SJSU SAN JOSÉ STATE UNIVERSITY

College of Science · Computer Science

Topics in Cloud Computing CS 218

Fall 2026 In Person 3 Unit(s) 08/19/2026 to 12/07/2026 Modified 08/19/2026

Contact Information

Instructor	Handoyo Sutanto
Email	handoyo.sutanto@sjsu.edu

Email is the fastest way to reach me. I aim to respond within 24 hours on weekdays, 48 hours on weekends. Please put **CS 218** in the subject line.

Course Information

Lecture

Tuesday, Thursday, 12:00 PM to 1:15 PM, CL306

Office Hours

Tuesday, Thursday, 1:30 PM to 2:30 PM, DH282

Course Description and Requisites

Topics in cloud computing, including distributed system models, virtual machines, virtualization, cloud platform architectures (IaaS, PaaS, SaaS), service-oriented architectures, cloud programming and software environments, peer-to-peer computing, ubiquitous cloud, cloud security and trust management.

Prerequisite(s): CS 149 and Graduate standing. Allowed Declared Major: Computer Science, Bioinformatics, Data Science. Or instructor consent.

Letter Graded

Classroom Protocols

Per University Policy F15-12, attendance is not used as a criterion for grading. However, this course moves quickly and much of it is discussion of trade-offs rather than recitation of facts. If you miss a class, it is your responsibility to get notes from a classmate and to check Canvas for announcements.

- Laptops are welcome and expected for in-class exercises. Use them for the course.
- Class recordings, if any, will be announced in advance. University policy governs consent for the recording of class sessions; see the Syllabus Information page linked below.
- Ask questions during lecture. In a graduate seminar, a question that stops the class is usually doing everyone a favor.

Remote Session — Thursday, November 12

This course is scheduled as an in-person section and all meetings are held in CL 306, with one exception. **The November 12 session will be held remotely over Zoom at our regular meeting time, 12:00 – 1:15 PM.** The Zoom link will be posted to Canvas in advance.

This session is **synchronous** — attend it live as you would any other class meeting. The session will be recorded and the recording posted to Canvas afterward for anyone who cannot attend live, but the recording is a courtesy rather than a substitute, and there is no separate asynchronous version of the class.

Because the session is recorded, please be aware that your participation may be captured. If you prefer not to appear, you may keep your camera off and participate through chat; this will not disadvantage you in any way, as attendance and participation are not grading criteria in this course.

The reading, the material covered, and the Critique 2 deadline are unchanged.

Use of AI Tools

You will use AI tools in your career. Pretending otherwise would be poor preparation. This course's policy:

Permitted, with disclosure: labs, paper critiques, and all project work. If you use an AI tool, include a short note at the end of your submission stating which tool and what you used it for. Disclosure is required; it costs you nothing.

Prohibited: the midterm and final exams. These are closed-book, in-person, no-device.

What this means in practice. The graded work in this course is designed so that AI assistance helps you without doing the work for you. Labs are graded on explaining *your* measurements from *your* machine. Critiques require connecting a paper to a specific lecture you attended. The project requires a live demo you must defend in person. A model can help you write; it cannot attend class for you or run your container on your laptop.

Submitting AI-generated work as your own **without disclosure** is an academic integrity violation and will be handled under University Policy S07-2.

Program Information

Diversity Statement - At SJSU, it is important to create a safe learning environment where we can explore, learn, and grow together. We strive to build a diverse, equitable, inclusive culture that values, encourages, and supports students from all backgrounds and experiences.

Course Learning Outcomes (CLOs)

Upon successful completion of this course, students will be able to:

CLO 1 Compare distributed system models (cluster, grid, peer-to-peer, cloud) and explain the consistency, availability, and partition-tolerance trade-offs that constrain each.

CLO 2 Distinguish hardware virtualization from OS-level containerization and analyze the isolation, density, and performance trade-offs between them.

CLO 3 Classify a given cloud offering as IaaS, PaaS, or SaaS and justify the classification by locating the shared-responsibility boundary.

CLO 4 Decompose a monolithic workload into services and justify the resulting interface, state-placement, and failure-handling decisions.

