



Clusterul “Service Fabric”

CRISTIAN KEVORCHIAN

Modele pentru servicii cloud

- Un **Serviciu Cloud** este orice serviciu IT disponibilizat, la cerere, către un utilizator peste Internet, de către un furnizor de cloud computing.
- Serviciile Microsoft Azure sunt structurate pe patru nivele:
 - **Nivelul client** – nivelul consumatorului de servicii cloud(on-prem)
 - **Nivelul de integrare** – mediator dintre primul nivel și serviciile cloud. Exemplu AD-autentificare și acces, Managerul de trafic-load balancing între app, dar și CDN(Content Delivery Network).
 - **Nivelul aplicație**-middleware-ul aplicațiilor instalate și executate.
 - **Nivelul de date** – nivelul dedicate serviciilor de stocare date structurate și nestructurate

Service Fabric

- Un cluster Service Fabric este o familie de mașini virtuale sau fizice conectate într-o rețea în care sunt implementate și gestionate sisteme de microservicii.
- O mașină sau VM care face parte dintr-un cluster se numește nod.

Distributia consensului

Distributia consensului asigură echilibrul între nodurile unui sistem distribuit în sensul de a ajunge la un "acord" asupra unui mod de folosire a resurselor sistemului.

Acest subiect este familiar oricăror tehnicieni care lucrează cu sisteme distribuite, cum ar fi HDFS, MQ, ZooKeeper, Kafka, Redis și Elasticsearch.

Odată cu creșterea rapidă a complexității a rețelelor distribuite, dezvoltatorii au explorat întotdeauna soluții posibile pentru a rezolva această problemă extrem de importantă atât în teorie, cât și în practică.

Distribuirea Consensului și Service Fabric

- Una dintre cele mai grele părți în scriere unui serviciu statefull fiabil este **replicarea datelor**. Service Fabric permite replicarea operație / eveniment printr-o implementare a unui algoritm pentru distribuirea consensului:
 - Replicile primare primesc comenzi și replică operațiile către secundari.
 - Dacă primarul eșuează, un secundar actualizat va fi ales ca noul primar.

Dacă doriți să permiteți citirea de la secundare, aceasta se constituie ca opțiune.

SF suportă atât servicii stateless în memorie (volatile), cât și persistente, statefull.

Concepte cheie

- **Managerul de resurse al clusterului(MRC)**
“**Fabric cluster**” furnizează mai multe mecanisme principale pentru descrierea clusterului. În timpul funcționării MRC-ul pentru asigurarea disponibilității serviciilor din cluster.
- **Concepte cheie:**
 - Domenii de Eșec(Fault Domains)
 - Domenii de Actualizare(Upgrade Domain)
 - Proprietăți ale nodurilor(Node Properties)
 - Capacități ale nodurilor(Node Capacities)

Domenii de Eșec(Fault Domains[FD])

- Un FD este orice parte a unui domeniu de eșec definit. Orice mașină este un FD, din moment ce poate eșua din diverse motive de la alimentarea cu energie electrică la un NIC cu probleme.
- Mașinile conectate la același switch au același FD
- FD-urile sunt inerent ierarhice și sunt reprezentate ca URI(Uniform Resource Identifier) în Service Fabric.
- Este important ca FD-urile să fie configurate corect, deoarece "Service Fabric" utilizează aceste informații pentru a plasa în siguranță serviciile.
- În Azure, SF folosește informațiile din FD furnizate de mediul de lucru pentru a configura corect nodurile din cluster în locul utilizatorului.

Exemplu:

- Este important ca informațiile despre FD-uri furnizate serviciului Fabric să fie corecte.
- De exemplu, să presupunem că nodurile clusterului SF rulează peste 10 mașini virtuale, care la rândul lor rulează pe cinci mașini gazdă fizice.
- În acest caz, chiar dacă există 10 mașini virtuale, există doar 5 FD-uri diferite (de nivel superior). Partajarea aceleiași gazde fizice determină ca VM-urile să partajeze același FD al rădăcinilor, VM-urilor eșuează coordonat în cazul în care gazda lor fizică eșuează.

