AISA

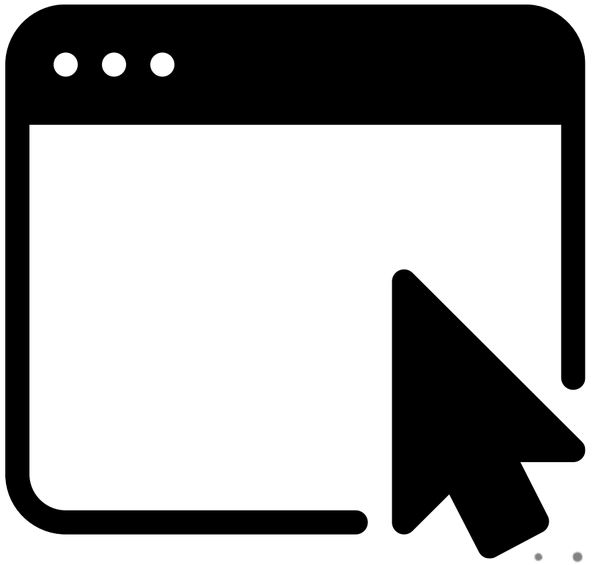
AUSTRALIAN CYBER CONFERENCE 2023

The ultimate guide to using DevSecOps to tame your cloud identity configuration

Ward van Besien
Merill Fernando

ClickOps

ClickOps is the error-prone & time-consuming process of having admins click-through various menu options in cloud providers' websites, to select and configure the correct security configuration.



OPPORTUNITIES

Active Directory (AD) and Azure AD security

88%

of impacted customers did not employ AD and Azure AD security best practices. This has become a common attack vector as attackers exploit misconfigurations and weaker security postures in critical identity systems to gain broader access and impact to businesses.

Microsoft Digital Defence Report, 2022
aka.ms/MDDR

The issues with ClickOps

- The network is no longer the control plane
- Identity is the new control plane
- Can you rely and trust on ClickOps to secure access to your organisations systems and data?

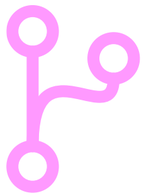
Identify and fix config drift?

Securely introduce new changes?

Backup of config?

Version history of config?

