AWS Inventory Dashboard – Operations One-Pager

Purpose

Deploy and operate the AWS Inventory Dashboard, which collects resource metadata across multiple AWS accounts and presents it via a secured web UI.

Architecture (High Level)

- **Backend**: AWS SAM, Lambda, API Gateway, DynamoDB, EventBridge
- **Auth**: Cognito User Pool with SAML federation
- **Cross-Account Access**: IAM AssumeRole (read-only)
- **Frontend**: Static web app hosted on S3 + CloudFront

Prerequisites

- Central AWS account for deployment
- Cognito User Pool with SAML IdP configured
- Target AWS accounts with inventory access requirement
- Route53 hosted zone for custom domain
- Node.js, AWS CLI, SAM CLI installed

Backend Deployment

1. Lambda Layers

- Package shared dependencies
- Ensure runtime compatibility
- Version layers appropriately

2. Build (SAM)

- `sam validate`
- `sam build`

3. Deploy (SAM)

- Deploy using `sam deploy --guided` or CI pipeline
- Required parameters:
- Cognito User Pool ID
- Cognito App Client ID

- DynamoDB table name
- Environment (dev/stage/prod)
- EventBridge schedule

4. Cross-Account IAM Setup

- In each target account:
- Create IAM role with read-only permissions
- Allow `sts:AssumeRole` from central account only
- (Optional) Use External ID for added security

5. Trust Policy Validation

- Verify trust policy restricts access to authorized account
- Test AssumeRole from central account

6. Deployment Verification

Ensure the following resources exist: - **Lambda Functions** - Inventory Collector Lambda (fetches AWS resources) - API Lambda (reads from DynamoDB and serves UI) - **DynamoDB Table** for inventory data - **API Gateway** secured with Cognito authorizer

7. Scheduled Updates

- EventBridge rule triggers inventory Lambda daily
- Custom schedules or manual triggers supported

Frontend Deployment

8. Build Frontend

- `npm install`
- `npm run build:static`

9. Deploy to S3

- Create S3 bucket for frontend hosting
- Sync build output:
- `aws s3 sync out/ s3://<frontend-bucket> --delete`

10. CloudFront Configuration

- Create distribution with S3 as origin
- Enforce HTTPS
- Configure Origin Access Control (recommended)

11. Domain & SSL

- Create ACM certificate (us-east-1)
- Attach certificate to CloudFront
- Update Route53 DNS records

12. Cache Invalidation

- Invalidate CloudFront cache after deployment

Final Validation Checklist

- SAML login works via Cognito
- UI loads successfully
- Inventory data visible in dashboard
- DynamoDB populated with recent data
- EventBridge schedule executing successfully

Operational Notes

- Monitor Lambda and EventBridge via CloudWatch
- Enable DynamoDB PITR (recommended)
- Rotate IAM credentials and review trust policies periodically
- Consider WAF on CloudFront for production

Ownership

- Backend: Cloud / Platform Team
- Frontend: UI / Dev Team
- IAM & Security: Cloud Security Team