Golang Warsaw #56

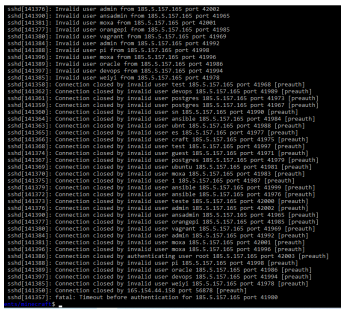
Jakub Wołynko

Secure access to EC2 (for developers)

Agenda

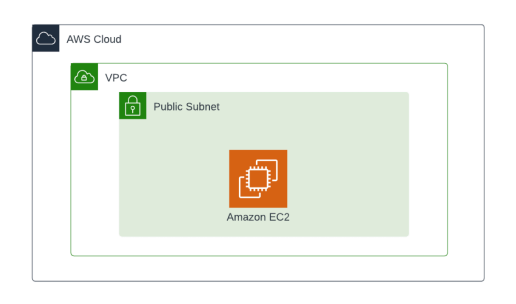
– introduction
– EC2 in **native** environment
– 3-tier architecture intro
– alternative methods of resource connection:
  – Bastion host
  – SSM
  – EC2 Instance Connect
– demo

Why we should care?



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

A regular virtual machine

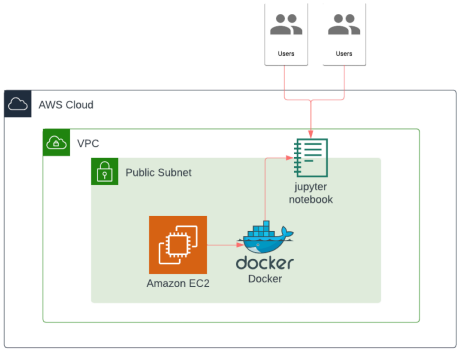


It's a VM. Classic EC2 instance in default setting.

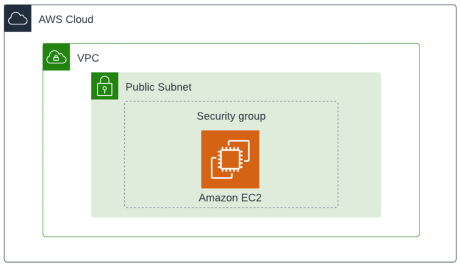
```
Resources:
  EC2SecurityGroup:
    Type: AWS:: EC2:: SecurityGroup
    Properties:
      GroupName: Launch-wizard-13
      GroupDescription: Allow traffic to EC2
      SecurityGroupIngress:
        - CidrIp: 0.0.0.0/0
          IpProtocol: -1
      SecurityGroupEgress:
        - CidrIp: 0.0.0.0/0
          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"

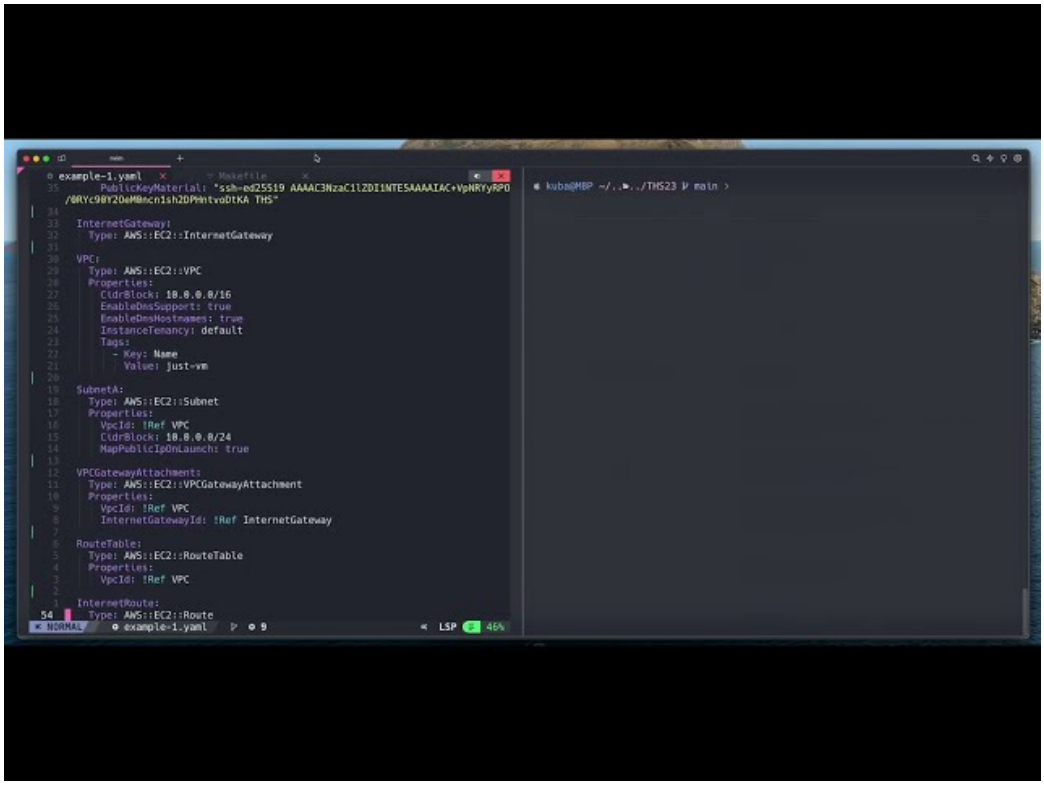
So that's the story about secured SSH server.



Simple fix



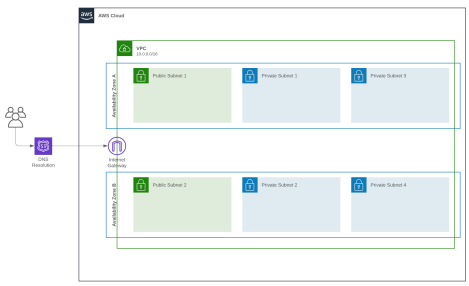

