Jenkins & Rancher
(Oltre Lo) Sviluppo Software

Doina Cristina Duma, Stefano Longo

INFN – CNAF

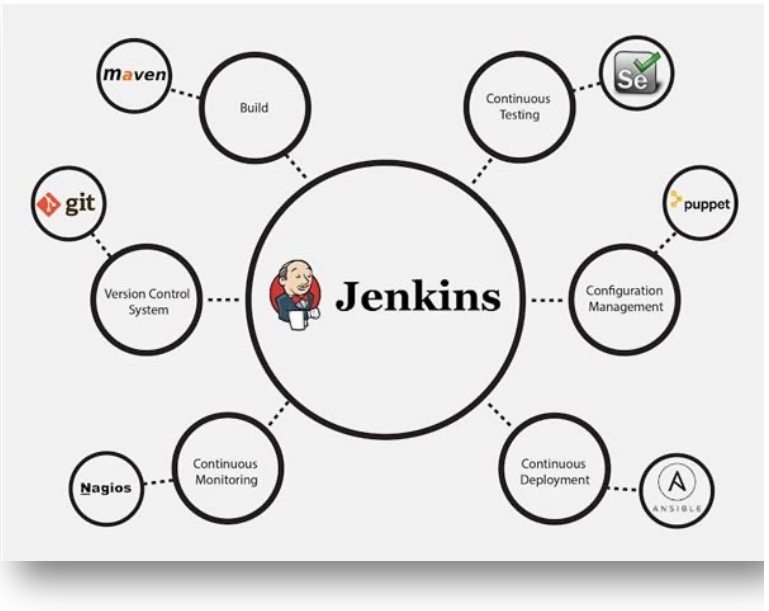
April 27th 2021, Bologna

Software Delivery – by Martin Fowler

- "Software Delivery"
  - indicate the steps from a developer finishing work on a **new feature**, to that feature being used in **production**.
- Agile software development, introduced **short cycle times and fast feedback**.
- Continuous Integration
  - encourages all members of a dev-team to integrate their work daily, instead of developing features in isolation for days or weeks.
- Devops
  - encourages software devs, ops staff, and everyone else involved in delivery to work together - avoiding hand-offs that add delays and brittleness.
- Infrastructure-As-Code
  - takes advantage of our cloud age ability to rapidly deploy and provision new servers.
- Continuous Delivery:
  - Putting all the above together => always **keeps the software product in a releasable state**, allowing fast release of features and rapid response to any failures.

Jenkins

- Jenkins
  - **leading open-source automation server**.
  - manages and controls software delivery processes throughout the entire lifecycle, including *build, document, test, package, stage, deployment, static code analysis* and much more.
  - watch for **any code changes** in places like
    - *GitHub, Bitbucket or GitLab*
  - **automatically do a build** a with tools
    - *Maven and Gradle*
  - Various platforms supported
    - Docker, Kubernetes, Linux, macOS, WAR files, Windows



History

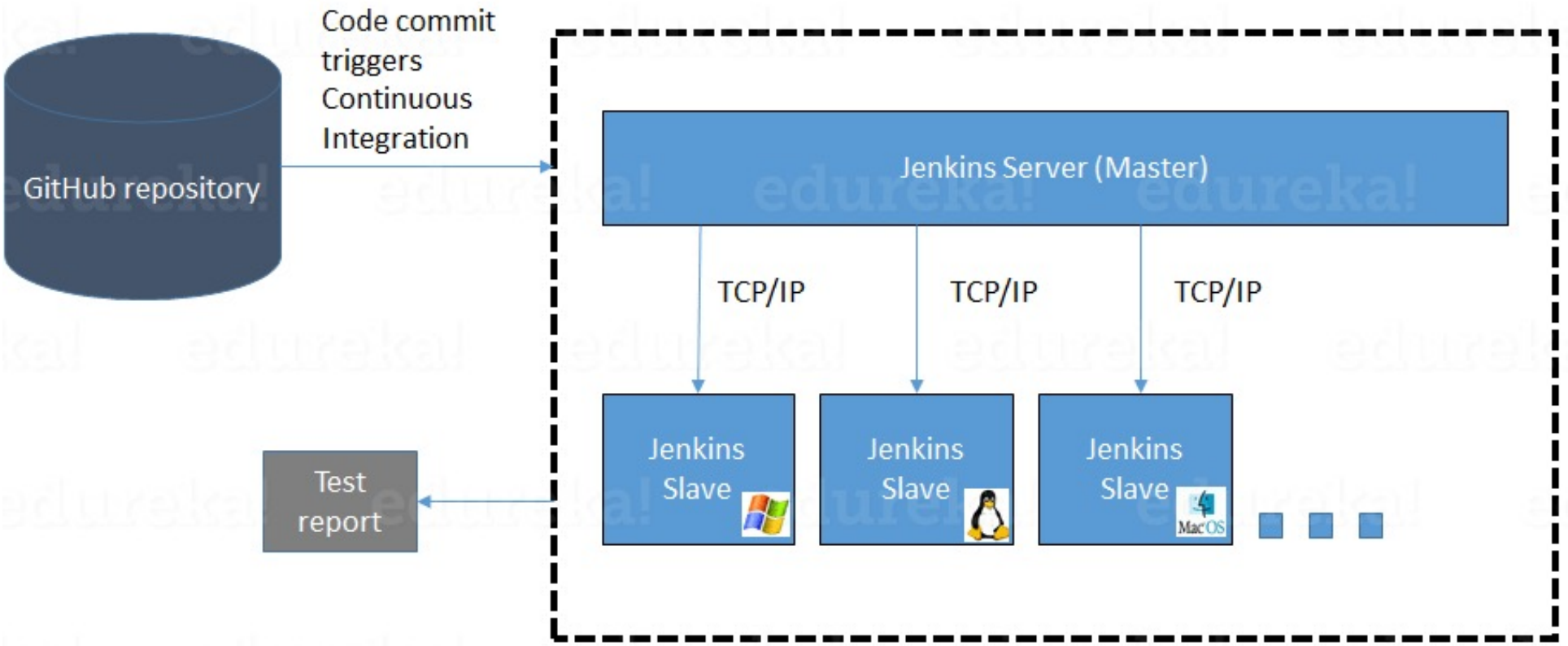
- In the beginning there was **Hudson**
  - Developed since 2004 (Sun Microsystems),
- 2010 - 2011 – dispute between Oracle (which had acquired Sun) and the independent Hudson open source community led to a fork with a name change, **Jenkins**
  - Kohsuke Kawaguchi received an O'Reilly Open Source Award for his work on the Hudson/Jenkins project.
- 2014 - Kawaguchi became CTO of CloudBees, which offers Jenkins-based continuous delivery products
- 2019 - the Continuous Delivery Foundation (CDF) was launched operating under the umbrella of the Linux Foundation
  - *developing, nurturing and promoting open source projects, best practices and industry specifications related to continuous delivery*
  - Jenkins is one of the supported projects
- 2020 - Kawaguchi announced he was moving to his new startup, Launchable.
  - officially stepping back from Jenkins, although staying on the Technical Oversight Committee of the Continuous Delivery Foundation
  - switching his role at CloudBees to an advisor.

Jenkins today

- Started as Continuous Integration tool
  - ➢ today it orchestrates the entire software delivery pipeline = Continuous Delivery
- The Jenkins community **offers more than 1,700 plugins** that enable Jenkins to integrate with virtually any tool,
- Jenkins continues to grow as the dominant solution for software process automation, continuous integration and continuous delivery and, as of **February 2018**, there are **more than 165,000 active installations** and an estimated **1.65 million users** around the world.