CLO 5 Build and deploy a containerized multi-service application with reproducible, version-controlled

configuration.

CLO 6 Evaluate a cloud threat model and specify appropriate identity, isolation, and trust-management controls.

CLO 7 Assess peer-to-peer and edge/ubiquitous models against centralized cloud architectures for a stated workload.

Course Materials

Required — research papers. Roughly ten papers, all freely available through the SJSU King Library or the authors' sites. Links posted on Canvas. Readings are listed in the course schedule below.

Recommended — textbook. Thomas Erl, Ricardo Puttini, Zaigham Mahmood, *Cloud Computing: Concepts, Technology, Security & Architecture*, 2nd Edition, Pearson. Useful as a reference and a second explanation. You are not required to buy it; no graded item depends on owning it.

Required — software (all free and open source).

- A laptop capable of running a Linux VM or Docker
- Git
- Docker Desktop, Podman, or equivalent
- A local Kubernetes environment (kind, minikube, or Docker Desktop's built-in cluster)

If your hardware cannot run the above, contact me in the first week. There are workarounds and I would rather solve this in week 1 than week 9.

GitHub Account

A GitHub account on the free tier is required. The free tier is fully sufficient for every graded item in this course. You will not be asked to pay for anything, to start a trial, to enter payment information, or to upgrade a plan at any point. If any instruction in this course appears to require a paid GitHub feature, that is a mistake on my part — tell me and I will fix it.

If you already have a GitHub account, use it. Many of you do, and some of you have professional or open-source histories attached to those accounts. There is no requirement to create a separate one for this course.

How your work is distributed and submitted. Starter materials are published as public template repositories that you copy into your own account. Your working repository should be **private**. Do not make your coursework public while the semester is running — it makes academic integrity harder to manage for everyone, including students who have done nothing wrong.

For each lab and project milestone you will submit three things through Canvas:

1. Your report as a **PDF**
2. Your repository **URL**
3. The **commit SHA** representing the work you are submitting

The commit SHA is what fixes your submission in time. Once you submit it, later commits to your repository do not affect the version being graded, so you are free to keep working on your project after a milestone without worrying about disturbing what was already turned in.

Privacy. Canvas is the system of record for this course. Grades, feedback, and the official record of submission live there. GitHub is used only as a place to hold and version your code. Your account does not need to carry your legal name, and I will not evaluate anything about your GitHub profile, history, contribution graph, or activity outside this course. If you would prefer not to use GitHub at all, contact me in the first week — GitLab, Codeberg, or a submitted archive are all acceptable substitutes and none of them will cost you points.

A note on why. Version control is not a course topic and is not graded here. It is the medium the work arrives in, because that is how this work arrives everywhere else.

Course Requirements and Assignments

Success in this course is based on the expectation that students will spend, for each unit of credit, a minimum of 45 hours over the length of the course (normally three hours per unit per week) for instruction, preparation/studying, or course-related activities. For this 3-unit course, expect approximately **nine hours per week total**, including class time. More detail is in University Policy S12-3.

Labs (2 × 5% = 10%)

Short, hands-on, individually completed. Each is a measurement exercise, not a tutorial — you run something, record what happens, and explain the result.

- **Lab 1 — Failure and latency.** Introduce artificial latency and partition into a small distributed key-value setup; record and explain the observed behavior against the consistency model you expected. (CLO 1)
- **Lab 2 — Isolation and density.** Run the same workload in a VM and in a container. Measure startup time, memory overhead, and isolation boundary. Explain the difference in terms of the underlying mechanism. (CLO 2, CLO 5)

Labs are graded on the **explanation of your own measurements**, not on getting a canonical number. Two students with different hardware should submit different numbers and both can earn full credit.

Paper Critiques (2 × 5% = 10%)

Maximum **500 words each**, submitted as PDF. Each critique must:

1. State the paper's central claim in one sentence.
2. Identify one assumption the paper makes that has weakened or broken since publication, and say what changed.
3. Connect the paper to a **specific claim made in a specific lecture** (cite the date), and say whether it supports or complicates that claim.