În graficul alăturat sunt colorate toate entitățile care contribuie la FD-uri și enumerăm toate FD-urile rezultate.

În exemplu, avem centre de date ("DC"), rack-uri ("R") și blade-uri ("B"). Este posibil să presupunem că, dacă fiecare blade are mai multe mașini virtuale, dar ar putea exista și un alt layer în ierarhia FD-ului.

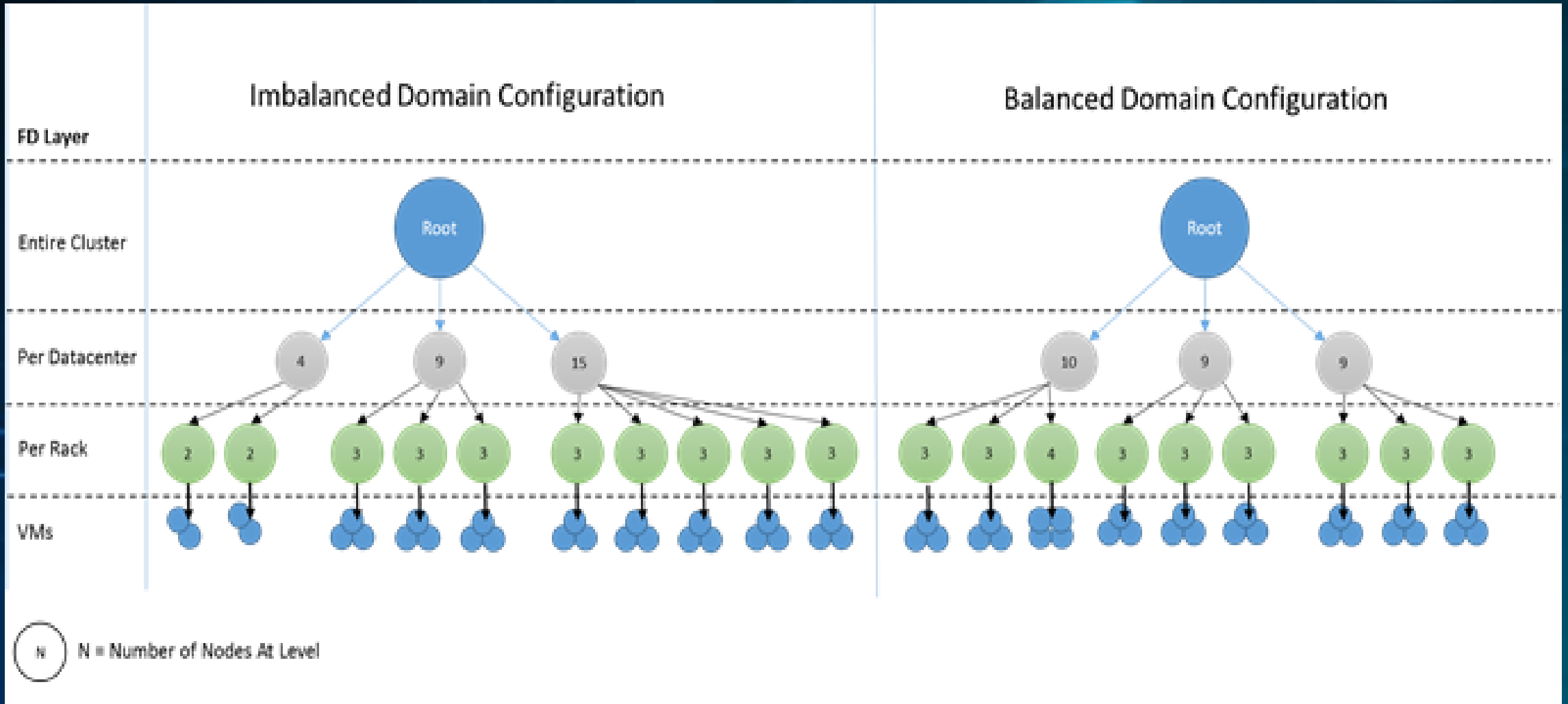


- În timpul execuției, Managerul de Resurse al clusterului Service Fabric identifică FD-urile din cluster și planifică schemele de lucru.
- Replicile stateful sau instanțele stateless pentru un anumit serviciu sunt distribuite astfel încât acestea să fie FD-uri separate.
- Distribuirea serviciului în toate FD-urile asigură faptul că disponibilitatea serviciului nu este compromisă atunci când un FD eșuează la orice nivel al ierarhiei.
- **Managerul de Resurse** al clusterului Service Fabric nu ține seamă de numărul nivelelor existente în ierarhia FD-ului, dar încearcă să se asigure că pierderea oricărui segment al ierarhiei nu afectează serviciile care rulează în cadrul acesteia.

