TESTINGWEEK

ALWAYS TESTING, NEVER RESTING

MAY 13 - 17

Scaling Automated Execution with Selenium & Kubernetes

By Catherine Cruz & Tatiana Tamayo

AGENDA

- OVERCOMING THE CHALLENGES OF EXECUTING AUTOMATED TEST AT HYPERSCALE
- WHAT SELENIUM IS?
- WHAT KUBERNETES IS?
- WHERE THE FIT TOGETHER?
- HOW TO USE IT?
- HANDS ON SESSION

OVERCOMING THE CHALLENGES OF EXECUTING AUTOMATED TEST AT HYPERSCALE

WE KNOW HOW COMPLICATED IS TO:

- Have a stable grid to run UI tests with Selenium
- Maintain it over time (keep up with a new browser, Selenium and drivers versions)
- Provide capabilities to cover all browsers and platforms

HOW TO USE THIS APPROACH:

This approach where docker-selenium nodes are created on demand. Your UI tests run faster in any browser because they are running in your own local network, on a node created from scratch and disposed of after the test completes.

Our main goal of using Selenium + Kubernetes is to allow anyone to have a disposable and flexible Selenium Grid infrastructure.

OVERCOMING THE CHALLENGES OF EXECUTING AUTOMATED TEST AT HYPERSCALE

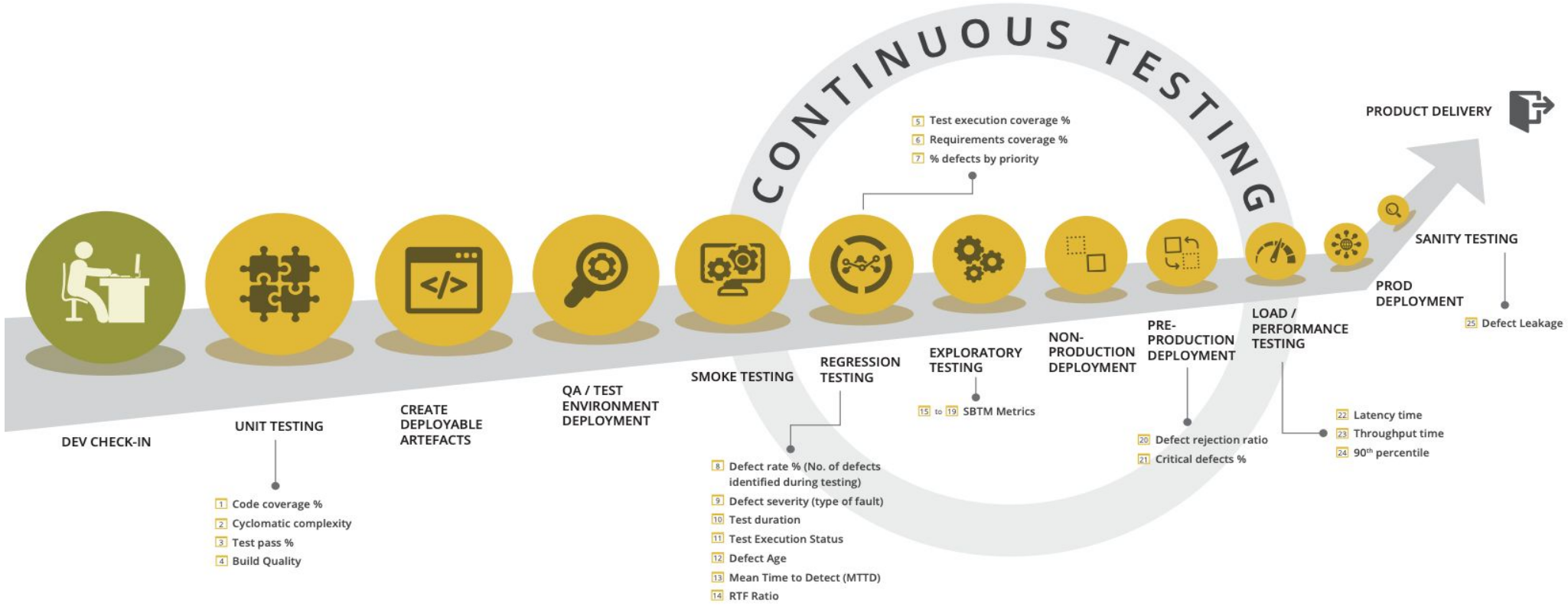
To understand why we need to scale our automated test execution, first we need to have an overview of our technology scenario:

- Time to Market
  - Ship It as soon as possible
- Cost Optimization
  - Compete with less cost
- Paradigm shift
  - Constantly changing environment: CI, CD, DevOps
- Distributed computing
  - Dockers, Cloud computing

Therefore it's all about:

WHAT CONTINUOUS DELIVERY MEANS?

