amazon web services

# Organizations Deep Dive

Tim Robinson

# Agenda

- Summary of Organizations

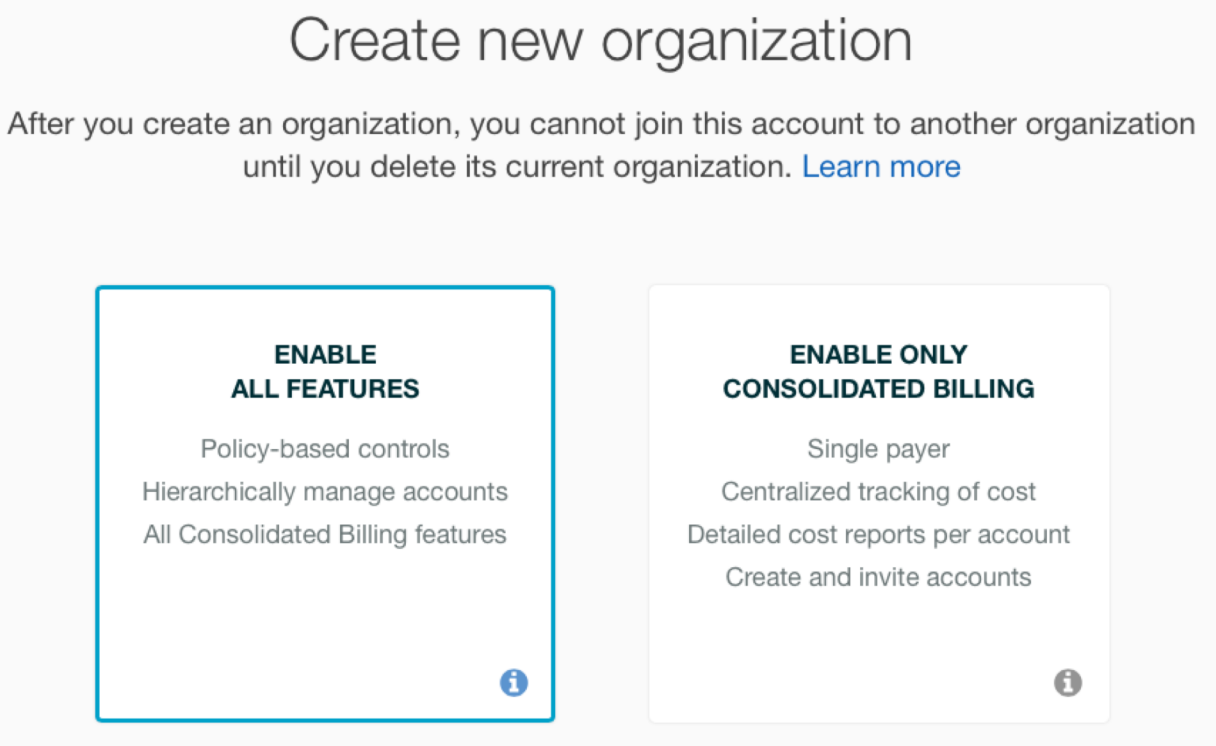
- How to create an Organization
- How existing accounts can join or leave an Organization
- How to create new accounts

- How to create Organization wide policies
- What these policies can, and can’t do
- Some example policies

# Organizations Summary

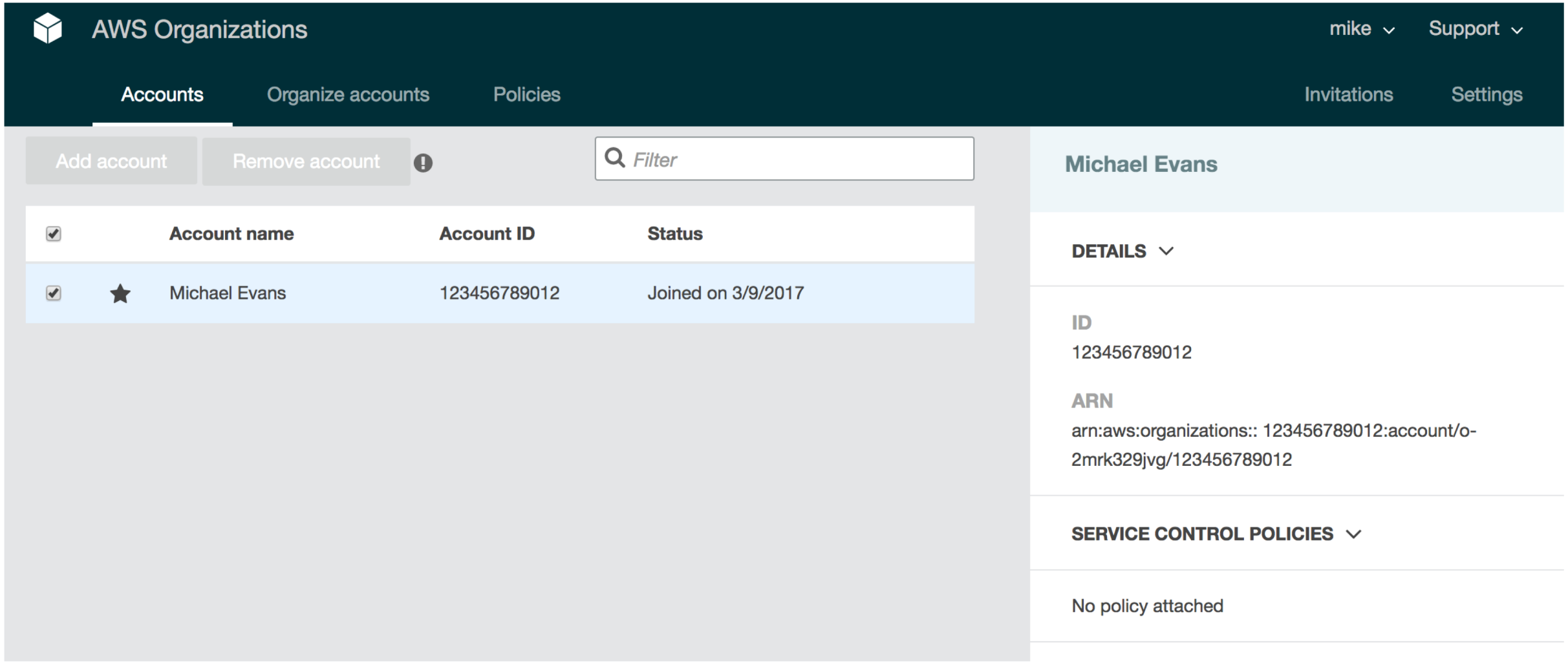
- A way to keep track of all your AWS accounts
- Organize accounts in hierarchies
- Setup consolidated billing
- Create policies that apply to your AWS accounts

