TCSS 562: SOFTWARE ENGINEERING FOR CLOUD COMPUTING

Cloud Computing Concepts and Models - III

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

1

OFFICE HOURS – FALL 2023

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

> Office Hours set based on Student Demographics survey feedback

2

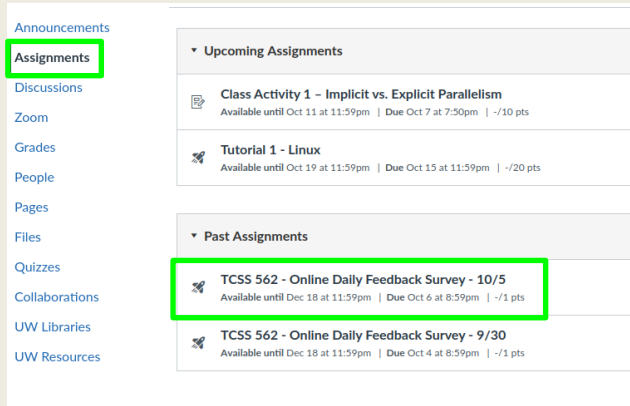
OBJECTIVES – 10/26

- **Questions from 10/24**
- Tutorials Questions
- Tutorial 5 - Files in S3 and CloudWatch Events
- From: Cloud Computing Concepts, Technology & Architecture: Chapter 4: Cloud Computing Concepts and Models:
  - Cloud computing delivery models
  - Cloud deployment models
- AWS Overview and demo
- 2nd hour:
  - Review Activity 2 - Horizontal Scaling in the Cloud
  - Term Project Planning

