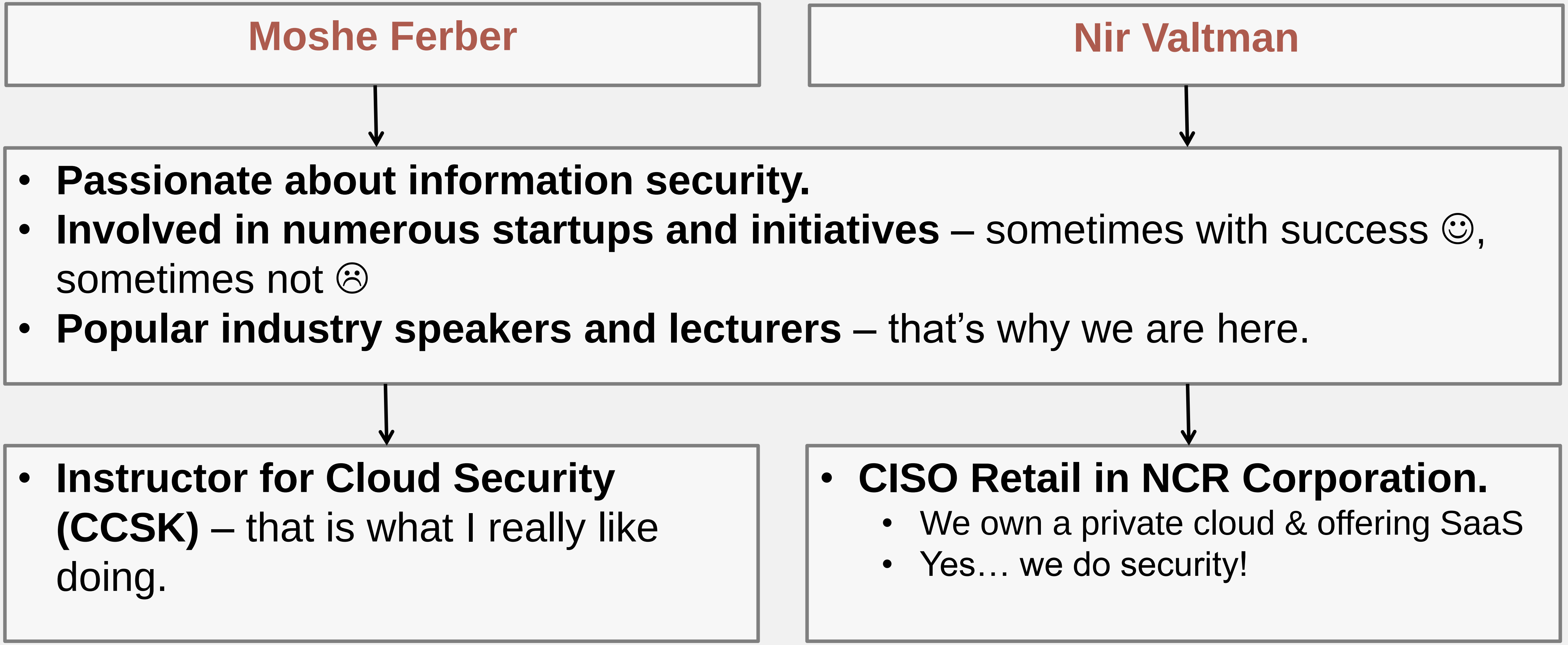
From Zero to Secure in 1 Minute

Securing IaaS

Nir Valtman & Moshe Ferber

Black Hat Asia 2015

About us



Moshe Ferber

Nir Valtman

- **Passionate about information security.**
- **Involved in numerous startups and initiatives** – sometimes with success ☺, sometimes not ☹
- **Popular industry speakers and lecturers** – that's why we are here.

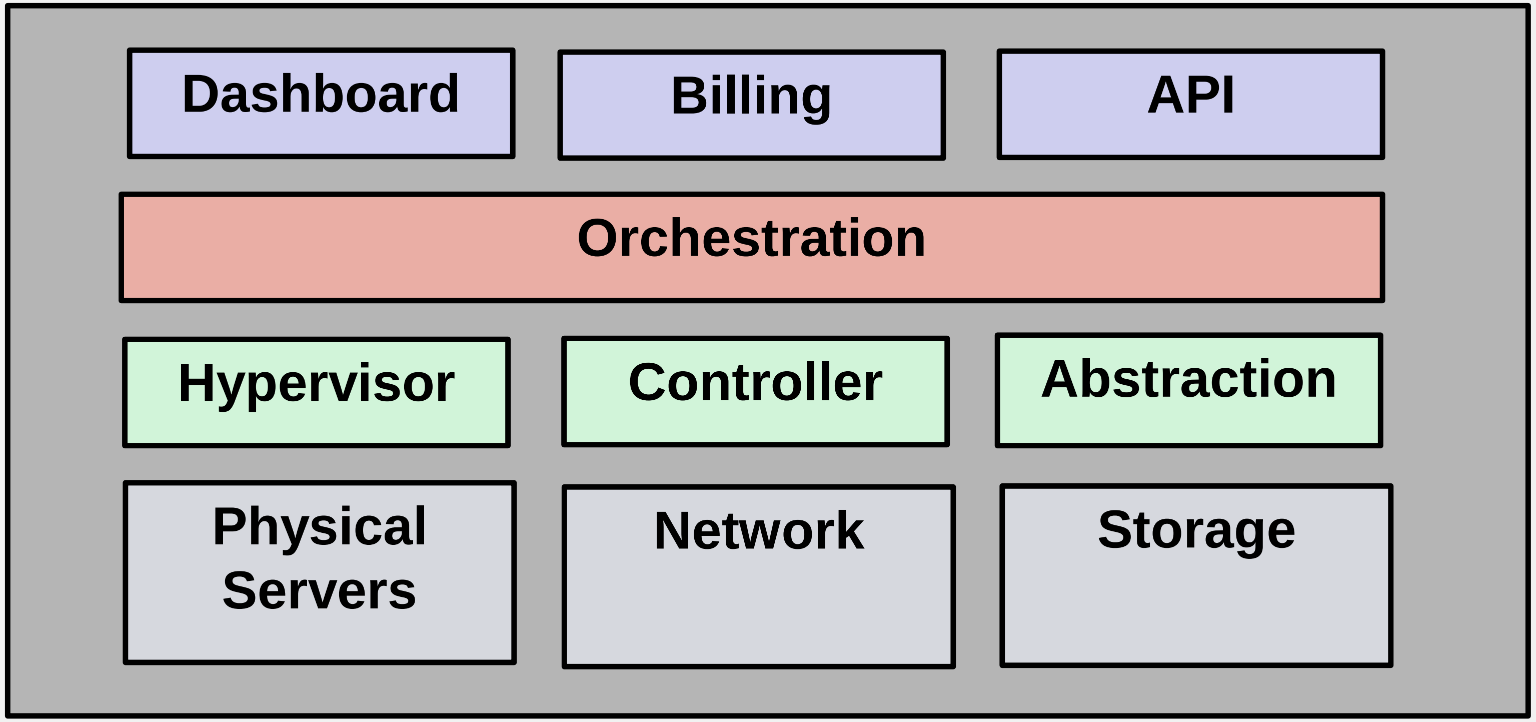
- **Instructor for Cloud Security (CCSK)** – that is what I really like doing.

