TCSS 562: SOFTWARE ENGINEERING FOR CLOUD COMPUTING

Cloud Computing: Concepts and Models – IV & AWS Demo

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

MW 5:50-7:50 PM

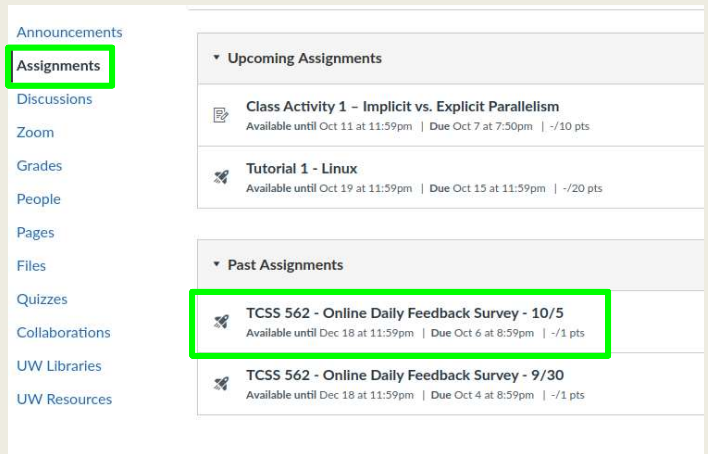
OBJECTIVES – 10/28

- Questions from 10/26
- Quiz 1 –posted on Canvas – available through 10/30
- Class Activity #2 (review)
- From: Cloud Computing Concepts, Technology & Architecture: Cloud Computing Concepts and Models:
  - Cloud deployment models
- AWS overview and demonstration

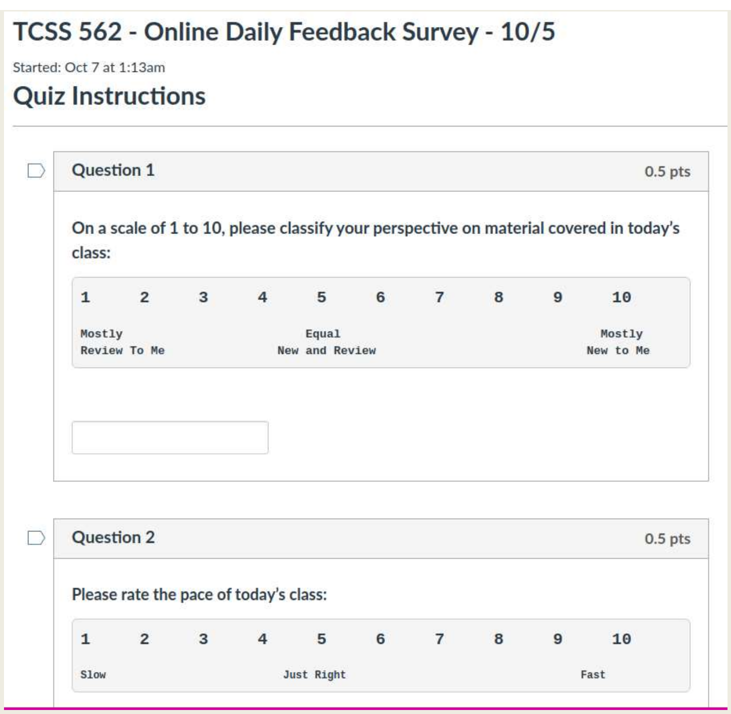
- 2nd hour:
- AWS overview and demonstration
- Tutorial questions
- Team planning

ONLINE DAILY FEEDBACK SURVEY

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



October 28, 2020 | TCSS562: Software Engineering for Cloud Computing [Fall 2020]
School of Engineering and Technology, University of Washington - Tacoma | L9.3



October 28, 2020 | TCSS562: Software Engineering for Cloud Computing [Fall 2020]
School of Engineering and Technology, University of Washington - Tacoma | L9.4

MATERIAL / PACE

- Please classify your perspective on material covered in today's class (21 respondents):
- 1-mostly review, 5-equal new/review, 10-mostly new
- **Average – 6.95 (↑ - *previous 6.36*)**

- Please rate the pace of today's class:
- 1-slow, 5-just right, 10-fast
- **Average – 6.14 (↑ - *previous 5.52*)**

