TCSS 562: SOFTWARE ENGINEERING FOR CLOUD COMPUTING

Cloud Computing Concepts and Models - II

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

1

OFFICE HOURS – FALL 2025

- **Thursdays:**
  - 6:00 to 7:00 pm - CP 229 & Zoom
- **Friday – *** THIS WEEK *****
  - 12:30 pm to 1:30 pm – ONLINE via Zoom
- Or email for appointment

> *Office Hours set based on Student Demographics survey feedback*
> ** - Friday office hours may be adjusted or canceled due meeting conflicts or other obligations. Adjustments will be announced via Canvas.*

2

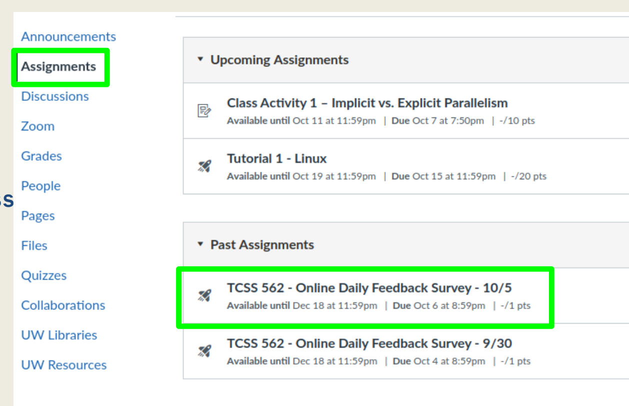
OBJECTIVES – 10/23

- Questions from 10/21
- Tutorials Questions
- Tutorial 4 - to be posted...
- From: Cloud Computing Concepts, Technology & Architecture: Chapter 4: Cloud Computing Concepts and Models:
  - Cloud computing delivery models
  - Cloud deployment models
- AWS Overview and demo
- 2nd hour:
  - Activity 2 - Intro to EC2
  - Term Project Planning

3

ONLINE DAILY FEEDBACK SURVEY

- Daily Feedback Quiz in Canvas – Take After Each Class
- **1-point Extra Credit** for completing online
- **2-points Extra Credit** for completing in-person in class
- 36 points possible
- 2.5% added to final course grade for (36/36)



4

WARNING

- DO NOT SUBMIT BOTH A PAPER AND AN ONLINE SURVEY OR YOU WILL LOOSE POINTS
- CANVAS WILL AUTOMATICALLY REPLACE THE PAPER SURVEY SCORE (2 PTS) WITH THE ONLINE SURVEY (1 PT)
- *** COMPLETE ONLY ONE SURVEY FOR EACH CLASS SESSION ***
- WE WILL NOT BE ABLE TO DUPLICATE CHECK SURVEYS FOR EACH CLASS SESSION AND MAKE CORRECTIONS

MATERIAL / PACE

- Please classify your perspective on material covered in today's class (38 respondents, 26 in-person, 12 online):
- ew, 10-mostly new
- **Average – 7.32 (↑ - *previous 7.00*)**
- Please rate the pace of today's class:
- 1-slow, 5-just right, 10-fast
- **Average – 5.24 (↓ - previous 5.44)**

FEEDBACK FROM 10/23

- ***How liable is AWS to service failure like the one seen recently?***
- The disruption on Monday was the largest I recall in several years. It is quite uncommon.

October 23, 2025	TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma	L8.7

7

FEEDBACK - 2

- ***What is CPU time share in the context of cloud computing?***
- For every minute, each CPU core has 60 seconds of "time".
- The "time share" is the fraction of time allocated to a VM, microVM, or container for a given time interval (i.e. minute)
- Two users can simultaneously share the same CPU core with a 50% time share (30 seconds each for 1 minute)
- In reality, many users share the CPU cores of cloud servers, each owning a different fraction of time
- Scheduling: if a user owns 30 seconds of CPU time share, the hypervisor is responsible for scheduling this time on a physical CPU core
- Cloud VMs have virtual CPU cores, and hypervisors schedule their time on the shared physical cores of servers
- This way, many users can share an expensive cloud server simultaneously

October 23, 2025	TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma	L8.8

8

FEEDBACK - 3

- ***When working on tutorial 3, I created instance but didn't connect to it. After a while the instance was gone w/o any action. Is that normal?***
- ***The AWS console shows there's no running instance and it can't find the instance created in "instances" section***
  - For spot instances, when the cloud region and instance type is in demand, it is typical that the instance can be reclaimed within a few hours to a day
  - The positive here is the charges stop, once the instance terminates

October 23, 2025	TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma	L8.9

9

FEEDBACK - 4

- ***How can the "garbage collection" of AWS Lambda function instances be avoided?***
  - In 2018, working with a UWT MSCSS Capstone student, we wrote a paper that investigated alternatives to "keeping alive" function instances by pinging them at different intervals
  - Paper: **https://faculty.washington.edu/wlloyd/papers/wosc_camera_ready.pdf**
  - Afterwards AWS Lambda introduced "**provisioned concurrency**".
  - Users can pay to keep a set number of function instances in the warm state.
- Charges for provision concurrency (x86_64 INTEL):
- $0.0000041667 for every GB-second (WARM)
- $0.0000097222 for every GB-second (COMPUTE – this is discounted)
- Example 100 function instances @ 1.5 GB for 8 hours:
- 8 hours * 3600 seconds/hour = 28,800 seconds
- 100 function instances * 1.5 GB = 150 GB
- 150 GB * 28,800 seconds = 4,320,000 GB-seconds
- 4,320,000 * $0.0000041667 = $18

BONUS: ON EC2, what is the 8-hr cost for an instance w/ 150GB & ~100 vCPUS?

October 23, 2025	TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma	L8.10

10

IN CLASS QUIZZES

- Anticipated dates

- Designed for 1 hour (starting at 4:40pm)
- Open notes & books
- Closed laptop, smartphone, neighbor
- No digital devices

- Quiz 1 – Thursday November 6
- Quiz 2 – Tuesday December 2

OBJECTIVES – 10/23

- Questions from 10/21
- **Tutorials Questions**
- Tutorial 4 - to be posted...
- From: Cloud Computing Concepts, Technology & Architecture: Chapter 4: Cloud Computing Concepts and Models:
  - Cloud computing delivery models
  - Cloud deployment models
- AWS Overview and demo
- 2nd hour:
  - Activity 2 - Intro to EC2
  - Term Project Planning

TUTORIAL 2 – DUE OCT 21 (CLOSES OCT 25 AOE)

- Introduction to Bash Scripting
- https://faculty.washington.edu/wlloyd/courses/tcss562/tutorials/TCSS462_562_f2025_tutorial_2.pdf
- Review tutorial sections:
- Create a BASH webservice client
  1. What is a BASH script?
  2. Variables
  3. Input
  4. Arithmetic
  5. If Statements
  6. Loops
  7. Functions
  8. User Interface
- Call service to obtain IP address & lat/long of computer
- Call weatherbit.io API to obtain weather forecast for lat/long

October 11, 2022	TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma	L4.13

13

TUTORIAL 0

- Getting Started with AWS
- https://faculty.washington.edu/wlloyd/courses/tcss562/tutorials/TCSS462_562_f2025_tutorial_0.pdf
- Create an AWS account
- Create account credentials for working with the CLI
- Install awsconfig package
- Setup awsconfig for working with the AWS CLI

October 23, 2025	TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma	L7/14

14

TUTORIAL 3 – OCT 30 (TEAMS OF 2)

- Best Practices for Working with Virtual Machines on Amazon EC2
- https://faculty.washington.edu/wlloyd/courses/tcss562/tutorials/TCSS462_562_f2025_tutorial_3.pdf
- Creating a spot VM
- Creating an image from a running VM
- Persistent spot request
- Stopping (pausing) VMs
- EBS volume types
- Ephemeral disks (local disks)
- Mounting and formatting a disk
- Disk performance testing with Bonnie++
- Cost Saving Best Practices

OBJECTIVES – 10/23

- Questions from 10/21
- Tutorials Questions
- **Tutorial 4 - to be posted…**
- From: Cloud Computing Concepts, Technology & Architecture: Chapter 4: Cloud Computing Concepts and Models:
  - Cloud computing delivery models
  - Cloud deployment models
- AWS Overview and demo
- 2nd hour:
  - Activity 2 - Intro to EC2
  - Term Project Planning

TUTORIAL 4 – TO BE POSTED (FRI?)

