TCSS 562: SOFTWARE ENGINEERING FOR CLOUD COMPUTING

AWS Overview and Demo II, 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 ***:**
  - 12:30 pm to 1:30 pm – ONLINE via Zoom
- Or email for appointment

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

2

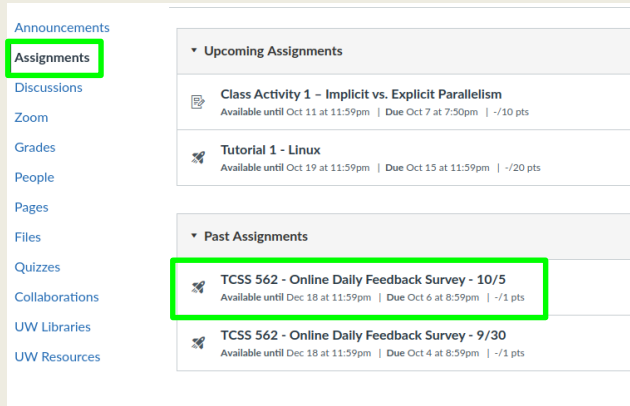
OBJECTIVES – 11/4

- **Questions from 10/30**
- Tutorials Questions
- Tutorial 6 – Serverless Databases
- AWS Overview and demo
- Class Activity 3 / Tutorial 4 Demo
- Ch. 5: Cloud Enabling Technology

