

Golang Warsaw #56

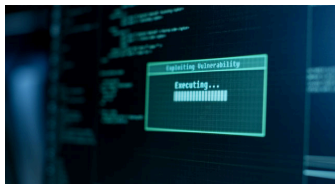
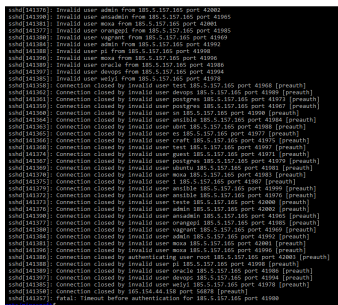
Jakub Wołynko

Secure access to EC2 (for developers)

Agenda

- introduction
- EC2 in **native** environment
- 3-tier architecture intro
- alternative methods of resource connection:
 - Bastion host
 - SSM
 - EC2 Instance Connect
- demo

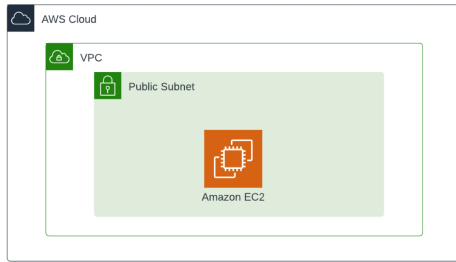
Why we should care?



- brute force attacks
- exploitation of security vulnerabilities
- weak-password attacks
- bots and scanners
- DDoS attacks

A regular virtual machine

It's a VM. Classic EC2 instance in default setting.



Resources:

EC2SecurityGroup:

Type: AWS::EC2::SecurityGroup

Properties:

GroupName: Launch-wizard-13

GroupDescription: Allow traffic to EC2

SecurityGroupIngress:

- CidrIp: 0.0.0.0/0

IpProtocol: -1

SecurityGroupEgress:

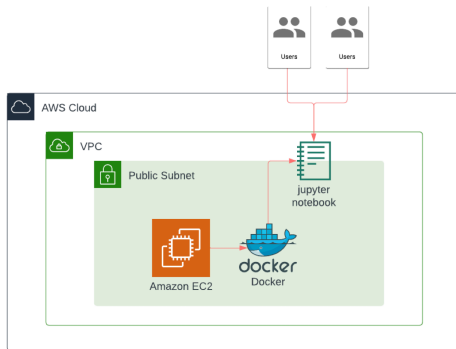
- CidrIp: 0.0.0.0/0

IpProtocol: -1

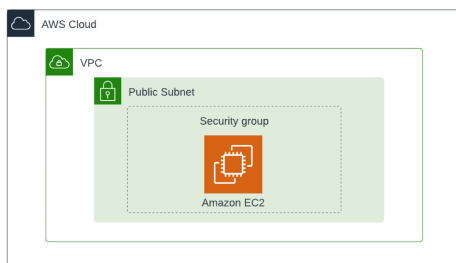
As you can see here, we have regular CloudFormation setup, which many of developers just produce. In 4 words - "it is not good"



So that's the story about secured SSH server.



Simple fix



Resources:

EC2SecurityGroup:

Type: AWS::EC2::SecurityGroup

Properties:

GroupName: Launch-wizard-13

GroupDescription: Allow traffic to EC2

SecurityGroupIngress:

- CidrIp: 3.145.12.1/32

IpProtocol: tcp

FromPort: 22

ToPort: 22

- CidrIp: 3.145.12.1/32

IpProtocol: tcp

FromPort: 8888

ToPort: 8888

SecurityGroupEgress:

- CidrIp: 0.0.0.0/0

IpProtocol: -1

```

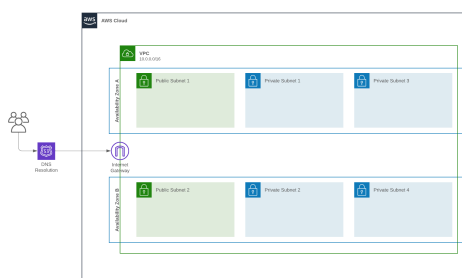
example-1.yaml
25: PublicKeyMaterial: "sch-ed25519 AAAAC3NzaC1lZ011NTE5AAAAIAC+VpNRyRPO
/GRYC98Y20Mhncn1sh2DPmtvo0TKA THS"
24:
23: InternetGateway:
22:   Type: AWS::EC2::InternetGateway
21:
20: VPC:
19:   Type: AWS::EC2::VPC
18:   Properties:
17:     CidrBlock: 10.0.0.0/16
16:     EnableDnsSupport: true
15:     EnableDnsHostnames: true
14:     InstanceTenancy: default
13:     Tags:
12:       - Key: Name
11:         Value: just-vpc
10:
9: SubnetA:
8:   Type: AWS::EC2::Subnet
7:   Properties:
6:     VpcId: !Ref VPC
5:     CidrBlock: 10.0.0.0/24
4:     MapPublicIpOnLaunch: true
3:
2: VPCGatewayAttachment:
1:   Type: AWS::EC2::VPCGatewayAttachment
0:   Properties:
- VpcId: !Ref VPC
- InternetGatewayId: !Ref InternetGateway
RouteTable:
Type: AWS::EC2::RouteTable
Properties:
VpcId: !Ref VPC
InternetRoute:
Type: AWS::EC2::Route
54:
NORMAL example-1.yaml 9 LSP 46%