```
Resources:
  EC2SecurityGroup:
    Type: AWS:: EC2:: SecurityGroup
    Properties:
      GroupName: Launch-wizard-13
      GroupDescription: Allow traffic to EC2
      SecurityGroupIngress:
        - CidrIp: 3.145.12.1/32
          IpProtocol: tcp
          FromPort: 22
          ToPort: 22
        - CidrIp: 3.145.12.1/32
          IpProtocol: tcp
          FromPort: 8888
          ToPort: 8888
      SecurityGroupEgress:
        - CidrIp: 0.0.0.0/0
          IpProtocol: -1
```



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



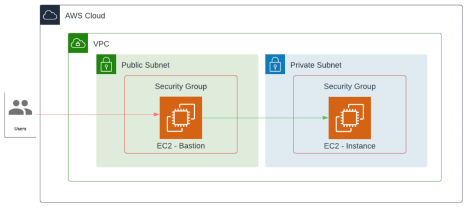
3-tier architecture - props and cons

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

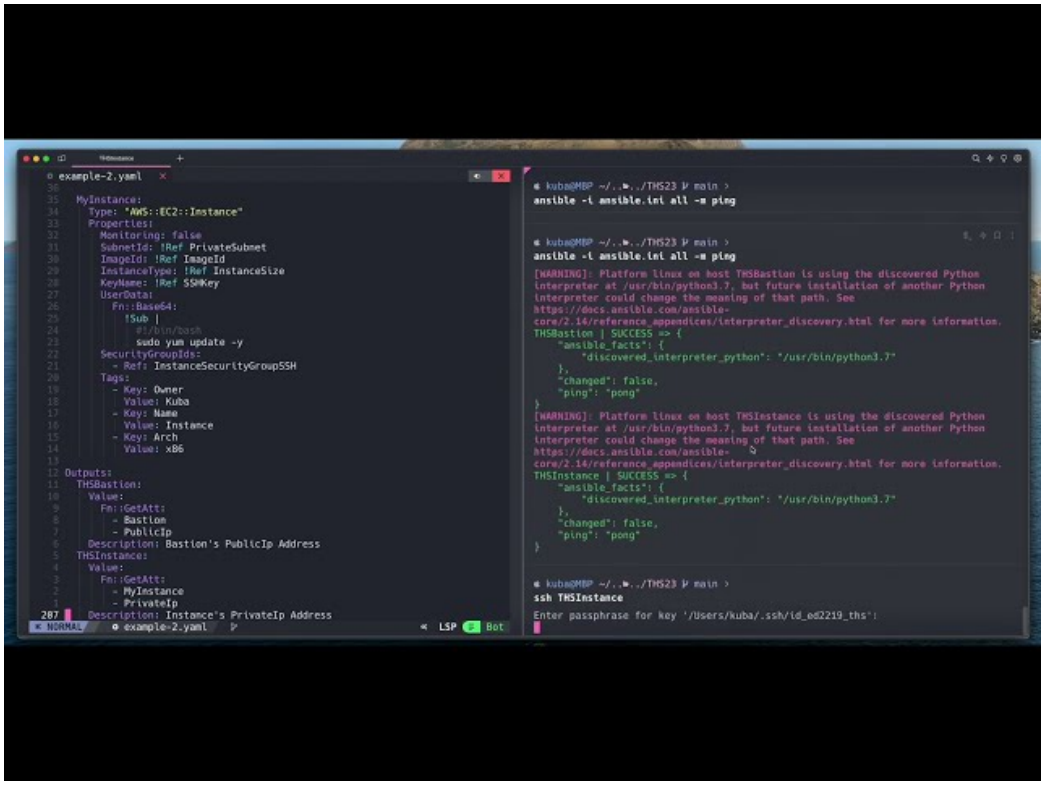
unfortunately...

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

solution one - bastion host



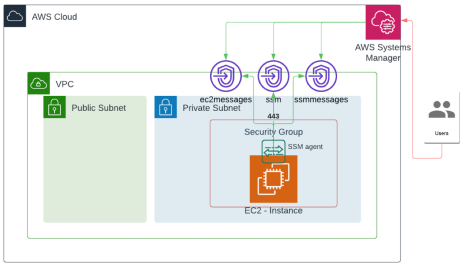

