Windows Azure

Python and Windows Azure

Wenming Ye
Sr. Program Manager
Microsoft Research

Twitter: @wenmingye

Cloud Computing

IaaS

Infrastructure-as-a-Service

host

PaaS

Platform-as-a-Service

build

SaaS

Software-as-a-Service

consume

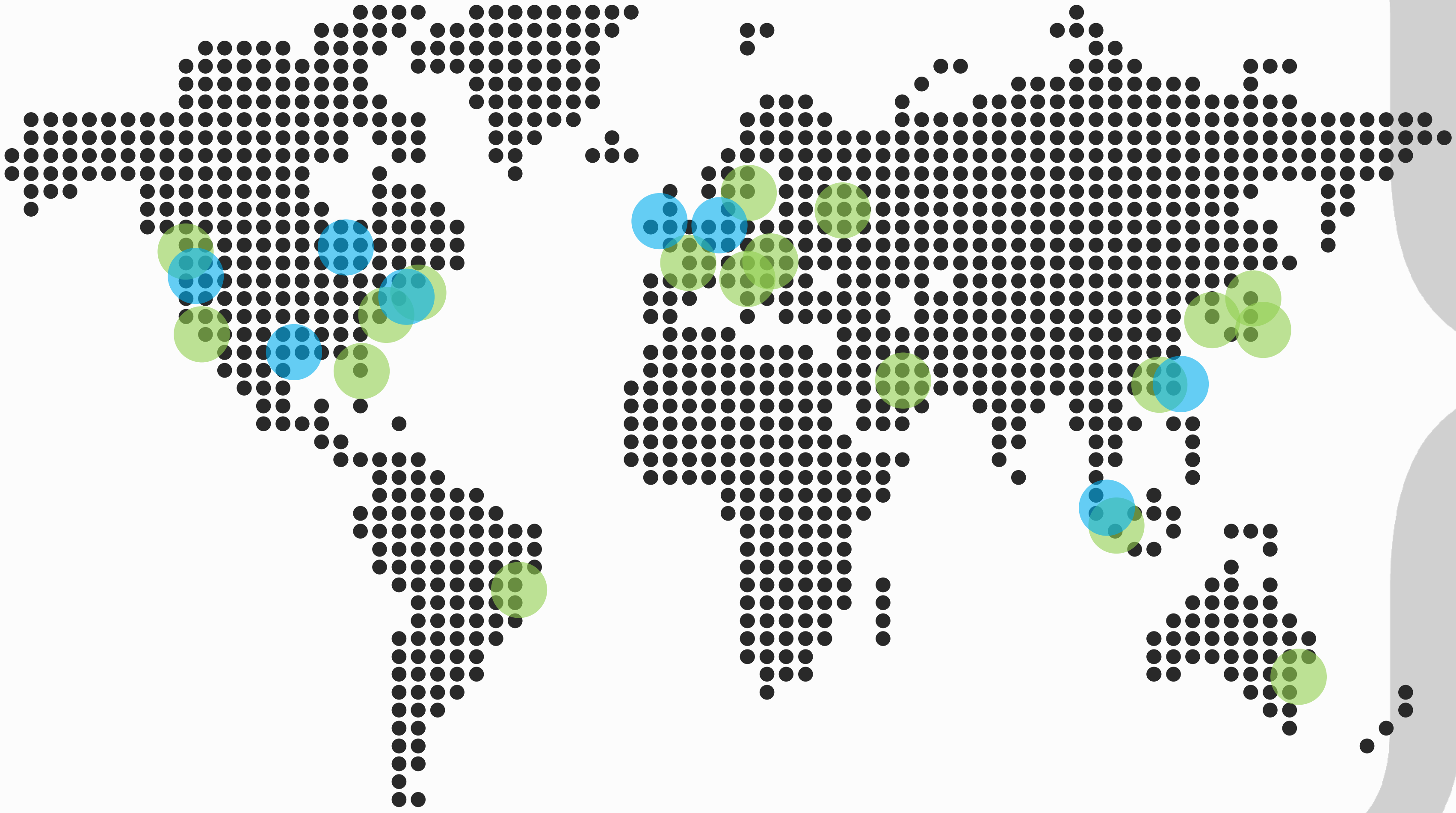
Windows Azure

Comprehensive set of services that enable you to quickly build, deploy and manage applications across a global network of Microsoft-managed datacenters

Flexible

<3 Open

Solid



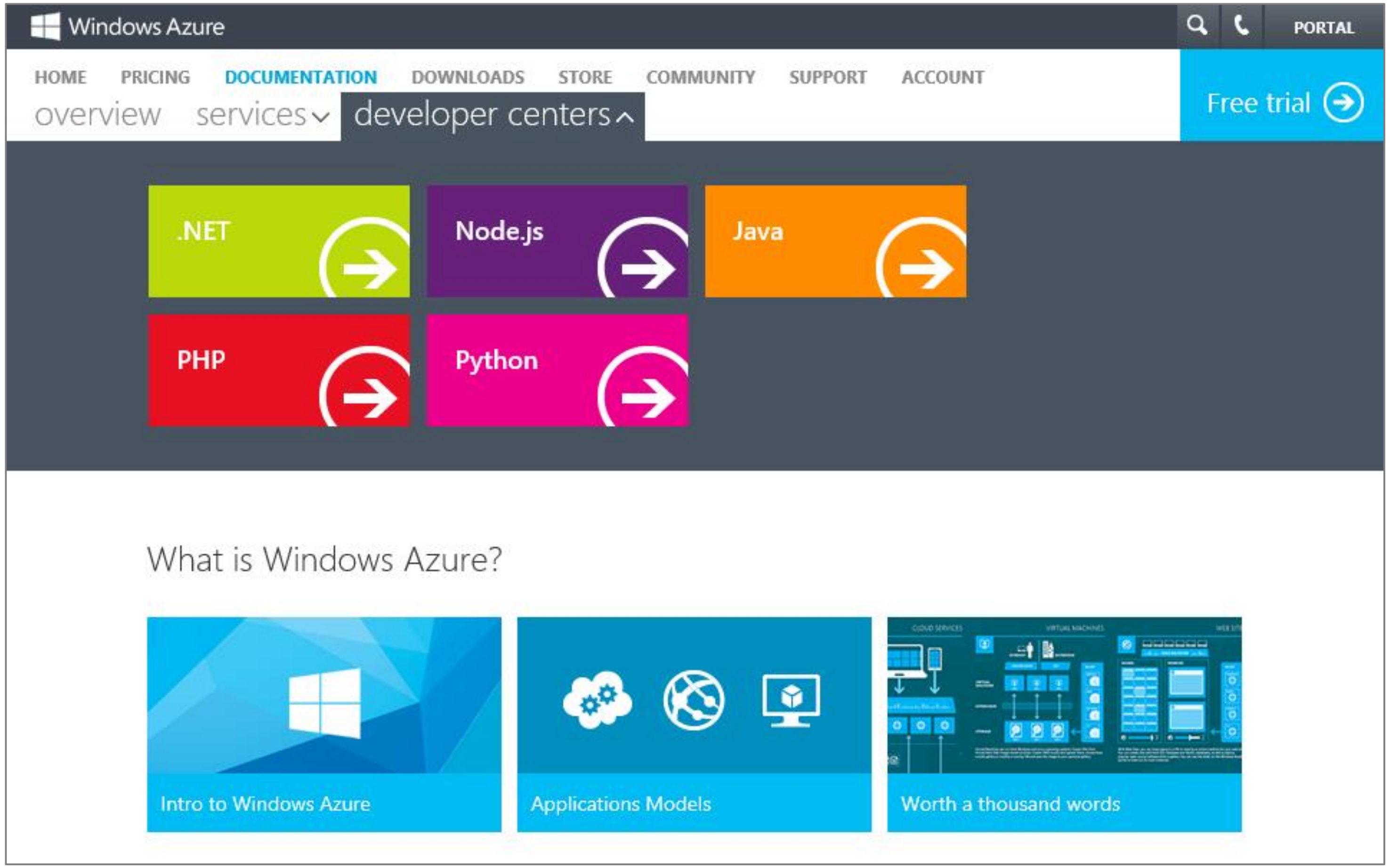
Global Footprint

Windows Azure

Virtual machines

Cloud services

Web sites



Multiple languages

→ http://WindowsAzure.com

PTVS: Python Tools for Visual Studio

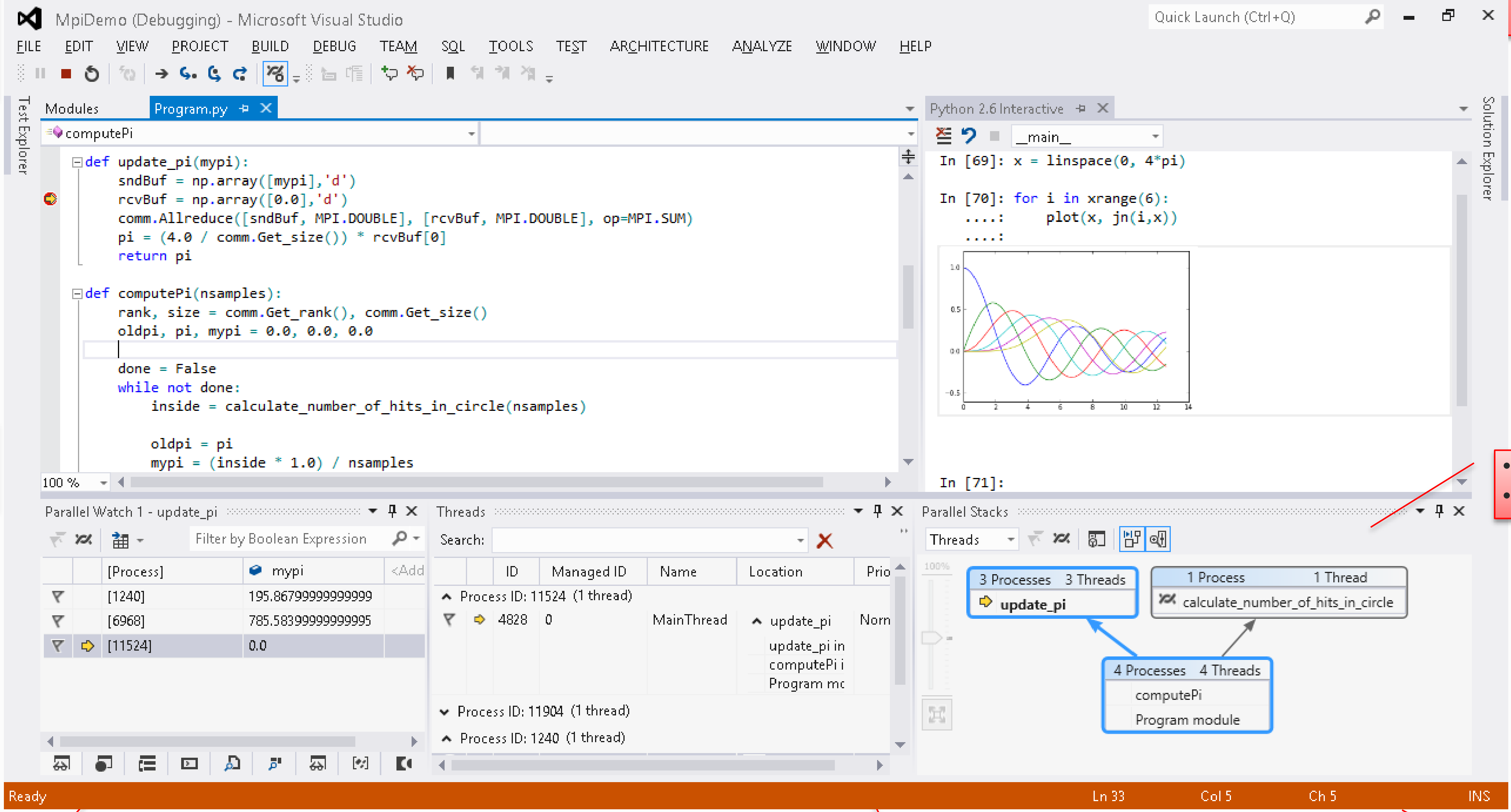
Windows Azure™

Python Tools for VS:

- Project mgmt
- Solutions
- Deployment

- Built-in REPL
- IPython REPL
- Interactive Parallel Computing
- Inline graphics

- Syntax hi-lite
- Intellisense
- Find / Browse



```
def update_pi(mypi):
    sndBuf = np.array([mypi],'d')
    rcvBuf = np.array([0.0],'d')
    comm.Allreduce([sndBuf, MPI.DOUBLE], [rcvBuf, MPI.DOUBLE], op=MPI.SUM)
    pi = (4.0 / comm.Get_size()) * rcvBuf[0]
    return pi

def computePi(nsamples):
    rank, size = comm.Get_rank(), comm.Get_size()
    oldpi, pi, mypi = 0.0, 0.0, 0.0

    done = False
    while not done:
        inside = calculate_number_of_hits_in_circle(nsamples)

        oldpi = pi
        mypi = (inside * 1.0) / nsamples
```

```
In [69]: x = linspace(0, 4*pi)

In [70]: for i in xrange(6):
   ....:     plot(x, jn(i,x))
   ....:

In [71]:
```

[Process]	mypi	<Add
[1240]	195.86799999999999	
[6968]	785.58399999999995	
[11524]	0.0	

ID	Managed ID	Name	Location	Prio
Process ID: 11524 (1 thread)				
4828	0	MainThread	update_pi update_pi in computePi i Program mc	Norn
Process ID: 11904 (1 thread)				
Process ID: 1240 (1 thread)				

- Profiler
- A/B comparions

- Parallel Watch view
- Parallel Stack view

- CPython
- IronPython
- Or any interpreter

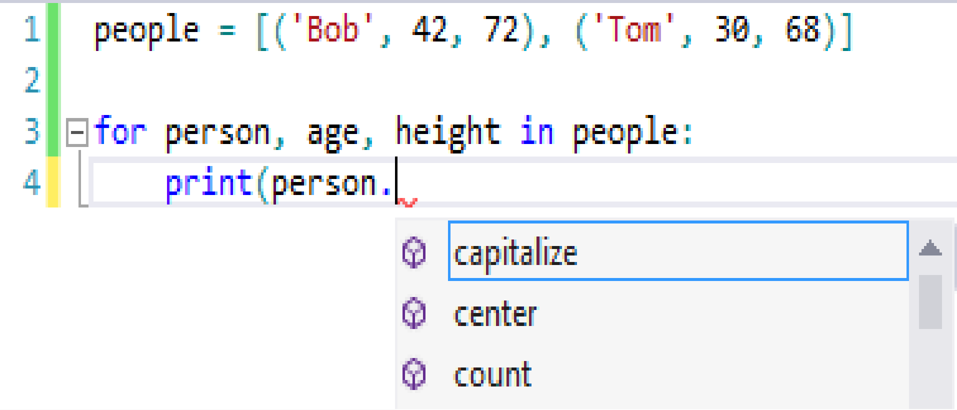
- Python Debugger
- .Net Debugger
- Remote Debugging

- HPC Support
- F5 MPI debugging
- Batch or Interactive

Typical Scenarios

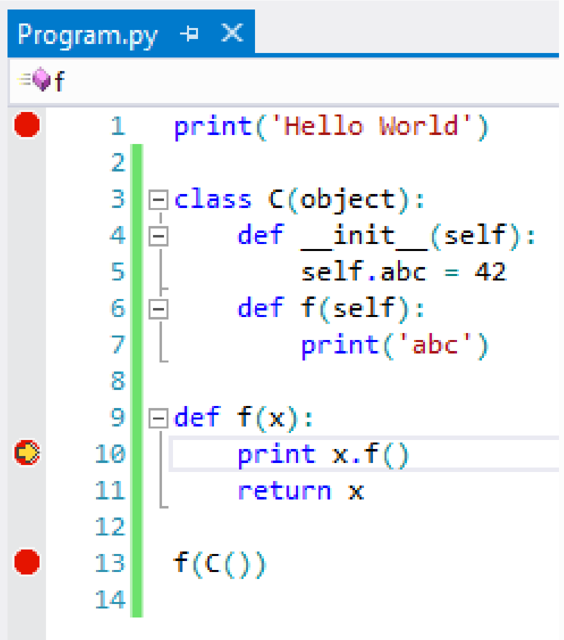
- General purpose programming
  - Projects
  - Edit/Intellisense/Browse
  - Debug
  - Profile
- Cloud / Web
  - Azure Cloud Service
  - Azure Web Sites
  - Django, WFastCGI included
  - IPython notebook
- Technical / Scientific Computing
  - HPC / MPI
  - Cluster Debugging
  - Inline REPL graphics
  - IPython REPL & notebook

PTVS Features



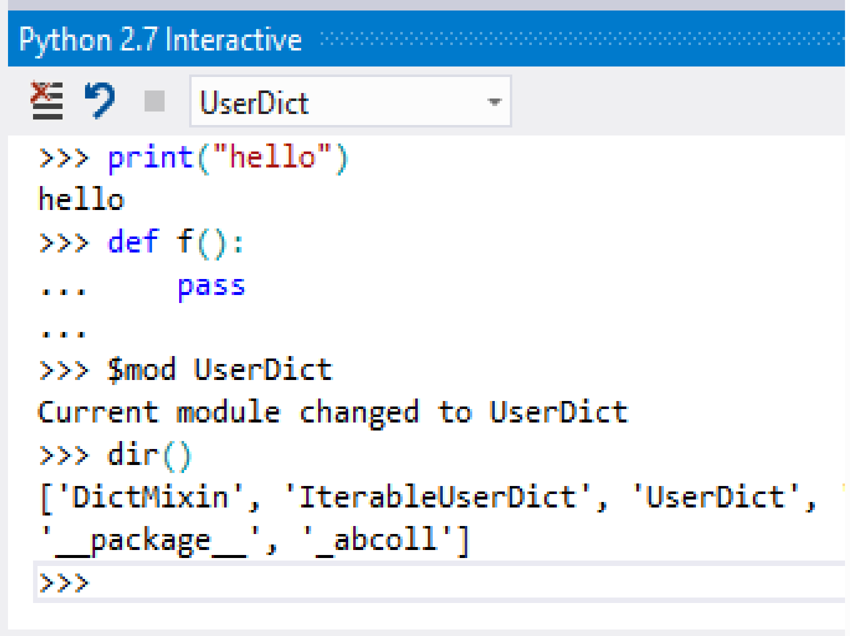
```
people = [('Bob', 42, 72), ('Tom', 30, 68)]

for person, age, height in people:
    print(person.
```