Cluster Echilibrat vs. Neechilibrat

- Este de preferat existența aceluiași număr de noduri la fiecare nivel de adâncime din ierarhia FD-ului.
- Dacă arborele FD-ului nu este echilibrat în clusterul asociat, misiunea managerului de resurse asociat clusterului este dificilă la alocarea serviciilor.
- O abordare neechilibrată a FD-urilor implică pierderea anumitor domenii și influențează disponibilizarea serviciilor.
- Managerul Resurselor asociat cluster-ului trebuie să-și optimizeze modul de alocare în condițiile în care:
 - dorește să folosească mașinile din acel domeniu "greu" prin plasarea serviciilor pe acestea
 - dorește să plaseze servicii în alte domenii, astfel încât pierderea unui domeniu să nu genereze probleme.

Exemplu de dispunere de clustere. Nodurile sunt distribuite uniform între FD-uri
vs. FD cu mai multe noduri decât celelalte domenii de defecțiuni.




EX. AZURE


- **Alegerea FD-ului care conține un nod este gestionată de Azure.**
- Cu toate acestea, în funcție de numărul de noduri pentru care optăm, putem ajunge la FD-uri cu mai multe noduri decât altele.
- De exemplu, presupunem că avem cinci FD-uri în cluster, dar furnizați șapte noduri pentru un **NodeType** dat. În acest caz, primele două FD-uri au mai multe noduri.
- Dacă se continuă scalarea pe mai multe NodeType-uri cu doar câteva instanțe, situația se înrăutățește considerabil. Din acest motiv, se recomandă ca numărul de noduri din fiecare tip să fie un număr multiplu al numărului de FD-uri.

Cluster cu două NodeType-uri(FrontEnd și BackEnd)

