



# COMMUNITY DAY

TEL AVIV



# Access Control in a Multi-Account Environment

Shimi Rokah | 2018



The following presentation contains confidential and proprietary information or data belonging exclusively to NICE LTD. The use of this information is provided under the specific undertaking that the information and data will be kept in confidence. The information cannot be provided to any third party. NICE LTD has and will file one or more patent applications covering the subject matter provided herein. The information and data are also copyrighted and cannot be copied for any purpose without the written approval of NICE, which approval has been provided prior to any such copying. Some or the entire document may contain trade secrets of NICE and cannot be provided to any other person. By viewing this presentation you agree to the above terms and conditions.

**NICE®**



# COMMUNITY DAY

## About Me

- Ex-8200
- Tech Enthusiast
- Expert Cloud Architect
- 9 Years at NICE

Currently working on NICE's cloud platform and applications

[shimi4@gmail.com](mailto:shimi4@gmail.com)

[www.linkedin.com/in/shimi-rokah](https://www.linkedin.com/in/shimi-rokah)





# COMMUNITY DAY

NASDAQ:  
**NICE**

**>25,000**  
Customers

**>85%**  
Fortune  
100 Customers

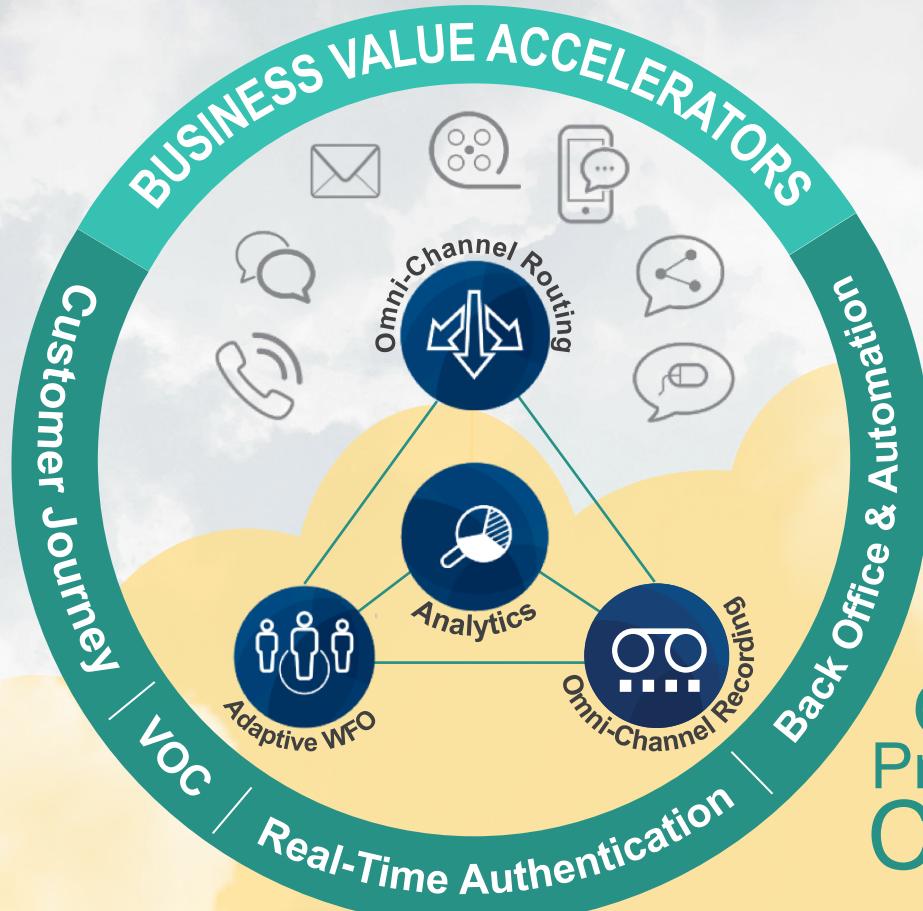
**>150**  
Countries

**~6,100**  
Professionals

**~1,800**  
R&D  
Professionals

**>35**  
Local Offices

# Open Cloud Platform



# On Premises/Private Cloud

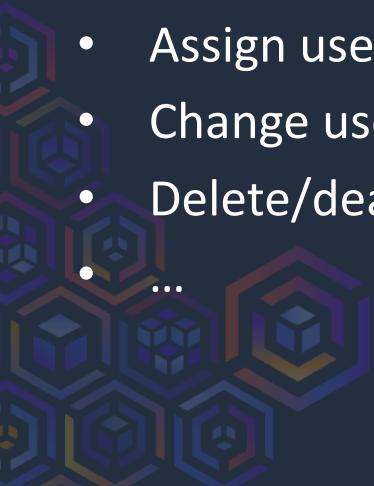




# COMMUNITY DAY

## IAM-Based Access Control

- Define roles
- Create users (could be hundreds or thousands)
- Assign users to roles
- Change user-role assignments
- Delete/deactivate users when they leave (product/company)
- ...





## COMMUNITY DAY

Now think about multiplying this work by the number of accounts you own...