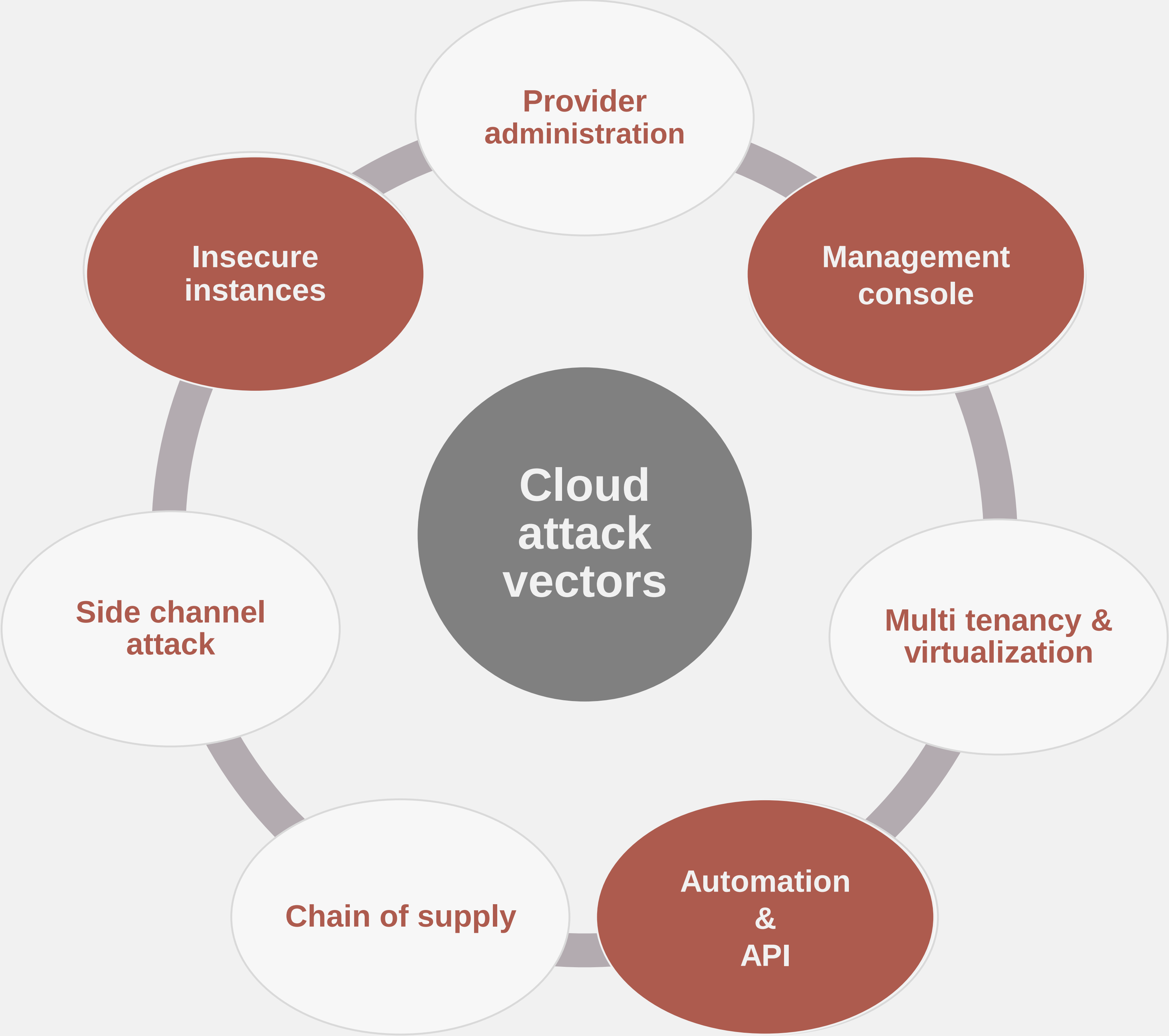
- **CISO Retail in NCR Corporation.**
  - We own a private cloud & offering SaaS
  - Yes… we do security!

About the talk

Cloud security challenges and benefits

And more specifically, using IaaS automation and orchestration features for increasing security on our servers.





Cloud attack vectors

Provider administration

Management console

Multi tenancy & virtualization

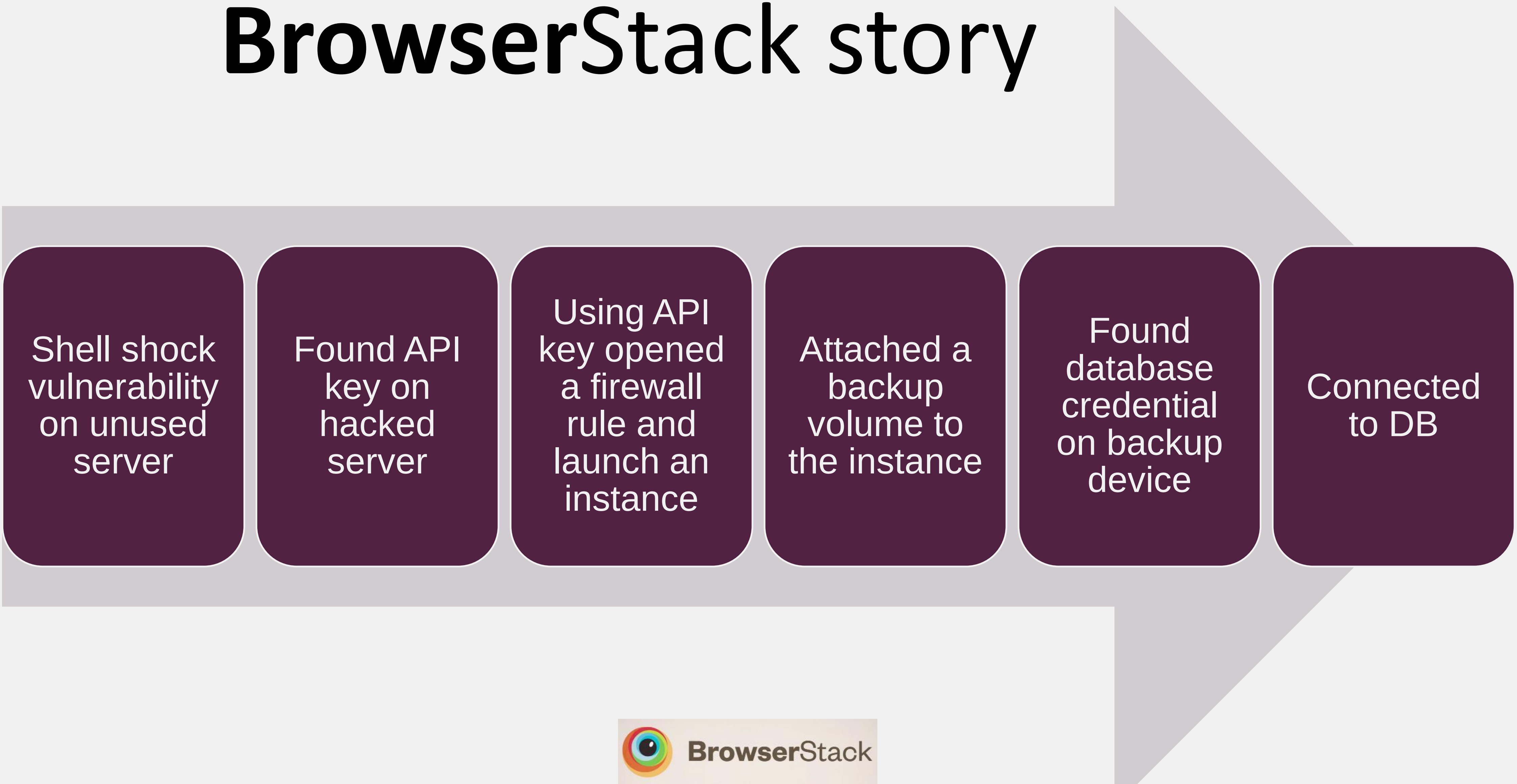
Automation & API

Chain of supply

Side channel attack

Insecure instances

Anatomy of a cloud hack – **Browser**Stack story

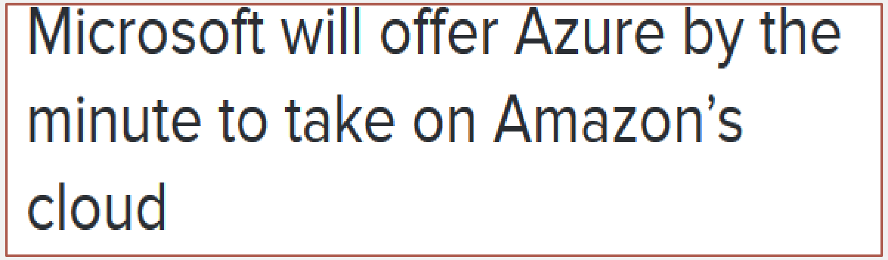


- Shell shock vulnerability on unused server
- Found API key on hacked server
- Using API key opened a firewall rule and launch an instance
- Attached a backup volume to the instance
- Found database credential on backup device
- Connected to DB

