

Ansible @ Handelsbanken



Love Grönvall & Christoffer Reijer

Agenda

- The Bank
- Why
- Then
 - Ansible team from Linux team
 - AAP1 -> AAP2 (and the opportunities that follow)
 - [REDACTED] [REDACTED] d [REDACTED]
- Now
 - Infrastructure as Code
 - [REDACTED] [REDACTED]
 - Knowledge transfer / Responsibility / Workshops
 - Standardization and efficiency

Handelsbanken



Bildades 1871



En av världens starkaste banker



All verksamhet utgår från kunden



**Decentraliserat arbetssätt
med tilltro och respekt för individen**



**Rikstäckande kontorsnät
på våra hemmamarknader***



Höga hållbarhetsambitioner



Låg risktolerans

Why?

- R&D – Ansible is a part of our innovation
- Standardization
- Reliability
- Security
- Efficiency

Then - history

- Part of linux team starts working with ansible
- Giant code base with automations – "monolith"
- Genius adaptations – special solutions for everything
- Upgrade from AAP1 to AAP2
- Using Red Hat Consulting Services: Christoffer Reijer

Now - Automation / Ansible team



Everyone is happy 😊

Now - Infrastructure as Code

- Why?
 - Traceability – Can see who did what when
 - Reliability – System restores functional state in case of undesired changes
 - Visibility – We see state in our code
 - Predictability – We define a state, and enforce it's permanence
- How?
 - Validated Content (infra.controller_configuration etc)
- Challenges
 - Higher threshold (processes and vcs, review, testing, understanding IaC instead of clickops)
 - Meta thinking (ansible to configure ansible)
 - [REDACTED] security [REDACTED]

Now - Workshops

- Automation team can't/shan't do all automations at the bank (competence/time)
- Knowledge transfer to other teams
- Catch existing competence/interest in teams
- Onboarding – How to get a team to learn and own automations?
- Workshops from real world use cases – creation of deliverables
- 1-2 days for workshop + before-meeting

Now - Ansible Collections

- Why?
- Breaking (down) the monolith
- Standardization / From templates
- [REDACTED]
- Ansible Molecule
- Ansible Lint
- [REDACTED] [REDACTED]
- CI/CD testing
- Roles should be variable agnostic

Now - Integration with order forms

- Ansible jobs to fetch data from order form and populate ansible variables
- Communication in ticket
- [REDACTED]
- Challenges
 - Integrations
 - RFC / Process
 - Testing
 - Approval
 - [REDACTED]

Now - Structure and cleaning (COMPLETED!)

- Migration from ginormous code base
- Standardization of customizations
- Variabel-architecture (prefixes; inventory, role, survey, playbook, credential)
- Use our collections
- Small repositories
- **Roles set states, variables define states, playbooks ties it all together** (order, tags)
- Dependencies should go in a single direction!

Thank you!

Questions?