



# UC San Diego CloudLabs

## Flexible, Cost-Minimizing Remote Computer Labs in Record Time

Larry L. Sautter Award Proposal 2020

*submitted by:* **Valerie E. Polichar, Ph.D.**

Director of Academic Technology Services, UC San Diego Information Technology Services  
858-534-1859 | [vpolichar@ucsd.edu](mailto:vpolichar@ucsd.edu)

### Project leads:

#### Sean Osterthaler

Manager, Educational Infrastructure Systems  
Architecture, Infrastructure and Innovation  
UC San Diego Information Technology Services

#### Adam Tilghman

Architect/Analyst  
Academic Technology Services  
UC San Diego Information Technology Service

### Implementation Team:

#### Wade Blomgren

#### Brian Hamilton

#### Sean Osterthaler

#### Scott Calman

#### Danny Kovacs

#### Brian Park

#### Caleb Crawford

#### Aaron McMahan

#### Miguel Rodriguez

#### Matthew Fedder

#### Mike Oliva

#### Adam Tilghman

### Executive/Management Support:

#### Vince Kellen, CIO

#### Valerie E. Polichar

#### Dan Suchy

#### Pierre Ouillet, CFO

#### Brian DeMeulle

#### Paul Jamason

#### Elizabeth Simmons, Exec. Vice Chancellor

#### Rebecca Sipili

#### Katherine Collins

When COVID-19 caused a pivot to remote instruction for Spring quarter with little notice, students lost access to 2200 on-campus computer lab seats they needed to complete coursework.

Our teams sprang into action and built **CloudLabs: a portfolio of services — easy to access with a customized interface — providing remote access to 264 apps on three operating systems for 114 courses** (so far), in under three weeks.

The team built it to be flexible, to accommodate challenges with vendors and licenses; and easily adjustable, so that on an ongoing basis we can minimize cost while maximizing student experience.

The service will meet student needs during lockdown and beyond — reducing wait times for lab access and allowing students near and far access to the same quality of instruction.

### The challenge: How can students use licensed software without on-campus labs?

UC San Diego's Educational Technology teams support well over 300 pieces of software in 160 physical lab spaces on campus (2200 seats). In a normal quarter, these are heavily used, with 15,000 logins in a typical week.

In most cases, our software licenses don't permit a student to copy the software to their home machine. When it *is* allowed, many students' home computers aren't fast or powerful enough to successfully run that software – think graphics rendering, chemical modeling, mechanical engineering in 3D.

This broad dependence on advanced hardware and software extends to hundreds of courses each year. So when the campus abruptly moved to remote instruction, it wasn't clear how students were going to get their coursework done.

UC San Diego *had* a virtual lab — an older on-premises server cluster running a version of VMware Horizon View for remote desktop access. Because of its age, it was really only suitable for the least demanding of applications, and students found it so slow as to be only minimally usable. Its software versions were out of date and out of support. It was clear it wasn't up to the task.

Finals week of Winter quarter 2020, everyone was sent home and we realized it was critical to jump to the cloud — but how to do so in a way that would work for most of our needs?

### **Complications: Licensing, cost, ease of use, and vendor maturity**

The teams had been discussing cloud-based options for a while, but there were several problems: cost was high, faculty were reluctant to embrace cloud alternatives, and, as a relatively new service, vendor maturity was low. It was unclear which option would be both palatable and dependable. Small pilots of AWS AppStream and Apporto that had been running during Winter quarter were not yet complete. It seemed as though some software might work well in one cloud environment, other software in another. We didn't have much information on which to make a decision, and multiple teams within IT Services would have to work together on whatever we selected.

Thanks to support from the CFO, the Executive Vice Chancellor's office, and our CIO, we had permission to spend money that wasn't in our budget in this emergency situation. But we were well aware that even emergency money is limited — and the less we'd spend of it, the better. Further, if we wanted to continue this experiment beyond a quarter or two, we needed it to be affordable.

What we did know: Of the cloud products, AWS AppStream had been around for longer, and was a larger company, but the product looked likely to be expensive and had a less graceful interface. Apporto seemed like it might be easier for students to use, and was likely to be somewhat cheaper, but the company was very young and small, and we were concerned about their ability to deliver under what was likely to be massive demand. We could upgrade our old, on-premises Horizon View virtual lab with a minimum of expense, but it still wasn't going to work for heavy CPU- or GPU-demanding software. Complicating things yet more, some of our software was licensed to use in a cloud or remote desktop environment; some was not, or would only work in one or the other.

None of these solutions would work for MacOS or Linux. For those, remote desktop was our only option — making a direct connection to an on-premises computer in a now-locked lab. But this meant we were limited by the number of computers we had on campus, so the same old wait times would hold. Worse, it wasn't going to be fast enough for all the software students needed to run.

### **The solution: A portfolio of solutions that can be snapped in and out**

We made the decision to go for a portfolio model: Using AWS AppStream *and* Apporto, to reduce cost and risk; upgrading our on-premises server, a cheap way of getting students access to a lot of less-processor-intensive applications; and running remote desktop over Apache Guacamole for our Mac and Linux users. (Because Adobe Creative Cloud wasn't going to work well over remote desktop, we encouraged students to take advantage of Adobe's free download for Spring quarter.)

