Introduction to Cloud Design Patterns

Global Azure Bootcamp 2017

Twin Cities, MN

Jerry Nettleton | @dotnett

- Sr. Consultant, Software Engineering @ Avanade
- Over 20 years of industry experience
  - ➢Worked for wide range of companies in Twin Cities
  - ➢Mostly new product development
- Other interests
  - ➢Boy Scouts, Lions Club
  - ➢Cycling
  - ➢FRC Programming Mentor
  - ➢CoderDojo TC Arduino Mentor
  - ➢EV Owner

What’s Next

- Take More Courses
  - Opsgility – http://www.opsgility.com
  - Microsoft Virtual Academy - http://mva.Microsoft.com
  - Pluralsight – http://www.pluralsight.com
  - LearnNowOnline – http://www.learnnowonline.com
  - Local Sponsors for Training, Consulting, and Development Services
- Get More Experience
  - Create sample applications
  - Find more labs and tutorials

Transition to Cloud Services

- Traditional IT Services
    - Own the entire service stack
    - Failure prevention
        - No single point of failure, redundancy, oversized for max capacity
    - Fragile software on robust infrastructure
        - Big iron machines

- Cloud Services
    - Embrace failure (e.g. Netflix outage)
    - Recovery oriented
    - Robust software on fragile infrastructure
        - Ability to scale up/down, scale out/in using commodity hardware

- Where to start
    - Migrate existing web app
    - Create a new web app

Modernize Your Websites with Azure PaaS

- Pluralsight Course by Troy Hunt
  - https://app.pluralsight.com/library/courses/modernizing-websites-microsoft-azure/table-of-contents
- Outline
  - Initial Project
  - Cleaning Up the Application
  - Provisioning, Configuring, and Deploying Website
  - Setting Up Slots and Staged Deployment from Source Control
  - Migrating the Database to Azure SQL Database Service
  - Monitoring and Diagnostics
  - Beyond the Basics

Web Application Performance and Scalability Testing

- MVA Course
  - https://mva.microsoft.com/en-US/training-courses/designing-applications-for-windows-azure-jump-start-8285
- Outline
  - Failsafe Overview
  - Reliability, Threats, and Mitigations
  - Modeling Applied
  - Patterns for Resiliency and Availability
  - Patterns for Scalability

Introduction to Design Patterns

- Definition
  - Reusable solution to a commonly occurring problem with a given context in software design
- Lots of resource
  - Books – Recipes, Frameworks, Cookbooks, Patterns, Tutorials, Samples
  - Samples – http://www.dofactory.com
- Books
  - "Design Patterns: Elements of Reusable Object-Oriented Software" (GoF)
  - "Patterns of Enterprise Application Architecture" – Martin Fowler
  - "Enterprise Integration Patterns" – Gregor Hohpe

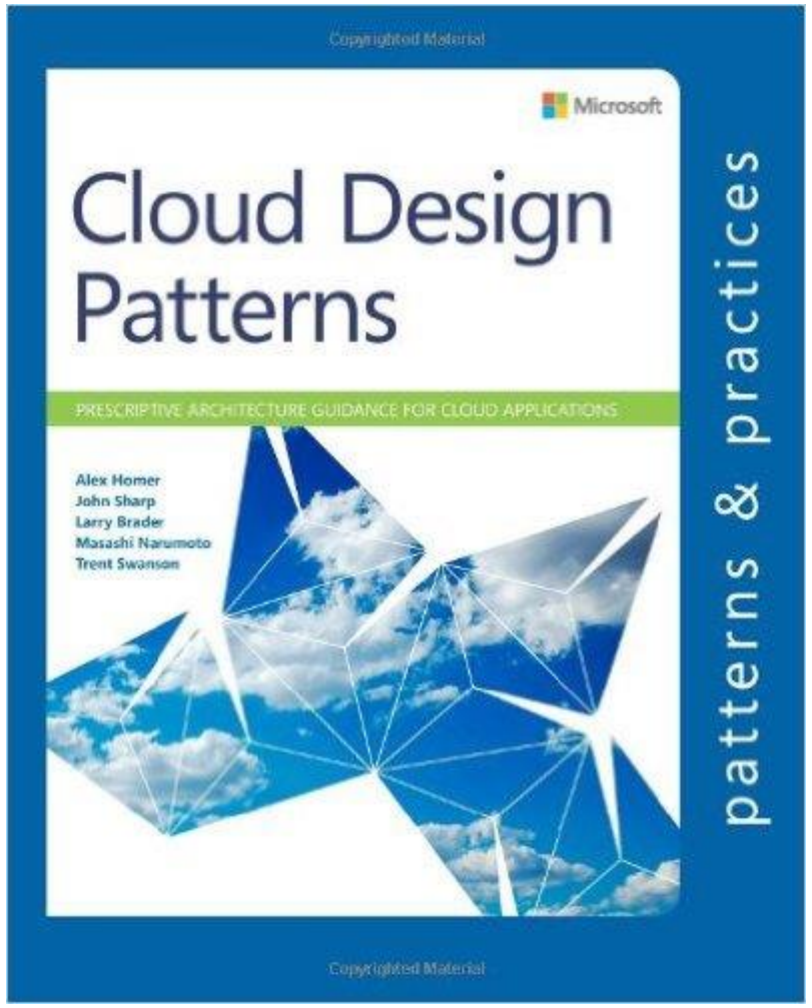
Cloud Design Patterns

24 design patterns

10 related guidance topics

Demonstrates benefits of applying patterns by showing how each piece can fit into the big picture of cloud application architectures. It includes code samples and general advice on using each pattern.

