

## TCSS 562: SOFTWARE ENGINEERING FOR CLOUD COMPUTING

### Group Presentations - I

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




1

## OVERVIEW

- Tutorial 7 & 8 are posted
- Only 7 tutorials are required
  - Additional tutorials beyond 7 provide extra credit
- Term project paper template – to be posted over Thanksgiving

November 25, 2019
L16.2
TCSS562: Software Engineering for Cloud Computing [Fall 2019]  
 School of Engineering and Technology, University of Washington - Tacoma

2

## PRESENTATION TIPS

- Include slide numbers on slides
  - Helps with pacing, and when audience asks questions
- Do not read slides
- Use short phrases, not complete sentences on slides
  - Consider differences from presenting vs. writing a paper
- Goal is to inform the audience regarding the paper/system
  - Define acronyms on first use
  - Define and describe new systems (don't assume prior knowledge)
- Don't make audience perform your analysis/critique
  - Don't just copy and paste large tables and text from paper to slides
  - For tables, can use popup to focus attention to a certain part
- Best presentations synthesize key points of papers/systems
  - Goal is to maximize accessibility of content

November 25, 2019
L16.3
TCSS562: Software Engineering for Cloud Computing [Fall 2019]  
 School of Engineering and Technology, University of Washington - Tacoma

3

## GROUP PRESENTATION SCHEDULE

**Monday November 25**

1. Group 6 - Amazon Dynamo DB
2. Group 8 - Paper: Serverless computation with Open Lambda
3. Group 2 - Paper: A Programming Model and Middleware for High Throughput Serverless Computing Applications

**Monday December 2**

1. Group 9 - Paper: Performance comparison of container-based technologies for the Cloud
2. Group 10 - Paper: An Investigation of the Impact of Language Runtime on the Performance and Cost of Serverless Functions
3. Group 4 - Paper: Exploring Serverless Computing for Neural Network Training

**Wednesday December 4**

1. Group 1 - Paper: Performance evaluation of heterogeneous cloud functions
2. Group 7 - Amazon Cognito
3. Group 3 - Paper: Serverless computing - economics and architecture impact

November 25, 2019
L16.4
TCSS562: Software Engineering for Cloud Computing [Fall 2019]  
 School of Engineering and Technology, University of Washington - Tacoma

4

## TERM PROJECT DELIVERABLES

- Nine project teams
- Term project lightning presentations
  - Monday December 9<sup>th</sup> (5:50-7:50pm)
  - Takes place of final exam
  - Presentation length: 5 minutes + questions, total 8 minutes
  - Format and rubric coming soon
- Term project final paper and source code repository
  - Friday December 13 @ 11:59pm
  - Paper template to be provided

November 25, 2019
L16.5
TCSS562: Software Engineering for Cloud Computing [Fall 2019]  
 School of Engineering and Technology, University of Washington - Tacoma

5

## FEEDBACK FROM 11/20

- **Cloud terminology: what does "persist" mean?**
- Persist means to preserve
- Typically persistent is used in relation to storage (e.g. disks)
- EBS volumes are considered **persistent** storage
- Local NVME SSD disks on c5d instances are called **ephemeral**
  - Ephemeral means temporary

November 25, 2019
L16.6
TCSS562: Software Engineering for Cloud Computing [Fall 2019]  
 School of Engineering and Technology, University of Washington - Tacoma

6

## FEEDBACK - 2

- **http API -> gRPC ?**
- Docker can be used to host Google RPC APIs in a container
- Docker exposes http interface to a GO microservice
- This enables containerized microservices without a web application server/container
  - Less overhead...
- This is a nice blog on GO, gRPC, and Docker:
  - <https://medium.com/@matzhouse/go-grpc-and-docker-c1fb2ec8e9f0>
- Java microservices w/ Springboot can similarly be hosted by Docker containers w/o a server
  - <https://spring.io/guides/gs/spring-boot-docker/>

November 25, 2019 TCCS562: Software Engineering for Cloud Computing [Fall 2019]  
 School of Engineering and Technology, University of Washington - Tacoma L16.7