Cloud Identity

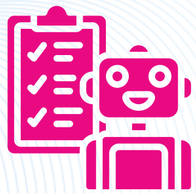


Version Control

Deploy

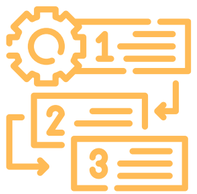
Config as Code

Monitor

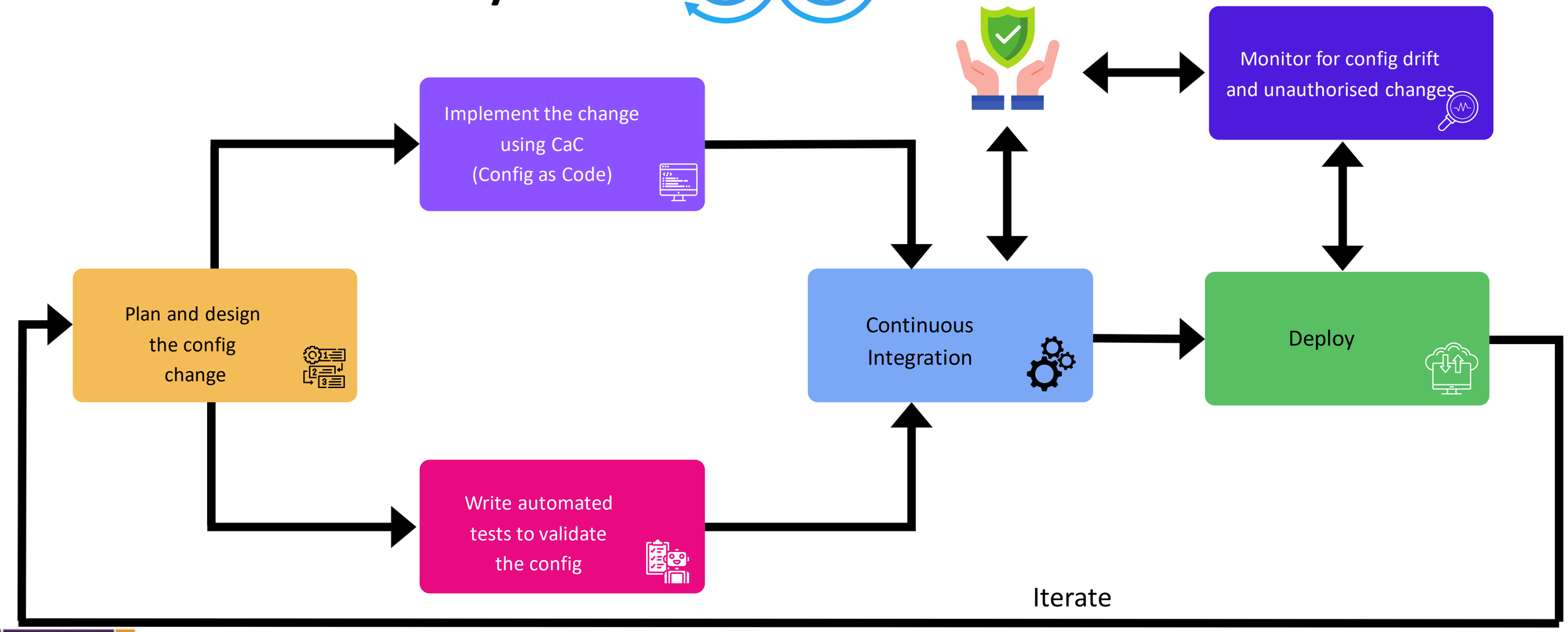


Test

Plan



Cloud Identity



Version Control

1. None / Manual code backups
2. System to Track Changes (Git etc.)
3. Unorganized branches or no branches
4. Branches created as backups or checkpoints
5. Short lived story branches that are merged to main

Maturity

GitHub

GitLab

Bitbucket

Azure Repos

Config as Code

1. No change control; ClickOps in Production
2. Use of manual run books to make changes
3. Use of ad-hoc scripts and manual checklists
4. Comprehensive, 70 to 80%, with declarative CaC where possible
5. 95% or better automation Manual run book where no API available

Maturity

HashiCorp Terraform

Pulumi

AWS CloudFormation

Microsoft Azure Resource Manager

Config as Code

Configuring an MFA policy using CaC (Config as Code) with Terraform for Okta and Microsoft Entra ID.

```
data "okta_default_policy" "example" {
  type = "MFA_ENROLL"
}

resource "okta_policy_rule_mfa" "example" {
  policy_id = data.okta_default_policy.example.id
  name      = "My Rule"
  status    = "ACTIVE"
  enroll    = "LOGIN"
  app_include {
    id   = okta_app_oauth.example.id
    type = "APP"
  }
  app_include {
    type = "APP_TYPE"
    name = "yahoo_mail"
  }
}

resource "okta_app_oauth" "example" {
  label          = "My App"
  type           = "web"
  grant_types    = ["authorization_code"]
  redirect_uris  = ["http://localhost:8000"]
  response_types = ["code"]
  skip_groups    = true
}
```

HashiCorp Terraform

okta

```
resource "azuread_conditional_access_policy" "example" {
  display_name = "example policy"
  state        = "enabled"

  conditions {
    client_app_types    = ["all"]
    user_risk_levels    = ["medium"]

    applications {
      included_applications = ["All"]
      excluded_applications = []
    }

    platforms {
      included_platforms = ["android"]
      excluded_platforms = ["iOS"]
    }

    users {
      included_users = ["All"]
      excluded_users = ["GuestsOrExternalUsers"]
    }
  }

  grant_controls {
    operator          = "OR"
    built_in_controls = ["mfa"]
  }
}
```

Microsoft Entra ID

Test

1. None / Manual Testing
2. Checklists ; Written scripts for manual testing
3. Feature Testing
4. Security testing ; Tests are stable and accurate
5. Full end-to-end regression testing continuous

Maturity

pester
POWERSHELL TESTING FRAMEWORK

Terratest

Selenium

Playwright

Test

Pester automation test to verify MFA is being applied to all users signing into the tenant.