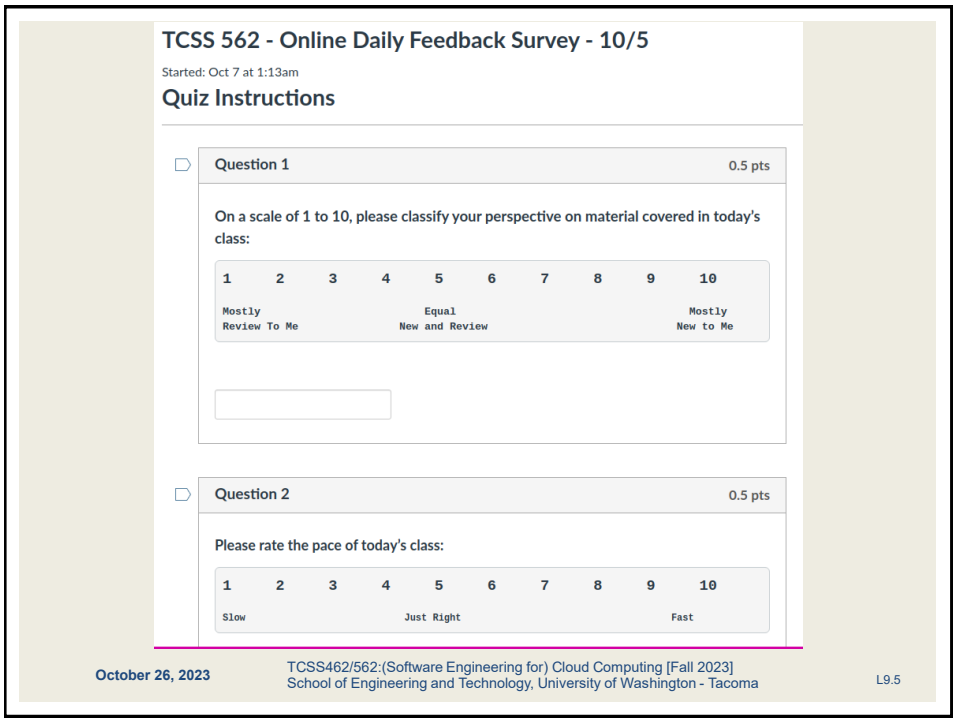
3

ONLINE DAILY FEEDBACK SURVEY

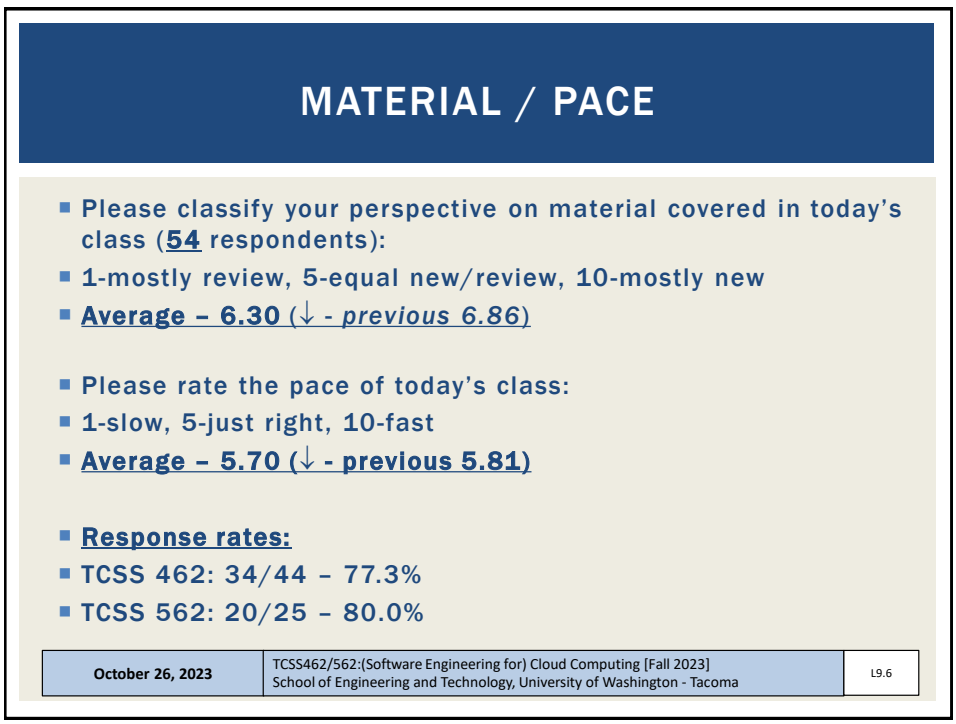
- Daily Feedback Quiz in Canvas – Take After Each Class
- Extra Credit for completing



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
October 26, 2023
TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023]
School of Engineering and Technology, University of Washington - Tacoma
L9.5

5


MATERIAL / PACE
Please classify your perspective on material covered in today's class (54 respondents):
1-mostly review, 5-equal new/review, 10-mostly new
Average – 6.30 (↓ - previous 6.86)
Please rate the pace of today's class:
1-slow, 5-just right, 10-fast
Average – 5.70 (↓ - previous 5.81)
Response rates:
TCSS 462: 34/44 – 77.3%
TCSS 562: 20/25 – 80.0%
October 26, 2023
TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023]
School of Engineering and Technology, University of Washington - Tacoma
L9.6

6

QUESTIONS FROM 10/24

- ***On the AWS Lambda vs. EC2 Cost Example (slide L8.60), the web service continuously uses two threads.***
- ***How does the number of threads impact the total cost?***
- The number of processing threads impacts the choice of hosting environment.
- If there are two continuous threads, then we would ideally have access to:

1. a c5.large ec2 instance with 2 vCPUs
2. an AWS Lambda function with 3GB to guarantee a 100% CPU time share for 2 vCPUs

- **Why shouldn't the webservice be hosted using a t2.micro instance ?**

7

QUESTIONS - 2

- ***Is there an efficient way to find the optimal component composition for an application across VMs?***
- ***While brute force testing to find the best possible application deployment with 4 components (15 configurations) is possible, it is not feasible for an application with 10+ components (115,975 configurations)***
- For explosive search spaces in CS, it is typical to employ heuristics or approximation methods to find near-optimal solutions
- Ideally, we try to prune (reduce) the search space
- **Genetic algorithms** and **reinforcement learning** approaches have recently become popular
  - These approaches randomly search an explosive search space and consider fitness of solutions while searching for nearby solutions that may be better
  - Reinforcement learning can find very good solutions by employing many epochs (search iterations) that are informed by considering the quality of prior results

8

QUESTIONS - 3

- ***When determining VM placement in the cloud, what is the function of placement algorithm(s) ?***
- Placement algorithms distribute VM launch requests across hardware. Commercial cloud providers typically do not disclose details regarding VM placement algorithms / VM placement.
- VM placement algorithms are similar to OS scheduling algorithms
- **Greedy:** place all VMs on the first server until it is full, then move to the next server
- **Round-robin:** place one VM on the first server, then go to the next server, rotate through the full list placing one VM on each node
- **Least-Busy:** based on some characterization of load (i.e. CPU, memory, disk I/O, network I/O) place the VM on the server with the lowest load
- ***What does the Virtual Machine Manager do?***
  - It's he software that orchestrates the cloud (cloud management system)