Intellisense



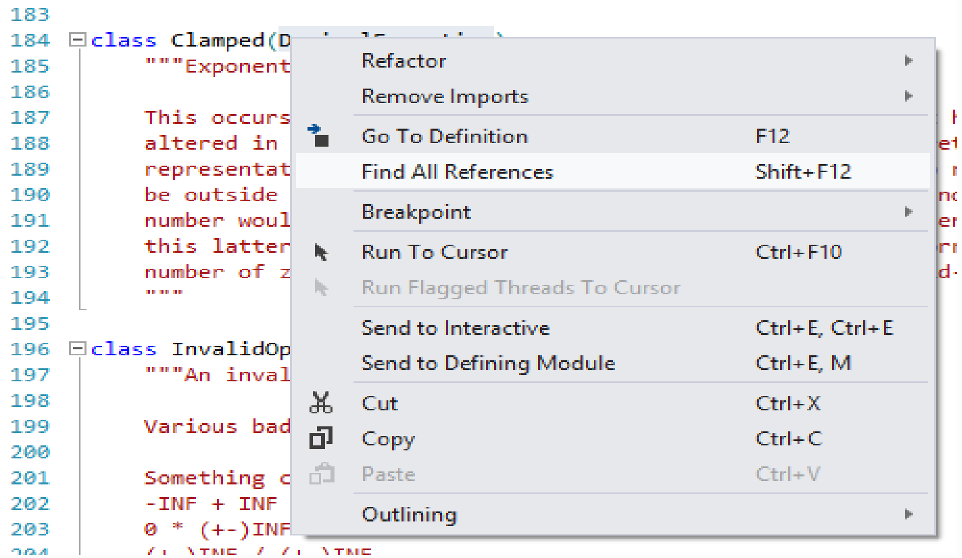
```
print('Hello World')

class C(object):
    def __init__(self):
        self.abc = 42
    def f(self):
        print('abc')

def f(x):
    print x.f()
    return x

f(C())
```

Debugging



```
>>> print("hello")
hello
>>> def f():
...     pass
...
>>> $mod UserDict
Current module changed to UserDict
>>> dir()
['DictMixin', 'IterableUserDict', 'UserDict',
'__package__', '_abcoll']
>>>
```

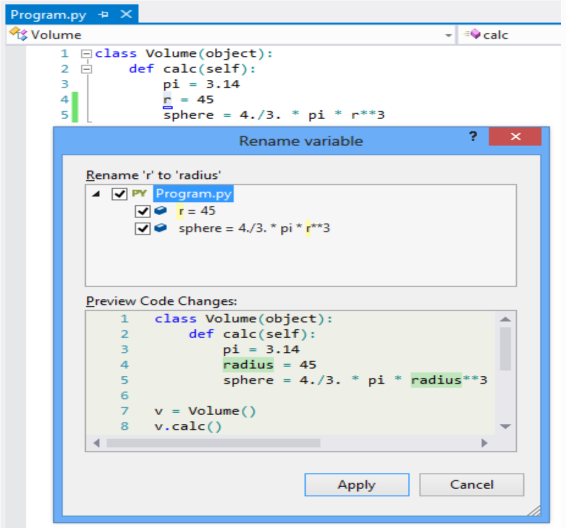
REPL – IPython too



```
class Clamped(D
    """Exponent

    This occurs
    altered in
    representat
    be outside
    number woul
    this latter
    number of z
    """

class InvalidOp
    """An inval

    Various bad

    Something c
    -INF + INF
    0 * (+-)INF
```

Refactor
Remove Imports
Go To Definition F12
Find All References Shift+F12
Breakpoint
Run To Cursor Ctrl+F10
Run Flagged Threads To Cursor
Send to Interactive Ctrl+E, Ctrl+E
Send to Defining Module Ctrl+E, M
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Outlining

Find All Refs



```
class Volume(object):
    def calc(self):
        pi = 3.14
        r = 45
        sphere = 4./3. * pi * r**3
```

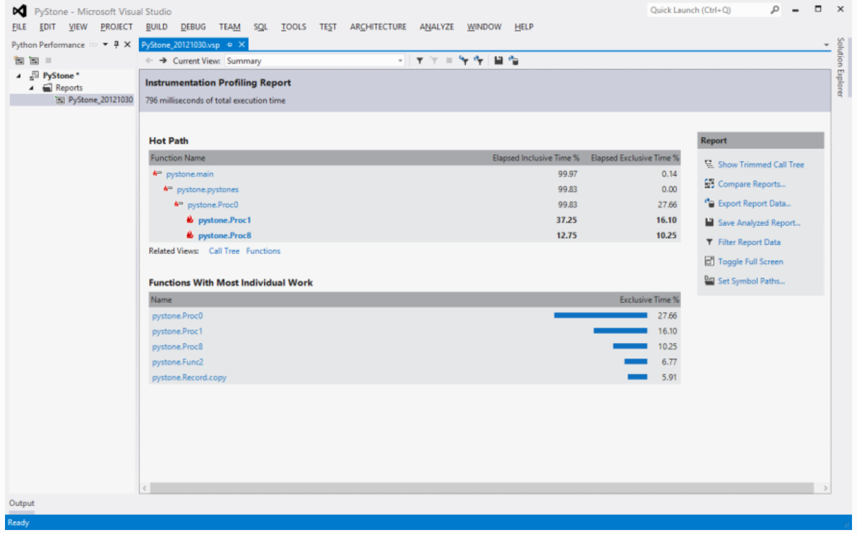
Rename 'r' to 'radius'

- ☑ Program.py
  - ☑ r = 45
  - ☑ sphere = 4./3. * pi * r**3

Preview Code Changes:

```
class Volume(object):
    def calc(self):
        pi = 3.14
        radius = 45
        sphere = 4./3. * pi * radius**3

v = Volume()
v.calc()
```

Refactor/Rename



Instrumentation Profiling Report

796 milliseconds of total execution time

Hot Path

Function Name	Elapsed Inclusive Time %	Elapsed Exclusive Time %
pystone.main	99.97	0.14
pystone.pystones	99.83	0.00
pystone.Proc0	99.83	27.66
pystone.Proc1	37.25	16.10
pystone.Proc8	12.75	10.25

Related Views: Call Tree Functions

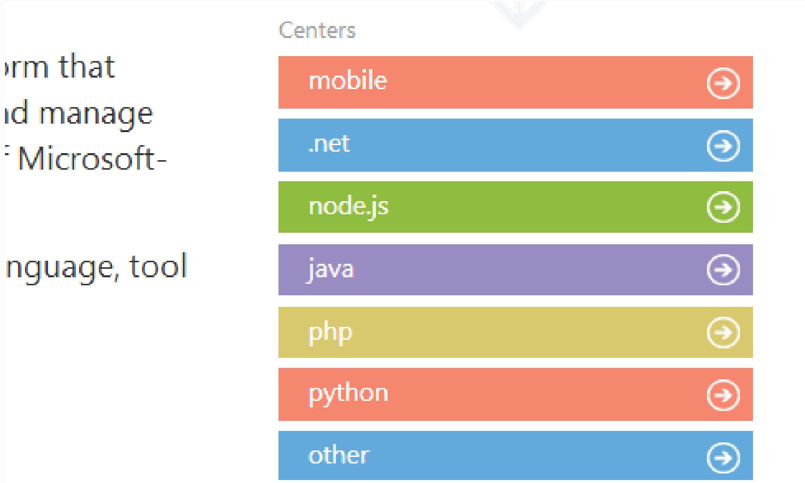
Functions With Most Individual Work

Name	Exclusive Time %
pystone.Proc0	27.66
pystone.Proc1	16.10
pystone.Proc8	10.25
pystone.Func2	6.77
pystone.Record.copy	5.91

Profiling

Windows Azure

Cloud / Web



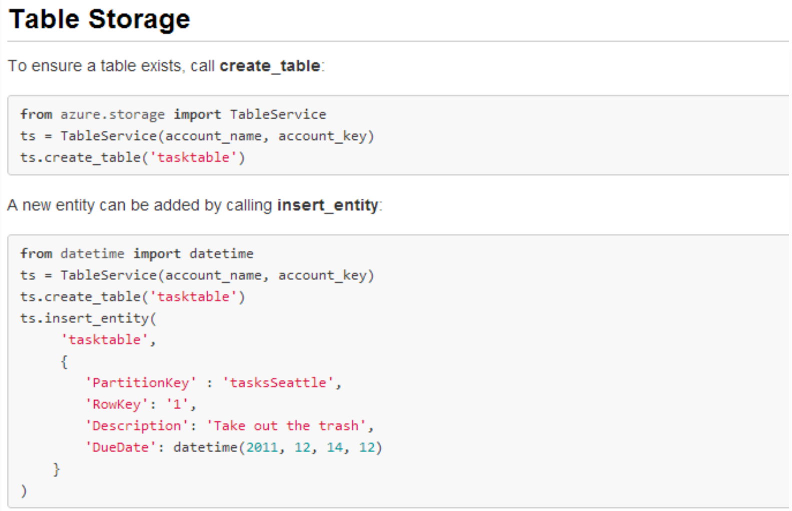


Table Storage

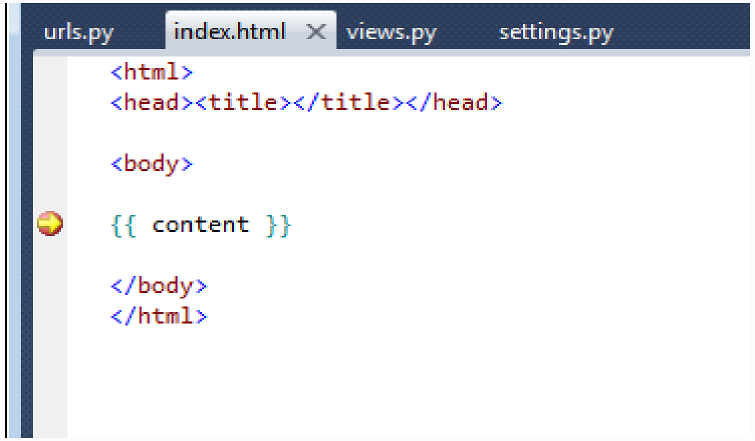
To ensure a table exists, call **create_table**:

```
from azure.storage import TableService
ts = TableService(account_name, account_key)
ts.create_table('tasktable')
```

A new entity can be added by calling **insert_entity**:

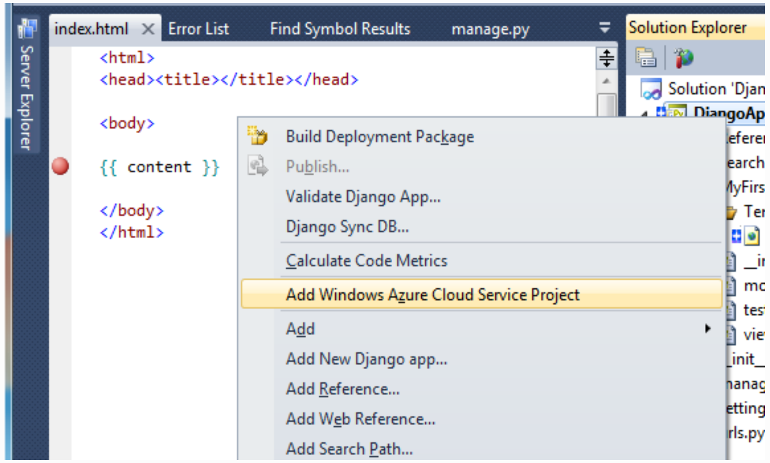
```
from datetime import datetime
ts = TableService(account_name, account_key)
ts.create_table('tasktable')
ts.insert_entity(
     'tasktable',
     {
        'PartitionKey' : 'tasksSeattle',
        'RowKey': '1',
        'Description': 'Take out the trash',
        'DueDate': datetime(2011, 12, 14, 12)
    }
)
```

Azure Python Client Libs for Windows, MacOS, Linux



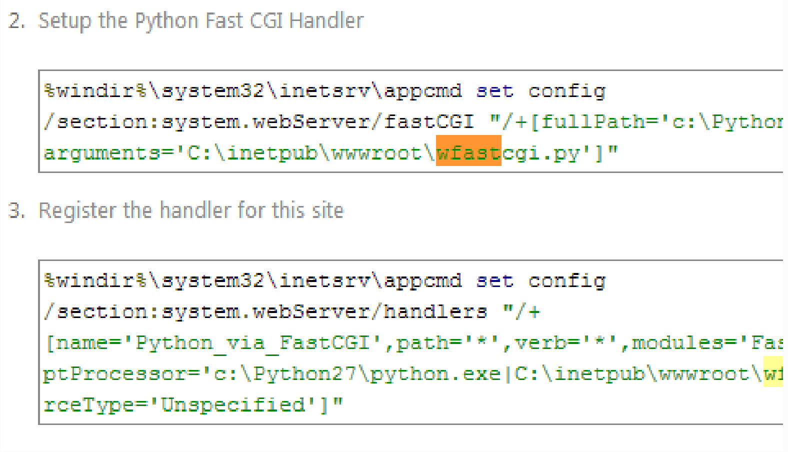
```
<html>
<head><title></title></head>

<body>

{{ content }}

</body>
</html>
```

Django template debug



```
<html>
<head><title></title></head>

<body>

{{ content }}

</body>
</html>
```

Django Deploy



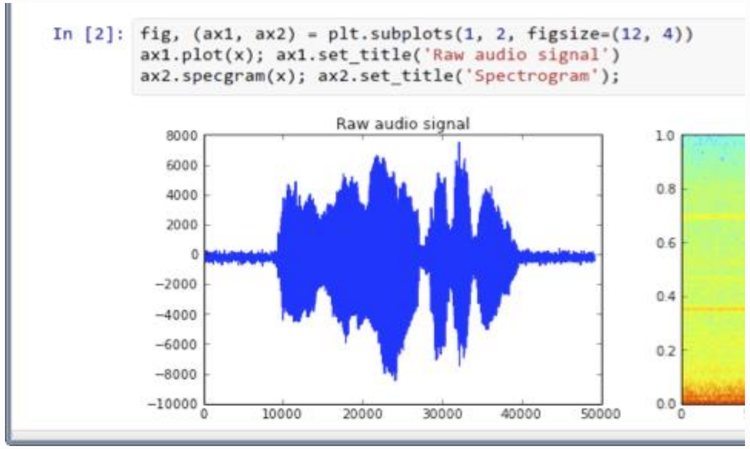
2. Setup the Python Fast CGI Handler

```
%windir%\system32\inetsrv\appcmd set config
/section:system.webServer/fastCGI "/+[fullPath='c:\Pythor
arguments='C:\inetpub\wwwroot\wfastcgi.py']"
```

3. Register the handler for this site

```
%windir%\system32\inetsrv\appcmd set config
/section:system.webServer/handlers "/+
[name='Python_via_FastCGI',path='*',verb='*',modules='Fas
ptProcessor='c:\Python27\python.exe|C:\inetpub\wwwroot\wf
rceType='Unspecified']"
```

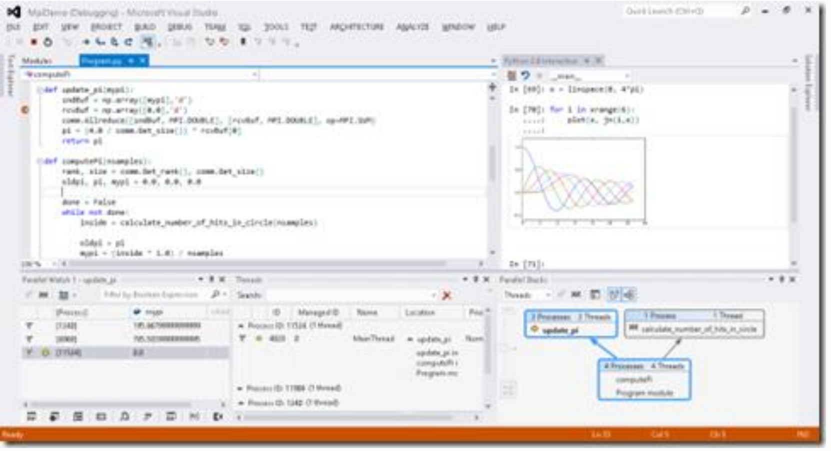
WFastCGI for use with Django, Flask, Bottle, Web2Py, …



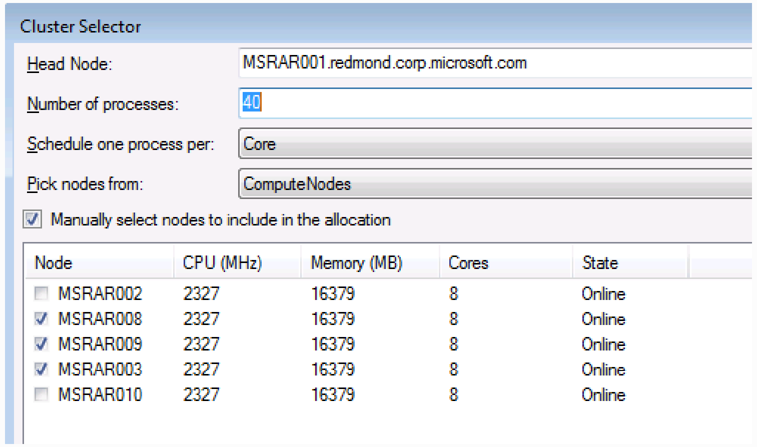
```
In [2]: fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 4))
        ax1.plot(x); ax1.set_title('Raw audio signal')
        ax2.specgram(x); ax2.set_title('Spectrogram');
```

IPython Notebook: Python in the browser

Technical / Scientific Computing



PTVS + Numpy + SciPy is a productive T.C. Workbench



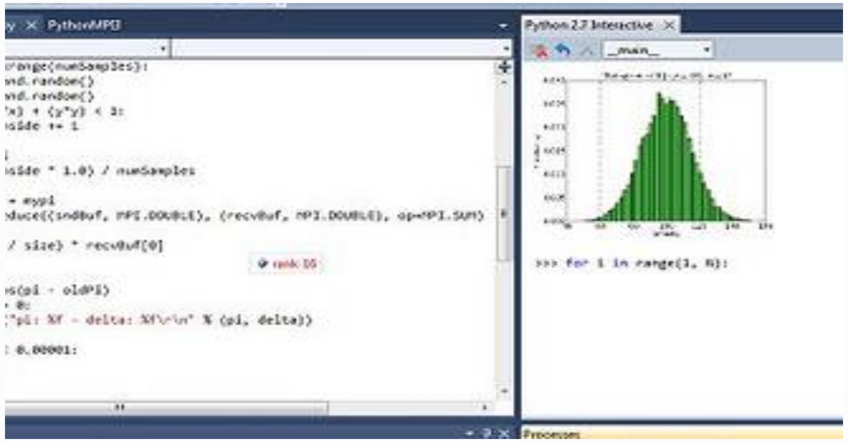
Cluster Selector

Head Node:	MSRAR001.redmond.corp.microsoft.com
Number of processes:	40
Schedule one process per:	Core
Pick nodes from:	ComputeNodes

☑ Manually select nodes to include in the allocation

Node	CPU (MHz)	Memory (MB)	Cores	State
☐ MSRAR002	2327	16379	8	Online
☑ MSRAR008	2327	16379	8	Online
☑ MSRAR009	2327	16379	8	Online
☑ MSRAR003	2327	16379	8	Online
☐ MSRAR010	2327	16379	8	Online

