TCSS 562: SOFTWARE ENGINEERING FOR CLOUD COMPUTING

AWS OVERVIEW & Cloud Enabling Technology

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

1

OFFICE HOURS – FALL 2025

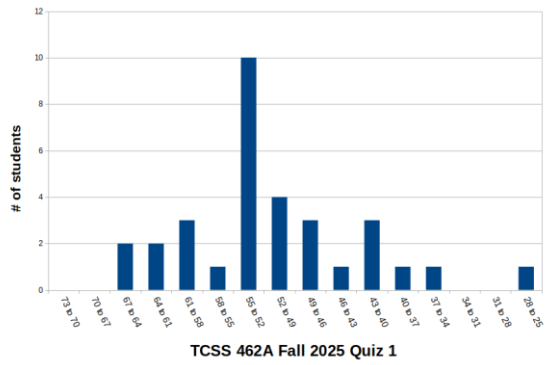
- **Thursdays:**
  - 6:00 to 7:00 pm - CP 229 & Zoom
- **Friday – *** THIS WEEK ***:**
  - 4:30 pm to 5:30 pm – ONLINE via Zoom
- **Or email for appointment**

- *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

QUIZ 1 RESULTS – TCSS 462

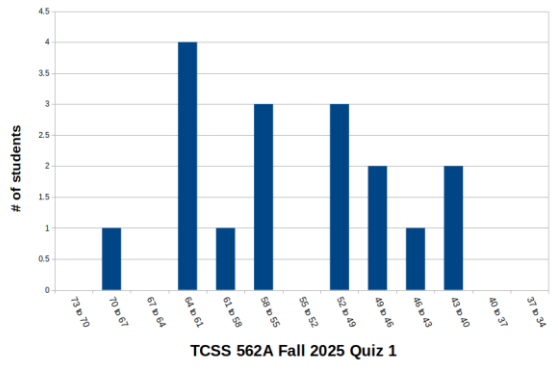
- **Statistics (out of 73)**
- Average: 50.84
- Mode: 52
- Median: 52
- Min score: 26
- Lower quartile: 48
- 2nd quartile: 52
- 3rd quartile: 54
- Max score: 65
- Standard deviation: 8.65
- Curve: +12.0
- Missing: 1



3

QUIZ 1 RESULTS – TCSS 562

- **Statistics (out of 73)**
- Average: 53.71
- Mode: 61
- Median: 55
- Min score: 40
- Lower quartile: 48
- 2nd quartile: 55
- 3rd quartile: 61
- Max score: 68
- Standard deviation: 7.95
- Curve: +9.0
- Missing: 0



4

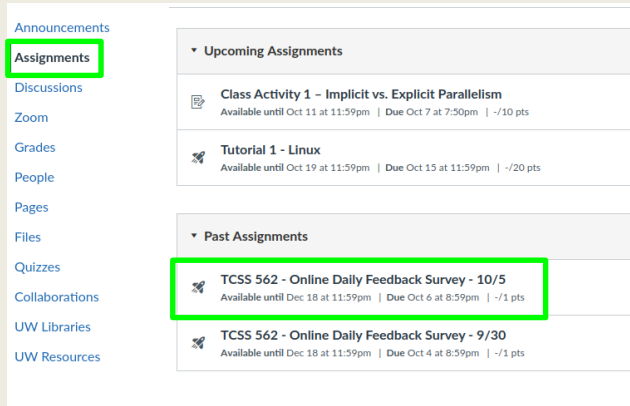
OBJECTIVES – 11/13

- **Questions from 11/6**
- Tutorials Questions
- Class Presentations:
  Cloud Technology or Research Paper Review
- Ch. 5: Cloud Enabling Technology
- Tutorial 5 Demo

