



TCSS 562: SOFTWARE ENGINEERING FOR CLOUD COMPUTING

Cloud Computing:
Intro to Cloud Computing
Cloud Delivery Models

Wes J. Lloyd
School of Engineering and Technology
University of Washington - Tacoma



1

FEEDBACK FROM 10/15

- What is vertical scaling in the cloud?

October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.2

2

REVIEW – 10/15

- What is the definition of Cloud Computing?
- How is capacity planning different in the cloud vs. with traditional server infrastructure?
- What is Cluster computing?
- What is Grid computing?
- What is Virtualization?
- What is the difference between Horizontal and Vertical scaling?

October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.3

3

OBJECTIVES - 2

- Term Project Proposal (10/19)
- From: Cloud Computing Concepts, Technology & Architecture:
- Introduction to Cloud Computing
 - Why study cloud computing?
 - History of cloud computing
 - Business drivers
 - Cloud enabling technologies
 - Terminology
 - Benefits of cloud adoption
 - Risks of cloud adoption

October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.4

4

KEY TERMINOLOGY

- **On-Premise Infrastructure**
 - Local server infrastructure not configured as a cloud
- **Cloud Provider**
 - Corporation or private organization responsible for maintaining cloud
- **Cloud Consumer**
 - User of cloud services
- **Scaling**
 - **Vertical scaling**
 - Scale up: increase resources of a single virtual server
 - Scale down: decrease resources of a single virtual server
 - **Horizontal scaling**
 - Scale out: increase number of virtual servers
 - Scale in: decrease number of virtual servers

October 17, 2018

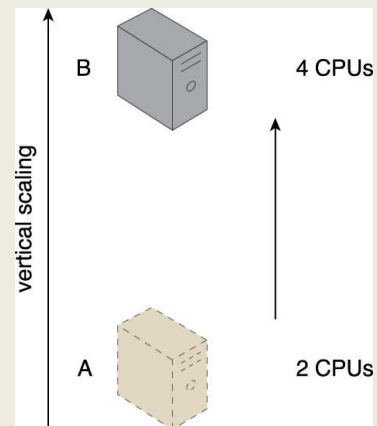
TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.5

5

VERTICAL SCALING

- **Reconfigure virtual machine to have different resources:**
 - CPU cores
 - RAM
 - HDD/SDD capacity
- **May require VM migration if physical host machine resources are exceeded**



October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.6

6

HORIZONTAL SCALING

- Increase (scale-out) or decrease (scale-in) number of virtual servers based on demand

Diagram illustrating Horizontal Scaling:

At the top, two **pooled physical servers** are shown. Arrows point from these physical servers to a sequence of **virtual servers**.

The sequence of virtual servers shows the process of scaling out:

- Initially, there is one virtual server labeled **A**.
- When **demand** increases, a second virtual server labeled **A** is added, resulting in two servers labeled **A** and **B**.
- When **demand** increases further, a third virtual server labeled **C** is added, resulting in three servers labeled **A**, **B**, and **C**.

A long arrow at the bottom, labeled **horizontal scaling**, points from left to right, indicating the direction of scaling out (increasing the number of servers).

October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.7

7

HORIZONTAL VS VERTICAL SCALING

Horizontal Scaling	Vertical Scaling
Less expensive using commodity HW	Requires expensive high capacity servers

October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.8

8

HORIZONTAL VS VERTICAL SCALING

Horizontal Scaling	Vertical Scaling
Less expensive using commodity HW	Requires expensive high capacity servers
IT resources instantly available	IT resources typically instantly available

October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.9

9

HORIZONTAL VS VERTICAL SCALING

Horizontal Scaling	Vertical Scaling
Less expensive using commodity HW	Requires expensive high capacity servers
IT resources instantly available	IT resources typically instantly available
Resource replication and automated scaling	Additional setup is normally needed

October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.10

10

HORIZONTAL VS VERTICAL SCALING

Horizontal Scaling	Vertical Scaling
Less expensive using commodity HW	Requires expensive high capacity servers
IT resources instantly available	IT resources typically instantly available
Resource replication and automated scaling	Additional setup is normally needed
Additional servers required	No additional servers required

October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.11

11

HORIZONTAL VS VERTICAL SCALING

Horizontal Scaling	Vertical Scaling
Less expensive using commodity HW	Requires expensive high capacity servers
IT resources instantly available	IT resources typically instantly available
Resource replication and automated scaling	Additional setup is normally needed
Additional servers required	No additional servers required
Not limited by individual server capacity	Limited by individual server capacity

October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.12

12

KEY TERMINOLOGY - 2

- Cloud services
 - Broad array of resources accessible “as-a-service”
 - Categorized as Infrastructure (IaaS), Platform (PaaS), Software (SaaS)
- Service-level-agreements (SLAs):
 - Establish expectations for: uptime, security, availability, reliability, and performance

October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.13

13

GOALS AND BENEFITS

- Cloud providers
 - Leverage economies of scale through mass-acquisition and management of large-scale IT resources
 - Locate datacenters to optimize costs where electricity is low
- Cloud consumers
 - Key business/accounting difference:
 - Cloud computing enables anticipated capital expenditures to be replaced with operational expenditures
 - Operational expenditures always scale with the business
 - Eliminates need to invest in server infrastructure based on anticipated business needs
 - Businesses become more agile and lower their financial risks by eliminating large capital investments in physical infrastructure

October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.14

14

CLOUD BENEFITS - 2

- On demand access to pay-as-you-go resources on a short-term basis (less commitment)
- Ability to acquire “unlimited” computing resources on demand when required for business needs
- Ability to add/remove IT resources at a fine-grained level
- Abstraction of server infrastructure so applications deployments are not dependent on specific locations, hardware, etc.
 - The cloud has made our software deployments more agile...



October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.15

15

CLOUD BENEFITS - 3

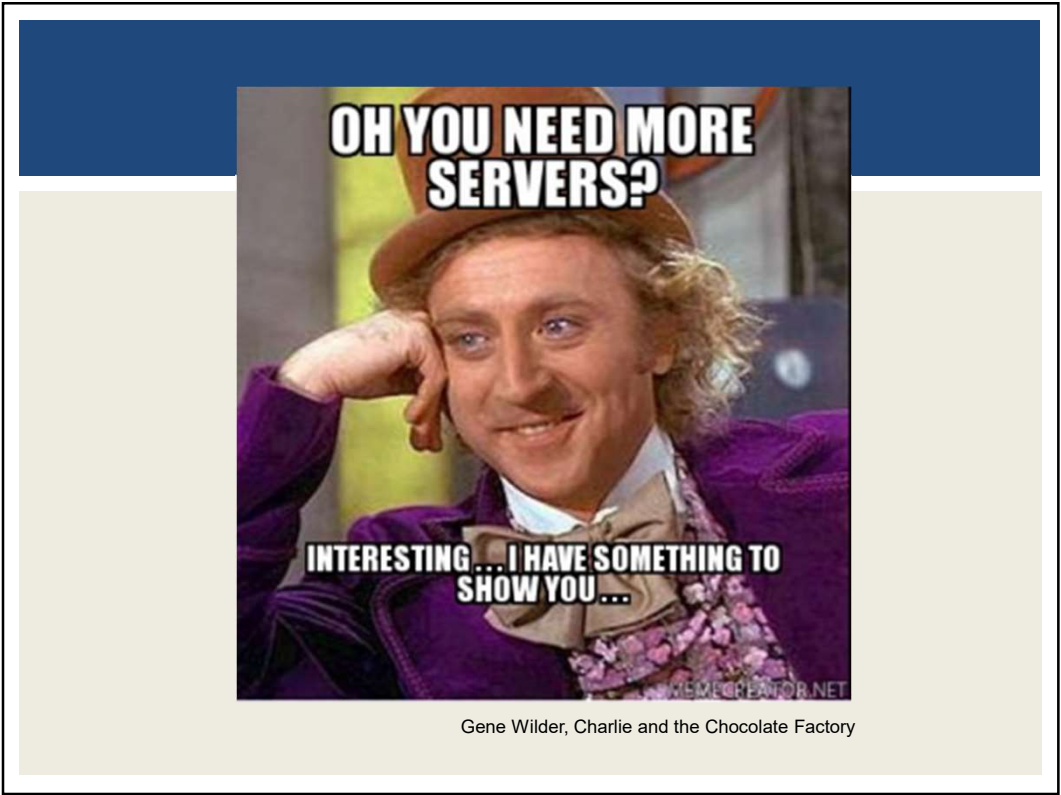
- Example: Using 100 servers for 1 hour costs the same as using 1 server for 100 hours
- Rosetta Protein Folding: Working with a UW-Tacoma graduate student, we recently deployed this science model across 5,900 compute cores on Amazon for 2-days...
- *What is the cost to purchase 5,900 compute cores?*
- Recent Dell Server purchase example:
20 cores on 2 servers for \$4,478...
- Using this ratio 5,900 cores costs \$1.3 million (purchase only)

