**Rocky Linux in Cloud Environments - Code and Scripts**

**1. Deploying a Rocky Linux Instance on AWS**

**Selecting and Launching a Rocky Linux AMI**

**Steps:**

1. **Log in to AWS Management Console**  
   Navigate to the AWS Console and enter your credentials.
2. **Launch an EC2 Instance**
3. aws ec2 run-instances --image-id ami-xxxxxxxxxxxxxxxxx --count 1 --instance-type t2.micro --key-name your-key --security-groups your-security-group
4. **Connect to the Instance via SSH**
5. ssh -i your-key.pem ec2-user@your-instance-ip
6. **Update the System**
7. sudo dnf update -y
8. **Create Additional Users (Optional)**
9. sudo useradd -m newuser
10. sudo passwd newuser
11. **Verify System Information**
12. uname -a
13. df -h

**2. Customizing and Creating a Custom AMI**

**Installing Essential Packages**

sudo dnf install -y nano wget curl git unzip

**Setting Up a Web Server (Optional)**

sudo dnf install -y httpd

sudo systemctl start httpd

sudo systemctl enable httpd

**Create a Custom AMI**

1. Stop the instance in AWS EC2 dashboard.
2. Create an image (AMI) from the stopped instance.

**3. Automating Deployment with Terraform**

**Terraform Configuration File (main.tf)**

provider "aws" {

region = "us-east-1"

}

resource "aws\_instance" "rocky\_linux" {

ami = "ami-xxxxxxxxxxxxxxxxx"

instance\_type = "t2.micro"

key\_name = "your-key"

}

**Deploy with Terraform**

terraform init

terraform apply -auto-approve

**4. Automating Deployment with AWS CloudFormation**

**CloudFormation Template (rocky-linux-template.yml)**

Resources:

RockyLinuxInstance:

Type: AWS::EC2::Instance

Properties:

ImageId: "ami-xxxxxxxxxxxxxxxxx"

InstanceType: "t2.micro"

KeyName: "your-key"

SecurityGroups:

- !Ref SecurityGroup

**Deploy the CloudFormation Stack**

aws cloudformation create-stack --stack-name rocky-linux-stack --template-body file://rocky-linux-template.yml

**5. Security and IAM Integration**

**Setting Up IAM Roles**

aws iam create-role --role-name EC2RockyLinuxRole --assume-role-policy-document file://trust-policy.json

**Verify IAM Role Access**

aws s3 ls

**Configuring Security Groups**

aws ec2 create-security-group --group-name rocky-security-group --description "Security group for Rocky Linux"

**6. Deploying Containers on Rocky Linux**

**Install Docker**

sudo dnf install -y docker

sudo systemctl start docker

sudo systemctl enable docker

**Run an Nginx Container**

docker pull nginx

docker run -d -p 80:80 nginx

**7. Monitoring and Logging**

**Install Prometheus**

sudo dnf install -y prometheus

**Install Grafana**

sudo dnf install -y grafana

**Enable AWS CloudWatch Logging**

sudo yum install -y amazon-cloudwatch-agent

**8. Automating with Ansible**

**Install Ansible**

sudo dnf install -y ansible

**Ansible Playbook for Nginx Deployment (nginx.yml)**

- hosts: servers

become: yes

tasks:

- name: Install Nginx

dnf:

name: nginx

state: present

**Run the Playbook**

ansible-playbook nginx.yml

This document compiles essential scripts and configurations required for deploying and managing Rocky Linux in cloud environments. It covers AWS EC2 instance setup, automation with Terraform and CloudFormation, security best practices, containerization, monitoring, and automation with Ansible.