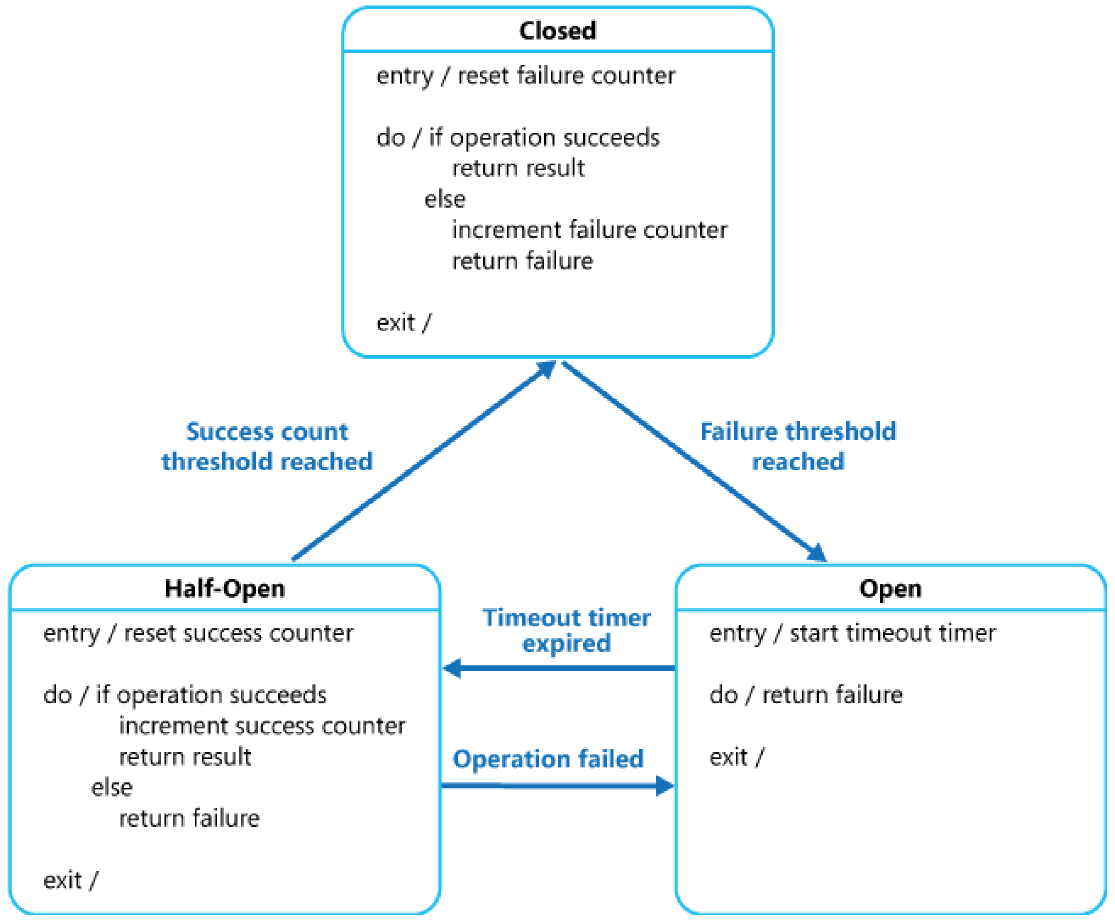
- https://msdn.microsoft.com/en-us/library/dn568099.aspx
- https://www.microsoft.com/en-us/download/details.aspx?id=42026



Circuit Breaker - Solution

- Handle faults that may take a variable amount of time to rectify when connecting to a remote service or resource.

https://msdn.microsoft.com/en-us/library/dn589784.aspx



Closed
entry / reset failure counter
do / if operation succeeds
return result
else
increment failure counter
return failure
exit /
Success count threshold reached
Failure threshold reached
Half-Open
entry / reset success counter
do / if operation succeeds
increment success counter
return result
else
return failure
exit /
Timeout timer expired
Operation failed
Open
entry / start timeout timer
do / return failure
exit /

Circuit Breaker - Considerations

- Exception Handling
    - Operation is unavailable, degrade temporarily, invoke alternative, report error to user, try again later
- Types of Exceptions
    - Remote service crashes and take several minutes to recover; timeout because remote service is overloaded; adjust strategy may depend on type of exception; multiple timeouts before tripping circuit breaker
- Logging
    - Log all failed requests to monitor health of encapsulated operation
- Recoverability
    - Match likely recovery pattern of operation it is protecting;
- Testing Failed Operations