Microsoft Azure

 Service Fabric Explorer

☒ OK ☒ Warning ☒ Error



Cluster

> Applications

Nodes

> _BackEnd_1

> _BackEnd_0

> _FrontEnd_3

> _FrontEnd_4

> _FrontEnd_1

> _BackEnd_2

> _BackEnd_4

> _BackEnd_3

> _FrontEnd_2

> _FrontEnd_0

Nodes

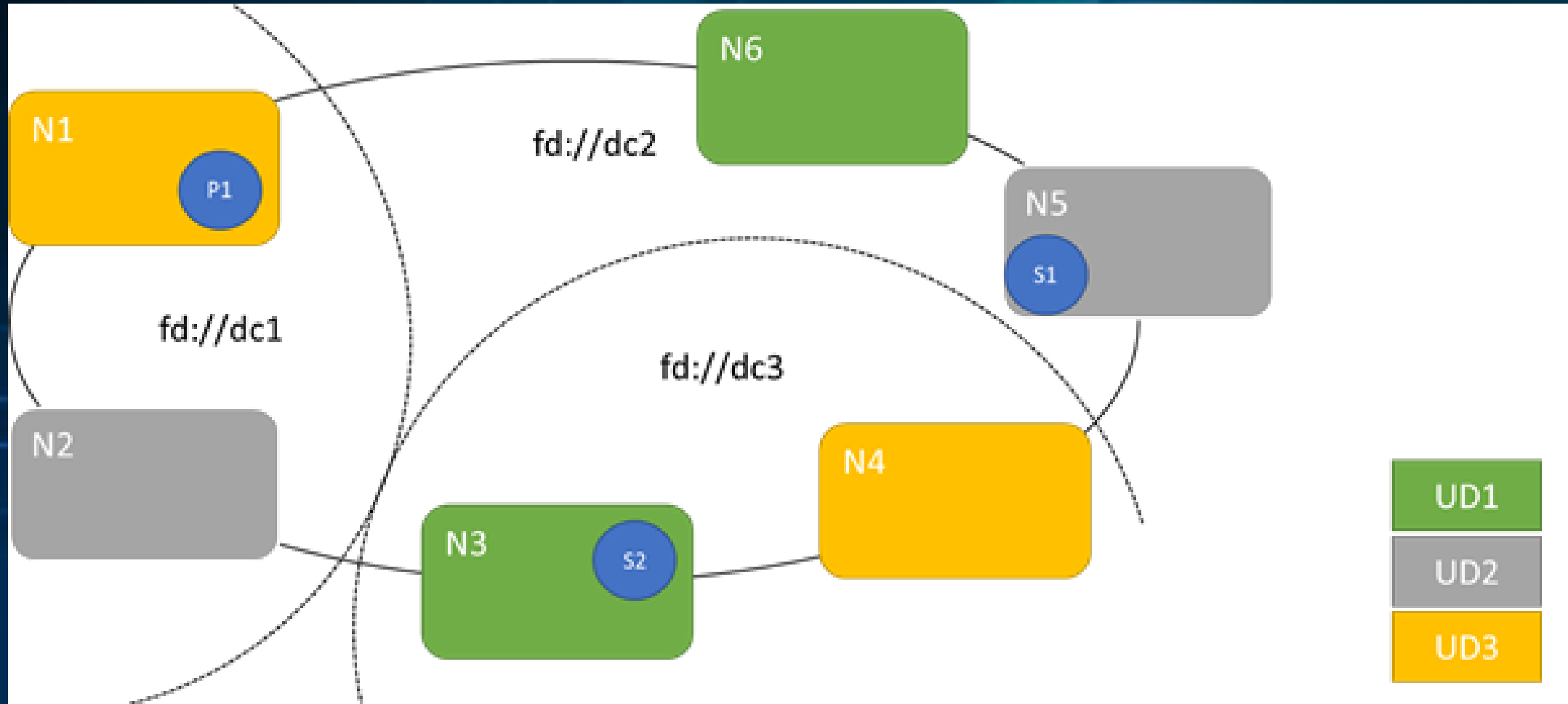
☒ OK ☒ Warning ☒ Error

Name	Address	Node Type	Upgrade Domain
_BackEnd_0	10.0.0.9	BackEnd	1
_BackEnd_1	10.0.0.10	BackEnd	4
_BackEnd_2	10.0.0.11	BackEnd	2
_BackEnd_3	10.0.0.12	BackEnd	0
_BackEnd_4	10.0.0.13	BackEnd	3
_FrontEnd_0	10.0.0.4	FrontEnd	1
_FrontEnd_1	10.0.0.5	FrontEnd	0
_FrontEnd_2	10.0.0.6	FrontEnd	4
_FrontEnd_3	10.0.0.7	FrontEnd	3
_FrontEnd_4	10.0.0.8	FrontEnd	2

Domenii de actualizare(Upgrade Domains[UD])

- UD-ul este o altă formă prin care Managerul de Resurse al clusterului SF să gestioneze structural clusterul.
- UD definește familii de noduri care sunt actualizate simultan. UD-urile ajută managerul de resurse al cluster-ului să orchestreze operații-de exemplu actualizările.
- UD este asemănător FD-ului dar aliniază și o serie de diferențe importante:
 - În primul rând, zonele de gestionare a eșecului hardware definesc FD-ul.
 - Actualizarea domeniilor, este definită prin politică.
 - Puteți decide asupra numărului de UD-uri cu care lucrați, ca alternativă la alocarea acestora de mediul de lucru.
 - Se poate opta asupra unui număr mare de UD-uri pe măsură ce crește numărul de noduri.
 - O alta diferență între FD și UD este că UD-urile nu sunt ierarhice, dar ele acționează mai mult ca simple etichete.

Exemplu: avem trei UD-uri peste trei FD-uri. De asemenea, plasăm trei replici diferite ale unui serviciu statefull, în care fiecare se termină în diferite FD-uri și UD-uri. Această distribuție permite pierderea unui FD în timpul actualizării serviciului cu păstrarea unei copii a codului și a datelor.



