# VCS Cloud Installation Guide

## 🌩️ Cloud Platform Setup

### AWS Installation

#### Step 1: Launch EC2 Instance

```bash
# Recommended instance types for VCS:
# - c5.4xlarge (16 vCPU, 32 GB RAM) - Basic simulations
# - c5.9xlarge (36 vCPU, 72 GB RAM) - Medium workloads
# - c5.18xlarge (72 vCPU, 144 GB RAM) - Large simulations
# - c6g.* (Graviton2) - ARM-based for optimized performance

# Launch instance with appropriate storage (minimum 100GB)
aws ec2 run-instances \
  --image-id ami-0abcdef1234567890 \
  --instance-type c5.9xlarge \
  --security-group-ids sg-903004f8 \
  --subnet-id subnet-6e7f829e \
  --block-device-mappings '[{
    "DeviceName": "/dev/sda1",
    "Ebs": {
      "VolumeSize": 200,
      "VolumeType": "gp3"
    }
  }]'
```

#### Step 2: Configure Security Groups

```bash
```

```
# Allow SSH access
aws ec2 authorize-security-group-ingress \
    --group-id sg-903004f8 \
    --protocol tcp \
    --port 22 \
    --cidr 0.0.0.0/0

# Allow license server access (if using remote license server)
aws ec2 authorize-security-group-ingress \
    --group-id sg-903004f8 \
    --protocol tcp \
    --port 27020-27030 \
    --cidr 10.0.0.0/16
```

### Azure Installation

#### Step 1: Create Resource Group and VM

```bash
# Create resource group
az group create --name VCS-ResourceGroup --location eastus

# Create VM with appropriate sizing
az vm create \
    --resource-group VCS-ResourceGroup \
    --name VCS-VM \
    --image UbuntuLTS \
    --size Standard_D16s_v3 \
    --admin-username vcsuser \
    --generate-ssh-keys \
    --os-disk-size-gb 200 \
    --storage-sku Premium_LRS

# Open necessary ports
az vm open-port --resource-group VCS-ResourceGroup --name VCS-VM --port 22
az vm open-port --resource-group VCS-ResourceGroup --name VCS-VM --port 27020-27030
```

### Google Cloud Platform (GCP)

#### Step 1: Create Compute Instance

```bash
```

```
# Create high-performance instance
gcloud compute instances create vcs-instance \
  --zone=us-central1-a \
  --machine-type=c2-standard-16 \
  --boot-disk-size=200GB \
  --boot-disk-type=pd-ssd \
  --image-family=ubuntu-2004-lts \
  --image-project=ubuntu-os-cloud \
  --tags=vcs-server

# Create firewall rules
gcloud compute firewall-rules create allow-vcs-license \
  --allow tcp:27020-27030 \
  --target-tags vcs-server
```

## 📦 VCS Software Installation

### Step 1: System Preparation

```bash
```

```
# Update system
sudo apt update && sudo apt upgrade -y

# Install required dependencies
sudo apt install -y \
    build-essential \
    gcc \
    g++ \
    make \
    tcsh \
    ksh \
    libxext6 \
    libxrender1 \
    libxtst6 \
    libxi6 \
    libxrandr2 \
    libxcursor1 \
    libxinerama1 \
    libfreetype6 \
    fontconfig \
    xauth \
    xvfb

# Create VCS user and directories
sudo useradd -m -s /bin/bash vcsuser
sudo mkdir -p /opt/synopsys
sudo chown vcsuser:vcsuser /opt/synopsys
```

## Step 2: Download and Extract VCS

```
bash
```

```
# Switch to VCS user
sudo su - vcsuser

# Create installation directory
mkdir -p /opt/synopsys/vcs
cd /opt/synopsys

# Download VCS installer (requires Synopsys account)
# Note: You need to download from https://solvnet.synopsys.com
# Example filename: vcs_vX.X-X_linux64.tar.gz

# Extract installer
tar -xzf vcs_vX.X-X_linux64.tar.gz
cd vcs_vX.X-X_linux64
```

### Step 3: Install VCS

```bash
# Run installer with silent mode
./installer -i silent \
  -DINSTALL_DIR=/opt/synopsys/vcs \
  -DLICENSE_FILE=27020@your-license-server.com \
  -DSELECTED_FEATURES=VCS,VCS_MX,Verdi

# Alternative: Interactive installation
./installer
```

## 🔐 License Configuration

### Step 1: Configure Cloud Licensing

```bash
```

```
# For cloud deployment, get VM UUID
export VM_UUID=$(sudo dmidecode -s system-uuid)
echo "VM UUID: $VM_UUID"

# For AWS specifically, you can also use Elastic IP
export AWS_EIP=$(curl -s http://169.254.169.254/latest/meta-data/public-ipv4)
echo "AWS EIP: $AWS_EIP"

# Configure license file for cloud
cat > /opt/synopsys/vcs/admin/license/synopsys.lic << EOF
# Cloud license configuration
SERVER your-license-server.com 27020
DAEMON snpslmd
USE_SERVER
# Add your actual license keys here
EOF
```

### Step 2: Start License Manager (if running local license server)

```bash
# Install SCL (Synopsys Common Licensing)
cd /path/to/scl_installer
./install_scl.sh -install_dir /opt/synopsys/scl

# Start license server
/opt/synopsys/scl/linux64/bin/lmgrd -c /opt/synopsys/vcs/admin/license/synopsys.lic -l /tmp/lmgrd.log

# Check license status
/opt/synopsys/scl/linux64/bin/lmstat -c /opt/synopsys/vcs/admin/license/synopsys.lic -a
```

