

# Containere și Fluxuri DEVOPS

**Conf. dr. Cristian KEVORCHIAN**

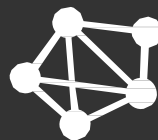
Facultatea de Matematică și Informatică

# Dezvoltatori vs ITPro



## Dezvoltatori

Crează aplicații la un preț competitiv fără ași pune problema infrastructurii IT



Noile aplicații rulează corect pe mașina pe care se realizează dezvoltarea dar cu probleme pe masinile de producție cu infrastructuri server tradiționale.



Productivitatea și inovația este suspendată când așteaptă schimbările de la nivelul infrastructurii IT.



## ITPro

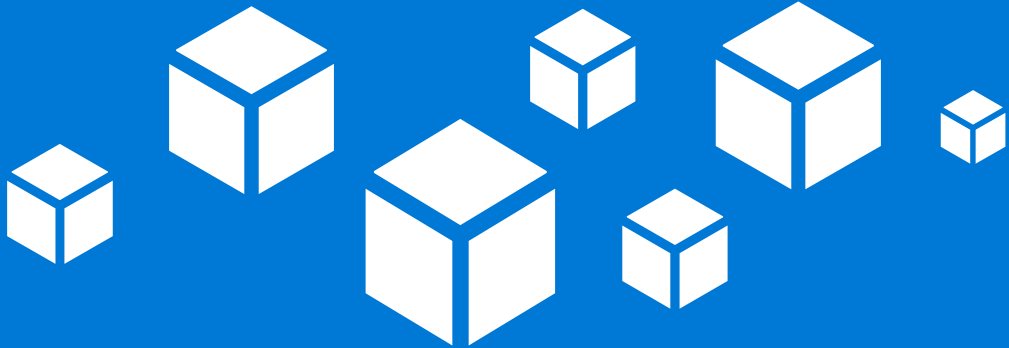
Trebuie realizat managementul serverelor și conformitatea cu [micile] tendințe disruptive.

Lipsește siguranța modului în care sunt integrate noile familii de aplicații și este necesar ajutorul dezvoltatorilor.

Este dificil de realizat consensul între managementul sistemului de servere și alinierea la cerințele aplicațiilor.

# Stabilirea consensului între inovare și controlul infrastructurii

Suficientă libertate acordată dezvoltatorilor pentru a crea aplicații inovatoare dar fără a influența capacitatea de gestionare a serverele.

