

aws
COMMUNITY DAY
INDONESIA

**EC2 Looked Normal, But Silently
Failed: Investigating Hidden
Issues Behind SSM**

Putu Sintia
Infrastructure Engineer at Zero One Group

Tangerang, 25 October 2025

aws
COMMUNITY DAY
INDONESIA

**EC2 Terlihat Normal, Tapi
Diam-Diam Gagal: Menyelidiki
Masalah Tersembunyi di Balik
SSM**

Putu Sintia
Infrastructure Engineer at Zero One Group

Tangerang, 25 Oktober 2025

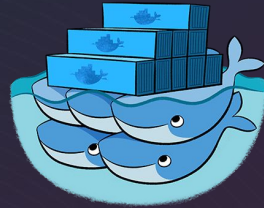
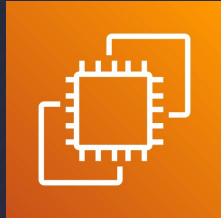


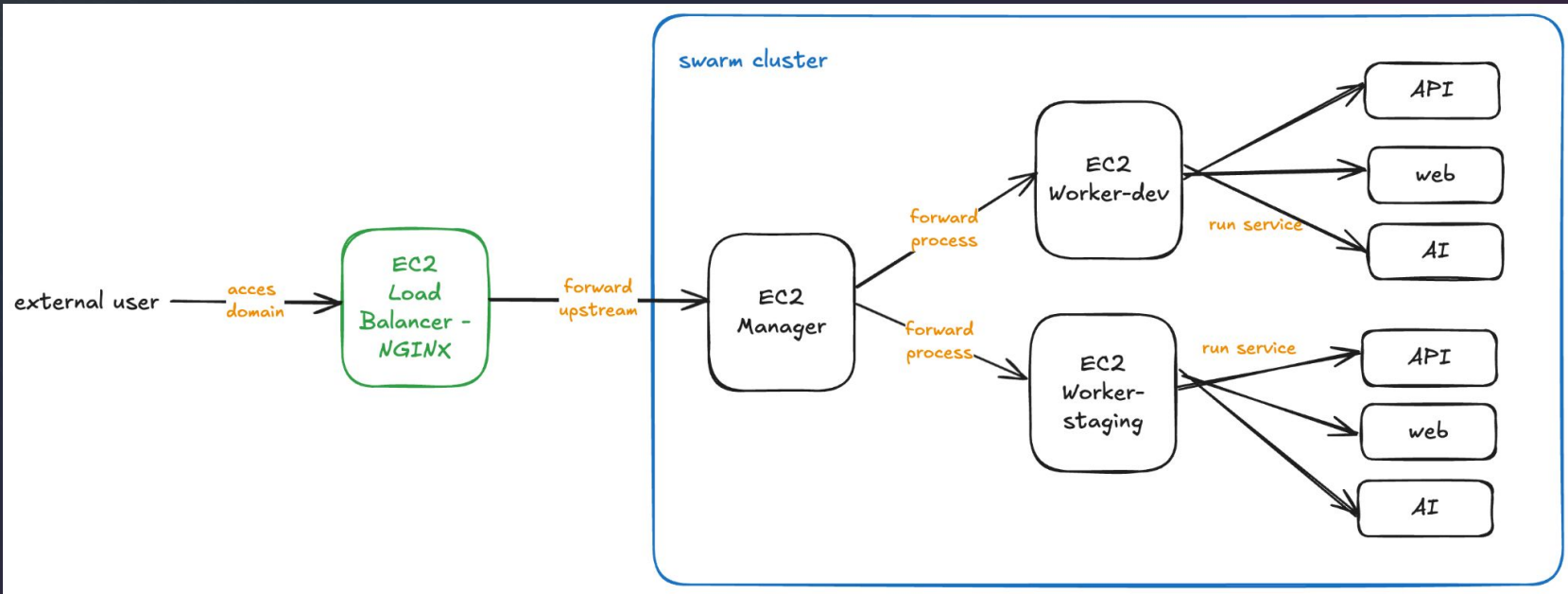
Sumber daya untuk sesi ini ada di link ini: <https://linktr.ee/SintiasInACDID2025>

Ini berdasarkan kisah nyata...

Semua itu berawal dari provisioning salah satu project . . .

Ngebangun lingkungan Development
dengan:





Awalnya setelah provision . . .

worker-dev	i-014bca27...	✔ Running	🔍 🔍	t3.small
worker-staging	i-06528c0f...	✔ Running	🔍 🔍	t3.micro
bastion	i-0eace075...	✔ Running	🔍 🔍	t3.micro
master-1	i-08ced2e4...	✔ Running	🔍 🔍	t3.micro
monitoring-1	i-09865cb5...	✔ Running	🔍 🔍	t3.medium
nginx	i-02feedab...	✔ Running	🔍 🔍	t3.small

Namun Ternyata . . .

```
ubuntu@master-1:~/...$ docker node ls
ID                HOSTNAME  STATUS  AVAILABILITY  MANAGER STATUS  ENGINE VERSION
tijnb27k1k1baqi125kh46d58 *  master-1  Ready   Active        Leader           28.3.0
ubuntu@master-1:~/...$ docker service ls
ID                NAME                MODE                REPLICAS  IMAGE                                  PORTS
p1sn5lki27a5     postgresql          replicated         0/1       prihuda22/postgresql:17.5-alpine3.22
ktpcy23harmq     node-exporter       global            1/1       prom/node-exporter:latest
d8e1sr61sjqf     portainer-agent     global            1/1       portainer/agent:2.21.5              *:9001->9001/tcp
ubuntu@master-1:~/...$
```