```
Host Instance
  ProxyJump Bastion
  PreferredAuthentications publickey
  IdentitiesOnly=yes
  IdentityFile /Users/kuba/.ssh/id_ed25519_gowaw56
  User kuba
  Hostname 10.0.10.60
  Port 22


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



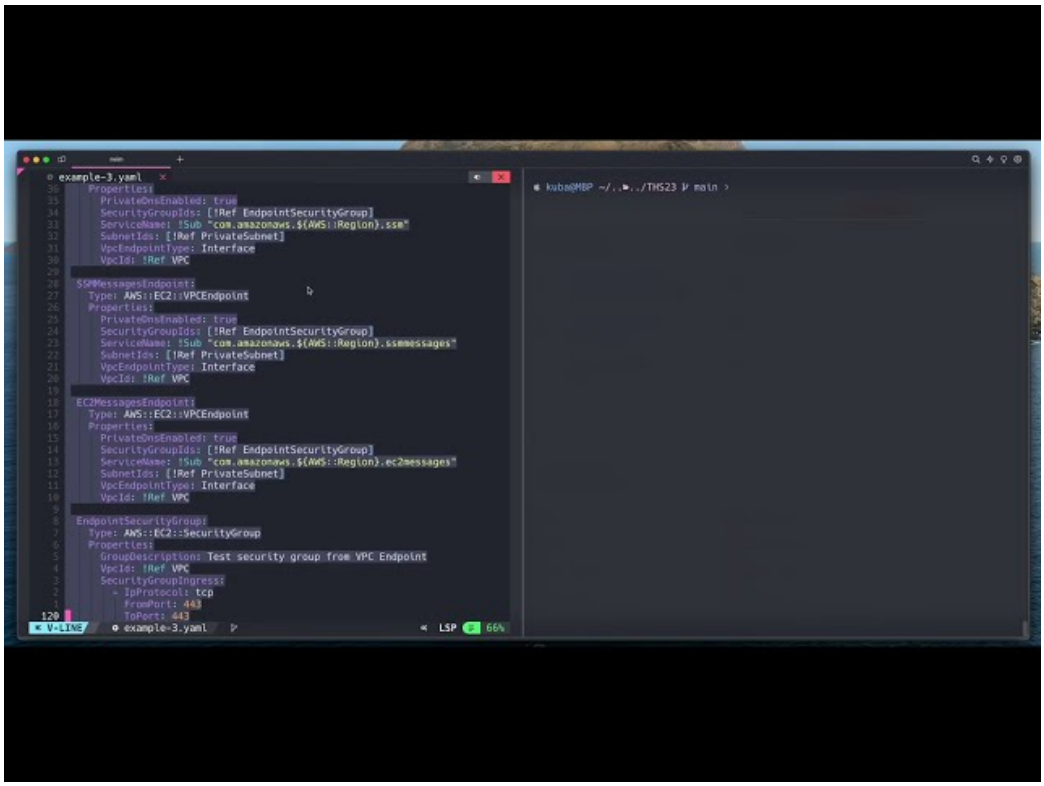
bastion host - props and cons

– very simple method, only ssh configuration required
– seamless support of tools like Ansible
– easy to start with
– bastion host is a host, we need to pay for it
– server as any other need to be maintained
– bastion host become single point of failure

solution two - ssm



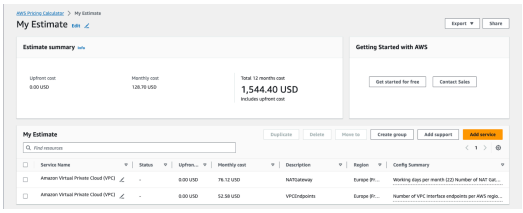
– to be precise AWS System Manager Session Manager
– sessions based on secure bi-directional channel
– traffic is encrypted with TLS 1.2



