**Real-time DevOps Day2day**

**Shell Scripting in a Deep Dive**

**Questionnaire**

**1. What is variable scope in shell scripting and why is it important?**

**2. How can external variables be passed to a shell script?**

**3. How can you check if variables need to be passed in a script?**

**4. How can you declare variables inside and outside a shell script?**

**5. How can you pass variables from an .env file to a script?**

**6. If a variable is defined in both an .env file and a script, which one takes precedence?**

**7. How can you handle outdated .env files in a shell script?**

**8. How can you force exit from a script if a required variable is missing?**

**9. How can you check for multiple required variables?**

**10. How can you check if a program exists before running a script?**

**11. How can you assign the output of a command to a variable?**

**12. What is AWS configure authentication and why is it important?**

**13. How can you prompt a user to enter variables in a shell script?**

**14. What is the purpose of exporting variables in shell scripting?**

**15. How can you run a command inside an if statement?**

**16. What is the purpose of $#?**

**17. How can you create an EC2 instance with a shell script?**

**In Detail**

**1. What is variable scope in shell scripting and why is it important?**

**Explanation:**  
Variable scope defines where a variable can be accessed in a script. Local variables are restricted to the function they are defined in, while global variables are accessible throughout the script. This prevents variable conflicts and enhances script reliability.

**Scenario:**  
You’re writing a script to manage EC2 instances. Use global variables for shared information like region and instance type, while local variables hold temporary data, e.g., response from an AWS CLI command.

#!/bin/bash

AWS\_REGION="us-east-1" **# Global variable**

create\_instance() {

local INSTANCE\_NAME="test-instance" **# Local variable**

echo "Creating $INSTANCE\_NAME in region $AWS\_REGION"

}

create\_instance

echo $INSTANCE\_NAME # This will **not** print anything as INSTANCE\_NAME is local.

**2. How can external variables be passed to a shell script?**

**Explanation:**  
Use command-line arguments ($1, $2, etc.) or environment variables to make scripts dynamic.

**Scenario:**  
Pass the EC2 name as an argument and the region as an environment variable:

#!/bin/bash

echo "Instance Name: $1"

echo "Region: $AWS\_REGION"

# Run the script

AWS\_REGION="us-east-1" ./**script.sh** my-instance

**3. How can you check if variables need to be passed in a script?**

**Explanation:**  
Validate inputs using conditionals to ensure required variables are provided.

**Scenario:**  
Ensure **both** the instance name and region are passed to avoid errors:

#!/bin/bash

**if** [ -z "$1" ] || [ -z "$AWS\_REGION" ]; then

echo "Usage: $0 <instance-name> (ensure AWS\_REGION is set)"

exit 1

**fi**

**4. How can you declare variables inside and outside a shell script?**

**Explanation:**  
Inside: Direct assignment (VAR=value).  
Outside: Use export to make it available to child processes.

**Scenario:**  
Use an exported variable for EC2 instance type:

# Outside the script

**export EC2\_TYPE="t2.micro"**

# Inside the script

#!/bin/bash

**echo** "EC2 Type: $EC2\_TYPE"

**5. How can you pass variables from an .env file to a script?**

**Explanation:**  
Use the source command to import variables.

**Scenario:**  
Manage AWS configurations in .env:

# .**env** file

AWS\_REGION="us-east-1"

INSTANCE\_NAME="dev-instance"

# Script

#!/bin/bash

**source** .env

echo "Region: $AWS\_REGION"

echo "Instance: $INSTANCE\_NAME"

**6. If a variable is defined in both an .env file and a script, which one takes precedence?**

**Explanation:**  
The script variable overrides the .env variable.

**Scenario:**  
Override the region for specific runs:

# Script

**source** .env

AWS\_REGION="eu-west-1" # Overrides the `.env` value

echo "Region: $AWS\_REGION"

**7. How can you handle outdated .env files in a shell script?**

**Explanation:**  
Check for the file’s existence and validate its content.

**Scenario:**  
Ensure the .env file contains necessary values:

#!/bin/bash

if [ ! -f .env ]; then

echo ".env file not found!"

exit 1

fi

source .env

if [ -z "$AWS\_REGION" ]; then

echo "Error: AWS\_REGION is missing in .env file."

exit 1

fi

**8. How can you force exit from a script if a required variable is missing?**

**Explanation:**  
Use exit with a non-zero status.

**Scenario:**  
Exit if INSTANCE\_TYPE is not set:

#!/bin/bash

if [ -z "$INSTANCE\_TYPE" ]; then

echo "Error: INSTANCE\_TYPE is missing."

exit 1

fi

**9. How can you check for multiple required variables?**

**Explanation:**  
Use the || operator.

**Scenario:**  
Ensure all required variables are present:

if [ -z "$AWS\_REGION" ] || [ -z "$INSTANCE\_TYPE" ]; then

echo "Error: Missing required variables."

exit 1

fi

**10. How can you check if a program exists before running a script?**

**Explanation:**  
Use command -v or which.

**Scenario:**  
Check if AWS CLI is installed:

if ! command -v aws &> /dev/null; then

echo "Error: AWS CLI is not installed."

exit 1

fi

**11. How can you assign the output of a command to a variable?**

**Explanation:**  
Use $(...) or backticks.

**Scenario:**  
Get the current date:

current\_date=$(date)

echo "Current Date: $current\_date"

**12. What is AWS configure authentication and why is it important?**

**Explanation:**  
aws configure stores credentials for secure AWS access. It ensures secure resource access.

**Scenario:**  
Configure AWS CLI before running automation scripts.

**13. How can you prompt a user to enter variables in a shell script?**

**Explanation:**  
Use read for user input.

**Scenario:**  
Prompt for EC2 instance name:

read -p "Enter EC2 name: " INSTANCE\_NAME

echo "Instance Name: $INSTANCE\_NAME"

**14. What is the purpose of exporting variables in shell scripting?**

**Explanation:**  
Exported variables are available to child processes.

**Scenario:**  
Pass credentials to a child script:

export AWS\_REGION="us-east-1"

./child\_script.sh

**15. How can you run a command inside an if statement?**

**Explanation:**  
Directly or with command substitution.

**Scenario:**  
Check for SSH key:

if [ -f ~/.ssh/id\_rsa ]; then

echo "SSH key exists."

fi

**16. What is the purpose of $#?**

**Explanation:**  
$# provides the number of arguments passed.

**Scenario:**  
Require at least two arguments:

if [ **$#** -lt 2 ]; then

echo "Error: Two arguments are required."

exit 1

fi

**17. How can you create an EC2 instance with a shell script?**

**Scenario:**  
Use AWS CLI to automate EC2 creation.

aws ec2 **run-instances** **\**

--image-id "ami-12345678" \

--instance-type "t2.micro" \

--key-name "my-key" \

--subnet-id "subnet-12345678" \

--count **1**

YouTube channel Link: <https://youtu.be/i7EjlQTGhbc?si=to92icvaOqzmct5M>