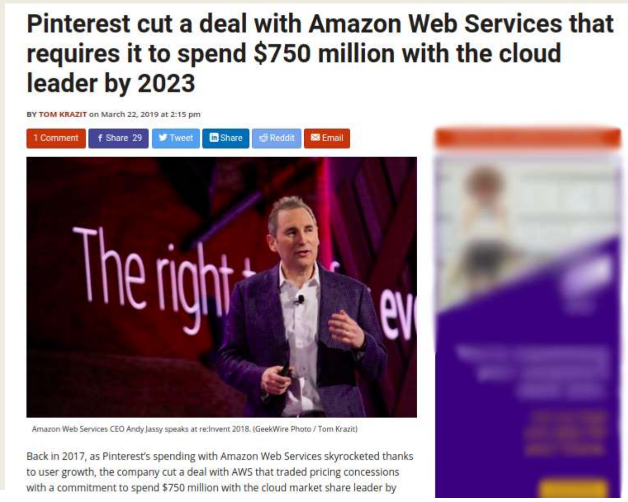
FEEDBACK FROM 10/26

- ***What's the difference between cold and warm on a Function-as-a-Service platform?***

- Example with AWS Lambda, Java, & SAAF

FEEDBACK - 2

- ***Does AWS provide discounts to big organizations (like Netflix) who are using AWS infrastructure ?***



OBJECTIVES – 10/28

- **Questions from 10/26**
- **Quiz 1 –posted on Canvas – available through 10/30**
- **Class Activity #2 (review)**
- **<u>From: Cloud Computing Concepts, Technology & Architecture:</u> Cloud Computing Concepts and Models:**
  - **Cloud deployment models**
- **AWS overview and demonstration**

- **<u>2nd hour:</u>**
- **AWS overview and demonstration**
- **Tutorial questions**
- **Team planning**

October 28, 2020	TCSS562:Software Engineering for Cloud Computing [Fall 2020] School of Engineering and Technology, University of Washington - Tacoma	L9.8

OBJECTIVES – 10/28

- Questions from 10/26
- Quiz 1 –posted on Canvas – available through 10/30
- Class Activity #2 (review)
- From: Cloud Computing Concepts, Technology & Architecture: Cloud Computing Concepts and Models:
  - Cloud deployment models
- AWS overview and demonstration
- 2nd hour:
- AWS overview and demonstration
- Tutorial questions
- Team planning

CLASS ACTIVITY 2

- Table provides average execution time of running a multi-threaded scientific model on a variety of cloud computing resources.

Resource Type	Run time (seconds)	Cost per hour
Lambda 3GB 2 vCPUs	236	11.8¢ (ea run)
r5.large 2 vCPUs	347	12.6¢
m5.xlarge 4 vCPUs	212	19.2¢
m5.8xlarge 32 vCPUs	123	$1.54
c5.18xlarge 72 vCPUs	129	$3.06
m5.24xlarge 96 vCPUs	120	$4.61
z1d.12xlarge 48 vCPUs	126	$4.64

CLASS ACTIVITY 2

Resource Type	Run time (seconds)	Cost per hour
Lambda 3GB 2 vCPUs	236	11.8¢ (ea run)
r5.large 2 vCPUs	347	12.6¢
m5.xlarge 4 vCPUs	212	19.2¢
m5.8xlarge 32 vCPUs	123	$1.54
c5.18xlarge 72 vCPUs	129	$3.06
m5.24xlarge 96 vCPUs	120	$4.61
z1d.12xlarge 48 vCPUs	126	$4.64

- 1.a- Determine which cloud computing resource above will complete 2,500 model runs the FASTEST using the provided average execution times for an individual run from the table. Assume VMs are pre-initialized.
- Lambda: 15.254 runs/compute hr/function_instance
  2500 runs x .118 ea = $295
- r5.large: 10.375 runs/compute hr/VM
  2500/10.375 = 240.97 hrs x .126 = $30.36
- m5.xlarge: 16.981 runs/compute hr/VM
  2500/16.981 = 147.22 hrs x .192 = $28.27
- m5.8xlarge: 29.268 runs/compute hr/VM
  2500/29.268 = 85.42 hrs x 1.54 = $131.54
- c5.18xlarge: 27.907 runs/compute hr/VM
  2500/27.907 = 89.58 hrs x 3.06 = $274.12

CLASS ACTIVITY 2 - 2

Resource Type	Run time (seconds)	Cost per hour
Lambda 3GB 2 vCPUs	236	11.8¢ (ea run)
r5.large 2 vCPUs	347	12.6¢
m5.xlarge 4 vCPUs	212	19.2¢
m5.8xlarge 32 vCPUs	123	$1.54
c5.18xlarge 72 vCPUs	129	$3.06
m5.24xlarge 96 vCPUs	120	$4.61
z1d.12xlarge 48 vCPUs	126	$4.64

- 1.a- Determine which cloud computing resource above will complete 2,500 model runs the FASTEST using the provided average execution times for an individual run from the table. Assume VMs are pre-initialized.
- Lambda: 15.254 runs/compute hr/function_instance
  2500 runs x .118 ea = $295
- m5.24xlarge: 30 runs/hr
  2500/30 = 83.33 hrs x 4.61 = $384.17
- z1d.12xlarge: 28.571 runs/hr
  2500/28.571 = 87.5 hrs x 4.64 = $406