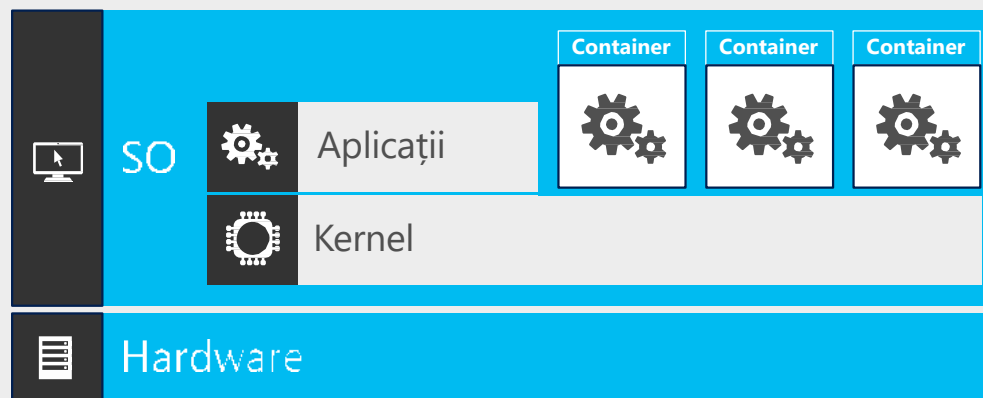


## Containere



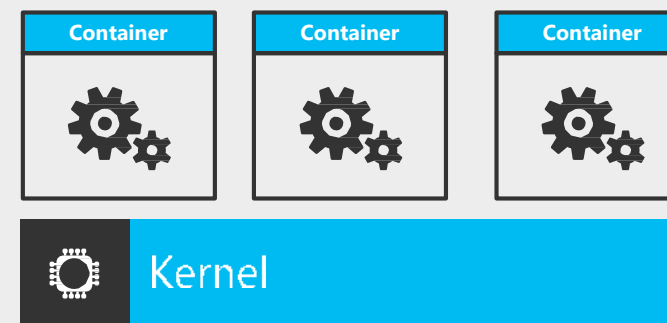
# Container, definiție

**Containere** = Virtualizarea sistemului de operare

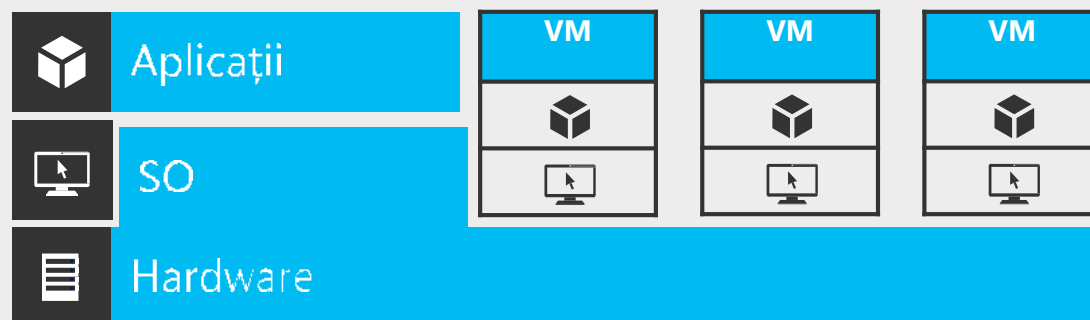


**Containere "Windows Server"**

Viteză și densitate maximă

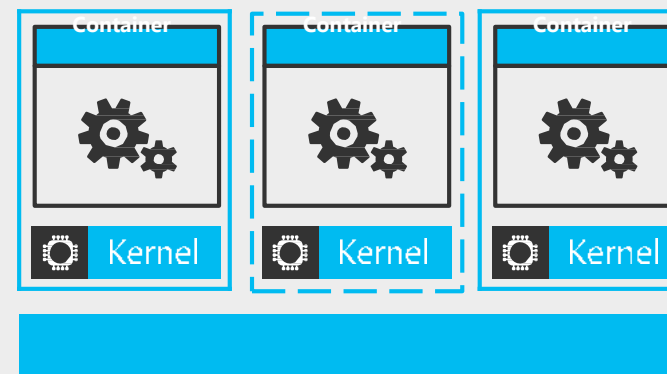


**Mașini virtuale tradiționale** = virtualizare hardware



**Containere Hyper-V**

Izolare și un plus de performanță



# Beneficiile Containerelor

O nouă abordare a dezvoltării, transferului, deployment-ului și instanțierii aplicațiilor



Fizic

Aplicațiile tradiționale sunt construite și implementate pe sisteme fizice și se poziționează față de acestea într-un raport de 1:1.

Noile aplicații necesită deseori noi sisteme fizice pentru izolarea resurselor.



Fizic/  
Virtual

Împachetare și rulare de aplicații în cadrul containerelor.



Virtual

Cote mai mari de consolidare și o utilizare mai eficientă a infrastructurii.

Dezvoltarea mai rapidă a aplicațiilor decât într-un mediu fizic tradițional.

Aplicațiile sunt implementate în VM-uri cu un grad înalt de compatibilitate.

Aplicațiile beneficiază de funcții VM cheie, cum ar fi migrarea live, HA.

## Beneficii

**Facilitează accelerarea implementării aplicațiilor.**

**Reduce efortul pentru implementarea aplicațiilor.**

**Optimizarea dezvoltării și testării de aplicații.**

**Costuri mai mici pentru implementarea aplicațiilor.**

**Creșterea nivelului de consolidare a serverelor.**

# Containere-generalități

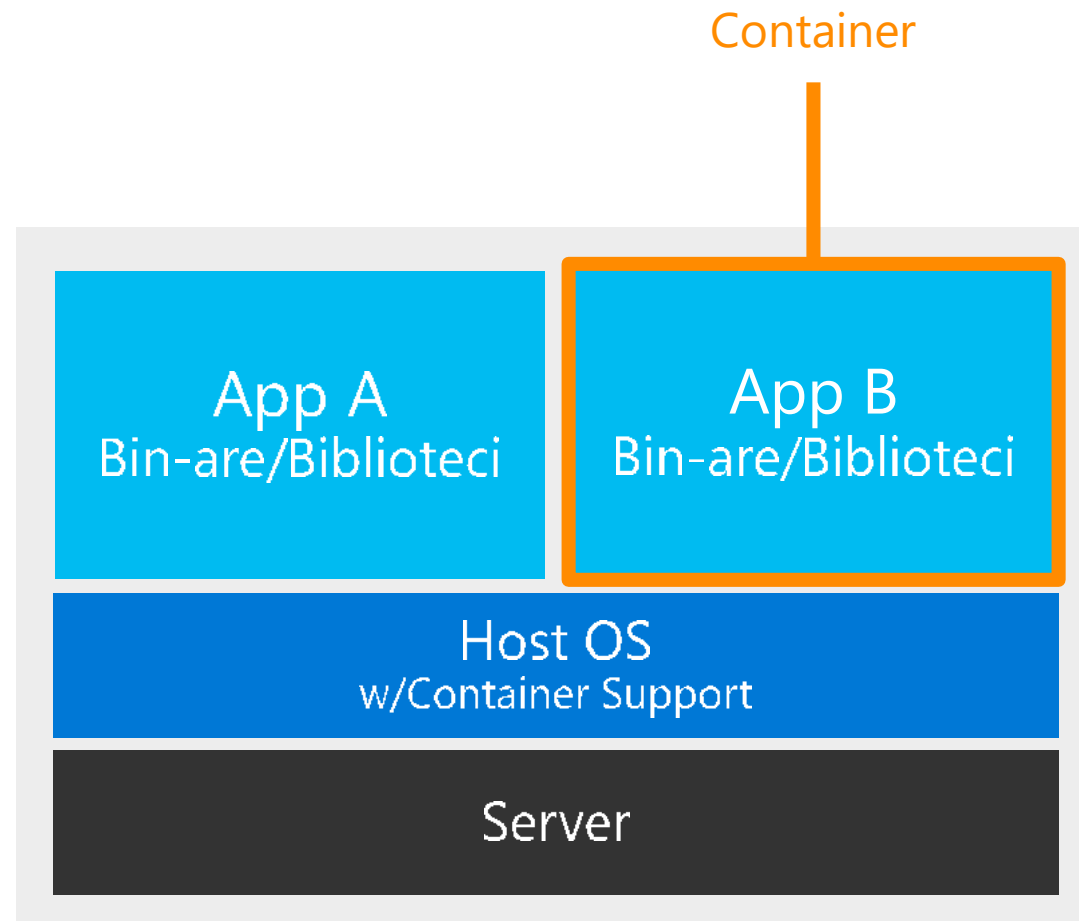
**Dependențe:** Fiecare aplicație are propriile dependențe care includ atât elemente software (servicii, biblioteci) cât și hardware (CPU, memorie, storage).