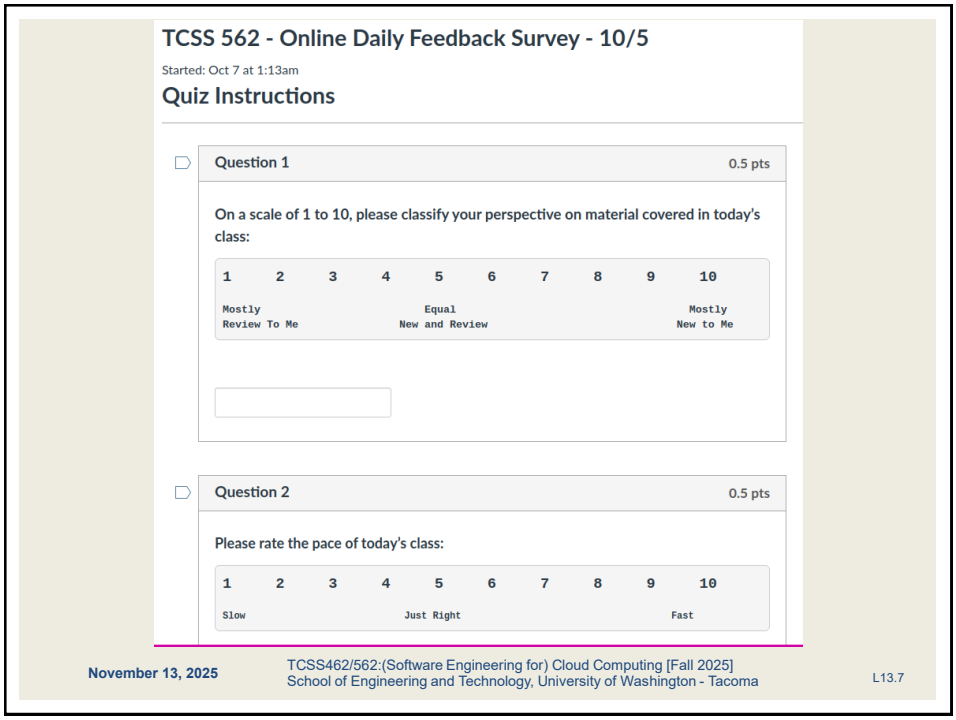
5

ONLINE DAILY FEEDBACK SURVEY

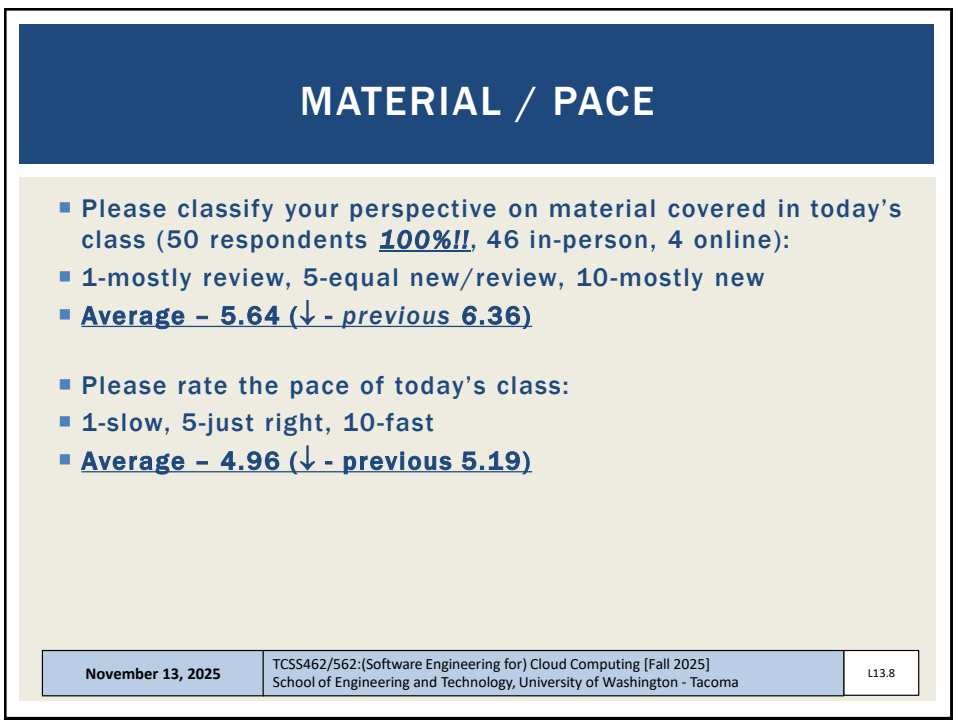
- Daily Feedback Quiz in Canvas – Take After Each Class
- Extra Credit for completing



6


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
November 13, 2025
TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025]
School of Engineering and Technology, University of Washington - Tacoma
L13.7

7


MATERIAL / PACE
Please classify your perspective on material covered in today's class (50 respondents 100%!!, 46 in-person, 4 online):
1-mostly review, 5-equal new/review, 10-mostly new
Average – 5.64 (↓ - previous 6.36)
Please rate the pace of today's class:
1-slow, 5-just right, 10-fast
Average – 4.96 (↓ - previous 5.19)
November 13, 2025
TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025]
School of Engineering and Technology, University of Washington - Tacoma
L13.8

8

FEEDBACK FROM 11/6

- ***Regarding a Cloud Technology Class Presentation, is a demonstration required ?***
  - Yes, a demonstration is required for a cloud technology presentation, though a LIVE demonstration is **not** required
  - The Assignment description provides some strategies
- ***Will Quiz 2 be similar to quiz 1 in terms of format, difficulty and/or scope ?***
  - It is likely to be similar
  - Quiz 2 will have less new content, and more review, since there are just 4 lectures before December 2nd
  - Lectures 12, 13, 14, 15, 16
  - Tutorials 5, 6, 7

9

OBJECTIVES – 11/13

- Questions from 11/6
- **Tutorials Questions**
- Class Presentations:
  Cloud Technology or Research Paper Review
- Ch. 5: Cloud Enabling Technology
- Tutorial 5 Demo

