THE SCALE FACTORY

LANDING ZONE ON AWS: UP AND RUNNING_

MARKO BEVC | he/him/his

RUNNING APPS **ON AWS_**

We’re not doing it because it’s easy, but because they said it going to be easy! 😜

/WHOAMI
MARKO_

- Principal Consultant @ The Scale Factory
- Ops background🦄, wearing different hats🤠 → 🎩, seen things😱
- Open source contributor, maintainer and supporter
- HashiCorp Ambassador, OpenUK Ambassador
- Fan of automation/simplifying things, hiking and travelling🗺️

@marko.social

https://www.linkedin.com/in/marko-bevc/

aws PARTNER Advanced Tier Services
- Public Sector
- Immersion Day
- SaaS Services Competency
- Well-Architected Partner Program

aws PARTNER OF THE YEAR WINNER 2023

kubernetes CERTIFIED SERVICE PROVIDER

British Assessment Bureau UKAS MANAGEMENT SYSTEMS 8289 ISO 27001 INFORMATION SECURITY MANAGEMENT


aws
USER GROUPS
RE:INVENT
PART OF THE
COMMUNITY_

The Scale Factory customer base_






1 Pre-Seed Stage

- Laying the foundation
- Ideation, testing and analysis
- Validation of the hypotheses
- Bringing in key stakeholders
- Clarify market needs
- Understand target group
- Conduct detailed market research
- Present your business plan to your network

2 Seed Stage

- Confirmation of market demand
- Gain financial support
- Validate your business model
- Verify hypothesis with a working prototype
- Finding the right market solution
- Make the right contacts
- Rely on multiple financial sources

3 Early Stage

- Reaching first round of venture capital funding
- Crafting minimum viable product (MVP)
- Establish a sizable customer base
- Steady stream of monthly revenue
- Proof the durability of the revenue
- Present the return on investment (ROI) to the startup's investors

4 Growth Stage

- Scale to a bigger market
- Maintain long-term profitability
- Raise funding
- More diverse funding sources
- Expand the team
- Aim for healthy growth
- Counting on the right people

5 Expansion Stage

- Profitable and self-sufficient
- Aim to expand
- Growth of at least 20% for 3 consecutive years
- Venture into the global market
- Expand to other market segments
- Consider acquisitions to accelerate expansion
- Rely on help, guidance and strategic partnerships

6 Exit Stage

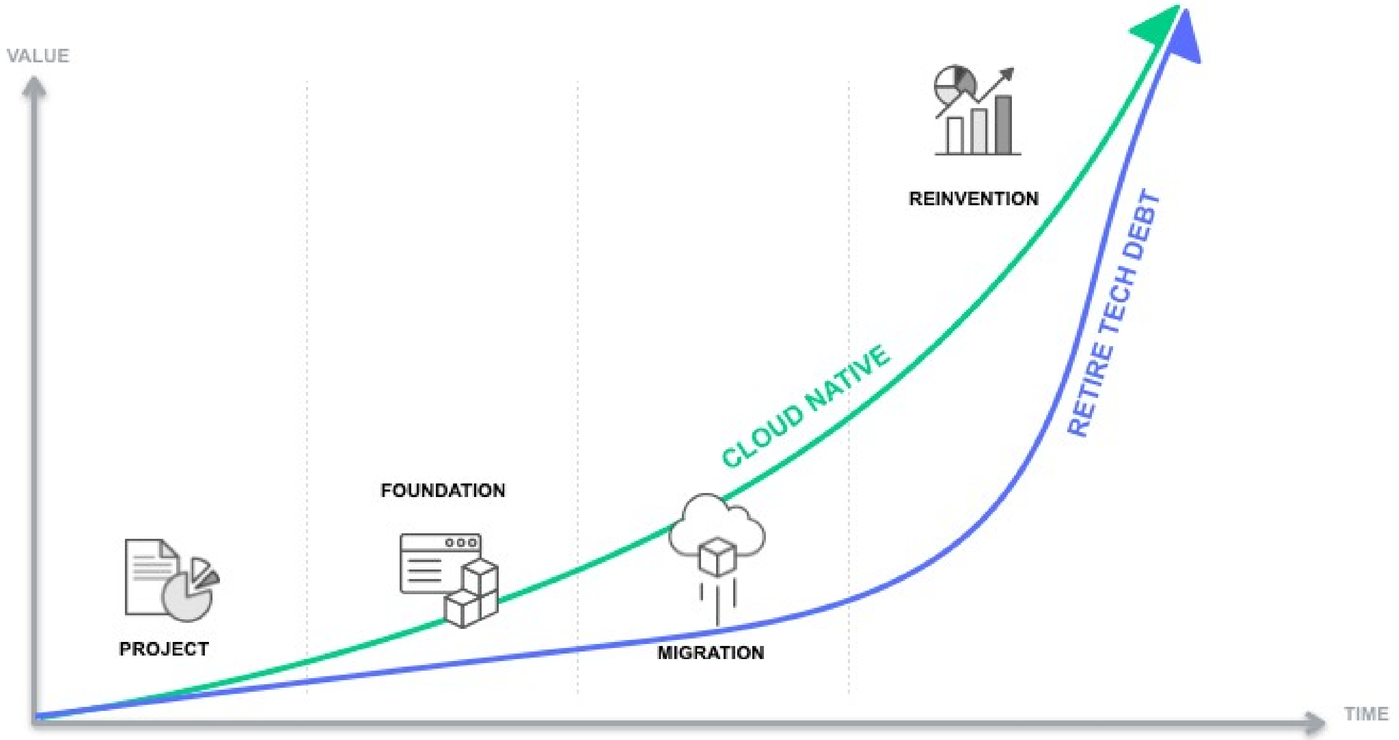
- Expand by acquiring more companies
- The possibility of selling
- Decide how to proceed
- The exit depends on your goals
- Plan an exit strategy
- Get support of a good team of lawyers