9

QUESTIONS - 4

- ***I am still uncertain what makes a given service serverless, and another service server-ful***
- For cloud services in AWS, if setting up the service involves associating a VM that is always-on, the service is not serverless
- Examples – these all involve selecting and running a VM:
  - Amazon RDS, hosted relational databases, MySQL, PostgreSQL
  - Amazon MQ (message queue) – Apache MQ, RabbitMQ msg brokers
  - Amazon MSK (Apache Kafka) - stream processing message broker
  - Amazon DocumentDB (MongoDB) – NoSQL document database
- Serverless – No VM selection:
  - Amazon S3 – object storage
  - Dynamo DB – NoSQL DB

10

AWS CLOUD CREDITS UPDATE

- AWS CLOUD CREDITS ARE NOW AVAILABLE FOR TCSS 462/562
- Credits provided on request with expiry of Sept 30, 2024
- Credit codes must be securely exchanged
- Request codes by sending an email with the subject "**AWS CREDIT REQUEST**" to **wlloyd@uw.edu**
- Codes can also be obtained in person (or zoom), in the class, during the breaks, after class, during office hours, by appt
  - 46 credit requests fulfilled as of Oct 24 @ 11:59p
- To track credit code distribution, codes not shared via discord
- 52 students have completed AWS Cloud Credits Survey
  - 17 survey responses missing
- NEXT: instructor will work to create IAM user accounts
  - One IAM user request (unconfirmed) in queue

11

OBJECTIVES – 10/26

- Questions from 10/24
- **Tutorials Questions**
- Tutorial 5 - Files in S3 and CloudWatch Events
- From: Cloud Computing Concepts, Technology & Architecture: Chapter 4: Cloud Computing Concepts and Models:
  - Cloud computing delivery models
  - Cloud deployment models
- AWS Overview and demo
- 2nd hour:
  - Review Activity 2 - Horizontal Scaling in the Cloud
  - Term Project Planning

12

TUTORIAL 0

- Getting Started with AWS
- http://faculty.washington.edu/wlloyd/courses/tcss562/tutorials/TCSS462_562_f2023_tutorial_0.pdf
- Create an AWS account
- Create account credentials for working with the CLI
- Install awsconfig package
- Setup awsconfig for working with the AWS CLI

13

TUTORIAL 3 – DUE OCT 30

- Best Practices for Working with Virtual Machines on Amazon EC2
- http://faculty.washington.edu/wlloyd/courses/tcss562/tutorials/TCSS462_562_f2023_tutorial_3.pdf
- Creating a spot VM
- Creating an image from a running VM
- Persistent spot request
- Stopping (pausing) VMs
- EBS volume types
- Ephemeral disks (local disks)
- Mounting and formatting a disk
- Disk performance testing with Bonnie++
- Cost Saving Best Practices

14

TUTORIAL 4 – DUE NOV 6

- Introduction to AWS Lambda with the Serverless Application Analytics Framework (SAAF)
- https://faculty.washington.edu/wlloyd/courses/tcss562/tutorials/TCSS462_562_f2023_tutorial_4.pdf (link to be posted)
- Obtaining a Java development environment
- Introduction to Maven build files for Java
- Create and Deploy "hello" Java AWS Lambda Function
  - Creation of API Gateway REST endpoint
- Sequential testing of "hello" AWS Lambda Function
  - API Gateway endpoint
  - AWS CLI Function invocation
- Observing SAAF profiling output
- Parallel testing of "hello" AWS Lambda Function with faas_runner
- Performance analysis using faas_runner reports
- Two function pipeline development task

TUTORIAL 5 – TO BE POSTED

- Introduction to Lambda II: Working with Files in S3 and CloudWatch Events
- https://faculty.washington.edu/wlloyd/courses/tcss562/tutorials/TCSS462_562_f2023_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 CloudWatch rule to capture events from CloudTrail
- Have the CloudWatch rule trigger a target 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 – 10/26

- Questions from 10/24
- Tutorials Questions
- **Tutorial 5 - Files in S3 and CloudWatch Events**
- From: Cloud Computing Concepts, Technology & Architecture: Chapter 4: Cloud Computing Concepts and Models:
  - Cloud computing delivery models
  - Cloud deployment models
- AWS Overview and demo
- 2nd hour:
  - Review Activity 2 - Horizontal Scaling in the Cloud
  - Term Project Planning

17

CLOUD COMPUTING: CONCEPTS AND MODELS

18

OBJECTIVES – 10/26

- Questions from 10/24
- Tutorials Questions
- Tutorial 5 - Files in S3 and CloudWatch Events
- From: Cloud Computing Concepts, Technology & Architecture: Chapter 4: Cloud Computing Concepts and Models:
  - Cloud computing delivery models
  - Cloud deployment models
- AWS Overview and demo
- 2nd hour:
  - Review Activity 2 - Horizontal Scaling in the Cloud
  - Term Project Planning

