Zero Trust Security and Identity Management with Boundary

Suman Chakraborty
(He/Him)

VMware

HashiCorp

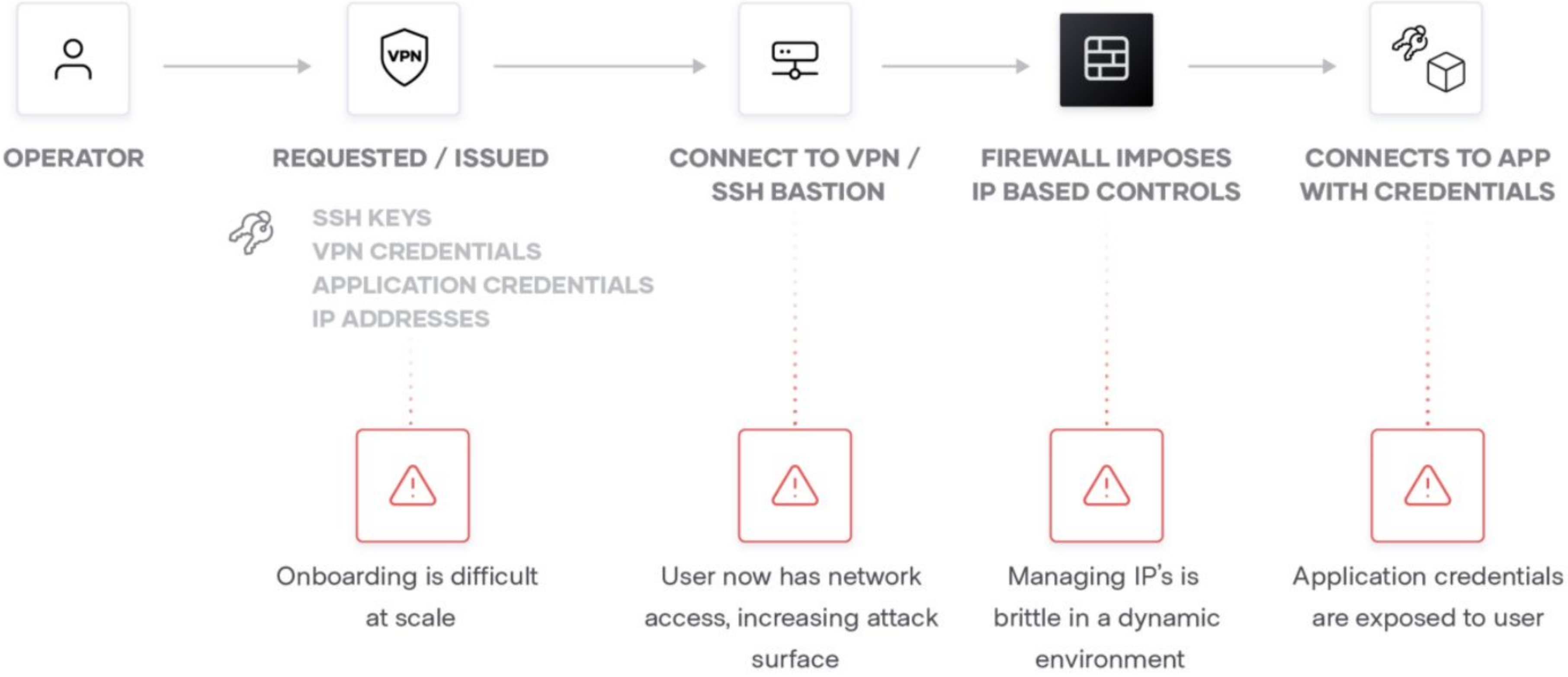
$ whoami

- Senior Cloud Native Architect @ VMware
- Speaker at Open Source Summit (LF), Hashitalks (2021), Devops India Summit, Docker India Conference
- Involved in tech community meet-ups and talks around DevOps, Cloud-Native tools, Kubernetes & Serverless technologies
- Traveller & Foodie 🙂

Agenda

- Understanding the traditional workflow for identity management
- Challenges with the current model
- How Boundary aims to solve the challenges in current access workflow
- Understanding the Boundary workflow
- Demo