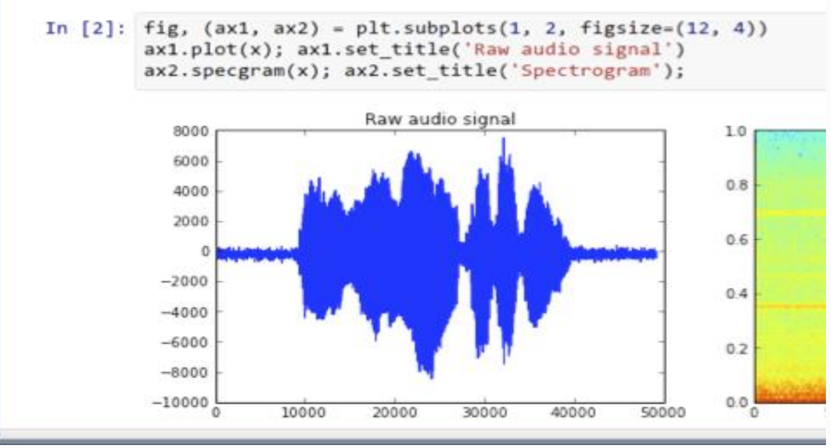
HPC / MPI w support for cluster debugging



```
range(numSamples):
nd.random()
nd.random()
*x) + (y*y) < 1:
iside += 1

iside * 1.0) / numSamples

= mypi
duce((sndBuf, MPI.DOUBLE), (recvBuf, MPI.DOUBLE), op=MPI.SUM)

/ size) * recvBuf[0]

s(pi - oldPI)
> 0:
("pi: %f - delta: %f\r\n" % (pi, delta))

: 0.00001:
```

```
>>> for i in range(1, 8):
```

Inline Graphics in REPL

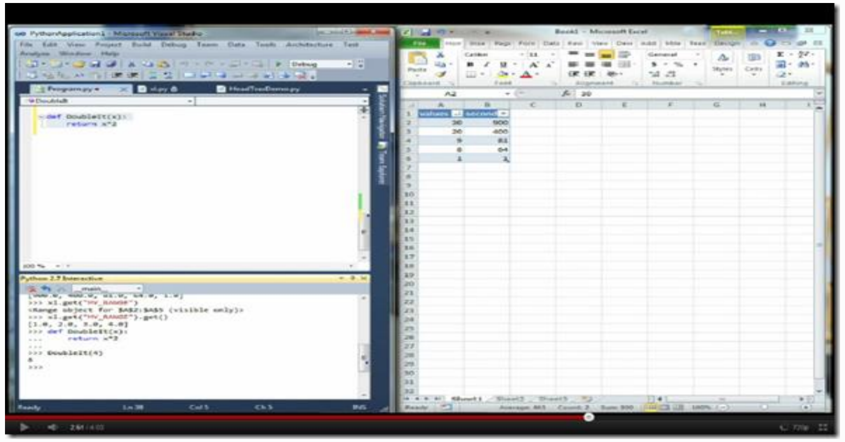


```
In [2]: fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 4))
        ax1.plot(x); ax1.set_title('Raw audio signal')
        ax2.specgram(x); ax2.set_title('Spectrogram');
```

IPython notebook: Python in browser: any OS/ any browser

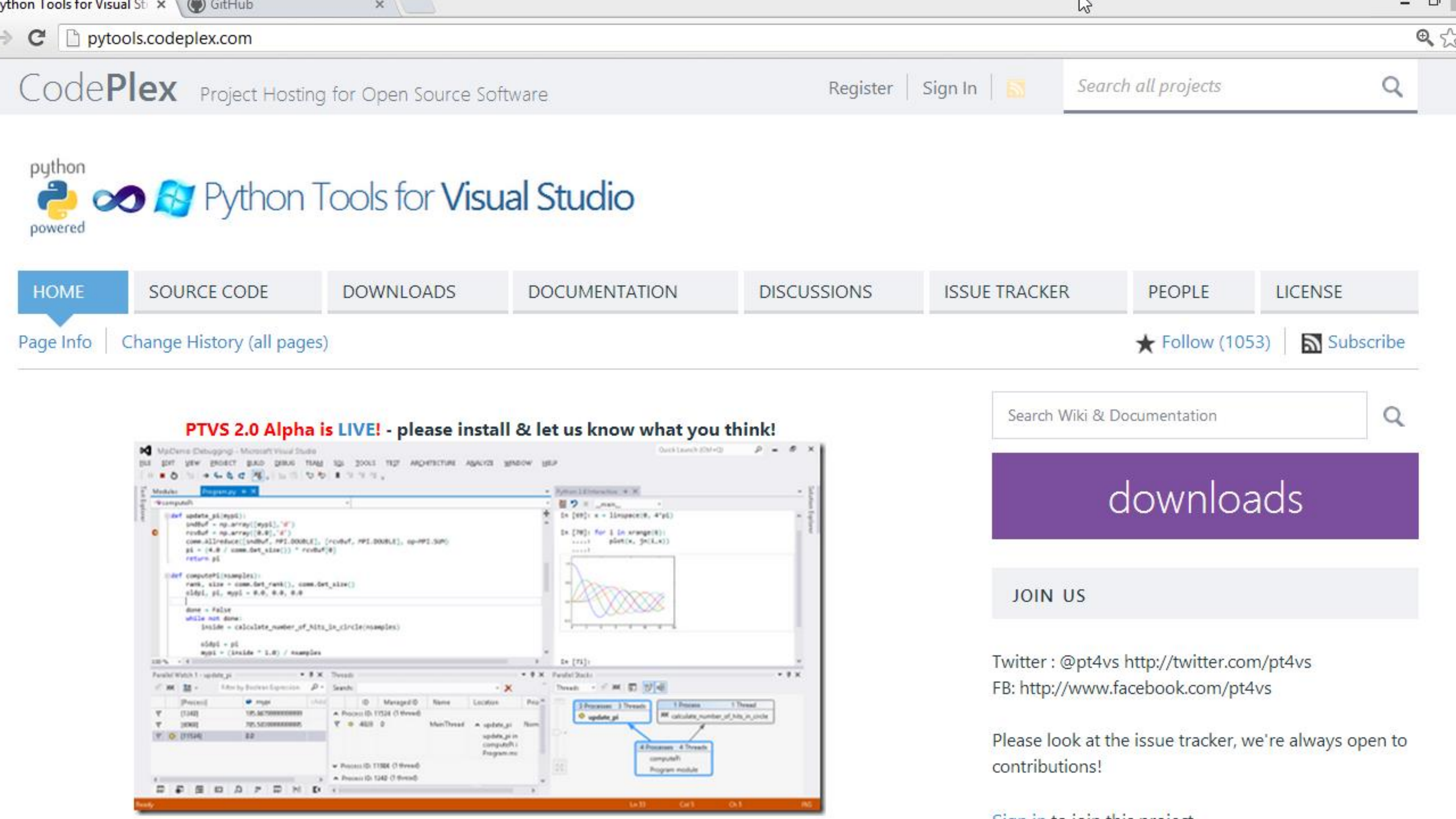
```
import numpy as np
import numpy.fft as fft

x = np.fromfile('signal.dat', dtype=('>i4,>i2,>i2'))

# Copy, convert to native floats, rescale
sig = np.array(x['f1'], dtype=np.float32) /  32768

# Alternate: since no file is provided, we can make up some data:
data = [math.sin(np.pi / 8 * i) for i in range(320)]
noise = [np.random.rand() * 0.02 for i in range(320)]
sig = np.array(data, dtype='f')
sig += np.array(noise)

f = fft.fft(sig)

print "Strongest frequency = %d n" % np.argmax(np.abs(f))
```

Experimental: .Net versions of Numpy + SciPy



Pyvot: A live bridge between PTVS and Excel

Windows Azure



CodePlex Project Hosting for Open Source Software

Python Tools for Visual Studio

PTVS 2.0 Alpha is LIVE! - please install & let us know what you think!

```
def update_pi(mypi):
    sndBuf = np.array([mypi],'d')
    rcvBuf = np.array([0.0],'d')
    comm.Allreduce([sndBuf, MPI.DOUBLE], [rcvBuf, MPI.DOUBLE], op=MPI.SUM)
    pi = (4.0 / comm.Get_size()) * rcvBuf[0]
    return pi

def computePi(nsamples):
    rank, size = comm.Get_rank(), comm.Get_size()
    oldpi, pi, mypi = 0.0, 0.0, 0.0

    done = False
    while not done:
        inside = calculate_number_of_hits_in_circle(nsamples)

        oldpi = pi
        mypi = (inside * 1.0) / nsamples
```

```
In [69]: x = linspace(0, 4*pi)
In [70]: for i in arange(6):
    ...:     plot(x, jn(i,x))
    ...:
In [71]:
```

JOIN US

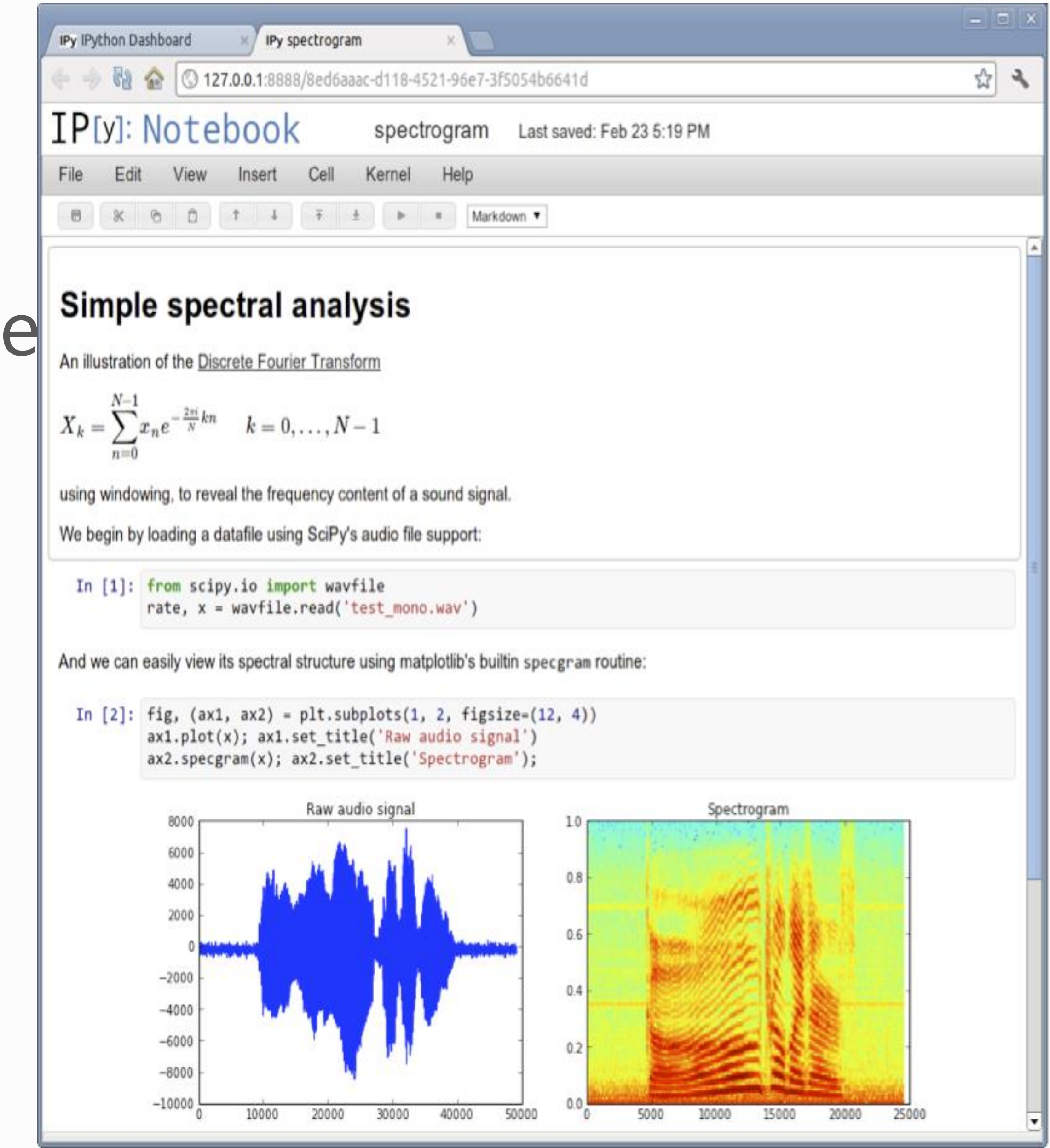
Twitter : @pt4vs http://twitter.com/pt4vs
FB: http://www.facebook.com/pt4vs

Please look at the issue tracker, we're always open to contributions!

Sign in to join this project

Python on Azure: IPython notebook

- Python IDE in browser
  - Any Browser
  - Any OS
- Backed by Python engine on Azure
  - Windows or Linux
- Key features
  - Intellisense, completion, …
  - Inline graphics
  - Markdown
- “Executable Document”
- IPython REPL also built-into PTVS



Simple spectral analysis

An illustration of the Discrete Fourier Transform

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} \qquad k = 0, \ldots, N-1$$

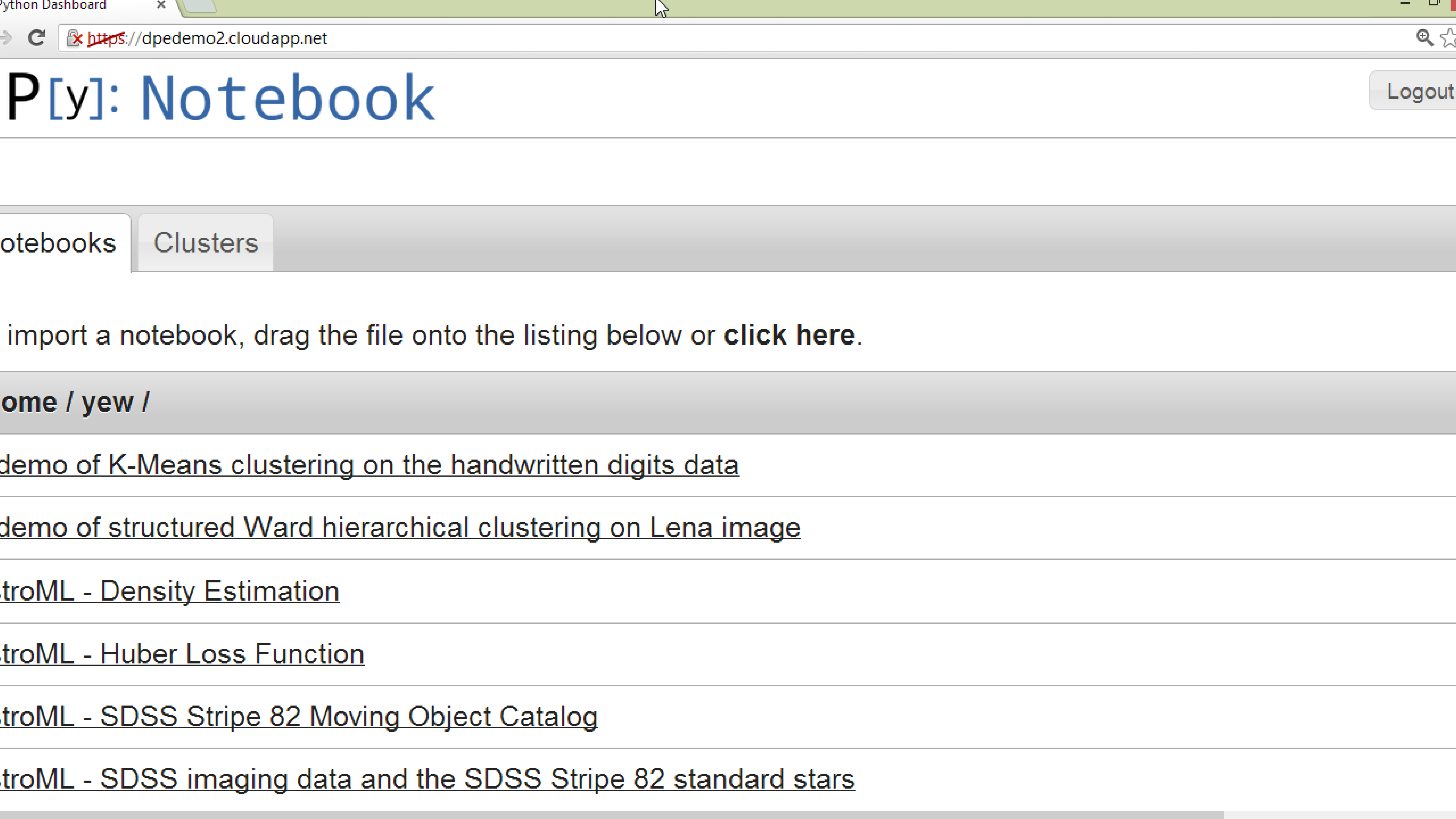
using windowing, to reveal the frequency content of a sound signal.

We begin by loading a datafile using SciPy's audio file support:

```
In [1]: from scipy.io import wavfile
        rate, x = wavfile.read('test_mono.wav')
```

And we can easily view its spectral structure using matplotlib's builtin `specgram` routine:

```
In [2]: fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 4))
        ax1.plot(x); ax1.set_title('Raw audio signal')
        ax2.specgram(x); ax2.set_title('Spectrogram');
```



import a notebook, drag the file onto the listing below or **click here**.

