Subdomain & Custom Domain Setup Guide

BudStack Domain Configuration

Table of Contents

Overview

Domain Structure

BudStack uses a multi-level domain system:

```
Main Platform:      budstack.to
Wildcard DNS:       *.budstack.to
Tenant Stores:      [tenant-slug].budstack.to
Custom Domains:     tenant-custom-domain.com
```

Examples: - `healingbuds.budstack.to` → HealingBuds Portugal store - `greenmedicine.budstack.to` → Another tenant - `healingbuds.pt` → Custom domain (points to HealingBuds)

Subdomain System

Automatic Subdomain Creation

Process: 1. Tenant completes onboarding 2. Super admin approves tenant 3. System generates unique slug from business name 4. Namecheap API creates CNAME record 5. Subdomain becomes active within minutes

Subdomain Naming Rules

Valid: - `healingbuds` - `green-medicine` - `cannabis-clinic-2024`

Invalid: - `Healing Buds` (spaces) - `-healingbuds` (starts with hyphen) - `healingbuds-` (ends with hyphen) - `hb` (less than 3 characters)

Rules: - Lowercase letters only - Numbers allowed - Hyphens allowed (not at start/end) - 3-50 characters - Must be unique

Slug Generation

Algorithm:

```typescript
function generateSlug(businessName: string): string {
  return businessName
    .toLowerCase()
    .trim()
```

```
    .replace(/[^a-z0-9\s-]/g, '') // Remove special chars
    .replace(/\s+/g, '-')         // Replace spaces with hyphens
    .replace(/-+/g, '-')          // Collapse multiple hyphens
    .substring(0, 50);            // Limit length
}
```

Examples: - "Healing Buds Portugal" → `healing-buds-portugal` - "Green Medicine & Wellness" → `green-medicine-wellness` - "Cannabis Clinic 2024" → `cannabis-clinic-2024`

Namecheap API Integration

API Setup

Requirements: 1. Namecheap account 2. API access enabled 3. API key generated 4. Server IP whitelisted

Enabling API Access

Step 1: Request API Access 1. Log in to Namecheap account 2. Go to Profile → Tools → Namecheap API Access 3. Click "Enable API Access" 4. Accept terms and conditions

Step 2: Get API Credentials - **API User:** Your Namecheap username - **API Key:** Generated key (copy and save securely)

Step 3: Whitelist Server IP 1. Find your server's public IP: `curl ifconfig.me` 2. Add IP to Namecheap API whitelist 3. Current whitelisted IP: `52.34.76.202`

Environment Configuration

Add to `.env`:

```
NAMECHEAP_API_USER="your_username"
NAMECHEAP_API_KEY="your_api_key_here"
NAMECHEAP_DOMAIN="budstack.to"
NAMECHEAP_IP_WHITELIST="52.34.76.202"
```

API Implementation

File: `lib/namecheap-api.ts`

```
export async function createSubdomainCNAME(subdomain: string) {
  const apiUrl = 'https://api.namecheap.com/xml.response';

  const params = {
    ApiUser: process.env.NAMECHEAP_API_USER,
    ApiKey: process.env.NAMECHEAP_API_KEY,
    UserName: process.env.NAMECHEAP_API_USER,
    Command: 'namecheap.domains.dns.setHosts',
    ClientIp: process.env.NAMECHEAP_IP_WHITELIST,
    SLD: 'budstack',
    TLD: 'io',
    HostName1: subdomain,
    RecordType1: 'CNAME',
    Address1: 'budstack.to',
    TTL1: '1800'
  };
```

```
  const response = await fetch(apiUrl + '?' + new URLSearchParams(params));
  return await response.text();
}
```

API Functions

1. Create CNAME Record - Creates subdomain pointing to main domain - Automatic during tenant onboarding

2. Update DNS Record - Modify existing subdomain - Change target address

3. Delete DNS Record - Remove subdomain - When tenant is deleted

4. List All Records - View all DNS records - Audit and troubleshooting

Testing API Connection

Test Script:

```
cd /home/ubuntu/healingbuds_website/nextjs_space
node test-namecheap-api.js
```

Expected Output:

```
  API connection successful
  CNAME record created for test-subdomain
  DNS propagating (may take 5-30 minutes)
```

Custom Domain Setup

Overview

Tenants can use their own domain (e.g., `healingbuds.pt`) instead of subdomain.