October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.16

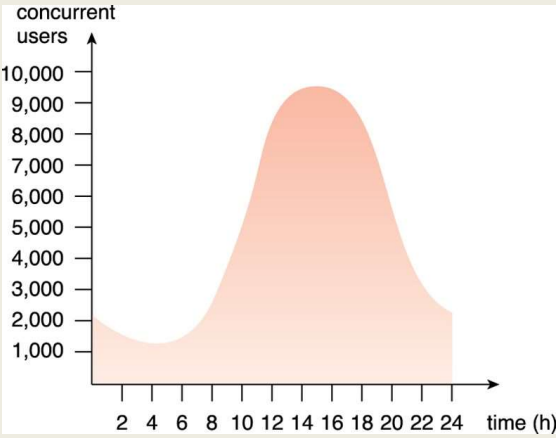
16



17

CLOUD BENEFITS

- Increased scalability
 - Example demand over a 24-hour day →
- Increased availability
- Increased reliability



Time (h)	Concurrent Users
2	1,500
4	1,200
6	1,500
8	2,500
10	5,000
12	8,000
14	9,000
16	9,500
18	8,000
20	4,000
22	2,000
24	1,500

October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.18

18

CLOUD ADOPTION RISKS

- **Increased security vulnerabilities**
 - Expansion of trust boundaries now include the external cloud
 - Security responsibility shared with cloud provider
- **Reduced operational governance / control**
 - Users have less control of physical hardware
 - Cloud user does not directly control resources to ensure quality-of-service
 - Infrastructure management is abstracted
 - Quality and stability of resources can vary
 - Network latency costs and variability

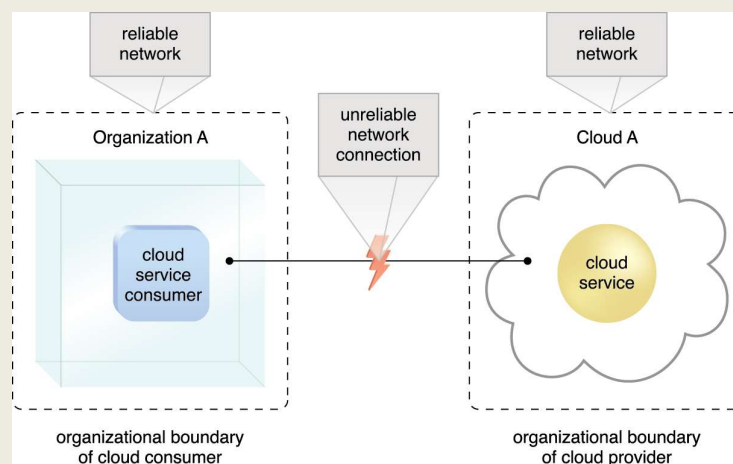
October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.19

19

NETWORK LATENCY COSTS



October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.20

20

CLOUD RISKS - 2

- **Performance monitoring of cloud applications**
 - Cloud metrics (AWS cloudwatch) support monitoring cloud infrastructure (network load, CPU utilization, I/O)
 - Performance of cloud applications depends on the health of aggregated cloud resources working together
 - User must monitor this aggregate performance
- **Limited portability among clouds**
 - Early cloud systems have significant “vendor” lock-in
 - Common APIs and deployment models are slow to evolve
 - Operating system containers help make applications more portable, but containers still must be deployed
- **Geographical issues**
 - Abstraction of cloud location leads to legal challenges with respect to laws for data privacy and storage

October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.21

21

CLOUD: VENDOR LOCK-IN

The diagram illustrates the concept of vendor lock-in in cloud computing. It shows a cloud consumer (represented by a computer icon) interacting with two different cloud providers: Cloud A (Cloud Provider X) and Cloud B (Cloud Provider Y). Cloud A is shown with a cloud icon containing server and storage symbols, and a label indicating it 'supports message encryption and digital signatures'. Cloud B is shown with a similar cloud icon and a label indicating it 'supports message encryption only'. The cloud consumer is shown with a label indicating it 'requires encryption and digital signing of messages'. A lightning bolt symbol is placed between the consumer and Cloud B, indicating a barrier or incompatibility. Arrows show the consumer sending data to Cloud A and receiving data back, and a similar path to Cloud B, but the lack of full support in Cloud B creates a dependency on Cloud A's specific capabilities.

October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.22

22

OBJECTIVES

- From: Cloud Computing Concepts, Technology & Architecture:
- Cloud Computing Concepts and Models
 - Roles and boundaries
 - Cloud characteristics
 - Cloud delivery models
 - Cloud deployment models

October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.23

23

ROLES

- Cloud provider
 - Organization that provides cloud-based resources
 - Responsible for fulfilling SLAs for cloud services
 - Some cloud providers “resell” IT resources from other cloud providers
 - Example: Heroku sells PaaS services running atop of Amazon EC2
- Cloud consumers
 - Cloud users that consume cloud services
- Cloud service owner
 - Both cloud providers and cloud consumers can own cloud services
 - A cloud service owner may use a cloud provider to provide a cloud service (e.g. Heroku)

October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.24

24

ROLES - 2

- **Cloud resource administrator**
 - Administrators provide and maintain cloud services
 - Both cloud providers and cloud consumers have administrators
- **Cloud auditor**
 - Third-party which conducts independent assessments of cloud environments to ensure security, privacy, and performance.
 - Provides unbiased assessments
- **Cloud brokers**
 - An intermediary between cloud consumers and cloud providers
 - Provides service aggregation
- **Cloud carriers**
 - Network and telecommunication providers which provide network connectivity between cloud consumers and providers

October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.25

25

ORGANIZATION BOUNDARY

The diagram illustrates the concept of organizational boundaries in cloud computing. It features two dashed-line boxes. The left box, titled 'Organization A', contains a blue 3D cube representing an organization, with a smaller blue rounded rectangle inside labeled 'cloud service consumer'. The right box, titled 'Cloud A', contains a yellow cloud shape representing a cloud service, with a smaller yellow circle inside labeled 'cloud service'. Both boxes are labeled 'organizational boundary' at the bottom.

October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.26

26

TRUST BOUNDARY

The diagram illustrates a 'trust boundary' as a large orange rectangle. Inside this boundary are two separate 'organizational boundaries', each represented by a dashed orange line. The left organizational boundary is labeled 'Organization A' and contains a light blue 3D cube with a blue rounded rectangle inside labeled 'cloud service consumer'. The right organizational boundary is labeled 'Cloud A' and contains a white cloud shape with a yellow circle inside labeled 'cloud service'. Labels 'trust boundary' and 'organizational boundary' are placed near their respective boundaries.

October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.27

27

OBJECTIVES

- From: Cloud Computing Concepts, Technology & Architecture:
- Cloud Computing Concepts and Models
 - Roles and boundaries
 - **Cloud characteristics**
 - Cloud delivery models
 - Cloud deployment models

October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.28

28

CLOUD CHARACTERISTICS

- On-demand usage
 - Ubiquitous access
 - Multitenancy (resource pooling)
 - Elasticity
 - Measured usage
 - Resiliency
-
- Assessing these features helps measure the value offered by a given cloud service or platform

October 17, 2018

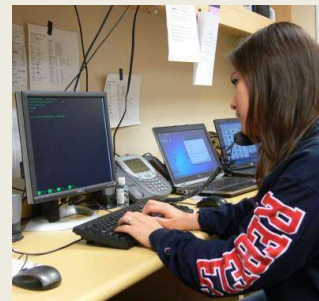
TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.29

29

ON-DEMAND USAGE

- The freedom to self-provision IT resources
- Generally with automated support
- Automated support requires no human involvement
- Automation through software services interface



October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.30

30

UBIQUITOUS ACCESS

- Cloud services are widely accessible
- Public cloud: internet accessible
- Private cloud: throughout segments of a company's intranet
- 24/7 availability

October 17, 2018	TCSS562: Software Engineering for Cloud Computing [Fall 2018] School of Engineering and Technology, University of Washington - Tacoma	L7.31
------------------	--	-------

31

MULTITENANCY

- Cloud providers pool resources together to share them with many users
- Serve multiple cloud service consumers
- IT resources can be dynamically assigned, reassigned based on demand
- Multitenancy can lead to performance variation

October 17, 2018	TCSS562: Software Engineering for Cloud Computing [Fall 2018] School of Engineering and Technology, University of Washington - Tacoma	L7.32
------------------	--	-------

32

SINGLE TENANT MODEL

The diagram illustrates the Single Tenant Model. It features a large cloud shape containing two separate service paths. On the left, 'Cloud Service Consumer A' (blue box) connects to 'Cloud Service A' (yellow circle), which then connects to 'Cloud Storage Device A' (green cylinder). On the right, 'Cloud Service Consumer B' (blue box) connects to 'Cloud Service B' (yellow circle), which then connects to 'Cloud Storage Device B' (green cylinder). This represents a dedicated infrastructure for each tenant.

October 17, 2018	TCSS562: Software Engineering for Cloud Computing [Fall 2018] School of Engineering and Technology, University of Washington - Tacoma	L7.33
------------------	--	-------

33

MULTITENANT MODEL

- Resource is “multiplexed” and share amongst multiple users
- Goal is to increase utilization
- Often server resources are underutilized
- There are many “sunk costs” whether usage is 0% or 100%
- Cloud computing tries to maximize “sunk cost” investments

The diagram illustrates the Multitenant Model. It features a large cloud shape containing two service paths. On the left, 'Cloud Service Consumer A' (blue box) connects to 'Cloud Service A' (yellow circle). On the right, 'Cloud Service Consumer B' (blue box) connects to 'Cloud Service B' (yellow circle). Both 'Cloud Service A' and 'Cloud Service B' have arrows pointing to a single, shared green cylinder labeled 'shared cloud storage device' with a leader line. This represents shared infrastructure across multiple tenants.