A lot of digital transformations are going into the Quality Assurance area, companies are ‘shifting to the left’, focusing more on how they can automate their test processes to make them quicker, easier to configure and more efficiently deployed.



WHAT SELENIUM IS?

Open source tool which is used for automating the tests carried out on web browsers. **Selenium** automates browsers. That's it! What you do with that power is entirely up to you.

Primarily, it is for automating web applications for testing purposes, but is certainly not limited to just that. Boring web-based administration tasks can (and should!) be automated as well.

Selenium has the support of some of the largest browser vendors who have taken (or are taking) steps to make Selenium a native part of their browser. It is also the core technology in countless other browser automation tools, APIs and frameworks.

SELENIUM SUITE

Selenium WebDriver

If you want to

- create robust, browser-based regression automation suites and tests
- scale and distribute scripts across many environments

Then you want to use Selenium WebDriver; a collection of language specific bindings to drive a browser -- the way it is meant to be driven.

Selenium WebDriver is the successor of Selenium Remote Control which has been officially deprecated. The Selenium Server (used by both WebDriver and Remote Control) now also includes built-in grid capabilities.

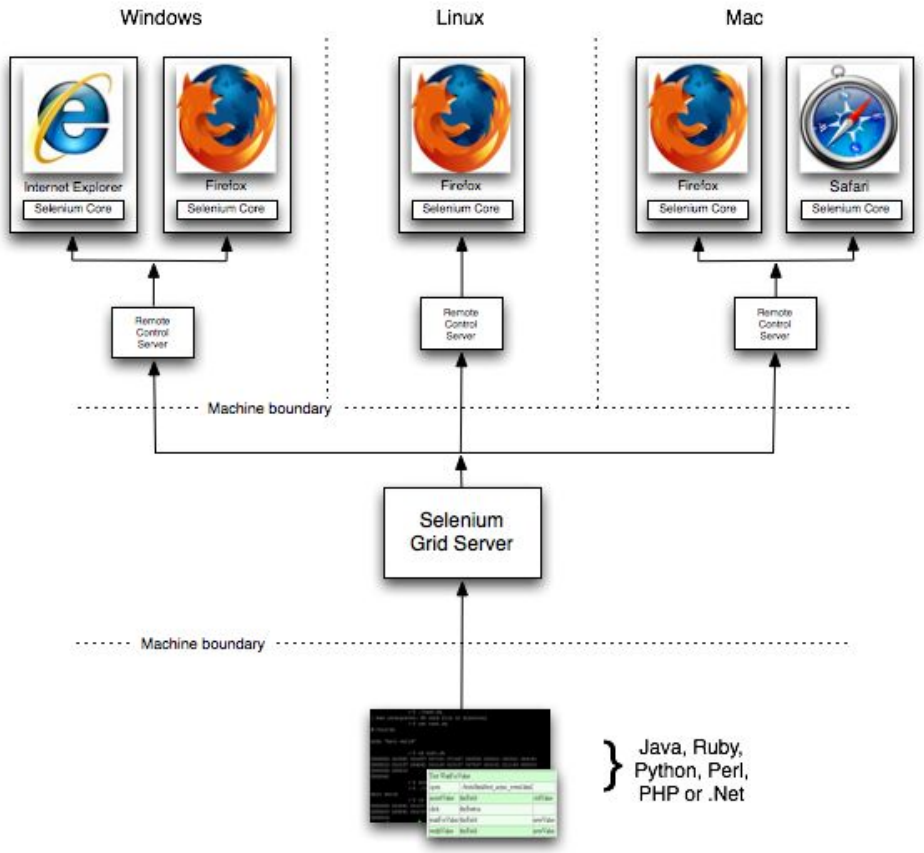
Selenium IDE

If you want to

- create quick bug reproduction scripts
- create scripts to aid in automation-aided exploratory testing

Then you want to use Selenium IDE; a Chrome and Firefox add-on that will do simple record-and-playback of interactions with the browser.

