**Slide 1: Ephemeral Infrastructure Automation**

**Implementing Ephemeral Resources with Cloud Custodian vs. Custom Lambda Solutions**

* Presenter: [Your Name / Team]
* Date: July 2025

**Slide 2: Agenda**

1. What Is Ephemeral Infrastructure?
2. Why Automate Lifecycle Management?
3. Solution A: AWS Cloud Custodian
4. Solution B: Custom Lambda Scripts
5. Feature & Cost Comparison
6. Security & Maintenance Considerations
7. Recommendations & Next Steps

**Slide 3: What Is Ephemeral Infrastructure?**

* **Definition**: Short-lived environments automatically created and destroyed
* **Common Use Cases**
  + On-demand dev/test sandboxes
  + Feature-branch deployments
  + Temporary data pipelines
* **Key Requirement**: Reliable, automated cleanup at a defined “end time”

**Slide 4: Why Automate Lifecycle Management?**

* **Cost Control**: Avoid paying for idle or forgotten resources
* **Security Hygiene**: Reduce attack surface by removing stale infra
* **Operational Efficiency**: Eliminate manual tear-down tickets
* **Compliance**: Enforce resource retention policies consistently

**Slide 5: Solution A – AWS Cloud Custodian**

* **Policy-as-Code**: YAML definitions of resource filters + actions
* **Serverless Scheduler**: Native mode: periodic → Lambda + EventBridge
* **Built-in Filters/Actions**: Tag, age, metrics, delete, notify, etc.
* **Multi-Account Support**: Use c7n-org for org-wide enforcement

**Slide 6: How Cloud Custodian Works**

1. **Tag on Creation**: e.g. end\_time: 2025-07-03T12:00:00Z
2. **Deploy Policy**:

custodian run \

--region us-east-1 \

-s s3://custodian-bucket \

policies/delete-by-end-time.yml

1. **EventBridge Rule**: Fires every 5 minutes
2. **Lambda Execution**: Lists → filters (age ≥ 0) → actions (set-protection: False, delete)

**Slide 7: Solution B – Custom Lambda Scripts**

* **Roll-Your-Own**: Write a Lambda function (Python/Node.js)
* **Schedule**: Configure EventBridge or CloudWatch cron rule
* **Logic**: SDK calls to list CFN stacks, parse tags, then call DeleteStack
* **Deployment**: Package, upload to S3, create Lambda + EventBridge manually or via IaC

**Slide 8: How Custom Lambda Works**

1. **Write Code**:

import boto3, datetime

cfn = boto3.client('cloudformation')

# list, filter by tag:end\_time < now, then delete

1. **Deploy**:
   * Zip handler + dependencies → S3
   * aws lambda create-function …
   * aws events put-rule … & put-targets
2. **Maintain**: Update code, bump versions, redeploy

**Slide 9: Comparison Criteria**

| **Criteria** | **Cloud Custodian** | **Custom Lambda** |
| --- | --- | --- |
| **Ease of Use** | High (YAML, single CLI command) | Medium (code + manual wiring) |
| **Time to Market** | Minutes | Hours–Days |
| **Flexibility** | Built-in + extensions | Unlimited (any SDK logic) |
| **Maintenance** | Low (managed by Custodian releases) | High (custom code upkeep) |
| **Cost** | Minimal Lambda invocations + S3 | Similar Lambda invocations + dev time |
| **Multi-Account** | Native via c7n-org | Custom orchestration required |

**Slide 10: Security & Compliance**

* **Cloud Custodian**
  + IAM roles scoped to policy actions
  + Auto-managed least privilege for built-in actions
* **Custom Lambda**
  + You define IAM permissions—risk of over-privileging
  + Testing & validation of custom logic required

**Slide 11: Pros & Cons – Cloud Custodian**

**Pros**

* Declarative, policy-as-code
* Rapid deployment & updates
* Rich library of filters/actions
* Built-in scheduling & org-wide support

**Cons**

* Learning YAML schema & conventions
* May need custom actions for niche cases

**Slide 12: Pros & Cons – Custom Lambda**

**Pros**

* Full control over logic and SDK calls
* No need to learn Custodian schema

**Cons**

* Manual deployment and wiring
* Higher maintenance burden
* Re-implement features Custodian provides out-of-the-box

**Slide 13: Recommendations**

* **Start with Cloud Custodian** for most ephemeral-infra use cases
  + Leverage mode: periodic + age filters
  + Extend with custom actions only when unavoidable
* **Use custom Lambda** only if you require very specialized workflows not supported by Custodian

**Slide 14: Next Steps**

1. **Pilot**: Deploy the delete-by-end-time policy in a non-prod account
2. **Extend**: Tag CDK/CloudFormation templates to emit end\_time automatically
3. **Monitor**: Build dashboards for policy execution metrics & errors
4. **Multi-Account**: Scale with c7n-org for enterprise environments

**Slide 15: Q&A**

* Questions or feedback?
* Discuss edge cases or advanced integrations?
* Contact: you@yourcompany.com