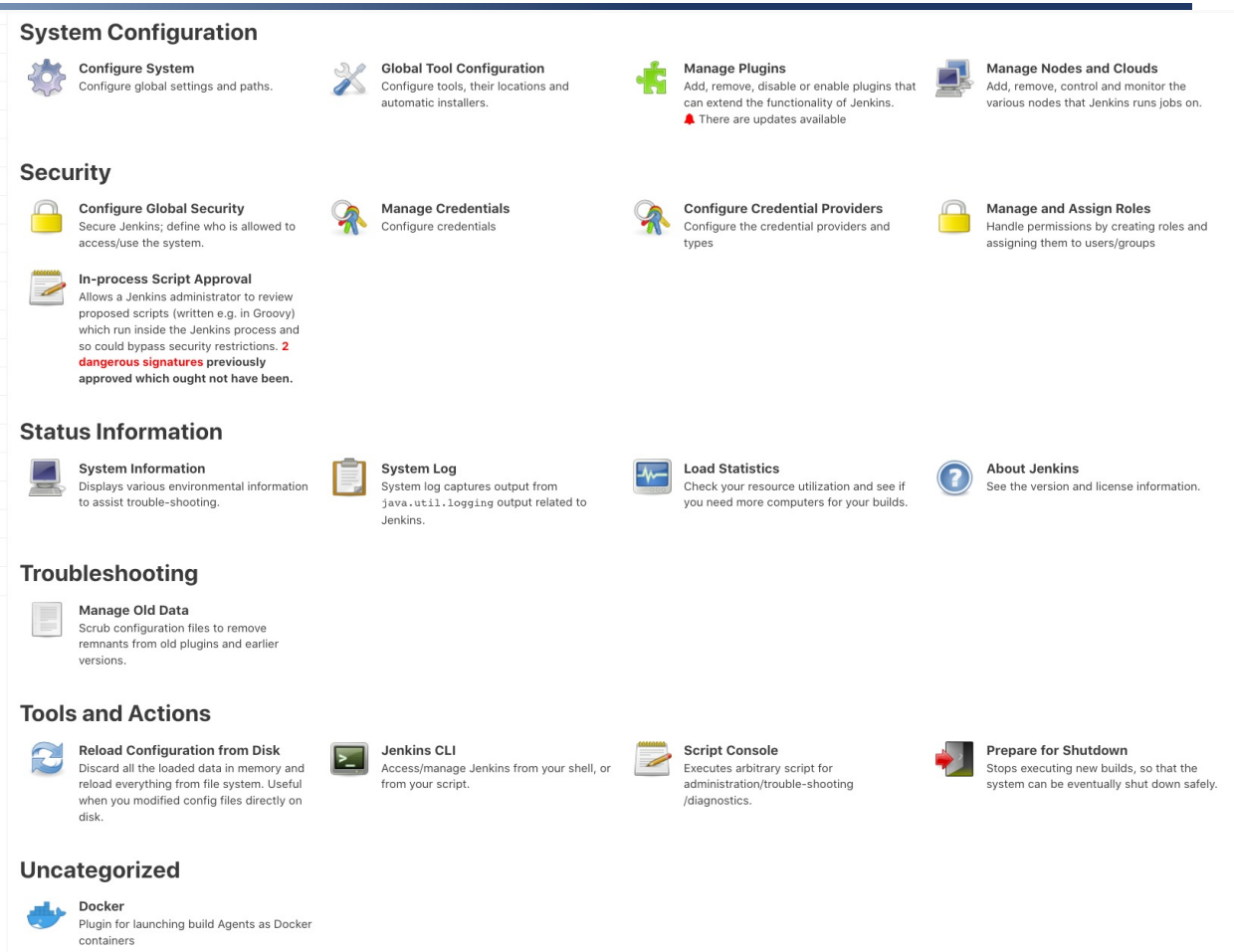

Over 1000 Jenkins Plugins
Integration with over 100 DevOps Tools
Orchestration of the DevOps Toolchain
End-to-End CD Pipeline Management
Code & Commit
Build & Config
Scan & Test
Release
Deploy
git
GitHub
Stash
SUBVERSION
PERFORCE
IBM Rational
ca technologies
SERENA
Borland StarTeam
maven
gradle
ANT
Gant
docker
CHEF
puppet labs
ANSIBLE
gerrit
sauce LABS
sonar
BlazeMeter
SOASTA
Gatling
RedLine13.com
Selenium WebDriver
uDeploy
SERENA
ca technologies
CollabNet
amazon web services
Pivotal
Azure
docker
vmware
OPENSHIFT
JBoss

How it works


Code commit
triggers
Continuous
Integration
GitHub repository
Jenkins Server (Master)
TCP/IP
TCP/IP
TCP/IP
Jenkins
Slave
Jenkins
Slave
Jenkins
Slave
MacOS
Test
report

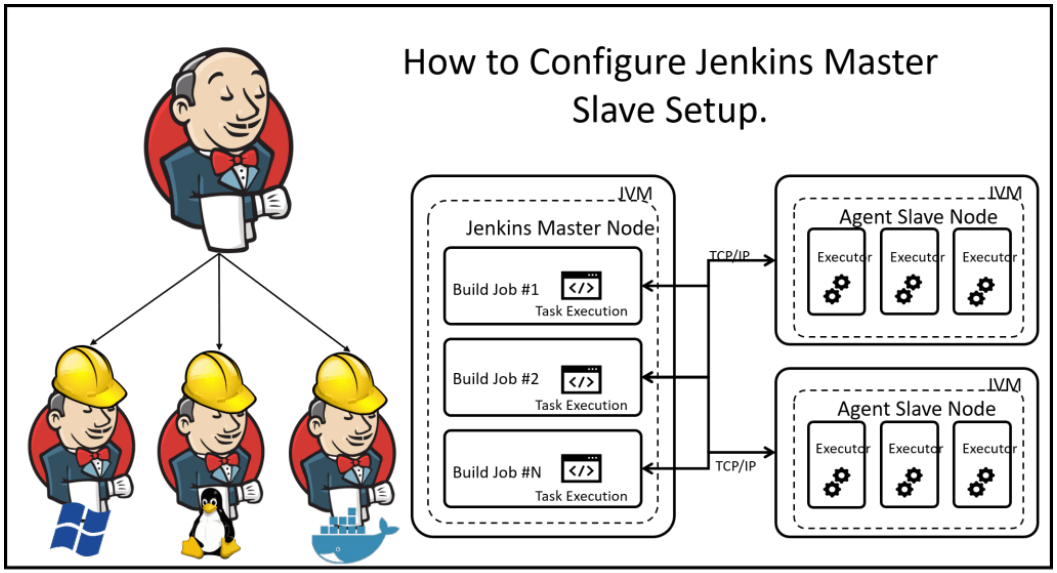
Architecture - master

- Master
  - Scheduling build jobs.
  - Dispatching builds to the slaves for the execution.
  - Monitor the slaves.
  - Recording and presenting the build results.
  - Can also execute build jobs directly
  - Easy configuration of all components and aspects



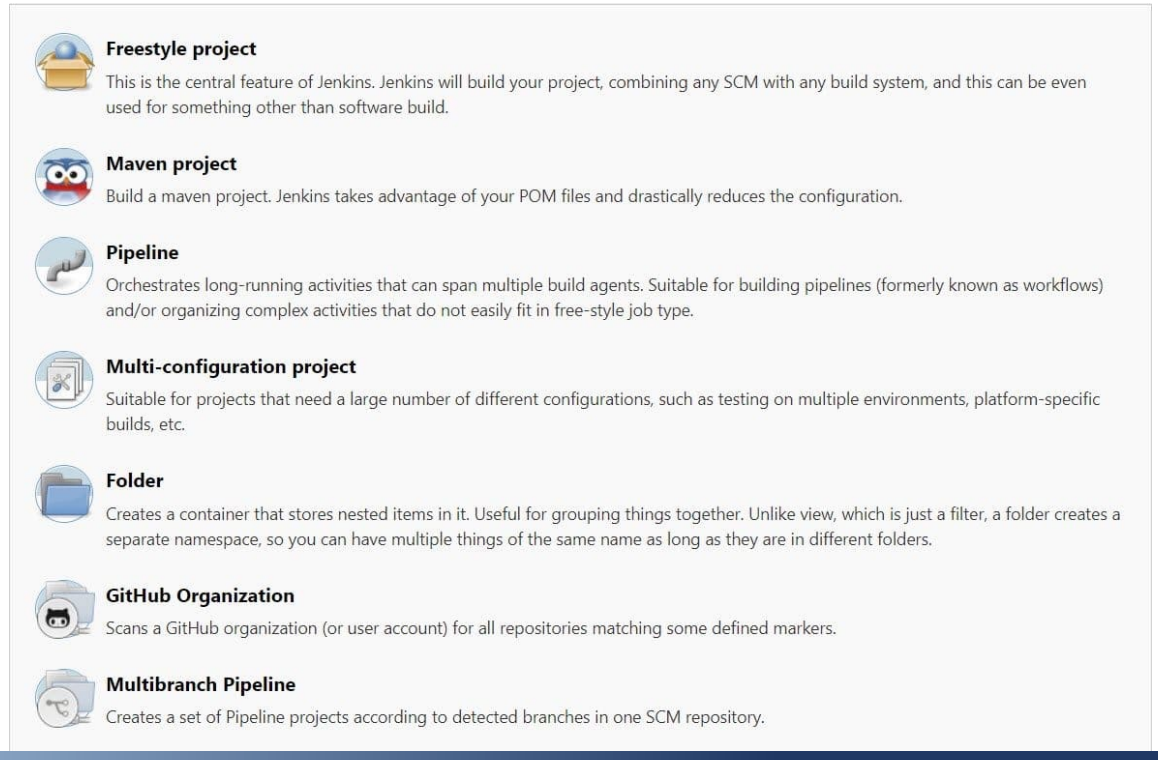
Architecture - slaves

- Slaves
  - take requests from the Jenkins Master instance.
  - Slaves can run on a variety of operating systems.
  - The job of a Slave is to do as they are told to, which involves executing build jobs dispatched by the Master.
  - One can configure a project to always run on a particular Slave machine or a particular type of Slave machine, or simply let Jenkins pick the next available Slave.

