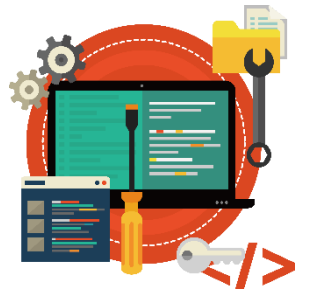




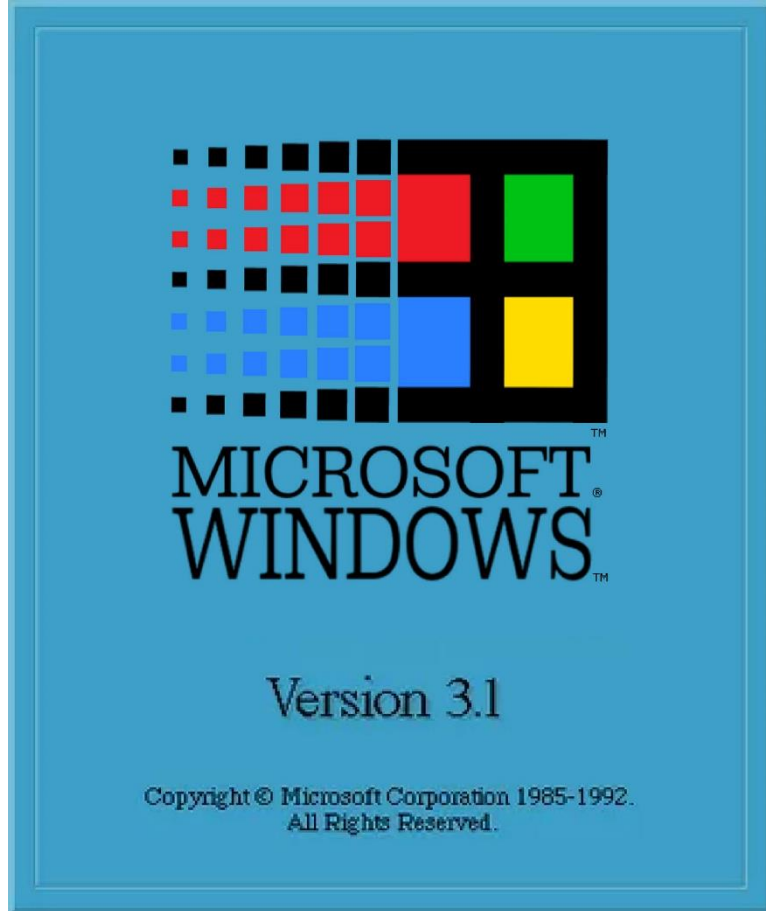
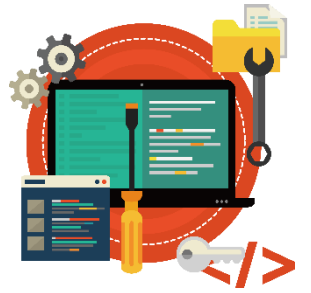
Systems & DevOps

The Contents



- Operating systems:
 - The role of an OS
 - Windows & Linux
 - Client & Server OS
 - Interacting with the OS via commands
- Virtualisation and the Datacenter:
 - Virtualisation
 - IaaS, PaaS, SaaS
 - Containers
- DevOps:
 - Culture
 - CI/CD pipelines
 - Infrastructure as Code

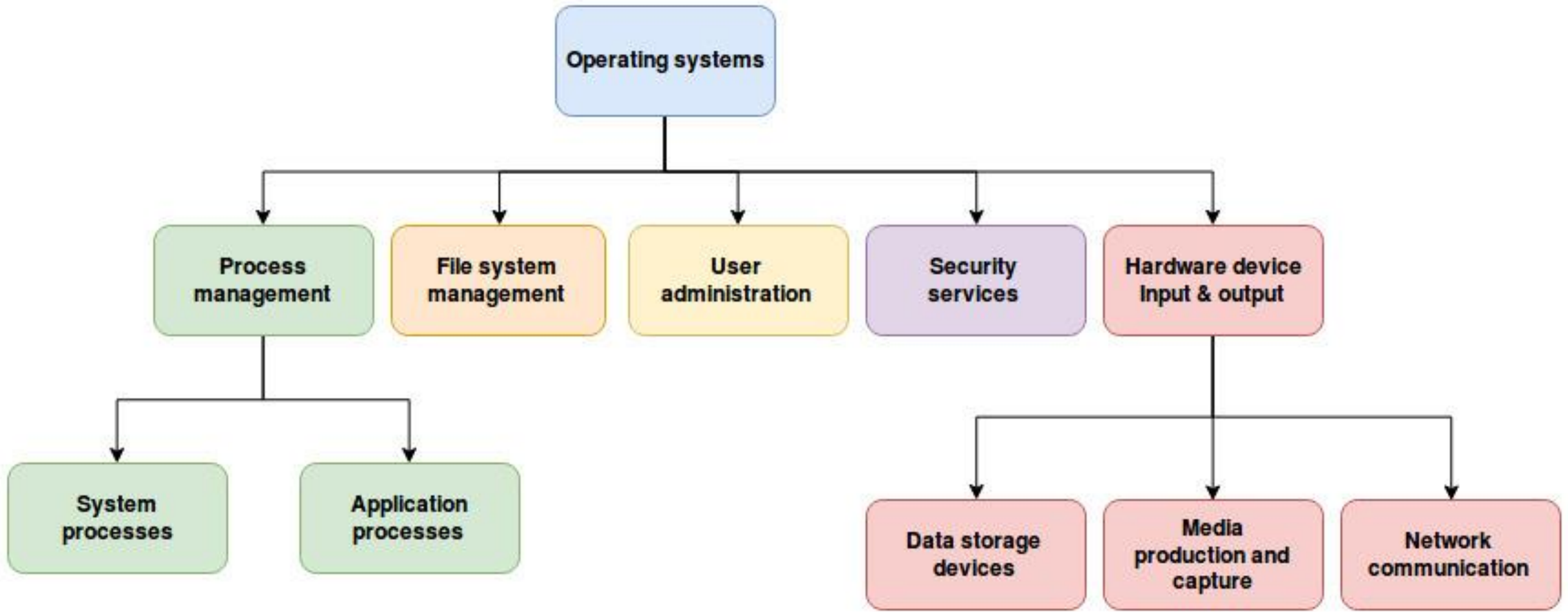
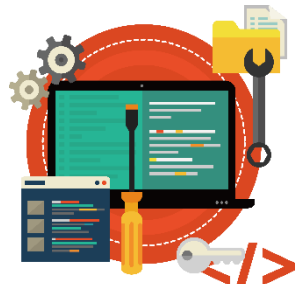
The role of an Operating System



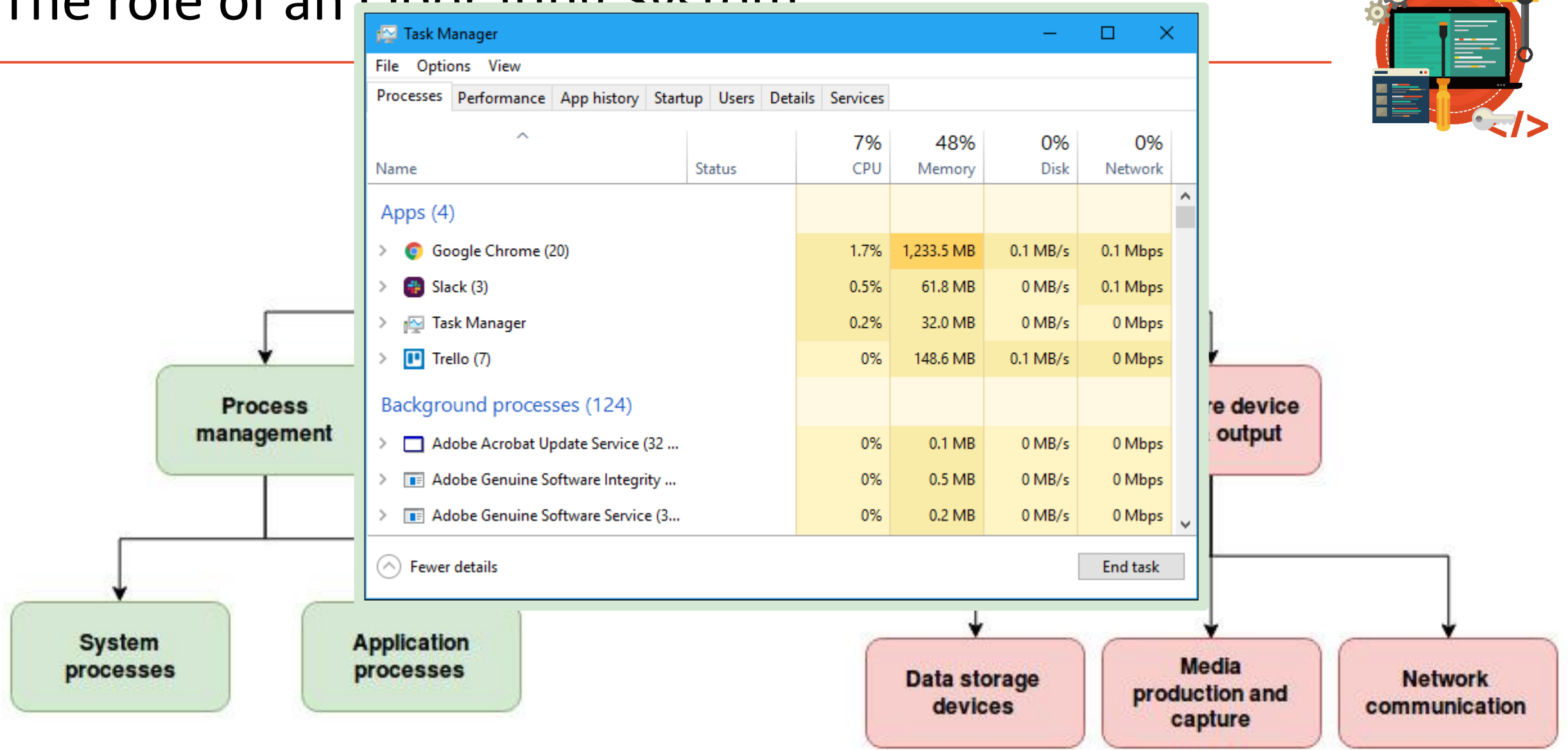


Try out the old
Windows!

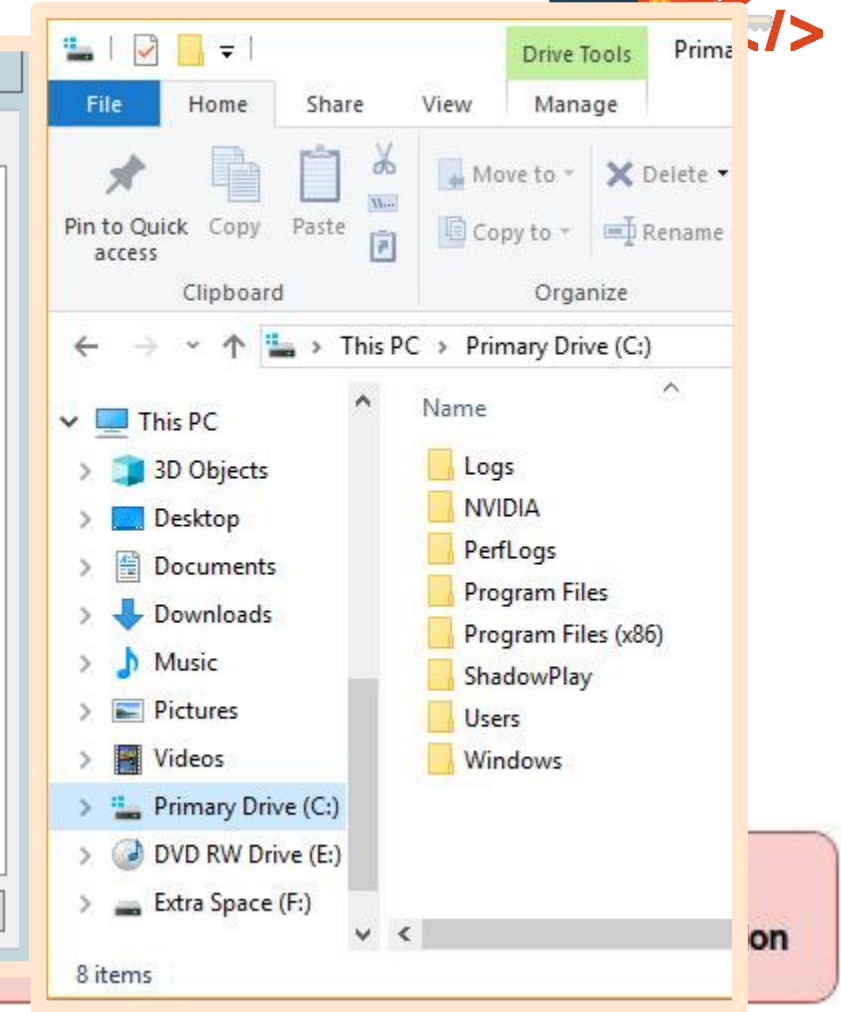
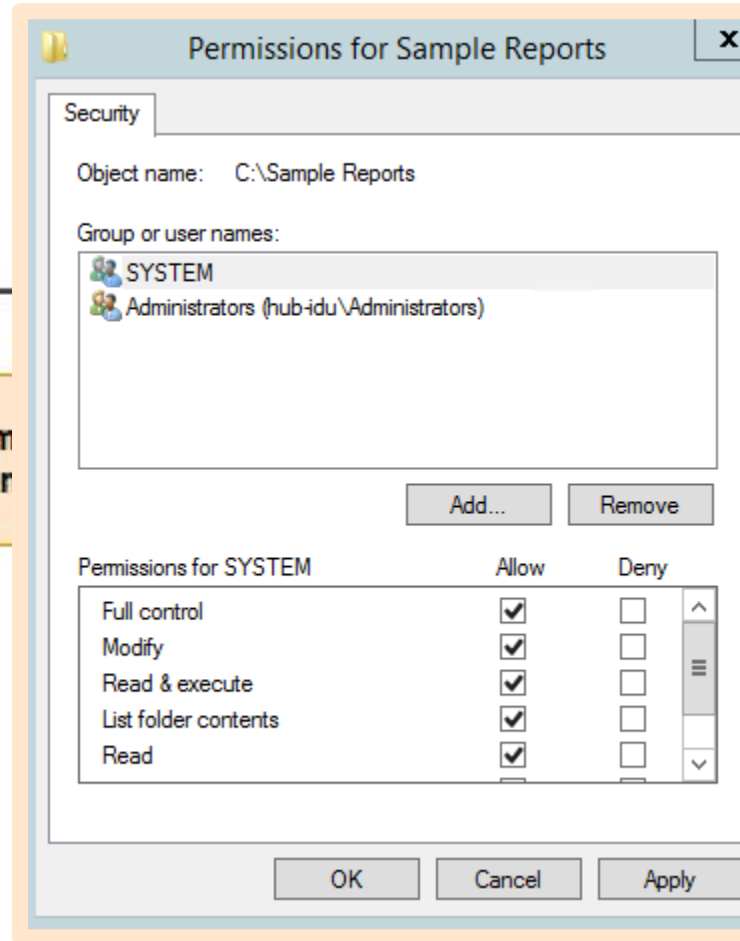
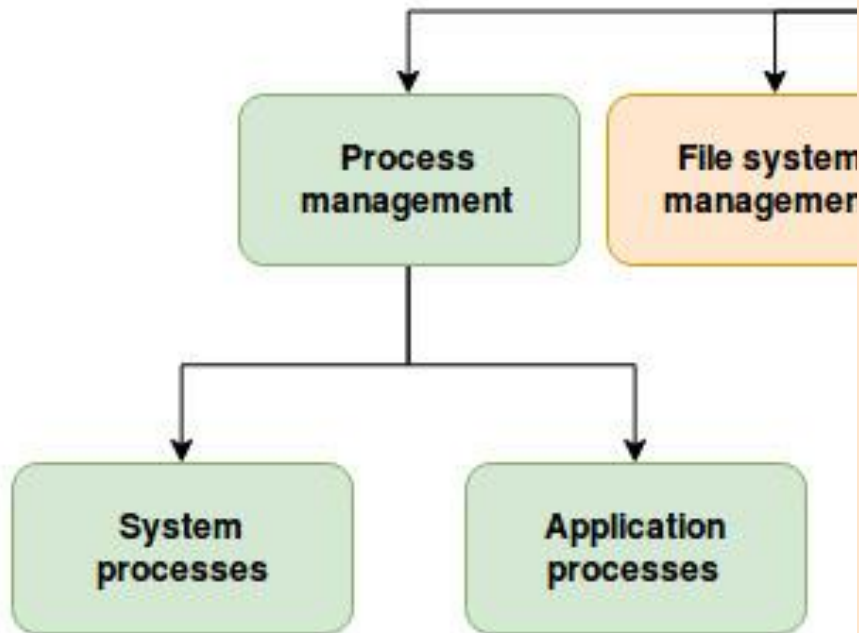
The role of an Operating System



