

Infrastructure Automation Test Framework – Test Design / Plan

Unit Test

Unit tests guarantee that module configuration updates are thoroughly validated, minimizing the risk of disruptions and ensuring seamless integration with your infrastructure

Test ID	Description	Input	Expected	AI – AutoFix
confProviderAWS1	Verify AWS provider configuration.	Terraform config with AWS provider	AWS provider configured correctly.	<ul style="list-style-type: none">Update the right provider versionUpdate the right region
confProviderAzure1	Verify Azure provider configuration.	Terraform config with Azure provider	Azure provider configured correctly.	Same as above
confResourceAWS1	Verify EC2 instance resource configuration	Terraform config with EC2 instance	EC2 instance resource configured correctly.	<ul style="list-style-type: none">Fix the Naming ConventionFix any difference in Image shape.Fix Image Type Fix any exposed IP / Ports
confResourceAzure1	Verify Azure VM resource configuration	Terraform config with Azure VM.	Azure VM resource configured correctly.	Same as above
confVariable1	Verify variable declaration	Terraform config with variable declaration	Variable declared correctly	<ul style="list-style-type: none">Update the right instance typeUpdate the right VPC configUpdate the right subnet configFix the naming convention.Fix if any deviation is policies
confVariable2	Verify variable type validation.	Terraform config with variable type mismatch	Error thrown for variable type mismatch.	<ul style="list-style-type: none">Update the right variable type
confOutput1	Verify output value	Terraform config with output value	Output value matches expected value	
confOutput2	Verify output format	Terraform config with output format.	Output format matches expected format.	

confErrorHandling1	Verify error handling for invalid provider.	Terraform config with invalid provider	Error thrown for invalid provider	NA
confErrorHandling2	Verify error handling for resource creation failure.	Terraform config with resource creation failure.	Error thrown for resource creation failure.	NA
Best Practice				
confBestPractice1	Verify Terraform formatting	Terraform config with formatting issues	Error thrown for formatting issues.	
confBestPractice2	Verify Terraform version.	Terraform config with outdated version.	Warning thrown for outdated version.	
confBestPractice3	Use Environment Variables	Parameterize tests for different environments	Fail if hardcoded values found	
confBestPractice4	Syntax Testing	Verify IaC code syntax and formatting	Catch syntax errors before deployment, preventing infrastructure failures	