- Introduction to AWS Lambda with the Serverless Application Analytics Framework (SAAF)
- Obtaining a Java development environment
- Introduction to Maven build files for Java
- Create and Deploy “hello” Java AWS Lambda Function
  - Creation of API Gateway REST endpoint
- Sequential testing of “hello” AWS Lambda Function
  - API Gateway endpoint
  - AWS CLI Function invocation
- Observing SAAF profiling output
- Parallel testing of “hello” AWS Lambda Function with faas_runner
- Performance analysis using faas_runner reports
- Two function pipeline development task

17

CLOUD COMPUTING: CONCEPTS AND MODELS

18

CATCH UP – FROM 10/21

- Questions from 10/15
- Tutorials Questions
- Tutorial 4 – Intro to FaaS – AWS Lambda
- Background on AWS Lambda for the Term Project - II
- **From: Cloud Computing Concepts, Technology & Architecture: Chapter 4: Cloud Computing Concepts and Models:**
  - **Roles and boundaries**
  - Cloud characteristics
  - Cloud delivery models
  - Cloud deployment models
- Team Planning - Breakout Rooms

October 17, 2024	TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma	L7.19

19

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)

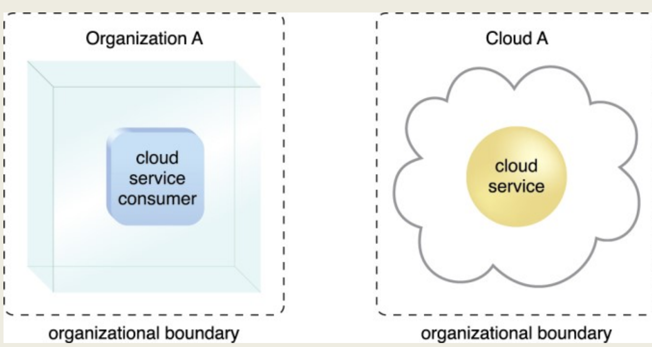
October 21, 2025	TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma	L7.20

20

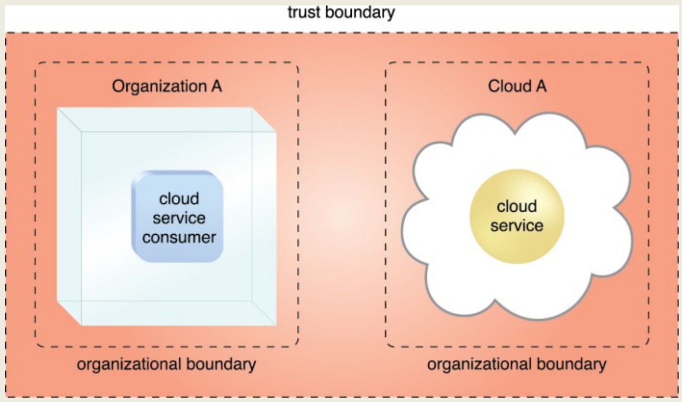
ROLES - 2

- 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

ORGANIZATION BOUNDARY



TRUST BOUNDARY



October 21, 2025	TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma	L7.23

23

OBJECTIVES – 10/21

- Questions from 10/16
- Tutorials Questions
- Tutorial 4 – Intro to FaaS – AWS Lambda
- Background on AWS Lambda for the Term Project - II
- **From: Cloud Computing Concepts, Technology & Architecture: Chapter 4: Cloud Computing Concepts and Models:**
  - Roles and boundaries
  - **Cloud characteristics**
  - Cloud delivery models
  - Cloud deployment models
- Team Planning - Breakout Rooms

October 21, 2025	TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma	L7.24

24

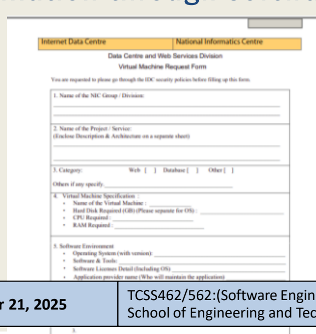
CLOUD CHARACTERISTICS

- 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

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



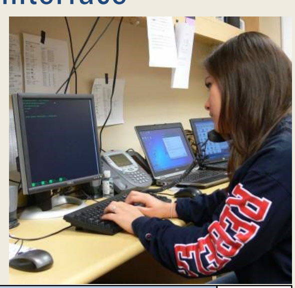
Internet Data Centre | National Informatics Centre

Data Centre and Web Services Division
Virtual Machine Request Form

You are requested to please go through the IDC security policies before filling up this form.

1. Name of the NIC Group / Division:
2. Name of the Project / Service:
(Enclose Description & Architecture on a separate sheet)
3. Category: Web [ ] Database [ ] Other [ ]
Others if any specify
4. Virtual Machine Specification :
- Name of the Virtual Machine :
- Hard Disk Required (GB) (Please separate for OS) :
- CPU Required :
- RAM Required :
5. Software Environment
- Operating System (with version):
- Software & Tools:
- Software Licenses Detail (Including OS)
- Application provider name (Who will maintain the application)

Please enclose the application details form.



UBIQUITOUS ACCESS

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

CLOUD CHARACTERISTICS

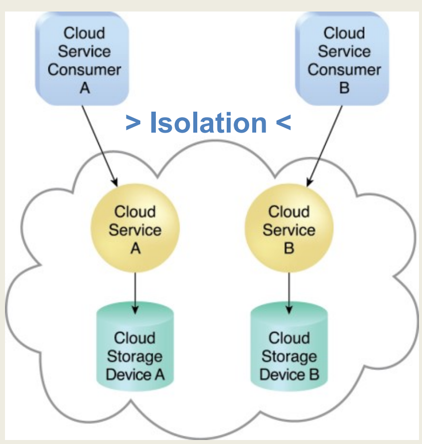
- On-demand usage
- Ubiquitous access
- **Multitenancy (resource pooling)**
- Elasticity
- Measured usage
- Resiliency

- Assessing these features helps measure the value offered by a given cloud service or platform

MULTITENANCY

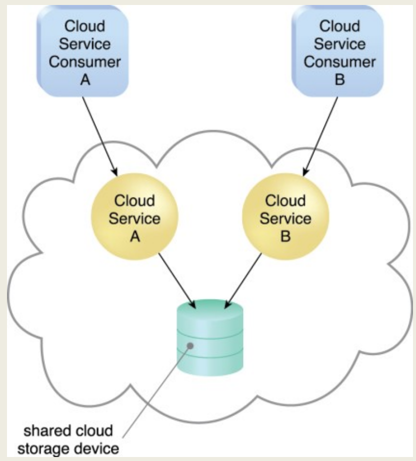
- Cloud providers pool resources together to share them with many users
- Serve multiple cloud service consumers
- IT resources can be dynamically assigned, reassigned based on demand
- Multitenancy can lead to performance variation

SINGLE TENANT MODEL



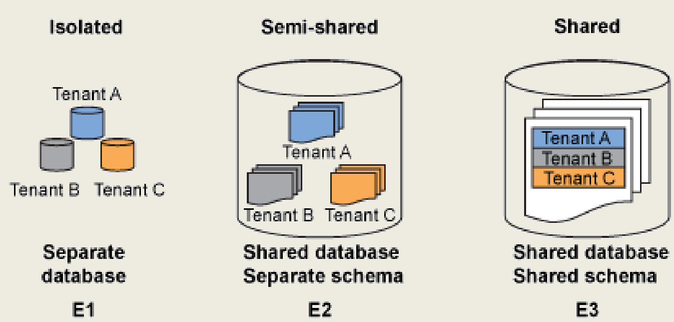
MULTITENANT MODEL

- 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 through **multi-tenancy**



31

MULTITENANT DATABASE

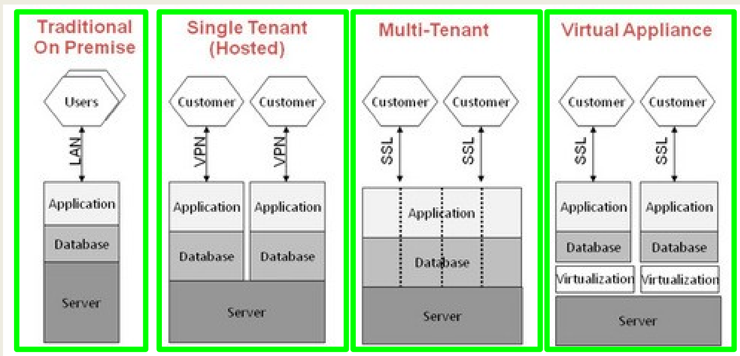


- Many users on a single database instance
- ***What issues may occur when sharing a single database instance?***

32

MULTITENANCY OF RESOURCES

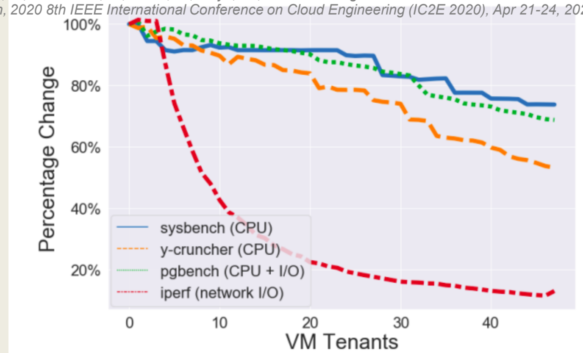
- Where is the multitenancy?
  - >> What is shared? What is isolated?



RESOURCE CONTENTION FROM MUTLI-TENANCY

- Despite best efforts at isolation, co-resident VMs on a single cloud server running identical benchmarks simultaneously do not perform equally.

From Han, X., Schooley, R., Mackenzie, D., David, O., Lloyd, W., Characterizing Public Cloud Resource Contention to Support Virtual Machine Co-residency Prediction, 2020 8th IEEE International Conference on Cloud Engineering (IC2E 2020), Apr 21-24, 2020.



Up to 48 VMs sharing same server !!

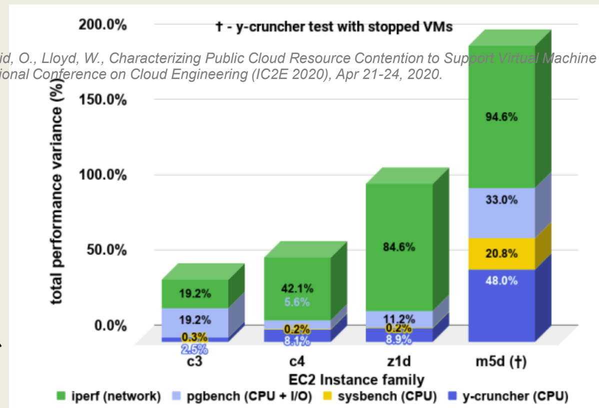
RESOURCE CONTENTION FROM MUTLI-TENANCY - 2

- Performance variation from multi-tenancy is increasing as cloud servers add more CPU cores

From Han, X., Schooley, R., Mackenzie, D., David, O., Lloyd, W., Characterizing Public Cloud Resource Contention to Support Virtual Machine Co-residency Prediction, 2020 8th IEEE International Conference on Cloud Engineering (IC2E 2020), Apr 21-24, 2020.