Optimizarea numărului de UD-uri

Maparea FD și UD, 1:1.

Un UD/Nod(fizic sau o instanța de OS virtual)

O matrice cu linii (FD) și coloane(UD)

UD Per Node

	FD1	FD2	FD3
UD1	Node1		
UD2		Node2	
UD3			Node3
UD4	Node4		
UD5		Node5	
UD6			Node6
UD7	Node7		
UD8		Node8	
UD9			Node9

FD/UD Matrix (Sparse/Diagonal)

	FD1	FD2	FD3	FD4	FD5
UD1	Node1				
UD2		Node2			
UD3			Node3		
UD4				Node4	
UD5					Node5

1:1 FD & UD

FD1/UD1	FD2/UD2	FD3/UD3
Node1	Node2	Node3
Node4	Node5	Node6
Node7	Node8	Node9

FD/UD Matrix

	FD1	FD2	FD3
UD1	Node1	Node2	Node3
UD2	Node4	Node5	Node6
UD3	Node7	Node8	Node9

FD și UD-constrângeri și comportamentul rezultat

- Managerul de Resurse al clusterului tratează echilibrul dintre FD și UD ca o constrângere.
- Pentru o anumită partiție de servicii nu ar trebui să existe niciodată o diferență mai mare de unu din numărul de obiecte asociate serviciului (instanțe de servicii stateless sau replici ale unui serviciu statefull) între două domenii.
- Acest lucru împiedică anumite proceduri care conduc la încălcarea acestei restricții.

Exemplu. Să presupunem că avem un cluster cu șase noduri, configurat cu cinci FD-uri și cinci UD-uri.

	FD0	FD1	FD2	FD3	FD4
UD0	N1				
UD1	N6	N2			
UD2			N3		
UD3				N4	
UD4					N5

Creare servicii stateless si statefull

- Acum, să spunem că vom crea un serviciu cu un **TargetReplicaSetSize** (sau, pentru un serviciu stateless, un **InstanceCount**) de cinci replici.
- Replicile au ajuns pe N1-N5. De fapt, N6 nu este niciodată folosită indiferent de numărul de servicii create.
- Iată aspectul pe care l-am primit și numărul total de replici pentru fiecare domeniu de defecțiuni și upgrade:

Modul de alocare și numărul total de replici pentru fiecare FD și UD:

	FD0	FD1	FD2	FD3	FD4	UDTotal
UD0	R1					1
UD1		R2				1
UD2			R3			1
UD3				R4		1
UD4					R5	1
FDTotal	1	1	1	1	1	-

Distribuția este echilibrată în termeni de noduri pe FD și UD. De asemenea, este echilibrată în ceea ce privește numărul de replici pentru fiecare FD și UD. Fiecare domeniu are același număr de noduri și același număr de replici