# Creating an Organization




- Login to your master account, click on your user name and select “My Organization”
- Click on ”Create organization”
- Choose between consolidated billing only or full

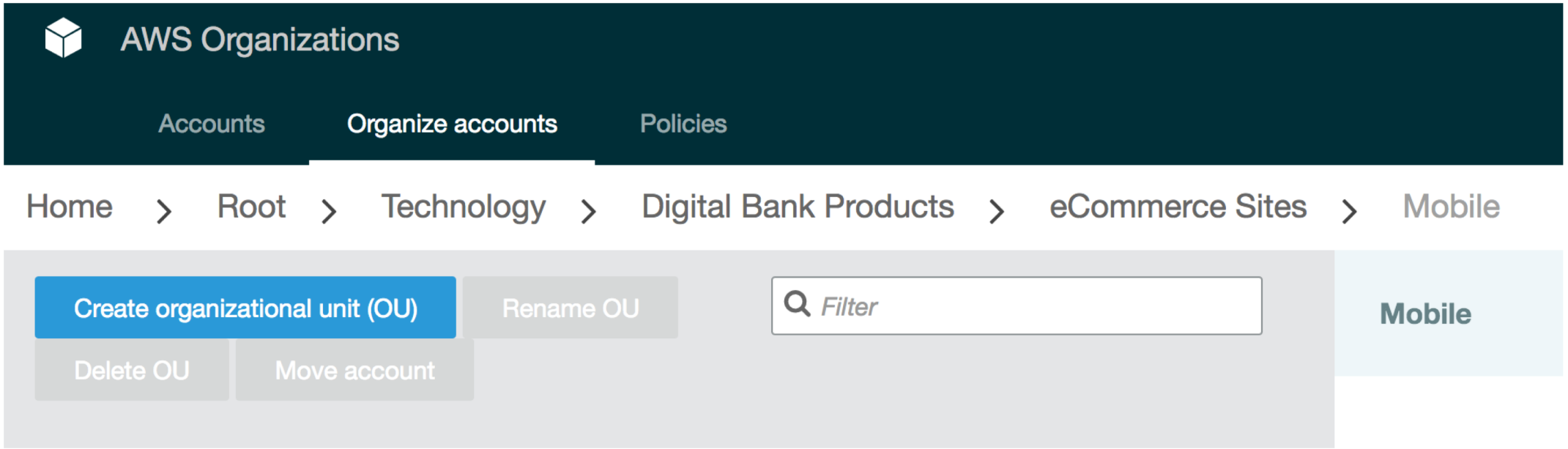
# Creating an Organization



# Master Account Security

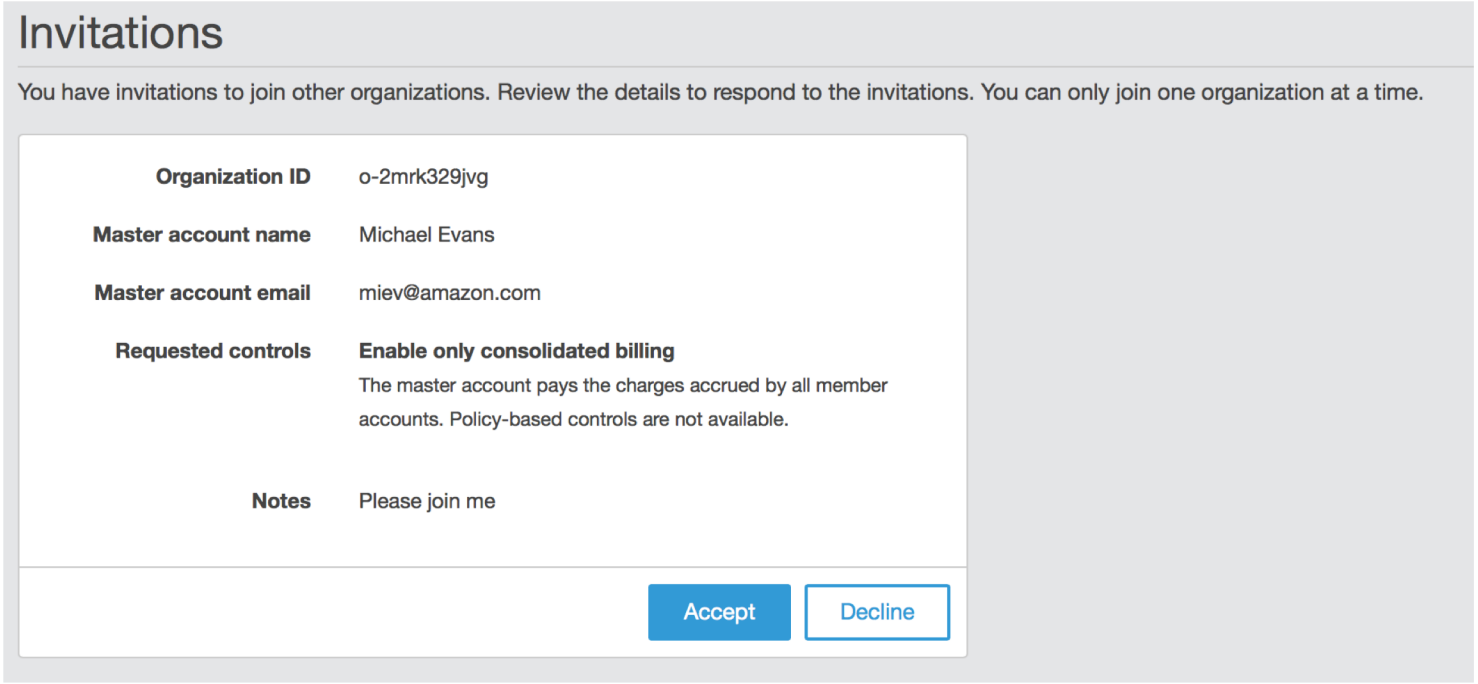
- Master account compromise can have a huge blast radius
- Configure MFA on users in the master account
- Create restricted policies and cross-account roles for the following tasks:
  - Billing reporting
  - Organizations management, creating OUs, moving accounts between OUs
  - Organizations policy management, create policy, attach policy, detach policy
- Try to treat the entire master account like a root account - lock it down, no single person has access, raise alerts if there are logons, do not host any services from the master account.

