

#TESTWORKSCONF #XPIRT



Scaling out your load tests with Visual Studio Team Services

How to avoid the pitfall of success

Marcel de Vries
CTO Xpirit

 @marcelv





Lets sell music online!

Assemble a team

THE LEAN
STARTUP

How Today's Entrepreneurs Use
Continuous Innovation to Create
Radically Successful Businesses

ERIC RIES

We do Scrum

Scrum Process



We read THE book



WEBSITE LAUNCH:
05.10.16

Marketing is doing great!



Website looks great

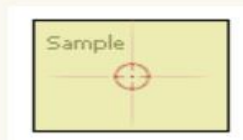


ASP.NET MVC MUSIC STORE Country:51.8558,6.3139

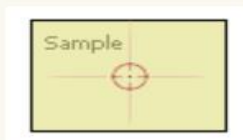
- Rock
- Classical
- Jazz
- Pop
- Disco
- Latin
- Metal
- Alternative
- Reggae
- Blues



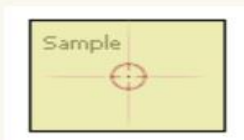
Fresh off the grill



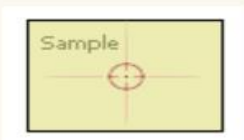
The Best Of Men At Work



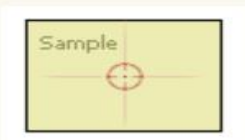
For Those About To Rock We Salute You



Let There Be Rock



Balls to the Wall



Restless and Wild

Launch date

Server Error in '/' Application.

There is a file sharing violation. A different process might be using the file. [D:\home\site\wwwroot\App_Data\MvcMusicStore.sdf]

Description: An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

Exception Details: System.Data.SqlClient.SqlCeException: There is a file sharing violation. A different process might be using the file. [D:\home\site\wwwroot\App_Data\MvcMusicStore.sdf]

Source Error:

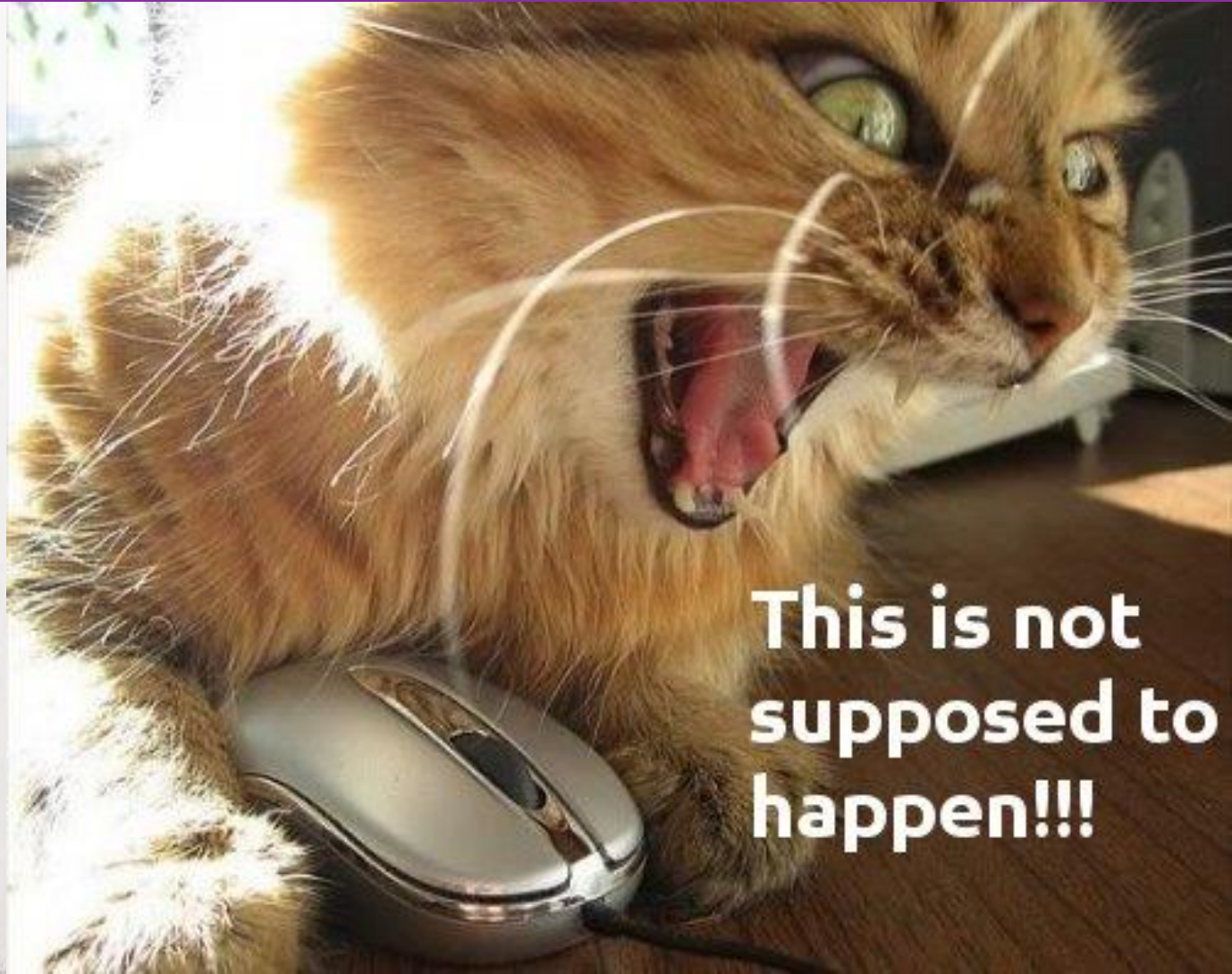
An unhandled exception was generated during the execution of the current web request. Information regarding the origin and location of the exception can be identified using the exception stack trace below.

