

# TCSS 562: SOFTWARE ENGINEERING FOR CLOUD COMPUTING

## AWS Demo, Cloud Enabling Technology

Wes J. Lloyd  
School of Engineering and Technology  
University of Washington – Tacoma  
TR 5:00-7:00 PM



1

## OFFICE HOURS – FALL 2021

- **Tuesdays:**
  - 4:00 to 4:30 pm - CP 229
  - 7:15 to 7:45+ pm – ONLINE via Zoom
- **Thursdays**
  - 4:15 to 4:45 pm – ONLINE via Zoom
  - 7:15 to 7:45+ pm – ONLINE via Zoom
- Or email for appointment
- Zoom Link sent as Canvas Announcement

> Office Hours set based on Student Demographics survey feedback

November 2, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L10.2
------------------	--	-------

2

OBJECTIVES – 11/2

■ Questions from 10/28

■ Tutorial 3 - Best Practices with EC2

■ Tutorial 4 – Intro to FaaS – AWS Lambda

■ Tutorial 5 – Intro to FaaS II – Files in S3, CloudWatch

■ Term Project Proposals – Review by Thursday

■ AWS overview and demonstration

■ 2<sup>nd</sup> hour:

■ AWS overview and demonstration

■ Ch. 5: Cloud Enabling Technology

■ Team planning

November 2, 2021

TCSS562:Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.3

3

ONLINE DAILY FEEDBACK SURVEY

■ Daily Feedback Quiz in Canvas – Take After Each Class

■ Extra Credit for completing

Announcements

Assignments

Discussions

Zoom

Grades

People

Pages

Files

Quizzes

Collaborations

UW Libraries

UW Resources

▼ Upcoming Assignments

Class Activity 1 – Implicit vs. Explicit Parallelism

Available until Oct 11 at 11:59pm | Due Oct 7 at 7:50pm | ~10 pts

Tutorial 1 - Linux

Available until Oct 19 at 11:59pm | Due Oct 15 at 11:59pm | ~20 pts

▼ Past Assignments

TCSS 562 - Online Daily Feedback Survey - 10/5

Available until Dec 18 at 11:59pm | Due Oct 6 at 8:59pm | ~1 pts

TCSS 562 - Online Daily Feedback Survey - 9/30

Available until Dec 18 at 11:59pm | Due Oct 4 at 8:59pm | ~1 pts

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.4

4

Slides by Wes J. Lloyd

L10.2

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:

12345678910

Mostly Review To MeEqual New and ReviewMostly New to Me

Question 2

0.5 pts

Please rate the pace of today's class:

12345678910

SlowJust RightFast

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.5

5

MATERIAL / PACE

Please classify your perspective on material covered in today's class (25 respondents):

1-mostly review, 5-equal new/review, 10-mostly new

Average - 6.52 (↑ - previous 6.29)

Please rate the pace of today's class:

1-slow, 5-just right, 10-fast

Average - 5.36 (↑ - previous 5.29)

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.6

6

FEEDBACK FROM 10/26

- What's the exact difference between containers, VM, and micro VM?
- This is a great question.
- We will discuss this in Chapter 5 and the AWS demo
- In brief:
- **Virtual machines** came first, to simulate a full computer instance
- **Containers** evolved as a lighter-weight alternative
  - **OS containers** – direct replacement for VMs, runs many processes
  - **Application containers** – this is **Docker**, encapsulate all dependencies to run one application into a reusable image
  - Key difference: many containers share the same OS kernel instance with each other
  - Because of kernel sharing, they offer less isolation and security than a virtual machine
  - In many cases containers on the cloud run on top of VMs for this reason

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.7

7

FEEDBACK - 2

- MicroVMs offer a trade-off between:  
containers (fully shared kernel) and VMs (full OS instance)
- MicroVMs run atop of a traditional VM hypervisor (KVM/XEN) but offer a much smaller footprint (low memory <5 MB, very few devices (~5), runs in user space)
- MicroVMs are specifically designed for FaaS functions and CaaS containers to more easily share the underlying host
- Can launch many more MicroVMs per server than Full VMs

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.8

8

## FEEDBACK - 2

- Would like to learn more about implementation, walkthroughs as opposed to slides if possible :)
  - AWS Lambda Demo (tutorial 4)
  - AWS Discussion (Demo)

November 2, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L10.9
------------------	--	-------

9

## OBJECTIVES - 11/2

- Questions from 10/28
- **Tutorial 3 - Best Practices with EC2**
  - Tutorial 4 - Intro to FaaS - AWS Lambda
  - Tutorial 5 - Intro to FaaS II - Files in S3, CloudWatch
  - Term Project Proposals - Review by Thursday
  - AWS overview and demonstration
- 2<sup>nd</sup> hour:
  - AWS overview and demonstration
  - Ch. 5: Cloud Enabling Technology
  - Team planning

November 2, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L10.10
------------------	--	--------

10

OBJECTIVES – 11/2

- Questions from 10/28
- Tutorial 3 - Best Practices with EC2
- **Tutorial 4 – Intro to FaaS – AWS Lambda**
- Tutorial 5 – Intro to FaaS II – Files in S3, CloudWatch
- Term Project Proposals – Review by Thursday
- AWS overview and demonstration

- 2<sup>nd</sup> hour:
- AWS overview and demonstration
- Ch. 5: Cloud Enabling Technology
- Team planning

November 2, 2021

TCSS562:Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.11

11

OBJECTIVES – 11/2

- Questions from 10/28
- Tutorial 3 - Best Practices with EC2
- Tutorial 4 – Intro to FaaS – AWS Lambda
- **Tutorial 5 – Intro to FaaS II – Files in S3, CloudWatch**
- Term Project Proposals – Review by Thursday
- AWS overview and demonstration

- 2<sup>nd</sup> hour:
- AWS overview and demonstration
- Ch. 5: Cloud Enabling Technology
- Team planning

November 2, 2021

TCSS562:Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.12

12

OBJECTIVES – 11/2

- Questions from 10/28
- Tutorial 3 - Best Practices with EC2
- Tutorial 4 – Intro to FaaS – AWS Lambda
- Tutorial 5 – Intro to FaaS II – Files in S3, CloudWatch
- **Term Project Proposals – Review by Thursday**
- AWS overview and demonstration