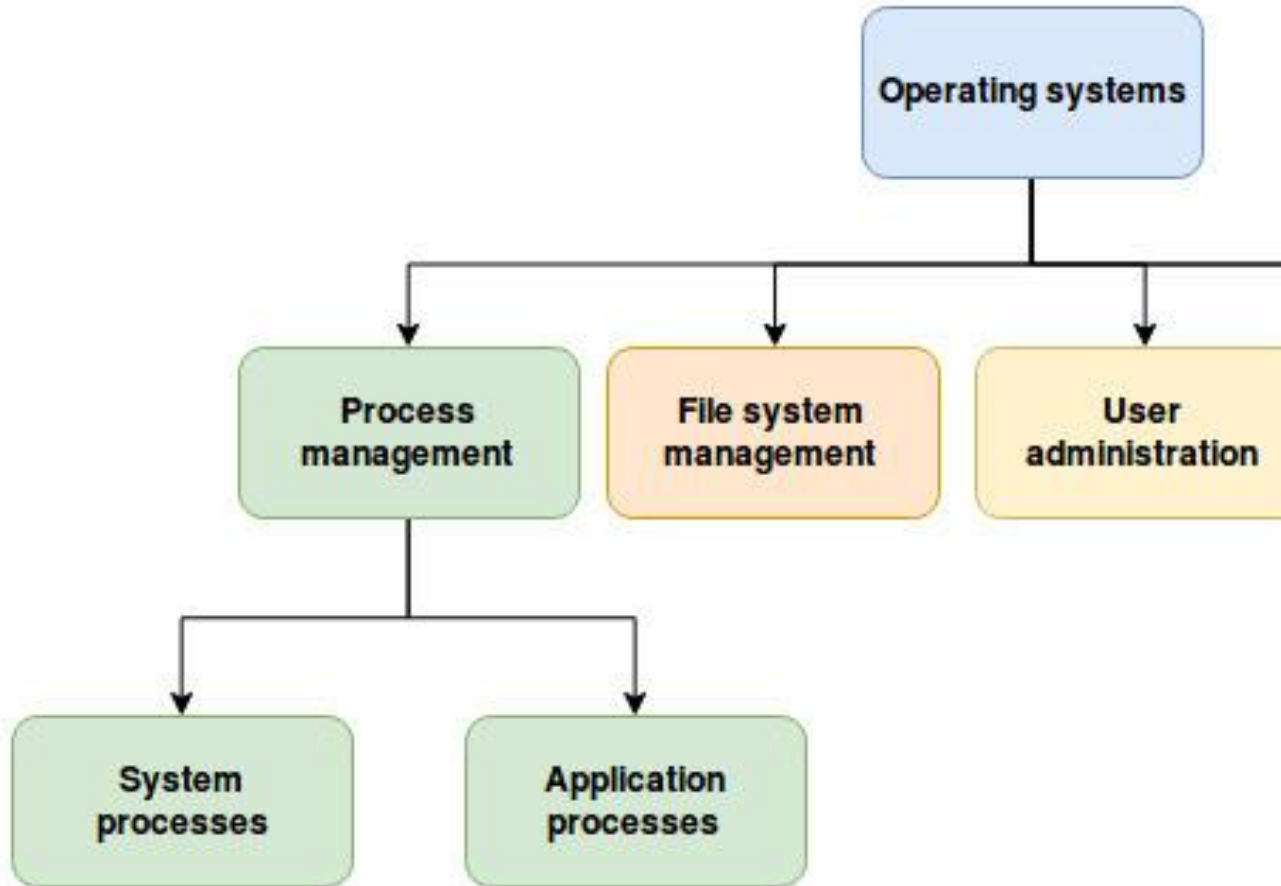
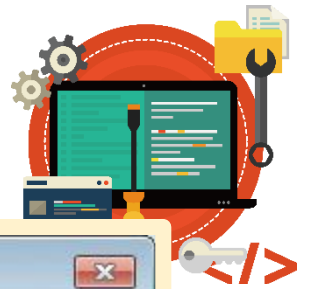
The role of an Operating System



