

This is a programmatically generated resume, for an up to date & formatted version please visit maxbbender.com

Max Bender

Principal Engineer

Email: maxbbender@gmail.com

Experience

KVM Solution Test Lead

July 2020 - Present <https://www.ibm.com/us-en/>

In July I took over a new position as a technical team lead for the KVM Solution test project with a team of 5 people. This position's responsibilities mainly revolved around project management but also involved architecting a new automation framework to achieve our goal of 100% automation coverage. Working on KVM (a Linux hypervisor) we utilized Ansible to manage system configurations as it is a very powerful tool due to the idempotency it implements.

While in the position, I learned a lot about how to balance business requirements & deadlines with necessary technical innovation. We worked with the 3 major Linux distributions (RHEL/SLES/Ubuntu) engineers to test & verify KVM on their platform, so there was a lot of external deadlines & processes I had to take into account with our test plan. Since starting this position I have helped reduce our overall test cycle length by 25% and shifted it to be 5+ months earlier by leveraging the previously mentioned end to end automation framework built to verify KVM on Z.

Principal Test Engineer

July 2017 - July 2020 <https://www.ibm.com/us-en/>

My first full time career at IBM was on the Linux Center of Competence team which focused on testing Linux on Z with a similar context as my internship. Part of the team's mission was to remove excuses when it came to running software on Linux on Z. I had the chance to work and architect many different CI/CD solutions, and participated in workshops with our customers, advocating open source software solutions on the Z platform.

Internship/Co-Op

December 2015 - July 2017 <https://www.ibm.com/us-en/>

During my schooling at Marist College, I worked for IBM during the school year and summer. It was a challenge to balance school work with my job's responsibilities but gave me a lot of opportunities to practice balancing priorities from different sources. I learned a lot during this experience, and it gave me a great jumpstart into my future career at IBM.

Help Desk Operator

2014 - 2015 <https://www.marist.edu/>

While my time at the Help Desk wasn't the most glamorous, I valued it a lot. I was resetting passwords and helping people out with basic technology issues but during that I was gaining so much experience in talking with customers. We were on the phone or helping people in person and looking back on it this experience has strengthened my ability to interact with customers in the field.

Marist College

2014 - 2017 [None](#)

Graduated with a bachelors in Computer Science with GPA of 3.8 and honors. During my time at Marist I participated in the MCCS and many other community activities which strengthened my value as a teammate and leader.

Projects

KVM Solution Test www.linux-kvm.org/page/Main_Page

Back in July of 2020, I took over the role as a Technical Team Lead for the KVM Solution Test project. When I joined, there was a foundation of automation but I quickly identified areas that could be improved on. Over the next 6 months my team and I worked to achieve a goal of 100% automated coverage of the test cases we were responsible for. We have achieved that goal and now are functioning as a mature development organization, evaluating new line items/features as they arise and have begun the process to modularize & externalize our environment.

We worked with the 3 major Linux distributions (RHEL/SLES/Ubuntu) during their Beta testing periods to verify KVM on their platforms. It was a challenging endeavor maintaining our relationships with 3 different sets of engineers, but I ended up learning a lot from the process about managing priorities and expectations for my team. Where applicable we accommodated large scale testing scenarios (involving 10TB+ of RAM or 1000+ KVM guests) which provided a unique value-add to the community, verifying the platform at an enterprise scale.

The DevOps framework was built with Ansible and consisted of 25k+ lines of code. We approached the framework with an Agile methodology in mind by building an MVP, getting hands on the solution in order to identify the next required evolution. It is going great with over 200 pull requests merged to date, and many more new ones currently open. The power of our collaboration across the team has built in an incredible level of resiliency to the code base, setting the team up to continue to innovate for years to come.

Travis on Z github.com/bendermlIBM/travis-cookbooks

A big part of my voyage in OpenSource has been learning how to use CICD tools to build automation around a product. For me it was a necessity, because the tediousness of re-running builds had gotten to vast. Travis offers a free CICD mechanism hooked natively into GitHub and many of the projects I was looking at were using it in their pipelines.

Internally we had a Travis environment for x86 and ppc64le, but none for s390x. I was tasked with supporting Travis on Linux on Z and so began a month dive into the source code of Travis. With this we were able to enable a regression pipeline for the release for IBM Cloud Private (ICP) on Z, a Kubernetes offering by IBM.

