Sensitive-Input-Variables

- [AWS RDS DB Instance](https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/db\_instance)

- [Vault Provider](https://learn.hashicorp.com/tutorials/terraform/secrets-vault?in=terraform/secrets)

- When using environment variables to set sensitive values, keep in mind that those values will be in your environment and command-line history

`Example: export TF\_VAR\_db\_username=admin TF\_VAR\_db\_password=adifferentpassword`

- Use sensitive = true

When you use sensitive variables in your Terraform configuration, you can use them as you would any other variable.

A screen shot of a computer program

Description automatically generated

Use sensitive = true

Create secrets.tfvars file to store your sensitive data

db\_username = "admin"

db\_password = "insecurepassword"

- Terraform will `redact` these values in command output and log files, and raise an error when it detects that they will be exposed in other ways.

- **\*\*Important Note-1:\*\*** Never check-in `secrets.tfvars` to git repositories

- **\*\*Important Note-2:\*\*** Terraform state file contains values for these sensitive variables `terraform.tfstate`. You must keep your state file secure to avoid exposing this data.

# Review the terraform plan

terraform plan -var-file="secrets.tfvars"

A screenshot of a computer

Description automatically generated

# Create Resources

terraform apply -var-file="secrets.tfvars"

# Verify Terraform State files

grep password terraform.tfstate

grep username terraform.tfstate

**## References**

- [Terraform Input Variables](https://www.terraform.io/docs/language/values/variables.html)

- [Resource: AWS EC2 Instance](https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/instance)

- [Resource: AWS Security Group](https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/security\_group)

- [Sensitive Variables - Additional Reference](https://learn.hashicorp.com/tutorials/terraform/sensitive-variables)