```

Low-hanging fruit

- install fail2ban
- change SSH port. Use for example 31
- lock password login
- enable regular firewall
- disable default users
 - root
 - ubuntu
 - ec2-user
- use security groups or equivalent

3-tier architecture



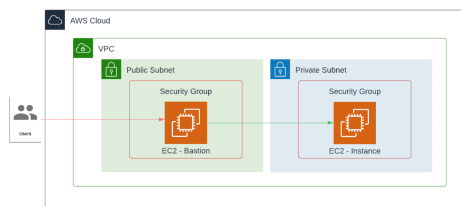
3-tier architecture - props and cons

- full control over resource access
- possibilities of disconnection resources from the internet
- in general, more secure

unfortunately...

- architecture getting more complex
- additional costs like VPN, NatGateway
- **regular resource access become more annoying**

solution one - bastion host



Host Instance

```
ProxyJump Bastion
PreferredAuthentications publickey
IdentitiesOnly=yes
IdentityFile /Users/kuba/.ssh/id_ed25519_gowaw56
User kuba
Hostname 10.0.10.60
Port 22
```

Host Bastion

```
PreferredAuthentications publickey
IdentitiesOnly=yes
IdentityFile /Users/kuba/.ssh/id_ed25519_gowaw56
User kuba
Hostname 35.158.161.105
Port 22
```




AWS Systems Manager

```
10 example-3.yaml
11 Properties:
12   PrivateEndpointEnabled: true
13   SecurityGroupIds: [!Ref EndpointSecurityGroup]
14   ServiceName: !Sub "com.amazonaws.${AWS::Region}.ssm"
15   SubnetIds: [!Ref PrivateSubnet]
16   VpcEndpointType: Interface
17   VpcId: !Ref VPC
18
19 SSMessagesEndpoint:
20   Type: AWS::EC2::VPCEndpoint
21   Properties:
22     PrivateEndpointEnabled: true
23     SecurityGroupIds: [!Ref EndpointSecurityGroup]
24     ServiceName: !Sub "com.amazonaws.${AWS::Region}.ssmmessages"
25     SubnetIds: [!Ref PrivateSubnet]
26     VpcEndpointType: Interface
27     VpcId: !Ref VPC
28
29 EC2MessagesEndpoint:
30   Type: AWS::EC2::VPCEndpoint
31   Properties:
32     PrivateEndpointEnabled: true
33     SecurityGroupIds: [!Ref EndpointSecurityGroup]
34     ServiceName: !Sub "com.amazonaws.${AWS::Region}.ec2messages"
35     SubnetIds: [!Ref PrivateSubnet]
36     VpcEndpointType: Interface
37     VpcId: !Ref VPC
38
39 EndpointSecurityGroup:
40   Type: AWS::EC2::SecurityGroup
41   Properties:
42     GroupDescription: Test security group from VPC Endpoint
43     VpcId: !Ref VPC
44     SecurityGroupIngress:
45       - IpProtocol: tcp
46         FromPort: 443
47         ToPort: 443
```

ssm - pros and cons

- connection can be created with a CLI and a GUI
- no need of jump station, or SSH key management
- integration with CloudTrail, IAM
- configuration is way more complex
- System Manager agent needs to be installed
- configuration of Ansible is possible, however annoying(aka use Saltstack)

ssm - costs

My Estimate

Estimate summary

| | | |
|-------------|--------------|----------------------|
| Uplink cost | Monthly cost | Total 12 months cost |
| 0.00 USD | 126.78 USD | 1,544.40 USD |
| | | Includes uplink cost |

Getting Started with AWS

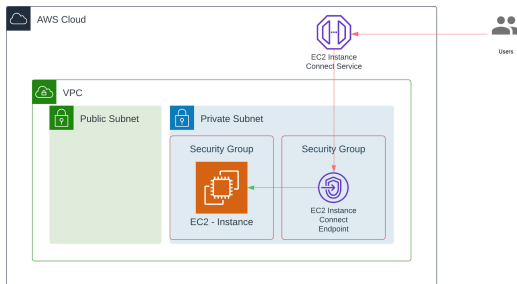
Get started for free | Contact Sales

My Estimate

Find resources

| Service Name | Status | Uplink... | Monthly cost | Description | Region | Cost Summary |
|--------------------------|--------|-----------|--------------|---------------|---------------|--|
| AWS Direct Connect (VPC) | | 0.00 USD | 76.12 USD | NAT Gateway | Europe (P...) | Working days per month (22) Number of Nat Gat... |
| AWS Direct Connect (VPC) | | 0.00 USD | 52.58 USD | VPC Endpoints | Europe (P...) | Number of VPC interface endpoints per AWS regio... |