Stack Trace:

```
[SqlCeException (0x80004005): There is a file sharing violation. A different process might be using the file. [ D:\home\site\wwwroot\App_Data\MvcMusicStore.sdf ]]
  System.Data.SqlClient.SqlCeConnection.Open(Boolean silent) +2165
  System.Data.SqlClient.SqlCeConnection.Open() +47
  System.Data.Entity.Infrastructure.Interception.DbConnectionDispatcher.<Open>b__36(DbConnection t, DbConnectionInterceptionContext c) +10
  System.Data.Entity.Infrastructure.Interception.InternalDispatcher`1.Dispatch(TTarget target, Action`2 operation, TInterceptionContext interceptionContext, Action`3 executing, Action`3 executed) +468
  System.Data.Entity.Infrastructure.Interception.DbConnectionDispatcher.Open(DbConnection connection, DbInterceptionContext interceptionContext) +360
  System.Data.Entity.Core.EntityClient.EntityConnection.<Open>b__2() +55
  System.Data.Entity.Infrastructure.DefaultExecutionStrategy.Execute(Action operation) +9
  System.Data.Entity.Core.EntityClient.EntityConnection.Open() +254

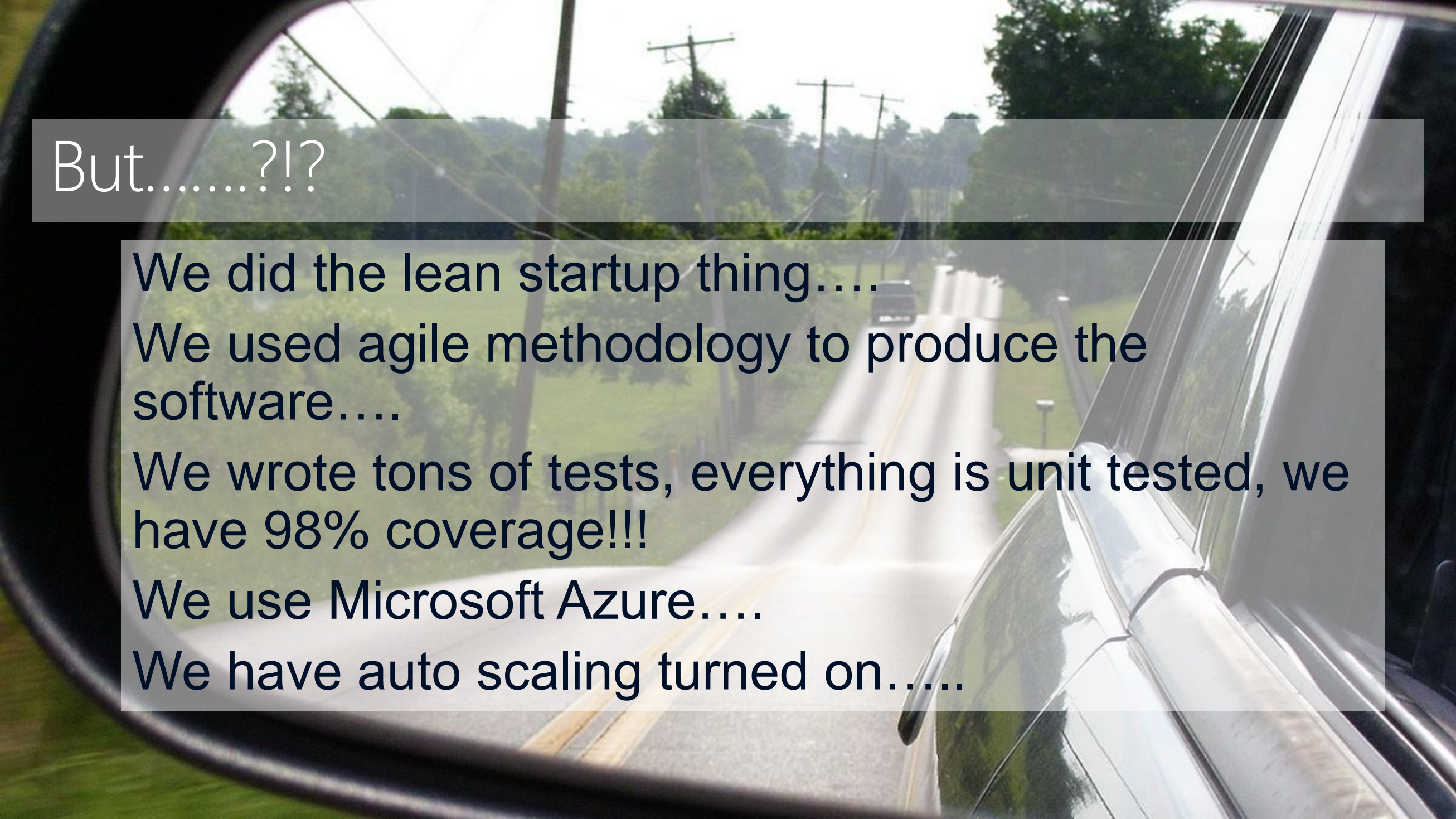
[EntityException: The underlying provider failed on Open.]
  System.Data.Entity.Core.EntityClient.EntityConnection.Open() +324
  System.Data.Entity.Core.Objects.ObjectContext.EnsureConnection(Boolean shouldMonitorTransactions) +133
  System.Data.Entity.Core.Objects.ObjectContext.ExecuteInTransaction(Func`1 func, IDbExecutionStrategy executionStrategy, Boolean startLocalTransaction, Boolean releaseConnectionOnSuccess) +46
  System.Data.Entity.Core.Objects.<>c__DisplayClass3.<GetResults>b__1() +154
  System.Data.Entity.Infrastructure.DefaultExecutionStrategy.Execute(Func`1 operation) +9
  System.Data.Entity.Core.Objects.ObjectQuery`1.GetResults(Nullable`1 forMergeOption) +279
  System.Data.Entity.Core.Objects.ObjectQuery`1.<System.Collections.Generic.IEnumerable<T>.GetEnumerator>b__0() +11
  System.Data.Entity.Internal.LazyEnumerator`1.MoveNext() +45
  System.Linq.Enumerable.Single(IEnumerable`1 source) +192
  System.Data.Entity.Core.Objects.ELinq.ObjectQueryProvider.<GetElementFunction>b__3(IEnumerable`1 sequence) +40
  System.Data.Entity.Core.Objects.ELinq.ObjectQueryProvider.ExecuteSingle(IEnumerable`1 query, Expression queryRoot) +60
  System.Data.Entity.Core.Objects.ELinq.ObjectQueryProvider.System.Linq.IQueryProvider.Execute(Expression expression) +113
  System.Data.Entity.Internal.Linq.DbQueryProvider.Execute(Expression expression) +112
  System.Linq.Queryable.Single(IQueryable`1 source, Expression`1 predicate) +249
  MvcMusicStore.Controllers.ShoppingCartController.AddToCart(Int32 id) in c:\Repos\Demos\MvcMusicStore-Appinsights\MvcMusicStore\Controllers\ShoppingCartController.cs:43
  lambda_method(Closure , ControllerBase , Object[] ) +97
  System.Web.Mvc.ActionMethodDispatcher.Execute(ControllerBase controller, Object[] parameters) +14
```