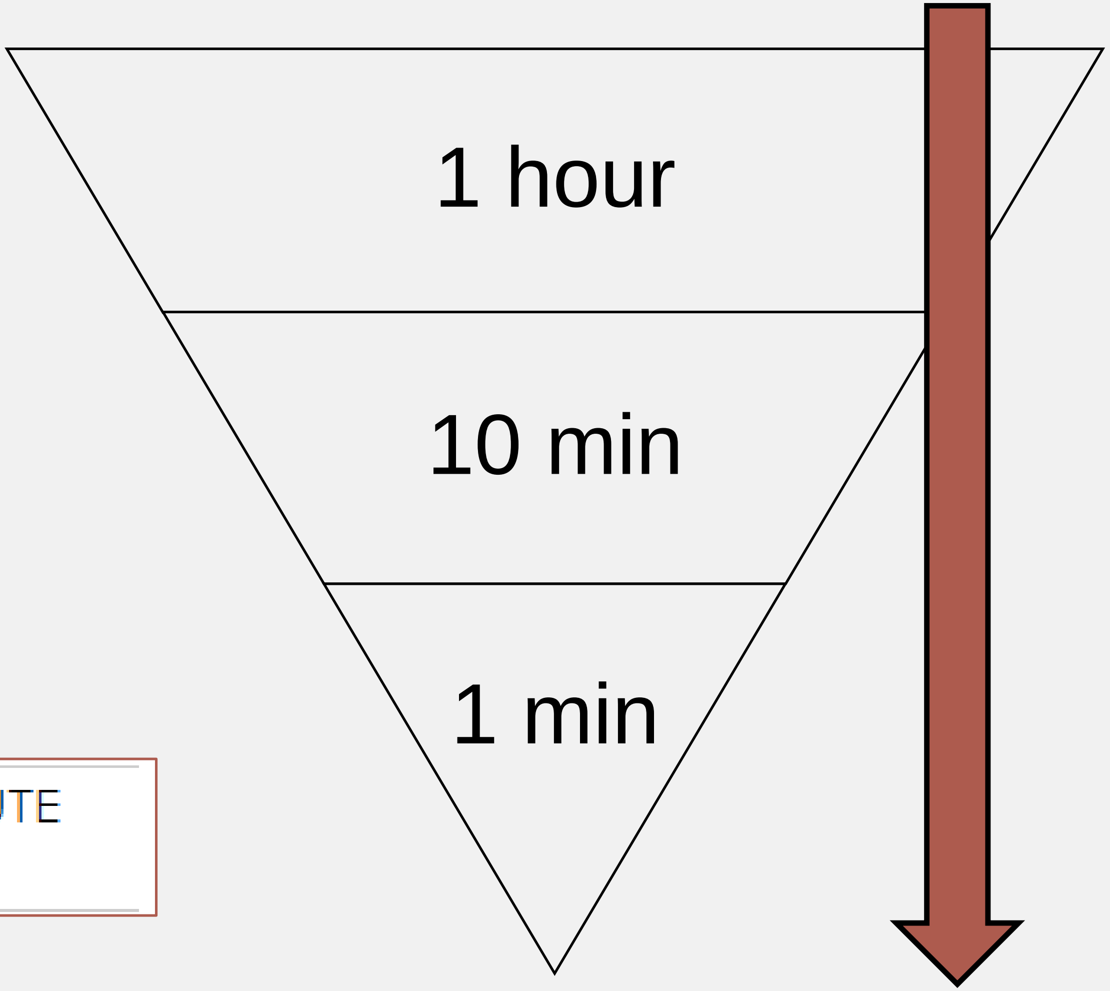
BrowserStack

The billing cycle is reducing

Google slashes cloud platform prices … again


Microsoft will offer Azure by the minute to take on Amazon's cloud

MICROSOFT FOLLOWS GOOGLE WITH BY-THE-MINUTE CLOUD BILLING



How to do security when servers alive for 10 minutes?