# Example OUs



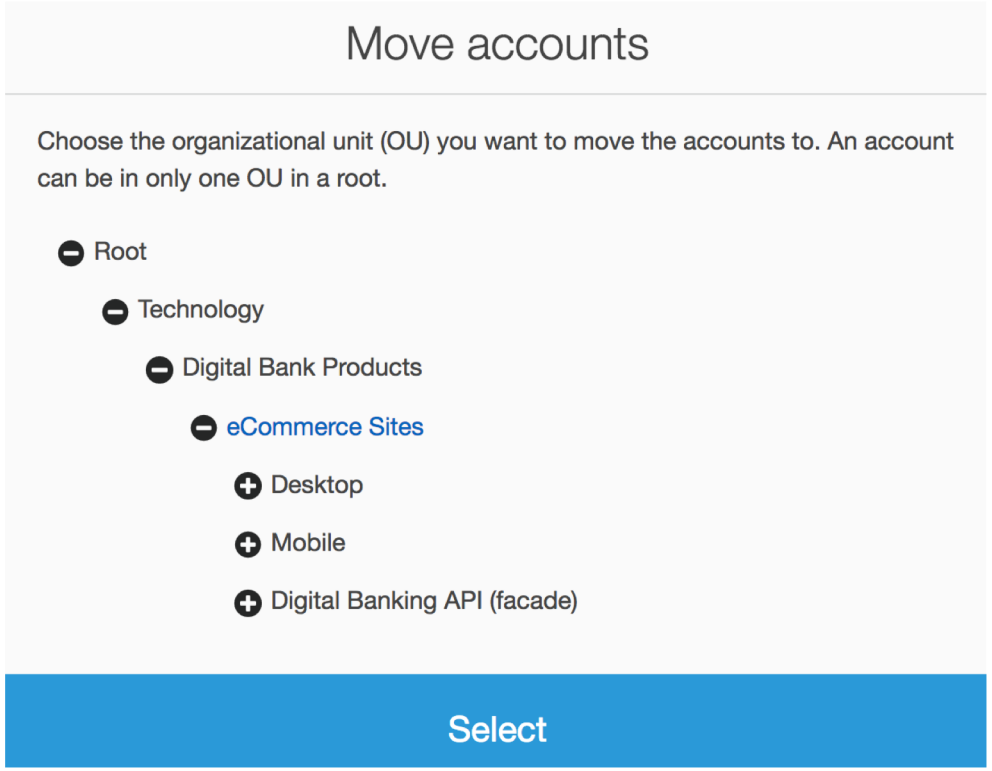

- Create a hierarchy that works for your business
- Match internal business structure and technology stack
- Create lots of accounts to isolate components and reduce blast radius

# Joining an Organization



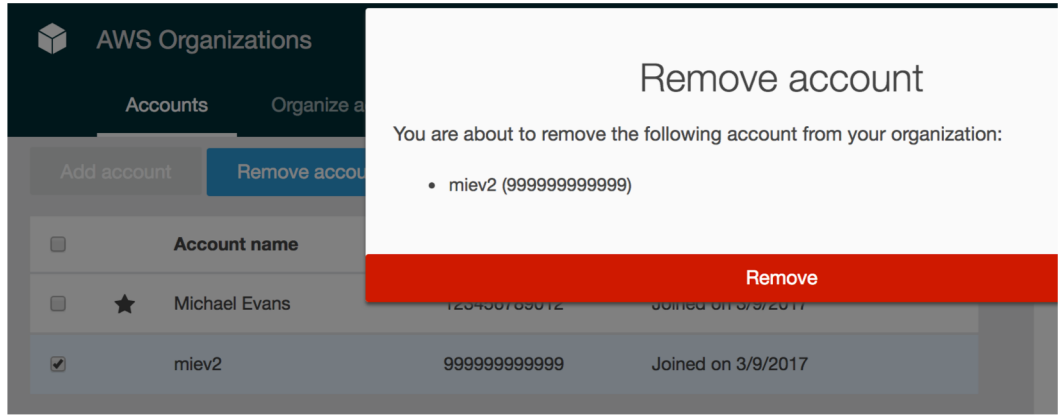

1. Master account invites the sub-account
2. Invite many accounts at the same time, and include a message
3. You can track invitations and see how they are progressing
4. Invitee clicks on a link, and accepts invitation