What just happened?



**This is not
supposed to
happen!!!**





But.....?!?

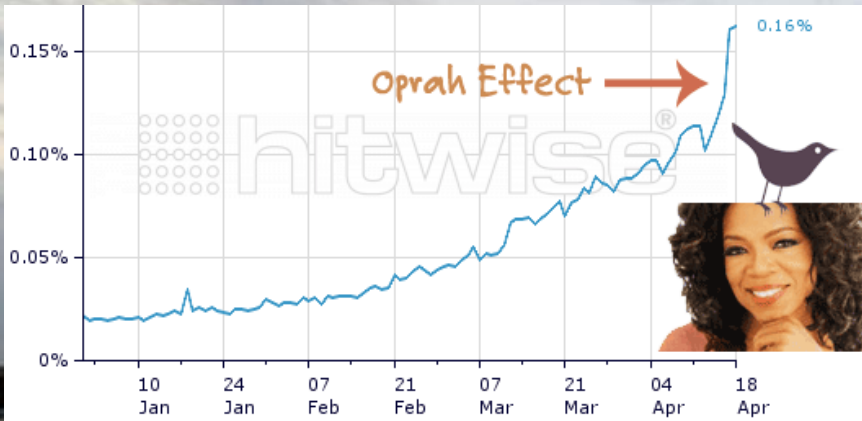
We did the lean startup thing....

We used agile methodology to produce the software....

We wrote tons of tests, everything is unit tested, we have 98% coverage!!!

We use Microsoft Azure....

We have auto scaling turned on.....



WELCOME TO THE PITFALL CALLED SUCCESS!

Applications always behave differently when running under high load and stress conditions!



I never make the same mistake twice. I make it five or six times, just to be sure."

— Anonymous

What could have been done to prevent this?

Different forms of performance testing



1

Performance testing:

How fast will the application code perform?



2

Load testing:

How will the application behave in front of the expected users load?



3

Stress testing:

What is point of users-load that will break your application?



4

Capacity planning:

Will the application be able to scale up to the expected future capacity?

Tools that can help in this space

Variation of Open Source and Commercial software

Some common tools used:

Apache JMeter (Open Source)

Tsung (Open Source)

Web Server Stress Tool (Commercial)

LoadUI (Commercial)

Visual Studio (Commercial)

Performance testing in VS

Visual Studio Ultimate(2010,12,13)