**Virtualizarea:** Motorul containerelor este un sistem de management și orchestrare a containerelor, echivalent cu virtualizarea prin care se pot izola dependențele pentru fiecare aplicație prin "ambalarea" lor în containere virtuale.

**Partajarea SO gazda:** Procesele în containere sunt izolate de celelalte containere în spațiul utilizatorilor, dar partajează kernel-ul cu gazda și cu celelalte containere.

**Flexibilitate:** Diferențele dintre sistemul de operare și infrastructura de bază se abstrăgionează, simplificând abordarea, "deployment oriunde"..

**Rapid:** Containerele pot fi create aproape instant permițând o scalare elastică asociată cererii.



# Containerele oferă funcționalități unice

O iterare  
rapida

Stari  
separate

Controlul  
resurselor

Imutabilitate

Deployment  
rapid

# Containere

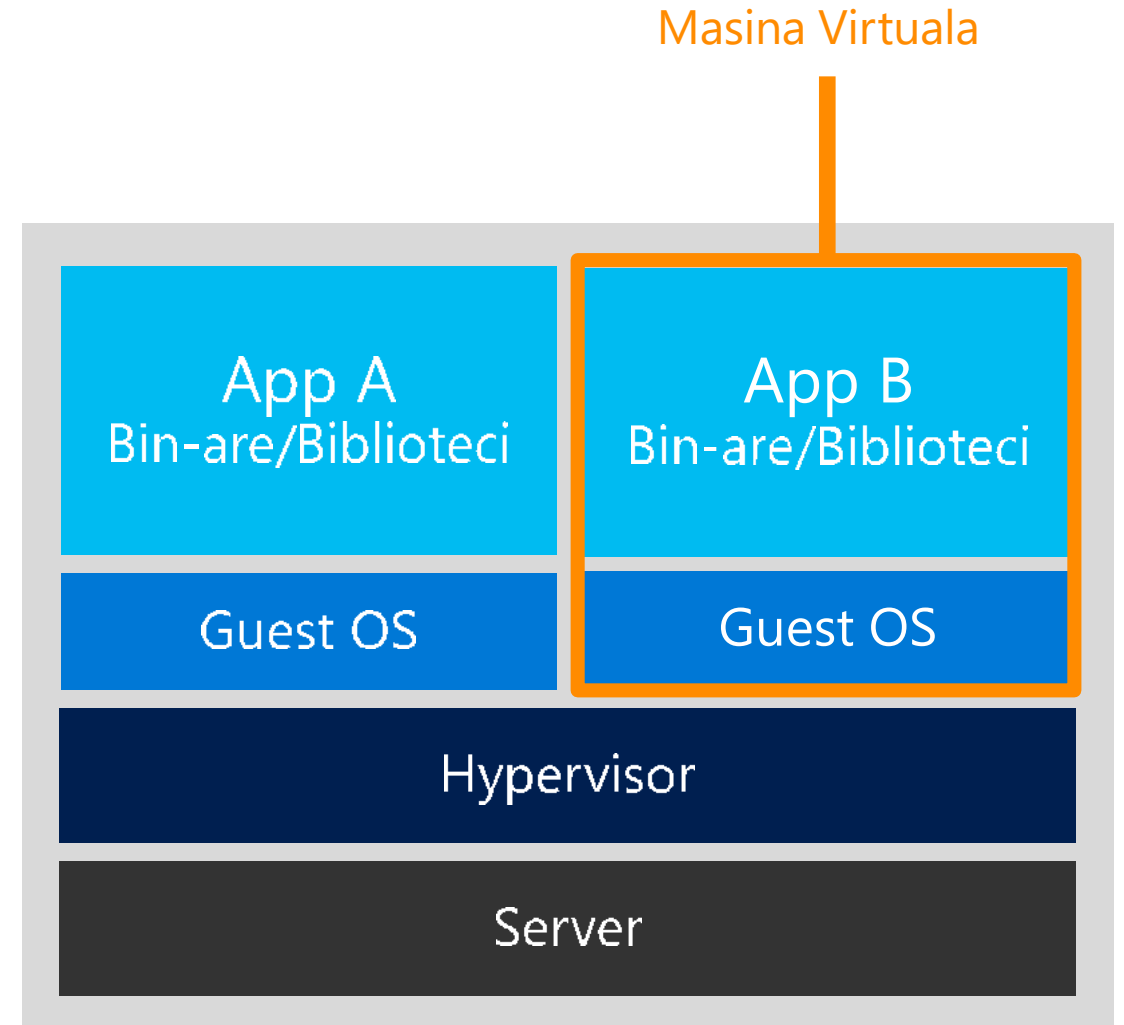