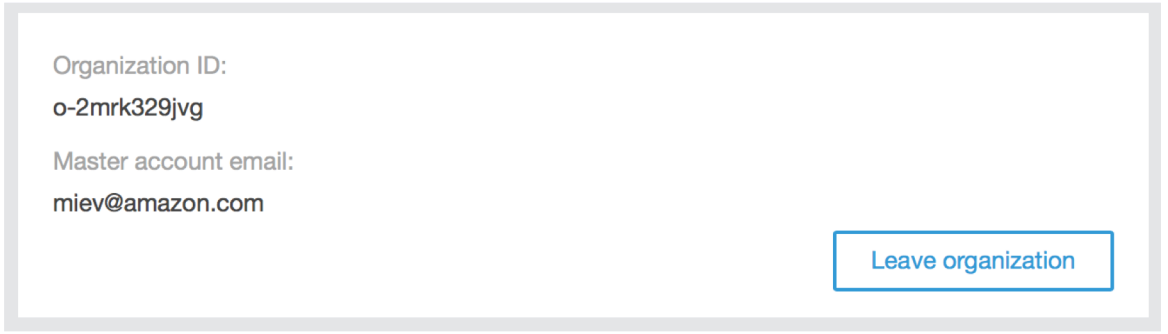
# Joining an Organization




1. New accounts will appear at the root of your Organization
2. Move accounts after they have joined to the right OU

# Leaving an Organization

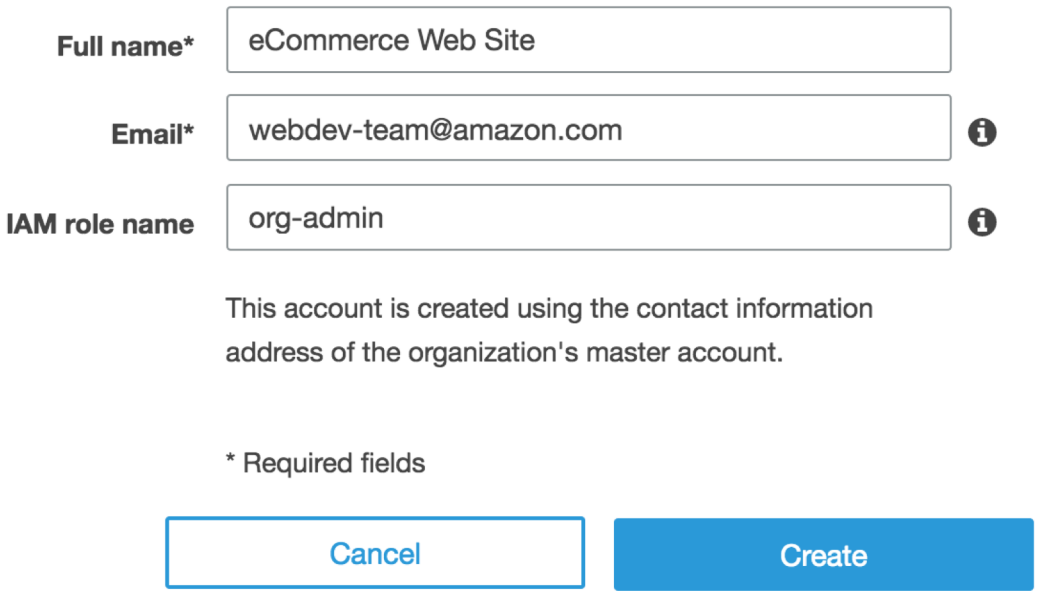



1. Master account – login to Console, select account and click Remove

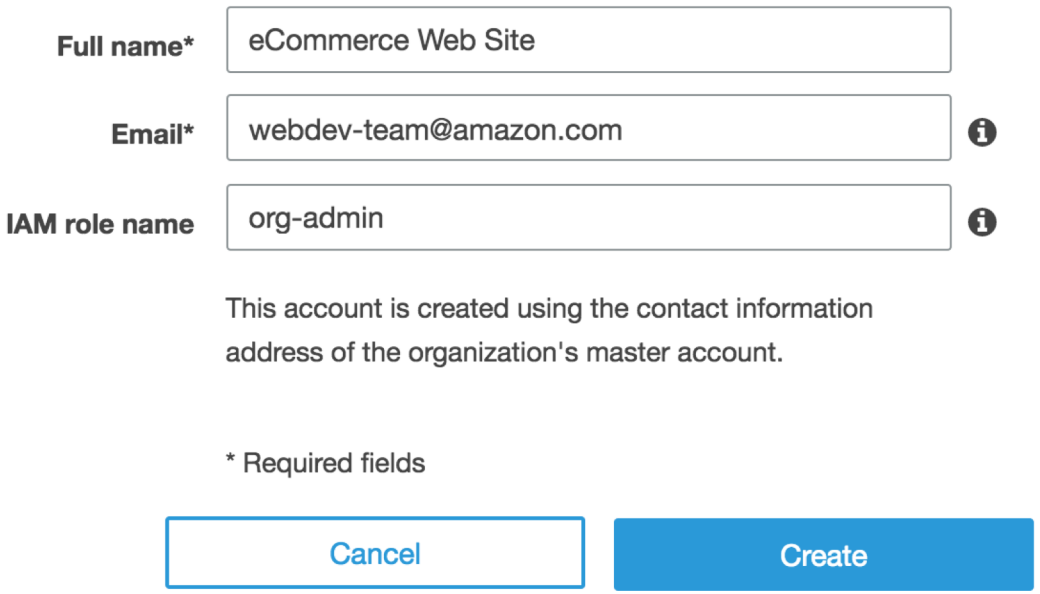
2. Sub-account – login to Console, click your name, select “My Organization”, click on “Leave organization”

# Creating new accounts




- The IAM role allows the master account to switch to an admin role inside the new account.
- If you leave the IAM role blank then a role is still created called “OrganizationAccountAccessRole”.
- See: “Switching to a role”

# More on “created” accounts




- You cannot remove “created” accounts from your Organization
- This means you won’t be able to delete your Organization
- You can’t move a “created” account from one Organization to another but you can with “invited” accounts

# Automation – Create New Account

```
[ec2-user ~]$ aws organizations create-account --email miev@amazon.com --
account-name "Mobile Web" --role-name "org-admins"
{
