Benjamin Burner

Cary, North Carolina | +1-484-947-1738 | benjamin.e.burner@gmail.com | biffkittz.com

Education

North Carolina State University, Raleigh, NC **December 2014**
Computer Programming Certificate

North Carolina State University, Raleigh, NC **December 2012**
Master of Statistics, Mathematical Statistics

West Chester University, West Chester, PA **May 2009**
Bachelor of Arts in Geography and Urban Planning, Mathematics Minor w/post-bacc Research in Statistical Cosmology

Skills and Recognition

- Azure Cloud Services / OpenAI - Cognitive
- AWS CloudFormation / IaC
- C# TPL / ASP.NET / JavaScript / AngularJS
- PowerShell Module Development
- Cloud Application Network / Compute / Database Load Test Design, Execution, and Analysis
- Cloud Cost Analysis and Cost Reduction Plan Development and Execution
- AWS OpenSearch Debugging, Monitoring, and DQL / Lucene Query Development
- Technical Product Management
- Application Release Management
- Python Script and Web App Development
- Grafana / Prometheus / Alert Manager Containerization and Administration
- SQL Server Management, Query Optimization, DDL / DML Scripting
- Bespoke Application Monitoring and Alerting Systems Development and Administration
- Disaster Recovery Plan Development and Testing
- Production Support and Debugging: Adept at Understanding and Troubleshooting Highly Distributed, Asynchronous Cloud Applications
- AWS SWF, S3, EC2, Route53, SES, SNS, SQS, Lambda, Redis
- Docker, AWS ECS, Kubernetes, Containerization
- GitLab CI/CD Development, YAML
- DMARC / DKIM / SPF / DNS Configuration
- Linux System Administration, Networking, Bash
- Architecting ASP.Net Core Web Apps on Linux with Nginx, Kestrel, Digital Ocean, and Cloudflare
- ASP.Net MVC, Razor, Blazor, SignalR, Websockets, Entity Framework, RabbitMQ, PostgreSQL
- US Patent 12177230: Systems and Methods of Managing Fraudulent Devices

Professional Experience

Senior Site Reliability Engineer II **March 2020– May 2025**
ConnectWise, ScreenConnect Raleigh, NC

- Mentored and led a team of colleagues responsible for ScreenConnect Cloud systems support, maintenance, and reliability
- Developed internal PowerShell and Python-based tooling modules and packages to support ScreenConnect Cloud operations
- Developed, deployed, and consumed observability, monitoring, and alerting systems to process cloud application network, compute, and storage metrics, AWS workflow state metrics, Azure Cloud Service traces, IIS log data, and Redis / SQL Server KPIs
- Managed responses to and resolutions of ScreenConnect Cloud incidents throughout the entire incident lifecycle: Generated post-mortem documentation and led cross-team project collaborations with development and product management to improve system resiliency, mitigate system domain failures, and prevent future incidents
- Developed an internal SQL script library to support ScreenConnect Cloud operations and development decision-making
- Collaborated closely with ScreenConnect development and product management to prioritize and scope new ScreenConnect Cloud development initiatives
- Coordinated with development, product management, and application security teams to implement application / infrastructure security posture improvement and management
- Completed production infrastructure migration from AWS EC2 to bare metal providers to realize an approximate 10x cost savings
- PowerShell: Developed and extended the functionality of various Git/Operations-related PowerShell modules to perform tasks like build cloud releases, update server configuration to the desired state, and deploy updates to various application environments
- C#: Developed various ScreenConnect Cloud triggers written in C# that integrate with the application's data stream and perform actions in real time, such as emailing alerts and altering application state based on patterns in the data stream over time
- Data Handling: Developed internal tools in PowerShell / Python that integrate with various application data stores like Redis and SQL Server to perform data retrieval and analysis tasks in support of internal development, security, sales, marketing, and product management teams

DevOps / Cloud Engineer I **March 2018– March 2020**
ConnectWise, ScreenConnect Raleigh, NC

- Designed and developed a subsequently patented Python-based internal app to analyze newly created ScreenConnect accounts in realtime for potential malicious activity and suspend accounts determined to be sufficiently suspicious; US Patent 12177230
- Managed ScreenConnect Cloud release builds and deployments: Reviewed and tested pre-release code, built cloud application release branches, documented release and rollback plans, and ultimately released code to production and performed post-release application validation
- Developed, documented, and tested ScreenConnect Cloud disaster recovery and failure mode plans
- Developed, documented, and tested PowerShell and Python-based internal tooling to support ScreenConnect Cloud operations
- Periodically reported on ScreenConnect Cloud infrastructure cost, cost projections, cost reduction opportunities, and infrastructure

availability to key business stakeholders
- PowerShell: Built a PowerShell module related to ScreenConnect Cloud disaster recovery that would quarantine and heal certain application subsystems in the case of a disaster
- C#: Worked closely with the development team to arrive at code fixes and architectural improvement plans and developed internal tools in C# that consume core ScreenConnect libraries to perform tasks pertaining to internal analytics and operational excellence
- Data Handling: Built a SQL script library to support ScreenConnect Cloud operations that remains in use to this day to help answer questions like “Which cloud regions should we prioritize for capacity expansion?” and “How many accounts from the last 7 days have a particular characteristic?”

Application Developer **Sept 2015– March 2018**
ConnectWise, ScreenConnect Raleigh, NC

- Debugged customer on-prem installations of ScreenConnect and developed bug fixes
- Developed ScreenConnect extension packages in C / C# / JavaScript / CSS to expand base ScreenConnect client / server functionality
  - Report Manager: Presents a modal UI from which ScreenConnect server administrators can build, save, and execute SQL-like queries to generate CSV / HTML / JSON reports about application and security activity associated with their ScreenConnect server
  - Bridge Service: Bundles a Windows service written in C / C# that translates a subset of the VNC / RDP protocol to and from the ScreenConnect relay protocol so that ScreenConnect technicians can connect to endpoints via VNC / RDP using the ScreenConnect remote support client and ScreenConnect relay server
  - Client Network Deployer: Standalone WPF app that scans a subnet for eligible Windows hosts, transfers the ScreenConnect Windows client installer to selected endpoints on the subnet, and initiates the remote installation of the ScreenConnect client
  - Zendesk Integration: Both a Zendesk app and ScreenConnect extension that work in tandem to sync activity between Zendesk support tickets and associated ScreenConnect remote support sessions
- PowerShell: Built a PowerShell script library that integrates with the ScreenConnect web API to authenticate, query, and consume recordings of remote session activity
- C#: Each of the ScreenConnect extensions listed above included at least some component written in C# for .NET Framework; for example, the server-side service exposing the Report Manager extension’s web API is written in C#
- Data Handling: Wrote an internal tool in C# that integrates with SQL Server and AWS S3 APIs to identify a set of application-related S3 objects eligible for deep archive