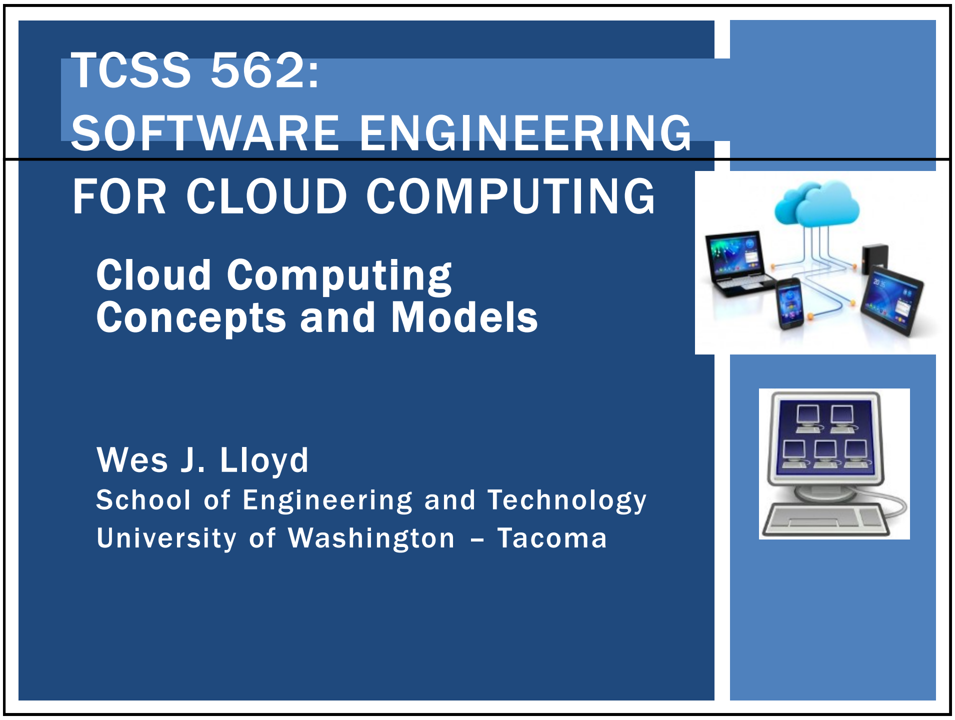


TCSS 562: SOFTWARE ENGINEERING FOR CLOUD COMPUTING

Cloud Computing Concepts and Models

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

1

OFFICE HOURS – FALL 2023

- **Tuesdays:**
  - 2:30 to 3:30 pm - CP 229
- **Fridays**
  - 11:00 am to 12:00 pm – ONLINE via Zoom
- Or email for appointment

> Office Hours set based on Student Demographics survey feedback

October 19, 2023 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma | L7/2

2

OBJECTIVES – 10/19

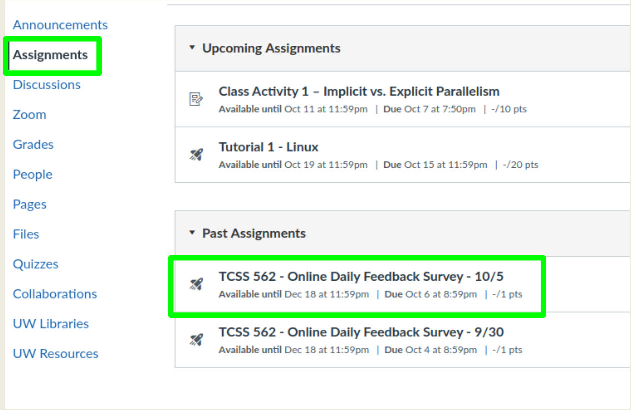
- **Questions from 10/17**
- Tutorials Questions
- Tutorial 4 – Intro to FaaS – AWS Lambda
- Background on AWS Lambda for the Term Project - II
- **From: Cloud Computing Concepts, Technology & Architecture: Chapter 4: Cloud Computing Concepts and Models:**
  - Roles and boundaries
  - Cloud characteristics
  - Cloud delivery models
  - Cloud deployment models
- Team Planning - Breakout Rooms

October 19, 2023	TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma	L7.3

3

ONLINE DAILY FEEDBACK SURVEY

- Daily Feedback Quiz in Canvas – Take After Each Class
- Extra Credit for completing



▾ Upcoming Assignments

Class Activity 1 – Implicit vs. Explicit Parallelism
Available until Oct 11 at 11:59pm | Due Oct 7 at 7:50pm | -/10 pts

Tutorial 1 - Linux
Available until Oct 19 at 11:59pm | Due Oct 15 at 11:59pm | -/20 pts

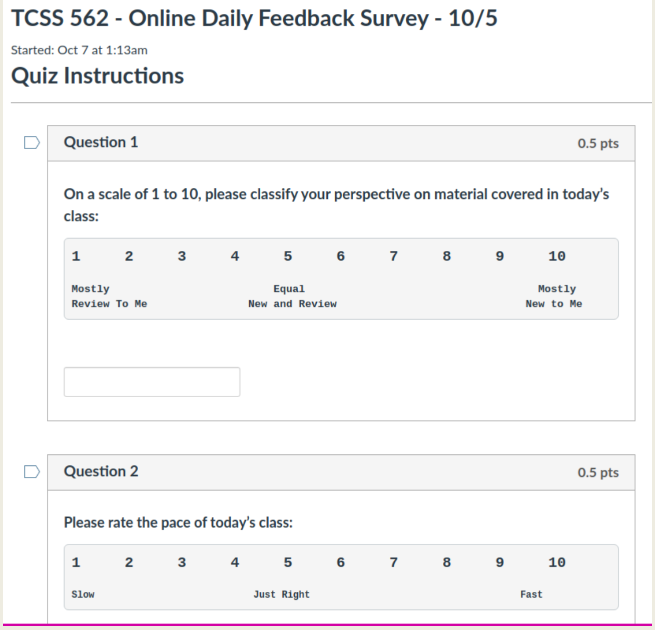
▾ Past Assignments

TCSS 562 - Online Daily Feedback Survey - 10/5
Available until Dec 18 at 11:59pm | Due Oct 6 at 8:59pm | -/1 pts

TCSS 562 - Online Daily Feedback Survey - 9/30
Available until Dec 18 at 11:59pm | Due Oct 4 at 8:59pm | -/1 pts

October 19, 2023	TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma	L7.4

4



TCSS 562 - Online Daily Feedback Survey - 10/5

Started: Oct 7 at 1:13am

Quiz Instructions

Question 1 — 0.5 pts

On a scale of 1 to 10, please classify your perspective on material covered in today's class:

1	2	3	4	5	6	7	8	9	10
Mostly Review To Me				Equal New and Review					Mostly New to Me

Question 2 — 0.5 pts

Please rate the pace of today's class:

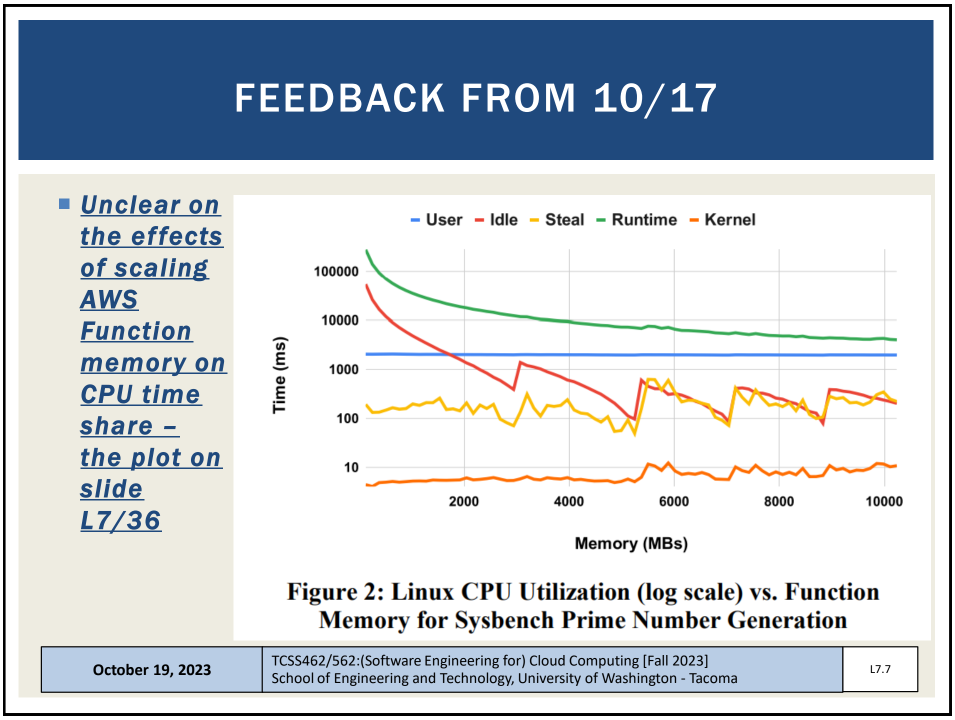
1	2	3	4	5	6	7	8	9	10
Slow				Just Right					Fast

MATERIAL / PACE

- Please classify your perspective on material covered in today's class (51 respondents):
- 1-mostly review, 5-equal new/review, 10-mostly new
- **Average – 6.55 (↓ - *previous* 6.29)**

- Please rate the pace of today's class:
- 1-slow, 5-just right, 10-fast
- **Average – 5.64 (↓ - previous 5.60)**

- **Response rates:**
- TCSS 462: 34/44 – 77.3%
- TCSS 562: 17/25 – 68.0%



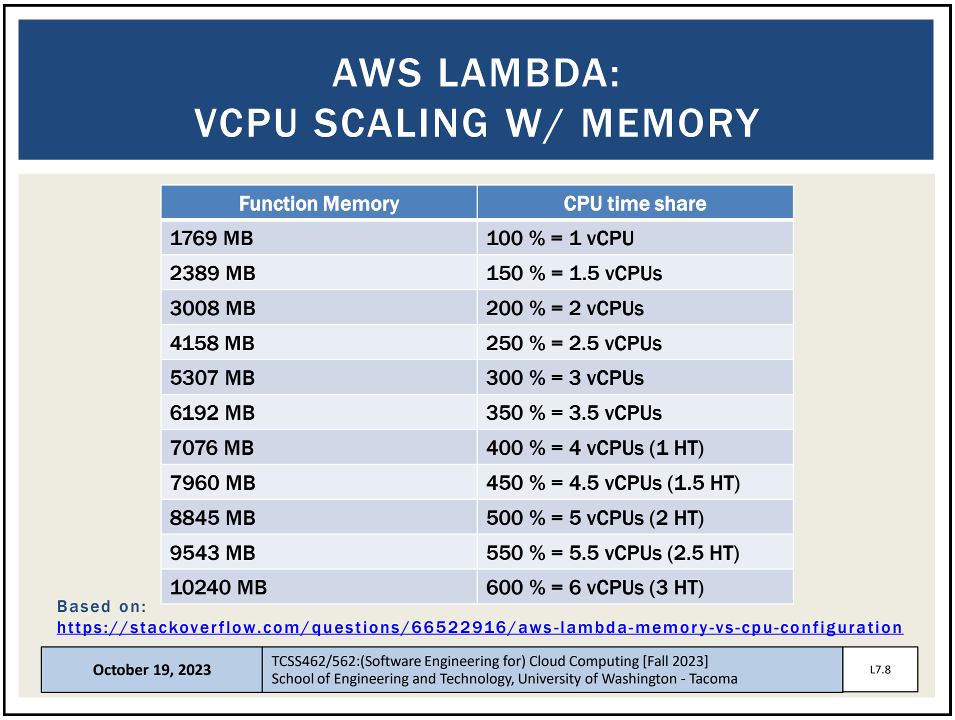
FEEDBACK FROM 10/17

- Unclear on the effects of scaling AWS Function memory on CPU time share – the plot on slide L7/36

Figure 2: Linux CPU Utilization (log scale) vs. Function Memory for Sysbench Prime Number Generation

October 19, 2023 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma | L7.7

7



AWS LAMBDA: VCPU SCALING W/ MEMORY

Function Memory	CPU time share
1769 MB	100 % = 1 vCPU
2389 MB	150 % = 1.5 vCPUs
3008 MB	200 % = 2 vCPUs
4158 MB	250 % = 2.5 vCPUs
5307 MB	300 % = 3 vCPUs
6192 MB	350 % = 3.5 vCPUs
7076 MB	400 % = 4 vCPUs (1 HT)
7960 MB	450 % = 4.5 vCPUs (1.5 HT)
8845 MB	500 % = 5 vCPUs (2 HT)
9543 MB	550 % = 5.5 vCPUs (2.5 HT)
10240 MB	600 % = 6 vCPUs (3 HT)

Based on: https://stackoverflow.com/questions/66522916/aws-lambda-memory-vs-cpu-configuration

October 19, 2023 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma | L7.8

8

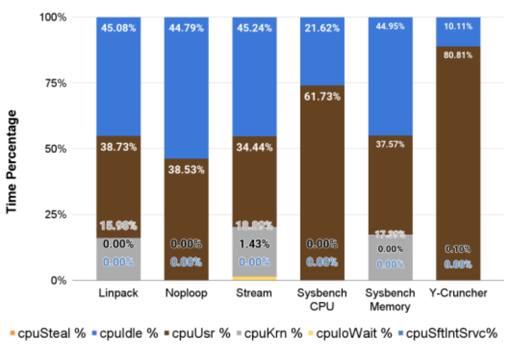
FEEDBACK - 2

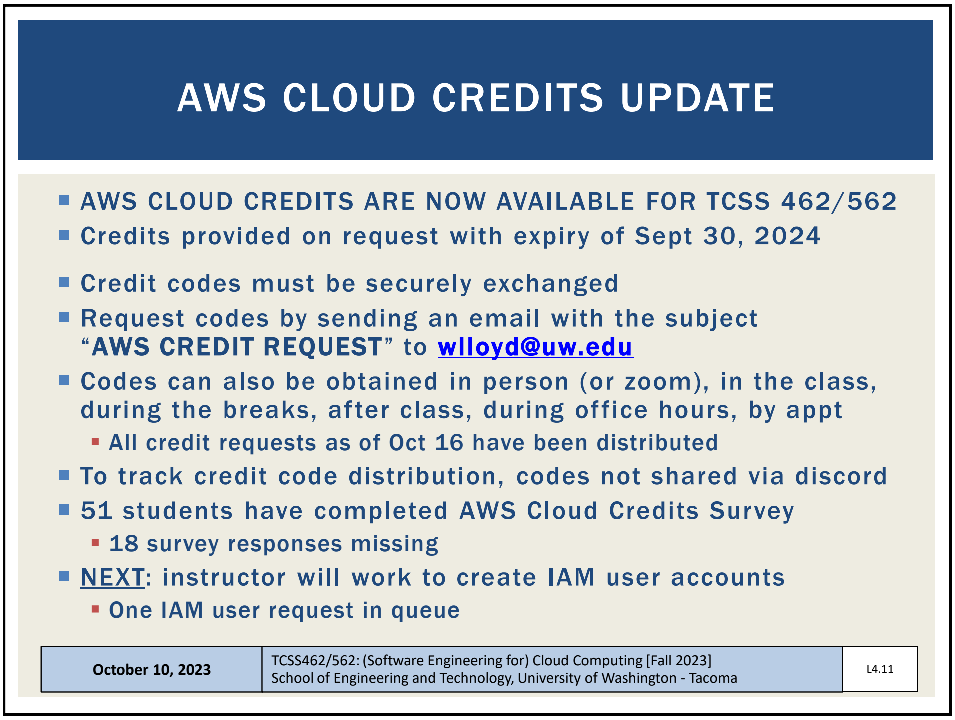
- ***Does AWS Lambda allow users to directly set their requested vCPU usage (→share) instead of indirectly through their RAM usage (→setting) ?***
- NO, the CPU time share is fixed based on function memory
- Same on other clouds: Google Cloud Functions, IBM Cloud Functions
- Azure Functions: if you want auto-scaling of function instances, use of the "consumption" plan is required where function instances are fixed with 1 vCPU and 1.5 GB RAM
  - Azure supports allocating VMs or containers with different sizes
  - See: https://learn.microsoft.com/en-us/azure/azure-functions/functions-scale
- ***If not is there any known reasoning for this?***
  - While VMs and containers support finely scaled resources in terms of vCPUs, cpu time share, and RAM, cloud providers do not allow users access to the full configurability presumably because this would leads to resource fragmentation and under utilization

FEEDBACK - 3

- ***Comparing the performance of the AWS Lambda based on different demands is still unclear.***
- CPU profiling let's us snapshot the CPU mode time distribution for a program or function
- This may mimic 'demand'

cpuIdle –	no action
cpuUsr –	run user code
cpuKrn –	run OS code
cpuIOwait –	wait for IO
cpuSftIntSrvc –	wait for interupt



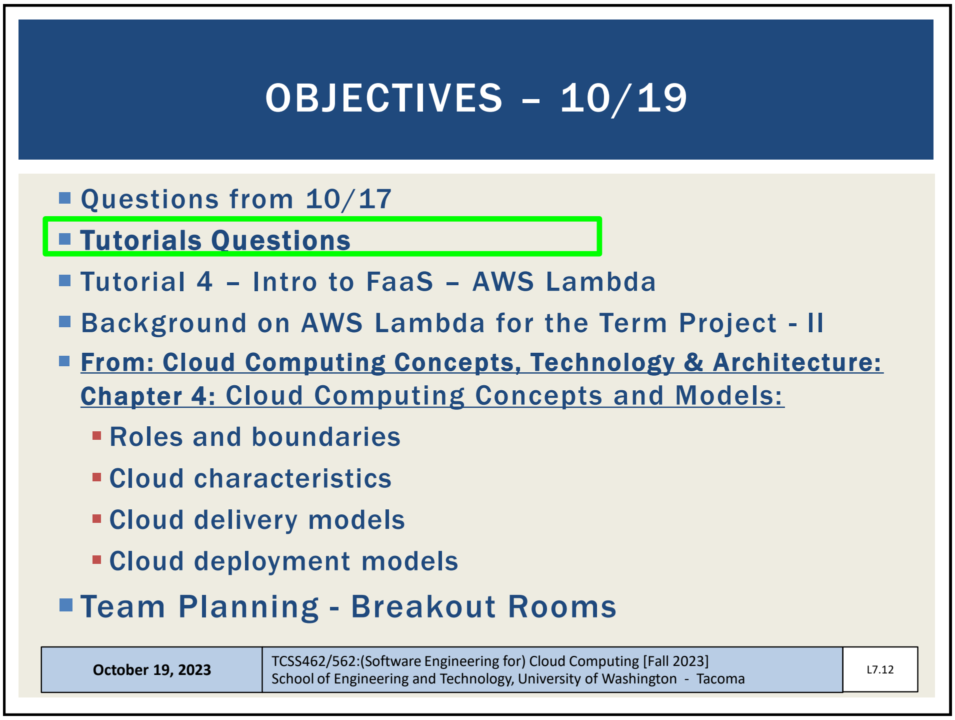


AWS CLOUD CREDITS UPDATE

- AWS CLOUD CREDITS ARE NOW AVAILABLE FOR TCSS 462/562
- Credits provided on request with expiry of Sept 30, 2024
- Credit codes must be securely exchanged
- Request codes by sending an email with the subject "**AWS CREDIT REQUEST**" to **wlloyd@uw.edu**
- Codes can also be obtained in person (or zoom), in the class, during the breaks, after class, during office hours, by appt
  - All credit requests as of Oct 16 have been distributed
- To track credit code distribution, codes not shared via discord
- 51 students have completed AWS Cloud Credits Survey
  - 18 survey responses missing
- NEXT: instructor will work to create IAM user accounts
  - One IAM user request in queue

October 10, 2023 | TCSS462/562: (Software Engineering for) Cloud Computing [Fall 2023]
School of Engineering and Technology, University of Washington - Tacoma | L4.11

11

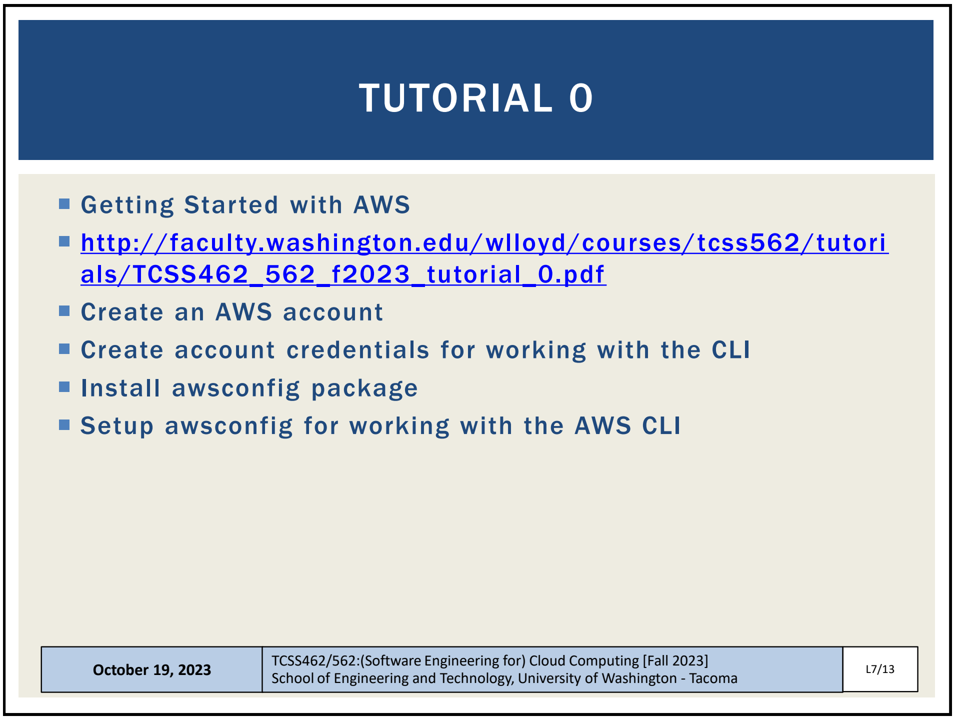


OBJECTIVES – 10/19

- Questions from 10/17
- **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 19, 2023 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023]
School of Engineering and Technology, University of Washington - Tacoma | L7.12

12

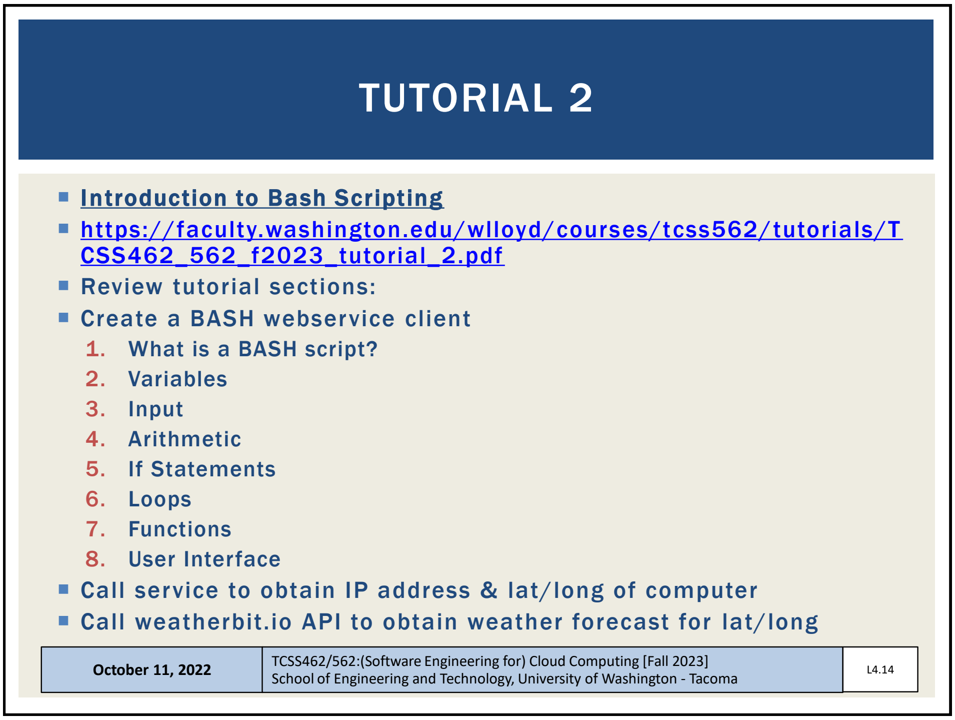


TUTORIAL 0

- Getting Started with AWS
- http://faculty.washington.edu/wlloyd/courses/tcss562/tutorials/TCSS462_562_f2023_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 19, 2023 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma | L7/13

13

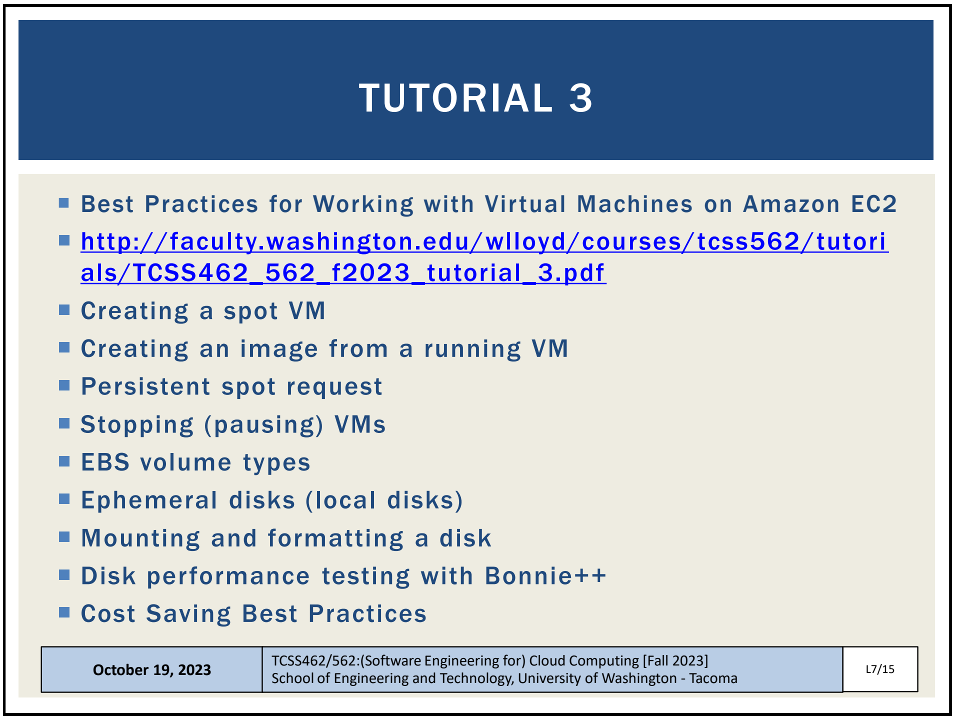


TUTORIAL 2

- Introduction to Bash Scripting
- https://faculty.washington.edu/wlloyd/courses/tcss562/tutorials/TCSS462_562_f2023_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 2023] School of Engineering and Technology, University of Washington - Tacoma | L4.14

