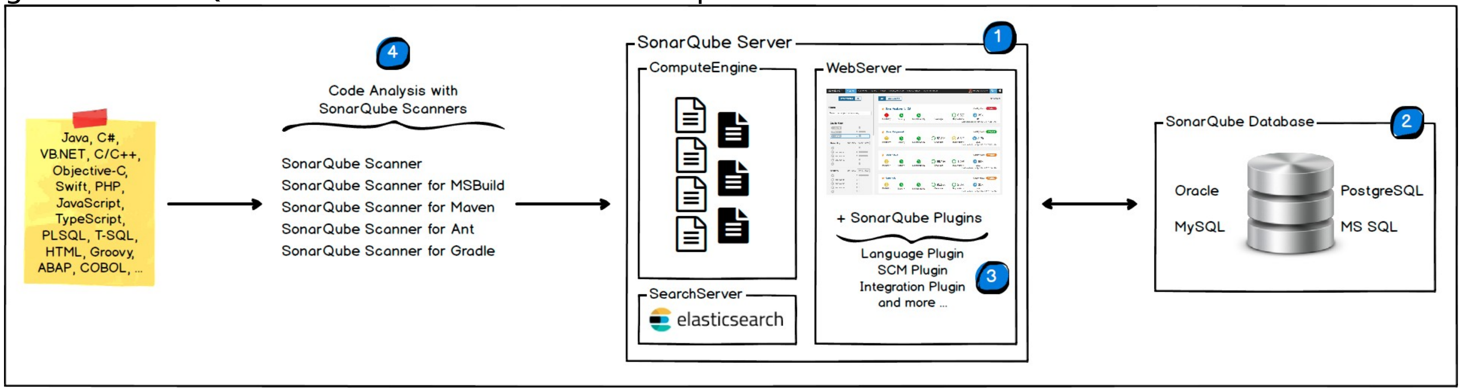
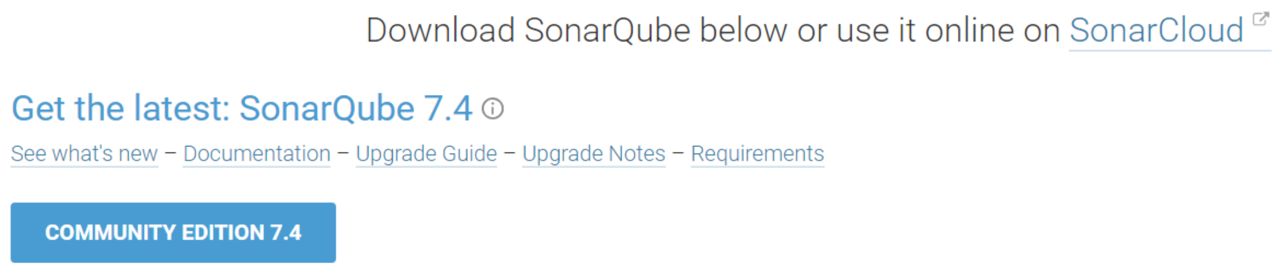
gtiThe SonarQube Platform is made of 4 components:


Java, C#, VB.NET, C/C++, Objective-C, Swift, PHP, JavaScript, TypeScript, PLSQL, T-SQL, HTML, Groovy, ABAP, COBOL, ...
4
Code Analysis with SonarQube Scanners
SonarQube Scanner
SonarQube Scanner for MSBuild
SonarQube Scanner for Maven
SonarQube Scanner for Ant
SonarQube Scanner for Gradle
SonarQube Server
1
ComputeEngine
WebServer
+ SonarQube Plugins
Language Plugin
SCM Plugin
Integration Plugin
and more ...
3
SearchServer
elasticsearch
SonarQube Database
2
Oracle
MySQL
PostgreSQL
MS SQL

1. One SonarQube Server starting 3 main processes:
   - Web Server for developers, managers to browse quality snapshots and configure the SonarQube instance
   - Search Server based on Elasticsearch to back searches from the UI
   - Compute Engine Server in charge of processing code analysis reports and saving them in the SonarQube Database
2. One SonarQube Database to store:
   - The configuration of the SonarQube instance (security, plugins settings, etc.)
   - The quality snapshots of projects, views, etc.
3. Multiple SonarQube Plugins installed on the server, possibly including language, SCM, integration, authentication, and governance plugins
4. One or more SonarScanners running on your Build / Continuous Integration Servers to analyze projects

We can work online on SonarQube in the last versions but we will be installing it locally for this project, let's start with installing and lunching SonarQube:


Download SonarQube below or use it online on SonarCloud

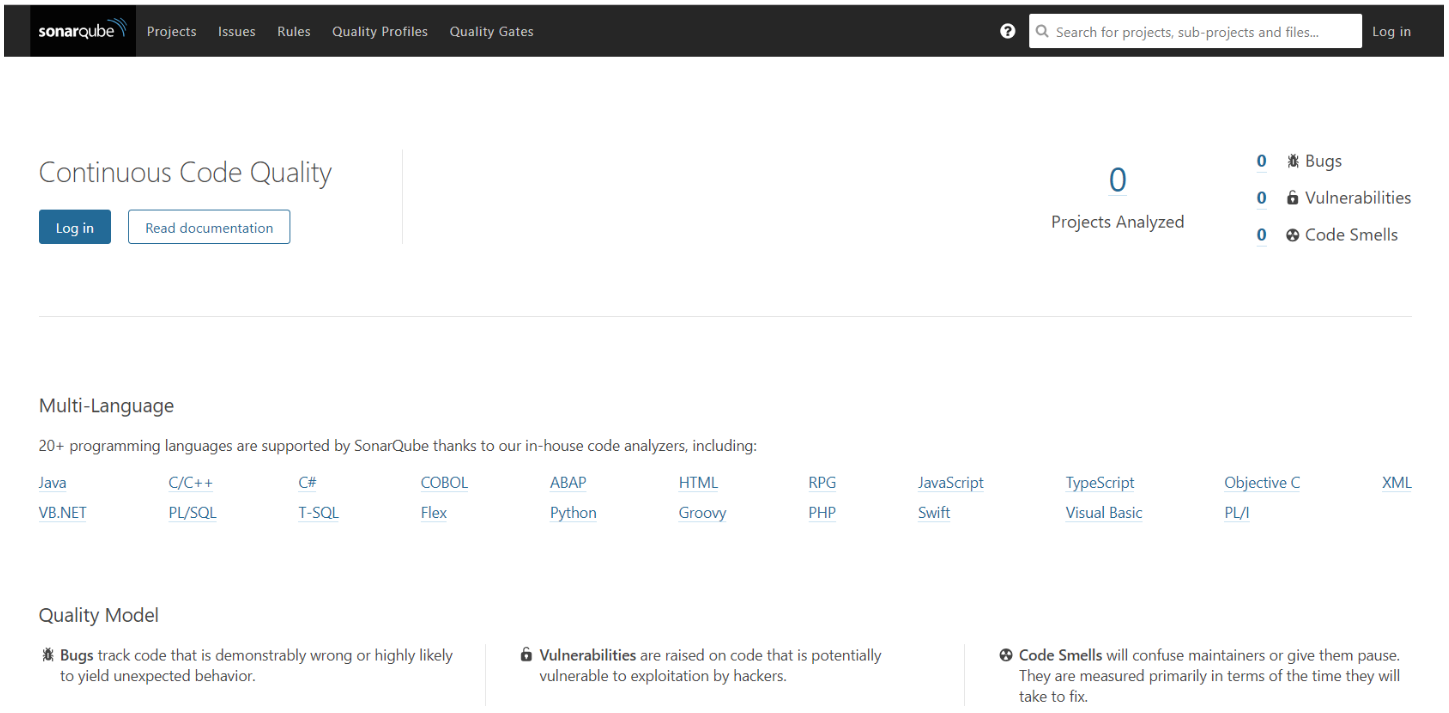
Get the latest: SonarQube 7.4

See what's new – Documentation – Upgrade Guide – Upgrade Notes – Requirements

After unzipping the content of the zip downloaded in C:\SonarQube-7.4 foe example, we can start it with a simple command

C:\SonarQube-7.4\bin\windows-x86-64\StartSonar.bat

Then going to http://localhost:9000 we will see the home page of SonarQube



Projects Issues Rules Quality Profiles Quality Gates

Search for projects, sub-projects and files...

Log in

Continuous Code Quality

Log in Read documentation

0 Projects Analyzed

0 Bugs
0 Vulnerabilities
0 Code Smells

Multi-Language

20+ programming languages are supported by SonarQube thanks to our in-house code analyzers, including:

Java	C/C++	C#	COBOL	ABAP	HTML	RPG	JavaScript	TypeScript	Objective C	XML
VB.NET	PL/SQL	T-SQL	Flex	Python	Groovy	PHP	Swift	Visual Basic	PL/I	

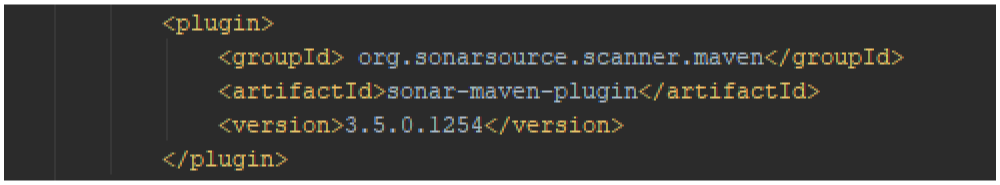
Quality Model

Bugs track code that is demonstrably wrong or highly likely to yield unexpected behavior.

Vulnerabilities are raised on code that is potentially vulnerable to exploitation by hackers.

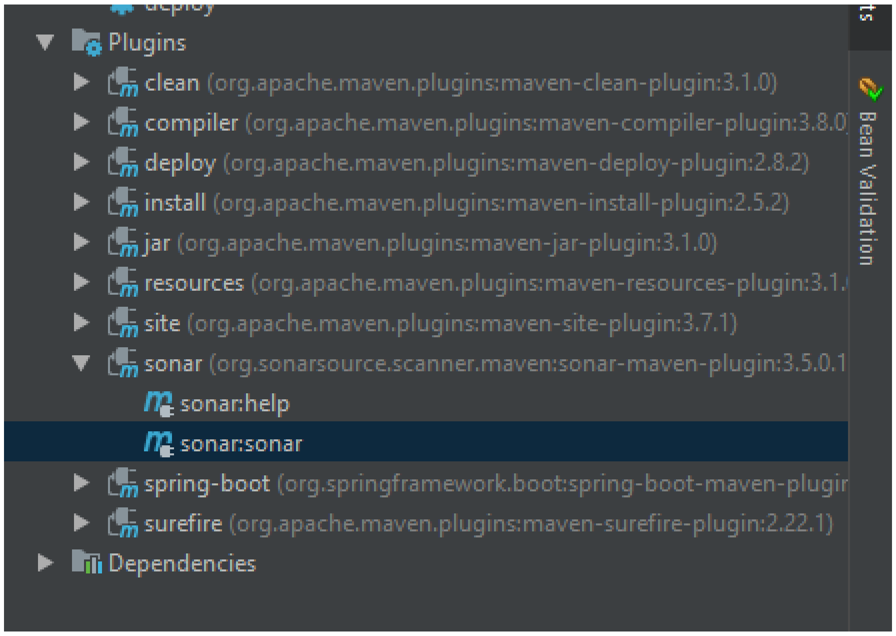
Code Smells will confuse maintainers or give them pause. They are measured primarily in terms of the time they will take to fix.