## Diferențe față de masini virtuale

**Dependențele:** Fiecare app virtualizată include aplicația în sine, cerând binare și biblioteci și un SO "oaspete", care poate integra mai multi GB de date.

**SO independent:** Fiecare MV poate avea un sistem de operare diferit de al altor MV, împreună cu un sistem de operare diferit de SO gazdă în sine.

**Flexibilitate:** MV pot fi migrate în alte locații pentru a balansa resursele utilizate și pentru mentenanta locației fara "downtime".

**Securitate:** Nivel înalt de izolare securizată a resurselor pentru of resource and security isolation for sarcinile cheie virtualizate.





# Containere incluse în MV

## Scenari multiple de implementare a aplicațiilor

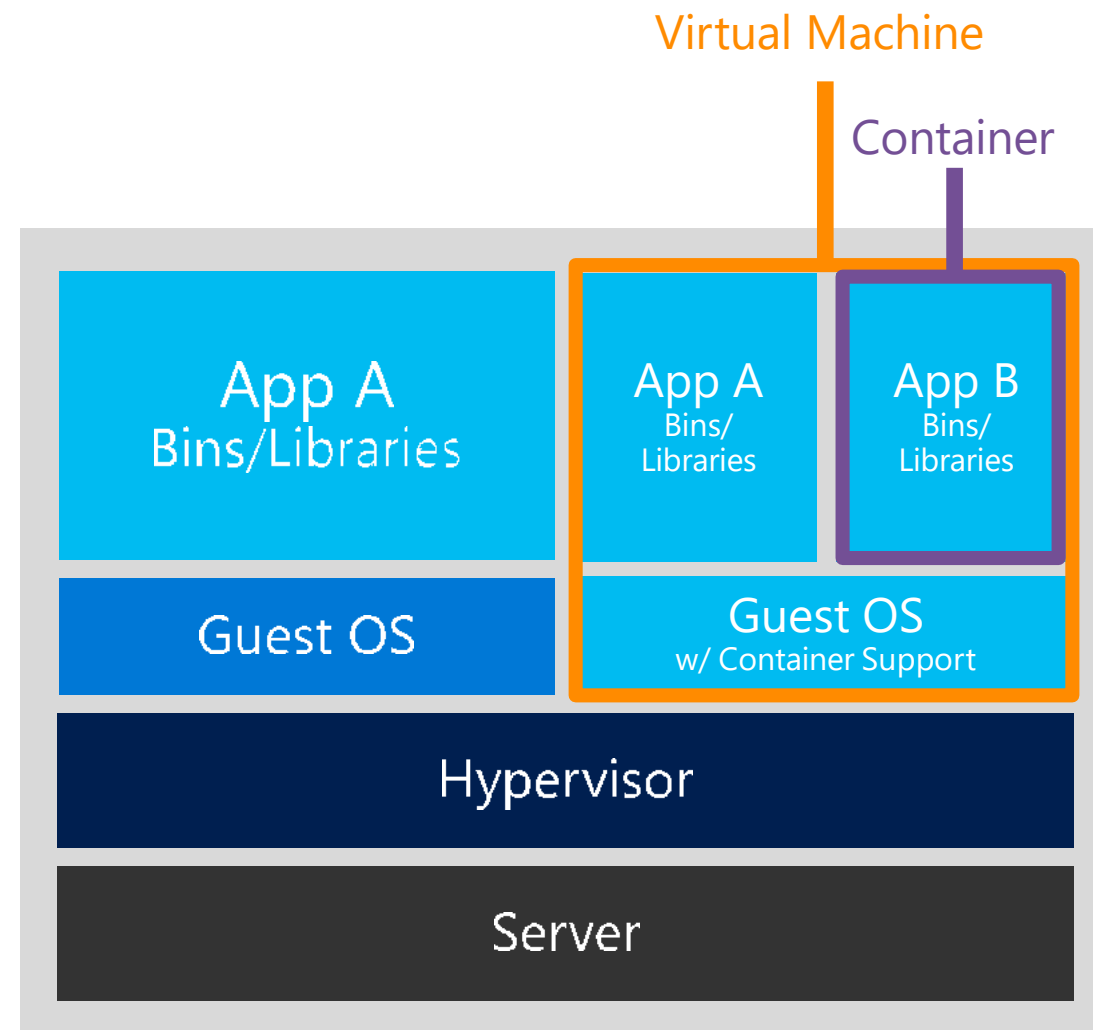
**Containere în MV:** Prin combinarea containerelor cu MV, utilizatorii pot implementa mai multe sisteme de operare MV diferite, iar în interior pot implementa mai multe containere în cadrul acelor SO.