10

Don't Forget to Terminate (Shutdown) all EC2 instances for Tutorials 3

Spot instances:
c5d.large instance @ ~3.2 cents / hour

$0.78 / day
$5.48 / week
$23.78 / month
$285.42 / year

AWS CREDITS → → → → → → → →

11

TUTORIAL 4 – DUE NOV 11

- Introduction to AWS Lambda with the Serverless Application Analytics Framework (SAAF)
  https://faculty.washington.edu/wlloyd/courses/tcss562/tutorials/TCSS462_562_f2025_tutorial_4.pdf
- Set up Java development environment
- Introduction to Maven build files for Java
- Create and Deploy "hello" Java AWS Lambda Function
- Create API Gateway REST endpoint
- Sequential testing of "hello" AWS Lambda Function
  - API Gateway endpoint, AWS Lambda CLI Function invocation, AWS Function URL
- Profiling function performance with SAAF
- Concurrent function testing with faas_runner
- Performance analysis using faas_runner reports
- Two function pipeline development task: Caesar Cipher

12

TUTORIAL 4 - NOTES

- Sometimes students get unexpected results when adding **Thread.sleep(10000)** to the hello Lambda function:
- It is possible that:

1. Results from the SAAF Report Generator were from a test run before the **Thread.Sleep()** statement was added to the code
   - OR -
2. The **Thread.Sleep()** statement was added in the incorrect location of the code
   - OR -
3. When opening the CSV output from the Report Generator, the file separator characters were set incorrectly.

- The only separator for a CSV file is the comma ","
  Be sure to correctly open the CSV file in the spreadsheet.
  Columns can be offset resulting in the wrong answers being provided for Question 6.

TUTORIAL 4 – NOTES - 2

- The sleep statement must go between the START FUNCTION and END FUNCTION comments in the handleRequest() method specified as the AWS Lambda function's handler under runtime settings in the AWS Lambda GUI.

```
//****************START FUNCTION IMPLEMENTATION*************************
    try
    {
        Thread.sleep(10000);
    }
    catch (InterruptedException ie)
    {
        System.out.println("Interruption occurred while sleeping.");
    }
//****************END FUNCTION IMPLEMENTATIO N**************************
```

- Code must be recompiled and redeployed after modification

TUTORIAL 4 – NOTES - 3

- **SANITY CHECK:** consider that adding 10 seconds of sleep to your AWS Lambda function will cause the function to run for at least 10 seconds. This will impact the outputs requested for Question 6:

- **avg_runtime** is the server-side (cloud) runtime of the function
- This is the time it takes for the function to run on AWS Lambda (cloud)
- Adding sleep of 10 seconds should increase a function's **avg_runtime**

- **avg_roundTripTime** is the total time for a request from a client (laptop?) to travel to the server (cloud), make the function call, and return.
- If trying to make 50 calls at once on a laptop with a small # of CPU cores this time may be slow
- Adding sleep of 10 seconds should increase a function's **avg_roundTripTime**

15

TUTORIAL 4 – NOTES - 4

- **avg_cpuIdleDelta** time is the amount of time the Lambda function's Firecracker two vCPUs are idle during the function call on the server measured in centiseconds:

100 centiseconds = 1 second
100 centiseconds = 1000 milliseconds

- By default, AWS Lambda functions with 512 MB run in a runtime environment with access to two vCPU cores
- This is the total vCPU idle time for both cores (it is doubled)
- Adding sleep of 10 seconds should increase your function's avg_cpuIdleDelta

- **How much should avg_cpuIdleDelta increase ?**

16

TUTORIAL 5 – DUE NOV 16

- Introduction to Lambda II: Working with Files in S3, Cloud Trail, and Amazon Event Bridge Rules
- https://faculty.washington.edu/wlloyd/courses/tcss562/tutorials/TCSS462_562_f2025_tutorial_5.pdf
- Customize the Request object (add getters/setters)
  - Why do this instead of HashMap ?
- Import dependencies (jar files) into project for AWS S3
- Create an S3 Bucket
- Give your Lambda function(s) permission to work with S3
- Write to the CloudWatch logs
- Use of CloudTrail to generate S3 events
- Creating Event Bridge rule to capture events from CloudTrail
- Have the Event Bridge rule trigger a Lambda function with a static JSON input object (hard-coded filename)
- **Optional**: for the S3 PutObject event, dynamically extract the name of the file put to the S3 bucket for processing

17

TUTORIAL 6 – NOV 23

- Introduction to Lambda III: Serverless Databases
- https://faculty.washington.edu/wlloyd/courses/tcss562/tutorials/TCSS462_562_f2025_tutorial_6.pdf