To test SonarQube we will be using maven project, so we need to add sonar plugin into pom.xml



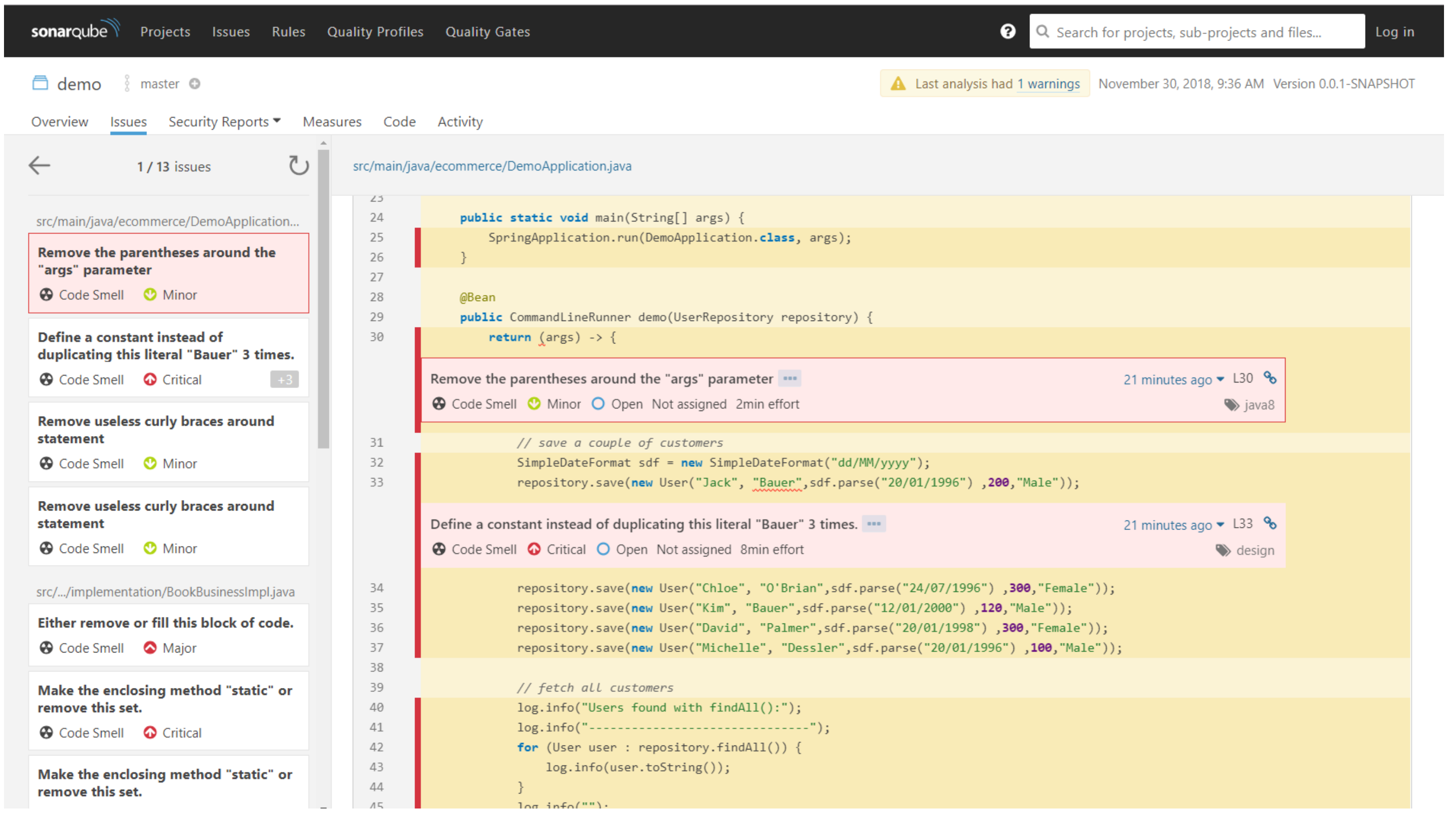
```
<plugin>
    <groupId> org.sonarsource.scanner.maven</groupId>
    <artifactId>sonar-maven-plugin</artifactId>
    <version>3.5.0.1254</version>
</plugin>
```

After running clean and install, we can run sonar:sonar plugin



```
Plugins
  clean (org.apache.maven.plugins:maven-clean-plugin:3.1.0)
  compiler (org.apache.maven.plugins:maven-compiler-plugin:3.8.0
  deploy (org.apache.maven.plugins:maven-deploy-plugin:2.8.2)
  install (org.apache.maven.plugins:maven-install-plugin:2.5.2)
  jar (org.apache.maven.plugins:maven-jar-plugin:3.1.0)
  resources (org.apache.maven.plugins:maven-resources-plugin:3.1.
  site (org.apache.maven.plugins:maven-site-plugin:3.7.1)
  sonar (org.sonarsource.scanner.maven:sonar-maven-plugin:3.5.0.1
    sonar:help
    sonar:sonar
  spring-boot (org.springframework.boot:spring-boot-maven-plugir
  surefire (org.apache.maven.plugins:maven-surefire-plugin:2.22.1)
Dependencies
```

After a build success we can now open http://localhost:9000 to observe this interface that contains the projects analyzed by SonarQube and the detected issues



Last analysis had 1 warnings November 30, 2018, 9:36 AM Version 0.0.1-SNAPSHOT

src/main/java/ecommerce/DemoApplication...