CLASS ACTIVITY 2 - 3

Resource Type	Run time (seconds)	Cost per hour
Lambda 3GB 2 vCPUs	236	11.8¢ (ea run)
r5.large 2 vCPUs	347	12.6¢
m5.xlarge 4 vCPUs	212	19.2¢
m5.8xlarge 32 vCPUs	123	$1.54
c5.18xlarge 72 vCPUs	129	$3.06
m5.24xlarge 96 vCPUs	120	$4.61
z1d.12xlarge 48 vCPUs	126	$4.64

- 1.b- How long will the FASTEST computing resource require to complete 2,500 runs? (in min:secs). Assume VMs are pre-initialized. Also assume infinite horizontal scalability in that you can create as many VMs as needed to complete all of the runs in parallel. Each VM or Lambda function completes a single model run sequentially.
- Lambda: 3min 56sec
  2500 function instances running in parallel
- r5.large: 5 min 47 sec
  2500 VMs running in parallel
- m5.xlarge: 3 min 32 sec
  2500 VMs running in parallel
- m5.8xlarge: 2 min 3 sec
  2500 VMs running in parallel
- c5.18xlarge: 2 min 9 sec
  2500 VMs running in parallel

CLASS ACTIVITY 2 - 4

Resource Type	Run time (seconds)	Cost per hour
Lambda 3GB 2 vCPUs	236	11.8¢ (ea run)
r5.large 2 vCPUs	347	12.6¢
m5.xlarge 4 vCPUs	212	19.2¢
m5.8xlarge 32 vCPUs	123	$1.54
c5.18xlarge 72 vCPUs	129	$3.06
m5.24xlarge 96 vCPUs	120	$4.61
z1d.12xlarge 48 vCPUs	126	$4.64

- 1.b- How long will the FASTEST computing resource require to complete 2,500 runs? (in min:secs). Assume VMs are pre-initialized. Also assume infinite horizontal scalability in that you can create as many VMs as needed to complete all of the runs in parallel. Each VM or Lambda function completes a single model run sequentially.
- Lambda: 3min 56sec
  2500 function instances running in parallel
- m5.24xlarge: 2 min 0 sec
  2500 VMs running in parallel
- z1d.12xlarge: 2 min 6 sec
  2500 VMs running in parallel

CLASS ACTIVITY 2 - 5

Resource Type	Run time (seconds)	Cost per hour
Lambda 3GB 2 vCPUs	236	11.8¢ (ea run)
r5.large 2 vCPUs	347	12.6¢
m5.xlarge 4 vCPUs	212	19.2¢
m5.8xlarge 32 vCPUs	123	$1.54
c5.18xlarge 72 vCPUs	129	$3.06
m5.24xlarge 96 vCPUs	120	$4.61
z1d.12xlarge 48 vCPUs	126	$4.64

- 1.c- What is the COST for the resource above offering the FASTEST execution time. Assume that VMs require 5-minutes to initialize before runs can be performed.
- **Lambda**: 15.254 runs/compute hr/function_instance
  2500 runs x .118 ea = $295
- **r5.large**: .1797 hrs x 2500 VMs x $.126
  = $56.61
- **m5.xlarge**: .1422 hrs x 2500 VMs x $.192
  = $68.26
- **m5.8xlarge**: .1175 hrs x 2500 VMs x $1.54
  = $452.38
- **c5.18xlarge**: .1192 hrs x 2500 VMs x $3.06
  = $911.63

CLASS ACTIVITY 2 - 6

Resource Type	Run time (seconds)	Cost per hour
Lambda 3GB 2 vCPUs	236	11.8¢ (ea run)
r5.large 2 vCPUs	347	12.6¢
m5.xlarge 4 vCPUs	212	19.2¢
m5.8xlarge 32 vCPUs	123	$1.54
c5.18xlarge 72 vCPUs	129	$3.06
m5.24xlarge 96 vCPUs	120	$4.61
z1d.12xlarge 48 vCPUs	126	$4.64

- 1.c- What is the COST for the resource above offering the FASTEST execution time. Assume that VMs require 5-minutes to initialize before runs can be performed.
- **Lambda**: 15.254 runs/compute hr/function_instance
  2500 runs x .118 ea = $295
- **m5.8xlarge**: .1167 hrs x 2500 VMs x $4.61
  = $1,344.58
- **c5.18xlarge**: .1183 hrs x 2500 VMs x $4.64
  = $1,372.67

CONCLUSION:

Initialization is expensive at scale (2,500 VMs)