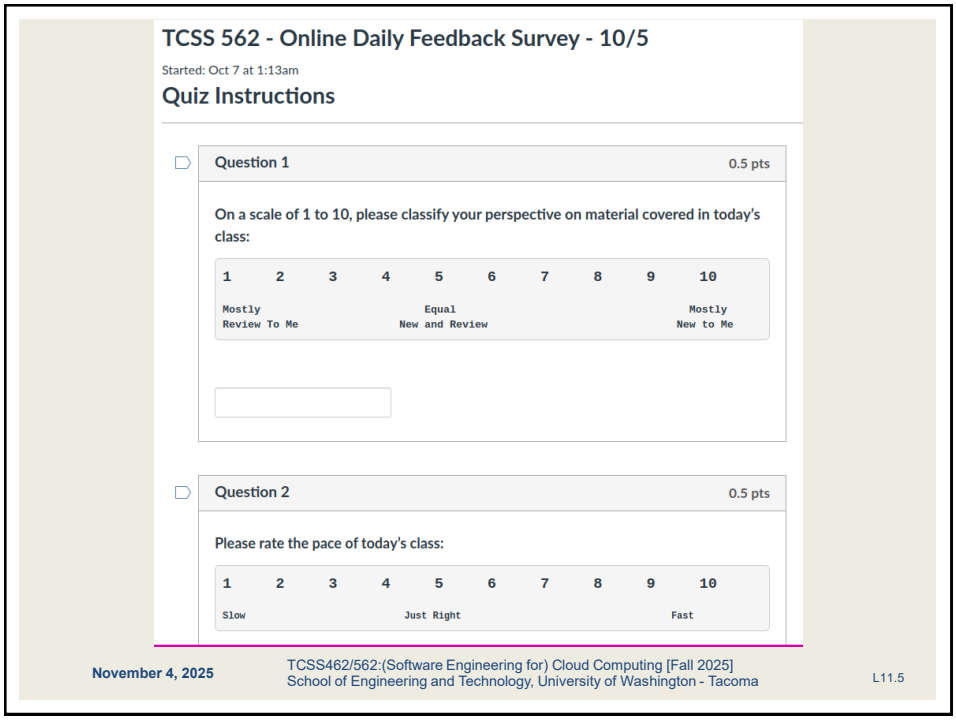
3

ONLINE DAILY FEEDBACK SURVEY

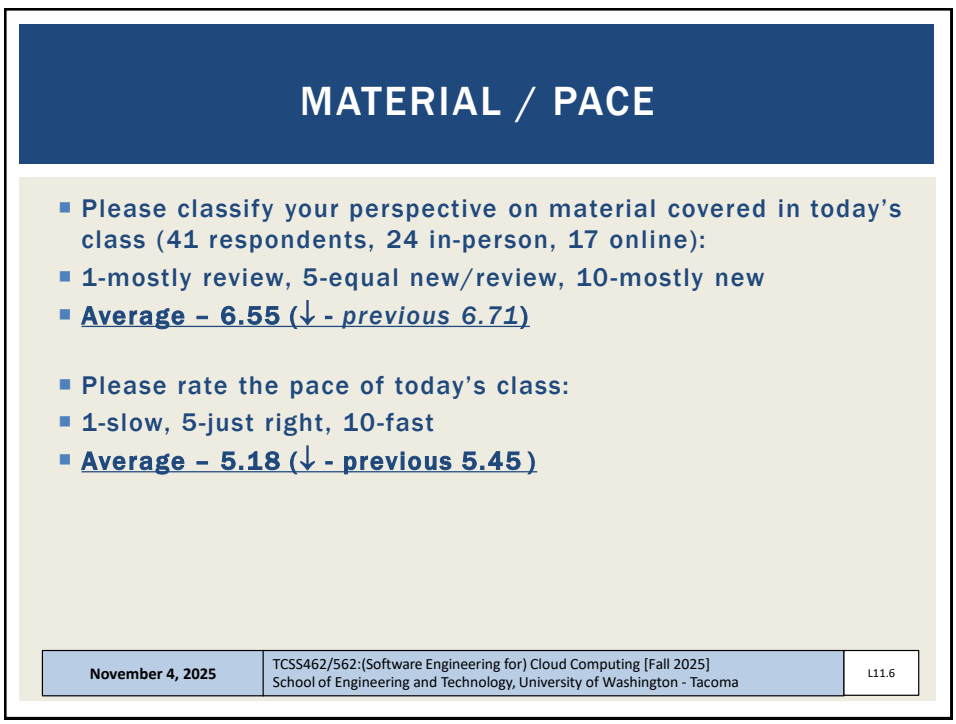
- Daily Feedback Quiz in Canvas – Take After Each Class
- Extra Credit for completing



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
November 4, 2025
TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025]
School of Engineering and Technology, University of Washington - Tacoma
L11.5

5


MATERIAL / PACE
Please classify your perspective on material covered in today's class (41 respondents, 24 in-person, 17 online):
1-mostly review, 5-equal new/review, 10-mostly new
Average – 6.55 (↓ - previous 6.71)
Please rate the pace of today's class:
1-slow, 5-just right, 10-fast
Average – 5.18 (↓ - previous 5.45)
November 4, 2025
TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2025]
School of Engineering and Technology, University of Washington - Tacoma
L11.6

6

FEEDBACK FROM 10/30

- ***For Tutorial 4, is it possible to implement the encode and decode functions in the same Lambda function and specify the operation in the parameter passed to it ?***
- It is fine to just create a single Java package (jar) that has both function handlers
- But for deployment, having two Lambda functions is required so the encode and decode operations are fully isolated from each other
  - If you pass in the raw string to encode, then it would be too easy to cache it and simply return it
  - Therefore, two distinct AWS Lambda endpoints are required for Tutorial 4…

7

PRACTICE - CALCULATE BREAK EVEN POINT: AWS-LAMBDA = EC2

- **At how many "compute" days will AWS Lambda processing costs equal the EC2 hosting cost ?**

- Assume a hypothetical microservice that runs for 1 second
- The function is called repeatedly and sequentially
- 1 endpoint is hosted with EC2, the other with AWS Lambda
- Requirements: ~4 vCPUs, 7 GB RAM
- EC2 instance: m5n.xlarge, on demand cost $0.238/hour
- AWS Lambda: $0.00001667 GB/sec
- Ignore the additional cost of AWS Lambda function calls
- Ignore the AWS Lambda Free Tier (400,000 GB/sec per month)

8

SOLUTION

- **Lambda cost:**
- Cost per second: 7 GB x $.00001667 = $.00011669
- **How much does one month cost on AWS Lambda @7GB?**
- 1 month = 30.416667 days x 24 hours/days = 730 hours/month
- 730 hours * 60 minutes * 60 seconds = 2,628,000 sec/month
- AWS Lambda = $306.66/month (about $43.81/GB/month)
- EC2 COST = 730 hours x $0.238 /hr = $173.74/month
- **How much AWS Lambda compute can you buy for $173.74 ?**
- Seconds = 1,488,902
- Minutes (/60) = 24,815
- Hours (/60) = 413.584
- Days (/24) = 17.23 ← Point when EC2 becomes cheaper than Lambda for continuous compute

