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>
```

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
```

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>
```

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
```