CLASS ACTIVITY 2 - 7

Resource Type	Run time (seconds)	Cost per hour
Lambda 3GB 2 vCPUs	236	11.8¢ (ea run)
r5.large 2 vCPUs	347	12.6¢
m5.xlarge 4 vCPUs	212	19.2¢
m5.8xlarge 32 vCPUs	123	$1.54
c5.18xlarge 72 vCPUs	129	$3.06
m5.24xlarge 96 vCPUs	120	$4.61
z1d.12xlarge 48 vCPUs	126	$4.64

- Assume infinite horizontal scalability in that you can create as many VMs as needed to complete all of the runs in parallel. Assume that VMs require 5-minutes to initialize before any runs can be performed. Note that initialization increases cost and should be minimized.
- 2.a- Determine which cloud computing resource above will complete 2,500 model runs for the LOWEST POSSIBLE COST.
- Can refer to results of 1.c to answer question:
- √ **r5.large**:

CLASS ACTIVITY 2 - 8

Resource Type	Run time (seconds)	Cost per hour
Lambda 3GB 2 vCPUs	236	11.8¢ (ea run)
r5.large 2 vCPUs	347	12.6¢
m5.xlarge 4 vCPUs	212	19.2¢
m5.8xlarge 32 vCPUs	123	$1.54
c5.18xlarge 72 vCPUs	129	$3.06
m5.24xlarge 96 vCPUs	120	$4.61
z1d.12xlarge 48 vCPUs	126	$4.64

- Assume infinite horizontal scalability in that you can create as many VMs as needed to complete all of the runs in parallel. Assume that VMs require 5-minutes to initialize before any runs can be performed. Note that initialization increases cost and should be minimized.
- 2.b- What is the lowest possible cost for performing these runs?
- Can refer to results of 1.c to answer question:

CLASS ACTIVITY 2 - 8

Resource Type	Run time (seconds)	Cost per hour
Lambda 3GB 2 vCPUs	236	11.8¢ (ea run)
r5.large 2 vCPUs	347	12.6¢
m5.xlarge 4 vCPUs	212	19.2¢
m5.8xlarge 32 vCPUs	123	$1.54
c5.18xlarge 72 vCPUs	129	$3.06
m5.24xlarge 96 vCPUs	120	$4.61
z1d.12xlarge 48 vCPUs	126	$4.64

- Assume infinite horizontal scalability in that you can create as many VMs as needed to complete all of the runs in parallel. Assume that VMs require 5-minutes to initialize before any runs can be performed. Note that initialization increases cost and should be minimized.
- 2.b- What is the lowest possible cost for performing these runs?
- Can refer to results of 1.c to answer question:
- √ **r5.large**: .1797 hrs x 2500 VMs x $.126 = **$56.61**
- vs. (z1d.12xlarge) **$1,372.67**

October 28, 2020 | TCSS562: Software Engineering for Cloud Computing [Fall 2020] School of Engineering and Technology, University of Washington - Tacoma | L9.19

CLASS ACTIVITY 2 - 8

Resource Type	Run time (seconds)	Cost per hour
Lambda 3GB 2 vCPUs	236	11.8¢ (ea run)
r5.large 2 vCPUs	347	12.6¢
m5.xlarge 4 vCPUs	212	19.2¢
m5.8xlarge 32 vCPUs	123	$1.54
c5.18xlarge 72 vCPUs	129	$3.06
m5.24xlarge 96 vCPUs	120	$4.61
z1d.12xlarge 48 vCPUs	126	$4.64

- Assume infinite horizontal scalability in that you can create as many VMs as needed to complete all of the runs in parallel. Assume that VMs

CONCLUSION:

Obtaining the last few % performance improvement involves paying a HIGH premium on the cloud…

- Can refer to results of 1.c to answer question:
- √ **r5.large**: .1797 hrs x 2500 VMs x $.126 = **$56.61**
- vs. (z1d.12xlarge) **$1,372.67**

October 28, 2020 | TCSS562: Software Engineering for Cloud Computing [Fall 2020] School of Engineering and Technology, University of Washington - Tacoma | L9.20

CLASS ACTIVITY 2 - 9

Resource Type	Run time (seconds)	Cost per hour
Lambda 3GB 2 vCPUs	236	11.8¢ (ea run)
r5.large 2 vCPUs	347	12.6¢
m5.xlarge 4 vCPUs	212	19.2¢
m5.8xlarge 32 vCPUs	123	$1.54
c5.18xlarge 72 vCPUs	129	$3.06
m5.24xlarge 96 vCPUs	120	$4.61
z1d.12xlarge 48 vCPUs	126	$4.64

- Assume infinite horizontal scalability in that you can create as many VMs as needed to complete all of the runs in parallel. Assume that VMs require 5-minutes to initialize before any runs can be performed. Note that initialization increases cost and should be minimized.
- 2.c- How long will these runs require with the LOWEST COST? (in minutes:seconds)
- Can refer to results of 1.c to answer question:

CLASS ACTIVITY 2 - 9

Resource Type	Run time (seconds)	Cost per hour
Lambda 3GB 2 vCPUs	236	11.8¢ (ea run)
r5.large 2 vCPUs	347	12.6¢
m5.xlarge 4 vCPUs	212	19.2¢
m5.8xlarge 32 vCPUs	123	$1.54
c5.18xlarge 72 vCPUs	129	$3.06
m5.24xlarge 96 vCPUs	120	$4.61
z1d.12xlarge 48 vCPUs	126	$4.64

- Assume infinite horizontal scalability in that you can create as many VMs as needed to complete all of the runs in parallel. Assume that VMs require 5-minutes to initialize before any runs can be performed. Note that initialization increases cost and should be minimized.
- 2.c- How long will these runs require with the LOWEST COST? (in minutes:seconds)
- Can refer to results of 1.c to answer question:
- √ **r5.large**: .1797 hrs = **10 min 47 sec**
- *m5.24xlarge (fastest) = 7 min 0 sec*

CLOUD COMPUTING: CONCEPTS AND MODELS

October 28, 2020 TCSS562: Software Engineering for Cloud Computing [Fall 2020]
School of Engineering and Technology, University of Washington - Tacoma L9.23

OBJECTIVES – 10/28

- Questions from 10/26
- Quiz 1 –posted on Canvas – available through 10/30
- Class Activity #2 (review)
- From: Cloud Computing Concepts, Technology & Architecture:
  Cloud Computing Concepts and Models:
  - Cloud deployment models
- AWS overview and demonstration

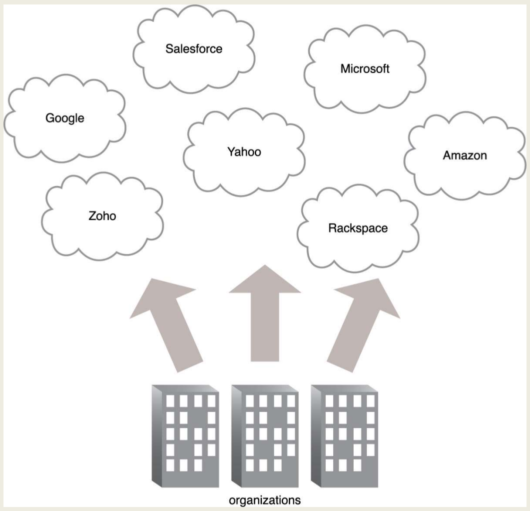
- 2nd hour:
- AWS overview and demonstration
- Tutorial questions
- Team planning

October 28, 2020	TCSS562:Software Engineering for Cloud Computing [Fall 2020] School of Engineering and Technology, University of Washington - Tacoma	L9.24

CLOUD DEPLOYMENT MODELS

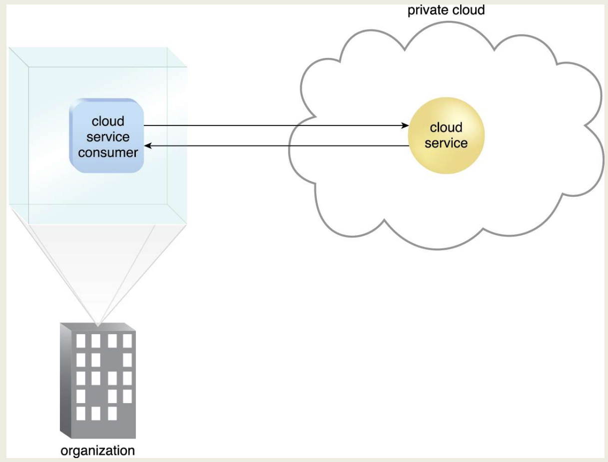
- Distinguished by ownership, size, access
- Common models
  - Public cloud
  - Private cloud
  - Hybrid cloud
  - Community cloud
  - Federated cloud