9

FEEDBACK - 2

- ***How is the quiz going to be structured?***
  Are we allowed to bring notes?
- Thursday November 6 @ 4:40pm in BHS 106
  - Moving rooms for quiz for more time
  - The room is vacant after 5:40p and the professor will stay late
- The quiz will be delivered using paper (not Canvas)
- Unlimited notes and books permitted
- No digital devices (ebook, laptop, smartphone) or Internet
- Sample questions in lectures 9, 10, 11

10

QUIZ 1 COVERAGE

- Review content from Tutorials 1, 2, 3, 4
- In class we only covered a subset of the slides each day:
  - Lecture 1: L1.1 – L1.46
  - Lecture 2: L2.1 – L2.50
  - Lecture 3: L3.1 – L3.34, Class Activity 1
  - Lecture 4: L4.1 – L4.35
  - Lecture 5: L5.1 – L5.56
  - Lecture 6: L6.1 – L6.49
  - Lecture 7: L7.1 – L7.43
  - Lecture 8: L8.1 – L8.38 , Class Activity 2 (*covers Tutorial 3*)
  - Lecture 9: L9.1 – L9.72
  - Lecture 10: L10.1 – L10.19, L9.72-9.85 , Class Activity 3 (*covers Tutorial 4*)

TERM PROJECT PROPOSALS

- 14 Total term project proposals received
- *Project proposals are under review*
- If asked for minor revisions, can simply address item(s) and submit a brief update
- Status & Feedback to be provided:
  - proposals accepted, or
  - revisions requested

SAMPLE QUESTION 1

- When an AWS Lambda function is scaled from 512 MB memory to 10 GB, what resources are scaled accordingly?

A. CPU timeshare, disk I/O throughput (iops), network I/O throughput (iops)
B. Number of concurrent threads
C. CPU timeshare
D. CPU timeshare, disk I/O throughput (iops)
E. CPU timeshare, function concurrency

13

SAMPLE QUESTION 2

- In tutorial 4, a Plain Old Java Object (POJO) is used inside of HelloPojo.java for what purpose?

A. To reduce overhead (time) incurred from transferring data using a HashMap.
B. To prevent camel case typographical errors from interfering with data transfer.
C. To allow any tag / attribute pair to be transferred seamlessly to the Lambda function handler.
D. To provide a class where a user can implement checks to verify if data is valid.
E. To reduce overhead (size) incurred from transferring data using a HashMap.

14

OBJECTIVES – 11/4

- Questions from 10/30
- **Tutorials Questions**
- Tutorial 6 - Serverless Databases
- AWS Overview and demo
- Class Activity 3 / Tutorial 4 Demo
- Ch. 5: Cloud Enabling Technology

15

TUTORIAL 3 – NOW CLOSED

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

16

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

TUTORIAL 5 – DUE NOV 14

- 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

OBJECTIVES – 11/4

- Questions from 10/30
- Tutorials Questions
- **Tutorial 6 - Serverless Databases**
- AWS Overview and demo
- Class Activity 3 / Tutorial 4 Demo
- Ch. 5: Cloud Enabling Technology

19

CATCH UP FROM LECTURE 9

- Switch to Lecture 9 slides
- L9.87 to L9.97

20

WE WILL RETURN AT 4:55 PM

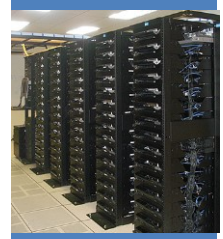
21

OBJECTIVES – 11/4

- Questions from 10/30
- Tutorials Questions
- Tutorial 6 - Serverless Databases
- **AWS Overview and demo**
- Class Activity 3 / Tutorial 4 Demo
- Ch. 5: Cloud Enabling Technology

22

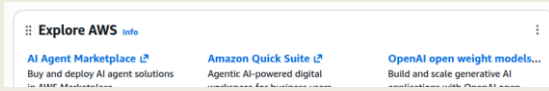
AWS OVERVIEW AND DEMO



23

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

24

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

25

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

26

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.