- **2<sup>nd</sup> hour:**
- AWS overview and demonstration
- Ch. 5: Cloud Enabling Technology
- Team planning

November 2, 2021

TCSS562:Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.13

13

OBJECTIVES – 11/2

- Questions from 10/28
- Tutorial 3 - Best Practices with EC2
- Tutorial 4 – Intro to FaaS – AWS Lambda
- Tutorial 5 – Intro to FaaS II – Files in S3, CloudWatch
- Term Project Proposals – Review by Thursday
- **AWS overview and demonstration**


- **2<sup>nd</sup> hour:**
- AWS overview and demonstration
- Ch. 5: Cloud Enabling Technology
- Team planning

November 2, 2021

TCSS562:Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.14

14



# FUNCTION-AS-A-SERVICE

AWS  
Lambda  
Demo

15

15

# AWS OVERVIEW AND DEMO



16



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/r\\_c\\_cloud101\\_immersion.html](https://cloudmaven.github.io/documentation/r_c_cloud101_immersion.html)
- AWS Educate provides access to many online tutorials / learning resources

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.17

17

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

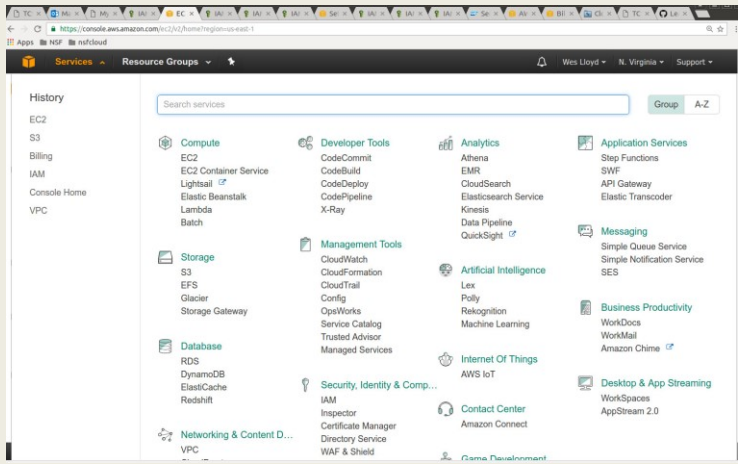
November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.1  
8

18

## AWS MANAGEMENT CONSOLE



The screenshot shows the AWS Management Console interface. At the top, there's a navigation bar with 'Services' and 'Resource Groups' tabs. Below this, a search bar and a 'Group A-Z' button are visible. The main area displays a grid of service categories: Compute (EC2, EC2 Container Service, Lightsail, Elastic Beanstalk, Lambda, Batch), Storage (S3, EFS, Glacier, Storage Gateway), Database (RDS, DynamoDB, ElastiCache, Redshift), Developer Tools (CodeCommit, CodeBuild, CodeDeploy, CodePipeline, X-Ray), Management Tools (CloudWatch, CloudFormation, CloudTrail, Config, OpsWorks, Service Catalog, Trusted Advisor, Managed Services), Analytics (Athena, EMR, CloudSearch, Elasticsearch Service, Kinesis, Data Pipeline, QuickSight), Artificial Intelligence (Lex, Polly, Rekognition, Machine Learning), Internet Of Things (AWS IoT), Contact Center (Amazon Connect), Application Services (Step Functions, SWF, API Gateway, Elastic Transcoder), Messaging (Simple Queue Service, Simple Notification Service, SES), Business Productivity (WorkDocs, WorkMail, Amazon Chime), and Desktop & App Streaming (WorkSpaces, AppStream 2.0). A footer bar contains the date 'November 2, 2021', the course name 'TCSS562: Software Engineering for Cloud Computing [Fall 2021]', the school name 'School of Engineering and Technology, University of Washington - Tacoma', and the slide number 'L10.19'.

19

## 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.

November 2, 2021 TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma L10.20

20

## 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

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.21

21

## 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...**

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.22

22

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...

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.23

23

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

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.24

24

## 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

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.25

25

## 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 per GB (for volumes sized 10GB to 1024 GB)
  - Credits - .72 minutes/day per GB

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.26

26

## ELASTIC FILE SYSTEM (EFS) - 2

### ■ Burstable Throughput Rates

*Information subject to revision*

- 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%

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.27

27

## ELASTIC FILE SYSTEM (EFS) - 3

### ■ Throughput Models

*Information subject to revision*

- 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

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.28

28

ELASTIC FILE SYSTEM (EFS) - 4

Performance Comparison, Amazon EFS and Amazon EBS

Information subject to revision

	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.

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.29

29

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

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.30

30

## EC2 VIRTUALIZATION - PARAVIRTUAL

- 1<sup>st</sup>, 2<sup>nd</sup>, 3<sup>rd</sup>, 4<sup>th</sup> generation → XEN-based
- 5<sup>th</sup> 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

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.31

31

## 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*)

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.32

32



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

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.33

33

EVOLUTION OF AWS VIRTUALIZATION

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

AWS EC2 Virtualization Types

Importance: Most → Least

CPU, Memory, Network I/O, Local Storage I/O, Remote Storage I/O, Interrupts, Timers, Motherboard, Boot

#	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
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
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): igbena driver, SR-IOV(storage): nvme driver.

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.34

34

INSTANCE ACTIONS

- Stop
  - Costs of “pausing” an instance
- Terminate
- Reboot
  
- Image management
- Creating an image
  - EBS (snapshot)
- Bundle image
  - Instance-store

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.35

35

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

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.36

36

SIMPLE VPC

Recommended when using Amazon EC2

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	igw-id

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.37

37

VPC SPANNING AVAILABILITY ZONES

Destination	Target
10.0.0.0/16	local

38

## 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

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.39

39

## 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.

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.40

40

# 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`

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.41

41

# AWS CLI - 2

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

Access key ID	Created	Last used	Status
AKIAJTVNQGFP6P6MZYQ	2017-04-02 22:56 PDT	2017-04-04 00:13 PDT with ec2 in N/A	Active   Make inactive

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.42

42

## 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`

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.43

43

## 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

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.44

44

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

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.45

45

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/setup-ami-tools.html?icmpid=docs\\_iam\\_console#ami-tools-create-certificate](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/setup-ami-tools.html?icmpid=docs_iam_console#ami-tools-create-certificate)

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.46

46

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 . . .

