

City University of New York (CUNY)

## CUNY Academic Works

---

Open Educational Resources

Hunter College

---

2020

### CSCI 49378: Lecture 7: Cloud Systems and Infrastructures I

Bonan Liu  
*CUNY Hunter College*

NYC Tech-in-Residence Corps

[How does access to this work benefit you? Let us know!](#)

More information about this work at: [https://academicworks.cuny.edu/hc\\_oers/18](https://academicworks.cuny.edu/hc_oers/18)

Discover additional works at: <https://academicworks.cuny.edu>

---

This work is made publicly available by the City University of New York (CUNY).  
Contact: [AcademicWorks@cuny.edu](mailto:AcademicWorks@cuny.edu)

# Cloud Systems and Infrastructures I

Bonan Liu

Tech-In-Residence Member, Hunter College, CUNY

Spring 2020



# Disclaimer

*The content of this presentation is being provided for educational and informational purposes only. The views, thoughts, and opinions expressed in this presentation belong solely to the author, and not necessarily to the author's employer.*

*The content of this presentation is not endorsed by the author's employer.*

- Use small instances whenever possible (f1 type)
- Available credits
  - GCP free tier services (near zero cost)
  - \$300 free credit if you are a new user
  - \$50 credit via Student Coupon Retrieval Link
- As a reference, I spent less than \$30/month for my hobby projects. (\$10+ for most of the months)

- Review Assignment 2
- IaaS: Google Compute Engine (GCE)
- Elastic Computing
- PaaS: Google App Engine (GAE)
- Potential Maintenance Tasks

- What makes a good design document?
  - A design document makes engineering decisions
  - A good design document manages the complexity of the system
  - A good design document explains and solves difficult problems clearly and concisely

- Improve your design document
  - Use diagrams
  - Use tables
  - Use examples
  - Use data
  - Use references

- Lifecycle of a VM
- GCE Concepts
  - Virtual Machine
  - Persistent Disk
  - Snapshot
  - Images
- Sole-Tenant Node



- Unmanaged Solutions
  - Instance Groups
  - Health Check
- Managed Solutions
  - Auto-Scale from template
  - Bidirectional Scaling or No Scale-down

Demo: Create a load balancer.

Practice: Create a small web application and instance group. Create a load balancer for it.

If you have enough time, try to kill one of your instance and see what happened.

# IaaS: Google Compute Engine (GCE)

```
from http.server import HTTPServer, BaseHTTPRequestHandler
import socket

class MyHandler(BaseHTTPRequestHandler):
    def do_GET(self):
        self.send_response(200)
        self.end_headers()
        self.wfile.write(str.encode('This is instance ' + socket.gethostname()))

httpd = HTTPServer(('', 80), MyHandler)
httpd.serve_forever()
```

- Google App Engine Types
  - Standard Environment
  - Flexible Environment
- Cloud Datastore: Google App Engine Storage
- Structure:
  - YAML file
  - Program in any language
  - Static file

## Potential Maintenance Tasks

- Service Rollout
  - Incrementally Split Traffic
- Cron Jobs
  - Periodically Cleanup
  - Batch Processing
- Monitoring
- Data Management

Let's revisit the Tiny URL application. What is the proposed architecture? How to implement it with Cloud services?

Demo: Tiny URL application.



Practice: Create an application to return today's date.

Useful method:

```
from datetime import date
today = date.today()
today.strftime("%m/%d/%y")
```

If you have enough time, create an application which serves POST request.

# GAE Demo

```
import webapp2

class MyPage(webapp2.RequestHandler):
    def get(self):
        self.response.headers['Content-Type'] = 'text/plain'
        self.response.write(
            'This is GAE demo! You are requesting: ' + self.request.path)

app = webapp2.WSGIApplication([
    ('/*.*', MyPage),
])
```

# GAE Demo

```
import webapp2

URL_MAPPING = {
    "/short1": "http://www.cnn.com",
    "/short2": "http://www.nyt.com",
    "/short3": "http://www.fox.com"
}

class MyPage(webapp2.RequestHandler):
    def get(self):
        self.response.headers['Content-Type'] = 'text/plain'
        if self.request.path in URL_MAPPING:
            self.redirect(URL_MAPPING[self.request.path], permanent=True)
            return
        self.response.write('The tiny URL %s is not found.' % self.request.path)

app = webapp2.WSGIApplication([
    ('/*', MyPage),
])
```

*PaaS is not middleware over IaaS. Reza Shafii. Jan 26, 2012.*

*Understanding the cloud computing stack: SaaS, PaaS, IaaS.*

*<https://support.rackspace.com/how-to/understanding-the-cloud-computing-stack-saas-paas-iaas/>*