- Create and use Sqlite databases using sqlite3
- Deploy Lambda function with Sqlite3 database under /tmp
- Compare in-memory vs. file-based Sqlite DBs on Lambda
- Create an Amazon Aurora "Serverless" v2 MySQL database
- Using the AWS CloudShell in the same VPC (Region + availability zone) connect and interact your Aurora serverless database using the mysql CLI app
- Deploy an AWS Lambda function that uses the MySQL "serverless" database

18

TUTORIAL 7

- Introduction to Docker
- To be posted

- Must complete using Ubuntu 24.04 (for cgroups v2)
- Use DOCX file for copying and pasting Docker install commands
- Topics:
  - Installing Docker
  - Creating a container using a Dockerfile
  - Using cgroups virtual filesystem to monitor CPU utilization of a container
  - Persisting container images to Docker Hub image repository
  - Container vertical scaling of CPU/memory resources
  - Testing container CPU and memory isolation

19

TUTORIAL COVERAGE

- **Docker CLI → Docker Engine (dockerd) → containerd → runc**

- Working with the docker CLI:

- docker run create a container
- docker ps -a list containers, find CONTAINER ID
- docker exec --it run a process in an existing container
- docker stop stop a container
- docker kill kill a container
- docker help list available commands
- man docker Docker Linux manual pages

20

21

TUTORIAL 7

- Tutorial introduces use of two common Linux performance benchmark applications
- stress-ng
- 100s of CPU, memory, disk, network stress tests
- Sysbench
- Used in tutorial for memory stress test

22

OBJECTIVES – 11/13

- Questions from 11/6
- Tutorials Questions
- **Class Presentations:**
  Cloud Technology or Research Paper Review
- Ch. 5: Cloud Enabling Technology
- Tutorial 5 Demo

23

GROUP PRESENTATION

- TWO OPTIONS:
- ***Cloud technology presentation***
- ***Cloud research paper presentation***
  - Recent & suggested papers will be posted at:
    http://faculty.washington.edu/wlloyd/courses/tcss562/papers/
- Submit presentation type and topics (paper or technology) with desired dates of presentation via Canvas by:
  Tuesday November 18th @ 11:59pm
- Presentation dates:
  - Tuesday November 25
  - Tuesday December 2*, Thursday December 4
    - ** - day of quiz 2. only 1 presentation slot*

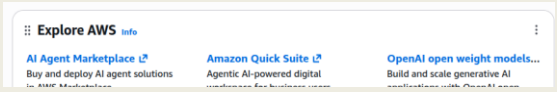
24

AWS OVERVIEW AND DEMO

25

EARN ADDITIONAL AWS CREDITS

- Earn additional $100 in AWS credits :
- Complete 5 activities from the "Explore AWS Widget":



  - Launch an instance using Amazon EC2
  - Use a foundational model in the Amazon Bedrock playground
  - Set up a cost budget using AWS Budgets
  - Create a web app using AWS Lambda
  - Create an Amazon RDS database
- https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/free-tier-plans-activities.html

26

LIST OF TOPICS

- AWS Management Console
- Elastic Compute Cloud (EC2)
- Instance Storage: Virtual Disks on VMs
- Elastic Block Store: Virtual Disks on VMs
- Elastic File System (EFS)
- Amazon Machine Images (AMIs)
- EC2 Paravirtualization
- EC2 Full Virtualization (hvm)
- EC2 Virtualization Evolution
- (VM) Instance Actions
- EC2 Networking
- EC2 Instance Metadata Service
- Simple Storage Service (S3)
- AWS Command Line Interface (CLI)
- Legacy / Service Specific CLIs
- AMI Tools
- Signing Certificates
- Backing up live disks
- Cost Savings Measures
- Disk images and S3

27

AWS: provides an ever expanding array of cloud services…

Services by category

Compute
EC2
Lightsail
Lambda
Batch
Elastic Beanstalk
Serverless Application Repository
AWS Outposts
EC2 Image Builder
AWS App Runner
AWS SimSpace Weaver

Containers
Elastic Container Registry
Elastic Container Service
Elastic Kubernetes Service
Red Hat OpenShift Service on AWS

Storage
S3
EFS
FSx
S3 Glacier
Storage Gateway
AWS Backup
AWS Elastic Disaster Recovery

Database
RDS
ElastiCache
Neptune
Amazon QLDB
Amazon DocumentDB
Amazon Keyspaces
Amazon Timestream
DynamoDB
Amazon MemoryDB for Redis

Migration & Transfer
AWS Migration Hub
AWS Application Migration Service
Application Discovery Service
Database Migration Service
AWS Transfer Family
AWS Snow Family
DataSync
AWS Mainframe Modernization