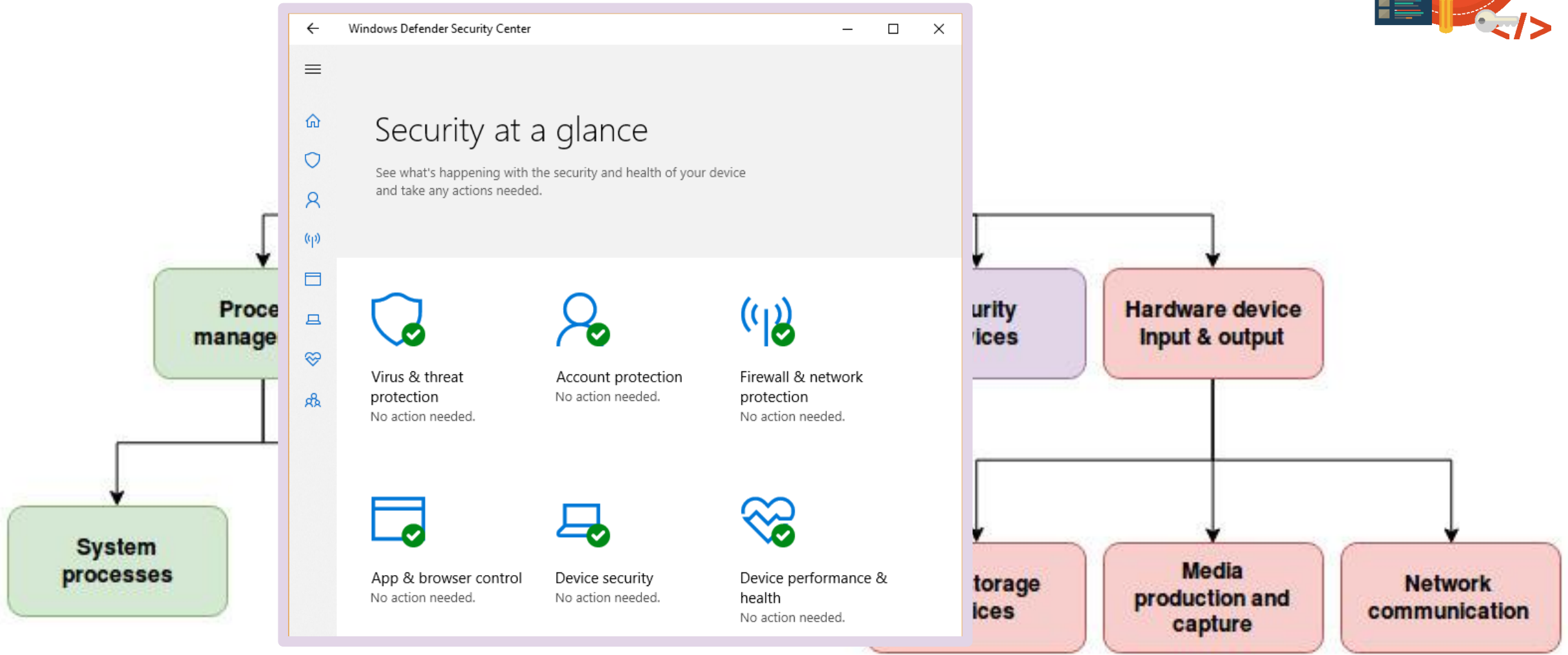
The role of an Operating System



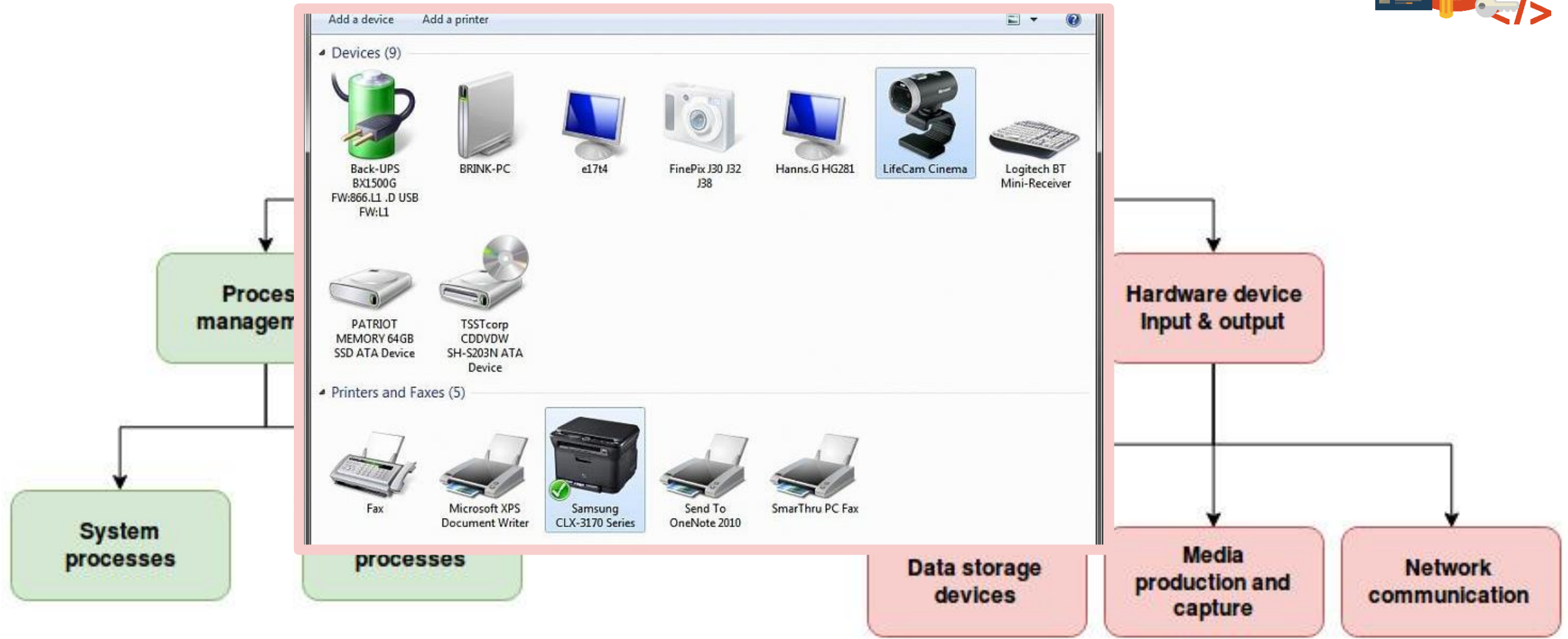
The role of an Operating System



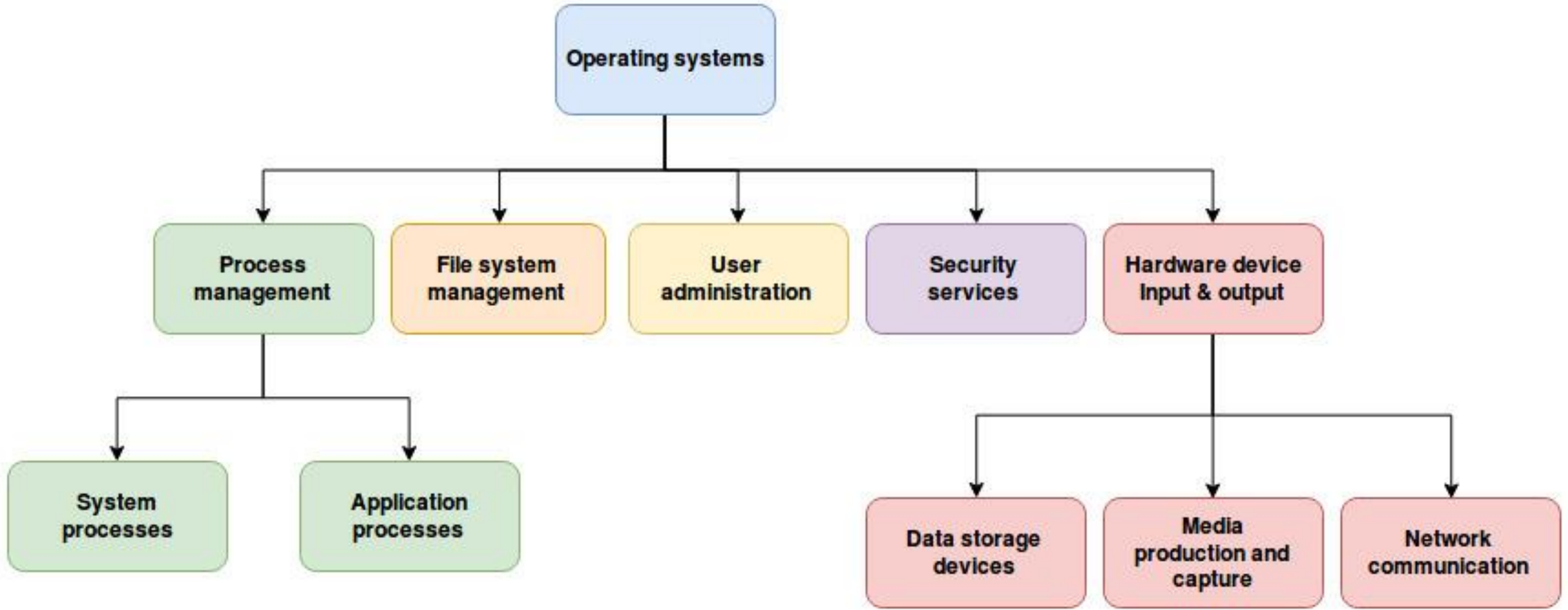
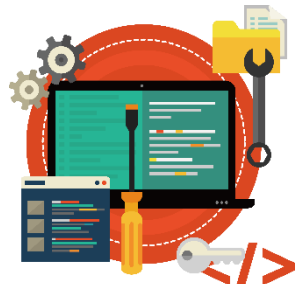
The role of an Operating System



The role of an Operating System



The role of an Operating System





Home

Install Ubuntu 21.-
10



Home

Install Ubuntu 21.-
10

Linux Operating System

What & Why?

Ubuntu Review

Other Distributions



Describe your local
company network

Client OS & Server OS



Add Roles and Features Wizard

Select server roles

DESTINATION SERVER
With ServerManager app get server roles details

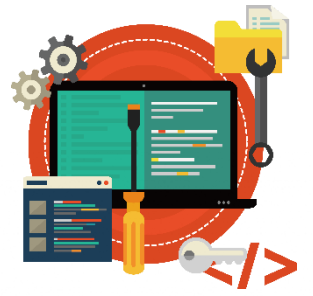
Before You Begin
Installation Type
Server Selection
Server Roles
Features
Application Server
Role Services
Web Server Role (IIS)
Role Services
Confirmation
Results

Select one or more roles to install on the selected server.

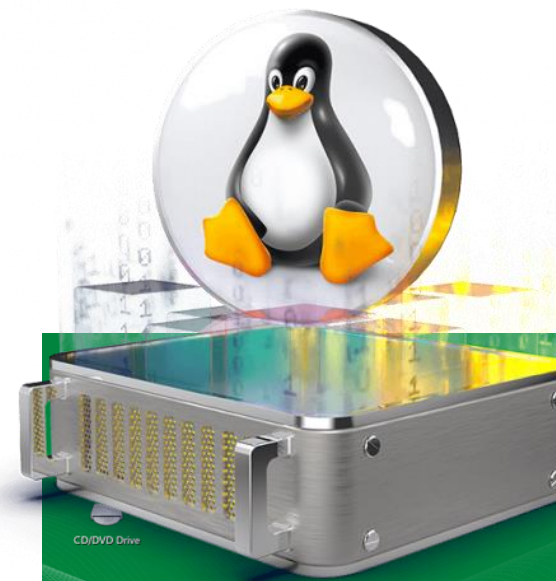
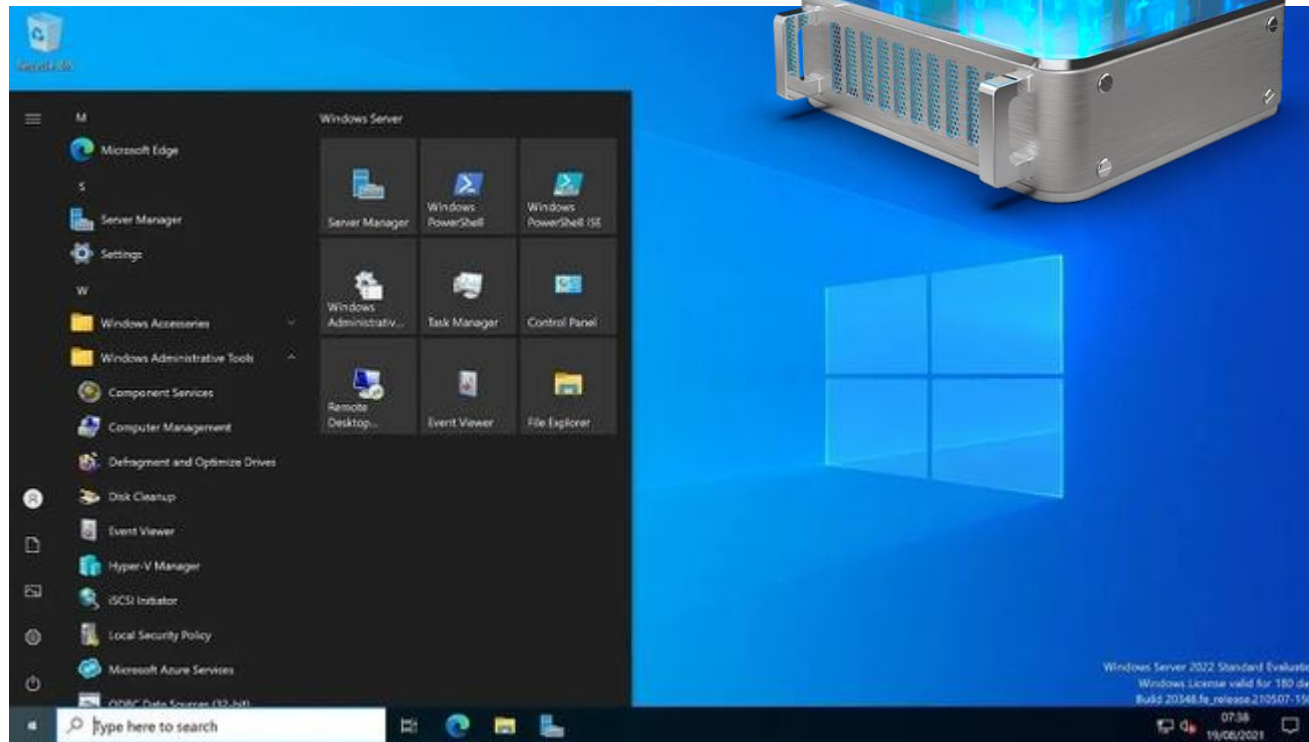
Roles	Description
<input type="checkbox"/> Active Directory Lightweight Directory Services	
<input type="checkbox"/> Active Directory Rights Management Services	
<input checked="" type="checkbox"/> Application Server	
<input type="checkbox"/> DHCP Server	
<input type="checkbox"/> DNS Server	
<input type="checkbox"/> Fax Server	
<input checked="" type="checkbox"/> File And Storage Services (Installed)	
<input type="checkbox"/> Hyper-V	
<input type="checkbox"/> Network Policy and Access Services	
<input type="checkbox"/> Print and Document Services	
<input type="checkbox"/> Remote Access	
<input type="checkbox"/> Remote Desktop Services	
<input type="checkbox"/> Volume Activation Services	
<input checked="" type="checkbox"/> Web Server (IIS)	Web Server (IIS) provides a reliable, manageable, and scalable Web application infrastructure.
<input type="checkbox"/> Windows Deployment Services	

< Previous Next > Install Cancel

Windows Server or Linux Server?

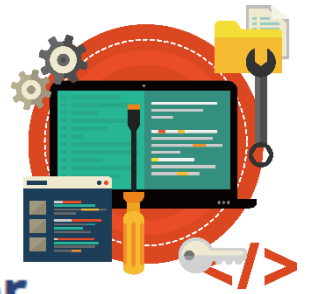


- Familiar GUI -> SME's
- Managing machines & users in the network



- The rest

Job openings



ICT-coördinator

IMMACULATA MARIA ROOSDAAL in ROOSDAAL

- een grondige kennis van het brede ICT-gebeuren (Windows, Office 365,...) en kennis inzake beheer van Office 365 en Teams.

Technisch ICT-Coördinator (deeltijds)

KUNSTHUMANORIA SINT-LUCAS in GENT

🕒 Online sinds 30 jun. 2022 - Vaste Job

Je beschikt over de nodige kennis van Windows server en Office 365. Termen als Azure AD en Endpoint klinken je niet vreemd in de oren.

Je hebt ervaring met en/of kennis van Hyper-V.

Allround Servicedesk Engineer

EAGLE IT in SCHOTEN

Een greep uit onze gebruikte technologieën: Windows 10 & 11, Windows Server (2008 – 2022), Microsoft Azure, Microsoft 365, Hyper-V, Active Directory, Kaseya, Apple (IOS). Werken met Hardware is uiteraard een deel van de job! Denk maar aan servers, firewalls, computers, NAS & wireless).

Verveling kennen we effectief niet.





The Role & Significance of Active Directory



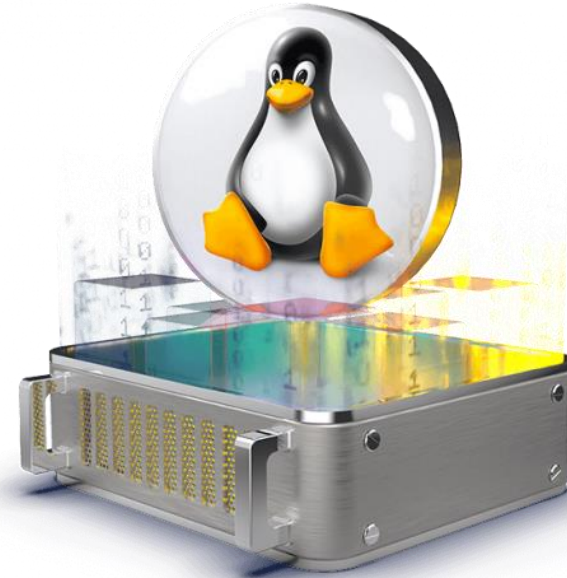
Job openings



Egov Select - Linux System Engineer

Egov Select in SCHAARBEEK

- U hebt een goede kennis van RedHat en CentOS
- U hebt een goede kennis van Virtualisation (intel based)
- U hebt een goede kennis van integratie, Storage, netwerk
- U hebt een goede kennis van HAProxy, NGINX
- Kennis in de volgende technologieën worden als troef beschouwd
 - Ansible en Ansible Tower
 - RedHat Satellitte
 - Docker Swarm
 - Kubernetes
 - CouchDb
 - Elastic search en ELK stack
 - GIT



Linux System Engineer

Smals in SINT-GILLIS

Je bent vertrouwd met de Linux-distributie Redhat (RHEL 6 en 7). Kennis van RHEL 8 is een pluspunt. Je hebt idealiter een goede kennis van scripting, Ansible, OpenStack, OpenShift, Docker en Kubernetes. Kennis van Zabbix, KVM en Puppet vormt ook een grote troef.

CLI = Command Line Interface



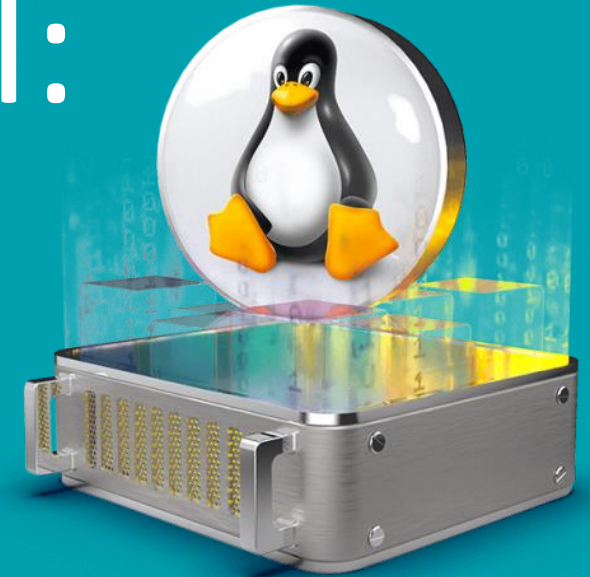
COMMAND LINE
INTERFACE



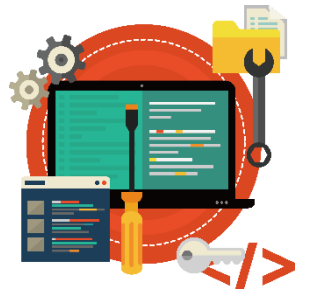
GRAPHICAL USER
INTERFACE



Try out a Linux in your
browser with bash CLI:
<https://webvm.io/>



PowerShell – The Microsoft shell CLI evolution



Jeffrey Snover:

We wanted to close the semantic gap between what admins thought and what they had to type.