- Running many idle operating system instances can impose significant overhead for some workloads

Maximum potential resource contention (i.e. worst-case scenario) →



35

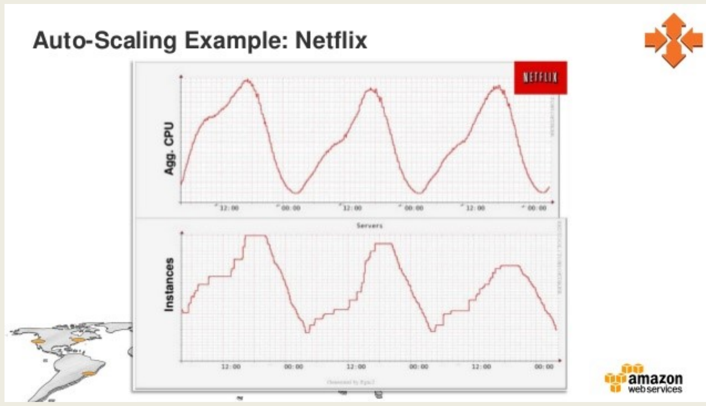
ELASTICITY

- 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

36

PREDICTABLE DEMAND

- AWS EC2 Scaling Example:



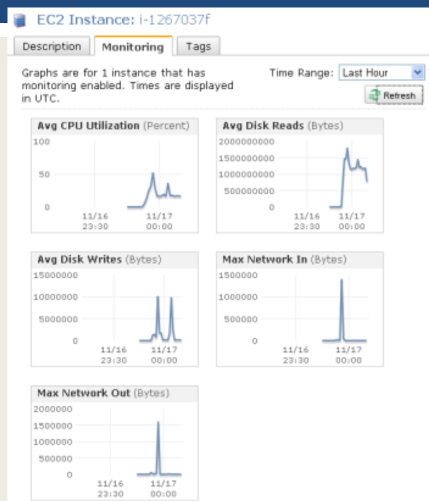
From: Kejariwal, A., 2013, March. Techniques for optimizing cloud footprint. In 2013 IEEE Int. Conf. on Cloud Engineering (IC2E), pp. 258-268.

MEASURED USAGE

- Cloud platform tracks usage of IT resources
- For billing purposes
- Enables charging only for IT resources actually used
- Can be time-based (millisec, second, minute, hour, day)
  - Granularity is increasing…
- Can be throughput-based (data transfer: MB/sec, GB/sec)
- Can be resource/reservation based (vCPU/hr, GB/hr)

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

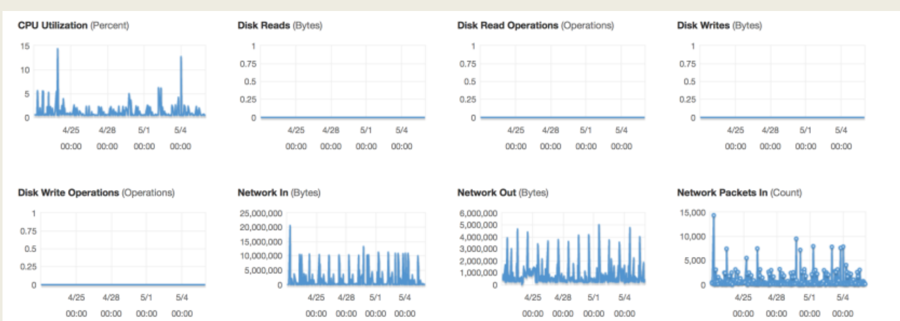
EC2 CLOUDWATCH METRICS



October 17, 2024	TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma	L7.39

39

EC2 CLOUDWATCH METRICS

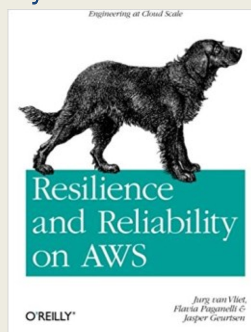


October 17, 2024	TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma	L7.40

40

RESILIENCY

- Distributed redundancy across physical locations (regions on AWS)
- 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



Elasticity is often provided using threshold based scaling. When can threshold based scaling (i.e. CPU utilization > 80%) under or over provision resources?

- **A** When the application is primarily I/O bound, a CPU threshold may never be met, or be met too late to scale up.
- **B** When the current resource utilization does not reflect future system demand.
- **C** When the current resource utilization (e.g. CPU) is temporarily increased as a result of external factors (i.e. resource contention from other tasks) that does not correlate to system demand.
- **D** When an application will soon complete a parallel phase, before executing a largely sequential phase
- **E** All of the above

When poll is active, respond at **pollev.com/wesleylloyd641**
Text **WESLEYLLOYD641** to **22333** once to join

The scaling threshold of "when CPU utilization > 80% scale up", is:

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

43

OBJECTIVES – 10/23

- Questions from 10/21
- Tutorials Questions
- Tutorial 4 - to be posted...
- From: Cloud Computing Concepts, Technology & Architecture: Chapter 4: Cloud Computing Concepts and Models:
  - Cloud computing delivery models
  - Cloud deployment models
- AWS Overview and demo
- 2nd hour:
  - Activity 2 - Intro to EC2
  - Term Project Planning

October 23, 2025 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma | L8.44

44

CLOUD COMPUTING DELIVERY MODELS

- Infrastructure-as-a-Service (IaaS)
- Platform-as-a-Service (PaaS)
- Software-as-a-Service (SaaS)

Serverless Computing:

- Function-as-a-Service (FaaS)
- Container-as-a-Service (CaaS)
- Other Delivery Models

October 23, 2025	TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma	L8.45

45

CLOUD COMPUTING DELIVERY MODELS

- Infrastructure-as-a-Service (IaaS) delivery model
- Virtualization is a key-enabling technology of IaaS cloud
- Uses virtual machines to deliver cloud resources to end users

October 23, 2025	TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma	L8.46

46

CLOUD COMPUTING DELIVERY MODELS

- Infrastructure-as-a-Service (IaaS) delivery model

Virtualization is key to sharing powerful servers among users by running _many_ isolated private virtual computers known as virtual machines (VMs)

…VMs are the basis of cloud v1.0

October 23, 2025	TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma	L8.47

47

CLOUD COMPUTING DELIVERY MODELS

- Infrastructure-as-a-Service (IaaS) delivery model

Virtual Machines are the building blocks for "Cloud Service Delivery Models"

They are the "vehicles" used to deliver compute resources to end users…

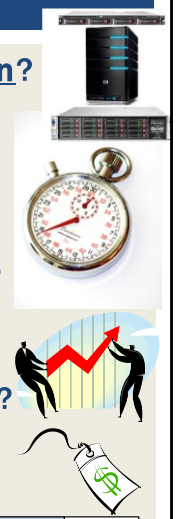
cloud 1.0

October 23, 2025	TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma	L8.48

48

CLOUD COMPUTING DELIVERY MODELS

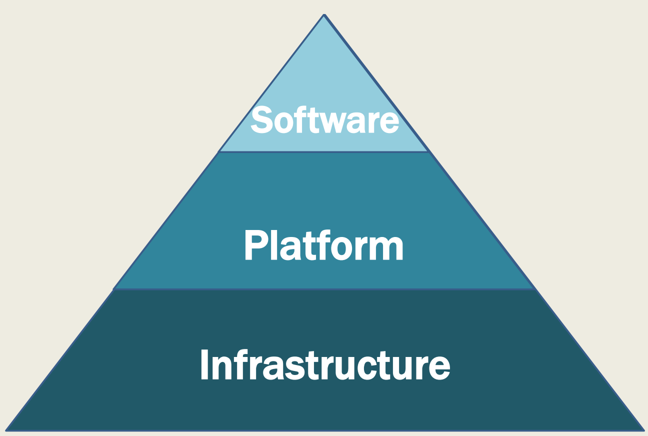
- What is the appropriate level of **abstraction**?
- How should applications be deployed?
  - IaaS, PaaS, SaaS, DbaaS, FaaS
- How do we ensure Quality-of-Service?
  - Performance, Availability, Responsiveness, Fault Tolerance
- How is ***scalability*** provided?
- As users, how do we minimize hosting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



October 23, 2025	TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma	L8.49

49

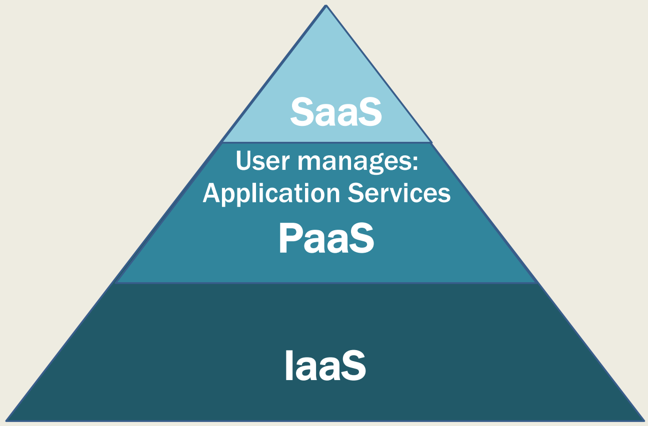
CLASSIC CLOUD COMPUTING DELIVERY MODELS



October 23, 2025	TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma	L8.50

50

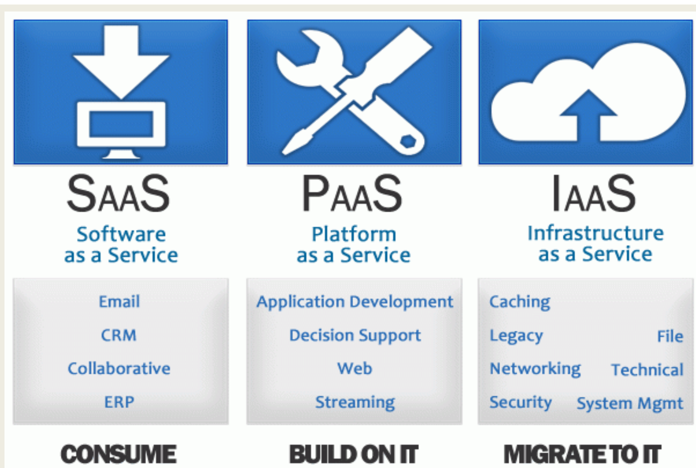
CLASSIC CLOUD COMPUTING DELIVERY MODELS



October 23, 2025 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025]
School of Engineering and Technology, University of Washington - Tacoma | L8.51

