# Introduction

In this lab we will install Terraform Enterprise in a private network with restricted access. This track is meant to simulate the experience of a system administrator working behind a corporate firewall. We'll configure Terraform Enterprise and connect it to a GitLab Server, and finally add a workspace with user and team access. The lab is broken into five parts:

- Install Terraform Enterprise
- Initial TFE Admin Setup
- Connect TFE to Version Control
- Create a VCS-backed workspace

- Configure Teams and Users

## The Lab Environment

The lab environment contains a Windows 2019 server workstation, a proxy server, a GitLab server and a Terraform Enterprise server. The TFE server and GitLab server are on a private network with no outbound access except through a proxy server. Your Windows workstation is a jump box that allows you to access the proxy, TFE, and GitLab servers. All your work during the labs should be done from your workstation via RDP. Here's a summary of the machines in the environment:

```
equipped-bedbug.workstation.training.hashidemos.io

equipped-bedbug.proxy.training.hashidemos.io
```

```
equipped-bedbug.gitlab.training.hashidemos.io

equipped-bedbug.tfe.training.hashidemos.io
```

# Log Onto Your Workstation

Note: It takes about ten minutes for your workstation to finish provisioning. If you try to log onto the workstation too early you may experience a reboot as the lab setup script requires it.
The first thing you should do is log onto your Windows workstation using an RDP Client. This workstation is a Windows Server 2019 instance with useful tools like Visual Studio Code, both Chrome and Edge web browsers, Notepad++, git, and openssh.
You need an RDP client to connect to a Windows workstation where you will be performing the steps for the rest of this lab. Check out the links below and install the client for your machine if you don't already have one.

- MacOS
- Windows
- Linux

Once installed you should add a new **PC** connection. Users on windows may see only a *Workspace* option, you need to download the full version of the RDP client from the Windows link above.

## Workstation Credentials

Use the credentials below to log onto your lab environment.

```
Workstation: equipped-bedbug.workstation.training.hashidemos.io

Username:    Administrator

Password:    HashiCorp123
```

Note: This is a technical training course on a private network. You can safely use an insecure password such as HashiCorp123 here. Don't do this in production!

## Configure Cmder

Cmder is a console emulator for Windows. We'll be using it for our SSH connections to our target instances.
Double-click on the lambda-shaped (λ) icon on your desktop. If you don't see the Cmder icon, your workstation setup has not finished - just give it a few more minutes.

## Test Your SSH Connections

Once you're logged onto your workstation you can simply open a Cmder prompt and SSH to your other machines. Try this now. Double click the Cmder icon on your desktop and connect to your TFE server:

```
ssh tfe
```

Type or copy/paste the following commands:

```
pwd

hostname

exit
```

Now do the same steps for your *gitlab* and *proxy* server as well. You can use the short names to connect just like the command above. You'll be running commands on the gitlab and tfe servers in the steps below. The easiest way to paste commands into the Cmder window is to use the right-click action of your mouse or trackpad. You can easily copy the commands listed below then paste them into your server's terminal window using right-click.

## Set Your Default Browser

Click the Windows start menu and select **Settings**. Type "browser" into the search field and select **Choose a default web browser**. Set your web browser to something other than Internet Explorer. Microsoft Edge and Google Chrome are preinstalled.

## Log onto Instruqt (again)

Now that you have a default browser set, you can log onto this track from within your remote workstation. This will make copy/paste operations a lot easier, trust me! Open Chrome or Edge and visit the following URL. Log on with your instruqt credentials:
https://play.instruqt.com/hashicorp/tracks/tfe-manual-install
Now you can stay within your remote workstation, and copying and pasting commands into your terminal should be seamless.

## Bookmark the Instruqt Lab URL

Bookmark this URL in case you have to close and re-open your browser window. This will allow you to easily get back to these instructions:
https://play.instruqt.com/hashicorp/tracks/tfe-manual-install/
Let's get started!