<http://www.jsnover.com/blog/>, <http://powershell.org/wp/tag/jeffrey-snover/>

PowerShell – The Microsoft shell evolution



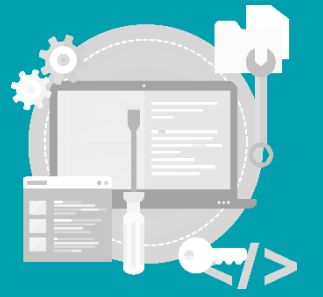
- Get files and folders:
 - DIR
 - Get-ChildItem
- Get all running processes:
 - Tasklist
 - Get-Process
- Get all services
 - Sc query
 - Get-Service
- Get all users in active directory
 - Dsget user ...
 - Get-ADuser

MS DoS:

- A lot of differently styled commands
- No filtering, only text-based
- Inconsistent use of parameters

PowerShell:

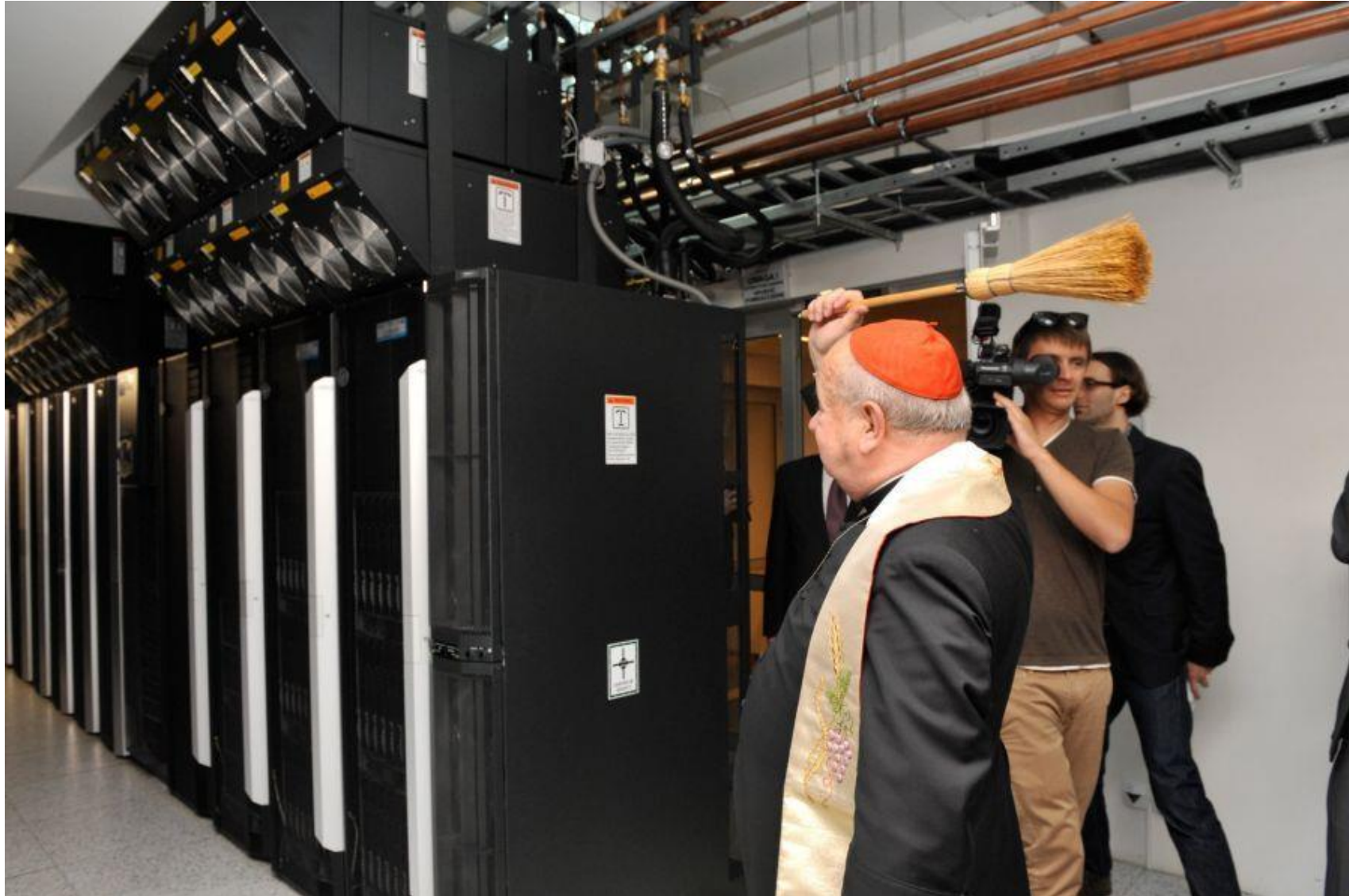
- The same cmdlet-structure
- Filtering on any property
- Consistent use of parameters



Hands-On PowerShell using the ISE



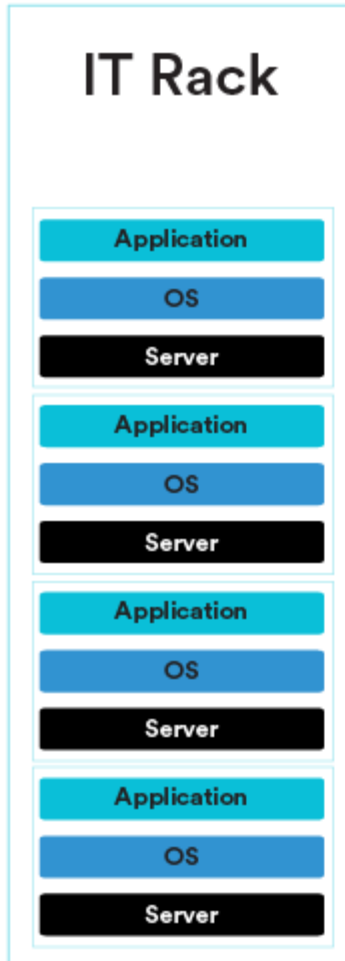
And then ... the datacenter



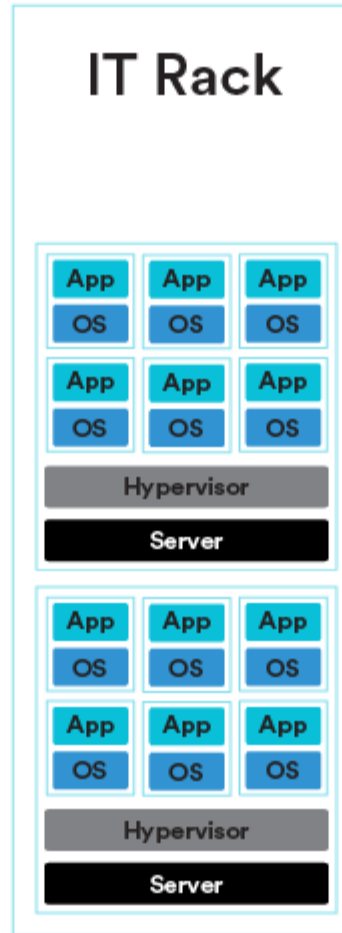
The rise of virtualisation



Early 2000s



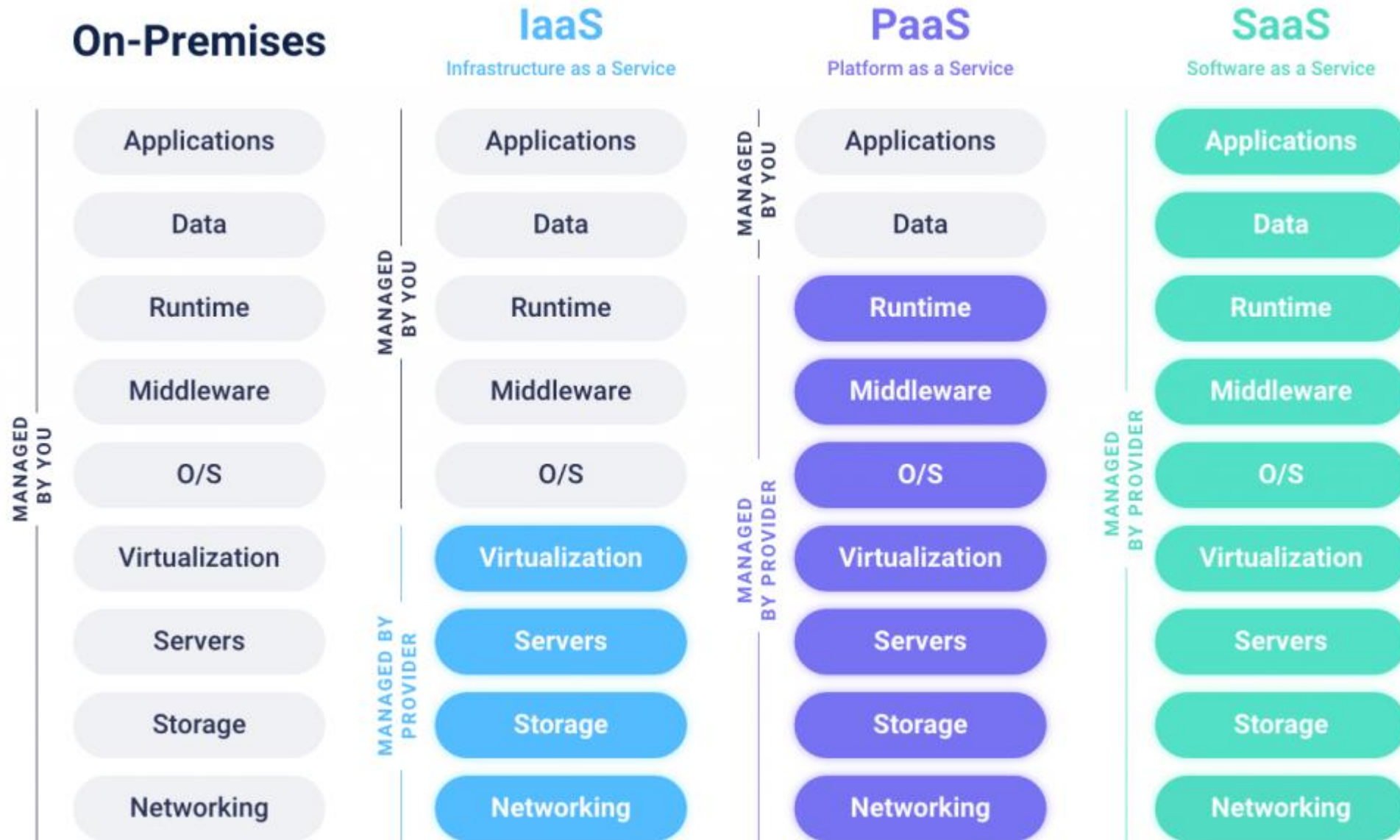
2005 to 2017



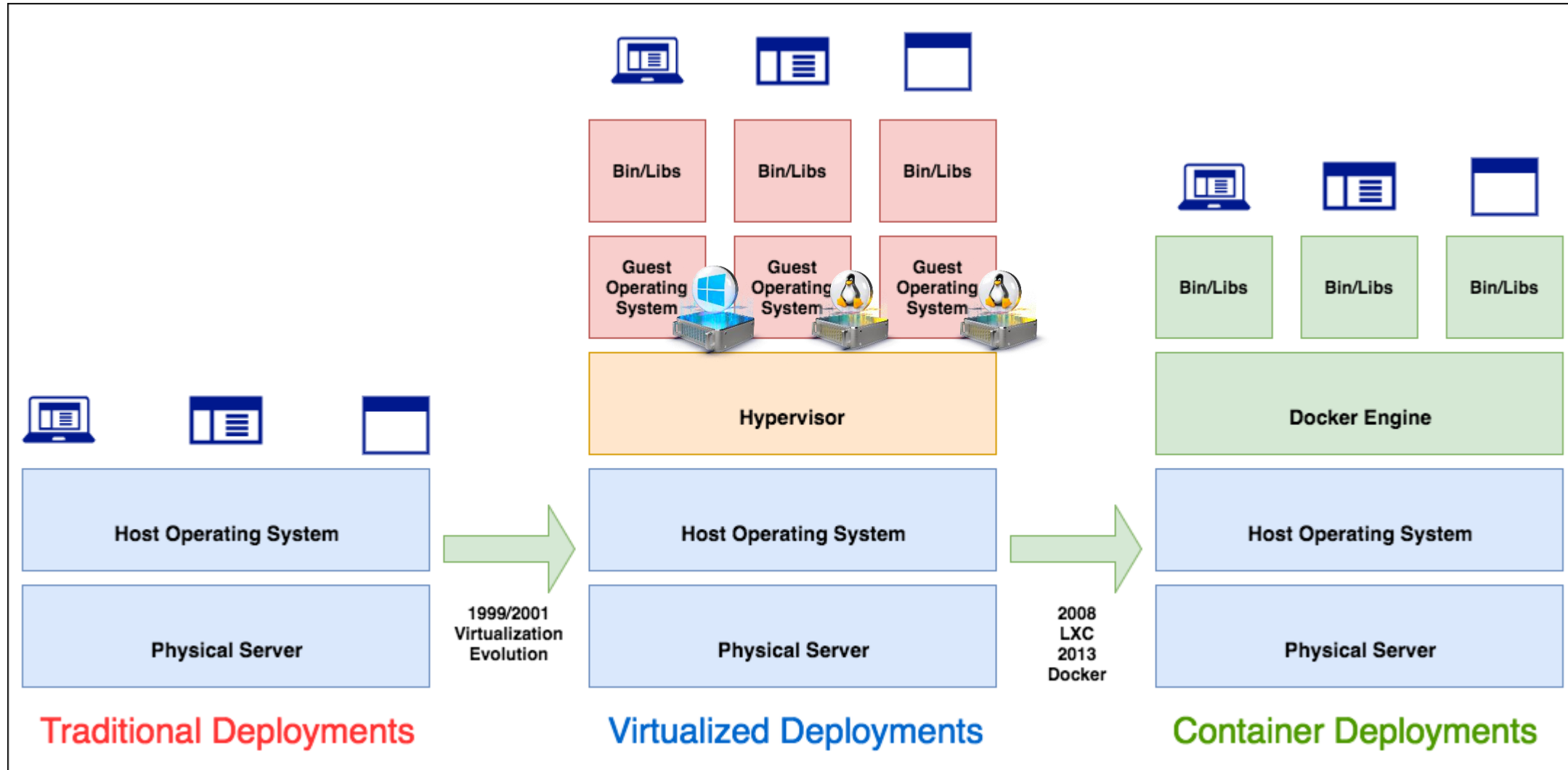
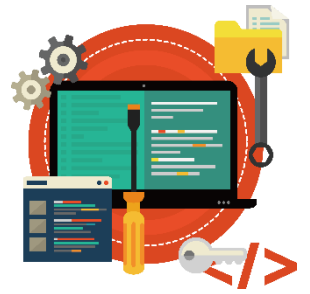
Future