Selenium Grid



SELENIUM GRID

Selenium Grid is a part of the Selenium Suite that specializes in running multiple tests across different browsers, operating systems, and machines in parallel.

Selenium Grid has 2 versions - The older Grid 1 and the newer Grid 2. We will only focus on Grid 2 because Grid 1 is gradually being deprecated by the Selenium Team.

Selenium Grid uses a **hub-node** concept where you only run the test on a single machine called a hub, but the execution will be done by different machines called nodes.

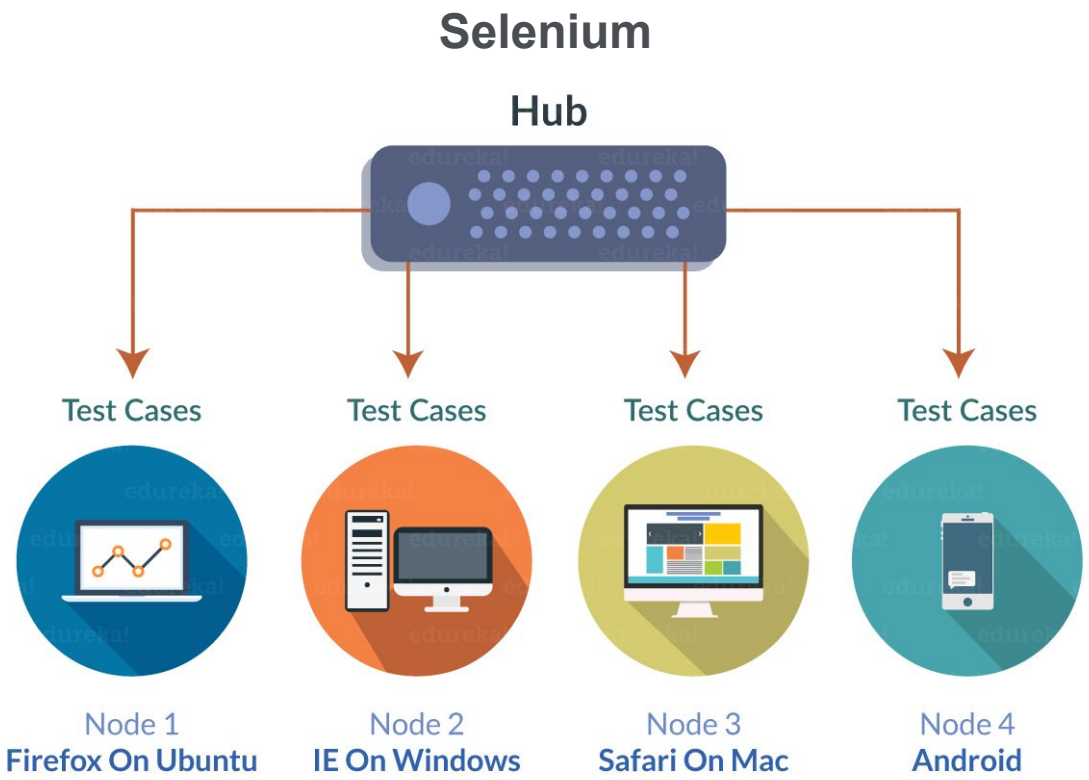
When to Use Selenium Grid?

You should use Selenium Grid when you want to do either one or both of following:

- **Run your tests against different browsers,** operating systems, and machines all at the same time. This will ensure that the application you are testing is fully compatible with a wide range of browser-O.S combinations.
- **Save time in the execution of your test suites.** If you set up Selenium Grid to run, say, 4 tests at a time, then you would be able to finish the whole suite around 4 times faster.