October 17, 2018	TCSS562: Software Engineering for Cloud Computing [Fall 2018] School of Engineering and Technology, University of Washington - Tacoma	L7.34
------------------	--	-------

34

MULTITENANT DATABASE

Isolated

Separate database

E1

Semi-shared

Shared database
Separate schema

E2

Shared

Shared database
Shared schema

E3

October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.35

35

MULTITENANCY OF RESOURCES

■ Where is the multitenancy?

Traditional On Premise

Single Tenant (Hosted)

Multi-Tenant

Virtual Appliance

October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.36

36

ELASTICITY

- Automated ability of cloud to transparently scale resources
- Scaling based on runtime conditions or pre-determined by cloud consumer or cloud provider
- Threshold based scaling
 - CPU-utilization > threshold_A, Response_time > 100ms
 - Application agnostic vs. application specific thresholds
 - Why might an application agnostic threshold be non-ideal?
- Load prediction
 - Historical models
 - Real-time trends

October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

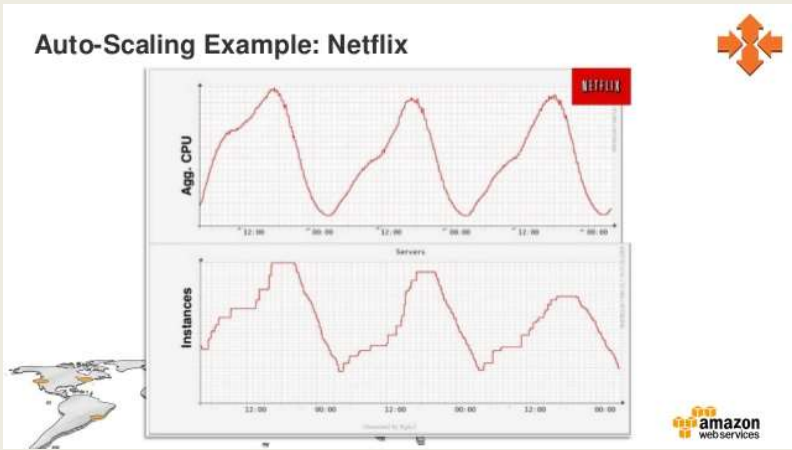
L7.37

37

PREDICTABLE DEMAND

- Example:

Auto-Scaling Example: Netflix



October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.38

38

MEASURED USAGE

- Cloud platform tracks usage of IT resources
- For billing purposes
- Enables charging only for IT resources actually used
- Can be time-based (minute, hour, day)
- Can be throughput-based (MB, GB)

- Not all measurements are for billing
- Some measurements can support auto-scaling
- For example CPU utilization

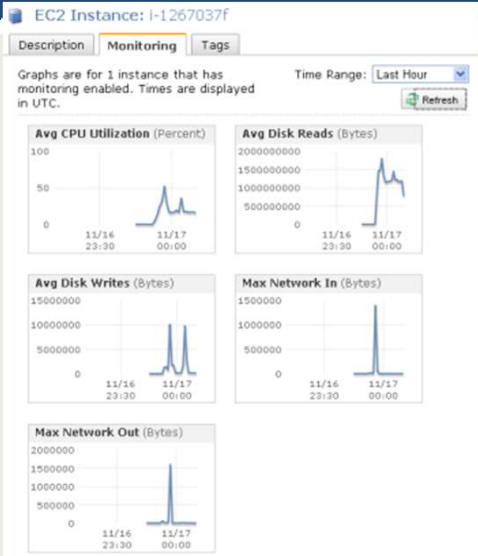
October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.39

39

EC2 CLOUDWATCH METRICS



EC2 Instance: i-1267037f

Graphs are for 1 instance that has monitoring enabled. Times are displayed in UTC.

Time Range: Last Hour

Refresh

Avg CPU Utilization (Percent)

Avg Disk Reads (Bytes)

Avg Disk Writes (Bytes)

Max Network In (Bytes)

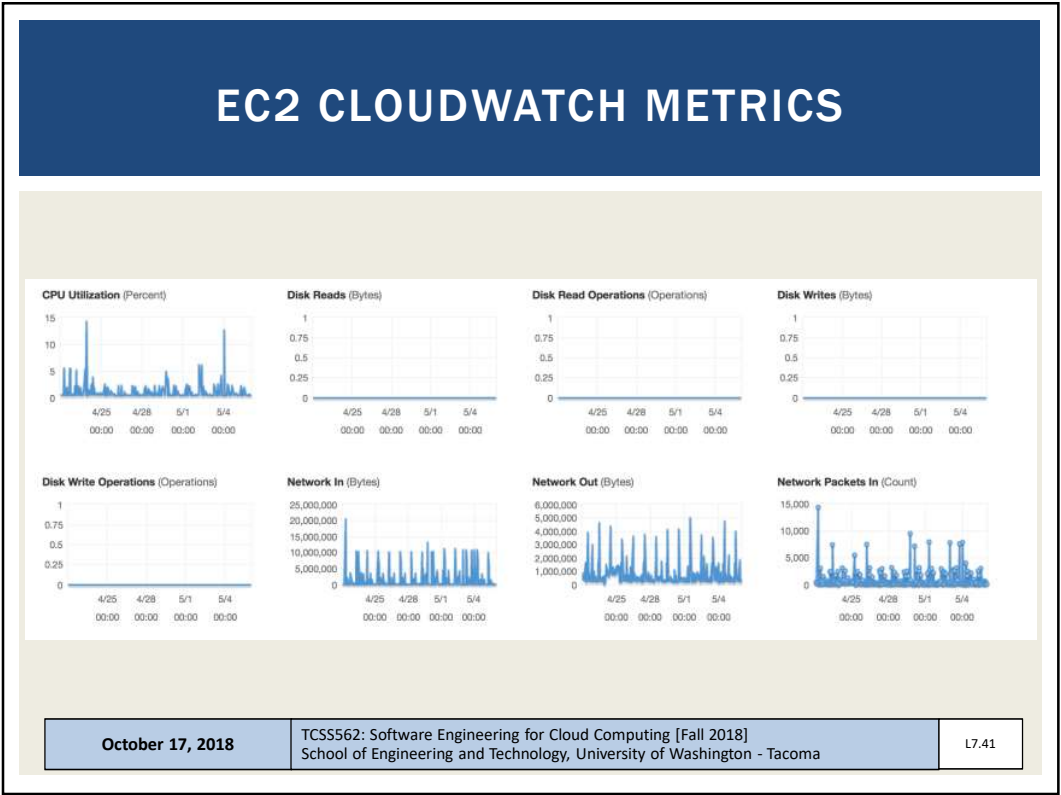
Max Network Out (Bytes)

October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.40

40



41

RESILIENCY

- Distributed redundancy across physical locations
- Used to improve reliability and availability of cloud-hosted applications
- Very much an engineering problem
- No “resiliency-as-a-service” for user deployed apps
- Unique characteristics of user applications make a one-size fits all service solution challenging

October 17, 2018	TCSS562: Software Engineering for Cloud Computing [Fall 2018] School of Engineering and Technology, University of Washington - Tacoma	L7.42
------------------	--	-------

42

OBJECTIVES

- From: Cloud Computing Concepts, Technology & Architecture:
- Cloud Computing Concepts and Models
 - Roles and boundaries
 - Cloud characteristics
 - **Cloud delivery models**
 - Cloud deployment models

October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.43

43

CLOUD DELIVERY MODELS



- What is the appropriate level of abstraction?
- How should applications be deployed?
 - IaaS, PaaS, SaaS, DbaaS, FaaS
- How do we ensure Quality-of-Service?
 - Performance, Availability, Responsiveness, Fault Tolerance
- How is scalability provided?
- How do we minimize hosting costs?
 - How do we estimate hosting costs?