Visual Studio Enterprise 2015

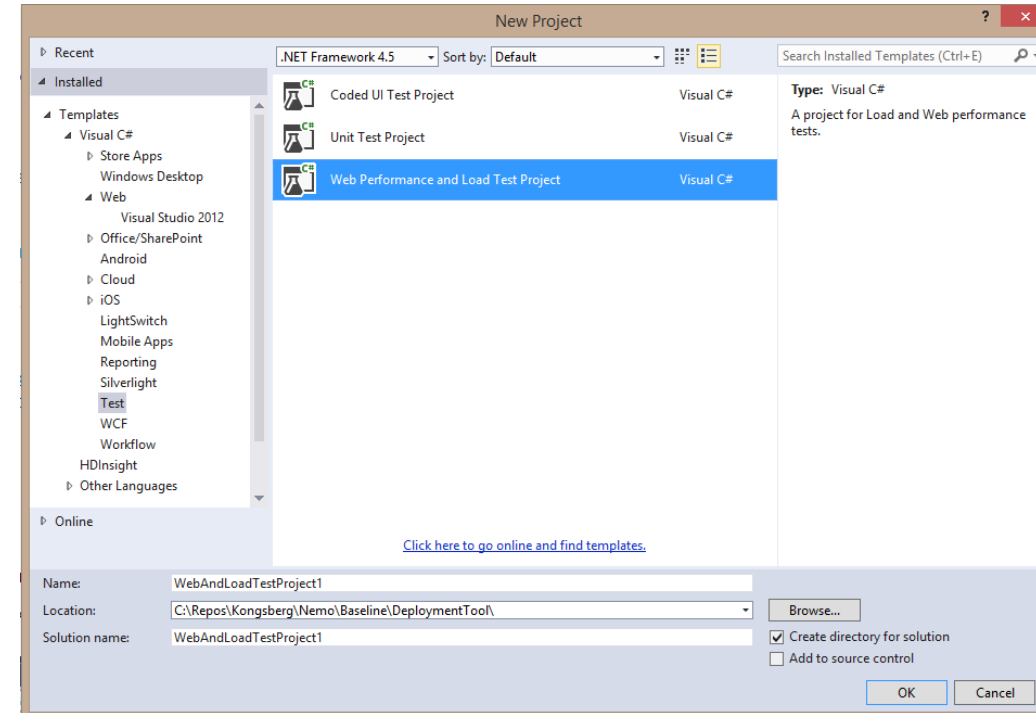
Web Performance Test

Record scenarios

Visit pages or call REST API

Any unit test

E.g. a test calling your WCF/REST services



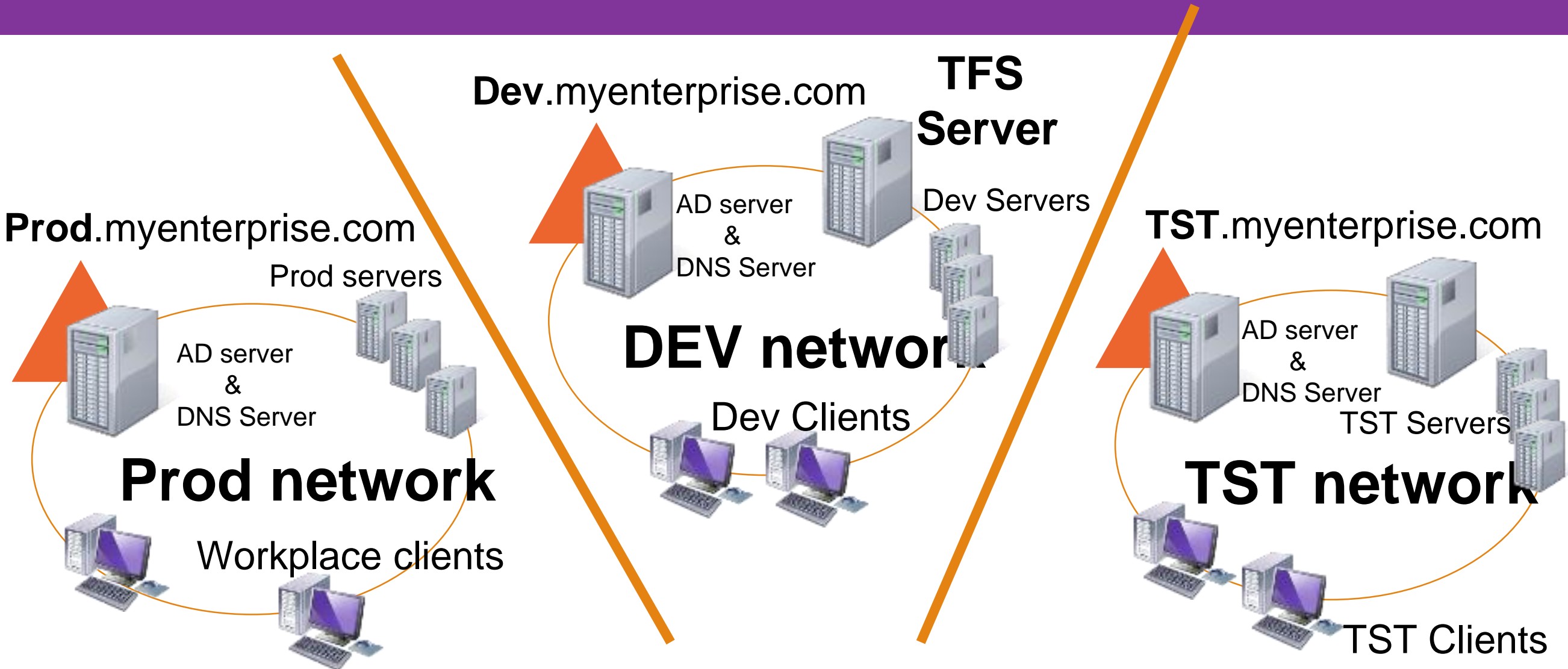
***We have done an amazing job of hiding this diamond
(a 10K+ load testing tool) inside the box for years***