pester
POWERSHELL TESTING FRAMEWORK

Microsoft Entra ID

```powershell
Describe "Conditional Access - Require multifactor authentication for all users" -Tag "ConditionalAccess" {

    Context "User <userPrincipalName> (<userId>)" -ForEach @( $AzureADUser ) {

        It "User using client app <_> should require MFA" -ForEach @( $clientAppType ) {

            # Definition of conditional access
            $ConditionalAccessWhatIfDefinition = @{
                "conditionalAccessWhatIfSubject"    = @{
                    "@odata.type" = "#microsoft.graph.userSubject"
                    "userId"      = "$userId"
                }
                "conditionalAccessContext"          = @{
                    "@odata.type" = "#microsoft.graph.whatifapplicationContext"
                    "authenticationContext": "c1"
                }
                "conditionalAccessWhatIfConditions" = @{
                    "clientAppType"  = "$_"
                    "devicePlatform" = "windows"
                }
            }

            $ConditionalAccessWhatIfResult = Invoke-MgGraphRequest -Method POST
              -Uri "https://graph.microsoft.com/beta/identity/conditionalAccess/evaluate"
              -Body ( $ConditionalAccessWhatIfDefinition
              |  ConvertTo-Json -Depth 99 -Compress )
              | Where-Object { $_.policyApplies -eq $true }

            $MergedControlsandAuthenticationStrength =
              $ConditionalAccessWhatIfResult.grantControls.builtInControls
              + $ConditionalAccessWhatIfResult.grantControls.authenticationStrength.requirementsSatisfied

            $MergedControlsandAuthenticationStrength | Should -Contain "fido2"
        }
    }
}
```

Deploy

1. Deploy direct to production; No Dev/Test environments
2. Test environment exists; changes tested here before prod
3. Change scripts are executed from desktop or on-prem server
4. Cloud based deployment pipeline with dev, test, & prod environments
5. Release pipeline with code scanning and gated approval workflows