- Remove the parentheses around the "args" parameter — Code Smell, Minor
- Define a constant instead of duplicating this literal "Bauer" 3 times. — Code Smell, Critical, +3
- Remove useless curly braces around statement — Code Smell, Minor
- Remove useless curly braces around statement — Code Smell, Minor

src/.../implementation/BookBusinessImpl.java

- Either remove or fill this block of code. — Code Smell, Major
- Make the enclosing method "static" or remove this set. — Code Smell, Critical
- Make the enclosing method "static" or remove this set.

src/main/java/ecommerce/DemoApplication.java

```
    public static void main(String[] args) {
        SpringApplication.run(DemoApplication.class, args);
    }

    @Bean
    public CommandLineRunner demo(UserRepository repository) {
        return (args) -> {
```

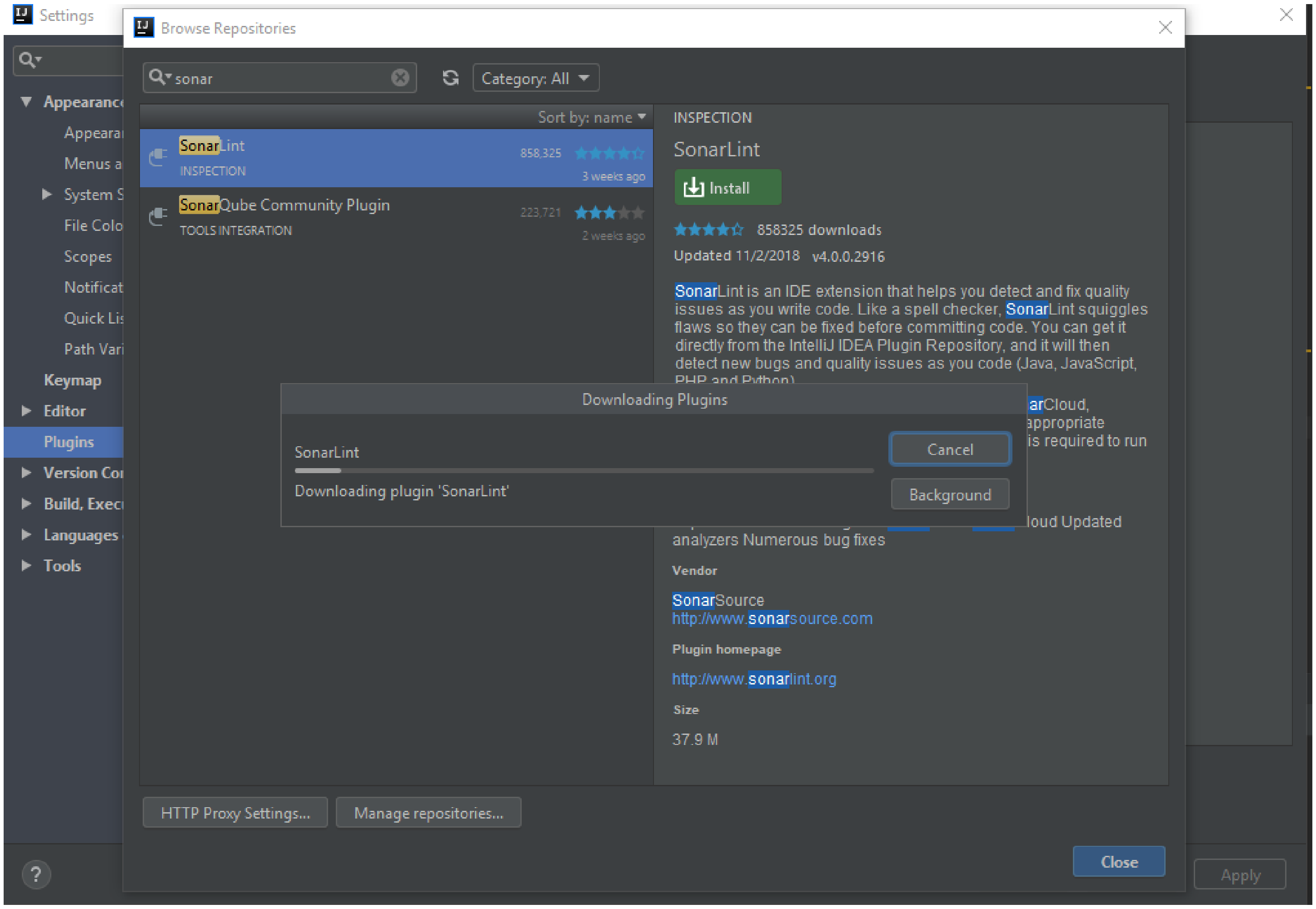
Remove the parentheses around the "args" parameter — 21 minutes ago L30
Code Smell Minor Open Not assigned 2min effort — java8

```
            // save a couple of customers
            SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");
            repository.save(new User("Jack", "Bauer",sdf.parse("20/01/1996") ,200,"Male"));
```

Define a constant instead of duplicating this literal "Bauer" 3 times. — 21 minutes ago L33
Code Smell Critical Open Not assigned 8min effort — design

```
            repository.save(new User("Chloe", "O'Brian",sdf.parse("24/07/1996") ,300,"Female"));
            repository.save(new User("Kim", "Bauer",sdf.parse("12/01/2000") ,120,"Male"));
            repository.save(new User("David", "Palmer",sdf.parse("20/01/1998") ,300,"Female"));
            repository.save(new User("Michelle", "Dessler",sdf.parse("20/01/1996") ,100,"Male"));

            // fetch all customers
            log.info("Users found with findAll():");
            log.info("-------------------------------");
            for (User user : repository.findAll()) {
                log.info(user.toString());
            }
```

Thus in intellj we can add a plugin called SonarLint that do scan the code for issues and errors



INSPECTION

SonarLint

858325 downloads

Updated 11/2/2018 v4.0.0.2916

SonarLint is an IDE extension that helps you detect and fix quality issues as you write code. Like a spell checker, SonarLint squiggles flaws so they can be fixed before committing code. You can get it directly from the IntelliJ IDEA Plugin Repository, and it will then detect new bugs and quality issues as you code (Java, JavaScript, PHP and Python).