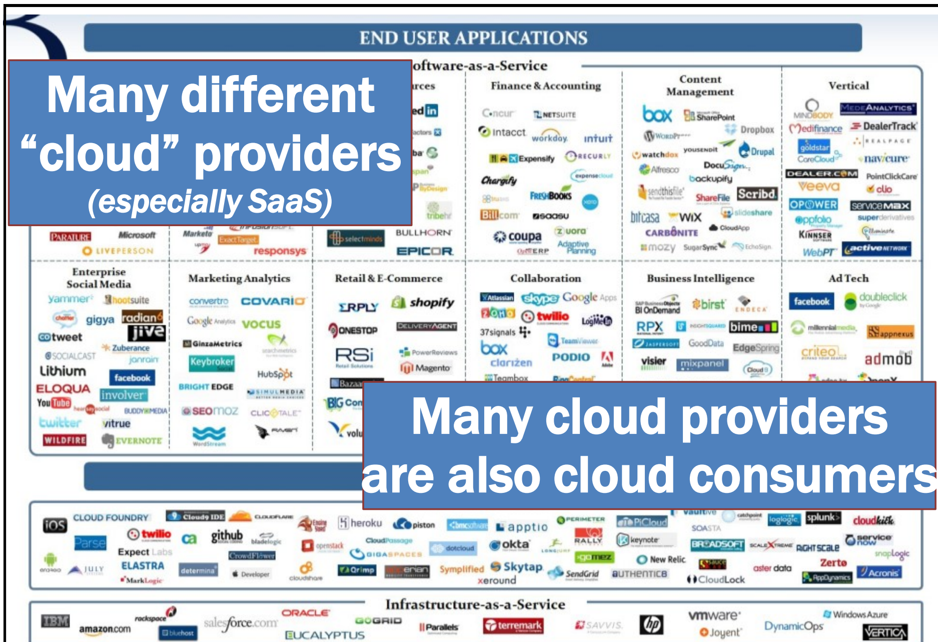
51

EXAMPLE CLOUD SERVICES



October 23, 2025 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025]
School of Engineering and Technology, University of Washington - Tacoma | L8.52

52



October 23, 2025 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma | L8.53

53

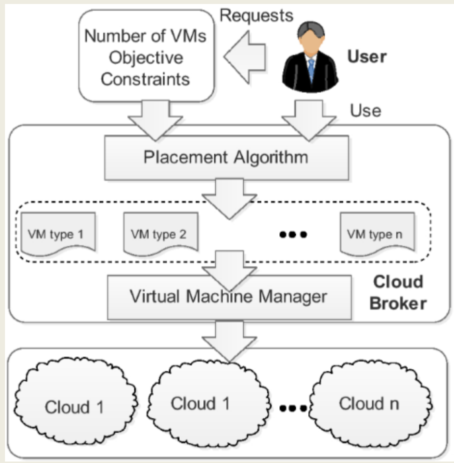
INFRASTRUCTURE-AS-A-SERVICE

- 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**: Scripts to specify creation of cloud infrastructures using JSON/YAML for app deployment

October 23, 2025 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma | L8.54

54

IAAS: VIRTUAL MACHINE PLACEMENT IN THE CLOUD

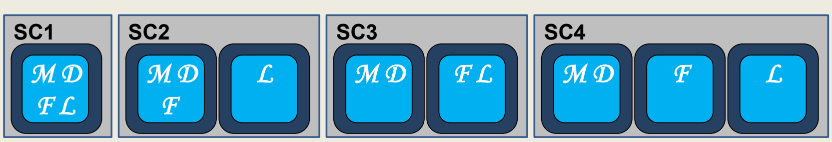


Controlled by the cloud provider

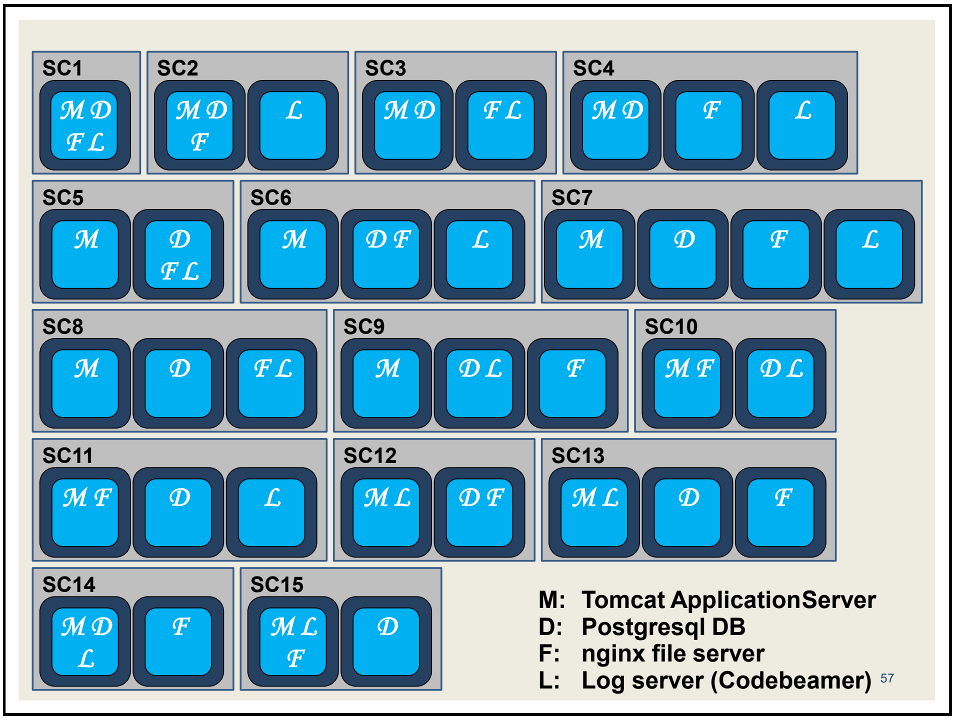
55

COMPONENT COMPOSITION

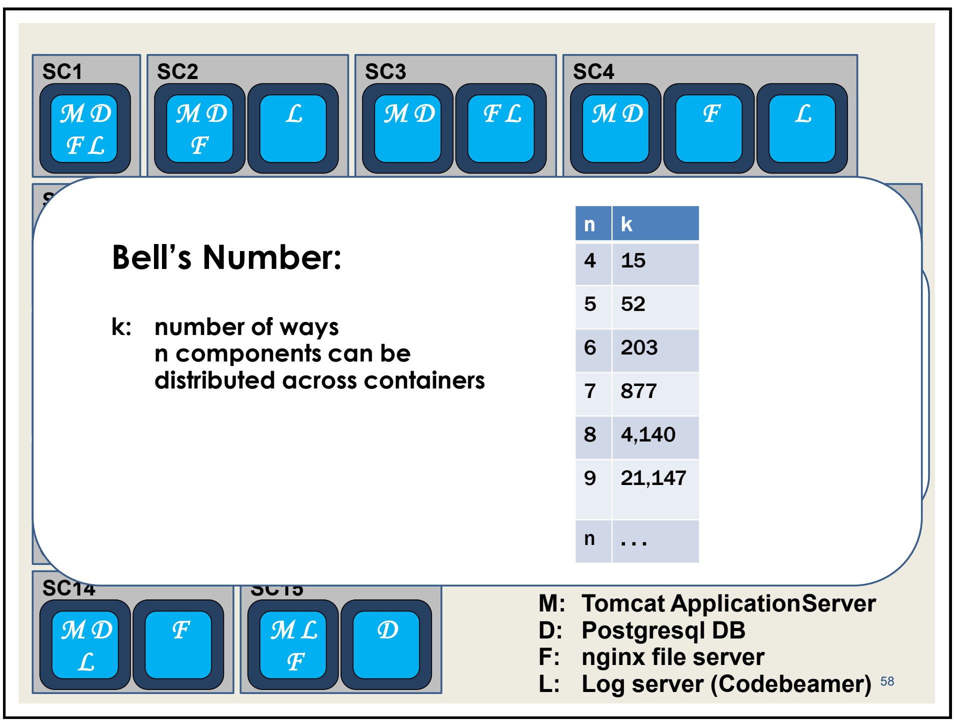
- Cloud provider maps VMs to physical servers
- User controls mapping of services to VM images
  - Should components be separated (isolated) ?
  - Should components be combined ?
  - M – modeling web service
  - D – relational database
  - F – file server
  - L – logging server