-- Charles Sterling .Net Rocks episode 1202

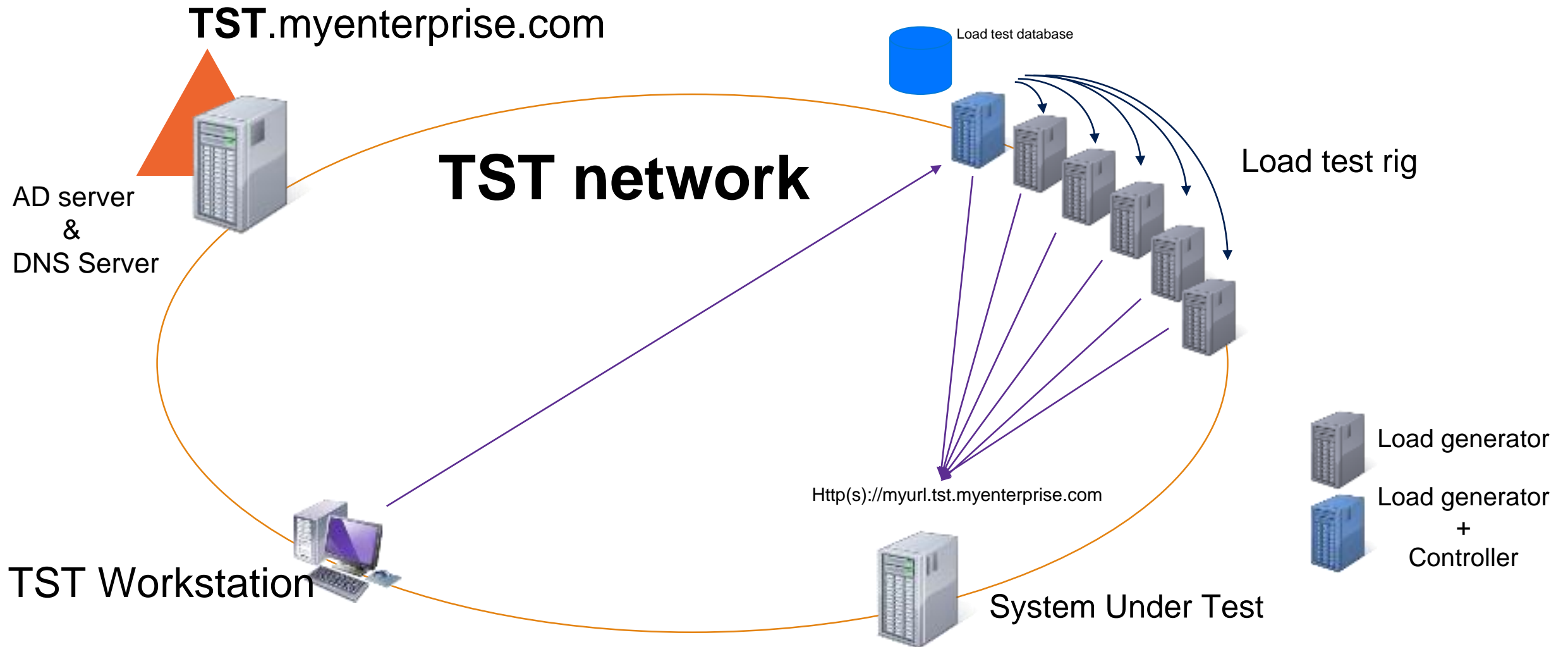
Demo

- Load testing using Visual Studio

Typical Enterprise setup



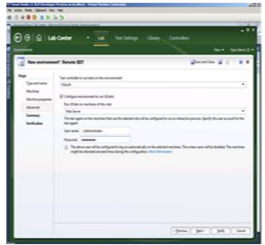
Load test configuration



What is needed to make this work?



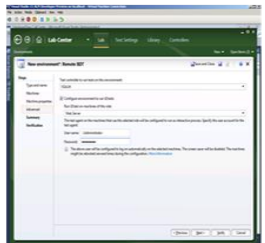
Appropriate/buy/
rent computers



Install your
software/agents



Physically setup
computers



Replicate this to all
the computers

- On premises Load testing can be:
 - Expensive to setup
 - Provisioning is complex
 - Slow to scale
 - Costly to maintain
- Benefit: works on private sites

Benefits of cloud load testing



1

No need to setup your own infrastructure and maintain it.



2

Get infrastructure in cloud when you need it.



3

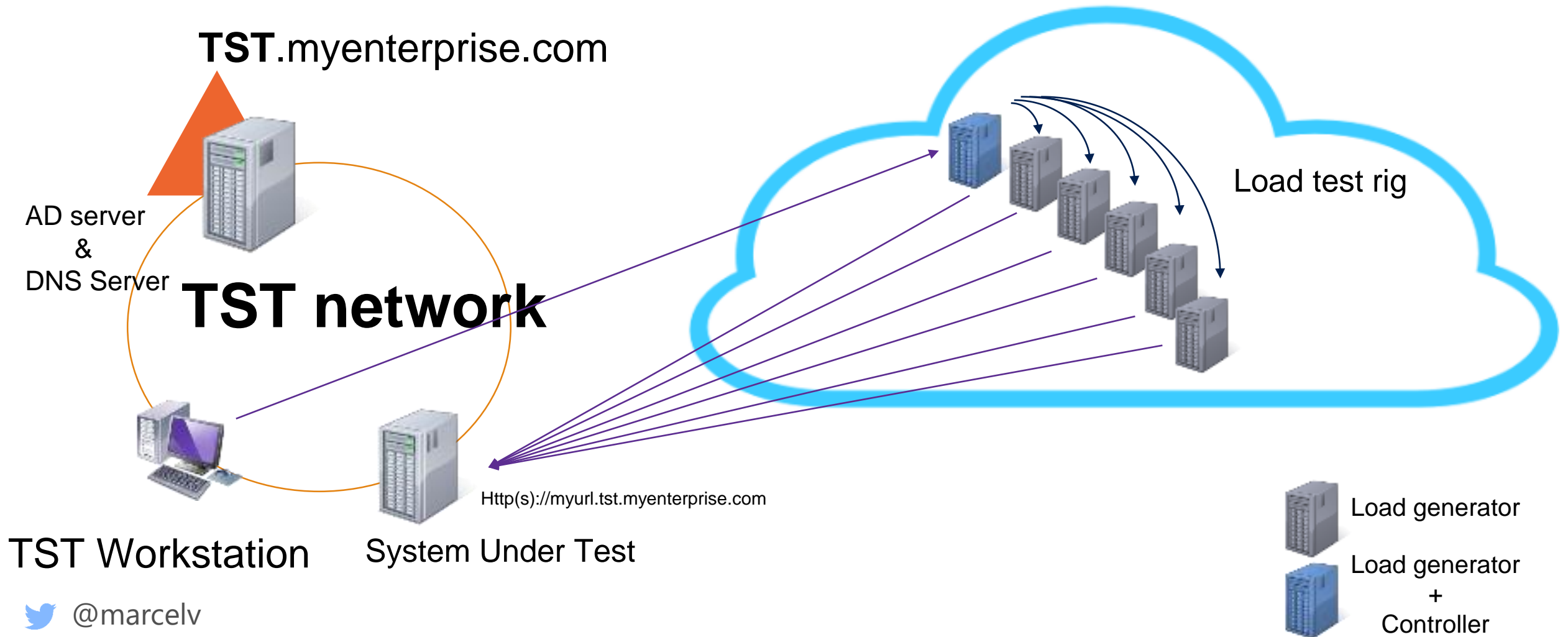
Use the same load test project which you use for on premises today.