    "CreateAccountStatus": {
        "RequestedTimestamp": 1489052895.231,
        "State": "IN_PROGRESS",
        "Id": "car-842ceec004ad11e7acb7500c66cd64c5",
        "AccountName": "Mobile Web"
    }
}
```

# Use AssumeRole on the new account

```
[ec2-user ~]$ aws sts assume-role --role-arn
arn:aws:iam::999999999999:role/org-admins --role-session-name "account-
bootstrap"
{
    "AssumedRoleUser": {
        "AssumedRoleId": "AROAJLV5KFI2Q4I2RNTZO:account-bootstrap",
        "Arn": "arn:aws:sts::999999999999:assumed-role/org-admins/account-
bootstrap"
    },
    "Credentials": {
        "SecretAccessKey": "<removed>",
        "SessionToken": "<removed>",
        "AccessKeyId": "ASIAJVVKXYXXBHROX4AQ"
    }
}
```

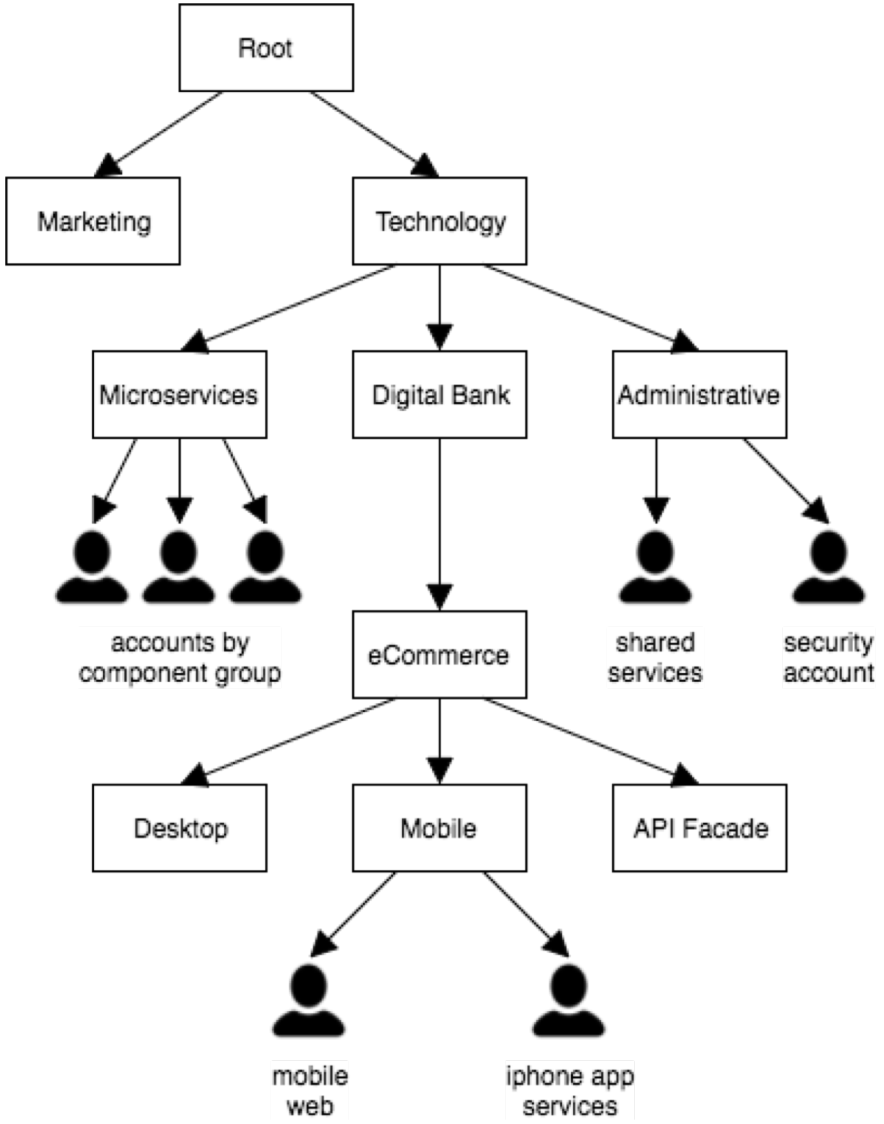
# Use CloudFormation To Setup Account

```
[ec2-user ~]$ export AWS_ACCESS_KEY_ID=ASIAJVVKXYXXBHROX4AQ
[ec2-user ~]$ export AWS_SECRET_ACCESS_KEY='<removed>'
[ec2-user ~]$ export AWS_SESSION_TOKEN='<removed>'
```

```
[ec2-user ~]$ aws cloudformation create-stack --stack-name "security-
config" --template-url 'https://s3-ap-southeast-1.amazonaws.com/security-
account-bucket/org-security-config.yaml'
```