7

## GROUP PRESENTATIONS



November 25, 2019 TCCS562: Software Engineering for Cloud Computing [Fall 2019]  
 School of Engineering and Technology, University of Washington - Tacoma L16.8

8

## TUTORIAL #7 DOCKER, CGROUPS, RESOURCE ISOLATION



November 25, 2019 TCCS562: Software Engineering for Cloud Computing [Fall 2019]  
 School of Engineering and Technology, University of Washington - Tacoma L16.9

9

## DOCKER CLI

- **Docker CLI → Docker Engine (dockerd) → containerd → runc**
- Docker installation
- Docker file
- Docker run
- Docker ps
- Docker exec -it
- Docker stop

November 25, 2019 TCCS562: Software Engineering for Cloud Computing [Fall 2019]  
 School of Engineering and Technology, University of Washington - Tacoma L16.10

10

```

commands:
  attach      Attach local standard input, output, and error streams to a running container
  build       Build an image from a Dockerfile
  commit      Create a new image from a container's changes
  cp          Copy files/folders between a container and the local filesystem
  create      Create a new container
  deploy      Deploy a new stack or update an existing stack
  diff        Inspect changes to files or directories on a container's filesystem
  events      Get real time events from the server
  exec        Run a command in a running container
  export      Export a container's filesystem as a tar archive
  history     Show the history of an image
  images     List images
  import      Import the contents from a tarball to create a filesystem image
  info        Display system-wide information
  inspect     Return low-level information on docker objects
  kill        Kill one or more running containers
  load        Load an image from a tar archive or STDIN
  login       Log in to a Docker registry
  logout      Log out from a Docker registry
  logs        Fetch the logs of a container
  pause       Pause all processes within one or more containers
  port        List port mappings or a specific mapping for the container
  ps          List containers
  pull        Pull an image or a repository from a registry
  push        Push an image or a repository to a registry
  rename      Rename a container
  restart     Restart one or more containers
  rm          Remove one or more containers
  rmi         Remove one or more images
  run         Run a command in a new container
  save        Save one or more images to a tar archive (streamed to STDOUT by default)
  search      Search the Docker Hub for images
  start       Start one or more stopped containers
  stats       Display a live stream of container(s) resource usage statistics
  stop        Stop one or more running containers
  tag         Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE
  top         Display the running processes of a container
  unpause     Unpause all processes within one or more containers
  update      Update configuration of one or more containers
  version     Show the Docker version information
  wait        Block until one or more containers stop, then print their exit codes
    
```

November 25, 2019 TCCS562: Software Engineering for Cloud Computing [Fall 2019]  
 School of Engineering and Technology, University of Washington - Tacoma L16.12

11

## TUTORIAL 7

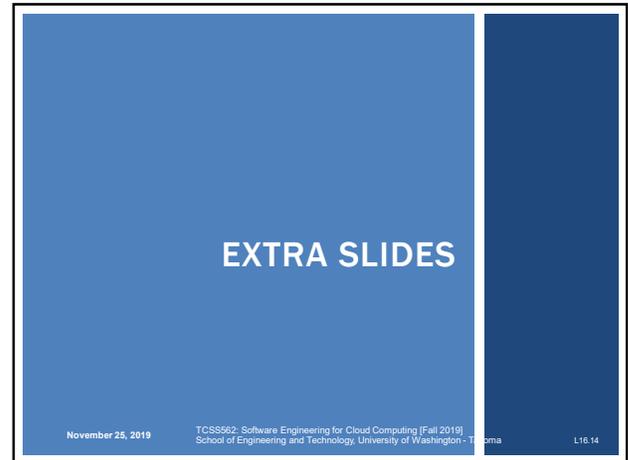
- Linux performance benchmarks
- stress-ng
- 100s of CPU, memory, disk, network stress tests
- Sysbench
- Used in tutorial for memory stress test

November 25, 2019 TCCS562: Software Engineering for Cloud Computing [Fall 2019]  
 School of Engineering and Technology, University of Washington - Tacoma L16.12

12



13



14