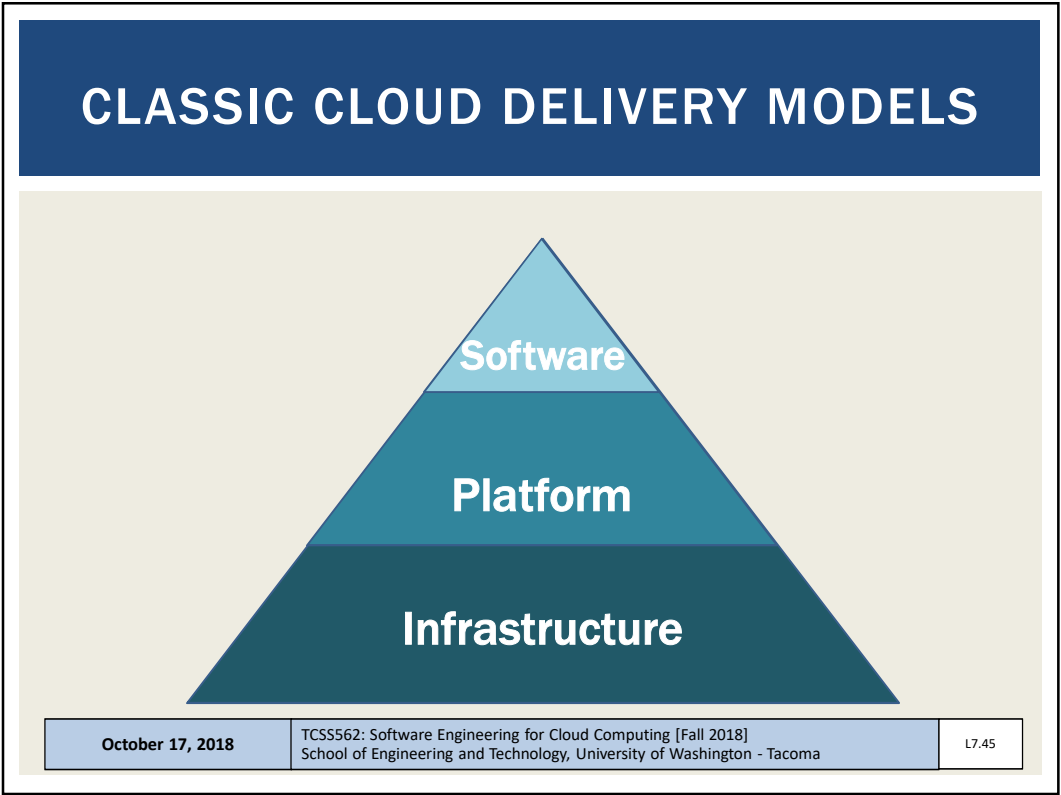


October 17, 2018

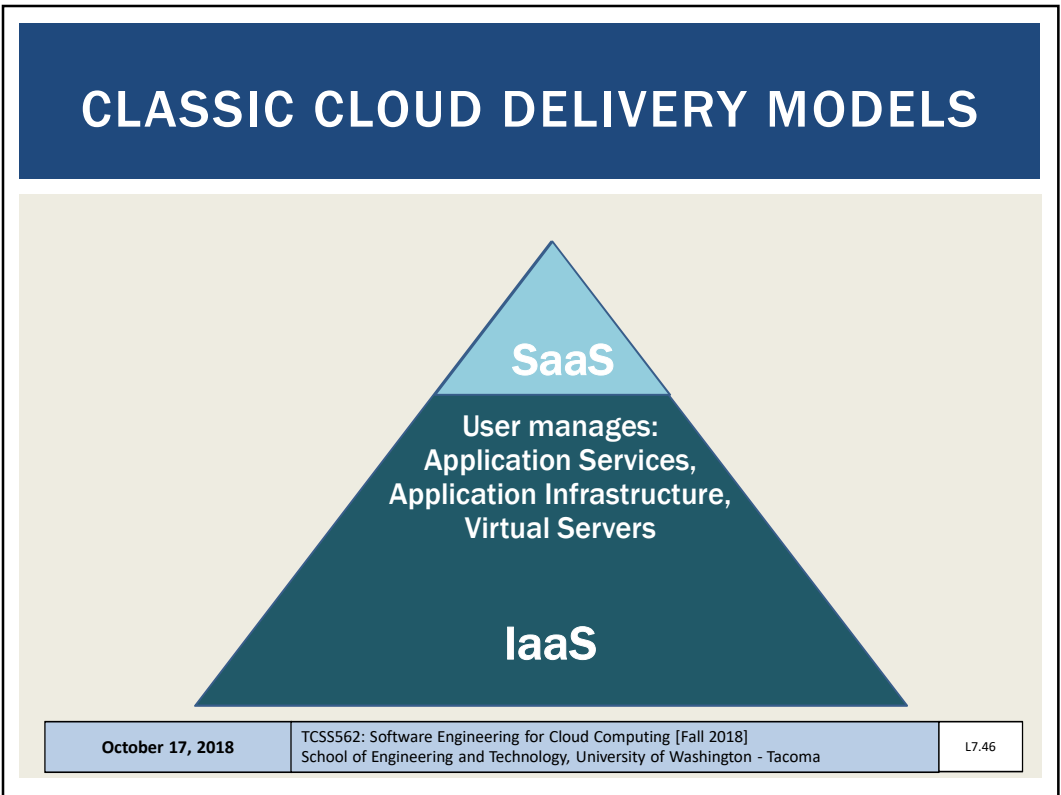
TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.44