14

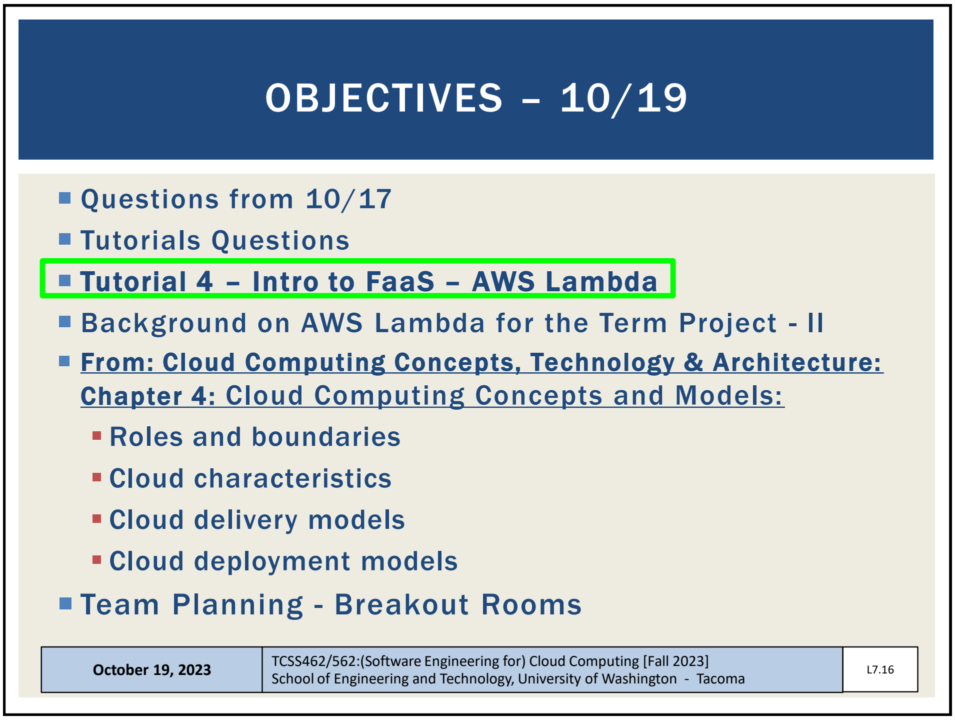


TUTORIAL 3

- Best Practices for Working with Virtual Machines on Amazon EC2
- http://faculty.washington.edu/wlloyd/courses/tcss562/tutorials/TCSS462_562_f2023_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

October 19, 2023 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma | L7/15

15

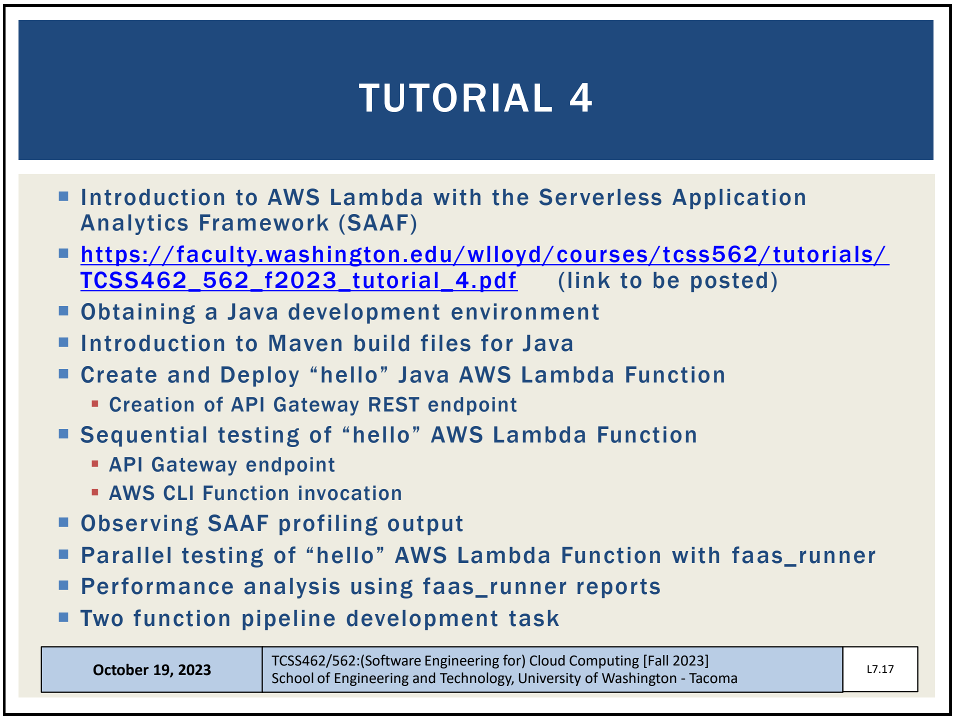


OBJECTIVES – 10/19

- Questions from 10/17
- 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 19, 2023 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma | L7.16

16

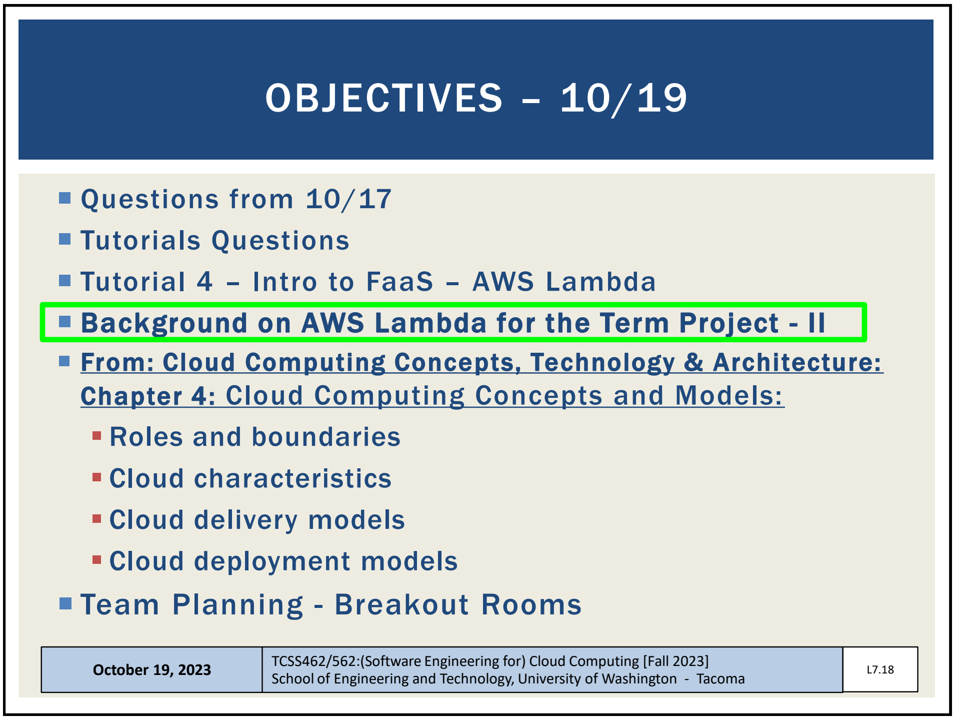


TUTORIAL 4

- Introduction to AWS Lambda with the Serverless Application Analytics Framework (SAAF)
- https://faculty.washington.edu/wlloyd/courses/tcss562/tutorials/TCSS462_562_f2023_tutorial_4.pdf (link to be posted)
- 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

October 19, 2023 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma | L7.17

17



OBJECTIVES – 10/19

- Questions from 10/17
- 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 19, 2023 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma | L7.18

18

FUNCTION INSTANCE LIFE CYCLES

- Function states:
- **COLD**: brand new function instance just initialized to run the request (more overhead)
  - Platform cold (first time ever run)
  - Host cold (function assets cached locally on servers)
- **WARM**: existing function instance that is reused
- All function instances persist for ~5 minutes before they begin to be "garbage collected" by the platform
  - 100% garbage collection may take up to ~30-40 minutes
- AWS Lambda appears to "recycle" infrastructure faster than other FaaS platforms
  - Presumably because of need, because the platform is busy

WARM VS COLD FUNCTION INSTANCES

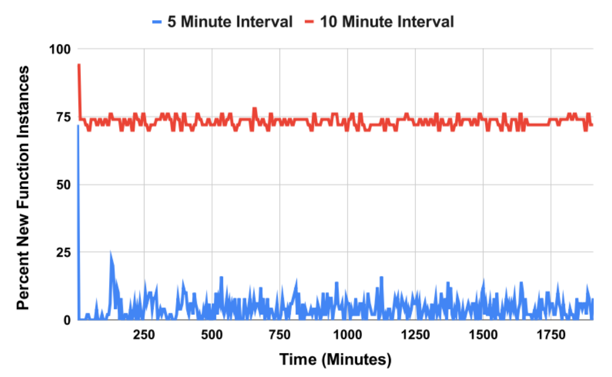


Figure 3: AWS Lambda Function Instance Replacement vs. Function Call Interval over 24-hours

CPU SCALING VS MEMORY: OTHER PLATFORMS: 0 TO 3 GB

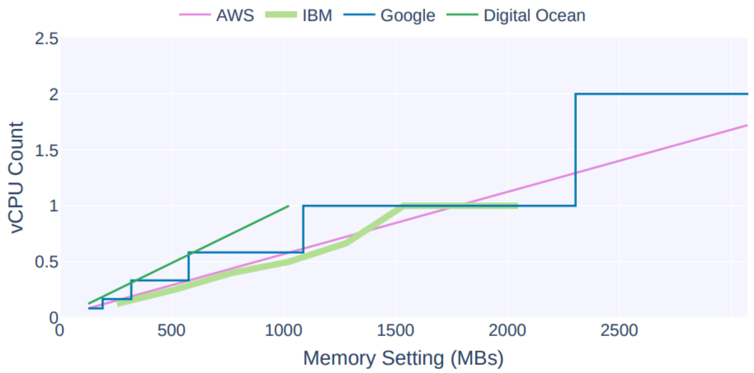
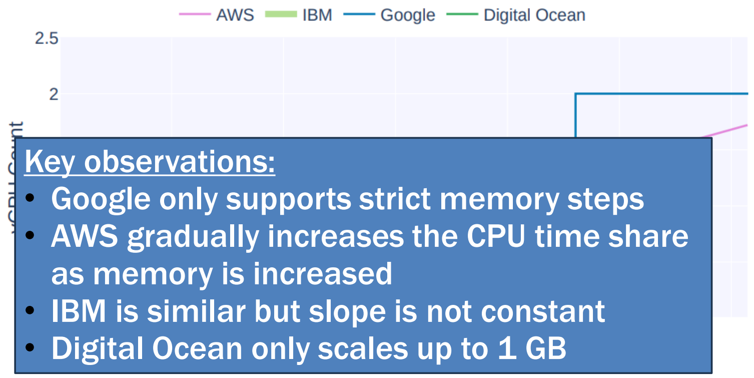


Fig. 2. Allocated vCPUs available at each memory setting on each platform.

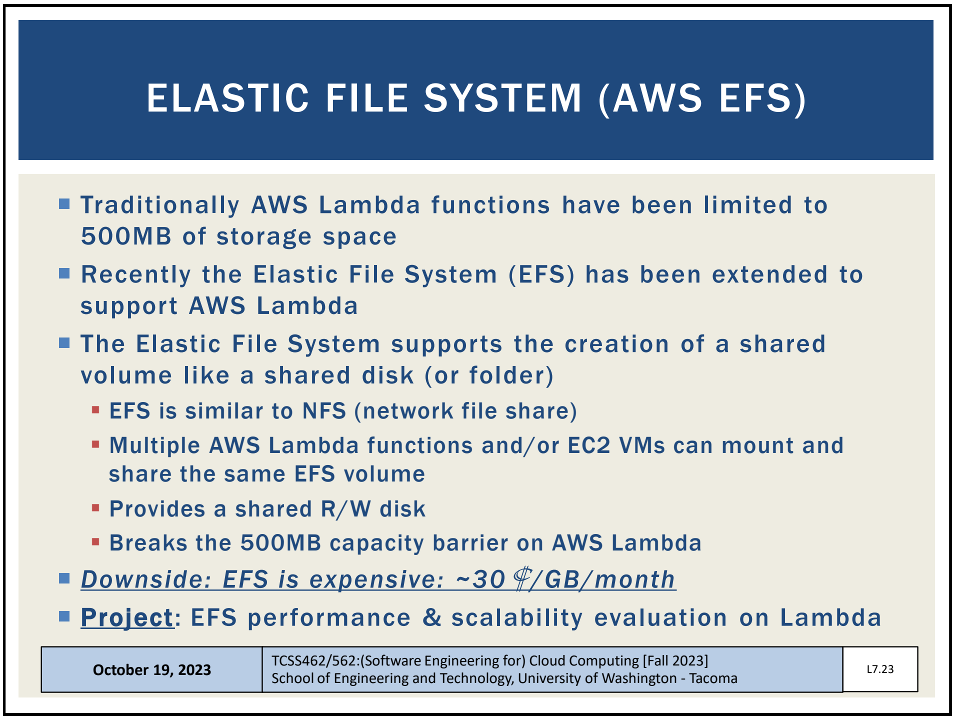
CPU SCALING VS MEMORY: OTHER PLATFORMS: 0 TO 3 GB



Key observations:

- Google only supports strict memory steps
- AWS gradually increases the CPU time share as memory is increased
- IBM is similar but slope is not constant
- Digital Ocean only scales up to 1 GB

Fig. 2. Allocated vCPUs available at each memory setting on each platform.

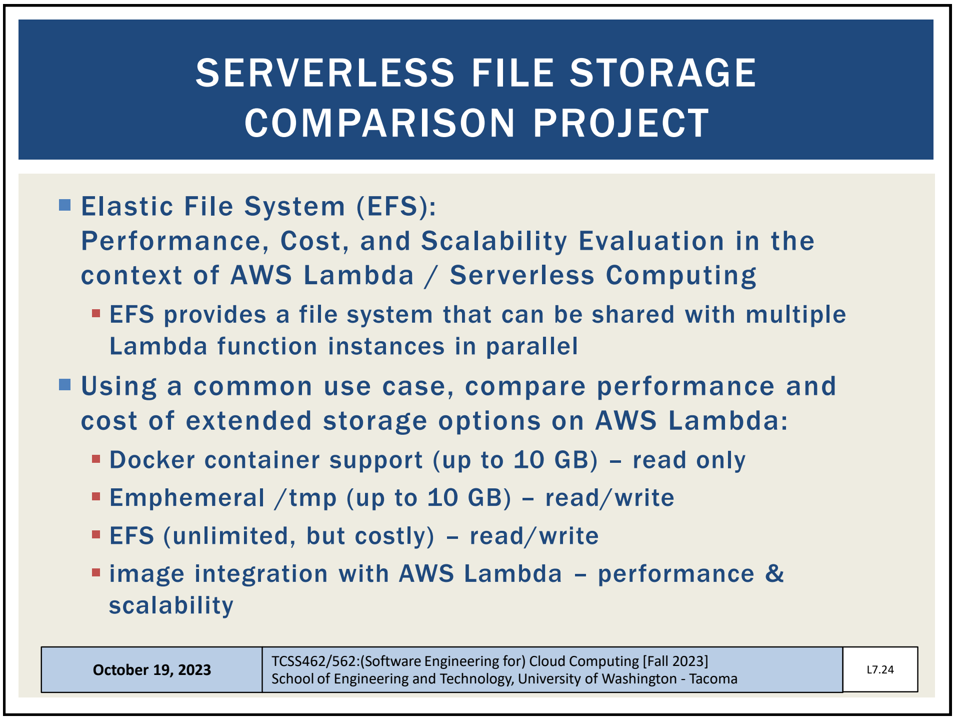


ELASTIC FILE SYSTEM (AWS EFS)

- Traditionally AWS Lambda functions have been limited to 500MB of storage space
- Recently the Elastic File System (EFS) has been extended to support AWS Lambda
- The Elastic File System supports the creation of a shared volume like a shared disk (or folder)
  - EFS is similar to NFS (network file share)
  - Multiple AWS Lambda functions and/or EC2 VMs can mount and share the same EFS volume
  - Provides a shared R/W disk
  - Breaks the 500MB capacity barrier on AWS Lambda
- *Downside: EFS is expensive: ~30 ¢/GB/month*
- **Project**: EFS performance & scalability evaluation on Lambda

October 19, 2023 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma | L7.23

23

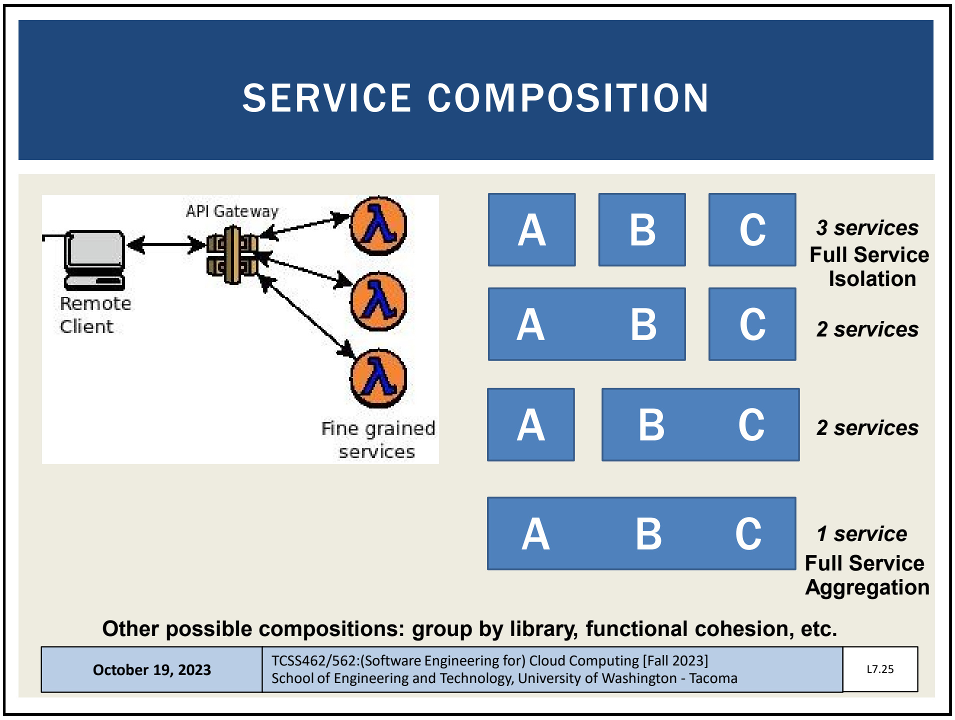


SERVERLESS FILE STORAGE COMPARISON PROJECT

- Elastic File System (EFS):
  Performance, Cost, and Scalability Evaluation in the context of AWS Lambda / Serverless Computing
  - EFS provides a file system that can be shared with multiple Lambda function instances in parallel
- Using a common use case, compare performance and cost of extended storage options on AWS Lambda:
  - Docker container support (up to 10 GB) – read only
  - Emphemeral /tmp (up to 10 GB) – read/write
  - EFS (unlimited, but costly) – read/write
  - image integration with AWS Lambda – performance & scalability

October 19, 2023 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma | L7.24

24



SERVICE COMPOSITION

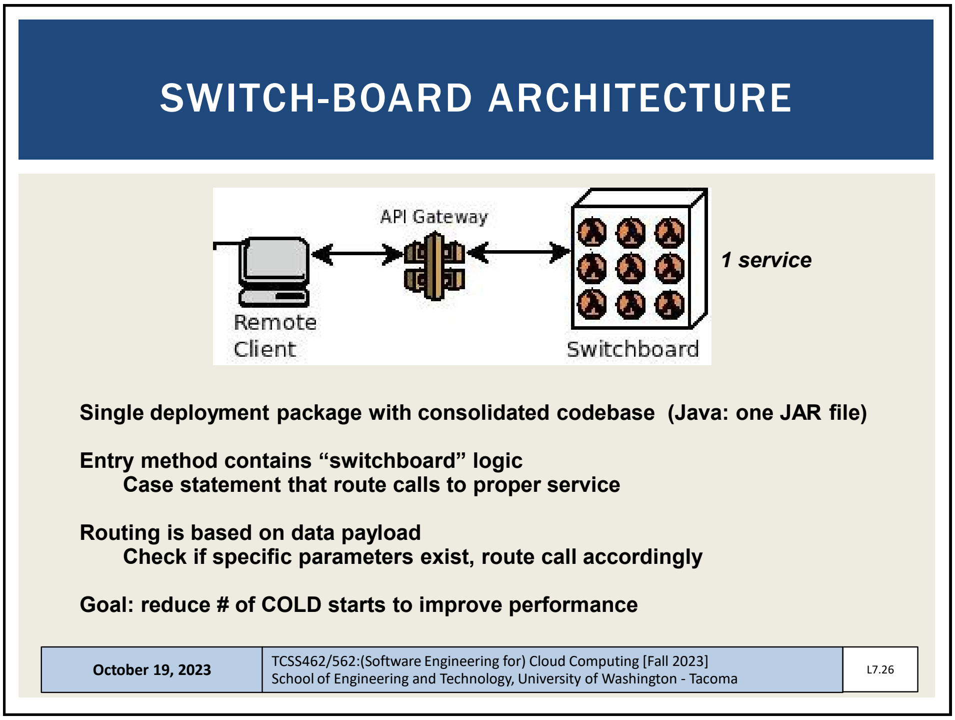
3 services **Full Service Isolation**

2 services

2 services

1 service **Full Service Aggregation**

Other possible compositions: group by library, functional cohesion, etc.



SWITCH-BOARD ARCHITECTURE

1 service

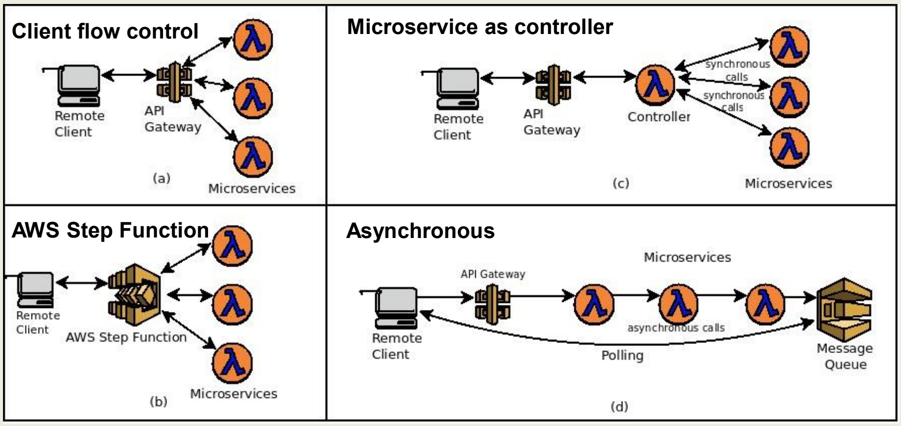
Single deployment package with consolidated codebase (Java: one JAR file)

Entry method contains “switchboard” logic
Case statement that route calls to proper service

Routing is based on data payload
Check if specific parameters exist, route call accordingly

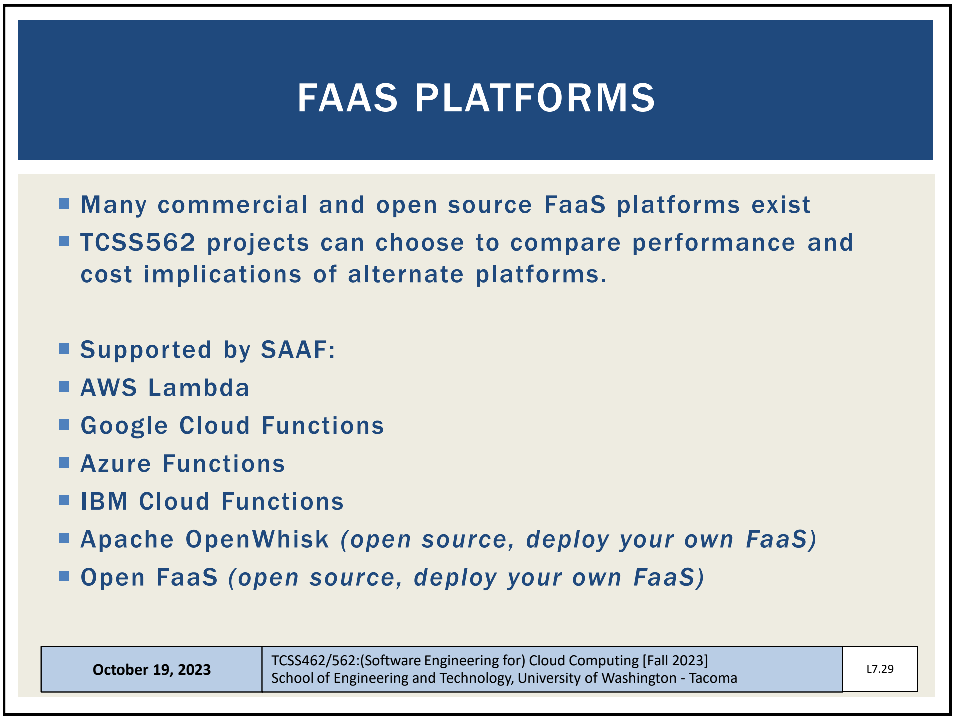
Goal: reduce # of COLD starts to improve performance

APPLICATION FLOW CONTROL - 3



PROGRAMMING LANGUAGE COMPARISON

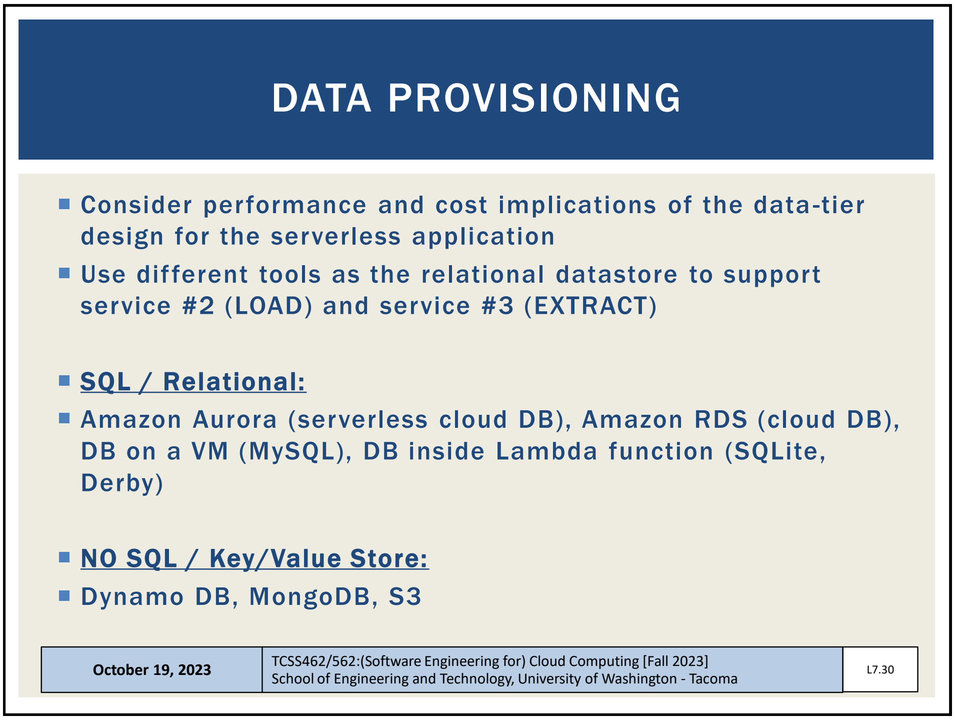
- FaaS platforms support hosting code in multiple languages
- AWS Lambda- common: Java, Node.js, Python
  - Plus others: Go, PowerShell, C#, and Ruby
- Also Runtime API (“BASH”) which allows deployment of binary executables from any programming language
- August 2020 – Our group’s paper:
- https://tinyurl.com/y46eq6np
- If wanting to perform a language study either:
  - Implement in C#, Ruby, or multiple versions of Java, Node.js, Python
  - OR implement different app than TLQ (ETL) data processing pipeline



FAAS PLATFORMS

- Many commercial and open source FaaS platforms exist
- TCSS562 projects can choose to compare performance and cost implications of alternate platforms.