Datacenter -> Cloud Service providers



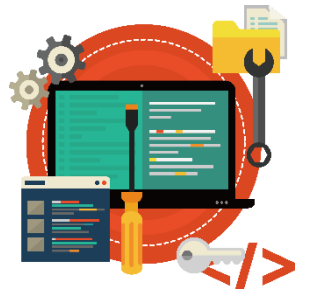
The rise of virtualisation - detail





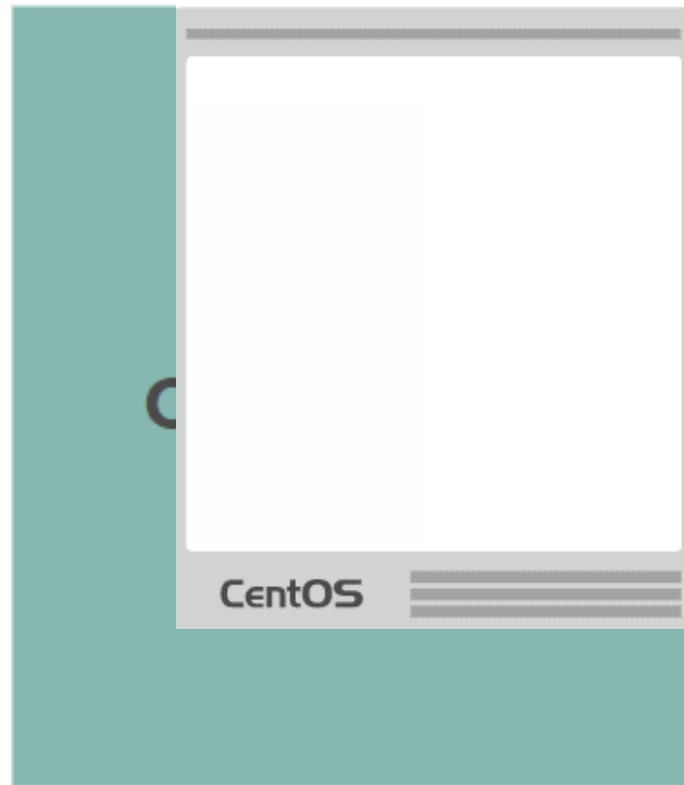
Look up some
'Hypervisors'

Docker: ONLY the app in a virtual container... why?



Virtual machine

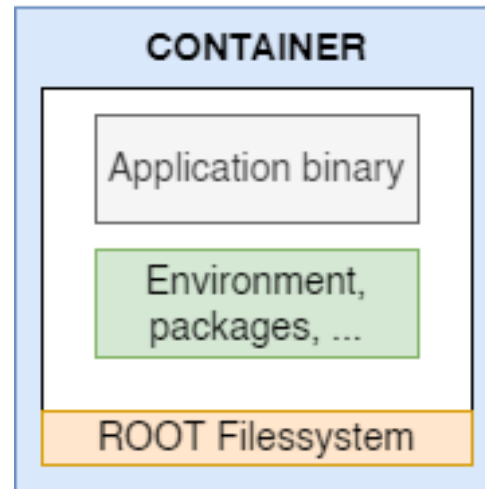
Compute resource



Docker: ONLY the app in a virtual container... what?



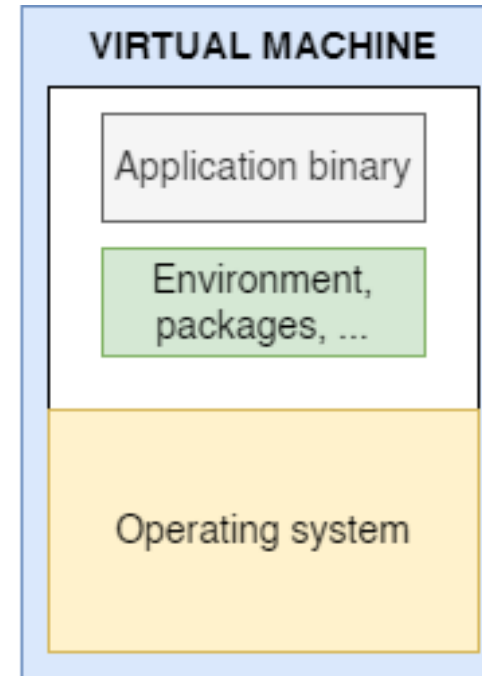
A container only has a **root filesystem** so that you can install and place applications inside of it. This together with all the files, packages, runtimes, ... the application needs to work.



Running engine (Docker)

Operating system

The host operating system has a **running engine** installed to be able to host and manage containers. For example **Docker**.



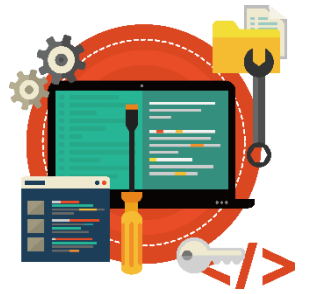
Hypervisor

Operating system

A virtual machine houses an **entire separate operating system** that works just like a physical machine.

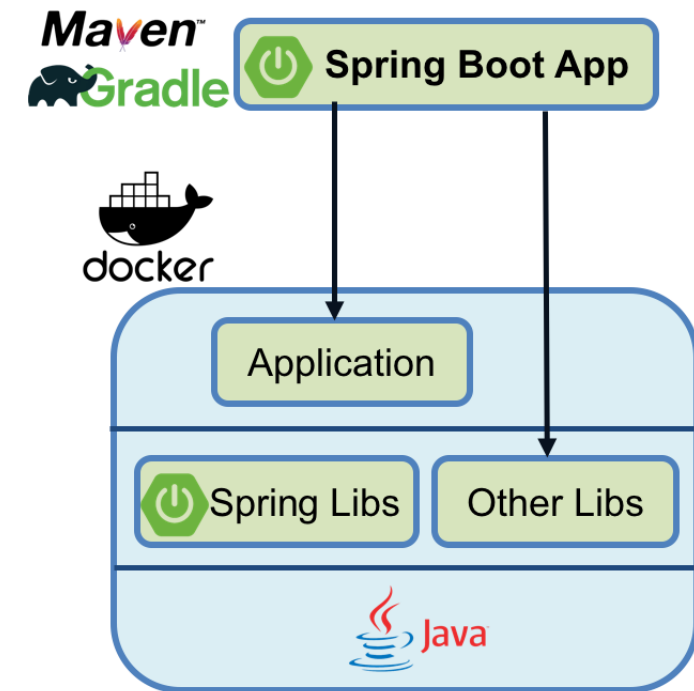
The host operating system has a **hypervisor** installed to be able to host and manage VMs.

Dockerfiles



5 lines (5 sloc) | 125 Bytes

```
1 FROM openjdk:8-jdk-alpine
2 EXPOSE 8052
3 ARG JAR_FILE=target/*.jar
4 ADD ${JAR_FILE} app.jar
5 ENTRYPOINT ["java","-jar","/app.jar"]
```

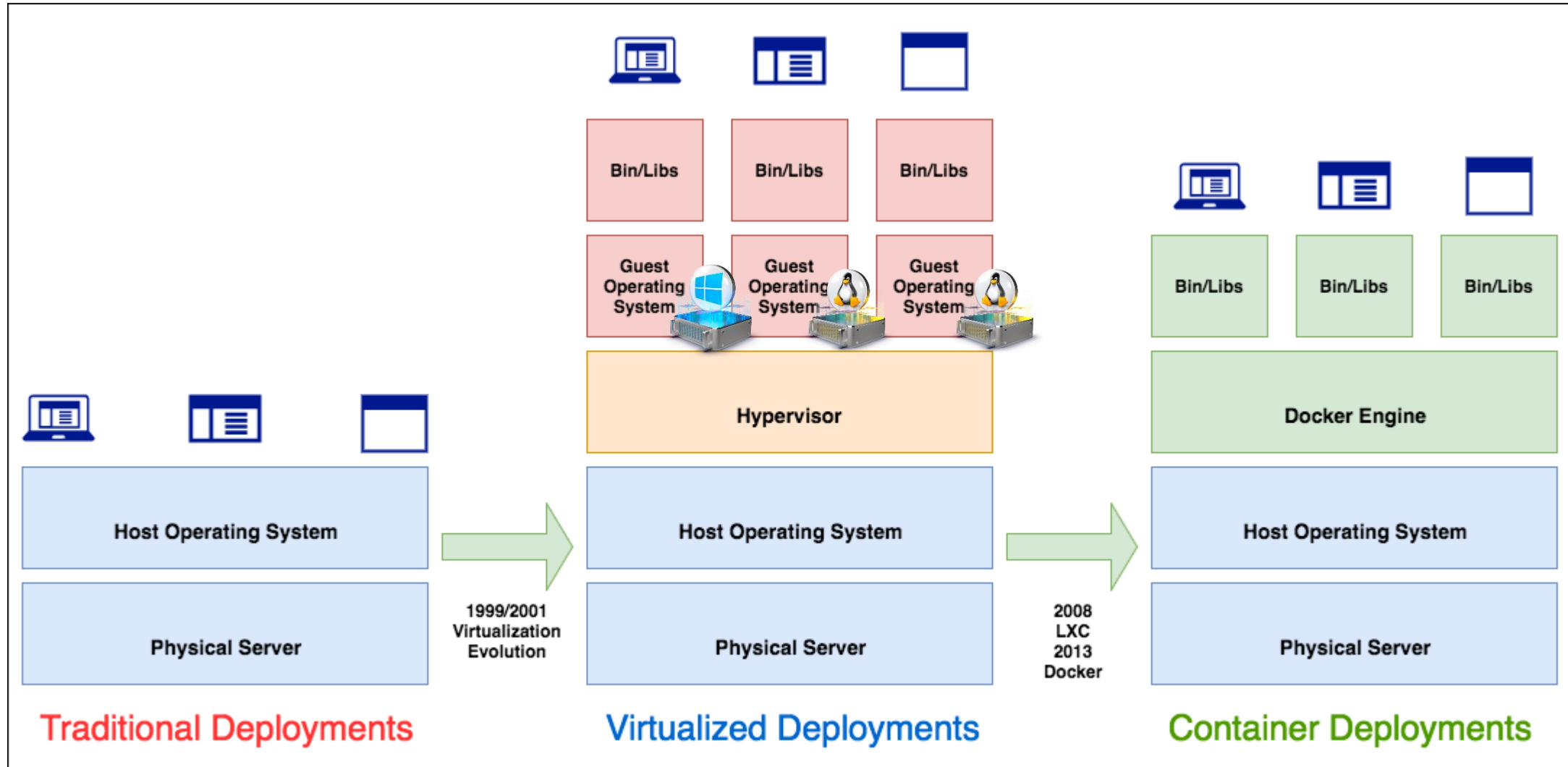
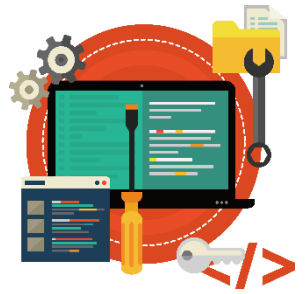




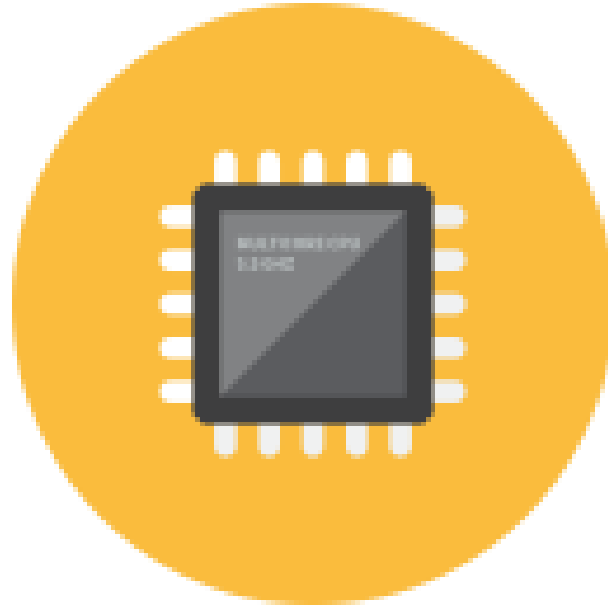
Docker Demo & Tryout:

- Register <https://hub.docker.com/signup>
- Try <https://training.play-with-docker.com/beginner-linux/>

The rise of virtualisation - detail



Where did it lead to?



