TCSS 462/562: (SOFTWARE ENGINEERING FOR) CLOUD COMPUTING

Team 6, 7, 8 Presentations

Practice Talks:
Naman Bhaia, Xinghan Chen

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

1

OFFICE HOURS – FALL 2023

- **THIS WEEK**
- **Tuesday:**
  - 2:30 to 3:30 pm - CP 229
- **Friday:**
  - 11:00 am to 12:00 pm – ONLINE via Zoom
- Or email for appointment

> Office Hours set based on Student Demographics survey feedback

2

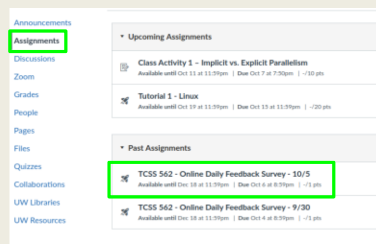
OBJECTIVES – 12/7

- Questions from 12/5
- Quiz 2 Posted - Available thru December 11 @ 11:59pm
- Class Presentations Schedule - Cloud Technology or Research Paper Review
- A2: Term Project Report or Presentation
- Tutorial 8: AWS Step Functions, AWS SQS
- Team 6: Paper - Evicting for the Greater Good: The Case for Reactive Check Pointing in Serverless Computing
- Team 7: Cloud Technology: Amazon Redshift
- Team 8: Paper - Rendezvous - Where Serverless Functions Find Consistency
- Practice Talks: Naman Bhaia WoC, Xinghan Chen WOSC

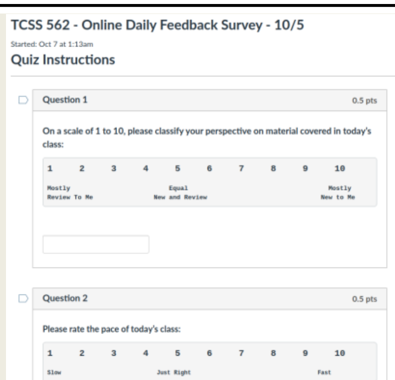
3

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

5

MATERIAL / PACE

- Please classify your perspective on material covered in today's class (**44** respondents):
- 1-mostly review, 5-equal new/review, 10-mostly new
- **Average – 5.95 (↓ - *previous 6.04*)**

- Please rate the pace of today's class:
- 1-slow, 5-just right, 10-fast
- **Average – 5.16 (↓ - previous 5.32)**

- **Response rates:**
- TCSS 462: 26/44 – 59.0%
- TCSS 562: 18/25 – 72.0%

6

FEEDBACK FROM 12/5

- ..

7

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
  - 63 credit requests fulfilled as of Dec 6 @ 11:59p
- Codes not provided using discord

8

Don't Forget to Terminate (Shutdown) all EC2 instances for Tutorials 3 & 7

Spot instances:
c5d.large instance @ ~3c cents / hour

$0.72 / day
$5.04 / week
$21.88 / month
$262.80 / year

AWS CREDITS → → → → → → → →

9

OBJECTIVES – 12/7

- Questions from 12/5
- **Quiz 2 Posted - Available thru December 11 @ 11:59pm**
- Class Presentations Schedule - Cloud Technology or Research Paper Review
- A2: Term Project Report or Presentation
- Tutorial 8: AWS Step Functions, AWS SQS
- Team 6: Paper - Evicting for the Greater Good: The Case for Reactive Check Pointing in Serverless Computing
- Team 7: Cloud Technology: Amazon Redshift
- Team 8: Paper - Rendezvous - Where Serverless Functions Find Consistency
- Practice Talks: Naman Bhaia WoC, Xinghan Chen WOSC

10

QUIZ 2

- Opened **Thursday December 7 at 8:00 am**
- Closes **Monday December 11 at 11:59 pm**
- Individual work only
- Please answer every question
- Book, notes, slides, calculator, and internet are allowed

- **Grading:**
- The Canvas autograder produces a preliminary score, not the final score.
- The instructor will manually review all quizzes and add partial credit
- A curve adjustment may be applied as appropriate
- Updates may not occur until several days after the quiz closes
- Please report suspected grading problems to the instructor

- **Attempts:**
- 1 quiz attempt, 120 minute limit, 20 questions.
- Coverage is inclusive of Lectures ~1-16
- Once started, there will be 2 hours to complete, please plan accordingly.