Patch management

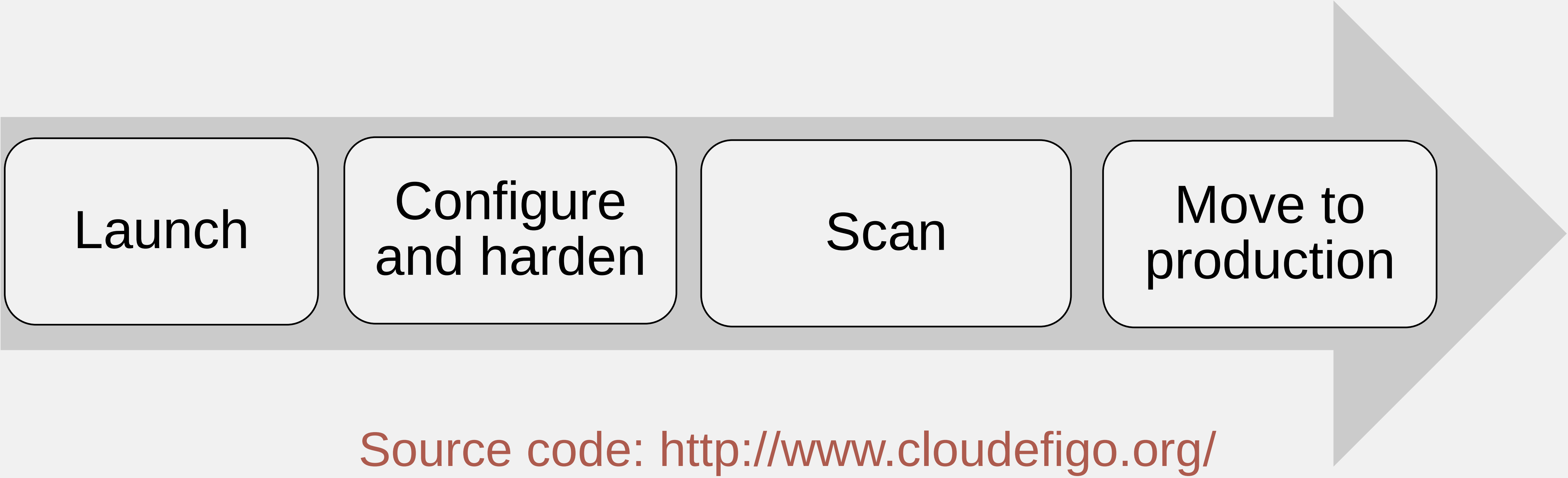
Maintenance windows

Periodic vulnerability scanning

Hardening

Introducing Cloudefigo

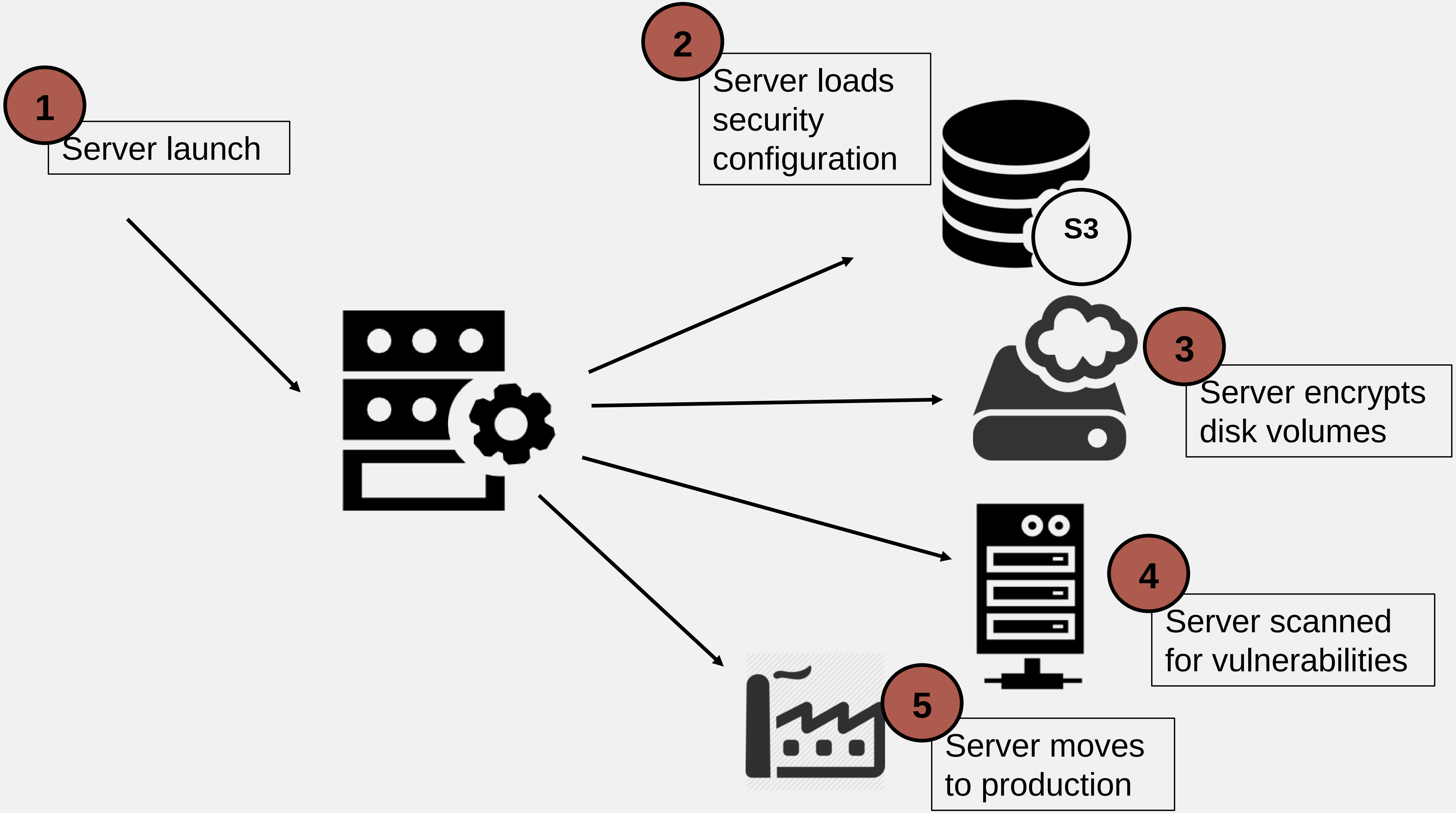
A fully automatic tool for:



Source code: http://www.cloudefigo.org/

Based on the work made by Rich Mogull from Securosis
https://github.com/rmogull/PragmaticNetSecManagement

Cloudefigo Lifecycle



1 Server launch

2 Server loads security configuration

3 Server encrypts disk volumes

4 Server scanned for vulnerabilities

5 Server moves to production

Components

Object Storage - AWS S3

A storage architecture that manage data as object. Files are stored along with metadata and unique identifier. Access is usually by HTTP/S.

Security Scanner

We use Nessus since it's very popular. There are commercial products with built in integration to AWS though:
http://www.tenable.com/products/nessus

CloudInit

CloudInit is a package (originally introduced by Ubuntu) that handles early initialization of a cloud instance.

Configuration management

We have used Chef, because it is open source and very integrative to our environment.
https://www.getchef.com/.

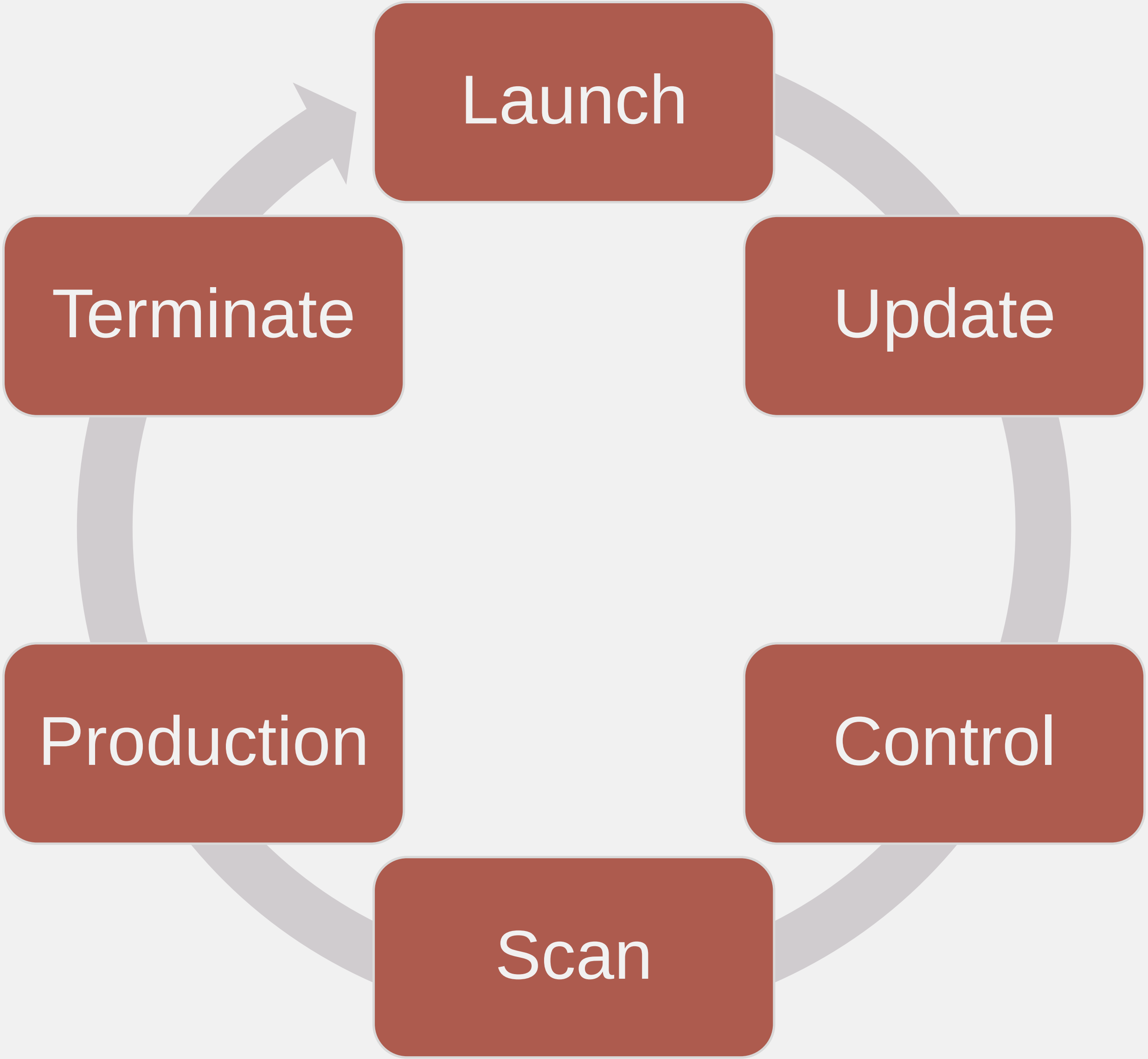
AWS IAM Role

IAM roles provide permissions for resources. Instances can be assign with an IAM Roles that will determine which resources inside AWS the instance can access.

Encryption

We have used Full Disk Encryption open source software called "dm-crypt".
https://code.google.com/p/cryptsetup/wiki/DMCrypt