ome / yew /

demo of K-Means clustering on the handwritten digits data

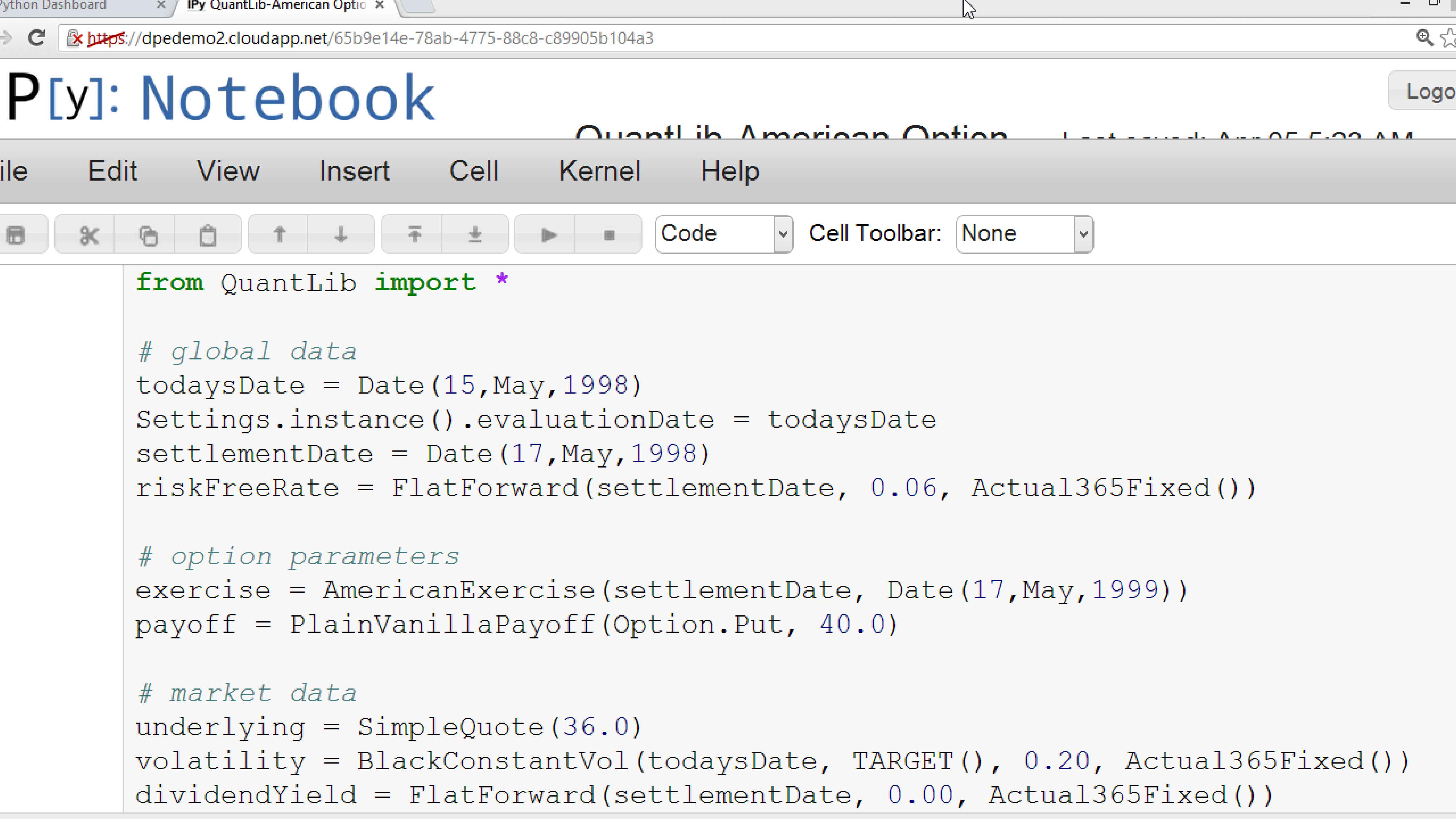
demo of structured Ward hierarchical clustering on Lena image

troML - Density Estimation

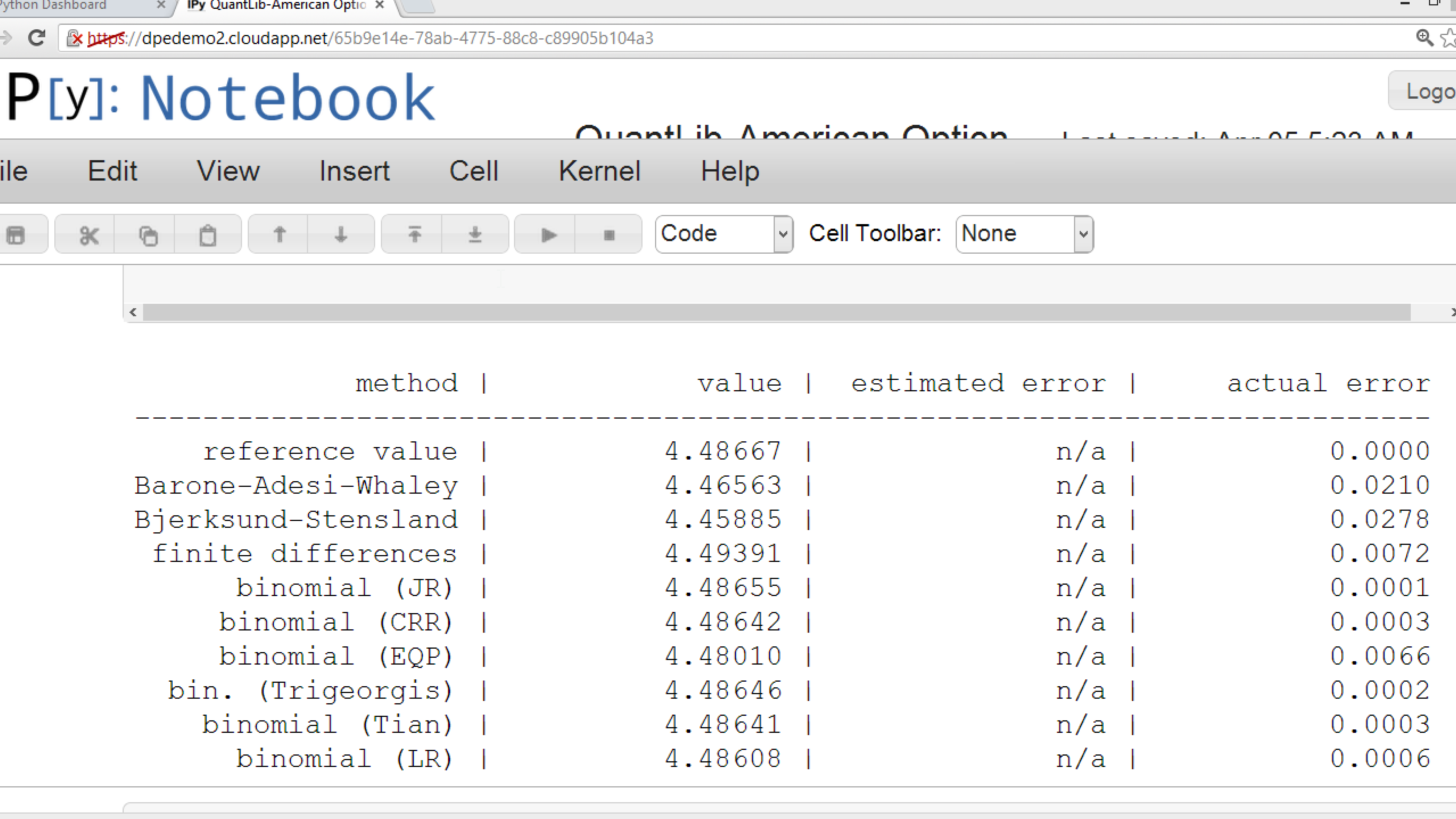
troML - Huber Loss Function

troML - SDSS Stripe 82 Moving Object Catalog

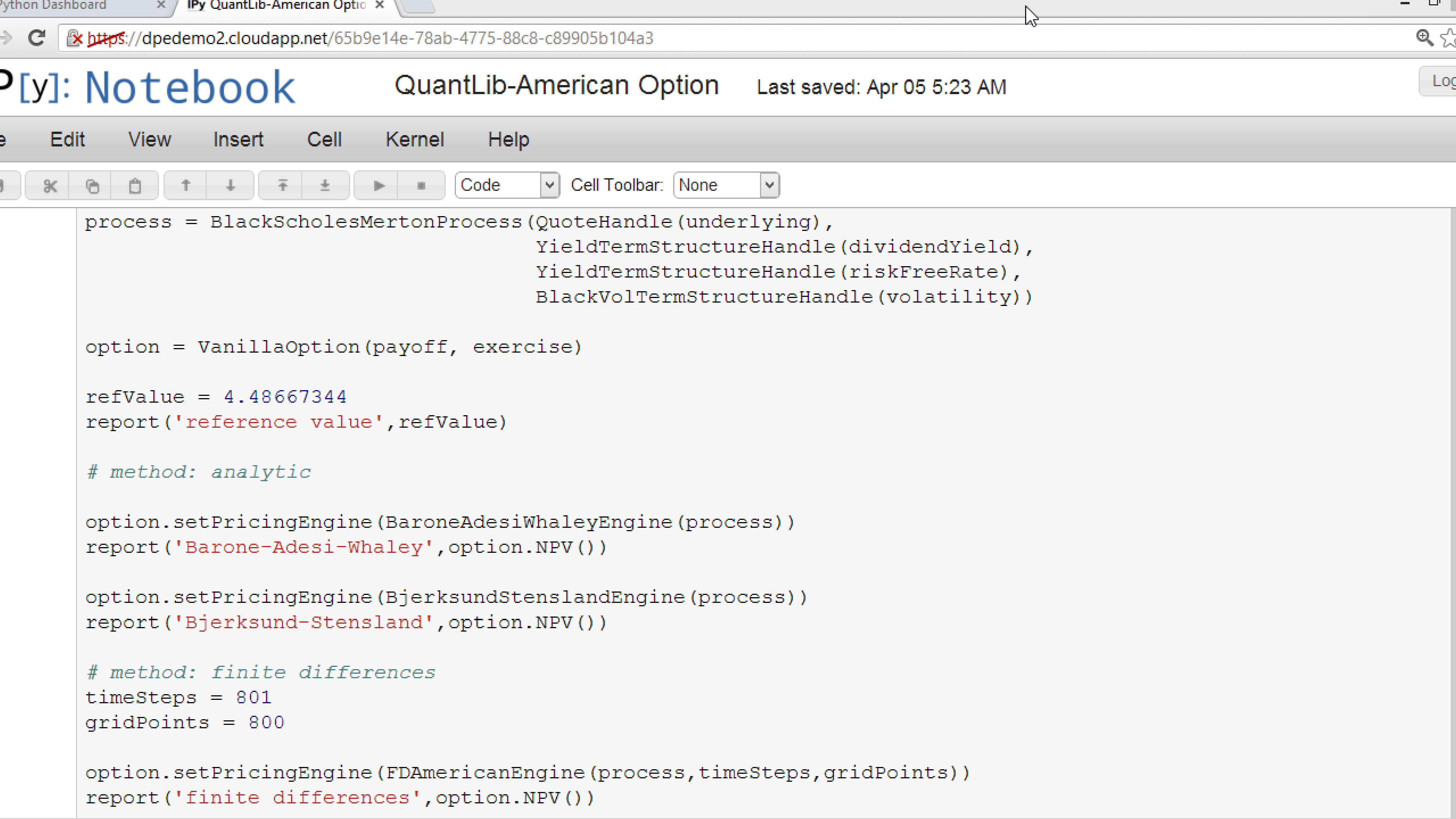
troML - SDSS imaging data and the SDSS Stripe 82 standard stars



```
from QuantLib import *

# global data
todaysDate = Date(15,May,1998)
Settings.instance().evaluationDate = todaysDate
settlementDate = Date(17,May,1998)
riskFreeRate = FlatForward(settlementDate, 0.06, Actual365Fixed())

# option parameters
exercise = AmericanExercise(settlementDate, Date(17,May,1999))
payoff = PlainVanillaPayoff(Option.Put, 40.0)

# market data
underlying = SimpleQuote(36.0)
volatility = BlackConstantVol(todaysDate, TARGET(), 0.20, Actual365Fixed())
dividendYield = FlatForward(settlementDate, 0.00, Actual365Fixed())
```


Python Dashboard
IPy QuantLib-American Optio
https://dpedemo2.cloudapp.net/65b9e14e-78ab-4775-88c8-c89905b104a3
IP[y]: Notebook
QuantLib-American Option
Last saved: Apr 05 5:23 AM
Logo
ile
Edit
View
Insert
Cell
Kernel
Help
Code
Cell Toolbar:
None

```
           method |        value |  estimated error |      actual error
-----------------------------------------------------------------------
  reference value |      4.48667 |              n/a |           0.0000
Barone-Adesi-Whaley |      4.46563 |              n/a |           0.0210
Bjerksund-Stensland |      4.45885 |              n/a |           0.0278
 finite differences |      4.49391 |              n/a |           0.0072
     binomial (JR) |      4.48655 |              n/a |           0.0001
    binomial (CRR) |      4.48642 |              n/a |           0.0003
    binomial (EQP) |      4.48010 |              n/a |           0.0066
 bin. (Trigeorgis) |      4.48646 |              n/a |           0.0002
   binomial (Tian) |      4.48641 |              n/a |           0.0003
     binomial (LR) |      4.48608 |              n/a |           0.0006
```


ython Dashboard
IPy QuantLib-American Optio
https://dpedemo2.cloudapp.net/65b9e14e-78ab-4775-88c8-c89905b104a3
P[y]: Notebook
Log
e
Edit
View
Insert
Cell
Kernel
Help

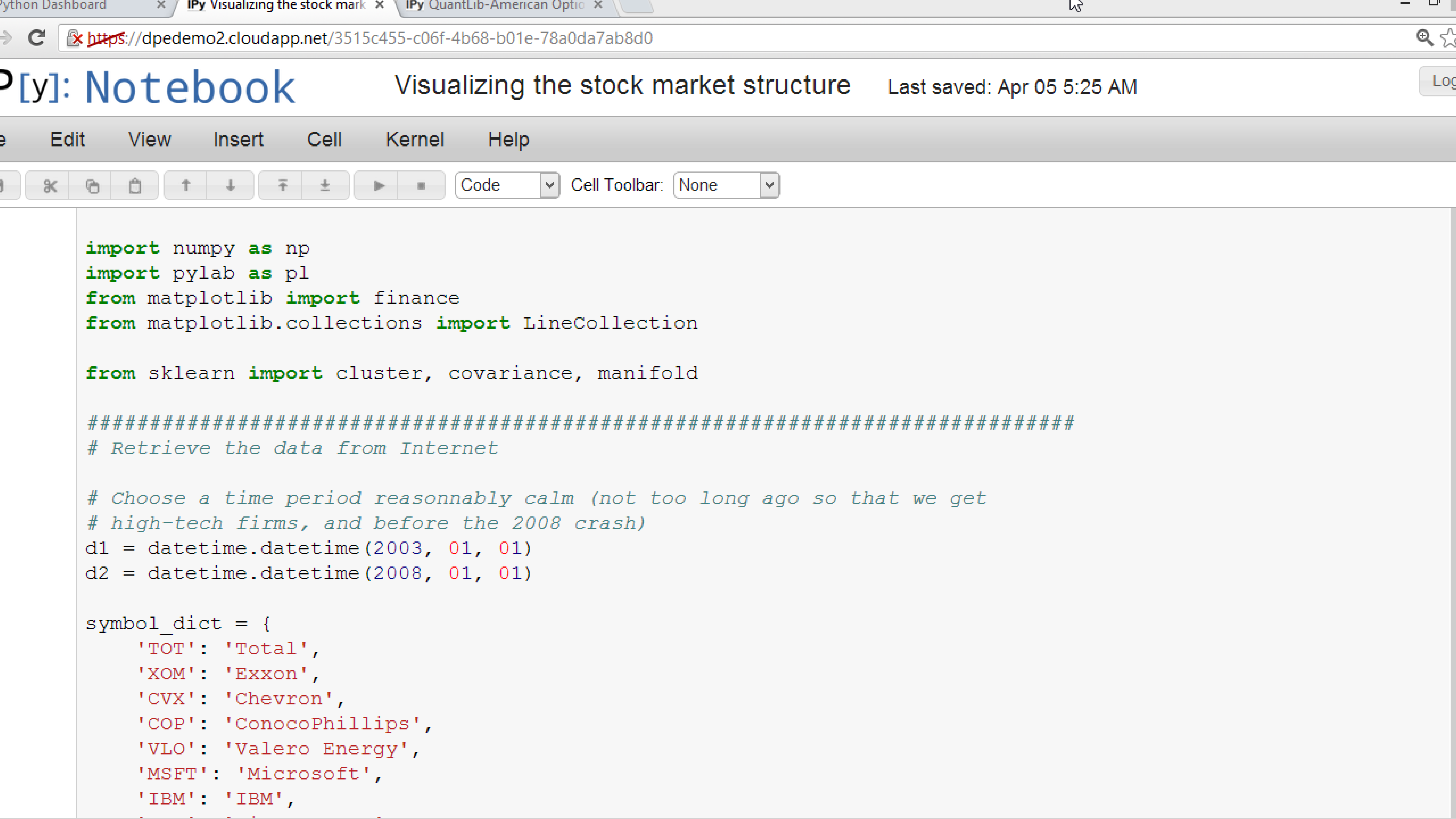
QuantLib-American Option

Last saved: Apr 05 5:23 AM

Code

Cell Toolbar: None

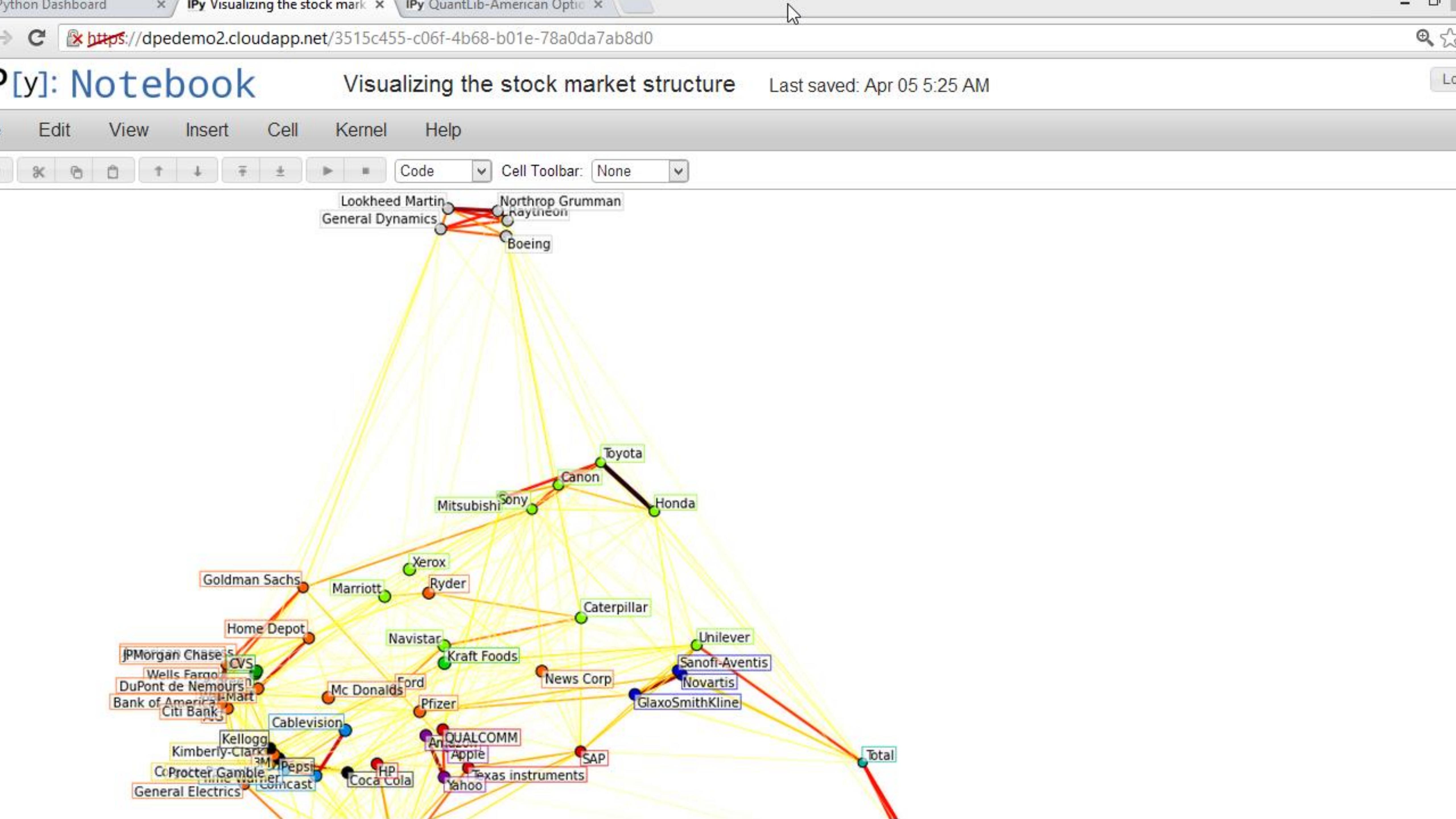
```
process = BlackScholesMertonProcess(QuoteHandle(underlying),
                                    YieldTermStructureHandle(dividendYield),
                                    YieldTermStructureHandle(riskFreeRate),
                                    BlackVolTermStructureHandle(volatility))

option = VanillaOption(payoff, exercise)

refValue = 4.48667344
report('reference value',refValue)

# method: analytic

option.setPricingEngine(BaroneAdesiWhaleyEngine(process))
report('Barone-Adesi-Whaley',option.NPV())

option.setPricingEngine(BjerksundStenslandEngine(process))
report('Bjerksund-Stensland',option.NPV())

# method: finite differences
timeSteps = 801
gridPoints = 800

option.setPricingEngine(FDAmericanEngine(process,timeSteps,gridPoints))
report('finite differences',option.NPV())
```


ython Dashboard
IPy Visualizing the stock mark
IPy QuantLib-American Optio
https://dpedemo2.cloudapp.net/3515c455-c06f-4b68-b01e-78a0da7ab8d0
[y]: Notebook
Log
e
Edit
View
Insert
Cell
Kernel
Help
Code
Cell Toolbar:
None