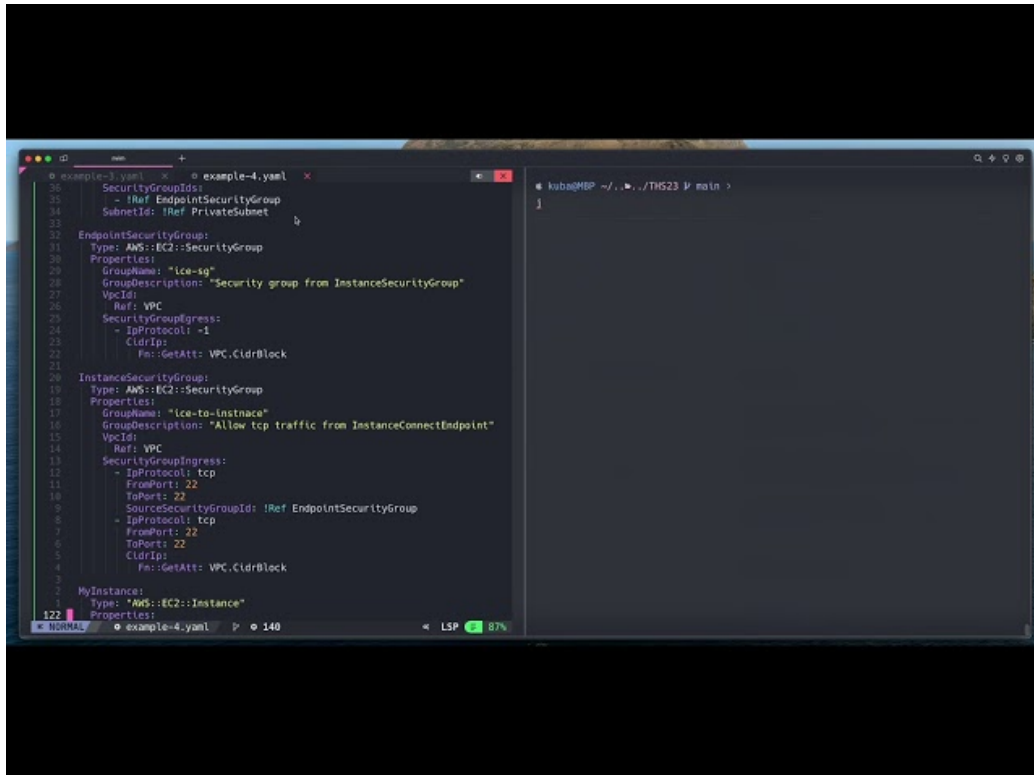
solution three - EC2 Instance Connect



- full name: EC2 Instance Connect Endpoint
- SSH/RDP session over dedicated service
- traffic goes through VPC Endpoint



EC2 Instance
Connect Service



EC2 Instance Connect - props and cons

- connection can be created with a CLI and a GUI
- no need of jump station, or SSH key management
- integration with CloudTrail, IAM
- configuration is more complex than bastion host, but easier than SSM
- not all instance families are supported

EC2 Instance Connect - costs

EIC Endpoint is available in all AWS commercial regions and the AWS GovCloud (US) Regions. There is no additional cost for using EIC endpoints. Standard data transfer charges apply. To learn more about EIC Endpoints see our documentation or blog post.

summary

- sometimes we still need to use SSH
- however we can **secure** our host properly
- and as a good thing, we can delegate it to provider
- unfortunately, it's not **free**

about services

- EC2 Instance Connect is most cost efficient
- SSM is most flexible solution

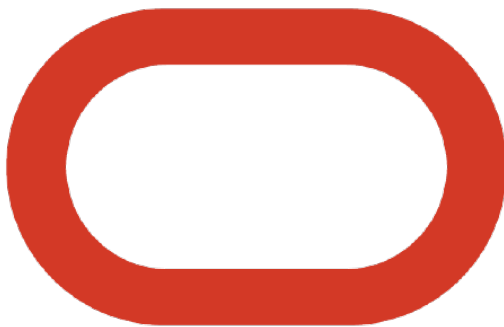
- Bastion host is the easiest solution

not only AWS

- Azure Bastion - fully managed service bastion host
- Run Command - solution similar to AWS System Manager(not Session manager)



- OCI Bastion - fully managed service bastion host



- OS Login - SSH Identity Management over GCP IAM
- Identity-Aware Proxy - similar solution to AWS EC2 Instance Connect



Any questions?