# COMMUNITY DAY

@NICE we manage:

- Dozens of AWS accounts
  - Development, testing, performance, staging, production...
- About a dozen roles
  - Developer, Architect, DevOps, Ops, Admins, GLs...
- Hundreds of users





# COMMUNITY DAY



## Problem Analysis

- What causes most of the overhead – user creation and updates
- What cannot be avoided – maintaining roles per account
  - On each account a persona (e.g. architect) may need different permissions
- Additional concerns
  - Inconsistency of user-role associations on different accounts
  - Leaving employee – chances of an IAM user still being active in some accounts





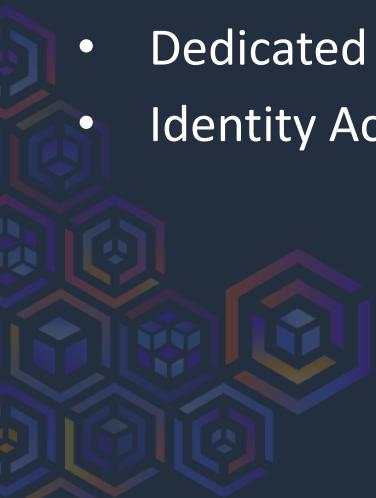
# COMMUNITY DAY



## The Solution – Centralized User Management

Possible implementations:

- Organizational IdP
- Dedicated IdP
- Identity Account





# COMMUNITY DAY



## The Solution – Centralized User Management

Possible implementations:

- Organizational IdP – pulls organizational IdP into R&D & prod environments and compliance audits.





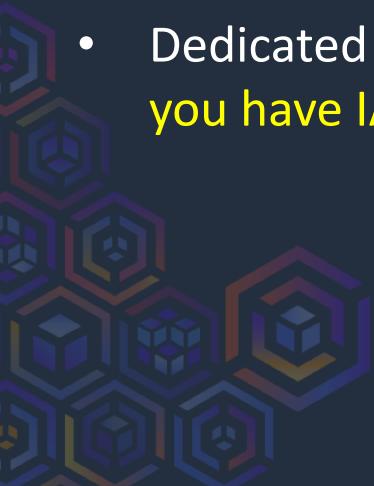
# COMMUNITY DAY



## The Solution – Centralized User Management

Possible implementations:

- Dedicated IdP – reasonable solution. But why manage a dedicated IdP when you have IAM as a managed service?





# COMMUNITY DAY

## The Solution – Centralized User Management

Possible implementations:

- Identity Account – Use IAM to define users once and allow them to assume roles defined in your accounts.





# COMMUNITY DAY



## How it works

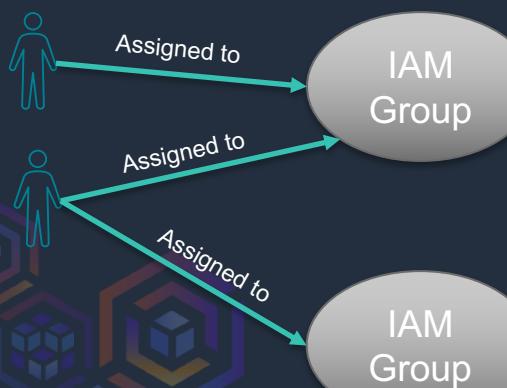
- A dedicated “Identity Account”
- Users are created once in the Identity Account
- Users are assigned to groups that represent their org. role
  - Developer, DevOps, Architect, etc.
- Groups have policies that allow role switching to other accounts’ roles
- Each account has a list of roles to which users switch
  - Roles names are identical in all accounts



# COMMUNITY DAY

## How it works

Identity Account



Associated with  
Policy



Policy  
...



Allow user to assume role  
Developer

Allow user to assume role  
Admin

Allow user to assume role  
Developer

Allow user to assume role  
Admin

Account #1

Account #2



Admin

Admin



# COMMUNITY DAY

## Group Policy Example

```
{  
  "version": "2015-11-17",  
  "Statement": [  
    {  
      "Action": [  
        "sts:AssumeRole"  
      ],  
      "Resource": [  
        "arn:aws:iam::123456789:role/MyAdminsRole",  
        "arn:aws:iam::987654321:role/MyAdminsRole",  
      ],  
      "Effect": "Allow"  
    }  
  ]  
}
```





# COMMUNITY DAY

## CI account “Developer” role

```
{  
  "version": "2012-10-17",  
  "Statement": [  
    {  
      "Action": ["cloudwatch:*", "logs:*"],  
      "Effect": "Allow",  
      "Resource": "*"  
    },  
    {  
      "Action": ["s3:GetObject"],  
      "Effect": "Allow",  
      "Resource": ["*"]  
    }]  
}
```

## Production account “Developer” role

```
{  
  "version": "2012-10-17",  
  "Statement": [  
    {  
      "Action": ["cloudwatch:*", "logs:*"],  
      "Effect": "Allow",  
      "Resource": "*"  
    }]  
}
```



# COMMUNITY DAY

## Role MFA Enforcement

Type: 'AWS::IAM::Role'

Properties:

AssumeRolePolicyDocument:

version: 2018-11-15

Statement:

- Effect: Allow

- Principal: \*\*\*\*

- Action:

- 'sts:AssumeRole'

- Condition:

- Bool:

- aws:MultiFactorAuthPresent: true

- ManagedPolicyArns: \*\*\*\*

- ...





# COMMUNITY DAY



## Benefits

- Users are managed in a single place
- Inconsistency of user role assignment in different accounts is not likely
  - Only if the configuration for a whole group is incorrect
- Leaving employees need only to be deleted in a single place
- Access Control is managed with CloudFormation - automated





# COMMUNITY DAY



## Benefits

- Implementation is based solely on AWS managed services
  - No 3<sup>rd</sup>-party deployment/management required
- Flexible MFA enforcement
  - Only for relevant roles, only on relevant accounts
  - Users activate MFA on their own





# COMMUNITY DAY

TEL AVIV

# Thank You



We Have  
Some Cool  
Opportunities  
**JOIN US!**



<https://www.nice.com/careers>

