Pivotal

Open.
Agile.
Cloud-Ready.

Agenda

- Stories from our Customers
- .NET on Pivotal Cloud Foundry
- .NET Apps
- Operating .NET Infrastructure
- Conclusion

Pivotal

.NET deployment

A few comments of what customers have told us about deploying on premise

Pivotal

Deploying a new app

- Provision a VM, IP, DNS, etc
- Windows updates...reboot...update...reboot
- Install IIS
- Create Website, configure app pool
- Configure SSL
- Configure Load Balancer

Pivotal

...now scale up

- get another VM
- template might have IIS installed
- duplicate your IIS config
- rework the load balancer

After all this, do you ever scale down?

Pivotal

Phew...now I can deploy my code

- manually doing one of:
  - Package your webapp as an MSI and coordinate pulling nodes out of the cluster to install
  - Login to the machine, download a zip file and copy it to c:\inetpub (or maybe run a .bat file)
  - The really modern teams are using NuGet for bits and most of the rest is the same

Pivotal

And when you have a problem

- Ops log in to each machine in the cluster and download log files and/or event log
  - A few hours (or days) later you get them and actually get to start diagnosing issues

Pivotal

Question

You want to do an isolated test of new code and need a place for it to run. Do you:

A. Build a new vm and install everything?
B. Clone an existing VM and hope you don't break anything? If so, how long do you wait?
C. Find a "roomy" dev server and add it there?

Pivotal

Bonus Question

After that new environment gets created:

A. It lives forever because no one will delete it?
B. I gave a developer admin rights on the box, they'll surely clean it up when they're done.
C. Here is the URL for my ticketing system.