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.47

47

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/amiutils/cert-ec2.pem --no-inherit -r x86_64 -p $image -i
/etc/ec2/amiutils/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
```

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.48

48



## 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**

November 2, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L10.49
------------------	--	--------

49

## WE WILL RETURN AT ~6:10 PM



50

OBJECTIVES – 11/2

- Questions from 10/28
- Tutorial 3 - Best Practices with EC2
- Tutorial 4 – Intro to FaaS – AWS Lambda
- Tutorial 5 – Intro to FaaS II – Files in S3, CloudWatch
- Term Project Proposals – Review by Thursday
- AWS overview and demonstration

▪ 2<sup>nd</sup> hour:

- AWS overview and demonstration
- Ch. 5: Cloud Enabling Technology
- Team planning

November 2, 2021	TCSS562:Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L10.51
------------------	---	--------

51

OBJECTIVES – 11/2

- Questions from 10/28
- Tutorial 3 - Best Practices with EC2
- Tutorial 4 – Intro to FaaS – AWS Lambda
- Tutorial 5 – Intro to FaaS II – Files in S3, CloudWatch
- Term Project Proposals – Review by Thursday
- AWS overview and demonstration

▪ 2<sup>nd</sup> hour:

- AWS overview and demonstration
- Ch. 5: Cloud Enabling Technology
- Team planning

November 2, 2021	TCSS562:Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L10.52
------------------	---	--------

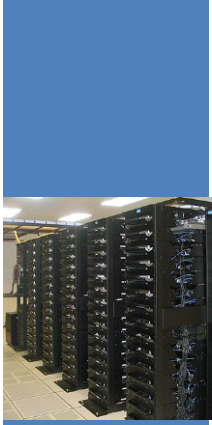
52

# CLOUD ENABLING TECHNOLOGY

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.53



53

## CLOUD ENABLING TECHNOLOGY

- Adapted from Ch. 5 from *Cloud Computing Concepts, Technology & Architecture*
- Broadband networks and internet architecture
- Data center technology
- Virtualization technology
- Multitenant technology
- Web/web services technology

November 2, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L10.54
------------------	--	--------

54

1. BROADBAND NETWORKS  
AND INTERNET ARCHITECTURE

- Clouds must be connected to a network
- Inter-networking: Users' network must connect to cloud's network
- Public cloud computing relies heavily on the Internet

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.55

55

PRIVATE CLOUD NETWORKING

- For institutions with in-house private clouds

The diagram illustrates a private cloud networking architecture. On the left, a cloud labeled 'private cloud network' contains three server icons and three desktop computer icons labeled 'in-office users'. A blue router icon is positioned at the edge of this cloud. A vertical dashed line separates the private network from the 'corporate Internet connection', which is represented by a globe with two red router icons on either side. To the right of the Internet connection, there are two user icons: one with a desktop computer labeled 'remote users' and another with a smartphone labeled 'users accessing cloud services remotely'. Blue lines represent network connections between the private cloud, the Internet, and the remote users.

November 2, 2021

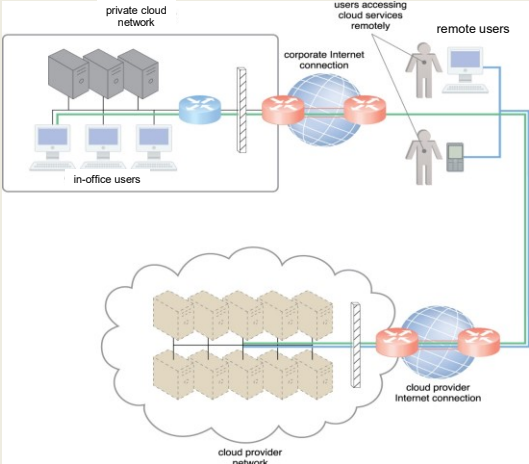
TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.56

56

## PUBLIC CLOUD NETWORKING

- Resources can be extended by adding public cloud
- Places further dependency on the internet to provide connectivity



The diagram illustrates the architecture of public cloud networking. It shows a 'private cloud network' on the left, containing 'in-office users' (represented by computer icons) and servers. This network is connected to a 'corporate Internet connection' (represented by a globe with red nodes). On the right, 'remote users' (represented by person icons with devices) are shown 'accessing cloud services remotely'. These remote users are connected to a 'cloud provider Internet connection' (another globe with red nodes), which in turn connects to the 'cloud provider network' (represented by a cloud shape containing server racks). The entire system relies on the Internet for connectivity between the private network and the cloud provider.