PUBLIC CLOUDS



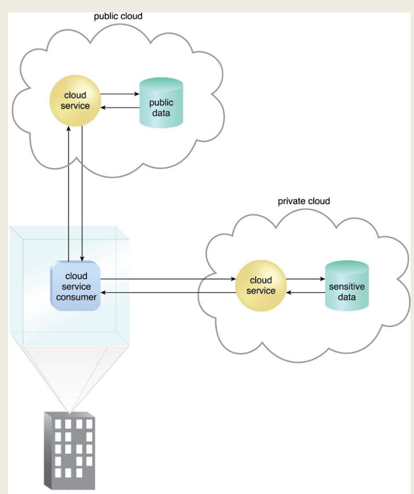
PRIVATE CLOUD

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



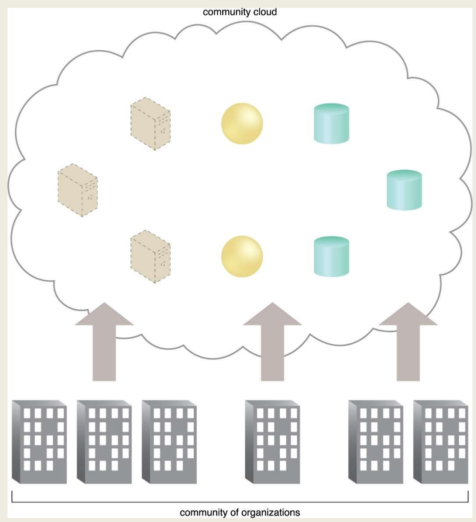
HYBRID CLOUD

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



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



October 28, 2020	TCSS562: Software Engineering for Cloud Computing [Fall 2020] School of Engineering and Technology, University of Washington - Tacoma	L9.29

OTHER CLOUDS

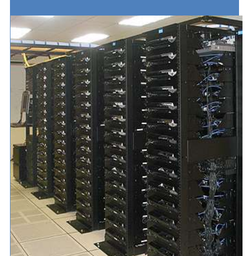
- Federated cloud
  - Simply means to aggregate two or more clouds together
  - Hybrid is typically private-public
  - Federated can be public-public, private-private, etc.
  - Also called inter-cloud
- Virtual private cloud (VPC) on AWS
  - Google and Microsoft simply call these “virtual networks”
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

October 28, 2020	TCSS562: Software Engineering for Cloud Computing [Fall 2020] School of Engineering and Technology, University of Washington - Tacoma	L9.30

OBJECTIVES – 10/28

- Questions from 10/26
- Quiz 1 –posted on Canvas – available through 10/30
- Class Activity #2 (review)
- From: Cloud Computing Concepts, Technology & Architecture: Cloud Computing Concepts and Models:
  - Cloud deployment models
- AWS overview and demonstration
- 2nd hour:
- AWS overview and demonstration
- Tutorial questions
- Team planning

AWS DEMO



CLOUD 101 WORKSHOP

- From the eScience Institute @ UW Seattle:
- https://escience.washington.edu/
- Offers 1-day cloud workshops
- Introduction to AWS, Azure, and Google Cloud
- Task: Deploying a Python DJANGO web application
- Self-guided workshop materials available online:
- https://cloudmaven.github.io/documentation/rc_cloud101_immersion.html
- AWS Educate provides access to many online tutorials / learning resources

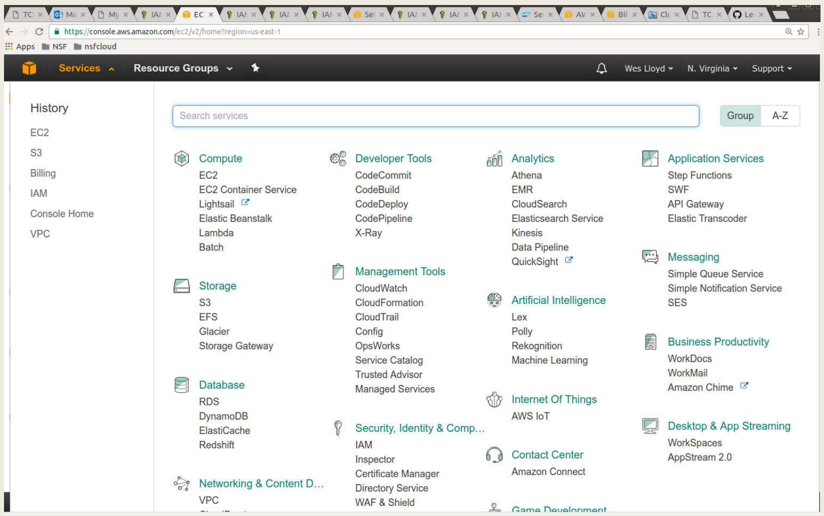
October 23, 2019	TCSS562: Software Engineering for Cloud Computing [Fall 2020] School of Engineering and Technology, University of Washington - Tacoma	L9.33

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

October 28, 2020 TCSS562: Software Engineering for Cloud Computing [Fall 2020]
School of Engineering and Technology, University of Washington - Tacoma L9.34

AWS MANAGEMENT CONSOLE



AWS EC2

- Elastic Compute Cloud
- Instance types: https://ec2instances.info
  - On demand instance – full price
  - Reserved instance – contract based
  - Spot instance – auction based, terminates with 2 minute warning
  - Dedicated/reserved host – reserved HW
  - Reserved host
  - Instance families:
    General, compute-optimized, memory-optimized, GPU, etc.
- Storage types
  - Instance storage - ephemeral storage
  - EBS - Elastic block store
  - EFS - Elastic file system

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
- Mutli-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…***