Visualizing the stock market structure

Last saved: Apr 05 5:25 AM

```
import numpy as np
import pylab as pl
from matplotlib import finance
from matplotlib.collections import LineCollection

from sklearn import cluster, covariance, manifold

###############################################################################
# Retrieve the data from Internet

# Choose a time period reasonnably calm (not too long ago so that we get
# high-tech firms, and before the 2008 crash)
d1 = datetime.datetime(2003, 01, 01)
d2 = datetime.datetime(2008, 01, 01)

symbol_dict = {
    'TOT': 'Total',
    'XOM': 'Exxon',
    'CVX': 'Chevron',
    'COP': 'ConocoPhillips',
    'VLO': 'Valero Energy',
    'MSFT': 'Microsoft',
    'IBM': 'IBM',
```



Visualizing the stock market structure

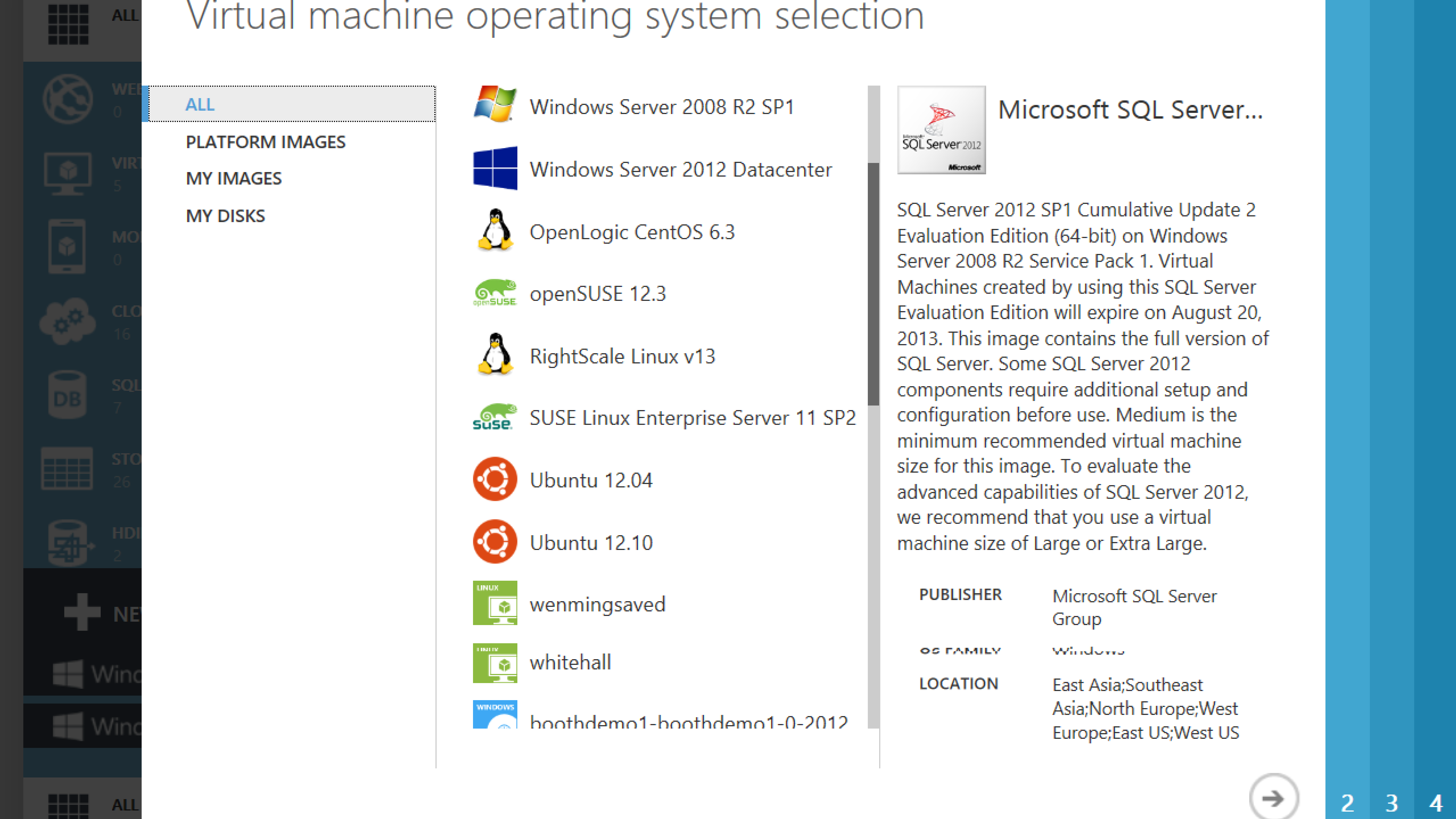
Last saved: Apr 05 5:25 AM

Virtual machines

Windows Server and Linux

Flexible Workload Support

Virtual Private Networking



Virtual machine operating system selection

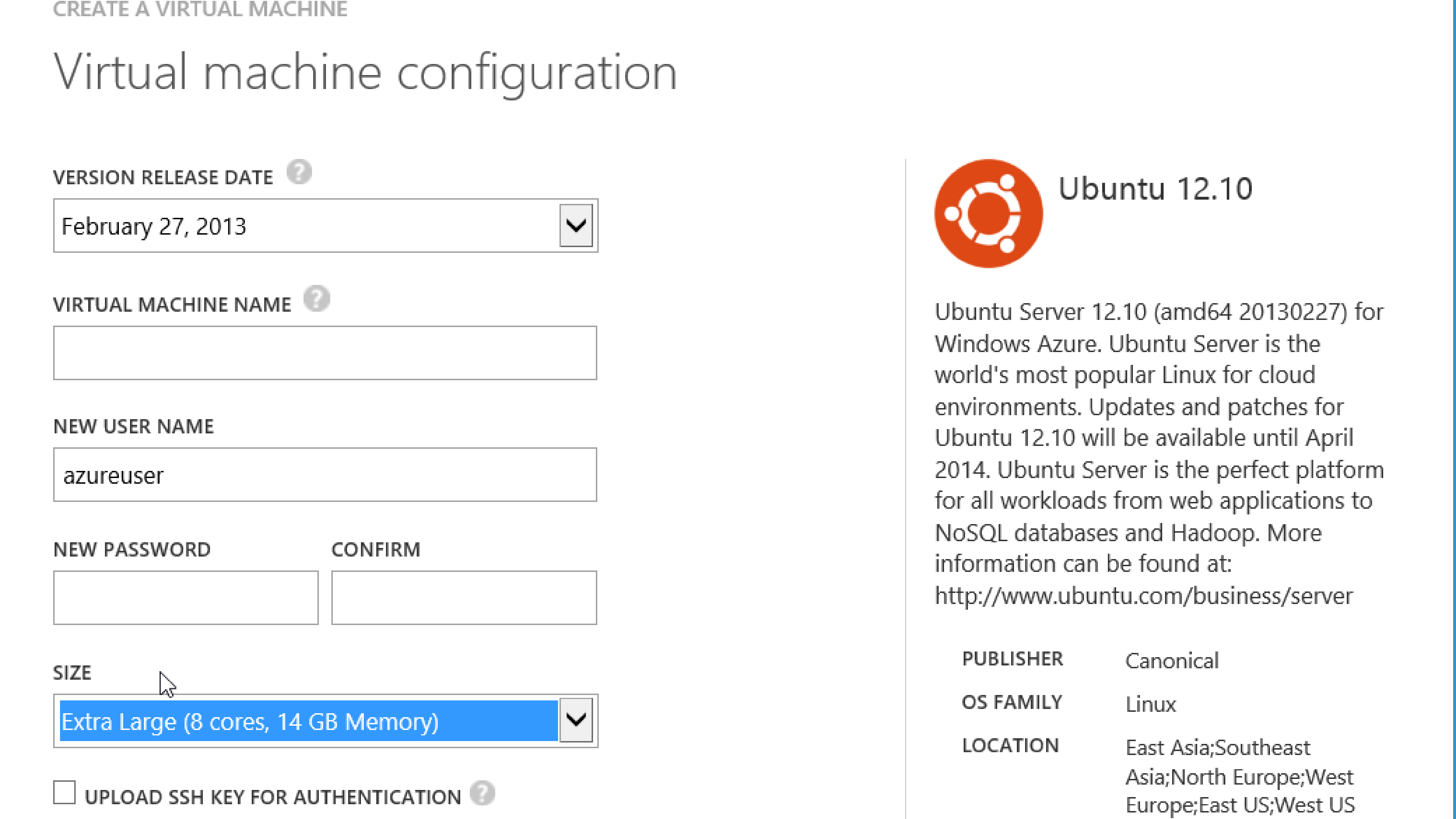
- ALL
- PLATFORM IMAGES
- MY IMAGES
- MY DISKS

- Windows Server 2008 R2 SP1
- Windows Server 2012 Datacenter
- OpenLogic CentOS 6.3
- openSUSE 12.3
- RightScale Linux v13
- SUSE Linux Enterprise Server 11 SP2
- Ubuntu 12.04
- Ubuntu 12.10
- wenmingsaved
- whitehall
- boothdemo1-boothdemo1-0-2012

Microsoft SQL Server...

SQL Server 2012 SP1 Cumulative Update 2 Evaluation Edition (64-bit) on Windows Server 2008 R2 Service Pack 1. Virtual Machines created by using this SQL Server Evaluation Edition will expire on August 20, 2013. This image contains the full version of SQL Server. Some SQL Server 2012 components require additional setup and configuration before use. Medium is the minimum recommended virtual machine size for this image. To evaluate the advanced capabilities of SQL Server 2012, we recommend that you use a virtual machine size of Large or Extra Large.

PUBLISHER	Microsoft SQL Server Group
OS FAMILY	Windows
LOCATION	East Asia;Southeast Asia;North Europe;West Europe;East US;West US



CREATE A VIRTUAL MACHINE

Virtual machine configuration

VERSION RELEASE DATE

February 27, 2013

VIRTUAL MACHINE NAME

NEW USER NAME

azureuser

NEW PASSWORD

CONFIRM

SIZE

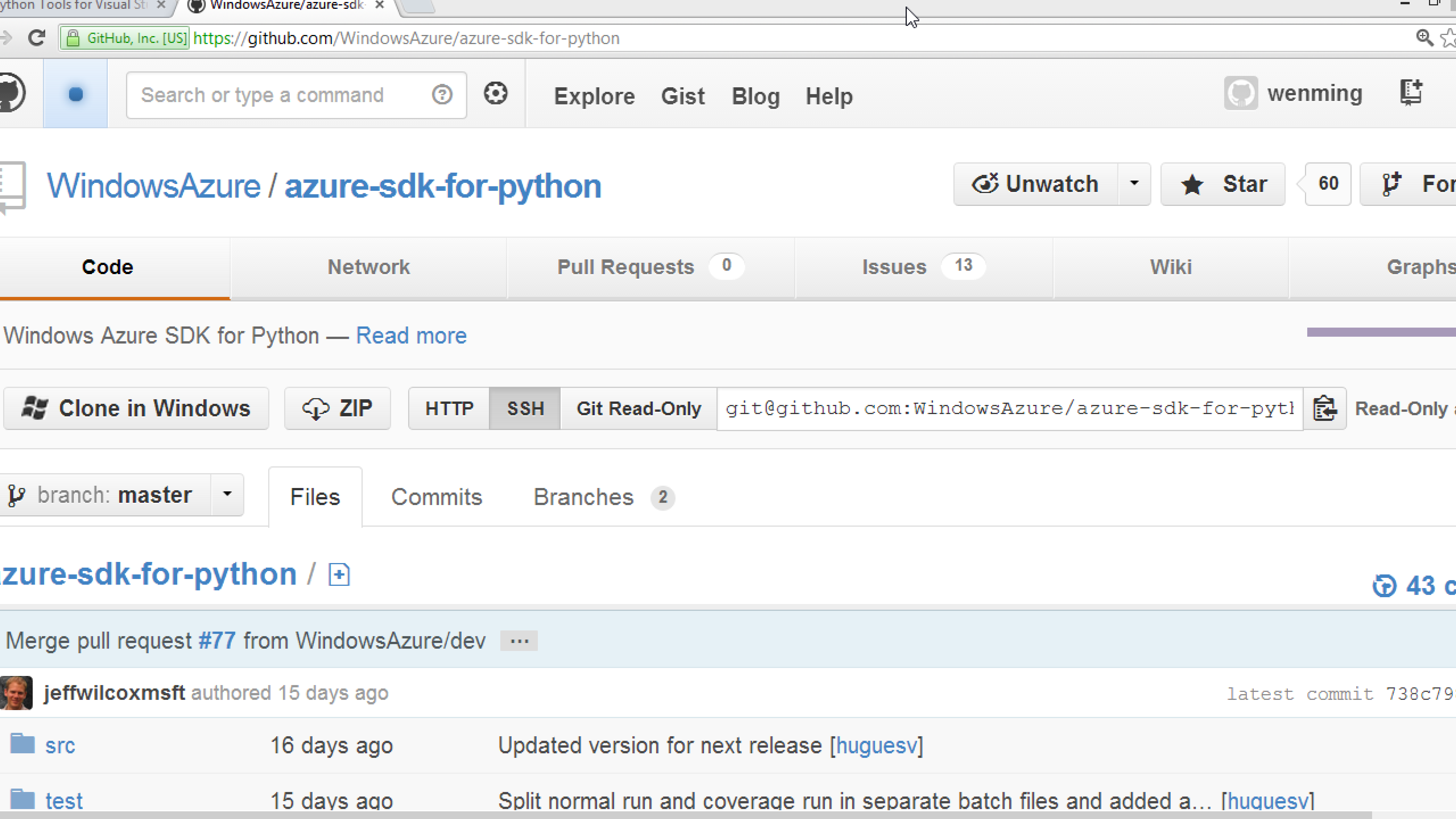
Extra Large (8 cores, 14 GB Memory)

□ UPLOAD SSH KEY FOR AUTHENTICATION

Ubuntu 12.10

Ubuntu Server 12.10 (amd64 20130227) for Windows Azure. Ubuntu Server is the world's most popular Linux for cloud environments. Updates and patches for Ubuntu 12.10 will be available until April 2014. Ubuntu Server is the perfect platform for all workloads from web applications to NoSQL databases and Hadoop. More information can be found at: http://www.ubuntu.com/business/server

PUBLISHER	Canonical
OS FAMILY	Linux
LOCATION	East Asia;Southeast Asia;North Europe;West Europe;East US;West US



WindowsAzure / azure-sdk-for-python

Windows Azure SDK for Python — Read more

git@github.com:WindowsAzure/azure-sdk-for-pytl

zure-sdk-for-python /

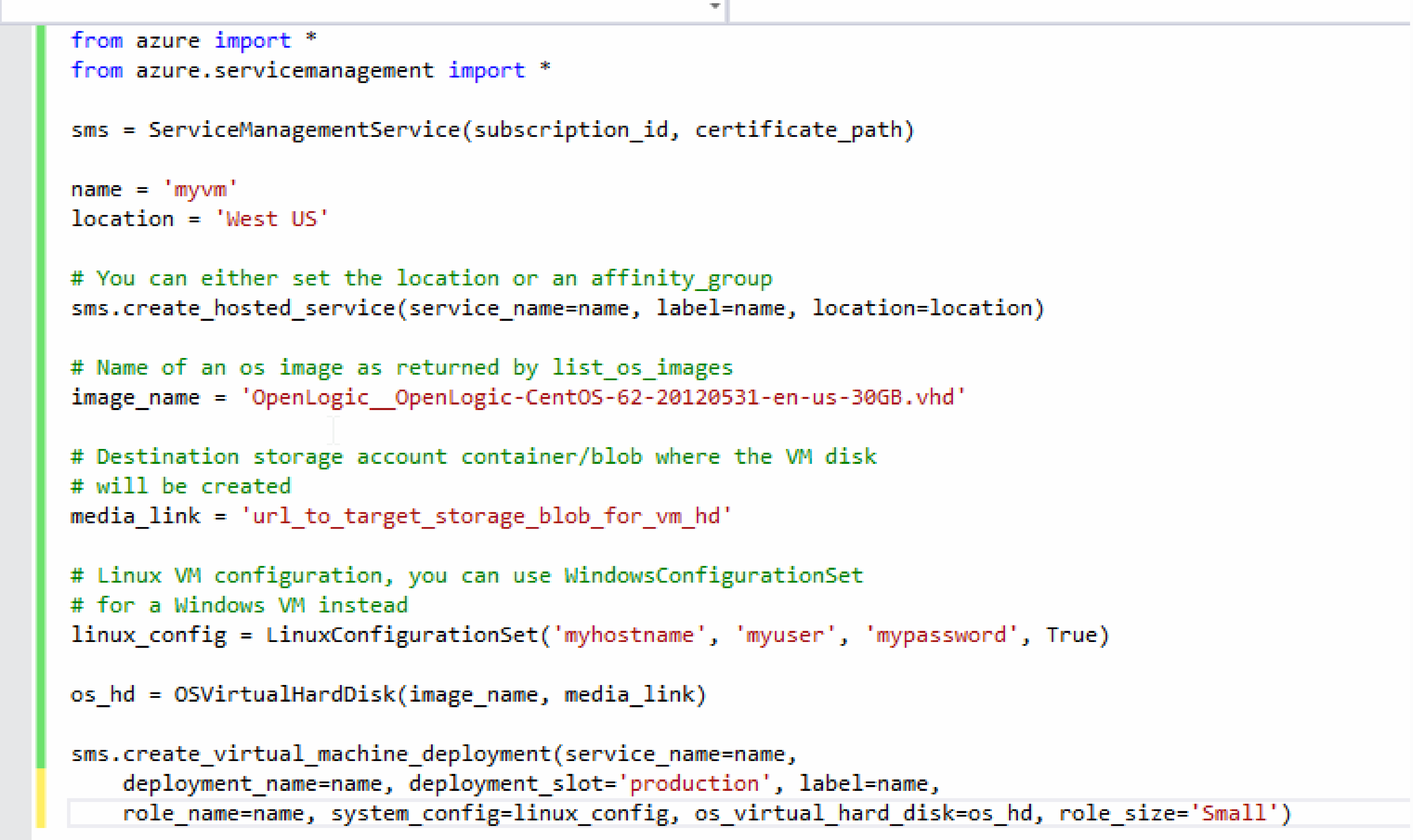
Merge pull request #77 from WindowsAzure/dev …

jeffwilcoxmsft authored 15 days ago latest commit 738c79

src	16 days ago	Updated version for next release [huguesv]
test	15 days ago	Split normal run and coverage run in separate batch files and added a… [huguesv]