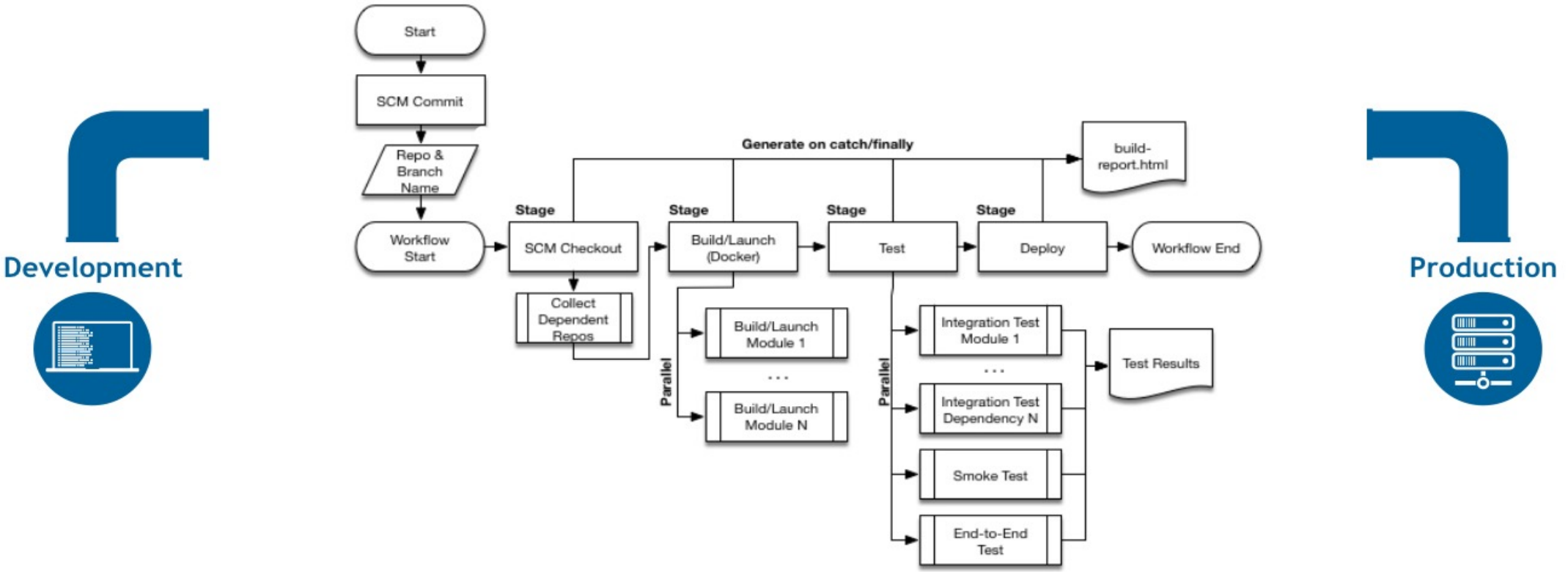


Jenkins jobs

- Jobs are the heart of *Jenkins's build* process.
- A job can be considered as a **particular task to achieve a required objective** in *Jenkins*



CD workflow



- CD scenario easily modeled *in Jenkins Pipeline*

Pipeline

- “Jenkins Pipeline (or simply **"Pipeline"** with a capital "P") is a **suite of plugins** which supports implementing and integrating *continuous delivery workflows* into Jenkins”
  - Pipeline provides an extensible set of tools for modeling simple-to-complex delivery pipelines "as code" via the Pipeline domain-specific language (DSL) syntax

Jenkinsfile (Declarative Pipeline)

```
pipeline { (1)
    agent any (2)
    stages {
        stage('Build') { (3)
            steps { (4)
                sh 'make' (5)
            }
        }
        stage('Test'){
            steps {
                sh 'make check'
                junit 'reports/**/*.xml' (6)
            }
        }
        stage('Deploy') {
            steps {
                sh 'make publish'
            }
        }
    }
}
```

Declarative specific:

- pipeline
- agent
- steps

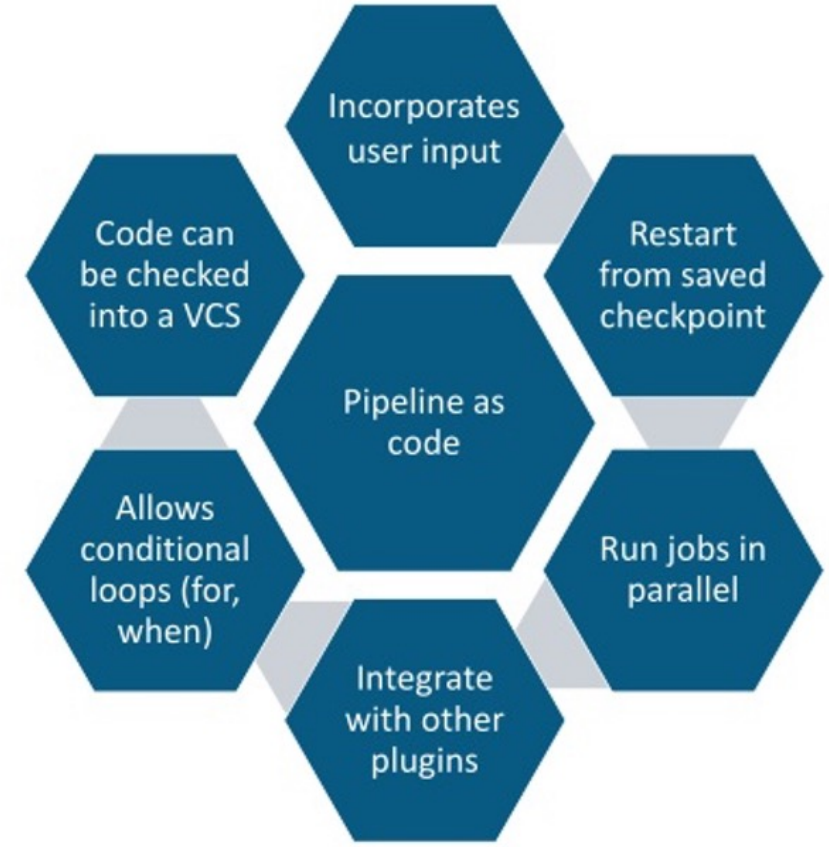
Scripted specific:

- node

Jenkinsfile (Scripted Pipeline)

```
node { (7)
    stage('Build') { (3)
        sh 'make' (5)
    }
    stage('Test') {
        sh 'make check'
        junit 'reports/**/*.xml' (6)
    }
    stage('Deploy') {
        sh 'make publish'
    }
}
```

Key features of Kenkins Pipeline



www.edureka.co/devops

Shared Libraries

- Useful to share parts of Pipelines between various projects to reduce redundancies and keep code "DRY"

```
stages {

    stage('Code fetching') {
        steps {
            checkout scm
            dir("$WORKSPACE/spring-social-oidc") {
                checkout([$class: 'GitSCM', branches: [[name: 'master']],
            }
            dir("$WORKSPACE/alien4cloud") {
                checkout([$class: 'GitSCM', branches: [[name: 'deep-dev']],
            }
        }
    }

    stage('Build Spring OIDC') {
      steps {
        dir("$WORKSPACE/spring-social-oidc") {
            MavenRun('-U clean install')
        }
      }
    }

    stage('Build local A4C (DEEP flavour)') {
      steps {
        dir("$WORKSPACE/alien4cloud") {
            MavenRun('-U clean install -Dmaven.wagon.http.ssl.insecure=true
        }
      }
    }

    stage('Build plugin') {
      steps {
        dir("$WORKSPACE/indigodc-orchestrator-plugin") {
            MavenRun('-U clean package')
        }
      }
    }
```

1.3.6 jenkins-pipeline-library / vars / MavenRun.groovy

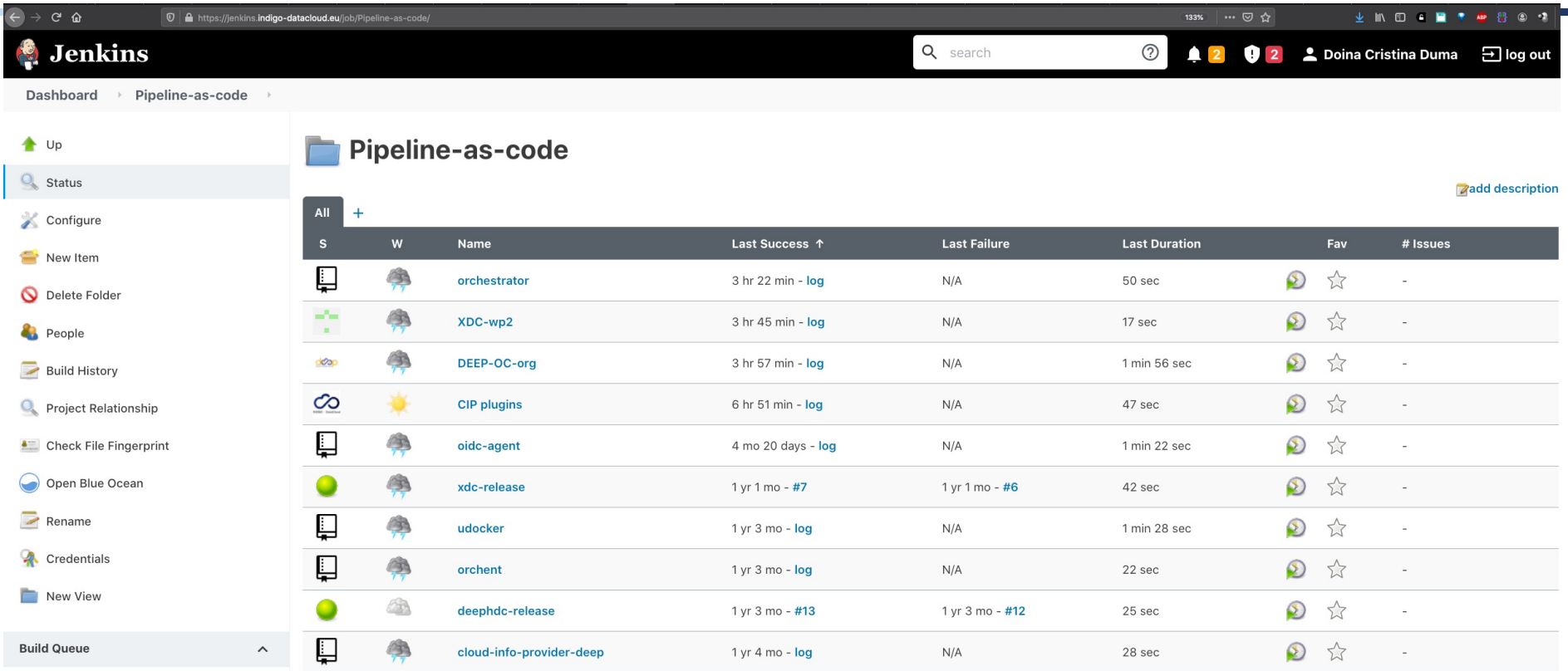
orviz Second stage of groovydoc adoption