November 2, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma
------------------	--

L10.57

57

## INTERNETWORKING KEY POINTS

- Cloud consumers and providers typically communicate via the internet
- Decentralized provisioning and management model is not controlled by the cloud consumers or providers
- Inter-networking (internet) relies on connectionless packet switching and route-based interconnectivity
- Routers and switches support communication
- Network bandwidth and latency influence QoS, which is heavily impacted by network congestion

November 2, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma
------------------	--

L10.58

58

## CLOUD ENABLING TECHNOLOGY

- *Adapted from Ch. 5 from Cloud Computing Concepts, Technology & Architecture*
- Broadband networks and internet architecture
- Data center technology
- Virtualization technology
- Multitenant technology
- Web/web services technology

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.59

59

## 2. DATA CENTER TECHNOLOGY

- Grouping servers together (clusters):
  - Enables power sharing
  - Higher efficiency in shared IT resource usage (less duplication of effort)
  - Improved accessibility and organization
- Key components:
  - Virtualized and physical server resources
  - Standardized, modular hardware
  - Automation support: enable server provisioning, configuration, patching, monitoring without supervision... **tool/API support is desirable**

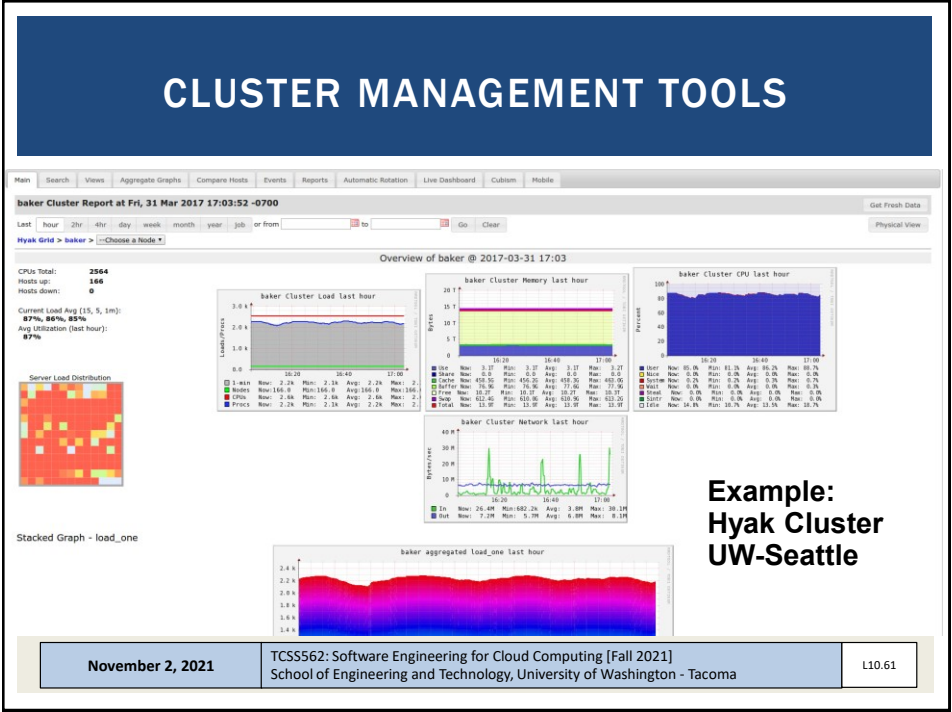


November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.60

60



Example:  
Hyak Cluster  
UW-Seattle

61

# DATA CENTER TECHNOLOGY – KEY COMPONENTS

- Remote operation / management
- High availability support:** \*\*redundant everything\*\*  
Includes: power supplies, cabling, environmental control systems, communication links, duplicate warm replica HW
- Secure design:** physical and logical access control
- Servers:** rackmount, etc.
- Storage:** hard disk arrays (RAID)
- storage area network (SAN): disk array w/ multiple servers (individual nodes w/ disks) and a dedicated network
- network attached storage (NAS): inexpensive single node with collection of disks, provides shared filesystems, for NFS, etc.
- Network hardware:** backbone routers (WAN to LAN connectivity), firewalls, VPN gateways, managed switches/routers

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.62

62

## CLOUD ENABLING TECHNOLOGY

- Broadband networks and internet architecture
- Data center technology
- Virtualization technology
- Multitenant technology
- Web/web services technology

November 2, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L10.63
------------------	--	--------

63

## 3. VIRTUALIZATION TECHNOLOGY

- Convert a physical IT resource into a virtual IT resource
- Servers, storage, network, power (virtual UPSs)
- Virtualization supports:
  - Hardware independence
  - Server consolidation
  - Resource replication
  - Resource pooling
  - Elastic scalability
- Virtual servers
  - Operating-system based virtualization
  - Hardware-based virtualization

November 2, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L10.64
------------------	--	--------

64



VIRTUAL MACHINES

- Emulation/simulation of a computer in software
- Provides a substitute for a real computer or server
- Virtualization platforms provide functionality to run an entire operating system
- Allows running multiple different operating systems, or operating systems with different versions simultaneously on the same computer

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.65


65

KEY VIRTUALIZATION TRADEOFF

▪ Tradeoff space:

**What is the “right” level of abstraction in the cloud for sharing resources with users?**

*Degree of Hardware Abstraction*



**Abstraction Concerns:**

- Overhead
- Performance
- Isolation
- Security

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.66

66