- Supported by SAAF:
- AWS Lambda
- Google Cloud Functions
- Azure Functions
- IBM Cloud Functions
- Apache OpenWhisk *(open source, deploy your own FaaS)*
- Open FaaS *(open source, deploy your own FaaS)*

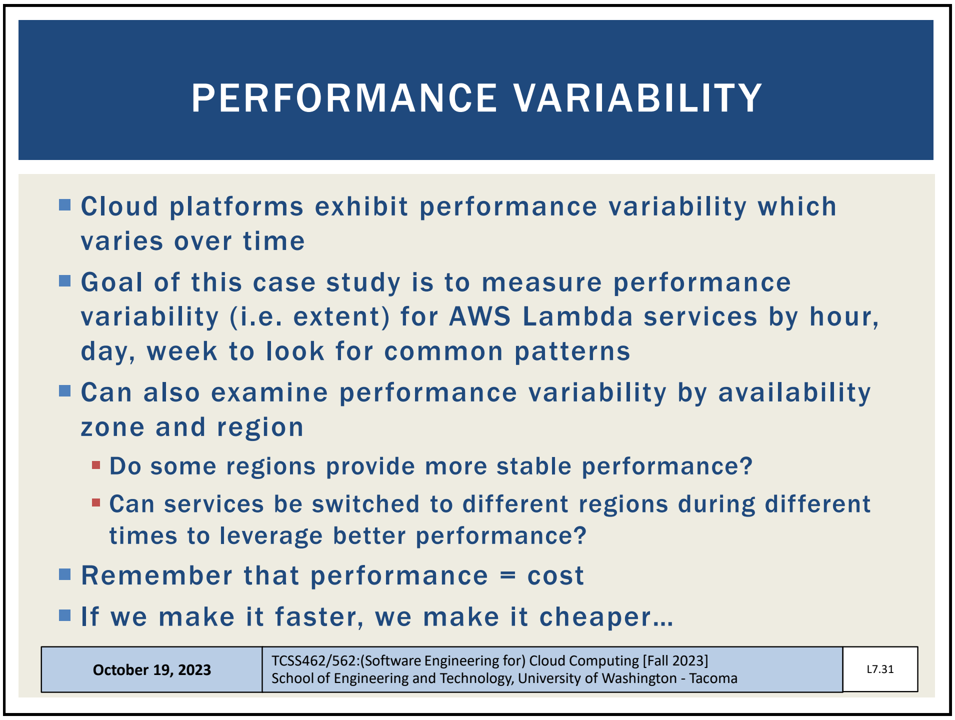


DATA PROVISIONING

- Consider performance and cost implications of the data-tier design for the serverless application
- Use different tools as the relational datastore to support service #2 (LOAD) and service #3 (EXTRACT)

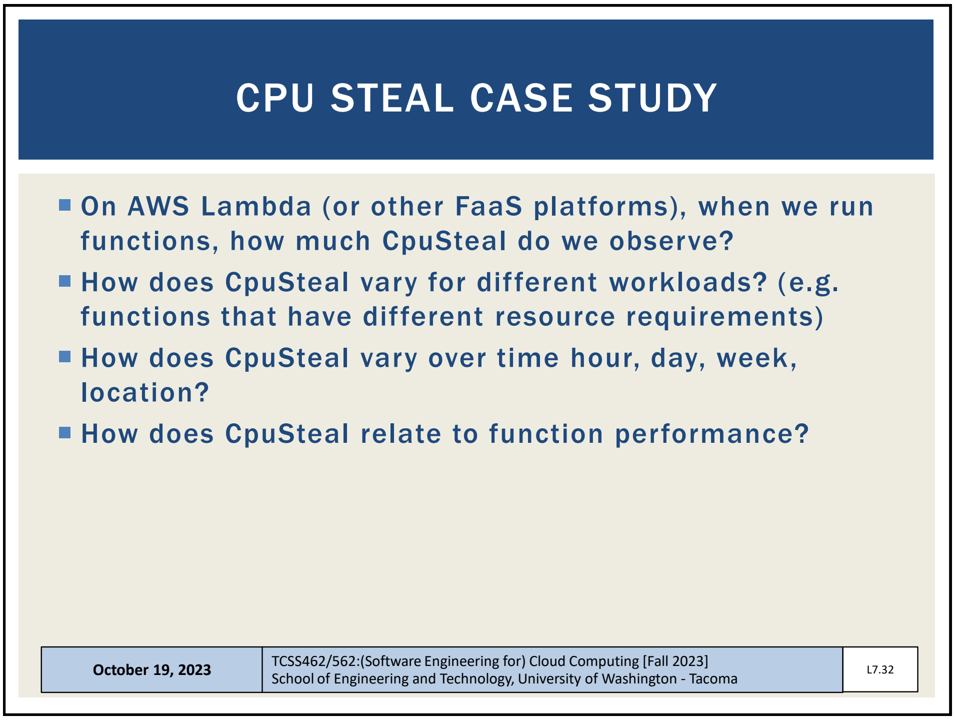
- **SQL / Relational:**
- Amazon Aurora (serverless cloud DB), Amazon RDS (cloud DB), DB on a VM (MySQL), DB inside Lambda function (SQLite, Derby)

- **NO SQL / Key/Value Store:**
- Dynamo DB, MongoDB, S3



PERFORMANCE VARIABILITY

- Cloud platforms exhibit performance variability which varies over time
- Goal of this case study is to measure performance variability (i.e. extent) for AWS Lambda services by hour, day, week to look for common patterns
- Can also examine performance variability by availability zone and region
  - Do some regions provide more stable performance?
  - Can services be switched to different regions during different times to leverage better performance?
- Remember that performance = cost
- If we make it faster, we make it cheaper…

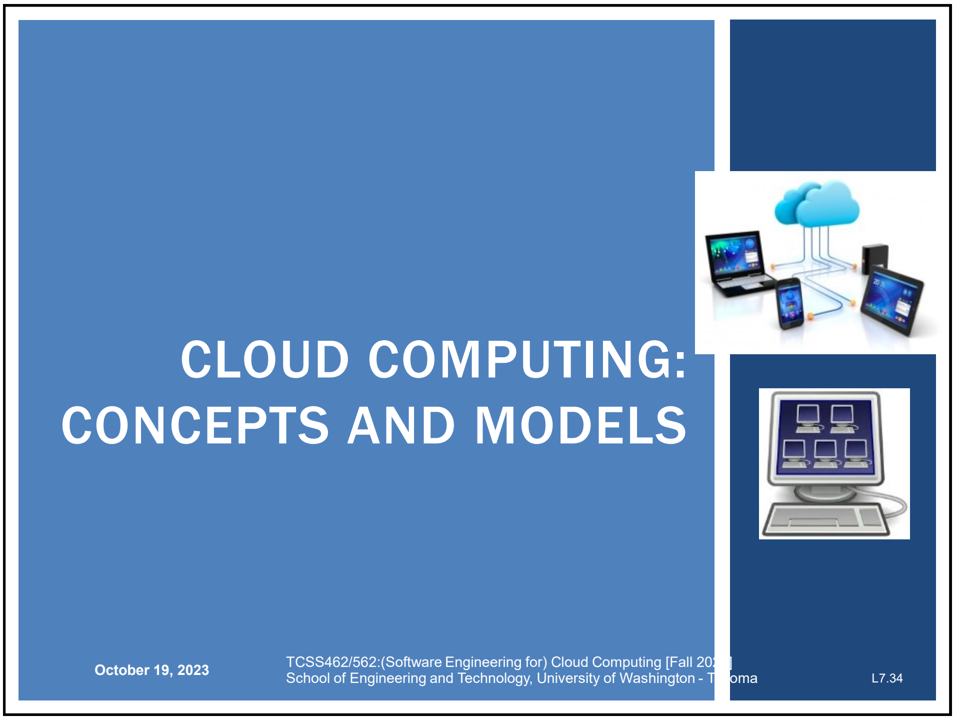


CPU STEAL CASE STUDY

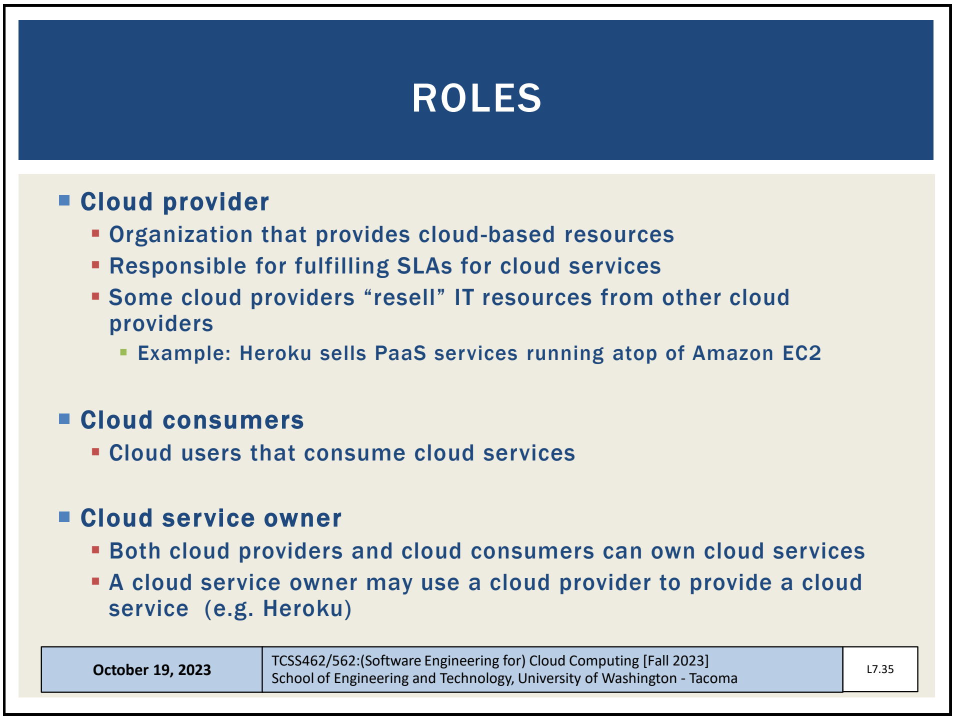
- On AWS Lambda (or other FaaS platforms), when we run functions, how much CpuSteal do we observe?
- How does CpuSteal vary for different workloads? (e.g. functions that have different resource requirements)
- How does CpuSteal vary over time hour, day, week, location?
- How does CpuSteal relate to function performance?

OBJECTIVES – 10/19

- Questions from 10/17
- 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

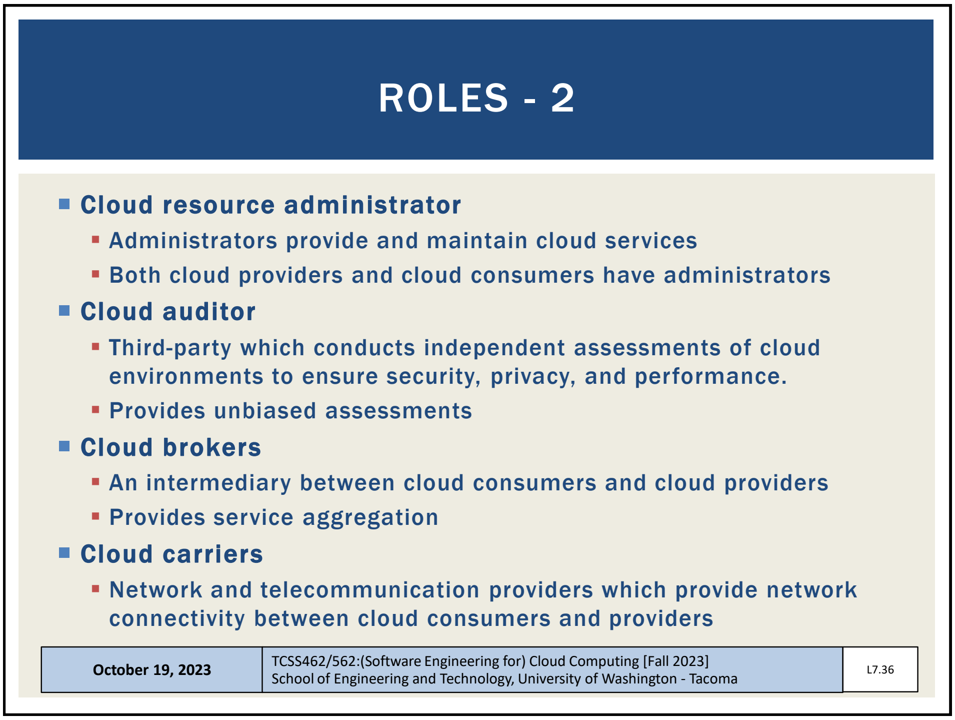


CLOUD COMPUTING: CONCEPTS AND MODELS



ROLES

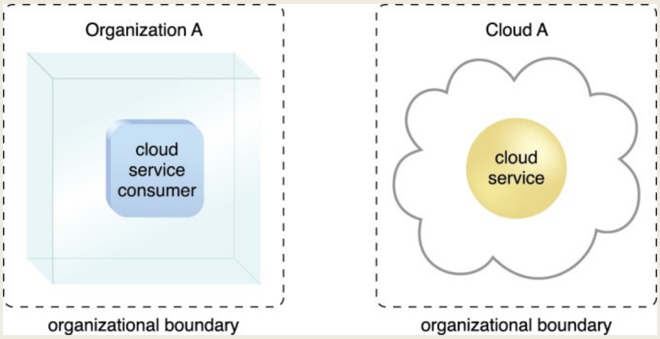
- **Cloud provider**
  - Organization that provides cloud-based resources
  - Responsible for fulfilling SLAs for cloud services
  - Some cloud providers “resell” IT resources from other cloud providers
    - Example: Heroku sells PaaS services running atop of Amazon EC2
- **Cloud consumers**
  - Cloud 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)



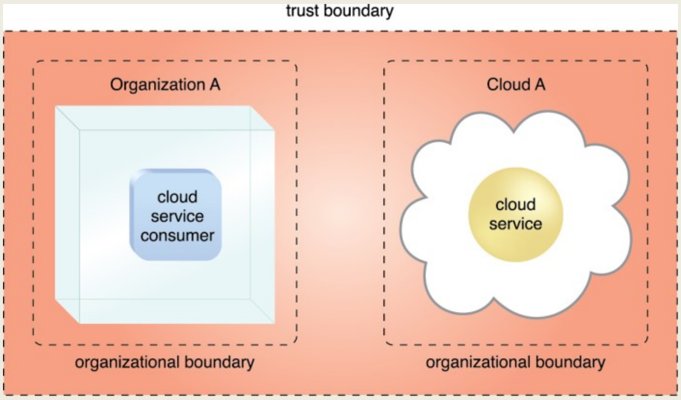
ROLES - 2

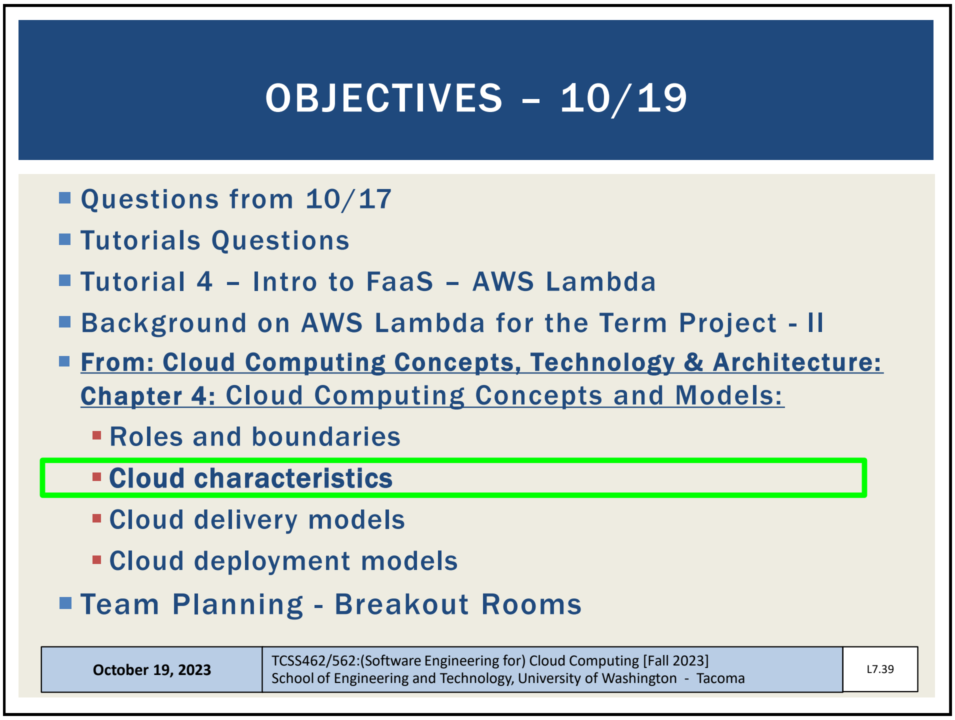
- **Cloud resource administrator**
  - Administrators provide and maintain cloud services
  - Both cloud providers and cloud consumers have administrators
- **Cloud auditor**
  - Third-party which conducts independent assessments of cloud environments to ensure security, privacy, and performance.
  - 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



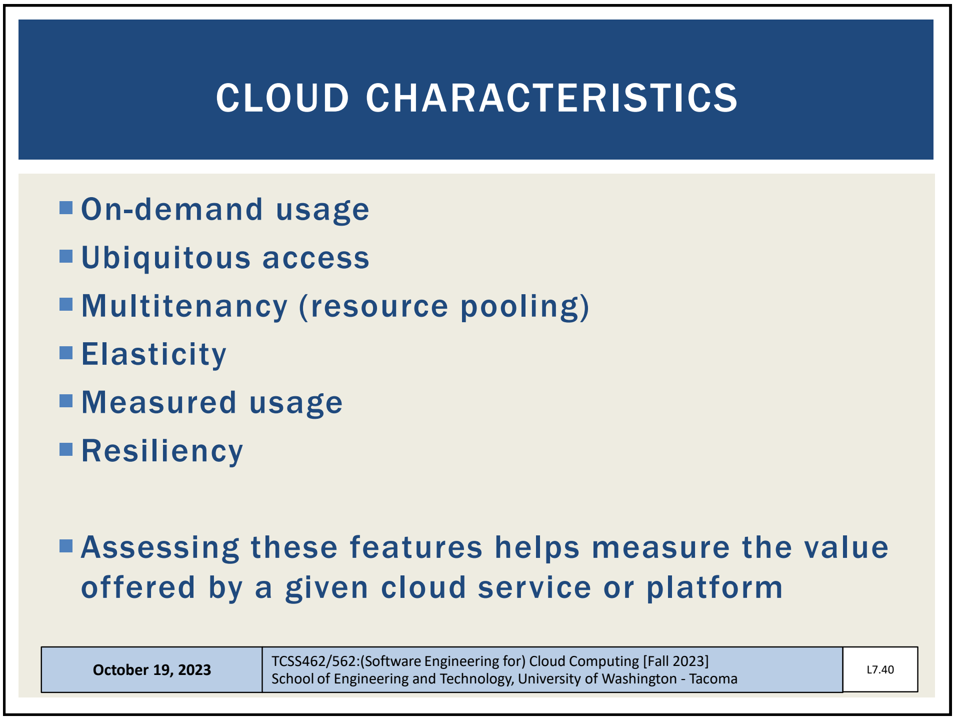
TRUST BOUNDARY





OBJECTIVES – 10/19

- Questions from 10/17
- 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



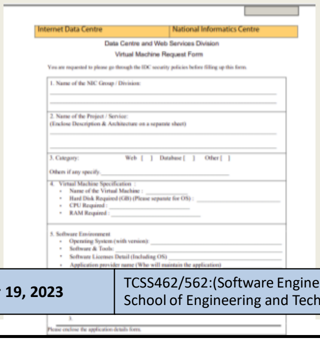
CLOUD CHARACTERISTICS

- 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 [ ]

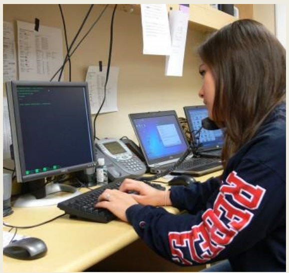
Others if any specify

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)

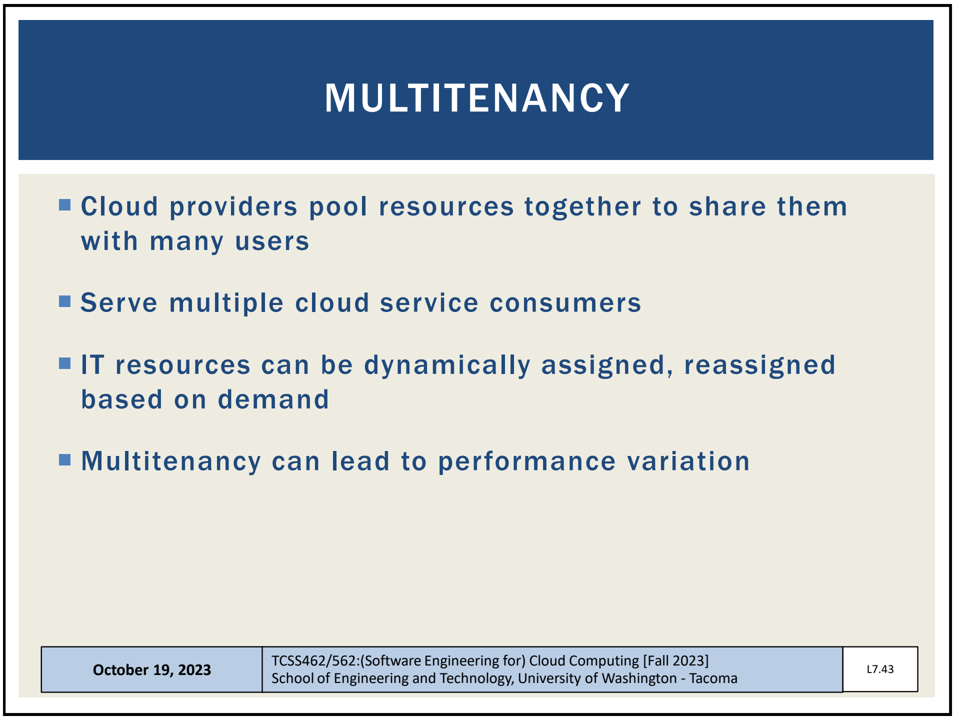
3.

Please enclose the application details form.