Maturity

Octopus Deploy

Bamboo

AWS CodePipeline

Jenkins

VERACODE

Azure Pipelines

GitHub Actions

Microsoft Defender for DevOps

sonarqube

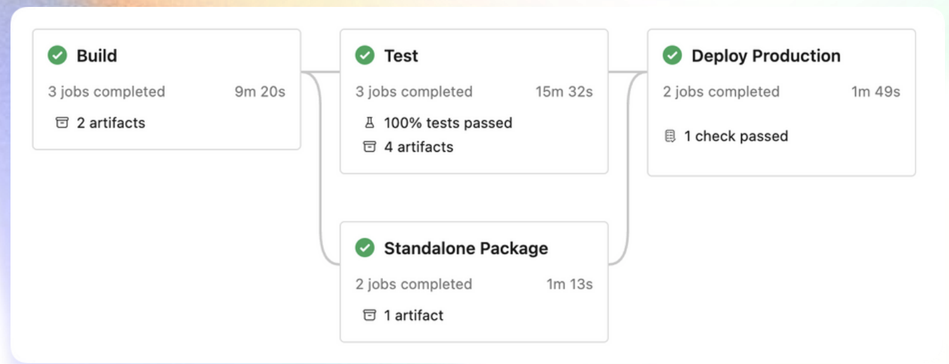
aqua trivy

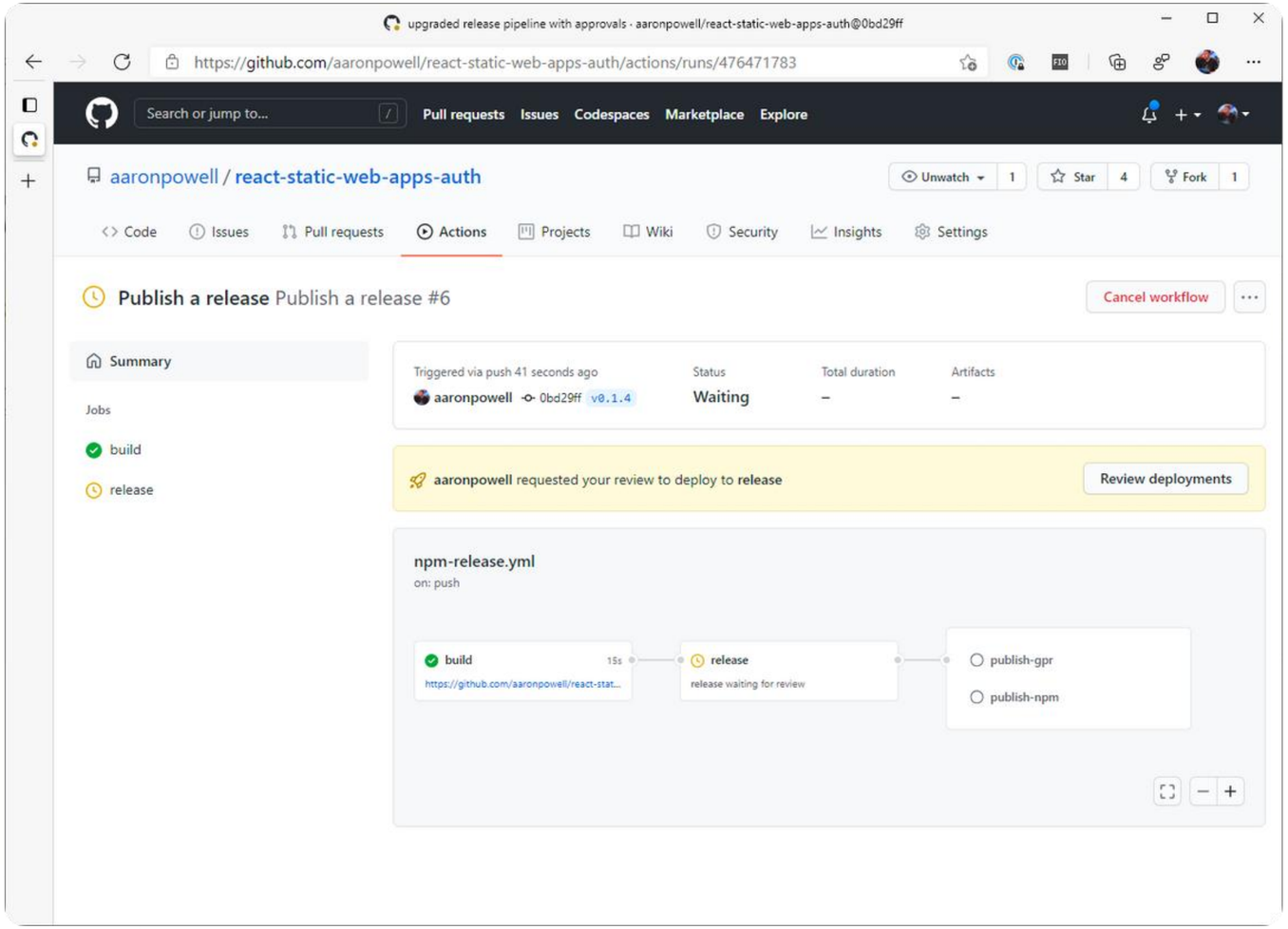
DevOps Tenant Model

	Dev	Test	Prod
On Prem Directory	Dev Active Directory / LDAP Instance	Test/Staging Active Directory / LDAP Instance	Production Active Directory / LDAP Instance
Cloud Directory	okta, Ping Identity, Microsoft Entra ID	okta, Ping Identity, Microsoft Entra ID	okta, Ping Identity, Microsoft Entra ID
Global Admin Access	Dev, InfoSec, IAM Team	InfoSec, IAM Team (< 10 admins)	IAM Team (3-5 admins)
Change Control	Low (Manual changes)	Medium	High
Optimised for...	Innovation	Troubleshooting	Reliability + Security

Deploy

Multi-environment deployments requiring manual approval to progress to next environment.





OPPORTUNITIES

Monitor

1	2	3	4	5
Minimal or no monitoring for config and tenant drift	Batched based monitoring via manual processes.	Automated daily regression tests with <50% coverage	Automated daily regression tests with >50% coverage	Monitoring audit logs and alerting for privileged actions

Maturity

Tests

+

Daily Continuous Integration

pester POWERSHELL TESTING FRAMEWORK

Playwright

Selenium

Terratest

AWS CodePipeline

GitHub Actions

Octopus Deploy

Jenkins

Azure Pipelines

Bamboo

DATADOG

splunk>

Amazon Cloudwatch

Microsoft Sentinel

sumo logic

OPPORTUNITIES

This is daunting.

Where and how do I start?

#1 Tests

Start writing tests for your current config and use monitoring to execute them daily

#2 Test new config

When new config changes are introduced write a test to ensure it stays on.

#3 Start small

Don’t try to boil the ocean on day #1.
Start with key security config maybe ~10 checks and expand over time.

#4 Be pragmatic

It’s not easy to automate all changes, be pragmatic about what you can automate.

AISA

THANK YOU

@merill | merill.net | entra.news