## ABSTRACTION CONCERNS

- **Overhead with too many instances w/ heavy abstractions**
  - Too many instances using a heavy abstraction can lead to hidden resource utilization and waste
  - Example: Dedicated server with 48 VMs each with separate instance of Ubuntu Linux
  - Idle VMs can reduce performance of co-resident jobs/tasks
- **“Virtualization” Overhead**
  - Cost of virtualization an OS instance
  - Overhead has dropped from ~100% to ~1% over last decade
- **Performance**
  - Impacted by weight of abstraction and virtualization overhead

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.67

67

## ABSTRACTION CONCERNS - 2

- **Isolation**
  - From others:  
What user A does should not impact user B in any noticeable way
- **Security**
  - User A and user B's data should be always separate
  - User A's actions are not perceivable by User B

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.68

68

## TYPES OF ABSTRACTION IN THE CLOUD

- **Virtual Machines** – original IaaS cloud abstraction
- **OS and Application Containers** – seen with CaaS
  - OS Container – replacement for VM, mimics full OS instance, heavier
  - OS containers run 100s of processes just like a VM
  - App Container – Docker: packages dependencies to easily transport and run an application anywhere
  - Application containers run only a few processes
- **Micro VMs** – FaaS / CaaS
  - Lighter weight alternative to full VM (KVM, XEN, VirtualBox)
  - Firecracker
- **Unikernel Operating Systems** – research mostly
  - Single process, multi-thread operating system
  - Designed for cloud, objective to reduce overhead of running too many OS instances

November 2, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L10.69
------------------	--	--------

69

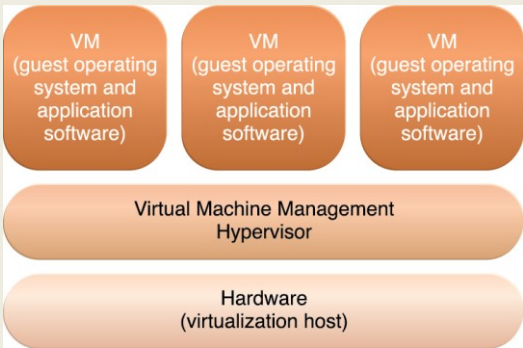
## VIRTUAL MACHINES

- Type 1 hypervisor
  - Typically involves a special virtualization kernel that runs directly on the system to share the underlying machine with many guest VMs
  - Paravirtualization introduced to directly share system resources with guests bypassing full emulation
  - VM becomes equal participant in sharing the network card for example
- Type 2 hypervisor
  - Typically involves the **Full Virtualization** of the guest, where everything is simulated/emulated
- **Hardware level support** (i.e. features introduced on CPUs) have made virtualization faster in all respects shrinking virtualization overhead

November 2, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L10.70
------------------	--	--------

70

## TYPE 1 HYPERVISOR



- Host OS and VMs run atop the hypervisor
- The boot OS is the hypervisor kernel
- Xen dom0

November 2, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L10.71
------------------	--	--------

71

## TYPE 1 HYPERVISOR

- Acts as a control program
- Miniature OS kernel that manages VMs
- Boots and runs on bare metal
- Also known as Virtual Machine Monitor (VMM)
- **Paravirtualization:** Kernel includes I/O drivers
- VM guest OSes must use special kernel to interoperate
- Paravirtualization provides hooks to the guest VMs
- Kernel traps instructions (i.e. device I/O) to implement sharing & multiplexing
- User mode instructions run directly on the CPU
- Objective: minimize virtualization overhead
- Classic example is XEN (dom0 kernel)

November 2, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L10.72
------------------	--	--------

72

COMMON VMMS:  
PARAVIRTUALIZATION

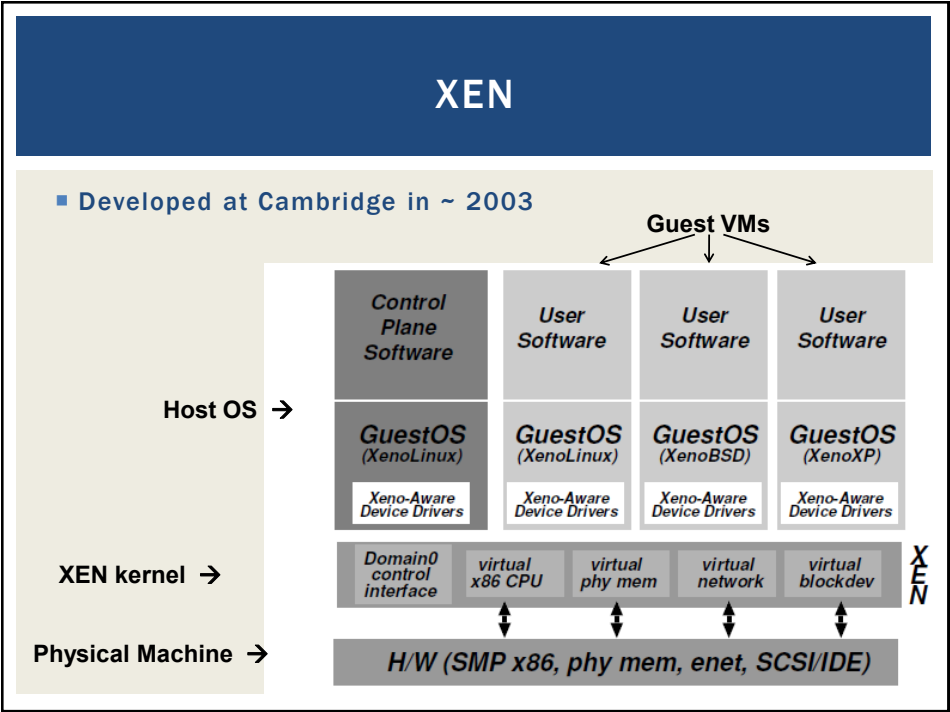
- TYPE 1 Hypervisor
  - XEN
    - Citrix Xen-server (a commercial version of XEN)
  - VMWare ESXi
  - KVM (virtualization support in kernel)  
  - Paravirtual I/O drivers introduced
    - XEN
    - KVM
    - Virtualbox

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.73

73



74

XEN - 2

- VMs managed as “domains”
- Domain 0 is the hypervisor domain
  - Host OS is installed to run on bare-metal, but doesn't directly facilitate virtualization (*unlike KVM*)
- Domains 1..n are guests (VMs) – not bare-metal