Circuit Breaker - Considerations

- Manual Override
- Concurrency
- Resource Differentiation
- Accelerated Circuit Breaking
- Replaying Failed Requests
- Inappropriate Timeouts on External Services

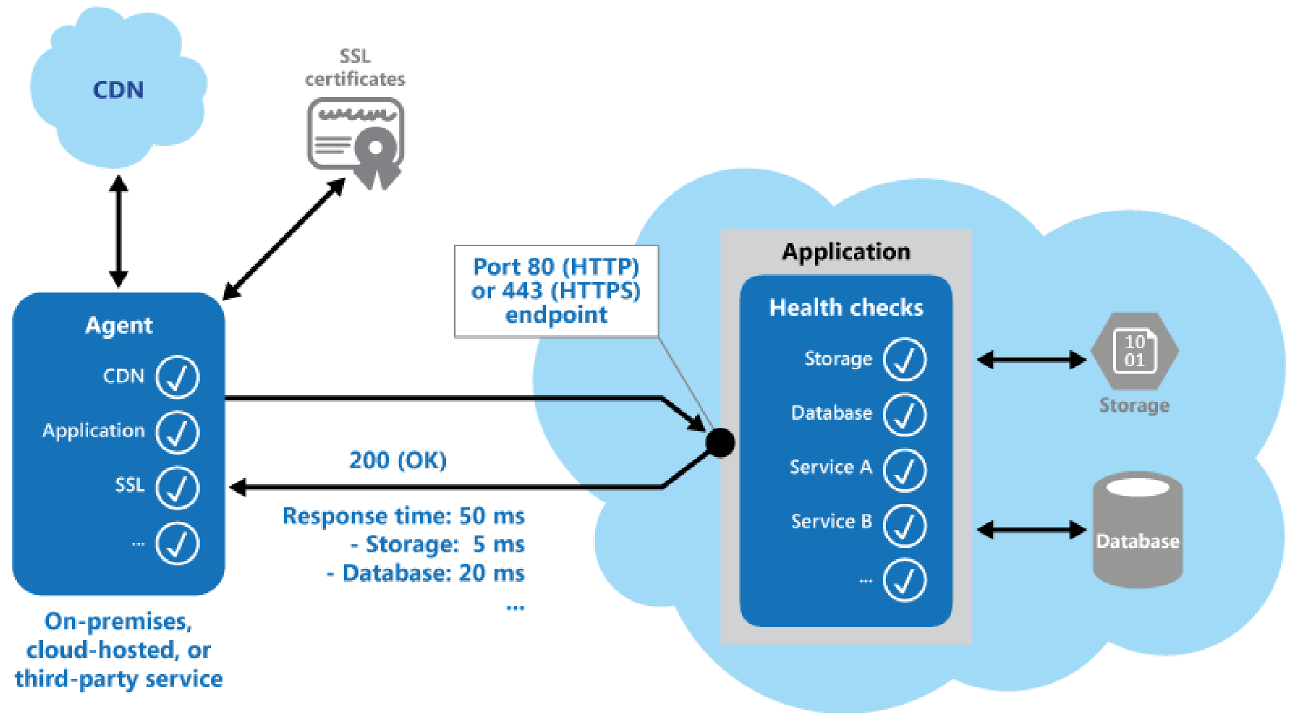
Circuit Breaker – When (not) to Use

- Use this pattern
  - Prevent applications from attempting to invoke a remote service that is highly likely to fail
- Not suitable
  - For handling access to local privates resource in an application (e.g. in-memory data structure)
  - As a substitute for handling exceptions in business logic

Health Endpoint Monitoring

- Implement functional checks within an application that external tools can access through exposed endpoints

https://msdn.microsoft.com/en-us/library/dn589789.aspx



Health Endpoint Monitoring - Considerations

- Validate the response
- Type of information to collect
- Number of endpoints to expose
- How to secure the endpoints
- How to ensure monitoring agent is performing correctly

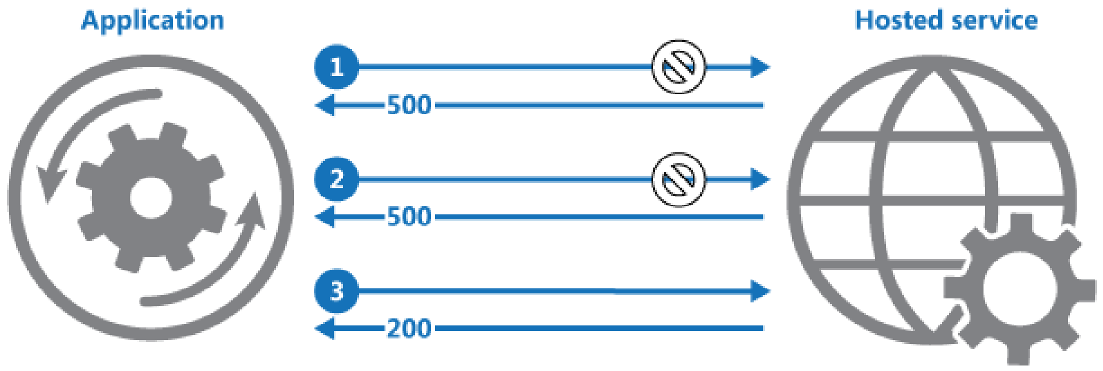
Health Endpoint Monitoring – When to Use