Instance Lifecycle



Launch

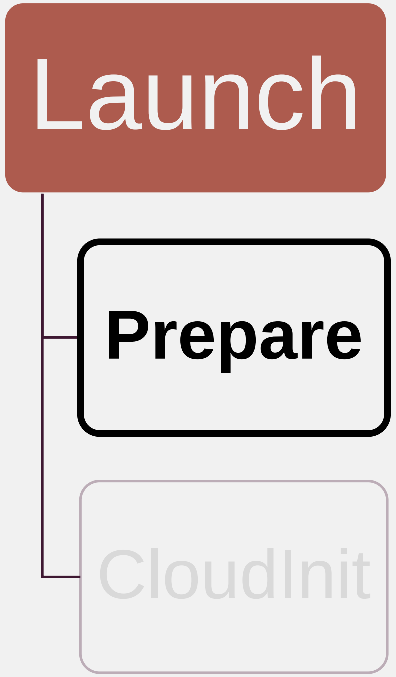
Update

Control

Scan

Production

Terminate

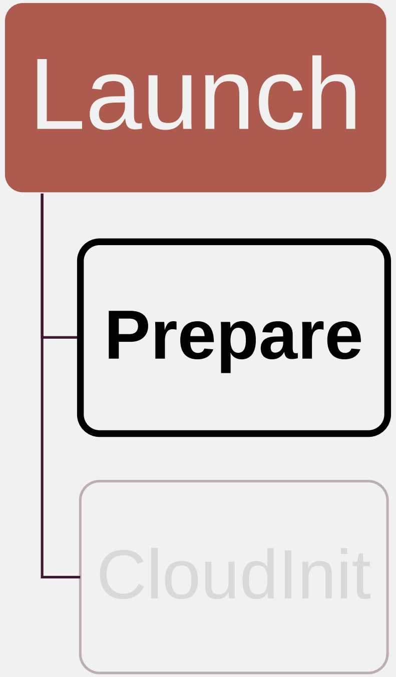


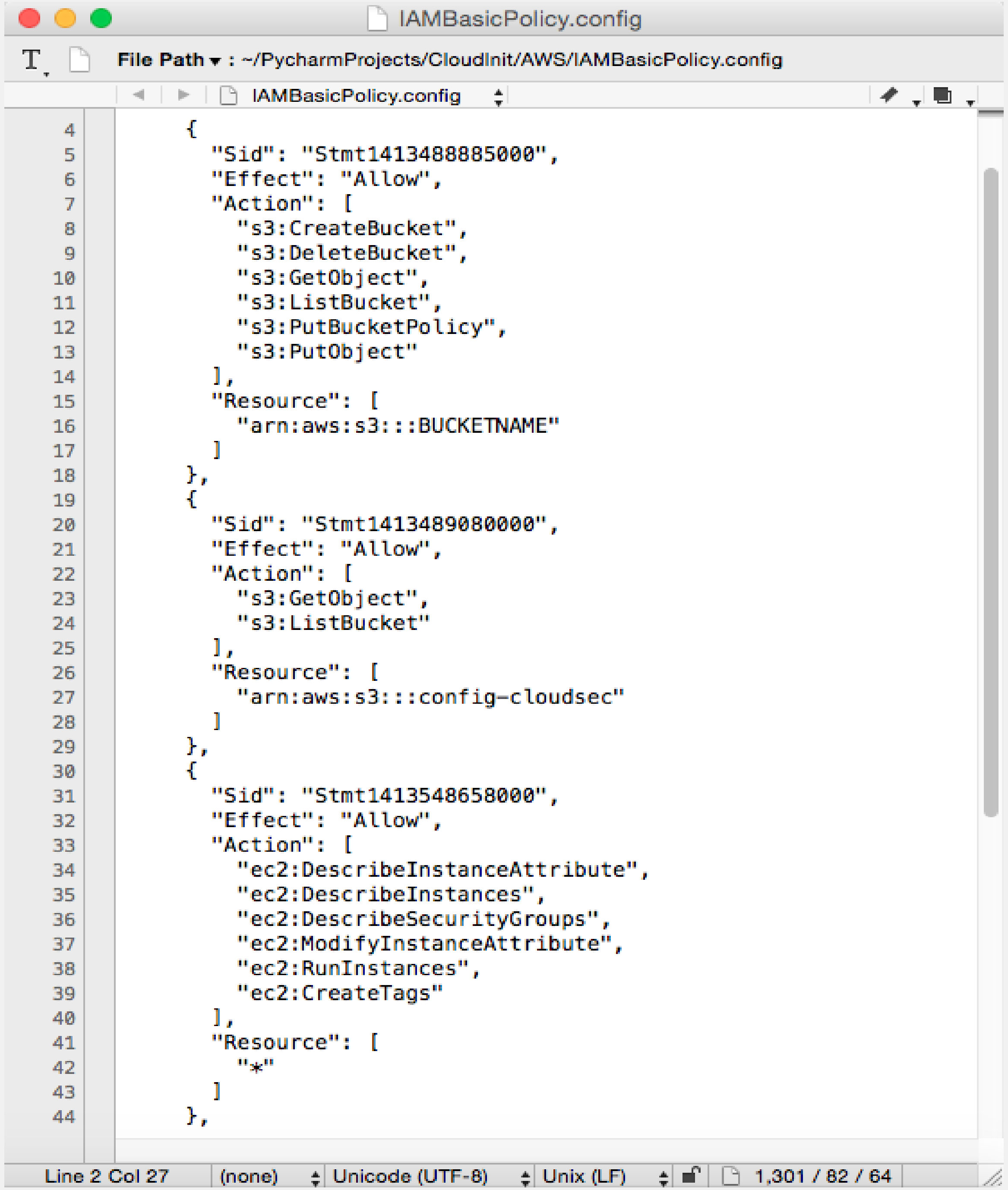
Launch

Prepare

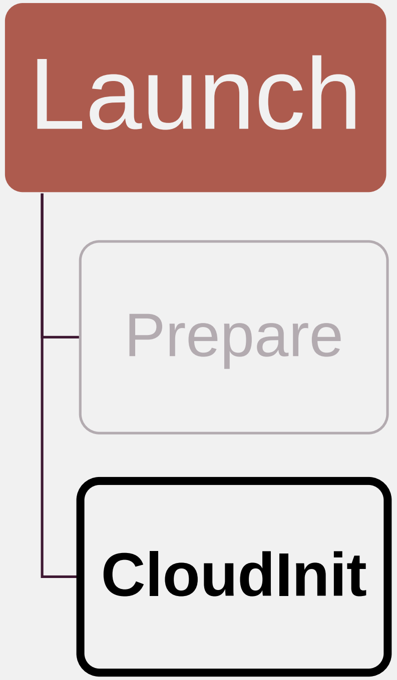
CloudInit

- Each machine manage its own attributes
  - Encryption keys
  - Remediation vs production groups.
- Management of these attributes require permissions.
- Permissions during launch > production
- Thus, a dynamic IAM role is required.





```
    {
      "Sid": "Stmt1413488885000",
      "Effect": "Allow",
      "Action": [
        "s3:CreateBucket",
        "s3:DeleteBucket",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:PutBucketPolicy",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::BUCKETNAME"
      ]
    },
    {
      "Sid": "Stmt1413489080000",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::config-cloudsec"
      ]
    },
    {
      "Sid": "Stmt1413548658000",
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstanceAttribute",
        "ec2:DescribeInstances",
        "ec2:DescribeSecurityGroups",
        "ec2:ModifyInstanceAttribute",
        "ec2:RunInstances",
        "ec2:CreateTags"
      ],
      "Resource": [
        "*"
      ]
    },
```

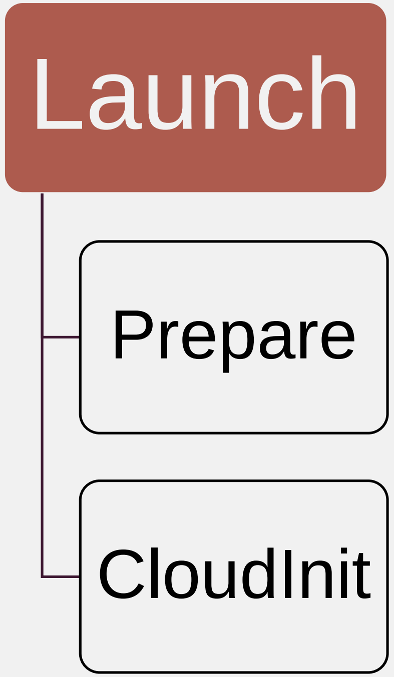


Launch

Prepare

CloudInit

- Executed in root permissions when image is launching.
- Responsible for building the infrastructure for the following steps.