11

OBJECTIVES – 12/7

- Questions from 12/5
- Quiz 2 Posted - Available thru December 11 @ 11:59pm
- **Class Presentations Schedule - Cloud Technology or Research Paper Review**
- A2: Term Project Report or Presentation
- Tutorial 8: AWS Step Functions, AWS SQS
- Team 6: Paper - Evicting for the Greater Good: The Case for Reactive Check Pointing in Serverless Computing
- Team 7: Cloud Technology: Amazon Redshift
- Team 8: Paper - Rendezvous - Where Serverless Functions Find Consistency
- Practice Talks: Naman Bhaia WoC, Xinghan Chen WOSC

12

GROUP PRESENTATIONS

- TWO OPTIONS:
- *Cloud technology presentation*
- *Cloud research paper presentation*
  - Recent & suggested papers will be posted at: http://faculty.washington.edu/wlloyd/courses/tcss562/papers/
- Presentation dates:
  - Tuesday November 28, Tuesday November 30
  - Tuesday December 5, Thursday December 7
- Peer Reviews
  - Word DOCX form will be provided, fill out, submit PDF on Canvas
  - Feedback shared with groups
  - TCSS 462: 1 review/day required, additional are extra credit
  - TCSS 562: same as 462, but no peer review req'd on day of your talk

December 7, 2023 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma | L20.13

13

PRESENTATION SCHEDULE

- **Tuesday December 5**

1. Kewei Liu, Sherry Liu (team 15)
Research paper: AWSomePy : A Dataset and Characterization of Serverless Applications
2. Sanjay Vuppugandla, Sai Prateek Atluri, Ankit Kadian (team 9*)
Cloud Technology: AWS CloudFormation

- **Thursday December 7**

1. Cynthia Pang, Lifan Cao (team 6)
Research paper: Evicting for the Greater Good: The Case for Reactive Check Pointing in Serverless Computing
2. Srishty, Angela C Farin, Tomoki Kusunoki (team 7)
Cloud Technology: Amazon Redshift
3. Xiaoqing Zhou, Mary Yang, Micaela Nomakchteinsky (team 8)
Research paper: Rendezvous - Where Serverless Functions Find Consistency

December 7, 2023 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma | L20.14

14

OBJECTIVES – 12/7

- Questions from 12/5
- Quiz 2 Posted - Available thru December 11 @ 11:59pm
- Class Presentations Schedule - Cloud Technology or Research Paper Review
- **A2: Term Project Report or Presentation**
- Tutorial 8: AWS Step Functions, AWS SQS
- Team 6: Paper - Evicting for the Greater Good: The Case for Reactive Check Pointing in Serverless Computing
- Team 7: Cloud Technology: Amazon Redshift
- Team 8: Paper - Rendezvous - Where Serverless Functions Find Consistency
- Practice Talks: Naman Bhaia WoC, Xinghan Chen WOSC

December 7, 2023 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma | L20.15

15

OBJECTIVES – 12/7

- Questions from 12/5
- Quiz 2 Posted - Available thru December 11 @ 11:59pm
- Class Presentations Schedule - Cloud Technology or Research Paper Review
- A2: Term Project Report or Presentation
- **Tutorial 8: AWS Step Functions, AWS SQS**
- Team 6: Paper - Evicting for the Greater Good: The Case for Reactive Check Pointing in Serverless Computing
- Team 7: Cloud Technology: Amazon Redshift
- Team 8: Paper - Rendezvous - Where Serverless Functions Find Consistency
- Practice Talks: Naman Bhaia WoC, Xinghan Chen WOSC

December 7, 2023 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma | L20.16

16

TUTORIAL 8 – DEC 15

- Introduction to AWS Step Functions and Amazon Simple Queue Service (SQS)
- Not Required, available for extra credit
  - adds points to overall tutorials score
- https://faculty.washington.edu/wlloyd/courses/tcss562/tutorials/TCSS462_562_f2023_tutorial_8.pdf
- Tasks
  - Adapt Caesar Cipher Lambda functions for use with AWS Step Functions
  - Create AWS Step Functions State Machine
  - Create a BASH client to invoke the AWS Step Function
  - Create Simple Queue Service Queue for messages
  - Add message to SQS queue from AWS Lambda function
  - Modify AWS Step Function Bash client script to retrieve AWS Step Function result from SQS queue