19

CLOUD COMPUTING DELIVERY MODELS

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

Serverless Computing:

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

20

FAAS PRICING

- Break-even point is the point where renting VMs or deploying to a serverless platform (e.g. Lambda) is exactly the same.
- Our example is for one month
- Could also consider one day, one hour, one minute
- **What factors influence the break-even point for an application running on AWS Lambda?**

21

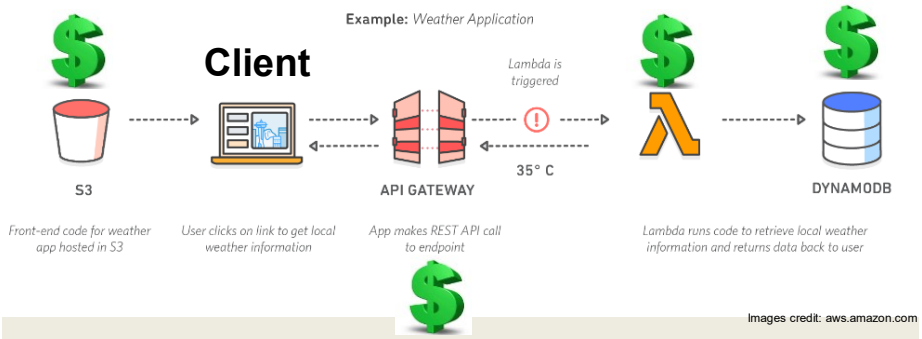
FAAS CHALLENGES

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

22

VENDOR ARCHITECTURAL LOCK-IN

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



- Increased dependencies → increased hosting costs

23

PRICING OBFUSCATION

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

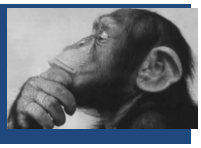
AWS Lambda Pricing

FREE TIER: first 1,000,000 function calls/month → FREE
first 400,000 GB-sec/month → FREE

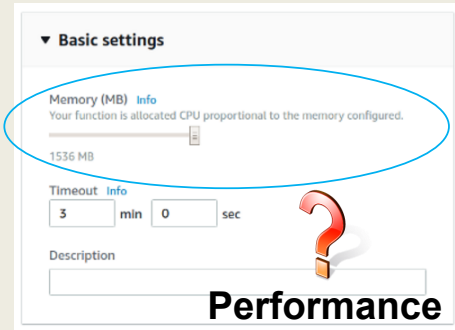
- Afterwards: $0.0000002 per request
  $0.000000208 to rent 128MB / 100-ms

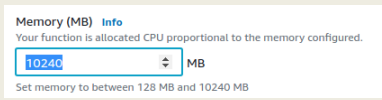
24

MEMORY RESERVATION QUESTION…



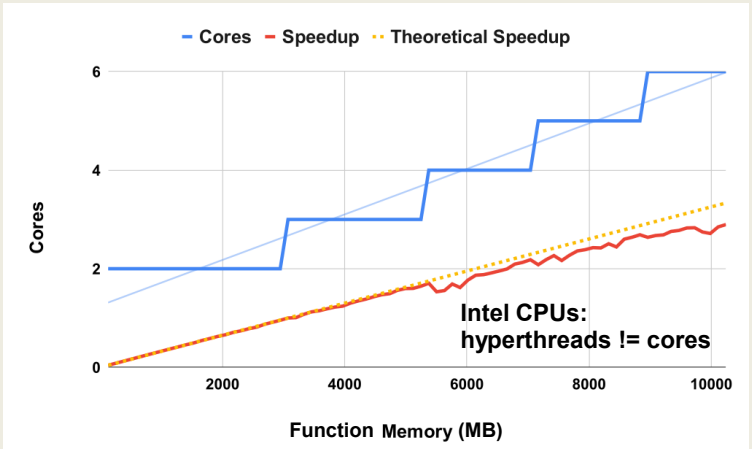
- Lambda memory reserved for functions
- UI provides text box formerly "slider bar" to set function's memory
- Resource capacity (CPU, disk, network) coupled to slider bar:
  "*every **doubling** of memory, **doubles** CPU…*"
- **But how much memory do FaaS functions require?**





25

AWS LAMBDA COUPLES FUNCTION MEMORY TO CPU CORES & TIME SHARE



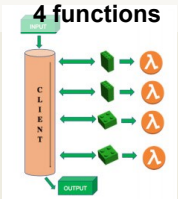
26

SERVICE COMPOSITION

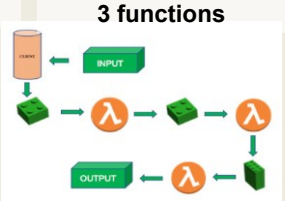
- How should application code be composed for deployment to serverless computing platforms?