Networking & Content Delivery
VPC
CloudFront
Route 53
API Gateway

Customer Enablement
AWS IQ
Managed Services
Activate for Startups
Support

Robotics
AWS RoboMaker

Blockchain
Amazon Managed Blockchain

Satellite
Ground Station

Quantum Technologies
Amazon Braket

Management & Governance
AWS Organizations
CloudWatch
AWS Auto Scaling
CloudFormation
Config
OpsWorks
Service Catalog
Systems Manager
Trusted Advisor
Control Tower
AWS License Manager
AWS Well-Architected Tool
AWS Health Dashboard
AWS Chatbot
Launch Wizard
AWS Compute Optimizer
Resource Groups & Tag Editor
Amazon Grafana
Amazon Prometheus
AWS Resilience Hub
Incident Manager
Service Quotas
AWS Proton
CloudTrail
AWS Resource Explorer
AWS User Notifications
AWS Telco Network Builder

Media Services
Kinesis Video Streams
MediaConvert
MediaLive
MediaPackage
MediaStore

Machine Learning
Amazon SageMaker
Amazon Augmented AI
Amazon CodeGuru
Amazon DevOps Guru
Amazon Comprehend
Amazon Forecast
Amazon Fraud Detector
Amazon Kendra
Amazon Personalize
Amazon Polly
Amazon Rekognition
Amazon Textract
Amazon Transcribe
Amazon Translate
AWS DeepComposer
AWS DeepLens
AWS DeepRacer
AWS Panorama
Amazon Monitron
AWS HealthLake
Amazon Lookout for Vision
Amazon Lookout for Equipment
Amazon Lookout for Metrics
Amazon Lex
Amazon Comprehend Medical
AWS HealthOmics
Amazon Bedrock
AWS HealthImaging

Analytics
Athena
Amazon Redshift
CloudSearch
Amazon OpenSearch Service
Kinesis
QuickSight
Data Pipeline
AWS Data Exchange
AWS Lake Formation
MSK
AWS Glue DataBrew
Amazon FinSpace
AWS Glue
EMR
AWS Clean Rooms
Amazon DataZone
AWS Entity Resolution
Managed Apache Flink

Security, Identity, & Compliance
IAM
Resource Access Manager
Cognito

AWS Cost Management
AWS Cost Explorer
AWS Budgets
AWS Marketplace Subscriptions
AWS Application Cost Profiler
AWS Billing Conductor
Billing

Front-end Web & Mobile
AWS Amplify
AWS AppSync
Device Farm
Amazon Location Service

Application Integration
Step Functions
Amazon AppFlow
Amazon EventBridge
Amazon MQ
Simple Notification Service
Simple Queue Service
SWF
Managed Apache Airflow

Business Applications
Amazon Connect
Amazon Pinpoint
Amazon Honeycode
Amazon Chime
Amazon Simple Email Service
Amazon WorkDocs
Amazon WorkMail
AWS Supply Chain
AWS AppFabric
AWS Wickr
Amazon Chime SDK

End User Computing
WorkSpaces
AppStream 2.0
WorkSpaces Web

Internet of Things
IoT 1-Click
IoT Analytics
IoT Device Defender
IoT Device Management
IoT Greengrass
IoT SiteWise
IoT Core
IoT Events
AWS IoT FleetWise
IoT RoboRunner

28

AWS EC2

- Elastic Compute Cloud
- Instance types: https://instances.vantage.sh/
  - **On demand Instance** – full price
  - **Reserved Instance** – contract based where customer guarantees VM rental for a fixed period of time (e.g. 1 year, 3 years, etc.)
    Deeper discounts with longer term commitments
  - **Spot Instance** – portion of cloud capacity reserved for low cost instances, when demand exceeds supply instances are randomly terminated with 2-minute warning
    - Users can make diverse VM requests using different types, zones, regions, etc. to minimize instance terminations
    - Developers can design for failure because often only 1 or 2 VMs in a cluster fail at any given time. They then need to be replaced.
  - **Dedicated host** – reserved private HW (server)
  - Instance families -
    General, compute-optimized, memory-optimized, GPU, etc.

29

AWS EC2 - 2

- Storage types
  - Instance storage - ephemeral storage
    - Temporary disk volumes stored on disks local to the VM
    - Evolution: physical hard disk drives (HDDs)
    - Solid state drives (SSDs)
    - Non-volatile memory express (NVMe) drives (closer to DRAM speed)
  - EBS - Elastic block store
    - Remotely hosted disk volumes
  - EFS - Elastic file system
    - Shared file system based on network file system
    - VMs, Lambdas, Containers mount/interact with shared file system
    - Somewhat expensive

30

INSTANCE STORAGE

- Also called ephemeral storage
- Persisted using images saved to S3 (simple storage service)
  - ~2.3¢ per GB/month on S3
  - 5GB of free tier storage space on S3
- Requires "burning" an image
- Multi-step process:
  - Create image files
  - Upload chunks to S3
  - Register image
- Launching a VM
  - Requires downloading image components from S3, reassembling them… is potentially slow
- VMs with instance store backed root volumes not pause-able
- Historically root volume limited to 10-GB max– ***faster imaging…***