arCloud, appropriate is required to run

loud Updated analyzers Numerous bug fixes

Vendor

SonarSource
http://www.sonarsource.com

Plugin homepage

http://www.sonarlint.org

Size

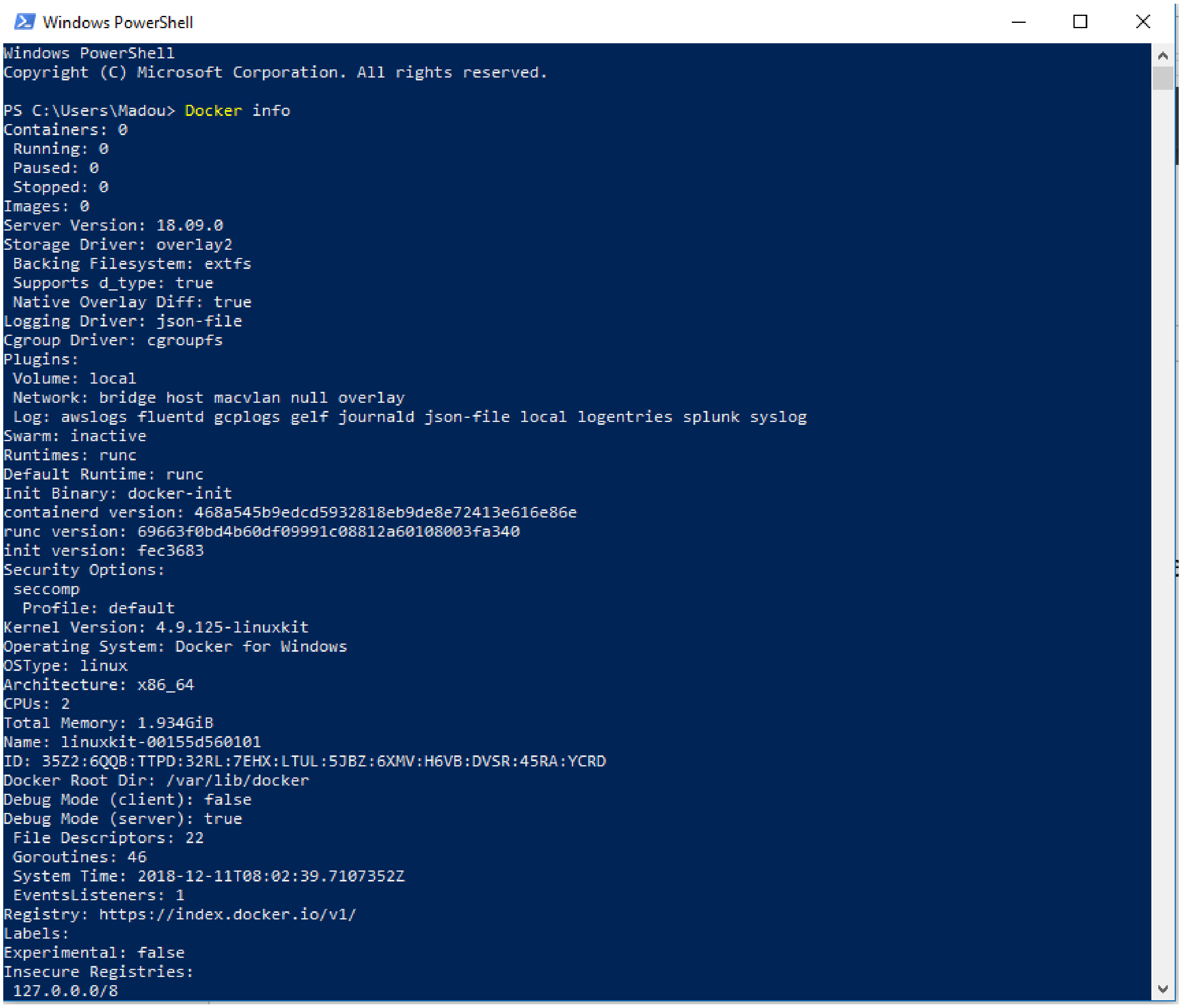
37.9 M

Downloading Plugins

SonarLint
Downloading plugin 'SonarLint'
Cancel Background

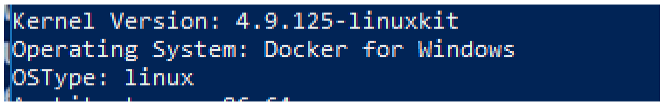
Now to Docker:

After downloading Docker for windows and installing it we need to check the installation by opening a Windows PowerShell and type "Docker info"