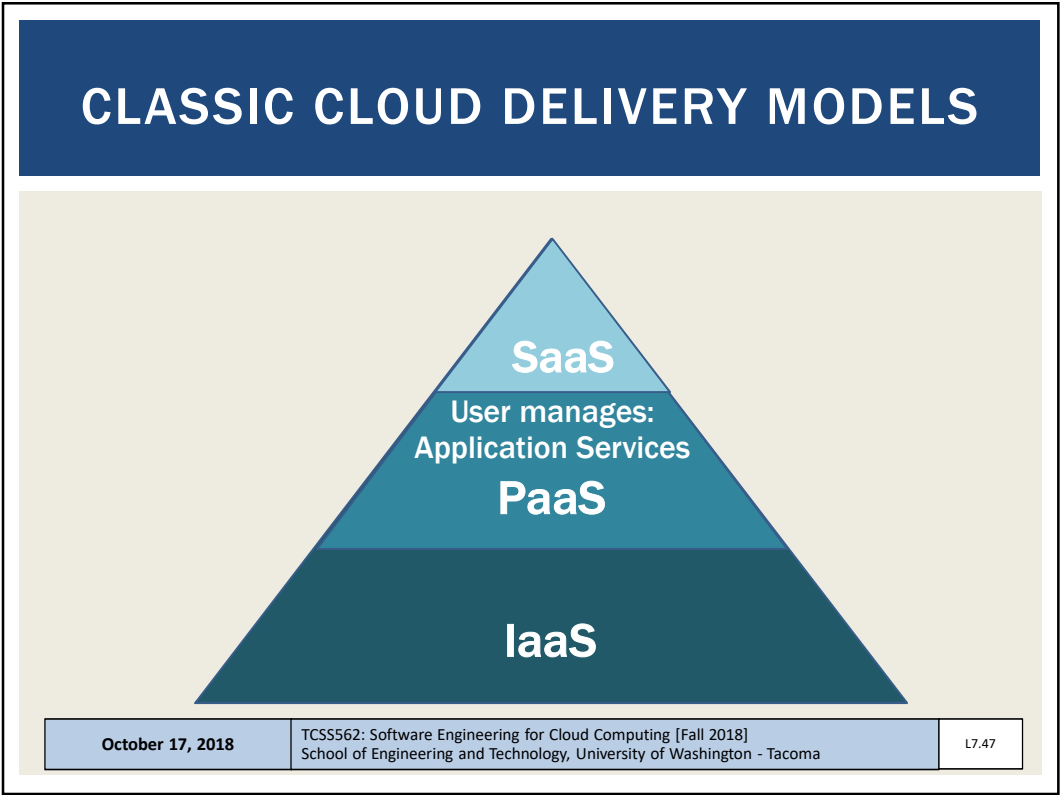
44



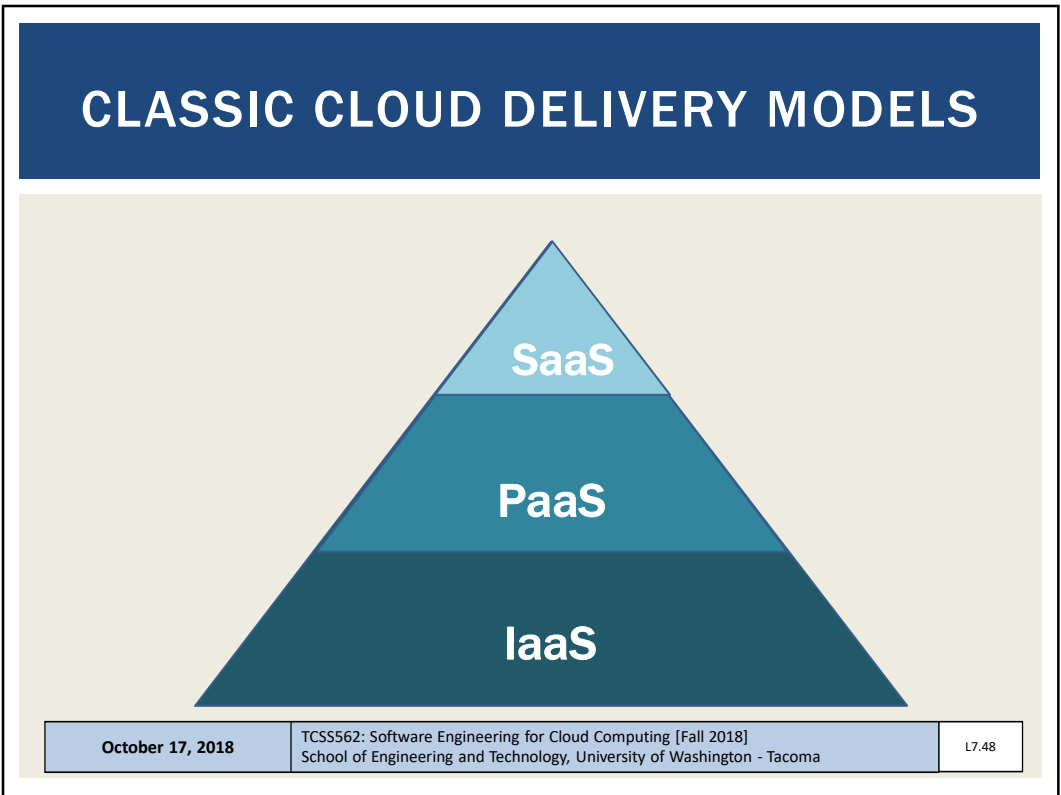
45



46



47



48

EXAMPLE CLOUD SERVICES

SAAS

Software as a Service

Email

CRM

Collaborative

ERP

CONSUME

PAAS

Platform as a Service

Application Development

Decision Support

Web

Streaming

BUILD ON IT

IAAS

Infrastructure as a Service

Caching

Legacy

Networking

Security

File

Technical

System Mgmt

MIGRATE TO IT

October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.49

49

END USER APPLICATIONS

Many different "cloud" providers

Software-as-a-Service

Finance & Accounting

Content Management

Vertical

Enterprise Social Media

Marketing Analytics

Retail & E-Commerce

Collaboration

Business Intelligence

Ad Tech

Many cloud providers are also cloud consumers

Cloud Foundry

Parse

twilio

github

bladelogic

heroku

piston

cloudfoundry

apptio

perimeter

ibm

amazon.com

rackspace

salesforce.com

ORACLE

GOGRID

parallels

terremark

savvis

hp

vmware

joyent

dynamicops

windows azure

VERTICA

October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.50

50

INFRASTRUCTURE-AS-A-SERVICE

- Compute resources, on demand, as-a-service
 - Generally raw “IT” resources
 - Hardware, network, containers, operating systems
- Typically provided through virtualization
- Generally not-preconfigured
- Administrative burden is owned by cloud consumer
- Best when high-level control over environment is needed
- Scaling is generally **not** automatic...
- Resources can be managed in bundles
- AWS CloudFormation: Allows specification in JSON/YAML of cloud infrastructures

October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.51

51

SC1

- M D FL

SC2

- M D L
- F

SC3

- M D FL

SC4

- M D F L

SC5

- M
- D FL

SC6

- M
- D F L

SC7

- M
- D F L

SC8

- M
- D FL

SC9

- M
- D L F

SC10

- M F
- D L

SC11

- M F
- D L

SC12

- M L
- D F

SC13

- M L
- D F

SC14

- M D L
- F

SC15

- M L D
- F

M: Tomcat ApplicationServer

D: Postgresql DB

F: nginx file server

L: Log server (Codebeamer)

52

Slides by Wes J. Lloyd

L7.26

SC1

M D

F L

SC2

M D

F

L

SC3

M D

F L

SC4

M D

F

L

Bell's Number:

k: number of ways
n components can be
distributed across containers

n	k
4	15
5	52
6	203
7	877
8	4,140
9	21,147
n	...

SC14

M D

L

F

SC15

M L

F

D

M: Tomcat ApplicationServer

D: Postgresql DB

F: nginx file server

L: Log server (Codebeamer)

53

SC1

M D

F L

SC2

M D

F

L

SC3

M D

F L

SC4

M D

F

L

Component Composition Example

An application with 4 components has 15 compositions

One or more component(s) deployed to each VM

Each VM launched to separate physical machine

SC5

M

D

SC6

M

D F

L

SC7

M

D

F

L

SC14

M D

L

F

SC15

M L

F

D

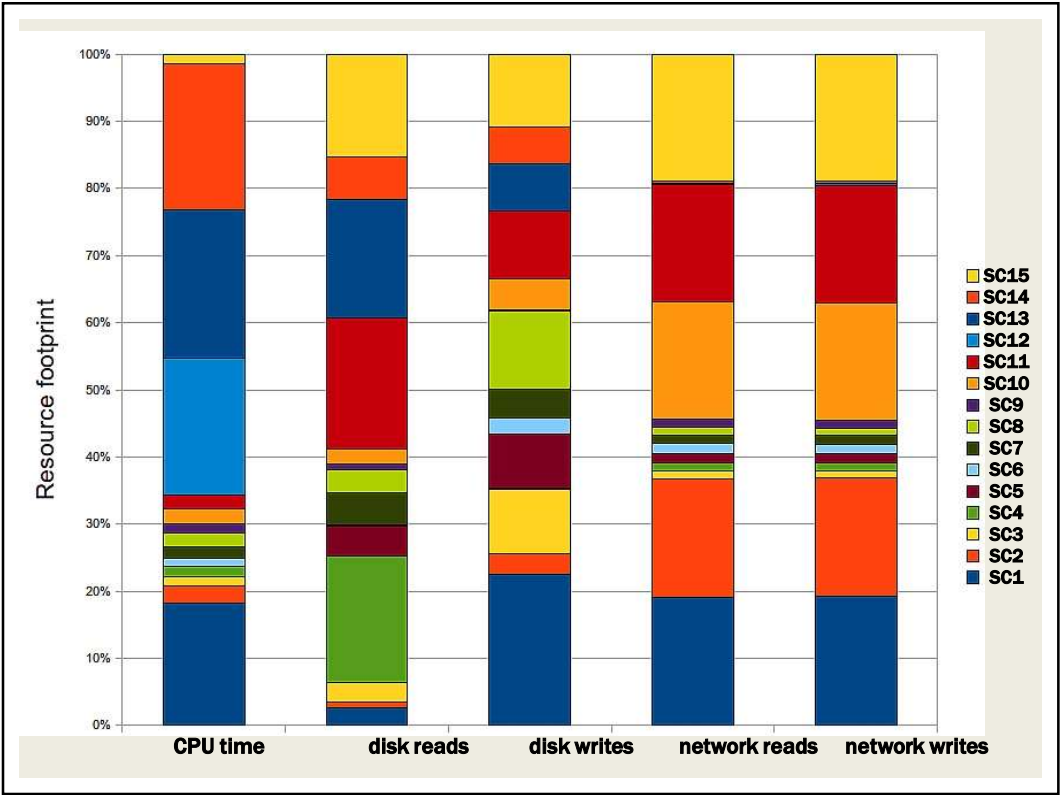
M: Tomcat ApplicationServer

D: Postgresql DB

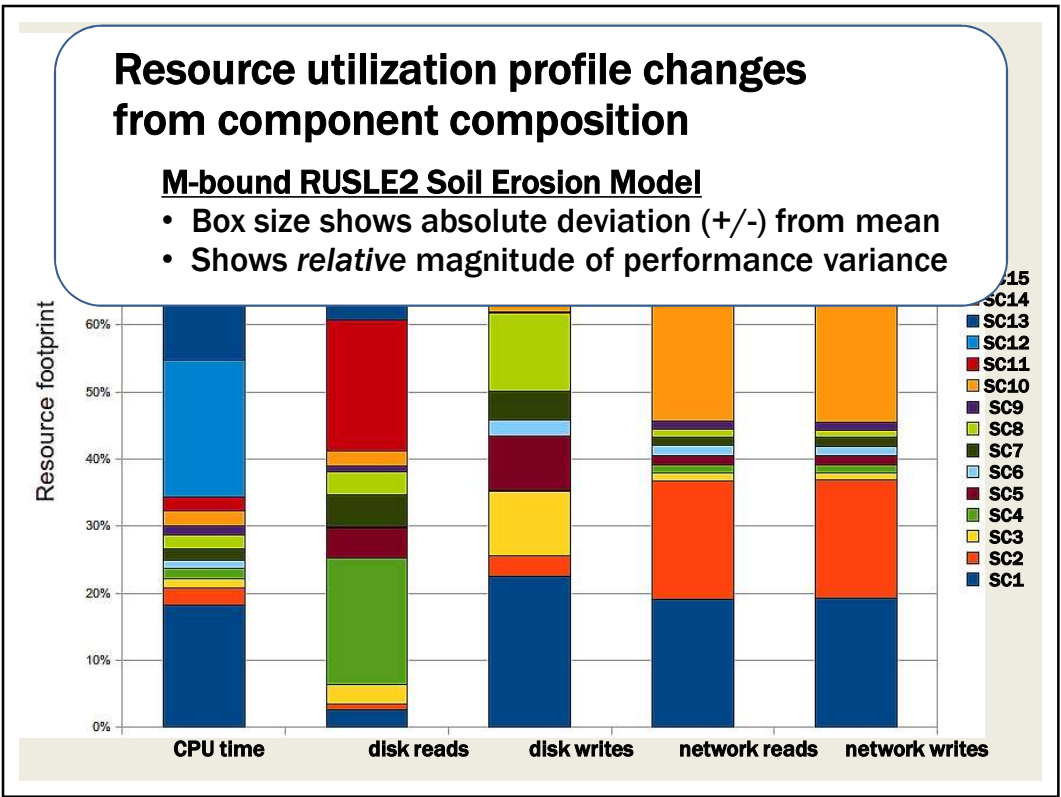
F: nginx file server

L: Log server (Codebeamer)