- Monitor applications to verify availability
- Monitor middle-tier or shared services to detect and isolate a failure
- Complement existing instrumentations

Retry Pattern

- Enable an application to handle anticipated, temporary failure when it attempts to connect to a service or network resource by transparently retrying an operation that has previously failed in the expectation that the cause of the failure is transient
- "Transient Fault Handling Block" to handle transient faults using a range of retry strategies

https://msdn.microsoft.com/en-us/library/dn589788.aspx



1: Application invokes operation on hosted service. The request fails, and the service host responds with HTTP response code 500 (internal server error).
2: Application waits for a short interval and tries again. The request still fails with HTTP response code 500.
3: Application waits for a longer interval and tries again. The request succeeds with HTTP response code 200 (OK).

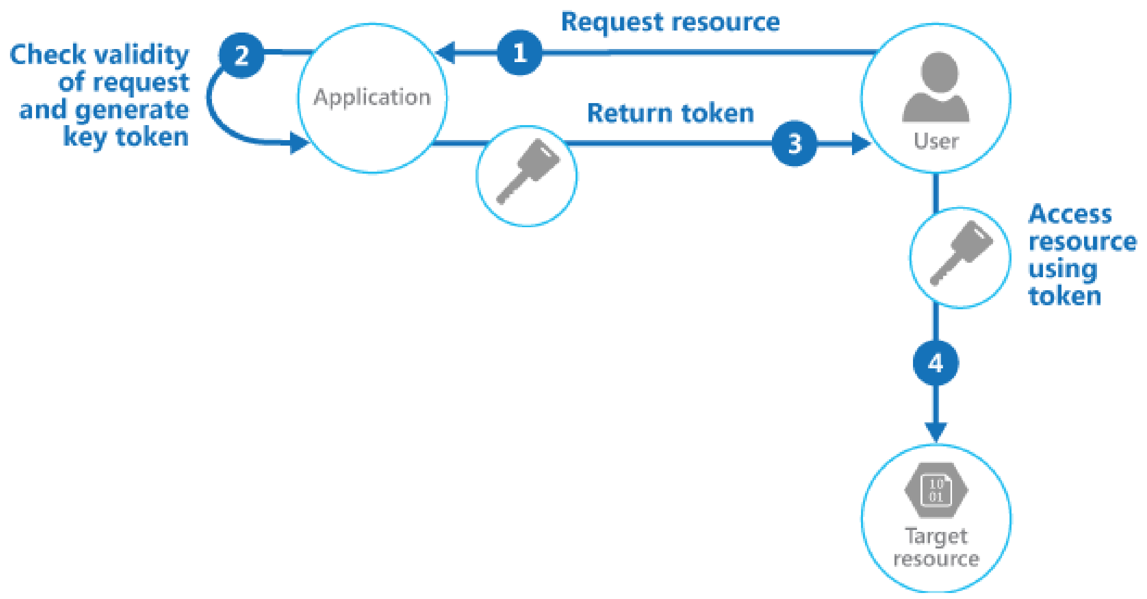
Retry - Considerations

- Retry policy should be tuned to match requirements of application and nature of failure
- Aggressive policy with minimal delay and large number of retries could degrade busy service
- Wait for a longer period after significant number of retries
- How retrying an operation that is part of transaction will affect overall transaction consistency

Valet Key Pattern

- Use a token or key that provides clients with restricted direct access to a specific resource or service in order to offload data transfer operations from application code

https://msdn.microsoft.com/en-us/library/dn568102.aspx


Check validity of request and generate key token
2
Application
1
Request resource
Return token
3
User
Access resource using token
4
Target resource

Valet Key - Considerations

- Manage the key validity and period
- Control the level of access
- Protect sensitive data in transit
- Validate or even sanitize uploaded data
- Ensure start time is little earlier than current server time