Traditional Workflow



OPERATOR

REQUESTED / ISSUED

SSH KEYS
VPN CREDENTIALS
APPLICATION CREDENTIALS
IP ADDRESSES

CONNECT TO VPN / SSH BASTION

FIREWALL IMPOSES IP BASED CONTROLS

CONNECTS TO APP WITH CREDENTIALS

Onboarding is difficult at scale

User now has network access, increasing attack surface

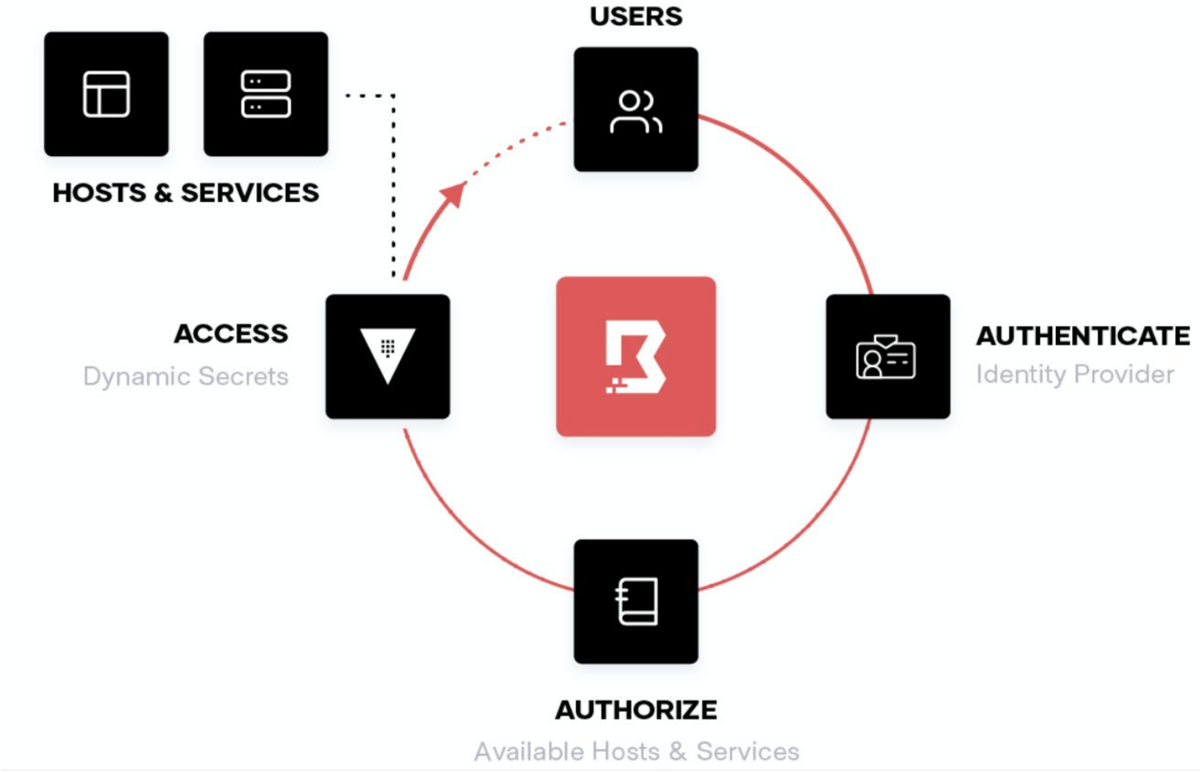
Managing IP's is brittle in a dynamic environment

Application credentials are exposed to user

Challenges with current model

- Offers a wider privilege connecting to systems in a private network
- Not suited for the cloud with highly ephemeral and dynamic environments
- Multiple credentials need to be shared which exposes security threat
- Scaling the solutions as workforces and infrastructure grow creates additional pain points and complexity for administrators to manage.
- Managing internal firewalls is time consuming and wasteful
- User de-boarding is a complex process and is barely manageable for larger environments