November 16, 2023 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma | L15.17

17

OBJECTIVES – 12/7

- Questions from 12/5
- Quiz 2 Posted - Available thru December 11 @ 11:59pm
- Class Presentations Schedule - Cloud Technology or Research Paper Review
- A2: Term Project Report or Presentation
- Tutorial 8: AWS Step Functions, AWS SQS
- **Team 6: Paper - Evicting for the Greater Good: The Case for Reactive Check Pointing in Serverless Computing**
- Team 7: Cloud Technology: Amazon Redshift
- Team 8: Paper - Rendezvous - Where Serverless Functions Find Consistency
- Practice Talks: Naman Bhaia WoC, Xinghan Chen WOSC

December 7, 2023 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma | L20.18

18

OBJECTIVES – 12/7

- Questions from 12/5
- Quiz 2 Posted - Available thru December 11 @ 11:59pm
- Class Presentations Schedule - Cloud Technology or Research Paper Review
- A2: Term Project Report or Presentation
- Tutorial 8: AWS Step Functions, AWS SQS
- Team 6: Paper - Evicting for the Greater Good: The Case for Reactive Check Pointing in Serverless Computing
- **Team 7: Cloud Technology: Amazon Redshift**
- Team 8: Paper - Rendezvous - Where Serverless Functions Find Consistency
- Practice Talks: Naman Bhaia WoC, Xinghan Chen WOSC

December 7, 2023 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma | L20.19

19

OBJECTIVES – 12/7

- Questions from 12/5
- Quiz 2 Posted - Available thru December 11 @ 11:59pm
- Class Presentations Schedule - Cloud Technology or Research Paper Review
- A2: Term Project Report or Presentation
- Tutorial 8: AWS Step Functions, AWS SQS
- Team 6: Paper - Evicting for the Greater Good: The Case for Reactive Check Pointing in Serverless Computing
- Team 7: Cloud Technology: Amazon Redshift
- **Team 8: Paper - Rendezvous - Where Serverless Functions Find Consistency**
- Practice Talks: Naman Bhaia WoC, Xinghan Chen WOSC

December 7, 2023 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma | L20.20

20

OBJECTIVES – 12/7

- Questions from 12/5
- Quiz 2 Posted - Available thru December 11 @ 11:59pm
- Class Presentations Schedule - Cloud Technology or Research Paper Review
- A2: Term Project Report or Presentation
- Tutorial 8: AWS Step Functions, AWS SQS
- Team 6: Paper - Evicting for the Greater Good: The Case for Reactive Check Pointing in Serverless Computing
- Team 7: Cloud Technology: Amazon Redshift
- Team 8: Paper - Rendezvous - Where Serverless Functions Find Consistency
- **Practice Talks: Naman Bhaia WoC**, Xinghan Chen WOSC

December 7, 2023 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma | L20.21

21

OBJECTIVES – 12/7

- Questions from 12/5
- Quiz 2 Posted - Available thru December 11 @ 11:59pm
- Class Presentations Schedule - Cloud Technology or Research Paper Review
- A2: Term Project Report or Presentation
- Tutorial 8: AWS Step Functions, AWS SQS
- Team 6: Paper - Evicting for the Greater Good: The Case for Reactive Check Pointing in Serverless Computing
- Team 7: Cloud Technology: Amazon Redshift
- Team 8: Paper - Rendezvous - Where Serverless Functions Find Consistency
- Practice Talks: Naman Bhaia WoC, **Xinghan Chen WOSC**

December 7, 2023 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma | L20.22

22

WE WILL RETURN AT ~4:50 PM

23

OBJECTIVES – 12/7

- Questions from 12/5
- Quiz 2 Posted - Available thru December 11 @ 11:59pm
- Class Presentations Schedule - Cloud Technology or Research Paper Review
- A2: Term Project Report or Presentation
- Tutorial 8: AWS Step Functions, AWS SQS
- Team 6: Paper - Evicting for the Greater Good: The Case for Reactive Check Pointing in Serverless Computing
- Team 7: Cloud Technology: Amazon Redshift
- **Kubernetes**