Source: https://baselarea.swiss/knowledge-hub/6-startup-stages/

THE TYPICAL START_

- Local development
  - Quick mocks/PoC
- Push something to the Cloud
  - Easy to set up, quick to learn
- Start experimentation → adding more services
- Expanding the team / multi-user
- Start deploying to multiple ENVs, building MVP
- Perhaps even pivot from another Cloud provider?
- ...before you know it it’s slowing you down!

CLOUD **ADOPTION_**



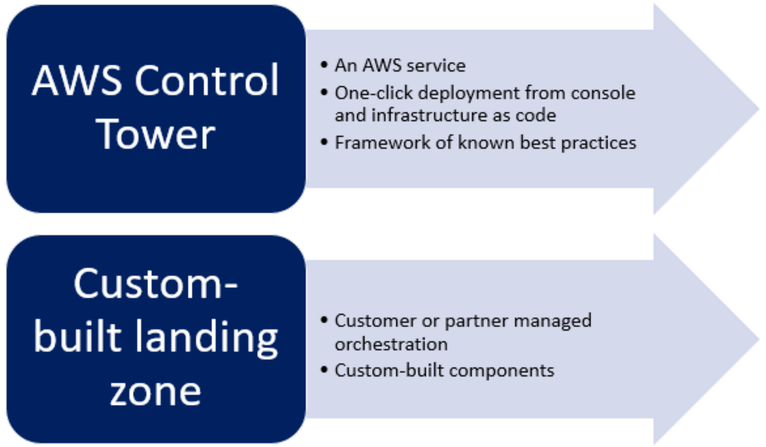
Source: https://baselarea.swiss/knowledge-hub/6-startup-stages/

SCALING UP_

- Run multiple AWS accounts
  - Well-Architected (Ops Excellence, Security, Reliability, Performance Efficiency, Cost Optimisation, Sustainability)
  - Security boundaries and isolation
  - Blast radius
  - Team responsibilities
- Billing and attribution
- Challenges:
  - Code reproducibility – lower margin for error
  - Central governance
  - Identity management
  - Dashboards and compliance
- How to standardise this?