56



57

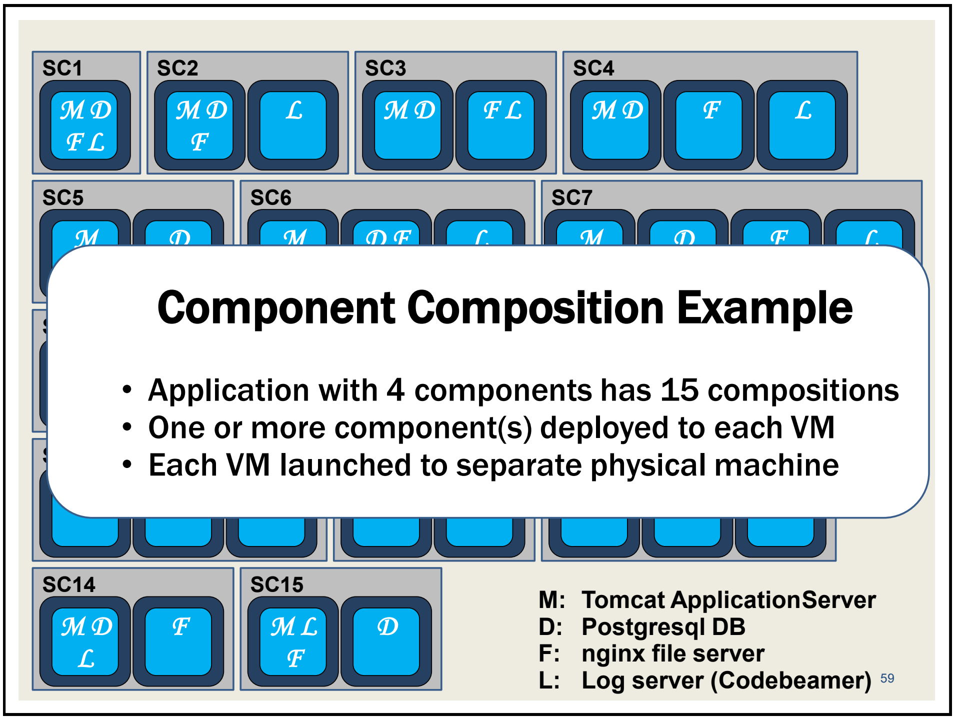


Bell's Number:

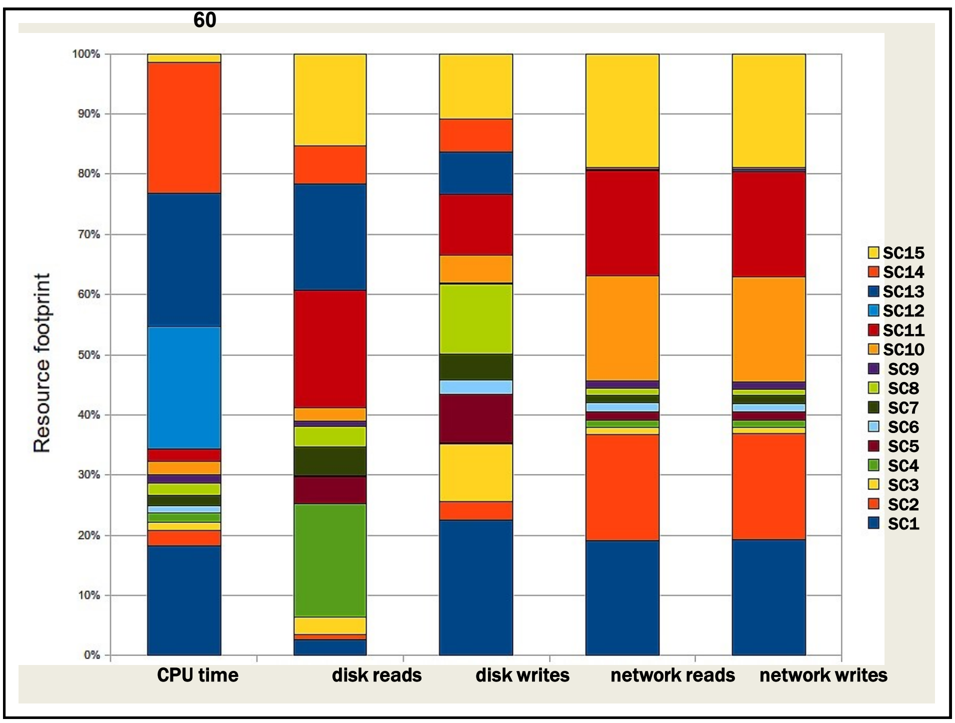
k: number of ways n components can be distributed across containers

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

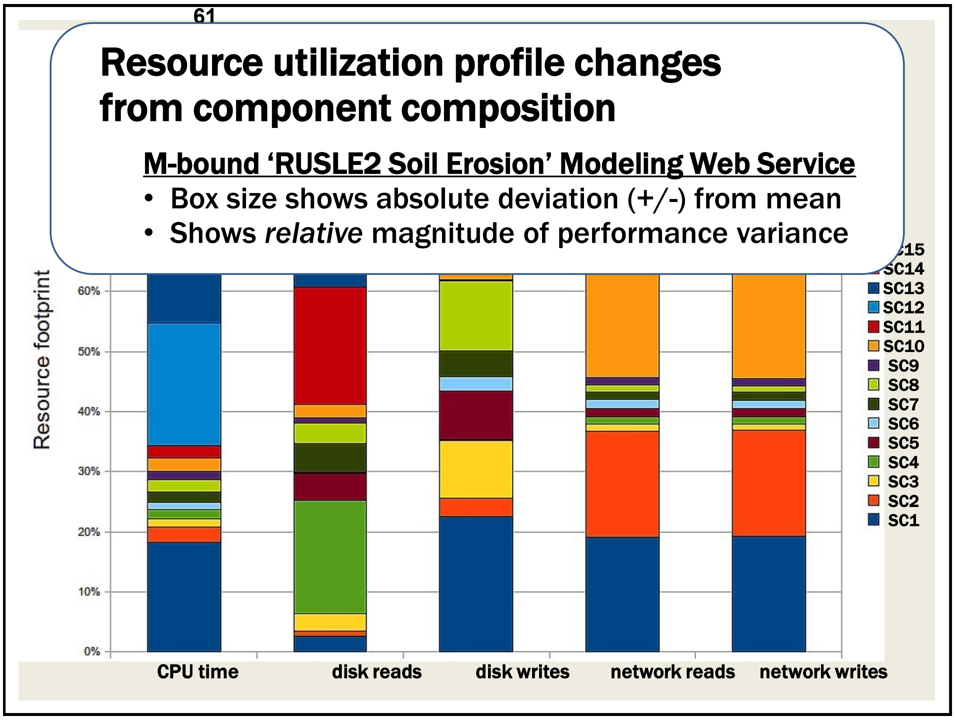
58


SC1
SC2
SC3
SC4
SC5
SC6
SC7
SC14
SC15
Component Composition Example
• Application with 4 components has 15 compositions
• One or more component(s) deployed to each VM
• Each VM launched to separate physical machine
M: Tomcat ApplicationServer
D: Postgresql DB
F: nginx file server
L: Log server (Codebeamer)
59

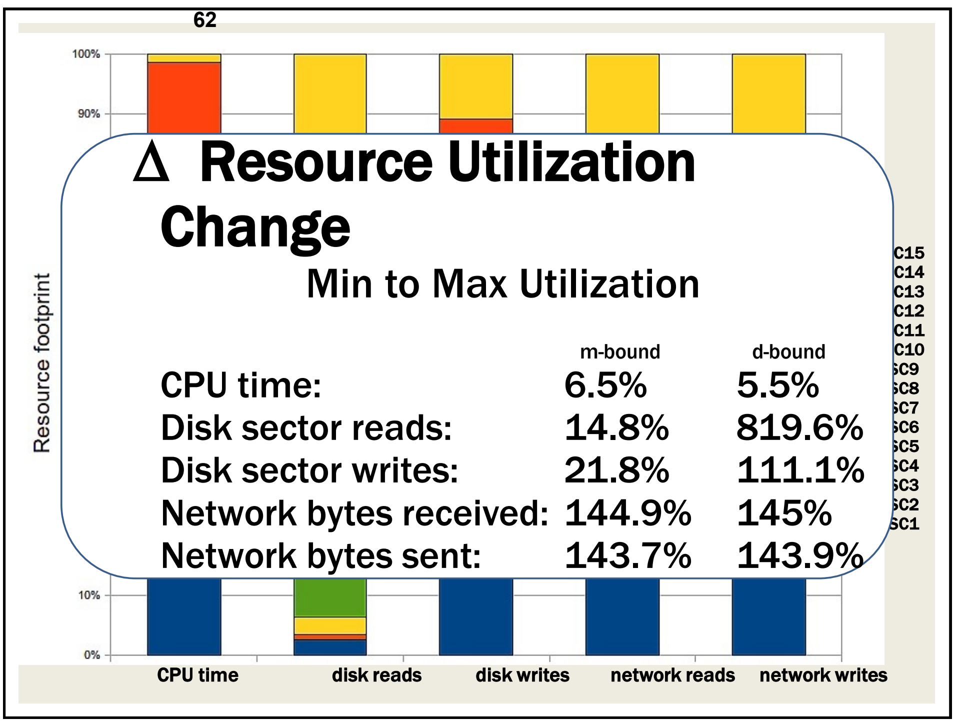
59


60
Resource footprint
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
CPU time
disk reads
disk writes
network reads
network writes
SC15
SC14
SC13
SC12
SC11
SC10
SC9
SC8
SC7
SC6
SC5
SC4
SC3
SC2
SC1