Monolithic Deployment

Client flow control, 4 functions



Server flow control, 3 functions



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

Performance

INFRASTRUCTURE FREEZE/THAW CYCLE

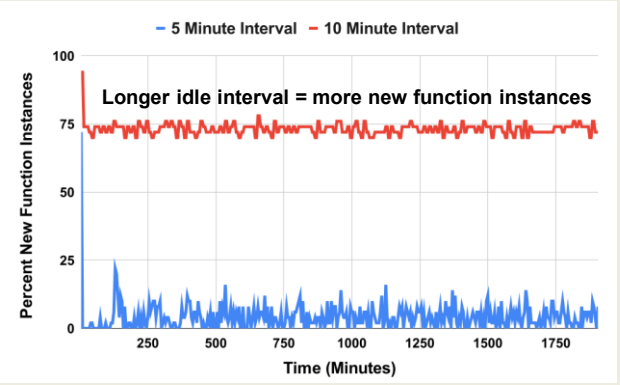
- Unused infrastructure is deprecated
  - *But after how long? (varies by platform)*
- Infrastructure: microVMs (on AWS Lambda), containers on some platforms
- COLD
  - Code image - built/transferred to physical host & cached
- WARM
  - Host has local code cache – create function instance (microVM) on host
- HOT
  - Function instance ready to use

Performance

Image from: Denver7 – The Denver Channel News

AWS LAMBDA – FREEZE/THAW

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



29

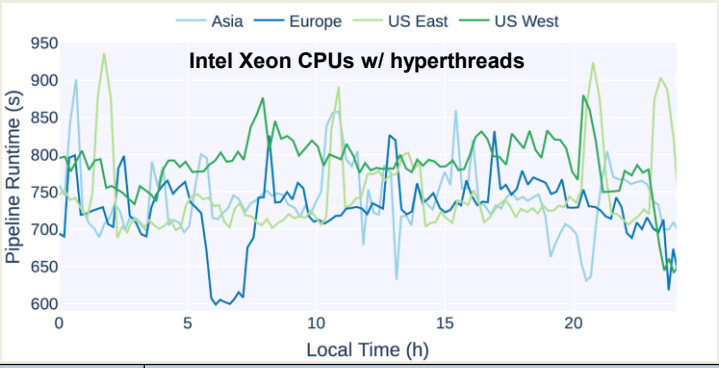
FACTORS IMPACTING PERFORMANCE OF FAAS COMPUTING PLATFORMS

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

30

AWS LAMBDA PERFORMANCE VARIATION

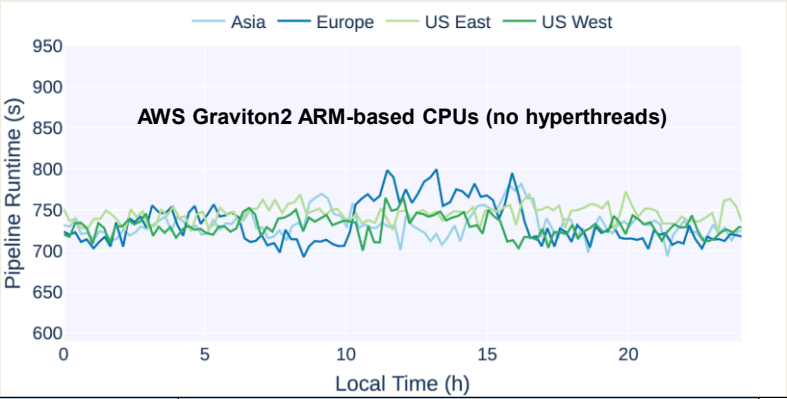
- NLP processing pipeline use case
- Performance variance from: diurnal changes in load (e.g. resource contention), Intel hyperthreading