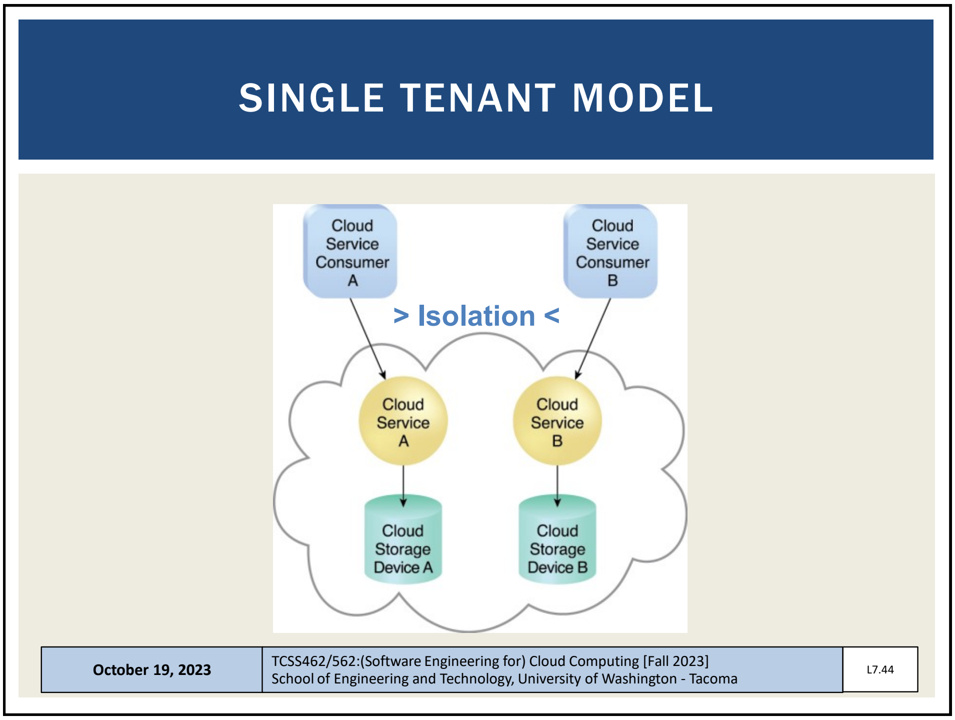
UBIQUITOUS ACCESS

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

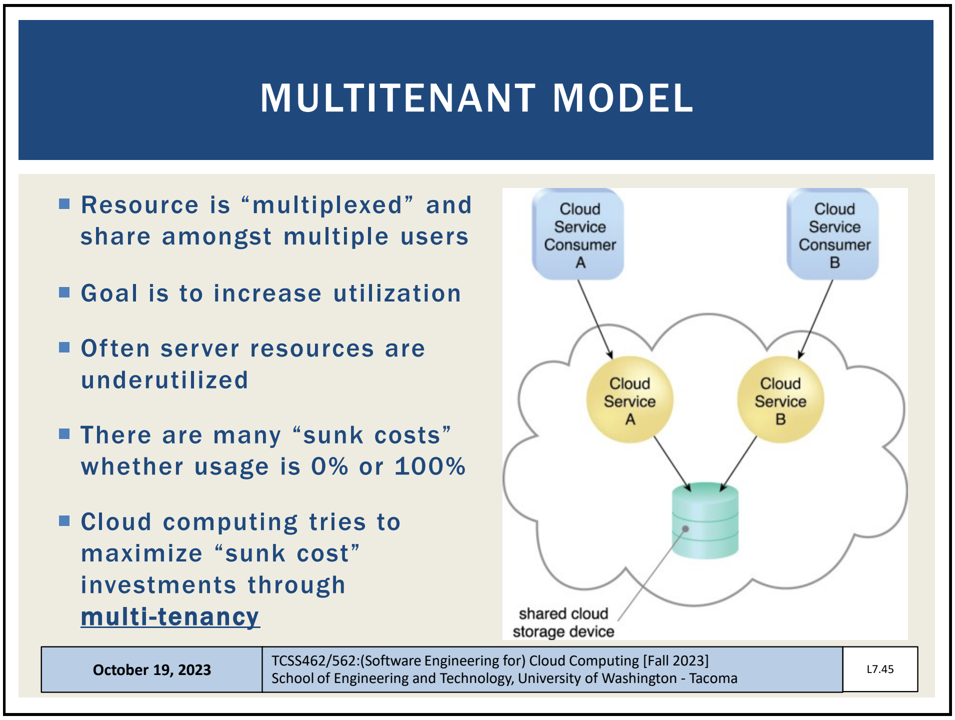


MULTITENANCY

- 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



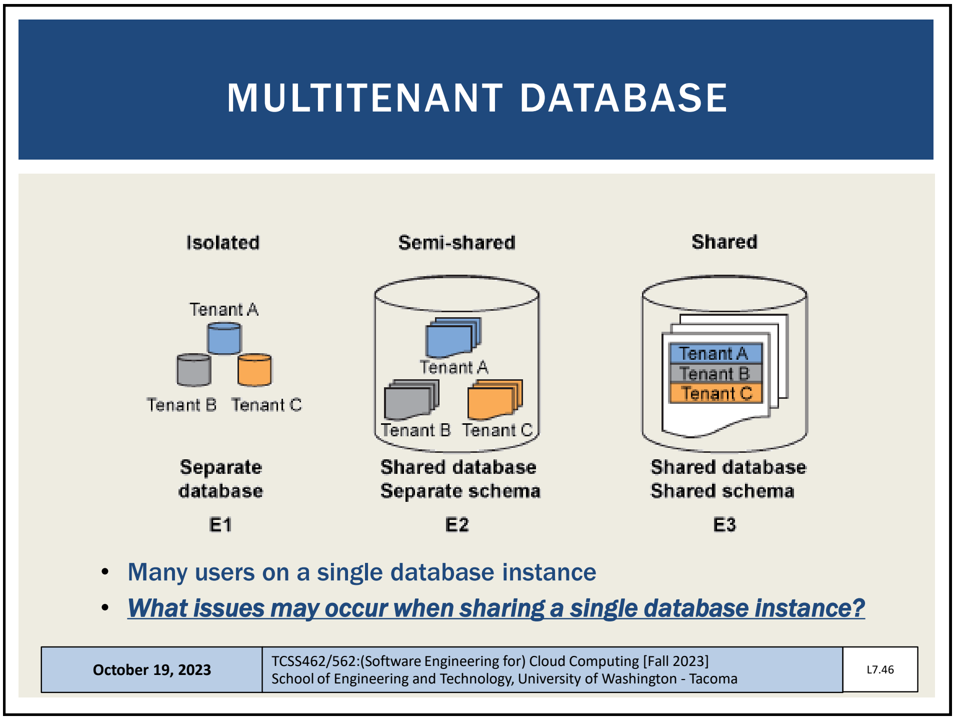
SINGLE TENANT MODEL



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 <u>**multi-tenancy**</u>

Cloud Service Consumer A
Cloud Service Consumer B
Cloud Service A
Cloud Service B
shared cloud storage device



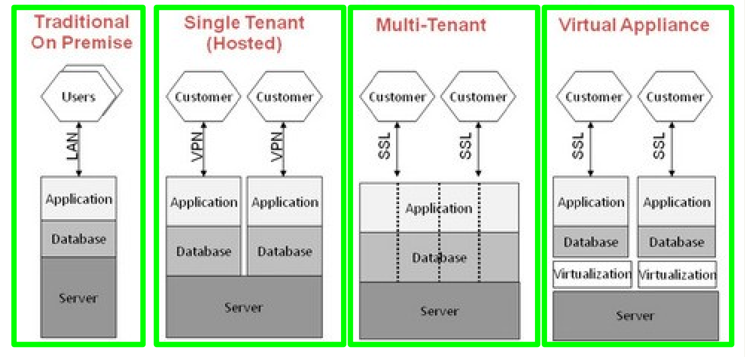
MULTITENANT DATABASE

Isolated
Semi-shared
Shared
Tenant A
Tenant B
Tenant C
Tenant A
Tenant B
Tenant C
Tenant A
Tenant B
Tenant C
Separate database
Shared database Separate schema
Shared database Shared schema
E1
E2
E3

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

MULTITENANCY OF RESOURCES

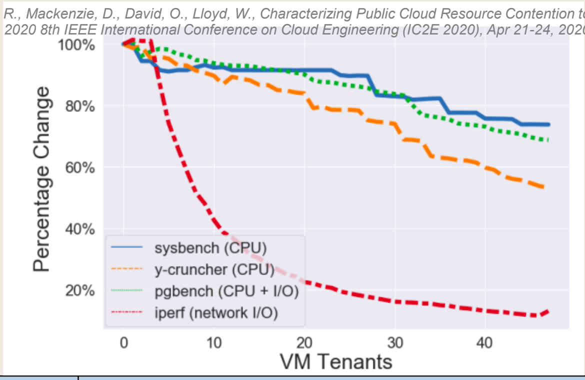
- Where is the multitenancy?
  - >> What is shared? What is isolated?



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 !!

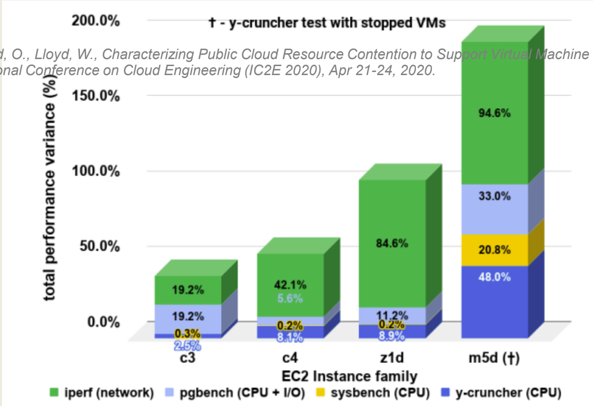
RESOURCE CONTENTION FROM MUTLI-TENANCY - 2

- 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) →

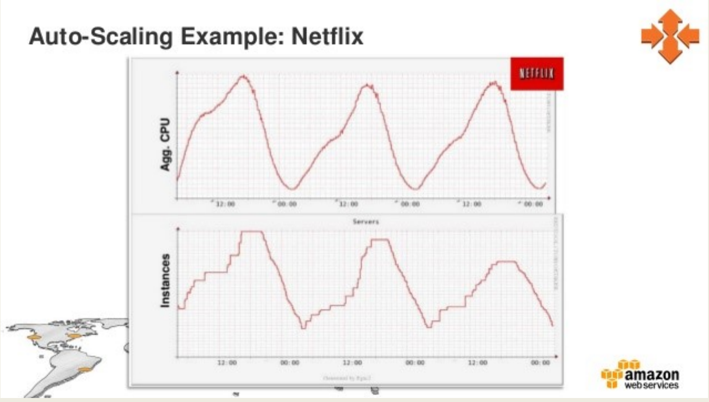


ELASTICITY

- 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

PREDICTABLE DEMAND

- AWS EC2 Scaling Example:



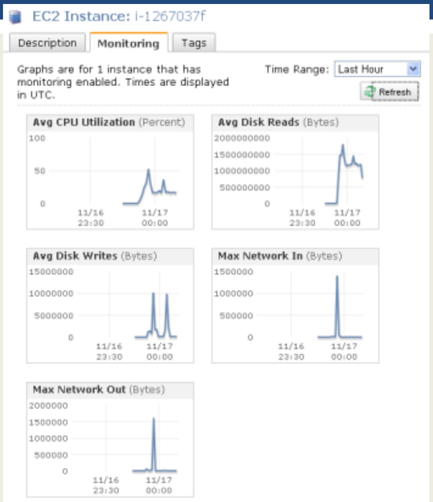
From: Kejariwal, A., 2013, March. Techniques for optimizing cloud footprint. In 2013 IEEE Int. Conf. on Cloud Engineering (IC2E), pp. 258-268.

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)

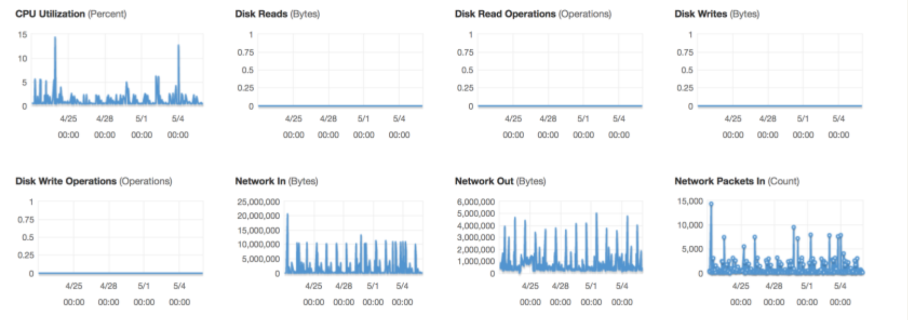
- Not all measurements are for billing
- Some measurements can support auto-scaling
- For example CPU utilization

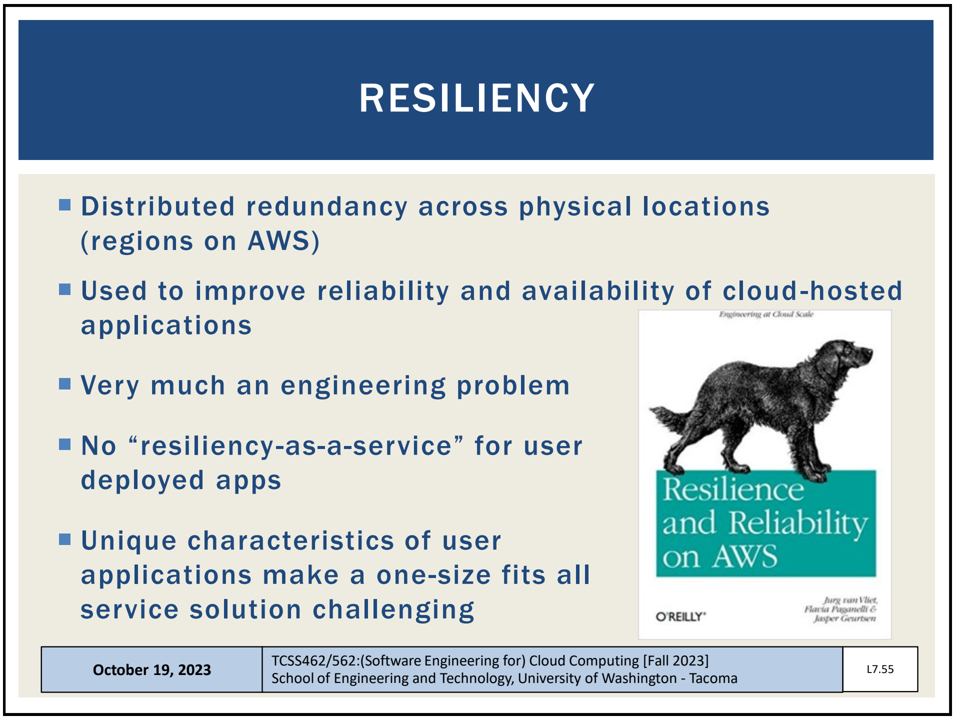
EC2 CLOUDWATCH METRICS



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

EC2 CLOUDWATCH METRICS





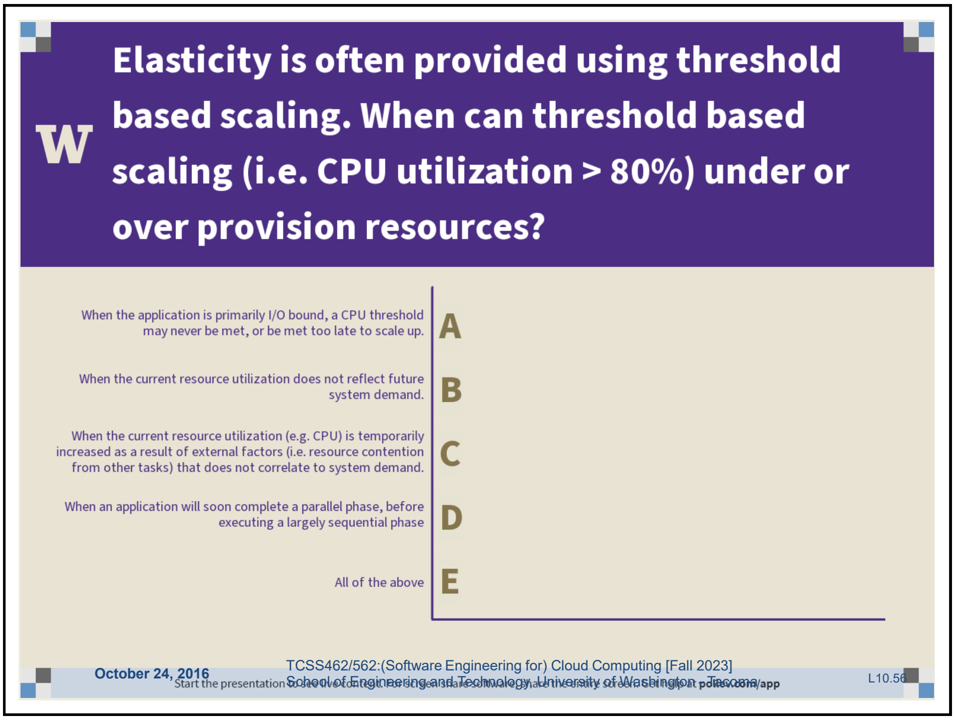
RESILIENCY

- 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

Engineering at Cloud Scale
Resilience and Reliability on AWS
Jurg van Vliet, Flavia Paganelli & Jasper Geurtsen
O'REILLY

October 19, 2023 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma | L7.55

55



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

When the application is primarily I/O bound, a CPU threshold may never be met, or be met too late to scale up. **A**

When the current resource utilization does not reflect future system demand. **B**

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. **C**

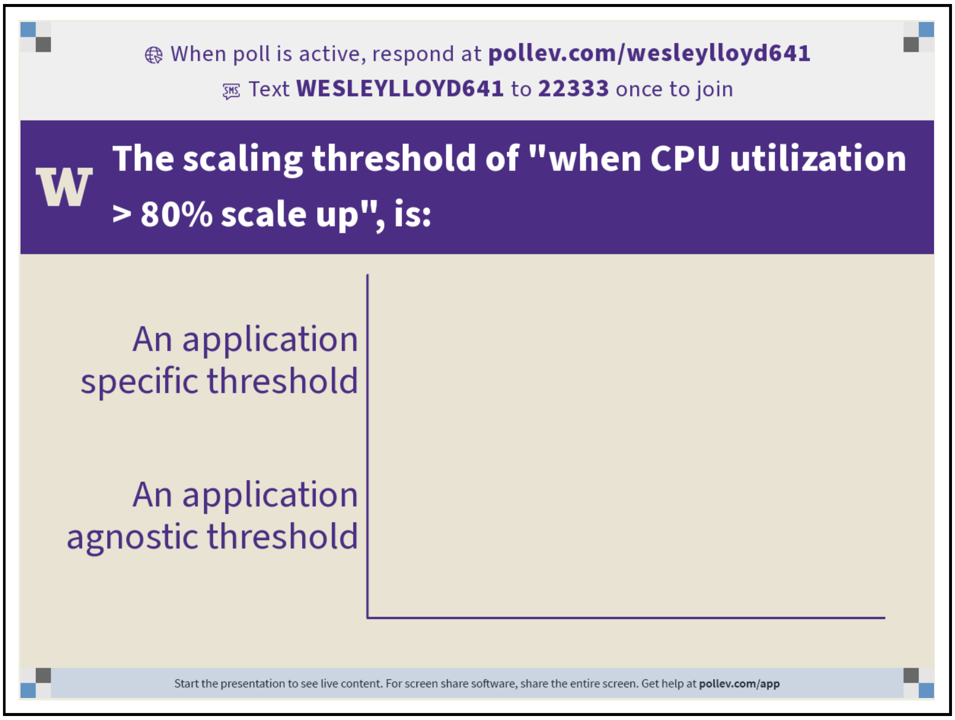
When an application will soon complete a parallel phase, before executing a largely sequential phase **D**

All of the above **E**

October 24, 2016 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma | L10.56

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

56



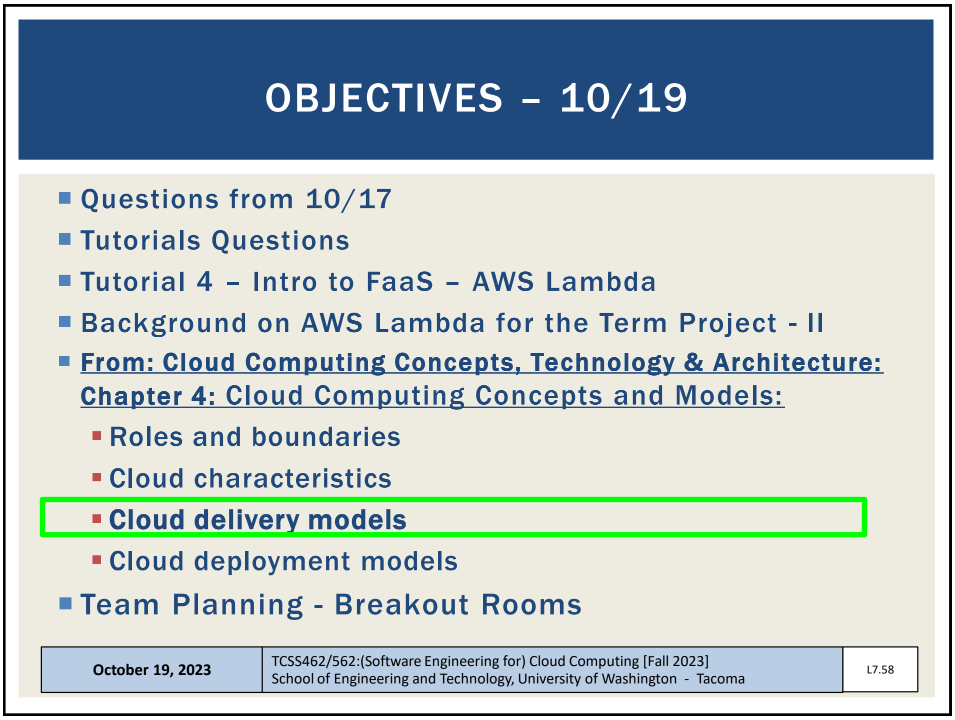
When poll is active, respond at **pollev.com/wesleylloyd641**
Text **WESLEYLLOYD641** to **22333** once to join

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

An application specific threshold
An application agnostic threshold

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

57

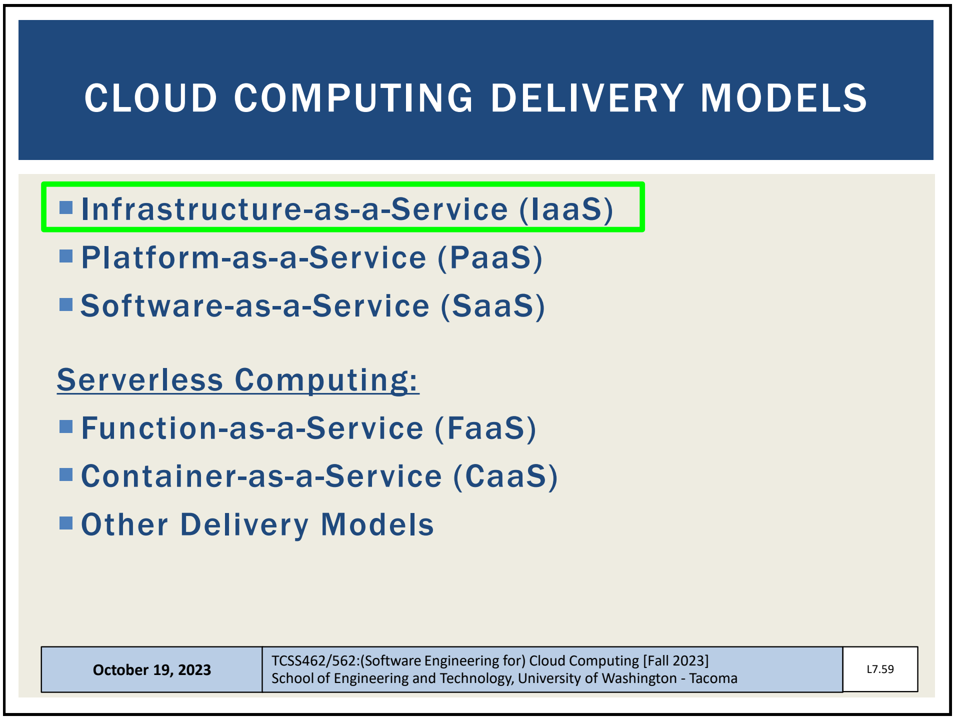


OBJECTIVES – 10/19

- Questions from 10/17
- 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 19, 2023	TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma	L7.58

58



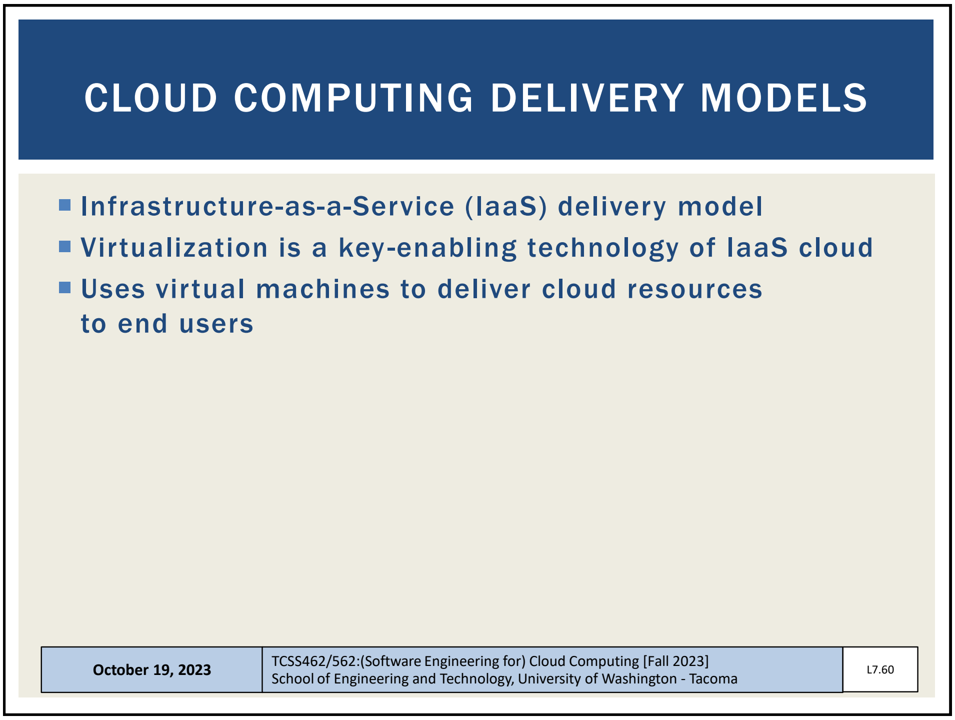
CLOUD COMPUTING DELIVERY MODELS

- 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

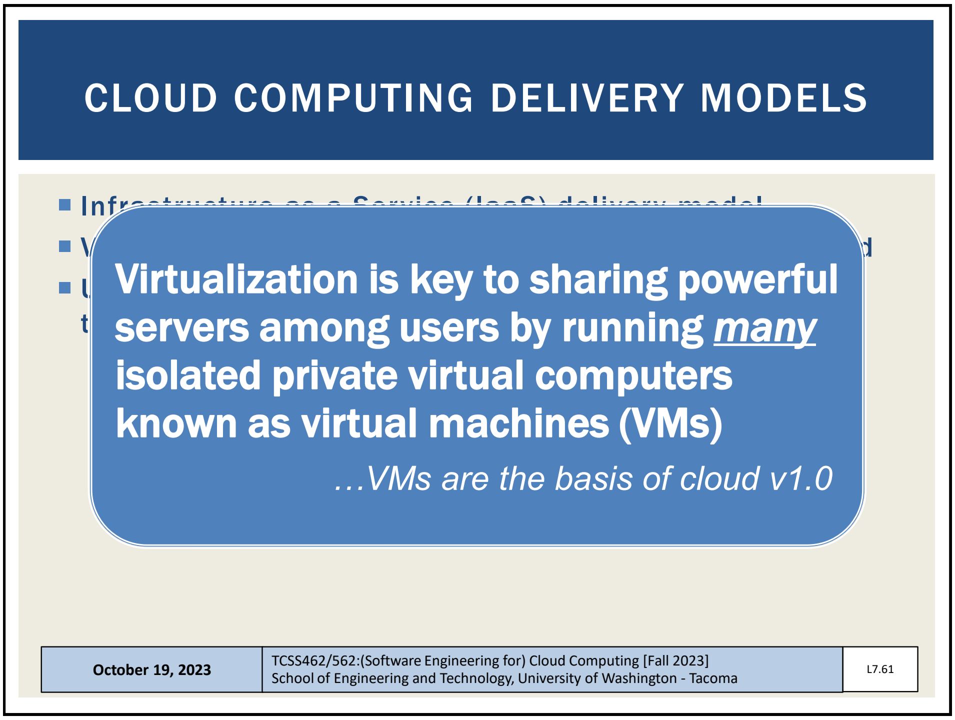
59



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

60



CLOUD COMPUTING DELIVERY MODELS

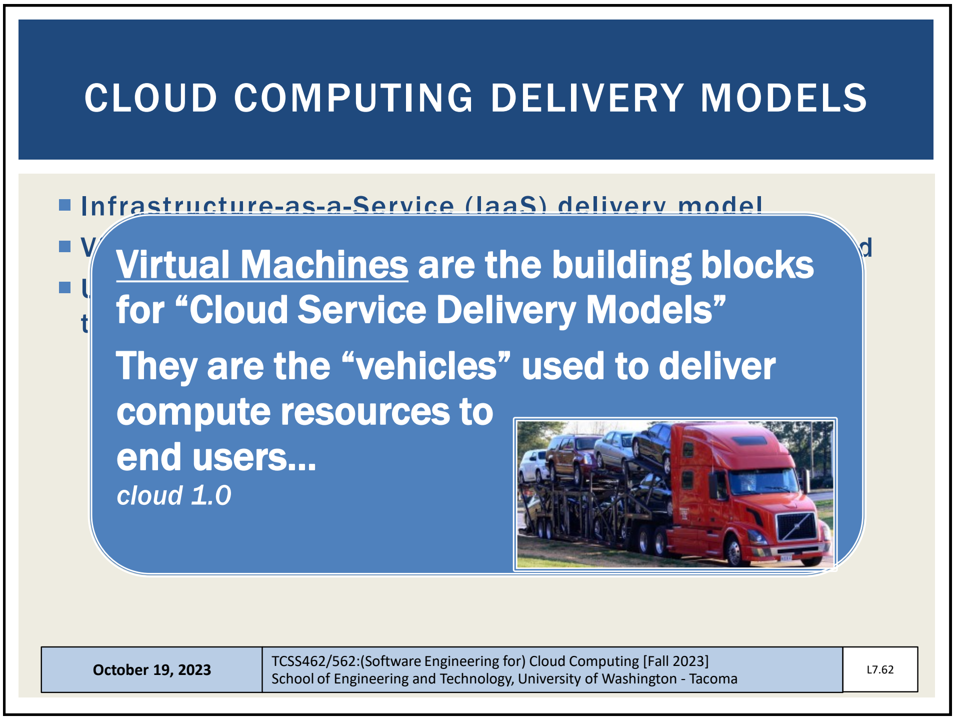
- Infrastructure-as-a-Service (IaaS) delivery model
- V d
- U
  t

