To debug and fix common Terraform issues, start by ensuring your provider configuration is correct. Avoid hardcoding AWS credentials in your Terraform files; instead, use environment variables or an AWS profile.
For example, instead of specifying access_key and secret_key in the provider block, configure AWS CLI with
'aws configure' or use IAM roles.

If Terraform fails due to an invalid security group rule, such as an incorrect CIDR block (e.g., 256.256.256.256/32),
correct it by using valid IP ranges like 0.0.0.0/0 only when necessary.

When facing missing IAM permissions errors, such as 'iam:CreateRole' access being denied, ensure your Terraform user
or role has the appropriate IAM policies. Grant 'AdministratorAccess' or attach a policy that allows 'iam:CreateRole',
'iam:AttachRolePolicy', and 'iam:PassRole' actions.

To debug further, run 'terraform init' to check provider configurations, 'terraform validate' to detect syntax issues,
and 'terraform plan' to preview changes. If issues persist, check AWS CloudTrail logs for permission failures or use
'TF_LOG=DEBUG terraform apply' for detailed logs. Following these steps will help you quickly diagnose and fix Terraform misconfigurations.