1 contributor

29 lines (28 sloc) 786 Bytes

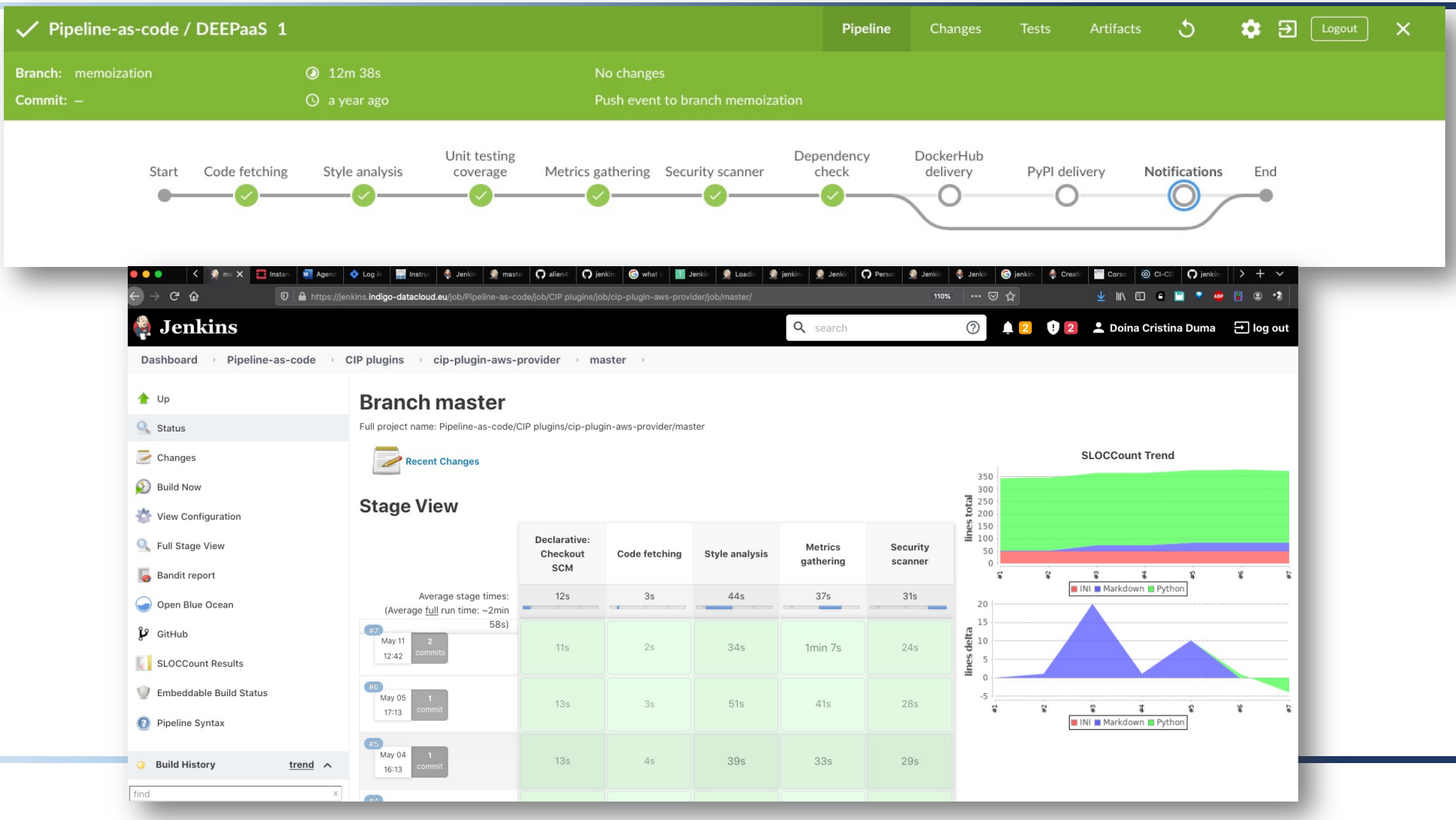
```
#!/usr/bin/groovy

/**
 * Runs either Checkstyle or Cobertura maven goals.
 *
 * @param report Report location [default]
 * @see https://plugins.jenkins.io/cobertura
 */
def call(String goal) {
    options = []
    if (goal.startsWith('checkstyle')) {
        options = [
            '-Dcheckstyle.failOnViolation=true',
            '-Dcheckstyle.console=true',
            '-Dcheckstyle.violationSeverity=warning',
            '-Dcheckstyle.config.location=google_checks.xml',
        ]
        goal = 'checkstyle:check'
    }
    else if (goal.startsWith('cobertura')) {
        options = [
            '-Dcobertura.report.format=xml'
        ]
        goal = 'cobertura:cobertura'
    }
    l_cmd = ['mvn ', options.join(' '), goal]
    cmd = l_cmd.join(' ')
    sh(script: cmd)
}
```

Jenkins - Main page

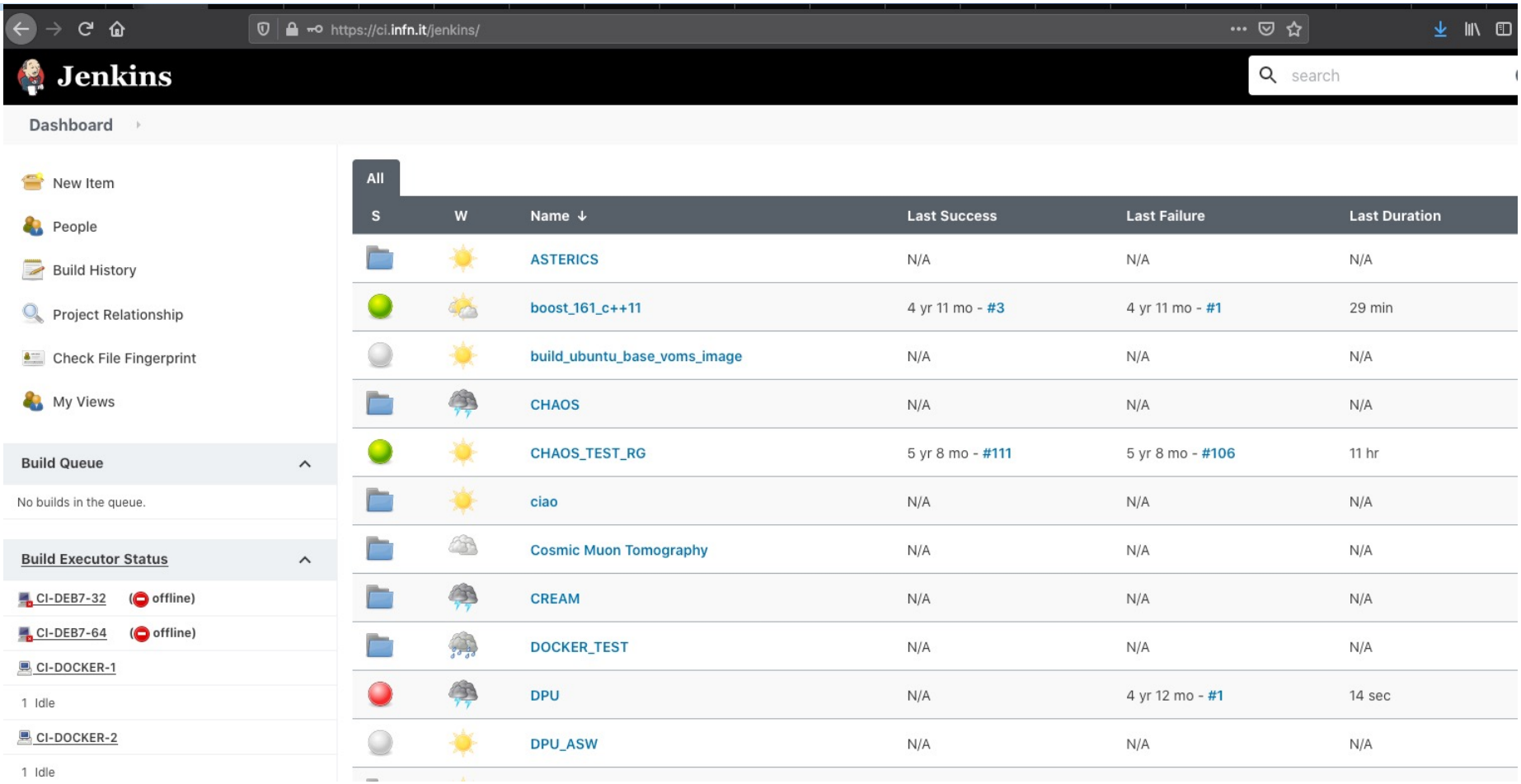


- Quick overview of builds
- “Weather” notification system
- Everything can be managed through this page