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 19, 2023 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023]
School of Engineering and Technology, University of Washington - Tacoma | L7.61

61



CLOUD COMPUTING DELIVERY MODELS

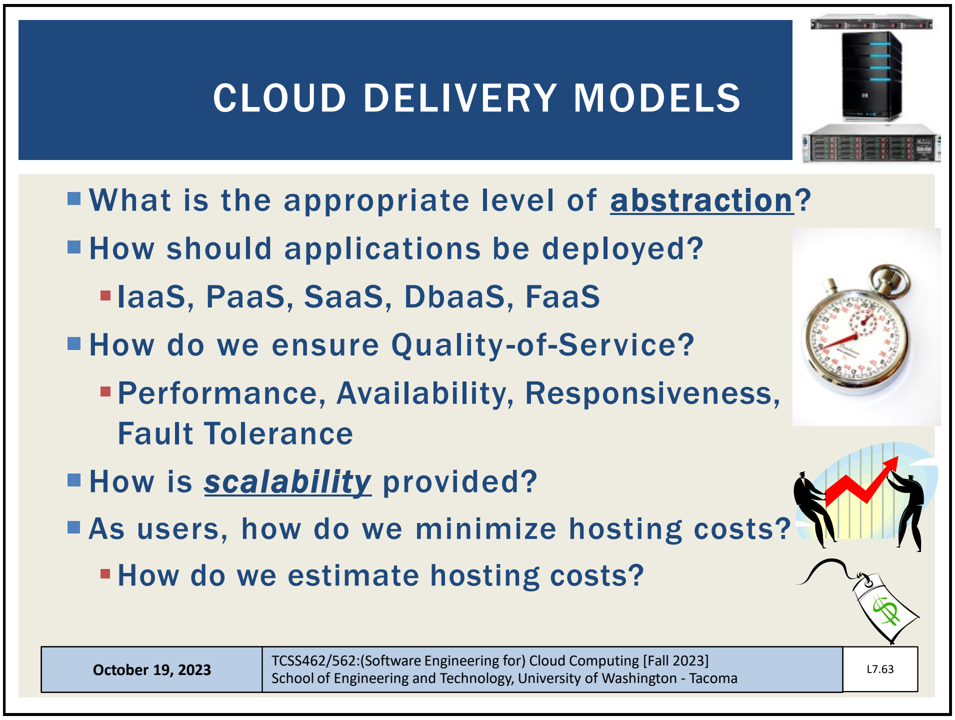
- Infrastructure-as-a-Service (IaaS) delivery model
- V d
- U
  t

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 19, 2023 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023]
School of Engineering and Technology, University of Washington - Tacoma | L7.62

62

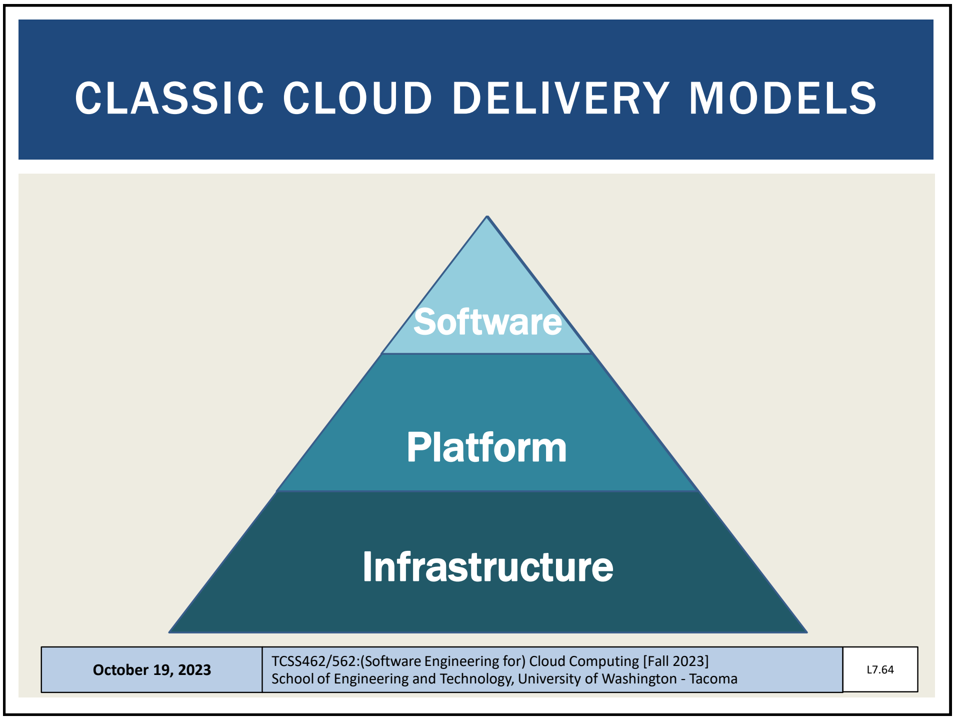


CLOUD DELIVERY MODELS

- 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?
- As users, how do we minimize hosting costs?
  - How do we estimate hosting costs?

October 19, 2023 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma | L7.63

63



CLASSIC CLOUD DELIVERY MODELS

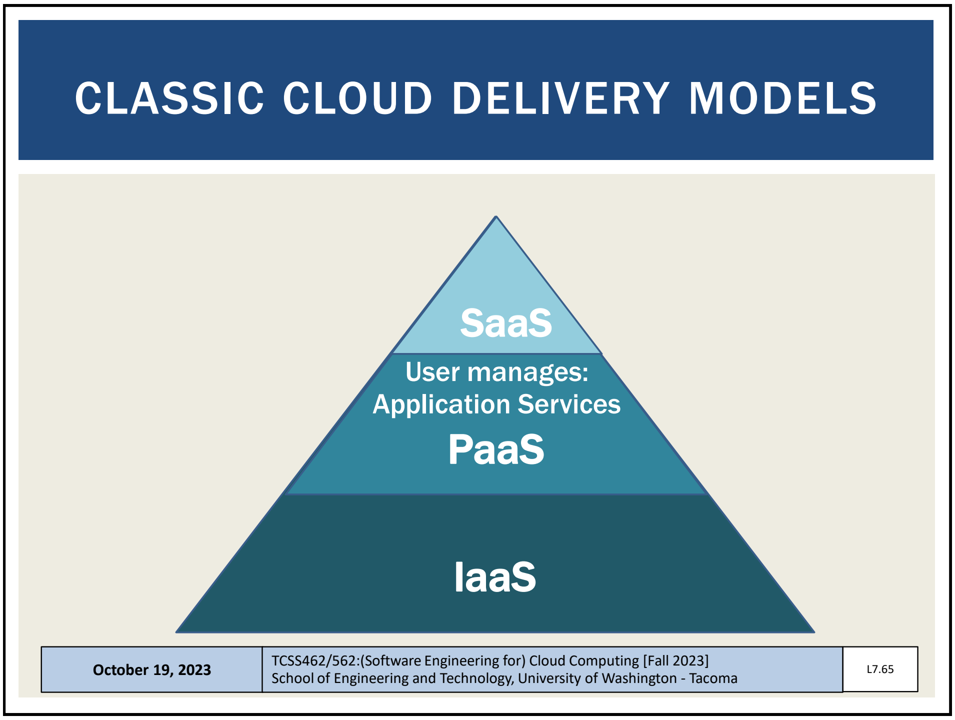
Software

Platform

Infrastructure

October 19, 2023 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma | L7.64

64



CLASSIC CLOUD DELIVERY MODELS

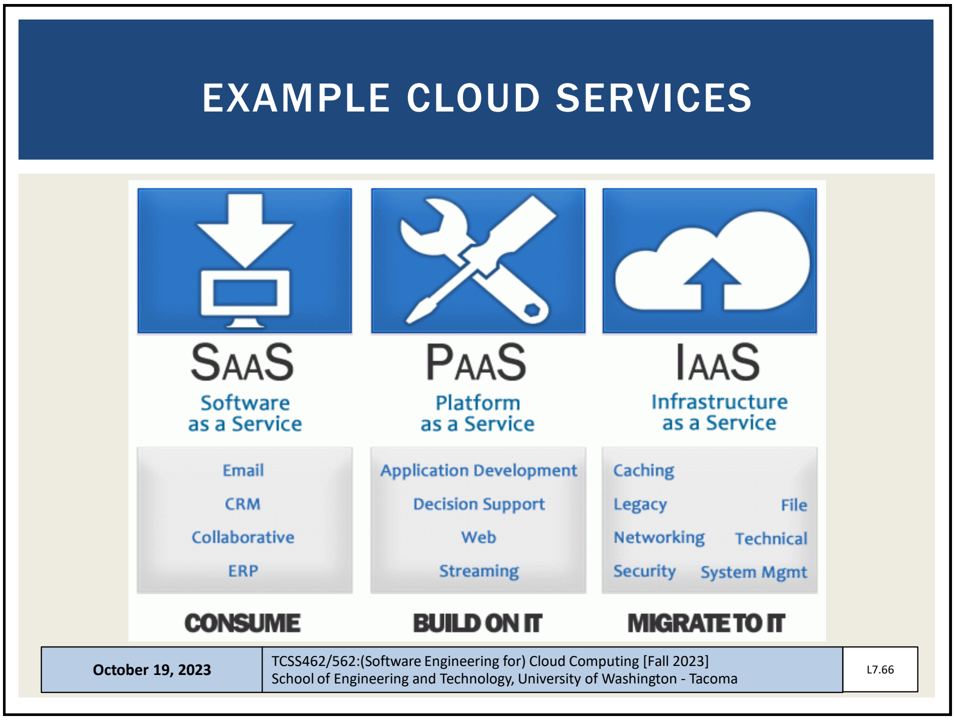
SaaS

User manages:
Application Services

PaaS

IaaS

65



EXAMPLE CLOUD SERVICES

SAAS
Software
as a Service

Email
CRM
Collaborative
ERP

CONSUME

PAAS
Platform
as a Service

Application Development
Decision Support
Web
Streaming

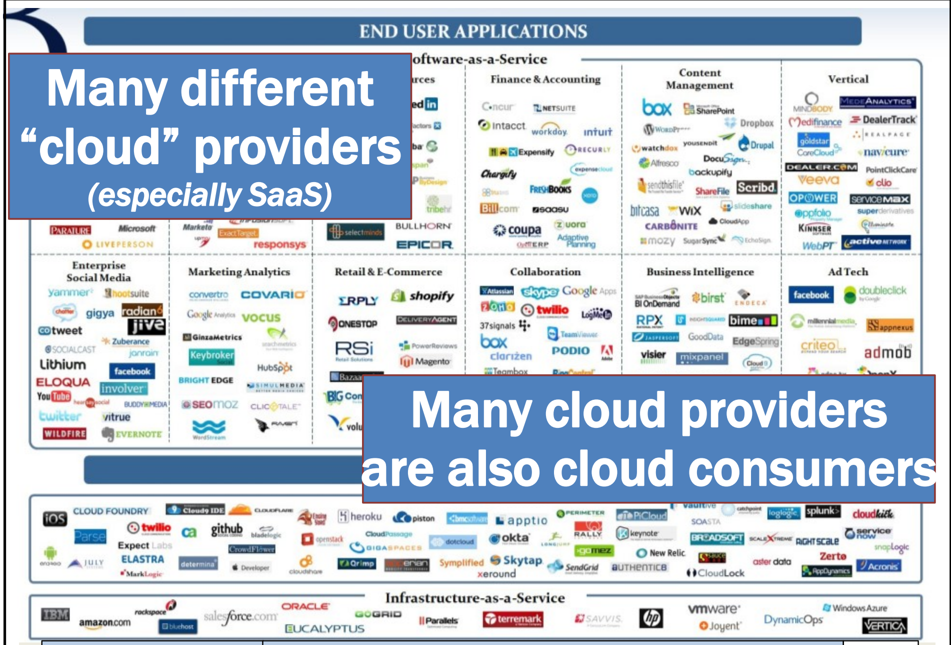
BUILD ON IT

IAAS
Infrastructure
as a Service

Caching
Legacy
File
Networking
Technical
Security
System Mgmt

MIGRATE TO IT

66



Many different "cloud" providers

(especially SaaS)

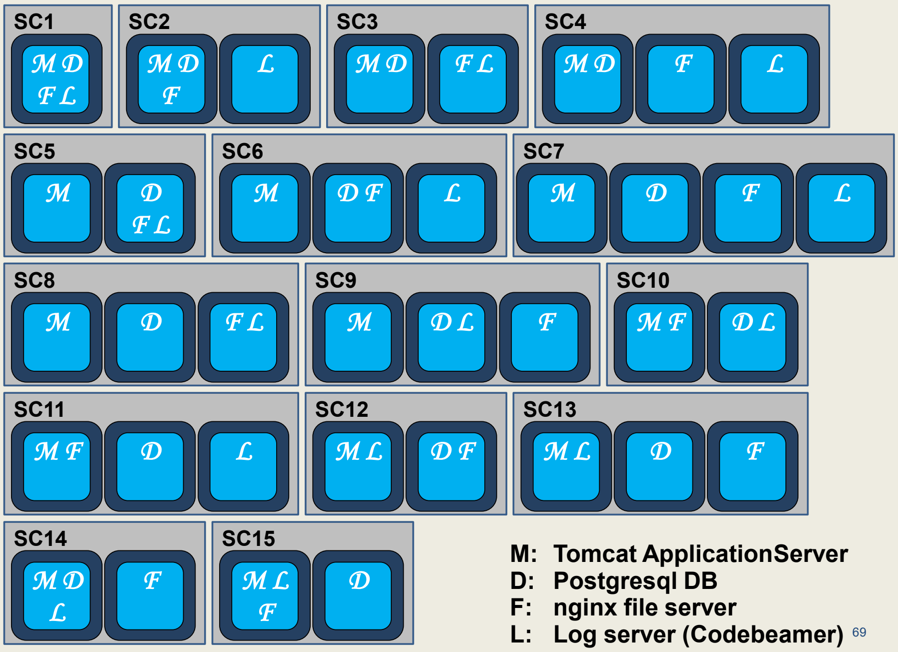
Many cloud providers are also cloud consumers

67

INFRASTRUCTURE-AS-A-SERVICE

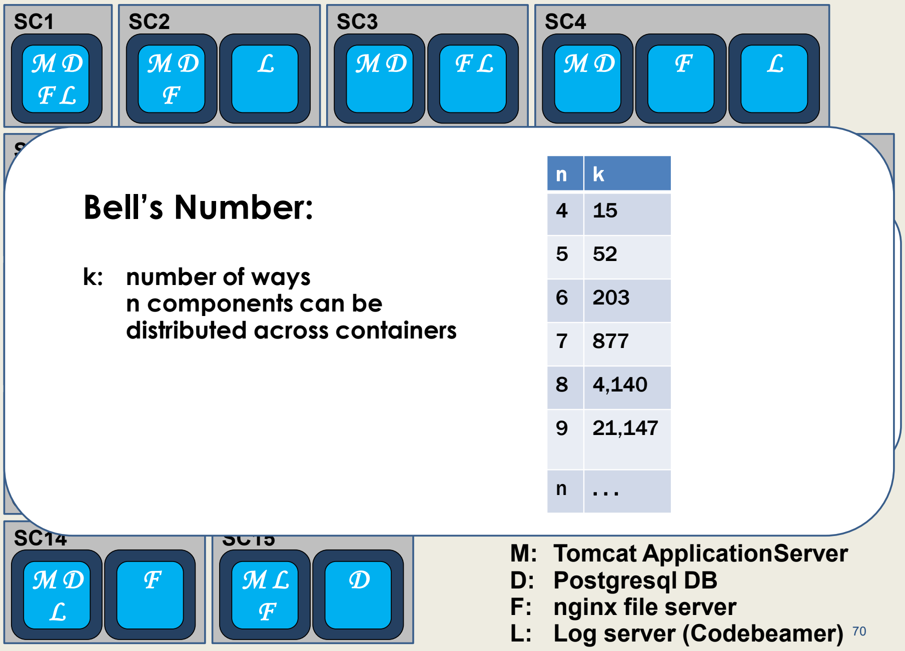
- Compute resources, on demand, as-a-service
  - Generally raw "IT" resources
  - Hardware, network, containers, operating systems
- Typically provided through virtualization
- Generally, not-preconfigured
- Administrative burden is owned by cloud consumer
- Best when high-level control over environment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

68



M: Tomcat ApplicationServer
D: Postgresql DB
F: nginx file server
L: Log server (Codebeamer) 69

69



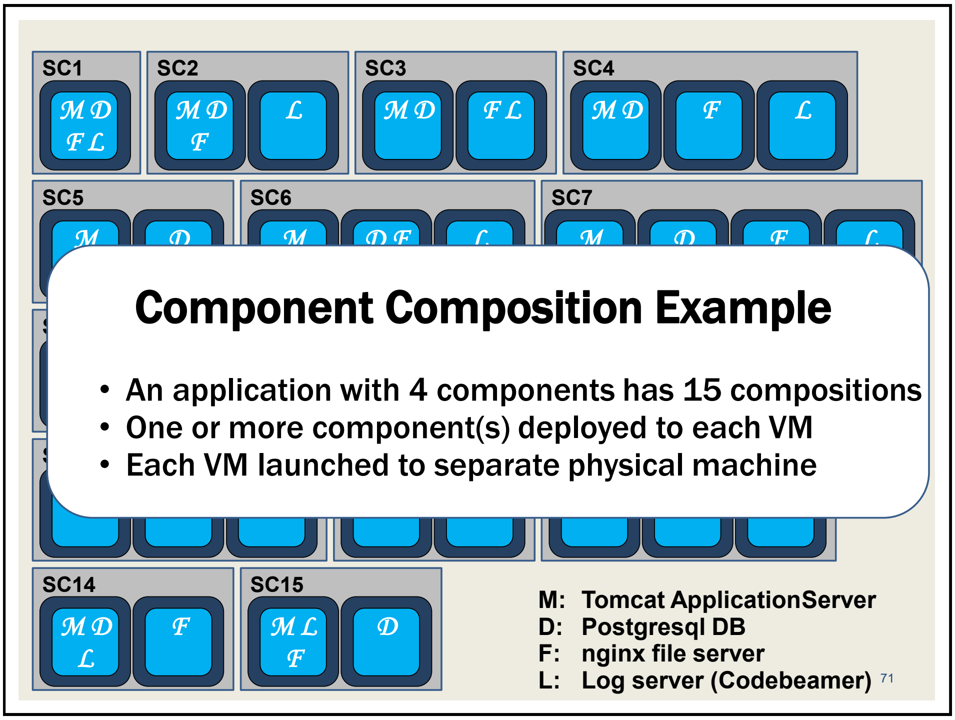
Bell's Number:

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	. . .

M: Tomcat ApplicationServer
D: Postgresql DB
F: nginx file server
L: Log server (Codebeamer) 70

70



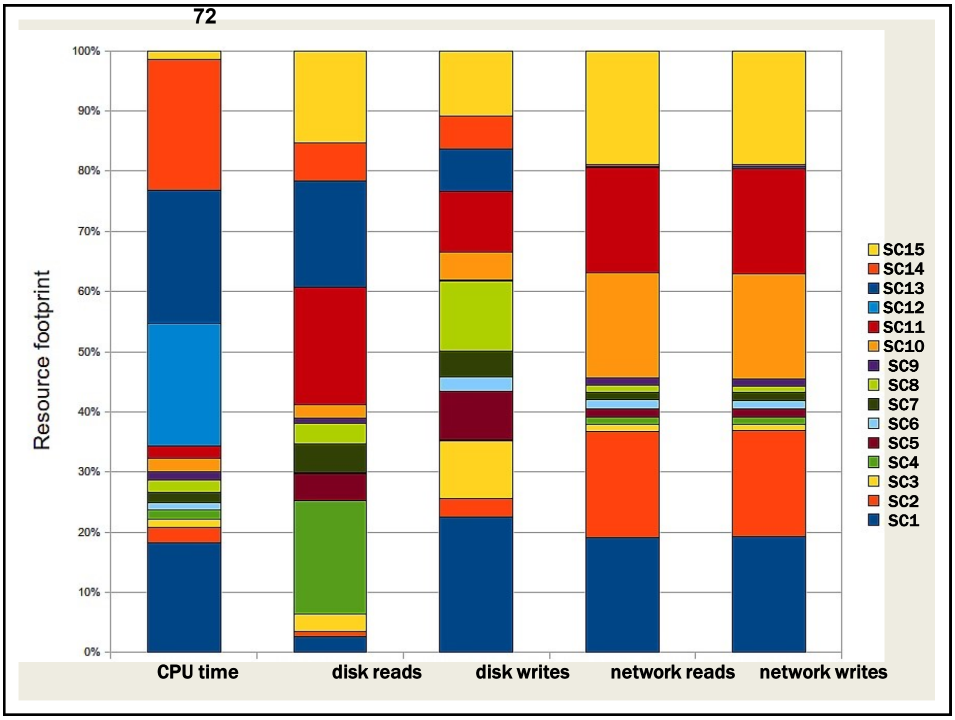
Component Composition Example

- An application with 4 components has 15 compositions
- One or more component(s) deployed to each VM
- Each VM launched to separate physical machine

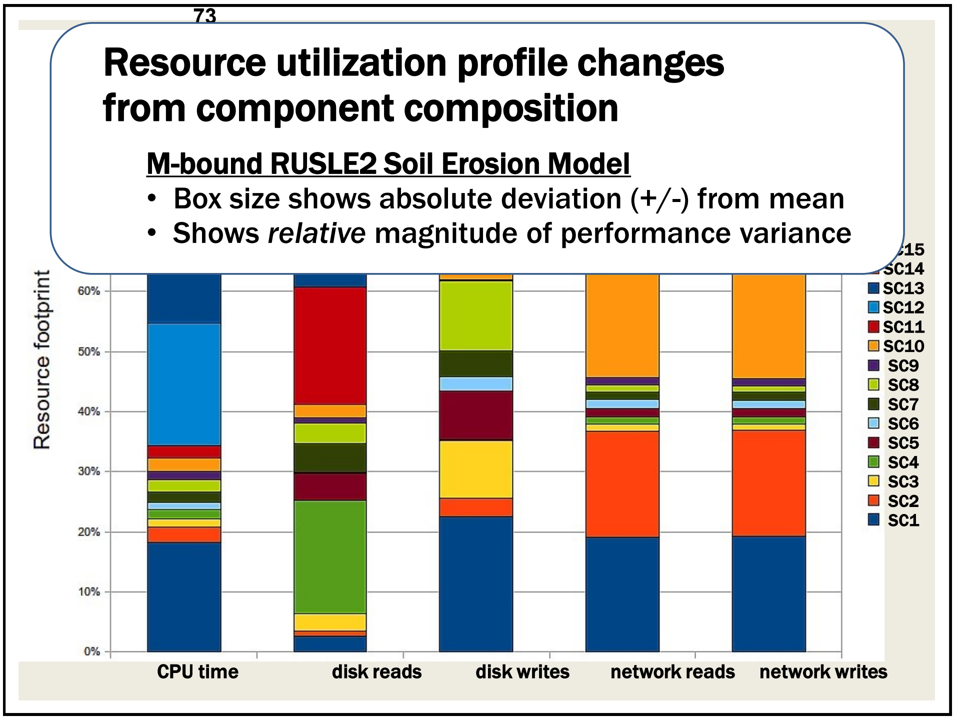
M: Tomcat ApplicationServer
D: Postgresql DB
F: nginx file server
L: Log server (Codebeamer) 71