AWS LANDING **ZONE_**

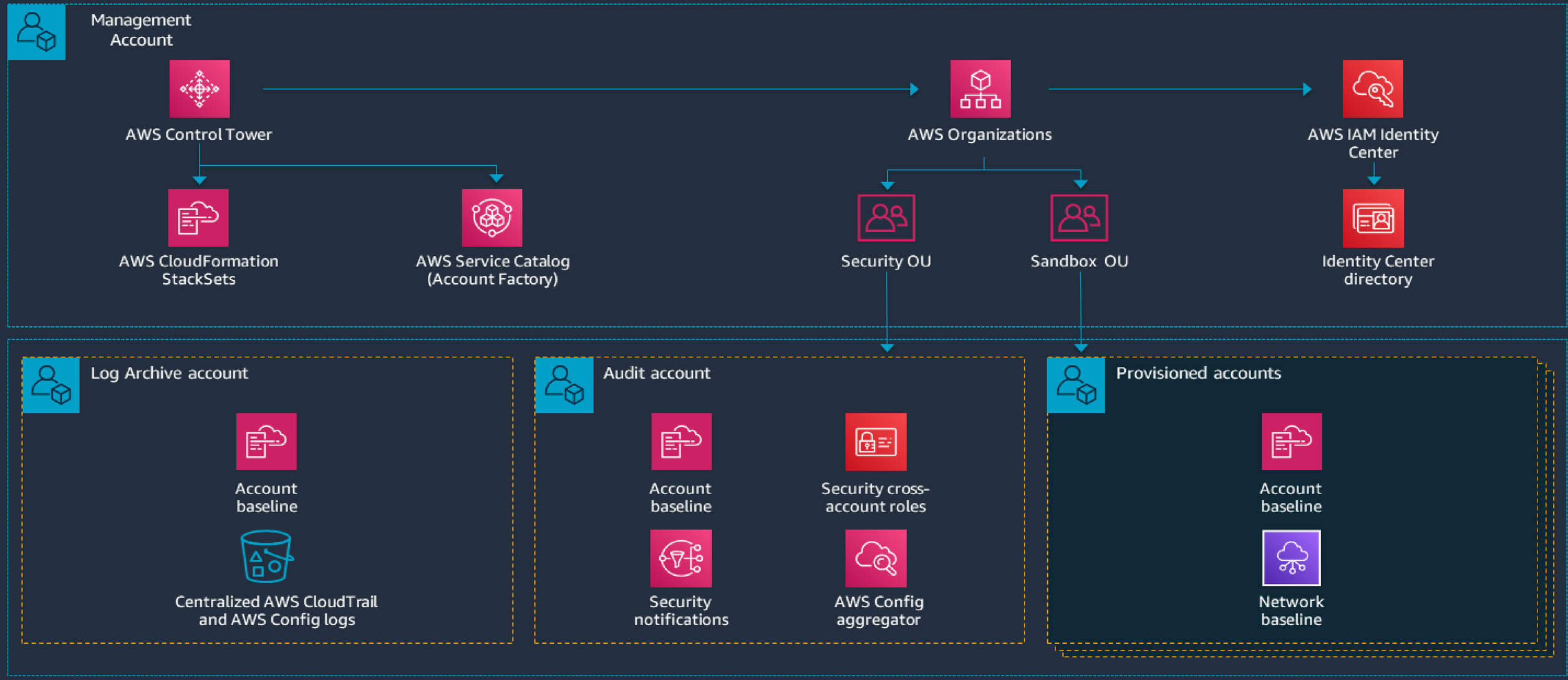
- Automation – provision workloads
- Streamline security – built-in and enforced
- Central governance – single glass of pane
- Improve consistency – across regions and partitions
- Compliance – regulations and data residency
- Scalability – resource elasticity