60


61
Resource utilization profile changes from component composition
M-bound 'RUSLE2 Soil Erosion' Modeling Web Service
• Box size shows absolute deviation (+/-) from mean
• Shows relative magnitude of performance variance
Resource footprint
60%
50%
40%
30%
20%
10%
0%
CPU time
disk reads
disk writes
network reads
network writes
SC14
SC13
SC12
SC11
SC10
SC9
SC8
SC7
SC6
SC5
SC4
SC3
SC2
SC1

61

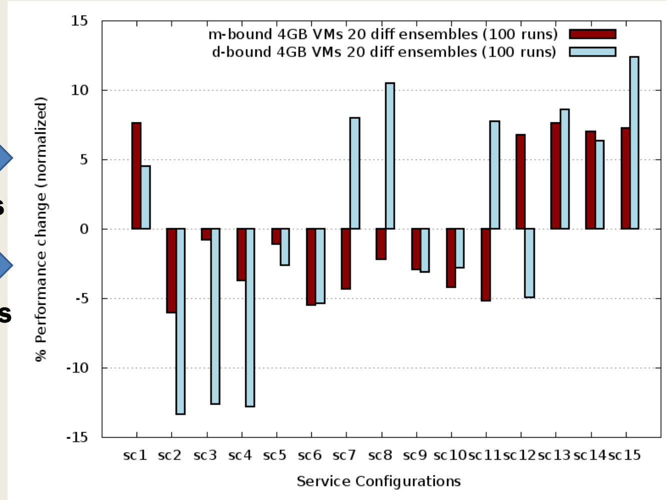

62
Δ Resource Utilization Change
Min to Max Utilization
m-bound d-bound
CPU time: 6.5% 5.5%
Disk sector reads: 14.8% 819.6%
Disk sector writes: 21.8% 111.1%
Network bytes received: 144.9% 145%
Network bytes sent: 143.7% 143.9%
Resource footprint
100%
90%
10%
0%
CPU time
disk reads
disk writes
network reads
network writes

62

PERFORMANCE IMPLICATIONS OF APPLICATION DEPLOYMENTS

Slower deployments

Faster deployments



63

PERFORMANCE IMPLICATIONS OF APPLICATION DEPLOYMENTS

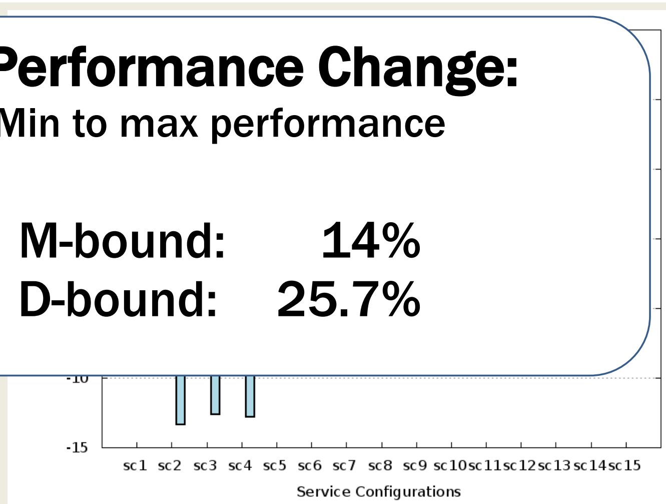
Slo

Fa

∆ Performance Change:

Min to max performance

M-bound: 14%
D-bound: 25.7%



64

CLOUD COMPUTING DELIVERY MODELS

- Infrastructure-as-a-Service (IaaS)
- Platform-as-a-Service (PaaS)
- Software-as-a-Service (SaaS)

Serverless Computing:

- Function-as-a-Service (FaaS)
- Container-as-a-Service (CaaS)
- Other Delivery Models

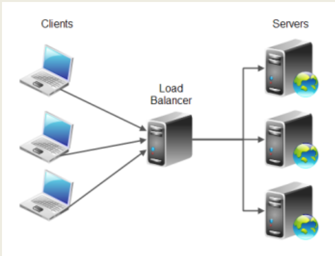
October 23, 2025	TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma	L8.65

65

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)



October 23, 2025	TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma	L8.66

66

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 23, 2025	TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma	L8.67

67

CLOUD COMPUTING DELIVERY MODELS

- Infrastructure-as-a-Service (IaaS)
- Platform-as-a-Service (PaaS)
- Software-as-a-Service (SaaS)

Serverless Computing:

- Function-as-a-Service (FaaS)
- Container-as-a-Service (CaaS)
- Other Delivery Models

October 23, 2025	TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma	L8.68

68

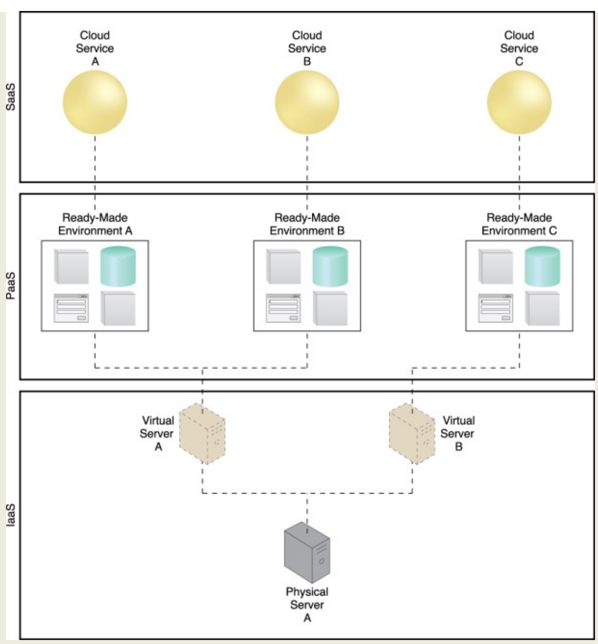
SOFTWARE-AS-A-SERVICE

- 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 23, 2025	TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma	L8.69

69



October 23, 2025 TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025]
School of Engineering and Technology, University of Washington - Tacoma L8.70

70

CLOUD COMPUTING DELIVERY MODELS

- Infrastructure-as-a-Service (IaaS)
- Platform-as-a-Service (PaaS)
- Software-as-a-Service (SaaS)

Serverless Computing:

- Function-as-a-Service (FaaS)
- Container-as-a-Service (CaaS)
- Other Delivery Models

October 23, 2025	TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma	L8.71

71

SERVERLESS COMPUTING

Introducing Cloud 2.0

Image from: https://mobisoftinfotech.com/resources/blog/serverless-computing-deploy-applications-without-fiddling-with-servers/

72

SERVERLESS COMPUTING

How should my app withstand a server **failing**?

How can I tell if a server has been **compromised**?

How can I increase **utilization** of my servers?

Which **OS** should my servers run?

How much remaining **capacity** do my servers have?

How should I implement dynamic **configuration changes** on my servers

When should I decide to **scale up** my servers?

What size servers are right for my budget?

How will I keep my server OS **patched**?

How can I control **access from** my servers?

Which packages should be baked into my **server images**?

Servers

(AAHHHHHHHHH!!)

How will new code be **deployed** to my servers?

How will the application handle server **hardware failure**?

How many users create **too much load** for my servers?

What size server is right for my **performance**?

Which users should have **access to** my servers?

Should I **tune OS settings** to optimize my application?

When should I decide to **scale out** my servers?

How many servers should I budget for?

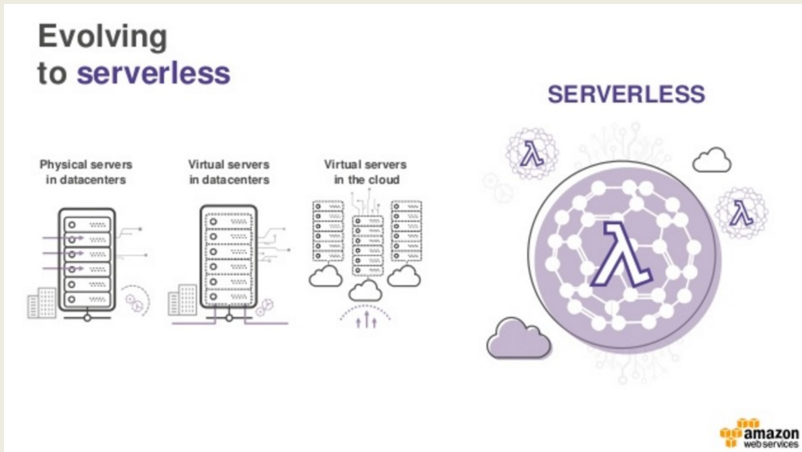
73

SERVERLESS COMPUTING

October 23, 2025	TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma	L8.74

74

SERVERLESS COMPUTING - 2



SERVERLESS COMPUTING

- Pay only for CPU/memory utilization
- High Availability
- Fault Tolerance
- Infrastructure Elasticity
- No Setup
- Function-as-a-Service (FAAS)

SERVERLESS COMPUTING

Why Serverless Computing?

