Site Reliability Engineering (SRE)

Level/Skill	Product Output	Communication/Writing	Networking	Security	Systems Engineering	Site Reliability Engineering (SRE)
1	Creates a design document based on well-defined scoped requirements and implements it.	Reports progress on a regular basis as required by the team's operational requirements. Actively solicits feedback. Participates on interview panels.	Understands common networking issues and troubleshooting techniques. Understands different networking layers (OSI model). Understands basic network concepts like subnets and routing. Understands basic network protocols like TCP/IP, ICMP, DHCP, and DNS.	Understands operating systems security principles like iptables, users/groups, file permissions, and capabilities. Can apply operating system hardening guides like CIS Benchmarks. Understands basic security principles like SSH keys and TLS certificates.		Understands and can make changes to existing Makefiles, shell scripts, and Dockerfiles. Understands system performance basics. Can monitor on CPU, memory, disk, and network utilization. Understands declarative configuration and can use tooling like Terraform or a Kubernetes operator to make changes to existing infrastructure and applications.
2	Can write high quality user and product focused documentation.	Provides constructive review on peers' code and design. Helps new team members during their first weeks.	Can independently troubleshoot most network issues. Can setup and configure production quality network infrastructure like DNS, Load Balancers, and encrypted overlay networks (using IPsec or WireGuard). Can reason about their reliability.	Understands and can apply basic cryptographic principles. Can configure SSH and TLS for a server (chooses appropriate key sizes, algorithms, and versions), pick strong authentication and authorization primitives, and appropriate encryption for data in transit and rest. Can setup production quality encrypted networking (like IPsec or WireGuard). Can reason about their reliability.	Understands the usage of POSIX and other APIs for Linux systems.	Can independently troubleshoot basic systems issues. Uses standard tools and logging to troubleshoot issues. Can use declarative languages like Terraform to build and manage infrastructure. Can configure alerts on latency, traffic, errors, and saturation issues. Uses Cloud native metrics, monitor, and alerting stacks (CloudWatch, Prometheus, Grafana) Is a member of on-call rotation and can resolve issues outlined in runbooks. Demonstrates knowledge of AWS. May have certification like AWS Certified SysOps Administrator.
3	Collaborates with the team to scope requirements, based on good understanding of existing longer term product vision and estimates of the system design of a feature of a product.	Coordinates project deliverables alongside parallel team efforts. Supports less experienced peers' technical skills, answering questions and being a resource. Documents and improves team practices.	Can setup and operate a multi-region infrastructure and networking environment. Can reason about it's performance, reliability, and failure modes. Has in-depth understanding of container networking. Can write own CNI plugin utilizing IPSec or WireGuard on Kubernetes. Understands advanced networking concepts like NAT traversal and BGP.	Can build secure systems that will pass quality security audit that will uncover few to no critical system design errors. Can apply security principals when building systems. Can utilize access control primitives (like IAM and RBAC) to limit access to infrastructure. Understands secret life cycle management in production environments. Understand API authentication systems, can reason about trade-offs between use of JWT, OIDC, and mTLS. Understands security critical events that occur within a system and can configure alerting on them. Understands advanced operating system security concepts. Can utilize Mandatory Access Control (MAC) systems like as SELinux or AppArmor. Understands container security. Can utilize control groups and namespaces to isolate and application.	Can write software (like tools and automation) that is readable/extensible and used in production. Understands basic testing concepts like unit and integration testing.	Excellent systems troubleshooter. Can diagnose and resolve cascading failures. Uses modern BPF tools for troubleshooting. Demonstrates advanced knowledge of Kubernetes deployments (CRDs, ingress controllers, Load Balancers). Can build and maintain reliable production CI/CD pipelines for build, test, and release. Demonstrates advanced knowledge of AWS. May have certification like AWS Certified Solutions Architect or Certified Kubernetes Administrator (CKA).
4	Leads the implementation of the isolated feature/improvement that measurably and significantly impacts business outcomes from gathering requirements to getting to the market stage.	Leads and clearly articulates project deliverables. Writes technical articles/blog posts, delivers tech and lightning talks representing the company's technical vision. Writes Root Cause Analysis (RCA) documents after incidents that help the team mitigate recurrence of that issue.	Can make changes to existing network infrastructure tooling (like load balancers, DNS servers, service meshes) to solve relevant business needs.	Writes technical articles on security aspects of the system, implements significant security product innovations in the area delivered to customers. Understands and can apply advanced cryptographic principles. Understands hashing (including for anonymization), when to use symmetric and asymmetric cryptography, cipher modes, and TLS versions. Understands and can apply advanced network security principles. Understands data extrusion prevention. Understands DDoS mitigation. Understands how to monitor systems for rootkits and can deploy mitigation strategies when a system is under attack.	Not only can write data-race and dead- lock free code, but implements safe and concurrent and/or parallel systems using minimum amount of shared state, granular locking - systems that are easy to read, extend and troubleshoot.	Can build and operate large scale, stable, and reliable production platform environments like Kubernetes. Understands service availability and helps developer Service Level Indicators (SLI) and Service Level Objectives (SLO). Can deploy and operate databases at large scale. Understands index compaction, failover, sharding, and query performance analysis. Writes high quality design documents with few to no critical system design errors.
5	Leads the implementation of a new product line or significant part of the product to deliver it to the market in collaboration with all other teams.	Writes advanced technical articles/blog posts, gaining significant industry traction or delivers techmical talks on major conferences representing the company's vision.	Can create network infrastructure tooling to provide service level load balancing, multi-region connectivity, and observability (like Cilium).	Researches and designs new security systems and protocols.	Can implement production grade systems leveraging advanced low-level and/or novel components like eBPF, control groups, or Noise Protocol Framework.	Understands and uses advanced system performance troubleshooting techniques like ptrace, strace, flamegraphs, or writing custom bpftrace programs. Can build advanced monitoring and anomaly detection systems.
6	Designs new data structures and algorithms solving relevant business problems and creating competitive advantage for the company.	Produces peer-reviewed research papers or patent applications.			Can design and build system for container orchestration and management like Kubernetes.	