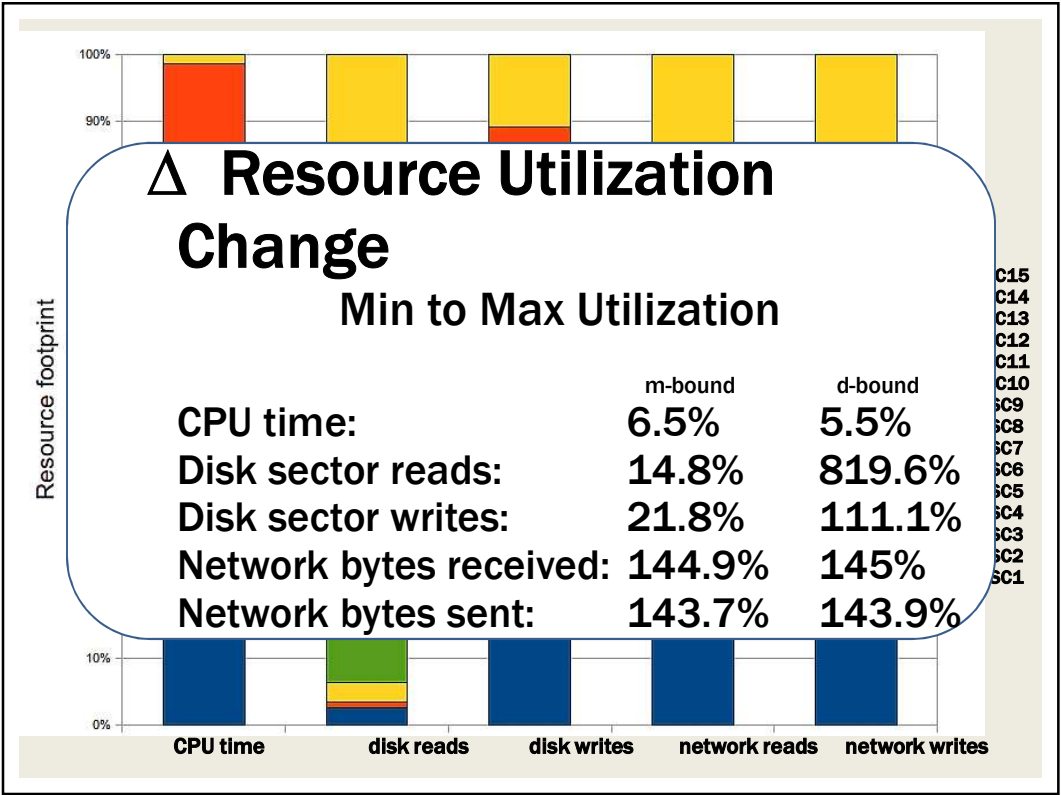
54



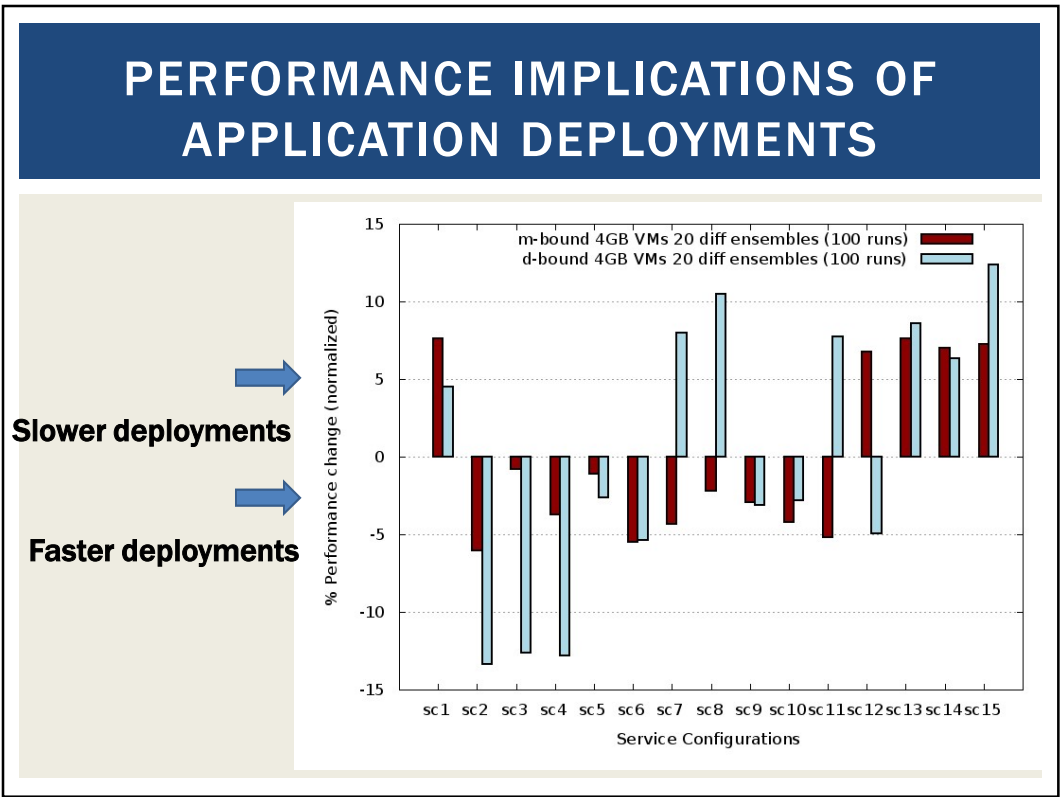
55



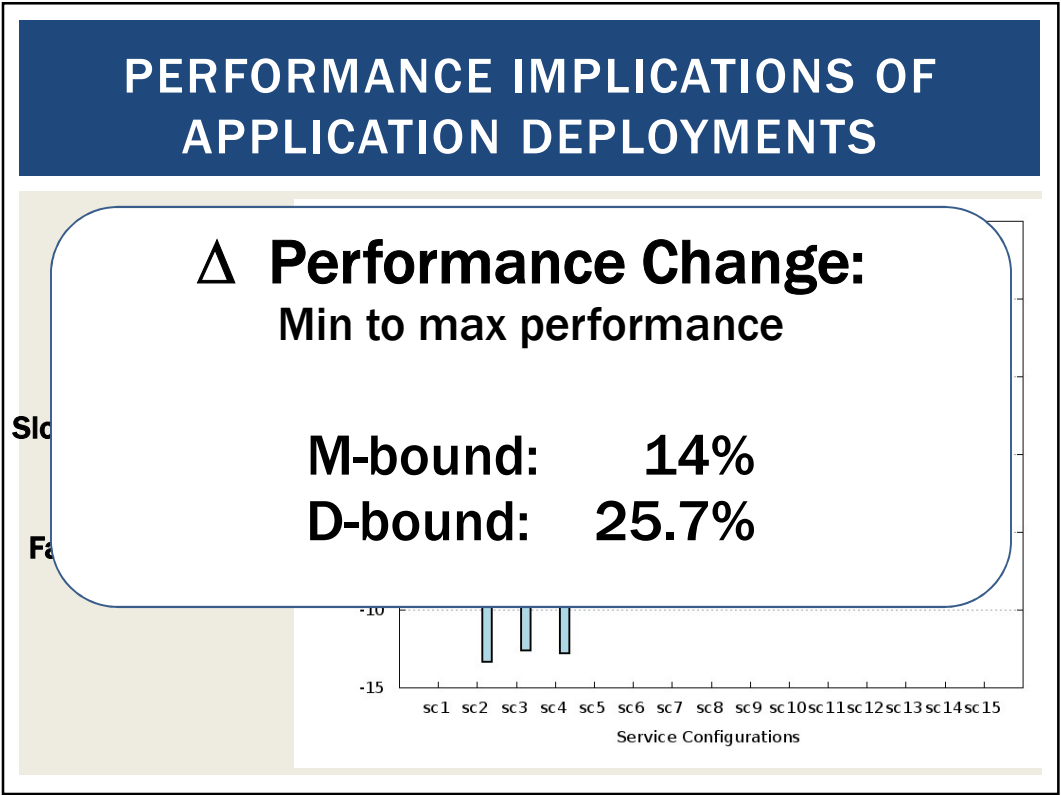
56



57



58



59

PLATFORM-AS-A-SERVICE

- Predefined, ready-to-use, hosting environment
- Infrastructure is further obscured from end user
- Scaling and load balancing may be automatically provided and automatic
- Variable to no ability to influence responsiveness

■ Examples:

- Google App Engine
- Heroku
- AWS Elastic Beanstalk
- AWS Lambda (FaaS)

```
graph LR; subgraph Clients; C1[Laptop]; C2[Laptop]; C3[Laptop]; end; subgraph Servers; S1[Server]; S2[Server]; S3[Server]; end; LB[Load Balancer]; C1 --> LB; C2 --> LB; C3 --> LB; LB --> S1; LB --> S2; LB --> S3;
```

October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.60

60

USES FOR PAAS

- Cloud consumer
 - Wants to extend on-premise environments into the cloud for “web app” hosting
 - Wants to entirely substitute an on-premise hosting environment
 - Cloud consumer wants to become a cloud provider and deploy its own cloud services to external users
- PaaS spares IT administrative burden compared to IaaS


October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.61


61

SERVERLESS COMPUTING



What is serverless?

Build and run applications without thinking about servers



October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.62

62

SERVERLESS COMPUTING - 2

Evolving to serverless

Physical servers in datacenters

Virtual servers in datacenters

Virtual servers in the cloud

SERVERLESS

amazon web services

October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.63

63

SERVERLESS COMPUTING

Pay only for CPU/memory utilization

High Availability

Fault Tolerance

Infrastructure Elasticity

No Setup

Function-as-a-Service (FAAS)

64

SERVERLESS COMPUTING

Why Serverless Computing?