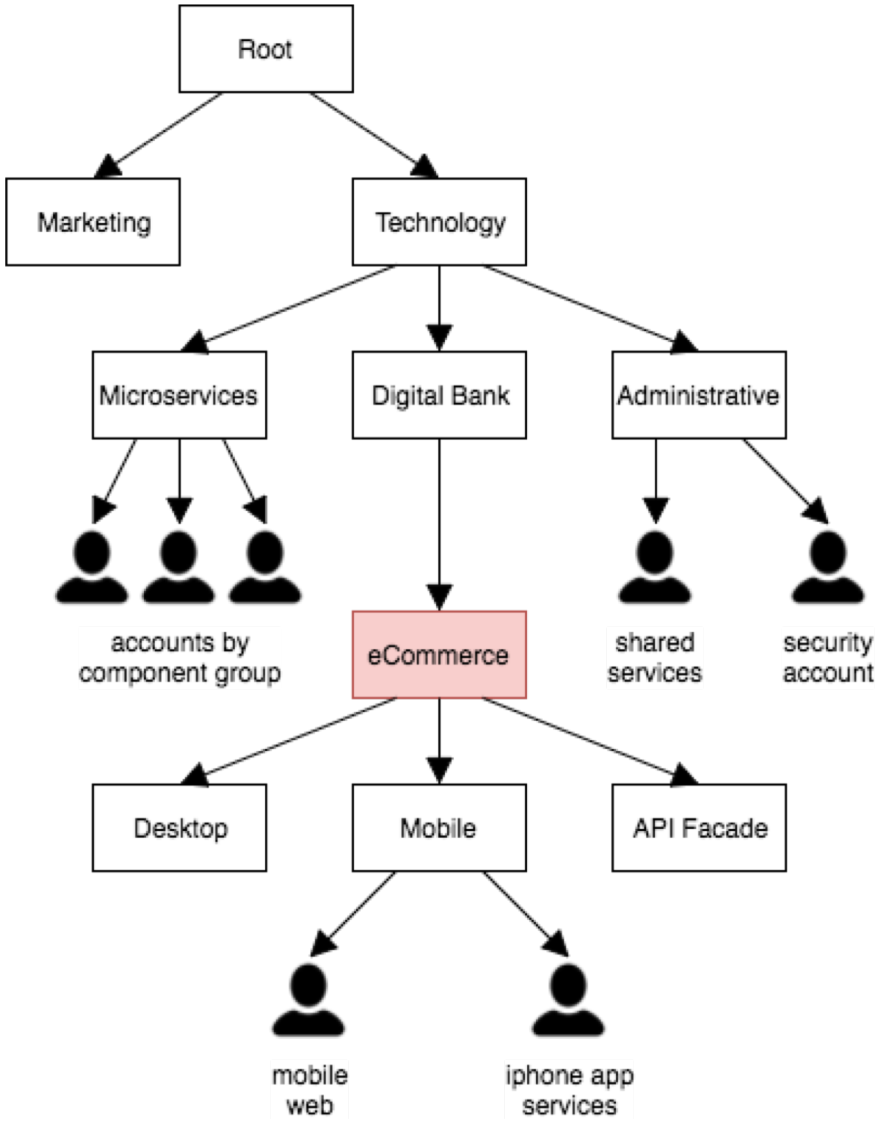
# Service Control Policies

# Policy Overview



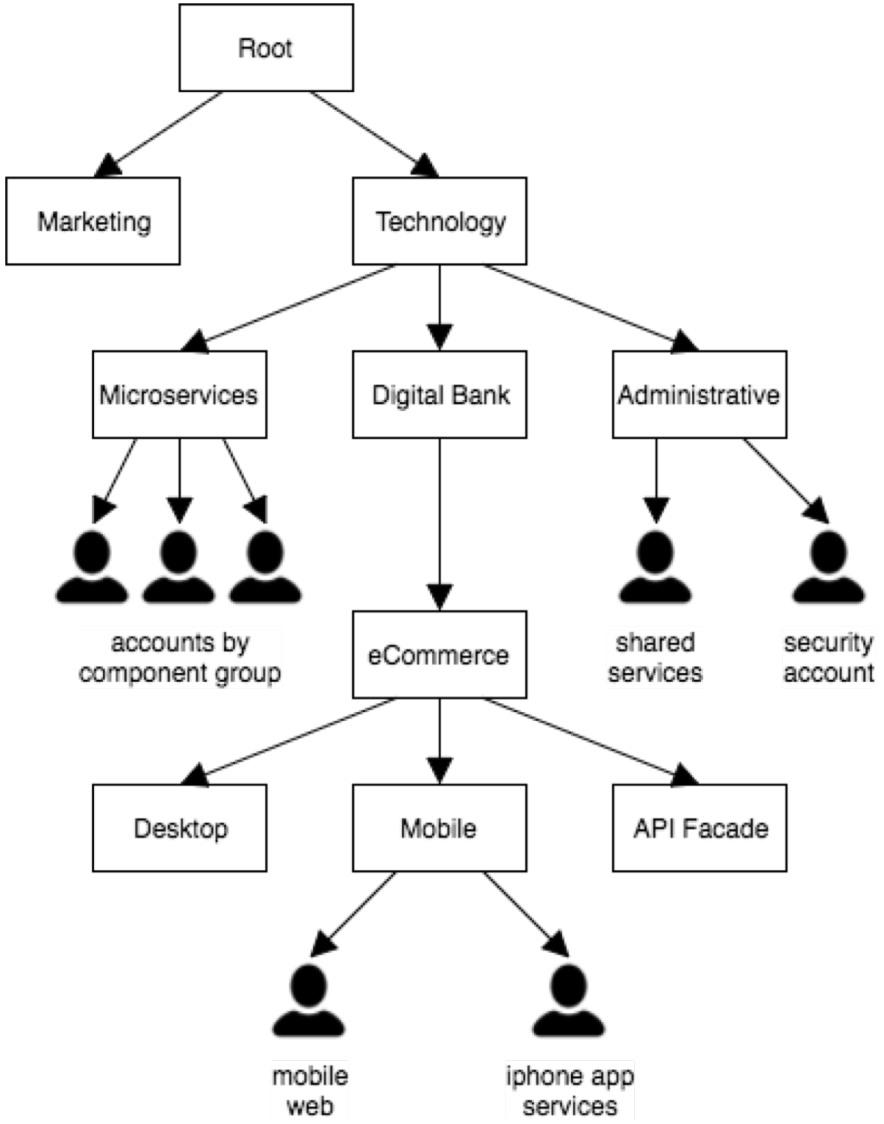

- Policies are checked from the Organizational root, through each OU and ultimately to the relevant account.
- All nodes must allow access. There is no implicit allow. If any node denies access then access will be denied.
- IAM policies inside the account are still checked.
- Policies apply to Administrators and the root users in sub-accounts.

# Policy Overview



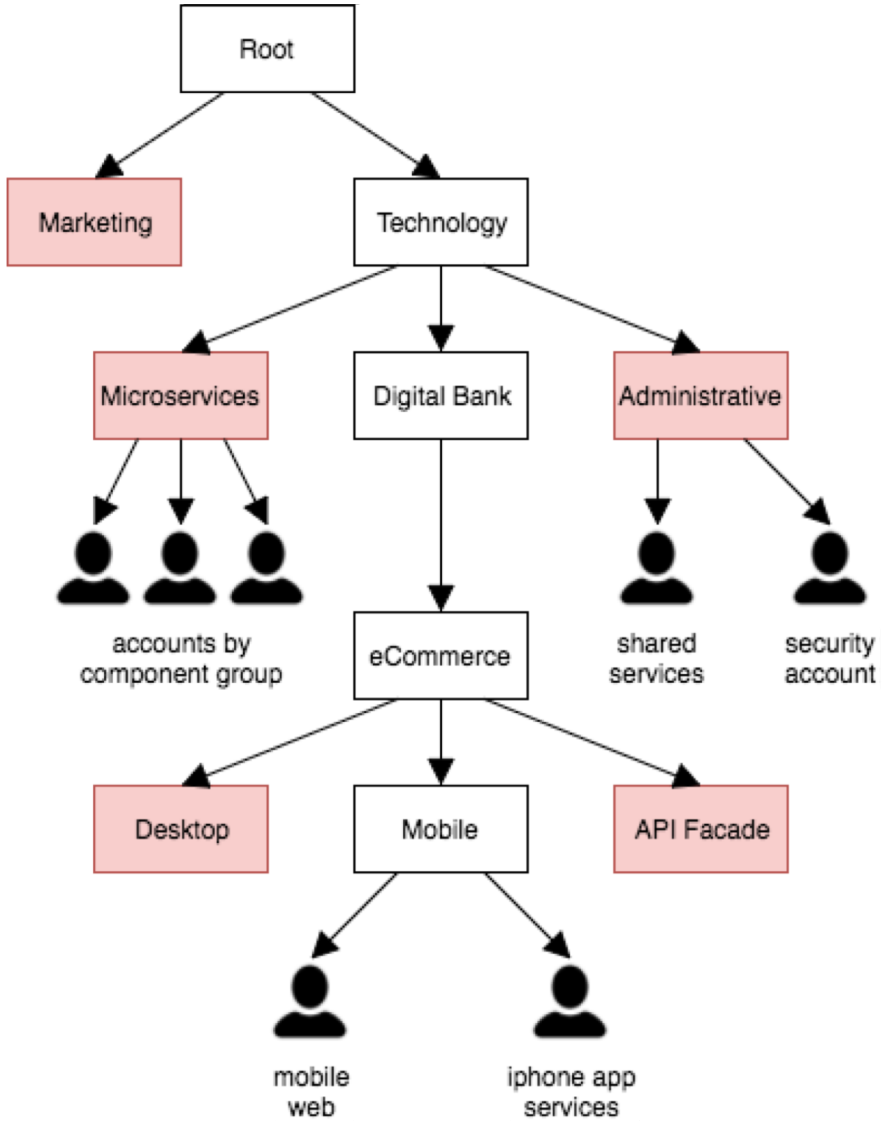

- By default a permissive AWSFullAccess policy is attached to all OUs and accounts.
- **What if: we remove AWSFullAccess from eCommerce?**