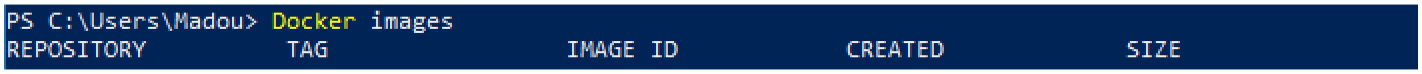
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\Madou> Docker info
Containers: 0
 Running: 0
 Paused: 0
 Stopped: 0
Images: 0
Server Version: 18.09.0
Storage Driver: overlay2
 Backing Filesystem: extfs
 Supports d_type: true
 Native Overlay Diff: true
Logging Driver: json-file
Cgroup Driver: cgroupfs
Plugins:
 Volume: local
 Network: bridge host macvlan null overlay
 Log: awslogs fluentd gcplogs gelf journald json-file local logentries splunk syslog
Swarm: inactive
Runtimes: runc
Default Runtime: runc
Init Binary: docker-init
containerd version: 468a545b9edcd5932818eb9de8e72413e616e86e
runc version: 69663f0bd4b60df09991c08812a60108003fa340
init version: fec3683
Security Options:
 seccomp
  Profile: default
Kernel Version: 4.9.125-linuxkit
Operating System: Docker for Windows
OSType: linux
Architecture: x86_64
CPUs: 2
Total Memory: 1.934GiB
Name: linuxkit-00155d560101
ID: 35Z2:6QQB:TTPD:32RL:7EHX:LTUL:5JBZ:6XMV:H6VB:DVSR:45RA:YCRD
Docker Root Dir: /var/lib/docker
Debug Mode (client): false
Debug Mode (server): true
 File Descriptors: 22
 Goroutines: 46
 System Time: 2018-12-11T08:02:39.7107352Z
 EventsListeners: 1
Registry: https://index.docker.io/v1/
Labels:
Experimental: false
Insecure Registries:
 127.0.0.0/8
```

We can see here that we are using Docker on windows but the OSType is linux



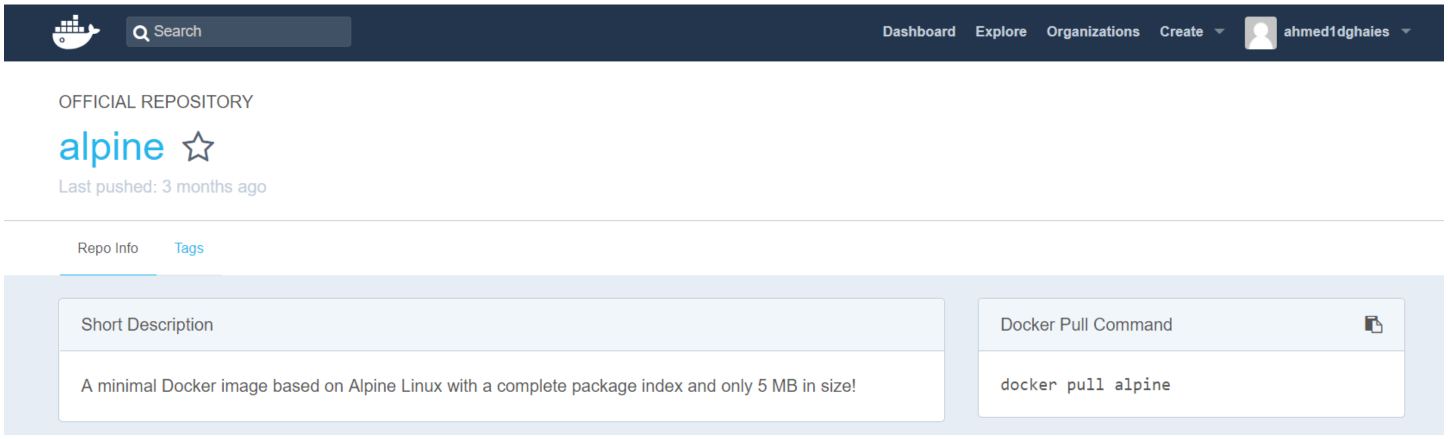
```
Kernel Version: 4.9.125-linuxkit
Operating System: Docker for Windows
OSType: linux
```

For the time being we have an empty Docker:



```
PS C:\Users\Madou> Docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
```

Now let's look for a small image to download and try going to hub.docker.com



OFFICIAL REPOSITORY

alpine

Last pushed: 3 months ago

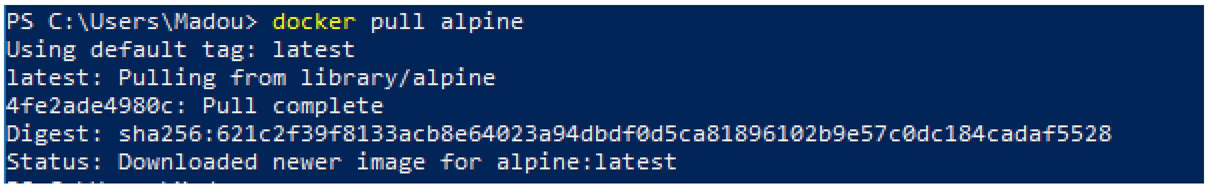
Repo Info | Tags

Short Description

A minimal Docker image based on Alpine Linux with a complete package index and only 5 MB in size!

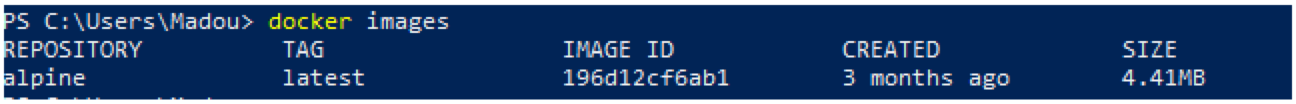
Docker Pull Command

```
docker pull alpine
```



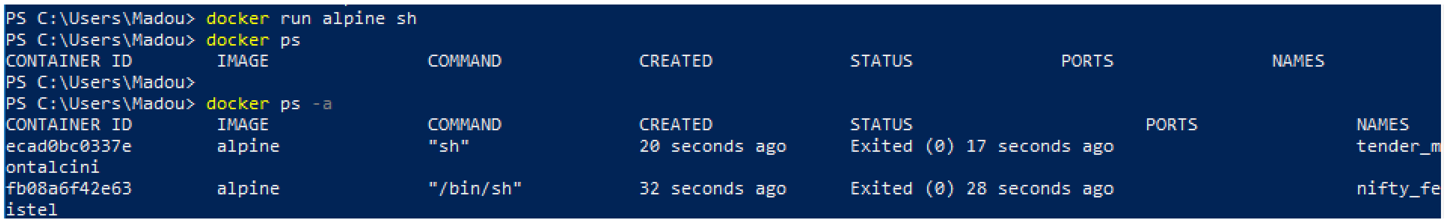
```
PS C:\Users\Madou> docker pull alpine
Using default tag: latest
latest: Pulling from library/alpine
4fe2ade4980c: Pull complete
Digest: sha256:621c2f39f8133acb8e64023a94dbdf0d5ca81896102b9e57c0dc184cadaf5528
Status: Downloaded newer image for alpine:latest
```

Now we can see the new images