Pivotal

Pivotal

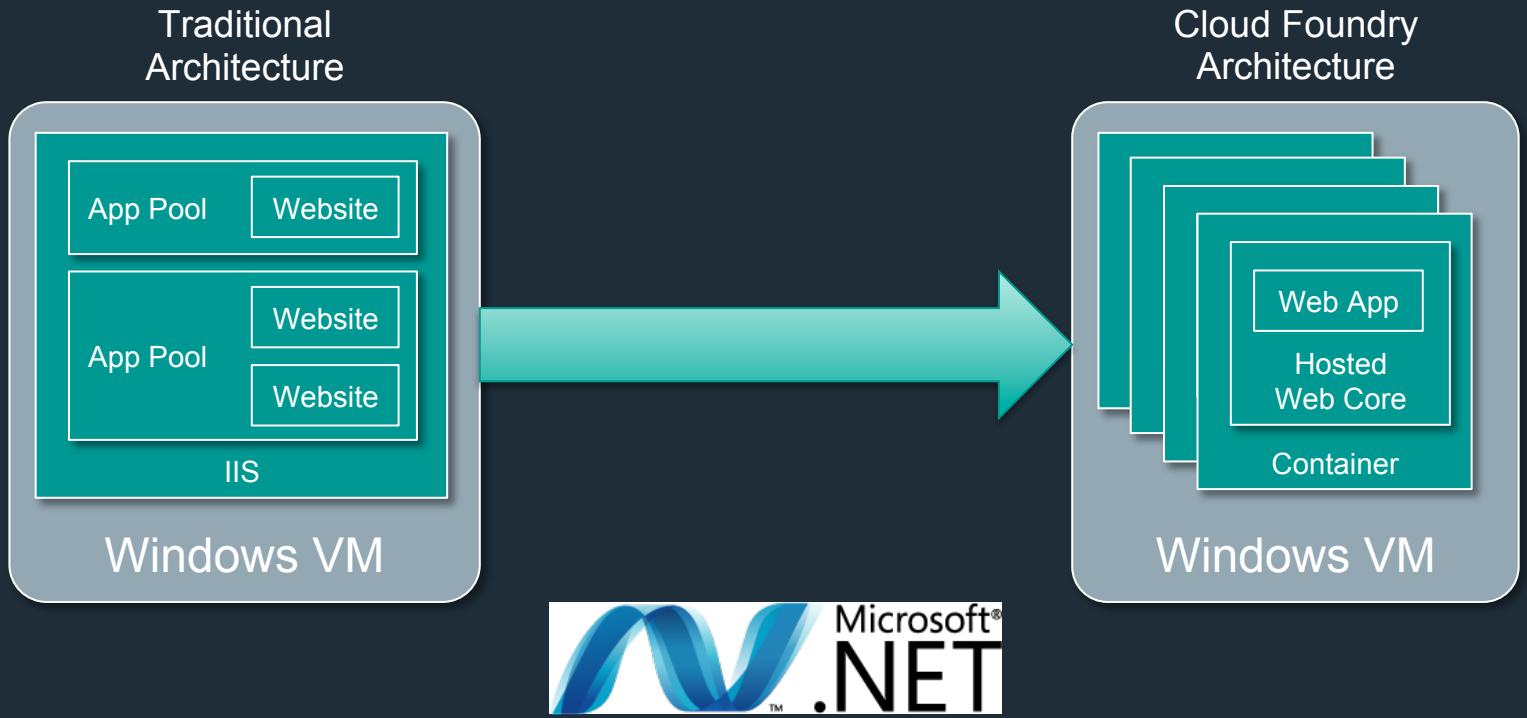
.NET on Pivotal Cloud Foundry

...with Cloud Foundry

- Deploy: cf push
  - deploy app, configure SSL/load balancing
- Scale: cf scale
  - add/remove app instances, updates load balancer
- View Logs: cf logs
  - Developers can see logs in real time!

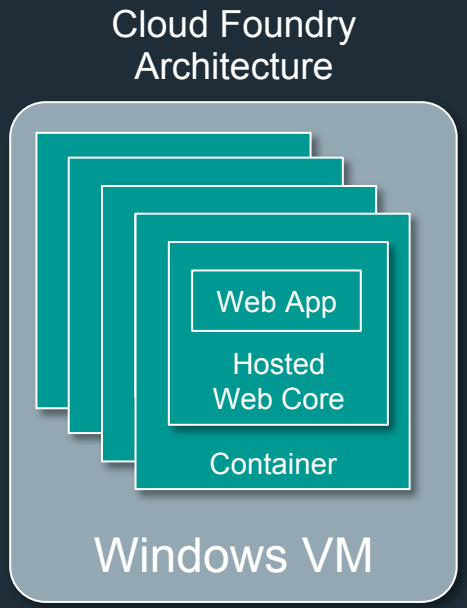
Pivotal

App Pools, IIS, Containers, VM's, oh my.