Deploy & run Python on Virtual Machine

Demo



```python
from azure import *
from azure.servicemanagement import *

sms = ServiceManagementService(subscription_id, certificate_path)

name = 'myvm'
location = 'West US'

# You can either set the location or an affinity_group
sms.create_hosted_service(service_name=name, label=name, location=location)

# Name of an os image as returned by list_os_images
image_name = 'OpenLogic__OpenLogic-CentOS-62-20120531-en-us-30GB.vhd'

# Destination storage account container/blob where the VM disk
# will be created
media_link = 'url_to_target_storage_blob_for_vm_hd'

# Linux VM configuration, you can use WindowsConfigurationSet
# for a Windows VM instead
linux_config = LinuxConfigurationSet('myhostname', 'myuser', 'mypassword', True)

os_hd = OSVirtualHardDisk(image_name, media_link)

sms.create_virtual_machine_deployment(service_name=name,
    deployment_name=name, deployment_slot='production', label=name,
    role_name=name, system_config=linux_config, os_virtual_hard_disk=os_hd, role_size='Small')
```

Virtual machines

Gallery Images

Virtual Machine Portability

VMs with persistent Drives

Gallery Images Available

Microsoft

Windows Server 2008 R2

SQL Server Eval 2012

Windows Server 2012

Biztalk Server 2013 Beta

Open Source

OpenSUSE 12.2

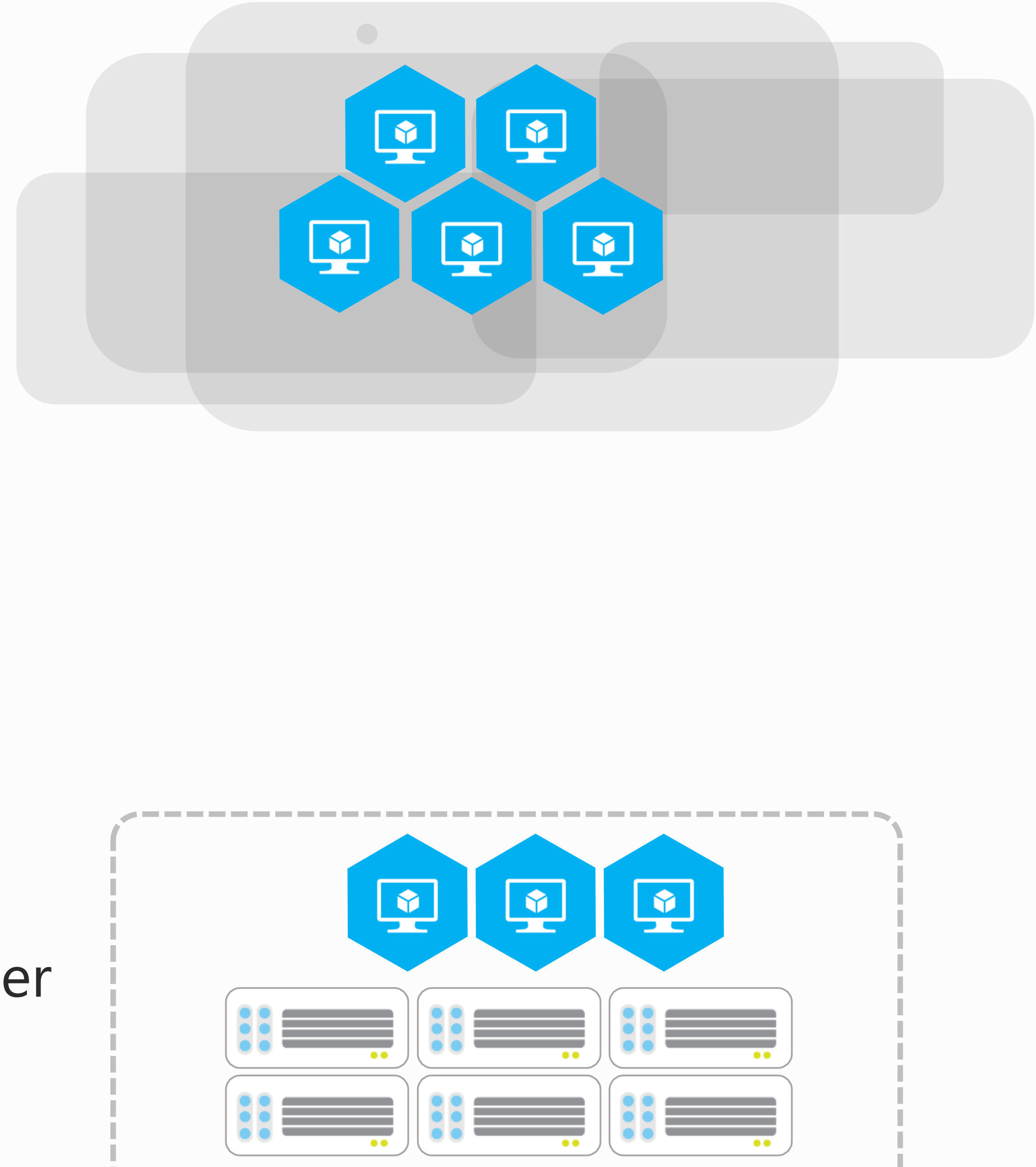
CentOS 6.3

Ubuntu 12.04/12.10

SUSE Linux Enterprise Server 11 SP2

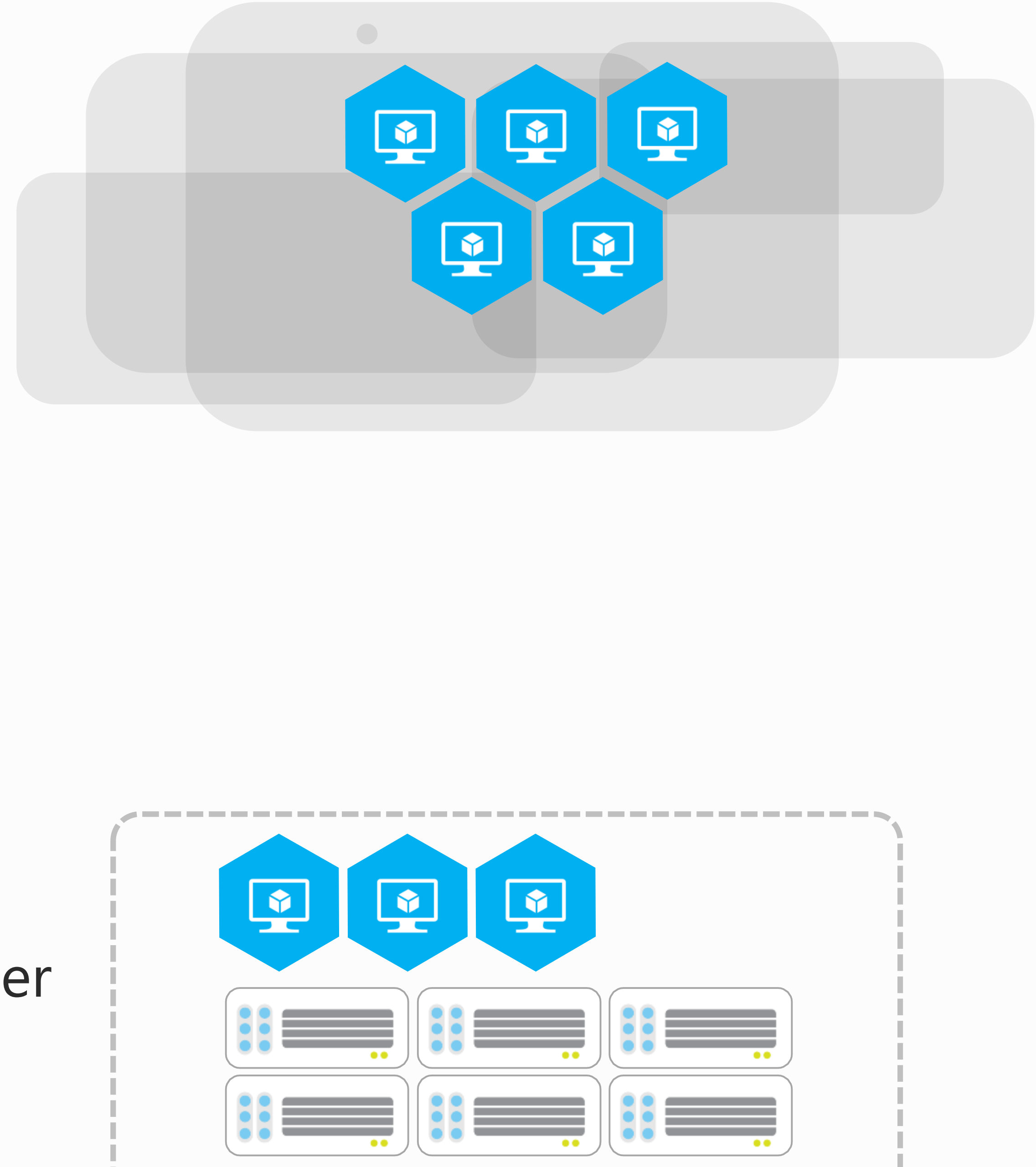
Virtual machine portability

Windows Azure

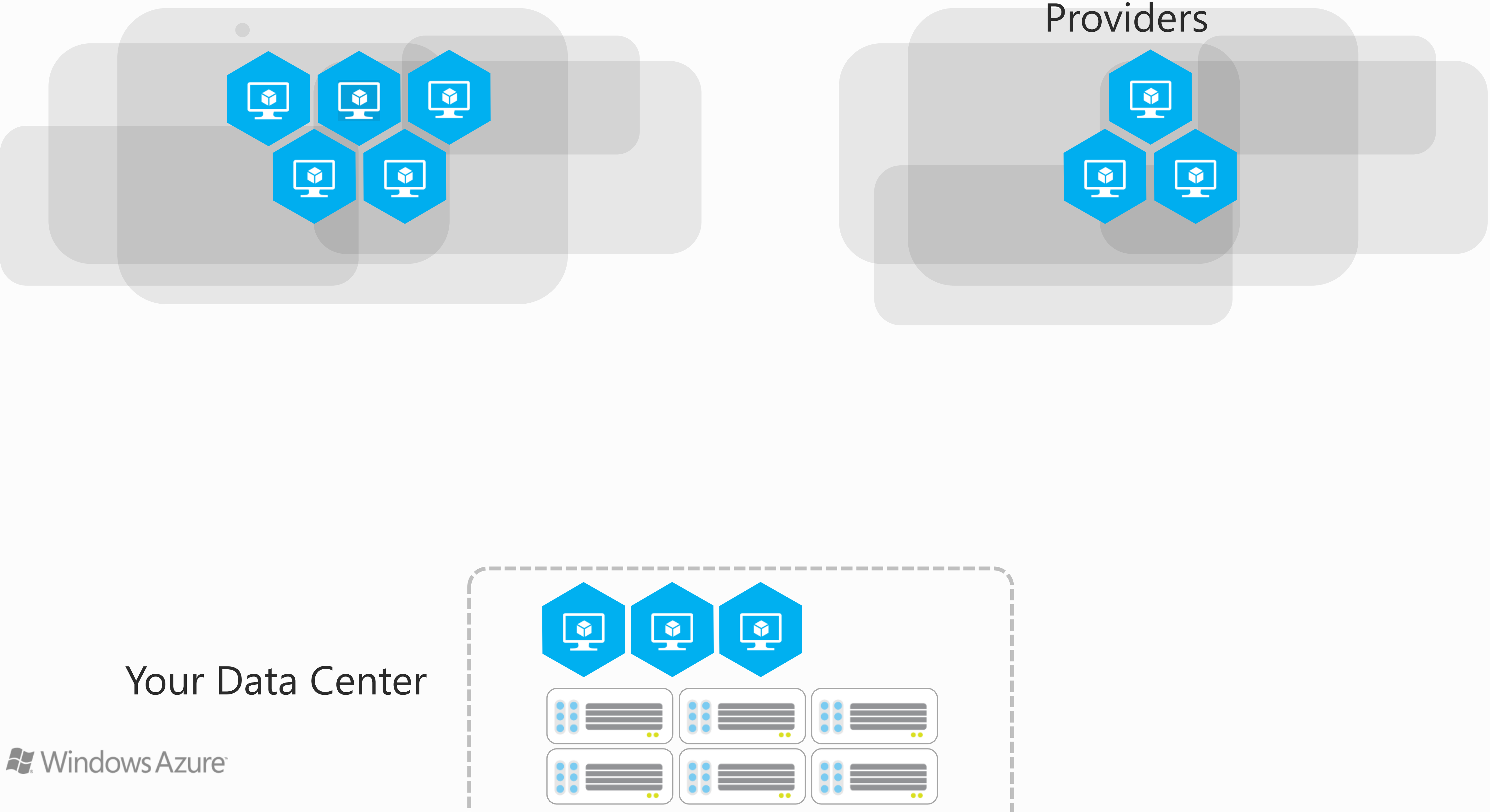


Your Data Center

Windows Azure



Your Data Center



Windows Azure

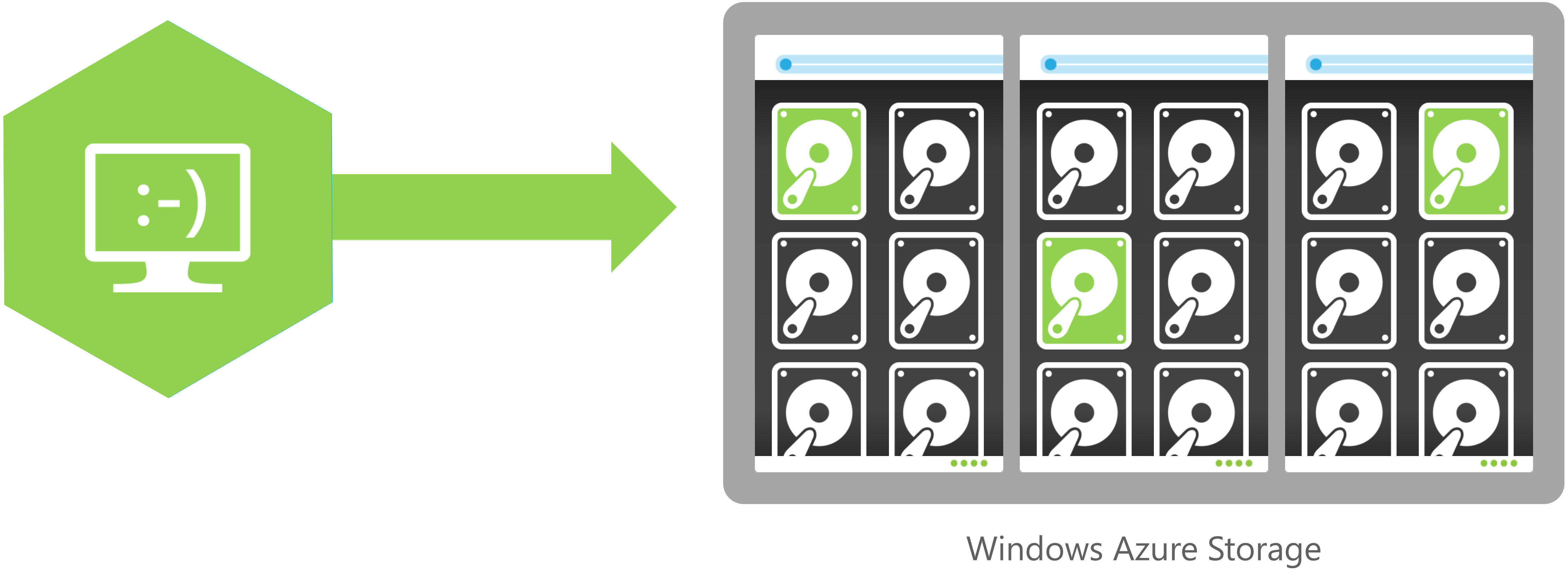
Other Service Providers

Your Data Center

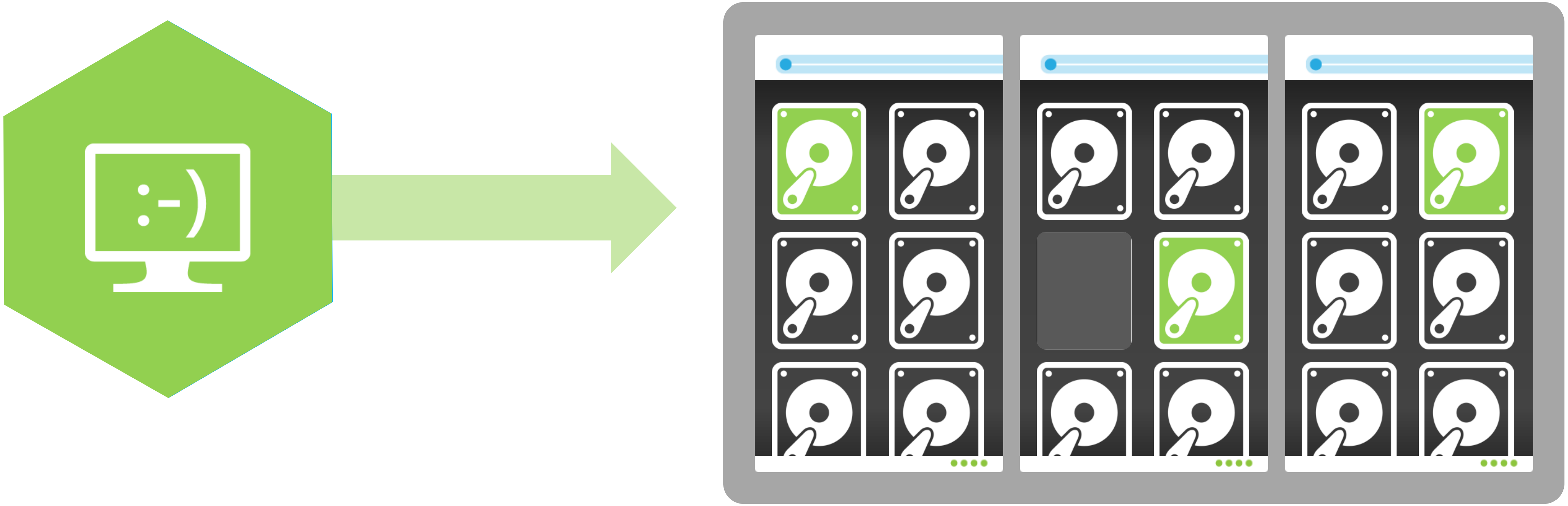
VM with persistent drive

Windows Azure Storage

VM with persistent drive



VM with persistent drive



Windows Azure Storage

Cloud services

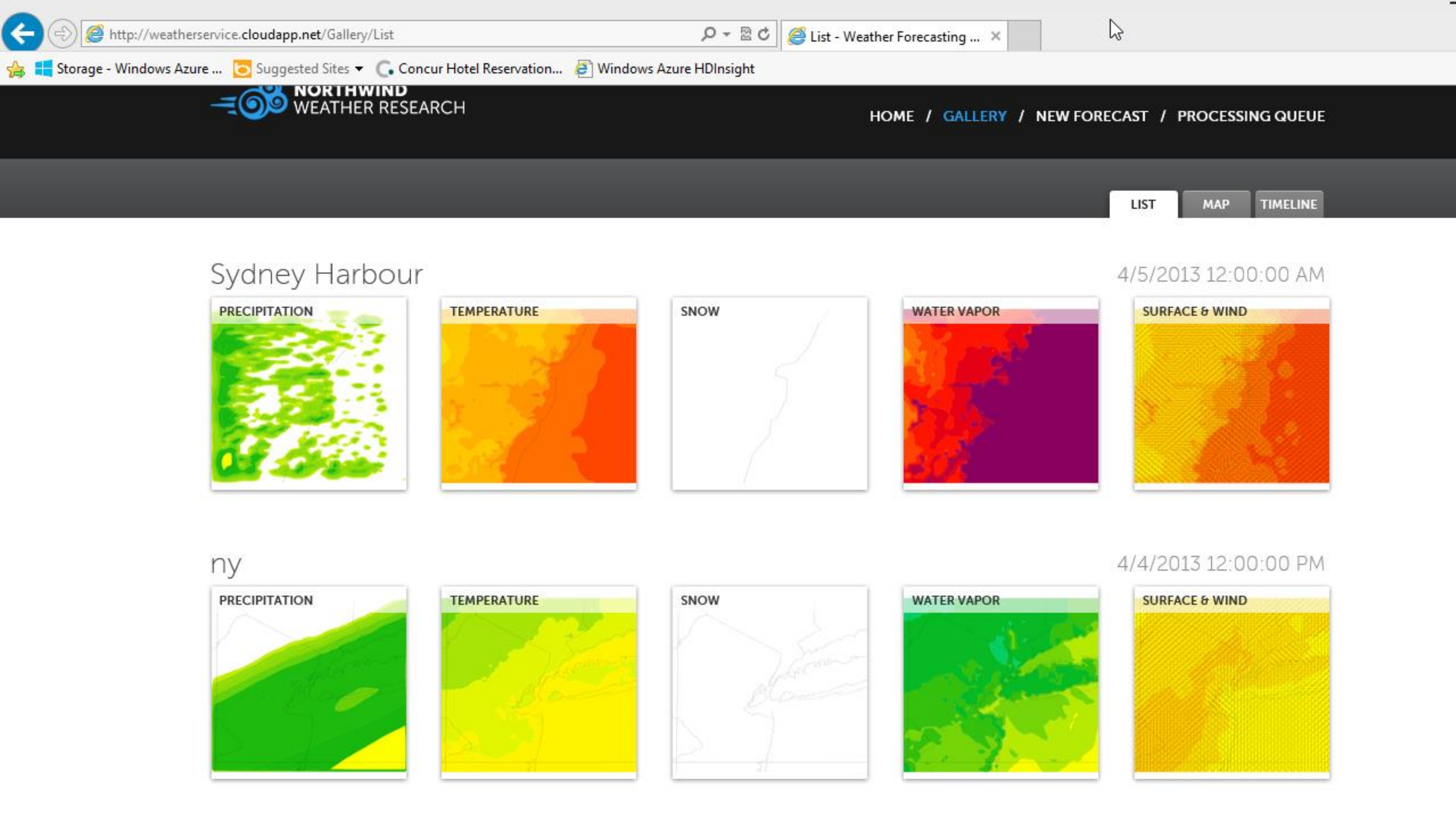
Build infinitely scalable apps and services