Launch

Prepare

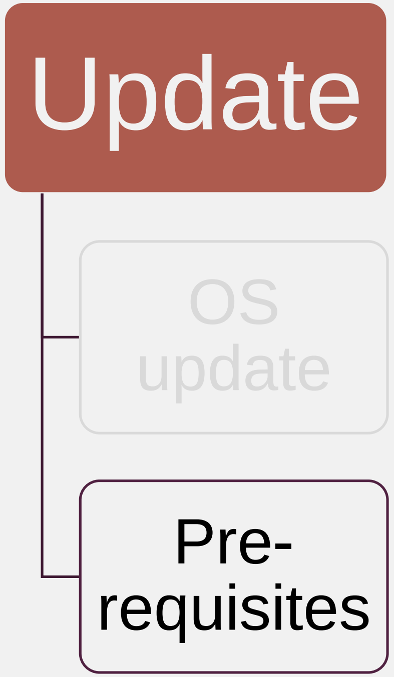
CloudInit

Update

OS update

Pre-requisites

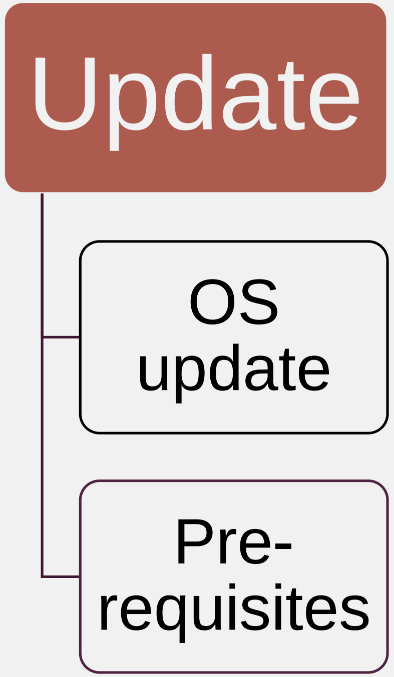
- CloudInit to update & upgrade software packages.
- Primary goal is to make sure the cloud instance is secure once upgraded.



- CloudInit to install the software packages required to operate:
  - Python + pip + wheel.
  - AWS SDK (Boto)
  - Chef Client + Chef SDK (PyChef)

- Download configurations and scripts from S3:
  - Cloudefigo script.
  - Chef client initialization files.

- Cloudinit to create and attach a volume for application files and data.



Update

- OS update
- Pre-requisites

Control

Chef Registration

Encrypt

- The Chef clients register to the Chef Management server using the initialization files loaded from S3.
- Once the client is registered, a policy is loaded and enforced on the instance.

Control

Chef Registration

Encrypt

- The volume to be encrypted using randomly generated key.
  - The key is kept in S3 for later use.

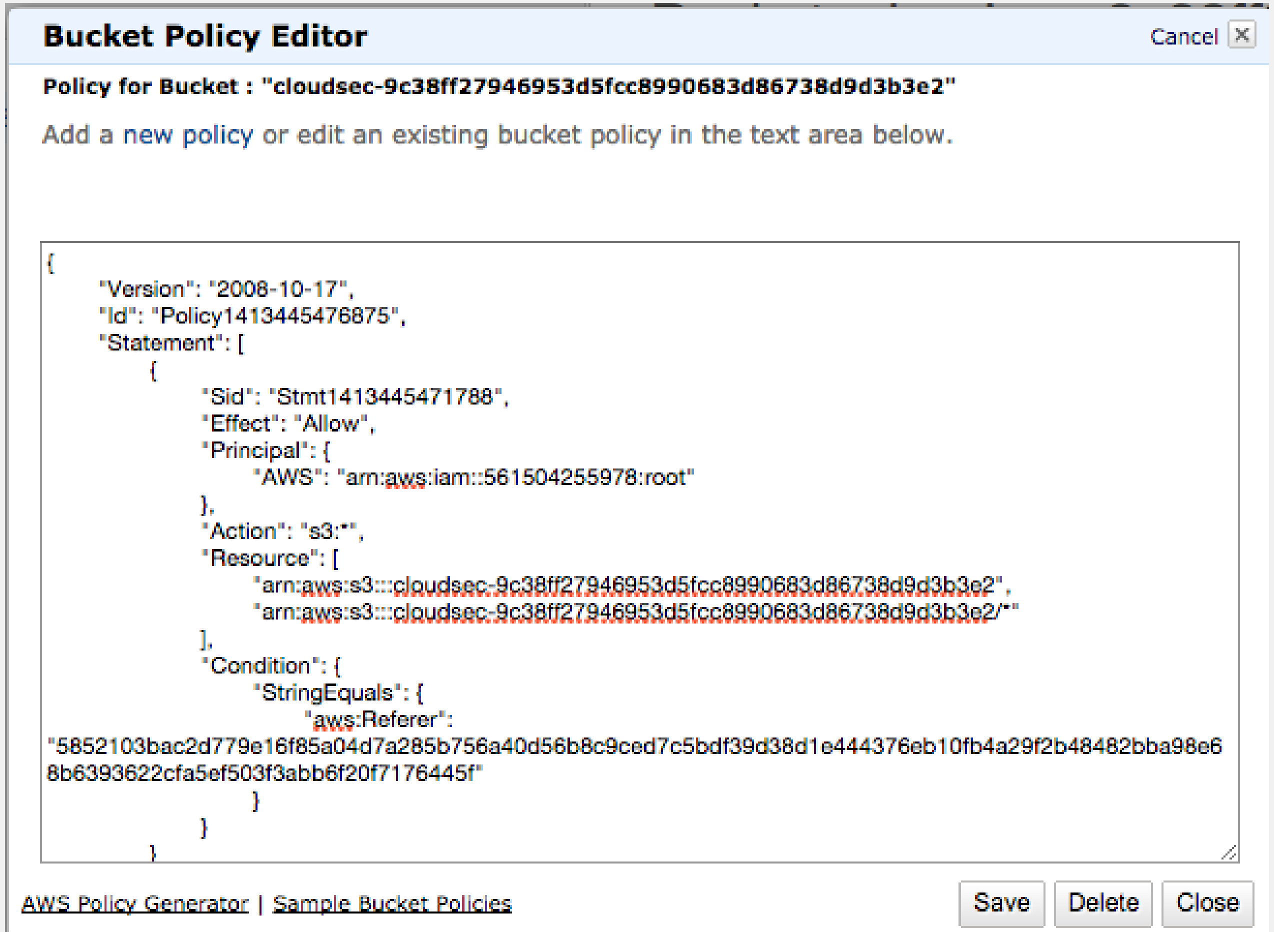
- The application database to be installed in the encrypted volume.

Control

Chef Registration

Encrypt

- Dynamic S3 policy: access to key require a referrer header that is generated based on attributes from the instance.

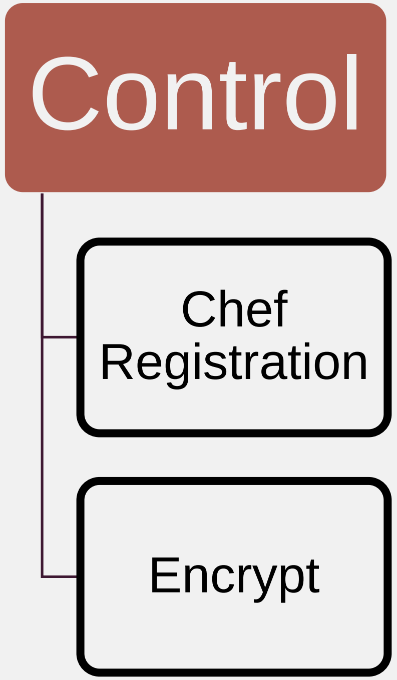


Bucket Policy Editor

Policy for Bucket : "cloudsec-9c38ff27946953d5fcc8990683d86738d9d3b3e2"

Add a new policy or edit an existing bucket policy in the text area below.