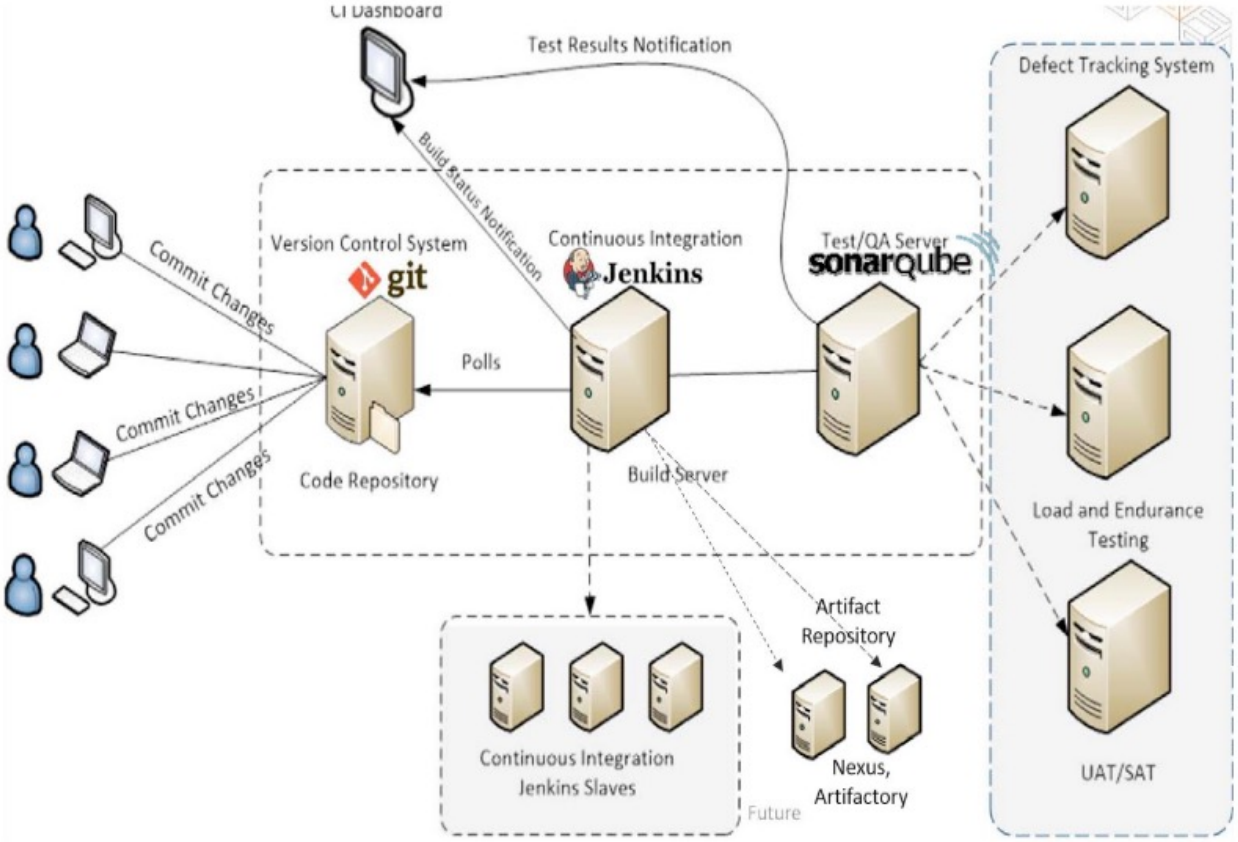
Examples


Pipeline-as-code / DEEPaaS 1
Pipeline
Changes
Tests
Artifacts
Logout
Branch: memoization
12m 38s
No changes
Commit: –
a year ago
Push event to branch memoization
Start
Code fetching
Style analysis
Unit testing coverage
Metrics gathering
Security scanner
Dependency check
DockerHub delivery
PyPI delivery
Notifications
End
Jenkins
search
Doina Cristina Duma
log out
Dashboard
Pipeline-as-code
CIP plugins
cip-plugin-aws-provider
master
Up
Status
Changes
Build Now
View Configuration
Full Stage View
Bandit report
Open Blue Ocean
GitHub
SLOCCount Results
Embeddable Build Status
Pipeline Syntax
Build History
trend
Branch master
Full project name: Pipeline-as-code/CIP plugins/cip-plugin-aws-provider/master
Recent Changes
Stage View
Declarative: Checkout SCM
Code fetching
Style analysis
Metrics gathering
Security scanner
Average stage times:
(Average full run time: ~2min 58s)
12s
3s
44s
37s
31s
#7
May 11 12:42
2 commits
11s
2s
34s
1min 7s
24s
#6
May 05 17:13
1 commit
13s
3s
51s
41s
28s
#5
May 04 16:13
1 commit
13s
4s
39s
33s
29s
SLOCCount Trend
lines total
lines delta
INI
Markdown
Python

Servizio nazionale ci.infn.it



CI/CD environment & Jenkins



Rancher

Open Source Container Management Platform

Running containers in production still isn't easy

Do you want to manage all this?... or this?

App Catalog	Helm, …
Orchestration	Compose, Kubernetes, Marathon,
Scheduling	Swarm, Kubernetes, Mesos, …
Monitoring	cAdvisor, Prometheus, Datadog, …
Access Control	LDAP, AD, GitHub, …
Registry	Nexus, Artifactory, DTR…
Engine	Docker, runC, Rocket …
Security	Notary, Vault, …
Network	VXLAN, IPSEC, HAProxy, …
Storage	Ceph, Gluster, Swift, …
Distributed DB	Etcd, Consul, MongoDB, …