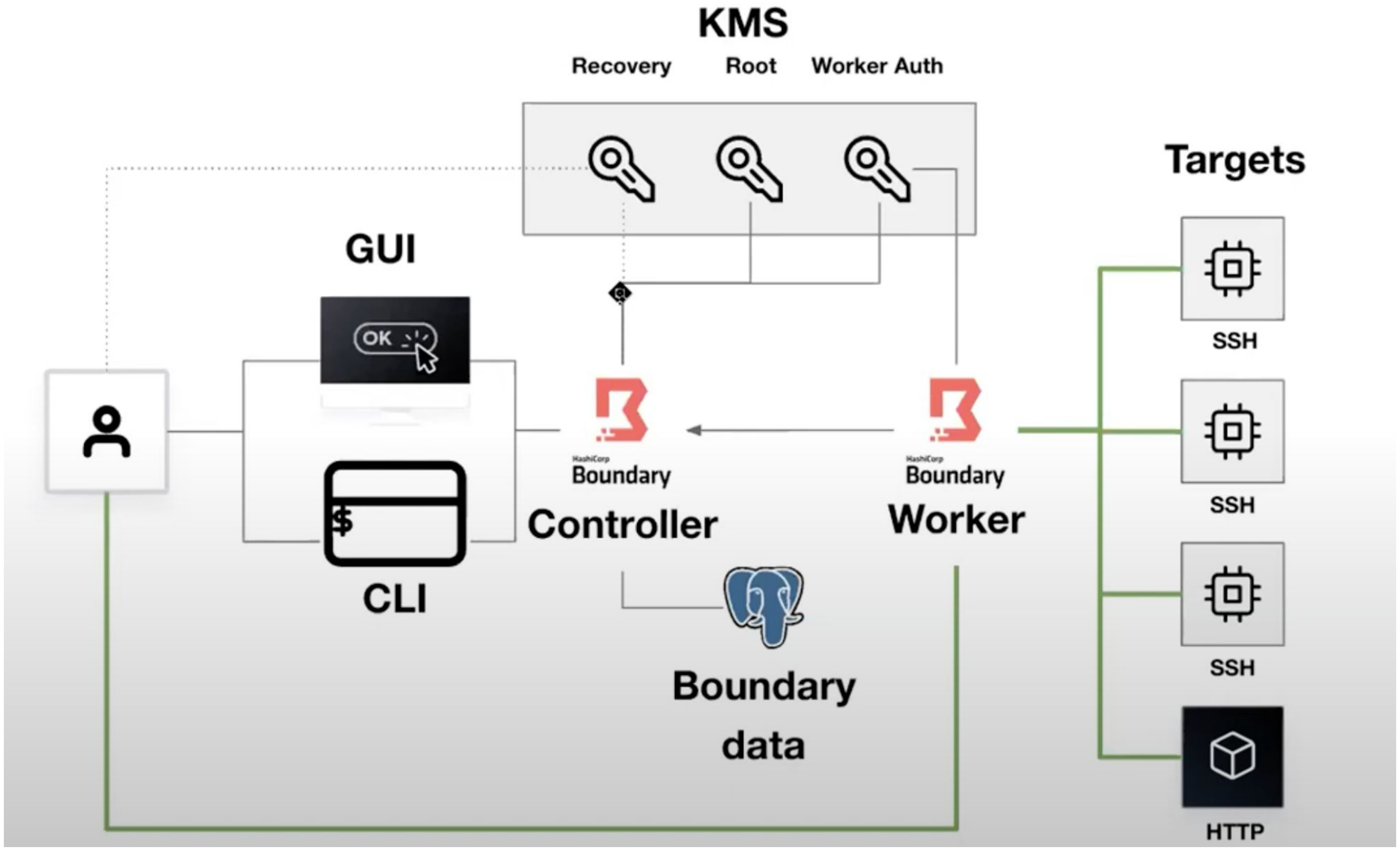
Boundary to the rescue !!