Proses Investigasi Masalah Lanjutan

Step 1: Cek Docker Swarm cluster

Step 2: Cek logs provisioning

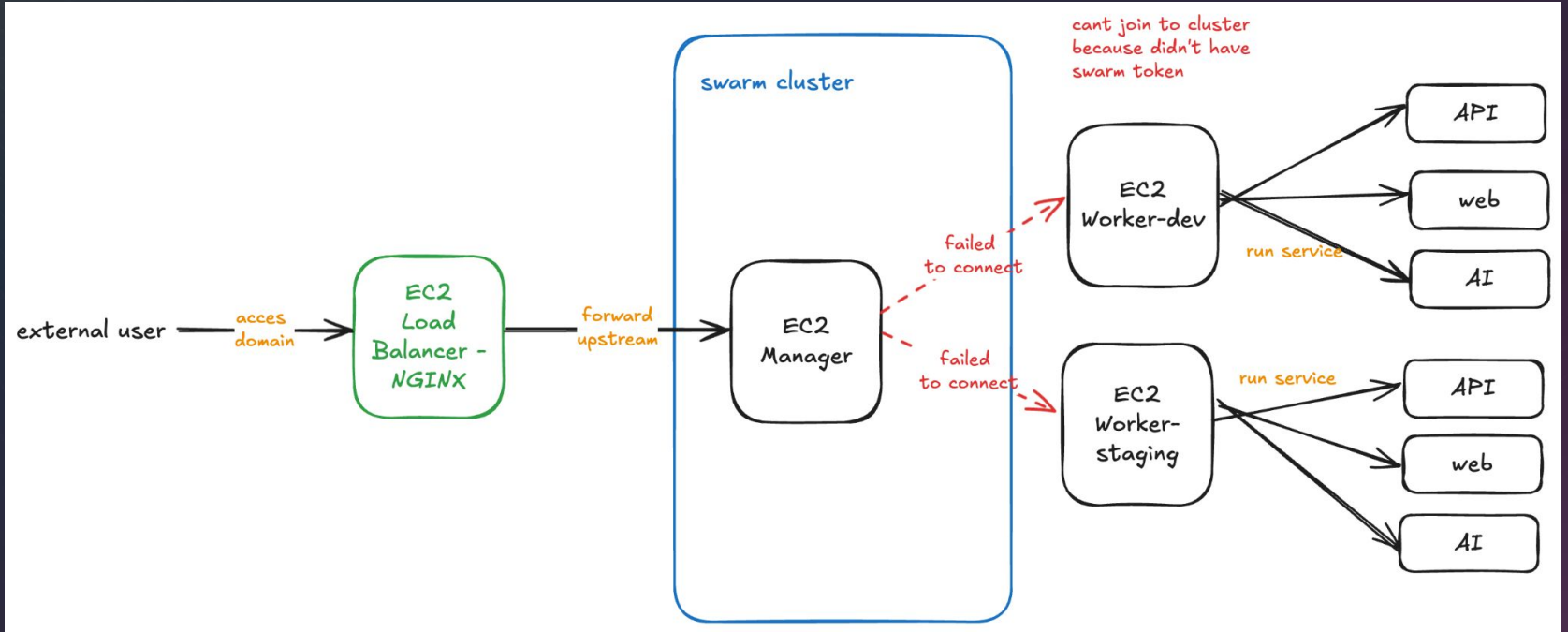
Step 3: Manual SSH ke worker

Step 4: Telusuri sumber token dari SSM

Step 5: Telusuri via source code Terraform



```
An error occurred (AccessDeniedException) when calling the GetParameter operation: User: arn:aws:sts: :  
<account-id>:assumed-role/worker-dev/i-05eba7b86f865b59e is not authorized to perform: ssm:GetParameter on  
resource: arn:aws:ssm:ap-southeast-1:<account-id> : parameter/<nama-projek>/swarm/worker-token because no  
identity-based policy allows the ssm:GetParameter action  
Failed to retrieve swarm information on attempt 1  
SSM Parameter Store status in region ap-southeast-1:
```



???

"Setelah menemukan sumber masalahnya ternyata ada pada parameter SSM yang tidak valid, saya pun mulai menggali lebih dalam—*apa sebenarnya SSM itu, dan kenapa bisa sepenting itu dalam proses provisioning?*"



AWS Systems Manager (SSM) adalah layanan dari AWS yang membantu kamu **mengelola server atau instance**. baik yang ada di AWS, di kantor (on-premise), atau di cloud lain. **secara otomatis dan terpusat**.

Beberapa fitur utamanya, yaitu **Session Manager, Run Command, Parameter Store, State Manager**, dan **Automation**

Struktur Parameter Store

Parameter Store menggunakan struktur hierarki

Contoh:

```
/<nama_projek>/swarm/join_token
```

IAM Policy

```
{  
  "Effect": "Allow",  
  "Action": ["ssm:GetParameter"],  
  "Resource": "arn:aws:ssm:<region>:<account-id>:parameter/buana/swarm/*"  
}
```

Contoh membaca parameter via CLI

```
aws ssm get-parameter \  
  --name "/<name_project>/swarm/join_token" \  
  --with-decryption \  
  --query "Parameter.Value" \  
  --output text
```

Contoh Implementasi di Terraform

<https://github.com/sintiasnn/iac-swarm>

Kesimpulan

- Provisioning otomatis bisa gagal tanpa error yang jelas
- Masalah kecil seperti path parameter bisa berdampak besar
- AWS SSM (Parameter Store) adalah alat powerful — tapi harus digunakan dengan teliti
- Validasi pasca provision penting untuk menghindari terjadinya silent failed

Terimakasih