MAKING YOUR APPLICATION CLOUD READY

Michael Friis
AppHarbor
Michael Friis
@friism

Co-founder of AppHarbor (@appharbor)
.NET Platform as a Service
San Francisco | Copenhagen
(we’re hiring)
What are we going to cover?

• Obstacles to scaling
• Make the move to the cloud an easy one
What do we know?

- AppHarbor is a .NET platform as a service (running on top of AWS EC2)
- We’re like Heroku, but for .NET
- 8500 developers are running around 4000 .NET applications on our platform
- Some of them need help getting applications to run and scale
- AppHarbor runs on AppHarbor
An Aside on Deployment

- I’m using Git to deploy to AppHarbor
- There are plenty other options
- Or you can roll your own
- (I think you should use AppHarbor)
Demo

Get example running
And see what happens if we scale it...
What’s wrong with applications people deploy on AppHarbor?

• They rely on instance file systems
  • Hard to sync between multiple instances
  • Data disappears if instance fails (and they do)

• Application state is not shared
  • (e.g. .NET session state stored in memory)
  • User state is stored in instance memory

• Unmanaged dependencies
Bad Mental Model

- Big honkin’ web server
- HTTP
- File Storage (including SQL Server CE, etc.)
- In-memory Session Storage
- Email
- Running .exe’s, etc.
Bad Mental Model

Big honkin’ web server

HTTP

File Storage (including SQL Server CE, etc.)

In-memory Session Storage

Email

Running .exe’s, etc.

Big honkin’ web server

HTTP

File Storage (including SQL Server CE, etc.)

In-memory Session Storage

Email

Running .exe’s, etc.
A better model for cloud-hosted apps

- Load Balancer
  - Perishable Web Instance
    - HTTP
      - Session State
      - File persistence
  - Perishable Web Instance
    - HTTP
      - Database
  - Perishable Web Instance
    - HTTP
      - Background processing
      - Email Service
Demo

Change from local file system to S3
Use email service
What else?

- Session state and cache: On ASP.NET, use memcached-backed provider
- Always use off-instance, shared databases
- Take advantage of all the great 3rd party services in the cloud
Easing deployment

• Application should be self-contained
  • Do not expect infrastructure to provide dependencies
Easing deployment 2

• Application should be self-configuring
  • No possibility of human mistakes in setup
  • No time wasted changing configuration strings etc.
  • Data schema management should be automated

• Project should be self-testing
Lessons

• There are good alternatives to managing own infrastructure
• Get the mental model right for scaling and failure
• Organize your application for the cloud
• Take advantage of all the great services
Questions?

http://appharbor.com/

@appharbor

San Francisco | Copenhagen