December 7, 2023 | TCSS462/562:(Software Engineering for) Cloud Computing [Fall 2023] School of Engineering and Technology, University of Washington - Tacoma | L20.24

24

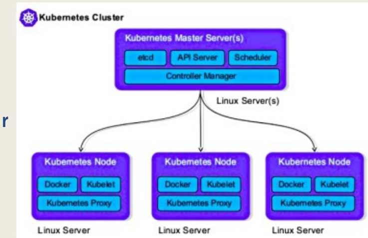
KUBERNETES

from: "The Kubernetes Book", Nigel Poulton and Pushkar Joglekar, Version 7.0, September 2020

25

MASTER SERVICES

- API Server
- Cluster store
- Controller Manager
- Scheduler
- Cloud controller



26

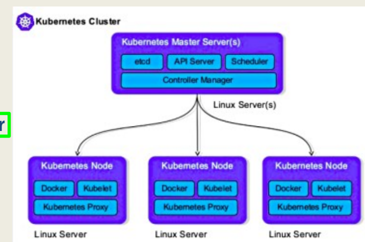
CLOUD CONTROLLER MANAGER

- Abstracts and manages integration with specific cloud(s)
- Manages vendor specific cloud infrastructure to provide instances (VMs), load balancing, storage, etc.
- Support for AWS, Azure, GCP, Digital Ocean, IBM, etc.

27

MASTER SERVICES

- API Server
- Cluster store
- Controller Manager
- Scheduler
- Cloud controller



28

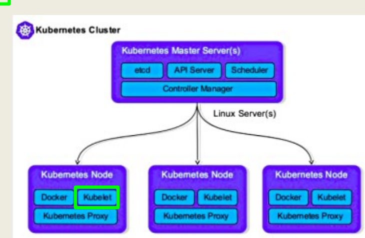
WORKER NODES

- Nodes perform tasks (i.e. host containers & services)
- Three primary functions:
  1. Wait for the scheduler to assign work
  2. Execute work (host containers, etc.)
  3. Report back state information, etc.
- Nodes are considerably simpler than masters

29

WORKER NODES

- Kubelet
- Container runtime *(Docker, etc.)*
- Kubernetes Proxy



30

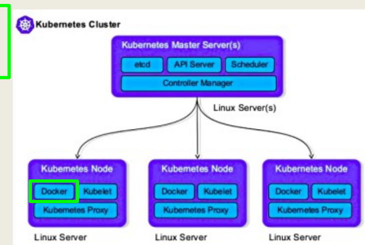
KUBELET

- Main Kubernetes agent
- Runs on every node
- Adding a new node installs the kubelet onto the node
- Kubelet registers the node with the cluster
- Monitors API server for new work assignments
- Maintains reporting back to control plane
- When a node can't run a task, kubelet is NOT responsible for finding an alternate node

31

WORKER NODES

- Kubelet
- Container runtime (*Docker, etc.)*
- Kubernetes Proxy



32

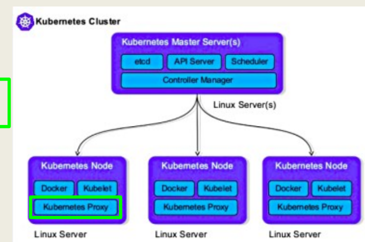
CONTAINER RUNTIME(S)

- Each node requires a container runtime to run containers
- Early versions had custom support for a limited number of container types, e.g. Docker
- Kubernetes now provides a standard Container Runtime Interface (CRI)
- CRI exposes a clean interface for 3rd party container runtimes to plug-in to
- Popular container runtimes: Docker, containerd, Kata

33

WORKER NODES

- Kubelet
- Container runtime (*Docker, etc.)*
- Kubernetes Proxy



34

KUBE-PROXY

- Runs on every node in the cluster
- Responsible for managing the cluster's networking
- Ensures each node obtains a unique IP address
- Implemented local IPTABLES and IPVS rules to route and load-balance traffic
- IPTABLES (ipv4) – enables configuration of IP packet filtering rules of the Linux kernel firewall
- IPVS – IP Virtual Server: provides transport-layer (layer 4) load balancing as part of the Linux kernel; Configured using ipvsadm tool in Linux