AWS CONTROL TOWER_

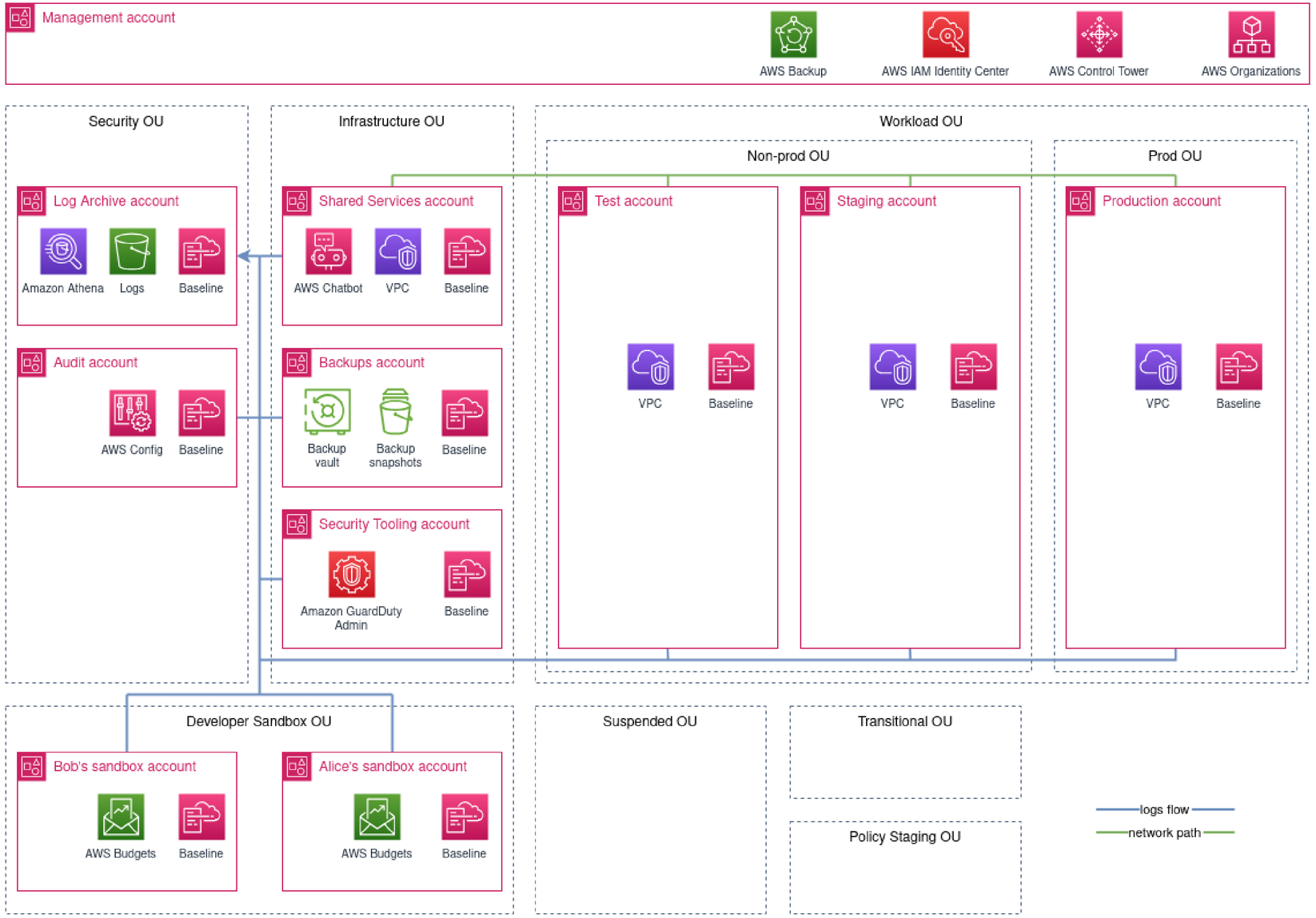
- AWS managed service (build vs. buy) – service catalogue
- Govern a secure multi-account environment
- Controls and guardrails
  - Detective
  - Preventative
- Vend out compliant accounts
- Identity management
  - Internal
  - External IdP – SAML 2.0
- Centralised dashboard
- Policies and IaC integration
- Available API*

