STEP-BY-STEP GUIDE: AWS IAM & EC2 SECURITY WALKTHROUGH

A comprehensive walkthrough for setting up AWS EC2 instances with proper IAM security controls using tag-based access management and least privilege principles.

1.LAUNCH AND TAG EC2 INSTANCES

LOG IN TO AWS CONSOLE

Go to **https://console.aws.amafion.com** and select your region.

OPEN EC2 DASHBOARD

Type "EC2" in the service search bar and click EC2 under "Compute".

CLICK "LAUNCH INSTANCES"

Find the Launch Instances button on the dashboard.

NAMING AND TAGGING 3CREATE TWO INSTANCES:

FIRST (PRODUCTION):

- In "Name and tags", add Name: Godfrey-prod.
- Add tag: Key: Env, Value: production.

SECOND (DEVELOPMENT):

- Add Name: Godfrey-dev.
- Add tag: Key: Env, Value: development.

ADDITIONAL CONFIGURATION STEPS:

- Choose OS and Instance Type: Select Amazon Linux 2023 (Free tier eligible). Choose instance type: t3.micro or t2.micro.
- Configure Network Settings: Leave default VPC/subnet (unless your org requires changes).
- Key Pair: Choose an existing key pair or create a new one if you want SSH
- access. Review and Launch: Confirm configuration, then click "Launch Instance".
- Repeat: Repeat to launch both prod and dev instances with correct tags.

2. WRITE AND APPLY A LEAST-PRIVILEGE, TAG-SPECIFIC IAM POLICY

NAVIGATE TO IAM SERVICE

In the AWS Console, search "IAM" and select the IAM service.

GO TO POLICIES > CREATE POLICY

Click Policies in the sidebar. Click the orange Create policy button.

SWITCH TO THE JSON TAB

Click "JSON" and paste the policy.

REVIEW AND NAME

Click "Next". Name: GodfreyDevEnvironmentPolicy. Add a clear description. Click Create policy.