#  Lab 1: Install Terraform Enterprise

# Download and run the Installer script

Connect to the TFE server and gain a root shell:

```
ssh tfe

sudo /bin/su - root
```

Download and run the TFE installer. Note our use of the proxy environment variable so that the curl command will work properly.

```
export https_proxy=http://equipped-bedbug.proxy.training.hashidemos.io:3128

curl https://install.terraform.io/ptfe/stable > /root/install.sh

chmod +x install.sh
```

Run the TFE installer script.

```
./install.sh http-proxy=http://equipped-bedbug.proxy.training.hashidemos.io:3128
```

The installer will ask you for a Service IP address.

```
The installer was unable to automatically detect the service IP address of this machi
ne.

Please enter the address or leave blank for unspecified.

Service IP address:
```

You may simply leave this blank and hit [Enter]. The installer will fetch a supported version of docker and configure the Replicated console. After the installer is complete proceed to the next step.
We will be using the **Mounted Disk** operational mode. This means Terraform Enterprise will manage its own PostgreSQL database and object storage using a separate directory on the host. For this lab we are using a directory on the existing EBS volume attached to our instance, you may choose to mount a separate external volume for your production installation.
Create a directory to be used by TFE for storing its data:

```
sudo mkdir -p /var/tfe
```

# Open the Graphical Installer

You'll need to open this URL **from your cloud workstation** since it's not publicly exposed on the Internet. Use either Google Chrome or Microsoft Edge to access the graphical installer. You'll need to right-click this link and select "Open in new tab". You can also hold down the CTRL or CMD key and click the link to open it in a new tab.
http://equipped-bedbug.tfe.training.hashidemos.io:8800
You can also double-click on the red Replicated icon on your desktop. You remembered to set your default browser in Lab 1, didn't you?

## Bypass Browser TLS Warning

Since we don't have an TLS certificate configured yet you'll have to bypass the browser security warning. You can safely ignore the "Not secure" error from your browser and **Continue to Setup**. You'll need to click on the **Advanced** button to override the security warning and continue.

## HTTPS for admin console

The first page of the installer wizard is titled **HTTPS for admin console**. Here is where you'll configure your TLS certificate.
Enter your hostname into the **Hostname** field:

```
equipped-bedbug.tfe.training.hashidemos.io
```

We have already placed the ceritifcate and key on the TFE server during lab setup. If we had not done this you would upload them via the choose file buttons.
Since the certificate and key are already on the server, we just need to telll TFE where to find them.
Click on the link that says **If your private key and cert are already on this server, click here.** This link is right above the *Use Self-Signed Cert* button and does not show up as button.
Warning: DO NOT select `Use Self-Signed Cert`, your Instruqt check will not pass.
Enter the paths to your private key and certificate:
Private Key:

```
/root/privkey.pem
```

Certificate:

```
/root/fullchain.pem
```

Click on **Save & Continue**.

## Upload your license

Click on the **Choose license** button and browse to the **tfe_license.rli** file on your desktop. This is a TFE trial license that you can use for the labs. The installer will validate your license file and load the next step.
When presented with the "Choose your installation type" options, select **Online**.

## Secure the Admin Console

On the next page, choose a password for the Replicated admin console. Click **Continue**.
Note: We recommend using the same password for everything in this lab so it's easy to remember. `HashiCorp123` is a tested password that works everywhere.

## Preflight Checks

All of the checks should pass. Hit **Continue** to move on to Admin Settings.

## Settings

- Enter an **Encryption Password**, `HashiCorp123` is a good one 😉

- Scroll down to **Installation Type**. Select the **Mounted Disk** radio button.
- Under **Mounted Disk Configuration** specify `/var/tfe` as the path.
- Scroll further to **Proxy Bypass**. Enter your GitLab server's URL here:

```
equipped-bedbug.gitlab.training.hashidemos.io
```

Scroll to the bottom of the page and click **Save**. Click **Restart Now**.

## Refresh Your Browser

Close your browser window and double-click on the red R icon on your desktop, or right-click to open the link below. This will take you to the Replicated dashboard where you can watch the TFE application install process.
https://equipped-bedbug.tfe.training.hashidemos.io:8800/dashboard. Remember to right-click to open in a new tab.

## Wait for Startup to Complete

It will take two to three minutes for the application to fetch all its containers and start up. You'll know it is complete when the box in the upper left corner of the UI will show an arrow and the word **Started**.
🛑 Click the 'Check' button. If your environment is not configured correctly, you may have to go back and correct your work. Instruqt will give you hints as to what's missing. Stop here once you have passed the check.

