# Windows Azure Hoạt Động Như Thế Nào

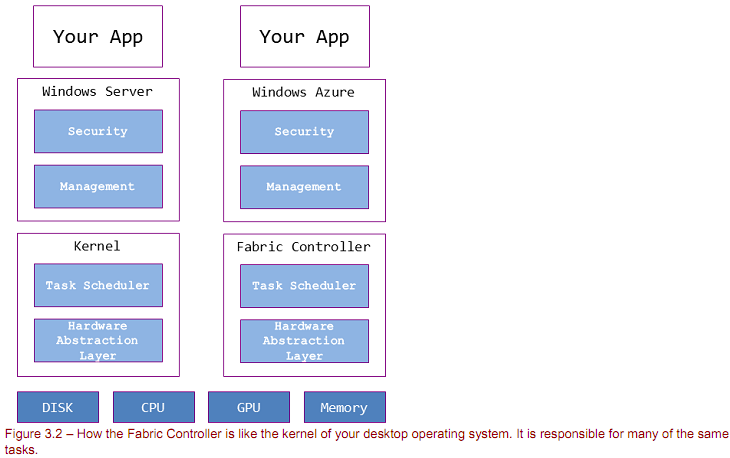
Chương này bao gồm:

* Microsoft xây dựng Azure bằng cách nào?
* Một hệ điều hành điện toán đám mây là gì?
* Làm thế nào ứng dụng của chúng ta được cung cấp và quản lý trong các đám mây.

## Windows Azure, hệ điều hành đám mây

Windows Azure, hệ điều hành điện toán đám mây, thực hiện công việc giống như hệ điều hành thông thường, nhưng trên một quy mô lớn hơn, và dành cho các ứng dụng phân tán.

Fabric bao gồm hàng ngàn máy chủ đang chạy, làm việc cùng nhau như một đơn vị gắn kết.



Windows Azure sẽ chăm sóc của toàn bộ nền tảng, nhờ đó chúng ta có thể tập trung vào việc xây dựng ứng dụng của mình. Hiểu được các mối quan hệ giữa mã nguồn của chúng ta, Azure, và fabric controller sẽ giúp chúng ta nhận được nhiều nhất của nền tảng này .

### Quản lý tài sản của đám mây

Trong thực tế, Azure không chỉ quản lý các máy chủ. Có rất nhiều tài sản khác được quản lý. Ngoài các máy chủ, Azure quản lý các thiết bị định tuyến, chuyển mạch, các địa chỉ IP, máy chủ DNS, cân bằng tải, và các VLAN động .

Nếu Azure fabric là hệ điều hành, thì fabric controller chính là hạt nhân.

## Fabric Controller

Các hệ điều hành có, cốt lõi của họ, một hạt nhân. Hạt nhân này chịu trách nhiệm cho là cảnh sát giao thông trong hệ thống. Nó quản lý việc chia sẻ các nguồn lực, lịch trình sử dụng tài sản quý (CPU), giao làm việc dòng cho phù hợp.

Vải Azure có một hạt nhân được gọi *là* bộ *điều khiển* vải, hoặc FC. Nó xử lý tất cả các công việc một hệ thống hoạt động bình thường của hạt nhân sẽ xử lý. Nó quản lý các máy chủ đang chạy, triển khai mã .

**Fabric Controller là một ứng dụng**

FC là một ứng dụng Azure trong và của chính nó, chạy nhiều bản sao của chính nó vì mục đích dự phòng.

FC chứa các trạng thái hoàn thành của vải nội bộ, và đây là thời gian thực nhân rộng cho tất cả các nút là một phần của FC . Trong trường hợp một trong các nút chính đi, các thông tin trạng thái mới nhất là có sẵn cho các nút còn lại, sau đó chọn một nút chính mới.

FC quản lý một máy nhà nước cho mỗi dịch vụ triển khai, thiết lập một trạng thái mục tiêu dựa trên những gì các mô hình dịch vụ cho dịch vụ đòi hỏi. FC, không có gì trong một nỗ lực để đạt được trạng thái này, và sau đó để duy trì trạng thái đó một khi nó đạt được.