Item 3 is the part that matters. A summary of the paper alone earns at most half credit.

Course Project (20%)

Teams of **up to 3**. You will design, build, and defend a multi-service application that runs entirely on a local container runtime.

Milestone	Weight	Due
Proposal (1 page: workload, services, failure hypothesis)	3%	Thu, Oct 1
Design review (architecture + trade-off justification)	5%	Thu, Oct 29
Final demo + report	12%	Dec 1 / Dec 3 (demo), report due Thu, Dec 3

The final deliverable must include:

- A running system, demonstrated live in class
- **A deliberate failure injection** — kill a service, partition the network, exhaust a resource — and evidence of how the system responded
- A written architecture justification (max 4 pages) explaining what you chose, what you rejected, and why
- A short security section identifying your trust boundaries (CLO 6)
- A **delivery and cost section** (max 1 page): classify how your system would be delivered to a consumer of it (IaaS, PaaS, or SaaS) and defend the classification by locating the responsibility boundary; state where your tenancy boundary sits and what you would have to change to serve a second tenant; and identify which single resource would dominate your operating cost at 10× your demonstrated load, using your own measurements as evidence (CLO 3, CLO 4)

Teams are graded as a unit. If a team has a contribution problem, tell me before the final milestone, not after.

Midterm Exam (25%)

Thursday, October 15, in class. Closed book, closed notes, no electronic devices. Covers material through Oct 8.

Final Exam (35%)

Tuesday, December 15, 10:45 AM – 12:45 PM, in our regular classroom unless otherwise announced.

This time is assigned by the University's Culminating Activities and Final Examinations schedule for Group II classes in the 11:30 AM–12:25 PM start band. It is not negotiable and cannot be taken early. Do not book travel before December 15.

The final is comprehensive, weighted toward material after the midterm. Closed book, closed notes, no electronic devices.

Late Work

- **Labs and paper critiques:** 10% penalty per day late, up to five days; not accepted after five days.
- **Project milestones:** no late submissions. The demo happens on the scheduled day.
- **Exams:** makeups only for documented emergencies, with notice to me as far in advance as circumstances allow.

Extra Credit and Curving

There is no extra credit in this course. Final letter grades are assigned from the scale below; I reserve the right to adjust boundaries downward (in students' favor) but never upward.

✓ Grading Information

Criteria

Component	Weight
Labs (2 × 5%)	10%
Paper Critiques (2 × 5%)	10%
Course Project (proposal 3% / design review 5% / demo + report 12%)	20%
Midterm Exam	25%
Final Exam	35%
Total	100%

Note that attendance is not a grading criterion, per University Policy F15-12. There is no participation grade.

Breakdown

Percentage	Grade
97% and above	A+
93 – 96.99%	A
90 – 92.99%	A−
87 – 89.99%	B+
83 – 86.99%	B
80 – 82.99%	B−
77 – 79.99%	C+
73 – 76.99%	C
70 – 72.99%	C−
67 – 69.99%	D+
63 – 66.99%	D
60 – 62.99%	D−
Below 60%	F

Graduate students should note their program's minimum grade requirements for coursework applied toward a degree.

University Policies

Per University Policy S16-9 (PDF) (http://www.sjsu.edu/senate/docs/S16-9.pdf), relevant university policy concerning all courses, such as student responsibilities, academic integrity, accommodations, dropping and adding, consent for recording of class, etc. and available student services (e.g. learning assistance, counseling, and other resources) are listed on the Syllabus Information (https://www.sjsu.edu/curriculum/courses/syllabus-info.php) web page. Make sure to visit this page to review and be aware of these university policies and resources.

Course Schedule

This schedule is tentative and subject to change with fair notice. Changes will be announced in class and posted to Canvas.

Erl chapter numbers below are suggested references only — the textbook is recommended, not required, and no graded item depends on it.