RANCHER

Open Source
Container Management platform

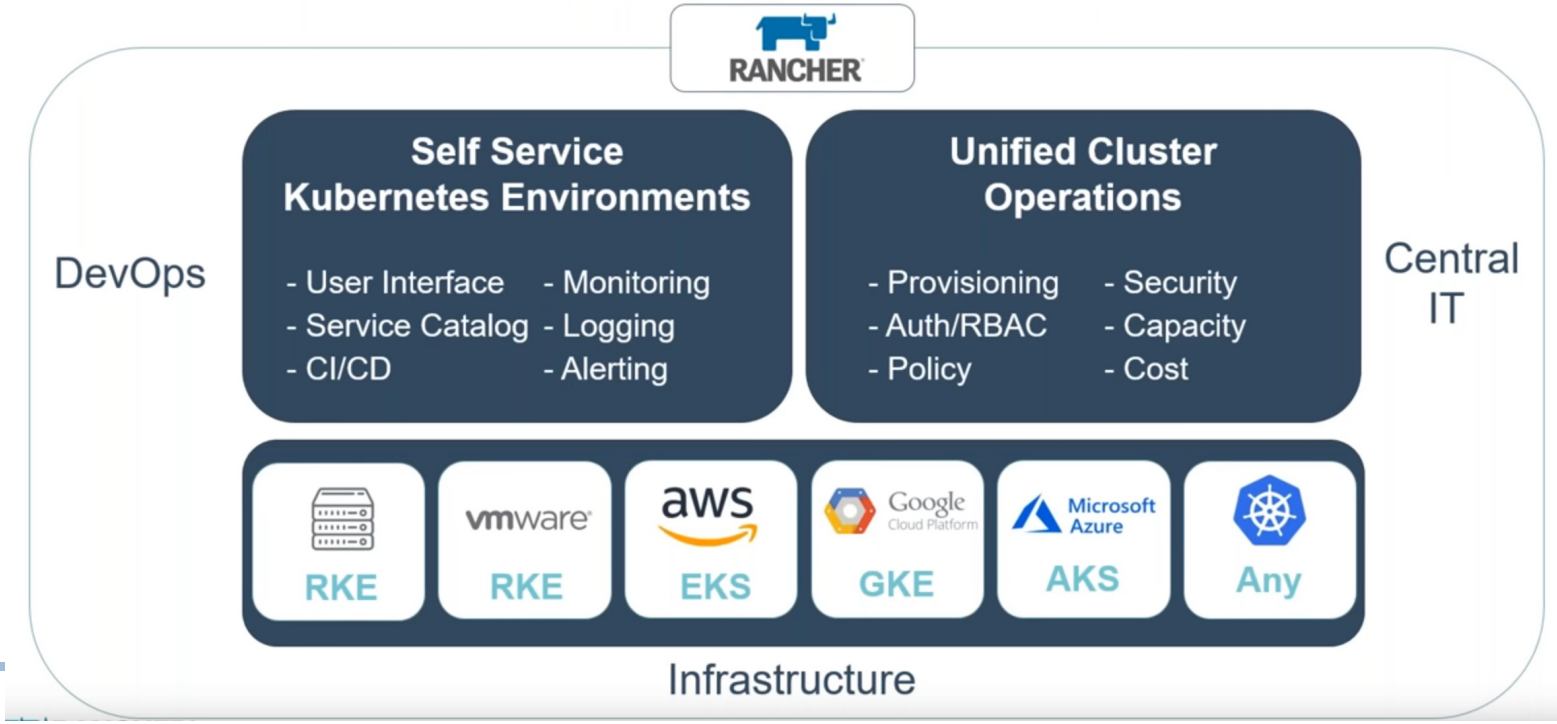
Rancher – Rancher Labs

- Founded in 2014
- Dic 2020 - R**ancher Labs is a part of SUSE**

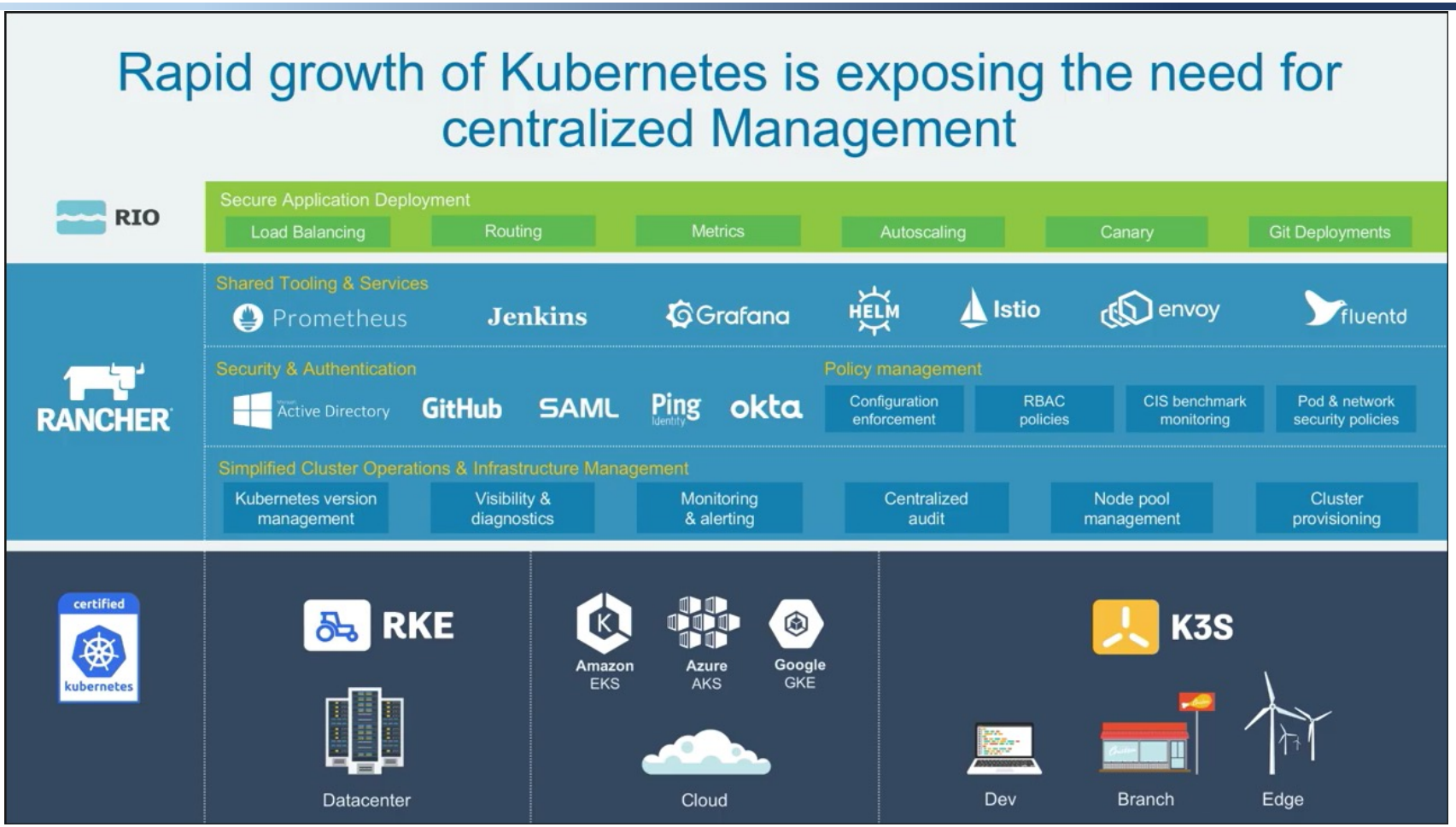
- #1 choice for enterprises looking to run containers and Kubernetes in production
  - “the **only product** that enables you to deliver Kubernetes-as-a-Service across any infrastructure”
- delivers open source software that enables organizations to
  - **deploy and manage** Kubernetes at scale
  - on **any infrastructure** across the data center, cloud, branch offices and the network edge
- +35,000 active users
- +100 million downloads

Rancher, the product

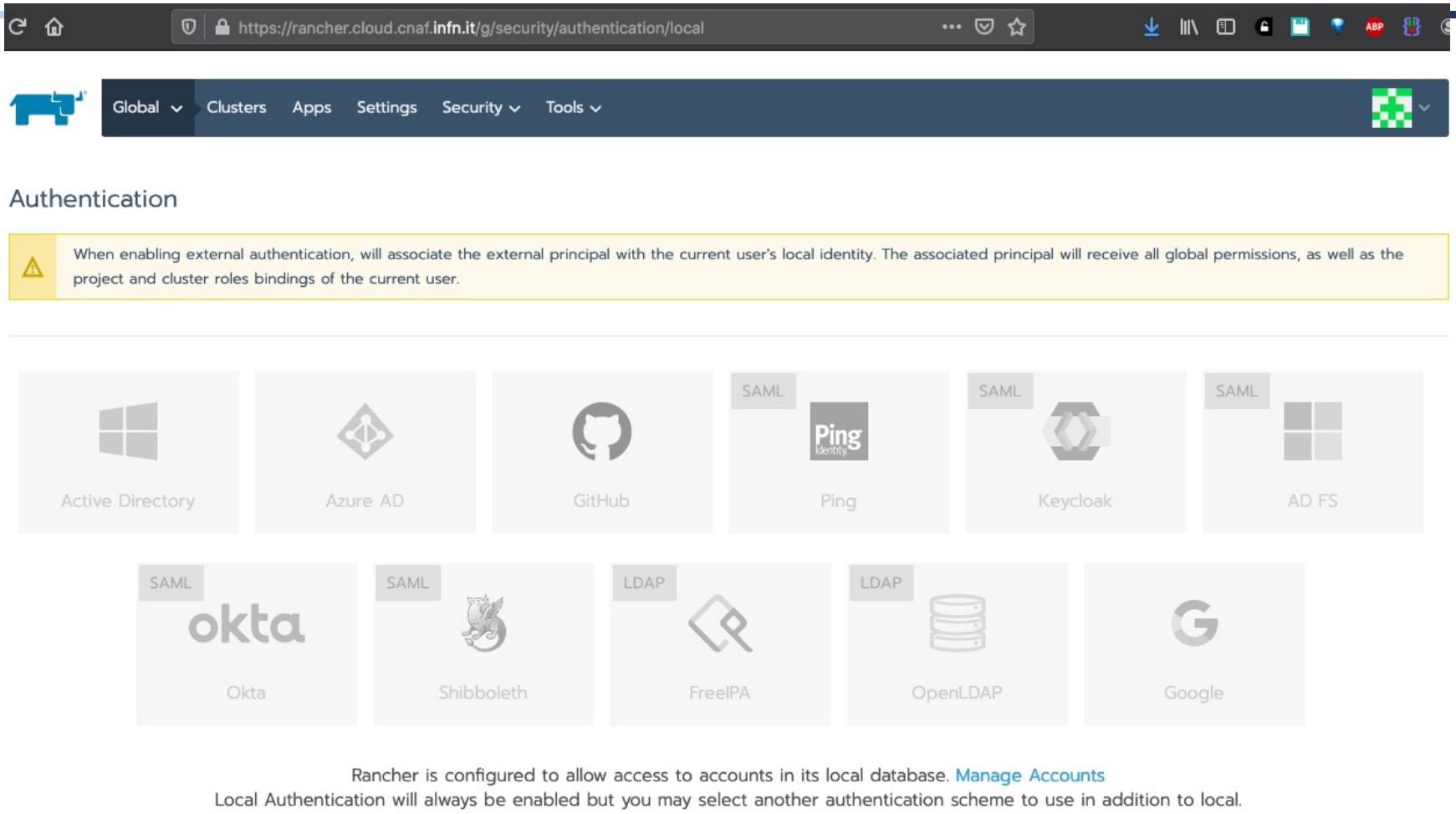
- Open source container management platform
- Main goals:
  - Managing and deploying K8S anywhere (ability to build cluster on bare metal, VM wherever is possible)
  - Ability to manage clusters in in hosted clusters like RKE or EKS and cloud, Amazon , Azure, or imported clusters
- Oriented to address different requests



