





# CRISTIAN GONZALES

SOFTWARE ENGINEER AT NORTHROP GRUMMAN

 <https://bit.ly/ee7W>

 <https://bit.ly/2McyUX>

 <https://bit.ly/2vA649>

 [xcristian.gonzales@gmail.com](mailto:xcristian.gonzales@gmail.com)

## EXPERIENCE

---

### Northrop Grumman

*Associate Software Engineer*

July 2019 – Present

- Developed an API to filter JMS messages from a legacy JMS service, using Google Guice injections to differentiate between different instances of the service and JAXB for object serialization to send messages across the wire.
- Refactored database schemas and their associated JPA mappings, along with protocol buffer definitions, to differentiate between different satellite missions for a server-side event message syslog implementation, where the mission identifier was injected using Google Guice.
- Developed Jenkins CI/CD pipelines to streamline build processes in different target environments—namely, Linux, Solaris, and Windows—decreasing the time of the build and release process from four days to two hours.
- Developed shell and Perl scripts to streamline build and release processes, along with artifact repository management and Jenkins job versioning, tailored for both humans outside of a CI/CD pipeline and a Jenkins daemon.
- Promoted Agile methodologies, as a Scrum Master, by leading daily Scrum meetings, retrospectives, and planning in a program-wide effort to migrate away from a waterfall model.
- Automated deployment for builds for ad hoc projects and ported build systems from Apache Ant to Apache Maven.

### Northrop Grumman

*Software Engineering Intern*

June 2017 – August 2017

- Interfaced with NASA's GMSEC API to build a visualization tool which tracked health statuses of satellites.
- Ported payload data, via web sockets, across multiple Northrop Grumman domains/applications while leveraging code reuse through dependency injections, using Google Guice.

## PROJECTS

---

### ShardedKVS

*A scalable, sharded Docker network of nodes that is fault tolerant, highly available, and eventually consistent.*

- Nodes were designed using a RESTful API interface (Python Flask), so clients may use the API to add and delete nodes, and keys, from the KVS.
- Favors availability over strong consistency, per CAP theorem, so eventual consistency is guaranteed in light of network partitions, with a property of bounded staleness.
- Uses vector clocks as an implementation of causal order to resolve conflicting keys, and consistent hashing to dynamically and uniformly distribute unique keys, across all shards.

## SKILLS

---

LANGUAGES: Bash, C, csh, Java, JavaScript, Perl, PowerShell, Python, Swift, SQL

BUILD SYSTEMS: Apache Ant, Apache Maven, make

TOOLS: Ansible, Docker, Eclipse, Git,  $\LaTeX$ , PostgreSQL, PyCharm, Jenkins, Xcode

CONCEPTS: Agile Development (Scrum framework), Operating Systems, Containerization and Virtualization, Continuous Integration/Delivery (CI/CD), DevOps, Dependency Injections, Distributed Systems, Object Relational Mapping (ORM), Publish-Subscribe Messaging Pattern, Release Engineering, RESTful APIs

## EDUCATION

---

### University of California, Santa Cruz

*Computer Science, B.S.*

June 2016 – June 2019

*References available upon request.*