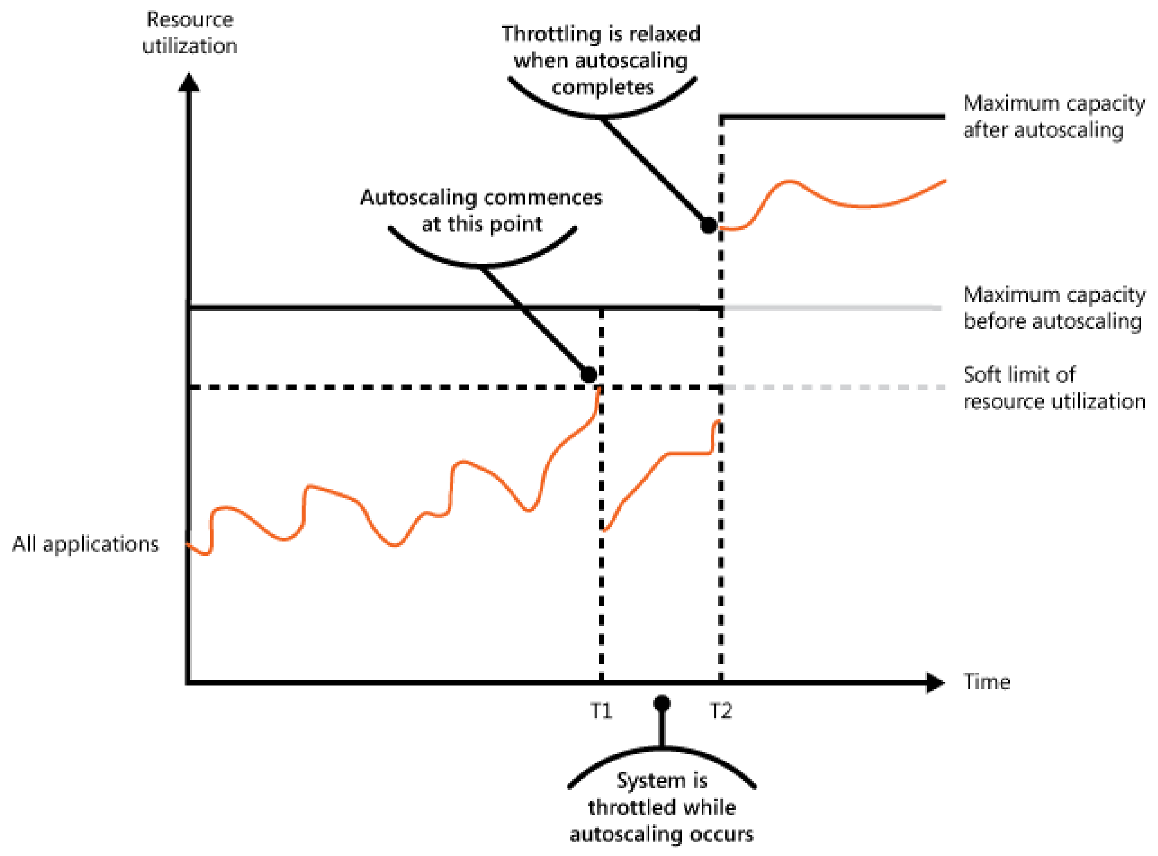
Valet Key – When (not) to Use

- Use this pattern
    - Minimize resource loading
    - Maximize performance and scalability
    - Minimize operational cost
- Not suitable
    - App must perform tasks on data
    - App must limit the size of data uploaded

Throttling

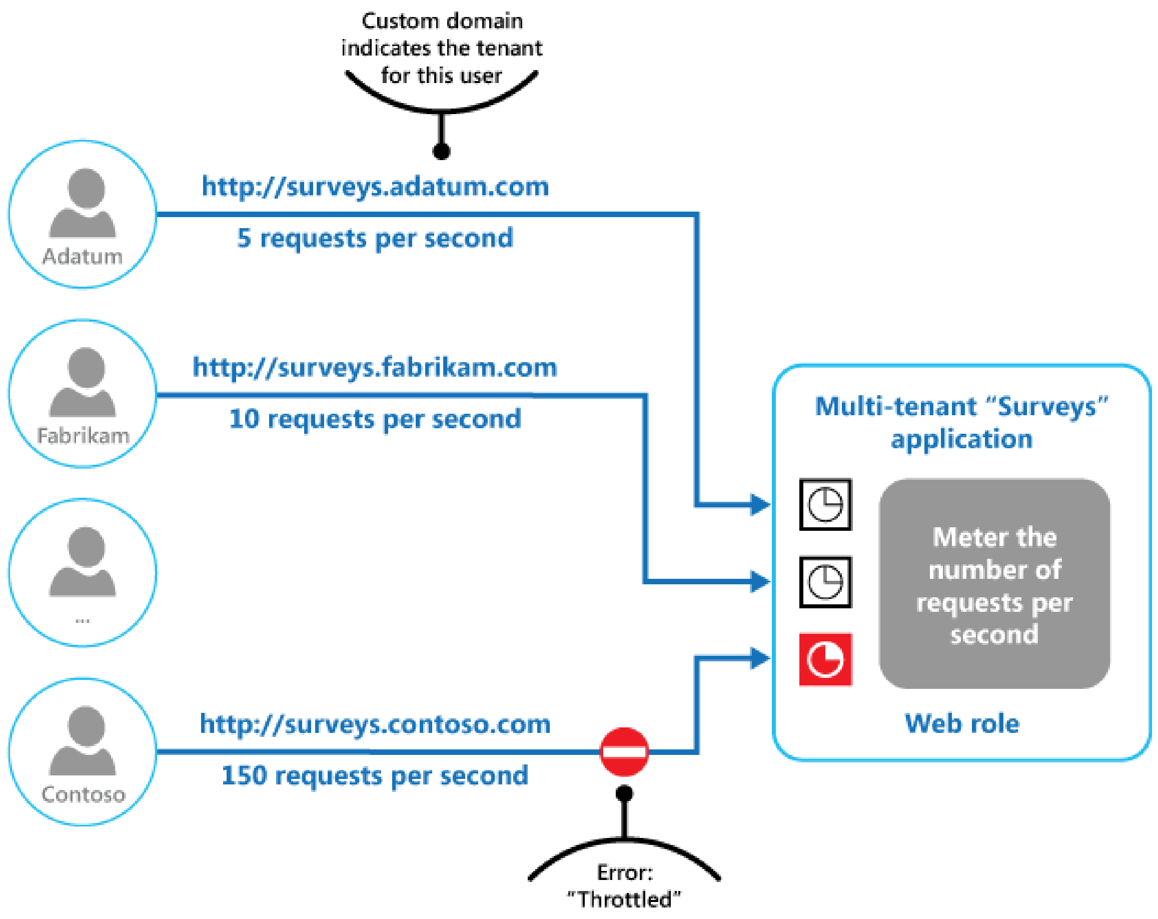
- Control consumption of resources used by instance of application. Allow system to continue to function and meet SLAs even when increased demand causes extreme load