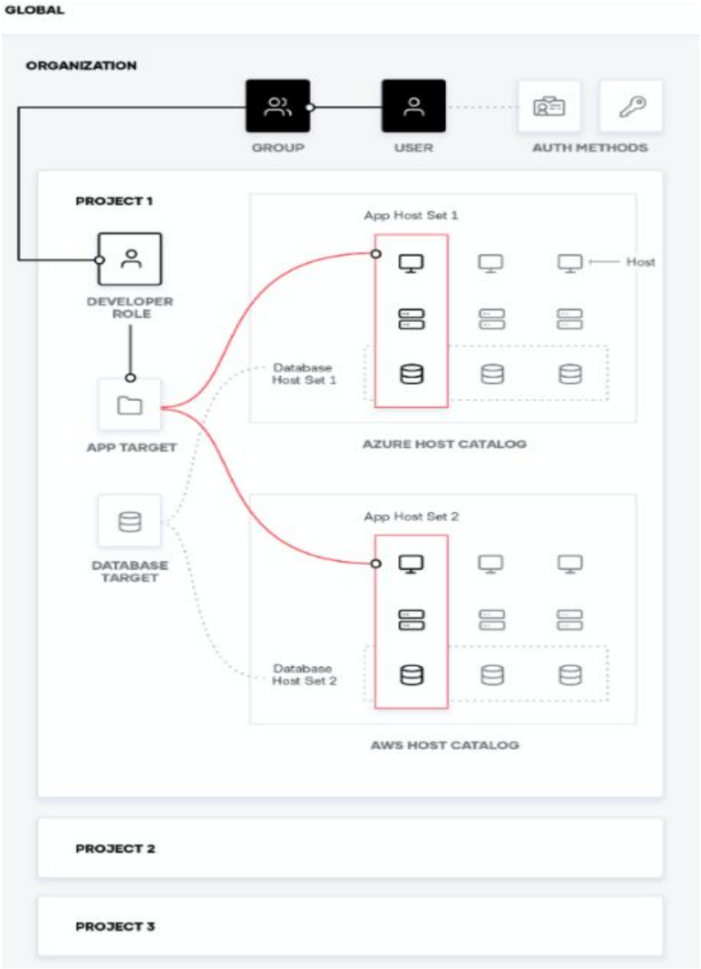
Boundary offers a secure access

- Identity management through role-based access control
- Access automation
- Sessions visibility