Process for Tenants

Step 1: Purchase Domain - Buy from any registrar (GoDaddy, Namecheap, etc.) - Choose relevant domain name - Complete registration

Step 2: Request Custom Domain - Contact BudStack support - Provide domain name - Verify ownership

Step 3: Configure DNS Tenant must add these DNS records:

A Record:

```
Type:  A
Host:  @
Value: [Platform Server IP]
TTL:   Automatic or 3600
```

CNAME Record (www):

```
Type:  CNAME
Host:  www
Value: [tenant-slug].budstack.to
TTL:   Automatic or 3600
```

Step 4: Verification - BudStack verifies DNS configuration - SSL certificate automatically generated - Domain activated (24-48 hours)

DNS Configuration Examples

GoDaddy: 1. Log in to GoDaddy 2. Go to "My Products" → "DNS" 3. Click "Manage" next to domain 4. Add A and CNAME records 5. Save changes

Namecheap: 1. Log in to Namecheap 2. Domain List → Manage 3. Advanced DNS tab 4. Add New Record 5. Save All Changes

Cloudflare: 1. Add site to Cloudflare 2. DNS Management 3. Add A and CNAME records 4. Set proxy status (orange cloud) 5. Save

---

DNS Configuration

Main Domain Setup

budstack.to DNS Records:

```
Type: A
Host: @
Value: [Server IP]

Type: A
Host: *
Value: [Server IP]
(This creates wildcard for all subdomains)

Type: MX
Host: @
Value: mail.budstack.to
Priority: 10

Type: TXT
Host: @
Value: "v=spf1 include:_spf.google.com ~all"
```

Subdomain DNS Structure

Automatic Creation: When tenant approved:

```
Type: CNAME
Host: healingbuds
Value: budstack.to
TTL: 1800 (30 minutes)
```

Result: - healingbuds.budstack.to → resolves to server IP - Traffic routed to correct tenant - Middleware identifies tenant by subdomain

DNS Propagation

Timeline: - Local cache: Immediate - ISP cache: 5-30 minutes - Global propagation: Up to 48 hours (typically 2-4 hours)

Check Propagation:

```
# Check DNS resolution
nslookup healingbuds.budstack.to
```

```
# Check from multiple locations
https://www.whatsmydns.net/#A/healingbuds.budstack.to
```

SSL Certificates

Automatic SSL

Let's Encrypt Integration: - SSL certificates auto-generated - Covers all subdomains (wildcard cert) - Auto-renewal every 90 days - Free of charge

Certificate Management

Main Certificate:

```
Domain: budstack.to
Alt Names: *.budstack.to, www.budstack.to
Issuer: Let's Encrypt
Valid: 90 days
```

Custom Domain Certificates: - Generated per custom domain - Automatic via ACME challenge - Managed by hosting provider

HTTPS Enforcement

Configuration:

```javascript
// next.config.js
module.exports = {
  async redirects() {
    return [
      {
        source: '/:path*',
        has: [
          {
            type: 'header',
            key: 'x-forwarded-proto',
            value: 'http',
          },
        ],
        destination: 'https://budstack.to/:path*',
        permanent: true,
      },
    ];
  },
};
```

Verifying SSL

Browser Check: 1. Visit `https://healingbuds.budstack.to` 2. Click padlock icon in address bar 3. View certificate details 4. Verify issuer and validity

Command Line:

```
openssl s_client -connect healingbuds.budstack.to:443 -servername healingbuds.budstack.to
```