**Làm thế nào điều khiển Vải hoạt động: lái xe mô hình**

FC sau một mô hình điều khiển, giống như một hệ thống hoạt động bình thường. FC làm việc với một loạt các trình điều khiển, đối với từng loại tài sản trong cơ cấu Azure. Điều này bao gồm máy móc, cũng như các bộ định tuyến, chuyển mạch, và cân bằng tải .

FC sử dụng các trình điều khiển để giao tiếp các lệnh mà nó cần phải gửi cho mỗi thiết bị để đạt đến trạng thái hoạt động mong muốn. Điều này có thể là lệnh để tạo ra một VLAN mới để chuyển đổi, hoặc bố trí một hồ bơi của địa chỉ IP ảo.

Hình 3.4 cho thấy sự tiến triển của nhà nước, từ các nhà phát triển viết mã, và xác định mô hình dịch vụ, FC phân bổ và quản lý các nguồn tài nguyên mà dịch vụ yêu cầu.

**Phân bổ tài nguyên**

Một trong những công việc chủ yếu của FC là phân bổ nguồn lực với các dịch vụ. Nó làm điều này bằng cách phân tích các mô hình dịch vụ của dịch vụ, bao gồm cả lỗi và cập nhật lĩnh vực, và sự có sẵn các nguồn lực trong vải. Nó tìm thấy các nút có thể hỗ trợ các nhu cầu của mỗi trường hợp trong mô hình.

Một khi nó đã đặt khả năng, nó cập nhật các cấu trúc dữ liệu FC trong một giao dịch .

**Instance Quản lý**

FC cũng chịu trách nhiệm quản lý sức khỏe của tất cả các nút trong vải, cũng như sức khỏe của các dịch vụ đang chạy.

Nếu nó phát hiện một lỗi trong một dịch vụ, nó sẽ cố gắng để khắc phục lỗi đó, có lẽ bằng cách khởi động lại các nút, hoặc dùng nó chưa có mặt trong diễn đàn và thay thế nó bằng một nút khác nhau trong vải.

Khi một container mới được thêm vào các trung tâm dữ liệu, FC thực hiện một loạt các cháy trong các thử nghiệm để đảm bảo rằng các phần cứng giao là làm việc đúng cách, triển lãm MAKS nó có sẵn được phân bổ bởi FC .

Nếu phần cứng là bao giờ được xác định bị lỗi, hoặc trong quá trình cài đặt, hoặc trong một lỗi, phần cứng đánh dấu là không sử dụng được trong hàng tồn kho, và để lại một mình cho đến sau này. Một khi các vấn đề đã được cố định, các container toàn là kiểm tra lại và quay trở lại vào dịch vụ.

**Mô hình dịch vụ và bạn**

Động lực đằng sau những gì FC là mô hình dịch vụ mà bạn xác định cho dịch vụ của bạn. Bạn xác định mô hình dịch vụ một cách gián tiếp. Khi bạn đang phát triển một dịch vụ, bạn định nghĩa sau đây:

* một số cấu hình trên các mảnh dịch vụ của bạn là gì
* làm thế nào các miếng giao tiếp
* mong đợi của bạn có về sự sẵn có của dịch vụ

Mô hình dịch vụ được chia thành hai phần của cấu hình và triển khai với dịch vụ của bạn.

**Xác định cấu hình**

Giải pháp của bạn trong Visual Studio sẽ có hai phần cấu hình trong tập tin khác nhau, cả hai được tìm thấy trong các dự án dịch vụ Azure trong giải pháp của bạn:

* dịch vụ định nghĩa tập tin *(ServiceDefinition.csdef)*
* cấu hình dịch vụ tập tin *(ServiceConfiguration.cscfg).*

Các tập tin định nghĩa dịch vụ xác định vai trò trong dịch vụ của bạn, và những gì thiết bị đầu cuối thông tin liên lạc của họ. Điều này sẽ bao gồm HTTP giao thông công cộng cho một trang web, hoặc các chi tiết thiết bị đầu cuối cho một dịch vụ web . Định nghĩa dịch vụ không có thể thay đổi thời gian chạy. Bạn có thể nghĩ này phần của cấu hình như xác định cơ sở hạ tầng dịch vụ của bạn là gì, và làm thế nào các bộ phận phù hợp với nhau.