# 👨‍🔧 Lab 2: Initial TFE Admin Setup

In this lab we'll do the initial administrator setup of your TFE server.

## Open the Terraform Enterprise App

Head over to the Replicated console:
https://equipped-bedbug.tfe.training.hashidemos.io:8800/dashboard
You can also double-click the Replicated icon on the desktop.
Click on the **Open** link underneath the **Stop Now** button. It's blue text on a purple background so you might not see it right away. This will bring you to the Terraform Enterprise initial setup page.
Note: You *must* use the link on the Replicated dashboard to get to the initial user setup page. There is a single-use token embedded into the sign-up URL that allows the first user to be created. If you try to access the Terraform Enterprise login page directly you'll simply be presented with a login screen.

## Create an Admin Account

Create your initial administrator account. Choose a password that you can remember easily. The password must be at least 10 characters in length. Please use `admin@example.com` as your administrator email address. This allows you to use your own email address later in the users and groups lab. Example:

```
Username: admin

Email:    admin@example.com

Password: HashiCorp123
```

## Create an Organization

Create an organization called `myorg` and enter `admin@example.com` for the email address. Do not create a workspace yet, as we'll be doing that in the next section.
Note: The org must be called `myorg` as this is what the check script verifies

## Enable MFA

Enable multi-factor authentication for the admin account. You can reach the settings page by clicking on your user icon in the upper right corner, or use this direct link: https://equipped-bedbug.tfe.training.hashidemos.io/app/settings/two-factor
You must use the **Application** option because your TFE server is unable to send SMS messages from the private network. If you don't have a TOTP application you can use Google Authenticator, or Microsoft Authenticator. Multi-factor authentication ensures that even if your admin credentials are exposed, that malicious users will not be able to compromise your account (unless they steal your cell phone and your password).
Note: MFA must be enabled as the check script verifies this

## Set up Email Notifications

Next you'll configure an outbound mail server so you can invite new users to your organization and receive notifications. Click on your user icon in the upper right corner and select **Admin**. Click on **SMTP** on the left side menu and configure the following settings. Here you can use your real email address in the **Send Test Email To:** field.

```
Enable email sending with SMTP (checked)

Sender Email:          training-bot@hashidemos.io

Send Test Email To:    you@example.com

Host:                  equipped-bedbug.proxy.training.hashidemos.io

Port:                  25

Authentication:        none

Username and Password: Leave blank
```

Click on **Save SMTP Settings**. You should receive an email at the test address you configured above.

## Check Your Work

Warning: From this point forward Instruqt requires credentials to check your work. Please run the `checkmywork` script in the **Shell** tab inside of Instruqt to enter your Terraform User token. These will be used to connect to your environment and verify each challenge. You can generate a Terraform User token at the following URL:
https://equipped-bedbug.tfe.training.hashidemos.io/app/settings/tokens
Once you have created a token run the script in the Shell tab of your Instruqt lab and enter your TFE credentials.
Note: This script must be run in the **Shell** tab inside of Instruqt, not on your Windows workstation.

```
/usr/bin/checkmywork
```

🛑 Click the 'Check' button. Stop here once you have passed the check.

# Lab 3: Connect TFE to GitLab

In this section we'll connect TFE to GitLab using OAuth to enable VCS-driven infrastructure and policy enforcement. TFE supports OAuth authentication with several popular VCS platforms.

## TFE - Add a VCS Provider

### TFE VCS Provider Configuration

Go into your organization settings by selecting your org name from the top left and then clicking **Settings**. This is where you configure settings that apply to all team members and workspaces in your oganization.
Let's configure a VCS connection. Click on **Providers** under **Version control** on the left side menu. Click on the purple **Add a VCS Provider** button.
Click GitLab and select "Gitlab Community Edition". You can see the other options for supported VCS platforms on this page.
Then provide the following values which tell Terraform to use our GitLab server's v4 API.

```
Name:            Gitlab Community Edition (unchanged)
HTTP URL:        https://equipped-bedbug.gitlab.training.hashidemos.io
API URL:         https://equipped-bedbug.gitlab.training.hashidemos.io/api/v4
```

Click **Continue** button on the bottom of the page. Leave this browser tab open, you'll need to copy some items from it in the next step.