https://msdn.microsoft.com/en-us/library/dn589798.aspx


Resource utilization
Throttling is relaxed when autoscaling completes
Maximum capacity after autoscaling
Autoscaling commences at this point
Maximum capacity before autoscaling
Soft limit of resource utilization
All applications
Time
T1
T2
System is throttled while autoscaling occurs

Throttling (cont.)

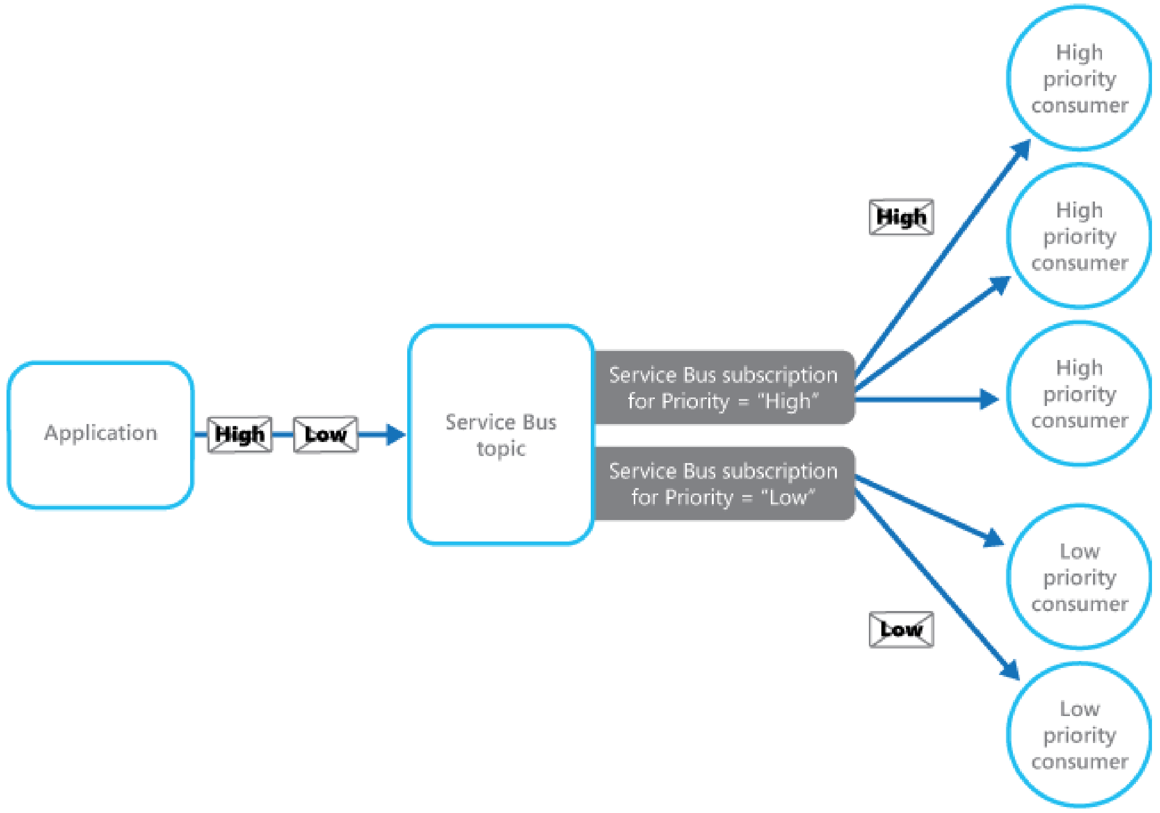
- Implemented in multi-tenant system – each user can access an API based on there subscription level (paid customer vs. free access)