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

...they are built into the platform

77

CLOUD COMPUTING DELIVERY MODELS

- Infrastructure-as-a-Service (IaaS)
- Platform-as-a-Service (PaaS)
- Software-as-a-Service (SaaS)

Serverless Computing:

- $\boxed{\text{Function-as-a-Service (FaaS)}}$
- Container-as-a-Service (CaaS)
- Other Delivery Models

October 23, 2025	TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma	L8.78

78

SERVERLESS VS. FAAS

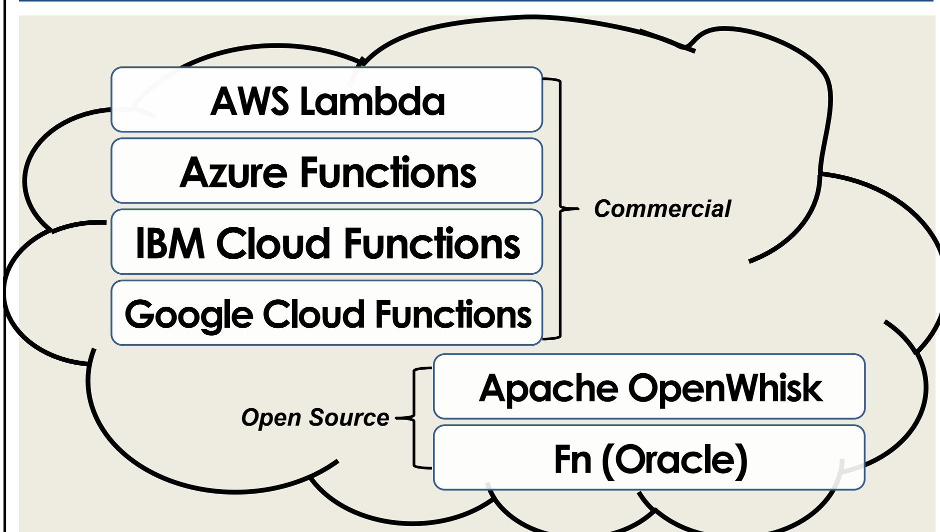
- **Serverless Computing**
- Refers to the avoidance of managing servers
- Can pertain to a number of "as-a-service" cloud offerings
- Function-as-a-Service (FaaS)
  - Developers write small code snippets (microservices) which are deployed separately
- Database-as-a-Service (DBaaS)
- Container-as-a-Service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

- Serverless is a buzzword
- This space is evolving…

October 23, 2025	TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma	L8.79

79

FAAS PLATFORMS



80

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

81

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

82

FAAS PLATFORMS - 3

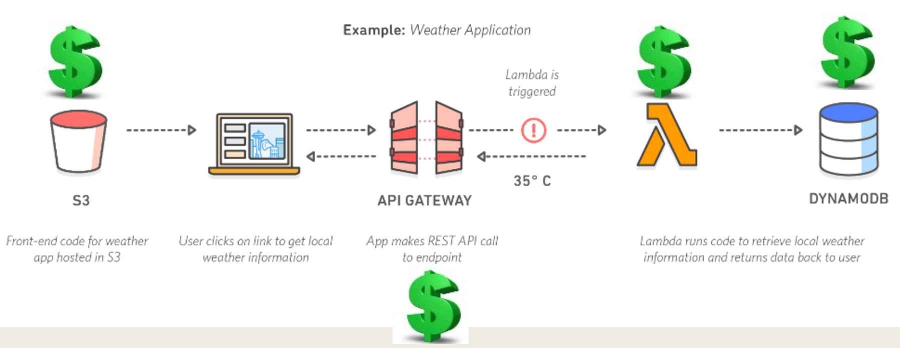
- Many challenging features of distributed systems are provided automatically
- ***<u>Built into the platform:</u>***
- Highly availability (24/7)
- Scalability
- Fault tolerance

October 23, 2025	TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma	L8.83

83

CLOUD NATIVE SOFTWARE ARCHITECTURE

- Every service with a different pricing model



October 23, 2025	TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma	L8.84

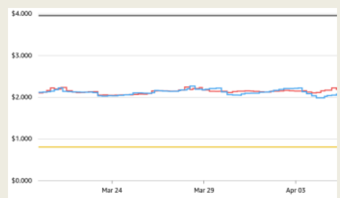
84

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 →



PRICING OBFUSCATION

- **VM pricing:** hourly rental pricing, billed to nearest second is intuitive…
- **FaaS pricing:** non-intuitive pricing policies
- **FREE TIER:**
  first 1,000,000 function calls/month → FREE
  first 400,000 GB-sec/month → FREE
- Afterwards: *obfuscated pricing (AWS Lambda):*
  $0.0000002 per request
  $0.000000208 to rent 128MB / 100-ms
  $0.00001667 GB /second

WEBSERVICE HOSTING EXAMPLE

- **ON AWS Lambda**
- Each service call: 100% of 2 CPU-cores
  100% of 4GB of memory
- Workload: uses 2 continuous threads
- Duration: 1 month (30.41667 days)

- **ON AWS EC2:** Amazon EC2 c5.large 2-vCPU VM x 4GB
- c5.large: 8.5¢/hour, 24 hrs/day x 30.41667 days
- Hosting cost: $62.05/month

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

87

PRICING OBFUSCATION

Assume 1 month = 30.41667 days (365d / 12)

- Workload: (4 GB) 10,512,000 GB-sec

Worst-case FaaS scenario = ~2.72x !

AWS EC2: $62.05
AWS Lambda: $168.91

Break Even: 3,702,459 GB-sec
@4GB ~10.71 days

- **BREAK-EVEN POINT: $62.05 - $0.33 (calls) = $61.72**
- $61.72/.00001667 GB-sec = ~3,702,459 GB-sec-mon/4GB/call= ~925,614 sec or ~10.71 days

Point at which using FaaS costs the same as IaaS

88

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 23, 2025	TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma	L8.89

89

FAAS CHALLENGES

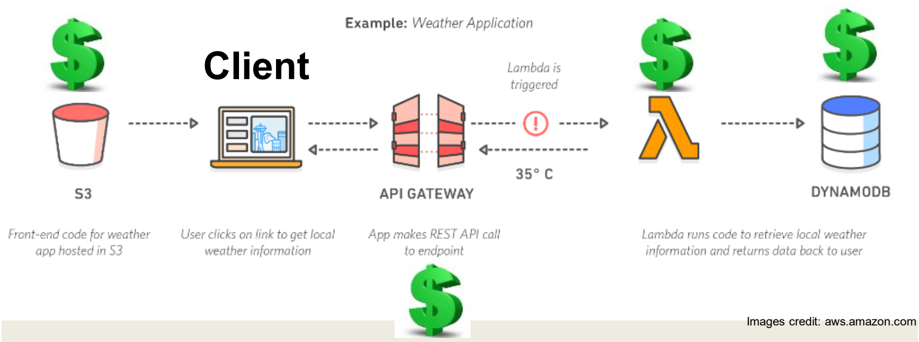
- Vendor architectural lock-in – how to migrate?
- Pricing obfuscation – is it cost effective?
- Memory reservation – how much to reserve?
- Service composition – how to compose software?
- Infrastructure freeze/thaw cycle – how to avoid?
- Performance – what will it be?

October 23, 2025	TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma	L8.90

90

VENDOR ARCHITECTURAL LOCK-IN

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



Images credit: aws.amazon.com

- Increased dependencies → increased hosting costs

91

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 400,000 GB-sec/month → FREE

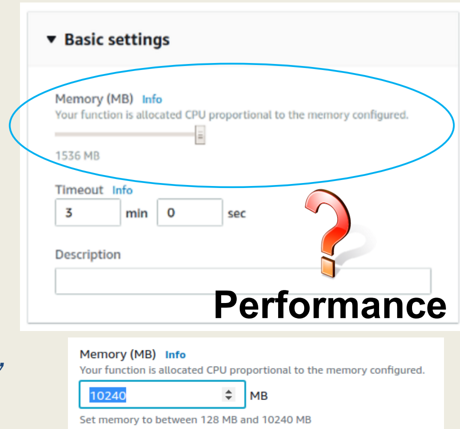
- Afterwards: $0.0000002 per request
$0.000000208 to rent 128MB / 100-ms

October 23, 2025	TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma	L8.92

92

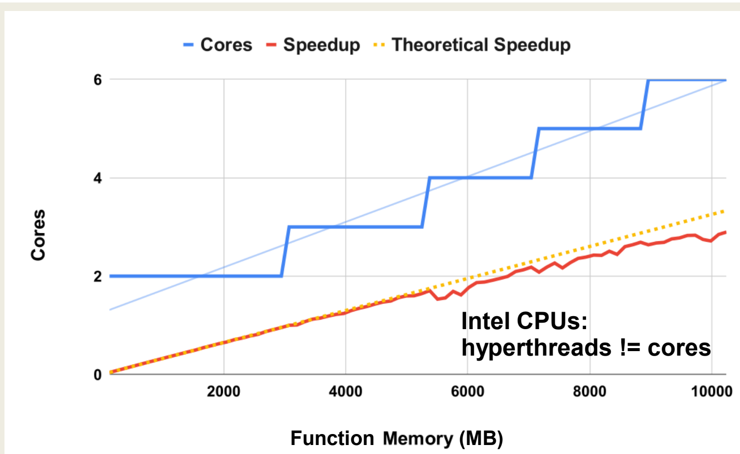
MEMORY RESERVATION QUESTION…

- Lambda memory reserved for functions
- UI provides text box formerly "slider bar" to set function's memory
- Resource capacity (CPU, disk, network) coupled to slider bar: *"every **doubling** of memory, **doubles** CPU…"*



- **But how much memory do FaaS functions require?**

AWS LAMBDA COUPLES FUNCTION MEMORY TO CPU CORES & TIME SHARE

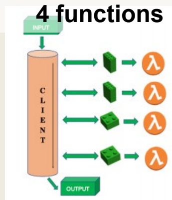


SERVICE COMPOSITION

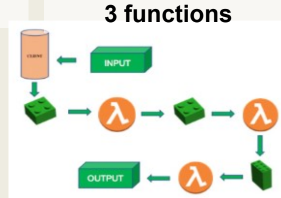
- **How should application code be composed for deployment to serverless computing platforms?**

Monolithic Deployment

Client flow control, 4 functions



Server flow control, 3 functions



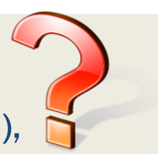
- Recommended practice:
  Decompose into many microservices
- Platform limits: code + libraries ~250MB
- **How does composition impact the number of function invocations, and memory utilization?**