ELASTIC BLOCK STORE

- EBS cost model is different than instance storage (uses S3)
  - ~10₵ per GB/month
  - 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…

EBS VOLUME TYPES - 2

- Metric: I/O Operations per Second (IOPS)
- General Purpose 2 (GP2)
  - 3 IOPS per GB, Max 10,000 IOPS, 160MB/sec per volume
- Provisioned IOPS (IO1)
  - 32,000 IOPS, and 500 MB/sec throughput per volume
- 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 800 MB/sec throughput
  - 5 ₵ per GB/month

ELASTIC FILE SYSTEM (EFS)

- Network file system (based on NFSv4 protocol)
- Shared file system for EC2 instances
- 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

ELASTIC FILE SYSTEM (EFS) - 2

- 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%

ELASTIC FILE SYSTEM (EFS) - 3

- <u>Throughput Models</u>
- 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

ELASTIC FILE SYSTEM (EFS) - 4

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.

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

EC2 VIRTUALIZATION - PARAVIRTUAL

- 1^{st}, 2^{nd}, 3^{rd}, 4^{th} generation → XEN-based
- 5^{th} 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*

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*)

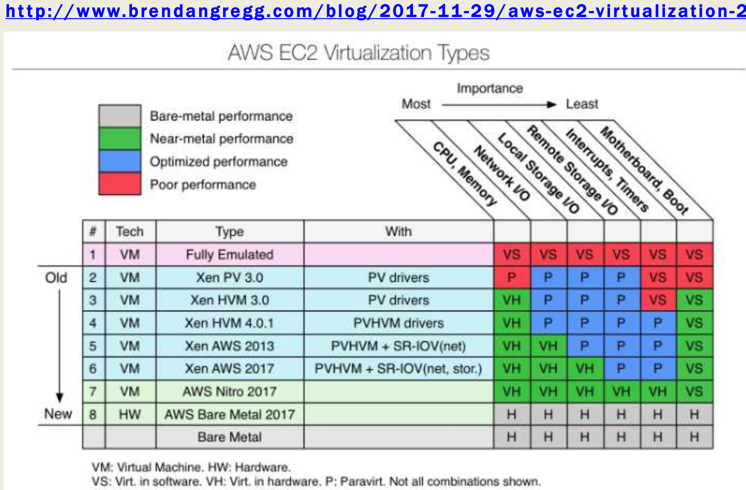
EC2 VIRTUALIZATION - NITRO

- 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

October 23, 2019	TCSS562: Software Engineering for Cloud Computing [Fall 2020] School of Engineering and Technology, University of Washington - Tacoma	L9.47

EVOLUTION OF AWS VIRTUALIZATION

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


AWS EC2 Virtualization Types

Bare-metal performance
Near-metal performance
Optimized performance
Poor performance

Importance: Most → Least

	#	Tech	Type	With	CPU, Memory	Network I/O	Local Storage I/O	Remote Storage I/O	Interrupts, Timers	Motherboard, Boot
	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.

October 23, 2019	TCSS562: Software Engineering for Cloud Computing [Fall 2020] School of Engineering and Technology, University of Washington - Tacoma	L9.48

INSTANCE ACTIONS

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

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

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)