**Computational power grows and
becomes cheaper**

Where did it lead to?



Storage becomes larger and cheaper. Servers transform into a 'pool'

Where did it lead to?

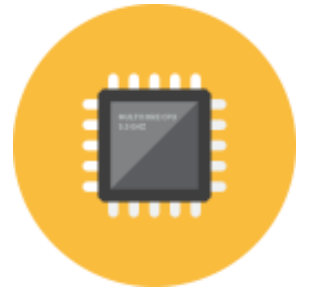
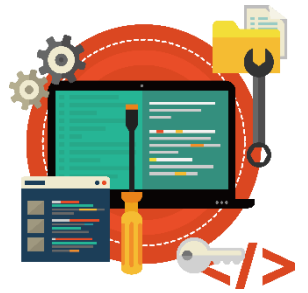


The Agile and DevOps way of working matures and gets used in more enterprises

Where did it lead to?

- The **Agile & DevOps** way of thinking is able to be transformed into a system that **continuously integrates, test code and adjusts planning boards** thanks to **cheaper storage** and **computational power**.
- **Continuous integration** is expanded with the possibility to automatically create **rehost and deliver** an application, which can then be to a new term: **Continuous delivery and continuous deployment**.
- **Centralized, pooled servers** make it possible to more efficiently **virtualise** applications using **containers** (Docker).

→ **DevOps culture &
Applications in DevOps CI/CD pipelines**



'DevOps culture' as Google sees it



Share ownership



Reduce
Organization Silos



SLOs & Blameless PMs



Accept
Failure as Normal



Reduce costs of failure



Implement
Gradual Change



Automate this year's job away



Leverage
Tooling & Automation



Measure toil and reliability



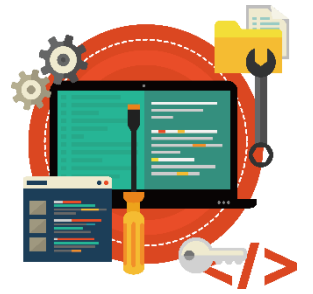
Measure
Everything



Plan

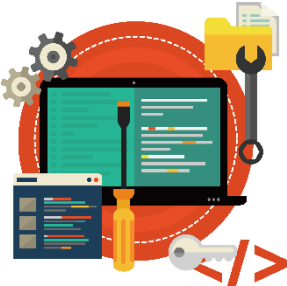


Trello



Basic DevOps pipeline

GitHub Environment ↔ Jira Environment



?

+

Q

◇

Settings

Add item

Repository

Issues

Components

Releases

Reports

Board

Backlog

Teams in Space

Software project

jira.teamsinspace.com

Board

Release

Quick Filters

TO DO 5

Engage Jupiter Express for outer solar system travel

SPACE TRAVEL PARTNERS

5

TIS-25

Create 90 day plans for all departments in the Mars Office

LOCAL MARS OFFICE

9

TIS-12

Engage Saturn's Rings Resort as a preferred provider

SPACE TRAVEL PARTNERS

3

TIS-17

Enable Speedy SpaceCraft as the preferred

SEESPACEEZ PLUS

IN PROGRESS 5

Requesting available flights is now taking > 5 seconds

SEESPACEEZ PLUS

3

TIS-8

Engage Saturn Shuttle Lines for group tours

SPACE TRAVEL PARTNERS

4

TIS-15

Establish a catering vendor to provide meal service

LOCAL MARS OFFICE

4

TIS-15

Engage Saturn Shuttle Lines for group tours

SPACE TRAVEL PARTNERS

CODE REVIEW 2

Register with the Mars Ministry of Revenue

LOCAL MARS OFFICE

3

TIS-11

Draft network plan for Mars Office

LOCAL MARS OFFICE

3

TIS-15

DONE 8

Homepage footer uses an inline style - should use a class

LARGE TEAM SUPPORT

1

TIS-68

Engage JetShuttle SpaceWays for travel

SPACE TRAVEL PARTNERS

5

TIS-23

Engage Saturn Shuttle Lines for group tours

SPACE TRAVEL PARTNERS

1

TIS-15

Establish a catering vendor to provide meal service

LOCAL MARS OFFICE

Plan	Code commit & automation
------	--------------------------



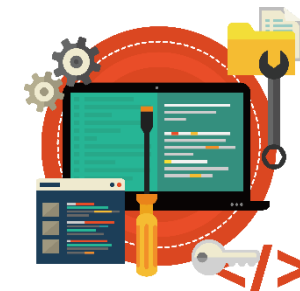
Trello



GitHub



GitHub
Actions



Basic DevOps pipeline

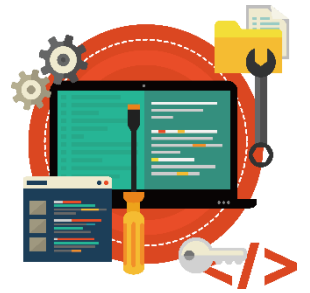
GitHub Actions: Infrastructure as Code



50 lines (48 sloc) | 1.37 KB

```
1  name: Test, Upload artifact and Deploy to Heroku
2
3  on: push
4
5  jobs:
6    test:
7      runs-on: ubuntu-latest
8      strategy:
9        matrix:
10         java: [ 8, 11 ]
11       name: Run unit tests on Java ${ matrix.java }
12       steps:
13         - uses: actions/checkout@master
14         - name: Setup java
15           uses: actions/setup-java@v1
16           with:
17             java-version: ${ matrix.java }
18         - run: mvn -f pom.xml clean test
19       release:
20         runs-on: ubuntu-latest
21         needs: test
22         name: Build, package and upload .jar artifact
23         steps:
24           - uses: actions/checkout@v1
25           - name: Set up JDK 8
26             uses: actions/setup-java@v1
27             with:
28               java-version: 8
29           - name: Build and package project
30             run: mvn -f pom.xml clean package
31           - name: Show contents of the current working directory
32             run: ls -la
33           - name: Show contents of the target directory
34             run: ls -la target
35           - name: Upload Maven build artifact
36             uses: actions/upload-artifact@v1
37             with:
38               name: artifact
39               path: ./target/spring-github-actions-demo-0.1.jar
```

Git management tool: GitKraken



The screenshot displays the GitKraken application interface. On the left, a sidebar shows repository statistics for 'vscode' on the 'master' branch, including 560/560 files, 405/405 remote files, 186 pull requests, 153/153 tags, 0 submodules, and 0 GitHub actions. The main area features a commit history timeline with various colored circles representing different branches and commits. A specific commit by Benjamin Pasero is highlighted, showing a merge of 'master' into 'electron-8.0.x'. The right panel displays the diff for this commit, listing changes to files like 'build/package.json', 'build/yarn.lock', and several source files in the 'src/vs' directory. The bottom status bar indicates the application is in 'PRO' mode, version 6.6.0, with a 100% zoom level and a 'Feedback' button.

GitHub Actions: Our example



The build-and-test.yml-file in our .github/workflows directory in our repository starts off like this:

```
1  name: Run tests on Java 8
2
3  on: push
4
5  jobs:
```

Plan	Code commit & automation
------	--------------------------



Trello



GitHub



GitHub Actions

Build

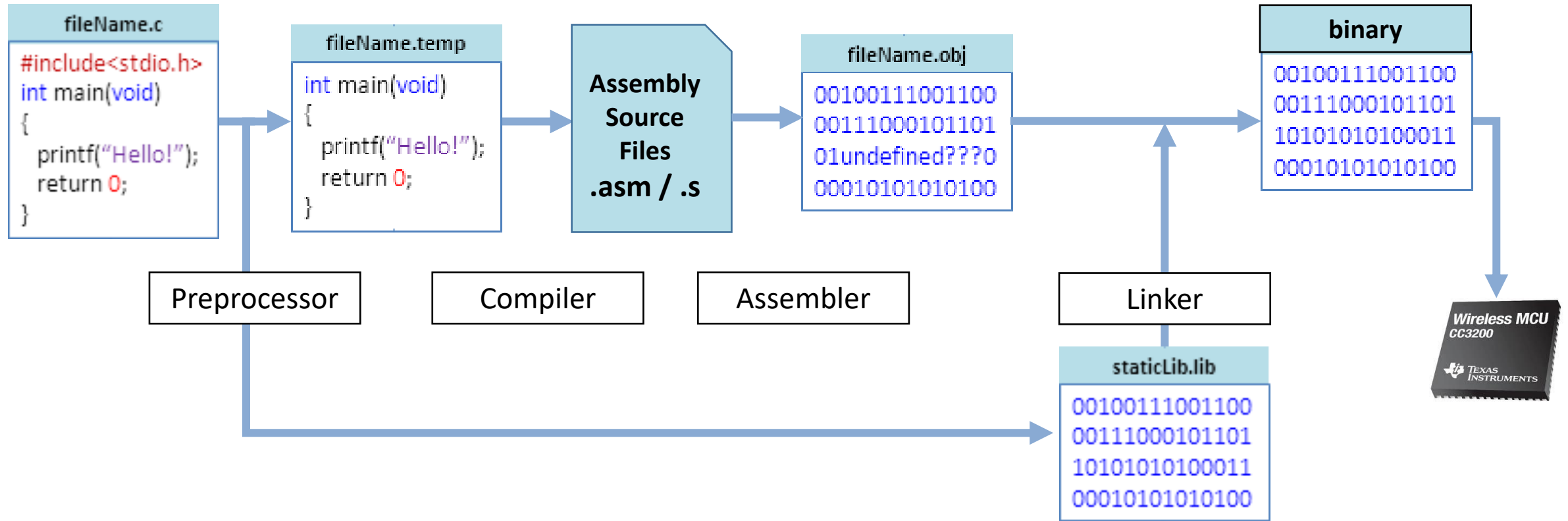


Basic DevOps pipeline

Building?



In essence: From code to a *binary* that a machine can run.



Building? 🛠️

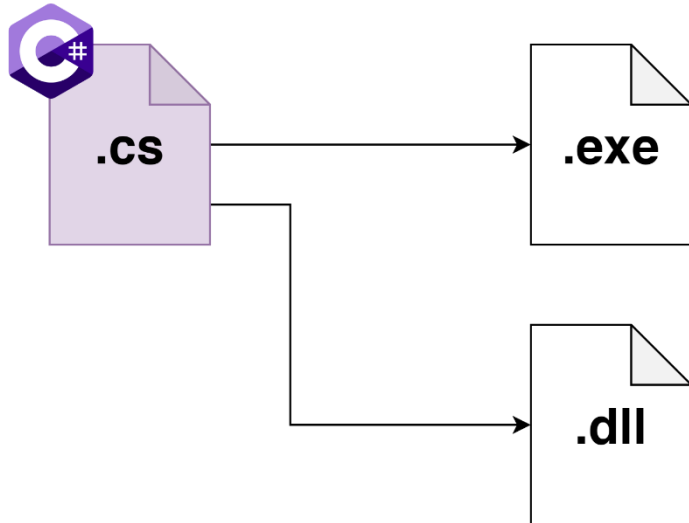


Different languages with their different compilers produce different *binaries*.



Java code can be compiled into .jar-files using a build command. These need to be run via cmd/bash:

```
C:\Users\user>java -jar C:\Users\user\Desktop\HelloWorld.jar  
Hello World
```



C# code can be compiled into .exe-file and .dll-files.

.exe-files can be run with a simple double-click in Windows.

.dll-files are run and used by other programs. Usually there is a single .exe-file that is coupled with many .dll-files.

Plan	Code commit & automation
------	--------------------------



Trello



GitHub



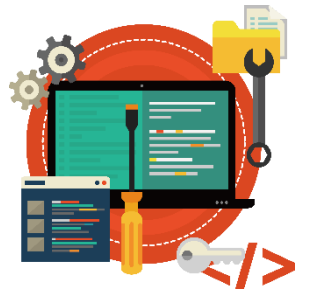
GitHub Actions

Build	Unit tests
-------	------------



Basic DevOps pipeline

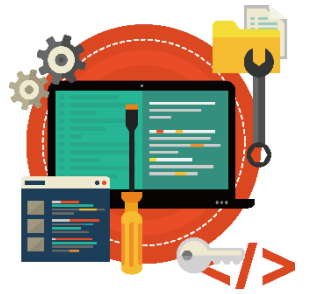
Unit tests



- Programming languages have their own (unit) testing frameworks:
 - **JUnit5** for Java, with Mockito, MockMVC, ... for creating stubs/mocks/fakes
 - **MSTest, VSTest**, ... for C# with Moq for creating stubs/mocks/fakes
 - **Mocha, Jasmine**, ... for JavaScript with Jest for creating stubs/mocks/fakes
 - ...
- Below an example of a Java unit test created with JUnit5 which tests the `.multiply()` method of the Calculator class, written inside a Java project

```
1. @Test
2.     void testMultiply() {
3.         assertEquals(20, calculator.multiply(4, 5));
4.     }
```

GitHub Actions: Our example – Building & Unit tests



```
1  name: Run tests on Java 8
2
3  on: push
4
5  jobs:
6    test:
7      runs-on: ubuntu-latest
8      name: Run unit tests on Java 8
9      steps:
10       - uses: actions/checkout@master
11       - name: Setup java
12         uses: actions/setup-java@v1
13         with:
14           java-version: 8
15       - run: mvn -f pom.xml test
```

<https://github.com/miverboven/spring-github-actions-demo/blob/master/.github/workflows/build-and-test.yml>



