Doug Black

9197404420
doug@dougblack.io
dougblack.io
github.com/dougblack

PROFILE

Articulate, enthusiastic lead engineer with strong experience in system architecture, data infrastructure, and platform engineering. Passionate about building tools, teams, and systems to increase resiliency and productivity.

EDUCATION

Georgia Institute of Technology   August 2009—May 2013

B.S. Computer Science

SKILLS

Languages: Python, Java, Scala, Go, Elixir, SQL, Javascript
Platforms: Amazon Web Services, Google Cloud Platform, *nix

EXPERIENCE

Discord   San Francisco, CA and Denver, CO
Senior Software Engineer   December 2017—January 2021
Data Platform Team

- First senior engineer hired to the team, responsible for scaling up and enhancing Discord's ability to ingest, store, and analyze 30MM+ events per second used to monitor key app metrics.

- Designed and shipped a data aliasing system written in Scala that allowed Discord to reach GDPR data privacy compliance in just four months. This system was in the real-time path of event ingestion—instantly becoming one of the most critical systems at the company—and the migration was a success, resulting in zero data platform downtime.

- Led a project to take ownership of, and add enhancements to, a core identity system at the company written in Elixir. This system had not only become crucial to the data platform, but was a critical real time identity system for almost all Discord traffic. The ownership transfer and new features were successful.

- Reviewed and added several platform technologies that greatly improved developer productivity (like Bazel and Terraform) to our team that later became standard technologies for all engineering teams at the company.

- Designed and shipped a real-time monitoring sidecar process written in Go that was used to monitor for stale or misbehaving jobs.

- Designed and shipped a data ingestion service in Python that pulled vast amounts of data into the data platform for analysis and review. It pulled from many different storage technologies, including Cassandra, ScyllaDB, Postgres, and TokuMX and we were able to keep it fast as Discord grew from

30MM to 140MM+ users.

- Mentored junior engineers and conducted close to one hundred technical interviews. Constantly advocated for team process improvements and heavily contributed to prioritization. Architected the team's on-call and incident response runbooks. One of the first employees to pioneer remote work (pre-pandemic) at the company.

Twilio   San Francisco, CA
Lead Software Engineer   October 2016—December, 2017
API Team

- Led a team of four engineers, responsible for the scalability and uptime of Twilio's REST API proxy. Scaled the Twilio API to 3B requests/week while maintaining 100% uptime and 99.999% success rate.

- Designed and built continuous delivery for the API proxy. This increased the API deployment frequency from two deployments a week to multiple deployments every day and reduced the time required to expose new APIs from multiple weeks to several hours.

- Led sprint planning, sprint retrospectives, and quarterly roadmap planning. Handled prioritization, customer escalation, and worked with engineering and product leaders daily to orchestrate successful product ships. Mentored team members and taught engineering best practices.

Senior Software Engineer   December 2015—October 2016
API Team

- Introduced request tracing to Twilio, which drastically simplified service debugging and incident response and empowered customer support teams to have more meaningful engagements with customers.

- Ran engineering-wide sprint demos, encouraged engineers from all teams and experience levels to share their work with others.

- Designed and built Twilio's next-generation deployments system, which quickly gained wide adoption and became the primary way to deploy hundreds of different microservices across the company.

Software Engineer   June 2013—December 2015
API Team

- Responsible for exposing new Twilio APIs and adding various enhancements to the Twilio REST API and Twilio SDKs.

- Created command line tool that automated many different parts of everyday Twilio developer life. To this day, tool enjoys 100% adoption and is used daily by every engineer at Twilio and is even a crucial part of incident response

- Created service to automatically identify and de-provision unused infrastructure. To date, this service has saved Twilio over $500k in cloud computing costs.