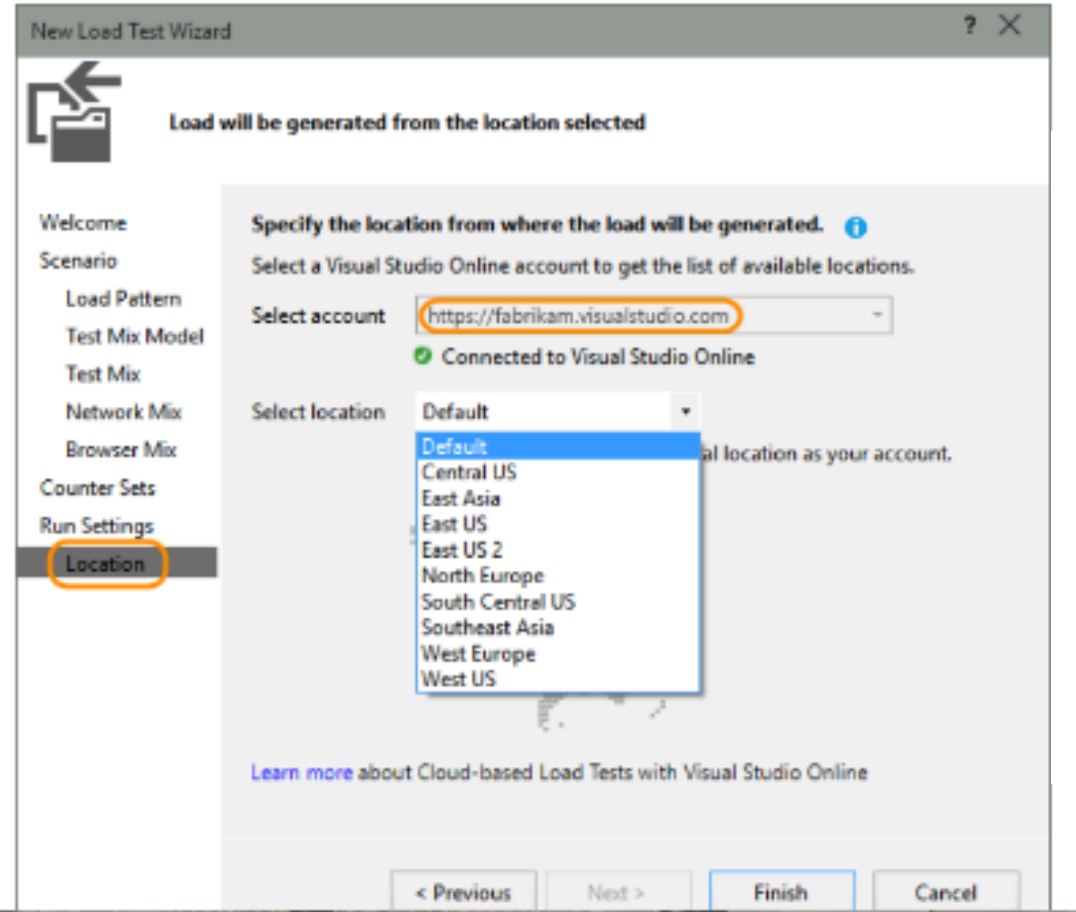


4

Scale out easily in case of doing stress testing.

Cloud Load test configuration





New capabilities in VS 2015 & Update 5
13 regions up to 100 Cores for 72 hrs!

Demo

- Cloud Based Load tests

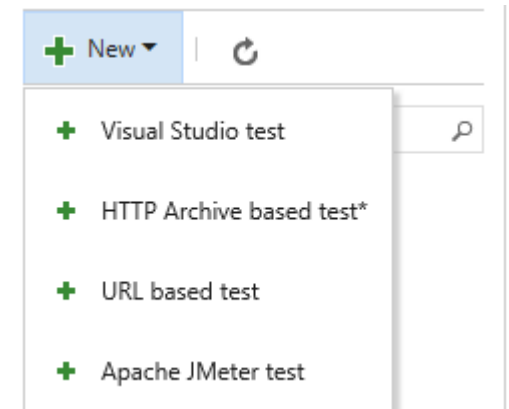
But I don't want Visual Studio!



No worries, don't need it



- You can create web tests with fiddler
 - Free tool
- You can create load tests with Jmeter
 - Http Sampler only
- VSTS supports out of the box:



Demo

- Cloud Load tests using Fiddler

Load testing Approach

Need to create representative scenarios

Need to provide a real life scenario mix

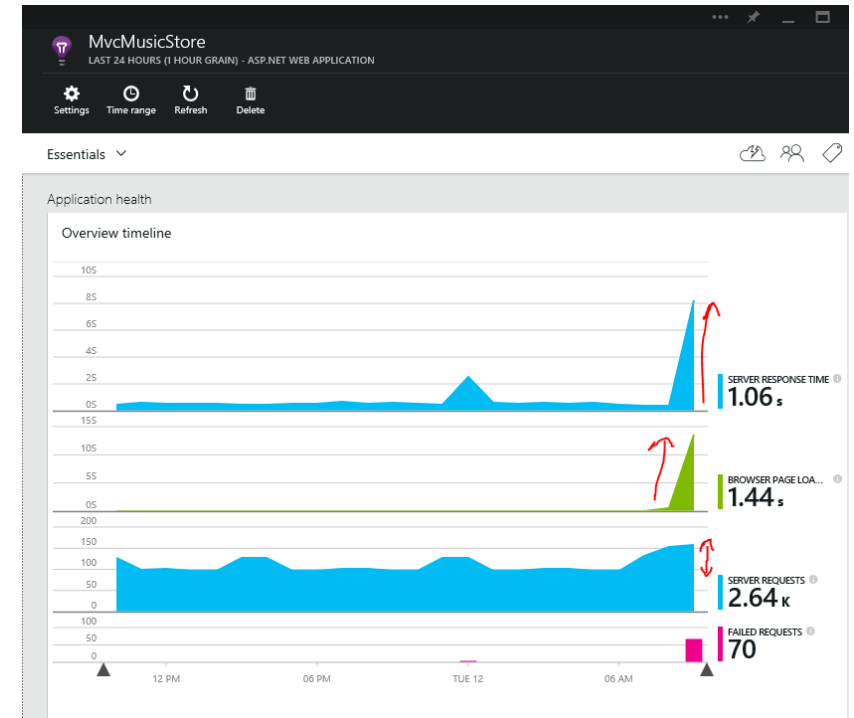
Not everyone who browses a site will buy the product