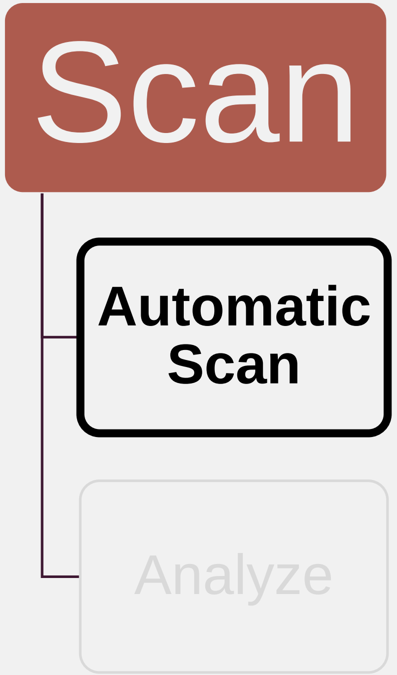
```
{
        "Version": "2008-10-17",
        "Id": "Policy1413445476875",
        "Statement": [
                {
                        "Sid": "Stmt1413445471788",
                        "Effect": "Allow",
                        "Principal": {
                                "AWS": "arn:aws:iam::561504255978:root"
                        },
                        "Action": "s3:*",
                        "Resource": [
                                "arn:aws:s3:::cloudsec-9c38ff27946953d5fcc8990683d86738d9d3b3e2",
                                "arn:aws:s3:::cloudsec-9c38ff27946953d5fcc8990683d86738d9d3b3e2/*"
                        ],
                        "Condition": {
                                "StringEquals": {
                                        "aws:Referer":
"5852103bac2d779e16f85a04d7a285b756a40d56b8c9ced7c5bdf39d38d1e444376eb10fb4a29f2b48482bba98e6
8b6393622cfa5ef503f3abb6f20f7176445f"
                                }
                        }
                }
```



Control

Chef Registration

Encrypt

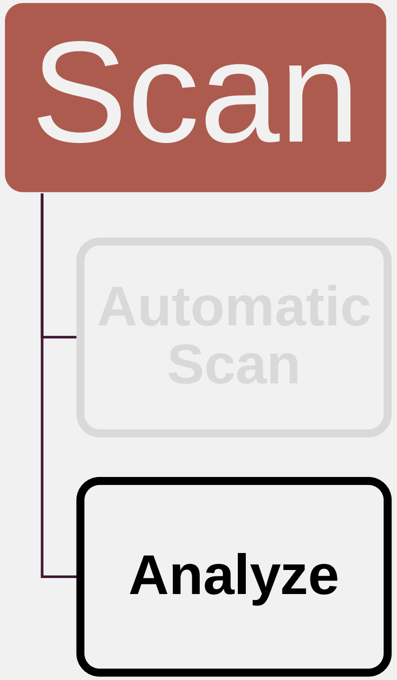


Scan

Automatic Scan

Analyze

- ○ A vulnerability scan to be launched automatically by CloudInit script.
- ○ The deeper the scan, the longer it takes to move to production.

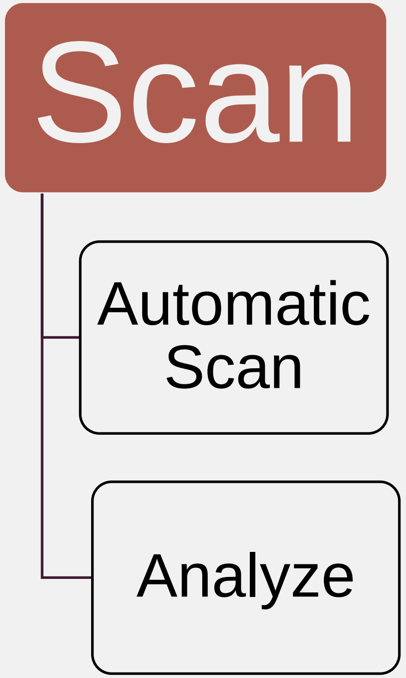


Scan

Automatic Scan

Analyze

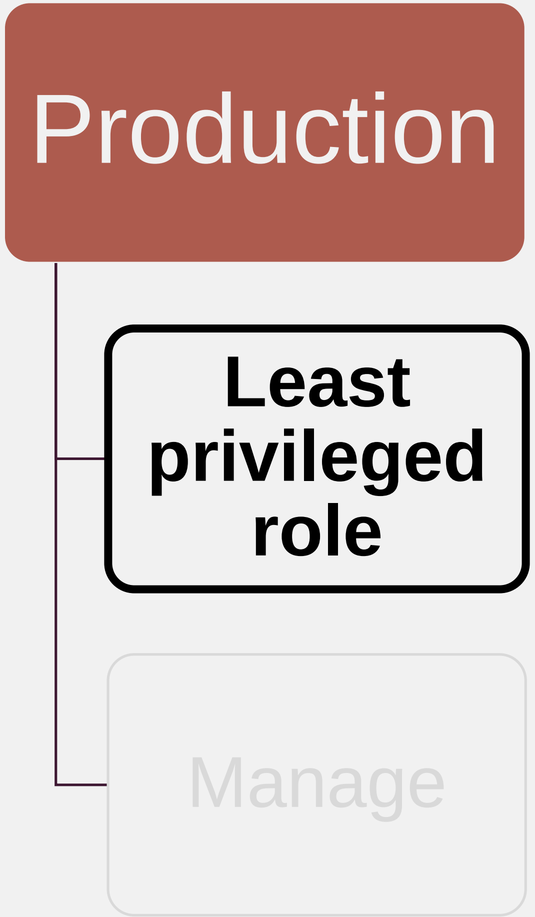
- ○ The results of the scan are analyzed by the Cloudefigo script.
- ○ Based on scan results – the instance to move to production or remain in the remediation group.
- ○ The lowest security risk severity can be defined.



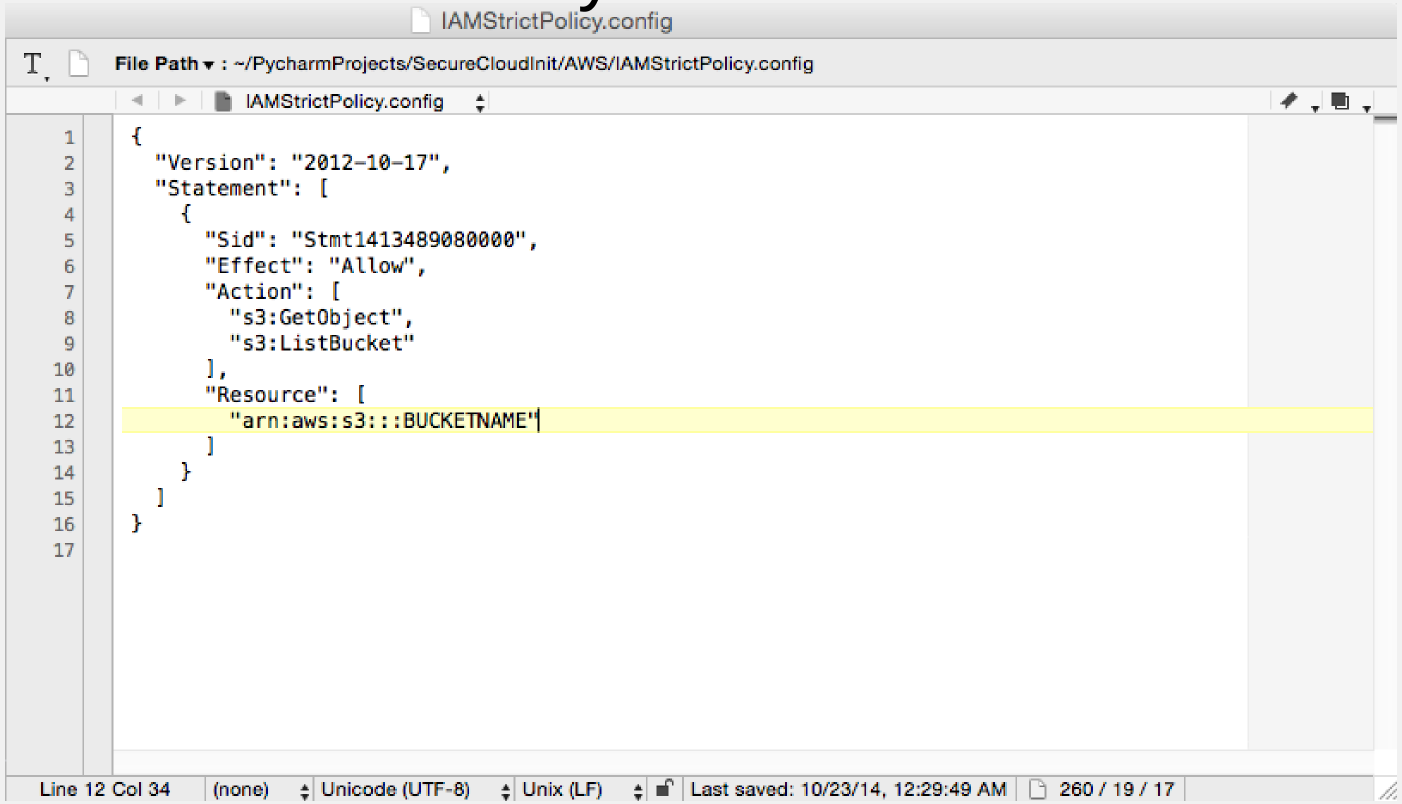
Scan

Automatic Scan

Analyze



- Reminder: Permissions in launch > production
- IAM role permissions reduced dynamically - contains read only access