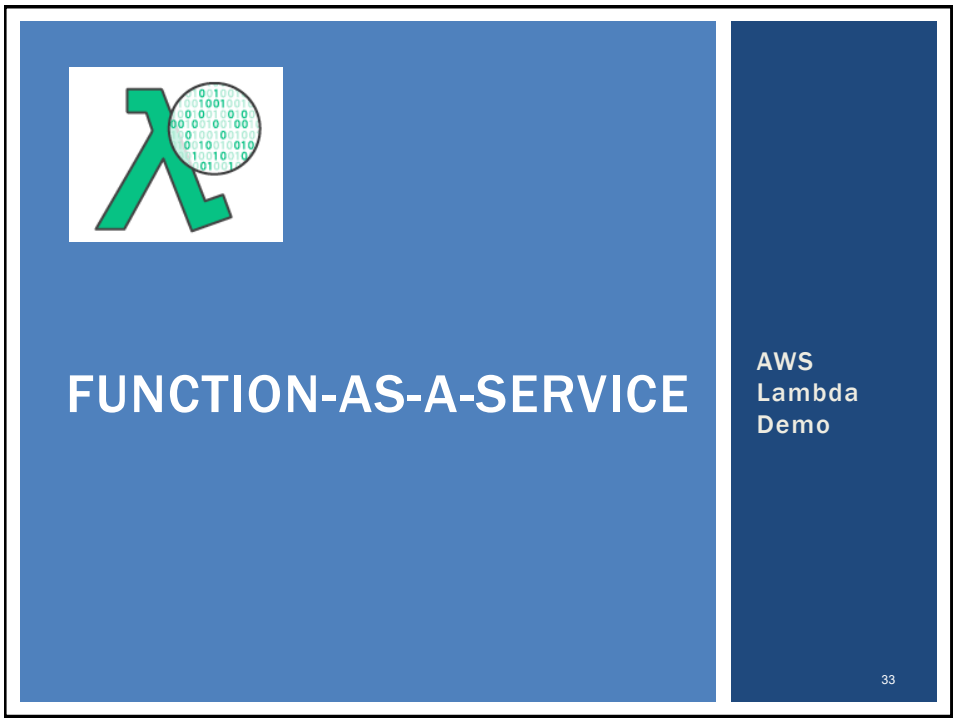
31

AWS LAMBDA PERFORMANCE VARIATION - 2

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



32



33

CLOUD COMPUTING DELIVERY MODELS

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

Serverless Computing:

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

34

CONTAINER-AS-A-SERVICE

- Cloud service model for deploying application containers (e.g. Docker containers) to the cloud
- Deploy containers without worrying about managing infrastructure:
  - Servers
  - Or container orchestration platforms
  - Container platform examples: Kubernetes, Docker swarm, Apache Mesos/Marathon, Amazon Elastic Container Service
  - Container platforms support creation of container clusters on the using cloud hosted VMs
- CaaS Examples:
  - AWS Fargate
  - Google Cloud Run
  - Azure Container Instances

35

CLOUD COMPUTING DELIVERY MODELS

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

Serverless Computing:

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

36

OTHER CLOUD SERVICE MODELS

- IaaS
  - Storage-as-a-Service
- PaaS
  - Integration-as-a-Service
- SaaS
  - Database-as-a-Service
  - Testing-as-a-Service
  - Model-as-a-Service
- ?
  - Security-as-a-Service
  - Integration-as-a-Service

37

OBJECTIVES – 10/26

- Questions from 10/24
- Tutorials Questions
- Tutorial 5 - Files in S3 and CloudWatch Events
- From: Cloud Computing Concepts, Technology & Architecture: Chapter 4: Cloud Computing Concepts and Models:
  - Cloud computing delivery models
  - **Cloud deployment models**
- AWS Overview and demo
- 2nd hour:
  - Review Activity 2 - Horizontal Scaling in the Cloud
  - Term Project Planning