Landing Zone provisioned by AWS Control Tower



ACCOUNT LAYOUT_

Did you say automated?_



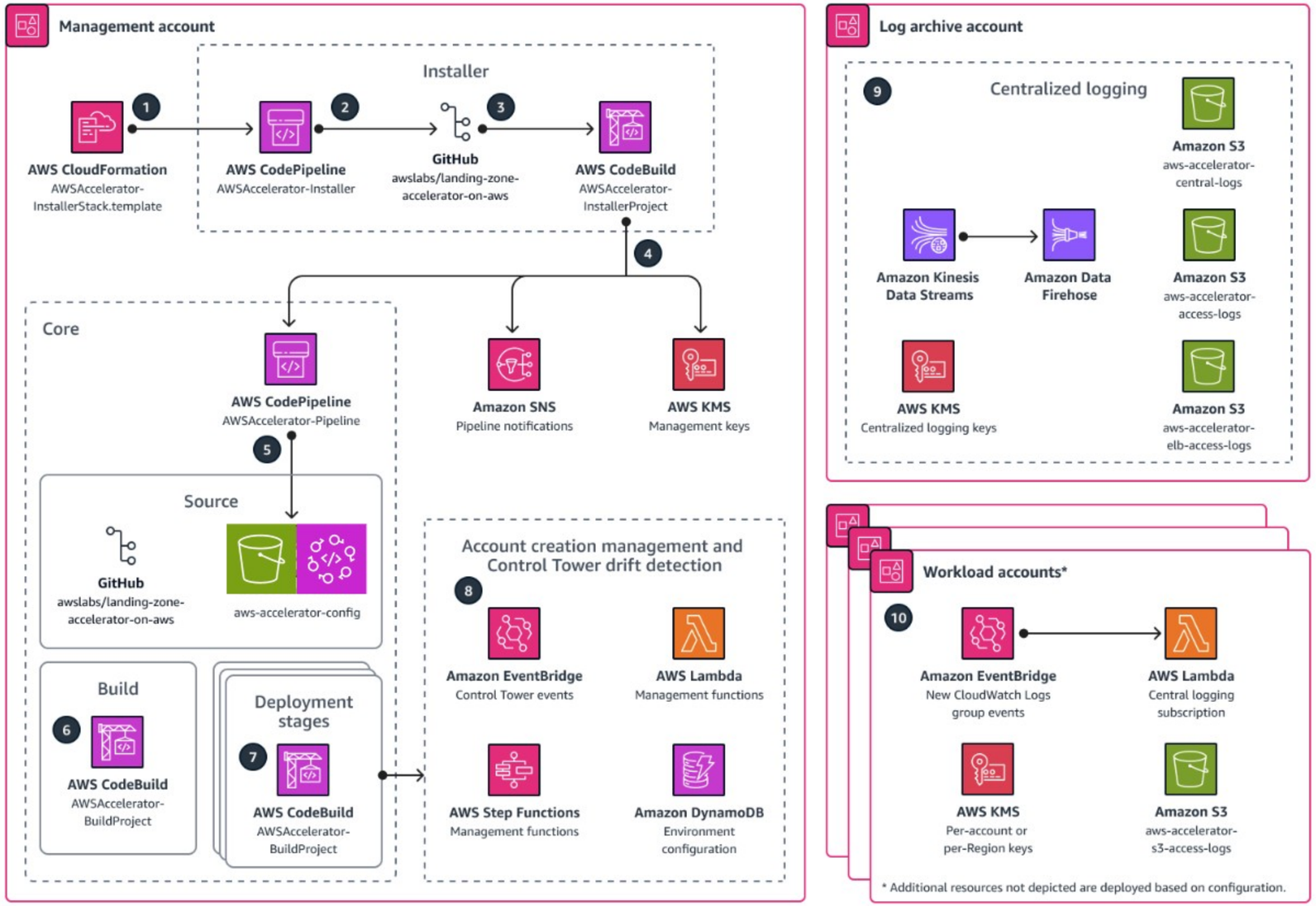
PATH TO **AUTOMATION_**

- Control Tower ClickOps doesn’t scale well
- Different options:
  - Custom tooling / APIs (build vs. buy)
  - Account Factory Customization (AFC) – Blueprints & Catalog
  - Customisations for Control Tower (CfCT)
  - Account Factory for Terraform (AFT)
  - Landing Zone Accelerator on AWS (LZA)*
- Analysis paralysis?
- Similar solutions using different tech
- Ergonomics might differ!