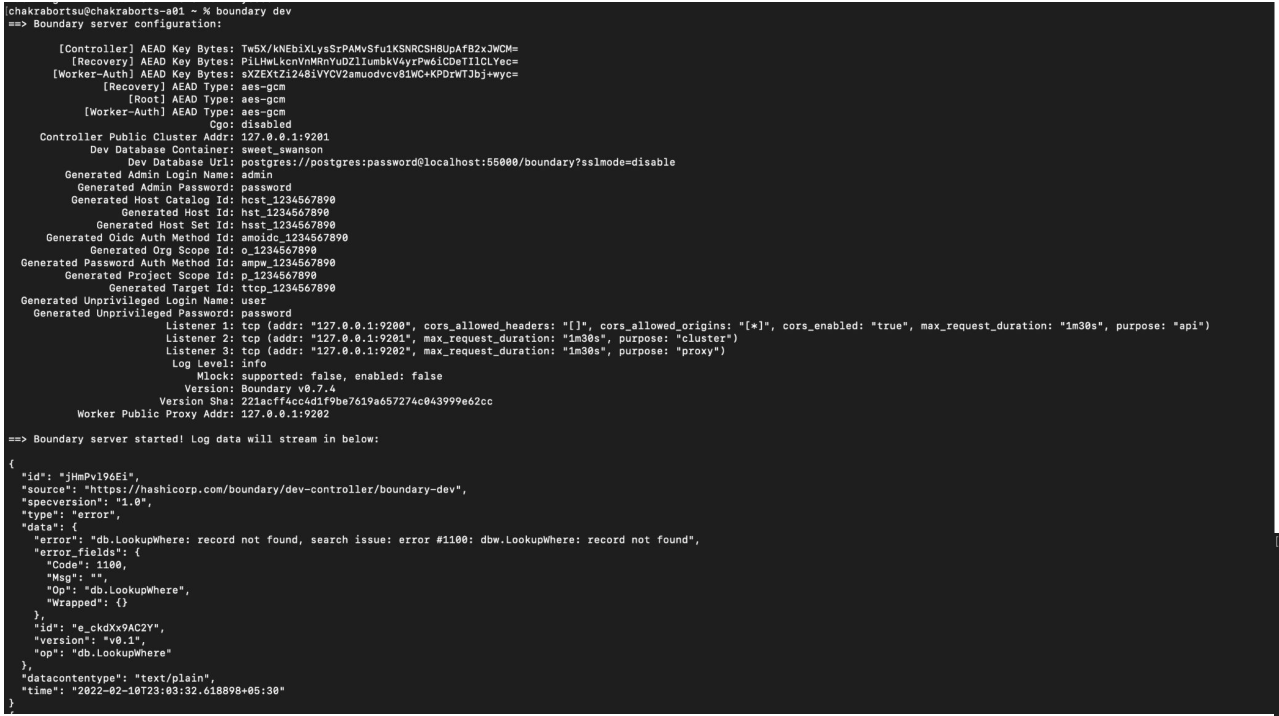
Inside Boundary



Scope Management in Boundary



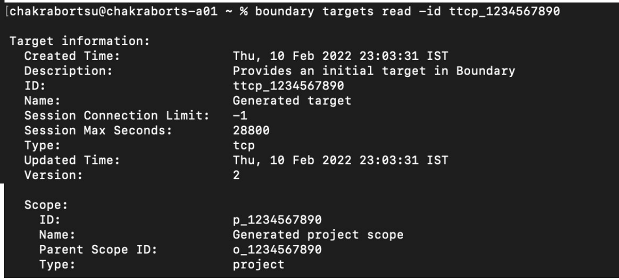
Boundary access and session management



```
[chakrabortsu@chakraborts-a01 ~ % boundary dev
==> Boundary server configuration:

        [Controller] AEAD Key Bytes: Tw5X/kNEbiXLysSrPAMvSfu1KSNRCSH8UpAfB2xJWCM=
          [Recovery] AEAD Key Bytes: PiLHwLkcnVnMRnYuDZlIumbkV4yrPw6iCDeTIlCLYec=
       [Worker-Auth] AEAD Key Bytes: sXZEXtZi248iVYCV2amuodvcv81WC+KPDrWTJbj+wyc=
               [Recovery] AEAD Type: aes-gcm
                   [Root] AEAD Type: aes-gcm
            [Worker-Auth] AEAD Type: aes-gcm
                                Cgo: disabled
     Controller Public Cluster Addr: 127.0.0.1:9201
             Dev Database Container: sweet_swanson
                   Dev Database Url: postgres://postgres:password@localhost:55000/boundary?sslmode=disable
         Generated Admin Login Name: admin
           Generated Admin Password: password
          Generated Host Catalog Id: hcst_1234567890
                  Generated Host Id: hst_1234567890
              Generated Host Set Id: hsst_1234567890
      Generated Oidc Auth Method Id: amoidc_1234567890
             Generated Org Scope Id: o_1234567890
  Generated Password Auth Method Id: ampw_1234567890
         Generated Project Scope Id: p_1234567890
                Generated Target Id: ttcp_1234567890
  Generated Unprivileged Login Name: user
    Generated Unprivileged Password: password
                         Listener 1: tcp (addr: "127.0.0.1:9200", cors_allowed_headers: "[]", cors_allowed_origins: "[*]", cors_enabled: "true", max_request_duration: "1m30s", purpose: "api")
                         Listener 2: tcp (addr: "127.0.0.1:9201", max_request_duration: "1m30s", purpose: "cluster")
                         Listener 3: tcp (addr: "127.0.0.1:9202", max_request_duration: "1m30s", purpose: "proxy")
                          Log Level: info
                              Mlock: supported: false, enabled: false
                            Version: Boundary v0.7.4
                        Version Sha: 221acff4cc4d1f9be7619a657274c043999e62cc
           Worker Public Proxy Addr: 127.0.0.1:9202

==> Boundary server started! Log data will stream in below:

{
  "id": "jHmPvl96Ei",
  "source": "https://hashicorp.com/boundary/dev-controller/boundary-dev",
  "specversion": "1.0",
  "type": "error",
  "data": {
    "error": "db.LookupWhere: record not found, search issue: error #1100: dbw.LookupWhere: record not found",
    "error_fields": {
      "Code": 1100,
      "Msg": "",
      "Op": "db.LookupWhere",
      "Wrapped": {}
    },
    "id": "e_ckdXx9AC2Y",
    "version": "v0.1",
    "op": "db.LookupWhere"
  },
  "datacontenttype": "text/plain",
  "time": "2022-02-10T23:03:32.618898+05:30"
}
```

Initiating boundary session

Reading the target host information



```
[chakrabortsu@chakraborts-a01 ~ % boundary targets read -id ttcp_1234567890

Target information:
  Created Time:               Thu, 10 Feb 2022 23:03:31 IST
  Description:                Provides an initial target in Boundary
  ID:                         ttcp_1234567890
  Name:                       Generated target
  Session Connection Limit:   -1
  Session Max Seconds:        28800
  Type:                       tcp
  Updated Time:               Thu, 10 Feb 2022 23:03:31 IST
  Version:                    2

  Scope:
    ID:                       p_1234567890
    Name:                     Generated project scope
    Parent Scope ID:          o_1234567890
    Type:                     project
```