BENEFITS OF USING SELENIUM

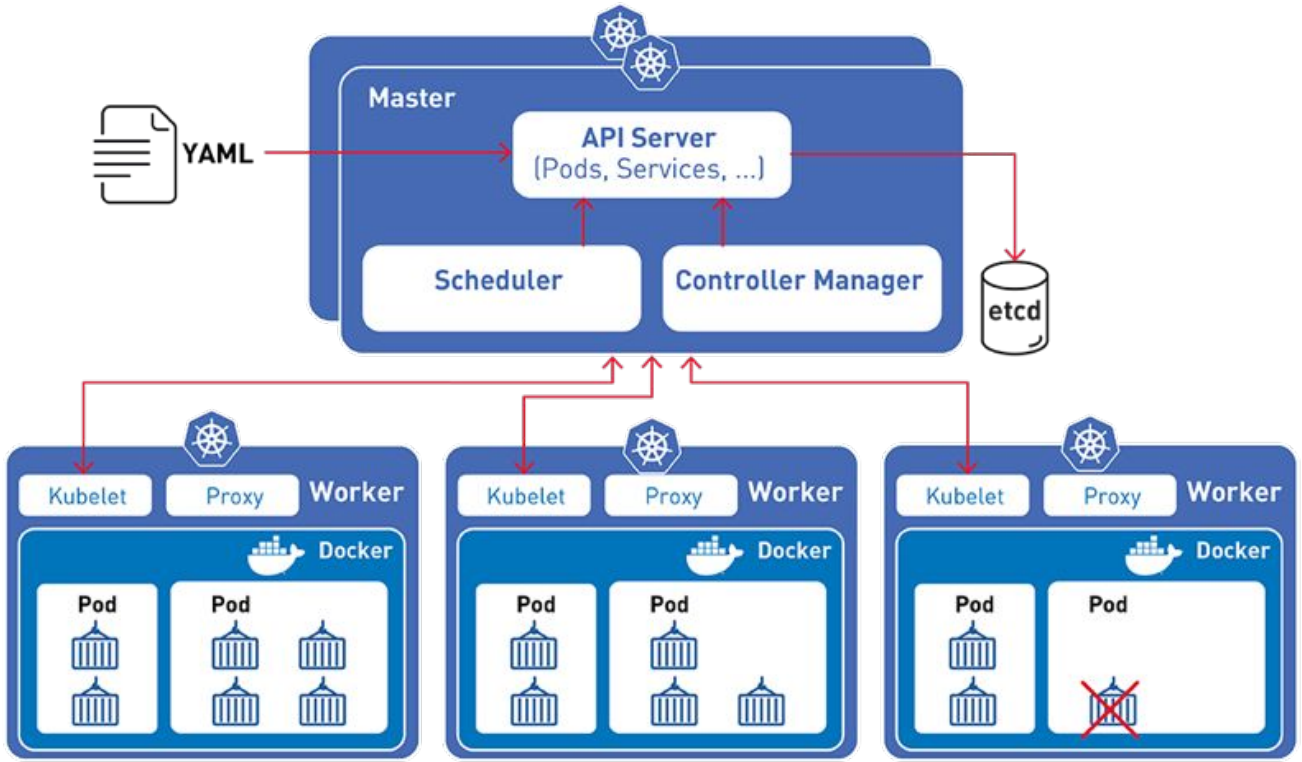
- Test scripts can be written in any of these **programming languages**: Java, Python, C#, PHP, Ruby, Perl & .Net
- Tests can be carried out in any of these **OS**: Windows, Macor Linux
- Tests can be carried out using any **browser**: Mozilla Firefox,Internet Explorer, Google Chrome, Safari or Opera
- It can be integrated with tools such as **TestNG & JUnit** for managing test cases and generating reports
- It can be integrated with **Maven, Jenkins & Docker** to achieve **Continuous Testing**



WHAT KUBERNETES IS?

Open-Source **container orchestration** that provides application deployment, scaling, and management.

Designed from the ground up to be an environment for building distributed applications from containers.

