CODE CREATOR

NetBird Private Access Gateway with Command Center

Customer User Guide

| Ubuntu LTS | AWS Marketplace AMI

Start here: Create and attach the IAM discovery role before launching the AMI. Without the role, the Secure Access Command Center can run, but AWS Resource Discovery will show zero resources.

Purpose: controlled remote access to private AWS resources through a self-hosted NetBird deployment, with Code Creator first-boot automation, AWS discovery, guided access policy creation, and live connection diagnostics.

1. What This Product Does

NetBird Private Access Gateway with Command Center by Code Creator is a self-hosted secure-access appliance. It is designed for teams that need employees, contractors, administrators, or managed devices to reach private resources in an AWS VPC without publishing those resources directly to the public internet.

Code Creator adds an operational layer around NetBird that reduces setup work and makes the private-access path visible. On first boot the appliance creates a customer-specific HTTPS endpoint, credentials, routing groups, a private-access network, a routing peer, and the Secure Access Command Center. The Command Center then discovers supported AWS resources and helps an administrator preview and apply narrowly scoped NetBird access rules.

Product scope: Code Creator Access Setup creates and manages NetBird resources and NetBird access policies. It does not modify AWS security groups, route tables, NACLs, or IAM policies for you.

NetBird feature note: NetBird geolocation-based posture functionality is disabled in this AMI. Core NetBird networking, routing, embedded identity, groups, peers, policies, setup keys, and the Code Creator private-access workflow remain available.

2. Before You Launch

- Launch the appliance into the VPC that contains the private resources you want remote users to reach.
- Use a public IPv4 address. The first-boot process automatically creates an sslip.io hostname from that address and configures HTTPS.
- Recommended instance type: t3.medium. Use a larger instance for heavier usage or larger peer counts.
- Create and attach the IAM discovery role described in Section 3 before launch.
- Allow the required inbound network ports shown below. Keep SSH limited to trusted administrator IP addresses.

Port	Protocol	Source	Purpose
22	TCP	Administrator IP only	SSH administration
80	TCP	Internet	HTTP redirect and Let's Encrypt certificate validation
443	TCP	Internet / approved client networks	NetBird dashboard, management, signal, relay, and Code Creator Command Center
3478	UDP	Internet	NetBird STUN for peer connectivity and NAT traversal

Outbound access should allow normal HTTPS/DNS connectivity required for package services, certificate issuance, AWS APIs, and NetBird client communication.

3. Required IAM Discovery Role

The Secure Access Command Center uses the EC2 instance role to read AWS resource and network metadata. The role is for discovery only; the Code Creator workflow does not require AWS write permissions to create security groups, routes, or instances.

3.1 Create the IAM permission policy

1. Open the AWS IAM console and choose Policies > Create policy.
2. Choose JSON and replace the editor contents with the policy below.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstances",
        "ec2:DescribeVpcs",
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeRouteTables",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeTags",
        "rds:DescribeDBInstances",
        "rds:DescribeDBClusters",
        "elasticloadbalancing:DescribeLoadBalancers",
        "elasticloadbalancing:DescribeTargetGroups",
        "elasticloadbalancing:DescribeListeners"
      ],
      "Resource": "*"
    }
  ]
}
```

3. Name the policy CodeCreatorNetBirdDiscoveryPolicy and create it.

3.2 Create the EC2 role

4. In IAM, choose Roles > Create role.
5. Trusted entity type: AWS service. Use case: EC2.
6. Attach CodeCreatorNetBirdDiscoveryPolicy.
7. Name the role CodeCreatorNetBirdDiscoveryRole and create it.

3.3 Attach the role when launching the AMI

8. On the EC2 launch page, open Advanced details.
9. For IAM instance profile, select CodeCreatorNetBirdDiscoveryRole.
10. Complete the launch. The role can also be attached later through EC2 > Actions > Security > Modify IAM role, but attaching it at launch provides the cleanest first-use experience.

If AWS Resources shows 0: First confirm that an EC2 IAM role is attached. After attaching or correcting the role, return to the Command Center and click Refresh Live Data. A reboot is normally not required for a newly attached instance profile.

4. Launch and First Login

11. Launch the AMI with a public IPv4 address, the IAM role above, your SSH key pair, the required security-group ports, and the VPC/subnet that contains or can route to the private resources you want to reach.
12. Wait for the EC2 status checks to pass and for first boot to complete. Do not manually start or reconfigure the NetBird containers during first boot.
13. SSH to the instance as **ubuntu** and display the generated login file:

```
cat ~/FIRST_LOGIN.txt
```

14. The file shows the generated NetBird Administration URL and credentials, plus the Code Creator Secure Access Command Center URL and credentials.

Protect this file: FIRST_LOGIN.txt contains customer-specific administrator credentials. Store the information securely and restrict SSH access to trusted administrators.

5. Secure Access Command Center

Open the Code Creator Secure Access Command Center URL shown in FIRST_LOGIN.txt. The Command Center provides an AWS-focused view of the private access path so administrators can see what is ready, what is missing, and what action to take next.

Area	What it tells you
AWS Resources	Shows discovered EC2, RDS, RDS clusters, and load balancers visible to the attached IAM role.
Remote Access	Shows connected remote-user peers, the Code Creator routing peer, and disconnected peer counts.
Access Policies	Shows active Code Creator private-access policies, protocol, and protected port.
Private AWS Resource	Shows the selected private EC2 target, private IP address, and current AWS instance state.
Refresh Live Data	Runs fresh AWS and NetBird collectors and updates the dashboard state.
Connection Diagnostics	Checks AWS discovery, target state, remote user, routing peer, access policy, and TCP reachability.
Access Setup	Guided dry-run/apply workflow that creates a NetBird private resource and a port/protocol-specific NetBird access policy.

Expected first view: Before a remote user is enrolled or an access policy is created, some diagnostic items can correctly show FAIL. Use the status labels as a setup checklist rather than treating every initial FAIL as a product failure.

6. Create Private Access with Access Setup

15. In the Command Center, click Refresh Live Data and confirm the target AWS resource is discovered and RUNNING.

16. Open Access Setup. Select the private target, choose TCP or UDP, and enter the service port you want remote users to reach.
17. Click Preview Access Plan. Preview is a dry run: nothing is changed.
18. Review the plan. Existing Code Creator network, groups, and routing peer should normally show REUSE. A new private resource and/or policy will show CREATE when required.
19. If the plan is correct, type APPLY when prompted and choose Apply Access.
20. Refresh Live Data again. Private Access Policy should show PASS. TCP Resource Reachability will show PASS when the selected TCP service is reachable from the routing appliance.

AWS firewall responsibility: The target resource must already allow the selected traffic from the routing appliance/VPC path. Code Creator does not open the target security group or NACL automatically.

7. NetBird Initial Helper

This section gets the first remote device connected. The product ships with NetBird local identity support and the Code Creator group cc-aws-remote-users. For human users, use a NetBird user/invite or your identity provider for production access. Setup keys are useful for quick validation and for managed/server devices.

7.1 Recommended for human users: invite a NetBird user

21. Open NetBird Administration using the URL from FIRST_LOGIN.txt.
22. Go to Team > Users and choose Add User / Invite User.
23. Enter the user name and email address. Where the UI offers auto-assigned groups, assign cc-aws-remote-users so that devices registered by that user receive the intended private-access policy.
24. Create the invite link and share it with the user through a secure channel. The user opens the link, sets a password, and signs in.
25. Install the NetBird client on the user device. Launch the desktop app and choose Connect, or run netbird up. The browser-based sign-in completes the enrollment.

NetBird local-user documentation: Local User Management

NetBird user onboarding documentation: Add users to your network

7.2 Quick validation / managed-device method: create a setup key

26. In NetBird Administration go to Settings > Setup Keys > Create Setup Key.
27. Name the key clearly, choose One-off for a single device, leave Ephemeral off for a persistent peer, and set Auto-assign groups to cc-aws-remote-users.
28. Create the key and copy it immediately to a secure location.

Setup-key scope: Current NetBird guidance recommends setup keys for servers, routing peers, automated deployments, and managed devices rather than normal human-user enrollment. Use user/IdP authentication when you need user identity, re-authentication, or identity-based policy behavior.

Setup-key documentation: Automate deployments and onboard machines with setup keys

7.3 Install the NetBird client

Download and install the NetBird client on the remote device. NetBird supports Windows, macOS, Linux, iOS, Android, and additional platforms.

All platforms: NetBird installation documentation

Windows: NetBird Windows installer documentation

7.4 Connect with a setup key

Open an elevated PowerShell/terminal on the remote device and use the exact management URL shown in FIRST_LOGIN.txt:

```
netbird up --management-url https://YOUR-NETBIRD-HOSTNAME --setup-key YOUR_SETUP_KEY
```

Example hostname format only: if the appliance public IP is 203.0.113.10, first boot may create a hostname such as 203-0-113-10.sslip.io. Always use the URL printed by your own FIRST_LOGIN.txt file.

29. Verify the client is connected:

```
netbird status
```

30. Return to the Code Creator Command Center and click Refresh Live Data. Remote Users should show a connected peer and Remote User Peer should show PASS when the device belongs to cc-aws-remote-users.

8. Validate Private Access

31. Make sure the private target is running and the intended application/service is listening on the port configured in Access Setup.
32. Refresh the Command Center. A fully prepared path should show PASS for AWS Resource Discovery, AWS Instance State, Remote User Peer, Routing Peer, Private Access Policy, and TCP Resource Reachability (for TCP targets).
33. From the enrolled remote device, connect to the target by its private VPC address and service port, for example:

```
http://10.0.1.25:8080
```

A successful connection proves the end-to-end path: remote device -> NetBird encrypted network -> Code Creator routing peer -> private AWS resource.

9. Fast Troubleshooting

Symptom	First action
AWS Resources = 0 / AWS Resource Discovery FAIL	Confirm the EC2 IAM role is attached and includes the discovery policy. Then click Refresh Live Data.
AWS Instance State FAIL	Start the selected target and refresh.
Remote User Peer FAIL	Confirm the client is connected and assigned to cc-aws-remote-users.
Routing Peer FAIL	Verify the appliance is running normally and review NetBird peer status in NetBird Administration.
Private Access Policy FAIL	Use Access Setup > Preview Access Plan, then apply the required policy if the plan is correct.
TCP Resource Reachability FAIL	Confirm the target application is listening, the target is running, and AWS security groups/NACLs permit the traffic from the appliance/VPC path.
Command Center data looks stale	Click Refresh Live Data once and allow the collectors to complete.

10. Operational Notes

- A normal reboot is supported; NetBird, Traefik, the dashboard, and the Code Creator Command Center are configured to return automatically.
- Do not delete or manually recreate the Code Creator groups or routing network unless you intentionally plan to rebuild the access model.
- Keep the NetBird administrator credentials and Command Center credentials private. Rotate credentials according to your organization's security practices.
- Use narrowly scoped ports and private targets. Avoid creating broad access policies when a specific private IP and service port will meet the requirement.
- This appliance is self-hosted in your AWS account. EC2, storage, data transfer, and other AWS service charges are separate from the AWS Marketplace software charge.

11. NetBird Documentation Links

- Install NetBird clients
- Windows client installation
- Setup keys
- Local user management
- Add users to your network
- Groups and access policies
- NetBird CLI
- Ports and firewalls

12. Quick Start Summary

34. Create CodeCreatorNetBirdDiscoveryPolicy and CodeCreatorNetBirdDiscoveryRole.
35. Launch the AMI in the target VPC with a public IPv4 address, required ports, SSH key, and IAM role attached.
36. Run cat ~/FIRST_LOGIN.txt and sign in to NetBird Administration and the Code Creator Command Center.
37. Click Refresh Live Data and confirm AWS resources are discovered.
38. Use Access Setup to preview and apply a narrow private-resource policy.
39. Invite a human NetBird user (recommended) or create a one-off setup key for quick/managed-device enrollment.
40. Install and connect the NetBird client, refresh the Command Center, and confirm the access path is ready.
41. Connect to the private AWS resource by its private IP and service port.

Code Creator - Customer-ready private access built around NetBird