## 🔧 Environment Setup

### Step 1: Create Environment Script

```bash
```

```
# Create VCS environment setup script
cat > /opt/synopsys/vcs/setup_vcs.sh << 'EOF'
#!/bin/bash

# VCS Installation Path
export VCS_HOME=/opt/synopsys/vcs
export PATH=$VCS_HOME/bin:$PATH

# License configuration
export SNPSLMD_LICENSE_FILE=27020@your-license-server.com
export LM_LICENSE_FILE=$SNPSLMD_LICENSE_FILE

# For cloud instances - use VM UUID
export SNPSLMD_HOSTID=$(sudo dmidecode -s system-uuid 2>/dev/null || echo "unknown")

# Library paths
export LD_LIBRARY_PATH=$VCS_HOME/linux64/lib:$LD_LIBRARY_PATH

# Display settings for X11 forwarding
export DISPLAY=${DISPLAY:-:0.0}

# VCS specific settings
export VCS_ARCH_OVERRIDE=linux64
export VCS_TARGET_ARCH=linux64

echo "VCS environment configured for cloud deployment"
echo "VCS_HOME: $VCS_HOME"
echo "License: $SNPSLMD_LICENSE_FILE"
echo "Host ID: $SNPSLMD_HOSTID"
EOF

chmod +x /opt/synopsys/vcs/setup_vcs.sh
```

## Step 2: Configure User Environment

```bash
# Add to user's bashrc
echo "source /opt/synopsys/vcs/setup_vcs.sh" >> ~/.bashrc

# Source the environment
source ~/.bashrc
```

## ✅ Installation Verification

### Step 1: Basic Tests

```bash
# Test VCS installation
vcs -help

# Test license connectivity
vcs -q -l vcs_license_test

# Test basic compilation
echo 'module test; initial $display("Hello VCS!"); endmodule' > test.v
vcs test.v -o test_sim
./test_sim
```

### Step 2: Performance Verification

```bash
# Test multicore capability
vcs -j 4 your_testbench.sv your_rtl.sv

# Test with coverage
vcs -cm line+cond+fsm+tgl your_design.sv
```

## 🚀 Optimization for Cloud

### Performance Tuning

```bash
# Configure for high-performance cloud instances
export VCS_EXEC_DONE=1
export VCS_MPI_SUPPORT=1

# Use all available cores
export VCS_NUM_PARALLEL_JOBS=$(nproc)

# Optimize memory usage
export VCS_MAX_MEMORY=80G  # Adjust based on instance memory
```

### Storage Optimization

```bash
# Use high-performance storage for simulation databases
mkdir -p /mnt/fast_storage/vcs_work
export VCS_WORK_DIR=/mnt/fast_storage/vcs_work

# Configure temporary directory on fast storage
export TMPDIR=/mnt/fast_storage/tmp
mkdir -p $TMPDIR
```

## 🔒 Security Considerations

### Firewall Configuration

```bash
# Restrict access to license ports
sudo ufw allow from 10.0.0.0/8 to any port 27020:27030
sudo ufw deny 27020:27030

# Allow SSH from specific IPs only
sudo ufw allow from YOUR_IP_RANGE to any port 22
```

### Data Protection

```bash
# Encrypt sensitive design data
sudo apt install ecryptfs-utils
mkdir -p ~/Private
chmod 700 ~/Private

# Use encrypted storage for design files
```

## 📊 Monitoring and Logging

### Setup Monitoring

```bash
```

```bash
# Monitor resource usage
htop
iostat -x 1

# Monitor VCS processes
ps aux | grep vcs

# Check license usage
lmstat -c $SNPSLMD_LICENSE_FILE -f
```

### Log Management

```bash
# Configure log rotation
sudo cat > /etc/logrotate.d/vcs << EOF
/tmp/vcs*.log {
    daily
    rotate 7
    compress
    missingok
    notifempty
}
EOF
```

## 🛠️ Troubleshooting

### Common Issues and Solutions

#### License Issues

```bash
# Check license server connectivity
telnet your-license-server.com 27020

# Verify hostid for cloud
lmhostid -ptype VM -uuid
# For AWS: lmhostid -ptype AMZN -eip
```

#### Performance Issues

```bash
```

```
# Check system resources
free -h
df -h
lscpu

# Monitor VCS processes
top -p $(pgrep vcs | tr '\n' ',')
```

### Network Issues

```bash
# Test license server connection
nmap -p 27020-27030 your-license-server.com

# Check DNS resolution
nslookup your-license-server.com
```

## 🔄 Automation Scripts

### Auto-start Script

```bash
# Create systemd service for VCS environment
sudo cat > /etc/systemd/system/vcs-setup.service << EOF
[Unit]
Description=VCS Environment Setup
After=network.target

[Service]
Type=oneshot
User=vcsuser
ExecStart=/opt/synopsys/vcs/setup_vcs.sh
RemainAfterExit=yes

[Install]
WantedBy=multi-user.target
EOF

sudo systemctl enable vcs-setup.service
```

### Backup and Deployment

```bash
# Create deployment snapshot
aws ec2 create-snapshot --volume-id vol-1234567890abcdef0 --description "VCS-Installation-Snapshot"

# Clone environment
aws ec2 create-image --instance-id i-1234567890abcdef0 --name "VCS-Ready-AMI"
```

This guide provides a comprehensive approach to installing VCS on major cloud platforms with proper licensing, security, and performance optimization.