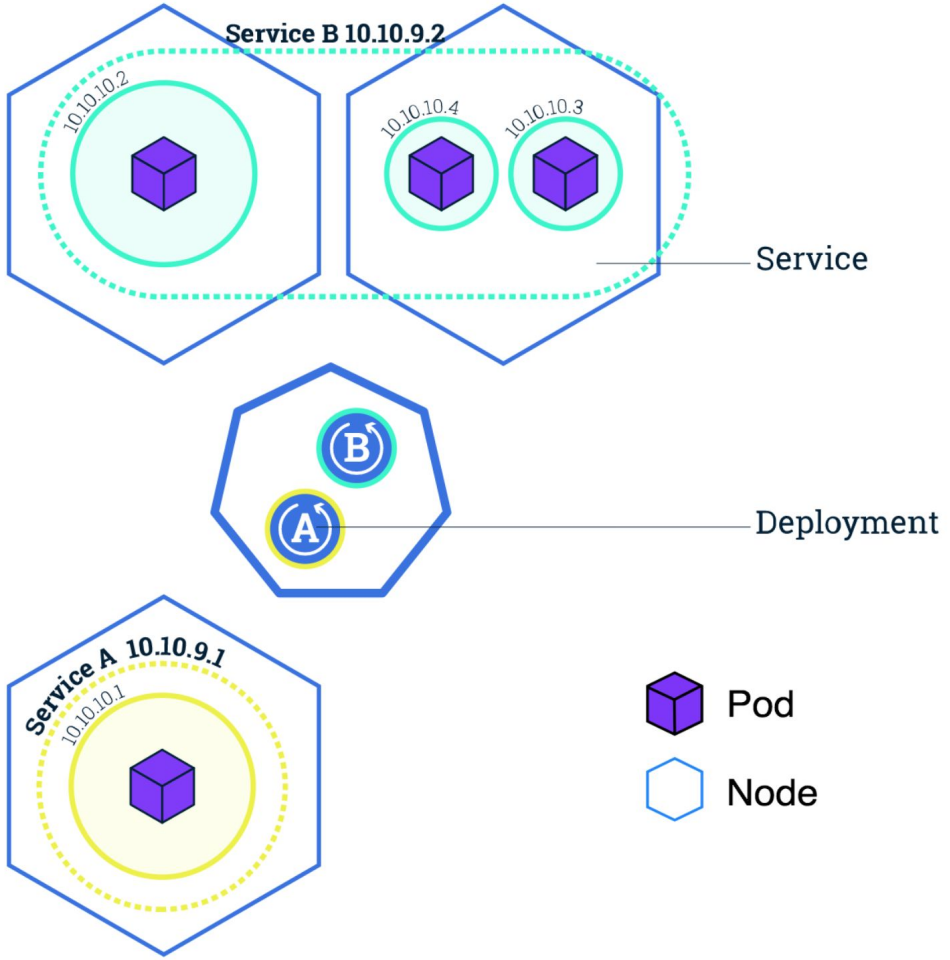


https://kubernetes.io/

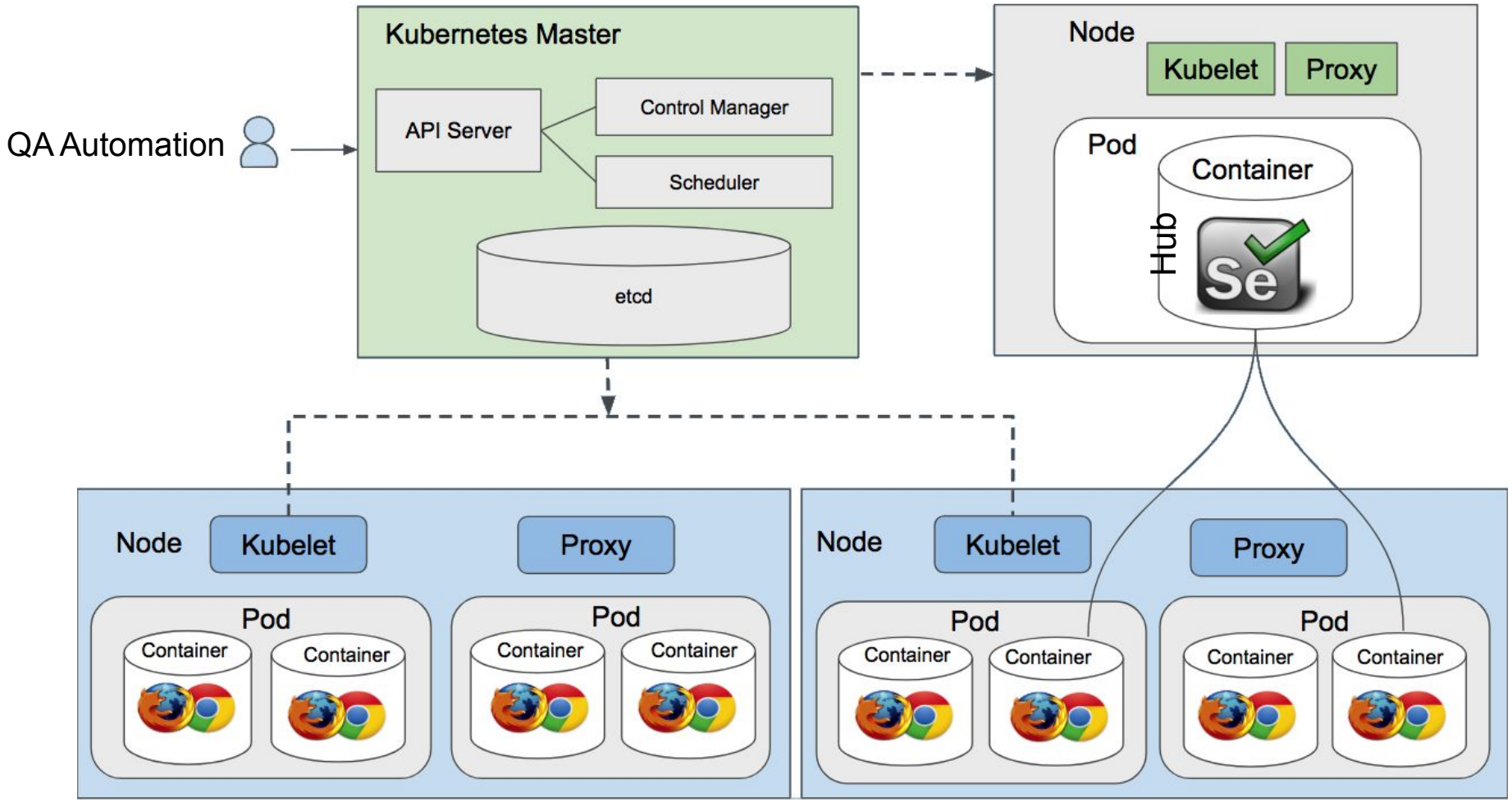
WHAT KUBERNETES IS?