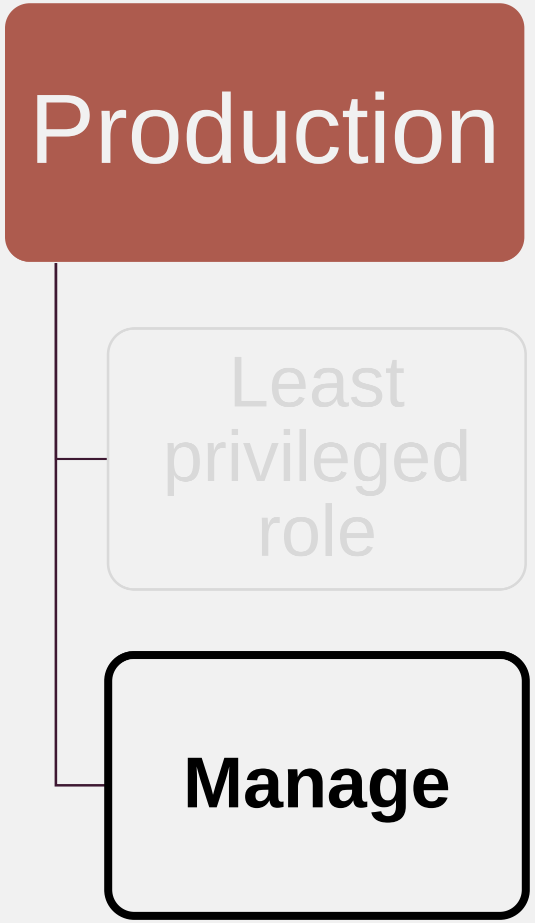
IAMStrictPolicy.config

File Path ▾ : ~/PycharmProjects/SecureCloudInit/AWS/IAMStrictPolicy.config

IAMStrictPolicy.config

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1413489080000",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::BUCKETNAME"
      ]
    }
  ]
}
```

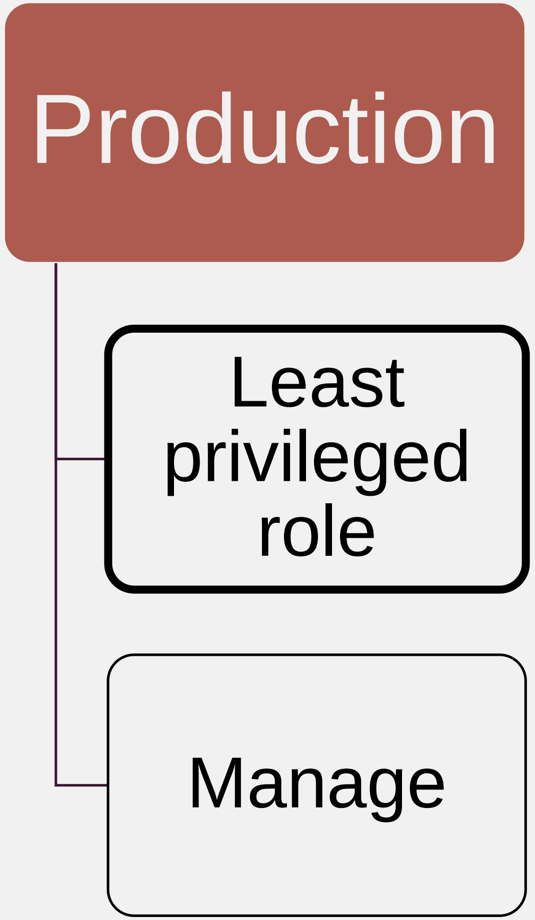
Line 12 Col 34 | (none) | Unicode (UTF-8) | Unix (LF) | Last saved: 10/23/14, 12:29:49 AM | 260 / 19 / 17


Production

Least privileged role

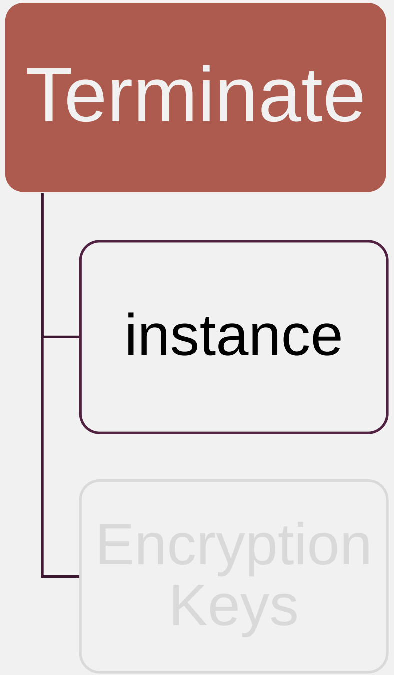
Manage

- For the ongoing operations – a compensating control is required to locate unmanaged instances.
- Cloudefigo management script lists cloud instances and validates they are managed by Chef.
- Unmanaged instance can move to remediation, forensics (not implemented in the current version)



Production

- Least privileged role
- Manage

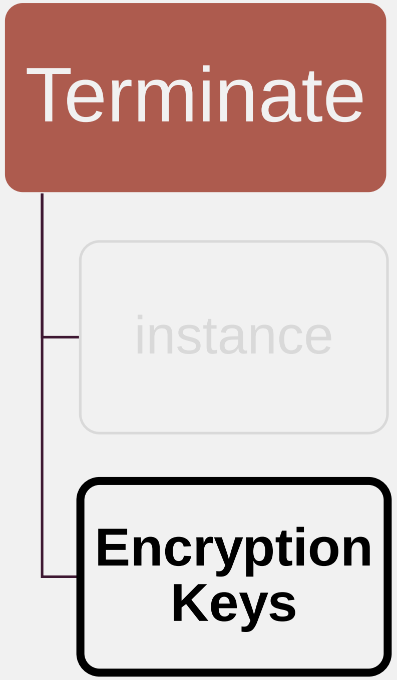


Terminate

instance

Encryption Keys

- The life cycle ends once a server is terminated along with:
  - Attached volumes
  - IAM role



Terminate

instance

Encryption Keys

- The instance data still exist in backups/snapshots or provider storage.
- Encryption keys to be deleted with instance in order to make sure the backup data remain inaccessible (not implemented in this version)

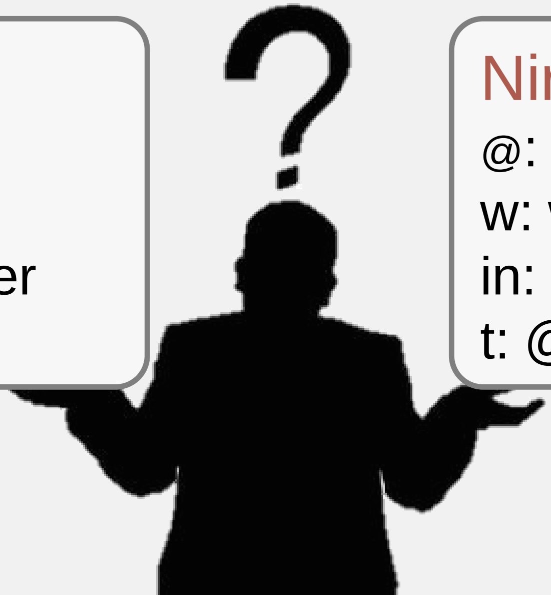
Wrapping up

- Instance launched using Cloudefigo management script
- Installed security configurations.
- Encrypted the volume.
- Scanned for vulnerabilities.
- Moved to production.

And all of it is automatic and secured

Questions

Moshe Ferber
@: moshe@onlinecloudsec.com
w: www.onlinecloudsec.com
in: www.linkedin.com/in/MosheFerber



Nir Valtman
@: nir.valtman@ncr.com
w: www.ncr.com | www.valtman.org
in: www.linkedin.com/in/valtmanir
t: @ValtmaNir