Short on time and person-hours, we engaged Trace3 via contract to help us get our local Horizon View server up to snuff. While our staff upgraded VMware ESXi and vCenter and changed out blades on campus for more powerful ones that had been freed up by recent upgrades elsewhere,

Trace3 did a greenfield config of the Horizon View software components, SSL certificates, and Trend Micro Hypervisor-level Anti-malware/Antivirus. This “extra pair of hands” made it possible to make our deadline and have the system revamped and running by the start of Spring quarter.

We got contracts started with AppStream and Apporto, and began the process of creating desktop environments on both services. Members of several teams together analyzed our hundreds of Windows applications, determining which would be used by classes during Spring quarter. We split high-need applications by discipline over the two cloud providers, and arranged to make the rest available on Horizon View.

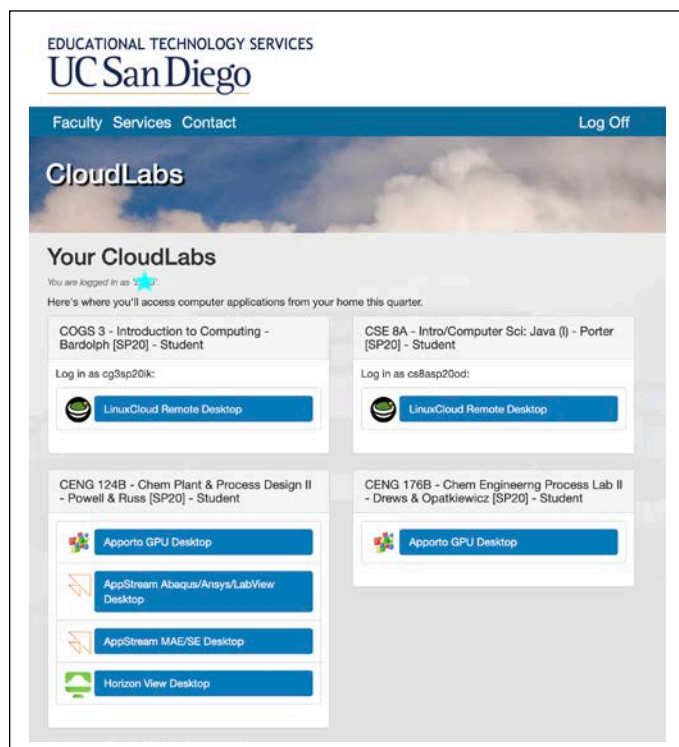
We very soon discovered that our concerns about both products were warranted. Many students and faculty preferred the Apporto environment, but that vendor was not able to get software up and running as quickly as we were able to install it in AppStream. Some license server issues persisted with Apporto after the quarter started, so more classes started in the AppStream environment than originally planned. Apporto worked with our team throughout the quarter to resolve the license server issues so that we were able to accommodate some classes there in Spring quarter; we expect to accommodate more in the Apporto environment in Summer. The flexibility to swap environments quickly for a given class turned out to be critical to course delivery.

At the same time as all of the above, we set up Apache Guacamole to provide remote desktop access (via RDP, VNC, SSH or Kubernetes) for MacOS and Linux in a secure, clientless way. Needed software was already installed on the lab machines. Students connect from the Web, authenticate, and get a familiar desktop environment.

## Clear, personalized access for students and faculty

If we’d simply put up links to all five environments, students would have spent hours trying to find the software they needed and tracing through a maze of different interfaces. So the critical piece to make **CloudLabs** a single service was our innovative entry point — a Web site (<https://cloudlabs.ucsd.edu/>, *UCSD authentication required*) where students would authenticate and be served up a custom menu, showing all their classes and providing links to get to the tools for each.

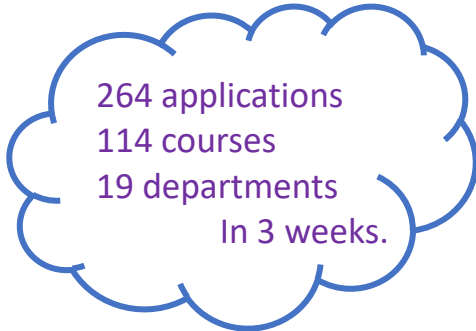
In the image at right, a student has Apache Guacamole links for her Linux classes in Cog Sci and CSE, Apporto, AppStream and Horizon View links for her Chemical Engineering classes. The Linux links let her know the course-based login name to use for access. For her Chem Eng classes, she has different desktops for the different packages of tools she’ll want to access.



C# ASP.Net was used to create the Web page experience you see above. The student or faculty member’s Active Directory username is used to query a RESTful API that provides student course

and lab information via JSON. We used a standard campus decorator for the Web page and Bootstrap framework components for the customization, making sure the layout was responsive for access on any size of screen or tablet.