31

ELASTIC BLOCK STORE

- EBS provides 1 drive to 1 virtual machine **(1 : 1)** **(not shared)**
- EBS cost model is different than instance storage (uses S3)
  - ~10¢ per GB/month for General Purpose Storage (GP2)
  - ~8¢ per GB/month for General Purpose Storage (GP3)
  - 30GB of free tier storage space
- EBS provides "live" mountable volumes
  - Listed under volumes
  - **Data volumes**: can be mounted/unmounted to any VM, dynamically at any time
  - **Root volumes**: hosts OS files and acts as a boot device for VM
  - In Linux drives are linked to a mount point "directory"
- Snapshots back up EBS volume data to S3
  - Enables replication (required for horizontal scaling)
  - EBS volumes not actively used should be snapshotted, and deleted to save EBS costs…

32

EBS VOLUME TYPES - 2

- Metric: I/O Operations per Second (IOPS)
- **General Purpose 2 (GP2)**
  - 3 IOPS per GB, min 100 IOPS (<34GB), max of 16,000 IOPS
  - 250MB/sec throughput per volume
- **General Purpose 3** (GP3 – new Dec 2020)
  - Max 16,000 IOPS, Default 3,000 IOPS
  - GP2 requires creating a 1TB volume to obtain 3,000 IOPS
  - GP3 all volumes start at 3000 IOPS and 125 MB/s throughput
  - 1000 additional IOPS beyond 3000 is $5/month up to 16000 IOPS
  - 125 MB/s additional throughput is $5/month up to 1000 MB/s throughput

EBS VOLUME TYPES - 3

- **Provisioned IOPS (IO1)**
  - Legacy, associated with GP2
  - Allows user to create custom disk volumes where they pay for a specified IOPS and throughput
  - 32,000 IOPS, and 500 MB/sec throughput per volume MAX
- **Throughput Optimized HDD (ST1)**
  - Up to 500 MB/sec throughput
  - 4.5 ¢ per GB/month
- **Cold HDD (SC1)**
  - Up to 250 MB/sec throughput
  - 2.5 ¢ per GB/month
- **Magnetic**
  - Up to 90 MB/sec throughput per volume
  - 5 ¢ per GB/month

ELASTIC FILE SYSTEM (EFS)

- EFS provides 1 volume to many client **(1 : n) shared storage**
- Network file system (based on NFSv4 protocol)
- Shared file system for EC2, Fargate/ECS, Lambda
- Enables mounting (sharing) the same disk "volume" for R/W access across multiple instances at the same time
- Different performance and limitations vs. EBS/Instance store

- Implementation uses abstracted EC2 instances
- ~ 30 ¢ per GB/month storage – ***default burstable throughput***
- **Throughput modes:**
- Can modify modes only once every 24 hours

- Burstable Throughput Model:
  - Baseline – 50kb/sec per GB
  - Burst – 100MB/sec pet GB (for volumes sized 10GB to 1024 GB)
  - Credits - .72 minutes/day per GB

35

ELASTIC FILE SYSTEM (EFS) - 2

Information subject to revision

- Burstable Throughput Rates
  - Throughput rates: baseline vs burst
  - Credit model for bursting: maximum burst per day

File System Size (GiB)	Baseline Aggregate Throughput (MiB/s)	Burst Aggregate Throughput (MiB/s)	Maximum Burst Duration (Min/Day)	% of Time File System Can Burst (Per Day)
10	0.5	100	7.2	0.5%
256	12.5	100	180	12.5%
512	25.0	100	360	25.0%
1024	50.0	100	720	50.0%
1536	75.0	150	720	50.0%
2048	100.0	200	720	50.0%
3072	150.0	300	720	50.0%
4096	200.0	400	720	50.0%

36

ELASTIC FILE SYSTEM (EFS) - 3

Information subject to revision

- **Throughput Models**
- Provisioned Throughput Model
- For applications with:
  high performance requirements, but low storage requirements
- Get high levels of performance w/o overprovisioning capacity
- $6 MB/s-Month (Virginia Region)
  - Default is 50kb/sec for 1 GB, .05 MB/s = 30 ¢ per GB/month
- If file system metered size has higher baseline rate based on size, file system follows default Amazon EFS Bursting Throughput model
  - No charges for Provisioned Throughput below file system's entitlement in Bursting Throughput mode
  - Throughput entitlement = 50kb/sec per GB

37

ELASTIC FILE SYSTEM (EFS) - 4

Information subject to revision

Performance Comparison, Amazon EFS and Amazon EBS

	Amazon EFS	Amazon EBS Provisioned IOPS
Per-operation latency	Low, consistent latency.	Lowest, consistent latency.
Throughput scale	10+ GB per second.	Up to 2 GB per second.