**Many features of distributed systems,
that are challenging to deliver, are
provided automatically**

...they are built into the platform

65

SERVERLESS VS. FAAS

- **Serverless Computing**
- Refers to the avoidance of managing servers
- Can pertain to a number of “as-a-service” cloud offerings
- **Function-as-a-Service (FaaS)**
 - Developers write small code snippets (microservices) which are deployed separately
- **Database-as-a-Service (DBaaS)**
- **Container-as-a-Service (CaaS)**
- **Others...**

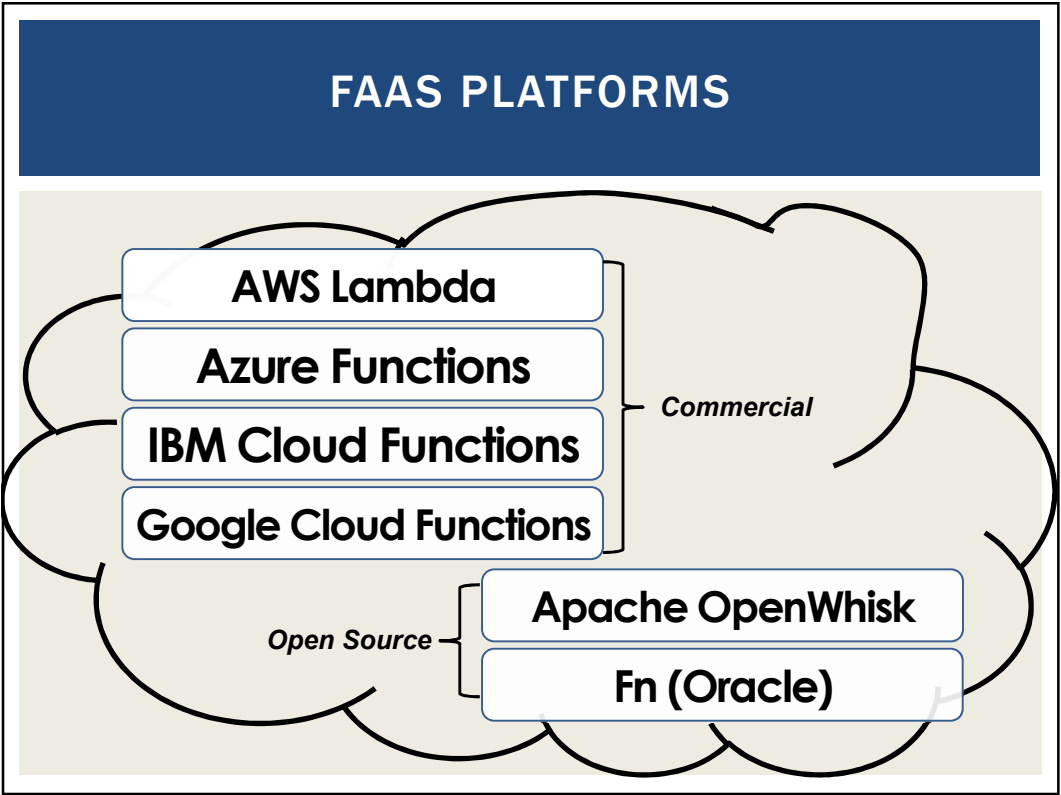
- **Serverless is a buzzword**
- **This space is evolving...**

October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.66

66



67

AWS LAMBDA

Using AWS Lambda

Bring your own code

- Node.js, Java, Python, C#
- Bring your own libraries (even native ones)

Simple resource model

- Select power rating from 128 MB to 3 GB
- CPU and network allocated proportionately

Flexible use

- Synchronous or asynchronous
- Integrated with other AWS services

Flexible authorization

- Securely grant access to resources and VPCs
- Fine-grained control for invoking your functions

Images credit: aws.amazon.com

68

FAAS PLATFORMS - 2

- New cloud platform for hosting application code
- Every cloud vendor provides their own:
 - AWS Lambda, Azure Functions, Google Cloud Functions, IBM OpenWhisk
- Similar to platform-as-a-service
- Replace opensource web container (e.g. Apache Tomcat) with abstracted vendor-provided **black-box** environment

October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.69

69

FAAS PLATFORMS - 3

- Many challenging features of distributed systems are provided automatically
- **Built into the platform:**
- Highly availability (24/7)
- Scalability
- Fault tolerance

October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.70

70

CLOUD NATIVE SOFTWARE ARCHITECTURE

- Every service with a different pricing model

Example: Weather Application

S3 API GATEWAY LAMBDA DYNAMODB

Front-end code for weather app hosted in S3 User clicks on link to get local weather information App makes REST API call to endpoint Lambda is triggered 35° C Lambda runs code to retrieve local weather information and returns data back to user

October 17, 2018	TCSS562: Software Engineering for Cloud Computing [Fall 2018] School of Engineering and Technology, University of Washington - Tacoma	L7.71
------------------	--	-------

71

IAAS BILLING MODELS

- Virtual machines as-a-service at ¢ per hour
- No premium to scale:

1000 computers

@

1 hour

=

1 computer

@

1000 hours
- Illusion of infinite scalability to cloud user
- As many computers as you can afford
- Billing models are becoming increasingly granular
 - By the minute, second, 1/10th sec
- Auction-based instances: Spot instances →

Spot Instance Pricing History

October 17, 2018	TCSS562: Software Engineering for Cloud Computing [Fall 2018] School of Engineering and Technology, University of Washington - Tacoma	L7.72
------------------	--	-------

72

FAAS COMPUTING BILLING MODELS

- AWS Lambda Pricing

- FREE TIER:

first 1,000,000 function calls/month → FREE

first 400 GB-sec/month → FREE

- Afterwards: *obfuscated pricing (AWS Lambda):*
\$0.0000002 per request
\$0.000000208 to rent 128MB / 100-ms

October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.73

73

WEBSERVICE HOSTING EXAMPLE

- ON AWS Lambda

- Each service call: 100% of 1 CPU-core
100% of 4GB of memory
- Workload: 2 continuous client threads
- Duration: 1 month (30 days)

- ON AWS EC2:

- Amazon EC2 c4.large 2-vCPU VM
- Hosting cost: \$72/month
c4.large: 10¢/hour, 24 hrs/day x 30 days

- **How much would hosting this workload cost on AWS Lambda?**

October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.74

74

PRICING OBFUSCATION

■ Workload:20,736,000 GB-sec

■ FREE:-400 GB-sec

■ Ch...

■ M...

■ In...

■ FF...

■ Charge...

■ Calls:

■ Total:

■ BREAK-EVEN POINT = ~4,326,927 GB-sec-month

Worst-case scenario = ~4.8x !

AWS EC2:\$72.00

AWS Lambda:\$345.88

7,184,000 calls

\$.84

\$345.88

75

FAAS PRICING

■ Break-even point is the point where renting VMs or deploying to a serverless platform (e.g. Lambda) is exactly the same.

■ Our example is for one month

■ Could also consider one day, one hour, one minute

■ What factors influence the break-even point for an application running on AWS Lambda?

October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.76

76

FACTORS IMPACTING PERFORMANCE OF FAAS COMPUTING PLATFORMS

- Infrastructure elasticity
- Load balancing
- Provisioning variation
- Infrastructure retention: COLD vs. WARM
 - Infrastructure freeze/thaw cycle
- Memory reservation
- Service composition

October 17, 2018	TCCS562: Software Engineering for Cloud Computing [Fall 2018] School of Engineering and Technology, University of Washington - Tacoma	L7.77
------------------	--	-------

77

FAAS CHALLENGES

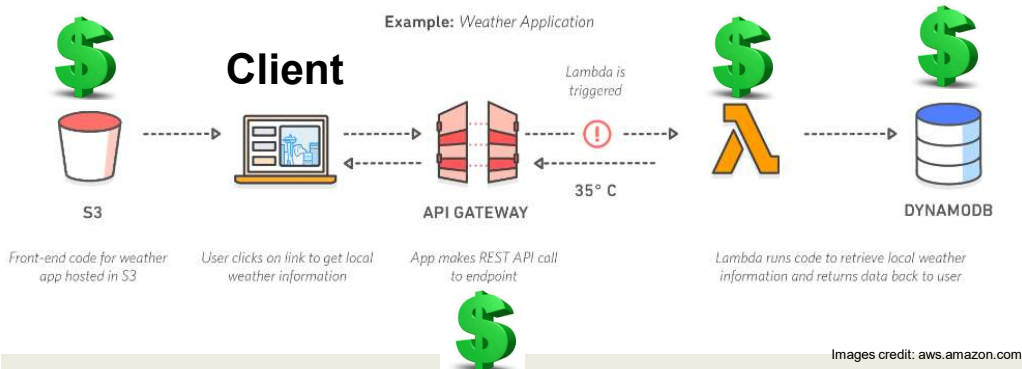
- Vendor architectural lock-in – how to migrate?
- Pricing obfuscation – is it cost effective?
- Memory reservation – how much to reserve?
- Service composition – how to compose software?
- Infrastructure freeze/thaw cycle – how to avoid?

October 17, 2018	TCCS562: Software Engineering for Cloud Computing [Fall 2018] School of Engineering and Technology, University of Washington - Tacoma	L7.78
------------------	--	-------

78

VENDOR ARCHITECTURAL LOCK-IN

- Cloud native (FaaS) software architecture requires external services/components



- Increased dependencies → increased hosting costs

79

PRICING OBFUSCATION

- VM pricing:** hourly rental pricing, billed to nearest second is intuitive...