```
xentop - 17:53:48 Xen 3.1.2-398.el5
3 domains: 1 running, 2 blocked, 0 paused, 0 crashed, 0 dying, 0 shutdown
Mem: 8379564k total, 8377876k used, 1688k free CPUs: 4 @ 2400MHz
```

	NAME	STATE	CPU(sec)	CPU(%)	MEM(k)	MEM(%)	MAXMEM(k)	MAXMEM(%)	VCPUS	
	NETS	NETTX(k)	NETRX(k)	VBDS	VBD OO	VBD RD	VBD WR	SSID		
	centos	--b---		46	0.0	532352	6.4	1064960	12.7	1
	1	27960	885	1	0	6313	37119	0		
	centos-2	--b---		17	0.0	1056640	12.6	2113536	25.2	1
	1	50	0	1	0	3981	541	0		
	Domain-0	-----r		2979	19.3	6568960	78.4	no limit	n/a	4
	4	1057374	290072	0	0	0	0	0		

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.75

75

XEN - 3

- Physical machine boots special XEN kernel
- Kernel provides paravirtual API to manage CPU & device multiplexing
- Guests require modified XEN-aware kernels
- Xen supports full-virtualization for unmodified OS guests in hvm mode
- Amazon EC2 largely based on modified version of XEN hypervisor (EC2 gens 1-4)
- XEN provides its own CPU schedulers, I/O scheduling

November 2, 2021

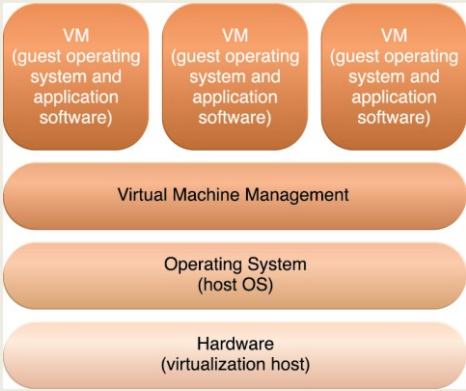
TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.76

76

## TYPE 2 HYPERVISOR

- Adds additional layer



November 2, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L10.77
------------------	--	--------

77

## TYPE 2 HYPERVISOR

- Problem: Original x86 CPUs could not trap special instructions
- Instructions not specially marked
- Solution: Use Full Virtualization
- Trap ALL instructions
- “Fully” simulate entire computer
- Tradeoff: Higher Overhead
- Benefit: Can virtualize any operating system without modification

November 2, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L10.78
------------------	--	--------

78

## CHECK FOR VIRTUALIZATION SUPPORT

- See:  
<https://cyberciti.biz/faq/linux-xen-vmware-kvm-intel-vt-amd-v-support>
- # check for Intl VT CPU virtualization extensions on Linux  
`grep -color vmx /proc/cpuinfo`
- # check for AMD V CPU virtualization extensions on Linux  
`grep -color svm /proc/cpuinfo`
- Also see 'lscpu' → "Virtualization:"
- Other Intel CPU features that help virtualization:  
`ept vpid tpr_shadow flexpriority vnmi`

November 2, 2021

TCCS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.79

79

## KERNEL BASED VIRTUAL MACHINES (KVM)

- x86 HW notoriously difficult to virtualize
- Extensions added to 64-bit Intel/AMD CPUs
  - Provides hardware assisted virtualization
  - New "guest" operating mode
  - Hardware state switch
  - Exit reason reporting
  - Intel/AMD implementations different
    - Linux uses vendor specific kernel modules

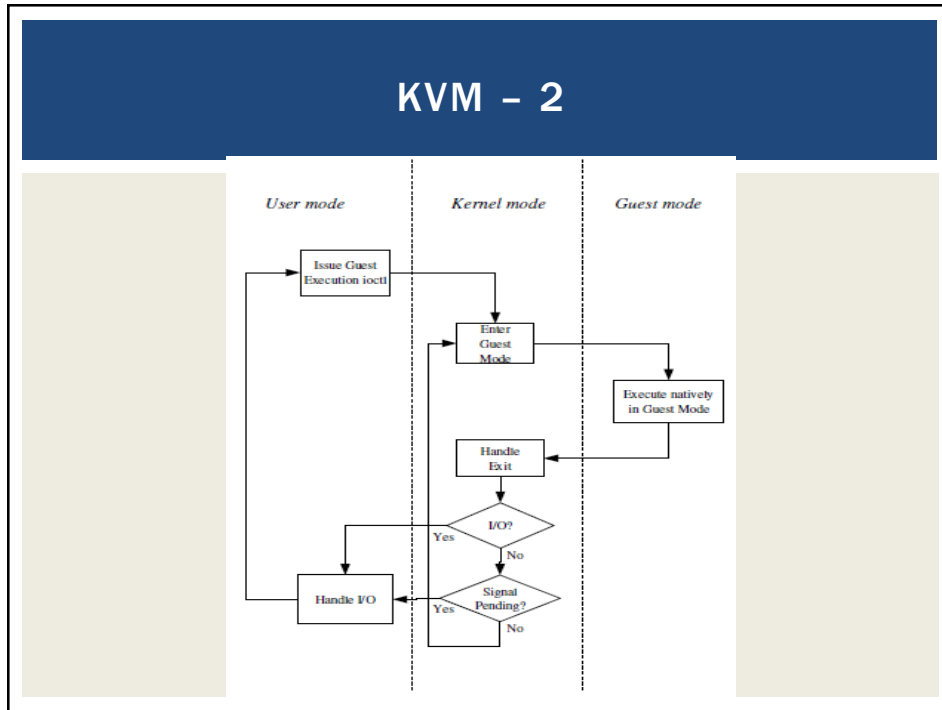
November 2, 2021

TCCS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.80

80





81

## KVM - 3

- **KVM has /dev/kvm device file node**
  - **Linux character device, with operations:**
    - Create new VM
    - Allocate memory to VM
    - Read/write virtual CPU registers
    - Inject interrupts into vCPUs
    - Running vCPUs
- **VMs run as Linux processes**
  - **Scheduled by host Linux OS**
  - **Can be pinned to specific cores with “taskset”**

November 2, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L10.82
------------------	--	--------

82

## KVM PARAVIRTUALIZED I/O

- KVM – Virtio
  - Custom Linux based paravirtual device drivers
  - Supersedes QEMU hardware emulation (full virt.)
  - Based on XEN paravirtualized I/O
  - Custom block device driver provides paravirtual device emulation
    - Virtual bus (memory ring buffer)
    - Requires hypercall facility
    - Direct access to memory

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.83

83

## KVM DIFFERENCES FROM XEN

- KVM requires CPU VMX support
  - Virtualization management extensions
- KVM can virtualize any OS without special kernels
  - Less invasive
- KVM was originally separate from the Linux kernel, but then integrated
- KVM is type 1 hypervisor because the machine boots Linux which has integrated support for virtualization
- Different than XEN because XEN kernel alone is not a full-fledged OS

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.84

84

KVM ENHANCEMENTS

- Paravirtualized device drivers
  - Virtio
- Guest Symmetric Multiprocessor (SMP) support
  - Leverages multiple on-board CPUs
  - Supported as of Linux 2.6.23
- VM Live Migration
- Linux scheduler integration
  - Optimize scheduler with knowledge that KVM processes are virtual machines

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.85

85

FIRECRACKER MICRO VM

From <https://firecracker-microvm.github.io/>

The following diagram depicts an example host running Firecracker microVMs.

The diagram illustrates the Firecracker microVM architecture. On the left, a stack of microVMs is shown running on a **BARE METAL SERVER** within the **HOST KERNEL SPACE**. Each microVM contains **KVM** and **I/O** components. A callout states: "Firecracker scales to thousands of multitenant microVMs." and "Configurable microVMs across CPU and memory, running as user space processes." On the right, a detailed view of a microVM shows its internal components: **RESTful API**, **Network**, **Storage**, **Metadata Service**, and **Rate limiting**. A **Client** is shown interacting with the **RESTful API**. The microVM is labeled **Guest OS & Container workload** and **Firecracker**. Above the microVM, a legend defines the **CONTROL PLANE** (dashed line) and **DATA PLANE** (solid line), and identifies the **VIRTUALIZATION BARRIER** (orange box) and **JAILER BARRIER** (grey box).

86

## FIRECRACKER MICRO VM

- Provides a virtual machine monitor (VMM) (i.e. hypervisor) using KVM to create and manage microVMs
- Has a minimalist design with goals to improve security, decreases the startup time, and increases hardware utilization
- Excludes unnecessary devices and guest functionality to reduce memory footprint and attack surface area of each microVM
- Supports boot time of <125ms, <5 MiB memory footprint
- Can run 100s of microVMs on a host, launching up to 150/sec
- Is available on 64-bit Intel, AMD, and Arm CPUs
- Used to host AWS Lambda and AWS Fargate
- Has been open sourced under the Apache 2.0 license

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.87

87

## FIRECRACKER - 2

- Minimalistic
- MicroVMs run as separate processes on the host
- Only 5 emulated devices are available: virtio-net, virtio-block, virtio-vsock, serial console, and a minimal keyboard controller used only to stop the microVM
- Rate limiters can be created and configured to provision resources to support bursts or specific bandwidth/operation limitations
- Configuration
- A RESTful API enables common actions such as configuring the number of vCPUs or launching microVMs
- A metadata service between the host and guest provides configuration information

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.88

88