```
PS C:\Users\Madou> docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
alpine              latest              196d12cf6ab1        3 months ago        4.41MB
```

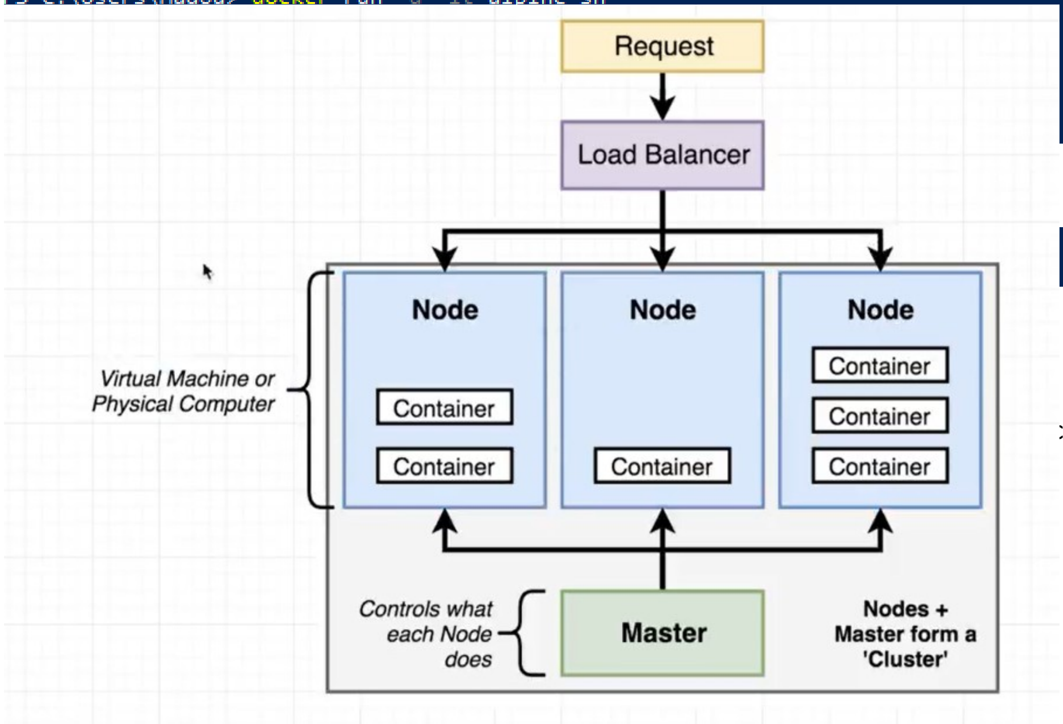
Now if we tried to run the alpine image it will be closed just after running it, because we didn't run the docker container in the interacting mode