Các tập tin cấu hình dịch vụ bao gồm toàn bộ các cấu hình cần thiết cho các trường hợp vai trò quan trọng trong dịch vụ của bạn. Mỗi vai trò có một phần của cấu hình riêng của nó. Các nội dung của file cấu hình có thể thay đổi thời gian chạy.

Bạn cũng có thể truy cập vào cấu hình trong mã, một cách tương tự mà bạn có thể đọc một *tập tin* web.config trong một ứng dụng ASP.NET .

**Thêm một yếu tố cấu hình tùy chỉnh**

Trong nhiều ứng dụng, chúng tôi lưu trữ các kết nối dây, các thiết lập mặc định trong app.config *hoặc* web.config . Bạn thường sẽ làm tương tự với một ứng dụng Azure.

Chúng tôi cần khai báo định dạng của các thiết lập cấu hình mới trong *tập tin.* Csdef . Chúng tôi làm điều này bằng cách thêm một nút ConfigurationSettings bên trong vai trò chúng ta muốn cấu hình thuộc về . Chúng tôi làm điều này để xác định sơ đồ của các *tập tin* cscfg. Cho vai trò đó.

<ConfigurationSettings>

<Setting Name="SuperSecretPassword"/>

</ ConfigurationSettings>

Chúng tôi thêm XML sau đây vào nút vai trò thích hợp trong *tập tin.* Cscfg .

    <ConfigurationSettings>

<Setting Name="SuperSecretPassword" value="KlatuBaradaNikto"/>

</ ConfigurationSettings>

Trong thời gian chạy, chúng tôi muốn đọc dữ liệu cấu hình này, và sử dụng nó cho một số mục đích . Hãy nhớ rằng tất cả các thiết lập cấu hình được lưu trữ như các chuỗi và phải được đúc các loại thích hợp khi cần thiết.

txtPassword.Text = RoleEnvironment.GetConfigurationSettingValue ("BannerText");

**Đọc mã tập tin tập trung**

Đó là một thực hành tốt nhất để đọc mã toàn bộ cấu hình tập tin của bạn, bao giờ nó là rắc, và di chuyển nó vào một lớp học ConfigurationManager thiết kế của riêng bạn. Điều này centralizes tất cả các mã mà biết làm thế nào để đọc các cấu hình ở một nơi, làm cho nó dễ dàng hơn để duy trì.

Trong khi vẫn giữ chúng tại chỗ làm giảm số lượng thay đổi mã của bạn khi bạn di chuyển nó đến Azure, nó không đi kèm với chi phí. Thật không may, các tập tin cấu hình là một phần của vai trò của bạn là đông lạnh, và chỉ được đọc tại thời gian chạy. Bạn không có thể thay đổi cho họ một khi gói của bạn được triển khai . Nếu bạn muốn thay đổi các thiết lập trong thời gian chạy, bạn sẽ cần để lưu trữ những thiết lập trong các tập *tin.* Cscfg . Sau đó, khi bạn muốn thực hiện một sự thay đổi, bạn chỉ phải tải lên *một file.* Cscfg mới, hoặc nhấn vào cấu hình trên trang quản lý dịch vụ cổng thông tin.

**Nhiều kích cỡ của vai trò**

Mỗi vai trò được định nghĩa trong mô hình dịch vụ của bạn là cơ bản một khuôn mẫu cho một máy chủ mà bạn muốn triển khai trong vải . Mỗi vai trò có thể có một công việc khác nhau, và một cấu hình khác nhau. Một phần của cấu hình bao gồm lưu trữ địa phương, và số trường hợp vai trò đó cần được triển khai . Làm thế nào các vai trò kết nối và làm việc cùng nhau là một phần của lý do tại sao các mô hình dịch vụ tồn tại .

Vì mỗi vai trò có thể có những nhu cầu khác nhau có rất nhiều các kích cỡ máy ảo mà bạn có thể yêu cầu trong mô hình của bạn.

Có kích thước mà bạn muốn được định nghĩa trong tập tin *ServiceDefinition.csdef,* một vai trò cơ sở vai trò. Kích thước mặc định, nếu bạn không khai báo, là nhỏ .

<WorkerRole Name="ImageCompresser" vmsize="ExtraLarge">

Nếu bạn đang sử dụng Visual Studio 2010, bạn có thể xác định cấu hình vai trò trong màn hình thuộc tính của các vai trò trong dự án Azure :