### GitLab Application Link

Open your GitLab server and log in with the following details:

```
Username:         root
Password:         HashiCorp123
```

In this lab environment we're using the default admin user, in real world usage this would probably be a dedicated service user. Note that this account MUST have admin access because TFE will use this account to configure webhooks on the repositories for which integration is enabled.
Click on the profile icon in the top left corner then click **Preferences** then select **Applications** from the left pane. Or simply copy and paste the link below into your workstation browser to go directly to the Applications page:
https://equipped-bedbug.gitlab.training.hashidemos.io/profile/applications
Add a new application by copying over the settings from the TFE window and hitting **Save application**:

```
Name:                   Terraform <YOUR ORG>
Redirect URI:           <CALLBACK URL DISPLAYED BY TFE>
Confidential:           ✅
Expire Access Tokens:   ✅
Scopes:                 ✅ api
```

On the next page you will see the unique Application ID and Secret for this app.

### Connect Your Organization

Back on the Terraform Enterprise tab, copy and paste the **Application ID** and **Secret**. Then, Click **Continue**
You'll be taken to a confirmation page on the GitLab server to confirm the link. Click on the **Authorize** button.
Congratulations, you've connected a Terraform Enterprise organization to your GitLab server! You can skip the optional **Set up SSH keypair** step. As noted on the page, it is only required if your organization used Git submodules which can only be accessed via SSH.
🛑 Click the 'Check' button. Stop here once you have passed the check.

#  Lab 4: Provision a VCS-backed Workspace

In this section we'll create a simple git repo, populate it with some terraform code, and connect it to a workspace.

## Create a New Project and Repo

Back on your GitLab Server, click the **New Project** button on the main page.
Select the **Create blank project** option. Call your project `myrepo`. The Project slug will autopopulate and the description is optional. Click the **Create project** button.

## Clone the Repo to your Desktop

Click the **Clone** button on your project page an copy the the **Clone with HTTPS** url. You can ignore the warning about SSH Keys at the top.
On the Windows **Workstation** open a Cmder prompt and run the following commands to clone the repo and open it in Visual Studio Code.

```
git clone <HTTPS URL>

code ./myrepo
```

You'll be prompted to enter your GitLab username (`root`) and password (`HashiCorp123`) when you clone the repository. You can avoid this by configuring SSH for authentication if you would like.
Warning: You'll be prompted for your GitLab username and password when you run the clone command. These will be stored on your workstation so you don't have to type them in for each git command. If you make a mistake while typing the password, you'll need to go into the Windows Credential Manager to reset it. You can find this by searching for 'Credential Manager' in your Windows settings.
You should now have a new folder on your desktop called **myrepo**.

## Create some Terraform

Right click on the **myrepo** directory and select **Open with Code**. This will open the Visual Studio Code editor inside the myrepo directory.
Create a new file in the **myrepo** folder called **main.tf**. Copy the following code into the file and save it.
Note: Float your mouse over the **MYREPO** text in the file explorer to reveal the **New File** button. It looks like a small piece of paper with a folded corner. Or you can right-click in the empty space in the Explorer sidebar and select **New File**. Be sure and name it `main.tf`.
Code for your `main.tf` file:

```
# A 20 sided die. Don't roll a 1!

resource "random_integer" "d20" {

  min = 1

  max = 20

  keepers = {

    # Generate a new id each time

    timestamp = timestamp()

  }
```

```
}


# Rolls the d20 and reports the result.
resource "null_resource" "roll-1d20" {
  provisioner "local-exec" {
    command = "echo 🎲 Your roll is: ${random_integer.d20.result} 🎲"
  }
}
```

## Commit and Push your Changes

Back in your Cmder prompt run the following commands to push your code to the remote GitLab repo. Note that you are switching to a bash shell by running `bash` so the Git commands will work correctly. You can simply copy and paste this block of code into a Cmder prompt:

```
bash
cd ~/Desktop/myrepo
git config --global user.email "admin@example.com"
git config --global user.name "Admin Example"
git add main.tf
git commit -m "initial commit"
git push
```

Note: You can also add, commit, and push files to your GitLab server right from the Visual Studio Code GUI if you would like.