Prin combinarea containerelor cu MV, ar fi necesare mai puține MV pentru a suporta un număr mai mare de aplicații.

Mai puține MV ar avea ca rezultat o reducere a necesarului de stocare.

Fiecare MV ar susține mai multe aplicații izolate, crescând astfel densitatea globală a soluției.

**Flexibilitate:** Rularea containerelor în interiorul MV permite implementarea de soluții precum "live migration" pentru utilizarea optimă a resurselor și întreținerea gazdei.



# Containere Windows Server

## Structura și funcționalitățile

**Build:** Dezvoltatorii vor folosi instrumente cunoscute de dezvoltare, cum ar fi Visual Studio, Eclipse pentru a scrie aplicații și a rula în containere.

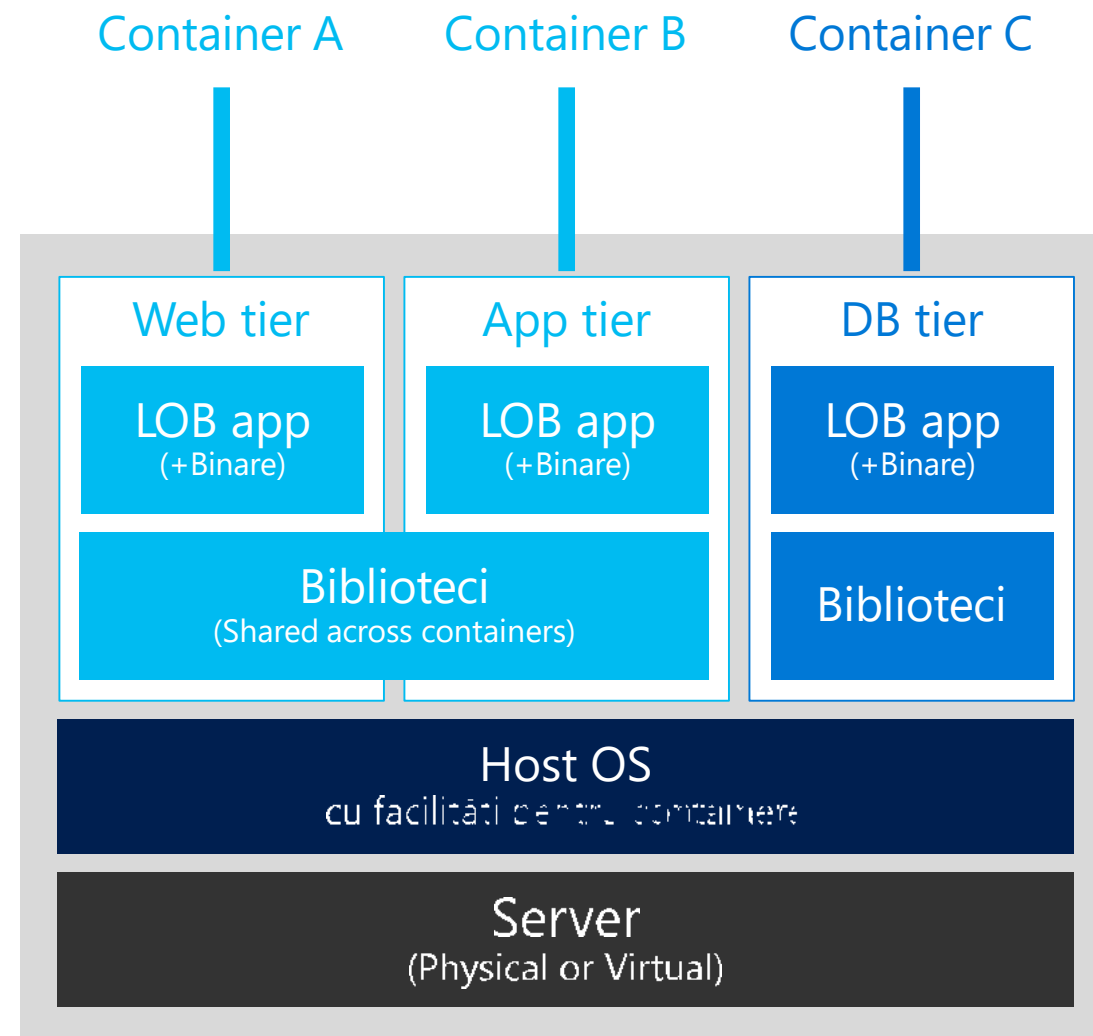
Prin construirea de aplicații modulare care folosesc containerele, modulele pot să fie actualizate independent – cu planuri independente.