Custom domain indicates the tenant for this user
Adatum
http://surveys.adatum.com
5 requests per second
Fabrikam
http://surveys.fabrikam.com
10 requests per second
...
Contoso
http://surveys.contoso.com
150 requests per second
Error: "Throttled"
Multi-tenant "Surveys" application
Meter the number of requests per second
Web role

Priority Queue Pattern

- Prioritize requests sent to services so that requests with a higher priority are received and processed more quickly

https://msdn.microsoft.com/en-us/library/dn589794.aspx



Cloud Patterns Resources

- Implementing Modern Cloud Patterns
  - https://channel9.msdn.com/Events/TechEd/Australia/2013/AZR322

- StockTrader
  - .NET StockTrader 6.1 Sample Application
    - End-to-end sample application for Microsoft Azure. It's a service-oriented application based on WCF and ASP.NET with a RESTful backend. If includes cross-platform clients for HTML5, Windows 8, Windows Phone, Android, and iOS that connect to the Azure Cloud Services backend. The backend database is Azure SQL Database (optional support for scale-out via SQL Database Federation).
    - https://msdn.microsoft.com/en-us/vstudio/bb499684.aspx
  - Updating Microsoft's StockTrader Sample Application
    - http://www.intergen.co.nz/blog/james-carpinter/dates/2012/10/updating-microsofts-stocktrader-sample-application/
  - Building Cross-Device Mobile Applications Powered by Windows Azure and SQL Azure
    - https://channel9.msdn.com/Events/TechEd/Australia/2012/AZR225

Training Resources

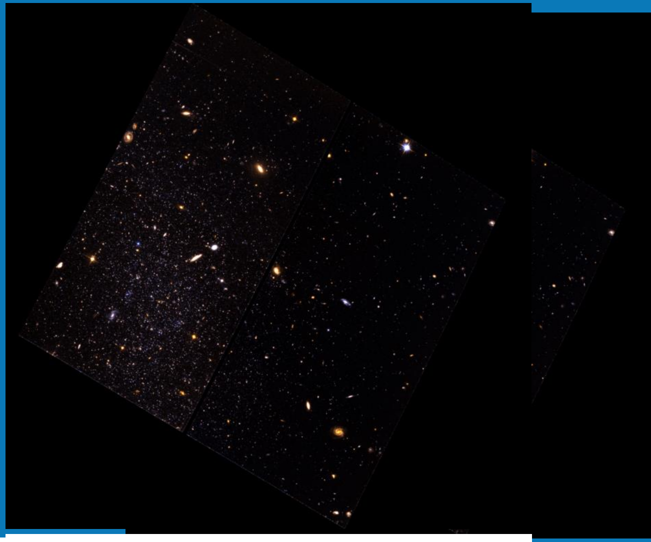
MVA

- Microsoft Azure Fundamentals
  https://mva.microsoft.com/training-courses/microsoft-azure-fundamentals
- Applications on Azure: Putting All the Pieces Together
  https://mva.microsoft.com/en-US/training-courses/applications-on-azure-putting-all-the-pieces-together-14429
- Hybrid Cloud Websites
  https://mva.microsoft.com/en-US/training-courses/hybrid-cloud-websites-8913
- Dev/Test in the Cloud
  https://mva.microsoft.com/en-us/training-courses/developing-microsoft-azure-solutions-8481

Global Azure Bootcamp 2017
Science Lab

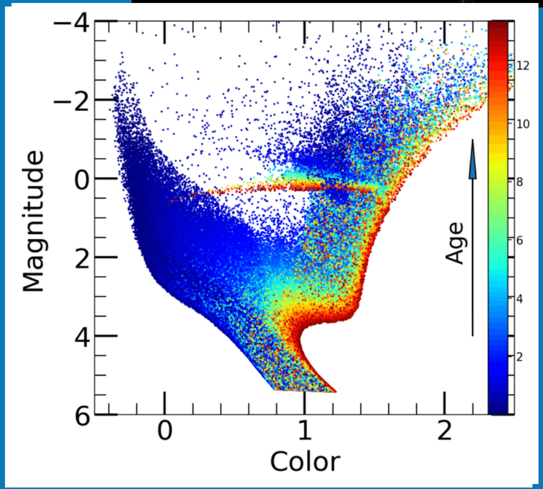
2017
Global Azure
BOOTCAMP

The Secret Life of Galaxies



In previous years we have tried to bring you a science lab that is both interesting and represents great research. This year is no exception, this year we bring you the stars themselves!

The Science Lab takes a look at Star Formation History (SFH) to see how many stars of a specific metallicity are formed in a galaxy during its lifetime. Traditional approaches are affected by uncertainties associated with observations or the limited accuracy of models when running computer code, making it difficult to accurate calculate the age of stars within a galaxy.



To overcome this, we need to correct for these effects to obtain the "true" SFH. The objective of the Seliga (SEcret LIfe of GAlaxies) algorithm, developed by Sebastian L. Hidalgo from the Instituto de Astrofísica de Canarias (IAC), is to limit the impact of all these effects so we can compare the predictions of the models more directly with the observations. This task needs a huge number of tests that can be performed successfully only by using distributed computing, like the one you will deploy in the Global Azure Bootcamp Science Lab.

