

# Terraform CLI Cheat Sheet

## Apply Resources

# Show proposed changes that an apply would perform  
terraform plan

# Apply all terraform changes  
terraform apply

# Apply a specific terraform resource  
terraform apply -target=<resource>  
i.e. terraform apply -target=aws\_db\_instance.cmdb

# Use a variables file with your terraform apply/plan  
terraform apply -var-file=<path\_to\_vars\_file>

## Destroy Resources

# Destroy all terraform resources  
terraform destroy

# Destroy select terraform resources  
terraform destroy -target=<resource>  
i.e. terraform destroy -target=aws\_db\_instance.cmdb

# Dry run terraform destroy  
terraform plan -destroy

# Mark a resource as tainted and force a destroy/recreate  
terraform taint <resource>

# Mark a tainted resource as clean  
terraform untaint <resource>

## Modules/Providers

# Initialize Terraform backend & download specified  
plugins/modules  
terraform init

# Clear out old modules/plugins & re-initialize terraform  
rm -rf ./.terraform/ && terraform init

# Pull modules into your .terraform directory  
terraform get -update=true

# Show terraform providers  
terraform providers

## State

# Pull the remote state  
terraform state pull > terraform.tfstate

# Push local state to remote state (uses file: terraform.tfstate)  
terraform state push

# Update local state file against real resources  
terraform refresh

# Tell Terraform a resource has been moved into a module  
terraform state mv <resource> <module>  
i.e. terraform state mv aws\_db\_instance.cmdb module.mydb

# Import an existing resource into terraform state  
# Note that this is different syntax for every resource  
terraform import <resource>  
i.e. terraform import aws\_instance.web1 i-abcd1234

## Debug

# Show current terraform resources  
terraform show

# Validate your terraform code  
terraform validate

# Show all terraform outputs  
terraform output

# Test resource interpolation (uses your state file)  
echo '<resource\_expression>' | terraform console  
i.e. echo 'aws\_db\_instance.cmdb.allocated\_storage ==  
"500" ? "lots" : "little"' | terraform console

# Visualize Terraform Dependency Graph (requires graphviz)  
terraform graph | dot -Tsvg > graph.svg

## Workspaces

# Create a terraform workspace  
terraform workspace new <workspace>

# Use a terraform workspace  
terraform workspace select <workspace>

# List terraform workspaces  
terraform workspace list

# Show current terraform workspace  
terraform workspace show