Storage Characteristics Comparison, Amazon EFS and Amazon EBS

	Amazon EFS	Amazon EBS Provisioned IOPS
Availability and durability	Data is stored redundantly across multiple AZs.	Data is stored redundantly in a single AZ.
Access	Up to thousands of Amazon EC2 instances, from multiple AZs, can connect concurrently to a file system.	A single Amazon EC2 instance in a single AZ can connect to a file system.
Use cases	Big data and analytics, media processing workflows, content management, web serving, and home directories.	Boot volumes, transactional and NoSQL databases, data warehousing, and ETL.

38

EC2

- EC2 Spot Instance Advisor:
- https://aws.amazon.com/ec2/spot/instance-advisor/
- Provides sortable list of ec2 instance types with interruption (termination) frequencies
- Helps you choose an instance type that is less likely to be terminated
- Best practices for using spot instances:
- https://docs.aws.amazon.com/whitepapers/latest/cost-optimization-leveraging-ec2-spot-instances/spot-best-practices.html

39

EC2 - 2

- ***On Amazon EC2, what is a "metal" instance?***
- A bare metal server is not shared with anyone
- There is no virtualization hypervisor *(program the contextualizes and hosts virtual machines)*
- The operating system is installed directly on the root disk and the machine is booted directly like a laptop or desktop computer
- The user can install any operating system and make configurations changes to the machine's base operating system
- The user can then install and control a virtualization hypervisor on bare metal servers
- Bare metal servers were offered on AWS starting in ~2017

40

AMAZON MACHINE IMAGES

- AMIs
- Unique for the operating system (root device image)
- Two types
  - Instance store
  - Elastic block store (EBS)
- Deleting requires multiple steps
  - Deregister AMI
  - Delete associated data - (*files in S3*)
- Forgetting both steps leads to costly "orphaned" data
  - No way to instantiate a VM from deregistered AMIs
  - Data still in S3 resulting in charges

41

EC2 VIRTUALIZATION HYPERVISORS

- 1st, 2nd, 3rd, 4th generation → XEN-based
- 5th+ generation instances → AWS Nitro virtualization

- XEN (pre-nitro) - two virtualization modes:
- XEN **Para-virtualization (PV)** "paravirtual"
  - 10GB Amazon Machine Image – base image size limit
  - Addressed poor performance of old XEN HVM mode
  - I/O performed using special XEN kernel with XEN paravirtual mode optimizations for better performance
  - Requires OS to have an available paravirtual kernel
  - PV VMs: use common **AKI** files on AWS – ***Amazon kernel image(s)***

42

EC2 VIRTUALIZATION - HVM

- XEN HVM (hardware virtualization) mode:
  - HVM just meant that virtualization used CPU extensions (Intel VT-x or AMD-V) to run an unmodified OS directly on the VM
    - Today HVM usually implies there is more than just native-CPU virtualization support, but also I/O-device support, etc.
  - Full virtualization – no special OS kernel required (no AKI)
  - Computer entirely simulated (… *i.e. slow*)
  - MS Windows pre-nitro ran in "HVM" mode
  - Allows work around: 10GB instance store root volume limit
  - Kernel is on the root volume (under /boot)
  - No AKIs (kernel images)
  - Commonly used today (***EBS-backed instances***)

43

XEN PV VS HVM ON EC2

Feature	Paravirtual (PV)	Hardware Virtual Machine (HVM)
Guest OS Modification	Requires the guest OS to be modified (aware of the hypervisor).	Allows the use of unmodified guest OS, as if running on bare metal hardware.
Boot Method	Boots with a special boot loader called PV-GRUB, which then chain loads the kernel.	Boots by executing the master boot record of the root block device, like a physical machine.
Hardware Access	Relies on a patched kernel and specialized drivers (hypercalls) for I/O operations and direct communication with the hypervisor.	Emulates a full set of hardware and takes advantage of hardware virtualization extensions (Intel VT-x, AMD-V) for fast access to the underlying physical hardware.
Performance (Historically)	Historically offered better performance than early HVM instances due to the use of direct drivers for network/storage I/O.	Early HVM had more overhead due to hardware emulation.
Performance (Modern Legacy)	Performance is limited and lacks modern enhancements.	With the availability of PV drivers for HVM AMIs and hardware advancements, HVM instances provide better or superior performance for most workloads.
Feature Support	Does not support modern features like enhanced networking or GPU processing.	Required for advanced features such as enhanced networking, GPU instances, and specific instance types (e.g., i2 instances).