27

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

28

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…***

29

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…

30

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

33

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%

34

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

35

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.

36

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

37

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

38

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

39

EC2 VIRTUALIZATION - PARAVIRTUAL

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

- XEN - two virtualization modes
- XEN Paravirtualization "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: will use common **AKI** files on AWS – ***Amazon kernel Image(s)***
    - *Look for common identifiers*

40

EC2 VIRTUALIZATION - HVM

- XEN HVM mode
  - Full virtualization – no special OS kernel required
  - Computer entirely simulated
  - MS Windows runs 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*)

41

EC2 VIRTUALIZATION - NITRO

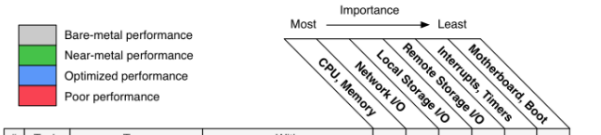
- Nitro based on Kernel-based-virtual-machines
  - Stripped down version of Linux KVM hypervisor
  - Uses KVM core kernel module
  - I/O access has a direct path to the device
- Goal: provide indistinguishable performance from bare metal

42

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.

43

INSTANCE ACTIONS

- Stop
  - Costs of "pausing" an instance
- Terminate
- Reboot

- Image management
- Creating an image
  - EBS (snapshot)
- Bundle image
  - Instance-store

44

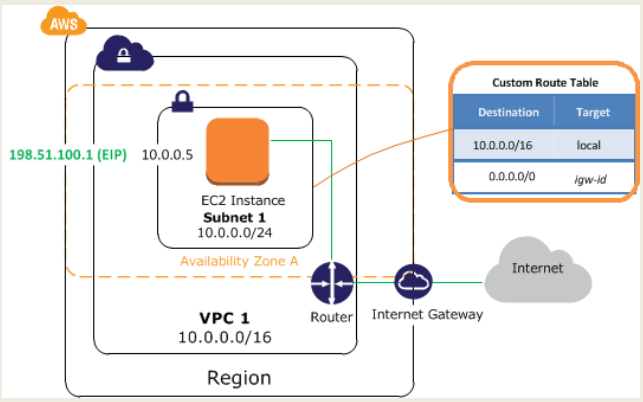
EC2 INSTANCE: NETWORK ACCESS

- Public IP address
- Elastic IPs
  - Costs: in-use FREE, not in-use ~12 ¢/day
  - Not in-use (e.g. "paused" EBS-backed instances)
- Security groups
  - E.g. firewall
- Identity access management (IAM)
  - AWS accounts, groups
- VPC / Subnet / Internet Gateway / Router
- NAT-Gateway

45

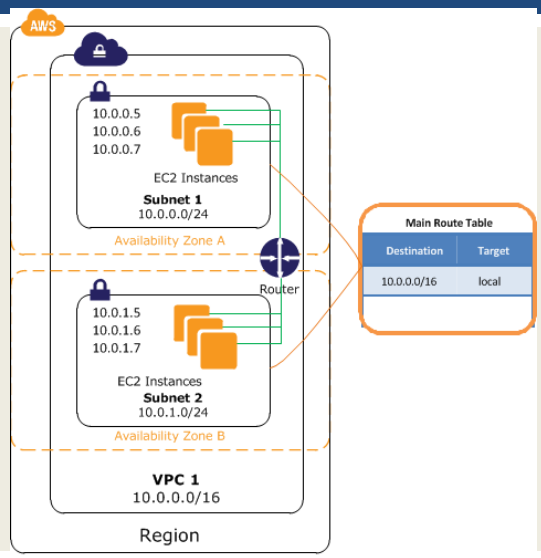
SIMPLE VPC

- Recommended when using Amazon EC2



46

VPC SPANNING AVAILABILITY ZONES



47

INSPECTING INSTANCE INFORMATION

- EC2 VMs run a local metadata service
- Can query instance metadata to self discover cloud config attributes
- **Version 2 (default) of the metadata service requires a token**
- Get Token:

```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api /token" -H
"X-aws-ec2-metadata-token-ttl-seconds: 21600"`
```