Simulate actual conversion

Start with a baseline

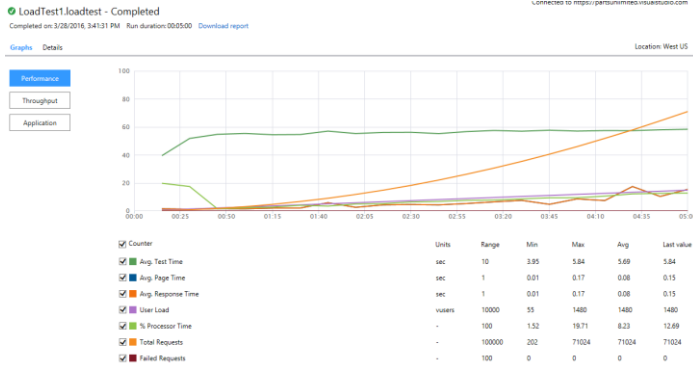
Always compare to the baseline and check improvements

Make it part of your daily routine

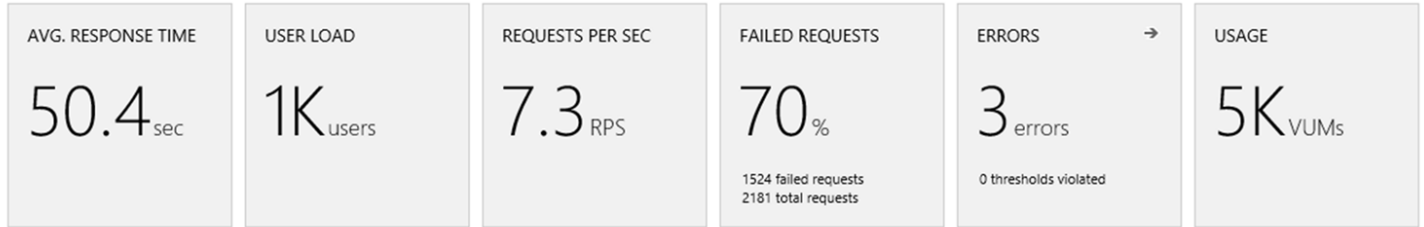


“It is not hard to break something with a load test but it is extremely hard to learn something when you do”

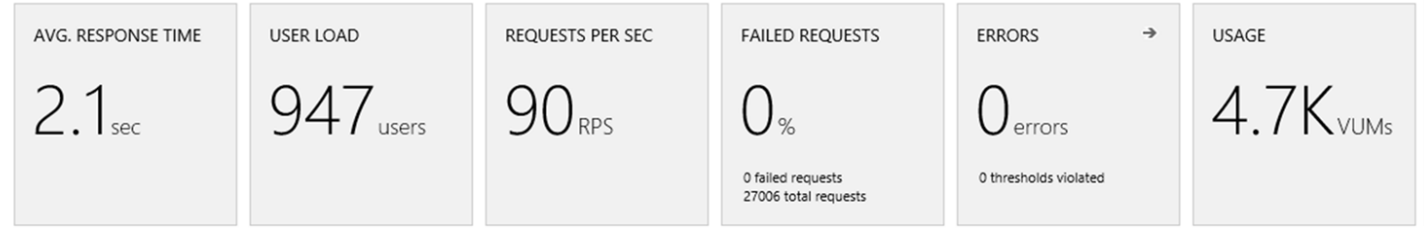
-- Richard Campbell, .Net Rocks episode 1202