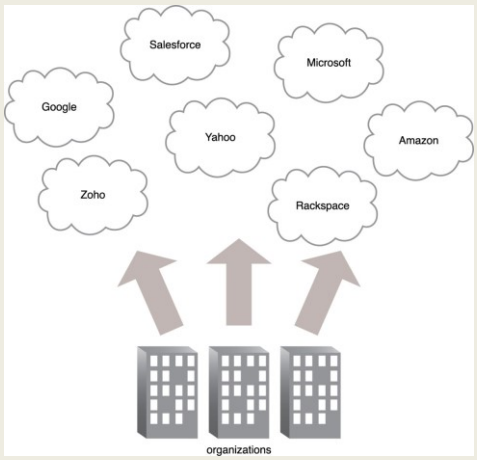
38

CLOUD DEPLOYMENT MODELS

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

39

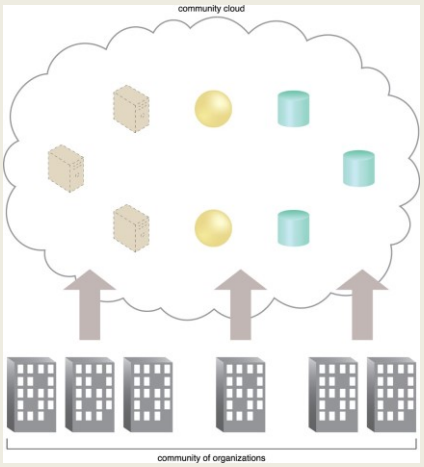
PUBLIC CLOUDS



40

COMMUNITY CLOUD

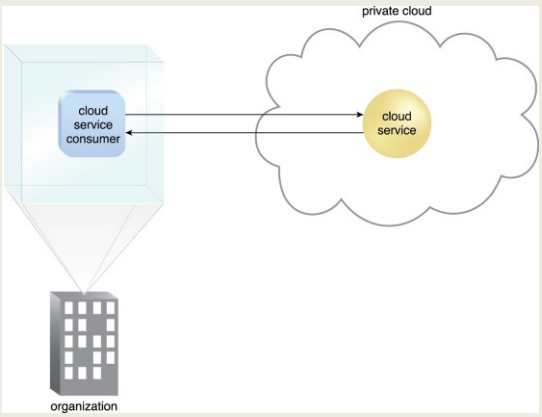
- Specialized cloud built and shared by a particular community
- Leverage economies of scale within a community
- Research oriented clouds
- Examples:
  - Bionimbus - bioinformatics
  - Chameleon
  - CloudLab



41

PRIVATE CLOUD

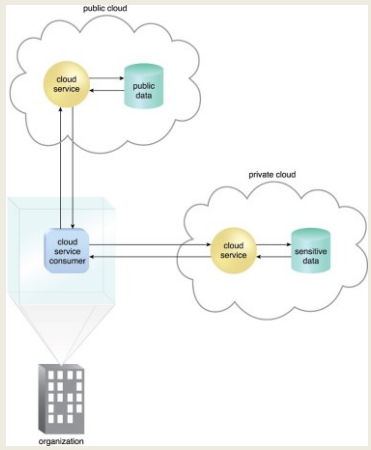
- Compute clusters configured as IaaS cloud
- Open source software
- Eucalyptus
- Openstack
- Apache Cloudstack
- Nimbus
- Virtualization: XEN, KVM, …



42

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



43

OTHER CLOUDS

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

44

WE WILL RETURN AT 4:50 PM

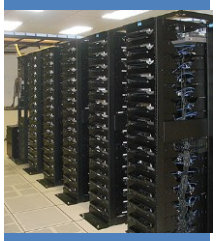
45

OBJECTIVES – 10/26

- Questions from 10/24
- Tutorials Questions
- Tutorial 5 - Files in S3 and CloudWatch Events
- From: Cloud Computing Concepts, Technology & Architecture: Chapter 4: Cloud Computing Concepts and Models:
  - Cloud computing delivery models
  - Cloud deployment models
- **AWS Overview and demo**
- 2nd hour:
  - Review Activity 2 - Horizontal Scaling in the Cloud
  - Term Project Planning

46

AWS OVERVIEW AND DEMO



47

ONLINE CLOUD TUTORIALS

- From the eScience Institute @ UW Seattle: https://escience.washington.edu/
- Online 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/

- AWS Educate provides access to many online tutorials / learning resources:
- https://aws.amazon.com/education/awseducate/

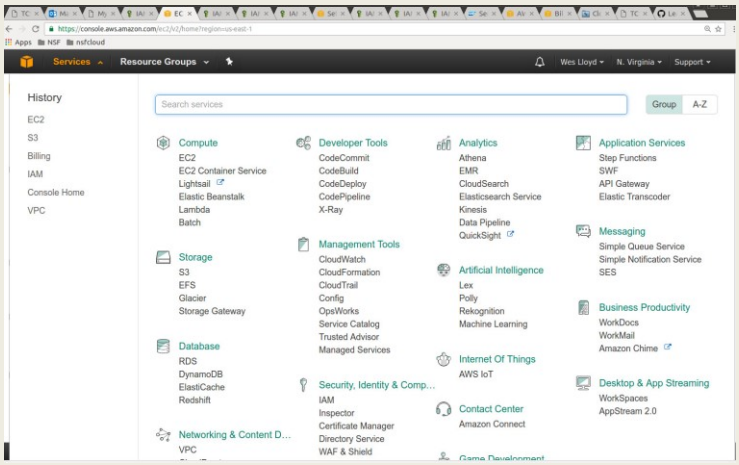
48

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

49

AWS MANAGEMENT CONSOLE



50

AWS EC2

- Elastic Compute Cloud
- Instance types: https://ec2instances.info
  - **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.

51

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

52

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…***

53

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…

54

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

57

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%

58

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

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.

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

61

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*

62

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*)

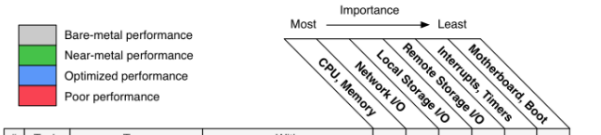
EC2 VIRTUALIZATION - NITRO

- 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

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.