Wk	Date	Topic	Readings	Due
1	Thu 8/20	Course intro; what "cloud" actually means; the economics of elasticity	Armbrust et al., *A View of Cloud Computing*	
2	Tue 8/25	Distributed system models: clusters, grids, P2P, clouds	Barroso & Hölzle, *The Datacenter as a Computer* (Ch. 1–2)	
2	Thu 8/27	Failure models, partial failure, the fallacies of distributed computing	Slides	
3	Tue 9/1	Latency, tail latency, and why p99 is the number that matters	Dean & Barroso, *The Tail at Scale*	
3	Thu 9/3	Consistency models; CAP and PACELC	Vogels, *Eventually Consistent*	
4	Tue 9/8	Time, ordering, and coordination	Slides	
4	Thu 9/10	Virtualization I: hypervisors, Type 1 vs Type 2, CPU and memory virtualization	Barham et al., *Xen and the Art of Virtualization*; Erl Ch. 5	Lab 1 due
5	Tue 9/15	Virtualization II: I/O virtualization, live migration, resource pooling	Erl Ch. 5	
5	Thu 9/17	Containers: namespaces, cgroups, images and layers	Felter et al., *An Updated Performance Comparison of VMs and Linux Containers*; Erl Ch. 6	
6	Tue 9/22	Containers vs. VMs: isolation boundaries and where they leak	Slides	
6	Thu 9/24	Orchestration as a control plane; declarative reconciliation	Burns et al., *Borg, Omega, and Kubernetes*	Critique 1 due
7	Tue 9/29	Cluster scheduling and bin-packing in practice	Verma et al., *Large-scale cluster management at Google with Borg*	

Wk	Date	Topic	Readings	Due
7	Thu 10/1	Cloud platform architectures: IaaS, PaaS, SaaS as delivery models; the shared responsibility boundary and who pays for what	Erl Ch. 4	Project proposal due
8	Tue 10/6	Serverless and FaaS: what the abstraction hides	Jonas et al., *A Berkeley View on Serverless Computing*	
8	Thu 10/8	Deployment models; multi-tenancy as isolation boundary and as cost structure; the noisy neighbor	Erl Ch. 4	Lab 2 due
9	Tue 10/13	Midterm review		
9	Thu 10/15	**MIDTERM EXAM** (in class)		
10	Tue 10/20	Service-oriented architecture: services, boundaries, and coupling	Erl Ch. 4	
10	Thu 10/22	APIs in practice: REST, gRPC, idempotency, retries, backpressure	Slides	
11	Tue 10/27	Cloud programming environments: infrastructure as code, configuration as data	Slides	
11	Thu 10/29	CI/CD, immutable infrastructure, deployment strategies	Slides	Design review due
12	Tue 11/3	Cloud storage: object, block, and file; durability; data gravity, egress, and lock-in as architectural forces	Erl Ch. 9; Slides	
12	Thu 11/5	Replication and quorums	DeCandia et al., *Dynamo*	
13	Tue 11/10	Elasticity and autoscaling; pay-per-use as mechanism; capacity planning and unit cost at scale	Hu et al., *Characterization of Large Language Model Development in the Datacenter*; Erl Ch. 8	
13	Thu 11/12	Observability, SLIs/SLOs, and error budgets — **remote session, Zoom (see note below)**	Beyer et al., *Site Reliability Engineering* (Ch. 4, free online)	Critique 2 due
14	Tue 11/17	Cloud security: threat model, isolation failures, side channels	Ristenpart et al., *Hey, You, Get Off of My Cloud*; Erl Ch. 7	
14	Thu 11/19	Trust management: identity, secrets, supply chain, confidential computing	Erl Ch. 10–11	
15	Tue 11/24	Peer-to-peer computing: DHTs, Chord, Kademlia; where P2P beats centralized	Stoica et al., *Chord*	
15	Thu 11/26	**No class — Thanksgiving holiday**		

Wk	Date	Topic	Readings	Due
16	Tue 12/1	Ubiquitous cloud and edge computing; project demos begin	Satyanarayanan, *The Emergence of Edge Computing*	Demos
16	Thu 12/3	Project demos; course synthesis and final review		Demos; project report due
—	Tue 12/15	FINAL EXAM, 10:45 AM – 12:45 PM		

Total class meetings: 30.