```
PS C:\Users\Madou> docker run alpine sh
PS C:\Users\Madou> docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
PS C:\Users\Madou>
PS C:\Users\Madou> docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS                      PORTS               NAMES
ecad0bc0337e        alpine              "sh"                20 seconds ago      Exited (0) 17 seconds ago                       tender_m
ontalcini
fb08a6f42e63        alpine              "/bin/sh"           32 seconds ago      Exited (0) 28 seconds ago                       nifty_fe
istel
```

So we need to run this command instead "docker run -d -it alpine sh" and then "docker run -it alpine sh". Now we can run linux commands:

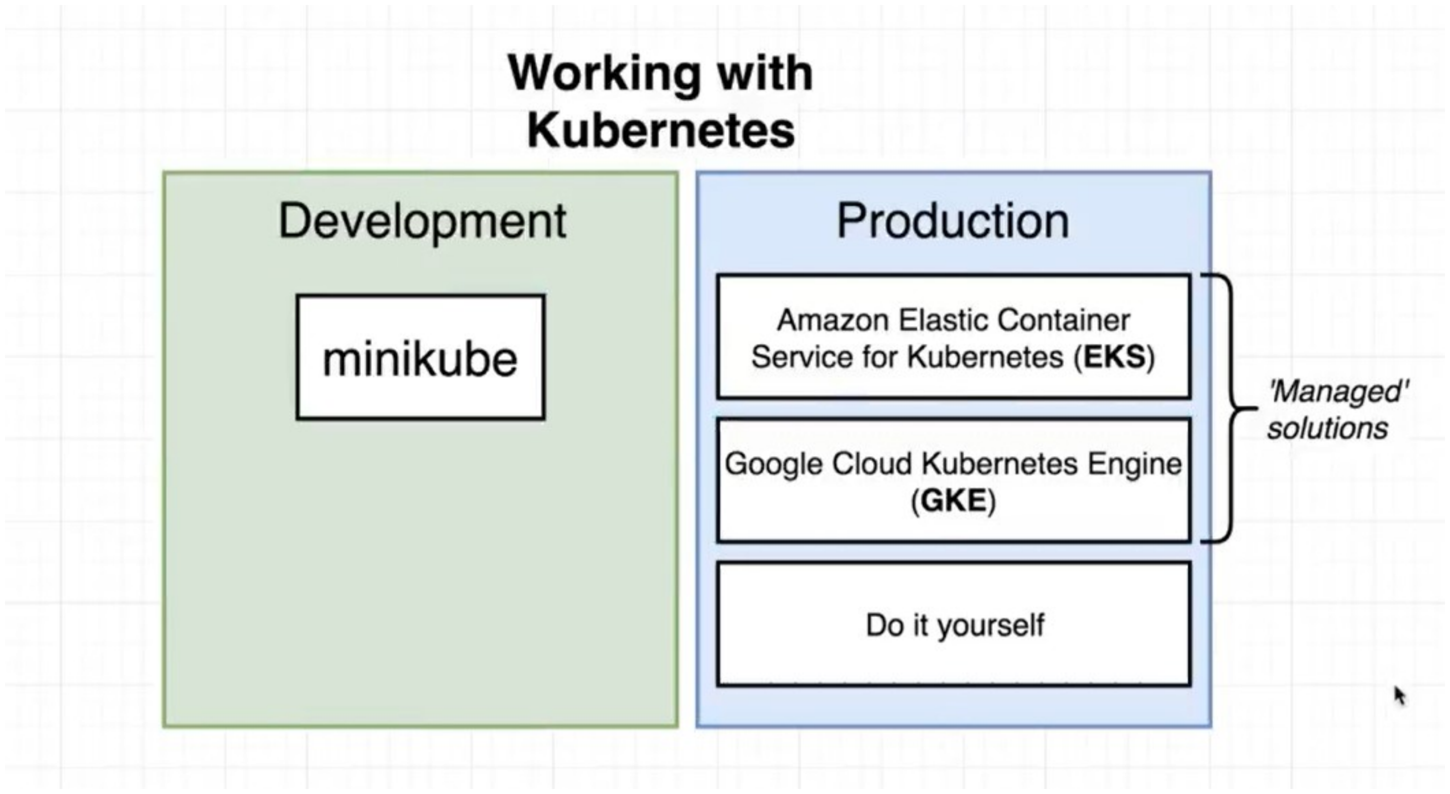
```
PS C:\Users\Madou> docker run -d -it alpine sh
                                                   sys       tmp       usr       var
```

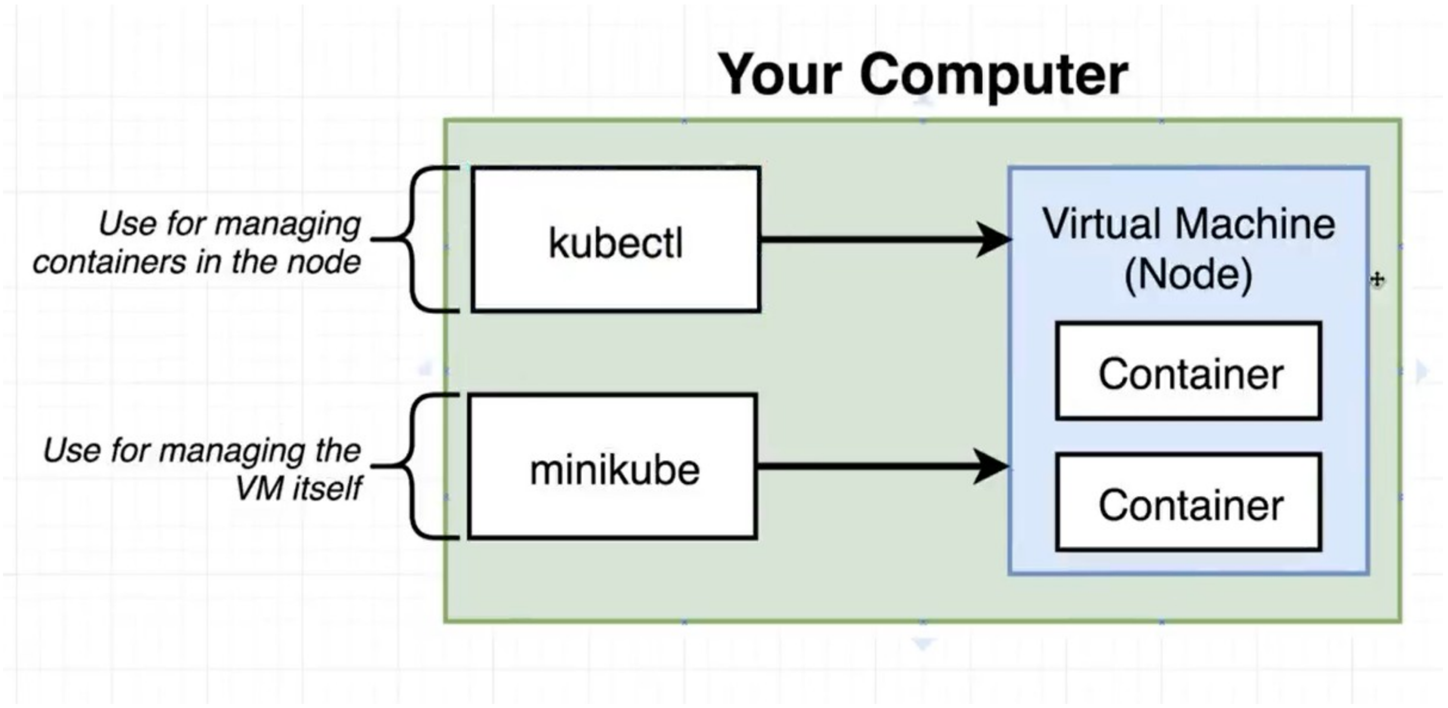

Request
Load Balancer
Node
Node
Node
Container
Container
Container
Container
Container
Container
Virtual Machine or Physical Computer
Controls what each Node does
Master
Nodes + Master form a 'Cluster'

hines

This figure is a diagram of a kubernetes cluster. It contains a certain number of nodes that represent a virtual machine or a physical computer, these nodes are controlled by a master. Then we have the load balancer that receives request and assign them to nodes

Now let's move on to how we can use kubernetes. Working with kubernetes can be divided into two big categories, working locally in the development process or during the production, the next figure can explain what each one needs


Working with Kubernetes
Development
minikube
Production
Amazon Elastic Container Service for Kubernetes (EKS)
Google Cloud Kubernetes Engine (GKE)
Do it yourself
'Managed' solutions


Your Computer
Use for managing containers in the node
kubectl
Use for managing the VM itself
minikube
Virtual Machine (Node)
Container
Container

We are going to use kubectl any time that we want to manage what the node is doing, but minikube is used to just create and run a local cluster.