- Find your instance ID:

```
curl -H "X-aws-ec2-metadata-token: $TOKEN“ http://169.254.169.254/

curl -H "X-aws-ec2-metadata-token: $TOKEN“
http://169.254.169.254/latest/

curl -H "X-aws-ec2-metadata-token: $TOKEN“
http://169.254.169.254/latest/meta-data/

curl -H "X-aws-ec2-metadata-token: $TOKEN“
http://169.254.169.254/latest/meta-data/instance-id ; echo
```

See: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/configuring-instance-metadata-service.html#instance-metadata-retrieval-examples

48

SIMPLE STORAGE SERVICE (S3)

- Key-value blob storage
- What is the difference vs. key-value stores (NoSQL DB)?
- Can mount an S3 bucket as a volume in Linux
  - Supports common file-system operations
- Provides eventual consistency
- Can store Lambda function state for life of container.

49

AWS CLI

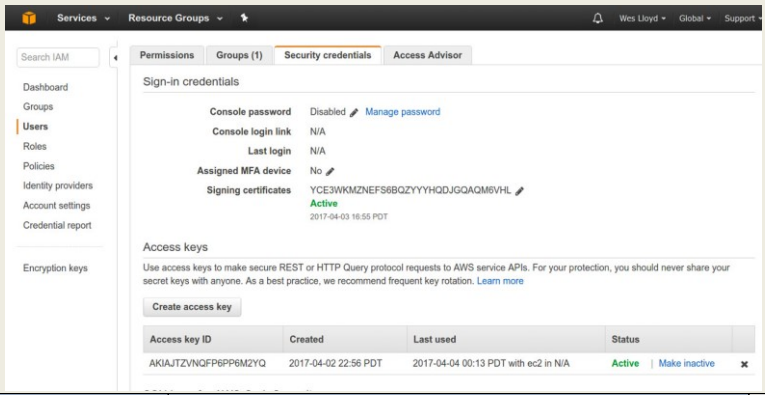
- Launch Ubuntu 16.04 VM
  - Instances | Launch Instance
- Install the general AWS CLI
  - sudo apt install awscli
- Create config file

```
[default]
aws_access_key_id = <access key id>
aws_secret_access_key = <secret access key>
region = us-east-1
```


50

AWS CLI - 2

- **Creating access keys:** IAM | Users | Security Credentials | Access Keys | Create Access Keys



AWS CLI - 3

- Export the config file
  - Add to /home/ubuntu/.bashrc

```
export AWS_CONFIG_FILE=$HOME/.aws/config
```

- Try some commands:
  - aws help
  - aws command help
  - aws ec2 help
  - aws ec2 describes-instances --output text
  - aws ec2 describe-instances --output json
  - aws s3 ls
  - aws s3 ls vmscaleruw

LEGACY / SERVICE SPECIFIC CLI(S)

- `sudo apt install ec2-api-tools`
- Provides more concise output
- Additional functionality
- Define variables in .bashrc or another sourced script:
- `export AWS_ACCESS_KEY={your access key}`
- `export AWS_SECRET_KEY={your secret key}`
- `ec2-describe-instances`
- `ec2-run-instances`
- `ec2-request-spot-instances`
- EC2 management from Java:
- http://docs.aws.amazon.com/AWSJavaSDK/latest/javadoc/index.html
- Some AWS services have separate CLI installable by package

AMI TOOLS

- Amazon Machine Images tools
- For working with disk volumes
- Can create live copies of any disk volume
  - Your local laptop, ec2 root volume (EBS), ec2 ephemeral disk
- Installation:
  https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ami-tools-commands.html
- AMI tools reference:
- https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ami-tools-commands.html
- Some functions may require private key & certificate files

PRIVATE KEY AND CERTIFICATE FILE

- Install openssl package on VM

```
# generate private key file
$openssl genrsa 2048 > mykey.pk

# generate signing certificate file
$openssl req -new -x509 -nodes -sha256 -days 36500 -key
mykey.pk -outform PEM -out signing.cert
```

- Add signing.cert to IAM | Users | Security Credentials | - - *new signing certificate* - -
- From: http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/set-up-ami-tools.html?icmpid=docs_iam_console#ami-tools-create-certificate

PRIVATE KEY, CERTIFICATE FILE

- These files, combined with your AWS_ACCESS_KEY and AWS_SECRET_KEY and AWS_ACCOUNT_ID enable you to publish new images from the CLI
- Objective:

1. Configure VM with software stack
2. Burn new image for VM replication **(horizontal scaling)**

- An alternative to bundling volumes and storing in S3 is to use a containerization tool such as Docker. . .
- Create image script . . .