- FaaS pricing:**

AWS Lambda Pricing

FREE TIER: first 1,000,000 function calls/month → FREE
 first 400 GB-sec/month → FREE

- Afterwards:** \$0.0000002 per request
 \$0.000000208 to rent 128MB / 100-ms

October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
 School of Engineering and Technology, University of Washington - Tacoma

L7.80

80

MEMORY RESERVATION QUEST



- Lambda memory reserved for functions
- UI provides “slider bar” to set function’s memory allocation
- Resource capacity (CPU, disk, network) coupled to slider bar:
“every doubling of memory, doubles CPU...”
- But how much memory do model services require?

October 17, 2018

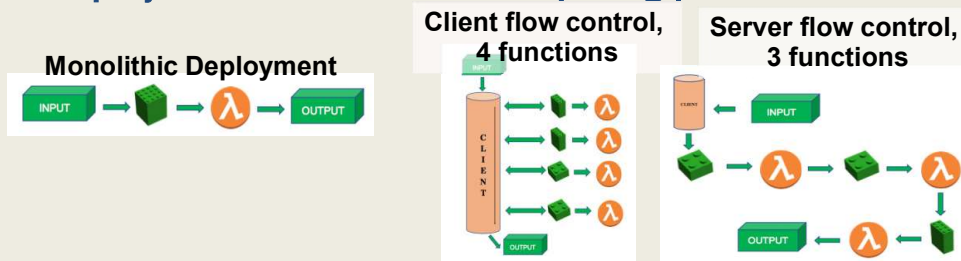
TCSS562: Software Engineering for Cloud Computing [Fall 2018]
 School of Engineering and Technology, University of Washington - Tacoma

L7.81

81

SERVICE COMPOSITION

- How should application code be composed for deployment to serverless computing platforms?



- Recommended practice:
 Decompose into many microservices
- Platform limits: code + libraries ~250MB
- How does composition impact the number of function invocations, and memory utilization?



Performance

82

INFRASTRUCTURE FREEZE/THAW CYCLE

- Unused infrastructure is deprecated
 - *But after how long?*
- Infrastructure: VMs, “containers”
- Provider-COLD / VM-COLD
 - “Container” images - built/transferred to VMs
- Container-COLD
 - Image cached on VM
- Container-WARM
 - “Container” running on VM



Performance



Image from: Denver7 – The Denver Channel News

83



FUNCTION-AS-A-SERVICE

AWS
Lambda
Demo

84

84

SOFTWARE-AS-A-SERVICE

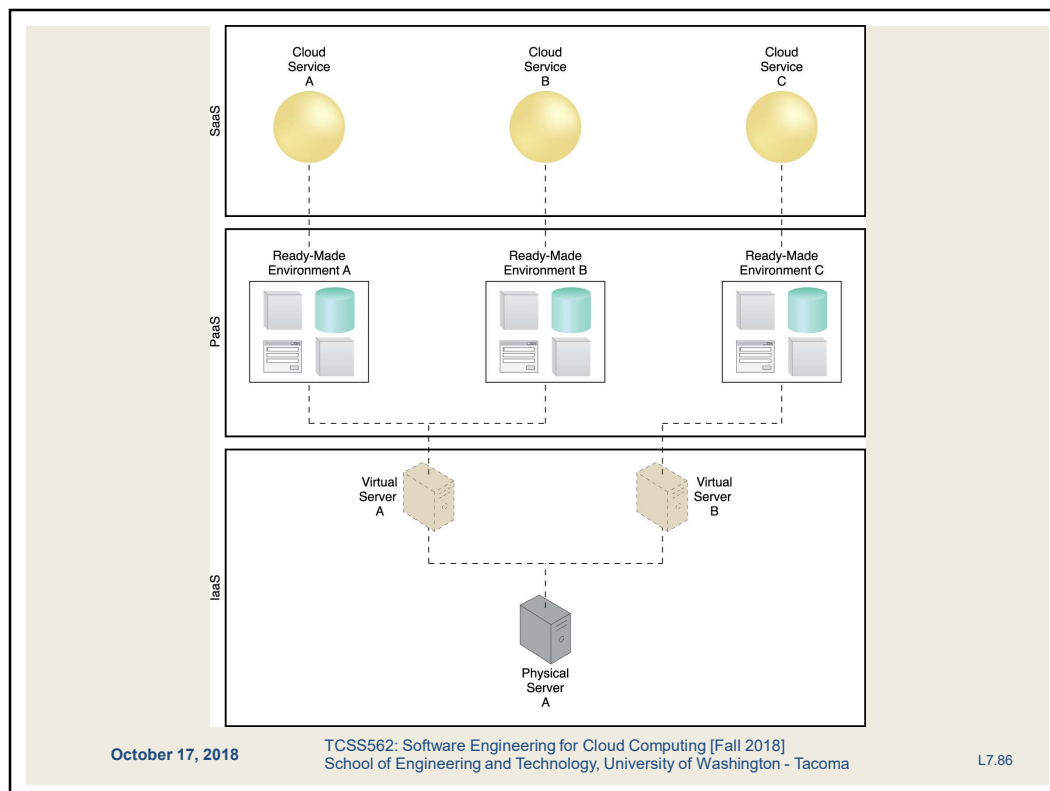
- Software applications as shared cloud service
- Nearly all server infrastructure management is abstracted away from the user
- Software is generally configurable
- SaaS can be a complete GUI/UI based environment
- Or UI-free (database-as-a-service)
- SaaS offerings
 - Google Docs
 - Office 365
 - Cloud9 Integrated Development Environment
 - Salesforce

October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.85

85



86

CONTAINER-AS-A-SERVICE

- Cloud service model for deploying application containers (e.g. Docker) to the cloud
- Deploy containers without worrying about managing infrastructure:
 - Servers
 - Or container orchestration platforms
 - Container platform examples: Kubernetes, Docker swarm, Apache Mesos/Marathon, Amazon Elastic Container Service
 - Container platforms support creation of container clusters on the using cloud hosted VMs
- CaaS Examples:
 - AWS Fargate
 - Azure Container Instances
 - Google KNative

October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.87

87

OTHER CLOUD SERVICE MODELS

- IaaS
 - Storage-as-a-Service
- PaaS
 - Integration-as-a-Service
- SaaS
 - Database-as-a-Service
 - Testing-as-a-Service
 - Model-as-a-Service
- ?
 - Security-as-a-Service
 - Integration-as-a-Service

October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L10.88

88

OBJECTIVES

- Cloud Computing Concepts and Models
 - Roles and boundaries
 - Cloud characteristics
 - Cloud delivery models
 - Cloud deployment models

October 17, 2018	TCSS562: Software Engineering for Cloud Computing [Fall 2018] School of Engineering and Technology, University of Washington - Tacoma	L7.89
------------------	--	-------

89

CLOUD DEPLOYMENT MODELS

- Distinguished by ownership, size, access
- Four common models
 - Public cloud
 - Community cloud
 - Hybrid cloud
 - Private cloud

October 17, 2018	TCSS562: Software Engineering for Cloud Computing [Fall 2018] School of Engineering and Technology, University of Washington - Tacoma	L7.90
------------------	--	-------

90

PUBLIC CLOUDS

The diagram illustrates the concept of public clouds. At the bottom, three server rack icons are labeled "organizations". Three large, upward-pointing arrows connect these organizations to a central cloud. Inside this cloud, several other cloud icons are shown, each labeled with a company name: Google, Salesforce, Microsoft, Yahoo, Amazon, Zoho, and Rackspace. This represents organizations using public cloud services provided by these major companies.

October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.91

91

COMMUNITY CLOUD

- Specialized cloud built and shared by a particular community
- Leverage economies of scale within a community
- Research oriented clouds
- Examples:
 - Bionimbus - bioinformatics
 - Chameleon
 - CloudLab

The diagram illustrates the concept of a community cloud. At the bottom, six server rack icons are labeled "community of organizations". Three large, upward-pointing arrows connect these organizations to a central cloud. Inside this cloud, several server rack icons and yellow and green spheres are shown, representing a specialized cloud environment shared by a specific community.

October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.92

92

PRIVATE CLOUD

- Compute clusters configured as IaaS cloud
- Open source software
 - Eucalyptus
 - Openstack
 - Apache Cloudstack
 - Nimbus
- Virtualization: XEN, KVM, ...

October 17, 2018 TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma L7.93

93

HYBRID CLOUD

- Extend private cloud typically with public or community cloud resources
- Cloud bursting:
Scale beyond one cloud when resource requirements exceed local limitations
- Some resources can remain local for security reasons

October 17, 2018 TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma L7.94

94

OTHER CLOUDS

- **Federated cloud**
 - Simply means to aggregate two or more clouds together
 - Hybrid is typically private-public
 - Federated can be public-public, private-private, etc.
 - Also called inter-cloud
- **Virtual private cloud**
 - Google and Microsoft simply call these virtual networks
 - Ability to interconnect multiple independent subnets of cloud resources together
 - Resources allocated private IPs from individual network subnets can communicate with each other (10.0.1.0/24) and (10.0.2.0/24)
 - Subnets can span multiple availability zones within an AWS region


October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.95

95

QUESTIONS



October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.96

96

SIMPLE VPC

■ Recommended when using Amazon EC2

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	igw-id

October 17, 2018

TCSS562: Software Engineering for Cloud Computing [Fall 2018]
School of Engineering and Technology, University of Washington - Tacoma

L7.97

97

VPC SPANNING AVAILABILITY ZONES

Destination	Target
10.0.0.0/16	local

98