LANDING ZONE **ACCELERATOR_**

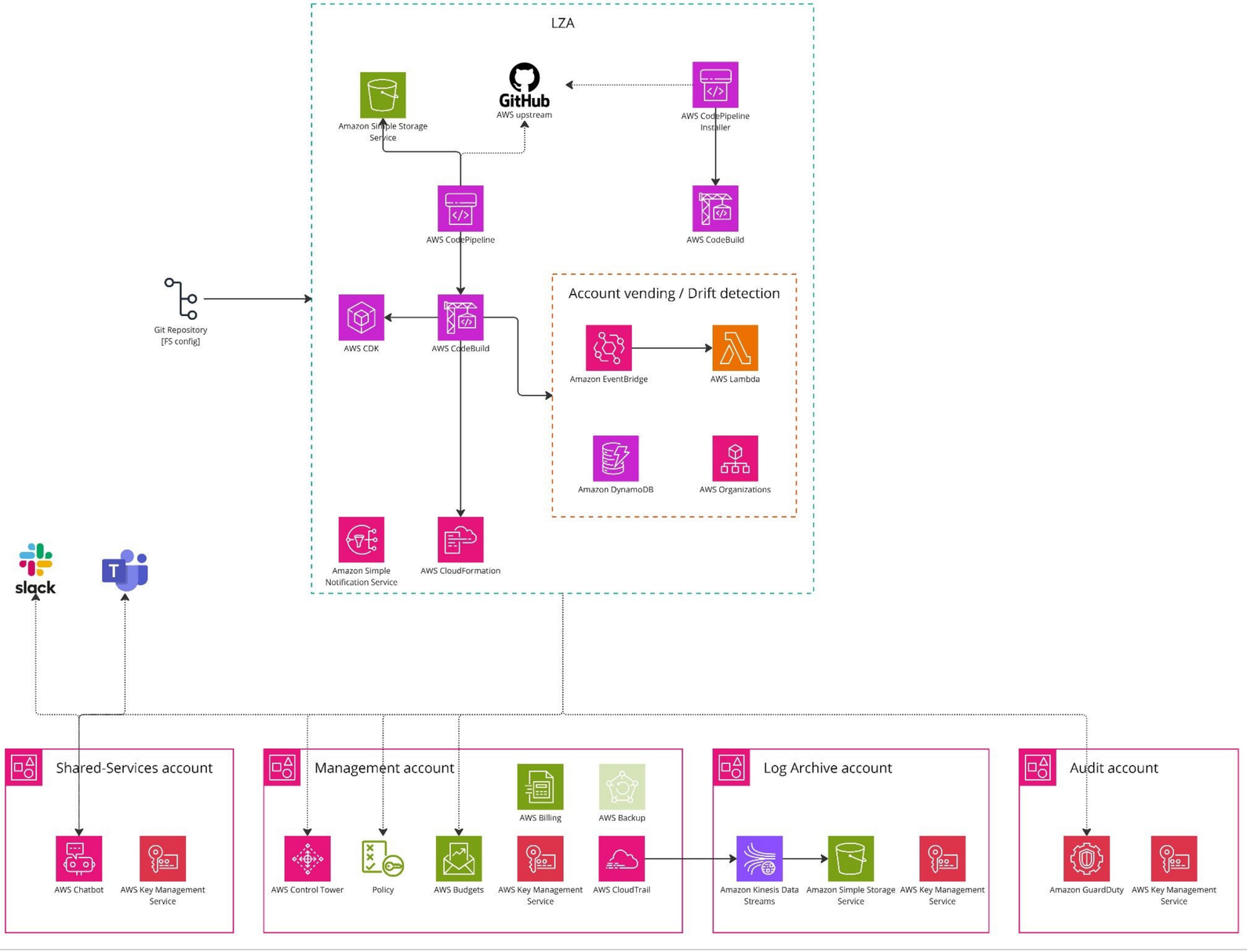
- AWS solution aligned with AWS best practices
- Foundations for security and compliance
- OSS and part of AWSlabs: https://github.com/AWSLABS
- AWS Supported (>= Developer)
- Using CDK under the hood, not preventing other integrations
- Infrastructure as Code (IaC) driven (GitOps)
- Cross industry support (different sectors, e.g. Public, PALZ)
- Low-code solution across 35+ AWS services

AWS SERVICES_



Source: https://docs.aws.amazon.com/prescriptive-guidance/latest/secure-architecture-dod/lza-overview.html

SAMPLE **ARCHITECTURE_**



MORE ABOUT **LZA_**

- CDK → CloudFormation, CLI, `lza-validate`
- AWS CodePipeline / CodeBuild – quick feedback loops
- Everything as Code: well formed and flexible YAML configs 😅
- CT, IAM IC, pre-warming accounts, quotas, GD, password policy, ...
- Good defaults: KMS, logging, drift detection, enforced policies, etc.
- Security and notifications - CloudWatch, SCP, SH, Macie, etc.
- Networking: VPC, Transit Gateway, AWS Firewall
- IAM IC associations, AWS account life-cycle management
- Some manual bits: Region deny, moving OUs or closing accounts
- Tesler's Law – certain amount of complexity that cannot be removed 