44

EC2 VIRTUALIZATION - NITRO

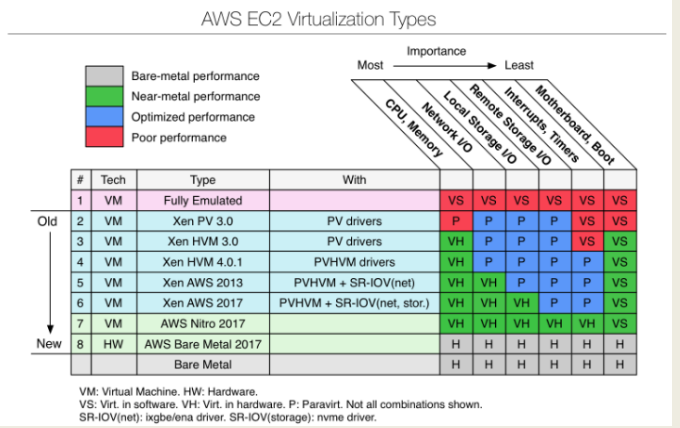
- Nitro based on Kernel-based-virtual-machines
  - Stripped down version of Linux KVM hypervisor
  - Uses KVM core kernel module
  - I/O access has a direct path to the device
- Goal: provide indistinguishable performance from bare metal
- There are 5 Nitro versions (v2 to v6)
- Article describes features and provides links to instance families (m, c, r, etc.) where it is possible to check the Nitro version for specific instance types to understand feature evolution
- https://docs.aws.amazon.com/ec2/latest/instancetypes/ec2-nitro-instances.html

45

EVOLUTION OF AWS VIRTUALIZATION

- From: http://www.brendangregg.com/blog/2017-11-29/aws-ec2-virtualization-2017.html

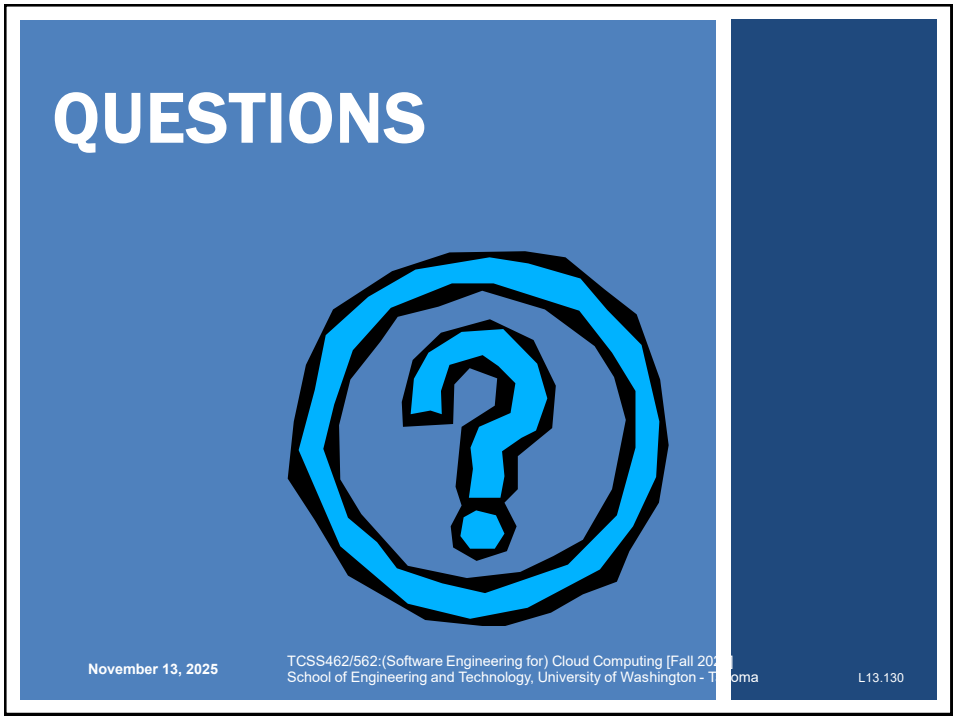
AWS EC2 Virtualization Types



	#	Tech	Type	With						
	1	VM	Fully Emulated		VS	VS	VS	VS	VS	VS
Old	2	VM	Xen PV 3.0	PV drivers	P	P	P	P	VS	VS
	3	VM	Xen HVM 3.0	PV drivers	VH	P	P	P	VS	VS
	4	VM	Xen HVM 4.0.1	PVHVM drivers	VH	P	P	P	P	VS
	5	VM	Xen AWS 2013	PVHVM + SR-IOV(net)	VH	VH	P	P	P	VS
	6	VM	Xen AWS 2017	PVHVM + SR-IOV(net, stor.)	VH	VH	VH	P	P	VS
	7	VM	AWS Nitro 2017		VH	VH	VH	VH	VH	VS
New	8	HW	AWS Bare Metal 2017		H	H	H	H	H	H
			Bare Metal		H	H	H	H	H	H

VM: Virtual Machine. HW: Hardware.
VS: Virt. in software. VH: Virt. in hardware. P: Paravirt. Not all combinations shown.
SR-IOV(net): ixgbe/ena driver. SR-IOV(storage): nvme driver.

46


QUESTIONS
November 13, 2025
TCSS462/562:(Software Engineering for) Cloud Computing [Fall 202
School of Engineering and Technology, University of Washington - T oma
L13.130

130