burwood group, inc. Blog Post | Article | Case Study
By Debby Wadsworth
https://www.burwood.com/case-studies-archive/brown-infrastructure-as-code

Brown University: Scaling Classroom Applications with Infrastructure as Code

As a world-renown Ivy League research university, Brown University maintains a reputation for excellence, creativity, and innovation in higher education. This drive extends to their technology infrastructure. When Brown recognized scaling issues with a critical classroom application, they engaged Burwood Group to help with creating an automated production environment.

From Single Classroom to Scaling Across Campus

The Brown University Center for Computation and Visualization (CCV) used JupyterHub notebooks for everyday classroom work. Data Scientist and Professor Dr. Isabel Restrepo wanted to scale adoption to more classes, so she asked the Brown Central IT team for help with deployment, management, and improving upon the current-state student experience:

- At times, students experienced long login times and delays in opening their code work
- Central IT had no visibility into the classrooms, and more specifically how many resources the students were consuming; limiting their ability to ensure uptime and availability
- New class deployment took longer than desired due to task coordination across multiple teams (allocating IP address, creating DNS records, testing, and more).

Brown's Central IT team used Kubernetes to host JupyterHub project containers in Google Cloud. The reliability challenges indicated there were opportunities to improve the foundational Kubernetes architecture. And with Google Cloud's inherent capabilities for supporting automation, Central IT knew they could create a larger, scaled environment. Improved performance would have a wide-ranging, positive impact to multiple faculty and student groups.

Brown had previously worked with Burwood Group for help with Google Cloud onboarding best practices. With their expertise in cloud architecture and automation, Brown knew that Burwood would be the right partner to help with executing their expanded vision.

Prioritizing To-Do's for Maximum Impact

With a new semester approaching, Brown teamed with Burwood to use the upcoming month-long break to accomplish as much work as possible to facilitate a seamless experience for new classes requiring JupyterHub. Together, they identified four critical priorities:

1. Re-configure the cloud architecture to improve performance, reliability, and availability
2. Automate the infrastructure steps required to support application deployment
3. Automate classroom deployments and then scale up/down, based on demand
4. Integrate the environment into the technical support structure (Brown's Service Desk)

The project team knew four weeks would be a challenging timeframe for these tasks. With better functionality being the number one priority, they decided to focus on items one, two, and three during the class break, and tackle the support integration when class resumed.

The Fast Journey to Automation

KUBERNETES ARCHITECTURE RE-CONFIGURATION

First, the Central IT team worked with Burwood to set up their Kubernetes environment for success. Brown's existing Kubernetes cluster was a single zone cluster, which presented availability issues with automatic updates and disruptions for classrooms.

Burwood consultants relied on best practice knowledge to re-configure the Kubernetes environment, with guidance from Central IT and Professor Restrepo on Brown's unique requirements. Together, they implemented six critical changes:

1. The default Google Kubernetes Engine (GKE) configuration is a zonal cluster with one master node. The master node provides APIs and manages worker nodes. The team reconfigured the environment from a zonal to regional cluster, which introduced multiple master nodes to achieve high availability.
2. The regional cluster configuration distributed resources across multiple zones and increased the overall node count, great for resiliency, but not great for cost. The team configured worker nodes to be elastic, scaling up during peak usage and vice versa during slow times. This helped control costs.
3. Login complaints peaked at the class start time when many students were logging on at once. To increase available resources, they "pre-warmed" the environment by scheduling additional nodes to start just before class.
4. The team scheduled scale-downs to control costs, including worker nodes pinned to specific zones and dual regional master controllers for redundancy.
5. The team found new ways to interact with existing infrastructure by requesting and creating new DNS records during the build process, saving approval time.
6. To monitor the environment, the project team hooked the Kubernetes cluster into Central IT's Service Desk. This increased eyes on system health.

IMPLEMENTING INFRASTRUCTURE AS CODE

An advantage of building in the public cloud is the potential for automation. With the Kubernetes and Google Cloud foundation in place, Brown and Burwood Group turned their attention to Terraform, an infrastructure as code automation framework.

Infrastructure as code automates the provisioning of infrastructure. For example: imagine the Brown CCV department needs to spin up a new classroom of students. The professor sends a request to the central IT team asking for a new project.

Without a tool like Terraform, these requests were fulfilled by sequentially executing a number of scripts or going through different user interfaces to manually configure the required resources. With Terraform, automated infrastructure as code does all this work for the team. Terraforms framework allows them to deploy projects, virtual machines, containers, and DNS records; and it automatically documents the deployed resources, ensuring all deployments and changes are recorded.

> The project team worked in tandem to create the new, automated environment. Burwood drafted approximately 1,000 lines of code to form the foundation. Then, with their specific knowledge of the application and workflows, Brown's team modified the code for production use.

The code included classroom spin-up and spin-down. Central IT could automatically manage the deployment and scaling of various room sizes. And at the end of a semester, they could run a single command to close out a classroom, free up resources, and document the process.

Testing the user capacity and code performance prior to semester go-live was critical. The team used Locust, an open-source load testing tool, to stress test the Kubernetes cluster for up to 400 classroom users (even though current class size was only 100 students). The new environment operated as they had hoped, and it was time to move into production.

Ready Or Not: The New Semester Begins

At the start of the new semester, the Central IT team received some surprising news from the CCV department. A new JupyterHub class had hit a record student registration high of 290 students. Having completed the appropriate stress testing, Central IT knew their system could handle it.

IMPROVED PRODUCTIVITY, SCALING, AND MORE

The re-configured Kubernetes cluster and Terraform infrastructure as code deployment had been completed in just a few short weeks between the fall and spring semesters. And a few weeks into the spring semester, it was clear that the team's efforts had paid off. Central IT and CCV faculty were thrilled with the outcomes:

- Classroom provisioning time was reduced to 10 minutes. New semesters were previously associated with a lot of manual effort to spin up new workloads. The streamlined provisioning process eliminated this concern.
- New classroom requests no longer required manual administration and set-up. Terraform infrastructure as code produced automatic classroom spin-up and take down, with commands so simple that anyone could now to create the classes. This not only freed up time, but fixed resource scaling issues.
- Automated scaling was aligned with actual resource demand, reducing Google Cloud and networking costs. The new system accounted for peak hours from 7am-7pm, and massive resource scale down after hours.

> *Before we started this project, we had problems like monolithic code bases and pricey Google Cloud bills. Now, we've created a modular code base that deploys a variety of classes with cost predictability. And we're moving towards a DevOps culture and methodology.*

What's Next? Dashboards, Use Cases

With automated classroom deployments under control, the Brown Central IT team has turned their attention towards improving the Service Desk integration. They're creating new dashboards for helpdesk staff and CCV faculty to monitor system performance and troubleshoot any issues. So far, they've had

just one challenge with optimizing these dashboards: the system is working so well, there's very little error data.

With monitoring in place, Central IT can confidently expand their infrastructure as code to serve more university classrooms and use cases. And Brown's students can increasingly rely on technology as an accelerator, not a hurdle, to their education.