## The impact



When we began, we hoped we could get at least the 50 most-used applications up and running for Spring quarter. In the end, the team loaded 264 applications and were able to meet the needs of 114 classes in 19 campus departments, both undergraduate and graduate. We are seeing several hundred students connecting each day. These classes represent a total enrollment of over 10,000 (many students may be enrolled in more than one course, so this is greater than the number of individual students) — a lot of courses that were able to be offered, and credits that students were able to achieve, in part thanks to CloudLabs!

In addition to the students who require CloudLabs for their coursework, we've had a stream of requests from grad students, undergraduate student organizations, and campus computing clubs anxious to also get access to what is increasingly perceived as a desirable environment.

We initially estimated a cost of \$400K to get CloudLabs up and running for Spring quarter, with an annual recurring cost of about \$1.3M/year. However, various efficiencies allowed us to reduce the Spring Quarter cost to about \$75K. Although we expect usage to grow, and Spring Quarter costs reflect deep discounts that will not be regularly available to us, we now think CloudLabs may cost us as little as \$400-500K *per year*.

Although we don't think this will eliminate the need to have on-campus labs, we do expect it will reduce the need for growth. Over time, as students adjust to remote access and appreciate the ability to do coursework from wherever they like, whenever they like, we believe we can gradually reduce the number of on-campus lab seats, a key goal at a time when campus space is at a premium. The reduced cost of refreshing on-campus equipment can then be applied towards any future growth in CloudLabs. We see CloudLabs, too, as increasing accessibility to students with disabilities and to our remote population.

## Ongoing challenges and next steps

Our solution was graceful and usable for courses in Spring quarter, and faculty and students expressed their gratitude for the options available. But a number of challenges lie ahead:

*Additional applications.* We would still like to uplift as many as possible of our remaining applications to a cloud environment. In some cases, licensing simply won't allow this, and we may have to discuss alternative options with faculty, but with the ability to use remote desktop as well as cloud, we hope to achieve installation of most.

*Adobe Creative Cloud.* Our current licenses for this product don't allow use in a third-party cloud environment, and remote desktop performance is unusable for many applications. Licensing for individual student use is very expensive. We're currently exploring alternatives for certain applications within the Suite, in order to minimize student and/or university cost, and exploring several alternative remote desktop applications to Guacamole in case one has more impressive performance.

*Overall cost.* We will need a bit more exploration and tweaking to find the best balance between our Horizon View, Apporto, and AppStream environments that balances usability, performance, user interface and minimal cost. Without the rush to rollout that Spring demanded, we'll be weighing our allocations over the summer to prepare for a reasonably-priced Fall.

*On-premises instruments and tools.* Forty-four percent of our total on-campus lab seats are in locations where they are connected or adjacent to instrumentation, tools (such as 3D scanners and printers), equipment or wet lab environments. We can't replicate that in the cloud. For classes that depend on this adjacency, there are no immediate easy solutions, but we'll be looking at the possibilities of remote instrument management for the future — and for potential use by online classes.

*Remote desktop.* Apache Guacamole works, but we're exploring other options to see if we can find one with better performance for more demanding apps. If we do, it'll be easy to snap into place!

*Replacing Horizon View.* Our upgrades to the on-premises server save us money in the short term, but in the long term, we'd like to avoid refreshing that hardware a second time. We plan to move all apps to a cloud option before that hardware ages out. This will increase the recurring cost of CloudLabs, but we expect cost savings in terms of managing and refreshing on-campus hardware.

## **Flexibility — even beyond the walls of UC San Diego**

A cloud-based environment means agility. We can spin up additional environments with relative ease. If we need more seats in a given environment, it's no problem to expand. Fewer seats in Winter quarter than in Fall? No problem. The ability to scale as needed is something we couldn't achieve with physical lab seats, where the count is always a compromise between peak need and available space and funding.

Other campuses can take advantage of our model. With a front end that draws on the student's course enrollments and a table that maps courses to lab allocations, it's easy to snap in and snap out different options, whether your campus embraces one or many. You can build a Web site that talks to your particular campus' student enrollment information and lays out options based on the cloud and RDP selections your campus opts to make available. When and if you change options or vendors, there's no need to re-brand or re-develop your interface. We are happy to share our code and methodology.

We expect to snap in on-premises lab allocations when they return to availability, too. CloudLabs provides one-stop shopping for a student to access the IT resources they need to complete coursework — a small but meaningful way to simplify the University experience in these sometimes confusing and overwhelming times.

## **Acknowledgements**

We are grateful for the vision and support of UC San Diego's CFO, Pierre Ouillet; our CIO, Vince Kellen; and our Executive Vice Chancellor, Elizabeth Simmons. They all saw the potential in CloudLabs and put their support — and campus dollars — behind this effort.

The team is also grateful for the patience, flexibility, partnership, and creativity of UC San Diego faculty and instructional assistants who participated in our first term with CloudLabs. In the midst of a fast pivot to remote teaching — when so many of them were already pressured to learn new approaches and software tools and to rethink all their lectures with little notice — they worked with us as partners, helping us discover what worked best for them and their students.