Troubleshooting

Common Issues

Subdomain Not Resolving

Symptoms: - "Site can't be reached" - DNS_PROBE_FINISHED_NXDOMAIN

Solutions: 1. Check DNS propagation (may take 30 minutes) 2. Clear DNS cache: `ipconfig /flushdns` (Windows) or `sudo dscacheutil -flushcache` (Mac) 3. Try different browser/device 4. Verify CNAME record created in Namecheap 5. Check wildcard DNS for `*.budstack.to`

SSL Certificate Error

Symptoms: - "Your connection is not private" - NET::ERR_CERT_COMMON_NAME_INVALID

Solutions: 1. Wait for certificate generation (up to 1 hour) 2. Verify DNS is fully propagated 3. Check certificate includes subdomain 4. Contact support to regenerate certificate

API Connection Failed

Symptoms: - "Failed to create subdomain" - "API authentication error"

Solutions: 1. Verify API credentials in `.env` 2. Check server IP is whitelisted 3. Confirm API access enabled on Namecheap 4. Test API connection with test script 5. Review API error logs

Custom Domain Not Working

Symptoms: - Domain shows error page - Redirects to wrong site - SSL error on custom domain

Solutions: 1. Verify A record points to correct IP 2. Check CNAME for www subdomain 3. Confirm DNS propagation complete 4. Request SSL certificate regeneration 5. Check domain is added to tenant settings

DNS Debugging Commands

```
# Check A record
dig budstack.to A

# Check CNAME record
dig healingbuds.budstack.to CNAME

# Trace DNS resolution
dig +trace healingbuds.budstack.to

# Check from specific DNS server
dig @8.8.8.8 healingbuds.budstack.to
```

Monitoring

DNS Health Check:

```
# Check all subdomains resolve
for subdomain in healingbuds greenmedicine test; do
  echo "Checking $subdomain.budstack.to"
  dig +short $subdomain.budstack.to
done
```

SSL Expiration Monitoring:

```
# Check certificate expiration
echo | openssl s_client -servername budstack.to -connect budstack.to:443 2>/dev/null | openssl x509 -no
```

Best Practices

Subdomain Management

Do: - Use descriptive, memorable subdomains - Keep subdomains short when possible - Test subdomain before tenant approval - Document all subdomains

Don't: - Reuse deleted subdomains immediately - Use offensive or inappropriate names - Create overly long subdomains - Change subdomain after launch

DNS Management

Do: - Set appropriate TTL values (1800-3600 seconds) - Monitor DNS health regularly - Keep DNS records documented - Use DNS monitoring service

Don't: - Set TTL too low (increases load) - Set TTL too high (delays changes) - Modify production DNS without testing - Delete records without verification

Security

Do: - Enforce HTTPS everywhere - Monitor certificate expiration - Keep API keys secure - Rotate API keys periodically - Use environment variables for secrets

Don't: - Commit API keys to version control - Share API keys insecurely - Allow HTTP traffic - Ignore certificate warnings

---

Quick Reference

Namecheap API Endpoints

```
Base URL: https://api.namecheap.com/xml.response

Commands:
- namecheap.domains.dns.setHosts (Create/update DNS)
- namecheap.domains.dns.getHosts (List DNS records)
- namecheap.domains.getList (List domains)
```

DNS Record Types

```
A Record:     Maps domain to IP address
CNAME:        Maps domain to another domain
MX Record:    Mail server routing
TXT Record:   Text data (SPF, DKIM, verification)
AAAA Record:  IPv6 address
```

Useful Tools

- **DNS Checker:** https://dnschecker.org
- **SSL Checker:** https://www.sslshopper.com/ssl-checker.html
- **Subdomain Finder:** https://subdomainfinder.c99.nl
- **DNS Propagation:** https://www.whatsmydns.net

Support

Contact Information

Technical Support: - Email: support@budstack.to - Hours: Monday-Friday, 9 AM - 6 PM PST

Emergency DNS Issues: - Priority support for production outages - Include: tenant name, subdomain, error message

Namecheap Support: - If API issues persist - Namecheap Live Chat - Have API username ready

---

Document Version: 1.0
Last Updated: November 2025