Objects used to orchestrate

- **Pod** – group of one or more running containers.
- **Pod template** – describes the desired state of a single container.
- **Labels**: key-value set each object in k8s to group the resources.
- **Replication Controller**: pod template + number of desired replicas.
- **Services**: provide abstracted static IP address for pods.
- **Deployment** – Manges pods, services, liveness checks and RC.



WHERE THE FIT TOGETHER?



HOW TO USE KUBERNETES?

Minikube a tool that makes it easy to run K8s locally. It runs a single-node K8s cluster inside a VM.

```
$ minikube start
```

```
$ minikube dashboard
```

Kubectl is a CLI command line interface for running commands against Kubernetes clusters.

```
$ kubectl cluster-info
```

```
$ kubectl get nodes
```

```
$ kubectl get all
```

KUBERNETES BASIC COMMANDS

Now we are going to review some basic k8s command for deployment and management.

```
$ kubectl run ghost --image=ghost:0.9
```

```
$ kubectl expose deployment ghost --port=2368 --type=LoadBalancer
```

Because we are using **Minikube** use --external-ip=$(minikube ip) flag when exposing

```
$ kubectl scale --replicas=3 deployment ghost
```

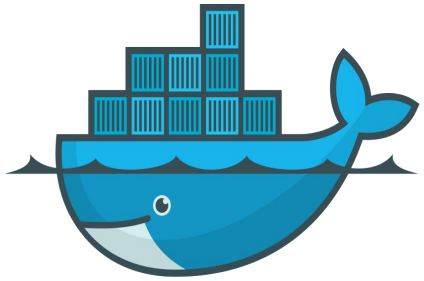
Here some pod specific access and management commands

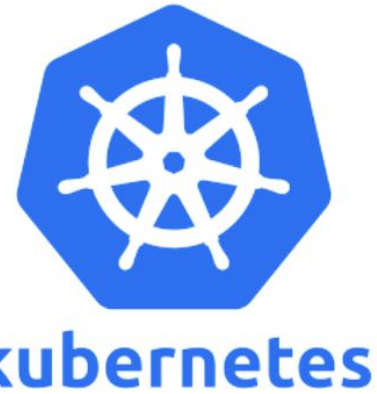
```
$ kubectl label pods <pod-name> owner=tester
```

```
$ kubectl logs <pod-name>
```

```
$ kubectl exec -ti <pod-name> /bin/bash
```

HANDS ON SESSION





https://github.com/twogg-git/k8s-selenium

THANKS!