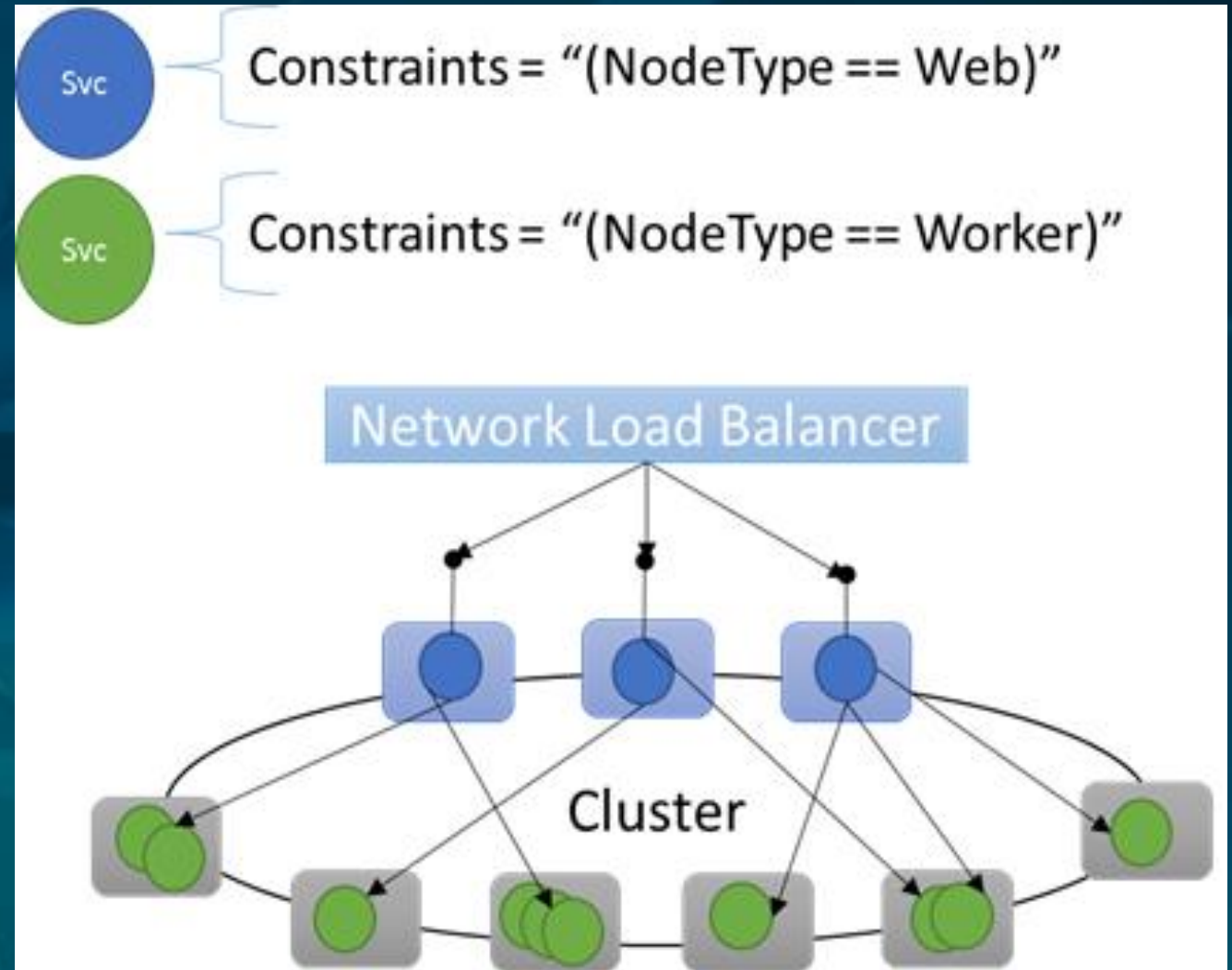
Sa analizam daca se foloseste N6 în locul lui N2

	FD0	FD1	FD2	FD3	FD4	UDTotal
UD0	R1					1
UD1	R5					1
UD2			R2			1
UD3				R3		1
UD4					R4	1
FDTotal	2	0	1	1	1	-

Această distribuție contravine restricției asupra FD. FD0 are două replici, în timp ce FD1 are zero, făcând diferența între FD0 și FD1 în total două. Managerul de resurse nu validează această distribuție de încărcare.