## Configure a New TFE Workspace

Back on your TFE server, click on **Workspaces**. Click on the **New workspace** button.
Select **Version control worfklow**, then **GitLab Community Edition** for your version control provider.
You'll see an option called `root/myrepo` in the repository list. Select it.
Leave the workspace name as is. It should read **myrepo**. Click on **Create workspace**.
In a minute or two you'll see the message change to **Configuration uploaded successfully**.

## Trigger a Terraform Apply

The first Terraform run must be kicked off manually. Subsequent runs will happen whenever you make changes to your VCS repo.

Click the **Actions** button in the upper right corner and select **Start new plan**. You can put anything you want as the reason, or leave the reason field blank. Team members may use these fields to communicate intentions to those that may review the run later.
You'll see the `terraform plan` phase run and stop. Examine the plan, and click **Confirm & Apply** at the bottom. Enter some text if you like, and click **Confirm Plan**.
You should see the apply finish and the result of your dice roll. Hope you rolled a 20!
Good job! You installed a VCS-backed infrastructure as code delivery pipeline behind your corporate firewall.
Note: GitLab blocks webhook requests to all private network endpoints. If you are working in a customer environment you may need to add TFE to an allowlist. For this lab we have pre-configured GitLab to allow all webhooks.
🛑 Click the 'Check' button. Stop here once you have passed the check.

#  Lab 5: Configure Teams and Users

In this lab you'll work with users and teams.

## Create an Admins Team

Select your **myorg** organization on your TFE server and go into the organization settings. It's the main **Settings** link at the top of the page.
Once there click on the **Teams** link on the left side of the page. Create a new team called **admins**.
Note: Team must be called admins as this is what the check script verifies
Next we'll grant some admin-level privileges to the admins team. Check all three of these boxes and click the **Update team organization access** button:

```
☑ Manage Policies
☑ Manage Workspaces
☑ Manage VCS Settings
```

## Grant Admin Access to Your Workspace

Click on the **Workspaces** menu and select your **myrepo** workspace. Go into the workspace settings by using the **Settings** pulldown menu on the right side of the page. Select **Team Access** from the pulldown menu.
Note: There are two **Settings** menus, one for the organization and one for your workspaces.
Grant the admins team admin level access to the workspace using the pulldown menus and **Add team** button.

## Invite a User

Head back into your organization's **Settings** menu. Click on the **Users** link on the left side menu. Next, click on the **Invite a user** button on the upper right side of the page.
Here you'll need to enter an email address where you can receive email. Usually your personal or work email will work fine. Place your new user on the **admins** team using the pulldown menu, and hit the **Invite user** button.
You should receive your welcome email within two or three minutes. If you don't receive this email check your spam or junk folder.
Once you have your invitation email, you'll need to copy the invitation link by right clicking on the purple **Accept Invitation** button. Because the TFE server is on a private network, you cannot simply click on this button to enable your account.

## Accept the Invite

Log out of Terraform Enterprise first so you can use your invitation link to activate your user. Click the icon in the upper right corner and select **Log Out**.
Copy the invitation link from your email and paste it into the web browser *inside* your workstation. You'll be taken to an **Accept your invitation** page where you can create a new username and password. These can be whatever you like.
Finally, choose the **myorg** organization on the next page by clicking the **Accept** button.
Congratulations, now you have an administrator account that you can use to work with the **myorg** organization. The initial account you created earlier should be reserved only for site admin activities like creating new organizations or configuring system-wide settings.
Click the 'Check' button. Stop here once you have passed the check.