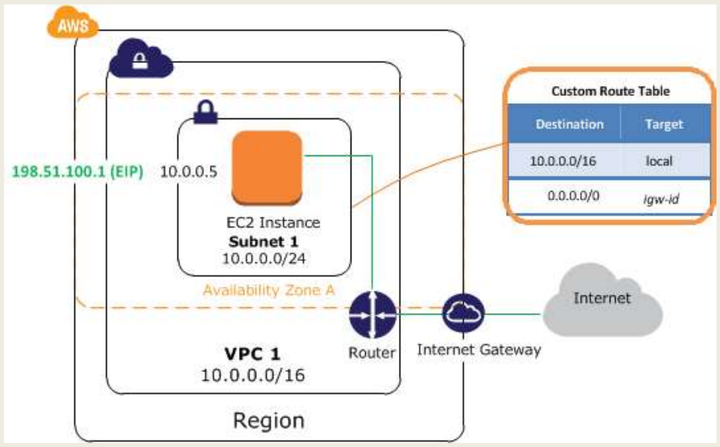
- Security groups
  - E.g. firewall

- Identity access management (IAM)
  - AWS accounts, groups

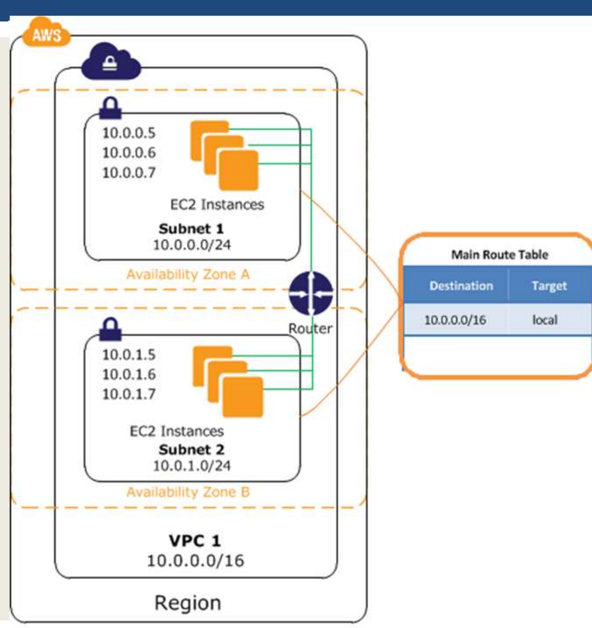
- VPC / Subnet / Internet Gateway / Router
- NAT-Gateway

SIMPLE VPC

- Recommended when using Amazon EC2



VPC SPANNING AVAILABILITY ZONES



INSPECTING INSTANCE INFORMATION

- EC2 VMs run a local metadata service
- Can query instance metadata to self discover cloud configuration attributes

- Find your instance ID:

```
curl http://169.254.169.254/
curl http://169.254.169.254/latest/
curl http://169.254.169.254/latest/meta-data/
curl http://169.254.169.254/latest/meta-data/instance-id
; echo
```

- `ec2-get-info` command
- Python API that provides easy/formatted access to metadata

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.

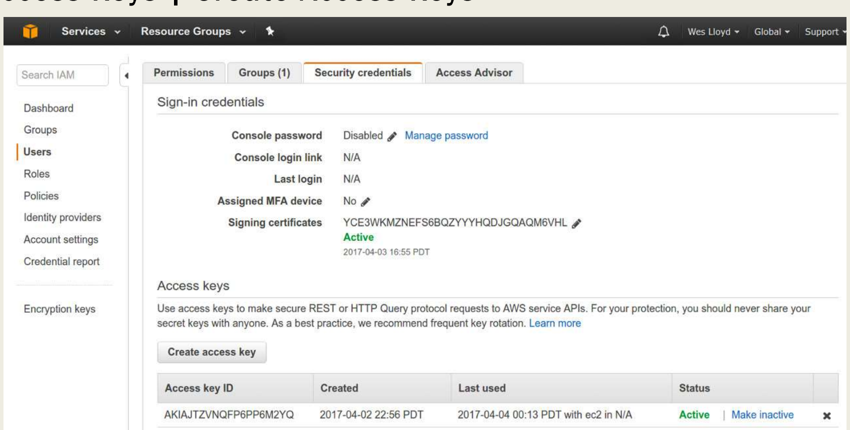
AWS CLI

- 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
```


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
```

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

October 28, 2020	TCSS562: Software Engineering for Cloud Computing [Fall 2020] School of Engineering and Technology, University of Washington - Tacoma	L9.63

WE WILL RETURN AT ~7:03PM

OBJECTIVES – 10/28

- Questions from 10/26
- Quiz 1 –posted on Canvas – available through 10/30
- Class Activity #2 (review)
- From: Cloud Computing Concepts, Technology & Architecture: Cloud Computing Concepts and Models:
  - Cloud deployment models
- AWS overview and demonstration

- 2nd hour:
- AWS overview and demonstration
- Tutorial questions
- Team planning

October 28, 2020	TCSS562:Software Engineering for Cloud Computing [Fall 2020] School of Engineering and Technology, University of Washington - Tacoma	L9.65

OBJECTIVES – 10/28

- Questions from 10/26
- Quiz 1 –posted on Canvas – available through 10/30
- Class Activity #2 (review)
- From: Cloud Computing Concepts, Technology & Architecture: Cloud Computing Concepts and Models:
  - Cloud deployment models
- AWS overview and demonstration

- 2nd hour:
- AWS overview and demonstration
- Tutorial questions
- Team planning

October 28, 2020	TCSS562:Software Engineering for Cloud Computing [Fall 2020] School of Engineering and Technology, University of Washington - Tacoma	L9.66

OBJECTIVES – 10/28

- Questions from 10/26
- Quiz 1 –posted on Canvas – available through 10/30
- Class Activity #2 (review)
- From: Cloud Computing Concepts, Technology & Architecture: Cloud Computing Concepts and Models:
  - Cloud deployment models
- AWS overview and demonstration

- 2nd hour:
- AWS overview and demonstration
- Tutorial questions
- Team planning

QUESTIONS

QUESTIONS

October 28, 2020

TCSS562: Software Engineering for Cloud Computing [Fall 2020]
School of Engineering and Technology, University of Washington - Tacoma

L9.69

TCSS 562
OFFICE HOURS

PLEASE SAY HELLO

L9.70