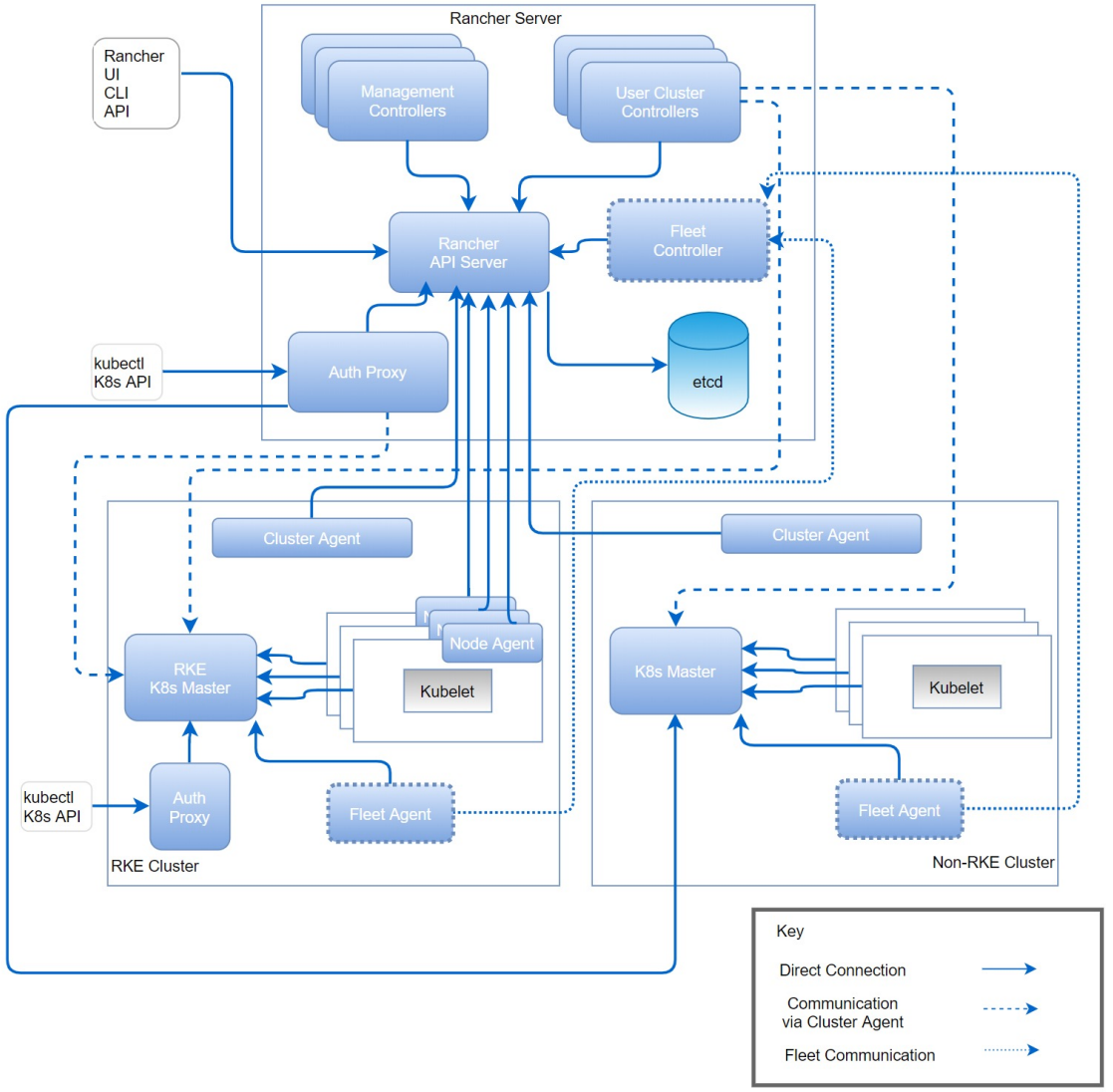
Need of Centralized Management



Authentication

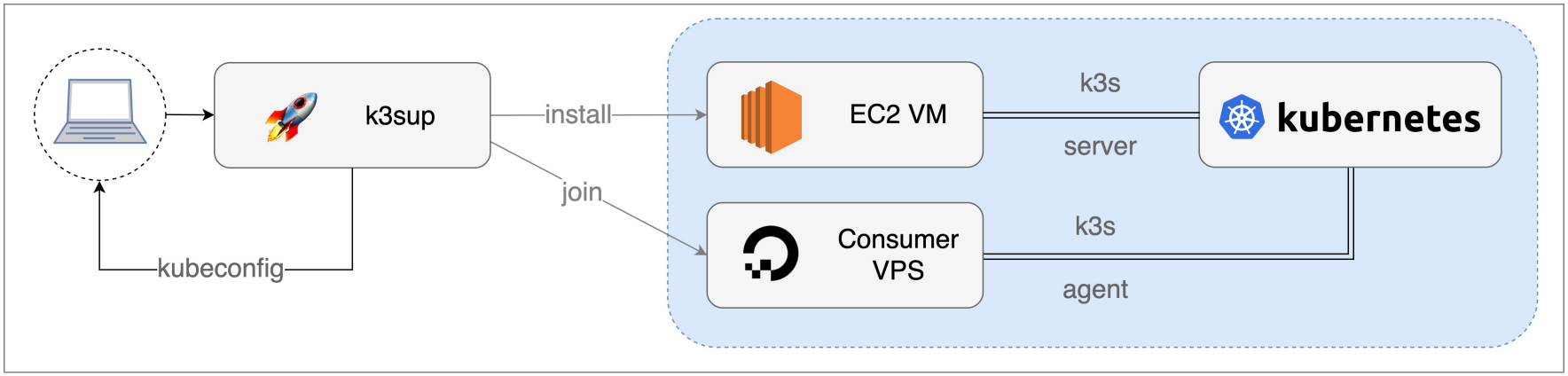


Architettura



K3S

- Super-lightweight CNCF-certified K8s distribution
- Different from non-production solutions like minikube/microk8s/kind
- Small binary / Uses 512MB of RAM for all of Kubernetes
- Easily installed with k3sup from Alex Ellis

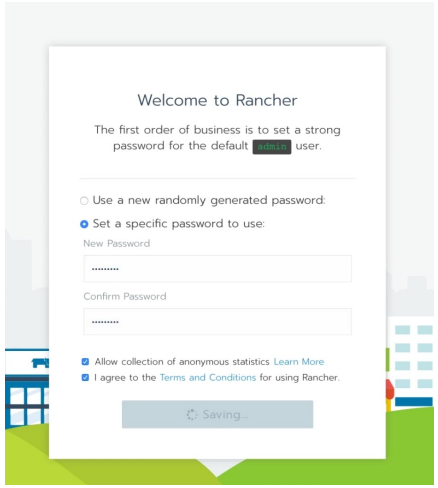


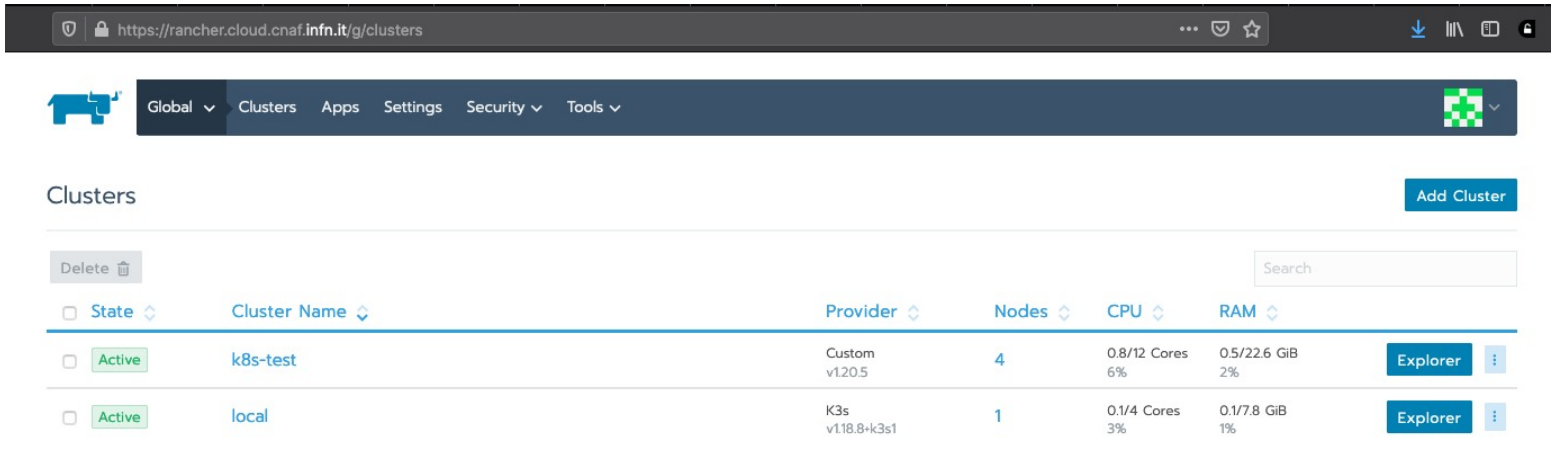
Architecture - k3sup running locally against any VM such as AWS EC2 or a VPS such as DigitalOcean