# Policy Overview



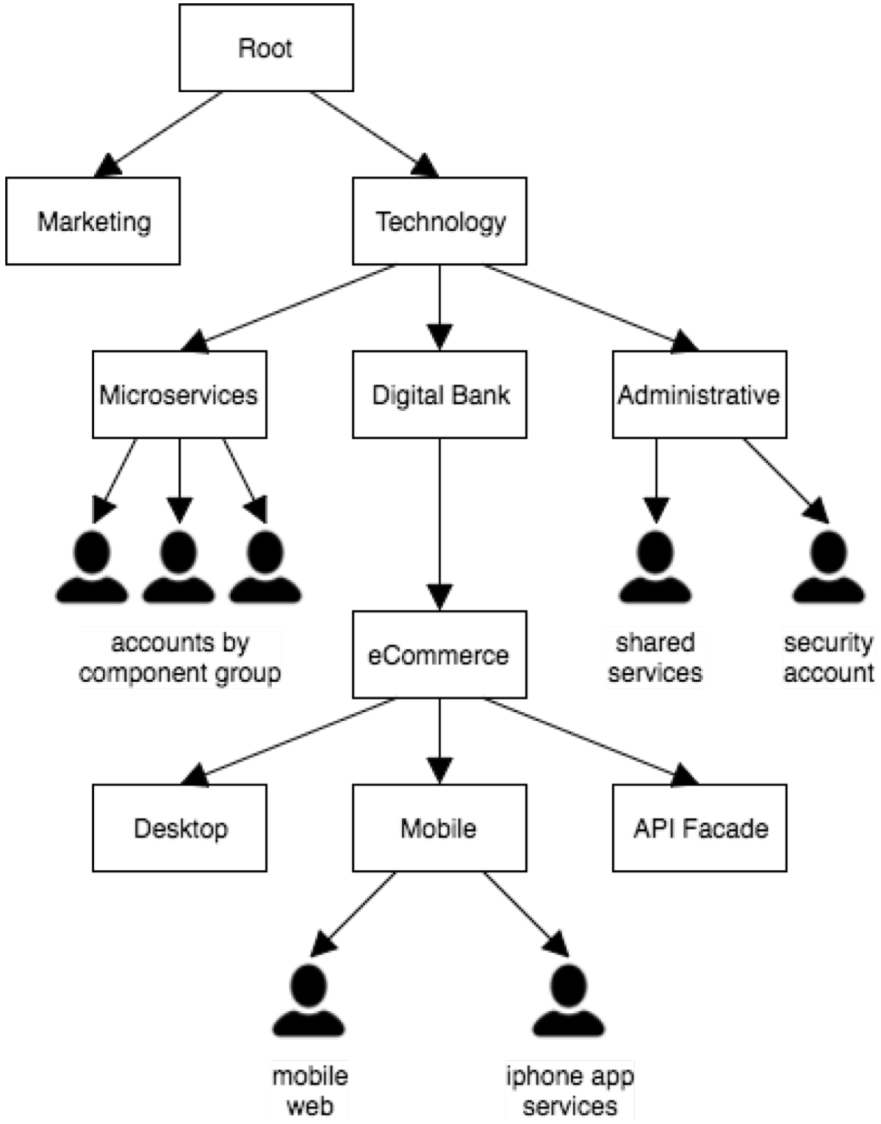

- Could we deny Device Farm from all users, except the “iphone app services” account?
- Any ideas?

# Policy Overview



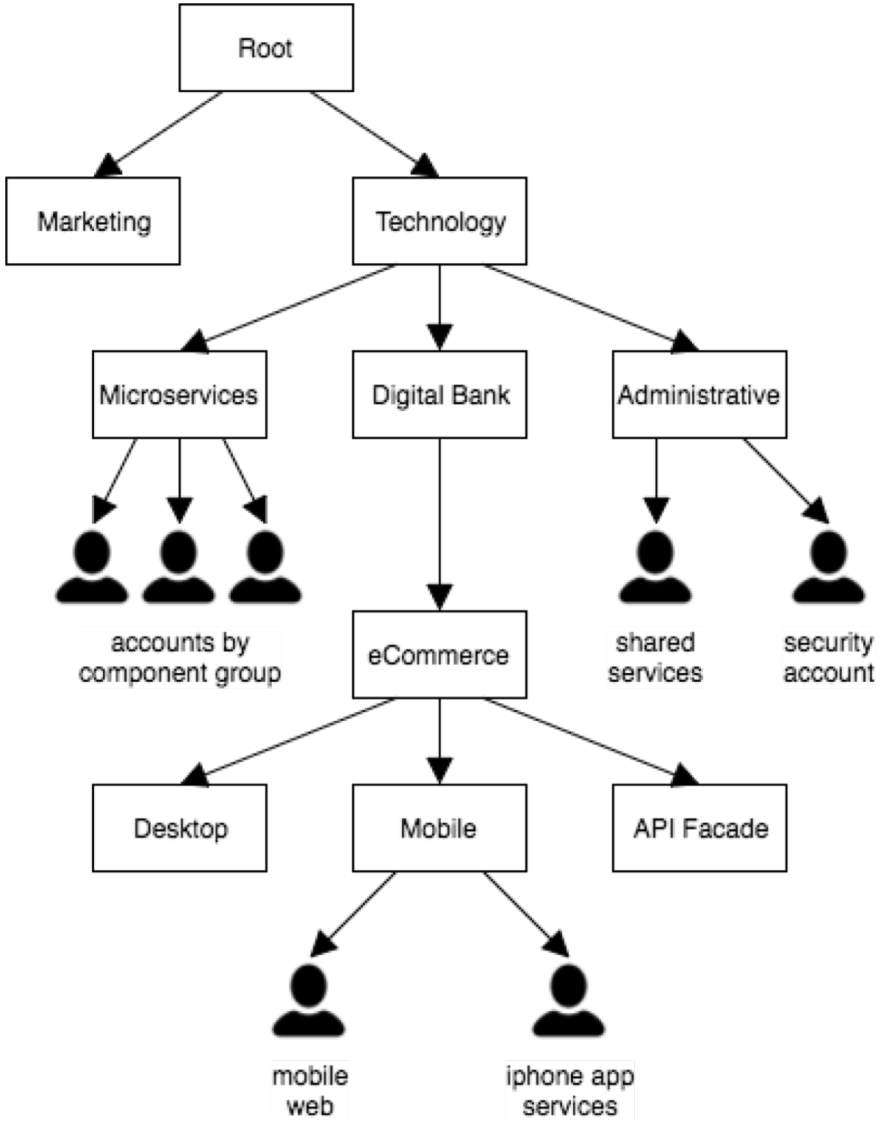

- Could we deny Device Farm from all users, except the “iphone app services” account?
- Not easily!  Create a deny policy for devicefarm:* and apply to everywhere that doesn’t need Device Farm.
- Think about how you want to use policies and make sure the OU hierarchy supports those goals.