Support rich multi-tier architectures

Automated application management

Cloud services: Development

Demo

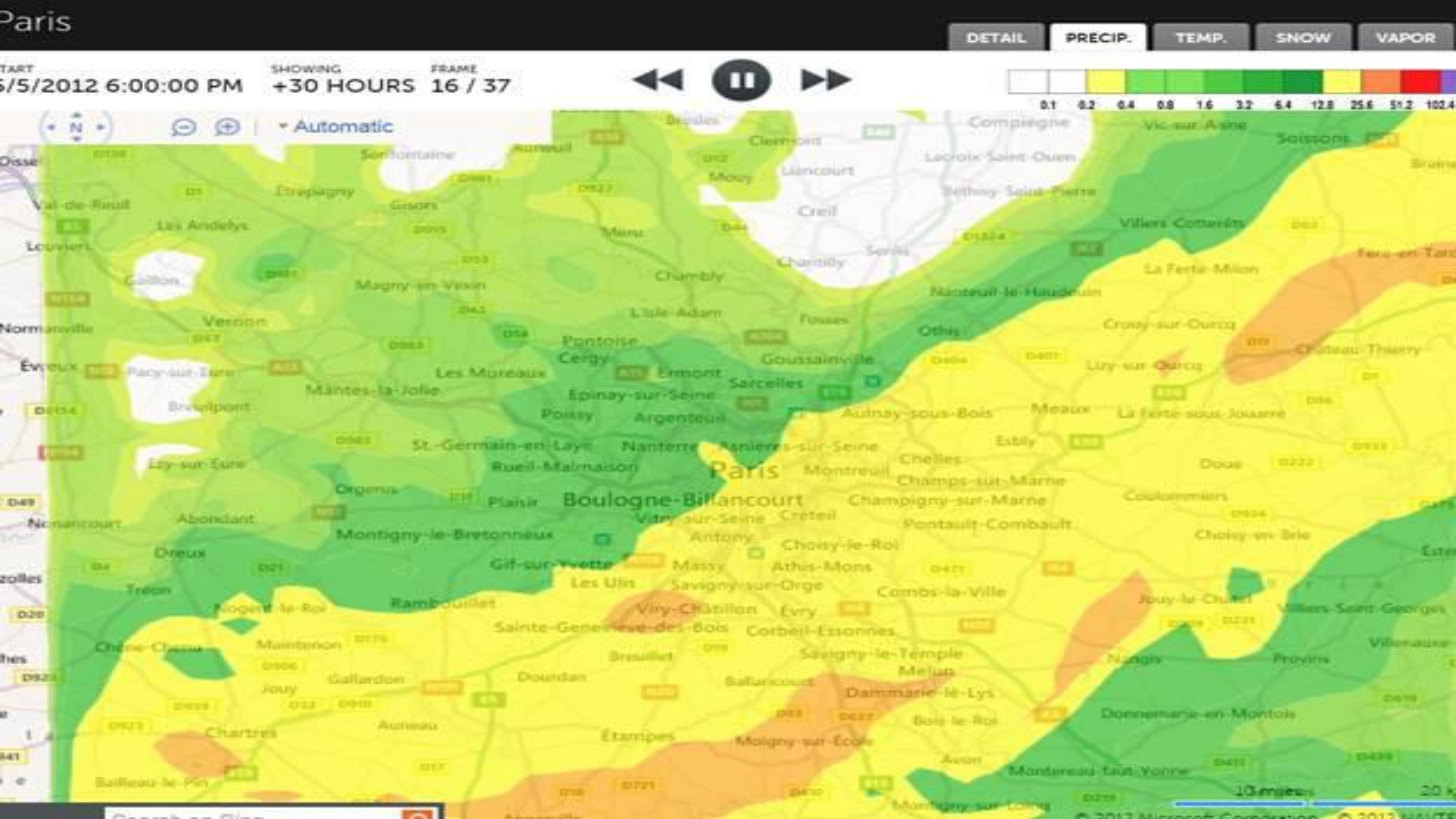

http://weatherservice.cloudapp.net/Gallery/List
List - Weather Forecasting ...
Storage - Windows Azure ...
Suggested Sites
Concur Hotel Reservation...
Windows Azure HDInsight
NORTHWIND
WEATHER RESEARCH
HOME / GALLERY / NEW FORECAST / PROCESSING QUEUE
LIST
MAP
TIMELINE

Sydney Harbour 4/5/2013 12:00:00 AM

PRECIPITATION TEMPERATURE SNOW WATER VAPOR SURFACE & WIND

ny 4/4/2013 12:00:00 PM

PRECIPITATION TEMPERATURE SNOW WATER VAPOR SURFACE & WIND

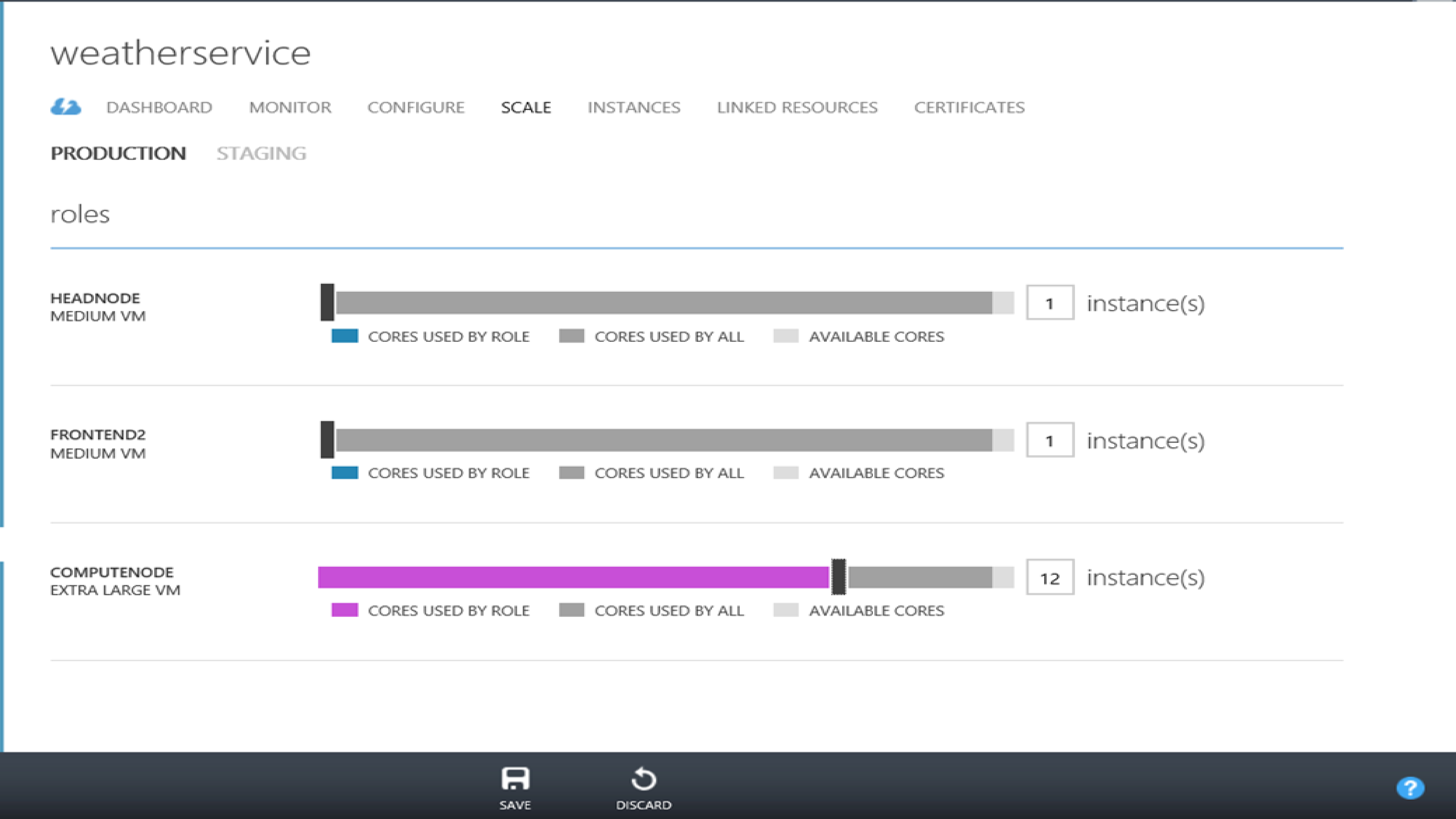


Paris

START
6/5/2012 6:00:00 PM

SHOWING
+30 HOURS

FRAME
16 / 37



weatherservice

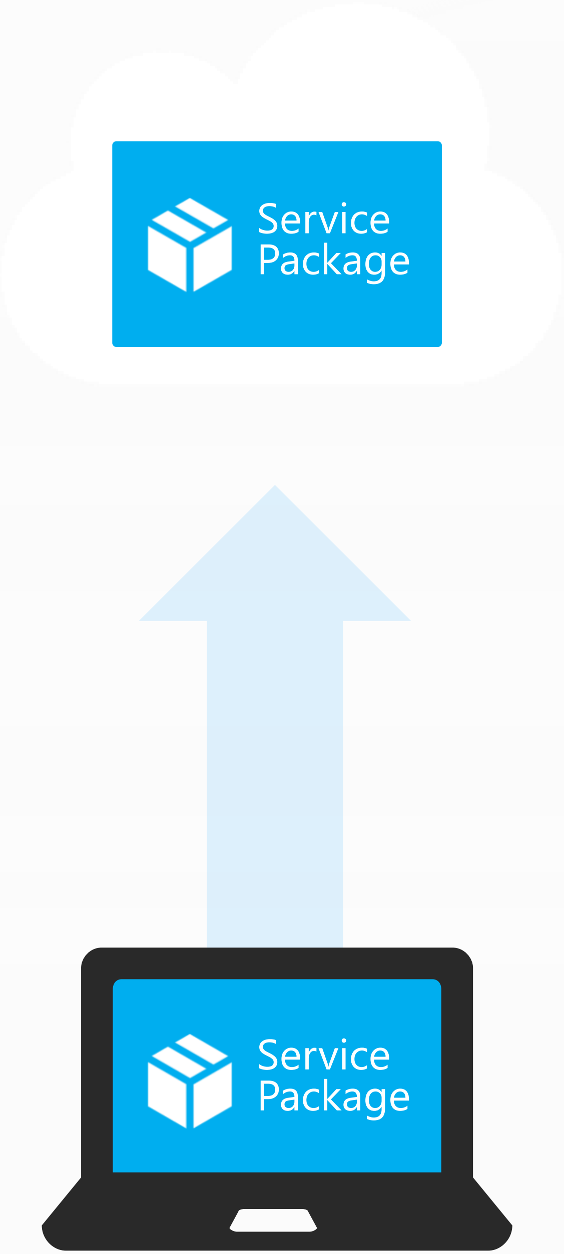
roles

HEADNODE
MEDIUM VM
1 instance(s)

FRONTEND2
MEDIUM VM
1 instance(s)

COMPUTENODE
EXTRA LARGE VM
12 instance(s)

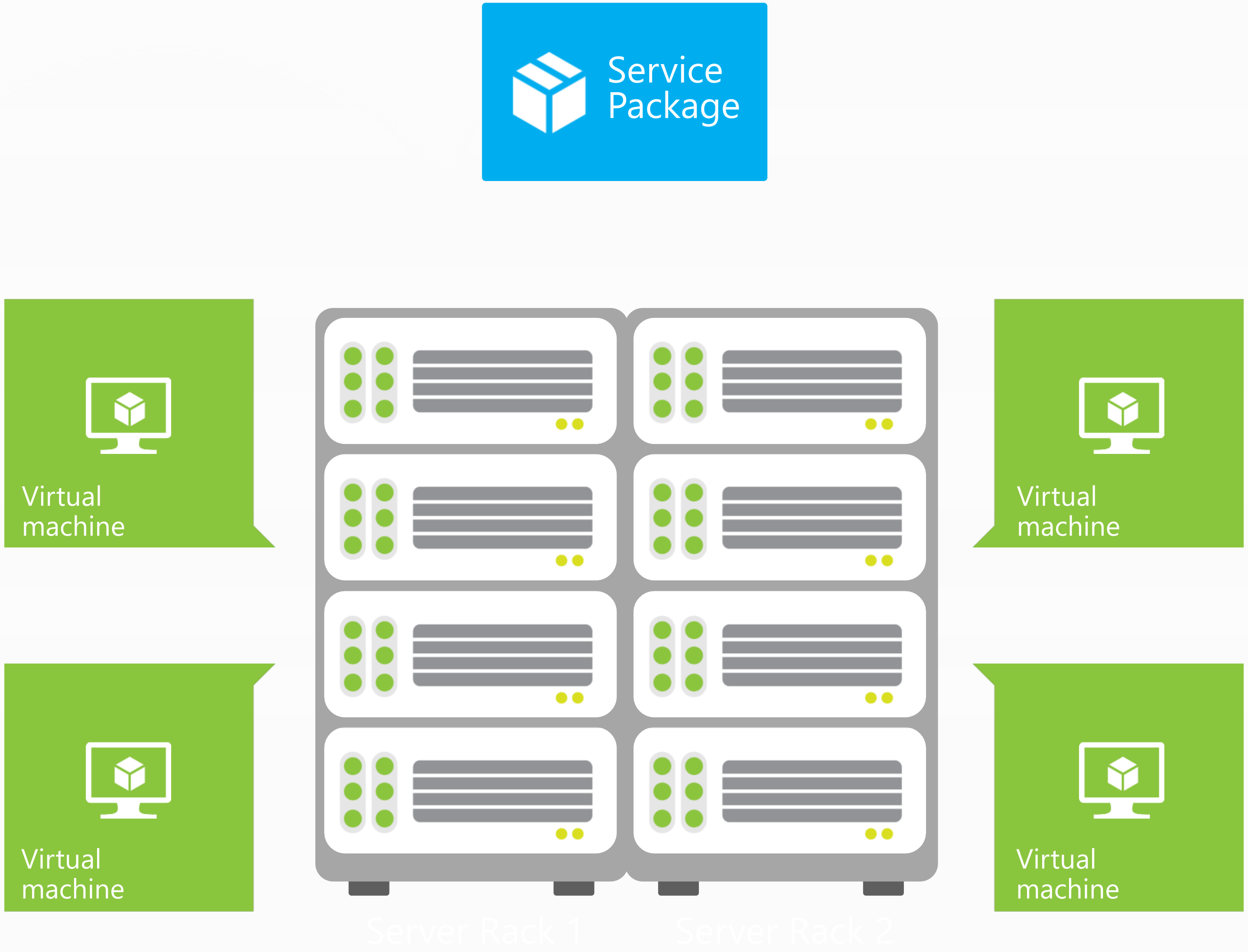
```python
# Create container
from azure.storage import BlobService
blob_service = BlobService(account_name, account_key)
blob_service.create_container('taskcontainer')

# Upload
from azure.storage import BlobService
blob_service = BlobService(account_name, account_key)
blob_service.put_blob('taskcontainer', 'task1',
file('task1-upload.txt').read(), 'BlockBlob')

#Download
from azure.storage import BlobService
blob_service = BlobService(account_name, account_key)
blob = blob_service.get_blob('taskcontainer', 'task1')
```