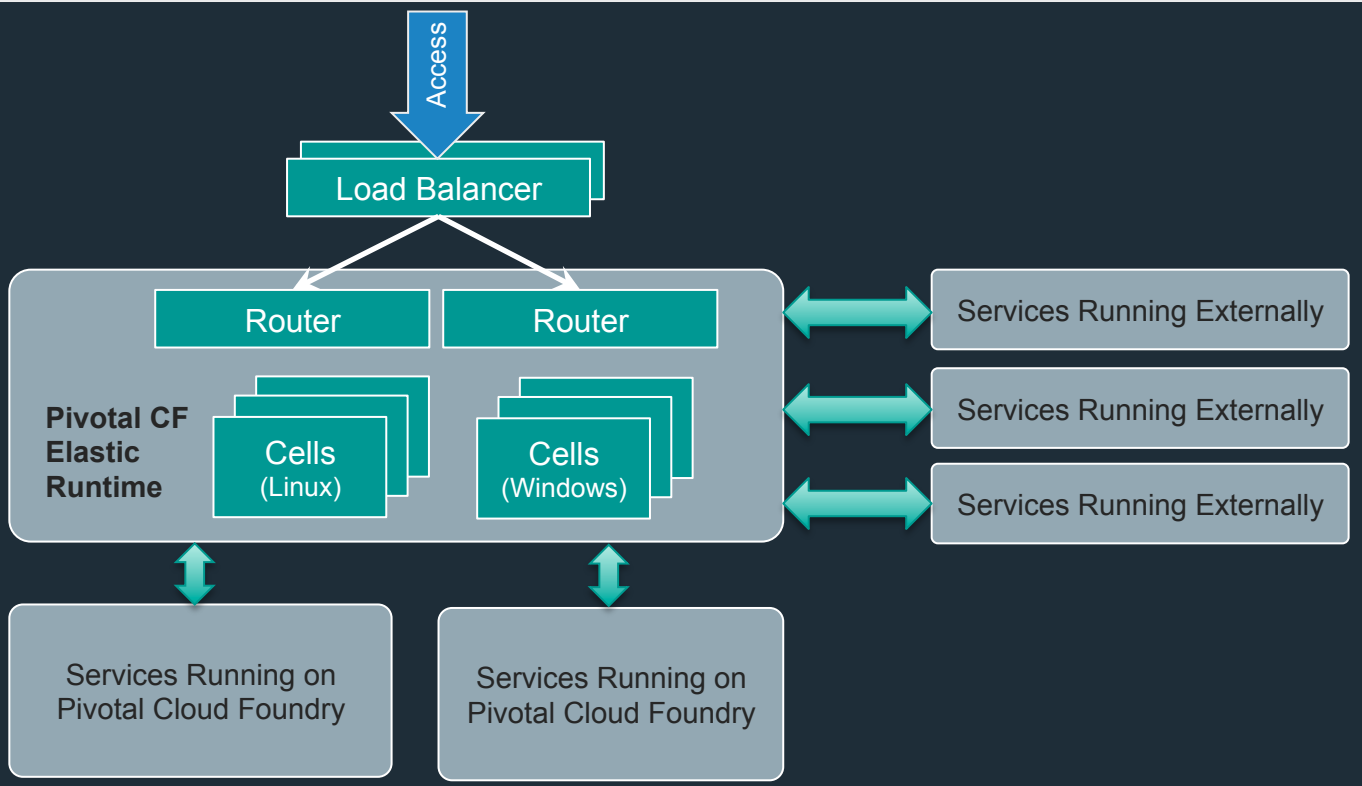
Pivotal

Isolation – Work in progress



Pivotal

Cloud Foundry Architecture



Pivotal

Key Features

- High Availability
- Scaling – Manual and Autoscaling
- Notifications
- Security Groups
- Aggregated Logs & Metrics
- Environment Variables
- Marketplace and User-Provided Services

Pivotal

Integrating with Services

- Marketplace Services
  - –Extract connection details from environment variables

- Connection Strings in Web.config
  - via User Provided Services
  - Spliced into Web.config during staging

Pivotal

Pivotal

.NET apps

Writing 12 Factor .NET Apps

- Use an external session/cache provider
  - RDBMS, Redis or GemFire
- Override MachineKey in Web.config
- Use environment variables for environment config
- Use OAuth instead of Integrated Windows Auth
- Avoid the GAC and custom ISAPI handlers

Pivotal

Pivotal

Operating .NET apps on Pivotal Cloud Foundry

Operations with .NET

- Build a pool of Windows VMs and connect to runtime
- Use a separate subnet or OpsManager exclusion list
- Monitor capacity using standard monitoring tools (SCOM, vROps, etc)

Pivotal

Pivotal

Conclusion

Conclusion

- Cloud Foundry runs .NET apps
  - –just like java, ruby, go, php, python, Node.js … … … ...
- on Windows (not mono)
- in any cloud

Pivotal

Pivotal

Open.
Agile.
Cloud-Ready.

Background Slides

These are probably not useful for customer facing presentations.

Resources

- Slack
  - #greenhouse
  - #sme-pcf-dotnet
- Source code https://github.com/cloudfoundry-incubator/diego-windows-release
- Workshop resources
  - https://github.com/afortener/Humana-Workshop
  - https://github.com/rossr3-pivotal/AudaExplore-Workshop
- Subject Matter Experts
  - Aaron Fortener
  - Rick Ross
- .NET Product Manager
  - Steven Benario

Pivotal

Pushing a .NET Application

- To push:
  - –binary buildpack
  - –windows2012R2 stack
- Either publish or push from your project folder

Pivotal

Supported .NET SDK Versions

- All versions of the .NET SDK supported by Windows Server 2012 R2
  - 3.5
  - 4.5.x
  - 4.6

Pivotal

Supported Application Types

- ASP.NET web applications
- OWIN apps
- Background processes
  - Command line apps

Pivotal

Pivotal

Supporting .NET apps on Pivotal Cloud Foundry

Diego on Windows - Preparation

- Windows Server 2012 R2
  - Install Windows Updates
  - Needs to be on a different network than BOSH
  - Alternatively assign IP from exclusion range
  - Using RDP? Disable CPU Fair Share http://bit.ly/cpufairshare
- Download bits for network.pivotal.io
  - Setup.ps1
  - Generate.exe

Pivotal

Diego on Windows - Configuration

- Run setup.ps1
  - Configures Windows features
- Run generate.exe in cmd prompt
  - Specify bosh url (https://un:pwd@director.sys.pcf.com)
  - Provide a Windows username – must be part of Admin group
  - Provide password for the username
  - Specify output directory

Pivotal

Diego on Windows - Install

- Download bits from network.pivotal.io
  - DiegoWindows.msi
  - GardenWindows.msi
  - Save/Move them to the output folder
- Run install.bat in cmd prompt
  - Generated in the output folder

Pivotal

Diego on Windows - Verification

- Download bits from network.pivotal.io
  - hakim.exe
  - Creates a container to make sure everything works
- Check Task Manager | Services for running processes
  - GardenWindowsService
  - ContainerizeService
  - MetronService
  - RepService
  - ConsulService
- Run Smoke Tests
  - https://github.com/cloudfoundry/cf-smoke-tests
  - Be sure you enable windows tests in the JSON config file

Pivotal

Diego on Windows - Uninstall

- Evacuate Cells
  - Use the script here: http://bit.ly/evacuatecell
- Uninstall from Control Panel | Uninstall a program
  - In the following order
    - DiegoWindows
    - GardenWindows

Pivotal