**Run:** Funcționalitățile containerelor sunt built-in în Windows Server 2016.

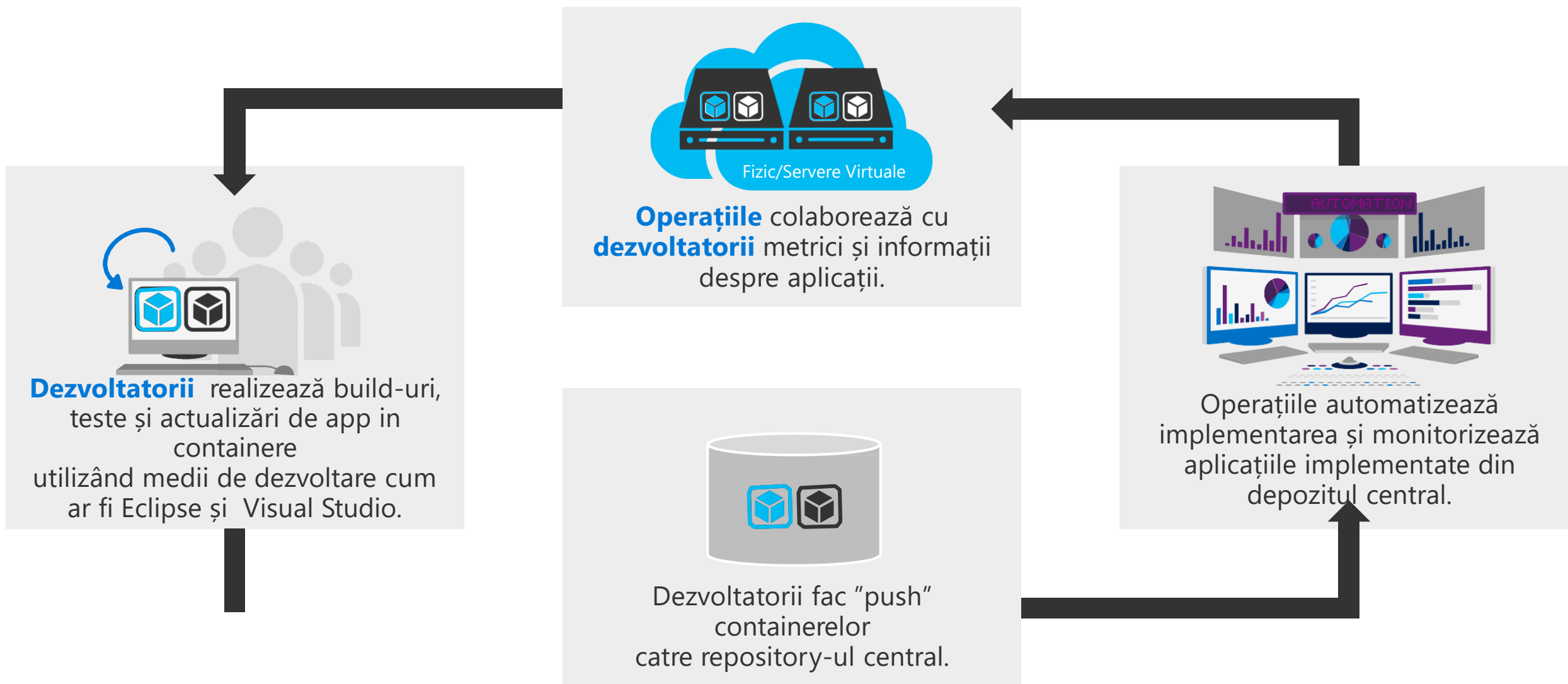
**Management-ul:** Deployment-ul și managementul containerelor se realizează cu PowerShell sau Docker.

**Resursele:** Consumul de CPU și resursele de memorie pe container în funcție de capacitatea de stocare și banda transzitată.

**Network:** Furnizează NAT sau DHCP/static IP pentru conectivitate în rețea



# Containerele sunt centrate pe procese DevOps



Rezumând, putem afirma că:

Prin containerizarea aplicațiilor vechi utilizând varianta de containerizare Windows Server, obținem o consistență și un management îmbunătățit al echipelor de dezvoltatori și tester, pe de-o parte și cele de implementare, pe de altă parte – într-un mediu unitar DevOps - fără a modifica aplicația..