WAKE UP...
IT'S DEMO TIME!
memegenerator.net

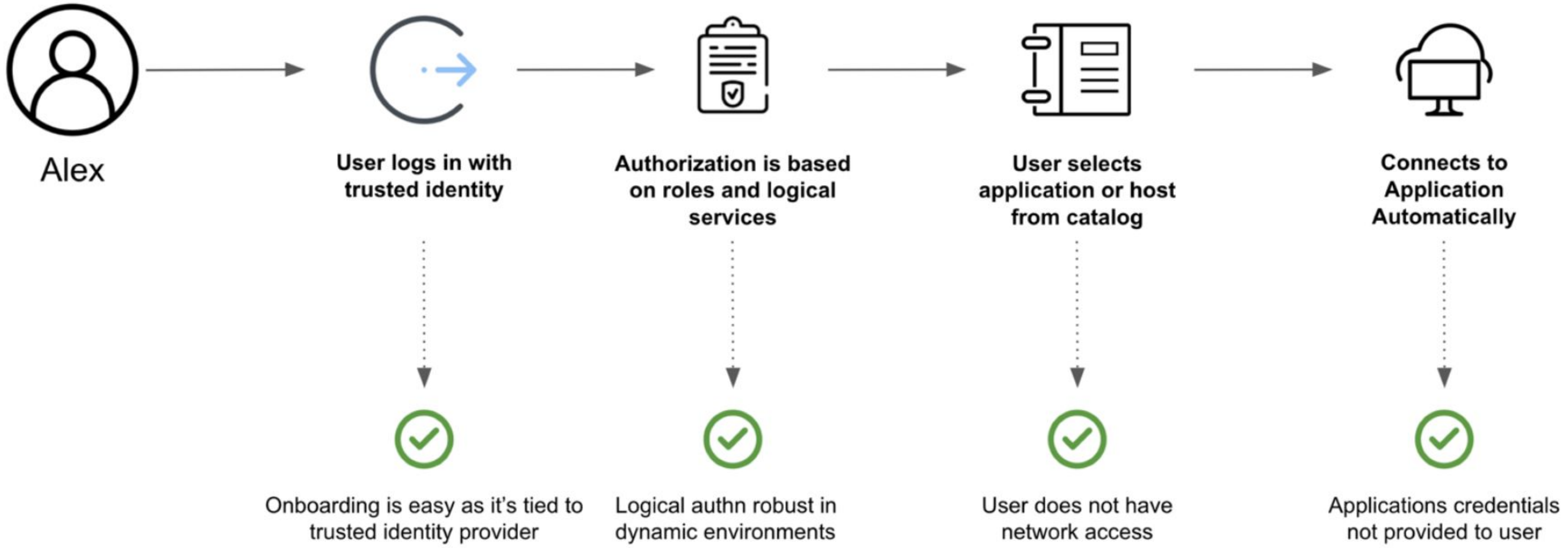
DEMO SCENARIOS

Scenario 1: - Understanding boundary UI and walkthrough common setup

Scenario 2: - Walkthrough automation with Terraform and Boundary

Type	Name	Remarks
Organization	hashitaks_corp	New Organization
Users	Multiple ~ 4	Jose, Joe, Bill, Jai
Group	read-only	Group with 3 users
Roles	multiple	Read-only & Admin
Auth Method	Corp Password	New Auth Method password
Project	core_infra	New project with hashitalks_corp
Host Catalog	backend_servers	Host catalog with one host set
Host Set	backend_servers_ssh	Host set with 2 hosts
Targets	Multiple	ssh_server & backend_server

How Boundary addresses existing problem!



Alex

User logs in with trusted identity

Onboarding is easy as it's tied to trusted identity provider

Authorization is based on roles and logical services

Logical authn robust in dynamic environments

User selects application or host from catalog

User does not have network access

Connects to Application Automatically

Applications credentials not provided to user

Resources

- Boundary official docs - https://www.boundaryproject.io/
- HashiCorp videos - https://www.youtube.com/watch?v=tUMe7EsXYBQ

Thank You !!!

https://www.linkedin.com/in/schakraborty007/

itsmesumanc