SCRIPT: CREATE A NEW INSTANCE STORE IMAGE FROM LIVE DISK VOLUME

```
image=$1
echo "Burn image $image"
echo "$image" > image.id
mkdir /mnt/tmp
AWS_KEY_DIR=/home/ubuntu/.aws
export EC2_URL=http://ec2.amazonaws.com
export S3_URL=https://s3.amazonaws.com
export EC2_PRIVATE_KEY=${AWS_KEY_DIR}/mykey.pk
export EC2_CERT=${AWS_KEY_DIR}/signing.cert
export AWS_USER_ID={your account id}
export AWS_ACCESS_KEY={your aws access key}
export AWS_SECRET_KEY={your aws secret key}
ec2-bundle-vol -s 5000 -u ${AWS_USER_ID} -c ${EC2_CERT} -k ${EC2_PRIVATE_KEY}
--ec2cert /etc/ec2/amitools/cert-ec2.pem --no-inherit -r x86_64 -p $image -i
/etc/ec2/amitools/cert-ec2.pem
cd /tmp
ec2-upload-bundle -b tcss562 -m $image.manifest.xml -a ${AWS_ACCESS_KEY} -s
${AWS_SECRET_KEY}  --url http://s3.amazonaws.com --location US
ec2-register tcss562/$image.manifest.xml --region us-east-1 --kernel aki-
88aa75e1
```


57

MAKE A DISK FROM AN IMAGE FILE

```
# ****************  ON THE LOCAL COMPUTER ******************
# create 1200 MB virtual disk = 1,258,291,200 bytes
sudo dd if=/dev/zero of=vhd.img bs=1M count=1200
# format the disk using the ext4 filesystem
sudo mkfs.ext4 vhd.img
# mount the disk at "/mnt"
sudo mount -t auto -o loop vhd.img /mnt
# check that the disk is mounted
df -h
# create a hello file (or copy data) to the new virtual disk
cd /mnt
sudo echo "hello world !" > hello.txt
ls -l
cd
# unmount the virtual disk
sudo umount /mnt
```


58

COMPRESS IMAGE, PUSH TO S3

```
# compress the disk
bzip2 vhd.img

# push the disk image to S3
aws s3 cp vhd.img.bz2 s3://tcss562-f21-images
```


59

RESTORE ON THE CLOUD

```
# ******************  ON THE AWS EC2 VM ********************
# with the awscli installed and configured

# download the image from S3
aws s3 cp s3://tcss562-f21-images/vhd.img.bz2 vhd.img.bz2

# uncompress the image
bzip2 -d vhd.img.bz2

# we need to calculate the number of sectors for the
partition
# disk sectors are 512 bytes each
# divide the disk size by 512 to determine sectors
# sectors = 1258291200 / 512 = 2459648

# create a disk partition for this disk that is
# 2459648 sectors in size using the ephemeral drive or
# a newly mounted EBS volume that is unformatted

sudo fdisk /dev/nvme1n1
```


60

PARTITION THE DISK

```
Welcome to fdisk (util-linux 2.34).

Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)

Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-97656249, default 2048): 2048
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-97656249, default
97656249): 2459648

Created a new partition 1 of type 'Linux' and of size 1.2 GiB.

Command (m for help): t
Selected partition 1
Hex code (type L to list all codes): 83
Changed type of partition 'Linux' to 'Linux'.

Command (m for help): w  (to write and exit)
```


COPY DATA TO NEW DISK PARTITION

```
# now check if the partition has been created.
# it should be listed as /dev/nvme1n1p1:
ls /dev/nvme1n1*

# now copy the data to the partition
sudo dd if=vhd.img of=/dev/nvme1n1p1

# mount the disk
sudo mount /dev/nvme1n1p1 /mnt

# and check if the hello file is there
cat /mnt/hello.txt

# we were able to copy the disk image to the cloud
# and we never had to format the cloud disk
# this examples copies a filesystem from a local disk
# to the cloud disk
```

