Victor Lin

Sr. Engineering Manager

Cloud Engineering @ Houzz

Agenda

- Highlights
- Project Outline - Infra Unification in EKS (2025)
- Project Outline - Modernize MySQL Clusters (2024)
- What’s Next?

Highlights

Current Role:
Lead Core and Data Infrastructure teams of 9 engineers (including 2 tech leads) across multiple time zones (East Coast, West Coast, APAC).

Core Focus Areas:
AWS, EKS, Observability, IaC, Databases, SRE, ETL pipelines, Security & Compliance,
Cost Efficiency & Optimizations

Major achievements:
Unified infrastructure stacks on EKS, reducing infrastructure costs by 35% over 12 months.

Career Progression at Houzz:
2017 - Senior Software Engineer
2019 - Staff Software Engineer/Tech Lead
2020 - Engineering Manager (team of 5)
2024 - Senior Engineering Manager (team of 9)

Mission:
Build and evolve a secure, scalable, and cost-effective Cloud and Data infrastructure that
supports Houzz products and analytics at scale. I focus on driving reliability, efficiency, and developer empowerment across AWS, Kubernetes, and our Data platforms.

Leadership Style:
- Facilitate cross-team initiatives with structured priorities, transparency, and shared goals.
- Champion scalable, low-maintenance systems with clear ownership and minimal operational overhead.
- Guide technical decisions while empowering leads to evaluate trade-offs and own designs.
- Assign outcomes, not tasks — trust leads to drive projects end-to-end with support, not micromanagement.
- Promote documentation, retros, and shared learning as part of team culture.

Project Outline - Infra Unification in EKS

Challenges:
- **Outdated kops clusters** with maintenance overhead
- **Architectural inefficiencies**: Monolithic clusters had tight coupling, cross-cluster dependencies, additional routing hops, and hardcoded ELBs.
- **Inconsistent tech stacks**: Observability (Thanos vs VM), Service Mesh (Istio vs Linkerd), CNI (Cilium vs Calico), Cluster Topology, etc.

Technical decisions:
- Business-oriented cluster topology (e.g., consumer, saas, bot, payment).
- **In-place upgrades**: EKS-native in-place upgrade over Blue/Green to reduce cost & operational complexity.
- **IaC tool split**: Used Terraform for infrastructure provisioning and ArgoCD for workload GitOps
- **Pre-cluster routing**: Shifted from EC2-based proxies (HAProxy/Nginx) to Fastly → ALB → K8s Ingress, reducing network hops and cost.

Outcome:
- Reduced overall infra + engineering cost.
- Established a repeatable, fully automated cluster provisioning process (i.e. provisioning a production ready cluster in 30mins).
- Enhanced **Disaster Recovery** posture —clusters are now backed by IaC and mesh-based routing.
- Achieved zero-downtime deployments using canary rollouts and in-place upgrades.

What would you do differently next time?
- Focus on one cluster and certain components once at a time instead of working on multiple areas.

Post-Migration Efforts
- Conducted a cost-performance analysis to refine instance types and savings plan utilization.

Project Outline – Modernize MySQL Clusters

Challenges:
- **Operational overhead**: Managing multiple replicas across EC2 MySQL clusters involved complex tooling.
- **Resilience gaps**: Manual failover workflows were error-prone, with outages tied to Orchestrator instability.
- Over-provisioned clusters without autoscaling
- Outdated MySQL version 5.7 vs 8.4

Technical decisions:
- **Real-time** data replication to AWS Aurora
- Read traffic canary ramp-up and write traffic flip
- Use **Aurora RDS** for its shared storage model, superior failover times, and better throughput under large dataset workloads.
- **Performance tuning**: Used Graviton instances and I/O-optimized configs; targeted 99% InnoDB buffer cache hit ratio to minimize disk I/O and improve latency

Outcome:
- No data loss and minimum downtime during operation
- Better performance
- Lower cost with autoscaling setup in Aurora RDS
- Less maintenance overhead
- Blue/Green failover with minimum downtime
- Updated to MySQL 8.3.x (8.4 now)
- Support existing operations and pipelines

What would you do differently next time?
- Implement **reverse replication** for EC2 fallback safety
- Set up a production traffic replay testbed earlier in the life cycle
- Improve staging/production environment parity to minimize surprise misconfigurations

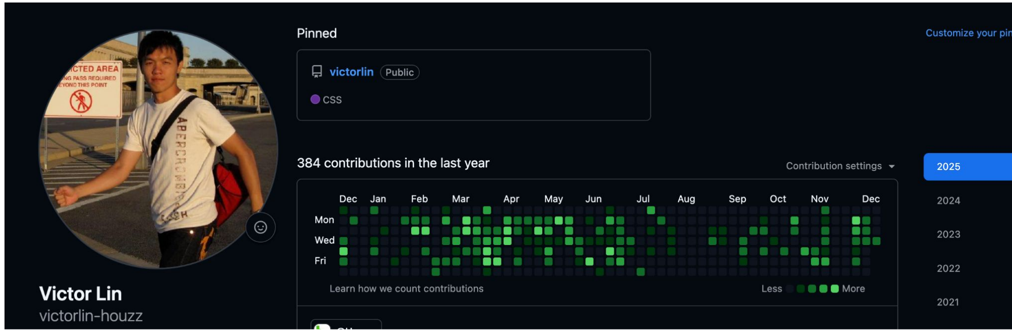
Post-Migration Efforts
- Reviewed and adjusted autoscaling policies and instance types based on observed workload patterns.

What's Next?

Challenges:
- AI Adoption
- Cost Efficiency
- Site Performance, Scalability & Reliability
- Next generation of Analytics Platform at Houzz

Ongoing Projects:
- Claudecode adoption for Infra organization
- Snowflake Partnership
- Istio Ambient Mode evaluation
- Open Telemetry Client + Collector Implementation
- MySQL performance regression detection
- Flink tuning and resource consolidation
- JupyterHub support in Kubeflow with shared data
- Next-Gen WAF adoption in Fastly

Reference:
- Resume (Link)
- LinkedIn (Link)
- Github (Link) - 384 Contributions in 2025


Pinned
Customize your pi
victorlin Public
CSS
384 contributions in the last year
Contribution settings
Dec Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
Mon
Wed
Fri
Learn how we count contributions
Less
More
2025
2024
2023
2022
2021
Victor Lin
victorlin-houzz