# Appendix A: TLS Certificates

Below are an TLS wildcard certificate and private key for equipped-bedbug.training.hashidemos.io. We have also put copies of the certificate and private key on your TFE and GitLab servers. These are provided here for reference and easy copy-and-paste. Note that the certificate contains the intermediate issuer certificate from LetsEncrypt as well as the actual cert for your animal name.
Private Key:

```
-----BEGIN RSA PRIVATE KEY-----
MIIEpgIBAAKCAQEA27TMiB9A3I0oF9SY3632tzs8rs1+JUBIDK9LNBSUN0hXea7u
i76UfqtLx7qg+qZCUncXE5Rh7D/l1h4O+0t8jTTIVjBf1W1DcGTkvcA8Accx3MbL
2RVI295Bkz3lRgT5mHb4RevUZrx15NUfnLwmLH9adRjfA3nzaA98pQd+CDClCC3P
TAvTPpTuapBXtkIFqHA2V1KVmVS3k/xoZY5vD1GCD/y+rZxGqyE/Zq+h2+bOQq7v
UhyLqARpIHCqzOC4/tm4nw5BVb3MnGbPuvAGMWYPzbK4lYQV+YQXMEX+GpFEifjX
YNGljv4lXHhvUiW0xp7wEaKHbLidt3pHz9r38QIDAQABAoIBAQCW0VnF545i2BM5
qJr4kTbXOTbC2BVMBQBwlLSPH8FO4b4Krebazwyol3YBuT9gUFkeutmAe09tGb/w
```

```
Z9no7zykwiLz52kh3Ut6EAhlqVyH6/FymJS+hDnrlHp3VPnaQvgDjUeI2AaKL7zO
RO7abN3XleTGlTgh7skEGf16W9ZMT7HWfUXQLMizPTjvcS5PbrKN63TKyPhNbSig
lIch+ZOPlLE9tIKEQe9Oax2oKxUulbjSfzSkuFOrZAsqigNhmM9co5dB0knK/rTz
C97koGk//btaIkXtN805afntBCRpmzyS4XKjVLO1Hf82Fhzj4ApvAkKbB6BDUCpx
JlFYRmpBAoGBAP45yI1ajvrmhs7du2luh2DHHIeLgQNUOW9ZKiBjbk/F/j43lDqm
Q2RCQwBaIB+JoxQDYiIUgrbP31ZF+QJf4Yo8neGEcrZL5RVXtnKQS6CFehOqSfmD
XEUG+uZQoibVJK3gdNBVAzYU27U8ys9Y3RRw+OsVhxPZNJjhMZKWMwKtAoGBAN09
VzwV2JHIZUbJkgwSWOrHEkAdy8W4aJ1Nhl92F4hHeuk9jQwAlZ1qygLB8ZqfC7bs
PiAX5vISZLkuPwnwVI5ZbCQZNdiryuwzrmsMX4hV0gSD7pWaJXsJwd8BeYqrGjOo
6ZTNFEA2rp/9sAVMFc8pH3GM52wjo86Q21L4pnbVAoGBAOOhZ6Kx2F5ttX1a6qkf
r5YHEtOHJPCPJJI/+0le3nINbXXWHCJbpvIJpqaVDDhQ7DVKRj8PsG7yOB6urqTi
pzc4yYEZMAFseWWf0RxJeX30pfcfvfbtGNwxwMwinN46cQpDdTTpOePdz1A7a7k2
71WTRj5wWy95h+/vi4yecnRNAoGBALqKL/WPje1NhGctKiFXjvVWKIM7HEd39206
oqP21FypuG7U4QpPq9bw5adxk58Fn0D67F7vCGDvNVIWNjzRENp9d3rwbsBkCVMj
OFs2MY6onc4E17P4JsYZMggwHCTZo9wO/fPi/sIt2Qs5QaMVxgXH7KU3YyGA42sW
JHdPBDyJAoGBANvetYpT2X04YfStmFz8RsGfcWfp0F6oYF3kTIEmwzAfUqLO2SkY
/okcRRfc+B9lN/Y4vVBtitIeKKhj7AcvolXKWJPhqkJNUHy0liVTn3qOkQDmrgjA
B+XKRwj1vsZECkvM+qbP/IfGFaKfjDsYU3O3aa7/xicxYHUZSUX5USOF
-----END RSA PRIVATE KEY-----
```