## FIRECRACKER - 2

- **Security**
- Runs in user space (***not the root user***) on top of the Linux Kernel-based Virtual Machine (KVM) hypervisor to create microVMs
- Lambda functions, Fargate containers, or container groups can be encapsulated using Firecracker through KVM, enabling workloads from different customers to run on the same machine, without sacrificing security or efficiency
- MicroVMs are further isolated with common Linux user-space security barriers using a companion program called “jailer” which provides a second line of defense if KVM is compromised

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.89

89

## UNIKERNELS

- Lightweight alternative to containers and VMs
  - Custom Cloud Operating System
  - Single process, multiple threads, runs one program
  - Launch separately atop of hypervisor (XEN/KVM)
  - Reduce overhead, duplication of heavy weight OS
- OSv is most well known unikernel
- Several others exist has research projects
- More information at: <http://unikernel.org/>
- Google Trends OSv →



November 7, 2018

TCSS598: Ma  
School of Eng

90

## VIRTUALIZATION MANAGEMENT

- Virtual infrastructure management (VIM) tools
- Tools that manage pools of virtual machines, resources, etc.
- Private cloud software systems can be considered as a VIM
- Considerations:
- Performance overhead
  - Paravirtualization: custom OS kernels, I/O passed directly to HW w/ special drivers
- Hardware compatibility for virtualization
- Portability: virtual resources tend to be difficult to migrate cross-clouds

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.91

91

## VIRTUAL INFRASTRUCTURE MANAGEMENT (VIM)

- Middleware to manage virtual machines and infrastructure of IaaS “clouds”
- Examples
  - OpenNebula
  - Nimbus
  - Eucalyptus
  - OpenStack

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.92

92

VIM FEATURES

- Create/destroy VM Instances
- Image repository
  - Create/Destroy/Update images
  - Image persistence
- Contextualization of VMs
  - Networking address assignment
    - DHCP / Static IPs
  - Manage SSH keys

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.93

93

VIM FEATURES - 2

- Virtual network configuration/management
  - Public/Private IP address assignment
  - Virtual firewall management
    - Configure/support isolated VLANs (private clusters)
- Support common virtual machine managers (VMMs)
  - XEN, KVM, VMware
  - Support via libvirt library

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.94

94

VIM FEATURES - 3

- Shared “Elastic” block storage
  - Facility to create/update/delete VM disk volumes
    - Amazon EBS
    - Eucalyptus SC
    - OpenStack Volume Controller

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.95

95

CONTAINER ORCHESTRATION  
FRAMEWORKS

- Middleware to manage Docker application container deployments across virtual clusters of Docker hosts (VMs)
- Considered Infrastructure-as-a-Service
- Opensource
  - Kubernetes framework
  - Docker swarm
  - Apache Mesos/Marathon
- Proprietary
  - Amazon Elastic Container Service

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.96

96



## CONTAINER SERVICES

- Public cloud container cluster services
  - Azure Kubernetes Service (AKS)
  - Amazon Elastic Container Service for Kubernetes (EKS)
  - Google Kubernetes Engine (GKE)
- Container-as-a-Service
  - Azure Container Instances (ACI – April 2018)
  - AWS Fargate (November 2017)
  - Google Kubernetes Engine Serverless Add-on (alpha-July 2018)

November 2, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L10.97
------------------	--	--------

97

## CLOUD ENABLING TECHNOLOGY


- *Adapted from Ch. 5 from Cloud Computing Concepts, Technology & Architecture*
- Broadband networks and internet architecture
- Data center technology
- Virtualization technology
- Multitenant technology
- Web/web services technology

November 2, 2021	TCSS562: Software Engineering for Cloud Computing [Fall 2021] School of Engineering and Technology, University of Washington - Tacoma	L10.98
------------------	--	--------

98

4. MULTITENANT APPLICATIONS

- Each tenant (like in an apartment) has their own view of the application
- Tenants are unaware of their neighbors
- Tenants can only access their data, no access to data and configuration that is not their own
- Customizable features
  - UI, business process, data model, access control
- Application architecture
  - User isolation, data security, recovery/backup by tenant, scalability for a tenant, for tenants, metered usage, data tier isolation



November 2, 2021

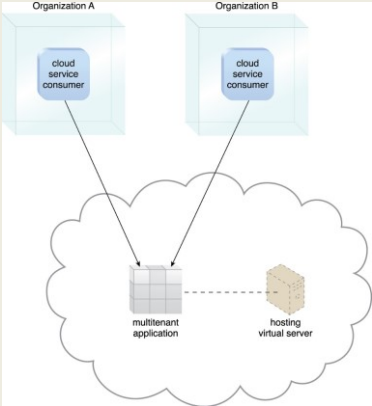
TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.99

99

MULTITENANT APPS - 2

- Forms the basis for SaaS (applications)



November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.100

100

## CLOUD ENABLING TECHNOLOGY

- *Adapted from Ch. 5 from Cloud Computing Concepts, Technology & Architecture*
- Broadband networks and internet architecture
- Data center technology
- Virtualization technology
- Multitenant technology
- Web/web services technology

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.101

101

## 5. WEB SERVICES/WEB

- Web services technology is a key foundation of cloud computing's "as-a-service" cloud delivery model
- SOAP – “Simple” object access protocol
  - First generation web services
  - WSDL – web services description language
  - UDDI – universal description discovery and integration
  - SOAP services have their own unique interfaces
- REST – instead of defining a custom technical interface REST services are built on the use of HTTP protocol
- HTTP GET, PUT, POST, DELETE

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.102

102

## HYPERTEXT TRANSPORT PROTOCOL (HTTP)

- An ASCII-based request/reply protocol for transferring information on the web
- HTTP request includes:
  - request method (GET, POST, etc.)
  - Uniform Resource Identifier (URI)
  - HTTP protocol version understood by the client
  - headers—extra info regarding transfer request
- HTTP response from server
  - Protocol version & status code →
  - Response headers
  - Response body

**HTTP status codes:**  
2xx — *all is well*  
3xx — *resource moved*  
4xx — *access problem*  
5xx — *server error*

November 2, 2021

TCCS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.103

103

## REST: REPRESENTATIONAL STATE TRANSFER

- Web services protocol
- *Supersedes SOAP* – Simple Object Access Protocol
- Access and manipulate web resources with a predefined set of stateless operations (known as web services)
- Requests are made to a URI
- Responses are most often in JSON, but can also be HTML, ASCII text, XML, no real limits as long as text-based
- HTTP verbs: GET, POST, PUT, DELETE, ...

November 2, 2021

TCCS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.104

104