# Containere Hyper-V

## Structură și funcționalități cheie

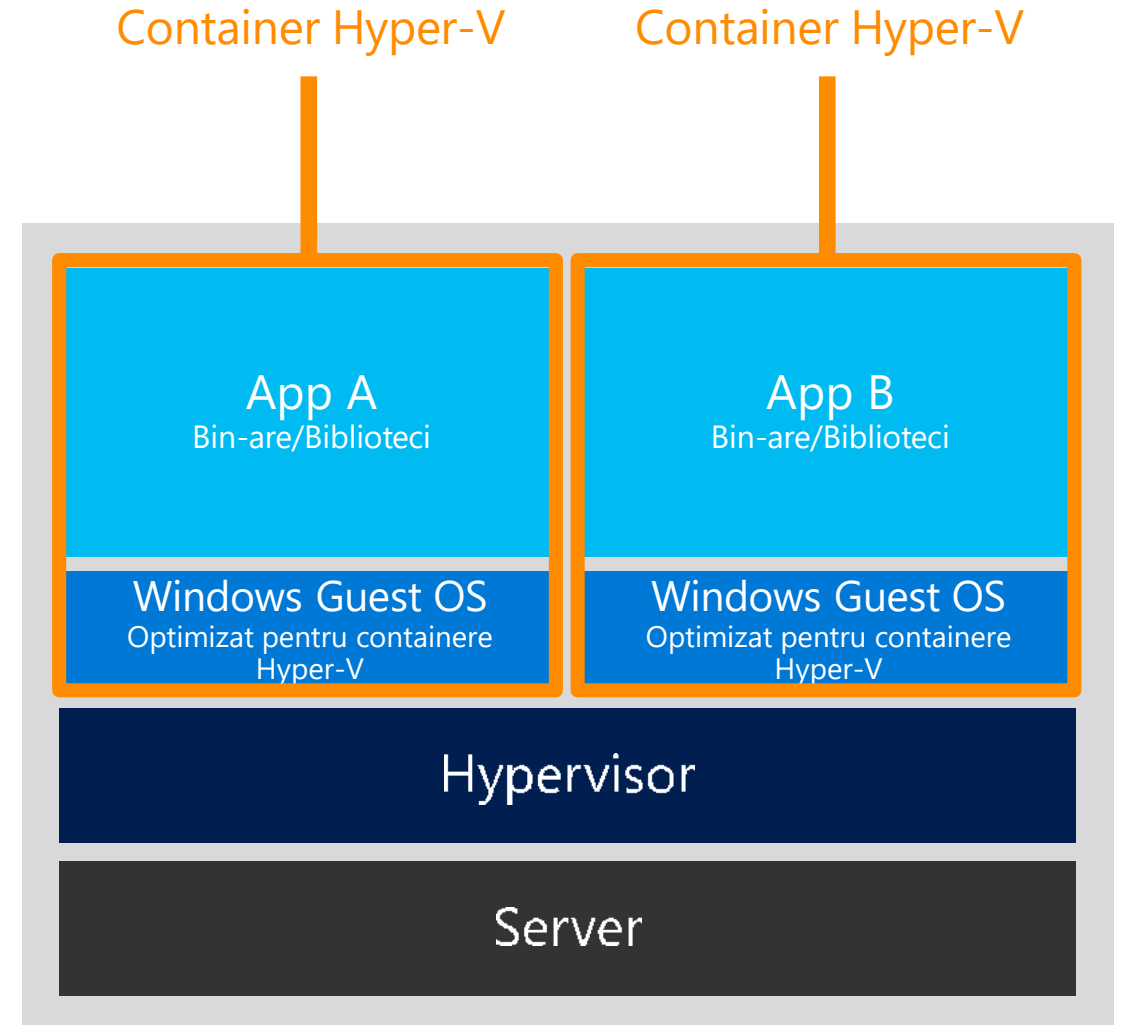
**Consistența:** Containerele Hyper-V utilizează aceleași API-uri ca și containerele Windows Server, asigurând coerența între seturile de instrumente de gestionare și implementare.