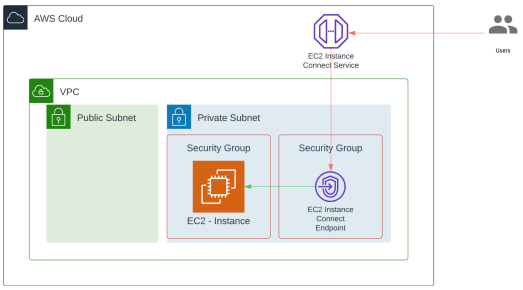
ssm - props and cons

– 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 way more complex
– System Manager agent needs to be installed
– configuration of Ansible is possible, however annoying(aka use Saltstack)

ssm - costs



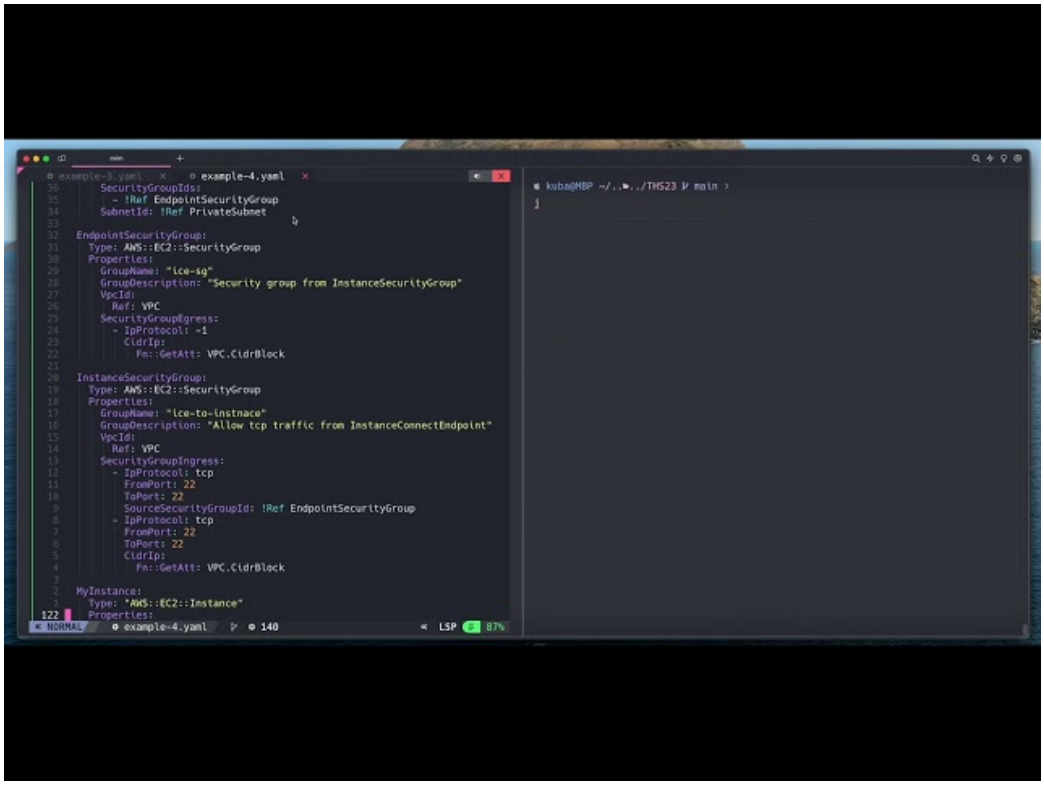
solution three - EC2 Instance Connect



– full name: EC2 Instance Connect Endpoint

– SSH/RDP session over dedicated service

– traffic goes through VPC Endpoint



EC2 Instance Connect - 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

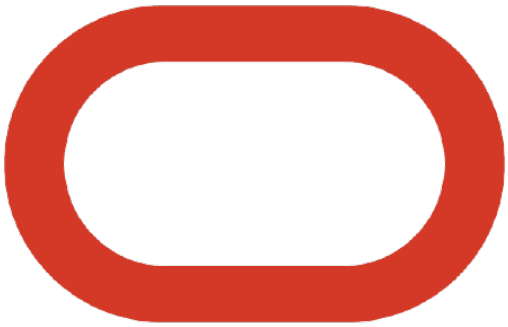
– EC2 Instance Connect is most cost efficient
– SSM is most flexible solution

– 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?