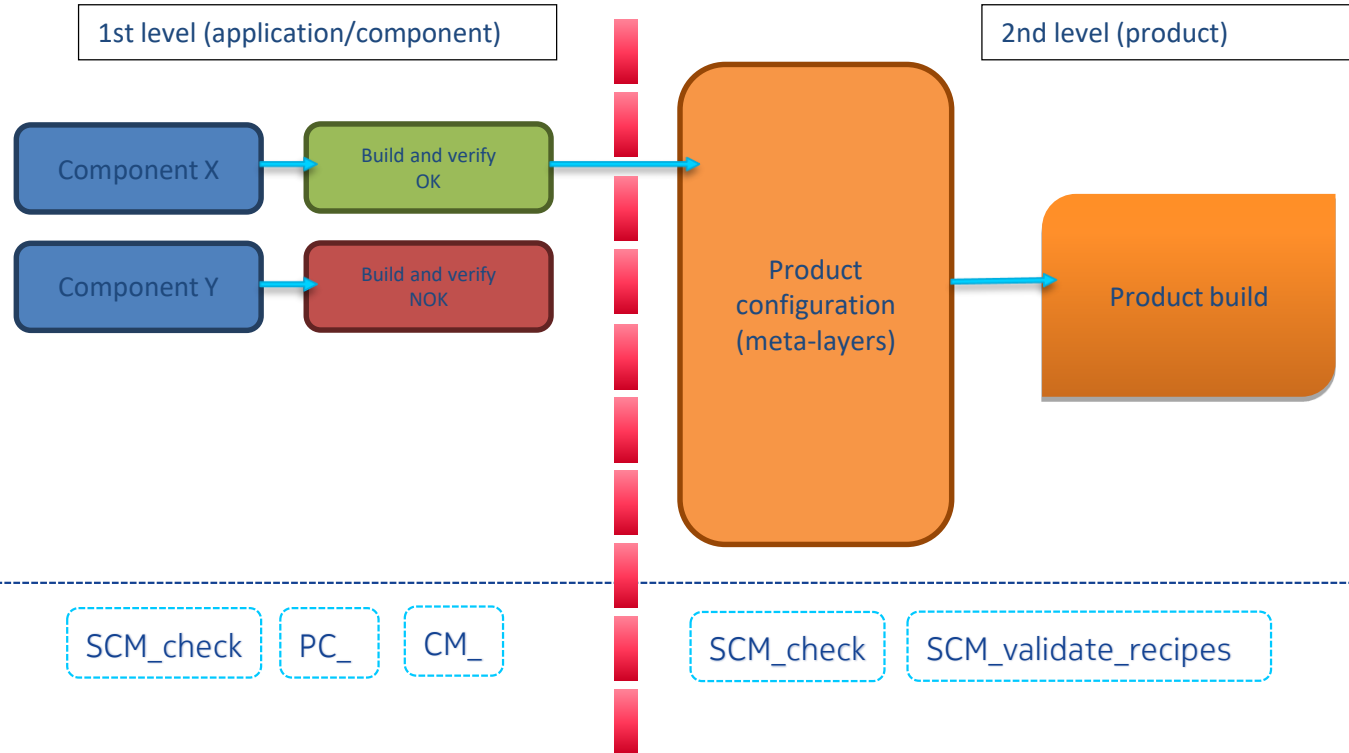


# Jenkinsfile Structure



# Two level build process

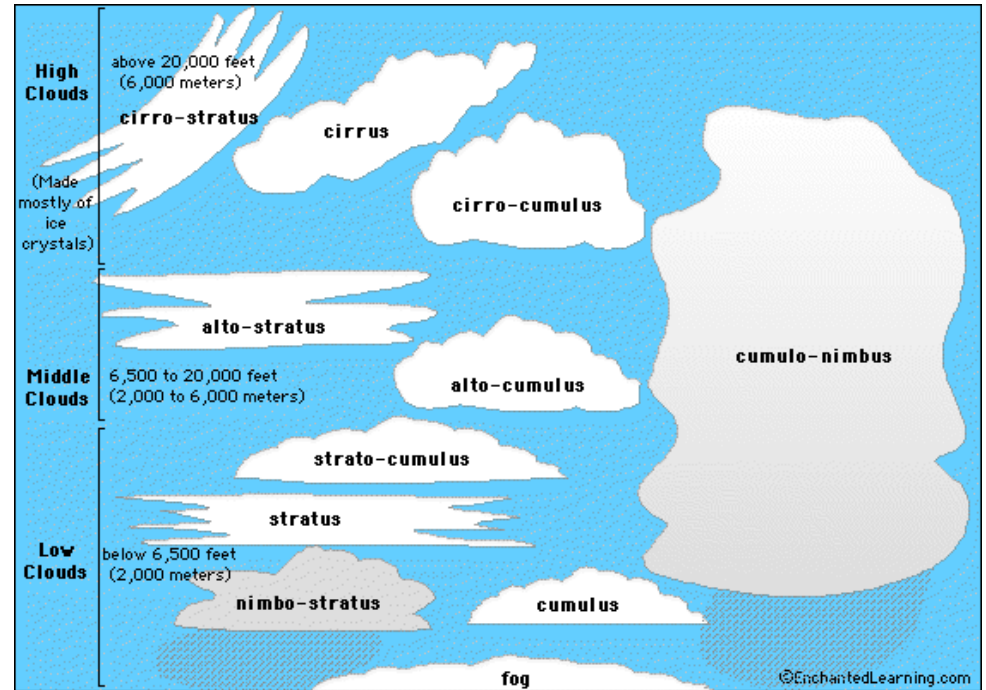
## Application/component and product



# Testing environment

## Cloud

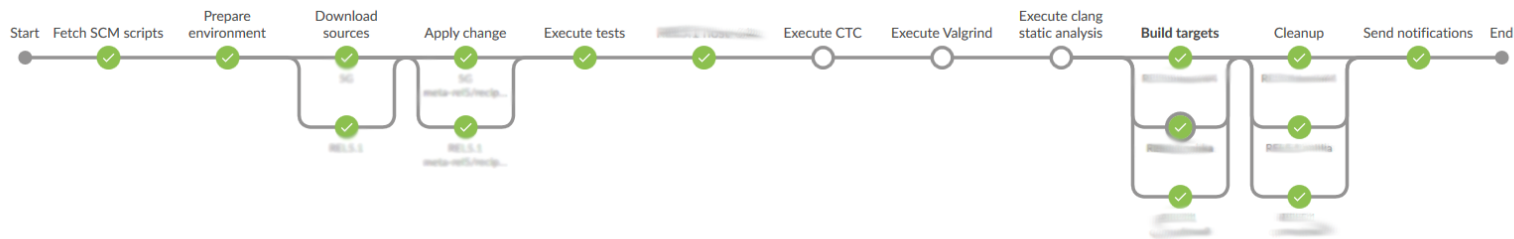
Cloud instance is basically a dynamically allocated build server that we, as RF SCM team administer, provision and maintain.





# Testing environment

## Blue Ocean



Steps

✓	> Print Message	<1s
✓	> [WRAPPER] coop: [test_name:..., category:compiling] — Print Message	11s
✓	> [BITBAKE] bitbake -c compile	5m 40s
✓	> Start post Bitbake actions — Print Message	<1s
✓	> End post Bitbake actions — Print Message	<1s
✓	> Print Message	3s
✓	> true — [COOP] Gerrit status	4s
✓	> [COOP] Test result	6s
✓	> [COOP] Test category	2s

## *Quo vadis?*

items to do

- clouds everywhere
- tests (faster, better, more)
- specified framework
- visibility on every level
- DoD automation
- monitoring everything, everywhere, everywhen

**NOKIA**

# Copyright and confidentiality

The contents of this document are proprietary and confidential property of Nokia. This document is provided subject to confidentiality obligations of the applicable agreement(s).

This document is intended for use of Nokia's customers and collaborators only for the purpose for which this document is submitted by Nokia. No part of this document may be reproduced or made available to the public or to any third party in any form or means without the prior written permission of Nokia. This document is to be used by properly trained professional personnel. Any use of the contents in this document is limited strictly to the use(s) specifically created in the applicable agreement(s) under which the document is submitted. The user of this document may voluntarily provide suggestions, comments or other feedback to Nokia in respect of the contents of this document ("Feedback").

Such Feedback may be used in Nokia products and related specifications or other documentation. Accordingly, if the user of this document gives Nokia Feedback on the contents of this document, Nokia may freely use, disclose, reproduce, license, distribute and otherwise commercialize the feedback in any Nokia product, technology, service, specification or other documentation.

Nokia operates a policy of ongoing development. Nokia reserves the right to make changes and improvements to any of the products and/or services described in this document or withdraw this document at any time without prior notice.

The contents of this document are provided "as is". Except as required by applicable law, no warranties of any kind, either express or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose,

are made in relation to the accuracy, reliability or contents of this document. NOKIA SHALL NOT BE RESPONSIBLE IN ANY EVENT FOR ERRORS IN THIS DOCUMENT or for any loss of data or income or any special, incidental, consequential, indirect or direct damages howsoever caused, that might arise from the use of this document or any contents of this document.

This document and the product(s) it describes are protected by copyright according to the applicable laws.

Nokia is a registered trademark of Nokia Corporation. Other product and company names mentioned herein may be trademarks or trade names of their respective owners.