**Compatibility:** Containerele Hyper-V utilizează aceleași imagini ca și containerele Windows Server.

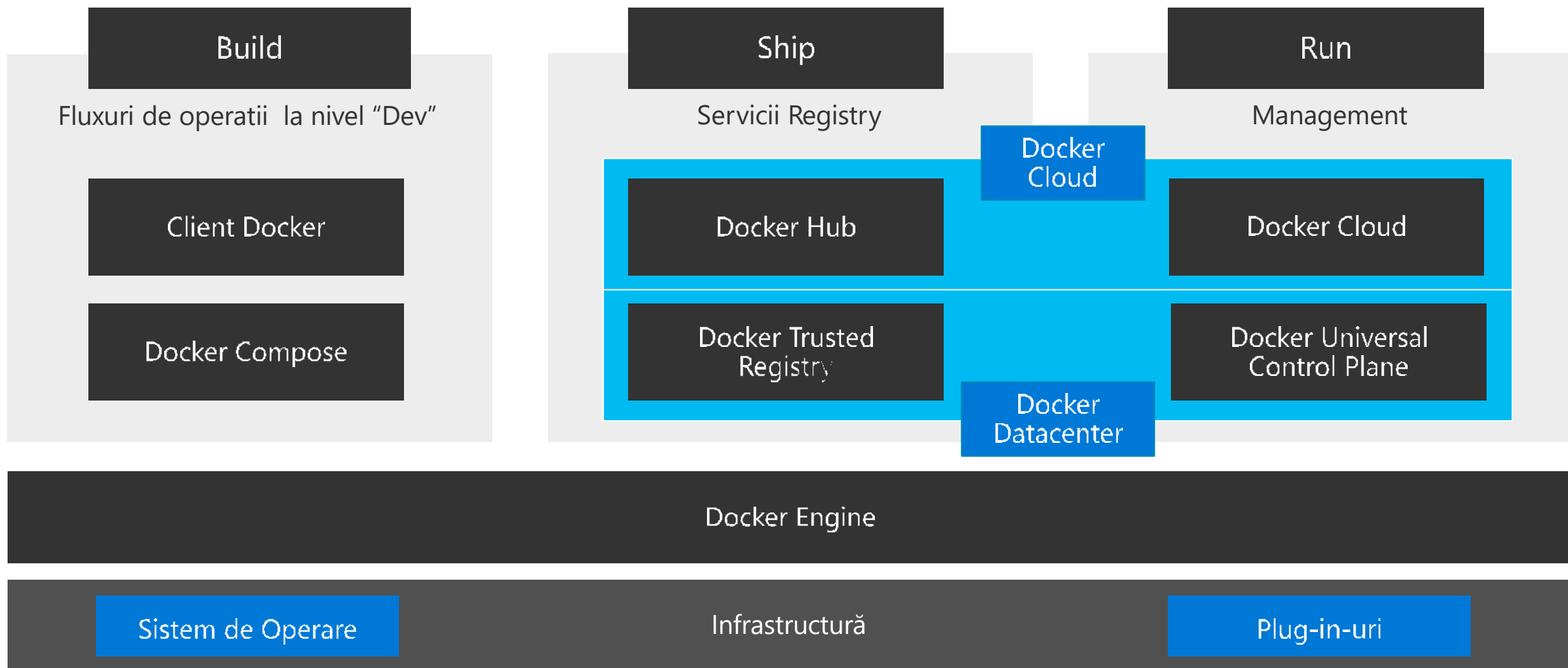
**Izolare Puternică:** Fiecare container Hyper-V are propria copie dedicată a kernel-ului.

**Foarte sigur:** Construit cu o tehnologie de virtualizare Hyper-V dovedită a fi foarte sigura.

**Optimizată:** Layer-ul de virtualizare și sistemul de operare au fost special optimizate pentru lucrul cu containere



# Componente Docker

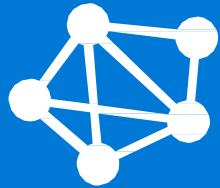


# Containerele reprezinta un excelent mediu pt.:

Calcul  
Distribuit

$f()$

Scalare



Baze de  
date



Task-uri

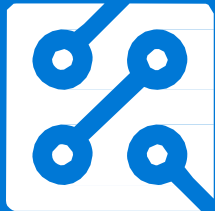


Web App



# SO pentru lucrul cu containere

## Nano Server

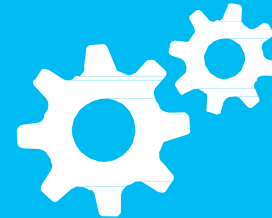


Bine  
optimizat



Aplicatii  
"Born-in-the-cloud"

## Server Core



Larg  
Compatibil



Aplicatii  
tradiționale



Windows Server



# Instrumente la dispoziție

## Managementul containerelor



PowerShell



Docker



Others

## Medii de dezvoltare

 Visual Studio

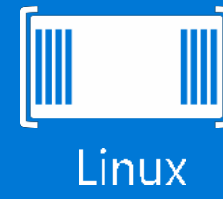
 eclipse

Altele...

## Tehnologii bazate pe containere



Windows Server



Linux

## Microsoft cloud



Azure



On Prem



Service Provider

# Instrumente pentru Dev

## Limbaje și Framework-uri pentru dezvoltare

PHP	Go	Python	.Net	Node	Perl
Ruby	JavaScript	Win32	Java	C++	

## Microsoft Cloud



Azure



On Prem



Service Provider

## Masini Virtuale



Windows Server



Linux

## Service fabric



## Tehnologii Bazate pe Containerere



Windows Server



Linux

# Opțiuni de studiu

Aprofundare Windows Server 2019:  
[www.microsoft.com/WindowsServer2019](http://www.microsoft.com/WindowsServer2019)

Documentatia pentru Containere Windows:  
<http://aka.ms/containers>

Blog-ul Windows Server:  
<http://blogs.technet.microsoft.com/windowsserver>

Resurse Docker  
<https://www.docker.com/resources>