71

72



72

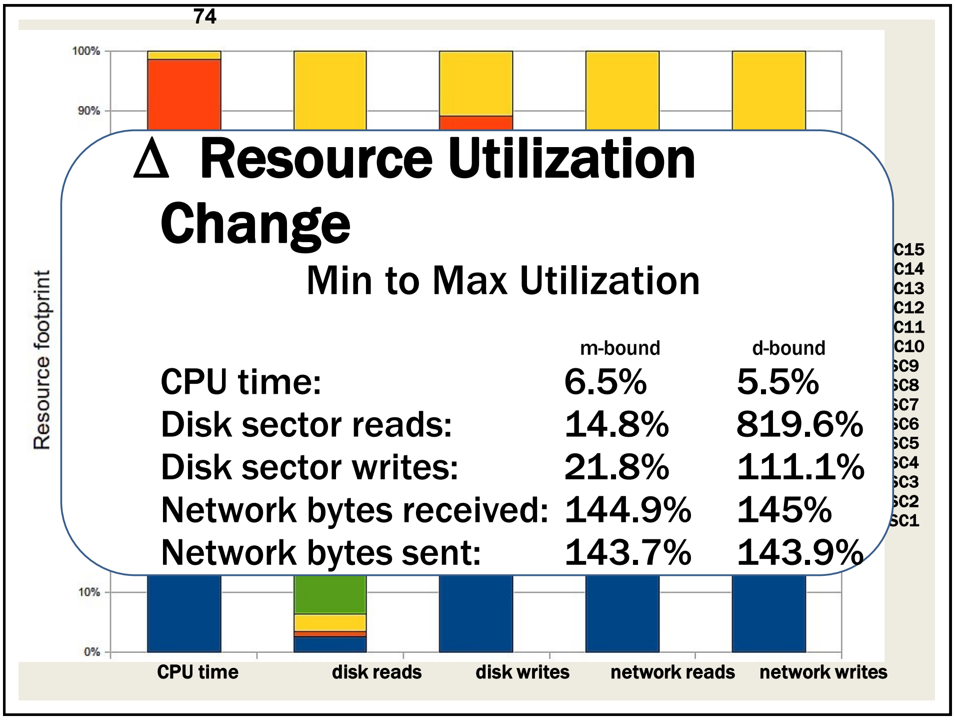


Resource utilization profile changes from component composition

M-bound RUSLE2 Soil Erosion Model

- Box size shows absolute deviation (+/-) from mean
- Shows *relative* magnitude of performance variance

73

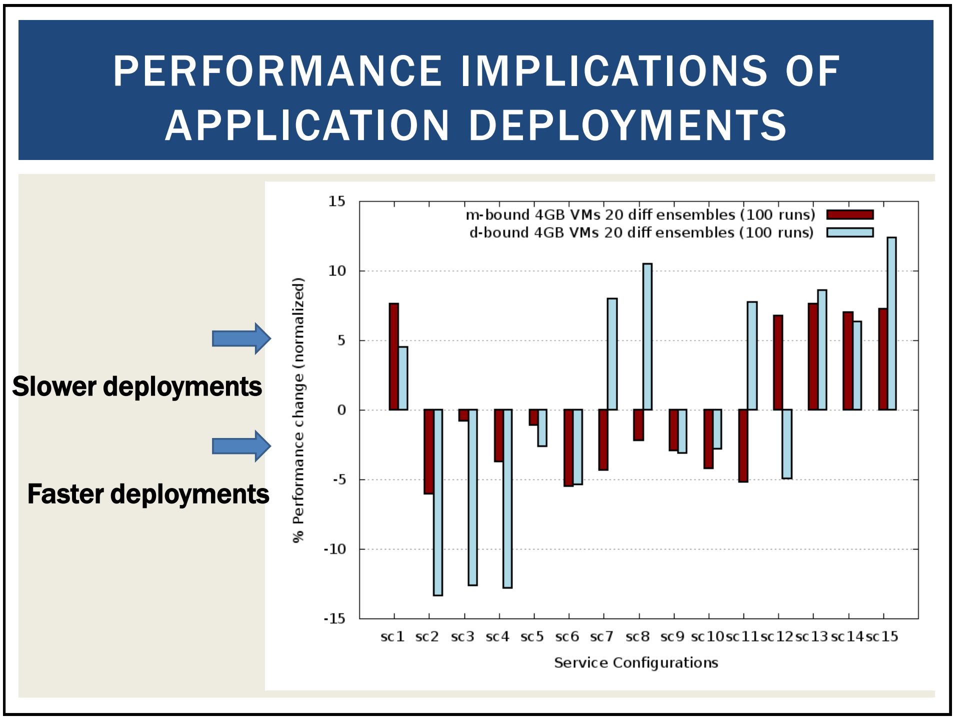


Δ Resource Utilization Change

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%

74

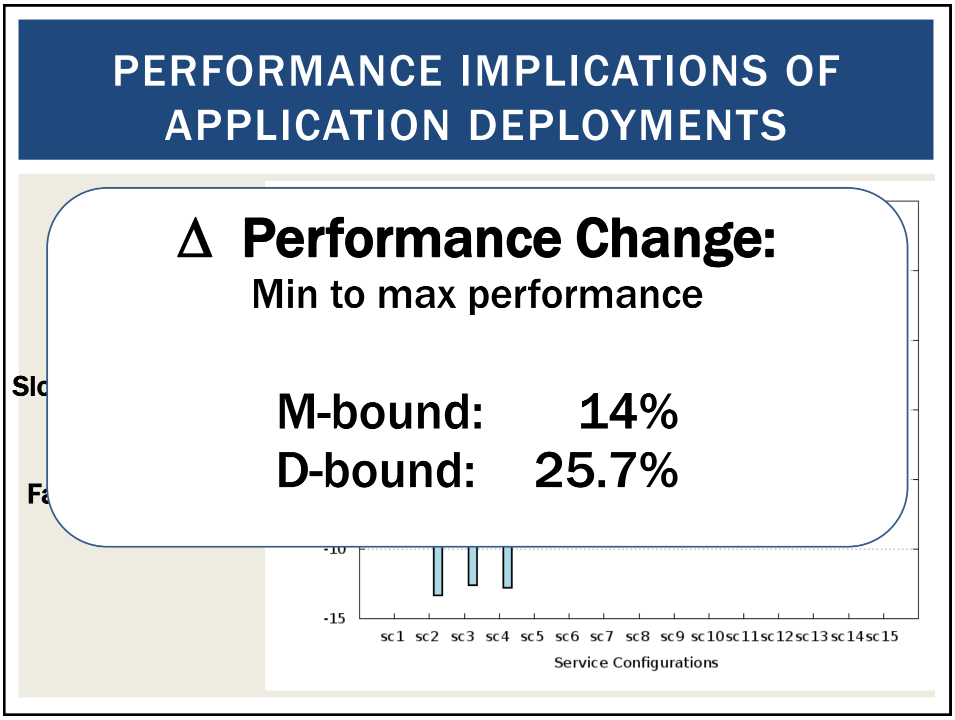


PERFORMANCE IMPLICATIONS OF APPLICATION DEPLOYMENTS

Slower deployments

Faster deployments

75



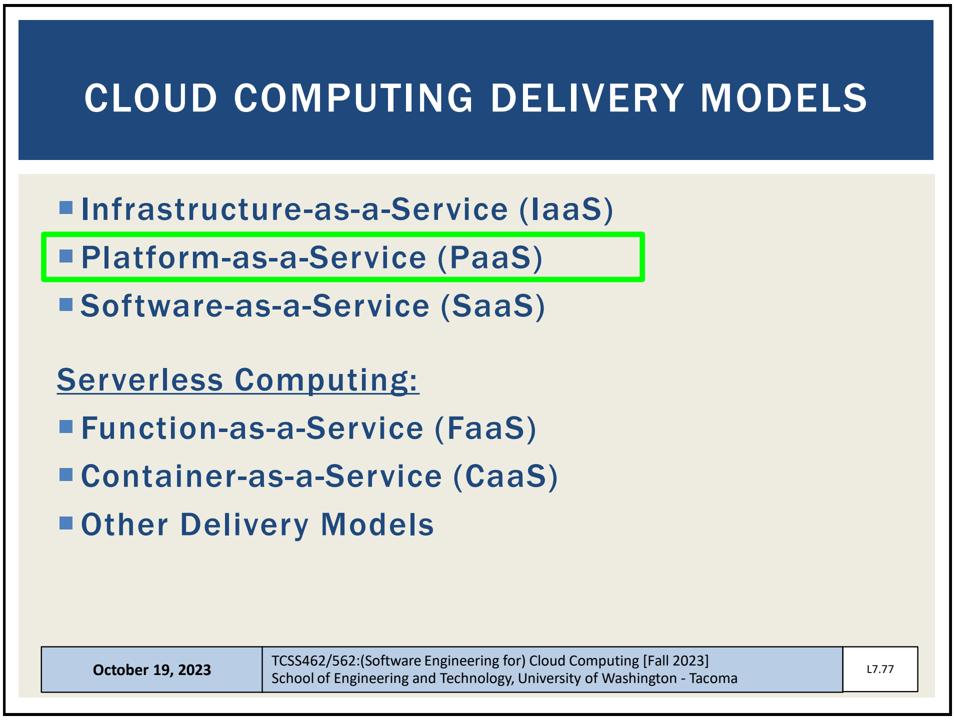
PERFORMANCE IMPLICATIONS OF APPLICATION DEPLOYMENTS

∆ Performance Change:

Min to max performance

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

76

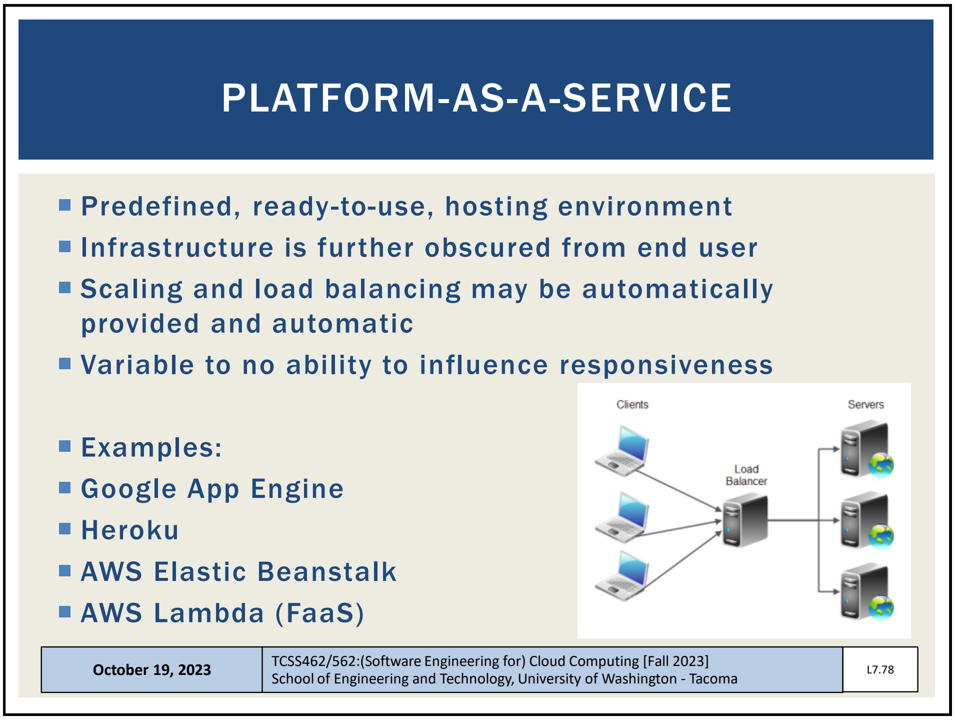


CLOUD COMPUTING DELIVERY MODELS

- 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

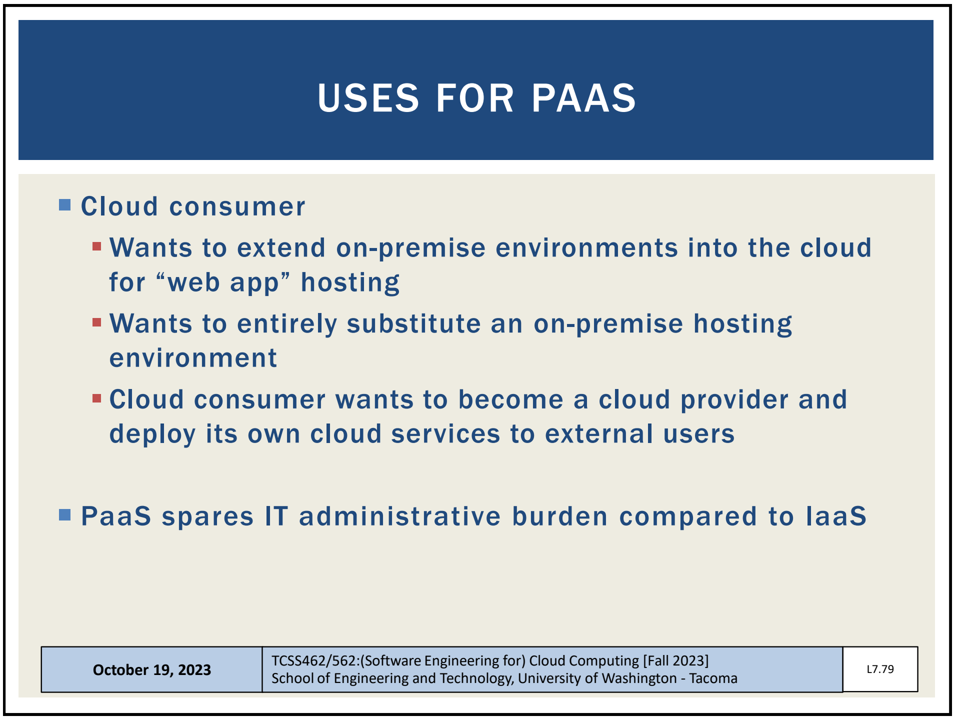


PLATFORM-AS-A-SERVICE

- 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)

Clients
Servers
Load Balancer

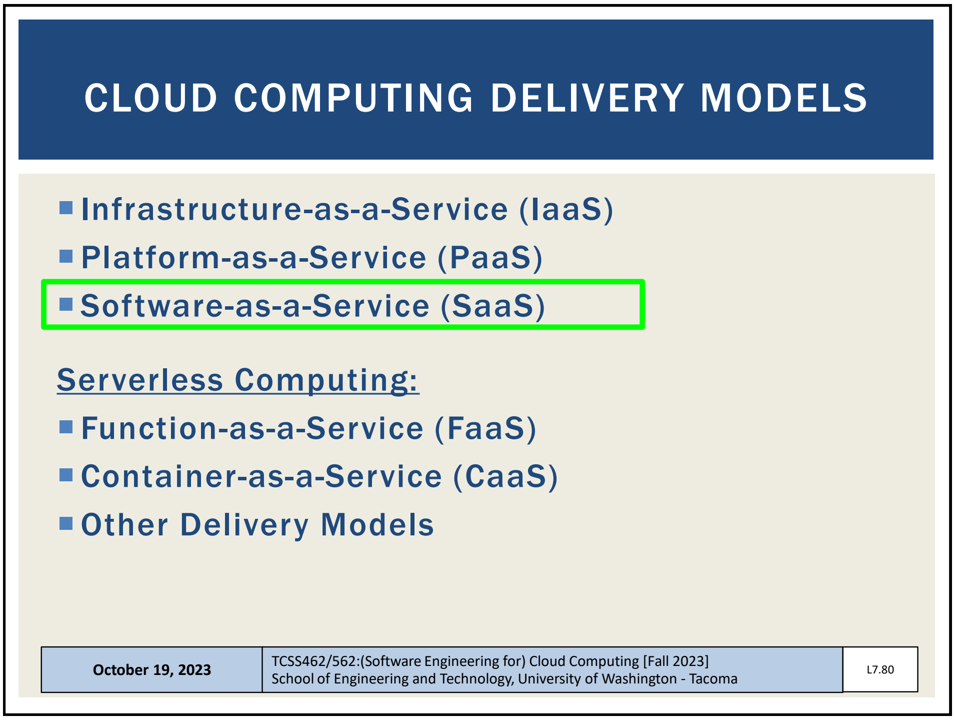


USES FOR PAAS

- 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 19, 2023	TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma	L7.79

79



CLOUD COMPUTING DELIVERY MODELS

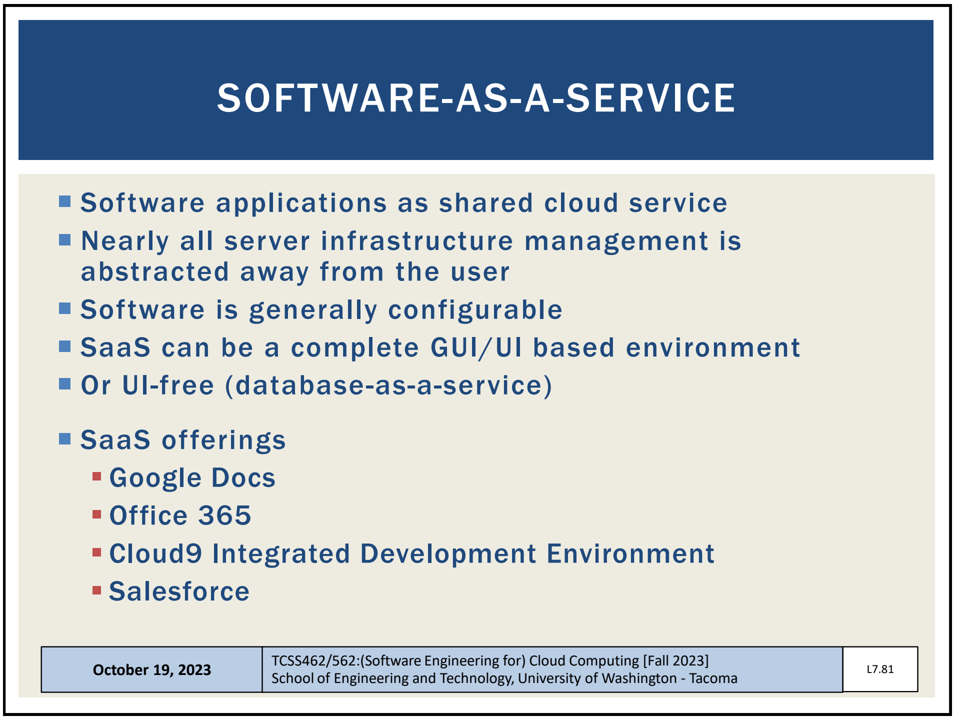
- Infrastructure-as-a-Service (IaaS)
- Platform-as-a-Service (PaaS)
- Software-as-a-Service (SaaS)

Serverless Computing:

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

October 19, 2023	TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma	L7.80

80

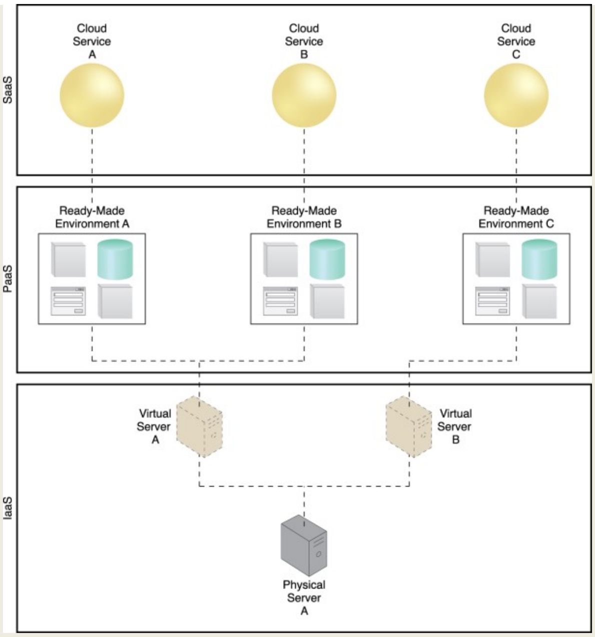


SOFTWARE-AS-A-SERVICE

- 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 19, 2023 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma | L7.81

81



October 19, 2023 TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma L7.82

82

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 19, 2023	TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma	L7.83

83

SERVERLESS COMPUTING

Introducing Cloud 2.0

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

84

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?

85

SERVERLESS COMPUTING

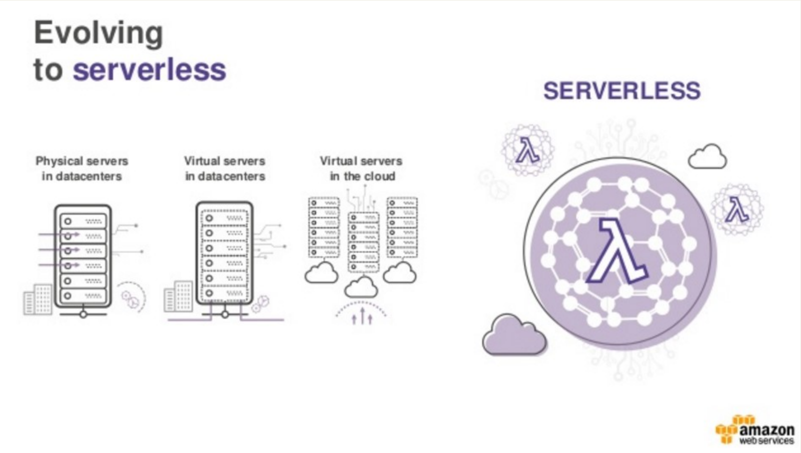
What is serverless?

Build and run applications without thinking about servers

October 19, 2023	TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma	L7.86

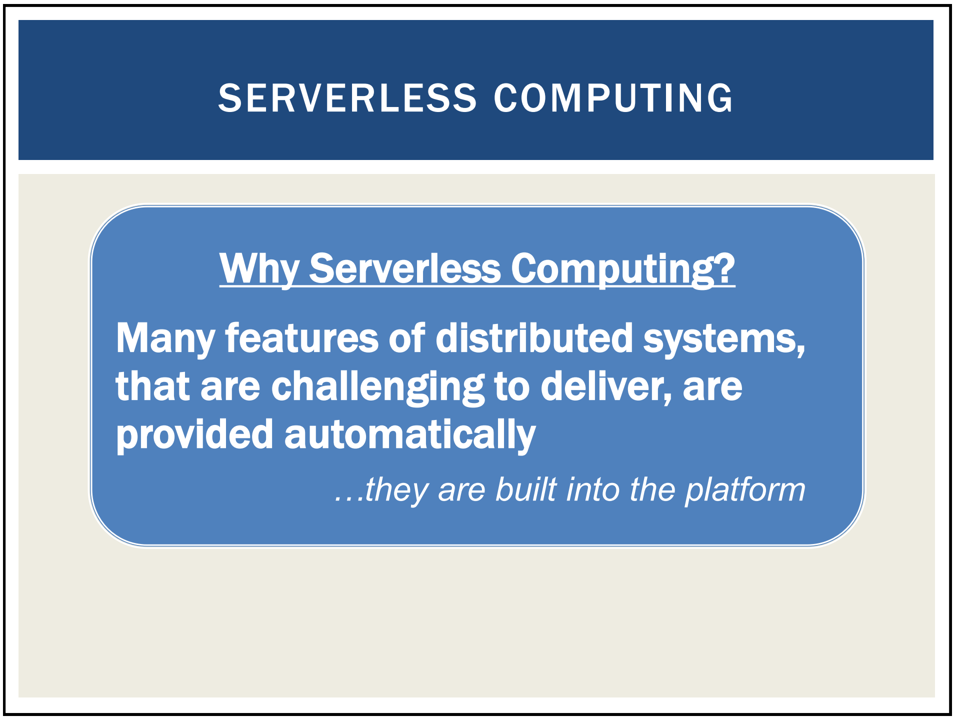
86

SERVERLESS COMPUTING - 2



SERVERLESS COMPUTING

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



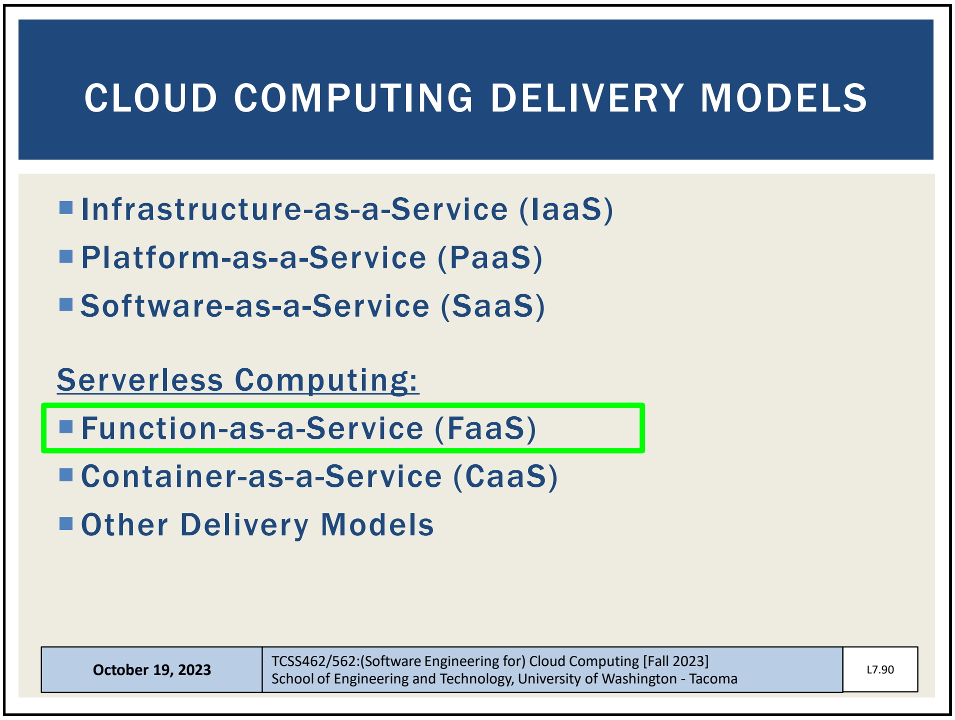
SERVERLESS COMPUTING

Why Serverless Computing?