Service Package
Service Package

Provision Role Instances

Deploy App Code

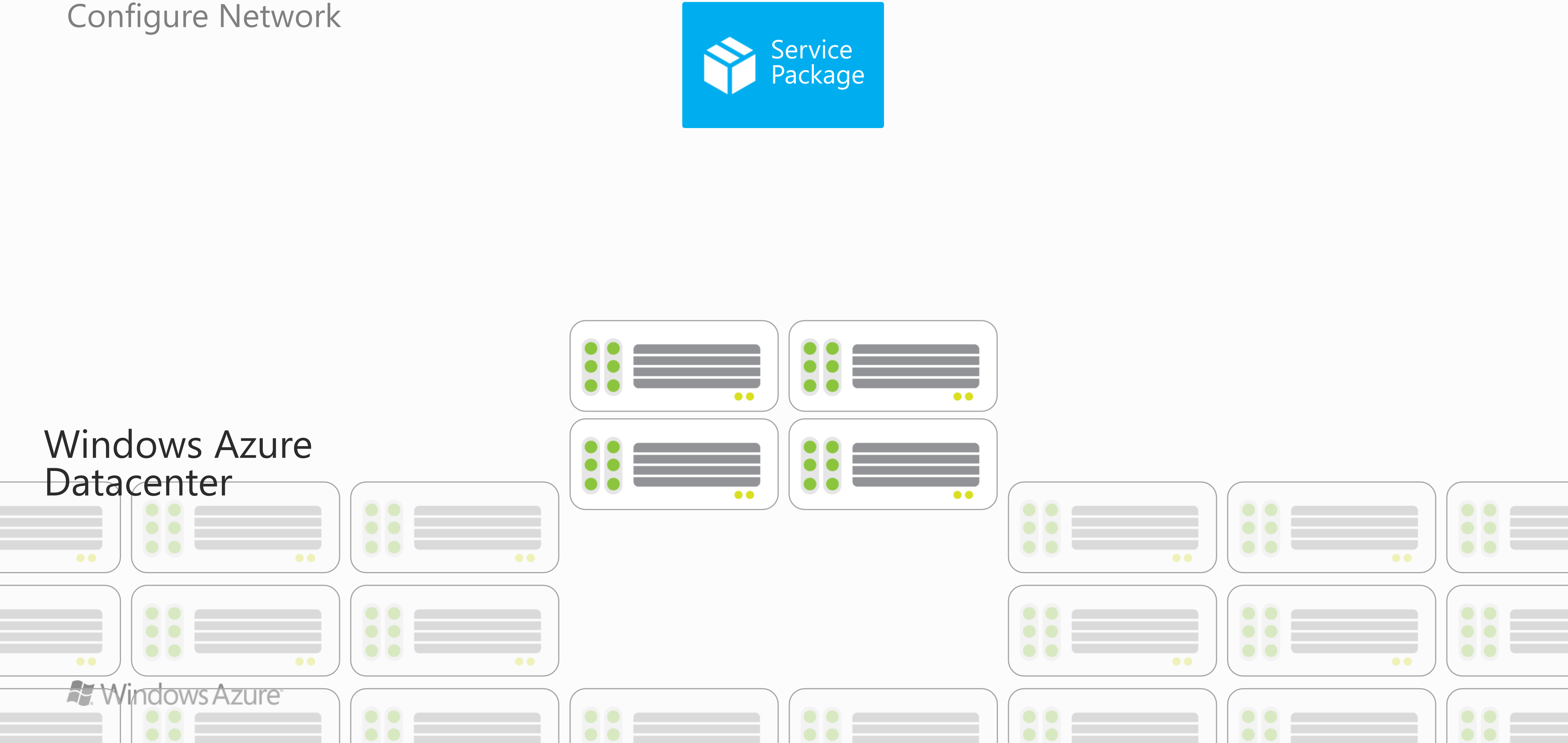
Configure Network



Provision Role Instances

Deploy App Code

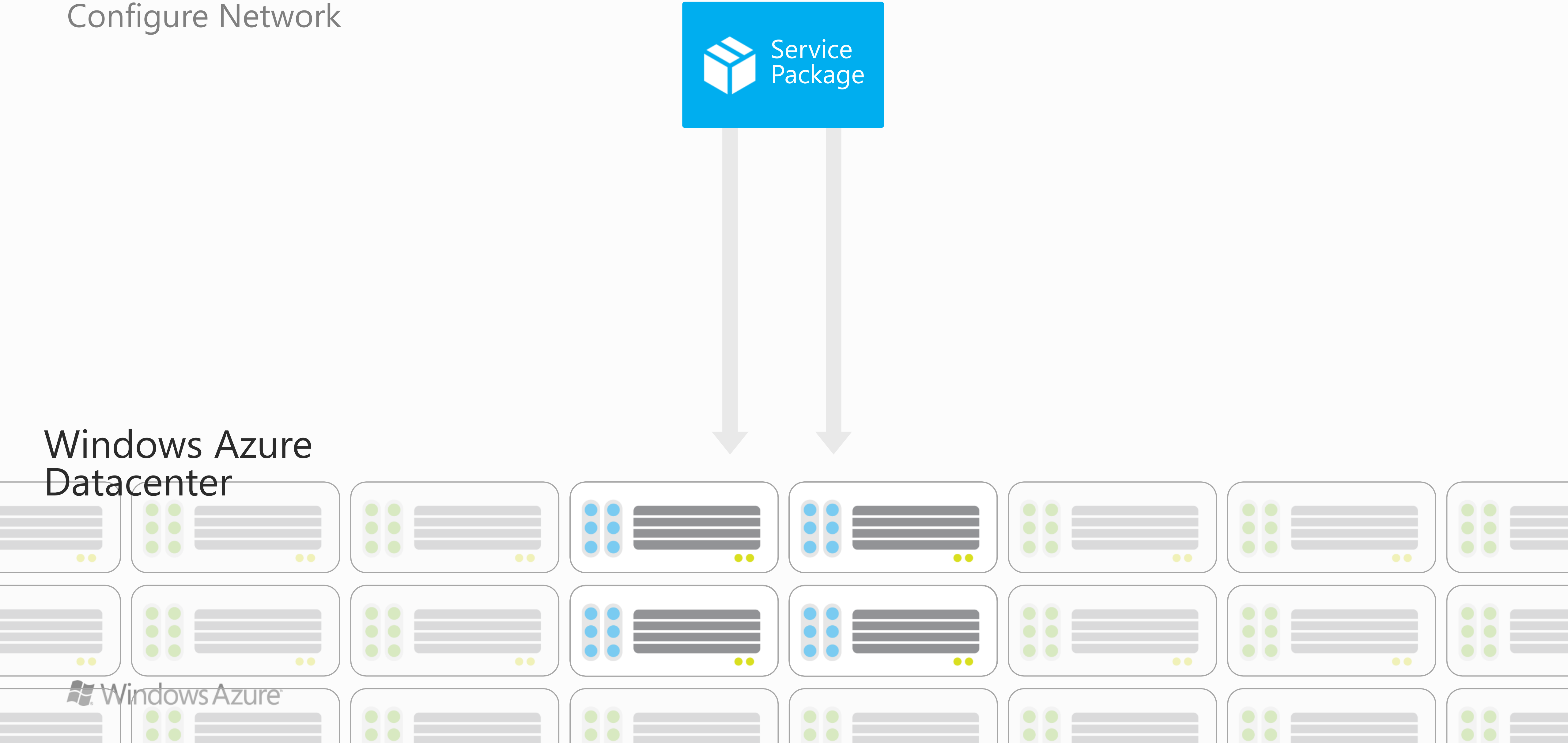
Configure Network



Provision Role Instances

Deploy App Code

Configure Network

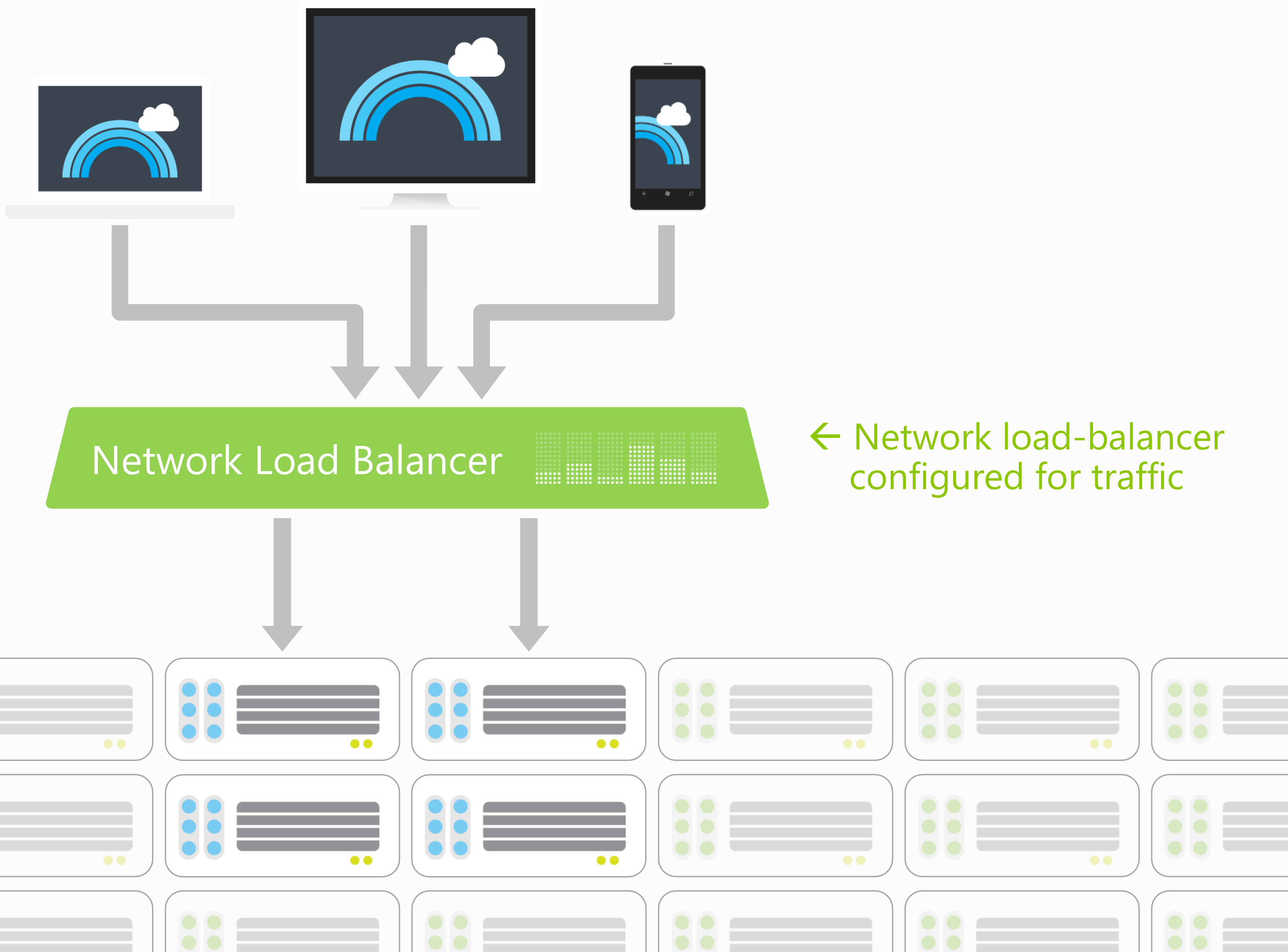


Windows Azure

Provision Role Instances

Deploy App Code

Configure Network



← Network load-balancer configured for traffic

Windows Azure Datacenter

Windows Azure

Application building blocks

Big data

Database

Storage

Traffic

Caching

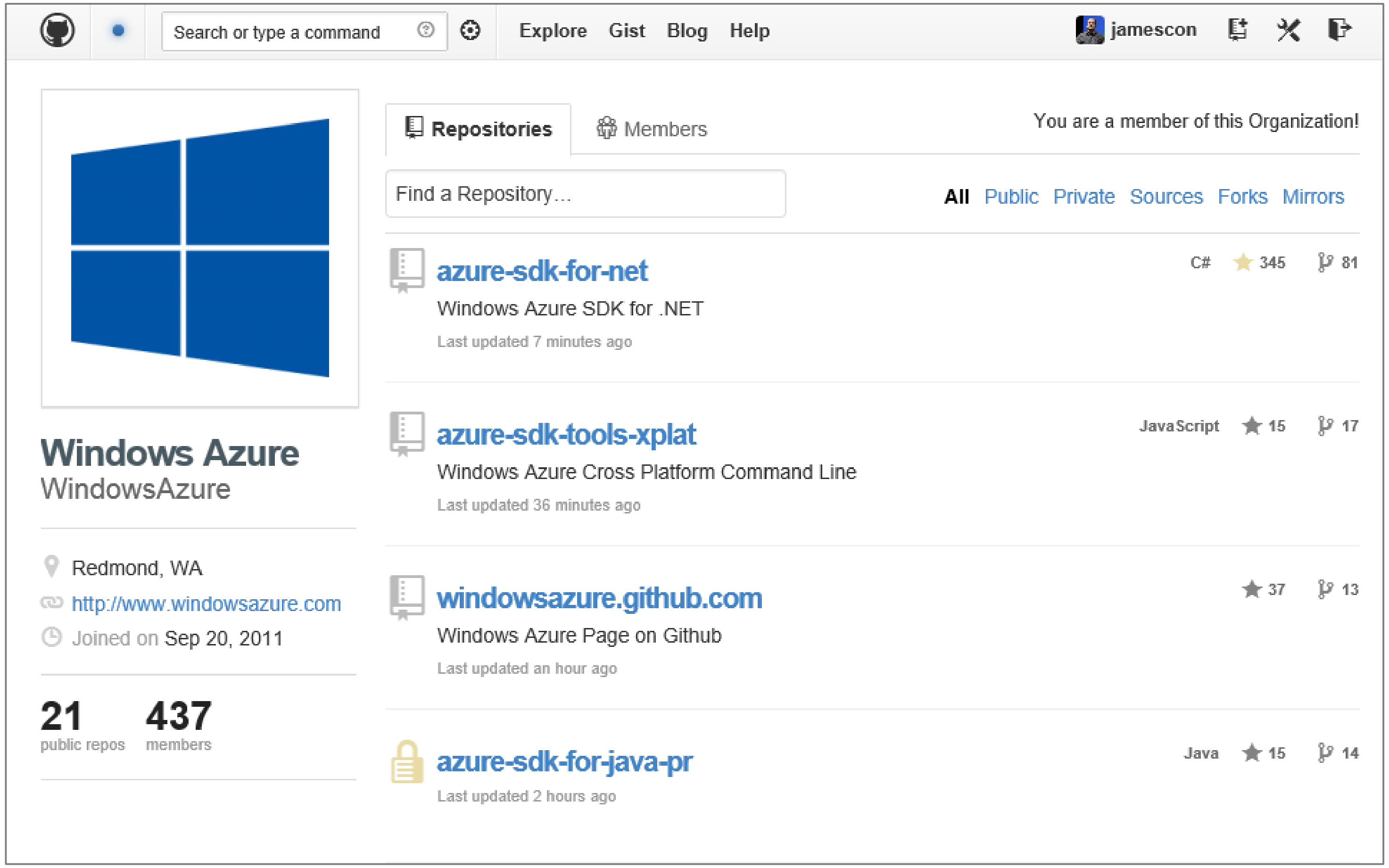
Messaging

Identity

Media

CDN

Networking



You are a member of this Organization!

Windows Azure
WindowsAzure

Redmond, WA
http://www.windowsazure.com
Joined on Sep 20, 2011

21 public repos
437 members

Repository	Description	Last updated	Language	Stars	Forks
azure-sdk-for-net	Windows Azure SDK for .NET	Last updated 7 minutes ago	C#	345	81
azure-sdk-tools-xplat	Windows Azure Cross Platform Command Line	Last updated 36 minutes ago	JavaScript	15	17
windowsazure.github.com	Windows Azure Page on Github	Last updated an hour ago		37	13
azure-sdk-for-java-pr		Last updated 2 hours ago	Java	15	14

Open source

→ http://github.com/windowsazure

Windows Azure Scenarios

Ideal for Applications Needing:

Scalability

Availability

Fault Tolerance

Common Application Uses:

Web Sites

Compute Intensive apps

Device Applications

Web APIs

Social Games

89 countries and territories

Australia
Austria
Belgium
Brazil
Canada
Chile
Colombia
Costa Rica
Cyprus
Czech Republic
Denmark
Finland
France
Germany
Greece
Hong Kong
Hungary
India
Ireland
Israel
Italy
Japan
Korea

Luxembourg
Malaysia
Mexico
Netherlands
New Zealand
Norway
Peru
Philippines
Poland
Portugal
Puerto Rico
Romania
Russia
Singapore
Spain
Sweden
Switzerland
Trinidad & Tobago
UK
United States
New Countries:
Algeria

Argentina
Belarus
Bulgaria
Croatia
Dominican Rep
Ecuador
Egypt
El Salvador
Estonia
Guatemala
Iceland
Indonesia
Jordan
Kazakhstan
Kenya

Kuwait
Latvia
Liechtenstein
Lithuania
Macedonia
Malta
Montenegro
Morocco
Azerbaijan
Nigeria
Oman
Pakistan
Panama
Paraguay
Qatar

Saudi Arabia
Serbia
Slovakia
Slovenia
South Africa
Sri Lanka
Taiwan
Thailand
Tunisia
Turkey
UAE
Ukraine
Uruguay
Venezuela
Bahrain

Windows Azure

Flexible

<3

Open

Solid

Summary

Windows Azure provides a comprehensive set of services that you can selectively compose to build your cloud apps

Global Data Center Footprint

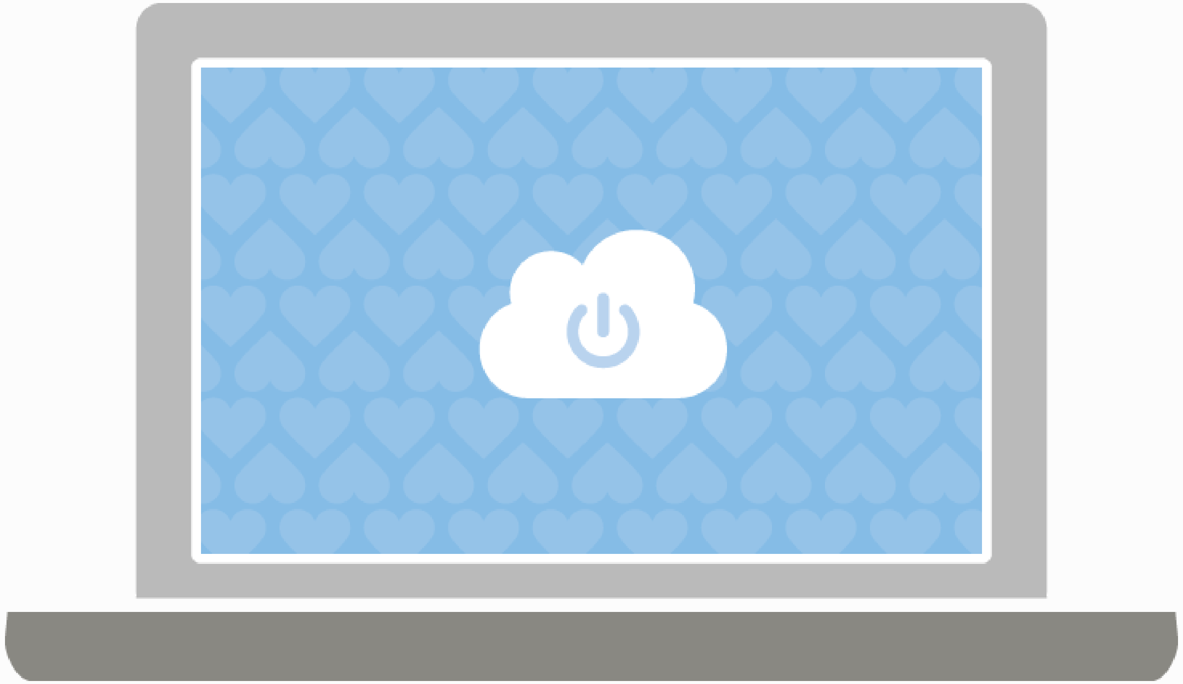
99.95% Monthly SLA. Pay only for what you use.

Flexible & Open Compute Options

Virtual Machines, Web Sites, & Cloud Services

Managed Building Block Services

SQL Database, Cache, Service Bus, & more



References

- http://pytools.codeplex.com
- Windows Azure Python SDK
- Windows Azure
- How to use Service Management from Python
- http://research.microsoft.com/en-us/projects/azure/

Microsoft®

© 2011 Microsoft Corporation. All rights reserved. Microsoft, Windows, Windows Vista and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries.

The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.