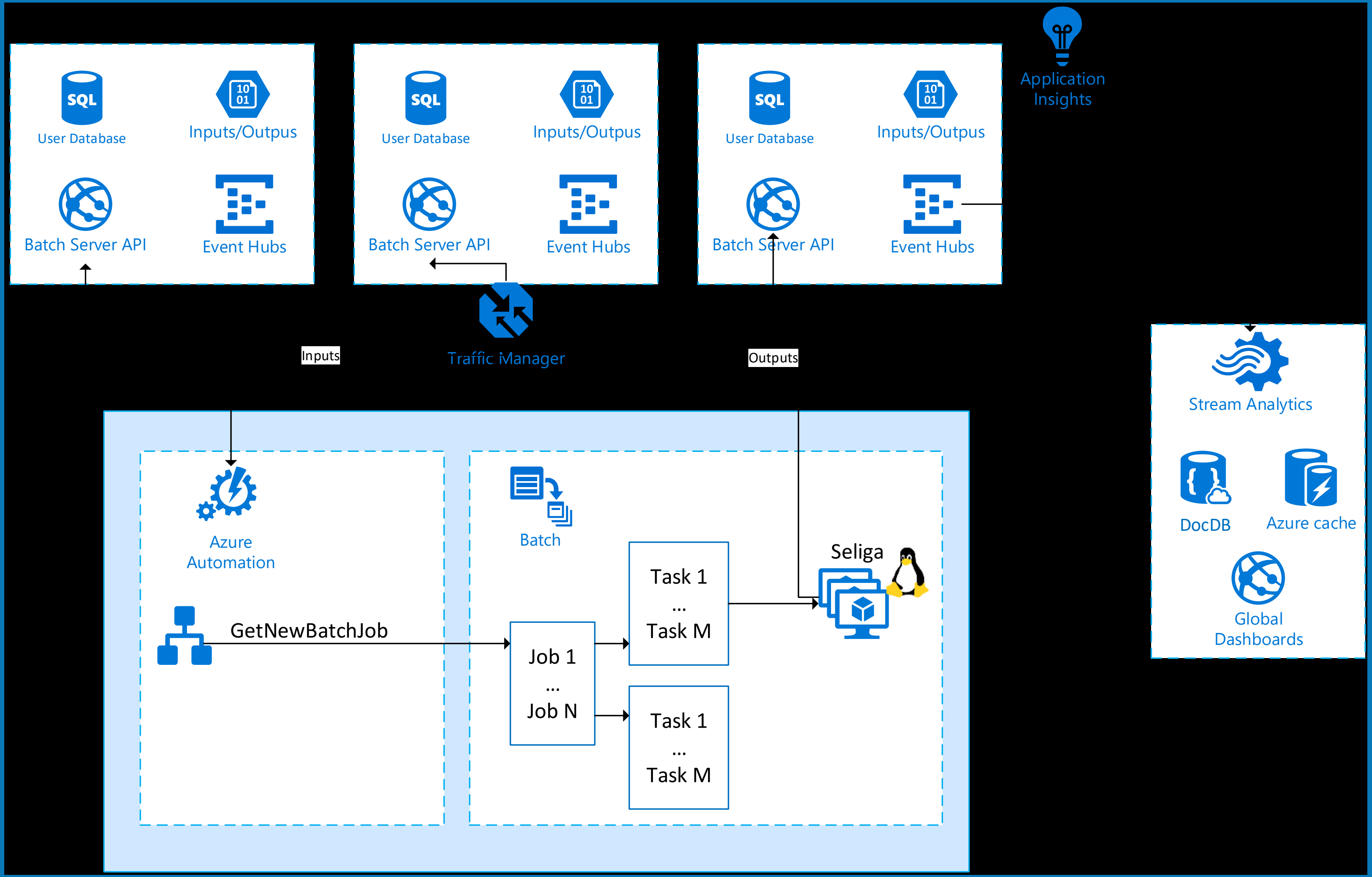
By taking part in the Science Lab, you will be helping to contribute to the body of knowledge in this important field that allows researchers to understand the very beginnings of the universe itself. We hope you choose to deploy the packages that will run the Seliga algorithm, and deliver real results to the researchers.

Science Lab Architecture

To show the power of Azure in the Global Azure Bootcamp Science Lab this year we have created a solution that maximises the use of Azure resources whilst providing the research team with large volumes of computing power.

The Seliga algorithm runs in an Azure Batch process that YOU deploy and scale as you want.

For extra fun, there is a dashboard showing global scores so you can compete against your friends or other people around the world.



Global Azure Bootcamp 2017 Science Lab

About

http://bit.ly/gablab2017about

Instructions

http://bit.ly/gablab2017run

Lab Key: CYW-JYN-BRU

Dashboard

http://gablab2017.azurewebsites.net

2017
Global Azure
BOOTCAMP