# Policy Tips



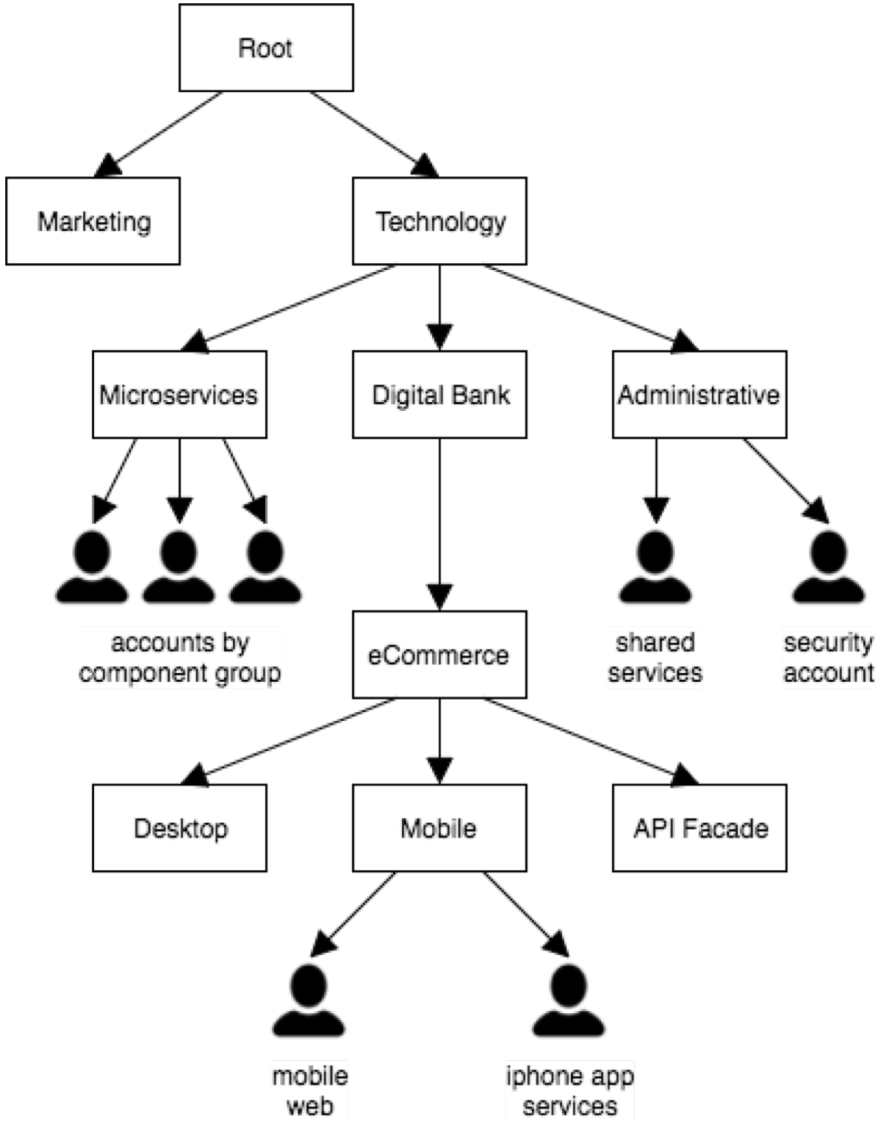

- Use AWSFullAccess above and below control points where you want policy decisions to flow down.
- Whitelist – remove AWSFullAccess and then list out the services approved for use within an OU and its children.
- Blacklist - create a list of services not allowed and deny them on the relevant OU.
- Use policy simulator in member accounts to test the effect on users of local policies and Organizational policies.

# Example Policies



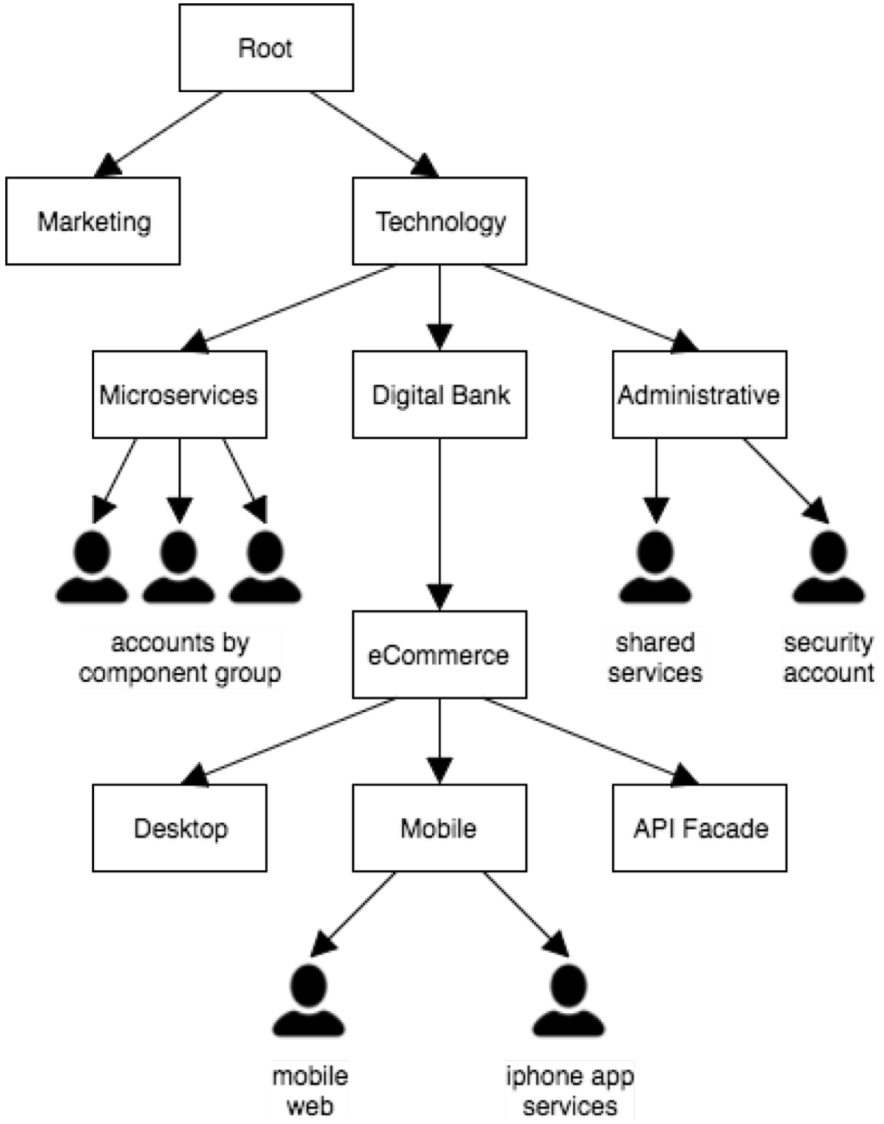

- Deny organisations:* on root.
  - Doesn't break the management of the organization as policies don't apply to the master account
  - Policies don't apply to the master account even if you move it into the Administrative OU
  - This prevents sub-accounts from knowing anything about the Organization and prevents them leaving.
- Deny attach ISG, NAT-GW on non-production OUs and internal systems.  Two layers of protection.
- With two layers of protection, it makes it easier and less risky to take a more permissive approach inside individual accounts.

# Example Policies




- Prevent accounts from removing security compliance checks / automation.
- Deny config:deleteConfigRule
- Move accounts to a more permissive OU to make changes and then move back.

# Example Policies




- Restrict OUs to only the right AWS certified services.
- Whitelist only PCI certified or HIPAA eligible services.
- Force a walled garden around sensitive accounts while leaving other out-of-scope accounts more open.

# AWS Organizations Limits

- Total number of accounts in a single Organization: 20
- Maximum invites that can be sent per day: 20
- Maximum depth of an OU hierarchy: 5

# Management Tools

- CloudFormation StackSets
- “pip install aws-with”

# Thank you.. questions?

teratim@amazon.com

**Dive even deeper?**

| | |
|---|---|
| AWS Organizations docs: | https://docs.aws.amazon.com/organizations/latest/userguide |
| AWS Organizations FAQ: | https://aws.amazon.com/organizations/faqs/ |
| Service Control Policies: | http://docs.aws.amazon.com/organizations/latest/userguide/orgs_manage_policies_about-scps.html |
| CLI reference: | https://docs.aws.amazon.com/cli/latest/reference/organizations/index.html |
| API reference: | http://docs.aws.amazon.com/organizations/latest/APIReference/Welcome.html |
| Forums: | https://forums.aws.amazon.com/forum.jspa?forumID=219 |