35

CORE KUBERNETES COMPONENTS

- Kubernetes DNS
- Pods
- Services

36

KUBERNETES DNS

- Every Kubernetes cluster has an internal DNS service
- Accessed with a static IP
- Hard-coded so that every container can find it
- Every service is registered with the DNS so that all components can find every Service on the cluster by **NAME**
- Is based on CoreDNS (https://coredns.io)

37

CORE KUBERNETES COMPONENTS

- Kubernetes DNS
- Pods
- Services

38

PODS

- Pod – atomic unit of deployment & scheduling in Kubernetes
- A Kubernetes Pod is defined to run a containerized application
- Kubernetes manages Pods, not individual containers
- Cannot run a container directly on Kubernetes
- All containers run through Pods
- Pod comes from "pod of whales"
- Docker logo shows a whale with containers stacked on top
- Whale represents the Docker engine that runs on a single host
- Pods encapsulate the definition of a single microservice for hosting purposes
- Pods can have a single container, or multiple containers if the service requires more than one

39

PODS - 2

- Examples of multi-container Pods:
  - Service meshes
  - Web containers with a helper container that pulls latest content
  - Containers with a tightly coupled log scraper or profiler
- YAML manifest files are used to provide a declarative description for how to run and manage a Pod
- To run a pod, POST a YAML to the API Server: "kubectl run <NAME>" where NAME is the service
- A Pod runs on a single node (host)
- Pods share:
  - Interprocess communication (IPC) namespace
  - Memory, Volumes, Network stack

40

PODS - 3

- Pods provide a "fenced" environment to run containers
- Provide a "sandbox"
- Only tightly coupled containers are deployed with a single pod
- Best practice: decouple individual containers to separate pods
  - *What is the best container composition into pods?* (1:1, 1:many)
- **Scaling**
  - Pods are the unit of scaling
  - Add and remove pods to scale up/down
  - Do not add containers to a pod, add pod instances
  - Pod instances can be scheduled on the same or different host
- **Atomic Operation**
  - Pods are either fully up and running their service (i.e. port open/exposed), or pods are down / offline

41

PODS - 4

- **Pod Lifecycle**
  - An application should not be tightly bound or dependent on a specific Pod instance
  - Pods are designed to fail and be replaced
  - Use of **service objects** in Kubernetes help decouple pods to offer resiliency upon failure
- **Deployments**
  - Higher level controllers often used to deploy pods
  - Controllers implement a controller and watch loop:
  - "Deployments" – offer scalability & rolling updates
  - "DaemonSets" – run instance of service on every cluster node
  - "StatefulSets" – used for stateful components
  - "CronJobs" – for short lived tasks that need to run at specified times

42

CORE KUBERNETES COMPONENTS

- Kubernetes DNS
- Pods
- Services

43

KUBERNETES "SERVICES"

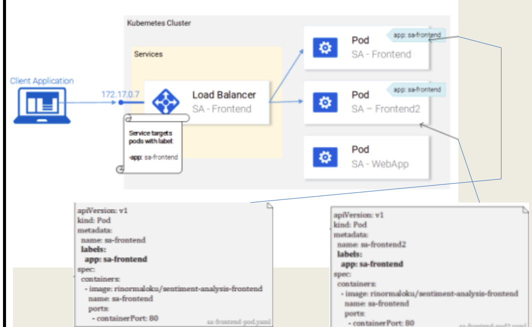
- Pods managed with "Deployments" or "DameonSets" controllers are automatically replaced when they die
  - This provides resiliency for the application
- **KEY IDEA**: Pods are unreliable
- **Services** provide reliability by acting as a "GATEWAY" to pods that implement the services
  - They underlying pods can change over time
  - The services endpoints remain and are always available
- Service objects provide an abstraction layer w/ a reliable name and load balancing of requests to a set of pods

44

SERVICES

- Provide reliable front-end with:
  - Stable DNS name
  - IP Address
  - Port
- Services do not posses application intelligence
- No support for application-layer host and path routing
- Services have a "label selector" which is a set of lables
- Requests/traffic is only sent to Pods with matching labels
- Services only send traffic to healthy Pods
- **KEY IDEA**: Services bring stable IP addresses and DNS names to unstable Pods

45



46

QUESTIONS

47