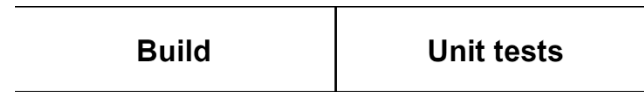
Trello



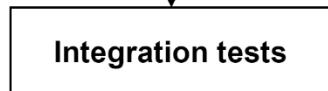
GitHub



GitHub
Actions



Towards Continuous Integration





Trello



GitHub



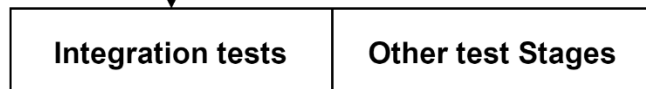
GitHub
Actions



Maven (Java)



JUnit 5 testing framework



JUnit 5 testing framework



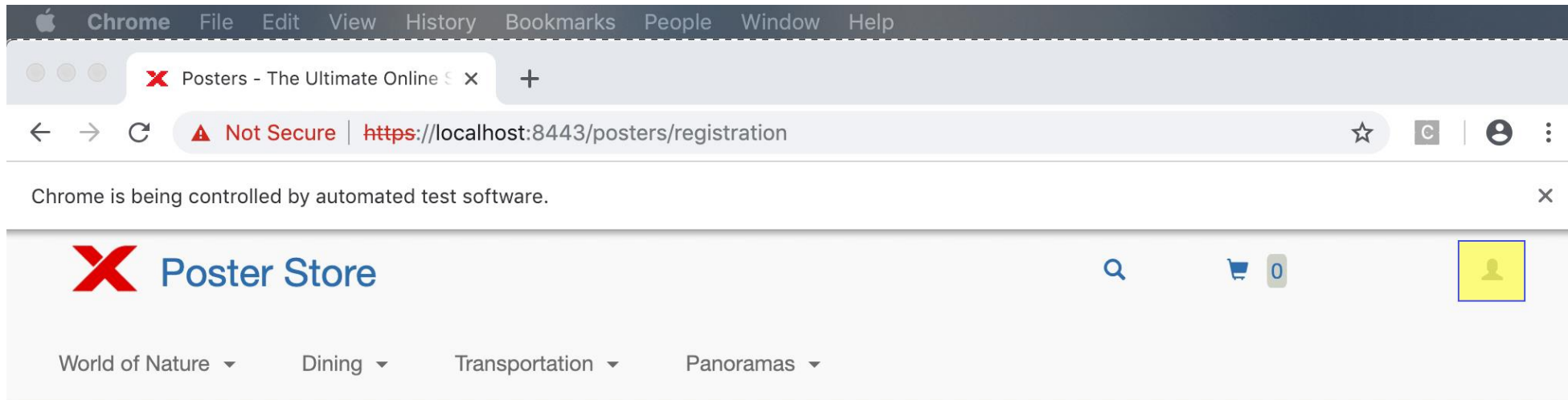
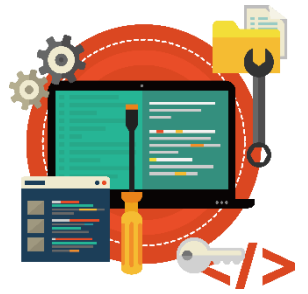
Locust
Load Testing



Selenium
Automated
In-browser testing

Towards Continuous Integration

Selenium Automated User (acceptance) Testing

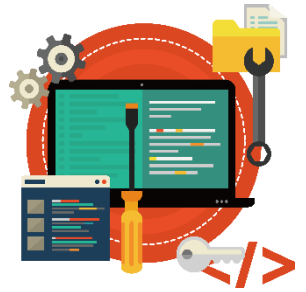
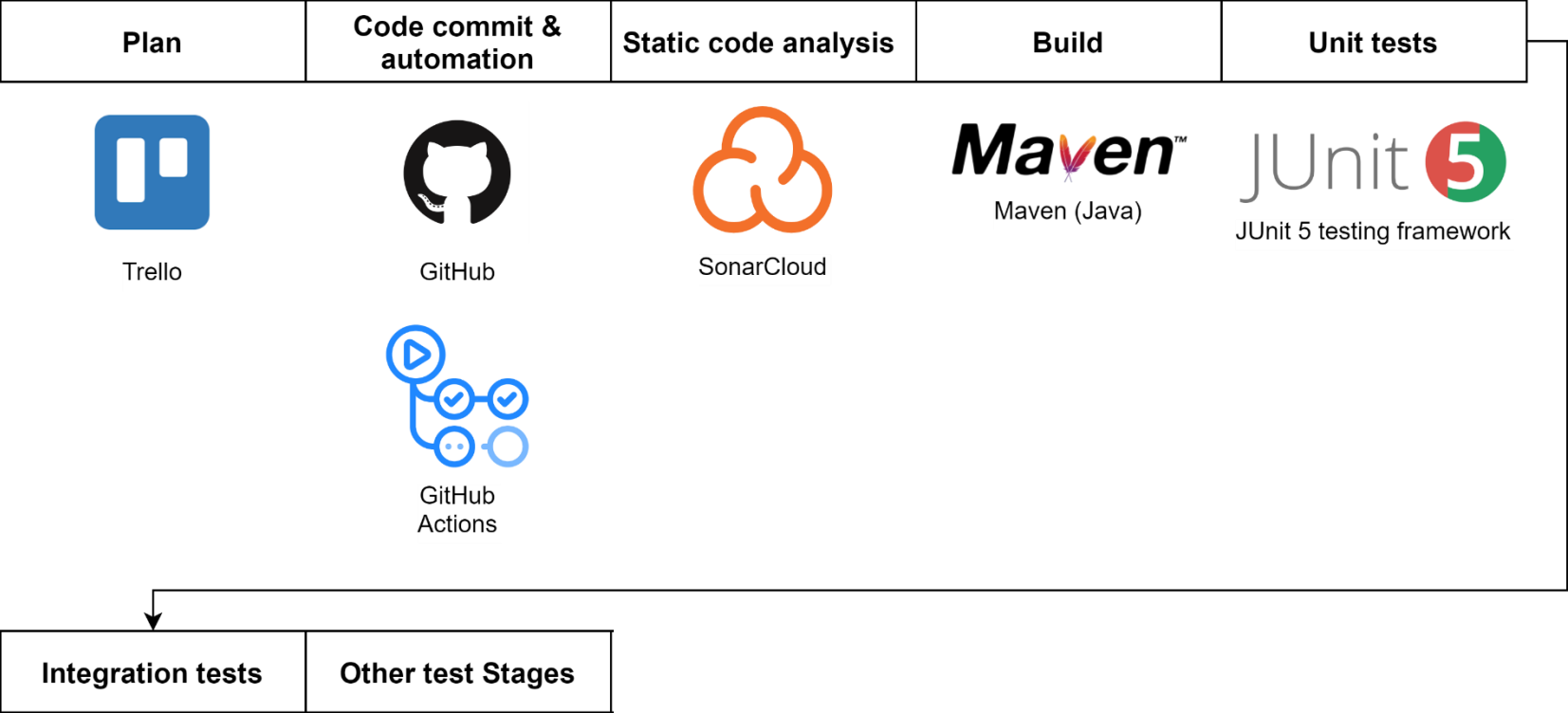


Create an account

Last name*	<input type="text" value="Last name"/>
First name*	<input type="text" value="First name"/>
Email address*	<input type="text" value="Email address"/>
Password*	<input type="password" value="Password"/>
Repeat password*	<input type="password" value="Confirm password"/>

* Required fields

Create account



Continuous Integration

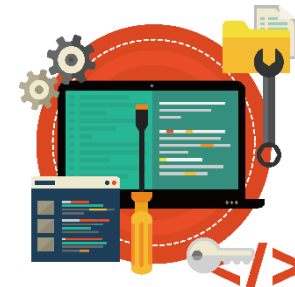
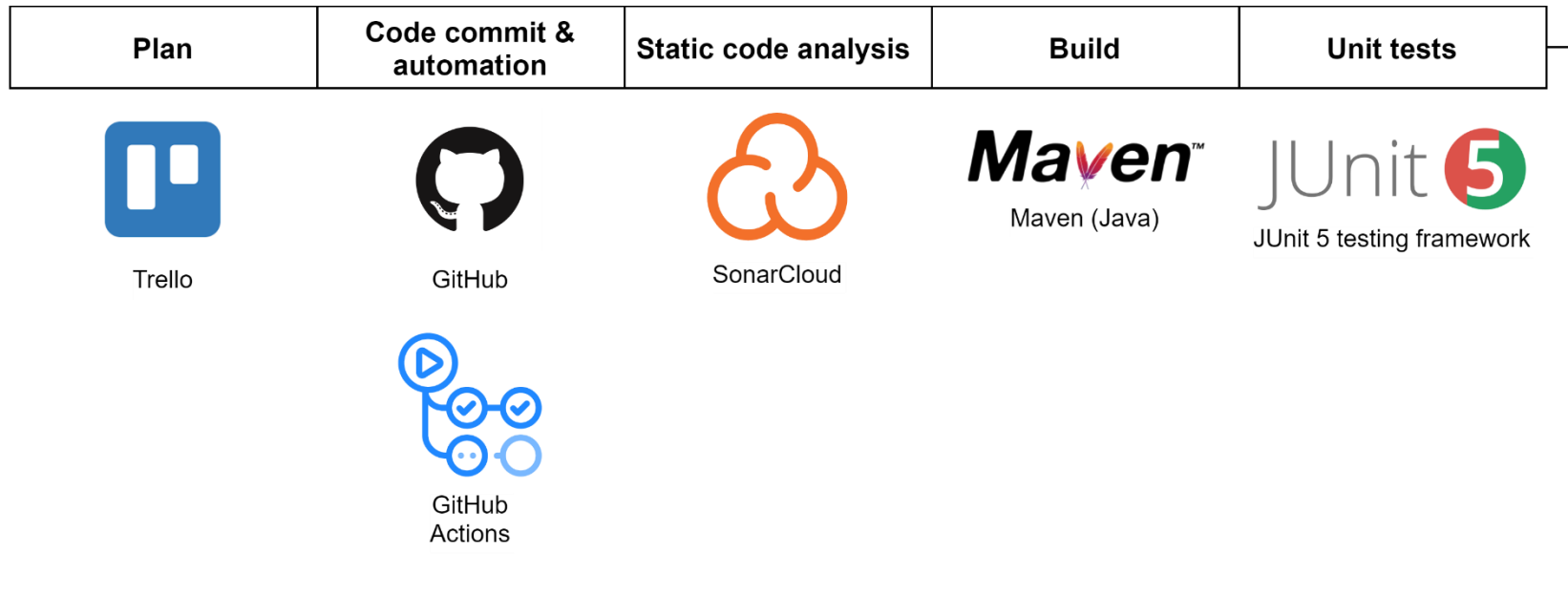
sonarcloud

 **0**  Bugs

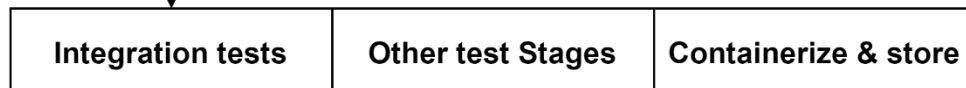
 **4**  Code Smells

 **0**  Vulnerabilities

 **1** Hotspots



Continuous Delivery



Locust
Load Testing



Selenium
Automated
In-browser testing



Docker Hub
Container Image
Library

We either produce:

- A binary
- A *container* 🤖?

That we store!

GitHub Actions: Our example – Continuous Delivery



```
1  name: Build and upload binary artifact
2
3  on: push
4
5  jobs:
6    release:
7      runs-on: ubuntu-latest
8      name: Build, package and upload .jar artifact
9      steps:
10     - uses: actions/checkout@v1
11     - name: Set up JDK 8
12       uses: actions/setup-java@v1
13       with:
14         java-version: 8
15     - name: Build and package project
16       run: mvn -f pom.xml package
17     - name: Upload Maven build artifact
18       uses: actions/upload-artifact@v1
19       with:
20         name: artifact
21         path: ./target/spring-github-actions-demo-0.1.jar
```

build-and-upload.yml

on: push



✓ Build, package and upload 27s



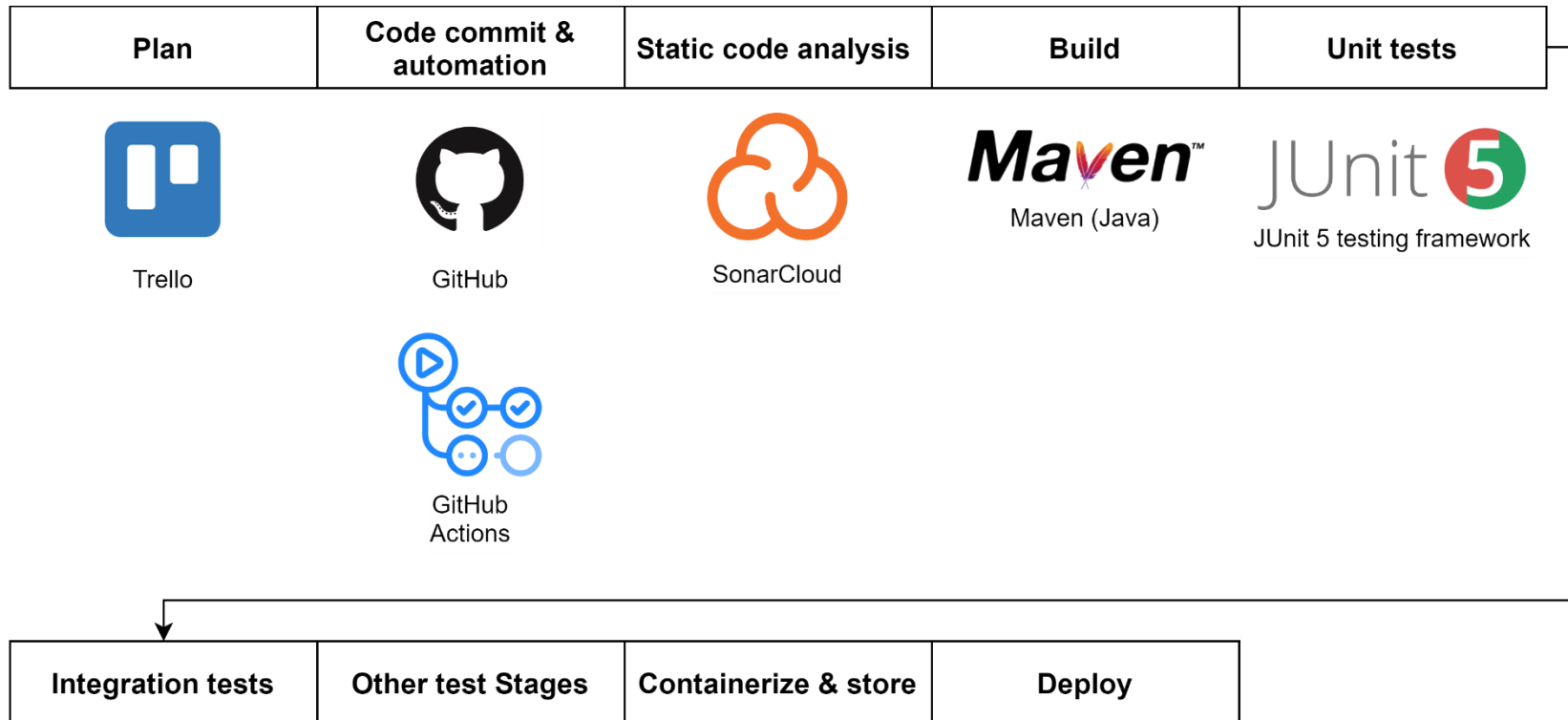
Just a binary (.jar-file), no container!