Certificate:

```
-----BEGIN CERTIFICATE-----
MIIFqTCCBJGgAwIBAgISBCMala+dqj8pdW2cuaqNSkt+MA0GCSqGSIb3DQEBCwUA
MDIxCzAJBgNVBAYTAlVTMRYwFAYDVQQKEw1MZXQncyBFbmNyeXB0MQswCQYDVQQD
EwJSMzAeFw0yMzA4MDcxNzM3MzJaFw0yMzExMDUxNzM3MzFaMCMxITAfBgNVBAMM
GCoudHJhaW5pbmcuaGFzaGlkZW1vcy5pbzCCASIwDQYJKoZIhvcNAQEBBQADggEP
ADCCAQoCggEBANu0zIgfQNyNKBfUmN+t9rc7PK7NfiVASAyvSzQUlDdIV3mu7ou+
lH6rS8e6oPqmQlJ3FxOUYew/5dYeDvtLfI00yFYwX9VtQ3Bk5L3APAHHMdzGy9kV
SNveQZM95UYE+Zh2+EXr1Ga8deTVH5y8Jix/WnUY3wN582gPfKUHfggwpQgtz0wL
0z6U7mqQV7ZCBahwNldSlZlUt5P8aGWObw9Rgg/8vq2cRqshP2avodvmzkKu71Ic
i6gEaSBwqszguP7ZuJ8OQVW9zJxmz7rwBjFmD82yuJWEFfmEFzBF/hqRRIn412DR
pY7+JVx4b1IltMae8BGih2y4nbd6R8/a9/ECAwEAAaOCAsYwggLCMA4GA1UdDwEB
/wQEAwIFoDAdBgNVHSUEFjAUBggrBgEFBQcDAQYIKwYBBQUHAwIwDAYDVR0TAQH/
BAIwADAdBgNVHQ4EFgQUh5/6VfZvgsJnoD66FGOwIdffSlkwHwYDVR0jBBgwFoAU
```

```
FC6zF7dYVsuuUAlA5h+vnYsUwsYwVQYIKwYBBQUHAQEESTBHMCEGCCsGAQUFBzAB
hhVodHRwOi8vcjMuby5sZW5jci5vcmcwIgYIKwYBBQUHMAKGFmh0dHA6Ly9yMy5p
LmxlbmNyLm9yZy8wgc4GA1UdEQSBxjCBw4IiKi5iaXRidWNrZXQudHJhaW5pbmcu
aGFzaGlkZW1vcy5pb4IfKi5naXRsYWIudHJhaW5pbmcuaGFzaGlkZW1vcy5pb4Ie
Ki5wcm94eS50cmFpbmluZy5oYXNoaWRlbW9zLmlvghwqLnRmZS50cmFpbmluZy5o
YXNoaWRlbW9zLmlvghgqLnRyYWluaW5nLmhhc2hpZGVtb3MuaW+CJCoud29ya3N0
YXRpb24udHJhaW5pbmcuaGFzaGlkZW1vcy5pbzATBgNVHSAEDDAKMAgGBmeBDAEC
ATCCAQQGCisGAQQB1nkCBAIEgfUEgfIA8AB2ALc++yTfnE26dfI5xbpY9Gxd/ELP
ep81xJ4dCYEl7bSZAAABidFKRFwAAAQDAEcwRQIhAN5rgSYsXXenXKyLi7zELd7I
5cHOe+8KFrcN7KizMLcJAiBcYvgH05Hpt250VRFcLQM31c+06D5OGFOdFr5xSind
HAB2AHoyjFTYty22IOo44FIe6YQWcDIThU070ivBOlejUutSAAABidFKRG0AAAQD
AEcwRQIgBuvoQvm7Mzv3F0Noj1+vmF/mPMuGJYYEueLRfYk2FLQCIQC7hJJ31fp8
DJTRAQosmOJqQwrXmDDsIMSlwHVNI7IdGDANBgkqhkiG9w0BAQsFAAOCAQEAYDwM
191wFYv92IjOTlNtJ5cjgyB8hwzWs0Mn7nzQqTU1nAPmEtYY9xnTHQmZxghnuAKU
1mCPJh2C6jTuGsvKFL4MI2YGV3NlGGqjJ3XJ20ujwVA8+M6FCabnep6V0evbDcOc
Rzh5DU4Sww0i454ZddNOqSR8oCMfACC2B96LUxAMt8+TIeQ/2cKRfWf9XPZoP6h5
CdmpzI4WOZ2FXqFYuWI/VvEilbAYB9DVb6qtoT5VpilxwroJojjoMmtKMWF+1fCx