Performance

95

INFRASTRUCTURE FREEZE/THAW CYCLE

- Unused infrastructure is deprecated
  - *But after how long? (varies by platform)*
- Infrastructure: microVMs (on AWS Lambda), containers on some platforms



Performance

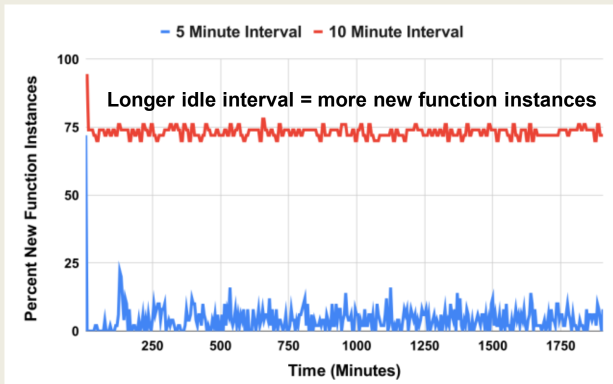
- <u>COLD</u>
  - Code image - built/transferred to physical host & cached
- <u>WARM</u>
  - Host has local code cache – create function instance (microVM) on host
- <u>HOT</u>
  - Function instance ready to use

Image from: Denver7 – The Denver Channel News

96

AWS LAMBDA – FREEZE/THAW

- Experiment: 50 concurrent calls, 5 or 10-min calling interval
- Evaluate % cold function instances



October 23, 2025 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma | L8.97

97

FACTORS IMPACTING PERFORMANCE OF FAAS COMPUTING PLATFORMS

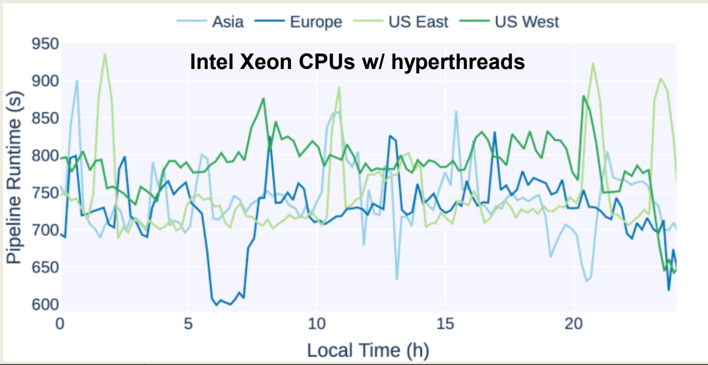
- Infrastructure scaling/elasticity
- Resource contention (CPU, network, memory caches)
- Hardware heterogeneity (CPU types, hyperthread, etc)
- Load balancing / provisioning variation
- Infrastructure retention: COLD vs. WARM
  - Infrastructure freeze/thaw cycle
- Function memory reservation size
- Application service composition

October 23, 2025 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma | L8.98

98

AWS LAMBDA PERFORMANCE VARIATION

- NLP processing pipeline use case
- Performance variance from: diurnal changes in load (e.g. resource contention), Intel hyperthreading

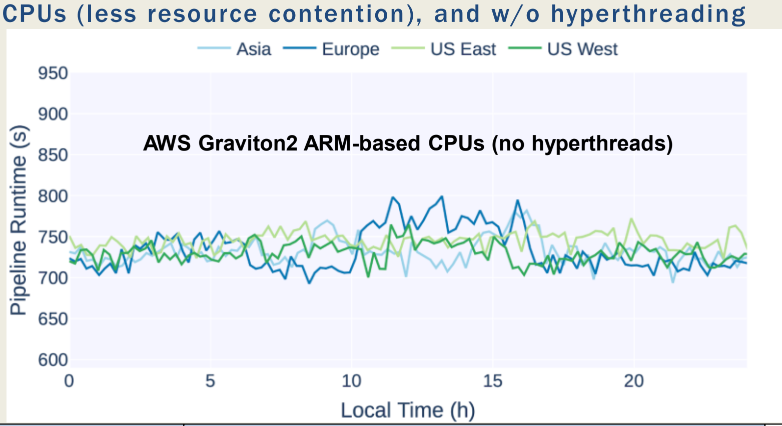


October 23, 2025	TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma	L8.99

99

AWS LAMBDA PERFORMANCE VARIATION - 2

- NLP use case: Less performance variance using ARM-based CPUs (less resource contention), and w/o hyperthreading



October 23, 2025	TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma	L8.100

100

FUNCTION-AS-A-SERVICE

AWS Lambda Demo

101

101

CLOUD COMPUTING DELIVERY MODELS

- Infrastructure-as-a-Service (IaaS)
- Platform-as-a-Service (PaaS)
- Software-as-a-Service (SaaS)

Serverless Computing:

- Function-as-a-Service (FaaS)
- Container-as-a-Service (CaaS)
- Other Delivery Models

October 23, 2025	TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025] School of Engineering and Technology, University of Washington - Tacoma	L8.102

102

CONTAINER-AS-A-SERVICE

- Cloud service model for deploying application containers (e.g. Docker containers) 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
  - Google Cloud Run
  - Azure Container Instances

CLOUD COMPUTING DELIVERY MODELS

- Infrastructure-as-a-Service (IaaS)
- Platform-as-a-Service (PaaS)
- Software-as-a-Service (SaaS)

Serverless Computing:

- Function-as-a-Service (FaaS)
- Container-as-a-Service (CaaS)
- Other Delivery Models

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

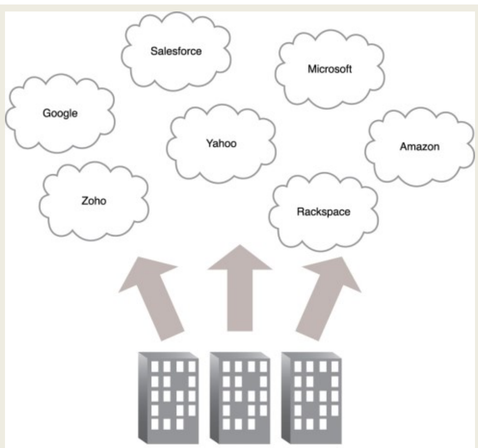
OBJECTIVES – 10/23

- Questions from 10/21
- Tutorials Questions
- Tutorial 4 - to be posted...
- **From: Cloud Computing Concepts, Technology & Architecture: Chapter 4: Cloud Computing Concepts and Models:**
  - Cloud computing delivery models
  - **Cloud deployment models**
- AWS Overview and demo
- 2nd hour:
  - Activity 2 - Intro to EC2
  - Term Project Planning

CLOUD DEPLOYMENT MODELS

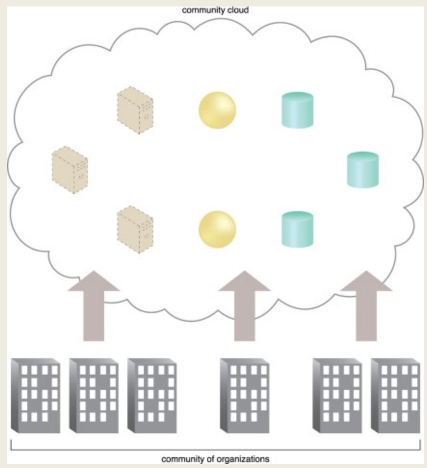
- Distinguished by ownership, size, access
- Four common models
  - Public cloud
  - Community cloud
  - Hybrid cloud
  - Private cloud

PUBLIC CLOUDS



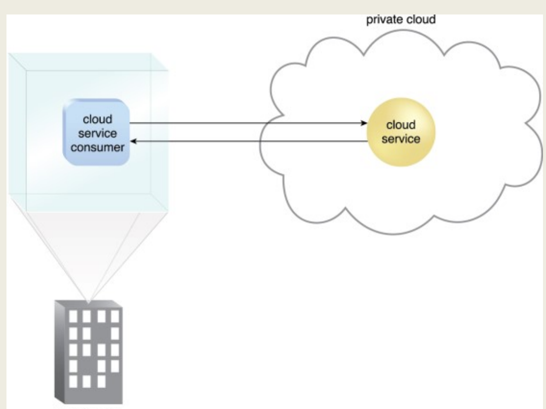
COMMUNITY CLOUD

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



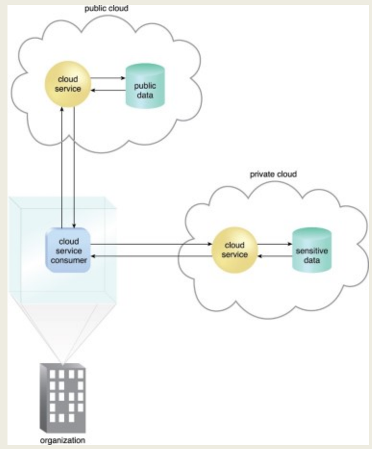
PRIVATE CLOUD

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



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



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

WE WILL RETURN AT 5:50 PM

113

OBJECTIVES – 10/23

- **Questions from 10/21**
- Tutorials Questions
- Tutorial 4 - to be posted...
- **From: Cloud Computing Concepts, Technology & Architecture: Chapter 4: Cloud Computing Concepts and Models:**
  - Cloud computing delivery models
  - Cloud deployment models
- AWS Overview and demo
- 2nd hour:
  - **Activity 2 - Intro to EC2**
  - Term Project Planning

149

QUESTIONS

October 23, 2025
TCSS462/562:(Software Engineering for) Cloud Computing [Fall 20]
School of Engineering and Technology, University of Washington - T oma
L8.152

152