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

…they are built into the platform

89



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

90

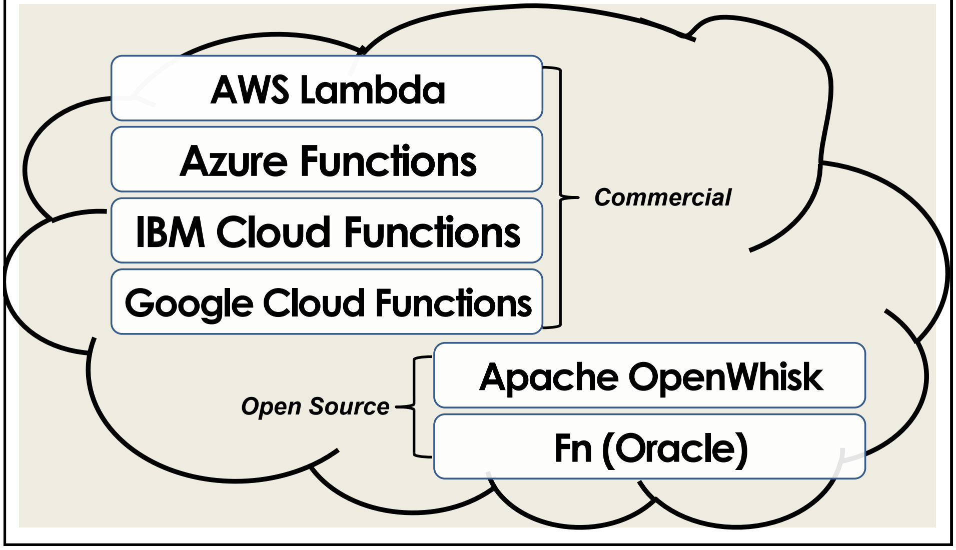
SERVERLESS VS. FAAS

- **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…

91

FAAS PLATFORMS



AWS Lambda

Azure Functions

IBM Cloud Functions

Google Cloud Functions

Commercial

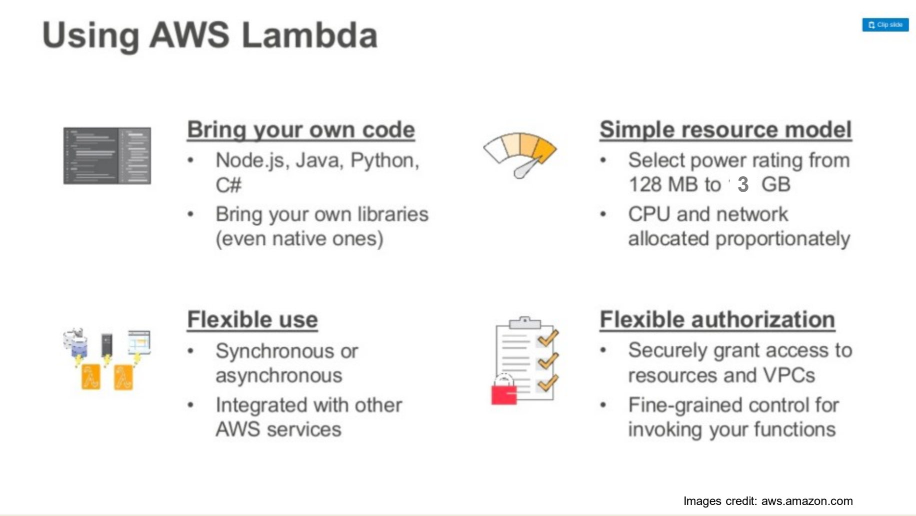
Apache OpenWhisk

Fn (Oracle)

Open Source

92

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

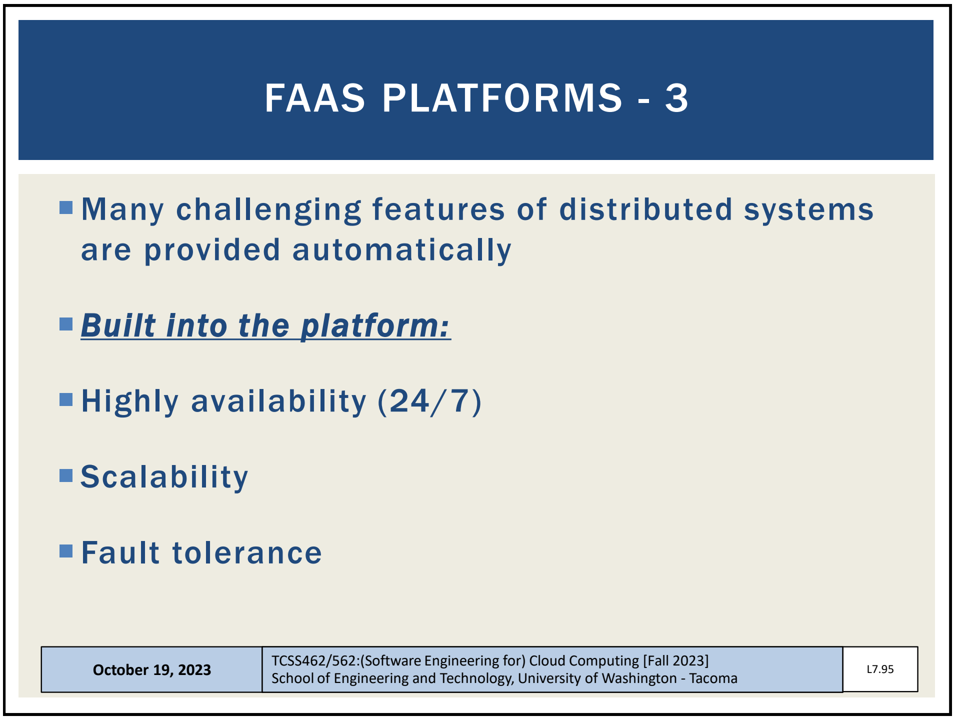
Flexible use
- Synchronous or asynchronous
- Integrated with other AWS services

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

Images credit: aws.amazon.com

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

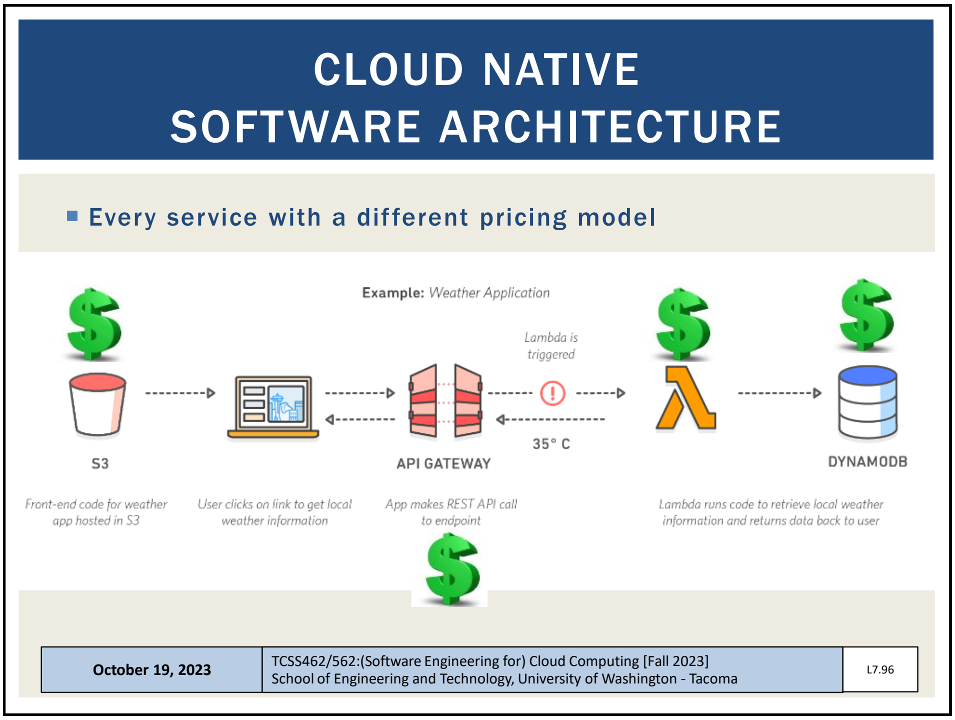


FAAS PLATFORMS - 3

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

October 19, 2023	TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma	L7.95

95



CLOUD NATIVE SOFTWARE ARCHITECTURE

- Every service with a different pricing model

Example: *Weather Application*

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 runs code to retrieve local weather information and returns data back to user

October 19, 2023	TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma	L7.96

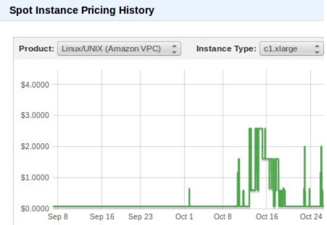
96

IAAS BILLING MODELS

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

```
          1000 computers   @     1 hour
     =       1 computer    @  1000 hours
```

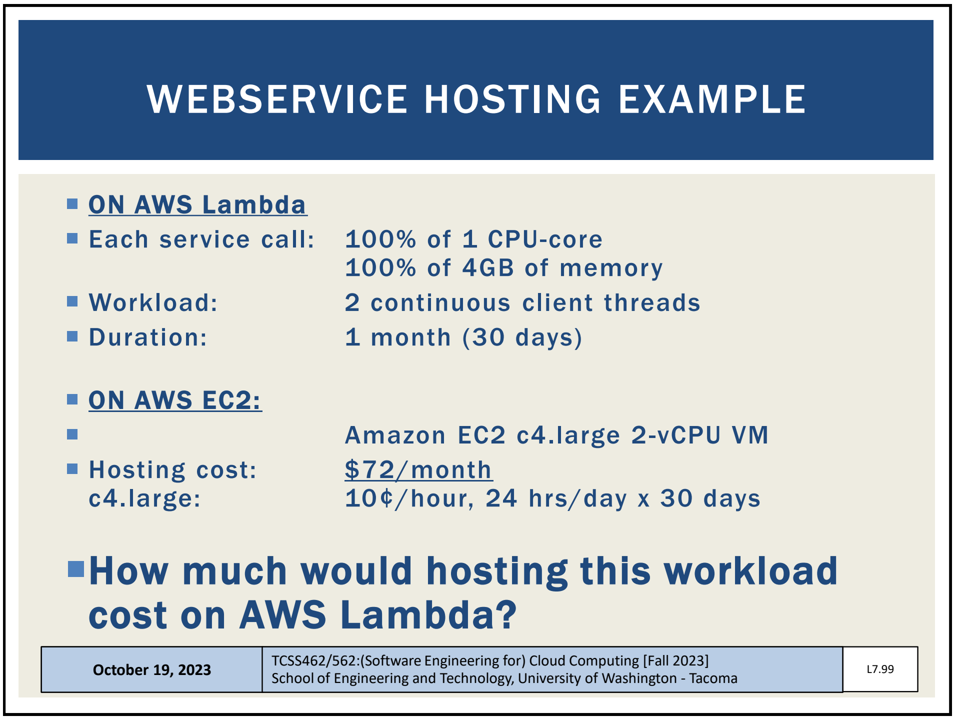
- Illusion of infinite scalability to cloud user
- As many computers as you can afford
- Billing models are becoming increasingly granular
  - By the minute, second, 1/10th sec
- Auction-based instances: Spot instances →


Spot Instance Pricing History

Product: Linux/UNIX (Amazon VPC)
Instance Type: c1.xlarge

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

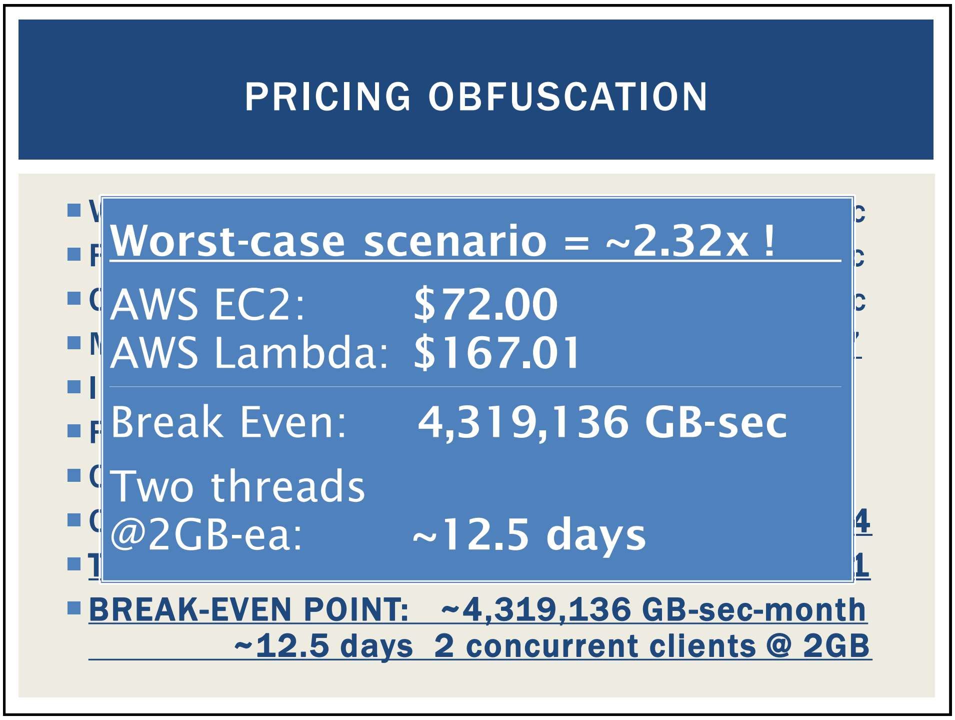
- **ON AWS Lambda**
- Each service call: 100% of 1 CPU-core
  100% of 4GB of memory
- Workload: 2 continuous client threads
- Duration: 1 month (30 days)

- **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 19, 2023 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma | L7.99

99



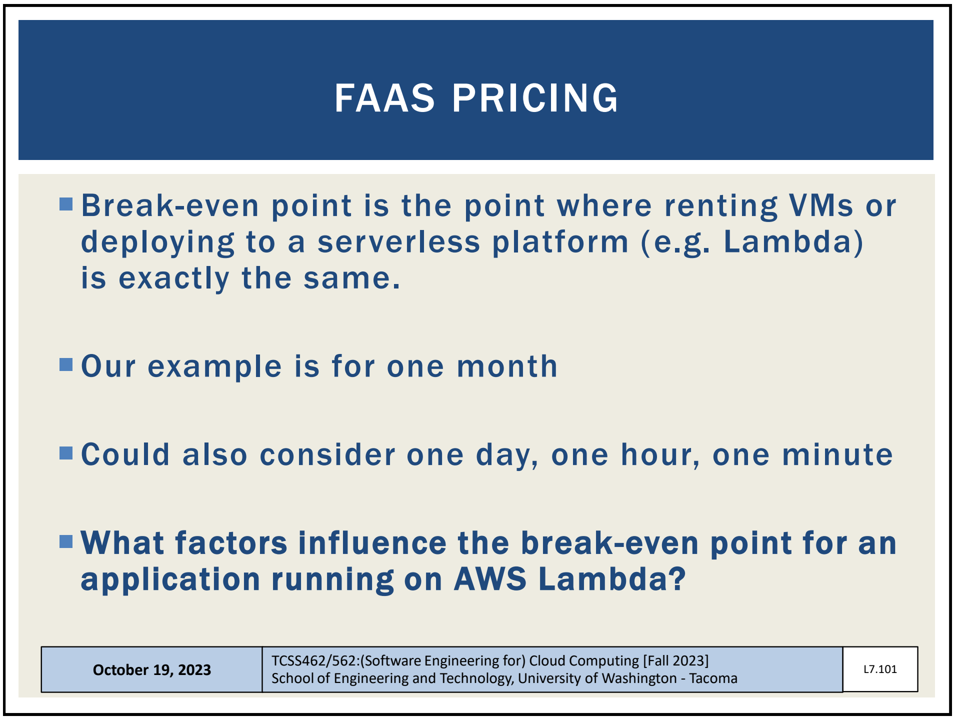
PRICING OBFUSCATION

Worst-case scenario = ~2.32x !

AWS EC2:	$72.00
AWS Lambda:	$167.01
Break Even:	4,319,136 GB-sec
Two threads @2GB-ea:	~12.5 days

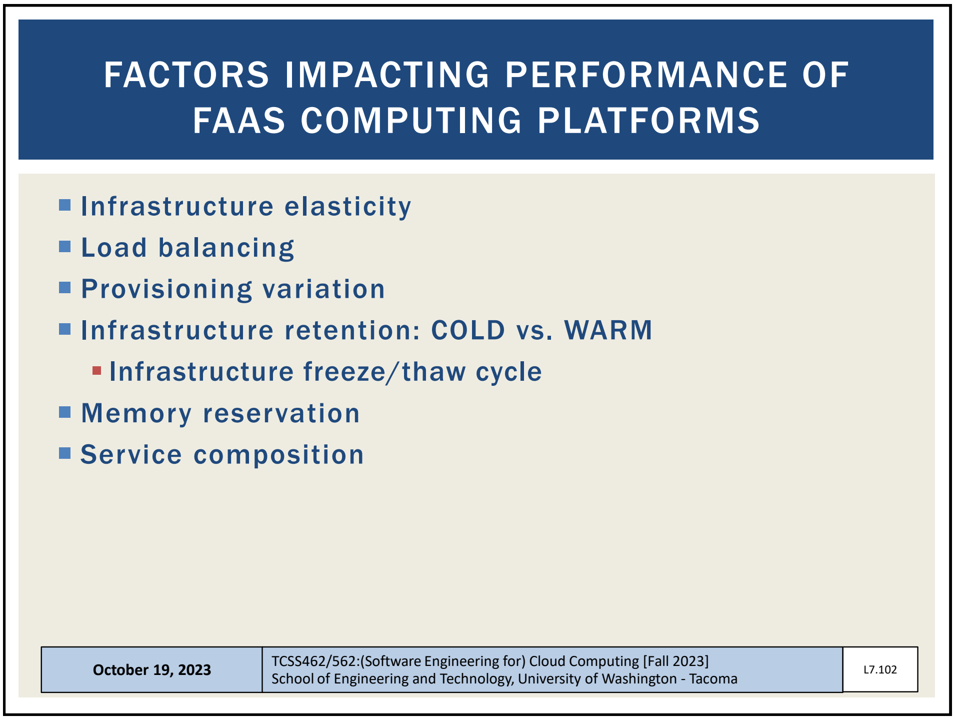
- **BREAK-EVEN POINT: ~4,319,136 GB-sec-month
  ~12.5 days 2 concurrent clients @ 2GB**

100



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?**



FACTORS IMPACTING PERFORMANCE OF FAAS COMPUTING PLATFORMS

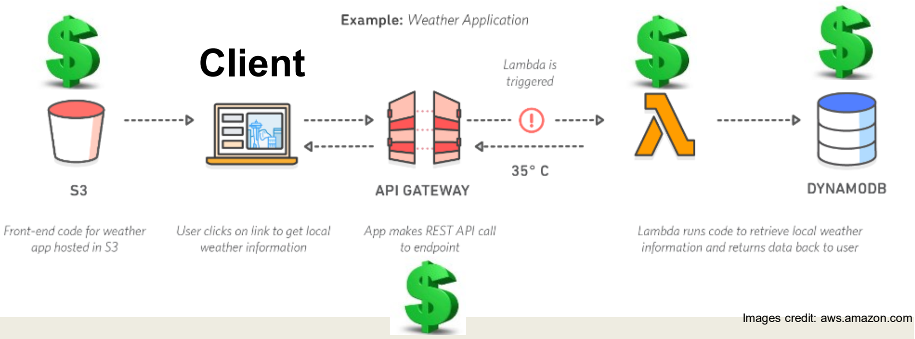
- Infrastructure elasticity
- Load balancing
- Provisioning variation
- Infrastructure retention: COLD vs. WARM
  - Infrastructure freeze/thaw cycle
- Memory reservation
- Service composition

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?

VENDOR ARCHITECTURAL LOCK-IN

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


Example: Weather Application
Client
Lambda is triggered
35° C
S3
API GATEWAY
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 runs code to retrieve local weather information and returns data back to user

Images credit: aws.amazon.com

- Increased dependencies → increased hosting costs

PRICING OBFUSCATION

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

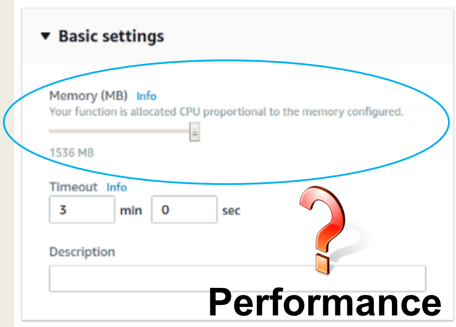
AWS Lambda Pricing

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

MEMORY RESERVATION QUESTION…

- 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…"*



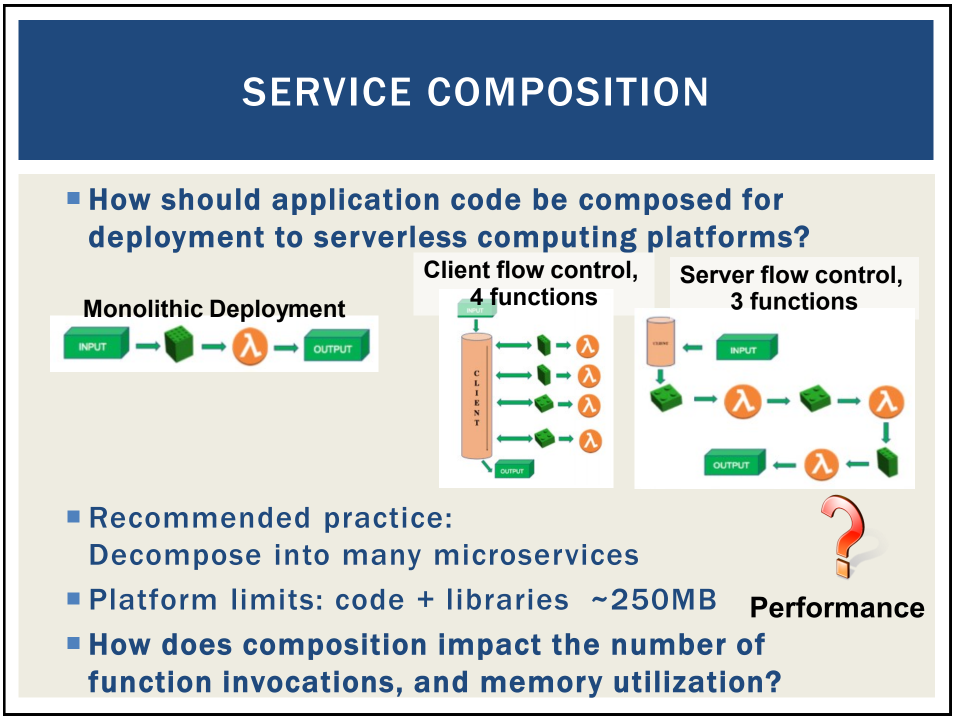
▼ Basic settings

Memory (MB) Info
Your function is allocated CPU proportional to the memory configured.
1536 MB

Timeout Info
3 min 0 sec

Description

- **But how much memory do model services require?**



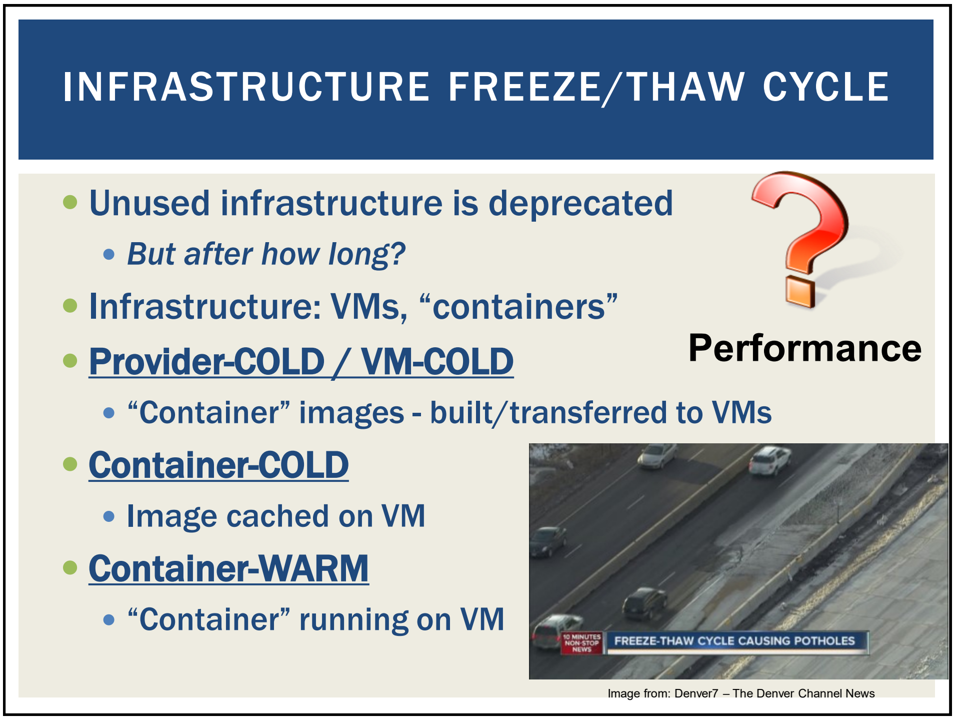
SERVICE COMPOSITION

- **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

107



INFRASTRUCTURE FREEZE/THAW CYCLE

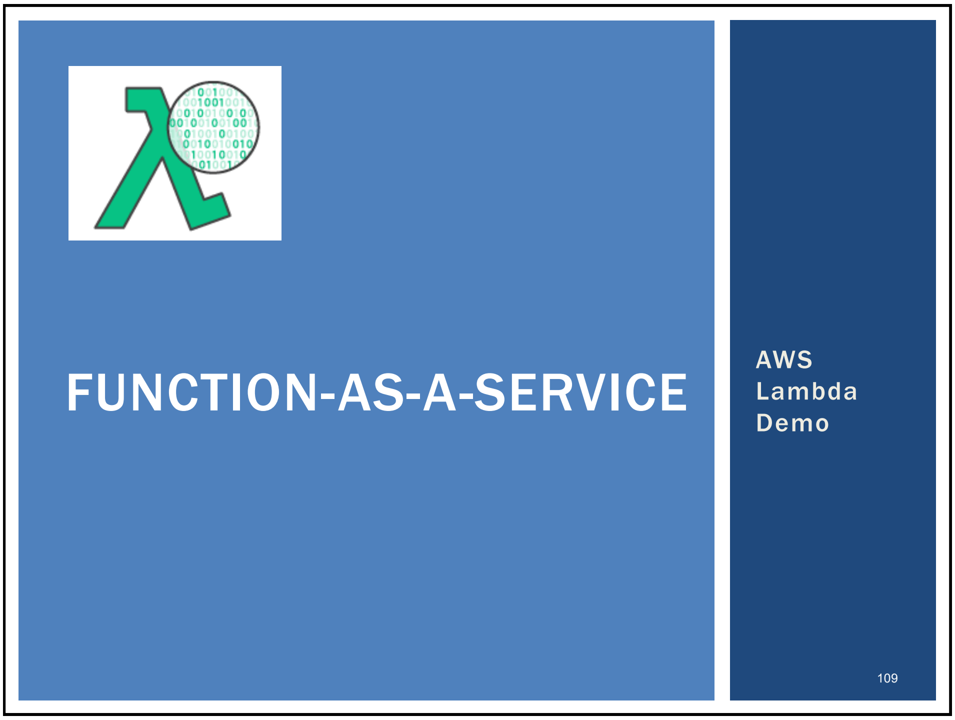
- Unused infrastructure is deprecated
  - *But after how long?*
- Infrastructure: VMs, “containers”
- **Provider-COLD / VM-COLD**
  - “Container” images - built/transferred to VMs
- **Container-COLD**
  - Image cached on VM
- **Container-WARM**
  - “Container” running on VM