```
// SOAP REQUEST

POST /InStock HTTP/1.1
Host: www.bookshop.org
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn

<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-
encoding">
<soap:Body xmlns:m="http://www.bookshop.org/prices">
  <m:GetBookPrice>
    <m:BookName>The Fleamarket</m:BookName>
  </m:GetBookPrice>
</soap:Body>
</soap:Envelope>
```

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.10  
5

105

```
// SOAP RESPONSE

POST /InStock HTTP/1.1
Host: www.bookshop.org
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn

<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-
encoding">
<soap:Body xmlns:m="http://www.bookshop.org/prices">
  <m:GetBookPriceResponse>
    <m: Price>10.95</m: Price>
  </m:GetBookPriceResponse>
</soap:Body>
</soap:Envelope>
```

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.10  
6

106

```
// WSDL Service Definition
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="DayOfWeek"
  targetNamespace="http://www.roguewave.com/soapworx/examples/DayOfWeek.wsdl"
  xmlns:tns="http://www.roguewave.com/soapworx/examples/DayOfWeek.wsdl"
  xmlns:soap="http://schemas.xmlsoap.org/soap/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://schemas.xmlsoap.org/wsdl/">
  <message name="DayOfWeekInput">
    <part name="date" type="xsd:date"/>
  </message>
  <message name="DayOfWeekResponse">
    <part name="dayOfWeek" type="xsd:string"/>
  </message>
  <portType name="DayOfWeekPortType">
    <operation name="GetDayOfWeek">
      <input message="tns:DayOfWeekInput"/>
      <output message="tns:DayOfWeekResponse"/>
    </operation>
  </portType>
  <binding name="DayOfWeekBinding" type="tns:DayOfWeekPortType">
    <soap:binding style="document"
      transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="GetDayOfWeek">
      <soap:operation soapAction="getdayofweek"/>
      <input>
        <soap:body use="encoded"
          namespace="http://www.roguewave.com/soapworx/examples"
          encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
      </input>
      <output>
        <soap:body use="encoded"
          namespace="http://www.roguewave.com/soapworx/examples"
          encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
      </output>
    </operation>
  </binding>
  <service name="DayOfWeekService" >
    <documentation>
      Returns the day-of-week name for a given date
    </documentation>
    <port name="DayOfWeekPort" binding="tns:DayOfWeekBinding">
      <soap:address location="http://localhost:8090/dayofweek/DayOfWeek" />
    </port>
  </service>
</definitions>
```

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.10  
7

107

# REST CLIMATE SERVICES EXAMPLE

■ USDA  
Lat/Long  
Climate  
Service  
Demo

■ Just provide  
a Lat/Long

```
// REST/JSON
// Request climate data for Washington

{
  "parameter": [
    {
      "name": "latitude",
      "value": 47.2529
    },
    {
      "name": "longitude",
      "value": -122.4443
    }
  ]
}
```

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.108

108

Slides by Wes J. Lloyd

L10.54

REST - 2

- App manipulates one or more types of resources.
- Everything the app does can be characterized as some kind of operation on one or more resources.
- Frequently services are CRUD operations (create/read/update/delete)
  - Create a new resource
  - Read resource(s) matching criterion
  - Update data associated with some resource
  - Destroy a particular a resource
- Resources are often implemented as objects in OO languages

November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.109

109

REST ARCHITECTURAL ADVANTAGES

- **Performance:** component interactions can be the dominant factor in user-perceived performance and network efficiency
- **Scalability:** to support large numbers of services and interactions among them
- **Simplicity:** of the Uniform Interface
- **Modifiability:** of services to meet changing needs (even while the application is running)
- **Visibility:** of communication between services
- **Portability:** of services by redeployment
- **Reliability:** resists failure at the system level as redundancy of infrastructure is easy to ensure


November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.110

110

QUESTIONS-



November 2, 2021

TCSS562: Software Engineering for Cloud Computing [Fall 2021]  
School of Engineering and Technology, University of Washington - Tacoma

L10.11  
1

111