**Expedia – “Designing, building and configuring a geo-available instance of GitHub Enterprise”**

Chris Jantzen – Systems Engineer - DevOps

GitHub is 60-70% remote

Meet for 4x/year

GitHub business cloud vs GitHub enterprise

Jeff Josephson and Chris Jantzen from Expedia

JIRA, confluence,

**Objective**

Consolidate Source Control

Git Usage at Expedia Group

* GitHub Enterprise 58% - 4673 users
* Bitbucket 41% - 3322 users
* GitLab 1% - 61 users

*Consolidate into a single platform*

**Requirements**

Available, Secure, Resilient

**Key Attributes**

* Built in the cloud
* Scalable
* Lower infrastructure costs
* Automatic deployment and configuration
* Public endpoint
* Easy connectivity from anywhere
* Single endpoint for all
* Allows third party integrations
* Secure
* IP Whitelist (*have to be approved and allowed*)
* Endpoint protection (*protect from ddos attack*)
* Geo-available
* Replicas in multiple regions
* Intelligent traffic routing *(single endpoint but can handle multiple)*
* Regional failover capability

*Easily integrate with knowledge-based tool, i.e. JIRA*

**Design**

Cloud Optimized

**Details**

* Built in AWS
* Regional VPCs with peering
* Network segmentation with subnets
* Akamai
* Edge Provider
* Geo-aware traffic routing
* Endpoint protection
* IP whitelist
* Direct Connect
* Hooks to legacy systems
* Instance administrative
* Connectivity to configuration management tools

*Akamai – Whitelist/Geolocation*

*GitHub proxies*

**Implementation**

GitHub, the ALM way

*Embrace the philosophy of devOps*

**Infrastructure as Code**

* AWS CloudFormation
* VPC
* EC2 Instance: GitHub Enterprise AMI
* Load Balancer: Classic ELB(443/22)
* Multi-region

**Configuration as Code**

* Automatic bootstrap
* Chef role determines appliance as primary or replica *(picks what gets deployed)*
* Chef Cookbook
* Uses GitHub Enterprise API/CLI *(configurations at CLI)*
* Initial appliance setup
* Instance configuration

*Helps with test approach*

Roles

* Instance Role
* Appliance Role

**Lessons Learned**

Tips for Success

*3-4-month project*

Communication is key

* To Users
* To Leaders

Take advantage of GitHub Support

* Fast response
* Expert knowledge
* Eliminate guessing or assumptions

Understand your network

* DNS limitations *(DNS must be able to route to the right place)*
* Connectivity paths
* Filtering technologies in use

Document, Document, Document *(Build your knowledge base)*

* User guides
* Appliance Configuration
* Design docs

*Question: Why GitHub Enterprise?*

*Security compliance.*

*Source control is heavily-controlled IP*

*Question: Azure?*

*You can replicate in Azure*

*Question: Solution for geographic DNS?*

*Cloud. On-premise network might be the problem.*

*Question: Go repositories?*

*Not a huge presence.*

**T-Mobile – “Open Source at the Uncarrier”**

Nicholas Criss – Sr Manager, Cloud CoE and Open Source Working Group

@nicholascriss

*Known as the un-carrier*

*Going and listening to customers without asking more money*

Been a dev and T-mobile customer for 20 years

T-mobile Cloud Team in 2016 to apply the startup mentality to enterprise problems

Evolution of Cloud @ T-Mobile

* B.C. (Before Cloud)
* AWS Console
* Some Automation
* Infra as Code
* Software Dev
* Platforms

*Zero to Hero (empty to production request with no human intervention)*

*Should attract really good software developers*

*Tagline: transforming wireless through technology innovation*

Opensource.t-mobile.com

*Problem with attracting and retaiining software developers*

*Solution: open source it. Dev: I like the project, I want that company to pay me to work on that project.*

*The needle moves quicker if it is from outside-in*

*Internal buys in when it was executed outside*

**Technology**

Jazz Severless *(tightly-connected to Git workflow)*

Conducktor (*Dual stack model. Too T-Mobiley apps becomes open-source)*

T-Vault *(auto magic integration, open-source only)*

PacMan *(policies code)*

Cloud Insights

Next Directory *(blockchain tech: donated the code to Linux, now it is open-source only. Microsoft and Intel participates)*

Adaptive Security

*Attempt magic and avoid crap*

*Permissions model, before accepting the user invite*

*More stats in GitHub now, better for project management for roadmaps, etc.*

*Question: how did you get your people to go open-source?*

*Better question: procurement? Vendor?*

*Answer: tricky, but we will do it.*

*3 Artifacts people get to review*

* Private unlisted video
* Private unlisted draft blog
* GitHub

*Like to have list*

* This is good, get me numbers. APIs to get insights.
* i.e. adoption curve, users, etc.

https://www.research.net/r/seattle11

(GitHub) Universe

Oct 16-17 | Palace of Fine Arts, San Francisco