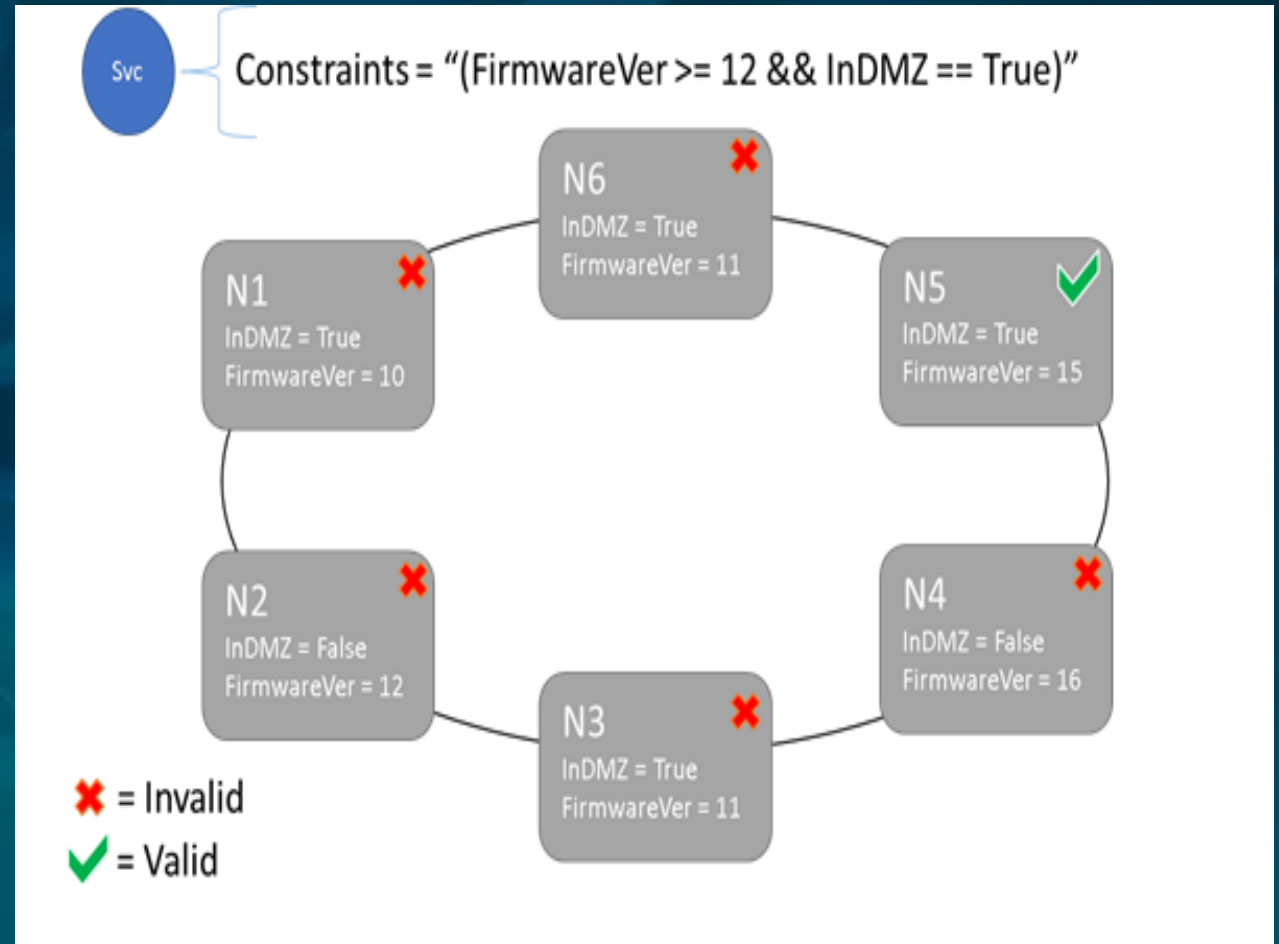
Noduri

Pentru a susține aceste tipuri de configurații, Service Fabric utilizează clase de etichete asociate nodurilor. Aceste etichete sunt numite **proprietăți ale nodurilor**. **Restricțiile de plasare** sunt instrucțiunile atașate la serviciile individuale care se selectează pentru una sau mai multe proprietăți ale nodurilor. Restricțiile de plasare definesc locul unde ar trebui să fie difuzate serviciile. Setul de constrângeri este extensibil - orice pereche cheie / valoare poate funcționa.



Proprietăți construite peste noduri

Service Fabric definește unele proprietăți implicite ale nodurilor care pot fi utilizate automat fără ca utilizatorul să le definească. Proprietățile implicite definite la fiecare nod sunt **NodeType** și **NodeName**. De exemplu, ai putea scrie o constrângere de plasare ca "(NodeType == NodeType03)". În general, am găsit că NodeType este una dintre cele mai frecvent utilizate proprietăți. Este utilă, pt.că se mapează 1:1 cu un tip de mașină. Fiecare tip de mașină corespunde unui tip de încărcare într-o aplicație tradițională n-tier.





Muțumesc!