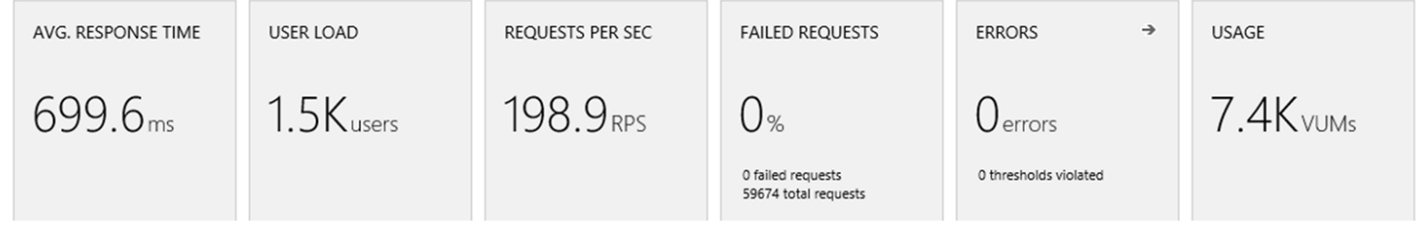
Baseline



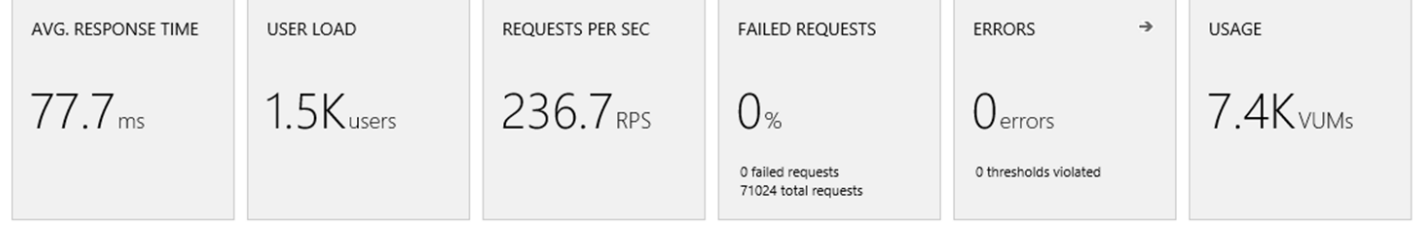
Iteration 1



Iteration 2



Iteration 3



Example Node.JS & Mongo optimization done in April

Load testing part of build & Release

VSTS standard build engine

Can build any product on any platform

Out of the box support for Java, iOS, Android, .NET and much more

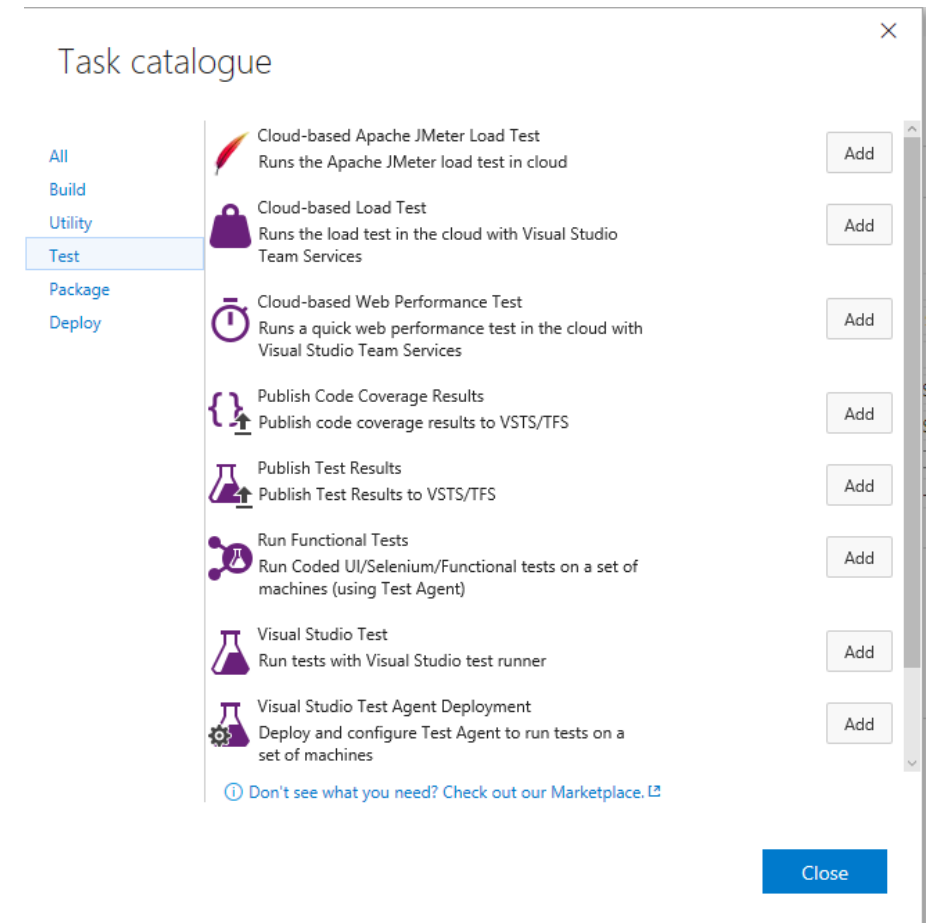
Out of the box support for Load tests

VSTS Release pipelines

Uses same infrastructure as build

So out of the box also support for Load Tests as part of your delivery pipeline

All fully automated!



The screenshot shows the 'Task catalogue' window in VSTS. On the left, there is a navigation menu with categories: All, Build, Utility, Test (highlighted), Package, and Deploy. The main area lists several tasks, each with an icon, a title, a description, and an 'Add' button:

- Cloud-based Apache JMeter Load Test**: Runs the Apache JMeter load test in cloud.
- Cloud-based Load Test**: Runs the load test in the cloud with Visual Studio Team Services.
- Cloud-based Web Performance Test**: Runs a quick web performance test in the cloud with Visual Studio Team Services.
- Publish Code Coverage Results**: Publish code coverage results to VSTS/TFS.
- Publish Test Results**: Publish Test Results to VSTS/TFS.
- Run Functional Tests**: Run Coded UI/Selenium/Functional tests on a set of machines (using Test Agent).
- Visual Studio Test**: Run tests with Visual Studio test runner.
- Visual Studio Test Agent Deployment**: Deploy and configure Test Agent to run tests on a set of machines.

At the bottom of the catalogue, there is a link: "Don't see what you need? Check out our Marketplace." and a "Close" button.

Demo

- Integration in release pipeline

But what does this cost?

- First we need to define a VUM..
 - Virtual User Minute
 - One minute of load testing of one Virtual user
 - Constant load 25 users 3 minutes
 - $25 * 3 = 75$ VUM

First 20,000 VUM / month	20,001-2M VUM	2,000,001-10M VUM	10,000,001 VUM-above
FREE	\$0.0004 / VUM	\$0.0002 / VUM	\$0.0001 / VUM

- E.g run a test for 1 hour to simulate 10,000 users =>
 - $60 * 10,000 => 600,000 * 0.0004 = \240

Conclusion

Cloud based load testing enables easy road to load testing for your external facing sites / API's

Visual Studio supports simple to set up load tests

Visual Studio Team Services(VSTS) provides cloud based load tests

supports various formats including JMeter

Make load testing part of you automated deployment strategy!



@marcelv



<http://fluentbytes.com>

mdevries@xpirit.com

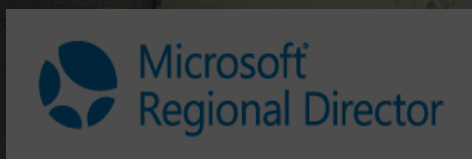
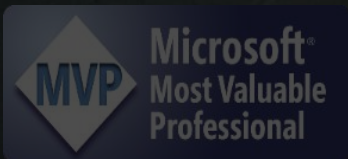




xpir.it/XpiritMagazine



@marcelv



<http://fluentbytes.com>

mdevries@xpirit.com