In collaboration with my team lead and STSM @ IBM, Jay Brenneman, we built and ported the Travis codebase to `s390x`. Jay focused on the hypervisor end, initially hooking OpenStack up to z/VM and eventually moving to KVM on Linux on Z. Meanwhile I focused on the provisioning tools that largely make up the magic of Travis. Largely this consisted of Chef cookbooks and custom bash scripts and took some 3 weeks to get our first working build, but it was glorious.

For the first time ever our developers were able to modify their Travis pipelines and run their builds on Linux on Z. However as they integrated the s390x builds, all of a sudden Travis on Z became mission critical to all of ICP development due to the nature of Travis and GitHub. If even one architecture fails to pass, Travis will block code from being merged into the master branch. When Travis on Z took an outage, the messages and alerts quickly started flooding in. After our second extended outage we quickly realized that we had to come up with a highly available solution for the tool.

Utilizing a load balancing feature in RabbitMQ we stood up another OpenStack instance located in Hursley, United Kingdom and replicated our port of Travis on Z to those systems. Since then we have been able to take independent outages without affecting an ever growing demand for Travis on Z.

â€” Extra Info

It began with connecting the [travis-ci worker](#) to an Openstack instance hosted on z/KVM. However we ran into issues with big bursts of builds which caused the networking layer in Openstack to intermittently not initiate. Eventually we upgraded to use LXD to run the builds which gave a huge speed increase and more portability. Using Packer I built [xenial/bionic base images](#) for Linux on Z and deployed a production configuration supporting 20,000+ builds a year. See some changes to the [travis-build scripts](#) which also needed some configurations

Once I moved to the LXD based builds I was able to automate the deployment of LXD and the Travis Worker on a Ubuntu Bionic host using Ansible. Still using the previous OpenStack environment we could now provision infrastructure with Terraform and configure workers using Ansible.

Let's Talk [None](#)

Teaching others and seeing that lightbulb go off is an experience that I get a lot of gratification from. In my effort to educate teams at IBM I realized that giving the same presentation to team after team was an inefficient method. So I created "Let's Talk" which is a group of education modules that have an intense focus on hands-on learning using examples hosted on GitHub. We scheduled live presentations and recorded them for future consumption and they featured code examples accompanied by a tutorial-like presentation.

My main goal with this series was to provide self-sufficient materials for onboarding new-hires and people new to the technology. Hands-on examples were integral for this effort as they gave people the resources & examples they need to get off the ground. External references and documentation were heavily leveraged to demonstrate that all the information is available online, you just need to know where to look.

Education Modules

1. Ansible
2. Kubernetes
3. Jenkins
4. Travis
5. Python

IBM Cloud Private ibm.com/cloud/private

My team was tasked with supporting a release of IBM Cloud Private (ICP), a Kubernetes offering by IBM, on Linux on Z. Our goal was to have full support by the 3.1.2 release and to do this we ported Travis to the Mainframe as mentioned above.

Alongside these efforts we saw an opportunity to further drive our teams mission, to remove excuses; specifically surrounding running your application on Linux on Z. I was responsible for provisioning and supporting 3 production ICP clusters that served as a Content Development portal and hosting utility for around 25 teams at IBM, 200+ people.

Our goal was to stay current with the releases of ICP so we supported an N-1, N and N+1 configuration where N was the current GA version of ICP. About once a quarter we had to cycle our clusters so quickly I realized that automation was necessary for my own sanity. Using Terraform we would interact with OpenStack to provision around 10 s390x (Z) nodes and 3 x86 nodes. After the nodes were provisioned I would utilize Ansible to configure those nodes with some packages, custom security policies, users and files required for the install.

Post install I once again utilized Ansible to configure RBAC roles for the 25 teams part of the Content Development Clusters. Manually it took about an hour and a half to click through the web-ui, but with Ansible to interact with APIs and a little custom Python it only took around 6 minutes running in the background.

Our team also used these clusters to host and port many different open source applications to Linux on Z. Using Travis on Z I containerized applications like Jenkins, OpenFaaS, Postgres, Redis, as well as custom application written in Python and Bash in regression. We use these examples to write public build scripts released by the [Linux on Z](#) outreach team on GitHub.

Skills

My experience has exposed me to the languages and skills listed below, some in greater detail than others.

Fluent

Ansible Docker Python Linux Flask Kubernetes KVM Git Travis

Proficient

Grafana/Prometheus Java Jenkins Terraform PHP Elasticsearch SQLAlchemy SQL

Familiar

Project-Management Postgres OpenStack RabbitMQ Packer AWS Chef Ruby Redis

References available upon request