Git repo

```
.
├── accounts-config.yaml
├── bucket-policies
│   └── central-log-bucket.json
├── cfn-templates
│   ├── compliance_notifications_chatbot_slack.yaml
│   ├── compliance_notifications_chatbot_teams.yaml
│   ├── compliance_notifications_cross_account_event_role.yaml
│   ├── compliance_notifications_cross_account_event.yaml
│   ├── compliance_notifications_event_to_sns.yaml
│   ├── control_tower_execution_role.yaml
├── customizations-config.yaml
├── dynamic-partitioning
│   └── log-filters.json
├── global-config.yaml
├── iam-config.yaml
├── iam-policies
│   └── sso-AWSBillingAccess-inline-policy.json
├── install
│   ├── AWSAccelerator-InstallerStack.template
│   ├── gh_ruleset-main.json
│   └── github_iam_role.yaml
├── Makefile
├── network-config.yaml
├── organization-config.yaml
├── replacements-config.yaml
├── security-config.yaml
├── service-control-policies
│   ├── deny_all.json
│   ├── org_wide_scps.json
│   ├── quarantine.json
│   └── workload_scps.json
└── yamllint.cfg
```

Git repo

```
.
├── accounts-config.yaml
├── bucket-policies
│   └── central-log-bucket.json
├── cfn-templates
│   ├── compliance_notifications_chatbot_slack.yaml
│   ├── compliance_notifications_chatbot_teams.yaml
│   ├── compliance_notifications_cross_account_event_role.yaml
│   ├── compliance_notifications_cross_account_event.yaml
│   ├── compliance_notifications_event_to_sns.yaml
│   ├── control_tower_execution_role.yaml
├── customizations-config.yaml
├── dynamic-partitioning
│   └── log-filters.json
├── global-config.yaml
├── iam-config.yaml
├── iam-policies
│   └── sso-AWSBillingAccess-inline-policy.json
├── install
│   ├── AWSAccelerator-InstallerStack.template
│   ├── gh_ruleset-main.json
│   └── github_iam_role.yaml
├── Makefile
├── network-config.yaml
├── organization-config.yaml
├── replacements-config.yaml
├── security-config.yaml
├── service-control-policies
│   ├── deny_all.json
│   ├── org_wide_scps.json
│   ├── quarantine.json
│   └── workload_scps.json
└── yamllint.cfg
```

terminal

```
$ aws codepipeline get-pipeline-state --name AWSAccelerator-Pipeline
--query 'stageStates[].[stageName,latestExecution.status]' --output
table

----------------------------------
|       GetPipelineState         |
+-----------------+--------------+
|  Source         |  InProgress  |
|  Build          |  Succeeded   |
|  Prepare        |  Succeeded   |
|  Accounts       |  Succeeded   |
|  Bootstrap      |  Succeeded   |
|  Logging        |  Succeeded   |
|  Organization   |  Succeeded   |
|  SecurityAudit  |  Succeeded   |
|  Deploy         |  Succeeded   |
+-----------------+--------------+
```

SHOW ME HOW
IT WORKS!_

Source: https://github.com/awslabs/landing-zone-accelerator-on-aws/

EXPECTATION_

vs. REALITY

CONCLUSIONS_

& TAKEAWAYS

- 🗼 Setting up Landing Zone is crucial to ensuring secure foundations
- 🔐AWS LZA gets you quite far and ticks a lot of boxes: compliance, security and governance
- 🤹Some rough edges and ‘gotchas’: GH token, long/not free pipeline runs, StackSet retain, limited service coverage (GH Lambda protection) https://github.com/awslabs/landing-zone-accelerator-on-aws/issues_
- 💭 **Boring** is powerful – use your innovation tokens wisely
  - Know your competitive edge
  - Smart decisions about abstractions and managed services

FURTHER **READING_**

- Resources:
  - https://www.scalefactory.com/blog/2024/10/03/building-a-landing-zone-on-aws/
  - https://www.scalefactory.com/blog/2023/08/30/beyond-aws-control-tower-maximising-your-landing-zone/
  - https://www.scalefactory.com/blog/2023/03/23/landing-zone-architectures-and-cloud-security/
  - https://docs.aws.amazon.com/prescriptive-guidance/latest/migration-aws-environment/understanding-landing-zones.html
  - https://docs.aws.amazon.com/controltower/latest/userguide/what-is-control-tower.html
  - https://docs.aws.amazon.com/solutions/latest/landing-zone-accelerator-on-aws/solution-overview.html

KEEP IN TOUCH_

Web: https://www.scalefactory.com/
BlueSky: @marko.social
GitHub: @mbevc1
GitLab: @mbevc1
LinkedIn: https://www.linkedin.com/in/marko-bevc/