Artifacts

Produced during runtime

Name	Size	
 artifact	18.2 MB	

<https://github.com/miverboven/spring-github-actions-demo/blob/master/.github/workflows/build-and-upload.yml>




Towards Continuous Deployment

DevOps: Infrastructure as Code – Kubernetes (k8s)

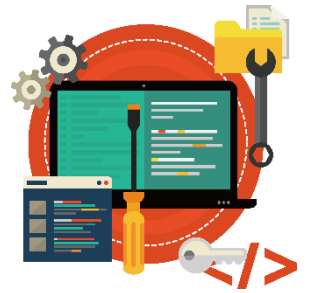


fe-service.yaml



```
apiVersion: v1
kind: Service
metadata:
  name: frontend
spec:
  ports:
    - nodePort: 32000
      port: 80
      targetPort: 8080
  selector:
    app: frontend
  type: NodePort
```

GitHub Actions: Our example – Continuous Deployment



<https://github.com/miverboven/spring-github-actions-demo/blob/master/.github/workflows/deploy.yml>

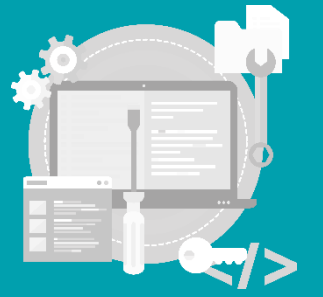
```
1  name: Deploy to Heroku
2
3  on: push
4
5  jobs:
6    deploy:
7      runs-on: ubuntu-latest
8      name: Deploy to Heroku
9      steps:
10       - uses: actions/checkout@v2
11       - uses: akhileshns/heroku-deploy@v3.12.12
12         with:
13           heroku_api_key: ${secrets.HEROKU_API_KEY}
14           heroku_email: ${secrets.HEROKU_EMAIL}
15           heroku_app_name: ${secrets.HEROKU_APP}
```

Repository secrets

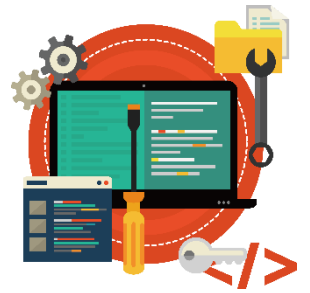
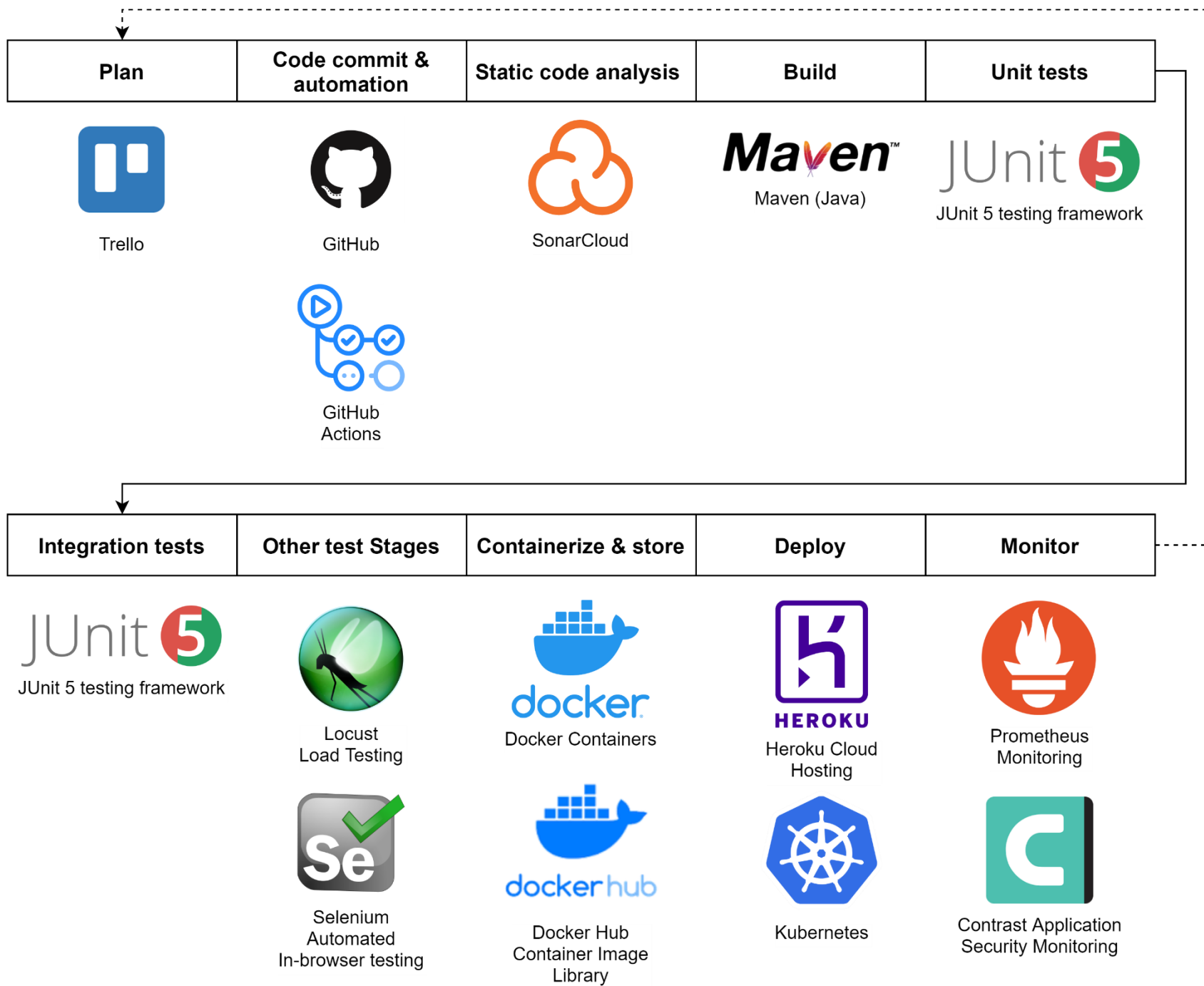
HEROKU_API_KEY

HEROKU_APP

HEROKU_EMAIL



Demo: Microservices & Deployment in Cloud k8s



Continuous Deployment



Monitor

Attacks

Attack Events

6 Active Attacks
YOU'RE BEING EXPLOITED







Live ▾

Production ▾

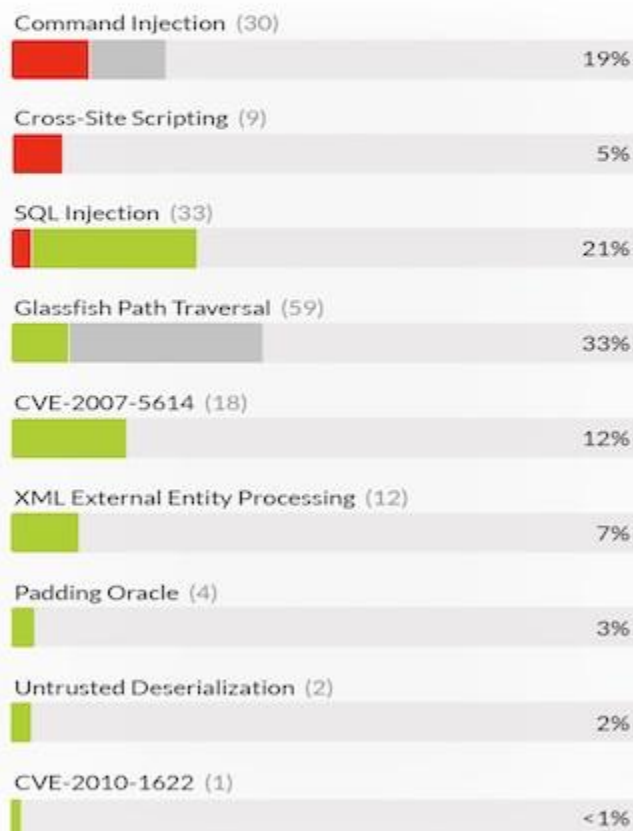


Show probed

6 Attackers

	0:0:0:0:0:0:0:1 Manual Attack	32
	192.168.1.4 Automated Attack	6
	134.147.39.127 Manual Attack	100
	194.65.20.3 (Fred Flintstone) Manual Attack	14
	134.155.23.248 Automated Attack	11
	0:0:0:9:9:0:0:1 Automated Attack	8

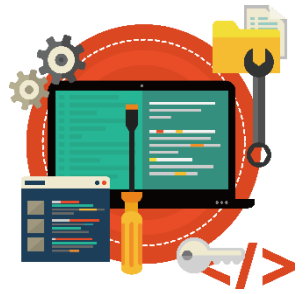
168 Attack Events



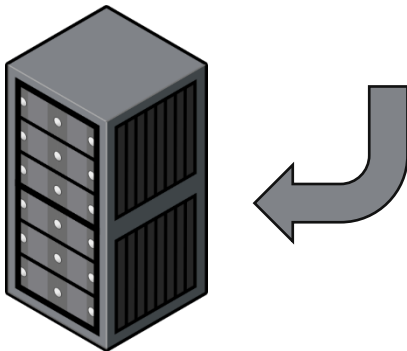
3 Target Applications



DevOps: IaC - Configuration Management



A N S I B L E

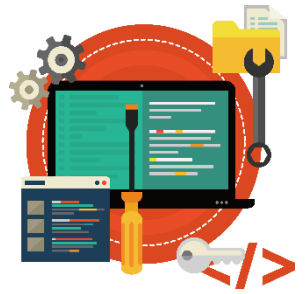


```
master-playbook.yml

- name: Adding apt repository for Kubernetes
  apt_repository:
    repo: deb https://apt.kubernetes.io/ kubernetes-xenial main
    state: present
    filename: kubernetes.list

- name: Install Kubernetes binaries
  apt:
    name: "{{ packages }}"
    state: present
    update_cache: yes
  vars:
    packages:
      - kubelet
      - kubeadm
      - kubectl
```

DevOps: IaC - Terraform Provisioning & Orchestration



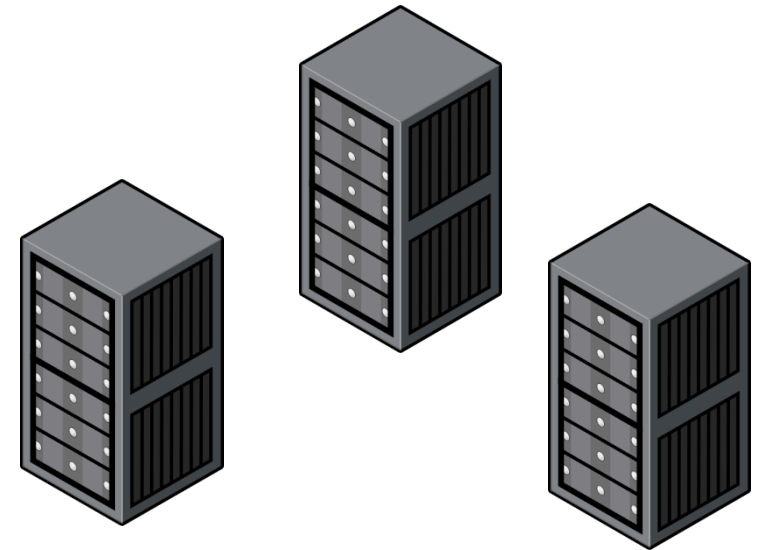
```
lb.tf

resource "azurerm_resource_group" "test" {
  name      = "LoadBalancerRG"
  location  = "West US"
}

resource "azurerm_public_ip" "test" {
  name                = "PublicIPForLB"
  location            = "West US"
  resource_group_name = "${azurerm_resource_group.test.name}"
  allocation_method   = "Static"
}

resource "azurerm_lb" "test" {
  name                = "TestLoadBalancer"
  location            = "West US"
  resource_group_name = "${azurerm_resource_group.test.name}"

  frontend_ip_configuration {
    name                = "PublicIPAddress"
    public_ip_address_id = "${azurerm_public_ip.test.id}"
  }
}
```



Cloud Native: Towards Serverless