Performance

10 MINUTES NON-STOP NEWS
FREEZE-THAW CYCLE CAUSING POTHOLES

Image from: Denver7 – The Denver Channel News

108

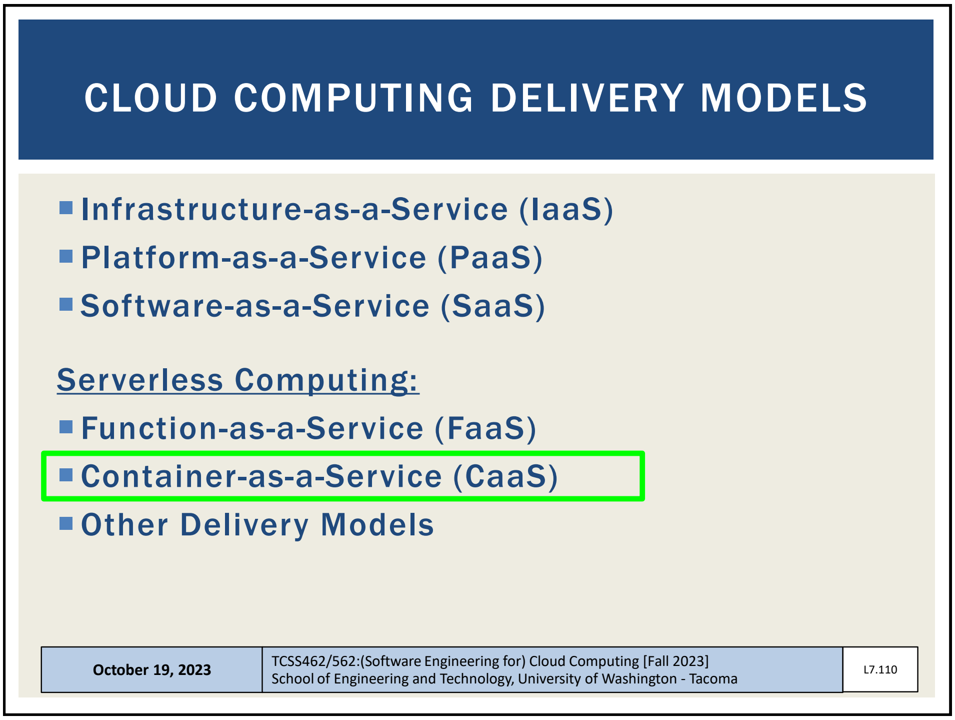


FUNCTION-AS-A-SERVICE

AWS Lambda Demo

109

109



CLOUD COMPUTING DELIVERY MODELS

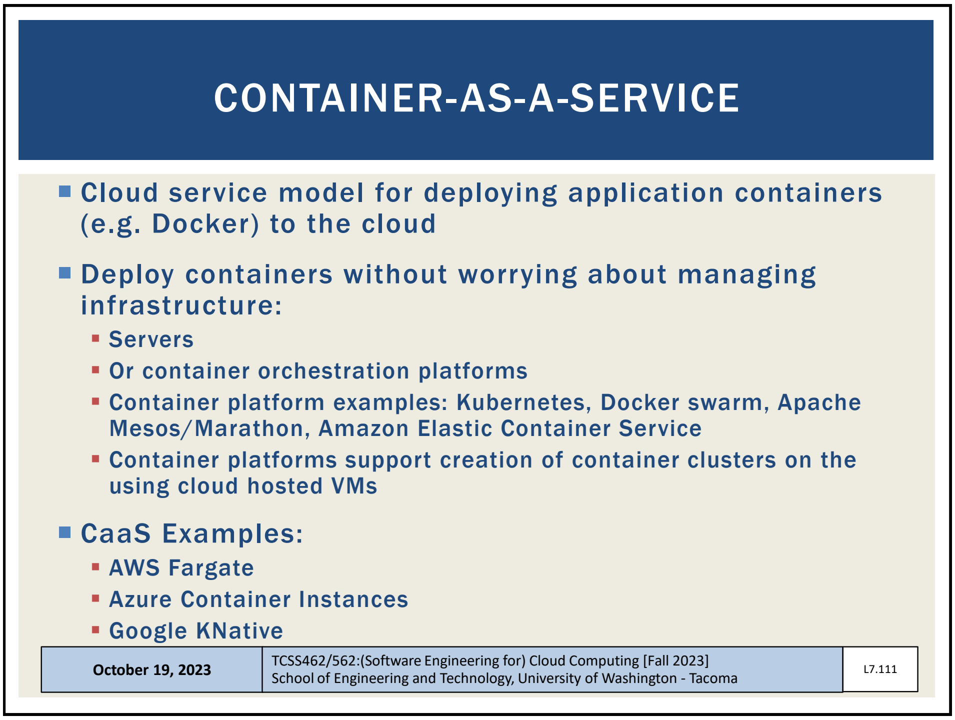
- Infrastructure-as-a-Service (IaaS)
- Platform-as-a-Service (PaaS)
- Software-as-a-Service (SaaS)

Serverless Computing:

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

October 19, 2023	TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma	L7.110

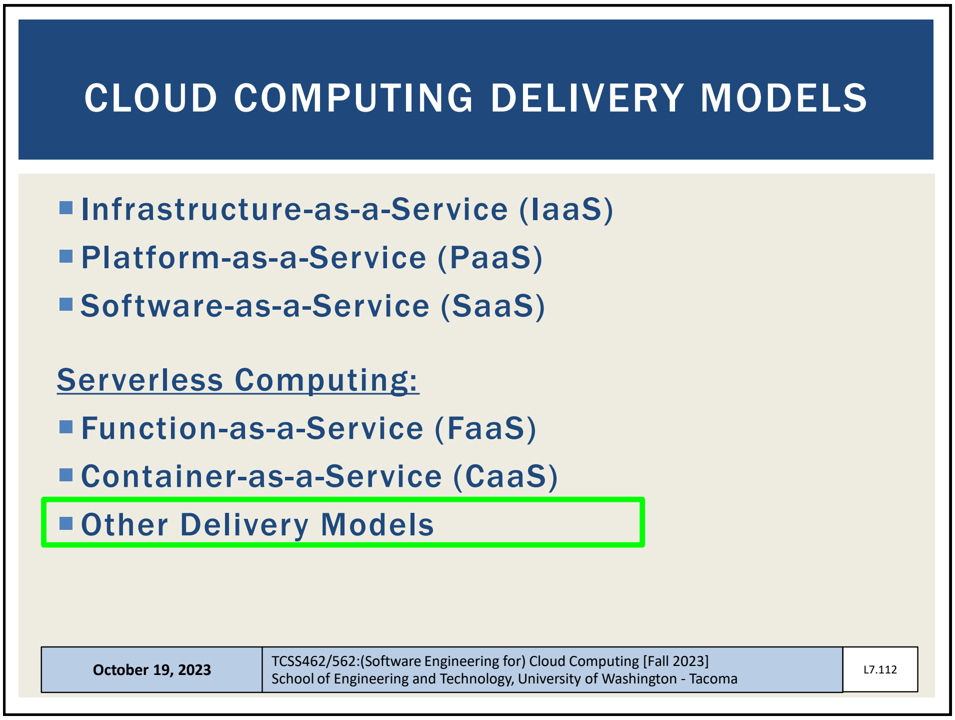
110



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

111



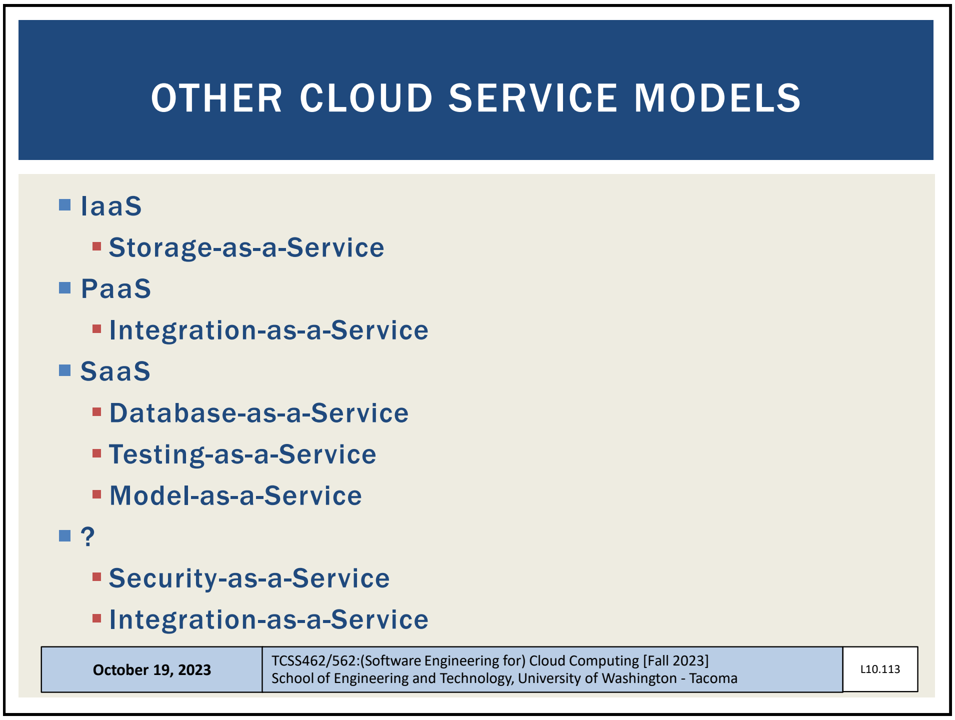
CLOUD COMPUTING DELIVERY MODELS

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

Serverless Computing:

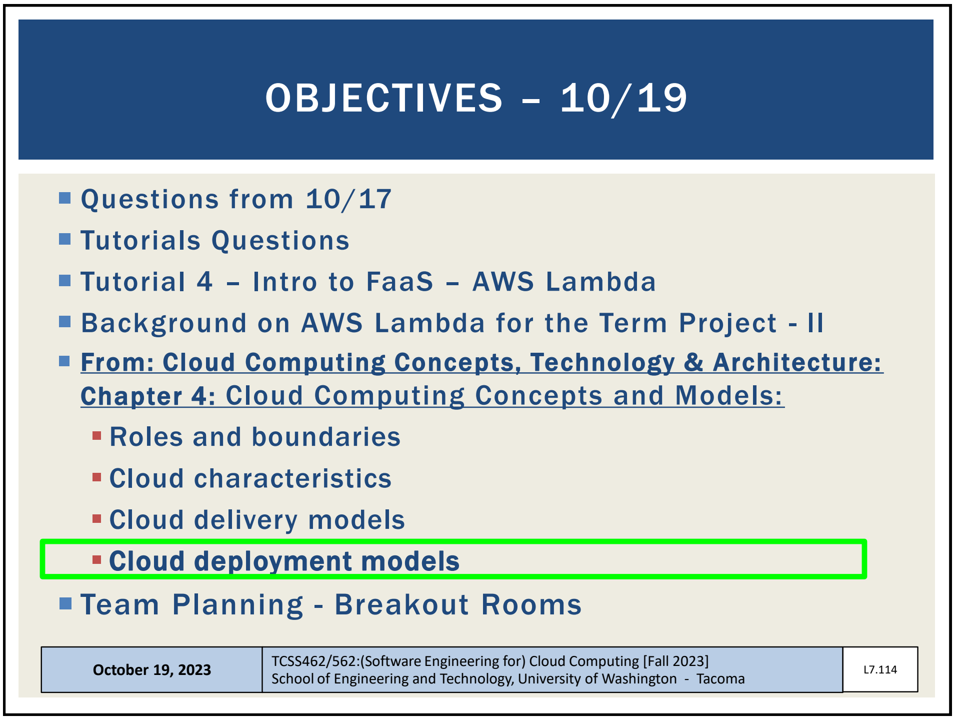
- Function-as-a-Service (FaaS)
- Container-as-a-Service (CaaS)
- Other Delivery Models

112



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



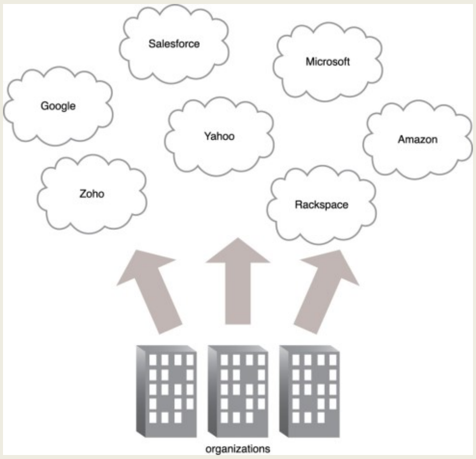
OBJECTIVES – 10/19

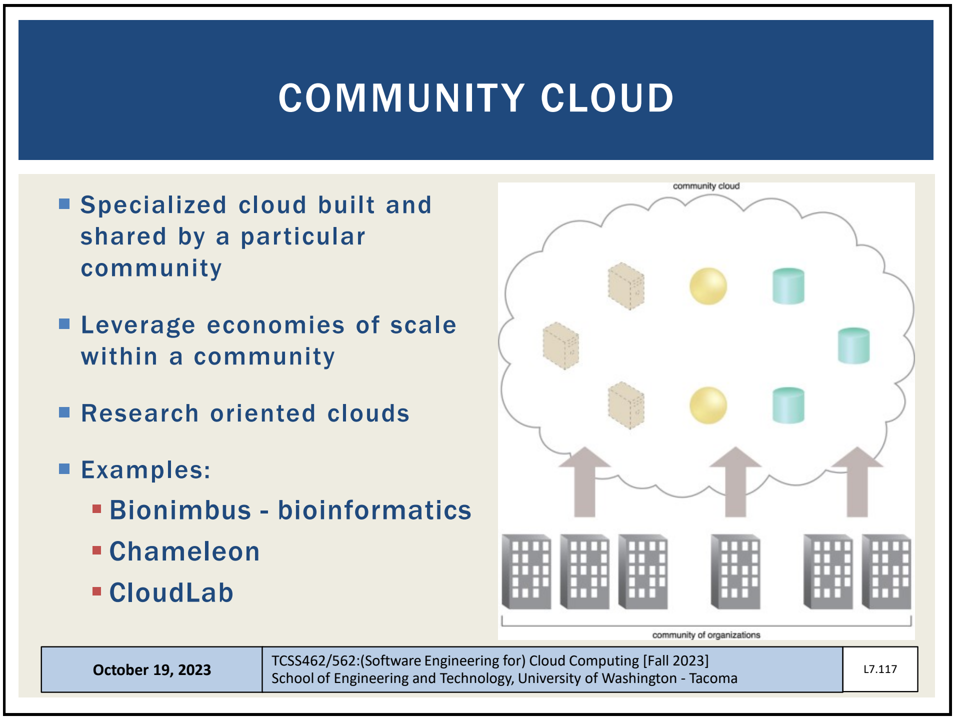
- Questions from 10/17
- Tutorials Questions
- Tutorial 4 – Intro to FaaS – AWS Lambda
- Background on AWS Lambda for the Term Project - II
- From: Cloud Computing Concepts, Technology & Architecture: Chapter 4: Cloud Computing Concepts and Models:
  - Roles and boundaries
  - Cloud characteristics
  - Cloud delivery models
  - **Cloud deployment models**
- Team Planning - Breakout Rooms

CLOUD DEPLOYMENT MODELS

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

PUBLIC CLOUDS





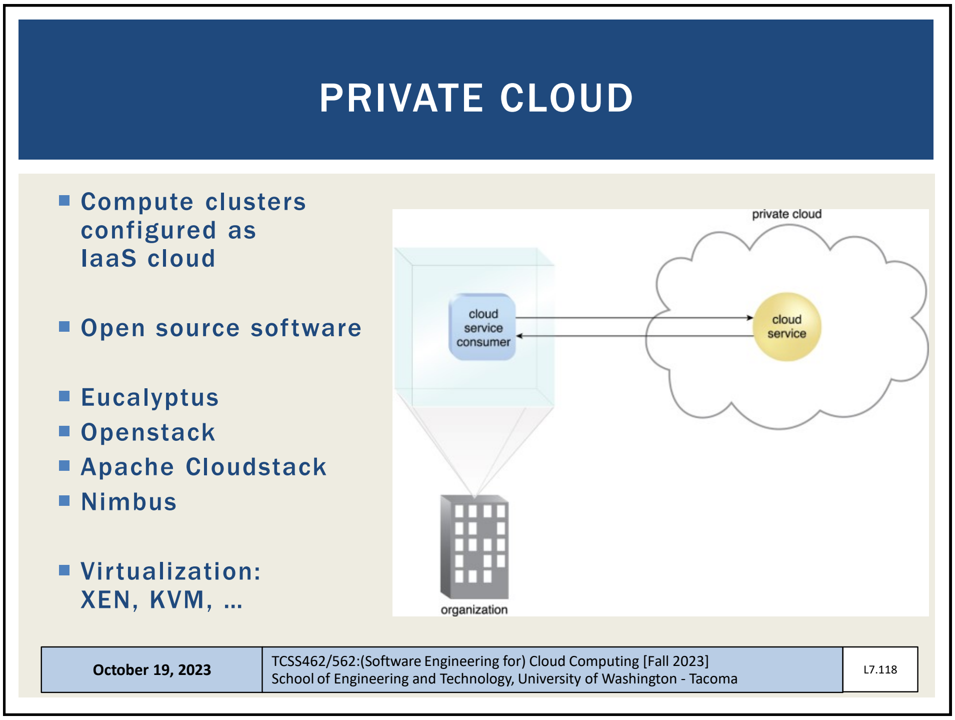
COMMUNITY CLOUD

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

community cloud
community of organizations

October 19, 2023 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma | L7.117

117



PRIVATE CLOUD

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

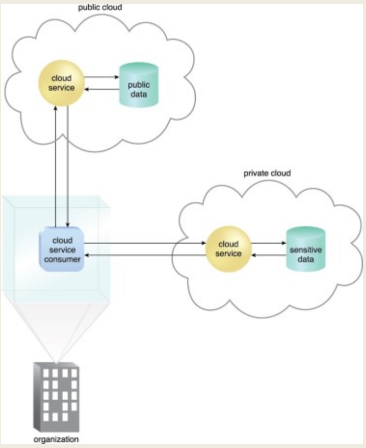
private cloud
cloud service consumer
cloud service
organization

October 19, 2023 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma | L7.118

118

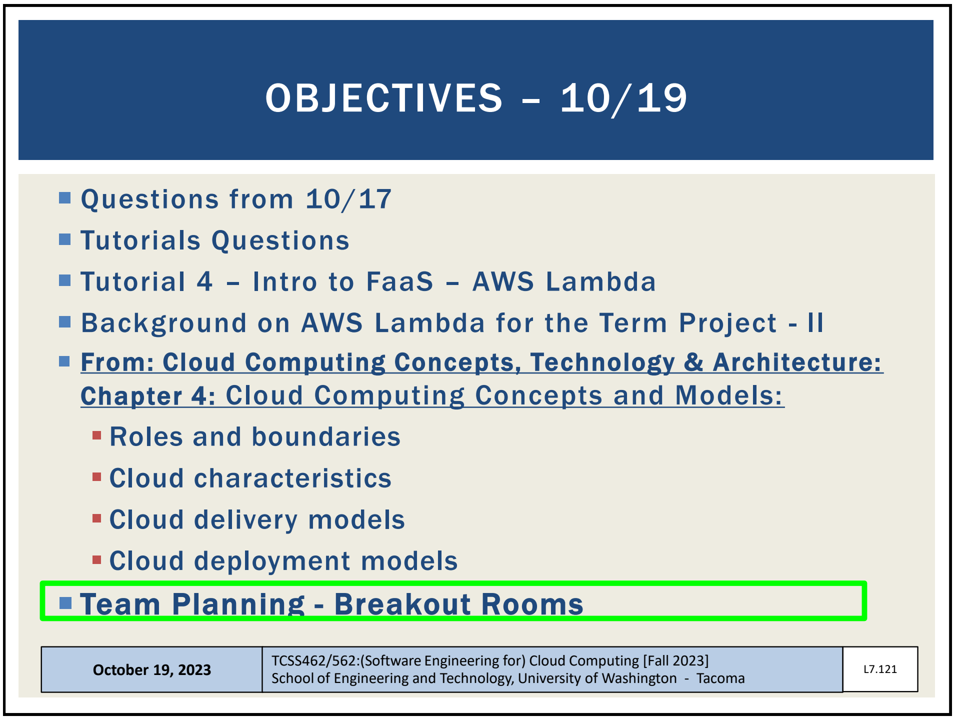
HYBRID CLOUD

- 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

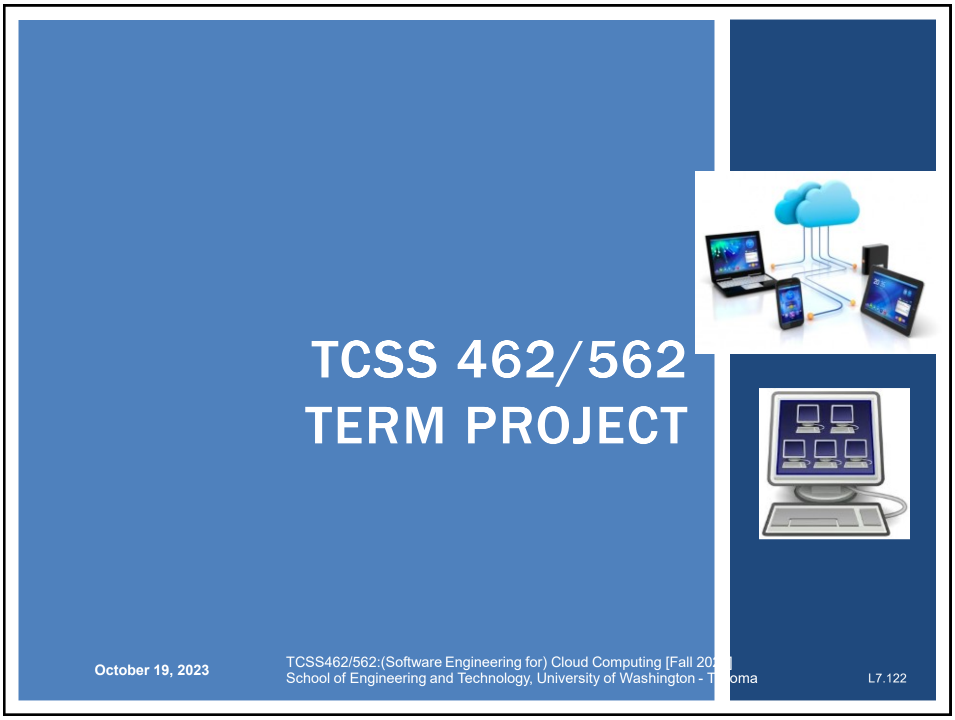


OBJECTIVES – 10/19

- Questions from 10/17
- 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 19, 2023	TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma	L7.121

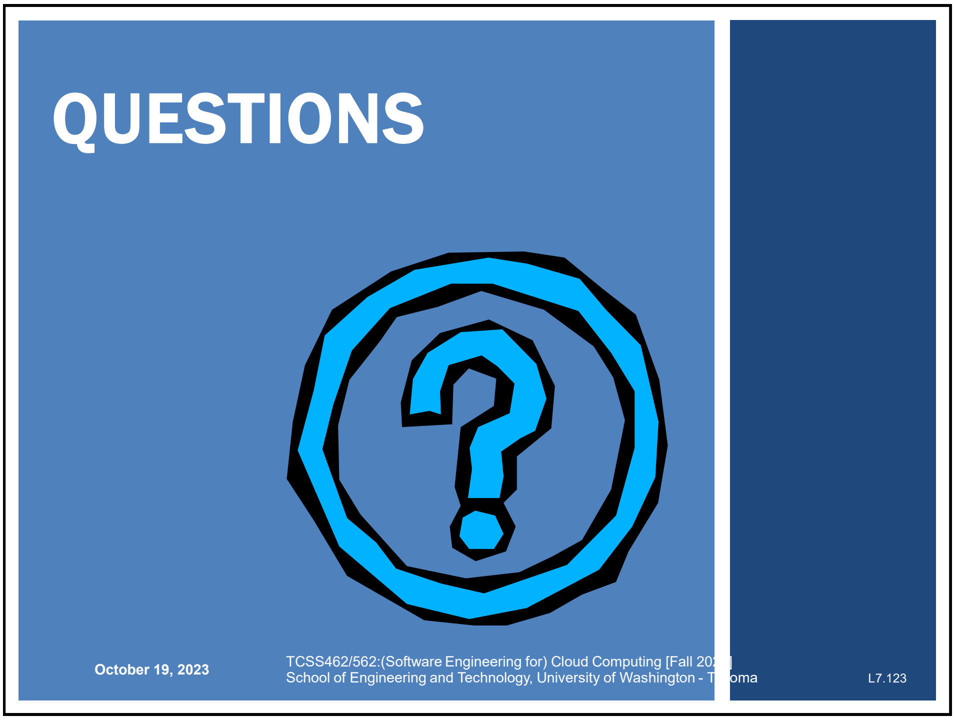
121



TCSS 462/562 TERM PROJECT

October 19, 2023 TCSS462/562:(Software Engineering for) Cloud Computing [Fall 20
School of Engineering and Technology, University of Washington - T oma L7.122

122



QUESTIONS

October 19, 2023

TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023]
School of Engineering and Technology, University of Washington - Tacoma

L7.123

123