FOR MORE INFORMATION

- Example script:
- https://faculty.washington.edu/wlloyd/courses/tcss562/examples/copy-disk-to-cloud.sh

- URLs:
- https://help.ubuntu.com/community/DriveImaging
- https://www.tecmint.com/create-virtual-harddisk-volume-in-linux/

63

COST SAVINGS MEASURES

- ***From Tutorial 3:***
- #1: ALWAYS USE SPOT INSTANCES FOR COURSE/RESEARCH RELATED PROJECTS
- #2: NEVER LEAVE AN EBS VOLUME IN YOUR ACCOUNT THAT IS NOT ATTACHED TO A RUNNING VM
- #3: BE CAREFUL USING PERSISTENT REQUESTS FOR SPOT INSTANCES
- #4: TO SAVE/PERSIST DATA, USE EBS SNAPSHOTS AND THEN
- #5: DELETE EBS VOLUMES FOR TERMINATED EC2 INSTANCES.
- #6: UNUSED SNAPSHOTS AND UNUSED EBS VOLUMES SHOULD BE PROMPTLY DELETED !!
- #7: USE PERSISTENT SPOT REQUESTS AND THE "STOP" FEATURE TO PAUSE VMS DURING SHORT BREAKS

64

OBJECTIVES – 11/4

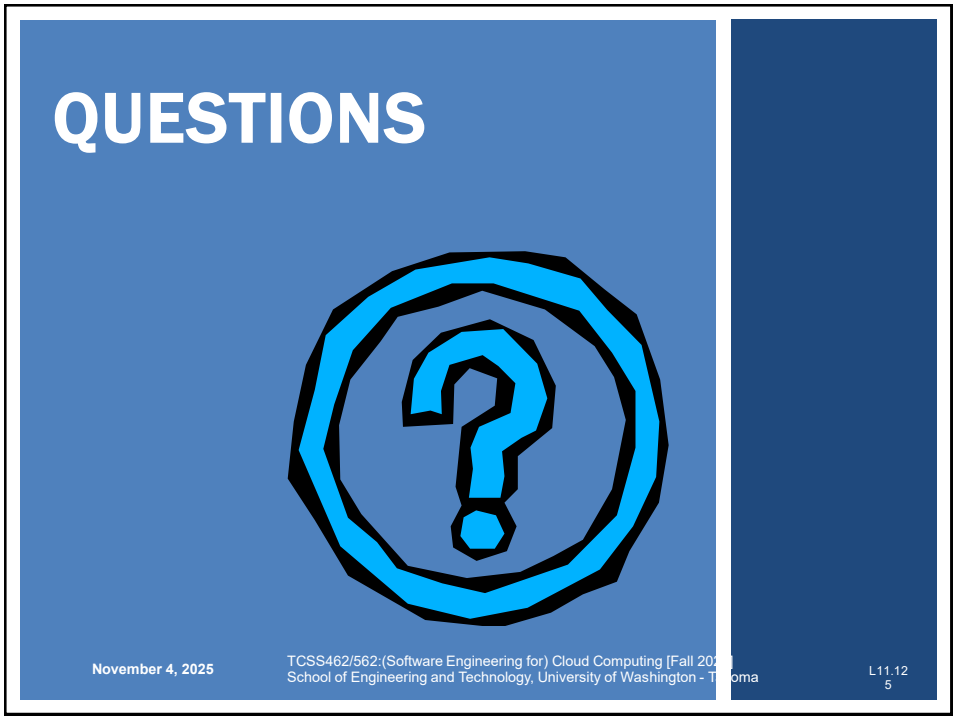
- Questions from 10/30
- Tutorials Questions
- Tutorial 6 - Serverless Databases
- AWS Overview and demo
- **Class Activity 3 / Tutorial 4 Demo**
- Ch. 5: Cloud Enabling Technology

65

OBJECTIVES – 11/4

- Questions from 10/30
- Tutorials Questions
- Tutorial 6 - Serverless Databases
- AWS Overview and demo
- Class Activity 3 / Tutorial 4 Demo
- **Ch. 5: Cloud Enabling Technology**

66


QUESTIONS
November 4, 2025
TCSS462/562:(Software Engineering for) Cloud Computing [Fall 202
School of Engineering and Technology, University of Washington - T oma
L11.12
5

125