How it works

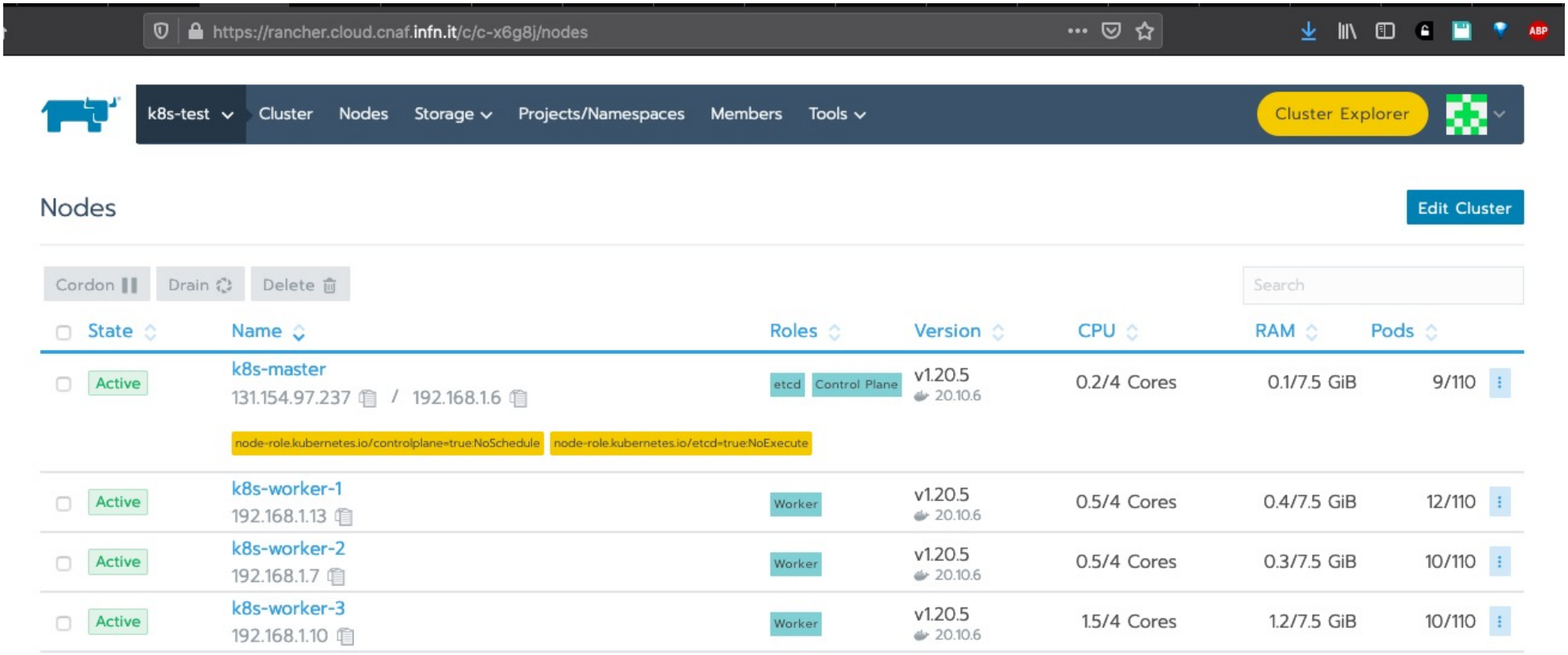
- https://rancher.com/docs/
  - 2 types of deployment: with HA and single instance
  - Demo - the single node:

  [centos@devops13 ~]$ docker run -d –privileged --restart=unless-stopped -p 80:80 -p 443:443 -v /opt/rancher:/var/lib/rancher rancher/rancher:v2.5.4

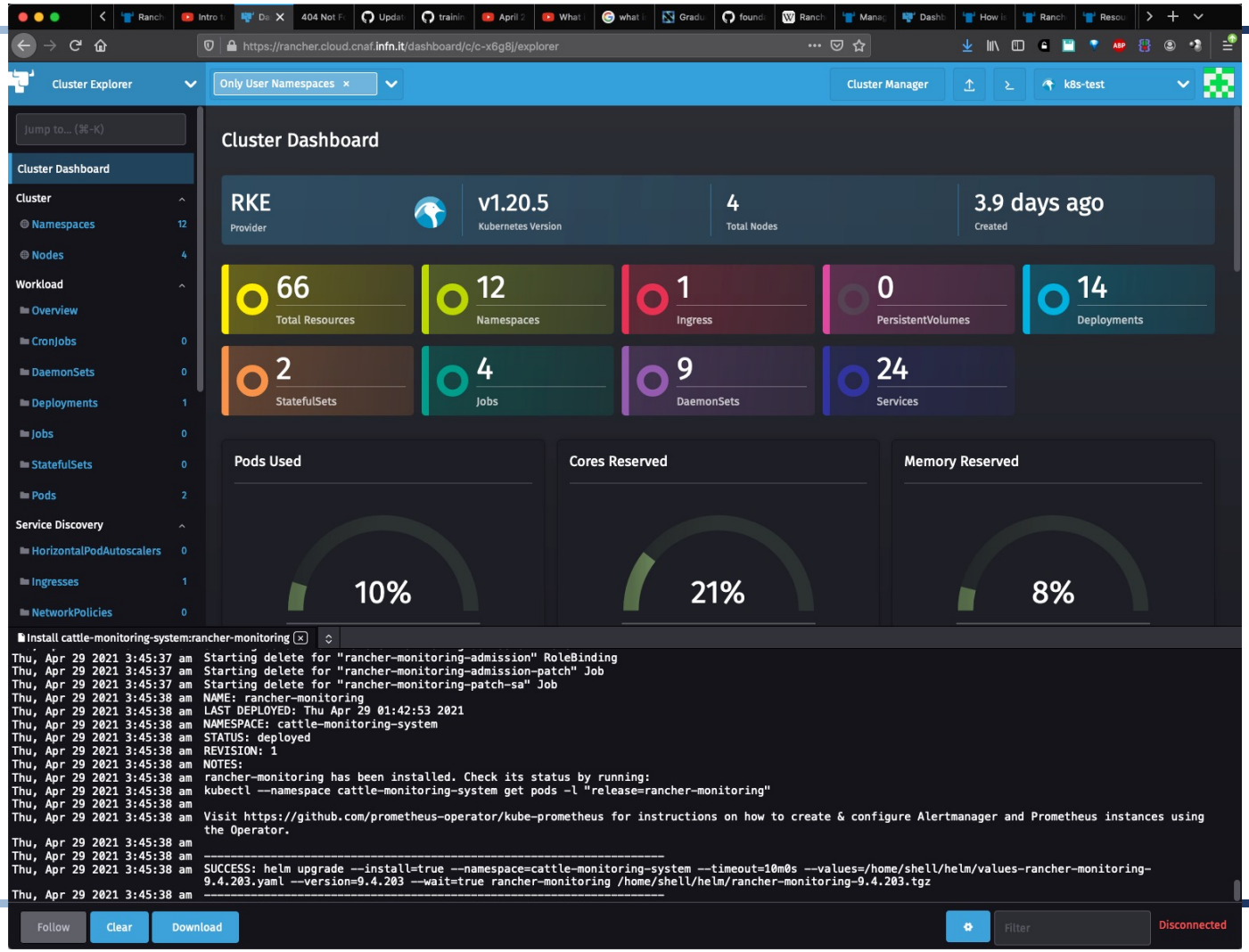




Test Cluster

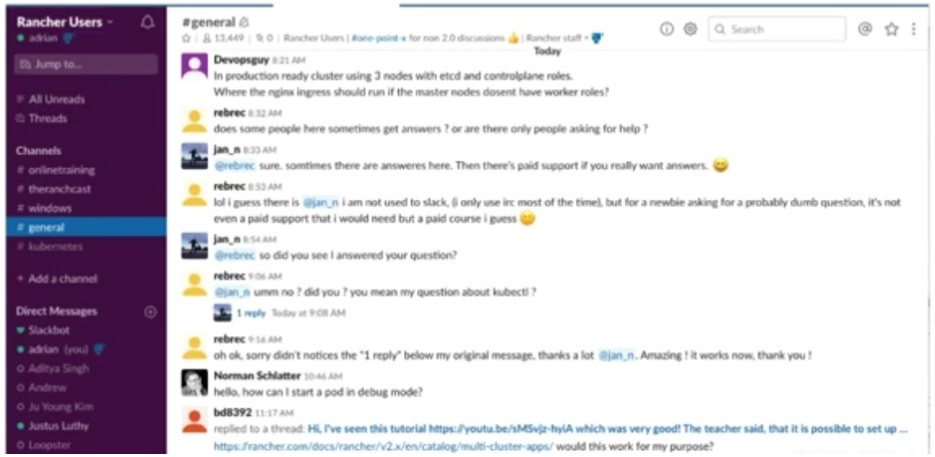


One click install applications



Resources – Community Support

http://slack.rancher.io



http://forums.rancher.com