q84GvJ3w9tpTTCmMYl3pHPm8iavUcEmUniH5lPcTpIoK/6FyXJtieZG1Sh8f4RA7
HgxbLAj25kbI4oLBZA==
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIFFjCCAv6gAwIBAgIRAJErCErPDBinU/bWLiWnX1owDQYJKoZIhvcNAQELBQAw
TzELMAkGA1UEBhMCVVMxKTAnBgNVBAoTIEludGVybmV0IFNlY3VyaXR5IFJlc2Vh
cmNoIEdyb3VwMRUwEwYDVQQDEwxJU1JHIFJvb3QgWDEwHhcNMjAwOTA0MDAwMDAw
WhcNMjUwOTE1MTYwMDAwWjAyMQswCQYDVQQGEwJVUzEWMBQGA1UEChMNTGV0J3Mg
RW5jcnlwdDELMAkGA1UEAxMCUjMwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEK
AoIBAQC7AhUozPaglNMPEuyNVZLD+ILxmaZ6QoinXSaqtSu5xUyxr45r+XXIo9cP
R5QUVTVXjJ6oojkZ9YI8QqlObvU7wy7bjcCwXPNZOOftz2nwWgsbvsCUJCWH+jdx
sxPnHKzhm+/b5DtFUkWWqcFTzjTIUu61ru2P3mBw4qVUq7ZtDpelQDRrK9O8Zutm
NHz6a4uPVymZ+DAXXbpyb/uBxa3Shlg9F8fnCbvxK/eG3MHacV3URuPMrSXBiLxg
Z3Vms/EY96Jc5lP/Ooi2R6X/ExjqmAl3P51T+c8B5fWmcBcUr2Ok/5mzk53cU6cG
/kiFHaFpriV1uxPMUgP17VGhi9sVAgMBAAGjggEIMIIBBDAOBgNVHQ8BAf8EBAMC
AYYwHQYDVR0lBBYwFAYIKwYBBQUHAwIGCCsGAQUFBwMBMBIGA1UdEwEB/wQIMAYB
Af8CAQAwHQYDVR0OBBYEFBQusxe3WFbLrlAJQOYfr52LFMLGMB8GA1UdIwQYMBaA
```

```
FHm0WeZ7tuXkAXOACIjIGlj26ZtuMDIGCCsGAQUFBwEBBCYwJDAiBggrBgEFBQcw
AoYWaHR0cDovL3gxLmkubGVuY3Iub3JnLzAnBgNVHR8EIDAeMBygGqAYhhZodHRw
Oi8veDEuYy5sZW5jci5vcmcvMCIGA1UdIAQbMBkwCAYGZ4EMAQIBMA0GCysGAQQB
gt8TAQEBMA0GCSqGSIb3DQEBCwUAA4ICAQCFyk5HPqP3hUSFvNVneLKYY611TR6W
PTNlclQtgaDqw+34IL9fzLdwALduO/ZelN7kIJ+m74uyA+eitRY8kc607TkC53wl
ikfmZW4/RvTZ8M6UK+5UzhK8jCdLuMGYL6KvzXGRSgi3yLgjewQtCPkIVz6D2QQz
CkcheAmCJ8MqyJu5zlzyZMjAvnnAT45tRAxekrsu94sQ4egdRCnbWSDtY7kh+BIm
lJNXoB1lBMEKIq4QDUOXoRgffuDghje1WrG9ML+Hbisq/yFOGwXD9RiX8F6sw6W4
avAuvDszue5L3sz85K+EC4Y/wFVDNvZo4TYXao6Z0f+lQKc0t8DQYzk1OXVu8rp2
yJMC6alLbBfODALZvYH7n7do1AZls4I9d1P4jnkDrQoxB3UqQ9hVl3LEKQ73xF1O
yK5GhDDX8oVfGKF5u+decIsH4YaTw7mP3GFxJSqv3+0lUFJoi5Lc5da149p90Ids
hCExroL1+7mryIkXPeFM5TgO9r0rvZaBFOvV2z0gp35Z0+L4WPlbuEjN/lxPFin+
HlUjr8gRsI3qfJOQFy/9rKIJR0Y/8Omwt/8oTWgy1mdeHmmjk7j1nYsvC9JSQ6Zv
MldlTTKB3zhThV1+XWYp6rjd5JW1zbVWEkLNxE7GJThEUG3szgBVGP7pSWTUTsqX
nLRbwHOoq7hHwg==
-----END CERTIFICATE-----
```