POLICY JSON:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "ec2:*",
 "Resource": "*",
 "Condition": {
 "StringEquals": {
 "ec2:ResourceTag/Env": "development"
 }
 }
 },
 {
 "Effect": "Allow",
 "Action": "ec2:Describe*",
 "Resource": "*"
 },
 {
 "Effect": "Deny",
 "Action": [
 "ec2:DeleteTags",
 "ec2:CreateTags"
 ],
 "Resource": "*"
 }
 ]
}
```

3. CREATE IAM GROUP, USER & ASSIGN POLICY

1

CREATE IAM GROUP

IAM > User Groups > Create group

- Click User groups in the sidebar, then "Create group".

2

CONFIGURE GROUP

- Name: Godfreydevgroup
- Attach policy: Select GodfreyDevEnvironmentPolicy
- Continue and click Create group.

3

CREATE USER

IAM > Users > Add Users

- Click Users, then Add users.

4

ADD USER DETAILS

- User name: Godfreydev
- Select "AWS Management Console
- access". Create or set a password for the user.

5

SET PERMISSIONS

- On "Set permissions", select "Add user to group".
- Choose: Godfreydevgroup

6

REVIEW AND CREATE USER

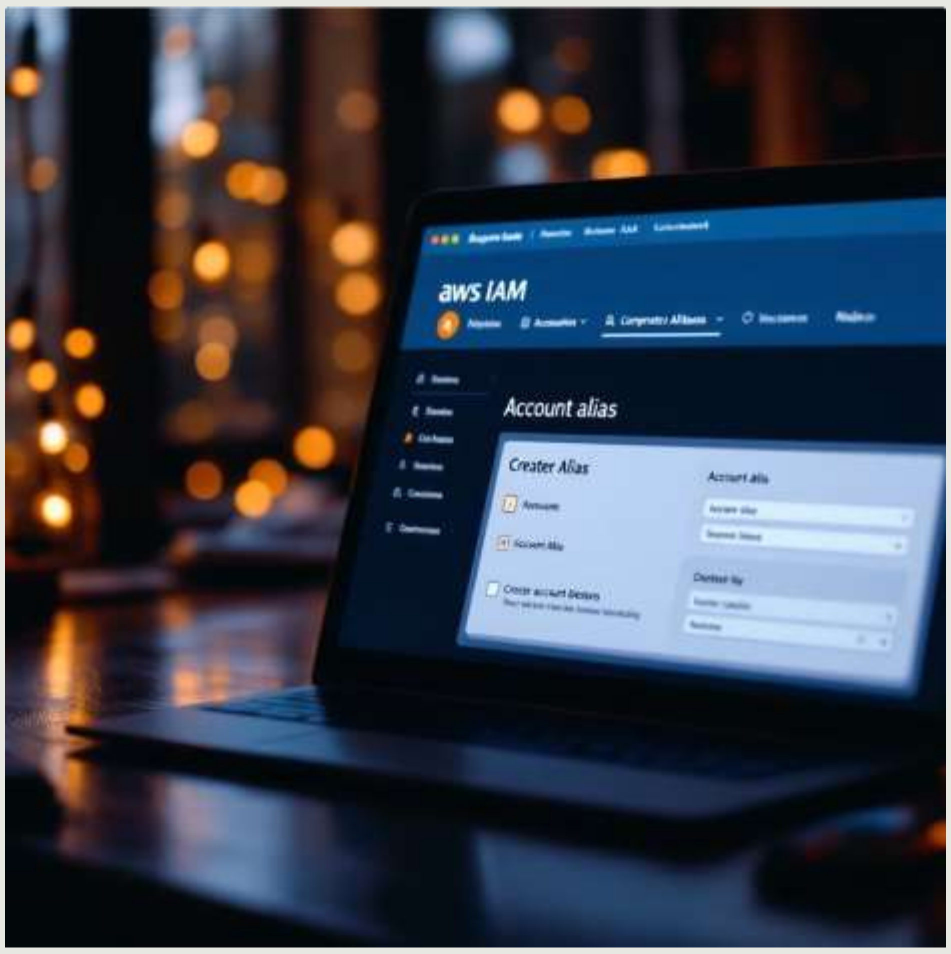
- Confirm settings, download credentials CSV.

> **Best Practice:** Always assign permissions via groups, not directly to users. This makes permission management more scalable and consistent.

4. SET ACCOUNT ALIAS & ENABLE MFA

SET ACCOUNT ALIAS

1. IAM dashboard › "Account Alias" ›
2. Click "Create Account Alias".
3. Choose: godfreyalias-1 (or similar).
4. Click Create alias.
5. Your login URL updates to: https://godfreyalias-1.signin.aws.amazon.com/console



ENABLE MFA FOR USER

IAM › Users › Click on Godfreydev

Go to the "Security credentials" tab

Under "Multi-factor authentication (MFA) device", click Assign MFA device

Choose "Virtual MFA device", scan QR code with phone app, enter two consecutive codes

✓ After completing these steps, your account will have a custom login URL and the user will be protected with multi-factor authentication, significantly improving your security posture.

5. TEST PERMISSIONS & USE IAM POLICY SIMULATOR

LOG IN AS IAM USER

- Visit the account alias URL: https://godfreyalias-1.signin.aws.amazon.com/console
- Log in as Godfreydev user with the password you set.

TRY MANAGING EC2 INSTANCES

DEVELOPMENT INSTANCE

Navigate to EC2 > Instances.
Start/stop the development instance (Env=development) 3 **should work**.

PRODUCTION INSTANCE

Try to do the same with the production instance 3 **should be blocked**.

TAG MANAGEMENT

Attempt to add/remove tags from any instance 3 **should be denied**.

VALIDATE POLICY IN POLICY SIMULATOR

1. As admin: In IAM, go to "Policy Simulator" (search for it in the IAM console).
2. Select the Godfreydev user.
3. Choose service "EC2", actions like StartInstances, StopInstances, CreateTags, DeleteTags.
4. Under "Resource tags", set Env=development or Env=production.
5. Click "Run Simulation".

EXPECTED POLICY SIMULATOR RESULTS

DEVELOPMENT RESOURCES

Allowed: ec2:* actions on Env=development

User can start, stop, reboot, and perform other operations on development instances.

PRODUCTION RESOURCES

Denied: Any action on Env=production

User cannot modify production instances in any way, enforcing environment separation.

TAG MANAGEMENT

Denied: CreateTags and DeleteTags everywhere

User cannot modify tags on any resources, preventing privilege escalation.

This validation confirms that our tag-based access control is working correctly, allowing the user to manage only development resources while preventing any modifications to production environments or tags.

TIPS FOR BEST PRACTICE

GROUP-BASED PERMISSIONS

Always assign permissions via groups, not directly to users.

DESCRIPTIVE TAGGING

Use descriptive tagging conventions (Name, Env, etc.).

DOCUMENTATION

Document your policies and group assignments clearly in the repo.

MULTI-FACTOR AUTHENTICATION

Enable MFA on all users.

REGULAR VALIDATION

Regularly use IAM Policy Simulator for validation.

SECURITY ARCHITECTURE OVERVIEW

TAG-BASED ACCESS CONTROL

Resources are controlled by their tags, allowing fine-grained permissions based on environment.

LEAST PRIVILEGE

Users only have access to exactly what they need -development resources only.

MFA PROTECTION

Multi-factor authentication adds an additional security layer to prevent unauthorized access.

GROUP-BASED MANAGEMENT

Permissions assigned to groups rather than individual users for consistency.

> This walkthrough demonstrates a complete implementation of AWS security best practices, creating a secure foundation for your cloud infrastructure that separates environments and enforces the principle of least privilege.