INSTANCE ACTIONS

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

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

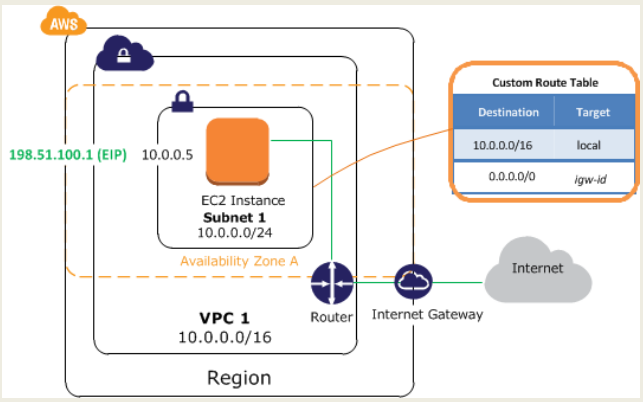
EC2 INSTANCE: NETWORK ACCESS

- 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

67

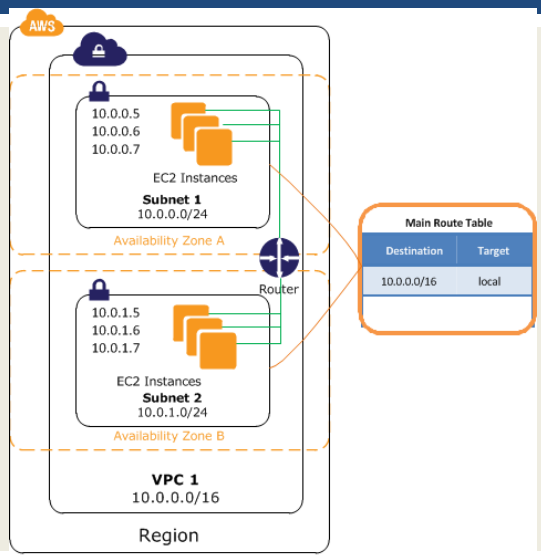
SIMPLE VPC

- Recommended when using Amazon EC2



68

VPC SPANNING AVAILABILITY ZONES



69

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

70

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.

71

AWS CLI

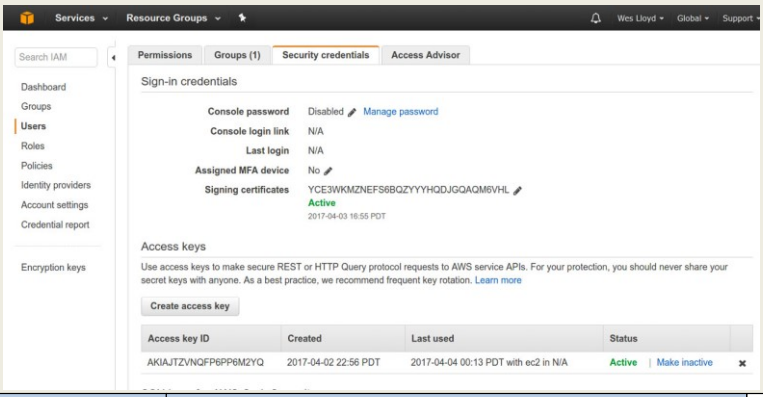
- Launch Ubuntu 16.04 VM
  - Instances | Launch Instance
- Install the general AWS CLI
  - sudo apt install awscli
- Create config file

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


72

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

75

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

76

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
```


79

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

80

OBJECTIVES – 10/26

- Questions from 10/24
- Tutorials Questions
- Tutorial 5 - Files in S3 and CloudWatch Events
- From: Cloud Computing Concepts, Technology & Architecture: Chapter 4: Cloud Computing Concepts and Models:
  - Cloud computing delivery models
  - Cloud deployment models
- AWS Overview and demo
- 2nd hour:
  - **Review Activity 2 - Horizontal Scaling in the Cloud**
  - Term Project Planning

81

OBJECTIVES – 10/26

- Questions from 10/24
- Tutorials Questions
- Tutorial 5 - Files in S3 and CloudWatch Events
- From: Cloud Computing Concepts, Technology & Architecture: Chapter 4: Cloud Computing Concepts and Models:
  - Cloud computing delivery models
  - Cloud deployment models
- AWS Overview and demo
- 2nd hour:
  - Review Activity 2 - Horizontal Scaling in the Cloud
  - **Term Project Planning**

82

TCSS 462/562 TERM PROJECT

83

QUESTIONS

84